aws

Benutzerhandbuch

Amazon EKS

Amazon EKS: Benutzerhandbuch

Table of Contents

Was ist Amazon EKS?

Amazon Elastic Kubernetes Service (Amazon EKS) ist ein verwalteter Service, der die Installation, den Betrieb und die Wartung Ihrer eigenen Kubernetes-Steuerebene in Amazon Web Services (AWS) überflüssig macht .Kubernetes ist ein Open-Source-System, das die Verwaltung, Skalierung und Bereitstellung von containerisierten Anwendungen automatisiert.

Features von Amazon EKS

Nachfolgend sind die wichtigsten Features von Amazon EKS aufgeführt:

Sicheres Netzwerk und Authentifizierung

Amazon EKS integriert Ihre Kubernetes Workloads mit AWS Netzwerk - und Sicherheitsservices. Es lässt sich auch in AWS Identity and Access Management (IAM) integrieren, um die Authentifizierung für Ihre Kubernetes Cluster zu ermöglichen.

Einfache Cluster-Skalierung

Amazon EKS ermöglicht Ihnen auch die einfache Hoch- und Herunterskalierung Ihrer Kubernetes-Cluster je nach Bedarf Ihrer Workloads. Amazon EKS unterstützt horizontale Pod-Autoskalierung basierend auf CPU- oder benutzerdefinierten Metriken sowie Cluster-Autoskalierung basierend auf dem Bedarf der gesamten Workload.

Verwaltete Kubernetes-Erfahrung

Sie können Änderungen an Ihren Kubernetes-Clustern mithilfe von `eksctl`, AWS Management Console, AWS Command Line Interface (AWS CLI), der API, `kubectl` und Terraform vornehmen.

Hohe Verfügbarkeit

Amazon EKS bietet hohe Verfügbarkeit für Ihre Steuerebene in mehreren Availability Zones.

Integration mit Diensten AWS

Amazon EKS lässt sich mit anderen AWS -Services integrieren und bietet eine umfassende Plattform für die Bereitstellung und Verwaltung Ihrer containerisierten Anwendungen. Außerdem können Sie Probleme mit Ihren Kubernetes-Workloads mit verschiedenen Beobachtbarkeits-Tools leichter beheben.

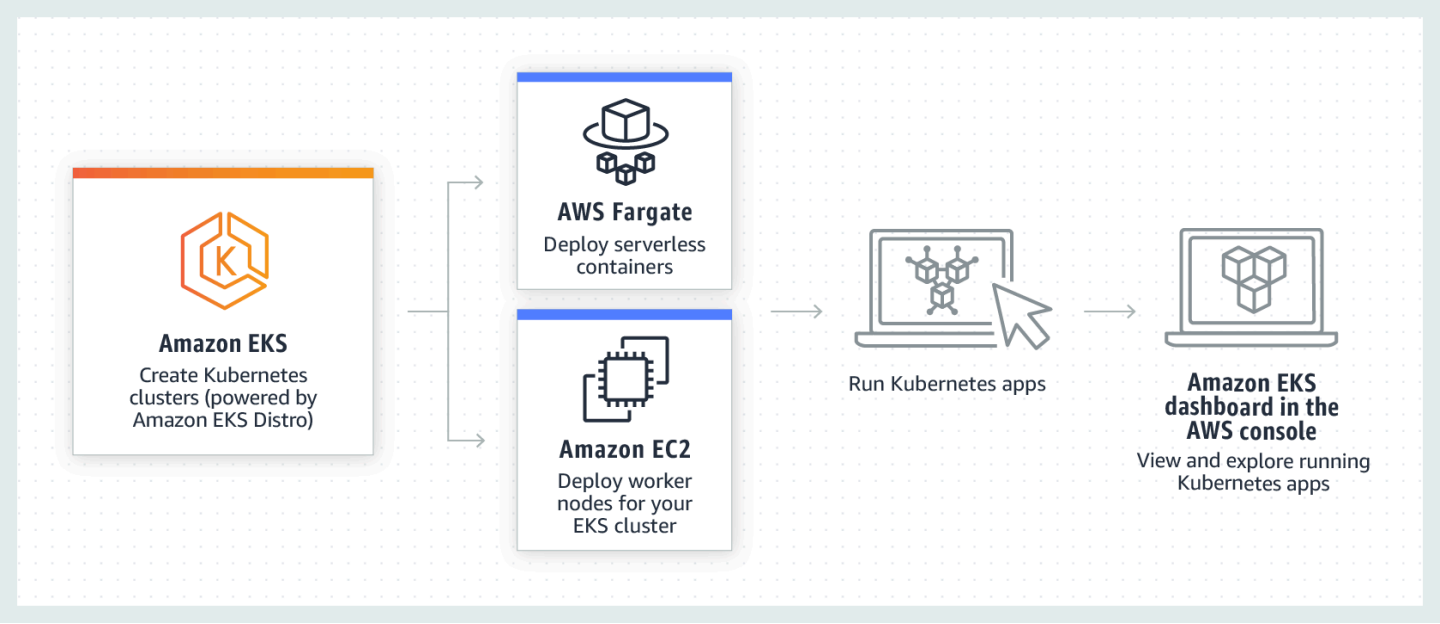
Details zu weiteren Features von Amazon EKS finden Sie unter Amazon EKS – Features.

Erste Schritte mit Amazon EKS

Informationen zum Erstellen Ihres ersten Clusters und der zugehörigen Ressourcen finden Sie unter Erste Schritte mit Amazon EKS. Im Allgemeinen umfasst der Einstieg in Amazon EKS die folgenden Schritte.

1. Einen Cluster erstellen — Erstellen Sie zunächst Ihren Cluster mithilfe von`eksctl`, AWS Management Console AWS CLI, oder einem der AWS SDKs.
2. Wählen Sie Ihren Ansatz zur Berechnung von Ressourcen — Entscheiden Sie sich zwischen AWS Fargate,Karpenter, verwalteten Knotengruppen und selbstverwalteten Knoten.
3. Einrichtung: Richten Sie die erforderlichen Controller, Treiber und Dienste ein.
4. Bereitstellung von Workloads: Passen Sie Ihre Kubernetes-Workloads an, um die Ressourcen und Funktionen des von Ihnen ausgewählten Knotentyps optimal zu nutzen.
5. Verwaltung: Überwachen Sie Ihre Workloads und integrieren Sie AWS -Services zur Rationalisierung von Abläufen und zur Verbesserung der Workload-Leistung. Sie können Informationen zu Ihren Workloads mit dem anzeigen. AWS Management Console

Das folgende Diagramm zeigt einen grundlegenden Ablauf der Ausführung von Amazon EKS in der Cloud. Informationen zu anderen Kubernetes-Bereitstellungsoptionen finden Sie unter Optionen für die Bereitstellung.



Preismodell für Amazon EKS

Ein Amazon-EKS-Cluster besteht aus einer Steuerebene und der Amazon Elastic Compute Cloud (Amazon EC2)- oder Fargate-Datenverarbeitung, für die Sie Pods ausführen. Weitere Informationen zu Preisen für die Steuerebene finden Sie unter Amazon EKS Sowohl Amazon EC2 als auch Fargate bieten:

On-Demand Instances

Sie zahlen für die von Ihnen verwendeten Instances nach Sekunde, ohne langfristige Verpflichtungen eingehen oder Vorauszahlungen leisten zu müssen. Weitere Informationen finden Sie unter Amazon EC2 – On-Demand-Preise und AWS Fargate – Preise.

, Savings Plans

Sie können Ihre Kosten reduzieren, indem Sie sich auf eine konsistente Nutzung (in USD/h) für einen Zeitraum von einem oder drei Jahren festlegen. Weitere Informationen finden Sie unter Preise und Savings Plans.

Häufige Anwendungsfälle in Amazon EKS

Amazon EKS bietet stabile verwaltete Kubernetes-Services in AWS, die für die Optimierung von containerisierten Anwendungen entwickelt wurden. Im Folgenden finden Sie einige der häufigsten Anwendungsfälle von Amazon EKS, die Ihnen dabei helfen, die Stärken von Amazon EKS für Ihre spezifischen Anforderungen zu nutzen.

Bereitstellung von Anwendungen mit hoher Verfügbarkeit

Mit Elastic Load Balancing können Sie sicherstellen, dass Ihre Anwendungen über mehrere Availability Zones hinweg hochverfügbar sind.

Aufbau von Microservices-Architekturen

Verwenden Sie die Kubernetes-Features zur Serviceerkennung mit AWS Cloud Map oder Amazon VPC Lattice, um ausfallsichere Systeme aufzubauen.

Automatisierung des Software-Veröffentlichungsprozesses

Verwalten Sie Pipelines für fortlaufende Integration und fortlaufende Bereitstellung (Continous Integration and Continous Deployment, CICD), die den Prozess der automatisierten Erstellung, Prüfung und Bereitstellung von Anwendungen vereinfachen.

Ausführung von Serverless-Anwendungen

Verwenden Sie AWS Fargate, um Serverless-Anwendungen auszuführen. Das bedeutet, dass Sie sich ausschließlich auf die Anwendungsentwicklung konzentrieren können, während Amazon EKS und Fargate die zugrunde liegende Infrastruktur verwalten.

Ausführung von Machine-Learning-Workloads

Amazon EKS ist mit gängigen Machine-Learning-Frameworks wie TensorFlow, MXNet und PyTorch kompatibel. Mit GPU-Unterstützung können Sie selbst komplexe Machine-Learning-Aufgaben effektiv bewältigen.

Konsistente Bereitstellung On-Premises und in der Cloud

Verwenden Sie Amazon EKS Anywhere für den Betrieb von Kubernetes-Clustern in Ihrer eigenen Infrastruktur mit Tools, die mit Amazon EKS in der Cloud konsistent sind.

Ausführung kosteneffizienter Batch-Verarbeitungs- und Big-Data-Workloads

Nutzen Sie Spot Instances, um Ihre Batch-Verarbeitungs- und Big-Data-Workloads wie Apache Hadoop und Spark zu einem Bruchteil der Kosten auszuführen. Auf diese Weise können Sie ungenutzte Amazon-EC2-Kapazität zu ermäßigten Preisen nutzen.

Anwendungssicherung und Sicherstellung der Compliance

Implementieren Sie strenge Sicherheitspraktiken und sorgen Sie für Compliance mit Amazon EKS, das sich in AWS-Sicherheitsservices wie AWS Identity and Access Management (IAM), Amazon Virtual Private Cloud (Amazon VPC) undAWS Key Management Service (AWS KMS) integrieren lässt. Dadurch werden der Datenschutz und die Datensicherheit gemäß den Branchenstandards gewährleistet.

Amazon-EKS-Architektur

Amazon EKS entspricht der allgemeinen Cluster-Architektur von Kubernetes. Weitere Informationen finden Sie unter Kubernetes Components (Kubernetes-Komponenten) in der Kubernetes-Dokumentation. In den folgenden Abschnitten werden einige zusätzliche Architekturdetails für Amazon EKS zusammengefasst.

Steuerebene

Amazon EKS sorgt dafür sicher, dass jeder Cluster über eine eigene, eindeutige Kubernetes-Steuerebene verfügt. Durch dieses Design bleibt die Infrastruktur jedes Clusters getrennt, ohne dass es zu Überschneidungen zwischen Clustern oder AWS-Konten kommt. Das Setup beinhaltet:

Verteilte Komponenten

Die Steuerebene positioniert mindestens zwei API-Server-Instances und drei `etcd`-Instances in drei AWS Availability Zones innerhalb einer AWS-Region.

Optimale Leistung

Amazon EKS überwacht und passt die Instances der Steuerebene aktiv an, um eine Spitzenleistung aufrechtzuerhalten.

Ausfallsicherheit

Wenn eine Instance der Steuerebene ausfällt, wird sie von Amazon EKS schnell ersetzt, wobei bei Bedarf eine andere Availability Zone verwendet wird.

Konsistente Verfügbarkeit

Durch den Betrieb von Clustern in mehreren Availability Zones wird ein zuverlässiges Service Level Agreement (SLA) für die Verfügbarkeit von API-Server-Endpunkten erzielt.

Amazon EKS verwendet Amazon Virtual Private Cloud (Amazon VPC), um den Datenverkehr zwischen Komponenten der Steuerebene innerhalb eines einzelnen Clusters zu begrenzen. Clusterkomponenten können keine Kommunikation von anderen Clustern oder AWS-Konten anzeigen oder empfangen, es sei denn, sie sind durch rollenbasierte Kubernetes-Zugriffskontrollrichtlinien (RBAC) autorisiert.

Datenverarbeitung

Zusätzlich zur Steuerebene verfügt ein Amazon-EKS-Cluster über eine Reihe von Worker-Computern, die als Knoten bezeichnet werden. Die Auswahl des geeigneten Amazon-EKS-Cluster-Knotentyps ist entscheidend, um Ihre spezifischen Anforderungen zu erfüllen und die Ressourcennutzung zu optimieren. Amazon EKS bietet die folgenden Primärknotentypen:

AWS Fargate

Fargate ist eine Serverless-Compute-Engine für Container, die die Verwaltung der zugrunde liegenden Instances überflüssig macht. Mit Fargate geben Sie den Ressourcenbedarf Ihrer Anwendung an, und AWS sorgt automatisch für die Bereitstellung, Skalierung und Wartung der Infrastruktur. Diese Option ist ideal für Benutzer, die Wert auf Benutzerfreundlichkeit legen und sich auf die Anwendungsentwicklung und -bereitstellung konzentrieren möchten, anstatt die Infrastruktur zu verwalten.

Karpenter

Karpenter ist ein flexibler, hochleistungsfähiger Kubernetes Cluster Autoscaler, der die Anwendungsverfügbarkeit und Clustereffizienz verbessert. Karpenter startet Datenverarbeitungsressourcen in der richtigen Größe als Reaktion auf die sich ändernde Anwendungslast. Mit dieser Option können Just-in-Time-Datenverarbeitungsressourcen bereitgestellt werden, die den Anforderungen Ihrer Workload entsprechen.

Verwaltete Knotengruppen

Verwaltete Knotengruppen sind eine Mischung aus Automatisierung und Anpassung für die Verwaltung einer Sammlung von Amazon-EC2-Instances innerhalb eines Amazon-EKS-Clusters. AWS kümmert sich um Aufgaben wie das Patchen, Aktualisieren und Skalieren von Knoten und vereinfacht so die betrieblichen Aspekte. Gleichzeitig werden benutzerdefinierte `kubelet`-Argumente unterstützt, die Möglichkeiten für erweiterte CPU- und Speicherverwaltungsrichtlinien eröffnen. Darüber hinaus wird die Sicherheit über AWS Identity and Access Management (IAM)-Rollen für Servicekonten verbessert, während gleichzeitig die Notwendigkeit separater Berechtigungen pro Cluster verringert wird.

Selbstverwaltete Knoten

Selbstverwaltete Knoten bieten volle Kontrolle über Ihre Amazon-EC2-Instances innerhalb eines Amazon-EKS-Clusters. Sie sind für die Verwaltung, Skalierung und Wartung der Knoten zuständig und haben somit die volle Kontrolle über die zugrunde liegende Infrastruktur. Diese Option eignet sich für Benutzer, die eine differenzierte Kontrolle und Anpassung ihrer Knoten benötigen und bereit sind, Zeit in die Verwaltung und Wartung ihrer Infrastruktur zu investieren.

Kubernetes-Konzepte

Amazon Elastic Kubernetes Service (Amazon EKS) ist ein AWS verwalteter Service, der auf dem KubernetesOpen-Source-Projekt basiert. Es gibt zwar Dinge, die Sie darüber wissen müssen, wie

der Amazon EKS-Service in die AWS Cloud integriert wird (insbesondere, wenn Sie zum ersten Mal einen Amazon EKS-Cluster erstellen), aber sobald er betriebsbereit ist, verwenden Sie Ihren Amazon EKS-Cluster auf die gleiche Weise wie jeden anderen Kubernetes Cluster. Um mit der Verwaltung von Kubernetes Clustern und der Bereitstellung von Workloads zu beginnen, benötigen Sie also zumindest ein grundlegendes Verständnis der Kubernetes Konzepte.

Auf dieser Seite sind die Kubernetes Konzepte in drei Abschnitte unterteilt: WarumKubernetes, Cluster und Workloads. Der erste Abschnitt beschreibt den Wert der Ausführung eines Kubernetes Dienstes, insbesondere als verwalteter Service wie Amazon EKS. Im Abschnitt Workloads wird beschrieben, wie Kubernetes Anwendungen erstellt, gespeichert, ausgeführt und verwaltet werden. Im Abschnitt Cluster werden die verschiedenen Komponenten beschrieben, aus denen Kubernetes Cluster bestehen, und Ihre Aufgaben für die Erstellung und Verwaltung von Kubernetes Clustern.

Themen

Wenn Sie diesen Inhalt durchgehen, führen Sie Links zu weiteren Beschreibungen der Kubernetes Konzepte in Amazon EKS und in der Kubernetes Dokumentation, falls Sie sich eingehend mit einem der Themen befassen möchten, die wir hier behandeln. Einzelheiten darüber, wie Amazon EKS Kubernetes Steuerungsebenen- und Rechenfunktionen implementiert, finden Sie unter Amazon EKS-Architektur.

Warum Kubernetes?

Kuberneteswurde entwickelt, um die Verfügbarkeit und Skalierbarkeit bei der Ausführung geschäftskritischer containerisierter Anwendungen in Produktionsqualität zu verbessern. Anstatt nur Kubernetes auf einem einzigen Computer zu laufen (obwohl das möglich ist), werden diese Ziele Kubernetes dadurch erreicht, dass Sie Anwendungen auf mehreren Computern ausführen können, die je nach Bedarf erweitert oder verkleinert werden können. Kubernetesumfasst Funktionen, die Ihnen Folgendes erleichtern:

- Stellen Sie Anwendungen auf mehreren Computern bereit (mithilfe von Containern, die in Pods bereitgestellt werden)
- Überwachen Sie den Zustand der Container und starten Sie ausgefallene Container neu

- Skalieren Sie Container je nach Auslastung nach oben oder unten
- Aktualisieren Sie Container mit neuen Versionen
- Ordnen Sie Ressourcen zwischen Containern zu
- Gleichen Sie den Verkehr zwischen den Maschinen aus

Durch die Kubernetes Automatisierung dieser Art komplexer Aufgaben können sich Anwendungsentwickler auf den Aufbau und die Verbesserung ihrer Anwendungs-Workloads konzentrieren, anstatt sich Gedanken über die Infrastruktur machen zu müssen. Der Entwickler erstellt in der Regel als YAML-Dateien formatierte Konfigurationsdateien, die den gewünschten Status der Anwendung beschreiben. Dies kann beinhalten, welche Container ausgeführt werden sollen, Ressourcenlimits, Anzahl der Pod-Replikate, CPU-/Speicherzuweisung, Affinitätsregeln und mehr.

Attribute von Kubernetes

Um seine Ziele zu erreichen, Kubernetes hat die folgenden Eigenschaften:

- Containerized — Kubernetes ist ein Tool zur Container-Orchestrierung. Um es verwenden zu könnenKubernetes, müssen Sie zuerst Ihre Anwendungen containerisiert haben. Je nach Art der Anwendung kann es sich dabei um eine Reihe von Microservices, als Batch-Jobs oder in anderen Formen handeln. Anschließend können Ihre Anwendungen von einem Kubernetes Workflow profitieren, der ein riesiges Ökosystem von Tools umfasst, in dem Container als Images in einer Container-Registry gespeichert, in einem KubernetesCluster bereitgestellt und auf einem verfügbaren Knoten ausgeführt werden können. Sie können einzelne Container auf Ihrem lokalen Computer mit Docker oder einer anderen Container-Runtime erstellen und testen, bevor Sie sie in Ihrem Kubernetes Cluster bereitstellen.
- Skalierbar — Wenn die Nachfrage nach Ihren Anwendungen die Kapazität der laufenden Instanzen dieser Anwendungen übersteigt, Kubernetes ist eine Skalierung möglich. KubernetesKann bei Bedarf feststellen, ob Anwendungen mehr CPU oder Arbeitsspeicher benötigen, und reagiert darauf, indem entweder die verfügbare Kapazität automatisch erweitert oder mehr der vorhandenen Kapazität genutzt wird. Die Skalierung kann auf Pod-Ebene erfolgen, wenn genügend Rechenleistung zur Verfügung steht, um einfach mehr Instanzen der Anwendung auszuführen (horizontales Pod-Autoscaling), oder auf Knotenebene, wenn mehr Knoten eingerichtet werden müssen, um die erhöhte Kapazität zu bewältigen (Cluster Autoscaler oder Karpenter). Da keine Kapazität mehr benötigt wird, können diese Dienste nicht benötigte Pods löschen und nicht benötigte Knoten herunterfahren.

- Verfügbar — Wenn eine Anwendung oder ein Knoten fehlerhaft oder nicht verfügbar ist, Kubernetes können laufende Workloads auf einen anderen verfügbaren Knoten verschoben werden. Sie können das Problem erzwingen, indem Sie einfach eine laufende Instanz eines Workloads oder Knotens löschen, auf dem Ihre Workloads ausgeführt werden. Die Quintessenz dabei ist, dass Workloads an anderen Standorten geladen werden können, wenn sie nicht mehr dort ausgeführt werden können, wo sie sich befinden.
- Deklarativ — Kubernetes verwendet einen aktiven Abgleich, um ständig zu überprüfen, ob der Status, den Sie für Ihren Cluster deklarieren, dem tatsächlichen Status entspricht. Indem Sie KubernetesObjekte auf einen Cluster anwenden, in der Regel mithilfe von Konfigurationsdateien im YAML-Format, können Sie beispielsweise verlangen, dass die Workloads, die Sie auf Ihrem Cluster ausführen möchten, gestartet werden. Sie können die Konfigurationen später ändern, um beispielsweise eine neuere Version eines Containers zu verwenden oder mehr Speicher zuzuweisen. Kuberneteswird alles Notwendige tun, um den gewünschten Status herzustellen. Dies kann das Hoch- oder Herunterfahren von Knoten, das Stoppen und Neustarten von Workloads oder das Abrufen aktualisierter Container beinhalten.
- Zusammensetzbar — Da eine Anwendung in der Regel aus mehreren Komponenten besteht, möchten Sie in der Lage sein, einen Satz dieser Komponenten (häufig dargestellt durch mehrere Container) gemeinsam zu verwalten. DockerCompose bietet zwar eine Möglichkeit, dies direkt mit zu tunDocker, aber der Befehl Kubernetes Kompose kann Ihnen dabei helfen. Kubernetes Ein Beispiel dafür finden Sie unter Translate einer Docker Compose-Datei in Kubernetes Ressourcen.
- Erweiterbar — Im Gegensatz zu proprietärer Software ist das Kubernetes Open-Source-Projekt so konzipiert, dass es Ihnen offen steht und auf Kubernetes jede Art und Weise erweitert werden kann, wie Sie möchten, um Ihren Bedürfnissen gerecht zu werden. APIs und Konfigurationsdateien können direkt geändert werden. Drittanbieter werden ermutigt, ihre eigenen Controller zu schreiben, um sowohl die Infrastruktur als auch die Funktionen für Endbenutzer Kubernetes zu erweitern. Mit Webhooks können Sie Clusterregeln einrichten, um Richtlinien durchzusetzen und sich an sich ändernde Bedingungen anzupassen. Weitere Ideen zur Erweiterung von Kubernetes Clustern finden Sie unter Erweitern Kubernetes.
- Portierbar — Viele Unternehmen haben ihre Abläufe auf standardisiert, Kubernetes weil sie damit alle ihre Anwendungsanforderungen auf die gleiche Weise verwalten können. Entwickler können dieselben Pipelines verwenden, um containerisierte Anwendungen zu erstellen und zu speichern. Diese Anwendungen können dann in Kubernetes Clustern bereitgestellt werden, die vor Ort, in Clouds, auf point-of-sales Terminals in Restaurants oder auf IOT-Geräten laufen, die über die Remote-Standorte des Unternehmens verteilt sind. Aufgrund seines Open-Source-Charakters können Benutzer diese speziellen Kubernetes Distributionen zusammen mit den Tools, die zu ihrer Verwaltung benötigt werden, entwickeln.

Verwalten von Kubernetes

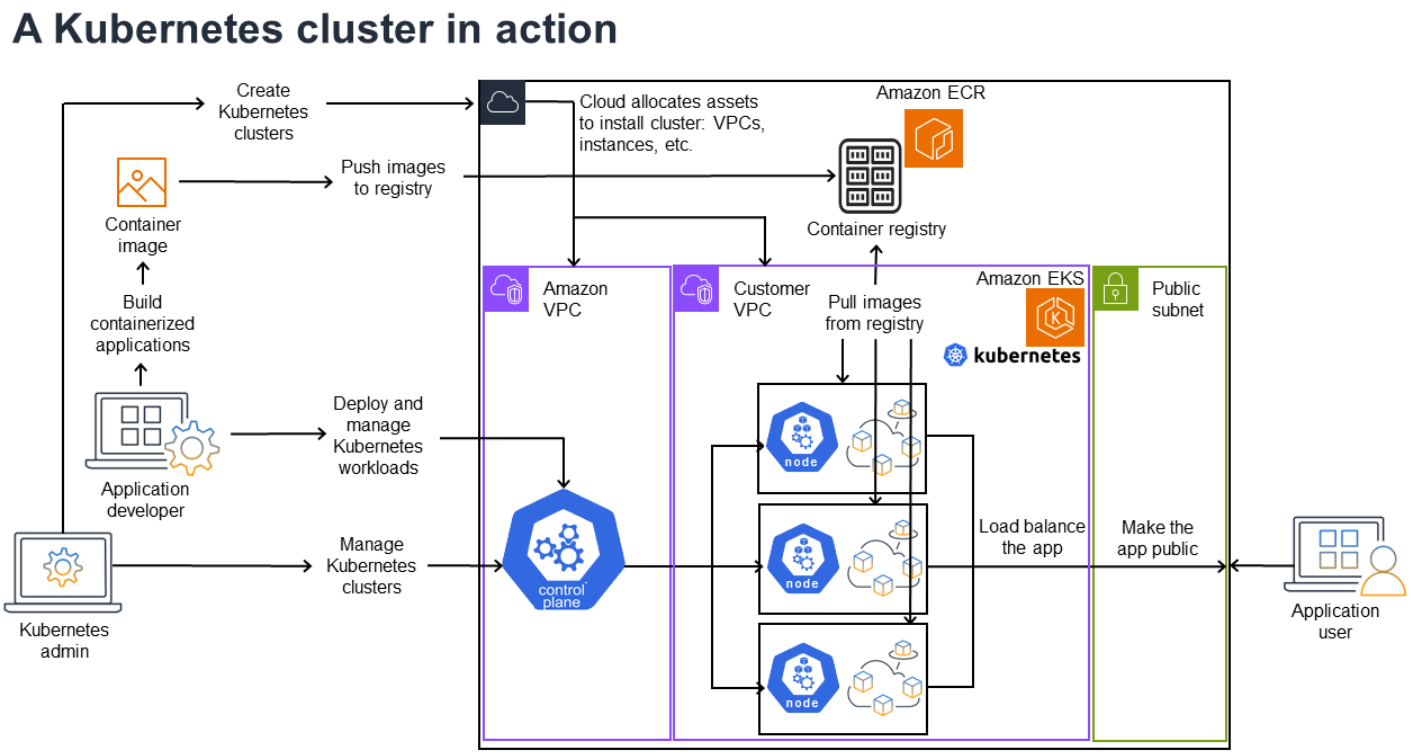
KubernetesDer Quellcode ist frei verfügbar, sodass Sie ihn mit Ihren eigenen Geräten Kubernetes selbst installieren und verwalten können. Die Selbstverwaltung Kubernetes erfordert jedoch umfangreiche betriebliche Fachkenntnisse und erfordert Zeit und Mühe bei der Wartung. Aus diesen Gründen entscheiden sich die meisten Benutzer, die Produktionsworkloads bereitstellen, für einen Cloud-Anbieter (wie Amazon EKS) oder einen lokalen Anbieter (wie Amazon EKS Anywhere) mit eigener getesteter Kubernetes Distribution und Support durch Experten. Kubernetes Auf diese Weise können Sie einen Großteil der undifferenzierten Schwerarbeit entlasten, die für die Wartung Ihrer Cluster erforderlich sind, darunter:

- Hardware — Wenn Sie keine Hardware zur Verfügung haben, die Kubernetes Ihren Anforderungen entspricht, kann Ihnen ein Cloud-Anbieter wie AWS Amazon EKS Vorabkosten sparen. Mit Amazon EKS bedeutet dies, dass Sie die besten Cloud-Ressourcen nutzen können, die Ihnen zur Verfügung stehen AWS, darunter Computer-Instances (Amazon Elastic Compute Cloud), Ihre eigene private Umgebung (Amazon VPC), zentrales Identitäts- und Berechtigungsmanagement (IAM) und Speicher (Amazon EBS). AWS verwaltet die Computer, Netzwerke, Rechenzentren und alle anderen physischen Komponenten, die für den Betrieb erforderlich sind. Kubernetes Ebenso müssen Sie Ihr Rechenzentrum nicht so planen, dass es die maximale Kapazität an den Tagen mit der höchsten Nachfrage bewältigen kann. Bei Amazon EKS Anywhere oder anderen lokalen Kubernetes Clustern sind Sie für die Verwaltung der in Ihren Kubernetes Bereitstellungen verwendeten Infrastruktur verantwortlich. Sie können sich aber trotzdem darauf verlassen, dass wir Ihnen helfen, auf dem AWS neuesten Kubernetes Stand zu bleiben.
- Verwaltung der Kontrollebene — Amazon EKS verwaltet die Sicherheit und Verfügbarkeit der AWS gehosteten Kubernetes Steuerungsebene, die für die Planung von Containern, die Verwaltung der Verfügbarkeit von Anwendungen und andere wichtige Aufgaben verantwortlich ist, sodass Sie sich auf Ihre Anwendungs-Workloads konzentrieren können. Sollte Ihr Cluster ausfallen, AWS sollten Sie über die Mittel verfügen, Ihren Cluster wieder in den Betriebszustand zu versetzen. Für Amazon EKS Anywhere würden Sie die Steuerungsebene selbst verwalten.
- Getestete Upgrades — Wenn Sie Ihre Cluster aktualisieren, können Sie sich darauf verlassen, dass Amazon EKS oder Amazon EKS Anywhere getestete Versionen ihrer Kubernetes Distributionen bereitstellen.
- Add-Ons — Es gibt Hunderte von Projekten, die zur Erweiterung und Bearbeitung konzipiert wurden und Kubernetes die Sie zur Infrastruktur Ihres Clusters hinzufügen oder zur Unterstützung der Ausführung Ihrer Workloads verwenden können. Anstatt diese Add-Ons selbst zu erstellen und zu verwalten, AWS bietet Amazon EKS Add-ons, die Sie mit Ihren Clustern verwenden können. Amazon EKS Anywhere bietet kuratierte Pakete, die Builds vieler beliebter Open-Source-Projekte

enthalten. Sie müssen die Software also nicht selbst erstellen oder wichtige Sicherheitspatches, Bugfixes oder Upgrades verwalten. Wenn die Standardeinstellungen Ihren Anforderungen entsprechen, ist es ebenfalls typisch, dass nur eine sehr geringe Konfiguration dieser Add-Ons erforderlich ist. Einzelheiten zur Erweiterung Ihres Clusters mit Add-Ons finden Sie unter Cluster erweitern.

Kubernetesin Aktion

Das folgende Diagramm zeigt die wichtigsten Aktivitäten, die Sie als Kubernetes Administrator oder Anwendungsentwickler ausführen würden, um einen Kubernetes Cluster zu erstellen und zu verwenden. Dabei wird veranschaulicht, wie Kubernetes Komponenten miteinander interagieren, wobei die AWS Cloud als Beispiel für den zugrunde liegenden Cloud-Anbieter verwendet wird.



Ein Kubernetes Administrator erstellt den Kubernetes Cluster mit einem Tool, das für den Anbietertyp spezifisch ist, auf dem der Cluster aufgebaut werden soll. In diesem Beispiel wird die AWS Cloud als Anbieter verwendet, der den verwalteten Kubernetes Service Amazon EKS anbietet. Der verwaltete Service weist automatisch die Ressourcen zu, die für die Erstellung des Clusters benötigt werden, einschließlich der Erstellung von zwei neuen Virtual Private Clouds (Amazon VPCs) für den Cluster, der Einrichtung von Netzwerken, der Zuordnung von Kubernetes Berechtigungen zu diesen zur Verwaltung von Ressourcen in der Cloud, der Sicherstellung, dass die Dienste der Kontrollebene

über Orte zum Ausführen von Workloads verfügen, und der Zuweisung von null oder mehr Amazon EC2 EC2-Instances als Kubernetes Knoten für die Ausführung von Workloads. AWS verwaltet eine Amazon-VPC selbst für die Kontrollebene, während die andere Amazon-VPC die Kundenknoten enthält, die Workloads ausführen.

Viele der zukünftigen Aufgaben des Kubernetes Administrators werden mithilfe von Kubernetes Tools wie kubectl erledigt. Dieses Tool sendet Anfragen für Dienste direkt an die Steuerungsebene des Clusters. Die Art und Weise, wie Abfragen und Änderungen am Cluster vorgenommen werden, ist dann der Art und Weise, wie Sie sie in jedem Kubernetes Cluster durchführen würden, sehr ähnlich.

Ein Anwendungsentwickler, der Workloads für diesen Cluster bereitstellen möchte, kann mehrere Aufgaben ausführen. Der Entwickler muss die Anwendung in ein oder mehrere Container-Images integrieren und diese Images dann in eine Container-Registry übertragen, auf die der Kubernetes Cluster zugreifen kann. AWS bietet zu diesem Zweck das Amazon Elastic Container Registry (Amazon ECR) an.

Um die Anwendung auszuführen, kann der Entwickler Konfigurationsdateien im YAML-Format erstellen, die dem Cluster mitteilen, wie die Anwendung ausgeführt werden soll, einschließlich der Container, die aus der Registrierung abgerufen werden sollen und wie diese Container in Pods verpackt werden sollen. Die Steuerungsebene (Scheduler) plant die Container für einen oder mehrere Knoten, und die Container-Laufzeit auf jedem Knoten ruft die benötigten Container ab und führt sie aus. Der Entwickler kann auch einen Application Load Balancer einrichten, der den Datenverkehr auf die verfügbaren Container verteilt, die auf jedem Knoten ausgeführt werden, und die Anwendung so verfügbar macht, dass sie in einem öffentlichen Netzwerk für die Außenwelt verfügbar ist. Wenn das alles erledigt ist, kann sich jemand, der die Anwendung verwenden möchte, mit dem Anwendungsendpunkt verbinden, um darauf zuzugreifen.

Im folgenden Abschnitt werden die einzelnen Funktionen aus der Sicht von Kubernetes Clustern und Workloads detailliert beschrieben.

Cluster

Wenn Ihre Aufgabe darin besteht, Kubernetes Cluster zu starten und zu verwalten, sollten Sie wissen, wie Kubernetes Cluster erstellt, erweitert, verwaltet und gelöscht werden. Sie sollten auch wissen, aus welchen Komponenten ein Cluster besteht und was Sie tun müssen, um diese Komponenten zu verwalten.

Tools für die Verwaltung von Clustern berücksichtigen die Überschneidung zwischen den Kubernetes Diensten und dem zugrunde liegenden Hardwareanbieter. Aus diesem Grund erfolgt die

Automatisierung dieser Aufgaben in der Regel durch den Kubernetes Anbieter (wie Amazon EKS oder Amazon EKS Anywhere) mithilfe von anbieterspezifischen Tools. Um beispielsweise einen Amazon EKS-Cluster zu starten, können Sie ihn verwenden`eksctl create cluster`, während Sie Amazon EKS Anywhere verwenden können`eksctl anywhere create cluster`. Beachten Sie, dass diese Befehle zwar einen Kubernetes Cluster erstellen, aber spezifisch für den Anbieter sind und nicht Teil des Kubernetes Projekts selbst sind.

Tools zur Clustererstellung und -verwaltung

Das Kubernetes Projekt bietet Tools für die manuelle Erstellung eines Kubernetes Clusters. Wenn Sie also Kubernetes auf einem einzelnen Computer installieren oder die Steuerungsebene auf einem Computer ausführen und Knoten manuell hinzufügen möchten, können Sie CLI-Tools wie kind, minikube oder kubeadm verwenden, die unter Tools installieren aufgeführt sind. Kubernetes Um den gesamten Lebenszyklus der Clustererstellung und -verwaltung zu vereinfachen und zu automatisieren, ist es viel einfacher, Tools zu verwenden, die von einem etablierten Kubernetes Anbieter wie Amazon EKS oder Amazon EKS Anywhere unterstützt werden.

In AWS Cloud können Sie Amazon EKS-Cluster mithilfe von CLI-Tools wie eksctl oder deklarativeren Tools wie Terraform erstellen (siehe Amazon EKS Blueprints for Terraform). Sie können einen Cluster auch von der Management Console aus erstellen. AWS Eine Liste der Vorteile von Amazon EKS finden Sie unter Amazon EKS-Funktionen. KubernetesZu den Aufgaben, die Amazon EKS für Sie übernimmt, gehören:

- Verwaltete Kontrollebene — AWS stellt sicher, dass der Amazon EKS-Cluster verfügbar und skalierbar ist, da er die Kontrollebene für Sie verwaltet und sie in allen AWS Availability Zones verfügbar macht.
- Knotenverwaltung — Anstatt Knoten manuell hinzuzufügen, können Sie Amazon EKS mithilfe von Managed Node Groups oder Karpenter Knoten bei Bedarf automatisch erstellen lassen. Verwaltete Knotengruppen verfügen über Integrationen mit Kubernetes Cluster Autoscaling. Mithilfe von Tools zur Knotenverwaltung können Sie Kosteneinsparungen wie Spot-Instances und Knotenkonsolidierung sowie Verfügbarkeit nutzen. Mithilfe von Planungsfunktionen können Sie festlegen, wie Workloads bereitgestellt und Knoten ausgewählt werden.
- Clusternetzwerke — Richten Sie mithilfe `eksctl` von CloudFormation Vorlagen die Vernetzung zwischen den Komponenten der Steuerungsebene und der Datenebene (Knoten) im Kubernetes Cluster ein. Außerdem werden Endpunkte eingerichtet, über die interne und externe Kommunikation stattfinden kann. Einzelheiten finden Sie unter Entmystifying Cluster Networking for Amazon EKS Worker Nodes. Die Kommunikation zwischen Pods in Amazon EKS erfolgt

über Amazon EKS Pod Identities, wodurch Pods AWS Cloud-Methoden zur Verwaltung von Anmeldeinformationen und Berechtigungen nutzen können.
- Add-Ons — Amazon EKS erspart Ihnen das Erstellen und Hinzufügen von Softwarekomponenten, die üblicherweise zur Unterstützung von Kubernetes Clustern verwendet werden. Wenn Sie beispielsweise einen Amazon EKS-Cluster von der AWS Management-Konsole aus erstellen, werden automatisch der Amazon EKS Kube-Proxy, das Amazon VPC CNI-Plug-In für Kubernetes und die CoreDNS-Add-Ons hinzugefügt. Weitere Informationen zu diesen Add-Ons, einschließlich einer Liste der verfügbaren Add-Ons, finden Sie unter Amazon EKS-Add-Ons.

Um Ihre Cluster auf Ihren eigenen lokalen Computern und Netzwerken auszuführen, bietet Amazon Amazon EKS Anywhere an. Anstatt dass die AWS Cloud der Anbieter ist, haben Sie die Wahl, Amazon EKS Anywhere auf VMware vSphere -, Bare Metal- (Tinkerbell-Anbieter), Snow - oder Nutanix-Plattformen CloudStackmit Ihren eigenen Geräten auszuführen.

Amazon EKS Anywhere basiert auf derselben Amazon EKS Distro-Software, die auch von Amazon EKS verwendet wird. Amazon EKS Anywhere stützt sich jedoch auf verschiedene Implementierungen der KubernetesCluster-API (CAPI) -Schnittstelle, um den gesamten Lebenszyklus der Maschinen in einem Amazon EKS Anywhere Anywhere-Cluster zu verwalten (wie CAPV für vSphere und CAPC für). CloudStack Da der gesamte Cluster auf Ihren Geräten läuft, übernehmen Sie zusätzlich die Verantwortung für die Verwaltung der Steuerungsebene und die Sicherung ihrer Daten (siehe etcd weiter unten in diesem Dokument).

Cluster-Komponenten

KubernetesDie Clusterkomponenten sind in zwei Hauptbereiche unterteilt: Steuerungsebene und Worker-Knoten. Komponenten der Steuerungsebene verwalten den Cluster und bieten Zugriff auf seine APIs. Worker-Knoten (manchmal auch einfach als Knoten bezeichnet) stellen die Orte dar, an denen die eigentlichen Workloads ausgeführt werden. Knotenkomponenten bestehen aus Diensten, die auf jedem Knoten ausgeführt werden, um mit der Steuerungsebene zu kommunizieren und Container auszuführen. Die Gruppe von Worker-Knoten für Ihren Cluster wird als Datenebene bezeichnet.

Steuerebene

Die Steuerungsebene besteht aus einer Reihe von Diensten, die den Cluster verwalten. Diese Dienste können alle auf einem einzigen Computer ausgeführt werden oder auf mehrere Computer verteilt sein. Intern werden diese als Control Plane Instances (CPIs) bezeichnet. Wie CPIs ausgeführt werden, hängt von der Größe des Clusters und den Anforderungen an die Hochverfügbarkeit ab.

Wenn die Nachfrage im Cluster steigt, kann ein Service auf der Steuerungsebene skaliert werden, um mehr Instanzen dieses Dienstes bereitzustellen, wobei die Anforderungen zwischen den Instanzen ausbalanciert werden.

Zu den Aufgaben, die Komponenten der Kubernetes Steuerungsebene ausführen, gehören:

- Kommunikation mit Cluster-Komponenten (API-Server) — Der API-Server (kube-apiserver) macht die Kubernetes API verfügbar, sodass Anfragen an den Cluster sowohl innerhalb als auch außerhalb des Clusters gestellt werden können. Mit anderen Worten, Anfragen zum Hinzufügen oder Ändern von Objekten eines Clusters (Pods, Dienste, Knoten usw.) können von externen Befehlen stammen, z. B. von `kubectl` Anfragen zum Ausführen eines Pods. Ebenso können Anfragen vom API-Server an Komponenten innerhalb des Clusters gestellt werden, z. B. eine Abfrage des Status eines Pods an den `kubelet` Dienst.
- Daten über den Cluster speichern (etcd key value store) — Der etcd-Dienst spielt die entscheidende Rolle darin, den aktuellen Status des Clusters zu verfolgen. Wenn auf den etcd-Dienst nicht mehr zugegriffen werden könnte, könnten Sie den Status des Clusters nicht aktualisieren oder abfragen, obwohl die Workloads noch eine Weile weiterlaufen würden. Aus diesem Grund verfügen kritische Cluster in der Regel über mehrere Instanzen des etcd-Dienstes mit Lastenausgleich, die gleichzeitig ausgeführt werden und bei Datenverlust oder Beschädigung regelmäßig Backups des etcd-Schlüsselwertspeichers durchführen. Denken Sie daran, dass in Amazon EKS dies alles standardmäßig automatisch für Sie erledigt wird. Amazon EKS Anywhere bietet Anweisungen für die Sicherung und Wiederherstellung von etcd. Sehen Sie sich das etcd-Datenmodell an, um zu erfahren, wie etcd Daten verwaltet.
- Pods für Knoten planen (Scheduler) — Anfragen zum Starten oder Stoppen eines Pods Kubernetes werden an den KubernetesScheduler (kube-scheduler) weitergeleitet. Da ein Cluster mehrere Knoten haben kann, die den Pod ausführen können, ist es Sache des Schedulers, auszuwählen, auf welchem Knoten (oder auf welchen Knoten, im Fall von Replikaten) der Pod ausgeführt werden soll. Wenn nicht genügend Kapazität verfügbar ist, um den angeforderten Pod auf einem vorhandenen Knoten auszuführen, schlägt die Anfrage fehl, sofern Sie keine anderen Vorkehrungen getroffen haben. Diese Bestimmungen könnten die Aktivierung von Diensten wie Managed Node Groups oder Karpenter beinhalten, die automatisch neue Knoten starten können, um die Workloads zu bewältigen.
- Komponenten im gewünschten Zustand belassen (Controller Manager) — Der Kubernetes Controller Manager wird als Daemon-Prozess (kube-controller-manager) ausgeführt, um den Status des Clusters zu überwachen und Änderungen am Cluster vorzunehmen, um die erwarteten Zustände wiederherzustellen. Insbesondere gibt es mehrere Controller, die verschiedene

Kubernetes Objekte überwachen, darunter einen Statefulset-Controller, einen Endpoint-Controller node-lifecycle-controller, einen Cronjob-Controller und andere.

- Cloud-Ressourcen verwalten (Cloud Controller Manager) — Interaktionen zwischen Kubernetes und dem Cloud-Anbieter, der Anfragen für die zugrunde liegenden Rechenzentrumsressourcen ausführt, werden vom Cloud Controller Manager () abgewickelt. cloud-controller-manager Controller, die vom Cloud Controller Manager verwaltet werden, können einen Route-Controller (für die Einrichtung von Cloud-Netzwerkrouten), einen Service-Controller (für die Nutzung von Cloud-Load-Balancing-Diensten) und einen Node-Controller (für die Verwendung von Cloud-APIs, um die Knoten mit den Kubernetes Cloud-Knoten synchron zu halten) umfassen.

Worker-Knoten (Datenebene)

Bei einem Kubernetes Cluster mit einem Knoten laufen die Workloads auf derselben Maschine wie die Steuerungsebene. Eine normalere Konfiguration besteht jedoch darin, über ein oder mehrere separate Computersysteme (Knoten) zu verfügen, die ausschließlich für die Ausführung Kubernetes von Workloads vorgesehen sind.

Wenn Sie einen Kubernetes Cluster zum ersten Mal erstellen, können Sie mit einigen Tools zur Clustererstellung eine bestimmte Anzahl von Knoten konfigurieren, die dem Cluster hinzugefügt werden sollen (entweder indem Sie vorhandene Computersysteme identifizieren oder den Anbieter neue erstellen lassen). Bevor Workloads zu diesen Systemen hinzugefügt werden, werden jedem Knoten Dienste hinzugefügt, um die folgenden Funktionen zu implementieren:

- Jeden Knoten verwalten (Kubelet) — Der API-Server kommuniziert mit dem Kubelet-Dienst, der auf jedem Knoten ausgeführt wird, um sicherzustellen, dass der Knoten ordnungsgemäß registriert ist und die vom Scheduler angeforderten Pods ausgeführt werden. Das Kubelet kann die Pod-Manifeste lesen und Speichervolumes oder andere Funktionen einrichten, die von den Pods auf dem lokalen System benötigt werden. Es kann auch den Zustand der lokal laufenden Container überprüfen.
- Container auf einem Knoten ausführen (Container-Laufzeit) — Die Container-Runtime auf jedem Knoten verwaltet die Container, die für jeden dem Knoten zugewiesenen Pod angefordert werden. Das bedeutet, dass sie Container-Images aus der entsprechenden Registry abrufen, den Container ausführen, stoppen und auf Anfragen zum Container antworten kann. Die Standard-Container-Laufzeit ist containerd. Ab Version Kubernetes 1.24 wurde die spezielle Integration von Docker (Dockershim), die als Container-Laufzeit verwendet werden konnte, gestrichen. Kubernetes Sie können Container zwar weiterhin auf Ihrem lokalen System testen und ausführen, aber um sie

Docker mit zu verwenden, müssten Kubernetes Sie jetzt die DockerEngine auf jedem Knoten installieren, mit dem Sie sie verwenden möchten. Docker Kubernetes

- Netzwerkverwaltung zwischen Containern (Kube-Proxy) — Um die Kommunikation zwischen Pods mithilfe von Diensten unterstützen zu können, war eine Möglichkeit Kubernetes erforderlich, Pod-Netzwerke einzurichten, um IP-Adressen und Ports zu verfolgen, die diesen Pods zugeordnet sind. Der Kube-Proxy-Service wird auf jedem Knoten ausgeführt, um die Kommunikation zwischen den Pods zu ermöglichen.

Erweitern Sie Cluster

Es gibt einige Dienste, die Sie Kubernetes zur Unterstützung des Clusters hinzufügen können, die jedoch nicht auf der Steuerungsebene ausgeführt werden. Diese Dienste werden häufig direkt auf Knoten im Kube-System-Namespace oder in einem eigenen Namespace ausgeführt (wie dies häufig bei Drittanbietern der Fall ist). Ein gängiges Beispiel ist der CoreDNS-Dienst, der DNS-Dienste für den Cluster bereitstellt. Informationen darüber, wie Sie feststellen können, welche Clusterdienste im Kube-System auf Ihrem Cluster ausgeführt werden, finden Sie unter Entdecken integrierter Dienste.

Es gibt verschiedene Arten von Add-Ons, die Sie in Betracht ziehen können, Ihren Clustern hinzuzufügen. Damit Ihre Cluster funktionsfähig bleiben, können Sie Observability-Funktionen hinzufügen, mit denen Sie beispielsweise Logging, Auditing und Metriken durchführen können. Mit diesen Informationen können Sie auftretende Probleme beheben, häufig über dieselben Observability-Schnittstellen. Beispiele für diese Arten von Diensten sind Amazon GuardDuty CloudWatch, AWS Distro for OpenTelemetry, Amazon VPC CNI Plugin für und Grafana Kubernetes Monitoring. Kubernetes Zu den Speichererweiterungen für Amazon EKS gehören der Amazon Elastic Block Store CSI-Treiber (zum Hinzufügen von Blockspeichergeräten), der Amazon Elastic File System CSI-Treiber (zum Hinzufügen von Dateisystemspeicher) und mehrere Speicher-Add-Ons von Drittanbietern (z. B. der Amazon FSx for NetApp ONTAP CSI-Treiber).

Eine vollständigere Liste der verfügbaren Amazon EKS-Add-Ons finden Sie unter Amazon EKS-Add-Ons.

Workloads

Kubernetesdefiniert einen Workload als „eine Anwendung, auf der“ ausgeführt wirdKubernetes. Diese Anwendung kann aus einer Reihe von Microservices bestehen, die als Container in Pods ausgeführt werden, oder sie kann als Batch-Job oder als eine andere Art von Anwendung ausgeführt werden. Die Aufgabe von Kubernetes besteht darin, sicherzustellen, dass die Anfragen, die Sie für die Einrichtung oder Bereitstellung dieser Objekte stellen, ausgeführt werden. Als jemand, der

Anwendungen bereitstellt, sollten Sie wissen, wie Container erstellt werden, wie Pods definiert werden und welche Methoden Sie für deren Bereitstellung verwenden können.

Container

Das grundlegendste Element eines Anwendungs-Workloads, in dem Sie bereitstellen und verwalten, Kubernetes ist ein Pod. Ein Pod stellt eine Möglichkeit dar, die Komponenten einer Anwendung zu speichern und Spezifikationen zu definieren, die die Eigenschaften des Pods beschreiben. Vergleichen Sie das mit etwas wie einem RPM- oder Deb-Paket, das Software für ein Linux-System zusammenpackt, aber selbst nicht als Einheit läuft.

Da der Pod die kleinste bereitstellbare Einheit ist, enthält er in der Regel einen einzelnen Container. In Fällen, in denen die Container eng miteinander verbunden sind, können sich jedoch mehrere Container in einem Pod befinden. Beispielsweise kann ein Webserver-Container in einem Pod mit einem Container vom Typ Sidecar verpackt sein, der Protokollierung, Überwachung oder andere Dienste bereitstellen kann, die eng mit dem Webserver-Container verknüpft sind. In diesem Fall stellt die Tatsache, dass sie sich im selben Pod befinden, sicher, dass für jede laufende Instanz des Pods beide Container immer auf demselben Knoten ausgeführt werden. Ebenso teilen sich alle Container in einem Pod dieselbe Umgebung, wobei die Container in einem Pod so ausgeführt werden, als ob sie sich auf demselben isolierten Host befinden würden. Dies hat zur Folge, dass sich die Container eine einzige IP-Adresse teilen, die den Zugriff auf den Pod ermöglicht, und dass die Container miteinander kommunizieren können, als würden sie auf ihrem eigenen lokalen Host laufen.

Die Pod-Spezifikationen (PodSpec) definieren den gewünschten Status des Pods. Sie können einen einzelnen Pod oder mehrere Pods bereitstellen, indem Sie Workload-Ressourcen zur Verwaltung von Pod-Vorlagen verwenden. Zu den Workload-Ressourcen gehören Bereitstellungen (zur Verwaltung mehrerer Pod-Repliken), StatefulSets(zur Bereitstellung von Pods, die einzigartig sein müssen, z. B. Datenbank-Pods) und DaemonSets(bei denen ein Pod kontinuierlich auf jedem Knoten ausgeführt werden muss). Dazu später mehr.

Während ein Pod die kleinste Einheit ist, die Sie einsetzen, ist ein Container die kleinste Einheit, die Sie erstellen und verwalten.

Container bauen

Der Pod ist eigentlich nur eine Struktur rund um einen oder mehrere Container, wobei jeder Container selbst das Dateisystem, die ausführbaren Dateien, Konfigurationsdateien, Bibliotheken und andere Komponenten enthält, um die Anwendung tatsächlich auszuführen. Da ein Unternehmen namens Docker Inc. zuerst Container populär gemacht hat, bezeichnen manche Leute Container als Container. Docker Seitdem hat die Open Container Initiative jedoch Container-Laufzeiten, Images

und Distributionsmethoden für die Branche definiert. Hinzu kommt die Tatsache, dass Container auf der Grundlage vieler vorhandener Linux-Funktionen erstellt wurden. Andere bezeichnen Container oft als OCI-Container, Linux-Container oder einfach Container.

Wenn Sie einen Container erstellen, beginnen Sie in der Regel mit einer Docker Datei (die wörtlich so genannt wird). In diesem Dockerfile identifizieren Sie:

- Ein Basis-Image — Ein Basis-Container-Image ist ein Container, der in der Regel entweder aus einer Minimalversion des Dateisystems eines Betriebssystems (wie Red Hat Enterprise Linux oder Ubuntu) oder aus einem Minimalsystem erstellt wird, das erweitert wurde, um Software für die Ausführung bestimmter Arten von Anwendungen (wie nodejs - oder Python-Apps) bereitzustellen.
- Anwendungssoftware — Sie können Ihre Anwendungssoftware auf die gleiche Weise zu Ihrem Container hinzufügen, wie Sie sie einem Linux-System hinzufügen würden. In Ihrem Dockerfile können Sie beispielsweise eine Java-Anwendung ausführen `npm` und `yarn` installieren oder `yum` `dnf` RPM-Pakete installieren. Mit anderen Worten, mit einem RUN-Befehl in einem Dockerfile können Sie jeden Befehl ausführen, der im Dateisystem Ihres Basis-Images verfügbar ist, um Software zu installieren oder Software innerhalb des resultierenden Container-Images zu konfigurieren.
- Anweisungen — Die Dockerfile-Referenz beschreibt die Anweisungen, die Sie einer Dockerfile hinzufügen können, wenn Sie sie konfigurieren. Dazu gehören Anweisungen, die verwendet werden, um den Inhalt des Containers selbst (`ADD`oder `COPY` Dateien aus dem lokalen System) zu erstellen, Befehle zu identifizieren, die ausgeführt werden sollen, wenn der Container ausgeführt wird (`CMD`oder`ENTRYPOINT`), und den Container mit dem System zu verbinden, auf dem er ausgeführt werden soll (indem der Container, unter dem er ausgeführt werden soll, ein Lokal, zu dem gemountet werden soll, oder die Ports, `USER` zu denen er ausgeführt werden `VOLUME` soll, oder die Ports identifiziert werden). `EXPOSE`

Während der `docker` Befehl und der Dienst traditionell zum Erstellen von Containern (`docker build`) verwendet wurden, gehören podman und nerdctl zu anderen Tools, die zum Erstellen von Container-Images verfügbar sind. Weitere Informationen zum Erstellen von Containern finden Sie unter Bessere Container-Images erstellen oder Build with Docker.

Speichern von Containern

Sobald Sie Ihr Container-Image erstellt haben, können Sie es in einer Container-Distributionsregistrierung auf Ihrer Workstation oder in einer öffentlichen Container-Registry speichern. Wenn Sie eine private Container-Registry auf Ihrer Workstation betreiben, können Sie Container-Images lokal speichern, sodass sie Ihnen jederzeit zur Verfügung stehen.

Um Container-Images öffentlicher zu speichern, können Sie sie in eine öffentliche Container-Registry übertragen. Öffentliche Container-Register bieten einen zentralen Ort für die Speicherung und Verteilung von Container-Images. Beispiele für öffentliche Container-Registries sind die Amazon Elastic Container Registry, Red Hat Quay Registry und DockerHub Registry.

Wenn Sie containerisierte Workloads auf Amazon Elastic Kubernetes Service (Amazon EKS) ausführen, empfehlen wir, Kopien von Docker offiziellen Images abzurufen, die in Amazon Elastic Container Registry gespeichert sind. AWS Amazon ECR speichert diese Bilder seit 2021. Sie können in der Amazon ECR Public Gallery nach beliebten Container-Bildern suchen, und speziell nach Docker Hub-Bildern können Sie in der Amazon ECR Docker Gallery suchen.

Container ausführen

Da Container in einem Standardformat erstellt werden, kann ein Container auf jedem Computer ausgeführt werden, auf dem eine Container-Laufzeit ausgeführt werden kann (z. B.Docker) und dessen Inhalt der Architektur des lokalen Computers entspricht (z. B. `x86_64` oder`arm`). Um einen Container zu testen oder ihn einfach auf Ihrem lokalen Desktop auszuführen, können Sie `podman run` Befehle `docker run` oder verwenden, um einen Container auf dem Localhost zu starten. Für Kubernetes jeden Worker-Knoten ist jedoch eine Container-Runtime bereitgestellt, und es liegt an Ihnen, Kubernetes anzufordern, dass ein Knoten einen Container ausführt.

Sobald einem Container die Ausführung auf einem Knoten zugewiesen wurde, prüft der Knoten, ob die angeforderte Version des Container-Images bereits auf dem Knoten vorhanden ist. Ist dies nicht der Fall, Kubernetes wird die Container-Laufzeit angewiesen, diesen Container aus der entsprechenden Container-Registry abzurufen und diesen Container dann lokal auszuführen. Denken Sie daran, dass sich ein Container-Image auf das Softwarepaket bezieht, das zwischen Ihrem Laptop, der Container-Registry und den Kubernetes Knoten hin und her bewegt wird. Ein Container bezieht sich auf eine laufende Instanz dieses Images.

Pods

Sobald Ihre Container bereit sind, umfasst die Arbeit mit Pods das Konfigurieren, Bereitstellen und Zugänglichmachen der Pods.

Konfiguration von Pods

Wenn Sie einen Pod definieren, weisen Sie ihm eine Reihe von Attributen zu. Diese Attribute müssen mindestens den Pod-Namen und das Container-Image enthalten, damit sie ausgeführt werden können. Es gibt jedoch noch viele andere Dinge, die Sie mit Ihren Pod-Definitionen konfigurieren

möchten (Einzelheiten darüber, was in einen Pod aufgenommen werden kann, finden Sie auf der PodSpecSeite). Dazu zählen:

- Speicher — Wenn ein laufender Container gestoppt und gelöscht wird, verschwindet der Datenspeicher in diesem Container, sofern Sie keinen permanenten Speicher einrichten. Kubernetesunterstützt viele verschiedene Speichertypen und abstrahiert sie unter dem Dach von Volumes. Zu den Speichertypen gehören CephFS, NFS, iSCSI und andere. Sie können sogar ein lokales Blockgerät vom lokalen Computer aus verwenden. Wenn einer dieser Speichertypen in Ihrem Cluster verfügbar ist, können Sie das Speichervolume an einem ausgewählten Einhängepunkt im Dateisystem Ihres Containers mounten. Ein persistentes Volume ist ein Volume, das auch nach dem Löschen des Pods weiterhin existiert, während ein flüchtiges Volume gelöscht wird, wenn der Pod gelöscht wird. Falls Ihr Clusteradministrator etwas anderes StorageClassesfür Ihren Cluster erstellt hat, haben Sie möglicherweise die Möglichkeit, die Attribute des von Ihnen verwendeten Speichers auszuwählen, z. B. ob das Volume gelöscht oder nach der Verwendung zurückgewonnen wird, ob es erweitert wird, wenn mehr Speicherplatz benötigt wird, und sogar, ob es bestimmte Leistungsanforderungen erfüllt.
- Secrets — Indem Sie Secrets in Pod-Spezifikationen für Container verfügbar machen, können Sie die Berechtigungen bereitstellen, die diese Container für den Zugriff auf Dateisysteme, Datenbanken oder andere geschützte Ressourcen benötigen. Schlüssel, Passwörter und Token gehören zu den Elementen, die als Geheimnisse gespeichert werden können. Durch die Verwendung von Geheimnissen müssen Sie diese Informationen nicht in Container-Images speichern, sondern müssen die Geheimnisse nur laufenden Containern zur Verfügung stellen. Ähnlich wie Secrets sind ConfigMaps. A enthält in der ConfigMap Regel weniger wichtige Informationen, wie z. B. Schlüssel-Wert-Paare für die Konfiguration eines Dienstes.
- Container-Ressourcen — Objekte für die weitere Konfiguration von Containern können die Form einer Ressourcenkonfiguration annehmen. Für jeden Container können Sie die Menge an Speicher und CPU anfordern, die er verwenden kann, sowie Beschränkungen für die Gesamtmenge dieser Ressourcen festlegen, die der Container verwenden kann. Beispiele finden Sie unter Ressourcenverwaltung für Pods und Container.
- Unterbrechungen — Pods können unfreiwillig (ein Knoten fällt aus) oder freiwillig (ein Upgrade ist gewünscht) unterbrochen werden. Durch die Konfiguration eines Budgets für Pod-Unterbrechungen können Sie eine gewisse Kontrolle darüber ausüben, wie verfügbar Ihre Anwendung bei Störungen bleibt. Beispiele finden Sie unter Festlegung eines Unterbrechungsbudgets für Ihre Anwendung.
- Namespaces — Kubernetes bietet verschiedene Möglichkeiten, Kubernetes Komponenten und Workloads voneinander zu isolieren. Das Ausführen aller Pods für eine bestimmte Anwendung im selben Namespace ist eine gängige Methode, um diese Pods gemeinsam zu sichern und zu

verwalten. Sie können Ihre eigenen Namespaces erstellen, die Sie verwenden möchten, oder sich dafür entscheiden, keinen Namespace anzugeben (was Kubernetes dazu führt, dass der Namespace verwendet wird). `default` KubernetesKomponenten der Steuerungsebene werden normalerweise im Kube-System-Namespace ausgeführt.

Die gerade beschriebene Konfiguration wird normalerweise in einer YAML-Datei zusammengefasst, die auf den Cluster angewendet wird. Kubernetes Für persönliche Kubernetes Cluster können Sie diese YAML-Dateien einfach auf Ihrem lokalen System speichern. Bei kritischeren Clustern und Workloads GitOpsist dies jedoch eine beliebte Methode, um die Speicherung und Aktualisierung von Workload- und Kubernetes Infrastrukturressourcen zu automatisieren.

Die Objekte, die zum Sammeln und Bereitstellen von Pod-Informationen verwendet werden, werden durch eine der folgenden Bereitstellungsmethoden definiert.

Bereitstellen von Pods

Die Methode, die Sie für die Bereitstellung von Pods wählen würden, hängt von der Art der Anwendung ab, die Sie mit diesen Pods ausführen möchten. Hier sind einige Ihrer Optionen:

- Zustandslose Anwendungen — Eine statuslose Anwendung speichert die Sitzungsdaten eines Clients nicht, sodass in einer anderen Sitzung nicht auf das zurückgegriffen werden muss, was mit einer vorherigen Sitzung passiert ist. Dadurch ist es einfacher, Pods einfach durch neue zu ersetzen, wenn sie defekt sind, oder sie zu verschieben, ohne den Status zu speichern. Wenn Sie eine statuslose Anwendung (z. B. einen Webserver) ausführen, können Sie Pods und mithilfe eines Deployments bereitstellen. ReplicaSets A ReplicaSet definiert, wie viele Instanzen eines Pods gleichzeitig ausgeführt werden sollen. Obwohl Sie einen ReplicaSet direkt ausführen können, ist es üblich, Replikate direkt in einem Deployment auszuführen, um zu definieren, wie viele Replikate eines Pods gleichzeitig ausgeführt werden sollen.
- Stateful-Anwendungen — Bei einer statusbehafteten Anwendung sind die Identität des Pods und die Reihenfolge, in der Pods gestartet werden, wichtig. Diese Anwendungen benötigen dauerhaften Speicher, der stabil ist und konsistent bereitgestellt und skaliert werden muss. Um eine statusbehaftete Anwendung bereitzustellenKubernetes, können Sie verwenden. StatefulSets Ein Beispiel für eine Anwendung, die normalerweise als ausgeführt wird, StatefulSet ist eine Datenbank. In einer StatefulSet könnten Sie Replikate, den Pod und seine Container, zu mountende Speichervolumes und Speicherorte im Container, an denen Daten gespeichert werden, definieren. Ein Beispiel für eine Datenbank, die als bereitgestellt wird, finden Sie unter Ausführen einer replizierten Stateful-Anwendung. ReplicaSet

- Anwendungen pro Knoten — Manchmal möchten Sie eine Anwendung auf jedem Knoten in Ihrem Cluster ausführen. Kubernetes In Ihrem Rechenzentrum kann es beispielsweise erforderlich sein, dass auf jedem Computer eine Überwachungsanwendung oder ein bestimmter RAS-Dienst ausgeführt wird. Denn Sie können a verwendenKubernetes, DaemonSetum sicherzustellen, dass die ausgewählte Anwendung auf jedem Knoten in Ihrem Cluster ausgeführt wird.
- Anwendungen werden bis zum Abschluss ausgeführt — Es gibt einige Anwendungen, die Sie ausführen möchten, um eine bestimmte Aufgabe abzuschließen. Dazu könnte eine Software gehören, die monatliche Statusberichte erstellt oder alte Daten bereinigt. Ein Job-Objekt kann verwendet werden, um eine Anwendung so einzurichten, dass sie gestartet und ausgeführt wird und dann beendet wird, wenn die Aufgabe erledigt ist. Mit einem CronJobObjekt können Sie eine Anwendung so einrichten, dass sie zu einer bestimmten Stunde, Minute, an einem bestimmten Tag des Monats, Monats oder Wochentags ausgeführt wird. Dabei wird eine Struktur verwendet, die durch das Linux-Crontab-Format definiert ist.

Anwendungen vom Netzwerk aus zugänglich machen

Da Anwendungen oft als eine Reihe von Microservices bereitgestellt wurden, die an verschiedene Orte verlagert wurden, Kubernetes musste für diese Microservices eine Möglichkeit geschaffen werden, einander zu finden. Damit andere auf eine Anwendung außerhalb des Kubernetes Clusters zugreifen konnten, Kubernetes musste außerdem eine Möglichkeit gefunden werden, diese Anwendung über externe Adressen und Ports zugänglich zu machen. Diese netzwerkbezogenen Funktionen werden mit Service- bzw. Ingress-Objekten ausgeführt:

- Dienste — Da sich ein Pod zu verschiedenen Knoten und Adressen bewegen kann, könnte es für einen anderen Pod, der mit dem ersten Pod kommunizieren musste, schwierig sein, seinen Standort zu finden. Um dieses Problem zu lösen, Kubernetes können Sie eine Anwendung als Dienst darstellen. Mit einem Dienst können Sie einen Pod oder eine Gruppe von Pods mit einem bestimmten Namen identifizieren und dann angeben, welcher Port den Dienst dieser Anwendung vom Pod aus verfügbar macht und welche Ports eine andere Anwendung verwenden könnte, um diesen Dienst zu kontaktieren. Ein anderer Pod innerhalb eines Clusters kann einen Service einfach anhand des Namens anfordern und leitet Kubernetes diese Anfrage an den richtigen Port für eine Pod-Instanz weiter, auf der dieser Dienst ausgeführt wird.
- Ingress — Durch Ingress können Anwendungen, die durch Kubernetes Dienste repräsentiert werden, für Clients verfügbar gemacht werden, die sich außerhalb des Clusters befinden. Zu den grundlegenden Funktionen von Ingress gehören ein Load Balancer (von Ingress verwaltet), der Ingress-Controller und Regeln für die Weiterleitung von Anfragen vom Controller an den Service. Es gibt mehrere Ingress-Controller, aus denen Sie wählen können. Kubernetes

Nächste Schritte

Wenn Sie die grundlegenden Kubernetes Konzepte und ihren Zusammenhang mit Amazon EKS verstehen, können Sie sich sowohl in der Amazon EKS-Dokumentation als auch in der Dokumentation zurechtfinden, um die Informationen zu finden, die Sie für die Verwaltung von Amazon EKS-Clustern und die Bereitstellung von Workloads in diesen Clustern benötigen. Kubernetes Um mit der Nutzung von Amazon EKS zu beginnen, wählen Sie aus den folgenden Optionen:

- Erstellen Sie einen einfachen Cluster
- Erstellen Sie einen komplexeren Cluster
- Stellen Sie eine Beispielanwendung bereit
- Erkunden Sie Möglichkeiten zur Verwaltung Ihres Clusters

Optionen für die Bereitstellung

Sie können Amazon EKS mit einer oder allen der folgenden Optionen bereitstellen:

Amazon EKS in der Cloud

Sie können Kubernetes in der AWS-Cloud ausführen, ohne dass Sie eine eigene Kubernetes-Steuerebene oder eigene Knoten installieren, betreiben und warten müssen. Diese Option wird in diesem Handbuch behandelt.

Amazon EKS in Outposts

AWS Outposts ermöglicht native AWS-Services, Infrastrukturen und Betriebsmodelle in Ihren On-Premises-Einrichtungen. Mit Amazon EKS in Outposts können Sie wählen, ob Sie erweiterte oder lokale Cluster ausführen möchten. Bei erweiterten Clustern wird die Kubernetes-Steuerebene in einer AWS-Region ausgeführt und die Knoten werden in Outposts ausgeführt. Mit lokalen Clustern läuft der gesamte Kubernetes-Cluster lokal auf Outposts, einschließlich der Kubernetes-Steuerebenen und Knoten. Weitere Informationen finden Sie unter auf Amazon EKSAWS Outposts.

Amazon EKS Anywhere

Amazon EKS Anywhere ist eine Bereitstellungsoption für Amazon EKS, mit der Sie Kubernetes-Cluster einfach On-Premises erstellen und betreiben können. Sowohl Amazon EKS als auch Amazon EKS Anywhere basieren auf Amazon EKS Distro. Weitere Informationen zu Amazon EKS Anywhere und den Unterschieden zu Amazon EKS finden Sie unter Übersicht und Vergleich

von Amazon EKS Anywhere mit Amazon EKS in der Dokumentation zu Amazon EKS Anywhere. Antworten auf einige häufig gestellte Fragen finden Sie unter Häufig gestellte Fragen zu Amazon EKS Anywhere.

Amazon EKS Distro

Amazon EKS Distro ist eine Distribution der gleichen Open-Source-Kubernetes-Software und -Abhängigkeiten, die von Amazon EKS in der Cloud bereitgestellt werden. Amazon EKS Distro folgt dem gleichen Veröffentlichungszyklus von Kubernetes-Versionen wie Amazon EKS und wird als Open-Source-Projekt bereitgestellt. Weitere Informationen finden Sie unter Amazon EKS Distro. Sie können auch den Quellcode für Amazon EKS Distro auf GitHub aufrufen und herunterladen.

Berücksichtigen Sie bei der Auswahl der Bereitstellungsoptionen für Ihren Kubernetes-Cluster Folgendes:

Funktion	Amazon EKS	Amazon EKS in Outposts	Amazon EKS Anywhere	Amazon EKS Distro
Hardware (Hardware)	Von AWS bereitgestellt	Von AWS bereitgestellt	Von Ihnen bereitgestellt	Von Ihnen bereitgestellt
Bereitstellungsort	AWS Cloud	Ihr Rechenzentrum	Ihr Rechenzentrum	Ihr Rechenzentrum
Kubernetes Position der Steuerebene	AWS Cloud	AWS-Cloud oder Ihr Rechenzentrum	Ihr Rechenzentrum	Ihr Rechenzentrum
Kubernetes-Position der Datenebene	AWSWolke	Ihr Rechenzentrum	Ihr Rechenzentrum	Ihr Rechenzentrum
Support	AWS Support	AWS Support	AWS Support	Support der OSS-Community

Einrichten von Amazon EKS

Für AWS-Ressourcen gelten üblicherweise Zugriffsbeschränkungen, die den Zugriff auf die AWS-Entität einschränken, von der sie erstellt wurden. Daher ist es wichtig, in der AWS Command Line Interface von Anfang an eine ordnungsgemäße Benutzerkonfiguration festzulegen. Darüber hinaus müssen Sie Ihren lokalen Computer mit grundlegenden Tools für eine effiziente Befehlszeilenverwaltung Ihres Amazon-EKS-Clusters versehen. Dieses Thema hilft Ihnen dabei, Ihren Cluster für die Befehlszeilenverwaltung vorzubereiten.

Schritt 1: Einrichten der AWS CLI

Die AWS CLI ist ein Befehlszeilen-Tool für die Arbeit mit AWS-Services (einschließlich Amazon EKS). Es wird auch verwendet, um IAM-Benutzer oder -Rollen für den Zugriff auf den Amazon-EKS-Cluster sowie auf andere AWS-Ressourcen von Ihrem lokalen Computer zu authentifizieren. Um Ressourcen über die Befehlszeile in AWS bereitstellen zu können, benötigen Sie eine AWS-Zugriffsschlüssel-ID und einen geheimen Schlüssel für die Verwendung in der Befehlszeile. Anschließend müssen diese Anmeldeinformationen in der AWS CLI konfiguriert werden. Falls Sie die AWS CLI noch nicht installiert haben, finden Sie entsprechende Installationsanweisungen im Benutzerhandbuch zu AWS Command Line Interface unter Installieren oder Aktualisieren der aktuellen Version der AWS CLI.

So erstellen Sie einen Zugriffsschlüssel

1. Melden Sie sich beim AWS Management Console an.
2. Wählen Sie rechts oben Ihren AWS-Benutzernamen aus, um das Navigationsmenü zu öffnen. Wählen Sie zum Beispiel aus **`webadmin`**. Wählen Sie anschließend Sicherheitsanmeldeinformationen aus.
3. Wählen Sie unter Zugriffsschlüssel die Option Zugriffsschlüssel erstellen aus.
4. Wählen Sie Befehlszeilenschnittstelle (CLI) und anschließend Weiter aus.
5. Wählen Sie Zugriffsschlüssel erstellen aus.
6. Wählen Sie CSV-Datei herunterladen aus.

Konfigurieren der AWS CLI

Führen Sie nach dem Installieren der AWS CLI die folgenden Schritte aus, um sie zu konfigurieren. Weitere Informationen finden Sie unter Konfigurieren der AWS CLI im AWS Command Line Interface-Benutzerhandbuch.

1. Geben Sie in einem Terminal-Fenster den folgenden Befehl ein:

   ```
   aws configure
   ```

   Optional können Sie ein benanntes Profil konfigurieren, z. B. **`--profile cluster-admin`**. Wenn Sie ein benanntes Profil in der AWS CLI konfigurieren, muss in nachfolgenden Befehlen immer dieses Flag übergeben werden.

2. Geben Sie Ihre AWS-Anmeldeinformationen ein. Beispielsweise:

   ```
   AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
   AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   Default region name [None]: region-code
   Default output format [None]: json
   ```

So rufen Sie ein Sicherheitstoken ab

Führen Sie bei Bedarf den folgenden Befehl aus, um ein neues Sicherheitstoken für die AWS CLI abzurufen. Weitere Informationen finden Sie unter `get-session-token` in der Referenz zum AWS CLI-Befehl.

Das Token ist standardmäßig 15 Minuten lang gültig. Das standardmäßige Sitzungs-Timeout kann durch Übergeben des Flags **`--duration-seconds`** geändert werden. Beispielsweise:

```
aws sts get-session-token --duration-seconds 3600
```

Dieser Befehl gibt die temporären Sicherheitsanmeldeinformationen für eine AWS CLI-Sitzung zurück. Die ausgegebene Antwort sollte wie folgt aussehen:

```
{
    "Credentials": {
        "AccessKeyId": "ASIA5FTRU3LOEXAMPLE",
```

```
        "SecretAccessKey": "JnKgvwfqUD9mNsPoi9IbxAYEXAMPLE",
        "SessionToken": "VERYLONGSESSIONTOKENSTRING",
        "Expiration": "2023-02-17T03:14:24+00:00"
    }
}
```

So überprüfen Sie die Benutzeridentität

Führen Sie bei Bedarf den folgenden Befehl aus, um die AWS-Anmeldeinformationen für Ihre IAM-Benutzeridentität (z. B. *ClusterAdmin*) für die Terminalsitzung zu überprüfen.

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

Dieser Befehl gibt den Amazon-Ressourcennamen (ARN) der IAM-Entität zurück, die für die AWS CLI konfiguriert ist. Die ausgegebene Antwort sollte beispielsweise wie folgt aussehen:

```
{
    "UserId": "AKIAIOSFODNN7EXAMPLE",
    "Account": "01234567890",
    "Arn": "arn:aws:iam::01234567890:user/ClusterAdmin"
}
```

Schritt 2: Installieren von Kubernetes-Tools

Für die Kommunikation mit einem Kubernetes-Cluster benötigen Sie ein Tool, über das Sie mit der Kubernetes-API interagieren können. Und Sie benötigen noch ein paar weitere Tools – beispielsweise eins für die Verwaltung von Kubernetes-Umgebungen auf Ihrem lokalen Computer.

So erstellen Sie AWS-Ressourcen

- Amazon-EKS-Clusterressourcen: Falls Sie noch nicht mit AWS vertraut sind, empfiehlt es sich, `eksctl` zu installieren. `eksctl` ist ein Infrastructure as Code (IaC)-Hilfsprogramm, das AWS CloudFormation verwendet, um Ihren Amazon-EKS-Cluster auf einfache Weise zu erstellen. Außerdem erstellt es zusätzliche Kubernetes-Ressourcen wie etwa Servicekonten. Eine Installationsanleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.
- AWS-Ressourcen: Wenn Sie es gewohnt sind, die Bereitstellung Ihrer AWS-Infrastruktur zu automatisieren, empfiehlt es sich, Terraform zu installieren. Terraform ist ein von HashiCorp

entwickeltes Open-Source-IaC-Tool. Damit können Sie die Infrastruktur mithilfe einer allgemeinen Konfigurationssprache wie HashiCorp Configuration Language (HCL) oder JSON definieren und bereitstellen. Eine Installationsanleitung für Terraform finden Sie in der Dokumentation zu Terraform unter Install Terraform.

So installieren Sie **kubectl**

`kubectl` ist ein Open-Source-Befehlszeilentool für die Kommunikation mit dem Kubernetes-API-Server in Ihrem Amazon-EKS-Cluster. Sollte es noch nicht auf Ihrem lokalen Computer installiert sein, wählen Sie eine der folgenden Optionen.

- AWS-Versionen: Informationen zum Installieren einer von Amazon EKS unterstützten `kubectl`-Version finden Sie hier: Installieren oder Aktualisieren von `kubectl`.
- Community-Versionen: Informationen zum Installieren der neuesten Community-Version von `kubectl` finden Sie in der Dokumentation zu Kubernetes auf der Seite Werkzeuge installieren.

So richten Sie eine Entwicklungsumgebung ein

- Lokales Bereitstellungstool: Wenn Sie noch nicht mit Kubernetes vertraut sind, empfiehlt es sich gegebenenfalls, ein lokales Bereitstellungstool wie `minikube` oder `kind` zu installieren. Mit diesen Tools können Sie einen Amazon-EKS-Cluster auf Ihrem lokalen Computer verwalten.
- Paketmanager: Helm ist ein beliebter Paketmanager für Kubernetes, der die Installation und Verwaltung komplexer Pakete vereinfacht. Mit Helm können Pakete wie der AWS-Load-Balancer-Controller in Ihrem Amazon-EKS-Cluster einfacher installiert und verwaltet werden.

Nächste Schritte

- Erste Schritte mit Amazon EKS

Installieren oder Aktualisieren von **kubectl**

`Kubectl` ist ein Befehlszeilen-Tool, mit dem Sie mit dem Kubernetes-API-Server kommunizieren. Die `kubectl`-Binärdatei ist in vielen Betriebssystem-Paketmanagern verfügbar. Die Verwendung eines Paketmanagers für Ihre Installation ist oft einfacher als ein manueller Download- und Installationsprozess.

In diesem Thema finden Sie Hinweise zum Herunterladen und Installieren der `kubectl`-Binärdatei auf Ihrem Gerät. Die Binärdatei ist identisch mit den Upstream-Gemeinschaftsversionen. Die Binärdatei gilt nicht nur für Amazon EKS oder AWS.

> **Note**
>
> Sie müssen eine `kubectl`-Version verwenden, die nur in der Minor-Version von Ihrer Amazon–EKS-Cluster-Steuerebene abweicht. Beispielsweise sollte ein `1.29`-`kubectl`-Client mit Kubernetes-, `1.28`-, `1.29`- und `1.30`-Clustern funktionieren.

Installieren oder aktualisieren Sie **`kubectl`** wie folgt

1. Stellen Sie fest, ob Sie `kubectl` bereits auf Ihrem Gerät installiert haben.

   ```
   kubectl version --client
   ```

   Wenn `kubectl` im Pfad Ihres Geräts installiert ist, enthält die Beispielausgabe ähnliche Informationen wie die folgende. Wenn Sie die Version, die Sie derzeit installiert haben, mit einer neueren Version aktualisieren möchten, führen Sie den nächsten Schritt aus und stellen Sie sicher, dass Sie die neue Version an demselben Speicherort installieren, an dem sich Ihre aktuelle Version befindet.

   ```
   Client Version: v1.30.X-eks-1234567
   ```

   Wenn Sie keine Ausgaben erhalten, haben Sie entweder keine `kubectl` installiert, oder es ist nicht an einem Ort installiert, der sich im Pfad Ihres Geräts befindet.
2. Installieren oder Aktualisieren von `kubectl` auf `macOS`-, Linux- und Windows-Betriebssysteme.

   macOS

   Installieren oder aktualisieren Sie **`kubectl`** auf **`macOS`** wie folgt

   1. Laden Sie die Binärdatei für die Kubernetes-Version Ihres Clusters von Amazon S3 herunter.
      - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/darwin/amd64/kubectl
```

- Kubernetes `1.29`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes `1.28`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes `1.27`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes `1.26`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/
bin/darwin/amd64/kubectl
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/
bin/darwin/amd64/kubectl
```

2. (Optional) Überprüfen Sie die heruntergeladene Binärdatei mit der `SHA-256`-Prüfsumme für Ihre Binärdatei.

   a. Laden Sie die `SHA-256`-Prüfsumme für die Kubernetes-Version Ihres Clusters herunter.

   - Kubernetes `1.30`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/
     bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes `1.29`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/
     bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes `1.28`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/
     bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes `1.27`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/
     bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes `1.26`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/
     bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/darwin/amd64/kubectl.sha256
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/darwin/amd64/kubectl.sha256
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/darwin/amd64/kubectl.sha256
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/darwin/amd64/kubectl.sha256
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/darwin/amd64/kubectl.sha256
```

b. Überprüfen Sie die `SHA-256`-Prüfsumme für Ihre heruntergeladene Binärdatei.

```
openssl sha1 -sha256 kubectl
```

c. Stellen Sie sicher, dass die generierte Prüfsumme in der Ausgabe mit der Prüfsumme in der heruntergeladenen `kubectl.sha256`-Datei übereinstimmt.

3. Wenden Sie Ausführungsberechtigungen auf die Binärdatei an.

```
chmod +x ./kubectl
```

4. Kopieren Sie die Binärdatei in ein Verzeichnis in Ihrer `PATH`-Variable. Wenn Sie bereits eine Version von `kubectl` installiert haben, empfehlen wir, eine `$HOME/bin/kubectl`-Datei zu erstellen und sicherzustellen, dass `$HOME/bin` in der `$PATH`-Variablen zuerst vorkommt.

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:
$PATH
```

5. (Optional) Fügen Sie den Pfad `$HOME/bin` zu Ihrer Shell-Initialisierungsdatei hinzu, um den Pfad bereits beim Öffnen einer Shell zu konfigurieren.

```
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bash_profile
```

Linux (amd64)

Installieren oder aktualisieren Sie **`kubectl`** auf Linux wie folgt (**`amd64`**)

1. Laden Sie die `kubectl`-Binärdatei für die Kubernetes-Version Ihres Clusters von Amazon S3 herunter.
   - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/
linux/amd64/kubectl
```

   - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/
linux/amd64/kubectl
```

   - Kubernetes `1.29`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/
linux/amd64/kubectl
```

   - Kubernetes `1.28`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/
linux/amd64/kubectl
```

   - Kubernetes `1.27`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/
bin/linux/amd64/kubectl
```

- Kubernetes `1.26`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/amd64/kubectl
```

2. (Optional) Überprüfen Sie die heruntergeladene Binärdatei mit der `SHA-256`-Prüfsumme für Ihre Binärdatei.

   a. Laden Sie die `SHA-256`-Prüfsumme für die Kubernetes-Version Ihres Clusters von Amazon S3 mit dem Befehl für Ihre Hardware-Plattform herunter. Der erste Link für jede Version ist für `amd64` und der zweite Link ist für `arm64`.

   - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.29`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.28`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.27`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.26`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/amd64/kubectl.sha256
```

b. Überprüfen Sie die `SHA-256`-Prüfsumme für die heruntergeladene Binärdatei mit einem der folgenden Befehle.

- ```
  sha256sum -c kubectl.sha256
  ```

  Wenn Sie diesen Befehl verwenden, stellen Sie sicher, dass Sie die folgende Ausgabe sehen:

  ```
  kubectl: OK
  ```

- ```
  openssl sha1 -sha256 kubectl
  ```

  Wenn Sie diesen Befehl verwenden, stellen Sie sicher, dass die generierte Prüfsumme in der Ausgabe mit der Prüfsumme in der heruntergeladenen `kubectl.sha256`-Datei übereinstimmt.

3. Wenden Sie Ausführungsberechtigungen auf die Binärdatei an.

   ```
   chmod +x ./kubectl
   ```

4. Kopieren Sie die Binärdatei in ein Verzeichnis in Ihrer `PATH`-Variable. Wenn Sie bereits eine Version von `kubectl` installiert haben, empfehlen wir, eine `$HOME/bin/kubectl`-Datei zu erstellen und sicherzustellen, dass `$HOME/bin` in der `$PATH`-Variablen zuerst vorkommt.

   ```
   mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
   ```

5. (Optional) Fügen Sie den Pfad `$HOME/bin` zu Ihrer Shell-Initialisierungsdatei hinzu, um den Pfad bereits beim Öffnen einer Shell zu konfigurieren.

> **Note**
>
> Bei diesem Schritt wird davon ausgegangen, dass Sie das Bash-Shell verwenden. Wenn Sie eine andere Shell nutzen, ändern Sie den Befehl zur Angabe der spezifischen Shell-Initialisierungsdatei.

```
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
```

Linux (arm64)

Installieren oder aktualisieren Sie **`kubectl`** auf Linux wie folgt (**`arm64`**)

1. Laden Sie die `kubectl`-Binärdatei für die Kubernetes-Version Ihres Clusters von Amazon S3 herunter.
   - Kubernetes `1.30`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.30`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.29`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.28`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.27`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.26`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/arm64/kubectl
```

2. (Optional) Überprüfen Sie die heruntergeladene Binärdatei mit der `SHA-256`-Prüfsumme für Ihre Binärdatei.

   a. Laden Sie die `SHA-256`-Prüfsumme für die Kubernetes-Version Ihres Clusters von Amazon S3 mit dem Befehl für Ihre Hardware-Plattform herunter. Der erste Link für jede Version ist für `amd64` und der zweite Link ist für `arm64`.

   - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.29`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.28`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.27`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.26`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/linux/arm64/kubectl.sha256
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/arm64/kubectl.sha256
```

b. Überprüfen Sie die `SHA-256`-Prüfsumme für die heruntergeladene Binärdatei mit einem der folgenden Befehle.

- ```
  sha256sum -c kubectl.sha256
  ```

  Wenn Sie diesen Befehl verwenden, stellen Sie sicher, dass Sie die folgende Ausgabe sehen:

  ```
  kubectl: OK
  ```

- ```
  openssl sha1 -sha256 kubectl
  ```

  Wenn Sie diesen Befehl verwenden, stellen Sie sicher, dass die generierte Prüfsumme in der Ausgabe mit der Prüfsumme in der heruntergeladenen `kubectl.sha256`-Datei übereinstimmt.

3. Wenden Sie Ausführungsberechtigungen auf die Binärdatei an.

```
chmod +x ./kubectl
```

4. Kopieren Sie die Binärdatei in ein Verzeichnis in Ihrer `PATH`-Variable. Wenn Sie bereits eine Version von `kubectl` installiert haben, empfehlen wir, eine `$HOME/bin/kubectl`-Datei zu erstellen und sicherzustellen, dass `$HOME/bin` in der `$PATH`-Variablen zuerst vorkommt.

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
```

5. (Optional) Fügen Sie den Pfad `$HOME/bin` zu Ihrer Shell-Initialisierungsdatei hinzu, um den Pfad bereits beim Öffnen einer Shell zu konfigurieren.

> **Note**
>
> Bei diesem Schritt wird davon ausgegangen, dass Sie das Bash-Shell verwenden. Wenn Sie eine andere Shell nutzen, ändern Sie den Befehl zur Angabe der spezifischen Shell-Initialisierungsdatei.

```
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
```

Windows

Installieren oder aktualisieren Sie **`kubectl`** auf Windows wie folgt

1. Öffnen Sie ein PowerShell-Terminalfenster.
2. Laden Sie die `kubectl`-Binärdatei für die Kubernetes-Version Ihres Clusters von Amazon S3 herunter.
   - Kubernetes `1.30`

     ```
     curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/windows/amd64/kubectl.exe
     ```

   - Kubernetes `1.29`

     ```
     curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/windows/amd64/kubectl.exe
     ```

   - Kubernetes `1.28`

     ```
     curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/windows/amd64/kubectl.exe
     ```

   - Kubernetes `1.27`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.27.12/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.26`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.26.15/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.25`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.25.16/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.24`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.24.17/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.23`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.23.17/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.22`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.22.17/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.21`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.21.14/2024-04-19/bin/windows/amd64/kubectl.exe
```

3. (Optional) Überprüfen Sie die heruntergeladene Binärdatei mit der `SHA-256`-Prüfsumme für Ihre Binärdatei.

   a. Laden Sie die `SHA-256`-Prüfsumme für die Kubernetes-Version Ihres Clusters für Windows herunter.

   - Kubernetes `1.30`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.30.0/2024-05-12/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes 1.29

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.29.3/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes 1.28

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.28.8/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes 1.27

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.27.12/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes 1.26

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.26.15/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes 1.25

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.25.16/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes 1.24

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.24.17/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes 1.23

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.23.17/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes 1.22

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.22.17/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes `1.21`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.21.14/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

b. Überprüfen Sie die `SHA-256`-Prüfsumme für Ihre heruntergeladene Binärdatei.

```
Get-FileHash kubectl.exe
```

c. Stellen Sie sicher, dass die generierte Prüfsumme in der Ausgabe mit der Prüfsumme in der heruntergeladenen `kubectl.sha256`-Datei übereinstimmt. Die PowerShell Ausgabe sollte eine entsprechende Zeichenfolge in Großbuchstaben sein.

4. Kopieren Sie die Binärdatei in ein Verzeichnis in Ihrer `PATH`-Variable. Wenn Sie in Ihrem `PATH` ein Verzeichnis vorhanden ist, das Sie für Befehlszeilen-Dienstprogramme verwenden, kopieren Sie die Binärdatei in dieses Verzeichnis. Führen Sie andernfalls die folgenden Schritte aus.

   a. Erstellen Sie ein neues Verzeichnis für Ihre Befehlszeilen-Binärdateien, z. B. `C:\bin`.

   b. Kopieren Sie die `kubectl.exe`-Binärdatei in Ihr neues Verzeichnis.

   c. Bearbeiten Sie Ihre Benutzer- oder System-`PATH`-Umgebungsvariable, um das neue Verzeichnis zu Ihrem `PATH` hinzuzufügen.

   d. Schließen Sie Ihr PowerShell-Terminal und öffnen Sie ein neues, um die neue `PATH`-Variablen aufzunehmen.

3. Nach der Installation von `kubectl` können Sie die Version prüfen:

```
kubectl version --client
```

Bei der Erstinstallation von `kubectl` ist es noch nicht für die Kommunikation mit einem Server konfiguriert. Wir werden diese Konfiguration bei Bedarf in anderen Verfahren behandeln. Wenn Sie jemals die Konfiguration aktualisieren müssen, um mit einem bestimmten Cluster zu kommunizieren, können Sie den folgenden Befehl ausführen. *`region-code`*Ersetzen Sie es durch AWS-Region das, in dem sich Ihr Cluster befindet. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

```
aws eks update-kubeconfig --region region-code --name my-cluster
```

Erste Schritte mit Amazon EKS

Vergewissern Sie sich, dass Sie für die Verwendung von Amazon EKS bereit sind, bevor Sie sich mit den Anleitungen für die ersten Schritte beschäftigen. Weitere Informationen finden Sie unter Einrichten von Amazon EKS.

Es gibt zwei Handbücher „Erste Schritte“ für das Erstellen eines neuen Kubernetes-Clusters mit Knoten in Amazon EKS:

- Erste Schritte mit Amazon EKS – `eksctl` – In diesem Handbuch „Erste Schritte“ finden Sie Hilfe bei der Installation aller erforderlichen Ressourcen für die ersten Schritte mit `eksctl` unter Verwendung von Amazon EKS, ein einfaches Befehlszeilen-Dienstprogramm zum Erstellen und Verwalten von Kubernetes-Clustern auf Amazon EKS. Am Ende des Tutorials haben Sie einen laufenden Amazon-EKS-Cluster, auf dem Sie Anwendungen bereitstellen können. Dies ist die schnellste und einfachste Möglichkeit zum Einstieg in Amazon EKS.
- Erste Schritte mit Amazon EKS — AWS Management Console und AWS CLI— Dieses Handbuch für die ersten Schritte hilft Ihnen dabei, alle erforderlichen Ressourcen für die ersten Schritte mit Amazon EKS mithilfe von AWS Management Console und zu erstellen AWS CLI. Am Ende des Tutorials haben Sie einen laufenden Amazon-EKS-Cluster, auf dem Sie Anwendungen bereitstellen können. In diesem Handbuch erstellen Sie manuell jede Ressource, die für einen Amazon-EKS-Cluster erforderlich ist. Die Verfahren geben Ihnen einen Überblick darüber, wie jede Ressource erstellt wird und wie die Ressourcen miteinander interagieren.

Wir bieten auch die folgenden Referenzen an:

- Eine kuratierte Sammlung praktischer Tutorials finden Sie unter Navigating Amazon EKS on Community.AWS
- Codebeispiele finden Sie unter Codebeispiele für Amazon EKS mit AWS SDKs.

Erste Schritte mit Amazon EKS – **`eksctl`**

Dieses Handbuch hilft Ihnen beim Erstellen aller erforderlichen Ressourcen für die ersten Schritte mit Amazon Elastic Kubernetes Service (Amazon EKS) mithilfe von `eksctl`, einem einfachen Befehlszeilendienstprogramm zum Erstellen und Verwalten von Kubernetes-Clustern auf Amazon EKS. Am Ende dieses Tutorials haben Sie einen laufenden Amazon-EKS-Cluster, auf dem Sie Anwendungen bereitstellen können.

Die Verfahren in diesem Handbuch erstellen automatisch mehrere Ressourcen für Sie, die Sie manuell erstellen müssen, wenn Sie Ihren Cluster mit dem AWS Management Console. Wenn Sie die meisten Ressourcen lieber manuell erstellen möchten, um besser zu verstehen, wie sie miteinander interagieren, verwenden Sie den, AWS Management Console um Ihren Cluster zu erstellen und zu berechnen. Weitere Informationen finden Sie unter Erste Schritte mit Amazon EKS — AWS Management Console und AWS CLI.

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Werkzeuge und Ressourcen installieren und konfigurieren, die Sie zum Erstellen und Verwalten eines Amazon-EKS-Clusters benötigen.

- **`kubectl`** – Ein Befehlszeilentool für die Arbeit mit Kubernetes-Clustern. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- **`eksctl`** – Ein Befehlszeilen-Tool für die Arbeit mit EKS-Clustern, das viele einzelne Aufgaben automatisiert. Weitere Informationen finden Sie in der Dokumentation zu `eksctl` unter Installation.
- Erforderliche IAM-Berechtigungen — Der von Ihnen verwendete IAM-Sicherheitsprinzipal muss über Berechtigungen verfügen, um mit Amazon EKS-IAM-Rollen, serviceverknüpften Rollen AWS CloudFormation, einer VPC und verwandten Ressourcen zu arbeiten. Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Elastic Container Service für Kubernetes und Verwenden von serviceverknüpften Rollen im IAM-Benutzerhandbuch. Sie müssen alle Schritte in diesem Handbuch als derselbe Benutzer ausführen. Führen Sie den folgenden Befehl aus, um den aktuellen Benutzer zu überprüfen:

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

Schritt 1: Erstellen Sie Ihre Amazon-EKS-Cluster und -Knoten

> **Important**
>
> Für einen möglichst einfachen und schnellen Einstieg enthält dieses Thema Schritte zum Erstellen eines Clusters und von Knoten mit Standardeinstellungen. Bevor Sie einen Cluster und Knoten für den Produktionseinsatz erstellen, empfehlen wir Ihnen, sich mit allen Einstellungen vertraut zu machen und einen Cluster und Knoten mit den Einstellungen bereitzustellen, die Ihren Anforderungen entsprechen. Weitere Informationen finden Sie

unter Erstellen eines Amazon-EKS-Clusters und Amazon-EKS-Knoten. Einige Einstellungen können nur aktiviert werden, wenn Sie Ihren Cluster und Ihre Knoten erstellen.

Sie können einen Cluster mit einem der folgenden Knotentypen erstellen. Weitere Informationen zu den einzelnen Typen finden Sie unter Amazon-EKS-Knoten. Nachdem Ihr Cluster bereitgestellt wurde, können Sie andere Knotentypen hinzufügen.

- Fargate – Linux – Wählen Sie diesen Knotentyp, wenn Sie Linux-Anwendungen auf AWS Fargate ausführen möchten. Fargate ist eine Serverless-Compute-Engine, mit der Sie Kubernetes-Pods bereitstellen können, ohne Amazon-EC2-Instances zu verwalten.
- Verwaltete Knoten – Linux – Wählen Sie diesen Knotentyp aus, wenn Sie Amazon-Linux-Anwendungen auf Amazon-EC2-Instances ausführen möchten. Obwohl dies in diesem Handbuch nicht behandelt wird, können Sie auch selbstverwaltete Windows- und Bottlerocket-Knoten zu Ihrem Cluster hinzufügen.

Erstellen Sie Ihren Amazon-EKS-Cluster mit dem folgenden Befehl. Sie können *my-cluster* durch Ihren eigenen Wert ersetzen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein. *region-code*Ersetzen Sie es durch ein beliebiges AWS-Region , das von Amazon EKS unterstützt wird. Eine Liste von AWS-Regionen finden Sie unter Amazon EKS-Endpunkte und Kontingente im AWS Allgemeinen Referenzhandbuch.

Fargate – Linux

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

Managed nodes – Linux

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

Die Clustererstellung dauert mehrere Minuten. Während der Erstellung werden mehrere Ausgabezeilen angezeigt. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

```
[...]
```

```
[#]  EKS cluster "my-cluster" in "region-code" region is ready
```

`eksctl` hat eine `kubectl config`-Datei in `~/.kube` erstellt oder die Konfiguration des neuen Clusters innerhalb einer vorhandenen `config`-Datei in `~/.kube` auf Ihrem Computer hinzugefügt.

Nachdem die Clustererstellung abgeschlossen ist, können Sie sich den `eksctl-`*`my-cluster`*`-cluster` in der AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation genannten AWS CloudFormation Stack ansehen, um alle Ressourcen zu sehen, die erstellt wurden.

Schritt 2: Kubernetes-Ressourcen anzeigen

1. Zeigen Sie Ihre Cluster-Knoten an.

   ```
   kubectl get nodes -o wide
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   Fargate – Linux

   ```
   NAME                                                STATUS   ROLES    AGE
      VERSION              INTERNAL-IP   EXTERNAL-IP   OS-IMAGE         KERNEL-
   VERSION                    CONTAINER-RUNTIME
   fargate-ip-192-0-2-0.region-code.compute.internal   Ready    <none>
     8m3s    v1.2.3-eks-1234567   192.0.2.0     <none>        Amazon Linux 2
     1.23.456-789.012.amzn2.x86_64   containerd://1.2.3
   fargate-ip-192-0-2-1.region-code.compute.internal   Ready    <none>
     7m30s   v1.2.3-eks-1234567   192-0-2-1     <none>        Amazon Linux 2
     1.23.456-789.012.amzn2.x86_64   containerd://1.2.3
   ```

   Managed nodes – Linux

   ```
   NAME                                          STATUS   ROLES    AGE    VERSION
             INTERNAL-IP   EXTERNAL-IP   OS-IMAGE         KERNEL-VERSION
          CONTAINER-RUNTIME
   ip-192-0-2-0.region-code.compute.internal   Ready    <none>   6m7s
      v1.2.3-eks-1234567   192.0.2.0     192.0.2.2     Amazon Linux 2
     1.23.456-789.012.amzn2.x86_64   containerd://1.2.3
   ip-192-0-2-1.region-code.compute.internal   Ready    <none>   6m4s
      v1.2.3-eks-1234567   192.0.2.1     192.0.2.3     Amazon Linux 2
     1.23.456-789.012.amzn2.x86_64   containerd://1.2.3
   ```

Weitere Informationen dazu, was in der Ausgabe angezeigt wird, finden Sie unter Anzeigen der Kubernetes-Ressourcen.

2. Zeigen Sie die Workloads an, die auf Ihrem Cluster ausgeführt werden.

```
kubectl get pods -A -o wide
```

Eine Beispielausgabe sieht wie folgt aus.

Fargate – Linux

```
NAMESPACE     NAME                       READY   STATUS    RESTARTS   AGE   IP
          NODE                                                NOMINATED NODE
 READINESS GATES
kube-system   coredns-1234567890-abcde   1/1     Running   0          18m
  192.0.2.0   fargate-ip-192-0-2-0.region-code.compute.internal   <none>
   <none>
kube-system   coredns-1234567890-12345   1/1     Running   0          18m
  192.0.2.1   fargate-ip-192-0-2-1.region-code.compute.internal   <none>
   <none>
```

Managed nodes – Linux

```
NAMESPACE     NAME                       READY   STATUS    RESTARTS   AGE     IP
          NODE                                        NOMINATED NODE   READINESS
 GATES
kube-system   aws-node-12345             1/1     Running   0          7m43s
   192.0.2.1   ip-192-0-2-1.region-code.compute.internal   <none>
 <none>
kube-system   aws-node-67890             1/1     Running   0          7m46s
   192.0.2.0   ip-192-0-2-0.region-code.compute.internal   <none>
 <none>
kube-system   coredns-1234567890-abcde   1/1     Running   0          14m
   192.0.2.3   ip-192-0-2-3.region-code.compute.internal   <none>
 <none>
kube-system   coredns-1234567890-12345   1/1     Running   0          14m
   192.0.2.4   ip-192-0-2-4.region-code.compute.internal   <none>
 <none>
kube-system   kube-proxy-12345           1/1     Running   0          7m46s
   192.0.2.0   ip-192-0-2-0.region-code.compute.internal   <none>
 <none>
```

```
kube-system   kube-proxy-67890           1/1     Running   0          7m43s
   192.0.2.1   ip-192-0-2-1.region-code.compute.internal   <none>
 <none>
```

Weitere Informationen dazu, was in der Ausgabe angezeigt wird, finden Sie unter Anzeigen der Kubernetes-Ressourcen.

Schritt 3: Löschen Ihrer Cluster und Knoten

Nachdem Sie mit dem Cluster und den Knoten fertig sind, die Sie für dieses Tutorial erstellt haben, sollten Sie den Cluster und die Knoten mit dem folgenden Befehl löschen. Wenn Sie vor dem Bereinigen mehr mit diesem Cluster tun möchten, lesen Sie Nächste Schritte.

```
eksctl delete cluster --name my-cluster --region region-code
```

Nächste Schritte

Die folgenden Dokumentationsthemen helfen Ihnen bei der Erweiterung der Funktionalität Ihres Clusters.

- Stellen Sie eine Beispielanwendung für Ihr Cluster bereit.
- Die IAM-Prinzipal, der den Cluster erstellt hat, ist der einzige Prinzipal, der Aufrufe an den Kubernetes-API-Server mit `kubectl` oder AWS Management Console tätigen kann. Wenn Sie möchten, dass andere IAM-Prinzipale Zugriff auf Ihren Cluster haben, müssen Sie sie hinzufügen. Weitere Informationen finden Sie unter Zugriff auf Kubernetes APIs gewähren und Erforderliche Berechtigungen.
- Bevor Sie einen Cluster für die Produktion bereitstellen, empfehlen wir Ihnen, sich mit allen Einstellungen für Cluster und Knoten vertraut zu machen. Einige Einstellungen, (z. B. das Aktivieren des SSH-Zugriffs auf Amazon-EC2-Knoten) müssen vorgenommen werden, wenn der Cluster erstellt wird.
- Um die Sicherheit für Ihren Cluster zu erhöhen, konfigurieren Sie das Amazon VPC Container Networking Interface-Plugin für die Verwendung von IAM-Rollen für Servicekonten.

Erste Schritte mit Amazon EKS — AWS Management Console und AWS CLI

Dieses Handbuch hilft Ihnen dabei, alle erforderlichen Ressourcen für den Einstieg in Amazon Elastic Kubernetes Service (Amazon EKS) mithilfe von AWS Management Console und zu erstellen. AWS CLI In diesem Leitfaden erstellen Sie jede Ressource manuell. Am Ende dieses Tutorials haben Sie einen laufenden Amazon-EKS-Cluster, auf dem Sie Anwendungen bereitstellen können.

Die Verfahren in diesem Handbuch geben Ihnen vollständige Einsichten in die Erstellung der einzelnen Ressourcen und in die Interaktionen der Ressourcen. Wenn die Mehrzahl der Ressourcen automatisch für Sie erstellt werden soll, verwenden Sie die `eksctl`-CLI, um Ihre Cluster- und Knoten zu erstellen. Weitere Informationen finden Sie unter Erste Schritte mit Amazon EKS – `eksctl`.

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Werkzeuge und Ressourcen installieren und konfigurieren, die Sie zum Erstellen und Verwalten eines Amazon-EKS-Clusters benötigen.

- AWS CLI— Ein Befehlszeilentool für die Arbeit mit AWS Diensten, einschließlich Amazon EKS. Weitere Informationen finden Sie unter Installieren, Aktualisieren und Deinstallieren von AWS CLI im AWS Command Line Interface -Benutzerhandbuch. Nach der Installation empfehlen wir AWS CLI, dass Sie es auch konfigurieren. Weitere Informationen finden Sie unter Schnellkonfiguration mit `aws configure` im AWS Command Line Interface -Benutzerhandbuch.
- **`kubectl`** – Ein Befehlszeilentool für die Arbeit mit Kubernetes-Clustern. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- Erforderliche IAM-Berechtigungen — Der von Ihnen verwendete IAM-Sicherheitsprinzipal muss über Berechtigungen für die Arbeit mit Amazon EKS-IAM-Rollen, serviceverknüpften Rollen AWS CloudFormation, einer VPC und verwandten Ressourcen verfügen. Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Elastic Kubernetes Service und Verwenden von serviceverknüpften Rollen im IAM-Benutzerhandbuch. Sie müssen alle Schritte in diesem Handbuch als derselbe Benutzer ausführen. Führen Sie den folgenden Befehl aus, um den aktuellen Benutzer zu überprüfen:

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

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

Schritt 1: Erstellen Sie Ihre Amazon-EKS-Cluster

> **Important**
>
> Für einen möglichst einfachen und schnellen Einstieg enthält dieses Thema Schritte zum Erstellen eines Clusters mit Standardeinstellungen. Bevor Sie einen Cluster für den Produktionseinsatz erstellen, empfehlen wir Ihnen, sich mit allen Einstellungen vertraut zu machen und einen Cluster mit den Einstellungen bereitzustellen, die Ihren Anforderungen entsprechen. Weitere Informationen finden Sie unter Erstellen eines Amazon-EKS-Clusters. Einige Einstellungen können nur aktiviert werden, wenn Sie Ihren Cluster erstellen.

Um Ihren Cluster zu erstellen

1. Erstellen Sie eine Amazon VPC mit öffentlichen und privaten Subnetzen, die die Amazon-EKS-Anforderungen erfüllt. Ersetzen Sie *region-code* durch eine AWS-Region , die von Amazon EKS unterstützt wird. Eine Liste von AWS-Regionen finden Sie unter Amazon EKS-Endpunkte und Kontingente im AWS Allgemeinen Referenzhandbuch.Sie können *my-eks-vpc-stack* mit einem beliebigen Namen ersetzen, den Sie wählen.

   ```
   aws cloudformation create-stack \
     --region region-code \
     --stack-name my-eks-vpc-stack \
     --template-url https://s3.us-west-2.amazonaws.com/amazon-
   eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml
   ```

   > **Tip**
   >
   > Für eine Liste aller Ressourcen, die der vorherige Befehl erstellt hat, öffnen Sie die AWS CloudFormation -Konsole unter https://console.aws.amazon.com/cloudformation.

Wählen Sie das *my-eks-vpc-stack*-Stack, und wählen Sie dann die Registerkarte Ressourcen.

2. Erstellen Sie eine Cluster-IAM-Rolle und fügen Sie ihr die erforderliche verwaltete Amazon EKS IAM-Richtlinie hinzu. KubernetesVon Amazon EKS verwaltete Cluster rufen in Ihrem Namen andere AWS Services auf, um die Ressourcen zu verwalten, die Sie mit dem Service verwenden.

   a. Kopieren Sie den folgenden Inhalt in eine Datei namens *eks-cluster-role-trust-policy.json*.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "eks.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
       }
     ]
   }
   ```

   b. Erstellen Sie die -Rolle.

   ```
   aws iam create-role \
     --role-name myAmazonEKSClusterRole \
     --assume-role-policy-document file://"eks-cluster-role-trust-policy.json"
   ```

   c. Hängen Sie die erforderliche von Amazon EKS verwaltete IAM-Richtlinie an die Rolle an.

   ```
   aws iam attach-role-policy \
     --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy \
     --role-name myAmazonEKSClusterRole
   ```

3. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

   Stellen Sie sicher, dass das oben rechts auf Ihrer Konsole AWS-Region angezeigte Objekt AWS-Region das ist, in dem Sie Ihren Cluster erstellen möchten. Ist dies nicht der Fall, wählen Sie das

Dropdownmenü neben dem AWS-Region Namen aus und wählen Sie den aus AWS-Region , den Sie verwenden möchten.

4. Wählen Sie Add cluster (Cluster hinzufügen) und dann Create (Erstellen) aus. Wenn diese Option nicht angezeigt wird, wählen Sie zunächst im linken Navigationsbereich Clusters (Cluster) aus.
5. Führen Sie auf der Seite Configure cluster (Cluster konfigurieren) die folgenden Schritte aus:
   a. Geben Sie einen Namen für Ihren Cluster ein, z. B. **`my-cluster`**. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Er muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
   b. Wählen Sie für Cluster Service Role die Option *`ClusterRoleMyAmazonEks`* aus.
   c. Belassen Sie die restlichen Einstellungen auf ihren Standardwerten und wählen Sie Weiter aus.
6. Gehen Sie auf der Seite Netzwerk angeben wie folgt vor:
   a. Wählen Sie in der Dropdown-Liste VPC (VPC) die ID der VPC aus, die Sie in einem vorherigen Schritt erstellt haben. Es ist so etwas wie *`vpc-00x0000x000x0x000`* | *`my-eks-vpc-stack-VPC`*.
   b. Belassen Sie die restlichen Einstellungen auf ihren Standardwerten und wählen Sie Weiter aus.
7. Wählen Sie auf der Seite Beobachtbarkeit konfigurieren die Option Weiter aus.
8. Wählen Sie auf der Seite Add-Ons auswählen die Option Weiter aus.

   Weitere Informationen zu Add-Ons finden Sie unter Amazon-EKS-Add-ons.
9. Wählen Sie auf der Seite Konfigurieren ausgewählter Add-Ons-Einstellungen die Option Weiter aus.
10. Wählen Sie auf der Seite Überprüfen und erstellen die Option Erstellen aus.

    Rechts neben dem Clusternamen lautet der Clusterstatus für einige Minuten Erstellen, bis der Cluster-Bereitstellungsprozess abgeschlossen ist. Fahren Sie nicht mit dem nächsten Schritt fort, bis der Status Aktiv lautet.

> **Note**
>
> Sie erhalten möglicherweise eine Fehlermeldung, dass eine der Availability Zones in Ihrer Anfrage nicht über genügend Kapazität zum Erstellen eines Amazon-EKS-Clusters verfügt. Wenn dies der Fall ist, enthält die Fehlerausgabe die Availability Zones, die einen neuen Cluster unterstützen können. Versuchen Sie, Ihren Cluster mit mindestens zwei Subnetzen erneut zu erstellen, die sich in den unterstützten Availability Zones für Ihr Konto befinden. Weitere Informationen finden Sie unter Unzureichende Kapazität.

Schritt 2: Konfigurieren Sie Ihren Computer für die Kommunikation mit Ihrem Cluster

In diesem Abschnitt erstellen Sie eine `kubeconfig`-Datei für Ihren Cluster. Die Einstellungen in dieser Datei ermöglichen der `kubectl`-CLI die Kommunikation mit Ihrem Cluster.

Konfigurieren Sie Ihren Computer wie folgt für die Kommunikation mit Ihrem Cluster

1. Erstellen oder aktualisieren Sie eine `kubeconfig`-Datei für Ihren Cluster. Ersetzen Sie *`region-code`* durch die AWS-Region , in der Sie Ihren Cluster erstellt haben. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

   ```
   aws eks update-kubeconfig --region region-code --name my-cluster
   ```

   Standardmäßig wird die `config`-Datei in `~/.kube` erstellt oder die Konfiguration des neuen Clusters wird einer vorhandenen `config`-Datei in `~/.kube` hinzugefügt.

2. Testen Sie Ihre Konfiguration.

   ```
   kubectl get svc
   ```

   > **Note**
   >
   > Wenn Sie Autorisierungs- oder Ressourcenfehler erhalten, finden Sie weitere Informationen unter Nicht autorisiert oder Zugriff verweigert (`kubectl`) im Thema zur Fehlerbehebung.

Eine Beispielausgabe sieht wie folgt aus.

```
NAME             TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
svc/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   1m
```

Schritt 3: Erstellen von Knoten

> **Important**
>
> Für einen möglichst einfachen und schnellen Einstieg enthält dieses Thema Schritte zum Erstellen von Knoten mit Standardeinstellungen. Bevor Sie Knoten für den Produktionseinsatz erstellen, empfehlen wir Ihnen, sich mit allen Einstellungen vertraut zu machen und Knoten mit den Einstellungen bereitzustellen, die Ihren Anforderungen entsprechen. Weitere Informationen finden Sie unter Amazon-EKS-Knoten. Einige Einstellungen können nur aktiviert werden, wenn Sie Ihre Knoten erstellen.

Sie können einen Cluster mit einem der folgenden Knotentypen erstellen. Weitere Informationen zu den einzelnen Typen finden Sie unter Amazon-EKS-Knoten. Nachdem Ihr Cluster bereitgestellt wurde, können Sie andere Knotentypen hinzufügen.

- Fargate – Linux – Wählen Sie diesen Knotentyp, wenn Sie Linux-Anwendungen auf AWS Fargate ausführen möchten. Fargate ist eine Serverless-Compute-Engine, mit der Sie Kubernetes-Pods bereitstellen können, ohne Amazon-EC2-Instances zu verwalten.
- Verwaltete Knoten – Linux – Wählen Sie diesen Knotentyp aus für die Ausführung von Amazon-Linux-Anwendungen auf Amazon-EC2-Instances. Obwohl dies in diesem Handbuch nicht behandelt wird, können Sie auch selbstverwaltete Windows- und Bottlerocket-Knoten zu Ihrem Cluster hinzufügen.

Fargate – Linux

Erstellen eines Fargate-Profils. Wenn Kubernetes-Pods mit Kriterien bereitgestellt werden, die den im Profil definierten Kriterien entsprechen, werden die Pods in Fargate bereitgestellt.

Ein Fargate-Profil erstellen

1. Erstellen Sie eine IAM-Rolle und fügen Sie ihr die erforderliche von Amazon EKS IAM verwaltete Richtlinie hinzu. Wenn Ihr Cluster Pods auf der Fargate-Infrastruktur erstellt wird, müssen die Komponenten, die auf der Fargate-Infrastruktur ausgeführt werden, in Ihrem Namen AWS API-Aufrufe tätigen. Auf diese Weise können sie Aktionen wie das Abrufen von Container-Images aus Amazon ECR oder das Weiterleiten von Protokollen an andere AWS Dienste ausführen. Die Amazon-EKS-Pod-Ausführungsrolle stellt die entsprechenden IAM-Berechtigungen bereit.

   a. Kopieren Sie den folgenden Inhalt in eine Datei namens *pod-execution-role-trust-policy*`.json`. *region-code*Ersetzen Sie es durch AWS-Region das, in dem sich Ihr Cluster befindet. Wenn Sie dieselbe Rolle für alle AWS-Regionen in Ihrem Konto verwenden möchten, *region-code* ersetzen Sie es durch*. Ersetzen Sie *111122223333* durch Ihre Konto-ID und *my-cluster* durch den Namen Ihres Clusters. Wenn Sie dieselbe Rolle für alle Cluster in Ihrem Konto verwenden möchten, ersetzen Sie *my-cluster* durch *.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Condition": {
         "ArnLike": {
            "aws:SourceArn": "arn:aws:eks:region-
code:111122223333:fargateprofile/my-cluster/*"
         }
      },
      "Principal": {
        "Service": "eks-fargate-pods.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

   b. Erstellen einer IAM-Pod-Ausführungsrolle

```
aws iam create-role \
  --role-name AmazonEKSFargatePodExecutionRole \
```

```
  --assume-role-policy-document file://"pod-execution-role-trust-
policy.json"
```

c. Hängen Sie die erforderliche von Amazon EKS verwaltete IAM-Richtlinie an die Rolle an.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/
AmazonEKSFargatePodExecutionRolePolicy \
  --role-name AmazonEKSFargatePodExecutionRole
```

2. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
3. Wählen Sie auf der Cluster-Seite den *my-cluster*-Cluster aus.
4. Führen Sie auf der Seite ***my-cluster*** die folgenden Schritte aus:
   a. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.
   b. Wählen Sie unter Fargate Profiles (Fargate-Profile) die Option Add Fargate Profile (Fargate-Profil hinzufügen) aus.
5. Gehen Sie auf der Seite Configure Fargate Profile (Fargate-Profil konfigurieren) wie folgt vor:
   a. Geben Sie unter Name einen eindeutigen Namen für Ihr Fargate-Profil ein, wie z. B. ***my-profile***.
   b. Wählen Sie für die Pod-Ausführungsrolle die AmazonEKS aus FargatePodExecutionRole, die Sie in einem vorherigen Schritt erstellt haben.
   c. Wählen Sie die Dropdown-Liste Subnets (Subnetze) aus und deaktivieren Sie alle Subnetze mit `Public` im Namen. Für Pods, die auf Fargate ausgeführt werden, werden nur private Subnetze unterstützt.
   d. Wählen Sie Weiter aus.
6. Führen Sie auf der Seite Configure Pod selection (-Auswahl konfigurieren) die folgenden Schritte aus:
   a. Geben Sie für Namespace **`default`** ein.
   b. Wählen Sie Next (Weiter).
7. Überprüfen Sie auf der Seite Überprüfen und erstellen die Informationen für Ihr -Profil und wählen Sie Erstellen aus.

8. Nach einigen Minuten ändert sich der Status im Abschnitt Konfiguration der Fargate-Gruppe von Wird erstellt auf Aktiv. Fahren Sie nicht mit dem nächsten Schritt fort, bis der Status Aktiv lautet.
9. Wenn Sie vorhaben, alle Pods auf Fargate (keine auf Amazon-EC2-Knoten) bereitzustellen, führen Sie die folgenden Schritte aus, um ein weiteres Fargate-Profil zu erstellen und den Standard-Namenslöser (CoreDNS) auf Fargate auszuführen.

   > **Note**
   >
   > Wenn Sie das nicht tun, haben Sie zu diesem Zeitpunkt keine Knoten.

   a. Auf der Seite Fargate Profile (Fargate-Profil) wählen Sie *`my-profile`* aus.
   b. Wählen Sie unter Fargate-Profile die Option Fargate-Profil hinzufügen aus.
   c. Geben Sie unter Name ***`CoreDNS`*** ein.
   d. Wählen Sie für die Pod-Ausführungsrolle die AmazonEKS aus FargatePodExecutionRole, die Sie in einem vorherigen Schritt erstellt haben.
   e. Wählen Sie die Dropdown-Liste Subnets (Subnetze) aus und deaktivieren Sie alle Subnetze mit `Public` im Namen. Für Pods, die auf Fargate ausgeführt werden, werden nur private Subnetze unterstützt.
   f. Wählen Sie Next (Weiter).
   g. Geben Sie für Namespace **`kube-system`** ein.
   h. Klicken Sie auf Übereinstimmung von Labels und wählen Sie dann Label hinzufügen aus.
   i. Geben Sie **`k8s-app`** für Key(Schlüssel) und **`kube-dns`** für Value (Wert) ein. Dies ist notwendig, damit der Standardname-Resolver (CoreDNS) auf Fargate bereitgestellt werden kann.
   j. Wählen Sie Next (Weiter).
   k. Überprüfen Sie auf der Seite Überprüfen und erstellen die Informationen für Ihr -Profil und wählen Sie Erstellen aus.
   l. Verwenden Sie den folgenden Befehl, um die standardmäßige `eks.amazonaws.com/compute-type : ec2`-Anmerkung aus den CoreDNS Pods zu entfernen.

```
kubectl patch deployment coredns \
    -n kube-system \
```

```
    --type json \
    -p='[{"op": "remove", "path": "/spec/template/metadata/annotations/
eks.amazonaws.com~1compute-type"}]'
```

> **Note**
>
> Das System erstellt und verteilt zwei Knoten basierend auf der von Ihnen hinzugefügten Fargate-Profilbeschriftung. In den Node groups (Knotengruppen) werden keine Elemente angezeigt, da sie für Fargate-Knoten nicht gelten, aber die neuen Knoten werden auf der Registerkarte Overview (Übersicht) aufgelistet.

Managed nodes – Linux

Erstellen Sie eine verwaltete Knotengruppe und geben Sie die Subnetze und Knoten-IAM-Rolle an, welche Sie in den vorherigen Schritten erstellt haben.

Erstellen Ihrer verwalteten Amazon-EC2-Linux-Knotengruppe

1. Erstellen Sie eine Knoten-IAM-Rolle und fügen Sie ihr die erforderliche von Amazon EKS IAM verwaltete Richtlinie hinzu. Der Amazon EKS Node `kubelet` Daemon ruft in Ihrem Namen AWS APIs auf. Knoten erhalten über ein IAM-Instance-Profil und zugehörige Richtlinien Berechtigungen für diese API-Aufrufe.

   a. Kopieren Sie den folgenden Inhalt in eine Datei namens *`node-role-trust-policy.json`*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

b. Erstellen Sie die Knoten-IAM-Rolle.

```
aws iam create-role \
  --role-name myAmazonEKSNodeRole \
  --assume-role-policy-document file://"node-role-trust-policy.json"
```

c. Hängen Sie die erforderlichen verwalteten IAM-Richtlinien an die Rolle an.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name myAmazonEKSNodeRole
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name myAmazonEKSNodeRole
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
  --role-name myAmazonEKSNodeRole
```

2. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
3. Wählen Sie den Namen des Clusters aus, den Sie in Schritt 1: Erstellen Sie Ihre Amazon-EKS-Cluster erstellt haben, z. B. *my-cluster*.
4. Führen Sie auf der Seite ***my-cluster*** die folgenden Schritte aus:

   a. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.

   b. Wählen Sie Add Node Group (Knotengruppe hinzufügen) aus.

5. Führen Sie auf der Seite Configure Node Group (Knotengruppe konfigurieren) die folgenden Schritte aus:

   a. Für Name geben Sie einen eindeutigen Namen für die verwaltete Knotengruppe ein, wie z. B. ***my-nodegroup***. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten.

   b. Wählen Sie für den Namen der Node-IAM-Rolle die *NodeRoleMyAmazonEKS-Rolle* aus, die Sie in einem vorherigen Schritt erstellt haben. Wir empfehlen, dass jede Knotengruppe ihre eigene eindeutige IAM-Rolle verwendet.

   c. Wählen Sie Next (Weiter).

6. Akzeptieren Sie auf der Seite Computing- und Skalierungskonfiguration festlegen die Standardwerte und wählen Sie Weiter aus.
7. Akzeptieren Sie auf der Seite Netzwerk angeben die Standardwerte und wählen Sie Weiter aus.
8. Überprüfen Sie auf der Seite Review and create (Überprüfen und Erstellen) die Konfiguration der verwalteten Knoten, und wählen Sie Create (Erstellen).
9. Nach einigen Minuten ändert sich der Status im Abschnitt Konfiguration der Knotengruppe von Wird erstellt auf Aktiv. Fahren Sie nicht mit dem nächsten Schritt fort, bis der Status Aktiv lautet.

Schritt 4: Ressourcen anzeigen

Sie können Ihre Knoten und Kubernetes-Workloads anzeigen.

So zeigen Sie Ihre Knoten und Workloads an

1. Wählen Sie im linken Navigationsbereich die Option Cluster aus. Wählen Sie in der Liste Clusters (Cluster) den Namen des erstellten Clusters aus, z. B. *my-cluster*.
2. Wählen Sie auf der Seite ***my-cluster*** Folgendes aus:
   a. Registerkarte Compute (Datenverarbeitung) – Es wird eine Liste der Knoten angezeigt, die für den Cluster bereitgestellt wurden. Sie können den Namen eines Knotens auswählen, um weitere Informationen darüber anzuzeigen.
   b. Registerkarte Ressources (Ressourcen) – Es werden alle Kubernetes-Ressourcen angezeigt, die standardmäßig in einem Amazon-EKS-Cluster bereitgestellt werden. Wählen Sie einen Ressourcentyp in der Konsole aus, um mehr über ihn zu erfahren.

Schritt 5: Löschen von Ressourcen

Nachdem Sie mit dem Cluster und den Knoten fertig sind, die Sie für dieses Tutorial erstellt haben, sollten Sie die Ressourcen, die Sie erstellt haben, löschen. Wenn Sie vor dem Löschen der Ressourcen mehr mit diesem Cluster tun möchten, lesen Sie Nächste Schritte.

So löschen Sie die Ressourcen, die Sie in diesem Leitfaden erstellt haben

1. Löschen Sie alle von Ihnen erstellten Knotengruppen oder Fargate-Profile.

a. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

b. Wählen Sie im linken Navigationsbereich die Option Cluster aus. Wählen Sie *my-cluster* in der Liste der Cluster aus.

c. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.

d. Wenn Sie eine Knotengruppe erstellt haben, Sie die Knotengruppe *my-nodegroup* und dann Delete (Löschen) aus. Geben Sie ***my-nodegroup*** ein und klicken Sie auf Delete (Löschen).

e. Wählen Sie dies für jedes von Ihnen erstellte Fargate-Profil und dann Delete (Löschen) aus. Geben Sie den Namen des Profils ein und wählen Sie dann Löschen aus.

> **Note**
>
> Wenn Sie ein zweites Fargate-Profil löschen, müssen Sie möglicherweise warten, bis das erste Profil gelöscht wurde.

f. Fahren Sie nicht fort, bis die Knotengruppe oder Fargate-Profile gelöscht sind.

2. Löschen Sie den -Cluster.

a. Wählen Sie im linken Navigationsbereich die Option Cluster aus. Wählen Sie *my-cluster* in der Liste der Cluster aus.

b. Wählen Sie Delete cluster (Cluster löschen) aus.

c. Geben Sie ***my-cluster*** ein und wählen Sie Delete (Löschen). Fahren Sie nicht fort, bis der Cluster gelöscht wurde.

3. Löschen Sie den AWS CloudFormation VPC-Stack, den Sie erstellt haben.

a. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

b. Wählen Sie den *my-eks-vpc-stack*-Stack und anschließend Delete Löschen aus.

c. Wählen Sie im Bestätigungsdialogfeld ***my-eks-vpc-stack*** löschen Stack löschen aus.

4. Löschen Sie die IAM-Rollen, die Sie erstellt haben.

a. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

b. Wählen Sie im linken Navigationsbereich Roles aus.

c. *Wählen Sie jede Rolle, die Sie erstellt habenClusterRole, aus der Liste aus (**MyAmazonEks** sowie AmazonEks FargatePodExecutionRoleoder **MyAmazonEks**). NodeRole* Klicken Sie auf Delete (Löschen), geben Sie den angeforderten Bestätigungstext ein und wählen Sie Delete (Löschen) aus.

Nächste Schritte

Die folgenden Dokumentationsthemen helfen Ihnen bei der Erweiterung der Funktionalität Ihres Clusters.

- Der IAM-Prinzipal, der den Cluster erstellt hat, ist der einzige Prinzipal, der Aufrufe an den Kubernetes-API-Server mit `kubectl` oder AWS Management Console tätigen kann. Wenn Sie möchten, dass andere IAM-Prinzipale Zugriff auf Ihren Cluster haben, müssen Sie sie hinzufügen. Weitere Informationen finden Sie unter Zugriff auf Kubernetes APIs gewähren und Erforderliche Berechtigungen.
- Stellen Sie eine Beispielanwendung für Ihr Cluster bereit.
- Bevor Sie einen Cluster für die Produktion bereitstellen, empfehlen wir Ihnen, sich mit allen Einstellungen für Cluster und Knoten vertraut zu machen. Einige Einstellungen, (z. B. das Aktivieren des SSH-Zugriffs auf Amazon-EC2-Knoten) müssen vorgenommen werden, wenn der Cluster erstellt wird.
- Um die Sicherheit für Ihren Cluster zu erhöhen, konfigurieren Sie das Amazon VPC Container Networking Interface-Plugin für die Verwendung von IAM-Rollen für Servicekonten.

Amazon-EKS-Cluster

Ein Amazon-EKS-Cluster besteht aus zwei primären Komponenten:

- Die Amazon-EKS-Steuerebene
- Amazon-EKS-Knoten, die in der Steuerebene registriert sind

Die Amazon-EKS-Steuerebene besteht aus Steuerebenenknoten, auf denen Kubernetes-Software wie `etcd` und derKubernetes-API-Server ausgeführt werden. Die Kontrollebene wird in einem Konto ausgeführt, das von verwaltet wird AWS, und die Kubernetes API wird über den Amazon EKS-Endpunkt verfügbar gemacht, der Ihrem Cluster zugeordnet ist. Jede Amazon-EKS-Cluster-Steuerungsebene ist mandantenfähig und einzigartig, sie wird über einen eigenen Satz von Amazon-EC2-Instances ausgeführt.

Alle von den `etcd` Knoten und den zugehörigen Amazon EBS-Volumes gespeicherten Daten werden mit AWS KMS verschlüsselt. Die Cluster-Steuerungsebene ist über mehrere Availability Zones verteilt und wird von einem Elastic Load Balancing Network Load Balancer unterstützt. Amazon EKS stellt in Ihren VPC-Subnetzen außerdem Elastic Network-Schnittstellen für die Verbindung zwischen Steuerebenen-Instances und -Knoten bereit (z. B. zur Unterstützung von `kubectl exec-logs-proxy`-Datenströmen).

Important

In der Amazon-EKS-Umgebung ist der `etcd`-Speicher gemäß Upstream-Empfehlungen auf 8 GiB beschränkt. Sie können eine Metrik für die aktuelle Datenbankgröße überwachen, indem Sie den folgenden Befehl ausführen. Falls die Kubernetes-Version Ihres Clusters niedriger ist als `1.28`, ersetzen Sie *`apiserver_storage_size_bytes`* durch Folgendes:

- Kubernetes-Versionen `1.27` und `1.26`: **`apiserver_storage_db_total_size_in_bytes`**
- Kubernetes-Version `1.25` und niedriger: **`etcd_db_total_size_in_bytes`**

```
kubectl get --raw=/metrics | grep "apiserver_storage_size_bytes"
```

Amazon EKS-Knoten werden in Ihrem AWS Konto ausgeführt und stellen über den API-Serverendpunkt und eine Zertifikatsdatei, die für Ihren Cluster erstellt wurde, eine Verbindung zur Kontrollebene Ihres Clusters her.

> **Note**
>
> - In Amazon-EKS-Netzwerk erfahren Sie, wie die verschiedenen Komponenten von Amazon EKS funktionieren.
> - Informationen zu verbundenen Clustern finden Sie unter Amazon-EKS-Anschluss.

Themen

Erstellen eines Amazon-EKS-Clusters

Dieses Thema bietet einen Überblick über die verfügbaren Optionen und beschreibt, was zu beachten ist, wenn Sie einen Amazon-EKS-Cluster erstellen. Informationen zum Erstellen eines Clusters auf einem AWS Outpost finden Sie unterLokale Cluster für Amazon EKS auf AWS Outposts. Wenn Sie zum ersten Mal einen Amazon-EKS-Cluster erstellen, empfehlen wir, entsprechend unseren Erste Schritte mit Amazon EKS-Leitfäden vorzugehen. Diese Leitfäden helfen, einen einfachen Standard-Cluster zu erstellen, ohne auf alle verfügbaren Optionen einzugehen.

Voraussetzungen

- Eine vorhandene VPC und Subnetze, die die Amazon-EKS-Anforderungen erfüllen. Bevor Sie einen Cluster für den Produktionseinsatz bereitstellen, empfehlen wir, dass Sie sich ein umfassendes Verständnis der VPC- und Subnetzanforderungen aneignen. Wenn Sie keine VPC und keine Subnetze haben, können Sie diese mithilfe einer von Amazon EKS bereitgestellten AWS CloudFormation Vorlage erstellen.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder. AWS CloudShell Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Ein IAM-Prinzipal mit Berechtigungen für das `create` und `describe` eines Amazon-EKS-Clusters. Weitere Informationen finden Sie unter Erstellen Sie einen lokalen Kubernetes-Cluster auf einem Outpost und Auflisten oder Beschreiben aller Cluster.

Einen Amazon-EKS-Cluster erstellen

1. Wenn Sie bereits eine Cluster-IAM-Rolle haben oder Ihren Cluster mit `eksctl` erstellen, können Sie diesen Schritt überspringen. Standardmäßig erstellt `eksctl` eine Rolle für Sie.

   Eine IAM-Rolle für einen Amazon-EKS-Cluster erstellen

   1. Führen Sie den folgenden Befehl aus, um eine JSON-Datei für eine IAM-Vertrauensrichtlinie zu erstellen.

      ```
      cat >eks-cluster-role-trust-policy.json <<EOF
      ```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

2. Erstellen Sie die Amazon-EKS-Cluster-IAM-Rolle. Falls erforderlich, stellen Sie dem Pfad auf Ihrem Computer, in den Sie im vorherigen Schritt die Datei geschrieben haben, *eks-cluster-role-trust-policy.json* voran. Der Befehl verknüpft die im vorherigen Schritt erstellte Vertrauensrichtlinie mit der Rolle. Um eine IAM-Rolle zu erstellen, muss dem IAM-Prinzipal, der die Rolle erstellt, die `iam:CreateRole`-Aktion (Berechtigung) zugewiesen werden.

   ```
   aws iam create-role --role-name myAmazonEKSClusterRole --assume-role-policy-
   document file://"eks-cluster-role-trust-policy.json"
   ```

3. Sie können entweder die von Amazon EKS verwaltete Richtlinie zuweisen oder Ihre eigene benutzerdefinierte Richtlinie erstellen. Informationen zu den Mindestberechtigungen, die Sie in Ihrer benutzerdefinierten Richtlinie verwenden müssen, finden Sie unter Amazon-EKS-Cluster-IAM-Rolle.

   Hängen Sie die von Amazon EKS verwaltete Richtlinie `AmazonEKSClusterPolicy` an die Rolle an. Um eine IAM-Richtlinie an einen IAM-Prinzipal anzuhängen, muss der Prinzipal, der die Richtlinie anhängt, eine der folgenden IAM-Aktionen (Berechtigungen) zugewiesen werden: `iam:AttachUserPolicy` oder `iam:AttachRolePolicy`.

   ```
   aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
   AmazonEKSClusterPolicy --role-name myAmazonEKSClusterRole
   ```

2. Erstellen Sie einen Amazon-EKS-Cluster.

   Sie können einen Cluster erstellen`eksctl`, indem Sie AWS Management Console, die oder die verwenden AWS CLI.

eksctl

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

Um Ihren Cluster zu erstellen

Erstellen Sie einen Amazon-EKS-`IPv4`-Cluster mit der standardmäßigen Amazon-EKS-Kubernetes-Version in Ihrer Standard- AWS-Region. Nehmen Sie vor der Ausführung des Befehls die folgenden Ersetzungen vor:

- *`region-code`*Ersetzen Sie es durch AWS-Region das, in dem Sie Ihren Cluster erstellen möchten.
- Ersetzen Sie *`my-cluster`* durch Ihren Cluster-Namen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
- Ersetzen Sie *`1.29`* durch eine von Amazon EKS unterstützte Version.

  > **Note**
  >
  > Um zu diesem Zeitpunkt einen 1.30-Cluster bereitzustellen, müssen Sie den AWS Management Console oder den AWS CLI verwenden.

- Ändern Sie die Werte für `vpc-private-subnets`, um Ihre Anforderungen zu erfüllen. Sie können auch zusätzliche IDs hinzufügen. Sie müssen mindestens zwei Subnetz-IDs angeben. Wenn Sie lieber öffentliche Subnetze angeben möchten, können Sie `--vpc-private-subnets` auf `--vpc-public-subnets` ändern. Öffentliche Subnetze verfügen über eine zugeordnete Routing-Tabelle mit einer Route zu einem Internet-Gateway, private Subnetze verfügen nicht über eine zugeordnete Routing-Tabelle. Wir empfehlen, nach Möglichkeit private Subnetze zu verwenden.

Die von Ihnen gewählten Subnetze müssen die Anforderungen für Amazon-EKS-Subnetze erfüllen. Bevor Sie Subnetze auswählen, empfehlen wir, dass Sie sich mit allen Anforderungen und Überlegungen für Amazon EKS VPC und Subnetze vertraut machen.

```
eksctl create cluster --name my-cluster --region region-code --version 1.29 --
vpc-private-subnets subnet-ExampleID1,subnet-ExampleID2 --without-nodegroup
```

Die Clusterbereitstellung dauert mehrere Minuten. Während der Cluster erstellt wird, werden mehrere Ausgabezeilen angezeigt. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

```
[#]  EKS cluster "my-cluster" in "region-code" region is ready
```

> **Tip**
>
> Sie können die meisten Optionen, die beim Erstellen eines Clusters mit `eksctl` angegeben werden können, über den Befehl **`eksctl create cluster --help`** anzeigen. Verwenden Sie eine `config`-Datei, um alle verfügbaren Optionen anzuzeigen. Weitere Informationen finden Sie unter Verwenden von Config-Dateien und im Config-Datei-Schema in der `eksctl`-Dokumentation. Auf GitHub finden Sie Beispiele für Config-Dateien.

Optionale Einstellungen

Im Folgenden finden Sie optionale Einstellungen, die bei Bedarf dem vorherigen Befehl hinzugefügt werden müssen. Sie können diese Optionen nur aktivieren, wenn Sie den Cluster erstellen, nicht danach. Wenn Sie diese Optionen angeben müssen, müssen Sie den Cluster mit einer `eksctl`-Konfigurationsdatei erstellen und die Einstellungen angeben, anstatt den vorherigen Befehl zu verwenden.

- Wenn Sie eine oder mehrere Sicherheitsgruppen angeben möchten, die Amazon EKS den erstellten Netzwerkschnittstellen zuweist, geben Sie die Option `securityGroup` an.

  Unabhängig davon, ob Sie Sicherheitsgruppen wählen oder nicht, erstellt Amazon EKS eine Sicherheitsgruppe, die die Kommunikation zwischen Ihrem Cluster und

Ihrer VPC ermöglicht. Amazon EKS verknüpft diese Sicherheitsgruppe und alle, die Sie wählen, mit den erstellten Netzwerkschnittstellen. Weitere Informationen zu der Cluster-Sicherheitsgruppe, die Amazon EKS erstellt, finden Sie unter the section called “Anforderungen an Sicherheitsgruppen”. Sie können die Regeln in der von Amazon EKS erstellten Cluster-Sicherheitsgruppe ändern.

- Wenn Sie angeben möchten welchem IPv4 CIDR-Block (Classless Inter-Domain Routing) Kubernetes Service-IP-Adressen zuweist, geben Sie die Option `serviceIPv4CIDR` an.

  Die Angabe eines eigenen Bereichs kann helfen, Konflikte zwischen Kubernetes-Services und anderen Netzwerken zu vermeiden, die mit Ihrer VPC verbunden sind. Geben Sie einen Bereich in CIDR-Notation ein. Beispiel: `10.2.0.0/16`.

  Der CIDR-Block muss die folgenden Anforderungen erfüllen:

  - Verwendet einen der folgenden Bereiche: `10.0.0.0/8`, `172.16.0.0/12`, oder `192.168.0.0/16`.
  - Hat eine Mindestgröße von `/24` und eine maximale Größe von `/12`.
  - Überschneidet sich nicht mit dem Bereich der VPC für Ihre Amazon-EKS-Ressourcen.

  Sie können diese Option nur angeben, wenn Sie die IPv4-Adressfamilie verwenden, und nur bei Erstellen des Clusters. Wenn Sie dies nicht angeben, weist Kubernetes Service-IP-Adressen aus dem CIDR-Block `10.100.0.0/16` oder `172.20.0.0/16` zu.

- Wenn Sie einen Cluster erstellen und der Cluster Pods und Services IPv6-Adressen statt IPv4-Adressen zuweisen soll, geben Sie die Option `ipFamily` an.

  Standardmäßig weist Kubernetes Ihren Pods und Services IPv4-Adressen zu. Bevor Sie sich entscheiden, die IPv6-Familie zu verwenden, stellen Sie sicher, dass Sie mit allen Überlegungen und Anforderungen in den Themen the section called “VPC-Anforderungen und -Überlegungen”, the section called “Subnetz-Anforderungen und -Überlegungen”, the section called “Anforderungen an Sicherheitsgruppen”, und the section called “IPv6” vertraut sind. Wenn Sie die IPv6-Familie verwenden, können Sie keinen Adressbereich für Kubernetes angeben, um IPv6-Serviceadressen zuzuweisen, wie Sie dies für die IPv4-Familie können. Kubernetes weist Serviceadressen aus dem eindeutigen lokalen Adressbereich zu (`fc00::/7`).

AWS Management Console

Um Ihren Cluster zu erstellen

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie Cluster hinzufügen und dann Erstellen aus.
3. Füllen Sie auf der Seite Configure cluster (Cluster konfigurieren) die folgenden Felder aus:
   - Name – Ein Name für Ihren Cluster. Der Name darf nur alphanumerische Zeichen (Groß- und Kleinschreibung beachten), Bindestriche und Unterstriche enthalten. Er muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
   - Kubernetes-Version – Die Kubernetes-Version, die für den Cluster verwendet wird. Wir empfehlen, die frühere Version auszuwählen, es sei denn, Sie benötigen eine ältere Version.
   - Cluster-Servicerolle — Wählen Sie die Amazon EKS-Cluster-IAM-Rolle aus, die Sie erstellt haben, damit die Kubernetes Kontrollebene AWS Ressourcen in Ihrem Namen verwalten kann.
   - Secrets-Verschlüsselung – (Optional) Aktivieren Sie die Secrets-Verschlüsselung von Kubernetes-Secrets mithilfe eines KMS-Schlüssels. Sie können diese auch aktivieren, nachdem Sie Ihren Cluster erstellt haben. Stellen Sie vor Aktivierung dieser Funktion sicher, dass Sie mit den Informationen in Aktivieren der Secret-Verschlüsselung in einem vorhandenen Cluster vertraut sind.
   - Tags – (Optional) Fügen Sie Ihrem Cluster beliebige Tags hinzu. Weitere Informationen finden Sie unter Kennzeichnen Ihrer Amazon EKS-Ressourcen .

     Wenn Sie mit dieser Seite fertig sind, wählen Sie Weiter aus.
4. Wählen Sie auf der Seite Specify networking (Netzwerk angeben) die Werte für die folgenden Felder aus:
   - VPC – Wählen Sie eine vorhandene VPC aus, die die Anforderungen für Amazon EKS VPC erfüllt, um Ihren Cluster zu erstellen. Bevor Sie sich für eine VPC entscheiden, empfehlen wir, sich mit allen Anforderungen und Überlegungen in Amazon EKS: VPC- und Subnetz-Anforderungen und -Überlegungen vertraut zu machen. Sie können nach

der Cluster-Erstellung nicht mehr ändern, welche VPC Sie verwenden möchten. Wenn keine VPCs aufgelistet sind, müssen Sie zuerst eine erstellen. Weitere Informationen finden Sie unter Erstellen einer VPC für Ihren Amazon-EKS-Cluster.

- Subnetze – Standardmäßig sind alle im vorherigen Feld angegebenen Subnetze in der VPC vorausgewählt. Sie müssen mindestens zwei auswählen.

  Die von Ihnen gewählten Subnetze müssen die Anforderungen für Amazon-EKS-Subnetze erfüllen. Bevor Sie Subnetze auswählen, empfehlen wir, dass Sie sich mit allen Anforderungen und Überlegungen für Amazon EKS VPC und Subnetze vertraut machen.

  Security groups (Sicherheitsgruppen) – (Optional) Geben Sie eine oder mehrere Sicherheitsgruppen an, die Amazon EKS den erstellten Netzwerkschnittstellen zuordnen soll.

  Unabhängig davon, ob Sie Sicherheitsgruppen wählen oder nicht, erstellt Amazon EKS eine Sicherheitsgruppe, die die Kommunikation zwischen Ihrem Cluster und Ihrer VPC ermöglicht. Amazon EKS verknüpft diese Sicherheitsgruppe und alle, die Sie wählen, mit den erstellten Netzwerkschnittstellen. Weitere Informationen zu der Cluster-Sicherheitsgruppe, die Amazon EKS erstellt, finden Sie unter the section called “Anforderungen an Sicherheitsgruppen”. Sie können die Regeln in der von Amazon EKS erstellten Cluster-Sicherheitsgruppe ändern.
- Cluster-IP-Adressfamilie auswählen – Sie können zwischen IPv4 und IPv6 wählen.

  Standardmäßig weist Kubernetes Ihren Pods und Services `IPv4`-Adressen zu. Bevor Sie sich entscheiden, die `IPv6`-Familie zu verwenden, stellen Sie sicher, dass Sie mit allen Überlegungen und Anforderungen in den Themen the section called “VPC-Anforderungen und -Überlegungen”, the section called “Subnetz-Anforderungen und -Überlegungen”, the section called “Anforderungen an Sicherheitsgruppen”, und the section called “IPv6” vertraut sind. Wenn Sie die `IPv6`-Familie verwenden, können Sie keinen Adressbereich für Kubernetes angeben, um `IPv6`-Serviceadressen zuzuweisen, wie Sie dies für die `IPv4`-Familie können. Kubernetes weist Serviceadressen aus dem eindeutigen lokalen Adressbereich zu (`fc00::/7`).

- (Optional) Wählen Sie Configure Kubernetes Service IP address range (Konfigurieren des IP-Adressbereichs des -Service) und geben Sie einen Service **`IPv4`** range (-Bereich für den Service) an.

Die Angabe eines eigenen Bereichs kann helfen, Konflikte zwischen Kubernetes-Services und anderen Netzwerken zu vermeiden, die mit Ihrer VPC verbunden sind. Geben Sie einen Bereich in CIDR-Notation ein. Beispiel: `10.2.0.0/16`.

Der CIDR-Block muss die folgenden Anforderungen erfüllen:

- Verwendet einen der folgenden Bereiche: `10.0.0.0/8`, `172.16.0.0/12`, oder `192.168.0.0/16`.
- Hat eine Mindestgröße von `/24` und eine maximale Größe von `/12`.
- Überschneidet sich nicht mit dem Bereich der VPC für Ihre Amazon-EKS-Ressourcen.

Sie können diese Option nur angeben, wenn Sie die `IPv4`-Adressfamilie verwenden, und nur bei Erstellen des Clusters. Wenn Sie dies nicht angeben, weist Kubernetes Service-IP-Adressen aus dem CIDR-Block `10.100.0.0/16` oder `172.20.0.0/16` zu.

- Wählen Sie eine Option für den Cluster-Endpunktzugriff aus. Nachdem Ihr Cluster erstellt wurde, können Sie diese Option ändern. Bevor Sie eine nicht standardmäßige Option auswählen, sollten Sie sich mit den Optionen und deren Auswirkungen vertraut machen. Weitere Informationen finden Sie unter Zugriffskontrolle für den Amazon-EKS-Cluster-Endpunkt.

  Wenn Sie mit dieser Seite fertig sind, wählen Sie Weiter aus.

5. (Optional) Auf der Seite Beobachtbarkeit konfigurieren können Sie Metriken und Optionen zur Steuerebenen-Protokollierung aktivieren. Standardmäßig sind alle Protokollierungstypen deaktiviert.

   - Weiteren Informationen zur Prometheus-Metrik-Option finden Sie unter Aktivieren von Prometheus-Metriken beim Erstellen eines Clusters.
   - Weitere Informationen zu den Optionen für die Steuerebenen-Protokollierung finden Sie unter Amazon-EKS-Steuerebenen-Protokollierung.

   Wenn Sie mit dieser Seite fertig sind, wählen Sie Weiter aus.

6. Wählen Sie auf der Seite Select add-ons (Add-Ons auswählen) die Add-Ons aus, die Sie Ihrem Cluster hinzufügen möchten. Sie können beliebig viele Add-ons vom Typ Amazon EKS und AWS Marketplace Add-ons auswählen. Wenn die AWS Marketplace -Add-ons, die Sie installieren möchten, nicht aufgeführt sind, können Sie nach verfügbaren AWS Marketplace -Add-ons suchen, indem Sie Text in das Suchfeld eingeben. Sie können auch nach category (Kategorie), vendor (Anbieter) oder pricing model (Preismodell) suchen und

dann die Add-Ons aus den Suchergebnissen auswählen. Wenn Sie mit dieser Seite fertig sind, wählen Sie Weiter aus.

7. Wählen Sie auf der Seite Einstellungen für ausgewählte Add-ons konfigurieren die Version aus, die Sie installieren möchten. Sie können nach der Clustererstellung jederzeit auf eine neuere Version aktualisieren. Sie können die Konfiguration jedes Add-Ons nach der Cluster-Erstellung aktualisieren. Weitere Informationen zum Konfigurieren eines Add-Ons finden Sie unter Aktualisieren eines Add-Ons. Wenn Sie mit dieser Seite fertig sind, wählen Sie Weiter aus.
8. Überprüfen Sie auf der Seite Review and create (Überprüfen und erstellen) die Informationen, die Sie auf den vorherigen Seiten eingegeben oder ausgewählt haben. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie zufrieden sind, klicken Sie auf Create app (Anwendung erstellen). Das Feld Status zeigt CREATING (WIRD ERSTELLT) an, während der Cluster bereitgestellt wird.

   > **Note**
   >
   > Sie erhalten möglicherweise eine Fehlermeldung, dass eine der Availability Zones in Ihrer Anfrage nicht über genügend Kapazität zum Erstellen eines Amazon-EKS-Clusters verfügt. Wenn dies der Fall ist, enthält die Fehlerausgabe die Availability Zones, die einen neuen Cluster unterstützen können. Versuchen Sie, Ihren Cluster mit mindestens zwei Subnetzen erneut zu erstellen, die sich in den unterstützten Availability Zones für Ihr Konto befinden. Weitere Informationen finden Sie unter Unzureichende Kapazität.

   Die Clusterbereitstellung dauert mehrere Minuten.

AWS CLI

Um Ihren Cluster zu erstellen

1. Erstellen Sie den Cluster mit dem folgenden Befehl. Nehmen Sie vor der Ausführung des Befehls die folgenden Ersetzungen vor:
   - *region-code*Ersetzen Sie es durch AWS-Region das, in dem Sie Ihren Cluster erstellen möchten.

- Ersetzen Sie *my-cluster* durch Ihren Cluster-Namen. Der Name darf nur alphanumerische Zeichen (Groß- und Kleinschreibung beachten), Bindestriche und Unterstriche enthalten. Er muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
- Ersetzen Sie *1.30* durch eine von Amazon EKS unterstützte Version.
- Ersetzen Sie *111122223333* durch Ihre Konto-ID und *myAmazonEKSClusterRole* durch den Namen Ihres IAM-Rollen-Clusters.
- Ersetzen Sie `subnetIds` durch Ihre eigenen Werte. Sie können auch zusätzliche IDs hinzufügen. Sie müssen mindestens zwei Subnetz-IDs angeben.

  Die von Ihnen gewählten Subnetze müssen die Anforderungen für Amazon-EKS-Subnetze erfüllen. Bevor Sie Subnetze auswählen, empfehlen wir, dass Sie sich mit allen Anforderungen und Überlegungen für Amazon EKS VPC und Subnetze vertraut machen.
- Wenn Sie keine Sicherheitsgruppen-ID angeben möchten, entfernen Sie `,securityGroupIds=sg-`*ExampleID1* aus dem Befehl. Wenn Sie eine oder mehrere Sicherheitsgruppen-IDs angeben möchten, ersetzen Sie die Werte für `securityGroupIds` durch Ihre eigenen Werte. Sie können auch zusätzliche IDs hinzufügen.

  Unabhängig davon, ob Sie Sicherheitsgruppen wählen oder nicht, erstellt Amazon EKS eine Sicherheitsgruppe, die die Kommunikation zwischen Ihrem Cluster und Ihrer VPC ermöglicht. Amazon EKS verknüpft diese Sicherheitsgruppe und alle, die Sie wählen, mit den erstellten Netzwerkschnittstellen. Weitere Informationen zu der Cluster-Sicherheitsgruppe, die Amazon EKS erstellt, finden Sie unter the section called “Anforderungen an Sicherheitsgruppen”. Sie können die Regeln in der von Amazon EKS erstellten Cluster-Sicherheitsgruppe ändern.

```
aws eks create-cluster --region region-code --name my-cluster --kubernetes-
version 1.30 \
   --role-arn arn:aws:iam::111122223333:role/myAmazonEKSClusterRole \
   --resources-vpc-config
 subnetIds=subnet-ExampleID1,subnet-ExampleID2,securityGroupIds=sg-ExampleID1
```

> **Note**
>
> Sie erhalten möglicherweise eine Fehlermeldung, dass eine der Availability Zones in Ihrer Anfrage nicht über genügend Kapazität zum Erstellen eines Amazon-EKS-Clusters verfügt. Wenn dies der Fall ist, enthält die Fehlerausgabe die Availability Zones, die einen neuen Cluster unterstützen können. Versuchen Sie, Ihren Cluster mit mindestens zwei Subnetzen erneut zu erstellen, die sich in den unterstützten Availability Zones für Ihr Konto befinden. Weitere Informationen finden Sie unter Unzureichende Kapazität.

Optionale Einstellungen

Im Folgenden finden Sie optionale Einstellungen, die bei Bedarf dem vorherigen Befehl hinzugefügt werden müssen. Sie können diese Optionen nur aktivieren, wenn Sie den Cluster erstellen, nicht danach.

- Wenn Sie angeben möchten welchem `IPv4` CIDR-Block (Classless Inter-Domain Routing) Kubernetes Service-IP-Adressen zuweist, geben Sie die Option **`--kubernetes-network-config serviceIpv4Cidr=`*`CIDR block`* für den folgenden Befehl an.

  Die Angabe eines eigenen Bereichs kann helfen, Konflikte zwischen Kubernetes-Services und anderen Netzwerken zu vermeiden, die mit Ihrer VPC verbunden sind. Geben Sie einen Bereich in CIDR-Notation ein. Beispiel: `10.2.0.0/16`.

  Der CIDR-Block muss die folgenden Anforderungen erfüllen:

  - Verwendet einen der folgenden Bereiche: `10.0.0.0/8`, `172.16.0.0/12`, oder `192.168.0.0/16`.
  - Hat eine Mindestgröße von `/24` und eine maximale Größe von `/12`.
  - Überschneidet sich nicht mit dem Bereich der VPC für Ihre Amazon-EKS-Ressourcen.

  Sie können diese Option nur angeben, wenn Sie die `IPv4`-Adressfamilie verwenden, und nur bei Erstellen des Clusters. Wenn Sie dies nicht angeben, weist Kubernetes Service-IP-Adressen aus dem CIDR-Block `10.100.0.0/16` oder `172.20.0.0/16` zu.

- Wenn Sie einen Cluster und der Cluster Pods und Services IPv6-Adressen statt IPv4-Adressen zuweisen soll, fügen Sie dem folgenden Befehl **--kubernetes-network-config ipFamily=ipv6** hinzu.

  Standardmäßig weist Kubernetes Ihren Pods und Services IPv4-Adressen zu. Bevor Sie sich entscheiden, die IPv6-Familie zu verwenden, stellen Sie sicher, dass Sie mit allen Überlegungen und Anforderungen in den Themen the section called “VPC-Anforderungen und -Überlegungen”, the section called “Subnetz-Anforderungen und -Überlegungen”, the section called “Anforderungen an Sicherheitsgruppen”, und the section called “IPv6” vertraut sind. Wenn Sie die IPv6-Familie verwenden, können Sie keinen Adressbereich für Kubernetes angeben, um IPv6-Serviceadressen zuzuweisen, wie Sie dies für die IPv4-Familie können. Kubernetes weist Serviceadressen aus dem eindeutigen lokalen Adressbereich zu (`fc00::/7`).

2. Die Bereitstellung des Clusters dauert mehrere Minuten. Sie können den Status Ihres Clusters mit dem folgenden Befehl überprüfen.

   ```
   aws eks describe-cluster --region region-code --name my-cluster --query
    "cluster.status"
   ```

   Fahren Sie nicht mit dem nächsten Schritt fort, bis die zurückgegebene Ausgabe `ACTIVE` ist.

3. Wenn Sie Ihren Cluster mit `eksctl` erstellt haben, können Sie diesen Schritt überspringen. Das liegt daran, dass `eksctl` diesen Schritt bereits für Sie durchgeführt hat. Aktivieren Sie `kubectl`, um mit Ihrem Cluster zu kommunizieren, indem Sie einen neuen Kontext zur Datei `kubectl config` hinzufügen. Weitere Informationen zum Erstellen und Aktualisieren der Datei finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

   ```
   aws eks update-kubeconfig --region region-code --name my-cluster
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   Added new context arn:aws:eks:region-code:111122223333:cluster/my-cluster to /home/
   username/.kube/config
   ```

4. Bestätigen Sie die Kommunikation mit Ihrem Cluster, indem Sie den folgenden Befehl ausführen.

```
kubectl get svc
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   28h
```

5. (Empfohlen) Um einige Amazon EKS-Add-Ons zu verwenden oder einzelnen Kubernetes Workloads bestimmte AWS Identity and Access Management (IAM-) Berechtigungen zuzuweisen, erstellen Sie einen IAM OpenID Connect (OIDC) -Anbieter für Ihren Cluster. Sie müssen einen IAM-OIDC-Anbieter für Ihren Cluster nur einmal erstellen. Weitere Informationen zu Amazon EKS-Add-ons finden Sie unter Amazon-EKS-Add-ons. Weitere Informationen zum Zuweisen bestimmter IAM-Berechtigungen zu Ihren Workloads finden Sie unter IAM-Rollen für Servicekonten.
6. (Empfohlen) Konfigurieren Sie Ihren Cluster für das Amazon VPC CNI plugin for Kubernetes-Plugin, bevor Sie Amazon-EC2-Knoten in Ihrem Cluster bereitstellen. Standardmäßig wurde das Plugin mit Ihrem Cluster installiert. Wenn Sie Ihrem Cluster Amazon-EC2-Knoten hinzufügen, wird das Plugin automatisch für jeden Amazon-EC2-Knoten bereitgestellt, den Sie hinzufügen. Für das Plugin müssen Sie eine der folgenden IAM-Richtlinien an eine IAM-Rolle anhängen:

   `AmazonEKS_CNI_Policy`-verwaltete IAM-Richtlinie

   Wenn Ihr Cluster die `IPv4`-Familie verwendet

   Eine IAM-Richtlinie, die Sie erstellen

   Wenn Ihr Cluster die `IPv6`-Familie verwendet

   Die IAM-Rolle, an die Sie die Richtlinie anhängen, kann die IAM-Rolle des Knotens oder eine dedizierte Rolle sein, die nur für das Plugin verwendet wird. Wir empfehlen, die Richtlinie an diese Rolle anzuhängen. Weitere Informationen zum manuellen Erstellen einer Rolle finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA) oder Amazon-EKS-Knoten-IAM-Rolle.
7. Wenn Sie Ihren Cluster mit dem bereitgestellt haben AWS Management Console, können Sie diesen Schritt überspringen. Die AWS Management Console stellt standardmäßig die Amazon-EKS-Add-ons Amazon VPC CNI plugin for Kubernetes, CoreDNS und `kube-proxy` bereit.

Wenn Sie Ihren Cluster entweder mit `eksctl` oder AWS CLI bereitstellen, werden die selbstverwalteten Add-Ons Amazon VPC CNI plugin for Kubernetes, CoreDNS und `kube-proxy` bereitgestellt. Sie können die selbstverwalteten Add-ons Amazon VPC CNI plugin for Kubernetes, CoreDNS und `kube-proxy`, die mit Ihrem Cluster bereitgestellt werden, zu Amazon-EKS-Add-ons migrieren. Weitere Informationen finden Sie unter Amazon-EKS-Add-ons.

8. (Optional) Falls noch nicht geschehen, können Sie Prometheus-Metriken für Ihren Cluster aktivieren. Weitere Informationen finden Sie unter Erstellen eine Scrapers im Benutzerhandbuch von Amazon Managed Service für Prometheus.
9. Wenn Sie Prometheus-Metriken aktiviert haben, müssen Sie Ihre `aws-auth-ConfigMap` so einrichten, dass der Scraper Cluster-interne Berechtigungen erhält. Weitere Informationen finden Sie unter Konfigurieren Ihres Amazon-EKS-Clusters im Benutzerhandbuch von Amazon Managed Service für Prometheus.
10. Wenn Sie planen, Workloads in Ihrem Cluster bereitzustellen, die Amazon-EBS-Volumes verwenden, und Sie einen Cluster der Version `1.23` oder höher erstellt haben, dann müssen Sie den Amazon-EBS-CSI-Treiber in Ihrem Cluster installieren, bevor Sie die Workloads bereitstellen.

Empfohlene nächste Schritte:

- Der IAM-Prinzipal, der den Cluster erstellt hat, ist der einzige Prinzipal, der Zugriff auf den Cluster hat. Erteilen Sie Berechtigungen für andere IAM-Prinzipalen, damit sie auf Ihren Cluster zugreifen können.
- Wenn der IAM-Prinzipal, der den Cluster erstellt hat, nur über die Mindest-IAM-Berechtigungen verfügt, auf die in den Voraussetzungen verwiesen wird, dann möchten Sie vielleicht weitere Amazon-EKS-Berechtigungen für diesen Prinzipal hinzufügen. Weitere Informationen über das Erteilen von Amazon-EKS-Berechtigungen für IAM-Prinzipals finden Sie unter Identitäts- und Zugriffsverwaltung für Amazon EKS.
- Wenn Sie möchten, dass der IAM-Prinzipal, der den Cluster erstellt hat, oder andere Prinzipalen Kubernetes-Ressourcen in der Amazon EKS-Konsole sehen können, gewähren Sie den Entitäten die Erforderliche Berechtigungen.
- Wenn Sie möchten, dass Knoten und IAM-Prinzipalen von Ihrer VPC aus auf Ihren Cluster zugreifen können, aktivieren Sie den privaten Endpunkt für Ihren Cluster. Standardmäßig ist der öffentliche Endpunkt aktiviert. Falls gewünscht, können Sie den öffentlichen Endpunkt deaktivieren, nachdem Sie den privaten Endpunkt aktiviert haben. Weitere Informationen finden Sie unter Zugriffskontrolle für den Amazon-EKS-Cluster-Endpunkt.

- Aktivieren Sie die Secrets-Verschlüsselung für Ihren Cluster.
- Konfigurieren Sie die Protokollierung für Ihren Cluster.
- Fügen Sie Ihrem Cluster Knoten hinzu.

Cluster-Erkenntnisse

Erkenntnisse zu Amazon-EKS-Clustern stellen Empfehlungen bereit, die Ihnen dabei helfen, Best Practices für Amazon EKS und Kubernetes umzusetzen. Für jeden Amazon-EKS-Cluster werden automatisch wiederkehrende Überprüfungen anhand einer von Amazon EKS zusammengestellten Liste von Erkenntnissen durchgeführt. Diese Erkenntnisüberprüfungen werden vollständig von Amazon EKS verwaltet und bieten Empfehlungen zum Umgang mit etwaigen Ergebnissen.

Empfohlene Verwendung von Cluster Insights:

- Bevor Sie Ihre Kubernetes Cluster-Version aktualisieren, überprüfen Sie Cluster Insights in der EKS-Konsole.
- Wenn in Ihrem Cluster Probleme festgestellt wurden, überprüfen Sie diese und nehmen Sie entsprechende Korrekturen vor. Zu den Problemen gehören Links zu Amazon EKS undKubernetes.
- Warten Sie nach der Behebung der Probleme, bis die Cluster-Einblicke aktualisiert sind. Wenn alle Probleme behoben wurden, aktualisieren Sie Ihren Cluster.

Derzeit gibt Amazon EKS nur Erkenntnisse im Zusammenhang mit der Bereitschaft für Kubernetes-Versions-Upgrades zurück.

Upgrade-Erkenntnisse identifizieren mögliche Probleme, die sich ggf. auf Kubernetes-Cluster-Upgrades auswirken. Dadurch müssen Administratoren weniger Zeit für die Vorbereitung von Upgrades aufwenden und die Zuverlässigkeit von Anwendungen unter neueren Kubernetes-Versionen erhöht sich. Cluster werden von Amazon EKS automatisch anhand einer Liste möglicher Probleme überprüft, die Kubernetes-Versions-Upgrades beeinträchtigen können. Amazon EKS aktualisiert die Liste der Erkenntnisüberprüfungen regelmäßig auf der Grundlage der Änderungen, die in den einzelnen veröffentlichten Kubernetes-Versionen vorgenommen wurden.

Upgrade-Erkenntnisse von Amazon EKS beschleunigen den Test- und Überprüfungsprozess für neue Versionen. Sie ermöglichen es Clusteradministratoren und Anwendungsentwicklern außerdem, die neuesten Kubernetes-Funktionen zu nutzen, indem sie auf mögliche Probleme hinweisen und Tipps zur Problembehandlung geben. Wenn Sie die Liste der durchgeführten

Erkenntnisüberprüfungen sowie die relevanten Probleme anzeigen möchten, die Amazon EKS identifiziert hat, können Sie den `ListInsights`-API-Vorgang von Amazon EKS aufrufen oder in der Amazon-EKS-Konsole nachsehen.

Cluster-Einblicke werden regelmäßig aktualisiert. Sie können Cluster-Insights nicht manuell aktualisieren. Wenn Sie ein Cluster-Problem beheben, dauert es einige Zeit, bis die Cluster-Erkenntnisse aktualisiert sind. Um festzustellen, ob ein Fix erfolgreich war, vergleichen Sie den Zeitpunkt, zu dem die Änderung bereitgestellt wurde, mit dem Zeitpunkt der letzten Aktualisierung von Cluster Insight.

Cluster-Einblicke anzeigen (Konsole)

So zeigen Sie die Erkenntnisse eines Amazon EKS-Clusters an:

a. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

b. Wählen Sie in der Clusterliste den Namen des Amazon-EKS-Clusters aus, für den Sie die Erkenntnisse anzeigen möchten.

c. Wählen Sie die Registerkarte Upgrade-Erkenntnisse aus.

d. Die Seite Upgrade-Erkenntnisse enthält folgende Felder:

- Name: Die Überprüfung, die von Amazon EKS für den Cluster durchgeführt wurde.
- Erkenntnisstatus: Eine Erkenntnis mit dem Status „Fehler“ bedeutet in der Regel, dass die betroffene Kubernetes-Version die nächsthöhere Version der aktuellen Cluster-Version ist. Der Status „Warnung“ bedeutet dagegen, dass die Erkenntnis für eine zukünftige Kubernetes-Version (N+2 oder mehr) gilt. Eine Erkenntnis mit dem Status „Bestanden“ bedeutet, dass Amazon EKS keine Probleme im Zusammenhang mit dieser Erkenntnisüberprüfung in Ihrem Cluster festgestellt hat. Bei einer Erkenntnis mit dem Status „Unbekannt“ kann Amazon EKS nicht feststellen, ob Ihr Cluster von dieser Erkenntnisüberprüfung betroffen ist.
- Version: Die Kubernetes-Version, die auf mögliche Probleme im Zusammenhang mit der Erkenntnis überprüft wurde.
- Zeit der letzten Aktualisierung (UTC-5:00): Der Zeitpunkt, zu dem der Status der Erkenntnis für diesen Cluster zuletzt aktualisiert wurde.
- Letzte Übergangszeit (UTC-5:00): Der Zeitpunkt, zu dem sich der Status dieser Erkenntnis zuletzt geändert hat.
- Beschreibung: Informationen der letzten Erkenntnisüberprüfung (einschließlich der Warnung und der empfohlenen Behandlungsmaßnahmen).

Cluster-Einblicke anzeigen (AWS CLI)

So zeigen Sie die Erkenntnisse eines Amazon EKS-Clusters an:

a. Bestimmen Sie, welchen Cluster Sie auf Erkenntnisse überprüfen möchten. Der folgende Befehl listet die Erkenntnisse für einen angegebenen Cluster auf. Nehmen Sie nach Bedarf die folgenden Änderungen am Befehl vor und führen Sie anschließend den geänderten Befehl aus:
   - Ersetzen Sie *region-code* durch den Code für Ihre AWS-Region.
   - Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.

   ```
   aws eks list-insights --region region-code --cluster-name my-cluster
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   {
       "insights": [
           {
               "category": "UPGRADE_READINESS",
               "name": "Deprecated APIs removed in Kubernetes v1.29",
               "insightStatus": {
                   "status": "PASSING",
                   "reason": "No deprecated API usage detected within the last 30 days."
               },
               "kubernetesVersion": "1.29",
               "lastTransitionTime": 1698774710.0,
               "lastRefreshTime": 1700157422.0,
               "id": "123e4567-e89b-42d3-a456-579642341238",
               "description": "Checks for usage of deprecated APIs that are scheduled
    for removal in Kubernetes v1.29. Upgrading your cluster before migrating to the
    updated APIs supported by v1.29 could cause application impact."
           }
       ]
   }
   ```

b. Führen Sie den folgenden Befehl aus, um beschreibende Informationen zu der Erkenntnis zu erhalten. Nehmen Sie nach Bedarf die folgenden Änderungen am Befehl vor und führen Sie anschließend den geänderten Befehl aus:
   - Ersetzen Sie *region-code* durch den Code für Ihre AWS-Region.
   - Ersetzen Sie *123e4567-e89b-42d3-a456-579642341238* durch die Erkenntnis-ID aus der Auflistung der Cluster-Erkenntnisse.

- Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.

```
aws eks describe-insight --region region-code --id 123e4567-e89b-42d3-
a456-579642341238 --cluster-name my-cluster
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "insight": {
        "category": "UPGRADE_READINESS",
        "additionalInfo": {
            "EKS update cluster documentation": "https://docs.aws.amazon.com/eks/
latest/userguide/update-cluster.html",
            "Kubernetes v1.29 deprecation guide": "https://kubernetes.io/docs/
reference/using-api/deprecation-guide/#v1-29"
        },
        "name": "Deprecated APIs removed in Kubernetes v1.29",
        "insightStatus": {
            "status": "PASSING",
            "reason": "No deprecated API usage detected within the last 30 days."
        },
        "kubernetesVersion": "1.29",
        "recommendation": "Update manifests and API clients to use newer Kubernetes
 APIs if applicable before upgrading to Kubernetes v1.29.",
        "lastTransitionTime": 1698774710.0,
        "lastRefreshTime": 1700157422.0,
        "categorySpecificSummary": {
            "deprecationDetails": [
                {
                    "usage": "/apis/flowcontrol.apiserver.k8s.io/v1beta2/
flowschemas",
                    "replacedWith": "/apis/flowcontrol.apiserver.k8s.io/v1beta3/
flowschemas",
                    "stopServingVersion": "1.29",
                    "clientStats": [],
                    "startServingReplacementVersion": "1.26"
                },
                {
                    "usage": "/apis/flowcontrol.apiserver.k8s.io/v1beta2/
prioritylevelconfigurations",
                    "replacedWith": "/apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations",
```

```
                    "stopServingVersion": "1.29",
                    "clientStats": [],
                    "startServingReplacementVersion": "1.26"
                }
            ]
        },
        "id": "f6a11fe4-77f7-48c6-8326-9a13f022ecb3",
        "resources": [],
        "description": "Checks for usage of deprecated APIs that are scheduled for
 removal in Kubernetes v1.29. Upgrading your cluster before migrating to the updated
 APIs supported by v1.29 could cause application impact."
    }
}
```

Aktualisieren einer Amazon-EKS-Cluster-Kubernetes-Version

Wenn eine neue Kubernetes-Version in Amazon EKS verfügbar ist, können Sie Ihren Amazon-EKS-Cluster auf die neueste Version aktualisieren.

> **Important**
>
> Sobald Sie einen Cluster aktualisiert haben, können Sie kein Downgrade auf eine frühere Version durchführen. Es wird empfohlen, dass Sie vor dem Aktualisieren auf eine neue Kubernetes-Version die Informationen in Kubernetes-Versionen für Amazon EKS und in den Aktualisierungsschritten in diesem Thema sichten.

Neue Kubernetes-Versionen führen oft bedeutende Änderungen ein. Daher empfehlen wir Ihnen, das Verhalten Ihrer Anwendungen mit einer neuen Kubernetes-Version zu testen, bevor Sie Ihre Produktionscluster aktualisieren. Hierzu erstellen Sie einen kontinuierlichen Integrations-Workflow, um das Verhalten Ihrer Anwendungen zu testen, bevor Sie auf eine neue Kubernetes-Version aktualisieren.

Der Aktualisierungsprozess besteht darin, dass Amazon EKS neue API-Serverknoten mit der aktualisierten Kubernetes-Version startet, um die vorhandenen zu ersetzen. Amazon EKS führt auf diesen neuen Knoten standardmäßige Infrastruktur- und Bereitschaftszustandsprüfungen für den Netzwerkverkehr durch, um sicherzustellen, dass sie wie erwartet funktionieren. Sobald Sie das Cluster-Upgrade gestartet haben, können Sie es jedoch weder anhalten noch beenden. Wenn eine dieser Prüfungen fehlschlägt, macht Amazon EKS die Infrastruktur-Bereitstellung rückgängig

und der Cluster verbleibt in der vorherigen Kubernetes-Version. Laufende Anwendungen sind davon nicht betroffen und Ihr Cluster befindet sich nie in einem nicht deterministischen oder nicht wiederherstellbaren Zustand. Amazon EKS sichert regelmäßig alle verwalteten Cluster, und es gibt Mechanismen, um Cluster bei Bedarf wiederherzustellen. Wir evaluieren und verbessern unsere Verwaltungsprozesse für die Kubernetes-Infrastruktur laufend.

Um den Cluster zu aktualisieren, benötigt Amazon EKS bis zu fünf verfügbare IP-Adressen aus den Subnetzen, die beim Erstellen des Clusters bereitgestellt wurden. Amazon EKS erstellt in jedem der von Ihnen angegebenen Subnetze neue Elastic-Network-Schnittstellen für den Cluster (Netzwerkschnittstellen). Die Netzwerkschnittstellen können in anderen Subnetzen erstellt werden als Ihre vorhandenen Netzwerkschnittstellen. Stellen Sie daher sicher, dass Ihre Sicherheitsgruppenregeln die erforderliche Cluster-Kommunikation für jedes der Subnetze zulassen, die Sie bei der Erstellung Ihres Clusters angegeben haben. Wenn eines der Subnetze, die Sie bei der Erstellung des Clusters angegeben haben, nicht existiert, nicht genügend verfügbare IP-Adressen hat oder nicht über Sicherheitsgruppenregeln verfügt, die die notwendige Clusterkommunikation zulassen, kann die Aktualisierung fehlschlagen.

> **Note**
>
> Um sicherzustellen, dass der API-Serverendpunkt für Ihren Cluster immer zugänglich ist, bietet Amazon EKS eine hochverfügbare Kubernetes-Steuerebene und führt während Aktualisierungsvorgängen fortlaufende Aktualisierungen der API-Serverinstances durch. Um den sich ändernden IP-Adressen von API-Serverinstances Rechnung zu tragen, die Ihren Kubernetes-API-Server-Endpunkt unterstützen, müssen Sie sicherstellen, dass Ihre API-Serverclients Wiederverbindungen effektiv verwalten. Aktuelle Versionen von `kubectl` und der Kubernetes-Client-Bibliotheken, die offiziell unterstützt werden, führen diesen Vorgang zur Wiederherstellung der Verbindung transparent durch.

Die Kubernetes-Version für Ihren Amazon-EKS-Cluster aktualisieren

Aktualisieren der Kubernetes-Version für Ihren Cluster

1. Vergleichen Sie die Kubernetes-Version Ihrer Cluster-Steuerebene mit der Kubernetes-Version Ihrer Knoten.

   - Rufen Sie die Kubernetes-Version Ihrer Cluster-Steuerebene ab.

```
kubectl version
```

- Holen Sie sich die Kubernetes-Version Ihrer Knoten. Dieser Befehl gibt alle selbstverwalteten und verwalteten Amazon-EC2- und Fargate-Knoten zurück. Jeder Fargate-Pod wird als eigener Knoten aufgeführt.

```
kubectl get nodes
```

Bevor Sie eine Steuerebene auf eine neue Kubernetes-Version aktualisieren, muss die Kubernetes-Nebenversion der verwalteten und Fargate-Knoten in Ihrem Cluster mit der Version der aktuellen Version Ihrer Steuerebene übereinstimmen. Wenn auf Ihrer Kontrollebene beispielsweise Version ausgeführt wird `1.29` und auf einem Ihrer Knoten Version ausgeführt wird`1.28`, müssen Sie Ihre Knoten auf Version aktualisieren, `1.29` bevor Sie Ihre Kontrollebene auf 1.30 aktualisieren. Wir empfehlen außerdem, dass Sie Ihre selbstverwalteten Knoten auf dieselbe Version wie Ihre Steuerebene aktualisieren, bevor Sie die Steuerebene aktualisieren. Weitere Informationen finden Sie unter Aktualisieren einer verwalteten Knotengruppe und Aktualisierungen des selbstverwalteten Worker-Knotens. Wenn Sie Fargate-Knoten mit einer Nebenversion haben, die niedriger als die Version der Steuerebene ist, löschen Sie zuerst den Pod, der durch den Knoten dargestellt wird. Aktualisieren Sie dann Ihre Steuerebene. Alle verbleibenden Pods werden auf die neue Version aktualisiert, nachdem Sie sie neu bereitgestellt haben.

2. Wenn die Kubernetes-Version, mit der Sie Ihren Cluster ursprünglich eingerichtet haben, Kubernetes `1.25` oder eine neuere war, überspringen Sie diesen Schritt.

   Der Zugangscontroller für die Pod-Sicherheitsrichtlinie ist auf Amazon-EKS-Clustern standardmäßig aktiviert. Stellen Sie vor der Aktualisierung Ihres Clusters sicher, dass die richtigen Pod-Sicherheitsrichtlinien vorhanden sind. Dies dient dazu, potenzielle Sicherheitsprobleme zu vermeiden. Sie können die Standardrichtlinie mit dem Befehl **`kubectl get psp eks.privileged`** überprüfen.

   ```
   kubectl get psp eks.privileged
   ```

   Wenn die folgende Fehlermeldung angezeigt wird, lesen Sie Amazon-EKS-Pod-Standardsicherheitsrichtlinie, bevor Sie fortfahren.

```
Error from server (NotFound): podsecuritypolicies.extensions "eks.privileged" not
 found
```

3. Wenn die Kubernetes-Version, mit der Sie Ihren Cluster ursprünglich eingerichtet haben, Kubernetes `1.18` oder eine neuere war, überspringen Sie diesen Schritt.

   Möglicherweise müssen Sie einen eingestellten Begriff aus Ihrem CoreDNS-Manifest entfernen.

   a. Überprüfen Sie, ob Ihr CoreDNS-Manifest eine Zeile enthält, die nur das Wort `upstream` enthält.

   ```
   kubectl get configmap coredns -n kube-system -o jsonpath='{$.data.Corefile}' |
    grep upstream
   ```

   Wenn keine Ausgabe zurückgegeben wird, bedeutet dies, dass Ihr Manifest die Zeile nicht enthält. In diesem Fall überspringen Sie den nächsten Schritt. Wenn das Wort `upstream` zurückgegeben wird, müssen Sie die Zeile entfernen.

   b. Entfernen Sie in der configmap-Datei die Zeile am oberen Rand der Datei, die nur das Wort `upstream` enthält. Ändern Sie sonst nichts in der Datei. Nachdem die Zeile entfernt wurde, speichern Sie die Änderungen.

   ```
   kubectl edit configmap coredns -n kube-system -o yaml
   ```

4. Aktualisieren Sie Ihren Cluster mit `eksctl` AWS Management Console, dem AWS CLI oder.

> **Important**
>
> - Wenn Sie auf Version `1.23` aktualisieren und in Ihrem Cluster Amazon-EBS-Volumes verwenden, müssen Sie zur Vermeidung von Workload-Unterbrechungen den CSI-Treiber von Amazon EBS im Cluster installieren, bevor Sie den Cluster auf die Version `1.23` aktualisieren. Weitere Informationen finden Sie unter Kubernetes1.23 und Amazon-EBS-CSI-Treiber.
> - Die Kubernetes-Versionen `1.24` und höher nutzen `containerd` als standardmäßige Container-Laufzeit. Wenn Sie zur Laufzeit `containerd` wechseln und Fluentd bereits für Container Insights konfiguriert ist, müssen Sie vor der Aktualisierung des Clusters Fluentd zu Fluent Bit migrieren. Die Fluentd-Parser sind so konfiguriert, dass sie nur Protokollnachrichten im JSON-Format analysieren. Im Gegensatz zu `dockerd`

enthält die Container-Laufzeit `containerd` Protokollnachrichten, die sich nicht im JSON-Format befinden. Wenn Sie nicht zu Fluent Bit migrieren, erzeugen einige der konfigurierten Fluentd's-Parser eine große Anzahl von Fehlern im Container Fluentd. Weitere Informationen zur Migration finden Sie unter Fluent BitSo einrichten, dass Protokolle DaemonSet an Logs gesendet werden CloudWatch .

- Da Amazon EKS eine hoch verfügbare Steuerebene ausführt, dürfen Sie jeweils nur um eine Unterversion aktualisieren. Weitere Informationen zu dieser Anforderung finden Sie unter Kubernetes-Version und Version-Skew-Supportrichtlinie. Angenommen, Ihre aktuelle Cluster-Version ist Version `1.28` und Sie möchten sie auf Version `1.30` aktualisieren. Sie müssen zuerst Ihren Version `1.28`-Cluster auf Version `1.29` und dann Ihren Version `1.29`-Cluster auf Version `1.30` aktualisieren.
- Überprüfen Sie den Versionsunterschied zwischen Kubernetes `kube-apiserver` und `kubelet` auf Ihren Knoten.
  - Ab der Kubernetes-Version `1.28` kann `kubelet` bis zu drei Nebenversionen älter sein als `kube-apiserver`. Weitere Informationen finden Sie in der Richtlinie zum Unterschied bei der Kubernetes-Upstream-Version.
  - Wenn das `kubelet` Ihrer verwalteten Knoten und Ihrer Fargate-Knoten mindestens über die Kubernetes-Version `1.25` verfügt, können Sie Ihren Cluster um bis zu drei Versionen aktualisieren, ohne die `kubelet`-Version zu aktualisieren. Wenn das `kubelet` also beispielsweise über die Version `1.25` verfügt, können Sie die Version Ihres Amazon EKS-Clusters von `1.25` auf`1.26`, auf `1.27` und auf `1.28` aktualisieren und für `kubelet` die Version `1.25` beibehalten.
  - Wenn das `kubelet` Ihrer verwalteten Knoten und Ihrer Fargate-Knoten über die Kubernetes-Version `1.24` oder über eine ältere Version verfügt, darf es maximal zwei Nebenversionen älter sein als `kube-apiserver`. Anders ausgedrückt: Wenn das `kubelet` über die Version `1.24` oder über eine ältere Version verfügt, können Sie Ihren Cluster nur um bis zu Versionen aktualisieren. Wenn das `kubelet` also beispielsweise über die Version `1.21` verfügt, können Sie die Version Ihres Amazon EKS-Clusters von `1.21` auf`1.22` und auf `1.23` aktualisieren. Es ist aber nicht möglich, den Cluster auf `1.24` zu aktualisieren, solange `kubelet` über die Version `1.21` verfügt.
- Vergewissern Sie sich daher vor dem Starten eines Updates, dass das `kubelet` auf Ihren Knoten die gleiche Kubernetes-Version hat wie Ihre Steuerebene.
- Wenn Ihr Cluster mit einer Version des Amazon VPC CNI plugin for Kubernetes konfiguriert ist, die älter ist als `1.8.0`, empfehlen wir, dass Sie das Plugin auf die

neueste Version aktualisieren, bevor Sie Ihren Cluster aktualisieren. Informationen zum Aktualisieren des Plugins finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

- Wenn Sie Ihren Cluster auf Version `1.25` oder höher aktualisieren und den AWS Load Balancer Controller in Ihrem Cluster bereitgestellt haben, aktualisieren Sie den Controller auf Version `2.4.7` oder höher, bevor Sie Ihre Cluster-Version auf `1.25` aktualisieren. Weitere Informationen finden Sie in den Kubernetes1,25-Versionshinweisen.

eksctl

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre -Version mit dem folgenden Befehl überprüfen:

```
eksctl version
```

Eine Installations- und Aktualisierungsanleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

Aktualisieren Sie die Kubernetes-Version Ihrer Amazon-EKS-Steuerebene. Ersetzen Sie *`my-cluster`* mit Ihrem Clusternamen. Ersetzen Sie *`1.30`* durch die von Amazon EKS unterstützte Versionsnummer, auf die Sie Ihren Cluster aktualisieren möchten. Eine Liste der unterstützten Versionsnummern finden Sie unter Kubernetes-Versionen für Amazon EKS.

```
eksctl upgrade cluster --name my-cluster --version 1.30 --approve
```

Die Aktualisierung dauert einige Minuten.

AWS Management Console

a. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
b. Wählen Sie den Namen des zu aktualisierenden Amazon-EKS-Clusters aus und klicken Sie dann auf Cluster-Version aktualisieren.
c. Wählen Sie unter Kubernetes version (-Version) die Version aus, auf die Sie den Cluster aktualisieren möchten, und klicken Sie auf Update (Aktualisieren).

d. Geben Sie als Cluster name (Clustername) den Namen Ihres Clusters ein und klicken Sie auf Confirm (Bestätigen).

   Die Aktualisierung dauert einige Minuten.

AWS CLI

a. Aktualisieren Sie Ihren Amazon-EKS-Cluster mit dem AWS CLI -Befehl. Ersetzen Sie das *example values* durch Ihr eigenes. Ersetzen Sie *1.30* durch die von Amazon EKS unterstützte Versionsnummer, auf die Sie Ihren Cluster aktualisieren möchten. Eine Liste der unterstützten Versionsnummern finden Sie unter Kubernetes-Versionen für Amazon EKS.

```
aws eks update-cluster-version --region region-code --name my-cluster --
kubernetes-version 1.30
```

   Eine Beispielausgabe sieht wie folgt aus.

```
{
    "update": {
        "id": "b5f0ba18-9a87-4450-b5a0-825e6e84496f",
        "status": "InProgress",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.30"
            },
            {
                "type": "PlatformVersion",
                "value": "eks.1"
            }
        ],
[...]
        "errors": []
    }
}
```

b. Überwachen Sie den Status Ihres Cluster-Updates mit dem folgenden Befehl. Verwenden Sie den Clusternamen und die Update-ID, die der vorherige Befehl zurückgegeben

hat. Wenn der Status `Successful` angezeigt wird, ist das Update abgeschlossen. Die Aktualisierung dauert einige Minuten.

```
aws eks describe-update --region region-code --name my-cluster --update-
id b5f0ba18-9a87-4450-b5a0-825e6e84496f
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "update": {
        "id": "b5f0ba18-9a87-4450-b5a0-825e6e84496f",
        "status": "Successful",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.30"
            },
            {
                "type": "PlatformVersion",
                "value": "eks.1"
            }
        ],
[...]
        "errors": []
    }
}
```

5. Nachdem Ihre Cluster-Aktualisierung abgeschlossen wurde, aktualisieren Sie Ihre Knoten auf dieselbe Kubernetes-Nebenversion wie Ihr aktualisierter Cluster. Weitere Informationen finden Sie unter Aktualisierungen des selbstverwalteten Worker-Knotens und Aktualisieren einer verwalteten Knotengruppe. Alle neuen Pods, die auf Fargate gestartet werden, verfügen über eine `kubelet`-Version, die Ihrer Cluster-Version entspricht. Bestehende Fargate-Pods werden nicht geändert.
6. (Optional) Wenn Sie den Kubernetes Cluster Autoscaler in Ihrem Cluster bereitgestellt haben, bevor Sie den Cluster aktualisiert haben, aktualisieren Sie den Cluster Autoscaler auf die neueste Version, die der Kubernetes-Haupt- und Nebenversion entspricht, auf die Sie aktualisiert haben.

a. Öffnen Sie die Seite Cluster Autoscaler releases in einem Webbrowser und suchen Sie die neuste Cluster Autoscaler-Version, die der Haupt- und Nebenversion Ihres Kubernetes-Clusters entspricht. Wenn beispielsweise die Kubernetes-Version Ihres Clusters `1.30` lautet, suchen Sie die neueste Cluster Autoscaler-Version, die mit `1.30` beginnt. Notieren Sie die semantische Versionsnummer (z. B. `1.30.n`) für diese Version, um sie im nächsten Schritt zu verwenden.

b. Legen Sie das Cluster Autoscaler-Abbild-Tag mit dem folgenden Befehl auf die Version fest, die Sie im vorherigen Schritt notiert haben. Ersetzen Sie ggf. *`1.30.n`* durch Ihren eigenen Wert.

```
kubectl -n kube-system set image deployment.apps/cluster-autoscaler cluster-autoscaler=registry.k8s.io/autoscaling/cluster-autoscaler:v1.30.n
```

7. (Nur Cluster mit GPU-Knoten) Wenn Ihr Cluster über Knoten-Gruppen mit GPU-Unterstützung (z. B. `p3.2xlarge`) verfügt, müssen Sie das NVIDIA-Geräte-Plugin für Kubernetes DaemonSet auf Ihrem Cluster aktualisieren. Ersetzen Sie *`vX.X.X`* mit Ihrer gewünschten Nvidia/K8S-Geräte-Plugin-Version, bevor Sie den folgenden Befehl ausführen.

```
kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
```

8. Aktualisieren Sie die Amazon VPC CNI plugin for Kubernetes, CoreDNS und `kube-proxy` Add-ons. Wir empfehlen, die Add-Ons auf die Mindestversionen zu aktualisieren, die unter Servicekonto-Tokens aufgeführt sind.

   - Wenn Sie Amazon-EKS-Add-Ons verwenden, wählen Sie Clusters (Cluster) in der Amazon-EKS-Konsole und dann im linken Navigationsbereich den Namen des Clusters aus, den Sie aktualisiert haben. In der Konsole werden Benachrichtigungen angezeigt. Diese informieren Sie darüber, dass für jedes Add-On, für das eine Aktualisierung verfügbar ist, eine neue Version verfügbar ist. Um ein Add-on zu aktualisieren, wählen Sie die Registerkarte Add-ons aus. Wählen Sie in einem der Felder für ein Add-on, für das eine Aktualisierung verfügbar ist, Jetzt aktualisieren aus, wählen Sie eine verfügbare Version aus, und wählen Sie dann Aktualisieren aus.
   - Alternativ können Sie das AWS CLI oder verwenden, um Add-Ons `eksctl` zu aktualisieren. Weitere Informationen finden Sie unter Aktualisieren eines Add-Ons.

9. Sofern erforderlich, aktualisieren Sie Ihre Version von `kubectl`. Sie müssen eine `kubectl`-Version verwenden, die nur in der Minor-Version von Ihrer Amazon–EKS-Cluster-Steuerebene

abweicht. Beispielsweise sollte ein `1.29`-`kubectl`-Client mit Kubernetes-, `1.28`-, `1.29`- und `1.30`-Clustern funktionieren. Sie können Ihre aktuell installierte Version mit dem folgenden Befehl überprüfen.

```
kubectl version --client
```

Löschen eines Amazon-EKS-Clusters

Wenn Sie einen Amazon-EKS-Cluster nicht mehr brauchen, sollten Sie die damit verbundenen Ressourcen löschen, um unnötige Kosten zu vermeiden.

Informationen zum Entfernen eines verbundenen Clusters finden Sie unter Einen Cluster abmelden.

Important

- Wenn noch Services in Ihrem Cluster aktiv sind, die einem Load Balancer zugeordnet sind, müssen Sie diese Services löschen, bevor Sie den Cluster löschen können, damit die Load Balancer korrekt gelöscht werden können. Andernfalls bleiben in Ihrer VPC eventuell verwaiste Ressourcen zurück, die verhindern, dass Sie die VPC löschen können.
- Wenn Sie eine Fehlermeldung erhalten, weil der Cluster-Ersteller entfernt wurde, lesen Sie diesen Artikel.
- Die Ressourcen von Amazon Managed Service for Prometheus befinden sich außerhalb des Cluster-Lebenszyklus und müssen unabhängig vom Cluster verwaltet werden. Wenn Sie Ihren Cluster löschen, stellen Sie sicher, dass Sie auch alle entsprechenden Scraper löschen, um die anfallenden Kosten zu stoppen. Weitere Informationen finden Sie unter Suchen und Löschen von Scrapern im Amazon Managed Service for Prometheus Benutzerhandbuch.

Sie können einen Cluster mit`eksctl`, dem oder dem AWS Management Console löschen. AWS CLI

eksctl

So löschen Sie einen Amazon-EKS-Cluster und -Knoten mit **`eksctl`**

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre -Version mit dem folgenden Befehl überprüfen:

```
eksctl version
```

Eine Installations- und Upgrade-Anleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

1. Listen Sie alle in Ihrem Cluster ausgeführten Services auf.

   ```
   kubectl get svc --all-namespaces
   ```

2. Löschen Sie alle Services, die dem Wert `EXTERNAL-IP` zugeordnet sind. Diese Services werden hinter einem Elastic-Load-Balancing-Load-Balancer ausgeführt und müssen in Kubernetes gelöscht werden, damit der Load Balancer und die zugeordneten Ressourcen korrekt freigegeben werden können.

   ```
   kubectl delete svc service-name
   ```

3. Löschen Sie den Cluster und die zugehörigen Knoten mit dem folgenden Befehl. Ersetzen Sie dabei *prod* durch den Namen Ihres Clusters.

   ```
   eksctl delete cluster --name prod
   ```

   Ausgabe:

   ```
   [#]  using region region-code
   [#]  deleting EKS cluster "prod"
   [#]  will delete stack "eksctl-prod-nodegroup-standard-nodes"
   [#]  waiting for stack "eksctl-prod-nodegroup-standard-nodes" to get deleted
   [#]  will delete stack "eksctl-prod-cluster"
   [#]  the following EKS cluster resource(s) for "prod" will be deleted: cluster.
    If in doubt, check CloudFormation console
   ```

AWS Management Console

Um einen Amazon EKS-Cluster mit dem AWS Management Console

1. Listen Sie alle in Ihrem Cluster ausgeführten Services auf.

   ```
   kubectl get svc --all-namespaces
   ```

2. Löschen Sie alle Services, die dem Wert `EXTERNAL-IP` zugeordnet sind. Diese Services werden hinter einem Elastic-Load-Balancing-Load-Balancer ausgeführt und müssen in Kubernetes gelöscht werden, damit der Load Balancer und die zugeordneten Ressourcen korrekt freigegeben werden können.

```
kubectl delete svc service-name
```

3. Alle Knotengruppen und Fargate-Profile löschen.

   a. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

   b. Wählen Sie im linken Navigationsbereich Clusters (Cluster) für Amazon EKS aus, und wählen Sie dann in der Registerkartenliste der Cluster den Namen des zu löschenden Clusters aus.

   c. Wählen Sie auf der Registerkarte Compute (Datenverarbeitung) eine zu löschende Knotengruppe aus. Klicken Sie auf Delete (Löschen), geben Sie den Namen der Knotengruppe ein und wählen Sie anschließend Delete (Löschen) aus. Alle Knotengruppen im Cluster löschen.

   > **Note**
   >
   > Die aufgelisteten Knotengruppen sind nur verwaltete Knotengruppen.

   d. Wählen Sie ein zu löschendes Fargate-Profil aus, wählen Sie Delete (Löschen) aus, geben Sie den Namen des Profils ein und wählen Sie dann Delete (Löschen). Alle Fargate-Profile im Cluster löschen.

4. Löschen Sie alle selbstverwalteten AWS CloudFormation Knotenstapel.

   a. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

   b. Wählen Sie den Knoten-Stack aus, den Sie löschen möchten, und wählen Sie Delete (Löschen) aus.

   c. Wählen Sie im Bestätigungsdialogfeld Stack löschen Stack löschen aus. Löschen Sie alle selbstverwalteten Knoten-Stacks im Cluster.

5. Löschen Sie den -Cluster.

a. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

b. Wählen Sie den Cluster aus, der gelöscht werden soll, und klicken Sie auf Delete (Löschen).

c. Klicken Sie auf dem Bestätigungsbildschirm zum Löschen des Clusters auf Delete (Löschen).

6. (Optional) Löschen Sie den AWS CloudFormation VPC-Stack.

a. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

b. Wählen Sie den zu löschenden VPC-Stack und dann Löschen aus.

c. Wählen Sie im Bestätigungsdialogfeld Stack löschen Stack löschen aus.

AWS CLI

Um einen Amazon EKS-Cluster mit dem AWS CLI

1. Listen Sie alle in Ihrem Cluster ausgeführten Services auf.

```
kubectl get svc --all-namespaces
```

2. Löschen Sie alle Services, die dem Wert `EXTERNAL-IP` zugeordnet sind. Diese Services werden hinter einem Elastic-Load-Balancing-Load-Balancer ausgeführt und müssen in Kubernetes gelöscht werden, damit der Load Balancer und die zugeordneten Ressourcen korrekt freigegeben werden können.

```
kubectl delete svc service-name
```

3. Alle Knotengruppen und Fargate-Profile löschen.

a. Listen Sie die Knotengruppen in Ihrem Cluster mit dem folgenden Befehl auf.

```
aws eks list-nodegroups --cluster-name my-cluster
```

> **Note**
>
> Die aufgelisteten Knotengruppen sind nur verwaltete Knotengruppen.

b. Löschen Sie jede Knotengruppe mit folgendem Befehl. Alle Knotengruppen im Cluster löschen.

```
aws eks delete-nodegroup --nodegroup-name my-nodegroup --cluster-name my-cluster
```

c. Listen Sie die Fargate-Profile in Ihrem Cluster mit dem folgenden Befehl auf.

```
aws eks list-fargate-profiles --cluster-name my-cluster
```

d. Löschen Sie jedes Fargate-Profil mit dem folgenden Befehl. Alle Fargate-Profile im Cluster löschen.

```
aws eks delete-fargate-profile --fargate-profile-name my-fargate-profile --cluster-name my-cluster
```

4. Löschen Sie alle selbstverwalteten AWS CloudFormation Knotenstapel.

a. Listen Sie Ihre verfügbaren AWS CloudFormation Stacks mit dem folgenden Befehl auf. Suchen Sie den Vorlagennamen des Knotens in der sich daraus ergebenden Ausgabe.

```
aws cloudformation list-stacks --query "StackSummaries[].StackName"
```

b. Löschen Sie jeden Knoten-Stack mit dem folgenden Befehl und ersetzen Sie dabei *node-stack* durch den Namen Ihres Knoten-Stacks. Löschen Sie alle selbstverwalteten Knoten-Stacks im Cluster.

```
aws cloudformation delete-stack --stack-name node-stack
```

5. Löschen Sie den Cluster mit dem folgenden Befehl und ersetzen Sie dabei *my-cluster* durch den Namen Ihres Clusters.

```
aws eks delete-cluster --name my-cluster
```

6. (Optional) Löschen Sie den AWS CloudFormation VPC-Stack.

a. Listen Sie Ihre verfügbaren AWS CloudFormation Stacks mit dem folgenden Befehl auf. Suchen Sie die VPC-Vorlage in der sich daraus ergebenden Ausgabe.

```
aws cloudformation list-stacks --query "StackSummaries[].StackName"
```

b. Löschen Sie den VPC-Stack mit dem folgenden Befehl und ersetzen Sie dabei *my-vpc-stack* durch den Namen Ihres VPC-Stacks.

```
aws cloudformation delete-stack --stack-name my-vpc-stack
```

Zugriffskontrolle für den Amazon-EKS-Cluster-Endpunkt

Dieses Thema hilft Ihnen, den privaten Zugriff für den Kubernetes-API-Server-Endpunkt Ihres Amazon-EKS-Clusters zu aktivieren und den öffentlichen Zugriff über das Internet einzuschränken oder vollständig zu deaktivieren.

Wenn Sie einen neuen Cluster erstellen, erstellt Amazon EKS einen Endpunkt für den verwalteten Kubernetes-API-Server, über den Sie mit Ihrem Cluster kommunizieren (mit Kubernetes-Verwaltungswerkzeugen wie `kubectl`). Standardmäßig ist dieser API-Serverendpunkt im Internet öffentlich, und der Zugriff auf den API-Server wird durch eine Kombination aus AWS Identity and Access Management (IAM) und systemeigener Kubernetes rollenbasierter Zugriffskontrolle (RBAC) gesichert.

Sie können den privaten Zugriff auf den Kubernetes-API-Server aktivieren, sodass die gesamte Kommunikation zwischen Ihren Knoten und dem API-Server innerhalb Ihrer VPC bleibt. Sie können die IP-Adressen einschränken, die über das Internet auf Ihren API-Server zugreifen können, oder den Internetzugriff auf den API-Server vollständig deaktivieren.

> **Note**
>
> Da dieser Endpunkt für den Kubernetes API-Server und kein herkömmlicher AWS PrivateLink Endpunkt für die Kommunikation mit einer AWS API ist, wird er in der Amazon VPC-Konsole nicht als Endpunkt angezeigt.

Wenn Sie den privaten Endpunktzugriff für Ihren Cluster aktivieren, erstellt Amazon EKS in Ihrem Namen eine privat gehostete Route-53-Zone und ordnet diese der VPC Ihres Clusters zu. Diese

private gehostete Zone wird von Amazon EKS verwaltet und wird nicht in den Route-53-Ressourcen Ihres Kontos angezeigt. Damit die private gehostete Zone Datenverkehr ordnungsgemäß an Ihren API-Server weiterleiten kann, müssen `enableDnsHostnames` und `enableDnsSupport` für Ihre VPC auf `true` gesetzt sein und die DHCP-Optionen für Ihre VPC müssen `AmazonProvidedDNS` in ihrer Domainnamen-Serverliste enthalten. Weitere Informationen finden Sie unter Aktualisieren der DNS-Unterstützung für Ihre VPC im Amazon-VPC-Benutzerhandbuch.

Sie können Ihre Anforderungen für den Zugriff auf Ihre API-Server-Endpunkte definieren, wenn Sie einen neuen Cluster erstellen. Sie können den Zugriff auf die API-Server-Endpunkte für einen Cluster jederzeit aktualisieren.

Ändern des Cluster-Endpunktzugriffs

Verwenden Sie die Verfahren in diesem Abschnitt, um den Endpunktzugriff für einen bestehenden Cluster zu ändern. Die folgende Tabelle zeigt die unterstützten Kombinationen von API-Server-Endpunktzugriffen und das damit verbundene Verhalten.

Zugriffsoptionen für API-Server-Endpunkte

Endpunkt für öffentlichen Zugriff	Endpunkt für privaten Zugriff	Behavior
Enabled	Disabled	• Dies ist das Standardv erhalten für neue Amazon-EKS-Cluster. • Kubernetes-API-Anfragen, die aus der VPC Ihres Clusters stammen (z. B. Knoten zur Steuerung der Steuerebenenkommun ikation), verlassen zwar die VPC, aber nicht das Netzwerk von Amazon. • Ihr Cluster-API-Server ist über das Internet erreichba r. Optional können Sie die CIDR-Blöcke einschrän ken, die auf den öffentlic

Endpunkt für öffentlichen Zugriff	Endpunkt für privaten Zugriff	Behavior
		hen Endpunkt zugreifen können. Wenn Sie den Zugriff auf bestimmte CIDR-Blöcke beschränken, wird empfohlen, auch den privaten Endpunkt zu aktivieren oder sicherzustellen, dass die von Ihnen angegebenen CIDR-Blöcke die Adressen enthalten, von denen Knoten und Fargate-Pods (wenn Sie sie verwenden) auf den öffentlichen Endpunkt zugreifen.
Aktiviert	Aktiviert	• Kubernetes-API-Anfragen innerhalb der VPC Ihres Clusters (z. B. Knoten zur Steuerung der Steuerebenenkommunikation) verwenden den privaten VPC-Endpunkt. • Ihr Cluster-API-Server ist über das Internet erreichbar. Optional können Sie die CIDR-Blöcke einschränken, die auf den öffentlichen Endpunkt zugreifen können.

Endpunkt für öffentlichen Zugriff	Endpunkt für privaten Zugriff	Behavior
Disabled	Enabled	• Der gesamte Datenverkehr zu Ihrem Cluster-API-Server muss aus der VPC des Clusters oder einem verbundenen Netzwerk stammen. • Es gibt keinen öffentlichen Zugriff auf Ihren API-Server aus dem Internet. Alle `kubectl`-Befehle müssen aus der VPC oder einem verbundenen Netzwerk stammen. Die Verbindungsoptionen finden Sie unter Zugriff auf einen privaten API-Server. • Der API-Server-Endpunkt des Clusters wird von öffentlichen DNS-Servern in eine private IP-Adresse von der VPC aufgelöst. In der Vergangenheit konnte der Endpunkt nur innerhalb der VPC aufgelöst werden. Wenn Ihr Endpunkt nicht in eine private IP-Adresse innerhalb der VPC für einen vorhandenen Cluster aufgelöst wird, können Sie: • Den öffentlichen Zugriff aktivieren und ihn dann erneut deaktivieren. Sie

Endpunkt für öffentlichen Zugriff	Endpunkt für privaten Zugriff	Behavior
		müssen dies nur einmal für einen Cluster tun und der Endpunkt wird von diesem Punkt an zu einer privaten IP-Adresse aufgelöst. • Aktualisieren Sie Ihren Cluster.

Sie können den Endpunktzugriff Ihres Cluster-API-Servers mit dem AWS Management Console oder AWS CLIändern.

AWS Management Console

Um den Endpunktzugriff Ihres Cluster-API-Servers zu ändern, verwenden Sie AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters aus, um Ihre Cluster-Informationen anzuzeigen.
3. Wählen Sie auf der Registerkarte Networking (Netzwerk) die Option Update (Aktualisieren) aus.
4. Wählen Sie fürPrivate access (Privater Zugriff) aus, ob der private Zugriff für den Kubernetes-API-Server-Endpunkt Ihres Clusters aktiviert oder deaktiviert werden soll. Wenn Sie den privaten Zugriff aktivieren, verwenden Kubernetes-API-Anfragen aus der VPC Ihres Clusters den privaten VPC-Endpunkt. Sie müssen den privaten Zugriff aktivieren, um den öffentlichen Zugriff zu deaktivieren.
5. Wählen Sie für Private access (Öffentlicher Zugriff) aus, ob der öffentliche Zugriff für den Kubernetes API-Server-Endpunkt Ihres Clusters aktiviert oder deaktiviert werden soll. Wenn Sie den öffentlichen Zugriff deaktivieren, kann der Kubernetes-API-Server Ihres Clusters nur Anfragen aus der VPC des Clusters empfangen.
6. (Optional) Wenn Sie Öffentlicher Zugriff aktiviert haben, können Sie angeben, welche Adressen aus dem Internet mit dem öffentlichen Endpunkt kommunizieren können. Wählen

Sie Erweiterte Einstellungen aus. Geben Sie einen CIDR-Block ein, z. B. *203.0.113.5/32*. Der Block darf keine reservierten Adressen enthalten. Sie können zusätzliche Blöcke eingeben, indem Sie Quelle hinzufügen auswählen. Es gibt eine maximale Anzahl von CIDR-Blöcken, die Sie angeben können. Weitere Informationen finden Sie unter Amazon-EKS-Service-Quotas. Wenn Sie keine Blöcke angeben, empfängt der öffentliche API-Server-Endpunkt Anfragen von allen (`0.0.0.0/0`) IP-Adressen. Wenn Sie den Zugriff auf Ihren öffentlichen Endpunkt mithilfe von CIDR-Blöcken einschränken, wird empfohlen, dass Sie auch den privaten Endpunktzugriff aktivieren, damit Knoten und Fargate-Pods (falls Sie diese verwenden) mit dem Cluster kommunizieren können. Wenn der private Endpunkt nicht aktiviert ist, müssen die CIDR-Quellen des öffentlichen Zugriffs die Ausgangsquellen aus Ihrer VPC enthalten. Wenn Sie beispielsweise einen Knoten in einem privaten Subnetz haben, der über ein NAT-Gateway mit dem Internet kommuniziert, müssen Sie die ausgehende IP-Adresse des NAT-Gateways als Teil eines erlaubten CIDR-Blocks auf Ihrem öffentlichen Endpunkt hinzufügen.

7. Wählen Sie zum Abschluss Update (Aktualisieren) aus.

AWS CLI

So ändern Sie den Zugriff auf Ihren Cluster-API-Server-Endpunkt mit der AWS CLI

Führen Sie die folgenden Schritte mit der AWS CLI Version `1.27.160` oder höher aus. Sie können Ihre aktuelle Version mit `aws --version` überprüfen. Informationen zur Installation oder zum AWS CLI Upgrade von finden Sie unter Installation von AWS CLI.

1. Aktualisieren Sie den Zugriff auf Ihren Cluster-API-Server-Endpunkt mit dem folgenden AWS CLI -Befehl. Verwenden Ihre eigenen Werte für den Clusternamen und den gewünschten Endpunkt. Wenn Sie `endpointPublicAccess=true` festlegen, können Sie (optional) einen einzelnen CIDR-Block oder eine kommagetrennte Liste von CIDR-Blöcken für `publicAccessCidrs` eingeben. Die Blöcke dürfen keine reservierten Adressen enthalten. Wenn Sie CIDR-Blöcke angeben, empfängt der öffentliche API-Server-Endpunkt nur Anforderungen von den aufgelisteten Blöcken. Es gibt eine maximale Anzahl von CIDR-Blöcken, die Sie angeben können. Weitere Informationen finden Sie unter Amazon-EKS-Service-Quotas. Wenn Sie den Zugriff auf Ihren öffentlichen Endpunkt mithilfe von CIDR-Blöcken einschränken, wird empfohlen, dass Sie auch den privaten Endpunktzugriff aktivieren, damit Knoten und Fargate-Pods (falls Sie diese verwenden) mit dem Cluster kommunizieren können. Wenn der private Endpunkt nicht aktiviert ist, müssen die CIDR-Quellen des öffentlichen Zugriffs die Ausgangsquellen aus Ihrer VPC enthalten. Wenn Sie

beispielsweise einen Knoten in einem privaten Subnetz haben, der über ein NAT-Gateway mit dem Internet kommuniziert, müssen Sie die ausgehende IP-Adresse des NAT-Gateways als Teil eines erlaubten CIDR-Blocks auf Ihrem öffentlichen Endpunkt hinzufügen. Wenn Sie keine CIDR-Blöcke angeben, empfängt der öffentliche API-Server-Endpunkt Anfragen von allen (0.0.0.0/0) IP-Adressen.

> **Note**
>
> Der folgende Befehl ermöglicht den privaten Zugriff und den öffentlichen Zugriff von einer einzelnen IP-Adresse für den API-Server-Endpunkt. Ersetzen Sie *203.0.113.5/32* durch einen einzelnen CIDR-Block oder eine kommagetrennte Liste von CIDR-Blöcken, auf die Sie den Netzwerkzugriff beschränken möchten.

```
aws eks update-cluster-config \
    --region region-code \
    --name my-cluster \
    --resources-vpc-config
 endpointPublicAccess=true,publicAccessCidrs="203.0.113.5/32",endpointPrivateAccess=true
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "update": {
        "id": "e6f0905f-a5d4-4a2a-8c49-EXAMPLE00000",
        "status": "InProgress",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "true"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            },
            {
                "type": "publicAccessCidrs",
                "value": "[\203.0.113.5/32\"]"
            }
```

```
        ],
        "createdAt": 1576874258.137,
        "errors": []
    }
}
```

2. Überwachen Sie den Status des aktualisierten Endpunktzugriffs mit dem folgenden Befehl unter Verwendung des Cluster-Namens und der Update-ID, die vom vorherigen Befehl zurückgegeben wurden. Ihre Aktualisierung ist abgeschlossen, wenn als Status `Successful` angezeigt wird.

```
aws eks describe-update \
    --region region-code \
    --name my-cluster \
    --update-id e6f0905f-a5d4-4a2a-8c49-EXAMPLE00000
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "update": {
        "id": "e6f0905f-a5d4-4a2a-8c49-EXAMPLE00000",
        "status": "Successful",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "true"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            },
            {
                "type": "publicAccessCidrs",
                "value": "[\203.0.113.5/32\"]"
            }
        ],
        "createdAt": 1576874258.137,
        "errors": []
    }
}
```

Zugriff auf einen privaten API-Server

Wenn Sie den öffentlichen Zugriff für den Kubernetes-API-Server-Endpunkt Ihres Clusters deaktiviert haben, können Sie nur über Ihre VPC oder ein verbundenes Netzwerk auf den API-Server zugreifen. Hier sind einige Möglichkeiten, um auf den Kubernetes-API-Server-Endpunkt zuzugreifen:

Verbundenes Netzwerk

Verbinden Sie Ihr Netzwerk über ein AWS -Transit-Gateway oder eine andere Konnektivitätsoption mit der VPC und verwenden Sie dann einen Computer im verbundenen Netzwerk. Sie müssen sicherstellen, dass Ihre Amazon-EKS-Steuerebenen-Sicherheitsgruppe Regeln enthält, die den eingehenden Datenverkehr auf Port 443 von Ihrem verbundenen Netzwerk zulassen.

Bastion-Host von Amazon EC2

Sie können eine Amazon-EC2-Instance in einem öffentlichen Subnetz in der VPC Ihres Clusters starten und sich dann über SSH in dieser Instance anmelden, um `kubectl`-Befehle auszuführen. Weitere Informationen finden Sie unter Linux-Bastion-Hosts in AWS. Sie müssen sicherstellen, dass Ihre Amazon-EKS-Steuerebenen-Sicherheitsgruppe Regeln enthält, die den eingehenden Datenverkehr auf Port 443 von Ihrem Bastion-Host zulassen. Weitere Informationen finden Sie unter Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe.

Wenn Sie `kubectl` für Ihren Bastion-Host konfigurieren, stellen Sie sicher, dass Sie AWS -Anmeldeinformationen verwenden, die der RBAC-Konfiguration Ihres Clusters bereits zugeordnet sind. Fügen Sie alternativ zur RBAC-Konfiguration den IAM-Prinzipal hinzu, die Ihre Bastion verwenden wird (bevor Sie den öffentlichen Zugriff auf den Endpunkt entfernen). Weitere Informationen finden Sie unter the section called “Gewähren Sie Zugriff auf Kubernetes-APIs” und Nicht autorisiert oder Zugriff verweigert (`kubectl`).

AWS Cloud9 IDE

AWS Cloud9 ist eine cloudbasierte integrierte Entwicklungsumgebung (IDE), mit der Sie Ihren Code mit nur einem Browser schreiben, ausführen und debuggen können. Sie können eine AWS Cloud9 IDE in der VPC Ihres Clusters erstellen und die IDE für die Kommunikation mit Ihrem Cluster verwenden. Weitere Informationen finden Sie unter Erstellen einer Umgebung in AWS Cloud9. Sie müssen sicherstellen, dass Ihre Amazon-EKS-Steuerebenen-Sicherheitsgruppe Regeln enthält, die den eingehenden Datenverkehr auf Port 443 von Ihrer IDE-Sicherheitsgruppe zulassen. Weitere Informationen finden Sie unter Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe.

Achten Sie bei der Konfiguration `kubectl` für Ihre AWS Cloud9 IDE darauf, AWS Anmeldeinformationen zu verwenden, die bereits der RBAC-Konfiguration Ihres Clusters zugeordnet sind, oder fügen Sie den IAM-Prinzipal, den Ihre IDE verwenden wird, zur RBAC-Konfiguration hinzu, bevor Sie den öffentlichen Zugriff auf Endgeräte entfernen. Weitere Informationen finden Sie unter Zugriff auf Kubernetes APIs gewähren und Nicht autorisiert oder Zugriff verweigert (`kubectl`).

Aktivieren der Secret-Verschlüsselung in einem vorhandenen Cluster

Wenn Sie die Secrets-Verschlüsselung aktivieren, werden die Kubernetes-Secrets mit dem von Ihnen ausgewählten AWS KMS key verschlüsselt. Der KMS-Schlüssel muss die folgenden Bedingungen erfüllen:

- Symmetrisch
- Kann Daten verschlüsseln und entschlüsseln
- Wurde in derselben AWS-Region wie der Cluster erstellt
- Wenn der KMS-Schlüssel in einem anderen Konto erstellt wurde, muss der IAM-Prinzipal Zugriff auf den KMS-Schlüssel haben.

Weitere Informationen finden Sie unter IAM-Prinzipalen in anderen Konten die Verwendung eines KMS-Schlüssels erlauben im AWS Key Management ServiceEntwicklerhandbuch.

> **Warning**
>
> Sie können die Verschlüsselung von Secrets nicht deaktivieren, nachdem Sie sie aktiviert haben. Diese Aktion ist unumkehrbar.

eksctl

Sie können die Verschlüsselung auf zwei Arten aktivieren:

- Fügen Sie Ihrem Cluster mit einem einzigen Befehl Verschlüsselung hinzu.

  Um Ihre Secrets automatisch erneut zu verschlüsseln, führen Sie den folgenden Befehl aus.

```
eksctl utils enable-secrets-encryption \
    --cluster my-cluster \
    --key-arn arn:aws:kms:region-code:account:key/key
```

Um die automatische Neuverschlüsselung Ihrer Secrets zu deaktivieren, führen Sie den folgenden Befehl aus.

```
eksctl utils enable-secrets-encryption
    --cluster my-cluster \
    --key-arn arn:aws:kms:region-code:account:key/key \
    --encrypt-existing-secrets=false
```

- Fügen Sie Ihrem Cluster eine Verschlüsselung mit einer `kms-cluster.yaml`-Datei hinzu.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code

secretsEncryption:
  keyARN: arn:aws:kms:region-code:account:key/key
```

Um Ihre Secrets automatisch erneut zu verschlüsseln, führen Sie den folgenden Befehl aus.

```
eksctl utils enable-secrets-encryption -f kms-cluster.yaml
```

Um die automatische Neuverschlüsselung Ihrer Secrets zu deaktivieren, führen Sie den folgenden Befehl aus.

```
eksctl utils enable-secrets-encryption -f kms-cluster.yaml --encrypt-existing-
secrets=false
```

AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unterhttps://console.aws.amazon.com/eks/home#/clusters.

2. Wählen Sie den Cluster aus, dem Sie die KMS-Verschlüsselung hinzufügen möchten.
3. Wählen Sie die Registerkarte Overview (Übersicht) aus (sie ist standardmäßig ausgewählt).
4. Scrollen Sie nach unten zum Abschnitt Secrets encryption (Secrets-Verschlüsselung) und wählen Sie Enable (Aktivieren) aus.
5. Wählen Sie einen Schlüssel in der Dropdown-Liste und dann die Schaltfläche Enable (Aktivieren) aus. Wenn keine Schlüssel aufgeführt sind, müssen Sie zuerst einen erstellen. Weitere Informationen finden Sie unter Erstellen von Schlüsseln
6. Wählen Sie die Schaltfläche Confirm (Bestätigen) aus, um den ausgewählten Schlüssel zu verwenden.

AWS CLI

1. Ordnen Sie Ihrem Cluster mit dem folgenden AWS CLI-Befehl die Konfiguration der Secrets-Verschlüsselung zu. Ersetzen Sie das *example values* durch Ihr eigenes.

```
aws eks associate-encryption-config \
    --cluster-name my-cluster \
    --encryption-config '[{"resources":["secrets"],"provider":
{"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
  "update": {
    "id": "3141b835-8103-423a-8e68-12c2521ffa4d",
    "status": "InProgress",
    "type": "AssociateEncryptionConfig",
    "params": [
      {
        "type": "EncryptionConfig",
        "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\"arn:aws:kms:region-code:account:key/key\"}}]"
      }
    ],
    "createdAt": 1613754188.734,
    "errors": []
  }
}
```

2. Sie können den Status Ihrer Verschlüsselungsaktualisierung mit dem folgenden Befehl überwachen. Verwenden Sie den `cluster name` und die `update ID`, die in der vorherigen Ausgabe zurückgegeben wurden. Wenn der Status `Successful` angezeigt wird, ist das Update abgeschlossen.

```
aws eks describe-update \
    --region region-code \
    --name my-cluster \
    --update-id 3141b835-8103-423a-8e68-12c2521ffa4d
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
  "update": {
    "id": "3141b835-8103-423a-8e68-12c2521ffa4d",
    "status": "Successful",
    "type": "AssociateEncryptionConfig",
    "params": [
      {
        "type": "EncryptionConfig",
        "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\"arn:aws:kms:region-code:account:key/key\"}}]"
      }
    ],
    "createdAt": 1613754188.734>,
    "errors": []
  }
}
```

3. Führen Sie den Befehl `describe-cluster` aus, um zu überprüfen, ob die Verschlüsselung in Ihrem Cluster aktiviert ist. Die Antwort enthält eine `EncryptionConfig`-Zeichenfolge.

```
aws eks describe-cluster --region region-code --name my-cluster
```

Nachdem Sie die Verschlüsselung in Ihrem Cluster aktiviert haben, müssen Sie alle vorhandenen Secrets mit dem neuen Schlüssel verschlüsseln:

> **Note**
>
> Wenn Sie `eksctl` verwenden, ist die Ausführung des folgenden Befehls nur dann erforderlich, wenn Sie die automatische Neuverschlüsselung Ihrer Secrets deaktivieren.

```
kubectl get secrets --all-namespaces -o json | kubectl annotate --overwrite -f - kms-encryption-timestamp="time value"
```

> **Warning**
>
> Wenn Sie die Secrets-Verschlüsselung für einen vorhandenen Cluster aktivieren und der von Ihnen verwendete KMS-Schlüssel jemals gelöscht wird, können Sie den Cluster nicht wiederherstellen. Wenn Sie den KMS-Schlüssel löschen, wird der Cluster dauerhaft auf einen degradierten Zustand festgelegt. Weitere Informationen finden Sie unter Löschen von AWS-KMS-Schlüsseln.

> **Note**
>
> Standardmäßig erstellt der `create-key`-Befehl einen symmetrischen KMS-Verschlüsselungsschlüssel mit einer Schlüsselrichtlinie, die dem Root-Administrator des Kontos Zugriff auf AWS KMS-Aktionen und -Ressourcen gewährt. Wenn Sie die Berechtigungen begrenzen möchten, müssen Sie sicherstellen, dass die Aktionen `kms:DescribeKey` und `kms:CreateGrant` für die Richtlinie für den Prinzipal zulässig sind, der die API `create-cluster` aufruft.
> Für Cluster, die KMS-Umschlagverschlüsselung verwenden, sind `kms:CreateGrant`-Berechtigungen erforderlich. Die -Bedingung `kms:GrantIsForAWSResource` wird für die CreateCluster -Aktion nicht unterstützt und sollte nicht in KMS-Richtlinien verwendet werden, um `kms:CreateGrant` Berechtigungen für Benutzer zu steuern, die ausführen CreateCluster.

Die Windows-Unterstützung für Ihre Amazon-EKS-Cluster aktivieren

Beachten Sie vor der Bereitstellung von Windows-Knoten die folgenden Überlegungen.

Überlegungen

- Sie können Host-Netzwerke auf Windows-Knoten mithilfe von `HostProcess`-Pods verwenden. Weitere Informationen finden Sie unter Ein Windows `HostProcess`Pod erstellen in der Kubernetes-Dokumentation.
- Amazon-EKS-Cluster müssen einen oder mehrere Linux- oder Fargate-Knoten enthalten, um Kernsystem-Pods auszuführen, die nur unter Linux, z. B. CoreDNS, ausgeführt werden.
- Die Ereignisprotokolle `kubelet` und `kube-proxy` werden in das EKS-Windows-Ereignisprotokoll umgeleitet und auf 200 MB begrenzt.
- Sie können Sicherheitsgruppen für Pods nicht mit Pods verwenden, die auf Windows-Knoten ausgeführt werden.
- Sie können benutzerdefinierte Netzwerke nicht mit Windows-Knoten verwenden.
- Sie können `IPv6` nicht mit Windows-Knoten verwenden.
- Windows-Knoten unterstützen eine Elastic-Network-Schnittstelle pro Knoten. Die Anzahl der Pods, die Sie pro Windows-Knoten ausführen können, entspricht standardmäßig der Anzahl der IP-Adressen, die pro Elastic-Network-Schnittstelle für den Instance-Typ des Knotens verfügbar sind, minus eins. Weitere Informationen finden Sie unter IP-Adressen pro Netzwerkschnittstelle pro Instance-Typ im Amazon EC2 EC2-Benutzerhandbuch.
- In einem Amazon-EKS-Cluster kann ein einzelner Service mit einem Load Balancer bis zu 1024 Backend-Pods unterstützen. Jeder Pod hat seine eigene eindeutige IP-Adresse. Das bisherige Limit von 64 Pods trifft nicht mehr zu, nachdem ein Windows-Server-Update beginnend mit Betriebssystem-Build 17763.2746 durchgeführt wurde.
- Windows-Container werden in Fargate in Amazon-EKS-Pods nicht unterstützt.
- Sie können keine Protokolle aus dem `vpc-resource-controller`-Pod abrufen. Dies war zuvor möglich, als Sie den Controller auf der Datenebene bereitgestellt haben.
- Es gibt eine Abkühlungsphase, bevor einem neuen Pod eine `IPv4`-Adresse zugewiesen wird. Dadurch wird verhindert, dass Datenverkehr aufgrund veralteter `IPv4`-Regeln an einen älteren Pod mit derselben `kube-proxy`-Adresse fließt.

- Die Quelle für den Controller wird auf GitHub verwaltet. Um zum Controller beizutragen oder Probleme gegen den Controller einzureichen, besuchen Sie das Projekt auf GitHub.
- Wenn Sie eine benutzerdefinierte AMI-ID für Windows verwaltete Knotengruppen angeben, fügen Sie `eks:kube-proxy-windows` diese Ihrer AWS IAM Authenticator-Konfigurationsübersicht hinzu. Weitere Informationen finden Sie unter Grenzen und Bedingungen bei der Angabe einer AMI-ID.

Voraussetzungen

- Einen vorhandenen -Cluster. Auf dem Cluster muss eine der in der folgenden Tabelle aufgeführten Kubernetes-Versionen und Plattform-Versionen ausgeführt werden. Alle Kubernetes- und Plattformversionen, die über die aufgeführten hinausgehen, werden ebenfalls unterstützt. Wenn Ihre Cluster- oder Plattformversion früher als eine der folgenden Versionen ist, müssen Sie Legacy-System-Windows-Support auf der Datenebene Ihres Clusters aktivieren. Sobald sich Ihr Cluster in einer der folgenden Kubernetes- und Plattformversionen oder höher befindet, können Sie Legacy-System-Windows-Support entfernen und Windows-Support auf Ihrer Steuerebene aktivieren.

Kubernetes-Version	Plattformversion
1.30	eks.2
1,29	eks.1
1,28	eks.1
1,27	eks.1
1,26	eks.1
1,25	eks.1
1,24	eks.2

- Ihr Cluster muss mindestens einen (wir empfehlen mindestens zwei) Linux-Knoten oder Fargate-Pod haben, um CoreDNS auszuführen. Wenn Sie den Legacy-System-Windows-Support aktivieren, müssen Sie einen Linux-Knoten verwenden (Sie können keinen Fargate-Pod verwenden), um CoreDNS auszuführen.
- Eine vorhandene Amazon-EKS-Cluster-IAM-Rolle.

Aktivieren des Windows-Supports

Wenn Ihr Cluster nicht eine der, oder höhere, Kubernetes- und Plattformversionen aufweist, die in den Voraussetzungen aufgelistet sind, müssen Sie stattdessen den Support für die ältere Windows aktivieren. Weitere Informationen finden Sie unter Aktivieren des Legacy-System-Windows-Supports.

Wenn Sie den Windows-Support für Ihren Cluster noch nie aktiviert haben, fahren Sie mit dem nächsten Schritt fort.

Wenn Sie den Windows-Support für einen Cluster aktiviert haben, der älter als eine der Kubernetes- oder Plattformversionen ist, die in den Voraussetzungen aufgelistet sind, dann müssen Sie zuerst den `vpc-resource-controller` und `vpc-admission-webhook` von Ihrer Datenebene entfernen. Sie sind veraltet und werden nicht mehr benötigt.

Aktivieren Sie den Windows-Support für Ihren Cluster

1. Wenn Sie keine Amazon-Linux-Knoten in Ihrem Cluster haben und Sicherheitsgruppen für Pods verwenden, fahren Sie mit dem nächsten Schritt fort. Ansonsten bestätigen Sie, dass die verwaltete Richtlinie `AmazonEKSVPCResourceController` Ihrer Cluster-Rolle angefügt ist. Ersetzen Sie den *`eksClusterRole`* durch Ihren Clusterrollennamen.

   ```
   aws iam list-attached-role-policies --role-name eksClusterRole
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   {
       "AttachedPolicies": [
           {
               "PolicyName": "AmazonEKSClusterPolicy",
               "PolicyArn": "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
           },
           {
               "PolicyName": "AmazonEKSVPCResourceController",
               "PolicyArn": "arn:aws:iam::aws:policy/AmazonEKSVPCResourceController"
           }
       ]
   }
   ```

   Wenn die Richtlinie wie in der vorherigen Ausgabe angehängt ist, überspringen Sie den nächsten Schritt.

2. Hängen Sie die von AmazonEksVPC ResourceController verwaltete Richtlinie an Ihre an. Amazon-EKS-Cluster-IAM-Rolle Ersetzen Sie den *eksClusterRole* durch Ihren Clusterrollennamen.

```
aws iam attach-role-policy \
  --role-name eksClusterRole \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSVPCResourceController
```

3. Erstellen Sie eine Datei mit dem Namen *vpc-resource-controller-configmap.yaml* und dem folgenden Inhalt.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: amazon-vpc-cni
  namespace: kube-system
data:
  enable-windows-ipam: "true"
```

4. Anwendung der `ConfigMap` auf Ihren Cluster.

```
kubectl apply -f vpc-resource-controller-configmap.yaml
```

5. Stellen Sie sicher, dass Ihre `aws-auth` `ConfigMap` eine Zuordnung für die Instance-Rolle des Windows-Knotens enthält, sodass sie die RBAC-Berechtigungsgruppe `eks:kube-proxy-windows` einschließt. Prüfen Sie dies durch Ausführung des folgenden Befehls.

```
kubectl get configmap aws-auth -n kube-system -o yaml
```

Eine Beispielausgabe sieht wie folgt aus.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
```

```
      - eks:kube-proxy-windows # This group is required for Windows DNS resolution
 to work
      rolearn: arn:aws:iam::111122223333:role/eksNodeRole
      username: system:node:{{EC2PrivateDNSName}}
[...]
```

`eks:kube-proxy-windows` sollte unter „Gruppen“ aufgelistet sein. Wenn die Gruppe nicht angegeben ist, müssen Sie Ihre `ConfigMap` aktualisieren oder sie so erstellen, dass sie die erforderliche Gruppe enthält. Weitere Informationen zur `aws-auth` `ConfigMap` finden Sie unter Anwenden von `aws-auth`ConfigMap auf Ihren Cluster.

Entfernen des Legacy-System-Windows-Supports von Ihrer Datenebene

Wenn Sie den Windows-Support für einen Cluster aktiviert haben, der älter als eine der Kubernetes- oder Plattformversionen ist, die in den Voraussetzungen aufgelistet sind, dann müssen Sie zuerst den `vpc-resource-controller` und `vpc-admission-webhook` von Ihrer Datenebene entfernen. Sie sind veraltet und werden nicht mehr benötigt, da die von ihnen bereitgestellte Funktionalität jetzt auf der Steuerebene aktiviert ist.

1. Deinstallieren Sie das `vpc-resource-controller` mit dem folgenden Befehl. Verwenden Sie diesen Befehl unabhängig davon, mit welchem Tool Sie ihn ursprünglich installiert haben. Ersetzen Sie *`region-code`* (nur die Instance des betreffenden Textes nach `/manifests/`) durch die AWS-Region , in der sich Ihr Cluster befindet.

   ```
   kubectl delete -f https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-resource-controller/latest/vpc-resource-controller.yaml
   ```

2. Deinstallieren Sie das `vpc-admission-webhook` anhand der Anweisungen für das Tool, mit dem Sie es installiert haben.

   eksctl

   Führen Sie die folgenden Befehle aus.

   ```
   kubectl delete deployment -n kube-system vpc-admission-webhook
   kubectl delete service -n kube-system vpc-admission-webhook
   kubectl delete mutatingwebhookconfigurations.admissionregistration.k8s.io vpc-admission-webhook-cfg
   ```

kubectl on macOS or Windows

Führen Sie den folgenden Befehl aus. Ersetzen Sie *region-code* (nur die Instanz des Textes danach `/manifests/`) durch den Text, in dem AWS-Region sich Ihr Cluster befindet.

```
kubectl delete -f https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/vpc-admission-webhook-deployment.yaml
```

3. Aktivieren des Windows-Supports für Ihren Cluster auf der Steuerebene.

Deaktivieren des Windows-Supports

So deaktivieren Sie den Windows-Support für Ihren Cluster

1. Wenn Ihr Cluster Amazon-Linux-Knoten enthält und Sie Sicherheitsgruppen für Pods mit ihnen verwenden, überspringen Sie dann diesen Schritt.

   Entfernen Sie die `AmazonVPCResourceController`-verwaltete IAM-Richtlinie von Ihrer Cluster-Rolle. Ersetzen Sie *eksClusterRole* durch den Namen Ihrer Clusterrolle und *111122223333* durch Ihre Konto-ID.

   ```
   aws iam detach-role-policy \
       --role-name eksClusterRole \
       --policy-arn arn:aws:iam::aws:policy/AmazonEKSVPCResourceController
   ```

2. Deaktivieren Sie Windows IPAM in der `amazon-vpc-cni` ConfigMap.

   ```
   kubectl patch configmap/amazon-vpc-cni \
                       -n kube-system \
                       --type merge \
                       -p '{"data":{"enable-windows-ipam":"false"}}'
   ```

Bereitstellen von Pods

Wenn Sie Pods in Ihrem Cluster bereitstellen, müssen Sie das Betriebssystem angeben, das sie verwenden, wenn Sie eine Mischung aus Knotentypen ausführen.

Verwenden Sie für Linux-Pods den folgenden Knotenauswahltext in Ihren Manifesten.

```
nodeSelector:
        kubernetes.io/os: linux
        kubernetes.io/arch: amd64
```

Verwenden Sie für Windows-Pods den folgenden Knotenauswahltext in Ihren Manifesten.

```
nodeSelector:
        kubernetes.io/os: windows
        kubernetes.io/arch: amd64
```

Sie können eine Beispielanwendung bereitstellen, um die verwendeten Knotenselektoren zu sehen.

Aktivieren des Legacy-System-Windows-Supports

Wenn Ihr Cluster eine der, oder spätere, Kubernetes- und Plattformversionen hat, die in den Voraussetzungen aufgelistet sind, wird empfohlen, den Windows-Support stattdessen auf der Steuerebene zu aktivieren. Weitere Informationen finden Sie unter Aktivieren des Windows-Supports.

Die folgenden Schritte helfen Ihnen, den Legacy-System-Windows-Support für die Datenebene Ihres Amazon-EKS-Clusters zu aktivieren, wenn Ihre Cluster- oder Plattformversion älter als die Versionen ist, die in den Voraussetzungen aufgelistet sind. Sobald Ihre Cluster- und Plattformversion gleich einer, oder höher als eine, Version ist, die in den Voraussetzungen aufgelistet ist, empfehlen wir Ihnen, dass Sie den Legacy-System-Windows-Support entfernen und ihn für die Steuerebene aktivieren.

Sie können `eksctl`, einen Windows-Client oder einen macOS- oder Linux-Client verwenden, um den Legacy-System-Windows-Support für Ihren Cluster zu aktivieren.

eksctl

So aktivieren Sie den Legacy-System-Windows-Support für Ihren Cluster mit **`eksctl`**

Voraussetzung

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre Version mit dem folgenden Befehl überprüfen.

```
eksctl version
```

Weitere Informationen zum Installieren oder Upgraden von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

1. Aktivieren Sie Windows-Unterstützung für Ihren Amazon-EKS-Cluster mit dem folgenden `eksctl`-Befehl. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters. Mit diesem Befehl wird der Webhook des VPC-Ressourcencontrollers und des VPC-Zugangscontrollers bereitgestellt, die auf Amazon-EKS-Clustern zum Ausführen von Windows-Workloads erforderlich sind.

   ```
   eksctl utils install-vpc-controllers --cluster my-cluster --approve
   ```

   > **Important**
   >
   > Der Webhook für den VPC-Zulassungscontroller ist mit einem Zertifikat signiert, das ein Jahr nach dem Ausstellungsdatum abläuft. Um Ausfallzeiten zu vermeiden, stellen Sie sicher, dass Sie das Zertifikat vor Ablauf erneuern. Weitere Informationen finden Sie unter Erneuerung des VPC-Zulassungs-Webhook-Zertifikats.

2. Nachdem Sie Windows-Unterstützung aktiviert haben, können Sie eine Windows-Knotengruppe in Ihrem Cluster starten. Weitere Informationen finden Sie unter Starten selbstverwalteter Windows-Knoten.

Windows

So aktivieren Sie den Legacy-System-Windows-Support für Ihren Cluster mit einem Windows-Client

Ersetzen Sie in den folgenden Schritten *`region-code`* durch die AWS-Region , in der sich Ihr Cluster befindet.

1. Stellen Sie den VPC-Ressourcencontroller in Ihrem Cluster bereit.

   ```
   kubectl apply -f https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-resource-controller/latest/vpc-resource-controller.yaml
   ```

2. Stellen Sie den Webhook des VPC-Zugangscontrollers für Ihren Cluster bereit.

   a. Laden Sie die erforderlichen Skripte und Bereitstellungsdateien herunter.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/
vpc-admission-webhook/latest/vpc-admission-webhook-deployment.yaml;
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/
vpc-admission-webhook/latest/Setup-VPCAdmissionWebhook.ps1;
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/
vpc-admission-webhook/latest/webhook-create-signed-cert.ps1;
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/
vpc-admission-webhook/latest/webhook-patch-ca-bundle.ps1;
```

b. Installieren Sie OpenSSL und jq.

c. Einrichten und Bereitstellen des VPC-Zugangswebhooks.

```
./Setup-VPCAdmissionWebhook.ps1 -DeploymentTemplate ".\vpc-admission-
webhook-deployment.yaml"
```

> **Important**
>
> Der Webhook für den VPC-Zulassungscontroller ist mit einem Zertifikat signiert, das ein Jahr nach dem Ausstellungsdatum abläuft. Um Ausfallzeiten zu vermeiden, stellen Sie sicher, dass Sie das Zertifikat vor Ablauf erneuern. Weitere Informationen finden Sie unter Erneuerung des VPC-Zulassungs-Webhook-Zertifikats.

3. Stellen Sie fest, ob Ihr Cluster über die erforderliche Clusterrollenbindung verfügt.

```
kubectl get clusterrolebinding eks:kube-proxy-windows
```

Wenn eine Ausgabe ähnlich der folgenden Beispielausgabe zurückgegeben wird, verfügt der Cluster über die erforderliche Rollenbindung.

```
NAME                       AGE
eks:kube-proxy-windows     10d
```

Wenn die Ausgabe `Error from server (NotFound)` enthält, verfügt der Cluster nicht über die erforderliche Clusterrollenbindung. Fügen Sie die Bindung hinzu, indem Sie eine Datei mit dem Namen *`eks-kube-proxy-windows-crb.yaml`* mit dem folgenden Inhalt erstellen.

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: eks:kube-proxy-windows
  labels:
    k8s-app: kube-proxy
    eks.amazonaws.com/component: kube-proxy
subjects:
  - kind: Group
    name: "eks:kube-proxy-windows"
roleRef:
  kind: ClusterRole
  name: system:node-proxier
  apiGroup: rbac.authorization.k8s.io
```

Wenden Sie die Konfiguration auf den Cluster an.

```
kubectl apply -f eks-kube-proxy-windows-crb.yaml
```

4. Nachdem Sie Windows-Unterstützung aktiviert haben, können Sie eine Windows-Knotengruppe in Ihrem Cluster starten. Weitere Informationen finden Sie unter Starten selbstverwalteter Windows-Knoten.

macOS and Linux

So aktivieren Sie den Legacy-System-Windows-Support für Ihren Cluster mit einem macOS oder Linux-Client

Dieses Verfahren erfordert, dass die `openssl`-Bibliothek und der `jq`-JSON-Prozessor auf Ihrem Clientsystem installiert sind.

Ersetzen Sie in den folgenden Schritten *`region-code`* durch die AWS-Region , in der sich Ihr Cluster befindet.

1. Stellen Sie den VPC-Ressourcencontroller in Ihrem Cluster bereit.

```
kubectl apply -f https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-resource-controller/latest/vpc-resource-controller.yaml
```

2. Erstellen Sie das Webhook-Manifest für den VPC-Zugangscontroller für Ihren Cluster.

a. Laden Sie die erforderlichen Skripte und Bereitstellungsdateien herunter.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/
vpc-admission-webhook/latest/webhook-create-signed-cert.sh
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/
vpc-admission-webhook/latest/webhook-patch-ca-bundle.sh
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/
vpc-admission-webhook/latest/vpc-admission-webhook-deployment.yaml
```

b. Fügen Sie den Shell-Skripten Berechtigungen hinzu, damit sie ausgeführt werden können.

```
chmod +x webhook-create-signed-cert.sh webhook-patch-ca-bundle.sh
```

c. Erstellen Sie ein Geheimnis für eine sichere Kommunikation.

```
./webhook-create-signed-cert.sh
```

d. Überprüfen Sie das Geheimnis.

```
kubectl get secret -n kube-system vpc-admission-webhook-certs
```

e. Konfigurieren Sie den Webhook und erstellen Sie eine Bereitstellungsdatei.

```
cat ./vpc-admission-webhook-deployment.yaml | ./webhook-patch-ca-bundle.sh >
 vpc-admission-webhook.yaml
```

3. Stellen Sie den VPC-Zugangswebhook bereit.

```
kubectl apply -f vpc-admission-webhook.yaml
```

> **Important**
>
> Der Webhook für den VPC-Zulassungscontroller ist mit einem Zertifikat signiert, das ein Jahr nach dem Ausstellungsdatum abläuft. Um Ausfallzeiten zu vermeiden, stellen Sie sicher, dass Sie das Zertifikat vor Ablauf erneuern. Weitere Informationen finden Sie unter Erneuerung des VPC-Zulassungs-Webhook-Zertifikats.

4. Stellen Sie fest, ob Ihr Cluster über die erforderliche Clusterrollenbindung verfügt.

```
kubectl get clusterrolebinding eks:kube-proxy-windows
```

Wenn eine Ausgabe ähnlich der folgenden Beispielausgabe zurückgegeben wird, verfügt der Cluster über die erforderliche Rollenbindung.

```
NAME                     ROLE                                AGE
eks:kube-proxy-windows   ClusterRole/system:node-proxier   19h
```

Wenn die Ausgabe `Error from server (NotFound)` enthält, verfügt der Cluster nicht über die erforderliche Clusterrollenbindung. Fügen Sie die Bindung hinzu, indem Sie eine Datei mit dem Namen *`eks-kube-proxy-windows-crb.yaml`* mit dem folgenden Inhalt erstellen.

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: eks:kube-proxy-windows
  labels:
    k8s-app: kube-proxy
    eks.amazonaws.com/component: kube-proxy
subjects:
  - kind: Group
    name: "eks:kube-proxy-windows"
roleRef:
  kind: ClusterRole
  name: system:node-proxier
  apiGroup: rbac.authorization.k8s.io
```

Wenden Sie die Konfiguration auf den Cluster an.

```
kubectl apply -f eks-kube-proxy-windows-crb.yaml
```

5. Nachdem Sie Windows-Unterstützung aktiviert haben, können Sie eine Windows-Knotengruppe in Ihrem Cluster starten. Weitere Informationen finden Sie unter Starten selbstverwalteter Windows-Knoten.

Erneuerung des VPC-Zulassungs-Webhook-Zertifikats

Das vom VPC-Zulassungswebhook verwendete Zertifikat läuft ein Jahr nach der Ausstellung ab. Um Ausfallzeiten zu vermeiden, ist es wichtig, dass Sie das Zertifikat vor Ablauf erneuern. Sie können das Ablaufdatum Ihres aktuellen Zertifikats mit dem folgenden Befehl überprüfen.

```
kubectl get secret \
    -n kube-system \
    vpc-admission-webhook-certs -o json | \
    jq -r '.data."cert.pem"' | \
    base64 -decode | \
    openssl x509 \
    -noout \
    -enddate | \
    cut -d= -f2
```

Eine Beispielausgabe sieht wie folgt aus.

```
May 28 14:23:00 2022 GMT
```

Sie können das Zertifikat mit `eksctl` oder einem Windows- oder Linux/macOS-Computer erneuern. Befolgen Sie die Anweisungen für das Tool, mit dem Sie den VPC-Zulassungswebhook ursprünglich installiert haben. Wenn Sie beispielsweise den VPC-Zulassungswebhook ursprünglich mit `eksctl` installiert haben, sollten Sie das Zertifikat gemäß den Anweisungen auf der Registerkarte `eksctl` erneuern.

eksctl

1. Installieren Sie das Zertifikat neu. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

   ```
   eksctl utils install-vpc-controllers -cluster my-cluster -approve
   ```

2. Stellen Sie sicher, dass Sie die folgende Ausgabe erhalten.

   ```
   2021/05/28 05:24:59 [INFO] generate received request
   2021/05/28 05:24:59 [INFO] received CSR
   2021/05/28 05:24:59 [INFO] generating key: rsa-2048
   2021/05/28 05:24:59 [INFO] encoded CSR
   ```

3. Starten Sie die Webhook-Bereitstellung neu.

```
kubectl rollout restart deployment -n kube-system vpc-admission-webhook
```

4. Wenn das von Ihnen erneuerte Zertifikat abgelaufen ist und Windows-Pods im `Container creating`-Zustand hängen bleiben, müssen Sie diese Pods löschen und erneut bereitstellen.

Windows

1. Rufen Sie das Skript ab, um ein neues Zertifikat zu generieren.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/webhook-create-signed-cert.ps1;
```

2. Parameter für das Skript vorbereiten.

```
./webhook-create-signed-cert.ps1 -ServiceName vpc-admission-webhook-svc -SecretName vpc-admission-webhook-certs -Namespace kube-system
```

3. Starten Sie die Webhook-Bereitstellung neu.

```
kubectl rollout restart deployment -n kube-system vpc-admission-webhook-deployment
```

4. Wenn das von Ihnen erneuerte Zertifikat abgelaufen ist und Windows-Pods im `Container creating`-Zustand hängen bleiben, müssen Sie diese Pods löschen und erneut bereitstellen.

Linux and macOS

Voraussetzung

Auf Ihrem Computer müssen OpenSSL und `jq` installiert sein.

1. Rufen Sie das Skript ab, um ein neues Zertifikat zu generieren.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/webhook-create-signed-cert.sh
```

2. Ändern Sie die Berechtigungen.

```
chmod +x webhook-create-signed-cert.sh
```

3. Führen Sie das Skript aus.

```
./webhook-create-signed-cert.sh
```

4. Starten Sie den Webhook neu.

```
kubectl rollout restart deployment -n kube-system vpc-admission-webhook-deployment
```

5. Wenn das von Ihnen erneuerte Zertifikat abgelaufen ist und Windows-Pods im `Container creating`-Zustand hängen bleiben, müssen Sie diese Pods löschen und erneut bereitstellen.

Unterstützung einer höheren Pod-Dichte auf Windows-Knoten

In Amazon EKS wird jedem Pod eine `IPv4`-Adresse von Ihrer VPC zugewiesen. Aus diesem Grund wird die Anzahl der Pods-Geräte, die Sie auf einem Knoten bereitstellen können, durch die verfügbaren IP-Adressen begrenzt, auch wenn auf dem Knoten genügend Ressourcen vorhanden sind, um mehr Pods auszuführen. Da von einem Windows-Knoten nur eine elastische Netzwerkschnittstelle unterstützt wird, entspricht die maximale Anzahl verfügbarer IP-Adressen auf einem Windows-Knoten standardmäßig:

```
Number of private IPv4 addresses for each interface on the node - 1
```

Eine IP-Adresse wird als primäre IP-Adresse der Netzwerkschnittstelle verwendet und kann daher Pods nicht zugewiesen werden.

Sie können eine höhere Pod-Dichte auf Windows-Knoten aktivieren, indem Sie die IP-Präfix-Delegierung aktivieren. Mit diesem Feature können Sie der primären Netzwerkschnittstelle ein `/28`-`IPv4`-Präfix zuweisen, anstatt sekundäre `IPv4`-Adressen zuzuweisen. Durch die Zuweisung eines IP-Präfixes wird die maximale Anzahl verfügbarer `IPv4`-Adressen auf dem Knoten auf Folgendes erhöht:

```
(Number of private IPv4 addresses assigned to the interface attached to the node - 1) *
 16
```

Angesichts dieser deutlich größeren Anzahl verfügbarer IP-Adressen sollten verfügbare IP-Adressen Ihre Fähigkeit, die Anzahl der Pods auf Ihren Knoten zu skalieren, nicht einschränken. Weitere Informationen finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten.

Anforderungen an private Cluster

In diesem Thema wird beschrieben, wie Sie einen Amazon EKS-Cluster bereitstellen AWS Cloud, der auf dem bereitgestellt wird, aber keinen ausgehenden Internetzugang hat. Wenn Sie einen lokalen Cluster aktiviert haben AWS OutpostsStarten selbstverwalteter Amazon Linux-Knoten auf einem Outpost, finden Sie statt dieses Themas weitere Informationen unter.

Wenn Sie mit dem Amazon-EKS-Netzwerk nicht vertraut sind, finden Sie unter Entmystifizierung von Cluster-Netzwerken für Amazon EKS-Worker-Knoten. Wenn Ihr Cluster nicht über einen ausgehenden Internetzugriff verfügt, muss es die folgenden Anforderungen erfüllen:

- Ihr Cluster muss Images von einer Container-Registry in Ihrer VPC beziehen. Sie können eine Amazon Elastic Container Registry in Ihrer VPC erstellen und Container-Images dorthin kopieren, damit Ihre Knoten daraus abrufen können. Weitere Informationen finden Sie unter Kopieren eines Container-Images von einem Repository in ein anderes.
- In Ihrem Cluster muss der private Endpunkt-Zugriff aktiviert sein. Dies ist erforderlich, damit sich Knoten beim Cluster-Endpunkt registrieren können. Der Endpunkt für öffentlichen Zugriff ist optional. Weitere Informationen finden Sie unter Zugriffskontrolle für den Amazon-EKS-Cluster-Endpunkt.
- Selbstverwaltete Linux- und Windows-Knoten müssen die folgenden Bootstrap-Argumente enthalten, bevor sie gestartet werden. Diese Argumente umgehen die Amazon-EKS-Introspektion und erfordern keinen Zugriff auf die Amazon-EKS-API innerhalb der VPC.

  1. Bestimmen Sie den Wert des Endpunkts Ihres Clusters mit dem folgenden Befehl. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

     ```
     aws eks describe-cluster --name my-cluster --query cluster.endpoint --output text
     ```

     Eine Beispielausgabe sieht wie folgt aus.

     ```
     https://EXAMPLE108C897D9B2F1B21D5EXAMPLE.sk1.region-code.eks.amazonaws.com
     ```

  2. Bestimmen Sie den Wert der Zertifizierungsstelle Ihres Clusters mit dem folgenden Befehl. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

     ```
     aws eks describe-cluster --name my-cluster --query cluster.certificateAuthority --output text
     ```

Die zurückgegebene Ausgabe ist eine lange Zeichenfolge.

3. Ersetzen Sie *cluster-endpoint* und *certificate-authority* in den folgenden Befehlen durch die Werte, die in der Ausgabe der vorherigen Befehle zurückgegeben wurden. Weitere Informationen zur Angabe von Bootstrap-Argumenten beim Start von selbstverwalteten Knoten finden Sie unter Starten selbstverwalteter Amazon Linux-Knoten und Starten selbstverwalteter Windows-Knoten.

- Für Linux-Knoten:

```
--apiserver-endpoint cluster-endpoint --b64-cluster-ca certificate-authority
```

Weitere Argumente finden Sie im Bootstrap-Skript auf GitHub.

- Für Windows-Knoten:

> **Note**
>
> Wenn Sie CIDR für den benutzerdefinierten Service verwenden, müssen Sie ihn mithilfe des `-ServiceCIDR`-Parameters angeben. Andernfalls schlägt die DNS-Auflösung für Pods im Cluster fehl.

```
-APIServerEndpoint cluster-endpoint -Base64ClusterCA certificate-authority
```

Weitere Argumente finden Sie unter Bootstrap-Skript-Konfigurationsparameter.

- Das `aws-auth ConfigMap` Ihres Clusters muss innerhalb Ihrer VPC erstellt werden. Weitere Informationen zum Erstellen und Hinzufügen von Einträgen zu `aws-auth ConfigMap` erhalten Sie durch Eingabe von **`eksctl create iamidentitymapping --help`** in Ihrem Terminal. Falls `ConfigMap` auf Ihrem Server nicht vorhanden ist, wird es von `eksctl` erstellt, wenn Sie den Befehl zum Hinzufügen einer Identitätszuordnung verwenden.
- Pods die mit IAM-Rollen für Service-Konten konfiguriert sind, beziehen Anmeldeinformationen von einem AWS Security Token Service (AWS STS)-API-Aufruf. Wenn es keinen ausgehenden Internetzugang gibt, müssen Sie einen AWS STS VPC-Endpunkt in Ihrer VPC erstellen und verwenden. Die meisten AWS `v1` SDKs verwenden standardmäßig den globalen AWS STS Endpunkt (`sts.amazonaws.com`), der den AWS STS VPC-Endpunkt nicht verwendet. Um den AWS STS VPC-Endpunkt zu verwenden, müssen Sie Ihr SDK möglicherweise so konfigurieren, dass es den regionalen AWS STS Endpunkt (`sts.`*region-code*`.amazonaws.com`) verwendet.

Weitere Informationen finden Sie unter Den AWS Security Token Service Endpunkt für ein Dienstkonto konfigurieren.

- Die VPC-Subnetze Ihres Clusters müssen über einen VPC-Schnittstellenendpunkt für jedes AWS-Services verfügen, auf das Ihr Pods Zugriff benötigt. Weitere Informationen finden Sie unter Zugriff auf einen AWS -Service über einen Schnittstellen-VPC-Endpunkt. Einige häufig verwendete Services und Endpunkte sind in der folgenden Tabelle aufgeführt. eine vollständige Liste der Endpunkte finden Sie unter AWS -Services, die mit AWS PrivateLink integriert sind im AWS PrivateLink -Handbuch.

Service	Endpunkt
Amazon EC2	com.amazonaws.*region-code* .ec2
Amazon Elastic Container Registry (zum Abrufen von Container-Images)	com.amazonaws.*region-code* .ecr.api, com.amazonaws.*region-code* .ecr.dkr, and com.amazonaws.*region-code* .s3
Application Load Balancer und Network Load Balancer	com.amazonaws.*region-code* .elasticloadbalancing
AWS X-Ray	com.amazonaws.*region-code* .xray
CloudWatch Amazon-Protokolle	com.amazonaws.*region-code* .logs
AWS Security Token Service (erforderlich, wenn IAM-Rollen für Dienstkonten verwendet werden)	com.amazonaws.*region-code* .sts

Überlegungen

- Alle selbstverwalteten Knoten müssen in Subnetzen bereitgestellt werden, die über die von Ihnen benötigten VPC-Schnittstellenendpunkte verfügen. Wenn Sie eine verwaltete Knotengruppe erstellen, muss die Endpunktsicherheitsgruppe der VPC-Schnittstelle das CIDR für die Subnetze zulassen, oder Sie müssen die erstellte Knotensicherheitsgruppe zur Endpunktsicherheitsgruppe der VPC-Schnittstelle hinzufügen.

- Wenn Sie Amazon EFS-Volumes Pods verwenden, muss vor der Bereitstellung von die Datei kustomization.yaml des Treibers geändert werdenAmazon EFS-CSI-Treiber, um festzulegen, dass die Container-Images dieselben AWS-Region wie der Amazon EKS-Cluster verwenden.
- Sie können die verwenden AWS Load Balancer Controller, um AWS Application Load Balancers (ALB) und Network Load Balancer in Ihrem privaten Cluster bereitzustellen. Beim Bereitstellen sollten Sie Befehlszeilen-Flags verwenden, um `enable-shield`, `enable-waf` und `enable-wafv2` auf falsch zu setzen. Die Zertifikatserkennung mit Hostnamen aus Ingress-Objekten wird nicht unterstützt. Dies liegt daran, dass der Controller eine Verbindung herstellen muss AWS Certificate Manager, die keinen VPC-Schnittstellenendpunkt hat.

  Der Controller unterstützt Network Load Balancer mit IP-Zielen, die für die Verwendung mit Fargate erforderlich sind. Weitere Informationen finden Sie unter Application Load Balancing auf Amazon EKS und Erstellen eines Network Load Balancers.
- Cluster Autoscaler wird unterstützt. Stellen Sie beim Bereitstellen von Cluster-Autoscaler-Pods sicher, dass die Befehlszeile `--aws-use-static-instance-list=true` enthält. Weitere Informationen finden Sie unter Verwenden der statischen Instance-Liste auf GitHub. Die Worker-Node-VPC muss auch den VPC-Endpunkt und den AWS STS VPC-Endpunkt für Autoscaling enthalten.
- Einige Container-Softwareprodukte verwenden API-Aufrufe, die auf sie zugreifen, um die Nutzung zu überwachen AWS Marketplace Metering Service . Private Cluster lassen diese Aufrufe nicht zu, daher können Sie diese Containertypen nicht in privaten Clustern verwenden.

Kubernetes-Versionen für Amazon EKS

Kubernetes entwickelt sich durch neue Features, Designaktualisierungen und Fehlerbehebungen schnell weiter. Die Community veröffentlicht im Durchschnitt alle vier Monate neue Kubernetes-Nebenversionen (z. B. `1.30`). Amazon EKS folgt für Nebenversionen dem Upstream-Release- und Einstellungszyklus. Wenn neue Kubernetes-Versionen in Amazon EKS verfügbar werden, empfehlen wir Ihnen, Ihre Cluster proaktiv auf die neueste verfügbare Version zu aktualisieren.

Eine Nebenversion wird in den ersten 14 Monaten nach ihrem Release standardmäßig in Amazon EKS unterstützt. Sobald eine Version den Zeitraum des Standard-Supports überschritten hat, wird sie die folgenden 12 Monate automatisch in den verlängerten Support aufgenommen. Mit dem verlängerten Support können Sie gegen Aufpreis pro Cluster-Stunde länger bei einer bestimmten Kubernetes-Version bleiben. Wenn Sie Ihren Cluster vor Ablauf des verlängerten Supportzeitraums

nicht aktualisiert haben, wird Ihr Cluster automatisch auf die älteste derzeit unterstützte erweiterte Version aktualisiert.

Wir empfehlen Ihnen, Ihren Cluster mit der neuesten verfügbaren Kubernetes-Version zu erstellen, die von Amazon EKS unterstützt wird. Wenn Ihre Anwendung eine bestimmte Version von Kubernetes erfordert, können Sie ältere Versionen auswählen. Sie können neue Amazon-EKS-Cluster auf jeder Version erstellen, die mit Standard- oder verlängertem Support angeboten wird.

Verfügbare Versionen mit Standard-Support

Die folgenden Kubernetes-Versionen sind derzeit mit Amazon-EKS-Standard-Support verfügbar:

- `1.30`
- `1.29`
- `1.28`
- `1.27`
- `1.26`

Wichtige Änderungen, die Sie für jede Version des Standard-Supports beachten sollten, finden Sie unter Versionshinweise für Standard-Supportversionen.

Verfügbare Versionen mit verlängertem Support

Die folgenden Kubernetes-Versionen sind derzeit mit Amazon EKS verlängertem Support verfügbar:

- `1.25`
- `1.24`
- `1.23`

Wichtige Änderungen, die Sie für jede Version des verlängerten Supports beachten sollten, finden Sie unter Versionshinweise für Versionen mit verlängerten Support.

Die folgenden Kubernetes Versionen sind derzeit im erweiterten Support von Amazon EKS verfügbar, mit der zusätzlichen Anforderung, dass Sie mit diesen Versionen keine neuen Cluster erstellen können:

- `1.22`

- 1.21

Informationen zu diesen Versionen finden Sie unter Versionshinweise für die Versionen 1.21 und 1.22

Amazon-EKS-Kubernetes-Release-Kalender

Die folgende Tabelle zeigt wichtige Release- und Supportdaten, die für jede Kubernetes-Version zu berücksichtigen sind.

> **Note**
>
> Daten mit nur einem Monat und einem Jahr sind ungefähre Angaben und werden mit einem genauen Datum aktualisiert, wenn es bekannt ist.

Kubernetes-Version	Upstream-Release	Amazon-EKS-Version	Ende des Standard-Supports	Ende des verlängerten Supports
1.30	17. April 2024	23. Mai 2024	23. Juli 2025	23. Juli 2026
1.29	13. Dezember 2023	23. Januar 2024	23. März 2025	23. März 2026
1.28	15. August 2023	26. September 202	26. November 2024	26. November 2025
1.27	11. April 2023	24. Mai 2023	24. Juli 2024	24. Juli 2025
1.26	9. Dezember 2022	11. April 2023	11. Juni 2024	11. Juni 2025
1.25	23. August 2022	22. Februar 2023	1. Mai 2024	1. Mai 2025
1.24	3. Mai 2022	15. November 202	31. Januar 2024	31. Januar 2025
1.23	7. Dezember 2021	11. August 2022	11. Oktober 2023	11. Oktober 2024

Kubernetes-Version	Upstream-Release	Amazon-EKS-Version	Ende des Standard-Supports	Ende des verlängerten Supports
1.22	4. August 2021	4. April 2022	4. Juni 2023	1. September 2024
1.21	8. April 2021	19. Juli 2021	16. Februar 2023	15. Juli 2024

FAQs zu Amazon-EKS-Versionen

Wie viele Kubernetes-Versionen sind mit Standard-Support verfügbar?

Im Einklang mit dem Kubernetes-Community-Support für Kubernetes-Versionen verpflichtet sich Amazon EKS, zu jeder Zeit Standard-Support für mindestens vier produktionsbereite Versionen von Kubernetes anzubieten. Wir kündigen das Ende des Standard-Supportzeitraums einer bestimmten Kubernetes-Nebenversion mindestens 60 Tage im Voraus an. Aufgrund des Amazon-EKS-Qualifizierungs- und Veröffentlichungsprozesses für neue Kubernetes-Versionen liegt das Standard-Supportende einer Kubernetes-Version auf Amazon EKS am oder nach dem Datum, an dem das Kubernetes-Projekt die Unterstützung des Versions-Upstreams eingestellt hat.

Wie lange erhält ein Kubernetes Standard-Support von Amazon EKS?

Eine Kubernetes-Version wird ab der ersten Verfügbarkeit auf Amazon EKS 14 Monate lang unterstützt. Dies gilt auch dann, wenn Kubernetes auf Upstream-Plattformen eine auf Amazon EKS verfügbare Version nicht mehr unterstützt. Wir rückportieren Sicherheitspatches, die für die Kubernetes-Versionen gelten, die auf Amazon EKS unterstützt werden.

Werde ich benachrichtigt, wenn der Standard-Support für eine Kubernetes-Version auf Amazon EKS endet?

Ja. Wenn auf Clustern in Ihrem Konto die Version ausgeführt wird, die sich dem Ende des Supports nähert, sendet Amazon EKS innerhalb von AWS Health Dashboard etwa 12 Monaten nach der Veröffentlichung der Kubernetes Version auf Amazon EKS eine Benachrichtigung. Die Mitteilung enthält das Datum des Support-Laufzeitendes. Dies ist mindestens 60 Tage ab dem Datum der Benachrichtigung.

Welche Kubernetes-Features werden von Amazon EKS unterstützt?

Amazon EKS unterstützt alle generell verfügbaren (GA) Features der Kubernetes-API. Ab Kubernetes-Version `1.24` sind neue Beta-APIs standardmäßig nicht in Clustern aktiviert. Allerdings sind bestehende Beta-APIs und neue Versionen vorhandener Beta-APIs weiterhin standardmäßig aktiviert. Alpha-Features werden nicht unterstützt.

Werden von Amazon EKS verwaltete Knotengruppen automatisch zusammen mit der Version der Cluster-Steuerebene aktualisiert?

Nein, eine verwaltete Knotengruppe erstellt Amazon-EC2-Instances in Ihrem Konto. Diese Instances werden nicht automatisch aktualisiert, wenn Sie oder Amazon EKS Ihre Steuerebene aktualisieren. Weitere Informationen finden Sie unter Aktualisieren einer verwalteten Knotengruppe. Wir empfehlen, dieselbe Kubernetes-Version auf Ihrer Steuerebene und Ihren Knoten beizubehalten.

Werden selbstverwaltete Knotengruppen automatisch zusammen mit der Version der Cluster-Steuerebene aktualisiert?

Nein, eine selbstverwaltete Knotengruppe umfasst Amazon-EC2-Instances in Ihrem Konto. Diese Instances werden nicht automatisch aktualisiert, wenn Sie oder Amazon EKS die Version der Steuerebene in Ihrem Namen aktualisieren. Für eine selbstverwaltete Knotengruppe gibt es in der Konsole keinen Hinweis darauf, dass sie aktualisiert werden muss. Sie können die auf einem Knoten installierte `kubelet`-Version anzeigen, indem Sie den Knoten in der Liste Knoten auf der Registerkarte Übersicht Ihres Clusters auswählen, um zu bestimmen, welche Knoten aktualisiert werden müssen. Sie müssen die Knoten manuell aktualisieren. Weitere Informationen finden Sie unter Aktualisierungen des selbstverwalteten Worker-Knotens.

Das Kubernetes-Projekt testet die Kompatibilität zwischen der Steuerebene und den Knoten für bis zu drei Nebenversionen. `1.27`-Knoten funktionieren beispielsweise weiterhin, wenn sie von einer `1.30`-Steuerebene orchestriert werden. Es wird jedoch nicht empfohlen, einen Cluster mit Knoten auszuführen, die sich dauerhaft drei Nebenversionen hinter der Steuerebene befinden. Weitere Informationen finden Sie unter Kubernetes-Version und Richtlinie zur Unterstützung der Versionsverzerrung in der Kubernetes-Dokumentation. Wir empfehlen, dieselbe Kubernetes-Version auf Ihrer Steuerebene und Ihren Knoten beizubehalten.

Werden Pods, die auf Fargate ausgeführt werden, automatisch mit einem automatischen Upgrade der Cluster-Steuerebenenversion aktualisiert?

Nein. Wir empfehlen dringend, Fargate-Pods als Teil eines Replikationscontrollers wie einer Kubernetes-Bereitstellung auszuführen. Dann führen Sie einen rollenden Neustart aller Fargate-

Pods durch. Die neue Version des Fargate-Pod wird mit einer `kubelet`-Version bereitgestellt, die dieselbe Version wie Ihre aktualisierte Version der Cluster-Steuerebene ist. Weitere Informationen finden Sie unter Deployments (Bereitstellungen) in der Kubernetes-Dokumentation.

> **Important**
>
> Wenn Sie die Steuerebene aktualisieren, müssen Sie die Fargate-Knoten nach wie vor selbst aktualisieren. Um Fargate-Knoten zu aktualisieren, löschen Sie den Fargate-Pod, der durch den Knoten repräsentiert wird, und stellen Sie den Pod erneut bereit. Der neue Pod wird mit einer `kubelet`-Version bereitgestellt, die der Version Ihres Clusters entspricht.

Häufig gestellte Fragen zum erweiterten Support von Amazon EKS

Die Begriffe Standard-Support und verlängerter Support sind mir neu. Was bedeuten diese Begriffe?

Der Standard-Support für eine Kubernetes-Version in Amazon EKS beginnt mit der Veröffentlichung einer Kubernetes-Version auf Amazon EKS und endet 14 Monate nach dem Release-Datum. Der verlängerte Support für eine Kubernetes-Version beginnt unmittelbar nach dem Ende des Standard-Supports und endet nach 12 Monaten. Beispielsweise endet der Standard-Support für die Version `1.23` in Amazon EKS am 11. Oktober 2023. Der erweiterte Support für die Version `1.23` begann am 12. Oktober 2023 und endet am 11. Oktober 2024.

Was muss ich tun, um verlängerten Support für Amazon-EKS-Cluster zu erhalten?

Sie müssen nichts tun, um verlängerten Support für Ihre Amazon-EKS-Cluster zu erhalten. Der Standard-Support beginnt mit der Veröffentlichung einer Kubernetes-Version auf Amazon EKS und endet 14 Monate nach dem Release-Datum. Der verlängerte Support für eine Kubernetes-Version beginnt unmittelbar nach dem Ende des Standard-Supports und endet nach 12 Monaten. Cluster, die auf einer Kubernetes-Version laufen, deren Standard-Support abgelaufen ist, werden automatisch in den verlängerten Support aufgenommen.

Für welche Kubernetes-Versionen kann ich verlängerten Support erhalten?

Verlängerter Support ist für Kubernetes-Versionen `1.23` und höher verfügbar. Sie können Cluster auf jeder Version bis zu 12 Monate nach dem Ende des Standard-Supports für diese Version ausführen. Das bedeutet, dass jede Version 26 Monate lang in Amazon EKS unterstützt wird (14 Monate Standard-Support plus 12 Monate verlängerter Support).

Was ist, wenn ich den verlängerten Support nicht nutzen möchte?

Wenn Sie nicht automatisch für den verlängerten Support registriert werden möchten, können Sie Ihren Cluster auf eine Kubernetes-Version aktualisieren, für die der Amazon-EKS-Standard-Support verfügbar ist. Cluster, die nicht auf eine Kubernetes-Version mit Standard-Support aktualisiert wurden, werden automatisch in den verlängerten Support aufgenommen.

Was passiert nach Ablauf des 12-monatigen verlängerten Supports?

Cluster, die auf einer Kubernetes-Version ausgeführt werden und am Ende des 26-monatigen Lebenszyklus (14 Monate Standard-Support plus 12 Monate verlängerter Support) angekommen sind, werden automatisch auf die nächste Version aktualisiert.

Nach Ablauf des verlängerten Supports können Sie mit der nicht unterstützten Version keine neuen Amazon-EKS-Cluster mehr erstellen. Vorhandene Steuerebenen werden von Amazon EKS durch einen schrittweisen Bereitstellungsprozess nach dem Datum des Support-Laufzeitendes automatisch auf die älteste unterstützte Version aktualisiert. Nach der automatischen Aktualisierung der Steuerebene müssen Sie Cluster-Add-ons und Amazon-EC2-Knoten manuell aktualisieren. Weitere Informationen finden Sie unter Die Kubernetes-Version für Ihren Amazon-EKS-Cluster aktualisieren.

Wann genau wird meine Steuerebene nach Ablauf des verlängerten Supports automatisch aktualisiert?

Amazon EKS kann keine spezifischen Zeitrahmen bereitstellen. Automatische Updates können jederzeit nach Ablauf des verlängerten Supports erfolgen. Sie erhalten vor dem Update keine Benachrichtigung. Wir empfehlen Ihnen, dass Sie Ihre Steuerebene proaktiv aktualisieren, ohne sich auf den automatischen Aktualisierungsprozess von Amazon EKS zu verlassen. Weitere Informationen finden Sie unter Aktualisieren einer Amazon-EKS-Cluster-Kubernetes-Version.

Kann ich meine Steuerebene auf unbestimmte Zeit auf einer Kubernetes-Version belassen?

Nein. Cloud-Sicherheit AWS hat höchste Priorität. Nach einem bestimmten Punkt (normalerweise einem Jahr) hört die Kubernetes-Community auf, CVE-Patches (Common Vulnerabilities and Exposures, häufige Schwachstellen und Risiken) zu veröffentlichen, und rät von der CVE-Einreichung für nicht unterstützte Versionen ab. Das bedeutet, dass Schwachstellen, die für eine ältere Version von Kubernetes spezifisch sind, möglicherweise nicht einmal gemeldet werden. Daher können Cluster im Falle einer Schwachstelle ohne Vorankündigung und ohne Behebungsoptionen offengelegt werden. Amazon EKS lässt daher nicht zu, dass Steuerebenen auf einer Version verbleiben, die das Ende des verlängerten Supports erreicht hat.

Fallen zusätzliche Kosten für den verlängerten Support an?

Ja, es fallen zusätzliche Kosten für Amazon EKS-Cluster an, die im erweiterten Support ausgeführt werden. Preisinformationen finden Sie im AWS Blog unter Amazon EKS Extended Support für Kubernetes Versionspreise.

Was ist im verlängerten Support enthalten?

Amazon-EKS-Cluster im verlängerten Support erhalten fortlaufend Sicherheitspatches für die Kubernetes-Steuerebene. Darüber hinaus wird Amazon EKS Patches für Amazon VPC CNI, `kube-proxy` und CoreDNS-Add-Ons für verlängerte Supportversionen veröffentlichen. Amazon EKS wird auch Patches für AWS veröffentlichte Amazon EKS-optimierte AMIs für Amazon Linux und Windows sowie für Amazon EKS Fargate-Knoten für diese Versionen veröffentlichen. Bottlerocket Alle Cluster im erweiterten Support erhalten weiterhin Zugriff auf technischen Support von AWS.

> **Note**
>
> Erweiterter Support für Amazon Windows EKS-optimierte AMIs, die von veröffentlicht wurden, ist AWS nicht für Kubernetes Version, `1.23` aber für Kubernetes Version `1.24` und höher verfügbar.

Gibt es im Rahmen des verlängerten Supports Einschränkungen bei Patches für Komponenten, die keine Kubernetes-Komponenten sind?

Der erweiterte Support deckt zwar alle Kubernetes spezifischen Komponenten von ab AWS, bietet aber jederzeit nur Unterstützung für AWS veröffentlichte Amazon EKS-optimierte AMIs für Amazon Linux und Windows. Bottlerocket Das bedeutet, dass Sie möglicherweise neuere Komponenten (wie Betriebssystem oder Kernel) auf Ihrem für Amazon EKS optimierten AMI haben werden, während Sie den verlängerten Support verwenden. Sobald Amazon Linux 2 beispielsweise im Jahr 2025 das Ende seines Lebenszyklus erreicht, werden die für Amazon EKS optimierten Amazon-Linux-AMIs mit einem neueren Amazon-Linux-Betriebssystem erstellt. Amazon EKS wird wichtige Abweichungen vom Support-Lebenszyklus wie diese für jede Kubernetes-Version bekannt geben und dokumentieren.

Kann ich mit einer Version mit erweitertem Support neue Cluster erstellen?

Ja, mit Ausnahme von `1.22` und`1.21`. Sie können beispielsweise einen `1.23` Cluster erstellen, aber keinen `1.22` Cluster.

Versionshinweise für Standard-Supportversionen

Dieses Thema erläutert wichtige Änderungen, die Sie für jede Kubernetes-Version des Standard-Supports beachten sollten. Überprüfen Sie beim Upgrade sorgfältig die Änderungen, die zwischen der alten und der neuen Version für Ihren Cluster vorgenommen wurden.

> **Note**
>
> Ab Einführung der `1.24`-Version enthalten die offiziell veröffentlichten Amazon-EKS-AMIs `containerd` als einzige Laufzeit. Kubernetes-Versionen, die älter sind als `1.24`, verwenden Docker als Standard-Laufzeit. Diese Versionen verfügen über eine Bootstrap-Flag-Option, mit der Sie Ihre Workloads auf jedem unterstützten Cluster mit `containerd` testen können. Weitere Informationen finden Sie unter Amazon EKS hat die Unterstützung für `Dockershim` eingestellt.

Kubernetes1,30

Kubernetes `1.30` ist nun in Amazon EKS verfügbar. Weitere Informationen zu Kubernetes `1.30` finden Sie in der offiziellen Versionsankündigung.

> **Important**
>
> - Ab der Amazon EKS-Version `1.30` oder neuer verwenden alle neu erstellten verwalteten Knotengruppen automatisch standardmäßig Amazon Linux 2023 (AL2023) als Knotenbetriebssystem. Bisher wurde für neue Knotengruppen standardmäßig Amazon Linux 2 (AL2) verwendet. Sie können AL2 weiterhin verwenden, indem Sie es beim Erstellen einer neuen Knotengruppe als AMI-Typ auswählen.
>   - Weitere Informationen zu Amazon Linux finden Sie unter Comparing AL2 and AL2023 im Amazon Linux-Benutzerhandbuch.
>   - Weitere Informationen zur Angabe des Betriebssystems für eine verwaltete Knotengruppe finden Sie unter. Erstellen einer verwalteten Knotengruppe

- Mit Amazon EKS `1.30` wird das `topology.k8s.aws/zone-id` Label zu Worker-Knoten hinzugefügt. Sie können Availability Zone IDs (AZ-IDs) verwenden, um den Standort von Ressourcen in einem Konto im Verhältnis zu den Ressourcen in einem anderen Konto zu

bestimmen. Weitere Informationen finden Sie im AWS RAM Benutzerhandbuch unter Availability Zone IDs für Ihre AWS Ressourcen.

- Ab sofort enthält Amazon EKS die `default` Anmerkung zur gp2 `StorageClass` Ressource`1.30`, die auf neu erstellte Cluster angewendet wurde, nicht mehr. Dies hat keine Auswirkungen, wenn Sie diese Speicherklasse namentlich referenzieren. Sie müssen Maßnahmen ergreifen, wenn Sie sich auf einen Standard `StorageClass` im Cluster verlassen haben. Sie sollten mit `StorageClass` dem Namen auf die verweisengp2. Alternativ können Sie die von Amazon EBS empfohlene Standardspeicherklasse bereitstellen, indem Sie den `defaultStorageClass.enabled` Parameter bei der Installation `v1.31.0` oder später von auf true setzen. `aws-ebs-csi-driver add-on`
- Die mindestens erforderliche IAM-Richtlinie für die Amazon EKS-Cluster-IAM-Rolle hat sich geändert. Die Aktion `ec2:DescribeAvailabilityZones` ist erforderlich. Weitere Informationen finden Sie unter Amazon-EKS-Cluster-IAM-Rolle.

Das vollständige Kubernetes `1.30`-Änderungsprotokoll finden Sie unter https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.30.md.

Kubernetes1.29

Kubernetes `1.29` ist nun in Amazon EKS verfügbar. Weitere Informationen zu Kubernetes `1.29` finden Sie in der offiziellen Versionsankündigung.

> **Important**
>
> - Die veraltete `flowcontrol.apiserver.k8s.io/v1beta2` API-Version von `FlowSchema` und `PriorityLevelConfiguration` wird in nicht mehr bereitgestellt. Kubernetes `v1.29` Wenn Sie über Manifeste oder Clientsoftware verfügen, die die veraltete Beta-API-Gruppe verwendet, sollten Sie diese ändern, bevor Sie ein Upgrade auf durchführen. `v1.29`

- Das `.status.kubeProxyVersion` Feld für Knotenobjekte ist jetzt veraltet, und das Kubernetes Projekt schlägt vor, dieses Feld in einer future Version zu entfernen. Das veraltete Feld ist nicht korrekt und wurde in der Vergangenheit von einer Person verwaltet, `kubelet` die weder die `kube-proxy` Version noch weiß, ob sie ausgeführt wird. `kube-proxy` Wenn Sie dieses Feld in der Client-Software verwendet haben, hören Sie auf. Die Informationen sind nicht zuverlässig und das Feld ist jetzt veraltet.

- Kubernetes`1.29`Um die potenzielle Angriffsfläche zu verringern, kennzeichnet die `LegacyServiceAccountTokenCleanUp` Funktion alte automatisch generierte geheime Token als ungültig, wenn sie lange Zeit nicht verwendet wurden (standardmäßig 1 Jahr), und entfernt sie automatisch, wenn sie nach ihrer Markierung als ungültig lange Zeit nicht versucht werden (standardmäßig 1 zusätzliches Jahr). Um solche Token zu identifizieren, können Sie Folgendes ausführen:

```
kubectl get cm kube-apiserver-legacy-service-account-token-tracking -nkube-system
```

Das vollständige Kubernetes `1.29`-Änderungsprotokoll finden Sie unter https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.29.md#changelog-since-v1280.

Kubernetes1.28

Kubernetes `1.28` ist nun in Amazon EKS verfügbar. Weitere Informationen zu Kubernetes `1.28` finden Sie in der offiziellen Versionsankündigung.

- Kubernetes `v1.28` erweiterte den unterstützten Unterschied zwischen den Komponenten des Core-Knotens und der Steuerebene um eine Nebenversion, von `n-2` bis `n-3`, sodass die Knotenkomponenten (`kubelet` und `kube-proxy`) für die älteste unterstützte Nebenversion mit den Steuerebenenkomponenten (`kube-apiserver`, `kube-scheduler`, `kube-controller-manager`, `cloud-controller-manager`) für die neueste unterstützte Nebenversion zusammenarbeiten können.
- Die Metriken `force_delete_pods_total` und `force_delete_pod_errors_total` im `Pod GC Controller` wurden dahingehend verbessert, dass alle erzwungen Löschungen von Pods berücksichtigt werden. Der Metrik wird ein Grund hinzugefügt, der angibt, ob der Pod gewaltsam gelöscht wurde, weil er beendet wurde, verwaist ist, mit dem out-of-service Makel endet oder weil er beendet und nicht geplant wurde.
- Der `PersistentVolume (PV)`-Controller wurde dahingehend geändert, dass allen ungebundenen `PersistentVolumeClaim`, bei denen der `storageClassName` nicht gesetzt ist, automatisch eine standardmäßige `StorageClass` zugewiesen wird. Darüber hinaus wurde der Mechanismus zur Überprüfung der Zulassung des `PersistentVolumeClaim` innerhalb des API-Servers so angepasst, dass Werte von einem nicht festgelegten Status in einen tatsächlichen `StorageClass` -Namen geändert werden können.

Das vollständige Kubernetes `1.28`-Änderungsprotokoll finden Sie unter https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.28.md#changelog-since-v1270.

Kubernetes1.27

Kubernetes `1.27` ist nun in Amazon EKS verfügbar. Weitere Informationen zu Kubernetes `1.27` finden Sie in der offiziellen Versionsankündigung.

> **Important**
>
> - Die Unterstützung für die Alpha-`seccomp`-Anmerkungen und die Anmerkungen `seccomp.security.alpha.kubernetes.io/pod` und `container.seccomp.security.alpha.kubernetes.io` wurde entfernt. Die Alpha-`seccomp`-Anmerkungen sind seit `1.19` veraltet. Aufgrund ihrer Entfernung in `1.27` werden `seccomp`-Felder für `Pods` nicht mehr automatisch mit `seccomp`-Anmerkungen gefüllt. Verwenden Sie stattdessen das Feld `securityContext.seccompProfile` für `Pods` oder Container, um `seccomp`-Profile zu konfigurieren. Um zu überprüfen, ob Sie die veralteten Alpha-`seccomp`-Anmerkungen im Cluster verwenden, führen Sie den folgenden Befehl aus:
>
>   ```
>   kubectl get pods --all-namespaces -o json | grep
>    -E 'seccomp.security.alpha.kubernetes.io/pod|
>   container.seccomp.security.alpha.kubernetes.io'
>   ```
>
> - Das Befehlszeilenargument `--container-runtime` für das `kubelet` wurde entfernt. `containerd`Seitdem ist die Standard-Container-Laufzeit für Amazon EKS gültig`1.24`, sodass die Container-Laufzeit nicht mehr angegeben werden muss. Ab `1.27` ignoriert Amazon EKS das Argument `--container-runtime`, das an Bootstrap-Skripte übergeben wird. Um Fehler beim Bootstrap-Prozess von Knoten zu vermeiden, dürfen Sie dieses Argument auf keinen Fall an `--kubelet-extra-args` übergeben. Sie müssen das Argument `--container-runtime` aus allen Workflows und Build-Skripten zur Knotenerstellung entfernen.

- Durch das `kubelet` in Kubernetes `1.27` wurde der Standardwert für `kubeAPIQPS` auf `50` und `kubeAPIBurst` auf `100` erhöht. Dank dieser Verbesserungen kann das `kubelet` eine größere Menge an API-Abfragen verarbeiten, wodurch die Antwortzeiten und die Leistung verbessert werden. Wenn der Bedarf an `Pods` aufgrund von Skalierungsanforderungen steigt, stellen die

überarbeiteten Standardwerte sicher, dass das `kubelet` die höhere Workload effizient bewältigen kann. Infolgedessen sind Pod-Starts schneller und Clustervorgänge effektiver.

- Zur Verteilung von Richtlinien wie `minDomain` können Sie eine detailliertere Pod-Topologie verwenden. Mit diesem Parameter können Sie die Mindestanzahl von Domains angeben, auf die die `Pods` verteilt werden sollen. `nodeAffinityPolicy` und `nodeTaintPolicy` ermöglichen ein zusätzliches Maß an Granularität bei der Steuerung der Pod-Verteilung. Dies entspricht den Knotenaffinitäten, den Taints und dem Feld `matchLabelKeys` in den `topologySpreadConstraints` der Spezifikation Ihres `Pod's`. Das ermöglicht die Auswahl von `Pods` für Verteilungsberechnungen nach einem fortlaufenden Upgrade.
- Kubernetes `1.27` hat einen neuen Richtlinienmechanismus für `StatefulSets`, der die Lebensdauer der `PersistentVolumeClaims` (`PVCs`) steuert, zur Beta-Version hochgestuft. Mit der neuen `PVC`-Aufbewahrungsrichtlinie können Sie angeben, ob die anhand der Spezifikationsvorlage `StatefulSet` generierten `PVCs` automatisch gelöscht oder beibehalten werden, wenn das `StatefulSet` gelöscht wird oder die Replikate im `StatefulSet` herunterskaliert werden.
- Die `goaway-chance`-Option im Kubernetes API-Server verhindert, dass `HTTP/2` Client-Verbindungen auf einer einzelnen API-Server-Instance hängen bleiben, indem eine Verbindung nach dem Zufallsprinzip geschlossen wird. Wenn die Verbindung geschlossen wird, versucht der Client erneut, eine Verbindung herzustellen, und landet wahrscheinlich aufgrund des Load Balancers auf einem anderen API-Server. Die Amazon-EKS-Version `1.27` hat das `goaway-chance`-Flag aktiviert. Wenn Ihr auf dem Amazon-EKS-Cluster ausgeführter Workload einen Client verwendet, der mit `HTTP GOAWAY` inkompatibel ist, empfehlen wir, dass Sie Ihren Client entsprechend aktualisieren, die Verbindung nach Beendigung der Verbindung erneut herzustellen, indem `GOAWAY` richtig verarbeitet wird.

Das vollständige Kubernetes `1.27`-Änderungsprotokoll finden Sie unter https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.27.md#changelog-since-v1260.

Kubernetes1.26

Kubernetes `1.26` ist nun in Amazon EKS verfügbar. Weitere Informationen zu Kubernetes `1.26` finden Sie in der offiziellen Versionsankündigung.

> **Important**
>
> Kubernetes `1.26` unterstützt CRI `v1alpha2` nicht mehr. Dies führt dazu, dass das `kubelet` den Knoten nicht mehr registriert, wenn die Container-Laufzeit CRI `v1` nicht unterstützt.

Das bedeutet auch, dass Kubernetes `1.26` Containerd-Minor-Version `1.5` und frühere Versionen nicht unterstützt. Wenn Sie Containerd verwenden, müssen Sie ein Upgrade auf die Containerd-Version `1.6.0` oder eine neuere Version durchführen, bevor Sie Knoten auf Kubernetes `1.26` aktualisieren. Sie müssen auch alle anderen Container-Laufzeiten aktualisieren, die nur die `v1alpha2` unterstützen. Weitere Informationen erhalten Sie beim Anbieter der Container-Laufzeit. Standardmäßig enthalten Amazon Linux- und Bottlerocket-AMIs die Containerd-Version `1.6.6`.

- Bevor Sie ein Upgrade auf Kubernetes `1.26` durchführen, aktualisieren Sie Ihre Version Amazon VPC CNI plugin for Kubernetes auf Version `1.12` oder höher. Wenn Sie kein Upgrade auf Amazon VPC CNI plugin for Kubernetes-Version `1.12` oder höher durchführen, wird Amazon VPC CNI plugin for Kubernetes abstürzen. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.
- Die `goaway-chance`-Option im Kubernetes API-Server verhindert, dass `HTTP/2` Client-Verbindungen auf einer einzelnen API-Server-Instance hängen bleiben, indem eine Verbindung nach dem Zufallsprinzip geschlossen wird. Wenn die Verbindung geschlossen wird, versucht der Client erneut, eine Verbindung herzustellen, und landet wahrscheinlich aufgrund des Load Balancers auf einem anderen API-Server. Die Amazon-EKS-Version `1.26` hat das `goaway-chance`-Flag aktiviert. Wenn Ihr auf dem Amazon-EKS-Cluster ausgeführter Workload einen Client verwendet, der mit `HTTP GOAWAY` inkompatibel ist, empfehlen wir, dass Sie Ihren Client entsprechend aktualisieren, die Verbindung nach Beendigung der Verbindung erneut herzustellen, indem `GOAWAY` richtig verarbeitet wird.

Das vollständige Kubernetes `1.26`-Änderungsprotokoll finden Sie unter https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.26.md#changelog-since-v1250.

Versionshinweise für Versionen mit verlängerten Support

Dieses Thema erläutert wichtige Änderungen, die Sie für jede Kubernetes-Version des verlängerten Supports beachten sollten. Überprüfen Sie beim Upgrade sorgfältig die Änderungen, die zwischen der alten und der neuen Version für Ihren Cluster vorgenommen wurden.

Kubernetes1,25

Kubernetes `1.25` ist nun in Amazon EKS verfügbar. Weitere Informationen zu Kubernetes `1.25` finden Sie in der offiziellen Versionsankündigung.

Important

- Ab Kubernetes-Version `1.25` können Sie Amazon-EC2-P2-Instances nicht mehr mit den Amazon-EKS-optimierten, beschleunigten Amazon-Linux-AMIs verwenden, die sofort einsatzbereit sind. Diese AMIs für Kubernetes-Versionen `1.25` oder höher werden Treiber der `NVIDIA 525`-Serie oder höher unterstützen, die mit den `P2`-Instances nicht kompatibel sind. Die Treiber der Serie `NVIDIA 525` oder höher sind jedoch mit den `P3`-, `P4`- und `P5`-Instances kompatibel, sodass Sie diese Instances mit den AMIs für die Kubernetes-Version `1.25` oder höher verwenden können. Bevor Ihre Amazon-EKS-Cluster auf Version `1.25` aktualisiert werden, migrieren Sie alle `P2`-Instances zu `P3`-, `P4`- und `P5`-Instances. Sie sollten Ihre Anwendungen auch proaktiv aktualisieren, damit sie mit der `NVIDIA 525`-Serie oder höher funktionieren. Wir planen, die Treiber der neueren `NVIDIA 525` Serie oder neuer auf die Kubernetes Versionen `1.23` und `1.24` Ende Januar 2024 zurückzuportieren.
- Die `PodSecurityPolicy` (PSP) wird in Kubernetes `1.25` entfernt. PSPs werden durch Pod Security Admission (PSA) und Pod Security Standards ersetzt (PSS). PSA ist ein integrierter Zulassungscontroller, der die Sicherheitskontrollen verwendet, die in der PSS beschrieben sind. PSA und PSS sind in Kubernetes `1.25` auf stabil abgestuft und in Amazon EKS standardmäßig aktiviert. Wenn Sie bereits PSPs in Ihrem Cluster sind, stellen Sie sicher, dass Sie von PSP der integrierten Version Kubernetes PSS oder zu einer policy-as-code Lösung migrieren, bevor Sie Ihren Cluster auf Version aktualisieren`1.25`. Wenn Sie nicht von PSP migrieren, kann es zu Unterbrechungen Ihrer Workloads kommen. Weitere Informationen hierzu finden Sie unter Entfernen der Pod-Sicherheitsrichtlinie (PSP) – Häufig gestellte Fragen.
- Kubernetes-Version `1.25` enthält Änderungen, die das Verhalten eines bestehenden Features ändern, das als API-Priorität und Fairness (APF) bekannt ist. APF dient dazu, den API-Server in Zeiten erhöhten Anforderungsvolumens vor einer möglichen Überlastung zu schützen. Dies geschieht, indem die Anzahl der gleichzeitigen Anfragen, die zu einem bestimmten Zeitpunkt bearbeitet werden können, begrenzt wird. Es wird durch die Anwendung unterschiedlicher Prioritätsstufen und Grenzwerte für Anfragen erreicht, die von verschiedenen Workloads oder Benutzern stammen. Dieser Ansatz stellt sicher, dass kritische Anwendungen oder Anfragen mit hoher Priorität bevorzugt behandelt werden, während gleichzeitig verhindert wird, dass Anfragen mit niedrigerer Priorität den API-Server überlasten. Weitere Informationen finden Sie unter API-Priorität und Fairness in der Kubernetes-Dokumentation oder API-Priorität und Fairness im EKS-Leitfaden für bewährte Methoden.

Diese Updates wurden eingeführt in PR #10352 und PR #118601. Bisher behandelte APF alle Arten von Anfragen einheitlich, wobei jede Anfrage eine einzige Einheit des Limits für gleichzeitige Anfragen verbrauchte. Die APF-Verhaltensänderung weist `LIST`-Anforderungen höhere Parallelitätseinheiten zu, da der API-Server durch diese Anforderungen außergewöhnlich stark belastet wird. Der API-Server schätzt die Anzahl der Objekte, die bei einer `LIST`-Anforderung zurückgegeben werden. Er weist eine Parallelitätseinheit zu, die proportional zur Anzahl der zurückgegebenen Objekte ist.

Nach dem Upgrade auf die Amazon-EKS-Version `1.25` oder höher, kann dieses aktualisierte Verhalten zu Workloads mit massiven `LIST`-Anfragen (die zuvor problemlos funktionierten) führen, die auf eine Ratenbegrenzung zu stoßen. Dies würde durch einen HTTP-429-Antwortcode angezeigt werden. Um mögliche Workloadunterbrechungen aufgrund von `LIST` zu vermeiden, da die Anzahl der Anfragen begrenzt ist, empfehlen wir Ihnen dringend, Ihre Workloads umzustrukturieren, um die Anzahl dieser Anfragen zu reduzieren. Alternativ können Sie dieses Problem beheben, indem Sie die APF-Einstellungen anpassen, um mehr Kapazität für wichtige Anfragen zuzuweisen und gleichzeitig die Kapazität für nicht wichtige Anfragen zu reduzieren. Weitere Informationen zu diesen Risikominderungstechniken finden Sie unter Verhinderung verworfener Anfragen im EKS-Leitfaden für bewährte Methoden.

- Amazon EKS `1.25` umfasst Verbesserungen der Cluster-Authentifizierung, die aktualisierte YAML-Bibliotheken enthalten. Wenn ein YAML-Wert in der `aws-auth ConfigMap` im `kube-system`-Namespace mit einem Makro beginnt, wobei das erste Zeichen eine geschweifte Klammer ist, sollten Sie Anführungszeichen (`" "`) vor und nach den geschweiften Klammern (`{ }`) hinzufügen. Dies ist erforderlich, um sicherzustellen, dass die `aws-iam-authenticator`-Version `v0.6.3` präzise die `aws-auth ConfigMap` in Amazon EKS `1.25` analysiert.
- Die Beta-API-Version (`discovery.k8s.io/v1beta1`) von `EndpointSlice` war veraltet in Kubernetes `1.21` und wird ab Kubernetes `1.25` nicht mehr bereitgestellt. Diese API wurde zu `discovery.k8s.io/v1` aktualisiert. Weitere Informationen finden Sie unter `EndpointSlice` in der Kubernetes-Dokumentation. AWS Load Balancer Controller `v2.4.6` und früher hat den `v1beta1`-Endpunkt verwendet, um mit `EndpointSlices` zu kommunizieren. Wenn Sie die `EndpointSlices`-Konfiguration für den AWS Load Balancer Controller verwenden, müssen Sie ein Upgrade auf AWS Load Balancer Controller `v2.4.7` durchführen, bevor Sie Ihren Amazon-EKS-Cluster auf `1.25` aktualisieren. Wenn Sie ein Upgrade auf `1.25` durchführen, während Sie die `EndpointSlices`-Konfiguration für den AWS Load Balancer Controller verwenden, stürzt

der Controller ab und es kommt zu Unterbrechungen Ihrer Workloads. Informationen zum Upgrade des Controllers finden Sie unter Was ist die AWS Load Balancer Controller?.

- `SeccompDefault` wird in Kubernetes `1.25` zur Betaversion hochgestuft. Wenn Sie die `--seccomp-default`-Markierung bei der Konfiguration von `kubelet` festlegen, verwendet die Container-Laufzeit ihr `RuntimeDefault` `seccomp`-Profil und nicht den Modus „uneingeschränkt“ (`seccomp disabled`). Die Standardprofile bieten eine Reihe starker Sicherheitsstandards und behalten gleichzeitig die Funktionalität des Workloads bei. Obwohl diese Markierung verfügbar ist, aktiviert Amazon EKS diese Markierung standardmäßig nicht, sodass das Verhalten von Amazon EKS praktisch unverändert bleibt. Wenn Sie möchten, können Sie damit beginnen, dies auf Ihren Knoten zu aktivieren. Weitere Informationen finden Sie im Tutorial Systemaufrufe mit Seccomp eines Containers beschränken in der Kubernetes-Dokumentation.
- Der Support für das Container Runtime Interface (CRI) für Docker (auch bekannt als `Dockershim`) wurde ab Kubernetes `1.24` entfernt. Die einzige offizielle Container-Laufzeit in Amazon EKS AMIs für Kubernetes `1.24` und spätere Cluster ist `containerd`. Bevor Sie zu Amazon EKS `1.24` oder höher wechseln, müssen Sie alle Verweise auf Bootstrap-Skript-Flags entfernen, die nicht mehr unterstützt werden. Weitere Informationen finden Sie unter Amazon EKS hat die Unterstützung für `Dockershim` eingestellt.
- Die Unterstützung für Platzhalterabfragen war in CoreDNS `1.8.7` veraltet und wurde in CoreDNS `1.9` entfernt. Dies wurde als Sicherheitsmaßnahme durchgeführt. Platzhalterabfragen funktionieren nicht mehr und geben `NXDOMAIN` anstelle einer IP-Adresse zurück.
- Die `goaway-chance`-Option im Kubernetes API-Server verhindert, dass `HTTP/2` Client-Verbindungen auf einer einzelnen API-Server-Instance hängen bleiben, indem eine Verbindung nach dem Zufallsprinzip geschlossen wird. Wenn die Verbindung geschlossen wird, versucht der Client erneut, eine Verbindung herzustellen, und landet wahrscheinlich aufgrund des Load Balancers auf einem anderen API-Server. Die Amazon-EKS-Version `1.25` hat das `goaway-chance`-Flag aktiviert. Wenn Ihr auf dem Amazon-EKS-Cluster ausgeführter Workload einen Client verwendet, der mit `HTTP GOAWAY` inkompatibel ist, empfehlen wir, dass Sie Ihren Client entsprechend aktualisieren, die Verbindung nach Beendigung der Verbindung erneut herzustellen, indem `GOAWAY` richtig verarbeitet wird.

Das vollständige Kubernetes `1.25`-Änderungsprotokoll finden Sie unter https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.25.md#changelog-since-v1240.

Kubernetes1.24

Kubernetes `1.24` ist nun in Amazon EKS verfügbar. Weitere Informationen zu Kubernetes `1.24` finden Sie in der offiziellen Versionsankündigung.

Important

- Ab Kubernetes `1.24` sind neue Beta-APIs standardmäßig nicht in Clustern aktiviert. Standardmäßig sind bestehende Beta-APIs und neue Versionen vorhandener Beta-APIs weiterhin aktiviert. Amazon EKS folgt demselben Verhalten wie Upstream Kubernetes `1.24`. Die Feature-Gates, die neue Features sowohl für neue als auch für bestehende API-Operationen steuern, sind standardmäßig aktiviert. Dies entspricht dem Upstream Kubernetes. Weitere Informationen finden Sie unter KEP-3136: Beta-APIs sind standardmäßig ausgeschaltet. GitHub
- Der Support für Container Runtime Interface (CRI) für Docker (auch bekannt als Dockershim) wurde von Kubernetes `1.24` entfernt. Offizielle Amazon-EKS-AMIs haben containerd als einzige Laufzeit. Bevor Sie zu Amazon EKS `1.24` oder höher wechseln, müssen Sie alle Verweise auf Bootstrap-Skript-Flags entfernen, die nicht mehr unterstützt werden. Sie müssen außerdem sicherstellen, dass die IP-Weiterleitung für Ihre Worker-Knoten aktiviert ist. Weitere Informationen finden Sie unter Amazon EKS hat die Unterstützung für `Dockershim` eingestellt.
- Wenn Fluentd für Container Insights bereits konfiguriert ist, müssen Sie Fluentd zu Fluent Bit migrieren, bevor Sie Ihren Cluster aktualisieren. Die Fluentd-Parser sind so konfiguriert, dass sie nur Protokollnachrichten im JSON-Format analysieren. Im Gegensatz zu `dockerd` enthält die Container-Laufzeit `containerd` Protokollnachrichten, die sich nicht im JSON-Format befinden. Wenn Sie nicht zu Fluent Bit migrieren, erzeugen einige der konfigurierten Fluentd's-Parser eine große Anzahl von Fehlern im Container Fluentd. Weitere Informationen zur Migration finden Sie unter Fluent BitSo einrichten, dass Protokolle an Logs DaemonSet gesendet werden. CloudWatch
- In Kubernetes `1.23` und älter werden `kubelet`-Bereitstellungszertifikate mit nicht verifizierbaren IP- und DNS-Subject Alternative Names (SANs) automatisch mit nicht verifizierbaren SANs ausgestellt. Diese nicht verifizierbaren SANs werden im bereitgestellten Zertifikat weggelassen. In Clustern der Version `1.24` und höher werden keine `kubelet`-Bereitstellungszertifikate ausgestellt, wenn kein SAN verifiziert werden kann. Dadurch wird verhindert, dass die Befehle `kubectl`-exec und

`kubectl`-logs funktionieren. Weitere Informationen finden Sie unter Überlegungen zur Zertifikatssignierung vor dem Upgrade Ihres Clusters auf Kubernetes 1.24.

- Wenn Sie einen Amazon-EKS-`1.23`-Cluster aktualisieren, der Fluent Bit verwendet, müssen Sie sicherstellen, dass er unter `k8s/1.3.12` oder neuer läuft. Sie können dies tun, indem Sie die neueste anwendbare Fluent Bit-YAML-Datei von GitHub erneut anwenden. Weitere Informationen finden Sie unter Einrichtung Fluent Bit im CloudWatch Amazon-Benutzerhandbuch.

- Sie können Topology Aware Hints verwenden, um anzugeben, dass Sie es vorziehen, den Datenverkehr in einer Zone zu halten, wenn Cluster-Worker-Knoten über mehrere Availability Zones bereitgestellt werden. Das Routing des Datenverkehrs innerhalb einer Zone kann dazu beitragen, die Kosten zu senken und die Netzwerkleistung zu verbessern. Standardmäßig sind Topology Aware Hints in Amazon EKS `1.24` aktiviert. Weitere Informationen finden Sie unter Topology Aware Hints in der Kubernetes-Dokumentation.
- `PodSecurityPolicy` (PSP) soll in Kubernetes `1.25` entfernt werden. PSPs werden durch Pod Security Admission (PSA) ersetzt. PSA ist ein integrierter Zulassungscontroller, der die Sicherheitskontrollen verwendet, die in den Pod Security Standards beschrieben sind. PSA und PSS sind beide Beta-Features und in Amazon EKS standardmäßig aktiviert. Um das Entfernen von PSP in Version `1.25` anzugehen, empfehlen wir Ihnen, PSS in Amazon EKS zu implementieren. Weitere Informationen finden Sie unter Implementieren von Pod-Sicherheitsstandards in Amazon EKS im AWS -Blog.
- Das `client.authentication.k8s.io/v1alpha1` ExecCredential ist entfernt in Kubernetes`1.24`. Die ExecCredential API war allgemein verfügbar in Kubernetes`1.22`. Wenn Sie ein Plugin für Anmeldeinformationen verwenden, das auf der `v1alpha1`-API basiert, wenden Sie sich an den Vertreiber Ihres Plugins, um zu erfahren, wie Sie zur `v1`-API migrieren können.
- Für Kubernetes `1.24` haben wir ein Feature zum Upstream-Cluster-Autoscaler-Projekt beigetragen, die das Skalieren von Amazon-EKS-verwalteten Knotengruppen auf und von null Knoten vereinfacht. Damit der Cluster-Autoscaler die Ressourcen, Labels und Taints einer verwalteten Knotengruppe, die auf null Knoten skaliert wurde, versteht, mussten Sie zuvor die zugrunde liegende Amazon-EC2-Auto-Scaling-Gruppe mit den Details der Knoten markieren, für die sie verantwortlich war. Wenn es keine ausgeführten Knoten in der verwalteten Knotengruppe gibt, ruft der Cluster-Autoscaler die `DescribeNodegroup`-API-Operation von Amazon EKS auf. Diese API-Operation stellt die Informationen bereit, die der Cluster-Autoscaler zu den Ressourcen, Labels und Taints der verwalteten Knotengruppe benötigt. Für dieses Feature müssen Sie die `eks:DescribeNodegroup`-Berechtigung zur IAM-Richtlinie des Cluster-Autoscaler-Servicekontos

hinzufügen. Wenn der Wert eines Cluster-Autoscaler-Tags in der Auto-Scaling-Gruppe, die eine von Amazon EKS verwaltete Knotengruppe unterstützt, mit der Knotengruppe selbst in Konflikt steht, bevorzugt der Cluster-Autoscaler den Wert des Auto-Scaling-Gruppen-Tags. Auf diese Weise können Sie Werte nach Bedarf überschreiben. Weitere Informationen finden Sie unter Auto Scaling.

- Wenn Sie beabsichtigen, unsere Trainium Instance-Typen mit Amazon EKS zu verwenden Inferentia`1.24`, müssen Sie ein Upgrade auf die AWS Neuron Geräte-Plug-in-Version 1.9.3.0 oder höher durchführen. Weitere Informationen finden Sie in der Dokumentation unter Neuron K8-Version [1.9.3.0]. AWS Neuron
- `Containerd` hat `IPv6` standardmäßig für Pods aktiviert. Es wendet die Knoten-Kernel-Einstellungen auf Pod-Netzwerk-Namespaces an. Aus diesem Grund binden Container in einem Pod sowohl an `IPv4` (`127.0.0.1`) als auch an `IPv6` (`::1`)-Loopback-Adressen an. `IPv6` ist das Standardprotokoll für die Kommunikation. Bevor Sie Ihr Cluster auf Version `1.24` aktualisieren, empfehlen wir Ihnen, Ihre Multi-Container-Pods zu testen. Ändern Sie Apps so, dass sie an alle IP-Adressen auf Loopback-Schnittstellen gebunden werden können. Die meisten Bibliotheken ermöglichen das `IPv6`-Binden, das mit `IPv4` abwärtskompatibel ist. Wenn es nicht möglich ist, Ihren Anwendungscode zu ändern, haben Sie zwei Möglichkeiten:
  - Führen Sie einen `init`-Container aus und setzen Sie `disable ipv6` auf `true` (`sysctl -w net.ipv6.conf.all.disable_ipv6=1`).
  - Konfigurieren Sie einen Webhook mit mutierendem Zugang, um einen `init`-Container neben Ihren Anwendungs-Pods einzufügen.

  Wenn Sie `IPv6` für alle Pods auf allen Knoten blockieren müssen, müssen Sie möglicherweise `IPv6` auf Ihren Instances deaktivieren.
- Die `goaway-chance`-Option im Kubernetes API-Server verhindert, dass `HTTP/2` Client-Verbindungen auf einer einzelnen API-Server-Instance hängen bleiben, indem eine Verbindung nach dem Zufallsprinzip geschlossen wird. Wenn die Verbindung geschlossen wird, versucht der Client erneut, eine Verbindung herzustellen, und landet wahrscheinlich aufgrund des Load Balancers auf einem anderen API-Server. Die Amazon-EKS-Version `1.24` hat das `goaway-chance`-Flag aktiviert. Wenn Ihr auf dem Amazon-EKS-Cluster ausgeführter Workload einen Client verwendet, der mit `HTTP GOAWAY` inkompatibel ist, empfehlen wir, dass Sie Ihren Client entsprechend aktualisieren, die Verbindung nach Beendigung der Verbindung erneut herzustellen, indem `GOAWAY` richtig verarbeitet wird.

Das vollständige Kubernetes `1.24`-Änderungsprotokoll finden Sie unter https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.24.md#changelog-since-v1230.

Kubernetes1.23

Kubernetes `1.23` ist nun in Amazon EKS verfügbar. Weitere Informationen zu Kubernetes `1.23` finden Sie in der offiziellen Versionsankündigung.

Important

- Das Volume-Migrationsfeature von Kubernetes In-Tree zu Container Storage Interface (CSI) ist aktiviert. Dieses Feature ermöglicht es, vorhandene Kubernetes-In-Tree-Speicher-Plugins für Amazon EBS durch einen entsprechenden CSI-Treiber von Amazon EBS zu ersetzen. Weitere Informationen finden Sie unter Kubernetes Feature 1.17: Kubernetes In-Tree to CSI Volume Migration Moves to Beta (Feature von 1.17: Migration von In-Tree- zu CSI-Volumes) im Kubernetes-Blog.

  Das Feature übersetzt In-Tree-APIs in entsprechende CSI-APIs und delegiert Vorgänge an einen Ersatz-CSI-Treiber. Wenn Sie vorhandene `StorageClass`-, `PersistentVolume`- und `PersistentVolumeClaim`-Objekte verwenden, die zu diesen Workloads gehören, werden Sie bei Einsatz dieses Features kaum eine Veränderung feststellen. Das Feature ermöglicht Kubernetes die Delegierung aller Speicherverwaltungsvorgänge vom In-Tree-Plugin an den CSI-Treiber. Wenn Sie in einem vorhandenen Cluster Amazon-EBS-Volumes verwenden, installieren Sie den CSI-Treiber von Amazon EBS im Cluster, bevor Sie den Cluster auf die Version `1.23` aktualisieren. Wenn Sie den Treiber nicht vor der Aktualisierung eines vorhandenen Clusters installieren, kann es zu Unterbrechungen Ihrer Workloads kommen. Wenn Sie Workloads bereitstellen möchten, die Amazon EBS-Volumes in einem neuen `1.23`-Cluster verwenden, installieren Sie den CSI-Treiber von Amazon EBS in Ihrem Cluster, bevor Sie die Workloads in Ihrem Cluster bereitstellen. Anweisungen zum Installieren des CSI-Treibers von Amazon EBS in Ihrem Cluster finden Sie unter Amazon-EBS-CSI-Treiber. Häufig gestellte Fragen zum Migrationsfeature finden Sie unter Häufig gestellte Fragen zur Migration des CSI von Amazon EBS.
- Erweiterter Support für Amazon Windows EKS-optimierte AMIs, die von veröffentlicht wurden, ist AWS nicht für Kubernetes Version, `1.23` aber für Kubernetes Version `1.24` und höher verfügbar.

- Kubernetes hat die Unterstützung von `dockershim` in Version `1.20` eingestellt und `dockershim` in Version `1.24` entfernt. Weitere Informationen finden Sie unter Kubernetes verabschiedet sich von Dockershim: Verpflichtungen und nächste Schritte im Kubernetes-Blog. Amazon EKS wird die Unterstützung für `dockershim` ab der Amazon-EKS-Version `1.24` einstellen. Ab der Amazon-EKS-Version `1.24` werden offizielle AMIs von Amazon EKS `containerd` als einzige Laufzeit aufweisen.

  Auch wenn Amazon-EKS-Version `1.23` weiterhin `dockershim` unterstützt, empfehlen wir Ihnen, jetzt mit dem Testen Ihrer Anwendungen zu beginnen, um alle Docker-Abhängigkeiten zu ermitteln und zu entfernen. So sind Sie bereit, Ihren Cluster auf die Version `1.24` zu aktualisieren. Weitere Informationen über das Entfernen von `dockershim` finden Sie unter Amazon EKS hat die Unterstützung für `Dockershim` eingestellt.
- Kubernetes hat `IPv4`/`IPv6`-Dual-Stack-Networking für Pods, Services und Knoten auf allgemeine Verfügbarkeit hochgestuft. Allerdings unterstützen Amazon EKS und das Amazon VPC CNI plugin for Kubernetes kein Dual-Stack-Networking. Ihre Cluster können `IPv4`- oder `IPv6`-Adressen zu Pods und Services zuweisen, aber nicht beide Adresstypen zuweisen.
- Kubernetes hat das PSA-Feature (Pod Security Admission) auf Beta hochgestuft. Das Feature ist standardmäßig aktiviert. Weitere Informationen finden Sie unter Pod Security Admission in der Kubernetes-Dokumentation. PSA ersetzt den Pod Security Policy (PSP)-Zulassungscontroller. Der PSP-Zugangscontroller wird nicht unterstützt und wird laut Plan in der Kubernetes-Version `1.25` entfernt.

  Der PSP-Zulassungscontroller setzt Pod-Sicherheitsstandards für Pods in einem Namespace basierend auf bestimmten Namespace-Labels durch, die den Grad der Durchsetzung festlegen. Weitere Informationen hierzu finden Sie unter Pod Security Standards (PSS) und Pod Security Admission (PSA) im Leitfaden zu bewährten Methoden für Amazon EKS.
- Das mit Clustern bereitgestellte `kube-proxy`-Image ist jetzt das von Amazon EKS Distro (EKS-D) verwaltete minimale Basis-Image. Das Image enthält nur minimale Pakete und verfügt über keine Shells oder Paketmanager.
- Kubernetes hat flüchtige Container auf Beta hochgestuft. Flüchtige Container sind temporäre Container, die im gleichen Namespace wie ein vorhandener Pod ausgeführt werden. Mit ihrer Hilfe können Sie den Status von Pods und Containern zur Fehlerbehebung und zum Debuggen beobachten. Dies ist insbesondere für die interaktive Fehlerbehebung hilfreich, wenn `kubectl exec` unzureichend ist, weil entweder ein Container abgestürzt ist oder ein Container-Image keine Debugging-Hilfsprogramme enthält. Ein Beispiel für einen Container, der ein Debugging-Dienstprogramm enthält, sind „distroless“ Images. Weitere Informationen finden Sie unter

Debugging with an ephemeral debug container (Debuggen mit einem flüchtigen Debug-Container) in der Kubernetes-Dokumentation.

- Kubernetes hat die stabile `autoscaling/v2`-API von `HorizontalPodAutoscaler` auf allgemeine Verfügbarkeit hochgestuft. Die `HorizontalPodAutoscaler` `autoscaling/v2beta2` API ist veraltet. Es wird in `1.26` nicht verfügbar sein.
- Die `goaway-chance`-Option im Kubernetes API-Server verhindert, dass `HTTP/2` Client-Verbindungen auf einer einzelnen API-Server-Instance hängen bleiben, indem eine Verbindung nach dem Zufallsprinzip geschlossen wird. Wenn die Verbindung geschlossen wird, versucht der Client erneut, eine Verbindung herzustellen, und landet wahrscheinlich aufgrund des Load Balancers auf einem anderen API-Server. Die Amazon-EKS-Version `1.23` hat das `goaway-chance`-Flag aktiviert. Wenn Ihr auf dem Amazon-EKS-Cluster ausgeführter Workload einen Client verwendet, der mit `HTTP GOAWAY` inkompatibel ist, empfehlen wir, dass Sie Ihren Client entsprechend aktualisieren, die Verbindung nach Beendigung der Verbindung erneut herzustellen, indem `GOAWAY` richtig verarbeitet wird.

Das vollständige Kubernetes `1.23`-Änderungsprotokoll finden Sie unter https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.23.md#changelog-since-v1220.

Versionshinweise für die Versionen 1.21 und 1.22

> **Important**
>
> Mit diesen Versionen können Sie keine neuen Cluster erstellen.

Dieses Thema enthält wichtige Änderungen, auf die Sie bei den Versionen `1.22` und achten sollten`1.21`. Überprüfen Sie beim Upgrade sorgfältig die Änderungen, die zwischen der alten und der neuen Version für Ihren Cluster vorgenommen wurden.

Kubernetes-Version **`1.22`**

Die folgenden Zugangscontroller sind für alle Versionen der Plattform `1.22` aktiviert: `DefaultStorageClass`, `DefaultTolerationSeconds`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`, `NodeRestriction`, `ResourceQuota`, `ServiceAccount`, `ValidatingAdmissionWebhook`, `PodSecurityPolicy`, `TaintNodesByCondition`, `StorageObjectInUseProtection`,

`PersistentVolumeClaimResize`, `ExtendedResourceToleration`, `CertificateApproval`, `PodPriority`, `CertificateSigning`, `CertificateSubjectRestriction`, `RuntimeClass` und `DefaultIngressClass`.

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
`1.22.17`	`eks.28`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	16. Mai 2024
`1.22.17`	`eks.26`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	1. April 2024
`1.22.17`	`eks.14`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	30. Juni 2023
`1.22.17`	`eks.13`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	9. Juni 2023
`1.22.17`	`eks.12`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	5. Mai 2023
`1.22.17`	`eks.11`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	24. März 2023
`1.22.16`	`eks.10`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	27. Januar 2023
`1.22.15`	`eks.9`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	5. Dezember 2022
`1.22.15`	`eks.8`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	18. November 2022
`1.22.15`	`eks.7`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. November 2022
`1.22.13`	`eks.6`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	21. September 2022

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.22.10	eks.5	Neue Plattformversion mit verbesserter etcd-Ausfallsicherheit.	15. August 2022
1.22.10	eks.4	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen. Diese Plattformversion führt auch einen neuen Tagging-Controller ein, der alle Worker-Knoten mit aws:eks:cluster-name kennzeichnet, um die Zuweisung von Kosten für diese Worker-Knoten zu vereinfachen. Weitere Informationen finden Sie unter Markieren von Ressourcen für die Fakturierung.	21. Juli 2022
1.22.10	eks.3	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. Juli 2022
1.22.9	eks.2	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	31. Mai 2022
1.22.6	eks.1	Erstversion von Kubernetes 1.22 für Amazon EKS.	4. April 2022

Kubernetes-Version **1.21**

Die folgenden Zugangscontroller sind für alle Versionen der Plattform 1.21 aktiviert: DefaultStorageClass, DefaultTolerationSeconds, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, ResourceQuota, ServiceAccount, ValidatingAdmissionWebhook, PodSecurityPolicy, TaintNodesByCondition, StorageObjectInUseProtection, PersistentVolumeClaimResize, ExtendedResourceToleration, CertificateApproval, PodPriority, CertificateSigning, CertificateSubjectRestriction, RuntimeClass und DefaultIngressClass.

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.21.14	eks.33	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	16. Mai 2024
1.21.14	eks.31	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	1. April 2024
1.21.14	eks.18	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	9. Juni 2023
1.21.14	eks.17	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	5. Mai 2023
1.21.14	eks.16	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	24. März 2023
1.21.14	eks.15	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	27. Januar 2023
1.21.14	eks.14	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	5. Dezember 2022
1.21.14	eks.13	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	18. November 2022

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.21.14	eks.12	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. November 2022
1.21.13	eks.11	Neue Plattformversion mit verbesserter etcd-Ausfallsicherheit.	10. Oktober 2022
1.21.13	eks.10	Neue Plattformversion mit verbesserter etcd-Ausfallsicherheit.	15. August 2022
1.21.13	eks.9	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen. Diese Plattformversion führt auch einen neuen Tagging-Controller ein, der alle Worker-Knoten mit aws:eks:cluster-name kennzeichnet, um die Zuweisung von Kosten für diese Worker-Knoten zu vereinfachen. Weitere Informationen finden Sie unter Markieren von Ressourcen für die Fakturierung.	21. Juli 2022
1.21.13	eks.8	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. Juli 2022
1.21.12	eks.7	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	31. Mai 2022

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.21.9	eks.6	Der AWS Security Token Service Endpunkt wird auf den globalen Endpunkt aus der vorherigen Plattformversion zurückgesetzt. Wenn Sie bei der Verwendung von IAM-Rollen für Servicekonten den regionalen Endpunkt verwenden möchten, müssen Sie ihn aktivieren. Hinweise zum Aktivieren des regionalen Endpunkts finden Sie unter Den AWS Security Token Service Endpunkt für ein Dienstkonto konfigurieren.	8. April 2022
1.21.5	eks.5	Bei Verwendung von IAM-Rollen für Servicekonten wird standardmäßig der regionale Endpunkt AWS Security Token Service anstelle des globalen Endpunkts verwendet. Diese Änderung wird in eks.6 allerdings auf den globalen Endpunkt zurückgesetzt. Ein aktualisierter Fargate-Scheduler stellt Knoten bei großen Bereitstellungen mit deutlich höherer Geschwindigkeit bereit.	10. März 2022

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.21.5	eks.4	Version 1.10.1-eksbuild.1 des selbstverwalteten Amazon VPC CNI und des Amazon-EKS-Add-ons ist jetzt die Standardversion, die bereitgestellt wird.	13. Dezember 2021
1.21.2	eks.3	Neue Plattformversion mit Support für die Windows-IPv4-Adressverwaltung auf dem VPC-Ressourcen-Controller, der auf der Kubernetes-Steuerebene ausgeführt wird. Die Kubernetes-Filterdirektive für die Fargate-Fluent-Bit-Protokollierung wurde hinzugefügt.	8. November 2021
1.21.2	eks.2	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	17. September 2021
1.21.2	eks.1	Erstversion von Kubernetes 1.21 für Amazon EKS.	19. Juli 2021

Amazon-EKS-Plattformversionen

Amazon-EKS-Plattformversionen definieren die Funktionalitäten der Amazon-EKS-Cluster-Steuerebene (z. B. welche Kubernetes-API-Server-Flags aktiviert sind) sowie die aktuelle Kubernetes-Patch-Version. Jede Kubernetes-Minor-Version hat eine oder mehrere zugehörige Amazon-EKS-Plattformversionen. Die Plattformversionen für verschiedene Kubernetes-Minor-Versionen sind voneinander unabhängig. Sie können die aktuelle Plattformversion Ihres Clusters mit dem AWS CLI oder abrufen AWS Management Console. Wenn Sie einen lokalen Cluster aktiviert

haben AWS Outposts, finden Sie Versionen der lokalen Amazon EKS-Clusterplattform statt dieses Themas weitere Informationen.

Wenn eine neue Kubernetes Nebenversion in Amazon EKS verfügbar ist, z. B. 1.30, beginnt die erste Amazon EKS-Plattformversion für diese Kubernetes Nebenversion bei`eks.1`. Allerdings veröffentlicht Amazon EKS regelmäßig neue Plattformversionen, um neue Einstellungen für die Kubernetes-Steuerebene und Sicherheitsfixes bereitzustellen.

Wenn neue Amazon-EKS-Plattformversionen für eine Minor-Version verfügbar werden:

- Die Versionsnummer der Amazon-EKS-Plattform wird erhöht (`eks.`*`n+1`*).
- Amazon EKS aktualisiert automatisch alle bestehenden Cluster auf die neueste Amazon-EKS-Plattformversion für die entsprechende Kubernetes-Minor-Version. Automatische Upgrades bestehender Amazon-EKS-Plattformversionen werden schrittweise durchgeführt. Der Rollout-Prozess kann einige Zeit in Anspruch nehmen. Wenn Sie die neuesten Funktionen der Amazon-EKS-Plattformversion sofort benötigen, sollten Sie einen neuen Amazon-EKS-Cluster erstellen.

  Wenn Ihr Cluster mehr als zwei Plattformversionen hinter der aktuellen Plattformversion zurückliegt, ist es möglich, dass Amazon EKS Ihren Cluster nicht automatisch aktualisieren konnte. Einzelheiten dazu, was dies verursachen kann, finden Sie unter Die Amazon-EKS-Plattformversion liegt mehr als zwei Versionen hinter der aktuellen Plattformversion.
- Amazon EKS veröffentlicht möglicherweise ein neues Knoten-AMI mit einer entsprechenden Patch-Version. Alle Patch-Versionen sind jedoch zwischen der EKS-Steuerebene und den AMIs der Knoten für eine bestimmte Kubernetes-Minor-Version kompatibel.

Neue Amazon-EKS-Plattformversionen führen keine kritischen Änderungen ein. Sie führen nicht zu Service-Unterbrechungen.

Cluster werden immer mit der neuesten verfügbaren Amazon-EKS-Plattformversion (`eks.`*`n`*) für die angegebene Kubernetes-Version erstellt. Wenn Sie Ihren Cluster auf eine neue Kubernetes-Minor-Version aktualisieren, erhält Ihr Cluster die aktuelle Amazon-EKS-Plattformversion für die Kubernetes-Minor-Version, auf die Sie aktualisiert haben.

Die aktuellen und kürzlichen Amazon-EKS-Plattformversionen sind in den folgenden Tabellen beschrieben.

Kubernetes-Version **1.30**

Die folgenden Zugangssteuerungen sind für alle aktiviert 1.30-Plattformversionen: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority, DefaultTolerationSeconds, DefaultStorageClass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, MutatingAdmissionWebhook, ValidatingAdmissionWebhook, ResourceQuota.

Kubernetes-Version	EKS-Plattformversionen	Versionshinweise	Datum der Veröffentlichung
1.30.1	eks.3	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. Juni 2024
1.30.0	eks.2	Erste Veröffentlichung der Kubernetes-Version 1.30 für EKS. Weitere Informationen finden Sie unter Kubernetes1,30.	23. Mai 2024

Kubernetes-Version **1.29**

Die folgenden Zugangssteuerungen sind für alle aktiviert 1.29-Plattformversionen: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority, DefaultTolerationSeconds, DefaultStorageClass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, MutatingAdmissionWebhook, ValidatingAdmissionWebhook, ResourceQuota.

Kubernetes-Version	EKS-Plattformversionen	Versionshinweise	Datum der Veröffentlichung
1.29.5	eks.8	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. Juni 2024

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.29.4	eks.7	Neue Plattformversion mit CoreDNS-Autoscaling, Sicherheitskorrekturen und Verbesserungen. Weitere Informationen zur automatischen CoreDNS-Skalierung finden Sie unter Automatische Skalierung CoreDNS	16. Mai 2024
1.29.3	eks.6	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	18. April 2024
1.29.1	eks.5	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	29. März 2024
1.29.1	eks.4	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	20. März 2024
1.29.1	eks.3	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	12. März 2024
1.29.0	eks.1	Erste Veröffentlichung der Kubernetes-Version 1.29 für EKS. Weitere Informationen finden Sie unter Kubernetes1.29.	23. Januar 2024

Kubernetes-Version **1.28**

Die folgenden Zugangssteuerungen sind für alle aktiviert 1.28-Plattformversionen: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority, DefaultTolerationSeconds, DefaultStorageClass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, MutatingAdmissionWebhook, ValidatingAdmissionWebhook, ResourceQuota.

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.28.10	eks.14	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	7. Juni 2024
1.28.9	eks.13	Neue Plattformversion mit CoreDNS-Autoscaling, Sicherheitskorrekt uren und Verbesserungen. Weitere Informationen zur automatischen CoreDNS-Skalierung finden Sie unter. Automatische Skalierung CoreDNS	16. Mai 2024
1.28.8	eks.12	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	18. April 2024
1.28.7	eks.11	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	29. März 2024
1.28.7	eks.10	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	20. März 2024
1.28.6	eks.9	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	12. März 2024
1.28.5	eks.7	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	17. Januar 2024
1.28.4	eks.6	Neue Plattformversion mit Zugriffse inträgen, Sicherheitsfixes und Verbesserungen.	14. Dezember 2023
1.28.4	eks.5	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	12. Dezember 2023
1.28.3	eks.4	Neue Plattformversion mit EKS-Pod-Identitäten, Sicherheitsfixes und Verbesserungen.	10. November 2023

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.28.3	eks.3	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	3. November 2023
1.28.2	eks.2	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	16. Oktober 2023
1.28.1	eks.1	Erste Veröffentlichung der Kubernetes-Version 1.28 für EKS. Weitere Informationen finden Sie unter Kubernetes1.28.	26. September 2023

Kubernetes-Version **1.27**

Die folgenden Zugangssteuerungen sind für alle aktiviert 1.27-Plattformversionen: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority, DefaultTolerationSeconds, DefaultStorageClass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, MutatingAdmissionWebhook, ValidatingAdmissionWebhook, ResourceQuota.

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.27.14	eks.18	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. Juni 2024
1.27.13	eks.17	Neue Plattformversion mit CoreDNS-Autoscaling, Sicherheitskorrekturen und Verbesserungen. Weitere Informationen zur automatischen CoreDNS-Skalierung finden Sie unter. Automatische Skalierung CoreDNS	16. Mai 2024

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.27.12	eks.16	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	18. April 2024
1.27.11	eks.15	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	29. März 2024
1.27.11	eks.14	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	20. März 2024
1.27.10	eks.13	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	12. März 2024
1.27.9	eks.11	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	17. Januar 2024
1.27.8	eks.10	Neue Plattformversion mit Zugriffse inträgen, Sicherheitsfixes und Verbesserungen.	14. Dezember 2023
1.27.8	eks.9	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	12. Dezember 2023
1.27.7	eks.8	Neue Plattformversion mit EKS-Pod-Identitäten, Sicherheitsfixes und Verbesserungen.	10. November 2023
1.27.7	eks.7	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	3. November 2023
1.27.6	eks.6	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	16. Oktober 2023
1.27.4	eks.5	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. August 2023

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.27.4	eks.4	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juli 2023
1.27.3	eks.3	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juni 2023
1.27.2	eks.2	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	9. Juni 2023
1.27.1	eks.1	Erste Veröffentlichung der Kubernete s-Version 1.27 für EKS. Weitere Informationen finden Sie unter Kubernetes1.27.	24. Mai 2023

Kubernetes-Version **1.26**

Die folgenden Zugangssteuerungen sind für alle aktiviert 1.26-Plattformversionen: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority, DefaultTolerationSeconds, DefaultStorageClass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, MutatingAdmissionWebhook, ValidatingAdmissionWebhook, ResourceQuota.

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.26.15	eks.19	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	7. Juni 2024
1.26.15	eks.18	Neue Plattformversion mit CoreDNS-Autoscaling, Sicherheitskorrekt uren und Verbesserungen. Weitere Informationen zur automatischen	16. Mai 2024

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
		CoreDNS-Skalierung finden Sie unter. Automatische Skalierung CoreDNS	
1.26.15	eks.17	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	18. April 2024
1.26.14	eks.16	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	29. März 2024
1.26.14	eks.15	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	20. März 2024
1.26.13	eks.14	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	12. März 2024
1.26.12	eks.12	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	17. Januar 2024
1.26.11	eks.11	Neue Plattformversion mit Zugriffseinträgen, Sicherheitsfixes und Verbesserungen.	14. Dezember 2023
1.26.11	eks.10	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	12. Dezember 2023
1.26.10	eks.9	Neue Plattformversion mit EKS-Pod-Identitäten, Sicherheitsfixes und Verbesserungen.	10. November 2023
1.26.10	eks.8	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	3. November 2023
1.26.9	eks.7	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	16. Oktober 2023

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.26.7	eks.6	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. August 2023
1.26.7	eks.5	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juli 2023
1.26.6	eks.4	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juni 2023
1.26.5	eks.3	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	9. Juni 2023
1.26.4	eks.2	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	5. Mai 2023
1.26.2	eks.1	Erste Veröffentlichung der Kubernete s-Version 1.26 für EKS. Weitere Informationen finden Sie unter Kubernetes1.26.	11. April 2023

Kubernetes-Version **1.25**

Die folgenden Zugangssteuerungen sind für alle aktiviert 1.25-Plattformversionen: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority, DefaultTolerationSeconds, DefaultStorageClass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, MutatingAdmissionWebhook, ValidatingAdmissionWebhook, ResourceQuota.

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.25.16	eks.20	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	7. Juni 2024
1.25.16	eks.19	Neue Plattformversion mit CoreDNS-Autoscaling, Sicherheitskorrekt uren und Verbesserungen. Weitere Informationen zur automatischen CoreDNS-Skalierung finden Sie unter. Automatische Skalierung CoreDNS	16. Mai 2024
1.25.16	eks.18	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	18. April 2024
1.25.16	eks.17	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	29. März 2024
1.25.16	eks.16	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	20. März 2024
1.25.16	eks.15	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	12. März 2024
1.25.16	eks.13	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	17. Januar 2024
1.25.16	eks.12	Neue Plattformversion mit Zugriffse inträgen, Sicherheitsfixes und Verbesserungen.	14. Dezember 2023
1.25.16	eks.11	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	12. Dezember 2023
1.25.15	eks.10	Neue Plattformversion mit EKS-Pod-Identitäten, Sicherheitsfixes und Verbesserungen.	10. November 2023

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.25.15	eks.9	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	3. November 2023
1.25.14	eks.8	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	16. Oktober 2023
1.25.12	eks.7	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. August 2023
1.25.12	eks.6	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juli 2023
1.25.11	eks.5	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juni 2023
1.25.10	eks.4	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	9. Juni 2023
1.25.9	eks.3	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	5. Mai 2023
1.25.8	eks.2	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	24. März 2023
1.25.6	eks.1	Erste Veröffentlichung der Kubernete s-Version 1.25 für EKS. Weitere Informationen finden Sie unter Kubernetes1,25.	21. Februar 2023

Kubernetes-Version **1.24**

Die folgenden Zugangscontroller sind für alle Versionen der Plattform 1.24 aktiviert: CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, ExtendedResourceToleration, LimitRanger, MutatingAdmissionWebhook,

NamespaceLifecycle, NodeRestriction, PersistentVolumeClaimResize, Priority, PodSecurityPolicy, ResourceQuota, RuntimeClass, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition und ValidatingAdmissionWebhook.

Kubernetes-Version	EKS-Plattformversionen	Versionshinweise	Datum der Veröffentlichung
1.24.17	eks.23	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. Juni 2024
1.24.17	eks.22	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	16. Mai 2024
1.24.17	eks.21	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	18. April 2024
1.24.17	eks.20	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	29. März 2024
1.24.17	eks.19	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	20. März 2024
1.24.17	eks.18	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	12. März 2024
1.24.17	eks.16	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	17. Januar 2024
1.24.17	eks.15	Neue Plattformversion mit Zugriffseinträgen, Sicherheitsfixes und Verbesserungen.	14. Dezember 2023
1.24.17	eks.14	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	12. Dezember 2023
1.24.17	eks.13	Neue Plattformversion mit EKS-Pod-Identitäten, Sicherheitsfixes und Verbesserungen.	10. November 2023

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.24.17	eks.12	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	3. November 2023
1.24.17	eks.11	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	16. Oktober 2023
1.24.16	eks.10	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. August 2023
1.24.16	eks.9	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juli 2023
1.24.15	eks.8	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juni 2023
1.24.14	eks.7	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	9. Juni 2023
1.24.13	eks.6	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	5. Mai 2023
1.24.12	eks.5	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	24. März 2023
1.24.8	eks.4	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	27. Januar 2023
1.24.7	eks.3	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	5. Dezember 2022
1.24.7	eks.2	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	18. November 2022

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.24.7	eks.1	Erste Veröffentlichung der Kubernetes-Version 1.24 für EKS. Weitere Informationen finden Sie unter Kubernetes1.24.	15. November 2022

Kubernetes-Version **1.23**

Die folgenden Zugangscontroller sind für alle Versionen der Plattform 1.23 aktiviert: CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, ExtendedResourceToleration, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, PersistentVolumeClaimResize, Priority, PodSecurityPolicy, ResourceQuota, RuntimeClass, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition und ValidatingAdmissionWebhook.

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.23.17	eks.25	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	7. Juni 2024
1.23.17	eks.24	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	16. Mai 2024
1.23.17	eks.23	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	18. April 2024
1.23.17	eks.22	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	29. März 2024
1.23.17	eks.21	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	20. März 2024

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.23.17	eks.20	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	12. März 2024
1.23.17	eks.18	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	17. Januar 2024
1.23.17	eks.17	Neue Plattformversion mit Zugriffse inträgen, Sicherheitsfixes und Verbesserungen.	14. Dezember 2023
1.23.17	eks.16	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	12. Dezember 2023
1.23.17	eks.15	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	10. November 2023
1.23.17	eks.14	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	3. November 2023
1.23.17	eks.13	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	16. Oktober 2023
1.23.17	eks.12	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. August 2023
1.23.17	eks.11	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juli 2023
1.23.17	eks.10	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	30. Juni 2023
1.23.17	eks.9	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	9. Juni 2023
1.23.17	eks.8	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	5. Mai 2023

Kubernetes-Version	EKS-Platt formversionen	Versionshinweise	Datum der Veröffentlichung
1.23.17	eks.7	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	24. März 2023
1.23.14	eks.6	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	27. Januar 2023
1.23.13	eks.5	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	5. Dezember 2022
1.23.13	eks.4	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	18. November 2022
1.23.12	eks.3	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	7. November 2022
1.23.10	eks.2	Neue Plattformversion mit Sicherhei tsfixes und -verbesserungen.	21. September 2022
1.23.7	eks.1	Erste Veröffentlichung der Kubernete s-Version 1.23 für EKS. Weitere Informationen finden Sie unter Kubernetes1.23.	11. August 2022

Holen Sie sich die aktuelle Plattformversion

Um die aktuelle Plattformversion für Ihren Cluster (Konsole) zu erhalten

1. Öffnen Sie die Amazon-EKS-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie in der Clusterliste den Clusternamen aus, von dem Sie die Plattformversion überprüfen möchten.
4. Wählen Sie die Übersicht-Registerkarte.
5. Die Plattformversion ist unter im Abschnitt Details verfügbar.

So rufen Sie die aktuelle Plattformversion für Ihren Cluster ab (AWS CLI)

1. Ermitteln Sie den Namen des Clusters, dessen Plattformversion Sie überprüfen möchten.
2. Führen Sie den folgenden Befehl aus:

```
aws eks describe-cluster --name my-cluster --query cluster.platformVersion
```

Eine Beispielausgabe sieht wie folgt aus.

```
"eks.10"
```

Auto Scaling

Die automatische Skalierung ist eine Funktion, die Ihre Ressourcen automatisch nach oben oder unten skaliert, um wechselnden Anforderungen gerecht zu werden. Dies ist eine wichtige Kubernetes-Funktion, die ansonsten umfangreiche Personalressourcen erfordern würde, um manuell zu arbeiten.

Amazon EKS unterstützt zwei Auto-Scaling-Produkte:

Karpenter

Karpenter ist ein flexibler, hochleistungsfähiger Kubernetes Cluster Autoscaler, der die Anwendungsverfügbarkeit und Clustereffizienz verbessert. Karpenter startet Datenverarbeitungsressourcen (z. B. Amazon-EC2-Instances) in der richtigen Größe als Reaktion auf die sich ändernde Anwendungslast in weniger als einer Minute. Durch die Integration von Kubernetes mit AWS kann Karpenter Just-In-Time-Datenverarbeitungsressourcen bereitstellen, die genau den Anforderungen Ihrer Workload entsprechen. Karpenter stellt automatisch neue Datenverarbeitungsressourcen basierend auf den spezifischen Anforderungen von Cluster-Workloads bereit. Dazu gehören Rechen-, Speicher-, Beschleunigungs- und Planungsanforderungen. Amazon EKS unterstützt Cluster, die Karpenter verwenden. Karpenter funktioniert allerdings mit jedem konformierten Kubernetes-Cluster. Weitere Informationen finden Sie in der Dokumentation zu Karpenter.

Cluster Autoscaler

Der Kubernetes Cluster Autoscaler passt die Anzahl der Knoten in Ihrem Cluster automatisch an, wenn Pods ausfallen oder auf andere Knoten umgeplant werden. Der Cluster Autoscaler

verwendet Auto-Scaling-Gruppen. Weitere Informationen finden Sie unter Cluster Autoscaler on AWS.

Zugriff verwalten

Erfahren Sie, wie Sie den Zugriff auf Ihren Amazon EKS-Cluster verwalten. Die Verwendung von Amazon EKS erfordert Kenntnisse darüber, wie Kubernetes sowohl als auch AWS Identity and Access Management (AWS IAM) die Zugriffskontrolle handhaben.

Dieser Abschnitt umfasst:

the section called “Gewähren Sie Zugriff auf Kubernetes-APIs”— Erfahren Sie, wie Sie Anwendungen oder Benutzern die Authentifizierung bei der Kubernetes API ermöglichen. Sie können Zugriffseinträge, aws-auth oder einen externen ConfigMap OIDC-Anbieter verwenden.

the section called “Greifen Sie mit kubectl auf meinen Cluster zu”— Erfahren Sie, wie Sie kubectl für die Kommunikation mit Ihrem Amazon EKS-Cluster konfigurieren. Verwenden Sie die AWS CLI, um eine kubeconfig-Datei zu erstellen.

the section called “Gewähren Sie Workloads Zugriff auf AWS”— Erfahren Sie, wie Sie ein Kubernetes Dienstkonto mit AWS IAM-Rollen verknüpfen. Sie können Pod Identity oder IAM-Rollen für Dienstkonten (IRSA) verwenden.

Allgemeine Aufgaben:

- Gewähren Sie Entwicklern Zugriff auf die Kubernetes API. KubernetesRessourcen anzeigen in der AWS Management Console.
  - Lösung: Verwenden Sie Access Entries, um Kubernetes RBAC-Berechtigungen AWS IAM-Benutzern oder -Rollen zuzuordnen.
- Konfigurieren Sie kubectl so, dass es mithilfe AWS von Anmeldeinformationen mit einem Amazon EKS-Cluster kommuniziert.
  - Lösung: Verwenden Sie die AWS CLI, um eine kubeconfig-Datei zu erstellen.
- Verwenden Sie einen externen Identitätsanbieter wie Ping Identity, um Benutzer bei der API zu authentifizieren. Kubernetes
  - Lösung: Verknüpfen Sie einen externen OIDC-Anbieter.
- Gewähren Sie Workloads auf Ihrem Kubernetes Cluster die Möglichkeit, APIs aufzurufen. AWS
  - Lösung: Verwenden Sie Pod Identity, um eine AWS IAM-Rolle einem Kubernetes Dienstkonto zuzuordnen.

Hintergrund:

- Erfahren Sie, wie Kubernetes-Dienstkonten funktionieren.
- Sehen Sie sich das RBAC-Modell (Role Based Access Control) von Kubernetes an
- Weitere Informationen zur Verwaltung des Zugriffs auf AWS Ressourcen finden Sie im AWS IAM-Benutzerhandbuch. Alternativ können Sie an einer kostenlosen Einführungsschulung zur Verwendung von AWS IAM teilnehmen.

Zugriff auf Kubernetes APIs gewähren

Ihr Cluster hat einen Kubernetes API-Endpunkt. Kubectl verwendet diese API. Sie können sich mit zwei Arten von Identitäten bei dieser API authentifizieren:

- Ein AWS Identity and Access Management (IAM-) Principal (Rolle oder Benutzer) — Dieser Typ erfordert eine Authentifizierung bei IAM. Benutzer können sich AWS als IAM-Benutzer oder mit einer föderierten Identität anmelden, indem sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. Benutzer können sich nur dann mit einer Verbundidentität anmelden, wenn Ihr Administrator zuvor mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet hat. Wenn Benutzer AWS mithilfe eines Verbunds darauf zugreifen, übernehmen sie indirekt eine Rolle. Wenn Benutzer diese Art von Identität verwenden, gilt Folgendes:
  - Sie können ihnen Kubernetes-Berechtigungen zuweisen, damit sie mit Kubernetes-Objekten in Ihrem Cluster arbeiten können. Weitere Informationen dazu, wie Sie Ihren IAM-Prinzipalen Berechtigungen zuweisen, damit sie auf Kubernetes-Objekte in Ihrem Cluster zugreifen können, finden Sie hier: Zugangseinträge verwalten.
  - Kann ihnen IAM-Berechtigungen zuweisen, sodass sie mithilfe der Amazon EKS-API,,, AWS CLI oder `eksctl` mit Ihrem Amazon EKS-Cluster und seinen Ressourcen arbeiten können. AWS CloudFormation AWS Management Console Weitere Informationen finden Sie in der Service-Authorization-Referenz unter Von Amazon Elastic Kubernetes Service definierte Aktionen.
  - Wenn Knoten in Ihren Cluster eingebunden werden, nehmen sie eine IAM-Rolle an. Die Möglichkeit, mit IAM-Prinzipalen auf Ihren Cluster zuzugreifen, wird durch den AWS -IAM-Authenticator für Kubernetes bereitgestellt, der auf der Steuerebene von Amazon EKS ausgeführt wird.
- Ein Benutzer in Ihrem eigenen OIDC-Anbieter (OpenID Connect): Bei diesen Typ ist eine Authentifizierung bei Ihrem OIDC-Anbieter erforderlich. Weitere Informationen zur Einrichtung Ihres eigenen OIDC-Anbieters mit Ihrem Amazon-EKS-Cluster finden Sie unter Authentifizieren Sie

Benutzer für Ihren Cluster von einem OpenID Connect Identitätsanbieter. Wenn Benutzer diese Art von Identität verwenden, gilt Folgendes:

- Sie können ihnen Kubernetes-Berechtigungen zuweisen, damit sie mit Kubernetes-Objekten in Ihrem Cluster arbeiten können.
- Ihnen können keine IAM-Berechtigungen zugewiesen werden, sodass sie mithilfe der Amazon EKS-API,,, AWS CLI oder `eksctl` mit Ihrem Amazon EKS-Cluster und seinen Ressourcen arbeiten können. AWS CloudFormation AWS Management Console

Sie können beide Arten von Identitäten mit Ihrem Cluster verwenden. Die IAM-Authentifizierungsmethode kann nicht deaktiviert werden. Die OIDC-Authentifizierungsmethode ist optional.

Verknüpfen Sie IAM-Identitäten mit Kubernetes-Berechtigungen

Die AWS -IAM-Authentifizierung für Kubernetes ist auf der Steuerebene Ihres Clusters installiert. Sie ermöglicht von Ihnen zugelassenen AWS Identity and Access Management (IAM)-Prinzipalen (Rollen und Benutzer) den Zugriff auf Kubernetes-Ressourcen in Ihrem Cluster. Sie können eine der folgenden Methoden verwenden, um IAM-Prinzipalen Zugriff auf Kubernetes-Objekte in Ihrem Cluster zu gewähren:

- Erstellen von Zugriffseinträgen: Wenn Ihr Cluster mindestens über die Plattformversion verfügt, die im Abschnitt Voraussetzungen für die Kubernetes-Version Ihres Clusters aufgeführt ist, sollten Sie diese Option verwenden.

  Verwenden Sie Zugriffseinträge, um die Kubernetes-Berechtigungen von IAM-Prinzipalen außerhalb des Clusters zu verwalten. Sie können den Zugriff auf den Cluster mithilfe der EKS-API, der AWS SDKs und und hinzufügen AWS Command Line Interface und verwalten. AWS CloudFormation AWS Management Console Für die Benutzerverwaltung können also die gleichen Tools verwendet werden wie für die Clustererstellung.

  Beginnen Sie mit Einrichten von Zugriffseinträgen, gefolgt von Migrieren vorhandener `aws-auth` `ConfigMap`-Einträge zu Zugriffseinträgen.
- Hinzufügen von Einträgen zu **`aws-auth` `ConfigMap`**: Wenn die Plattformversion Ihres Clusters älter ist als die im Abschnitt Voraussetzungen aufgeführte Version, muss diese Option verwendet werden. Falls die Plattformversion Ihres Clusters mindestens der Plattformversion entspricht, die im Abschnitt Voraussetzungen für die Kubernetes-Version Ihres Clusters angegeben ist, und Sie Einträge zu `ConfigMap` hinzugefügt haben, empfiehlt es sich, diese Einträge

zu Zugriffseinträgen zu migrieren. Sie können allerdings keine Einträge migrieren, die von Amazon EKS zu `ConfigMap` hinzugefügt wurden (also beispielsweise Einträge für IAM-Rollen, die mit verwalteten Knotengruppen oder Fargate-Profilen verwendet werden). Weitere Informationen finden Sie unter the section called “Gewähren Sie Zugriff auf Kubernetes-APIs”.

- Wenn Sie `aws-auth` `ConfigMap` verwenden müssen, können Sie mithilfe des Befehls **`eksctl create iamidentitymapping`** Einträge zu `ConfigMap` hinzufügen. Weitere Informationen finden Sie in der Dokumentation zu `eksctl` unter Manage IAM users and roles.

Stellen Sie den Cluster-Authentifizierungsmodus ein

Jeder Cluster hat einen Authentifizierungsmodus. Der Authentifizierungsmodus bestimmt, mit welchen Methoden Sie IAM-Prinzipalen den Zugriff auf Kubernetes-Objekte in Ihrem Cluster ermöglichen können. Es gibt drei Authentifizierungsmodi.

> **Important**
>
> Sobald die Zugriffseingabemethode aktiviert ist, kann sie nicht deaktiviert werden. Wenn die `ConfigMap` Methode bei der Clustererstellung nicht aktiviert wurde, kann sie später nicht aktiviert werden. Bei allen Clustern, die vor der Einführung von Zugriffseinträgen erstellt wurden, ist die `ConfigMap` Methode aktiviert.

`aws-auth` `ConfigMap` innerhalb des Clusters

Dies ist der ursprüngliche Authentifizierungsmodus für Amazon-EKS-Cluster. Der IAM-Prinzipal, der den Cluster erstellt hat, ist zunächst der einzige Benutzer, der über `kubectl` auf den Cluster zugreifen kann. Dieser Benutzer muss der Liste in `aws-auth` `ConfigMap` weitere Benutzer hinzufügen und Berechtigungen zuweisen, die sich auf die anderen Benutzer innerhalb des Clusters auswirken. Die anderen Benutzer können den ursprünglichen Benutzer nicht verwalten oder entfernen, da `ConfigMap` keinen zu verwaltenden Eintrag enthält.

Kombination aus `ConfigMap` und Zugriffseinträgen

In diesem Authentifizierungsmodus können Sie beide Methoden verwenden, um dem Cluster IAM-Prinzipale hinzuzufügen. Beachten Sie, dass jede Methode separate Einträge speichert. Wenn Sie beispielsweise einen Zugriffseintrag aus dem hinzufügen AWS CLI, `aws-auth` `ConfigMap` wird der nicht aktualisiert.

Nur Zugriffseinträge

In diesem Authentifizierungsmodus können Sie die EKS-API, die AWS SDKs und verwenden AWS Command Line Interface, AWS Management Console um den Zugriff auf den Cluster für IAM-Prinzipale zu verwalten. AWS CloudFormation

Jeder Zugriffseintrag hat einen Typ und Sie können eine Kombination aus Zugriffsbereich und Zugriffsrichtlinie verwenden, um mithilfe des Zugriffsbereichs den Prinzipal auf einen bestimmten Namespace zu beschränken und mithilfe der Zugriffsrichtlinie vorkonfigurierte, wiederverwendbare Berechtigungsrichtlinien festzulegen. Alternativ können Sie den Typ STANDARD und Kubernetes-RBAC-Gruppen verwenden, um benutzerdefinierte Berechtigungen zuzuweisen.

Authentifizierungsmodus	Methoden
Nur `ConfigMap` (`CONFIG_MAP`)	`aws-auth` `ConfigMap`
EKS-API und `ConfigMap` (`API_AND_CONFIG_MAP`)	auf Einträge in der EKS-API AWS Command Line Interface, den AWS SDKs und und zugreifen AWS CloudFormation AWS Management Console `aws-auth` `ConfigMap`
Nur EKS-API (`API`)	auf Einträge in der EKS-API, AWS SDKs AWS Command Line Interface, und AWS CloudFormation zugreifen AWS Management Console

Zugangseinträge verwalten

Voraussetzungen

- Vertrautheit mit den Cluster-Zugriffsoptionen für Ihren Amazon-EKS-Cluster. Weitere Informationen finden Sie unter Zugriff auf Kubernetes APIs gewähren .
- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS. Um Zugriffseinträge verwenden und den Authentifizierungsmodus eines Clusters ändern zu können, muss der Cluster über eine Plattformversion verfügen, die mindestens der Version aus der folgenden Tabelle entspricht, oder über eine Kubernetes-Version, die neuer als die in der Tabelle aufgeführten Versionen ist.

Kubernetes-Version	Plattformversion
1.30	eks.2
1.29	eks.1
1.28	eks.6
1.27	eks.10
1.26	eks.11
1.25	eks.12
1.24	eks.15
1.23	eks.17

Sie können Ihre aktuelle Kubernetes- und Plattformversion überprüfen, indem Sie *my-cluster* im folgenden Befehl durch den Namen Ihres Clusters ersetzen und dann den geänderten Befehl ausführen: **aws eks describe-cluster --name *my-cluster* --query 'cluster.{"Kubernetes Version": version, "Platform Version": platformVersion}'**.

> **Important**
>
> Nachdem Amazon EKS Ihren Cluster auf die in der Tabelle aufgeführte Plattformversion aktualisiert hat, erstellt Amazon EKS für den IAM-Prinzipal, der den Cluster ursprünglich erstellt hat, einen Zugriffseintrag mit Administratorberechtigungen für den Cluster. Falls dieser IAM-Prinzipal nicht über Administratorberechtigungen für den Cluster verfügen soll, entfernen Sie den von Amazon EKS erstellten Zugriffseintrag.
> Bei Clustern mit Plattformversionen, die älter sind als die in der vorherigen Tabelle aufgeführten Versionen, ist der Cluster-Ersteller immer ein Cluster-Administrator. Es ist nicht möglich, dem IAM-Benutzer oder der IAM-Rolle, der bzw. die den Cluster erstellt hat, die Cluster-Administratorberechtigungen zu entziehen.

- Ein IAM-Prinzipal mit folgenden Berechtigungen für Ihren Cluster: `CreateAccessEntry`, `ListAccessEntries`, `DescribeAccessEntry`, `DeleteAccessEntry` und

`UpdateAccessEntry`. Weitere Informationen zu Amazon-EKS-Berechtigungen finden Sie in der Service-Authorization-Referenz unter Von Amazon Elastic Kubernetes Service definierte Aktionen.

- Ein vorhandener IAM-Prinzipal, für den ein Zugriffseintrag erstellt werden soll, oder ein vorhandener Zugriffseintrag, der aktualisiert oder gelöscht werden soll.

Einrichten von Zugriffseinträgen

Wenn Sie Zugriffseinträge verwenden möchten, müssen Sie zunächst den Authentifizierungsmodus des Clusters in `API_AND_CONFIG_MAP` oder `API` ändern. Dadurch wird die API für Zugriffseinträge hinzugefügt.

AWS Management Console

So erstellen Sie einen Zugriffseintrag

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters aus, in dem Sie einen Zugriffseintrag erstellen möchten.
3. Wählen Sie die Registerkarte Zugriff aus.
4. Unter Authentifizierungsmodus wird der aktuelle Authentifizierungsmodus des Clusters angezeigt. Wenn der Modus EKS API lautet, können Sie bereits Zugriffseinträge hinzufügen und die restlichen Schritte überspringen.
5. Wählen Sie Zugriff verwalten aus.
6. Wählen Sie unter Cluster-Authentifizierungsmodus einen Modus mit der EKS API aus. Beachten Sie, dass der Authentifizierungsmodus nicht mehr in einen Modus geändert werden kann, in dem die EKS API und die Zugriffseinträge entfernt werden.
7. Wählen Sie Änderungen speichern aus. Amazon EKS beginnt mit der Aktualisierung des Clusters, der Status des Clusters ändert sich in Updating und die Änderung wird auf der Registerkarte Aktualisierungsverlauf erfasst.
8. Warten Sie, bis der Cluster wieder den Status Active hat. Wenn der Clusterstatus wieder Active lautet, können Sie die unter Erstellen von Zugriffseinträgen angegebenen Schritte ausführen, um Clusterzugriff für IAM-Prinzipale hinzuzufügen.

AWS CLI

Voraussetzung

Die neueste Version von Version AWS CLI 1 ist auf Ihrem Gerät installiert und konfiguriert oder. AWS CloudShell AWS CLI v2 unterstützt seit einigen Tagen keine neuen Funktionen. Sie können Ihre aktuelle Version mit `aws --version | cut -d / -f2 | cut -d ' ' -f1` überprüfen. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version finden Sie unter Installation, Aktualisierung und Deinstallation AWS CLI und Schnellkonfiguration mit `aws configure` im AWS Command Line Interface Benutzerhandbuch. Die in der installierte AWS CLI Version AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

1.
2. Führen Sie den folgenden Befehl aus. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters. Wenn Sie die `ConfigMap`-Methode dauerhaft deaktivieren möchten, ersetzen Sie `API_AND_CONFIG_MAP` durch `API`.

   Amazon EKS beginnt mit der Aktualisierung des Clusters, der Status des Clusters ändert sich in UPDATING und die Änderung wird in aws eks list-updates erfasst.

   ```
   aws eks update-cluster-config --name my-cluster --access-config
    authenticationMode=API_AND_CONFIG_MAP
   ```

3. Warten Sie, bis der Cluster wieder den Status Active hat. Wenn der Clusterstatus wieder Active lautet, können Sie die unter Erstellen von Zugriffseinträgen angegebenen Schritte ausführen, um Clusterzugriff für IAM-Prinzipale hinzuzufügen.

Erstellen von Zugriffseinträgen

Überlegungen

Berücksichtigen Sie Folgendes, bevor Sie Zugriffseinträge erstellen:

- Ein Zugriffseintrag enthält den Amazon-Ressourcennamen (ARN) genau eines vorhandenen IAM-Prinzipals. Ein IAM-Prinzipal kann nicht in mehreren Zugriffseinträgen enthalten sein. Zusätzliche Überlegungen zu dem von Ihnen angegebenen ARN:
  - In den bewährten Methoden für IAM wird empfohlen, für den Zugriff auf Ihren Cluster IAM-Rollen mit kurzfristigen Anmeldeinformationen zu verwenden anstatt IAM-Benutzer mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch

unter Erfordern, dass menschliche Benutzer für den Zugriff AWS mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

- Ein ARN für eine IAM-Rolle kann einen Pfad enthalten. ARNs in `aws-auth ConfigMap`-Einträgen können keinen Pfad enthalten. Ihr ARN kann beispielsweise `arn:aws:iam::111122223333:role/development/apps/my-role` oder `arn:aws:iam::111122223333:role/my-role` lauten.
- Wenn der Typ des Zugriffseintrags etwas anderes ist als `STANDARD` (siehe nächste Überlegung zu Typen), muss sich der ARN in demselben befinden AWS-Konto , in dem sich Ihr Cluster befindet. Wenn der Typ ist`STANDARD`, kann sich der ARN in demselben oder einem anderen Konto befinden AWS-Konto als das Konto, in dem sich Ihr Cluster befindet.
- Nach der Erstellung des Zugriffseintrags kann der IAM-Prinzipal nicht mehr geändert werden.
- Sollten Sie den IAM-Prinzipal mit diesem ARN löschen, wird der Zugriffseintrag nicht automatisch gelöscht. Es empfiehlt sich, den Zugriffseintrag mit einem ARN für einen gelöschten IAM-Prinzipal zu löschen. Wenn Sie den Zugriffseintrag nicht löschen und den IAM-Prinzipal später einmal erneut erstellen, funktioniert der Zugriffseintrag nicht mehr, auch wenn der ARN gleich ist. Dies liegt daran, dass, obwohl der ARN für den neu erstellten IAM-Prinzipal derselbe ist, der `roleID` oder `userID` (Sie können dies mit dem `aws sts get-caller-identity` AWS CLI Befehl sehen) für den neu erstellten IAM-Prinzipal anders ist als für den ursprünglichen IAM-Prinzipal. Auch wenn Sie die Rollen-ID (`roleID`) bzw. die Benutzer-ID (`userID`) des IAM-Prinzipals für einen Zugriffseintrag nicht sehen, wird sie von Amazon EKS zusammen mit dem Zugriffseintrag gespeichert.

- Jeder Zugriffseintrag hat einen Typ. Sie können `EC2 Linux` (für eine IAM-Rolle, die mit selbstverwalteten Linux- oder Bottlerocket-Knoten verwendet wird), `EC2 Windows` (für eine IAM-Rolle, die mit selbstverwalteten Windows-Knoten verwendet wird), `FARGATE_LINUX` (für IAM-Rollen, die mit verwendet werden AWS Fargate (Fargate)) oder `STANDARD` als Typ angeben. Wenn Sie keinen Typ angeben, wird er von Amazon EKS automatisch auf `STANDARD` festgelegt. Für eine IAM-Rolle, die für eine verwaltete Knotengruppe oder für ein Fargate-Profil verwendet wird, muss kein Zugriffseintrag erstellt werden, da Amazon EKS Einträge für diese Rollen zu `aws-auth ConfigMap` hinzufügt (unabhängig davon, welche Plattformversion Ihr Cluster verwendet).

  Nach der Erstellung des Zugriffseintrags kann der Typ nicht mehr geändert werden.
- Wenn der Typ des Zugriffseintrags `STANDARD` lautet, können Sie einen Benutzernamen für den Zugriffseintrag angeben. Wenn Sie keinen Wert für den Benutzernamen angeben, legt Amazon EKS einen der folgenden Werte für Sie fest, abhängig vom Typ des Zugriffseintrags und davon, ob es sich bei dem von Ihnen angegebenen IAM-Prinzipal um eine IAM-Rolle oder um

einen IAM-Benutzer handelt. Sofern Sie keinen spezifischen Grund für die Angabe eines eigenen Benutzernamens haben, empfehlen wir, keinen anzugeben und ihn automatisch von Amazon EKS generieren zu lassen. Beachten Sie bei Angabe eines eigenen Benutzernamens Folgendes:

- Er darf nicht mit `system:`, `eks:`, `aws:`, `amazon:` oder `iam:` beginnen.
- Bei einem Benutzername für eine IAM-Rolle empfiehlt es sich, am Ende des Benutzernamens `{{SessionName}}` hinzuzufügen. Wenn Sie Ihrem Benutzernamen etwas hinzufügen*`{{SessionName}}`*, muss der Benutzername einen Doppelpunkt vor {{}} enthalten. SessionName Wenn diese Rolle übernommen wird, wird der Name der Sitzung, der bei der Übernahme der Rolle angegeben wurde, automatisch an den Cluster übergeben und erscheint in den CloudTrail Protokollen. Der Benutzername `john{{SessionName}}` ist beispielsweise nicht zulässig. Er müsste `:john{{SessionName}}` oder `jo:hn{{SessionName}}` lauten. Der Doppelpunkt muss sich nur vor `{{SessionName}}` befinden. Der von Amazon EKS generierte Benutzername in der folgenden Tabelle enthält einen ARN. Da ein ARN Doppelpunkte enthält, erfüllt er diese Anforderung. Der Doppelpunkt ist nicht erforderlich, wenn Sie `{{SessionName}}` nicht in Ihren Benutzernamen einschließen.

IAM-Prinzipaltyp	Typ	Von Amazon EKS automatisch festgelegter Wert für den Benutzernamen
Benutzer	`STANDARD`	Der ARN des Benutzers. Beispiel: `arn:aws:iam:: `*`111122223333`*` :user/`*`my-user`*

IAM-Prinzipaltyp	Typ	Von Amazon EKS automatisch festgelegter Wert für den Benutzernamen
Rolle	STANDARD	Der STS-ARN der Rolle, wenn sie angenommen wird. Amazon EKS fügt am Ende der Rolle {{SessionName}} hinzu. Beispiel: arn:aws:sts:: *111122223333* :assumed-role/ *my-role*/{{SessionName}} Wenn der ARN der von Ihnen angegebenen Rolle einen Pfad enthalten hat, wird dieser von Amazon EKS im generierten Benutzernamen entfernt.
Rolle	EC2 Linux oder EC2 Windows	system:node:{{EC2PrivateDNSName}}
Rolle	FARGATE_LINUX	system:node:{{SessionName}}

Sie können den Benutzernamen ändern, nachdem der Zugriffseintrag erstellt wurde.

- Wenn der Typ eines Zugriffseintrags STANDARD ist und Sie die Kubernetes-RBAC-Autorisierung verwenden möchten, können Sie dem Zugriffseintrag einen oder mehrere Gruppennamen hinzufügen. Gruppennamen können nach der Erstellung eines Zugriffseintrags hinzugefügt und entfernt werden. Damit der IAM-Prinzipal auf Kubernetes-Objekte in Ihrem Cluster zugreifen kann, müssen Sie Kubernetes-RBAC-Objekte (rollenbasierte Autorisierung) erstellen und verwalten. Erstellen Sie Kubernetes-, RoleBinding- oder ClusterRoleBinding-Objekte für Ihren Cluster, die den Gruppennamen als subject für kind: Group angeben. Kubernetes autorisiert den IAM-Prinzipalzugriff auf alle Clusterobjekte, die Sie in einem Kubernetes-Objekt vom Typ Role oder

`ClusterRole` angegeben haben, das Sie auch in `roleRef` Ihrer Bindung angegeben haben. Wenn Sie Gruppennamen angeben, sollten Sie mit den Kubernetes-RBAC-Objekten (rollenbasierte Autorisierung) vertraut sein. Weitere Informationen finden Sie unter Using RBAC authorization in der Kubernetes-Dokumentation.

> **Important**
>
> Amazon EKS überprüft nicht, ob ggf. in Ihrem Cluster vorhandene Kubernetes-RBAC-Objekte einen der von Ihnen angegebenen Gruppennamen enthalten.

Anstelle der Autorisierung des IAM-Prinzipalzugriffs auf Kubernetes-Objekte in Ihrem Cluster durch Kubernetes (oder zusätzlich dazu) können Sie einem Zugriffseintrag Amazon EKS-Zugriffsrichtlinien zuordnen. Amazon EKS autorisiert für IAM-Prinzipale den Zugriff auf Kubernetes-Objekte in Ihrem Cluster mit den in der Zugriffsrichtlinie festgelegten Berechtigungen. Sie können die Berechtigungen einer Zugriffsrichtlinie auf von Ihnen angegebene Kubernetes-Namespaces ausrichten. Für die Verwendung von Zugriffsrichtlinien müssen keine Kubernetes-RBAC-Objekte verwaltet werden. Weitere Informationen finden Sie unter Zuordnen von Zugriffsrichtlinien zu Zugriffseinträgen und Aufheben dieser Zuordnung.

- Wenn Sie einen Zugriffseintrag mit dem Typ `EC2 Linux` oder `EC2 Windows` erstellen, muss der IAM-Prinzipal, der den Zugriffseintrag erstellt, über die Berechtigung `iam:PassRole` verfügen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erteilen von Berechtigungen, mit denen ein Benutzer eine Rolle an einen AWS-Serviceübergeben kann.
- Ähnlich wie beim standardmäßigen IAM-Verhalten sind die Erstellung und Aktualisierung von Zugriffseinträgen letztlich konsistent und es kann mehrere Sekunden dauern, bis sie wirksam werden, nachdem der erste API-Aufruf erfolgreich abgeschlossen wurde. Sie müssen Ihre Anwendungen unter Berücksichtigung dieser potenziellen Verzögerungen konzipieren. Es empfiehlt sich, in die kritischen, hochverfügbaren Code-Pfade Ihrer Anwendung keine Erstellungen oder Aktualisierungen von Zugriffseinträgen einzuschließen. Nehmen Sie -Änderungen stattdessen in einer separaten Initialisierungs- oder Einrichtungsroutine vor, die seltener ausgeführt wird. Vergewissern Sie sich auch, dass die Änderungen weitergegeben wurden, bevor die Produktionsarbeitsabläufe davon abhängen.
- Zugriffseinträge unterstützen keine dienstbezogenen Rollen. Sie können keine Zugriffseinträge erstellen, bei denen der Prinzipal-ARN eine dienstverknüpfte Rolle ist. Sie können dienstverknüpfte Rollen anhand ihrer ARN identifizieren, die im folgenden Format angegeben ist`arn:aws:iam::*:role/aws-service-role/*`.

Sie können einen Zugriffseintrag mit dem AWS Management Console oder dem erstellen AWS CLI.

AWS Management Console

So erstellen Sie einen Zugriffseintrag

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters aus, in dem Sie einen Zugriffseintrag erstellen möchten.
3. Wählen Sie die Registerkarte Zugriff aus.
4. Wählen Sie Zugriffseintrag erstellen aus.
5. Wählen Sie unter IAM-Prinzipal eine bereits vorhandene IAM-Rolle oder einen bereits vorhandenen IAM-Benutzer aus. In den bewährten Methoden für IAM wird empfohlen, für den Zugriff auf Ihren Cluster IAM-Rollen mit kurzfristigen Anmeldeinformationen zu verwenden anstatt IAM-Benutzer mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zuzugreifen.
6. Wählen Sie unter Typ die Option EC2 Linux oder EC2 Windows aus, wenn sich der Zugriffseintrag auf die Knotenrolle bezieht, die für selbstverwaltete Amazon-EC2-Knoten verwendet wird. Übernehmen Sie andernfalls die Standardeinstellung (Standard).
7. Wenn Sie unter Typ die Option Standard ausgewählt haben, können Sie bei Bedarf unter Benutzername einen Benutzernamen angeben.
8. Wenn Sie unter Typ die Option Standard ausgewählt haben und die Kubernetes-RBAC-Autorisierung für den IAM-Prinzipal verwenden möchten, können Sie unter Gruppen einen oder mehrere Namen angeben. Wenn Sie keine Gruppennamen angeben und die Amazon-EKS-Autorisierung verwenden möchten, können Sie in einem späteren Schritt (oder nach der Erstellung des Zugriffseintrags) eine Zugriffsrichtlinie zuordnen.
9. (Optional) Weisen Sie dem Zugriffseintrag unter Tags Beschriftungen zu. So können Sie beispielsweise ganz einfach nach allen Ressourcen mit dem gleichen Tag suchen.
10. Wählen Sie Weiter aus.
11. Wenn Sie den Typ Standard ausgewählt haben und möchten, dass Amazon EKS den IAM-Prinzipal autorisiert, damit dieser über Berechtigungen für die Kubernetes-Objekte in Ihrem Cluster verfügt, können Sie auf der Seite Zugriffsrichtlinie hinzufügen die folgenden Schritte ausführen. Klicken Sie andernfalls auf Next (Weiter).

a. Wählen Sie unter Richtlinienname eine Zugriffsrichtlinie aus. Sie können zwar die Berechtigungen der Zugriffsrichtlinien nicht anzeigen, sie enthalten aber ähnliche Berechtigungen wie die für Benutzer vorgesehenen Kubernetes-Objekte vom Typ `ClusterRole`. Weitere Informationen finden Sie in der Dokumentation zu Kubernetes unter User-facing roles.

b. Wählen Sie eine der folgenden Optionen:

- Cluster: Wählen Sie diese Option aus, wenn Amazon EKS den IAM-Prinzipal so autorisieren soll, dass er für alle Kubernetes-Objekte in Ihrem Cluster über die in der Zugriffsrichtlinie festgelegten Berechtigungen verfügt.
- Kubernetes-Namespace: Wählen Sie diese Option aus, wenn Amazon EKS den IAM-Prinzipal so autorisieren soll, dass er für alle Kubernetes-Objekte in einem spezifischen Kubernetes-Namespace Ihres Clusters über die in der Zugriffsrichtlinie festgelegten Berechtigungen verfügt. Geben Sie unter Namespace den Namen des Kubernetes-Namespace in Ihrem Cluster ein. Wenn Sie zusätzliche Namespaces hinzufügen möchten, wählen Sie Neuen Namespace hinzufügen aus und geben Sie den Namespace-Namen ein.

c. Wenn Sie weitere Richtlinien hinzufügen möchten, wählen Sie Richtlinie hinzufügen aus. Sie können für jede Richtlinie unterschiedliche Geltungsbereiche festlegen, aber jede Richtlinie kann nur einmal hinzugefügt werden.

d. Wählen Sie Weiter aus.

12. Überprüfen Sie die Konfiguration für Ihren Zugriffseintrag. Sollten Sie einen Fehler entdecken, wählen Sie Zurück aus, um schrittweise zurück zu navigieren und den Fehler zu korrigieren. Ist die Konfiguration korrekt, wählen Sie Erstellen aus.

AWS CLI

Voraussetzung

Die neueste Version von Version AWS CLI 1 ist auf Ihrem Gerät installiert und konfiguriert oder AWS CloudShell. AWS CLI v2 unterstützt seit einigen Tagen keine neuen Funktionen. Sie können Ihre aktuelle Version mit `aws --version | cut -d / -f2 | cut -d ' ' -f1` überprüfen. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version finden Sie unter Installation, Aktualisierung und Deinstallation AWS CLI und Schnellkonfiguration mit `aws configure` im AWS Command Line Interface Benutzerhandbuch. Die in der installierte AWS

CLI Version AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

So erstellen Sie einen Zugriffseintrag

Sie können eines der folgenden Beispiele verwenden, um Zugriffseinträge zu erstellen:

- Erstellen Sie einen Zugriffseintrag für eine selbstverwaltete Amazon-EC2-Linux-Knotengruppe. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters, *111122223333* durch Ihre AWS-Konto ID und *eks-my-cluster-self-managed-ng-1* durch den Namen Ihrer Node-IAM-Rolle. Wenn es sich bei Ihrer Knotengruppe um eine Windows-Knotengruppe handelt, ersetzen Sie *EC2_Linux* durch **EC2_Windows**.

  ```
  aws eks create-access-entry --cluster-name my-cluster --principal-arn
   arn:aws:iam::111122223333:role/EKS-my-cluster-self-managed-ng-1 --type EC2_Linux
  ```

  Bei Angabe eines anderen Typs als `STANDARD` kann die Option `--kubernetes-groups` nicht verwendet werden. Sie können diesem Zugriffseintrag keine Zugriffsrichtlinie zuordnen, da der Typ nicht den Wert `STANDARD` hat.

- Erstellen Sie einen Zugriffseintrag, der eine IAM-Rolle zulässt, die nicht für eine selbstverwaltete Amazon-EC2-Knotengruppe verwendet wird, mit der Kubernetes Zugriff auf Ihren Cluster autorisieren soll. *Ersetzen Sie my-cluster durch den Namen Ihres Clusters, 111122223333 durch Ihre ID und my-role durch den Namen Ihrer IAM-Rolle. AWS-Konto* Ersetzen Sie *Viewers* durch den Namen einer Gruppe, die Sie in einem Kubernetes-Objekt vom Typ `ClusterRoleBinding` oder `RoleBinding` in Ihrem Cluster angegeben haben.

  ```
  aws eks create-access-entry --cluster-name my-cluster --principal-arn
   arn:aws:iam::111122223333:role/my-role --type STANDARD --user Viewers --
  kubernetes-groups Viewers
  ```

- Erstellen Sie einen Zugriffseintrag, der es einem IAM-Benutzer ermöglicht, sich bei Ihrem Cluster zu authentifizieren. Dieses Beispiel soll zeigen, dass es diese Möglichkeit gibt. In den bewährten Methoden für IAM wird empfohlen, für den Zugriff auf Ihren Cluster IAM-Rollen mit kurzfristigen Anmeldeinformationen zu verwenden anstatt IAM-Benutzer mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch

unter Erfordern, dass menschliche Benutzer für den Zugriff AWS mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

```
aws eks create-access-entry --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:user/my-user --type STANDARD --username my-user
```

Wenn dieser Benutzer mehr Zugriff auf Ihren Cluster haben soll als ihm durch die Berechtigungen in den Kubernetes-API-Erkennungsrollen gewährt wird, müssen Sie dem Zugriffseintrag eine Zugriffsrichtlinie zuordnen, da die Option `--kubernetes-groups` nicht verwendet wird. Weitere Informationen finden Sie in der Dokumentation zu Kubernetes unter Zuordnen von Zugriffsrichtlinien zu Zugriffseinträgen und Aufheben dieser Zuordnung sowie unter API discovery roles.

Aktualisieren von Zugriffseinträgen

Sie können einen Zugriffseintrag mit dem AWS Management Console oder dem AWS CLI aktualisieren.

AWS Management Console

So aktualisieren Sie einen Zugriffseintrag

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters aus, in dem Sie einen Zugriffseintrag erstellen möchten.
3. Wählen Sie die Registerkarte Zugriff aus.
4. Wählen Sie den Zugriffseintrag aus, den Sie aktualisieren möchten.
5. Wählen Sie Bearbeiten aus.
6. Unter Benutzername können Sie den vorhandenen Wert ändern.
7. Unter Gruppen können Sie Gruppennamen entfernen oder neue Gruppennamen hinzufügen. Die Gruppennamen system:nodes und system:bootstrappers dürfen nicht entfernt werden (sofern vorhanden). Wenn Sie diese Gruppen entfernen, funktioniert Ihr Cluster unter Umständen nicht mehr ordnungsgemäß. Wenn Sie keine Gruppennamen angeben und die Amazon-EKS-Autorisierung verwenden möchten, ordnen Sie in einem späteren Schritt eine Zugriffsrichtlinie zu.

8. Unter Tags können Sie dem Zugriffseintrag Beschriftungen zuweisen. So können Sie beispielsweise ganz einfach nach allen Ressourcen mit dem gleichen Tag suchen. Außerdem können Sie vorhandene Tags entfernen.
9. Wählen Sie Änderungen speichern aus.
10. Informationen zum Zuordnen einer Zugriffsrichtlinie zu dem Eintrag finden Sie unter Zuordnen von Zugriffsrichtlinien zu Zugriffseinträgen und Aufheben dieser Zuordnung.

AWS CLI

Voraussetzung

Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

So aktualisieren Sie einen Zugriffseintrag

Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters, *`111122223333`* durch Ihre AWS-Konto ID und *`eks-my-cluster-my-namespace-viewers`* durch den Namen einer IAM-Rolle.

```
aws eks update-access-entry --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/EKS-my-cluster-my-namespace-Viewers --kubernetes-
groups Viewers
```

Die Option `--kubernetes-groups` kann nur verwendet werden, wenn der Zugriffseintrag den Typ `STANDARD` hat. Auch kann eine Zugriffsrichtlinie nur einem Zugriffseintrag mit dem Typ `STANDARD` zugeordnet werden.

Löschen von Zugriffseinträgen

Wenn Sie einen Zugriffseintrag versehentlich gelöscht haben, können Sie ihn jederzeit erneut erstellen. Wenn dem zu löschenden Zugriffseintrag Zugriffsrichtlinien zugeordnet sind, werden die Zuordnungen automatisch gelöscht. Es ist nicht erforderlich, die Zuordnung von Zugriffsrichtlinien zu einem Zugriffseintrag vor dem Löschen aufzuheben.

Sie können einen Zugriffseintrag AWS CLI mit dem oder dem löschen. AWS Management Console

AWS Management Console

So löschen Sie einen Zugriffseintrag

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters aus, aus dem Sie einen Zugriffseintrag löschen möchten.
3. Wählen Sie die Registerkarte Zugriff aus.
4. Wählen Sie in der Liste Zugriffseinträge den Zugriffseintrag aus, den Sie löschen möchten.
5. Wählen Sie Löschen aus.
6. Wählen Sie im Bestätigungs-Dialogfeld die Option Delete (Löschen).

AWS CLI

Voraussetzung

Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

So löschen Sie einen Zugriffseintrag

Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters, *111122223333* durch Ihre AWS-Konto ID und *my-role durch den Namen der IAM-Rolle*, auf die Sie keinen Zugriff mehr auf Ihren Cluster haben möchten.

```
aws eks delete-access-entry --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/my-role
```

Zuordnen von Zugriffsrichtlinien zu Zugriffseinträgen und Aufheben dieser Zuordnung

Sie können Zugriffseinträgen vom Typ `STANDARD` eine oder mehrere Zugriffsrichtlinien zuweisen. Amazon EKS erteilt den anderen Arten von Zugriffseinträgen automatisch die erforderlichen Berechtigungen, damit sie in Ihrem Cluster ordnungsgemäß funktionieren. Die Amazon EKS-Zugriffsrichtlinien beinhalten Kubernetes-Berechtigungen (keine IAM-Berechtigungen). Machen Sie sich mit den Kubernetes-Berechtigungen vertraut, die in den einzelnen Zugriffsrichtlinien-Optionen enthalten sind, bevor Sie einem Zugriffseintrag eine Zugriffsrichtlinie zuweisen. Weitere Informationen finden Sie unter Berechtigungen von Zugriffsrichtlinien. Wenn keine der Zugriffsrichtlinien Ihre Anforderungen erfüllt, ordnen Sie einem Zugriffseintrag keine Zugriffsrichtlinie zu. Geben Sie stattdessen mindestens einen Gruppennamen für den Zugriffseintrag an und erstellen und verwalten Sie Kubernetes-Objekte für die rollenbasierte Zugriffssteuerung. Weitere Informationen finden Sie unter Erstellen von Zugriffseinträgen.

Voraussetzungen

- Ein vorhandener Zugriffseintrag. Informationen zum Erstellen finden Sie unter Erstellen von Zugriffseinträgen.
- Eine AWS Identity and Access Management Rolle oder ein Benutzer mit den folgenden Berechtigungen: `ListAccessEntries`,`DescribeAccessEntry`,`UpdateAccessEntry`,`ListAccessPolicies`,`Associ` und`DisassociateAccessPolicy`. Weitere Informationen finden Sie in der Service-Authorization-Referenz unter Von Amazon Elastic Kubernetes Service definierte Aktionen.

Berücksichtigen Sie die folgenden Anforderungen, bevor Sie Zugriffsrichtlinien mit Zugriffseinträgen verknüpfen:

- Sie können jedem Zugriffseintrag mehrere Zugriffsrichtlinien zuordnen, aber Sie können jede Richtlinie nur einmal einem Zugriffseintrag zuordnen. Wenn Sie mehrere Zugriffsrichtlinien zuordnen, verfügt der IAM-Prinzipal des Zugriffseintrags über alle Berechtigungen aus allen zugeordneten Zugriffsrichtlinien.
- Sie können eine Zugriffsrichtlinie auf alle Ressourcen in einem Cluster ausrichten oder den Namen eines oder mehrerer Kubernetes-Namespaces angeben. Sie können Platzhalterzeichen für einen Namespace-Namen verwenden. Wenn Sie beispielsweise eine Zugriffsrichtlinie auf alle Namespaces ausrichten möchten, die mit `dev-` beginnen, können Sie `dev-*` als Namespace-Namen angeben. Stellen Sie sicher, dass die Namespaces in Ihrem Cluster vorhanden sind und dass Ihre Schreibweise dem tatsächlichen Namespace-Namen im Cluster entspricht. Amazon EKS überprüft weder die Schreibweise noch das Vorhandensein der Namespaces in Ihrem Cluster.
- Der Zugriffsbereich für eine Zugriffsrichtlinie kann geändert werden, nachdem sie einem Zugriffseintrag zugeordnet wurde. Wenn Sie die Zugriffsrichtlinie auf Kubernetes-Namespaces beschränkt haben, können Sie bei Bedarf Namespaces für die Zuordnung hinzufügen und entfernen.
- Wenn Sie eine Zugriffsrichtlinie einem Zugriffseintrag zuordnen, für den auch Gruppennamen angegeben sind, verfügt der IAM-Prinzipal über alle Berechtigungen in allen zugehörigen Zugriffsrichtlinien. Außerdem verfügt er über alle Berechtigungen aus jedem Kubernetes-Objekt vom Typ `Role` oder `ClusterRole`, das in einem beliebigen Kubernetes-Objekt vom Typ `Role` oder `RoleBinding` zum Angeben der Gruppennamen angegeben ist.
- Wenn Sie den Befehl `kubectl auth can-i --list` ausführen, werden keine Kubernetes-Berechtigungen angezeigt, die durch Zugriffsrichtlinien zugewiesen wurden, die einem Zugriffseintrag für den IAM-Prinzipal zugeordnet sind, den Sie beim Ausführen des Befehls verwenden. Der Befehl zeigt nur Kubernetes-Berechtigungen an, wenn Sie sie in Kubernetes-Objekten vom Typ `Role` oder `ClusterRole` erteilt haben, die Sie an die Gruppennamen oder an den Benutzernamen gebunden haben, die bzw. den Sie für einen Zugriffseintrag angegeben haben.
- Wenn Sie bei der Interaktion mit Kubernetes-Objekten in Ihrem Cluster die Identität eines Kubernetes-Benutzers oder einer entsprechenden Gruppe annehmen (beispielsweise, wenn Sie den Befehl `kubectl` mit **`--as`** ***`username`*** oder **`--as-group`** ***`group-name`*** verwenden), erzwingen Sie die Verwendung der Kubernetes-RBAC-Autorisierung. Daher werden dem IAM-Prinzipal keine Berechtigungen durch eine Zugriffsrichtlinie zugewiesen, die dem Zugriffseintrag zugeordnet ist. Die einzigen Kubernetes-Berechtigungen, über die der Benutzer oder die Gruppe verfügt, dessen bzw. deren Identität der IAM-Prinzipal angenommen hat, sind die Kubernetes-Berechtigungen, die Sie in Kubernetes-Objekten vom Typ `Role` oder `ClusterRole` erteilt haben,

die von Ihnen an die Gruppennamen oder an den Benutzernamen gebunden wurden. Nehmen Sie nicht die Identität eines Kubernetes-Benutzers oder einer entsprechenden Gruppe an, wenn Ihr IAM-Prinzipal über die Berechtigungen in zugeordneten Zugriffsrichtlinien verfügen soll. Der IAM-Prinzipal verfügt auch weiterhin über alle Berechtigungen, die Sie ihm in den Kubernetes-Objekten vom Typ `Role` oder `ClusterRole` erteilt haben, die Sie an die Gruppennamen oder an den Benutzernamen gebunden haben, die bzw. den Sie für den Zugriffseintrag angegeben haben. Weitere Informationen finden Sie in der Dokumentation zu Kubernetes unter User impersonation.

Mit dem AWS Management Console oder dem können Sie einem Zugriffseintrag eine Zugriffsrichtlinie zuordnen AWS CLI.

AWS Management Console

So verwenden Sie die AWS Management Console, um eine Zugriffsrichtlinie einem Zugriffseintrag zuzuordnen

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters aus, der über einen Zugriffseintrag verfügt, dem Sie eine Zugriffsrichtlinien zuordnen möchten.
3. Wählen Sie die Registerkarte Zugriff aus.
4. Wenn es sich um einen Zugriffseintrag vom Typ Standard handelt, können Sie Zugriffsrichtlinien von Amazon EKS zuordnen oder deren Zuordnung aufheben. Bei Zugriffseinträgen eines anderen Typs (also nicht Standard) steht diese Option nicht zur Verfügung.
5. Wählen Sie Zugriffsrichtlinie zuordnen aus.
6. Wählen Sie unter Richtlinienname die Richtlinie mit den Berechtigungen aus, über die der IAM-Prinzipal verfügen soll. Informationen zu den Berechtigungen, die in der jeweiligen Richtlinie enthalten sind, finden Sie unter Berechtigungen von Zugriffsrichtlinien.
7. Wählen Sie unter Zugriffsbereich einen Zugriffsbereich aus. Bei Verwendung der Option Cluster werden die Berechtigungen in der Zugriffsrichtlinie dem IAM-Prinzipal für Ressourcen in allen Kubernetes-Namespaces erteilt. Bei Verwendung der Option Kubernetes-Namespace können Sie anschließend Neuen Namespace hinzufügen auswählen. In dem daraufhin angezeigten Feld Namespace können Sie den Namen eines Kubernetes-Namespace in Ihrem Cluster eingeben. Wenn die Berechtigungen für den IAM-Prinzipal in mehreren Namespaces gelten sollen, können Sie mehrere Namespaces eingeben.

8. Wählen Sie Zugriffsrichtlinie hinzufügen aus.

AWS CLI

Voraussetzung

Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

So ordnen Sie eine Zugriffsrichtlinie einem Zugriffseintrag zu

1. Sehen Sie sich die verfügbaren Zugriffsrichtlinien an.

```
aws eks list-access-policies --output table
```

Eine Beispielausgabe sieht wie folgt aus.

```
-------------------------------------------------------------------------------------
|                                          ListAccessPolicies
                        |
+-----------------------------------------------------------------------------------
+
||                                           accessPolicies
                       ||
|+-------------------------------------------------------------------
+------------------------------+|
||                                arn                                |
     name              ||
|+-------------------------------------------------------------------
+------------------------------+|
||  arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy       |
 AmazonEKSAdminPolicy         ||
```

```
||  arn:aws:eks::aws:cluster-access-policy/AmazonEKSClusterAdminPolicy |
 AmazonEKSClusterAdminPolicy  ||
||  arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy         |
 AmazonEKSEditPolicy          ||
||  arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy         |
 AmazonEKSViewPolicy          ||
|+-----------------------------------------------------------------------
+------------------------------+|
```

Informationen zu den Berechtigungen, die in der jeweiligen Richtlinie enthalten sind, finden Sie unter Berechtigungen von Zugriffsrichtlinien.

2. Sehen Sie sich Ihre vorhandenen Zugriffseinträge an. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

```
aws eks list-access-entries --cluster-name my-cluster
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "accessEntries": [
        "arn:aws:iam::111122223333:role/my-role",
        "arn:aws:iam::111122223333:user/my-user"
    ]
}
```

3. Ordnen Sie eine Zugriffsrichtlinie einem Zugriffseintrag zu. Im folgenden Beispiel wird die Zugriffsrichtlinie `AmazonEKSViewPolicy` einem Zugriffseintrag zugeordnet. Wenn die IAM-Rolle *my-role* versucht, auf Kubernetes-Objekte im Cluster zuzugreifen, autorisiert Amazon EKS die Rolle nur dazu, die Berechtigungen in der Richtlinie für den Zugriff auf Kubernetes-Objekte in den Kubernetes-Namespaces *my-namespace1* und *my-namespace2* zu verwenden. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters, *111122223333* durch Ihre AWS-Konto -ID und *my-role* durch den Namen der IAM-Rolle, für die Amazon EKS den Zugriff auf Kubernetes-Cluster-Objekte autorisieren soll.

```
aws eks associate-access-policy --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/my-role \
    --access-scope type=namespace,namespaces=my-namespace1,my-namespace2 --
policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy
```

Wenn die Berechtigungen für den IAM-Prinzipal clusterweit gelten sollen, ersetzen Sie **type=namespace,namespaces=*my-namespace1*,*my-namespace2*** durch **type=cluster**. Wenn Sie dem Zugriffseintrag mehrere Zugriffsrichtlinien zuordnen möchten, führen Sie den Befehl mehrmals aus und verwenden Sie dabei jeweils eine individuelle Zugriffsrichtlinie. Jede zugeordnete Zugriffsrichtlinie hat ihren eigenen Geltungsbereich.

> **Note**
>
> Wenn Sie später den Geltungsbereich einer zugehörigen Zugriffsrichtlinie ändern möchten, können Sie den vorherigen Befehl erneut ausführen und dabei den neuen Bereich angeben. Wenn Sie also beispielsweise *my-namespace2* entfernen möchten, führen Sie den Befehl erneut aus und verwenden Sie dabei nur **type=namespace,namespaces=*my-namespace1***. Wenn Sie den Bereich von **namespace** in **cluster** ändern möchten, führen Sie den Befehl erneut aus und verwenden Sie dabei **type=cluster**, um **type=namespace,namespaces=*my-namespace1*,*my-namespace2*** zu entfernen.

So heben Sie die Zuordnung zwischen einer Zugriffsrichtlinie und einem Zugriffseintrag auf

1. Ermitteln Sie, welche Zugriffsrichtlinien einem Zugriffseintrag zugeordnet sind.

```
aws eks list-associated-access-policies --cluster-name my-cluster --principal-
arn arn:aws:iam::111122223333:role/my-role
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "clusterName": "my-cluster",
    "principalArn": "arn:aws:iam::111122223333",
    "associatedAccessPolicies": [
        {
            "policyArn": "arn:aws:eks::aws:cluster-access-
policy/AmazonEKSViewPolicy",
            "accessScope": {
                "type": "cluster",
                "namespaces": []
```

```
            },
            "associatedAt": "2023-04-17T15:25:21.675000-04:00",
            "modifiedAt": "2023-04-17T15:25:21.675000-04:00"
        },
        {
            "policyArn": "arn:aws:eks::aws:cluster-access-
policy/AmazonEKSAdminPolicy",
            "accessScope": {
                "type": "namespace",
                "namespaces": [
                    "my-namespace1",
                    "my-namespace2"
                ]
            },
            "associatedAt": "2023-04-17T15:02:06.511000-04:00",
            "modifiedAt": "2023-04-17T15:02:06.511000-04:00"
        }
    ]
}
```

Im vorherigen Beispiel verfügt der IAM-Prinzipal für diesen Zugriffseintrag über Leseberechtigungen für alle Namespaces im Cluster und über Administratorberechtigungen für zwei Kubernetes-Namespaces.

2. Heben Sie die Zuordnung zwischen einer Zugriffsrichtlinie und einem Zugriffseintrag auf. In diesem Beispiel wird die Zuordnung zwischen der Richtlinie `AmazonEKSAdminPolicy` und einem Zugriffseintrag aufgehoben. Die in der Zugriffsrichtlinie `AmazonEKSViewPolicy` enthaltenen Berechtigungen für Objekte in den Namespaces *my-namespace1* und *my-namespace2* bleiben für den IAM-Prinzipal allerdings erhalten, da die Zuordnung zwischen dieser Zugriffsrichtlinie und dem Zugriffseintrag nicht aufgehoben wird.

```
aws eks disassociate-access-policy --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/my-role \
    --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy
```

Berechtigungen von Zugriffsrichtlinien

Zugriffsrichtlinien beinhalten Regeln (`rules`), die Kubernetes-Verben (`verbs`; Berechtigungen) und Ressourcen (`resources`) enthalten. Zugriffsrichtlinien beinhalten keine IAM-Berechtigungen oder -Ressourcen. Ähnlich wie bei Kubernetes-Objekten vom Typ `Role` und `ClusterRole`

beinhalten Zugriffsrichtlinien nur Regeln (`rules`) vom Typ „Zulassen“ (`allow`). Der Inhalt einer Zugriffsrichtlinie kann nicht geändert werden. Sie können keine eigenen Zugriffsrichtlinien erstellen. Wenn die Berechtigungen in den Zugriffsrichtlinien Ihre Anforderungen nicht erfüllen, können Sie Kubernetes-RBAC-Objekte erstellen und Gruppennamen für Ihre Zugriffseinträge angeben. Weitere Informationen finden Sie unter Erstellen von Zugriffseinträgen. Die in Zugriffsrichtlinien enthaltenen Berechtigungen sind vergleichbar mit den Berechtigungen in den für Benutzer vorgesehenen Kubernetes-Clusterrollen. Weitere Informationen finden Sie in der Dokumentation zu Kubernetes unter User-facing roles.

Wählen Sie eine beliebige Zugriffsrichtlinie aus, um ihren Inhalt anzuzeigen. Die Zeilen der einzelnen Tabellen in den Zugriffsrichtlinien sind jeweils separate Regeln.

AmazonEKS AdminPolicy

Diese Zugriffsrichtlinie umfasst Berechtigungen, die einem IAM-Prinzipal die meisten Berechtigungen für Ressourcen erteilen. Wenn diese Zugriffsrichtlinie einem Zugriffseintrag zugeordnet ist, umfasst der Zugriffsbereich in der Regel mindestens einen Kubernetes-Namespace. Wenn ein IAM-Prinzipal über Administratorzugriff auf alle Ressourcen in Ihrem Cluster verfügen soll, ordnen Sie die Zugriffsrichtlinie AmazonEKS ClusterAdminPolicy stattdessen Ihrem Zugriffseintrag zu.

ARN – `arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy`

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
`apps`	`daemonsets` , `deployments` , `deployments/rollback` , `deployments/scale` , `replicasets` , `replicasets/scale`, `statefulsets` , `statefulsets/scale`	`create`, `delete`, `deletecollection` , `patch`, `update`
`apps`	`controllerrevisions` , `daemonsets` , `daemonsets/status` , `deployments` , `deployments/scale` , `deployments/status` ,	`get`, `list`, `watch`

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
	replicasets , replicasets/scale , replicasets/status , statefulsets , statefulsets/scale , statefulsets/status	
authorization.k8s.io	localsubjectaccessreviews	create
autoscaling	horizontalpodautoscalers	create, delete, deletecollection , patch, update
autoscaling	horizontalpodautoscalers , horizontalpodautoscalers/status	get, list, watch
batch	cronjobs, jobs	create, delete, deletecollection , patch, update
batch	cronjobs, cronjobs/status , jobs, jobs/status	get, list, watch
discovery.k8s.io	endpointslices	get, list, watch
extensions	daemonsets , deployments , deployments/rollback , deployments/scale , ingresses , networkpolicies , replicasets , replicasets/scale , replicationcontrollers/scale	create, delete, deletecollection , patch, update

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
extensions	daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , ingresses , ingresses/status , networkpolicies , replicasets , replicasets/scale , replicasets/status , replicationcontrollers/scale	get, list, watch
networking.k8s.io	ingresses , ingresses/status , networkpolicies	get, list, watch
networking.k8s.io	ingresses , networkpolicies	create, delete, deletecollection , patch, update
policy	poddisruptionbudgets	create, delete, deletecollection , patch, update
policy	poddisruptionbudgets , poddisruptionbudgets/status	get, list, watch
rbac.authorization.k8s.io	rolebindings , roles	create, delete, deletecollection , get, list, patch, update, watch

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
	configmaps , endpoints , persistentvolumeclaims , persistentvolumeclaims/status , pods, replicationcontrollers , replicationcontrollers/scale , serviceaccounts , services, services/status	get,list, watch
	pods/attach , pods/exec , pods/portforward , pods/proxy , secrets, services/proxy	get, list, watch
	configmaps , events, persistentvolumeclaims , replicationcontrollers , replicationcontrollers/scale , secrets, serviceaccounts , services, services/proxy	create, delete, deletecollection , patch, update
	pods, pods/attach , pods/exec , pods/portforward , pods/proxy	create, delete, deletecollection , patch, update
	serviceaccounts	impersonate

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
	`bindings`, `events`, `limitranges` , `namespaces/status` , `pods/log`, `pods/status` , `replicationcontrollers/status` , `resourcequotas` , `resourcequotas/status`	`get`, `list`, `watch`
	`namespaces`	`get`,`list`, `watch`

AmazonEKS ClusterAdminPolicy

Diese Zugriffsrichtlinie umfasst Berechtigungen, die einem IAM-Prinzipaladministrator Zugriff auf einen Cluster gewähren. Wenn diese Zugriffsrichtlinie einem Zugriffseintrag zugeordnet ist, umfasst der Zugriffsbereich in der Regel keinen Kubernetes-Namespace, sondern den Cluster. Wenn der Administratorbereich eines IAM-Prinzipals stärker eingeschränkt sein soll, können Sie stattdessen Ihrem Zugriffseintrag ggf. die Zugriffsrichtlinie AmazonEKS AdminPolicy zuordnen.

ARN – `arn:aws:eks::aws:cluster-access-policy/AmazonEKSClusterAdminPolicy`

Kubernetes-API-Gruppen	Kubernetes nonResourceURLs	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
*		*	*
	*		*

AmazonEKS AdminViewPolicy

Diese Zugriffsrichtlinie umfasst Berechtigungen, die einem IAM-Prinzipal Zugriff auf die Liste/Anzeige aller Ressourcen in einem Cluster gewähren. Beachten Sie, dass dies auch Secrets beinhaltetKubernetes.

ARN – `arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminViewPolicy`

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
*	*	get, list, watch

AmazonEKS EditPolicy

Diese Zugriffsrichtlinie umfasst Berechtigungen, die einem IAM-Prinzipal die Bearbeitung der meisten Kubernetes-Ressourcen ermöglichen.

ARN – arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
apps	daemonsets , deployments , deployments/rollback , deployments/scale , replicasets , replicasets/scale, statefulsets , statefulsets/scale	create, delete, deletecollection , patch, update
apps	controllerrevisions , daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , replicasets , replicasets/scale , replicasets/status , statefulsets , statefulsets/scale , statefulsets/status	get, list, watch
autoscaling	horizontalpodautoscalers , horizonta	get, list, watch

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
	lpodautoscalers/status	
autoscaling	horizontalpodautoscalers	create, delete, deletecollection , patch, update
batch	cronjobs, jobs	create, delete, deletecollection , patch, update
batch	cronjobs, cronjobs/status , jobs, jobs/status	get, list, watch
discovery.k8s.io	endpointslices	get, list, watch
extensions	daemonsets , deployments , deployments/rollback , deployments/scale , ingresses , networkpolicies , replicasets , replicasets/scale , replicationcontrollers/scale	create, delete, deletecollection , patch, update
extensions	daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , ingresses , ingresses/status , networkpolicies , replicasets , replicasets/scale , replicasets/status , replicationcontrollers/scale	get, list, watch

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
networking.k8s.io	ingresses , networkpolicies	create, delete, deletecollection , patch, update
networking.k8s.io	ingresses , ingresses/status , networkpolicies	get, list, watch
policy	poddisruptionbudgets	create, delete, deletecollection , patch, update
policy	poddisruptionbudgets , poddisruptionbudgets/status	get, list, watch
	namespaces	get, list, watch
	pods/attach , pods/exec , pods/portforward , pods/proxy , secrets, services/proxy	get, list, watch
	serviceaccounts	impersonate
	pods, pods/attach , pods/exec , pods/portforward , pods/proxy	create, delete, deletecollection , patch, update

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
	configmaps , events, persistentvolumeclaims , replicationcontrollers , replicationcontrollers/scale , secrets, serviceaccounts , services, services/proxy	create, delete, deletecollection , patch, update
	configmaps , endpoints , persistentvolumeclaims , persistentvolumeclaims/status , pods, replicationcontrollers , replicationcontrollers/scale , serviceaccounts , services, services/status	get, list, watch
	bindings, events, limitranges , namespaces/status , pods/log, pods/status , replicationcontrollers/status , resourcequotas , resourcequotas/status	get, list, watch

AmazonEKS ViewPolicy

Diese Zugriffsrichtlinie umfasst Berechtigungen, die einem IAM-Prinzipal die Anzeige der meisten Kubernetes-Ressourcen ermöglichen.

ARN – arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
apps	controllerrevisions , daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , replicasets , replicasets/scale , replicasets/status , statefulsets , statefulsets/scale , statefulsets/status	get, list, watch
autoscaling	horizontalpodautoscalers , horizontalpodautoscalers/status	get, list, watch
batch	cronjobs, cronjobs/status , jobs, jobs/status	get, list, watch
discovery.k8s.io	endpointslices	get, list, watch
extensions	daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , ingresses , ingresses/status , networkpolicies , replicasets , replicasets/scale , replicasets/status	get, list, watch

Kubernetes-API-Gruppen	Kubernetes-Ressourcen	Kubernetes-Verben (Berechtigungen)
	, replicationcontrollers/scale	
networking.k8s.io	ingresses , ingresses/status , networkpolicies	get, list, watch
policy	poddisruptionbudgets , poddisruptionbudgets/status	get, list, watch
	configmaps , endpoints , persistentvolumeclaims , persistentvolumeclaims/status , pods, replicationcontrollers , replicationcontrollers/scale , serviceaccounts , services, services/status	get, list, watch
	bindings, events, limitranges , namespaces/status , pods/log, pods/status , replicationcontrollers/status , resourcequotas , resourcequotas/status	get, list, watch
	namespaces	get, list, watch

Aktualisierungen für Zugriffsrichtlinien

Sehen Sie sich an, welche Aktualisierungen für Zugriffsrichtlinien seit ihrer Einführung vorgenommen wurden. Wenn Sie automatisch über Änderungen an dieser Seite benachrichtigt werden möchten, abonnieren Sie den RSS-Feed auf der Dokumentverlaufsseite von Amazon EKS.

Änderung	Beschreibung	Datum
Add `AmazonEKS AdminView Policy`	Fügen Sie eine neue Richtlinie für erweiterten Anzeigezugriff hinzu, einschließlich Ressourcen wie Secrets.	23. April 2024
Zugriffsrichtlinien wurden eingeführt.	In Amazon EKS wurden Zugriffsrichtlinien eingeführt.	29. Mai 2023

Migrieren vorhandener **`aws-auth ConfigMap`**-Einträge zu Zugriffseinträgen

Wenn Sie Einträge zu `aws-auth ConfigMap` für Ihren Cluster hinzugefügt haben, empfiehlt es sich, Zugriffseinträge für die vorhandenen Einträge in `aws-auth ConfigMap` zu erstellen. Nach der Erstellung der Zugriffseinträge können Sie die Einträge aus `ConfigMap` entfernen. Einträgen in `aws-auth ConfigMap` können keine Zugriffsrichtlinien zugeordnet werden. Wenn Sie Ihren IAM-Prinzipalen Zugriffsrichtlinien zuordnen möchten, müssen Sie Zugriffseinträge erstellen.

> **Important**
>
> Entfernen Sie keine `aws-auth ConfigMap`-Einträge, die von Amazon EKS erstellt wurden, als Sie Ihrem Cluster eine verwaltete Knotengruppe oder ein Fargate-Profil hinzugefügt haben. Wenn Sie Einträge entfernen, die von Amazon EKS in `ConfigMap` erstellt wurden, funktioniert Ihr Cluster nicht ordnungsgemäß. Sie können allerdings alle Einträge für selbstverwaltete Knotengruppen entfernen, nachdem Sie Zugriffseinträge für sie erstellt haben.

Voraussetzungen

- Vertrautheit mit Zugriffseinträgen und Zugriffsrichtlinien. Weitere Informationen finden Sie unter Zugangseinträge verwalten und Zuordnen von Zugriffsrichtlinien zu Zugriffseinträgen und Aufheben dieser Zuordnung.
- Ein Cluster mit einer Plattformversion, die mindestens den Versionen entspricht, die in den Voraussetzungen des Themas Zulassen des Zugriffs auf Kubernetes-Objekte in Ihrem Amazon EKS-Cluster durch IAM-Rollen oder -Benutzer aufgeführt sind.
- Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.
- Kubernetes-Berechtigungen zum Ändern von `aws-auth ConfigMap` im `kube-system`-Namespace.
- Eine AWS Identity and Access Management Rolle oder ein Benutzer mit den folgenden Berechtigungen: `CreateAccessEntry` und`ListAccessEntries`. Weitere Informationen finden Sie in der Service-Authorization-Referenz unter Von Amazon Elastic Kubernetes Service definierte Aktionen.

So migrieren Sie einen Eintrag von **`aws-auth ConfigMap`** zu einem Zugriffseintrag

1. Sehen Sie sich die vorhandenen Einträge in `aws-auth ConfigMap` an. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters.

   ```
   eksctl get iamidentitymapping --cluster my-cluster
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   ARN
                USERNAME                                GROUPS
                                ACCOUNT
   arn:aws:iam::111122223333:role/EKS-my-cluster-Admins
                Admins                                  system:masters
   arn:aws:iam::111122223333:role/EKS-my-cluster-my-namespace-Viewers
                my-namespace-Viewers                    Viewers
   arn:aws:iam::111122223333:role/EKS-my-cluster-self-managed-ng-1
                                system:node:{{EC2PrivateDNSName}}
    system:bootstrappers,system:nodes
   ```

```
arn:aws:iam::111122223333:user/my-user
             my-user
arn:aws:iam::111122223333:role/EKS-my-cluster-fargateprofile1
                            system:node:{{SessionName}}
 system:bootstrappers,system:nodes,system:node-proxier
arn:aws:iam::111122223333:role/EKS-my-cluster-managed-ng
                            system:node:{{EC2PrivateDNSName}}
 system:bootstrappers,system:nodes
```

2. Erstellen Sie Zugriffseinträge für alle von Ihnen erstellten `ConfigMap`-Einträge aus der vorherigen Ausgabe. Achten Sie beim Erstellen der Zugriffseinträge darauf, für `ARN`, `USERNAME`, `GROUPS` und `ACCOUNT` die gleichen Werte anzugeben, die in der Ausgabe zurückgegeben wurden. Im Falle der Beispielausgabe würden Sie Zugriffseinträge für alle Einträge außer den letzten beiden Einträgen erstellen, da diese von Amazon EKS für ein Fargate-Profil bzw. für eine verwaltete Knotengruppe erstellt wurden.
3. Löschen Sie die Einträge aus `ConfigMap` für alle von Ihnen erstellten Zugriffseinträge. Wenn Sie den Eintrag nicht aus `ConfigMap` löschen, wird der `ConfigMap`-Eintrag durch die Einstellungen für den Zugriffseintrag für den IAM-Prinzipal-ARN außer Kraft gesetzt. Ersetzen Sie *111122223333* durch Ihre AWS-Konto ID und *eks-my-cluster-my-namespace-viewers* durch den Namen der Rolle im Eintrag in Ihrem. `ConfigMap` Wenn es sich bei dem zu entfernenden Eintrag nicht um einen Eintrag für eine IAM-Rolle, sondern um einen Eintrag für einen IAM-Benutzer handelt, ersetzen Sie **`role`** durch **`user`** und *EKS-my-cluster-my-namespace-Viewers* durch den Benutzernamen.

```
eksctl delete iamidentitymapping --arn arn:aws:iam::111122223333:role/EKS-my-
cluster-my-namespace-Viewers --cluster my-cluster
```

IAM-Prinzipal-Zugriff auf Ihrem Cluster aktivieren

> **Important**
>
> Das `aws-auth` ConfigMap ist veraltet. Die empfohlene Methode zur Verwaltung des Zugriffs auf Kubernetes APIs ist Access Entries.

Zugriff auf Ihren Cluster mit IAM-Prinzipal wird durch AWS -IAM-Authenticator für Kubernetes aktiviert, das auf der Amazon-EKS-Steuerebene läuft. Der Authenticator erhält seine

Konfigurationsinformationen von `aws-auth ConfigMap`. Alle `aws-auth ConfigMap`-Einstellungen finden Sie unter Vollständiges Konfigurationsformat auf GitHub.

Hinzufügen von IAM-Prinzipal zu Ihrem Amazon EKS-Cluster

Wenn Sie einen Amazon-EKS-Cluster erstellen, werden dem IAM-Prinzipal, der den Cluster erstellt, automatisch `system:masters`-Berechtigungen in der rollenbasierten Zugriffssteuerungs(RBAC)-Konfiguration des Clusters in der Amazon-EKS-Steuerebene erteilt. Dieser Prinzipal wird in keiner sichtbaren Konfiguration angezeigt. Achten Sie daher darauf, welcher Prinzipal den Cluster ursprünglich erstellt hat. Um zusätzlichen IAM-Prinzipalen die Möglichkeit zu geben, mit Ihrem Cluster zu interagieren, müssen Sie die `aws-auth ConfigMap` innerhalb von Kubernetes bearbeiten und ein Kubernetes `rolebinding` oder `clusterrolebinding` mit dem Namen einer `group` erstellen, den Sie in der `aws-auth ConfigMap` angeben.

> **Note**
>
> Weitere Informationen zur Kubernetes rollenbasierten Zugriffssteuerungs (RBAC)-Konfiguration finden Sie unter Verwenden der RBAC-Autorisierung in der Kubernetes-Dokumentation.

So fügen Sie einem Amazon-EKS-Cluster einen IAM-Prinzipal hinzu

1. Bestimmen Sie, welche Anmeldeinformationen `kubectl` für den Zugriff auf Ihren Cluster verwendet. Amazon Resource Name (ARN)Auf Ihrem Computer können Sie mit dem folgenden Befehl sehen, welche Anmeldeinformationen `kubectl` verwendet. Ersetzen Sie ***~/.kube/config*** durch den Pfad zu Ihrer `kubeconfig`-Datei, wenn Sie nicht den Standardpfad verwenden.

   ```
   cat ~/.kube/config
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   [...]
   contexts:
   - context:
       cluster: my-cluster.region-code.eksctl.io
       user: admin@my-cluster.region-code.eksctl.io
     name: admin@my-cluster.region-code.eksctl.io
   ```

```
current-context: admin@my-cluster.region-code.eksctl.io
[...]
```

In der vorherigen Beispielausgabe werden die Anmeldeinformationen für einen Benutzer mit dem Namen *admin* für einen Cluster mit dem Namen *my-cluster* konfiguriert. Wenn dies der Benutzer ist, der den Cluster erstellt hat, hat er bereits Zugriff auf Ihren Cluster. Wenn es nicht der Benutzer ist, der den Cluster erstellt hat, müssen Sie die verbleibenden Schritte ausführen, um den Clusterzugriff für andere IAM-Prinzipale zu aktivieren. Bewährte Methoden für IAM empfehlen, dass Sie Rollen statt Benutzern Berechtigungen gewähren. Mit dem folgenden Befehl können Sie sehen, welche anderen Prinzipale derzeit Zugriff auf Ihren Cluster haben:

```
kubectl describe -n kube-system configmap/aws-auth
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:         aws-auth
Namespace:    kube-system
Labels:       <none>
Annotations:  <none>

Data
====
mapRoles:
----
- groups:
  - system:bootstrappers
  - system:nodes
  rolearn: arn:aws:iam::111122223333:role/my-node-role
  username: system:node:{{EC2PrivateDNSName}}


BinaryData
====

Events:  <none>
```

Das vorherige Beispiel ist ein standardmäßiger `aws-auth` `ConfigMap`. Nur die Knoten-Instance-Rolle hat Zugriff auf den Cluster.

2. Stellen Sie sicher, dass Kubernetes-`roles` und `rolebindings` oder `clusterroles` und `clusterrolebindings` vorhanden sind, denen Sie IAM-Prinzipale zuordnen können. Weitere

Informationen über diese Ressourcen finden Sie unter Using RBAC Authorization (RBAC-Autorisierung verwenden) in der Kubernetes-Dokumentation.

1. Zeigen Sie Ihre vorhandenen Kubernetes `roles` oder `clusterroles` an. `Roles` sind auf einen `namespace` ausgelegt, aber `clusterroles` sind auf den Cluster ausgelegt.

   ```
   kubectl get roles -A
   ```

   ```
   kubectl get clusterroles
   ```

2. Zeigen Sie die Details einer beliebigen `role` oder `clusterrole` an, die in der vorherigen Ausgabe zurückgegeben wurden, und bestätigen Sie, dass Sie die Berechtigungen (`rules`) hat, die Ihre IAM-Prinzipale in Ihrem Cluster haben sollen.

   Ersetzen Sie *`role-name`* durch einen `role`-Namen, den der vorherige Befehl zurückgegeben hat. Ersetzen Sie *`kube-system`* durch den Namespace der `role`.

   ```
   kubectl describe role role-name -n kube-system
   ```

   Ersetzen Sie *`cluster-role-name`* durch einen `clusterrole`-Namen, den der vorherige Befehl zurückgegeben hat.

   ```
   kubectl describe clusterrole cluster-role-name
   ```

3. Zeigen Sie Ihre vorhandenen Kubernetes `rolebindings` oder `clusterrolebindings` an. `Rolebindings` sind auf einen `namespace` ausgelegt, aber `clusterrolebindings` sind auf den Cluster ausgelegt.

   ```
   kubectl get rolebindings -A
   ```

   ```
   kubectl get clusterrolebindings
   ```

4. Zeigen Sie die Details einer beliebigen `rolebinding` oder `clusterrolebinding` an und bestätigen Sie, dass sie eine `role` oder `clusterrole` aus dem vorherigen Schritt hat, die als `roleRef` aufgeführt wird, und einen Gruppennamen, der für `subjects` aufgeführt wird.

Ersetzen Sie *role-binding-name* durch einen `rolebinding`-Namen, den der vorherige Befehl zurückgegeben hat. Ersetzen SIe *kube-system* mit dem `namespace` der `rolebinding`.

```
kubectl describe rolebinding role-binding-name -n kube-system
```

Eine Beispielausgabe sieht wie folgt aus.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: eks-console-dashboard-restricted-access-role-binding
  namespace: default
subjects:
- kind: Group
  name: eks-console-dashboard-restricted-access-group
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: eks-console-dashboard-restricted-access-role
  apiGroup: rbac.authorization.k8s.io
```

Ersetzen Sie *cluster-role-binding-name* durch einen `clusterrolebinding`-Namen, den der vorherige Befehl zurückgegeben hat.

```
kubectl describe clusterrolebinding cluster-role-binding-name
```

Eine Beispielausgabe sieht wie folgt aus.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: eks-console-dashboard-full-access-binding
subjects:
- kind: Group
  name: eks-console-dashboard-full-access-group
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: eks-console-dashboard-full-access-clusterrole
```

```
  apiGroup: rbac.authorization.k8s.io
```

3. Bearbeiten Sie `aws-auth ConfigMap`. Sie können ein Tool wie `eksctl` verwenden, um die `ConfigMap` zu aktualisieren, oder Sie können sie durch manuelle Bearbeitung aktualisieren.

> **Important**
>
> Wir empfehlen die Verwendung von `eksctl` oder einem anderen Tool, um die `ConfigMap` zu bearbeiten. Weitere Informationen zu anderen Tools, die Sie verwenden können, finden Sie unter Verwenden von Tools zur Änderung von `aws-authConfigMap` in den Best-Practice-Leitfäden zu Amazon EKS. Ist `aws-auth ConfigMap` falsch formatiert, können Sie den Zugriff auf Ihren Cluster verlieren.

eksctl

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

1. Zeigen Sie die aktuellen Mappings in der `ConfigMap` an. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters. Ersetzen Sie es *`region-code`* durch AWS-Region das, in dem sich Ihr Cluster befindet.

```
eksctl get iamidentitymapping --cluster my-cluster --region=region-code
```

   Eine Beispielausgabe sieht wie folgt aus.

```
ARN
                    USERNAME                                GROUPS
              ACCOUNT
arn:aws:iam::111122223333:role/eksctl-my-cluster-my-nodegroup-
NodeInstanceRole-1XLS7754U3ZPA    system:node:{{EC2PrivateDNSName}}
 system:bootstrappers,system:nodes
```

2. Fügen Sie ein Mapping für eine Rolle hinzu. Ersetzen Sie *`my-role`* durch den Namen Ihrer Rolle. Ersetzen Sie *`eks-console-dashboard-full-access-group`* durch den Namen der Gruppe, die in Ihrem Kubernetes `RoleBinding`- oder

`ClusterRoleBinding`-Objekt angegeben wurde. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Sie können *admin* durch einen beliebigen Namen ersetzen, den Sie wählen.

```
eksctl create iamidentitymapping --cluster my-cluster --region=region-code \
    --arn arn:aws:iam::111122223333:role/my-role --username admin --group eks-console-dashboard-full-access-group \
    --no-duplicate-arns
```

> **Important**
>
> Der Rollen-ARN darf keinen Pfad wie `role/my-team/developers/my-role` enthalten. Das Format des ARN muss `arn:aws:iam::111122223333:role/my-role` sein. In diesem Beispiel muss `my-team/developers/` entfernt werden.

Eine Beispielausgabe sieht wie folgt aus.

```
[...]
2022-05-09 14:51:20 [#]  adding identity "arn:aws:iam::111122223333:role/my-role" to auth ConfigMap
```

3. Fügen Sie ein Mapping für einen Benutzer hinzu. Bewährte Methoden für IAM empfehlen, dass Sie Rollen statt Benutzern Berechtigungen gewähren. Ersetzen Sie *my-user* durch den Benutzernamen. Ersetzen Sie *eks-console-dashboard-restricted-access-group* durch den Namen der Gruppe, die in Ihrem Kubernetes `RoleBinding`- oder `ClusterRoleBinding`-Objekt angegeben wurde. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Sie können *my-user* durch einen beliebigen Namen ersetzen, den Sie wählen.

```
eksctl create iamidentitymapping --cluster my-cluster --region=region-code \
    --arn arn:aws:iam::111122223333:user/my-user --username my-user --group eks-console-dashboard-restricted-access-group \
    --no-duplicate-arns
```

Eine Beispielausgabe sieht wie folgt aus.

```
[...]
2022-05-09 14:53:48 [#]  adding identity "arn:aws:iam::111122223333:user/my-
user" to auth ConfigMap
```

4. Zeigen Sie wieder die Mappings in der `ConfigMap` an.

```
eksctl get iamidentitymapping --cluster my-cluster --region=region-code
```

Eine Beispielausgabe sieht wie folgt aus.

```
ARN
                  USERNAME                                GROUPS
                    ACCOUNT
arn:aws:iam::111122223333:role/eksctl-my-cluster-my-nodegroup-
NodeInstanceRole-1XLS7754U3ZPA    system:node:{{EC2PrivateDNSName}}
 system:bootstrappers,system:nodes
arn:aws:iam::111122223333:role/admin
                      my-role                                 eks-console-
dashboard-full-access-group
arn:aws:iam::111122223333:user/my-user
                      my-user                                 eks-console-
dashboard-restricted-access-group
```

Edit ConfigMap manually

1. Öffnen Sie `ConfigMap` zum Bearbeiten.

```
kubectl edit -n kube-system configmap/aws-auth
```

> **Note**
>
> Wenn die Fehlermeldung „`Error from server (NotFound): configmaps "aws-auth" not found`“ angezeigt wird, verwenden Sie das Verfahren in Anwenden von `aws-auth` `ConfigMap` auf Ihren Cluster, um die lokal gespeicherte `ConfigMap` anzuwenden.

2. Fügen Sie Ihre IAM-Prinzipale zu der `ConfigMap` hinzu. Eine IAM-Gruppe ist kein IAM-Prinzipal und kann daher nicht zu der `ConfigMap` hinzugefügt werden.

- So fügen Sie eine IAM-Rolle hinzu (z. B. für Verbundbenutzer): Fügen Sie die Rollendetails zum Abschnitt `mapRoles` der `ConfigMap` unter `data` hinzu. Fügen Sie diesen Abschnitt hinzu, wenn er nicht bereits in der Datei vorhanden sind. Jeder Eintrag unterstützt die folgenden Parameter:
  - rolearn: Der ARN der IAM-Rolle, den Sie hinzufügen möchten. Dieser Wert darf keinen Pfad enthalten. Sie können beispielsweise keinen ARN wie `arn:aws:iam::111122223333:role/my-team/developers/role-name` angeben. Der ARN muss stattdessen `arn:aws:iam::111122223333:role/role-name` sein.
  - username: Der Benutzername in Kubernetes für die Zuweisung zur IAM-Rolle.
  - Gruppen: Die Gruppe oder Liste der Kubernetes-Gruppen, denen die Rolle zugeordnet werden soll. Die Gruppe kann eine Standardgruppe oder eine Gruppe sein, die in einer `clusterrolebinding` oder `rolebinding` angegeben ist. Weitere Informationen finden Sie unter Default Roles and Role Bindings in der Kubernetes-Dokumentation.
- Um einen IAM-Benutzer hinzuzufügen: Bewährte Methoden von IAM empfehlen, dass Sie Rollen statt Benutzern Berechtigungen gewähren. Fügen Sie die Benutzerdetails zum `mapUsers`-Abschnitt der `ConfigMap` unter `data` hinzu. Fügen Sie diesen Abschnitt hinzu, wenn er nicht bereits in der Datei vorhanden sind. Jeder Eintrag unterstützt die folgenden Parameter:
  - userarn: Die ARN des IAM-Benutzers, den Sie hinzufügen möchten.
  - username: Der Benutzername in Kubernetes für die Zuweisung zum IAM-Benutzer.
  - Gruppen: Die Gruppe oder Liste der Kubernetes-Gruppen, denen die Benutzer zugeordnet werden soll. Die Gruppe kann eine Standardgruppe oder eine Gruppe sein, die in einer `clusterrolebinding` oder `rolebinding` angegeben ist. Weitere Informationen finden Sie unter Default Roles and Role Bindings in der Kubernetes-Dokumentation.

Beispiel: Der folgende YAML-Block enthält:

- Einen `mapRoles`-Abschnitt, der die IAM-Knoten-Instance Kubernetes-Gruppen zuordnet, sodass sich Knoten selbst bei dem Cluster und der `my-console-viewer-role`-IAM-Rolle registrieren können, die einer Kubernetes-Gruppe zugeordnet ist, die alle Kubernetes-Ressourcen für alle Cluster anzeigen kann. Eine Liste der erforderlichen IAM- und Kubernetes-Gruppenberechtigungen für die `my-console-viewer-role`-IAM-Rolle finden Sie unter Erforderliche Berechtigungen.

- Ein `mapUsers` Abschnitt, der den `admin` IAM-Benutzer aus dem AWS Standardkonto der `system:masters` Kubernetes Gruppe und den `my-user` Benutzer aus einem anderen AWS Konto zuordnet, das einer Kubernetes Gruppe zugeordnet ist, die Kubernetes Ressourcen für einen bestimmten Namespace anzeigen kann. Eine Liste der erforderlichen IAM- und Kubernetes-Gruppenberechtigungen für die `my-user`-IAM-Benutzer finden Sie unter Erforderliche Berechtigungen.

Fügen Sie nach Bedarf Zeilen hinzu oder entfernen Sie sie und ersetzen Sie alle *example values* durch eigene Werte.

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this
 file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
      rolearn: arn:aws:iam::111122223333:role/my-role
      username: system:node:{{EC2PrivateDNSName}}
    - groups:
      - eks-console-dashboard-full-access-group
      rolearn: arn:aws:iam::111122223333:role/my-console-viewer-role
      username: my-console-viewer-role
  mapUsers: |
    - groups:
      - system:masters
      userarn: arn:aws:iam::111122223333:user/admin
      username: admin
    - groups:
      - eks-console-dashboard-restricted-access-group
      userarn: arn:aws:iam::444455556666:user/my-user
      username: my-user
```

3. Speichern Sie die Datei und beenden Sie den Text-Editor.

Anwenden von **aws-authConfigMap** auf Ihren Cluster

`aws-auth ConfigMap` wird automatisch erstellt und auf Ihren Cluster angewendet, wenn Sie eine verwaltete Knotengruppe erstellen oder wenn Sie eine Knotengruppe mit `eksctl` erstellen. Sie wird anfänglich erstellt, um Knoten zu erlauben, Ihrem Cluster beizutreten, aber Sie verwenden diese `ConfigMap` auch, um rollenbasierten Zugriffssteuerungs(RBAC)-Zugriff auf IAM-Prinzipale hinzuzufügen. Wenn Sie keine selbstverwalteten Knoten gestartet und `aws-auth ConfigMap` nicht auf Ihren Cluster angewendet haben, können Sie das folgende Verfahren dafür verwenden.

Anwendung der **aws-authConfigMap** auf Ihren Cluster

1. Überprüfen Sie, ob Sie `aws-auth ConfigMap` bereits angewendet haben.

   ```
   kubectl describe configmap -n kube-system aws-auth
   ```

   Wenn die Fehlermeldung „`Error from server (NotFound): configmaps "aws-auth" not found`“ angezeigt wird, fahren Sie mit den folgenden Schritten fort, um die lokal gespeicherte `ConfigMap` anzuwenden.

2. Laden Sie die AWS Authenticator-Konfigurationsübersicht herunter, bearbeiten Sie sie und wenden Sie sie an.

   a. Laden Sie die Konfigurationszuordnung herunter.

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-
   eks/cloudformation/2020-10-29/aws-auth-cm.yaml
   ```

   b. Stellen Sie in der Datei `aws-auth-cm.yaml` `rolearn` auf den Amazon-Ressourcennamen (ARN) der IAM-Rolle ein, die Ihren Knoten zugeordnet ist. Hierzu können Sie einen Texteditor verwenden oder *`my-node-instance-role`* ersetzen und den folgenden Befehl ausführen:

   ```
   sed -i.bak -e 's|<ARN of instance role (not instance profile)>|my-node-
   instance-role|' aws-auth-cm.yaml
   ```

   Ändern Sie keine weiteren Zeilen in dieser Datei.

> **Important**
>
> Der Rollen-ARN darf keinen Pfad wie `role/my-team/developers/my-role` enthalten. Das Format des ARN muss `arn:aws:iam::111122223333:role/my-role` sein. In diesem Beispiel muss `my-team/developers/` entfernt werden.

Sie können die AWS CloudFormation Stack-Ausgaben für Ihre Knotengruppen überprüfen und nach den folgenden Werten suchen:

- InstanceRoleARN — Für Knotengruppen, die erstellt wurden mit `eksctl`
- NodeInstanceRolle — Für Knotengruppen, die mit von Amazon EKS bereitgestellten AWS CloudFormation Vorlagen erstellt wurden, in AWS Management Console

c. Wenden Sie die Konfiguration an. Die Ausführung dieses Befehls kann einige Minuten dauern.

```
kubectl apply -f aws-auth-cm.yaml
```

> **Note**
>
> Wenn Sie Autorisierungs- oder Ressourcenfehler erhalten, finden Sie weitere Informationen unter Nicht autorisiert oder Zugriff verweigert (`kubectl`) im Thema zur Fehlerbehebung.

3. Sehen Sie sich den Status Ihrer Knoten an und warten Sie, bis diese in den `Ready`-Status eintreten.

```
kubectl get nodes --watch
```

Geben Sie `Ctrl+C` ein, um zu einer Shell-Eingabeaufforderung zurückzukehren.

Authentifizieren Sie Benutzer für Ihren Cluster von einem OpenID Connect Identitätsanbieter

Amazon EKS unterstützt die Verwendung von OpenID Connect (OIDC) -Identitätsanbietern als Methode zur Authentifizierung von Benutzern in Ihrem Cluster. OIDCIdentitätsanbieter können zusammen mit oder als Alternative zu AWS Identity and Access Management (IAM) verwendet werden. Weitere Informationen zur Verwendung von IAM finden Sie unter the section called "Gewähren Sie Zugriff auf Kubernetes-APIs". Nachdem Sie die Authentifizierung für Ihren Cluster konfiguriert haben, können Sie Kubernetes-`roles` und `clusterroles` erstellen, um den Rollen Berechtigungen zuzuweisen, und dann die Rollen mithilfe von Kubernetes-`rolebindings` und `clusterrolebindings` an die Identitäten binden. Weitere Informationen finden Sie unter Using RBAC authorization in der Kubernetes-Dokumentation.

Überlegungen

- Sie können Ihrem Cluster einen einzelnen OIDC-Identitätsanbieter zuordnen.
- Kubernetes stellt keinen OIDC-Identitätsanbieter bereit. Sie können einen vorhandenen öffentlichen OIDC-Identitätsanbieter verwenden oder Ihren eigenen Identitätsanbieter ausführen. Eine Liste zertifizierter Anbieter finden Sie unter OpenID-Zertifizierung auf der OpenID-Website.
- Die Aussteller-URL des OIDC-Identitätsanbieters muss öffentlich zugänglich sein, damit Amazon EKS die Signaturschlüssel erkennen kann. Amazon EKS unterstützt keine OIDC-Identitätsanbieter mit selbstsignierten Zertifikaten.
- Sie können die IAM-Authentifizierung in Ihrem Cluster nicht deaktivieren, da sie weiterhin erforderlich ist, um einem Cluster Knoten hinzufügen zu können.
- Ein Amazon EKS-Cluster muss weiterhin von einem AWS IAM-Prinzipal und nicht von einem OIDC Identitätsanbieter-Benutzer erstellt werden. Dies liegt daran, dass der Cluster-Ersteller mit den Amazon-EKS-APIs interagiert und nicht mit den Kubernetes-APIs.
- OIDCBenutzer, die vom Identitätsanbieter authentifiziert wurden, werden im Audit-Protokoll des Clusters aufgeführt, wenn die CloudWatch Protokolle für die Kontrollebene aktiviert sind. Weitere Informationen finden Sie unter Aktivieren und Deaktivieren von Steuerebenenprotokollen.
- Sie können sich nicht AWS Management Console mit einem Konto eines Anbieters bei der anmelden. OIDC Sie können KubernetesRessourcen in der Konsole nur anzeigen, wenn Sie sich AWS Management Console mit einem AWS Identity and Access Management Konto bei der anmelden.

Zuordnen eines OIDC-Identitätsanbieters

Bevor Sie Ihrem Cluster einen OIDC-Identitätsanbieter zuordnen können, benötigen Sie die folgenden Informationen von Ihrem Anbieter:

URL des Ausstellers

Die URL des OIDC-Identitätsanbieters, die es dem API-Server ermöglicht, öffentliche Signierschlüssel zur Verifizierung von Token zu ermitteln. Die URL muss mit `https://` beginnen und sollte dem `iss`-Anspruch in den OIDC-ID-Token des Anbieters entsprechen. Gemäß dem OIDC-Standard sind Pfadkomponenten erlaubt, Abfrageparameter jedoch nicht. Normalerweise besteht die URL nur aus einem Hostnamen wie `https://server.example.org` oder `https://example.com`. Diese URL sollte auf die Ebene unterhalb von `.well-known/openid-configuration` verweisen und muss über das Internet öffentlich zugänglich sein.

Client-ID (auch als Zielgruppe bezeichnet)

Die ID für die Client-Anwendung, die Authentifizierungsanforderungen an den OIDC-Identitätsanbieter stellt.

Sie können einen Identitätsanbieter mit `eksctl` oder AWS Management Console zuordnen.

eksctl

So verwenden Sie **`eksctl`**, um Ihrem Cluster einen OIDC-Identitätsanbieter zuzuordnen

1. Erstellen Sie eine Datei mit dem Namen *`associate-identity-provider.yaml`* und dem folgenden Inhalt. Ersetzen Sie das *`example values`* durch Ihr eigenes. Die Werte im Abschnitt `identityProviders` erhalten Sie von Ihrem OIDC-Identitätsanbieter. Werte werden nur für die `name`-, `type`-, `issuerUrl`- und `clientId`-Einstellungen unter `identityProviders` benötigt.

```
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: your-region-code

identityProviders:
```

```
  - name: my-provider
    type: oidc
    issuerUrl: https://example.com
    clientId: kubernetes
    usernameClaim: email
    usernamePrefix: my-username-prefix
    groupsClaim: my-claim
    groupsPrefix: my-groups-prefix
    requiredClaims:
      string: string
    tags:
      env: dev
```

> **Important**
>
> Geben Sie weder `system:` noch einen Teil dieser Zeichenfolge für `groupsPrefix` oder `usernamePrefix` an.

2. Erstellen Sie den Anbieter.

```
eksctl associate identityprovider -f associate-identity-provider.yaml
```

3. Informationen zur Verwendung von `kubectl` für die Arbeit mit Ihrem Cluster und dem OIDC-Identitätsanbieter finden Sie in der Dokumentation zu Kubernetes unter Using `kubectl`.

AWS Management Console

Um Ihrem Cluster einen OIDC Identitätsanbieter zuzuordnen, verwenden Sie AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie Ihren Cluster und dann die Registerkarte Zugriff aus.
3. Wählen Sie im Abschnitt OIDCIdentity Providers die Option Associate Identity Provider aus.
4. Geben Sie auf der Seite OIDC-Identitätsanbieter zuordnen die folgenden Optionen ein bzw. wählen Sie sie aus und wählen Sie anschließend Zuordnen aus.
   - Geben Sie für Name einen eindeutigen Namen für den Anbieter ein.

- Geben Sie für Aussteller-URL die URL Ihres Anbieters ein. Diese URL muss über das Internet zugänglich sein.
- Geben Sie unter Client-ID die Client-ID des OIDC-Identitätsanbieters (auch als Zielgruppe bezeichnet) ein.
- Geben Sie für Benutzernamenanspruch den Anspruch ein, der als Benutzername verwendet werden soll.
- Geben Sie für Gruppenanspruch den Anspruch ein, der als Benutzergruppe verwendet werden soll.
- (Optional) Wählen Sie Erweiterte Optionen, geben Sie die folgenden Informationen ein oder wählen Sie sie aus.
  - Benutzernamen-Präfix – Geben Sie ein Präfix ein, das den Benutzernamenansprüchen vorangestellt wird. Das Präfix wird den Benutzernamensansprüchen vorangestellt, um Konflikte mit bestehenden Namen zu vermeiden. Wenn Sie keinen Wert angeben und der Benutzername ein anderer Wert als `email` ist, wird als Präfix standardmäßig der Wert für die Aussteller-URL verwendet. Sie können den Wert `-` verwenden, um alle Präfixe zu deaktivieren. Geben Sie weder `system:` noch einen Teil dieser Zeichenfolge an.
  - Gruppen-Präfix – Geben Sie ein Präfix ein, um Gruppenansprüchen voranzustellen. Das Präfix wird Gruppenansprüchen vorangestellt, um Konflikte mit vorhandenen Namen (wie z. B. `system: groups`) zu vermeiden. Der Wert `oidc:` erstellt beispielsweise Gruppennamen wie `oidc:engineering` und `oidc:infra`. Geben Sie weder `system:` noch einen Teil dieser Zeichenfolge an.
  - Erforderliche Ansprüche – Wählen Sie Anspruch hinzufügen aus, und geben Sie ein oder mehrere Schlüsselwertpaare ein, die erforderliche Ansprüche im Client-ID-Token beschreiben. Der Paris beschreibt erforderliche Ansprüche im ID-Token. Wenn festgelegt, wird überprüft, ob jeder Anspruch im ID-Token mit einem übereinstimmenden Wert vorhanden ist.

5. Informationen zur Verwendung von `kubectl` für die Arbeit mit Ihrem Cluster und dem OIDC-Identitätsanbieter finden Sie in der Dokumentation zu Kubernetes unter Using `kubectl`.

Aufheben der Zuordnung zwischen einem OIDC-Identitätsanbieter und Ihrem Cluster

Wenn Sie die Zuordnung zwischen einem OIDC-Identitätsanbieter und Ihrem Cluster aufheben, können im Anbieter enthaltene Benutzer nicht mehr auf den Cluster zugreifen. Sie können jedoch weiterhin mit IAM-Prinzipalen auf den Cluster zugreifen.

So verwenden Sie die AWS Management Console, um die Zuordnung zwischen einem OIDC-Identitätsanbieter und Ihrem Cluster aufzuheben

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im Abschnitt OIDC-Identitätsanbieter die Option Zuordnung aufheben aus, geben Sie den Namen des Identitätsanbieters ein und wählen Sie anschließend `Disassociate` aus.

Beispiel für eine IAM-Richtlinie

Wenn Sie verhindern möchten, dass ein OIDC-Identitätsanbieter einem Cluster zugeordnet wird, erstellen Sie die folgende IAM-Richtlinie und ordnen Sie sie den IAM-Konten Ihrer Amazon-EKS-Administratoren zu. Weitere Informationen finden Sie unter Erstellen von IAM-Richtlinien und Hinzufügen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch und Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Elastic Kubernetes Service in der Service-Autorisierungsreferenz.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "denyOIDC",
            "Effect": "Deny",
            "Action": [
                "eks:AssociateIdentityProviderConfig"
            ],
            "Resource": "arn:aws:eks:us-west-2.amazonaws.com:111122223333:cluster/*"

        },
        {
            "Sid": "eksAdmin",
            "Effect": "Allow",
            "Action": [
                "eks:*"
            ],
            "Resource": "*"
        }
    ]
}
```

Die folgende Beispielrichtlinie lässt die Zuordnung von OIDC-Identitätsanbietern zu, wenn `clientID` den Wert `kubernetes` und `issuerUrl` den Wert `https://cognito-idp.us-west-2amazonaws.com/*` hat.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCognitoOnly",
            "Effect": "Deny",
            "Action": "eks:AssociateIdentityProviderConfig",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-instance",
            "Condition": {
                "StringNotLikeIfExists": {
                    "eks:issuerUrl": "https://cognito-idp.us-west-2.amazonaws.com/*"
                }
            }
        },
        {
            "Sid": "DenyOtherClients",
            "Effect": "Deny",
            "Action": "eks:AssociateIdentityProviderConfig",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-instance",
            "Condition": {
                "StringNotEquals": {
                    "eks:clientId": "kubernetes"
                }
            }
        },
        {
            "Sid": "AllowOthers",
            "Effect": "Allow",
            "Action": "eks:*",
            "Resource": "*"
        }
    ]
}
```

Von Partnern validierte OIDC-Identitätsanbieter

Amazon EKS pflegt Beziehungen zu einem Netzwerk von Partnern, die Support für alternative kompatible OIDC-Identitätsanbieter bieten. Weitere Details zur Integration des Identitätsanbieters in der Dokumentation der folgenden Partner mit Amazon EKS.

Partner	Produkt	Dokumentation
PingIdentity	PingOne für Unternehmen	Installationsanleitungen

Amazon EKS bemüht sich, Ihnen eine große Auswahl an Optionen zu bieten, um alle Anwendungsfälle abzudecken. Wenn Sie einen kommerziell unterstützten OIDC-kompatiblen Identitätsanbieter entwickeln, der hier nicht aufgeführt ist, wenden Sie sich an unser Partnerteam unter aws-container-partners@amazon.com, um weitere Informationen zu erhalten.

Erstellen oder Aktualisieren einer **`kubeconfig`**-Datei für einen Amazon-EKS-Cluster

In diesem Thema erstellen Sie eine `kubeconfig`-Datei für Ihren Cluster (oder aktualisieren eine vorhandene).

Das `kubectl`-Befehlszeilentool verwendet Konfigurationsinformationen in `kubeconfig`-Dateien für die Kommunikation mit dem API-Server eines Clusters. Weitere Informationen finden Sie unter Organisieren des Cluster-Zugriffs mit kubeconfig-Dateien in der Kubernetes-Dokumentation.

Amazon EKS verwendet den Befehl `aws eks get-token` mit `kubectl` für die Cluster-Authentifizierung. Standardmäßig AWS CLI verwendet der dieselben Anmeldeinformationen, die mit dem folgenden Befehl zurückgegeben werden:

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

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter

oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.

- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Ein IAM-Benutzer oder eine Rolle mit Berechtigung zur Verwendung der API-Aktion `eks:DescribeCluster` für den angegebenen Cluster. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Amazon-EKS-Richtlinien. Wenn Sie für den Zugriff auf Ihren Cluster eine Identität Ihres eigenen OpenID Connect-Anbieters verwenden, erfahren Sie in der Dokumentation zu Kubernetes unter Using `kubectl`, wie Sie die Datei `kube config` erstellen oder aktualisieren.

`kubeconfig`-Datei automatisch erstellen

Voraussetzungen

- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

- Berechtigung zur Nutzung der `eks:DescribeCluster`-API-Aktion für den Cluster, den Sie angeben. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Amazon-EKS-Richtlinien.

Um Ihre **`kubeconfig`** Datei mit dem zu erstellen AWS CLI

1. Erstellen oder aktualisieren Sie eine `kubeconfig`-Datei für Ihren Cluster. *Ersetzen Sie den Regionalcode durch den Regionscode, in dem sich Ihr Cluster befindet AWS-Region , und ersetzen Sie my-cluster durch den Namen Ihres Clusters.*

   ```
   aws eks update-kubeconfig --region region-code --name my-cluster
   ```

   Standardmäßig wird die resultierende Konfigurationsdatei im Standard `kubeconfig`-Pfad (`.kube`) in Ihrem Stammverzeichnis erstellt oder mit einer vorhandenen `config`-Datei an diesem Speicherort zusammengeführt. Sie können mit der Option **`--kubeconfig`** einen anderen Pfad angeben.

   Sie können einen IAM-Rollen-ARN mit der Option **`--role-arn`** für die Authentifizierung verwenden, wenn Sie `kubectl`-Befehle ausgeben. Andernfalls wird der IAM-Prinzipal in Ihrer Standard- oder SDK-Anmeldeinformationskette verwendet. AWS CLI Sie können Ihre Standard AWS CLI - oder SDK-Identität anzeigen, indem Sie den `aws sts get-caller-identity` Befehl ausführen.

   Eine Übersicht über alle verfügbaren Optionen erhalten Sie durch Ausführen des Befehls `aws eks update-kubeconfig help` oder in der AWS CLI -Befehlsreferenz unter `update-kubeconfig`.
2. Testen Sie Ihre Konfiguration.

   ```
   kubectl get svc
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   NAME             TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
   svc/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   1m
   ```

Wenn Sie Autorisierungs- oder Ressourcenfehler erhalten, finden Sie weitere Informationen unter Nicht autorisiert oder Zugriff verweigert (`kubectl`) im Thema zur Fehlerbehebung.

Gewähren Sie Kubernetes-Workloads Zugriff auf die Verwendung von Service AWS Accounts Kubernetes

Ein Kubernetes-Servicekonto stellt eine Identität für Prozesse bereit, die in einem Pod ausgeführt werden. Weitere Informationen finden Sie unter Managing Service Accounts (Servicekonten verwalten) in der Kubernetes-Dokumentation. Wenn Sie Zugriff auf AWS Dienste Pod benötigen, können Sie das Dienstkonto einer AWS Identity and Access Management Identität zuordnen, um diesen Zugriff zu gewähren. Weitere Informationen finden Sie unter IAM-Rollen für Servicekonten.

Servicekonto-Tokens

Das `BoundServiceAccountTokenVolume`-Feature ist in Kubernetes-Versionen standardmäßig aktiviert. Dieses Feature verbessert die Sicherheit von Servicekonto-Tokens, indem sie es auf Kubernetes ausgeführten Workloads ermöglicht, JSON-Web-Tokens, die an Zielgruppen, Zeit und Schlüssel gebunden sind, anzufordern. Die Ablaufzeit für Servicekonto-Tokens beträgt eine Stunde. In früheren Kubernetes-Versionen hatten Token kein Ablaufdatum. Clients, die diese Tokens verwenden, müssen die Tokens nun also innerhalb einer Stunde aktualisieren. Die folgenden Kubernetes-Client-SKDs aktualisieren Tokens automatisch im erforderlichen Zeitrahmen:

- Go-Version `0.15.7` und höher
- Python-Version `12.0.0` und höher
- Java-Version `9.0.0` und höher
- JavaScript Version `0.10.3` und später
- Ruby `master`-Branch
- Haskell-Version `0.3.0.0`
- C#-Version `7.0.5` und höher

Wenn Ihre Workload eine frühere Client-Version verwendet, ist eine Aktualisierung erforderlich. Um Clients reibungslos zu neueren zeitlich begrenzten Servicekonto-Tokens zu migrieren, fügt Kubernetes eine verlängerte Ablaufzeit für Servicekonto-Tokens, die die standardmäßige Stunde übersteigt. Für Amazon-EKS-Cluster gilt eine verlängerte Ablaufzeit von 90 Tagen. Der Kubernetes-

API-Server Ihres Amazon-EKS-Clusters lehnt Anfragen mit Token ab, die älter als 90 Tage sind. Sie sollten Ihre Anwendungen und ihre Abhängigkeiten überprüfen, um sicherzustellen, dass die Kubernetes-Client-SDKs mit den zuvor aufgeführten Versionen identisch oder höher sind.

Wenn der API-Server Anfragen mit Token erhält, die älter als eine Stunde sind, kommentiert er das Audit-Protokollereignis der API mit `annotations.authentication.k8s.io/stale-token`. Die Anmerkung sieht zum Beispiel wie folgt aus:

```
subject: system:serviceaccount:common:fluent-bit, seconds after warning threshold:
 4185802.
```

Wenn für Ihren Cluster die Steuerebenen-Protokollierung aktiviert ist, dann befinden sich die Anmerkungen in den Überwachungsprotokollen. Sie können die folgende CloudWatch Logs Insights-Abfrage verwenden, um alle Pods in Ihrem Amazon EKS-Cluster zu identifizieren, die veraltete Token verwenden:

```
fields @timestamp
| filter @logStream like /kube-apiserver-audit/
| filter @message like /seconds after warning threshold/
| parse @message "subject: *, seconds after warning threshold:*\"" as subject,
 elapsedtime
```

`subject` bezieht sich auf das Servicekonto, das der Pod verwendet hat. `elapsedtime` gibt die verstrichene Zeit (in Sekunden) nach dem Lesen des neuesten Tokens an. Die Anfragen an den API-Server werden abgelehnt, wenn `elapsedtime` 90 Tage (7 776 000 Sekunden) überschreitet. Sie sollten die Kubernetes-Client-SDKs Ihrer Anwendungen proaktiv auf eine der zuvor aufgeführten Versionen aktualisieren, die das Token automatisch aktualisieren. Wenn das verwendete Servicekonto-Token fast 90 Tage erreicht hat, und Sie nicht genügend Zeit haben, Ihre Client-SDK-Versionen vor Ablauf des Tokens zu aktualisieren, können Sie die vorhandenen Pods beenden und neue erstellen. Dadurch wird das Servicekonto-Token erneut abgerufen, sodass Sie weitere 90 Tage Zeit haben, die SDKs Ihrer Client-Version zu aktualisieren.

Wenn der Pod Teil einer Bereitstellung ist, besteht die empfohlene Vorgehensweise zum Beenden der Pods und gleichzeitigen Aufrechterhaltung hoher Verfügbarkeit darin, mit dem folgenden Befehl ein Rollout durchzuführen. Ersetzen Sie *my-deployment* mit dem Namen Ihrer Bereitstellung.

```
kubectl rollout restart deployment/my-deployment
```

Cluster-Add-ons

Die folgenden Cluster-Add-ons wurden für die Verwendung der Kubernetes-Client-SDKs aktualisiert, die Servicekonto-Token automatisch neu abrufen können. Wir empfehlen Ihnen, sicherzustellen, dass die aufgeführten Versionen oder neuere Versionen auf Ihrem Cluster installiert sind.

- Amazon VPC CNI plugin for Kubernetes- und Metrikhelferobjekt-Plugins ab Version `1.8.0`. Informationen zur Überprüfung oder Aktualisierung Ihrer aktuellen Version finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on und cni-metrics-helper.
- CoreDNS-Version `1.8.4` und höher. Zum Überprüfen oder Aktualisieren Ihrer aktuellen Version siehe Arbeiten mit dem CoreDNS-Amazon-EKS-Add-on.
- AWS Load Balancer Controller-Version `2.0.0` und höher. Zum Überprüfen oder Aktualisieren Ihrer aktuellen Version siehe Was ist die AWS Load Balancer Controller?.
- Eine aktuelle `kube-proxy`-Version. Zum Überprüfen oder Aktualisieren Ihrer aktuellen Version siehe Arbeiten mit dem `kube-proxy` Kubernetes-Add-on.
- AWS für die Fluent Bit-Version `2.25.0` oder höher. Informationen zum Aktualisieren Ihrer aktuellen Version finden Sie unter Releases auf GitHub.
- Fluentd-Imageversion 1.14.6-1.2 oder höher und Fluentd-Filter-Plugin für Kubernetes-Metadatenversion 2.11.1 oder höher.

Erteilen von AWS Identity and Access Management Berechtigungen für Workloads auf Amazon Elastic Kubernetes Service Service-Clustern

Amazon EKS bietet zwei Möglichkeiten, AWS Identity and Access Management Berechtigungen für Workloads zu erteilen, die in Amazon EKS-Clustern ausgeführt werden: IAM-Rollen für Dienstkonten und EKS-Pod-Identitäten.

IAM-Rollen für Servicekonten

> IAM-Rollen für Dienstkonten (IRSA) konfiguriert Kubernetes-Anwendungen, auf denen sie ausgeführt werden, AWS mit detaillierten IAM-Berechtigungen für den Zugriff auf verschiedene andere AWS Ressourcen wie Amazon S3 S3-Buckets, Amazon DynamoDB-Tabellen und mehr. Sie können mehrere Anwendungen zusammen in demselben Amazon EKS-Cluster ausführen und sicherstellen, dass jede Anwendung nur über die erforderlichen Mindestberechtigungen verfügt. IRSA wurde entwickelt, um verschiedene Kubernetes Bereitstellungsoptionen zu unterstützen, die von AWS Amazon EKS, Amazon EKS Anywhere und selbstverwalteten Kubernetes Clustern auf

Amazon EC2 EC2-Instances unterstützt werden. Red Hat OpenShift Service in AWS Daher wurde IRSA mithilfe grundlegender AWS Dienste wie IAM erstellt und war nicht direkt vom Amazon EKS-Service und der EKS-API abhängig. Weitere Informationen finden Sie unter IAM-Rollen für Servicekonten.

EKS-Pod-Identitäten

EKS Pod Identity bietet Clusteradministratoren einen vereinfachten Arbeitsablauf für die Authentifizierung von Anwendungen für den Zugriff auf verschiedene andere AWS Ressourcen wie Amazon S3 S3-Buckets, Amazon DynamoDB-Tabellen und mehr. EKS Pod Identity ist nur für EKS vorgesehen und vereinfacht Cluster-Administratoren so die Konfiguration von Kubernetes-Anwendungen, um IAM-Berechtigungen zu erhalten. Diese Berechtigungen können jetzt einfach und mit weniger Schritten direkt über AWS Management Console die EKS-API und konfiguriert werden AWS CLI, und es müssen keine Aktionen innerhalb des Clusters für Objekte ergriffen werden. Kubernetes Cluster-Administratoren müssen nicht zwischen den EKS- und IAM-Services wechseln oder privilegierte IAM-Operationen verwenden, um die für Ihre Anwendungen erforderlichen Berechtigungen zu konfigurieren. IAM-Rollen können jetzt in mehreren Clustern verwendet werden, ohne dass bei der Erstellung neuer Cluster die Rollenvertrauensrichtlinie aktualisiert werden muss. Die von EKS Pod Identity bereitgestellten IAM-Anmeldeinformationen umfassen Rollensitzungs-Tags mit Attributen wie Cluster-Name, Namespace und dem Namen des Servicekontos. Mithilfe von Rollensitzungs-Tags können Administratoren eine einzelne Rolle erstellen, die für alle Dienstkonten verwendet werden kann, indem sie den Zugriff auf AWS Ressourcen auf der Grundlage übereinstimmender Tags ermöglicht. Weitere Informationen finden Sie unter EKS-Pod-Identitäten.

Vergleich von EKS Pod Identity und IRSA

Ganz allgemein ermöglichen es Ihnen sowohl EKS Pod Identity als auch IRSA, Anwendungen, die auf Kubernetes-Clustern ausgeführt werden, IAM-Berechtigungen zu gewähren. Sie unterscheiden sich jedoch grundlegend in ihrer Konfiguration, den unterstützten Grenzwerten und den aktivierten Features. Im Folgenden vergleichen wir einige der wichtigsten Aspekte der beiden Lösungen.

	EKS Pod Identity	IRSA
Erweiterbarkeit von Rollen	Sie müssen jede Rolle einmal einrichten, um das Vertrauen des neu eingeführten Amazon EKS-Service-Prinzipal	Sie müssen die Vertrauen srichtlinie der IAM-Rolle jedes Mal mit dem neuen Endpunkt des EKS-Cluster–OIDCAn

	EKS Pod Identity	IRSA
	zu erhalten `pods.eks.amazonaws.com` . Nach diesem einmaligen Schritt müssen Sie die Vertrauen srichtlinie der Rolle nicht jedes Mal aktualisieren, wenn sie in einem neuen Cluster verwendet wird.	bieters aktualisieren, wenn Sie die Rolle in einem neuen Cluster verwenden möchten.
Skalierbarkeit von Clustern	Bei EKS Pod Identity müssen Benutzer keinen IAM-OIDC-Anbieter einrichten, dieses Limit gilt also nicht.	Jedem EKS-Cluster ist eine OpenID Connect (OIDC)-Au ssteller-URL zugeordnet. Um IRSA verwenden zu können, muss für jeden EKS-Cluster in IAM ein eigener OpenID Connect-Anbieter erstellt werden. IAM hat ein globales Standardlimit von jeweils 100 OIDC-Anbietern pro AWS-Konto. Wenn Sie planen, mehr als 100 EKS-Cluster für jeden AWS-Konto mit IRSA zu haben, dann erreichen Sie das Limit für OIDC IAM-Anbieter.

	EKS Pod Identity	IRSA
Skalierbarkeit von Rollen	Bei EKS Pod Identity müssen Benutzer in der Vertrauen srichtlinie keine Vertrauen sbeziehung zwischen der IAM-Rolle und dem Servicekonto definieren, dieses Limit gilt also nicht.	In IRSA muss die Vertrauen sbeziehung zwischen einer IAM-Rolle und einem Servicekonto in der Vertrauen srichtlinie der Rolle definiert werden. Standardmäßig beträgt die Größe der Vertrauensrichtlinie 2048. Das bedeutet, dass Sie in der Regel vier Vertrauen sbeziehungen in einer Vertrauensrichtlinie definiere n können. Sie können zwar das Limit für die maximale Größe der Vertrauensrichtlin ie erhöhen, sind jedoch in der Regel auf maximal acht Vertrauensbeziehungen innerhalb einer Vertrauen srichtlinie beschränkt.

	EKS Pod Identity	IRSA
Wiederverwendbarkeit von Rollen	AWS STS Zu den von EKS Pod Identity bereitgestellten temporären Anmeldeinformationen gehören Rollensitzungs-Tags wie Clustername, Namespace und Dienstkontoname. Mithilfe von Rollensitzungs-Tags können Administratoren eine einzelne IAM-Rolle erstellen, die von mehreren Servicekonten mit unterschiedlichen effektiven Berechtigungen verwendet werden kann, indem auf Grundlage der ihnen zugewiesenen Tags Zugriff auf AWS -Ressourcen ermöglicht wird. Dies wird auch als attributbasierte Zugriffskontrolle (ABAC) bezeichnet. Weitere Informationen finden Sie unter Definieren von Berechtigungen für EKS-Pod-Identitäten, um Rollen basierend auf Tags anzunehmen.	AWS STS Sitzungs-Tags werden nicht unterstützt. Sie können eine Rolle in verschiedenen Clustern wiederverwenden, aber jeder Pod erhält alle Berechtigungen der Rolle.
Unterstützte Umgebungen	EKS Pod Identity ist nur in Amazon EKS verfügbar.	IRSA kann wie Amazon EKS, Amazon EKS Anywhere und selbstverwaltete Kubernetes Cluster auf Amazon EC2 EC2-Instances verwendet werden. Red Hat OpenShift Service in AWS

	EKS Pod Identity	IRSA
Unterstützte EKS-Versionen	EKS Kubernetes-Versionen `1.24` oder höher. Informationen über die jeweiligen Plattformversionen finden Sie unter Cluster-Versionen von EKS Pod Identity.	Alle unterstützten EKS-Cluster-Versionen.

EKS-Pod-Identitäten

Anwendungen in Containern Pod von A können ein AWS SDK oder das verwenden, AWS CLI um API-Anfragen an Nutzungs- AWS Identity and Access Management (IAM-) Berechtigungen zu AWS-Services stellen. Anwendungen müssen ihre AWS API-Anfragen mit AWS Anmeldeinformationen signieren.

EKS-Pod-Identitäten bieten die Möglichkeit, Anmeldeinformationen für Ihre Anwendungen zu verwalten, ähnlich wie Amazon-EC2-Instance-Profile Anmeldeinformationen für Amazon-EC2-Instances bereitstellen. Anstatt Ihre AWS Anmeldeinformationen zu erstellen und an die Container zu verteilen oder die Rolle der Amazon EC2 EC2-Instance zu verwenden, verknüpfen Sie eine IAM-Rolle mit einem Kubernetes Dienstkonto und konfigurieren Ihr Pods Konto für die Verwendung des Dienstkontos.

Jede EKS-Pod-Identity-Zuordnung ordnet einem Servicekonto in einem Namespace im angegebenen Cluster eine Rolle zu. Wenn Sie dieselbe Anwendung in mehreren Clustern haben, können Sie in jedem Cluster identische Zuordnungen vornehmen, ohne die Vertrauensrichtlinie der Rolle zu ändern.

Wenn ein Pod ein Servicekonto mit einer Zuordnung verwendet, legt Amazon EKS Umgebungsvariablen in den Containern des Pods fest. Die Umgebungsvariablen konfigurieren die AWS SDKs, einschließlich der, für die AWS CLI Verwendung der EKS Pod Identity-Anmeldeinformationen.

Vorteile von EKS-Pod-Identitäten

EKS-Pod-Identitäten bieten die folgenden Vorteile:

- Geringste Berechtigung – Sie können IAM-Berechtigungen auf ein Servicekonto beschränken, und nur Pods, die dieses Servicekonto verwenden, haben Zugriff auf diese Berechtigungen. Mit diesem Feature entfällt auch die Notwendigkeit von Drittanbieterlösungen wie `kiam` oder `kube2iam`.
- Isolation von Anmeldeinformationen – Ein Container eines Pod's kann nur Anmeldeinformationen für die IAM-Rolle abrufen, die dem Servicekonto zugeordnet ist, zu dem er gehört. Ein Container hat nie Zugriff auf Anmeldeinformationen, die von anderen Containern in anderen Pods verwendet werden. Wenn Pod-Identitäten verwendet werden, haben die Container des Pod's auch die Berechtigungen, die der IAM-Rolle des Amazon-EKS-Knotens zugeordnet sind, es sei denn, Sie blockieren den Pod-Zugang zum Amazon EC2 Instance Metadata Service (IMDS). Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.
- Überprüfbarkeit — Die Zugriffs- und Ereignisprotokollierung ist verfügbar AWS CloudTrail , um nachträgliche Prüfungen zu erleichtern.

EKS Pod Identity ist eine einfachere Methode als IAM-Rollen für Servicekonten, da diese keine OIDC-Identitätsanbieter nutzt. EKS Pod Identity bietet die folgenden Verbesserungen:

- Unabhängiger Betrieb – In vielen Organisationen liegt die Verantwortung für die Erstellung von OIDC-Identitätsanbietern bei anderen Teams als für die Verwaltung der Kubernetes-Cluster. EKS Pod Identity hat eine klare Aufgabentrennung: Die gesamte Konfiguration der EKS-Pod-Identity-Zuordnungen erfolgt in Amazon EKS und die gesamte Konfiguration der IAM-Berechtigungen erfolgt in IAM.
- Wiederverwendbarkeit – EKS Pod Identity verwendet einen einzigen IAM-Prinzipal anstelle der separaten Prinzipale für jeden Cluster, den IAM-Rollen für Servicekonten verwenden. Ihr IAM-Administrator fügt der Vertrauensrichtlinie jeder Rolle den folgenden Prinzipal hinzu, damit sie von EKS-Pod-Identitäten verwendet werden kann.

```
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            }
```

- Skalierbarkeit — Jeder Satz temporärer Anmeldeinformationen wird vom EKS Auth Dienst in EKS Pod Identity angenommen und nicht von jedem AWS SDK, das Sie in jedem Pod ausführen. Anschließend stellt der Amazon EKS Pod Identity Agent, der auf jedem Knoten ausgeführt wird, die Anmeldeinformationen an die SDKs aus. Dadurch wird die Last für jeden Knoten reduziert und nicht in jedem Pod dupliziert. Weitere Details zu diesem Prozess finden Sie unter Funktionsweise von EKS Pod Identity.

Weitere Informationen zum Vergleich der beiden Alternativen finden Sie unter Gewähren Sie Kubernetes-Workloads Zugriff auf die Verwendung von Service AWS Accounts Kubernetes.

Übersicht über die Einrichtung von EKS-Pod-Identitäten

Aktivieren Sie EKS-Pod-Identitäten, indem Sie die folgenden Verfahren ausführen:

1. Richten Sie den Amazon EKS Pod Identity Agent ein – Sie führen dieses Verfahren für jeden Cluster nur einmal durch.
2. Konfigurieren Sie ein Kubernetes Dienstkonto, um eine IAM-Rolle mit EKS Pod Identity anzunehmen – Führen Sie dieses Verfahren für jeden eindeutigen Satz von Berechtigungen aus, über den eine Anwendung verfügen soll.
3. Für Pods die Verwendung eines Kubernetes Dienstkontos konfigurieren— Führen Sie dieses Verfahren für jeden ausPod, auf den Sie Zugriff benötigen AWS-Services.
4. Verwenden Sie ein unterstütztes AWS SDK— Vergewissern Sie sich, dass der Workload ein AWS SDK einer unterstützten Version verwendet und dass der Workload die standardmäßige Anmeldeinformationskette verwendet.

Überlegungen zu EKS Pod Identity

- Sie können jedem Kubernetes-Servicekonto in jedem Cluster eine IAM-Rolle zuordnen. Sie können ändern, welche Rolle dem Servicekonto zugeordnet ist, indem Sie die EKS-Pod-Identity-Zuordnung bearbeiten.
- Sie können nur Rollen zuordnen, die sich im selben AWS-Konto Cluster befinden. Sie können den Zugriff von einem anderen Konto auf die Rolle in diesem Konto delegieren, die Sie für die Verwendung durch EKS-Pod-Identitäten konfigurieren. Eine Anleitung zum Delegieren von Zugriff finden Sie unter AWS Kontenübergreifendes Delegieren des Zugriffs mithilfe von IAM-Rollen im IAM-Benutzerhandbuch. `AssumeRole`
- Der EKS Pod Identity Agent ist erforderlich. Er wird als Kubernetes `DaemonSet` auf Ihren Knoten ausgeführt und stellt Anmeldeinformationen nur für Pods auf dem Knoten bereit, auf dem er ausgeführt wird. Weitere Informationen zur Kompatibilität des EKS Pod Identity Agent finden Sie im folgenden Abschnitt (Einschränkungen von EKS Pod Identity).
- Der EKS Pod Identity Agent verwendet das `hostNetwork` des Knotens sowie Port `80` und Port `2703` in einer lokalen Adresse im Knoten. Diese Adresse ist `169.254.170.23` für IPv4- und `[fd00:ec2::23]` für IPv6-Cluster.

Wenn Sie IPv6 Adressen deaktivieren oder Localhost-IP-Adressen anderweitig verhindern, kann der IPv6 Agent nicht gestartet werden. Um den Agenten auf Knoten zu starten, die nicht verwendet werden könnenIPv6, folgen Sie den Schritten unter IPv6Im EKS Pod Identity Agent deaktivieren So deaktivieren Sie die IPv6 Konfiguration.

Cluster-Versionen von EKS Pod Identity

Um EKS-Pod-Identitäten verwenden zu können, muss der Cluster eine Plattformversion haben, die mindestens der Version in der folgenden Tabelle entspricht, oder eine Kubernetes-Version, die neuer als die in der Tabelle aufgeführten Versionen ist.

Kubernetes-Version	Plattformversion
1.30	eks.2
1.29	eks.1
1.28	eks.4
1.27	eks.8
1.26	eks.9
1.25	eks.10
1.24	eks.13

Mit EKS Pod Identity kompatible Zusatzversionen

Important

Um EKS Pod Identity mit einem EKS Add-on zu verwenden, müssen Sie die EKS Pod Identity-Zuordnung manuell erstellen. Wählen Sie in der Add-On-Konfiguration keine IAM-Rolle aus AWS Management Console, diese Rolle wird nur mit IRSA verwendet.

Amazon EKS-Add-Ons und selbstverwaltete Add-Ons, die IAM-Anmeldeinformationen benötigen, können EKS Pod Identity, IRSA oder die Instance-Rolle verwenden. Die Liste der Add-Ons, die IAM-Anmeldeinformationen verwenden und EKS Pod Identity unterstützen, lautet wie folgt:

- Amazon VPC CNI plugin for Kubernetes`1.15.5-eksbuild.1`oder später
- AWS Load Balancer Controller`2.7.0`oder später. Beachten Sie, dass das AWS Load Balancer Controller nicht als EKS-Add-on verfügbar ist, sondern als selbstverwaltetes Add-On.

Einschränkungen von EKS Pod Identity

EKS-Pod-Identitäten sind auf folgenden Plattformen verfügbar:

- Amazon-EKS-Cluster-Versionen, die im vorherigen Thema (Cluster-Versionen von EKS Pod Identity) aufgeführt wurden.
- Worker-Knoten im Cluster, die Linux-Amazon-EC2-Instances sind.

EKS-Pod-Identitäten sind auf folgenden Plattformen nicht verfügbar:

- China-Regionen.
- AWS GovCloud (US).
- AWS Outposts.
- Amazon EKS Anywhere.
- Kubernetes-Cluster, die Sie in Amazon EC2 erstellen und ausführen. Die EKS-Pod-Identity-Komponenten sind nur in Amazon EKS verfügbar.

Sie können EKS-Pod-Identitäten nicht verwenden mit:

- Pods, die nicht auf Linux-Amazon-EC2-Instances ausgeführt werden. Linux- und Windows-Pods, die auf laufen, AWS Fargate (Fargate) werden nicht unterstützt. Auf Windows-Amazon-EC2-Instances ausgeführte Pods werden nicht unterstützt.
- Amazon-EKS-Add-ons, die IAM-Anmeldeinformationen benötigen. Die EKS-Add-ons können nur IAM-Rollen für Servicekonten verwenden. Die Liste der EKS-Add-ons, die IAM-Anmeldeinformationen verwenden, umfasst:
  - Die CSI Speichertreiber: EBS CSI, EFS CSI, Amazon FSx for Lustre CSI-Treiber, Amazon FSx für NetApp ONTAP CSI-Treiber, Amazon FSx für OpenZFS CSI-Treiber, Amazon File Cache

CSI-Treiber, AWS Secrets and Configuration Provider (ASCP) für den Secrets Store CSI-Treiber Kubernetes

> **Note**
>
> Wenn diese Controller, Treiber und Plug-ins als selbstverwaltete Add-Ons statt als EKS-Add-Ons installiert werden, unterstützen sie EKS Pod Identities, sofern sie aktualisiert wurden, um die neuesten AWS SDKs zu verwenden.

Funktionsweise von EKS Pod Identity

Amazon-EKS-Pod-Identity-Zuordnungen bieten die Möglichkeit, Anmeldeinformationen für Ihre Anwendungen zu verwalten, ähnlich wie Amazon-EC2-Instance-Profile Anmeldeinformationen für Amazon-EC2-Instances bereitstellen.

Amazon EKS Pod Identity bietet Anmeldeinformationen für Ihre Workloads mit einer zusätzlichen EKS-Auth-API und einem Agent-Pod, der auf jedem Knoten ausgeführt wird.

In Ihren Add-Ons, wie Amazon EKS-Add-Ons und selbstverwalteten Controllern, Operatoren und anderen Add-Ons, muss der Autor seine Software aktualisieren, um die neuesten AWS SDKs verwenden zu können. Die Liste der Kompatibilität zwischen EKS Pod Identity und den von Amazon EKS produzierten Add-ons finden Sie im vorherigen Abschnitt (Einschränkungen von EKS Pod Identity).

Verwenden von EKS-Pod-Identitäten in Ihrem Code

In Ihrem Code können Sie die AWS SDKs verwenden, um auf Dienste zuzugreifen. AWS Sie schreiben Code, um einen Client für einen AWS Dienst mit einem SDK zu erstellen. Standardmäßig sucht das SDK in einer Kette von Speicherorten nach AWS Identity and Access Management Anmeldeinformationen, die verwendet werden können. Nachdem gültige Anmeldeinformationen gefunden wurden, wird die Suche beendet. Weitere Informationen zu den verwendeten Standardspeicherorten finden Sie unter Credential Provider Chain im Referenzhandbuch für AWS SDKs und Tools.

EKS-Pod-Identitäten wurden dem Anbieter für Container-Anmeldeinformationen hinzugefügt, der in einem Schritt in der standardmäßigen Anmeldeinformationskette durchsucht wird. Wenn Ihre Workloads derzeit Anmeldeinformationen verwenden, die in der Kette der Anmeldeinformationen weiter vorn stehen, werden diese Anmeldeinformationen auch dann weiter verwendet, wenn Sie eine

EKS-Pod-Identity-Zuordnung für dieselbe Workload konfigurieren. Auf diese Weise können Sie sicher von anderen Anmeldeinformationstypen migrieren, indem Sie die Zuordnung erstellen, bevor Sie die alten Anmeldeinformationen entfernen.

Der Anbieter für Container-Anmeldeinformationen stellt temporäre Anmeldeinformationen über einen Agent bereit, der auf jedem Knoten ausgeführt wird. In Amazon EKS ist dies der Amazon EKS Pod Identity Agent und bei Amazon Elastic Container Service ist es der `amazon-ecs-agent`. Die SDKs verwenden Umgebungsvariablen, um den Agent zu finden, zu dem eine Verbindung hergestellt werden soll.

Im Gegensatz dazu stellen IAM-Rollen für Dienstkonten ein Web-Identitätstoken bereit, mit AWS Security Token Service dem das AWS SDK mithilfe von `AssumeRoleWithWebIdentity`

Funktionsweise von EKS Pod Identity Agent mit einem Pod

1. Wenn Amazon EKS einen neuen Pod startet, der ein Servicekonto mit einer EKS-Pod-Identity-Zuordnung verwendet, fügt der Cluster dem Pod-Manifest den folgenden Inhalt hinzu:

```
    env:
    - name: AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE
      value: "/var/run/secrets/pods.eks.amazonaws.com/serviceaccount/eks-pod-
identity-token"
    - name: AWS_CONTAINER_CREDENTIALS_FULL_URI
      value: "http://169.254.170.23/v1/credentials"
    volumeMounts:
    - mountPath: "/var/run/secrets/pods.eks.amazonaws.com/serviceaccount/"
      name: eks-pod-identity-token
  volumes:
  - name: eks-pod-identity-token
    projected:
      defaultMode: 420
      sources:
      - serviceAccountToken:
          audience: pods.eks.amazonaws.com
          expirationSeconds: 86400 # 24 hours
          path: eks-pod-identity-token
```

2. Kubernetes wählt aus, auf welchem Knoten der Pod ausgeführt werden soll. Anschließend verwendet der Amazon EKS Pod Identity Agent auf dem Knoten die AssumeRoleForPodIdentityAktion, um temporäre Anmeldeinformationen von der EKS Auth API abzurufen.

3. Der EKS Pod Identity Agent stellt diese Anmeldeinformationen für die AWS SDKs zur Verfügung, die Sie in Ihren Containern ausführen.
4. Sie nutzen das SDK in Ihrer Anwendung, ohne einen Anmeldeinformationsanbieter für die Verwendung der standardmäßigen Anmeldeinformationskette anzugeben. Oder Sie geben den Anbieter für Container-Anmeldeinformationen an. Weitere Informationen zu den verwendeten Standardspeicherorten finden Sie unter Credential Provider Chain im Referenzhandbuch für AWS SDKs und Tools.
5. Das SDK verwendet die Umgebungsvariablen, um eine Verbindung zum EKS Pod Identity Agent herzustellen und die Anmeldeinformationen abzurufen.

> **Note**
>
> Wenn Ihre Workloads derzeit Anmeldeinformationen verwenden, die in der Kette der Anmeldeinformationen weiter vorn stehen, werden diese Anmeldeinformationen auch dann weiter verwendet, wenn Sie eine EKS-Pod-Identity-Zuordnung für dieselbe Workload konfigurieren.

Richten Sie den Amazon EKS Pod Identity Agent ein

Amazon-EKS-Pod-Identity-Zuordnungen bieten die Möglichkeit, Anmeldeinformationen für Ihre Anwendungen zu verwalten, ähnlich wie Amazon-EC2-Instance-Profile Anmeldeinformationen für Amazon-EC2-Instances bereitstellen.

Amazon EKS Pod Identity bietet Anmeldeinformationen für Ihre Workloads mit einer zusätzlichen EKS-Auth-API und einem Agent-Pod, der auf jedem Knoten ausgeführt wird.

Überlegungen

- **`IPv6`**

  Standardmäßig überwacht der EKS Pod Identity Agent eine `IPv4` `IPv6` AND-Adresse für Pods, um Anmeldeinformationen anzufordern. Der Agent verwendet die Loopback-IP-Adresse (localhost) `169.254.170.23` für `IPv4` und die Localhost-IP-Adresse für. `[fd00:ec2::23]` `IPv6`

  Wenn Sie `IPv6` Adressen deaktivieren oder auf andere Weise `IPv6` Localhost-IP-Adressen verhindern, kann der Agent nicht gestartet werden. Um den Agenten auf Knoten zu starten, die nicht verwendet werden können`IPv6`, folgen Sie den Schritten unter `IPv6`Im EKS Pod Identity Agent deaktivieren So deaktivieren Sie die `IPv6` Konfiguration.

Erstellen des Amazon EKS Pod Identity Agent

Voraussetzungen für den Agent

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS. Die Cluster- und Plattformversion müssen den unter Cluster-Versionen von EKS Pod Identity aufgeführten Versionen entsprechen oder neuer sein.
- Die Knotenrolle verfügt über Berechtigungen für den Agent, um die `AssumeRoleForPodIdentity`-Aktion in der EKS-Auth-API auszuführen. Sie können AWS verwaltete Richtlinie: AmazonEKS WorkerNodePolicy verwenden oder eine benutzerdefinierten Richtlinie ähnlich dem Folgenden hinzufügen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "eks-auth:AssumeRoleForPodIdentity"
            ],
            "Resource": "*"
        }
    ]
}
```

  Diese Aktion kann durch Tags beschränkt werden, um einzuschränken, welche Rollen von Pods übernommen werden können, die den Agent verwenden.
- Die Knoten können auf Amazon ECR zugreifen und Images von Amazon ECR herunterladen. Das Container-Image für das Add-on befindet sich in den Registrierungen, die unter Registrierungen für Amazon-Container-Images aufgeführt sind.

  Beachten Sie, dass Sie den Speicherort des Images ändern und EKS-Add-Ons in den AWS Management Console optionalen Konfigurationseinstellungen in und `--configuration-values` in der angeben `imagePullSecrets` können AWS CLI.
- Die Knoten können die Amazon-EKS-Auth-API erreichen. Für private Cluster ist der `eks-auth` Endpunkt-In AWS PrivateLink erforderlich.

AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich Cluster aus. Wählen Sie anschließend den Namen des Clusters aus, für den Sie das Add-on „EKS Pod Identity Agent“ konfigurieren möchten.
3. Wählen Sie die Registerkarte Add-ons.
4. Wählen Sie Weitere Add-Ons erhalten.
5. Wählen Sie das Kästchen oben rechts im Add-on-Bereich für EKS Pod Identity Agent aus und gehen Sie dann auf Bearbeiten.
6. Wählen Sie auf der Seite Konfigurieren ausgewählter Add-on-Einstellungen eine beliebige Version in der Dropdown-Liste Version aus.
7. (Optional) Erweitern Sie Optionale Konfigurationseinstellungen, um eine zusätzliche Konfiguration einzugeben. Sie können beispielsweise einen alternativen Speicherort für das Container-Image und `ImagePullSecrets` angeben. Das JSON Schema mit den akzeptierten Schlüsseln wird im Add-on-Konfigurationsschema angezeigt.

   Geben Sie die Konfigurationsschlüssel und Werte in das Feld Konfigurationswerte ein
8. Wählen Sie Weiter aus.
9. Vergewissern Sie sich, dass die EKS-Pod-Identity-Pods auf Ihrem Cluster ausgeführt werden.

   ```
   kubectl get pods -n kube-system | grep 'eks-pod-identity-agent'
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   eks-pod-identity-agent-gmqp7                                          1/1
    Running   1 (24h ago)   24h
   eks-pod-identity-agent-prnsh                                          1/1
    Running   1 (24h ago)   24h
   ```

   Sie können jetzt EKS-Pod-Identity-Zuordnungen in Ihrem Cluster verwenden. Weitere Informationen finden Sie unter Konfigurieren Sie ein Kubernetes Dienstkonto, um eine IAM-Rolle mit EKS Pod Identity anzunehmen.

AWS CLI

1. Führen Sie den folgenden AWS CLI Befehl aus. Ersetzen Sie `my-cluster` mit dem Namen Ihres Clusters.

```
aws eks create-addon --cluster-name my-cluster --addon-name eks-pod-identity-agent --addon-version v1.0.0-eksbuild.1
```

> **Note**
>
> Der EKS Pod Identity Agent verwendet den `service-account-role-arn` nicht für IAM-Rollen für Servicekonten. Sie müssen dem EKS Pod Identity Agent Berechtigungen für die Knotenrolle gewähren.

2. Vergewissern Sie sich, dass die EKS-Pod-Identity-Pods auf Ihrem Cluster ausgeführt werden.

```
kubectl get pods -n kube-system | grep 'eks-pod-identity-agent'
```

Eine Beispielausgabe sieht wie folgt aus.

```
eks-pod-identity-agent-gmqp7                                          1/1
 Running   1 (24h ago)   24h
eks-pod-identity-agent-prnsh                                          1/1
 Running   1 (24h ago)   24h
```

Sie können jetzt EKS-Pod-Identity-Zuordnungen in Ihrem Cluster verwenden. Weitere Informationen finden Sie unter Konfigurieren Sie ein Kubernetes Dienstkonto, um eine IAM-Rolle mit EKS Pod Identity anzunehmen.

Aktualisieren des Amazon EKS Pod Identity Agent

Erstellen Sie das Add-on vom Typ Amazon EKS. Wenn Sie den Amazon-EKS-Typ des Add-ons nicht zu Ihrem Cluster hinzugefügt haben, siehe Erstellen des Amazon EKS Pod Identity Agent.

AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

2. Wählen Sie im linken Navigationsbereich Cluster aus. Wählen Sie anschließend den Namen des Clusters aus, für den Sie das Add-on „EKS Pod Identity Agent“ konfigurieren möchten.
3. Wählen Sie die Registerkarte Add-ons.
4. Wenn eine neue Version des Add-ons verfügbar ist, verfügt der EKS Pod Identity Agent über die Schaltfläche Version aktualisieren. Wählen Sie Version aktualisieren aus.
5. Wählen Sie auf der Seite Amazon EKS Pod Identity Agent konfigurieren die neue Version in der Dropdown-Liste Version aus.
6. Wählen Sie Änderungen speichern aus.

   Es kann einige Sekunden dauern, bis das Update abgeschlossen ist. Stellen Sie anschließend sicher, dass die Add-on-Version aktualisiert wurde, indem Sie den Status überprüfen.

AWS CLI

1. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist. Ersetzen Sie *my-cluster* mit Ihrem Clusternamen.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name eks-pod-identity-agent --query "addon.addonVersion" --output text
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.0.0-eksbuild.1
   ```

   Sie müssen das Add-on erstellen, bevor Sie es mit diesem Verfahren aktualisieren können.
2. Aktualisieren Sie Ihr Add-on mit der AWS CLI. Informationen dazu, wie Sie mit AWS Management Console oder `eksctl` das Add-on aktualisieren, finden Sie unter Aktualisieren eines Add-Ons. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie bei Bedarf die folgenden Änderungen am Befehl vor, und führen Sie dann den geänderten Befehl aus.

   - Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.
   - Ersetzen Sie *v1.0.0-eksbuild.1* durch die gewünschte Version.
   - Ersetzen Sie *111122223333* durch Ihre Konto-ID.
   - Führen Sie den folgenden Befehl aus:

```
aws eks update-addon --cluster-name my-cluster --addon-name eks-pod-identity-
agent --addon-version v1.0.0-eksbuild.1'
```

Es kann einige Sekunden dauern, bis das Update abgeschlossen ist.

3. Vergewissern Sie sich, dass die Add-on-Version aktualisiert wurde. Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.

```
aws eks describe-addon --cluster-name my-cluster --addon-name eks-pod-identity-
agent
```

Es kann einige Sekunden dauern, bis das Update abgeschlossen ist.

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "addon": {
        "addonName": "eks-pod-identity-agent",
        "clusterName": "my-cluster",
        "status": "ACTIVE",
        "addonVersion": "v1.0.0-eksbuild.1",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:region:111122223333:addon/my-cluster/eks-pod-
identity-agent/74c33d2f-b4dc-8718-56e7-9fdfa65d14a9",
        "createdAt": "2023-04-12T18:25:19.319000+00:00",
        "modifiedAt": "2023-04-12T18:40:28.683000+00:00",
        "tags": {}
    }
}
```

Konfiguration des EKS Pod Identity Agent

IPv6 Im EKS Pod Identity Agent deaktivieren

AWS Management Console

Deaktivieren Sie **IPv6** im AWS Management Console

1. Fügen Sie zur Deaktivierung `IPv6` im EKS Pod Identity Agent die folgende Konfiguration zu den optionalen Konfigurationseinstellungen des EKS Add-ons hinzu.

   a. Öffnen Sie die Amazon-EKS-Konsole unterhttps://console.aws.amazon.com/eks/home#/clusters.

   b. Wählen Sie im linken Navigationsbereich Clusters (Cluster) aus. Wählen Sie anschließend den Namen des Clusters aus, für den Sie das Add-On konfigurieren möchten.

   c. Wählen Sie die Registerkarte Add-ons.

   d. Wählen Sie das Feld oben rechts im EKS Pod Identity Agent-Add-On-Feld aus und wählen Sie dann Bearbeiten.

   e. Gehen Sie auf der Seite EKS Pod Identity Agent konfigurieren wie folgt vor:

      i. Wählen Sie die Version aus, die Sie verwenden möchten. Wir empfehlen, dieselbe Version wie im vorherigen Schritt beizubehalten und die Version und Konfiguration in separaten Aktionen zu aktualisieren.

      ii. Erweitern Sie Optionale Konfigurationseinstellungen.

      iii. Geben Sie den JSON-Schlüssel `"agent":` und den JSON-Wert eines verschachtelten JSON-Objekts mit einem Schlüssel `"additionalArgs":` im Feld Konfigurationswerte ein. Der resultierende Text muss ein gültiges JSON-Objekt sein. Wenn dieser Schlüssel und dieser Wert die einzigen Daten im Textfeld sind, setzen Sie den Schlüssel und den Wert in geschweifte Klammern `{}`. Das folgende Beispiel zeigt, dass die Netzwerkrichtlinie aktiviert ist:

```
{
    "agent": {
        "additionalArgs": {
            "-b": "169.254.170.23"
        }
    }
```

```
}
```

Diese Konfiguration legt fest, dass die `IPv4` Adresse die einzige Adresse ist, die vom Agenten verwendet wird.

f. Um die neue Konfiguration anzuwenden, indem die EKS Pod Identity Agent-Pods ersetzt werden, wählen Sie Änderungen speichern.

Amazon EKS wendet Änderungen an den EKS-Add-Ons mithilfe eines Rollouts des Kubernetes `DaemonSet` for EKS Pod Identity Agent an. Sie können den Status des Rollouts im Update-Verlauf des Add-ons in AWS Management Console und mit verfolgen. `kubectl rollout status daemonset/eks-pod-identity-agent --namespace kube-system`

`kubectl rollout`hat die folgenden Befehle:

```
$ kubectl rollout

history  -- View rollout history
pause    -- Mark the provided resource as paused
restart  -- Restart a resource
resume   -- Resume a paused resource
status   -- Show the status of the rollout
undo     -- Undo a previous rollout
```

Wenn der Rollout zu lange dauert, macht Amazon EKS den Rollout rückgängig und eine Meldung mit dem Typ Addon-Update und dem Status Fehlgeschlagen wird zum Update-Verlauf des Add-ons hinzugefügt. Um Probleme zu untersuchen, beginnen Sie mit dem Verlauf des Rollouts und führen Sie ihn `kubectl logs` auf einem EKS Pod Identity Agent-Pod aus, um die Protokolle von EKS Pod Identity Agent einzusehen.

2. Wenn der neue Eintrag im Update-Verlauf den Status Erfolgreich hat, ist der Rollout abgeschlossen und das Add-on verwendet die neue Konfiguration in allen EKS Pod Identity Agent-Pods.

AWS CLI

Deaktivieren Sie **`IPv6`** im AWS CLI

- Fügen Sie zur Deaktivierung `IPv6` im EKS Pod Identity Agent die folgende Konfiguration zu den Konfigurationswerten des EKS Add-ons hinzu.

  Führen Sie den folgenden AWS CLI Befehl aus. Ersetzen Sie `my-cluster` durch den Namen Ihres Clusters und den IAM-Rollen-ARN durch die Rolle, die Sie verwenden.

  ```
  aws eks update-addon --cluster-name my-cluster --addon-name eks-pod-identity-
  agent \
      --resolve-conflicts PRESERVE --configuration-values '{"agent":
  {"additionalArgs": { "-b": "169.254.170.23"}}}'
  ```

  Diese Konfiguration legt fest, dass die `IPv4` Adresse die einzige Adresse ist, die vom Agenten verwendet wird.

  Amazon EKS wendet Änderungen an den EKS-Add-Ons mithilfe eines Rollouts des Kubernetes DaemonSet for EKS Pod Identity Agent an. Sie können den Status des Rollouts im Update-Verlauf des Add-ons in AWS Management Console und mit verfolgen. `kubectl rollout status daemonset/eks-pod-identity-agent --namespace kube-system`

  `kubectl rollout`hat die folgenden Befehle:

  ```
  kubectl rollout

  history  -- View rollout history
  pause    -- Mark the provided resource as paused
  restart  -- Restart a resource
  resume   -- Resume a paused resource
  status   -- Show the status of the rollout
  undo     -- Undo a previous rollout
  ```

  Wenn der Rollout zu lange dauert, macht Amazon EKS den Rollout rückgängig und eine Meldung mit dem Typ Addon-Update und dem Status Fehlgeschlagen wird zum Update-Verlauf des Add-ons hinzugefügt. Um Probleme zu untersuchen, beginnen Sie mit dem

Verlauf des Rollouts und führen Sie ihn `kubectl logs` auf einem EKS Pod Identity Agent-Pod aus, um die Protokolle von EKS Pod Identity Agent einzusehen.

Konfigurieren Sie ein Kubernetes Dienstkonto, um eine IAM-Rolle mit EKS Pod Identity anzunehmen

In diesem Thema wird beschrieben, wie Sie ein Kubernetes Dienstkonto so konfigurieren, dass es eine AWS Identity and Access Management (IAM-) Rolle mit EKS Pod Identity annimmt. JederPods, der für die Verwendung des Dienstkontos konfiguriert ist, kann dann auf jedes Konto zugreifen, für AWS-Service das die Rolle über Zugriffsberechtigungen verfügt.

Um eine EKS Pod Identity-Zuordnung zu erstellen, gibt es nur einen einzigen Schritt: Sie erstellen die Zuordnung in EKS mithilfe der AWS Management Console, AWS CLI, AWS SDKs AWS CloudFormation und anderer Tools. Es gibt keine Daten oder Metadaten zu den Zuordnungen innerhalb des Clusters in Kubernetes-Objekten und Sie fügen den Servicekonten keine Anmerkungen hinzu.

Voraussetzungen

- Einen vorhandenen -Cluster. Wenn Sie keine haben, können Sie eine mit einem der Erste Schritte mit Amazon EKS-Leitfäden erstellen.
- Der IAM-Prinzipal, der die Zuordnung erstellt, muss `iam:PassRole` haben.
- Die neueste Version von, die auf Ihrem Gerät AWS CLI installiert und konfiguriert ist, oder AWS CloudShell. Sie können Ihre aktuelle Version mit `aws --version | cut -d / -f2 | cut -d ' ' -f1` überprüfen. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit `aws configure` im AWS Command Line Interface Benutzerhandbuch. Die in der installierte AWS CLI Version AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.

- Eine vorhandene `kubectl config`-Datei, die Ihre Clusterkonfiguration enthält. Informationen zum Erstellen einer `kubectl config`-Datei finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

Erstellen der EKS-Pod-Identity-Zuordnung

AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich Cluster aus. Wählen Sie anschließend den Namen des Clusters aus, für den Sie das Add-on „EKS Pod Identity Agent“ konfigurieren möchten.
3. Wählen Sie die Registerkarte Zugriff aus.
4. Wählen Sie in den Pod-Identity-Zuordnungen die Option Erstellen aus.
5. Wählen Sie als IAM-Rolle die IAM-Rolle mit den Berechtigungen aus, die die Workload haben soll.

> **Note**
>
> Die Liste enthält nur Rollen mit der folgenden Vertrauensrichtlinie, die EKS Pod Identity ermöglicht, diese Rollen zu verwenden.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEksAuthToAssumeRoleForPodIdentity",
            "Effect": "Allow",
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
    ]
```

```
}
```

`sts:AssumeRole`

EKS Pod Identity nutzt `AssumeRole`, um die IAM-Rolle anzunehmen, bevor es die temporären Anmeldeinformationen an Ihre Pods übergibt.

`sts:TagSession`

EKS Pod Identity nutzt `TagSession`, um Sitzungs-Tags in die Anfragen an AWS STS einzufügen.

Sie können diese Tags im Element condition keys in der Vertrauensrichtlinie verwenden, um zu beschränken, welche Servicekonten, Namespaces und Cluster diese Rolle verwenden dürfen.

Eine Liste der Amazon EKS-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Elastic Kubernetes Service in der Service-Autorisierungsreferenz. Um zu erfahren, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, lesen Sie von Amazon Elastic Kubernetes Service definierte Aktionen.

6. Wählen Sie als Kubernetes-Namespace den Kubernetes-Namespace aus, der das Servicekonto und die Workload enthält. Optional können Sie den Namen eines Namespace angeben, der nicht im Cluster vorhanden ist.
7. Wählen Sie als Kubernetes-Servicekonto das zu verwendende Kubernetes-Servicekonto aus. Im Manifest für Ihre Kubernetes-Workload muss dieses Servicekonto angegeben werden. Optional können Sie den Namen eines Servicekontos angeben, das nicht im Cluster vorhanden ist.
8. (Optional) Wählen Sie für die Tags die Option Tag hinzufügen aus, um Metadaten in einem Schlüssel-Wert-Paar hinzuzufügen. Diese Tags werden auf die Zuordnung angewendet und können in IAM-Richtlinien verwendet werden.

   Sie können diesen Schritt wiederholen, um mehrere Regionen hinzuzufügen.
9. Wählen Sie Erstellen.

AWS CLI

1. Wenn Sie Ihrer IAM-Rolle eine vorhandene IAM-Richtlinie zuordnen möchten, fahren Sie mit dem nächsten Schritt fort.

Erstellen Sie eine IAM-Richtlinie. Sie können eine eigene Richtlinie erstellen oder eine von AWS verwaltete Richtlinie kopieren, die bereits einige der benötigten Berechtigungen erteilt, und sie an Ihre spezifischen Anforderungen anpassen. Weitere Informationen finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

a. Erstellen Sie eine Datei, die die Berechtigungen für die AWS-Services enthält, auf die Ihre Pods Zugriff haben sollen. Eine Liste aller Aktionen für alle AWS-Services finden Sie in der Service Authorization Reference.

   Sie können den folgenden Befehl ausführen, um eine Beispiel-Richtliniendatei zu erstellen, die schreibgeschützten Zugriff auf einen Amazon-S3-Bucket gewährt. Sie können optional Konfigurationsinformationen oder ein Bootstrap-Skript in diesem Bucket speichern und die Container in Ihrem Pod können die Datei aus dem Bucket lesen und in Ihre Anwendung laden. Wenn Sie diese Beispielrichtlinie erstellen möchten, kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie *my-pod-secrets-bucket* durch Ihren Bucket-Namen und führen Sie den Befehl aus.

```
cat >my-policy.json <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-pod-secrets-bucket"
        }
    ]
}
EOF
```

b. Erstellen Sie die IAM-Richtlinie.

```
aws iam create-policy --policy-name my-policy --policy-document file://my-policy.json
```

2. Erstellen Sie eine IAM-Rolle und verknüpfen Sie sie mit einem Kubernetes-Servicekonto.

   1. Wenn Sie ein bestehendes Kubernetes-Servicekonto haben, das eine IAM-Rolle annehmen soll, können Sie diesen Schritt überspringen.

Ein Kubernetes-Service-Konto aktualisieren Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie *my-service-account* durch Ihren Wunschnamen und *default* durch einen anderen Namespace, falls erforderlich. Wenn Sie *default* ändern, muss der Namespace bereits vorhanden sein.

```
cat >my-service-account.yaml <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-service-account
  namespace: default
EOF
kubectl apply -f my-service-account.yaml
```

Führen Sie den folgenden Befehl aus.

```
kubectl apply -f my-service-account.yaml
```

2. Führen Sie den folgenden Befehl aus, um eine Vertrauensrichtlinie für die IAM-Rolle zu erstellen.

```
cat >trust-relationship.json <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEksAuthToAssumeRoleForPodIdentity",
            "Effect": "Allow",
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
    ]
}
EOF
```

3. Erstellen Sie die -Rolle. Ersetzen Sie *my-role* durch einen Namen für Ihre IAM-Rolle und *my-role-description* durch eine Beschreibung für Ihre Rolle.

```
aws iam create-role --role-name my-role --assume-role-policy-document
 file://trust-relationship.json --description "my-role-description"
```

4. Hängen Sie eine IAM-Richtlinie an Ihre Rolle an. Ersetzen Sie *my-role* durch den Namen Ihrer IAM-Rolle und *my-policy* durch den Namen einer vorhandenen Richtlinie, die Sie erstellt haben.

```
aws iam attach-role-policy --role-name my-role --policy-
arn=arn:aws:iam::111122223333:policy/my-policy
```

> **Note**
>
> Im Gegensatz zu IAM-Rollen für Servicekonten verwendet EKS Pod Identity keine Anmerkung im Servicekonto.

5. Führen Sie den folgenden Befehl aus, um die Zuordnung zu erstellen. Ersetzen Sie `my-cluster` durch den Namen des Clusters, *my-service-account* durch Ihren Wunschnamen und *default* durch einen anderen Namespace, falls erforderlich.

```
aws eks create-pod-identity-association --cluster-name my-cluster --role-
arn arn:aws:iam::111122223333:role/my-role --namespace default --service-
account my-service-account
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "association": {
        "clusterName": "my-cluster",
        "namespace": "default",
        "serviceAccount": "my-service-account",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws::111122223333:podidentityassociation/my-
cluster/a-abcdefghijklmnop1",
        "associationId": "a-abcdefghijklmnop1",
        "tags": {},
        "createdAt": 1700862734.922,
        "modifiedAt": 1700862734.922
```

```
    }
}
```

> **Note**
>
> Sie können den Namen eines Namespace und eines Servicekontos angeben, der bzw. das nicht im Cluster vorhanden ist. Sie müssen den Namespace, das Servicekonto und die Workload erstellen, die das Servicekonto verwendet, damit die EKS-Pod-Identity-Zuordnung funktioniert.

3. Stellen Sie sicher, dass die Rolle und das Servicekonto korrekt konfiguriert sind.

   a. Stellen Sie sicher, dass die Vertrauensrichtlinie der IAM-Rolle korrekt konfiguriert ist.

   ```
   aws iam get-role --role-name my-role --query Role.AssumeRolePolicyDocument
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "Allow EKS Auth service to assume this role for Pod
    Identities",
               "Effect": "Allow",
               "Principal": {
                   "Service": "pods.eks.amazonaws.com"
               },
               "Action": [
                   "sts:AssumeRole",
                   "sts:TagSession"
               ]
           }
       ]
   }
   ```

   b. Stellen Sie sicher, dass die Richtlinie, die Sie in einem vorherigen Schritt an Ihre Rolle angehängt haben, an die Rolle angehängt ist.

```
aws iam list-attached-role-policies --role-name my-role --query
 AttachedPolicies[].PolicyArn --output text
```

Eine Beispielausgabe sieht wie folgt aus.

```
arn:aws:iam::111122223333:policy/my-policy
```

c. Legen Sie eine Variable fest, um den Amazon-Ressourcennamen (ARN) der Richtlinie zu speichern, die Sie verwenden möchten. Ersetzen Sie *my-policy* durch den Namen der Richtlinie, für die Sie Berechtigungen überprüfen möchten.

```
export policy_arn=arn:aws:iam::111122223333:policy/my-policy
```

d. Zeigen Sie die Standardversion der Richtlinie an.

```
aws iam get-policy --policy-arn $policy_arn
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "EXAMPLEBIOWGLDEXAMPLE",
        "Arn": "arn:aws:iam::111122223333:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        [...]
    }
}
```

e. Zeigen Sie den Inhalt der Richtlinie an, um sicherzustellen, dass die Richtlinie alle Berechtigungen enthält, die Ihr Pod erfordert. Ersetzen Sie im folgenden Befehl *1* durch die Version, die in der vorherigen Ausgabe zurückgegeben wurde.

```
aws iam get-policy-version --policy-arn $policy_arn --version-id v1
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-pod-secrets-bucket"
        }
    ]
}
```

Wenn Sie die Beispielrichtlinie in einem vorherigen Schritt erstellt haben, ist Ihre Ausgabe dieselbe. Wenn Sie eine andere Richtlinie erstellt haben, sieht der *Beispiel*-Inhalt anders aus.

Nächster Schritt

Für Pods die Verwendung eines Kubernetes Dienstkontos konfigurieren

Für Pods die Verwendung eines Kubernetes Dienstkontos konfigurieren

Wenn ein Zugriff Pod erforderlich ist AWS-Services, müssen Sie es für die Verwendung eines Kubernetes Dienstkontos konfigurieren. Das Dienstkonto muss einer AWS Identity and Access Management (IAM-) Rolle zugeordnet sein, die über Zugriffsberechtigungen für die AWS-Services verfügt.

Voraussetzungen

- Einen vorhandenen -Cluster. Wenn Sie keine haben, können Sie eine mit einer der Erste Schritte mit Amazon EKS-Hilfslinien erstellen.
- Ein vorhandenes Kubernetes-Servicekonto und eine EKS-Pod-Identity-Zuordnung, die das Servicekonto einer IAM-Rolle zuordnet. Die Rolle muss über eine zugeordnete IAM-Richtlinie verfügen, die die Berechtigungen enthält, die Ihre Pods zur Verwendung von AWS-Services haben sollen. Weitere Informationen zu Rollen, ihren Vorteilen sowie zu ihrer Erstellung und Konfiguration finden Sie unter Konfigurieren Sie ein Kubernetes Dienstkonto, um eine IAM-Rolle mit EKS Pod Identity anzunehmen.
- Die neueste Version von, die auf Ihrem Gerät AWS CLI installiert und konfiguriert ist, oder AWS CloudShell. Sie können Ihre aktuelle Version mit `aws --version | cut -d / -f2 | cut -d ' ' -f1` überprüfen. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der

neuesten Version finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit `aws configure` im AWS Command Line Interface Benutzerhandbuch. Die in der installierte AWS CLI Version AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- Eine vorhandene `kubectl config`-Datei, die Ihre Clusterkonfiguration enthält. Informationen zum Erstellen einer `kubectl config`-Datei finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

Konfigurieren von Pod zur Verwendung eines Servicekontos

1. Verwenden Sie den folgenden Befehl, um ein Bereitstellungsmanifest zu erstellen, mit dem Sie einen Pod bereitstellen können, um die Konfiguration zu bestätigen. Ersetzen Sie *`example values`* durch Ihre eigenen Werte.

```
cat >my-deployment.yaml <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      serviceAccountName: my-service-account
      containers:
      - name: my-app
        image: public.ecr.aws/nginx/nginx:X.XX
EOF
```

2. Stellen Sie das Manifest in Ihrem Cluster bereit.

```
kubectl apply -f my-deployment.yaml
```

3. Vergewissern Sie sich, dass die erforderlichen Umgebungsvariablen für Ihren Pod vorhanden sind.

   a. Zeigen Sie die Pods an, die im vorherigen Schritt bereitgestellt wurden.

   ```
   kubectl get pods | grep my-app
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   my-app-6f4dfff6cb-76cv9   1/1     Running   0          3m28s
   ```

   b. Vergewissern Sie sich, dass der Pod ein Token-Datei-Mount für ein Servicekonto hat.

   ```
   kubectl describe pod my-app-6f4dfff6cb-76cv9 | grep
    AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE:
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE:  /var/run/secrets/
   pods.eks.amazonaws.com/serviceaccount/eks-pod-identity-token
   ```

4. Vergewissern Sie Pods sich, dass Sie AWS-Services mit den Berechtigungen interagieren können, die Sie in der IAM-Richtlinie für Ihre Rolle zugewiesen haben.

> **Note**
>
> Wenn a AWS Anmeldeinformationen aus einer IAM-Rolle Pod verwendet, die einem Dienstkonto zugeordnet ist, Pod verwenden die AWS CLI oder andere SDKs in den entsprechenden Containern die Anmeldeinformationen, die von dieser Rolle bereitgestellt werden. Der Pod hat weiterhin Zugriff auf die der Amazon-EKS-Knoten-IAM-Rolle bereitgestellten Anmeldeinformationen, es sei denn, Sie beschränken den Zugriff auf diese Anmeldeinformationen. Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Wenn Ihre Pods nicht wie erwartet mit den Services interagieren können, führen Sie die folgenden Schritte aus, um sicherzustellen, dass alles richtig konfiguriert ist.

a. Vergewissern Sie sich, dass Sie eine AWS SDK-Version Pods verwenden, die die Übernahme einer IAM-Rolle über eine EKS Pod Identity-Zuordnung unterstützt. Weitere Informationen finden Sie unter Verwenden Sie ein unterstütztes AWS SDK.

b. Stellen Sie sicher, dass die Bereitstellung das Servicekonto verwendet.

```
kubectl describe deployment my-app | grep "Service Account"
```

Eine Beispielausgabe sieht wie folgt aus.

```
Service Account:  my-service-account
```

Definieren von Berechtigungen für EKS-Pod-Identitäten, um Rollen basierend auf Tags anzunehmen

EKS Pod Identity fügt den temporären Anmeldeinformationen für jeden Pod Tags mit Attributen wie Cluster-Name, Namespace und Servicekontoname hinzu. Diese Rollensitzungs-Tags ermöglichen es Administratoren, eine einzelne Rolle zu erstellen, die für alle Dienstkonten verwendet werden kann, indem sie den Zugriff auf AWS Ressourcen auf der Grundlage übereinstimmender Tags ermöglicht. Durch die zusätzliche Unterstützung für Rollensitzungs-Tags können Kunden engere Sicherheitsgrenzen zwischen Clustern und Workloads innerhalb von Clustern durchsetzen und gleichzeitig dieselben IAM-Rollen und IAM-Richtlinien wiederverwenden.

Die folgende Richtlinie beispielsweise erlaubt die Aktion `s3:GetObject`, wenn das Objekt mit dem Namen des EKS-Clusters gekennzeichnet ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/eks-cluster-name": "${aws:PrincipalTag/eks-
cluster-name}"
                }
            }
        }
    ]
}
```

Liste der von EKS Pod Identity hinzugefügten Sitzungs-Tags

Die folgende Liste enthält alle Schlüssel für Tags, die der `AssumeRole`-Anfrage von Amazon EKS hinzugefügt werden. Um diese Tags in Richtlinien zu verwenden, geben Sie `${aws:PrincipalTag/` gefolgt vom Schlüssel an, zum Beispiel `${aws:PrincipalTag/kubernetes-namespace}`.

- `eks-cluster-arn`
- `eks-cluster-name`
- `kubernetes-namespace`
- `kubernetes-service-account`
- `kubernetes-pod-name`
- `kubernetes-pod-uid`

Kontoübergreifende Tags

Alle Sitzungs-Tags, die von EKS Pod Identity hinzugefügt werden, sind transitiv. Die Tag-Schlüssel und -Werte werden an alle `AssumeRole`-Aktionen weitergegeben, die Ihre Workloads verwenden, um Rollen auf ein anderes Konto zu wechseln. Sie können diese Tags in Richtlinien in anderen Konten verwenden, um den Zugriff in kontoübergreifenden Szenarien einzuschränken. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Verketten von Rollen mit Sitzungs-Tags.

Benutzerdefinierte Tags

EKS Pod Identity kann der `AssumeRole`-Aktion, die es ausführt, keine zusätzlichen benutzerdefinierten Tags hinzufügen. Tags, die Sie auf die IAM-Rolle anwenden, sind jedoch immer im gleichen Format verfügbar: `${aws:PrincipalTag/` gefolgt vom Schlüssel, zum Beispiel `${aws:PrincipalTag/MyCustomTag}`.

> **Note**
>
> Tags, die der Sitzung über die `sts:AssumeRole`-Anfrage hinzugefügt wurden, haben im Konfliktfall Vorrang. Nehmen wir beispielsweise an, dass Amazon EKS der Sitzung einen Schlüssel `eks-cluster-name` und einen Wert `my-cluster` hinzufügt, wenn EKS die Kundenrolle übernimmt. Sie haben der IAM-Rolle auch ein `eks-cluster-name`-Tag mit dem Wert `my-own-cluster` hinzugefügt. In diesem Fall hat Ersteres Vorrang und der Wert für das `eks-cluster-name`-Tag ist `my-cluster`.

Verwenden Sie ein unterstütztes AWS SDK

> **Important**
>
> Eine frühere Version der Dokumentation war falsch. Das AWS SDK for Java v1 unterstützt EKS Pod Identity nicht.

Bei der Verwendung EKS-Pod-Identitäten Pods müssen die Container in Ihrem System eine AWS SDK-Version verwenden, die die Übernahme einer IAM-Rolle vom EKS Pod Identity Agent unterstützt. Stellen Sie sicher, dass Sie die folgenden Versionen oder höher für Ihr AWS SDK verwenden:

- Java (Version 2) – 2.21.30
- Go v1 – v1.47.11
- Go v2 – release-2023-11-14
- Python (Boto3) — 1.34.41
- Python (Botocore) — 1.34.41
- AWS CLI — 1,30,0

AWS CLI — 2,15,0

- JavaScript v2 — 2,150,0
- JavaScript v3 — v3.458.0
- Kotlin — v1.0.1
- Ruby – 3.188.0
- Rust — Veröffentlichung 2024-03-13
- C++ — 1.11.263
- .NET — 3.7.734.0
- PowerShell — 4.1.502
- PHP – 3.287.1

Um sicherzustellen, dass Sie ein unterstütztes SDK verwenden, befolgen Sie die Installationsanweisungen für Ihr bevorzugtes SDK unter Tools fzum Entwickeln in AWS, wenn Sie Ihre Container entwickeln.

Eine Liste der Add-Ons, die EKS Pod Identity unterstützen, finden Sie unter. Mit EKS Pod Identity kompatible Zusatzversionen

Verwenden von EKS-Pod-Identity-Anmeldeinformationen

Um die Anmeldeinformationen einer EKS Pod Identity-Zuordnung zu verwenden, kann Ihr Code jedes AWS SDK verwenden, um einen Client für einen AWS Dienst mit einem SDK zu erstellen. Standardmäßig sucht das SDK in einer Kette von Speicherorten nach zu verwendenden AWS Identity and Access Management Anmeldeinformationen. Die Anmeldeinformationen für EKS Pod Identity werden verwendet, wenn Sie bei der Erstellung des Clients keinen Anmeldeinformationsanbieter angeben oder Sie das SDK anderweitig initialisiert haben.

Das funktioniert, weil EKS-Pod-Identitäten dem Anbieter für Container-Anmeldeinformationen hinzugefügt wurden, der in einem Schritt in der standardmäßigen Anmeldeinformationskette durchsucht wird. Wenn Ihre Workloads derzeit Anmeldeinformationen verwenden, die in der Kette der Anmeldeinformationen weiter vorn stehen, werden diese Anmeldeinformationen auch dann weiter verwendet, wenn Sie eine EKS-Pod-Identity-Zuordnung für dieselbe Workload konfigurieren.

Weitere Informationen zur Funktionsweise von EKS-Pod-Identitäten finden Sie unter Funktionsweise von EKS Pod Identity.

EKS-Pod-Identity-Rolle

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEksAuthToAssumeRoleForPodIdentity",
            "Effect": "Allow",
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
    ]
}
```

`sts:AssumeRole`

EKS Pod Identity nutzt `AssumeRole`, um die IAM-Rolle anzunehmen, bevor es die temporären Anmeldeinformationen an Ihre Pods übergibt.

`sts:TagSession`

EKS Pod Identity nutzt `TagSession`, um Sitzungs-Tags in die Anfragen an AWS STS einzufügen.

Sie können diese Tags im Element condition keys in der Vertrauensrichtlinie verwenden, um zu beschränken, welche Servicekonten, Namespaces und Cluster diese Rolle verwenden dürfen.

Eine Liste der Amazon EKS-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Elastic Kubernetes Service in der Service-Autorisierungsreferenz. Um zu erfahren, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, lesen Sie von Amazon Elastic Kubernetes Service definierte Aktionen.

IAM-Rollen für Servicekonten

Anwendungen in den Containern Pod von a können ein AWS SDK oder das verwenden AWS CLI , um API-Anfragen an Nutzungs- AWS Identity and Access Management (IAM-) Berechtigungen zu AWS-Services stellen. Anwendungen müssen ihre AWS API-Anfragen mit

AWS Anmeldeinformationen signieren. IAM-Rollen für Servicekonten bieten die Möglichkeit, Anmeldeinformationen für Ihre Anwendungen zu verwalten, ähnlich wie Amazon-EC2-Instance-Profile Anmeldeinformationen für Amazon-EC2-Instances bereitstellen. Anstatt Ihre AWS Anmeldeinformationen zu erstellen und an die Container zu verteilen oder die Rolle der Amazon EC2 EC2-Instance zu verwenden, verknüpfen Sie eine IAM-Rolle mit einem Kubernetes Dienstkonto und konfigurieren Ihr Pods Konto für die Verwendung des Dienstkontos. Sie können keine IAM-Rollen für Servicekonten mit lokalen Clustern für Amazon EKS auf AWS Outposts verwenden.

IAM-Rollen für Servicekonten bietet die folgenden Vorteile:

- Geringste Berechtigung – Sie können IAM-Berechtigungen auf ein Servicekonto beschränken, und nur Pods, die dieses Servicekonto verwenden, haben Zugriff auf diese Berechtigungen. Mit diesem Feature entfällt auch die Notwendigkeit von Drittanbieterlösungen wie `kiam` oder `kube2iam`.
- Isolation von Anmeldeinformationen – Ein Container eines Pod's kann nur Anmeldeinformationen für die IAM-Rolle abrufen, die dem Servicekonto zugeordnet ist, zu dem er gehört. Ein Container hat nie Zugriff auf Anmeldeinformationen, die von anderen Containern in anderen Pods verwendet werden. Wenn IAM-Rollen für Servicekonten verwendet werden, haben Pod's-Container auch die Berechtigungen, die der Amazon-EKS-Knoten-IAM-Rolle zugeordnet sind, es sei denn, Sie blockieren den Pod-Zugang zum Amazon EC2 Instance Metadata Service (IMDS). Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.
- Überprüfbarkeit — Die Zugriffs- und Ereignisprotokollierung ist verfügbar, um eine AWS CloudTrail nachträgliche Prüfung zu gewährleisten.

Aktivieren Sie IAM-Rollen für Servicekonten, indem Sie die folgenden Verfahren ausführen:

1. Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster – Sie führen dieses Verfahren für jeden Cluster nur einmal durch.

> **Note**
>
> Wenn Sie den EKS-VPC-Endpunkt aktivieren, kann von dieser VPC aus nicht auf den EKS-OIDC-Service-Endpunkt zugegriffen werden. Folglich funktionieren Ihre Operationen wie das Erstellen eines OIDC-Anbieters mit `eksctl` in der VPC nicht und führen zu einem Timeout, wenn Sie versuchen, `https://oidc.eks.region.amazonaws.com` anzufordern. Es folgt ein Beispiel für eine Fehlermeldung:

```
** server can't find oidc.eks.region.amazonaws.com: NXDOMAIN
```

Um diesen Schritt abzuschließen, können Sie den Befehl außerhalb der VPC ausführen, z. B. in AWS CloudShell oder auf einem Computer, der mit dem Internet verbunden ist.

2. Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle – Führen Sie dieses Verfahren für jeden eindeutigen Satz von Berechtigungen aus, über den eine Anwendung verfügen soll.
3. Für Pods die Verwendung eines Kubernetes Dienstkontos konfigurieren— Führen Sie dieses Verfahren für jeden ausPod, auf den Sie Zugriff AWS-Services benötigen.
4. Verwendung eines unterstützten AWS -SDK— Vergewissern Sie sich, dass der Workload ein AWS SDK einer unterstützten Version verwendet und dass der Workload die standardmäßige Anmeldeinformationskette verwendet.

Hintergrundinformationen zu IAM, Kubernetes und OpenID Connect (OIDC)

2014 wurde die Unterstützung für föderierte Identitäten mit OpenID Connect () AWS Identity and Access Management hinzugefügt. OIDC Mit dieser Funktion können Sie AWS API-Aufrufe bei unterstützten Identitätsanbietern authentifizieren und ein gültiges OIDC JSON Web-Token () erhalten. JWT Sie können dieses Token an den AWS STS `AssumeRoleWithWebIdentity` API-Vorgang übergeben und temporäre IAM-Rollenanmeldedaten erhalten. Sie können diese Anmeldeinformationen verwenden, um mit jedem AWS-Service wie Amazon S3 und DynamoDB zu interagieren.

Jedes JWT-Token ist mit einem Signaturschlüsselpaar signiert. Die Schlüssel werden für den von Amazon EKS verwalteten OIDC-Anbieter bereitgestellt und der private Schlüssel wechselt alle 7 Tage. Amazon EKS bewahrt die öffentlichen Schlüssel auf, bis sie ablaufen. Wenn Sie externe OIDC-Clients verbinden, beachten Sie, dass Sie die Signaturschlüssel aktualisieren müssen, bevor der öffentliche Schlüssel abläuft. Weitere Informationen erhalten Sie unter the section called “Signaturschlüssel abrufen”.

Kubernetes verwendet seit langem Servicekonten als eigenes internes Identitätssystem. Pods kann sich mit dem authentifizieren KubernetesAPI-Server unter Verwendung eines automatisch gemounteten Tokens (ein Nicht-OIDC JWT) authentifizieren, das nur der Kubernetes-API-Server validieren kann. Diese Legacy-Servicekonto-Token laufen nicht ab und das Rotieren des Signaturschlüssels ist ein schwieriger Prozess. In Kubernetes der Version `1.12` wurde Unterstützung

für ein neues `ProjectedServiceAccountToken`-Feature hinzugefügt. Dieses Feature ist ein OIDC-JSON-Web-Token, das auch die Identität des Servicekontos enthält und eine konfigurierbare Zielgruppe unterstützt.

Amazon EKS hostet jetzt einen öffentlichen OIDC-Erkennungsendpunkt pro Cluster, der die Signaturschlüssel für die `ProjectedServiceAccountToken`-JSON-Web-Token enthält, sodass externe Systeme wie IAM die von Kubernetes ausgegebenen OIDC-Token validieren und akzeptieren können.

Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster

Ihrem Cluster ist eine OpenID Connect (OIDC)-Aussteller-URL zugeordnet. Um AWS Identity and Access Management (IAM-) Rollen für Dienstkonten verwenden zu können, muss ein OIDC IAM-Anbieter für die OIDC Aussteller-URL Ihres Clusters vorhanden sein.

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.
- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder. AWS CloudShell Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- Eine vorhandene `kubectl config`-Datei, die Ihre Clusterkonfiguration enthält. Informationen zum Erstellen einer `kubectl config`-Datei finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

Sie können einen OIDC-IAM-Anbieter für Ihren Cluster mit `eksctl` oder AWS Management Console erstellen.

eksctl

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

So erstellen Sie einen IAM-OIDC-Identitätsanbieter für Ihren Cluster mit **`eksctl`**

1. Ermitteln Sie die OIDC-Aussteller-ID für Ihren Cluster.

   Rufen Sie die OIDC-Aussteller-ID Ihres Clusters ab und speichern Sie sie in einer Variable. Ersetzen Sie *`my-cluster`* durch Ihren eigenen Wert.

   ```
   cluster_name=my-cluster
   ```

   ```
   oidc_id=$(aws eks describe-cluster --name $cluster_name --query
    "cluster.identity.oidc.issuer" --output text | cut -d '/' -f 5)
   ```

   ```
   echo $oidc_id
   ```

2. Bestimmen Sie, ob ein IAM-OIDC-Anbieter mit der Aussteller-ID Ihres Clusters bereits in Ihrem Konto vorhanden ist.

   ```
   aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
   ```

   Wenn die Ausgabe zurückgegeben wird, verfügen Sie bereits über einenOIDC-IAM-Anbieter für Ihren Cluster und können den nächsten Schritt überspringen. Wenn keine Ausgabe erfolgt, müssen Sie einen IAM-OIDC-Anbieter für Ihren Cluster erstellen.

3. Erstellen Sie einen IAM-OIDC-Identitätsanbieter für Ihren Cluster mit dem folgenden Befehl.

   ```
   eksctl utils associate-iam-oidc-provider --cluster $cluster_name --approve
   ```

> **Note**
>
> Wenn Sie den EKS-VPC-Endpunkt aktivieren, kann von dieser VPC aus nicht auf den EKS-OIDC-Service-Endpunkt zugegriffen werden. Folglich funktionieren Ihre Operationen wie das Erstellen eines OIDC-Anbieters mit `eksctl` in der VPC nicht und führen zu einem Timeout, wenn Sie versuchen, `https://oidc.eks.region.amazonaws.com` anzufordern. Es folgt ein Beispiel für eine Fehlermeldung:
>
> ```
> ** server can't find oidc.eks.region.amazonaws.com: NXDOMAIN
> ```
>
> Um diesen Schritt abzuschließen, können Sie den Befehl außerhalb der VPC ausführen, z. B. in AWS CloudShell oder auf einem Computer, der mit dem Internet verbunden ist.

AWS Management Console

Um einen OIDC IAM-Identitätsanbieter für Ihren Cluster mit dem zu erstellen AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Bereich Cluster aus und wählen Sie dann den Namen Ihres Clusters auf der Seite Cluster.
3. Notieren Sie im Abschnitt Details der Registerkarte Overview (Übersicht) den Wert der OpenID-Connect-Provider-URL.
4. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
5. Wählen Sie im linken Navigationsbereich Identity Providers (Identitätsanbieter) unter Access management (Zugriffsverwaltung) aus. Wenn ein Anbieter aufgeführt ist, der mit der URL für Ihren Cluster übereinstimmt, haben Sie bereits einen Anbieter für Ihren Cluster. Wenn kein Anbieter aufgeführt ist, der mit der URL für Ihren Cluster übereinstimmt, müssen Sie einen erstellen.
6. Um einen Anbieter zu erstellen, wählen Sie Add provider (Anbieter hinzufügen) aus.
7. Als Provider type (Provider-Typ) wählen Sie OpenID Connect aus.

8. Geben Sie als Provider URL (Anbieter-URL) die OIDC-Anbieter-URL für Ihren Cluster ein, und wählen Sie dann Get thumbprint (Thumbprint abrufen) aus.
9. Geben Sie für Zielgruppe **`sts.amazonaws.com`** ein und wählen Sie Anbieter hinzufügen.

Nächster Schritt

Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle

Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle

In diesem Thema wird beschrieben, wie Sie ein Kubernetes Dienstkonto so konfigurieren, dass es eine AWS Identity and Access Management (IAM-) Rolle annimmt. Alle Pods, die für die Verwendung des Servicekontos konfiguriert sind, können dann auf alle AWS-Service zugreifen, für die die Rolle Zugriffsberechtigungen hat.

Voraussetzungen

- Einen vorhandenen -Cluster. Wenn Sie keine haben, können Sie eine mit einem der Erste Schritte mit Amazon EKS-Leitfäden erstellen.
- Ein vorhandener IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits einen haben oder wie Sie einen erstellen können, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.

- Eine vorhandene `kubectl config`-Datei, die Ihre Clusterkonfiguration enthält. Informationen zum Erstellen einer `kubectl config`-Datei finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

Verknüpfen Sie eine IAM-Rolle mit einem Kubernetes-Servicekonto wie folgt

1. Wenn Sie Ihrer IAM-Rolle eine vorhandene IAM-Richtlinie zuordnen möchten, fahren Sie mit dem nächsten Schritt fort.

   Erstellen Sie eine IAM-Richtlinie. Sie können Ihre eigene Richtlinie erstellen oder eine AWS verwaltete Richtlinie kopieren, die bereits einige der benötigten Berechtigungen gewährt, und sie an Ihre spezifischen Anforderungen anpassen. Weitere Informationen finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

   a. Erstellen Sie eine Datei, die die Berechtigungen für AWS-Services die Personen enthält, auf die Sie zugreifen Pods möchten. Eine Liste aller Aktionen für alle AWS-Services finden Sie in der Service Authorization Reference.

      Sie können den folgenden Befehl ausführen, um eine Beispiel-Richtliniendatei zu erstellen, die schreibgeschützten Zugriff auf einen Amazon-S3-Bucket gewährt. Sie können optional Konfigurationsinformationen oder ein Bootstrap-Skript in diesem Bucket speichern und die Container in Ihrem Pod können die Datei aus dem Bucket lesen und in Ihre Anwendung laden. Wenn Sie diese Beispielrichtlinie erstellen möchten, kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie *`my-pod-secrets-bucket`* durch Ihren Bucket-Namen und führen Sie den Befehl aus.

      ```
      cat >my-policy.json <<EOF
      {
          "Version": "2012-10-17",
          "Statement": [
              {
                  "Effect": "Allow",
                  "Action": "s3:GetObject",
                  "Resource": "arn:aws:s3:::my-pod-secrets-bucket"
              }
          ]
      }
      EOF
      ```

   b. Erstellen Sie die IAM-Richtlinie.

```
aws iam create-policy --policy-name my-policy --policy-document file://my-policy.json
```

2. Erstellen Sie eine IAM-Rolle und verknüpfen Sie sie mit einem Kubernetes-Servicekonto. Sie können entweder `eksctl` oder AWS CLI verwenden.

   eksctl

   Voraussetzung

   Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

   Ersetzen Sie *my-service-account* durch das Kubernetes-Servicekonto, das `eksctl` erstellen und mit einer IAM-Rolle verknüpfen soll. Ersetzen Sie *default* durch den Namen des Clusters, in dem `eksctl` das Servicekonto erstellen soll. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters. Ersetzen Sie *my-role* durch den Namen der Rolle, mit der das Servicekonto verknüpft werden soll. Wenn es noch nicht vorhanden ist, erstellt `eksctl` es für Sie. Ersetzen Sie *111122223333* durch Ihre Konto-ID und *my-policy* durch den Namen einer vorhandenen Richtlinie.

   ```
   eksctl create iamserviceaccount --name my-service-account --namespace default --cluster my-cluster --role-name my-role \
       --attach-policy-arn arn:aws:iam::111122223333:policy/my-policy --approve
   ```

   > **Important**
   >
   > Wenn die Rolle oder das Servicekonto bereits vorhanden ist, schlägt der vorherige Befehl möglicherweise fehl. `eksctl` hat verschiedene Optionen, die Sie in diesen Situationen angeben können. Führen Sie **`eksctl create iamserviceaccount --help`** aus, um weitere Informationen zu erhalten.

   AWS CLI

   1. Wenn Sie ein bestehendes Kubernetes-Servicekonto haben, das eine IAM-Rolle annehmen soll, können Sie diesen Schritt überspringen.

Ein Kubernetes-Service-Konto aktualisieren Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie *my-service-account* durch Ihren Wunschnamen und *default* durch einen anderen Namespace, falls erforderlich. Wenn Sie *default* ändern, muss der Namespace bereits vorhanden sein.

```
cat >my-service-account.yaml <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-service-account
  namespace: default
EOF
kubectl apply -f my-service-account.yaml
```

2. Setzen Sie Ihre AWS-Konto ID mit dem folgenden Befehl auf eine Umgebungsvariable.

```
account_id=$(aws sts get-caller-identity --query "Account" --output text)
```

3. Legen Sie des OIDC-Identitätsanbieter Ihres Clusters mit dem folgenden Befehl auf eine Umgebungsvariable fest. Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.

```
oidc_provider=$(aws eks describe-cluster --name my-cluster --region
 $AWS_REGION --query "cluster.identity.oidc.issuer" --output text | sed -e "s/
^https:\/\///")
```

4. Legen Sie Variablen für den Namespace und den Namen des Servicekontos fest. Ersetzen Sie *my-service-account* durch das Kubernetes-Servicekonto, das die Rolle annehmen soll. Ersetzen Sie *default* durch den Namespace des Servicekontos.

```
export namespace=default
export service_account=my-service-account
```

5. Führen Sie den folgenden Befehl aus, um eine Vertrauensrichtlinie für die IAM-Rolle zu erstellen. Wenn Sie allen Servicekonten innerhalb eines Namespaces die Verwendung der Rolle erlauben möchten, kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie durch **`StringLike`** und *`StringEquals`*ersetzen Sie *$service_account* durch. ***** Sie können mehrere Einträge in den Bedingungen `StringEquals` und `StringLike` unten hinzufügen, um mehreren Servicekonten oder Namespaces das Annehmen der Rolle zu erlauben. Damit Rollen von einem anderen AWS-Konto als dem Konto, in dem

sich Ihr Cluster befindet, die Rolle übernehmen können, siehe Kontoübergreifende IAM-Berechtigungen für weitere Informationen.

```
cat >trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::$account_id:oidc-provider/$oidc_provider"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "$oidc_provider:aud": "sts.amazonaws.com",
          "$oidc_provider:sub": "system:serviceaccount:
$namespace:$service_account"
        }
      }
    }
  ]
}
EOF
```

6. Erstellen Sie die -Rolle. Ersetzen Sie *my-role* durch einen Namen für Ihre IAM-Rolle und *my-role-description* durch eine Beschreibung für Ihre Rolle.

```
aws iam create-role --role-name my-role --assume-role-policy-document
 file://trust-relationship.json --description "my-role-description"
```

7. Hängen Sie eine IAM-Richtlinie an Ihre Rolle an. Ersetzen Sie *my-role* durch den Namen Ihrer IAM-Rolle und *my-policy* durch den Namen einer vorhandenen Richtlinie, die Sie erstellt haben.

```
aws iam attach-role-policy --role-name my-role --policy-arn=arn:aws:iam::
$account_id:policy/my-policy
```

8. Annotieren Sie Ihr Servicekonto mit dem Amazon-Ressourcennamen (ARN) der IAM-Rolle, die das Servicekonto annehmen soll. Ersetzen Sie *my-role* durch den Namen Ihrer bestehenden IAM-Rolle. Angenommen, Sie haben in einem vorherigen Schritt zugelassen, dass eine Rolle von einem anderen Konto AWS-Konto als dem Konto, in dem sich Ihr

Cluster befindet, die Rolle übernimmt. Stellen Sie dann sicher, dass Sie die Rolle AWS-Konto und aus dem anderen Konto angeben. Weitere Informationen finden Sie unter Kontoübergreifende IAM-Berechtigungen.

```
kubectl annotate serviceaccount -n $namespace $service_account
 eks.amazonaws.com/role-arn=arn:aws:iam::$account_id:role/my-role
```

3. Stellen Sie sicher, dass die Rolle und das Servicekonto korrekt konfiguriert sind.

   a. Stellen Sie sicher, dass die Vertrauensrichtlinie der IAM-Rolle korrekt konfiguriert ist.

   ```
   aws iam get-role --role-name my-role --query Role.AssumeRolePolicyDocument
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Principal": {
                   "Federated": "arn:aws:iam::111122223333:oidc-provider/
   oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
               },
               "Action": "sts:AssumeRoleWithWebIdentity",
               "Condition": {
                   "StringEquals": {
                       "oidc.eks.region-code.amazonaws.com/
   id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:default:my-
   service-account",
                       "oidc.eks.region-code.amazonaws.com/
   id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
                   }
               }
           }
       ]
   }
   ```

   b. Stellen Sie sicher, dass die Richtlinie, die Sie in einem vorherigen Schritt an Ihre Rolle angehängt haben, an die Rolle angehängt ist.

```
aws iam list-attached-role-policies --role-name my-role --query
 AttachedPolicies[].PolicyArn --output text
```

Eine Beispielausgabe sieht wie folgt aus.

```
arn:aws:iam::111122223333:policy/my-policy
```

c. Legen Sie eine Variable fest, um den Amazon-Ressourcennamen (ARN) der Richtlinie zu speichern, die Sie verwenden möchten. Ersetzen Sie *my-policy* durch den Namen der Richtlinie, für die Sie Berechtigungen überprüfen möchten.

```
export policy_arn=arn:aws:iam::111122223333:policy/my-policy
```

d. Zeigen Sie die Standardversion der Richtlinie an.

```
aws iam get-policy --policy-arn $policy_arn
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "EXAMPLEBIOWGLDEXAMPLE",
        "Arn": "arn:aws:iam::111122223333:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        [...]
    }
}
```

e. Zeigen Sie den Inhalt der Richtlinie an, um sicherzustellen, dass die Richtlinie alle Berechtigungen enthält, die Ihr Pod erfordert. Ersetzen Sie im folgenden Befehl *1* durch die Version, die in der vorherigen Ausgabe zurückgegeben wurde.

```
aws iam get-policy-version --policy-arn $policy_arn --version-id v1
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-pod-secrets-bucket"
        }
    ]
}
```

Wenn Sie die Beispielrichtlinie in einem vorherigen Schritt erstellt haben, ist Ihre Ausgabe dieselbe. Wenn Sie eine andere Richtlinie erstellt haben, sieht der *Beispiel*-Inhalt anders aus.

f. Stellen Sie sicher, dass das Kubernetes-Servicekonto mit der Rolle annotiert ist.

```
kubectl describe serviceaccount my-service-account -n default
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:                my-service-account
Namespace:           default
Annotations:         eks.amazonaws.com/role-arn:
 arn:aws:iam::111122223333:role/my-role
Image pull secrets:  <none>
Mountable secrets:   my-service-account-token-qqjfl
Tokens:              my-service-account-token-qqjfl
[...]
```

4. (Fakultativ)Den AWS Security Token Service Endpunkt für ein Dienstkonto konfigurieren. AWS empfiehlt die Verwendung eines regionalen AWS STS Endpunkts anstelle des globalen Endpunkts. Dies reduziert die Latenz, bietet integrierte Redundanz und erhöht die Gültigkeit der Sitzungstoken.

Nächster Schritt

Für Pods die Verwendung eines Kubernetes Dienstkontos konfigurieren

Für Pods die Verwendung eines Kubernetes Dienstkontos konfigurieren

Wenn ein Zugriff Pod erforderlich ist AWS-Services, müssen Sie es für die Verwendung eines Kubernetes Dienstkontos konfigurieren. Das Dienstkonto muss einer AWS Identity and Access Management (IAM-) Rolle zugeordnet sein, die über Zugriffsberechtigungen für die AWS-Services verfügt.

Voraussetzungen

- Einen vorhandenen -Cluster. Wenn Sie keine haben, können Sie eine mit einer der Erste Schritte mit Amazon EKS-Hilfslinien erstellen.
- Ein vorhandener IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits einen haben oder wie Sie einen erstellen können, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Ein vorhandenes Kubernetes-Servicekonto, das einer IAM-Rolle zugeordnet ist. Das Servicekonto muss mit dem Amazon-Ressourcennamen (ARN) der IAM-Rolle versehen sein. Die Rolle muss über eine zugeordnete IAM-Richtlinie verfügen, die die Berechtigungen enthält, die Ihre Pods zur Verwendung von AWS-Services haben sollen. Weitere Informationen zu Rollen, ihren Vorteilen sowie zu ihrer Erstellung und Konfiguration finden Sie unter Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.
- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.

- Eine vorhandene `kubectl config`-Datei, die Ihre Clusterkonfiguration enthält. Informationen zum Erstellen einer `kubectl config`-Datei finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

Konfigurieren von Pod zur Verwendung eines Servicekontos

1. Verwenden Sie den folgenden Befehl, um ein Bereitstellungsmanifest zu erstellen, mit dem Sie einen Pod bereitstellen können, um die Konfiguration zu bestätigen. Ersetzen Sie *`example values`* durch Ihre eigenen Werte.

```
cat >my-deployment.yaml <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      serviceAccountName: my-service-account
      containers:
      - name: my-app
        image: public.ecr.aws/nginx/nginx:X.XX
EOF
```

2. Stellen Sie das Manifest in Ihrem Cluster bereit.

```
kubectl apply -f my-deployment.yaml
```

3. Vergewissern Sie sich, dass die erforderlichen Umgebungsvariablen für Ihren Pod vorhanden sind.

   a. Zeigen Sie die Pods an, die im vorherigen Schritt bereitgestellt wurden.

```
kubectl get pods | grep my-app
```

Eine Beispielausgabe sieht wie folgt aus.

```
my-app-6f4dfff6cb-76cv9   1/1     Running   0          3m28s
```

b. Zeigen Sie den ARN der IAM-Rolle an, den der Pod verwendet.

```
kubectl describe pod my-app-6f4dfff6cb-76cv9 | grep AWS_ROLE_ARN:
```

Eine Beispielausgabe sieht wie folgt aus.

```
AWS_ROLE_ARN:                 arn:aws:iam::111122223333:role/my-role
```

Der Rollen-ARN muss mit dem Rollen-ARN übereinstimmen, mit dem Sie das vorhandene Servicekonto mit Anmerkungen versehen haben. Weitere Informationen zum Annotieren des Servicekontos finden Sie unter Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.

c. Bestätigen Sie, dass der Pod ein Webidentitäts-Token-Datei-Mount hat.

```
kubectl describe pod my-app-6f4dfff6cb-76cv9 | grep
 AWS_WEB_IDENTITY_TOKEN_FILE:
```

Eine Beispielausgabe sieht wie folgt aus.

```
AWS_WEB_IDENTITY_TOKEN_FILE:  /var/run/secrets/eks.amazonaws.com/
serviceaccount/token
```

Das `kubelet` fordert das Token im Namen des Pod an und speichert es. Standardmäßig aktualisiert das `kubelet` das Token, wenn es älter als 80 Prozent seiner gesamten TTL ist oder wenn das Token älter als 24 Stunden ist. Sie können die Ablaufdauer für jedes Konto mit Ausnahme des Standardservicekontos mit den Einstellungen in Ihrer Pod-Spezifikation ändern. Weitere Informationen finden Sie unter Service Account Token Volume Projection in der Kubernetes-Dokumentation.

Der Pod-Identitäts-Webhook von Amazon EKS auf dem Cluster sucht nach Pods, die ein Servicekonto mit der folgenden Anmerkung verwenden:

```
eks.amazonaws.com/role-arn: arn:aws:iam::111122223333:role/my-role
```

Der Webhook wendet die vorherigen Umgebungsvariablen auf diese Pods an. Ihr Cluster muss nicht den Webhook verwenden, um die Umgebungsvariablen und die Token-Datei-Mounts zu konfigurieren. Sie können Pods manuell so konfigurieren, dass diese Umgebungsvariablen vorhanden sind. Die unterstützten Versionen des AWS SDK suchen zuerst im Credential Chain Provider nach diesen Umgebungsvariablen. Die Anmeldeinformationen der Rolle werden für Pods verwendet, die diese Kriterien erfüllen.

4. Vergewissern Sie Pods sich, dass Sie AWS-Services mit den Berechtigungen interagieren können, die Sie in der Ihrer Rolle zugewiesenen IAM-Richtlinie zugewiesen haben.

> **Note**
>
> Wenn a AWS Anmeldeinformationen aus einer IAM-Rolle Pod verwendet, die einem Dienstkonto zugeordnet ist, Pod verwenden die AWS CLI oder andere SDKs in den entsprechenden Containern die Anmeldeinformationen, die von dieser Rolle bereitgestellt werden. Der Pod hat weiterhin Zugriff auf die der Amazon-EKS-Knoten-IAM-Rolle bereitgestellten Anmeldeinformationen, es sei denn, Sie beschränken den Zugriff auf diese Anmeldeinformationen. Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Wenn Ihre Pods nicht wie erwartet mit den Services interagieren können, führen Sie die folgenden Schritte aus, um sicherzustellen, dass alles richtig konfiguriert ist.

a. Vergewissern Sie sich, dass Sie eine AWS SDK-Version Pods verwenden, die die Übernahme einer IAM-Rolle über eine OpenID Connect Web-Identity-Token-Datei unterstützt. Weitere Informationen finden Sie unter Verwendung eines unterstützten AWS -SDK.

b. Stellen Sie sicher, dass die Bereitstellung das Servicekonto verwendet.

```
kubectl describe deployment my-app | grep "Service Account"
```

Eine Beispielausgabe sieht wie folgt aus.

```
Service Account:  my-service-account
```

c. Wenn Ihre Pods immer noch nicht auf die Services zugreifen können, überprüfen Sie die in Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle beschriebenen Schritte, um sicherzustellen, dass Ihre Rolle und Ihr Servicekonto richtig konfiguriert sind.

Den AWS Security Token Service Endpunkt für ein Dienstkonto konfigurieren

Wenn Sie ein Kubernetes Dienstkonto mit verwendenIAM-Rollen für Servicekonten, können Sie den AWS Security Token Service Endpunkttyp konfigurieren, der vom Dienstkonto verwendet wird, falls Ihr Cluster und Ihre Plattformversion mit den in der folgenden Tabelle aufgeführten Versionen identisch oder höher sind. Wenn Ihre Kubernetes- oder Plattformversion älter als die in der Tabelle aufgeführten ist, können Ihre Servicekonten nur den globalen Endpunkt verwenden.

Kubernetes-Version	Plattformversion	Standardtyp für Endpunkt
`1.30`	`eks.2`	Regional
`1.29`	`eks.1`	Regional
`1.28`	`eks.1`	Regional
`1.27`	`eks.1`	Regional
`1.26`	`eks.1`	Regional
`1.25`	`eks.1`	Regional
`1.24`	`eks.2`	Regional
`1.23`	`eks.1`	Regional

AWS empfiehlt die Verwendung der regionalen AWS STS Endpunkte anstelle des globalen Endpunkts. Dies reduziert die Latenz, bietet integrierte Redundanz und erhöht die Gültigkeit der Sitzungstoken. Der AWS Security Token Service muss dort aktiv sein AWS-Region , wo der ausgeführt Pod wird. Darüber hinaus muss Ihre Anwendung über eine integrierte Redundanz für den AWS-Region Fall eines Ausfalls des Dienstes im AWS-Region verfügen. Weitere Informationen finden Sie unter Managing AWS STS in an AWS-Region im IAM-Benutzerhandbuch.

Voraussetzungen

- Einen vorhandenen -Cluster. Wenn Sie keine haben, können Sie eine mit einer der Erste Schritte mit Amazon EKS-Hilfslinien erstellen.
- Ein vorhandener IAM-OIDC-Anbieter für Ihren Cluster. Weitere Informationen finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Ein vorhandenes Kubernetes-Servicekonto, das für die Verwendung mit dem Feature Amazon-EKS-IAM für Servicekonten.

Konfigurieren Sie den von Ihrem Kubernetes-Servicekonto verwendeten Endpunkttyp wie folgt

Die folgenden Beispiele verwenden alle das `aws-node`-Kubernetes-Servicekonto, das vom Amazon-VPC-CNI-Plugin verwendet wird. Sie können die *`example values`* durch eigene Servicekonten, Pods, Namespaces und anderen Ressourcen ersetzen.

1. Wählen Sie einen Pod aus, der ein Servicekonto verwendet, für das Sie den Endpunkt ändern möchten. Ermitteln Sie, in welchem AWS-Region das Pod ausgeführt wird. Ersetzen Sie *`aws-node-6mfgv`* mit Ihrem Pod-Namen und *`kube-system`* mit Ihrem Pod-Namespace.

   ```
   kubectl describe pod aws-node-6mfgv -n kube-system |grep Node:
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   ip-192-168-79-166.us-west-2/192.168.79.166
   ```

   AWS-Region
2. Ermitteln Sie den Endpunkt-Typ, den das Pod's-Servicekonto verwendet.

   ```
   kubectl describe pod aws-node-6mfgv -n kube-system |grep AWS_STS_REGIONAL_ENDPOINTS
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   AWS_STS_REGIONAL_ENDPOINTS: regional
   ```

   Wenn der aktuelle Endpunkt global ist, dann wird `global` in der Ausgabe zurückgegeben. Wenn keine Ausgabe zurückgegeben wird, wird der Standard-Endpunkttyp verwendet und wurde nicht überschrieben.

3. Wenn Ihre Cluster- oder Plattformversion dieselbe oder höher ist als die in der Tabelle aufgeführten, können Sie den von Ihrem Servicekonto verwendeten Endpunkttyp mit einem der folgenden Befehle vom Standardtyp in einen anderen Typ ändern. Ersetzen Sie *aws-node* mit dem Namen Ihres Servicekontos, *kube-system* mit dem Namespace Ihres vorhandenen -Servicekontos.

   - Wenn Ihr Standard- oder aktueller Endpunkttyp global ist und Sie ihn in regional ändern möchten:

     ```
     kubectl annotate serviceaccount -n kube-system aws-node eks.amazonaws.com/sts-regional-endpoints=true
     ```

     Wenn Sie IAM-Rollen für Servicekonten verwenden, um vorsignierte S3-URLs in Ihrer Anwendung zu generieren, die in Containern von Pods ausgeführt wird, ähnelt das Format der URL für regionale Endpunkte dem folgenden Beispiel:

     ```
     https://bucket.s3.us-west-2.amazonaws.com/path?...&X-Amz-Credential=your-access-key-id/date/us-west-2/s3/aws4_request&...
     ```

   - Wenn Ihr Standard- oder aktueller Endpunkttyp regional ist und Sie ihn in global ändern möchten:

     ```
     kubectl annotate serviceaccount -n kube-system aws-node eks.amazonaws.com/sts-regional-endpoints=false
     ```

     Wenn Ihre Anwendung explizit Anfragen an AWS STS globale Endpunkte sendet und Sie das Standardverhalten der Verwendung regionaler Endpunkte in Amazon EKS-Clustern nicht außer Kraft setzen, schlagen Anfragen mit einem Fehler fehl. Weitere Informationen finden Sie unter Pod-Container erhalten folgenden Fehler: `An error occurred (SignatureDoesNotMatch) when calling the GetCallerIdentity operation: Credential should be scoped to a valid region`.

     Wenn Sie IAM-Rollen für Servicekonten verwenden, um vorsignierte S3-URLs in Ihrer Anwendung zu generieren, die in Containern von Pods ausgeführt wird, ähnelt das Format der URL für globale Endpunkte dem folgenden Beispiel:

     ```
     https://bucket.s3.amazonaws.com/path?...&X-Amz-Credential=your-access-key-id/date/us-west-2/s3/aws4_request&...
     ```

Wenn Ihre Automatisierung die vorsignierte URL in einem bestimmten Format erwartet, oder wenn Ihre Anwendung oder Downstream-Abhängigkeiten, die vorsignierte URLs verwenden, Erwartungen an das Ziel stellen, nehmen Sie die AWS-Region erforderlichen Änderungen vor, um den entsprechenden Endpunkt zu verwenden. AWS STS

4. Löschen Sie alle vorhandenen Pods, die dem Servicekonto zugeordnet sind, und erstellen Sie sie neu, um die Umgebungsvariablen für Anmeldeinformationen anzuwenden. Der mutierende Webhook wendet sie nicht auf Pods an, die bereits ausgeführt werden. Sie können *Pods*, *kube-system* und *-l k8s-app=aws-node* mit den Informationen für die Pods ersetzen, für die Sie Ihre Anmerkung festlegen.

```
kubectl delete Pods -n kube-system -l k8s-app=aws-node
```

5. Bestätigen Sie, dass alle Pods neu gestartet wurden.

```
kubectl get Pods -n kube-system -l k8s-app=aws-node
```

6. Zeigen Sie die Umgebungsvariablen für einen der Pods an. Stellen Sie sicher, dass der `AWS_STS_REGIONAL_ENDPOINTS`-Wert der ist, den Sie im vorherigen Schritt festgelegt haben.

```
kubectl describe pod aws-node-kzbtr -n kube-system |grep AWS_STS_REGIONAL_ENDPOINTS
```

Eine Beispielausgabe sieht wie folgt aus.

```
AWS_STS_REGIONAL_ENDPOINTS=regional
```

Kontoübergreifende IAM-Berechtigungen

Sie können kontoübergreifende IAM-Berechtigungen konfigurieren, indem Sie entweder einen Identitätsanbieter aus dem Cluster eines anderen Kontos erstellen oder verkettete `AssumeRole`-Operationen verwenden. In den folgenden Beispielen besitzt Konto A einen Amazon-EKS-Cluster, der IAM-Rollen für Servicekonten unterstützt. Pods, die auf diesem Cluster ausgeführt werden, müssen IAM-Berechtigungen von Konto B annehmen.

Example Erstellen eines Identitätsanbieters aus dem Cluster eines anderen Kontos

Example

In diesem Beispiel stellt Konto A dem Konto B die OpenID Connect (OIDC)-Aussteller-URL aus seinem Cluster bereit. Konto B befolgt die Anweisungen in Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster und Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle unter Verwendung der OIDC-Aussteller-URL aus dem Cluster von Konto A. Anschließend kommentiert ein Cluster-Administrator das Servicekonto im Cluster von Konto A, um die Rolle aus Konto B zu verwenden (*444455556666*).

```
apiVersion: v1
kind: ServiceAccount
metadata:
  annotations:
    eks.amazonaws.com/role-arn: arn:aws:iam::444455556666:role/account-b-role
```

Example Verwenden von verketteten **`AssumeRole`**-Operationen

Example

In diesem Beispiel erstellt Konto B eine IAM-Richtlinie mit den Berechtigungen, die Pods im Cluster von Konto A erteilt werden sollen. Konto B (*444455556666*) fügt diese Richtlinie einer IAM-Rolle mit einer Vertrauensstellung an, die `AssumeRole`-Berechtigungen für Konto A (*111122223333*) gewährt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

Konto A erstellt eine Rolle mit einer Vertrauensrichtlinie, die Anmeldeinformationen von dem Identitätsanbieter erhält, der mit der OIDC-Aussteller-Adresse des Clusters erstellt wurde.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.region-
code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity"
    }
  ]
}
```

Konto A fügt dieser Rolle eine Richtlinie mit den folgenden Berechtigungen an, um die Rolle anzunehmen, die Konto B erstellt hat.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::444455556666:role/account-b-role"
        }
    ]
}
```

Der Anwendungscode für Pods, die die Rolle von Konto B übernehmen sollen, verwendet zwei Profile: `account_b_role` und `account_a_role`. Das `account_b_role`-Profil verwendet das `account_a_role`-Profil als Quelle. Für AWS CLI die ähnelt die `~/.aws/config` Datei der folgenden.

```
[profile account_b_role]
source_profile = account_a_role
role_arn=arn:aws:iam::444455556666:role/account-b-role

[profile account_a_role]
web_identity_token_file = /var/run/secrets/eks.amazonaws.com/serviceaccount/token
role_arn=arn:aws:iam::111122223333:role/account-a-role
```

Informationen zur Angabe verketteter Profile für andere AWS SDKs finden Sie in der Dokumentation zu dem SDK, das Sie verwenden. Weitere Informationen finden Sie unter Tools, auf denen Sie aufbauen können. AWS

Verwendung eines unterstützten AWS -SDK

Bei der Verwendung IAM-Rollen für Servicekonten Pods müssen die Container in Ihrem eine AWS SDK-Version verwenden, die die Übernahme einer IAM-Rolle über eine OpenID Connect Web-Identity-Tokendatei unterstützt. Stellen Sie sicher, dass Sie die folgenden Versionen oder höher für Ihr AWS SDK verwenden:

- Java (Version 2) – 2.10.11
- Java – 1.11.704
- Go – 1.23.13
- Python (Boto3) — 1.9.220
- Python (botocore) —1.12.200
- AWS CLI — 1.16.232
- Knoten – 2.525.0 und 3.27.0
- Ruby – 3.58.0
- C++ – 1.7.174
- .NET – 3.3.659.1 – Sie müssen auch `AWSSDK.SecurityToken` angeben.
- PHP – 3.110.7

Viele beliebte Kubernetes-Add-ons wie der Cluster-Autoscaler, Was ist die AWS Load Balancer Controller? und die Amazon VPC CNI plugin for Kubernetes, unterstützen IAM-Rollen für Servicekonten.

Um sicherzustellen, dass Sie ein unterstütztes SDK verwenden, befolgen Sie die Installationsanweisungen für Ihr bevorzugtes SDK unter Tools fzum Entwickeln in AWS, wenn Sie Ihre Container entwickeln.

Verwenden von Anmeldeinformationen

Um die Anmeldeinformationen von IAM-Rollen für Dienstkonten zu verwenden, kann Ihr Code ein beliebiges AWS SDK verwenden, um einen Client für einen AWS Service mit einem SDK zu erstellen. Standardmäßig sucht das SDK in einer Kette von Speicherorten nach zu

verwendenden AWS Identity and Access Management Anmeldeinformationen. Die IAM-Rollen für Anmeldeinformationen für Servicekonten werden verwendet, wenn Sie bei der Erstellung des Clients keinen Anmeldeinformationsanbieter angeben oder Sie das SDK anderweitig initialisiert haben.

Das funktioniert, weil IAM-Rollen für Servicekonten als Schritt in der standardmäßigen Anmeldeinformationskette hinzugefügt wurden. Wenn Ihre Workloads derzeit Anmeldeinformationen verwenden, die sich an früherer Stelle in der Anmeldeinformationskette befinden, werden diese Anmeldeinformationen auch dann weiterhin verwendet, wenn Sie IAM-Rollen für Servicekonten für dieselbe Workload konfigurieren.

Das SDK tauscht mithilfe der `AssumeRoleWithWebIdentity` Aktion automatisch das OIDC Dienstkonto-Token gegen temporäre Anmeldeinformationen AWS Security Token Service von aus aus. Amazon EKS und diese SDK-Aktion rotieren weiterhin die temporären Anmeldeinformationen, indem sie erneuert werden, bevor sie ablaufen.

Signaturschlüssel abrufen

Kubernetesgibt jeweils `ProjectedServiceAccountToken` einen aus KubernetesService Account. Dieses Token ist ein OIDC Token, das außerdem ein Typ von istJSON web token (JWT). Amazon EKS hostet für jeden Cluster einen öffentlichen OIDC Endpunkt, der die Signaturschlüssel für das Token enthält, sodass externe Systeme es validieren können.

Um einen zu validieren`ProjectedServiceAccountToken`, müssen Sie die OIDC öffentlichen Signaturschlüssel, auch bekannt als, abrufen. JSON Web Key Set (JWKS) Verwenden Sie diese Schlüssel in Ihrer Anwendung, um das Token zu validieren. Sie können beispielsweise die Python-Bibliothek PyJWT verwenden, um Token mit diesen Schlüsseln zu validieren. Weitere Informationen zu finden Sie unter`ProjectedServiceAccountToken`. the section called “Hintergrundinformationen zu IAM, Kubernetes und OpenID Connect (OIDC)”

Voraussetzungen

- Ein vorhandener AWS Identity and Access Management (IAM) OpenID Connect (OIDC) -Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- AWS CLI— Ein Befehlszeilentool für die Arbeit mit AWS Diensten, einschließlich Amazon EKS. Weitere Informationen finden Sie unter Installieren, Aktualisieren und Deinstallieren von AWS CLI im AWS Command Line Interface -Benutzerhandbuch. Nach der Installation empfehlen wir AWS CLI, dass Sie es auch konfigurieren. Weitere Informationen finden Sie unter Schnellkonfiguration mit `aws configure` im AWS Command Line Interface -Benutzerhandbuch.

Rufen Sie OIDC öffentliche Signaturschlüssel ab ()AWS CLI

1. Rufen Sie die OIDC URL für Ihren Amazon EKS-Cluster mit dem ab AWS CLI.

```
$ aws eks describe-cluster --name my-cluster --query 'cluster.identity.oidc.issuer'
"https://oidc.eks.us-west-2.amazonaws.com/id/8EBDXXXX00BAE"
```

2. Rufen Sie den öffentlichen Signaturschlüssel mit curl oder einem ähnlichen Tool ab. Das Ergebnis ist ein JSON Web Key Set (JWKS).

> **Important**
>
> Amazon EKS drosselt Anrufe an den OIDC Endpunkt. Sie sollten den öffentlichen Signaturschlüssel zwischenspeichern. Respektieren Sie den in der Antwort enthaltenen `cache-control` Header.

> **Important**
>
> Amazon EKS rotiert den OIDC Signaturschlüssel alle sieben Tage.

```
$ curl https://oidc.eks.us-west-2.amazonaws.com/id/8EBDXXXX00BAE/keys
{"keys":
[{"kty":"RSA","kid":"2284XXXX4a40","use":"sig","alg":"RS256","n":"wklbXXXXMVfQ","e":"AQAB"}
```

Amazon-EKS-Knoten

Ein Kubernetes-Knoten ist ein Computer, die containerisierte Anwendungen ausführt. Jeder Knoten umfasst die folgenden Komponenten:

- Container-Laufzeit – Software, die für den Betrieb der Container verantwortlich ist.
- **`kubelet`** – Stellt sicher, dass die Container fehlerfrei sind und innerhalb ihres zugeordneten Pod ausgeführt werden.
- **`kube-proxy`** – Behält die Netzwerkregeln bei, die die Kommunikation mit Ihren Pods ermöglichen.

Weitere Informationen finden Sie unter Knoten in der Kubernetes-Dokumentation.

Ihr Amazon-EKS-Cluster kann Pods für eine beliebige Kombination von selbstverwalteten Knoten, von Amazon EKS verwalteten Knotengruppen und AWS Fargate planen. Weitere Informationen zu Knoten, die in Ihrem Cluster bereitgestellt werden, finden Sie unter Anzeigen der Kubernetes-Ressourcen.

> **Important**
>
> AWS Fargate mit Amazon EKS ist in AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) nicht verfügbar.

> **Note**
>
> Die Knoten müssen sich in derselben VPC wie die Subnetze befinden, die Sie beim Erstellen des Clusters ausgewählt haben. Die Knoten müssen sich jedoch nicht in denselben Subnetzen befinden.

Die folgende Tabelle enthält mehrere Kriterien, die bei der Entscheidung ausgewertet werden müssen, welche Optionen Ihren Anforderungen am besten entsprechen. Diese Tabelle enthält keine verbundenen Knoten, die außerhalb von Amazon EKS erstellt wurden, die nur eingesehen werden können.

> **Note**
>
> Bottlerocket weist einige spezifische Unterschiede zu den allgemeinen Informationen in dieser Tabelle auf. Weitere Informationen zu Bottlerocket finden Sie in der Dokumentation zu GitHub.

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Kann auf AWS Outposts bereitgestellt werden	Nein	Ja	Nein
Kann auf eine AWS Lokale Zone bereitgestellt werden	Nein	Ja – Weitere Informationen dazu finden Sie unter Amazon EKS und AWS Local Zones.	Nein
Kann Container ausführen, die Windows benötigen	Ja	Ja – Ihr Cluster benötigt jedoch immer noch mindestens einen Linux-Knoten (zwei für die Verfügbarkeit empfohlen).	Nein
Kann Container ausführen, die Linux benötigen	Ja	Ja	Ja
Kann Workloads ausführen, die den Inferentia-Chip benötigen	Ja – nur Amazon-Linux-Knoten	Ja – nur Amazon Linux	Nein

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Kann Workloads ausführen, die eine GPU benötigen	Ja – nur Amazon-Linux-Knoten	Ja – nur Amazon Linux	Nein
Kann Workloads ausführen, die Arm-Prozessoren benötigen	Ja	Ja	Nein
Kann AWS Bottlerocket ausführen	Ja	Ja	Nein
Pods teilen eine Kernel-Laufzeitumgebung mit anderen Pods	Ja – Alle Ihre Pods auf jedem Ihrer Knoten	Ja – Alle Ihre Pods auf jedem Ihrer Knoten	Nein – Jeder Pod hat einen dedizierten Kernel
Pods teilen CPU, Arbeitsspeicher, Speicher und Netzwerkressourcen mit anderen Pods.	Ja – Kann zu ungenutzten Ressourcen auf jedem Knoten führen	Ja – Kann zu ungenutzten Ressourcen auf jedem Knoten führen	Nein – Jeder Pod hat dedizierte Ressourcen und kann unabhängig voneinander dimensioniert werden, um die Ressourcenauslastung zu maximieren

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Pods können mehr Hardware und Speicher verwenden, als in den Pod-Spezifikationen angefordert	Ja – Wenn der Pod mehr Ressourcen benötigt als angefordert und Ressourcen auf dem Knoten verfügbar sind, kann der Pod zusätzliche Ressourcen verwenden.	Ja – Wenn der Pod mehr Ressourcen benötigt als angefordert und Ressourcen auf dem Knoten verfügbar sind, kann der Pod zusätzliche Ressourcen verwenden.	Nein – Der Pod kann jedoch mit einer größeren vCPU und Speicherk onfiguration erneut bereitges tellt werden.
Bereitstellen und Verwalten von Amazon-EC2-Instances	Ja – automatis iert über Amazon EKS, wenn Sie ein für Amazon EKS optimiertes AMI bereitgestellt haben. Wenn Sie ein benutzerd efiniertes AMI bereitgestellt haben, müssen Sie die Instance manuell aktualisi eren.	Ja – Für die manuelle Konfigura tion oder die Verwendung von Amazon EKS wurden AWS CloudFormation -Vorlagen für die Bereitstellung von Linux (x86)-, Linux (Arm)-, oder Windows-Knoten bereitges tellt.	Nein
Das Betriebssystem von Amazon-EC 2-Instances muss gesichert, gewartet und gepatcht werden	Ja	Ja	Nein

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Kann Bootstrap-Argumente bei der Bereitstellung eines Knotens bereitstellen, z. B. zusätzliche `kubelet`-Argumente.	Ja – mithilfe von `eksctl` oder einer Startvorlage mit einem benutzerdefinierten AMI	Weitere Informationen finden Sie in den Bootstrap-Skript-Nutzungsinformationen auf GitHub.	Nein
IP-Adressen können Pods aus einem anderen CIDR-Block als der dem Knoten zugewiesenen IP-Adresse zuweisen.	Ja – Verwenden einer Startvorlage mit einem benutzerdefinierten AMI. Weitere Informationen finden Sie unter Anpassen verwalteter Knoten mit Startvorlagen.	Ja – Weitere Informationen dazu finden Sie unter Benutzerdefinierte Netzwerke für Pods.	Nein
SSH im Knoten nicht möglich	Ja	Ja	Nein – Es gibt kein Knoten-Host-Betriebssystem für SSH.
Kann Ihr eigenes benutzerdefiniertes AMI auf Knoten bereitstellen	Ja – Verwenden eines Startvorlage	Ja	Nein

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Kann Ihr eigenes benutzerdefiniertes CNI auf Knoten bereitstellen	Ja – Verwenden eines Startvorlage mit einem benutzerdefinierten AMI	Ja	Nein

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Knoten-AMI muss selbst aktualisiert werden	Ja – Wenn Sie ein für Amazon EKS optimiertes AMI bereitgestellt haben, werden Sie in der Amazon-EKS-Konsole benachrichtigt, wenn Aktualisierungen verfügbar sind. Sie können die Aktualisierung mit einem Klick in der Konsole durchführen. Wenn Sie ein benutzerdefiniertes AMI bereitgestellt haben, werden Sie in der Amazon-EKS-Konsole nicht benachrichtigt, wenn Aktualisierungen verfügbar sind. Sie müssen die Aktualisierung selbst durchführen.	Ja – Verwenden anderer Tools als der Amazon-EKS-Konsole. Dies liegt daran, dass selbstverwaltete Knoten nicht mit der Amazon-EKS-Konsole verwaltet werden können.	Nein

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Muss die Kubernetes-Version des Knotens selbst aktualisieren	Ja – Wenn Sie ein für Amazon EKS optimiertes AMI bereitgestellt haben, werden Sie in der Amazon-EKS-Konsole benachrichtigt, wenn Updates verfügbar sind. Sie können die Aktualisierung mit einem Klick in der Konsole durchführen. Wenn Sie ein benutzerdefiniertes AMI bereitgestellt haben, werden Sie in der Amazon-EKS-Konsole nicht benachrichtigt, wenn Aktualisierungen verfügbar sind. Sie müssen die Aktualisierung selbst durchführen.	Ja – Verwenden anderer Tools als der Amazon-EKS-Konsole. Dies liegt daran, dass selbstverwaltete Knoten nicht mit der Amazon-EKS-Konsole verwaltet werden können.	Nein – Sie verwalten keine Knoten.

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Kann Amazon-EBS-Speicher mit Pods verwenden	Ja	Ja	Nein
Kann Amazon EFS Speicher mit Pods verwenden	Ja	Ja	Ja
Kann Speicher Amazon FSx für Lustre mit Pods verwenden	Ja	Ja	Nein
Kann Network Load Balancer für Services verwenden	Ja	Ja	Ja, wenn Sie die Erstellen eines Network Load Balancers
Pods können in einem öffentlichen Subnetz ausgeführt werden	Ja	Ja	Nein
Kann einzelnen Pods verschiedene VPC -Sicherheitsgruppen zuweisen	Ja – nur Linux-Knoten	Ja – nur Linux-Knoten	Ja
Kann Kubernetes DaemonSets ausführen	Ja	Ja	Nein
Unterstützen von `HostPort` und `HostNetwork` im Pod-Manifest	Ja	Ja	Nein
AWS-Region Verfügbarkeit	Alle von Amazon EKS unterstützten Regionen	Alle von Amazon EKS unterstützten Regionen	Einige von Amazon EKS unterstützte Regionen
Kann Container auf Amazon-EC2-Dedicated-Hosts ausführen	Ja	Ja	Nein

Kriterien	EKS-verwaltete Knotengruppen	Selbstverwaltete Knoten	AWS Fargate
Preise	Die Kosten für Amazon-EC2-Instance, auf der mehrere Pods ausgeführt werden. Weitere Informationen dazu finden Sie unter Amazon EC2 – Preise.	Die Kosten für Amazon-EC2-Instance, auf der mehrere Pods ausgeführt werden. Weitere Informationen dazu finden Sie unter Amazon EC2 – Preise.	Kosten für eine individuelle Fargate-Speicher- und CPU-Konfiguration. Jeder Pod hat seine eigenen Kosten. Weitere Informationen finden Sie unter AWS Fargate Preise.

Verwaltete Knotengruppen

Von Amazon EKS verwaltete Knotengruppen automatisieren die Bereitstellung und das Lebenszyklusmanagement von Knoten (Amazon-EC2-Instances) für Amazon-EKS-Kubernetes-Cluster.

Mit von Amazon EKS verwalteten Knotengruppen müssen Sie die Amazon-EC2-Instances, die Rechenkapazität zum Ausführen Ihrer Kubernetes-Anwendungen bereitstellen, nicht separat bereitstellen oder registrieren. Sie können Knoten für Ihren Cluster mit einem einzigen Vorgang erstellen, aktualisieren oder beenden. Knotenaktualisierungen und -beendigungen entleeren Knoten automatisch, um sicherzustellen, dass Ihre Anwendungen verfügbar bleiben.

Jeder verwaltete Knoten wird als Teil einer Amazon-EC2-Auto-Scaling-Gruppe bereitgestellt, die von Amazon EKS für Sie verwaltet wird. Jede Ressource, einschließlich der Instances und Auto-Scaling-Gruppen, wird in Ihrem AWS -Konto ausgeführt. Jede Knotengruppe wird in mehreren Availability Zones ausgeführt, die Sie definieren.

Sie können eine verwaltete Knotengruppe zu neuen oder vorhandenen Clustern hinzufügen, indem Sie die Amazon EKS-Konsole,`eksctl`, AWS CLI; AWS API oder Infrastructure-as-Code-Tools verwenden, einschließlich AWS CloudFormation. Knoten, die als Teil einer verwalteten Knotengruppe gestartet werden, werden vom Kubernetes Cluster Autoscaler automatisch für die automatische

Erkennung gekennzeichnet. Sie können die Knotengruppe verwenden, um Kubernetes-Labels auf Knoten anzuwenden und sie jederzeit zu aktualisieren.

Es gibt keine zusätzlichen Kosten für die Nutzung von durch Amazon EKS verwalteten Knotengruppen, Sie zahlen nur für die AWS -Ressourcen, die Sie bereitstellen. Dazu gehören Amazon EC2 EC2-Instances, Amazon EBS-Volumes, Amazon EKS-Cluster-Stunden und jede andere AWS Infrastruktur. Es fallen keine Mindestgebühren oder Vorauszahlungen an.

Weitere Informationen zu den ersten Schritten mit einem neuen Amazon-EKS-Cluster und einer verwalteten Knotengruppe finden Sie unter Erste Schritte mit Amazon EKS — AWS Management Console und AWS CLI.

Informationen zum Hinzufügen einer verwalteten Knotengruppe zu einem bestehenden Cluster finden Sie unter Erstellen einer verwalteten Knotengruppe.

Konzepte für verwaltete Knotengruppen

- Die von Amazon EKS verwalteten Knotengruppen erstellen und verwalten Amazon-EC2-Instances für Sie.
- Jeder verwaltete Knoten wird als Teil einer Amazon-EC2-Auto-Scaling-Gruppe bereitgestellt, die von Amazon EKS für Sie verwaltet wird. Darüber hinaus werden alle Ressourcen, einschließlich Amazon EC2 EC2-Instances und Auto Scaling Scaling-Gruppen, in Ihrem AWS Konto ausgeführt.
- Die Auto-Scaling-Gruppe einer verwalteten Knotengruppe umfasst alle Subnetze, die Sie beim Erstellen der Gruppe angeben.
- Amazon EKS markiert verwaltete Knotengruppenressourcen, sodass sie für die Verwendung des Kubernetes Cluster Autoscaler konfiguriert sind.

  > **Important**
  >
  > Wenn Sie eine zustandsbehaftete Anwendung über mehrere Availability Zones hinweg ausführen, die von Amazon-EBS-Volumes gesichert wird und den Kubernetes Auto Scaling verwendet, sollten Sie mehrere Knotengruppen konfigurieren, die jeweils für eine einzelne Availability Zone gelten. Außerdem sollten Sie das `--balance-similar-node-groups`-Feature aktivieren.

- Sie können eine benutzerdefinierte Startvorlage für ein höheres Maß an Flexibilität und Anpassung bei der Bereitstellung verwalteter Knoten verwenden. Zum Beispiel können Sie zusätzliche `kubelet`-Argumente angeben und ein benutzerdefiniertes AMI verwenden. Weitere Informationen

finden Sie unter Anpassen verwalteter Knoten mit Startvorlagen. Wenn Sie beim ersten Erstellen einer verwalteten Knotengruppe keine benutzerdefinierte Startvorlage verwenden, gibt es eine automatisch generierte Startvorlage. Ändern Sie diese automatisch generierte Vorlage nicht manuell sonst treten Fehler auf.

- Amazon EKS folgt dem Modell der gemeinsamen Verantwortlichkeit für CVEs und Sicherheitspatches in verwalteten Knotengruppen. Wenn verwaltete Knoten ein für Amazon EKS optimiertes AMI ausführen, ist Amazon EKS für die Erstellung von Patch-Versionen des AMI verantwortlich, wenn Fehler oder Probleme gemeldet werden. Wir können eine Korrektur veröffentlichen. Sie sind jedoch dafür verantwortlich, diese Patch-AMI-Versionen für Ihre verwalteten Knotengruppen bereitzustellen. Wenn verwaltete Knoten ein benutzerdefiniertes AMI ausführen, sind Sie für die Erstellung von Patch-Versionen des AMI verantwortlich, wenn Fehler oder Probleme gemeldet werden, und für die anschließende Bereitstellung des AMI. Weitere Informationen dazu finden Sie unter Aktualisieren einer verwalteten Knotengruppe.
- Von Amazon EKS verwaltete Knotengruppen können sowohl in öffentlichen als auch privaten Subnetzen gestartet werden. Wenn Sie eine verwaltete Knotengruppe am oder nach dem 22. April 2020 in einem öffentlichen Subnetz starten, muss `MapPublicIpOnLaunch` für das Subnetz auf „Wahr“ gesetzt sein, damit die Instances einem Cluster hinzugefügt werden können. Wenn das öffentliche Subnetz am `eksctl` oder nach dem 26. März 2020 mithilfe der von Amazon EKS bereitgestellten AWS CloudFormation Vorlagen erstellt wurde, ist diese Einstellung bereits auf true gesetzt. Wenn die öffentlichen Subnetze vor dem 26. März 2020 erstellt wurden, müssen Sie die Einstellung manuell ändern. Weitere Informationen finden Sie unter Ändern des öffentlichen `IPv4`-Adressierungsattributs für Ihr Subnetz.
- Wenn Sie eine verwaltete Knotengruppe in privaten Subnetzen bereitstellen, müssen Sie sicherstellen, dass diese auf Amazon ECR zugreifen kann, um Container-Images abzurufen. Sie können dies tun, indem Sie ein NAT-Gateway mit der Routing-Tabelle des Subnetzes verbinden oder indem Sie die folgenden AWS PrivateLink -VPC-Endpunkte hinzufügen:
  - Amazon-ECR-API-Endpunktschnittstelle – `com.amazonaws.`*`region-code`*`.ecr.api`
  - API-Endpunktschnittstelle der Amazon-ECR-Docker-Registrierung – `com.amazonaws.`*`region-code`*`.ecr.dkr`
  - Amazon-S3-Gateway-Endpunkt – `com.amazonaws.`*`region-code`*`.s3`

  Weitere häufig verwendete Services und Endpunkte finden Sie unter Anforderungen an private Cluster.
- Verwaltete Knotengruppen können nicht in AWS Outposts AWS Wavelength oder in AWS Local Zones bereitgestellt werden.

- Sie können mehrere verwaltete Knotengruppen innerhalb eines einzelnen Clusters erstellen. Sie können beispielsweise eine Knotengruppe mit dem standardmäßigen Amazon EKS-optimierten Amazon Linux-AMI für einige Workloads und eine weitere mit der GPU-Variante für Workloads erstellen, die GPU-Unterstützung erfordern.
- Wenn Ihre verwaltete Knotengruppe auf einen Fehler bei der Statusprüfung für die Amazon-EC2-Instance stößt, gibt Amazon EKS einen Fehlercode zurück, der Ihnen bei der Diagnose des Problems hilft. Weitere Informationen finden Sie unter Fehlercodes bei verwalteten Knotengruppen.
- Amazon EKS fügt Kubernetes-Labels zu verwalteten Knotengruppen-Instances hinzu. Diese von Amazon EKS bereitgestellten Labels sind mit dem Präfix `eks.amazonaws.com` versehen.
- Amazon EKS leert Knoten automatisch mit der Kubernetes-API während Beendigungs- oder Aktualisierungsvorgängen.
- Die Budgets für Pod-Disruptionen werden nicht eingehalten, wenn ein Knoten mit `AZRebalance` beendet oder die Anzahl der gewünschten Knoten reduziert wird. Bei diesen Aktionen wird versucht, Pods auf dem Knoten zu bereinigen. Wenn dies jedoch länger als 15 Minuten dauert, wird der Knoten beendet, unabhängig davon, ob alle Pods auf dem Knoten beendet sind. Um den Zeitraum bis zum Beenden des Knotens zu verlängern, fügen Sie der Auto-Scaling-Gruppe einen Lebenszyklus-Hook hinzu. Weitere Informationen finden Sie unter Lebenszyklus-Hooks hinzufügen im Benutzerhandbuch zu Amazon EC2 Auto Scaling.
- Damit der Draining-Prozess nach Erhalt einer Benachrichtigung über eine Spot-Unterbrechung oder eine Kapazitätsumverteilung korrekt ausgeführt werden kann, muss `CapacityRebalance` auf `true` eingestellt sein.
- Aktualisierungen verwalteter Knotengruppen respektieren die Pod-Disruptions-Budgets, die Sie für Ihre Pods festgelegt haben. Weitere Informationen finden Sie unter Verhalten der Aktualisierung verwalteter Knoten.
- Für die Verwendung verwalteter Amazon-EKS-Knotengruppen fallen keine zusätzlichen Kosten an. Sie zahlen nur für die AWS Ressourcen, die Sie bereitstellen.
- Wenn Sie Amazon-EBS-Volumes für Ihre Knoten verschlüsseln möchten, können Sie die Knoten mithilfe einer Startvorlage bereitstellen. Um verwaltete Knoten mit verschlüsselten Amazon-EBS-Volumes ohne Verwendung einer Startvorlage bereitzustellen, verschlüsseln Sie alle neuen Amazon-EBS-Volumes, die in Ihrem Konto erstellt wurden. Weitere Informationen finden Sie unter Standardverschlüsselung im Amazon EC2 EC2-Benutzerhandbuch.

Kapazitätstypen für verwaltete Knotengruppen

Beim Erstellen einer verwalteten Knotengruppe können Sie entweder den Kapazitätstyp On-Demand oder Spot-Kapazität auswählen. Amazon EKS stellt eine verwaltete Knotengruppe mit einer Amazon-EC2-Auto-Scaling-Gruppe bereit, die entweder nur On-Demand- oder nur Amazon-EC2-Spot-Instances enthält. Sie können Pods für fehlertolerante Anwendungen für verwaltete Spot-Knotengruppen und fehlerintolerante Anwendungen für On-Demand-Knotengruppen in einem einzelnen Kubernetes-Cluster planen. Standardmäßig stellt eine verwaltete Knotengruppe Amazon-EC2-Instances auf Abruf bereit.

On-Demand

Mit On-Demand-Instances zahlen Sie für Rechenkapazität bis zur zweiten Stunde ohne langfristige Verpflichtungen.

Funktionsweise

Standardmäßig: Wenn Sie keinen Capacity Type (Kapazitätstyp) festlegen, wird die verwaltete Knotengruppe mit On-Demand-Instances bereitgestellt. Eine verwaltete Knotengruppe konfiguriert in Ihrem Namen eine Amazon-EC2-Auto-Scaling-Gruppe mit den folgenden Einstellungen:

- Die Allokationsstrategie zur Bereitstellung von On-Demand-Kapazität ist auf `prioritized` eingestellt. Verwaltete Knotengruppen verwenden die Reihenfolge der in der API übergebenen Instance-Typen, um zu bestimmen, welcher Instance-Typ bei der Erfüllung der On-Demand-Kapazität zuerst verwendet werden soll. Sie können beispielsweise drei Instance-Typen in der folgenden Reihenfolge angeben: `c5.large`, `c4.large`, und `c3.large`. Wenn Ihre On-Demand-Instances gestartet werden, erfüllt die verwaltete Knotengruppe On-Demand-Kapazität beginnend mit `c5.large`, dann `c4.large`, und dann `c3.large`. Weitere Informationen finden Sie unter Amazon-EC2-Auto-Scaling-Gruppe im Amazon-EC2-Auto-Scaling-Benutzerhandbuch.
- Amazon EKS fügt allen Knoten in Ihrer verwalteten Knotengruppe das folgende Kubernetes-Label hinzu, das den Kapazitätstyp angibt: `eks.amazonaws.com/capacityType: ON_DEMAND`. Sie können dieses Label verwenden, um statusbehaftete oder fehlerintolerante Anwendungen auf On-Demand-Knoten zu planen.

Spot-Instances

Amazon EC2 Spot Instances sind freie Amazon-EC2-Kapazitäten, die hohe Rabatte auf die On-Demand-Preise bieten. Amazon EC2 Spot Instances kann mit einer Vorwarnzeit von zwei Minuten

unterbrochen werden, wenn EC2 die Kapazität wieder benötigt. Weitere Informationen finden Sie unter Spot-Instances im Amazon EC2 EC2-Benutzerhandbuch. Sie können eine verwaltete Knotengruppe mit Amazon EC2 Spot Instances konfigurieren, um die Kosten für die Rechenknoten zu optimieren, die in Ihrem Amazon-EKS-Cluster ausgeführt werden.

Funktionsweise

Um Spot-Instances in einer verwalteten Knotengruppe zu verwenden, müssen Sie eine verwaltete Knotengruppe erstellen, indem Sie den Kapazitätstyp als `spot` einstellen. Eine verwaltete Knotengruppe konfiguriert in Ihrem Namen eine Amazon-EC2-Auto-Scaling-Gruppe mit den folgenden bewährten Spot-Practices:

- Um sicherzustellen, dass Ihre Spot-Knoten in den optimalen Spot-Kapazitätspools bereitgestellt werden, ist die Zuweisungsstrategie auf eine der folgenden Optionen eingestellt:
  - `price-capacity-optimized` (PCO) – Beim Erstellen neuer Knotengruppen in einem Cluster mit Kubernetes-Version `1.28` oder höher ist die Zuweisungsstrategie auf `price-capacity-optimized` eingestellt. Allerdings wird die Zuweisungsstrategie für Knotengruppen nicht geändert, die bereits mit `capacity-optimized` erstellt wurden, bevor von Amazon EKS verwaltete Knotengruppen PCO unterstützt hatten.
  - `capacity-optimized` (CO) – Beim Erstellen neuer Knotengruppen in einem Cluster mit Kubernetes-Version `1.27` oder niedriger ist die Zuweisungsstrategie auf `capacity-optimized` eingestellt.

  Um die Anzahl der Spot-Kapazitätspools zu erhöhen, die für die Zuweisung von Kapazität verfügbar sind, konfigurieren Sie eine verwaltete Knotengruppe für die Verwendung mehrerer Instance-Typen.
- Amazon EC2 Spot Capacity Rebalancing ist aktiviert, sodass Amazon EKS Ihre Spot-Knoten ordnungsgemäß entladen und neu ausgleichen kann, um Anwendungsunterbrechungen zu minimieren, wenn ein Spot-Knoten ein erhöhtes Risiko einer Unterbrechung aufweist. Weitere Informationen dazu finden Sie unter Amazon EC2 Auto Scaling Capacity Rebalancing im Amazon EC2 Auto Scaling User Guide.
  - Wenn ein Spot-Knoten eine Neuausgleichsempfehlung erhält, versucht Amazon EKS automatisch, einen neuen Ersatz-Spot-Knoten zu starten.
  - Wenn ein Spot-Unterbrechungsnachweis in zwei Minuten eintrifft, bevor sich der Ersatz-Spot-Knoten in einem`Ready`, beginnt Amazon EKS mit dem Entleeren des Spot-Knotens, der die Neuausgleichsempfehlung erhalten hat. Amazon EKS entleert den Knoten nach bestem Wissen.

Daher gibt es keine Garantie dafür, dass Amazon EKS wartet, bis der Ersatzknoten dem Cluster beitritt, bevor der vorhandene Knoten entleert wird.

- Wenn ein Ersatz-Spot-Knoten gestartet wird und in dem `Ready`-Zustand auf Kubernetes ist, sperrt Amazon EKS den Spot-Knoten, der die Neuausgleichs-Empfehlung erhalten hat, und leert ihn. Durch das Cordoning des Spot-Knotens wird sichergestellt, dass der Service-Controller keine neuen Anforderungen an diesen Spot-Knoten sendet. Es entfernt sie auch aus der Liste der gesunden, aktiven Spot-Knoten. Durch das Löschen des Spot-Knotens wird sichergestellt, dass laufende Pods anmutig entfernt werden.

- Amazon EKS fügt allen Knoten in Ihrer verwalteten Knotengruppe das folgende Kubernetes-Label hinzu, das den Kapazitätstyp angibt: `eks.amazonaws.com/capacityType: SPOT`. Sie können dieses Label verwenden, um fehlertolerante Anwendungen auf Spot-Knoten zu planen.

Überlegungen zur Auswahl eines Kapazitätstyps

Bei der Entscheidung, ob eine Knotengruppe mit On-Demand- oder Spot-Kapazität bereitgestellt werden soll, sollten Sie die folgenden Bedingungen berücksichtigen:

- Spot-Instances eignen sich gut für zustandslose, fehlertolerante, flexible Anwendungen. Dazu gehören Workloads für Batch- und Machine Learning-Trainings, Big-Data-ETLs wie Apache Spark, Warteschlangen-Verarbeitungsanwendungen und zustandslose API-Endpunkte. Da es sich bei Spot um eine Ersatzkapazität von Amazon EC2 handelt, die sich im Laufe der Zeit ändern kann, wird empfohlen, Spot-Kapazität für unterbrechungstolerante Workloads zu verwenden. Insbesondere eignet sich die Spot-Kapazität für Workloads, die Zeiten tolerieren können, in denen die erforderliche Kapazität nicht verfügbar ist.
- Es wird empfohlen, On-Demand für Anwendungen zu verwenden, die fehlerintolerant sind. Dazu gehören Cluster-Verwaltungs-Tools wie Überwachungs- und Betriebstools, Bereitstellungen, die `StatefulSets` erfordern, und zustandsbehaftete Anwendungen wie Datenbanken.
- Um die Verfügbarkeit Ihrer Anwendungen bei der Verwendung von Spot-Instances zu maximieren, wird empfohlen, eine verwaltete Spot-Knotengruppe für die Verwendung mehrerer Instance-Typen zu konfigurieren. Es wird empfohlen, die folgenden Regeln anzuwenden, wenn mehrere Instance-Typen verwendet werden:
  - Wenn Sie innerhalb einer verwalteten Knotengruppe den Cluster Autoscaler verwenden, wird empfohlen, einen flexiblen Satz von Instance-Typen mit der gleichen Menge an vCPU und Arbeitsspeicherressourcen zu verwenden. Dies soll sicherstellen, dass die Knoten in Ihrem Cluster wie erwartet skalieren. Wenn Sie beispielsweise vier vCPUs und acht GiB Speicher

benötigen, verwenden Sie `c3.xlarge`, `c4.xlarge`, `c5.xlarge`, `c5d.xlarge`, `c5a.xlarge`, `c5n.xlarge` oder andere ähnliche Instance-Typen.

- Um die Anwendungsverfügbarkeit zu verbessern, empfehlen wir Ihnen, mehrere Spot-verwaltete Knotengruppen bereitzustellen. Dafür sollte jede Gruppe einen flexiblen Satz von Instance-Typen verwenden, die über die gleichen vCPU- und Speicherressourcen verfügen. Wenn Sie beispielsweise 4 vCPUs und 8 GiB Speicher benötigen, empfehlen wir Ihnen, eine verwaltete Knotengruppe mit `c3.xlarge`, `c4.xlarge`, `c5.xlarge`, `c5d.xlarge`, `c5a.xlarge`, `c5n.xlarge` oder andere ähnliche Instance-Typen und eine zweite verwaltete Knotengruppe mit `m3.xlarge`, `m4.xlarge`, `m5.xlarge`, `m5d.xlarge`, `m5a.xlarge`, `m5n.xlarge` oder andere ähnliche Instance-Typen zu erstellen.
- Wenn Sie Ihre Knotengruppe mit dem Spot-Kapazitätstyp bereitstellen, der eine benutzerdefinierte Startvorlage verwendet, verwenden Sie die API, um mehrere Instance-Typen zu übergeben. Übergeben Sie keinen einzelnen Instance-Typ durch die Startvorlage. Weitere Informationen zum Bereitstellen einer Knotengruppe mithilfe einer Startvorlage finden Sie unterAnpassen verwalteter Knoten mit Startvorlagenaus.

Erstellen einer verwalteten Knotengruppe

Dieses Thema beschreibt, wie Sie von Amazon EKS verwaltete Knotengruppen von Knoten starten können, die sich bei Ihrem Amazon-EKS-Cluster registrieren. Nachdem die Knoten dem Cluster beigetreten sind, können Sie Kubernetes-Anwendungen darin bereitstellen.

Wenn Sie zum ersten Mal eine von Amazon EKS verwaltete Knotengruppe starten, empfehlen wir Ihnen, stattdessen eine unserer Erste Schritte mit Amazon EKS Anleitungen zu befolgen. Die Leitfäden bieten ausführliche Erklärungen zur Erstellung eines Amazon-EKS-Clusters mit Knoten.

> **Important**
>
> - Amazon-EKS-Knoten sind Standard-Amazon-EC2-Instances. Sie werden auf der Grundlage der normalen Amazon-EC2-Preise in Rechnung gestellt. Weitere Informationen dazu finden Sie unter Amazon EC2 – Preise.
> - Sie können keine verwalteten Knoten in einem Land erstellen AWS Outposts AWS Wavelength, in AWS-Region dem Sie AWS Local Zones aktiviert haben. Sie können selbstverwaltete Knoten in einer Umgebung erstellen, AWS-Region in der Sie Local Zones aktiviert haben. Weitere Informationen finden Sie unter Starten selbstverwalteter Amazon Linux-Knoten, Starten selbstverwalteter Windows-Knoten und Starten

selbstverwalteter Bottlerocket-Knoten. Sie können auch eine selbstverwaltete Amazon-Linux-Knotengruppe auf einem Outpost erstellen. Weitere Informationen finden Sie unter Starten selbstverwalteter Amazon Linux-Knoten auf einem Outpost.

- Wenn Sie keine AMI-ID für die `bootstrap.sh`-Datei angeben, die in Amazon-EKS-optimiertem Linux oder Bottlerocket enthalten ist, erzwingen verwaltete Knotengruppen eine maximale Anzahl für den Wert von`maxPods`. Für Instances mit weniger als 30 vCPUs beträgt die maximale Anzahl `110`. Für Instances mit mehr als 30 vCPUs wird die maximale Anzahl auf `250` erhöht. Diese Zahlen basieren auf Kubernetes-Skalierbarkeitsschwellenwerten und empfohlenen Einstellungen, die nach internen Tests des Amazon-EKS-Skalierbarkeitsteams empfohlen werden. Weitere Informationen finden Sie im Blogbeitrag Amazon VPC CNI plugin increases pods per node limits (VPC-CNI-Plugin von Amazon erhöht Grenzwerte für Pods pro Knoten).

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erstellen eines Amazon-EKS-Clusters.
- Eine vorhandene IAM-Rolle, die von den Knoten verwendet werden soll. Informationen zum Erstellen finden Sie unter Amazon-EKS-Knoten-IAM-Rolle. Wenn diese Rolle keine der Richtlinien für das VPC-CNI hat, ist die folgende separate Rolle für die VPC-CNI-Pods erforderlich.
- (Optional, aber empfohlen) Das Amazon VPC CNI plugin for Kubernetes-Add-on wurde mit einer eigenen IAM-Rolle konfiguriert, an die die erforderliche IAM-Richtlinie angehängt ist. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).
- Kenntnisse über die in Auswählen eines Amazon-EC2-Instance-Typs aufgelisteten Überlegungen. Je nachdem, welchen Instance-Typ Sie wählen, kann es zusätzliche Voraussetzungen für Ihren Cluster und Ihre VPC geben.
- Um eine verwaltete Windows.Knotengruppe hinzuzufügen, müssen Sie zuerst die Windows-Unterstützung für Ihren Cluster aktivieren. Weitere Informationen finden Sie unter Die Windows-Unterstützung für Ihre Amazon-EKS-Cluster aktivieren.

Sie können eine verwaltete Knotengruppe mit `eksctl` oder AWS Management Console erstellen.

eksctl

Erstellen einer verwalteten Knotengruppe mit **`eksctl`**

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre -Version mit dem folgenden Befehl überprüfen:

```
eksctl version
```

Eine Installations- und Upgrade-Anleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

1. (Optional) Wenn die von verwaltete AmazonEKS_CNI_Policy IAM-Richtlinie an Ihre Amazon-EKS-Knoten-IAM-Rolle angefügt ist, sollten Sie sie stattdessen einer IAM-Rolle zuzuweisen, die Sie dem `aws-node`-Servicekonto für Kubernetes zuordnen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).
2. Erstellen Sie eine verwaltete Knotengruppe mit oder ohne Verwendung einer benutzerdefinierten Startvorlage. Das manuelle Angeben einer Startvorlage ermöglicht eine bessere Anpassung einer Knotengruppe. Zum Beispiel kann es die Bereitstellung eines benutzerdefinierten AMI oder die Bereitstellung von Argumenten für das `boostrap.sh`-Skript in einem für Amazon EKS optimierten AMI ermöglichen. Geben Sie den folgenden Befehl ein, um eine vollständige Liste aller verfügbaren Optionen und Standardwerte anzuzeigen.

   ```
   eksctl create nodegroup --help
   ```

   Ersetzen Sie im folgenden Befehl *`my-cluster`* durch den Namen Ihres Clusters und ersetzen Sie *`my-mng`* durch den Namen Ihrer Knotengruppe. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten.

   > **Important**
   >
   > Wenn Sie beim erstmaligen Erstellen einer verwalteten Knotengruppe keine benutzerdefinierte Startvorlage verwenden, sollten Sie später auch keine für die Knotengruppe verwenden. Wenn Sie keine benutzerdefinierte Startvorlage angegeben haben, generiert das System automatisch eine Startvorlage, die Sie

nicht manuell ändern können. Das manuelle Ändern dieser automatisch generierten Startvorlage kann zu Fehlern führen.

Ohne Startvorlage

`eksctl` erstellt eine standardmäßige Amazon-EC2-Startvorlage in Ihrem Konto und stellt die Knotengruppe mithilfe einer Startvorlage bereit, die basierend auf den von Ihnen angegebenen Optionen erstellt wird. Bevor Sie einen Wert für `--node-type` festlegen, siehe Auswählen eines Amazon-EC2-Instance-Typs.

Ersetzen Sie *`ami-family`* durch ein zulässiges Schlüsselwort. Weitere Informationen finden Sie unter Einrichtung der Knoten-AMI-Familie in der `eksctl`-Dokumentation. Ersetzen Sie *`my-key`* mit dem Namen Ihres Amazon-EC2-Schlüsselpaars oder öffentlichen Schlüssels. Dieser Schlüssel wird für den SSH-Zugriff zu Ihren Knoten verwendet, nachdem diese gestartet wurden.

> **Note**
>
> Für Windows aktiviert dieser Befehl SSH nicht. Stattdessen wird das Amazon-EC2-Schlüsselpaar mit der Instance verknüpft. Außerdem ermöglicht dies eine RDP-Verbindung mit der Instance.

Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Linux Informationen finden Sie unter Amazon EC2 EC2-Schlüsselpaare und Linux -Instances im Amazon EC2 EC2-Benutzerhandbuch. Weitere Windows Informationen finden Sie unter Amazon EC2 EC2-Schlüsselpaare und Windows -Instances im Amazon EC2 EC2-Benutzerhandbuch.

Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

- Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
- Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Wenn Sie den Pod-Zugriff auf IMDS blockieren möchten, fügen Sie dann **`--disable-pod-imds`** auf den folgenden Befehl.

```
eksctl create nodegroup \
  --cluster my-cluster \
  --region region-code \
  --name my-mng \
  --node-ami-family ami-family \
  --node-type m5.large \
  --nodes 3 \
  --nodes-min 2 \
  --nodes-max 4 \
  --ssh-access \
  --ssh-public-key my-key
```

Ihre Instances können Pods optional eine deutlich höhere Anzahl von IP-Adressen zuweisen, Pods aus einem anderen CIDR-Block als der Instance IP-Adressen zuweisen und in einem Cluster ohne Internetzugang bereitgestellt werden. Weitere Informationen dazu finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten, Benutzerdefinierte Netzwerke für Pods und Anforderungen an private Cluster für weitere Optionen, die dem vorherigen Befehl hinzugefügt werden.

Verwaltete Knotengruppen berechnen und wenden einen einzelnen Wert für die maximale Anzahl von Pods an, die auf jedem Knoten Ihrer Knotengruppe ausgeführt werden können, basierend auf Instance-Typ. Wenn Sie eine Knotengruppe mit unterschiedlichen Instance-Typen erstellen, wird der kleinste Wert, der für alle Instance-Typen berechnet wird, als die maximale Anzahl von Pods angewendet, die für jeden Instance-Typ in der Knotengruppe ausgeführt werden können. Verwaltete Knotengruppen berechnen den Wert anhand des Skripts, auf das in Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ verwiesen wird.

Mit einer Startvorlage

Die Startvorlage muss bereits vorhanden sein und muss die Anforderungen erfüllen, die in Grundlagen der Konfiguration der Vorlage starten festgelegt sind.

Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

- Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
- Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Wenn Sie den Pod-Zugriff auf IMDS blockieren möchten, geben Sie die erforderlichen Einstellungen in der Startvorlage an.

a. Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie die *example values* und führen Sie dann den geänderten Befehl aus, um die `eks-nodegroup.yaml`-Datei zu erstellen. Mehrere Einstellungen, die Sie bei der Bereitstellung ohne Startvorlage angeben, werden in die Startvorlage verschoben. Wenn Sie keine `version` angeben, wird die Standardversion verwendet.

```
cat >eks-nodegroup.yaml <<EOF
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: my-cluster
  region: region-code
managedNodeGroups:
- name: my-mng
  launchTemplate:
    id: lt-id
    version: "1"
EOF
```

Eine vollständige Liste der `eksctl` Konfigurationsdateieinstellungen finden Sie unter Konfigurationsdateischema in der `eksctl` Dokumentation. Ihre Instances können optional eine deutlich höhere Anzahl von IP-Adressen an Pods zuweisen, IP-Adressen an Pods aus einem anderen CIDR-Block als dem der Instance zuweisen, die `containerd`-Laufzeit verwenden und in einem Cluster ohne ausgehenden Internetzugang bereitgestellt werden. Weitere Informationen dazu finden Sie unter

Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten, Benutzerdefinierte Netzwerke für Pods, Testen Sie die Migration von Docker nach `containerd` und Anforderungen an private Cluster für zusätzliche Optionen, die der Konfigurationsdatei hinzugefügt werden.

Wenn Sie in Ihrer Startvorlage keine AMI-ID angegeben haben, berechnet verwaltete Knotengruppen einen einzelnen Wert für die maximale Anzahl von Pods, die auf jedem Knoten Ihrer Knotengruppe ausgeführt werden können, basierend auf Instance-Typ. Wenn Sie eine Knotengruppe mit unterschiedlichen Instance-Typen erstellen, wird der kleinste Wert, der für alle Instance-Typen berechnet wird, als die maximale Anzahl von Pods angewendet, die für jeden Instance-Typ in der Knotengruppe ausgeführt werden können. Verwaltete Knotengruppen berechnen den Wert anhand des Skripts, auf das in Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ verwiesen wird.

Wenn Sie eine AMI-ID in Ihrer Startvorlage angegeben haben, geben Sie die maximale Anzahl von Pods an, die auf jedem Knoten Ihrer Knotengruppe ausgeführt werden können, wenn Sie Benutzerdefinierte Netzwerke verwenden oder die Anzahl der Ihrer Instance zugewiesenen IP-Adressen erhöhen möchten. Weitere Informationen finden Sie unter Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ.

b. Stellen Sie die Knotengruppe mit dem folgenden Befehl bereit.

```
eksctl create nodegroup --config-file eks-nodegroup.yaml
```

AWS Management Console

Um eine verwaltete Knotengruppe zu erstellen, verwenden Sie AWS Management Console

1. Warten Sie, bis der Status des Clusters als `ACTIVE` angezeigt wird. Sie können keine verwaltete Knotengruppe für einen Cluster erstellen, der noch nicht `ACTIVE` ist.
2. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
3. Wählen Sie den Namen des Clusters aus, in dem Sie eine verwaltete Knotengruppe erstellen möchten.
4. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.

5. Wählen Sie Add Node Group (Knotengruppe hinzufügen) aus.
6. Geben Sie auf der Seite Configure node group (Knotengruppe konfigurieren) die Parameter entsprechend aus, und wählen Sie dann Next (Weiter).

   - Name – Geben Sie einen eindeutigen Namen für die verwaltete Knotengruppe ein. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten.
   - Node IAM role (Knoten-IAM-Rolle) – Wählen Sie die Knoten-Instance-Rolle aus, die mit Ihrer Knotengruppe verwendet werden soll. Weitere Informationen finden Sie unter Amazon-EKS-Knoten-IAM-Rolle.

     > **Important**
     >
     > - Sie können nicht dieselbe Rolle verwenden, die zum Erstellen von Clustern verwendet wurde.
     > - Es wird empfohlen, eine Rolle zu verwenden, die derzeit nicht von einer selbstverwalteten Knotengruppe genutzt wird. Andernfalls planen Sie die Verwendung mit einer neuen selbstverwalteten Knotengruppe. Weitere Informationen finden Sie unter Löschen einer verwalteten Knotengruppe.

   - Startvorlage verwenden – (Optional) Wählen Sie aus, ob Sie eine bestehende Startvorlage verwenden möchten. Wählen Sie einen Namen für die Startvorlage aus. Wählen Sie dann eine Launch template version (Startvorlagenversion) aus. Wenn Sie keine Version auswählen, verwendet Amazon EKS die Standardversion der Vorlage. Startvorlagen ermöglichen eine stärkere Anpassung Ihrer Knotengruppe, z. B. die Bereitstellung eines benutzerdefinierten AMI, die Zuweisung einer deutlich höheren Anzahl von IP-Adressen an Pods, die Zuweisung von IP-Adressen an Pods aus einem anderen CIDR-Block als dem der Instance, die Aktivierung der `containerd` Laufzeit für Ihre Instances und die Bereitstellung von Knoten in einem Cluster ohne ausgehenden Internetzugang. Weitere Informationen dazu finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten, Benutzerdefinierte Netzwerke für Pods, Testen Sie die Migration von Docker nach `containerd` und Anforderungen an private Cluster.

     Die Startvorlage muss die Anforderungen in Anpassen verwalteter Knoten mit Startvorlagen erfüllen. Wenn Sie keine eigene Startvorlage verwenden, erstellt die Amazon EKS API eine standardmäßige Amazon-EC2-Startvorlage in Ihrem Konto und stellt die Knotengruppe mithilfe der Standardstartvorlage bereit.

Wenn Sie IAM-Rollen für Servicekonten implementieren, weisen Sie die erforderlichen Berechtigungen direkt jedem Pod zu, der Zugriff auf AWS -Services benötigt. Wenn keine Pods im Cluster aus anderen Gründen Zugriff auf IMDS benötigen, z. B. für das Abrufen der aktuellen AWS-Region, können Sie den Zugriff auf IMDS auch für Pods deaktivieren, bei denen kein Hostnetzwerk in einer Startvorlage verwendet wird. Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

- Kubernetes-Labels – (Optional) Sie können Kubernetes-Labels auf die Knoten in Ihrer verwalteten Knotengruppe anwenden.
- Kubernetes-Taints – (Optional) Sie können Kubernetes-Taints auf die Knoten in Ihrer verwalteten Knotengruppe anwenden. Die verfügbaren Optionen im Menü Effect (Effekt) sind **`NoSchedule`**, **`NoExecute`** und **`PreferNoSchedule`**. Weitere Informationen finden Sie unter Knotenfehler auf verwalteten Knotengruppen.
- Tags – (Optional) Sie können wählen, ob Sie Ihre mit Amazon EKS verwaltete Knotengruppe taggen möchten. Diese Markierungen werden nicht an andere Ressourcen in der Knotengruppe, z. B. Auto-Scaling-Gruppen oder Instances, weitergegeben. Weitere Informationen finden Sie unter Kennzeichnen Ihrer Amazon EKS-Ressourcen .

7. Geben Sie auf der Seite Set compute configuration (Datenverarbeitung-Konfiguration einrichten) die Parameter entsprechend ein, und wählen Sie dann Next (Weiter).

   - AMI type (AMI-Typ) – Wählen Sie einen AMI-Typ aus. Wenn Sie Arm-Instances bereitstellen, lesen Sie vor der Bereitstellung die Überlegungen in Amazon-EKS-optimierte Arm-Amazon-Linux-AMIs.

     Wenn Sie auf der vorherigen Seite eine Startvorlage angegeben haben und in der Startvorlage ein AMI angegeben haben, können Sie keinen Wert auswählen. Der Wert aus der Vorlage wird angezeigt. Die in der Vorlage angegebene AMI muss die Anforderungen unter Angeben eines AMI erfüllen.
   - Capacity type (Kapazitätstyp) — Wählen Sie einen Kapazitätstyp aus. Weitere Informationen zur Auswahl eines Kapazitätstyps finden Sie unter Kapazitätstypen für verwaltete Knotengruppen. Es ist nicht möglich, verschiedene Kapazitätstypen innerhalb derselben Knotengruppe zu mischen. Wenn Sie beide Kapazitätstypen verwenden möchten, erstellen Sie separate Knotengruppen mit jeweils eigenen Kapazitäts- und Instance-Typen.

- Instance-Typen- Standardmäßig wird ein oder mehrere Instance-Typen angegeben. Um einen Standard-Instance-Typ zu entfernen, wählen Sie X auf der rechten Seite des Instance-Typs. Wählen Sie den Instance-Typ aus, der in der verwalteten Knotengruppe verwendet werden soll. Weitere Informationen finden Sie unter Auswählen eines Amazon-EC2-Instance-Typs.

  Die Konsole zeigt eine Reihe von häufig verwendeten Instance-Typen an. Wenn Sie eine verwaltete Knotengruppe mit einem Instance-Typ erstellen müssen, der nicht angezeigt wird, verwenden Sie `eksctl`, AWS CLI, AWS CloudFormation oder ein SDK, um die Knotengruppe zu erstellen. Wenn Sie auf der vorherigen Seite eine Startvorlage angegeben haben, können Sie keinen Wert auswählen, da der Instance-Typ in der Startvorlage angegeben werden muss. Der Wert aus der Startvorlage wird angezeigt. Wenn Sie die Option Spot für Capacity type (Kapazitätstyp) ausgewählt haben, empfehlen wir Ihnen, mehrere Instance-Typen anzugeben, um die Verfügbarkeit zu verbessern.
- Disk size (Datenträgergröße) – Geben Sie die Datenträgergröße (in GiB) ein, die für das Stamm-Volume des Knotens verwendet werden soll.

  Wenn Sie auf der vorherigen Seite eine Startvorlage angegeben haben, können Sie keinen Wert auswählen, da dieser in der Startvorlage angegeben werden muss.
- Desired size (Gewünschte Größe) – Geben Sie die aktuelle Anzahl von Knoten an, die die verwaltete Knotengruppe beim Start beibehalten soll.

  > **Note**
  >
  > Amazon EKS skaliert Ihre Knotengruppe nicht automatisch. Sie können jedoch den Kubernetes Cluster Autoscaler so konfigurieren, dass er dies für Sie übernimmt.
- Minimum size (Mindestgröße) – Geben Sie die Mindestanzahl von Knoten an, auf die die verwaltete Knotengruppe skaliert werden kann.
- Maximum size (Maximale Größe) – Geben Sie die maximale Anzahl von Knoten an, auf die die verwaltete Knotengruppe skaliert werden kann.
- Konfiguration der Knotengruppe aktualisieren— (Optional) Sie können die Anzahl oder den Prozentsatz der Knoten auswählen, die parallel aktualisiert werden sollen. Diese Knoten werden während der Aktualisierung nicht verfügbar sein. FürMaximal nicht verfügbarWählen Sie eine der folgenden Optionen und geben Sie einenValue:

- Zahl – Wählen Sie die Anzahl der Knoten in Ihrer Knotengruppe, die parallel aktualisiert werden können, und geben Sie diese an.
- Prozentsatz – Wählen und geben Sie den Prozentsatz der Knoten in der Knotengruppe an, die parallel aktualisiert werden können. Dies ist nützlich, wenn Sie eine große Anzahl von Knoten in Ihrer Knotengruppe haben.

8. Füllen Sie auf der Seite Specify Details (Details angeben) die Parameter entsprechend aus, und klicken Sie dann auf Next.

- Subnets (Subnetze) – Wählen Sie die Subnetze aus, in die die verwalteten Knoten gestartet werden sollen.

> **Important**
>
> Wenn Sie eine zustandsbehaftete Anwendung über mehrere Availability Zones hinweg ausführen, die von Amazon-EBS-Volumes gesichert wird und den Kubernetes Auto Scaling verwendet, sollten Sie mehrere Knotengruppen konfigurieren, die jeweils für eine einzelne Availability Zone gelten. Außerdem sollten Sie das `--balance-similar-node-groups`-Feature aktivieren.

> **Important**
>
> - Wenn Sie ein öffentliches Subnetz wählen und in Ihrem Cluster nur der öffentliche API-Server-Endpunkt aktiviert ist, muss das Subnetz `MapPublicIPOnLaunch` auf `true` eingestellt sein, damit die Instances erfolgreich einem Cluster zugeordnet werden können. Wenn das Subnetz mit `eksctl` oder den von Amazon EKS vertriebenen AWS CloudFormation - Vorlagen am oder nach dem 26. März 2020 erstellt wurde, ist diese Einstellung bereits auf `true` gesetzt. Wenn die Subnetze `eksctl` oder die AWS CloudFormation Vorlagen vor dem 26. März 2020 erstellt wurden, müssen Sie die Einstellung manuell ändern. Weitere Informationen finden Sie unter Ändern des öffentlichen `IPv4`-Adressierungsattributs für Ihr Subnetz.
> - Wenn Sie eine Startvorlage verwenden und mehrere Netzwerkschnittstellen angeben, weist Amazon EC2 keine öffentliche `IPv4`-Adresse automatisch zu, selbst wenn `MapPublicIpOnLaunch` auf `true` gesetzt wird. Damit Knoten dem Cluster in diesem Szenario beitreten können, müssen Sie entweder den privaten

API-Serverendpunkt des Clusters aktivieren oder Knoten in einem privaten Subnetz mit ausgehendem Internetzugriff starten, der über eine alternative Methode wie ein NAT-Gateway bereitgestellt wird. Weitere Informationen finden Sie unter Amazon EC2 EC2-Instance-IP-Adressierung im Amazon EC2 EC2-Benutzerhandbuch.

- Konfigurieren des SSH-Zugriffs auf Knoten (Optional). Mit der Aktivierung von SSH können Sie eine Verbindung zu Ihren Instances herstellen und Diagnoseinformationen erfassen, wenn Probleme auftreten. Wir empfehlen dringend, den Fernzugriff zu aktivieren, sobald Sie eine Knotengruppe erstellen. Sie können den Fernzugriff nicht mehr aktivieren, nachdem die Knotengruppe erstellt wurde.

  Wenn Sie eine Startvorlage verwenden möchten, wird diese Option nicht angezeigt. Um den Remotezugriff auf Ihre Knoten zu aktivieren, geben Sie in der Startvorlage ein Schlüsselpaar an und stellen Sie sicher, dass die richtige Schnittstelle für die Knoten in den Sicherheitsgruppen geöffnet ist, die Sie in der Startvorlage angeben. Weitere Informationen finden Sie unter Benutzerdefinierte Sicherheitsgruppen.

  > **Note**
  >
  > Für Windows aktiviert dieser Befehl SSH nicht. Stattdessen wird das Amazon-EC2-Schlüsselpaar mit der Instance verknüpft. Außerdem ermöglicht dies eine RDP-Verbindung mit der Instance.

- Für SSH-Schlüsselpaar (optional) wählen Sie einen Amazon-EC2-SSH-Schlüssel aus, der verwendet werden soll. Weitere Linux Informationen finden Sie unter Amazon EC2 EC2-Schlüsselpaare und Linux -Instances im Amazon EC2 EC2-Benutzerhandbuch. Weitere Windows Informationen finden Sie unter Amazon EC2 EC2-Schlüsselpaare und Windows -Instances im Amazon EC2 EC2-Benutzerhandbuch. Wenn Sie eine Startvorlage verwenden möchten, können Sie keine auswählen. Wenn ein Amazon-EC2-SSH-Schlüssel für Knotengruppen bereitgestellt wird, die Bottlerocket-AMIs verwenden, wird auch der administrative Container aktiviert. Weitere Informationen finden Sie unter Administrator-Container auf GitHub.
- Für Zulassen des SSH-Remote-Zugriffs von, wenn Sie den Zugriff auf bestimmte Instances beschränken möchten, wählen Sie die Sicherheitsgruppen aus, die diesen Instances zugeordnet sind. Wenn Sie keine bestimmten Sicherheitsgruppen auswählen, ist SSH-Zugriff von überall im Internet erlaubt (`0.0.0.0/0`).

9. Überprüfen Sie auf der Seite Review and create (Überprüfen und Erstellen) die Konfiguration der verwalteten Knoten, und wählen Sie Create (Erstellen).

   Wenn Knoten dem Cluster nicht beitreten können, finden Sie weitere Informationen unter Knoten können nicht mit dem Cluster verknüpft werden im Handbuch zur Fehlerbehebung.

10. Sehen Sie sich den Status Ihrer Knoten an und warten Sie, bis diese in den `Ready`-Status eintreten.

    ```
    kubectl get nodes --watch
    ```

11. (Nur GPU-Knoten) Wenn Sie einen GPU-Instance-Typ und das mit Amazon EKS optimierte beschleunigte AMI gewählt haben, müssen Sie das NVIDIA-Geräte-Plugin für Kubernetes mit dem folgenden Befehl als DaemonSet auf Ihren Cluster anwenden. Ersetzen Sie *vX.X.X* mit Ihrer gewünschten Nvidia/K8S-Geräte-Plugin-Version, bevor Sie den folgenden Befehl ausführen.

    ```
    kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
    ```

Nachdem Sie nun über einen funktionierenden Amazon-EKS-Cluster mit Worker-Knoten verfügen, können Sie mit der Installation von Kubernetes-Add-ons und -Anwendungen auf Ihrem Cluster beginnen. Die folgenden Dokumentationsthemen helfen Ihnen bei der Erweiterung der Funktionalität Ihres Clusters.

- Der IAM-Prinzipal, der den Cluster erstellt hat, ist der einzige Prinzipal, der Aufrufe an den Kubernetes-API-Server mit `kubectl` oder AWS Management Console tätigen kann. Wenn Sie möchten, dass andere IAM-Prinzipale Zugriff auf Ihren Cluster haben, müssen Sie sie hinzufügen. Weitere Informationen finden Sie unter Zugriff auf Kubernetes APIs gewähren und Erforderliche Berechtigungen.
- Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:
  - Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
  - Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

- Auto Scaling – Konfigurieren Sie den Kubernetes Cluster Autoscaler, um die Anzahl der Knoten in Ihren Knotengruppen automatisch anzupassen.
- Stellen Sie eine Beispielanwendung für Ihr Cluster bereit.
- Clusterverwaltung – Erfahren Sie, wie Sie wichtige Werkzeuge für die Verwaltung Ihres Clusters verwenden.

Aktualisieren einer verwalteten Knotengruppe

Wenn Sie eine Aktualisierung der verwalteten Knotengruppe initiieren, aktualisiert Amazon EKS Ihre Knoten automatisch für Sie. Führen Sie die Schritte aus, die unter Verhalten der Aktualisierung verwalteter Knoten aufgelistet sind. Wenn Sie ein für Amazon EKS optimiertes AMI verwenden, wendet Amazon EKS automatisch die neuesten Sicherheits-Patches und Betriebssystem-Aktualisierungen auf Ihre Knoten als Teil der neuesten AMI-Versionsversion an.

Es gibt mehrere Szenarien, in denen Sie die Version oder Konfiguration Ihrer verwalteten Amazon-EKS-Knotengruppe aktualisieren:

- Sie haben die Kubernetes-Version für Ihren Amazon-EKS-Cluster aktualisiert und möchten Ihre Arbeitsknoten so aktualisieren, dass sie dieselbe Kubernetes-Version verwenden.
- Eine neue AMI-Version ist für Ihre verwaltete Knotengruppe verfügbar. Weitere Informationen zu AMI-Versionen finden Sie unter diesen Abschnitten:
  - Amazon-EKS-optimierte Amazon-Linux-AMI-Versionen
  - Amazon-EKS-optimierte Bottlerocket-AMIs
  - Amazon ECS-optimierte Windows-AMI-Versionen
- Sie möchten die minimale, maximale oder gewünschte Anzahl der Instances in Ihrer verwalteten Knotengruppe anpassen.
- Sie möchten Kubernetes-Labels hinzufügen oder aus den Instances in Ihrer verwalteten Knotengruppe entfernen.
- Sie möchten Ihrer verwalteten Knotengruppe AWS Tags hinzufügen oder daraus entfernen.
- Sie müssen eine neue Version einer Startvorlage mit Konfigurationsänderungen bereitstellen, z. B. ein aktualisiertes benutzerdefiniertes AMI.

- Sie haben Version `1.9.0` oder höher des Amazon VPC CNI-Add-ons bereitgestellt, das Add-on für die Präfix-Delegierung aktiviert und möchten, dass neue AWS Nitro System Instances in einer Knotengruppe eine deutlich höhere Anzahl von unterstützen. Pods Weitere Informationen finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten.
- Sie haben die IP-Präfix-Delegierung für Windows-Knoten aktiviert und möchten, dass neue AWS Nitro System-Instances in einer Knotengruppe eine deutlich höhere Anzahl von unterstützen. Pods Weitere Informationen finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten.

Wenn es eine neuere AMI-Version für die Kubernetes-Version Ihrer verwalteten Knotengruppe gibt als die, die Ihre Knotengruppe derzeit ausführt, können Sie sie aktualisieren, um diese neue AMI-Version zu verwenden. Wenn Ihr Cluster eine neuere Kubernetes-Version als Ihre Knotengruppe ausführt, können Sie die Knotengruppe so aktualisieren, dass sie die neueste AMI-Version verwendet, die der Kubernetes-Version Ihres Clusters entspricht.

Wenn ein Knoten in einer verwalteten Knotengruppe aufgrund eines Skalierungsvorgangs oder Aktualisierung beendet wird, werden die Pods in diesem Knoten zuerst geleert. Weitere Informationen finden Sie unter Verhalten der Aktualisierung verwalteter Knoten.

Aktualisieren einer Knotengruppenversion

Sie können eine Knotengruppenversion mit `eksctl` oder AWS Management Console aktualisieren. Die Version, auf die Sie aktualisieren, kann nicht neuer als die Version der Steuerebene sein.

eksctl

So aktualisieren Sie eine Knotengruppenversion mit **`eksctl`**

- Aktualisieren Sie eine verwaltete Knotengruppe mit dem folgenden Befehl auf die neueste AMI-Version derselben Kubernetes-Version, die derzeit auf den Arbeitsknoten bereitgestellt wird. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte.

```
eksctl upgrade nodegroup \
  --name=node-group-name \
  --cluster=my-cluster \
  --region=region-code
```

> **Note**
>
> Wenn Sie eine Knotengruppe, die mit einer Startvorlage bereitgestellt wurde, auf eine neue Version der Startvorlage aktualisieren, fügen Sie `--launch-template-version` *`version-number`* dem vorangehenden Befehl hinzu. Die Startvorlage muss die unter Anpassen verwalteter Knoten mit Startvorlagen beschriebenen Anforderungen erfüllen. Wenn die Startvorlage ein benutzerdefiniertes AMI enthält, muss das AMI die Anforderungen in Angeben eines AMI erfüllen. Wenn Sie Ihre Knotengruppe auf eine neuere Version Ihrer Startvorlage aktualisieren, wird jeder Knoten wiederverwertet, um mit der neuen Konfiguration der angegebenen Startvorlagenversion übereinzustimmen.
> Sie können eine Knotengruppe, die ohne eine Startvorlage bereitgestellt wird, nicht direkt auf eine neue Startvorlagenversion aktualisieren. Stattdessen müssen Sie mithilfe der Startvorlage eine neue Knotengruppe bereitstellen, um die Knotengruppe auf eine neue Version der Startvorlage zu aktualisieren.

Sie können eine Knotengruppe auf dieselbe Version aktualisieren wie die Kubernetes-Version der Steuerebene. Wenn Sie beispielsweise über einen Cluster mit Kubernetes `1.29` verfügen, können Sie Knoten, auf denen derzeit Kubernetes Kubernetes `1.28` ausgeführt wird, mit dem folgenden Befehl auf Version `1.29` aktualisieren.

```
eksctl upgrade nodegroup \
  --name=node-group-name \
  --cluster=my-cluster \
  --region=region-code \
  --kubernetes-version=1.29
```

AWS Management Console

Um eine Knotengruppenversion mit dem zu aktualisieren AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Cluster aus, der die zu aktualisierende Knotengruppe enthält.

3. Wenn mindestens eine Knotengruppe über ein verfügbares Update verfügt, wird oben auf der Seite ein Feld angezeigt, in dem Sie über das verfügbare Update informiert werden. Wenn Sie die Registerkarte Compute (Datenverarbeitung) wählen, finden Sie den Eintrag Update now (Jetzt aktualisieren) in der Spalte AMI release version (AMI-Vorversion) in der Tabelle Node groups (Knotengruppen) für die Knotengruppe, für die eine Aktualisierung verfügbar ist. Um die Knotengruppe zu aktualisieren, wählen Sie Update now (Jetzt aktualisieren).

   Es wird keine Benachrichtigung für Knotengruppen angezeigt, die mit einem benutzerdefinierten AMI bereitgestellt wurden. Wenn Ihre Knoten mit einem benutzerdefinierten AMI bereitgestellt werden, führen Sie die folgenden Schritte aus, um ein neues aktualisiertes benutzerdefiniertes AMI bereitzustellen.

   a. Erstellen Sie eine neue Version Ihres AMI.

   b. Erstellen Sie eine neue Version der Startvorlage mit der neuen AMI-ID.

   c. Aktualisieren Sie die Knoten auf die neue Version der Startvorlage.

4. Aktivieren oder deaktivieren Sie im Dialogfeld Update node group version (Knotengruppenversion aktualisieren) die folgenden Optionen:

   - Update node group version (Knotengruppenversion aktualisieren) – Diese Option ist nicht verfügbar, wenn Sie ein benutzerdefiniertes AMI bereitgestellt haben oder Ihr Amazon EKS-optimiertes AMI derzeit über die neueste Version für Ihren Cluster verfügt.
   - Change launch template version (Startvorlagenversion ändern) – Diese Option ist nicht verfügbar, wenn die Knotengruppe ohne eine benutzerdefinierte Startvorlage bereitgestellt wird. Sie können die Version der Startvorlage nur für eine Knotengruppe aktualisieren, die mit einer benutzerdefinierten Startvorlage bereitgestellt wurde. Wählen Sie die Launch template version (Startvorlagenversion) aus, auf die Sie die Knotengruppe aktualisieren möchten. Wenn Ihre Knotengruppe mit einem benutzerdefinierten AMI konfiguriert ist, muss die ausgewählte Version auch ein AMI angeben. Wenn Sie ein Upgrade auf eine neuere Version Ihrer Startvorlage durchführen, wird jeder Knoten wiederverwertet, damit er der neuen Konfiguration der angegebenen Startvorlagenversion entspricht.

5. Wählen Sie für Update strategy (Aktualisierungsstrategie) eine der folgenden Optionen aus:

   - Fortlaufende Aktualisierung – Diese Option berücksichtigt die Budgets für die Pod-Unterbrechung Ihres Clusters. Aktualisierungen schlagen fehl, wenn ein Budgetproblem für Pod-Unterbrechung auftritt, das dazu führt, dass Amazon EKS die Pods, die auf dieser Knotengruppe ausgeführt werden, nicht ordnungsgemäß entwässert werden kann.

- Force Update -Bei dieser Option werden Budgets für Pod-Unterbrechungen nicht berücksichtigt. Aktualisierungen erfolgen unabhängig von Problemen mit Pod-Unterbrechung des Budgets, indem Knotenneustarts erzwungen werden.

6. Wählen Sie Aktualisieren.

Bearbeiten einer Knotengruppenkonfiguration

Sie können einige Konfigurationen einer verwalteten Knotengruppe ändern.

So bearbeiten Sie eine Knotengruppenkonfiguration:

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Cluster aus, der die zu bearbeitende Knotengruppe enthält.
3. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.
4. Wählen Sie die zu bearbeitende Knotengruppe aus und wählen Sie dann Edit (Bearbeiten).
5. (Optional) Gehen Sie auf der Seite Edit node group (Knotengruppe bearbeiten) wie folgt vor:

   a. Bearbeiten Sie die Node group scaling configuration (Skalierungskonfiguration der Knotengruppe).

      - Desired size (Gewünschte Größe) – Geben Sie die aktuelle Anzahl von Knoten an, die die verwaltete Knotengruppe beibehalten soll.
      - Minimum size (Mindestgröße) – Geben Sie die Mindestanzahl von Knoten an, auf die die verwaltete Knotengruppe skaliert werden kann.
      - Maximum size (Maximale Größe) – Geben Sie die maximale Anzahl von Knoten an, auf die die verwaltete Knotengruppe skaliert werden kann. Die maximale Anzahl von Knoten, die in einer Knotengruppe unterstützt werden, finden Sie unter Amazon-EKS-Service-Quotas.

   b. (Optional) Fügen Sie den Knoten in Ihrer Knotengruppe Kubernetes-Labels hinzu oder entfernen Sie sie. Die hier gezeigten Labels sind nur die Labels, die Sie mit Amazon EKS angewendet haben. Andere Labels können auf Ihren Knoten vorhanden sein, die hier nicht angezeigt werden.

   c. (Optional) Fügen Sie den Knoten in Ihrer Knotengruppe Kubernetes-Taints hinzu oder entfernen Sie sie. Hinzugefügte Färbungen können die Wirkung von

`NoSchedule`,**`NoExecute`**, oder**`PreferNoSchedule`** haben. Weitere Informationen finden Sie unter Knotenfehler auf verwalteten Knotengruppen.

d. (Optional) Fügen Sie Tags zu Ihrer Knotengruppenressource hinzu oder entfernen Sie sie. Diese Markierungen werden nur auf die Amazon-EKS-Knotengruppe angewendet. Knotengruppen-Tags werden nicht auf andere Ressourcen übertragen, die der Knotengruppe zugeordnet sind, z. B. die Amazon-EC2-Instances oder Subnetze.

e. (Optional) Bearbeiten Sie Konfiguration der Knotengruppe aktualisieren. Wählen Sie entweder Zahl oder Prozentanteil aus.

- Zahl – Wählen und geben Sie die Anzahl der Knoten in Ihrer Knotengruppe an, die parallel aktualisiert werden können. Diese Knoten sind während der Aktualisierung nicht verfügbar.
- Prozentsatz – Wählen und geben Sie den Prozentsatz der Knoten in der Knotengruppe an, die parallel aktualisiert werden können. Diese Knoten sind während der Aktualisierung nicht verfügbar. Dies ist nützlich, wenn Sie viele Knoten in Ihrer Knotengruppe haben.

f. Wenn Sie mit der Bearbeitung fertig sind, wählen Sie Save changes (Änderungen speichern).

Verhalten der Aktualisierung verwalteter Knoten

Die Upgrade-Strategie für verwaltete Amazon-EKS-Worker-Knoten umfasst vier verschiedene Phasen, die in den folgenden Abschnitten beschrieben werden.

Einrichtungsphase

Die Einrichtungsphase umfasst folgende Schritte:

1. Es erstellt eine neue Amazon-EC2-Startvorlage für die Auto-Scaling-Gruppe, die Ihrer Knotengruppe zugeordnet ist. Die neue Version der Startvorlage verwendet das Ziel-AMI oder die vom Kunden bereitgestellte Startvorlagenversion für die Aktualisierung.
2. Die Auto-Scaling-Gruppe wird so aktualisiert, dass sie die neueste Startvorlage verwendet.
3. Es bestimmt die maximale Anzahl der parallel zu aktualisierenden Knoten mithilfe der `updateConfig`-Eigenschaft für die Knotengruppe. Der maximal nicht verfügbare Wert hat ein Kontingent von 100 Knoten. Der Standardwert beträgt einen Knoten. Weitere Informationen dazu finden Sie unter der Eigenschaft `updateConfig` in der Amazon-EKS-API-Referenz.

Aufskalierungsphase

Beim Upgrade der Knoten in einer verwalteten Knotengruppe werden die aktualisierten Knoten in derselben Availability Zone gestartet wie diejenigen, die aktualisiert werden. Um diese Platzierung zu garantieren, verwenden wir den Availability-Zone-Rebalancing von Amazon EC2. Weitere Informationen dazu finden Sie unter Availability-Zone-Rebalancing im Amazon-EC2-Auto-Scaling-Benutzerhandbuch. Um diese Anforderung zu erfüllen, ist es möglich, dass wir bis zu zwei Instances pro Availability Zone in Ihrer verwalteten Knotengruppe starten.

Die Aufskalierungsphase umfasst folgende Schritte:

1. Es erhöht die maximale Größe und die gewünschte Größe der Auto Scaling-Gruppe um den jeweils größeren Wert:
   - Bis zu doppelt so viele Availability Zones, in denen die Auto-Scaling-Gruppe bereitgestellt wird.
   - Die maximale Nichtverfügbarkeit der Aktualisierung.

     Wenn Ihre Knotengruppe beispielsweise über fünf Availability Zones und `maxUnavailable` als eine verfügt, kann der Upgrade-Prozess maximal zehn Knoten starten. Wenn jedoch 20 (oder etwas höher als 10) `maxUnavailable` ist, würde der Prozess 20 neue Knoten starten.
2. Nach dem Skalieren der Auto-Scaling-Gruppe prüft es, ob die Knoten, die die neueste Konfiguration verwenden, in der Knotengruppe vorhanden sind. Dieser Schritt ist nur erfolgreich, wenn er diese Kriterien erfüllt:
   - Mindestens ein neuer Knoten wird in jeder Availability Zone gestartet, in der der Knoten existiert.
   - Jeder neue Knoten sollte im `Ready`-Zustand sein.
   - Neue Knoten sollten Amazon-EKS-Labels angewandt haben.

     Dies sind die von Amazon EKS auf die Worker-Knoten in einer regulären Knotengruppe angewandten Labels:

     - `eks.amazonaws.com/nodegroup-image=`*`$amiName`*
     - `eks.amazonaws.com/nodegroup=`*`$nodeGroupName`*

     Dies sind die von Amazon EKS angewendeten Labels auf den Worker-Knoten in einer benutzerdefinierten Startvorlage oder AMI-Knotengruppe:

     - `eks.amazonaws.com/nodegroup-image=`*`$amiName`*
     - `eks.amazonaws.com/nodegroup=`*`$nodeGroupName`*
     - `eks.amazonaws.com/sourceLaunchTemplateId=`*`$launchTemplateId`*

- `eks.amazonaws.com/sourceLaunchTemplateVersion=`*`$launchTemplateVersion`*

3. Es markiert Knoten als nicht planbar, um zu vermeiden, dass neue Pods geplant werden. Es kennzeichnet Knoten auch mit `node.kubernetes.io/exclude-from-external-load-balancers=true`, um die Knoten aus den Load Balancern zu entfernen, bevor die Knoten beendet werden.

Die folgenden sind bekannte Gründe, die zu einem `NodeCreationFailure`-Fehler in dieser Phase führen:

Nicht genügend Kapazität in der Availability Zone

Es besteht die Möglichkeit, dass die Availability Zone nicht über die Kapazität der angeforderten Instance-Typen verfügt. Es wird empfohlen, beim Erstellen einer verwalteten Knotengruppe mehrere Instance-Typen zu konfigurieren.

EC2-Instance-Limits in Ihrem Konto

Möglicherweise müssen Sie die Anzahl der Amazon-EC2-Instances erhöhen, die Ihr Konto gleichzeitig mit Service Quotas ausführen kann. Weitere Informationen finden Sie unter EC2-Service-Quotas im Amazon-Elastic-Compute-Cloud-Benutzerhandbuch für Linux-Instances.

Anwenderbezogene Benutzerdaten

Anwenderbezogene Benutzerdaten können manchmal den Bootstrap-Prozess stören. Dieses Szenario kann dazu führen, dass `kubelet` nicht auf dem Knoten startet oder Knoten nicht die erwarteten Amazon-EKS-Labels auf ihnen erhalten. Weitere Informationen finden Sie unter Angeben eines AMI.

Alle Änderungen, die dazu führen, dass ein Knoten fehlerhaft ist oder nicht bereit ist

Knotenfestplattendruck, Speicherdruck und ähnliche Bedingungen können dazu führen, dass ein Knoten nicht in den `Ready`-Zustand wechselt.

Aktualisierungs-Phase

Die Aktualisierungs-Phase umfasst folgende Schritte:

1. Es wählt nach dem Zufallsprinzip einen Knoten aus, der aktualisiert werden muss, bis zum Maximum der für die Knotengruppe nicht verfügbar ist.
2. Es entleert die Pods vom Knoten. Wenn die Pods den Knoten nicht innerhalb von 15 Minuten verlassen und es kein Force-Flag gibt, schlägt die Upgrade-Phase mit einem

`PodEvictionFailure`-Fehler fehl. Für dieses Szenario können Sie das Force-Flag mit der `update-nodegroup-version`-Anforderung anwenden, um die Pods zu löschen.

3. Es sperrt den Knoten ab, nachdem jeder Pod entfernt wurde, und wartet 60 Sekunden. Dies geschieht, damit der Service-Controller keine neuen Anforderungen an diesen Knoten sendet und diesen Knoten aus der Liste der aktiven Knoten entfernt.
4. Es sendet eine Beendigungsanforderung an die Auto-Scaling-Gruppe für den abgesperrten Knoten.
5. Es wiederholt die vorherigen Aktualisierungsschritte, bis keine Knoten in der Knotengruppe mehr vorhanden sind, die mit der früheren Version der Startvorlage bereitgestellt wurden.

Die folgenden sind bekannte Gründe, die zu einem `PodEvictionFailure`-Fehler in dieser Phase führen:

Aggressive PDB

Aggressive PDB ist auf dem Pod definiert oder es gibt mehrere PDBs, die auf denselben Pod zeigen.

Bereitstellung toleriert alle Taints

Sobald jeder Pod entfernt wurde, wird erwartet, dass der Knoten leer ist, da der Knoten in den vorherigen Schritten verunreinigt ist. Wenn die Bereitstellung jedoch jeden Taint toleriert, ist der Knoten eher nicht leer, was zu einem Pod-Bereinigungsfehler führt.

Abskalierungsphase

Die Abskalierungsphase verringert die maximale Größe der Auto-Scaling-Gruppe und die gewünschte Größe um eins, um zu den Werten zurückzukehren, bevor die Aktualisierung gestartet wurde.

Wenn der Upgrade-Workflow feststellt, dass der Cluster-Autoscaler die Knotengruppe während der Herunterskalierungsphase des Workflows skaliert, wird er sofort beendet, ohne die Knotengruppe wieder auf ihre ursprüngliche Größe zu bringen.

Knotenfehler auf verwalteten Knotengruppen

Amazon EKS unterstützt die Konfiguration von Kubernetes-Taints über verwaltete Knotengruppen. Taints und Tolerationen arbeiten zusammen, um sicherzustellen, dass Pods nicht auf ungeeigneten

Knoten geplant werden. Ein oder mehrere Taints können auf einen Knoten angewendet werden. Dies markiert, dass der Knoten keine Pods akzeptieren sollte, die die Taints nicht tolerieren. Tolerationen werden auf Pods angewendet und erlauben, aber es ist nicht erforderlich, dass die Pods auf Knoten mit übereinstimmenden Taints einplanen. Weitere Informationen zu finden Sie unter Taints and Tolerations (Taints und Toleranzen) in der Kubernetes-Dokumentation.

Kubernetes-Knoten-Taints können über die AWS Management Console oder die Amazon EKS API auf neue und vorhandene verwaltete Knotengruppen angewendet werden.

- Hinweise zum Erstellen einer Knotengruppe mit einem Taint mithilfe der AWS Management Console finden Sie unter Erstellen einer verwalteten Knotengruppe.
- Im Folgenden finden Sie ein Beispiel für das Erstellen einer Knotengruppe mit einem Taint mithilfe derAWS CLI:

```
aws eks create-nodegroup \
 --cli-input-json '
{
  "clusterName": "my-cluster",
  "nodegroupName": "node-taints-example",
  "subnets": [
     "subnet-1234567890abcdef0",
     "subnet-abcdef01234567890",
     "subnet-021345abcdef67890"
   ],
  "nodeRole": "arn:aws:iam::111122223333:role/AmazonEKSNodeRole",
  "taints": [
     {
         "key": "dedicated",
         "value": "gpuGroup",
         "effect": "NO_SCHEDULE"
     }
   ]
}'
```

Weitere Informationen und Beispiele zur Verwendung finden Sie unter Taint in der Kubernetes-Referenzdokumentation.

Note

- Taints können aktualisiert werden, nachdem Sie die Knotengruppe mit dem `UpdateNodegroupConfig`-API erstellen.
- Der Taint-Schlüssel muss mit einem Buchstaben oder einer Zahl beginnen. Es kann Buchstaben, Zahlen, Bindestriche (`-`), Punkte (`.`) und Unterstriche (`_`) enthalten. Er kann bis zu 63 Zeichen lang sein.
- Optional kann der Taint-Schlüssel mit einem DNS-Subdomänenpräfix und einem einzelnen `/` beginnen. Wenn er mit einem DNS-Subdomänenpräfix beginnt, kann er 253 Zeichen lang sein.
- Der Wert ist optional und muss mit einem Buchstaben oder einer Zahl beginnen. Es kann Buchstaben, Zahlen, Bindestriche (`-`), Punkte (`.`) und Unterstriche (`_`) enthalten. Er kann bis zu 63 Zeichen lang sein.
- Bei direkter Anwendung von Kubernetes oder der AWS Management Console muss der Taint-Effekt **`NoSchedule`**, **`PreferNoSchedule`** oder **`NoExecute`** sein. Bei Verwendung der AWS CLI oder der API muss der Taint-Effekt jedoch **`NO_SCHEDULE`**, **`PREFER_NO_SCHEDULE`** oder **`NO_EXECUTE`** sein.
- Für eine Knotengruppe sind maximal 50 Taints zulässig.
- Wenn Taints, die mithilfe einer verwalteten Knotengruppe erstellt wurden, manuell von einem Knoten entfernt werden, fügt Amazon EKS die Taints nicht wieder zum Knoten hinzu. Dies gilt auch dann, wenn die Taints in der Konfiguration der verwalteten Knotengruppe angegeben sind.

Sie können den AWS CLI-Befehl `aws eks update-nodegroup-config` verwenden, um Eigenschaften für verwaltete Knotengruppen hinzuzufügen, zu entfernen oder zu ersetzen.

Anpassen verwalteter Knoten mit Startvorlagen

Für die höchste Anpassungsstufe können Sie verwaltete Knoten mithilfe Ihrer eigenen Startvorlage bereitstellen. Die Verwendung einer Startvorlage ermöglicht Funktionen wie die folgenden:

- Bereitstellen von Bootstrap-Argumenten bei der Bereitstellung eines Knotens bereitstellen, z. B. zusätzliche `kubelet`-Argumente
- Zuweisen von IP-Adressen zu Pods aus einem anderen CIDR-Block als der dem Knoten zugewiesenen IP-Adresse

- Bereitstellen Ihres eigenen benutzerdefinierten AMI auf Knoten
- Bereitstellen Ihres eigenen benutzerdefinierten CNI auf Knoten

Wenn Sie beim ersten Erstellen einer verwalteten Knotengruppe Ihre eigene Startvorlage angeben, haben Sie auch später mehr Flexibilität. Wenn Sie eine verwaltete Knotengruppe mit Ihrer eigenen Startvorlage bereitstellen, können Sie sie schrittweise mit einer anderen Version derselben Startvorlage aktualisieren. Wenn Sie Ihre Knotengruppe auf eine andere Version Ihrer Startvorlage aktualisieren, werden alle Knoten in der Gruppe wiederverwertet, damit sie der neuen Konfiguration der angegebenen Startvorlagenversion entsprechen.

Verwaltete Knotengruppen werden immer mit einer Startvorlage bereitgestellt, die mit der Amazon-EC2-Auto-Scaling-Gruppe verwendet werden soll. Wenn Sie keine Startvorlage bereitstellen, erstellt die Amazon-EKS-API automatisch eine mit Standardwerten in Ihrem Konto. Es wird jedoch nicht empfohlen, automatisch generierte Startvorlagen zu ändern. Außerdem können vorhandene Knotengruppen, die keine benutzerdefinierte Startvorlage verwenden, nicht direkt aktualisiert werden. Stattdessen müssen Sie eine neue Knotengruppe mit einer benutzerdefinierten Startvorlage erstellen.

Grundlagen der Konfiguration der Vorlage starten

Sie können eine Amazon EC2 Auto Scaling Scaling-Startvorlage mit dem AWS Management Console AWS CLI, oder einem AWS SDK erstellen. Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto-Scaling-Gruppe im Amazon-EC2-Auto-Scaling-Benutzerhandbuch. Einige der Einstellungen in einer Startvorlage ähneln den Einstellungen, die für die Konfiguration verwalteter Knoten verwendet werden. Beim Bereitstellen oder Aktualisieren einer Knotengruppe mit einer Startvorlage müssen einige Einstellungen entweder in der Knotengruppenkonfiguration oder in der Startvorlage angegeben werden. Geben Sie nicht an beiden Stellen eine Einstellung an. Wenn eine Einstellung vorhanden ist, wo sie nicht sollte, schlagen Vorgänge wie das Erstellen oder Aktualisieren einer Knotengruppe fehl.

In der folgenden Tabelle werden die Einstellungen aufgelistet, die in einer Startvorlage nicht zulässig sind. Es listet auch ähnliche Einstellungen auf, falls verfügbar, die in der Konfiguration der verwalteten Knotengruppe erforderlich sind. Die aufgelisteten Einstellungen sind die Einstellungen, die in der Konsole angezeigt werden. Sie haben möglicherweise ähnliche, aber unterschiedliche Namen im SDK AWS CLI und SDK.

Startvorlage - Verboten	Amazon-EKS-Knotengruppenkonfiguration
Subnetz in Netzwerkschnittstellen (Hinzufügen einer Netzwerkschnittstelle)	Subnets (Subnetze) unter Node group network configuration (Netzwerkkonfiguration der Knotengruppe) auf der Seite Specify networkin g (Netzwerk angeben)
IAM-Instance-Profil in Erweiterte Details	Node IAM role (Knoten-IAM-Rolle) unter Node group configuration (Knotengruppenkonf iguration) auf der Seite Configure Node group (Knotengruppe konfigurieren)
Verhalten beim Herunterfahren und Stop – Ruhezustand Verhalten in Erweiterte Details. Beibehalten von Standard Nicht in die Startvorl ageneinstellung einschließen in der Startvorl age für beide Einstellungen.	Keine Entsprechung Amazon EKS muss den Instance-Lebenszyklus steuern, nicht die Auto-Scaling-Gruppe.

In der folgenden Tabelle sind die unzulässigen Einstellungen in einer Konfiguration einer verwalteten Knotengruppe aufgeführt. Sie listet auch ähnliche Einstellungen auf, falls vorhanden, die in einer Startvorlage erforderlich sind. Die aufgelisteten Einstellungen sind die Einstellungen, die in der Konsole angezeigt werden. Sie haben möglicherweise ähnliche Namen im SDK AWS CLI und SDK.

Konfiguration der Amazon-EKS-Knotengruppe — Verboten	Startvorlage
(Nur wenn Sie ein benutzerdefiniertes AMI in einer Startvorlage angegeben haben) AMI type (AMI-Typ) unter Node Group compute configuration (Computing-Konfiguration der Knotengruppe) auf der Seite Set compute and scaling configuration (Computing- und Skalierungskonfiguration festlegen) – Die Konsole zeigt Specified in launch template (In Startvorlage angegeben) und die angegebene AMI-ID an.	Application and OS Images (Amazon Machine Image) (Anwendungs- und Betriebssystem-Ima ges (Amazon Machine Image)) unter Inhalt der Startvorlage – Sie müssen eine ID angeben, wenn Sie eine der folgenden Anforderungen haben: • Verwenden eines benutzerdefinierten AMI Wenn Sie ein AMI angeben, das nicht den unter aufgeführten Anforderungen in

Konfiguration der Amazon-EKS-Knotengruppe — Verboten	Startvorlage
Wenn Application and OS Images (Amazon Machine Image) (Anwendungs- und Betriebssystem-Images (Amazon Machine Image)) nicht in der Startvorlage kein AMI-Typ angegeben wurde, können Sie in der Knotengruppenkonfiguration ein AMI auswählen.	Angeben eines AMI entspricht, schlägt die Knotengruppenbereitstellung fehl. • Sie möchten Benutzerdaten bereitstellen, um Argumente für die Datei bereitzustellen, die in einem für Amazon EKS optimierten AMI enthalten ist. Sie können Ihren Instances ermöglichen, eine deutlich höhere Anzahl von IP-Adressen zuzuweisenPods, IP-Adressen Pods aus einem anderen CIDR-Block als dem der Instance zuzuweisen oder einen privaten Cluster ohne ausgehenden Internetzugang bereitzustellen. Weitere Informationen finden Sie unter den folgenden Themen: • Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten • Benutzerdefinierte Netzwerke für Pods • Anforderungen an private Cluster • Angeben eines AMI
Datenträgergröße unter Node Group compute configuration (Computing-Konfiguration der Knotengruppe) auf der Seite Set compute and scaling configuration – Die Konsole zeigt Specified in launch template (In Startvorlage angegeben) an.	Größe unter Speicher (Volumes) (Hinzufügen eines neuen Volumes) enthalten. Sie müssen dies in der Startvorlage angeben.
SSH key pair (SSH-Schlüsselpaar) unter Node group configuration (Knotengruppenkonfiguration) auf der Seite Specify Networking (Netzwerk angeben) – Die Konsole zeigt den in der Startvorlage angegebenen Schlüssel an oder Not specified in launch template (Nicht in der Startvorlage angegeben).	Schlüsselpaarname in Schlüsselpaar (Login).

Konfiguration der Amazon-EKS-Knotengruppe — Verboten	Startvorlage
Bei Verwendung einer Startvorlage können Sie keine Quellsicherheitsgruppen angeben, für die der Remotezugriff zulässig ist.	Sicherheitsgruppen unter Netzwerkeinstellungen für die Instance oder Sicherheitsgruppen unter Netzwerkschnittstellen (Netzwerkschnittstelle hinzufügen), aber nicht beides. Weitere Informationen dazu finden Sie unter Benutzerdefinierte Sicherheitsgruppen.

Note

- Wenn Sie eine Knotengruppe mithilfe einer Startvorlage bereitstellen, geben Sie in einer Startvorlage unter Inhalt der Startvorlage null oder einen Instance-Typ an. Alternativ können Sie 0 bis 20 Instance-Typen für Instance-Typen auf der Seite Einstellen von Compute- und Skalierungskonfiguration in der Konsole angeben. Oder Sie können dies mit anderen Tools tun, die die Amazon EKS API verwenden. Wenn Sie in einer Startvorlage einen Instance-Typ angeben und diese Startvorlage zum Bereitstellen Ihrer Knotengruppe verwenden, können Sie keine Instance-Typen in der Konsole oder mit anderen Tools angeben, die die Amazon EKS API verwenden. Wenn Sie für eine Startvorlage, in der Konsole oder mit anderen Tools, die die Amazon EKS API verwenden, keinen Instance-Typ angeben, wird der `t3.medium`-Instance-Typ verwendet. Wenn Ihre Knotengruppe den Spot-Kapazitätstyp verwendet, empfehlen wir, mehrere Instance-Typen über die Konsole anzugeben. Weitere Informationen dazu finden Sie unter Kapazitätstypen für verwaltete Knotengruppen.
- Wenn Container, die Sie für die Knotengruppe bereitstellen, den Instance-Metadaten-Service Version 2 verwenden, stellen Sie sicher, dass Sie die Hop-Limits für Metadaten auf 2 in Ihrer Startvorlage festsetzen. Weitere Informationen finden Sie unter Instance-Metadaten und Benutzerdaten im Amazon-EC2-Benutzerhandbuch. Wenn Sie eine verwaltete Knotengruppe bereitstellen, ohne eine benutzerdefinierte Startvorlage zu verwenden, wird dieser Wert automatisch für die Knotengruppe in der Standardstartvorlage festgelegt.

Markieren von Amazon-EC2-Instances

Sie können das `TagSpecification`-Parameter einer Startvorlage verwenden, um anzugeben, welche Markierungen auf Amazon-EC2-Instances in Ihrer Knotengruppe angewendet werden sollen. Die IAM-Entität, die die `CreateNodegroup`- oder `UpdateNodegroupVersion`-APIs aufruft muss über die Berechtigungen für `ec2:RunInstances` und `ec2:CreateTags` verfügen, und die Markierungen müssen der Startvorlage hinzugefügt werden.

Benutzerdefinierte Sicherheitsgruppen

Sie können eine Startvorlage verwenden, um benutzerdefinierte Amazon-EC2-Sicherheitsgruppen anzugeben die auf Instances in Ihrer Knotengruppe angewendet werden. Dies kann entweder im Parameter für Sicherheitsgruppen auf Instance-Ebene oder als Teil der Konfigurationsparameter der Netzwerkschnittstelle sein. Sie können eine Instance nicht über eine Startvorlage starten, die Sicherheitsgruppen und eine Netzwerkschnittstelle angibt. Beachten Sie die folgenden Bedingungen, die für die Verwendung benutzerdefinierter Sicherheitsgruppen mit verwalteten Knotengruppen gelten:

- Amazon EKS erlaubt nur Startvorlagen mit einer einzigen Netzwerkschnittstellenspezifikation.
- Standardmäßig wendet Amazon EKS die Cluster-Sicherheitsgruppe zu den Instances in Ihrer Knotengruppe hinzu, um die Kommunikation zwischen Knoten und der Steuerungsebene zu erleichtern. Wenn Sie benutzerdefinierte Sicherheitsgruppen in der Startvorlage mit einer der oben genannten Optionen angeben, fügt Amazon EKS die Cluster-Sicherheitsgruppe nicht hinzu. Sie müssen daher sicherstellen, dass die eingehenden und ausgehenden Regeln Ihrer Sicherheitsgruppen die Kommunikation mit dem Endpunkt Ihres Clusters ermöglichen. Wenn Ihre Sicherheitsgruppenregeln falsch sind, können die Worker-Knoten dem Cluster nicht beitreten. Weitere Informationen zu Sicherheitsgruppenregeln finden Sie unter Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe.
- Wenn Sie SSH-Zugriff auf die Instances in Ihrer Knotengruppe benötigen, müssen Sie eine Sicherheitsgruppe einschließen, die diesen Zugriff zulässt.

Amazon-EC2-Benutzerdaten

Die Startvorlage enthält einen Abschnitt für benutzerdefinierte Benutzerdaten. Sie können in diesem Abschnitt Konfigurationseinstellungen für Ihre Knotengruppe angeben, ohne einzelne benutzerdefinierte AMIs manuell zu erstellen. Weitere Informationen zu diesen Einstellungen für Bottlerocket finden Sie unter Verwendung von Benutzerdaten auf GitHub.

Sie können Amazon-EC2-Benutzerdaten in Ihrer Startvorlage mithilfe von `cloud-init`, wenn Sie Ihre Instances starten. Weitere Informationen dazu finden Sie in der cloud-init-Dokumentation. Ihre Benutzerdaten können verwendet werden, um allgemeine Konfigurationsvorgänge durchzuführen. Dieser Filter umfasst die folgenden Optionen:

- Einschließen von Benutzern oder Gruppen
- Installieren von Paketen

Amazon-EC2-Benutzerdaten in Startvorlagen, die mit verwalteten Knotengruppen verwendet werden, müssen im mehrteiligen MIME-Archiv-Format für Amazon Linux AMIs und im TOML-Format für Bottlerocket-AMIs vorliegen. Dies liegt daran, dass Ihre Benutzerdaten mit Amazon-EKS-Benutzerdaten zusammengeführt werden, die für Knoten erforderlich sind, um dem Cluster beizutreten. Geben Sie keine Befehle in Ihren Benutzerdaten an, die `kubelet` starten oder ändern. Dies wird als Teil der von Amazon EKS zusammengeführten Benutzerdaten durchgeführt. Bestimmte `kubelet`-Parameter, z. B. das Festlegen von Beschriftungen auf Knoten, können direkt über die API für verwaltete Knotengruppen konfiguriert werden.

> **Note**
>
> Weitere Hinweise zur erweiterten `kubelet`-Anpassung, einschließlich des manuellen Startens oder Übergebens benutzerdefinierter Konfigurationsparameter, finden Sie unter Angeben eines AMI. Wenn in einer Startvorlage eine benutzerdefinierte AMI-ID angegeben ist, führt Amazon EKS keine Benutzerdaten zusammen.

Die folgenden Details enthalten weitere Informationen zum Abschnitt Benutzerdaten.

Amazon Linux 2 user data

Sie können mehrere Benutzerdatenblöcke in einer einzelnen mehrteiligen MIME-Datei kombinieren. So können Sie beispielsweise einen Cloud-Boothook, der den Docker-Daemon konfiguriert, mit einem Benutzerdaten-Shell-Skript kombinieren, das ein benutzerdefiniertes Paket installiert. Eine mehrteilige MIME-Datei umfasst folgende Komponenten:

- Deklaration von Inhaltstyp und Teilgrenze – `Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="`
- Deklaration der MIME-Version – `MIME-Version: 1.0`
- Ein oder mehrere Benutzerdatenblöcke mit folgenden Komponenten:

- Eröffnungsgrenze, die den Beginn eines Benutzerdatenblocks signalisiert – `--==MYBOUNDARY==`
- Die Inhaltstypdeklaration für den Block: `Content-Type: text/cloud-config; charset="us-ascii"`. Weitere Informationen zu Inhaltstypen finden Sie in der Cloud-Init-Dokumentation.
- Der Inhalt der Benutzerdaten (z. B. eine Liste von Shell-Befehlen oder `cloud-init`-Direktiven).
- Abschlussgrenze, die das Ende der mehrteiligen MIME-Datei signalisiert: `--==MYBOUNDARY==--`

Im Folgenden finden Sie ein Beispiel für eine mehrteilige MIME-Datei, die Sie verwenden können, um Ihre eigene zu erstellen.

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

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

#!/bin/bash
echo "Running custom user data script"

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

Amazon Linux 2023 user data

Amazon Linux 2023 (AL2023) führt einen neuen Knoteninitialisierungsprozess ein`nodeadm`, der ein YAML-Konfigurationsschema verwendet. Wenn Sie selbstverwaltete Knotengruppen oder ein AMI mit einer Startvorlage verwenden, müssen Sie jetzt beim Erstellen einer neuen Knotengruppe explizit zusätzliche Cluster-Metadaten angeben. Ein Beispiel für die mindestens erforderlichen Parameter lautet wie folgt, wobei `apiServerEndpoint``certificateAuthority`, und Service jetzt erforderlich `cidr` sind:

```
---
apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig
spec:
  cluster:
```

```
    name: my-cluster
    apiServerEndpoint: https://example.com
    certificateAuthority: Y2VydGlmaWNhdGVBdXRob3JpdHk=
    cidr: 10.100.0.0/16
```

In der Regel legen Sie diese Konfiguration in Ihren Benutzerdaten fest, entweder unverändert oder eingebettet in ein mehrteiliges MIME-Dokument:

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

--BOUNDARY
Content-Type: application/node.eks.aws

---
apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig spec: [...]

--BOUNDARY--
```

In AL2 wurden die Metadaten dieser Parameter aus dem Amazon `DescribeCluster` EKS-API-Aufruf ermittelt. Mit AL2023 hat sich dieses Verhalten geändert, da durch den zusätzlichen API-Aufruf die Gefahr einer Drosselung bei der Skalierung großer Knoten besteht. Diese Änderung wirkt sich nicht auf Sie aus, wenn Sie verwaltete Knotengruppen ohne Startvorlage verwenden oder wenn Sie Karpenter Weitere Informationen zu `certificateAuthority` und Service `cidr` finden Sie `DescribeCluster` in der Amazon EKS API-Referenz.

Bottlerocket user data

Bottlerocket strukturiert Benutzerdaten im TOML-Format. Sie können Benutzerdaten angeben, die mit den von Amazon EKS bereitgestellten Benutzerdaten zusammengeführt werden sollen. Zum Beispiel können Sie zusätzliche `kubelet`-Einstellungen bereitstellen.

```
[settings.kubernetes.system-reserved]
cpu = "10m"
memory = "100Mi"
ephemeral-storage= "1Gi"
```

Weitere Informationen zu unterstützten Einstellungen finden Sie in der Bottlerocket-Dokumentation. Sie können Knotenbezeichnungen und Taints in Ihren Benutzerdaten

konfigurieren. Es wird jedoch empfohlen, diese stattdessen in Ihrer Knotengruppe zu konfigurieren. Amazon EKS wendet diese Konfigurationen an, wenn Sie dies tun.

Wenn Benutzerdaten zusammengeführt werden, wird die Formatierung nicht beibehalten, der Inhalt bleibt jedoch unverändert. Die Konfiguration, die Sie in Ihren Benutzerdaten angeben, überschreibt alle Einstellungen, die von Amazon EKS konfiguriert wurden. Wenn Sie also `settings.kubernetes.max-pods` oder festlegen`settings.kubernetes.cluster-dns-ip`, werden diese Werte in Ihren Benutzerdaten auf die Knoten angewendet.

Amazon EKS unterstützt nicht alle gültigen TOML. Im Folgenden finden Sie eine Liste bekannter nicht unterstützter Formate:

- Zitate in Anführungszeichen: `'quoted "value"' = "value"`
- Anführungszeichen mit Escapezeichen in Werten: `str = "I'm a string. \"You can quote me\""`
- Gemischte Gleitkommazahlen und ganze Zahlen: `numbers = [ 0.1, 0.2, 0.5, 1, 2, 5 ]`
- Gemischte Typen in Arrays: `contributors = ["foo@example.com", { name = "Baz", email = "baz@example.com" }]`
- Kopfzeilen in Klammern mit Anführungszeichen: `[foo."bar.baz"]`

Windows user data

Windows-Benutzerdaten verwenden PowerShell-Befehle. Beim Erstellen einer verwalteten Knotengruppe werden Ihre benutzerdefinierten Benutzerdaten mit den von Amazon EKS verwalteten Benutzerdaten kombiniert. Ihre PowerShell-Befehle stehen an erster Stelle, gefolgt von den verwalteten Benutzerdatenbefehlen, alle innerhalb eines **<powershell></powershell>**-Tags.

> **Note**
>
> Wenn in der Startvorlage keine AMI-ID angegeben ist, verwenden Sie nicht das Amazon-EKS-Bootstrap-Skript von Windows in den Benutzerdaten, um Amazon EKS zu konfigurieren.

Beispielbenutzerdaten lauten wie folgt.

```
<powershell>
Write-Host "Running custom user data script"
</powershell>
```

Angeben eines AMI

Wenn Sie eine der folgenden Anforderungen haben, geben Sie eine AMI-ID in der `ImageId`-Startvorlage. Wählen Sie die Anforderung, die Sie für zusätzliche Informationen haben.

Stellen Sie Benutzerdaten bereit, um Argumente an die **`bootstrap.sh`**-Datei zu übergeben, die in einem für Amazon EKS optimierten Linux/Bottlerocket-AMI enthalten ist

Bootstrapping ist ein Begriff, der verwendet wird, um das Hinzufügen von Befehlen zu beschreiben, die beim Start einer Instance ausgeführt werden können. Bootstrapping ermöglicht beispielsweise die Verwendung zusätzlicher `kubelet`-Argumente. Sie können Argumente mithilfe von `eksctl` an das `bootstrap.sh`-Skript übergeben, ohne eine Startvorlage anzugeben. Oder Sie können dies tun, indem Sie die Informationen im Abschnitt Benutzerdaten einer Startvorlage angeben.

eksctl without specifying a launch template

Erstellen Sie eine Datei mit dem Namen *`my-nodegroup.yaml`* und dem folgenden Inhalt. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte. Die Argumente `--apiserver-endpoint`, `--b64-cluster-ca` und `--dns-cluster-ip` sind optional. Wenn sie definiert werden, verhindert dies jedoch, dass das `bootstrap.sh`-Skript einen `describeCluster`-Aufruf durchführt. Dies ist nützlich in privaten Cluster-Setups oder Clustern, in denen Sie häufig Knoten ab- und ausskalieren. Weitere Informationen zum `bootstrap.sh`-Skript finden Sie in der Datei `bootstrap.sh` auf GitHub.

- Das einzige erforderliche Argument ist der Clustername (*`my-cluster`*).
- Um die optimierte AMI-ID für `ami-`*`1234567890abcdef0`* abzurufen, können Sie die Tabellen in den folgenden Abschnitten verwenden:
  - Abrufen von Amazon-EKS-optimierten Amazon-Linux-AMI-IDs
  - Abrufen von Amazon-EKS-optimierten Bottlerocket-AMI-IDs
  - Abrufen von Amazon-EKS-optimierten Windows-AMI-IDs
- Um die *`certificate-authority`* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus.

```
aws eks describe-cluster --query "cluster.certificateAuthority.data" --output text
 --name my-cluster --region region-code
```

- Um den *api-server-endpoint* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus.

```
aws eks describe-cluster --query "cluster.endpoint" --output text --name my-
cluster --region region-code
```

- Der Wert für `--dns-cluster-ip`ist Ihr Service CIDR mit `.10`am Ende. Um das *service-cidr* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus. Wenn der zurückgegebene Wert beispielsweise `ipv4 10.100.0.0/16` ist, ist Ihr Wert *10.100.0.10*.

```
aws eks describe-cluster --query "cluster.kubernetesNetworkConfig.serviceIpv4Cidr"
 --output text --name my-cluster --region region-code
```

- Dieses Beispiel umfasst ein `kubelet`-Argument, mit dem ein benutzerdefinierter `max-pods`-Wert anhand des `bootstrap.sh`-Skripts festgelegt wird, das im Amazon-EKS-optimierten AMI enthalten ist. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten. Hilfe zur Auswahl von *my-max-pods-value* finden Sie unter Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ.

```
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code

managedNodeGroups:
  - name: my-nodegroup
    ami: ami-1234567890abcdef0
    instanceType: m5.large
    privateNetworking: true
    disableIMDSv1: true
    labels: { x86-al2-specified-mng }
    overrideBootstrapCommand: |
```

```
    #!/bin/bash
    /etc/eks/bootstrap.sh my-cluster \
      --b64-cluster-ca certificate-authority \
      --apiserver-endpoint api-server-endpoint \
      --dns-cluster-ip service-cidr.10 \
      --kubelet-extra-args '--max-pods=my-max-pods-value' \
      --use-max-pods false
```

Informationen zu allen verfügbaren `eksctl-config`-Dateioptionen finden Sie im Config file schema (Config-Datei-Schema) in der `eksctl`-Dokumentation. Das `eksctl`-Dienstprogramm erstellt eine Startvorlage für Sie und füllt die Benutzerdaten mit den Daten aus, die Sie in der `config`-Datei angeben.

Erstellen Sie eine Knoten-Gruppe mit dem folgenden Befehl.

```
eksctl create nodegroup --config-file=my-nodegroup.yaml
```

User data in a launch template

Geben Sie die folgenden Informationen im Abschnitt Benutzerdaten Ihrer Startvorlage an. Ersetzen Sie jede *example value* durch Ihre eigenen Werte. Die Argumente `--apiserver-endpoint`, `--b64-cluster-ca` und `--dns-cluster-ip` sind optional. Wenn sie definiert werden, verhindert dies jedoch, dass das `bootstrap.sh`-Skript einen `describeCluster`-Aufruf durchführt. Dies ist nützlich in privaten Cluster-Setups oder Clustern, in denen Sie häufig Knoten ab- und ausskalieren. Weitere Informationen zum `bootstrap.sh`-Skript finden Sie in der Datei `bootstrap.sh` auf GitHub.

- Das einzige erforderliche Argument ist der Clustername (*my-cluster*).
- Um die *certificate-authority* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus.

  ```
  aws eks describe-cluster --query "cluster.certificateAuthority.data" --output text
   --name my-cluster --region region-code
  ```

- Um den *api-server-endpoint* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus.

  ```
  aws eks describe-cluster --query "cluster.endpoint" --output text --name my-
  cluster --region region-code
  ```

- Der Wert für `--dns-cluster-ip` ist Ihr Service CIDR mit `.10` am Ende. Um das *service-cidr* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus. Wenn der zurückgegebene Wert beispielsweise `ipv4 10.100.0.0/16` ist, ist Ihr Wert *10.100.0.10*.

```
aws eks describe-cluster --query "cluster.kubernetesNetworkConfig.serviceIpv4Cidr"
 --output text --name my-cluster --region region-code
```

- Dieses Beispiel umfasst ein `kubelet`-Argument, mit dem ein benutzerdefinierter `max-pods`-Wert anhand des `bootstrap.sh`-Skripts festgelegt wird, das im Amazon-EKS-optimierten AMI enthalten ist. Hilfe zur Auswahl von *my-max-pods-value* finden Sie unter Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ.

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

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

#!/bin/bash
set -ex
/etc/eks/bootstrap.sh my-cluster \
  --b64-cluster-ca certificate-authority \
  --apiserver-endpoint api-server-endpoint \
  --dns-cluster-ip service-cidr.10 \
  --kubelet-extra-args '--max-pods=my-max-pods-value' \
  --use-max-pods false

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

Stellen Sie Benutzerdaten bereit, um Argumente an die **`Start-EKSBootstrap.ps1`**-Datei zu übergeben, die in einem für Amazon EKS optimierten Windows -AMI enthalten ist

Bootstrapping ist ein Begriff, der verwendet wird, um das Hinzufügen von Befehlen zu beschreiben, die beim Start einer Instance ausgeführt werden können. Sie können Argumente mithilfe von `eksctl` an das `Start-EKSBootstrap.ps1`-Skript übergeben, ohne eine Startvorlage anzugeben. Oder Sie können dies tun, indem Sie die Informationen im Abschnitt Benutzerdaten einer Startvorlage angeben.

Beachten Sie beim Angeben einer benutzerdefinierten Windows-AMI-ID die folgenden Überlegungen:

- Sie müssen eine Startvorlage verwenden und die erforderlichen Bootstrap-Befehle im Abschnitt Benutzerdaten angeben. Um Ihre gewünschte Windows-ID abzurufen, können Sie die Tabelle in Amazon-EKS-optimierte Windows-AMIs verwenden.
- Es gibt verschiedene Grenzwerte und Bedingungen. Sie müssen beispielsweise `eks:kube-proxy-windows` zu Ihrer AWS IAM Authenticator-Konfigurationsübersicht etwas hinzufügen. Weitere Informationen finden Sie unter Grenzen und Bedingungen bei der Angabe einer AMI-ID.

Geben Sie die folgenden Informationen im Abschnitt Benutzerdaten Ihrer Startvorlage an. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte. Die Argumente `-APIServerEndpoint`, `-Base64ClusterCA` und `-DNSClusterIP` sind optional. Wenn sie definiert werden, verhindert dies jedoch, dass das `Start-EKSBootstrap.ps1`-Skript einen `describeCluster`-Aufruf durchführt.

- Das einzige erforderliche Argument ist der Clustername (*`my-cluster`*).
- Um die *`certificate-authority`* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus.

```
aws eks describe-cluster --query "cluster.certificateAuthority.data" --output text --name my-cluster --region region-code
```

- Um den *`api-server-endpoint`* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus.

```
aws eks describe-cluster --query "cluster.endpoint" --output text --name my-cluster --region region-code
```

- Der Wert für`--dns-cluster-ip`ist Ihr Service CIDR mit`.10`am Ende. Um das *`service-cidr`* für Ihren Cluster zu überprüfen, führen Sie den folgenden Befehl aus. Wenn der zurückgegebene Wert beispielsweise `ipv4 10.100.0.0/16` ist, ist Ihr Wert *`10.100.0.10`*.

```
aws eks describe-cluster --query "cluster.kubernetesNetworkConfig.serviceIpv4Cidr" --output text --name my-cluster --region region-code
```

- Weitere Argumente finden Sie unter Bootstrap-Skript-Konfigurationsparameter.

> **Note**
>
> Wenn Sie CIDR für den benutzerdefinierten Service verwenden, müssen Sie ihn mithilfe des `-ServiceCIDR`-Parameters angeben. Andernfalls schlägt die DNS-Auflösung für Pods im Cluster fehl.

```
<powershell>
[string]$EKSBootstrapScriptFile = "$env:ProgramFiles\Amazon\EKS\Start-EKSBootstrap.ps1"
& $EKSBootstrapScriptFile -EKSClusterName my-cluster `
  -Base64ClusterCA certificate-authority `
  -APIServerEndpoint api-server-endpoint `
  -DNSClusterIP service-cidr.10
</powershell>
```

Führen Sie ein benutzerdefiniertes AMI aufgrund bestimmter Sicherheits-, Compliance- oder interner Richtlinienanforderungen aus

Weitere Informationen dazu finden Sie unter Amazon Machine Images (AMI) im Amazon-EC2-Benutzerhandbuch. Die Amazon EKS AMI-Build-Spezifikation enthält Ressourcen und Konfigurationsskripts für die Erstellung eines benutzerdefinierten Amazon EKS-AMI auf Basis von Amazon Linux. Weitere Informationen finden Sie unter Amazon-EKS-AMI-Build-Spezifikation auf GitHub. Informationen zum Erstellen benutzerdefinierter AMIs, die mit anderen Betriebssystemen installiert werden, finden Sie Amazon-EKS-Beispiel-AMIs auf GitHub:

> **Important**
>
> Bei der Angabe eines AMI führt Amazon EKS keine Benutzerdaten zusammen. Vielmehr sind Sie verantwortlich für die Bereitstellung der erforderlichen `bootstrap`-Befehle für Knoten, um dem Cluster beizutreten. Wenn Ihre Knoten dem Cluster nicht beitreten können, wird Amazon EKS `CreateNodegroup` und `UpdateNodegroupVersion`-Aktionen ebenfalls fehlschlagen.

Grenzen und Bedingungen bei der Angabe einer AMI-ID

Im Folgenden sind die Grenzen und Bedingungen aufgeführt, die mit der Angabe einer AMI-ID mit verwalteten Knotengruppen verbunden sind:

- Sie müssen eine neue Knotengruppe erstellen, um zwischen der Angabe einer AMI-ID in einer Startvorlage und der Nicht-Angabe einer AMI-ID zu wechseln.
- Sie werden in der Konsole nicht benachrichtigt, wenn eine neuere AMI-Version verfügbar ist. Um Ihre Knotengruppe auf eine neuere AMI-Version zu aktualisieren, müssen Sie eine neue Version Ihrer Startvorlage mit einer aktualisierten AMI-ID erstellen. Dann müssen Sie die Knotengruppe mit der neuen Version der Startvorlage aktualisieren.
- Die folgenden Felder können in der API nicht festgelegt werden, wenn Sie eine AMI-ID angeben:
  - `amiType`
  - `releaseVersion`
  - `version`
- Alle in der API festgelegten `taints` werden asynchron angewendet, wenn Sie eine AMI-ID angeben. Um Taints anzuwenden, bevor ein Knoten mit dem Cluster verbunden wird, müssen Sie die Taints über das Befehlszeilen-Flag `--register-with-taints` an `kubelet` in Ihren Benutzerdaten weitergeben. Weitere Informationen finden Sie unter `kubelet` in der Kubernetes-Dokumentation.
- Wenn Sie eine benutzerdefinierte AMI-ID für Windows verwaltete Knotengruppen angeben, fügen Sie `eks:kube-proxy-windows` diese Ihrer AWS IAM Authenticator-Konfigurationsübersicht hinzu. Dies ist erforderlich, damit DNS ordnungsgemäß funktioniert.

  1. Öffnen Sie die AWS IAM Authenticator-Konfigurationsübersicht zur Bearbeitung.

     ```
     kubectl edit -n kube-system cm aws-auth
     ```

  2. Fügen Sie diesen Eintrag der `groups`-Liste unter jedem `rolearn` hinzu, das Windows-Knoten zugeordnet ist. Ihre Konfigurationszuordnung sollte ähnlich aussehen wie `aws-auth-cm-windows.yaml`.

     ```
     - eks:kube-proxy-windows
     ```

  3. Speichern Sie die Datei und beenden Sie den Text-Editor.

Löschen einer verwalteten Knotengruppe

In diesem Thema wird beschrieben, wie Sie eine verwaltete Amazon-EKS-Knotengruppe löschen. Wenn Sie eine verwaltete Knotengruppe löschen, legt Amazon EKS zunächst die minimale, maximale

und gewünschte Größe Ihrer Auto-Scaling-Gruppe auf Null fest. Dies bewirkt dann, dass Ihre Knotengruppe abskaliert wird.

Bevor jede Instance beendet wird, sendet Amazon EKS ein Signal, um die Pods von diesem Knoten zu entleeren. Wenn die Pods nach einigen Minuten nicht abgelassen wurden, lässt Amazon EKS Auto Scaling die Beendigung der Instance fortsetzen. Nachdem jede Instance beendet wurde, wird die Auto-Scaling-Gruppe gelöscht.

> **Important**
>
> Wenn Sie eine verwaltete Knotengruppe löschen, die eine Knoten-IAM-Rolle verwendet, die von keiner anderen verwalteten Knotengruppe im Cluster genutzt wird, wird die Rolle aus `aws-auth ConfigMap` entfernt. Wenn eine der selbstverwalteten Knotengruppen im Cluster dieselbe Knoten-IAM-Rolle verwendet, werden die selbstverwalteten Knoten in den `NotReady`-Status verschoben. Außerdem wird auch der Clusterbetrieb unterbrochen. Informationen dazu, wie Sie eine Zuordnung für die Rolle hinzuzufügen, die Sie nur für die selbstverwalteten Knotengruppen verwenden, finden Sie unter Erstellen von Zugriffseinträgen, sofern die Plattformversion Ihres Clusters mindestens der Mindestversion entspricht, die unter Zugangseinträge verwalten im Abschnitt mit den Voraussetzungen angegeben ist. Wenn Ihre Plattformversion älter ist als die erforderliche Mindestversion für Zugriffseinträge, können Sie den Eintrag wieder `aws-auth ConfigMap` hinzufügen. Weitere Informationen erhalten Sie durch Eingabe von `eksctl create iamidentitymapping --help` im Terminal.

Sie können eine verwaltete Knotengruppe mit `eksctl` oder AWS Management Console löschen.

eksctl

So löschen Sie eine verwaltete Knotengruppe mit **`eksctl`**

Geben Sie den folgenden Befehl ein. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte.

```
eksctl delete nodegroup \
  --cluster my-cluster \
  --name my-mng \
  --region region-code
```

Weitere Optionen finden Sie unter Löschen und Löschen von Knotengruppen in der `eksctl`-Dokumentation.

AWS Management Console

So löschen Sie Ihre verwaltete Knotengruppe mit der AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie auf der Seite Clusters den Cluster aus, der die zu löschende Knotengruppe enthält.
3. Wählen Sie auf der ausgewählten Cluster-Seite die Registerkarte Datenverarbeitung aus.
4. Wählen Sie im Abschnitt Node groups (Knotengruppen) die zu löschende Knotengruppe aus. Wählen Sie dann Löschen.
5. Geben Sie im Bestätigungsdialogfeld Knotengruppe löschen den Namen der Knotengruppe ein. Wählen Sie dann Löschen.

AWS CLI

So löschen Sie Ihre verwaltete Knotengruppe mit der AWS CLI

1. Geben Sie den folgenden Befehl ein. Ersetzen Sie jede *example value* durch Ihre eigenen Werte.

```
aws eks delete-nodegroup \
  --cluster-name my-cluster \
  --nodegroup-name my-mng \
  --region region-code
```

2. Verwenden Sie die Pfeiltasten auf Ihrer Tastatur, um durch die Antwortausgabe zu blättern. Drücken Sie die **q**-Taste, wenn Sie fertig sind.

Weitere Optionen finden Sie unter dem Befehl `delete-nodegroup` in der AWS CLI - Befehlsreferenz

Selbstverwaltete Knoten

Ein Cluster enthält einen oder mehrere Amazon-EC2-Knoten, auf denen Pods geplant sind. Amazon EKS-Knoten werden in Ihrem AWS Konto ausgeführt und stellen über den Cluster-API-Serverendpunkt eine Verbindung zur Steuerungsebene Ihres Clusters her. Sie werden basierend auf Amazon-EC2-Preisen in Rechnung gestellt. Weitere Informationen dazu finden Sie unter Amazon EC2 – Preise.

Ein Cluster kann mehrere Knotengruppen enthalten. Jede Knotengruppe enthält eine oder mehrere Knoten, die in einer Amazon EC2 Auto Scaling -Gruppe bereitgestellt werden. Der Instance-Typ der Knoten innerhalb der Gruppe kann variieren, z. B. wenn die attributbasierte Instance-Typauswahl mit Karpenter verwendet wird. Alle Instances in einer Knotengruppe müssen die Amazon-EKS-Knoten-IAM-Rolle verwenden.

Amazon EKS bietet spezialisierte Amazon Machine Images (AMIs), die als für Amazon EKS optimierte AMIs bezeichnet werden. Die AMIs sind so konfiguriert, dass sie mit Amazon EKS funktionieren. Zu ihren Komponenten gehören `containerd`, `kubelet`, und der AWS IAM Authenticator. Das AMI enthält auch ein spezialisiertes Bootstrap-Skript, mit dem Sie die Steuerebene Ihres Clusters automatisch erkennen und eine Verbindung damit herstellen können.

Wenn Sie den Zugriff auf den öffentlichen Endpunkt Ihres Clusters mithilfe von CIDR-Blöcken einschränken, empfehlen wir, dass Sie auch den privaten Endpunktzugriff aktivieren. Auf diese Weise können Knoten mit dem Cluster kommunizieren. Wenn der private Endpunkt nicht aktiviert ist, müssen die CIDR-Blöcke, die Sie für den öffentlichen Zugriff angeben, die Ausgangsquellen aus Ihrer VPC enthalten. Weitere Informationen dazu finden Sie unter Zugriffskontrolle für den Amazon-EKS-Cluster-Endpunkt.

Informationen zum Hinzufügen selbstverwalteter Knoten zu Ihrem Amazon-EKS-Cluster finden Sie in den folgenden Themen. Wenn Sie selbstverwaltete Knoten manuell starten, fügen Sie jedem Knoten die folgende Markierung hinzu. Weitere Informationen dazu finden Sie unter Hinzufügen und Löschen von Markierungen für einzelne Ressourcen. Wenn Sie die Schritte in der Anleitung ausführen, wird die erforderliche Markierung für Sie zum Knoten hinzugefügt.

Schlüssel	Wert
`kubernetes.io/cluster/ my-cluster`	`owned`

Weitere Informationen zu Knoten aus einer allgemeinen Kubernetes-Perspektive finden Sie unter Nodes (Knoten) in der Kubernetes-Dokumentation.

Themen

- Starten selbstverwalteter Amazon Linux-Knoten
- Starten selbstverwalteter Bottlerocket-Knoten
- Starten selbstverwalteter Windows-Knoten
- Starten selbstverwalteter Ubuntu-Knoten
- Aktualisierungen des selbstverwalteten Worker-Knotens

Starten selbstverwalteter Amazon Linux-Knoten

Dieses Thema beschreibt Hinweise zum Starten von Auto-Scaling-Gruppen von Linux-Knoten, die mit Ihrem Amazon-EKS-Cluster registriert sind. Nachdem die Knoten dem Cluster beigetreten sind, können Sie Kubernetes-Anwendungen darin bereitstellen. Sie können auch selbstverwaltete Amazon Linux-Knoten mit `eksctl` oder dem AWS Management Console starten. Wenn Sie Knoten auf starten müssen, finden Sie weitere Informationen AWS Outposts unterStarten selbstverwalteter Amazon Linux-Knoten auf einem Outpost.

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erstellen eines Amazon-EKS-Clusters. Wenn Sie Subnetze in dem Bereich haben AWS Outposts AWS Wavelength, in AWS-Region dem Sie AWS Local Zones aktiviert haben, dürfen diese Subnetze bei der Erstellung Ihres Clusters nicht übergeben worden sein.
- Eine vorhandene IAM-Rolle, die von den Knoten verwendet werden soll. Informationen zum Erstellen finden Sie unter Amazon-EKS-Knoten-IAM-Rolle. Wenn diese Rolle keine der Richtlinien für das VPC-CNI hat, ist die folgende separate Rolle für die VPC-CNI-Pods erforderlich.
- (Optional, aber empfohlen) Das Amazon VPC CNI plugin for Kubernetes-Add-on wurde mit einer eigenen IAM-Rolle konfiguriert, an die die erforderliche IAM-Richtlinie angehängt ist. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).
- Kenntnisse über die in Auswählen eines Amazon-EC2-Instance-Typs aufgelisteten Überlegungen. Je nachdem, welchen Instance-Typ Sie wählen, kann es zusätzliche Voraussetzungen für Ihren Cluster und Ihre VPC geben.

eksctl

> **Note**
>
> `eksctl`unterstützt Amazon Linux 2023 derzeit nicht.

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

So starten Sie selbstverwaltete Linux-Knoten mit **`eksctl`**

1. (Optional) Wenn die von verwaltete AmazonEKS_CNI_Policy IAM-Richtlinie an Ihre Amazon-EKS-Knoten-IAM-Rolle angefügt ist, sollten Sie sie stattdessen einer IAM-Rolle zuzuweisen, die Sie dem `aws-node`-Servicekonto für Kubernetes zuordnen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).
2. Der folgende Befehl erstellt eine Knotengruppe in einem bestehenden Cluster. Ersetzen Sie *`al-nodes`* durch einen Namen für Ihre Knotengruppe. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein. Ersetzen Sie den Rest der *`example value`* durch Ihre eigenen Werte. Die Knoten werden standardmäßig mit derselben Kubernetes-Version wie die Steuerebene erstellt.

   Bevor Sie einen Wert für `--node-type` wählen, sehen Sie sich Auswählen eines Amazon-EC2-Instance-Typs an.

   Ersetzen Sie *`my-key`* mit dem Namen Ihres Amazon-EC2-Schlüsselpaars oder öffentlichen Schlüssels. Dieser Schlüssel wird für den SSH-Zugriff zu Ihren Knoten verwendet, nachdem diese gestartet wurden. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie

eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Schlüsselpaare im Amazon-EC2-Benutzerhandbuch.

Erstellen Sie Ihre Knoten-Gruppe mit dem folgenden Befehl.

> **Important**
>
> Wenn Sie eine Knotengruppe in Subnetzen AWS Outposts, Wellenlängensubnetzen oder lokalen Zonensubnetzen bereitstellen möchten, müssen Sie zusätzliche Überlegungen berücksichtigen:
>
> - Die Subnetze dürfen beim Erstellen des Clusters nicht übergeben worden sein.
> - Sie müssen die Knotengruppe mit einer Konfigurationsdatei erstellen, die die Subnetze und `volumeType: gp2` angibt. Weitere Informationen dazu finden Sie unter Verwenden von Config-Dateien und im Config-Datei-Schema in der `eksctl`-Dokumentation.

```
eksctl create nodegroup \
  --cluster my-cluster \
  --name al-nodes \
  --node-type t3.medium \
  --nodes 3 \
  --nodes-min 1 \
  --nodes-max 4 \
  --ssh-access \
  --managed=false \
  --ssh-public-key my-key
```

Zum Bereitstellen einer Knotengruppe, die:

- Pods eine deutlich höhere Anzahl von IP-Adressen zuweisen kann als die Standardkonfiguration, finden Sie Informationen unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten;
- Pods `IPv4`-Adressen aus einem anderen CIDR-Block als dem der Instance zuweisen kann, finden Sie Informationen unter Benutzerdefinierte Netzwerke für Pods;
- Pods und Services `IPv6`-Adressen zuweisen kann, finden Sie Informationen unter `IPv6`Adressen für ClusterPods, und services.

- die die `containerd`-Laufzeit verwendet, müssen Sie die Knotengruppe mithilfe einer `config`-Datei bereitstellen; Weitere Informationen finden Sie unter Testen Sie die Migration von Docker nach `containerd`.
- keinen ausgehenden Internetzugriff hat, finden Sie Informationen unter Anforderungen an private Cluster.

Geben Sie den folgenden Befehl ein, um eine vollständige Liste aller verfügbaren Optionen und Standardwerte anzuzeigen.

```
eksctl create nodegroup --help
```

Wenn Arbeitsknoten dem Cluster nicht beitreten können, finden Sie weitere Informationen unter Knoten können nicht mit dem Cluster verknüpft werden im Handbuch zur Fehlerbehebung.

Eine Beispielausgabe sieht wie folgt aus. Mehrere Zeilen werden ausgegeben, während die Knoten erstellt werden. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Optional) Stellen Sie eine Beispielanwendung bereit, um Ihren Cluster und Ihre Linux-Worker-Knoten zu testen.
4. Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

   - Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
   - Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

   Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

AWS Management Console

Schritt 1: Um selbstverwaltete Linux-Knoten mit dem AWS Management Console zu starten

1. Laden Sie die neueste Version der Vorlage herunter. AWS CloudFormation

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2022-12-23/amazon-eks-nodegroup.yaml
   ```

2. Warten Sie, bis der Status des Clusters als `ACTIVE` angezeigt wird. Wenn Sie Ihre Knoten starten, bevor der Cluster aktiv ist, werden die Knoten nicht mit dem Cluster registriert und Sie müssen sie neu starten.
3. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
4. Wählen Sie Create stack (Stack erstellen) und dann With new resources (standard) (Mit neuen Ressourcen [Standard]) aus.
5. Wählen Sie für Specify template (Vorlage festlegen) Upload a template file (Vorlagendatei hochladen) aus und wählen Sie dann Choose file (Datei wählen).
6. Wählen Sie die heruntergeladene `amazon-eks-nodegroup.yaml`-Datei aus.
7. Klicken Sie auf Weiter.
8. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) die folgenden Parameter ein und klicken Sie dann auf Next (Weiter):

   - Stack name (Stack-Name): Wählen Sie einen Stack-Namen für Ihren AWS CloudFormation -Stack aus. Sie können ihn beispielsweise ***`my-cluster-nodes`*** nennen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
   - ClusterName: Geben Sie den Namen ein, den Sie bei der Erstellung Ihres Amazon EKS-Clusters verwendet haben. Dieser Name muss exakt mit dem Clusternamen übereinstimmen, andernfalls werden Ihre Knoten dem Cluster nicht beitreten.
   - ClusterControlPlaneSecurityGruppe: Wählen Sie den SecurityGroupsWert aus der AWS CloudFormation Ausgabe aus, die Sie bei der Erstellung Ihrer VPC generiert haben.

     Die folgenden Schritte zeigen einen Vorgang zum Abrufen der entsprechenden Gruppe.

1. Öffnen Sie die Amazon-EKS-Konsole unterhttps://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters.
3. Wählen Sie die Registerkarte Network (Network) aus.
4. Verwenden Sie den Wert Zusätzliche Sicherheitsgruppen als Referenz, wenn Sie aus der Dropdownliste ClusterControlPlaneSecurityGruppe auswählen.

- NodeGroupName: Geben Sie einen Namen für Ihre Knotengruppe ein. Dieser Name kann zu einem späteren Zeitpunkt zum Identifizieren der Auto-Scaling-Knotengruppe verwendet werden, die für Ihre Knoten erstellt wurde. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten.
- NodeAutoScalingGroupMinSize: Geben Sie die Mindestanzahl von Knoten ein, auf die Ihre Auto Scaling Scaling-Gruppe für Knoten skalieren kann.
- NodeAutoScalingGroupDesiredCapacity: Geben Sie die gewünschte Anzahl von Knoten ein, auf die bei der Erstellung Ihres Stacks skaliert werden soll.
- NodeAutoScalingGroupMaxSize: Geben Sie die maximale Anzahl von Knoten ein, auf die Ihre Auto Scaling Scaling-Gruppe für Knoten skalieren kann.
- NodeInstanceTyp: Wählen Sie einen Instance-Typ für Ihre Knoten. Weitere Informationen finden Sie unter Auswählen eines Amazon-EC2-Instance-Typs.
- NodeImageidssmParam: Vorab mit dem Amazon EC2 Systems Manager Manager-Parameter eines kürzlich für Amazon EKS optimierten AMI für eine variable Version gefüllt. Kubernetes Um eine andere Kubernetes-Nebenversion zu verwenden, die von Amazon EKS unterstützt wird, ersetzen Sie *1.XX* durch eine andere unterstützte Version. Wir empfehlen, dieselbe Kubernetes-Version wie Ihr Cluster anzugeben.

  Sie können es auch *amazon-linux-2* durch einen anderen AMI-Typ ersetzen. Weitere Informationen finden Sie unter Abrufen von Amazon-EKS-optimierten Amazon-Linux-AMI-IDs.

  > **Note**
  >
  > Das Amazon EKS Node AMI basiert auf Amazon Linux. Sie können Sicherheits- oder Datenschutzereignisse für Amazon Linux 2 im Amazon Linux-Sicherheitszentrum verfolgen oder den zugehörigen RSS-Feed abonnieren. Sicherheits- oder Datenschutzereignisse enthalten eine Übersicht über das

Problem, welche Pakete betroffen sind und wie Sie Ihre Instances aktualisieren, um das Problem zu beheben.

- NodeImageID: (Optional) Wenn Sie Ihr eigenes benutzerdefiniertes AMI (anstelle des für Amazon EKS optimierten AMI) verwenden, geben Sie eine Knoten-AMI-ID für Ihr AWS-Region. Wenn Sie hier einen Wert angeben, überschreibt dieser alle Werte im Feld NodeImageidsSMParam.
- NodeVolumeGröße: Geben Sie eine Root-Volume-Größe für Ihre Knoten in GiB an.
- NodeVolumeTyp: Geben Sie einen Root-Volume-Typ für Ihre Knoten an.
- KeyName: Geben Sie den Namen eines Amazon EC2 SSH-Schlüsselpaars ein, mit dem Sie sich nach dem Start über SSH mit Ihren Knoten verbinden können. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Schlüsselpaare im Amazon-EC2-Benutzerhandbuch.

  > **Note**
  >
  > Wenn Sie hier kein key pair angeben, schlägt die AWS CloudFormation Stack-Erstellung fehl.

- BootstrapArguments: Geben Sie alle optionalen Argumente an, die an das Node-Bootstrap-Skript übergeben werden sollen, z. B. zusätzliche `kubelet` Argumente. Weitere Informationen finden Sie in den Bootstrap-Skript-Nutzungsinformationen auf GitHub.

  Zum Bereitstellen einer Knotengruppe, die:

  - Pods eine deutlich höhere Anzahl von IP-Adressen zuweisen kann als die Standardkonfiguration, finden Sie Informationen unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten;
  - Pods `IPv4`-Adressen aus einem anderen CIDR-Block als dem der Instance zuweisen kann, finden Sie Informationen unter Benutzerdefinierte Netzwerke für Pods;
  - Pods und Services `IPv6`-Adressen zuweisen kann, finden Sie Informationen unter `IPv6`Adressen für ClusterPods, und services.
  - die die `containerd`-Laufzeit verwendet, müssen Sie die Knotengruppe mithilfe einer `config`-Datei bereitstellen; Weitere Informationen finden Sie unter Testen Sie die Migration von Docker nach `containerd`.

  - keinen ausgehenden Internetzugriff hat, finden Sie Informationen unter Anforderungen an private Cluster.
- DisableIMDSv1: Standardmäßig unterstützt jeder Knoten die Instance-Metadaten-Service-Version 1 (IMDSv1) und IMDSv2. Sie können IMDSv1 deaktivieren. Um zu verhindern, dass künftige Knoten und Pods in der Knotengruppe MDSv1 verwenden, legen Sie DisableIMDSv1 auf true fest. Weitere Informationen finden Sie unter Konfiguration des Instance-Metadatenservice. Weitere Informationen zum Einschränken des Zugriffs darauf auf Ihre Knoten finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.
- VpcId: Geben Sie die ID für die VPC ein, die Sie erstellt haben.
- Subnetze: Wählen Sie die Subnetze aus, die Sie für Ihre VPC erstellt haben. Wenn Sie Ihre VPC anhand der unter Erstellen einer VPC für Ihren Amazon-EKS-Cluster beschriebenen Schritte erstellt haben, geben Sie nur die privaten Subnetze innerhalb der VPC an, in denen Ihre Knoten gestartet werden sollen. Sie können sehen, welche Subnetze privat sind, indem Sie den jeweiligen Subnetzlink in der Registerkarte Networking (Netzwerk) Ihres Clusters öffnen.

  > **Important**
  >
  > - Wenn es sich bei einem oder einigen der Subnetze um öffentliche Subnetze handelt, muss die Einstellung für die automatische Zuweisung öffentlicher IP-Adressen aktiviert sein. Wenn die Einstellung für das öffentliche Subnetz nicht aktiviert ist, wird allen Knoten, die Sie in diesem öffentlichen Subnetz bereitstellen, keine öffentliche IP-Adresse zugewiesen und sie können nicht mit dem Cluster oder anderen Diensten kommunizieren. AWS Wenn das Subnetz vor dem 26. März 2020 mithilfe einer der Amazon AWS CloudFormation EKS-VPC-Vorlagen oder mithilfe `eksctl` von bereitgestellt wurde, ist die automatische Zuweisung öffentlicher IP-Adressen für öffentliche Subnetze deaktiviert. Informationen zum Aktivieren der öffentlichen IP-Adresszuweisung für ein Subnetz finden Sie unter Ändern des öffentlichen `IPv4`-Adressattributs für Ihr Subnetz. Wenn der Knoten in einem privaten Subnetz bereitgestellt wird, kann er über ein NAT-Gateway mit dem Cluster und anderen AWS Diensten kommunizieren.
  > - Wenn die Subnetze keinen Internetzugang haben, stellen Sie sicher, dass Sie die Überlegungen und zusätzlichen Schritte in Anforderungen an private Cluster kennen.

- Wenn Sie Wellenlängen-Subnetze oder Local Zone-Subnetze auswählen AWS Outposts, dürfen die Subnetze bei der Erstellung des Clusters nicht übergeben worden sein.

9. Treffen Sie die gewünschte Auswahl auf der Seite Configure stack options (Stackoptionen konfigurieren) und wählen Sie dann Next (Weiter) aus.
10. Aktivieren Sie das Kontrollkästchen links neben I acknowledge that AWS CloudFormation might create IAM resources with custom names („Mir ist bewusst, dass IAM-Ressourcen mit eigenen Namen erstellen kann“) und wählen Sie dann Create Stack (Stack erstellen) aus.
11. Wenn Ihr Stack fertig erstellt wurde, wählen Sie ihn in der Konsole aus und klicken Sie auf Outputs (Ausgänge).
12. Notieren Sie sich die NodeInstanceRolle für die Knotengruppe, die erstellt wurde. Sie benötigen diese, wenn Sie Ihre Amazon-EKS--Arbeitsknoten konfigurieren.

Schritt 2: So aktivieren Sie die Knoten, die Ihrem Cluster beitreten sollen

> **Note**
>
> Wenn Sie Knoten innerhalb einer privaten VPC ohne ausgehenden Internetzugang gestartet haben, müssen Sie ihnen ermöglichen, Ihrem Cluster innerhalb der VPC beizutreten.

1. Überprüfen Sie, ob Sie bereits über eine `aws-auth` `ConfigMap` verfügen.

```
kubectl describe configmap -n kube-system aws-auth
```

2. Wenn eine `aws-auth` `ConfigMap` angezeigt wird, aktualisieren Sie sie nach Bedarf.

   a. Öffnen Sie `ConfigMap` zum Bearbeiten.

   ```
   kubectl edit -n kube-system configmap/aws-auth
   ```

   b. Fügen Sie nach Bedarf einen neuen `mapRoles`-Eintrag hinzu. Setzen Sie den `rolearn` Wert auf den NodeInstanceRollenwert, den Sie im vorherigen Verfahren aufgezeichnet haben.

```
[...]
data:
  mapRoles: |
    - rolearn: <ARN of instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
[...]
```

c. Speichern Sie die Datei und beenden Sie den Text-Editor.

3. Wenn die Fehlermeldung `Error from server (NotFound): configmaps "aws-auth" not found` angezeigt wird, wenden Sie die standardmäßige `ConfigMap` an.

   a. Laden Sie die Konfigurationszuordnung herunter.

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/aws-auth-cm.yaml
   ```

   b. Legen Sie in der `aws-auth-cm.yaml` Datei den `rolearn` Wert auf den NodeInstanceRollenwert fest, den Sie im vorherigen Verfahren aufgezeichnet haben. Hierzu können Sie einen Texteditor verwenden oder *`my-node-instance-role`* ersetzen und den folgenden Befehl ausführen:

   ```
   sed -i.bak -e 's|<ARN of instance role (not instance profile)>|my-node-instance-role|' aws-auth-cm.yaml
   ```

   c. Wenden Sie die Konfiguration an. Die Ausführung dieses Befehls kann einige Minuten dauern.

   ```
   kubectl apply -f aws-auth-cm.yaml
   ```

4. Sehen Sie sich den Status Ihrer Knoten an und warten Sie, bis diese in den `Ready`-Status eintreten.

   ```
   kubectl get nodes --watch
   ```

   Geben Sie `Ctrl+C` ein, um zu einer Shell-Eingabeaufforderung zurückzukehren.

> **Note**
>
> Wenn Sie Autorisierungs- oder Ressourcenfehler erhalten, finden Sie weitere Informationen unter Nicht autorisiert oder Zugriff verweigert (`kubectl`) im Thema zur Fehlerbehebung.

Wenn Knoten dem Cluster nicht beitreten können, finden Sie weitere Informationen unter Knoten können nicht mit dem Cluster verknüpft werden im Handbuch zur Fehlerbehebung.

5. (Nur GPU-Knoten) Wenn Sie einen GPU-Instance-Typ und das mit Amazon EKS optimierte beschleunigte AMI gewählt haben, müssen Sie das NVIDIA-Geräte-Plugin für Kubernetes mit dem folgenden Befehl als DaemonSet auf Ihren Cluster anwenden. Ersetzen Sie *vX.X.X* mit Ihrer gewünschten Nvidia/K8S-Geräte-Plugin-Version, bevor Sie den folgenden Befehl ausführen.

```
kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
```

Schritt 3: Zusätzliche Aktionen

1. (Optional) Stellen Sie eine Beispielanwendung bereit, um Ihren Cluster und Ihre Linux-Worker-Knoten zu testen.
2. (Optional) Wenn die von AmazonEKS_CNI_Policy verwaltete IAM-Richtlinie (bei einem `IPv4`-Cluster) oder die *AmazonEKS_CNI_IPv6_Policy* (die Sie bei einem `IPv6`-Cluster selbst erstellt haben) an Ihre the section called “Knoten-IAM-Rolle” angefügt ist, sollten Sie sie stattdessen einer IAM-Rolle zuweisen, die Sie dem Kubernetes-Servicekonto für `aws-node` zuordnen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).
3. Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

   - Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
   - Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Kapazitätsblöcke für ML

> **Important**
>
> - Kapazitätsblöcke sind nur für bestimmte Amazon EC2 EC2-Instance-Typen und AWS-Regionen verfügbar. Informationen zur Kompatibilität finden Sie unter Voraussetzungen für das Arbeiten mit Kapazitätsblöcken im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.
> - Kapazitätsblöcke können derzeit nicht mit von Amazon EKS verwalteten Knotengruppen oder verwendet werdenKarpenter.

Kapazitätsblöcke für Machine Learning (ML) ermöglichen es Ihnen, GPU-Instances zu einem zukünftigen Zeitpunkt zu reservieren, um Ihre ML-Workloads mit kurzer Dauer zu unterstützen. Instances, die innerhalb eines Capacity Blocks ausgeführt werden, werden innerhalb von Amazon EC2 automatisch nahe beieinander platziert UltraClusters, sodass keine Cluster-Platzierungsgruppe verwendet werden muss. Weitere Informationen finden Sie unter Capacity Blocks for ML im Amazon EC2 EC2-Benutzerhandbuch für Linux-Instances.

Sie können Kapazitätsblöcke mit Amazon EKS für die Bereitstellung und Skalierung Ihrer selbstverwalteten Knoten verwenden. Die folgenden Schritte geben einen allgemeinen Beispielüberblick.

1. Erstellen Sie eine Startvorlage in AWS Management Console. Weitere Informationen finden Sie unter Verwenden von Kapazitätsblöcken für maschinelles Lernen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

   Stellen Sie sicher, dass Sie die Konfiguration des Instance-Typs und des Amazon Machine Image (AMI) angeben.
2. Verknüpfen Sie den Kapazitätsblock mithilfe der Kapazitätsreservierungs-ID mit einer Startvorlage.

Im Folgenden finden Sie eine AWS CloudFormation Beispielvorlage zum Erstellen einer Startvorlage für einen Kapazitätsblock:

```
NodeLaunchTemplate:
    Type: "AWS::EC2::LaunchTemplate"
    Properties:
      LaunchTemplateData:
        InstanceMarketOptions:
          MarketType: "capacity-block"
        CapacityReservationSpecification:
          CapacityReservationTarget:
            CapacityReservationId: "cr-02168da1478b509e0"
        IamInstanceProfile:
          Arn: iam-instance-profile-arn
        ImageId: image-id
        InstanceType: p5.48xlarge
        KeyName: key-name
        SecurityGroupIds:
        - sg-05b1d815d1EXAMPLE
        UserData: user-data
```

Sie müssen das Subnetz in der Availability Zone übergeben, in der die Reservierung vorgenommen wurde, da Kapazitätsblöcke zonal sind.

3. Wenn Sie die selbstverwaltete Knotengruppe erstellen, bevor die Kapazitätsreservierung aktiv wird, legen Sie die gewünschte Kapazität auf `0` fest. Achten Sie beim Erstellen der Knotengruppe darauf, dass Sie nur das entsprechende Subnetz für die Availability Zone angeben, in der die Kapazität reserviert ist.

   Im Folgenden finden Sie eine CloudFormation Beispielvorlage, die verwendet werden kann. In diesem Beispiel werden das `LaunchTemplateId` und das `Version` der im vorherigen Beispiel gezeigten `AWS::Amazon EC2::LaunchTemplate`-Ressource ermittelt. Sie erhält auch die Werte für `DesiredCapacity`, `MaxSize`, `MinSize`, und `VPCZoneIdentifier`, die an anderer Stelle in derselben Vorlage deklariert sind.

   ```
   NodeGroup:
     Type: "AWS::AutoScaling::AutoScalingGroup"
     Properties:
       DesiredCapacity: !Ref NodeAutoScalingGroupDesiredCapacity
       LaunchTemplate:
         LaunchTemplateId: !Ref NodeLaunchTemplate
   ```

```
        Version: !GetAtt NodeLaunchTemplate.LatestVersionNumber
      MaxSize: !Ref NodeAutoScalingGroupMaxSize
      MinSize: !Ref NodeAutoScalingGroupMinSize
      VPCZoneIdentifier: !Ref Subnets
      Tags:
        - Key: Name
          PropagateAtLaunch: true
          Value: !Sub ${ClusterName}-${NodeGroupName}-Node
        - Key: !Sub kubernetes.io/cluster/${ClusterName}
          PropagateAtLaunch: true
          Value: owned
```

4. Nachdem die Knotengruppe erfolgreich erstellt wurde, stellen Sie sicher, dass Sie die `NodeInstanceRole` für die Knotengruppe aufzeichnen, die erstellt wurde. Sie benötigen dies, um sicherzustellen, dass beim Skalieren der Knotengruppe die neuen Knoten dem Cluster beitreten und Kubernetes die Knoten erkennen kann. Weitere Informationen finden Sie in der Anleitung AWS Management Console in Starten selbstverwalteter Amazon Linux-Knoten.
5. Wir empfehlen Ihnen, eine geplante Skalierungsrichtlinie für die Auto-Scaling-Gruppe zu erstellen, die sich an den Reservierungszeiten für Kapazitätsblöcke orientiert. Weitere Informationen finden Sie unter Geplante Skalierung für Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling-Benutzerhandbuch.

   Sie können alle von Ihnen reservierten Instances bis 30 Minuten vor Ablauf der Endzeit des Kapazitätsblocks verwenden. Instances, die zu diesem Zeitpunkt noch laufen, werden beendet. Um genügend Zeit zu haben, den/die Knoten ordnungsgemäß zu entleeren, empfehlen wir, die Skalierung so zu planen, dass sie mehr als 30 Minuten vor der Endzeit der Kapazitätsblock-Reservierung auf Null skaliert.

   Wenn Sie stattdessen manuell hochskalieren wollen, wenn die Kapazitätsreservierung `Active` wird, müssen Sie die gewünschte Kapazität der Auto-Scaling-Gruppe zum Startzeitpunkt der Kapazitätsblockreservierung aktualisieren. Dann müssten Sie auch mehr als 30 Minuten vor dem Ende der Kapazitätsblock-Reservierung manuell herunterskalieren.
6. Die Knotengruppe ist jetzt bereit für Workloads und Pods muss geplant werden.
7. Damit Sie Pods ordnungsgemäß ausgelaugt werden, empfehlen wir Ihnen, AWS Node Termination Handler einzurichten. Dieser Handler kann mithilfe von Amazon EC2 Auto Scaling nach „ASG Scale-in“ -Lebenszyklusereignissen Ausschau halten EventBridge und es der Kubernetes Kontrollebene ermöglichen, die erforderlichen Maßnahmen zu ergreifen, bevor die Instance nicht verfügbar ist. Andernfalls bleiben Ihre Pods- und Kubernetes-Objekte in einem

ausstehenden Zustand stecken. Weitere Informationen finden Sie unter AWS Node Termination Handler on. GitHub

Wenn Sie keinen Node Termination Handler einrichten, empfehlen wir Ihnen, mit der manuellen Entleerung Ihrer Pods zu beginnen, bevor das 30-Minuten-Fenster erreicht ist, damit sie genügend Zeit haben, um ordnungsgemäß entleert zu werden.

Starten selbstverwalteter Bottlerocket-Knoten

> **Note**
>
> Verwaltete Knotengruppen bieten möglicherweise einige Vorteile für Ihren Anwendungsfall. Weitere Informationen finden Sie unter Verwaltete Knotengruppen.

In diesem Thema wird beschrieben, wie Sie Auto Scaling Scaling-Gruppen von Bottlerocket-Knoten starten, die sich bei Ihrem Amazon EKS-Cluster registrieren. Bottlerocketist ein Linux Open-Source-Betriebssystem, das Sie für AWS die Ausführung von Containern auf virtuellen Maschinen oder Bare-Metal-Hosts verwenden können. Nachdem die Knoten dem Cluster beigetreten sind, können Sie Kubernetes-Anwendungen darin bereitstellen. Weitere Informationen zu Bottlerocket finden Sie unter Verwenden eines Bottlerocket-AMI mit Amazon EKS auf GitHub und Benutzerdefinierter AMI-Support in der `eksctl`-Dokumentation.

Weitere Informationen zu direkten Upgrades finden Sie unter Update-Operator für Bottlerocket auf GitHub.

> **Important**
>
> - Amazon-EKS-Worker-Knoten sind Standard-Amazon-EC2-Instances und werden Ihnen basierend auf normalen Amazon-EC2-Instance-Preisen berechnet. Weitere Informationen dazu finden Sie unter Amazon EC2 – Preise.
> - Sie können Bottlerocket-Knoten in erweiterten Amazon EKS-Clustern auf AWS Outposts starten, aber Sie können sie nicht in lokalen Clustern auf Outposts starten. AWS Weitere Informationen finden Sie unter auf Amazon EKSAWS Outposts.
> - Sie können auf Amazon-EC2-Instances mit x86- oder Arm-Prozessoren bereitstellen. Sie können jedoch nicht auf Instances bereitstellen, die über Inferentia-Chips verfügen.

- Bottlerocketist AWS CloudFormation kompatibel mit. Es gibt jedoch keine offizielle CloudFormation Vorlage, die kopiert werden kann, um Bottlerocket Knoten für Amazon EKS bereitzustellen.
- Bottlerocket-Images werden nicht mit einem SSH-Server oder einer Shell geliefert. Sie können out-of-band Zugriffsmethoden verwenden, um die SSH Aktivierung des Admin-Containers zu ermöglichen und einige Bootstrapping-Konfigurationsschritte mit Benutzerdaten zu durchlaufen. Weitere Informationen finden Sie in diesen Abschnitten im bottlerocket README.md auf GitHub:
  - Exploration (Erkundung)
  - Administrator-Container
  - Kubernetes-Einstellungen

Bottlerocket-Knoten mit **`eksctl`** starten

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre -Version mit dem folgenden Befehl überprüfen:

```
eksctl version
```

Eine Installations- und Upgrade-Anleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

> **Note**
>
> Dieses Verfahren funktioniert nur für Cluster, die mit `eksctl` erstellt wurden.

1. Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein. Ersetzen Sie *`ng-bottlerocket`* durch einen Namen für Ihre Knotengruppe. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten. Um die Bereitstellung auf Arm-Instances durchzuführen, ersetzen Sie *`m5.large`* durch einen Arm-

Instance-Typ. Ersetzen Sie *my-ec2-keypair-name* mit dem Namen eines Amazon-EC2-SSH-Schlüsselpaars ein, das Sie für die Verbindung über SSH in Ihre Arbeitsknoten verwenden können, nachdem sie gestartet wurden. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Schlüsselpaare im Amazon-EC2-Benutzerhandbuch. Ersetzen Sie *example values* durch Ihre eigenen Werte. Nachdem Sie das Ersetzen vorgenommen haben, führen Sie den geänderten Befehl aus, um die `bottlerocket.yaml`-Datei zu erstellen.

Wenn Sie einen Arm Amazon-EC2-Instance-Typ angeben, überprüfen Sie die Überlegungen inAmazon-EKS-optimierte Arm-Amazon-Linux-AMIsbevor Sie eine Bereitstellung durchführen. Anweisungen zum Bereitstellen mit einem benutzerdefinierten AMI finden Sie unter Erstellen von Bottlerocket auf GitHub und Benutzerdefinierter AMI-Support in der `eksctl`-Dokumentation. Um eine verwaltete Knotengruppe bereitzustellen, stellen Sie ein benutzerdefiniertes AMI mithilfe einer Startvorlage bereit. Weitere Informationen finden Sie unter Anpassen verwalteter Knoten mit Startvorlagen.

> **Important**
>
> Um eine Knotengruppe in Subnetzen oder AWS lokalen Zonensubnetzen bereitzustellen AWS Outposts AWS Wavelength, übergeben Sie bei der Erstellung des Clusters keine Subnetze der AWS lokalen Zone. AWS Outposts AWS Wavelength Sie müssen die Subnetze im folgenden Beispiel angeben. Weitere Informationen finden Sie unter Verwenden von Config-Dateien und im Config-Datei-Schema in der `eksctl`-Dokumentation. Ersetzen Sie es *region-code* durch AWS-Region das, in dem sich Ihr Cluster befindet.

```
cat >bottlerocket.yaml <<EOF
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code
  version: '1.30'

iam:
  withOIDC: true
```

```
nodeGroups:
  - name: ng-bottlerocket
    instanceType: m5.large
    desiredCapacity: 3
    amiFamily: Bottlerocket
    ami: auto-ssm
    iam:
       attachPolicyARNs:
          - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
          - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
          - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
          - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
    ssh:
        allow: true
        publicKeyName: my-ec2-keypair-name
EOF
```

2. Stellen Sie den Treiber mit dem folgenden Befehl bereit.

```
eksctl create nodegroup --config-file=bottlerocket.yaml
```

Eine Beispielausgabe sieht wie folgt aus.

Mehrere Zeilen werden ausgegeben, während die Knoten erstellt werden. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Optional) Erstellen Sie ein persistentes Kubernetes-Volume auf einem Bottlerocket-Knoten, indem Sie das Amazon-EBS-CSI-Plugin verwenden. Der standardmäßige Amazon-EBS-Treiber basiert auf Dateisystem-Tools, die nicht in Bottlerocket enthalten sind. Weitere Informationen zur Erstellung einer Speicherklasse mit dem Treiber finden Sie unter Amazon-EBS-CSI-Treiber.
4. (Optional) Standardmäßig setzt `kube-proxy` den Kernelparameter `nf_conntrack_max` auf einen Standardwert, der sich von dem unterscheiden kann, was Bottlerocket ursprünglich beim Booten festgelegt hat. Um die Standardeinstellung von Bottlerocket beizubehalten, bearbeiten Sie die `kube-proxy`-Konfiguration im folgenden Befehl.

```
kubectl edit -n kube-system daemonset kube-proxy
```

Fügen Sie `--conntrack-max-per-core` und `--conntrack-min` zu den `kube-proxy`-Argumenten im folgenden Beispiel hinzu. Eine Einstellung von`0`impliziert keine Änderung.

```
containers:
- command:
  - kube-proxy
  - --v=2
  - --config=/var/lib/kube-proxy-config/config
  - --conntrack-max-per-core=0
  - --conntrack-min=0
```

5. (Optional) Stellen Sie eine Beispielanwendung bereit, um Ihre Bottlerocket-Knoten zu testen.
6. Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

   - Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
   - Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

   Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Starten selbstverwalteter Windows-Knoten

Dieses Thema beschreibt Hinweise zum Starten von Auto-Scaling-Gruppen von Windows-Knoten, die mit Ihrem Amazon-EKS-Cluster registriert sind. Nachdem die Knoten dem Cluster beigetreten sind, können Sie Kubernetes-Anwendungen darin bereitstellen.

> **Important**
>
> - Amazon-EKS-Worker-Knoten sind Standard-Amazon-EC2-Instances und werden Ihnen basierend auf normalen Amazon-EC2-Instance-Preisen berechnet. Weitere Informationen dazu finden Sie unter Amazon EC2 – Preise.

- Sie können Windows-Knoten in erweiterten Amazon EKS-Clustern auf AWS Outposts starten, aber Sie können sie nicht in lokalen Clustern auf AWS Outposts starten. Weitere Informationen finden Sie unter auf Amazon EKSAWS Outposts.

Aktivieren Sie den Windows-Support für Ihren Cluster. Es wird empfohlen, wichtige Überlegungen zu berücksichtigen, bevor Sie eine Windows-Knotengruppe starten. Weitere Informationen finden Sie unter Aktivieren des Windows-Supports.

Sie können selbstverwaltete Windows-Knoten mit `eksctl` oder AWS Management Console launchen.

eksctl

So starten Sie selbstverwaltete Windows-Knoten mit **`eksctl`**

Bei diesem Verfahren wird davon ausgegangen, dass Sie `eksctl` installiert haben und dass Ihre `eksctl`-Version mindestens `0.183.0` ist. Sie können Ihre Version mit dem folgenden Befehl überprüfen.

```
eksctl version
```

Eine Installations- und Upgrade-Anleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

> **Note**
>
> Dieses Verfahren funktioniert nur für Cluster, die mit `eksctl` erstellt wurden.

1. (Optional) Wenn die von AmazonEKS_CNI_Policy verwaltete IAM-Richtlinie (bei einem `IPv4`-Cluster) oder die *`AmazonEKS_CNI_IPv6_Policy`* (die Sie bei einem `IPv6`-Cluster selbst erstellt haben) an Ihre the section called “Knoten-IAM-Rolle” angefügt ist, sollten Sie sie stattdessen einer IAM-Rolle zuweisen, die Sie dem Kubernetes-Servicekonto für `aws-node` zuordnen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).
2. Bei diesem Verfahren wird davon ausgegangen, dass Sie einen bestehenden Cluster haben. Wenn Sie noch keinen Amazon EKS-Cluster und keine Amazon Linux-Knotengruppe haben, zu der Sie eine Windows Knotengruppe hinzufügen können, empfehlen wir Ihnen, die Erste

Schritte mit Amazon EKS – `eksctl` Anleitung zu befolgen. Das Handbuch enthält eine vollständige Anleitung zum Erstellen eines Amazon-EKS-Clusters mit Amazon-Linux-Knoten.

Erstellen Sie Ihre Knoten-Gruppe mit dem folgenden Befehl. *`region-code`*Ersetzen Sie es durch AWS-Region das, in dem sich Ihr Cluster befindet. Ersetzen Sie *`my-cluster`* mit Ihrem Clusternamen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein. Ersetzen Sie *`ng-windows`* durch einen Namen für Ihre Knotengruppe. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten. Für Kubernetes-Version `1.24` oder höher können Sie *`2019`* durch `2022` ersetzen, um Windows Server 2022 zu verwenden. Ersetzen Sie den Rest der *`example values`* durch Ihre eigenen Werte.

> **Important**
>
> Um eine Knotengruppe in Subnetzen AWS Outposts AWS Wavelength, oder AWS Local Zone bereitzustellen, übergeben Sie die Subnetze AWS Outposts, Wavelength oder Local Zone nicht, wenn Sie den Cluster erstellen. Erstellen Sie die Knotengruppe mit einer Konfigurationsdatei, in der Sie die Subnetze AWS Outposts, Wavelength oder Local Zone angeben. Weitere Informationen finden Sie unter Verwenden von Config-Dateien und im Config-Datei-Schema in der `eksctl`-Dokumentation.

```
eksctl create nodegroup \
    --region region-code \
    --cluster my-cluster \
    --name ng-windows \
    --node-type t2.large \
    --nodes 3 \
    --nodes-min 1 \
    --nodes-max 4 \
    --managed=false \
    --node-ami-family WindowsServer2019FullContainer
```

> **Note**
>
> - Wenn Arbeitsknoten dem Cluster nicht beitreten können, finden Sie weitere Informationen unter Knoten können nicht mit dem Cluster verknüpft werden im Handbuch zur Fehlerbehebung.
> - Geben Sie den folgenden Befehl ein, um die verfügbaren Optionen für `eksctl`-Befehle anzuzeigen.
>
> ```
> eksctl command -help
> ```

Eine Beispielausgabe sieht wie folgt aus. Mehrere Zeilen werden ausgegeben, während die Knoten erstellt werden. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Optional) Stellen Sie eine Beispielanwendung bereit, um Ihren Cluster und Ihre Windows-Worker-Knoten zu testen.
4. Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

   - Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
   - Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

   Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

AWS Management Console

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster und eine Linux-Knotengruppe. Wenn Sie nicht über diese Ressourcen verfügen, empfehlen wir Ihnen, einem unserer Erste Schritte mit Amazon

EKS-Leitfäden zu folgen, um sie zu erstellen. Diese Leitfäden beschreiben, wie Sie ein Amazon-EKS-Cluster mit Linux-Knoten erstellen.

- Eine vorhandene VPC und eine Sicherheitsgruppe, die die Voraussetzungen für einen Amazon-EKS-Cluster erfüllen. Weitere Informationen erhalten Sie unter Amazon EKS: VPC- und Subnetz-Anforderungen und -Überlegungen und Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe. Der Erste Schritte mit Amazon EKS-Leitfaden erstellt eine VPC, die diese Voraussetzungen erfüllt. Alternativ können Sie auch Erstellen einer VPC für Ihren Amazon-EKS-Cluster folgen, um eine manuell zu erstellen.
- Ein vorhandener Amazon-EKS-Cluster, der eine VPC und eine Sicherheitsgruppe verwendet, die die Voraussetzungen eines Amazon-EKS-Clusters erfüllt. Weitere Informationen finden Sie unter Erstellen eines Amazon-EKS-Clusters. Wenn Sie Subnetze in dem Bereich haben AWS Outposts AWS Wavelength, in AWS-Region dem Sie AWS Local Zones aktiviert haben, dürfen diese Subnetze bei der Erstellung des Clusters nicht übergeben worden sein.

Schritt 1: So starten Sie selbstverwaltete Knoten Windows mit dem AWS Management Console

1. Warten Sie, bis der Status des Clusters als `ACTIVE` angezeigt wird. Wenn Sie Ihre Knoten starten, bevor der Cluster aktiv ist, werden die Knoten nicht mit dem Cluster registriert und Sie müssen sie neu starten.
2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation
3. Wählen Sie Stack erstellen aus.
4. Wählen Sie unter Vorlage angeben die Option Amazon-S3-URL aus.
5. Kopieren Sie die folgende URL und fügen Sie sie in die Amazon S3 URL (Amazon-S3-URL) ein.

   ```
   https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2023-02-09/amazon-eks-windows-nodegroup.yaml
   ```

6. Wählen Sie zweimal Next (Weiter) aus.
7. Füllen Sie auf der Seite Quick create stack (Stack schnell erstellen) die folgenden Parameter entsprechend aus:

   - Stack name (Stack-Name): Wählen Sie einen Stack-Namen für Ihren AWS CloudFormation -Stack aus. Sie können ihn beispielsweise ***my-cluster*-nodes** nennen.

- ClusterName: Geben Sie den Namen ein, den Sie bei der Erstellung Ihres Amazon EKS-Clusters verwendet haben.

  > **Important**
  >
  > Dieser Name muss genau mit dem Namen übereinstimmen, den Sie in Schritt 1: Erstellen Sie Ihre Amazon-EKS-Cluster verwendet haben. Andernfalls können Ihre Knoten dem Cluster nicht beitreten.

- ClusterControlPlaneSecurityGruppe: Wählen Sie die Sicherheitsgruppe aus der AWS CloudFormation Ausgabe aus, die Sie bei der Erstellung Ihrer VPC generiert haben.

  Die folgenden Schritte zeigen eine Methode zum Abrufen der entsprechenden Gruppe.

  1. Öffnen Sie die Amazon-EKS-Konsole unterhttps://console.aws.amazon.com/eks/home#/clusters.
  2. Wählen Sie den Namen des Clusters.
  3. Wählen Sie die Registerkarte Network (Network) aus.
  4. Verwenden Sie den Wert Zusätzliche Sicherheitsgruppen als Referenz, wenn Sie aus der Dropdownliste ClusterControlPlaneSecurityGruppe auswählen.

- NodeGroupName: Geben Sie einen Namen für Ihre Knotengruppe ein. Dieser Name kann zu einem späteren Zeitpunkt zum Identifizieren der Auto-Scaling-Knotengruppe verwendet werden, die für Ihre Knoten erstellt wurde. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten.
- NodeAutoScalingGroupMinSize: Geben Sie die Mindestanzahl von Knoten ein, auf die Ihre Auto Scaling Scaling-Gruppe für Knoten skalieren kann.
- NodeAutoScalingGroupDesiredCapacity: Geben Sie die gewünschte Anzahl von Knoten ein, auf die bei der Erstellung Ihres Stacks skaliert werden soll.
- NodeAutoScalingGroupMaxSize: Geben Sie die maximale Anzahl von Knoten ein, auf die Ihre Auto Scaling Scaling-Gruppe für Knoten skalieren kann.
- NodeInstanceTyp: Wählen Sie einen Instance-Typ für Ihre Knoten. Weitere Informationen finden Sie unter Auswählen eines Amazon-EC2-Instance-Typs.

> **Note**
>
> Die unterstützten Instance-Typen für die neueste Version des Amazon VPC CNI plugin for Kubernetes für sind in vpc_ip_resource_limit.go auf GitHub aufgeführt. Möglicherweise müssen Sie Ihre CNI-Version aktualisieren, um die neuesten unterstützten Instance-Typen zu nutzen. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

- NodeImageidssmParam: Vorab mit dem Amazon EC2 Systems Manager Manager-Parameter der aktuell empfohlenen Amazon EKS-optimierten Core-AMI-ID gefüllt. Windows Um die Vollversion von Windows zu verwenden, ersetzen Sie `Core` durch `Full`.
- NodeImageID: (Optional) Wenn Sie Ihr eigenes benutzerdefiniertes AMI (anstelle des für Amazon EKS optimierten AMI) verwenden, geben Sie eine Knoten-AMI-ID für Ihr AWS-Region. Wenn Sie einen Wert für dieses Feld angeben, überschreibt dieser alle Werte im Feld NodeImageidssmParam.
- NodeVolumeGröße: Geben Sie eine Root-Volume-Größe für Ihre Knoten in GiB an.
- KeyName: Geben Sie den Namen eines Amazon EC2 SSH-Schlüsselpaars ein, mit dem Sie sich nach dem Start über SSH mit Ihren Knoten verbinden können. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Schlüsselpaare im Amazon-EC2-Benutzerhandbuch.

> **Note**
>
> Wenn Sie hier kein key pair angeben, kann der AWS CloudFormation Stack nicht erstellt werden.

- BootstrapArguments: Geben Sie alle optionalen Argumente an, die an das Node-Bootstrap-Skript übergeben werden sollen, z. B. zusätzliche `kubelet` Argumente mit-`KubeletExtraArgs`.
- DisableIMDSv1: Standardmäßig unterstützt jeder Knoten die Instance-Metadaten-Service-Version 1 (IMDSv1) und IMDSv2. Sie können IMDSv1 deaktivieren. Um zu verhindern, dass künftige Knoten und Pods in der Knotengruppe MDSv1 verwenden, legen Sie DisableIMDSv1 auf true fest. Weitere Informationen finden Sie unter Konfiguration des Instance-Metadatenservice.

- VpcId: Wählen Sie die ID für die VPC aus, die Sie erstellt haben.
- NodeSecurityGruppen: Wählen Sie die Sicherheitsgruppe aus, die für Ihre Linux Knotengruppe erstellt wurde, als Sie Ihre VPC erstellt haben. Wenn Ihren Linux-Knoten mehr als eine Sicherheitsgruppe angehängt ist, geben Sie alle an. Dies z. B., wenn die Linux-Knotengruppe mit `eksctl` erstellt wurde.
- Subnets (Subnetze): Wählen Sie die Subnetze aus, die Sie erstellt haben. Wenn Sie Ihre VPC anhand der unter Erstellen einer VPC für Ihren Amazon-EKS-Cluster beschriebenen Schritte erstellt haben, geben Sie nur die privaten Subnetze innerhalb der VPC an, in denen Ihre Knoten gestartet werden sollen.

> **Important**
>
> - Wenn es sich bei einem oder einigen der Subnetze um öffentliche Subnetze handelt, muss die Einstellung für die automatische Zuweisung öffentlicher IP-Adressen aktiviert sein. Wenn die Einstellung für das öffentliche Subnetz nicht aktiviert ist, wird allen Knoten, die Sie in diesem öffentlichen Subnetz bereitstellen, keine öffentliche IP-Adresse zugewiesen und sie können nicht mit dem Cluster oder anderen Diensten kommunizieren. AWS Wenn das Subnetz vor dem 26. März 2020 mithilfe einer der Amazon AWS CloudFormation EKS-VPC-Vorlagen oder mithilfe `eksctl` von bereitgestellt wurde, ist die automatische Zuweisung öffentlicher IP-Adressen für öffentliche Subnetze deaktiviert. Informationen zum Aktivieren der öffentlichen IP-Adresszuweisung für ein Subnetz finden Sie unter Ändern des öffentlichen `IPv4`-Adressattributs für Ihr Subnetz. Wenn der Knoten in einem privaten Subnetz bereitgestellt wird, kann er über ein NAT-Gateway mit dem Cluster und anderen AWS Diensten kommunizieren.
> - Wenn die Subnetze keinen Internetzugang haben, stellen Sie sicher, dass Sie die Überlegungen und zusätzlichen Schritte in Anforderungen an private Cluster kennen.
> - Wenn Sie Wellenlängen-Subnetze oder Local Zone-Subnetze auswählen AWS Outposts, dürfen die Subnetze bei der Erstellung des Clusters nicht übergeben worden sein.

8. Bestätigen Sie, dass der Stack IAM-Ressourcen erstellen kann, und wählen Sie dann Create stack (Stack erstellen) aus.

9. Wenn Ihr Stack fertig erstellt wurde, wählen Sie ihn in der Konsole aus und klicken Sie auf Outputs (Ausgaben).
10. Notieren Sie sich die NodeInstanceRolle für die Knotengruppe, die erstellt wurde. Sie benötigen diese, wenn Sie Ihre Amazon-EKS-Windows-Arbeitsknoten konfigurieren.

Schritt 2: So aktivieren Sie die Knoten, die Ihrem Cluster beitreten sollen

1. Überprüfen Sie, ob Sie bereits über eine `aws-auth` `ConfigMap` verfügen.

```
kubectl describe configmap -n kube-system aws-auth
```

2. Wenn eine `aws-auth` `ConfigMap` angezeigt wird, aktualisieren Sie sie nach Bedarf.

   a. Öffnen Sie `ConfigMap` zum Bearbeiten.

```
kubectl edit -n kube-system configmap/aws-auth
```

   b. Fügen Sie nach Bedarf neue `mapRoles`-Einträge hinzu. Stellen Sie die `rolearn` Werte auf die NodeInstanceRollenwerte ein, die Sie in den vorherigen Verfahren aufgezeichnet haben.

```
[...]
data:
  mapRoles: |
- rolearn: <ARN of linux instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
    - rolearn: <ARN of windows instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
        - eks:kube-proxy-windows
[...]
```

   c. Speichern Sie die Datei und beenden Sie den Text-Editor.

3. Wenn die Fehlermeldung `Error from server (NotFound): configmaps "aws-auth" not found` angezeigt wird, wenden Sie die standardmäßige `ConfigMap` an.

a. Laden Sie die Konfigurationszuordnung herunter.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-
eks/cloudformation/2020-10-29/aws-auth-cm-windows.yaml
```

b. Stellen Sie in der `aws-auth-cm-windows.yaml` Datei die `rolearn` Werte auf die entsprechenden NodeInstanceRollenwerte ein, die Sie in den vorherigen Verfahren aufgezeichnet haben. Hierzu können Sie einen Texteditor verwenden oder die *`example values`* ersetzen und den folgenden Befehl ausführen:

```
sed -i.bak -e 's|<ARN of linux instance role (not instance profile)>|my-
node-linux-instance-role|' \
    -e 's|<ARN of windows instance role (not instance profile)>|my-node-
windows-instance-role|' aws-auth-cm-windows.yaml
```

> **Important**
>
> - Ändern Sie keine weiteren Zeilen in dieser Datei.
> - Verwenden Sie nicht dieselbe IAM-Rolle sowohl für Windows- als auch für Linux-Knoten.

c. Wenden Sie die Konfiguration an. Die Ausführung dieses Befehls kann einige Minuten dauern.

```
kubectl apply -f aws-auth-cm-windows.yaml
```

4. Sehen Sie sich den Status Ihrer Knoten an und warten Sie, bis diese in den `Ready`-Status eintreten.

```
kubectl get nodes --watch
```

Geben Sie `Ctrl+C` ein, um zu einer Shell-Eingabeaufforderung zurückzukehren.

> **Note**
>
> Wenn Sie Autorisierungs- oder Ressourcenfehler erhalten, finden Sie weitere Informationen unter Nicht autorisiert oder Zugriff verweigert (`kubectl`) im Thema zur Fehlerbehebung.

Wenn Knoten dem Cluster nicht beitreten können, finden Sie weitere Informationen unter Knoten können nicht mit dem Cluster verknüpft werden im Handbuch zur Fehlerbehebung.

Schritt 3: Zusätzliche Aktionen

1. (Optional) Stellen Sie eine Beispielanwendung bereit, um Ihren Cluster und Ihre Windows-Worker-Knoten zu testen.
2. (Optional) Wenn die von AmazonEKS_CNI_Policy verwaltete IAM-Richtlinie (bei einem `IPv4`-Cluster) oder die *`AmazonEKS_CNI_IPv6_Policy`* (die Sie bei einem `IPv6`-Cluster selbst erstellt haben) an Ihre the section called “Knoten-IAM-Rolle” angefügt ist, sollten Sie sie stattdessen einer IAM-Rolle zuweisen, die Sie dem Kubernetes-Servicekonto für `aws-node` zuordnen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).
3. Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

   - Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
   - Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

   Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Starten selbstverwalteter Ubuntu-Knoten

> **Note**
>
> Verwaltete Knotengruppen bieten möglicherweise einige Vorteile für Ihren Anwendungsfall. Weitere Informationen finden Sie unter Verwaltete Knotengruppen.

In diesem Thema wird beschrieben, wie Sie Auto Scaling Scaling-Gruppen von Knoten Ubuntuauf Amazon Elastic Kubernetes Service (EKS) oder Ubuntu ProAmazon Elastic Kubernetes Service (EKS) starten, die sich bei Ihrem Amazon EKS-Cluster registrieren. Ubuntuund Ubuntu Pro für EKS basieren auf dem offiziellen Ubuntu Minimal LTS, enthalten den benutzerdefinierten AWS Kernel, der gemeinsam mit EKS entwickelt wurde AWS, und wurden speziell für EKS entwickelt. Ubuntu Probietet zusätzlichen Sicherheitsschutz durch die Unterstützung erweiterter EKS-Unterstützungszeiträumelivepatch, Kernel- und FIPS-Konformität und die Möglichkeit, unbegrenzt viele Pro Container auszuführen.

Nachdem die Knoten dem Cluster beigetreten sind, können Sie containerisierte Anwendungen für sie bereitstellen. Weitere Informationen finden Sie in der Dokumentation für UbuntuOn AWS - und Custom AMI-Support in der `eksctl` Dokumentation.

> **Important**
>
> - Amazon-EKS-Worker-Knoten sind Standard-Amazon-EC2-Instances und werden Ihnen basierend auf normalen Amazon-EC2-Instance-Preisen berechnet. Weitere Informationen dazu finden Sie unter Amazon EC2 – Preise.
> - Sie können Ubuntu Knoten in erweiterten Amazon EKS-Clustern auf AWS Outposts starten, aber Sie können sie nicht in lokalen Clustern auf AWS Outposts starten. Weitere Informationen finden Sie unter auf Amazon EKSAWS Outposts.
> - Sie können auf Amazon-EC2-Instances mit x86- oder Arm-Prozessoren bereitstellen. Instances mit Inferentia Chips müssen jedoch möglicherweise zuerst das NeuronSDK installieren.

Zum Starten Ubuntu für EKS oder Ubuntu Pro für EKS-Knoten verwenden **`eksctl`**

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre -Version mit dem folgenden Befehl überprüfen:

```
eksctl version
```

Eine Installations- und Upgrade-Anleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

> **Note**
>
> Dieses Verfahren funktioniert nur für Cluster, die mit `eksctl` erstellt wurden.

1. Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie `my-cluster` mit dem Namen Ihres Clusters. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Er muss mit einem alphabetischen Zeichen beginnen und darf nicht mehr als 100 Zeichen umfassen. Ersetzen Sie `ng-ubuntu` durch einen Namen für Ihre Knotengruppe. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten. Um die Bereitstellung auf Arm Instances durchzuführen, `m5.large` ersetzen Sie es durch einen Arm Instance-Typ. Ersetzen Sie `my-ec2-keypair-name` mit dem Namen eines Amazon-EC2-SSH-Schlüsselpaars ein, das Sie für die Verbindung über SSH in Ihre Arbeitsknoten verwenden können, nachdem sie gestartet wurden. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon EC2 EC2-Schlüsselpaare im Amazon EC2 EC2-Benutzerhandbuch. Ersetzen Sie *`example values`* durch Ihre eigenen Werte. Nachdem Sie das Ersetzen vorgenommen haben, führen Sie den geänderten Befehl aus, um die `ubuntu.yaml`-Datei zu erstellen.

   > **Important**
   >
   > Um eine Knotengruppe in Subnetzen oder AWS lokalen Zonensubnetzen bereitzustellen AWS Outposts AWS Wavelength, übergeben Sie bei der Erstellung des Clusters keine Subnetze der AWS lokalen Zone. AWS Outposts AWS Wavelength Sie müssen die Subnetze im folgenden Beispiel angeben. Weitere Informationen finden Sie unter Verwenden von Config-Dateien und im Config-Datei-Schema in der `eksctl`-

Dokumentation. Ersetzen Sie es *region-code* durch AWS-Region das, in dem sich Ihr Cluster befindet.

```
cat >ubuntu.yaml <<EOF
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code
  version: '1.30'

iam:
  withOIDC: true

nodeGroups:
  - name: ng-ubuntu
    instanceType: m5.large
    desiredCapacity: 3
    amiFamily: Ubuntu22.04
    ami: auto-ssm
    iam:
       attachPolicyARNs:
          - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
          - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
          - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
          - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
    ssh:
        allow: true
        publicKeyName: my-ec2-keypair-name
EOF
```

Um eine Ubuntu Pro Knotengruppe zu erstellen, ändern Sie einfach den `amiFamily` Wert in`UbuntuPro2204`.

2. Stellen Sie den Treiber mit dem folgenden Befehl bereit.

```
eksctl create nodegroup --config-file=ubuntu.yaml
```

Eine Beispielausgabe sieht wie folgt aus.

Mehrere Zeilen werden ausgegeben, während die Knoten erstellt werden. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Optional) Stellen Sie eine Beispielanwendung bereit, um Ihre Ubuntu-Knoten zu testen.
4. Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

   - Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
   - Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

   Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Aktualisierungen des selbstverwalteten Worker-Knotens

Wenn eine neue Amazon-EKS-optimierte AMI veröffentlicht wird, ziehen Sie in Betracht, die Knoten in Ihrer selbstverwalteten Knoten-Gruppe durch die neue AMI zu ersetzen. Ebenso gilt, dass Sie bei einer Aktualisierung der Kubernetes-Version für Ihren Amazon-EKS-Cluster auch die Worker-Knoten aktualisieren, um Knoten mit derselben Kubernetes-Version zu verwenden.

> **Important**
>
> In diesem Thema werden Worker-Knotenaktualisierungen für selbstverwaltete Knotengruppen behandelt. Wenn Sie Verwaltete Knotengruppen verwenden, vgl. Aktualisieren einer verwalteten Knotengruppe.

Es gibt zwei grundlegende Möglichkeiten, selbstverwaltete Knotengruppen in Ihren Clustern zu aktualisieren, um ein neues AMI zu verwenden:

Migrieren zu einer neuen Worker-Knoten-Gruppe

Erstellen Sie eine neue Worker-Knotengruppe und migrieren Sie Ihre Pods zu dieser Gruppe. Die Migration zu einer neuen Knotengruppe ist sinnvoller als die Aktualisierung der AMI-ID in einem bestehenden AWS CloudFormation-Stack. Dies liegt daran, dass der Migrationsprozess die alte Knotengruppe als `NoSchedule` verunreinigt und die Knoten leert, nachdem ein neuer Stack bereit ist, die vorhandene Pod-Workload zu akzeptieren.

Aktualisieren einer vorhandenen selbstverwalteten Knotengruppe

Aktualisieren Sie den AWS CloudFormation-Stapel für eine vorhandene Worker-Knotengruppe, um das neue AMI zu verwenden. Diese Methode wird nicht bei Knoten-Gruppen unterstützt, die mit `eksctl` erstellt wurden.

Migrieren zu einer neuen Worker-Knoten-Gruppe

Dieses Thema beschreibt Hinweise zum Erstellen einer neuen Knoten-Gruppe, zum ordnungsgemäßen Migrieren Ihrer vorhandenen Anwendungen zur neuen Gruppe und zum Entfernen der alten Knoten-Gruppe aus Ihrem Cluster. Sie können zu einer neuen Knotengruppe migrieren, indem Sie `eksctl` oder AWS Management Console benutzen.

eksctl

So migrieren Sie Ihre Anwendungen mit **`eksctl`** zu einer neuen Worker-Knoten-Gruppe

Weitere Informationen zur Verwendung von eksctl für die Migration finden Sie in der Dokumentation unter Unmanaged Nodegroups. `eksctl`

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre -Version mit dem folgenden Befehl überprüfen:

```
eksctl version
```

Eine Installations- und Upgrade-Anleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

> **Note**
>
> Dieses Verfahren funktioniert nur für Cluster, die mit `eksctl` erstellt wurden.

1. Rufen Sie den Namen Ihrer vorhandenen Knotengruppen ab und ersetzen Sie *my-cluster* durch Ihren Clusternamen.

```
eksctl get nodegroups --cluster=my-cluster
```

Eine Beispielausgabe sieht wie folgt aus.

```
CLUSTER      NODEGROUP          CREATED               MIN SIZE      MAX SIZE
 DESIRED CAPACITY     INSTANCE TYPE     IMAGE ID
default      standard-nodes   2019-05-01T22:26:58Z  1             4             3
                    t3.medium         ami-05a71d034119ffc12
```

2. Starten Sie eine neue Knotengruppe mit `eksctl` mit dem folgenden Befehl. Ersetzen Sie im Befehl jedes *example value* durch Ihre eigenen Werte. Die Versionsnummer darf nicht höher als die Kubernetes-Version für Ihre Steuerebene sein. Außerdem darf sie nicht mehr als zwei Nebenversionen älter sein als die Kubernetes-Version für Ihre Steuerebene. Es wird empfohlen, dieselbe Version wie die Steuerebene zu verwenden.

Wir empfehlen, den Pod-Zugriff auf das IMDS zu blockieren, wenn die folgenden Bedingungen erfüllt sind:

- Sie planen, allen Ihren Kubernetes-Servicekonten IAM-Rollen zuzuweisen, damit Pods nur die Mindestberechtigungen haben, die sie benötigen.
- Nein Pods im Cluster benötigen aus anderen Gründen Zugriff auf den Amazon EC2 EC2-Instance-Metadaten-Service (IMDS), z. B. zum Abrufen der aktuellen Version. AWS-Region

Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Um den Pod-Zugriff auf im IMDS zu blockieren, fügen Sie dem folgenden Befehl die Option `--disable-pod-imds` hinzu.

> **Note**
>
> Weitere verfügbare Flags und deren Beschreibungen finden Sie unterhttps://eksctl.io/.

```
eksctl create nodegroup \
```

```
--cluster my-cluster \
--version 1.30 \
--name standard-nodes-new \
--node-type t3.medium \
--nodes 3 \
--nodes-min 1 \
--nodes-max 4 \
--managed=false
```

3. Wenn der vorherige Befehl abgeschlossen ist, bestätigen Sie mit folgendem Befehl, dass alle Ihre Worker-Knoten den `Ready`-Status erreicht haben:

```
kubectl get nodes
```

4. Löschen Sie die ursprüngliche Knotengruppe mit dem folgenden Befehl. Ersetzen Sie im Befehl alle *example value* mit Ihren Cluster- und Knotengruppennamen:

```
eksctl delete nodegroup --cluster my-cluster --name standard-nodes-old
```

AWS Management Console and AWS CLI

Um Ihre Anwendungen mit dem und auf eine neue Knotengruppe zu migrieren AWS Management ConsoleAWS CLI

1. Starten Sie eine neue Knoten-Gruppe, indem Sie die unter Starten selbstverwalteter Amazon Linux-Knoten beschriebenen Schritte ausführen.
2. Wenn Ihr Stack fertig erstellt wurde, wählen Sie ihn in der Konsole aus und klicken Sie auf Outputs (Ausgänge).
3. Notieren Sie sich die NodeInstanceRolle für die Knotengruppe, die erstellt wurde. Sie benötigen diese Informationen zum Hinzufügen der neuen Amazon EKS-Knoten zu Ihrem Cluster.

> **Note**
>
> Wenn Sie der IAM-Rolle Ihrer alten Knotengruppen zusätzliche IAM-Richtlinien angefügt haben, dann sollten Sie die gleichen Richtlinien auch der IAM-Rolle Ihrer neuen Knotengruppe zuweisen, um diese Funktionalität für die neue Gruppe

zu erhalten. Dies gilt für Sie, wenn Sie beispielsweise Berechtigungen für den Kubernetes Cluster-Autoscaler hinzugefügt haben.

4. Aktualisieren Sie die Sicherheitsgruppen für beide Worker-Knoten-Gruppen, sodass sie miteinander kommunizieren können. Weitere Informationen finden Sie unter Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe.

   a. Notieren Sie die Sicherheitsgruppen-IDs beider Worker-Knoten-Gruppen. Dies wird in den AWS CloudFormation Stack-Ausgaben als NodeSecurityGruppenwert angezeigt.

      Sie können die folgenden AWS CLI Befehle verwenden, um die Sicherheitsgruppen-IDs aus den Stack-Namen abzurufen. In diesen Befehlen `oldNodes` steht der AWS CloudFormation Stack-Name für Ihren älteren Knotenstapel und `newNodes` der Name des Stacks, zu dem Sie migrieren. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte.

      ```
      oldNodes="old_node_CFN_stack_name"
      newNodes="new_node_CFN_stack_name"

      oldSecGroup=$(aws cloudformation describe-stack-resources --stack-name
       $oldNodes \
      --query 'StackResources[?
      ResourceType==`AWS::EC2::SecurityGroup`].PhysicalResourceId' \
      --output text)
      newSecGroup=$(aws cloudformation describe-stack-resources --stack-name
       $newNodes \
      --query 'StackResources[?
      ResourceType==`AWS::EC2::SecurityGroup`].PhysicalResourceId' \
      --output text)
      ```

   b. Fügen Sie Regeln für eingehenden Datenverkehr für jede Worker-Knoten-Sicherheitsgruppe hinzu, sodass sie voneinander Datenverkehr annehmen können.

      Mit den folgenden AWS CLI Befehlen werden jeder Sicherheitsgruppe Regeln für eingehenden Datenverkehr hinzugefügt, die den gesamten Datenverkehr über alle Protokolle der anderen Sicherheitsgruppe zulassen. Auf diese Weise können Pods in jeder Knoten-Gruppe miteinander kommunizieren, während Sie Ihr Workload zur neuen Gruppe migrieren.

      ```
      aws ec2 authorize-security-group-ingress --group-id $oldSecGroup \
      ```

```
--source-group $newSecGroup --protocol -1
aws ec2 authorize-security-group-ingress --group-id $newSecGroup \
--source-group $oldSecGroup --protocol -1
```

5. Bearbeiten Sie die `aws-auth`-configmap, um die neue Worker-Knoten-Instance-Rolle in RBAC zuzuordnen.

```
kubectl edit configmap -n kube-system aws-auth
```

Fügen Sie einen neuen `mapRoles`-Eintrag für die neue Worker-Knoten-Gruppe hinzu. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (US-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch. `arn:aws:` `arn:aws-us-gov:`

```
apiVersion: v1
data:
  mapRoles: |
    - rolearn: ARN of instance role (not instance profile)
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes>
    - rolearn: arn:aws:iam::111122223333:role/nodes-1-16-NodeInstanceRole-
U11V27W93CX5
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
```

Ersetzen Sie das *ARN of instance role (not instance profile)* Snippet durch den NodeInstanceRollenwert, den Sie in einem vorherigen Schritt aufgezeichnet haben. Speichern und schließen Sie dann die Datei, um die aktualisierte configmap anzuwenden.

6. Achten Sie auf den Status Ihrer Knoten und warten Sie, bis Ihre neuen Worker-Knoten Ihrem Cluster beigetreten sind und den Status `Ready` angenommen haben.

```
kubectl get nodes --watch
```

7. (Optional) Wenn Sie den Kubernetes Cluster Autoscaler verwenden, skalieren Sie die Bereitstellung nach unten auf null (0) Replikate, um Konflikte zwischen Skalierungsaktionen zu vermeiden.

```
kubectl scale deployments/cluster-autoscaler --replicas=0 -n kube-system
```

8. Verwenden Sie den folgenden Befehl, um jeden der Knoten, die Sie mit `NoSchedule` entfernen möchten, mit einem Taint zu versehen. Auf diese Weise werden neue Pods auf den Knoten, die Sie ersetzen, nicht geplant oder neu geplant. Weitere Informationen zu finden Sie unter Taints and Tolerations (Taints und Toleranzen) in der Kubernetes-Dokumentation.

   ```
   kubectl taint nodes node_name key=value:NoSchedule
   ```

   Wenn Sie Ihre Knoten auf eine neue Kubernetes-Version aktualisieren, können Sie alle Knoten einer bestimmten Kubernetes-Version (in diesem Fall `1.28`) mit dem folgenden Codeausschnitt identifizieren und mit einem Taint versehen. Die Versionsnummer darf nicht höher als die Kubernetes-Version Ihrer Steuerebene sein. Sie darf auch nicht mehr als zwei Nebenversionen älter sein als die Kubernetes-Version Ihrer Steuerebene. Es wird empfohlen, dieselbe Version wie die Steuerebene zu verwenden.

   ```
   K8S_VERSION=1.28
   nodes=$(kubectl get nodes -o jsonpath="{.items[?
   (@.status.nodeInfo.kubeletVersion==\"v$K8S_VERSION\")].metadata.name}")
   for node in ${nodes[@]}
   do
       echo "Tainting $node"
       kubectl taint nodes $node key=value:NoSchedule
   done
   ```

9. Bestimmen Sie den DNS-Anbieter Ihres Clusters.

   ```
   kubectl get deployments -l k8s-app=kube-dns -n kube-system
   ```

   Eine Beispielausgabe sieht wie folgt aus. Dieser Cluster verwendet CoreDNS für die DNS-Auflösung, aber Ihr Cluster kann stattdessen `kube-dns` zurückgeben):

   ```
   NAME      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
   coredns   1         1         1            1           31m
   ```

10. Wenn Ihre aktuelle Bereitstellung weniger als zwei Replikate ausführt, skalieren die Bereitstellung auf zwei Replikate. Ersetzen Sie **kubedns** durch *coredns*, falls Ihre vorherige Befehlsausgabe dies stattdessen zurückgegeben hat.

```
kubectl scale deployments/coredns --replicas=2 -n kube-system
```

11. Lassen Sie die einzelnen Knoten, die Sie aus Ihrem Cluster entfernen möchten, mit dem folgenden Befehl sperren:

    ```
    kubectl drain node_name --ignore-daemonsets --delete-local-data
    ```

    Wenn Sie Ihre Knoten auf eine neue Kubernetes-Version aktualisieren, können Sie alle Knoten einer bestimmten Kubernetes-Version (in diesem Fall *1.28*) mit dem folgenden Codeausschnitt identifizieren und leeren.

    ```
    K8S_VERSION=1.28
    nodes=$(kubectl get nodes -o jsonpath="{.items[?
    (@.status.nodeInfo.kubeletVersion==\"v$K8S_VERSION\")].metadata.name}")
    for node in ${nodes[@]}
    do
        echo "Draining $node"
        kubectl drain $node --ignore-daemonsets --delete-local-data
    done
    ```

12. Nachdem die alten Knoten entladen wurden, widerrufen Sie die Regeln für eingehenden Datenverkehr für Sicherheitsgruppen, die Sie zuvor autorisiert haben. Löschen Sie anschließend den AWS CloudFormation Stack, um die Instances zu beenden.

    > **Note**
    >
    > Wenn Sie Ihrer alten Knotengruppe (IAM-Rolle) zusätzliche IAM-Richtlinien hinzugefügt haben (z. B. das Hinzufügen von Berechtigungen für den Kubernetes Cluster Autoscaler), trennen Sie diese zusätzlichen Richtlinien von der Rolle, bevor Sie Ihren Stack löschen können. AWS CloudFormation

    a. Heben Sie die Regeln für eingehenden Datenverkehr auf, die Sie zuvor für die Knoten-Sicherheitsgruppen erstellt haben. In diesen Befehlen `oldNodes` steht der AWS CloudFormation Stack-Name für Ihren älteren Knoten-Stack und `newNodes` der Name des Stacks, zu dem Sie migrieren.

    ```
    oldNodes="old_node_CFN_stack_name"
    ```

```
newNodes="new_node_CFN_stack_name"

oldSecGroup=$(aws cloudformation describe-stack-resources --stack-name
 $oldNodes \
--query 'StackResources[?
ResourceType==`AWS::EC2::SecurityGroup`].PhysicalResourceId' \
--output text)
newSecGroup=$(aws cloudformation describe-stack-resources --stack-name
 $newNodes \
--query 'StackResources[?
ResourceType==`AWS::EC2::SecurityGroup`].PhysicalResourceId' \
--output text)
aws ec2 revoke-security-group-ingress --group-id $oldSecGroup \
--source-group $newSecGroup --protocol -1
aws ec2 revoke-security-group-ingress --group-id $newSecGroup \
--source-group $oldSecGroup --protocol -1
```

b. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

c. Wählen Sie Ihren alten Worker-Knoten-Stack aus.

d. Wählen Sie Delete (Löschen).

e. Wählen Sie im Bestätigungsdialogfeld Stack löschen Stack löschen aus.

13. Bearbeiten Sie die `aws-auth`-configmap, um die alten Worker-Knoten-Instance-Rolle aus RBAC zu entfernen.

```
kubectl edit configmap -n kube-system aws-auth
```

Löschen Sie den `mapRoles`-Eintrag für die alte Worker-Knoten-Gruppe. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (USA-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch`arn:aws:`. `arn:aws-us-gov:`

```
apiVersion: v1
data:
  mapRoles: |
    - rolearn: arn:aws:iam::111122223333:role/nodes-1-16-NodeInstanceRole-
W70725MZQFF8
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
```

```
    - rolearn: arn:aws:iam::111122223333:role/nodes-1-15-NodeInstanceRole-
U11V27W93CX5
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes>
```

Speichern und schließen Sie die Datei, um die aktualisierte configmap anzuwenden.

14. (Optional) Wenn Sie den Kubernetes Cluster Autoscaler verwenden, skalieren Sie die Bereitstellung wieder auf ein Replikat.

> **Note**
>
> Außerdem müssen Sie Ihre neue Auto-Scaling-Gruppe (z. B. `k8s.io/cluster-autoscaler/enabled,k8s.io/cluster-autoscaler/my-cluster`) mit einem Tag versehen und den Befehl für die Cluster-Autoscaler-Bereitstellung so aktualisieren, dass er auf die neu getaggte Auto-Scaling-Gruppe verweist. Weitere Informationen finden Sie unter Cluster Autoscaler on. AWS

```
kubectl scale deployments/cluster-autoscaler --replicas=1 -n kube-system
```

15. (Optional) Stellen Sie sicher, dass Sie die neueste Version des Amazon-VPC-CNI-Plugins für Kubernetes verwenden. Möglicherweise müssen Sie Ihre CNI-Version aktualisieren, um die neuesten unterstützten Instance-Typen zu nutzen. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.
16. Wenn Ihr Cluster für die DNS-Auflösung `kube-dns` verwendet (siehe voriger Schritt), skalieren Sie in der `kube-dns`-Bereitstellung auf ein Replikat.

```
kubectl scale deployments/kube-dns --replicas=1 -n kube-system
```

Aktualisieren einer vorhandenen selbstverwalteten Knotengruppe

In diesem Thema wird beschrieben, wie Sie einen vorhandenen AWS CloudFormation selbstverwalteten Knotenstapel mit einem neuen AMI aktualisieren können. Sie können diese Anleitung verwenden, um Ihre Knoten nach einer Cluster-Aktualisierung auf eine neue Version von

Kubernetes zu aktualisieren. Andernfalls können Sie auf das neueste von Amazon EKS optimierte AMI für eine vorhandene Kubernetes-Version aktualisieren.

> **Important**
>
> In diesem Thema werden Worker-Knotenaktualisierungen für selbstverwaltete Knotengruppen behandelt. Für weitere Informationen zur Nutzung von Verwaltete Knotengruppen siehe Aktualisieren einer verwalteten Knotengruppe.

Die neueste Amazon AWS CloudFormation EKS-Standardknotenvorlage ist so konfiguriert, dass eine Instance mit dem neuen AMI in Ihrem Cluster gestartet wird, bevor nacheinander eine alte entfernt wird. Diese Konfiguration stellt sicher, dass Sie immer über die gewünschte Anzahl der aktiven Instances Ihre Auto-Scaling-Gruppe in Ihrem Cluster verfügen, während das Update durchgeführt wird.

> **Note**
>
> Diese Methode wird nicht bei Knoten-Gruppen unterstützt, die mit `eksctl` erstellt wurden. Wenn Sie Ihr Cluster oder Ihre Worker-Knoten-Gruppe Sie `eksctl` erstellt haben, finden Sie Informationen unter Migrieren zu einer neuen Worker-Knoten-Gruppe.

So aktualisieren Sie eine vorhandene Worker-Knoten-Gruppe

1. Bestimmen Sie den DNS-Anbieter Ihres Clusters.

   ```
   kubectl get deployments -l k8s-app=kube-dns -n kube-system
   ```

   Eine Beispielausgabe sieht wie folgt aus. Dieser Cluster verwendet CoreDNS für die DNS-Auflösung, aber Ihr Cluster kann stattdessen `kube-dns` zurückgeben. Ihre Ausgabe kann je nach verwendeter `kubectl`-Version anders aussehen.

   ```
   NAME      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
   coredns   1         1         1            1           31m
   ```

2. Wenn Ihre aktuelle Bereitstellung weniger als zwei Replikate ausführt, skalieren die Bereitstellung auf zwei Replikate. Ersetzen Sie **`kube-dns`** durch *`coredns`*, falls Ihre vorherige Befehlsausgabe dies stattdessen zurückgegeben hat.

```
kubectl scale deployments/coredns --replicas=2 -n kube-system
```

3. (Optional) Wenn Sie den Kubernetes Cluster Autoscaler verwenden, skalieren Sie die Bereitstellung nach unten auf null (0) Replikate, um Konflikte zwischen Skalierungsaktionen zu vermeiden.

```
kubectl scale deployments/cluster-autoscaler --replicas=0 -n kube-system
```

4. Bestimmen Sie den gewünschten Instance-Typ und die gewünschte Anzahl von Instances Ihrer aktuellen Worker-Knoten-Gruppe. Sie geben diese Werte später ein, wenn Sie die AWS CloudFormation Vorlage für die Gruppe aktualisieren.

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

   b. Wählen Sie im linken Navigationsbereich Launch Configurations (Startkonfigurationen) aus und beachten Sie den Instance-Typ für die Startkonfiguration der vorhandenen Knoten.

   c. Wählen Sie im linken Navigationsbereich Auto Scaling Groups (Auto-Scaling-Gruppen) aus und beachten Sie die Desired (gewünschte) Instance-Anzahl für die Auto-Scaling-Gruppe der vorhandenen Knoten.

5. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

6. Wählen Sie Ihren Workerknoten-Gruppen-Stack aus und klicken Sie dann auf Update (Aktualisieren).

7. Wählen Sie Replace current template (Aktuelle Vorlage ersetzen) und dann Amazon S3 URL (Amazon S3-URL) aus.

8. Fügen Sie für Amazon S3 S3-URL die folgende URL in den Textbereich ein, um sicherzustellen, dass Sie die neueste Version der AWS CloudFormation Node-Vorlage verwenden. Klicken Sie dann auf Next (Weiter):

```
https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2022-12-23/amazon-eks-nodegroup.yaml
```

9. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) die folgenden Parameter ein und wählen Sie Next (Weiter) aus:

   - NodeAutoScalingGroupDesiredCapacity— Geben Sie die gewünschte Anzahl von Instanzen ein, die Sie in einem vorherigen Schritt aufgezeichnet haben. Oder geben Sie die neue

gewünschte Anzahl von Knoten ein, auf die bei der Aktualisierung Ihres Stacks skaliert werden soll.

- NodeAutoScalingGroupMaxSize— Geben Sie die maximale Anzahl von Knoten ein, auf die Ihre Node-Auto-Scaling-Gruppe skalieren kann. Dieser Wert muss mindestens einen Knoten größer sein als Ihre gewünschte Kapazität. Auf diese Weise können Sie eine fortlaufende Aktualisierung Ihrer Knoten durchführen, ohne die Knotenanzahl während der Aktualisierung zu reduzieren.
- NodeInstanceTyp — Wählen Sie den Instance-Typ, den Sie in einem vorherigen Schritt aufgezeichnet haben. Wählen Sie alternativ einen anderen Instance-Typ für Ihre Knoten aus. Bevor Sie einen Instance-Typ auswählen, lesen Sie Auswählen eines Amazon-EC2-Instance-Typs. Jeder Amazon-EC2-Instance-Typ unterstützt eine maximale Anzahl von Elastic-Network-Interfaces (ENIs) und jedes ENI unterstützt eine maximale Anzahl von IP-Adressen. Da jedem Worker-Knoten und Pod eine eigene IP-Adresse zugewiesen wird, ist es wichtig, einen Instance-Typ auszuwählen, der die maximale Anzahl von Pods unterstützt, die auf jedem Amazon-EC2-Knoten ausgeführt werden sollen. Eine Liste der Netzwerkschnittstellen und IP-Adressen, die von Instance-Typen unterstützt werden, finden Sie unter IP-Adressen pro Netzwerkschnittstelle pro Instance-Typ. Der Instance–Typ `m5.large` unterstützt zum Beispiel maximal 30 IP-Adressen für den Worker-Knoten und die Pods.

  > **Note**
  >
  > Die unterstützten Instance-Typen für die neueste Version des Amazon VPC CNI plugin for Kubernetes werden in vpc_ip_resource_limit.go auf GitHub angezeigt. Möglicherweise müssen Sie Ihre Amazon VPC CNI plugin for Kubernetes-Version aktualisieren, um die neuesten unterstützten Instance-Typen zu verwenden. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

  > **Important**
  >
  > Einige Instance-Typen sind möglicherweise nicht in allen verfügbar AWS-Regionen.

- NodeImageidssmParam — Der Amazon EC2 Systems Manager Manager-Parameter der AMI-ID, auf die Sie aktualisieren möchten. Der folgende Wert verwendet das neueste Amazon-EKS-optimierte AMI für Kubernetes-Version `1.30`.

```
/aws/service/eks/optimized-ami/1.30/amazon-linux-2/recommended/image_id
```

Sie können *1.30* mit einer unterstützten Kubernetes-Version ersetzen, die identisch ist. Oder er sollte bis zu einer Version älter sein als die Kubernetes-Version, die auf Ihrer Steuerebene läuft. Es wird empfohlen, die Knoten auf der gleichen Version wie die Steuerungsebene zu halten. Sie können es auch *amazon-linux-2* durch einen anderen AMI-Typ ersetzen. Weitere Informationen finden Sie unter Abrufen von Amazon-EKS-optimierten Amazon-Linux-AMI-IDs.

> **Note**
>
> Mit dem Amazon-EC2-Systems-Manager-Parameter können Sie Ihre Worker-Knoten in Zukunft aktualisieren, ohne eine AMI-ID suchen und angeben zu müssen. Wenn Ihr AWS CloudFormation -Stack diesen Wert verwendet, startet jedes Stack-Update immer das neueste empfohlene Amazon-EKS-optimierte AMI für die angegebene Kubernetes-Version. Dies ist auch dann der Fall, wenn Sie keine Werte in der Vorlage ändern.

- NodeImageID — Um Ihr eigenes benutzerdefiniertes AMI zu verwenden, geben Sie die ID ein, die das AMI verwenden soll.

> **Important**
>
> Dieser Wert hat Vorrang vor allen für NodeImageidssmParam angegebenen Wert. Wenn Sie den Wert NodeImageidssmParam verwenden möchten, stellen Sie sicher, dass der Wert für Id leer ist. NodeImage

- DisableIMDSv1 - Standardmäßig unterstützt jeder Knoten die Instance-Metadaten-Service-Version 1 (IMDSv1) und IMDSv2. Sie können IMDSv1 jedoch deaktivieren. Wählen Sie true, wenn keine Knoten oder Pods, die in der Knotengruppe geplant sind, IMDSv1 verwenden sollen. Weitere Informationen finden Sie unter Konfiguration des Instance-Metadatenservice. Wenn Sie IAM-Rollen für Dienstkonten implementiert haben, weisen Sie allen, die Zugriff auf Dienste benötigen, Pods die erforderlichen Berechtigungen direkt zu. AWS Auf diese Weise benötigen Sie Pods in Ihrem Cluster keinen Zugriff auf IMDS aus anderen Gründen, z. B. zum Abrufen der aktuellen Daten. AWS-Region Dann können Sie auch den Zugriff auf IMDSv2 für

Pods deaktivieren, die kein Host-Netzwerk verwenden. Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

10. (Optional) Markieren Sie auf der Seite Options (Optionen) Ihre Stack-Ressourcen. Wählen Sie Next (Weiter).
11. Überprüfen Sie Ihre Angaben auf der Seite Review (Überprüfen), bestätigen Sie, dass der Stack IAM-Ressourcen erstellen kann, und klicken Sie dann auf Update stack (Stack aktualisieren).

> **Note**
>
> Die Aktualisierung jedes Knotens im Cluster dauert mehrere Minuten. Warten Sie, bis die Aktualisierung aller Knoten abgeschlossen ist, bevor Sie die nächsten Schritte durchführen.

12. Wenn der DNS-Anbieter Ihres Clusters `kube-dns` ist, skalieren Sie die `kube-dns`-Bereitstellung auf ein Replikat.

```
kubectl scale deployments/kube-dns --replicas=1 -n kube-system
```

13. (Optional) Wenn Sie den Kubernetes-Cluster Autoscaler verwenden, skalieren Sie die Bereitstellung zurück auf die gewünschte Zahl von Replikaten.

```
kubectl scale deployments/cluster-autoscaler --replicas=1 -n kube-system
```

14. (Optional) Stellen Sie sicher, dass Sie die neueste Version des Amazon VPC CNI plugin for Kubernetes verwenden. Möglicherweise müssen Sie Ihre Amazon VPC CNI plugin for Kubernetes-Version aktualisieren, um die neuesten unterstützten Instance-Typen zu verwenden. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

AWS Fargate

> **Important**
>
> AWS Fargate mit Amazon EKS ist in AWS GovCloud (USA-Ost) und AWS GovCloud (USA-West) nicht verfügbar.

In diesem Thema wird die Verwendung von Amazon EKS zum Ausführen von Kubernetes Pods auf AWS Fargate erläutert. Fargate ist eine Technologie, die bedarfsgerechte On-Demand-Rechenkapazität für Container bereitstellt. Mit -Fargate müssen Sie keine Gruppen virtueller Maschinen selbst bereitstellen, konfigurieren oder skalieren, um Container auszuführen. Sie müssen auch keine Servertypen auswählen, entscheiden, wann Sie Ihre Knotengruppen skalieren oder das Cluster-Packaging optimieren.

Sie können steuern, welche Pods auf Fargate starten und wie sie mit Fargate-Profilen ausgeführt werden. Fargate-Profile werden als Teil Ihres Amazon-EKS-Clusters definiert. Amazon EKS lässt sich Kubernetes mit Fargate integrieren, indem Controller verwendet werden, die von AWS mithilfe des Upstream-Erweiterungsmodells von erstellt wurdenKubernetes. Diese Controller werden als Teil der von Amazon EKS verwalteten Kubernetes-Steuerebene ausgeführt und sind für die Planung nativer Kubernetes-Pods in Fargate verantwortlich. Die Fargate-Controller enthalten einen neuen Scheduler, der neben dem Standard-Kubernetes-Scheduler und zusätzlich zu mehreren mutierenden und validierenden Zugangscontrollern ausgeführt wird. Wenn Sie einen Pod starten, der die Kriterien für die Ausführung in Fargate erfüllt, erkennen die Fargate-Controller, die im Cluster ausgeführt werden den Pod, und aktualisieren und planen ihn in Fargate.

In diesem Thema werden die verschiedenen Komponenten von Pods beschrieben, die auf Fargate ausgeführt werden, und auf besondere Überlegungen für die Verwendung von Fargate mit Amazon EKS hingewiesen.

AWS Fargate Überlegungen

Hier sind einige Dinge, die Sie bei der Verwendung von Fargate auf Amazon EKS beachten sollten.

- Jeder Pod, der auf Fargate ausgeführt wird, hat seine eigene Isolationsgrenze. Sie teilen sich den zugrunde liegenden Kernel, die CPU-Ressourcen, die Arbeitsspeicherressourcen oder die Elastic-Network-Schnittstelle nicht mit einem anderen Pod.
- Network Load Balancer und Application Load Balancer (ALBs) können mit Fargate nur mit IP-Zielen verwendet werden. Weitere Informationen finden Sie unter Erstellen eines Network Load Balancers und Application Load Balancing auf Amazon EKS.
- Fargate-exponierte Services werden nur im IP-Modus des Zieltyps und nicht im Knoten-IP-Modus ausgeführt. Die empfohlene Möglichkeit, die Konnektivität von einem Service zu überprüfen, der auf einem verwalteten Knoten ausgeführt wird, und einem Service, der auf Fargate ausgeführt wird, besteht darin, eine Verbindung über den Servicenamen herzustellen.
- Pods müssen zu dem Zeitpunkt, zu dem sie auf Fargate ausgeführt werden sollen, mit einem Fargate-Profil übereinstimmen. Pods, die nicht mit einem Fargate-Profil übereinstimmen, können

als `Pending` stecken bleiben. Wenn ein übereinstimmendes Fargate-Profil vorhanden ist, können Sie ausstehende Pods, die Sie erstellt haben, löschen, um sie in Fargate neu zu planen.

- Daemonsets werden von Fargate nicht unterstützt. Wenn Ihre Anwendung einen Daemon benötigt, konfigurieren Sie diesen Daemon neu, sodass er als Sidecar-Container in Ihren Pods ausgeführt wird.
- Privilegierte Container werden in Fargate nicht unterstützt.
- Pods, die auf Fargate ausgeführt werden, können im `HostPort`-Manifest kein `HostNetwork` oder Pod angeben.
- Das standardmäßige weiche `nofile`- und `nproc`-Limit beträgt 1024 und das harte Limit 65535 für Fargate-Pods.
- GPUs sind derzeit auf Fargate nicht verfügbar.
- Pods, die auf Fargate ausgeführt werden, werden nur in privaten Subnetzen unterstützt (mit NAT-Gateway-Zugriff auf - AWS Services, aber ohne direkte Route zu einem Internet-Gateway). Daher müssen für die VPC Ihres Clusters private Subnetze verfügbar sein. Weitere Informationen zu Clustern ohne ausgehenden Internetzugang finden Sie unter Anforderungen an private Cluster.
- Sie können Vertical Pod Autoscaler verwenden, um die CPU und den Speicher für Ihre Fargate-Pods anfänglich richtig zu dimensionieren, und dann Horizontal Pod Autoscaler verwenden, um diese Pods zu skalieren. Wenn Sie möchten, dass der Vertical Pod Autoscaler Pods mit größeren CPU- und Speicherkombinationen automatisch erneut in Fargate bereitstellt, stellen Sie den Modus für den Vertical Pod Autoscaler entweder auf `Auto` oder `Recreate` ein, um die korrekte Funktionalität sicherzustellen. Weitere Informationen finden Sie in der Vertical Pod Autoscaler-Dokumentation auf GitHub.
- DNS-Auflösung und DNS-Hostnamen müssen für Ihre VPC aktiviert sein. Weitere Informationen finden Sie unter Anzeigen und Aktualisieren der DNS-Unterstützung für Ihre VPC.
- Amazon-EKS-Fargate fügt defense-in-depth für Kubernetes Anwendungen hinzu, indem jeder Pod innerhalb einer virtuellen Maschine (VM) isoliert wird. Diese VM-Grenze verhindert den Zugriff auf hostbasierte Ressourcen, die von anderen Pods im Falle eines Container-Escapes verwendet werden. Dies ist eine übliche Methode, um containerisierte Anwendungen anzugreifen und Zugriff auf Ressourcen außerhalb des Containers zu erhalten.

  Die Verwendung von Amazon EKS ändert nichts an Ihren Verantwortlichkeiten im Rahmen des Modells der geteilten Verantwortung. Sie sollten die Konfiguration von Cluster-Sicherheits- und Governance-Kontrollen sorgfältig prüfen. Der sicherste Weg, eine Anwendung zu isolieren, besteht immer darin, sie in einem separaten Cluster auszuführen.

- Fargate-Profile unterstützen die Angabe von Subnetzen aus sekundären VPC-CIDR-Blöcken. Möglicherweise möchten Sie einen sekundären CIDR-Block angeben. Dies liegt daran, dass in einem Subnetz nur eine begrenzte Anzahl von IP-Adressen verfügbar ist. Daher gibt es auch eine begrenzte Anzahl von Pods, die im Cluster erstellt werden können. Durch die Verwendung verschiedener Subnetze für Pods können Sie die Anzahl der verfügbaren IP-Adressen erhöhen. Weitere Informationen finden Sie unter Hinzufügen von `IPv4`-CIDR-Blöcken zu einer VPC.
- Der Amazon EC2 Instance Metadata Service (IMDS) ist für Pods nicht verfügbar, die auf Fargate-Knoten bereitgestellt werden. Wenn Sie Pods haben, die in Fargate bereitgestellt werden und die IAM-Anmeldeinformationen benötigen, weisen Sie sie Ihren Pods mit IAM-Rollen für Servicekonten zu. Wenn Ihre Pods Zugriff auf andere Informationen benötigen, die über das IMDS verfügbar sind, müssen Sie diese Informationen in Ihre Pod-Spezifikation hart codieren. Dazu gehören die AWS-Region oder Availability Zone, in der ein bereitgestellt Pod wird.
- Sie können Fargate nicht Pods in AWS Outposts AWS Wavelength, oder AWS Local Zones bereitstellen.
- Amazon EKS muss Fargate Pods regelmäßig patchen, damit diese sicher bleiben. Wir versuchen, die Auswirkungen von Updates möglichst gering zu halten. Es kann jedoch vorkommen, dass Pods gelöscht werden müssen, wenn sie nicht erfolgreich entfernt wurden. Es gibt einige Maßnahmen, die Sie durchführen können, um Unterbrechungen zu minimieren. Weitere Informationen finden Sie unter Betriebssystem-Patching für Fargate.
- Das Amazon-VPC-CNI-Plugin für Amazon EKS ist standardmäßig auf Fargate-Knoten installiert. Sie können Alternative kompatible CNI-Plugins nicht mit Fargate-Knoten verwenden.
- Ein Pod, der in Fargate ausgeführt wird, mountet automatisch ein Amazon-EFS-Dateisystem. Sie können die dynamische Bereitstellung von persistenten Volumes nicht mit Fargate-Knoten verwenden, aber Sie können die statische Bereitstellung verwenden.
- Sie können Amazon-EBS-Volumes nicht in Fargate-Pods mounten.
- Sie können den Amazon-EBS-CSI-Controller auf Fargate-Knoten ausführen, aber der Amazon-EBS-CSI-Knoten DaemonSet kann nur auf Amazon-EC2-Instances ausgeführt werden.
- Nachdem ein Kubernetes-Job als `Completed` oder `Failed` gekennzeichnet wurde, existieren die vom Job erstellten Pods normalerweise weiter. Durch dieses Verhalten können Sie die Protokolle und Ergebnisse anzeigen. Bei Fargate entstehen jedoch Kosten, wenn Sie den Job nachher nicht bereinigen.

  Um das zugehörige automatisch zu löschen, Pods nachdem ein Job abgeschlossen ist oder fehlschlägt, können Sie mithilfe des time-to-live (TTL)-Controllers einen Zeitraum angeben. Das

folgende Beispiel veranschaulicht die Angabe von `.spec.ttlSecondsAfterFinished` im Job-Manifest.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: busybox
spec:
  template:
    spec:
      containers:
      - name: busybox
        image: busybox
        command: ["/bin/sh", "-c", "sleep 10"]
      restartPolicy: Never
  ttlSecondsAfterFinished: 60 # <-- TTL controller
```

Erste Schritte mit der AWS Fargate Verwendung von Amazon EKS

> **Important**
>
> AWS Fargate mit Amazon EKS ist in AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) nicht verfügbar.

In diesem Thema wird beschrieben, wie Sie AWS Fargate mit Pods der Ausführung Ihres Amazon EKS-Clusters beginnen können.

Wenn Sie den Zugriff auf den öffentlichen Endpunkt Ihres Clusters mithilfe von CIDR-Blöcken einschränken, empfehlen wir, dass Sie auch den privaten Endpunktzugriff aktivieren. Auf diese Weise können Fargate-Pods mit dem Cluster kommunizieren. Wenn der private Endpunkt nicht aktiviert ist, müssen die CIDR-Blöcke, die Sie für den öffentlichen Zugriff angeben, die Ausgangsquellen aus Ihrer VPC enthalten. Weitere Informationen finden Sie unter Zugriffskontrolle für den Amazon-EKS-Cluster-Endpunkt.

Voraussetzung

Einen vorhandenen -Cluster. Wenn Sie noch keinen Amazon-EKS-Cluster haben, lesen Sie Erste Schritte mit Amazon EKS.

Sicherstellen, dass vorhandene Knoten mit Fargate-Pods kommunizieren können

Wenn Sie mit einem neuen Cluster ohne Knoten oder einem Cluster nur mit verwalteten Knotengruppen arbeiten, können Sie mit Erstellen einer Fargate-Pod-Ausführungsrolle fortfahren.

Angenommen, Sie arbeiten mit einem vorhandenen Cluster, dem bereits Knoten zugeordnet sind. Stellen Sie sicher, dass Pods auf diesen Knoten frei mit den Pods kommunizieren können, die auf Fargate ausgeführt werden. Pods, die auf Fargate ausgeführt werden, werden automatisch so konfiguriert, dass die Cluster-Sicherheitsgruppe für den Cluster verwendet wird, dem sie zugeordnet sind. Stellen Sie sicher, dass alle vorhandenen Knoten im Cluster Datenverkehr an und von der Cluster-Sicherheitsgruppe senden und empfangen können. Verwaltete Knotengruppen werden automatisch für die Verwendung der Cluster-Sicherheitsgruppe konfiguriert, sodass Sie diese nicht ändern oder auf diese Kompatibilität überprüfen müssen.

Für bestehende Knotengruppen, die mit `eksctl` oder den von Amazon EKS verwalteten AWS CloudFormation Vorlagen erstellt wurden, können Sie die Cluster-Sicherheitsgruppe manuell zu den Knoten hinzufügen. Alternativ können Sie die Auto-Scaling-Gruppestartvorlage für die Knotengruppe ändern, um die Cluster-Sicherheitsgruppe an die Instances anzuhängen. Weitere Informationen finden Sie unter Ändern der Sicherheitsgruppen einer Instance im Amazon-VPC-Benutzerhandbuch.

Sie können im Abschnitt Netzwerk für den Cluster AWS Management Console nach einer Sicherheitsgruppe für Ihren Cluster suchen. Sie können dies auch mit dem folgenden AWS CLI Befehl tun. Wenn Sie diesen Befehl verwenden, ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

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

Erstellen einer Fargate-Pod-Ausführungsrolle

Wenn Ihr Cluster erstellt wird Pods AWS Fargate, müssen die Komponenten, die auf der Fargate-Infrastruktur ausgeführt werden, in Ihrem Namen AWS API-Aufrufe tätigen. Die Amazon-EKS-Pod-Ausführungsrolle stellt die entsprechenden IAM-Berechtigungen bereit. Informationen zum Erstellen einer AWS Fargate Pod Ausführungsrolle finden Sie unterIAM-Rolle zur Ausführung von Amazon-EKS-Pod.

> **Note**
>
> Wenn Sie den Cluster mithilfe von `eksctl` und der Option `--fargate` erstellt haben, verfügt der Cluster bereits über eine Pod-Ausführungsrolle, die Sie in der IAM-Konsole mit dem Muster `eksctl-`*`my-cluster`*`-FargatePodExecutionRole-`*`ABCDEFGHIJKL`* finden können. Wenn Sie Ihre Fargate-Profile mit `eksctl` anlegen, erstellt `eksctl` Ihre Pod-Ausführungsrolle, sofern noch keine erstellt wurde.

Erstellen eines Fargate-Profils für Ihren Cluster

Bevor Sie Pods planen können, die in Fargate in Ihrem Cluster ausgeführt werden, müssen Sie ein Fargate-Profil definieren, das angibt, welche Pods Fargate beim Start verwenden soll. Weitere Informationen finden Sie unter AWS Fargate Profil.

> **Note**
>
> Wenn Sie Ihren Cluster mithilfe von `eksctl` und der Option `--fargate` erstellt haben, wurde bereits ein Fargate-Profil für Ihren Cluster mit Selektoren für alle Pods in den `kube-system`- und `default`-Namespaces erstellt. Gehen Sie wie folgt vor, um Fargate-Profile für alle anderen Namespaces zu erstellen, die Sie mit Fargate verwenden möchten.

Sie können ein Fargate-Profil mit `eksctl` oder AWS Management Console erstellen.

eksctl

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre -Version mit dem folgenden Befehl überprüfen:

```
eksctl version
```

Eine Installations- und Upgrade-Anleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

Erstellen eines Fargate-Profils mit **`eksctl`**

Erstellen Sie Ihr Fargate-Profil mit dem folgenden `eksctl`-Befehl und ersetzen Sie jede *`example value`* durch Ihre eigenen Werte. Sie müssen einen Namespace angeben. Die Option `--labels` ist jedoch nicht erforderlich.

```
eksctl create fargateprofile \
    --cluster my-cluster \
    --name my-fargate-profile \
    --namespace my-kubernetes-namespace \
    --labels key=value
```

Sie können bestimmte Platzhalter für *`my-kubernetes-namespace`*- und *`key=value`*-Labels verwenden. Weitere Informationen finden Sie unter Fargate-Profil-Platzhalter.

AWS Management Console

Um ein Fargate-Profil für einen Cluster mit dem zu erstellen AWS Management Console

1. Öffnen Sie die Amazon EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Cluster aus, für den Sie ein Fargate-Profil erstellen möchten.
3. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.
4. Wählen Sie unter Fargate-Profile die Option Fargate-Profil hinzufügen aus.
5. Auf der Seite Konfigurieren des Fargate Profils führen Sie folgende Schritte aus:

    a. Geben Sie unter Name einen Namen für Ihr Fargate-Profil ein. Der Name muss eindeutig sein.

    b. Wählen Sie für Pod-Ausführungsrolle die Pod-Ausführungsrolle aus, die mit Ihrem Fargate-Profil verwendet werden soll. Es werden nur IAM-Rollen mit dem `eks-fargate-pods.amazonaws.com`-Service-Prinzipal angezeigt. Wenn keine Rollen aufgelistet sind, müssen Sie eine erstellen. Weitere Informationen finden Sie unter IAM-Rolle zur Ausführung von Amazon-EKS-Pod.

    c. Ändern Sie die ausgewählten Subnetze nach Bedarf.

    > **Note**
    >
    > Für Pods, die auf Fargate ausgeführt werden, werden nur private Subnetze unterstützt.

d. Für Tags können Sie Ihr Fargate-Profil wahlweise markieren. Diese Tags werden nicht an andere Ressourcen weitergegeben, die dem Profil zugeordnet sind, z. B. Pods.

e. Wählen Sie Weiter aus.

6. Führen Sie auf der Seite Configure Pod selection (-Auswahl konfigurieren) die folgenden Schritte aus:

   a. Geben Sie für Namespace einen Namespace ein, der mit Pods übereinstimmt.

      - Sie können bestimmte Namespaces für den Abgleich verwenden, z. B. **`kube-system`** oder **`default`**.
      - Sie können bestimmte Platzhalter verwenden (z. B. **`prod-*`**), um mehrere Namespaces abzugleichen (z. B. `prod-deployment` und `prod-test`). Weitere Informationen finden Sie unter Fargate-Profil-Platzhalter.

   b. (Optional) Fügen Sie dem Selektor Kubernetes-Labels hinzu. Fügen Sie sie insbesondere demjenigen hinzu, dem die Pods im angegebenen Namespace entsprechen müssen.

      - Sie können dem Selektor das Label **`infrastructure: fargate`** hinzufügen, sodass nur Pods im angegebenen Namespace, die ebenfalls die `infrastructure: fargate`-Kubernetes-Bezeichnung tragen, mit dem Selektor übereinstimmen.
      - Sie können bestimmte Platzhalter verwenden (z. B. **`key?: value?`**), um mehrere Namespaces abzugleichen (z. B. `keya: valuea` und `keyb: valueb`). Weitere Informationen finden Sie unter Fargate-Profil-Platzhalter.

   c. Wählen Sie Weiter.

7. Überprüfen Sie auf der Seite Überprüfen und erstellen die Informationen für Ihr -Profil und wählen Sie Erstellen aus.

Aktualisieren: CoreDNS

Standardmäßig ist CoreDNS zum Ausführen in der Amazon-EC2-Infrastruktur auf Amazon-EKS-Clustern konfiguriert. Wenn Sie Ihre Pods auf Fargate in Ihrem Cluster nur ausführen möchten, führen Sie die folgenden Schritte aus.

> **Note**
>
> Wenn Sie einen Cluster mit `eksctl` unter Verwendung von `--fargate`-Option erstellt haben, können Sie zu Nächste Schritte springen.

1. Erstellen Sie jedes Fargate-Profil für CoreDNS mit dem folgenden Befehl. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters, *111122223333* durch die ID Ihres Kontos, *AmazonEKSFargatePodExecutionRole* durch den Namen Ihrer Pod-Ausführungsrolle und *0000000000000001*, *0000000000000002* und *0000000000000003* durch die IDs Ihrer privaten Subnetze. Wenn Sie über keine Pod-Ausführungsrolle verfügen, müssen Sie zuerst eine erstellen.

   > **Important**
   >
   > Der Rollen-ARN darf als Pfad nur `/` enthalten. Wenn der Name Ihrer Rolle also beispielsweise `development/apps/my-role` lautet, müssen Sie ihn beim Angeben des ARN für die Rolle in `my-role` ändern. Das Format des Rollen-ARN muss `arn:aws:iam::111122223333:role/role-name` sein.

   ```
   aws eks create-fargate-profile \
       --fargate-profile-name coredns \
       --cluster-name my-cluster \
       --pod-execution-role-arn
    arn:aws:iam::111122223333:role/AmazonEKSFargatePodExecutionRole \
       --selectors namespace=kube-system,labels={k8s-app=kube-dns} \
       --subnets subnet-0000000000000001 subnet-0000000000000002
    subnet-0000000000000003
   ```

2. Verwenden Sie den folgenden Befehl, um die `eks.amazonaws.com/compute-type : ec2`-Anmerkung aus den CoreDNS-Pods zu entfernen.

   ```
   kubectl patch deployment coredns \
       -n kube-system \
       --type json \
       -p='[{"op": "remove", "path": "/spec/template/metadata/annotations/
   eks.amazonaws.com~1compute-type"}]'
   ```

Nächste Schritte

- Mit dem folgenden Workflow können Sie mit der Migration Ihrer vorhandenen Anwendungen zum Ausführen in Fargate beginnen.
  1. Erstellen Sie ein Fargate-Profil das dem Kubernetes-Namespace und den Kubernetes-Labels Ihrer Anwendung entspricht.
  2. Löschen Sie alle vorhandenen Pods, und erstellen Sie sie neu, sodass sie in Fargate geplant sind. Mit dem folgenden Befehl wird beispielsweise ein Rollout der `coredns`-Bereitstellung ausgelöst. Sie können den Namespace und den Bereitstellungstyp ändern, um Ihre spezifischen Pods zu aktualisieren.

     ```
     kubectl rollout restart -n kube-system deployment coredns
     ```
- Stellen Sie das Application Load Balancing auf Amazon EKS bereit, um Ingress-Objekte für Ihre Pods zuzulassen, die auf Fargate ausgeführt werden.
- Sie können Vertical Pod Autoscaler verwenden, um die CPU und den Speicher für Ihre Fargate-Pods anfänglich richtig zu dimensionieren, und dann Horizontal Pod Autoscaler verwenden, um diese Pods zu skalieren. Wenn Sie möchten, dass der Vertical Pod Autoscaler Pods mit höheren CPU- und Speicherkombinationen automatisch erneut in Fargate bereitstellt, stellen Sie den Modus für den Vertical-Pod-Autoscaler-Modus entweder auf `Auto` oder `Recreate`. Dies dient der Gewährleistung einer korrekten Funktion. Weitere Informationen finden Sie in der Vertical Pod Autoscaler-Dokumentation auf GitHub.
- Sie können den AWS Distro für OpenTelemetry(ADOT)-Kollektor zur Anwendungsüberwachung durch Befolgen dieser Anweisungen einrichten.

AWS Fargate Profil

> **Important**
>
> AWS Fargate mit Amazon EKS ist in AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) nicht verfügbar.

Bevor Sie Pods planen, die in Fargate in Ihrem Cluster ausgeführt werden, müssen Sie mindestens ein Fargate-Profil definieren, das angibt, welche Pods Fargate beim Start verwendet.

Mit dem Fargate-Profil kann ein Administrator deklarieren, welche Pods in Fargate ausgeführt werden. Sie können dies über die Profil-Selektoren tun. Sie können bis zu fünf Selektoren zu jedem Profil hinzufügen. Jeder Selektor muss einen Namespace enthalten. Der Selector kann auch Labels enthalten. Das Label-Feld besteht aus mehreren optionalen Schlüssel-Wert-Paaren. Pods, die den Selektoren entsprechen, werden auf Fargate geplant. Pods werden mit einem Namespace und den Labels abgeglichen, die im Selektor angegeben sind. Wenn ein Namespace-Selektor ohne Labels definiert ist, versucht Amazon EKS, alle Pods, die in diesem Namespace ausgeführt werden, mithilfe des Profils in Fargate zu planen. Wenn a to-be-scheduled Pod mit einem der Selektoren im Fargate-Profil übereinstimmt, Pod ist dies auf Fargate geplant.

Wenn ein Pod mit mehreren Fargate-Profilen übereinstimmt, können Sie angeben, welches Profil ein Pod verwendet, indem Sie das folgende Kubernetes-Label zur Pod-Spezifikation hinzufügen: `eks.amazonaws.com/fargate-profile: my-fargate-profile`. Der Pod muss mit einem Selektor in diesem Profil übereinstimmen, damit er in Fargate geplant werden kann. Kubernetes-Affinitäts- und Anti-Affinitätsregeln werden nicht berücksichtigt und sind bei Amazon-EKS-Fargate-Pods nicht erforderlich.

Wenn Sie ein Fargate-Profil erstellen, müssen Sie eine Pod-Ausführungsrolle angeben. Diese Ausführungsrolle ist für die Amazon-EKS-Komponenten vorgesehen, die auf der Fargate-Infrastruktur mit dem Profil ausgeführt werden. Sie wird zur rollenbasierten Kubernetes-Zugriffssteuerung (RBAC) des Clusters zur Autorisierung hinzugefügt. Auf diese Weise kann sich `kubelet`, das in der Fargate-Infrastruktur ausgeführt wird, bei Ihrem Amazon-EKS-Cluster registrieren und als Knoten in Ihrem Cluster angezeigt werden. Die Pod-Ausführungsrolle bietet auch IAM-Berechtigungen für die Fargate-Infrastruktur, um Lesezugriff auf Amazon ECR-Image-Repositorys zu ermöglichen. Weitere Informationen finden Sie unter IAM-Rolle zur Ausführung von Amazon-EKS-Pod.

Fargate-Profile können nicht geändert werden. Sie können jedoch ein neues aktuelles Profil erstellen, um ein vorhandenes Profil zu ersetzen, und dann das Original löschen.

> **Note**
>
> Alle Pods, die mit einem Fargate-Profil ausgeführt werden, werden gestoppt und als ausstehend gekennzeichnet, wenn das Profil gelöscht wird.

Wenn Fargate-Profile in einem Cluster den Status `DELETING` haben, müssen Sie warten, bis dieses Fargate-Profil gelöscht wurde, bevor Sie andere Profile in diesem Cluster erstellen können.

Amazon EKS und Fargate versuchen, Pods auf jedes der im Fargate-Profil definierten Subnetze zu verteilen. Es kann jedoch zu einer ungleichmäßigen Verteilung kommen. Wenn Sie eine gleichmäßige Verteilung benötigen, verwenden Sie zwei Fargate-Profile. Eine gleichmäßige Verteilung ist in Szenarien wichtig, in denen Sie zwei Replikate bereitstellen möchten und keine Ausfallzeiten wünschen. Wir empfehlen, dass jedes Profil nur ein Subnetz hat.

Fargate-Profilkomponenten

Die folgenden Komponenten sind in einem Fargate-Profil enthalten.

Pod-Ausführungsrolle

Wenn Ihr Cluster erstellt wird Pods AWS Fargate, muss der`kubelet`, der auf der Fargate-Infrastruktur läuft, in Ihrem Namen AWS API-Aufrufe tätigen. Beispielsweise muss er Aufrufe tätigen, um Container-Images aus Amazon ECR zu ziehen. Die Amazon-EKS-Pod-Ausführungsrolle stellt die entsprechenden IAM-Berechtigungen bereit.

Wenn Sie ein Fargate-Profil erstellen, müssen Sie eine Pod-Ausführungsrolle angeben, die mit Ihren Pods verwendet werden soll. Diese Rolle wird zur rollenbasierten Kubernetes-Zugriffssteuerung (RBAC) des Clusters zur Autorisierung hinzugefügt. Auf diese Weise kann sich `kubelet`, das in der Fargate-Infrastruktur ausgeführt wird, bei Ihrem Amazon-EKS-Cluster registrieren und als Knoten in Ihrem Cluster angezeigt werden. Weitere Informationen finden Sie unter IAM-Rolle zur Ausführung von Amazon-EKS-Pod.

Subnetze

Die IDs von Subnetzen, in denen Pods gestartet werden, die dieses Profil verwenden. Derzeit werden Pods, welche auf Fargate ausgeführt werden, keine öffentlichen IP-Adressen zugewiesen. Daher werden für diesen Parameter nur private Subnetze ohne direkte Route zu einem Internet-Gateway akzeptiert.

Selektoren

Die Selektoren, mit denen Pods, die dieses Fargate-Profil verwenden, übereinstimmen sollen. Sie können bis zu fünf Selektoren in einem Fargate-Profil angeben. Die Selektoren bestehen aus folgenden Komponenten:

- Namespace – Sie müssen einen Namespace für einen Selektor angeben. Der Selektor stimmt nur mit Pods überein, die in diesem Namespace erstellt werden. Sie können jedoch mehrere Selektoren erstellen, um mehrere Namespaces zu adressieren.

- Labels – Optional können Sie Kubernetes-Labels angeben, mit denen der Selektor übereinstimmen muss. Der Selektor stimmt nur mit Pods überein, die alle im Selektor angegebenen Labels aufweisen.

Fargate-Profil-Platzhalter

Zusätzlich zu den Zeichen, die von Kubernetes erlaubt sind, dürfen Sie **`*`** und **`?`** in den Selektorkriterien für Namespaces, Label-Schlüssel und Label-Werte verwenden:

- **`*`** steht für kein, ein oder mehrere Zeichen. Zum Beispiel kann **`prod*`** `prod` und `prod-metrics` darstellen.
- **`?`** steht für ein einzelnes Zeichen (z. B. kann **`value?`** `valuea` darstellen). Es kann jedoch nicht `value` und `value-a` darstellen, da ? nur genau ein Zeichen darstellen kann.

Diese Platzhalterzeichen können an jeder Position und in Kombination verwendet werden (z. B. **`prod*`**, **`*dev`** und **`frontend*?`**). Andere Platzhalter und Formen des Musterabgleichs, wie z. B. reguläre Ausdrücke, werden nicht unterstützt.

Wenn mehrere übereinstimmende Profile für den Namespace und die Labels in der Pod-Spezifikation vorhanden sind, übernimmt Fargate das Profil basierend auf einer alphanumerischen Sortierung nach Profilnamen. Wenn zum Beispiel Profil A (mit dem Namen `beta-workload`) und Profil B (mit dem Namen `prod-workload`) passende Selektoren für die zu startenden Pods haben, wählt Fargate Profil A (`beta-workload`) für die Pods aus. Die Pods haben Labels mit Profil A auf den Pods (z. B. `eks.amazonaws.com/fargate-profile=beta-workload`).

Wenn Sie vorhandene Fargate-Pods zu neuen Profilen migrieren möchten, die Platzhalter verwenden, gibt es zwei Möglichkeiten, dies zu tun:

- Erstellen Sie ein neues Profil mit passenden Selektoren und löschen Sie dann die alten Profile. Mit alten Profilen gekennzeichnete Pods werden in neue übereinstimmende Profile verschoben.
- Wenn Sie Workloads migrieren möchten, sich aber nicht sicher sind, welche Fargate-Labels sich auf jedem Fargate-Pod befinden, können Sie die folgende Methode verwenden. Erstellen Sie ein neues Profil mit einem Namen, das zuerst alphanumerisch unter den Profilen auf demselben Cluster sortiert. Recyceln Sie dann die Fargate-Pods, die in neue Profile migriert werden müssen.

Erstellen eines Fargate-Profils

In diesem Thema wird beschrieben, wie Sie ein Fargate-Profil erstellen. Sie müssen auch eine Pod-Ausführungsrolle erstellt haben, die für Ihr Fargate-Profil verwendet werden soll. Weitere Informationen finden Sie unterIAM-Rolle zur Ausführung von Amazon-EKS-Pod. Podsdie auf Fargate laufen, werden nur in privaten Subnetzen mit NAT-Gateway-Zugriff auf AWS-Services, aber nicht mit direkter Route zu einem Internet Gateway unterstützt. Dies ist so, da in der VPC Ihres Clusters private Subnetze verfügbar sein müssen. Sie können ein Profil mit `eksctl` oder der AWS Management Console aus.

Für diesen Vorgang ist `eksctl` Version `0.183.0` oder höher erforderlich. Sie können Ihre -Version mit dem folgenden Befehl überprüfen:

```
eksctl version
```

Eine Installations- und Upgrade-Anleitung für `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

eksctl

Erstellen eines Fargate-Profils mit **`eksctl`**

Erstellen Sie Ihr Fargate-Profil mit dem folgenden `eksctl`-Befehl und ersetzen Sie jede *`example value`* durch Ihre eigenen Werte. Sie müssen einen Namespace angeben. Die Option `--labels` ist jedoch nicht erforderlich.

```
eksctl create fargateprofile \
    --cluster my-cluster \
    --name my-fargate-profile \
    --namespace my-kubernetes-namespace \
    --labels key=value
```

Sie können bestimmte Platzhalter für *`my-kubernetes-namespace`*- und *`key=value`*-Labels verwenden. Weitere Informationen finden Sie unter Fargate-Profil-Platzhalter.

AWS Management Console

Um ein Fargate-Profil für einen Cluster mit dem zu erstellen AWS Management Console

1. Öffnen Sie die Amazon EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

2. Wählen Sie den Cluster aus, für den Sie ein Fargate-Profil erstellen möchten.
3. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.
4. Wählen Sie unter Fargate-Profile die Option Fargate-Profil hinzufügen aus.
5. Auf der Seite Konfigurieren des Fargate Profils führen Sie folgende Schritte aus:

   a. Geben Sie unter Name einen eindeutigen Namen für Ihr Fargate-Profil ein, wie z. B. ***my-profile***.

   b. Wählen Sie für Pod-Ausführungsrolle die Pod-Ausführungsrolle aus, die mit Ihrem Fargate-Profil verwendet werden soll. Es werden nur IAM-Rollen mit dem `eks-fargate-pods.amazonaws.com`-Service-Prinzipal angezeigt. Wenn keine Rollen aufgelistet sind, müssen Sie eine erstellen. Weitere Informationen finden Sie unter IAM-Rolle zur Ausführung von Amazon-EKS-Pod.

   c. Ändern Sie die ausgewählten Subnetze nach Bedarf.

   > **Note**
   >
   > Für Pods, die auf Fargate ausgeführt werden, werden nur private Subnetze unterstützt.

   d. Für Tags können Sie Ihr Fargate-Profil wahlweise markieren. Diese Tags werden nicht an andere Ressourcen weitergegeben, die dem Profil zugeordnet sind, z. B. Pods.

   e. Wählen Sie Weiter aus.

6. Führen Sie auf der Seite Configure Pod selection (-Auswahl konfigurieren) die folgenden Schritte aus:

   a. Geben Sie für Namespace einen Namespace ein, der mit Pods übereinstimmt.

   - Sie können bestimmte Namespaces für den Abgleich verwenden, z. B. **`kube-system`** oder **`default`**.
   - Sie können bestimmte Platzhalter verwenden (z. B. **`prod-*`**), um mehrere Namespaces abzugleichen (z. B. `prod-deployment` und `prod-test`). Weitere Informationen finden Sie unter Fargate-Profil-Platzhalter.

   b. (Optional) Fügen Sie dem Selektor Kubernetes-Labels hinzu. Fügen Sie sie insbesondere demjenigen hinzu, dem die Pods im angegebenen Namespace entsprechen müssen.

- Sie können dem Selektor das Label **`infrastructure: fargate`** hinzufügen, sodass nur Pods im angegebenen Namespace, die ebenfalls die `infrastructure: fargate`-Kubernetes-Bezeichnung tragen, mit dem Selektor übereinstimmen.
- Sie können bestimmte Platzhalter verwenden (z. B. **`key?: value?`**), um mehrere Namespaces abzugleichen (z. B. `keya: valuea` und `keyb: valueb`). Weitere Informationen finden Sie unter Fargate-Profil-Platzhalter.

c. Wählen Sie Weiter.

7. Überprüfen Sie auf der Seite Überprüfen und erstellen die Informationen für Ihr -Profil und wählen Sie Erstellen aus.

Löschen eines Fargate-Profils

In diesem Thema wird beschrieben, wie Sie ein Fargate-Profil löschen.

Wenn Sie ein Fargate-Profil löschen, werden alle Pods gelöscht, die in Fargate mit dem Profil geplant wurden. Wenn diese Pods mit einem anderen Fargate-Profil übereinstimmen, werden sie mit diesem Profil in Fargate geplant. Wenn sie nicht mehr mit einem der Fargate-Profile übereinstimmen, werden sie nicht in Fargate geplant und ggf. als ausstehend gekennzeichnet.

Nur ein Fargate-Profil in einem Cluster kann sich jeweils im `DELETING`-Status befinden. Warten Sie, bis das Löschen eines Fargate-Profils abgeschlossen ist, bevor Sie andere Profile in diesem Cluster löschen können.

Sie können ein Profil mit `eksctl` AWS Management Console, dem oder dem AWS CLI löschen. Wählen Sie die Registerkarte mit dem Namen des Werkzeugen aus, mit dem Sie Ihr Profil löschen möchten.

eksctl

So löschen Sie ein Fargate-Profil mit **`eksctl`**

Verwenden Sie den folgenden Befehl, um ein Profil aus einem Cluster zu löschen. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte.

```
eksctl delete fargateprofile  --name my-profile --cluster my-cluster
```

AWS Management Console

Um ein Fargate-Profil aus einem Cluster zu löschen mit dem AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus. Wählen Sie in der Clusterliste den Cluster aus, in dem Sie das Fargate-Profil löschen möchten.
3. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.
4. Wählen Sie das zu löschende Fargate-Profil und dann Delete (Löschen) aus.
5. Geben Sie auf der Seite Delete Fargate profile (Fargate-Profil löschen) den Namen des Profils ein und wählen Sie dann Delete (Löschen) aus.

AWS CLI

So löschen Sie ein Fargate-Profil mit AWS CLI

Verwenden Sie den folgenden Befehl, um ein Profil aus einem Cluster zu löschen. Ersetzen Sie jede *example value* durch Ihre eigenen Werte.

```
aws eks delete-fargate-profile --fargate-profile-name my-profile --cluster-name my-cluster
```

Fargate-Pod-Konfiguration

> **Important**
>
> AWS Fargate mit Amazon EKS ist in AWS GovCloud (USA-Ost) und AWS GovCloud (USA-West) nicht verfügbar.

In diesem Abschnitt werden einige der eindeutigen Pod-Konfigurationsdetails für die Ausführung von Kubernetes-Pods in AWS Fargate beschrieben.

Pod-CPU und -Arbeitsspeicher

Mit Kubernetes können Sie Anforderungen, eine Mindestmenge an vCPUs und Arbeitsspeicherressourcen für jedem Container in einem Pod definieren. Pods sind

von Kubernetes geplant, um sicherzustellen, dass mindestens die angeforderten Ressourcen für jeden Podauf der Datenverarbeitungsressource verfügbar sind. Weitere Informationen finden Sie unter Managing Compute Resources for Containers (Verwalten von Datenverarbeitungsressourcen für Container) in der Kubernetes-Dokumentation.

> **Note**
>
> Da Amazon-EKS-Fargate nur einen Pod pro Knoten ausführt, tritt das Szenario des Räumens von Pods bei weniger Ressourcen nicht auf. Alle Amazon-EKS-Fargate-Pods laufen mit garantierter Priorität, daher müssen die angeforderte CPU und der Arbeitsspeicher dem Limit für alle Container entsprechen. Weitere Informationen finden Sie unter Konfigurieren von Servicequalität für Pods in der Kubernetes-Dokumentation.

Wenn Pods in Fargate geplant sind, bestimmen die vCPU- und Speicherreservierungen innerhalb der Pod-Spezifikation, wie viel CPU und Speicher für den Pod bereitgestellt werden sollen.

- Die maximale Anforderung von Init-Containern wird verwendet, um die vCPU- und Speicheranforderungen für die Init-Anforderung zu bestimmen.
- Anforderungen für alle lang laufenden Container werden addiert, um die Anforderungen an die vCPU und den Arbeitsspeicher für lange laufende Anforderungen zu bestimmen.
- Der größere der beiden zuvor genannten Werte wird für die vCPU und die Speicheranforderung für Ihren Pod ausgewählt.
- Fargate fügt 256 MB zur Speicherreservierung jedes Pod für die erforderlichen Kubernetes-Komponenten (`kubelet`, `kube-proxy` und `containerd`) hinzu.

Fargate rundet auf die folgende Rechenkonfiguration auf, die der Summe von vCPU und Speicheranforderungen am nächsten liegt, damit Pods immer über die Ressourcen verfügen, die sie für ihre Ausführung benötigen.

Wenn Sie keine vCPU- und Speicherkombination angeben, wird die kleinste verfügbare Kombination verwendet (0,25 vCPU und 0,5 GB Arbeitsspeicher).

Die folgende Tabelle zeigt die vCPU- und Speicherkombinationen, die für Pods verfügbar sind, die in Fargate ausgeführt werden.

vCPU-Wert	Speicherwert
0,25 vCPU	0,5 GB, 1 GB, 2 GB
0,5 vCPU	1 GB, 2 GB, 3 GB, 4 GB
1 vCPU	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB
2 vCPU	Zwischen 4 GB und 16 GB in 1-GB-Schritten
4 vCPU	Zwischen 8 GB und 30 GB in 1-GB-Schritten
8 vCPU	Zwischen 16 GB und 60 GB in 4-GB-Schritten
16 vCPU	Zwischen 32 GB und 120 GB in 8-GB-Schritten

Der für die Kubernetes-Komponenten reservierte zusätzliche Arbeitsspeicher kann dazu führen, dass eine Fargate-Aufgabe mit mehr vCPUs als angefordert bereitgestellt wird. Bei einer Anforderung von 1 vCPU und 8 GB Speicher werden beispielsweise 256 MB zu ihrer Speicheranforderung hinzugefügt und eine Fargate-Aufgabe mit 2 vCPUs und 9 GB Speicher bereitgestellt, da keine Aufgabe mit 1 vCPU und 9 GB Speicher verfügbar ist.

Es gibt keine Korrelation zwischen der Größe des Pod, der auf Fargate ausgeführt wird, und der von Kubernetes gemeldeten Knotengröße mit `kubectl get nodes`. Die gemeldete Knotengröße ist oft größer als die Kapazität des Pod. Sie können die Pod-Kapazität mit dem folgenden Befehl überprüfen. Ersetzen Sie *`default`* mit Ihrem Pod-Namespace und *`pod-name`* mit Ihrem Pod-Namen.

```
kubectl describe pod --namespace default pod-name
```

Eine Beispielausgabe sieht wie folgt aus.

```
[...]
annotations:
    CapacityProvisioned: 0.25vCPU 0.5GB
[...]
```

Die `CapacityProvisioned`-Anmerkung stellt die erzwungene Pod-Kapazität dar und bestimmt die Kosten Ihres Pod, der auf Fargate ausgeführt wird. Preisinformationen für die Computing-Konfigurationen finden Sie unter AWS Fargate-Preise.

Fargate-Speicher

Ein Pod, der in Fargate ausgeführt wird, mountet automatisch ein Amazon-EFS-Dateisystem. Sie können die dynamische Bereitstellung von persistenten Volumes nicht mit Fargate-Knoten verwenden, aber Sie können die statische Bereitstellung verwenden. Weitere Informationen finden Sie unter Amazon-EFS-CSI-Treiber auf GitHub.

Bei der Bereitstellung erhält jede Pod, die auf Fargate läuft, standardmäßig 20 GB flüchtigem Speicher. Diese Art von Speicher wird gelöscht, nachdem Pod stoppt. Bei neu auf Fargate gestarteten Pods ist die Verschlüsselung des flüchtigen Speichervolumens standardmässig aktiviert. Der flüchtige Pod-Speicher wird mit einem AES-256-Verschlüsselungsalgorithmus mit AWS Fargate-verwalteten Schlüsseln verschlüsselt.

> **Note**
>
> Der standardmäßig verwendbare Speicher für Amazon-EKS-Pods, die auf Fargate laufen, ist weniger als 20 GB groß. Dies liegt daran, dass ein Teil des Speicherplatzes genutzt wird von `kubelet` und anderen Kubernetes-Modulen, die in den Pod geladen werden.

Sie können die Gesamtmenge des flüchtigen Speichers bis zu einem Maximum von 175 GB erhöhen. Um die Größe mit Kubernetes zu konfigurieren, spezifizieren Sie die Anfragen von `ephemeral-storage`-Ressource für jeden Container in einem Pod. Wenn Kubernetes Pods plant, stellt es sicher, dass die Summe der Ressourcenanfragen für jeden Pod geringer ist als die Kapazität der Fargate-Aufgabe. Weitere Informationen finden Sie unter Ressourcenverwaltung für Pods und Container in der Kubernetes-Dokumentation.

Amazon EKS Fargate stellt mehr kurzlebigen Speicher bereit, als für die Systemnutzung angefordert wurde. Eine Anforderung von 100 GB stellt beispielsweise eine Fargate-Aufgabe mit 115 GB flüchtigem Speicher bereit.

Betriebssystem-Patching für Fargate

> **Important**
>
> AWS Fargate mit Amazon EKS ist in AWS GovCloud (USA-Ost) und AWS GovCloud (USA-West) nicht verfügbar.

Amazon EKS patcht regelmäßig das Betriebssystem für AWS Fargate-Knoten, damit sie sicher bleiben. Im Rahmen des Patching-Prozesses werden die Knoten recycelt, um Betriebssystem-Patches zu installieren. Updates werden so durchgeführt, dass sie Ihre Services möglichst wenig beeinflussen. Wenn Pods jedoch nicht erfolgreich bereinigt wurden, kann es sein, dass sie gelöscht werden müssen. Die folgenden Maßnahmen können Ihnen helfen, potenzielle Unterbrechungen zu minimieren:

- Legen Sie geeignete Pod Disruption Budgets (PBDs, Pod-Unterbrechungsbudgets) fest, um die Anzahl der Pods, die gleichzeitig ausfallen, zu steuern.
- Erstellen Sie Amazon- EventBridge Regeln, um fehlgeschlagene Bereinigungen zu behandeln, bevor die gelöscht Pods werden.
- Erstellen Sie eine Benachrichtigungskonfiguration in AWS-Benutzerbenachrichtigungen.

Amazon EKS arbeitet eng mit der Kubernetes-Community zusammen, um Fehlerbehebungen und Sicherheitspatches so schnell wie möglich bereitzustellen. Alle Fargate-Pods starten mit der aktuellsten Kubernetes-Patch-Version, die über die über Amazon EKS für die Kubernetes-Version Ihres Clusters verfügbar ist. Wenn Sie einen Pod mit einer älteren Patch-Version haben, recycelt Amazon EKS ihn möglicherweise, um ihn auf die neueste Version zu aktualisieren. Das stellt sicher, dass Ihre Pods mit den neuesten Sicherheitsupdates ausgestattet sind. Auf diese Weise bleiben Sie bei kritischen Common Vulnerabilities and Exposures (CVE)-Problemen auf dem Laufenden, um Sicherheitsrisiken zu reduzieren.

Um die Anzahl der Pods zu begrenzen, die beim Recyceln von Pods gleichzeitig ausfallen, können Sie Pod-Unterbrechungsbudgets (PDBs) festlegen. Sie können PDBs verwenden, um die Mindestverfügbarkeit basierend auf den Anforderungen jeder Ihrer Anwendungen zu definieren und gleichzeitig Aktualisierungen zuzulassen. Weitere Informationen finden Sie unter Specifying a Disruption Budget for your Application (Angabe eines Unterbrechungsbudgets in Ihrer Anwendung) in der Kubernetes-Dokumentation.

Amazon EKS verwendet die Bereinigungs-AMI, um den Pod sicher zu leeren. Dabei werden die PDBs eingehalten, die Sie für die Anwendung festgelegt haben. Pods werden von der Availability Zone bereinigt, um Auswirkungen zu minimieren. Wenn die Bereinigung erfolgreich ist, erhält der neue Pod den aktuellsten Patch und es sind keine weiteren Maßnahmen erforderlich.

Wenn die Bereinigung für einen Pod fehlschlägt, sendet Amazon EKS ein Ereignis an Ihr Konto, das Details über die Pods enthält, deren Bereinigung fehlgeschlagen ist. Sie können vor der geplanten Beendigungszeit auf die Nachricht reagieren. Die spezifische Zeit variiert je nach Dringlichkeit des Patches. Zum festgelegten Zeitpunkt versucht Amazon EKS erneut, die Pods zu bereinigen. Diesmal wird jedoch kein neues Ereignis gesendet, wenn die Bereinigung fehlschlägt. Wenn die Bereinigung erneut fehlschlägt, werden Ihre vorhandenen Pods regelmäßig gelöscht, damit neue Pods den aktuellsten Patch haben.

Im Folgenden sehen Sie ein Beispielereignis, das Sie erhalten könnten, wenn die Pod-Bereinigung fehlschlägt. Es enthält Informationen über den Cluster, den Pod-Namen, den Pod-Namespace, das Fargate-Profil und die geplante Beendigungszeit.

```
{
    "version": "0",
    "id": "12345678-90ab-cdef-0123-4567890abcde",
    "detail-type": "EKS Fargate Pod Scheduled Termination",
    "source": "aws.eks",
    "account": "111122223333",
    "time": "2021-06-27T12:52:44Z",
    "region": "region-code",
    "resources": [
        "default/my-database-deployment"
    ],
    "detail": {
        "clusterName": "my-cluster",
        "fargateProfileName": "my-fargate-profile",
        "podName": "my-pod-name",
        "podNamespace": "default",
        "evictErrorMessage": "Cannot evict pod as it would violate the pod's disruption
 budget",
        "scheduledTerminationTime": "2021-06-30T12:52:44.832Z[UTC]"
    }
}
```

Einer der Gründe für ein Bereinigungsfehlerereignis kann sein, dass einem Pod mehrere PDBs zugeordnet sind. In diesem Fall gibt das Ereignis die folgende Fehlernachricht zurück.

```
"evictErrorMessage": "This pod has multiple PodDisruptionBudget, which the eviction
 subresource does not support",
```

Basierend auf diesem Ereignis können Sie eine gewünschte Aktion erstellen. Sie können beispielsweise Ihr Pod-Unterbrechungsbudget (PDB) anpassen, um zu steuern, wie die Pods bereinigt werden. Nehmen Sie zum Beispiel an, dass Sie ein PDB starten, dass den Zielprozentsatz von verfügbaren Pods angibt. Bevor Ihre Pods während eines Upgrades automatisch beendet werden, können Sie das PDB an einen unterschiedlichen Prozentsatz von Pods anpassen. Um dieses Ereignis zu empfangen, müssen Sie eine Amazon- EventBridge Regel in der AWS-Konto und der erstellenAWS-Region, zu der der Cluster gehört. Die Regel muss das folgende benutzerdefinierte Muster verwenden. Weitere Informationen finden Sie unter Erstellen von Amazon- EventBridge Regeln, die auf Ereignisse reagieren im Amazon- EventBridge Benutzerhandbuch.

```
{
  "source": ["aws.eks"],
  "detail-type": ["EKS Fargate Pod Scheduled Termination"]
}
```

Es kann ein geeignetes Ziel zur Erfassung durch das Ereignis festgelegt werden. Eine vollständige Liste der verfügbaren Ziele finden Sie unter Amazon- EventBridge Ziele im Amazon- EventBridge Benutzerhandbuch. Sie können eine Benachrichtigungskonfiguration in AWS-Benutzerbenachrichtigungen erstellen. Wenn Sie die AWS Management Console verwenden, um die Benachrichtigung zu erstellen, wählen Sie unter Ereignisregeln Elastic Kubernetes Service (EKS) als AWS-Service-Namen und EKS Fargate Pod Scheduled Termination als Ereignistyp aus. Weitere Informationen finden Sie unter Erste Schritte mit AWS Benutzerbenachrichtigungen im AWS-Benutzerhandbuch für Benutzerbenachrichtigungen.

Fargate-Metriken

> **Important**
>
> AWS Fargate mit Amazon EKS ist in AWS GovCloud (USA Ost) und AWS GovCloud (USA West) nicht verfügbar.

Sie können Systemmetriken und CloudWatch-Nutzungsmetriken für AWS Fargate erfassen.

Anwendungsmetriken

Für Anwendungen, die in Amazon EKS und AWS Fargate ausgeführt werden, können Sie AWS Distro for OpenTelemetry (ADOT) verwenden. Mit ADOT können Sie Systemmetriken sammeln und an Dashboards von CloudWatch Container Insights senden. Weitere Informationen zum Einstieg in ADOT für Anwendungen, die in Fargate ausgeführt werden, finden Sie unter Verwenden von CloudWatch Container Insights mit AWS Distro for OpenTelemetry in der ADOT-Dokumentation.

Nutzungsmetriken

Sie können CloudWatch-Nutzungsmetriken verwenden, um einen Einblick in die Ressourcennutzung Ihres Kontos zu gewähren. Verwenden Sie diese Metriken, um Ihre aktuelle Servicenutzung für CloudWatch-Diagramme und -Dashboards zu visualisieren.

AWS Fargate-Nutzungsmetriken entsprechen AWS-Servicekontingenten. Sie können Alarme konfigurieren, mit denen Sie benachrichtigt werden, wenn sich Ihre Nutzung einem Servicekontingent nähert. Weitere Hinweise zu Servicekontingenten für Fargate finden Sie unter Amazon-EKS-Service-Quotas.

AWS Fargate veröffentlicht die folgenden Metriken im `AWS/Usage`-Namespace.

Metrik	Beschreibung
`ResourceCount`	Die Gesamtanzahl der angegebenen Ressourcen, die in Ihrem Konto ausgeführt werden. Die Ressource wird durch die Dimensionen definiert, die der Metrik zugeordnet sind.

Die folgenden Dimensionen werden verwendet, um die Nutzungsmetriken zu verfeinern, die von AWS Fargate veröffentlicht werden.

Dimension	Beschreibung
`Service`	Der Name des AWS-Service, der die Ressource enthält. Für AWS Fargate-Nutzungsmetriken lautet der Wert für diese Dimension `Fargate`.
`Type`	Der Typ von Entität, die gemeldet wird. Derzeit ist der einzige gültige Wert für AWS Fargate-Nutzungsmetriken `Resource`.

Dimension	Beschreibung
`Resource`	Der Typ der Ressource, die ausgeführt wird. Derzeit gibt AWS Fargate Informationen zu Ihrer Fargate-on-Demand-Nutzung zurück. Der Ressourcenwert für Fargate On-Demand-Nutzung ist `OnDemand`. **Note** Die Fargate On-Demand-Nutzung kombiniert Amazon EKS Pods mit Fargate, Amazon-ECS-Aufgaben mit dem Fargate-Starttyp und Amazon-ECS-Aufgaben mithilfe des `FARGATE`-Kapazitätsanbieters.
`Class`	Die Klasse der nachverfolgten Ressource. Derzeit verwendet AWS Fargate die Klassendimension nicht.

Erstellen eines CloudWatch-Alarms zur Überwachung von Fargate Ressourcennutzungsmetriken

AWS Fargate stellt CloudWatch-Nutzungsmetriken bereit, die den AWS-Service-Quotas für Fargate On-Demand und Ressourcennutzung entsprechen. In der Service-Quotas-Konsole können Sie Ihre Nutzung in einem Diagramm visualisieren. Sie können auch Alarme konfigurieren, die Sie warnen, wenn sich Ihre Nutzung einem Service Quotas nähert. Weitere Informationen finden Sie unter Fargate-Metriken.

Führen Sie die folgenden Schritte aus, um einen CloudWatch-Alarm basierend auf den Fargate-Ressourcen-Nutzungsmetriken zu erstellen.

So erstellen Sie einen Alarm basierend auf Ihren Fargate-Nutzungskontingenten (AWS Management Console)

1. Öffnen Sie die Service-Quotas-Konsole unter https://console.aws.amazon.com/servicequotas/.
2. Wählen Sie im linken Navigationsbereich AWS services aus.
3. Suchen Sie in der Liste der AWS-Services nach AWS Fargate und wählen Sie es aus.
4. Wählen Sie in der Liste Service quotas (Servicekontingente) das Fargate-Nutzungskontingent aus, für das Sie einen Alarm erstellen möchten.

5. Wählen Sie im Abschnitt mit den Amazon-CloudWatch-Alarmen die Option Create (Erstellen) aus.
6. Wählen Sie bei Alarmschwellenwert den Prozentsatz des angewendeten Kontingentwerts aus, den Sie als Alarmwert festlegen möchten.
7. Geben Sie bei Alarmname einen Namen für den Alarm ein und wählen Sie dann Erstellen aus.

Fargate-Protokollierung

> **Important**
>
> AWS Fargate mit Amazon EKS ist in AWS GovCloud (USA-Ost) und AWS GovCloud (USA-West) nicht verfügbar.

Amazon EKS auf Fargate bietet einen integrierten Log-Router basierend auf Fluent Bit. Dies bedeutet, dass Sie einen Fluent Bit-Container nicht explizit als Sidecar ausführen. Amazon übernimmt die Ausführung für Sie. Alles, was Sie tun müssen, ist den Log-Router zu konfigurieren. Die Konfiguration erfolgt über ein dediziertes `ConfigMap`, das die folgenden Kriterien erfüllen muss:

- Benannte `aws-logging`
- Erstellt in einem dedizierten Namespace namens `aws-observability`
- Mehr als 5 300 Zeichen sind nicht zulässig.

Sobald Sie die `ConfigMap` erstellt haben, erkennt Amazon EKS auf Fargate sie automatisch und konfiguriert den Protokollrouter damit. Fargate verwendet eine Version von AWS für Fluent Bit, eine Upstream-konforme Distribution von , die von Fluent Bit verwaltet wird AWS. Weitere Informationen finden Sie unter AWS für Fluent Bit auf GitHub.

Mit dem Protokoll-Router können Sie die Bandbreite der Services in AWS für Protokollanalysen und -speicher verwenden. Sie können Protokolle von Fargate direkt an Amazon CloudWatch, Amazon OpenSearch Service, streamen. Sie können Protokolle auch über Amazon Data Firehose an Ziele wie Amazon S3, Amazon Kinesis Data Streams und Partnertools streamen. https://aws.amazon.com/kinesis/data-firehose/

Voraussetzungen

- Ein vorhandenes Fargate-Profil, das einen vorhandenen Kubernetes-Namespace angibt, in dem Sie Fargate-Pods bereitstellen. Weitere Informationen finden Sie unter Erstellen eines Fargate-Profils für Ihren Cluster.
- Eine vorhandene Fargate-Pod-Ausführungsrolle. Weitere Informationen finden Sie unter Erstellen einer Fargate-Pod-Ausführungsrolle.

Router-Konfiguration protokollieren

So konfigurieren Sie den Protokollrouter

Ersetzen Sie in den folgenden Schritten jede *example value* durch Ihre eigenen Werte.

1. Erstellen Sie einen dedizierten Kubernetes-Namespace mit dem Namen `aws-observability`.

   a. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *aws-observability-namespace*`.yaml` auf Ihrem Computer. Der Wert für `name` muss `aws-observability` sein und das Label `aws-observability: enabled` ist erforderlich.

   ```
   kind: Namespace
   apiVersion: v1
   metadata:
     name: aws-observability
     labels:
       aws-observability: enabled
   ```

   b. Erstellen Sie den Namespace.

   ```
   kubectl apply -f aws-observability-namespace.yaml
   ```

2. Erstellen Sie ein `ConfigMap` mit einem `Fluent Conf`-Datenwert, um Container-Logs an ein Ziel zu versenden. Fluent Conf ist Fluent Bit, eine schnelle und leichte Konfigurationssprache für den Protokollprozessor, die verwendet wird, um Containerprotokolle an ein Protokollziel Ihrer Wahl weiterzuleiten. Weitere Informationen finden Sie unter Configuration File (Konfigurationsdatei) in der Fluent Bit-Dokumentation.

> **Important**
>
> In einer typischen `Fluent Conf` sind die Hauptabschnitte `Service`, `Input`, `Filter` und `Output` enthalten. Der Fargate-Protokoll-Router akzeptiert jedoch nur:
>
> - Die Abschnitte `Filter` und `Output`.
> - Ein `Parser`-Abschnitt.
>
> Wenn Sie andere Abschnitte angeben, werden diese abgelehnt.

Der Fargate-Protokollrouter verwaltet die Abschnitte `Service` und `Input`. Es verfügt über den folgenden Abschnitt `Input`, der nicht geändert werden kann und in Ihrem `ConfigMap` nicht benötigt wird. Sie können daraus jedoch Erkenntnisse gewinnen, z. B. die Speicherpuffergrenze und das für Protokolle angewendete Tag.

```
[INPUT]
    Name tail
    Buffer_Max_Size 66KB
    DB /var/log/flb_kube.db
    Mem_Buf_Limit 45MB
    Path /var/log/containers/*.log
    Read_From_Head On
    Refresh_Interval 10
    Rotate_Wait 30
    Skip_Long_Lines On
    Tag kube.*
```

Berücksichtigen Sie beim Erstellen des `ConfigMap`, die folgenden Regeln, die Fargate verwendet, um Felder zu validieren:

- `[FILTER]`, `[OUTPUT]`, und `[PARSER]` sollen unter jedem entsprechenden Schlüssel angegeben werden. `[FILTER]` muss beispielsweise unter `filters.conf` liegen. Sie können ein oder mehrere `[FILTER]` in der `filters.conf` haben. Die Abschnitte `[OUTPUT]` und `[PARSER]` sollten sich auch unter den entsprechenden Schlüssel befinden. Durch die Angabe mehrerer `[OUTPUT]`-Abschnitte können Sie Ihre Protokolle gleichzeitig an verschiedene Ziele weiterleiten.

- Fargate validiert die erforderlichen Schlüssel für jeden Abschnitt. `Name` und `matchsind` für jedes `[FILTER]` und `[OUTPUT]` erforderlich. `Name` und `format` werden jeweils für jeden `[PARSER]` benötigt. Bei den Schlüssel wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- Umgebungsvariablen wie `${ENV_VAR}` sind im `ConfigMap` nicht erlaubt.
- Die Einrückung muss für die Direktive oder das Schlüssel-Wert-Paar innerhalb von `filters.conf`, `output.conf` und `parsers.conf` gleich sein. Schlüssel-Wert-Paare müssen stärker eingerückt werden als Direktiven.
- Fargate validiert gegen die folgenden unterstützten Filter: `grep`, `parser`, `record_modifier`, `rewrite_tag`, `throttle`, `nest`, `modify`, und `kubernetes`.
- Fargate validiert mit der folgenden unterstützten Ausgabe: `es`, `firehose`, `kinesis_firehose`, `cloudwatch`, `cloudwatch_logs`, und `kinesis`.
- Mindestens ein unterstütztes `Output`-Plugin muss im `ConfigMap` bereitgestellt werden, um die Protokollierung zu ermöglichen. `Filter` und `Parser` sind nicht erforderlich, um die Protokollierung zu aktivieren.

Sie können Fluent Bit auch auf Amazon EC2 mit der gewünschten Konfiguration ausführen, um alle Probleme zu beheben, die sich aus der Validierung ergeben. Erstellen Sie Ihr `ConfigMap` mit einem der folgenden Beispiele.

> **Important**
>
> Die Amazon-EKS-Fargate-Protokollierung unterstützt keine dynamische Konfiguration von `ConfigMaps`. Alle Änderungen an `ConfigMaps` werden nur auf neue Pods angewendet. Änderungen werden nicht auf vorhandene Pods angewendet.

Erstellen Sie anhand des Beispiels ein `ConfigMap` für Ihr gewünschtes Protokollziel.

> **Note**
>
> Sie können auch Amazon Kinesis Data Streams als Protokollziel verwenden. Stellen Sie bei der Nutzung von Kinesis Data Streams sicher, dass der Pod-Ausführungsrolle die Berechtigung `kinesis:PutRecords` erteilt wurde. Weitere Informationen finden Sie

unter „Berechtigungen für Amazon Kinesis Data Streams“ im offiziellen Handbuch von Fluent Bit.

CloudWatch

So erstellen Sie eine **`ConfigMap`** für CloudWatch

Bei Verwendung von haben Sie zwei Ausgabeoptionen CloudWatch:

- Ein Ausgabe-Plug-In in C geschrieben
- Ein in Golang geschriebenes Ausgabe-Plug-In

Das folgende Beispiel zeigt Ihnen, wie Sie das `cloudwatch_logs`-Plugin verwenden, um Protokolle an zu senden CloudWatch.

1. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *`aws-logging-cloudwatch-configmap`*`.yaml`. Ersetzen Sie durch *`region-code`* die AWS-Region , in der sich Ihr Cluster befindet. Die Parameter unter `[OUTPUT]` sind erforderlich.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
data:
  flb_log_cw: "false"  # Set to true to ship Fluent Bit process logs to
 CloudWatch.
  filters.conf: |
    [FILTER]
        Name parser
        Match *
        Key_name log
        Parser crio
    [FILTER]
        Name kubernetes
        Match kube.*
        Merge_Log On
        Keep_Log Off
        Buffer_Size 0
        Kube_Meta_Cache_TTL 300s
```

```
  output.conf: |
    [OUTPUT]
        Name cloudwatch_logs
        Match   kube.*
        region region-code
        log_group_name my-logs
        log_stream_prefix from-fluent-bit-
        log_retention_days 60
        auto_create_group true
  parsers.conf: |
    [PARSER]
        Name crio
        Format Regex
        Regex ^(?<time>[^ ]+) (?<stream>stdout|stderr) (?<logtag>P|F) (?
<log>.*)$
        Time_Key    time
        Time_Format %Y-%m-%dT%H:%M:%S.%L%z
```

2. Wenden Sie das Manifest auf Ihren Cluster an.

```
kubectl apply -f aws-logging-cloudwatch-configmap.yaml
```

3. Laden Sie die CloudWatch IAM-Richtlinie auf Ihren Computer herunter. Sie können die Richtlinie auch auf anzeigen GitHub.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-eks-fluent-
logging-examples/mainline/examples/fargate/cloudwatchlogs/permissions.json
```

Amazon OpenSearch Service

So erstellen Sie einen **ConfigMap** für Amazon OpenSearch Service

Wenn Sie Protokolle an Amazon OpenSearch Service senden möchten, können Sie es output verwenden, ein Plugin, das in geschrieben istC. Das folgende Beispiel zeigt Ihnen, wie Sie das -Plugin verwenden, um Protokolle an zu senden OpenSearch.

1. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *aws-logging-opensearch-configmap*.yaml. Ersetzen Sie jede *example value* durch Ihre eigenen Werte.

```
kind: ConfigMap
```

```
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
data:
  output.conf: |
    [OUTPUT]
      Name  es
      Match *
      Host  search-example-gjxdcilagiprbglqn42jsty66y.region-
code.es.amazonaws.com
      Port  443
      Index example
      Type  example_type
      AWS_Auth On
      AWS_Region region-code
      tls   On
```

2. Wenden Sie das Manifest auf Ihren Cluster an.

```
kubectl apply -f aws-logging-opensearch-configmap.yaml
```

3. Laden Sie die OpenSearch IAM-Richtlinie auf Ihren Computer herunter. Sie können die Richtlinie auch auf anzeigen GitHub.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-eks-
fluent-logging-examples/mainline/examples/fargate/amazon-elasticsearch/
permissions.json
```

   Stellen Sie sicher, dass die Zugriffskontrolle von OpenSearch Dashboards ordnungsgemäß konfiguriert ist. Dem `all_access role` in OpenSearch Dashboards müssen die Fargate-PodAusführungsrolle und die IAM-Rolle zugeordnet sein. Das gleiche Mapping muss für die `security_manager`-Rolle durchgeführt werden. Sie können die vorherigen Zuordnungen hinzufügen, indem Sie `Menu`, dann `Security`, dann `Roles` und dann die entsprechenden Rollen auswählen. Weitere Informationen finden Sie unter Wie behebe ich Probleme mit CloudWatch Protokollen, damit sie zu meiner Amazon-ES-Domain gestreamt werden?.

Firehose

So erstellen Sie eine **`ConfigMap`** für Firehose

Beim Senden von Protokollen an Firehose haben Sie zwei Ausgabeoptionen:

- `kinesis_firehose` – Ein in C geschriebenes Ausgabe-Plugin.
- `firehose` – Ein in Golang geschriebenes Ausgabe-Plug-In.

Das folgende Beispiel zeigt Ihnen, wie Sie das -`kinesis_firehose`Plugin verwenden, um Protokolle an Firehose zu senden.

1. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *`aws-logging-firehose-configmap`*`.yaml`. Ersetzen Sie durch *`region-code`* die AWS-Region , in der sich Ihr Cluster befindet.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
data:
  output.conf: |
    [OUTPUT]
     Name  kinesis_firehose
     Match *
     region region-code
     delivery_stream my-stream-firehose
```

2. Wenden Sie das Manifest auf Ihren Cluster an.

```
kubectl apply -f aws-logging-firehose-configmap.yaml
```

3. Laden Sie die Firehose-IAM-Richtlinie auf Ihren Computer herunter. Sie können die Richtlinie auch auf anzeigen GitHub.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-eks-fluent-logging-examples/mainline/examples/fargate/kinesis-firehose/permissions.json
```

3. Erstellen Sie eine IAM-Richtlinie aus der Richtliniendatei, die Sie in einem vorherigen Schritt heruntergeladen haben.

```
aws iam create-policy --policy-name eks-fargate-logging-policy --policy-document
 file://permissions.json
```

4. Hängen Sie die IAM-Richtlinie an die Pod-Ausführungsrolle an, die für Ihr Fargate-Profil angegeben ist, mit dem folgenden Befehl. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Ersetzen Sie *AmazonEKSFargatePodExecutionRole* durch Ihre Pod-Ausführungsrolle (weitere Informationen finden Sie unter Erstellen einer Fargate-Pod-Ausführungsrolle).

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::111122223333:policy/eks-fargate-logging-policy \
  --role-name AmazonEKSFargatePodExecutionRole
```

Kubernetes-Filter-Support

Dieses Feature erfordert die folgende minimale Kubernetes-Version und Plattformebene oder höher.

Kubernetes-Version	Plattformebene
1.23 und höher	eks.1

Der Fluent Bit-Kubernetes-Filter ermöglicht es Ihnen, Kubernetes-Metadaten zu Ihren Protokolldateien hinzuzufügen. Weitere Informationen über den Filter finden Sie unter Kubernetes in der Fluent Bit-Dokumentation. Sie können einen Filter unter Verwendung des Endpunkts des API-Servers anwenden.

```
filters.conf: |
    [FILTER]
        Name                kubernetes
        Match               kube.*
        Merge_Log           On
        Buffer_Size         0
        Kube_Meta_Cache_TTL 300s
```

> **Important**
>
> - `Kube_URL`,`Kube_CA_File`,`Kube_Token_Command`, und `Kube_Token_File` sind Konfigurationsparameter im Servicebesitz und dürfen nicht angegeben werden. Amazon-EKS-Fargate füllt diese Werte aus.
> - `Kube_Meta_Cache_TTL` ist die Zeit, in der Fluent Bit wartet, bis es mit dem API-Server für die neuesten Metadaten kommuniziert. Wenn `Kube_Meta_Cache_TTL` nicht angegeben wird, dann hängt Amazon-EKS-Fargate einen Standardwert von 30 Minuten an, um die Belastung des API-Servers zu verringern.

Fluent Bit-Prozessprotokolle an Ihr Konto senden

Sie können optional Fluent Bit Prozessprotokolle CloudWatch mit den folgenden an Amazon sendenConfigMap. Für den Versand von Fluent-Bit-Prozessprotokollen an CloudWatch sind zusätzliche Kosten für die Protokollaufnahme und -speicherung erforderlich. Ersetzen Sie durch *region-code* die AWS-Region , in der sich Ihr Cluster befindet.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
  labels:
data:
  # Configuration files: server, input, filters and output
  # ======================================================
  flb_log_cw: "true"  # Ships Fluent Bit process logs to CloudWatch.

  output.conf: |
    [OUTPUT]
        Name cloudwatch
        Match kube.*
        region region-code
        log_group_name fluent-bit-cloudwatch
        log_stream_prefix from-fluent-bit-
        auto_create_group true
```

Die Protokolle befinden sich in der AWS-Region , in der sich der Cluster befindet, unter CloudWatch. Der Name der Protokollgruppe lautet *my-cluster*`-fluent-bit-logs` und der Name des Fluent Bit-Logstreams lautet `fluent-bit-`*podname*`-`*pod-namespace*.

> **Note**
>
> - Die Prozessprotokolle werden nur ausgeliefert, wenn der Fluent Bit-Prozess erfolgreich gestartet wird. Wenn beim Starten von Fluent Bit ein Fehler auftritt, werden die Prozessprotokolle verpasst. Sie können nur Prozessprotokolle an senden CloudWatch.
> - Um Fehler beim Senden von Prozessprotokollen an Ihr Konto zu beheben, können Sie die vorherige `ConfigMap` verwenden, um die Prozessprotokolle abzurufen. Dass Fluent Bit nicht gestartet werden kann, liegt normalerweise daran, dass Ihre `ConfigMap` beim Starten nicht von Fluent Bit analysiert oder akzeptiert wird.

Den Versand von Fluent Bit-Prozessprotokollen anhalten

Für den Versand von Fluent Bit Prozessprotokollen an CloudWatch sind zusätzliche Kosten für die Protokollaufnahme und -speicherung erforderlich. Gehen Sie wie folgt vor, um Prozessprotokolle in einem vorhandenen `ConfigMap`-Setup auszuschließen.

1. Suchen Sie die CloudWatch Protokollgruppe, die automatisch für die Fluent Bit Prozessprotokolle Ihres Amazon-EKS-Clusters erstellt wurde, nachdem Sie die Fargate-Protokollierung aktiviert haben. Es folgt dem Format `{cluster_name}-fluent-bit-logs`.
2. Löschen Sie die vorhandenen CloudWatch Protokollstreams, die für jedes Pod's Prozessprotokoll in der CloudWatch Protokollgruppe erstellt wurden.
3. Bearbeiten Sie die `ConfigMap` und stellen Sie `flb_log_cw: "false"` ein.
4. Starten Sie alle vorhandenen Pods im Cluster neu.

Testanwendung

1. Stellen Sie einen Beispiel-Pod bereit.
   a. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *sample-app*`.yaml` auf Ihrem Computer.

```
apiVersion: apps/v1
```

```
kind: Deployment
metadata:
  name: sample-app
  namespace: same-namespace-as-your-fargate-profile
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - name: http
              containerPort: 80
```

b. Wenden Sie das Manifest auf den Cluster an.

```
kubectl apply -f sample-app.yaml
```

2. Zeigen Sie die NGINX-Protokolle mit den Zielen an, die Sie im `ConfigMap`.

Größenüberlegungen

Wir empfehlen Ihnen, für den Protokoll-Router bis zu 50 MB Speicher einzuplanen. Wenn Sie erwarten, dass Ihre Anwendung Protokolle mit sehr hohem Durchsatz generiert, sollten Sie bis zu 100 MB einplanen.

Fehlerbehebung

Um zu überprüfen, ob das Protokollierungsfeature aus irgendeinem Grund aktiviert oder deaktiviert ist, z. B. ein ungültiges `ConfigMap`, und warum es ungültig ist, überprüfen Sie Ihre Pod-Ereignisse mit **`kubectl describe pod pod_name`**. Die Ausgabe kann Pod-Ereignisse enthalten, die klarstellen, ob die Protokollierung aktiviert ist oder nicht, wie die folgende Beispielausgabe.

```
[...]
```

```
Annotations:          CapacityProvisioned: 0.25vCPU 0.5GB
                      Logging: LoggingDisabled: LOGGING_CONFIGMAP_NOT_FOUND
                      kubernetes.io/psp: eks.privileged
[...]
Events:
  Type     Reason           Age        From
               Message
  ----     ------           ----       ----
               -------
  Warning  LoggingDisabled  <unknown>  fargate-scheduler
                Disabled logging because aws-logging configmap was not found. configmap
 "aws-logging" not found
```

Die Pod-Ereignisse sind kurzlebig mit einem Zeitraum, der von den Einstellungen abhängt. Sie können die Anmerkungen eines Pod's auch mit **`kubectl describe pod`** ***`pod-name`*** anzeigen. In der Pod-Anmerkung finden Sie Informationen darüber, ob das Protokollierungsfeature aktiviert oder deaktiviert ist, und den Grund.

Auswählen eines Amazon-EC2-Instance-Typs

Amazon EC2 bietet eine große Auswahl an Instance-Typen für Worker-Knoten. Jeder Instance-Typ bietet andere Merkmale in Bezug auf Datenverarbeitung, Arbeitsspeicher, Speicher und Netzwerkfunktionen. Jede Instance wird abhängig von diesen Eigenschaften auch in Instance-Familien eingeordnet. Eine Liste finden Sie unter Verfügbare Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch und Verfügbare Instance-Typen im Amazon EC2 EC2-Benutzerhandbuch. Amazon EKS veröffentlicht mehrere Varianten von Amazon-EC2-AMIs, um Support zu ermöglichen. Berücksichtigen Sie die folgenden Kriterien, um sicherzustellen, dass der ausgewählte Instance-Typ mit Amazon EKS kompatibel ist.

- Keine der Amazon-EKS-AMIs unterstützen derzeit die Familien `g5g` und `mac`.
- Arm und nicht beschleunigte Amazon-EKS-AMIs unterstützen die Familien `g3`, `g4`, `inf` und `p`.
- Beschleunigte Amazon-EKS-AMIs unterstützen die Familien `a`, `c`, `hpc`, `m` und `t`.
- Für ARM-basierte Instances unterstützt Amazon Linux 2023 (AL2023) nur Instance-Typen, die Prozessoren Graviton2 oder höher verwenden. AL2023 unterstützt keine Instances. `A1`

Berücksichtigen Sie bei der Auswahl zwischen Instance-Typen, die von Amazon EKS unterstützt werden, die folgenden Funktionen jedes Typs.

Anzahl der Instances in einer Knotengruppe

Im Allgemeinen sind weniger, größere Instances besser, besonders wenn Sie viele Daemonsets haben. Jede Instance erfordert API-Aufrufe an den API-Server. Je mehr Instances Sie haben, desto mehr Last auf dem API-Server.

Betriebssystem

Überprüfen Sie die unterstützten Instance-Typen für Linux, Windows und Bottlerocket. Bevor Sie Windows-Instances erstellen, lesen Sie Die Windows-Unterstützung für Ihre Amazon-EKS-Cluster aktivieren.

Hardware-Architektur

Benötigen Sie x86 oder Arm? Sie können sie nur Linux auf Arm bereitstellen. Bevor Sie Arm-Instances bereitstellen, lesen Sie Amazon-EKS-optimierte Arm-Amazon-Linux-AMIs. Benötigen Sie Instances, die auf dem Nitro System (Linux oder Windows) basieren oder über beschleunigte Funktionen verfügen? Wenn Sie beschleunigte Funktionen benötigen, können Sie Linux nur mit Amazon EKS verwenden.

Maximale Anzahl von Pods

Da jedem Pod eine eigene IP-Adresse zugewiesen wird, ist die Anzahl der IP-Adressen für einen Instance-Typ ein Faktor bei der Bestimmung der Anzahl der Pods, die auf der Instance ausgeführt werden können. Um manuell zu bestimmen, wie viele Pods ein Instance-Typ unterstützt, siehe Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ.

> **Note**
>
> Wenn Sie ein Amazon-EKS-optimiertes Amazon-Linux-2-AMI der Version `v20220406` oder neuer verwenden, können Sie einen neuen Instance-Typ verwenden, ohne auf das neueste AMI zu aktualisieren. Für diese AMIs berechnet das AMI automatisch den nötigen `max-pods`-Wert, wenn er nicht in der Datei `eni-max-pods.txt` aufgeführt ist. Instance-Typen, die sich derzeit in der Vorschau befinden, werden möglicherweise von Amazon EKS standardmäßig nicht unterstützt. Werte für `max-pods` für solche Typen müssen noch zu `eni-max-pods.txt` in unserem AMI hinzugefügt werden.

AWS Nitro System-Instanztypen unterstützen optional deutlich mehr IP-Adressen als Instanztypen, die nicht zu Nitro System gehören. Allerdings stehen Pods nicht alle IP-Adressen zur Verfügung, die einer Instance zugewiesen wurden. Um Ihren Instances eine deutlich größere

Anzahl von IP-Adressen zuzuweisen, müssen Sie Version `1.9.0` oder höher des Amazon VPC CNI-Add-ons in Ihrem Cluster installiert und entsprechend konfiguriert haben. Weitere Informationen finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten. Um Ihren Instances die größte Anzahl von IP-Adressen zuzuweisen, müssen Sie Version `1.10.1` oder höher des Amazon VPC CNI-Add-ons in Ihrem Cluster installiert haben und den Cluster mit der `IPv6`-Familie bereitstellen.

IP-Familie

Sie können jeden unterstützten Instance-Typ verwenden, wenn Sie die `IPv4`-Familie für einen Cluster nutzen, was es Ihrem Cluster ermöglicht, Ihren Pods und Services private `IPv4`-Adressen zuzuweisen. Wenn Sie jedoch die `IPv6`-Familie für Ihren Cluster verwenden möchten, müsste Sie die AWS -Nitro-System-Instance-Typen oder Bare-Metal-Instance-Typen verwenden. Nur `IPv4` wird für Windows-Instances unterstützt. Ihr Cluster muss auf dem Version `1.10.1` oder höher des Amazon-VPC-CNI-Add-ons ausgeführt wird. Weitere Informationen zur Verwendung von `IPv6` finden Sie unter `IPv6`Adressen für ClusterPods, und services.

Version des Amazon-VPC-CNI-Add-ons, das Sie ausführen

Die aktuelle Version des Amazon-CNI-Plugins für Kubernetes unterstützt diese Instance-Typen. Möglicherweise müssen Sie Ihre Amazon VPC-CNI-Add-on-Version aktualisieren, um die Vorteile der neuesten unterstützten Instance-Typen zu nutzen. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on. Die neueste Version unterstützt die neuesten Features für die Verwendung mit Amazon EKS. Frühere Versionen unterstützen nicht alle Features. Sie können die von verschiedenen Versionen unterstützten Features im Änderungsverlauf auf GitHub anzeigen.

AWS-Region in dem Sie Ihre Knoten erstellen

Nicht alle Instance-Typen stehen in allen AWS-Regionen zur Verfügung.

Ob Sie Sicherheitsgruppen für Pods verwenden

Wenn Sie Sicherheitsgruppen für Pods verwenden, werden nur bestimmte Instance-Typen unterstützt. Weitere Informationen finden Sie unter Sicherheitsgruppen für Pods.

Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ

Da jedem Pod eine eigene IP-Adresse zugewiesen wird, ist die Anzahl der IP-Adressen für einen Instance-Typ ein Faktor bei der Bestimmung der Anzahl der Pods, die auf der Instance

ausgeführt werden können. Amazon EKS stellt ein Skript bereit, das Sie herunterladen und ausführen können, um die von Amazon EKS empfohlene maximale Anzahl von Pods zur Ausführung auf jedem Instance-Typ zu bestimmen. Das Skript verwendet Hardwareattribute jeder Instance und Konfigurationsoptionen, um die maximale Pods-Anzahl zu bestimmen. Sie können die in diesen Schritten zurückgegebene Zahl verwenden, um Funktionen wie das Zuweisen von IP-Adressen zu Pods aus einem anderen Subnetz als dem der Instance zu aktivieren und die Anzahl der IP-Adressen für Ihre Instance erheblich zu erhöhen. Wenn Sie eine verwaltete Knotengruppe mit mehreren Instance-Typen verwenden, verwenden Sie einen Wert, der für alle Instance-Typen funktionieren würde.

1. Laden Sie ein Skript herunter, mit dem Sie die maximale Anzahl von Pods für jeden Instance-Typ berechnen können.

   ```
   curl -O https://raw.githubusercontent.com/awslabs/amazon-eks-ami/master/templates/al2/runtime/max-pods-calculator.sh
   ```

2. Markieren Sie das Skript auf Ihrem Computer als ausführbar.

   ```
   chmod +x max-pods-calculator.sh
   ```

3. Führen Sie das Skript aus und ersetzen Sie *`m5.large`* durch den Instance-Typ, den Sie bereitstellen möchten, und *`1.9.0-eksbuild.1`* durch Ihre Amazon-VPC-CNI-Add-on-Version. Informationen zum Ermitteln der Add-on-Version finden Sie in den Update-Verfahren unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

   ```
   ./max-pods-calculator.sh --instance-type m5.large --cni-version 1.9.0-eksbuild.1
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   29
   ```

   Sie können dem Skript die folgenden Optionen hinzufügen, um die maximal unterstützten Pods anzuzeigen, wenn optionale Funktionen verwendet werden.

   - `--cni-custom-networking-enabled` – Verwenden Sie diese Option, wenn Sie IP-Adressen aus einem anderen Subnetz als der Ihrer Instance zuweisen möchten. Weitere Informationen finden Sie unter Benutzerdefinierte Netzwerke für Pods. Das Hinzufügen dieser Option zum vorherigen Skript mit denselben Beispielwerten ergibt `20`.

- `--cni-prefix-delegation-enabled` – Verwenden Sie diese Option, wenn Sie jeder Elastic-Network-Schnittstelle deutlich mehr IP-Adressen zuweisen möchten. Diese Funktion erfordert eine Amazon Linux-Instance, die auf dem Nitro System und der Version `1.9.0` oder höher des Amazon VPC CNI-Add-ons ausgeführt wird. Weitere Informationen finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten. Das Hinzufügen dieser Option zum vorherigen Skript mit denselben Beispielwerten ergibt `110`.

Sie können das Skript auch mit der Option `--help` ausführen, um alle verfügbaren Optionen anzuzeigen.

> **Note**
>
> Das Berechnungsskript für maximale Pods begrenzt den Rückgabewert auf Grundlage von Kubernetes-Skalierbarkeitsschwellenwerten und empfohlenen Einstellungen auf `110`. Wenn Ihr Instance-Typ mehr als 30 vCPUs hat, steigt dieser Grenzwert auf `250`, eine Zahl, die auf internen Tests des Amazon-EKS-Skalierbarkeitsteams basiert. Weitere Informationen finden Sie im Blogbeitrag Amazon VPC CNI plugin increases pods per node limits (VPC-CNI-Plugin von Amazon erhöht Grenzwerte für Pods pro Knoten).

Amazon EKS-optimierte AMIs

Sie können Knoten mit vordefinierten Amazon-EKS-optimierten Amazon Machine Images (AMIs) oder Ihren eigenen benutzerdefinierten AMIs bereitstellen. Informationen über die einzelnen Typen von Amazon-EKS-optimierten AMIs finden Sie in den folgenden Themen. Anweisungen zum Erstellen eines eigenen benutzerdefinierten AMI finden Sie unter Amazon-EKS-optimiertes Amazon Linux-AMI-Build-Skript.

Themen

- Amazon EKS hat die Unterstützung für Dockershim eingestellt
- Amazon EKS-optimierte Amazon Linux-AMIs
- Amazon-EKS-optimierte Bottlerocket-AMIs
- Für Amazon EKS optimierte Ubuntu Linux-AMIs
- Amazon-EKS-optimierte Windows-AMIs

Amazon EKS hat die Unterstützung für **`Dockershim`** eingestellt

Kubernetes unterstützt `Dockershim` nicht mehr. Das Kubernetes-Team hat die Laufzeit in der Kubernetes-Version `1.24` entfernt. Weitere Informationen finden Sie unter Kubernetes is Moving on From Dockershim: Commitments and Next Steps im Kubernetes-Blog.

Amazon EKS beendet auch den Support für `Dockershim` ab der Kubernetes-Version der `1.24`-Freigabe. Offiziell veröffentlichte Amazon EKS AMIs verfügen über `containerd` als einzige Laufzeit, beginnend mit Version `1.24`. In diesem Thema werden einige Details behandelt, aber weitere Informationen finden Sie unter Alles, was Sie über die Umstellung auf containerd in Amazon EKS wissen müssen.

Es gibt ein `kubectl`-Plugin, mit dem Sie anzeigen können, welche Ihrer Kubernetes-Workloads das Docker-Socket-Volume mounten. Weitere Informationen finden Sie unter Detector for Docker Socket (DDS) auf GitHub. Amazon-EKS-AMIs, auf denen Kubernetes-Versionen ausgeführt werden, die älter sind als `1.24` verwenden Docker als Standard-Laufzeit. Diese Amazon-EKS-AMIs verfügen jedoch über eine Bootstrap-Flag-Option, mit der Sie Ihre Workloads auf jedem unterstützten Cluster mit `containerd` verwenden können. Weitere Informationen finden Sie unter Testen Sie die Migration von Docker nach `containerd`.

Wir werden bis zum Ende des Support-Datums weiterhin AMIs für bestehende Kubernetes-Versionen veröffentlichen. Weitere Informationen finden Sie unter Amazon-EKS-Kubernetes-Release-Kalender. Wenn Sie mehr Zeit benötigen, um Ihre Workloads auf `containerd` zu testen, verwenden Sie eine unterstützte Version vor `1.24`. Wenn Sie jedoch offizielle Amazon-EKS-AMIs auf Version `1.24` oder höher aktualisieren möchten, stellen Sie sicher, dass Ihre Workloads auf `containerd` ausgeführt werden.

Die `containerd`-Laufzeit bietet eine zuverlässigere Leistung und Sicherheit. `containerd` ist die Laufzeit, auf die Amazon EKS standardisiert wird. Fargate und Bottlerocket verwenden bereits nur noch `containerd`. `containerd` trägt dazu bei, die Anzahl der Amazon EKS AMI-Versionen zu minimieren, die zur Behebung von `Dockershim` Common Vulnerabilities and Exposures (CVEs) erforderlich sind. Da `Dockershim` bereits `containerd` intern verwendet, müssen Sie möglicherweise keine Änderungen vornehmen. Es gibt jedoch einige Situationen, in denen Änderungen erforderlich sein können:

- Sie müssen Änderungen an Anwendungen vornehmen, die den Docker-Socket mounten. Beispielsweise sind Container-Images, die mit einem Container entwickelt werden, davon betroffen. Viele Überwachungstools montieren auch den Docker-Socket. Möglicherweise müssen Sie auf Updates warten oder Workloads für die Laufzeitüberwachung erneut bereitstellen.

- Möglicherweise müssen Sie Änderungen für Anwendungen vornehmen, die auf bestimmte Docker-Einstellungen angewiesen ist. Beispielsweise wird das HTTPS_PROXY-Protokoll nicht mehr unterstützt. Sie müssen die Anwendungen, die dieses Protokoll verwenden, aktualisieren. Weitere Informationen finden Sie unter `dockerd` und in Docker Docs.
- Wenn Sie die Amazon ECR-Anmeldeinformation zum Abrufen von Images verwenden, müssen Sie zum Anbieter von `kubelet`-Image-Anmeldeinformationen wechseln. Weitere Informationen finden Sie unter Konfigurieren eines Anbieters von `kubelet`-Image-Anmeldeinformationen in der Kubernetes-Dokumentation.
- Da Amazon EKS `1.24` Dockernicht mehr unterstützt, werden einige Flags, die zuvor vom Amazon EKS-Bootstrap-Skript unterstützt wurden, nicht mehr unterstützt. Bevor Sie zu Amazon EKS `1.24` oder höher wechseln, müssen Sie alle Verweise auf Flags entfernen, die jetzt nicht mehr unterstützt werden:
  - `--container-runtime dockerd` (`containerd` ist der einzige unterstützte Wert)
  - `--enable-docker-bridge`
  - `--docker-config-json`
- Wenn Fluentd bereits für Container Insights konfiguriert ist, müssen Sie Fluentd zu Fluent Bit migrieren, bevor Sie zu `containerd` wechseln. Die Fluentd-Parser sind so konfiguriert, dass sie nur Protokollnachrichten im JSON-Format analysieren. Im Gegensatz zu `dockerd` enthält die Container-Laufzeit `containerd` Protokollnachrichten, die sich nicht im JSON-Format befinden. Wenn Sie nicht zu Fluent Bit migrieren, erzeugen einige der konfigurierten Fluentd's-Parser eine große Anzahl von Fehlern im Container Fluentd. Weitere Informationen zur Migration finden Sie unter Einrichten Fluent Bit als DaemonSet zum Senden von Protokollen an CloudWatch Logs.
- Wenn Sie ein benutzerdefiniertes AMI verwenden und ein Upgrade auf Amazon EKS `1.24` durchführen, müssen Sie sicherstellen, dass die IP-Weiterleitung für Ihre Worker-Knoten aktiviert ist. Diese Einstellung wurde bei Docker nicht benötigt, ist aber für `containerd` erforderlich. Sie ist erforderlich, um Probleme mit den Netzwerkkonnektivitäten Pod-zu-Pod, Pod-zu-extern und Pod-zu-apiserver zu lösen.

  Führen Sie einen der folgenden Befehle aus, um diese Einstellung auf einem Worker-Knoten zu überprüfen:

  - `sysctl net.ipv4.ip_forward`
  - `cat /proc/sys/net/ipv4/ip_forward`

  Wenn die Ausgabe `0` lautet, führen Sie einen der folgenden Befehle aus, um die Kernelvariable `net.ipv4.ip_forward` zu aktivieren:

- `sysctl -w net.ipv4.ip_forward=1`
- `echo 1 > /proc/sys/net/ipv4/ip_forward`

Informationen zur Aktivierung der Einstellung auf Amazon-EKS-AMIs in der `containerd`-Laufzeit finden Sie unter `install-worker.sh` auf GitHub.

Amazon EKS-optimierte Amazon Linux-AMIs

Das für Amazon EKS optimierte Amazon Linux AMI basiert auf Amazon Linux 2 (AL2) und Amazon Linux 2023 (AL2023). Es ist so konfiguriert, dass es als Basisimage für Amazon-EKS-Knoten dient. Das AMI ist für die Nutzung mit Amazon EKS konfiguriert und enthält die folgenden Komponenten:

- `kubelet`
- AWS IAM-Authentifikator
- Docker (Amazon EKS-Version `1.23` und früher)
- `containerd`

> **Note**
>
> - Sie können Sicherheits- oder Datenschutzereignisse für AL2 im Amazon Linux-Sicherheitszentrum verfolgen oder den zugehörigen RSS-Feed abonnieren. Sicherheits- oder Datenschutzereignisse enthalten eine Übersicht über das Problem, welche Pakete betroffen sind und wie Sie Ihre Instances aktualisieren, um das Problem zu beheben.
> - Lesen Sie vor der Bereitstellung eines beschleunigten Arm-AMI die Informationen unter Amazon EKS-optimierte beschleunigte Amazon Linux-AMIs und Amazon-EKS-optimierte Arm-Amazon-Linux-AMIs.
> - Für die Kubernetes Version `1.23` können Sie ein optionales Bootstrap-Flag verwenden, um die Migration von Docker zu zu zu testen. `containerd` Weitere Informationen finden Sie unter Testen Sie die Migration von Docker nach `containerd`.
> - Ab Kubernetes-Version `1.25` können Sie Amazon-EC2-P2-Instances nicht mehr mit den Amazon-EKS-optimierten, beschleunigten Amazon-Linux-AMIs verwenden, die sofort einsatzbereit sind. Diese AMIs für Kubernetes-Versionen `1.25` oder höher werden Treiber der `NVIDIA 525`-Serie oder höher unterstützen, die mit den P2-Instances nicht kompatibel sind. Die Treiber der Serie `NVIDIA 525` oder höher sind jedoch mit den P3-, P4- und

P5-Instances kompatibel, sodass Sie diese Instances mit den AMIs für die Kubernetes-Version 1.25 oder höher verwenden können. Bevor Ihre Amazon-EKS-Cluster auf Version 1.25 aktualisiert werden, migrieren Sie alle P2-Instances zu P3-, P4- und P5-Instances. Sie sollten Ihre Anwendungen auch proaktiv aktualisieren, damit sie mit der NVIDIA 525-Serie oder höher funktionieren. Wir planen, die Treiber der neueren NVIDIA 525 Serie oder neuerer Kubernetes Versionen 1.23 bis Ende Januar 2024 zurück zu portieren. 1.24

- Alle neu erstellten verwalteten Knotengruppen in Clustern der Version 1.30 oder neuer verwenden standardmäßig AL2023 als Knotenbetriebssystem. Bisher wurde für neue Knotengruppen standardmäßig AL2 verwendet. Sie können AL2 weiterhin verwenden, indem Sie es beim Erstellen einer neuen Knotengruppe als AMI-Typ auswählen.
- Die Support für AL2 endet am 30. Juni 2025. Weitere Informationen finden Sie unter Amazon Linux 2 – Häufig gestellte Fragen.

Führen Sie ein Upgrade von AL2 auf AL2023 durch

Das für Amazon EKS optimierte AMI ist in zwei Familien erhältlich, die auf AL2 und AL2023 basieren. AL2023 ist ein neues Linux-basiertes Betriebssystem, das eine sichere, stabile und leistungsstarke Umgebung für Ihre Cloud-Anwendungen bietet. Es ist die nächste Generation von Amazon Linux von Amazon Web Services und ist in allen unterstützten Amazon EKS-Versionen verfügbar, einschließlich Versionen 1.23 und mit 1.24 erweitertem Support. Beschleunigte Amazon EKS-AMIs, die auf AL2023 basieren, werden zu einem späteren Zeitpunkt verfügbar sein. Wenn Sie beschleunigte Workloads haben, sollten Sie weiterhin das AL2-beschleunigte AMI oder Bottlerocket verwenden.

AL2023 bietet mehrere Verbesserungen gegenüber AL2. Einen vollständigen Vergleich finden Sie unter Vergleich von AL2 und Amazon Linux 2023 im Amazon Linux 2023-Benutzerhandbuch. Es wurden mehrere Pakete hinzugefügt, aktualisiert und aus AL2 entfernt. Es wird dringend empfohlen, Ihre Anwendungen vor dem Upgrade mit AL2023 zu testen. Eine Liste aller Paketänderungen in AL2023 finden Sie unter Paketänderungen in Amazon Linux 2023 in den Versionshinweisen zu Amazon Linux 2023.

Zusätzlich zu diesen Änderungen sollten Sie Folgendes beachten:

- AL2023 führt einen neuen Knoteninitialisierungsprozess einnodeadm, der ein YAML-Konfigurationsschema verwendet. Wenn Sie selbstverwaltete Knotengruppen oder ein AMI mit einer Startvorlage verwenden, müssen Sie jetzt beim Erstellen einer neuen Knotengruppe explizit zusätzliche Cluster-Metadaten angeben. Ein Beispiel für die mindestens erforderlichen

Parameter lautet wie folgt, wobei `apiServerEndpointcertificateAuthority`, und Service jetzt erforderlich `cidr` sind:

```
---
apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig
spec:
  cluster:
    name: my-cluster
    apiServerEndpoint: https://example.com
    certificateAuthority: Y2VydGlmaWNhdGVBdXRob3JpdHk=
    cidr: 10.100.0.0/16
```

In AL2 wurden die Metadaten dieser Parameter aus dem Amazon `DescribeCluster` EKS-API-Aufruf ermittelt. Mit AL2023 hat sich dieses Verhalten geändert, da durch den zusätzlichen API-Aufruf die Gefahr einer Drosselung bei der Skalierung großer Knoten besteht. Diese Änderung wirkt sich nicht auf Sie aus, wenn Sie verwaltete Knotengruppen ohne Startvorlage verwenden oder wenn Sie Karpenter Weitere Informationen zu `certificateAuthority` und Service `cidr` finden Sie `DescribeCluster` in der Amazon EKS API-Referenz.

- Dockerwird in AL2023 nicht für alle unterstützten Amazon EKS-Versionen unterstützt. Die Support für Docker wurde mit der Amazon EKS-Version `1.24` oder höher in AL2 beendet und entfernt. Weitere Informationen zu veralteten Versionen finden Sie unter Amazon EKS hat den Support für beendet. `Dockershim`
- Für AL2023 ist eine Amazon VPC CNI-Version `1.16.2` oder höher erforderlich.
- AL2023 erfordert standardmäßig. `IMDSv2` `IMDSv2`hat mehrere Vorteile, die zur Verbesserung der Sicherheitslage beitragen. Es verwendet eine sitzungsorientierte Authentifizierungsmethode, die die Erstellung eines geheimen Tokens in einer einfachen HTTP-PUT-Anfrage erfordert, um die Sitzung zu starten. Das Token einer Sitzung kann zwischen 1 Sekunde und 6 Stunden gültig sein. Weitere Informationen zum Übergang von `IMDSv1` zu `IMDSv2` finden Sie unter Umstellung auf Instance Metadata Service Version 2 und Nutzen Sie alle Vorteile von IMDSv2 und deaktivieren Sie IMDSv1 in Ihrer gesamten Infrastruktur. AWS Wenn Sie es verwenden möchten, können Sie dies dennoch tun`IMDSv1`, indem Sie die Einstellungen mithilfe der Starteigenschaften der Instanz-Metadatenoption manuell überschreiben.

> **Note**
>
> Für `IMDSv2` ist die Standard-Hop-Anzahl für verwaltete Knotengruppen auf 1 gesetzt. Das bedeutet, dass Container mithilfe von IMDS keinen Zugriff auf die

Anmeldeinformationen des Knotens haben. Wenn Sie Container-Zugriff auf die Anmeldeinformationen des Knotens benötigen, können Sie dies dennoch tun, indem Sie die `HttpPutResponseHopLimit` in einer benutzerdefinierten Amazon EC2 EC2-Startvorlage manuell überschreiben und auf 2 erhöhen. Alternativ können Sie stattdessen Amazon EKS Pod Identity verwenden, um Anmeldeinformationen bereitzustellen`IMDSv2`.

- AL2023 bietet die nächste Generation einheitlicher Kontrollgruppenhierarchien (`cgroupv2`). `cgroupv2`wird verwendet, um eine Container-Laufzeit zu implementieren, und von`systemd`. AL2023 enthält zwar immer noch Code, mit dem das System ausgeführt werden kann`cgroupv1`, dies ist jedoch keine empfohlene oder unterstützte Konfiguration. Diese Konfiguration wird in einer future Hauptversion von Amazon Linux vollständig entfernt.
- `eksctl`Für `eksctl` die Unterstützung von AL2023 ist eine Version `0.176.0` oder höher erforderlich.

Für bereits bestehende verwaltete Knotengruppen können Sie entweder ein direktes Upgrade oder ein blaues/grünes Upgrade durchführen, je nachdem, wie Sie eine Startvorlage verwenden:

- Wenn Sie ein benutzerdefiniertes AMI mit einer verwalteten Knotengruppe verwenden, können Sie ein direktes Upgrade durchführen, indem Sie die AMI-ID in der Startvorlage austauschen. Sie sollten sicherstellen, dass Ihre Anwendungen und alle Benutzerdaten zuerst auf AL2023 übertragen werden, bevor Sie diese Upgrade-Strategie durchführen.
- Wenn Sie verwaltete Knotengruppen entweder mit der Standardstartvorlage oder mit einer benutzerdefinierten Startvorlage verwenden, die die AMI-ID nicht angibt, müssen Sie das Upgrade mit einer Blau/Grün-Strategie durchführen. Ein Blau/Grün-Upgrade ist in der Regel komplexer und beinhaltet die Erstellung einer völlig neuen Knotengruppe, für die Sie AL2023 als AMI-Typ angeben würden. Die neue Knotengruppe muss anschließend sorgfältig konfiguriert werden, um sicherzustellen, dass alle benutzerdefinierten Daten aus der AL2-Knotengruppe mit dem neuen Betriebssystem kompatibel sind. Sobald die neue Knotengruppe mit Ihren Anwendungen getestet und validiert wurde, Pods kann sie von der alten Knotengruppe zur neuen Knotengruppe migriert werden. Sobald die Migration abgeschlossen ist, können Sie die alte Knotengruppe löschen.

Wenn Sie AL2023 verwenden Karpenter und verwenden möchten, müssen Sie das `EC2NodeClass` `amiFamily` Feld mit AL2023 ändern. Standardmäßig ist Drift in aktiviert. Karpenter Das bedeutet, dass Ihre Worker-Knoten, sobald das `amiFamily` Feld geändert wurde, Karpenter automatisch auf das neueste AMI aktualisiert werden, sofern verfügbar.

Amazon EKS-optimierte beschleunigte Amazon Linux-AMIs

> **Note**
>
> Beschleunigte Amazon EKS-AMIs, die auf AL2023 basieren, werden zu einem späteren Zeitpunkt verfügbar sein. Wenn Sie beschleunigte Workloads haben, sollten Sie weiterhin das AL2-beschleunigte AMI oder verwenden. Bottlerocket

Das für Amazon EKS optimierte beschleunigte Amazon-Linux-AMI basiert auf dem standardmäßigen Amazon-EKS-optimierten Amazon-Linux-AMI. Es ist so konfiguriert, dass es als optionales Image für Amazon EKS-Knoten zur Unterstützung von GPU-, Inferentia - und Trainium-basierten Workloads dient.

Zusätzlich zur Amazon EKS-optimierten AMI-Standardkonfiguration enthält das beschleunigte AMI Folgendes:

- NVIDIA-Treiber
- `nvidia-container-runtime`
- AWS NeuronTreiber

Eine Liste der neuesten Komponenten, die im beschleunigten AMI enthalten sind, finden Sie in den `amazon-eks-ami` Releases unterGitHub.

> **Note**
>
> - Das Amazon-EKS-optimierte beschleunigte AMI unterstützt nur GPU- und Inferentia-basierte Instance-Typen. Stellen Sie sicher, dass Sie diese Instance-Typen in Ihrer AWS CloudFormation Node-Vorlage angeben. Durch Verwendung des Amazon EKS-optimierten beschleunigten AMI stimmen Sie der Benutzer-Lizenzvereinbarung (EULA) von NVIDIA zu.
> - Das Amazon EKS-optimierte beschleunigte AMI wurde zuvor als Amazon EKS-optimiertes AMI mit GPU-Unterstützung bezeichnet.
> - In früheren Versionen des Amazon-EKS-optimierten beschleunigten AMI war das `nvidia-docker`-Repository installiert. Das Repository ist nicht mehr im Amazon EKS-AMI Version `v20200529` und höher enthalten.

Um Workloads zu aktivieren, AWS die auf Neuron (ML Accelerator) basieren

Einzelheiten zu Trainings- und Inferenz-Workloads, die Neuron in Amazon EKS verwendet werden, finden Sie in den folgenden Referenzen:

- Container — Kubernetes — Erste Schritte in der Dokumentation AWS Neuron
- Schulungen zu AWS Neuron EKS-Beispielen am GitHub
- Machine Learning-Inferenz mit AWS Inferentia

Um GPU-basierte Workloads zu aktivieren

Im folgenden Verfahren wird beschrieben, wie Sie auf einer GPU-basierten Instance mit dem Amazon EKS-optimierten beschleunigten AMI einen Workload ausführen.

1. Nachdem Ihre GPU-Knoten Ihrem Cluster beigetreten sind, müssen Sie das NVIDIA-Geräte-Plugin für Kubernetes als DaemonSet auf Ihrem Cluster festlegen. Ersetzen Sie *vX.X.X* mit Ihrer gewünschten Nvidia/K8S-Geräte-Plugin-Version, bevor Sie den folgenden Befehl ausführen.

   ```
   kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/
   nvidia-device-plugin.yml
   ```

2. Sie können mit dem folgenden Befehl überprüfen, ob Ihre Knoten über zuordnungsfähige GPUs verfügen.

   ```
   kubectl get nodes "-o=custom-
   columns=NAME:.metadata.name,GPU:.status.allocatable.nvidia\.com/gpu"
   ```

Um einen Pod bereitzustellen, um zu testen, ob Ihre GPU-Knoten ordnungsgemäß konfiguriert sind

1. Erstellen Sie eine Datei mit dem Namen `nvidia-smi.yaml` und dem folgenden Inhalt. Ersetzen Sie *tag* mit Ihrem gewünschten Tag für `nvidia/cuda`. Dieses Manifest startet einen NVIDIA CUDA-Container, der `nvidia-smi` auf einem Knoten ausführt.

   ```
   apiVersion: v1
   kind: Pod
   metadata:
     name: nvidia-smi
   spec:
   ```

```
  restartPolicy: OnFailure
  containers:
  - name: nvidia-smi
    image: nvidia/cuda:tag
    args:
    - "nvidia-smi"
    resources:
      limits:
        nvidia.com/gpu: 1
```

2. Wenden Sie das Manifest mit dem folgenden Befehl an.

```
kubectl apply -f nvidia-smi.yaml
```

3. Nachdem der Pod ausgeführt wurde, zeigen Sie mit dem folgenden Befehl seine Protokolle an.

```
kubectl logs nvidia-smi
```

Eine Beispielausgabe sieht wie folgt aus.

```
Mon Aug  6 20:23:31 20XX
+-----------------------------------------------------------------------------+
| NVIDIA-SMI XXX.XX                 Driver Version: XXX.XX                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:00:1C.0 Off |                    0 |
| N/A   46C    P0    47W / 300W |      0MiB / 16160MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Amazon-EKS-optimierte Arm-Amazon-Linux-AMIs

Arm-Instances führen zu deutlichen Kosteneinsparungen für skalierbare und Arm-basierte Anwendungen wie Webserver, Container-Microservices, Zwischenspeicherflotten und verteilte Datenspeicher. Beachten Sie beim Hinzufügen von Arm-Knoten zu Ihrem Cluster die folgenden Überlegungen.

Überlegungen

- Wenn Ihr Cluster vor dem 17. August 2020 bereitgestellt wurde, müssen Sie ein einmaliges Upgrade kritischer Cluster-Add-on-Manifeste durchführen. Das ist erforderlich, damit Kubernetes für jede Hardwarearchitektur, die in Ihrem Cluster verwendet wird, das richtige Image abrufen kann. Weitere Informationen zum Aktualisieren von Cluster-Add-ons finden Sie unter Die Kubernetes-Version für Ihren Amazon-EKS-Cluster aktualisieren. Wenn Sie Ihren Cluster am oder nach dem 17. August 2020 bereitgestellt haben, sind Ihre Add-ons für CoreDNS, `kube-proxy` und Amazon VPC CNI plugin for Kubernetes bereits Multi-Architektur-fähig.
- Anwendungen, die auf Arm-Knoten bereitgestellt werden, müssen für Arm kompiliert werden.
- Wenn Sie DaemonSets in einem vorhandenen Cluster bereitgestellt haben oder sie in einem neuen Cluster bereitstellen möchten, in dem Sie auch Arm-Knoten bereitstellen möchten, stellen Sie sicher, dass Ihr DaemonSet auf allen Hardware-Architekturen in Ihrem Cluster ausgeführt werden kann.
- Sie können Arm-Knotengruppen und x86-Knotengruppen im selben Cluster ausführen. In diesem Fall sollten Sie Container-Images mit mehreren Architekturen in einem Container-Repository wie Amazon Elastic Container Registry bereitstellen und dann Knotenselektoren zu Ihren Manifesten hinzufügen, damit Kubernetes weiß, auf welcher Hardwarearchitektur ein Pod bereitgestellt werden kann. Weitere Informationen finden Sie unter Übertragen eines Multi-Architektur-Images im Amazon-ECR-Benutzerhandbuch und im Blogbeitrag Einführung in Multi-Architektur-Container-Images für Amazon ECR.

Testen Sie die Migration von Docker nach **`containerd`**

Amazon EKS hat die Unterstützung für Docker mit der Einführung der Kubernetes-Version `1.24` eingestellt. Weitere Informationen finden Sie unter Amazon EKS hat die Unterstützung für `Dockershim` eingestellt.

Für Kubernetes Version können Sie ein optionales Bootstrap-Flag verwenden`1.23`, um die `containerd` Laufzeit für Amazon EKS-optimierte AL2-AMIs zu aktivieren. Mit diesem Feature

erhalten Sie einen klaren Pfad für die Migration zu `containerd`, wenn Sie auf Version `1.24` oder höher aktualisieren. Amazon EKS hat die Unterstützung für Docker mit der Einführung der Kubernetes-Version `1.24` eingestellt. Die `containerd`-Laufzeit ist in der Kubernetes-Community weit verbreitet und ist ein abgestuftes Projekt der CNCF. Sie können es testen, indem Sie zu einem neuen oder vorhandenen Cluster eine Knotengruppe hinzufügen.

Sie können das Boostrap-Flag aktivieren, indem Sie eine der folgenden Arten von Knotengruppen erstellen.

Selbstverwaltet

Erstellen Sie die Knotengruppe mithilfe der Anweisungen in Starten selbstverwalteter Amazon Linux-Knoten. Geben Sie ein Amazon-EKS-optimiertes AMI und den folgenden Text für den Parameter `BootstrapArguments` an.

```
--container-runtime containerd
```

Verwaltet

Wenn Sie `eksctl` verwenden, erstellen Sie eine Datei namens *`my-nodegroup`*`.yaml` mit folgendem Inhalt. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten. Informationen zum Abrufen einer optimierten AMI-ID für `ami-`*`1234567890abcdef0`* finden Sie unter Abrufen von Amazon-EKS-optimierten Amazon-Linux-AMI-IDs.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: my-cluster
  region: region-code
  version: 1.23
managedNodeGroups:
  - name: my-nodegroup
    ami: ami-1234567890abcdef0
    overrideBootstrapCommand: |
      #!/bin/bash
      /etc/eks/bootstrap.sh my-cluster --container-runtime containerd
```

> **Note**
>
> Wenn Sie viele Knoten gleichzeitig starten, ist es sinnvoll, ebenfalls Werte für die Bootstrap-Argumente `--apiserver-endpoint`, `--b64-cluster-ca` und `--dns-cluster-ip` anzugeben, um Fehler zu vermeiden. Weitere Informationen finden Sie unter Angeben eines AMI.

Führen Sie den folgenden Befehl aus, um die Benutzergruppe zu erstellen.

```
eksctl create nodegroup -f my-nodegroup.yaml
```

Wenn Sie die verwaltete Knotengruppe lieber mit einem anderen Werkzeug erstellen möchten, müssen Sie die Knotengruppe mithilfe einer Startvorlage bereitstellen. Geben Sie in Ihrer Startvorlage eine Amazon EKS-optimierte AMI-ID an, stellen Sie dann die Knotengruppe mithilfe einer Startvorlage bereit und geben Sie die folgenden Benutzerdaten an. Diese Benutzerdaten übergeben Argumente an die `bootstrap.sh`-Datei. Weitere Informationen zur Bootstrap-Datei finden Sie unter bootstrap.sh auf GitHub.

```
/etc/eks/bootstrap.sh my-cluster --container-runtime containerd
```

Weitere Informationen

Weitere Informationen zu Amazon-EKS-optimierten Amazon Linux AMIs finden Sie in den folgenden Abschnitten:

- Informationen zur Verwendung von Amazon Linux mit verwalteten Knotengruppen finden Sie unter Verwaltete Knotengruppen.
- Informationen zum Starten selbstverwalteter Amazon-Linux-Knoten finden Sie unter Abrufen von Amazon-EKS-optimierten Amazon-Linux-AMI-IDs.
- Versionsinformationen finden Sie unter Amazon-EKS-optimierte Amazon-Linux-AMI-Versionen.
- Informationen zum Abrufen der neuesten IDs der für Amazon EKS optimierten Amazon Linux AMIs finden Sie unter Abrufen von Amazon-EKS-optimierten Amazon-Linux-AMI-IDs.
- Informationen zu Open-Source-Skripten, die zum Erstellen des Amazon-EKS-optimierten AMI verwendet werden, finden Sie unter Amazon-EKS-optimiertes Amazon Linux-AMI-Build-Skript.

Amazon-EKS-optimierte Amazon-Linux-AMI-Versionen

Für Amazon EKS optimierte Amazon-Linux-AMIs werden anhand der Kubernetes-Version und des Veröffentlichungsdatums des AMI in folgendem Format versioniert:

```
k8s_major_version.k8s_minor_version.k8s_patch_version-release_date
```

Jede AMI-Version enthält verschiedene Versionen von `kubelet`, Docker, dem Linux Kernel und`containerd`. Das beschleunigte AMI umfasst auch verschiedene Versionen des NVIDIA-Treibers. Diese Informationen finden Sie im Changelog (Änderungsprotokoll) in GitHub.

Abrufen von Amazon-EKS-optimierten Amazon-Linux-AMI-IDs

Sie können die Amazon Machine Image (AMI) -ID für Amazon EKS-optimierte AMIs programmgesteuert abrufen, indem Sie die AWS Systems Manager Parameter Store-API abfragen. Mit diesem Parameter müssen Sie Amazon-EKS-optimierte AMI-IDs nicht manuell abrufen. Weitere Informationen zur Systems Manager Parameter Store-API finden Sie unter GetParameter.

Um eine AMI-ID für Amazon EKS-optimierte AMIs abzurufen, verwenden Sie den AWS CLI

1. Ermitteln Sie die Region, in der Ihre Node-Instance bereitgestellt werden soll, z. `us-west-2` B.
2. Ermitteln Sie den AMI-Typ, den Sie benötigen. Informationen zu den Typen von Amazon EC2 EC2-Instances finden Sie unter Instance-Typen.
   - `amazon-linux-2`ist für Amazon Linux 2 (AL2) `x86` -basierte Instances.
   - `amazon-linux-2-arm64`ist für AL2-ARM-Instances, wie z. B. AWS Graviton-basierte Instances.
   - `amazon-linux-2-gpu`ist für GPU-beschleunigte AL2-Instanzen.
   - `amazon-linux-2023/x86_64/standard`dist für auf Amazon Linux 2023 (AL2023) `x86` basierende Instances.
   - `amazon-linux-2023/arm64/standard`dist für AL2023 ARM-Instances.
3. Ermitteln Sie die Kubernetes Version des Clusters, an den Ihr Knoten angehängt werden soll, z. B. 1.30.
4. Führen Sie den folgenden AWS CLI Befehl aus, um die entsprechende AMI-ID abzurufen. Ersetzen Sie AWS-Region die Kubernetes Version und die Plattform nach Bedarf. Sie müssen AWS CLI mit einem IAM-Prinzipal angemeldet sein, der über die `ssm:GetParameter` IAM-Berechtigung zum Abrufen der für Amazon EKS optimierten AMI-Metadaten verfügt.

```
aws ssm get-parameter --name /aws/service/eks/optimized-ami/1.30/amazon-linux-2/
recommended/image_id \
                --region region-code --query "Parameter.Value" --output text
```

Eine Beispielausgabe sieht wie folgt aus.

```
ami-1234567890abcdef0
```

Amazon-EKS-optimiertes Amazon Linux-AMI-Build-Skript

Amazon Elastic Kubernetes Service (Amazon EKS) hat die Open-Source-Skripts, die zum Erstellen des Amazon-EKS-optimierten AMI verwendet werden, als Open-Source-Software zur Verfügung gestellt. Diese Build-Skripts sind ab sofort auf GitHub verfügbar.

Das für Amazon EKS optimierte Amazon Linux-AMI basiert auf Amazon Linux 2 (AL2) und Amazon Linux 2023 (AL2023) und ist speziell für die Verwendung als Knoten in Amazon EKS-Clustern konzipiert. Sie können dieses Repository verwenden, um zu erfahren, wie das Amazon EKS-Team den AWS IAM Authenticator konfiguriert `kubelet` und Ihr eigenes Amazon Linux-basiertes AMI von Grund Kubernetes auf neu erstellt. Docker

Das Build-Skript-Repository enthält eine HashiCorpPacker-Vorlage und Build-Skripte zur Generierung eines AMI. Diese Skripts sind die Informationsquelle für Amazon-EKS-optimierte AMI-Builds. Sie können also dem GitHub-Repository folgen, um Änderungen an unseren AMIs zu überwachen. Beispielsweise möchten Sie vielleicht, dass Ihr eigenes AMI dieselbe Version von Docker nutzt, die das Amazon-EKS-Team für das offizielle AMI verwendet.

Das GitHub Repository enthält auch das spezielle Bootstrap-Skript und das Nodeadm-Skript, die beim Booten ausgeführt werden, um die Zertifikatsdaten Ihrer Instance, den Endpunkt der Kontrollebene, den Clusternamen und mehr zu konfigurieren.

Darüber hinaus enthält das GitHub Repository unsere Amazon AWS CloudFormation EKS-Knotenvorlagen. Diese Vorlagen vereinfachen die Einrichtung einer Instance, die das Amazon-EKS-optimierte AMI ausführt, und ihre Registrierung bei einem Cluster.

Weitere Informationen finden Sie in den Repositorys auf GitHub unter https://github.com/awslabs/amazon-eks-ami.

Das für Amazon EKS optimierte AL2 enthält ein optionales Bootstrap-Flag zur Aktivierung der `containerd` Laufzeit.

Konfiguration VT1 für Ihr benutzerdefiniertes Amazon Linux AMI

Benutzerdefinierte Amazon Linux-AMIs in Amazon EKS können die VT1-Video-Transcoding-Instance-Familie für Amazon Linux 2 (AL2), Ubuntu 18 und 20 unterstützen. Ubuntu VT1unterstützt die Xilinx U30-Medientranscodierungskarten mit beschleunigten H.264/AVC- und H.265/HEVC-Codecs. Um die Vorteile dieser beschleunigten Instances zu nutzen, müssen Sie die folgenden Schritte ausführen:

1. Erstellen und starten Sie ein Basis-AMI von AL2, Ubuntu 18 oder Ubuntu 20.
2. Nachdem das basierte AMI gestartet wurde, installieren Sie den XRT-Treiber und die Laufzeit auf dem Knoten.
3. Erstellen eines Amazon-EKS-Clusters.
4. Installieren Sie die KubernetesFPGA-Erweiterung auf Ihrem Cluster.

   ```
   kubectl apply -f fpga-device-plugin.yml
   ```

Das Plugin kündigt jetzt Xilinx U30-Geräte pro Knoten in Ihrem Amazon EKS-Cluster an. Sie können das FFMPEG Docker-Image verwenden, um Beispiel-Workloads für die Videotranskodierung auf Ihrem Amazon EKS-Cluster auszuführen.

Konfiguration DL1 für Ihr benutzerdefiniertes Amazon Linux 2 AMI

Benutzerdefinierte Amazon Linux 2 (AL2) -AMIs in Amazon EKS können Deep-Learning-Workloads durch zusätzliche Konfigurationen und Kubernetes Add-Ons in großem Umfang unterstützen. In diesem Dokument werden die Komponenten beschrieben, die zum Einrichten einer generischen Kubernetes-Lösung für ein On-Premise-Setup oder als Baseline in einer größeren Cloud-Konfiguration erforderlich sind. Um diese Funktion zu unterstützen, müssen Sie in Ihrer benutzerdefinierten Umgebung die folgenden Schritte ausführen:

- SynapaseAI® SoftwareAuf das System geladene Treiber — Diese sind in den auf Github verfügbaren AMIs enthalten.
- Das Habana Geräte-Plugin — ADaemonSet, mit dem Sie die Registrierung von Habana Geräten in Ihrem Kubernetes Cluster automatisch aktivieren und den Zustand der Geräte verfolgen können.
- Helm 3.x

- Helm-Chart zur Installation von MPI-Operator.
- MPI-Operator

1. Erstellen und starten Sie ein Basis-AMI von AL2, Ubuntu 18 oder Ubuntu 20.
2. Folgen Sie diesen Anweisungen, um die Umgebung für DL1 einzurichten.

Amazon-EKS-optimierte Bottlerocket-AMIs

Bottlerocket ist eine Open-Source-Linux-Distribution, die gesponsert und unterstützt wird von AWS. Bottlerocket wurde speziell für das Hosten von Container-Workloads entwickelt. Mit Bottlerocket können Sie die Verfügbarkeit von containerisierten Bereitstellungen verbessern und die Betriebskosten senken, indem Sie Updates Ihrer Container-Infrastruktur automatisieren. Bottlerocket beinhaltet nur die für den Betrieb von Containern notwendige Software, wodurch die Ressourcennutzung verbessert, Sicherheitsbedrohungen reduziert und der Verwaltungsaufwand gesenkt wird. Das Bottlerocket-AMI umfasst `containerd`, `kubelet`, und AWS-IAM-Authenticator. Bottlerocket wird neben verwalteten Knotengruppen und selbstverwalteten Knoten auch unterstützt von Karpenter.

Vorteile

Die Verwendung von Bottlerocket mit Ihrem Amazon-EKS-Cluster hat die folgenden Vorteile:

- Höhere Verfügbarkeit bei geringeren Betriebskosten und geringerer Verwaltungskomplexität – Bottlerocket hat einen geringeren Ressourcenbedarf, kürzere Startzeiten und ist weniger anfällig für Sicherheitsbedrohungen als andere Linux-Distributionen. Der geringere Umfang von Bottlerocket's trägt zur Kostensenkung bei, da weniger Speicher-, Datenverarbeitungs- und Netzwerkressourcen verwendet werden.
- Verbesserte Sicherheit durch automatische Betriebssystemupdates – Updates für Bottlerocket werden als eine Einheit installiert, die bei Bedarf rückgängig gemacht werden kann. Dadurch wird das Risiko beschädigter oder fehlgeschlagener Updates vermieden, die das System in einen unbrauchbaren Zustand versetzen können. Mit Bottlerocket können Sicherheitsupdates automatisch angewendet werden, sobald sie verfügbar sind, und das mit minimaler Unterbrechung. Bei Ausfällen können sie zurückgesetzt werden.
- Premium-Support – Von AWS bereitgestellte Versionen von Bottlerocket auf Amazon EC2 sind im Rahmen derselben AWS Support-Pläne abgedeckt, die auch AWS-Services wie Amazon EC2, Amazon EKS und Amazon ECR abdecken.

Überlegungen

Bei der Verwendung von Bottlerocket für Ihren AMI-Typ sollten Sie Folgendes beachten:

- Bottlerocket unterstützt Amazon-EC2-Instances mit `x86_64`- und `arm64`-Prozessoren. Die Bottlerocket-AMI wird nicht für die Verwendung mit Amazon-EC2-Instances mit einem Inferentia-Chip empfohlen.
- Derzeit gibt es keine AWS-CloudFormation-Vorlage, mit der Sie Bottlerocket-Knoten bereitstellen können.
- Bottlerocket-Images enthalten keine SSH-Server oder Shell. Sie können Out-of-Band-Zugriffsmethoden verwenden, um SSH zuzulassen. Diese Ansätze ermöglichen es dem Administrator-Container, einige Bootstrapping-Konfigurationsschritte mit Benutzerdaten zu übergeben. Weitere Informationen finden Sie in den folgenden Abschnitten unter Bottlerocket OS in GitHub:
  - Exploration (Erkundung)
  - Administrator-Container
  - Kubernetes-Einstellungen
- Bottlerocket verwendet verschiedene Containertypen:
  - Standardmäßig ist ein Steuerungs-Container aktiviert. Dieser Container führt den AWS Systems Manager-Agenten aus, den Sie verwenden können, um Befehle auszuführen oder Shell-Sitzungen auf Amazon-EC2-Bottlerocket-Instances zu starten. Weitere Informationen finden Sie unter Session Manager einrichten im AWS Systems Manager-Benutzerhandbuch.
  - Wenn bei der Erstellung der Knotengruppe ein SSH-Schlüssel angegeben wird, wird ein Admin-Container aktiviert. Wir empfehlen, den Administrator-Container nur für Entwicklungs- und Testszenarien zu verwenden. Es wird nicht empfohlen, ihn in Produktionsumgebungen zu verwenden. Weitere Informationen finden Sie unter Administrator-Container auf GitHub.

Weitere Informationen

Weitere Informationen zu Amazon-EKS-optimierten Bottlerocket-AMIs finden Sie in den folgenden Abschnitten:

- Informationen zu Bottlerocket finden Sie in der Dokumentation und den Releases in GitHub.
- Informationen zur Verwendung von Bottlerocket mit verwalteten Knotengruppen finden Sie unter Verwaltete Knotengruppen.

- Informationen zum Starten von selbstverwalteten Bottlerocket-Knoten finden Sie unter Starten selbstverwalteter Bottlerocket-Knoten.
- Informationen zum Abrufen der neuesten IDs der für Amazon EKS optimierten Bottlerocket-AMIs finden Sie unter Abrufen von Amazon-EKS-optimierten Bottlerocket-AMI-IDs.
- Einzelheiten zur Compliance-Unterstützung finden Sie unter Bottlerocket-Compliance-Unterstützung.

Abrufen von Amazon-EKS-optimierten Bottlerocket-AMI-IDs

Sie können die Amazon Machine Image (AMI) -ID für Amazon EKS-optimierte AMIs abrufen, indem Sie die AWS Systems Manager Parameter Store-API abfragen. Mit diesem Parameter müssen Sie Amazon-EKS-optimierte AMI-IDs nicht manuell nachschlagen. Weitere Informationen zur Systems Manager Parameter Store-API finden Sie unter GetParameter. Das von Ihnen verwendete IAM-Prinzipal muss über die `ssm:GetParameter`-IAM-Berechtigung zum Abrufen der Amazon EKS-optimierten AMI-Metadaten verfügen.

Sie können die Image-ID des neuesten empfohlenen Amazon EKS-optimierten Bottlerocket AMIs mit dem folgenden AWS CLI Befehl abrufen, indem Sie den Unterparameter `image_id` verwenden. Ersetzen Sie *`1.30`* durch eine unterstützte Version und *`region-code`* durch eine Amazon-EKS-unterstützte Region, für die Sie die AMI-ID verwenden möchten.

```
aws ssm get-parameter --name /aws/service/bottlerocket/aws-k8s-1.30/x86_64/latest/
image_id --region region-code --query "Parameter.Value" --output text
```

Eine Beispielausgabe sieht wie folgt aus.

```
ami-1234567890abcdef0
```

Bottlerocket-Compliance-Unterstützung

Bottlerocket entspricht den von verschiedenen Organisationen festgelegten Empfehlungen:

- Es ist ein CIS-Benchmark definiert für Bottlerocket. In einer Standardkonfiguration verfügt das Bottlerocket-Image über die meisten Steuerelemente, die für das CIS-Level-1-Konfigurationsprofil erforderlich sind. Sie können die für ein CIS-Level-2-Konfigurationsprofil erforderlichen Steuerungen implementieren. Weitere Informationen finden Sie im AWS-Blog unter Validieren des für Amazon EKS optimierten Bottlerocket-AMI anhand des CIS-Benchmarks.

- Der optimierte Funktionsumfang und die reduzierte Angriffsfläche bedeuten, dass Bottlerocket-Instances weniger Konfiguration benötigen, um die PCI-DSS-Anforderungen zu erfüllen. Der CIS-Benchmark fürBottlerocket ist eine hervorragende Quelle für Härtungsrichtlinien und unterstützt Ihre Anforderungen an sichere Konfigurationsstandards gemäß der PCI-DSS-Anforderung 2.2. Sie können Fluent Bit auch nutzen, um Ihre Anforderungen an die Auditprotokollierung auf Betriebssystemebene gemäß der PCI DSS-Anforderung 10.2 zu erfüllen. AWS veröffentlicht regelmäßig neue (gepatchte) Bottlerocket-Instances, damit Sie die PCI-DSS-Anforderungen 6.2 (für v3.2.1) und 6.3.3 (für v4.0) erfüllen können.
- Bottlerocket ist eine HIPAA-fähige Funktion, die für die Verwendung mit regulierten Workloads sowohl für Amazon EC2 als auch für Amazon EKS autorisiert ist. Weitere Informationen finden Sie im Whitepaper Architecting for HIPAA Security and Compliance on Amazon EKS.

Für Amazon EKS optimierte Ubuntu Linux-AMIs

Canonical hat in enger Zusammenarbeit mit Amazon EKS Knoten-AMIs erstellt, die Sie in Ihren Clustern verwenden können.

Canonical liefert ein built-for-purpose Kubernetes Node OS-Image. Dieses minimierte Ubuntu Image ist für Amazon EKS optimiert und enthält den benutzerdefinierten AWS Kernel, der gemeinsam mit AWS entwickelt wurde. Weitere Informationen finden Sie unter UbuntuAmazon Elastic Kubernetes Service (EKS) und. Starten selbstverwalteter Ubuntu-Knoten Informationen zum Support finden Sie im Abschnitt Software von Drittanbietern in den häufig gestellten Fragen zum AWS Premium-Support.

Amazon-EKS-optimierte Windows-AMIs

Windows Amazon-EKS-optimierte AMIs werden auf Windows-Server 2019 und Windows-Server 2022 entwickelt. Sie sind so konfiguriert, dass sie als Basis-Image für Amazon-EKS-Knoten dienen. Standardmäßig enthalten die AMIs die folgenden Komponenten:

- `kubelet`
- `kube-proxy`
- AWS IAM Authenticator für Kubernetes
- `csi-proxy`
- `containerd`

> **Note**
>
> Sie können Sicherheits- oder Datenschutzereignisse für Windows Server mit dem Microsoft Security Update Guide verfolgen.

Amazon EKS bietet AMIs, die für Windows-Container optimiert sind, in den folgenden Varianten an.

- Amazon-EKS-optimierte Windows-Server-2019-Core-AMI
- Amazon-EKS-optimierte Windows-Server-2019-Full-AMI
- Amazon-EKS-optimierte Windows-Server-2022-Core-AMI
- Amazon-EKS-optimierte Windows-Server-2022-Full-AMI

> **Important**
>
> - Das für Amazon EKS optimierte Windows-Server-20H2-Core-AMI ist veraltet. Es werden keine neuen Versionen dieses AMI veröffentlicht.
> - Um sicherzustellen, dass Sie standardmäßig über die neuesten Sicherheitsupdates verfügen, verwaltet Amazon EKS optimierte Windows AMIs für die letzten 4 Monate. Jedes neue AMI wird ab dem Zeitpunkt der ersten Veröffentlichung für 4 Monate verfügbar sein. Nach Ablauf dieses Zeitraums werden ältere AMIs als privat eingestuft und sind nicht mehr zugänglich. Wir empfehlen die Verwendung der neuesten AMIs, um Sicherheitslücken und den Verlust des Zugriffs auf ältere AMIs zu vermeiden, deren unterstützte Lebensdauer abgelaufen ist. Wir können zwar nicht garantieren, dass wir Zugriff auf AMIs gewähren können, die privat gemacht wurden, aber Sie können den Zugriff beantragen, indem Sie ein Ticket bei einreichen AWS Support.

Veröffentlichungskalender

In der folgenden Tabelle sind die Veröffentlichungs- und Daten für das Ende der Unterstützung für Windows-Versionen auf Amazon EKS aufgeführt. Wenn ein Enddatum leer ist, liegt dies daran, dass die Version weiterhin unterstützt wird.

Windows-Version	Amazon-EKS-Version	Ende der Unterstützung für Amazon EKS
Windows Server 2022 Core	10/17/2022	
Windows Server 2022 Full	10/17/2022	
Windows Server 20H2 Core	8/12/2021	8/9/2022
Windows Server 2004 Core	8/19/2020	12/14/2021
Windows Server 2019 Core	10/7/2019	
Windows Server 2019 Full	10/7/2019	
Windows Server 1909 Core	10/7/2019	12/8/2020

Bootstrap-Skript-Konfigurationsparameter

Wenn Sie einen Windows-Knoten erstellen, gibt es auf dem Knoten ein Skript, mit dem unterschiedliche Parameter konfiguriert werden können. Abhängig von Ihrem Setup kann dieses Skript auf dem Knoten an einem Ort gefunden werden, der ähnlich ist wie: `C:\Program Files\Amazon\EKS\Start-EKSBootstrap.ps1`. Sie können benutzerdefinierte Parameterwerte angeben, indem Sie sie als Argumente für das Bootstrap-Skript angeben. So können Sie beispielsweise die Benutzerdaten in der Startvorlage aktualisieren. Weitere Informationen finden Sie unter Amazon-EC2-Benutzerdaten.

Das Skript enthält die folgenden Befehlszeilenparameter:

- `-EKSClusterName` - Gibt den Namen des Amazon-EKS-Clusters an, dem dieser Worker-Knoten beitreten soll.
- `-KubeletExtraArgs` - Gibt zusätzliche Argumente für `kubelet` an (optional).
- `-KubeProxyExtraArgs` - Gibt zusätzliche Argumente für `kube-proxy` an (optional).
- `-APIServerEndpoint` - Gibt den Amazon-EKS-Cluster-API-Server-Endpunkt an (optional). Nur gültig bei Verwendung mit `-Base64ClusterCA`. Umgehung des Anrufs `Get-EKSCluster`.
- `-Base64ClusterCA` - Gibt den base64-codierten Cluster-CA-Inhalt an (optional). Nur gültig bei Verwendung mit `-APIServerEndpoint`. Umgehung des Anrufs `Get-EKSCluster`.

- `-DNSClusterIP` - Überschreibt die IP-Adresse, die für DNS-Abfragen innerhalb des Clusters verwendet werden soll (optional). Der Standardwert ist `10.100.0.10` oder `172.20.0.10`, basierend auf der IP-Adresse der primären Schnittstelle.
- `-ServiceCIDR` – Überschreibt den Kubernetes-Service-IP-Adressbereich, aus dem Cluster adressiert werden. Der Standardwert ist `172.20.0.0/16` oder `10.100.0.0/16`, basierend auf der IP-Adresse der primären Schnittstelle.
- `-ExcludedSnatCIDRs` – Eine Liste von `IPv4`-CIDRs, die von der Source Network Address Translation (SNAT) ausgeschlossen werden sollen. Das bedeutet, dass die private Pod-IP, die über VPC adressierbar ist, nicht in die IP-Adresse der primären `IPv4`-Adresse der ENI der Instance für ausgehenden Datenverkehr übersetzt wird. Standardmäßig wird das `IPv4`-CIDR der VPC für den Amazon-EKS–Windows-Knoten hinzugefügt. Durch die Angabe von CIDRs für diesen Parameter werden auch die angegebenen CIDRs zusätzlich ausgeschlossen. Weitere Informationen finden Sie unter SNAT für Pods.

Zusätzlich zu den Befehlszeilenparametern können Sie auch einige Parameter für Umgebungsvariablen angeben. Wenn Sie einen Befehlszeilenparameter angeben, hat dieser Vorrang vor der jeweiligen Umgebungsvariablen. Die Umgebungsvariablen sollten als Maschinen- (oder System-) Bereich definiert werden, da das Bootstrap-Skript nur maschinenspezifische Variablen liest.

Das Skript berücksichtigt die folgenden Umgebungsvariablen:

- `SERVICE_IPV4_CIDR` – Die Definition finden Sie im `ServiceCIDR`-Befehlszeilenparameter.
- `EXCLUDED_SNAT_CIDRS` – Sollte eine durch Kommas getrennte Zeichenfolge sein. Die Definition finden Sie im `ExcludedSnatCIDRs`-Befehlszeilenparameter.

Selbstverwaltete Windows-Server 2022-Knoten mit **`eksctl`** starten

Sie können das folgende **`test-windows-2022.`**`yaml` als Referenz für die Ausführung von Windows-Server 2022 als selbstverwaltete Knoten verwenden. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte.

> **Note**
>
> Sie müssen `eksctl`-Version `0.116.0` oder höher verwenden, um selbstverwaltete Knoten für Windows-Server 2022 auszuführen.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: windows-2022-cluster
  region: region-code
  version: '1.30'

nodeGroups:
  - name: windows-ng
    instanceType: m5.2xlarge
    amiFamily: WindowsServer2022FullContainer
    volumeSize: 100
    minSize: 2
    maxSize: 3
  - name: linux-ng
    amiFamily: AmazonLinux2
    minSize: 2
    maxSize: 3
```

Die Knotengruppen können dann mit dem folgenden Befehl erstellt werden.

```
eksctl create cluster -f test-windows-2022.yaml
```

gMSA-Authentifizierungsunterstützung

Amazon-EKS-Windows-Pods ermöglichen verschiedene Arten der Gruppenauthentifizierung für Managed Service Account (gMSA).

- Amazon EKS unterstützt Active Directory-Domain-Identitäten für die Authentifizierung. Weitere Informationen zu Domain-joined gMSA finden Sie unter Windows-Authentifizierung in Amazon-EKS-Windows-pods im AWS -Blog.
- Amazon EKS bietet ein Plugin, mit dem non-domain-joined Windows Knoten gMSA Anmeldeinformationen mit einer portablen Benutzeridentität abrufen können. Weitere Informationen zu domailosen gMSA finden Sie unter Domainlose Windows-Authentifizierung für Amazon-EKS-Windows-pods im AWS -Blog.

Zwischengespeicherte Container-Images

Für Windows optimierte AMIs von Amazon EKS werden bestimmte Container-Images für die `containerd` Laufzeit zwischengespeichert. Container-Images werden zwischengespeichert, wenn benutzerdefinierte AMIs mit von Amazon verwalteten Build-Komponenten erstellt werden. Weitere Informationen finden Sie unter Verwenden der von Amazon verwalteten Build-Komponente.

Die folgenden zwischengespeicherten Container-Images sind für die `containerd`-Laufzeit bestimmt:

- `amazonaws.com/eks/pause-windows`
- `mcr.microsoft.com/windows/nanoserver`
- `mcr.microsoft.com/windows/servercore`

Weitere Informationen

Weitere Informationen zu Amazon-EKS-optimierten Windows-AMIs finden Sie in den folgenden Abschnitten:

- Informationen zur Verwendung von Windows mit verwalteten Knotengruppen finden Sie unter Verwaltete Knotengruppen.
- Informationen zum Starten von selbstverwalteten Windows-Knoten finden Sie unter Starten selbstverwalteter Windows-Knoten.
- Versionsinformationen finden Sie unter Amazon ECS-optimierte Windows-AMI-Versionen.
- Informationen zum Abrufen der neuesten IDs der für Amazon EKS optimierten Windows-AMIs finden Sie unter Abrufen von Amazon-EKS-optimierten Windows-AMI-IDs.
- Informationen dazu, wie Sie Amazon EC2 Image Builder verwenden, um benutzerdefinierte Amazon-EKS-optimierte Windows-AMIs zu erstellen, finden Sie unter Erstellen benutzerdefinierter Amazon-EKS-optimierter Windows-AMIs.
- Bewährte Methoden finden Sie unter Optimiertes Windows AMI-Management von Amazon EKS im EKS Best Practices Guide.

Amazon ECS-optimierte Windows-AMI-Versionen

> **Important**
>
> Erweiterter Support für Amazon Windows EKS-optimierte AMIs, die von veröffentlicht wurden, ist AWS nicht für Kubernetes Version, `1.23` aber für Kubernetes Version `1.24` und höher verfügbar.

In diesem Thema werden Versionen der für Amazon EKS optimierten Windows AMIs und die entsprechenden Versionen von `kubelet`, `containerd`, und aufgeführt `csi-proxy`.

Die Amazon-EKS-optimierten AMI-Metadaten, einschließlich der AMI-ID für jede Variante, können programmgesteuert abgerufen werden. Weitere Informationen finden Sie unter Abrufen von Amazon-EKS-optimierten Windows-AMI-IDs.

AMIs werden von der Kubernetes-Version und dem Veröffentlichungsdatum des AMI in folgendem Format versioniert:

```
k8s_major_version.k8s_minor_version-release_date
```

> **Note**
>
> Von Amazon EKS verwaltete Knotengruppen unterstützen die Windows AMI-Versionen vom November 2022 und später.

Amazon-EKS-optimierte Windows-Server-2022-Core-AMI

In den folgenden Tabellen sind die aktuellen und früheren Versionen des Amazon-EKS-optimierten Windows-Server-2022-Core-AMI aufgeführt.

Kubernetes version 1.30

Kubernetes-Version **`1.30`**

AMI-Version	**`kubelet`**-Version	**`containerd`** - Version	**`csi-proxy`** - Version	Versionshinweise
`1.30-2024.05.15`	`1.30.0`	`1.6.28`	`1.1.2`	

Kubernetes version 1,29

Kubernetes-Version **`1.29`**

AMI-Version	**`kubelet`**-Version	**`containerd`** - Version	**`csi-proxy`** - Version	Versionshinweise
`1.29-2024.05.15`	`1.29.3`	`1.7.11`	`1.1.2`	`containerd` wurde auf `1.7.11` aktualisiert. `kubelet` wurde auf `1.29.3` aktualisiert.
`1.29-2024.04.09`	`1.29.0`	`1.6.28`	`1.1.2`	`containerd` wurde auf `1.6.28` aktualisiert. CNI neu erstellt und `csi-proxy` verwendet. `golang 1.22.1`
`1.29-2024.03.12`	`1.29.0`	`1.6.25`	`1.1.2`	
`1.29-2024.02.13`	`1.29.0`	`1.6.25`	`1.1.2`	
`1.29-2024.02.06`	`1.29.0`	`1.6.25`	`1.1.2`	Es wurde ein Fehler behoben, bei dem das Pause-Bild fälschlicherweise beim `kubelet` Garbage-Collection-Prozess gelöscht wurde.

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.29-2024.01.11	1.29.0	1.6.18	1.1.2	Ausgeschlossen ist das eigenständige Windows Update KB5034439 auf Windows Server 2022 Core-AMIs. Die KB gilt nur für Windows Installationen mit einer separaten WinRE Partition, die in keinem unserer Amazon EKS-optimierten Windows AMIs enthalten sind.

Kubernetes version 1.28

Kubernetes-Version **1.28**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.28-2024.05.14	1.28.8	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert. kubelet wurde auf 1.28.8 aktualisiert.
1.28-2024.04.09	1.28.5	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.28-2024.03.12	1.28.5	1.6.18	1.1.2	
1.28-2024.02.13	1.28.5	1.6.18	1.1.2	

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.28-2024.01.11	1.28.5	1.6.18	1.1.2	Ausgenommen das eigenständige Windows Update KB5034439 auf Windows Server 2022 Core-AMIs. Die KB gilt nur für Windows Installationen mit einer separaten WinRE Partition, die in keinem unserer Amazon EKS-optimierten Windows AMIs enthalten sind.
1.28-2023.12.12	1.28.3	1.6.18	1.1.2	
1.28-2023.11.14	1.28.3	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.28-2023.10.19	1.28.2	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.28-2023-09.27	1.28.2	1.6.6	1.1.2	Ein Security Advisory wurde in kubelet behoben.
1.28-2023.09.12	1.28.1	1.6.6	1.1.2	

Kubernetes version 1.27

Kubernetes-Version **`1.27`**

AMI-Version	**`kubelet`**-Version	**`containerd`** - Version	**`csi-proxy`** - Version	Versionshinweise
`1.27-2024.05.14`	`1.27.12`	`1.6.28`	`1.1.2`	`containerd` wurde auf `1.6.28` aktualisiert. `kubelet` wurde auf `1.27.12` aktualisiert.
`1.27-2024.04.09`	`1.27.9`	`1.6.25`	`1.1.2`	`containerd` wurde auf `1.6.25` aktualisiert. CNI neu erstellt und `csi-proxy` verwendet. `golang 1.22.1`
`1.27-2024.03.12`	`1.27.9`	`1.6.18`	`1.1.2`	
`1.27-2024.02.13`	`1.27.9`	`1.6.18`	`1.1.2`	
`1.27-2024.01.11`	`1.27.9`	`1.6.18`	`1.1.2`	Ausgenommen das eigenständige Windows Update KB5034439 auf Windows Server 2022 Core-AMIs. Die KB gilt nur für Windows Installationen mit einer separaten WinRE Partition, die in keinem unserer Amazon EKS-optimierten Windows AMIs enthalten sind.
`1.27-2023.12.12`	`1.27.7`	`1.6.18`	`1.1.2`	
`1.27-2023.11.14`	`1.27.7`	`1.6.18`	`1.1.2`	Beinhaltet Patches für `CVE-2023-5528` .

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.27-2023.10.19	1.27.6	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.27-2023-09.27	1.27.6	1.6.6	1.1.2	Ein Security Advisory wurde in kubelet behoben.
1.27-2023.09.12	1.27.4	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.27-2023.08.17	1.27.4	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.27-2023.08.08	1.27.3	1.6.6	1.1.1	
1.27-2023.07.11	1.27.3	1.6.6	1.1.1	
1.27-2023.06.20	1.27.1	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2023.06.14	1.27.1	1.6.6	1.1.1	Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.27-2023.06.06	1.27.1	1.6.6	1.1.1	Das Problem #2042 von containers-roadmap , das dazu führte, dass Knoten private Amazon-ECR-Images nicht abrufen konnten, wurde behoben.
1.27-2023.05.17	1.27.1	1.6.6	1.1.1	

Kubernetes version 1.26

Kubernetes-Version **1.26**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.26-2024.05.14	1.26.15	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert. kubelet wurde auf 1.26.15 aktualisiert.
1.26-2024.04.09	1.26.12	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.26-2024.03.12	1.26.12	1.6.18	1.1.2	

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.26-2024.02.13	1.26.12	1.6.18	1.1.2	
1.26-2024.01.11	1.26.12	1.6.18	1.1.2	Ausgenommen das eigenständige Windows Update KB5034439 auf Windows Server 2022 Core-AMIs. Die KB gilt nur für Windows Installationen mit einer separaten WinRE Partition, die in keinem unserer Amazon EKS-optimierten Windows AMIs enthalten sind.
1.26-2023.12.12	1.26.10	1.6.18	1.1.2	
1.26-2023.11.14	1.26.10	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.26-2023.10.19	1.26.9	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.26.9 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.26-2023.09.12	1.26.7	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.26-2023.08.17	1.26.7	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.26-2023.08.08	1.26.6	1.6.6	1.1.1	
1.26-2023.07.11	1.26.6	1.6.6	1.1.1	
1.26-2023.06.20	1.26.4	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.26-2023.06.14	1.26.4	1.6.6	1.1.1	Kubernetes wurde auf 1.26.4 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.26-2023.05.09	1.26.2	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.26-2023.04.26	1.26.2	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.26-2023.04.11	1.26.2	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.26-2023.03.24	1.26.2	1.6.6	1.1.1	

Kubernetes version 1.25

Kubernetes-Version **1.25**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2024.05.14	1.25.16	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert.
1.25-2024.04.09	1.25.16	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.25-2024.03.12	1.25.16	1.6.18	1.1.2	
1.25-2024.02.13	1.25.16	1.6.18	1.1.2	
1.25-2024.01.11	1.25.16	1.6.18	1.1.2	Ausgenommen das eigenständige Windows Update KB5034439 auf Windows Server 2022 Core-AMIs. Die KB gilt nur

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
				für Windows Installationen mit einer separaten WinRE Partition , die in keinem unserer Amazon EKS-optimierten Windows AMIs enthalten sind.
1.25-2023.12.12	1.25.15	1.6.18	1.1.2	
1.25-2023.11.14	1.25.15	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.25-2023.10.19	1.25.14	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.25.14 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.25-2023.09.12	1.25.12	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.25-2023.08.17	1.25.12	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.25-2023.08.08	1.25.9	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2023.07.11	1.25.9	1.6.6	1.1.1	
1.25-2023.06.20	1.25.9	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.25-2023.06.14	1.25.9	1.6.6	1.1.1	Kubernetes wurde auf 1.25.9 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.25-2023.05.09	1.25.7	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.25-2023.04.11	1.25.7	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.25-2023.03.27	1.25.6	1.6.6	1.1.1	Es wurde ein domainloses gMSA-Plugin installiert, um die gMSA-Authentifizierung für Windows-Container in Amazon EKS zu erleichtern.

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2023.03.20	1.25.6	1.6.6	1.1.1	
1.25-2023.02.14	1.25.6	1.6.6	1.1.1	

Kubernetes version 1.24

Kubernetes-Version **1.24**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2024.05.14	1.24.17	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert.
1.24-2024.04.09	1.24.17	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.24-2024.03.12	1.24.17	1.6.18	1.1.2	
1.24-2024.02.13	1.24.17	1.6.18	1.1.2	
1.24-2024.01.11	1.24.17	1.6.18	1.1.2	Ausgenommen das eigenständige Windows Update KB5034439 auf Windows Server 2022 Core-AMIs. Die KB gilt nur für Windows Installationen mit einer separaten WinRE Partition

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
				, die in keinem unserer Amazon EKS-optimierten Windows AMIs enthalten sind.
1.24-2023.12.12	1.24.17	1.6.18	1.1.2	
1.24-2023.11.14	1.24.17	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.24-2023.10.19	1.24.17	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.24.17 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.24-2023.09.12	1.24.16	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.24-2023.08.17	1.24.16	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.24-2023.08.08	1.24.13	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2023.07.11	1.24.13	1.6.6	1.1.1	
1.24-2023.06.20	1.24.13	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.24-2023.06.14	1.24.13	1.6.6	1.1.1	Kubernetes wurde auf 1.24.13 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.24-2023.05.09	1.24.7	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.24-2023.04.11	1.24.7	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.24-2023.03.27	1.24.7	1.6.6	1.1.1	Es wurde ein domainloses gMSA-Plugin installiert, um die gMSA-Authentifizierung für Windows-Container in Amazon EKS zu erleichtern.

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2023.03.20	1.24.7	1.6.6	1.1.1	Kubernetes-Version wurde auf 1.24.7 herabgestuft, weil 1.24.10 ein Problem in kube-proxy gemeldet hat.
1.24-2023.02.14	1.24.10	1.6.6	1.1.1	
1.24-2023.01.23	1.24.7	1.6.6	1.1.1	
1.24-2023.01.11	1.24.7	1.6.6	1.1.1	
1.24-2022.12.13	1.24.7	1.6.6	1.1.1	
1.24-2022.10.11	1.24.7	1.6.6	1.1.1	

Amazon-EKS-optimierte Windows-Server-2022-Full-AMI

In den folgenden Tabellen sind die aktuellen und früheren Versionen des Amazon-EKS-optimierten Windows-Server-2022-Full-AMI aufgeführt.

Kubernetes version 1.30

Kubernetes-Version **1.30**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.30-2024.05.15	1.30.0	1.6.28	1.1.2	

Kubernetes version 1,29

Kubernetes-Version **`1.29`**

AMI-Version	**`kubelet`**-Version	**`containerd`** -Version	**`csi-proxy`** -Version	Versionshinweise
`1.29-2024.05.15`	`1.29.3`	`1.7.11`	`1.1.2`	`containerd` wurde auf `1.7.11` aktualisiert. `kubelet` wurde auf `1.29.3` aktualisiert.
`1.29-2024.04.09`	`1.29.0`	`1.6.28`	`1.1.2`	`containerd` wurde auf `1.6.28` aktualisiert. CNI neu erstellt und `csi-proxy` verwendet. `golang 1.22.1`
`1.29-2024.03.12`	`1.29.0`	`1.6.25`	`1.1.2`	
`1.29-2024.02.13`	`1.29.0`	`1.6.25`	`1.1.2`	
`1.29-2024.02.06`	`1.29.0`	`1.6.25`	`1.1.2`	Es wurde ein Fehler behoben, bei dem das Pause-Bild fälschlic herweise beim `kubelet` Garbage-Collection-Prozess gelöscht wurde.
`1.29-2024.01.09`	`1.29.0`	`1.6.18`	`1.1.2`	

Kubernetes version 1.28

Kubernetes-Version **1.28**

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.28-2024.05.14	1.28.8	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert. kubelet wurde auf 1.28.8 aktualisiert.
1.28-2024.04.09	1.28.5	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.28-2024.03.12	1.28.5	1.6.18	1.1.2	
1.28-2024.02.13	1.28.5	1.6.18	1.1.2	
1.28-2024.01.09	1.28.5	1.6.18	1.1.2	
1.28-2023.12.12	1.28.3	1.6.18	1.1.2	
1.28-2023.11.14	1.28.3	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.28-2023.10.19	1.28.2	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.28-2023-09.27	1.28.2	1.6.6	1.1.2	Ein Security Advisory wurde in kubelet behoben.
1.28-2023.09.12	1.28.1	1.6.6	1.1.2	

Kubernetes version 1.27

Kubernetes-Version **1.27**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2024.05.14	1.27.12	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert. kubelet wurde auf 1.27.12 aktualisiert.
1.27-2024.04.09	1.27.9	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.27-2024.03.12	1.27.9	1.6.18	1.1.2	
1.27-2024.02.13	1.27.9	1.6.18	1.1.2	
1.27-2024.01.09	1.27.9	1.6.18	1.1.2	
1.27-2023.12.12	1.27.7	1.6.18	1.1.2	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2023.11.14	1.27.7	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.27-2023.10.19	1.27.6	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.27-2023-09.27	1.27.6	1.6.6	1.1.2	Ein Security Advisory wurde in kubelet behoben.
1.27-2023.09.12	1.27.4	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.27-2023.08.17	1.27.4	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.27-2023.08.08	1.27.3	1.6.6	1.1.1	
1.27-2023.07.11	1.27.3	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2023.06.20	1.27.1	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.27-2023.06.14	1.27.1	1.6.6	1.1.1	Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.27-2023.06.06	1.27.1	1.6.6	1.1.1	Das Problem #2042 von `containers-roadmap` , das dazu führte, dass Knoten private Amazon-ECR-Images nicht abrufen konnten, wurde behoben.
1.27-2023.05.18	1.27.1	1.6.6	1.1.1	

Kubernetes version 1.26

Kubernetes-Version **1.26**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.26-2024.05.14	1.26.15	1.6.28	1.1.2	`containerd` wurde auf `1.6.28` aktualisiert. `kubelet` wurde auf `1.26.15` aktualisiert.
1.26-2024.04.09	1.26.12	1.6.25	1.1.2	`containerd` wurde auf `1.6.25` aktualisiert. CNI neu

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
				erstellt und `csi-proxy` verwendet. `golang 1.22.1`
`1.26-2024.03.12`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2024.02.13`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2024.01.09`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2023.12.12`	`1.26.10`	`1.6.18`	`1.1.2`	
`1.26-2023.11.14`	`1.26.10`	`1.6.18`	`1.1.2`	Beinhaltet Patches für `CVE-2023-5528` .
`1.26-2023.10.19`	`1.26.9`	`1.6.18`	`1.1.2`	`containerd` wurde auf `1.6.18` aktualisiert. `kubelet` wurde auf `1.26.9` aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (`SERVICE_IPV4_CIDR` und `EXCLUDED_SNAT_CIDRS`).
`1.26-2023.09.12`	`1.26.7`	`1.6.6`	`1.1.2`	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.26-2023.08.17	1.26.7	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.26-2023.08.08	1.26.6	1.6.6	1.1.1	
1.26-2023.07.11	1.26.6	1.6.6	1.1.1	
1.26-2023.06.20	1.26.4	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.26-2023.06.14	1.26.4	1.6.6	1.1.1	Kubernetes wurde auf 1.26.4 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.26-2023.05.09	1.26.2	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.26-2023.04.26	1.26.2	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.26-2023.04.11	1.26.2	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.26-2023.03.24	1.26.2	1.6.6	1.1.1	

Kubernetes version 1.25

Kubernetes-Version **1.25**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2024.05.14	1.25.16	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert.
1.25-2024.04.09	1.25.16	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.25-2024.03.12	1.25.16	1.6.18	1.1.2	
1.25-2024.02.13	1.25.16	1.6.18	1.1.2	
1.25-2024.01.09	1.25.16	1.6.18	1.1.2	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2023.12.12	1.25.15	1.6.18	1.1.2	
1.25-2023.11.14	1.25.15	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.25-2023.10.19	1.25.14	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.25.14 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.25-2023.09.12	1.25.12	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.25-2023.08.17	1.25.12	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.25-2023.08.08	1.25.9	1.6.6	1.1.1	
1.25-2023.07.11	1.25.9	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.25-2023.06.20	1.25.9	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.25-2023.06.14	1.25.9	1.6.6	1.1.1	Kubernetes wurde auf 1.25.9 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.25-2023.05.09	1.25.7	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.25-2023.04.11	1.25.7	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.25-2023.03.27	1.25.6	1.6.6	1.1.1	Es wurde ein domainloses gMSA-Plugin installiert, um die gMSA-Authentifizierung für Windows-Container in Amazon EKS zu erleichtern.
1.25-2023.03.20	1.25.6	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2023.02.14	1.25.6	1.6.6	1.1.1	

Kubernetes version 1.24

Kubernetes-Version **1.24**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2024.05.14	1.24.17	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert.
1.24-2024.04.09	1.24.17	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.24-2024.03.12	1.24.17	1.6.18	1.1.2	
1.24-2024.02.13	1.24.17	1.6.18	1.1.2	
1.24-2024.01.09	1.24.17	1.6.18	1.1.2	
1.24-2023.12.12	1.24.17	1.6.18	1.1.2	
1.24-2023.11.14	1.24.17	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.24-2023.10.19	1.24.17	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.24.17 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.24-2023.09.12	1.24.16	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.24-2023.08.17	1.24.16	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.24-2023.08.08	1.24.13	1.6.6	1.1.1	
1.24-2023.07.11	1.24.13	1.6.6	1.1.1	
1.24-2023.06.20	1.24.13	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.24-2023.06.14	1.24.13	1.6.6	1.1.1	Kubernetes wurde auf 1.24.13 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.24-2023.05.09	1.24.7	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.24-2023.04.11	1.24.7	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.24-2023.03.27	1.24.7	1.6.6	1.1.1	Es wurde ein domainloses gMSA-Plugin installiert, um die gMSA-Authentifizierung für Windows-Container in Amazon EKS zu erleichtern.
1.24-2023.03.20	1.24.7	1.6.6	1.1.1	Kubernetes-Version wurde auf 1.24.7 herabgestuft, weil 1.24.10 ein Problem in kube-proxy gemeldet hat.
1.24-2023.02.14	1.24.10	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2023.01.23	1.24.7	1.6.6	1.1.1	
1.24-2023.01.11	1.24.7	1.6.6	1.1.1	
1.24-2022.12.14	1.24.7	1.6.6	1.1.1	
1.24-2022.10.11	1.24.7	1.6.6	1.1.1	

Amazon-EKS-optimierte Windows-Server-2019-Core-AMI

In den folgenden Tabellen sind die aktuellen und früheren Versionen des Amazon-EKS-optimierten Windows-Server-2019-Core-AMI aufgeführt.

Kubernetes version 1,30

Kubernetes-Version **1.30**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.30-2024.05.15	1.30.0	1.6.28	1.1.2	

Kubernetes version 1,29

Kubernetes-Version **`1.29`**

AMI-Version	**`kubelet`**-Version	**`containerd`** -Version	**`csi-proxy`** -Version	Versionshinweise
`1.29-2024.05.15`	`1.29.3`	`1.7.11`	`1.1.2`	`containerd` wurde auf `1.7.11` aktualisiert. `kubelet` wurde auf `1.29.3` aktualisiert.
`1.29-2024.04.09`	`1.29.0`	`1.6.28`	`1.1.2`	`containerd` wurde auf `1.6.28` aktualisiert. CNI neu erstellt und `csi-proxy` verwendet. `golang 1.22.1`
`1.29-2024.03.13`	`1.29.0`	`1.6.25`	`1.1.2`	
`1.29-2024.02.13`	`1.29.0`	`1.6.25`	`1.1.2`	
`1.29-2024.02.06`	`1.29.0`	`1.6.25`	`1.1.2`	Es wurde ein Fehler behoben, bei dem das Pause-Bild fälschlicherweise beim `kubelet` Garbage-Collection-Prozess gelöscht wurde.
`1.29-2024.01.09`	`1.29.0`	`1.6.18`	`1.1.2`	

Kubernetes version 1.28

Kubernetes-Version **1.28**

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.28-2024.05.14	1.28.8	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert. kubelet wurde auf 1.28.8 aktualisiert.
1.28-2024.04.09	1.28.5	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.28-2024.03.13	1.28.5	1.6.18	1.1.2	
1.28-2024.02.13	1.28.5	1.6.18	1.1.2	
1.28-2024.01.09	1.28.5	1.6.18	1.1.2	
1.28-2023.12.12	1.28.3	1.6.18	1.1.2	
1.28-2023.11.14	1.28.3	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.28-2023.10.19	1.28.2	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.28-2023-09.27	1.28.2	1.6.6	1.1.2	Ein Security Advisory wurde in kubelet behoben.
1.28-2023.09.12	1.28.1	1.6.6	1.1.2	

Kubernetes version 1.27

Kubernetes-Version **1.27**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2024.05.14	1.27.12	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert. kubelet wurde auf 1.27.12 aktualisiert.
1.27-2024.04.09	1.27.9	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.27-2024.03.13	1.27.9	1.6.18	1.1.2	
1.27-2024.02.13	1.27.9	1.6.18	1.1.2	
1.27-2024.01.09	1.27.9	1.6.18	1.1.2	
1.27-2023.12.12	1.27.7	1.6.18	1.1.2	

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.27-2023.11.14	1.27.7	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.27-2023.10.19	1.27.6	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.27-2023-09.27	1.27.6	1.6.6	1.1.2	Ein Security Advisory wurde in kubelet behoben.
1.27-2023.09.12	1.27.4	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.27-2023.08.17	1.27.4	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.27-2023.08.08	1.27.3	1.6.6	1.1.1	
1.27-2023.07.11	1.27.3	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2023.06.20	1.27.1	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.27-2023.06.14	1.27.1	1.6.6	1.1.1	Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.27-2023.06.06	1.27.1	1.6.6	1.1.1	Das Problem #2042 von `containers-roadmap` , das dazu führte, dass Knoten private Amazon-ECR-Images nicht abrufen konnten, wurde behoben.
11.27-2023.05.18	1.27.1	1.6.6	1.1.1	

Kubernetes version 1.26

Kubernetes-Version **1.26**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.26-2024.05.14	1.26.15	1.6.28	1.1.2	`containerd` wurde auf `1.6.28` aktualisiert. `kubelet` wurde auf `1.26.15` aktualisiert.
1.26-2024.04.09	1.26.12	1.6.25	1.1.2	`containerd` wurde auf `1.6.25` aktualisiert. CNI neu

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
				erstellt und `csi-proxy` verwendet. `golang 1.22.1`
`1.26-2024.03.13`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2024.02.13`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2024.01.09`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2023.12.12`	`1.26.10`	`1.6.18`	`1.1.2`	
`1.26-2023.11.14`	`1.26.10`	`1.6.18`	`1.1.2`	Beinhaltet Patches für `CVE-2023-5528` .
`1.26-2023.10.19`	`1.26.9`	`1.6.18`	`1.1.2`	`containerd` wurde auf `1.6.18` aktualisiert. `kubelet` wurde auf `1.26.9` aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (`SERVICE_IPV4_CIDR` und `EXCLUDED_SNAT_CIDRS`).
`1.26-2023.09.12`	`1.26.7`	`1.6.6`	`1.1.2`	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.26-2023.08.17	1.26.7	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.26-2023.08.08	1.26.6	1.6.6	1.1.1	
1.26-2023.07.11	1.26.6	1.6.6	1.1.1	
1.26-2023.06.20	1.26.4	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.26-2023.06.14	1.26.4	1.6.6	1.1.1	Kubernetes wurde auf 1.26.4 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.26-2023.05.09	1.26.2	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.26-2023.04.26	1.26.2	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.26-2023.04.11	1.26.2	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.26-2023.03.24	1.26.2	1.6.6	1.1.1	

Kubernetes version 1.25

Kubernetes-Version **1.25**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2024.05.14	1.25.16	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert.
1.25-2024.04.09	1.25.16	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.25-2024.03.13	1.25.16	1.6.18	1.1.2	
1.25-2024.02.13	1.25.16	1.6.18	1.1.2	
1.25-2024.01.09	1.25.16	1.6.18	1.1.2	

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.25-2023.12.12	1.25.15	1.6.18	1.1.2	
1.25-2023.11.14	1.25.15	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.25-2023.10.19	1.25.14	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.25.14 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.25-2023.09.12	1.25.12	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.25-2023.08.17	1.25.12	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.25-2023.08.08	1.25.9	1.6.6	1.1.1	
1.25-2023.07.11	1.25.9	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.25-2023.06.20	1.25.9	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.25-2023.06.14	1.25.9	1.6.6	1.1.1	Kubernetes wurde auf 1.25.9 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.25-2023.05.09	1.25.7	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.25-2023.04.11	1.25.7	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.25-2023.03.27	1.25.6	1.6.6	1.1.1	Es wurde ein domainloses gMSA-Plugin installiert, um die gMSA-Authentifizierung für Windows-Container in Amazon EKS zu erleichtern.
1.25-2023.03.20	1.25.6	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2023.02.14	1.25.6	1.6.6	1.1.1	

Kubernetes version 1.24

Kubernetes-Version **1.24**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2024.05.14	1.24.17	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert.
1.24-2024.04.09	1.24.17	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.24-2024.03.13	1.24.17	1.6.18	1.1.2	
1.24-2024.02.13	1.24.17	1.6.18	1.1.2	
1.24-2024.01.09	1.24.17	1.6.18	1.1.2	
1.24-2023.12.12	1.24.17	1.6.18	1.1.2	
1.24-2023.11.14	1.24.17	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.24-2023.10.19	1.24.17	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.24.17 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.24-2023.09.12	1.24.16	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.24-2023.08.17	1.24.16	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.24-2023.08.08	1.24.13	1.6.6	1.1.1	
1.24-2023.07.11	1.24.13	1.6.6	1.1.1	
1.24-2023.06.20	1.24.13	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.24-2023.06.14	1.24.13	1.6.6	1.1.1	Kubernetes wurde auf 1.24.13 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.24-2023.05.09	1.24.7	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.24-2023.04.11	1.24.7	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.24-2023.03.27	1.24.7	1.6.6	1.1.1	Es wurde ein domainloses gMSA-Plugin installiert, um die gMSA-Authentifizierung für Windows-Container in Amazon EKS zu erleichtern.
1.24-2023.03.20	1.24.7	1.6.6	1.1.1	Kubernetes-Version wurde auf 1.24.7 herabgestuft, weil 1.24.10 ein Problem in kube-proxy gemeldet hat.
1.24-2023.02.14	1.24.10	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2023.01.23	1.24.7	1.6.6	1.1.1	
1.24-2023.01.11	1.24.7	1.6.6	1.1.1	
1.24-2022.12.13	1.24.7	1.6.6	1.1.1	
1.24-2022.11.08	1.24.7	1.6.6	1.1.1	

Amazon-EKS-optimierte Windows-Server-2019-Full-AMI

In den folgenden Tabellen sind die aktuellen und früheren Versionen des Amazon-EKS-optimierten Windows-Server-2019-Full-AMI aufgeführt.

Kubernetes version 1,30

Kubernetes-Version **1.30**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.30-2024.05.15	1.30.0	1.6.28	1.1.2	

Kubernetes version 1,29

Kubernetes-Version **`1.29`**

AMI-Version	**`kubelet`**-Version	**`containerd`** - Version	**`csi-proxy`** - Version	Versionshinweise
`1.29-2024.05.15`	`1.29.3`	`1.7.11`	`1.1.2`	`containerd` wurde auf `1.7.11` aktualisiert. `kubelet` wurde auf `1.29.3` aktualisiert.
`1.29-2024.04.09`	`1.29.0`	`1.6.28`	`1.1.2`	`containerd` wurde auf `1.6.28` aktualisiert. CNI neu erstellt und `csi-proxy` verwendet. `golang 1.22.1`
`1.29-2024.03.13`	`1.29.0`	`1.6.25`	`1.1.2`	
`1.29-2024.02.13`	`1.29.0`	`1.6.25`	`1.1.2`	
`1.29-2024.02.06`	`1.29.0`	`1.6.25`	`1.1.2`	Es wurde ein Fehler behoben, bei dem das Pause-Bild fälschlicherweise beim `kubelet` Garbage-Collection-Prozess gelöscht wurde.
`1.29-2024.01.09`	`1.29.0`	`1.6.18`	`1.1.2`	

Kubernetes version 1.28

Kubernetes-Version **1.28**

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.28-2024.05.14	1.28.8	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert. kubelet wurde auf 1.28.8 aktualisiert.
1.28-2024.04.09	1.28.5	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.28-2024.03.13	1.28.5	1.6.18	1.1.2	
1.28-2024.02.13	1.28.5	1.6.18	1.1.2	
1.28-2024.01.09	1.28.5	1.6.18	1.1.2	
1.28-2023.12.12	1.28.3	1.6.18	1.1.2	
1.28-2023.11.14	1.28.3	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.28-2023.10.19	1.28.2	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.28-2023-09.27	1.28.2	1.6.6	1.1.2	Ein Security Advisory wurde in kubelet behoben.
1.28-2023.09.12	1.28.1	1.6.6	1.1.2	

Kubernetes version 1.27

Kubernetes-Version **1.27**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2024.05.14	1.27.12	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert. kubelet wurde auf 1.27.12 aktualisiert.
1.27-2024.04.09	1.27.9	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.27-2024.03.13	1.27.9	1.6.18	1.1.2	
1.27-2024.02.13	1.27.9	1.6.18	1.1.2	
1.27-2024.01.09	1.27.9	1.6.18	1.1.2	
1.27-2023.12.12	1.27.7	1.6.18	1.1.2	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2023.11.14	1.27.7	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.27-2023.10.19	1.27.6	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.27-2023-09.27	1.27.6	1.6.6	1.1.2	Ein Security Advisory wurde in kubelet behoben.
1.27-2023.09.12	1.27.4	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.27-2023.08.17	1.27.4	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.27-2023.08.08	1.27.3	1.6.6	1.1.1	
1.27-2023.07.11	1.27.3	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.27-2023.06.20	1.27.1	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.27-2023.06.14	1.27.1	1.6.6	1.1.1	Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.27-2023.06.06	1.27.1	1.6.6	1.1.1	Das Problem #2042 von `containers-roadmap` , das dazu führte, dass Knoten private Amazon-ECR-Images nicht abrufen konnten, wurde behoben.
1.27-2023.05.17	1.27.1	1.6.6	1.1.1	

Kubernetes version 1.26

Kubernetes-Version **1.26**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.26-2024.05.14	1.26.15	1.6.28	1.1.2	`containerd` wurde auf `1.6.28` aktualisiert. `kubelet` wurde auf `1.26.15` aktualisiert.
1.26-2024.04.09	1.26.12	1.6.25	1.1.2	`containerd` wurde auf `1.6.25` aktualisiert. CNI neu

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
				erstellt und `csi-proxy` verwendet. `golang 1.22.1`
`1.26-2024.03.13`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2024.02.13`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2024.01.09`	`1.26.12`	`1.6.18`	`1.1.2`	
`1.26-2023.12.12`	`1.26.10`	`1.6.18`	`1.1.2`	
`1.26-2023.11.14`	`1.26.10`	`1.6.18`	`1.1.2`	Beinhaltet Patches für `CVE-2023-5528` .
`1.26-2023.10.19`	`1.26.9`	`1.6.18`	`1.1.2`	`containerd` wurde auf `1.6.18` aktualisiert. `kubelet` wurde auf `1.26.9` aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (`SERVICE_IPV4_CIDR` und `EXCLUDED_SNAT_CIDRS`).
`1.26-2023.09.12`	`1.26.7`	`1.6.6`	`1.1.2`	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.26-2023.08.17	1.26.7	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.26-2023.08.08	1.26.6	1.6.6	1.1.1	
1.26-2023.07.11	1.26.6	1.6.6	1.1.1	
1.26-2023.06.20	1.26.4	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.26-2023.06.14	1.26.4	1.6.6	1.1.1	Kubernetes wurde auf 1.26.4 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.26-2023.05.09	1.26.2	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.26-2023.04.26	1.26.2	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.26-2023.04.11	1.26.2	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.26-2023.03.24	1.26.2	1.6.6	1.1.1	

Kubernetes version 1.25

Kubernetes-Version **1.25**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2024.05.14	1.25.16	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert.
1.25-2024.04.09	1.25.16	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.25-2024.03.13	1.25.16	1.6.18	1.1.2	
1.25-2024.02.13	1.25.16	1.6.18	1.1.2	
1.25-2024.01.09	1.25.16	1.6.18	1.1.2	

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.25-2023.12.12	1.25.15	1.6.18	1.1.2	
1.25-2023.11.14	1.25.15	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .
1.25-2023.10.19	1.25.14	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.25.14 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.25-2023.09.12	1.25.12	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.25-2023.08.17	1.25.12	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.25-2023.08.08	1.25.9	1.6.6	1.1.1	
1.25-2023.07.11	1.25.9	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2023.06.20	1.25.9	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.
1.25-2023.06.14	1.25.9	1.6.6	1.1.1	Kubernetes wurde auf 1.25.9 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.25-2023.05.09	1.25.7	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.25-2023.04.11	1.25.7	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.25-2023.03.27	1.25.6	1.6.6	1.1.1	Es wurde ein domainloses gMSA-Plugin installiert, um die gMSA-Authentifizierung für Windows-Container in Amazon EKS zu erleichtern.
1.25-2023.03.20	1.25.6	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.25-2023.02.14	1.25.6	1.6.6	1.1.1	

Kubernetes version 1.24

Kubernetes-Version **1.24**

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2024.05.14	1.24.17	1.6.28	1.1.2	containerd wurde auf 1.6.28 aktualisiert.
1.24-2024.04.09	1.24.17	1.6.25	1.1.2	containerd wurde auf 1.6.25 aktualisiert. CNI neu erstellt und csi-proxy verwendet. golang 1.22.1
1.24-2024.03.13	1.24.17	1.6.18	1.1.2	
1.24-2024.02.13	1.24.17	1.6.18	1.1.2	
1.24-2024.01.09	1.24.17	1.6.18	1.1.2	
1.24-2023.12.12	1.24.17	1.6.18	1.1.2	
1.24-2023.11.14	1.24.17	1.6.18	1.1.2	Beinhaltet Patches für CVE-2023-5528 .

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.24-2023.10.19	1.24.17	1.6.18	1.1.2	containerd wurde auf 1.6.18 aktualisiert. kubelet wurde auf 1.24.17 aktualisiert. Neue Umgebungsvariablen für Bootstrap-Skripte hinzugefügt (SERVICE_IPV4_CIDR und EXCLUDED_SNAT_CIDRS).
1.24-2023.09.12	1.24.16	1.6.6	1.1.2	Das Amazon-VPC-CNI-Plugin wurde aktualisiert, sodass es die Kubernetes-Connector-Binärdatei verwendet, die die Pod-IP-Adresse vom Kubernetes-API-Server bezieht. Die Pull-Anforderung #100 wurde zusammengeführt.
1.24-2023.08.17	1.24.16	1.6.6	1.1.2	Beinhaltet Patches für CVE-2023-3676 , CVE-2023-3893 , und CVE-2023-3955 .
1.24-2023.08.08	1.24.13	1.6.6	1.1.1	
1.24-2023.07.11	1.24.13	1.6.6	1.1.1	
1.24-2023.06.21	1.24.13	1.6.6	1.1.1	Das Problem, das dazu führte, dass die DNS-Suffix-Suchliste falsch gefüllt wurde, wurde behoben.

AMI-Version	**kubelet**-Version	**containerd** -Version	**csi-proxy** -Version	Versionshinweise
1.24-2023.06.14	1.24.13	1.6.6	1.1.1	Kubernetes wurde auf 1.24.13 aktualisiert. Die Unterstützung für die Zuordnung von Host-Ports in CNI wurde hinzugefügt. Die Pull-Anforderung #93 wurde zusammengeführt.
1.24-2023.05.09	1.24.7	1.6.6	1.1.1	Ein Fehler wurde behoben, der das Netzwerkverbindungsproblem #1126 auf Pods nach dem Neustart des Knotens verursachte. Ein neuer Bootstrap-Skriptkonfigurationsparameter (ExcludedSnatCIDRs) wurde eingeführt.
1.24-2023.04.11	1.24.7	1.6.6	1.1.1	Wiederherstellungsmechanismus für kubelet und kube-proxy bei einem Serviceabsturz hinzugefügt.
1.24-2023.03.27	1.24.7	1.6.6	1.1.1	Es wurde ein domainloses gMSA-Plugin installiert, um die gMSA-Authentifizierung für Windows-Container in Amazon EKS zu erleichtern.
1.24-2023.03.20	1.24.7	1.6.6	1.1.1	Kubernetes-Version wurde auf 1.24.7 herabgestuft, weil 1.24.10 ein Problem in kube-proxy gemeldet hat.
1.24-2023.02.14	1.24.10	1.6.6	1.1.1	

AMI-Version	**kubelet**-Version	**containerd** - Version	**csi-proxy** - Version	Versionshinweise
1.24-2023.01.23	1.24.7	1.6.6	1.1.1	
1.24-2023.01.11	1.24.7	1.6.6	1.1.1	
1.24-2022.12.14	1.24.7	1.6.6	1.1.1	
1.24-2022.10.12	1.24.7	1.6.6	1.1.1	

Abrufen von Amazon-EKS-optimierten Windows-AMI-IDs

Sie können die Amazon Machine Image (AMI) -ID für Amazon EKS-optimierte AMIs programmgesteuert abrufen, indem Sie die AWS Systems Manager Parameter Store-API abfragen. Mit diesem Parameter müssen Sie Amazon-EKS-optimierte AMI-IDs nicht manuell abrufen. Weitere Informationen zur Systems Manager Parameter Store-API finden Sie unter GetParameter. Das von Ihnen verwendete IAM-Prinzipal muss über die `ssm:GetParameter`-IAM-Berechtigung zum Abrufen der Amazon EKS-optimierten AMI-Metadaten verfügen.

Sie können die Image-ID des aktuellen empfohlenen für Amazon EKS optimierten Windows-AMI mit dem folgenden Befehl abrufen, indem Sie den Sub-Parameter `image_id` verwenden. Sie können *`1.30`* durch jede unterstützte Amazon-EKS-Version und *`region-code`* durch eine von Amazon EKS unterstützte Region ersetzen, für die Sie die AMI-ID benötigen. Ersetzen Sie *`Core`* durch `Full`, um die vollständige AMI-ID von Windows-Server anzuzeigen. Für Kubernetes-Version`1.24` oder höher können Sie *`2019`* durch `2022` ersetzen, um die Windows-Server 2022-AMI-ID anzuzeigen.

```
aws ssm get-parameter --name /aws/service/ami-windows-latest/Windows_Server-2019-
English-Core-EKS_Optimized-1.30/image_id --region region-code --query "Parameter.Value"
 --output text
```

Eine Beispielausgabe sieht wie folgt aus.

```
ami-1234567890abcdef0
```

Erstellen benutzerdefinierter Amazon-EKS-optimierter Windows-AMIs

Sie können EC2 Image Builder verwenden, um benutzerdefinierte Amazon-EKS-optimierte Windows-AMIs mit den folgenden Optionen zu erstellen:

- Verwendung eines für Amazon EKS optimierten Windows-AMI als Basis
- Verwenden der von Amazon verwalteten Build-Komponente

Bei beiden Methoden müssen Sie Ihr eigenes Image-Builder-Rezept erstellen. Weitere Informationen finden Sie unter Erstellen einer neuen Version eines Bildrezeptes im Image-Builder-Benutzerhandbuch.

> **Important**
>
> Die folgenden von Amazon verwalteten Komponenten für `eks` enthalten Patches für `CVE-2023-5528`.
>
> - `1.24.3` und höher
> - `1.25.2` und höher
> - `1.26.2` und höher
> - `1.27.0` und höher
> - `1.28.0` und höher

Verwendung eines für Amazon EKS optimierten Windows-AMI als Basis

Diese Option ist die empfohlene Methode, um Ihre benutzerdefinierten Windows-AMIs zu erstellen. Die von uns bereitgestellten für Amazon EKS optimierten Windows-AMIs werden häufiger aktualisiert als die von Amazon verwaltete Build-Komponente.

1. Starten Sie ein neues Image-Builder-Rezept.
   a. Öffnen Sie die EC2 Image Builder-Konsole unter https://console.aws.amazon.com/imagebuilder.
   b. Wählen Sie im linken Navigationsbereich Image recipes (Image-Rezepte) aus.

c. Wählen Sie Create image recipe (Image-Rezept erstellen) aus.

2. Geben Sie im Abschnitt Recipe details (Rezeptdetails) einen Namen und eine Version ein.
3. Geben Sie die ID des für Amazon EKS optimierten Windows-AMI im Abschnitt Base image (Basis-Image) an.

   a. Wählen Sie Enter custom AMI ID (Benutzerdefinierte AMI-ID eingeben) aus.

   b. Rufen Sie die AMI-ID für die Windows-Betriebssystemversion ab, die Sie benötigen. Weitere Informationen finden Sie unter Abrufen von Amazon-EKS-optimierten Windows-AMI-IDs.

   c. Geben Sie die benutzerdefinierte AMI-ID ein. Wenn die AMI-ID nicht gefunden wird, stellen Sie sicher, dass die AWS-Region für die AMI-ID mit der oben rechts in Ihrer Konsole AWS-Region angezeigten übereinstimmt.
4. (Optional) Um die neuesten Sicherheitsupdates zu erhalten, fügen Sie die `update-windows`-Komponente im Abschnitt Build components – (Build-Komponenten –) hinzu.

   a. Wählen Sie in der Dropdownliste rechts neben dem Suchfeld Find components by name (Komponenten nach Namen suchen) Amazon-managed (Von Amazon verwaltet) aus.

   b. Geben Sie im Suchfeld Find components by name (Komponenten nach Namen suchen) **`update-windows`** ein.

   c. Markieren Sie das Kontrollkästchen des **`update-windows`**-Suchergebnisses. Diese Komponente schließt die neuesten Windows-Patches für das Betriebssystem ein.
5. Füllen Sie die verbleibenden Image-Rezepteingaben mit Ihren erforderlichen Konfigurationen aus. Weitere Informationen finden Sie unter Erstellen einer neuen Version eines Image-Rezeptes (Konsole) im Image-Builder-Benutzerhandbuch.
6. Wählen Sie Create Recipe (Rezept erstellen) aus.
7. Verwenden Sie das neue Image-Rezept in einer neuen oder vorhandenen Image-Pipeline. Sobald Ihre Image-Pipeline erfolgreich ausgeführt wurde, wird Ihr benutzerdefiniertes AMI als Ausgabe-Image aufgeführt und ist einsatzbereit. Weitere Informationen finden Sie unter Erstellen einer Image-Pipeline mit dem Assistenten von EC2 Image Builder.

Verwenden der von Amazon verwalteten Build-Komponente

Wenn die Verwendung eines für Amazon EKS optimierten Windows-AMI als Basis nicht praktikabel ist, können Sie stattdessen die von Amazon verwaltete Build-Komponente verwenden. Diese Option kann hinter den neuesten unterstützten Kubernetes-Versionen zurückbleiben.

1. Starten Sie ein neues Image-Builder-Rezept.

   a. Öffnen Sie die EC2 Image Builder-Konsole unter https://console.aws.amazon.com/imagebuilder.

   b. Wählen Sie im linken Navigationsbereich Image recipes (Image-Rezepte) aus.

   c. Wählen Sie Create image recipe (Image-Rezept erstellen) aus.

2. Geben Sie im Abschnitt Recipe details (Rezeptdetails) einen Namen und eine Version ein.

3. Legen Sie im Abschnitt Base image (Basis-Image) fest, welche Option Sie verwenden werden, um Ihr benutzerdefiniertes AMI zu erstellen:

   - Select managed images (Verwaltete Images auswählen) – Wählen Sie Windows als Ihr Image Operating System (OS) (Image-Betriebssystem (OS)) aus. Wählen Sie dann eine der folgenden Optionen für Image origin (Image-Ursprung) aus:
     - Quick start (Amazon-managed) (Schnellstart (von Amazon verwaltet)) – Wählen Sie in der Dropdown-Liste Image name (Image-Nam) eine von Amazon EKS unterstützte Windows-Server-Version aus. Weitere Informationen finden Sie unter Amazon-EKS-optimierte Windows-AMIs.
     - Images owned by me (Bilder in meinem Besitz) – Wählen Sie für Image-Name den ARN Ihres eigenen Image mit Ihrer eigenen Lizenz aus. Auf dem von Ihnen bereitgestellten Image können Amazon-EKS-Komponenten nicht bereits installiert sein.
   - Enter custom AMI ID (Benutzerdefinierte AMI-ID eingeben) – Geben Sie für AMI-ID die ID für Ihr AMI mit Ihrer eigenen Lizenz ein. Auf dem von Ihnen bereitgestellten Image können Amazon-EKS-Komponenten nicht bereits installiert sein.

4. Gehen Sie im Abschnitt Build components – Windows (Komponenten erstellen – Windows) wie folgt vor:

   a. Wählen Sie in der Dropdownliste rechts neben dem Suchfeld Find components by name (Komponenten nach Namen suchen) Amazon-managed (Von Amazon verwaltet) aus.

   b. Geben Sie im Suchfeld Find components by name (Komponenten nach Namen suchen) **`eks`** ein.

   c. Markieren Sie das Kästchen des Suchergebnisses **`eks-optimized-ami-windows`**, auch wenn das zurückgegebene Ergebnis möglicherweise nicht die gewünschte Version ist.

   d. Geben Sie im Suchfeld Find components by name (Komponenten nach Namen suchen) **`update-windows`** ein.

e. Markieren Sie das Kontrollkästchen des Suchergebnisses update-windows. Diese Komponente schließt die neuesten Windows-Patches für das Betriebssystem ein.

5. Gehen Sie im Abschnitt Selected components (Ausgewählte Komponenten) wie folgt vor:

   a. Wählen Sie Versioning options (Versionierungsoptionen) für **`eks-optimized-ami-windows`** aus.

   b. Wählen Sie Specify component versio (Komponentenversion angeben) aus.

   c. Geben Sie im Feld Component version (Komponentenversion) den Wert ***`version.x`*** ein und ersetzen Sie *`version`* durch eine unterstützte Kubernetes-Version. Wenn Sie *`x`* als einen Teil der Versionsnummer eingeben, wird die neueste Komponentenversion verwendet, die auch mit dem Teil der Version übereinstimmt, den Sie explizit definieren. Achten Sie auf die Konsolenausgabe, da sie Sie darüber informiert, ob Ihre gewünschte Version als verwaltete Komponente verfügbar ist. Beachten Sie, dass die neuesten Kubernetes-Versionen für die Build-Komponente möglicherweise nicht verfügbar sind. Weitere Informationen zu den verfügbaren Versionen finden Sie unter Abrufen von Informationen zu `eks-optimized-ami-windows`-Komponentenversionen.

   > **Note**
   >
   > Für die folgenden Versionen der `eks-optimized-ami-windows`-Build-Komponenten ist eine `eksctl`-Version `0.129` oder eine niedrigere Version erforderlich:
   >
   > - `1.24.0`

6. Füllen Sie die verbleibenden Image-Rezepteingaben mit Ihren erforderlichen Konfigurationen aus. Weitere Informationen finden Sie unter Erstellen einer neuen Version eines Image-Rezeptes (Konsole) im Image-Builder-Benutzerhandbuch.

7. Wählen Sie Create Recipe (Rezept erstellen) aus.

8. Verwenden Sie das neue Image-Rezept in einer neuen oder vorhandenen Image-Pipeline. Sobald Ihre Image-Pipeline erfolgreich ausgeführt wurde, wird Ihr benutzerdefiniertes AMI als Ausgabe-Image aufgeführt und ist einsatzbereit. Weitere Informationen finden Sie unter Erstellen einer Image-Pipeline mit dem Assistenten von EC2 Image Builder.

Abrufen von Informationen zu **eks-optimized-ami-windows**-Komponentenversionen

Sie können spezifische Informationen darüber abrufen, was mit den einzelnen Komponenten installiert ist. Sie können beispielsweise überprüfen, welche `kubelet`-Version installiert ist. Die Komponenten durchlaufen Funktionstests auf den von Amazon EKS unterstützten Windows-Betriebssystemversionen. Weitere Informationen finden Sie unter Veröffentlichungskalender. Alle anderen Windows-Betriebssystemversionen, die nicht aufgelistet sind oder deren Support eingestellt wird, sind möglicherweise nicht mit der Komponente kompatibel.

1. Öffnen Sie die EC2 Image Builder-Konsole unter https://console.aws.amazon.com/imagebuilder.
2. Wählen Sie im linken Navigationsbereich Components (Komponenten) aus.
3. Ändern Sie in der Dropdownliste rechts neben dem Suchfeld Find components by name (Komponenten nach Namen suchen) Owned by me (In meinem Besitz) zu Quick start (Amazon-managed) (Schnellstart (Von Amazon verwaltet)).
4. Geben Sie im Feld Komponenten nach Name suchen **eks** ein.
5. (Optional) Wenn Sie eine aktuelle Version verwenden, sortieren Sie die Spalte Version in absteigender Reihenfolge, indem Sie sie zweimal auswählen.
6. Wählen Sie den **eks-optimized-ami-windows**-Link mit der gewünschten Version aus.

Unter Description (Beschreibung) auf der resultierenden Seite werden die spezifischen Informationen angezeigt.

Speicher

In diesem Kapitel werden Speicheroptionen für Amazon EKS-Cluster behandelt.

Themen

Amazon-EBS-CSI-Treiber

Der Amazon Elastic Block Store (Amazon EBS) Container-Storage-Interface-Treiber (CSI) verwaltet den Lebenszyklus von Amazon-EBS-Volumes als Speicher für von Ihnen erstellte Kubernetes-Volumes. Der Amazon-EBS-CSI-Treiber erstellt Amazon-EBS-Volumes für diese Arten von Kubernetes Volumes: generische flüchtige Volumes und persistente Volumes.

Hier sind einige Dinge, die Sie bei der Verwendung des Amazon-EBS-CSI-Treibers beachten sollten.

- Das Amazon-EBS-CSI-Plugin benötigt IAM-Berechtigungen, um in Ihrem Namen Aufrufe an - AWS APIs zu tätigen. Weitere Informationen finden Sie unter Erstellen der IAM-Rolle des Amazon-EBS-CSI-Treibers.
- Sie können Amazon-EBS-Volumes nicht in Fargate-Pods mounten.
- Sie können den Amazon-EBS-CSI-Controller auf Fargate-Knoten ausführen, aber der Amazon-EBS-CSI-Knoten DaemonSet kann nur auf Amazon-EC2-Instances ausgeführt werden.

Der Amazon-EBS-CSI-Treiber wird beim ersten Erstellen eines Clusters nicht installiert. Um den Treiber verwenden zu können, müssen Sie ihn als Amazon-EKS-Add-on oder als selbstverwaltetes Add-on hinzufügen.

- Anweisungen zum Hinzufügen als Amazon-EKS-Add-on finden Sie unter Verwalten des Amazon-EBS-CSI-Treibers als Amazon-EKS-Add-on.
- Anweisungen zum Hinzufügen als selbstverwaltetes Add-on finden Sie im Projekt Amazon EBS Container Storage Interface (CSI)-Treiber in GitHub.

Nachdem Sie den CSI-Treiber mit einer der beiden Methoden installiert haben, können Sie die Funktionalität mit einer Beispielanwendung testen. Weitere Informationen finden Sie unter Bereitstellen einer Beispielanwendung bereit und Überprüfung, ob der CSI-Treiber funktioniert.

Erstellen der IAM-Rolle des Amazon-EBS-CSI-Treibers

Das Amazon-EBS-CSI-Plugin benötigt IAM-Berechtigungen, um in Ihrem Namen Aufrufe an - AWS APIs zu tätigen. Weitere Informationen finden Sie unter Treiberberechtigung einrichten auf GitHub.

> **Note**
>
> Pods haben Zugriff auf die Berechtigungen, die der IAM-Rolle zugewiesen sind, es sei denn, Sie blockieren den Zugriff auf IMDS. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in Amazon EKS.

Voraussetzungen

- Einen vorhandenen -Cluster.
- Ein vorhandener AWS Identity and Access Management (IAM) OpenID Connect (OIDC)-Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.

Das folgende Verfahren zeigt Ihnen, wie Sie eine IAM-Rolle erstellen und ihr die von AWS verwaltete Richtlinie hinzufügen. Sie können `eksctl`, die AWS Management Console oder die AWS CLI verwenden.

> **Note**
>
> Die spezifischen Schritte in diesem Verfahren sind für die Verwendung des Treibers als Amazon-EKS-Add-on geschrieben. Es sind verschiedene Schritte erforderlich, um den

Treiber als selbstverwaltetes Add-on zu verwenden. Weitere Informationen finden Sie im GitHub unter Treiberberechtigungen einrichten.

eksctl

So erstellen Sie Ihre Amazon-EBS-CSI-Plugin-IAM-Rolle mit **`eksctl`**

1. Erstellen Sie eine IAM-Rolle und fügen Sie eine policy. AWS maintains- AWS verwaltete Richtlinie an. Sie können auch eine eigene benutzerdefinierte Richtlinie erstellen. Sie können eine IAM-Rolle erstellen und die AWS verwaltete Richtlinie mit dem folgenden Befehl anfügen. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters. Der Befehl stellt einen - AWS CloudFormation Stack bereit, der eine IAM-Rolle erstellt und ihr die IAM-Richtlinie anfügt. Wenn sich Ihr Cluster in der AWS GovCloud (USA-Ost) oder AWS GovCloud (USA-West) befindet AWS-Regionen, ersetzen Sie durch `arn:aws:` `arn:aws-us-gov:`.

   ```
   eksctl create iamserviceaccount \
       --name ebs-csi-controller-sa \
       --namespace kube-system \
       --cluster my-cluster \
       --role-name AmazonEKS_EBS_CSI_DriverRole \
       --role-only \
       --attach-policy-arn arn:aws:iam::aws:policy/service-role/
   AmazonEBSCSIDriverPolicy \
       --approve
   ```

2. Wenn Sie einen benutzerdefinierten KMS-Schlüssel für die Verschlüsselung auf Ihren Amazon-EBS-Volumes verwenden, passen Sie die IAM-Rolle nach Bedarf an. Führen Sie beispielsweise folgende Schritte aus:

   a. Kopieren Sie den folgenden Code und fügen Sie diesen in eine neue *`kms-key-for-encryption-on-ebs`*`.json`-Datei ein. Ersetzen Sie *`custom-key-arn`* durch den benutzerdefinierten KMS-Schlüssel-ARN.

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

```
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": ["custom-key-arn"],
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": ["custom-key-arn"]
    }
  ]
}
```

b. Erstellen Sie die Richtlinie. Sie können *KMS_Key_For_Encryption_On_EBS_Policy* in einen anderen Namen ändern. In diesem Fall ändern Sie sie auch in den späteren Schritten.

```
aws iam create-policy \
  --policy-name KMS_Key_For_Encryption_On_EBS_Policy \
  --policy-document file://kms-key-for-encryption-on-ebs.json
```

c. Hängen Sie die IAM-Richtlinie mit dem folgenden Befehl an die Rolle an. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Wenn sich Ihr Cluster in der AWS GovCloud (USA-Ost) oder AWS GovCloud (USA-West) befindet AWS-Regionen, ersetzen Sie durch `arn:aws:` `arn:aws-us-gov:`.

```
aws iam attach-role-policy \
  --policy-arn
 arn:aws:iam::111122223333:policy/KMS_Key_For_Encryption_On_EBS_Policy \
  --role-name AmazonEKS_EBS_CSI_DriverRole
```

AWS Management Console

So erstellen Sie Ihre IAM-Rolle des Amazon-EBS-CSI-Plugins mit der AWS Management Console

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Klicken Sie auf der Seite Roles (Rollen) auf Create role (Rolle erstellen).
4. Gehen Sie auf der Seite Select trusted entity (Vertrauenswürdige Entität auswählen) wie folgt vor:
   a. Wählen Sie im Abschnitt Trusted entity type (Typ der vertrauenswürdigen Entität) die Option Web identity (Web-Identität) aus.
   b. Wählen Sie für Identity provider (Identitätsanbieter) die Option OpenID Connect provider URL (-Anbieter-URL) für Ihren Cluster aus (wie unter Overview (Übersicht) in Amazon EKS gezeigt).
   c. Wählen Sie für Audience (Zielgruppe) `sts.amazonaws.com`.
   d. Wählen Sie Next (Weiter).
5. Gehen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) wie folgt vor:
   a. Geben Sie im Feld Filter policies (Filterrichtlinien) `AmazonEBSCSIDriverPolicy` ein.
   b. Aktivieren Sie das Kontrollkästchen links neben der `AmazonEBSCSIDriverPolicy`, die bei der Suche zurückgegeben wurde.
   c. Wählen Sie Next (Weiter).
6. Gehen Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) wie folgt vor:
   a. Geben Sie unter Role name (Rollenname) einen eindeutigen Namen für die Rolle ein, z. B. ***`AmazonEKS_EBS_CSI_DriverRole`***.
   b. Fügen Sie der Rolle unter Tags hinzufügen (optional) Metadaten hinzu, indem Sie Tags als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.
   c. Wählen Sie Create role (Rolle erstellen) aus.
7. Nachdem die Rolle erstellt wurde, wählen Sie die Rolle in der Konsole aus, um sie zur Bearbeitung zu öffnen.

8. Wählen Sie die Registerkarte Trust Relationships (Vertrauensstellungen) und dann Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.
9. Suchen Sie die Zeile, die der folgenden Zeile ähnelt:

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud":
 "sts.amazonaws.com"
```

Fügen Sie am Ende der vorherigen Zeile ein Komma hinzu und fügen Sie anschließend die folgende Zeile nach der vorherigen Zeile hinzu. Ersetzen Sie durch *region-code* die AWS-Region , in der sich Ihr Cluster befindet. Ersetzen von *EXAMPLED539D4633E53DE1B71EXAMPLE* mit der OIDC-Anbieter-ID Ihres Clusters.

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub":
 "system:serviceaccount:kube-system:ebs-csi-controller-sa"
```

10. Wählen Sie Update Policy (Richtlinie aktualisieren) aus, um den Vorgang abzuschließen.
11. Wenn Sie einen benutzerdefinierten KMS-Schlüssel für die Verschlüsselung auf Ihren Amazon-EBS-Volumes verwenden, passen Sie die IAM-Rolle nach Bedarf an. Führen Sie beispielsweise folgende Schritte aus:

    a. Wählen Sie im linken Navigationsbereich die Option Policies aus.
    b. Wählen Sie auf der Seite Policies (Richtlinien) die Option Create a policy (Richtlinie erstellen).
    c. Wählen Sie auf der Seite Create policy (Richtlinie erstellen) die Registerkarte JSON aus.
    d. Kopieren Sie den folgenden Code in den Editor und ersetzen Sie *custom-key-arn* durch den benutzerdefinierten KMS-Schlüssel-ARN:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": ["custom-key-arn"],
      "Condition": {
```

```
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": ["custom-key-arn"]
    }
  ]
}
```

e. Wählen Sie Next: Markierungen (Weiter: Markierungen).

f. Wählen Sie auf der Seite Add Tags (optional) (Tags hinzufügen (optional)) die Option Next: Review (Weiter: Prüfen).

g. Geben Sie unter Name einen eindeutigen Namen für Ihre Richtlinie ein (z. B. ***KMS_Key_For_Encryption_On_EBS_Policy***).

h. Wählen Sie Create Policy (Richtlinie erstellen) aus.

i. Wählen Sie im linken Navigationsbereich Roles aus.

j. Wählen Sie ***AmazonEKS _EBS_CSI_DriverRole*** in der Konsole aus, um es zur Bearbeitung zu öffnen.

k. Wählen Sie in der Dropdown-Liste Add permissions (Berechtigungen hinzufügen) Attach policies (Richtlinien hinzufügen) aus.

l. Geben Sie im Feld Filter policies (Filterrichtlinien) *KMS_Key_For_Encryption_On_EBS_Policy* ein.

m. Aktivieren Sie das Kontrollkästchen links neben der *KMS_Key_For_Encryption_On_EBS_Policy*, die bei der Suche zurückgegeben wurde.

n. Wählen Sie Attach Policies (Richtlinien hinzufügen).

AWS CLI

So erstellen Sie Ihre IAM-Rolle des Amazon-EBS-CSI-Plugins mit der AWS CLI

1. Zeigen Sie die OIDC-Anbieter-URL Ihres Clusters an. Ersetzen Sie *my-cluster* durch Ihren Clusternamen. Wenn die Ausgabe des Befehls `None` ist, überprüfen Sie die Voraussetzungen.

   ```
   aws eks describe-cluster --name my-cluster --query
    "cluster.identity.oidc.issuer" --output text
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
   ```

2. Erstellen Sie die IAM-Rolle und gewähren Sie ihr die `AssumeRoleWithWebIdentity`-Aktion.

   a. Kopieren Sie den folgenden Inhalt in eine Datei namens *aws-ebs-csi-driver-trust-policy*`.json`. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Ersetzen Sie *EXAMPLED539D4633E53DE1B71EXAMPLE* und *region-code* mit den im vorherigen Schritt zurückgegebenen Werten. Wenn sich Ihr Cluster in der AWS GovCloud (USA-Ost) oder AWS GovCloud (USA-West) befindet AWS-Regionen, ersetzen Sie durch `arn:aws:` `arn:aws-us-gov:`.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Federated": "arn:aws:iam::111122223333:oidc-provider/
   oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
         },
         "Action": "sts:AssumeRoleWithWebIdentity",
         "Condition": {
           "StringEquals": {
             "oidc.eks.region-code.amazonaws.com/
   id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com",
   ```

```
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:ebs-csi-controller-sa"
        }
      }
    }
  ]
}
```

b. Erstellen Sie die -Rolle. Sie können *AmazonEKS_EBS_CSI_DriverRole* in einen anderen Namen ändern. Wenn Sie es ändern, ändern Sie es in späteren Schritten.

```
aws iam create-role \
  --role-name AmazonEKS_EBS_CSI_DriverRole \
  --assume-role-policy-document file://"aws-ebs-csi-driver-trust-
policy.json"
```

3. Fügen Sie eine policy. AWS maintains-Richtlinie an AWS oder Sie können Ihre eigene benutzerdefinierte Richtlinie erstellen. Hängen Sie die AWS verwaltete Richtlinie mit dem folgenden Befehl an die Rolle an. Wenn sich Ihr Cluster in der AWS GovCloud (USA-Ost) oder AWS GovCloud (USA-West) befindet AWS-Regionen, ersetzen Sie durch `arn:aws:` `arn:aws-us-gov:`.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
  --role-name AmazonEKS_EBS_CSI_DriverRole
```

4. Wenn Sie einen benutzerdefinierten KMS-Schlüssel für die Verschlüsselung auf Ihren Amazon-EBS-Volumes verwenden, passen Sie die IAM-Rolle nach Bedarf an. Führen Sie beispielsweise folgende Schritte aus:

a. Kopieren Sie den folgenden Code und fügen Sie diesen in eine neue *kms-key-for-encryption-on-ebs*`.json`-Datei ein. Ersetzen Sie *custom-key-arn* durch den benutzerdefinierten KMS-Schlüssel-ARN.

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

```
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": ["custom-key-arn"],
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": ["custom-key-arn"]
    }
  ]
}
```

b. Erstellen Sie die Richtlinie. Sie können *KMS_Key_For_Encryption_On_EBS_Policy* in einen anderen Namen ändern. In diesem Fall ändern Sie sie auch in den späteren Schritten.

```
aws iam create-policy \
  --policy-name KMS_Key_For_Encryption_On_EBS_Policy \
  --policy-document file://kms-key-for-encryption-on-ebs.json
```

c. Hängen Sie die IAM-Richtlinie mit dem folgenden Befehl an die Rolle an. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Wenn sich Ihr Cluster in der AWS GovCloud (USA-Ost) oder AWS GovCloud (USA-West) befindet AWS-Regionen, ersetzen Sie durch `arn:aws:` `arn:aws-us-gov:`.

```
aws iam attach-role-policy \
  --policy-arn
 arn:aws:iam::111122223333:policy/KMS_Key_For_Encryption_On_EBS_Policy \
  --role-name AmazonEKS_EBS_CSI_DriverRole
```

Nachdem Sie die IAM-Rolle des Amazon-EBS-CSI-Treibers erstellt haben, können Sie mit Hinzufügen des Amazon-EBS-CSI-Add-ons fortfahren. Wenn Sie das Plugin in diesem Verfahren bereitstellen, wird es erstellt und für die Verwendung eines Servicekontos mit dem Namen `ebs-csi-controller-sa` konfiguriert. Das Servicekonto ist an eine Kubernetes `clusterrole` gebunden, der die erforderlichen Kubernetes-Berechtigungen zugewiesen sind.

Verwalten des Amazon-EBS-CSI-Treibers als Amazon-EKS-Add-on

Um die Sicherheit zu verbessern und den Arbeitsaufwand zu reduzieren, können Sie den Amazon-EBS-CSI-Treiber als Amazon-EKS-Add-on verwalten. Informationen zu Amazon-EKS-Add-ons finden Sie unter Amazon-EKS-Add-ons. Sie können das Amazon-EBS-CSI-Add-on hinzufügen, indem Sie die Schritte in Hinzufügen des Amazon-EBS-CSI-Add-ons ausführen.

Wenn Sie das Amazon-EBS-CSI-Add-on hinzugefügt haben, können Sie es verwalten, indem Sie die Schritte in den Abschnitten Aktualisieren des Amazon-EBS-CSI-Treibers als Amazon-EKS-Add-on und Entfernen des Amazon-EBS-CSI-Add-ons ausführen.

Voraussetzungen

- Einen vorhandenen -Cluster. Führen Sie den folgenden Befehl aus, um die erforderliche Plattformversion anzuzeigen.

  ```
  aws eks describe-addon-versions --addon-name aws-ebs-csi-driver
  ```

- Ein vorhandener AWS Identity and Access Management-IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Eine Amazon-EBS-CSI-Treiber-IAM-Rolle. Wenn Sie diese Voraussetzung nicht erfüllen, wird beim Versuch, das Add-On zu installieren und `kubectl describe pvc` auszuführen, `failed to provision volume with StorageClass` zusammen mit einem `could not create volume in EC2: UnauthorizedOperation`-Fehler angezeigt. Weitere Informationen finden Sie unter Erstellen der IAM-Rolle des Amazon-EBS-CSI-Treibers.
- Wenn Sie ein clusterweites eingeschränktes `PodSecurityPolicy` verwenden, stellen Sie sicher, dass das Add-On über ausreichende Berechtigungen für die Bereitstellung verfügt. Die für jedes Add-On erforderlichen Berechtigungen finden Sie Podin der entsprechenden Add-On-Manifestdefinition auf GitHub.

Important

Um die Snapshot-Funktionalität des Amazon-EBS-CSI-Treibers nutzen zu können, müssen Sie den externen Snapshotter vor der Installation des Add-ons installieren. Die Komponenten des externen Snapshotters müssen in der folgenden Reihenfolge installiert werden:

- `CustomResourceDefinition` (CRD) für `volumesnapshotclasses`, `volumesnapshots` und `volumesnapshotcontents`
- RBAC (`ClusterRole`, `ClusterRoleBinding` usw.)
- Controller-Bereitstellung

Weitere Informationen finden Sie unter CSI Snapshotter auf GitHub.

Hinzufügen des Amazon-EBS-CSI-Add-ons

Important

Bevor Sie den Amazon-EBS-Treiber als Amazon-EKS-Add-on hinzufügen, stellen Sie sicher, dass Sie keine selbstverwaltete Version des Treibers auf Ihrem Cluster installiert haben. Falls ja, siehe Deinstallation eines selbstverwalteten Amazon-EBS-CSI-Treibers in GitHub.

Sie können `eksctl`, die AWS Management Console oder die AWS CLI verwenden, um das Amazon-EBS-CSI-Add-on zu Ihrem Cluster hinzuzufügen.

eksctl

So fügen Sie das Amazon-EBS-CSI-Add-on mit hinzu **`eksctl`**

Führen Sie den folgenden Befehl aus. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters, *`111122223333`* mit Ihrer Konto-ID und *`AmazonEKS_EBS_CSI_DriverRole`* mit dem Namen der zuvor erstellten IAM-Rolle. Wenn sich Ihr Cluster in der AWS GovCloud (USA-Ost) oder AWS GovCloud (USA-West) befindetAWS-Regionen, ersetzen Sie durch `arn:aws:` `arn:aws-us-gov:`.

```
eksctl create addon --name aws-ebs-csi-driver --cluster my-cluster --service-account-role-arn arn:aws:iam::111122223333:role/AmazonEKS_EBS_CSI_DriverRole --force
```

Wenn Sie die Option `--force` entfernen und eine der Amazon-EKS-Add-on-Einstellungen mit Ihren vorhandenen Einstellungen in Konflikt steht, schlägt die Aktualisierung des Amazon-EKS-Add-ons fehl und Sie erhalten eine Fehlermeldung, die Sie bei der Lösung des Konflikts unterstützt. Stellen Sie vor dem Angeben dieser Option sicher, dass das Amazon-EKS-Add-on keine Einstellungen verwaltet, die Sie verwalten müssen, da diese Einstellungen mit dieser Option überschrieben werden. Weitere Informationen zu anderen Optionen für diese Einstellung finden Sie unter Add-Ons in der `eksctl`-Dokumentation. Weitere Informationen zur Amazon-EKS-Kubernetes-Feldverwaltung finden Sie unter Kubernetes-Feldverwaltung.

AWS Management Console

So fügen Sie das Amazon-EBS-CSI-Add-on mit hinzu AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie das Amazon-EBS-CSI-Add-on konfigurieren möchten.
4. Wählen Sie die Registerkarte Add-ons.
5. Wählen Sie Weitere Add-Ons erhalten.
6. Führen Sie auf der Seite Add-Ons auswählen die folgenden Schritte aus:
   a. Aktivieren Sie im Abschnitt Amazon-EKS-Add-Ons das Kontrollkästchen Amazon-EBS-CSI-Treiber.
   b. Wählen Sie Weiter aus.
7. Gehen Sie auf der Seite Konfigurieren ausgewählter Add-Ons-Einstellungen wie folgt vor:
   a. Wählen Sie die Version aus, die Sie verwenden möchten.
   b. Wählen Sie für IAM-Rolle auswählen den Namen einer IAM-Rolle aus, an die Sie die IAM-Richtlinie „Amazon-EBS-CSI-Treiber“ angehängt haben.
   c. (Optional) Sie können Optionale Konfigurationseinstellungen erweitern. Wenn Sie Überschreiben für Methode zur Konfliktlösung auswählen, können eine oder mehrere der Einstellungen für das vorhandene Add-on mit den Einstellungen des Amazon-EKS-Add-ons überschrieben werden. Wenn Sie diese Option nicht aktivieren und ein Konflikt mit Ihren vorhandenen Einstellungen vorliegt, schlägt der Vorgang fehl. Sie können die sich daraus ergebende Fehlermeldung heranziehen, um den Konflikt zu beheben.

Stellen Sie vor der Auswahl dieser Option sicher, dass das Amazon-EKS-Add-on keine Einstellungen verwaltet, die Sie selbst verwalten müssen.

d. Wählen Sie Weiter aus.

8. Wählen Sie auf der Seite Überprüfen und hinzufügen die Option Erstellen aus. Nachdem die Installation der Add-Ons abgeschlossen ist, wird Ihr installiertes Add-On angezeigt.

AWS CLI

So fügen Sie das Amazon-EBS-CSI-Add-on mit hinzu AWS CLI

Führen Sie den folgenden Befehl aus. Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters, *111122223333* mit Ihrer Konto-ID und *AmazonEKS_EBS_CSI_DriverRole* mit dem Namen der zuvor erstellten Rolle. Wenn sich Ihr Cluster in der AWS GovCloud (USA-Ost) oder AWS GovCloud (USA-West) befindetAWS-Regionen, ersetzen Sie durch `arn:aws:` `arn:aws-us-gov:`.

```
aws eks create-addon --cluster-name my-cluster --addon-name aws-ebs-csi-driver \
  --service-account-role-arn
 arn:aws:iam::111122223333:role/AmazonEKS_EBS_CSI_DriverRole
```

Nachdem Sie den CSI-Treiber von Amazon EBS als Amazon-EKS-Add-on hinzugefügt haben, können Sie mit Bereitstellen einer Beispielanwendung bereit und Überprüfung, ob der CSI-Treiber funktioniert fortfahren. Dieses Verfahren beinhaltet das Einrichten der Speicherklasse.

Aktualisieren des Amazon-EBS-CSI-Treibers als Amazon-EKS-Add-on

Amazon EKS aktualisiert Amazon EBS CSI für Ihren Cluster nicht automatisch, wenn neue Versionen veröffentlicht werden oder nachdem Sie Ihren Cluster auf eine neue Kubernetes-Nebenversion aktualisiert haben. Um Amazon EBS CSI auf einem vorhandenen Cluster zu aktualisieren, müssen Sie das Update initiieren und Amazon EKS aktualisiert dann das Add-on für Sie.

eksctl

So aktualisieren Sie das Amazon-EBS-CSI-Add-on mit **`eksctl`**

1. Überprüfen Sie die aktuelle Version Ihres Amazon-EBS-CSI-Add-ons. Ersetzen Sie *my-cluster* mit Ihrem Clusternamen.

```
eksctl get addon --name aws-ebs-csi-driver --cluster my-cluster
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                    VERSION                 STATUS  ISSUES  IAMROLE
 UPDATE AVAILABLE
aws-ebs-csi-driver      v1.11.2-eksbuild.1      ACTIVE  0
  v1.11.4-eksbuild.1
```

2. Aktualisieren Sie das Add-on auf die unter `UPDATE AVAILABLE` in der Ausgabe des vorherigen Schritts zurückgegebene Version.

```
eksctl update addon --name aws-ebs-csi-driver --version v1.11.4-eksbuild.1 --
cluster my-cluster \
  --service-account-role-arn
 arn:aws:iam::111122223333:role/AmazonEKS_EBS_CSI_DriverRole --force
```

Wenn Sie die Option **`--force`** entfernen und eine der Amazon-EKS-Add-on-Einstellungen mit Ihren vorhandenen Einstellungen in Konflikt steht, schlägt die Aktualisierung des Amazon-EKS-Add-ons fehl und Sie erhalten eine Fehlermeldung, die Sie bei der Lösung des Konflikts unterstützt. Stellen Sie vor dem Angeben dieser Option sicher, dass das Amazon-EKS-Add-on keine Einstellungen verwaltet, die Sie verwalten müssen, da diese Einstellungen mit dieser Option überschrieben werden. Weitere Informationen zu anderen Optionen für diese Einstellung finden Sie unter Add-Ons in der `eksctl`-Dokumentation. Weitere Informationen zur Amazon-EKS-Kubernetes-Feldverwaltung finden Sie unter Kubernetes-Feldverwaltung.

AWS Management Console

So aktualisieren Sie das Amazon-EBS-CSI-Add-on mit dem AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie das Amazon-EBS-CSI-Add-on aktualisieren möchten.
4. Wählen Sie die Registerkarte Add-ons.
5. Wählen Sie Amazon-EBS-CSI-Treiber.

6. Wählen Sie Bearbeiten aus.
7. Gehen Sie auf der Seite Amazon-EBS-CSI-Treiber konfigurieren wie folgt vor:
   a. Wählen Sie die Version aus, die Sie verwenden möchten.
   b. Wählen Sie für IAM-Rolle auswählen den Namen einer IAM-Rolle aus, an die Sie die IAM-Richtlinie „Amazon-EBS-CSI-Treiber“ angehängt haben.
   c. (Optional) Sie können Optionale Konfigurationseinstellungen erweitern und nach Bedarf Änderungen vornehmen.
   d. Wählen Sie Änderungen speichern aus.

AWS CLI

So aktualisieren Sie das Amazon-EBS-CSI-Add-on mit dem AWS CLI

1. Überprüfen Sie die aktuelle Version Ihres Amazon-EBS-CSI-Add-ons. Ersetzen Sie *my-cluster* mit Ihrem Clusternamen.

```
aws eks describe-addon --cluster-name my-cluster --addon-name aws-ebs-csi-driver
 --query "addon.addonVersion" --output text
```

   Eine Beispielausgabe sieht wie folgt aus.

```
v1.11.2-eksbuild.1
```

2. Bestimmen Sie, welche Versionen des Amazon-EBS-CSI-Add-ons für die Version Ihres Clusters verfügbar sind.

```
aws eks describe-addon-versions --addon-name aws-ebs-csi-driver --kubernetes-
version 1.23 \
  --query "addons[].addonVersions[].[addonVersion,
 compatibilities[].defaultVersion]" --output text
```

   Eine Beispielausgabe sieht wie folgt aus.

```
v1.11.4-eksbuild.1
True
v1.11.2-eksbuild.1
False
```

Die Version mit `True` darunter ist die Standardversion, die beim Erstellen des Add-Ons bereitgestellt wird. Die Version, die bei der Erstellung des Add-Ons bereitgestellt wird, ist möglicherweise nicht die neueste verfügbare Version. In der vorherigen Ausgabe ist eine neuere Version als die beim Erstellen des Add-Ons bereitgestellte Version verfügbar.

3. Aktualisieren Sie das Add-on auf die Version mit `True`, die in der Ausgabe des vorherigen Schritts zurückgegeben wurde. Wenn sie in der Ausgabe zurückgegeben wurde, können Sie auch auf eine neuere Version aktualisieren.

```
aws eks update-addon --cluster-name my-cluster --addon-name aws-ebs-csi-driver
 --addon-version v1.11.4-eksbuild.1 \
  --service-account-role-arn
 arn:aws:iam::111122223333:role/AmazonEKS_EBS_CSI_DriverRole --resolve-
conflicts PRESERVE
```

Die Option *PRESERVE* behält alle benutzerdefinierten Einstellungen bei, die Sie für das Add-On festgelegt haben. Weitere Informationen zu anderen Optionen für diese Einstellung finden Sie unter update-addon unter Amazon EKS Command Line Reference (Amazon-EKS-Befehlszeilenreferenz). Weitere Informationen zur Amazon EKS-Add-on-Konfigurationsverwaltung finden Sie unter Kubernetes-Feldverwaltung.

Entfernen des Amazon-EBS-CSI-Add-ons

Beim Entfernen eines Amazon-EKS-Add-ons stehen Ihnen zwei Optionen zur Verfügung.

- Beibehalten von Add-on-Software auf Ihrem Cluster - Diese Option entfernt die Amazon-EKS-Verwaltung aller Einstellungen. Amazon EKS wird außerdem die Möglichkeit aufgehoben, Sie über Updates zu informieren und das Amazon-EKS-Add-on automatisch zu aktualisieren, nachdem Sie ein Update eingeleitet haben. Es behält jedoch die Add-on-Software auf Ihrem Cluster bei. Diese Option macht das Add-on zu einem selbstverwalteten Add-on statt einem Amazon-EKS-Add-on. Bei dieser Option gibt es keine Ausfallzeiten für das Add-on. Die Befehle in diesem Verfahren verwenden diese Option.
- Entfernen Sie die Add-on-Software vollständig aus Ihrem Cluster – Wir empfehlen, dass Sie das Amazon-EKS-Add-on nur aus Ihrem Cluster entfernen, wenn auf Ihrem Cluster keine Ressourcen vorhanden sind, die davon abhängig sind. Um diese Option durchzuführen, löschen Sie `--preserve` aus dem Befehl, den Sie für dieses Verfahren verwenden.

Wenn dem Add-on ein IAM-Konto zugeordnet ist, wird das IAM-Konto nicht entfernt.

Sie können `eksctl`, die AWS Management Console oder die AWS CLI verwenden, um das Amazon-EBS-CSI-Add-on zu entfernen.

eksctl

So entfernen Sie das Amazon-EBS-CSI-Add-on mit **`eksctl`**

Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und führen Sie dann den folgenden Befehl aus.

```
eksctl delete addon --cluster my-cluster --name aws-ebs-csi-driver --preserve
```

AWS Management Console

So entfernen Sie das Amazon-EBS-CSI-Add-on mit AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie das Amazon-EBS-CSI-Add-on entfernen möchten.
4. Wählen Sie die Registerkarte Add-ons.
5. Wählen Sie Amazon-EBS-CSI-Treiber.
6. Wählen Sie Remove (Entfernen) aus.
7. Gehen Sie im Bestätigungsdialogfeld Entfernen: aws-ebs-csi-driver wie folgt vor:
   a. Wenn Amazon EKS die Verwaltung von Einstellungen für das Add-on einstellen soll, wählen Sie Auf dem Cluster beibehalten. Tun Sie dies, wenn Sie die Add-on-Software auf Ihrem Cluster behalten möchten. Auf diese Weise können Sie alle Einstellungen des Add-ons selbst verwalten.
   b. Geben Sie **`aws-ebs-csi-driver`** ein.
   c. Wählen Sie Entfernen aus.

AWS CLI

So entfernen Sie das Amazon-EBS-CSI-Add-on mit AWS CLI

Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters und führen Sie dann den folgenden Befehl aus.

```
aws eks delete-addon --cluster-name my-cluster --addon-name aws-ebs-csi-driver --
preserve
```

Bereitstellen einer Beispielanwendung bereit und Überprüfung, ob der CSI-Treiber funktioniert

Sie können die CSI-Treiberfunktionalität mit einer Beispielanwendung testen. Dieses Thema zeigt ein Beispiel, Sie können jedoch auch Folgendes durchführen:

- Stellen Sie eine Beispielanwendung bereit, die Volume-Snapshots mit dem externen Snapshotter verwendet. Weitere Informationen finden Sie unter Volume Snapshots (Volume-Snapshots) auf GitHub.
- Stellen Sie eine Beispielanwendung bereit, die die Volume-Größenänderung verwendet. Weitere Informationen finden Sie unter Volume Resizing (Volume-Größenänderung) auf GitHub.

In diesem Verfahren wird das Beispiel Dynamische Volume-Bereitstellung aus dem GitHub-Repository Amazon-EBS-Container-Storage-Interface-Treiber (CSI) verwendet, um ein dynamisch bereitgestelltes Amazon-EBS-Volume zu verwenden.

1. Klonen Sie das GitHub-Repository Amazon-EBS-Container-Storage-Interface-Treiber (CSI) auf Ihr lokales System.

   ```
   git clone https://github.com/kubernetes-sigs/aws-ebs-csi-driver.git
   ```

2. Navigieren Sie zum `dynamic-provisioning`-Beispielverzeichnis.

   ```
   cd aws-ebs-csi-driver/examples/kubernetes/dynamic-provisioning/
   ```

3. (Optional) Die Datei `manifests/storageclass.yaml` stellt standardmäßig gp2-Volumes von Amazon EBS bereit. Um stattdessen gp3-Volumes zu verwenden, fügen Sie `type: gp3` zu `manifests/storageclass.yaml` hinzu.

   ```
   echo "parameters:
     type: gp3" >> manifests/storageclass.yaml
   ```

4. Stellen Sie die Speicherklasse `ebs-sc`, den persistenten Volume-Anspruch `ebs-claim` und die Beispielanwendung `app` aus dem `manifests`-Verzeichnis bereit.

```
kubectl apply -f manifests/
```

5. Beschreiben Sie die `ebs-sc`-Speicherklasse.

```
kubectl describe storageclass ebs-sc
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:            ebs-sc
IsDefaultClass:  No
Annotations:     kubectl.kubernetes.io/last-applied-
configuration={"apiVersion":"storage.k8s.io/v1","kind":"StorageClass","metadata":
{"annotations":{},"name":"ebs-
sc"},"provisioner":"ebs.csi.aws.com","volumeBindingMode":"WaitForFirstConsumer"}

Provisioner:           ebs.csi.aws.com
Parameters:            <none>
AllowVolumeExpansion:  <unset>
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     WaitForFirstConsumer
Events:                <none>
```

> **Note**
>
> Die Speicherklasse verwendet den `WaitForFirstConsumer`-Volume-Bindungsmodus. Das bedeutet, dass Volumes erst dann dynamisch bereitgestellt werden, wenn ein Pod einen persistenten Volume-Anspruch stellt. Weitere Informationen finden Sie unter Volume-Bindungsmodus in der Kubernetes-Dokumentation.

6. Beobachten Sie die Pods im Standard-Namespace. Nach einigen Minuten ändert sich der Status des app-Pod in `Running`.

```
kubectl get pods --watch
```

Geben Sie `Ctrl+C` ein, um zu einer Shell-Eingabeaufforderung zurückzukehren.

7. Listen Sie die persistenten Volumes im Standard-Namespace auf. Suchen Sie nach einem persistenten Volume mit dem `default/ebs-claim`-Anspruch.

```
kubectl get pv
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY
   STATUS   CLAIM               STORAGECLASS   REASON   AGE
pvc-37717cd6-d0dc-11e9-b17f-06fad4858a5a   4Gi        RWO            Delete
   Bound    default/ebs-claim   ebs-sc                  30s
```

8. Beschreiben Sie das persistente Volume. Ersetzen Sie `pvc-37717cd6-d0dc-11e9-b17f-06fad4858a5a` durch den Wert aus der Ausgabe im vorherigen Schritt.

```
kubectl describe pv pvc-37717cd6-d0dc-11e9-b17f-06fad4858a5a
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:              pvc-37717cd6-d0dc-11e9-b17f-06fad4858a5a
Labels:            <none>
Annotations:       pv.kubernetes.io/provisioned-by: ebs.csi.aws.com
Finalizers:        [kubernetes.io/pv-protection external-attacher/ebs-csi-aws-com]
StorageClass:      ebs-sc
Status:            Bound
Claim:             default/ebs-claim
Reclaim Policy:    Delete
Access Modes:      RWO
VolumeMode:        Filesystem
Capacity:          4Gi
Node Affinity:
  Required Terms:
    Term 0:        topology.ebs.csi.aws.com/zone in [region-code]
Message:
Source:
    Type:              CSI (a Container Storage Interface (CSI) volume source)
    Driver:            ebs.csi.aws.com
    VolumeHandle:      vol-0d651e157c6d93445
    ReadOnly:          false
    VolumeAttributes:      storage.kubernetes.io/
csiProvisionerIdentity=1567792483192-8081-ebs.csi.aws.com
```

```
Events:                <none>
```

Die Amazon EBS-Volume-ID ist der Wert für `VolumeHandle` in der vorherigen Ausgabe.

9. Überprüfen Sie, ob der Pod Daten auf das Volume schreibt.

```
kubectl exec -it app -- cat /data/out.txt
```

Eine Beispielausgabe sieht wie folgt aus.

```
Wed May 5 16:17:03 UTC 2021
Wed May 5 16:17:08 UTC 2021
Wed May 5 16:17:13 UTC 2021
Wed May 5 16:17:18 UTC 2021
[...]
```

10. Wenn Sie fertig sind, löschen Sie die Ressourcen für diese Beispielanwendung.

```
kubectl delete -f manifests/
```

Häufig gestellte Fragen zur Migration des CSI von Amazon EBS

> **Important**
>
> Wenn bei Ihnen Pods in einem Cluster der Version `1.22` oder niedriger ausgeführt werden, müssen Sie zur Vermeidung einer Serviceunterbrechung die Amazon-EBS-CSI-Treiber installieren, bevor Sie Ihren Cluster auf die Version `1.23` aktualisieren.

Das Amazon-EBS-Migrationsfeature für Container Storage Interface (CSI) verlagert die Verantwortung für die Verwaltung von Speichervorgängen vom Amazon-EBS-internen EBS-Speicher-Provisioner auf den Amazon-EBS-CSI-Treiber.

Was sind CSI-Treiber?

CSI-Treiber:

- ersetzen die „In-Tree“-Speichertreiber von Kubernetes, die im Kubernetes-Projektquellcode vorhanden sind;

- arbeiten mit Speicheranbietern wie Amazon EBS zusammen;
- bieten ein vereinfachtes Plugin-Modell, das Speicheranbietern wie AWS die Veröffentlichung von Features und die Aufrechterhaltung des Supports erleichtert, ohne vom Veröffentlichungszyklus von Kubernetes anhängig zu sein.

Weitere Informationen finden Sie unter Introduction (Einführung) in der Dokumentation zum Kubernetes-CSI.

Was ist CSI-Migration?

Das Feature zur Migration des Kubernetes-CSI überträgt die Verantwortung für Speichervorgänge von den vorhandenen In-Tree-Speicher-Plugins wie `kubernetes.io/aws-ebs` auf die entsprechenden CSI-Treiber. Vorhandene Objekte vom Typ `StorageClass`, `PersistentVolume` und `PersistentVolumeClaim` (PVC) funktionieren weiterhin, solange der entsprechende CSI-Treiber installiert ist. Wenn das Feature aktiviert ist:

- funktionieren vorhandene Workloads, die PVCs verwenden, weiterhin wie bisher;
- übergibt Kubernetes die Kontrolle über alle Speicherverwaltungsvorgänge an CSI-Treiber.

Weitere Informationen finden Sie unter Kubernetes`1.23`: Kubernetes-Status-Update für die In-Tree-zu-CSI-Volume-Migration im Kubernetes-Blog.

Um Ihnen bei der Migration vom In-Tree-Plugin zu CSI-Treibern zu helfen, sind die Flags `CSIMigration` und `CSIMigrationAWS` in Clustern der Amazon-EKS-Version `1.23` und höher standardmäßig aktiviert. Mithilfe dieser Flags kann der Cluster die In-Tree-APIs in ihre entsprechenden CSI-APIs übersetzen. Diese Flags werden auf der von Amazon EKS verwalteten Kubernetes-Steuerebene und in den `kubelet`-Einstellungen gesetzt, die in den für Amazon EKS optimierten AMIs konfiguriert sind. Wenn Sie in Ihrem Cluster Pods Amazon-EBS-Volumes verwenden, müssen Sie den CSI-Treiber von Amazon EBS installieren, bevor Sie den Cluster auf die Version **`1.23`** aktualisieren. Andernfalls funktionieren Volume-Vorgänge wie Provisioning und Mounting möglicherweise nicht wie erwartet. Weitere Informationen finden Sie unter Amazon-EBS-CSI-Treiber.

> **Note**
>
> Der In-Tree-`StorageClass`-Provisioner heißt `kubernetes.io/aws-ebs`. Der CSI-`StorageClass`-Provisioner von Amazon EBS heißt `ebs.csi.aws.com`.

Kann ich **`kubernetes.io/aws-ebs StorageClass`**-Volumes in Clustern der Version **`1.23`** und höher mounten?

Ja, solange der CSI-Treiber von Amazon EBS installiert ist. Für neu erstellte Cluster der Version `1.23` und höher empfiehlt es sich, die CSI-Treiber von Amazon EBS beim Erstellen der Cluster zu installieren. Außerdem sollten Sie `StorageClasses` nur basierend auf dem `ebs.csi.aws.com`-Provisioner verwenden.

Wenn Sie Ihre Cluster-Steuerebene auf Version `1.23` aktualisiert, Ihre Knoten aber noch nicht auf `1.23` aktualisiert haben, dann sind die `kubelet`-Flags `CSIMigration` und`CSIMigrationAWS` nicht aktiviert. In diesem Fall wird zum Mounten `kubernetes.io/aws-ebs`-basierter Volumes der In-Tree-Treiber verwendet. Der CSI-Treiber von Amazon EBS muss jedoch trotzdem installiert sein, damit Pods, die `kubernetes.io/aws-ebs`-basierte Volumes nutzen, geplant werden können. Der Treiber ist auch erforderlich, damit andere Volume-Vorgänge gelingen.

Kann ich **`kubernetes.io/aws-ebs StorageClass`**-Volumes in Clustern von Amazon EKS **`1.23`** und höher bereitstellen?

Ja, solange der CSI-Treiber von Amazon EBS installiert ist.

Wird der **`kubernetes.io/aws-ebs StorageClass`**-Provisioner jemals aus Amazon EKS entfernt?

Der Provisioner `kubernetes.io/aws-ebs` `StorageClass` und der Volume-Typ `awsElasticBlockStore` werden zwar nicht mehr unterstützt, aber es gibt keine Pläne, sie zu entfernen. Diese Ressourcen werden als Teil der Kubernetes-API behandelt.

Wie installiere ich den CSI-Treiber von Amazon EBS?

Wir empfehlen, das Amazon-EKS-Add-on für den Amazon-EBS-CSI-Treiber zu installieren. Wenn das Amazon-EKS-Add-on aktualisiert werden muss, initiieren Sie das Update. Amazon EKS aktualisiert dann das Add-on für Sie. Wenn Sie den Treiber selbst verwalten möchten, können Sie ihn mit dem Open-Source-Helm-Chart installieren.

> **Important**
>
> Der In-Tree-Amazon-EBS-Treiber von Kubernetes wird auf der Kubernetes-Steuerebene ausgeführt. Er nutzt IAM-Berechtigungen, die der Amazon-EKS-Cluster-IAM-Rolle

zugewiesen sind, um Amazon-EBS-Volumes bereitzustellen. Der CSI-Treiber von Amazon EBS wird auf Knoten ausgeführt. Der Treiber benötigt IAM-Berechtigungen, um Volumes bereitzustellen. Weitere Informationen finden Sie unter Erstellen der IAM-Rolle des Amazon-EBS-CSI-Treibers.

Wie kann ich überprüfen, ob der CSI-Treiber von Amazon EBS in meinem Cluster installiert ist?

Um zu überprüfen, ob der Treiber auf Ihrem Cluster installiert ist, führen Sie den folgenden Befehl aus:

```
kubectl get csidriver ebs.csi.aws.com
```

Um zu überprüfen, ob diese Installation von Amazon EKS verwaltet wird, führen Sie den folgenden Befehl aus:

```
aws eks list-addons --cluster-name my-cluster
```

Verhindert Amazon EKS ein Cluster-Update auf Version **1.23**, wenn ich den CSI-Treiber von Amazon EBS noch nicht installiert habe?

Nein.

Was passiert, wenn ich vergesse, den CSI-Treiber von Amazon EBS zu installieren, bevor ich meinen Cluster auf Version 1.23 aktualisiere? Kann ich den Treiber nach der Aktualisierung meines Clusters installieren?

Ja, aber Volume-Vorgänge, für die der CSI-Treiber von Amazon EBS erforderlich ist, schlagen nach der Aktualisierung des Clusters so lange fehl, bis der Treiber installiert ist.

Welche Standard-**StorageClass** wird in neu erstellten Clustern der Amazon-EKS-Version **1.23** und höher angewendet?

Das Standardverhalten von `StorageClass` bleibt unverändert. Bei jedem neuen Cluster wendet Amazon EKS eine `kubernetes.io/aws-ebs`-basierte `StorageClass` mit dem Namen `gp2` an. Wir planen nicht, diese `StorageClass` jemals aus neu erstellten Clustern zu entfernen.

Unabhängig von der Standard-`StorageClass` für Cluster verwendet der CSI-Treiber von Amazon EBS standardmäßig `gp3`, wenn Sie eine `ebs.csi.aws.com`-basierte `StorageClass` ohne Angabe eines Volume-Typs erstellen.

Nimmt Amazon EKS Änderungen an **`StorageClasses`** vor, die bereits in meinem bestehenden Cluster vorhanden sind, wenn ich meinen Cluster auf Version **`1.23`** aktualisiere?

Nein.

Wie migriere ich ein persistentes Volume mit Snapshots aus der **`kubernetes.io/aws-ebsStorageClass`** zu **`ebs.csi.aws.com`**?

Informationen zum Migrieren eines persistenten Volumes finden Sie unter Migrating Amazon EKS clusters from gp2 to gp3 EBS volumes (Migrieren von Amazon-EKS-Clustern von gp2- zu gp3-EBS-Volumes) im AWS-Blog.

Wie ändere ich ein Amazon-EBS-Volume mithilfe von Anmerkungen?

Ab `aws-ebs-csi-driver` `v1.19.0-eksbuild.2` können Sie Amazon-EBS-Volumes mithilfe von Anmerkungen in ihren `PersistentVolumeClaims` (PVC) ändern. Das neue Feature zur Volumenänderung ist als zusätzliches Sidecar namens `volumemodifier` implementiert. Weitere Informationen finden Sie unter Vereinfachen der Amazon-EBS-Volume-Migration und -Modifikation in Kubernetes mit dem EBS-CSI-Treiber im AWS-Blog.

Wird die Migration für Windows-Workloads unterstützt?

Ja. Wenn Sie den CSI-Treiber von Amazon EBS mithilfe des Open-Source-Helm-Charts installieren, stellen Sie `node.enableWindows` auf `true` ein. Dies ist standardmäßig so eingestellt, wenn Sie den CSI-Treiber von Amazon EBS als Amazon-EKS-Add-on installieren. Beim Erstellen von `StorageClasses` stellen Sie den `fsType` auf ein Windows-Dateisystem wie `ntfs` ein. Volume-Vorgänge für Windows-Workloads werden dann genauso wie für Linux-Workloads zum CSI-Treiber von Amazon EBS migriert.

Amazon EFS-CSI-Treiber

Amazon Elastic File System (Amazon EFS) bietet vollständig elastischen Serverless-Dateispeicher, sodass Sie Dateidaten gemeinsam nutzen können, ohne Speicherkapazität und Leistung bereitstellen

oder verwalten zu müssen. Der Amazon EFS Container Storage Interface (CSI) -Treiber bietet eine CSI-Schnittstelle, mit der Kubernetes Cluster, auf denen AWS sie ausgeführt werden, den Lebenszyklus von Amazon EFS-Dateisystemen verwalten können. In diesem Thema erfahren Sie, wie Sie den Amazon-EFS-CSI-Treiber für Ihren Amazon-EKS-Cluster bereitstellen.

Überlegungen

- Der Amazon-EFS-CSI-Treiber ist nicht mit Windows-basierten Container-Images kompatibel.
- Sie können keine dynamische Bereitstellung für persistente Volumes mit Fargate-Knoten verwenden, aber Sie können statische Bereitstellung verwenden.
- Für die dynamische Bereitstellung ist der Treiber `1.2` oder höher erforderlich. Sie können die statische Bereitstellung für persistente Volumes verwenden, indem Sie die Version `1.1` des Treibers auf jeder unterstützten Amazon EKS-Cluster-Version verwenden.
- Version `1.3.2` oder höher dieses Treibers unterstützt die Arm64-Architektur, einschließlich auf Amazon EC2 Graviton basierenden Instances.
- Version `1.4.2` oder höher dieses Treibers unterstützt die Verwendung von FIPS zum Mounten von Dateisystemen.
- Beachten Sie die Ressourcenkontingente für Amazon EFS. Beispielsweise gibt es ein Kontingent von 1000 Zugriffspunkten, die für jedes Amazon-EFS-Dateisystem erstellt werden können. Weitere Informationen finden Sie unter Amazon-EFS-Ressourcenkontingente, die Sie nicht ändern können.

Voraussetzungen

- Ein vorhandener AWS Identity and Access Management (IAM) OpenID Connect (OIDC) Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.

> **Note**
>
> A Pod running on mountet AWS Fargate automatisch ein Amazon EFS-Dateisystem.

Erstellen einer IAM-Rolle

Der Amazon-EFS-CSI-Treiber benötigt IAM-Berechtigungen, um mit Ihrem Dateisystem zu interagieren. Erstellen Sie eine IAM-Rolle und fügen Sie ihr die erforderliche AWS verwaltete Richtlinie hinzu. Sie können `eksctl`, die AWS Management Console oder die AWS CLI verwenden.

> **Note**
>
> Die spezifischen Schritte in diesem Verfahren sind für die Verwendung des Treibers als Amazon-EKS-Add-on geschrieben. Ausführliche Informationen zu selbstverwalteten Installationen finden Sie auf GitHub unter Set up driver permission.

eksctl

So erstellen Sie Ihre IAM-Rolle des Amazon-EFS-CSI-Treibers mit **`eksctl`**

Führen Sie die folgenden Befehle aus, um die IAM-Rolle zu erstellen. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und *`AmazonEKS_EFS_CSI_DriverRole`* durch den Namen Ihrer Rolle.

```
export cluster_name=my-cluster
export role_name=AmazonEKS_EFS_CSI_DriverRole
eksctl create iamserviceaccount \
    --name efs-csi-controller-sa \
    --namespace kube-system \
    --cluster $cluster_name \
    --role-name $role_name \
```

```
    --role-only \
    --attach-policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEFSCSIDriverPolicy \
    --approve
TRUST_POLICY=$(aws iam get-role --role-name $role_name --query
 'Role.AssumeRolePolicyDocument' | \
    sed -e 's/efs-csi-controller-sa/efs-csi-*/' -e 's/StringEquals/StringLike/')
aws iam update-assume-role-policy --role-name $role_name --policy-document
 "$TRUST_POLICY"
```

AWS Management Console

So erstellen Sie Ihre Amazon EFS CSI-Treiber-IAM-Rolle mit dem AWS Management Console

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Klicken Sie auf der Seite Roles (Rollen) auf Create role (Rolle erstellen).
4. Gehen Sie auf der Seite Select trusted entity (Vertrauenswürdige Entität auswählen) wie folgt vor:
   a. Wählen Sie im Abschnitt Trusted entity type (Typ der vertrauenswürdigen Entität) die Option Web identity (Web-Identität) aus.
   b. Wählen Sie für Identity provider (Identitätsanbieter) die Option OpenID Connect provider URL (-Anbieter-URL) für Ihren Cluster aus (wie unter Overview (Übersicht) in Amazon EKS gezeigt).
   c. Wählen Sie für Audience (Zielgruppe) `sts.amazonaws.com`.
   d. Wählen Sie Next (Weiter).
5. Gehen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) wie folgt vor:
   a. Geben Sie im Feld Filter policies (Filterrichtlinien) *AmazonEFSCSIDriverPolicy* ein.
   b. Aktivieren Sie das Kontrollkästchen links neben der *AmazonEFSCSIDriverPolicy*, die bei der Suche zurückgegeben wurde.
   c. Wählen Sie Next (Weiter).
6. Gehen Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) wie folgt vor:
   a. Geben Sie unter Role name (Rollenname) einen eindeutigen Namen für die Rolle ein, z. B. ***AmazonEKS_EFS_CSI_DriverRole***.

b. Fügen Sie der Rolle unter Tags hinzufügen (optional) Metadaten hinzu, indem Sie Tags als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.

c. Wählen Sie Create role (Rolle erstellen) aus.

7. Nachdem die Rolle erstellt wurde, wählen Sie die Rolle in der Konsole aus, um sie zur Bearbeitung zu öffnen.
8. Wählen Sie die Registerkarte Trust Relationships (Vertrauensstellungen) und dann Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.
9. Suchen Sie die Zeile, die der folgenden Zeile ähnelt:

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud":
 "sts.amazonaws.com"
```

Fügen Sie die folgende Zeile über der vorherigen Zeile hinzu. *region-code*Ersetzen Sie es durch AWS-Region das, in dem sich Ihr Cluster befindet. Ersetzen von *EXAMPLED539D4633E53DE1B71EXAMPLE* mit der OIDC-Anbieter-ID Ihres Clusters.

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub":
 "system:serviceaccount:kube-system:efs-csi-*",
```

10. Ändern Sie den `Condition`-Operator von `"StringEquals"` zu `"StringLike"`.
11. Wählen Sie Update Policy (Richtlinie aktualisieren) aus, um den Vorgang abzuschließen.

AWS CLI

So erstellen Sie Ihre Amazon EFS CSI-Treiber-IAM-Rolle mit dem AWS CLI

1. Zeigen Sie die OIDC-Anbieter-URL Ihres Clusters an. Ersetzen Sie *my-cluster* durch Ihren Clusternamen. Wenn die Ausgabe des Befehls `None` ist, überprüfen Sie die Voraussetzungen.

```
aws eks describe-cluster --name my-cluster --query
 "cluster.identity.oidc.issuer" --output text
```

Eine Beispielausgabe sieht wie folgt aus.

```
https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
```

2. Erstellen Sie die IAM-Rolle, die die Aktion `AssumeRoleWithWebIdentity` erlaubt.

   a. Kopieren Sie den folgenden Inhalt in eine Datei namens *aws-efs-csi-driver-trust-policy*`.json`. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Ersetzen Sie *EXAMPLED539D4633E53DE1B71EXAMPLE* und *region-code* mit den im vorherigen Schritt zurückgegebenen Werten. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (USA-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch. `arn:aws:` `arn:aws-us-gov:`

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Federated": "arn:aws:iam::111122223333:oidc-provider/
   oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
         },
         "Action": "sts:AssumeRoleWithWebIdentity",
         "Condition": {
           "StringLike": {
             "oidc.eks.region-code.amazonaws.com/
   id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
   system:efs-csi-*",
             "oidc.eks.region-code.amazonaws.com/
   id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
           }
         }
       }
     ]
   }
   ```

   b. Erstellen Sie die -Rolle. Sie können *AmazonEKS_EFS_CSI_DriverRole* in einen anderen Namen ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie ihn auch in späteren Schritten ändern.

   ```
   aws iam create-role \
     --role-name AmazonEKS_EFS_CSI_DriverRole \
   ```

```
  --assume-role-policy-document file://"aws-efs-csi-driver-trust-
policy.json"
```

3. Fügen Sie der Rolle mit dem folgenden Befehl die erforderliche AWS verwaltete Richtlinie hinzu. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (USA-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch`arn:aws:`. `arn:aws-us-gov:`

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEFSCSIDriverPolicy \
  --role-name AmazonEKS_EFS_CSI_DriverRole
```

Installation des Amazon-EFS-CSI-Treibers

Wir empfehlen, den Amazon-EFS-CSI-Treiber über das Amazon-EKS-Add-on zu installieren. Wenn Sie ein Amazon-EKS-Add-on zu Ihrem Cluster hinzufügen möchten, lesen Sie Erstellen eines Add-Ons. Weitere Informationen zu Add-ons finden Sie unter Amazon-EKS-Add-ons. Wenn Sie das Amazon-EKS-Add-on nicht verwenden können, empfehlen wir Ihnen, ein Problem zu den Gründen, warum Sie es nicht verwenden können, an das GitHub-Repository für Containers-Roadmap zu senden.

Wenn Sie alternativ eine selbstverwaltete Installation des Amazon-EFS-CSI-Treibers wünschen, finden Sie Informationen unter Installation in GitHub.

Erstellen eines Amazon-EFS-Dateisystems

Informationen zum Erstellen eines Amazon-EFS-Dateisystems finden Sie unter Erstellen eines Amazon-EFS-Dateisystems für Amazon EKS in GitHub.

Bereitstellen einer Beispielanwendung

Sie können eine Vielzahl von Beispiel-Apps bereitstellen und diese nach Bedarf ändern. Weitere Informationen finden Sie unter Beispiele in GitHub.

Amazon FSx for Lustre-CSI-Treiber

Der Treiber für FSx for Lustre Container Storage Interface (CSI) bietet eine CSI-Schnittstelle, mit der Amazon-EKS-Cluster den Lebenszyklus von FSx-for-Lustre-Dateisystemen verwalten können. Weitere Informationen finden Sie im FSx-for-Lustre-Benutzerhandbuch.

In diesem Thema erfahren Sie, wie Sie den FSx-for-Lustre-CSI-Treiber für Ihren Amazon-EKS-Cluster bereitstellen und überprüfen, ob er funktioniert. Wir empfehlen die neueste Version des Treibers zu verwenden. Verfügbare Versionen finden Sie in der Kompatibilitätsmatrix der CSI-Spezifikation unter GitHub.

> **Note**
>
> Der Treiber wird auf Fargate nicht unterstützt.

Detaillierte Beschreibungen der verfügbaren Parameter und vollständige Beispiele, die die Features des Treibers demonstrieren, finden Sie im Projekt Treiber für FSx for Lustre Container Storage Interface (CSI) auf GitHub.

Voraussetzungen

Sie müssen über Folgendes verfügen:

- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.

Die folgenden Verfahren helfen Ihnen beim Erstellen eines einfachen Testclusters mit dem CSI-Treiber für FSx für Lustre, damit Sie sehen können, wie es funktioniert. Es wird nicht empfohlen, den Test-Cluster für Produktions-Workloads zu verwenden. Für dieses Tutorial empfehlen wir die Verwendung der *example values*, außer es gibt eine Anmerkung, diese zu ersetzen. Sie können jeden *example value* ersetzen, wenn Sie die Schritte für einen Produktionscluster ausführen. Wir empfehlen, alle Schritte auf demselben Terminal auszuführen, weil Variablen während der Schritte festgelegt und verwendet werden und diese in anderen Terminals nicht vorhanden sind.

So stellen Sie den Treiber für FSx for Lustre CSI in einem Amazon-EKS-Cluster bereit

1. Legen Sie einige Variablen fest, die in den verbleibenden Schritten verwendet werden sollen. *my-csi-fsx-cluster*Ersetzen Sie es durch den Namen des Testclusters, den Sie erstellen möchten, und *region-code* durch den Namen AWS-Region , in dem Sie Ihren Testcluster erstellen möchten.

   ```
   export cluster_name=my-csi-fsx-cluster
   export region_code=region-code
   ```

2. Erstellen Sie einen Testcluster.

   ```
   eksctl create cluster \
     --name $cluster_name \
     --region $region_code \
     --with-oidc \
     --ssh-access \
     --ssh-public-key my-key
   ```

   Die Clusterbereitstellung dauert mehrere Minuten. Während der Clustererstellung werden mehrere Ausgabezeilen angezeigt. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

   ```
   [#]  EKS cluster "my-csi-fsx-cluster" in "region-code" region is ready
   ```

3. Erstellen Sie ein Kubernetes Dienstkonto für den Treiber und fügen Sie die `AmazonFSxFullAccess` AWS-managed Policy mit dem folgenden Befehl an das Dienstkonto an. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (US-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch. `arn:aws:` `arn:aws-us-gov:`

   ```
   eksctl create iamserviceaccount \
     --name fsx-csi-controller-sa \
     --namespace kube-system \
   ```

```
  --cluster $cluster_name \
  --attach-policy-arn arn:aws:iam::aws:policy/AmazonFSxFullAccess \
  --approve \
  --role-name AmazonEKSFSxLustreCSIDriverFullAccess \
  --region $region_code
```

Beim Erstellen des Servicekontos werden Ihnen mehrere Ausgabezeilen angezeigt. Die letzten Ausgabezeilen ähneln den folgenden.

```
[#]  1 task: {
    2 sequential sub-tasks: {
        create IAM role for serviceaccount "kube-system/fsx-csi-controller-sa",
        create serviceaccount "kube-system/fsx-csi-controller-sa",
    } }
[#]  building iamserviceaccount stack "eksctl-my-csi-fsx-cluster-addon-
iamserviceaccount-kube-system-fsx-csi-controller-sa"
[#]  deploying stack "eksctl-my-csi-fsx-cluster-addon-iamserviceaccount-kube-
system-fsx-csi-controller-sa"
[#]  waiting for CloudFormation stack "eksctl-my-csi-fsx-cluster-addon-
iamserviceaccount-kube-system-fsx-csi-controller-sa"
[#]  created serviceaccount "kube-system/fsx-csi-controller-sa"
```

Notieren Sie sich den Namen des AWS CloudFormation Stacks, der bereitgestellt wurde. In der vorigen Beispielausgabe lautet der Name des Stacks `eksctl-`*`my-csi-fsx-cluster`*`-addon-iamserviceaccount-kube-system-`*`fsx-csi-controller-sa`*.

4. Stellen Sie den Treiber mit dem folgenden Befehl bereit. Ersetzen Sie *`release-X.XX`* durch die gewünschte Branch. Die Master-Branch wird nicht unterstützt, da sie möglicherweise zukünftige Features enthält, die mit der aktuell veröffentlichten stabilen Version des Treibers nicht kompatibel sind. Wir empfehlen die Verwendung der neuesten veröffentlichten Version. Eine Liste der aktiven Branches finden Sie aws-fsx-csi-driverunter GitHub.

> **Note**
>
> Sie die angewendeten Inhalte unter aws-fsx-csi-driver auf GitHub anzeigen.

```
kubectl apply -k "github.com/kubernetes-sigs/aws-fsx-csi-driver/deploy/kubernetes/
overlays/stable/?ref=release-X.XX"
```

Eine Beispielausgabe sieht wie folgt aus.

```
serviceaccount/fsx-csi-controller-sa created
serviceaccount/fsx-csi-node-sa created
clusterrole.rbac.authorization.k8s.io/fsx-csi-external-provisioner-role created
clusterrole.rbac.authorization.k8s.io/fsx-external-resizer-role created
clusterrolebinding.rbac.authorization.k8s.io/fsx-csi-external-provisioner-binding
 created
clusterrolebinding.rbac.authorization.k8s.io/fsx-csi-resizer-binding created
deployment.apps/fsx-csi-controller created
daemonset.apps/fsx-csi-node created
csidriver.storage.k8s.io/fsx.csi.aws.com created
```

5. Notieren Sie sich den ARN für die Rolle, die erstellt wurde. Wenn Sie es vorher nicht bemerkt haben und es in der AWS CLI Ausgabe nicht mehr verfügbar ist, können Sie wie folgt vorgehen, um es in der zu sehen AWS Management Console.

   a. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

   b. Stellen Sie sicher, dass die Konsole auf die Konsole eingestellt ist AWS-Region , in der Sie Ihre IAM-Rolle erstellt haben, und wählen Sie dann Stacks aus.

   c. Wählen Sie den Stack mit dem Namen `eksctl-`*`my-csi-fsx-cluster`*`-addon-iamserviceaccount-kube-system-`*`fsx-csi-controller-sa`* aus.

   d. Wählen Sie die Registerkarte Ausgaben aus. Der Role1-ARN wird auf der Seite Outputs (1) ((Ausgaben (1)) aufgeführt.

6. Patchen Sie die Treiberbereitstellung, um das zuvor erstellte Servicekonto mit dem folgenden Befehl hinzuzufügen. Ersetzen Sie den ARN durch den ARN, den Sie notiert haben. Ersetzen Sie *`111122223333`* durch Ihre Konto-ID. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (US-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch. `arn:aws:` `arn:aws-us-gov:`

```
kubectl annotate serviceaccount -n kube-system fsx-csi-controller-sa \
  eks.amazonaws.com/role-
arn=arn:aws:iam::111122223333:role/AmazonEKSFSxLustreCSIDriverFullAccess --
overwrite=true
```

Eine Beispielausgabe sieht wie folgt aus.

```
serviceaccount/fsx-csi-controller-sa annotated
```

Eine Kubernetes-Speicherklasse, einen dauerhaften Volume-Anspruch und eine Beispielanwendung bereitstellen, um zu überprüfen, ob der CSI-Treiber funktioniert

In diesem Verfahren wird das GitHub-Repository-Treiber für FSx für Lustre Container Storage Interface (CSI) verwendet, um ein dynamisch bereitgestelltes FSx-für-Lustre-Volume zu verwenden.

1. Beachten Sie die Sicherheitsgruppe für Ihren Cluster. Sie können es im AWS Management Console Bereich Netzwerk oder mit dem folgenden AWS CLI Befehl sehen.

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

2. Erstellen Sie gemäß den im Benutzerhandbuch für Amazon FSx für Lustre unter Amazon-VPC-Sicherheitsgruppen aufgeführten Kriterien eine Sicherheitsgruppe für Ihr Amazon-FSx-Dateisystem. Als VPC wählen Sie die VPC Ihres Clusters aus, die im Abschnitt Networking (Netzwerk) gezeigt wird. Unter „the security groups associated with your Lustre clients“ (die mit Ihren Lustre-Clients verknüpften Sicherheitsgruppen) wählen Sie Ihre Cluster-Sicherheitsgruppe aus. Wenn Sie keine Regeln für ausgehenden Datenverkehr festlegen, können Sie den All traffic (gesamten Datenverkehr) erlauben.
3. Laden Sie das Speicherklassen-Manifest mit dem folgenden Befehl herunter.

   ```
   curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-fsx-csi-driver/
   master/examples/kubernetes/dynamic_provisioning/specs/storageclass.yaml
   ```

4. Bearbeiten Sie den Parameterabschnitt in der Datei `storageclass.yaml`. Ersetzen Sie jede *`example value`* durch Ihre eigenen Werte.

   ```
   parameters:
     subnetId: subnet-0eabfaa81fb22bcaf
     securityGroupIds: sg-068000ccf82dfba88
     deploymentType: PERSISTENT_1
     automaticBackupRetentionDays: "1"
     dailyAutomaticBackupStartTime: "00:00"
     copyTagsToBackups: "true"
     perUnitStorageThroughput: "200"
     dataCompressionType: "NONE"
   ```

```
  weeklyMaintenanceStartTime: "7:09:00"
  fileSystemTypeVersion: "2.12"
```

- **`subnetId`** – Die Subnetz-ID, in der das Amazon-FSx-for-Lustre-Dateisystem erstellt werden soll. Amazon FSx for Lustre wird nicht in allen Availability Zones unterstützt. Öffnen Sie die Amazon FSx for Lustre-Konsole unter https://console.aws.amazon.com/fsx/, um zu bestätigen, dass sich das Subnetz, das Sie verwenden möchten, in einer unterstützten Availability Zone befindet. Das Subnetz kann Ihre Knoten enthalten oder ein anderes Subnetz oder eine andere VPC sein:
  - Sie können nach den Knoten-Subnetzen in suchen, AWS Management Console indem Sie die Knotengruppe im Abschnitt Compute auswählen.
  - Wenn das von Ihnen angegebene Subnetz nicht das Subnetz ist, in dem Knoten vorhanden sind, müssen die VPCs verbunden sein und Sie müssen sicherstellen, dass die erforderlichen Ports in Ihren Sicherheitsgruppen geöffnet sind.
- **`securityGroupIds`** – Die ID der Sicherheitsgruppe, die Sie für das Dateisystem erstellt haben.
- **`deploymentType`**(optional) – Der Bereitstellungstyp des Dateisystems. Gültige Werte sind `SCRATCH_1`, `SCRATCH_2`, `PERSISTENT_1` und `PERSISTENT_2`. Weitere Informationen zu Bereitstellungstypen finden Sie unter Erstellen Sie Ihr Amazon-FSx-for-Lustre-Dateisystem.
- andere Parameter (optional) — Informationen zu den anderen Parametern finden Sie unter Bearbeiten StorageClass amGitHub.

5. Erstellen Sie das Speicherklassen-Manifest.

```
kubectl apply -f storageclass.yaml
```

Eine Beispielausgabe sieht wie folgt aus.

```
storageclass.storage.k8s.io/fsx-sc created
```

6. Laden Sie das Manifest für den dauerhaften Volume-Anspruch herunter.

```
curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-fsx-csi-driver/
master/examples/kubernetes/dynamic_provisioning/specs/claim.yaml
```

7. (Optional) Bearbeiten Sie die `claim.yaml`-Datei. Ändern Sie *`1200Gi`* in einen der folgenden Inkrementwerte, basierend auf Ihren Speicheranforderungen und dem `deploymentType`, den Sie in einem vorherigen Schritt ausgewählt haben.

```
storage: 1200Gi
```

- SCRATCH_2 und PERSISTENT – **1.2 TiB**, **2.4 TiB** oder Schritte von 2,4 TiB über 2,4 TiB.
- SCRATCH_1 – **1.2 TiB**, **2.4 TiB**, **3.6 TiB** oder Schritte von 3,6 TiB über 3,6 TiB.

8. Erstellen Sie den dauerhaften Volume-Anspruch.

```
kubectl apply -f claim.yaml
```

Eine Beispielausgabe sieht wie folgt aus.

```
persistentvolumeclaim/fsx-claim created
```

9. Vergewissern Sie sich, dass das Dateisystem bereitgestellt wurde.

```
kubectl describe pvc
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:          fsx-claim
Namespace:     default
StorageClass:  fsx-sc
Status:        Bound
[...]
```

> **Note**
>
> Der `Status` kann für 5-10 Minuten als `Pending` angezeigt werden, bevor er zu `Bound` wechselt. Fahren Sie nicht mit dem nächsten Schritt fort, bis der `Status` `Bound` ist. Wenn der `Status` länger als 10 Minuten `Pending` anzeigt, verwenden Sie die Warnmeldungen in den `Events` als Referenz zur Behebung von Problemen.

10. Stellen Sie die Beispielanwendung bereit.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-fsx-csi-driver/master/examples/kubernetes/dynamic_provisioning/specs/pod.yaml
```

11. Stellen Sie sicher, dass die Beispielanwendung ausgeführt wird.

```
kubectl get pods
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME      READY   STATUS              RESTARTS   AGE
fsx-app   1/1     Running             0          8s
```

12. Überprüfen Sie, ob das Dateisystem ordnungsgemäß von der Anwendung aufgespielt wurde.

```
kubectl exec -ti fsx-app -- df -h
```

Eine Beispielausgabe sieht wie folgt aus.

```
Filesystem                  Size  Used Avail Use% Mounted on
overlay                      80G  4.0G   77G   5% /
tmpfs                        64M     0   64M   0% /dev
tmpfs                       3.8G     0  3.8G   0% /sys/fs/cgroup
192.0.2.0@tcp:/abcdef01     1.1T  7.8M  1.1T   1% /data
/dev/nvme0n1p1               80G  4.0G   77G   5% /etc/hosts
shm                          64M     0   64M   0% /dev/shm
tmpfs                       6.9G   12K  6.9G   1% /run/secrets/kubernetes.io/
serviceaccount
tmpfs                       3.8G     0  3.8G   0% /proc/acpi
tmpfs                       3.8G     0  3.8G   0% /sys/firmware
```

13. Stellen Sie sicher, dass Daten von der Beispiel-App in das FSx-for-Lustre-Dateisystem geschrieben wurden.

```
kubectl exec -it fsx-app -- ls /data
```

Eine Beispielausgabe sieht wie folgt aus.

```
out.txt
```

Diese Beispielausgabe zeigt, dass die Beispiel-App erfolgreich die `out.txt`-Datei in das Dateisystem geschrieben hat.

> **Note**
>
> Stellen Sie vor dem Löschen des Clusters sicher, dass Sie das FSx-for-Lustre-Dateisystem löschen. Weitere Informationen finden Sie unter Bereinigen von Ressourcen im Benutzerhandbuch zu FSx for Lustre.

Amazon FSx für NetApp ONTAP CSI-Treiber

NetApp'sAstra Tridentbietet dynamische Speicher-Orchestrierung mithilfe eines Container-Storage-Interface (CSI) -kompatiblen Treibers. Auf diese Weise können Amazon EKS-Cluster den Lebenszyklus persistenter Volumes (PVs) verwalten, die von Amazon FSx für NetApp ONTAP-Dateisysteme unterstützt werden. Erste Schritte finden Sie in der Dokumentation unter Verwenden von Astra Trident mit Amazon FSx für NetApp ONTAP. Astra Trident

Amazon FSx for NetApp ONTAP ist ein Speicherservice, mit dem Sie vollständig verwaltete ONTAP Dateisysteme in der Cloud starten und ausführen können. ONTAPist eine NetApp's Dateisystemtechnologie, die eine breite Palette von Datenzugriffs- und Datenverwaltungsfunktionen bietet. FSx for ONTAP bietet die Funktionen, Leistung und APIs von lokalen NetApp Dateisystemen mit der Agilität, Skalierbarkeit und Einfachheit eines vollständig verwalteten Dienstes. AWS Weitere Informationen finden Sie im FSx für ONTAP-Benutzerhandbuch.

Amazon FSx für OpenZFS-CSI-Treiber

Amazon FSx für OpenZFS ist ein vollständig verwalteter Dateispeicherservice, der das Verschieben von Daten zu AWS von On-Premises-ZFS- oder anderen Linux-basierten Dateiservern erleichtert. Sie können dies tun, ohne Ihren Anwendungscode oder die Art und Weise, wie Sie Daten verwalten, zu ändern. Es bietet einen äußerst zuverlässigen, skalierbaren, effizienten und featurereichen Dateispeicher, der auf dem Open-Source-Dateisystem OpenZFS basiert. Es kombiniert diese Funktionen mit der Agilität, Skalierbarkeit und Einfachheit eines vollständig verwalteten AWS-Services. Weitere Informationen finden Sie im Benutzerhandbuch für Amazon FSx für OpenZFS.

Der Container Storage Interface (CSI)-Treiber für Amazon FSx für OpenZFS bietet eine CSI-Schnittstelle, die es Amazon-EKS-Clustern ermöglicht, den Lebenszyklus von Amazon-FSx-für-OpenZFS-Volumes zu verwalten. Informationen zur Bereitstellung des CSI-Treibers für Amazon FSx für OpenZFS in Ihrem Amazon-EKS-Cluster finden Sie unter `aws-fsx-openzfs-csi-driver` auf GitHub.

CSI-Treiber für Amazon File Cache

Amazon File Cache ist ein vollständig verwalteter Hochgeschwindigkeits-Cache in AWS, der zur Verarbeitung von Dateidaten verwendet wird, unabhängig davon, wo die Daten gespeichert sind. Amazon File Cache lädt Daten automatisch in den Cache, wenn zum ersten Mal darauf zugegriffen wird, und gibt Daten frei, wenn sie nicht verwendet werden. Weitere Informationen finden Sie in Benutzerhandbuch zu Amazon File Cache.

Der Container Storage Interface (CSI)-Treiber für Amazon File Cache bietet eine CSI-Schnittstelle, die es Amazon-EKS-Clustern ermöglicht, den Lebenszyklus von Amazon-Datei-Caches zu verwalten. Informationen zur Bereitstellung des CSI-Treibers für Amazon File Cache in Ihrem Amazon-EKS-Cluster finden Sie unter `aws-file-cache-csi-driver` auf GitHub.

Mountpoint für Amazon S3-CSI-Treiber

Mit dem Treiber Mountpoint für Amazon S3 Container Storage Interface (CSI) können Ihre Kubernetes Anwendungen über eine Dateisystemschnittstelle auf Amazon S3 S3-Objekte zugreifen und so einen hohen Gesamtdurchsatz erzielen, ohne dass der Anwendungscode geändert werden muss. Der CSI-Treiber basiert auf Mountpoint für Amazon S3 und stellt einen Amazon-S3-Bucket als Volume dar, auf das Container in Amazon EKS und selbstverwalteten Kubernetes-Clustern zugreifen können. In diesem Thema erfahren Sie, wie Sie den Mountpoint für Amazon S3-CSI-Treiber für Ihren Amazon-EKS-Cluster bereitstellen.

Überlegungen

- Der Mountpoint für Amazon S3-CSI-Treiber ist derzeit nicht mit Windows-basierten Container-Images kompatibel.
- AWS Fargate wird vom Mountpoint für Amazon S3-CSI-Treiber nicht unterstützt. Container, die in Amazon EC2 ausgeführt werden (entweder mit Amazon EKS oder einer benutzerdefinierten Kubernetes-Installation), werden allerdings unterstützt.
- Der Mountpoint für Amazon S3-CSI-Treiber unterstützt nur die statische Bereitstellung. Dynamische Bereitstellung oder die Erstellung neuer Buckets wird nicht unterstützt.

> **Note**
>
> Statische Bereitstellung bezieht sich auf die Verwendung eines vorhandenen Amazon S3 S3-Buckets, der als `bucketName` `volumeAttributes` im `PersistentVolume` Objekt angegeben ist. Weitere Informationen finden Sie im GitHub unter Statische Bereitstellung.

- Volumes, die mit dem Mountpoint für Amazon S3-CSI-Treiber bereitgestellt wurden, unterstützen nicht alle Features des POSIX-Dateisystems. Einzelheiten zum Verhalten des Dateisystems finden Sie im GitHub unter Mountpoint für Amazon S3 Dateisystemverhalten.

Voraussetzungen

- Ein vorhandener AWS Identity and Access Management (IAM) OpenID Connect (OIDC) -Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Version 2.12.3 oder höher der auf AWS CLI Ihrem Gerät installierten und konfigurierten Version oder. AWS CloudShell
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.

Erstellen einer IAM-Richtlinie

Der Mountpoint für Amazon S3-CSI-Treiber benötigt Amazon S3-Berechtigungen, um mit Ihrem Dateisystem zu interagieren. In diesem Abschnitt wird erläutert, wie Sie eine IAM-Richtlinie erstellen, die die erforderlichen Berechtigungen gewährt.

Die folgende Beispielrichtlinie richtet sich nach den IAM-Berechtigungsempfehlungen für Mountpoint. Sie können auch die AWS verwaltete Richtlinie verwenden AmazonS3FullAccess, aber diese verwaltete Richtlinie gewährt mehr Berechtigungen, als erforderlich sind. Mountpoint

Weitere Informationen zu den empfohlenen Berechtigungen für Mountpoint finden Sie im GitHub unter Mountpoint IAM-Berechtigungen.

Eine IAM-Richtlinie mithilfe der IAM-Konsole erstellen

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich die Option Policies aus.
3. Wählen Sie auf der Seite Richtlinien die Option Richtlinie erstellen.
4. Wählen Sie für Richtlinien-Editor die Option JSON aus.
5. Kopieren Sie unter Richtlinien-Editor Folgendes und fügen Sie es ein:

> **Important**
>
> Ersetzen Sie `DOC-EXAMPLE-BUCKET1` durch den Namen Ihres eigenen Amazon-S3-Bucket.

```
{
   "Version": "2012-10-17",
   "Statement": [
        {
            "Sid": "MountpointFullBucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
            ]
        },
        {
            "Sid": "MountpointFullObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:AbortMultipartUpload",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"
            ]
        }
```

```
    ]
}
```

Directory-Buckets, die mit der Amazon S3 Express One Zone-Speicherklasse eingeführt wurden, verwenden einen anderen Authentifizierungsmechanismus als Buckets für allgemeine Zwecke. Anstatt `s3:*` Aktionen zu verwenden, sollten Sie die `s3express:CreateSession` Aktion verwenden. Informationen zu Directory-Buckets finden Sie unter Directory-Buckets im Amazon S3 S3-Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine Richtlinie mit den geringsten Rechten, die Sie für einen Directory-Bucket verwenden würden.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3express:CreateSession",
            "Resource": "arn:aws:s3express:aws-region:111122223333:bucket/DOC-
EXAMPLE-BUCKET1--az_id--x-s3"
        }
    ]
}
```

6. Wählen Sie Weiter aus.
7. Geben Sie Ihrer Richtlinie auf der Seite Überprüfen und erstellen einen Namen. In dieser Beispielanleitung wird der Name `AmazonS3CSIDriverPolicy` verwendet.
8. Wählen Sie Richtlinie erstellen aus.

Erstellen einer IAM-Rolle

Der Mountpoint für Amazon S3-CSI-Treiber benötigt Amazon S3-Berechtigungen, um mit Ihrem Dateisystem zu interagieren. In diesem Abschnitt wird erläutert, wie Sie eine IAM-Rolle erstellen, mit der Sie diese Berechtigungen delegieren. Sie können diese Rolle mithilfe von `eksctl`, der IAM-Konsole oder der AWS CLI erstellen.

> **Note**
>
> Die IAM-Richtlinie `AmazonS3CSIDriverPolicy` wurde im vorherigen Abschnitt erstellt.

eksctl

Erstellen Ihrer IAM-Rolle für den Mountpoint für Amazon S3-CSI-Treiber mit **`eksctl`**

Führen Sie die folgenden Befehle aus, um die IAM-Rolle und das Kubernetes-Servicekonto zu erstellen. Über diese Befehle wird außerdem die `AmazonS3CSIDriverPolicy` IAM-Richtlinie an die Rolle angehängt, das Kubernetes-Servicekonto (`s3-csi-controller-sa`) mit dem Amazon-Ressourcennamen (ARN) der IAM-Rolle versehen und der Name des Kubernetes-Servicekontos zur Vertrauensrichtlinie für die IAM-Rolle hinzugefügt.

```
CLUSTER_NAME=my-cluster
REGION=region-code
ROLE_NAME=AmazonEKS_S3_CSI_DriverRole
POLICY_ARN=AmazonEKS_S3_CSI_DriverRole_ARN
eksctl create iamserviceaccount \
    --name s3-csi-driver-sa \
    --namespace kube-system \
    --cluster $CLUSTER_NAME \
    --attach-policy-arn $POLICY_ARN \
    --approve \
    --role-name $ROLE_NAME \
    --region $REGION \
    --role-only
```

IAM console

So erstellen Sie Ihre IAM-Rolle Mountpoint für Amazon S3 CSI-Treiber mit dem AWS Management Console

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Klicken Sie auf der Seite Roles (Rollen) auf Create role (Rolle erstellen).
4. Gehen Sie auf der Seite Select trusted entity (Vertrauenswürdige Entität auswählen) wie folgt vor:

a. Wählen Sie im Abschnitt Trusted entity type (Typ der vertrauenswürdigen Entität) die Option Web identity (Web-Identität) aus.

b. Wählen Sie für Identity provider (Identitätsanbieter) die Option OpenID Connect provider URL (-Anbieter-URL) für Ihren Cluster aus (wie unter Overview (Übersicht) in Amazon EKS gezeigt).

   Wenn keine URLs angezeigt werden, lesen Sie den Abschnitt Voraussetzungen.

c. Wählen Sie für Audience (Zielgruppe) `sts.amazonaws.com`.

d. Wählen Sie Next (Weiter).

5. Gehen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) wie folgt vor:

   a. Geben Sie im Feld Filter policies (Filterrichtlinien) **`AmazonS3CSIDriverPolicy`** ein.

   > **Note**
   >
   > Diese Richtlinie wurde im vorherigen Abschnitt erstellt.

   b. Aktivieren Sie das Kontrollkästchen links neben dem Ergebnis für `AmazonS3CSIDriverPolicy`, das bei der Suche zurückgegeben wurde.

   c. Wählen Sie Next (Weiter).

6. Gehen Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) wie folgt vor:

   a. Geben Sie unter Role name (Rollenname) einen eindeutigen Namen für die Rolle ein, z. B. **`AmazonEKS_S3_CSI_DriverRole`**.

   b. Fügen Sie der Rolle unter Tags hinzufügen (optional) Metadaten hinzu, indem Sie Tags als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.

   c. Wählen Sie Create role (Rolle erstellen) aus.

7. Nachdem die Rolle erstellt wurde, wählen Sie die Rolle in der Konsole aus, um sie zur Bearbeitung zu öffnen.

8. Wählen Sie die Registerkarte Trust Relationships (Vertrauensstellungen) und dann Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.

9. Suchen Sie die Zeile, die wie folgt aussieht:

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud":
 "sts.amazonaws.com"
```

Fügen Sie am Ende der vorherigen Zeile ein Komma und anschließend die folgende Zeile hinzu. *region-code*Ersetzen Sie es durch AWS-Region das, in dem sich Ihr Cluster befindet. Ersetzen von *EXAMPLED539D4633E53DE1B71EXAMPLE* mit der OIDC-Anbieter-ID Ihres Clusters.

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub":
 "system:serviceaccount:kube-system:s3-csi-*"
```

10. Ändern Sie die `Condition`-Operator von `"StringEquals"` zu `"StringLike"`.
11. Wählen Sie Update Policy (Richtlinie aktualisieren) aus, um den Vorgang abzuschließen.

AWS CLI

So erstellen Sie Ihre IAM-Rolle Mountpoint für Amazon S3 CSI-Treiber mit dem AWS CLI

1. Zeigen Sie die OIDC-Anbieter-URL Ihres Clusters an. Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters. Wenn die Ausgabe des Befehls `None` ist, überprüfen Sie die Voraussetzungen.

   ```
   aws eks describe-cluster --name my-cluster --query
    "cluster.identity.oidc.issuer" --output text
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
   ```

2. Erstellen Sie die IAM-Rolle und weisen Sie dem Kubernetes-Servicekonto die `AssumeRoleWithWebIdentity`-Aktion zu.

   a. Kopieren Sie den folgenden Inhalt in eine Datei namens *aws-s3-csi-driver-trust-policy*`.json`. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Ersetzen Sie *EXAMPLED539D4633E53DE1B71EXAMPLE* und *region-code* mit den im vorherigen Schritt zurückgegebenen Werten.

   ```
   {
   ```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringLike": {
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:s3-csi-*",
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
        }
      }
    }
  ]
}
```

b. Erstellen Sie die -Rolle. Sie können *AmazonEKS_S3_CSI_DriverRole* in einen anderen Namen ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie ihn auch in späteren Schritten ändern.

```
aws iam create-role \
  --role-name AmazonEKS_S3_CSI_DriverRole \
  --assume-role-policy-document file://"aws-s3-csi-driver-trust-policy.json"
```

3. Hängen Sie die zuvor erstellte IAM-Richtlinie mit dem folgenden Befehl an die Rolle an.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonS3CSIDriverPolicy \
  --role-name AmazonEKS_S3_CSI_DriverRole
```

> **Note**
>
> Die IAM-Richtlinie `AmazonS3CSIDriverPolicy` wurde im vorherigen Abschnitt erstellt.

4. Überspringen Sie diesen Schritt, wenn Sie den Treiber als Amazon-EKS-Add-on installieren. Für selbstverwaltete Installationen des Treibers erstellen Sie Kubernetes-Servicekonten, die mit dem ARN der IAM-Rolle versehen sind, die Sie erstellt haben.

   a. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *mountpoint-s3-service-account*`.yaml`. Ersetzen Sie *111122223333* durch Ihre Konto-ID.

```
---
apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/name: aws-mountpoint-s3-csi-driver
  name: mountpoint-s3-csi-controller-sa
  namespace: kube-system
  annotations:
    eks.amazonaws.com/role-arn:
 arn:aws:iam::111122223333:role/AmazonEKS_S3_CSI_DriverRole
```

   b. Erstellen Sie das Kubernetes-Servicekonto in Ihrem Cluster. Das Kubernetes-Servicekonto (`mountpoint-s3-csi-controller-sa`) wird mit der von Ihnen erstellten IAM-Rolle namens *AmazonEKS_S3_CSI_DriverRole* annotiert.

```
kubectl apply -f mountpoint-s3-service-account.yaml
```

> **Note**
>
> Wenn Sie das Plugin in diesem Verfahren bereitstellen, wird es erstellt und für die Verwendung eines Servicekontos mit dem Namen `s3-csi-driver-sa` konfiguriert.

Installation des Mountpoint für Amazon S3-CSI-Treibers

Sie können den Mountpoint für Amazon S3-CSI-Treiber über das Amazon-EKS-Add-on installieren. Sie können`eksctl`, das oder das verwenden AWS Management Console, AWS CLI um das Add-on zu Ihrem Cluster hinzuzufügen.

Sie können den Amazon S3 S3-CSI-Treiber optional als selbstverwaltete Installation installierenMountpoint. Anweisungen zur Durchführung einer selbstverwalteten Installation finden Sie im GitHub unter Installation.

eksctl

Um das Amazon S3 CSI-Add-on hinzuzufügen, verwenden Sie **`eksctl`**

Führen Sie den folgenden Befehl aus. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters, *`111122223333`* mit Ihrer Konto-ID und *`AmazonEKS_S3_CSI_DriverRole`* mit dem Namen der zuvor erstellten IAM-Rolle.

```
eksctl create addon --name aws-mountpoint-s3-csi-driver --cluster my-cluster --
service-account-role-arn arn:aws:iam::111122223333:role/AmazonEKS_S3_CSI_DriverRole
 --force
```

Wenn Sie die Option **`--force`** entfernen und eine der Amazon-EKS-Add-on-Einstellungen mit Ihren vorhandenen Einstellungen in Konflikt steht, schlägt die Aktualisierung des Amazon-EKS-Add-ons fehl und Sie erhalten eine Fehlermeldung, die Sie bei der Lösung des Konflikts unterstützt. Stellen Sie vor dem Angeben dieser Option sicher, dass das Amazon-EKS-Add-on keine Einstellungen verwaltet, die Sie verwalten müssen, da diese Einstellungen mit dieser Option überschrieben werden. Weitere Informationen zu anderen Optionen für diese Einstellung finden Sie unter Add-Ons in der `eksctl`-Dokumentation. Weitere Informationen zur Amazon-EKS-Kubernetes-Feldverwaltung finden Sie unter Kubernetes-Feldverwaltung.

AWS Management Console

Um das CSI-Add-on Mountpoint für Amazon S3 hinzuzufügen, verwenden Sie den AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie den Namen des Clusters, für den Sie das CSI-Add-on Mountpoint für Amazon S3 konfigurieren möchten.
4. Wählen Sie die Registerkarte Add-ons.
5. Wählen Sie Weitere Add-Ons erhalten.
6. Führen Sie auf der Seite Add-Ons auswählen die folgenden Schritte aus:

a. Aktivieren Sie im Bereich Amazon EKS-Addons das Kontrollkästchen Mountpointfür Amazon S3 CSI Driver.

b. Wählen Sie Weiter aus.

7. Gehen Sie auf der Seite Konfigurieren ausgewählter Add-Ons-Einstellungen wie folgt vor:

a. Wählen Sie die Version aus, die Sie verwenden möchten.

b. Wählen Sie unter IAM-Rolle auswählen den Namen einer IAM-Rolle aus, der Sie die IAM-Richtlinie Mountpoint für den Amazon S3 S3-CSI-Treiber angehängt haben.

c. (Optional) Sie können Optionale Konfigurationseinstellungen erweitern. Wenn Sie Überschreiben für Methode zur Konfliktlösung auswählen, können eine oder mehrere der Einstellungen für das vorhandene Add-on mit den Einstellungen des Amazon-EKS-Add-ons überschrieben werden. Wenn Sie diese Option nicht aktivieren und ein Konflikt mit Ihren vorhandenen Einstellungen vorliegt, schlägt der Vorgang fehl. Sie können die sich daraus ergebende Fehlermeldung heranziehen, um den Konflikt zu beheben. Stellen Sie vor der Auswahl dieser Option sicher, dass das Amazon-EKS-Add-on keine Einstellungen verwaltet, die Sie selbst verwalten müssen.

d. Wählen Sie Weiter aus.

8. Wählen Sie auf der Seite Überprüfen und hinzufügen die Option Erstellen aus. Nachdem die Installation der Add-Ons abgeschlossen ist, wird Ihr installiertes Add-On angezeigt.

AWS CLI

Um das CSI-Add-on Mountpoint für Amazon S3 hinzuzufügen, verwenden Sie den AWS CLI

Führen Sie den folgenden Befehl aus. Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters, *111122223333* mit Ihrer Konto-ID und *AmazonEKS_S3_CSI_DriverRole* mit dem Namen der zuvor erstellten Rolle.

```
aws eks create-addon --cluster-name my-cluster --addon-name aws-mountpoint-s3-csi-
driver \
  --service-account-role-arn
 arn:aws:iam::111122223333:role/AmazonEKS_S3_CSI_DriverRole
```

Konfiguration von Mountpoint für Amazon S3

In den meisten Fällen können Sie Mountpoint für Amazon S3 mit lediglich einem Bucket-Namen konfigurieren. Anweisungen zur Konfiguration von Mountpoint für Amazon S3 finden Sie im GitHub unter Konfiguration von Mountpoint für Amazon S3.

Bereitstellen einer Beispielanwendung

Sie können die statische Bereitstellung des Treibers in einem vorhandenen Amazon S3-Bucket einsetzen. Weitere Informationen finden Sie im GitHub unter Statische Bereitstellung.

Entfernen des Mountpoint Amazon S3 S3-CSI-Treibers

Beim Entfernen eines Amazon-EKS-Add-ons stehen Ihnen zwei Optionen zur Verfügung.

- Beibehalten von Add-on-Software auf Ihrem Cluster - Diese Option entfernt die Amazon-EKS-Verwaltung aller Einstellungen. Amazon EKS wird außerdem die Möglichkeit aufgehoben, Sie über Updates zu informieren und das Amazon-EKS-Add-on automatisch zu aktualisieren, nachdem Sie ein Update eingeleitet haben. Es behält jedoch die Add-on-Software auf Ihrem Cluster bei. Diese Option macht das Add-on zu einem selbstverwalteten Add-on statt einem Amazon-EKS-Add-on. Bei dieser Option gibt es keine Ausfallzeiten für das Add-on. Die Befehle in diesem Verfahren verwenden diese Option.
- Entfernen Sie die Add-on-Software vollständig aus Ihrem Cluster – Wir empfehlen, dass Sie das Amazon-EKS-Add-on nur aus Ihrem Cluster entfernen, wenn auf Ihrem Cluster keine Ressourcen vorhanden sind, die davon abhängig sind. Um diese Option durchzuführen, löschen Sie `--preserve` aus dem Befehl, den Sie für dieses Verfahren verwenden.

Wenn dem Add-on ein IAM-Konto zugeordnet ist, wird das IAM-Konto nicht entfernt.

Sie können das`eksctl`, oder das verwenden AWS Management Console, AWS CLI um das Amazon S3 CSI-Add-on zu entfernen.

eksctl

Um das Amazon S3 CSI-Add-on zu entfernen, verwenden Sie **`eksctl`**

Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und führen Sie dann den folgenden Befehl aus.

```
eksctl delete addon --cluster my-cluster --name aws-mountpoint-s3-csi-driver --
preserve
```

AWS Management Console

Um das Amazon S3 CSI-Add-on zu entfernen, verwenden Sie den AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie das Amazon-EBS-CSI-Add-on entfernen möchten.
4. Wählen Sie die Registerkarte Add-ons.
5. Wählen MountpointSie Amazon S3 CSI Driver.
6. Wählen Sie Remove (Entfernen) aus.
7. Gehen Sie im Bestätigungsdialogfeld Remove: aws-mountpoint-s 3-csi-driver wie folgt vor:
   a. Wenn Amazon EKS die Verwaltung von Einstellungen für das Add-on einstellen soll, wählen Sie Auf dem Cluster beibehalten. Tun Sie dies, wenn Sie die Add-on-Software auf Ihrem Cluster behalten möchten. Auf diese Weise können Sie alle Einstellungen des Add-ons selbst verwalten.
   b. Geben Sie **`aws-mountpoint-s3-csi-driver`** ein.
   c. Wählen Sie Entfernen aus.

AWS CLI

Um das Amazon S3 CSI-Add-on zu entfernen, verwenden Sie den AWS CLI

Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und führen Sie dann den folgenden Befehl aus.

```
aws eks delete-addon --cluster-name my-cluster --addon-name aws-mountpoint-s3-csi-
driver --preserve
```

CSI-Snapshot-Controller

Der Container Storage Interface (CSI)-Snapshot-Controller ermöglicht die Verwendung der Snapshot-Funktionalität in kompatiblen CSI-Treibern, wie dem Amazon-EBS-CSI-Treiber.

Die folgenden Punkte sind bei der Verwendung des CSI-Snapshot-Controllers zu beachten.

- Der Snapshot-Controller muss zusammen mit einem CSI-Treiber mit Snapshot-Funktion installiert werden. Der Amazon-EBS-CSI-Treiber unterstützt die Erstellung von Amazon-EBS-Snapshots von verwalteten Amazon-EBS-CSI-Volumes. Installationsanweisungen finden Sie unter Amazon-EBS-CSI-Treiber.
- Kubernetes unterstützt keine Snapshots von Volumes, die über die CSI-Migration bereitgestellt werden, wie z. B. Amazon-EBS-Volumes, die eine `StorageClass` mit Provisioner `kubernetes.io/aws-ebs` verwenden. Volumes müssen mit einer `StorageClass` erstellt werden, die auf den CSI-Treiber-Provisioner `ebs.csi.aws.com` verweist. Weitere Informationen zur CSI-Migration finden Sie unter Häufig gestellte Fragen zur Migration des CSI von Amazon EBS.

Wir empfehlen, den CSI-Snapshot über das von Amazon EKS verwaltete Add-on zu installieren. Wenn Sie ein Amazon-EKS-Add-on zu Ihrem Cluster hinzufügen möchten, lesen Sie Erstellen eines Add-Ons. Weitere Informationen zu Add-ons finden Sie unter Amazon-EKS-Add-ons.

Wenn Sie alternativ eine selbstverwaltete Installation des Amazon-EBS-CSI-Snapshot-Controllers wünschen, finden Sie weitere Informationen unter Verwendung im Upstream-Kubernetes-`external-snapshotter` auf GitHub.

Amazon-EKS-Netzwerk

Ihr Amazon-EKS-Cluster wird in einer VPC erstellt. Pod-Networking wird vom Amazon VPC Container Network Interface (CNI)-Plugin bereitgestellt. In diesem Kapitel finden Sie die folgenden Themen, um mehr über Networking für Ihren Cluster zu erfahren.

Themen

Amazon EKS: VPC- und Subnetz-Anforderungen und -Überlegungen

Wenn Sie einen Cluster erstellen, geben Sie eine VPC und mindestens zwei Subnetze an, die sich in unterschiedlichen Availability Zones befinden. Dieses Thema bietet einen Überblick über die spezifischen Amazon-EKS-Anforderungen und Überlegen zur VPC und den Subnetzen, die Sie in Ihrem Cluster verwenden. Wenn Sie keine VPC zur Verwendung mit Amazon EKS haben, können Sie eine mit einer von Amazon EKS bereitgestellten AWS CloudFormation Vorlage erstellen. Informationen zum Erstellen eines lokalen oder erweiterten Clusters finden Sie Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen statt dieses Themas unter. AWS Outposts

VPC-Anforderungen und -Überlegungen

Wenn Sie einen Cluster erstellen, muss die von Ihnen angegebene VPC die folgenden Anforderungen und Überlegungen erfüllen:

- Die VPC muss über eine ausreichende Anzahl von IP-Adressen für den Cluster, alle Knoten und andere Kubernetes-Ressourcen, die Sie erstellen möchten, verfügen. Wenn die VPC, die Sie verwenden möchten, nicht über eine ausreichende Anzahl von IP-Adressen verfügt, versuchen Sie, die Anzahl der verfügbaren IP-Adressen zu erhöhen.

Sie können dies tun, indem Sie die Clusterkonfiguration aktualisieren, um zu ändern, welche Subnetze und Sicherheitsgruppen der Cluster verwendet. Sie können von der AWS Management Console, der neuesten Version der AWS CLI AWS CloudFormation, und Version `v0.164.0-rc.0` oder einer späteren `eksctl` Version aus aktualisieren. Möglicherweise müssen Sie dies tun, um Subnetzen mehr verfügbare IP-Adressen zur Verfügung zu stellen, damit eine Clusterversion erfolgreich aktualisiert werden kann.

> **Important**
>
> Alle Subnetze, die Sie hinzufügen, müssen sich in derselben Gruppe von AZs wie ursprünglich beim Erstellen des Clusters befinden. Neue Subnetze müssen alle anderen Anforderungen erfüllen, z. B. müssen sie über ausreichend IP-Adressen verfügen. Angenommen, Sie haben einen Cluster erstellt und vier Subnetze angegeben. In der Reihenfolge, in der Sie sie angegeben haben, befindet sich das erste Subnetz in der `us-west-2a` Availability Zone, das zweite und dritte Subnetz in der `us-west-2b` Availability Zone und das vierte Subnetz in der `us-west-2c` Availability Zone. Wenn Sie die Subnetze ändern möchten, müssen Sie in jeder der drei Availability Zones mindestens ein Subnetz bereitstellen, und die Subnetze müssen sich in derselben VPC wie die ursprünglichen Subnetze befinden.

Wenn Sie mehr IP-Adressen benötigen, als die CIDR-Blöcke in der VPC haben, können Sie zusätzliche CIDR-Blöcke hinzufügen, indem Sie Ihrer VPC zusätzliche Classless Inter-Domain Routing (CIDR)-Blöcke zuordnen. Sie können entweder vor oder nach der Erstellung Ihres Clusters private (RFC 1918) und öffentliche (nicht RFC 1918) CIDR-Blöcke mit Ihrer VPC verbinden. Es kann bis zu fünf Stunden dauern, bis ein Cluster einen CIDR-Block, den Sie mit einer VPC verbunden haben, erkennt.

Sie können bei der Verwendung von IP-Adressen sparen, indem Sie ein Transit-Gateway mit einer Shared-Services-VPC verwenden. Weitere Informationen finden Sie unter Isolierte VPCs mit gemeinsam genutzten Services und Amazon-EKS-VPC-Routable-IP-Adressspeicherungsmuster in einem Hybridnetzwerk.

- Wenn Sie möchten, dass Kubernetes `IPv6`-Adressen zu Pods und Services zuweist, ordnen Sie Ihrer VPC einen `IPv6`-CIDR-Block zu. Weitere Informationen finden Sie unter Zuordnen eines `IPv6`-CIDR-Blocks zu Ihrer VPC im Amazon-VPC-Benutzerhandbuch.

- Die VPC muss DNS-Hostnamen und die DNS-Auflösung unterstützen. Andernfalls können keine Knoten dem Cluster beitreten. Weitere Informationen finden Sie unter DNS-Attribute für Ihre VPC im Amazon-VPC-Benutzerhandbuch.
- Die VPC erfordert möglicherweise die Verwendung von VPC-Endpunkten. AWS PrivateLink Weitere Informationen finden Sie unter Subnetz-Anforderungen und -Überlegungen.

Wenn Sie einen Cluster mit Kubernetes `1.14` oder früher erstellt haben, hat Amazon EKS das folgende Tag zu Ihrer VPC hinzugefügt:

Schlüssel	Wert
`kubernetes.io/cluster/` *`my-cluster`*	`owned`

Dieses Tag wurde nur von Amazon EKS verwendet. Sie können das Tag entfernen, ohne Ihre Services zu beeinträchtigen. Es wird nicht mit Clustern der Version `1.15` oder höher verwendet.

Subnetz-Anforderungen und -Überlegungen

Wenn Sie einen Cluster erstellen, erstellt Amazon EKS 2–4 Elastic-Network-Schnittstellen in den von Ihnen angegebenen Subnetzen. Diese Netzwerkschnittstellen ermöglichen die Kommunikation zwischen Ihrem Cluster und Ihrer VPC. Die Netzwerkschnittstellen ermöglichen außerdem Kubernetes-Features wie `kubectl exec` und `kubectl logs`. Jede von Amazon EKS erstellte Netzwerkschnittstelle enthält den Text `Amazon EKS` *`cluster-name`* in ihrer Beschreibung.

Amazon EKS kann Netzwerkschnittstellen in jedem Subnetz erstellen, dass Sie bei der Erstellung eines Clusters angeben. Nach der Erstellung Ihres Clusters können Sie ändern, in welchen Subnetzen Amazon EKS seine Netzwerkschnittstellen erstellt. Wenn Sie die Kubernetes-Version eines Clusters aktualisieren, löscht Amazon EKS die ursprünglichen Netzwerkschnittstellen, die es erstellt hat, und erstellt neue Netzwerkschnittstellen. Diese Netzwerkschnittstellen können in denselben Subnetzen wie die ursprünglichen Netzwerkschnittstellen oder in anderen Subnetzen als die ursprünglichen Netzwerkschnittstellen erstellt werden. Um zu steuern, in welchen Subnetzen Netzwerkschnittstellen erstellt werden, können Sie die Anzahl der von Ihnen angegebenen Subnetze beim Erstellen eines Clusters auf zwei beschränken oder die Subnetze nach der Erstellung des Clusters aktualisieren.

Subnetzanforderungen für Cluster

Die Subnetze, die Sie angeben, wenn Sie einen Cluster erstellen oder aktualisieren, müssen die folgenden Anforderungen erfüllen:

- Die Subnetze müssen jeweils mindestens 6 IP-Adressen zur Verwendung durch Amazon EKS haben. Wir empfehlen jedoch mindestens 16 IP-Adressen.
- Die Subnetze können sich nicht in AWS Outposts, AWS Wavelength oder einer lokalen Zone befinden. AWS Wenn Sie sich in Ihrer VPC befinden, können Sie jedoch selbstverwaltete Knoten und Kubernetes-Ressourcen für diese Subnetz-Typen bereitstellen.
- Die Subnetze können öffentlich oder privat sein. Es wird jedoch empfohlen, wenn möglich private Subnetze anzugeben. Ein öffentliches Subnetz ist ein Subnetz mit einer Routing-Tabelle, die eine Route zu einem Internet-Gateway enthält. Ein privates Subnetz ist ein Subnetz mit einer Routing-Tabelle, die keine Route zu einem Internet-Gateway enthält.
- Die Subnetze dürfen sich nicht in folgenden Availability Zones befinden:

AWS-Region	Name der Region	Unzulässige Availability Zone-IDs
`us-east-1`	USA Ost (Nord-Virginia)	`use1-az3`
`us-west-1`	USA West (Nordkalifornien)	`usw1-az2`
`ca-central-1`	Kanada (Zentral)	`cac1-az3`

Verwendung der IP-Adressfamilie nach Komponenten

Die folgende Tabelle enthält die IP-Adressfamilie, die von jeder Komponente von Amazon EKS verwendet wird. Sie können ein Network Address Translation (NAT) oder ein anderes Kompatibilitätssystem verwenden, um über Quell-IP-Adressen in Familien mit dem "No" Wert für einen Tabelleneintrag eine Verbindung zu diesen Komponenten herzustellen.

Die Funktionalität kann je nach Einstellung IP family (`ipFamily`) des Clusters unterschiedlich sein. Diese Einstellung ändert den Typ der IP-Adressen, die für den CIDR Block verwendet werden, der Kubernetes zugewiesen Services wird. Ein Cluster mit dem Einstellungswert von IPv4 wird als bezeichnetIPv4 cluster, und ein Cluster mit dem Einstellungswert von IPv6 wird als ein IPv6 cluster bezeichnet.

Komponente	**IPv4**nur Adressen	**IPv6**nur Adressen	Dual-Stack-Adressen
Öffentlicher Endpunkt der EKS-API	Ja	Nein	Nein
EKS-API-VPC-Endpunkt	Ja	Nein	Nein
Öffentlicher Endpunkt der EKS Auth API	Ja[1]	Ja[1]	Ja[1]
VPC-Endpunkt der EKS-Authentifizierungs-API	Ja[1]	Ja[1]	Ja[1]
Öffentlicher EKS-Cluster-Endpunkt	Ja	Nein	Nein
Privater Endpunkt des EKS-Clusters	Ja[2]	Ja[2]	Nein
EKS-Cluster-Subnetze	Ja[2]	Nein	Ja[2]
Primäre IP-Adressen der Knoten	Ja[2]	Nein	Ja[2]
CIDRClusterbereich für Service IP-Adressen	Ja[2]	Ja[2]	Nein
PodIP-Adressen von der VPC CNI	Ja[2]	Ja[2]	Nein

> **Note**
>
> [1] Der Endpunkt ist ein Dual-Stack mit beiden Adressen`IPv4`. `IPv6` Ihre Anwendungen außerhalb AWS, Ihre Knoten für den Cluster und Ihre Pods innerhalb des Clusters können diesen Endpunkt entweder mit `IPv4` oder erreichen`IPv6`.
>
> [2] Wenn Sie einen `IPv4` Cluster erstellen, wählen Sie in der Einstellung IP family (`ipFamily`) des Clusters zwischen einem Cluster und einem Cluster. Diese Einstellung kann nicht geändert werden. `IPv6` Stattdessen müssen Sie eine andere Einstellung wählen, wenn Sie einen weiteren Cluster erstellen und Ihre Workloads migrieren.

Subnetzanforderungen für Knoten

Sie können Knoten und Kubernetes-Ressourcen in denselben Subnetzen bereitstellen, die Sie beim Erstellen Ihres Clusters angeben. Das ist aber nicht unbedingt notwendig. Sie können Knoten und Kubernetes-Ressourcen auch in Subnetzen bereitstellen, die Sie nicht beim Erstellen des Clusters angegeben haben. Wenn Sie Knoten in verschiedenen Subnetzen bereitstellen, erstellt Amazon EKS keine Cluster-Netzwerkschnittstellen in diesen Subnetzen. Jedes Subnetz, für das Sie Knoten und Kubernetes-Ressourcen bereitstellen, muss die folgenden Anforderungen erfüllen:

- Die Subnetze müssen über genügend verfügbare IP-Adressen verfügen, um alle Ihre Knoten und Kubernetes-Ressourcen bereitzustellen.
- Wenn Sie möchten, dass Kubernetes Pods und Services `IPv6`-Adressen zuweist, benötigen Sie einen `IPv6`-CIDR-Block und einen `IPv4`-CIDR-Block, die Ihrem Subnetz zugeordnet sind. Weitere Informationen finden Sie unter Zuordnen eines `IPv6`-CIDR-Blocks zu Ihrem Subnetz im Amazon-VPC-Benutzerhandbuch. Die Routing-Tabellen, die Ihren Subnetzen zugeordnet sind, müssen Routen zu `IPv4`- und `IPv6`-Adressen enthalten. Weitere Informationen finden Sie unter Routen im Amazon-VPC-Benutzerhandbuch. Pods werden nur einer `IPv6`-Adresse zugewiesen. Die Netzwerkschnittstellen, die Amazon EKS für Ihren Cluster und Ihre Knoten erstellt, werden jedoch einer `IPv4`- und einer `IPv6`-Adresse zugewiesen.
- Wenn Sie eingehenden Zugriff aus dem Internet auf Ihre Pods benötigen, stellen Sie sicher, dass Sie mindestens ein öffentliches Subnetz mit genügend verfügbaren IP-Adressen haben, in denen Sie Load Balancer und Ingresse bereitstellen können. Sie können Load Balancer in öffentlichen Subnetzen bereitstellen. Load Balancer können ein Load Balancing zu Pods in privaten und öffentlichen Subnetzen vornehmen. Wir empfehlen, Ihre Knoten nach Möglichkeit in privaten Subnetzen bereitzustellen.

- Wenn Sie planen, Knoten in einem öffentlichen Subnetz bereitzustellen, muss das Subnetz automatisch öffentliche `IPv4`- oder `IPv6`-Adressen zuweisen. Wenn Sie Knoten in einem privaten Subnetz bereitstellen, das einen zugeordneten `IPv6`-CIDR-Block hat, muss das private Subnetz zudem `IPv6`-Adressen automatisch zuweisen. Wenn Sie nach dem 26. März 2020 eine Amazon AWS CloudFormation EKS-Vorlage für die Bereitstellung Ihrer VPC verwendet haben, ist diese Einstellung aktiviert. Wenn Sie die Vorlagen verwendet haben, um Ihre VPC vor diesem Datum bereitzustellen oder Ihre eigene VPC verwenden, müssen Sie diese Einstellung manuell aktivieren. Weitere Informationen finden Sie unter Ändern des `IPv4`-Adressierungsattributs Ihres Subnetzes und Ändern des `IPv6`-Adressierungsattributs Ihres Subnetzes im Amazon-VPC-Benutzerhandbuch.
- Wenn das Subnetz, in dem Sie einen Knoten bereitstellen, ein privates Subnetz ist und seine Routing-Tabelle keine Route zu einem NAT-Gerät (Network Address Translation) (`IPv4`) oder einem Gateway für ausgehenden Datenverkehr (`IPv6`) enthält, fügen Sie mit AWS PrivateLink VPC-Endpunkte zu Ihrer VPC hinzu. VPC-Endpunkte werden für all AWS-Services das benötigt, was Ihre Knoten und mit denen Sie kommunizieren Pods müssen. Beispiele hierfür sind Amazon ECR, Elastic Load Balancing CloudWatch AWS Security Token Service, Amazon und Amazon Simple Storage Service (Amazon S3). Der Endpunkt muss das Subnetz enthalten, in dem sich die Knoten befinden. Nicht alle AWS-Services unterstützen VPC-Endpunkte. Weitere Informationen finden Sie unter Was ist? AWS PrivateLink und AWS Dienste, die sich in integrieren lassen AWS PrivateLink. Eine Liste weiterer Amazon-EKS-Anforderungen finden Sie unter Anforderungen an private Cluster.
- Wenn Sie Load Balancer in einem Subnetz bereitstellen möchten, muss das Subnetz das folgende Tag haben:
  - Private Subnetze

Schlüssel	Wert
`kubernetes.io/role/internal-elb`	`1`

  - Öffentliche Subnetze

Schlüssel	Wert
`kubernetes.io/role/elb`	`1`

Wenn ein Kubernetes-Cluster mit der Version `1.18` oder früher erstellt wurde, fügt Amazon EKS das folgende Tag zu allen Subnetzen hinzu, die angegeben wurden.

Schlüssel	Wert
`kubernetes.io/cluster/` *`my-cluster`*	`shared`

Wenn Sie jetzt einen neuen Kubernetes-Cluster erstellen, fügt Amazon EKS das Tag nicht zu Ihren Subnetzen hinzu. Wenn sich das Tag in Subnetzen befand, die von einem Cluster verwendet wurden, der zuvor eine Version vor `1.19` war, wurde das Tag nicht automatisch aus den Subnetzen entfernt, als der Cluster auf eine neuere Version aktualisiert wurde. Version `2.1.1` oder früher des AWS Load Balancer Controller benötigt dieses Tag. Wenn Sie eine neuere Version des Load Balancer Controllers verwenden, können Sie das Tag entfernen, ohne Ihre Services zu unterbrechen.

Wenn Sie eine VPC mithilfe einer `eksctl`oder einer der Amazon AWS CloudFormation EKS-VPC-Vorlagen bereitgestellt haben, gilt Folgendes:

- Am oder nach dem 26. März 2020 – Öffentliche `IPv4`-Adressen werden von öffentlichen Subnetzen automatisch neuen Knoten zugewiesen, die in öffentlichen Subnetzen bereitgestellt werden.
- Vor dem 26. März 2020 – Öffentliche `IPv4`-Adressen werden von öffentlichen Subnetzen nicht automatisch neuen Knoten zugewiesen, die in öffentlichen Subnetzen bereitgestellt werden.

Diese Änderung wirkt sich folgendermaßen auf neue Knotengruppen aus, die in öffentlichen Subnetzen bereitgestellt werden:

- Verwaltete Knotengruppen – Wenn die Knotengruppe am oder nach dem 22. April 2020 in einem öffentlichen Subnetz bereitgestellt wird, muss für das öffentliche Subnetz die automatische Zuweisung öffentlicher IP-Adressen aktiviert sein. Weitere Informationen finden Sie unter Ändern des öffentlichen `IPv4`-Adressierungsattributs für Ihr Subnetz.
- Selbstverwaltete Linux, Windows- oder Arm-Knotengruppen – Wenn die Knotengruppe am oder nach dem 26. März 2020 in einem öffentlichen Subnetz bereitgestellt wird, muss für das öffentliche Subnetz die automatische Zuweisung öffentlicher IP-Adressen aktiviert sein. Andernfalls müssen die Knoten mit einer öffentlichen IP-Adresse gestartet werden. Weitere Informationen finden Sie unter Ändern des öffentlichen `IPv4`-Adressattributs für Ihr Subnetz oder Zuweisen einer öffentlichen `IPv4`-Adresse während des Instance-Starts.

Anforderungen und Überlegungen für gemeinsam genutzte Subnetze

Sie können VPC-Freigabe nutzen, um Subnetze mit anderen AWS -Konten innerhalb derselben AWS Organizations zu teilen. Sie können Amazon-EKS-Cluster in gemeinsam genutzten Subnetzen erstellen. Beachten Sie dabei die folgenden Überlegungen:

- Der Eigentümer des VPC-Subnetzes muss ein Subnetz für ein Teilnehmerkonto freigeben, bevor dieses Konto darin einen Amazon-EKS-Cluster erstellen kann.
- Sie können keine Ressourcen mit der Standardsicherheitsgruppe für die VPC starten, da diese dem Eigentümer gehört. Außerdem können Teilnehmer keine Ressourcen mit Sicherheitsgruppen starten, die anderen Teilnehmern oder dem Eigentümer gehören.
- In einem gemeinsam genutzten Subnetz kontrollieren der Teilnehmer und der Eigentümer die Sicherheitsgruppen innerhalb des jeweiligen Kontos separat. Der Subnetzbesitzer kann von den Teilnehmern erstellte Sicherheitsgruppen zwar anzeigen, jedoch keine Aktionen bei diesen durchführen. Wenn der Subnetzbesitzer diese Sicherheitsgruppen entfernen oder ändern möchte, muss der Teilnehmer, der die Sicherheitsgruppe erstellt hat, die Aktion durchführen.
- Wenn ein Cluster von einem Teilnehmer erstellt wird, gelten die folgenden Überlegungen:
  - Die Cluster-IAM-Rolle und die Knoten-IAM-Rollen müssen in diesem Konto erstellt werden. Weitere Informationen finden Sie unter Amazon-EKS-Cluster-IAM-Rolle und Amazon-EKS-Knoten-IAM-Rolle.
  - Alle Knoten müssen vom selben Teilnehmer erstellt werden, einschließlich verwalteter Knotengruppen.
- Der Eigentümer einer gemeinsam genutzten VPC kann einen Cluster, den ein Teilnehmer im gemeinsam genutzten Subnetz erstellt, nicht anzeigen, aktualisieren oder löschen. Dies gilt zusätzlich zu den VPC-Ressourcen, auf die jedes Konto unterschiedlich zugreifen kann. Weitere Informationen finden Sie unter Verantwortlichkeiten und Berechtigungen für Besitzer und Teilnehmer im Amazon-VPC-Benutzerhandbuch.
- Wenn Sie das Feature benutzerdefinierte Netzwerke des Amazon VPC CNI plugin for Kubernetes verwenden, müssen Sie die im Eigentümerkonto aufgeführten ID-Zuordnungen der Availability Zone verwenden, um jede `ENIConfig` zu erstellen. Weitere Informationen finden Sie unter Benutzerdefinierte Netzwerke für Pods.

Weitere Informationen zur Freigabe von VPC-Subnetzen finden Sie unter Freigeben Ihrer VPC für andere Konten im Amazon-VPC-Benutzerhandbuch.

Erstellen einer VPC für Ihren Amazon-EKS-Cluster

Sie können Amazon Virtual Private Cloud (Amazon VPC) verwenden, um AWS Ressourcen in einem von Ihnen definierten virtuellen Netzwerk zu starten. Dieses virtuelle Netzwerk ist einem herkömmlichen Netzwerk, das Sie in Ihrem eigenen Rechenzentrum betreiben, sehr ähnlich. Es bietet jedoch die Vorzüge, die mit der Nutzung der skalierbaren Infrastruktur von Amazon Web Services einhergehen. Bevor Sie sich ein umfassendes Verständnis des Amazon-VPC-Service aneignet, empfehlen wir, dass Sie sich ein umfassendes Verständnis des Amazon-VPC-Service aneignet, bevor Sie sich ein umfassendes Verständnis aneignet. Weitere Informationen finden Sie im Amazon VPC-Benutzerhandbuch.

Ein Amazon-EKS-Cluster, Knoten und Kubernetes-Ressourcen werden auf einer VPC bereitgestellt. Wenn Sie eine vorhandene VPC mit Amazon EKS verwenden möchten, muss diese VPC die unter Amazon EKS: VPC- und Subnetz-Anforderungen und -Überlegungen beschriebenen Voraussetzungen erfüllen. In diesem Thema wird beschrieben, wie Sie mithilfe einer von Amazon EKS bereitgestellten AWS CloudFormation Vorlage eine VPC erstellen, die die Amazon EKS-Anforderungen erfüllt. Sobald Sie eine Vorlage bereitgestellt haben, können Sie sich die mit der Vorlage erstellten Ressourcen ansehen, um genau zu erfahren, welche Ressourcen sie erstellt hat und wie diese Ressourcen konfiguriert sind.

Voraussetzung

Um eine VPC für Amazon EKS zu erstellen, benötigen Sie die erforderlichen IAM-Berechtigungen, um Amazon-VPC-Ressourcen zu erstellen. Diese Ressourcen sind VPCs, Subnetze, Sicherheitsgruppen, Routing-Tabellen und Routen sowie Internet- und NAT-Gateways. Weitere Informationen finden Sie unter Erstellen einer VPC mit einer öffentlichen Subnetz-Beispielrichtlinie im Amazon-VPC-Benutzerhandbuch und in der vollständigen Liste von Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EC2 in der Service-Authorization-Referenz.

Sie können eine VPC mit öffentlichen und privaten Subnetzen, nur öffentlichen Subnetzen oder nur privaten Subnetzen erstellen.

Public and private subnets

> Diese VPC verfügt über zwei öffentliche und zwei private Subnetze. Eine Routing-Tabelle, die einem öffentlichen Subnetz zugeordnet ist, verfügt über eine Route zu einem Internet-Gateway. Die Routing-Tabelle eines privaten Subnetzes verfügt jedoch nicht über eine Route zu einem Internet-Gateway. Ein öffentliches und ein privates Subnetz werden in derselben Availability Zone

bereitgestellt. Die anderen öffentlichen und privaten Subnetze werden in einer zweiten Availability Zone in derselben AWS-Region bereitgestellt. Diese Option wird für die meisten Bereitstellungen empfohlen.

Mit dieser Option können Sie Ihre Knoten in privaten Subnetzen bereitstellen. Mit dieser Option können Sie es Kubernetes ermöglichen, Load Balancer in den öffentlichen Subnetzen bereitzustellen, die das Load Balancing für den Datenverkehr auf Pods, die auf Knoten in den privaten Subnetzen ausgeführt werden, durchführen können. Öffentliche `IPv4`-Adressen werden automatisch den Knoten zugewiesen, die in öffentlichen Subnetzen bereitgestellt werden, öffentliche `IPv4`-Adressen werden jedoch nicht Knoten zugewiesen, die in privaten Subnetzen bereitgestellt werden.

Sie können Knoten in öffentlichen und privaten Subnetzen auch `IPv6`-Adressen zuweisen. Die Knoten in privaten Subnetzen können mit dem Cluster und anderen AWS-Services-Services kommunizieren. Pods können über ein NAT-Gateway mit `IPv4`-Adressen oder ein Internet-Gateway für ausgehenden Traffic mit `IPv6`-Adressen, das in jeder Availability Zone bereitgestellt wird, mit dem Internet kommunizieren. Es wird eine Sicherheitsgruppe bereitgestellt, die über Regeln verfügt, die den gesamten eingehenden Datenverkehr von anderen Quellen als dem Cluster oder den Knoten ablehnen, jedoch den gesamten ausgehenden Datenverkehr zulassen. Die Subnetze sind getaggt, damit Kubernetes ihnen Load Balancer bereitstellen kann.

So erstellen Sie Ihre VPC

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste ein Gerät aus AWS-Region , das Amazon EKS unterstützt.
3. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
4. Stellen Sie unter Vorraussetzung – Vorbereiten der Vorlage sicher, dass Vorlage ist bereit ausgewählt ist und wählen Sie dann unter Vorlage angeben Amazon-S3-URL aus.
5. Sie können eine VPC erstellen, die nur `IPv4` unterstützt, oder eine VPC, die `IPv4` und `IPv6` unterstützt. Fügen Sie eine der folgenden URLs in den Textbereich unter Amazon-S3-URL ein und wählen Sie Next (Weiter) aus:

   - `IPv4`

```
https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/
amazon-eks-vpc-private-subnets.yaml
```

- IPv4 und IPv6

```
https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/
amazon-eks-ipv6-vpc-public-private-subnets.yaml
```

6. Ändern Sie auf der Seite Specify stack details (Stack-Details angeben) alle Parameter und wählen Sie dann Next (Weiter) aus.

   - Stack name (Stack-Name): Wählen Sie einen Stack-Namen für Ihren AWS CloudFormation -Stack aus. Sie können beispielsweise den im vorigen Schritt verwendeten Vorlagennamen nutzen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
   - VpcBlock: Wählen Sie einen IPv4 CIDR-Bereich für Ihre VPC. Jedem von Ihnen bereitgestellten Knoten, Pod und jedem Load Balancer wird eine IPv4-Adresse aus diesem Block zugewiesen. Die IPv4-Standardwerte bieten für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern. Weitere Informationen finden Sie unter Dimensionierung der VPC und der Subnetze im Amazon VPC Benutzerhandbuch. Sie können der VPC nach der Erstellung auch zusätzliche CIDR-Blöcke hinzufügen. Wenn Sie eine IPv6-VPC erstellen, werden Ihnen IPv6-CIDR-Bereiche automatisch aus dem globalen Unicast-Adressraum von Amazon zugewiesen.
   - PublicSubnet01Block: Geben Sie einen IPv4 CIDR-Block für das öffentliche Subnetz 1 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern. Wenn Sie eine IPv6-VPC erstellen, wird dieser Block in der Vorlage für Sie angegeben.
   - PublicSubnet02Block: Geben Sie einen IPv4 CIDR-Block für das öffentliche Subnetz 2 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern. Wenn Sie eine IPv6-VPC erstellen, wird dieser Block in der Vorlage für Sie angegeben.

- PrivateSubnet01Block: Geben Sie einen `IPv4` CIDR-Block für das private Subnetz 1 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern. Wenn Sie eine `IPv6`-VPC erstellen, wird dieser Block in der Vorlage für Sie angegeben.
- PrivateSubnet02Block: Geben Sie einen `IPv4` CIDR-Block für das private Subnetz 2 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern. Wenn Sie eine `IPv6`-VPC erstellen, wird dieser Block in der Vorlage für Sie angegeben.

7. (Optional) Auf der Seite Configure stack options (Stack-Optionen konfigurieren) können Sie Ihre Stack-Ressourcen markieren und dann Next (Weiter) auswählen.
8. Wählen Sie auf der Seite Prüfen Stack erstellen aus.
9. Wenn Ihr Stack erstellt wurde, wählen Sie ihn in der Konsole aus und klicken Sie auf Outputs (Ausgänge).
10. Notieren Sie das VpcIdfür die VPC, die erstellt wurde. Dies benötigen Sie, wenn Sie Ihren Cluster und Ihre Knoten erstellen.
11. Notieren Sie SubnetIdsfür die Subnetze, die erstellt wurden, und ob Sie sie als öffentliche oder private Subnetze erstellt haben. Sie benötigen mindestens zwei davon, wenn Sie Ihren Cluster und Ihre Knoten erstellen.
12. Wenn Sie eine `IPv4`-VPC erstellt haben, überspringen Sie diesen Schritt. Wenn Sie eine `IPv6`-VPC erstellt haben, müssen Sie die Option zum automatischen Zuweisen von `IPv6`-Adressen für die öffentlichen Subnetze aktivieren, die von der Vorlage erstellt wurden. Diese Einstellung ist für die privaten Subnetze bereits aktiviert. Führen Sie die folgenden Schritte aus, um die Einstellung zu aktivieren:

    a. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.

    b. Wählen Sie im linken Navigationsbereich die Option Subnets (Subnetze) aus.

    c. Wählen Sie eines Ihrer öffentlichen Subnetze aus (***stack-name SubnetPublic /01 oder stack-name SubnetPublic*** /02 enthält das Wort public) und wählen Sie Aktionen, Subnetzeinstellungen bearbeiten.

    d. Aktivieren Sie das Kontrollkästchen Enable auto-assign **`IPv6`** address (Automatische Zuweisung der -Adresse aktivieren) und wählen Sie dann Save (Speichern) aus.

    e. Wiederholen Sie die vorherigen Schritte für Ihr anderes öffentliches Subnetz.

Only public subnets

Diese VPC verfügt über drei öffentliche Subnetze, die in verschiedenen Availability Zones in einer AWS-Region bereitgestellt werden. Allen Knoten werden automatisch öffentliche `IPv4`-Adressen zugewiesen und alle Worker-Knoten können Internetdatenverkehr über ein Internet-Gateway senden und empfangen. Es wird eine Sicherheitsgruppe bereitgestellt, die den gesamten eingehenden Datenverkehr ablehnt und den gesamten ausgehenden Datenverkehr zulässt. Die Subnetze sind getaggt, damit Kubernetes ihnen Load Balancer bereitstellen kann.

So erstellen Sie Ihre VPC

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudformation. AWS CloudFormation
2. Wählen Sie in der Navigationsleiste ein Gerät aus AWS-Region , das Amazon EKS unterstützt.
3. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
4. Stellen Sie unter Vorbereiten der Vorlage sicher, dass Vorlage ist bereit ausgewählt ist und wählen Sie dann unterVorlagenquelle Amazon-S3-URLaus.
5. Fügen Sie die folgende URL in den Textbereich unter Amazon-S3-URL ein und wählen Sie Weiter aus:

   ```
   https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-sample.yaml
   ```

6. Füllen Sie auf der Seite Specify Details (Details angeben) die Parameter entsprechend aus und klicken Sie dann auf Next (Weiter).

   - Stack name (Stack-Name): Wählen Sie einen Stack-Namen für Ihren AWS CloudFormation -Stack aus. Sie können ihn beispielsweise ***`amazon-eks-vpc-sample`*** nennen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
   - VpcBlock: Wählen Sie einen CIDR-Block für Ihre VPC. Jedem von Ihnen bereitgestellten Knoten, Pod und jedem Load Balancer wird eine `IPv4`-Adresse aus diesem Block zugewiesen. Die `IPv4`-Standardwerte bieten für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern. Weitere

Informationen finden Sie unter Dimensionierung der VPC und der Subnetze im Amazon VPC Benutzerhandbuch. Sie können der VPC nach der Erstellung auch zusätzliche CIDR-Blöcke hinzufügen.

- Subnet01Block: Geben Sie einen CIDR-Block für das Subnetz 1 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern.
- Subnet02Block: Geben Sie einen CIDR-Block für das Subnetz 2 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern.
- Subnet03Block: Geben Sie einen CIDR-Block für das Subnetz 3 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern.

7. (Optional) Markieren Sie auf der Seite Options (Optionen) Ihre Stack-Ressourcen. Wählen Sie Next (Weiter).
8. Klicken Sie auf der Seite Review auf Create.
9. Wenn Ihr Stack erstellt wurde, wählen Sie ihn in der Konsole aus und klicken Sie auf Outputs (Ausgänge).
10. Notieren Sie das VpcIdfür die VPC, die erstellt wurde. Dies benötigen Sie, wenn Sie Ihren Cluster und Ihre Knoten erstellen.
11. Notieren Sie das SubnetIdsfür die Subnetze, die erstellt wurden. Sie benötigen mindestens zwei davon, wenn Sie Ihren Cluster und Ihre Knoten erstellen.
12. (Optional) Jeder Cluster, den Sie für diese VPC bereitstellen, kann Ihren Pods und services private `IPv4`-Adressen zuweisen. Wenn Sie Cluster in dieser VPC bereitstellen möchten, um Ihren Pods und services private `IPv6`-Adressen zuzuweisen, müssen Sie Aktualisierungen an Ihrer VPC, Ihrem Subnetz, Routing-Tabellen und Sicherheitsgruppen vornehmen. Weitere Informationen finden Sie unter Migrieren vorhandener VPCs von `IPv4` zu `IPv6` im Amazon-VPC-Benutzerhandbuch. Amazon EKS verlangt, dass bei Ihren Subnetzen die Option zum `Auto-assign` von `IPv6`-Adressen aktiviert ist. Dies ist standardmäßig deaktiviert.

Only private subnets

Diese VPC verfügt über drei private Subnetze, die in verschiedenen Availability Zones in der AWS-Region bereitgestellt werden. Ressourcen, die in den Subnetzen bereitgestellt werden, können weder auf das Internet zugreifen, noch kann das Internet auf Ressourcen in den Subnetzen zugreifen. Die Vorlage erstellt VPC-Endpoints und verwendet AWS PrivateLink

mehrere AWS-Services , auf die Knoten normalerweise zugreifen müssen. Wenn Ihre Knoten einen ausgehenden Internetzugang benötigen, können Sie ein öffentliches NAT-Gateway in der Availability Zone jedes Subnetzes hinzufügen, nachdem die VPC erstellt wurde. Eine Sicherheitsgruppe wird erstellt, der den gesamten eingehenden Datenverkehr verweigert, mit Ausnahme von Ressourcen, die in den Subnetzen bereitgestellt werden. Eine Sicherheitsgruppe lässt auch den gesamten ausgehenden Datenverkehr zu. Die Subnetze sind getaggt, damit Kubernetes ihnen interne Load Balancer bereitstellen kann. Wenn Sie eine VPC mit dieser Konfiguration erstellen, siehe Anforderungen an private Cluster für weitere Informationen und Gesichtspunkte.

So erstellen Sie Ihre VPC

1. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie in der Navigationsleiste ein Gerät aus AWS-Region , das Amazon EKS unterstützt.
3. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
4. Stellen Sie unter Vorbereiten der Vorlage sicher, dass Vorlage ist bereit ausgewählt ist und wählen Sie dann unterVorlagenquelle Amazon-S3-URLaus.
5. Fügen Sie die folgende URL in den Textbereich unter Amazon-S3-URL ein und wählen Sie Weiter aus:

   ```
   https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-fully-private-vpc.yaml
   ```

6. Füllen Sie auf der Seite Specify Details (Details angeben) die Parameter entsprechend aus und klicken Sie dann auf Next (Weiter).

   - Stack name (Stack-Name): Wählen Sie einen Stack-Namen für Ihren AWS CloudFormation -Stack aus. Sie können ihn beispielsweise ***amazon-eks-fully-private-vpc*** nennen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
   - VpcBlock: Wählen Sie einen CIDR-Block für Ihre VPC. Jedem von Ihnen bereitgestellten Knoten, Pod und jedem Load Balancer wird eine `IPv4`-Adresse aus diesem Block

zugewiesen. Die IPv4-Standardwerte bieten für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern. Weitere Informationen finden Sie unter Dimensionierung der VPC und der Subnetze im Amazon VPC Benutzerhandbuch. Sie können der VPC nach der Erstellung auch zusätzliche CIDR-Blöcke hinzufügen.

- PrivateSubnet01Block: Geben Sie einen CIDR-Block für Subnetz 1 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern.
- PrivateSubnet02Block: Geben Sie einen CIDR-Block für Subnetz 2 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern.
- PrivateSubnet03Block: Geben Sie einen CIDR-Block für Subnetz 3 an. Der Standardwert bietet für die meisten Implementierungen genügend IP-Adressen, aber wenn dies nicht der Fall ist, können Sie ihn ändern.

7. (Optional) Markieren Sie auf der Seite Options (Optionen) Ihre Stack-Ressourcen. Wählen Sie Next (Weiter).
8. Klicken Sie auf der Seite Review auf Create.
9. Wenn Ihr Stack erstellt wurde, wählen Sie ihn in der Konsole aus und klicken Sie auf Outputs (Ausgänge).
10. Notieren Sie das VpcIdfür die VPC, die erstellt wurde. Dies benötigen Sie, wenn Sie Ihren Cluster und Ihre Knoten erstellen.
11. Notieren Sie das SubnetIdsfür die Subnetze, die erstellt wurden. Sie benötigen mindestens zwei davon, wenn Sie Ihren Cluster und Ihre Knoten erstellen.
12. (Optional) Jeder Cluster, den Sie für diese VPC bereitstellen, kann Ihren Pods und services private IPv4-Adressen zuweisen. Wenn Sie Cluster in dieser VPC bereitstellen möchten, um Ihren Pods und services private IPv6-Adressen zuzuweisen, müssen Sie Aktualisierungen an Ihrer VPC, Ihrem Subnetz, Routing-Tabellen und Sicherheitsgruppen vornehmen. Weitere Informationen finden Sie unter Migrieren vorhandener VPCs von IPv4 zu IPv6 im Amazon-VPC-Benutzerhandbuch. Amazon EKS verlangt, dass bei Ihren Subnetzen die Option zum Auto-assign von IPv6-Adressen aktiviert ist (sie ist standardmäßig deaktiviert).

Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe

In diesem Thema geht es um die Anforderungen an Sicherheitsgruppen in einem Amazon-EKS-Cluster.

Wenn Sie einen Cluster anlegen, erstellt Amazon EKS eine Sicherheitsgruppe mit dem Namen `eks-cluster-sg-`*`my-cluster`*`-`*`uniqueID`*. Diese Sicherheitsgruppe hat die folgenden Standardregeln:

Regeltyp	Protocol (Protokoll)	Ports	Quelle	Ziel
Eingehend	Alle	Alle	Selbst	
Ausgehend	Alle	Alle		0.0.0.0/0 (IPv4) oder ::/0 (IPv6)

> **Important**
>
> Wenn Ihr Cluster die ausgehende Regel nicht benötigt, können Sie sie entfernen. Wenn Sie es entfernen, müssen Sie immer noch über die Mindestregeln verfügen, die unter Einschränkung des Cluster-Datenverkehrs aufgeführt sind. Wenn Sie die eingehende Regel entfernen, wird sie von Amazon EKS bei jeder Aktualisierung des Clusters neu erstellt.

Amazon EKS fügt der Sicherheitsgruppe die folgenden Tags hinzu. Wenn Sie die Tags entfernen, fügt Amazon EKS sie bei jeder Aktualisierung Ihres Clusters wieder zur Sicherheitsgruppe hinzu.

Schlüssel	Wert
`kubernetes.io/cluster/` *`my-cluster`*	`owned`
`aws:eks:cluster-name`	*`my-cluster`*
`Name`	`eks-cluster-sg-` *`my-cluster`* *`-uniqueid`*

Amazon EKS verknüpft die Sicherheitsgruppe automatisch mit den folgenden Ressourcen, die es ebenfalls erstellt:

- 2 bis 4 Elastic-Network-Schnittstellen (für den Rest dieses Dokuments als Netzwerkschnittstelle bezeichnet), die beim Anlegen des Clusters erstellt werden.
- Netzwerkschnittstellen der Knoten in jeder verwalteten Knotengruppe, die Sie erstellen.

Die Standardregeln ermöglichen es, dass der gesamte Datenverkehr frei zwischen Ihrem Cluster und Ihren Knoten fließt. Außerdem lassen sie den gesamten ausgehenden Datenverkehr zu jedem Ziel zu. Beim Erstellen eines Clusters können Sie (optional) Ihre eigenen Sicherheitsgruppen angeben. Wenn Sie das tun, verknüpft Amazon EKS auch die von Ihnen angegebenen Sicherheitsgruppen mit den Netzwerkschnittstellen, die es für Ihren Cluster erstellt. Es verknüpft sie jedoch nicht mit Knotengruppen, die Sie erstellen.

Sie können die ID Ihrer Cluster-Sicherheitsgruppe in der AWS Management Console im Abschnitt Networking (Netzwerk) für Ihren Cluster ermitteln. Sie können auch den folgenden AWS CLI-Befehl ausführen.

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

Einschränken des Clusterverkehrs

Wenn Sie die offenen Ports zwischen dem Cluster und den Knoten einschränken müssen, können Sie die standardmäßige Ausgangsregel entfernen und die folgenden Mindestregeln hinzufügen, die für den Cluster erforderlich sind. Wenn Sie die standardmäßige Eingangsregel entfernen, erstellt Amazon EKS sie bei jeder Aktualisierung des Clusters neu.

Regeltyp	Protocol (Protokoll)	Port	Bestimmungsort
Ausgehend	TCP	443	Cluster-Sicherheitsgruppe
Ausgehend	TCP	10250	Cluster-Sicherheitsgruppe
Ausgehend (DNS)	TCP und UDP	53	Cluster-Sicherheitsgruppe

Sie müssen auch Regeln für den folgenden Datenverkehr hinzufügen:

- Jedes Protokoll und alle Ports, von dem Sie erwarten, dass Ihre Knoten für die Kommunikation zwischen Knoten verwenden
- Ausgehenden Internetzugang, sodass die Knoten zwecks Cluster-Introspektion und Knotenregistrierung beim Start auf die Amazon-EKS-APIs zugreifen können. Wenn Ihre Knoten keinen Internetzugang haben, lesen Sie Anforderungen an private Cluster für weitere Hinweise.
- Zugriff auf Knoten, um Container-Images aus Amazon ECR abzurufen, oder andere Container-Registry-APIs, die zum Laden von Images benötigt werden, wie z. B. DockerHub. Weitere Informationen finden Sie unter AWS IP-Adressbereiche im Allgemeine AWS-Referenz.
- Knotenzugriff auf Amazon S3.
- Für `IPv4`- und `IPv6`-Adressen sind separate Regeln erforderlich

Wenn Sie Regelbeschränkungen planen, empfehlen wir, alle Pods gründlich zu testen, bevor Sie Ihre geänderten Regeln auf einen Produktionscluster anwenden.

Wenn Sie ursprünglich einen Cluster mit Kubernetes `1.14` und der Plattformversion `eks.3` oder älter bereitgestellt haben, dann bedenken Sie Folgendes:

- Möglicherweise gibt es Sicherheitsgruppen für die Steuerebene und Knoten. Als diese Gruppen erstellt wurden, enthielten sie die in der vorherigen Tabelle aufgeführten eingeschränkten Regeln. Diese Sicherheitsgruppen sind nicht mehr erforderlich und können entfernt werden. Sie müssen jedoch sicherstellen, dass Ihre Cluster-Sicherheitsgruppe die Regeln enthält, die diese Gruppen enthalten.
- Wenn Sie den Cluster direkt über die API bereitgestellt haben oder die AWS CLI bzw. AWS CloudFormation zur Clustererstellung verwendet und dabei keine Sicherheitsgruppe angegeben haben, wurde die Standardsicherheitsgruppe für die VPC auf die Cluster-Netzwerkschnittstellen angewendet, die Amazon EKS erstellt hat.

Netzwerk-Add-ons für Amazon EKS

Für Ihren Amazon-EKS-Cluster sind mehrere Netzwerk-Add-Ons verfügbar.

Integrierte Add-Ons

> **Note**
>
> Wenn Sie Cluster auf irgendeine Weise erstellen, außer mithilfe der Konsole, enthält jeder Cluster die selbstverwalteten Versionen der integrierten Add-Ons. Die selbstverwalteten Versionen können nicht über die AWS Management Console AWS Command Line Interface, oder SDKs verwaltet werden. Sie verwalten die Konfiguration und Upgrades von selbstverwalteten Add-Ons.
> Wir empfehlen, den Amazon-EKS-Typ des Add-Ons zu Ihrem Cluster hinzuzufügen, anstatt den selbstverwalteten Typ des Add-Ons zu verwenden. Wenn Sie Cluster in der Konsole erstellen, wird der Amazon-EKS-Typ dieser Add-ons installiert.

Amazon VPC CNI plugin for Kubernetes

Dieses CNI-Add-on erstellt elastische Netzwerkschnittstellen und fügt sie an Ihre Amazon-EC2-Knoten an. Das Add-On weist jedem Pod und Service auch eine private `IPv4`- oder `IPv6`-Adresse von Ihrer VPC zu. Dieses Add-on ist standardmäßig auf Ihrem Cluster installiert. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

CoreDNS

CoreDNS ist ein flexibler, erweiterbarer DNS-Server, der als Kubernetes-Cluster-DNS dienen kann. CoreDNS bietet die Namensauflösung für alle Pods im Cluster. Dieses Add-on ist standardmäßig auf Ihrem Cluster installiert. Weitere Informationen finden Sie unter Arbeiten mit dem CoreDNS-Amazon-EKS-Add-on.

`kube-proxy`

Dieses Add-on verwaltet Netzwerkregeln auf Ihren Amazon-EC2-Knoten und ermöglicht die Netzwerkkommunikation mit Ihren Pods. Dieses Add-on ist standardmäßig auf Ihrem Cluster installiert. Weitere Informationen finden Sie unter Arbeiten mit dem `kube-proxy` Kubernetes-Add-on.

Optionale Netzwerk-Add-Ons AWS

AWS Load Balancer Controller

Wenn Sie Kubernetes Dienstobjekte dieses Typs bereitstellen`loadbalancer`, erstellt der Controller AWS Network Load Balancer. Wenn Sie Kubernetes Eingangsobjekte erstellen, erstellt der Controller AWS Application Load Balancer. Wir empfehlen, diesen Controller für die Bereitstellung von Network Load Balancern zu verwenden, anstatt den in Kubernetes integrierten Legacy-Cloud-Provider-Controller zu verwenden. Weitere Informationen finden Sie in der Dokumentation zu AWS Load Balancer Controller.

AWS Gateway-API-Controller

Mit diesem Controller können Sie mithilfe der Kubernetes-Gateway-API Services über mehrere Kubernetes-Cluster hinweg verbinden. Der Controller verbindet Kubernetes-Services, die auf Amazon-EC2-Instances, -Containern und -Serverless-Funktionen ausgeführt werden, indem er den Amazon-VPC-Lattice-Service verwendet. Weitere Informationen finden Sie in der Dokumentation zu AWS -Gateway-API-Controller.

Weitere Informationen zu Add-ons finden Sie unter Amazon-EKS-Add-ons.

Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on

Das Amazon VPC CNI plugin for Kubernetes-Add-On wird auf jedem Amazon-EC2-Knoten in Ihrem Amazon-EKS-Cluster bereitgestellt. Das Add-On erstellt Elastic-Network-Schnittstellen (Netzwerkschnittstellen) und fügt sie an Ihre Amazon-EC2-Knoten an. Das Add-On weist jedem Pod und Service auch eine private `IPv4`- oder `IPv6`-Adresse von Ihrer VPC zu.

Eine Version des Add-Ons wird mit jedem Fargate-Knoten in Ihrem Cluster bereitgestellt, aber Sie aktualisieren sie nicht auf Fargate-Knoten. Andere kompatible CNI-Plugins sind für die Verwendung in Amazon-EKS-Clustern verfügbar, aber dies ist das einzige CNI-Plugin, das von Amazon EKS unterstützt wird.

In der folgenden Tabelle finden Sie die neueste Version des Kubernetes-Add-Ons, die für jede Amazon-EKS-Cluster-Version verfügbar ist.

Kubernetes-Version	1.30	1.29	1.28	1.27	1.26	1.25	1.24	1.23
VPC-CNI-Version vom Typ Amazon EKS	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild.1

Important

Wenn Sie dieses Add-on selbst verwalten, stimmen die Versionen in der Tabelle möglicherweise nicht mit den verfügbaren selbstverwalteten Versionen überein. Weitere Hinweise zur Aktualisierung des selbstverwalteten Typs dieses Add-ons finden Sie unter Aktualisieren des selbstverwalteten -Add-ons.

Important

Um auf VPC CNI v1.12.0 oder höher zu aktualisieren, müssen Sie zuerst auf VPC CNI v1.7.0 aktualisieren. Wir empfehlen Ihnen, jeweils eine Nebenversion zu aktualisieren.

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.
- Ein vorhandener AWS Identity and Access Management (IAM) () -Anbieter für Ihren Cluster. OpenID Connect OIDC Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Eine IAM-Rolle mit angefügter AmazonEKS_CNI_Policy-IAM-Richtlinie (wenn Ihr Cluster die `IPv4`-Produktfamilie verwendet) oder IPv6-Richtlinie (wenn Ihr Cluster die `IPv6`-Familie verwendet). Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

- Wenn Sie Version `1.7.0` oder höher des Amazon VPC CNI plugin for Kubernetes verwenden und benutzerdefinierte Pod-Sicherheitsrichtlinien verwenden, siehe Löschen der standardmäßigen Amazon-EKS-Pod-SicherheitsrichtliniePod-Sicherheitsrichtlinie.
-

  Important

  Amazon VPC CNI plugin for KubernetesVersionen`v1.16.0`, bei denen die Kompatibilität mit Kubernetes Versionen `1.23` und früheren Versionen `v1.16.1` entfernt wurde. Die VPC CNI-Version `v1.16.2` stellt die Kompatibilität mit Kubernetes Versionen `1.23` und früheren Versionen sowie mit den CNI-Spezifikationen wieder her. `v0.4.0`

  Amazon VPC CNI plugin for KubernetesVersionen `v1.16.0` zur `v1.16.1` Implementierung der CNI-Spezifikationsversion. `v1.0.0` Die CNI-Spezifikation `v1.0.0` wird auf EKS-Clustern unterstützt, auf denen die Kubernetes Versionen `v1.24` oder höher ausgeführt werden. VPC CNI-Version `v1.16.0` bis `v1.16.1` und CNI-Spezifikation `v1.0.0` werden in Version oder früher nicht unterstützt. Kubernetes `v1.23` Weitere Informationen `v1.0.0` zur CNI-Spezifikation finden Sie unter Container Network Interface (CNI) -Spezifikation auf

Überlegungen

- Versionen sind angegeben als `major-version.minor-version.patch-version-eksbuild.build-number`.
- Überprüfen der Versionskompatibilität für jedes Feature

  Für einige Funktionen der einzelnen Versionen sind bestimmte Versionen Amazon VPC CNI plugin for Kubernetes erforderlich. Kubernetes Wenn Sie verschiedene Amazon-EKS-Features verwenden und eine bestimmte Version des Add-ons erforderlich ist, ist dies in der Featuredokumentationen vermerkt. Falls Sie keinen bestimmten Grund haben, eine frühere Version auszuführen, empfehlen wir, die neueste Version auszuführen.

Add-on vom Typ Amazon EKS erstellen

Erstellen Sie das Add-on vom Typ Amazon EKS.

1. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist.

```
kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni:
 | cut -d : -f 3
```

Eine Beispielausgabe sieht wie folgt aus.

```
v1.16.4-eksbuild.2
```

2. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist. Je nachdem, mit welchem Tool Sie Ihr Cluster erstellt haben, ist der Add-on vom Typ Amazon EKS möglicherweise derzeit nicht auf Ihrem Cluster installiert. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

```
$ aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
 addon.addonVersion --output text
```

Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Führen Sie die verbleibenden Schritte dieses Prozesses aus, um es zu installieren.

3. Speichern Sie die Konfiguration Ihres aktuell installierten Add-ons ab.

```
kubectl get daemonset aws-node -n kube-system -o yaml > aws-k8s-cni-old.yaml
```

4. Erstellen Sie das Add-on mit dem AWS CLI. Wenn Sie das AWS Management Console oder verwenden möchten, um das Add-on `eksctl` zu erstellen, finden Sie weitere Informationen unter Erstellen eines Add-Ons und geben Sie `vpc-cni` den Namen des Add-ons an. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie bei Bedarf die folgenden Änderungen am Befehl vor, und führen Sie dann den geänderten Befehl aus.

   - Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.
   - Ersetzen Sie *v1.18.2-eksbuild.1* durch die neueste Version, die in der neuesten Versionstabelle für Ihre Cluster-Version aufgeführt ist.
   - Ersetzen Sie *111122223333* durch Ihre Konto-ID und *AmazonEKSVPCCNIRole* durch den Namen einer vorhandenen IAM-Rolle, die Sie erstellt haben. Die Angabe einer Rolle erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen für Ihren Cluster verfügen, oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.

```
aws eks create-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version v1.18.2-eksbuild.1 \
    --service-account-role-arn arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
```

Wenn Sie benutzerdefinierte Einstellungen auf Ihr aktuelles Add-on angewendet haben, die mit den Standardeinstellungen des Amazon EKS-Add-ons in Konflikt stehen, schlägt die Erstellung möglicherweise fehl. Wenn die Erstellung fehlschlägt, erhalten Sie eine Fehlermeldung, die Sie bei der Problembehebung unterstützt. Alternativ können Sie den vorherigen Befehl mit **`--resolve-conflicts OVERWRITE`** ergänzen. Dadurch kann das Add-on alle vorhandenen benutzerdefinierten Einstellungen überschreiben. Sobald Sie das Add-on erstellt haben, können Sie es mit Ihren benutzerdefinierten Einstellungen aktualisieren.

5. Vergewissern Sie sich, dass die neueste Version des Add-ons für die Kubernetes-Version Ihres Clusters zu Ihrem Cluster hinzugefügt wurde. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
    addon.addonVersion --output text
   ```

   Es kann einige Sekunden dauern, bis die Erstellung des Add-ons abgeschlossen ist.

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.18.2-eksbuild.1
   ```

6. Wenn Sie benutzerdefinierte Einstellungen für Ihr ursprüngliches Add-on vorgenommen haben, bevor Sie das Add-on vom Typ Amazon EKS erstellt haben, verwenden Sie die Konfiguration, die Sie in einem vorherigen Schritt gespeichert haben, um das Add-on vom Typ Amazon EKS mit Ihren benutzerdefinierten Einstellungen zu aktualisieren.

7. (Optional) Installieren Sie `cni-metrics-helper` in Ihrem Cluster. Es erfasst elastic network interface- und IP-Adressinformationen, aggregiert sie auf Clusterebene und veröffentlicht die Metriken auf Amazon. CloudWatch Weitere Informationen finden Sie unter cni-metrics-helper on. GitHub

Aktualisieren des Amazon-EKS-Add-ons

Erstellen Sie das Add-on vom Typ Amazon EKS. Wenn Sie das Add-on vom Typ Amazon EKS nicht zu Ihrem Cluster hinzugefügt haben, fügen Sie ihn entweder hinzu oder sehen Sie sich Aktualisieren des selbstverwalteten -Add-ons an, anstatt dieses Verfahren abzuschließen.

1. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist. Ersetzen Sie *my-cluster* mit Ihrem Clusternamen.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
    "addon.addonVersion" --output text
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.16.4-eksbuild.2
   ```

   Wenn die zurückgegebene Version mit der Version für die Kubernetes-Version Ihres Clusters in der aktuellen Versionstabelle übereinstimmt, haben Sie die neueste Version bereits auf Ihrem Cluster installiert und müssen den Rest dieses Verfahrens nicht abschließen. Wenn Sie eine Fehlermeldung statt einer Versionsnummer in Ihrer Ausgabe erhalten, haben Sie den Add-on vom Typ Amazon EKS nicht auf Ihrem Cluster installiert. Sie müssen das Add-on erstellen, bevor Sie es mit diesem Verfahren aktualisieren können.
2. Speichern Sie die Konfiguration Ihres aktuell installierten Add-ons ab.

   ```
   kubectl get daemonset aws-node -n kube-system -o yaml > aws-k8s-cni-old.yaml
   ```
3. Aktualisieren Sie Ihr Add-on mit der AWS CLI. Wenn Sie das AWS Management Console oder verwenden möchten, um das Add-on zu aktualisieren`eksctl`, finden Sie unter. Aktualisieren eines Add-Ons Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie bei Bedarf die folgenden Änderungen am Befehl vor, und führen Sie dann den geänderten Befehl aus.

   - Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.
   - Ersetzen Sie *v1.18.2-eksbuild.1* durch die neueste Version, die in der neuesten Versionstabelle für Ihre Cluster-Version aufgeführt ist.
   - Ersetzen Sie *111122223333* durch Ihre Konto-ID und *AmazonEKSVPCCNIRole* durch den Namen einer vorhandenen IAM-Rolle, die Sie erstellt haben. Die Angabe einer Rolle erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um

festzustellen, ob Sie über einen solchen für Ihren Cluster verfügen, oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.

- Die **`--resolve-conflicts`**-Option *`PRESERVE`* (Beibehalten) behält die vorhandenen Werte für das Add-on bei. Wenn Sie benutzerdefinierte Werte für Zusatzeinstellungen festgelegt haben und diese Option nicht verwenden, überschreibt Amazon EKS Ihre Werte mit seinen Standardwerten. Wenn Sie diese Option verwenden, empfehlen wir, dass Sie alle Feld- und Wertänderungen auf einem Nicht-Produktionscluster testen, bevor Sie das Add-on auf Ihrem Produktionscluster aktualisieren. Wenn Sie diesen Wert auf `OVERWRITE` ändern, werden alle Einstellungen auf die Amazon-EKS-Standardwerte geändert. Wenn Sie benutzerdefinierte Werte für Einstellungen festgelegt haben, werden diese möglicherweise mit den Amazon-EKS-Standardwerten überschrieben. Wenn Sie diesen Wert auf `none` ändern, ändert Amazon EKS den Wert der Einstellungen nicht, aber das Update schlägt möglicherweise fehl. Wenn das Update fehlschlägt, erhalten Sie eine Fehlermeldung, die Sie bei der Behebung des Konflikts unterstützt.
- Wenn Sie eine Konfigurationseinstellung nicht aktualisieren, entfernen Sie **`--configuration-values '{`***`"env":{"AWS_VPC_K8S_CNI_EXTERNALSNAT":"true"}`***`}'`** aus dem Befehl. Wenn Sie eine Konfigurationseinstellung aktualisieren, ersetzen Sie *`"env":{"AWS_VPC_K8S_CNI_EXTERNALSNAT":"true"}`* durch die Einstellung, die Sie festlegen möchten. In diesem Beispiel ist die Umgebungsvariable `AWS_VPC_K8S_CNI_EXTERNALSNAT` auf `true` gesetzt. Der von Ihnen angegebene Wert muss für das Konfigurationsschema gültig sein. Wenn Sie das Konfigurationsschema nicht kennen**`aws eks describe-addon-configuration --addon-name vpc-cni --addon-version`** ***`v1.18.2-eksbuild.1`***, führen Sie den Befehl aus und ersetzen Sie *`v1.18.2-eksbuild.1`* durch die Versionsnummer des Add-ons, für das Sie die Konfiguration anzeigen möchten. Das Schema wird in der Ausgabe zurückgegeben. Wenn Sie bereits eine benutzerdefinierte Konfiguration haben, diese komplett entfernen und die Werte für alle Einstellungen auf die Amazon EKS-Standardwerte zurücksetzen möchten, entfernen Sie *`"env":{"AWS_VPC_K8S_CNI_EXTERNALSNAT":"true"}`* aus dem Befehl, sodass nichts übrig bleibt**`{}`**. Eine Erläuterung der einzelnen Einstellungen finden Sie unter CNI-Konfigurationsvariablen auf. GitHub

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version v1.18.2-eksbuild.1 \
    --service-account-role-arn arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
 \
```

```
    --resolve-conflicts PRESERVE --configuration-values '{"env":
{"AWS_VPC_K8S_CNI_EXTERNALSNAT":"true"}}'
```

Es kann einige Sekunden dauern, bis das Update abgeschlossen ist.

4. Vergewissern Sie sich, dass die Add-on-Version aktualisiert wurde. Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni
```

Es kann einige Sekunden dauern, bis das Update abgeschlossen ist.

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "addon": {
        "addonName": "vpc-cni",
        "clusterName": "my-cluster",
        "status": "ACTIVE",
        "addonVersion": "v1.18.2-eksbuild.1",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:region:111122223333:addon/my-cluster/vpc-
cni/74c33d2f-b4dc-8718-56e7-9fdfa65d14a9",
        "createdAt": "2023-04-12T18:25:19.319000+00:00",
        "modifiedAt": "2023-04-12T18:40:28.683000+00:00",
        "serviceAccountRoleArn":
 "arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole",
        "tags": {},
        "configurationValues": "{\"env\":{\"AWS_VPC_K8S_CNI_EXTERNALSNAT\":\"true
\"}}"
    }
}
```

Aktualisieren des selbstverwalteten -Add-ons

> **Important**
>
> Wir empfehlen, den Amazon-EKS-Typ des Add-Ons zu Ihrem Cluster hinzuzufügen, anstatt den selbstverwalteten Typ des Add-Ons zu verwenden. Wenn Sie noch keine Erfahrung

mit den Unterschieden zwischen den Typen haben, finden Sie weitere Informationen unter the section called “Amazon-EKS-Add-ons”. Weitere Informationen zum Hinzufügen eines Amazon-EKS-Add-ons zu Ihrem Cluster finden Sie unter the section called “Erstellen eines Add-Ons”. Wenn Sie das Amazon EKS-Add-on nicht verwenden können, empfehlen wir Ihnen, ein Problem mit der Begründung, warum Sie das nicht können, an das GitHub Container-Roadmap-Repository zu senden.

1. Vergewissern Sie sich, dass Sie nicht den Amazon-EKS-Typ des Add-ons auf Ihrem Cluster installiert haben. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
 addon.addonVersion --output text
```

   Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-ons auf Ihrem Cluster installiert. Um das Add-on selbst zu verwalten, führen Sie die verbleibenden Schritte in diesem Verfahren aus, um das Add-on zu aktualisieren. Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Verwenden Sie zum Aktualisieren das Verfahren in Aktualisieren eines Add-Ons und nicht das Verfahren in diesem Thema. Wenn Sie mit den Unterschieden zwischen den Add-On-Typen nicht vertraut sind, finden Sie Informationen unter Amazon-EKS-Add-ons.
2. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist.

```
kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni:
 | cut -d : -f 3
```

   Eine Beispielausgabe sieht wie folgt aus.

```
v1.16.4-eksbuild.2
```

   Ihre Ausgabe enthält möglicherweise nicht die Build-Nummer.
3. Sichern Sie Ihre aktuellen Einstellungen, damit Sie dieselben Einstellungen konfigurieren können, wenn Sie Ihre Version aktualisiert haben.

```
kubectl get daemonset aws-node -n kube-system -o yaml > aws-k8s-cni-old.yaml
```

4. Um die verfügbaren Versionen anzuzeigen und sich mit den Änderungen in der Version, auf die Sie aktualisieren möchten, vertraut zu machen, siehe `releases` auf GitHub. Beachten Sie, dass wir empfehlen, auf dasselbe `major` zu aktualisieren. `minor`. `patch`Die Version ist in der Tabelle mit den neuesten verfügbaren Versionen aufgeführt, auch wenn spätere Versionen verfügbar sind auf GitHub.. Die in der Tabelle aufgeführten Buildversionen sind nicht in den selbstverwalteten Versionen aufgeführt, die unter GitHub aufgeführt sind. Aktualisieren Sie Ihre Version, indem Sie die Aufgaben mit einer der folgenden Optionen ausführen:

   - Wenn Sie keine benutzerdefinierten Einstellungen für das Add-on haben, führen Sie den Befehl unter der `To apply this release:` Überschrift GitHub für die Version aus, auf die Sie aktualisieren möchten.
   - Wenn Sie benutzerdefinierte Einstellungen haben, laden Sie die Manifestdatei mit dem folgenden Befehl herunter, anstatt sie anzuwenden. Ändern Sie *`https://raw.githubusercontent.com/aws/amazon-vpc-cni-k8s/v1.18.2/config/master/aws-k8s-cni.yaml`* in die URL für die Version GitHub , auf die Sie aktualisieren möchten.

     ```
     curl -O https://raw.githubusercontent.com/aws/amazon-vpc-cni-k8s/v1.18.2/config/master/aws-k8s-cni.yaml
     ```

     Ändern Sie bei Bedarf die Datei mit den benutzerdefinierten Einstellungen aus dem von Ihnen erstellten Backup und wenden Sie die geänderte Datei dann auf Ihren Cluster an. Wenn Ihre Knoten keinen Zugriff auf die privaten Amazon-EKS-/ Amazon-ECR-Repositorys haben, aus denen die Images abgerufen werden (siehe die Zeilen, die mit `image:` im Manifest beginnen), müssen Sie die Images herunterladen, in Ihr eigenes Repository kopieren und das Manifest ändern, um die Images aus Ihrem Repository abzurufen. Weitere Informationen finden Sie unter Kopieren eines Container-Images von einem Repository in ein anderes.

     ```
     kubectl apply -f aws-k8s-cni.yaml
     ```

5. Vergewissern Sie sich, dass die neue Version jetzt auf Ihrem Cluster installiert ist.

   ```
   kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni: | cut -d : -f 3
   ```

   Eine Beispielausgabe sieht wie folgt aus.

```
v1.18.2
```

6. (Optional) Installieren Sie `cni-metrics-helper` in Ihrem Cluster. Es erfasst elastic network interface- und IP-Adressinformationen, aggregiert sie auf Clusterebene und veröffentlicht die Metriken auf Amazon. CloudWatch Weitere Informationen finden Sie unter cni-metrics-helper on. GitHub

Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA)

Das Amazon VPC CNI plugin for Kubernetes ist das Netzwerk-Plugin für Pod-Netzwerke in Amazon-EKS-Clustern. Das Plugin ist verantwortlich für die Zuweisung von VPC-IP-Adressen zu Kubernetes-Knoten und für das Konfigurieren der erforderlichen Netzwerke für Pods auf jedem Knoten. Das Plugin:

- Erfordert AWS Identity and Access Management (IAM)-Berechtigungen. Wenn Ihr Cluster die IPv4-Familie verwendet, werden die Berechtigungen in der `AmazonEKS_CNI_Policy` AWS verwalteten Richtlinie angegeben. Wenn Ihr Cluster die IPv6-Familie verwendet, müssen die Berechtigungen einer von Ihnen erstellten IAM-Richtlinie hinzugefügt werden. Sie können diese Richtlinie der Amazon-EKS-Knoten-IAM-Rolle oder einer separaten IAM-Rolle anfügen. Wir empfehlen Ihnen, sie einer separaten Rolle zuzuweisen, wie in diesem Thema beschrieben.
- Erstellt ein Kubernetes-Servicekonto namens `aws-node` und ist so konfiguriert, dass es bei der Bereitstellung verwendet wird. Das Servicekonto ist an eine Kubernetes `clusterrole` namens `aws-node` gebunden, der die erforderlichen Kubernetes-Berechtigungen zugewiesen sind.

> **Note**
>
> Die Pods für das Amazon VPC CNI plugin for Kubernetes haben Zugriff auf die Berechtigungen, die der Amazon-EKS-Knoten-IAM-Rolle zugewiesen sind, es sei denn, Sie blockieren den Zugriff auf IMDS. Weitere Informationen finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.
- Ein vorhandener AWS Identity and Access Management (IAM) OpenID Connect (OIDC)-Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.

Schritt 1: Erstellen Sie die IAM-Rolle für das Amazon VPC CNI plugin for Kubernetes

So erstellen Sie die IAM-Rolle

1. Ermitteln Sie die IP-Familie Ihres Clusters.

   ```
   aws eks describe-cluster --name my-cluster | grep ipFamily
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   "ipFamily": "ipv4"
   ```

   Die Ausgabe kann stattdessen `ipv6` zurückgeben.
2. Erstellen Sie die IAM-Rolle. Sie können `eksctl` oder `kubectl` und verwenden AWS CLI , um Ihre IAM-Rolle zu erstellen.

   eksctl

   Erstellen Sie eine IAM-Rolle und fügen Sie der Rolle die IAM-Richtlinie mit dem Befehl zu, der der IP-Familie Ihres Clusters entspricht. Der Befehl erstellt und stellt einen - AWS CloudFormation Stack bereit, der eine IAM-Rolle erstellt, die von Ihnen angegebene Richtlinie anfügt und das vorhandene `aws-node` Kubernetes Servicekonto mit dem ARN der erstellten IAM-Rolle annotiert.

   - `IPv4`

     Ersetzen Sie *my-cluster* durch Ihren eigenen Wert.

     ```
     eksctl create iamserviceaccount \
         --name aws-node \
         --namespace kube-system \
     ```

```
    --cluster my-cluster \
    --role-name AmazonEKSVPCCNIRole \
    --attach-policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
    --override-existing-serviceaccounts \
    --approve
```

- IPv6

  Ersetzen Sie *my-cluster* durch Ihren eigenen Wert. Ersetzen Sie *111122223333* durch den Namen Ihres Clusters und *AmazonEKS_CNI_IPv6_Policy* durch Ihre IPv6-Richtlinie. Wenn Sie noch keine IPv6-Richtlinie haben, lesen Sie Erstellen Sie eine IAM-Richtlinie für Cluster, die die IPv6-Familie verwendet, um eine zu erstellen. Um IPv6 mit Ihrem Cluster zu verwenden, muss dieser mehrere Anforderungen erfüllen. Weitere Informationen finden Sie unter IPv6Adressen für ClusterPods, und services.

  ```
  eksctl create iamserviceaccount \
      --name aws-node \
      --namespace kube-system \
      --cluster my-cluster \
      --role-name AmazonEKSVPCCNIRole \
      --attach-policy-arn
   arn:aws:iam::111122223333:policy/AmazonEKS_CNI_IPv6_Policy \
      --override-existing-serviceaccounts \
      --approve
  ```

kubectl and the AWS CLI

1. Zeigen Sie die OIDC-Anbieter-URL Ihres Clusters an.

   ```
   aws eks describe-cluster --name my-cluster --query
    "cluster.identity.oidc.issuer" --output text
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
   ```

   Wenn keine Ausgabe erfolgt, müssen Sie einen IAM-OIDC-Anbieter für Ihr Cluster erstellen.

2. Kopieren Sie den folgenden Inhalt in eine Datei namens *vpc-cni-trust-policy.json*. Ersetzen Sie *111122223333* durch die ID Ihres Kontos und *EXAMPLED539D4633E53DE1B71EXAMPLE* durch den Wert, der im vorherigen Schritt zurückgegeben wurde. Ersetzen Sie durch *region-code* die AWS-Region , in der sich Ihr Cluster befindet.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com",
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:aws-node"
                }
            }
        }
    ]
}
```

3. Erstellen Sie die -Rolle. Sie können *AmazonEKSVPCCNIRole* mit einem beliebigen Namen ersetzen, den Sie wählen.

```
aws iam create-role \
  --role-name AmazonEKSVPCCNIRole \
  --assume-role-policy-document file://"vpc-cni-trust-policy.json"
```

4. Fügen Sie der Rolle die erforderliche IAM-Richtlinie an. Führen Sie den Befehl aus, der der IP-Familie Ihres Clusters entspricht.

   - IPv4

```
aws iam attach-role-policy \
```

```
  --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
  --role-name AmazonEKSVPCCNIRole
```

- IPv6

  Ersetzen Sie *111122223333* durch den Namen Ihres Clusters und *AmazonEKS_CNI_IPv6_Policy* durch Ihre IPv6-Richtlinie. Wenn Sie noch keine IPv6-Richtlinie haben, lesen Sie Erstellen Sie eine IAM-Richtlinie für Cluster, die die IPv6-Familie verwendet, um eine zu erstellen. Um IPv6 mit Ihrem Cluster zu verwenden, muss dieser mehrere Anforderungen erfüllen. Weitere Informationen finden Sie unter IPv6Adressen für ClusterPods, und services.

  ```
  aws iam attach-role-policy \
    --policy-arn arn:aws:iam::111122223333:policy/AmazonEKS_CNI_IPv6_Policy \
    --role-name AmazonEKSVPCCNIRole
  ```

5. Führen Sie den folgenden Befehl aus und fügen Sie dem aws-node-Servicekonto den ARN der IAM-Rolle hinzu, die Sie zuvor erstellt haben. Ersetzen Sie *example values* durch Ihre eigenen Werte.

   ```
   kubectl annotate serviceaccount \
       -n kube-system aws-node \
       eks.amazonaws.com/role-
   arn=arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
   ```

3. (Optional) Konfigurieren Sie den AWS Security Token Service Endpunkttyp, der von Ihrem Kubernetes Servicekonto verwendet wird. Weitere Informationen finden Sie unter Den AWS Security Token Service Endpunkt für ein Dienstkonto konfigurieren.

Schritt 2: Stellen Sie Amazon VPC CNI plugin for KubernetesPods erneut bereit

1. Löschen Sie alle vorhandenen Pods, die dem Servicekonto zugeordnet sind, und erstellen Sie sie neu, um die Umgebungsvariablen für Anmeldeinformationen anzuwenden. Die Anmerkung wird nicht auf Pods angewendet, die derzeit ohne die Anmerkung laufen. Mit dem folgenden Befehl werden die vorhandenen aws-node DaemonSet Pods gelöscht und mit der Servicekonto-Anmerkung bereitgestellt.

   ```
   kubectl delete Pods -n kube-system -l k8s-app=aws-node
   ```

2. Bestätigen Sie, dass alle Pods neu gestartet wurden.

```
kubectl get pods -n kube-system -l k8s-app=aws-node
```

3. Beschreiben Sie einen der Pods und überprüfen Sie, ob die Umgebungsvariablen `AWS_WEB_IDENTITY_TOKEN_FILE` und `AWS_ROLE_ARN` vorhanden sind. Ersetzen Sie *`cpjw7`* mit dem Namen eines Ihrer Pods, der in der Ausgabe des vorherigen Schritts zurückgegeben wurde.

```
kubectl describe pod -n kube-system aws-node-cpjw7 | grep 'AWS_ROLE_ARN:\|
AWS_WEB_IDENTITY_TOKEN_FILE:'
```

Eine Beispielausgabe sieht wie folgt aus.

```
AWS_ROLE_ARN:                 arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
      AWS_WEB_IDENTITY_TOKEN_FILE:  /var/run/secrets/eks.amazonaws.com/
serviceaccount/token
      AWS_ROLE_ARN:
 arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
      AWS_WEB_IDENTITY_TOKEN_FILE:           /var/run/secrets/eks.amazonaws.com/
serviceaccount/token
```

Zwei Sätze doppelter Ergebnisse werden zurückgegeben, da der Pod zwei Container enthält. Beide Container haben die gleichen Werte.

Wenn Ihr den AWS-Region-Endpunkt Pod verwendet, wird in der vorherigen Ausgabe auch die folgende Zeile zurückgegeben.

```
AWS_STS_REGIONAL_ENDPOINTS=regional
```

Schritt 3: Entfernen Sie die CNI-Richtlinie aus der Knoten-IAM-Rolle

Wenn Ihrer Amazon-EKS-Knoten-IAM-Rolle derzeit die `AmazonEKS_CNI_Policy` IAM-Richtlinie (`IPv4`) oder eine -`IPv6`Richtlinie zugeordnet ist und Sie stattdessen eine separate IAM-Rolle erstellt, ihr die Richtlinie angefügt und sie dem `aws-node` Kubernetes Servicekonto zugewiesen haben, empfehlen wir Ihnen, die Richtlinie mit dem AWS CLI Befehl aus Ihrer Knotenrolle zu entfernen, der der IP-Familie Ihres Clusters entspricht. Ersetzen Sie *`AmazonEKSNodeRole`* durch den Namen Ihrer Knotenrolle.

- `IPv4`

```
aws iam detach-role-policy --role-name AmazonEKSNodeRole --policy-arn
 arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
```

- IPv6

  Ersetzen Sie *111122223333* durch den Namen Ihres Clusters und *AmazonEKS_CNI_IPv6_Policy* durch Ihre IPv6-Richtlinie.

```
aws iam detach-role-policy --role-name AmazonEKSNodeRole --policy-arn
 arn:aws:iam::111122223333:policy/AmazonEKS_CNI_IPv6_Policy
```

Erstellen Sie eine IAM-Richtlinie für Cluster, die die **IPv6**-Familie verwendet

Wenn Sie einen Cluster erstellt haben, der die IPv6-Familie verwendet, und der Cluster Version 1.10.1oder höher des Amazon VPC CNI plugin for Kubernetes -Add-ons konfiguriert hat, müssen Sie eine IAM-Richtlinie erstellen, die Sie einer IAM-Rolle zuweisen können. Wenn Sie einen vorhandenen Cluster haben, den Sie bei der Erstellung nicht mit der IPv6-Familie konfiguriert haben, müssen Sie zur Verwendung von IPv6 einen neuen Cluster erstellen. Weitere Informationen zur Verwendung von IPv6 mit Ihrem Cluster finden Sie unter IPv6Adressen für ClusterPods, und services.

1. Kopieren Sie den folgenden Text und speichern Sie ihn in einer Datei mit dem Namen *vpc-cni-ipv6-policy*.json.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssignIpv6Addresses",
                "ec2:DescribeInstances",
                "ec2:DescribeTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeInstanceTypes"
            ],
            "Resource": "*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ]
        }
    ]
}
```

2. Erstellen Sie die IAM-Richtlinie.

```
aws iam create-policy --policy-name AmazonEKS_CNI_IPv6_Policy --policy-document
 file://vpc-cni-ipv6-policy.json
```

Auswahl von Anwendungsfällen für Pod-Netzwerke

Das Amazon VPC CNI plugin for Kubernetes bietet Networking für Pods. Die folgende Tabelle hilft Ihnen zu verstehen, welche Netzwerkanwendungsfälle Sie zusammen verwenden können und welche Funktionen und Amazon VPC CNI plugin for Kubernetes-Einstellungen Sie mit verschiedenen Amazon-EKS-Knotentypen verwenden können. Alle Informationen in der Tabelle gelten nur für Linux-IPv4-Knoten.

Amazon-EKS-Knoten-Typ	Amazon EC2			Fargate
Anwendungsfall	Individuelle IP-Adressen, die der Netzwerkschnittstelle zugewiesen werden	Der Netzwerkschnittstelle zugewiesene IP-Präfixe	Sicherheitsgruppen für Pods	
Benutzerdefinierte Netzwerke für Pods – Zuweisen von IP-Adress	Ja	Ja	Ja	Ja (Subnetze, die über das Fargate-Profil gesteuert werden)

Amazon-EKS-Knoten-Typ	Amazon EC2			Fargate
Anwendungsfall	Individuelle IP-Adressen, die der Netzwerkschnittstelle zugewiesen werden	Der Netzwerkschnittstelle zugewiesene IP-Präfixe	Sicherheitsgruppen für Pods	
en aus einem anderen Subnetz als dem Subnetz des Knotens				
SNAT für Pods	Ja (Standard mäßig ist `false`)	Ja (Standard mäßig ist `false`)	Ja (Nur `true`)	Ja (Nur `true`)
Funktionen				

Amazon-EKS-Knoten-Typ	Amazon EC2			Fargate
Anwendungsfall	Individuelle IP-Adressen, die der Netzwerkschnittstelle zugewiesen werden	Der Netzwerkschnittstelle zugewiesene IP-Präfixe	Sicherheitsgruppen für Pods	
Sicherheitsgruppen-Bereich	Knoten	Knoten	Pod (Wenn Sie `POD_SECURITY_GROUP_ENFORCING_MODE = standard` und `AWS_VPC_K8S_CNI_EXTERNALSNAT = false` festgelegt haben, verwendet der Datenverkehr, der für Endpunkte außerhalb der VPC bestimmt ist, die Sicherheitsgruppen des Knotens, nicht die Sicherheitsgruppen des Pod's)	Pod

Amazon-EKS-Knoten-Typ	Amazon EC2			Fargate
Anwendungsfall	Individuelle IP-Adressen, die der Netzwerkschnittstelle zugewiesen werden	Der Netzwerkschnittstelle zugewiesene IP-Präfixe	Sicherheitsgruppen für Pods	
Amazon-VPC-Subnetztypen	Privat und öffentlich	Privat und öffentlich	Nur privat	Nur privat
Netzwerkrichtlinie (VPC CNI)	Kompatibel	Kompatibel	Kompatibel Nur mit Version `1.14.0` oder höher des Amazon-VPC-CNI-Plugins	Nicht unterstützt
Poddichte pro Knoten	Medium	Hoch	Niedrig	One
Pod-Startzeit	Besser	Am besten	Gut	Mittel
Amazon-VPC-CNI-Plugin-Einstellungen (weitere Informationen zu den einzelnen Einstellungen finden Sie unter amazon-vpc-cni-k8s auf GitHub)				
`WARM_ENI_TARGET`	Ja	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
`WARM_IP_TARGET`	Ja	Ja	Nicht zutreffend	Nicht zutreffend
`MINIMUM_IP_TARGET`	Ja	Ja	Nicht zutreffend	Nicht zutreffend

Amazon-EKS-Knoten-Typ	Amazon EC2			Fargate
Anwendungsfall	Individuelle IP-Adressen, die der Netzwerkschnittstelle zugewiesen werden	Der Netzwerkschnittstelle zugewiesene IP-Präfixe	Sicherheitsgruppen für Pods	
`WARM_PREFIX_TARGET`	Nicht zutreffend	Ja	Nicht zutreffend	Nicht zutreffend

Note

- Sie können `IPv6` nicht mit benutzerdefinierten Netzwerken verwenden.
- `IPv6`-Adressen werden nicht übersetzt, daher gilt SNAT nicht.
- Der Datenverkehr zu und von Pods mit zugehörigen Sicherheitsgruppen unterliegt nicht der Durchsetzung von Calico-Netzwerkrichtlinien und ist nur auf die Durchsetzung von Amazon VPC-Sicherheitsgruppen beschränkt.
- Wenn Sie die Durchsetzung von Calico Netzwerkrichtlinien verwenden, empfehlen wir, die Umgebungsvariable `ANNOTATE_POD_IP` auf festzulegen`true`, um ein bekanntes Problem mit zu vermeidenKubernetes. Um dieses Feature verwenden zu können, müssen Sie die -`patch`Berechtigung für Pods zur hinzufügen`aws-node`ClusterRole. Beachten Sie, dass das Hinzufügen von Patch-Berechtigungen zu dem den Sicherheitsbereich für das Plugin `aws-node` DaemonSet erhöht. Weitere Informationen finden Sie unter ANNOTATE_POD_IP im VPC-CNI-Repo auf GitHub.
- IP-Präfixe und IP-Adressen sind mit standardmäßigen elastischen Amazon-EC2-Netzwerkschnittstellen verknüpft. Pods, die bestimmte Sicherheitsgruppen erfordern, wird die primäre IP-Adresse einer Zweigstellen-Netzwerkschnittstelle zugewiesen. Sie können Pods, die IP-Adressen erhalten, oder IP-Adressen von IP-Präfixen mit Pods kombinieren, die Zweigstellennetzwerkschnittstellen auf demselben Knoten erhalten.

Windows-Knoten

Jeder Knoten unterstützt nur eine Netzwerkschnittstelle. Sie können sekundäre IPv4-Adressen und IPv4-Präfixe verwenden. Standardmäßig ist die Anzahl der verfügbaren IPv4-Adressen auf dem Knoten gleich der Anzahl der sekundären IPv4-Adressen, die Sie jeder elastischen Netzwerkschnittstelle zuweisen können, minus eins. Sie können jedoch die verfügbaren IPv4-Adressen und die Pod-Dichte auf dem Knoten erhöhen, indem Sie IP-Präfixe aktivieren. Weitere Informationen finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten.

Calico-Netzwerkrichtlinien werden unter Windows unterstützt. Sie können keine Sicherheitsgruppen für Pods oder benutzerdefinierte Netzwerke unter Windows verwenden.

IPv6 Adressen für ClusterPods, und services

Standardmäßig weist Kubernetes Ihren Pods and services IPv4-Adressen zu. Anstatt Ihren Pods und services IPv4-Adressen zuzuweisen, können Sie Ihren Cluster so konfigurieren, dass er ihnen IPv6-Adressen zuweist. Amazon EKS unterstützt keine Dual-Stack-Pods oder -services, wenngleich sie von Kubernetes in Version 1.23 und höher unterstützt werden. Daher können Sie Ihren Pods und services nicht sowohl IPv4- als auch IPv6-Adressen zuweisen.

Sie wählen aus, welche IP-Familie Sie für Ihren Cluster verwenden möchten, wenn Sie ihn erstellen. Sie können die Familie nach dem Erstellen des Clusters nicht mehr ändern.

Überlegungen zur Verwendung der **IPv6** Familie für Ihren Cluster

- Sie müssen einen neuen Cluster erstellen und angeben, dass Sie die IPv6-Familie für diesen Cluster verwenden möchten. Sie können die IPv6-Familie nicht für einen Cluster aktivieren, den Sie von einer früheren Version aktualisiert haben. Eine Anleitung zur Erstellung eines neuen Clusters finden Sie unter Erstellen eines Amazon-EKS-Clusters.
- Die Version des Amazon-VPC-CNI-Add-ons, das Sie in Ihrem Cluster bereitstellen, muss Version 1.10.1 oder höher sein. Diese Version oder höher wird standardmäßig bereitgestellt. Nachdem Sie das Add-on bereitgestellt haben, können Sie Ihr Amazon VPC CNI-Add-on nicht auf eine niedrigere Version als 1.10.1 herabstufen, ohne zuerst alle Knoten in allen Knotengruppen in Ihrem Cluster zu entfernen.
- Windows, Pods und services werden nicht unterstützt.
- Wenn Sie Amazon EC2-Knoten verwenden, müssen Sie das Amazon VPC CNI-Add-on mit IP-Präfix-Delegation und IPv6 konfigurieren. Wenn Sie beim Erstellen Ihres Clusters die IPv6-Familie wählen, wird die Version 1.10.1 des Add-ons standardmäßig auf diese Konfiguration festgelegt. Dies ist sowohl bei einem selbstverwalteten als auch bei Amazon EKS Add-on der Fall.

Weitere Informationen über die IP-Präfix-Delegation finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten.

- Wenn Sie einen Cluster erstellen, müssen die VPC und die von Ihnen angegebenen Subnetze über einen IPv6-CIDR-Block verfügen, der der VPC und den von Ihnen angegebenen Subnetzen zugewiesen ist. Sie müssen auch über einen IPv4-CIDR-Block verfügen. Dies liegt daran, dass eine VPC, selbst wenn Sie nur IPv6 verwenden möchten, immer noch einen IPv4-CIDR-Block benötigt, um zu funktionieren. Weitere Informationen finden Sie unter Zuordnen eines IPv6-CIDR-Blocks zu Ihrer VPC im Amazon-VPC-Benutzerhandbuch.
- Wenn Sie Ihren Cluster und Ihre Knoten erstellen, müssen Sie Subnetze angeben, die für die automatische Zuweisung von IPv6-Adressen konfiguriert sind. Andernfalls können Sie Ihren Cluster und Ihre Knoten nicht bereitstellen. Diese Konfiguration ist standardmäßig deaktiviert. Weitere Informationen finden Sie unter Ändern des öffentlichen IPv6-Adressierungsattributs für Ihr Subnetz im Amazon-VPC-Benutzerhandbuch.
- Die Routing-Tabellen, die Ihren Subnetzen zugewiesen sind, müssen Routen für IPv6-Adressen haben. Weitere Informationen finden Sie unter Migrieren zu IPv6 im Amazon VPC-Benutzerhandbuch.
- Ihre Sicherheitsgruppen müssen IPv6-Adressen zulassen. Weitere Informationen finden Sie unter Migrieren zu IPv6 im Amazon VPC-Benutzerhandbuch.
- Sie können es nur IPv6 mit AWS Nitro-basierten Amazon EC2- oder Fargate-Knoten verwenden.
- Sie können IPv6 nicht mit Sicherheitsgruppen für Pods mit Amazon EC2-Knoten verwenden. Sie können es jedoch mit Fargate-Knoten verwenden. Wenn Sie separate Sicherheitsgruppen für einzelne Pods benötigen, verwenden Sie die IPv4-Familie weiterhin mit Amazon-EC2-Knoten, oder verwenden Sie stattdessen Fargate-Knoten.
- Wenn Sie zuvor benutzerdefinierte Netzwerke verwendet haben, um die Erschöpfung der IP-Adresse zu verringern, können Sie stattdessen IPv6 verwenden. Sie können benutzerdefinierte Netzwerke nicht mit IPv6 verwenden. Wenn Sie benutzerdefinierte Netzwerke für die Netzwerkisolierung verwenden, müssen Sie möglicherweise weiterhin benutzerdefinierte Netzwerke und die IPv4-Produktfamilie für Ihre Cluster verwenden.
- Sie können IPv6 nicht mit AWS Outposts verwenden.
- Pods und services wird nur eine IPv6-Adresse zugewiesen. Ihnen ist keine IPv4-Adresse zugewiesen. Da Pods über NAT auf der Instance selbst mit IPv4-Endpunkten kommunizieren können, werden DNS64 und NAT64 nicht benötigt. Wenn der Datenverkehr eine öffentliche IP-Adresse benötigt, wird die Quell-Netzwerkadresse des Datenverkehrs in eine öffentliche IP übersetzt.

- Die Quell-IPv6-Adresse eines Pod ist keine Quell-Netzwerkadresse, die bei der Kommunikation außerhalb der VPC in die IPv6-Adresse des Knotens übersetzt wird. Sie wird über ein Internet-Gateway oder ein Internet-Gateway für nur ausgehenden Verkehr weitergeleitet.
- Allen Knoten wird eine IPv4- und IPv6-Adresse zugewiesen.
- Der Amazon FSx for Lustre-CSI-Treiber wird nicht unterstützt.
- Sie können Version 2.3.1 oder höher des Load AWS Balancer Controllers verwenden, um den Anwendungs - oder Netzwerkverkehr IPv6 Pods im IP-Modus, aber nicht im Instanzmodus, zu verteilen. Weitere Informationen finden Sie unter Was ist die AWS Load Balancer Controller?.
- Sie müssen eine IPv6-IAM-Richtlinie an die IAM- oder CNI IAM-Rolle Ihres Knotens anhängen. Zwischen den beiden empfehlen wir, dass Sie sie an eine CNI IAM-Rolle anfügen. Weitere Informationen finden Sie unter Erstellen Sie eine IAM-Richtlinie für Cluster, die die IPv6-Familie verwendet und Schritt 1: Erstellen Sie die IAM-Rolle für das Amazon VPC CNI plugin for Kubernetes.
- Jeder Fargate-Pod erhält eine IPv6-Adresse vom CIDR, die für das Subnetz angegeben ist, in dem er bereitgestellt wird. Die zugrunde liegende Hardwareeinheit, die Fargate-Pods ausführt, erhält eine eindeutige IPv4- und IPv6-Adresse von den CIDRs, die dem Subnetz zugewiesen sind, in dem die Hardwareeinheit bereitgestellt wird.
- Wir empfehlen Ihnen, vor der Bereitstellung von IPv6 Clustern eine gründliche Bewertung Ihrer Anwendungen, Amazon EKS-Add-Ons und AWS Services durchzuführen, die Sie integrieren. Dies soll sicherstellen, dass mit IPv6 alles wie erwartet funktioniert.
- Die Verwendung des IPv6-Endpunkts des Instance-Metadatenservice von Amazon EC2 wird bei Amazon EKS nicht unterstützt.
- Beim Erstellen einer selbstverwalteten Knotengruppe in einem Cluster, der die IPv6-Familie nutzt, müssen die Benutzerdaten die folgenden BootstrapArguments für die Datei bootstrap.sh enthalten, die beim Starten des Knotens ausgeführt wird. Ersetzen Sie *your-cidr* durch den IPv6-CIDR-Bereich der VPC Ihres Clusters.

  ```
  --ip-family ipv6 --service-ipv6-cidr your-cidr
  ```

  Wenn Sie den IPv6 CIDR Bereich für Ihren Cluster nicht kennen, können Sie ihn mit dem folgenden Befehl anzeigen (erfordert die AWS CLI Version 2.4.9 oder höher).

  ```
  aws eks describe-cluster --name my-cluster --query
   cluster.kubernetesNetworkConfig.serviceIpv6Cidr --output text
  ```

Bereitstellen eines **IPv6**-Clusters und verwalteter Amazon-Linux-Knoten

In diesem Tutorial stellen Sie eine IPv6-Amazon-VPC, einen Amazon-EKS-Cluster mit der IPv6-Familie und eine verwaltete Knotengruppe mit Amazon-Linux-Knoten von Amazon EC2 bereit. Sie können Amazon-EC2-Windows-Knoten nicht in einem IPv6-Cluster bereitstellen. Sie können Fargate-Knoten auch in Ihrem Cluster bereitstellen, obwohl diese Anweisungen in diesem Thema der Einfachheit halber nicht enthalten sind.

Bevor Sie einen Cluster für den Produktionseinsatz erstellen, empfehlen wir Ihnen, sich mit allen Einstellungen vertraut zu machen und einen Cluster mit den Einstellungen bereitzustellen, die Ihren Anforderungen entsprechen. Weitere Informationen finden Sie unter Erstellen eines Amazon-EKS-Clusters, Verwaltete Knotengruppen und in den Überlegungen zu diesem Thema. Sie können nur einige Einstellungen aktivieren, wenn Sie Ihren Cluster erstellen.

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Werkzeuge und Ressourcen installieren und konfigurieren, die Sie zum Erstellen und Verwalten eines Amazon-EKS-Clusters benötigen.

- Das kubectl-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise 1.29 ist, können Sie kubectl-Version 1.28, 1.29, oder 1.30 damit verwenden. Informationen zum Installieren oder Aktualisieren von kubectl finden Sie unter Installieren oder Aktualisieren von kubectl.
- Der von Ihnen verwendete IAM-Sicherheitsprinzipal muss über Berechtigungen für die Arbeit mit Amazon EKS-IAM-Rollen, serviceverknüpften Rollen AWS CloudFormation, einer VPC und verwandten Ressourcen verfügen. Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Elastic Kubernetes Service und Verwenden von serviceverknüpften Rollen im IAM-Benutzerhandbuch.

Es werden Verfahren bereitgestellt, um die Ressourcen mit entweder eksctl oder der AWS CLI zu erstellen. Sie können die Ressourcen auch mithilfe von bereitstellen AWS Management Console, aber diese Anweisungen werden der Einfachheit halber nicht in diesem Thema bereitgestellt.

eksctl

Voraussetzung

`eksctl`-Version `0.183.0` oder höher ist auf Ihrem Computer installiert. Informationen zum Installieren oder Aktualisieren finden Sie in der Dokumentation zu `eksctl` unter Installation.

Bereitstellen eines **IPv6**-Clusters mit **`eksctl`**

1. Erstellen Sie die Datei `ipv6-cluster.yaml`. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie nach Bedarf die folgenden Änderungen am Befehl vor und führen Sie anschließend den geänderten Befehl aus:

   - Ersetzen Sie *`my-cluster`* durch Ihren Cluster-Namen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
   - Ersetzen Sie *`region-code`* durch eine AWS-Region , die von Amazon EKS unterstützt wird. Eine Liste von AWS-Regionen finden Sie unter Amazon EKS-Endpunkte und Kontingente im AWS Allgemeinen Referenzhandbuch.
   - Der Wert für `version` mit der Version Ihres Clusters. Weitere Informationen finden Sie unter Unterstützte Amazon-EKS-Kubernetes-Version.
   - Ersetzen Sie *`my-nodegroup`* durch einen Namen für Ihre Knotengruppe. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten.
   - Ersetzen Sie *`t3.medium`* durch jeden beliebigen Instance-Typ von AWS Nitro System.

```
cat >ipv6-cluster.yaml <<EOF
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code
  version: "X.XX"

kubernetesNetworkConfig:
  ipFamily: IPv6

addons:
```

```
  - name: vpc-cni
    version: latest
  - name: coredns
    version: latest
  - name: kube-proxy
    version: latest

iam:
  withOIDC: true

managedNodeGroups:
  - name: my-nodegroup
    instanceType: t3.medium
EOF
```

2. Erstellen Sie Ihren Cluster.

```
eksctl create cluster -f ipv6-cluster.yaml
```

Die Clustererstellung dauert mehrere Minuten. Fahren Sie nicht fort, bis Sie die letzte Ausgabezeile sehen, die der folgenden Ausgabe ähnelt.

```
[...]
[#]  EKS cluster "my-cluster" in "region-code" region is ready
```

3. Vergewissern Sie sich, dass Standard-Pods IPv6-Adressen zugewiesen wurden.

```
kubectl get pods -n kube-system -o wide
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                       READY   STATUS    RESTARTS   AGE     IP
                             NODE
 NOMINATED NODE   READINESS GATES
aws-node-rslts             1/1     Running   1          5m36s
  2600:1f13:b66:8200:11a5:ade0:c590:6ac8   ip-192-168-34-75.region-
code.compute.internal   <none>           <none>
aws-node-t74jh             1/1     Running   0          5m32s
  2600:1f13:b66:8203:4516:2080:8ced:1ca9   ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
```

```
coredns-85d5b4454c-cw7w2   1/1     Running   0          56m
  2600:1f13:b66:8203:34e5::                ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
coredns-85d5b4454c-tx6n8   1/1     Running   0          56m
  2600:1f13:b66:8203:34e5::1               ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
kube-proxy-btpbk           1/1     Running   0          5m36s
  2600:1f13:b66:8200:11a5:ade0:c590:6ac8   ip-192-168-34-75.region-
code.compute.internal   <none>           <none>
kube-proxy-jjk2g           1/1     Running   0          5m33s
  2600:1f13:b66:8203:4516:2080:8ced:1ca9   ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
```

4. Vergewissern Sie sich, dass Standard-Services IPv6-Adressen zugewiesen wurden.

   ```
   kubectl get services -n kube-system -o wide
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   NAME       TYPE        CLUSTER-IP          EXTERNAL-IP   PORT(S)         AGE
    SELECTOR
   kube-dns   ClusterIP   fd30:3087:b6c2::a   <none>        53/UDP,53/TCP   57m
    k8s-app=kube-dns
   ```

5. (Optional) Stellen Sie eine Beispielanwendung oder den AWS Load Balancer Controller und eine Beispielanwendung bereit, um ein Load Balancing des Datenverkehrs von Anwendungen oder Netzwerken in Richtung IPv6-Pods vorzunehmen.
6. Nachdem Sie mit dem Cluster und den Knoten fertig sind, die Sie für dieses Tutorial erstellt haben, sollten Sie die Ressourcen, die Sie erstellt haben, mit dem folgenden Befehl bereinigen.

   ```
   eksctl delete cluster my-cluster
   ```

AWS CLI

Voraussetzung

Version 2.12.3 oder höher oder Version 1.27.160 oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **aws --version | cut -d / -f2**

`| cut -d ' ' -f1`. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch. Wenn Sie die verwenden AWS CloudShell, müssen Sie möglicherweise die Version `2.12.3` oder eine spätere Version von installieren AWS CLI, da es sich bei der in der installierten AWS CLI Standardversion AWS CloudShell möglicherweise um eine frühere Version handelt. `1.27.160`

> **Important**
>
> - Sie müssen alle Schritte in diesem Verfahren als derselbe Benutzer ausführen. Führen Sie den folgenden Befehl aus, um den aktuellen Benutzer zu überprüfen:
>
>   ```
>   aws sts get-caller-identity
>   ```
>
> - Sie müssen alle Schritte in diesem Verfahren in derselben Shell ausführen. In mehreren Schritten in den vorherigen Schritten werden festgelegte Variablen verwendet. Schritte, die Variablen verwenden, funktionieren nicht ordnungsgemäß, wenn die Variablenwerte in einer anderen Shell festgelegt sind. Wenn Sie die AWS CloudShell verwenden, um das folgende Verfahren abzuschließen, denken Sie daran, dass Ihre Shell-Sitzung endet, wenn Sie etwa 20 bis 30 Minuten lang nicht mit der Tastatur oder dem Zeiger damit interagieren. Laufende Prozesse zählen nicht als Interaktionen.
> - Die Anweisungen sind für die Bash-Shell geschrieben und müssen möglicherweise in anderen Shells angepasst werden.

Um Ihren Cluster mit dem zu erstellen AWS CLI

Ersetzen Sie alle *`example values`* in den Schritten dieses Verfahrens mit eigenen Werten.

1. Führen Sie die folgenden Befehle aus, um einige in späteren Schritten verwendete Variablen festzulegen. *`region-code`*Ersetzen Sie es durch AWS-Region das, in dem Sie Ihre Ressourcen einsetzen möchten. Der Wert kann ein beliebiger Wert sein AWS-Region , der von Amazon EKS unterstützt wird. Eine Liste von AWS-Regionen finden Sie unter Amazon EKS-Endpunkte und Kontingente im AWS Allgemeinen Referenzhandbuch. Ersetzen Sie

my-cluster durch Ihren Cluster-Namen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein. Ersetzen Sie *my-nodegroup* durch einen Namen für Ihre Knotengruppe. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten. Ersetzen Sie *111122223333* durch Ihre Konto-ID.

```
export region_code=region-code
export cluster_name=my-cluster
export nodegroup_name=my-nodegroup
export account_id=111122223333
```

2. Erstellen Sie eine Amazon VPC mit öffentlichen und privaten Subnetzen, die die Anforderungen von Amazon EKS und IPv6 erfüllt.

   a. Führen Sie den folgenden Befehl aus, um eine Variable für Ihren AWS CloudFormation Stack-Namen festzulegen. Sie können *my-eks-ipv6-vpc* mit einem beliebigen Namen ersetzen, den Sie wählen.

   ```
   export vpc_stack_name=my-eks-ipv6-vpc
   ```

   b. Erstellen Sie eine IPv6 VPC mithilfe einer AWS CloudFormation Vorlage.

   ```
   aws cloudformation create-stack --region $region_code --stack-name
    $vpc_stack_name \
     --template-url https://s3.us-west-2.amazonaws.com/amazon-
   eks/cloudformation/2020-10-29/amazon-eks-ipv6-vpc-public-private-
   subnets.yaml
   ```

   Das Erstellen des Stacks nimmt einige Minuten in Anspruch. Führen Sie den folgenden Befehl aus. Fahren Sie erst mit dem nächsten Schritt fort, wenn die Ausgabe des Befehls CREATE_COMPLETE ist.

   ```
   aws cloudformation describe-stacks --region $region_code --stack-name
    $vpc_stack_name --query Stacks[].StackStatus --output text
   ```

   c. Rufen Sie die IDs der erstellten öffentlichen Subnetze ab.

```
aws cloudformation describe-stacks --region $region_code --stack-name
 $vpc_stack_name \
    --query='Stacks[].Outputs[?OutputKey==`SubnetsPublic`].OutputValue' --
output text
```

Eine Beispielausgabe sieht wie folgt aus.

```
subnet-0a1a56c486EXAMPLE,subnet-099e6ca77aEXAMPLE
```

d. Aktivieren Sie die Option „IPv6-Adressen automatisch zuweisen“ für die erstellten öffentlichen Subnetze.

```
aws ec2 modify-subnet-attribute --region $region_code --
subnet-id subnet-0a1a56c486EXAMPLE --assign-ipv6-address-on-
creation
aws ec2 modify-subnet-attribute --region $region_code --subnet-id
 subnet-099e6ca77aEXAMPLE --assign-ipv6-address-on-creation
```

e. Rufen Sie die Namen der Subnetze und Sicherheitsgruppen, die mit der Vorlage erstellt wurden, aus dem bereitgestellten AWS CloudFormation Stack ab und speichern Sie sie in Variablen, um sie in einem späteren Schritt zu verwenden.

```
security_groups=$(aws cloudformation describe-stacks --region $region_code
 --stack-name $vpc_stack_name \
    --query='Stacks[].Outputs[?OutputKey==`SecurityGroups`].OutputValue' --
output text)

public_subnets=$(aws cloudformation describe-stacks --region $region_code --
stack-name $vpc_stack_name \
    --query='Stacks[].Outputs[?OutputKey==`SubnetsPublic`].OutputValue' --
output text)

private_subnets=$(aws cloudformation describe-stacks --region $region_code
 --stack-name $vpc_stack_name \
    --query='Stacks[].Outputs[?OutputKey==`SubnetsPrivate`].OutputValue' --
output text)

subnets=${public_subnets},${private_subnets}
```

3. Erstellen Sie eine Cluster-IAM-Rolle und fügen Sie ihr die erforderliche verwaltete Amazon EKS IAM-Richtlinie hinzu. KubernetesVon Amazon EKS verwaltete Cluster rufen in Ihrem

Namen andere AWS Services auf, um die Ressourcen zu verwalten, die Sie mit dem Service verwenden.

a. Führen Sie den folgenden Befehl aus, um die Datei `eks-cluster-role-trust-policy.json` zu erstellen.

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

b. Führen Sie den folgenden Befehl aus, um eine Variable für Ihren Rollennamen festzulegen. Sie können *myAmazonEKSClusterRole* mit einem beliebigen Namen ersetzen, den Sie wählen.

```
export cluster_role_name=myAmazonEKSClusterRole
```

c. Erstellen Sie die -Rolle.

```
aws iam create-role --role-name $cluster_role_name --assume-role-policy-
document file://"eks-cluster-role-trust-policy.json"
```

d. Rufen Sie den ARN der IAM-Rolle ab und speichern Sie ihn für einen späteren Schritt in einer Variablen.

```
cluster_iam_role=$(aws iam get-role --role-name $cluster_role_name --
query="Role.Arn" --output text)
```

e. Hängen Sie die erforderliche von Amazon EKS verwaltete IAM-Richtlinie an die Rolle an.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSClusterPolicy --role-name $cluster_role_name
```

4. Erstellen Sie Ihren Cluster.

```
aws eks create-cluster --region $region_code --name $cluster_name --kubernetes-
version 1.XX \
   --role-arn $cluster_iam_role --resources-vpc-config subnetIds=
$subnets,securityGroupIds=$security_groups \
   --kubernetes-network-config ipFamily=ipv6
```

-
> **Note**
>
> Sie erhalten möglicherweise eine Fehlermeldung, dass eine der Availability Zones in Ihrer Anfrage nicht über genügend Kapazität zum Erstellen eines Amazon-EKS-Clusters verfügt. Wenn dies der Fall ist, enthält die Fehlerausgabe die Availability Zones, die einen neuen Cluster unterstützen können. Versuchen Sie, Ihren Cluster mit mindestens zwei Subnetzen erneut zu erstellen, die sich in den unterstützten Availability Zones für Ihr Konto befinden. Weitere Informationen finden Sie unter Unzureichende Kapazität.

  Die Erstellung des Clusters dauert mehrere Minuten. Führen Sie den folgenden Befehl aus. Fahren Sie erst mit dem nächsten Schritt fort, wenn die Ausgabe aus dem Befehl `ACTIVE` ist.

```
aws eks describe-cluster --region $region_code --name $cluster_name --query
 cluster.status
```

5. Erstellen oder aktualisieren Sie eine `kubeconfig`-Datei für Ihren Cluster, sodass Sie mit Ihrem Cluster kommunizieren können.

```
aws eks update-kubeconfig --region $region_code --name $cluster_name
```

   Standardmäßig wird die `config`-Datei in `~/.kube` erstellt oder die Konfiguration des neuen Clusters wird einer vorhandenen `config`-Datei in `~/.kube` hinzugefügt.

6. Erstellen Sie eine Knoten-IAM-Rolle.

a. Führen Sie den folgenden Befehl aus, um die Datei `vpc-cni-ipv6-policy.json` zu erstellen.

```
cat >vpc-cni-ipv6-policy <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssignIpv6Addresses",
                "ec2:DescribeInstances",
                "ec2:DescribeTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeInstanceTypes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ]
        }
    ]
}
EOF
```

b. Erstellen Sie die IAM-Richtlinie.

```
aws iam create-policy --policy-name AmazonEKS_CNI_IPv6_Policy --policy-
document file://vpc-cni-ipv6-policy.json
```

c. Führen Sie den folgenden Befehl aus, um die Datei `node-role-trust-relationship.json` zu erstellen.

```
cat >node-role-trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

d. Führen Sie den folgenden Befehl aus, um eine Variable für Ihren Rollennamen festzulegen. Sie können *AmazonEKSNodeRole* mit einem beliebigen Namen ersetzen, den Sie wählen.

```
export node_role_name=AmazonEKSNodeRole
```

e. Erstellen Sie die IAM-Rolle.

```
aws iam create-role --role-name $node_role_name --assume-role-policy-
document file://"node-role-trust-relationship.json"
```

f. Fügen Sie die IAM-Richtlinie an die IAM-Rolle an.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::
$account_id:policy/AmazonEKS_CNI_IPv6_Policy \
    --role-name $node_role_name
```

> **Important**
>
> Der Einfachheit halber wird in diesem Tutorial die Richtlinie an diese IAM-Rolle angehängt. In einem Produktions-Cluster empfehlen wir jedoch, die Richtlinie an eine separate IAM-Rolle anzuhängen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

g. Hängen Sie die beiden erforderlichen IAM-verwalteten Richtlinien an die IAM-Rolle an.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSWorkerNodePolicy \
```

```
  --role-name $node_role_name
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly \
  --role-name $node_role_name
```

h. Rufen Sie den ARN der IAM-Rolle ab und speichern Sie ihn für einen späteren Schritt in einer Variablen.

```
node_iam_role=$(aws iam get-role --role-name $node_role_name --
query="Role.Arn" --output text)
```

7. Erstellen Sie eine verwaltete Knotengruppe.

a. Zeigen Sie die IDs der Subnetze an, die Sie in einem vorherigen Schritt erstellt haben.

```
echo $subnets
```

Eine Beispielausgabe sieht wie folgt aus.

```
subnet-0a1a56c486EXAMPLE,subnet-099e6ca77aEXAMPLE,subnet-
0377963d69EXAMPLE,subnet-0c05f819d5EXAMPLE
```

b. Erstellen Sie die Knotengruppe. Ersetzen Sie *0a1a56c486EXAMPLE*, *099e6ca77aEXAMPLE*, *0377963d69EXAMPLE* und *0c05f819d5EXAMPLE* mit den im vorherigen Schritt zurückgegebenen Werten. Entfernen Sie unbedingt die Kommata zwischen Subnetz-IDs aus der vorherigen Ausgabe im folgenden Befehl. Sie können *t3.medium* mit jedem beliebigen AWS -Nitro-System-Instance-Typ ersetzen.

```
aws eks create-nodegroup --region $region_code --cluster-name $cluster_name
 --nodegroup-name $nodegroup_name \
    --subnets subnet-0a1a56c486EXAMPLE subnet-099e6ca77aEXAMPLE
 subnet-0377963d69EXAMPLE subnet-0c05f819d5EXAMPLE \
    --instance-types t3.medium --node-role $node_iam_role
```

Das Erstellen der Knotengruppe nimmt einige Minuten in Anspruch. Führen Sie den folgenden Befehl aus. Fahren Sie nicht mit dem nächsten Schritt fort, bis die zurückgegebene Ausgabe `ACTIVE` ist.

```
aws eks describe-nodegroup --region $region_code --cluster-name
 $cluster_name --nodegroup-name $nodegroup_name \
```

```
    --query nodegroup.status --output text
```

8. Vergewissern Sie sich, dass den Standard-Pods in der IP-Spalte IPv6-Adressen zugewiesen wurden.

```
kubectl get pods -n kube-system -o wide
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                       READY   STATUS    RESTARTS   AGE     IP
                           NODE
 NOMINATED NODE   READINESS GATES
aws-node-rslts             1/1     Running   1          5m36s
  2600:1f13:b66:8200:11a5:ade0:c590:6ac8   ip-192-168-34-75.region-
code.compute.internal   <none>           <none>
aws-node-t74jh             1/1     Running   0          5m32s
  2600:1f13:b66:8203:4516:2080:8ced:1ca9   ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
coredns-85d5b4454c-cw7w2   1/1     Running   0          56m
  2600:1f13:b66:8203:34e5::                ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
coredns-85d5b4454c-tx6n8   1/1     Running   0          56m
  2600:1f13:b66:8203:34e5::1               ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
kube-proxy-btpbk           1/1     Running   0          5m36s
  2600:1f13:b66:8200:11a5:ade0:c590:6ac8   ip-192-168-34-75.region-
code.compute.internal   <none>           <none>
kube-proxy-jjk2g           1/1     Running   0          5m33s
  2600:1f13:b66:8203:4516:2080:8ced:1ca9   ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
```

9. Vergewissern Sie sich, dass den Standard-Services in der IP-Spalte IPv6-Adressen zugewiesen wurden.

```
kubectl get services -n kube-system -o wide
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME       TYPE        CLUSTER-IP          EXTERNAL-IP   PORT(S)         AGE
 SELECTOR
```

```
kube-dns   ClusterIP   fd30:3087:b6c2::a   <none>        53/UDP,53/TCP   57m
 k8s-app=kube-dns
```

10. (Optional) Stellen Sie eine Beispielanwendung oder den AWS Load Balancer Controller und eine Beispielanwendung bereit, um ein Load Balancing des Datenverkehrs von Anwendungen oder Netzwerken in Richtung IPv6-Pods vorzunehmen.
11. Nachdem Sie mit dem Cluster und den Knoten fertig sind, die Sie für dieses Tutorial erstellt haben, sollten Sie die Ressourcen, die Sie erstellt haben, mit den folgenden Befehlen bereinigen. Stellen Sie sicher, dass Sie keine der Ressourcen außerhalb dieses Tutorials verwenden, bevor Sie sie löschen.

    a. Wenn Sie diesen Schritt in einer anderen Shell als die vorherigen Schritte ausführen, legen Sie die Werte aller in den vorherigen Schritten verwendeten Variablen fest und ersetzen Sie die *example values* mit den Werten, die Sie angegeben haben, als Sie die vorherigen Schritte ausgeführt haben. Wenn Sie diesen Schritt in derselben Shell ausführen, in der Sie die vorherigen Schritte ausgeführt haben, fahren Sie mit dem nächsten Schritt fort.

    ```
    export region_code=region-code
    export vpc_stack_name=my-eks-ipv6-vpc
    export cluster_name=my-cluster
    export nodegroup_name=my-nodegroup
    export account_id=111122223333
    export node_role_name=AmazonEKSNodeRole
    export cluster_role_name=myAmazonEKSClusterRole
    ```

    b. Löschen Sie Ihre Knotengruppe.

    ```
    aws eks delete-nodegroup --region $region_code --cluster-name $cluster_name
     --nodegroup-name $nodegroup_name
    ```

    Dies dauert in der Regel einige Minuten. Führen Sie den folgenden Befehl aus. Fahren Sie nicht mit dem nächsten Schritt fort, bis die Ausgabe zurückgegeben wurde.

    ```
    aws eks list-nodegroups --region $region_code --cluster-name $cluster_name
     --query nodegroups --output text
    ```

    c. Löschen Sie den -Cluster.

```
aws eks delete-cluster --region $region_code --name $cluster_name
```

Das Löschen des Clusters nimmt einige Minuten in Anspruch. Stellen Sie vor dem Fortfahren sicher, dass der Cluster mit dem folgenden Befehl gelöscht wird.

```
aws eks describe-cluster --region $region_code --name $cluster_name
```

Fahren Sie erst mit dem nächsten Schritt fort, wenn Ihre Ausgabe der folgenden Ausgabe ähnelt.

```
An error occurred (ResourceNotFoundException) when calling the
 DescribeCluster operation: No cluster found for name: my-cluster.
```

d. Löschen Sie die IAM-Ressourcen, die Sie erstellt haben. Ersetzen Sie *AmazonEKS_CNI_IPv6_Policy* mit dem von Ihnen gewählten Namen, wenn Sie einen anderen Namen als den in den vorherigen Schritten verwendeten gewählt haben.

```
aws iam detach-role-policy --role-name $cluster_role_name --policy-arn
 arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
aws iam detach-role-policy --role-name $node_role_name --policy-arn
 arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
aws iam detach-role-policy --role-name $node_role_name --policy-arn
 arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
aws iam detach-role-policy --role-name $node_role_name --policy-arn
 arn:aws:iam::$account_id:policy/AmazonEKS_CNI_IPv6_Policy
aws iam delete-policy --policy-arn arn:aws:iam::
$account_id:policy/AmazonEKS_CNI_IPv6_Policy
aws iam delete-role --role-name $cluster_role_name
aws iam delete-role --role-name $node_role_name
```

e. Löschen Sie den AWS CloudFormation Stack, der die VPC erstellt hat.

```
aws cloudformation delete-stack --region $region_code --stack-name
 $vpc_stack_name
```

SNAT für Pods

Wenn Sie Ihren Cluster mit der `IPv6`-Familie bereitgestellt haben, treffen die Informationen in diesem Thema nicht auf Ihren Cluster zu, da für `IPv6`-Adressen keine Netzwerkübersetzung möglich ist. Weitere Informationen zur Verwendung von `IPv6` mit Ihrem Cluster finden Sie unter IPv6Adressen für ClusterPods, und services.

Standardmäßig ist jedem Pod in Ihrem Cluster eine private `IPv4`-Adresse von einem Classless Inter-Domain Routing (CIDR)-Block zugewiesen, der mit der VPC verknüpft ist, in der der Pod bereitgestellt wird. Pods in derselben VPC kommunizieren miteinander, indem sie diese privaten IP-Adressen als Endpunkte verwenden. Wenn ein Pod mit einer `IPv4`-Adresse kommuniziert, die nicht Teil eines mit Ihrer VPC verknüpften CIDR-Blocks ist, übersetzt das Amazon-VPC-CNI-Plugin (für sowohl Linux als auch Windows) die `IPv4`-Adresse des Pod's standardmäßig in die primäre private `IPv4`-Adresse der primären Elastic-Network-Schnittstelle des Knotens, auf dem der Pod ausgeführt wird*.

> **Note**
>
> Bei Windows-Knoten sind zusätzliche Details zu beachten. Standardmäßig ist das VPC CNI-Plugin für Windows mit einer Netzwerkkonfiguration definiert, in der der Datenverkehr zu einem Ziel innerhalb derselben VPC für SNAT ausgeschlossen ist. Dies bedeutet, dass für die interne VPC-Kommunikation SNAT deaktiviert ist und die einem Pod zugewiesene IP-Adresse innerhalb der VPC routingfähig ist. Beim Datenverkehr zu einem Ziel außerhalb der VPC wird diePod-Quell-IP jedoch an die primäre IP-Adresse der ENI der Instance per SNAT übersetzt. Diese Standardkonfiguration für Windows stellt sicher, dass der Pod auf die gleiche Weise wie die Host-Instance auf Netzwerke außerhalb Ihrer VPC zugreifen kann.

Folgen dieses Verhaltens:

- Ihre Pods können nur dann mit Internetressourcen kommunizieren, wenn dem Knoten, auf dem sie ausgeführt werden, eine öffentliche oder elastische IP-Adresse zugeordnet ist und sie sich in einem öffentlichen Subnetz befinden. Eine Routing-Tabelle, die einem öffentlichen Subnetz zugeordnet ist, verfügt über eine Route zu einem Internet-Gateway. Wir empfehlen, Knoten nach Möglichkeit in privaten Subnetzen bereitzustellen.
- Bei Versionen des Plugins vor `1.8.0` können Ressourcen in Netzwerken oder VPCs, die über VPC-Peering, eine Transit-VPC oder AWS Direct Connect mit Ihrer Cluster-VPC verbunden sind, keine Kommunikation mit Ihren Pods hinter sekundären elastischen Netzwerkschnittstellen

initiieren. Ihre Pods können jedoch eine Kommunikation zu diesen Ressourcen initiieren und Antworten von ihnen erhalten.

Wenn eine der folgenden Aussagen in Ihrer Umgebung zutrifft, ändern Sie die Standardkonfiguration mit dem folgenden Befehl.

- Sie haben Ressourcen in Netzwerken oder VPCs, die über VPC-Peering, eine Transit-VPC oder AWS Direct Connect mit Ihrer Cluster-VPC verbunden sind und die Kommunikation mit Ihren Pods über eine `IPv4`-Adresse initiieren müssen, und Ihre Plugin-Version ist älter als `1.8.0`.
- Ihre Pods befinden sich in einem privaten Subnetz und müssen ausgehend mit dem Internet kommunizieren. Das Subnetz hat eine Route zu einem NAT-Gateway.

```
kubectl set env daemonset -n kube-system aws-node AWS_VPC_K8S_CNI_EXTERNALSNAT=true
```

> **Note**
>
> Die CNI-Konfigurationsvariablen `AWS_VPC_K8S_CNI_EXTERNALSNAT` und `AWS_VPC_K8S_CNI_EXCLUDE_SNAT_CIDRS` gelten nicht für Windows-Knoten. Die Deaktivierung von SNAT wird für Windows nicht unterstützt. Was den Ausschluss einer Liste von `IPv4`-CIDRs aus SNAT angeht, können Sie dies definieren, indem Sie im Windows-Bootstrap-Skript den `ExcludedSnatCIDRs`-Parameter angeben. Weitere Informationen zur Verwendung dieses Parameters finden Sie unter Bootstrap-Skript-Konfigurationsparameter.

*Wenn die Spezifikationen eines Pod's `hostNetwork=true` enthalten (der Standard ist `false`) wird die IP-Adresse nicht in eine andere Adresse übersetzt. Das gilt standardmäßig für die `kube-proxy`- und Amazon VPC CNI plugin for Kubernetes-Pods, die in Ihrem Cluster ausgeführt werden. Für diese Pods ist die IP-Adresse dieselbe wie die primäre IP-Adresse des Knotens. Daher wird die IP-Adresse des Pod's nicht übersetzt. Weitere Informationen zu einer Pod's `hostNetwork` Einstellung finden Sie unter PodSpec v1 core in der Kubernetes API-Referenz.

Konfigurieren Ihres Clusters für Kubernetes-Netzwerkrichtlinien

Standardmäßig gibt es in Kubernetes keine Einschränkungen für IP-Adressen, Ports oder Verbindungen zwischen Pods in Ihrem Cluster oder zwischen Ihren Pods und Ressourcen in anderen Netzwerken. Sie können die Kubernetes-Netzwerkrichtlinie verwenden, um Netzwerkverkehr zu und

von Ihren Pods einzuschränken. Weitere Informationen finden Sie unter Netzwerkrichtlinien in der Kubernetes-Dokumentation.

Wenn Sie `1.13` oder eine frühere Version des Amazon VPC CNI plugin for Kubernetes für Ihren Cluster verwenden, müssen Sie eine Drittanbieterlösung implementieren, um Kubernetes-Netzwerkrichtlinien auf Ihren Cluster anzuwenden. `1.14` oder eine höhere Version des Plug-ins kann Netzwerkrichtlinien implementieren, sodass Sie keine Drittanbieterlösung verwenden müssen. In diesem Thema erfahren Sie, wie Sie Ihren Cluster so konfigurieren, dass Sie Kubernetes-Netzwerkrichtlinien für Ihren Cluster nutzen können, ohne ein Drittanbieter-Add-on zu verwenden.

Netzwerkrichtlinien im Amazon VPC CNI plugin for Kubernetes werden in den folgenden Konfigurationen unterstützt.

- Amazon-EKS-Cluster ab Version `1.25`.
- Version 1.14 oder höher von Amazon VPC CNI plugin for Kubernetes in Ihrem Cluster.
- Für `IPv4`- oder `IPv6`-Adressen konfigurierter Cluster.
- Sie können Netzwerkrichtlinien mit Sicherheitsgruppen für Pods verwenden. Mit Netzwerkrichtlinien können Sie die gesamte Kommunikation innerhalb eines Clusters steuern. Mit Sicherheitsgruppen für Pods können Sie den Zugriff auf AWS-Services Anwendungen innerhalb eines steuernPod.
- Sie können Netzwerkrichtlinien mit benutzerdefinierten Netzwerken und Präfixdelegierung verwenden.

Überlegungen

- Bei der Anwendung von Netzwerkrichtlinien des Amazon VPC CNI plugin for Kubernetes auf Ihren Cluster mit dem Amazon VPC CNI plugin for Kubernetes können die Richtlinien nur auf Amazon-EC2-Linux-Knoten angewendet werden. Sie können nicht auf Fargate- oder Windows-Knoten angewendet werden.
- Wenn Ihr Cluster derzeit eine Drittanbieterlösung zur Verwaltung von Kubernetes-Netzwerkrichtlinien verwendet, können Sie dieselben Richtlinien mit dem Amazon VPC CNI plugin for Kubernetes verwenden. Sie müssen jedoch Ihre vorhandene Lösung entfernen, damit sie nicht dieselben Richtlinien verwaltet.
- Sie können mehrere Netzwerkrichtlinien auf denselben Pod anwenden. Wenn zwei oder mehr Richtlinien konfiguriert werden, die denselben Pod auswählen, werden alle Richtlinien auf den Pod angewendet.

- Die maximale Anzahl eindeutiger Kombinationen von Ports für jedes Protokoll in jedem Protokoll `ingress:` oder für jeden `egress:` Selektor in einer Netzwerkrichtlinie beträgt 24.
- Für jeden Ihrer Kubernetes-Services muss der Port des Services mit dem Port des Containers identisch sein. Wenn Sie benannte Ports verwenden, verwenden Sie denselben Namen auch in der Servicespezifikation.
- Durchsetzung von Richtlinien beim Start Pod

  Das Amazon VPC CNI plugin for Kubernetes konfiguriert die Netzwerkrichtlinien für Pods parallel zur Pod-Bereitstellung. Bis alle Richtlinien für den neuen Pod konfiguriert sind, beginnen Container im neuen Pod mit einer standardmäßigen Zulassungsrichtlinie. Dies wird als Standardmodus bezeichnet. Eine standardmäßige Zulassungsrichtlinie bedeutet, dass der gesamte eingehende und ausgehende Datenverkehr zu und von den neuen Pods zugelassen ist.

  Sie können diese Standard-Netzwerkrichtlinie ändern, indem Sie die Umgebungsvariable `NETWORK_POLICY_ENFORCING_MODE` `strict` im `aws-node` Container der `DaemonSet` VPC-CNI auf setzen.

  ```
  env:
    - name: NETWORK_POLICY_ENFORCING_MODE
      value: "strict"
  ```

  Wenn die `NETWORK_POLICY_ENFORCING_MODE` Variable auf gesetzt ist`strict`, beginnen Pods, die das VPC-CNI verwenden, mit einer Standardverweigerungsrichtlinie. Anschließend werden die Richtlinien konfiguriert. Dies wird als strikter Modus bezeichnet. Im strikten Modus benötigen Sie eine Netzwerkrichtlinie für jeden Endpunkt, auf den Ihre Pods in Ihrem Cluster zugreifen müssen. Beachten Sie, dass diese Anforderung für die CoreDNS Pods gilt. Die standardmäßige Ablehnungsrichtlinie ist nicht für Pods mit Host-Netzwerk konfiguriert.
- Das Netzwerkrichtlinienfeature erstellt und erfordert die Definition einer benutzerdefinierten `PolicyEndpoint`-Ressource (Custom Resource Definition, CRD) namens `policyendpoints.networking.k8s.aws`. `PolicyEndpoint`-Objekte der benutzerdefinierten Ressource werden von Amazon EKS verwaltet. Ändern oder löschen Sie diese Ressourcen nicht.
- Wenn Sie Pods ausführen, die die IAM-Anmeldeinformationen der Instance-Rolle verwenden oder eine Verbindung mit EC2 IMDS herstellen, achten Sie auf Netzwerkrichtlinien, die den Zugriff auf EC2 IMDS blockieren würden. Möglicherweise müssen Sie eine Netzwerkrichtlinie hinzufügen, um den Zugriff auf EC2 IMDS zu ermöglichen. Weitere Informationen finden Sie unter Instance-Metadaten und Benutzerdaten im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Pods, die IAM-Rollen für Servicekonten verwenden, haben keinen Zugriff auf EC2 IMDS.

- Das Amazon VPC CNI plugin for Kubernetes wendet nur Netzwerkrichtlinien auf die primäre Schnittstelle für jeden Pod (`eth0`) an, nicht auf zusätzliche Netzwerkschnittstellen für die einzelnen Pods. Das hat Auswirkungen auf folgende Architekturen:
  - `IPv6`-Pods, bei denen die Variable `ENABLE_V4_EGRESS` auf `true` festgelegt ist. Diese Variable ermöglicht es der `IPv4`-Ausgangsfunktion, eine Verbindung zwischen den IPv6-Pods und `IPv4`-Endpunkten (beispielsweise außerhalb des Clusters) herzustellen. Die `IPv4`-Ausgangsfunktion erstellt eine zusätzliche Netzwerkschnittstelle mit einer lokalen IPv4-Loopback-Adresse.
  - Bei der Verwendung verketteter Netzwerk-Plugins wieMultus. Da diese Plugins jedem Pod Netzwerkschnittstellen hinzufügen, werden Netzwerkrichtlinien nicht auf die verketteten Netzwerk-Plugins angewendet.
- Das Netzwerkrichtlinien-Feature verwendet standardmäßig den Port `8162` auf dem Knoten für Metriken. Außerdem verwendete das Feature den Port `8163` für Zustandstests. Wenn Sie eine andere Anwendung auf die Knoten oder innerhalb von Pods ausführen, die diese Ports verwenden müssen, kann die App nicht ausgeführt werden. In der VPC-CNI-Version `v1.14.1` oder höher können Sie diese Ports an den folgenden Stellen ändern:

  AWS Management Console

  1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
  2. Wählen Sie im linken Navigationsbereich die Option Cluster aus. Wählen Sie dann den Namen des Clusters aus, für den Sie das Amazon-VPC-CNI-Add-on konfigurieren möchten.
  3. Wählen Sie die Registerkarte Add-ons.
  4. Wählen Sie das Kästchen oben rechts in der Add-On-Box aus und wählen Sie dann Edit (Bearbeiten).
  5. Gehen Sie auf der Seite Configure ***`name of addon`*** (Namen des Add-Ons konfigurieren) wie folgt vor:

     a. Wählen Sie `v1.14.0-eksbuild.3` oder eine neuere Version in der Dropdown-Liste aus.

     b. Erweitern Sie Optionale Konfigurationseinstellungen.

     c. Geben Sie den JSON-Schlüssel `"enableNetworkPolicy":` und den Wert `"true"` in Konfigurationswerte ein. Der resultierende Text muss ein gültiges JSON-Objekt sein.

Wenn dieser Schlüssel und dieser Wert die einzigen Daten im Textfeld sind, setzen Sie den Schlüssel und den Wert in geschweifte Klammern `{}`.

Im folgenden Beispiel ist die Netzwerkrichtlinien-Funktion aktiviert, die Netzwerkrichtlinien-Logs sind aktiviert, die CloudWatch Netzwerkrichtlinien-Logs werden an Amazon Logs gesendet und die Metriken und Integritätstests sind auf die Standard-Portnummern gesetzt:

```
{
    "enableNetworkPolicy": "true",
    "nodeAgent": {
        "enablePolicyEventLogs": "true",
        "enableCloudWatchLogs": "true",
        "healthProbeBindAddr": "8163",
        "metricsBindAddr": "8162"
    }
}
```

Helm

Wenn Sie das Amazon VPC CNI plugin for Kubernetes über `helm` installiert haben, können Sie die Konfiguration aktualisieren, um die Ports zu ändern.

- Führen Sie den folgenden Befehl aus, um die Ports zu ändern. Legen Sie die Portnummer im Wert für Schlüssel `nodeAgent.metricsBindAddr` bzw. Schlüssel `nodeAgent.healthProbeBindAddr` fest.

```
helm upgrade --set nodeAgent.metricsBindAddr=8162 --set
 nodeAgent.healthProbeBindAddr=8163 aws-vpc-cni --namespace kube-system eks/
aws-vpc-cni
```

kubectl

1. Öffnen Sie das `DaemonSet` `aws-node` in Ihrem Editor.

```
kubectl edit daemonset -n kube-system aws-node
```

2. Ersetzen Sie die Portnummern in den folgenden Befehlsargumenten im `args:` im `aws-network-policy-agent`-Container im `aws-node`-DaemonSet-Manifest von VPC CNI.

```
- args:
        - --metrics-bind-addr=:8162
        - --health-probe-bind-addr=:8163
```

Voraussetzungen

- Minimale Cluster-Version

  Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS. Der Cluster muss Kubernetes Version `1.25` oder höher aufweisen. Auf dem Cluster muss eine der in der folgenden Tabelle aufgeführten Kubernetes-Versionen und Plattform-Versionen ausgeführt werden. Beachten Sie, dass alle Kubernetes- und Plattformversionen, die über die aufgeführten hinausgehen, ebenfalls unterstützt werden. Sie können Ihre aktuelle Kubernetes-Version überprüfen, indem Sie *`my-cluster`* im folgenden Befehl durch den Namen Ihres Clusters ersetzen und dann den geänderten Befehl ausführen:

```
aws eks describe-cluster
              --name my-cluster --query cluster.version --output
              text
```

Kubernetes-Version	Plattformversion
`1.27.4`	`eks.5`
`1.26.7`	`eks.6`
`1.25.12`	`eks.7`

- VPC-CNI-Mindestversion

  Version `1.14` oder höher von Amazon VPC CNI plugin for Kubernetes in Ihrem Cluster. Sie können Ihre aktuelle Version mit dem folgenden Befehl überprüfen:

```
kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni: |
 cut -d : -f 3
```

Wenn Ihre Version älter als `1.14` ist, finden Sie unter Aktualisieren des Amazon-EKS-Add-ons Informationen zum Aktualisieren des Plugins auf Version `1.14` oder höher.

- Linux-Kernel-Mindestversion

  Ihre Knoten müssen eine Linux-Kernelversion ab Version `5.10` haben. Sie können Ihre Kernelversion mit `uname -r` überprüfen. Wenn Sie die aktuellen Versionen der für Amazon EKS optimierten Amazon-Linux-, für Amazon EKS optimierten beschleunigten Amazon-Linux- und Bottlerocket-AMIs verwenden, verfügen diese bereits über die erforderliche Kernelversion.

  Amazon-EKS-optimierte Amazon-Linux-AMI-Versionen ab `v20231116` verfügen über Kernel-Version `5.10`.

Konfigurieren Ihres Clusters für die Verwendung von Kubernetes-Netzwerkrichtlinien

1. Mounten des BPF-Dateisystems

   > **Note**
   >
   > Wenn Ihr Cluster Version `1.27` oder höher hat, können Sie diesen Schritt überspringen, da alle für Amazon EKS optimierten Amazon-Linux- und Bottlerocket-AMIs für `1.27` oder höher dieses Feature bereits haben.
   > Für alle anderen Clusterversionen können Sie diesen Schritt überspringen, wenn Sie das für Amazon EKS optimierte Amazon-Linux-AMI mindestens auf Version `v20230703` aktualisieren oder das Bottlerocket-AMI mindestens auf Version `v1.0.2` aktualisieren.

   a. Mounten Sie das Berkeley Packet Filter (BPF)-Dateisystem auf jedem Ihrer Knoten.

      ```
      sudo mount -t bpf bpffs /sys/fs/bpf
      ```

   b. Fügen Sie dann denselben Befehl Ihren Benutzerdaten in Ihrer Startvorlage für Ihre Amazon-EC2-Auto-Scaling-Gruppen hinzu.

2. Aktivieren der Netzwerkrichtlinie in VPC CNI

   a. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist. Je nachdem, mit welchem Tool Sie Ihr Cluster erstellt haben, ist der Add-on vom Typ Amazon

EKS möglicherweise derzeit nicht auf Ihrem Cluster installiert. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
 addon.addonVersion --output text
```

Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert.

b. • Amazon-EKS-Add-On

AWS Management Console

a. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

b. Wählen Sie im linken Navigationsbereich die Option Cluster aus. Wählen Sie dann den Namen des Clusters aus, für den Sie das Amazon-VPC-CNI-Add-on konfigurieren möchten.

c. Wählen Sie die Registerkarte Add-ons.

d. Wählen Sie das Kästchen oben rechts in der Add-On-Box aus und wählen Sie dann Edit (Bearbeiten).

e. Gehen Sie auf der Seite Configure ***name of addon*** (Namen des Add-Ons konfigurieren) wie folgt vor:

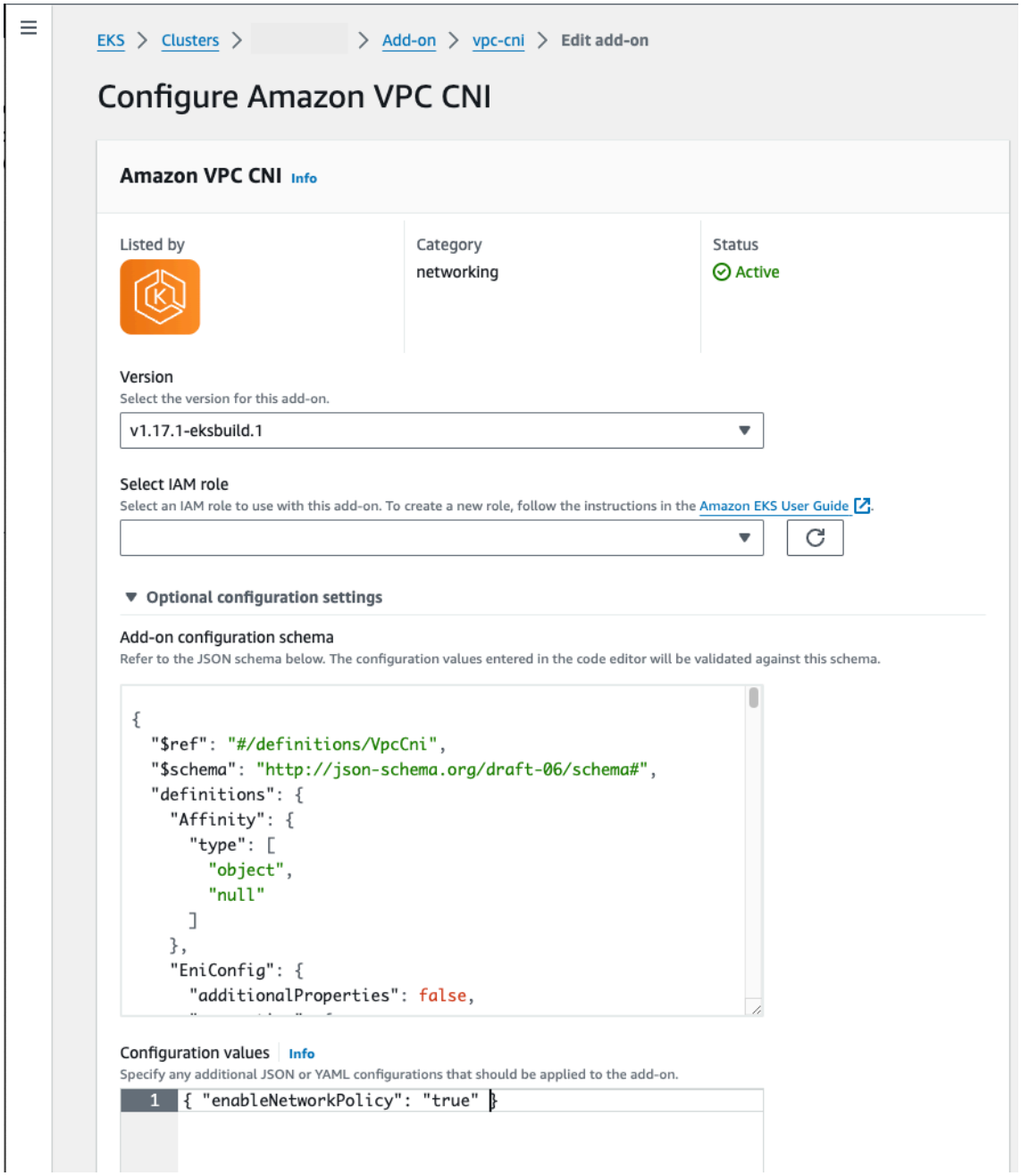
i. Wählen Sie `v1.14.0-eksbuild.3` oder eine neuere Version in der Dropdown-Liste aus.

ii. Erweitern Sie Optionale Konfigurationseinstellungen.

iii. Geben Sie den JSON-Schlüssel `"enableNetworkPolicy":` und den Wert `"true"` in Konfigurationswerte ein. Der resultierende Text muss ein gültiges JSON-Objekt sein. Wenn dieser Schlüssel und dieser Wert die einzigen Daten im Textfeld sind, setzen Sie den Schlüssel und den Wert in geschweifte Klammern `{}`. Das folgende Beispiel zeigt, dass die Netzwerkrichtlinie aktiviert ist:

```
{ "enableNetworkPolicy": "true" }
```

Der folgende Screenshot zeigt ein Beispiel für dieses Szenario.



AWS CLI

- Führen Sie den folgenden Befehl aus AWS CLI . Ersetzen Sie `my-cluster` durch den Namen Ihres Clusters und den IAM-Rollen-ARN durch die Rolle, die Sie verwenden.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni
 --addon-version v1.14.0-eksbuild.3 \
    --service-account-role-arn arn:aws:iam::123456789012:role/
AmazonEKSVPCCNIRole \
    --resolve-conflicts PRESERVE --configuration-values
 '{"enableNetworkPolicy": "true"}'
```

- Selbstverwaltetes Add-On

  Helm

  Wenn Sie das Amazon VPC CNI plugin for Kubernetes über `helm` installiert haben, können Sie die Konfiguration aktualisieren, um die Netzwerkrichtlinie zu aktivieren.

  - Führen Sie den folgenden Befehl aus, um die Netzwerkrichtlinie zu aktivieren.

    ```
    helm upgrade --set enableNetworkPolicy=true aws-vpc-cni --namespace
     kube-system eks/aws-vpc-cni
    ```

  kubectl

  a. Öffnen Sie das `ConfigMap` `amazon-vpc-cni` in Ihrem Editor.

     ```
     kubectl edit configmap -n kube-system amazon-vpc-cni -o yaml
     ```

  b. Fügen Sie `data` in `ConfigMap` die folgende Zeile hinzu:

     ```
     enable-network-policy-controller: "true"
     ```

     Sobald Sie die Zeile hinzugefügt haben, sollte `ConfigMap` wie das folgende Beispiel aussehen:

     ```
     apiVersion: v1
      kind: ConfigMap
      metadata:
       name: amazon-vpc-cni
       namespace: kube-system
      data:
     ```

```
  enable-network-policy-controller: "true"
```

c. Öffnen Sie das `DaemonSet` `aws-node` in Ihrem Editor.

```
kubectl edit daemonset -n kube-system aws-node
```

d. Ersetzen Sie im Befehlsargument `--enable-network-policy=false` in `args:` im Container `aws-network-policy-agent` im DaemonSet-Manifest `aws-node` von VPC CNI den Wert `false` durch `true`.

```
    - args:
       - --enable-network-policy=true
```

3. Vergewissern Sie sich, dass die `aws-node`-Pods auf Ihrem Cluster ausgeführt werden.

```
kubectl get pods -n kube-system | grep 'aws-node\|amazon'
```

Eine Beispielausgabe sieht wie folgt aus.

```
aws-node-gmqp7                                          2/2     Running   1 (24h
 ago)   24h
aws-node-prnsh                                          2/2     Running   1 (24h
 ago)   24h
```

Wenn die Netzwerkrichtlinie aktiviert ist, befinden sich 2 Container in den `aws-node`-Pods. In früheren Versionen und wenn die Netzwerkrichtlinie deaktiviert ist, gibt es nur einen einzigen Container in den `aws-node`-Pods.

Sie können jetzt Kubernetes-Netzwerkrichtlinien für Ihren Cluster bereitstellen. Weitere Informationen finden Sie unter Kubernetes-Netzwerkrichtlinien.

Stars-Demo der Netzwerkrichtlinie

Diese Demo erstellt einen Front-End-, Back-End- und Client-Service in Ihrem Amazon-EKS-Cluster. Außerdem wird in der Demo eine grafische Benutzeroberfläche für die Verwaltung erstellt, in der die verfügbaren Wege für ein- und ausgehenden Datenverkehr zwischen den einzelnen Services dargestellt werden. Wir empfehlen, die Demo in einem Cluster durchzuführen, in dem keine Produktions-Workloads ausgeführt werden.

Bevor Sie Netzwerkrichtlinien erstellen, können alle Services bidirektional kommunizieren. Nachdem Sie die Netzwerkrichtlinien angewendet haben, können Sie feststellen, dass der Client nur mit dem Front-End-Service kommuniziert und das Back-End nur Datenverkehr vom Front-End akzeptiert.

So führen Sie die Stars Policy-Demo aus

1. Wenden Sie die Services für Front-End, Back-End, Client und Verwaltungs-Benutzeroberfläche an:

```
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/namespace.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/management-ui.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/backend.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/frontend.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/client.yaml
```

2. Zeigen Sie alle Pods im Cluster an.

```
kubectl get pods -A
```

Eine Beispielausgabe sieht wie folgt aus.

In Ihrer Ausgabe sollten Pods in den Namespaces aufgeführt sein, die in der folgenden Ausgabe angezeigt werden. Die *NAMEN* Ihrer Pods und die Anzahl der Pods in der Spalte READY unterscheiden sich von den Angaben in der folgenden Ausgabe. Fahren Sie erst dann fort, wenn Pods mit ähnlichen Namen angezeigt werden und bei allen Running in der Spalte STATUS aufgeführt wird.

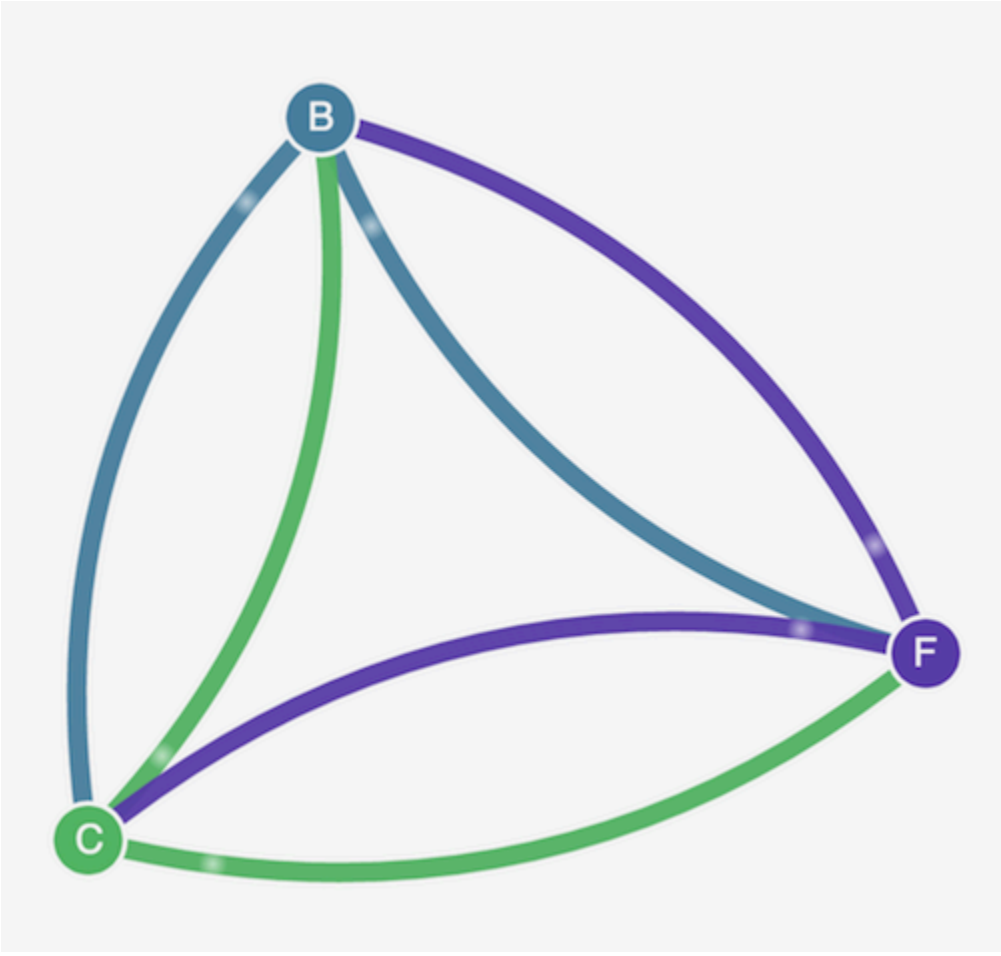
```
NAMESPACE         NAME                                       READY   STATUS
 RESTARTS   AGE
[...]
client            client-xlffc                               1/1     Running   0
       5m19s
[...]
management-ui     management-ui-qrb2g                        1/1     Running   0
       5m24s
```

```
stars              backend-sz87q                              1/1     Running   0
       5m23s
stars              frontend-cscnf                             1/1     Running   0
       5m21s
[...]
```

3. Zum Herstellen einer Verbindung mit der Verwaltungsbenutzeroberfläche stellen Sie eine Verbindung mit der `EXTERNAL-IP` des Services her, der auf Ihrem Cluster ausgeführt wird.

```
kubectl get service/management-ui -n management-ui
```

4. Öffnen Sie den Pfad aus dem vorherigen Schritt in einem Browser. Die Verwaltungs-Benutzeroberfläche sollte angezeigt werden. Der C--Knoten ist der Client-Service, der F-Knoten ist der Front-End-Service und der B--Knoten ist der Back-End-Service. Jeder Knoten verfügt über vollständigen Kommunikationszugriff auf alle anderen Knoten, wie durch die fett markierten, farbigen Zeilen angegeben.



5. Wenden Sie die folgende Netzwerkrichtlinie sowohl in den `stars`- als auch in den `client`-Namespaces an, um die Services voneinander zu isolieren:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny
spec:
  podSelector:
    matchLabels: {}
```

Sie können die folgenden Befehle verwenden, um die Richtlinie auf beide Namespaces anzuwenden:

```
kubectl apply -n stars -f https://eksworkshop.com/beginner/120_network-policies/
calico/stars_policy_demo/apply_network_policies.files/default-deny.yaml
kubectl apply -n client -f https://eksworkshop.com/beginner/120_network-policies/
calico/stars_policy_demo/apply_network_policies.files/default-deny.yaml
```

6. Aktualisieren Sie Ihren Browser. Wie Sie feststellen, kann nun keiner der Knoten mehr über die Verwaltungs-Benutzeroberfläche erreicht werden. Die Knoten werden daher nicht mehr in der Benutzeroberfläche angezeigt.
7. Wenden Sie die folgenden verschiedenen Netzwerkrichtlinien an, um der Verwaltungsbenutzeroberfläche den Zugriff auf die Services zu erlauben: Wenden Sie diese Richtlinie an, um die Benutzeroberfläche zuzulassen:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
  name: allow-ui
spec:
  podSelector:
    matchLabels: {}
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              role: management-ui
```

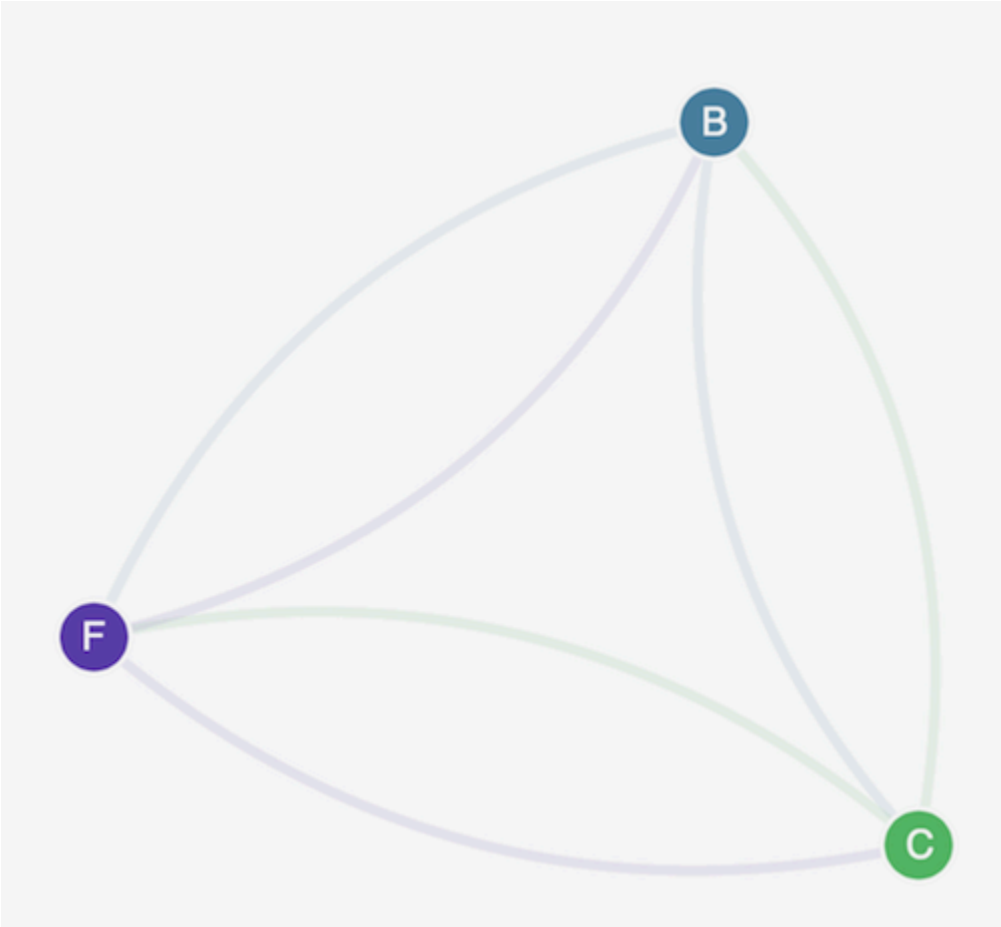
Wenden Sie diese Richtlinie an, um den Client zuzulassen:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: client
  name: allow-ui
spec:
  podSelector:
    matchLabels: {}
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              role: management-ui
```

Sie können die folgenden Befehle verwenden, um beide Richtlinien anzuwenden:

```
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/apply_network_policies.files/allow-ui.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/apply_network_policies.files/allow-ui-client.yaml
```

8. Aktualisieren Sie Ihren Browser. Wie Sie feststellen, können die Knoten nun wieder über die Verwaltungs-Benutzeroberfläche erreicht werden, sie können jedoch nicht miteinander kommunizieren.

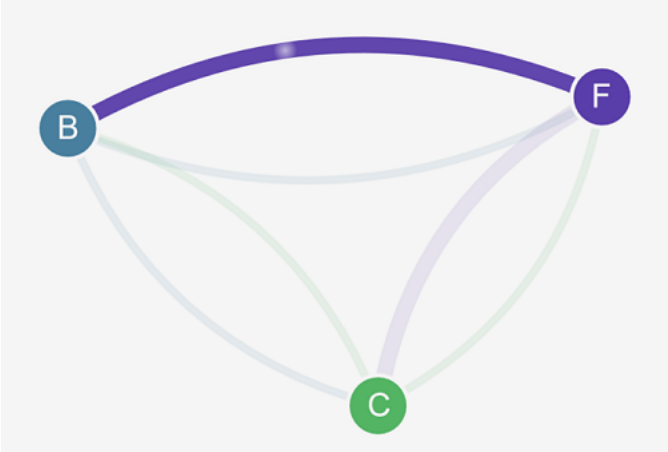


9. Wenden Sie die folgende Netzwerkrichtlinie an, um Verkehr vom Front-End-Service zum Back-End-Service zuzulassen:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
  name: backend-policy
spec:
  podSelector:
    matchLabels:
      role: backend
  ingress:
    - from:
        - podSelector:
            matchLabels:
              role: frontend
      ports:
        - protocol: TCP
```

```
          port: 6379
```

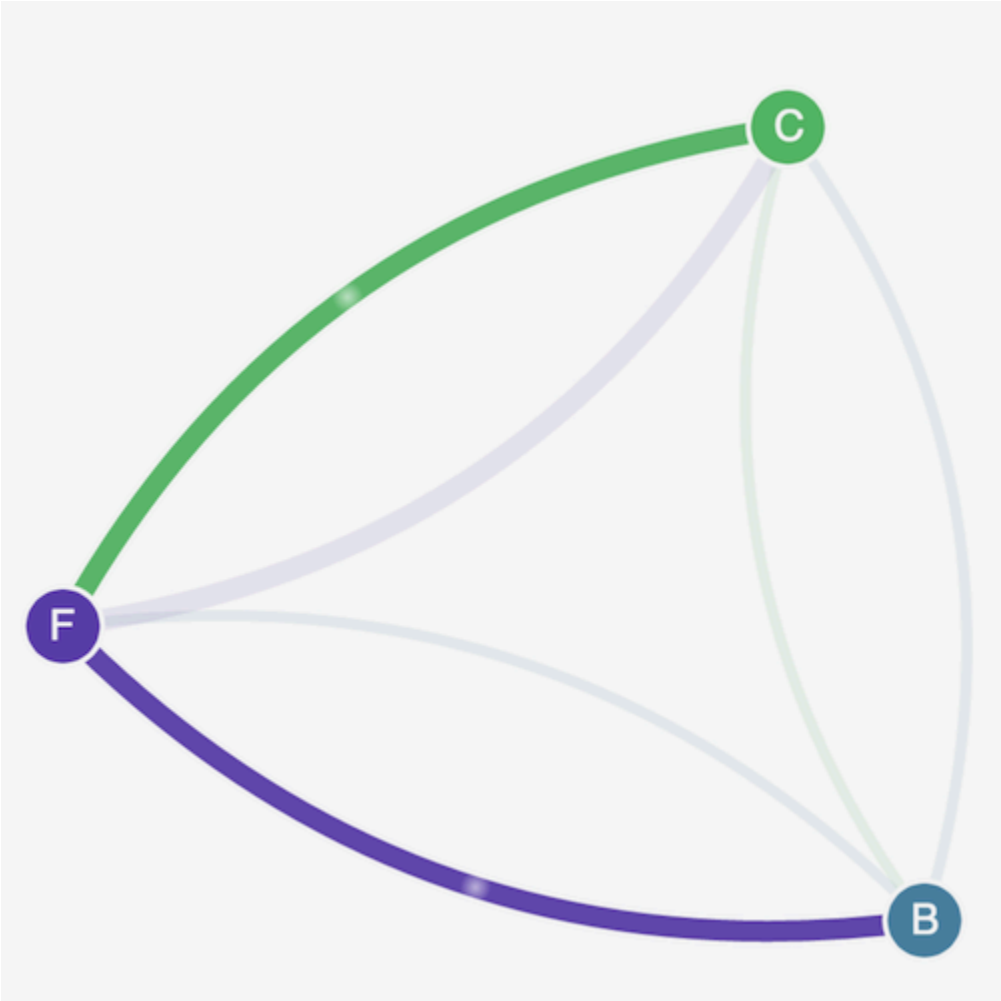
10. Aktualisieren Sie Ihren Browser. Wie Sie feststellen, kann das Front-End mit dem Back-End kommunizieren.



11. Wenden Sie die folgende Netzwerkrichtlinie an, um Datenverkehr vom Client zum Front-End-Service zuzulassen:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
  name: frontend-policy
spec:
  podSelector:
    matchLabels:
      role: frontend
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              role: client
      ports:
        - protocol: TCP
          port: 80
```

12. Aktualisieren Sie Ihren Browser. Wie Sie feststellen, kann der Client mit dem Front-End-Service kommunizieren. Der Front-End-Service kann nach wie vor mit dem Back-End-Service kommunizieren.



13. (Optional) Wenn Sie die Demo abgeschlossen haben, können Sie die zugehörigen Ressourcen löschen.

```
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/client.yaml
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/frontend.yaml
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/backend.yaml
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/management-ui.yaml
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/namespace.yaml
```

Auch nach dem Löschen der Ressourcen können noch Netzwerkrichtlinien-Endpunkte auf den Knoten vorhanden sein, die die Vernetzung in Ihrem Cluster auf unerwartete Weise stören

können. Die einzige sichere Möglichkeit, diese Regeln zu entfernen, besteht darin, die Knoten neu zu starten oder alle Knoten zu beenden und sie zu recyceln. Um alle Knoten zu beenden, legen Sie entweder die Anzahl der gewünschten Auto Scaling-Gruppen auf 0 und anschließend wieder auf die gewünschte Zahl fest oder beenden einfach die Knoten.

Fehlerbehebung bei Netzwerkrichtlinien

Sie können Netzwerkverbindungen, die Netzwerkrichtlinien verwenden, durch Lesen der Netzwerkrichtlinien-Protokolle und durch Ausführen von Tools aus dem eBPF-SDK untersuchen und Fehler beheben.

Netzwerkrichtlinien-Protokolle

Ob Verbindungen durch eine Netzwerkrichtlinie zugelassen oder verweigert werden, wird in Flussprotokollen protokolliert. Die Netzwerkrichtlinien-Protokolle auf jedem Knoten enthalten die Flussprotokolle für jeden Pod, der über eine Netzwerkrichtlinie verfügt. Netzwerkrichtlinien-Protokolle werden unter `/var/log/aws-routed-eni/network-policy-agent.log` gespeichert. Das folgende Beispiel stammt aus einer `network-policy-agent.log`-Datei:

```
{"level":"info","timestamp":"2023-05-30T16:05:32.573Z","logger":"ebpf-
client","msg":"Flow Info: ","Src
IP":"192.168.87.155","Src Port":38971,"Dest IP":"64.6.160","Dest
Port":53,"Proto":"UDP","Verdict":"ACCEPT"}
```

Netzwerkrichtlinien-Protokolle sind standardmäßig deaktiviert. Gehen Sie wie folgt vor, um die Netzwerkrichtlinienprotokolle zu aktivieren:

> **Note**
>
> Netzwerkrichtlinienprotokolle erfordern eine zusätzliche vCPU für den `aws-network-policy-agent` Container im VPC `aws-node` CNI-Daemonset-Manifest.

Amazon-EKS-Add-On

AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

2. Wählen Sie im linken Navigationsbereich die Option Cluster aus. Wählen Sie dann den Namen des Clusters aus, für den Sie das Amazon-VPC-CNI-Add-on konfigurieren möchten.
3. Wählen Sie die Registerkarte Add-ons.
4. Wählen Sie das Kästchen oben rechts in der Add-On-Box aus und wählen Sie dann Edit (Bearbeiten).
5. Gehen Sie auf der Seite Configure ***name of addon*** (Namen des Add-Ons konfigurieren) wie folgt vor:

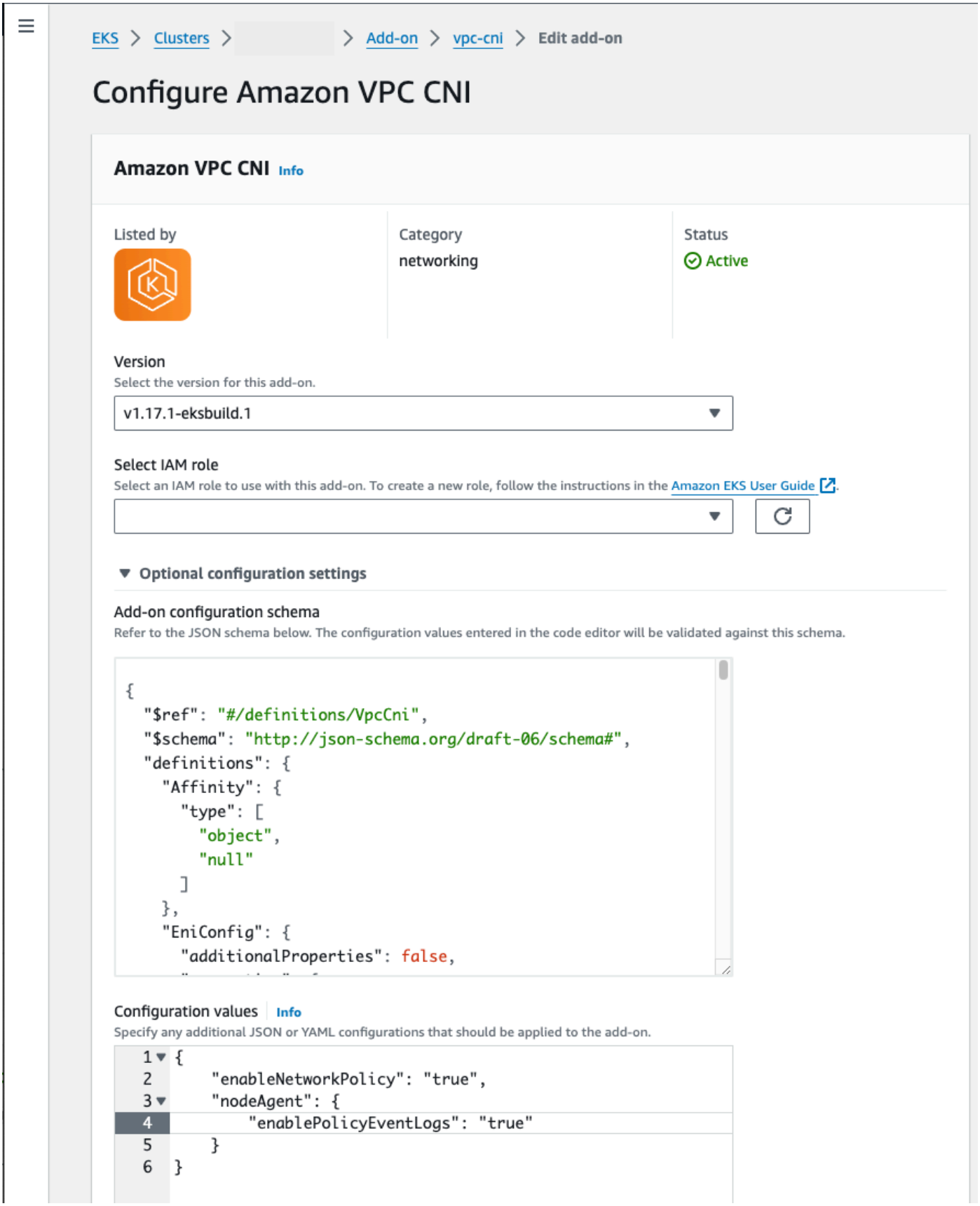
    a. Wählen Sie `v1.14.0-eksbuild.3` oder eine neuere Version in der Dropdown-Liste aus.

    b. Erweitern Sie Optionale Konfigurationseinstellungen.

    c. Geben Sie den JSON-Schlüssel der obersten Ebene `"nodeAgent":` ein, und der Wert ist ein Objekt mit dem Schlüssel `"enablePolicyEventLogs":` und dem Wert `"true"` in Konfigurationswerte. Der resultierende Text muss ein gültiges JSON-Objekt sein. Das folgende Beispiel zeigt, dass Netzwerkrichtlinien und Netzwerkrichtlinien-Protokolle aktiviert sind und die Netzwerkrichtlinien-Protokolle an Logs gesendet werden: CloudWatch

    ```
    {
        "enableNetworkPolicy": "true",
        "nodeAgent": {
            "enablePolicyEventLogs": "true"
        }
    }
    ```

Der folgende Screenshot zeigt ein Beispiel für dieses Szenario.


EKS > Clusters > > Add-on > vpc-cni > Edit add-on
Configure Amazon VPC CNI
Amazon VPC CNI Info
Listed by
Category
networking
Status
Active
Version
Select the version for this add-on.
v1.17.1-eksbuild.1
Select IAM role
Select an IAM role to use with this add-on. To create a new role, follow the instructions in the Amazon EKS User Guide.
Optional configuration settings
Add-on configuration schema
Refer to the JSON schema below. The configuration values entered in the code editor will be validated against this schema.
{
"$ref": "#/definitions/VpcCni",
"$schema": "http://json-schema.org/draft-06/schema#",
"definitions": {
"Affinity": {
"type": [
"object",
"null"
]
},
"EniConfig": {
"additionalProperties": false,
Configuration values Info
Specify any additional JSON or YAML configurations that should be applied to the add-on.
1 {
2 "enableNetworkPolicy": "true",
3 "nodeAgent": {
4 "enablePolicyEventLogs": "true"
5 }
6 }

AWS CLI

- Führen Sie den folgenden Befehl aus AWS CLI . Ersetzen Sie `my-cluster` durch den Namen Ihres Clusters und den IAM-Rollen-ARN durch die Rolle, die Sie verwenden.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version v1.14.0-eksbuild.3 \
    --service-account-role-arn arn:aws:iam::123456789012:role/
AmazonEKSVPCCNIRole \
    --resolve-conflicts PRESERVE --configuration-values '{"nodeAgent":
 {"enablePolicyEventLogs": "true"}}'
```

Selbstverwaltetes Add-On

Helm

Wenn Sie den Amazon VPC CNI plugin for Kubernetes Through installiert haben`helm`, können Sie die Konfiguration aktualisieren, um die Netzwerkrichtlinien-Protokolle zu schreiben.

- Führen Sie den folgenden Befehl aus, um die Netzwerkrichtlinie zu aktivieren.

```
helm upgrade --set nodeAgent.enablePolicyEventLogs=true aws-vpc-cni --namespace
 kube-system eks/aws-vpc-cni
```

kubectl

Wenn Sie den Amazon VPC CNI plugin for Kubernetes Durchgang installiert haben`kubectl`, können Sie die Konfiguration aktualisieren, um die Netzwerkrichtlinienprotokolle zu schreiben.

1. Öffnen Sie das `DaemonSet` `aws-node` in Ihrem Editor.

```
kubectl edit daemonset -n kube-system aws-node
```

2. Ersetzen Sie im Befehlsargument `--enable-policy-event-logs=false` in `args:` im Container `aws-network-policy-agent` im DaemonSet-Manifest `aws-node` von VPC CNI den Wert `false` durch `true`.

```
     - args:
```

```
        - --enable-policy-event-logs=true
```

Netzwerkrichtlinien-Protokolle an Amazon CloudWatch Logs senden

Sie können die Netzwerkrichtlinienprotokolle mithilfe von Diensten wie Amazon CloudWatch Logs überwachen. Sie können die folgenden Methoden verwenden, um die Netzwerkrichtlinien-Protokolle an Logs zu CloudWatch senden.

Bei EKS-Clustern befinden sich die Richtlinienprotokolle unter `/aws/eks/`*`cluster-name`*`/cluster/` und bei selbstverwalteten K8S-Clustern unter `/aws/k8s-cluster/cluster`.

Netzwerkrichtlinien-Protokolle mit Amazon VPC CNI plugin for Kubernetes senden

Wenn Sie die Netzwerkrichtlinie aktivieren, wird den `aws-node`-Pods ein zweiter Container für einen Konten-Agent hinzugefügt. Dieser Node-Agent kann die CloudWatch Netzwerkrichtlinien-Protokolle an Logs senden.

> **Note**
>
> Nur die Netzwerkrichtlinien-Protokolle werden vom Knoten-Agent gesendet. Andere von VPC CNI erstellte Protokolle sind nicht enthalten.

Voraussetzungen

- Fügen Sie der IAM-Rolle, die Sie für VPC CNI verwenden, die folgenden Berechtigungen als Abschnitt oder separate Richtlinie hinzu.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

Amazon-EKS-Add-On

AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus. Wählen Sie dann den Namen des Clusters aus, für den Sie das Amazon-VPC-CNI-Add-on konfigurieren möchten.
3. Wählen Sie die Registerkarte Add-ons.
4. Wählen Sie das Kästchen oben rechts in der Add-On-Box aus und wählen Sie dann Edit (Bearbeiten).
5. Gehen Sie auf der Seite Configure ***name of addon*** (Namen des Add-Ons konfigurieren) wie folgt vor:

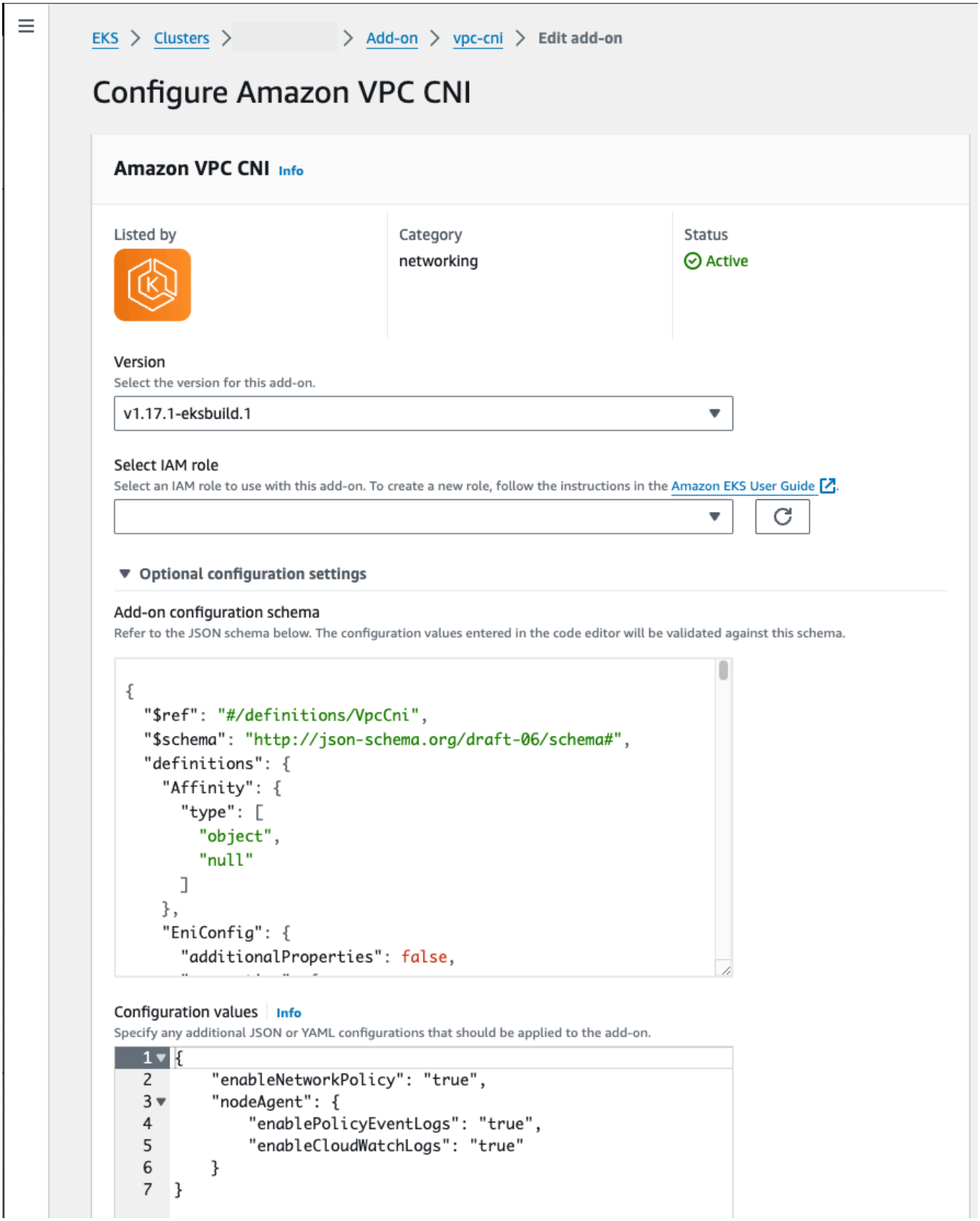
   a. Wählen Sie `v1.14.0-eksbuild.3` oder eine neuere Version in der Dropdown-Liste aus.

   b. Erweitern Sie Optionale Konfigurationseinstellungen.

   c. Geben Sie den JSON-Schlüssel der obersten Ebene `"nodeAgent":` ein, und der Wert ist ein Objekt mit dem Schlüssel `"enableCloudWatchLogs":` und dem Wert `"true"` in Konfigurationswerte. Der resultierende Text muss ein gültiges JSON-Objekt sein. Das folgende Beispiel zeigt, dass Netzwerkrichtlinien und Netzwerkrichtlinien-Protokolle aktiviert sind und die Protokolle an CloudWatch Logs gesendet werden:

   ```
   {
       "enableNetworkPolicy": "true",
       "nodeAgent": {
           "enablePolicyEventLogs": "true",
           "enableCloudWatchLogs": "true",
       }
   }
   ```

   Der folgende Screenshot zeigt ein Beispiel für dieses Szenario.


EKS > Clusters > > Add-on > vpc-cni > Edit add-on
Configure Amazon VPC CNI
Amazon VPC CNI Info
Listed by
Category
networking
Status
Active
Version
Select the version for this add-on.
v1.17.1-eksbuild.1
Select IAM role
Select an IAM role to use with this add-on. To create a new role, follow the instructions in the Amazon EKS User Guide.
Optional configuration settings
Add-on configuration schema
Refer to the JSON schema below. The configuration values entered in the code editor will be validated against this schema.
{
"$ref": "#/definitions/VpcCni",
"$schema": "http://json-schema.org/draft-06/schema#",
"definitions": {
"Affinity": {
"type": [
"object",
"null"
]
},
"EniConfig": {
"additionalProperties": false,
Configuration values Info
Specify any additional JSON or YAML configurations that should be applied to the add-on.
1 {
2 "enableNetworkPolicy": "true",
3 "nodeAgent": {
4 "enablePolicyEventLogs": "true",
5 "enableCloudWatchLogs": "true"
6 }
7 }

AWS CLI

- Führen Sie den folgenden Befehl aus AWS CLI . Ersetzen Sie `my-cluster` durch den Namen Ihres Clusters und den IAM-Rollen-ARN durch die Rolle, die Sie verwenden.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version v1.14.0-eksbuild.3 \
    --service-account-role-arn arn:aws:iam::123456789012:role/
AmazonEKSVPCCNIRole \
    --resolve-conflicts PRESERVE --configuration-values '{"nodeAgent":
 {"enablePolicyEventLogs": "true", "enableCloudWatchLogs": "true"}}'
```

Selbstverwaltetes Add-On

Helm

Wenn Sie den Amazon VPC CNI plugin for Kubernetes Through installiert haben`helm`, können Sie die Konfiguration so aktualisieren, dass CloudWatch Netzwerkrichtlinienprotokolle an Logs gesendet werden.

- Führen Sie den folgenden Befehl aus, um Netzwerkrichtlinien-Protokolle zu aktivieren und sie an CloudWatch Logs zu senden.

```
helm upgrade --set nodeAgent.enablePolicyEventLogs=true --set
 nodeAgent.enableCloudWatchLogs=true aws-vpc-cni --namespace kube-system eks/
aws-vpc-cni
```

kubectl

1. Öffnen Sie das `DaemonSet` `aws-node` in Ihrem Editor.

```
kubectl edit daemonset -n kube-system aws-node
```

2. Ersetzen Sie das `false` durch `true` in zwei Befehlsargumenten `--enable-policy-event-logs=false` und `--enable-cloudwatch-logs=false` im `aws-network-policy-agent` Container `args:` im VPC `aws-node` CNI-Daemonset-Manifest.

```
    - args:
```

```
        - --enable-policy-event-logs=true
        - --enable-cloudwatch-logs=true
```

Netzwerkrichtlinien-Protokolle mit einem Fluent Bit-DaemonSet senden

Wenn Sie Fluent Bit in einem Daemonset verwenden, um Protokolle von Ihren Knoten zu senden, können Sie eine Konfiguration hinzufügen, um die Netzwerkrichtlinien-Protokolle von Netzwerkrichtlinien einzuschließen. Sie können die folgende Beispielkonfiguration verwenden:

```
    [INPUT]
        Name              tail
        Tag               eksnp.*
        Path              /var/log/aws-routed-eni/network-policy-agent*.log
        Parser            json
        DB                /var/log/aws-routed-eni/flb_npagent.db
        Mem_Buf_Limit     5MB
        Skip_Long_Lines   On
        Refresh_Interval  10
```

eBPF-SDK eingeschlossen

Das Amazon VPC CNI plugin for Kubernetes installiert eine Sammlung von eBPF-SDK-Tools auf den Knoten. Sie können die eBPF-SDK-Tools verwenden, um Probleme mit Netzwerkrichtlinien zu identifizieren. Der folgende Befehl listet zum Beispiel die Programme auf, die auf dem Knoten ausgeführt werden.

```
sudo /opt/cni/bin/aws-eks-na-cli ebpf progs
```

Zum Ausführen dieses Befehls können Sie eine beliebige Methode verwenden, um eine Verbindung mit dem Knoten herzustellen.

Kubernetes-Netzwerkrichtlinien

Zum Implementieren von Kubernetes-Netzwerkrichtlinien erstellen Sie Kubernetes `NetworkPolicy`-Objekte und stellen diese in Ihrem Cluster bereit. `NetworkPolicy`-Objekte sind auf einen Namespace beschränkt. Sie implementieren Richtlinien, um Datenverkehr zwischen Pods auf der Grundlage von Bezeichnungsselektoren, Namespaces und IP-Adressbereichen zuzulassen oder zu verweigern. Weitere Informationen zum Erstellen von `NetworkPolicy`-Objekten finden Sie unter Netzwerkrichtlinien in der Kubernetes-Dokumentation.

Die Durchsetzung von Kubernetes `NetworkPolicy`-Objekten wird mit dem Extended Berkeley Packet Filter (eBPF) implementiert. Im Vergleich zu `iptables`-basierten Implementierungen bietet er geringere Latenzzeiten und Leistungsmerkmale, einschließlich einer geringeren CPU-Auslastung und der Vermeidung sequenzieller Suchvorgänge. Darüber hinaus bieten eBPF-Probes Zugriff auf kontextreiche Daten, die bei der Fehlersuche in komplexen Problemen auf Kernel-Ebene helfen und die Beobachtbarkeit verbessern. Amazon EKS unterstützt einen eBPF-basierten Exporter, der die Probes nutzt, um Richtlinienergebnisse auf jedem Knoten zu protokollieren und die Daten an externe Protokollsammler zu exportieren, um bei der Fehlersuche zu helfen. Weitere Informationen finden Sie in der eBPF-Dokumentation.

Benutzerdefinierte Netzwerke für Pods

Standardmäßig erstellt das Amazon VPC CNI plugin for Kubernetes sekundäre Elastic-Network-Schnittstellen (Netzwerkschnittstellen) für Ihren Amazon EC2-Knoten im gleichen Subnetz wie die primäre Netzwerkschnittstelle des Knotens. Es verknüpft auch dieselben Sicherheitsgruppen mit der sekundären Netzwerkschnittstelle, die mit der primären Netzwerkschnittstelle verknüpft sind. Aus einem oder mehreren der folgenden Gründe möchten Sie möglicherweise, dass das Plugin sekundäre Netzwerkschnittstellen in einem anderen Subnetz erstellt oder Sie möchten den sekundären Netzwerkschnittstellen oder beiden verschiedene Sicherheitsgruppen zuordnen:

- Es gibt eine begrenzte Anzahl `IPv4`-Adressen, die in dem Subnetz verfügbar sind, in dem sich die primäre Netzwerkschnittstelle befindet. Dies könnte die Anzahl der Pods einschränken, die Sie in dem Subnetz erstellen können. Durch Verwendung eines anderen Subnetzes für sekundäre Netzwerkschnittstellen können Sie die Anzahl der verfügbaren `IPv4`-Adressen für Pods erhöhen.
- Aus Sicherheitsgründen müssen Ihre Pods unter Umständen andere Sicherheitsgruppen oder Subnetze verwenden als die primäre Netzwerkschnittstelle des Knotens.
- Die Knoten sind in öffentlichen Subnetzen konfiguriert und die Pods sollen in privaten Subnetzen platziert werden. Die einem öffentlichen Subnetz zugeordnete Routing-Tabelle enthält eine Route zu einem Internet-Gateway. Die einem privaten Subnetz zugeordnete Routing-Tabelle enthält keine Route zu einem Internet-Gateway.

Überlegungen

- Wenn benutzerdefinierte Netzwerke aktiviert sind, werden Pods keine IP-Adressen zugewiesen, die der primären Netzwerkschnittstelle zugewiesen sind. `Pods` werden nur IP-Adressen von sekundären Netzwerkschnittstellen zugewiesen.

- Wenn Ihr Cluster die IPv6-Familie verwendet, können Sie keine benutzerdefinierten Netzwerke verwenden.
- Wenn Sie vorhaben, benutzerdefinierte Netzwerke zu verwenden, um die IPv4-Adresserschöpfung zu verhindern, können Sie stattdessen einen Cluster über die IPv6-Familie erstellen. Weitere Informationen finden Sie unter IPv6Adressen für ClusterPods, und services.
- Auch wenn in Subnetzen für sekundäre Netzwerkschnittstellen bereitgestellte Pods andere Subnetz- und Sicherheitsgruppen verwenden können als die primäre Netzwerkschnittstelle des Knotens, müssen sich die Subnetze und Sicherheitsgruppen in derselben VPC befinden wie der Knoten.

Voraussetzungen

- Sie müssen damit vertraut sein, wie das Amazon VPC CNI plugin for Kubernetes sekundäre Netzwerkschnittstellen erstellt und Pods IP-Adressen zuweist. Weitere Informationen finden Sie unter ENI-Zuweisung auf GitHub.
- Version 2.12.3 oder höher oder Version 1.27.160 oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **aws --version | cut -d / -f2 | cut -d ' ' -f1**. Paket-Manager wie yum, apt-get oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Das kubectl-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise 1.29 ist, können Sie kubectl-Version 1.28, 1.29, oder 1.30 damit verwenden. Informationen zum Installieren oder Aktualisieren von kubectl finden Sie unter Installieren oder Aktualisieren von kubectl.
- Wir empfehlen, dass Sie die Schritte in diesem Thema in einer Bash-Shell ausführen. Wenn Sie keine Bash-Shell verwenden, erfordern einige Skriptbefehle wie Zeilenfortsetzungszeichen und die Art und Weise, wie Variablen gesetzt und verwendet werden, eine Anpassung für Ihre Shell. Darüber hinaus können die Zitier- und Escape-Regeln für Ihre Shell unterschiedlich sein. Weitere

Informationen finden Sie im Benutzerhandbuch unter Verwenden von Anführungszeichen mit Zeichenfolgen. AWS CLI AWS Command Line Interface

Für dieses Tutorial empfehlen wir die Verwendung der *example values*, außer es gibt eine Anmerkung, diese zu ersetzen. Sie können jeden *example value* ersetzen, wenn Sie die Schritte für einen Produktions-Cluster ausführen. Wir empfehlen, alle Schritte auf demselben Terminal auszuführen. Grund dafür ist, dass Variablen während der Schritte festgelegt und verwendet werden und diese in anderen Terminals nicht vorhanden sind.

Die Befehle in diesem Thema sind gemäß den Konventionen formatiert, die unter Verwenden der AWS CLI Beispiele aufgeführt sind. Wenn Sie Befehle über die Befehlszeile für Ressourcen ausführen, die sich AWS-Region nicht in der Standardeinstellung befinden, die in dem AWS CLI von Ihnen verwendeten Profil AWS-Region definiert ist, müssen Sie weitere Befehle hinzufügen**--region** ***region-code***.

Wenn Sie benutzerdefinierte Netzwerke in Ihrem Produktions-Cluster bereitstellen möchten, fahren Sie mit Schritt 2: Konfigurieren Ihrer VPC fort.

Schritt 1: Erstellen einer Test-VPC und eines Clusters

So erstellen Sie einen Cluster

Mit den folgenden Verfahren können Sie eine Test-VPC und einen Cluster erstellen und benutzerdefinierte Netzwerke für diesen Cluster konfigurieren. Wir empfehlen, den Test-Cluster nicht für Produktions-Workloads zu verwenden, da in diesem Thema verschiedene nicht verwandte Features nicht behandelt werden, die Sie vielleicht in Ihrem Produktions-Cluster verwenden. Weitere Informationen finden Sie unter Erstellen eines Amazon-EKS-Clusters.

1. Definieren Sie einige Variablen, die in den verbleibenden Schritten verwendet werden sollen.

```
export cluster_name=my-custom-networking-cluster
account_id=$(aws sts get-caller-identity --query Account --output text)
```

2. Erstellen Sie eine VPC.

   1. Erstellen Sie eine VPC mit einer Amazon AWS CloudFormation EKS-Vorlage.

```
aws cloudformation create-stack --stack-name my-eks-custom-networking-vpc \
  --template-url https://s3.us-west-2.amazonaws.com/amazon-
eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml \
```

```
  --parameters ParameterKey=VpcBlock,ParameterValue=192.168.0.0/24 \
  ParameterKey=PrivateSubnet01Block,ParameterValue=192.168.0.64/27 \
  ParameterKey=PrivateSubnet02Block,ParameterValue=192.168.0.96/27 \
  ParameterKey=PublicSubnet01Block,ParameterValue=192.168.0.0/27 \
  ParameterKey=PublicSubnet02Block,ParameterValue=192.168.0.32/27
```

Die Erstellung des AWS CloudFormation Stacks dauert einige Minuten. Führen Sie den folgenden Befehl aus, um den Status der Bereitstellung des Stacks zu überprüfen.

```
aws cloudformation describe-stacks --stack-name my-eks-custom-networking-vpc --
query Stacks\[\].StackStatus  --output text
```

Fahren Sie erst mit dem nächsten Schritt fort, wenn die Ausgabe des Befehls `CREATE_COMPLETE` ist.

2. Definieren Sie Variablen mit den Werten der von der Vorlage erstellten privaten Subnetz-IDs.

```
subnet_id_1=$(aws cloudformation describe-stack-resources --stack-name my-eks-
custom-networking-vpc \
    --query "StackResources[?
LogicalResourceId=='PrivateSubnet01'].PhysicalResourceId" --output text)
subnet_id_2=$(aws cloudformation describe-stack-resources --stack-name my-eks-
custom-networking-vpc \
    --query "StackResources[?
LogicalResourceId=='PrivateSubnet02'].PhysicalResourceId" --output text)
```

3. Definieren Sie Variablen mit den Availability Zones der im vorherigen Schritt abgerufenen Subnetze.

```
az_1=$(aws ec2 describe-subnets --subnet-ids $subnet_id_1 --query
 'Subnets[*].AvailabilityZone' --output text)
az_2=$(aws ec2 describe-subnets --subnet-ids $subnet_id_2 --query
 'Subnets[*].AvailabilityZone' --output text)
```

3. Erstellen Sie eine Cluster-IAM-Rolle.

   a. Führen Sie den folgenden Befehl aus, um eine JSON-Datei für eine IAM-Vertrauensrichtlinie zu erstellen.

```
cat >eks-cluster-role-trust-policy.json <<EOF
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

b. Erstellen Sie die Amazon-EKS-Cluster-IAM-Rolle. Falls erforderlich, stellen Sie `eks-cluster-role-trust-policy.json` den Pfad auf Ihrem Computer voran, in den Sie im vorherigen Schritt die Datei geschrieben haben. Der Befehl verknüpft die im vorherigen Schritt erstellte Vertrauensrichtlinie mit der Rolle. Um eine IAM-Rolle zu erstellen, muss dem IAM-Prinzipal, der die Rolle erstellt, die `iam:CreateRole`-Aktion (Berechtigung) zugewiesen werden.

```
aws iam create-role --role-name myCustomNetworkingAmazonEKSClusterRole --
assume-role-policy-document file://"eks-cluster-role-trust-policy.json"
```

c. Hängen Sie die von Amazon EKS verwaltete Richtlinie `AmazonEKSClusterPolicy` an die Rolle an. Um eine IAM-Richtlinie an einen IAM-Prinzipal anzuhängen, muss dem Prinzipal, der die Richtlinie anhängt, eine der folgenden IAM-Aktionen (Berechtigungen) zugewiesen werden: `iam:AttachUserPolicy` oder `iam:AttachRolePolicy`.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSClusterPolicy --role-name myCustomNetworkingAmazonEKSClusterRole
```

4. Erstellen Sie einen Amazon-EKS-Cluster und konfigurieren Sie Ihr Gerät für die Kommunikation mit diesem.

a. Erstellen Sie einen Cluster.

```
aws eks create-cluster --name my-custom-networking-cluster \
   --role-arn arn:aws:iam::$account_id:role/
myCustomNetworkingAmazonEKSClusterRole \
   --resources-vpc-config subnetIds=$subnet_id_1","$subnet_id_2
```

> **Note**
>
> Sie erhalten möglicherweise eine Fehlermeldung, dass eine der Availability Zones in Ihrer Anfrage nicht über genügend Kapazität zum Erstellen eines Amazon-EKS-Clusters verfügt. Wenn dies der Fall ist, enthält die Fehlerausgabe die Availability Zones, die einen neuen Cluster unterstützen können. Versuchen Sie, Ihren Cluster mit mindestens zwei Subnetzen erneut zu erstellen, die sich in den unterstützten Availability Zones für Ihr Konto befinden. Weitere Informationen finden Sie unter Unzureichende Kapazität.

b. Die Erstellung des Clusters dauert mehrere Minuten. Führen Sie den folgenden Befehl aus, um den Status der Bereitstellung des Clusters zu überprüfen.

```
aws eks describe-cluster --name my-custom-networking-cluster --query
 cluster.status
```

Fahren Sie erst mit dem nächsten Schritt fort, wenn die Ausgabe des Befehls `"ACTIVE"` ist.

c. Konfigurieren Sie `kubectl` für die Kommunikation mit Ihrem Cluster.

```
aws eks update-kubeconfig --name my-custom-networking-cluster
```

Schritt 2: Konfigurieren Ihrer VPC

Für dieses Tutorial wird die VPC in Schritt 1: Erstellen einer Test-VPC und eines Clusters erstellt. Passen Sie für einen Produktions-Cluster die Schritte entsprechend Ihrer VPC an, indem Sie alle *`example values`* durch eigene Werte ersetzen.

1. Bestätigen Sie, dass Ihr derzeit installiertes Amazon VPC CNI plugin for Kubernetes die neueste Version ist. Informationen zur Ermittlung der neuesten Version für den Amazon-EKS-Add-On-Typ und zur Aktualisierung Ihrer Version auf diese Version finden Sie unter Aktualisieren eines Add-Ons. Informationen dazu, wie Sie die neueste Version für den selbstverwalteten Add-On-Typ ermitteln und Ihre Version darauf aktualisieren können, finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.
2. Rufen Sie die ID Ihrer Cluster-VPC ab und speichern Sie diese zur Verwendung in späteren Schritten in einer Variable. Ersetzen Sie für einen Produktions-Cluster *`my-custom-networking-cluster`* durch den Namen Ihres Clusters.

```
vpc_id=$(aws eks describe-cluster --name my-custom-networking-cluster --query
 "cluster.resourcesVpcConfig.vpcId" --output text)
```

3. Verknüpfen Sie einen zusätzlichen Classless Inter-Domain Routing (CIDR)-Block mit der VPC Ihres Clusters. Der CIDR-Block kann sich nicht mit vorhandenen zugewiesenen CIDR-Blöcken überschneiden.

   1. Zeigen Sie die aktuellen CIDR-Blöcke an, die Ihrer VPC zugeordnet sind.

      ```
      aws ec2 describe-vpcs --vpc-ids $vpc_id \
          --query 'Vpcs[*].CidrBlockAssociationSet[*].{CIDRBlock: CidrBlock, State:
       CidrBlockState.State}' --out table
      ```

      Eine Beispielausgabe sieht wie folgt aus.

      ```
      ---------------------------------
      |          DescribeVpcs          |
      +-----------------+--------------+
      |    CIDRBlock    |    State     |
      +-----------------+--------------+
      |  192.168.0.0/24 |  associated  |
      +-----------------+--------------+
      ```

   2. Ordnen Sie Ihrer VPC zusätzliche CIDR-Blöcke zu. Weitere Informationen finden Sie unter Zuordnen zusätzlicher IPv4 CIDR-Blöcke zu Ihrer VPC im Amazon-VPC-Benutzerhandbuch.

      ```
      aws ec2 associate-vpc-cidr-block --vpc-id $vpc_id --cidr-block 192.168.1.0/24
      ```

   3. Bestätigen Sie, dass der neue Block zugeordnet ist.

      ```
      aws ec2 describe-vpcs --vpc-ids $vpc_id --query
       'Vpcs[*].CidrBlockAssociationSet[*].{CIDRBlock: CidrBlock, State:
       CidrBlockState.State}' --out table
      ```

      Eine Beispielausgabe sieht wie folgt aus.

      ```
      ---------------------------------
      |          DescribeVpcs          |
      +-----------------+--------------+
      |    CIDRBlock    |    State     |
      ```

```
+-----------------+-------------+
|  192.168.0.0/24 |  associated  |
|  192.168.1.0/24 |  associated  |
+-----------------+-------------+
```

Fahren Sie erst mit dem nächsten Schritt fort, wenn der `State` Ihrer neuen CIDR-Blöcke `associated` ist.

4. Erstellen Sie so viele Subnetze, wie Sie in jeder Availability Zone verwenden möchten, in denen sich Ihre vorhandenen Subnetze befinden. Geben Sie einen CIDR-Block an, der sich innerhalb des CIDR-Blocks befindet, den Sie in einem vorherigen Schritt mit Ihrer VPC verknüpft haben.

   1. Erstellen Sie neue Subnetze. Die Subnetze müssen in einem anderen VPC-CIDR-Block erstellt werden als Ihre vorhandenen Subnetze, jedoch in denselben Availability Zones wie Ihre vorhandenen Subnetze. In diesem Beispiel wird ein Subnetz im neuen CIDR-Block jeder Availability Zone erstellt, in der die aktuellen privaten Subnetze vorhanden sind. Die IDs der erstellten Subnetze werden zur Verwendung in späteren Schritten in Variablen gespeichert. Die `Name`-Werte entsprechen den Werten, die den Subnetzen zugewiesen wurden, die in einem vorherigen Schritt mit der Amazon-EKS-VPC-Vorlage erstellt wurden. Namen sind nicht erforderlich. Sie können verschiedene Namen verwenden.

```
new_subnet_id_1=$(aws ec2 create-subnet --vpc-id $vpc_id --availability-zone
 $az_1 --cidr-block 192.168.1.0/27 \
    --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=my-eks-
custom-networking-vpc-PrivateSubnet01},{Key=kubernetes.io/role/internal-
elb,Value=1}]' \
    --query Subnet.SubnetId --output text)
new_subnet_id_2=$(aws ec2 create-subnet --vpc-id $vpc_id --availability-zone
 $az_2 --cidr-block 192.168.1.32/27 \
    --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=my-eks-
custom-networking-vpc-PrivateSubnet02},{Key=kubernetes.io/role/internal-
elb,Value=1}]' \
    --query Subnet.SubnetId --output text)
```

> **Important**
>
> Standardmäßig sind Ihre neuen Subnetze implizit mit der Haupt-Routing-Tabelle Ihrer VPCs verknüpft. Diese Routing-Tabelle ermöglicht die Kommunikation zwischen allen Ressourcen, die in der VPC bereitgestellt werden. Sie erlaubt jedoch keine

Kommunikation mit Ressourcen mit IP-Adressen, die sich außerhalb der mit Ihrer VPC verknüpften CIDR-Blöcke befinden. Sie können Ihre eigene Routing-Tabelle Ihren Subnetzen zuordnen, um dieses Verhalten zu ändern. Weitere Informationen finden Sie unter Subnetz-Routing-Tabellen im Amazon-VPC-Benutzerhandbuch.

2. Zeigen Sie die aktuellen Subnetze in Ihrer VPC an.

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=$vpc_id" \
    --query 'Subnets[*].{SubnetId: SubnetId,AvailabilityZone:
 AvailabilityZone,CidrBlock: CidrBlock}' \
    --output table
```

Eine Beispielausgabe sieht wie folgt aus.

```
-------------------------------------------------------------------
|                          DescribeSubnets                        |
+------------------+--------------------+---------------------------+
| AvailabilityZone |     CidrBlock      |         SubnetId          |
+------------------+--------------------+---------------------------+
|  us-west-2d      |  192.168.0.0/27    |     subnet-example1       |
|  us-west-2a      |  192.168.0.32/27   |     subnet-example2       |
|  us-west-2a      |  192.168.0.64/27   |     subnet-example3       |
|  us-west-2d      |  192.168.0.96/27   |     subnet-example4       |
|  us-west-2a      |  192.168.1.0/27    |     subnet-example5       |
|  us-west-2d      |  192.168.1.32/27   |     subnet-example6       |
+------------------+--------------------+---------------------------+
```

Sie können sehen, dass die Subnetze im CIDR-Block `192.168.1.0`, den Sie erstellt haben, in denselben Availability Zones liegen wie die Subnetze im CIDR-Block `192.168.0.0`.

Schritt 3: Konfigurieren von Kubernetes-Ressourcen

Konfigurieren Sie die Kubernetes-Ressourcen wie folgt

1. Setzen Sie die `AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG`-Umgebungsvariable im `aws-node` `DaemonSet` auf `true`.

```
kubectl set env daemonset aws-node -n kube-system
 AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG=true
```

2. Rufen Sie die ID der Cluster-Sicherheitsgruppe ab und speichern Sie diese in einer Variablen zur Verwendung im nächsten Schritt. Amazon EKS erstellt diese Sicherheitsgruppe automatisch, wenn Sie Ihren Cluster erstellen.

```
cluster_security_group_id=$(aws eks describe-cluster --name $cluster_name --query
 cluster.resourcesVpcConfig.clusterSecurityGroupId --output text)
```

3. Erstellen Sie eine benutzerdefinierte `ENIConfig`-Ressource für jedes Subnetz, in dem Sie Pods bereitstellen möchten.

   a. Erstellen Sie für jede Konfiguration der Netzwerkschnittstelle eine eindeutige Datei.

   Die folgenden Befehle erstellen separate `ENIConfig`-Dateien für die beiden Subnetze, die in einem vorherigen Schritt erstellt wurden. Der Wert für `name` muss eindeutig sein. Der Name entspricht der Availability Zone, in der sich das Subnetz befindet. Die Cluster-Sicherheitsgruppe ist dem `ENIConfig` zugeordnet.

```
cat >$az_1.yaml <<EOF
apiVersion: crd.k8s.amazonaws.com/v1alpha1
kind: ENIConfig
metadata:
  name: $az_1
spec:
  securityGroups:
    - $cluster_security_group_id
  subnet: $new_subnet_id_1
EOF
```

```
cat >$az_2.yaml <<EOF
apiVersion: crd.k8s.amazonaws.com/v1alpha1
kind: ENIConfig
metadata:
  name: $az_2
spec:
  securityGroups:
    - $cluster_security_group_id
  subnet: $new_subnet_id_2
EOF
```

Für einen Produktions-Cluster können Sie die folgenden Änderungen an den vorherigen Befehlen vornehmen:

- Ersetzen Sie *$cluster_security_group_id* mit der ID einer bestehenden Sicherheitsgruppe, die Sie für jeden `ENIConfig` verwenden möchten.
- Wir empfehlen, Ihrer `ENIConfigs` denselben Namen zu geben wie der Availability Zone, in der Sie die `ENIConfig` verwenden, falls möglich. Unter Umständen müssen Sie für Ihre `ENIConfigs` andere Namen verwenden als für die Availability Zones. Wenn Sie beispielsweise mehr als zwei Subnetze in derselben Availability Zone haben und beide mit benutzerdefinierten Netzwerken verwenden möchten, benötigen Sie mehrere `ENIConfigs` für dieselbe Availability Zone. Da jede `ENIConfig` einen eindeutigen Namen erfordert, können Sie nur einer Ihrer `ENIConfigs` den Namen der Availability Zone geben.

  Wenn Ihre `ENIConfig`-Namen nicht alle identisch mit den Availability-Zone-Namen sind, ersetzen Sie *$az_1* und *$az_2* in den vorherigen Befehlen durch eigene Namen und kommentieren Sie Ihre Knoten mit dem `ENIConfig` weiter unten in diesem Tutorial.

> **Note**
>
> Wenn Sie keine gültige Sicherheitsgruppe für die Verwendung mit einem Produktions-Cluster angeben und
>
> - Version `1.8.0` oder höher des Amazon VPC CNI plugin for Kubernetes verwenden, dann werden die Sicherheitsgruppen verwendet, die mit der primären Elastic-Network-Schnittstelle des Knotens verknüpft sind.
> - eine Version des Amazon VPC CNI plugin for Kubernetes vor `1.8.0` verwenden, wird die Standardsicherheitsgruppe für die VPC sekundären Netzwerkschnittstellen zugewiesen.

> **Important**
>
> - `AWS_VPC_K8S_CNI_EXTERNALSNAT=false` ist eine Standardeinstellung in der Konfiguration für das Amazon-VPC-CNI-Plugin für Kubernetes. Wenn Sie die Standardeinstellung verwenden, verwendet Datenverkehr, der

für IP-Adressen bestimmt ist, die sich nicht innerhalb eines mit Ihrer VPC verknüpften CIDR-Blöcke befinden, die Sicherheitsgruppen und Subnetze der primären Netzwerkschnittstelle Ihres Knotens. Die in Ihrem `ENIConfigs` definierten Subnetze und Sicherheitsgruppen, die zum Erstellen sekundärer Netzwerkschnittstellen verwendet werden, werden für diesen Datenverkehr nicht verwendet. Weitere Informationen zu dieser Einstellung finden Sie unter SNAT für Pods.

- Wenn Sie auch Sicherheitsgruppen für Pods verwenden, wird die in einer `SecurityGroupPolicy` angegebene Sicherheitsgruppe anstelle der Sicherheitsgruppe verwendet, die im `ENIConfigs` angegeben ist. Weitere Informationen finden Sie unter Sicherheitsgruppen für Pods.

b. Wenden Sie alle benutzerdefinierten Ressourcendateien, die Sie zuvor erstellt haben, mit den folgenden Befehlen auf Ihren Cluster an:

```
kubectl apply -f $az_1.yaml
kubectl apply -f $az_2.yaml
```

4. Bestätigen Sie, dass Ihre `ENIConfigs` erstellt wurden.

```
kubectl get ENIConfigs
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME         AGE
us-west-2a   117s
us-west-2d   105s
```

5. Wenn Sie benutzerdefinierte Netzwerke in einem Produktions-Cluster aktivieren und Ihren `ENIConfigs` einen anderen Namen gegeben haben als die Availability Zone, für die Sie diese verwenden, springen Sie zum nächsten Schritt, um Amazon-EC2-Knoten bereitzustellen.

   Aktivieren Sie Kubernetes, um die `ENIConfig` für eine Availability Zone für alle neuen Amazon-EC2-Knoten zu verwenden, die in Ihrem Cluster erstellt wurden.

   1. Gehen Sie für den Test-Cluster in diesem Tutorial zum nächsten Schritt.

      Prüfen Sie für einen Produktions-Cluster, ob eine annotation mit dem Schlüssel `k8s.amazonaws.com/eniConfig` für die `ENI_CONFIG_ANNOTATION_DEF`

Umgebungsvariable in der Container-Spezifikation für den `aws-node DaemonSet` vorhanden ist.

```
kubectl describe daemonset aws-node -n kube-system | grep
 ENI_CONFIG_ANNOTATION_DEF
```

Wenn eine Ausgabe zurückgegeben wird, ist die Anmerkung vorhanden. Wenn keine Ausgabe zurückgegeben wird, ist die Variable nicht gesetzt. Für einen Produktions-Cluster können Sie entweder diese Einstellung oder die Einstellung im folgenden Schritt verwenden. Wenn Sie diese Einstellung verwenden, überschreibt sie die Einstellung im folgenden Schritt. In diesem Tutorial wird die Einstellung im nächsten Schritt verwendet.

2. Aktualisieren Sie Ihren `aws-node DaemonSet`, um die `ENIConfig` für eine Availability Zone automatisch für alle neuen Amazon-EC2-Knoten zu verwenden, die in Ihrem Cluster erstellt wurden.

```
kubectl set env daemonset aws-node -n kube-system
 ENI_CONFIG_LABEL_DEF=topology.kubernetes.io/zone
```

Schritt 4: Bereitstellen von Amazon-EC2-Knoten

So stellen Sie Amazon-EC2-Knoten bereit

1. Erstellen Sie eine Knoten-IAM-Rolle.

   a. Führen Sie den folgenden Befehl aus, um eine JSON-Datei für eine IAM-Vertrauensrichtlinie zu erstellen.

```
cat >node-role-trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
EOF
```

b. Führen Sie den folgenden Befehl aus, um eine Variable für Ihren Rollennamen festzulegen. Sie können *myCustomNetworkingAmazonEKSNodeRole* mit einem beliebigen Namen ersetzen, den Sie wählen.

```
export node_role_name=myCustomNetworkingAmazonEKSNodeRole
```

c. Erstellen Sie die IAM-Rolle und speichern Sie den zurückgegebenen Amazon-Ressourcennamen (ARN) in einer Variablen zur Verwendung in einem späteren Schritt.

```
node_role_arn=$(aws iam create-role --role-name $node_role_name --assume-role-
policy-document file://"node-role-trust-relationship.json" \
    --query Role.Arn --output text)
```

d. Hängen Sie die drei erforderlichen verwalteten IAM-Richtlinien an die IAM-Rolle an.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name $node_role_name
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name $node_role_name
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
    --role-name $node_role_name
```

> **Important**
>
> Der Einfachheit halber wird in diesem Tutorial die Richtlinie AmazonEKS_CNI_Policy an diese Knoten-IAM-Rolle angehängt. In einem Produktions-Cluster empfehlen wir jedoch, die Richtlinie an eine separate IAM-Rolle anzuhängen, die nur mit dem Amazon VPC CNI plugin for Kubernetes verwendet wird. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

2. Erstellen Sie einen der folgenden Typen von Knotengruppen. Informationen zum Bestimmen des Instance-Typs, den Sie bereitstellen möchten, finden Sie unter Auswählen eines Amazon-EC2-Instance-Typs. Füllen Sie für dieses Tutorial die Option Verwaltet, Ohne Startvorlage oder

mit Startvorlage ohne angegebene AMI-ID aus. Wenn Sie die Knotengruppe für Produktions-Workloads verwenden, empfehlen wir, dass Sie sich vor Bereitstellen der Knotengruppe mit allen verwalteten und selbstverwalteten Knotengruppenoptionen vertraut machen.

- Verwaltet – Stellen Sie Ihre Knotengruppe mit einer der folgenden Optionen bereit:
  - Ohne Startvorlage oder mit Startvorlage ohne angegebene AMI-ID – Führen Sie den folgenden Befehl aus. Verwenden Sie in diesem Tutorial die *example values*. Ersetzen Sie für eine Produktionsknotengruppe alle *example values* durch Ihre eigenen Werte. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten.

    ```
    aws eks create-nodegroup --cluster-name $cluster_name --nodegroup-name my-nodegroup \
        --subnets $subnet_id_1 $subnet_id_2 --instance-types t3.medium --node-role $node_role_arn
    ```

  - Mit einer Startvorlage mit einer angegebenen AMI-ID
    1. Bestimmen Sie die von Amazon EKS empfohlenen maximalen Pods für Ihre Knoten. Befolgen Sie die Anweisungen in Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ und fügen Sie **--cni-custom-networking-enabled** zu Schritt 3 in diesem Thema hinzu. Notieren Sie die Ausgabe zur Verwendung im nächsten Schritt.
    2. Geben Sie in Ihrer Startvorlage eine Amazon-EKS-optimierte AMI-ID oder ein benutzerdefiniertes AMI an, das auf dem Amazon-EKS-optimierten AMI basiert. Stellen Sie dann die Knotengruppe mithilfe einer Startvorlage bereit und geben Sie die folgenden Benutzerdaten in der Startvorlage an. Diese Benutzerdaten übergeben Argumente an die `bootstrap.sh`-Datei. Weitere Informationen zur Bootstrap-Datei finden Sie unter bootstrap.sh auf GitHub. Sie können *20* entweder durch den Wert aus dem vorherigen Schritt (empfohlen) oder Ihren eigenen Wert ersetzen.

       ```
       /etc/eks/bootstrap.sh my-cluster --use-max-pods false --kubelet-extra-args '--max-pods=20'
       ```

       Wenn Sie ein benutzerdefiniertes AMI erstellt haben, das nicht auf dem für Amazon EKS optimierten AMI erstellt wurde, müssen Sie die Konfiguration selbst erstellen.
- Selbstverwaltet

1. Bestimmen Sie die von Amazon EKS empfohlenen maximalen Pods für Ihre Knoten. Befolgen Sie die Anweisungen in Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ und fügen Sie **`--cni-custom-networking-enabled`** zu Schritt 3 in diesem Thema hinzu. Notieren Sie die Ausgabe zur Verwendung im nächsten Schritt.
2. Stellen Sie die Knotengruppe mithilfe der Anweisungen in Starten selbstverwalteter Amazon Linux-Knoten bereit. Geben Sie den folgenden Text für den BootstrapArgumentsParameter an. Sie können *`20`* entweder durch den Wert aus dem vorherigen Schritt (empfohlen) oder Ihren eigenen Wert ersetzen.

   ```
   --use-max-pods false --kubelet-extra-args '--max-pods=20'
   ```

> **Note**
>
> Wenn Sie möchten, dass Knoten in einem Produktions-Cluster eine deutlich höhere Anzahl von Pods unterstützen, führen Sie das Skript in Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ erneut aus. Fügen Sie außerdem die Option **`--cni-prefix-delegation-enabled`** zu dem Befehl hinzu. Beispielsweise wird *`110`* für einen `m5.large`-Instance-Typ zurückgegeben. Anweisungen, wie Sie diese Funktion aktivieren, finden Sie unter Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten. Sie können diese Funktion mit benutzerdefinierten Netzwerken verwenden.

Die Erstellung der Knotengruppe dauert mehrere Minuten. Sie können den Status der Erstellung einer verwalteten Knotengruppe mit dem folgenden Befehl überprüfen.

```
aws eks describe-nodegroup --cluster-name $cluster_name --nodegroup-name my-nodegroup --query nodegroup.status --output text
```

Fahren Sie erst mit dem nächsten Schritt fort, wenn die zurückgegebene Ausgabe `ACTIVE` ist.

3. Für das Tutorial können Sie diesen Schritt überspringen.

   Wenn Sie Ihre `ENIConfigs` für einen Produktions-Cluster nicht genauso benannt haben wie die Availability Zone, für die Sie diese verwenden, müssen Sie in Ihren Knoten den Namen der `ENIConfig` anmerken, die mit dem Knoten verwendet werden soll. Dieser Schritt ist nicht

erforderlich, wenn Sie in jeder Availability Zone nur ein Subnetz haben und Ihre `ENIConfigs` denselben Namen haben wie Ihre Availability Zones. Das liegt daran, dass das Amazon VPC CNI plugin for Kubernetes automatisch die richtige `ENIConfig` mit dem Knoten verknüpft, wenn Sie dies in einem vorherigen Schritt aktiviert haben.

a. Rufen Sie die Liste der Knoten in Ihrem Cluster ab.

```
kubectl get nodes
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                                           STATUS   ROLES    AGE     VERSION
ip-192-168-0-126.us-west-2.compute.internal   Ready    <none>   8m49s
 v1.22.9-eks-810597c
ip-192-168-0-92.us-west-2.compute.internal    Ready    <none>   8m34s
 v1.22.9-eks-810597c
```

b. Bestimmen Sie, in welcher Availability Zone sich jeder Knoten befindet. Führen Sie den folgenden Befehl für jeden Knoten aus, der im vorherigen Schritt ausgegeben wurde.

```
aws ec2 describe-instances --filters Name=network-interface.private-dns-
name,Values=ip-192-168-0-126.us-west-2.compute.internal \
--query 'Reservations[].Instances[].{AvailabilityZone:
 Placement.AvailabilityZone, SubnetId: SubnetId}'
```

Eine Beispielausgabe sieht wie folgt aus.

```
[
    {
        "AvailabilityZone": "us-west-2d",
        "SubnetId": "subnet-Example5"
    }
]
```

c. Kommentieren Sie jeden Knoten mit der `ENIConfig`, die Sie für die Subnetz-ID und die Availability Zone erstellt haben. Sie können einen Knoten nur mit einer `ENIConfig` kommentieren, aber es können mehrere Knoten mit derselben `ENIConfig` kommentiert werden. Ersetzen Sie das *`example values`* durch Ihr eigenes.

```
kubectl annotate node ip-192-168-0-126.us-west-2.compute.internal
 k8s.amazonaws.com/eniConfig=EniConfigName1
kubectl annotate node ip-192-168-0-92.us-west-2.compute.internal
 k8s.amazonaws.com/eniConfig=EniConfigName2
```

4. Wenn Sie Knoten in einem Produktions-Cluster hatten, in dem Pods ausgeführt wurden, bevor Sie zur Verwendung des benutzerdefinierten Netzwerkfeatures gewechselt haben, führen Sie die folgenden Schritte aus:

   a. Stellen Sie sicher, dass es verfügbare Knoten gibt, das benutzerdefinierte Netzwerkfeature verwenden.

   b. Sperren und entleeren Sie die Knoten, um die Pods herunterzufahren. Weitere Informationen finden Sie unter Sicheres Entleeren eines Knotens in der Kubernetes-Dokumentation.

   c. Beenden Sie die Knoten. Wenn sich die Knoten in einer vorhandenen verwalteten Knotengruppe befinden, können Sie die Knotengruppe löschen. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie nach Bedarf die folgenden Änderungen am Befehl vor und führen Sie anschließend den geänderten Befehl aus:

      - Ersetzen Sie *my-cluster* durch den Namen für Ihren Cluster.
      - Ersetzen Sie *my-nodegroup* durch den Namen für Ihre Knotengruppe.

      ```
      aws eks delete-nodegroup --cluster-name my-cluster --nodegroup-name my-
      nodegroup
      ```

   Nur neue Knoten, die mit der Bezeichnung `k8s.amazonaws.com/eniConfig` registriert sind, verwenden das neue benutzerdefinierte Netzwerkfeature.

5. Bestätigen Sie, dass Pods eine IP-Adresse aus einem CIDR-Block zugewiesen wird, der mit einem der in einem vorherigen Schritt erstellten Subnetze verknüpft ist.

   ```
   kubectl get pods -A -o wide
   ```

   Eine Beispielausgabe sieht wie folgt aus.

```
NAMESPACE     NAME                          READY   STATUS    RESTARTS   AGE     IP
           NODE                                                NOMINATED NODE   READINESS
 GATES
kube-system   aws-node-2rkn4                1/1     Running   0          7m19s
 192.168.0.92    ip-192-168-0-92.us-west-2.compute.internal    <none>
 <none>
kube-system   aws-node-k96wp                1/1     Running   0          7m15s
 192.168.0.126   ip-192-168-0-126.us-west-2.compute.internal   <none>
 <none>
kube-system   coredns-657694c6f4-smcgr      1/1     Running   0          56m
 192.168.1.23    ip-192-168-0-92.us-west-2.compute.internal    <none>
 <none>
kube-system   coredns-657694c6f4-stwv9      1/1     Running   0          56m
 192.168.1.28    ip-192-168-0-92.us-west-2.compute.internal    <none>
 <none>
kube-system   kube-proxy-jgshq              1/1     Running   0          7m19s
 192.168.0.92    ip-192-168-0-92.us-west-2.compute.internal    <none>
 <none>
kube-system   kube-proxy-wx9vk              1/1     Running   0          7m15s
 192.168.0.126   ip-192-168-0-126.us-west-2.compute.internal   <none>
 <none>
```

Sie sehen, dass den `coredns` Pods IP-Adressen aus dem CIDR-Block `192.168.1.0` zugewiesen werden, den Sie Ihrer VPC hinzugefügt haben. Ohne benutzerdefiniertes Netzwerk wären diesen Adressen aus dem CIDR-Block `192.168.0.0` zugewiesen worden, da dies der einzige ursprünglich mit der VPC verbundene CIDR-Block war.

Wenn die `spec` eines Pod's `hostNetwork=true` enthalten, wird diesem die primäre IP-Adresse des Knotens zugewiesen. Es wird keine Adresse aus den hinzugefügten Subnetzen zugewiesen. Standardmäßig ist dieser Wert auf `false` festgelegt. Dieser Wert ist auf `true` festgelegt für `kube-proxy` und Amazon VPC CNI plugin for Kubernetes (`aws-node`) Pods, die auf Ihrem Cluster laufen. Aus diesem Grund sind der `kube-proxy` und der `aws-node`Pods des Plugins in der vorherigen Ausgabe nicht `192.168.1.`*x*-Adressen zugewiesen. Weitere Informationen zu einer Pod's `hostNetwork` Einstellung finden Sie unter PodSpec v1 core in der Kubernetes API-Referenz.

Schritt 5: Löschen von Tutorial-Ressourcen

Nachdem Sie das Tutorial abgeschlossen haben, empfehlen wir, dass Sie die erstellten Ressourcen löschen. Sie können dann die Schritte anpassen, um benutzerdefinierte Netzwerke für einen Produktions-Cluster zu aktivieren.

Löschen Sie die Tutorial-Ressourcen wie folgt

1. Wenn die von Ihnen erstellte Knotengruppe nur zum Testen gedacht ist, löschen Sie sie.

   ```
   aws eks delete-nodegroup --cluster-name $cluster_name --nodegroup-name my-nodegroup
   ```

   Selbst wenn in der AWS CLI Ausgabe angegeben wird, dass der Cluster gelöscht wurde, ist der Löschvorgang möglicherweise noch nicht abgeschlossen. Der Löschvorgang dauert einige Minuten. Bestätigen Sie, dass der Vorgang abgeschlossen ist, indem Sie den folgenden Befehl ausführen.

   ```
   aws eks describe-nodegroup --cluster-name $cluster_name --nodegroup-name my-
   nodegroup --query nodegroup.status --output text
   ```

   Fahren Sie erst fort, wenn die zurückgegebene Ausgabe der folgenden Ausgabe entspricht.

   ```
   An error occurred (ResourceNotFoundException) when calling the DescribeNodegroup
    operation: No node group found for name: my-nodegroup.
   ```

2. Wenn die von Ihnen erstellte Knotengruppe nur zum Testen gedacht ist, löschen Sie den Knoten IAM-Rolle.

   a. Trennen Sie die Richtlinien von der Rolle.

      ```
      aws iam detach-role-policy --role-name myCustomNetworkingAmazonEKSNodeRole --
      policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
      aws iam detach-role-policy --role-name myCustomNetworkingAmazonEKSNodeRole --
      policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
      aws iam detach-role-policy --role-name myCustomNetworkingAmazonEKSNodeRole --
      policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
      ```

   b. Löschen Sie die Rolle.

      ```
      aws iam delete-role --role-name myCustomNetworkingAmazonEKSNodeRole
      ```

3. Löschen Sie den Cluster.

```
aws eks delete-cluster --name $cluster_name
```

Bestätigen Sie, dass der Cluster gelöscht ist, indem Sie den folgenden Befehl ausführen.

```
aws eks describe-cluster --name $cluster_name --query cluster.status --output text
```

Wird eine Ausgabe ähnlich der folgenden zurückgegeben, wurde der Cluster erfolgreich gelöscht.

```
An error occurred (ResourceNotFoundException) when calling the DescribeCluster
 operation: No cluster found for name: my-cluster.
```

4. Löschen Sie die Cluster-IAM-Rolle.

   a. Trennen Sie die Richtlinien von der Rolle.

   ```
   aws iam detach-role-policy --role-name myCustomNetworkingAmazonEKSClusterRole
    --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
   ```

   b. Löschen Sie die Rolle.

   ```
   aws iam delete-role --role-name myCustomNetworkingAmazonEKSClusterRole
   ```

5. Löschen Sie die Subnetze, die Sie in einem vorherigen Schritt erstellt haben.

```
aws ec2 delete-subnet --subnet-id $new_subnet_id_1
aws ec2 delete-subnet --subnet-id $new_subnet_id_2
```

6. Löschen Sie die VPC, die Sie erstellt haben.

```
aws cloudformation delete-stack --stack-name my-eks-custom-networking-vpc
```

Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten

Jede Amazon-EC2-Instance unterstützt eine maximale Anzahl von elastischen Netzwerkschnittstellen und eine maximale Anzahl von IP-Adressen, die jeder Netzwerkschnittstelle zugewiesen werden können. Jeder Knoten benötigt eine IP-Adresse für jede Netzwerkschnittstelle. Alle anderen

verfügbaren IP-Adressen können Pods zugewiesen werden. Jeder Pod benötigt seine eigene IP-Adresse. Daher verfügen Sie möglicherweise über Knoten, die über verfügbare Datenverarbeitungs- und Speicherressourcen verfügen, aber keine weiteren Pods aufnehmen können, da der Knoten nicht mehr über IP-Adressen verfügt, die Pods zugewiesen werden können.

In diesem Thema erfahren Sie, wie Sie die Anzahl der IP-Adressen, die Knoten Pods zuweisen können, deutlich erhöhen können, indem Sie IP-Präfixe zuweisen, anstatt Ihren Knoten einzelne sekundäre IP-Adressen zuzuweisen. Jedes Präfix enthält mehrere IP-Adressen. Wenn Sie Ihren Cluster nicht für die IP-Präfixzuweisung konfigurieren, muss Ihr Cluster mehr Amazon-EC2-API-Aufrufe (Application Programming Interface) ausführen, um Netzwerkschnittstellen und IP-Adressen zu konfigurieren, die für die Pod-Konnektivität erforderlich sind. Wenn die Cluster größer werden, kann die Häufigkeit dieser API-Aufrufe zu längeren Startzeiten der Pod und Instances führen. Dies führt zu Skalierungsverzögerungen, um die Nachfrage nach großen und spitzen Workloads zu decken, und erhöht den Kosten- und Verwaltungsaufwand, da Sie zusätzliche Cluster und VPCs bereitstellen müssen, um die Skalierungsanforderungen zu erfüllen. Weitere Informationen finden Sie unter Schwellenwerte für Kubernetes Skalierbarkeit. GitHub

Überlegungen

- Jeder Amazon-EC2-Instance-Typ unterstützt eine maximale Anzahl von Pods. Wenn Ihre verwaltete Knotengruppe aus mehreren Instance-Typen besteht, wird die kleinste maximale Anzahl von Pods für eine Instance im Cluster auf alle Knoten im Cluster angewendet.
- Standardmäßig ist die maximale Anzahl von Pods, die Sie auf einem Knoten ausführen können, 110, aber Sie können diese Anzahl ändern. Wenn Sie die Anzahl ändern und eine verwaltete Knotengruppe haben, führt die nächste AMI- oder Startvorlagenaktualisierung Ihrer Knotengruppe dazu, dass neue Knoten mit dem geänderten Wert erstellt werden.
- Beim Übergang von der Zuweisung von IP-Adressen zur Zuweisung von IP-Präfixen empfehlen wir, dass Sie neue Knotengruppen erstellen, um die Anzahl der verfügbaren IP-Adressen zu erhöhen, anstatt die vorhandenen Knoten fortlaufend zu ersetzen. Die Ausführung von Pods auf einem Knoten, dem sowohl IP-Adressen als auch Präfixe zugewiesen sind, kann zu Inkonsistenzen bei der angekündigten IP-Adresskapazität führen, was sich auf die zukünftigen Workloads auf dem Knoten auswirkt. Die empfohlene Methode zur Durchführung des Übergangs finden Sie unter Ersetzen aller Knoten während der Migration vom sekundären IP-Modus in den Präfix-Delegierungsmodus oder umgekehrt im Amazon-EKS-Leitfaden mit bewährten Methoden.
- Nur für Cluster mit Linux-Knoten.
  - Nachdem Sie das Add-On so konfiguriert haben, dass Netzwerkschnittstellen Präfixe zugewiesen werden, können Sie Ihr Amazon VPC CNI plugin for Kubernetes-Add-On nicht

auf eine niedrigere Version als `1.9.0` (oder`1.10.1`) herabstufen, ohne alle Knoten in allen Knotengruppen in Ihrem Cluster zu entfernen.

- Wenn Sie auch Sicherheitsgruppen für Pods verwenden, mit `POD_SECURITY_GROUP_ENFORCING_MODE` = `standard` und `AWS_VPC_K8S_CNI_EXTERNALSNAT` werden bei der Kommunikation Ihrer Pods mit Endpunkten außerhalb Ihrer VPC die Sicherheitsgruppen des Knotens verwendet und nicht die Sicherheitsgruppen, die Sie Ihren Pods zugewiesen haben.

  Wenn Sie Sicherheitsgruppen auch für Pods verwenden, mit `POD_SECURITY_GROUP_ENFORCING_MODE` = `strict`, werden die `Pod's`- Sicherheitsgruppen verwendet, wenn Ihre`Pods` mit Endpunkten außerhalb Ihrer VPC kommunizieren.

Voraussetzungen

- Einen vorhandenen -Cluster. Informationen zum Bereitstellen finden Sie unter Erstellen eines Amazon-EKS-Clusters.
- Die Subnetze, in denen sich Ihre Amazon-EKS-Knoten befinden, müssen über ausreichend zusammenhängende `/28` (für `IPv4`-Cluster) oder `/80` (für `IPv6`-Cluster) CIDR-Blöcke (Classless Inter-Domain Routing) verfügen. Sie können nur Linux-Knoten in einem `IPv6`-Cluster haben. Die Verwendung von IP-Präfixen kann fehlschlagen, wenn IP-Adressen im Subnetz-CIDR verstreut sind. Wir empfehlen Folgendes:
  - Verwendung einer Subnetz-CIDR-Reservierung, sodass selbst dann, wenn IP-Adressen innerhalb des reservierten Bereichs noch verwendet werden, die IP-Adressen nach ihrer Veröffentlichung nicht neu zugewiesen werden. Dadurch wird sichergestellt, dass Präfixe für die Zuweisung ohne Segmentierung verfügbar sind.
  - Verwenden Sie neue Subnetze, die speziell für die Ausführung der Workloads verwendet werden, denen IP-Präfixe zugewiesen sind. Sowohl Windows- als auch Linux-Workloads können bei der Zuweisung von IP-Präfixen im selben Subnetz ausgeführt werden.
- Um Ihren Knoten IP-Präfixe zuzuweisen, müssen Ihre Knoten Nitro-basiert sein AWS . Instanzen, die nicht auf Nitro basieren, weisen weiterhin individuelle sekundäre IP-Adressen zu, haben aber eine deutlich geringere Anzahl von IP-Adressen, die Pods zugewiesen werden können, als Nitro-based-Instances.
- Nur für Cluster mit Linux Knoten – Wenn Ihr Cluster für die `IPv4`-Familie konfiguriert ist, müssen Sie die Version `1.9.0` oder eine höhere Version des Amazon VPC CNI plugin for Kubernetes-Add-ons installiert haben. Sie können Ihre aktuelle Version mit dem folgenden Befehl überprüfen.

```
kubectl describe daemonset aws-node --namespace kube-system | grep Image | cut -d "/"
 -f 2
```

Wenn Ihr Cluster für die `IPv6`-Familie konfiguriert ist, müssen Sie die Version `1.10.1` des Add-ons installiert haben. Wenn Ihre Plugin-Version älter ist als die erforderlichen Versionen, müssen Sie sie aktualisieren. Weitere Informationen finden Sie in den Aktualisierungsabschnitten von Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

- Nur für Cluster mit Windows-Knoten
  - Ihr Cluster und seine Plattformversion müssen mindestens den Versionen in der folgenden Tabelle entsprechen. Lesen Sie Aktualisieren einer Amazon-EKS-Cluster-Kubernetes-Version, um Ihre Cluster-Version zu aktualisieren. Wenn Ihr Cluster nicht die Mindestplattformversion hat, können Sie Ihren Knoten keine IP-Präfixe zuweisen, bis Amazon EKS Ihre Plattformversion aktualisiert hat.

Kubernetes-Version	Plattformversion
`1.27`	`eks.3`
`1.26`	`eks.4`
`1.25`	`eks.5`

    Sie können Ihre aktuelle Version Kubernetes und die Plattformversion überprüfen, indem Sie *`my-cluster`* im folgenden Befehl durch den Namen Ihres Clusters ersetzen und dann den geänderten Befehl ausführen: **`aws eks describe-cluster --name my-cluster --query 'cluster.{"Kubernetes Version": version, "Platform Version": platformVersion}'`**.
  - Windows-Unterstützung für Ihren Cluster aktiviert. Weitere Informationen finden Sie unter Die Windows-Unterstützung für Ihre Amazon-EKS-Cluster aktivieren.

So erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten

1. Konfigurieren Sie Ihren Cluster, um Knoten IP-Adresspräfixe zuzuweisen. Führen Sie das Verfahren auf der Registerkarte aus, die dem Betriebssystem Ihres Knotens entspricht.

Linux

1. Aktivieren Sie den Parameter, um den Netzwerkschnittstellen für das Amazon VPC CNI `DaemonSet` Präfixe zuzuweisen. Wenn Sie einen Cluster ab `1.21` bereitstellen, wird Version `1.10.1` oder höher des Amazon VPC CNI plugin for Kubernetes-Add-ons zusammen mit ihm bereitgestellt. Wenn Sie den Cluster mit der `IPv6`-Familie erstellt haben, wurde für diese Einstellung standardmäßig `true` festgelegt. Wenn Sie den Cluster mit der `IPv4`-Familie erstellt haben, wurde für diese Einstellung standardmäßig `false` festgelegt.

```
kubectl set env daemonset aws-node -n kube-system
 ENABLE_PREFIX_DELEGATION=true
```

> **Important**
>
> Auch wenn Ihr Subnetz über verfügbare IP-Adressen verfügt, wird in den Amazon VPC CNI plugin for Kubernetes-Protokollen der folgende Fehler angezeigt, wenn im Subnetz keine zusammenhängenden /28-Blöcke verfügbar sind:
>
> ```
> InsufficientCidrBlocks: The specified subnet does not have enough free
>  cidr blocks to satisfy the request
> ```
>
> Dies kann aufgrund der Fragmentierung vorhandener sekundärer IP-Adressen auftreten, die über ein Subnetz verteilt sind. Um diesen Fehler zu beheben, erstellen Sie entweder ein neues Subnetz und starten Sie dort Pods, oder verwenden Sie eine Amazon-EC2-Subnetz-CIDR-Reservierung, um Speicherplatz in einem Subnetz für die Verwendung mit Präfixzuweisung zu reservieren. Weitere Informationen erhalten Sie unter Subnetz-CIDR-Reservierungen im Amazon-VPC-Benutzerhandbuch.

2. Wenn Sie planen, eine verwaltete Knotengruppe ohne Startvorlage bereitzustellen oder mit einer Startvorlage, in der Sie keine AMI-ID angegeben haben, und Sie eine Version des Amazon VPC CNI plugin for Kubernetes verwenden, die mindestens der in den Voraussetzungen gelisteten Version entspricht, fahren Sie mit dem nächsten Schritt fort. Verwaltete Knotengruppen berechnen automatisch die maximale Anzahl von Pods für Sie.

Wenn Sie eine selbstverwaltete Knotengruppe oder eine verwaltete Knotengruppe mit einer Startvorlage bereitstellen, in der Sie eine AMI-ID angegeben haben, müssen Sie die von Amazon EKS empfohlene Anzahl der maximalen Pods für Ihre Knoten ermitteln. Befolgen Sie die Anweisungen in Von Amazon EKS empfohlene maximale Pods für jeden Amazon-EC2-Instance-Typ und fügen Sie **`--cni-prefix-delegation-enabled`** zu Schritt 3 hinzu. Notieren Sie sich die Ausgabe zur Verwendung in einem späteren Schritt.

> **Important**
>
> Verwaltete Knotengruppen erzwingen eine maximale Anzahl für den Wert von `maxPods`. Für Instances mit weniger als 30 vCPUs beträgt die Höchstzahl 110 und für alle anderen Instances beträgt die Höchstzahl 250. Diese maximale Anzahl wird angewendet, unabhängig davon, ob die Präfixdelegierung aktiviert ist oder nicht.

3. Wenn Sie einen Cluster `1.21` oder spätere Version verwenden, der für `IPv6` konfiguriert ist, fahren Sie mit dem nächsten Schritt fort.

   Geben Sie die Parameter in einer der folgenden Optionen an. Um zu bestimmen, welche Option für Sie die richtige ist und welchen Wert Sie dafür bereitstellen müssen, lesen Sie `WARM_PREFIX_TARGET`, `WARM_IP_TARGET` und `MINIMUM_IP_TARGET` auf GitHub.

   Sie können die *`example values`* durch einen Wert größer als Null ersetzen.

   - `WARM_PREFIX_TARGET`

     ```
     kubectl set env ds aws-node -n kube-system WARM_PREFIX_TARGET=1
     ```

   - `WARM_IP_TARGET` oder `MINIMUM_IP_TARGET` – Wenn einer der Werte festgelegt ist, überschreibt er alle Werte, die für `WARM_PREFIX_TARGET` festgelegt wurden.

     ```
     kubectl set env ds aws-node -n kube-system WARM_IP_TARGET=5
     ```

     ```
     kubectl set env ds aws-node -n kube-system MINIMUM_IP_TARGET=2
     ```

4. Erstellen Sie einen der folgenden Typen von Knotengruppen mit mindestens einem Amazon EC2 Nitro Amazon-Linux-2-Instance-Typ. Eine Liste der Nitro-Instance-Typen finden Sie unter Instances built on the Nitro System im Amazon EC2 EC2-Benutzerhandbuch. Diese Funktion wird unter Windows nicht unterstützt. Für die Optionen,

die *110* enthalten, ersetzen Sie es entweder durch den Wert aus Schritt 3 (empfohlen) oder Ihren eigenen Wert.

- Selbstverwaltet – Stellen Sie die Knotengruppe mithilfe der Anweisungen in Starten selbstverwalteter Amazon Linux-Knoten bereit. Geben Sie den folgenden Text für den Parameter an. BootstrapArguments

  ```
  --use-max-pods false --kubelet-extra-args '--max-pods=110'
  ```

  Wenn Sie `eksctl` zum Erstellen der Knotengruppe verwenden, können Sie den folgenden Befehl verwenden.

  ```
  eksctl create nodegroup --cluster my-cluster --managed=false --max-pods-per-node 110
  ```

- Verwaltet – Stellen Sie Ihre Knotengruppe mit einer der folgenden Optionen bereit:
  - Ohne Startvorlage oder mit Startvorlage ohne angegebene AMI-ID – Führen Sie das Verfahren in Erstellen einer verwalteten Knotengruppe aus. Verwaltete Knotengruppen berechnen automatisch den von Amazon EKS empfohlenen `max-pods`-Wert für Sie.
  - Mit einer Startvorlage mit einer angegebenen AMI-ID – Geben Sie in Ihrer Startvorlage eine Amazon EKS-optimierte AMI-ID oder ein benutzerdefiniertes AMI an, das auf dem Amazon EKS-optimierten AMI basiert, stellen Sie dann die Knotengruppe mithilfe einer Startvorlage bereit und geben Sie die folgenden Benutzerdaten an in der Startvorlage. Diese Benutzerdaten übergeben Argumente an die `bootstrap.sh`-Datei. Weitere Informationen zur Bootstrap-Datei finden Sie unter bootstrap.sh auf GitHub.

    ```
    /etc/eks/bootstrap.sh my-cluster \
      --use-max-pods false \
      --kubelet-extra-args '--max-pods=110'
    ```

    Wenn Sie `eksctl` zum Erstellen der Knotengruppe verwenden, können Sie den folgenden Befehl verwenden.

    ```
    eksctl create nodegroup --cluster my-cluster --max-pods-per-node 110
    ```

Wenn Sie ein benutzerdefiniertes AMI erstellt haben, das nicht auf dem für Amazon EKS optimierten AMI erstellt wurde, müssen Sie die Konfiguration selbst erstellen.

> **Note**
>
> Wenn Sie Pods auch IP-Adressen aus einem anderen Subnetz als dem der Instance zuweisen möchten, müssen Sie die Funktion in diesem Schritt aktivieren. Weitere Informationen finden Sie unter Benutzerdefinierte Netzwerke für Pods.

Windows

1. Aktivieren Sie die Zuweisung von IP-Präfixen.
   a. Öffnen Sie die `amazon-vpc-cni` `ConfigMap` zum Bearbeiten.

   ```
   kubectl edit configmap -n kube-system amazon-vpc-cni -o yaml
   ```

   b. Fügen Sie dem Abschnitt `data` die folgende Zeile hinzu.

   ```
     enable-windows-prefix-delegation: "true"
   ```

   c. Speichern Sie die Datei und schließen Sie den Editor.
   d. Vergewissern Sie sich, dass die Anmerkung der `ConfigMap` hinzugefügt wurde.

   ```
   kubectl get configmap -n kube-system amazon-vpc-cni -o
    "jsonpath={.data.enable-windows-prefix-delegation}"
   ```

   Wenn die zurückgegebene Ausgabe nicht `true` ist, ist möglicherweise ein Fehler aufgetreten. Versuchen Sie erneut, den Schritt abzuschließen.

   > **Important**
   >
   > Selbst wenn Ihr Subnetz über verfügbare IP-Adressen verfügt, werden Sie in den Knotenereignissen den folgenden Fehler sehen, wenn das Subnetz keine zusammenhängenden `/28`-Blöcke zur Verfügung hat.

```
"failed to allocate a private IP/Prefix address:
 InsufficientCidrBlocks: The specified subnet does not have enough
 free cidr blocks to satisfy the request"
```

Dies kann aufgrund der Fragmentierung vorhandener sekundärer IP-Adressen auftreten, die über ein Subnetz verteilt sind. Um diesen Fehler zu beheben, erstellen Sie entweder ein neues Subnetz und starten Sie dort Pods, oder verwenden Sie eine Amazon-EC2-Subnetz-CIDR-Reservierung, um Speicherplatz in einem Subnetz für die Verwendung mit Präfixzuweisung zu reservieren. Weitere Informationen erhalten Sie unter Subnetz-CIDR-Reservierungen im Amazon-VPC-Benutzerhandbuch.

2. (Optional) Geben Sie eine zusätzliche Konfiguration an, um das Verhalten Ihres Clusters vor der Skalierung und der dynamischen Skalierung zu steuern. Weitere Informationen finden Sie unter Konfigurationsoptionen mit aktiviertem Windows Präfix-Delegierungsmodus GitHub.

   a. Öffnen Sie die `amazon-vpc-cni` `ConfigMap` zum Bearbeiten.

   ```
   kubectl edit configmap -n kube-system amazon-vpc-cni -o yaml
   ```

   b. Ersetzen Sie die *`example values`* durch einen Wert größer als Null und fügen Sie die benötigten Einträge zum Abschnitt `data` der `ConfigMap` hinzu. Wenn Sie für `warm-ip-target` oder `minimum-ip-target` einen Wert festlegen, überschreibt der Wert jeden Wert, der für `warm-prefix-target` festgelegt wurde.

   ```
   warm-prefix-target: "1"
   warm-ip-target: "5"
   minimum-ip-target: "2"
   ```

   c. Speichern Sie die Datei und schließen Sie den Editor.

3. Erstellen Sie Windows-Knotengruppen mit mindestens einem Amazon-EC2-Nitro-Instance-Typ. Eine Liste der Nitro Instance-Typen finden Sie unter Instances built on the Nitro System im Amazon Amazon EC2-Benutzerhandbuch. Standardmäßig ist die maximale Anzahl von Pods, die Sie auf einem Knoten bereitstellen können, 110. Wenn Sie diese Zahl erhöhen oder verringern möchten, geben Sie in den Benutzerdaten für die Bootstrap-Konfiguration Folgendes an. Ersetzen Sie *`max-pods-quantity`* durch Ihren Wert für die maximale Anzahl von Pods.

```
-KubeletExtraArgs '--max-pods=max-pods-quantity'
```

Wenn Sie verwaltete Knotengruppen bereitstellen, muss diese Konfiguration zur Startvorlage hinzugefügt werden. Weitere Informationen finden Sie unter Anpassen verwalteter Knoten mit Startvorlagen. Weitere Hinweise zu den Konfigurationsparametern für das Windows-Bootstrap-Skript finden Sie unter Bootstrap-Skript-Konfigurationsparameter.

2. Nachdem Ihre Knoten bereitgestellt wurden, zeigen Sie die Knoten in Ihrem Cluster an.

```
kubectl get nodes
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                                                STATUS     ROLES    AGE   VERSION
ip-192-168-22-103.region-code.compute.internal   Ready      <none>   19m   v1.XX.X-eks-6b7464
ip-192-168-97-94.region-code.compute.internal    Ready      <none>   19m   v1.XX.X-eks-6b7464
```

3. Beschreiben Sie einen der Knoten, um den Wert von `max-pods` für den Knoten und die Anzahl der verfügbaren IP-Adressen zu bestimmen. Ersetzen Sie *192.168.30.193* durch die `IPv4`-Adresse im Namen eines Ihrer Knoten, der in der Ausgabe des vorherigen Schritts zurückgegeben wurde.

```
kubectl describe node ip-192-168-30-193.region-code.compute.internal | grep 'pods\|PrivateIPv4Address'
```

Eine Beispielausgabe sieht wie folgt aus.

```
pods:                                       110
vpc.amazonaws.com/PrivateIPv4Address:  144
```

In der vorherigen Ausgabe ist `110` die maximale Anzahl von Pods, die Kubernetes auf dem Knoten bereitstellt, auch wenn *144* IP-Adressen verfügbar sind.

Sicherheitsgruppen für Pods

Sicherheitsgruppen für Pods integrieren Amazon-EC2-Sicherheitsgruppen in KubernetesPods. Sie können Amazon-EC2-Sicherheitsgruppen verwenden, um Regeln zu definieren, die ein- und ausgehenden Netzwerkverkehr zu und von Pods zulassen, die Sie auf Knoten bereitstellen, die auf vielen Amazon-EC2-Instance-Typen und Fargate ausgeführt werden. Eine ausführliche Erläuterung dieser Funktion finden Sie im Blogbeitrag Einführung in Sicherheitsgruppen für Pods.

Überlegungen

- Berücksichtigen Sie vor der Bereitstellung von Sicherheitsgruppen für Pods die folgenden Einschränkungen und Bedingungen:
- Sicherheitsgruppen für Pods können nicht mit Windows-Knoten verwendet werden.
- Sicherheitsgruppen für Pods können mit für die `IPv6`-Produktfamilie konfigurierten Clustern verwendet werden, die Amazon-EC2-Knoten enthalten. Hierfür muss mindestens die Version 1.16.0 des Amazon-VPC-CNI-Plug-ins verwendet werden. Sie können Sicherheitsgruppen für Pods mit für die `IPv6`-Produktfamilie konfigurierten Clustern verwenden, die nur Fargate-Knoten enthalten. Hierfür muss mindestens die Version 1.7.7 des Amazon-VPC-CNI-Plug-ins verwendet werden. Weitere Informationen finden Sie unter `IPv6`Adressen für ClusterPods, und services.
- Sicherheitsgruppen für Pods werden von den meisten Nitro-basierten Amazon-EC2-Instance-Familien unterstützt, jedoch nicht von allen Generationen einer Familie. Zum Beispiel werden die Familie und die Generationen `m5` `c5` `r5` `m6g`c`6g`,,, und `r6g` Instance unterstützt. Es werden keine Instance-Typen in der `t`-Familie unterstützt. Eine vollständige Liste der unterstützten Instance-Typen finden Sie in der limits.go-Datei auf Github. Ihre Knoten müssen von einem der aufgelisteten Instance-Typen sein, die in der Datei mit `IsTrunkingCompatible: true` gekennzeichnet sind.
- Wenn Sie auch Pod-Sicherheitsrichtlinien verwenden, um den Zugriff auf die Pod-Mutation einzuschränken, dann muss der `eks:vpc-resource-controller` Kubernetes-Benutzer im Kubernetes `ClusterRoleBinding` für das `role` angegeben werden, dem Ihr `psp` zugewiesen ist. Bei Verwendung der standardmäßigen Amazon-EKS-Elemente `psp`, `role` und `ClusterRoleBinding` ist dies `eks:podsecuritypolicy:authenticated` `ClusterRoleBinding`. Fügen Sie beispielsweise den Benutzer zum `subjects:`-Abschnitt hinzu, wie im folgenden Beispiel gezeigt:

```
[...]
subjects:
  - kind: Group
```

```
  apiGroup: rbac.authorization.k8s.io
  name: system:authenticated
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: eks:vpc-resource-controller
- kind: ServiceAccount
  name: eks-vpc-resource-controller
```

- Wenn Sie benutzerdefinierte Netzwerk- und Sicherheitsgruppen für Pods zusammen verwenden, wird anstelle der in `ENIConfig` angegebenen Sicherheitsgruppe die durch Sicherheitsgruppen für Pods angegebene Sicherheitsgruppe verwendet.
- Wenn Sie Version `1.10.2` oder älter des Amazon-VPC-CNI-Plugins verwenden und die Einstellung `terminationGracePeriodSeconds` in Ihre Pod-Spezifikation aufnehmen, darf der Wert der Einstellung nicht Null sein.
- Wenn Sie Version `1.10` oder eine frühere Version des Amazon-VPC-CNI-Plugins verwenden oder Version `1.11` mit `POD_SECURITY_GROUP_ENFORCING_MODE` = `strict`, was die Standardeinstellung ist, dann werden Kubernetes-Services vom Typ `NodePort` und `LoadBalancer`, die Instance-Ziele verwenden, für die `externalTrafficPolicy` auf `Local` eingestellt ist, nicht für Pods unterstützt, denen Sie Sicherheitsgruppen zuweisen. Weitere Informationen zur Verwendung eines Load Balancers mit Instance-Zielen finden Sie unter Network Load Balancing in Amazon EKS.
- Wenn Sie Version `1.10` des Amazon-VPC-CNI-Plugins oder höher verwenden oder Version `1.11` mit `POD_SECURITY_GROUP_ENFORCING_MODE=strict`, was die Standardeinstellung ist, ist die Quell-NAT für von Pods mit zugewiesenen Sicherheitsgruppen ausgehenden Datenverkehr deaktiviert, damit Sicherheitsgruppenregeln für ausgehenden Datenverkehr angewendet werden können. Um auf das Internet zuzugreifen, müssen Pods mit zugewiesenen Sicherheitsgruppen auf Knoten gestartet werden, die in einem privaten Subnetz bereitgestellt werden, das mit einem NAT-Gateway oder einer NAT-Instance konfiguriert ist. Pods mit zugewiesenen Sicherheitsgruppen, die in öffentlichen Subnetzen bereitgestellt werden, haben keinen Internetzugriff.

  Wenn Sie Version `1.11` oder höher des Plugins mit `POD_SECURITY_GROUP_ENFORCING_MODE=standard` verwenden, wird Pod-Datenverkehr, der für außerhalb der VPC bestimmt ist, an die IP-Adresse der primären Netzwerkschnittstelle der Instance weitergeleitet. Für diesen Datenverkehr werden die Regeln in den Sicherheitsgruppen für die primäre Netzwerkschnittstelle anstelle der Regeln in den Pod's-Sicherheitsgruppen verwendet.
- Um die Calico-Netzwerkrichtlinie mit Pods zu verwenden, denen Sicherheitsgruppen zugewiesen sind, müssen Sie Version `1.11.0` oder höher des Amazon-VPC-CNI-Plugins verwenden und

`POD_SECURITY_GROUP_ENFORCING_MODE=standard` festlegen. Andernfalls wird Datenverkehr zu und von Pods mit zugehörigen Sicherheitsgruppen nicht von den Calico-Netzwerkrichtlinien geregelt, sondern es gelten ausschließlich die Amazon-EC2-Sicherheitsgruppen. Informationen zum Aktualisieren Ihrer Version von Amazon VPC CNI finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

- Pods, die auf Amazon EC2-Knoten ausgeführt werden, die Sicherheitsgruppen in Clustern verwenden, die Nodelocal DNSCache verwenden, werden nur mit Version `1.11.0` oder höher des Amazon-VPC-CNI-Plugins mit der Einstellung `POD_SECURITY_GROUP_ENFORCING_MODE=standard` unterstützt. Informationen zum Aktualisieren Ihrer Version des Amazon-VPC-CNI-Plug-ins finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.
- Sicherheitsgruppen für Pods können zu einer höheren Pod-Startlatenz für Pods mit hoher Abwanderung führen. Dies ist auf eine Ratenbegrenzung im Ressourcencontroller zurückzuführen.

Konfigurieren Sie das Amazon VPC CNI plugin for Kubernetes für Sicherheitsgruppen für Pods

Sicherheitsgruppen für Pods bereitstellen

Wenn Sie Sicherheitsgruppen nur für Fargate-Pods verwenden und keine Amazon-EC2-Knoten in Ihrem Cluster haben, fahren Sie mit Schritt Eine Beispielanwendung bereitstellen fort.

1. Überprüfen Sie Ihre aktuelle Amazon VPC CNI plugin for Kubernetes-Version mit dem folgenden Befehl:

   ```
   kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni:
    | cut -d : -f 3
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.7.6
   ```

   Wenn Ihre Amazon VPC CNI plugin for Kubernetes-Version älter als `1.7.7` ist, aktualisieren Sie das Plugin auf Version `1.7.7` oder höher. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.
2. Fügen Sie die verwaltete `AmazonEKSVPCResourceController`-IAM-Richtlinie der Cluster-Rolle hinzu, die Ihrem Amazon-EKS-Cluster zugeordnet ist. Die Richtlinie ermöglicht es der

Rolle, Netzwerkschnittstellen, ihre privaten IP-Adressen sowie deren An- und Abkopplung an und von Netzwerk-Instances zu verwalten.

a. Rufen Sie den Namen Ihrer Cluster-IAM-Rolle ab und speichern Sie ihn in einer Variablen. Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.

```
cluster_role=$(aws eks describe-cluster --name my-cluster --query
 cluster.roleArn --output text | cut -d / -f 2)
```

b. Fügen Sie der Rolle die -Richtlinie an.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSVPCResourceController --role-name $cluster_role
```

3. Aktivieren Sie das Amazon-VPC-CNI-Add-on, um Netzwerkschnittstellen für Pods zu verwalten, indem Sie die `ENABLE_POD_ENI`-Variable im `aws-node` DaemonSet auf `true` setzen. Wenn diese Einstellung auf `true` festgelegt wurde, erstellt das Add-on für jeden Knoten im Cluster eine benutzerdefinierte cninode-Ressource. Der VPC-Ressourcen-Controller erstellt und verknüpft eine spezielle Netzwerkschnittstelle, die als Trunk-Netzwerkschnittstelle mit der Beschreibung `aws-k8s-trunk-eni` bezeichnet wird.

```
kubectl set env daemonset aws-node -n kube-system ENABLE_POD_ENI=true
```

> **Note**
>
> Die Trunk-Netzwerkschnittstelle ist in der maximalen Anzahl von Netzwerkschnittstellen enthalten, die vom Instance-Typ unterstützt werden. Eine Liste der maximalen Anzahl von Netzwerkschnittstellen, die von jedem Instance-Typ unterstützt werden, finden Sie unter IP-Adressen pro Netzwerkschnittstelle pro Instance-Typ im Amazon EC2 EC2-Benutzerhandbuch. Wenn an Ihren Knoten bereits die maximale Anzahl an Standardnetzwerkschnittstellen angeschlossen ist, reserviert der VPC-Ressourcencontroller einen Speicherplatz. Sie müssen Ihre laufenden Pods so weit herunterskalieren, dass der Controller eine Standard-Netzwerkschnittstelle trennen und löschen, die Trunk-Netzwerkschnittstelle erstellen und an die Instance anhängen kann.

4. Mit dem folgenden Befehl können Sie ermitteln, welche Ihrer Knoten über eine benutzerdefinierte Ressource vom Typ `CNINode` verfügen. Wenn `No resources found` zurückgegeben wird, warten Sie einige Sekunden und versuchen Sie es dann erneut. Der vorherige Schritt erfordert

einen Neustart des Amazon VPC CNI plugin for Kubernetes Pods, was mehrere Sekunden dauert.

```
$ kubectl get cninode -A
     NAME FEATURES
     ip-192-168-64-141.us-west-2.compute.internal
 [{"name":"SecurityGroupsForPods"}]
     ip-192-168-7-203.us-west-2.compute.internal [{"name":"SecurityGroupsForPods"}]
```

Wenn Sie VPC-CNI-Versionen verwenden, die älter als `1.15` sind, wurden anstelle der benutzerdefinierten `CNINode`-Ressource Knotenbezeichnungen verwendet. Mit dem folgenden Befehl können Sie ermitteln, bei welchen Ihrer Knoten die Knotenbezeichnung `aws-k8s-trunk-eni` auf `true` festgelegt ist. Wenn `No resources found` zurückgegeben wird, warten Sie einige Sekunden und versuchen Sie es dann erneut. Der vorherige Schritt erfordert einen Neustart des Amazon VPC CNI plugin for Kubernetes Pods, was mehrere Sekunden dauert.

```
kubectl get nodes -o wide -l vpc.amazonaws.com/has-trunk-attached=true
-
```

Nachdem die Trunk-Netzwerkschnittstelle erstellt wurde, werden Pods sekundäre IP-Adressen von der Trunk- oder Standardnetzwerkschnittstelle zugewiesen. Die Trunk-Schnittstelle wird automatisch gelöscht, wenn der Knoten gelöscht wird.

Wenn Sie in einem späteren Schritt eine Sicherheitsgruppe für einen Pod bereitstellen, erstellt der VPC-Ressourcen-Controller eine spezielle Netzwerkschnittstelle, die als Zweigstellennetzwerkschnittstelle bezeichnet wird, mit einer Beschreibung von `aws-k8s-branch-eni` und ordnet ihr die Sicherheitsgruppen zu. Neben den an den Knoten angeschlossenen Standard- und Amtsnetzschnittstellen werden Nebenstellennetzschnittstellen erstellt.

Wenn Sie Lebend- oder Bereitschaftserkennung verwenden, müssen Sie auch das frühe TCP-Demux deaktivieren, damit das `kubelet` über TCP eine Verbindung zu Pods auf Zweigstellennetzwerkschnittstellen herstellen kann. Führen Sie den folgenden Befehl aus, um das frühe TCP-Demux zu deaktivieren:

```
kubectl patch daemonset aws-node -n kube-system \
```

```
  -p '{"spec": {"template": {"spec": {"initContainers": [{"env":
[{"name":"DISABLE_TCP_EARLY_DEMUX","value":"true"}],"name":"aws-vpc-cni-
init"}]}}}}'
```

> **Note**
>
> Wenn Sie `1.11.0` oder höher des Amazon VPC CNI plugin for Kubernetes-Add-ons verwenden und `POD_SECURITY_GROUP_ENFORCING_MODE=standard` gilt, wie im nächsten Schritt beschrieben, müssen Sie den vorherigen Befehl nicht ausführen.

5. Wenn Ihr Cluster `NodeLocal DNSCache` verwendet oder Sie auf Pods, die eigene Sicherheitsgruppen haben, die Calico-Netzwerkrichtlinie anwenden möchten, oder wenn Sie für Pods, denen Sie Sicherheitsgruppen zuweisen möchten, Kubernetes-Services vom Typ `NodePort` und `LoadBalancer` haben, die Instance-Ziele mit einer `externalTrafficPolicy` auf `Local` verwenden, dann müssen Sie Version `1.11.0` oder höher des Amazon VPC CNI plugin for Kubernetes-Add-Ons verwenden, und Sie müssen die folgende Einstellung aktivieren:

```
kubectl set env daemonset aws-node -n kube-system
 POD_SECURITY_GROUP_ENFORCING_MODE=standard
```

> **Important**
>
> - Pod-Sicherheitsgruppenregeln werden nicht auf den Datenverkehr zwischen Pods oder zwischen Pods und services, wie beispielsweise `kubelet` oder `nodeLocalDNS`, angewendet, die sich auf demselben Knoten befinden. Pods, die verschiedene Sicherheitsgruppen auf demselben Knoten verwenden, können nicht kommunizieren, da sie in verschiedenen Subnetzen konfiguriert sind und das Routing zwischen diesen Subnetzen deaktiviert ist.
> - Für ausgehenden Datenverkehr von Pods an Adressen außerhalb der VPC wird die Netzwerkadresse in die IP-Adresse der primären Netzwerkschnittstelle der Instance übersetzt (es sei denn, Sie haben auch `AWS_VPC_K8S_CNI_EXTERNALSNAT=true` festgelegt). Für diesen Datenverkehr werden die Regeln in den Sicherheitsgruppen für die primäre Netzwerkschnittstelle anstelle der Regeln in den Pod's-Sicherheitsgruppen verwendet.
> - Damit diese Einstellung auf vorhandene Pods angewendet wird, müssen Sie die Pods oder die Knoten, auf denen die Pods ausgeführt werden, neu starten.

Eine Beispielanwendung bereitstellen

Um Sicherheitsgruppen für Pods zu verwenden, müssen Sie über eine bestehende Sicherheitsgruppe verfügen und auf Ihrem Cluster die Amazon-EKS-SecurityGroupPolicy bereitstellen, wie nachfolgend beschrieben. In den folgenden Schritten wird gezeigt, wie Sie die Sicherheitsgruppenrichtlinie für ein Pod verwenden. Wenn nicht anders angegeben, führen Sie alle Schritte vom selben Terminal aus, da Variablen in den folgenden Schritten verwendet werden, die nicht über Terminals hinweg bestehen bleiben.

Ein Beispiel-Pod mit einer Sicherheitsgruppe bereitstellen

1. Erstellen Sie einen Kubernetes-Namespace zum Bereitstellen von Ressourcen. Sie können *my-namespace* durch den Namen eines Namespace ersetzen, den Sie verwenden möchten.

```
kubectl create namespace my-namespace
```

2. Stellen Sie ein Amazon EKS `SecurityGroupPolicy` in Ihrem Cluster bereit.

   a. Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Sie können *podSelector* durch **serviceAccountSelector** ersetzen, wenn Sie Pods lieber basierend auf Servicekontomarkierungen auswählen möchten. Sie müssen den einen oder anderen Selektor angeben. Ein leeres `podSelector` (Beispiel: `podSelector: {}`) wählt alle Pods im Namespace aus. Sie können *my-role* in den Namen Ihrer Rolle ändern. Ein leeres `serviceAccountSelector` wählt alle Servicekonten im Namespace aus. Sie können *my-security-group-policy* durch einen Namen für Ihr `SecurityGroupPolicy` und *my-namespace* durch den Namespace ersetzen, in dem Sie das `SecurityGroupPolicy` erstellen möchten.

   Sie müssen *my_pod_security_group_id* durch die ID einer bestehenden Sicherheitsgruppe ersetzen. Wenn Sie nicht über eine bestehende Sicherheitsgruppe verfügen, müssen Sie eine erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Sicherheitsgruppen für Linux-Instances im Amazon-EC2-Benutzerhandbuch. Sie können 1–5 Sicherheitsgruppen-IDs angeben. Falls Sie mehr als eine ID angeben, gilt die Kombination aller Regeln in allen Sicherheitsgruppen für die ausgewählten Pods.

```
cat >my-security-group-policy.yaml <<EOF
apiVersion: vpcresources.k8s.aws/v1beta1
kind: SecurityGroupPolicy
metadata:
```

```
  name: my-security-group-policy
  namespace: my-namespace
spec:
  podSelector:
    matchLabels:
      role: my-role
  securityGroups:
    groupIds:
      - my_pod_security_group_id
EOF
```

Important

Die Sicherheitsgruppe oder -gruppen, die Sie für Ihre Pods angeben, müssen die folgenden Kriterien erfüllen:

- Sie müssen vorhanden sein. Wenn sie nicht vorhanden sind, bleibt Ihr Pod im Erstellungsprozess hängen, wenn Sie einen Pod bereitstellen, der dem Selektor entspricht. Wenn Sie den Pod beschreiben, wird eine Fehlermeldung ähnlich der folgenden angezeigt: `An error occurred (InvalidSecurityGroupID.NotFound) when calling the CreateNetworkInterface operation: The securityGroup ID 'sg-05b1d815d1EXAMPLE' does not exist.`
- Diese müssen eingehende Kommunikation von der auf Ihre Knoten angewendeten Sicherheitsgruppe (für `kubelet`) über alle Ports erlauben, für die Sie Tests konfiguriert haben.
- Diese müssen ausgehende Kommunikation über die TCP- und UDP-Ports 53 zu einer Sicherheitsgruppe zulassen, die dem Pods (oder den Knoten, auf denen die Pods ausgeführt werden) zugeordnet ist, auf denen CoreDNS ausgeführt wird. Die Sicherheitsgruppe für Ihr CoreDNS Pods muss eingehenden TCP- und UDP-Port-53-Datenverkehr von der von Ihnen angegebenen Sicherheitsgruppe zulassen.
- Sie müssen über notwendige Regeln für eingehenden und ausgehenden Datenverkehr verfügen, um mit anderen Pods zu kommunizieren.
- Wenn Sie die Sicherheitsgruppe mit Fargate verwenden, stellen Sie sicher, dass sie über Regeln verfügen, die es den Pods ermöglichen, mit der Kubernetes-

Steuerebene zu kommunizieren. Am einfachsten erreichen Sie dies, indem Sie die Cluster-Sicherheitsgruppe als eine der Sicherheitsgruppen angeben.

Sicherheitsgruppenrichtlinien gelten nur für neu geplante Pods. Sie haben keinen Einfluss auf laufende Pods.

b. Die Richtlinie bereitstellen.

```
kubectl apply -f my-security-group-policy.yaml
```

3. Stellen Sie eine Beispielanwendung mit einem Label bereit, die dem *my-role*-Wert für *podSelector* entspricht, den Sie in einem vorherigen Schritt angegeben haben.

a. Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie die *Beispielwerte* durch Ihre eigenen, und führen Sie dann den geänderten Befehl aus. Wenn Sie *my-role* ersetzen, stellen Sie sicher, dass es dem Wert entspricht, den Sie in einem vorherigen Schritt für den Selektor angegeben haben.

```
cat >sample-application.yaml <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
  namespace: my-namespace
  labels:
    app: my-app
spec:
  replicas: 4
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        role: my-role
    spec:
      terminationGracePeriodSeconds: 120
      containers:
      - name: nginx
        image: public.ecr.aws/nginx/nginx:1.23
```

```
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: my-app
  namespace: my-namespace
  labels:
    app: my-app
spec:
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
EOF
```

b. Stellen Sie die Anwendung mit dem folgenden Befehl bereit. Wenn Sie die Anwendung bereitstellen, entspricht das Amazon VPC CNI plugin for Kubernetes der `role`-Markierung und die Sicherheitsgruppen, die Sie im vorherigen Schritt angegeben haben, werden auf den Pod angewendet.

```
kubectl apply -f sample-application.yaml
```

4. Zeigen Sie die Pods an, die mit der Beispielanwendung bereitgestellt wurden. Für den Rest dieses Themas wird dieses Terminal als `TerminalA` bezeichnet.

```
kubectl get pods -n my-namespace -o wide
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                          READY   STATUS    RESTARTS   AGE     IP
       NODE                                            NOMINATED NODE   READINESS
 GATES
my-deployment-5df6f7687b-4fbjm   1/1     Running   0          7m51s   192.168.53.48
    ip-192-168-33-28.region-code.compute.internal   <none>           <none>
my-deployment-5df6f7687b-j9fl4   1/1     Running   0          7m51s
  192.168.70.145   ip-192-168-92-33.region-code.compute.internal   <none>
 <none>
```

```
my-deployment-5df6f7687b-rjxcz   1/1     Running   0          7m51s
  192.168.73.207   ip-192-168-92-33.region-code.compute.internal   <none>
 <none>
my-deployment-5df6f7687b-zmb42   1/1     Running   0          7m51s   192.168.63.27
    ip-192-168-33-28.region-code.compute.internal   <none>           <none>
```

> **Note**
>
> - Wenn Pods im `Waiting`-Zustand hängen bleiben, führen Sie `kubectl describe pod` ***my-deployment-xxxxxxxxxx-xxxxx*** `-n` *my-namespace* aus. Wenn Sie `Insufficient permissions: Unable to create Elastic Network Interface.` sehen, bestätigen Sie, dass Sie die IAM-Richtlinie in einem vorherigen Schritt zur IAM-Cluster-Rolle hinzugefügt haben.
> - Wenn Pods im `Pending`-Status hängen bleiben, überprüfen Sie, ob der Instance-Typ Ihres Knotens in `limits.go` aufgeführt ist, und dass die maximale Anzahl der vom Instance-Typ unterstützten Zweigstellennetzwerkschnittstellen multipliziert mit der Anzahl der Knoten in Ihrer Knotengruppe noch nicht erreicht wurde. Eine `m5.large`-Instance unterstützt beispielsweise neun Zweigstellen-Netzwerkschnittstellen. Wenn Ihre Knotengruppe fünf Knoten hat, können maximal 45 Zweigstellen-Netzwerkschnittstellen für die Knotengruppe erstellt werden. Der 46. Pod, den Sie bereitstellen möchten, bleibt im `Pending`-Zustand, bis ein anderer Pod mit zugehörigen Sicherheitsgruppen gelöscht wird.

Wenn Sie `kubectl describe pod` ***my-deployment-xxxxxxxxxx-xxxxx*** **`-n`** ***my-namespace*** ausführen und eine Meldung ähnlich der folgenden Meldung angezeigt wird, kann sie problemlos ignoriert werden. Diese Meldung kann erscheinen, wenn das Amazon VPC CNI plugin for Kubernetes versucht, das Host-Netzwerk einzurichten, und dies fehlschlägt, während die Netzwerkschnittstelle erstellt wird. Das Plug-In protokolliert dieses Ereignis, bis die Netzwerkschnittstelle erstellt wird.

```
Failed to create Pod sandbox: rpc error: code = Unknown desc = failed to set up
 sandbox container
 "e24268322e55c8185721f52df6493684f6c2c3bf4fd59c9c121fd4cdc894579f" network for Pod
 "my-deployment-5df6f7687b-4fbjm": networkPlugin
cni failed to set up Pod "my-deployment-5df6f7687b-4fbjm-c89wx_my-namespace"
 network: add cmd: failed to assign an IP address to container
```

Die maximale Anzahl von Pods, die auf dem Instance-Typ ausgeführt werden können, darf nicht überschritten werden. Eine Liste der maximalen Anzahl von Pods, die Sie auf jedem Instance-Typ ausführen können, finden Sie unter eni-max-pods.txt auf GitHub. Wenn Sie einen Pod mit zugeordneten Sicherheitsgruppen oder den Knoten löschen, auf dem der Pod ausgeführt wird, löscht der VPC-Ressourcen-Controller die Zweigstellennetzwerkschnittstelle. Wenn Sie einen Cluster mit Pods löschen, indem Sie Pods für Sicherheitsgruppen verwenden, löscht der Controller die Netzwerkschnittstellen der Zweigstelle nicht, sodass Sie diese selbst löschen müssen. Informationen zum Löschen von Netzwerkschnittstellen finden Sie unter Löschen einer Netzwerkschnittstelle im Amazon EC2 EC2-Benutzerhandbuch.

5. Setzen Sie in einem separaten Terminal eine Shell in einen der Pods ein. Für den Rest dieses Themas wird dieses Terminal als `TerminalB` bezeichnet. Ersetzen Sie *`5df6f7687b-4fbjm`* mit der ID eines der Pods, die Sie in der Ausgabe im vorherigen Schritt erhalten haben.

```
kubectl exec -it -n my-namespace my-deployment-5df6f7687b-4fbjm -- /bin/bash
```

6. Überprüfen Sie in der Shell von `TerminalB`, dass die Beispielanwendung funktioniert.

```
curl my-app
```

Eine Beispielausgabe sieht wie folgt aus.

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
[...]
```

Sie haben diese Ausgabe erhalten, weil alle Pods, auf denen die Anwendung ausgeführt wird, mit der Sicherheitsgruppe verknüpft sind, die Sie erstellt haben. Diese Gruppe enthält eine Regel, die den gesamten Datenverkehr zwischen allen Pods zulässt, mit denen die Sicherheitsgruppe verknüpft ist. DNS-Datenverkehr wird von dieser Sicherheitsgruppe an die Cluster-Sicherheitsgruppe übertragen, die mit Ihren Knoten verknüpft ist. Auf den Knoten werden CoreDNS-Pods ausgeführt, für die Ihr Pods eine Namenssuche durchgeführt hat.

7. Entfernen Sie von `TerminalA` aus die Sicherheitsgruppenregeln, die die DNS-Kommunikation mit der Cluster-Sicherheitsgruppe ermöglichen, aus Ihrer Sicherheitsgruppe. Wenn Sie die DNS-Regeln in einem der vorherigen Schritte nicht zur Cluster-Sicherheitsgruppe hinzugefügt haben,

ersetzen Sie *$my_cluster_security_group_id* mit der ID der Sicherheitsgruppe, in der Sie die Regeln erstellt haben.

```
aws ec2 revoke-security-group-ingress --group-id $my_cluster_security_group_id --
security-group-rule-ids $my_tcp_rule_id
aws ec2 revoke-security-group-ingress --group-id $my_cluster_security_group_id --
security-group-rule-ids $my_udp_rule_id
```

8. Versuchen Sie über `TerminalB` noch einmal, auf die Anwendung zuzugreifen.

```
curl my-app
```

Eine Beispielausgabe sieht wie folgt aus.

```
curl: (6) Could not resolve host: my-app
```

Der Versuch schlägt fehl, weil der Pod nicht mehr auf die CoreDNS-Pods zugreifen kann, denen die Cluster-Sicherheitsgruppe zugeordnet ist. Die Cluster-Sicherheitsgruppe verfügt nicht mehr über die Sicherheitsgruppenregeln, die eine DNS-Kommunikation von der Sicherheitsgruppe, die Ihrem Pod zugeordnet ist, erlauben.

Wenn Sie versuchen, über die IP-Adressen auf die Anwendung zuzugreifen, die für einen der Pods in einem vorherigen Schritt zurückgegeben wurden, erhalten Sie dennoch eine Antwort, da zwischen Pods, denen die Sicherheitsgruppe zugeordnet ist, alle Ports erlaubt sind, und eine Namenssuche daher nicht erforderlich ist.

9. Wenn Sie mit dem Experimentieren fertig sind, können Sie die von Ihnen erstellte Sicherheitsgruppen-Richtlinie, Anwendung und Sicherheitsgruppe entfernen. Führen Sie die folgenden Befehle über `TerminalA` aus.

```
kubectl delete namespace my-namespace
aws ec2 revoke-security-group-ingress --group-id $my_pod_security_group_id --
security-group-rule-ids $my_inbound_self_rule_id
wait
sleep 45s
aws ec2 delete-security-group --group-id $my_pod_security_group_id
```

Mehrere Netzwerkschnittstellen für Pods

Multus CNI ist ein Container-Network-Interface(CNI)-Plugin für Amazon EKS, das das Anfügen mehrerer Netzwerkschnittstellen an einen Pod ermöglicht. Weitere Informationen finden Sie in der Multus-CNI-Dokumentation auf GitHub.

In Amazon EKS hat jeder Pod eine Netzwerkschnittstelle, die vom Amazon-VPC-CNI-Plugin zugewiesen wird. Mit Multus können Sie einen mehrfach vernetzten Pod mit mehreren Schnittstellen erstellen. Dies wird durch Multus erreicht, der als "meta-plugin" fungiert; ein CNI-Plugin, das mehrere andere CNI-Plugins aufrufen kann. Die AWS-Unterstützung für Multus wird mit dem Amazon-VPC-CNI-Plugin als Standard-Delegierungs-Plugin konfiguriert.

Überlegungen

- Amazon EKS wird keine Single-Root-I/O-Virtualisierung-(SR-IOV)- und Data-Plane-Development-Kit-(DPDK)-CNI-Plugins erstellen und veröffentlichen. Sie können jedoch eine Paketbeschleunigung erreichen, indem Sie sich über ein von Multus verwaltetes Hostgerät und `ipvlan`-Plugins direkt mit Amazon EC2 Elastic Network Adapters (ENA) verbinden.
- Amazon EKS unterstützt Multus, das einen generischen Prozess bereitstellt, der eine einfache Verkettung zusätzlicher CNI-Plugins ermöglicht. Multus und der Verkettungsprozess werden unterstützt, aber AWS bietet keine Unterstützung für alle kompatiblen CNI-Plugins, die verkettet werden können, oder Probleme, die in diesen CNI-Plugins auftreten können, die nichts mit der Verkettungskonfiguration zu tun haben.
- Amazon EKS bietet Support und Lebenszyklus-Verwaltung für das Multus-Plugin, ist jedoch nicht für IP-Adressen oder zusätzliche Verwaltung im Zusammenhang mit den zusätzlichen Netzwerkschnittstellen verantwortlich. Die IP-Adresse und Verwaltung der Standard-Netzwerkschnittstelle, die das Amazon-VPC-CNI-Plugin verwendet, bleibt unverändert.
- Nur das Amazon-VPC-CNI-Plugin wird offiziell als Standard-Delegierungs-Plugin unterstützt. Sie müssen das veröffentlichte Multus-Installationsmanifest ändern, um das Standard-Delegierten-Plugin auf ein alternatives CNI umzukonfigurieren, wenn Sie das Amazon VPC-CNI-Plugin nicht für das primäre Netzwerk verwenden möchten.
- Multus wird nur unterstützt, wenn das Amazon-VPC-CNI als primäres CNI verwendet wird. Wir unterstützen das Amazon-VPC-CNI nicht, wenn es für Schnittstellen höherer Ordnung verwendet wird, sekundär oder anderweitig.
- Um zu verhindern, dass das Amazon-VPC-CNI-Plugin versucht, zusätzliche Netzwerkschnittstellen zu verwalten, die Pods zugewiesen sind, können Sie den Netzwerkschnittstellen den folgenden Tag hinzufügen.

Schlüssel: `node.k8s.amazonaws.com/no_manage`

Wert: `true`

- Multus ist mit Netzwerkrichtlinien kompatibel, die Richtlinie muss jedoch um Ports und IP-Adressen erweitert werden, die Teil von zusätzlichen Netzwerkschnittstellen sein können, die an Pods angeschlossen sind.

Eine Einführung in die Implementierung finden Sie im Multus-Einrichtungsanleitung auf GitHub.

Alternative kompatible CNI-Plugins

Das Amazon VPC CNI plugin for Kubernetes ist das einzige CNI-Plugin, das von Amazon EKS unterstützt wird. Amazon EKS wird Upstream-Kubernetes ausgeführt, sodass Sie alternative kompatible CNI-Plugins auf Amazon-EC2-Knoten in Ihrem Cluster installieren können. Wenn Sie Fargate-Knoten in Ihrem Cluster haben, befindet sich Amazon VPC CNI plugin for Kubernetes bereits auf Ihren Fargate-Knoten. Es ist das einzige CNI-Plugin, das Sie mit Fargate-Knoten verwenden können. Ein Versuch, ein alternatives CNI-Plugin auf Fargate-Knoten zu installieren, schlägt fehl.

Wenn Sie beabsichtigen, ein alternatives CNI-Plugin auf Amazon-EC2-Knoten zu verwenden, empfehlen wir Ihnen, sich kommerziellen Support für das Plugin zu holen oder über das interne Fachwissen zu verfügen, um Fehler zu beheben und Korrekturen zum CNI-Plugin-Projekt beizutragen.

Amazon EKS pflegt Beziehungen zu einem Netzwerk von Partnern, die Support für alternative kompatible CNI-Plugins bieten. Einzelheiten zu den Versionen, Qualifizierungen und durchgeführten Tests finden Sie in der folgenden Partnerdokumentation.

Partner	Produkt	Dokumentation
Tigera	Calico	Installationsanleitungen
Isovalent	Cilium	Installationsanleitungen
Juniper	Cloudnatives Contrail Networking (CN2)	Installationsanleitungen
VMware	Antrea	Installationsanleitungen

Amazon EKS bemüht sich, Ihnen eine große Auswahl an Optionen zu bieten, um alle Anwendungsfälle abzudecken.

Alternative kompatible Netzwerkrichtlinien-Erweiterungen

Calico ist eine weit verbreitete Lösung für Container-Netzwerke und Sicherheit. Die Verwendung Calico auf EKS bietet eine vollständig konforme Durchsetzung der Netzwerkrichtlinien für Ihre EKS-Cluster. Darüber hinaus können Sie sich für die Verwendung Calico von Netzwerken entscheiden, bei denen IP-Adressen von Ihrer zugrunde liegenden VPC gespeichert werden. Calico Cloud erweitert die Funktionen von Calico Open Source und bietet erweiterte Sicherheits- und Beobachtungsfunktionen.

Was ist die AWS Load Balancer Controller?

Der AWS Load Balancer Controller verwaltet AWS Elastic Load Balancer für einen Kubernetes Cluster. Sie können den Controller verwenden, um Ihre Cluster-Apps dem Internet zugänglich zu machen. Der Controller stellt AWS Load Balancer bereit, die auf Clusterdienst- oder Ingress-Ressourcen verweisen. Mit anderen Worten, der Controller erstellt eine einzelne IP-Adresse oder einen einzelnen DNS-Namen, der auf mehrere Pods in Ihrem Cluster verweist.

Der Controller überwacht unsere Ressourcen KubernetesIngress. Service Als Reaktion darauf erstellt es die entsprechenden AWS Elastic Load Balancing Balancing-Ressourcen. Sie können das spezifische Verhalten der Load Balancer konfigurieren, indem Sie Anmerkungen auf die Kubernetes Ressourcen anwenden. Beispielsweise können Sie mithilfe von Anmerkungen AWS Sicherheitsgruppen an Load Balancer anhängen.

Der Controller stellt die folgenden Ressourcen bereit:

Kubernetes `Ingress`

> Der LBC erstellt einen AWS Application Load Balancer (ALB), wenn Sie einen erstellen. Kubernetes `Ingress` Überprüfen Sie die Anmerkungen, die Sie auf eine Ingress-Ressource anwenden können.

Kubernetes-Service des Typs `LoadBalancer`

> Der LBC erstellt einen AWS Network Load Balancer (NLB), wenn Sie einen Kubernetes Dienst vom Typ erstellen. `LoadBalancer` Prüfen Sie die Anmerkungen, die Sie auf eine Dienstressource anwenden können.

In der Vergangenheit wurde der Kubernetes Network Load Balancer für Instanzziele verwendet, aber der LBC wurde für IP-Ziele verwendet. Mit der AWS Load Balancer Controller-Version `2.3.0` oder höher können Sie NLBs für beide Zieltypen erstellen. Weitere Informationen zu NLB-Zieltypen finden Sie unter Zieltyp im Benutzerhandbuch für Network Load Balancer.

Der Controller ist ein Open-Source-Projekt, auf dem gemanagt wird. GitHub

Bevor Sie den Controller bereitstellen, empfehlen wir Ihnen, die Voraussetzungen und Überlegungen in Application Load Balancing auf Amazon EKS und Network Load Balancing in Amazon EKS zu überprüfen. In diesen Themen stellen Sie eine Beispiel-App bereit, die einen AWS Load Balancer enthält.

Installieren Sie den Controller

- Erfahren Sie, wie das gehtthe section called “Mit Helm installieren”.Gehen Sie wie folgt vor, wenn Sie Amazon EKS noch nicht kennen. Bei diesem Verfahren wird Helm, ein Paketmanager für und `eksctl`zur Vereinfachung der Installation von LBCKubernetes, verwendet.
- Alternativ. the section called “Installieren Sie es mit Manifesten” Dieses Verfahren eignet sich für erweiterte Clusterkonfigurationen. Dazu gehören Cluster mit eingeschränktem Netzwerkzugriff auf öffentliche Container-Registries.

Migrieren Sie von veralteten Controller-Versionen

- Wenn Sie veraltete Versionen von AWS Load Balancer Controller installiert haben, erfahren Sie hier, wie das geht. the section called “Migrieren Sie von einem veralteten Controller”
- Veraltete Versionen können nicht aktualisiert werden. Sie müssen entfernt und eine aktuelle Version der AWS Load Balancer Controller installiert werden.
- Zu den veralteten Versionen gehören:
  - AWS ALB Ingress Controller für Kubernetes („Ingress Controller“), ein Vorgänger des. AWS Load Balancer Controller
  - Jede Version `0.1.`*x* des AWS Load Balancer Controller

Legacy-Cloud-Anbieter

Kubernetesbeinhaltet einen Legacy-Cloud-Anbieter für AWS. Der ältere Cloud-Anbieter ist in der Lage, AWS Load Balancer bereitzustellen, ähnlich wie der. AWS Load Balancer Controller Der

Legacy-Cloud-Anbieter erstellt Classic Load Balancers. Wenn Sie den nicht installierenAWS Load Balancer Controller, Kubernetes wird standardmäßig der Legacy-Cloud-Anbieter verwendet. Sie sollten den alten Cloud-Anbieter installieren AWS Load Balancer Controller und vermeiden, ihn zu verwenden.

> **Important**
>
> In den Versionen 2.5 und neuer AWS Load Balancer Controller wird der mit dem zum Standardcontroller für Kubernetes Dienstressourcen `type: LoadBalancer` und erstellt für jeden Dienst einen AWS Network Load Balancer (NLB). Dies geschieht durch einen mutierenden Webhook für Services, der das Feld `spec.loadBalancerClass` auf `service.k8s.aws/nlb` für neue Services von `type: LoadBalancer` setzt. Sie können dieses Feature deaktivieren und wieder den alten Cloud Provider als Standard-Controller verwenden, indem Sie den Wert für den Helm-Chart `enableServiceMutatorWebhook` auf `false` setzen. Der Cluster stellt keine neuen Classic Load Balancer für Ihre Services bereit, es sei denn, Sie schalten dieses Feature aus. Bestehende Classic Load Balancer werden weiterhin funktionieren.

Installieren Sie den AWS Load Balancer Controller benutzenden Helm

In diesem Thema wird beschrieben, wie Sie das AWS Load Balancer Controller mithilfe von Helm, einem Paketmanager fürKubernetes, und installieren`eksctl`. Der Controller wird mit Standardoptionen installiert. Weitere Informationen zum Controller, einschließlich Einzelheiten zur Konfiguration mit Anmerkungen, finden Sie in der AWS Load Balancer ControllerDokumentation unterGitHub.

Ersetzen Sie in den folgenden Schritten das *`example values`* durch Ihre eigenen Werte.

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Werkzeuge und Ressourcen installieren und konfigurieren, die Sie zum Erstellen und Verwalten eines Amazon-EKS-Clusters benötigen.

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.

- Ein vorhandener AWS Identity and Access Management (IAM) OpenID Connect (OIDC) -Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Stellen Sie sicher, dass das Ihre Amazon VPC CNI plugin for Kubernetes-, `kube-proxy` und CoreDNS-Add-ons mindestens die unter Servicekonto-Tokens genannten Versionen haben.
- Vertrautheit mit AWS Elastic Load Balancing. Weitere Informationen finden Sie im Elastic Load Balancing-Benutzerhandbuch.
- Vertrautheit mit Kubernetes- Service und Ingress -Ressourcen.

- Helm ist lokal installiert.

Schritt 1: Erstellen Sie eine IAM-Rolle mit **`eksctl`**

> **Note**
>
> Sie müssen nur eine IAM-Rolle für die AWS Load Balancer Controller Rolle pro AWS Konto erstellen. Prüfen Sie, ob sie in der IAM-Konsole `AmazonEKSLoadBalancerControllerRole` vorhanden ist. Wenn diese Rolle existiert, fahren Sie mit the section called “Schritt 2: Installieren AWS Load Balancer Controller” fort.

Erstellen Sie eine IAM-Richtlinie.

1. Laden Sie eine IAM-Richtlinie für den AWS Load Balancer Controller herunter, die es ihm ermöglicht, in Ihrem Namen Aufrufe an AWS -APIs zu tätigen.

   AWS

   ```
   $ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/install/iam_policy.json
   ```

   AWS GovCloud (US)

   ```
   $ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/install/iam_policy_us-gov.json
   ```

```
$ mv iam_policy_us-gov.json iam_policy.json
```

2. Erstellen Sie eine IAM-Richtlinie mit der im vorherigen Schritt heruntergeladenen Richtlinie.

```
$ aws iam create-policy \
    --policy-name AWSLoadBalancerControllerIAMPolicy \
    --policy-document file://iam_policy.json
```

> **Note**
>
> Wenn Sie sich die Richtlinie in ansehen AWS Management Console, zeigt die Konsole Warnungen für den ELB-Dienst an, aber nicht für den ELB v2-Dienst. Dies liegt daran, dass einige der Maßnahmen in der Richtlinie für ELB v2, aber nicht für ELB gelten. Sie können diese Warnungen für ELB ignorieren.

Erstellen Sie eine IAM-Rolle mit **`eksctl`**

- Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters, *`111122223333`* durch Ihre Konto-ID und führen Sie dann den Befehl aus. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (USA-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch. `arn:aws:` `arn:aws-us-gov:`

```
$ eksctl create iamserviceaccount \
  --cluster=my-cluster \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --role-name AmazonEKSLoadBalancerControllerRole \
  --attach-policy-
arn=arn:aws:iam::111122223333:policy/AWSLoadBalancerControllerIAMPolicy \
  --approve
```

Schritt 2: Installieren AWS Load Balancer Controller

Installieren Sie AWS Load Balancer Controller mit Helm V3

1. Fügen Sie das `eks-charts` Helm-Chart-Repository hinzu. AWS unterhält dieses Repository auf GitHub.

```
$ helm repo add eks https://aws.github.io/eks-charts
```

2. Aktualisieren Sie Ihr lokales Repository, um sicherzustellen, dass Sie über die neuesten Charts verfügen.

```
$ helm repo update eks
```

3. Installieren Sie die AWS Load Balancer Controller.

   Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters. Im folgenden Befehl ist `aws-load-balancer-controller` das Kubernetes-Servicekonto, das Sie in einem vorherigen Schritt erstellt haben.

   Weitere Informationen zur Konfiguration des Helmdiagramms finden Sie `values.yaml` unter GitHub.

```
$ helm install aws-load-balancer-controller eks/aws-load-balancer-controller \
  -n kube-system \
  --set clusterName=my-cluster \
  --set serviceAccount.create=false \
  --set serviceAccount.name=aws-load-balancer-controller
```

   a. Wenn Sie den Controller auf Amazon-EC2-Knoten bereitstellen, die eingeschränkten Zugriff auf den Instance Metadata Service (IMDS) von Amazon EC2 haben, oder wenn die Bereitstellung auf Fargate erfolgt, fügen Sie dem folgenden `helm`-Befehl die folgenden Flags hinzu:

      - **--set region=*region-code***
      - **--set vpcId=*vpc-xxxxxxxx***

   b. Verwenden Sie den folgenden Befehl, um die verfügbaren Versionen von Helm Chart und Load Balancer Controller anzuzeigen:

```
helm search repo eks/aws-load-balancer-controller --versions
```

> **Important**
>
> Das bereitgestellte Diagramm erhält keine automatischen Sicherheitsupdates. Sie müssen manuell auf ein neueres Diagramm aktualisieren, wenn es verfügbar wird. Wechseln Sie beim Upgrade *`install`* zum **`upgrade`** vorherigen Befehl.
> Der `helm install` Befehl installiert automatisch die benutzerdefinierten Ressourcendefinitionen (CRDs) für den Controller. Der `helm upgrade` Befehl tut dies nicht. Wenn Sie verwendenhelm upgrade,, müssen Sie das manuell installierenCRDs. Führen Sie den folgenden Befehl aus, um das zu installierenCRDs:
>
> ```
> wget https://raw.githubusercontent.com/aws/eks-charts/master/stable/aws-load-balancer-controller/crds/crds.yaml
> kubectl apply -f crds.yaml
> ```

Schritt 3: Stellen Sie sicher, dass der Controller installiert ist

1. Stellen Sie sicher, dass der Controller installiert ist.

   ```
   $ kubectl get deployment -n kube-system aws-load-balancer-controller
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
   aws-load-balancer-controller   2/2     2            2           84s
   ```

   Sie erhalten die vorherige Ausgabe, wenn Sie mit Helm bereitgestellt haben. Wenn Sie mit dem Kubernetes-Manifest bereitgestellt haben, haben Sie nur ein Replikat.

2. Bevor Sie den Controller zur Bereitstellung von AWS Ressourcen verwenden können, muss Ihr Cluster bestimmte Anforderungen erfüllen. Weitere Informationen finden Sie unter Application Load Balancing auf Amazon EKS und Network Load Balancing in Amazon EKS.

Installieren Sie das AWS Load Balancer Controller Add-on mithilfe von Kubernetes Manifesten

In diesem Thema wird beschrieben, wie Sie den Controller installieren, indem Sie Kubernetes Manifeste herunterladen und anwenden. Sie können die vollständige Dokumentation für den Controller auf GitHub anzeigen.

Ersetzen Sie in den folgenden Schritten das *example values* durch Ihre eigenen Werte.

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Werkzeuge und Ressourcen installieren und konfigurieren, die Sie zum Erstellen und Verwalten eines Amazon-EKS-Clusters benötigen.

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.
- Ein vorhandener AWS Identity and Access Management (IAM) OpenID Connect (OIDC) -Anbieter für Ihren Cluster. Informationen zum Feststellen, ob Sie bereits über einen verfügen oder einen erstellen müssen, finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
- Stellen Sie sicher, dass das Ihre Amazon VPC CNI plugin for Kubernetes-, `kube-proxy` und CoreDNS-Add-ons mindestens die unter Servicekonto-Tokens genannten Versionen haben.
- Vertrautheit mit AWS Elastic Load Balancing Weitere Informationen finden Sie im Elastic Load Balancing-Benutzerhandbuch.
- Vertrautheit mit Kubernetes- Service und Ingress -Ressourcen.

Schritt 1: IAM konfigurieren

> **Note**
>
> Sie müssen nur eine IAM-Rolle für die AWS Load Balancer Controller Rolle pro AWS Konto erstellen. Prüfen Sie, ob sie in der IAM-Konsole `AmazonEKSLoadBalancerControllerRole` vorhanden ist. Wenn diese Rolle existiert, fahren Sie mit the section called “Schritt 2: Installieren `cert-manager`” fort.

Erstellen Sie eine IAM-Richtlinie.

1. Laden Sie eine IAM-Richtlinie für den AWS Load Balancer Controller herunter, die es ihm ermöglicht, in Ihrem Namen Aufrufe an AWS -APIs zu tätigen.

   AWS

   ```
   $ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/install/iam_policy.json
   ```

   AWS GovCloud (US)

   ```
   $ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/install/iam_policy_us-gov.json
   ```

   ```
   $ mv iam_policy_us-gov.json iam_policy.json
   ```

2. Erstellen Sie eine IAM-Richtlinie mit der im vorherigen Schritt heruntergeladenen Richtlinie.

   ```
   $ aws iam create-policy \
       --policy-name AWSLoadBalancerControllerIAMPolicy \
       --policy-document file://iam_policy.json
   ```

   > **Note**
   >
   > Wenn Sie sich die Richtlinie in ansehen AWS Management Console, zeigt die Konsole Warnungen für den ELB-Dienst an, aber nicht für den ELB v2-Dienst. Dies liegt daran, dass einige der Maßnahmen in der Richtlinie für ELB v2, aber nicht für ELB gelten. Sie können diese Warnungen für ELB ignorieren.

eksctl

Erstellen Sie eine IAM-Rolle mit **`eksctl`**

- Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters, *`111122223333`* durch Ihre Konto-ID und führen Sie dann den Befehl aus. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (USA-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch. `arn:aws:` `arn:aws-us-gov:`

```
$ eksctl create iamserviceaccount \
  --cluster=my-cluster \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --role-name AmazonEKSLoadBalancerControllerRole \
  --attach-policy-
arn=arn:aws:iam::111122223333:policy/AWSLoadBalancerControllerIAMPolicy \
  --approve
```

AWS CLI and kubectl

Erstellen Sie eine IAM-Rolle mit dem und AWS CLI**kubectl**

1. Rufen Sie die OIDC-Anbieter-ID Ihres Clusters ab und speichern Sie sie in einer Variable.

```
oidc_id=$(aws eks describe-cluster --name my-cluster --query
 "cluster.identity.oidc.issuer" --output text | cut -d '/' -f 5)
```

2. Bestimmen Sie, ob ein IAM-OIDC-Anbieter mit der ID Ihres Clusters bereits in Ihrem Konto vorhanden ist. Sie müssen sowohl für den Cluster als auch für IAM OIDC konfiguriert sein.

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

   Wenn eine Ausgabe erfolgt, verfügen Sie bereits über einen IAM-OIDC-Anbieter für Ihren Cluster. Wenn keine Ausgabe erfolgt, müssen Sie einen IAM-OIDC-Anbieter für Ihren Cluster erstellen. Weitere Informationen finden Sie unter Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.

3. Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie *111122223333* durch Ihre Konto-ID. *region-code*Ersetzen Sie es durch AWS-Region das, in dem sich Ihr Cluster befindet. Ersetzen Sie *EXAMPLED539D4633E53DE1B71EXAMPLE* mit den im vorherigen Schritt zurückgegebenen Ausgaben. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (US-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch`arn:aws:`. `arn:aws-us-gov:` Nachdem Sie den Text ersetzt haben, führen Sie den geänderten Befehl aus, um die Datei `load-balancer-role-trust-policy.json` zu erstellen.

```
cat >load-balancer-role-trust-policy.json <<EOF
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com",
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:aws-load-balancer-controller"
                }
            }
        }
    ]
}
EOF
```

4. Erstellen Sie die IAM-Rolle.

```
aws iam create-role \
  --role-name AmazonEKSLoadBalancerControllerRole \
  --assume-role-policy-document file://"load-balancer-role-trust-policy.json"
```

5. Hängen Sie die erforderliche von Amazon EKS verwaltete IAM-Richtlinie an die IAM-Rolle an. Ersetzen Sie *111122223333* durch Ihre Konto-ID.

```
aws iam attach-role-policy \
  --policy-arn
 arn:aws:iam::111122223333:policy/AWSLoadBalancerControllerIAMPolicy \
  --role-name AmazonEKSLoadBalancerControllerRole
```

6. Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (USA-Ost) oder AWS GovCloud (US-West) befindet AWS-Regionen, ersetzen Sie ihn durch. `arn:aws: arn:aws-`

`us-gov`: Nachdem Sie den Text ersetzt haben, führen Sie den geänderten Befehl aus, um die Datei `aws-load-balancer-controller-service-account.yaml` zu erstellen.

```
cat >aws-load-balancer-controller-service-account.yaml <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/name: aws-load-balancer-controller
  name: aws-load-balancer-controller
  namespace: kube-system
  annotations:
    eks.amazonaws.com/role-arn:
 arn:aws:iam::111122223333:role/AmazonEKSLoadBalancerControllerRole
EOF
```

7. Erstellen Sie das Kubernetes-Servicekonto in Ihrem Cluster. Das Kubernetes-Servicekonto namens `aws-load-balancer-controller` wird mit der von Ihnen erstellten IAM-Rolle namens *AmazonEKSLoadBalancerControllerRole* annotiert.

```
$ kubectl apply -f aws-load-balancer-controller-service-account.yaml
```

Schritt 2: Installieren **`cert-manager`**

Installieren Sie `cert-manager` mit einer der beiden folgenden Methoden, um die Zertifikatskonfiguration in die Webhooks einzufügen. Weitere Informationen dazu finden Sie unter Jetzt starten in der *`cert-manager`*-Dokumentation.

Wir empfehlen, für die Installation die `quay.io` Container-Registry zu verwendencert-manager. Wenn Ihre Knoten keinen Zugriff auf die `quay.io` Container-Registry haben, installieren Sie die Installation `cert-manager` mithilfe von Amazon ECR (siehe unten).

Quay.io

Mit Quay.io installieren **`cert-manager`**

- Wenn die Knoten Zugriff auf die `quay.io`-Container-Registry haben, installieren Sie `cert-manager`, um die Zertifikatskonfiguration in die Webhooks einzufügen.

```
$ kubectl apply \
    --validate=false \
    -f https://github.com/jetstack/cert-manager/releases/download/v1.13.5/cert-manager.yaml
```

Amazon ECR

Installation **`cert-manager`** mit Amazon ECR

1. Installieren Sie `cert-manager` mit einer der beiden folgenden Methoden, um die Zertifikatskonfiguration in die Webhooks einzufügen. Weitere Informationen dazu finden Sie unter Jetzt starten in der *`cert-manager`*-Dokumentation.
2. Laden Sie das Manifest herunter.

```
curl -Lo cert-manager.yaml https://github.com/jetstack/cert-manager/releases/download/v1.13.5/cert-manager.yaml
```

3. Rufen Sie die folgenden Images ab und verschieben Sie sie in ein Repository, auf das die Knoten Zugriff haben. Weitere Informationen zum Abrufen, Markieren und Verschieben der Images in ein eigenes Repository finden Sie unter Kopieren eines Container-Images von einem Repository in ein anderes.

```
quay.io/jetstack/cert-manager-cainjector:v1.13.5
quay.io/jetstack/cert-manager-controller:v1.13.5
quay.io/jetstack/cert-manager-webhook:v1.13.5
```

4. Ersetzen Sie `quay.io` im Manifest für die drei Images durch den Namen Ihrer eigenen Registrierung. Beim folgenden Befehl wird davon ausgegangen, dass der Name Ihres privaten Repositorys mit dem des Quell-Repositorys übereinstimmt. Ersetzen Sie *`111122223333.dkr.ecr.region-code.amazonaws.com`* durch Ihr privates Register.

```
$ sed -i.bak -e 's|quay.io|111122223333.dkr.ecr.region-code.amazonaws.com|' ./cert-manager.yaml
```

5. Das Manifest anwenden.

```
$ kubectl apply \
    --validate=false \
```

```
    -f ./cert-manager.yaml
```

Schritt 3: Installieren AWS Load Balancer Controller

Installieren Sie AWS Load Balancer Controller mithilfe eines Kubernetes Manifests

1. Laden Sie die Controller-Spezifikation herunter. Weitere Informationen zum Controller finden Sie in der Dokumentation auf GitHub.

```
curl -Lo v2_7_2_full.yaml https://github.com/kubernetes-sigs/aws-load-balancer-controller/releases/download/v2.7.2/v2_7_2_full.yaml
```

2. Nehmen Sie die folgenden Änderungen an der Datei vor.

   a. Wenn Sie die Datei `v2_7_2_full.yaml` heruntergeladen haben, führen Sie den folgenden Befehl aus, um den Abschnitt `ServiceAccount` im Manifest zu entfernen. Wenn Sie diesen Abschnitt nicht entfernen, wird die erforderliche Anmerkung überschrieben, die Sie in einem vorherigen Schritt für das Servicekonto erstellt haben. Außerdem wird durch das Entfernen dieses Abschnitts das in einem vorherigen Schritt erstellte Servicekonto beibehalten, wenn Sie den Controller löschen.

   ```
   $ sed -i.bak -e '596,604d' ./v2_7_2_full.yaml
   ```

   Wenn Sie eine andere Dateiversion heruntergeladen haben, öffnen Sie die Datei in einem Editor und entfernen Sie die folgenden Zeilen.

   ```
   apiVersion: v1
   kind: ServiceAccount
   metadata:
     labels:
       app.kubernetes.io/component: controller
       app.kubernetes.io/name: aws-load-balancer-controller
     name: aws-load-balancer-controller
     namespace: kube-system
   ---
   ```

   b. Ersetzen Sie `your-cluster-name` im `Deployment spec`-Abschnitt der Datei durch den Namen Ihres Clusters, indem Sie *`my-cluster`* durch den Namen Ihres Cluster ersetzen.

```
$ sed -i.bak -e 's|your-cluster-name|my-cluster|' ./v2_7_2_full.yaml
```

c. Wenn Ihre Knoten keinen Zugriff auf die Amazon-ECR-Image-Repositorys von Amazon EKS haben, müssen Sie das folgende Image abrufen und in ein Repository verschieben, auf das die Knoten Zugriff haben. Weitere Informationen zum Abrufen, Markieren und Verschieben von Images in ein eigenes Repository finden Sie unter Kopieren eines Container-Images von einem Repository in ein anderes.

```
public.ecr.aws/eks/aws-load-balancer-controller:v2.7.2
```

Fügen Sie dem Manifest den Namen Ihrer Registrierung hinzu. Beim folgenden Befehl wird davon ausgegangen, dass der Name Ihres privaten Repositorys mit dem des Quell-Repositorys übereinstimmt. Dessen Name wird der Datei hinzugefügt. Ersetzen Sie *111122223333.dkr.ecr.region-code.amazonaws.com* durch Ihr Register. In dieser Zeile wird davon ausgegangen, dass Sie Ihr privates Repository genauso benannt haben wie das Quell-Repository. Andernfalls ändern Sie den Text `eks/aws-load-balancer-controller` nach den Namen Ihrer privaten Registrierung in den Namen Ihres Repositorys.

```
$ sed -i.bak -e 's|public.ecr.aws/eks/aws-load-balancer-
controller|111122223333.dkr.ecr.region-code.amazonaws.com/eks/aws-load-
balancer-controller|' ./v2_7_2_full.yaml
```

d. (Nur für Fargate oder Restricted IMDS erforderlich)

Wenn Sie den Controller auf Amazon-EC2-Knoten bereitstellen, die eingeschränkten Zugriff auf den Amazon EC2 Instance Metadata Service (IMDS) haben, oder wenn Sie Fargate bereitstellen, fügen Sie **`following parameters`** unter `- args:` hinzu.

```
[...]
spec:
      containers:
        - args:
            - --cluster-name=your-cluster-name
            - --ingress-class=alb
            - --aws-vpc-id=vpc-xxxxxxxx
            - --aws-region=region-code
```

```
[...]
```

3. Wenden Sie die Datei an.

```
$ kubectl apply -f v2_7_2_full.yaml
```

4. Laden Sie das `IngressClass`- und `IngressClassParams`-Manifest in Ihren Cluster herunter.

```
$ curl -Lo v2_7_2_ingclass.yaml https://github.com/kubernetes-sigs/aws-load-balancer-controller/releases/download/v2.7.2/v2_7_2_ingclass.yaml
```

5. Wenden Sie das Manifest auf Ihren Cluster an.

```
$ kubectl apply -f v2_7_2_ingclass.yaml
```

Schritt 4: Stellen Sie sicher, dass der Controller installiert ist

1. Stellen Sie sicher, dass der Controller installiert ist.

```
$ kubectl get deployment -n kube-system aws-load-balancer-controller
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2            2           84s
```

Sie erhalten die vorherige Ausgabe, wenn Sie mit Helm bereitgestellt haben. Wenn Sie mit dem Kubernetes-Manifest bereitgestellt haben, haben Sie nur ein Replikat.

2. Bevor Sie den Controller zur Bereitstellung von AWS Ressourcen verwenden können, muss Ihr Cluster bestimmte Anforderungen erfüllen. Weitere Informationen finden Sie unter Application Load Balancing auf Amazon EKS und Network Load Balancing in Amazon EKS.

Von einem veralteten Controller migrieren

In diesem Thema wird beschrieben, wie Sie von veralteten Controller-Versionen migrieren. Insbesondere wird beschrieben, wie veraltete Versionen von entfernt werden. AWS Load Balancer Controller

- Veraltete Versionen können nicht aktualisiert werden. Sie müssen entfernt und eine aktuelle Version des LBC installiert werden.
- Zu den veralteten Versionen gehören:
  - AWS ALB Ingress Controller für Kubernetes („Ingress Controller“), ein Vorgänger des. AWS Load Balancer Controller
  - Jede Version `0.1.x` des AWS Load Balancer Controller

Entfernen Sie die veraltete Controller-Version

> **Note**
>
> Möglicherweise haben Sie die veraltete Version mit Helm oder manuell mit Manifesten installiert. Kubernetes Führen Sie den Vorgang mit dem Tool aus, mit dem Sie ihn ursprünglich installiert haben.

Entfernen Sie den Ingress Controller mit Helm

1. Wenn Sie das `incubator/aws-alb-ingress-controller`-Helm-Chart installiert haben, deinstallieren Sie es.

   ```
   $ helm delete aws-alb-ingress-controller -n kube-system
   ```

2. Wenn Sie Version `0.1.x` des `eks-charts/aws-load-balancer-controller`-Charts installiert haben, deinstallieren Sie es. Das Upgrade von `0.1.x` auf Version `1.0.0` funktioniert nicht, da die Webhook-API-Version inkompatibel ist.

   ```
   $ helm delete aws-load-balancer-controller -n kube-system
   ```

Entfernen Sie den Ingress Controller mithilfe des Manifests Kubernetes

1. Überprüfen Sie, ob der Controller derzeit installiert ist.

   ```
   $ kubectl get deployment -n kube-system alb-ingress-controller
   ```

   Dies ist die Ausgabe, falls der Controller nicht installiert ist.

Fehler vom Server (NotFound): deployments.apps "" wurde nicht gefunden alb-ingress-controller

Dies ist die Ausgabe, falls der Controller installiert ist.

```
NAME                      READY UP-TO-DATE AVAILABLE AGE
alb-ingress-controller 1/1   1          1         122d
```

2. Geben Sie die folgenden Befehle ein, um den Controller zu entfernen.

```
$ kubectl delete -f https://raw.githubusercontent.com/kubernetes-sigs/aws-alb-
ingress-controller/v1.1.8/docs/examples/alb-ingress-controller.yaml
kubectl delete -f https://raw.githubusercontent.com/kubernetes-sigs/aws-alb-
ingress-controller/v1.1.8/docs/examples/rbac-role.yaml
```

Migrieren zu AWS Load Balancer Controller

Um vom ALB Ingress Controller für zum Kubernetes zu migrieren, müssen Sie: AWS Load Balancer Controller

1. Entfernen Sie den ALB Ingress Controller (siehe oben).
2. Installieren Sie den. AWS Load Balancer Controller
3. Fügen Sie der vom LBC verwendeten IAM-Rolle eine zusätzliche Richtlinie hinzu. Diese Richtlinie ermöglicht es dem LBC, Ressourcen zu verwalten, die vom ALB Ingress Controller für erstellt wurden. Kubernetes

Fügen Sie der IAM-Rolle eine Migrationsrichtlinie hinzu. AWS Load Balancer Controller

1. Laden Sie die IAM-Richtlinie herunter. Diese Richtlinie ermöglicht es dem LBC, Ressourcen zu verwalten, die vom ALB Ingress Controller für erstellt wurden. Kubernetes Sie können auch die Richtlinie anzeigen.

```
$ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-
controller/v2.7.2/docs/install/iam_policy_v1_to_v2_additional.json
```

2. Wenn sich Ihr Cluster in den Ländern AWS GovCloud (USA-Ost) oder AWS GovCloud (US-West) AWS-Regionen befindet, ersetzen Sie ihn durch. `arn:aws:` `arn:aws-us-gov:` durch. `arn:aws:`

```
$ sed -i.bak -e 's|arn:aws:|arn:aws-us-gov:|' iam_policy_v1_to_v2_additional.json
```

3. Erstellen Sie die IAM-Richtlinie und notieren Sie den zurückgegebenen ARN.

```
$ aws iam create-policy \
  --policy-name AWSLoadBalancerControllerAdditionalIAMPolicy \
  --policy-document file://iam_policy_v1_to_v2_additional.json
```

4. Hängen Sie die IAM-Richtlinie an die IAM-Rolle an, die vom LBC verwendet wird. *your-role-name*Ersetzen Sie durch den Namen der Rolle, z. B. `AmazonEKSLoadBalancerControllerRole`

   Wenn Sie die Rolle mit`eksctl`, erstellt haben, öffnen Sie die AWS CloudFormation Konsole und wählen Sie den Stack eksctl- ***my-cluster*** - - addon-iamserviceaccount-kube-system aus, um den erstellten Rollennamen zu finden. aws-load-balancer-controller Wählen Sie die Registerkarte für Resources (Ressourcen). Der Rollenname befindet sich in der Spalte Physische ID. Wenn sich Ihr Cluster in den Regionen AWS GovCloud (USA-Ost) oder AWS GovCloud (US-West) AWS-Regionen befindet, ersetzen Sie ihn durch. `arn:aws: arn:aws-us-gov:`

```
$ aws iam attach-role-policy \
  --role-name your-role-name \
  --policy-arn
 arn:aws:iam::111122223333:policy/AWSLoadBalancerControllerAdditionalIAMPolicy
```

Arbeiten mit dem CoreDNS-Amazon-EKS-Add-on

CoreDNS ist ein flexibler, erweiterbarer DNS-Server, der als Kubernetes-Cluster-DNS dienen kann. Wenn Sie einen Amazon-EKS-Cluster mit mindestens einem Knoten starten, werden standardmäßig zwei Replikate des CoreDNS-Image bereitgestellt, unabhängig von der Anzahl der in Ihrem Cluster bereitgestellten Knoten. Die CoreDNS-Pods bieten die Namensauflösung für alle Pods im Cluster. Die CoreDNS-Pods können auf Fargate-Knoten bereitgestellt werden, wenn Ihr Cluster ein AWS Fargate Profil mit einem Namespace enthält, der dem Namespace für das CoreDNS-`deployment` entspricht. Weitere Informationen zu CoreDNS finden Sie unter Verwenden von CoreDNS für die Serviceerkennung in der Kubernetes-Dokumenation.

In der folgenden Tabelle finden Sie die aktuelle Version des Kubernetes-Add-ons, die für jede Amazon-EKS-Cluster-Version verfügbar ist.

Kubernetes-Version	1.30	1.29	1.28	1.27	1.26	1.25	1.24	1.23
	v1.11.1-eksbuild	v1.11.1-eksbuild	v1.10.1-eksbuild.1	v1.10.1-eksbuild.1	v1.9.3-eksbuild.	v1.9.3-eksbuild.	v1.9.3-eksbuild.	v1.8.7-eksbuild.10

> **Important**
>
> Wenn Sie dieses Add-on selbst verwalten, stimmen die Versionen in der Tabelle möglicherweise nicht mit den verfügbaren selbstverwalteten Versionen überein. Weitere Hinweise zur Aktualisierung des selbstverwalteten Typs dieses Add-ons finden Sie unter Aktualisieren des selbstverwalteten -Add-ons.

Wichtige Überlegungen zum CoreDNS-Upgrade

- Um die Stabilität und Verfügbarkeit der CoreDNS Deployment zu verbessern, werden die Versionen `v1.9.3-eksbuild.6` und neuer und `v1.10.1-eksbuild.3` mit einem `PodDisruptionBudget` bereitgestellt. Wenn Sie ein vorhandenes `PodDisruptionBudget` bereitgestellt haben, schlägt Ihr Upgrade auf diese Versionen möglicherweise fehl. Schlägt das Upgrade fehl, sollte das Problem durch Ausführen einer der folgenden Aufgaben gelöst werden:
  - Wenn Sie das Upgrade des Amazon-EKS-Add-ons durchführen, entscheiden Sie sich dafür, die vorhandenen Einstellungen als Konfliktlösungsoption zu überschreiben. Wenn Sie weitere benutzerdefinierte Einstellungen für die Deployment vorgenommen haben, stellen Sie sicher, dass Sie Ihre Einstellungen vor dem Upgrade sichern, damit Sie Ihre anderen benutzerdefinierten Einstellungen nach dem Upgrade erneut anwenden können.
  - Entfernen Sie Ihr vorhandenes `PodDisruptionBudget` und versuchen Sie das Upgrade erneut.
- In EKS-Add-On-Versionen `v1.9.3-eksbuild.3` und höher sowie `v1.10.1-eksbuild.6` und höher legt die CoreDNS-Deployment die `readinessProbe` für die Verwendung des `/ready` Endpunkts fest. Dieser Endpunkt ist in der `Corefile`-Konfigurationsdatei für CoreDNS aktiviert.

Wenn Sie eine benutzerdefinierte `Corefile` verwenden, müssen Sie das `ready` Plugin zur Konfiguration hinzufügen, sodass der `/ready` Endpunkt in CoreDNS aktiv ist, damit er von der Probe verwendet werden kann.

- In EKS-Add-On-Versionen `v1.9.3-eksbuild.7` und höher sowie `v1.10.1-eksbuild.4` und höher können Sie das `PodDisruptionBudget` ändern. Sie können das Add-on bearbeiten und diese Einstellungen in den optionalen Konfigurationseinstellungen mithilfe der Felder im folgenden Beispiel ändern. Dieses Beispiel zeigt das Standard-`PodDisruptionBudget`.

```
{
    "podDisruptionBudget": {
        "enabled": true,
        "maxUnavailable": 1
        }
}
```

Sie können `maxUnavailable` oder `minAvailable` festlegen, aber Sie können nicht beide gleichzeitig in einem einzelnen `PodDisruptionBudget` festlegen. Weitere Informationen zu `PodDisruptionBudgets` finden Sie unter Angeben eines `PodDisruptionBudget` in der Kubernetes-Dokumentation.

Beachten Sie, dass, wenn Sie `enabled` auf `false` festlegen, das `PodDisruptionBudget` nicht entfernt wird. Nachdem Sie dieses Feld auf `false` gesetzt haben, müssen Sie das `PodDisruptionBudget`-Objekt löschen. Ähnlich verhält es sich, wenn Sie das Add-on bearbeiten, um nach dem Upgrade auf eine Version mit einem `PodDisruptionBudget` eine ältere Version des Add-ons zu verwenden (das Add-On herunterstufen). Das `PodDisruptionBudget` wird nicht entfernt. Führen Sie den folgenden Befehl aus, um das `PodDisruptionBudget` zu löschen:

```
kubectl delete poddisruptionbudget coredns -n kube-system
```

- Ändern Sie in EKS-Add-On-Versionen `v1.10.1-eksbuild.5` und höher die Standardtoleranz von `node-role.kubernetes.io/master:NoSchedule` auf, `node-role.kubernetes.io/control-plane:NoSchedule` um KEP 2067 zu entsprechen. Weitere Informationen über KEP 2067 finden Sie unter KEP-2067: Rename the kubeadm "master" label and taint in den Kubernetes Enhancement Proposals (KEPs) auf GitHub.

In EKS-Add-On-Versionen `v1.8.7-eksbuild.8` `v1.9.3-eksbuild.9` und späteren Versionen sind beide Toleranzen so eingestellt, dass sie mit jeder Version kompatibel sind. Kubernetes

- In EKS-Add-On-Versionen `v1.9.3-eksbuild.11` `v1.10.1-eksbuild.7` und höher CoreDNS Deployment legt der einen Standardwert für `topologySpreadConstraints` fest. Der Standardwert stellt sicher, dass sie auf die Availability Zones verteilt CoreDNS Pods sind, wenn Knoten in mehreren Availability Zones verfügbar sind. Sie können einen benutzerdefinierten Wert festlegen, der anstelle des Standardwerts verwendet wird. Der Standardwert lautet wie folgt:

```
topologySpreadConstraints:
  - maxSkew: 1
    topologyKey: topology.kubernetes.io/zone
    whenUnsatisfiable: ScheduleAnyway
    labelSelector:
      matchLabels:
        k8s-app: kube-dns
```

CoreDNSÜberlegungen zum **`1.11`** Upgrade

- In EKS-Add-On-Versionen `v1.11.1-eksbuild.4` und höher basiert das Container-Image auf einem minimalen Basis-Image, das von Amazon EKS Distro verwaltet wird. Es enthält nur minimale Pakete und keine Shells. Weitere Informationen finden Sie unter Amazon-EKS-Distro. Die Verwendung und Fehlerbehebung des CoreDNS Images bleiben unverändert.

Add-on vom Typ Amazon EKS erstellen

Erstellen Sie das Add-on vom Typ Amazon EKS. Check

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.

1. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist.

```
kubectl describe deployment coredns --namespace kube-system | grep coredns: | cut -
d : -f 3
```

Eine Beispielausgabe sieht wie folgt aus.

```
v1.10.1-eksbuild.11
```

2. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist. Je nachdem, mit welchem Tool Sie Ihr Cluster erstellt haben, ist der Add-on vom Typ Amazon EKS möglicherweise derzeit nicht auf Ihrem Cluster installiert. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
 addon.addonVersion --output text
```

Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Führen Sie die verbleibenden Schritte dieses Prozesses aus, um es zu installieren.

3. Speichern Sie die Konfiguration Ihres aktuell installierten Add-ons ab.

```
kubectl get deployment coredns -n kube-system -o yaml > aws-k8s-coredns-old.yaml
```

4. Erstellen Sie das Add-on mit dem AWS CLI. Wenn Sie das AWS Management Console oder verwenden möchten, um das Add-on `eksctl` zu erstellen, finden Sie weitere Informationen unter Erstellen eines Add-Ons und geben Sie `coredns` den Namen des Add-ons an. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie bei Bedarf die folgenden Änderungen am Befehl vor, und führen Sie dann den geänderten Befehl aus.

   - Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.
   - Ersetzen Sie *v1.11.1-eksbuild.9* durch die neueste Version, die in der Tabelle mit den neuesten Versionen für Ihre Cluster-Version aufgeführt ist.

```
aws eks create-addon --cluster-name my-cluster --addon-name coredns --addon-
version v1.11.1-eksbuild.9
```

Wenn Sie benutzerdefinierte Einstellungen auf Ihr aktuelles Add-on angewendet haben, die mit den Standardeinstellungen des Amazon EKS-Add-ons in Konflikt stehen, schlägt die Erstellung möglicherweise fehl. Wenn die Erstellung fehlschlägt, erhalten Sie eine Fehlermeldung, die

Sie bei der Problembehebung unterstützt. Alternativ können Sie den vorherigen Befehl mit **`--resolve-conflicts OVERWRITE`** ergänzen. Dadurch kann das Add-on alle vorhandenen benutzerdefinierten Einstellungen überschreiben. Sobald Sie das Add-on erstellt haben, können Sie es mit Ihren benutzerdefinierten Einstellungen aktualisieren.

5. Vergewissern Sie sich, dass die neueste Version des Add-ons für die Kubernetes-Version Ihres Clusters zu Ihrem Cluster hinzugefügt wurde. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
    addon.addonVersion --output text
   ```

   Es kann einige Sekunden dauern, bis die Erstellung des Add-ons abgeschlossen ist.

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.11.1-eksbuild.9
   ```

6. Wenn Sie benutzerdefinierte Einstellungen für Ihr ursprüngliches Add-on vorgenommen haben, bevor Sie das Add-on vom Typ Amazon EKS erstellt haben, verwenden Sie die Konfiguration, die Sie in einem vorherigen Schritt gespeichert haben, um das Add-on vom Typ Amazon EKS mit Ihren benutzerdefinierten Einstellungen zu aktualisieren.

Aktualisieren des Amazon-EKS-Add-ons

Erstellen Sie das Add-on vom Typ Amazon EKS. Wenn Sie das Add-on vom Typ Amazon EKS nicht zu Ihrem Cluster hinzugefügt haben, fügen Sie ihn entweder hinzu oder sehen Sie sich Aktualisieren des selbstverwalteten -Add-ons an, anstatt dieses Verfahren abzuschließen.

1. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist. Ersetzen Sie *`my-cluster`* mit Ihrem Clusternamen.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
    "addon.addonVersion" --output text
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.10.1-eksbuild.11
   ```

Wenn die zurückgegebene Version mit der Version für die Kubernetes-Version Ihres Clusters in der aktuellen Versionstabelle übereinstimmt, haben Sie die neueste Version bereits auf Ihrem Cluster installiert und müssen den Rest dieses Verfahrens nicht abschließen. Wenn Sie eine Fehlermeldung statt einer Versionsnummer in Ihrer Ausgabe erhalten, haben Sie den Add-on vom Typ Amazon EKS nicht auf Ihrem Cluster installiert. Sie müssen das Add-on erstellen, bevor Sie es mit diesem Verfahren aktualisieren können.

2. Speichern Sie die Konfiguration Ihres aktuell installierten Add-ons ab.

```
kubectl get deployment coredns -n kube-system -o yaml > aws-k8s-coredns-old.yaml
```

3. Aktualisieren Sie Ihr Add-on mit der AWS CLI. Wenn Sie das AWS Management Console oder verwenden möchten, um das Add-on zu aktualisieren, finden Sie weitere Informationen unter`eksctl`. Aktualisieren eines Add-Ons Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie bei Bedarf die folgenden Änderungen am Befehl vor, und führen Sie dann den geänderten Befehl aus.

   - Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.
   - Ersetzen Sie *v1.11.1-eksbuild.9* durch die neueste Version, die in der Tabelle mit den neuesten Versionen für Ihre Cluster-Version aufgeführt ist.
   - Die **--resolve-conflicts**-Option *PRESERVE* (Beibehalten) behält die vorhandenen Werte für das Add-on bei. Wenn Sie benutzerdefinierte Werte für Zusatzeinstellungen festgelegt haben und diese Option nicht verwenden, überschreibt Amazon EKS Ihre Werte mit seinen Standardwerten. Wenn Sie diese Option verwenden, empfehlen wir, dass Sie alle Feld- und Wertänderungen auf einem Nicht-Produktionscluster testen, bevor Sie das Add-on auf Ihrem Produktionscluster aktualisieren. Wenn Sie diesen Wert auf `OVERWRITE` ändern, werden alle Einstellungen auf die Amazon-EKS-Standardwerte geändert. Wenn Sie benutzerdefinierte Werte für Einstellungen festgelegt haben, werden diese möglicherweise mit den Amazon-EKS-Standardwerten überschrieben. Wenn Sie diesen Wert auf `none` ändern, ändert Amazon EKS den Wert der Einstellungen nicht, aber das Update schlägt möglicherweise fehl. Wenn das Update fehlschlägt, erhalten Sie eine Fehlermeldung, die Sie bei der Behebung des Konflikts unterstützt.
   - Wenn Sie eine Konfigurationseinstellung nicht aktualisieren, entfernen Sie **--configuration-values '{*"replicaCount":3*}'** aus dem Befehl. Wenn Sie eine Konfigurationseinstellung aktualisieren, ersetzen Sie *"replicaCount":3* durch die Einstellung, die Sie festlegen möchten. In diesem Beispiel ist die Anzahl der Replikate von CoreDNS auf 3 festgelegt. Der von Ihnen angegebene Wert muss für das

Konfigurationsschema gültig sein. Wenn Sie das Konfigurationsschema nicht kennen, führen Sie den Befehl aus und ersetzen Sie *v1.11.1-eksbuild.9* durch die Versionsnummer des Add-ons**aws eks describe-addon-configuration --addon-name coredns --addon-version *v1.11.1-eksbuild.9***, für das Sie die Konfiguration sehen möchten. Das Schema wird in der Ausgabe zurückgegeben. Wenn Sie bereits eine benutzerdefinierte Konfiguration haben, diese komplett entfernen und die Werte für alle Einstellungen auf die Amazon EKS-Standardwerte zurücksetzen möchten, entfernen Sie *"replicaCount":3* aus dem Befehl, sodass **{}** leer ist. Weitere Informationen zu CoreDNS-Einstellungen finden Sie in der Kubernetes-Dokumentation unter Customizing DNS Service (Anpassen des DNS-Service).

```
aws eks update-addon --cluster-name my-cluster --addon-name coredns --addon-
version v1.11.1-eksbuild.9 \
    --resolve-conflicts PRESERVE --configuration-values '{"replicaCount":3}'
```

Es kann einige Sekunden dauern, bis das Update abgeschlossen ist.

4. Vergewissern Sie sich, dass die Add-on-Version aktualisiert wurde. Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns
```

Es kann einige Sekunden dauern, bis das Update abgeschlossen ist.

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "addon": {
        "addonName": "coredns",
        "clusterName": "my-cluster",
        "status": "ACTIVE",
        "addonVersion": "v1.11.1-eksbuild.9",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:region:111122223333:addon/my-cluster/coredns/
d2c34f06-1111-2222-1eb0-24f64ce37fa4",
        "createdAt": "2023-03-01T16:41:32.442000+00:00",
        "modifiedAt": "2023-03-01T18:16:54.332000+00:00",
        "tags": {},
        "configurationValues": "{\"replicaCount\":3}"
    }
```

```
}
```

Aktualisieren des selbstverwalteten -Add-ons

> **Important**
>
> Wir empfehlen, den Amazon-EKS-Typ des Add-Ons zu Ihrem Cluster hinzuzufügen, anstatt den selbstverwalteten Typ des Add-Ons zu verwenden. Wenn Sie noch keine Erfahrung mit den Unterschieden zwischen den Typen haben, finden Sie weitere Informationen unter the section called “Amazon-EKS-Add-ons”. Weitere Informationen zum Hinzufügen eines Amazon-EKS-Add-ons zu Ihrem Cluster finden Sie unter the section called “Erstellen eines Add-Ons”. Wenn Sie das Amazon EKS-Add-on nicht verwenden können, empfehlen wir Ihnen, ein Problem mit der Begründung, warum Sie das nicht können, an das GitHub Container-Roadmap-Repository zu senden.

1. Vergewissern Sie sich, dass Sie das selbstverwaltete Add-On auf Ihrem Cluster installiert haben. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
    addon.addonVersion --output text
   ```

   Wenn Sie eine Fehlermeldung erhalten, wird das Add-On als selbstverwaltetes Add-On auf Ihrem Cluster installiert. Führen Sie die verbleibenden Schritte in diesem Verfahren aus. Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Verwenden Sie zum Aktualisieren des Amazon-EKS-Typs des Add-ons das Verfahren in Aktualisieren des Amazon-EKS-Add-ons und nicht das Verfahren in diesem Thema. Wenn Sie mit den Unterschieden zwischen den Add-On-Typen nicht vertraut sind, finden Sie Informationen unter Amazon-EKS-Add-ons.
2. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist.

   ```
   kubectl describe deployment coredns -n kube-system | grep Image | cut -d ":" -f 3
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.8.7-eksbuild.2
   ```

3. Wenn Ihre aktuelle CoreDNS-Version `v1.5.0` oder höher, aber älter als die in der Tabelle der CoreDNS-Versionen aufgeführte Version ist, überspringen Sie diesen Schritt. Wenn Ihre aktuelle Version älter als `1.5.0` ist, müssen Sie die `ConfigMap` für CoreDNS ändern, um das Forward-Add-on anstelle des Proxy-Add-ons zu verwenden.

   1. Öffnen Sie die configmap mit dem folgenden Befehl.

      ```
      kubectl edit configmap coredns -n kube-system
      ```

   2. Ersetzen Sie `proxy` in der folgenden Zeile durch `forward`. Speichern Sie die Datei und schließen Sie den Editor.

      ```
      proxy . /etc/resolv.conf
      ```

4. Wenn Sie Ihren Cluster ursprünglich auf Kubernetes `1.17` oder älter bereitgestellt haben, müssen Sie möglicherweise eine eingestellte Zeile aus Ihrem CoreDNS-Manifest entfernen.

   > **Important**
   >
   > Sie müssen diesen Schritt ausführen, bevor Sie auf CoreDNS-Version `1.7.0` aktualisieren. Es wird jedoch empfohlen, diesen Schritt auch dann auszuführen, wenn Sie auf eine ältere Version aktualisieren.

   1. Überprüfen Sie, ob Ihr CoreDNS-Manifest die Zeile enthält.

      ```
      kubectl get configmap coredns -n kube-system -o jsonpath='{$.data.Corefile}' |
       grep upstream
      ```

      Wenn keine Ausgabe zurückgegeben wird, enthält Ihr Manifest die Zeile nicht und Sie können mit dem nächsten Schritt fortfahren, um CoreDNS zu aktualisieren. Wenn eine Ausgabe zurückgegeben wird, müssen Sie die Zeile entfernen.

   2. Bearbeiten Sie das `ConfigMap` mit dem folgenden Befehl und entfernen Sie die Zeile in der Datei, die das Wort `upstream` enthält. Ändern Sie sonst nichts in der Datei. Nachdem die Linie entfernt wurde, speichern Sie die Änderungen.

      ```
      kubectl edit configmap coredns -n kube-system -o yaml
      ```

5. Rufen Sie Ihre aktuelle CoreDNS-Image-Version ab:

```
kubectl describe deployment coredns -n kube-system | grep Image
```

Eine Beispielausgabe sieht wie folgt aus.

```
602401143452.dkr.ecr.region-code.amazonaws.com/eks/coredns:v1.8.7-eksbuild.2
```

6. Wenn Sie auf CoreDNS `1.8.3` aktualisieren, müssen Sie die `endpointslices`-Berechtigung zu `system:coredns` Kubernetes `clusterrole` hinzufügen.

```
kubectl edit clusterrole system:coredns -n kube-system
```

Fügen Sie die folgenden Zeilen unter den vorhandenen Berechtigungszeilen im `rules`-Abschnitt der Datei hinzu.

```
[...]
- apiGroups:
  - discovery.k8s.io
  resources:
  - endpointslices
  verbs:
  - list
  - watch
[...]
```

7. Aktualisieren Sie das CoreDNS-Add-On, indem Sie *602401143452* und *region-code* durch die Werte aus der Ausgabe ersetzen, die in einem vorherigen Schritt zurückgegeben wurde. Ersetzen Sie *v1.11.1-eksbuild.9* durch die aktuelle CoreDNS-Version, die in der aktuellen Versionstabelle für Ihre Kubernetes-Version aufgeführt ist.

```
kubectl set image deployment.apps/coredns -n kube-system
 coredns=602401143452.dkr.ecr.region-code.amazonaws.com/eks/coredns:v1.11.1-
eksbuild.9
```

Eine Beispielausgabe sieht wie folgt aus.

```
deployment.apps/coredns image updated
```

8. Überprüfen Sie die Container-Image-Version erneut, um sicherzustellen, dass sie auf die Version aktualisiert wurde, die Sie im vorherigen Schritt angegeben haben.

```
kubectl describe deployment coredns -n kube-system | grep Image | cut -d ":" -f 3
```

Eine Beispielausgabe sieht wie folgt aus.

```
v1.11.1-eksbuild.9
```

Automatische Skalierung CoreDNS

Wenn Sie einen Amazon EKS-Cluster mit mindestens einem Knoten starten, werden standardmäßig zwei Replikate des CoreDNS Images bereitgestellt, unabhängig von der Anzahl der in Ihrem Cluster bereitgestellten Knoten. Deployment Die CoreDNS Pods bieten die Namensauflösung für alle Pods im Cluster. Anwendungen verwenden die Namensauflösung, um eine Verbindung zu Pods und Diensten im Cluster sowie zu Diensten außerhalb des Clusters herzustellen. Wenn die Anzahl der Anfragen zur Namensauflösung (Abfragen) von Pods zunimmt, können die CoreDNS Pods überlastet und langsamer werden und Anfragen zurückweisen, die die Pods nicht verarbeiten können.

Um die erhöhte Belastung der CoreDNS Pods zu bewältigen, sollten Sie ein Autoscaling-System für in Betracht ziehen. CoreDNS Amazon EKS kann die automatische Skalierung der CoreDNS Bereitstellung in der EKS-Add-on-Version von CoreDNS verwalten. Dieser CoreDNS Autoscaler überwacht kontinuierlich den Clusterstatus, einschließlich der Anzahl der Knoten und CPU-Kerne. Auf der Grundlage dieser Informationen passt der Controller die Anzahl der Replikate der CoreDNS Bereitstellung in einem EKS-Cluster dynamisch an. Diese Funktion funktioniert für jede CoreDNS `v1.9` EKS-Release-Version `1.25` und höher. Weitere Informationen darüber, welche Versionen mit CoreDNS Autoscaling kompatibel sind, finden Sie im folgenden Abschnitt.

Wir empfehlen, diese Funktion zusammen mit anderen bewährten Methoden für EKS-Cluster-Autoscaling zu verwenden, um die allgemeine Anwendungsverfügbarkeit und die Skalierbarkeit des Clusters zu verbessern.

Voraussetzungen

Damit Amazon EKS Ihre CoreDNS Bereitstellung skalieren kann, gibt es drei Voraussetzungen:

- Sie müssen die EKS Add-on-Version von verwendenCoreDNS.

- Auf Ihrem Cluster müssen mindestens die Cluster-Mindestversionen und Plattformversionen ausgeführt werden.
- Auf Ihrem Cluster muss mindestens die Mindestversion des EKS-Add-ons von ausgeführt CoreDNS werden.

Minimale Cluster-Version

Die automatische Skalierung von CoreDNS erfolgt durch eine neue Komponente in der Cluster-Steuerebene, die von Amazon EKS verwaltet wird. Aus diesem Grund müssen Sie Ihren Cluster auf eine EKS-Version aktualisieren, die die minimale Plattformversion unterstützt, die die neue Komponente enthält.

Ein neuer Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS. Der Cluster muss Kubernetes Version `1.25` oder höher aufweisen. Auf dem Cluster muss eine der in der folgenden Tabelle aufgeführten Kubernetes Versionen und Plattformversionen oder eine neuere Version ausgeführt werden. Beachten Sie, dass alle Kubernetes- und Plattformversionen, die über die aufgeführten hinausgehen, ebenfalls unterstützt werden. Sie können Ihre aktuelle Kubernetes-Version überprüfen, indem Sie *`my-cluster`* im folgenden Befehl durch den Namen Ihres Clusters ersetzen und dann den geänderten Befehl ausführen:

```
aws eks describe-cluster
              --name my-cluster --query cluster.version --output
              text
```

Kubernetes-Version	Plattformversion
1.29.3	eks.7
1.28.8	eks.13
1.27.12	eks.17
1.26.15	eks.18
1.25.16	eks.19

> **Note**
>
> Jede Plattformversion späterer Kubernetes Versionen wird ebenfalls unterstützt, z. B. Kubernetes Versionen `1.30` von `eks.1` und ab.

Mindestversion des EKS-Add-ons

Kubernetes-Version	**1.29**	**1.28**	**1.27**	**1.26**	**1.25**
	`v1.11.1-eksbuild.9`	`v1.10.1-eksbuild.11`	`v1.10.1-eksbuild.11`	`v1.9.3-eksbuild.15`	`v1.9.3-eksbuild.15`

Konfiguration von CoreDNS Autoscaling im AWS Management Console

1. Stellen Sie sicher, dass Ihr Cluster der Cluster-Mindestversion entspricht oder diese übersteigt.

   Amazon EKS aktualisiert Cluster zwischen Plattformversionen derselben Kubernetes Version automatisch, und Sie können diesen Vorgang nicht selbst starten. Stattdessen können Sie Ihren Cluster auf die nächste Kubernetes Version aktualisieren, und der Cluster wird auf diese K8s-Version und die neueste Plattformversion aktualisiert. Wenn Sie beispielsweise von auf aktualisieren `1.25` `1.26`, wird der Cluster auf aktualisiert. `1.26.15 eks.18`

   Neue Kubernetes-Versionen führen oft bedeutende Änderungen ein. Daher empfehlen wir Ihnen, das Verhalten Ihrer Anwendungen zu testen, indem Sie einen separaten Cluster der neuen Kubernetes Version verwenden, bevor Sie Ihre Produktionscluster aktualisieren.

   Gehen Sie wie unter beschrieben vor, um einen Cluster auf eine neue Kubernetes Version zu aktualisierenAktualisieren einer Amazon-EKS-Cluster-Kubernetes-Version.
2. Stellen Sie sicher, dass Sie das EKS-Add-on für CoreDNS und nicht für die selbstverwaltete CoreDNS Bereitstellung haben.

   Je nachdem, mit welchem Tool Sie Ihr Cluster erstellt haben, ist der Add-on vom Typ Amazon EKS möglicherweise derzeit nicht auf Ihrem Cluster installiert. Um zu sehen, welcher Typ des

Add-Ons auf Ihrem Cluster installiert ist, können Sie den folgenden Befehl ausführen. Ersetzen Sie `my-cluster` mit dem Namen Ihres Clusters.

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
 addon.addonVersion --output text
```

Wenn eine Versionsnummer zurückgegeben wird, haben Sie den Amazon EKS-Typ des Add-ons auf Ihrem Cluster installiert und können mit dem nächsten Schritt fortfahren. Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Führen Sie die verbleibenden Schritte des Verfahrens ausAdd-on vom Typ Amazon EKS erstellen, um die selbstverwaltete Version durch das Amazon EKS-Add-on zu ersetzen.

3. Stellen Sie sicher, dass Ihr EKS-Add-on für dieselbe oder eine höhere Version als die Mindestversion des EKS-Add-ons installiert CoreDNS ist.

   Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist. Sie können den folgenden Befehl einchecken AWS Management Console oder ausführen:

   ```
   kubectl describe deployment coredns --namespace kube-system | grep coredns: | cut -
   d : -f 3
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.10.1-eksbuild.11
   ```

   Vergleichen Sie diese Version mit der Mindestversion des EKS Add-ons im vorherigen Abschnitt. Aktualisieren Sie das EKS-Add-on bei Bedarf auf eine höhere Version, indem Sie das Verfahren befolgenAktualisieren des Amazon-EKS-Add-ons.

4. Fügen Sie die Autoscaling-Konfiguration zu den optionalen Konfigurationseinstellungen des EKS-Add-ons hinzu.

   a. Öffnen Sie die Amazon-EKS-Konsole unterhttps://console.aws.amazon.com/eks/home#/clusters.

   b. Wählen Sie im linken Navigationsbereich Clusters (Cluster) aus. Wählen Sie anschließend den Namen des Clusters aus, für den Sie das Add-On konfigurieren möchten.

   c. Wählen Sie die Registerkarte Add-ons.

d. Wählen Sie das Feld oben rechts im CoreDNS Add-On-Feld aus und wählen Sie dann Bearbeiten.

e. Auf der CoreDNS Seite „Konfigurieren“:

i. Wählen Sie die Version aus, die Sie verwenden möchten. Wir empfehlen, dieselbe Version wie im vorherigen Schritt beizubehalten und Version und Konfiguration in separaten Aktionen zu aktualisieren.

ii. Erweitern Sie Optionale Konfigurationseinstellungen.

iii. Geben Sie den JSON-Schlüssel `"autoscaling":` und den JSON-Wert eines verschachtelten JSON-Objekts mit einem Schlüssel `"enabled":` und einem Wert `true` in das Feld Konfigurationswerte ein. Der resultierende Text muss ein gültiges JSON-Objekt sein. Wenn dieser Schlüssel und dieser Wert die einzigen Daten im Textfeld sind, setzen Sie den Schlüssel und den Wert in geschweifte Klammern `{}`. Das folgende Beispiel zeigt, dass Autoscaling aktiviert ist:

```
{
  "autoScaling": {
    "enabled": true
  }
}
```

iv. (Optional) Sie können Mindest- und Höchstwerte angeben, auf die Autoscaling die Anzahl der CoreDNS Pods skalieren kann.

Das folgende Beispiel zeigt, dass Autoscaling aktiviert ist und alle optionalen Schlüssel Werte haben. Wir empfehlen, dass die Mindestanzahl von CoreDNS Pods immer größer als 2 ist, um die Stabilität des DNS-Dienstes im Cluster zu gewährleisten.

```
{
  "autoScaling": {
    "enabled": true,
    "minReplicas": 2,
    "maxReplicas": 10
  }
}
```

f. Um die neue Konfiguration durch Ersetzen der CoreDNS Pods anzuwenden, wählen Sie Änderungen speichern.

Amazon EKS wendet Änderungen an den EKS-Add-Ons mithilfe eines Rollouts von Kubernetes Deployment for CoreDNS an. Sie können den Status des Rollouts im Update-Verlauf des Add-ons in und mit verfolgen. AWS Management Console `kubectl rollout status deployment/coredns --namespace kube-system`

`kubectl rollout`hat die folgenden Befehle:

```
$ kubectl rollout

history  -- View rollout history
pause    -- Mark the provided resource as paused
restart  -- Restart a resource
resume   -- Resume a paused resource
status   -- Show the status of the rollout
undo     -- Undo a previous rollout
```

Wenn der Rollout zu lange dauert, macht Amazon EKS den Rollout rückgängig und eine Meldung mit dem Typ Addon-Update und dem Status Fehlgeschlagen wird zum Update-Verlauf des Add-ons hinzugefügt. Um Probleme zu untersuchen, beginnen Sie mit dem Verlauf des Rollouts und starten Sie es `kubectl logs` auf einem CoreDNS Pod, um die Protokolle von einzusehen. CoreDNS

5. Wenn der neue Eintrag im Update-Verlauf den Status Erfolgreich hat, ist der Rollout abgeschlossen und das Add-on verwendet die neue Konfiguration in allen Pods. CoreDNS Wenn Sie die Anzahl der Knoten und CPU-Kerne der Knoten im Cluster ändern, skaliert Amazon EKS die Anzahl der Replikate der CoreDNS Bereitstellung.

Konfiguration von CoreDNS Autoscaling im AWS Command Line Interface

1. Stellen Sie sicher, dass Ihr Cluster der Cluster-Mindestversion entspricht oder diese übersteigt.

   Amazon EKS aktualisiert Cluster zwischen Plattformversionen derselben Kubernetes Version automatisch, und Sie können diesen Vorgang nicht selbst starten. Stattdessen können Sie Ihren Cluster auf die nächste Kubernetes Version aktualisieren, und der Cluster wird auf diese K8s-Version und die neueste Plattformversion aktualisiert. Wenn Sie beispielsweise von auf aktualisieren `1.251.26`, wird der Cluster auf aktualisiert. `1.26.15 eks.18`

Neue Kubernetes-Versionen führen oft bedeutende Änderungen ein. Daher empfehlen wir Ihnen, das Verhalten Ihrer Anwendungen zu testen, indem Sie einen separaten Cluster der neuen Kubernetes Version verwenden, bevor Sie Ihre Produktionscluster aktualisieren.

Gehen Sie wie unter beschrieben vor, um einen Cluster auf eine neue Kubernetes Version zu aktualisierenAktualisieren einer Amazon-EKS-Cluster-Kubernetes-Version.

2. Stellen Sie sicher, dass Sie das EKS-Add-on für CoreDNS und nicht für die selbstverwaltete CoreDNS Bereitstellung haben.

   Je nachdem, mit welchem Tool Sie Ihr Cluster erstellt haben, ist der Add-on vom Typ Amazon EKS möglicherweise derzeit nicht auf Ihrem Cluster installiert. Um zu sehen, welcher Typ des Add-Ons auf Ihrem Cluster installiert ist, können Sie den folgenden Befehl ausführen. Ersetzen Sie `my-cluster` mit dem Namen Ihres Clusters.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
    addon.addonVersion --output text
   ```

   Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Führen Sie die verbleibenden Schritte des Verfahrens ausAdd-on vom Typ Amazon EKS erstellen, um die selbstverwaltete Version durch das Amazon EKS-Add-on zu ersetzen.

3. Stellen Sie sicher, dass Ihr EKS-Add-on für dieselbe oder eine höhere Version als die Mindestversion des EKS-Add-ons installiert CoreDNS ist.

   Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist. Sie können den folgenden Befehl einchecken AWS Management Console oder ausführen:

   ```
   kubectl describe deployment coredns --namespace kube-system | grep coredns: | cut -
   d : -f 3
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.10.1-eksbuild.11
   ```

Vergleichen Sie diese Version mit der Mindestversion des EKS Add-ons im vorherigen Abschnitt. Aktualisieren Sie das EKS-Add-on bei Bedarf auf eine höhere Version, indem Sie das Verfahren befolgenAktualisieren des Amazon-EKS-Add-ons.

4. Fügen Sie die Autoscaling-Konfiguration zu den optionalen Konfigurationseinstellungen des EKS-Add-ons hinzu.

   Führen Sie den folgenden AWS CLI Befehl aus. Ersetzen Sie `my-cluster` durch den Namen Ihres Clusters und den IAM-Rollen-ARN durch die Rolle, die Sie verwenden.

   ```
   aws eks update-addon --cluster-name my-cluster --addon-name coredns \
       --resolve-conflicts PRESERVE --configuration-values '{"autoScaling":
   {"enabled":true}}'
   ```

   Amazon EKS wendet Änderungen an den EKS-Add-Ons mithilfe eines Rollouts von Kubernetes Deployment for CoreDNS an. Sie können den Status des Rollouts im Update-Verlauf des Add-ons in und mit verfolgen. AWS Management Console `kubectl rollout status deployment/coredns --namespace kube-system`

   `kubectl rollout`hat die folgenden Befehle:

   ```
   kubectl rollout

   history  -- View rollout history
   pause    -- Mark the provided resource as paused
   restart  -- Restart a resource
   resume   -- Resume a paused resource
   status   -- Show the status of the rollout
   undo     -- Undo a previous rollout
   ```

   Wenn der Rollout zu lange dauert, macht Amazon EKS den Rollout rückgängig und eine Meldung mit dem Typ Addon-Update und dem Status Fehlgeschlagen wird zum Update-Verlauf des Add-ons hinzugefügt. Um Probleme zu untersuchen, beginnen Sie mit dem Verlauf des Rollouts und starten Sie es `kubectl logs` auf einem CoreDNS Pod, um die Protokolle von einzusehen. CoreDNS

5. (Optional) Sie können Mindest- und Höchstwerte angeben, auf die Autoscaling die Anzahl der CoreDNS Pods skalieren kann.

Das folgende Beispiel zeigt, dass Autoscaling aktiviert ist und alle optionalen Schlüssel Werte haben. Wir empfehlen, dass die Mindestanzahl von CoreDNS Pods immer größer als 2 ist, um die Stabilität des DNS-Dienstes im Cluster zu gewährleisten.

```
aws eks update-addon --cluster-name my-cluster --addon-name coredns \
    --resolve-conflicts PRESERVE --configuration-values '{"autoScaling":
{"enabled":true}, "minReplicas": 2, "maxReplicas": 10}'
```

6. Überprüfen Sie den Status des Updates für das Add-on, indem Sie den folgenden Befehl ausführen:

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns \
```

Wenn Sie diese Zeile sehen:`"status": "ACTIVE"`, ist der Rollout abgeschlossen und das Add-on verwendet die neue Konfiguration in allen CoreDNS Pods. Wenn Sie die Anzahl der Knoten und CPU-Kerne der Knoten im Cluster ändern, skaliert Amazon EKS die Anzahl der Replikate der CoreDNS Bereitstellung.

CoreDNS-Metriken

CoreDNS als EKS-Add-on stellt die Metriken von CoreDNS einem Port `9153` im Prometheus-Format im `kube-dns`-Service zur Verfügung. Sie können Prometheus, den Amazon-CloudWatch-Agenten oder ein anderes kompatibles System verwenden, um diese Metriken zu erfassen.

Ein Beispiel für eine Scrape-Konfiguration, die sowohl mit Prometheus als auch mit dem CloudWatch-Agenten kompatibel ist, finden Sie unter CloudWatch-Agentenkonfiguration für Prometheus im Amazon-CloudWatch-Benutzerhandbuch.

Arbeiten mit dem **`kube-proxy`** Kubernetes-Add-on

> **Important**
>
> Wir empfehlen, den Amazon-EKS-Typ des Add-Ons zu Ihrem Cluster hinzuzufügen, anstatt den selbstverwalteten Typ des Add-Ons zu verwenden. Wenn Sie noch keine Erfahrung mit den Unterschieden zwischen den Typen haben, finden Sie weitere Informationen unter the section called “Amazon-EKS-Add-ons”. Weitere Informationen zum Hinzufügen eines

Amazon-EKS-Add-ons zu Ihrem Cluster finden Sie unter the section called “Erstellen eines Add-Ons”. Wenn Sie das Amazon EKS-Add-on nicht verwenden können, empfehlen wir Ihnen, ein Problem mit der Begründung, warum Sie das nicht können, an das GitHub Container-Roadmap-Repository zu senden.

Das `kube-proxy`-Add-On wird auf jedem Amazon-EC2-Knoten in Ihrem Amazon-EKS-Cluster bereitgestellt. Es verwaltet Netzwerkregeln auf Ihren Knoten und ermöglicht die Netzwerkkommunikation mit Ihren Pods. Das Add-On wird nicht auf Fargate Knoten in Ihrem Cluster bereitgestellt. Weitere Informationen finden Sie unter `kube-proxy` in der Kubernetes-Dokumentation.

In der folgenden Tabelle finden Sie die aktuelle Version des Kubernetes-Add-ons, die für jede Amazon-EKS-Cluster-Version verfügbar ist.

Kubernetes-Version	**1.30**	**1.29**	**1.28**	**1.27**	**1.26**	**1.25**	**1.24**	**1.23**
	v1.30.0-eksbuild	v1.29.3-eksbuild	v1.28.8-eksbuild	v1.27.16-eksbuild.5	v1.26.15-eksbuild.5	v1.25.16-eksbuild.8	v1.24.17-eksbuild.8	v1.23.17-eksbuild.9

Important

Eine frühere Version der Dokumentation war falsch. `kube-proxy`Versionen `v1.28.5`v`1.27.9`, und `v1.26.12` sind nicht verfügbar.
Wenn Sie dieses Add-on selbst verwalten, stimmen die Versionen in der Tabelle möglicherweise nicht mit den verfügbaren selbstverwalteten Versionen überein.

Für jede Amazon-EKS-Cluster-Version sind zwei Typen des `kube-proxy`-Container-Images verfügbar:

- Standard – Dieser Image-Typ basiert auf einem Debian-basierten Docker-Image, das von der Kubernetes-Upstream-Community gewartet wird.

- Minimal – Dieser Image-Typ basiert auf einem von Amazon EKS Distro gewarteten minimalen Basis-Image, das minimale Pakete enthält und keine Shells enthält. Weitere Informationen finden Sie unter Amazon-EKS-Distro.

Neueste verfügbare selbstverwaltete **`kube-proxy`**-Container-Image-Version für die einzelnen Clusterversionen von Amazon EKS

Image Type	**1.30**	**1.29**	**1.28**	**1.27**	**1.26**	**1.25**	**1.24**	**1.23**
kube-proxy (Standardtyp)	Es ist nur ein Minimalty p verfügba .	Es ist nur ein Minimalty p verfügba .	Es ist nur ein Minimalty p verfügba .	Es ist nur ein Minimalty p verfügba .	Es ist nur ein Minimalty p verfügba .	Es ist nur ein Minimalty p verfügba .	v1.24.1 eksbuil 2	v1.23.16-eksbuild. 2
kube-proxy (minimaler Typ)	v1.30.(m inimal- ek sbuild.	v1.29.3 m inimal- ek sbuild.	v1.28.8 m inimal- ek sbuild.	v1.27.1 minimal e ksbuilc	v1.26.1 minimal e ksbuilc	v1.25.1 minimal e ksbuilc	v1.24.1 minimal e ksbuilc	v1.23.17-minimal-e ksbuild.5

Important

- Der standardmäßige Image-Typ ist für die Kubernetes-Version `1.25` und höher nicht verfügbar. Sie müssen den minimalen Image-Typ verwenden.
- Wenn Sie ein Add-on vom Typ Amazon EKS aktualisieren, geben Sie eine gültige Amazon-EKS-Add-on-Version an, bei der es sich möglicherweise nicht um eine in dieser Tabelle aufgeführte Version handelt. Dies liegt daran, dass die Amazon-EKS-Add-on-Versionen nicht immer mit den Container-Image-Versionen übereinstimmen, die bei der Aktualisierung des selbstverwalteten Typs dieses Add-ons angegeben wurden. Wenn Sie den selbstverwalteten Typ dieses Add-ons aktualisieren, geben Sie eine gültige Container-Image-Version an, die in dieser Tabelle aufgeführt ist.

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.

Überlegungen

- `Kube-proxy` auf einem Amazon-EKS-Cluster verfügt über die gleiche Kompatibilitäts- und Skew-Richtlinie wie Kubernetes. Weitere Informationen erhalten Sie unter Rufen Sie die Kompatibilität der Addon-Version ab.
- `Kube-proxy` muss dieselbe Nebenversion wie `kubelet` auf Ihren Amazon-EC2-Knoten sein.
- `Kube-proxy` darf nicht höher als die Nebenversion Ihrer Cluster-Steuerebene sein.
- Wenn Sie Ihren Cluster kürzlich auf eine neue Kubernetes-Nebenversion aktualisiert haben, aktualisieren Sie Ihre Amazon-EC2-Knoten auf dieselbe Nebenversion, bevor Sie `kube-proxy` auf dieselbe Nebenversion wie Ihre Knoten aktualisieren.

Wie Sie das selbstverwaltete **`kube-proxy`**-Add-On aktualisieren

1. Vergewissern Sie sich, dass Sie das selbstverwaltete Add-On auf Ihrem Cluster installiert haben. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name kube-proxy --query
    addon.addonVersion --output text
   ```

   Wenn Sie eine Fehlermeldung erhalten, wird das Add-On als selbstverwaltetes Add-On auf Ihrem Cluster installiert. Die verbleibenden Schritte in diesem Thema beziehen sich auf die Aktualisierung des selbstverwalteten Typs des Add-Ons. Wenn Sie eine Versionsnummer zurückgeben, wird der Amazon-EKS-Typ des Add-Ons auf Ihrem Cluster installiert. Verwenden Sie zum Aktualisieren das Verfahren in Aktualisieren eines Add-Ons und nicht das Verfahren in diesem Thema. Wenn Sie mit den Unterschieden zwischen den Add-On-Typen nicht vertraut sind, finden Sie Informationen unter Amazon-EKS-Add-ons.
2. Sehen Sie, welche Version des Container-Images derzeit auf Ihrem Cluster installiert ist.

   ```
   kubectl describe daemonset kube-proxy -n kube-system | grep Image
   ```

   Eine Beispielausgabe sieht wie folgt aus.

```
Image:    602401143452.dkr.ecr.region-code.amazonaws.com/eks/kube-proxy:v1.29.1-eksbuild.2
```

In der Beispielausgabe ist *v1.29.1-eksbuild.2* die auf dem Cluster installierte Version.

3. Aktualisieren Sie das kube-proxy-Add-On, indem Sie *602401143452* und *region-code* durch die Werte aus Ihrer Ausgabe im vorherigen Schritt ersetzen. Ersetzen Sie *v1.30.0-eksbuild.3* durch die kube-proxy-Version, die in der neuesten verfügbaren kube-proxy-Container-Image-Version für die einzelnen Clusterversionen von Amazon EKS aufgeführt ist. Sie können eine Versionsnummer für den Image-Typ Standard oder Minimal angeben.

```
kubectl set image daemonset.apps/kube-proxy -n kube-system kube-proxy=602401143452.dkr.ecr.region-code.amazonaws.com/eks/kube-proxy:v1.30.0-eksbuild.3
```

Eine Beispielausgabe sieht wie folgt aus.

```
daemonset.apps/kube-proxy image updated
```

4. Vergewissern Sie sich, dass die neue Version jetzt auf Ihrem Cluster installiert ist.

```
kubectl describe daemonset kube-proxy -n kube-system | grep Image | cut -d ":" -f 3
```

Eine Beispielausgabe sieht wie folgt aus.

```
v1.30.0-eksbuild.3
```

5. Wenn Sie x86- und Arm-Knoten im selben Cluster verwenden und Ihr Cluster vor dem 17. August 2020 bereitgestellt wurde. Bearbeiten Sie dann Ihr kube-proxy-Manifest, um einen Knotenselektor für mehrere Hardwarearchitekturen mit dem folgenden Befehl einzuschließen. Dies ist ein einmaliger Vorgang. Nachdem Sie den Selektor zu Ihrem Manifest hinzugefügt haben, müssen Sie ihn nicht bei jeder Aktualisierung des Add-Ons hinzufügen. Wenn Ihr Cluster am oder nach dem 17. August 2020 bereitgestellt wurde, ist kube-proxy bereits Multi-Architektur-fähig.

```
kubectl edit -n kube-system daemonset/kube-proxy
```

Fügen Sie der Datei im Editor den folgenden Knotenselektor hinzu und speichern Sie die Datei. Ein Beispiel dafür, wo dieser Text in den Editor eingefügt werden soll, finden Sie in der CNI-Manifest-Datei auf GitHub. Dadurch kann Kubernetes basierend auf der Hardwarearchitektur des Knotens das richtige Hardware-Image abrufen.

```
- key: "kubernetes.io/arch"
  operator: In
  values:
  - amd64
  - arm64
```

6. Wenn Ihr Cluster ursprünglich mit der Kubernetes-Version `1.14` oder höher erstellt wurde, können Sie diesen Schritt überspringen, da `kube-proxy` bereits diese `Affinity Rule` enthält. Wenn Sie ursprünglich einen Amazon-EKS-Cluster mit der Kubernetes-Version `1.13` oder früher erstellt haben und Fargate-Knoten in Ihrem Cluster verwenden älter, bearbeiten Sie Ihr `kube-proxy`-Manifest so, dass es eine `NodeAffinity`-Regel enthält, um zu verhindern, dass `kube-proxy` Pods auf Fargate-Knoten geplant werden. Dies ist eine einmalige Bearbeitung. Sobald Sie das `Affinity Rule` zu Ihrem Manifest hinzugefügt haben, müssen Sie es nicht jedes Mal hinzufügen, wenn Sie das Add-on aktualisieren. Bearbeiten Sie Ihr `kube-proxy-DaemonSet`.

   ```
   kubectl edit -n kube-system daemonset/kube-proxy
   ```

   Fügen Sie dem Abschnitt `DaemonSet spec` der Datei im Editor das folgende `Affinity Rule` hinzu und speichern Sie die Datei. Ein Beispiel dafür, wo dieser Text in den Editor eingefügt werden soll, finden Sie in der CNI-Manifest-Datei auf GitHub.

   ```
   - key: eks.amazonaws.com/compute-type
     operator: NotIn
     values:
     - fargate
   ```

Zugriff auf Amazon Elastic Kubernetes Service über einen Schnittstellen-Endpunkt (AWS PrivateLink)

Sie können AWS PrivateLink es verwenden, um eine private Verbindung zwischen Ihrer VPC und Amazon Elastic Kubernetes Service herzustellen. Sie können auf Amazon EKS zugreifen, als wäre

es in Ihrer VPC, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder AWS Direct Connect eine Verbindung verwenden zu müssen. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um auf Amazon EKS zuzugreifen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellenendpunkt erstellen, der von AWS PrivateLink unterstützt wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Hierbei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Eingangspunkt für den Datenverkehr dienen, der für Amazon EKS bestimmt ist.

Weitere Informationen finden Sie unter Zugriff auf AWS-Services über AWS PrivateLink im AWS PrivateLink -Leitfaden.

Überlegungen zu Amazon EKS

- Bevor Sie einen Schnittstellenendpunkt für Amazon EKS einrichten, lesen Sie die Überlegungen im AWS PrivateLink -Leitfaden.
- Amazon EKS unterstützt Aufrufe an alle seine API-Aktionen über den Schnittstellenendpunkt, aber nicht an die Kubernetes-APIs. Der Kubernetes-API-Server unterstützt bereits einen privaten Endpunkt. Der private Endpunkt des Kubernetes-API-Servers erstellt einen privaten Endpunkt für den Kubernetes-API-Server, den Sie verwenden, um mit Ihrem Cluster zu kommunizieren (unter Verwendung von Kubernetes-Verwaltungstools wie `kubectl`). Sie können den privaten Zugriff auf den Kubernetes API-Server aktivieren, sodass die gesamte Kommunikation zwischen Ihren Knoten und dem API-Server in Ihrer VPC bleibt. AWS PrivateLink für die Amazon EKS-API können Sie die Amazon EKS-APIs von Ihrer VPC aus aufrufen, ohne den Datenverkehr dem öffentlichen Internet zugänglich zu machen.
- Sie können Amazon EKS nicht so konfigurieren, dass der Zugriff nur über einen Schnittstellenendpunkt erfolgt.
- Die Standardpreise für AWS PrivateLink gelten für Schnittstellenendpunkte für Amazon EKS. Ihnen wird jede Stunde in Rechnung gestellt, die ein Schnittstellenendpunkt in jeder Availability Zone bereitgestellt wird, sowie für Daten, die über den Schnittstellenendpunkt verarbeitet werden. Weitere Informationen finden Sie unter AWS PrivateLink Preise.
- VPC-Endpunktrichtlinien werden für Amazon EKS nicht unterstützt. Standardmäßig ist der vollständige Zugriff auf Amazon EKS über den Schnittstellenendpunkt zulässig. Alternativ können Sie den Netzwerkschnittstellen der Endpunkte eine Sicherheitsgruppe zuordnen, um den Datenverkehr zu Amazon EKS über den Schnittstellenendpunkt zu steuern.

- Sie können VPC-Ablaufprotokolle verwenden, um Informationen über IP-Datenverkehr zu und von Netzwerkschnittstellen, einschließlich Schnittstellenendpunkten, zu erfassen. Sie können Flow-Protokolldaten auf Amazon CloudWatch oder Amazon S3 veröffentlichen. Weitere Informationen finden Sie unter Protokollierung des IP-Verkehrs mithilfe von VPC-Ablaufprotokollen im Amazon-VPC-Benutzerhandbuch.
- Sie können von einem On-Premises-Rechenzentrum aus auf die Amazon-EKS-APIs zugreifen, indem Sie es mit einer VPC verbinden, die über einen Schnittstellenendpunkt verfügt. Sie können AWS Direct Connect oder verwenden AWS Site-to-Site VPN , um Ihre lokalen Sites mit einer VPC zu verbinden.
- Sie können andere VPCs mit einem Schnittstellenendpunkt über ein AWS Transit Gateway - oder VPC-Peering mit der VPC verbinden. VPC-Peering ist eine Netzwerkverbindung zwischen zwei VPCs. Sie können eine VPC-Peering-Verbindung zwischen Ihren VPCs oder mit einer VPC in einem anderen Konto herstellen. Die VPCs können unterschiedlich sein. AWS-Regionen Der Verkehr zwischen Peer-VPCs verbleibt im Netzwerk. AWS Der Datenverkehr wird nicht über das öffentliche Internet abgewickelt. Ein Transit-Gateway ist ein Netzwerk-Transit-Hub, den Sie zum Verbinden von VPCs verwenden können. Der Datenverkehr zwischen einer VPC und einem Transit Gateway verbleibt im AWS -globalen privaten Netzwerk. Der Datenverkehr ist nicht für das öffentliche Internet zugänglich.
- Auf VPC-Schnittstellenendpunkte für Amazon EKS kann nur über `IPv4` zugegriffen werden. `IPv6` wird nicht unterstützt.
- AWS PrivateLink Support ist in den Ländern Asien-Pazifik (Hyderabad), Asien-Pazifik (Melbourne), Asien-Pazifik (Osaka), Westkanada (Calgary), Europa (Spanien), Europa (Zürich) und Naher Osten (VAE) nicht verfügbar. AWS-Regionen

Erstellen eines Schnittstellen-Endpunkts für Amazon EKS

Sie können einen Schnittstellenendpunkt für Amazon EKS entweder mit der Amazon VPC-Konsole oder mit AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie unter Erstellen eines VPC-Endpunkts im AWS PrivateLink -Leitfaden.

Erstellen Sie einen Schnittstellen-Endpunkt für Amazon EKS mit dem folgenden Service-Namen:

```
com.amazonaws.region-code.eks
```

Das private DNS-Feature ist standardmäßig aktiviert, wenn ein Schnittstellenendpunkt für Amazon EKS und andere AWS-Services erstellt wird. Sie müssen jedoch sicherstellen, dass die folgenden

VPC-Attribute auf `true` festgelegt sind: `enableDnsHostnames` und `enableDnsSupport`. Weitere Informationen finden Sie unter Anzeigen und Aktualisieren von DNS-Attributen für Ihre VPC im Amazon-VPC-Benutzerhandbuch. Wenn das private DNS-Feature für den Schnittstellenendpunkt aktiviert ist:

- Sie können jede API-Anfrage an Amazon EKS stellen, indem Sie den standardmäßigen regionalen DNS-Namen verwenden. z. B. `eks.region.amazonaws.com`. Eine Liste der APIs finden Sie unter Aktionen in der Amazon-EKS-API-Referenz.
- Sie müssen keine Änderungen an Ihren Anwendungen vornehmen, die die EKS-APIs aufrufen.
- Jeder Anruf an den Amazon EKS-Standard-Serviceendpunkt wird automatisch über den Schnittstellenendpunkt über das private AWS Netzwerk weitergeleitet.

Workloads

Ihre Workloads werden in Containern bereitgestellt, die in Pods in Kubernetes bereitgestellt werden. Ein Pod enthält einen oder mehrere Container. In der Regel werden ein oder mehrere Pods, die denselben Service bereitstellen, in einem Kubernetes-Service bereitgestellt. Nachdem Sie mehrere Pods bereitgestellt haben, die denselben Service bereitstellen, können Sie:

- Anzeigen von Informationen über die Workloads, die auf jedem Ihrer Cluster laufen mithilfe von AWS Management Console ausgeführt wird.
- Skalieren Sie die Pods vertikal mit den Kubernetes Vertical Pod Autoscaler nach oben oder unten skalieren.
- Skalieren Sie horizontal die Anzahl der Pods, die benötigt werden, um die Nachfrage nach oben oder unten mit den Kubernetes zu decken Horizontal Pod Autoscaler.
- Erstellen Sie einen externen (für Internet-zugängliche Pods) oder eine interne (für private Pods) Network Load Balancer, um den Netzwerkverkehr über Pods auszugleichen. Der Load Balancer leitet den Datenverkehr an Layer 4 des OSI-Modells weiter.
- Erstellen Sie ein Application Load Balancing auf Amazon EKS, um den Anwendungsverkehr über Pods hinweg auszugleichen. Der Application Load Balancer leitet den Datenverkehr auf Layer 7 des OSI-Modells weiter.
- Wenn Sie neu bei Kubernetes sind, hilft Ihnen dieses Thema Bereitstellen einer Beispielanwendung.
- Sie können IP-Adressen einschränken, die einem Service zugewiesen werden können mit `externalIPs`.

Bereitstellen einer Beispielanwendung

In diesem Thema stellen Sie eine Beispielanwendung in Ihrem Cluster bereit.

Voraussetzungen

- Ein vorhandener Kubernetes-Cluster mit mindestens einem Knoten. Wenn Sie noch keinen Amazon-EKS-Cluster haben, können Sie einen mit einem der Erste Schritte mit Amazon EKS-Leitfäden bereitstellen. Wenn Sie eine Windows-Anwendung bereitstellen, müssen Sie Windows-Support für Ihren Cluster und mindestens einen Amazon-EC2-Windows-Knoten aktiviert haben.

- `Kubectl` auf Ihrem Computer installiert. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- `Kubectl` für die Kommunikation mit Ihrem Cluster konfiguriert. Weitere Informationen finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.
- Wenn Sie vorhaben, Ihre Beispiel-Workload in Fargate bereitzustellen, müssen Sie über ein vorhandenes Fargate-Profil verfügen, das denselben in diesem Tutorial erstellten Namespace enthält, nämlich `eks-sample-app`, es sei denn, Sie ändern den Namen. Wenn Sie einen der Erste-Schritte-Leitfäden benutzt haben, um Ihren Cluster zu erstellen, müssen Sie ein neues Profil erstellen oder den Namespace zu Ihrem vorhandenen Profil hinzufügen, da das in den Erste-Schritte-Leitfäden erstellte Profil den in diesem Tutorial verwendeten Namespace nicht angibt. Ihre VPC muss auch über mindestens ein privates Subnetz verfügen.

Bereitstellen einer Beispielanwendung

Obwohl viele Variablen in den folgenden Schritten veränderbar sind, empfehlen wir, Variablenwerte nur zu ändern, wo angegeben. Sobald Sie Kubernetes-Pods, Bereitstellungen und Services besser verstanden haben, können Sie mit dem Ändern anderer Werte experimentieren.

1. Erstellen Sie einen -Namespace. Ein Namespace ermöglicht es Ihnen, Ressourcen in Kubernetes zu gruppieren. Weitere Informationen finden Sie unter Namespaces in der Kubernetes-Dokumentation. Wenn Sie die Beispielanwendung für AWS Fargate bereitstellen möchten, stellen Sie sicher, dass der Wert für `namespace` in Ihrem AWS Fargate Profil `eks-sample-app` ist.

   ```
   kubectl create namespace eks-sample-app
   ```

2. Erstellen einer Kubernetes-Bereitstellung. Diese Beispielbereitstellung ruft ein Container-Image aus einem öffentlichen Repository ab und stellt drei Replikate (individuelle Pods) davon in Ihrem Cluster bereit. Weitere Informationen finden Sie unter Bereitstellungen in der Kubernetes-Dokumentation. Sie können die Anwendung an Linux- oder Windows-Knoten bereitstellen. Wenn Sie an Fargate bereitstellen, können Sie nur eine Linux-Anwendung bereitstellen.

   a. Speichern Sie die folgenden Inhalte in einer Datei namens `eks-sample-deployment.yaml`. Die Container in der Beispielanwendung verwenden keinen Netzwerkspeicher, aber möglicherweise haben Sie Anwendungen, die dies benötigen. Weitere Informationen finden Sie unter Speicher.

Linux

Das `amd64` oder `arm64` `values` unter dem Schlüssel `kubernetes.io/arch` bedeutet, dass die Anwendung in einer der beiden Hardwarearchitekturen bereitgestellt werden kann (wenn Sie beide in Ihrem Cluster haben). Dies ist möglich, weil dieses Image ein Multi-Architektur-Image ist, aber das sind nicht alle. Sie können die Hardwarearchitektur bestimmen, auf der das Image unterstützt wird, indem Sie die Image-Details im Repository anzeigen, aus dem Sie es abrufen. Wenn Sie Images bereitstellen, die keinen Hardwarearchitektur-Typ unterstützen oder an die das Image nicht bereitgestellt werden soll, entfernen Sie diesen Typ aus dem Manifest. Weitere Informationen finden Sie unter Bekannte Labels, Anmerkungen und Taints in der Kubernetes-Dokumentation.

Der Eintrag `kubernetes.io/os: linux` `nodeSelector` bedeutet, dass, wenn Sie beispielsweise Linux- und Windows-Knoten in Ihrem Cluster hätten, das Image nur auf Linux-Knoten bereitgestellt würde. Weitere Informationen finden Sie unter Bekannte Labels, Anmerkungen und Taints in der Kubernetes-Dokumentation.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: eks-sample-linux-deployment
  namespace: eks-sample-app
  labels:
    app: eks-sample-linux-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: eks-sample-linux-app
  template:
    metadata:
      labels:
        app: eks-sample-linux-app
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: kubernetes.io/arch
```

```
                operator: In
                values:
                - amd64
                - arm64
      containers:
      - name: nginx
        image: public.ecr.aws/nginx/nginx:1.23
        ports:
        - name: http
          containerPort: 80
        imagePullPolicy: IfNotPresent
      nodeSelector:
        kubernetes.io/os: linux
```

Windows

Der Eintrag `kubernetes.io/os: windows` `nodeSelector` bedeutet, dass, wenn Sie beispielsweise Windows- und Linux-Knoten in Ihrem Cluster hätten, das Image nur auf Windows-Knoten bereitgestellt würde. Weitere Informationen finden Sie unter Bekannte Labels, Anmerkungen und Taints in der Kubernetes-Dokumentation.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: eks-sample-windows-deployment
  namespace: eks-sample-app
  labels:
    app: eks-sample-windows-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: eks-sample-windows-app
  template:
    metadata:
      labels:
        app: eks-sample-windows-app
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
```

```
                - key: beta.kubernetes.io/arch
                  operator: In
                  values:
                  - amd64
      containers:
      - name: windows-server-iis
        image: mcr.microsoft.com/windows/servercore:ltsc2019
        ports:
        - name: http
          containerPort: 80
        imagePullPolicy: IfNotPresent
        command:
        - powershell.exe
        - -command
        - "Add-WindowsFeature Web-Server; Invoke-WebRequest -UseBasicParsing
 -Uri 'https://dotnetbinaries.blob.core.windows.net/servicemonitor/2.0.1.6/
ServiceMonitor.exe' -OutFile 'C:\\ServiceMonitor.exe'; echo
 '<html><body><br/><br/><marquee><H1>Hello EKS!!!<H1><marquee></body><html>'
 > C:\\inetpub\\wwwroot\\default.html; C:\\ServiceMonitor.exe 'w3svc'; "
      nodeSelector:
        kubernetes.io/os: windows
```

b. Wenden Sie das Bereitstellungs-Manifest auf Ihren Cluster an.

```
kubectl apply -f eks-sample-deployment.yaml
```

3. Erstellen Sie einen Service. Mit einem Service können Sie über eine einzige IP-Adresse oder einen einzigen Namen auf alle Replikate zugreifen. Weitere Informationen finden Sie unter Service in der Kubernetes-Dokumentation. Obwohl nicht in der Beispielanwendung implementiert, empfehlen wir Ihnen, für Anwendungen, die mit anderen AWS Diensten interagieren müssen, Kubernetes Dienstkonten für Ihre Pods Dienste zu erstellen und diese mit AWS IAM-Konten zu verknüpfen. Durch die Angabe von Servicekonten verfügen Ihre Pods nur über die Mindestberechtigungen, die Sie für die Interaktion mit anderen Services angeben. Weitere Informationen finden Sie unter IAM-Rollen für Servicekonten.

   a. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen `eks-sample-service.yaml` aus. Kubernetes weist dem Dienst eine eigene IP-Adresse zu, auf die nur innerhalb des Clusters zugegriffen werden kann. Um von außerhalb des Clusters auf den Service zuzugreifen, stellen Sie den AWS Load Balancer Controller als Load Balancer für den Anwendungs- oder den Netzwerk-Datenverkehr bereit.

Linux

```
apiVersion: v1
kind: Service
metadata:
  name: eks-sample-linux-service
  namespace: eks-sample-app
  labels:
    app: eks-sample-linux-app
spec:
  selector:
    app: eks-sample-linux-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

Windows

```
apiVersion: v1
kind: Service
metadata:
  name: eks-sample-windows-service
  namespace: eks-sample-app
  labels:
    app: eks-sample-windows-app
spec:
  selector:
    app: eks-sample-windows-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

b. Wenden Sie das Service-Manifest auf Ihren Cluster an.

```
kubectl apply -f eks-sample-service.yaml
```

4. Zeigen Sie alle im `eks-sample-app`-Namespace vorhandenen Ressourcen an.

```
kubectl get all -n eks-sample-app
```

Eine Beispielausgabe sieht wie folgt aus.

Wenn Sie Windows-Ressourcen bereitgestellt haben, sind alle Instances von *linux* in der folgenden Ausgabe `windows`. Die anderen *Beispielwerte* können von Ihrer Ausgabe abweichen.

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/eks-sample-linux-deployment-65b7669776-m6qxz     1/1     Running   0          27m
pod/eks-sample-linux-deployment-65b7669776-mmxvd     1/1     Running   0          27m
pod/eks-sample-linux-deployment-65b7669776-qzn22     1/1     Running   0          27m

NAME                                 TYPE         CLUSTER-IP      EXTERNAL-IP
 PORT(S)   AGE
service/eks-sample-linux-service     ClusterIP    10.100.74.8     <none>        80/
TCP    32m

NAME                                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/eks-sample-linux-deployment    3/3     3            3           27m

NAME                                                      DESIRED   CURRENT   READY
   AGE
replicaset.apps/eks-sample-linux-deployment-776d8f8fd8    3         3         3
   27m
```

In der Ausgabe sehen Sie den Service und die Bereitstellung, die in den Beispielmanifesten, der vorherigen Schritte, angegeben wurden. Sie sehen auch drei Pods. Dies liegt daran, dass `3` `replicas` im Beispielmanifest angegeben wurden. Weitere Informationen zu Pods finden Sie unter Pods in der Kubernetes-Dokumentation. Kubernetes erstellt automatisch die `replicaset`-Ressource, obwohl sie in den Beispielmanifesten nicht angegeben ist. Weitere Informationen zu `ReplicaSets` finden Sie ReplicaSetin der Kubernetes Dokumentation.

> **Note**
>
> Kubernetes behält die Anzahl der Replikate bei, die im Manifest angegeben sind. Wenn es sich um eine Produktionsbereitstellung handelt und Sie möchten, dass Kubernetes die Anzahl der Replikate horizontal skaliert oder die Rechenressourcen für die Pods vertikal skaliert, verwenden Sie dazu das Horizontal Pod Autoscaler und das Vertical Pod Autoscaler.

5. Zeigen Sie die Details des bereitgestellten Services an. Wenn Sie einen Windows-Service bereitgestellt haben, ersetzen Sie *linux* mit **windows**.

```
kubectl -n eks-sample-app describe service eks-sample-linux-service
```

Eine Beispielausgabe sieht wie folgt aus.

Wenn Sie Windows-Ressourcen bereitgestellt haben, sind alle Instances von *linux* in der folgenden Ausgabe `windows`. Die anderen *Beispielwerte* können von Ihrer Ausgabe abweichen.

```
Name:              eks-sample-linux-service
Namespace:         eks-sample-app
Labels:            app=eks-sample-linux-app
Annotations:       <none>
Selector:          app=eks-sample-linux-app
Type:              ClusterIP
IP Families:       <none>
IP:                10.100.74.8
IPs:               10.100.74.8
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         192.168.24.212:80,192.168.50.185:80,192.168.63.93:80
Session Affinity:  None
Events:            <none>
```

In der vorherigen Ausgabe ist der Wert für `IP:` eine eindeutige IP-Adresse, die von jedem Knoten oder Pod innerhalb des Clusters aus erreicht werden kann, jedoch nicht von außerhalb des Clusters. Die Werte für `Endpoints` sind IP-Adressen, die innerhalb Ihrer VPC den Pods zugewiesen werden, die Teil des Services sind.

6. Zeigen Sie die Details eines der in der Ausgabe aufgeführten Pods an, wenn Sie in einem vorherigen Schritt den Namespace angezeigt haben. Wenn Sie eine Windows-App bereitgestellt haben, ersetzen Sie *linux* mit **windows** und *776d8f8fd8-78w66* mit dem Wert, der für einen Ihrer Pods zurückgegeben wird.

```
kubectl -n eks-sample-app describe pod eks-sample-linux-deployment-65b7669776-m6qxz
```

Gekürzte Ausgabe

Wenn Sie Windows-Ressourcen bereitgestellt haben, sind alle Instances von *linux* in der folgenden Ausgabe windows. Die anderen *example values* können von Ihrer Ausgabe abweichen.

```
Name:         eks-sample-linux-deployment-65b7669776-m6qxz
Namespace:    eks-sample-app
Priority:     0
Node:         ip-192-168-45-132.us-west-2.compute.internal/192.168.45.132
[...]
IP:           192.168.63.93
IPs:
  IP:           192.168.63.93
Controlled By:  ReplicaSet/eks-sample-linux-deployment-65b7669776
[...]
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
[...]
Events:
  Type    Reason     Age    From
 Message
  ----    ------     ----   ----
 -------
  Normal  Scheduled  3m20s  default-scheduler
 Successfully assigned eks-sample-app/eks-sample-linux-deployment-65b7669776-m6qxz
 to ip-192-168-45-132.us-west-2.compute.internal
[...]
```

In der vorherigen Ausgabe ist der Wert für IP: eine eindeutige IP-Adresse, die dem Pod aus dem CIDR-Block zugewiesen wird, der dem Subnetz zugewiesen ist, in dem sich der Knoten befindet. Wenn Sie es vorziehen, Pods IP-Adressen aus verschiedenen CIDR-Blöcken zuzuweisen, können Sie das Standardverhalten ändern. Weitere Informationen finden Sie unter Benutzerdefinierte Netzwerke für Pods. Sie können auch sehen, dass der Kubernetes-Scheduler den Pod auf dem Node mit der IP-Adresse *192.168.45.132* geplant hat.

> **Tip**
>
> Anstatt die Befehlszeile zu verwenden, können Sie viele Details zu Pods, Services, Bereitstellungen und anderen Kubernetes-Ressourcen in der AWS Management Console anzeigen. Weitere Informationen finden Sie unter Anzeigen der Kubernetes-Ressourcen.

7. Führen Sie eine Shell auf dem Pod aus, die Sie im vorherigen Schritt beschrieben haben, und ersetzen Sie *65b7669776-m6qxz* mit der ID eines Ihrer Pods.

   Linux

   ```
   kubectl exec -it eks-sample-linux-deployment-65b7669776-m6qxz -n eks-sample-app
    -- /bin/bash
   ```

   Windows

   ```
   kubectl exec -it eks-sample-windows-deployment-65b7669776-m6qxz -n eks-sample-
   app -- powershell.exe
   ```

8. Zeigen Sie in der Pod-Shell die Ausgabe des Webservers an, der in einem vorherigen Schritt mit Ihrer Bereitstellung installiert wurde. Sie müssen nur den Servicenamen angeben. Er wird von CoreDNS, das mit einem Amazon-EKS-Cluster bereitgestellt wird, standardmäßig als IP-Adresse des Services aufgelöst.

   Linux

   ```
   curl eks-sample-linux-service
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   <!DOCTYPE html>
   <html>
   <head>
   <title>Welcome to nginx!</title>
   [...]
   ```

Windows

```
Invoke-WebRequest -uri eks-sample-windows-service/default.html -UseBasicParsing
```

Eine Beispielausgabe sieht wie folgt aus.

```
StatusCode        : 200
StatusDescription : OK
Content           : < h t m l > < b o d y > < b r / > < b r / > < m a r q u e e
 > < H 1 > H e l l o
                    E K S ! ! ! < H 1 > < m a r q u e e > < / b o d y > < h t
 m l >
```

9. Zeigen Sie in der Pod-Shell den DNS-Server für den Pod an.

Linux

```
cat /etc/resolv.conf
```

Eine Beispielausgabe sieht wie folgt aus.

```
nameserver 10.100.0.10
search eks-sample-app.svc.cluster.local svc.cluster.local cluster.local us-
west-2.compute.internal
options ndots:5
```

In der vorherigen Ausgabe wird `10.100.0.10` automatisch als `nameserver` für alle Pods zugewiesen, die im Cluster bereitgestellt werden.

Windows

```
Get-NetIPConfiguration
```

Gekürzte Ausgabe

```
InterfaceAlias       : vEthernet
[...]
IPv4Address          : 192.168.63.14
[...]
```

```
DNSServer            : 10.100.0.10
```

In der vorherigen Ausgabe wird `10.100.0.10` automatisch als DNS-Server für alle Pods zugewiesen, die im Cluster bereitgestellt werden.

10. Trennen Sie die Verbindung mit dem Pod, indem Sie `exit` eingeben.
11. Wenn Sie die Beispielanwendung nicht mehr benötigen, können Sie den Beispiel-Namespace, den Service und die Bereitstellung mit dem folgenden Befehl entfernen.

```
kubectl delete namespace eks-sample-app
```

Nächste Schritte

Nachdem Sie die Beispielanwendung bereitgestellt haben, möchten Sie vielleicht einige der folgenden Übungen ausprobieren:

- the section called “Application Load Balancing”
- the section called “Netzwerk-Load-Balancer”

Vertical Pod Autoscaler

Der Kubernetes Vertical Pod Autoscaler passt die CPU- und Speicherreservierungen für Ihre Pods automatisch an, um Ihre Anwendungen auf die richtige Größe zu skalieren. Diese Anpassung kann die Cluster-Ressourcennutzung verbessern und CPU und Arbeitsspeicher für andere Pods freigeben. In diesem Thema können Sie den Vertical Pod Autoscaler für Ihren Cluster bereitstellen und überprüfen, ob er funktioniert.

Voraussetzungen

- Sie haben einen vorhandenen Amazon-EKS-Cluster. Falls nicht, finden Sie weitere Informationen unter Erste Schritte mit Amazon EKS.
- Sie haben den Kubernetes Metrics Server installiert. Weitere Informationen finden Sie unter Installieren von Kubernetes Metrics Server.
- Sie verwenden einen `kubectl`-Client, der für die Kommunikation mit Ihrem Amazon EKS-Cluster konfiguriert ist.
- OpenSSL `1.1.1` oder höher ist auf Ihrem Gerät installiert.

Bereitstellen des Vertical Pod Autoscalers

In diesem Abschnitt stellen Sie den Vertical Pod Autoscaler für Ihren Cluster bereit.

So stellen Sie den Vertical Pod Autoscaler bereit

1. Öffnen Sie ein Terminalfenster und navigieren Sie zu einem Verzeichnis, in das Sie den Vertical Pod Autoscaler-Quellcode herunterladen möchten.
2. Klonen Sie das kubernetes/autoscaler-GitHub-Repository.

   ```
   git clone https://github.com/kubernetes/autoscaler.git
   ```

3. Wechseln Sie in das `vertical-pod-autoscaler`-Verzeichnis.

   ```
   cd autoscaler/vertical-pod-autoscaler/
   ```

4. (Optional) Wenn Sie bereits eine andere Version des Vertical Pod Autoscalers bereitgestellt haben, entfernen Sie ihn mit dem folgenden Befehl.

   ```
   ./hack/vpa-down.sh
   ```

5. Wenn die Knoten keinen Internetzugriff auf die Container-Registry `registry.k8s.io` haben, müssen Sie die folgenden Images abrufen und in ein eigenes privates Repository verschieben. Weitere Informationen zum Abrufen und Verschieben der Images in ein eigenes privates Repository finden Sie unter Kopieren eines Container-Images von einem Repository in ein anderes.

   ```
   registry.k8s.io/autoscaling/vpa-admission-controller:0.10.0
   registry.k8s.io/autoscaling/vpa-recommender:0.10.0
   registry.k8s.io/autoscaling/vpa-updater:0.10.0
   ```

   Wenn Sie die Images in ein privates Amazon-ECR-Repository verschieben, ersetzen Sie `registry.k8s.io` in den Manifesten durch Ihrer Registrierung. Ersetzen Sie *`111122223333`* durch Ihre Konto-ID. Ersetzen Sie *`region-code`* durch die AWS-Region, in der sich Ihr Cluster befindet. Beim folgenden Befehl wird davon ausgegangen, dass Sie Ihr Repository genauso benannt haben wie das Repository im Manifest. Wenn Sie das Repository anders benannt haben, müssen Sie es ebenfalls ändern.

```
sed -i.bak -e 's/registry.k8s.io/111122223333.dkr.ecr.region-
code.amazonaws.com/' ./deploy/admission-controller-deployment.yaml
sed -i.bak -e 's/registry.k8s.io/111122223333.dkr.ecr.region-
code.amazonaws.com/' ./deploy/recommender-deployment.yaml
sed -i.bak -e 's/registry.k8s.io/111122223333.dkr.ecr.region-
code.amazonaws.com/' ./deploy/updater-deployment.yaml
```

6. Stellen Sie den Vertical Pod Autoscaler mit dem folgenden Befehl für Ihren Cluster bereit.

```
./hack/vpa-up.sh
```

7. Überprüfen Sie, ob die Vertical-Pod-Autoscaler-Pods erfolgreich erstellt wurden.

```
kubectl get pods -n kube-system
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                                        READY   STATUS    RESTARTS   AGE
[...]
metrics-server-8459fc497-kfj8w              1/1     Running   0          83m
vpa-admission-controller-68c748777d-ppspd   1/1     Running   0          7s
vpa-recommender-6fc8c67d85-gljpl            1/1     Running   0          8s
vpa-updater-786b96955c-bgp9d                1/1     Running   0          8s
```

Testen der Installation von Vertical Pod Autoscaler

In diesem Abschnitt stellen Sie eine Beispielanwendung bereit, um zu überprüfen, ob der Vertical Pod Autoscaler funktioniert.

So testen Sie die Vertical Pod Autoscaler-Installation

1. Stellen Sie das Vertical Pod Autoscaler-Beispiel `hamster.yaml` mit dem folgenden Befehl bereit.

```
kubectl apply -f examples/hamster.yaml
```

2. Rufen Sie die Pods aus der `hamster`-Beispielanwendung ab.

```
kubectl get pods -l app=hamster
```

Eine Beispielausgabe sieht wie folgt aus.

```
hamster-c7d89d6db-rglf5   1/1     Running   0          48s
hamster-c7d89d6db-znvz5   1/1     Running   0          48s
```

3. Beschreiben Sie einen der Pods, um die zugehörige `cpu`- und `memory`-Reservierung anzuzeigen. Ersetzen Sie *c7d89d6db-rglf5* durch eine der IDs, die in der Ausgabe im vorherigen Schritt zurückgegeben wurden.

```
kubectl describe pod hamster-c7d89d6db-rglf5
```

Eine Beispielausgabe sieht wie folgt aus.

```
[...]
Containers:
  hamster:
    Container ID:  docker://
e76c2413fc720ac395c33b64588c82094fc8e5d590e373d5f818f3978f577e24
    Image:         registry.k8s.io/ubuntu-slim:0.1
    Image ID:      docker-pullable://registry.k8s.io/ubuntu-
slim@sha256:b6f8c3885f5880a4f1a7cf717c07242eb4858fdd5a84b5ffe35b1cf680ea17b1
    Port:          <none>
    Host Port:     <none>
    Command:
      /bin/sh
    Args:
      -c
      while true; do timeout 0.5s yes >/dev/null; sleep 0.5s; done
    State:          Running
      Started:      Fri, 27 Sep 2019 10:35:16 -0700
    Ready:          True
    Restart Count:  0
    Requests:
      cpu:        100m
      memory:     50Mi
[...]
```

Sie können sehen, dass der ursprüngliche Pod 100 Millicpu CPU und 50 Mebibyte Arbeitsspeicher reserviert. Für diese Beispielanwendung ist 100 Millicpu weniger als der Pod zum Ausführen benötigt, sodass er CPU-eingeschränkt ist. Außerdem reserviert er viel weniger Arbeitsspeicher, als er benötigt. Die Vertical-Pod-Autoscaler-Bereitstellung `vpa-recommender` analysiert die `hamster`-Pods, um festzustellen, ob die CPU- und Speicheranforderungen angemessen sind. Wenn Anpassungen erforderlich sind, werden die `vpa-updater` mit aktualisierten Werten vom Pods neu geladen.

4. Warten Sie, bis der `vpa-updater` einen neuen `hamster`-Pod startet. Dies sollte ein oder zwei Minuten dauern. Sie können die Pods mit dem folgenden Befehl überwachen.

> **Note**
>
> Wenn Sie nicht sicher sind, ob ein neuer Pod gestartet wurde, vergleichen Sie die Pod-Namen mit Ihrer vorherigen Liste. Wenn der neue Pod gestartet wird, wird ein neuer Pod-Name angezeigt.

```
kubectl get --watch Pods -l app=hamster
```

5. Wenn ein neuer `hamster`-Pod gestartet wird, beschreiben Sie ihn und zeigen Sie die aktualisierten CPU- und Speicherreservierungen an.

```
kubectl describe pod hamster-c7d89d6db-jxgfv
```

Eine Beispielausgabe sieht wie folgt aus.

```
[...]
Containers:
  hamster:
    Container ID:
 docker://2c3e7b6fb7ce0d8c86444334df654af6fb3fc88aad4c5d710eac3b1e7c58f7db
    Image:         registry.k8s.io/ubuntu-slim:0.1
    Image ID:      docker-pullable://registry.k8s.io/ubuntu-
slim@sha256:b6f8c3885f5880a4f1a7cf717c07242eb4858fdd5a84b5ffe35b1cf680ea17b1
    Port:          <none>
    Host Port:     <none>
    Command:
      /bin/sh
```

```
    Args:
      -c
      while true; do timeout 0.5s yes >/dev/null; sleep 0.5s; done
    State:          Running
      Started:      Fri, 27 Sep 2019 10:37:08 -0700
    Ready:          True
    Restart Count:  0
    Requests:
      cpu:        587m
      memory:     262144k
[...]
```

In der vorherigen Ausgabe können Sie feststellen, dass sich die `cpu`-Reservierung auf 587 Millicpu erhöht hat, was mehr als dem Fünffachen des ursprünglichen Werts entspricht. Der `memory` wurde auf 262.144 Kilobyte erhöht, was etwa 250 Mebibyte bzw. dem Fünffachen des ursprünglichen Werts entspricht. Diesem Pod standen zu wenige Ressourcen zur Verfügung. Daher hat der Vertical Pod Autoscaler unsere Schätzung mit einem wesentlich geeigneteren Wert korrigiert.

6. Beschreiben Sie die `hamster-vpa`-Ressource, um die neue Empfehlung anzuzeigen.

```
kubectl describe vpa/hamster-vpa
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:         hamster-vpa
Namespace:    default
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"autoscaling.k8s.io/
v1beta2","kind":"VerticalPodAutoscaler","metadata":{"annotations":
{},"name":"hamster-vpa","namespace":"d...
API Version:  autoscaling.k8s.io/v1beta2
Kind:         VerticalPodAutoscaler
Metadata:
  Creation Timestamp:  2019-09-27T18:22:51Z
  Generation:          23
  Resource Version:    14411
  Self Link:           /apis/autoscaling.k8s.io/v1beta2/namespaces/default/
verticalpodautoscalers/hamster-vpa
  UID:                 d0d85fb9-e153-11e9-ae53-0205785d75b0
Spec:
```

```
  Target Ref:
    API Version:  apps/v1
    Kind:         Deployment
    Name:         hamster
Status:
  Conditions:
    Last Transition Time:  2019-09-27T18:23:28Z
    Status:                True
    Type:                  RecommendationProvided
  Recommendation:
    Container Recommendations:
      Container Name:  hamster
      Lower Bound:
        Cpu:     550m
        Memory:  262144k
      Target:
        Cpu:     587m
        Memory:  262144k
      Uncapped Target:
        Cpu:     587m
        Memory:  262144k
      Upper Bound:
        Cpu:     21147m
        Memory:  387863636
Events:          <none>
```

7. Wenn Sie mit dem Experimentieren mit der Beispielanwendung fertig sind, können Sie sie mit dem folgenden Befehl löschen.

```
kubectl delete -f examples/hamster.yaml
```

Horizontal Pod Autoscaler

Der Kubernetes Horizontal Pod Autoscaler skaliert automatisch die Anzahl der Pods in einer Bereitstellung, einem Replikations-Controller oder einem Replikatsatz basierend auf der CPU-Auslastung dieser Ressource. Auf diese Weise können Ihre Anwendungen aufskalieren, um den erhöhten Bedarf zu erfüllen, oder abskalieren, wenn Ressourcen nicht benötigt werden, sodass Ihre Knoten für andere Anwendungen freigegeben werden. Wenn Sie einen CPU-Auslastungsprozentsatz festlegen, skaliert der Horizontal Pod Autoscaler Ihre Anwendung in beide Richtungen, um zu versuchen, dieses Ziel zu erreichen.

Beim Horizontal Pod Autoscaler handelt es sich um eine Standard-API-Ressource in Kubernetes, für die eine Metrikquelle (z. B. der Kubernetes-Metrikserver) in Ihrem Amazon-EKS-Cluster installiert sein muss. Den Horizontal Pod Autoscaler müssen Sie nicht in Ihrem Cluster bereitstellen oder installieren, um mit der Skalierung Ihrer Anwendungen zu beginnen. Weitere Informationen finden Sie unter Horizontal Pod Autoscaler in der Kubernetes-Dokumentation.

Verwenden Sie dieses Thema, um den Horizontal Pod Autoscaler für Ihren Amazon-EKS-Cluster vorzubereiten und zu überprüfen, ob er mit einer Beispielanwendung funktioniert.

> **Note**
>
> Dieses Thema basiert auf der Anleitung zu Horizontal Pod autoscaler in der Kubernetes-Dokumentation.

Voraussetzungen

- Sie haben einen vorhandenen Amazon-EKS-Cluster. Falls nicht, finden Sie weitere Informationen unter Erste Schritte mit Amazon EKS.
- Sie haben den Kubernetes Metrics Server installiert. Weitere Informationen finden Sie unter Installieren von Kubernetes Metrics Server.
- Sie verwenden einen `kubectl`-Client, der für die Kommunikation mit Ihrem Amazon EKS-Cluster konfiguriert ist.

Ausführen einer Horizontal Pod Autoscaler-Testanwendung

In diesem Abschnitt stellen Sie eine Beispielanwendung bereit, um zu überprüfen, ob der Horizontal Pod Autoscaler funktioniert.

> **Note**
>
> Dieses Beispiel basiert auf dem Horizontal-Pod-Autoscaler-Walkthrough in der Kubernetes-Dokumentation.

So testen Sie die Installation von Horizontal Pod Autoscaler

1. Stellen Sie mit dem folgenden Befehl eine einfache Apache-Webserveranwendung bereit.

```
kubectl apply -f https://k8s.io/examples/application/php-apache.yaml
```

Dieser Apache-Webserver-Pod verfügt über ein CPU-Limit von 500 milliCPU und wird auf Port 80 bereitgestellt.

2. Erstellen Sie eine Horizontal Pod Autoscaler-Ressource für die `php-apache`-Bereitstellung.

```
kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
```

Dieser Befehl erstellt einen Autoscaler, der 50 Prozent CPU-Auslastung für die Bereitstellung anvisiert, mit mindestens einem Pod und maximal zehn Pods. Wenn die durchschnittliche CPU-Auslastung unter 50 Prozent liegt, versucht der Autoscaler, die Anzahl der Pods in der Bereitstellung auf minimal eins reduzieren. Wenn die Last größer als 50 Prozent ist, versucht der Autoscaler, die Anzahl der Pods in der Bereitstellung auf maximal zehn zu erhöhen. Weitere Informationen finden Sie unter Wie HorizontalPodAutoscaler funktioniert ein ? in der -KubernetesDokumentation.

3. Beschreiben Sie den Autoscaler mit dem folgenden Befehl, um seine Details anzuzeigen.

```
kubectl get hpa
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME         REFERENCE                TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache   0%/50%    1         10        1          51s
```

Wie Sie sehen, ist die aktuelle CPU-Last `0%`, weil es noch keine Last auf dem Server gibt. Die Pod-Anzahl befindet sich bereits an der untersten Grenze (eins), daher kann sie nicht weiter nach unten skaliert werden.

4. Erstellen Sie eine Last für den Webserver, indem Sie einen Container ausführen.

```
kubectl run -i \
    --tty load-generator \
    --rm --image=busybox \
    --restart=Never \
    -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache; done"
```

5. Um die Skalierung der Bereitstellung zu beobachten, führen Sie in regelmäßigen Abständen den folgenden Befehl in einem anderen Terminal als demjenigen aus, in dem Sie den vorherigen Schritt ausgeführt haben.

```
kubectl get hpa php-apache
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME         REFERENCE                  TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache   250%/50%   1         10        5
 4m44s
```

Es kann mehr als eine Minute dauern, bis die Anzahl der Replikate zunimmt. Solange der tatsächliche CPU-Prozentsatz höher als der Zielprozentsatz ist, erhöht sich die Replikatanzahl auf bis zu 10. In diesem Fall ist dies`250%`, so dass die Anzahl der `REPLICAS` weiter steigt.

> **Note**
>
> Es kann einige Minuten dauern, bis die Replikatanzahl ihr Maximum erreicht. Wenn beispielsweise nur 6 Replikate erforderlich sind, damit die CPU-Last bei oder unter 50 % bleibt, wird die Last nicht über 6 Replikate hinaus skaliert.

6. Stoppen Sie die Last. Halten Sie im Terminalfenster, in dem Sie die Last erzeugen, die Last an, indem Sie die `Ctrl+C`-Tasten gedrückt halten. Sie können beobachten, wie die Replikate auf 1 zurückskalieren, indem Sie den folgenden Befehl erneut in dem Terminal ausführen, in dem Sie die Skalierung beobachten.

```
kubectl get hpa
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME         REFERENCE                  TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache   0%/50%    1         10        1          25m
```

> **Note**
>
> Der Standardzeitrahmen für das Zurückskalieren beträgt fünf Minuten. Es wird daher einige Zeit dauern, bis die Replikatanzahl wieder 1 erreicht, selbst wenn

der aktuelle CPU-Prozentsatz 0 Prozent beträgt. Der Zeitrahmen ist veränderbar. Weitere Informationen finden Sie unter Horizontal Pod Autoscaler in der Kubernetes-Dokumentation.

7. Wenn Sie mit dem Experimentieren mit Ihrer Beispielanwendung fertig sind, löschen Sie die `php-apache`-Ressourcen.

```
kubectl delete deployment.apps/php-apache service/php-apache
 horizontalpodautoscaler.autoscaling/php-apache
```

Network Load Balancing in Amazon EKS

Das Load Balancing des Netzwerkverkehrs erfolgt auf `L4` des OSI-Modells. Um den Anwendungsdatenverkehr unter zu verteilen`L7`, stellen Sie einen bereit Kubernetes`ingress`, der einen AWS Application Load Balancer bereitstellt. Weitere Informationen finden Sie unter Application Load Balancing auf Amazon EKS. Weitere Informationen zu den Unterschieden zwischen den beiden Arten von Load Balancing finden Sie auf der AWS Website unter Elastic Load Balancing Balancing-Funktionen.

Wenn Sie einen Typ vom Typ erstellen`LoadBalancer`, erstellt der Load Balancer-Controller Kubernetes `Service` des AWS Cloud-Anbieters standardmäßig AWS Classic Load Balancer, kann aber auch AWS Network Load Balancer erstellen. Dieser Controller erhält in Zukunft nur kritische Fehlerbehebungen. Weitere Informationen zur Verwendung des AWS Cloud-Provider-Load Balancers finden Sie in der Dokumentation unter AWS Cloud-Provider-Load Balancer-Controller. Kubernetes Seine Verwendung wird in diesem Thema nicht behandelt.

Wir empfehlen die Verwendung der Version `2.7.2` oder höher von AWS Load Balancer Controller anstelle des Load-Balancer-Controllers für AWS -Cloud-Anbieter. Der AWS Load Balancer Controller erstellt AWS Network Load Balancer, aber keine AWS Classic Load Balancer. Der Rest dieses Themas befasst sich mit der Verwendung des Load AWS Balancer Controllers.

Ein AWS Network Load Balancer kann den Netzwerkdatenverkehr auf Pods Amazon EC2 EC2-IP- und Instance-Ziele oder auf AWS Fargate IP-Ziele verteilen. Weitere Informationen zum AWS -Load-Balancer-Controller finden Sie unter GitHub.

Voraussetzungen

Bevor Sie mit AWS Load Balancer Controller ein Load Balancing des Netzwerkverkehrs durchführen können, müssen folgende Voraussetzungen erfüllt sein.

- Einen vorhandenen Cluster haben. Falls Sie keinen bestehenden Cluster haben, siehe Erste Schritte mit Amazon EKS. Wenn Sie die Version eines vorhandenen Clusters aktualisieren müssen, finden Sie unter Aktualisieren einer Amazon-EKS-Cluster-Kubernetes-Version.
- Stellen Sie den AWS Load Balancer Controller in Ihrem Cluster bereit. Weitere Informationen finden Sie unter Was ist die AWS Load Balancer Controller?. Wir empfehlen Version `2.7.2` oder höher.
- Mindestens ein Subnetz. Werden in einer Availability Zone mehrere markierte Subnetze gefunden, wählt der Controller das erste Subnetz, dessen Subnetz-ID lexikografisch an erster Stelle steht. Das Subnetz muss mindestens acht verfügbare IP-Adressen aufweisen.
- Wenn Sie die Version `2.1.1` oder früher des AWS Load Balancer Controller verwenden, müssen Subnetze wie folgt markiert werden. Bei Verwendung der Version `2.1.2` oder höher ist dieses Tag optional. Möglicherweise möchten Sie ein Subnetz taggen, wenn mehrere Cluster in derselben VPC ausgeführt werden, oder wenn mehrere AWS Dienste Subnetze in einer VPC gemeinsam nutzen, und Sie mehr Kontrolle darüber haben möchten, wo Load Balancer für jeden Cluster bereitgestellt werden. Wenn Sie Subnetz-IDs explizit als Anmerkung für ein Service-Objekt angeben, werden Kubernetes und die AWS Load Balancer Controller diese Subnetze direkt verwenden, um den Load Balancer zu erstellen. Subnetz-Tagging ist nicht erforderlich, wenn Sie diese Methode für die Bereitstellung von Lastausgleichsdiensten verwenden und Sie die folgenden Anforderungen für private und öffentliche Subnetz-Tagging überspringen können. Ersetzen Sie *`my-cluster`* mit Ihrem Clusternamen.
  - Schlüssel – `kubernetes.io/cluster/`*`my-cluster`*
  - Wert – `shared` oder `owned`
- Ihre öffentlichen und privaten Subnetze müssen die folgenden Anforderungen erfüllen, es sei denn, Sie geben Subnetz-IDs explizit als Anmerkung für ein Service- oder Ingress-Objekt an. Wenn Sie Load Balancer bereitstellen, indem Sie Subnetz-IDs explizit als Anmerkung für ein Service- oder Ingress-Objekt angeben, werden Kubernetes und die AWS Load Balancer Controller diese Subnetze direkt verwenden, um den Load Balancer zu erstellen, und die folgenden Tags sind nicht erforderlich.
  - Private Subnetze – Müssen im folgenden Format markiert sein. Dadurch wissen Kubernetes und der Load AWS Balancer Controller wissen, dass die Subnetze für interne Load Balancer verwendet werden können. Wenn Sie eine Amazon AWS AWS CloudFormation EKS-Vorlage

verwendeneksctl, um Ihre VPC nach dem 26. März 2020 zu erstellen, werden die Subnetze bei der Erstellung entsprechend gekennzeichnet. Weitere Informationen über die Amazon-EKS- AWS AWS CloudFormation -VPC-Vorlagen finden Sie unter Erstellen einer VPC für Ihren Amazon-EKS-Cluster.

- Schlüssel – kubernetes.io/role/internal-elb
- Wert – 1

- Öffentliche Subnetze – Müssen im folgenden Format markiert sein. Sie müssen öffentliche Subnetze in Ihrer VPC entsprechend kennzeichnen, so dass Kubernetes nur diese Subnetze für externe Load Balancer verwendet und kein öffentliches Subnetz in jeder Availability Zone wählt (in lexikographischer Reihenfolge nach Subnetz-ID): Wenn Sie eine Amazon AWS CloudFormation EKS-Vorlage verwendeneksctl, um Ihre VPC nach dem 26. März 2020 zu erstellen, werden die Subnetze bei der Erstellung entsprechend gekennzeichnet. Weitere Informationen zu den Amazon AWS CloudFormation EKS-VPC-Vorlagen finden Sie unterErstellen einer VPC für Ihren Amazon-EKS-Cluster.
  - Schlüssel – kubernetes.io/role/elb
  - Wert – 1

Wenn die Subnetzrollen-Tags nicht explizit hinzugefügt werden, prüft der Kubernetes-Dienstcontroller die Routingtabelle der Cluster-VPC-Subnetze, um festzustellen, ob das Subnetz privat oder öffentlich ist. Es wird empfohlen, dass Sie sich nicht auf dieses Verhalten verlassen und stattdessen explizit die privaten oder öffentlichen Rollentags hinzufügen. Der AWS Load Balancer Controller untersucht keine Routentabellen und erfordert, dass die privaten und öffentlichen Tags für eine erfolgreiche automatische Erkennung vorhanden sind.

Überlegungen

- Die Konfiguration Ihres Load Balancers wird durch Anmerkungen gesteuert, die zu dem Manifest für Ihren Service hinzugefügt werden. Service-Anmerkungen unterscheiden sich bei der Verwendung von von von AWS Load Balancer Controller als bei der Verwendung des Load Balancer-Controllers des AWS Cloud-Anbieters. Überprüfen Sie die Anmerkungen für den AWS Load Balancer Controller vor der Bereitstellung von Services.
- Bei Verwendung des Amazon VPC CNI plugin for Kubernetes kann der AWS Load Balancer Controller das Load Balancing zu Amazon-EC2-IP- oder -Instance-Zielen und Fargate-IP-Zielen vornehmen. Bei Verwendung von Alternativen kompatiblen CNI-Plugins kann der Controller nur das Load Balancing zu Instance-Zielen vornehmen. Weitere Informationen zu Network-Load-Balancer-Zieltypen finden Sie unter Zieltyp im Benutzerhandbuch für Network Load Balancers.

- Wenn Sie dem Load Balancer Tags hinzufügen möchten, wenn oder nachdem er erstellt wurde, fügen Sie die folgende Anmerkung in Ihre Service-Spezifikation ein. Weitere Informationen finden Sie unter AWS -Ressourcen-Tags in der AWS Load Balancer Controller-Dokumentation.

```
service.beta.kubernetes.io/aws-load-balancer-additional-resource-tags
```

- Sie können Elastic IP-Adressen zum Network Load Balancer hinzufügen, indem Sie die folgende Anmerkung hinzufügen. Ersetzen Sie die *example values* mit den `Allocation IDs` Ihrer elastischen IP-Adressen. Die Anzahl der `Allocation IDs` muss mit der Anzahl der Subnetze übereinstimmen, die für den Load Balancer verwendet werden. Weitere Informationen finden Sie in der Dokumentation zu AWS Load Balancer Controller.

```
service.beta.kubernetes.io/aws-load-balancer-eip-allocations:
 eipalloc-xxxxxxxxxxxxxxxxx,eipalloc-yyyyyyyyyyyyyyyyy
```

- Für jeden Network Load Balancer, den Sie erstellen, fügt Amazon EKS der Sicherheitsgruppe des Knotens für Clientdatenverkehr eine eingehende Regel und für jedes Load-Balancer-Subnetz in der VPC für Integritätsprüfungen eine Regel hinzu. Bereitstellung eines Dienstes vom Typ`LoadBalancer`kann fehlschlagen, wenn Amazon EKS versucht, Regeln zu erstellen, die das Kontingent für die maximal zulässige Anzahl von Regeln für eine Sicherheitsgruppe überschreiten. Weitere Informationen dazu finden Sie im Abschnitt Sicherheitsgruppe in Amazon VPC-Kontingenten im Amazon VPC-Benutzerhandbuch. Berücksichtigen Sie die folgenden Optionen, um die Wahrscheinlichkeit zu minimieren, dass die maximale Anzahl von Regeln für eine Sicherheitsgruppe überschritten wird:
  - Fordern Sie eine Erhöhung Ihrer Regeln pro Sicherheitsgruppenkontingent an. Weitere Informationen dazu finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.
  - Verwenden Sie IP-Ziele anstelle von Instance-Zielen. Mit IP-Zielen können Sie Regeln für dieselben Zielports freigeben. Sie können Load-Balancer-Subnetze manuell mit einer Anmerkung angeben. Weitere Informationen finden Sie unter Add-ons auf GitHub.
  - Verwenden Sie einen Eingang anstelle eines Services vom Typ `LoadBalancer`, um Datenverkehr an Ihren Service zu senden. Der AWS Application Load Balancer benötigt weniger Regeln als Network Load Balancer. Sie können einen ALB für mehrere Eingänge freigeben. Weitere Informationen dazu finden Sie unter Application Load Balancing auf Amazon EKS. Sie können einen Network Load Balancer nicht für mehrere Services freigeben.
  - Stellen Sie Ihre Cluster für mehrere Konten bereit.

- Wenn Ihre Pods unter Windows ausgeführt werden, kann in einem Amazon-EKS-Cluster ein einzelner Dienst mit einem Load Balancer bis zu 1 024 Backend-Pods unterstützen. Jeder Pod hat seine eigene eindeutige IP-Adresse.
- Es wird empfohlen, mit dem AWS Load Balancer Controller nur neue Network Load Balancer zu erstellen. Der Versuch, vorhandene Network Load Balancer zu ersetzen, die mit dem Load Balancer-Controller des AWS Cloud-Anbieters erstellt wurden, kann zu mehreren Network Load Balancers führen, was zu Anwendungsausfällen führen kann.

Erstellen eines Network Load Balancers

Sie können einen Netzwerk-Load-Balancer mit IP- oder Instance-Zielen erstellen.

IP targets

Sie können IP-Ziele mit Pods verwenden, die auf Amazon-EC2-Knoten oder Fargate bereitgestellt werden. Ihr Kubernetes-Service muss als Typ `LoadBalancer` erstellt werden. Weitere Informationen finden Sie LoadBalancer in der Dokumentation unter Typ. Kubernetes

Um einen Load Balancer zu erstellen, der IP-Ziele verwendet, fügen Sie die folgende Anmerkung zu einem Service-Manifest hinzu und stellen Sie den Dienst bereit. Der `external` Wert für `aws-load-balancer-type` bewirktAWS Load Balancer Controller, dass der Network Load Balancer Controller und nicht der Load Balancer Controller des AWS Cloud-Anbieters den Network Load Balancer erstellt. Sie können ein Beispiel-Service-Manifest mit den Anmerkungen anzeigen.

```
service.beta.kubernetes.io/aws-load-balancer-type: "external"
service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: "ip"
```

> **Note**
>
> Wenn Sie Load Balancing für `IPv6` Pods verwenden, fügen Sie die folgende Anmerkungen hinzu. Load Balancing über `IPv6` funktioniert nur auf IP-Ziele, nicht auf Instance-Ziele. Ohne diese Anmerkung verläuft das Load Balancing über `IPv4`.
>
> ```
> service.beta.kubernetes.io/aws-load-balancer-ip-address-type: dualstack
> ```

Network Load Balancer werden standardmäßig mit dem `internal aws-load-balancer-scheme` erstellt. Sie können Network Load Balancer in jedem Subnetz in der VPC Ihres Clusters starten, einschließlich Subnetze, die beim Erstellen Ihres Clusters nicht angegeben wurden.

Kubernetes untersucht die Routing-Tabelle auf Ihre Subnetze, um zu identifizieren, ob sie öffentlich oder privat sind. Öffentliche Subnetze verfügen über einen direkten Zugang zum Internet über ein Internet-Gateway, nicht aber private Subnetze.

Wenn Sie einen Network Load Balancer in einem öffentlichen Subnetz erstellen möchten, um die Lastenverteilung auf Amazon-EC2-Knoten vorzunehmen (Fargate kann nur privat sein), geben Sie `internet-facing` mit folgender Anmerkung an:

```
service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"
```

> **Note**
>
> Aus Gründen der Abwärtskompatibilität wird `service.beta.kubernetes.io/aws-load-balancer-type: "nlb-ip"` weiterhin unterstützt. Wir empfehlen jedoch die Verwendung der vorherigen Anmerkungen für neue Load Balancer anstelle von `service.beta.kubernetes.io/aws-load-balancer-type: "nlb-ip"`.

> **Important**
>
> Bearbeiten Sie die Anmerkungen nicht, nachdem Sie Ihren Service erstellt haben. Wenn Sie es ändern müssen, löschen Sie das Service-Objekt und erstellen es erneut mit dem gewünschten Wert für diese Anmerkung.

Instance targets

Der Load Balancer-Controller des AWS Cloud-Anbieters erstellt Network Load Balancer nur mit Instanzzielen. Version `2.2.0` und höher des AWS -Load-Balancing-Controllers erstellt Network Load Balancer auch mit Instance-Zielen. Wir empfehlen, ihn anstelle des Load Balancer-Controllers des AWS Cloud-Anbieters zu verwenden, um neue Network Load Balancer zu erstellen. Sie können Network Load Balancer-Instance-Ziele mit Pods verwenden, die auf Amazon-EC2-Knoten bereitgestellt werden, jedoch nicht in Fargate. Um den Netzwerkverkehr über Pods auszugleichen, die in Fargate bereitgestellt werden, müssen Sie IP-Ziele verwenden.

Um einen Network Load Balancer in einem privaten Subnetz bereitzustellen, muss Ihre Service-Spezifikation über die folgenden Anmerkungen verfügen. Sie können ein Beispiel-Service-Manifest mit den Anmerkungen anzeigen. Der `external` `aws-load-balancer-type` Wert für veranlasst den Load AWS Balancer Controller und nicht den Load Balancer Controller des AWS Cloud-Anbieters, den Network Load Balancer zu erstellen.

```
service.beta.kubernetes.io/aws-load-balancer-type: "external"
service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: "instance"
```

Network Load Balancer werden standardmäßig mit dem `internal` `aws-load-balancer-scheme` erstellt. Für interne Network Load Balancer muss Ihr Amazon-EKS-Cluster so konfiguriert werden, dass er mindestens ein privates Subnetz in Ihrer VPC verwendet. Kubernetes untersucht die Routing-Tabelle auf Ihre Subnetze, um zu identifizieren, ob sie öffentlich oder privat sind. Öffentliche Subnetze verfügen über einen direkten Zugang zum Internet über ein Internet-Gateway, nicht aber private Subnetze.

Wenn Sie einen Network Load Balancer in einem öffentlichen Subnetz erstellen möchten, um die Lastenverteilung auf Amazon-EC2-Knoten vorzunehmen, geben Sie `internet-facing` mit folgender Anmerkung an:

```
service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"
```

> **Important**
>
> Bearbeiten Sie die Anmerkungen nicht, nachdem Sie Ihren Service erstellt haben. Wenn Sie es ändern müssen, löschen Sie das Service-Objekt und erstellen es erneut mit dem gewünschten Wert für diese Anmerkung.

(Optional) Bereitstellen einer Beispielanwendung

Voraussetzungen

- Mindestens ein öffentliches oder privates Subnetz in Ihrer Cluster-VPC.
- Stellen Sie den AWS Load Balancer Controller in Ihrem Cluster bereit. Weitere Informationen finden Sie unter Was ist die AWS Load Balancer Controller?. Wir empfehlen Version `2.7.2` oder höher.

So stellen Sie eine Beispielanwendung bereit

1. Wenn Sie für Fargate bereitstellen, stellen Sie sicher, dass Sie ein verfügbares privates Subnetz in Ihrer VPC haben, und erstellen Sie ein Fargate-Profil. Wenn Sie keine Bereitstellung für Fargate durchführen, überspringen Sie diesen Schritt. Sie können das Profil erstellen, indem Sie den folgenden Befehl ausführen oder in AWS Management Console die gleichen Werte für `name` und `namespace` verwenden, die im Befehl enthalten sind. Ersetzen Sie das *`example values`* durch Ihr eigenes.

   ```
   eksctl create fargateprofile \
       --cluster my-cluster \
       --region region-code \
       --name nlb-sample-app \
       --namespace nlb-sample-app
   ```

2. Bereitstellen einer Beispielanwendung

   a. Erstellen Sie einen Namespace für die Anwendung.

   ```
   kubectl create namespace nlb-sample-app
   ```

   b. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *`sample-deployment`*`.yaml` auf Ihrem Computer.

   ```
   apiVersion: apps/v1
   kind: Deployment
   metadata:
     name: nlb-sample-app
     namespace: nlb-sample-app
   spec:
     replicas: 3
     selector:
       matchLabels:
         app: nginx
     template:
       metadata:
         labels:
           app: nginx
       spec:
         containers:
           - name: nginx
             image: public.ecr.aws/nginx/nginx:1.23
   ```

```
          ports:
            - name: tcp
              containerPort: 80
```

c. Wenden Sie die Manifestdatei auf Ihren Cluster an.

```
kubectl apply -f sample-deployment.yaml
```

3. Erstellen Sie einen Service mit einem zum Internet offenen Network Load Balancer, der Load Balancing auf IP-Ziele vornimmt.

a. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *sample-service*.yaml auf Ihrem Computer. Wenn Sie auf Fargate-Knoten bereitstellen, entfernen Sie die `service.beta.kubernetes.io/aws-load-balancer-scheme: internet-facing`-Zeile.

```
apiVersion: v1
kind: Service
metadata:
  name: nlb-sample-service
  namespace: nlb-sample-app
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-type: external
    service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: ip
    service.beta.kubernetes.io/aws-load-balancer-scheme: internet-facing
spec:
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
  type: LoadBalancer
  selector:
    app: nginx
```

b. Wenden Sie die Manifestdatei auf Ihren Cluster an.

```
kubectl apply -f sample-service.yaml
```

4. Stellen Sie sicher, dass der Service bereitgestellt wurde.

```
kubectl get svc nlb-sample-service -n nlb-sample-app
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME             TYPE           CLUSTER-IP         EXTERNAL-IP
                                                  PORT(S)        AGE
sample-service   LoadBalancer   10.100.240.137
 k8s-nlbsampl-nlbsampl-xxxxxxxxxx-xxxxxxxxxxxxxxxx.elb.region-code.amazonaws.com
  80:32400/TCP   16h
```

> **Note**
>
> Die Werte für *10.100.240.137* und *xxxxxxxxxx* - *xxxxxxxxxxxxxxxx* unterscheiden sich von der Beispielausgabe (sie gelten nur für Ihren Load Balancer), und *us-west-2* können für Sie unterschiedlich sein, je nachdem, in welchem Cluster sich Ihr Cluster befindet. AWS-Region

5. Öffnen Sie Amazon EC2 AWS Management Console. Wählen Sie Target Groups (Zielgruppen) unter Load Balancing (Lastausgleich) im linken Navigationsbereich aus. Wählen Sie in der Spalte Name den Namen der Zielgruppe aus, wobei der Wert in der Spalte Load Balancer einem Teil des Namens in der `EXTERNAL-IP`-Spalte der Ausgabe im vorherigen Schritt entspricht. Sie wählen beispielsweise die Zielgruppe `k8s-default-samplese-`*xxxxxxxxxx*, wenn Ihre Ausgabe mit der vorigen Ausgabe übereinstimmte. Der Zieltyp ist `IP`, weil dies im Beispiel-Manifest des Services angegeben wurde.
6. Wählen Sie Ihre Zielgruppe und danach die Registerkarte Ziele aus. Unter Registrierte Ziele sollten Sie drei IP-Adressen der drei Replikate sehen, die in einem vorherigen Schritt bereitgestellt wurden. Warten Sie, bis der Status aller Ziele fehlerfrei ist, bevor Sie fortfahren. Möglicherweise dauert es ein paar Minuten, bis alle Ziele `healthy` sind. Die Ziele können sich in einem `unhealthy`-Zustand befinden, bevor sie in einen `healthy`-Zustand wechseln.
7. Senden Sie Datenverkehr an den Dienst, indem Sie *xxxxxxxxxx-xxxxxxxxxxxxxxxx* und *us-west-2* durch die Werte ersetzen, die in der Ausgabe für einen vorherigen Schritt für `EXTERNAL-IP` zurückgegeben wurden. Wenn Sie in einem privaten Subnetz bereitgestellt haben, müssen Sie die Seite von einem Gerät in Ihrer VPC aus anzeigen, z. B. von einem Bastion-Host. Weitere Informationen finden Sie unter Linux-Bastion-Hosts in AWS.

```
curl k8s-default-samplese-xxxxxxxxxx-xxxxxxxxxxxxxxxx.elb.region-code.amazonaws.com
```

Eine Beispielausgabe sieht wie folgt aus.

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
[...]
```

8. Wenn Sie mit der Beispielbereitstellung, dem Dienst und Namespace fertig sind, entfernen Sie sie.

```
kubectl delete namespace nlb-sample-app
```

Application Load Balancing auf Amazon EKS

Wenn Sie einen erstellen Kubernetes`ingress`, wird ein AWS Application Load Balancer (ALB) bereitgestellt, der den Anwendungsdatenverkehr ausgleicht. Weitere Informationen hierzu finden Sie unter Was ist ein Application Load Balancer? im Benutzerhandbuch für Application Load Balancer und Ingress in der Kubernetes-Dokumentation. ALBs können mit Pods verwendet werden, die auf Knoten oder AWS Fargate bereitgestellt werden. Sie können einen ALB für öffentliche oder private Subnetze bereitstellen.

Der Anwendungsverkehr wird bei L7 des OSI-Modells ausgeglichen. Um den Netzwerkverkehr auf L4 auszugleichen, stellen Sie einen Kubernetes-`service` vom Typ `LoadBalancer` bereit. Dieser Typ stellt einen AWS Network Load Balancer bereit. Weitere Informationen finden Sie unter Network Load Balancing in Amazon EKS. Weitere Informationen zu den Unterschieden zwischen den beiden Load-Balancing-Arten finden Sie unter Elastic-Load-Balancing-Features auf der AWS -Website.

Voraussetzungen

Bevor Sie den Anwendungsdatenverkehr auf eine Anwendung verteilen können, müssen Sie die folgenden Anforderungen erfüllen.

- Einen vorhandenen Cluster haben. Falls Sie keinen bestehenden Cluster haben, siehe Erste Schritte mit Amazon EKS. Wenn Sie die Version eines vorhandenen Clusters aktualisieren müssen, finden Sie unter Aktualisieren einer Amazon-EKS-Cluster-Kubernetes-Version.
- Stellen Sie den AWS Load Balancer Controller in Ihrem Cluster bereit. Weitere Informationen finden Sie unter Was ist die AWS Load Balancer Controller?. Wir empfehlen Version `2.7.2` oder höher.

- Mindestens zwei Subnetze in verschiedenen Availability Zones. Der AWS Load Balancer Controller wählt ein Subnetz aus jeder Availability Zone aus. Wenn mehrere markierte Subnetze in einer Availability Zone gefunden werden, wählt der Controller das Subnetz aus, dessen Subnetz-ID lexikografisch an erster Stelle steht. Ein Subnetz muss jeweils mindestens acht verfügbare IP-Adressen aufweisen.

  Wenn Sie mehrere an den Worker-Knoten angehängte Sicherheitsgruppen verwenden, muss genau eine Sicherheitsgruppe wie folgt gekennzeichnet werden. Ersetzen Sie *my-cluster* durch Ihren Clusternamen.

  - Schlüssel – `kubernetes.io/cluster/`*my-cluster*
  - Wert – `shared` oder `owned`

- Wenn Sie die AWS Load Balancer Controller-Version `2.1.1` oder früher verwenden, müssen Subnetze im folgenden Format markiert werden. Wenn Sie Version `2.1.2` oder höher verwenden, ist die Markierung optional. Wir empfehlen jedoch, ein Subnetz zu markieren, falls einer der folgenden Fälle zutrifft. Sie haben mehrere Cluster, die in derselben VPC ausgeführt werden, oder haben mehrere AWS Dienste, die sich Subnetze in einer VPC teilen. Oder Sie möchten mehr Kontrolle darüber haben, wo Load Balancer für jeden Cluster bereitgestellt werden. Ersetzen Sie *my-cluster* durch Ihren Clusternamen.

  - Schlüssel – `kubernetes.io/cluster/`*my-cluster*
  - Wert – `shared` oder `owned`

- Ihre öffentlichen und privaten Subnetze müssen die folgenden Anforderungen erfüllen. Dies gilt, es sei denn, Sie geben Subnetz-IDs explizit als Anmerkung zu einem Dienst- oder Ingress-Objekt an. Angenommen, Sie stellen Load Balancer bereit, indem Sie Subnetz-IDs explizit als Anmerkung zu einem Service- oder Ingress-Objekt angeben. In dieser Situation verwenden Kubernetes und der AWS -Load-Balancer-Controller diese Subnetze direkt, um den Load Balancer zu erstellen, und die folgenden Tags sind nicht erforderlich.

  - Private Subnetze – Müssen im folgenden Format markiert sein. Auf diese Weise weiß der AWS Load Balancer-Controller, dass die Subnetze für interne Load Balancer verwendet werden können. Kubernetes Wenn Sie nach dem 26. März 2020 eine Amazon AWS CloudFormation EKS-Vorlage verwenden`eksctl`, um Ihre VPC zu erstellen, werden die Subnetze bei der Erstellung entsprechend gekennzeichnet. Weitere Informationen zu den Amazon AWS CloudFormation EKS-VPC-Vorlagen finden Sie unterErstellen einer VPC für Ihren Amazon-EKS-Cluster.

    - Schlüssel – `kubernetes.io/role/internal-elb`
    - Wert – 1

- Öffentliche Subnetze – Müssen im folgenden Format markiert sein. So erkennt Kubernetes, dass nur die Subnetze verwendbar sind, die für externe Load Balancer angegeben wurden. Auf diese Weise wählt Kubernetes nicht in jeder Availability Zone ein öffentliches Subnetz (lexikografisch basierend auf ihrer Subnetz-ID). Wenn Sie nach dem 26. März 2020 eine Amazon AWS CloudFormation EKS-Vorlage verwenden`eksctl`, um Ihre VPC zu erstellen, werden die Subnetze bei der Erstellung entsprechend gekennzeichnet. Weitere Informationen zu den Amazon AWS CloudFormation EKS-VPC-Vorlagen finden Sie unterErstellen einer VPC für Ihren Amazon-EKS-Cluster.
  - Schlüssel – `kubernetes.io/role/elb`
  - Wert – 1

  Wenn die Subnetz-Rollen-Tags nicht explizit hinzugefügt werden, untersucht der Kubernetes-Dienst-Controller die Routing-Tabelle Ihrer Cluster-VPC-Subnetze. Dadurch wird festgestellt, ob das Subnetz privat oder öffentlich ist. Wir empfehlen, sich nicht auf dieses Verhalten zu verlassen. Fügen Sie stattdessen explizit die Rollen-Tags privat oder öffentlich hinzu. Der AWS Load Balancer Controller untersucht keine Routing-Tabellen. Außerdem müssen die Tags privat und öffentlich für eine erfolgreiche automatische Erkennung vorhanden sein.

Überlegungen

- Der Load AWS Balancer Controller erstellt ALBs und die erforderlichen unterstützenden AWS Ressourcen, wenn eine Kubernetes Eingangsressource auf dem Cluster mit der Anmerkung erstellt wird. `kubernetes.io/ingress.class: alb` Die Ressource für eingehenden Datenverkehr konfiguriert den ALB so, dass HTTP- oder HTTPS-Datenverkehr an verschiedene Pods im Cluster weitergeleitet wird. Um sicherzustellen, dass Ihre Ingress-Objekte den AWS Load Balancer Controller verwenden, fügen Sie die folgende Anmerkung zu Ihrer Kubernetes-Spezifikation für eingehenden Datenverkehr hinzu. Weitere Informationen finden Sie unter Ingress-Spezifikation auf GitHub.

```
annotations:
    kubernetes.io/ingress.class: alb
```

> **Note**
>
> Wenn Sie Load Balancing für `IPv6` Pods verwenden, fügen Sie die folgende Anmerkung zu Ihrer Ingress-Spezifikation hinzu. Load Balancing über `IPv6` funktioniert nur auf IP-

Ziele, nicht auf Instance-Ziele. Ohne diese Anmerkung verläuft das Load Balancing über `IPv4`.

```
alb.ingress.kubernetes.io/ip-address-type: dualstack
```

- Der AWS Load Balancer Controller unterstützt die folgenden Datenverkehrsmodi.
  - Instance – Registriert Knoten in Ihrem Cluster als Ziele für den ALB. Der Datenverkehr, der den ALB erreicht, wird für Ihren Service an `NodePort` und dann an Ihre Pods weitergeleitet. Dies ist der standardmäßige Datenverkehrsmodus. Sie können ihn auch explizit mit der `alb.ingress.kubernetes.io/target-type: instance`-Anmerkung angeben.

    Note

    Ihr Kubernetes Dienst muss den Typ `NodePort` oder "LoadBalancer" angeben, um diesen Verkehrsmodus verwenden zu können.

  - IP – Registriert Pods als Ziele für den ALB. Der Datenverkehr, der den ALB erreicht, wird für Ihren Service direkt an Pods weitergeleitet. Sie müssen die `alb.ingress.kubernetes.io/target-type: ip`-Anmerkung angeben, um diesen Datenverkehrsmodus verwenden zu können. Der IP-Zieltyp ist erforderlich, wenn Ziel-Pods auf Fargate ausgeführt werden.
- Um vom Controller erstellte ALBs mit Tags zu versehen, fügen Sie dem Controller die folgende Anmerkung hinzu: `alb.ingress.kubernetes.io/tags`. Eine Liste aller verfügbaren Anmerkungen, die vom AWS Load Balancer Controller unterstützt werden, finden Sie unter Ingress-Anmerkungen auf GitHub.
- Ein Upgrade oder Downgrade der ALB-Controller-Version kann zu einer Unterbrechung der Abwärtskompatibilität für Features führen, die darauf angewiesen sind. Weitere Informationen zu den Breaking Changes, die in jeder Version eingeführt werden, finden Sie in den Versionshinweisen zum ALB-Controller auf GitHub.

So teilen Sie einen Application Load Balancer über mehrere Service-Ressourcen mit `IngressGroups`

Um einen Ingress einer Gruppe hinzuzufügen, fügen Sie die folgende Anmerkung zu einer Kubernetes-Ingress-Ressourcenspezifikation hinzu.

```
alb.ingress.kubernetes.io/group.name: my-group
```

Der Gruppenname muss:

- Eine Länge von 63 oder weniger Zeichen haben.
- Aus Kleinbuchstaben, Zahlen, `-` und `.` bestehen
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.

Der Controller führt automatisch Ingress-Regeln für alle Ingresses in derselben Ingress-Gruppe zusammen. Es unterstützt sie mit einem einzigen ALB. Die meisten Anmerkungen, die auf einem Ingress definiert sind, gelten nur für die von diesem Ingress definierten Pfade. Standardmäßig gehören Ingress-Ressourcen keiner Ingress-Gruppe an.

> **Warning**
>
> Potenzielles Sicherheitsrisiko: Geben Sie nur dann eine Ingress-Gruppe für einen Ingress an, wenn alle Kubernetes-Benutzer mit RBAC-Berechtigung zum Erstellen oder Ändern von Ingress-Ressourcen innerhalb derselben Vertrauensgrenze liegen. Wenn Sie die Anmerkung mit einem Gruppennamen hinzufügen, können andere Kubernetes-Benutzer ihre Ingresses erstellen oder ändern, sodass sie derselben Ingress-Gruppe angehören. Dies kann zu unerwünschtem Verhalten führen, z. B. zum Überschreiben vorhandener Regeln durch Regeln mit höherer Priorität.

Sie können eine Reihenfolgennummer zu Ihrer Ingress-Ressource hinzufügen.

```
alb.ingress.kubernetes.io/group.order: '10'
```

Die Zahl kann 1-1000 sein. Die niedrigste Zahl für alle Ingresse in derselben Ingress-Gruppe wird zuerst ausgewertet. Alle Ingresses ohne diese Anmerkung werden mit dem Wert null bewertet. Doppelte Regeln mit einer höheren Nummer können Regeln mit einer niedrigeren Nummer überschreiben. Standardmäßig wird die Regelreihenfolge zwischen Ingressen innerhalb derselben Ingress-Gruppe lexikografisch basierend auf Namespace und Name bestimmt.

> **Important**
>
> Stellen Sie sicher, dass jeder Ingress in derselben Ingress-Gruppe eine eindeutige Prioritätsnummer hat. In Ingresses dürfen keine doppelten Bestellnummern vorhanden sein.

(Optional) Bereitstellen einer Beispielanwendung

Voraussetzungen

- Mindestens ein öffentliches oder privates Subnetz in Ihrer Cluster-VPC.
- Stellen Sie den AWS Load Balancer Controller in Ihrem Cluster bereit. Weitere Informationen finden Sie unter Was ist die AWS Load Balancer Controller?. Wir empfehlen Version `2.7.2` oder höher.

Bereitstellen einer Beispielanwendung

Sie können die Beispielanwendung auf einem Cluster ausführen, der über Amazon-EC2-Knoten, Fargate-Pods oder beides verfügt.

1. Wenn Sie keine Bereitstellung für Fargate durchführen, überspringen Sie diesen Schritt. Wenn Sie Fargate bereitstellen, erstellen Sie ein Fargate-Profil. Sie können das Profil erstellen, indem Sie den folgenden Befehl ausführen oder in AWS Management Console die gleichen Werte für `name` und `namespace` verwenden, die im Befehl enthalten sind. Ersetzen Sie das *`example values`* durch Ihr eigenes.

   ```
   eksctl create fargateprofile \
       --cluster my-cluster \
       --region region-code \
       --name alb-sample-app \
       --namespace game-2048
   ```

2. Stellen Sie das Spiel 2048 als Beispielanwendung bereit, um zu überprüfen, ob das als Ergebnis des Eingangsobjekts ein AWS ALB AWS Load Balancer Controller erstellt. Führen Sie die Schritte für den Typ des Subnetzes aus, in dem Sie bereitstellen.

   a. Wenn Sie auf Pods in einem Cluster bereitstellen, den Sie mit der `IPv6`-Familie erstellt haben, fahren Sie mit dem nächsten Schritt fort.

      - Öffentlich

        ```
        kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/examples/2048/2048_full.yaml
        ```

      - Privat

1. Laden Sie das Manifest herunter.

```
curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/examples/2048/2048_full.yaml
```

2. Bearbeiten Sie die Datei und suchen Sie die Zeile mit der Aufschrift `alb.ingress.kubernetes.io/scheme: internet-facing`.
3. Ändern Sie *`internet-facing`* in **`internal`** und speichern Sie die Datei.
4. Wenden Sie das Manifest auf Ihren Cluster an.

```
kubectl apply -f 2048_full.yaml
```

b. Wenn Sie auf Pods in einem Cluster bereitstellen, den Sie mit der IPv6-Familie erstellt haben, fahren Sie mit den nächsten Schritten fort.

1. Laden Sie das Manifest herunter.

```
curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/examples/2048/2048_full.yaml
```

2. Öffnen Sie die Datei in einem Editor und fügen Sie die folgende Zeile zu den Anmerkungen in der Ingress-Spezifikation hinzu.

```
alb.ingress.kubernetes.io/ip-address-type: dualstack
```

3. Wenn Sie den Lastenausgleich auf interne Pods anstelle von Internet-Pods anwenden, ändern Sie die Zeile, die `alb.ingress.kubernetes.io/scheme: internet-facing` zu `alb.ingress.kubernetes.io/scheme: internal` angibt
4. Speichern Sie die Datei.
5. Wenden Sie das Manifest auf Ihren Cluster an.

```
kubectl apply -f 2048_full.yaml
```

3. Überprüfen Sie nach ein paar Minuten mit dem folgenden Befehl, ob die Ressource für eingehenden Datenverkehr erstellt wurde.

```
kubectl get ingress/ingress-2048 -n game-2048
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME           CLASS    HOSTS   ADDRESS
                       PORTS   AGE
ingress-2048   <none>   *       k8s-game2048-ingress2-xxxxxxxxxx-yyyyyyyyyy.region-
code.elb.amazonaws.com   80      2m32s
```

> **Note**
>
> Wenn Sie den Load Balancer in einem privaten Subnetz erstellt haben, wird dem Wert unter ADDRESS in der vorherigen Ausgabe `internal-` vorangestellt.
> Wenn Ihr Ingress nach einigen Minuten nicht erstellt wurde, führen Sie den folgenden Befehl aus, um die AWS Load Balancer Controller-Protokolle anzuzeigen. Diese Protokolle können Fehlermeldungen enthalten, mit denen Sie Probleme mit Ihrer Bereitstellung diagnostizieren können.
>
> ```
> kubectl logs -f -n kube-system -l app.kubernetes.io/instance=aws-load-
> balancer-controller
> ```

4. Wenn Sie in einem öffentlichen Subnetz bereitgestellt haben, öffnen Sie einen Browser, und navigieren Sie zur ADDRESS-URL aus der vorherigen Befehlsausgabe, um die Beispielanwendung anzuzeigen. Wenn nichts angezeigt wird, aktualisieren Sie Ihren Browser und versuchen Sie es erneut. Wenn Sie in einem privaten Subnetz bereitgestellt haben, müssen Sie die Seite von einem Gerät in Ihrer VPC aus anzeigen, z. B. von einem Bastion-Host. Weitere Informationen finden Sie unter Linux-Bastion-Hosts in AWS.
5. Wenn Sie mit dem Experimentieren mit Ihrer Beispielanwendung fertig sind, löschen Sie sie, indem Sie einen der folgenden Befehle ausführen.

   - Wenn Sie das Manifest angewendet haben, anstatt eine heruntergeladene Kopie anzuwenden, verwenden Sie den folgenden Befehl.

     ```
     kubectl delete -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-
     balancer-controller/v2.7.2/docs/examples/2048/2048_full.yaml
     ```

   - Wenn Sie das Manifest heruntergeladen und bearbeitet haben, verwenden Sie den folgenden Befehl.

     ```
     kubectl delete -f 2048_full.yaml
     ```

Einschränken externer IP-Adressen, die Services zugewiesen werden können

Kubernetes-Services können von innerhalb eines Clusters aus erreicht werden durch:

- Eine Cluster-IP-Adresse, die automatisch von Kubernetes zugewiesen wird
- Jede IP-Adresse, die Sie für die `externalIPs`-Eigenschaft in einer Service-Spezifikation festlegen. Externe IP-Adressen werden nicht von Kubernetes verwaltet und liegen in der Verantwortung des Cluster-Administrators. Externe IP-Adressen, die mit `externalIPs` festgelegt sind, unterscheiden sich von der externen IP-Adresse, die einem Service vom Typ `LoadBalancer` von einem Cloud-Anbieter zugewiesen ist.

Für weitere Informationen über Kubernetes Service, siehe Service in der Kubernetes-Dokumentation. Sie können die IP-Adressen einschränken, die für `externalIPs` in einer Service-Spezifikation festgelegt werden können.

Um die IP-Adressen einzuschränken, die für **`externalIPs`** in einer Service-Spezifikation festgelegt werden können

1. Stellen Sie `cert-manager` bereit, um Webhook-Zertifikate zu verwalten. Weitere Informationen finden Sie in der Dokumentation zu `cert-manager`.

   ```
   kubectl apply -f https://github.com/jetstack/cert-manager/releases/download/v1.5.4/cert-manager.yaml
   ```

2. Stellen Sie sicher, dass die `cert-manager`-Pods ausgeführt werden.

   ```
   kubectl get pods -n cert-manager
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   NAME                                       READY   STATUS    RESTARTS   AGE
   cert-manager-58c8844bb8-nlx7q              1/1     Running   0          15s
   cert-manager-cainjector-745768f6ff-696h5   1/1     Running   0          15s
   cert-manager-webhook-67cc76975b-4v4nk      1/1     Running   0          14s
   ```

3. Überprüfen Sie Ihre vorhandenen Services, um sicherzustellen, dass keiner von ihnen externe IP-Adressen zugewiesen sind, die nicht in dem CIDR-Block enthalten sind, auf den Sie Adressen beschränken möchten.

```
kubectl get services -A
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAMESPACE                      NAME                                      TYPE
   CLUSTER-IP       EXTERNAL-IP     PORT(S)         AGE
cert-manager                   cert-manager                              ClusterIP
   10.100.102.137   <none>          9402/TCP        20m
cert-manager                   cert-manager-webhook                      ClusterIP
   10.100.6.136     <none>          443/TCP         20m
default                        kubernetes                                ClusterIP
   10.100.0.1       <none>          443/TCP         2d1h
externalip-validation-system   externalip-validation-webhook-service     ClusterIP
   10.100.234.179   <none>          443/TCP         16s
kube-system                    kube-dns                                  ClusterIP
   10.100.0.10      <none>          53/UDP,53/TCP   2d1h
my-namespace                   my-service                                ClusterIP
   10.100.128.10    192.168.1.1     80/TCP          149m
```

Wenn es sich bei einem der Werte um IP-Adressen handelt, die sich nicht innerhalb des Blocks befinden, auf den Sie den Zugriff einschränken möchten, müssen Sie die Adressen so ändern, dass sie sich innerhalb des Blocks befinden, und die Services erneut bereitstellen. Zum Beispiel, der `my-service` Service in der vorherigen Ausgabe hat eine externe IP-Adresse zugewiesen, die sich nicht innerhalb des CIDR-Blockbeispiels in Schritt 5 befindet.

4. Laden Sie das externe IP-Webhook-Manifest herunter. Sie können auch den Quellcode für Webhook auf GitHub ansehen.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/docs/externalip-webhook.yaml
```

5. CIDR-Blöcke angeben. Öffnen Sie die heruntergeladene Datei in Ihrem Editor und entfernen Sie # beim Start der folgenden Zeilen.

```
#args:
#- --allowed-external-ip-cidrs=10.0.0.0/8
```

Ersetzen Sie `10.0.0.0/8` mit Ihrem eigenen CIDR-Block. Sie können so viele Blöcke wie gewünscht festlegen. Wenn Sie mehrere Blöcke angeben, fügen Sie ein Komma zwischen Blöcken hinzu.

6. Wenn sich Ihr Cluster nicht in der AWS-Region `us-west-2` befindet, dann ersetzen Sie `us-west-2`, `602401143452` und `amazonaws.com` in der Datei mit den folgenden Befehlen. Bevor Sie die Befehle ausführen, ersetzen Sie *`region-code`* und *`111122223333`* mit dem Wert für Ihre AWS-Region aus der Liste in Registrierungen für Amazon-Container-Images.

```
sed -i.bak -e 's|602401143452|111122223333|' externalip-webhook.yaml
sed -i.bak -e 's|us-west-2|region-code|' externalip-webhook.yaml
sed -i.bak -e 's|amazonaws.com||' externalip-webhook.yaml
```

7. Wenden Sie das Manifest auf Ihren Cluster an.

```
kubectl apply -f externalip-webhook.yaml
```

Ein Versuch, einen Service auf Ihrem Cluster mit einer IP-Adresse bereitzustellen, die für `externalIPs` festgelegt ist, die nicht in den Blöcken enthalten ist, die Sie im Schritt CIDR-Blöcke angeben angegeben haben, schlägt fehl.

Kopieren eines Container-Images von einem Repository in ein anderes

In diesem Thema wird beschrieben, wie Sie ein Container-Image aus einem Repository abrufen, auf das Ihre Knoten keinen Zugriff haben, und in ein Repository verschieben, auf das sie Zugriff haben. Das Image können Sie in Amazon ECR oder ein alternatives Repository verschieben, auf das die Knoten Zugriff haben.

Voraussetzungen

- Die Docker-Engine ist auf Ihrem Computer installiert und konfiguriert. Weitere Informationen finden Sie unter Installieren der Docker-Engine in der Docker-Dokumentation.
- Version `2.12.3` oder höher oder Version `1.27.160` oder höher der AWS Command Line Interface (AWS CLI) auf Ihrem Gerät oder in AWS CloudShell installiert und konfiguriert. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere

Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface-Benutzerhandbuch. Die in AWS CloudShell installierte AWS CLI-Version kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zum Aktualisieren der Version finden Sie unter Installieren von AWS CLI im Stammverzeichnis im AWS CloudShell-Benutzerhandbuch.

- Ein Schnittstellen-VPC-Endpunkt für Amazon ECR, wenn die Nodes Container-Images aus einem privaten Amazon-ECR-Repository über das Amazon-Netzwerk abrufen bzw. in ein solches verschieben sollen. Weitere Informationen finden Sie unter Erstellen der VPC Endpunkte für Amazon ECR im Benutzerhandbuch von Amazon Elastic Container Registry.

Gehen Sie wie folgt vor, um ein Container-Image aus einem Repository abzurufen und es in ein eigenes Repository zu verschieben. In den folgenden Beispielen in diesem Thema wird das Image für das Metrikhelferobjekt von Amazon VPC CNI plugin for Kubernetes abgerufen. Ersetzen Sie beim Ausführen dieser Schritte die *example values* unbedingt durch eigene Werte.

So kopieren Sie ein Container-Image aus einem Repository in ein anderes

1. Wenn Sie noch nicht über ein Amazon-ECR-Repository oder ein anderes Repository verfügen, erstellen Sie eines, auf das die Knoten zugreifen können. Mit dem folgenden Befehl wird ein privates Amazon-ECR-Repository erstellt. Der Name eines privaten Amazon-ECR-Repositorys muss mit einem Buchstaben beginnen. Er darf nur Kleinbuchstaben, Zahlen, Bindestriche (-), Unterstriche (_) und Schrägstriche (/) enthalten. Weitere Informationen finden Sie unter Erstellen eines privaten Repositorys im Benutzerhandbuch von Amazon Elastic Container Registry.

   Sie können *cni-metrics-helper* mit einem beliebigen Namen ersetzen, den Sie wählen. Erstellen Sie – als bewährte Methode – ein separates Repository für jedes Image. Das wird empfohlen, weil Image-Tags in einem Repository eindeutig sein müssen. Ersetzen Sie *region-code* durch eine von Amazon ECR unterstützte AWS-Region.

   ```
   aws ecr create-repository --region region-code --repository-name cni-metrics-helper
   ```

2. Bestimmen Sie die Registrierung, das Repository und das Tag (optional) des Images, das die Knoten abrufen müssen. Diese Informationen liegen im Format `registry/repository[:tag]` vor.

   Bei vielen der Amazon-EKS-Themen zur Installation von Images müssen Sie eine Manifestdatei anwenden oder das Image mithilfe eines Helm-Charts installieren. Bevor Sie eine Manifestdatei

anwenden oder ein Helm-Chart installieren, sollten Sie sich jedoch zunächst den Inhalt des Manifests oder der Datei `values.yaml` des Charts ansehen. So können Sie die Registrierung, das Repository und das Tag für den Abrufvorgang bestimmen.

Die folgende Zeile beispielsweise finden Sie in der Manifestdatei für das Metrikhelferobjekt von Amazon VPC CNI plugin for Kubernetes. Die Registrierung ist `602401143452.dkr.ecr.us-west-2.amazonaws.com`, eine private Amazon-ECR-Registrierung. Das Repository ist `cni-metrics-helper`.

```
image: "602401143452.dkr.ecr.us-west-2.amazonaws.com/cni-metrics-helper:v1.12.6"
```

Die folgenden Variationen sind bei einem Image-Speicherort möglich:

- Nur `repository-name:tag`. In diesem Fall ist in der Regel `docker.io` die Registrierung, aber nicht spezifiziert, da Kubernetes sie standardmäßig einem Repository-Namen voranstellt, wenn keine Registrierung angegeben ist.
- `repository-name/repository-namespace/repository:tag`. Ein Repository-Namespace ist optional, wird jedoch manchmal vom Repository-Besitzer zum Kategorisieren von Images angegeben. Alle Amazon-EC2-Images in der Amazon ECR Public Gallery nutzen beispielsweise den Namespace `aws-ec2`.

Zeigen Sie vor dem Installieren eines Images mit Helm die Helm-Datei `values.yaml` an, um den Image-Speicherort zu bestimmen. Die Datei `values.yaml` für das Metrikhelferobjekt von Amazon VPC CNI plugin for Kubernetes enthält zum Beispiel die folgenden Zeilen.

```
image:
  region: us-west-2
  tag: v1.12.6
  account: "602401143452"
  domain: "amazonaws.com"
```

3. Rufen Sie das in der Manifestdatei angegebene Container-Image ab.

   a. Beim Abruf aus einer öffentlichen Registrierung wie zum Beispiel Amazon ECR Public Gallery können Sie zum nächsten Unterschritt springen, da keine Authentifizierung erforderlich ist. In diesem Beispiel authentifizieren Sie sich bei einer privaten Amazon-ECR-Registrierung, die das Repository für das Helper-Image für CNI-Kennzahlen enthält. Amazon EKS verwaltet das Image in jeder Registrierung, die in Registrierungen für Amazon-Container-

Images aufgeführt ist. Die Authentifizierung ist bei jeder der Registrierungen möglich. Dazu ersetzen Sie *602401143452* und *region-code* durch die Informationen für eine andere Registrierung. Für jede AWS-Region, in der Amazon EKS unterstützt wird, existiert eine separate Registrierung.

```
aws ecr get-login-password --region region-code | docker login --username AWS --
password-stdin 602401143452.dkr.ecr.region-code.amazonaws.com
```

b. Rufen Sie das Image ab. In diesem Beispiel erfolgt der Abruf aus der Registrierung, bei der Sie sich im letzten Unterschritt authentifiziert haben. Ersetzen Sie *602401143452* und *region-code* durch die Informationen, die Sie im vorherigen Unterschritt angegeben haben.

```
docker pull 602401143452.dkr.ecr.region-code.amazonaws.com/cni-metrics-
helper:v1.12.6
```

4. Markieren Sie das abgerufene Image mit Ihrer Registrierung, Ihrem Repository und Ihrem Tag. Im folgenden Beispiel wird davon ausgegangen, dass Sie das Image über die Manifestdatei abgerufen haben und es in das private Amazon-ECR-Repository verschieben, das Sie im ersten Schritt erstellt haben. Ersetzen Sie *111122223333* durch Ihre Konto-ID. Ersetzen Sie *region-code* mit der AWS-Region, in der Sie Ihr privates Amazon-ECR-Pflichtfeld erstellt haben.

```
docker tag cni-metrics-helper:v1.12.6 111122223333.dkr.ecr.region-
code.amazonaws.com/cni-metrics-helper:v1.12.6
```

5. Authentifizieren Sie sich bei Ihrer Registrierung. In diesem Beispiel authentifizieren Sie sich bei der privaten Amazon-ECR-Registrierung, die Sie im ersten Schritt erstellt haben. Weitere Informationen finden Sie unter Registrierungsauthentifizierung im Benutzerhandbuch von Amazon Elastic Container Registry.

```
aws ecr get-login-password --region region-code | docker login --username AWS --
password-stdin 111122223333.dkr.ecr.region-code.amazonaws.com
```

6. Verschieben Sie das Image in Ihr Repository. In diesem Beispiel verschieben Sie das Image in das private Amazon-ECR-Repository, das Sie im ersten Schritt erstellt haben. Weitere Informationen finden Sie unter Verschieben eines Docker-Images im Benutzerhandbuch von Amazon Elastic Container Registry.

```
docker push 111122223333.dkr.ecr.region-code.amazonaws.com/cni-metrics-
helper:v1.12.6
```

7. Aktualisieren Sie die Manifestdatei, mit deren Hilfe Sie das Image in einem vorherigen Schritt bestimmt haben, mit den Werten `registry/repository:tag` für das verschobene Image. Bei der Installation mit einem Helm-Chart gibt es oft eine Option zur Angabe der Werte `registry/repository:tag`. Geben Sie bei der Installation des Charts die Werte `registry/repository:tag` für das Image an, das Sie in das Repository verschoben haben.

Registrierungen für Amazon-Container-Images

Wenn Sie AWSAmazon-EKS-Add-Ons in Ihrem Cluster bereitstellen, rufen Ihre Knoten die erforderlichen Container-Images aus der Registrierung ab, die im Installationsmechanismus für das Add-On angegeben ist, z. B. ein Installationsmanifest oder eine `values.yaml`-Helm-Datei. Die Images werden aus einem privaten Amazon-ECR-Repository abgerufen. Amazon EKS repliziert die Images in jeder von Amazon EKS unterstützten AWS-Region in ein Repository. Die Knoten können das Container-Image über das Internet aus einer der folgenden Registrierungen abrufen. Alternativ können die Knoten das Image über das Amazon-Netzwerk abrufen, wenn Sie einen Schnittstellen-VPC-Endpunkt für Amazon ECR (AWS PrivateLink) in Ihrer VPC erstellt haben. Für die Registrierungen ist eine Authentifizierung bei einem AWS-IAM-Konto erforderlich. Die Knoten authentifizieren sich mit der Amazon-EKS-Knoten-IAM-Rolle, die über die Berechtigungen in der von AmazonEC2ContainerRegistryReadOnly verwalteten IAM-Richtlinie verfügt.

AWS-Region	Registrierung
af-south-1	877085696533.dkr.ecr.af-south-1.amazonaws.com
ap-east-1	800184023465.dkr.ecr.ap-east-1.amazonaws.com
ap-northeast-1	602401143452.dkr.ecr.ap-northeast-1.amazonaws.com
ap-northeast-2	602401143452.dkr.ecr.ap-northeast-2.amazonaws.com
ap-northeast-3	602401143452.dkr.ecr.ap-northeast-3.amazonaws.com

AWS-Region	Registrierung
ap-south-1	602401143452.dkr.ecr.ap-south-1.amazonaws.com
ap-south-2	900889452093.dkr.ecr.ap-south-2.amazonaws.com
ap-southeast-1	602401143452.dkr.ecr.ap-southeast-1.amazonaws.com
ap-southeast-2	602401143452.dkr.ecr.ap-southeast-2.amazonaws.com
ap-southeast-3	296578399912.dkr.ecr.ap-southeast-3.amazonaws.com
ap-southeast-4	491585149902.dkr.ecr.ap-southeast-4.amazonaws.com
ca-central-1	602401143452.dkr.ecr.ca-central-1.amazonaws.com
ca-west-1	761377655185.dkr.ecr.ca-west-1.amazonaws.com
cn-north-1	918309763551.dkr.ecr.cn-north-1.amazonaws.com.cn
cn-northwest-1	961992271922.dkr.ecr.cn-northwest-1.amazonaws.com.cn
eu-central-1	602401143452.dkr.ecr.eu-central-1.amazonaws.com
eu-central-2	900612956339.dkr.ecr.eu-central-2.amazonaws.com
eu-north-1	602401143452.dkr.ecr.eu-north-1.amazonaws.com

AWS-Region	Registrierung
eu-south-1	590381155156.dkr.ecr.eu-south-1.amazonaws.com
eu-south-2	455263428931.dkr.ecr.eu-south-2.amazonaws.com
eu-west-1	602401143452.dkr.ecr.eu-west-1.amazonaws.com
eu-west-2	602401143452.dkr.ecr.eu-west-2.amazonaws.com
eu-west-3	602401143452.dkr.ecr.eu-west-3.amazonaws.com
il-central-1	066635153087.dkr.ecr.il-central-1.amazonaws.com
me-south-1	558608220178.dkr.ecr.me-south-1.amazonaws.com
me-central-1	759879836304.dkr.ecr.me-central-1.amazonaws.com
sa-east-1	602401143452.dkr.ecr.sa-east-1.amazonaws.com
us-east-1	602401143452.dkr.ecr.us-east-1.amazonaws.com
us-east-2	602401143452.dkr.ecr.us-east-2.amazonaws.com
us-gov-east-1	151742754352.dkr.ecr.us-gov-east-1.amazonaws.com
us-gov-west-1	013241004608.dkr.ecr.us-gov-west-1.amazonaws.com

AWS-Region	Registrierung
us-west-1	602401143452.dkr.ecr.us-west-1.amazonaws.com
us-west-2	602401143452.dkr.ecr.us-west-2.amazonaws.com

Amazon-EKS-Add-ons

Ein Add-on ist Software, die unterstützende Betriebsfunktionen für Kubernetes-Anwendungen bereitstellt, aber nicht anwendungsspezifisch ist. Dazu gehören Software wie Observability-Agents oder Kubernetes-Treiber, die es dem Cluster ermöglichen, mit den zugrunde liegenden AWS -Ressourcen für Netzwerke, Datenverarbeitung und Speicher zu interagieren. Zusatzsoftware wird in der Regel von der Kubernetes Community, Cloud-Anbietern oder AWS Drittanbietern entwickelt und verwaltet. Amazon EKS installiert automatisch selbstverwaltete Add-ons wie Amazon VPC CNI plugin for Kubernetes, `kube-proxy` und CoreDNS für jeden Cluster. Sie können die Standardkonfiguration der Add-ons ändern und sie bei Bedarf aktualisieren.

Amazon-EKS-Add-ons bieten die Installation und Verwaltung eines ausgewählten Satzes von Add-ons für Amazon-EKS-Cluster. Alle Amazon EKS-Add-Ons enthalten die neuesten Sicherheitspatches und Bugfixes und sind für die Verwendung AWS mit Amazon EKS validiert. Mit Amazon-EKS-Add-ons können Sie konsequent sicherstellen, dass Ihre Amazon-EKS-Cluster sicher und stabil sind und den Arbeitsaufwand reduzieren, den Sie für die Installation, Konfiguration und Aktualisierung von Add-ons benötigen. Wenn ein selbstverwaltetes Add-on wie `kube-proxy` bereits auf Ihrem Cluster ausgeführt wird und als Amazon-EKS-Add-on verfügbar ist, können Sie das Amazon-EKS-Add-on `kube-proxy` installieren, um die Vorteile der Funktionen von Amazon-EKS-Add-ons zu nutzen.

Sie können bestimmte von Amazon EKS verwaltete Konfigurationsfelder für Amazon-EKS-Add-ons über die Amazon EKS API aktualisieren. Sie können auch Konfigurationsfelder, die nicht von Amazon EKS verwaltet werden, direkt im Kubernetes-Cluster ändern, sobald das Add-on gestartet wird. Dazu gehört das Definieren spezifischer Konfigurationsfelder für ein Add-on, falls zutreffend. Diese Änderungen werden von Amazon EKS nicht überschrieben, sobald sie vorgenommen wurden. Dies ist möglich mit dem serverseitigen Anwendungsfeature von Kubernetes. Weitere Informationen finden Sie unter Kubernetes-Feldverwaltung.

Sie können Amazon-EKS-Add-Ons mit allen Amazon-EKS-Knotentypen verwenden.

Überlegungen

- Um Add-ons für den Cluster zu konfigurieren, muss der IAM–Prinzipal über IAM-Berechtigungen verfügen, um mit Add-ons zu arbeiten. Weitere Informationen finden Sie in den Aktionen mit Addon in ihrem Namen in Von Amazon Elastic Kubernetes Service definierte Aktionen.
- Amazon-EKS-Add-ons werden auf den Knoten ausgeführt, die Sie für Ihren Cluster bereitstellen oder konfigurieren. Knotentypen umfassen Amazon-EC2-Instances und Fargate.
- Sie können Felder ändern, die nicht von Amazon EKS verwaltet werden, um die Installation eines Amazon-EKS-Add-ons anzupassen. Weitere Informationen finden Sie unter Kubernetes-Feldverwaltung.
- Wenn Sie einen Cluster mit den AWS Management Console Add-Ons erstellen `kube-proxy` Amazon VPC CNI plugin for Kubernetes, werden CoreDNS Amazon EKS und Amazon EKS Ihrem Cluster automatisch hinzugefügt. Wenn Sie `config` verwenden, um Ihren Cluster mit einer `eksctl`-Datei zu erstellen, kann `eksctl`den Cluster auch mit Amazon-EKS-Add-ons erstellen. Wenn Sie einen Cluster mit `eksctl` ohne eine `config`-Datei oder mit einem anderen Tool erstellen, werden die selbstverwalteten `kube-proxy`, Amazon VPC CNI plugin for Kubernetes und CoreDNS-Add-ons anstelle der Amazon-EKS-Add-ons installiert. Sie können sie entweder selbst verwalten oder die Amazon-EKS-Add-ons nach der Clustererstellung manuell hinzufügen.
- Die `eks:addon-cluster-admin` `ClusterRoleBinding` verknüpft das `cluster-admin` `ClusterRole` mit der `eks:addon-manager` Kubernetes-Identität. Die Rolle verfügt über die erforderlichen Berechtigungen für die `eks:addon-manager`-Identität, um Kubernetes-Namespaces zu erstellen und Add-Ons in Namespaces zu installieren. Wenn `eks:addon-cluster-admin` `ClusterRoleBinding` entfernt wird, funktioniert der Amazon-EKS-Cluster weiterhin, Amazon EKS kann jedoch keine Add-Ons mehr verwalten. Alle Cluster ab den folgenden Plattformversionen verwenden das neue `ClusterRoleBinding`.

Kubernetes-Version	EKS-Plattformversionen
1.25	eks.2
1.24	eks.4
1.23	eks.5

EKS Kubernete s-Platt form-Versi onen	
eksbuild.235	
eksbuild.243	

Sie können Amazon EKS-Add-Ons mithilfe der Amazon EKS-API,, und hinzufügen AWS Management Console AWS CLI, aktualisieren oder löschen`eksctl`. Weitere Informationen finden Sie unter Verwalten von Amazon EKS-Add-Ons. Sie können Amazon-EKS-Add-ons auch mit AWS CloudFormation erstellen.

Verfügbare Amazon-EKS-Add-Ons von Amazon EKS

Die folgenden Amazon-EKS-Add-Ons können in Ihrem Cluster erstellt werden. Sie können jederzeit die aktuelle Liste der verfügbaren Add-Ons mit `eksctl` AWS Management Console, dem oder dem einsehen AWS CLI. Informationen zu allen verfügbaren Add-Ons oder zur Installation eines Add-Ons finden Sie unter Erstellen eines Add-Ons. Wenn für ein Add-On IAM-Berechtigungen erforderlich sind, müssen Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen verfügen oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster. Sie können ein Add-On aktualisieren oder löschen, sobald Sie es installiert haben.

Wählen Sie ein Add-on aus, um mehr darüber und seine Installationsanforderungen zu erfahren.

Amazon VPC CNI plugin for Kubernetes

- Name (Name – `vpc-cni`
- Beschreibung – Ein Kubernetes Container Network Interface (CNI)-Plugin, das native VPC-Netzwerke für Ihren Cluster bereitstellt. Der selbstverwaltete oder verwaltete Typ dieses Add-Ons ist standardmäßig auf jedem Amazon-EC2-Knoten installiert.
- Erforderliche IAM-Berechtigungen: Dieses Add-On nutzt die Funktion IAM-Rollen für Servicekonten von Amazon EKS. Wenn Ihr Cluster die `IPv4`-Familie verwendet, sind die Berechtigungen in der AmazonEKS_CNI_Policy erforderlich. Wenn Ihr Cluster die `IPv6`-Familie verwendet, müssen die Berechtigungen einer von Ihnen erstellten IAM-Richtlinie im IPv6-Modus hinzugefügt werden.

Sie können eine IAM-Rolle erstellen, ihr eine der Richtlinien anhängen und das vom Add-On verwendete Kubernetes-Servicekonto mit dem folgenden Befehl kommentieren.

Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters und *AmazonEKSVPCCNIRole* durch den Namen Ihrer Rolle. Wenn der Cluster die IPv6-Familie verwendet, ersetzen Sie *AmazonEKS_CNI_Policy* durch den Namen der Richtlinie, die Sie erstellt haben. Dieser Befehl erfordert, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool verwenden müssen, um die Rolle zu erstellen, ihr die Richtlinie zuzuordnen und das Kubernetes-Servicekonto mit Anmerkungen zu versehen, siehe Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.

```
eksctl create iamserviceaccount --name aws-node --namespace kube-system --cluster my-cluster --role-name AmazonEKSVPCCNIRole \
    --role-only --attach-policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy --approve
```

- Zusätzliche Informationen — Weitere Informationen zu den konfigurierbaren Einstellungen des Add-ons finden Sie unter aws-vpc-cni-k8s. GitHub Weitere Informationen zum Plugin finden Sie unter Vorschlag: CNI-Plugin für Kubernetes Netzwerke über AWS VPC. Weitere Informationen zum Erstellen eines Add-ons finden Sie unter Add-on vom Typ Amazon EKS erstellen.
- Update-Informationen – Sie können jeweils nur eine Nebenversion aktualisieren. Wenn Ihre aktuelle Version beispielsweise `1.28.`*x*`-eksbuild.`*y* ist und Sie auf `1.30.`*x*`-eksbuild.`*y* aktualisieren möchten, müssen Sie zuerst ihre aktuelle Version auf `1.29.`*x*`-eksbuild.`*y* und dann auf `1.30.`*x*`-eksbuild.`*y* aktualisieren. Weitere Informationen zum Aktualisieren des Add-ons finden Sie unter Aktualisieren des Amazon-EKS-Add-ons.

CoreDNS

- Name (Name – `coredns`
- Beschreibung – ein flexibler, erweiterbarer DNS-Server, der als Kubernetes-Cluster-DNS dienen kann. Der selbstverwaltete oder verwaltete Typ dieses Add-Ons wurde standardmäßig installiert, als Sie Ihren Cluster erstellt haben. Wenn Sie einen Amazon-EKS-Cluster mit mindestens einem Knoten starten, werden standardmäßig zwei Replikate des CoreDNS-Image bereitgestellt, unabhängig von der Anzahl der in Ihrem Cluster bereitgestellten Knoten. Die CoreDNS-Pods bieten die Namensauflösung für alle Pods im Cluster. Sie können CoreDNS-Pods auf Fargate-Knoten bereitstellen, wenn Ihr Cluster ein AWS Fargate Profil mit einem Namespace enthält, der dem Namespace für die CoreDNS-`deployment` entspricht.

- Erforderliche IAM-Berechtigungen – Für dieses Add-On sind keine Berechtigungen erforderlich.
- Zusätzliche Informationen – Weitere Informationen zu CoreDNS finden Sie in der Kubernetes-Dokumentation unter Verwenden von CoreDNS für die Serviceerkennung and Anpassen des DNS-Services.

`Kube-proxy`

- Name (Name – `kube-proxy`
- Beschreibung – verwaltet Netzwerkregeln auf jedem Amazon-EC2-Knoten. Es ermöglicht die Netzwerkkommunikation zu Ihren Pods. Der selbstverwaltete oder verwaltete Typ dieses Add-Ons wird standardmäßig auf jedem Amazon-EC2-Knoten in Ihrem Cluster installiert.
- Erforderliche IAM-Berechtigungen – Für dieses Add-On sind keine Berechtigungen erforderlich.
- Zusätzliche Informationen – Weitere Informationen zu `kube-proxy` finden Sie unter `kube-proxy` in der Kubernetes-Dokumentation.
- Update-Informationen – Bevor Sie Ihre aktuelle Version aktualisieren, sollten Sie die folgenden Anforderungen berücksichtigen:
  - `Kube-proxy` auf einem Amazon-EKS-Cluster verfügt über die gleiche Kompatibilitäts- und Skew-Richtlinie wie Kubernetes.
  - `Kube-proxy` muss dieselbe Nebenversion wie `kubelet` auf Ihren Amazon-EC2-Knoten sein.
  - `Kube-proxy` darf nicht höher als die Nebenversion Ihrer Cluster-Steuerebene sein.
  - Die `kube-proxy`-Version auf Ihren Amazon EC2-Knoten darf nicht älter als zwei Nebenversionen Ihrer Steuerebene sein. Wenn auf Ihrer Steuerungsebene beispielsweise Kubernetes 1.30 ausgeführt wird, darf die `kube-proxy` Nebenversion nicht vor 1.28 sein.
  - Wenn Sie Ihren Cluster kürzlich auf eine neue Kubernetes-Nebenversion aktualisiert haben, aktualisieren Sie Ihre Amazon-EC2-Knoten auf dieselbe Nebenversion, bevor Sie `kube-proxy` auf dieselbe Nebenversion wie Ihre Knoten aktualisieren.

Amazon-EBS-CSI-Treiber

- Name (Name – `aws-ebs-csi-driver`
- Beschreibung – Ein Kubernetes-Container-Speicherschnittstellen-Plugin, das Amazon-EBS-Speicher für Ihren Cluster bereitstellt.
- Erforderliche IAM-Berechtigungen: Dieses Add-On nutzt die Funktion IAM-Rollen für Servicekonten von Amazon EKS. Die Berechtigungen in der `AmazonEBSCSIDriverPolicy` AWS verwalteten

Richtlinie sind erforderlich. Sie können eine IAM-Rolle erstellen und ihr die verwaltete Richtlinie mit dem folgenden Befehl anfügen. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters und *AmazonEKS_EBS_CSI_DriverRole* durch den Namen Ihrer Rolle. Dieser Befehl erfordert, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool oder einen benutzerdefinierten KMS-Schlüssel für die Verschlüsselung verwenden müssen, finden Sie weitere Informationen unter Erstellen der IAM-Rolle des Amazon-EBS-CSI-Treibers.

```
eksctl create iamserviceaccount \
    --name ebs-csi-controller-sa \
    --namespace kube-system \
    --cluster my-cluster \
    --role-name AmazonEKS_EBS_CSI_DriverRole \
    --role-only \
    --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy
 \
    --approve
```

- Zusätzliche Informationen – Weitere Informationen zum Add-On finden Sie unter Amazon-EBS-CSI-Treiber.

Amazon EFS-CSI-Treiber

- Name (Name – `aws-efs-csi-driver`
- Beschreibung – Ein Kubernetes-Container-Speicherschnittstellen-Plugin (CSI), das Amazon-EFS-Speicher für Ihren Cluster bereitstellt.
- Erforderliche IAM-Berechtigungen: Dieses Add-On nutzt die Funktion IAM-Rollen für Servicekonten von Amazon EKS. Die Berechtigungen in der `AmazonEFSCSIDriverPolicy` AWS verwalteten Richtlinie sind erforderlich. Sie können eine IAM-Rolle erstellen und ihr die verwaltete Richtlinie mit dem folgenden Befehl anfügen. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters und *AmazonEKS_EFS_CSI_DriverRole* durch den Namen Ihrer Rolle. Diese Befehle erfordern, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool verwenden müssen, finden Sie Informationen unter Erstellen einer IAM-Rolle.

```
export cluster_name=my-cluster
export role_name=AmazonEKS_EFS_CSI_DriverRole
eksctl create iamserviceaccount \
    --name efs-csi-controller-sa \
    --namespace kube-system \
    --cluster $cluster_name \
```

```
    --role-name $role_name \
    --role-only \
    --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEFSCSIDriverPolicy
 \
    --approve
TRUST_POLICY=$(aws iam get-role --role-name $role_name --query
 'Role.AssumeRolePolicyDocument' | \
    sed -e 's/efs-csi-controller-sa/efs-csi-*/' -e 's/StringEquals/StringLike/')
aws iam update-assume-role-policy --role-name $role_name --policy-document
 "$TRUST_POLICY"
```

- Zusätzliche Informationen – Weitere Informationen zum Add-On finden Sie unter Amazon EFS-CSI-Treiber.

Mountpoint für Amazon-S3-CSI-Treiber

- Name (Name – `aws-mountpoint-s3-csi-driver`
- Beschreibung – Ein Kubernetes-Container-Speicherschnittstellen-Plugin (CSI), das Amazon-S3-Speicher für Ihren Cluster bereitstellt.
- Erforderliche IAM-Berechtigungen: Dieses Add-On nutzt die Funktion IAM-Rollen für Servicekonten von Amazon EKS. Für die erstellte IAM-Rolle ist eine Richtlinie erforderlich, die Zugriff auf S3 gewährt. Folgen Sie bei der Erstellung der Richtlinie den Empfehlungen für Mountpoint-IAM-Berechtigungen. Sie können auch die AWS verwaltete Richtlinie verwenden AmazonS3FullAccess, aber diese verwaltete Richtlinie gewährt mehr Berechtigungen, als erforderlich sindMountpoint.

  Sie können eine IAM-Rolle erstellen und ihr Ihre Richtlinie mit dem folgenden Befehl anfügen. *Ersetzen Sie my-cluster durch den Namen Ihres Clusters, den Regionscode durch den richtigen AWS-Region Code, amazoneks_S3_CSI_ durch den Namen Ihrer Rolle und DriverRoleamazonEKS_S3_CSI_ _ARN durch den Rollen-ARN. DriverRole* Diese Befehle erfordern, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Anweisungen zur Verwendung Erstellen einer IAM-Rolle der IAM-Konsole oder finden Sie unter. AWS CLI

  ```
  CLUSTER_NAME=my-cluster
  REGION=region-code
  ROLE_NAME=AmazonEKS_S3_CSI_DriverRole
  POLICY_ARN=AmazonEKS_S3_CSI_DriverRole_ARN
  eksctl create iamserviceaccount \
      --name s3-csi-driver-sa \
      --namespace kube-system \
  ```

```
    --cluster $CLUSTER_NAME \
    --attach-policy-arn $POLICY_ARN \
    --approve \
    --role-name $ROLE_NAME \
    --region $REGION \
    --role-only
```

- Zusätzliche Informationen – Weitere Informationen zum Add-On finden Sie unter Mountpoint für Amazon S3-CSI-Treiber.

CSI-Snapshot-Controller

- Name (Name – `snapshot-controller`
- Beschreibung – Der Container Storage Interface (CSI)-Snapshot-Controller ermöglicht die Verwendung der Snapshot-Funktionalität in kompatiblen CSI-Treibern, wie dem Amazon-EBS-CSI-Treiber.
- Erforderliche IAM-Berechtigungen – Für dieses Add-On sind keine Berechtigungen erforderlich.
- Zusätzliche Informationen – Weitere Informationen zum Add-On finden Sie unter CSI-Snapshot-Controller.

AWS Distribution für OpenTelemetry

- Name (Name – `adot`
- Beschreibung — Die AWS Distribution für OpenTelemetry (ADOT) ist eine sichere, produktionsbereite und AWS unterstützte Distribution des Projekts. OpenTelemetry
- Erforderliche IAM-Berechtigungen – Für dieses Add-on sind nur IAM-Berechtigungen erforderlich, wenn Sie eine der vorkonfigurierten benutzerdefinierten Ressourcen verwenden, für die Sie sich über die erweiterte Konfiguration anmelden können.
- Zusätzliche Informationen — Weitere Informationen finden Sie in der Dokumentation unter Erste Schritte mit AWS Distro zur OpenTelemetry Verwendung von EKS-Add-Ons in der Distribution. AWS OpenTelemetry

  Als Voraussetzung für ADOT muss `cert-manager` im Cluster bereitgestellt werden. Andernfalls funktioniert dieses Add-on nicht, wenn es direkt über die Eigenschaft `cluster_addons` von Amazon EKS Terraform bereitgestellt wird. Weitere Anforderungen finden Sie in der Dokumentation unter Anforderungen für die ersten Schritte mit AWS Distro für die OpenTelemetry Verwendung von EKS-Add-Ons in der AWS Distro. OpenTelemetry

Amazon GuardDuty Agent

- Name (Name – `aws-guardduty-agent`
- Beschreibung — Amazon GuardDuty ist ein Sicherheitsüberwachungsservice, der grundlegende Datenquellen wie AWS CloudTrail Verwaltungsereignisse und Amazon VPC-Flow-Logs analysiert und verarbeitet. Amazon verarbeitet GuardDuty auch Funktionen wie Kubernetes Audit-Logs und Laufzeitüberwachung.
- Erforderliche IAM-Berechtigungen – Für dieses Add-On sind keine Berechtigungen erforderlich.
- Zusätzliche Informationen — Weitere Informationen finden Sie unter Laufzeitüberwachung für Amazon EKS-Cluster in Amazon GuardDuty.
  - Um potenzielle Sicherheitsbedrohungen in Ihren Amazon EKS-Clustern zu erkennen, aktivieren Sie Amazon GuardDuty Runtime Monitoring und stellen Sie den GuardDuty Security Agent auf Ihren Amazon EKS-Clustern bereit.

Amazon CloudWatch Observability-Agent

- Name (Name – `amazon-cloudwatch-observability`
- Beschreibung Amazon CloudWatch Agent ist der von AWS bereitgestellte Überwachungs- und Beobachtungsservice. Dieses Add-on installiert den CloudWatch Agenten und aktiviert sowohl CloudWatch Application Signals als auch CloudWatch Container Insights mit verbesserter Observability für Amazon EKS.
- Erforderliche IAM-Berechtigungen: Dieses Add-On nutzt die Funktion IAM-Rollen für Servicekonten von Amazon EKS. Die Berechtigungen in den CloudWatchAgentServerRichtlinien AWSXrayWriteOnlyAccessund in den Richtlinien AWS verwalteten Richtlinien sind erforderlich. Sie können eine IAM-Rolle erstellen, ihr die verwalteten Richtlinien anhängen und das vom Add-On verwendete Kubernetes-Servicekonto mit dem folgenden Befehl kommentieren. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und *`AmazonEKS_Observability_role`* durch den Namen Ihrer Rolle. Dieser Befehl erfordert, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool verwenden müssen, um die Rolle zu erstellen, ihr die Richtlinie zuzuordnen und das Kubernetes-Servicekonto mit Anmerkungen zu versehen, siehe Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.

```
eksctl create iamserviceaccount \
    --name cloudwatch-agent \
    --namespace amazon-cloudwatch \
    --cluster my-cluster \
```

```
    --role-name AmazonEKS_Observability_Role \
    --role-only \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
    --approve
```

- Zusätzliche Informationen — Weitere Informationen finden Sie unter Installieren des CloudWatch Agenten.

Amazon EKS Pod Identity Agent

- Name (Name – `eks-pod-identity-agent`
- Beschreibung — Amazon EKS Pod Identity bietet die Möglichkeit, Anmeldeinformationen für Ihre Anwendungen zu verwalten, ähnlich wie Amazon EC2 Instance-Profile Anmeldeinformationen für EC2-Instances bereitstellen.
- Erforderliche IAM-Berechtigungen – Für dieses Add-on sind Berechtigungen von der Amazon-EKS-Knoten-IAM-Rolle erforderlich.
- Update-Informationen – Sie können jeweils nur eine Nebenversion aktualisieren. Wenn Ihre aktuelle Version beispielsweise `1.28.`*x*`-eksbuild.`*y* ist und Sie auf `1.30.`*x*`-eksbuild.`*y* aktualisieren möchten, müssen Sie zuerst ihre aktuelle Version auf `1.29.`*x*`-eksbuild.`*y* und dann auf `1.30.`*x*`-eksbuild.`*y* aktualisieren. Weitere Informationen zum Aktualisieren des Add-ons finden Sie unter Aktualisieren des Amazon-EKS-Add-ons.

Zusätzliche Amazon-EKS-Add-Ons von unabhängigen Softwareanbietern

Zusätzlich zur vorherigen Liste der Amazon-EKS-Add-Ons können Sie auch eine große Auswahl an Amazon-EKS-Add-Ons für Betriebssoftware von unabhängigen Softwareanbietern hinzufügen. Wählen Sie ein Add-on aus, um mehr darüber und seine Installationsanforderungen zu erfahren.

Suchen, beschaffen und implementieren Sie Add-Ons von AWS Marketplace für Amazon EKS (YouTube).

Accuknox

- Publisher – Accuknox
- Name (Name – `accuknox_kubearmor`
- Namespace – `kubearmor`

- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Bei diesem Add-on wird keine verwaltete Richtlinie verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie KubeArmor in der KubeArmor Dokumentation unter Erste Schritte mit.

Akuity

- Publisher – Akuity
- Name (Name – `akuity_agent`
- Namespace – `akuity`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie unter Installation des Akuity Agents auf Amazon EKS mit dem Akuity EKS-Add-on in der Akuity-Plattform-Dokumentation.

Calyptia

- Publisher – Calyptia
- Name (Name – `calyptia_fluent-bit`
- Namespace – `calytia-fluentbit`
- Servicekonto-Name – `clyptia-fluentbit`
- AWS verwaltete IAM-Richtlinie —. AWSMarketplaceMeteringRegisterUsage
- Befehl zum Erstellen der erforderlichen IAM-Rolle – Der folgende Befehl erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen verfügen oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und *`my-calyptia-role`* durch den Namen Ihrer Rolle. Dieser Befehl erfordert, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool verwenden müssen, um die Rolle

zu erstellen und das Kubernetes-Servicekonto zu annotieren, lesen Sie Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.

```
eksctl create iamserviceaccount --name service-account-name  --namespace calyptia-
fluentbit --cluster my-cluster --role-name my-calyptia-role \
    --role-only --attach-policy-arn arn:aws:iam::aws:policy/
AWSMarketplaceMeteringRegisterUsage --approve
```

- Anweisungen zur Einrichtung und Verwendung – Lesen Sie Calyptia for Fluent Bit in der Calyptia-Dokumentation.

Cisco Observability Collector

- Publisher – Cisco
- Name (Name – `cisco_cisco-cloud-observability-collectors`
- Namespace – `appdynamics`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Bei diesem Add-on wird keine verwaltete Richtlinie verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie unter Verwenden der Cisco Cloud Observability AWS Marketplace Marketplace-Add-Ons in der AppDynamics Cisco-Dokumentation.

Cisco Observability Operator

- Publisher – Cisco
- Name (Name – `cisco_cisco-cloud-observability-operators`
- Namespace – `appdynamics`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Bei diesem Add-on wird keine verwaltete Richtlinie verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.

- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie unter Verwenden der Cisco Cloud Observability AWS Marketplace Marketplace-Add-Ons in der AppDynamics Cisco-Dokumentation.

CLOUDSOFT

- Publisher – CLOUDSOFT
- Name (Name – `cloudsoft_cloudsoft-amp`
- Namespace – `cloudsoft-amp`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Bei diesem Add-on wird keine verwaltete Richtlinie verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — siehe Amazon EKS ADDON in der CLOUDSOFT-Dokumentation.

Cribl

- Publisher – Cribl
- Name (Name – `cribl_cribledge`
- Namespace – `cribledge`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Lesen Sie Installing the Cribl Amazon EKS Add-on for Edge in der Cribl-Dokumentation.

Dynatrace

- Publisher – Dynatrace
- Name (Name – `dynatrace_dynatrace-operator`
- Namespace – `dynatrace`

- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Lesen Sie die Kubernetes-Überwachung in der dynatrace-Dokumentation.

Datree

- Publisher – Datree
- Name (Name – `datree_engine-pro`
- Namespace – `datree`
- Servicekonto-Name – datree-webhook-server-awsmp
- AWS verwaltete IAM-Richtlinie —. AWSLicenseManagerConsumptionPolicy
- Befehl zum Erstellen der erforderlichen IAM-Rolle – Der folgende Befehl erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen verfügen oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und *`my-datree-role`* durch den Namen Ihrer Rolle. Dieser Befehl erfordert, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool verwenden müssen, um die Rolle zu erstellen und das Kubernetes-Servicekonto zu annotieren, lesen Sie Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.

```
eksctl create iamserviceaccount --name datree-webhook-server-awsmp --namespace datree
 --cluster my-cluster --role-name my-datree-role \
    --role-only --attach-policy-arn arn:aws:iam::aws:policy/service-role/
AWSLicenseManagerConsumptionPolicy --approve
```

- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Weitere Informationen finden Sie unter Amazon-EKS-Integration in der Datree-Dokumentation.

Datadog

- Publisher – Datadog
- Name (Name – `datadog_operator`
- Namespace – `datadog-agent`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Bei diesem Add-on wird keine verwaltete Richtlinie verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Lesen Sie Installing the Datadog Agent on Amazon EKS with the Datadog Operator Add-on in der Datadog-Dokumentation.

Groundcover

- Publisher – groundcover
- Name (Name – `groundcover_agent`
- Namespace – `groundcover`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Lesen Sie Installing the groundcover Amazon EKS Add-on in der groundcover-Dokumentation.

Grafana Labs

- Publisher – Grafana Labs
- Name (Name – `grafana-labs_kubernetes-monitoring`
- Namespace – `monitoring`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.

- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung: Informationen zum Konfigurieren von Kubernetes Monitoring als Add-On mit Amazon EKS finden Sie hier in der Dokumentation zu Grafana Labs.

HA Proxy

- Publisher – HA Proxy
- Name (Name – `haproxy-technologies_kubernetes-ingress-ee`
- Namespace – `haproxy-controller`
- Servicekonto-Name – `customer defined`
- AWS verwaltete IAM-Richtlinie —. AWSLicenseManagerConsumptionPolicy
- Befehl zum Erstellen der erforderlichen IAM-Rolle – Der folgende Befehl erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen verfügen oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und *`my-haproxy-role`* durch den Namen Ihrer Rolle. Dieser Befehl erfordert, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool verwenden müssen, um die Rolle zu erstellen und das Kubernetes-Servicekonto zu annotieren, lesen Sie Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.

```
eksctl create iamserviceaccount --name service-account-name  --namespace haproxy-
controller --cluster my-cluster --role-name my-haproxy-role \
    --role-only --attach-policy-arn arn:aws:iam::aws:policy/service-role/
AWSLicenseManagerConsumptionPolicy --approve
```

- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Siehe Installieren Sie HAProxy- Enterprise-Kubernetes-Ingress-Controller auf Amazon EKS von AWSin der HAProxy-Dokumentation.

Kpow

- Publisher – Factorhouse
- Name (Name – `factorhouse_kpow`
- Namespace – `factorhouse`

- Servicekonto-Name – `kpow`
- AWS verwaltete IAM-Richtlinie — AWSLicenseManagerConsumptionPolicy
- Befehl zum Erstellen der erforderlichen IAM-Rolle – Der folgende Befehl erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen verfügen oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und *`my-kpow-role`* durch den Namen Ihrer Rolle. Dieser Befehl erfordert, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool verwenden müssen, um die Rolle zu erstellen und das Kubernetes-Servicekonto zu annotieren, lesen Sie Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.

  ```
  eksctl create iamserviceaccount --name kpow --namespace factorhouse --cluster my-cluster --role-name my-kpow-role \
      --role-only --attach-policy-arn arn:aws:iam::aws:policy/service-role/AWSLicenseManagerConsumptionPolicy --approve
  ```

- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Lesen Sie AWS Marketplace LM in der Kpow-Dokumentation.

Kubecost

- Publisher – Kubecost
- Name (Name – `kubecost_kubecost`
- Namespace – `kubecost`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Nutzung — Weitere Informationen finden Sie in der Kubecost Dokumentation zur Integration von AWS Cloud Billing.
- Wenn Ihr Cluster Version `1.23` oder höher aufweist, muss the section called “Amazon-EBS-CSI-Treiber” auf Ihrem Cluster installiert sein. Andernfalls erhalten Sie eine Fehlermeldung.

Kasten

- Publisher – Kasten by Veeam
- Name (Name – `kasten_k10`
- Namespace – `kasten-io`
- Servicekonto-Name – `k10-k10`
- AWS verwaltete IAM-Richtlinie — AWSLicenseManagerConsumptionPolicy.
- Befehl zum Erstellen der erforderlichen IAM-Rolle – Der folgende Befehl erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen verfügen oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und *`my-kasten-role`* durch den Namen Ihrer Rolle. Dieser Befehl erfordert, dass Sie `eksctl` auf Ihrem Gerät installiert haben. Wenn Sie ein anderes Tool verwenden müssen, um die Rolle zu erstellen und das Kubernetes-Servicekonto zu annotieren, lesen Sie Konfigurieren Sie ein Kubernetes Dienstkonto für die Übernahme einer IAM-Rolle.

  ```
  eksctl create iamserviceaccount --name k10-k10 --namespace kasten-io --cluster my-cluster --role-name my-kasten-role \
      --role-only --attach-policy-arn arn:aws:iam::aws:policy/service-role/AWSLicenseManagerConsumptionPolicy --approve
  ```

- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie unter Installation von K10 unter AWS Verwendung des Amazon EKS Add-ons in der Kasten-Dokumentation.
- Zusätzliche Informationen – Wenn es sich bei Ihrem Amazon-EKS-Cluster um die Version Kubernetes `1.23` oder höher handelt, muss der CSI-Treiber von Amazon EBS im Cluster mit einer standardmäßigen `StorageClass` installiert sein.

Kong

- Publisher – Kong
- Name (Name – `kong_konnect-ri`
- Namespace – `kong`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.

- AWS verwaltete IAM-Richtlinie — Bei diesem Add-on wird keine verwaltete Richtlinie verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Lesen Sie Installing the Kong Gateway EKS Add-on in der Kong-Dokumentation.

LeakSignal

- Publisher – LeakSignal
- Name (Name – `leaksignal_leakagent`
- Namespace – `leakagent`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie in der LeakSignal Dokumentation unter Installieren des LeakAgent Add-ons.

NetApp

- Publisher – NetApp
- Name (Name – `netapp_trident-operator`
- Namespace – `trident`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie in der Dokumentation unter Konfiguration des Astra Trident EKS-Add-ons. NetApp

New Relic

- Publisher – New Relic
- Name (Name – `new-relic_kubernetes-operator`
- Namespace – `newrelic`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Lesen Sie Installing the New Relic Add-on for EKS in der Dokumentation von New Relic.

Rafay

- Publisher – Rafay
- Name (Name – `rafay-systems_rafay-operator`
- Namespace – `rafay-system`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Lesen Sie Installing the Rafay Amazon EKS Add-on in der Rafay-Dokumentation.

Solo.io

- Publisher – Solo.io
- Name (Name – `solo-io_istio-distro`
- Namespace – `istio-system`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.

- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie unter Installation von Istio in der Solo.io-Dokumentation.

Stormforge

- Publisher – Stormforge
- Name (Name – `stormforge_optimize-Live`
- Namespace – `stormforge-system`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Weitere Informationen finden Sie in der StormForge Dokumentation unter Installation des StormForge Agenten.

Splunk

- Publisher – Splunk
- Name (Name – `splunk_splunk-otel-collector-chart`
- Namespace – `splunk-monitoring`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Bei diesem Add-on wird keine verwaltete Richtlinie verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung: Informationen zum Installieren des Splunk-Add-Ons für Amazon EKS finden Sie hier in der Dokumentation zu Splunk.

Teleport

- Publisher – Teleport

- Name (Name – `teleport_teleport`
- Namespace – `teleport`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Informationen zur Funktionsweise von Teleport finden Sie in der Teleport-Dokumentation.

Tetrate

- Publisher – Tetrate Io
- Name (Name – `tetrate-io_istio-distro`
- Namespace – `istio-system`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung – Weitere Informationen finden Sie auf der Website von Tetrate Istio Distro.

Upbound Universal Crossplane

- Publisher – Upbound
- Name (Name – `upbound_universal-crossplane`
- Namespace – `upbound-system`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.

- Anweisungen zur Einrichtung und Verwendung – Lesen Sie Upbound Universal Crossplane (UXP) in der Upbound-Dokumentation.

Upwind

- Publisher – Upwind
- Name (Name – `upwind`
- Namespace – `upwind`
- Servicekontoname – Ein Servicekonto wird mit diesem Add-On nicht verwendet.
- AWS verwaltete IAM-Richtlinie — Eine verwaltete Richtlinie wird mit diesem Add-on nicht verwendet.
- Benutzerdefinierte IAM-Berechtigungen – Benutzerdefinierte Berechtigungen werden mit diesem Add-On nicht verwendet.
- Anweisungen zur Einrichtung und Verwendung — Die Installationsschritte finden Sie in der Upwind-Dokumentation.

Verwalten von Amazon EKS-Add-Ons

Amazon EKS-Add-Ons sind ein kuratierter Satz von Add-On-Software für Amazon EKS-Cluster. Alle Amazon-EKS-Add-Ons:

- enthalten die neuesten Sicherheitspatches und Fehlerbehebungen.
- wurden von AWS für die Zusammenarbeit mit Amazon EKS validiert.
- reduzieren den Arbeitsaufwand für die Verwaltung der Add-On-Software.

Der AWS Management Console benachrichtigt Sie, wenn eine neue Version für ein Amazon EKS-Add-on verfügbar ist. Sie können das Update einfach initiieren, und Amazon EKS aktualisiert die Add-On-Software für Sie.

Eine Liste der verfügbaren Add-Ons finden Sie unter Verfügbare Amazon-EKS-Add-Ons von Amazon EKS. Weitere Informationen zur Kubernetes-Feldverwaltung finden Sie unter Kubernetes-Feldverwaltung

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS.

Erstellen eines Add-Ons

Sie können ein Amazon EKS-Add-on mit `eksctl` AWS Management Console, dem oder dem erstellen AWS CLI. Wenn das Add-On eine IAM-Rolle erfordert, lesen Sie die Details für das spezifische Add-On in Verfügbare Amazon-EKS-Add-Ons von Amazon EKS für Details zum Erstellen der Rolle.

eksctl

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

So erstellen Sie ein Amazon-EKS-Add-On mit **`eksctl`**

1. Zeigen Sie die Namen der Add-Ons an, die für eine Cluster-Version verfügbar sind. Ersetzen Sie *`1.30`* durch die Version Ihres Clusters.

```
eksctl utils describe-addon-versions --kubernetes-version 1.30 | grep AddonName
```

Eine Beispielausgabe sieht wie folgt aus.

```
"AddonName": "aws-ebs-csi-driver",
                        "AddonName": "coredns",
                        "AddonName": "kube-proxy",
                        "AddonName": "vpc-cni",
                        "AddonName": "adot",
                        "AddonName": "dynatrace_dynatrace-operator",
                        "AddonName": "upbound_universal-crossplane",
                        "AddonName": "teleport_teleport",
                        "AddonName": "factorhouse_kpow",
                        [...]
```

2. Zeigen Sie die Versionen des Add-Ons an, die für das Add-On verfügbar sind, das Sie erstellen möchten. Ersetzen Sie *1.30* durch die Version Ihres Clusters. Ersetzen Sie *name-of-addon* mit dem Namen des Add-Ons, für das Sie die Versionen anzeigen möchten. Der Name muss einer der Namen sein, die in den vorherigen Schritten zurückgegeben wurden.

```
eksctl utils describe-addon-versions --kubernetes-version 1.30 --name name-of-addon | grep AddonVersion
```

Die folgende Ausgabe ist ein Beispiel dafür, was für das Add-On mit dem Namen `vpc-cni` zurückgegeben wird. Sie können sehen, dass das Add-On in mehreren Versionen verfügbar ist.

```
"AddonVersions": [
    "AddonVersion": "v1.12.0-eksbuild.1",
    "AddonVersion": "v1.11.4-eksbuild.1",
    "AddonVersion": "v1.10.4-eksbuild.1",
    "AddonVersion": "v1.9.3-eksbuild.1",
```

3. Bestimmen Sie, ob es sich bei dem Add-On, das Sie erstellen möchten, um ein Amazon-EKS- oder AWS Marketplace -Add-On handelt. Das AWS Marketplace hat Add-Ons von Drittanbietern, für die Sie zusätzliche Schritte ausführen müssen, um das Add-on zu erstellen.

```
eksctl utils describe-addon-versions --kubernetes-version 1.30 --name name-of-addon | grep ProductUrl
```

Wenn keine Ausgabe zurückgegeben wird, handelt es sich bei dem Add-On um ein Amazon EKS. Wenn eine Ausgabe zurückgegeben wird, handelt es sich bei dem Add-On um ein AWS Marketplace Add-On. Die folgende Ausgabe bezieht sich auf ein Add-On mit dem Namen `teleport_teleport`.

```
"ProductUrl": "https://aws.amazon.com/marketplace/pp?
sku=3bda70bb-566f-4976-806c-f96faef18b26"
```

Sie können mehr über das Add-on in der AWS Marketplace mit der zurückgegebenen URL erfahren. Wenn für das Add-On ein Abonnement erforderlich ist, können Sie das Add-On über das AWS Marketplace abonnieren. Wenn Sie aus dem ein Add-On erstellen möchten AWS Marketplace, muss der IAM-Principal, den Sie zum Erstellen des Add-ons verwenden, über die Berechtigung verfügen, die

AWSServiceRoleForAWSLicenseManagerRoleserviceverknüpfte Rolle zu erstellen. Weitere Informationen zum Zuweisen von Berechtigungen zu einer IAM-Entität finden Sie unter Hinzufügen und Entfernen von IAM-Identitäts-Berechtigungen im IAM-Benutzerhandbuch.

4. Erstellen Sie ein Amazon-EKS-Add-On. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie nach Bedarf die folgenden Änderungen am Befehl vor und führen Sie anschließend den geänderten Befehl aus:

   - Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.
   - Ersetzen Sie *name-of-addon* durch den Namen des Add-Ons, das Sie erstellen möchten.
   - Wenn Sie eine Version des Add-Ons wünschen, die älter als die neueste Version ist, ersetzen Sie *latest* durch die Versionsnummer, die in der Ausgabe eines vorherigen Schritts zurückgegeben wurde, den Sie verwenden möchten.
   - Wenn das Add-On eine Servicekonto-Rolle verwendet, ersetzen Sie *111122223333* durch Ihre Konto-ID und *role-name* durch den Namen der Rolle. Anweisungen zum Erstellen einer Rolle für Ihr Servicekonto finden Sie in der Dokumentation für das Add-On, das Sie erstellen. Die Angabe einer Servicekonto-Rolle erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen für Ihren Cluster verfügen, oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.

     Wenn das Add-On keine Servicekonto-Rolle verwendet, löschen Sie ***--service-account-role-arn*** **arn:aws:iam::*111122223333*:role/*role-name***.
   - Dieser Beispielbefehl überschreibt die Konfiguration aller vorhandenen selbstverwalteten Versionen des Add-Ons, falls es eine gibt. Wenn Sie nicht möchten, dass die Konfiguration eines vorhandenen selbstverwalteten Add-Ons überschreibt, entfernen Sie die Option *--force*. Wenn Sie die Option entfernen und das Amazon-EKS-Add-On die Konfiguration eines vorhandenen selbstverwalteten Add-Ons benötigt, scheitert die Erstellung des Amazon-EKS-Add-Ons mit einer Fehlermeldung, die Sie bei der Behebung des Konflikts unterstützt. Stellen Sie vor dem Angeben dieser Option sicher, dass das Amazon-EKS-Add-on keine Einstellungen verwaltet, die Sie verwalten müssen, da diese Einstellungen mit dieser Option überschrieben werden.

     ```
     eksctl create addon --cluster my-cluster --name name-of-addon --version latest
      \
         --service-account-role-arn arn:aws:iam::111122223333:role/role-name --
     force
     ```

Sie finden eine Liste mit allen verfügbaren Optionen für den Befehl.

```
eksctl create addon --help
```

Weitere Informationen zu verfügbaren Optionen finden Sie unter Add-Ons in der `eksctl`-Dokumentation.

AWS Management Console

So erstellen Sie ein Amazon EKS-Add-on mit dem AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich Clusters (Cluster) und wählen Sie dann den Namen des Clusters aus, für den Sie das Add-On erstellen möchten.
3. Wählen Sie die Registerkarte Add-ons.
4. Wählen Sie Weitere Add-Ons erhalten.
5. Wählen Sie die Add-Ons aus, die Sie Ihrem Cluster hinzufügen möchten. Sie können beliebig viele Amazon-EKS-Add-ons und AWS Marketplace -Add-ons hinzufügen.

   Für AWS MarketplaceAdd-Ons muss der IAM-Principal, den Sie zur Erstellung des Add-ons verwenden, über Berechtigungen zum Lesen von Berechtigungen für das Add-on aus dem verfügen. AWS LicenseManager AWS LicenseManager erfordert eine AWSServiceRoleForAWSLicenseManagerRoleserviceverknüpfte Rolle (SLR), die es AWS Ressourcen ermöglicht, Lizenzen in Ihrem Namen zu verwalten. Das SLR ist eine einmalige Anforderung pro Konto, und Sie müssen keine separaten SLRs für jedes Add-On oder jeden Cluster erstellen. Weitere Informationen zum Zuweisen von Berechtigungen zu einem IAM-Prinzipal finden Sie unter Hinzufügen und Entfernen von IAM-Identitäts-Berechtigungen im IAM-Benutzerhandbuch.

   Wenn die AWS Marketplace -Add-ons, die Sie installieren möchten, nicht aufgeführt sind, können Sie nach verfügbaren Add-ons suchen, indem Sie Text in das Suchfeld eingeben. In den Filtering options (Filteroptionen) können Sie auch nach category (Kategorie), vendor (Anbieter) oder pricing model (Preismodell) filtern und dann die Add-ons aus den Suchergebnissen auswählen. Nachdem Sie die Add-Ons ausgewählt haben, die Sie installieren möchten, wählen Sie Next (Weiter).
6. Gehen Sie auf der Seite Configure selected add-ons settings (Ausgewählte Add-On-Einstellungen konfigurieren) wie folgt vor:

- Wählen Sie View subscription options (Abonnementoptionen anzeigen), um das Formular Subscription options (Abonnementoptionen) zu öffnen. Lesen Sie die Abschnitte Pricing details (Preisinformationen) und Legal (Rechtliche Hinweise) und klicken Sie dann auf Subscribe (Abonnieren), um fortzufahren.
- Wählen Sie für Version die Version aus, die Sie installieren möchten. Wir empfehlen die als latest (neueste) Version markierte Version, es sei denn, das einzelne Add-On, das Sie erstellen, empfiehlt eine andere Version. Um festzustellen, ob ein Add-On über eine empfohlene Version verfügt, lesen Sie die Dokumentation für das Add-On, das Sie erstellen.
- Wenn für alle von Ihnen ausgewählten Add-Ons Requires subscription (Abonnement erforderlich) ist unter Status, wählen Sie Next (Weiter) aus. Sie können diese Add-Ons konfigurieren, wenn Sie sie nach der Erstellung Ihres Clusters abonniert haben. Für die Add-ons, die kein Requires subscription (Abonnement erforderlich) unter Status haben:
  - Übernehmen Sie für Select IAM role (IAM-Rolle auswählen) die Standardoption, es sei denn, das Add-On erfordert IAM-Berechtigungen. Wenn für das Add-on AWS Berechtigungen erforderlich sind, können Sie die IAM-Rolle des Knotens (nicht festgelegt) oder eine vorhandene Rolle verwenden, die Sie für die Verwendung mit dem Add-on erstellt haben. Wenn Sie keine Rolle auswählen können, verfügen Sie über keine vorhandene Rolle. Unabhängig davon, für welche Option Sie sich entscheiden, lesen Sie die Dokumentation für das Add-on, das Sie erstellen, um eine IAM-Richtlinie zu erstellen und sie einer Rolle anzufügen. Die Auswahl einer IAM-Rolle erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen für Ihren Cluster verfügen, oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
  - Wählen Sie Optional configuration settings (Optionale Konfigurationseinstellungen) aus.
    - Wenn für das Add-On eine Konfiguration erforderlich ist, geben Sie diese in das Feld Configuration values (Konfigurationswerte) ein. Um festzustellen, ob für das Add-On Konfigurationsinformationen erforderlich sind, lesen Sie die Dokumentation für das Add-On, das Sie erstellen.
    - Wählen Sie eine der verfügbaren Optionen für die Conflict resolution method (Konfliktlösungsmethode) aus.
  - Wählen Sie Weiter aus.

7. Wählen Sie auf der Seite Überprüfen und hinzufügen die Option Erstellen aus. Nachdem die Installation der Add-Ons abgeschlossen ist, werden Ihre installierten Add-Ons angezeigt.

8. Wenn für eines der von Ihnen installierten Add-Ons ein Abonnement erforderlich ist, gehen Sie wie folgt vor:

    1. Wählen Sie die Schaltfläche Subscribe (Abonnieren) in der unteren rechten Ecke für das Add-On. Sie werden auf die Seite für das Add-On in der AWS Marketplace weitergeleitet. Lesen Sie die Informationen zum Add-On, z. B. die Product Overview (Produktübersicht) und die Pricing Information (Preisinformationen).
    2. Wählen Sie oben rechts auf der Add-on-Seite die Schaltfläche Continue to Subscribe (Mit dem Abonnement fortfahren) aus.
    3. Lesen Sie sich die (Allgemeinen Geschäftsbedingungen) durch. Wenn Sie damit einverstanden sind, wählen Sie Accept Terms (Bedingungen akzeptieren) aus. Die Bearbeitung des Abonnements kann mehrere Minuten dauern. Während das Abonnement bearbeitet wird, ist die Schaltfläche Return to Amazon EKS Console (Zurück zur Amazon-EKS-Konsole) ausgegraut.
    4. Sobald die Bearbeitung des Abonnements abgeschlossen ist, ist die Schaltfläche Return to Amazon EKS Console (Zurück zur Amazon-EKS-Konsole) nicht mehr ausgegraut. Wählen Sie die Schaltfläche aus, um zur Registerkarte Add-ons (Add-Ons) der Amazon-EKS-Konsole für Ihren Cluster zurückzukehren.
    5. Wählen Sie für das Add-On, das Sie abonniert haben, Remove and reinstall (Entfernen und erneut installieren) und dann Reinstall add-on (Add-On erneut installieren) aus. Die Installation des Add-Ons kann einige Minuten dauern. Wenn die Installation abgeschlossen ist, können Sie das Add-On konfigurieren.

AWS CLI

Voraussetzung

Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder AWS CloudShell. Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

So erstellen Sie ein Amazon EKS-Add-on mit dem AWS CLI

1. Bestimmen Sie, welche Add-Ons verfügbar sind. Sie können alle verfügbaren Add-Ons, ihren Typ und ihren Publisher anzeigen. Sie können auch die URL für Add-ons anzeigen, die über das AWS Marketplace verfügbar sind. Ersetzen Sie *1.30* durch die Version Ihres Clusters.

```
aws eks describe-addon-versions --kubernetes-version 1.30 \
    --query 'addons[].{MarketplaceProductUrl: marketplaceInformation.productUrl,
 Name: addonName, Owner: owner Publisher: publisher, Type: type}' --output table
```

Eine Beispielausgabe sieht wie folgt aus.

```
-------------------------------------------------------------------------------------------
|
 DescribeAddonVersions
       |
+-----------------------------------------------------------------
+------------------------------+------------------+--------------
+---------------------+
|                     MarketplaceProductUrl                      |
 Name              |      Owner       |  Publisher   |        Type         |
+-----------------------------------------------------------------
+------------------------------+------------------+--------------
+---------------------+
|  None                                                          |  aws-ebs-csi-
driver           |  aws             |  eks         |  storage            |
|  None                                                          |  coredns
                |  aws             |  eks         |  networking         |
|  None                                                          |  kube-proxy
                |  aws             |  eks         |  networking         |
|  None                                                          |  vpc-cni
                |  aws             |  eks         |  networking         |
|  None                                                          |  adot
                |  aws             |  eks         |  observability      |
|  https://aws.amazon.com/marketplace/pp/prodview-brb73nceicv7u |
 dynatrace_dynatrace-operator |  aws-marketplace |  dynatrace   |  monitoring
      |
|  https://aws.amazon.com/marketplace/pp/prodview-uhc2iwi5xysoc |
 upbound_universal-crossplane |  aws-marketplace |  upbound     |  infra-
management   |
```

```
|  https://aws.amazon.com/marketplace/pp/prodview-hd2ydsrgqy4li |
 teleport_teleport            |  aws-marketplace |  teleport    |  policy-
management  |
|  https://aws.amazon.com/marketplace/pp/prodview-vgghgqdsplhvc |
 factorhouse_kpow             |  aws-marketplace |  factorhouse |  monitoring
      |
|  [...]                                                       |  [...]
                |  [...]           |  [...]       |  [...]              |
+-------------------------------------------------------------
+------------------------------+------------------+--------------
+--------------------+
```

Ihre Ausgabe ist möglicherweise anders. In dieser Beispielausgabe sind drei verschiedene Add-Ons vom Typ `networking` und fünf Add-Ons mit einem Publisher vom Typ `eks` verfügbar. Für die Add-Ons mit `aws-marketplace` in der `Owner`-Spalte ist möglicherweise ein Abonnement erforderlich, bevor Sie sie installieren können. Sie können die URL besuchen, um mehr über das Add-On zu erfahren und es zu abonnieren.

2. Sie können anzeigen, welche Versionen für jedes Add-On verfügbar sind. Ersetzen Sie *`1.30`* durch die Version Ihres Clusters und *`vpc-cni`* durch den Namen eines Add-Ons, das im vorherigen Schritt zurückgegeben wurde.

```
aws eks describe-addon-versions --kubernetes-version 1.30 --addon-name vpc-cni \
    --query 'addons[].addonVersions[].{Version: addonVersion, Defaultversion:
 compatibilities[0].defaultVersion}' --output table
```

Eine Beispielausgabe sieht wie folgt aus.

```
-----------------------------------------
|          DescribeAddonVersions        |
+-----------------+---------------------+
| Defaultversion  |       Version       |
+-----------------+---------------------+
|  False          |  v1.12.0-eksbuild.1 |
|  True           |  v1.11.4-eksbuild.1 |
|  False          |  v1.10.4-eksbuild.1 |
|  False          |  v1.9.3-eksbuild.1  |
+-----------------+---------------------+
```

Die Version `True` in der `Defaultversion`-Spalte ist standardmäßig die Version, mit der das Add-On erstellt wurde.

3. (Optional) Finden Sie die Konfigurationsoptionen für das von Ihnen gewählte Add-On, indem Sie den folgenden Befehl ausführen:

```
aws eks describe-addon-configuration --addon-name vpc-cni --addon-
version v1.12.0-eksbuild.1
```


```
{
    "addonName": "vpc-cni",
    "addonVersion": "v1.12.0-eksbuild.1",
    "configurationSchema": "{\"$ref\":\"#/definitions/VpcCni\",\"$schema
\":\"http://json-schema.org/draft-06/schema#\",\"definitions\":{\"Cri\":
{\"additionalProperties\":false,\"properties\":{\"hostPath\":{\"$ref\":
\"#/definitions/HostPath\"}},\"title\":\"Cri\",\"type\":\"object\"},\"Env
\":{\"additionalProperties\":false,\"properties\":{\"ADDITIONAL_ENI_TAGS
\":{\"type\":\"string\"},\"AWS_VPC_CNI_NODE_PORT_SUPPORT\":{\"format\":
\"boolean\",\"type\":\"string\"},\"AWS_VPC_ENI_MTU\":{\"format\":\"integer
\",\"type\":\"string\"},\"AWS_VPC_K8S_CNI_CONFIGURE_RPFILTER\":{\"format
\":\"boolean\",\"type\":\"string\"},\"AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG\":
{\"format\":\"boolean\",\"type\":\"string\"},\"AWS_VPC_K8S_CNI_EXTERNALSNAT
\":{\"format\":\"boolean\",\"type\":\"string\"},\"AWS_VPC_K8S_CNI_LOGLEVEL
\":{\"type\":\"string\"},\"AWS_VPC_K8S_CNI_LOG_FILE\":{\"type
\":\"string\"},\"AWS_VPC_K8S_CNI_RANDOMIZESNAT\":{\"type\":
\"string\"},\"AWS_VPC_K8S_CNI_VETHPREFIX\":{\"type\":\"string
\"},\"AWS_VPC_K8S_PLUGIN_LOG_FILE\":{\"type\":\"string\"},
\"AWS_VPC_K8S_PLUGIN_LOG_LEVEL\":{\"type\":\"string\"},\"DISABLE_INTROSPECTION
\":{\"format\":\"boolean\",\"type\":\"string\"},\"DISABLE_METRICS\":{\"format
\":\"boolean\",\"type\":\"string\"},\"DISABLE_NETWORK_RESOURCE_PROVISIONING
\":{\"format\":\"boolean\",\"type\":\"string\"},\"ENABLE_POD_ENI\":{\"format
\":\"boolean\",\"type\":\"string\"},\"ENABLE_PREFIX_DELEGATION\":{\"format
\":\"boolean\",\"type\":\"string\"},\"WARM_ENI_TARGET\":{\"format\":\"integer
\",\"type\":\"string\"},\"WARM_PREFIX_TARGET\":{\"format\":\"integer\",
\"type\":\"string\"}},\"title\":\"Env\",\"type\":\"object\"},\"HostPath\":
{\"additionalProperties\":false,\"properties\":{\"path\":{\"type\":\"string\"}},
\"title\":\"HostPath\",\"type\":\"object\"},\"Limits\":{\"additionalProperties
\":false,\"properties\":{\"cpu\":{\"type\":\"string\"},\"memory\":{\"type
\":\"string\"}},\"title\":\"Limits\",\"type\":\"object\"},\"Resources\":
{\"additionalProperties\":false,\"properties\":{\"limits\":{\"$ref\":\"#/
definitions/Limits\"},\"requests\":{\"$ref\":\"#/definitions/Limits\"}},
\"title\":\"Resources\",\"type\":\"object\"},\"VpcCni\":{\"additionalProperties
\":false,\"properties\":{\"cri\":{\"$ref\":\"#/definitions/Cri\"},\"env\":
{\"$ref\":\"#/definitions/Env\"},\"resources\":{\"$ref\":\"#/definitions/
Resources\"}},\"title\":\"VpcCni\",\"type\":\"object\"}}}"
```

```
}
```

Die Ausgabe ist ein standardmäßiges JSON-Schema.

Hier ist ein Beispiel für gültige Konfigurationswerte im JSON-Format, die mit dem obigen Schema funktionieren.

```
{
  "resources": {
    "limits": {
      "cpu": "100m"
    }
  }
}
```

Hier ist ein Beispiel für gültige Konfigurationswerte im YAML-Format, die mit dem obigen Schema funktionieren.

```
  resources:
    limits:
      cpu: 100m
```

4. Stellen Sie fest, ob für das Add-on IAM-Berechtigungen erforderlich sind. Wenn ja, müssen Sie (1) ermitteln, ob Sie EKS Pod Identities oder IAM Roles for Service Accounts (IRSA) verwenden möchten, (2) den ARN der IAM-Rolle bestimmen, die mit dem Add-on verwendet werden soll, und (3) den Namen des Kubernetes-Dienstkontos bestimmen, das vom Add-on verwendet wird. Sie finden diese Informationen in der Dokumentation oder mithilfe der AWS API, siehe IAM-Informationen zu einem Add-on abrufen.

   - Amazon EKS schlägt vor, EKS Pod Identities zu verwenden, sofern das Add-on dies unterstützt. Dies setzt voraus, dass der Pod Identity Agent auf Ihrem Cluster installiert ist. Weitere Informationen zur Verwendung von Pod-Identitäten mit Add-Ons finden Sie unterHängen Sie mithilfe von Pod Identity eine IAM-Rolle an ein Amazon EKS-Add-on an.
   - Wenn das Add-on oder Ihr Cluster nicht für EKS Pod Identities eingerichtet ist, verwenden Sie IRSA. Vergewissern Sie sich, dass IRSA auf Ihrem Cluster eingerichtet ist.
   - Lesen Sie in der Dokumentation zu Amazon EKS Add-ons nach, ob für das Add-on IAM-Berechtigungen erforderlich sind, und geben Sie den Namen des zugehörigen Kubernetes-Servicekontos an.

5. Erstellen Sie ein Amazon-EKS-Add-On. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie nach Bedarf die folgenden Änderungen am Befehl vor und führen Sie anschließend den geänderten Befehl aus:

   - Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.
   - Ersetzen Sie *vpc-cni* durch einen Add-On-Namen, der in der Ausgabe des vorherigen Schritts zurückgegeben wurde, den Sie erstellen möchten.
   - Ersetzen Sie *version-number* durch die in der Ausgabe des vorherigen Schritts zurückgegebene Version, die Sie verwenden möchten.
   - Wenn für das Add-on keine IAM-Berechtigungen erforderlich sind, löschen Sie es. *<service-account-configuration>*
   - Wenn für das Add-on (1) IAM-Berechtigungen erforderlich sind und (2) Ihr Cluster EKS-Pod-Identitäten verwendet, *<service-account-configuration>* ersetzen Sie es durch die folgende Pod-Identitätszuordnung. Ersetzen Sie es *<service-account-name>* durch den Namen des Dienstkontos, der vom Add-on verwendet wird. Durch *<role-arn>* den ARN einer IAM-Rolle ersetzen. Die Rolle muss über die von EKS Pod Identities geforderte Vertrauensrichtlinie verfügen.
     - `--pod-identity-associations 'serviceAccount=<service-account-name>,roleArn=<role-arn>'`
   - Wenn für das Add-on (1) IAM-Berechtigungen erforderlich sind und (2) Ihr Cluster IRSA verwendet, *<service-account-configuration>* ersetzen Sie es durch die folgende IRSA-Konfiguration. *111122223333*Ersetzen Sie es durch Ihre Konto-ID und *role-name* den Namen einer vorhandenen IAM-Rolle, die Sie erstellt haben. Anweisungen zum Erstellen der Rolle finden Sie in der Dokumentation für das Add-On, das Sie erstellen. Die Angabe einer Servicekonto-Rolle erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen für Ihren Cluster verfügen, oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
     - `--service-account-role-arn arn:aws:iam::111122223333:role/role-name`
   - Diese Beispielbefehle überschreiben die Option `--configuration-values` von gegebenenfalls vorhandenen selbstverwalteten Versionen des Add-Ons. Ersetzen Sie diese durch die gewünschten Konfigurationswerte, z. B. eine Zeichenfolge oder eine Dateieingabe. Wenn Sie keine Konfigurationswerte angeben möchten, löschen Sie die `--configuration-values`-Option. Wenn Sie nicht möchten, dass die AWS CLI

Konfiguration eines vorhandenen selbstverwalteten Add-ons überschrieben wird, entfernen Sie die Option. *--resolve-conflicts OVERWRITE* Wenn Sie die Option entfernen und das Amazon-EKS-Add-On die Konfiguration eines vorhandenen selbstverwalteten Add-Ons benötigt, scheitert die Erstellung des Amazon-EKS-Add-Ons mit einer Fehlermeldung, die Sie bei der Behebung des Konflikts unterstützt. Stellen Sie vor dem Angeben dieser Option sicher, dass das Amazon-EKS-Add-on keine Einstellungen verwaltet, die Sie verwalten müssen, da diese Einstellungen mit dieser Option überschrieben werden.

```
aws eks create-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version version-number \
    <service-account-configuration> --configuration-values '{"resources":
{"limits":{"cpu":"100m"}}}' --resolve-conflicts OVERWRITE
```

```
aws eks create-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version version-number \
   <service-account-configuration> --configuration-values 'file://example.yaml'
 --resolve-conflicts OVERWRITE
```

Eine vollständige Liste der verfügbaren Optionen finden Sie unter `create-addon` in der Amazon-EKS-Befehlszeilenreferenz. Wenn für das von Ihnen erstellte Add-On `aws-marketplace` in der `Owner`-Spalte eines vorherigen Schritts aufgeführt ist, kann die Erstellung fehlschlagen und Sie erhalten eine Fehlermeldung ähnlich der folgenden.

```
{
    "addon": {
        "addonName": "addon-name",
        "clusterName": "my-cluster",
        "status": "CREATE_FAILED",
        "addonVersion": "version",
        "health": {
            "issues": [
                {
                    "code": "AddonSubscriptionNeeded",
                    "message": "You are currently not subscribed to this add-
on. To subscribe, visit the AWS Marketplace console, agree to the seller EULA,
 select the pricing type if required, then re-install the add-on"
[...]
```

Wenn Sie einen Fehler erhalten, der dem Fehler in der vorherigen Ausgabe ähnelt, besuchen Sie die URL in der Ausgabe eines vorherigen Schritts, um das Add-On zu abonnieren. Führen Sie den `create-addon`-Befehl nach dem Abonnieren erneut aus.

Aktualisieren eines Add-Ons

Amazon EKS aktualisiert das Add-On nicht automatisch, wenn neue Versionen veröffentlicht werden oder nachdem Sie Ihren Cluster auf eine neue Kubernetes-Nebenversion aktualisiert haben. Um ein Add-On für einen vorhandenen Cluster zu aktualisieren, müssen Sie das Update initiieren. Nachdem Sie das Update initiiert haben, aktualisiert Amazon EKS das Add-On für Sie. Lesen Sie vor dem Aktualisieren eines Add-Ons die aktuelle Dokumentation für das Add-On. Eine Liste der verfügbaren Add-Ons finden Sie unter Verfügbare Amazon-EKS-Add-Ons von Amazon EKS. Wenn das Add-On eine IAM-Rolle erfordert, lesen Sie die Details für das spezifische Add-On in Verfügbare Amazon-EKS-Add-Ons von Amazon EKS für Details zum Erstellen der Rolle.

Sie können ein Amazon EKS-Add-on mit `eksctl` AWS Management Console, dem oder dem aktualisieren AWS CLI.

eksctl

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

So aktualisieren Sie ein Amazon-EKS-Add-On mit **`eksctl`**

1. Bestimmen Sie die aktuellen Add-Ons und Add-On-Versionen, die auf Ihrem Cluster installiert sind. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

```
eksctl get addon --cluster my-cluster
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME        VERSION              STATUS  ISSUES  IAMROLE  UPDATE AVAILABLE
coredns     v1.8.7-eksbuild.2    ACTIVE  0
kube-proxy  v1.23.7-eksbuild.1   ACTIVE  0                v1.23.8-eksbuild.2
```

```
vpc-cni     v1.10.4-eksbuild.1   ACTIVE  0                 v1.12.0-
eksbuild.1,v1.11.4-eksbuild.1,v1.11.3-eksbuild.1,v1.11.2-eksbuild.1,v1.11.0-
eksbuild.1
```

IIhre Ausgabe sieht möglicherweise anders aus, je nachdem, welche Add-Ons und Versionen Sie auf Ihrem Cluster haben. Sie können sehen, dass in der vorherigen Beispielausgabe für zwei vorhandene Add-Ons auf dem Cluster neuere Versionen in der `UPDATE AVAILABLE`-Spalte verfügbar sind.

2. Aktualisieren Sie das Add-On.

   1. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie nach Bedarf die folgenden Änderungen am Befehl vor:

      - Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.
      - *`region-code`*Ersetzen Sie es durch AWS-Region das, in dem sich Ihr Cluster befindet.
      - Ersetzen Sie *`vpc-cni`* durch den Namen eines Add-Ons, das in der Ausgabe des vorherigen Schritts zurückgegeben wurde, das Sie aktualisieren möchten.
      - Wenn Sie eine ältere Version als die neueste verfügbare Version aktualisieren möchten, dann ersetzen Sie *`latest`* durch die Versionsnummer, die in der Ausgabe des vorherigen Schritts zurückgegeben wurde, die Sie verwenden möchten. Für einige Add-Ons gibt es empfohlene Versionen. Weitere Informationen finden Sie in der Dokumentation für das Add-on, das Sie aktualisieren.
      - Wenn das Add-On ein Kubernetes-Servicekonto und eine IAM-Rolle verwendet, ersetzen Sie *`111122223333`* durch Ihre Konto-ID und *`role-name`* durch den Namen einer vorhandenen IAM-Rolle, die Sie erstellt haben. Anweisungen zum Erstellen der Rolle finden Sie in der Dokumentation für das Add-On, das Sie erstellen. Die Angabe einer Servicekonto-Rolle erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen für Ihren Cluster verfügen, oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.

        Wenn das Add-On kein Kubernetes-Servicekonto und keine IAM-Rolle verwendet, löschen Sie die **`serviceAccountRoleARN: arn:aws:iam::`*`111122223333`*`:role/`*`role-name`*** -Zeile.
      - Die *`preserve`* Option behält die vorhandenen Werte für das Add-on bei. Wenn Sie benutzerdefinierte Werte für Zusatzeinstellungen festgelegt haben und diese Option nicht verwenden, überschreibt Amazon EKS Ihre Werte mit seinen Standardwerten.

Wenn Sie diese Option verwenden, empfehlen wir, dass Sie alle Feld- und Wertänderungen auf einem Nicht-Produktionscluster testen, bevor Sie das Add-on auf Ihrem Produktionscluster aktualisieren. Wenn Sie diesen Wert auf `overwrite` ändern, werden alle Einstellungen auf die Amazon-EKS-Standardwerte geändert. Wenn Sie benutzerdefinierte Werte für Einstellungen festgelegt haben, werden diese möglicherweise mit den Amazon-EKS-Standardwerten überschrieben. Wenn Sie diesen Wert auf none ändern, ändert Amazon EKS den Wert der Einstellungen nicht, aber das Update schlägt möglicherweise fehl. Wenn das Update fehlschlägt, erhalten Sie eine Fehlermeldung, die Sie bei der Behebung des Konflikts unterstützt.

```
cat >update-addon.yaml <<EOF
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: my-cluster
  region: region-code

addons:
- name: vpc-cni
  version: latest
  serviceAccountRoleARN: arn:aws:iam::111122223333:role/role-name
  resolveConflicts: preserve
EOF
```

2. Führen Sie den bearbeiteten Befehl aus, um die `update-addon.yaml`-Datei zu erstellen.
3. Wenden Sie die Konfigurationsdatei auf Ihren Cluster an.

```
eksctl update addon -f update-addon.yaml
```

Weitere Informationen zum Aktualisieren von Add-Ons finden Sie unter Add-Ons in der `eksctl`-Dokumentation.

AWS Management Console

So aktualisieren Sie ein Amazon EKS-Add-on mit dem AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unterhttps://console.aws.amazon.com/eks/home#/clusters.

2. Wählen Sie im linken Navigationsbereich Clusters (Cluster) aus. Wählen Sie anschließend den Namen des Clusters aus, für den Sie das Add-On konfigurieren möchten.
3. Wählen Sie die Registerkarte Add-ons.
4. Wählen Sie das Kästchen oben rechts in der Add-On-Box aus und wählen Sie dann Edit (Bearbeiten).
5. Gehen Sie auf der Seite Configure ***name of addon*** (Namen des Add-Ons konfigurieren) wie folgt vor:

   - Wählen Sie die Version aus, die Sie verwenden möchten. Das Add-On verfügt möglicherweise über eine empfohlene Version. Weitere Informationen finden Sie in der Dokumentation für das Add-on, das Sie aktualisieren.
   - Für die IAM-Rolle auswählen können Sie die IAM-Rolle des Knotens (nicht festgelegt) oder eine vorhandene Rolle verwenden, die Sie für die Verwendung mit dem Add-on erstellt haben. Wenn Sie keine Rolle auswählen können, verfügen Sie über keine vorhandene Rolle. Unabhängig davon, für welche Option Sie sich entscheiden, lesen Sie die Dokumentation für das Add-on, das Sie erstellen, um eine IAM-Richtlinie zu erstellen und sie einer Rolle anzufügen. Die Auswahl einer IAM-Rolle erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen für Ihren Cluster verfügen, oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.
   - Geben Sie für `Code editor` beliebige Add-On-spezifische Konfigurationsinformationen ein. Weitere Informationen finden Sie in der Dokumentation für das Add-on, das Sie aktualisieren.
   - Wählen Sie für die Conflict resolution method (Methode zur Konfliktlösung) eine der Optionen aus. Wenn Sie benutzerdefinierte Werte für Add-On-Einstellungen festgelegt haben, empfehlen wir die Option Preserve (Beibehalten). Wenn Sie diese Option nicht auswählen, überschreibt Amazon EKS Ihre Werte mit seinen Standardwerten. Wenn Sie diese Option verwenden, empfehlen wir, dass Sie alle Feld- und Wertänderungen auf einem Nicht-Produktionscluster testen, bevor Sie das Add-On auf Ihrem Produktionscluster aktualisieren.
6. Wählen Sie Aktualisieren.

AWS CLI

Voraussetzung

Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder. AWS CloudShell Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.

So aktualisieren Sie ein Amazon EKS-Add-on mit dem AWS CLI

1. Hier finden Sie eine Liste der installierten Add-Ons. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

   ```
   aws eks list-addons --cluster-name my-cluster
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   {
       "addons": [
           "coredns",
           "kube-proxy",
           "vpc-cni"
       ]
   }
   ```

2. Zeigen Sie die aktuelle Version des Add-Ons an, die Sie aktualisieren möchten. Ersetzen Sie *`my-cluster`* mit Ihrem Clusternamen und *`vpc-cni`* mit dem Namen des Add-Ons, das Sie aktualisieren möchten.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
    "addon.addonVersion" --output text
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v1.10.4-eksbuild.1
   ```

3. Sie können anzeigen, welche Versionen des Add-Ons für die Version Ihres Clusters verfügbar sind. Ersetzen Sie *1.30* mit Ihrer Cluster-Version und *vpc-cni* mit dem Namen des Add-Ons, das Sie aktualisieren möchten.

```
aws eks describe-addon-versions --kubernetes-version 1.30 --addon-name vpc-cni \
    --query 'addons[].addonVersions[].{Version: addonVersion, Defaultversion:
 compatibilities[0].defaultVersion}' --output table
```

Eine Beispielausgabe sieht wie folgt aus.

```
-----------------------------------------
|          DescribeAddonVersions         |
+-----------------+----------------------+
| Defaultversion  |       Version        |
+-----------------+----------------------+
|  False          |  v1.12.0-eksbuild.1  |
|  True           |  v1.11.4-eksbuild.1  |
|  False          |  v1.10.4-eksbuild.1  |
|  False          |  v1.9.3-eksbuild.1   |
+-----------------+----------------------+
```

Die Version `True` in der `Defaultversion`-Spalte ist standardmäßig die Version, mit der das Add-On erstellt wurde.

4. Aktualisieren Sie Ihr Add-on. Kopieren Sie den folgenden Befehl auf Ihr Gerät. Nehmen Sie bei Bedarf die folgenden Änderungen am Befehl vor, und führen Sie dann den geänderten Befehl aus.

- Ersetzen Sie *my-cluster* mit dem Namen Ihres Clusters.
- Ersetzen Sie *vpc-cni* durch den Namen des Add-Ons, das Sie aktualisieren möchten und das in der Ausgabe eines vorherigen Schritts zurückgegeben wurde.
- Ersetzen Sie *version-number* durch die in der Ausgabe des vorherigen Schritts zurückgegebene Version, auf die Sie aktualisieren möchten. Für einige Add-Ons gibt es empfohlene Versionen. Weitere Informationen finden Sie in der Dokumentation für das Add-on, das Sie aktualisieren.
- Wenn das Add-On ein Kubernetes-Servicekonto und eine IAM-Rolle verwendet, ersetzen Sie *111122223333* durch Ihre Konto-ID und *role-name* durch den Namen einer vorhandenen IAM-Rolle, die Sie erstellt haben. Anweisungen zum Erstellen der Rolle finden Sie in der Dokumentation für das Add-On, das Sie erstellen. Die Angabe einer

Servicekonto-Rolle erfordert, dass Sie über einen IAM OpenID Connect (OIDC)-Anbieter für Ihren Cluster verfügen. Um festzustellen, ob Sie über einen solchen für Ihren Cluster verfügen, oder um einen zu erstellen, lesen Sie Erstellen Sie einen OIDC IAM-Anbieter für Ihren Cluster.

Wenn das Add-On kein Kubernetes-Servicekonto und keine IAM-Rolle verwendet, löschen Sie die **`serviceAccountRoleARN: arn:aws:iam::`*`111122223333`*`:role/`*`role-name`*-Zeile.

- Die **`--resolve-conflicts`** *`PRESERVE`*-Option behält die vorhandenen Werte für das Add-On bei. Wenn Sie benutzerdefinierte Werte für Zusatzeinstellungen festgelegt haben und diese Option nicht verwenden, überschreibt Amazon EKS Ihre Werte mit seinen Standardwerten. Wenn Sie diese Option verwenden, empfehlen wir, dass Sie alle Feld- und Wertänderungen auf einem Nicht-Produktionscluster testen, bevor Sie das Add-on auf Ihrem Produktionscluster aktualisieren. Wenn Sie diesen Wert auf `overwrite` ändern, werden alle Einstellungen auf die Amazon-EKS-Standardwerte geändert. Wenn Sie benutzerdefinierte Werte für Einstellungen festgelegt haben, werden diese möglicherweise mit den Amazon-EKS-Standardwerten überschrieben. Wenn Sie diesen Wert auf `none` ändern, ändert Amazon EKS den Wert der Einstellungen nicht, aber das Update schlägt möglicherweise fehl. Wenn das Update fehlschlägt, erhalten Sie eine Fehlermeldung, die Sie bei der Behebung des Konflikts unterstützt.

- Wenn Sie die gesamte benutzerdefinierte Konfiguration entfernen möchten, führen Sie das Update mit der *`--configuration-values '{}'`*-Option durch. Dadurch werden alle benutzerdefinierten Konfigurationen auf die Standardwerte zurückgesetzt. Wenn Sie Ihre benutzerdefinierte Konfiguration nicht ändern möchten, geben Sie das *`--configuration-values`*-Flag nicht an. Wenn Sie eine benutzerdefinierte Konfiguration anpassen möchten, ersetzen Sie *`{}`* durch die neuen Parameter. Eine Liste der Parameter finden Sie unter Konfigurationsschema anzeigen im Abschnitt zum Erstellen eines Add-Ons.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version version-number \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name --
configuration-values '{}' --resolve-conflicts PRESERVE
```

5. Überprüfen Sie den Status des Updates. Ersetzen Sie *my-cluster* mit Ihrem Clusternamen und *vpc-cni* mit dem Namen des Add-Ons, das Sie aktualisieren möchten.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "addon": {
        "addonName": "vpc-cni",
        "clusterName": "my-cluster",
        "status": "UPDATING",
[...]
```

Das Update ist abgeschlossen, wenn der Status `ACTIVE` ist.

Löschen eines Add-Ons

Wenn Sie ein Amazon-EKS-Add-On löschen:

- Es gibt keine Ausfallzeit für die Funktionalität, die das Add-On bereitstellt.
- Wenn Sie IAM-Rollen for Service Accounts (IRSA) verwenden und dem Add-on eine IAM-Rolle zugeordnet ist, wird die IAM-Rolle nicht entfernt.
- Wenn Sie Pod-Identitäten verwenden, werden alle Pod-Identitätszuordnungen gelöscht, die dem Add-on gehören. Wenn Sie die `--preserve` Option für angeben AWS CLI, werden die Zuordnungen beibehalten.
- Amazon EKS beendet die Verwaltung der Einstellungen für das Add-On.
- Die Konsole benachrichtigt Sie nicht mehr, wenn neue Versionen verfügbar sind.
- Sie können das Add-on nicht mithilfe von AWS Tools oder APIs aktualisieren.
- Sie können die Add-On-Software auf Ihrem Cluster beibehalten, damit Sie sie selbst verwalten können, oder Sie können die Add-On-Software aus Ihrem Cluster entfernen. Sie sollten die Add-On-Software nur dann aus Ihrem Cluster entfernen, wenn keine Ressourcen in Ihrem Cluster von der Funktionalität abhängen, die das Add-On bereitstellt.

Sie können ein Amazon-EKS-Add-On mit `eksctl`, AWS Management Console oder AWS CLI aus Ihrem Cluster löschen.

eksctl

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

So löschen Sie ein Amazon-EKS-Add-On mit **`eksctl`**

1. Bestimmen Sie die aktuell auf Ihrem Cluster installierten Add-Ons. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

   ```
   eksctl get addon --cluster my-cluster
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   NAME        VERSION              STATUS  ISSUES  IAMROLE  UPDATE AVAILABLE
   coredns     v1.8.7-eksbuild.2    ACTIVE  0
   kube-proxy  v1.23.7-eksbuild.1   ACTIVE  0
   vpc-cni     v1.10.4-eksbuild.1   ACTIVE  0
   [...]
   ```

   Ihre Ausgabe sieht möglicherweise anders aus, je nachdem, welche Add-Ons und Versionen Sie auf Ihrem Cluster haben.
2. Löschen Sie das Add-On. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters und *`name-of-add-on`* mit dem Namen des Add-Ons, das in der Ausgabe des vorherigen Schritts zurückgegeben wurde. Wenn Sie die ***`--preserve`***-Option entfernen, wird zusätzlich dazu, dass Amazon EKS das Add-On nicht mehr verwaltet, die Add-On-Software aus Ihrem Cluster entfernt.

   ```
   eksctl delete addon --cluster my-cluster --name name-of-addon --preserve
   ```

AWS Management Console

Um ein Amazon EKS-Add-on mit dem AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

2. Wählen Sie im linken Navigationsbereich Clusters (Cluster) aus. Wählen Sie dann den Namen des Clusters aus, für den Sie das Amazon-EKS-Add-on entfernen möchten.
3. Wählen Sie die Registerkarte Add-ons.
4. Aktivieren Sie das Kontrollkästchen oben rechts im Feld Add-On und wählen Sie dann Entfernen. Wählen Sie Preserve on the cluster (Auf dem Cluster beibehalten) aus, wenn Sie möchten, dass Amazon EKS die Verwaltung der Einstellungen für das Add-on beendet, aber die Add-on-Software auf Ihrem Cluster beibehalten möchten, damit Sie alle Einstellungen für das Add-on selbst verwalten können. Geben Sie den Namen des Add-Ons ein und wählen Sie dann Entfernen aus.

AWS CLI

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

Um ein Amazon EKS-Add-on mit dem AWS CLI

1. Hier finden Sie eine Liste der installierten Add-Ons. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters.

```
aws eks list-addons --cluster-name my-cluster
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "addons": [
        "coredns",
        "kube-proxy",
        "vpc-cni",
        "name-of-addon"
    ]
}
```

2. Löschen Sie das installierte Add-On. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und *`name-of-add-on`* durch den Namen des Add-Ons, das Sie entfernen. Durch das Entfernen von ***`--preserve`*** wird die Add-on-Software aus Ihrem Cluster entfernt.

```
aws eks delete-addon --cluster-name my-cluster --addon-name name-of-addon --
preserve
```

Die gekürzte Beispielausgabe lautet wie folgt.

```
{
    "addon": {
        "addonName": "name-of-add-on",
        "clusterName": "my-cluster",
        "status": "DELETING",
[...]
```

3. Überprüfen Sie den Status der Löschung. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters und *name-of-addon* durch den Namen des Add-Ons, das Sie entfernen.

```
aws eks describe-addon --cluster-name my-cluster --addon-name name-of-addon
```

Nachdem das Add-on gelöscht wurde, sieht die Beispielausgabe wie folgt aus.

```
An error occurred (ResourceNotFoundException) when calling the DescribeAddon
 operation: No addon: name-of-addon found in cluster: my-cluster
```

Rufen Sie die Kompatibilität der Addon-Version ab

Verwenden Sie die `describe-addon-verisions`API, um die verfügbaren Versionen von EKS-Addons aufzulisten und aufzulisten, welche Kubernetes-Versionen jede Addon-Version unterstützt.

Rufen Sie die Kompatibilität der Addon-Version ab ()AWS CLI

1. Stellen Sie sicher, AWS CLI dass das installiert ist und mit `aws sts get-caller-identity` funktioniert. Wenn dieser Befehl nicht funktioniert, erfahren Sie, wie Sie mit dem beginnen AWS CLI.
2. Ermitteln Sie den Namen des Addons, für das Sie Versionskompatibilitätsinformationen abrufen möchten, z. B. `amazon-cloudwatch-observability`
3. Ermitteln Sie die Kubernetes-Version Ihres Clusters, z. B. `1.28`
4. Verwenden Sie die AWS CLI , um die Addon-Versionen abzurufen, die mit der Kubernetes-Version Ihres Clusters kompatibel sind.

```
aws eks describe-addon-versions --addon-name amazon-cloudwatch-observability --
kubernetes-version 1.29
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "addons": [
        {
            "addonName": "amazon-cloudwatch-observability",
            "type": "observability",
            "addonVersions": [
                {
                    "addonVersion": "v1.5.0-eksbuild.1",
                    "architecture": [
                        "amd64",
                        "arm64"
                    ],
                    "compatibilities": [
                        {
                            "clusterVersion": "1.28",
                            "platformVersions": [
                                "*"
                            ],
                            "defaultVersion": true
                        }
                    ],
[...]
```

Diese Ausgabe zeigt, dass die Addon-Version mit der `v1.5.0-eksbuild.1` Kubernetes-Cluster-Version kompatibel ist. `1.28`

Kubernetes-Feldverwaltung

Amazon-EKS-Add-ons werden in Ihrem Cluster unter Verwendung von bewährten Standardkonfigurationen installiert. Weitere Informationen zum Hinzufügen eines Amazon-EKS-Add-ons zu Ihrem Cluster finden Sie unter Amazon-EKS-Add-ons.

Möglicherweise möchten Sie die Konfiguration eines Amazon-EKS-Add-ons anpassen, um erweiterte Funktionen zu aktivieren. Amazon EKS verwendet die serverseitige Funktion Kubernetes, um

die Verwaltung eines Add-ons durch Amazon EKS zu ermöglichen, ohne Ihre Konfiguration für Einstellungen zu überschreiben, die nicht von Amazon EKS verwaltet werden. Weitere Informationen finden Sie unter Server-side Apply? in der Kubernetes-Dokumentation. Um dies zu erreichen, verwaltet Amazon EKS einen Mindestsatz von Feldern für jedes Add-on, das installiert wird. Sie können alle Felder problemlos ändern, die nicht von Amazon EKS oder einem anderen Prozess der Kubernetes-Steuerebene verwaltet werden, z. B. `kube-controller-manager`.

> **Important**
>
> Das Ändern eines von Amazon EKS verwalteten Felds verhindert, dass Amazon EKS das Add-on verwaltet und kann dazu führen, dass Ihre Änderungen überschrieben werden, wenn ein Add-on aktualisiert wird.

Status der Feldverwaltung anzeigen

Sie können `kubectl` verwenden, um zu sehen, welche Felder von Amazon EKS für ein Amazon EKS Add-on verwaltet werden.

Verwaltungsstatus eines Felds anzeigen

1. Bestimmen Sie das Add-on, das Sie überprüfen möchten. Informationen zu allen `deployments` und `DaemonSets`, die in Ihrem Cluster bereitgestellt werden, finden Sie unter Anzeigen der Kubernetes-Ressourcen.
2. Führen Sie den folgenden Befehl aus, um die verwalteten Felder für ein Add-on anzuzeigen:

   ```
   kubectl get type/add-on-name -n add-on-namespace -o yaml
   ```

   Sie können beispielsweise die verwalteten Felder für das CoreDNS-Add-on mit dem folgenden Befehl sehen.

   ```
   kubectl get deployment/coredns -n kube-system -o yaml
   ```

   Die Feldverwaltung wird im folgenden Abschnitt in der zurückgegebenen Ausgabe aufgeführt.

   ```
   [...]
   managedFields:
     - apiVersion: apps/v1
   ```

```
    fieldsType: FieldsV1
    fieldsV1:
[...]
```

> **Note**
>
> Wenn `managedFields` in der Ausgabe nicht angezeigt wird, fügen Sie dem Befehl **`--show-managed-fields`** hinzu, und führen Sie ihn erneut aus. Die Version von `kubectl`, die Sie verwenden, bestimmt, ob verwaltete Felder standardmäßig zurückgegeben werden.

Grundlegendes zur Feldverwaltungssyntax in der Kubernetes API

Wenn Sie Details für ein Kubernetes-Objekt anzeigen, werden sowohl verwaltete als auch nicht verwaltete Felder in der Ausgabe zurückgegeben. Verwaltete Felder können einer der folgenden Typen sein:

- Vollständig verwaltet – Alle Schlüssel für das Feld werden von Amazon EKS verwaltet. Änderungen an einem beliebigen Wert verursachen einen Konflikt.
- Teilweise verwaltet – Einige Schlüssel für das Feld werden von Amazon EKS verwaltet. Nur Änderungen an den Schlüsseln, die explizit von Amazon EKS verwaltet werden, verursachen einen Konflikt.

Beide Arten von Feldern sind gekennzeichnet mit `manager: eks`.

Jeder Schlüssel ist entweder ein `.`, welches das Feld darstellt, das immer einem leeren Satz zugeordnet wird, oder eine Zeichenfolge, die ein Unterfeld oder Element darstellt. Die Ausgabe für die Feldverwaltung besteht aus folgenden Angabetypen:

- `f:`*`name`*, wobei *`name`* der Name eines Feldes in einer Liste ist.
- `k:`*`keys`*, wobei *`keys`* eine Karte der Felder eines Listenelements ist.
- `v:`*`value`*, wobei *`value`* der exakte JSON-formatierte Wert eines Listenelements ist.
- `i:`*`index`*, wobei *`index`* die Position eines Elements in der Liste ist.

Die folgenden Ausgabeteile für das CoreDNS-Add-on veranschaulichen die vorherigen Angaben:

- Vollständig verwaltete Felder – Wenn ein verwaltetes Feld ein `f:` (Feld) festgelegt hat, aber keinen `k:` (Schlüssel), dann wird das gesamte Feld verwaltet. Änderungen an Werten in diesem Feld verursachen einen Konflikt.

  In der folgenden Ausgabe sehen Sie, dass der Container mit dem Namen `coredns` verwaltet von `eks`. Die`args`, `image` und `imagePullPolicy` Unterfelder werden auch verwaltet von `eks`. Änderungen an Werten in diesen Feldern verursachen einen Konflikt.

  ```
  [...]
  f:containers:
    k:{"name":"coredns"}:
    .: {}
    f:args: {}
    f:image: {}
    f:imagePullPolicy: {}
  [...]
  manager: eks
  [...]
  ```

- Teilweise verwaltete Felder – Wenn für einen verwalteten Schlüssel ein Wert angegeben ist, werden die angegebenen Schlüssel für dieses Feld verwaltet. Das Ändern der angegebenen Schlüssel führt zu einem Konflikt.

  In der folgenden Ausgabe sehen Sie, dass `eks` die `config-volume` und `tmp` Volumes verwaltet, welche mit dem `name`-Schlüssel eingestellt wurden.

  ```
  [...]
  f:volumes:
    k:{"name":"config-volume"}:
      .: {}
      f:configMap:
        f:items: {}
        f:name: {}
      f:name: {}
    k:{"name":"tmp"}:
      .: {}
      f:name: {}
  [...]
  manager: eks
  [...]
  ```

- Hinzufügen von Schlüsseln zu teilweise verwalteten Feldern – Wenn nur ein bestimmter Schlüsselwert verwaltet wird, können Sie sicher zusätzliche Schlüssel wie Argumente zu einem Feld hinzufügen, ohne einen Konflikt zu verursachen. Wenn Sie zusätzliche Schlüssel hinzufügen, stellen Sie sicher, dass das Feld nicht zuerst verwaltet wird. Das Hinzufügen oder Ändern eines verwalteten Werts verursacht einen Konflikt.

  In der folgenden Ausgabe sehen Sie, dass sowohl der Schlüssel `name` als auch das Feld `name` verwaltet werden. Das Hinzufügen oder Ändern eines Containernamens verursacht einen Konflikt mit diesem verwalteten Schlüssel.

```
[...]
f:containers:
  k:{"name":"coredns"}:
[...]
    f:name: {}
[...]
manager: eks
[...]
```

Hängen Sie mithilfe von Pod Identity eine IAM-Rolle an ein Amazon EKS-Add-on an

Bestimmte Amazon EKS-Add-Ons benötigen IAM-Rollenberechtigungen, um AWS APIs aufzurufen. Das Amazon VPC CNI-Add-on ruft beispielsweise bestimmte AWS APIs auf, um Netzwerkressourcen in Ihrem Konto zu konfigurieren. Diesen Add-Ons muss eine Genehmigung mithilfe AWS von IAM erteilt werden. Insbesondere muss das Dienstkonto des Pods, auf dem das Add-on ausgeführt wird, einer IAM-Rolle mit einer ausreichenden IAM-Richtlinie zugeordnet sein.

Die empfohlene Methode zur Erteilung von AWS Berechtigungen für Cluster-Workloads ist die Verwendung der Amazon EKS-Funktion Pod Identities. Sie können eine Pod Identity Association verwenden, um das Dienstkonto eines Add-ons einer IAM-Rolle zuzuordnen. Wenn ein Pod ein Servicekonto mit einer Zuordnung verwendet, legt Amazon EKS Umgebungsvariablen in den Containern des Pods fest. Die Umgebungsvariablen konfigurieren die AWS SDKs, einschließlich der AWS CLI, so, dass sie die EKS Pod Identity-Anmeldeinformationen verwenden. Erfahren Sie mehr über EKS Pod Identities.

Amazon EKS-Add-Ons können dabei helfen, den Lebenszyklus von Pod-Identitätszuordnungen zu verwalten, die dem Add-on entsprechen. Sie können beispielsweise ein Amazon EKS-Add-on und

die erforderliche Pod-Identitätszuordnung in einem einzigen API-Aufruf erstellen oder aktualisieren. Amazon EKS bietet auch eine API zum Abrufen von vorgeschlagenen IAM-Richtlinien.

Vorgeschlagene Verwendung:

1. Vergewissern Sie sich, dass der Amazon EKS Pod Identity Agent auf Ihrem Cluster eingerichtet ist.
2. Stellen Sie mithilfe des `describe-addon-versions` AWS CLI Vorgangs fest, ob für das Add-on, das Sie installieren möchten, IAM-Berechtigungen erforderlich sind. Wenn das `requiresIamPermissions` Kennzeichen aktiviert ist`true`, sollten Sie den `describe-addon-configurations` Vorgang verwenden, um die für das Addon benötigten Berechtigungen zu ermitteln. Die Antwort enthält eine Liste der vorgeschlagenen verwalteten IAM-Richtlinien.
3. Rufen Sie den Namen des Kubernetes-Dienstkontos und die vorgeschlagene IAM-Richtlinie mithilfe der `describe-addon-configuration` CLI-Operation ab. Vergleichen Sie den Umfang der vorgeschlagenen Richtlinie anhand Ihrer Sicherheitsanforderungen.
4. Erstellen Sie eine IAM-Rolle mithilfe der vorgeschlagenen Berechtigungsrichtlinie und der von Pod Identity geforderten Vertrauensrichtlinie. Weitere Informationen finden Sie unter Erstellen der EKS-Pod-Identity-Zuordnung.
5. Erstellen oder aktualisieren Sie ein Amazon EKS-Add-on mithilfe der CLI. Geben Sie mindestens eine Pod-Identitätszuordnung an. Eine Pod-Identitätszuordnung ist (1) der Name eines Kubernetes-Dienstkontos und (2) der ARN einer IAM-Rolle.

Überlegungen:

- Pod-Identitätszuordnungen, die mithilfe der Add-On-APIs erstellt wurden, gehören dem jeweiligen Add-on. Wenn Sie das Add-on löschen, wird auch die Pod-Identitätszuordnung gelöscht. Sie können dieses kaskadierende Löschen verhindern, indem Sie die `preserve` Option beim Löschen eines Addons mithilfe der AWS CLI oder -API verwenden. Sie können die Pod-Identitätszuordnung bei Bedarf auch direkt aktualisieren oder löschen. Add-Ons können nicht das Eigentum an bestehenden Pod-Identitätszuordnungen übernehmen. Sie müssen die bestehende Zuordnung löschen und sie mithilfe eines Add-On-Erstellungs- oder Aktualisierungsvorgangs neu erstellen.
- Amazon EKS empfiehlt die Verwendung von Pod-Identitätszuordnungen zur Verwaltung von IAM-Berechtigungen für Add-Ons. Die vorherige Methode, IAM-Rollen für Dienstkonten (IRSA), wird weiterhin unterstützt. Sie können für ein Add-On sowohl eine IRSA `serviceAccountRoleArn` - als auch eine Pod-Identitätszuordnung angeben. Wenn der EKS-Pod-Identity-Agent auf dem Cluster installiert ist, `serviceAccountRoleArn` wird er ignoriert und EKS

verwendet die bereitgestellte Pod-Identitätszuordnung. Wenn Pod Identity nicht aktiviert ist, `serviceAccountRoleArn` wird der verwendet.

- Wenn Sie die Pod-Identitätszuordnungen für ein vorhandenes Add-on aktualisieren, leitet Amazon EKS einen fortlaufenden Neustart der Add-On-Pods ein.

Rufen Sie IAM-Informationen zu einem Add-on ab

Mithilfe von können Sie ermitteln AWS CLI , (1) ob für ein Add-on IAM-Berechtigungen erforderlich sind, und (2) eine vorgeschlagene IAM-Richtlinie für dieses Add-on angeben.

IAM-Informationen zu einem Amazon EKS-Add-on abrufen ()AWS CLI

1. Ermitteln Sie den Namen des Add-ons, das Sie installieren möchten, und die Kubernetes-Version Ihres Clusters. Erfahren Sie mehr über verfügbare Amazon EKS Add-ons.
2. Verwenden Sie den AWS CLI , um festzustellen, ob für das Add-on IAM-Berechtigungen erforderlich sind.

   ```
   aws eks describe-addon-versions \
   --addon-name <addon-name> \
   --kubernetes-version <kubernetes-version>
   ```

   Beispielsweise:

   ```
   aws eks describe-addon-versions \
   --addon-name aws-ebs-csi-driver \
   --kubernetes-version 1.30
   ```

   Sehen Sie sich die folgende Beispielausgabe an. Beachten Sie, dass `requiresIamPermissions` dies `true` und die Standard-Add-On-Version sind. Sie müssen die Add-On-Version angeben, wenn Sie die empfohlene IAM-Richtlinie abrufen.

   ```
   {
       "addons": [
           {
               "addonName": "aws-ebs-csi-driver",
               "type": "storage",
               "addonVersions": [
                   {
   ```

```
                    "addonVersion": "v1.31.0-eksbuild.1",
                    "architecture": [
                        "amd64",
                        "arm64"
                    ],
                    "compatibilities": [
                        {
                            "clusterVersion": "1.30",
                            "platformVersions": [
                                "*"
                            ],
                            "defaultVersion": true
                        }
                    ],
                    "requiresConfiguration": false,
                    "requiresIamPermissions": true
                },
[...]
```

3. Wenn für das Add-on IAM-Berechtigungen erforderlich sind, verwenden Sie die, AWS CLI um eine empfohlene IAM-Richtlinie abzurufen.

```
aws eks describe-addon-configuration \
--query podIdentityConfiguration \
--addon-name <addon-name> \
--addon-version <addon-version>
```

Beispielsweise:

```
aws eks describe-addon-configuration \
--query podIdentityConfiguration \
--addon-name aws-ebs-csi-driver \
--addon-version v1.31.0-eksbuild.1
```

Überprüfen Sie die folgende Ausgabe. Beachten Sie den `recommendedManagedPolicies`.

```
[
    {
        "serviceAccount": "ebs-csi-controller-sa",
        "recommendedManagedPolicies": [
            "arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy"
        ]
```

```
    }
]
```

4. Erstellen Sie eine IAM-Rolle und fügen Sie die empfohlene verwaltete Richtlinie hinzu. Sie können auch die verwaltete Richtlinie überprüfen und den Umfang der Berechtigungen entsprechend einschränken. Lesen Sie die Anweisungen zum Erstellen einer IAM-Rolle zur Verwendung mit EKS Pod Identities.

Aktualisieren Sie das Add-on mit der IAM-Rolle

Aktualisieren Sie ein Amazon EKS-Add-on, um eine Pod Identity Association zu verwenden (AWS CLI)

1. Ermitteln Sie:
   - `cluster-name`— Der Name des EKS-Clusters, auf dem das Add-on installiert werden soll.
   - `addon-name`— Der Name des zu installierenden Amazon EKS-Add-ons.
   - `service-account-name`— Der Name des Kubernetes-Servicekontos, das vom Add-on verwendet wird.
   - `iam-role-arn`— Der ARN einer IAM-Rolle mit ausreichenden Berechtigungen für das Add-on. Die IAM-Rolle muss über die erforderliche Vertrauensrichtlinie für EKS Pod Identity verfügen.
2. Aktualisieren Sie das Add-on mit der AWS CLI. Sie können beim Erstellen eines Add-Ons auch Pod-Identitätszuordnungen angeben, indem Sie dieselbe `--pod-identity-assocations` Syntax verwenden. Beachten Sie, dass, wenn Sie beim Aktualisieren eines Add-ons Pod-Identitätszuordnungen angeben, alle vorherigen Pod-Identitätszuordnungen überschrieben werden.

   ```
   aws eks update-addon --cluster-name <cluster-name> \
   --addon-name <addon-name> \
   --pod-identity-associations 'serviceAccount=<service-account-name>,roleArn=<role-arn>'
   ```

   Beispielsweise:

   ```
   aws eks update-addon --cluster-name mycluster \
   --addon-name aws-ebs-csi-driver \
   ```

```
--pod-identity-associations 'serviceAccount=ebs-csi-controller-
sa,roleArn=arn:aws:iam::123456789012:role/StorageDriver'
```

3. Stellen Sie sicher, dass die Pod-Identitätszuordnung erstellt wurde:

```
aws eks list-pod-identity-associations --cluster-name <cluster-name>
```

Wenn Sie erfolgreich waren, sollte die Ausgabe folgendermaßen oder ähnlich aussehen. Notieren Sie sich den OwnerARN des EKS-Add-Ons.

```
{
    "associations": [
        {
            "clusterName": "mycluster",
            "namespace": "kube-system",
            "serviceAccount": "ebs-csi-controller-sa",
            "associationArn": "arn:aws:eks:us-
west-2:123456789012:podidentityassociation/mycluster/a-4wvljrezsukshq1bv",
            "associationId": "a-4wvljrezsukshq1bv",
            "ownerArn": "arn:aws:eks:us-west-2:123456789012:addon/mycluster/aws-
ebs-csi-driver/9cc7ce8c-2e15-b0a7-f311-426691cd8546"
        }
    ]
}
```

Verknüpfungen aus dem Add-on entfernen

Alle Pod-Identitätszuordnungen aus einem Amazon EKS-Add-on entfernen (AWS CLI)

1. Ermitteln Sie:
   - `cluster-name`— Der Name des EKS-Clusters, auf dem das Add-on installiert werden soll.
   - `addon-name`— Der Name des zu installierenden Amazon EKS-Add-ons.
2. Aktualisieren Sie das Addon, um ein leeres Array von Pod-Identitätszuordnungen anzugeben.

```
aws eks update-addon --cluster-name <cluster-name> \
--addon-name <addon-name> \
--pod-identity-associations "[]"
```

Problembehandlung bei Pod-Identitäten für EKS-Add-Ons

Wenn bei Ihren Add-Ons beim Versuch von AWS API-, SDK- oder CLI-Vorgängen Fehler auftreten, überprüfen Sie Folgendes:

- Der Pod Identity Agent ist in Ihrem Cluster installiert.
  - Lesen Sie, wie Sie den Pod Identity Agent einrichten.
- Das Add-on hat eine gültige Pod-Identitätszuordnung.
  - Verwenden Sie die AWS CLI , um die Zuordnungen für den vom Add-on verwendeten Dienstkontonamen abzurufen.

```
aws eks list-pod-identity-associations --cluster-name <cluster-name>
```

- Die vorgesehene IAM-Rolle verfügt über die erforderliche Vertrauensrichtlinie für EKS-Pod-Identitäten.
  - Verwenden Sie die AWS CLI , um die Vertrauensrichtlinie für ein Add-on abzurufen.

```
aws iam get-role --role-name <role-name> --query Role.AssumeRolePolicyDocument
```

- Die vorgesehene IAM-Rolle verfügt über die erforderlichen Berechtigungen für das Add-on.
  - Wird AWS CloudTrail zur Überprüfung `AccessDenied` von `UnauthorizedOperation` Ereignissen verwendet.
- Der Dienstkontoname in der Pod-Identitätszuordnung entspricht dem vom Add-on verwendeten Dienstkontonamen.
  - Lesen Sie in der Dokumentation für das Add-on nach, um den Namen des Dienstkontos zu ermitteln.

Überprüfen eines Container-Images bei der Bereitstellung

Wenn Sie AWS Signer nutzen und zum Zeitpunkt der Bereitstellung signierte Container-Images verifizieren möchten, können Sie auf eine der folgenden Lösungen zurückgreifen:

- Gatekeeper und Ratify – Verwenden Sie Gatekeeper als Zugangscontroller und Ratify (mit einem konfigurierten AWS Signer-Plugin) als Webhook für die Validierung von Signaturen.
- Kyverno – Eine Kubernetes-Richtlinien-Engine, die mit einem AWS Signer-Plugin zur Validierung von Signaturen konfiguriert ist.

> **Note**
>
> Bevor Sie die Signaturen von Container-Images verifizieren, müssen Sie den Trust Store und die Vertrauensrichtlinie für die Notation entsprechend den Anforderungen des ausgewählten Zugangscontrollers konfigurieren.

Machine Learning-Training mit Elastic Fabric Adapter

In diesem Thema wird beschrieben, wie Elastic Fabric Adapter (EFA) in Pods integriert werden, die in Ihrem Amazon-EKS-Cluster bereitgestellt werden. Elastic Fabric Adapter (EFA) ist eine Netzwerkschnittstelle für Amazon-EC2-Instances, mit der Sie Anwendungen ausführen können, die eine hohe Kommunikation zwischen den Knoten erfordern in einer Größenordnung von AWS Die speziell für die Umgehung von Betriebssystemen entwickelte Hardware-Schnittstelle verbessert die Leistung der Instance-übergreifenden Kommunikation, was für die Skalierung dieser Anwendungen von entscheidender Bedeutung ist. Mit EFA können High Performance Computing (HPC)-Anwendungen, die die Message Passing Interface (MPI) und Machine Learning (ML)-Anwendungen verwenden, die NVIDIA Collective Communications Library (NCCL) verwenden, auf Tausende von CPUs oder GPUs skaliert werden. Dadurch erhalten Sie die Anwendungsleistung von lokalen HPC-Clustern mit der On-Demand-Elastizität und Flexibilität der AWS Cloud. Durch die Integration von EFA in Anwendungen, die auf Amazon-EKS-Clustern ausgeführt werden, können Sie die Zeit für die Durchführung umfangreicher verteilter Schulungs-Workloads verkürzen, ohne dass zusätzliche Instances zu Ihrem Cluster hinzugefügt werden müssen. Weitere Informationen zu EFA finden Sie unter Elastic Fabric Adapter.

Das in diesem Thema beschriebene EFA-Plugin unterstützt Amazon-EC2–P4dInstances, die den aktuellen Stand der Technik beim verteilten Machine Learning in der Cloud darstellen. Jede `p4d.24xlarge`-Instance verfügt über acht NVIDIA A100-GPUs und 400 Gbit/s GPUDirectRDMA über EFA. GPUDirectRDMA ermöglicht Ihnen eine direkte GPU-zu-GPU-Kommunikation über Knoten mit CPU-Bypass, wodurch die kollektive Kommunikationsbandbreite erhöht und die Latenz verringert wird. Integration von Amazon EKS und EFA mit P4d-Instances bietet eine nahtlose Methode, um die Vorteile der leistungsstärksten Amazon-EC2-Computing-Instance für verteiltes Machine Learning-Training zu nutzen.

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Wenn Sie nicht über einen Cluster verfügen, verwenden Sie eines unsererErste Schritte mit Amazon EKS-Hilfslinien erstellen Sie eines. Ihr Cluster muss in

einer VPC bereitgestellt werden, in der mindestens ein privates Subnetz mit genügend verfügbaren IP-Adressen zum Bereitstellen von Knoten vorhanden sind. Das private Subnetz muss über einen ausgehenden Internetzugang verfügen, der von einem externen Gerät, z. B. einem NAT-Gateway, bereitgestellt wird.

Wenn Sie planen, `eksctl` zu verwenden, um Ihre Knotengruppe zu erstellen, kann `eksctl` auch einen Cluster für Sie erstellen.

- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder. AWS CloudShell Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- Sie müssen die Amazon VPC CNI plugin for Kubernetes-Version `1.7.10` installiert haben, bevor Sie Worker-Knoten starten, die mehrere Elastic Fabric Adapter unterstützen, z. B. `p4d.24xlarge`. Weitere Informationen zum Aktualisieren Ihrer Amazon VPC CNI plugin for Kubernetes-Version finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on.

Knotengruppen erstellen

Das folgende Verfahren hilft Ihnen, eine Knotengruppe mit einer `p4d.24xlarge` gesicherten Knotengruppe mit EFA-Schnittstellen und GPUDirect RDMA zu erstellen und einen Beispieltest der NVIDIA Collective Communications Library (NCCL) für die NCCL-Leistung mit mehreren Knoten unter Verwendung von EFAs durchzuführen Das Beispiel kann eine Vorlage für verteiltes Deep-Learning-Training auf Amazon EKS mit EFAs verwendet werden.

1. Ermitteln Sie, welche Amazon EC2 EC2-Instance-Typen, die EFA unterstützen AWS-Region , in dem Sie Knoten bereitstellen möchten, verfügbar sind. *region-code*Ersetzen Sie durch AWS-Region die, in der Sie Ihre Knotengruppe bereitstellen möchten.

   ```
   aws ec2 describe-instance-types --region region-code --filters Name=network-
   info.efa-supported,Values=true \
       --query "InstanceTypes[*].[InstanceType]" --output text
   ```

   Wenn Sie Knoten bereitstellen, muss der Instanztyp, den Sie bereitstellen möchten, in dem sich Ihr Cluster befindet AWS-Region , verfügbar sein.

2. Bestimmen Sie, in welchen Availability Zones der Instance-Typ bereitgestellt werden soll. In diesem Tutorial wird der `p4d.24xlarge` Instanztyp verwendet und muss in der Ausgabe für den Instanztyp zurückgegeben werden AWS-Region , den Sie im vorherigen Schritt angegeben haben. Wenn Sie Knoten in einem Produktionscluster bereitstellen, *p4d.24xlarge* ersetzen Sie ihn durch einen beliebigen Instanztyp, der im vorherigen Schritt zurückgegeben wurde.

   ```
   aws ec2 describe-instance-type-offerings --region region-code --location-type
    availability-zone --filters Name=instance-type,Values=p4d.24xlarge \
       --query 'InstanceTypeOfferings[*].Location' --output text
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   us-west-2a    us-west-2c    us-west-2b
   ```

   Beachten Sie, dass Availability Zones zur Verwendung in späteren Schritten zurückgegeben wurden Wenn Sie Knoten in einem Cluster bereitstellen, muss Ihre VPC über Subnetze mit verfügbaren IP-Adressen in einer der in der Ausgabe zurückgegebenen Availability Zones verfügen.

3. Erstellen Sie eine Knotengruppe mit einem `eksctl` oder dem AWS CLI und AWS CloudFormation.

   eksctl

   Voraussetzung

   Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

1. Kopieren Sie den folgenden Inhalt in eine Datei namens *efa-cluster.yaml*. Ersetzen Sie das *example values* durch Ihr eigenes. Sie können *p4d.24xlarge* mit einer anderen Instance ersetzen. Wenn Sie dies tun, stellen Sie sicher, dass die Werte für `availabilityZones` Availability Zones sind, die in Schritt 1 für den Instance-Typ zurückgegeben wurden.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-efa-cluster
  region: region-code
  version: "1.XX"

iam:
  withOIDC: true

availabilityZones: ["us-west-2a", "us-west-2c"]

managedNodeGroups:
  - name: my-efa-ng
    instanceType: p4d.24xlarge
    minSize: 1
    desiredCapacity: 2
    maxSize: 3
    availabilityZones: ["us-west-2a"]
    volumeSize: 300
    privateNetworking: true
    efaEnabled: true
```

2. Erstellen einer verwalteten Knotengruppe in einem bestehenden Cluster

```
eksctl create nodegroup -f efa-cluster.yaml
```

   Wenn Sie über keinen bestehenden Cluster verfügen, können Sie den folgenden Befehl ausführen, um einen Cluster und die Knotengruppe zu erstellen.

```
eksctl create cluster -f efa-cluster.yaml
```

> **Note**
>
> Da der in diesem Beispiel verwendete Instance-Typ GPUs hat, installiert `eksctl` automatisch das NVIDIA-Kubernetes-Geräte-Plugin auf jeder Instance für Sie.

AWS CLI and AWS CloudFormation

Es gibt mehrere Anforderungen für EFA-Netzwerke, einschließlich der Erstellung einer EFA-spezifischen Sicherheitsgruppe, der Erstellung einer Amazon-EC2-Platzierungsgruppe, und der Erstellung einer Startvorlage, die eine oder mehrere EFA-Schnittstellen angibt und die EFA-Treiberinstallation als Teil der Amazon-EC2-Benutzerdaten enthält. Weitere Informationen zu den EFA-Anforderungen finden Sie unter Erste Schritte mit EFA und MPI im Amazon EC2 EC2-Benutzerhandbuch. Die folgenden Schritte erstellen all dies für Sie. Ersetzen Sie alle *Beispielwerte* durch Ihre eigenen.

1. Legen Sie einige Variablen fest, die in späteren Schritten verwendet werden. Ersetzen Sie alle *example values* mit Ihren eigenen. Ersetzen Sie *my-cluster* durch den Namen Ihres bestehenden Clusters. Der Wert für `node_group_resources_name` wird später verwendet, um einen Stack zu erstellen. AWS CloudFormation Der Wert für `node_group_name` wird später verwendet, um die Knotengruppe in Ihrem Cluster zu erstellen.

   ```
   cluster_name="my-cluster"
   cluster_region="region-code"
   node_group_resources_name="my-efa-nodegroup-resources"
   node_group_name="my-efa-nodegroup"
   ```

2. Identifizieren Sie ein privates Subnetz in Ihrer VPC, das sich in derselben Availability Zone befindet, in der der Instance-Typ bereitgestellt werden soll.

   a. Rufen Sie die Version des Clusters ab und speichern Sie sie in einer Variablen für die Verwendung in einem späteren Schritt.

   ```
   cluster_version=$(aws eks describe-cluster \
       --name $cluster_name \
       --query "cluster.version" \
       --output text)
   ```

b. Rufen Sie die VPC ID ab, in der sich Ihr Cluster befindet, und speichern Sie sie in einer Variablen für die Verwendung in einem späteren Schritt.

```
vpc_id=$(aws eks describe-cluster \
    --name $cluster_name \
    --query "cluster.resourcesVpcConfig.vpcId" \
    --output text)
```

c. Rufen Sie die ID der Sicherheitsgruppe der Kontrollebene für Ihren Cluster ab und speichern Sie sie in einer Variablen zur Verwendung in einem späteren Schritt

```
control_plane_security_group=$(aws eks describe-cluster \
    --name $cluster_name \
    --query "cluster.resourcesVpcConfig.clusterSecurityGroupId" \
    --output text)
```

d. Rufen Sie die Liste der Subnetz-IDs in Ihrer VPC ab, die sich in einer Availability Zone befinden, die in Schritt 1 zurückgegeben wird.

```
aws ec2 describe-subnets \
    --filters "Name=vpc-id,Values=$vpc_id" "Name=availability-
zone,Values=us-west-2a" \
    --query 'Subnets[*].SubnetId' \
    --output text
```

Wenn keine Ausgabe zurückgegeben wird, versuchen Sie es mit einer anderen Availability Zone, die in Schritt 1 zurückgegeben wird. Wenn sich keines Ihrer Subnetze in einer Availability Zone befindet, die in Schritt 1 zurückgegeben wird, müssen Sie ein Subnetz in einer Availability Zone erstellen, die in Schritt 1 zurückgegeben wird. Wenn Sie in Ihrer VPC keinen Platz zum Erstellen eines anderen Subnetzes haben, können Sie der VPC einen CIDR-Block hinzufügen und Subnetze im neuen CIDR-Block erstellen oder einen neuen Cluster in einer neuen VPC erstellen.

e. Um zu ermitteln, ob das Subnetz ein privates Subnetz ist, überprüfen Sie die Routing-Tabelle für das Subnetz.

```
aws ec2 describe-route-tables \
    --filter Name=association.subnet-id,Values=subnet-0d403852a65210a29 \
    --query "RouteTables[].Routes[].GatewayId" \
    --output text
```

Eine Beispielausgabe sieht wie folgt aus.

```
local
```

Wenn die Ausgabe `local igw-02adc64c1b72722e2` ist, ist das Subnetz ein öffentliches Subnetz. Sie müssen ein privates Subnetz in einer Availability Zone auswählen, die in Schritt 1 zurückgegeben wird. Wenn Sie ein privates Subnetz identifiziert haben, notieren Sie die ID zur Verwendung in einem späteren Schritt.

f. Legen Sie eine Variable mit der ID des privaten Subnetzes aus dem vorherigen Schritt für die Verwendung in späteren Schritten fest.

```
subnet_id=your-subnet-id
```

3. Laden Sie die AWS CloudFormation Vorlage herunter.

```
curl -O https://raw.githubusercontent.com/aws-samples/aws-efa-eks/main/
cloudformation/efa-p4d-managed-nodegroup.yaml
```

4. Kopieren Sie den folgenden Text auf Ihren Computer. Ersetzen Sie *p4d.24xlarge* mit einem Instance-Typ aus Schritt 1. Ersetzen Sie *subnet-0d403852a65210a29* durch die ID des privaten Subnetzes, das Sie in Schritt 2.b.v identifiziert haben. Ersetzen Sie *path-to-downloaded-cfn-template* mit dem Pfad zu `efa-p4d-managed-nodegroup.yaml`, den Sie im vorherigen Schritt heruntergeladen haben. Ersetzen Sie *your-public-key-name* mit dem Namen Ihres öffentlichen Schlüssels. Nachdem Sie das Ersetzen vorgenommen haben, führen Sie den geänderten Befehl aus.

```
aws cloudformation create-stack \
    --stack-name ${node_group_resources_name} \
    --capabilities CAPABILITY_IAM \
    --template-body file://path-to-downloaded-cfn-template \
    --parameters \
       ParameterKey=ClusterName,ParameterValue=${cluster_name} \
       ParameterKey=ClusterControlPlaneSecurityGroup,ParameterValue=
${control_plane_security_group} \
       ParameterKey=VpcId,ParameterValue=${vpc_id} \
       ParameterKey=SubnetId,ParameterValue=${subnet_id} \
       ParameterKey=NodeGroupName,ParameterValue=${node_group_name} \
       ParameterKey=NodeImageIdSSMParam,ParameterValue=/aws/service/eks/
optimized-ami/${cluster_version}/amazon-linux-2-gpu/recommended/image_id \
```

```
        ParameterKey=KeyName,ParameterValue=your-public-key-name \
        ParameterKey=NodeInstanceType,ParameterValue=p4d.24xlarge
```

5. Bestimmen Sie, wann der Stack, den Sie im vorherigen Schritt bereitgestellt haben, bereitgestellt wird.

```
aws cloudformation wait stack-create-complete --stack-name
 $node_group_resources_name
```

   Es gibt keine Ausgabe des vorherigen Befehls, aber Ihre Shell-Eingabeaufforderung wird erst zurückgegeben, wenn der Stack erstellt wurde.

6. Erstellen Sie Ihre Knotengruppe mit den Ressourcen, die von dem AWS CloudFormation -Stack im vorherigen Schritt erstellt wurden.

   a. Rufen Sie Informationen aus dem bereitgestellten AWS CloudFormation Stack ab und speichern Sie sie in Variablen.

```
node_instance_role=$(aws cloudformation describe-stacks \
    --stack-name $node_group_resources_name \
    --query='Stacks[].Outputs[?OutputKey==`NodeInstanceRole`].OutputValue'
 \
    --output text)
launch_template=$(aws cloudformation describe-stacks \
    --stack-name $node_group_resources_name \
    --query='Stacks[].Outputs[?OutputKey==`LaunchTemplateID`].OutputValue'
 \
    --output text)
```

   b. Erstellen Sie eine verwaltete Knotengruppe, die die Startvorlage und die Knoten-IAM-Rolle verwendet, die im vorherigen Schritt erstellt wurden.

```
aws eks create-nodegroup \
    --cluster-name $cluster_name \
    --nodegroup-name $node_group_name \
    --node-role $node_instance_role \
    --subnets $subnet_id \
    --launch-template id=$launch_template,version=1
```

   c. Bestätigen Sie, dass die Knoten erstellt wurden.

```
aws eks describe-nodegroup \
   --cluster-name ${cluster_name} \
```

```
    --nodegroup-name ${node_group_name} | jq -r .nodegroup.status
```

Fahren Sie nicht fort, bis der vom vorherigen Befehl zurückgegebene Status `ACTIVE` ist. Es kann einige Minuten dauern, bis die Knoten bereit sind.

7. Wenn Sie einen GPU-Instance-Typ ausgewählt haben, müssen Sie das NVIDIA-Geräte-Plugin für Kubernetes bereitstellen. Ersetzen Sie *vX.X.X* mit Ihrer gewünschten Nvidia/k8s-Geräte-Plugin-Version, bevor Sie den folgenden Befehl ausführen.

```
kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
```

4. Stellen Sie das EFA Kubernetes Geräte-Plugin bereit.

Das EFA Kubernetes Geräte-Plugin erkennt und bewirbt EFA-Schnittstellen als zuweisbare Ressourcen für Kubernetes. Eine Anwendung kann den erweiterten Ressourcentyp `vpc.amazonaws.com/efa` in einer Pod-Anforderungsspezifikation genau wie CPU und Speicher. Weitere Informationen finden Sie unter Verbrauch erweiterter Ressourcen in der Kubernetes-Dokumentation. Nachdem es angefordert wurde, weist das Plugin automatisch eine EFA-Schnittstelle zu und installiert sie auf dem Pod. Die Verwendung des Geräte-Plugins vereinfacht die EFA-Setup und erfordert keinen Pod, um im privilegierten Modus ausgeführt zu werden.

```
helm repo add eks https://aws.github.io/eks-chart
helm install aws-efa-k8s-device-plugin --namespace kube-system eks/aws-efa-k8s-device-plugin
```

(Optional) Bereitstellen einer EFA-kompatiblen Musteranwendung

Bereitstellen des Kubeflow-MPI-Operators

Für die NCCL-Tests können Sie den Kubeflow MPI Operator anwenden. Der MPI Operator macht es einfach, auf Kubernetes verteiltes Training im Allreduce-Stil durchzuführen. Weitere Informationen finden Sie unter MPI Operator auf GitHub.

```
kubectl apply -f https://raw.githubusercontent.com/kubeflow/mpi-operator/master/deploy/v2beta1/mpi-operator.yaml
```

Führen Sie den NCCL-Leistungstest mit mehreren Knoten aus, um GPUDirectRDMA/EFA zu überprüfen

Um die NCCL-Leistung mit GPUDirectRDMA über EFA zu überprüfen, führen Sie den standardmäßigen NCCL-Leistungstest aus. Weitere Informationen finden Sie im offiziellen NCCL-Tests-Repo auf GitHub. Sie können das Beispiel-Dockerfile verwenden, das in diesem Test enthalten ist, da es bereits für NVIDIA CUDA `11.2` und die neueste EFA-Version erstellt wurde.

Alternativ können Sie ein AWS Docker Bild herunterladen, das aus einem Amazon ECR-Repo verfügbar ist.

> **Important**
>
> Eine wichtige Überlegung, die für die Übernahme von EFA mit Kubernetes erforderlich ist, ist die Konfiguration und Verwaltung von Huge Pages als Ressource im Cluster. Weitere Informationen finden Sie unter Verwalten großer Seiten in der Kubernetes-Dokumentation. Amazon-EC2-Instances mit dem installierten EFA-Treiber weisen 5128 2M Huge Pages vor, die Sie als Ressourcen anfordern können, die Sie in Ihren Auftragsspezifikationen verwenden können.

Führen Sie die folgenden Schritte aus, um einen NCCL-Leistungstest mit zwei Knoten auszuführen. Im Beispiel NCCL-Testauftrag fordert jeder Worker acht GPUs, 5210Mi HugePages-2MI, vier EFAs und 8000Mi Speicher an, was effektiv bedeutet, dass jeder Worker alle Ressourcen einer `p4d.24xlarge`-Instance konsumiert.

1. Erstellen Sie den NCCL-Tests-Auftrag.

   ```
   kubectl apply -f https://raw.githubusercontent.com/aws-samples/aws-efa-eks/main/examples/simple/nccl-efa-tests.yaml
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   nccl-tests-efa mpijob.kubeflow.org/ wurde erstellt
2. Zeigen Sie Ihr laufendes Pods an.

   ```
   kubectl get pods
   ```

   Eine Beispielausgabe sieht wie folgt aus.

```
NAME                             READY   STATUS     RESTARTS   AGE
nccl-tests-efa-launcher-nbql9  0/1     Init:0/1   0          2m49s
nccl-tests-efa-worker-0          1/1     Running    0          2m49s
nccl-tests-efa-worker-1          1/1     Running    0          2m49s
```

Der MPI-Operator erstellt einen Launcher-Pod und 2 Worker-Pods (einen auf jedem Knoten).

3. Zeigen Sie das Protokoll für `efa-launcher` Pod. Ersetzen Sie *`wzr8j`* mit dem Wert aus Ihrer Ausgabe.

```
kubectl logs -f nccl-tests-efa-launcher-nbql9
```

Weitere Beispiele finden Sie im Amazon EKS EFA-Proben-Repository auf GitHub.

Machine Learning-Inferenz mit AWS Inferentia

In diesem Thema wird beschrieben, wie Sie einen Amazon-EKS-Cluster mit Knoten erstellen, die Amazon EC2 Inf1 ausführen, und wie Sie (optional) eine Beispielanwendung bereitstellen. Amazon EC2 Inf1-Instances werden mit AWS Inferentia-Chips betrieben, die speziell für hohe Leistung und kostengünstigste Inferenz in der Cloud entwickelt wurden. AWS Modelle für maschinelles Lernen werden mithilfe von AWS Neuron, einem speziellen Software Development Kit (SDK), das aus Compiler-, Runtime- und Profiling-Tools besteht, in Containern bereitgestellt, die die Inferenzleistung von Inferentia-Chips für maschinelles Lernen optimieren. AWS Neuron unterstützt beliebte Frameworks für maschinelles Lernen wie TensorFlow PyTorch, und MXNet.

> **Note**
>
> Die logischen IDs von Neuron-Geräten müssen konsekutiv sein. Wenn ein Pod, der mehrere Neuron-Geräte anfordert, auf den Instance-Typen `inf1.6xlarge` oder `inf1.24xlarge` geplant wird (die mehr als ein Neuron-Gerät besitzen), wird dieser Pod nicht gestartet, wenn der Kubernetes Scheduler keine konsekutiven Geräte-IDs auswählt. Weitere Informationen finden Sie unter Logische Geräte-IDs müssen konsekutiv sein auf GitHub.

Voraussetzungen

- `eksctl` ist auf Ihrem Computer installiert. Eine Installationsanleitung finden Sie bei Bedarf in der Dokumentation zu `eksctl` unter Installation.
- Installieren Sie `kubectl` auf Ihrem Computer. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- Installieren Sie `python3` auf Ihrem Computer (optional). Wenn Sie es noch nicht installiert haben, finden Sie unter Python-Downloads entsprechende Installationsanweisungen.

Erstellen eines Clusters

Erstellen Sie einen Cluster mit Amazon-EC2-Instance-Knoten, auf denen Inf1-Instances ausgeführt werden.

1. Erstellen Sie einen Cluster mit Knoten, auf denen Inf1 Amazon-EC2-Instances ausgeführt werden. Sie können *`inf1.2xlarge`* mit jedem Inf1-Instance-Typ ersetzen. Das `eksctl`-Dienstprogramm erkennt, dass Sie eine Knotengruppe mit einem `Inf1`-Instance-Typ starten, und startet Ihre Knoten mit einem der für Amazon EKS optimierten beschleunigten Amazon-Linux-AMIs.

   > **Note**
   >
   > Sie können IAM-Rollen nicht für Dienstkonten mit Serving verwenden. TensorFlow

   ```
   eksctl create cluster \
       --name inferentia \
       --region region-code \
       --nodegroup-name ng-inf1 \
       --node-type inf1.2xlarge \
       --nodes 2 \
       --nodes-min 1 \
       --nodes-max 4 \
       --ssh-access \
       --ssh-public-key your-key \
       --with-oidc
   ```

> **Note**
>
> Notieren Sie den Wert der folgenden Ausgabezeile. Dieser wird in einem späteren (optionalen) Schritt verwendet.
>
> ```
> [9]  adding identity "arn:aws:iam::111122223333:role/
> eksctl-inferentia-nodegroup-ng-in-NodeInstanceRole-FI7HIYS3BS09" to auth
>  ConfigMap
> ```

Beim Starten einer Knotengruppe mit `Inf1` Instanzen wird das AWS Kubernetes Neuron-Geräte-Plugin `eksctl` automatisch installiert. Dieses Plugin zeigt dem Kubernetes Scheduler Neuron-Geräte als Systemressourcen an, die von einem Container angefordert werden können. Zusätzlich zu den Standardrichtlinien für Amazon-EKS-Knoten wird die IAM-Richtlinie für den schreibgeschützten Zugriff von Amazon S3 hinzugefügt. So kann die in einem späteren Schritt behandelte Beispielanwendung ein trainiertes Modell aus Amazon S3 laden.

2. Überprüfen Sie, ob alle Pods korrekt gestartet wurden.

```
kubectl get pods -n kube-system
```

Gekürzte Ausgabe:

```
NAME                                  READY   STATUS    RESTARTS   AGE
[...]
neuron-device-plugin-daemonset-6djhp  1/1     Running   0          5m
neuron-device-plugin-daemonset-hwjsj  1/1     Running   0          5m
```

(Optional) Stellen Sie ein TensorFlow Serving-Anwendungs-Image bereit

Ein trainiertes Modell muss für ein Inferentia-Ziel zusammengetragen werden, bevor es auf Inferentia-Instances bereitgestellt werden kann. Um fortzufahren, benötigen Sie ein für Neuron optimiertes TensorFlow Modell, das in Amazon S3 gespeichert ist. Falls Sie noch keines haben SavedModel, folgen Sie bitte dem Tutorial zum Erstellen eines Neuron-kompatiblen ResNet 50-Modells und laden Sie das Ergebnis SavedModel auf S3 hoch. ResNet-50 ist ein beliebtes Modell für maschinelles Lernen, das für Bilderkennungsaufgaben verwendet wird. Weitere Informationen zur Kompilierung

von Neuron-Modellen finden Sie unter The AWS Inferentia Chip With DLAMI im Developer Guide. AWS Deep Learning AMI

Das Beispiel-Deployment-Manifest verwaltet einen vorgefertigten Inferenz-Serving-Container, der von AWS Deep Learning Containers TensorFlow bereitgestellt wird. Im Container befinden sich die AWS Neuron Runtime und die TensorFlow Serving-Anwendung. Eine vollständige Liste der vorgefertigten Deep Learning Containers, die für Neuron optimiert sind, wird auf GitHub unter Verfügbare Images aus. Beim Start ruft der DLC Ihr Modell von Amazon S3 ab, startet Neuron TensorFlow Serving mit dem gespeicherten Modell und wartet auf Vorhersageanfragen.

Die Anzahl der Neuron-Geräte, die Ihrer Serving-Anwendung zugewiesen sind, kann angepasst werden, indem Sie die `aws.amazon.com/neuron`-Ressource in der Bereitstellung yaml verändern. Bitte beachten Sie, dass die Kommunikation zwischen TensorFlow Serving und der Neuron-Laufzeit über GRPC erfolgt, weshalb die Fähigkeit an den `IPC_LOCK` Container übergeben werden muss.

1. Fügen Sie die IAM-Richtlinie-`AmazonS3ReadOnlyAccess` der Knoten-Instance-Rolle hinzu, die in Schritt 1 unter Erstellen eines Clusters erstellt wurde. Dies ist notwendig, damit die Beispielanwendung ein trainiertes Modell aus Amazon S3 laden kann.

   ```
   aws iam attach-role-policy \
       --policy-arn arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
       --role-name eksctl-inferentia-nodegroup-ng-in-NodeInstanceRole-FI7HIYS3BS09
   ```

2. Erstellen Sie eine Datei mit dem Namen `rn50_deployment.yaml` und dem folgenden Inhalt. Aktualisieren Sie den Regions-Code und den Modellpfad entsprechend den gewünschten Einstellungen. Der Modellname dient zu Identifikationszwecken, wenn ein Client eine Anfrage an den TensorFlow Server stellt. In diesem Beispiel wird ein Modellname verwendet, der einem Beispiel-Clientskript von ResNet 50 entspricht, das in einem späteren Schritt zum Senden von Vorhersageanforderungen verwendet wird.

   ```
   aws ecr list-images --repository-name neuron-rtd --registry-id 790709498068 --
   region us-west-2
   ```

   ```
   kind: Deployment
   apiVersion: apps/v1
   metadata:
     name: eks-neuron-test
     labels:
       app: eks-neuron-test
       role: master
   ```

```
spec:
  replicas: 2
  selector:
    matchLabels:
      app: eks-neuron-test
      role: master
  template:
    metadata:
      labels:
        app: eks-neuron-test
        role: master
    spec:
      containers:
        - name: eks-neuron-test
          image: 763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-inference-
neuron:1.15.4-neuron-py37-ubuntu18.04
          command:
            - /usr/local/bin/entrypoint.sh
          args:
            - --port=8500
            - --rest_api_port=9000
            - --model_name=resnet50_neuron
            - --model_base_path=s3://your-bucket-of-models/resnet50_neuron/
          ports:
            - containerPort: 8500
            - containerPort: 9000
          imagePullPolicy: IfNotPresent
          env:
            - name: AWS_REGION
              value: "us-east-1"
            - name: S3_USE_HTTPS
              value: "1"
            - name: S3_VERIFY_SSL
              value: "0"
            - name: S3_ENDPOINT
              value: s3.us-east-1.amazonaws.com
            - name: AWS_LOG_LEVEL
              value: "3"
          resources:
            limits:
              cpu: 4
              memory: 4Gi
              aws.amazon.com/neuron: 1
            requests:
```

```
            cpu: "1"
            memory: 1Gi
          securityContext:
            capabilities:
              add:
                - IPC_LOCK
```

3. Stellen Sie das Modell bereit.

```
kubectl apply -f rn50_deployment.yaml
```

4. Erstellen Sie eine Datei mit dem Namen `rn50_service.yaml` und dem folgenden Inhalt. Die HTTP- und gRPC-Ports werden für die Annahme von Prognoseanforderungen geöffnet.

```
kind: Service
apiVersion: v1
metadata:
  name: eks-neuron-test
  labels:
    app: eks-neuron-test
spec:
  type: ClusterIP
  ports:
    - name: http-tf-serving
      port: 8500
      targetPort: 8500
    - name: grpc-tf-serving
      port: 9000
      targetPort: 9000
  selector:
    app: eks-neuron-test
    role: master
```

5. Erstellen Sie einen Kubernetes Dienst für Ihre TensorFlow Model Serving-Anwendung.

```
kubectl apply -f rn50_service.yaml
```

(Optional) Treffen Sie Prognosen für Ihren TensorFlow Serving-Service

1. Um dies lokal zu testen, leiten Sie den gRPC-Port an den Service `eks-neuron-test` weiter.

```
kubectl port-forward service/eks-neuron-test 8500:8500 &
```

2. Erstellen Sie das Python-Skript namens `tensorflow-model-server-infer.py` mit folgendem Inhalt. Dieses Skript führt die Inferenz über gRPC (Service-Framework) aus.

```
import numpy as np
   import grpc
   import tensorflow as tf
   from tensorflow.keras.preprocessing import image
   from tensorflow.keras.applications.resnet50 import preprocess_input
   from tensorflow_serving.apis import predict_pb2
   from tensorflow_serving.apis import prediction_service_pb2_grpc
   from tensorflow.keras.applications.resnet50 import decode_predictions

   if __name__ == '__main__':
       channel = grpc.insecure_channel('localhost:8500')
       stub = prediction_service_pb2_grpc.PredictionServiceStub(channel)
       img_file = tf.keras.utils.get_file(
           "./kitten_small.jpg",
           "https://raw.githubusercontent.com/awslabs/mxnet-model-server/master/
docs/images/kitten_small.jpg")
       img = image.load_img(img_file, target_size=(224, 224))
       img_array = preprocess_input(image.img_to_array(img)[None, ...])
       request = predict_pb2.PredictRequest()
       request.model_spec.name = 'resnet50_inf1'
       request.inputs['input'].CopyFrom(
           tf.make_tensor_proto(img_array, shape=img_array.shape))
       result = stub.Predict(request)
       prediction = tf.make_ndarray(result.outputs['output'])
       print(decode_predictions(prediction))
```

3. Führen Sie das Skript aus, um Prognosen an den Service zu senden.

```
python3 tensorflow-model-server-infer.py
```

Eine Beispielausgabe sieht wie folgt aus.

```
[[(u'n02123045', u'tabby', 0.68817204), (u'n02127052', u'lynx', 0.12701613),
 (u'n02123159', u'tiger_cat', 0.08736559), (u'n02124075', u'Egyptian_cat',
 0.063844085), (u'n02128757', u'snow_leopard', 0.009240591)]]
```

Clusterverwaltung

In diesem Kapitel finden Sie die folgenden Themen, die Ihnen beim Verwalten Ihres Clusters helfen. Sie können auch Informationen über Ihre Kubernetes-Ressourcen mit der AWS Management Console anzeigen.

- Das Kubernetes-Dashboard ist eine allgemeine, webbasierte Benutzeroberfläche für Kubernetes-Cluster. Es ermöglicht Benutzern, im Cluster ausgeführte Anwendungen zu verwalten und Fehler zu beheben sowie den Cluster selbst zu verwalten. Weitere Informationen finden Sie unter Kubernetes-Dashboard im GitHub-Repository.
- Installieren von Kubernetes Metrics Server – Der Kubernetes Metrics Server ist ein Aggregator für Ressourcenverbrauchsdaten in Ihrem Cluster. Er wird standardmäßig nicht im Cluster bereitgestellt, wird jedoch von Kubernetes-Add-Ons wie dem Kubernetes-Dashboard und Horizontal Pod Autoscaler verwendet. In diesem Thema erfahren Sie, wie der Metrics Server installiert wird.
- Verwendung von Helm mit Amazon EKS – Der Helm-Paketmanager für Kubernetes unterstützt Sie bei der Installation und Verwaltung von Anwendungen in Ihrem Kubernetes-Cluster. Dieses Thema hilft Ihnen bei der Installation und Ausführung der Helm-Binärdateien, sodass Sie Charts mit der Helm-CLI auf Ihrem lokalen Computer installieren und verwalten können.
- Kennzeichnen Ihrer Amazon EKS-Ressourcen - Um Sie bei der Verwaltung Ihrer Amazon EKS-Ressourcen zu unterstützen, können Sie jeder Ressource eigene Metadaten in Form von Tags zuweisen. In diesem Thema werden Tags (Markierungen) und deren Erstellung beschrieben.
- Amazon-EKS-Service-Quotas - Das AWS-Konto verfügt über Standardkontingente (früher als Limits bezeichnet) für jeden AWS-Service. Erfahren Sie mehr über Amazon EKS-Ressourcenkontingente, und wie Sie sie erhöhen.

Kostenüberwachung

Die Kostenüberwachung ist ein wesentlicher Aspekt der Verwaltung Ihrer Kubernetes Cluster auf Amazon EKS. Indem Sie sich einen Überblick über Ihre Cluster-Kosten verschaffen, können Sie die Ressourcennutzung optimieren, Budgets festlegen und datengestützte Entscheidungen über Ihre Bereitstellungen treffen. Amazon EKS bietet zwei Kostenüberwachungslösungen mit jeweils eigenen Vorteilen, mit denen Sie Ihre Kosten effektiv verfolgen und zuordnen können:

AWS Daten zur Aufteilung der Fakturierungskosten für Amazon EKS — Diese native Funktion lässt sich nahtlos in die AWS Billing Console integrieren, sodass Sie Kosten mit derselben vertrauten Oberfläche und denselben Workflows analysieren und zuordnen können, die Sie für andere AWS Services verwenden. Mit der geteilten Kostenzuweisung können Sie Einblicke in Ihre Kubernetes Kosten direkt mit Ihren anderen AWS Ausgaben gewinnen, was es einfacher macht, die Kosten in Ihrer gesamten Umgebung ganzheitlich zu optimieren. AWS Sie können auch bestehende AWS Abrechnungsfunktionen wie Cost Categories und Erkennung von Kostenanomalien nutzen, um Ihre Kostenmanagementfunktionen weiter zu verbessern. Weitere Informationen finden Sie unter Grundlegendes zu Daten zur geteilten Kostenzuweisung im AWS Billing User Guide.

Kubecost— Amazon EKS unterstützt Kubecost, ein Tool zur Kostenüberwachung von Kubernetes. Kubecost bietet einen funktionsreichen, Kubernetes-nativen Ansatz zur Kostenüberwachung, der detaillierte Kostenaufschlüsselungen nach Kubernetes-Ressourcen, Empfehlungen zur Kostenoptimierung sowie Dashboards und Berichte bietet. out-of-the-box Kubecost ruft durch die Integration in den AWS Kosten- und Nutzungsbericht auch genaue Preisdaten ab, sodass Sie einen genauen Überblick über Ihre Amazon EKS-Kosten erhalten. Erfahren Sie, wie Sie es installieren. Kubecost

AWS Abrechnung — Aufteilung der Kosten

Kostenüberwachung mithilfe von Daten zur AWS geteilten Kostenzuweisung für Amazon EKS

Sie können Daten zur AWS geteilten Kostenzuweisung für Amazon EKS verwenden, um eine detaillierte Kostentransparenz für Ihre Amazon EKS-Cluster zu erhalten. Auf diese Weise können Sie die Kosten und die Nutzung Ihrer Anwendungen analysieren, optimieren und rückbuchen. Kubernetes Sie weisen die Anwendungskosten einzelnen Geschäftseinheiten und Teams auf der Grundlage der von Ihrer Kubernetes Anwendung verbrauchten CPU- und Speicherressourcen von Amazon EC2 zu. Geteilte Kostenzuweisungsdaten für Amazon EKS bieten Einblick in die Kosten pro Pod und ermöglichen es Ihnen, die Kostendaten pro Pod mithilfe von Namespace, Cluster und anderen Kubernetes Primitiven zu aggregieren. Im Folgenden finden Sie Beispiele für Kubernetes Primitive, mit denen Sie Amazon EKS-Kostenzuordnungsdaten analysieren können.

- Clustername
- Bereitstellung
- Namespace
- Knoten
- Name der Workload

- Workload-Typ

Weitere Informationen zur Verwendung von Daten zur geteilten Kostenzuweisung finden Sie unter Grundlegendes zu geteilten Kostenzuordnungsdaten im AWS Billing User Guide.

Richten Sie Kosten- und Nutzungsberichte ein

Sie können Split Cost Allocation Data für EKS in der Cost Management Console oder in den AWS SDKs aktivieren. AWS Command Line Interface

Verwenden Sie Folgendes für geteilte Kostenzuordnungsdaten:

1. Entscheiden Sie sich für das Teilen von Kostenverrechnungsdaten. Weitere Informationen finden Sie im AWS Cost and Usage Report Benutzerhandbuch unter Aktivieren von geteilten Kostenzuordnungsdaten.
2. Nehmen Sie die Daten in einen neuen oder vorhandenen Bericht auf.
3. Sehen Sie sich den Bericht an. Sie können die Konsole Fakturierung und Kostenmanagement verwenden oder die Berichtsdateien in Amazon Simple Storage Service anzeigen.

Kubecost

Amazon EKS unterstützt Kubecost, mit dem Sie Ihre Kosten überwachen können, aufgeschlüsselt nach Kubernetes-Ressourcen einschließlich Pods, Knoten, Namespaces und Labels. Als Kubernetes-Plattformadministrator und Finanzfachkraft können Sie Kubecost verwenden, um eine Aufschlüsselung der Amazon-EKS-Gebühren zu visualisieren, Kosten zuzuweisen und Organisationseinheiten wie Anwendungsteams Gebühren zu berechnen. Sie können Ihren internen Teams und Geschäftseinheiten transparente und genaue Kostendaten auf der Grundlage ihrer tatsächlichen AWS Rechnung zur Verfügung stellen. Darüber hinaus können Sie angepasste Empfehlungen für die Kostenoptimierung erhalten, die auf der Infrastrukturumgebung und den Nutzungsmustern innerhalb ihrer Cluster basieren. Weitere Informationen über Kubecost finden Sie in der Kubecost-Dokumentation.

Amazon EKS bietet ein AWS optimiertes Paket Kubecost für die Transparenz der Cluster-Kosten. Sie können Ihre bestehenden AWS Supportverträge verwenden, um Support zu erhalten.

Voraussetzungen

- Ein vorhandener Amazon-EKS-Cluster. Informationen zum Bereitstellen finden Sie unter Erste Schritte mit Amazon EKS. Der Cluster muss über Amazon-EC2-Knoten verfügen, da Sie Kubecost nicht auf Fargate-Knoten ausführen können.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Gerät oder in der AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- Helm-Version 3.9.0 oder höher ist auf Ihrem Gerät oder in der AWS CloudShell konfiguriert. Informationen zum Installieren oder Aktualisieren von Helm finden Sie unter the section called “Verwenden von Helm”.
- Wenn Ihr Cluster in der Version `1.23` oder höher vorliegt, müssen Sie den the section called “Amazon-EBS-CSI-Treiber” auf Ihrem Cluster installiert haben.

Um Kubecost zu installieren

1. Ermitteln Sie die Version von, die installiert werden Kubecost soll. Die verfügbaren Versionen finden Sie unter kubecost/cost-analyzer in der Amazon ECR Public Gallery. Weitere Informationen zur Kompatibilität von Kubecost Versionen und Amazon EKS finden Sie in den Environment Requirements in der Kubecost-Dokumentation.
2. Installieren Sie Kubecost mit dem folgenden Befehl: *Ersetzen Sie kubecost-version durch den aus ECR abgerufenen Wert, z. B. 1.108.1.*

   ```
   helm upgrade -i kubecost oci://public.ecr.aws/kubecost/cost-analyzer --
   version kubecost-version \
       --namespace kubecost --create-namespace \
       -f https://raw.githubusercontent.com/kubecost/cost-analyzer-helm-chart/develop/
   cost-analyzer/values-eks-cost-monitoring.yaml
   ```

   Kubecost veröffentlicht regelmäßig neue Versionen. Sie können Ihre Version mit `helm upgrade` aktualisieren. Die Installation beinhaltet standardmäßig einen lokalen Prometheus-Server und `kube-state-metrics`. Sie können Ihre Bereitstellung für die Verwendung von Amazon Managed Service für Prometheus anpassen, indem Sie der Dokumentation unter Integration in die Amazon-EKS-Kostenüberwachung folgen. Eine Liste aller anderen Einstellungen, die Sie konfigurieren können, finden Sie in der Beispielkonfigurationsdatei unter. GitHub

3. Stellen Sie sicher, dass die erforderlichen Pods ausgeführt werden.

```
kubectl get pods -n kubecost
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME                                           READY   STATUS    RESTARTS   AGE
kubecost-cost-analyzer-b9788c99f-5vj5b         2/2     Running   0          3h27m
kubecost-kube-state-metrics-99bb8c55b-bn2br    1/1     Running   0          3h27m
kubecost-prometheus-server-7d9967bfc8-9c8p7    2/2     Running   0          3h27m
```

4. Aktivieren Sie auf Ihrem Gerät die Portweiterleitung, um das Kubecost-Dashboard anzuzeigen.

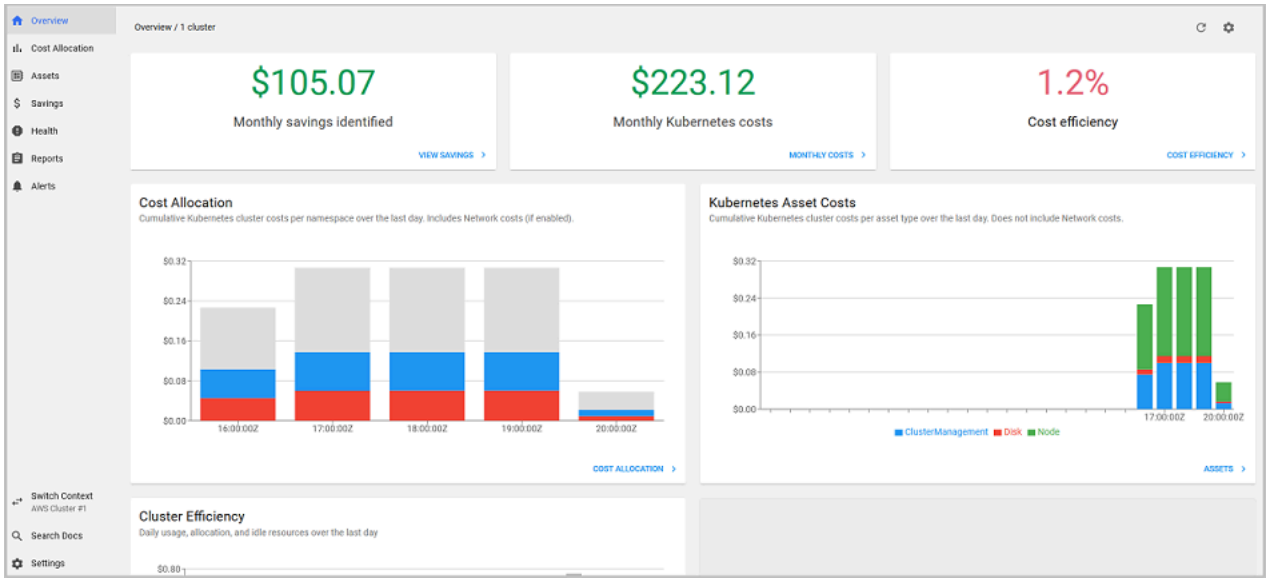
```
kubectl port-forward --namespace kubecost deployment/kubecost-cost-analyzer 9090
```

Alternativ hierzu können Sie auch den AWS Load Balancer Controller verwenden, um Kubecost anzuzeigen, und Amazon Cognito für Authentifizierung, Autorisierung und Benutzerverwaltung verwenden. Weitere Informationen finden Sie unter So verwenden Sie den Application Load Balancer und Amazon Cognito authentifizieren Benutzer für Ihre Kubernetes Web-Apps.

5. Öffnen Sie auf demselben Gerät, auf dem Sie den vorherigen Schritt ausgeführt haben, einen Webbrowser und geben Sie die folgende Adresse ein.

```
http://localhost:9090
```

Sie sehen die Kubecost-Übersichtsseite in Ihrem Browser. Es kann 5–10 Minuten dauern, bis Kubecost Kennzahlen gesammelt hat. Sie können Ihre Amazon-EKS-Ausgaben einschließlich der kumulierten Clusterkosten, zugeordneten Kubernetes-Assetkosten und monatlichen Gesamtausgaben anzeigen.



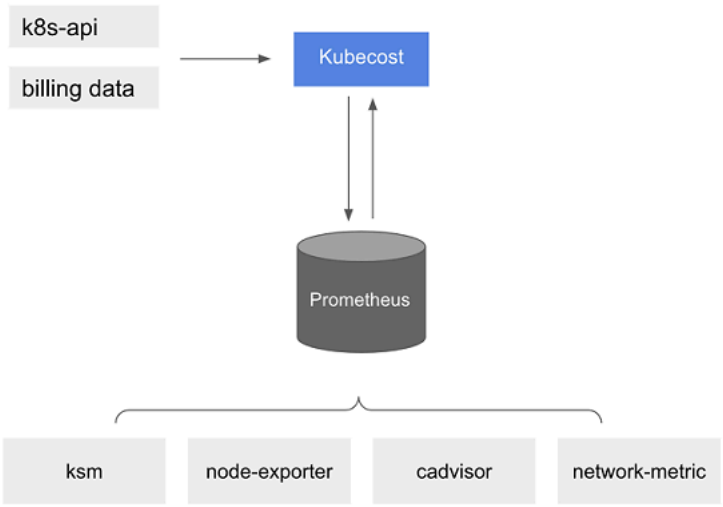
6. Um die Kosten auf Clusterebene zu verfolgen, kennzeichnen Sie Ihre Amazon-EKS-Ressourcen für die Abrechnung mit Tags. Weitere Informationen finden Sie unter Markieren von Ressourcen für die Fakturierung.

Sie können die folgenden Informationen auch anzeigen, indem Sie sie im linken Bereich des Dashboards auswählen:

- Cost allocation (Kostenzuordnung) – Zeigen Sie die monatlichen Amazon-EKS-Kosten und kumulierten Kosten für jeden Ihrer Namespaces und andere Dimensionen der letzten sieben Tage an. Dies ist hilfreich, um zu verstehen, welche Teile Ihrer Anwendung zu den Amazon-EKS-Ausgaben beitragen.
- Assets – Zeigen Sie die Kosten der AWS -Infrastrukturressourcen an, die Ihren Amazon-EKS-Ressourcen zugeordnet sind.

Weitere Features

- Export cost metrics (Kostenkennzahlen exportieren) – Die optimierte Amazon-EKS-Kostenüberwachung wird mit Kubecost und Prometheus bereitgestellt, ein Open-Source-Überwachungssystem und eine Zeitreihendatenbank. Kubecost liest Kennzahlen aus Prometheus und führt dann Kostenverteilungsberechnungen durch und schreibt die Kennzahlen zurück in Prometheus. Das Kubecost-Frontend liest Kennzahlen aus Prometheus und zeigt sie in der Kubecost-Benutzeroberfläche an. Das folgende Diagramm veranschaulicht die Architektur.



Wenn Prometheus vorinstalliert ist, können Sie Abfragen schreiben, um Kubecost-Daten zur weiteren Analyse in Ihr aktuelles Business-Intelligence-System aufzunehmen. Sie können es auch als Datenquelle für Ihr aktuelles Grafana-Dashboard verwenden, mit dem Ihre internen Teams vertraut sind, um die Amazon-EKS-Clusterkosten anzuzeigen. Weitere Informationen zum Schreiben von Prometheus Abfragen finden Sie in der Prometheus`readme`Konfigurationsdatei unter GitHub oder verwenden Sie die Grafana JSON-Beispielmodelle im KubecostGithub-Repository als Referenz.

- AWS Cost and Usage Report Integration — Kubecost Ruft die öffentlichen Preisinformationen AWS-Services und AWS Ressourcen von der AWS Preislisten-API ab, um Berechnungen der Kostenzuweisung für Ihren Amazon EKS-Cluster durchzuführen. Sie können auch eine Integration Kubecost mit AWS Cost and Usage Reportdurchführen, um die Genauigkeit der für Sie spezifischen Preisinformationen zu erhöhen AWS-Konto. Zu diesen Informationen gehören Rabattprogramme für Unternehmen, Nutzung von reservierten Instances, Savings Plans und Spot-Nutzung. Weitere Informationen zur Funktionsweise der AWS Cost and Usage Report Integration finden Sie in der Kubecost Dokumentation unter AWS Cloud Billing Integration.

Remove Kubecost

Sie können Kubecost mit den folgenden Befehlen aus Ihrem Cluster entfernen.

```
helm uninstall kubecost --namespace kubecost
kubectl delete ns kubecost
```

Häufig gestellte Fragen

Lesen Sie die folgenden häufig gestellten Fragen und Antworten zur Verwendung von Kubecost mit Amazon EKS.

Was ist der Unterschied zwischen dem benutzerdefinierten Bundle von Kubecost und der kostenlosen Version von Kubecost (auch bekannt als OpenCost)?

AWS und haben Kubecost zusammengearbeitet, um eine maßgeschneiderte Version von Kubecost anzubieten. Diese Version enthält eine Teilmenge kommerzieller Features ohne zusätzliche Kosten. In der folgenden Tabelle finden Sie Features, die im benutzerdefinierten Paket von Kubecost enthalten sind.

Funktion	Kubecost kostenloses Kontingent	Amazon-EKS-optimiertes benutzerdefiniertes Kubecost-Paket	Kubecost Enterprise
Bereitstellung	Vom Benutzer gehostet	Vom Benutzer gehostet	Vom Benutzer gehostet oder Kubecost-gehostet (SaaS)
Anzahl der unterstützten Cluster	Unbegrenzt	Unbegrenzt	Unbegrenzt
Unterstützte Datenbanken	Lokales Prometheus	Lokales Prometheus oder Amazon Managed Service für Prometheus	Prometheus, Amazon Managed Service für Prometheus, Cortex oder Thanos
Unterstützung für die Aufbewahrung von Datenbanken	15 Tage	Unbegrenzte historische Daten	Unbegrenzte historische Daten
Kubecost-API-Aufbewahrung (ETL)	15 Tage	15 Tage	Unbegrenzte historische Daten
Transparenz der Cluster-Kosten	Einzelne Cluster	Einheitliches Multi-Cluster	Einheitliches Multi-Cluster

Funktion	Kubecost kostenloses Kontingent	Amazon-EKS-optimiertes benutzerdefiniertes Kubecost-Paket	Kubecost Enterprise
Sichtbarkeit in der Hybrid Cloud	-	Amazon EKS und Amazon-EKS-Anywhere-Cluster	Multi-Cloud- und Hybrid-Cloud-Unterstützung
Benachrichtigungen und wiederkehrende Berichte	-	Unterstützung von Effizienzwarnungen, Budgetwarnungen, Ausgabenänderungswarnungen und mehr	Unterstützung von Effizienzwarnungen, Budgetwarnungen, Ausgabenänderungswarnungen und mehr
Gespeicherte Berichte	-	Berichte mit Daten aus 15 Tagen	Berichte mit unbegrenzten historischen Daten
Integration der Cloud-Abrechnung	Für jeden einzelnen Cluster erforderlich	Unterstützung bei kundenspezifischen Preisen für AWS (einschließlich mehrerer Cluster und mehrerer Konten)	Unterstützung bei kundenspezifischen Preisen für AWS (einschließlich mehrerer Cluster und mehrerer Konten)
Empfehlungen für Einsparungen	Erkenntnisse aus einzelnen Clustern	Erkenntnisse aus einzelnen Clustern	Erkenntnisse aus mehreren Clustern
Unternehmensführung: Prüfungen	-	-	Prüfen Sie historische Kostenereignisse
Unterstützung von Single Sign-On (SSO)	-	Unterstützung von Amazon Cognito	Okta, Auth0, PingID, KeyCloak

Funktion	Kubecost kostenloses Kontingent	Amazon-EKS-optimiertes benutzerdefiniertes Kubecost-Paket	Kubecost Enterprise
Rollenbasierte Zugriffskontrolle (RBAC) mit SAML **2.0**	-	-	Okta, Auth0, PingID, Keycloak
Training und Onboarding für Unternehmen	-	-	Full-Service-Training und FinOps-Onboarding

Was ist das Kubecost-Feature zur API-Aufbewahrung (ETL)?

Das Kubecost-ETL-Feature aggregiert und organisiert Kennzahlen, um Kostentransparenz auf verschiedenen Granularitätsebenen zu gewährleisten (z. B. `namespace-level`, `pod-level` und `deployment-level`). Für das benutzerdefinierte Kubecost-Paket erhalten die Kunden Daten und Einblicke in die Metriken der letzten 15 Tage.

Was ist das Feature für Benachrichtigungen und wiederkehrende Berichte? Welche Warnmeldungen und Berichte sind enthalten?

Kubecost Warnmeldungen ermöglichen es den Teams, aktuelle Informationen über die Kubernetes-Ausgaben in Echtzeit sowie über die Ausgaben in der Cloud zu erhalten. Wiederkehrende Berichte ermöglichen es Teams, individuelle Ansichten der historischen Kubernetes- und Cloud-Ausgaben zu erhalten. Beide sind konfigurierbar mit der Kubecost-UI oder Helm-Werten. Sie unterstützen E-Mail, Slack und Microsoft Teams.

Was beinhalten gespeicherte Berichte?

Gespeicherte Kubecost-Berichte sind vordefinierte Ansichten von Kosten- und Effizienzkennzahlen. Sie beinhalten die Kosten nach Cluster, Namespace, Label und mehr.

Was ist Integration der Cloud-Abrechnung?

Die Integration mit AWS Fakturierungs-APIs Kubecost ermöglicht die Anzeige von out-of-cluster Kosten (wie Amazon S3). Darüber hinaus ermöglicht es Kubecost den Abgleich von Kubecosts In-Cluster-Prognosen mit den tatsächlichen Abrechnungsdaten, um Spotnutzung, Savings Plans und Unternehmensrabatte zu berücksichtigen.

Was beinhalten Sparempfehlungen?

Kubecost bietet Einblicke und Automatisierung, um Benutzern bei der Optimierung ihrer Kubernetes-Infrastruktur und Ausgaben.

Entstehen Kosten für diese Funktion?

Nein. Sie können diese Version von Kubecost ohne Zusatzkosten verwenden. Wenn Sie zusätzliche Kubecost Funktionen wünschen, die nicht in diesem Paket enthalten sind, können Sie eine Unternehmenslizenz von Kubecost über oder Kubecost direkt bei erwerben. AWS Marketplace

Ist Support verfügbar?

Ja. Sie können unter Kontakt eine Support-Anfrage mit dem AWS Support Team eröffnen AWS.

Benötige ich eine Lizenz für die Verwendung von Kubecost-Features, die von der Amazon-EKS-Integration bereitgestellt werden?

Nein.

Kann ich es AWS Cost and Usage Report für genauere Berichte integrierenKubecost?

Ja. Sie können Kubecost konfigurieren, um Daten von AWS Cost and Usage Report aufzunehmen, um eine genaue Kostentransparenz zu erhalten, einschließlich Rabatten, Spot-Preisen, Preisen für reservierte Instances und mehr. Weitere Informationen finden Sie in der Kubecost Dokumentation unter AWS Cloud Billing Integration.

Unterstützt diese Version das Kostenmanagement von selbstverwalteten Kubernetes-Clustern auf Amazon EC2?

Nein. Diese Version ist nur mit Amazon-EKS-Clustern kompatibel.

Kann Kubecost die Kosten für Amazon EKS auf AWS Fargate verfolgen?

Kubecost bietet beste Anstrengungen, um die Sichtbarkeit der Cluster-Kosten für Amazon EKS auf Fargate zu zeigen, jedoch mit geringerer Genauigkeit als mit Amazon EKS auf Amazon EC2. Dies ist hauptsächlich auf den Unterschied zurückzuführen, wie Ihnen Ihre Nutzung in Rechnung gestellt wird. Mit Amazon EKS auf Fargate werden Ihnen die verbrauchten Ressourcen in Rechnung gestellt. Mit Amazon EKS auf Amazon-EC2-Knoten werden Ihnen bereitgestellte Ressourcen in Rechnung gestellt. Kubecost berechnet die Kosten eines Amazon-EC2-Knotens basierend auf der Knotenspezifikation, die CPU, RAM und kurzlebigen Speicher umfasst. Mit Fargate werden die Kosten basierend auf den angeforderten Ressourcen für die Fargate-Pods berechnet.

Wie erhalte ich Updates und neue Versionen von Kubecost?

Sie können Ihre Kubecost-Version mit Standard-Helm-Upgrade-Verfahren aktualisieren. Die neuesten Versionen sind in der Amazon ECR Public Gallery.

Wird der **`kubectl-cost`**-CLI unterstützt? Wie installiere ich sie?

Ja. `Kubectl-cost` ist ein Open-Source-Tool von Kubecost (Apache 2.0-Lizenz), das CLI Zugriff auf Kubernetes-Kennzahlen zur Kostenzuordnungen bietet. Informationen zur Installation `kubectl-cost` finden Sie unter Installation auf GitHub.

Wird die Kubecost-Benutzeroberfläche unterstützt? Wie greife ich darauf zu?

Kubecost bietet ein Web-Dashboard, auf das Sie über `kubectl`-Portweiterleitung, eine Eingangsregel oder einen Load Balancer zugreifen können. Sie können auch den AWS Load Balancer Controller verwenden, um Kubecost anzuzeigen, und Amazon Cognito für Authentifizierung, Autorisierung und Benutzerverwaltung verwenden. Weitere Informationen finden Sie im Blog unter So verwenden Sie Application Load Balancer und Amazon Cognito zur Authentifizierung von Benutzern für Ihre Kubernetes Web-Apps. AWS

Wird Amazon EKS Anywhere unterstützt?

Nein.

Installieren von Kubernetes Metrics Server

Der Kubernetes Metrics Server aggregiert Daten zur Ressourcennutzung in Ihrem Cluster und wird standardmäßig in Amazon-EKS-Clustern nicht bereitgestellt. Weitere Informationen finden Sie unter Kubernetes Metrics Server auf GitHub. Der Metrics Server wird üblicherweise von anderen Kubernetes-Add-Ons, wie dem Horizontal Pod Autoscaler oder dem Kubernetes-Dashboard verwendet. Weitere Informationen finden Sie unter Ressourcen-Kennzahlenpipeline in der Kubernetes-Dokumentation. In diesem Thema wird erläutert, wie Sie den Kubernetes Metrics Server in Ihrem Amazon-EKS-Cluster bereitstellen.

> **Important**
>
> Die Kennzahlen sind für point-in-time Analysen gedacht und stellen keine genaue Quelle für historische Analysen dar. Sie können nicht als Überwachungslösung oder für andere Zwecke verwendet werden, die nichts mit Auto Scaling zu tun haben. Weitere Informationen zu Überwachungstools finden Sie unter Obervability in Amazon EKS.

Den Metrics Server bereitstellen

1. Stellen Sie den Metrics Server mit dem folgenden Befehl bereit:

   ```
   kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
   ```

   Wenn Sie Fargate verwenden, müssen Sie diese Datei ändern. In der Standardkonfiguration verwendet der Metrikserver Port 10250. Dieser Port ist auf Fargate reserviert. Ersetzen Sie Verweise auf Port 10250 in components.yaml durch einen anderen Port, z. B. 10251.
2. Überprüfen Sie mit dem folgenden Befehl, ob die `metrics-server`-Bereitstellung die gewünschte Anzahl Pods umfasst.

   ```
   kubectl get deployment metrics-server -n kube-system
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   NAME             READY   UP-TO-DATE   AVAILABLE   AGE
   metrics-server   1/1     1            1           6m
   ```

Verwendung von Helm mit Amazon EKS

Der Helm-Paketmanager für Kubernetes unterstützt Sie bei der Installation und Verwaltung von Anwendungen in Ihrem Kubernetes-Cluster. Weitere Informationen finden Sie in der Helm-Dokumentation. Dieses Thema hilft Ihnen bei der Installation und Ausführung der Helm-Binärdateien, sodass Sie Charts mit der Helm-CLI auf Ihrem lokalen System installieren und verwalten können.

> **Important**
>
> Bevor Sie Helm-Charts in Ihrem Amazon-EKS-Cluster installieren können, müssen Sie `kubectl` so konfigurieren, dass es für Amazon EKS funktioniert. Wenn Sie dies noch nicht getan haben, lesen Sie Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster, bevor Sie fortfahren. Wenn der folgende Befehl für Ihren Cluster erfolgreich ist, ist die Konfiguration korrekt.
>
> ```
> kubectl get svc
> ```

Installieren Sie die Helm-Binärdateien auf Ihrem lokalen System wie folgt

1. Führen Sie den entsprechenden Befehl für das Client-Betriebssystem aus.

   - Wenn Sie macOS mit Homebrew verwenden, installieren Sie die Binärdateien mit dem folgenden Befehl.

     ```
     brew install helm
     ```

   - Wenn Sie Windows mit Chocolatey verwenden, installieren Sie die Binärdateien mit dem folgenden Befehl.

     ```
     choco install kubernetes-helm
     ```

   - Wenn Sie Linux verwenden, installieren Sie die Binärdateien mit den folgenden Befehlen.

     ```
     curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 >
      get_helm.sh
     chmod 700 get_helm.sh
     ./get_helm.sh
     ```

     > **Note**
     >
     > Wenn Sie eine Nachricht bekommen, dass zunächst `openssl` installiert werden muss, können Sie es mit dem folgenden Befehl installieren.
     >
     > ```
     > sudo yum install openssl
     > ```

2. Um die neue Binärdatei in Ihren PATH aufzunehmen, schließen Sie Ihr aktuelles Terminalfenster und öffnen Sie ein neues.
3. Sehen Sie die Version von Helm, die Sie installiert haben.

   ```
   helm version | cut -d + -f 1
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   v3.9.0
   ```

4. An dieser Stelle können Sie beliebige Helm-Befehle ausführen (z. B. `helm install` *chart-name*), um Helm-Charts in Ihrem Cluster zu installieren, zu ändern, zu löschen oder abzufragen. Wenn Sie neu bei Helm sind und kein bestimmtes Chart installieren möchten, können Sie:

   - Experimentieren, indem Sie ein Beispiel-Chart installieren. Ein Beispiel-Chart installieren in der Helm- Schnellstartanleitung lesen.
   - Erstellen Sie ein Beispieldiagramm und übertragen Sie es an Amazon ECR. Weitere Informationen finden Sie unter Übertragen eines Helm-Diagramms im Amazon-Elastic-Container-Registry-Benutzerhandbuch.
   - Installieren Sie ein Amazon-EKS-Diagramm aus dem eks-charts-GitHubRepo oder von ArtifactHub.

Kennzeichnen Ihrer Amazon EKS-Ressourcen

Sie können Tags verwenden, die Sie bei der Verwaltung Ihrer Amazon-EKS-Ressourcen unterstützen. Dieses Thema bietet einen Überblick über die Tags -Funktion und zeigt, wie Sie Tags erstellen können.

Themen

- Grundlagen zu Tags (Markierungen)
- Markieren Ihrer -Ressourcen
- Tag-Einschränkungen
- Markieren von Ressourcen für die Fakturierung
- Arbeiten mit Tags über die Konsole
- Arbeiten mit Tags mittels CLI, API oder eksctl

> **Note**
>
> Tags sind eine Art von Metadaten, die von Kubernetes-Labels und Anmerkungen getrennt sind. Weitere Informationen zu diesen anderen Metadatentypen finden Sie in folgenden Abschnitten der Kubernetes-Dokumentation:
>
> - Labels und Selektoren
> - Anmerkungen

Grundlagen zu Tags (Markierungen)

Ein Tag ist eine Bezeichnung, die Sie einer AWS Ressource zuweisen. Jedes Tag (Markierung) besteht aus einem Schlüssel und einem optionalen Wert.

Mit Tags können Sie Ihre AWS Ressourcen kategorisieren. Sie können Ressourcen beispielsweise nach Zweck, Inhaber oder Umgebung kategorisieren. Wenn Sie viele Ressourcen desselben Typs haben, können Sie die Tags, die Sie einer bestimmten Ressource zugewiesen haben, verwenden, um diese Ressource schnell zu identifizieren. Sie können beispielsweise eine Reihe von Tags für Ihre Amazon EKS-Cluster definieren, um Ihnen dabei zu helfen, den Eigentümer und die Stack-Ebene jedes einzelnen Clusters nachzuverfolgen. Sie sollten für jeden Ressourcentyp einen konsistenten Satz von Tag-Schlüsseln entwickeln. Anschließend können Sie die Ressourcen basierend auf den hinzugefügten Tags suchen und filtern.

Nachdem Sie ein Tag hinzugefügt haben, können Sie jederzeit Tag-Schlüssel und -Werte bearbeiten oder Tags aus einer Ressource entfernen. Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht.

Tags (Markierungen) haben keine semantische Bedeutung für Amazon EKS und werden ausschließlich als Zeichenfolgen interpretiert. Sie können den Wert eines Tags auf eine leere Zeichenfolge setzen. Sie können jedoch den Wert eines Tags nicht auf null setzen. Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben.

Wenn Sie AWS Identity and Access Management (IAM) verwenden, können Sie steuern, welche Benutzer in Ihrem AWS Konto berechtigt sind, Tags zu verwalten.

Markieren Ihrer -Ressourcen

Die folgenden Amazon-EKS-Ressourcen unterstützen Tags:

- Cluster
- Verwaltete Knotengruppen
- Fargate-Profile

So wenden Sie Tags auf Ressourcen an:

- Wenn Sie die Amazon-EKS-Konsole verwenden, können Sie Tags jederzeit auf neue oder vorhandene Ressourcen anwenden. Dies können Sie über dasTagsauf der entsprechenden Ressourcenseite. Weitere Informationen finden Sie unter Arbeiten mit Tags über die Konsole.
- Wenn Sie `eksctl` verwenden, können Sie Tags auf Ressourcen anwenden, wenn sie mit der `--tags`-Option erstellt wurden.
- Wenn Sie die AWS CLI Amazon EKS-API oder ein AWS SDK verwenden, können Sie mithilfe des `tags` Parameters in der entsprechenden API-Aktion Tags auf neue Ressourcen anwenden. Sie können auch über die `TagResource`-API Tags auf Ressourcen anwenden. Weitere Informationen finden Sie unter TagResource.

Wenn Sie einige Aktionen zur Erstellung von Ressourcen verwenden, können Sie gleichzeitig mit der Erstellung auch Tags für die Ressource angeben. Wenn Tags nicht angewendet werden können, während die Ressource erstellt wird, kann die Ressource nicht erstellt werden. Dieser Mechanismus stellt sicher, dass Ressourcen, die Sie mit Tags versehen möchten, entweder mit den von Ihnen angegebenen Tags oder gar nicht erstellt werden. Wenn Sie Ressourcen bei ihrer Erstellung mit Tags versehen, müssen Sie nach der Erstellung der Ressource keine benutzerdefinierten Tagging-Skripte ausführen.

Tags werden nicht auf andere Ressourcen übertragen, die mit der von Ihnen erstellten Ressource verknüpft sind. Fargate-Profil-Tags werden beispielsweise nicht zu weiteren mit dem Fargate-Profil verknüpften Ressourcen propagiert, z. B. zu den Pods, die mit ihm geplant sind.

Tag-Einschränkungen

Für Tags gelten die folgenden Einschränkungen:

- Maximal 50 Tags können einer Ressource zugeordnet werden.
- Tag-Schlüssel können nicht für eine Ressource wiederholt werden. Jeder Tag-Schlüssel muss eindeutig sein, und jeder Schlüssel darf nur einen Wert besitzen.
- Schlüssel können aus bis zu 128 Zeichen im UTF-8-Format bestehen.
- Werte können aus bis zu 256 Zeichen im UTF-8-Format bestehen.
- Wenn mehrere AWS-Services Ressourcen Ihr Tagging-Schema verwenden, schränken Sie die verwendeten Zeichentypen ein. Einige Services haben möglicherweise Einschränkungen für zulässige Zeichen. Allgemein erlaubte Zeichen sind: Buchstaben, Zahlen, Leerzeichen und die folgenden Sonderzeichen: **`+ - = . _ : / @`**.
- Bei Tag-Schlüsseln und -Werten muss die Groß-/Kleinschreibung beachtet werden.

- Verwenden Sie weder `aws:` noch `AWS:` oder Kombinationen aus Groß- und Kleinbuchstaben von diesen als Präfix für Schlüssel oder Werte, da sie für die -Verwendung reserviert sind. Diese sind nur für die AWS Verwendung reserviert. Sie können keine Tag-Schlüssel oder -Werte mit diesem Präfix bearbeiten oder löschen. Tags mit diesem Präfix werden nicht auf dein tags-per-resource Limit angerechnet.

Markieren von Ressourcen für die Fakturierung

Wenn Sie Tags auf Amazon-EKS-Cluster anwenden, können Sie sie für die Kostenzuweisung in Ihren Kosten- und Nutzungsberichten verwenden. Die Messdaten in Ihren Kosten- und Nutzungsberichten zeigen die Nutzung für alle Ihre Amazon-EKS-Cluster an. Weitere Informationen finden Sie im Benutzerhandbuch zu AWS Billing en unter Erstellen von AWS -Kosten- und Nutzungsberichten.

Insbesondere `aws:eks:cluster-name` mit dem AWS generierten Kostenzuweisungs-Tag können Sie die Kosten für Amazon EC2 EC2-Instances im Cost Explorer nach einzelnen Amazon EKS-Clustern aufschlüsseln. Dieses Tag erfasst jedoch nicht die Kosten der Steuerebene. Das Tag wird automatisch zu Amazon-EC2-Instances hinzugefügt, die Teil eines Amazon-EKS-Clusters sind. Dieses Verhalten tritt unabhängig davon auf, ob die Instances mit verwalteten Amazon-EKS-Knotengruppen, Karpenter oder direkt mit Amazon EC2 bereitgestellt werden. Dieses spezielle Tag wird nicht auf das Limit von 50 Tags angerechnet. Um das Tag zu verwenden, muss der Kontoinhaber es in der AWS Billing -Konsole oder mithilfe der API aktivieren. Wenn ein Inhaber eines AWS Organizations Verwaltungskontos das Tag aktiviert, wird es auch für alle Mitgliedskonten der Organisation aktiviert.

Sie können Ihre Rechnungsinformationen auch auf der Grundlage von Ressourcen organisieren, die dieselben Tag-Schlüsselwerte haben. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen taggen und dann Ihre Fakturierungsinformationen organisieren. Auf diese Weise können Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen. Weitere Informationen zum Einrichten eines Kostenzuordnungsberichts mit Markierungen finden Sie unter Monatlicher Kostenzuordnungsbericht im Benutzerhandbuch für AWS Billing .

> **Note**
>
> Wenn Sie die Berichterstellung gerade erst aktiviert haben, werden die Daten für den aktuellen Monat nach 24 Stunden bereitgestellt.

Cost Explorer ist ein Berichtstool, das im Rahmen des AWS kostenlosen Kontingents verfügbar ist. Sie können Cost Explorer verwenden, um Diagramme Ihrer Amazon-EKS-Ressourcen der letzten 13 Monate anzuzeigen. Sie können auch prognostizieren, wie viel Sie wahrscheinlich für die nächsten drei Monate ausgeben werden. Sie können Muster sehen, wie viel Sie für AWS -Ressourcen im Zeitablauf ausgeben. Sie können es zum Beispiel verwenden, um Bereiche zu identifizieren, die eine genauere Untersuchung erfordern, sowie um Trends auszumachen, die Ihnen helfen, Ihre Kosten zu verstehen. Sie können auch Zeitbereiche für die Daten angeben und die Daten nach Tagen oder Monate anzeigen lassen.

Arbeiten mit Tags über die Konsole

Mithilfe der Amazon-EKS-Konsole können Sie die Tags im Zusammenhang mit neuen oder vorhandenen Clustern und verwalteten Knotengruppen verwalten.

Wenn Sie eine ressourcenspezifische Seite in der Amazon-EKS-Konsole auswählen, wird auf der Seite eine Liste der Ressourcen angezeigt. Wenn Sie beispielsweise Clusters (Cluster) im Navigationsbereich auswählen, zeigt die Konsole eine Liste der Amazon EKS-Cluster an. Wenn Sie eine Ressource aus einer dieser Listen auswählen (z. B. einen bestimmten Cluster) und die Ressource Tags unterstützt, können Sie deren Tags auf der Registerkarte Tags anzeigen und verwalten.

Sie können auch den Tag-Editor in der verwenden AWS Management Console, der eine einheitliche Methode zur Verwaltung Ihrer Tags bietet. Weitere Informationen finden Sie unter Taggen Ihrer AWS Ressourcen mit dem Tag-Editor im AWS Tag-Editor-Benutzerhandbuch.

Hinzufügen von Tags zu einer Ressource bei der Erstellung

Sie können Tags zu Amazon EKS-Clustern und verwalteten Knotengruppen hinzufügen, wenn Sie sie erstellen. Weitere Informationen finden Sie unter Erstellen eines Amazon-EKS-Clusters.

Hinzufügen und Löschen von Tags für eine Ressource

Sie können die Tags, die mit Ihren Clustern verbunden sind, direkt auf der Seite der Ressource hinzufügen oder löschen.

So fügen Sie ein Tag zu einer einzelnen Ressource hinzu oder löschen es

1. Öffnen Sie die Amazon EKS-Konsole unterhttps://console.aws.amazon.com/eks/home#/clustersaus.
2. Wählen Sie in der Navigationsleiste den aus, den Sie verwenden AWS-Region möchten.

3. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
4. Wählen Sie einen bestimmten Cluster aus.
5. Wählen Sie die Registerkarte Tags und dann Manage tags (Tags verwalten).
6. Fügen Sie auf der Seite Manage tags (Tags verwalten) Ihre Tags bei Bedarf hinzu oder löschen Sie sie.
   - Um einen Tag hinzuzufügen, wählen Sie Add tag (Tag hinzufügen). Geben Sie dann den Schlüssel und den Wert für jedes Tag an.
   - Um ein Tag zu entfernen, wählen Sie Remove tag (Tag entfernen) aus.
7. Wiederholen Sie diesen Vorgang für jedes Tag, das Sie hinzufügen oder löschen möchten.
8. Wählen Sie zum Abschluss Update (Aktualisieren) aus.

Arbeiten mit Tags mittels CLI, API oder `eksctl`

Verwenden Sie die folgenden AWS CLI Befehle oder Amazon EKS-API-Operationen, um die Tags für Ihre Ressourcen hinzuzufügen, zu aktualisieren, aufzulisten und zu löschen. Sie können `eksctl` nur dann zum Hinzufügen von Tags verwenden, wenn Sie gleichzeitig neue Ressourcen mit einem Befehl erstellen.

Markierungsunterstützung für Amazon EKS-Ressourcen

Aufgabe	AWS CLI	AWS Tools for Windows PowerShell	API-Aktion
Fügen Sie einen oder mehrere Tags hinzu oder überschreiben Sie sie.	`tag-resource`	`Add-EKSResourceTag`	`TagResource`
Löschen Sie ein oder mehrere Tags.	`untag-resource`	`Remove-EKSResourceTag`	`UntagResource`

Die folgenden Beispiele zeigen, wie man Tags an Ressourcen mithilfe der AWS CLI hinzufügt oder entfernt.

Beispiel 1: Markieren eines bestehenden Clusters

Der folgende Befehl markiert einen vorhandenen Cluster.

```
aws eks tag-resource --resource-arn resource_ARN --tags team=devs
```

Beispiel 2: Entfernen von Tags von einem bestehenden Cluster

Der folgende Befehl löscht ein Tag von einem bestehenden Cluster.

```
aws eks untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Beispiel 3: Tags für eine Ressource auflisten

Der folgende Befehl listet die Tags auf, die einer vorhandenen Ressource zugeordnet sind.

```
aws eks list-tags-for-resource --resource-arn resource_ARN
```

Wenn Sie einige Aktionen zur Erstellung von Ressourcen verwenden, können Sie gleichzeitig mit der Erstellung der Ressource auch Tags angeben. Die folgenden Aktionen unterstützen das Taggen beim Erstellen einer Ressource.

Aufgabe	AWS CLI	AWS Tools for Windows PowerShell	API-Aktion	**eksctl**
Erstellen eines Clusters	create-cluster	New-EKSCluster	CreateCluster	create cluster
Erstellen einer verwalteten Knotengruppe	create-nodegroup	New-EKSNodegroup	CreateNodegroup	create nodegroup
Erstellen eines Fargate-Profils	create-fargate-profile	New-EKSFargateProfile	CreateFargateProfile.html	create fargateprofile

* Wenn Sie beim Erstellen einer verwalteten Knotengruppe auch die Amazon-EC2-Instances mit Tags versehen möchten, erstellen Sie die verwaltete Knotengruppe mithilfe einer Startvorlage.

Weitere Informationen finden Sie unter Markieren von Amazon-EC2-Instances. Wenn Ihre Instances bereits vorhanden sind, können Sie die Instanzen manuell mit Tags versehen. Weitere Informationen finden Sie unter Tagging your resources im Amazon EC2 EC2-Benutzerhandbuch.

Amazon-EKS-Service-Quotas

Amazon EKS hat Service Quotas integriert, einen AWS Service, mit dem Sie Ihre Kontingente von einem zentralen Ort aus einsehen und verwalten können. Weitere Informationen zu Service Quotas finden Sie unter Was sind Service Quotas im Benutzerhandbuch für Service Quotas. Mit der Integration von Service Quotas können Sie mithilfe von und schnell den Wert Ihrer Amazon EKS- und AWS Fargate Servicekontingente AWS Management Console ermitteln AWS CLI.

AWS Management Console

Um Amazon EKS- und Fargate-Servicekontingente anzuzeigen, verwenden Sie den AWS Management Console

1. Öffnen Sie die Service-Quotas-Konsole unter https://console.aws.amazon.com/servicequotas/.
2. Wählen Sie im linken Navigationsbereich die Option AWS-Services aus.
3. Von der AWS-Services-Liste suchen Sie nach und wählen Sie Amazon Elastic Kubernetes Service (Amazon EKS) oder AWS Fargate aus.

   In der Liste der Service-Kontingente finden Sie den Namen des Servicekontingents, den angewendeten Wert (falls verfügbar), das AWS Standardkontingent und ob der Kontingentwert anpassbar ist.
4. Wählen Sie den Kontingentnamen, um zusätzliche Informationen zu einem Service Quota anzuzeigen, z. B. seine Beschreibung.
5. (Optional) Um eine Kontingenterhöhung zu beantragen, wählen Sie das Kontingent, das Sie erhöhen möchten, und dann Request quota increase (Kontingenterhöhung beantragen) aus, geben Sie die erforderlichen Informationen ein, und wählen Sie dann Request (Beantragen) aus.

Weitere Informationen zum Umgang mit Servicekontingenten mithilfe von finden Sie im Service Quotas User Guide. AWS Management Console Informationen zum Beantragen einer Kontingenterhöhung finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

AWS CLI

Um Amazon EKS- und Fargate-Servicekontingente anzuzeigen, verwenden Sie den AWS CLI

Führen Sie den folgenden Befehl aus, um Ihre Amazon-EKS-Kontingente anzuzeigen.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code eks \
    --output table
```

Führen Sie den folgenden Befehl aus, um Ihre Fargate-Kontingente anzuzeigen.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code fargate \
    --output table
```

> **Note**
>
> Das zurückgegebene Kontingent entspricht der Anzahl von Amazon-ECS-Aufgaben oder Amazon-EKS-Pods, die in diesem Konto in der aktuellen AWS-Region gleichzeitig in Fargate ausgeführt werden können.

Weitere Informationen zum Umgang mit Servicekontingenten mithilfe von finden Sie `service-quotas`in der AWS CLI Befehlsreferenz. AWS CLI Informationen zum Beantragen einer Kontingenterhöhung finden Sie unter dem `request-service-quota-increase`-Befehl in der AWS CLI -Befehlsreferenz.

Service Quotas

Name	Standard	Anpas	Beschreibung
Zugriffseinträge pro Cluster	Jede unterstützte Region: 3 000	Nein	Die maximale Anzahl an Zugriffseinträgen pro Cluster.

Name	Standard	Anpas	Beschreibung
Cluster	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von EKS-Clustern in diesem Konto in der aktuellen Region.
Sicherheitsgruppen der Steuerebene pro Cluster	Jede unterstützte Region: 4	Nein	Die maximale Anzahl von Steuerebenen-Sicherheitsgruppen pro Cluster (werden beim Erstellen des Clusters angegeben)
EKS Anywhere Enterprise-Abonnements	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl an EKS Anywhere Enterprise-Abonnements für dieses Konto in der aktuellen Region.
Fargate-Profile pro Cluster	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von Fargate-Profilen pro Cluster.
Label-Paare pro Fargate-Profilselektor	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Label-Paaren pro Fargate-Profilauswahl.
Verwaltete Knotengruppen pro Cluster	Jede unterstützte Region: 30	Yes (Ja)	Die maximale Anzahl verwalteter Knotengruppen pro Cluster.
Knoten pro verwalteter Knotengruppe	Jede unterstützte Region: 450	Yes (Ja)	Die maximale Anzahl von Knoten pro verwalteter Knotengruppe.

Name	Standard	Anpas	Beschreibung
CIDR-Bereiche für öffentlichen Endpunktzugriff pro Cluster	Jede unterstützte Region: 40	Nein	Die maximale Anzahl von CIDR-Bereichen für öffentliche Endpunkte pro Cluster (werden beim Erstellen oder Aktualisieren des Clusters angegeben).
Registrierte Cluster	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von registrierten Clustern in diesem Konto in der aktuellen Region.
Selektoren pro Fargate-Profil	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von Selektoren pro Fargate-Profil.

> **Note**
>
> Die Standardwerte sind die anfänglichen Kontingente, die von festgelegt wurden AWS. Diese Standardwerte sind unabhängig von den tatsächlich angewendeten Kontingentwerten und den maximal möglichen Servicekontingenten. Weitere Informationen finden Sie unter Terminologie in Service Quotas im Service Quotas User Guide.

Diese Servicekontingente werden unter Amazon Elastic Kubernetes Service (Amazon EKS) in der Service-Quotas-Konsole aufgeführt. Informationen zum Anfordern einer Kontingenterhöhung für Werte, die als anpassbar angezeigt werden, finden Sie unter Beantragen einer Kontingenterhöhung im Benutzerhandbuch von Service Quotas.

AWS Fargate Dienstkontingente

Diese Service Quotas sind unter dem AWS Fargate-Service in der Service Quotas-Konsole aufgeführt. In der folgenden Tabelle werden nur die Kontingente beschrieben, die auch für Amazon EKS gelten. Sie können Alarme konfigurieren, mit denen Sie benachrichtigt werden, wenn sich Ihre

Nutzung einem Servicekontingent nähert. Weitere Informationen finden Sie unter Erstellen eines CloudWatch-Alarms zur Überwachung von Fargate Ressourcennutzungsmetriken.

Neue haben AWS-Konten möglicherweise niedrigere anfängliche Kontingente, die sich im Laufe der Zeit erhöhen können. Fargate überwacht ständig die Kontonutzung innerhalb der einzelnen AWS-Region Konten und erhöht dann automatisch die Kontingente je nach Nutzung. Außerdem können Sie eine Kontingenterhöhung für Werte anfordern, die als anpassbar angezeigt werden. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Name	Standard	Anpassbar	Beschreibung
Anzahl der On-Demand-vCPU-Ressourcen von Fargate	6	Ja	Die Anzahl der Fargate-vCPUs, die in diesem Konto in der aktuellen Region gleichzeitig als Fargate-On-Demand-Ressourcen ausgeführt werden können.

> **Note**
>
> Die Standardwerte sind die anfänglichen Kontingente, die von festgelegt wurden AWS. Diese Standardwerte sind unabhängig von den tatsächlich angewendeten Kontingentwerten und den maximal möglichen Servicekontingenten. Weitere Informationen finden Sie unter Terminologie in Service Quotas im Service Quotas User Guide.

> **Note**
>
> Fargate erzwingt zusätzlich Ratenkontingente für Amazon-ECS-Aufgaben und den Start von Amazon-EKS-Pods. Weitere Informationen finden Sie unter AWS Fargate Drosselung von Kontingenten im Amazon ECS-Handbuch.

Sicherheit in Amazon EKs

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS Ist bei Amazon EKS für die Kubernetes Steuerungsebene verantwortlich, die die Knoten und die `etcd` Datenbank der Steuerungsebene umfasst. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS -Compliance-Programme regelmäßig. Informationen zu den Compliance-Programmen, die für Amazon EKS gelten, finden Sie unter AWS -Services im Geltungsbereich nach Compliance-Programm.
- Sicherheit in der Cloud – Ihre Verantwortlichkeit umfasst die folgenden Bereiche.
  - Die Sicherheitskonfiguration der Datenebene, einschließlich der Konfiguration der Sicherheitsgruppen, die den Durchgang des Datenverkehrs aus der Amazon-EKS-Steuerebene in die VPC des Kunden zulassen.
  - Die Konfiguration der Knoten und der Container selbst.
  - Das Betriebssystem (einschließlich Updates und Sicherheits-Patches) des Knotens
  - Andere zugehörige Anwendungssoftware:
    - Einrichten und Verwalten von Kontrollelementen für das Netzwerk (wie beispielsweise Firewall-Regeln)
    - Verwalten des Identity and Access Management auf Plattformebene, entweder mit oder zusätzlich zu IAM
  - Die Vertraulichkeit Ihrer Daten, die Anforderungen Ihres Unternehmens und geltende Gesetze und Vorschriften

Diese Dokumentation zeigt Ihnen, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Amazon EKS einsetzen können. Die folgenden Themen zeigen Ihnen, wie Sie Amazon EKS konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Services nutzen können, die Ihnen helfen, Ihre Amazon EKS-Ressourcen zu überwachen und zu sichern.

> **Note**
>
> Linux-Container bestehen aus Kontrollgruppen (cgroups) und Namespaces, die helfen, den Zugriff eines Containers einzuschränken, aber alle Container teilen sich denselben Linux-Kernel wie die Host-Amazon-EC2-Instance. Es wird dringend davon abgeraten, einen Container als Root-Benutzer (UID 0) auszuführen oder einem Container Zugriff auf Hostressourcen oder Namespaces wie das Hostnetzwerk oder den Host-PID-Namespace zu gewähren, da dies die Effektivität der von Containern bereitgestellten Isolation verringert.

Themen

Zertifikatsignierung

Die Kubernetes-Zertifikats-API automatisiert die Bereitstellung der X.509-Anmeldeinformationen. Das API-Feature verfügt über eine Befehlszeilenschnittstelle, über die Kubernetes-API-Clients X.509-Zertifikate von einer Zertifizierungsstelle (CA) anfordern und abrufen können. Sie können die `CertificateSigningRequest` (CSR)-Ressource verwenden, um anzufordern, dass ein benannter Unterzeichner das Zertifikat signiert. Ihre Anfragen werden entweder genehmigt oder abgelehnt, bevor sie signiert werden. Kubernetes unterstützt sowohl integrierte Unterzeichner als auch benutzerdefinierte Unterzeichner mit klar definierten Verhaltensweisen. Auf diese Weise können Kunden vorhersehen, was mit ihren CSRs passiert. Weitere Informationen zur Zertifikatsignatur finden Sie unter Signieren von Anforderungen.

Einer der integrierten Unterzeichner ist `kubernetes.io/legacy-unknown`. Die `v1beta1`-API der CSR-Ressource berücksichtigte diesen Unterzeichnertyp „legacy-unknown“. Die stabile `v1`-API von CSR lässt jedoch nicht zu, dass `signerName` auf `kubernetes.io/legacy-unknown` festgelegt wird.

Amazon-EKS-Version `1.21` und ältere Versionen ermöglichten es, den Wert `legacy-unknown` als den `signerName` in der `v1beta1`-CSR-API festzulegen. Diese API ermöglicht es der Amazon EKS Certificate Authority (CA), Zertifikate zu generieren. In Kubernetes Version `1.22` wurde die `v1beta1`-CSR-API durch die `v1`-CSR-API ersetzt. Diese API unterstützt den signerName von „legacy-unknown“ nicht. Wenn Sie Amazon EKS CA zum Generieren von Zertifikaten verwenden möchten, müssen Sie einen benutzerdefinierten Unterzeichner verwenden. Dies wurde in Version `1.22` von Amazon EKS eingeführt. Um die CSR-`v1`-API-Version zu verwenden und ein neues Zertifikat zu generieren, müssen Sie alle vorhandenen Manifeste und API-Clients migrieren. Bestehende Zertifikate, die mit der vorhandenen `v1beta1`-API erstellt wurden, sind gültig und funktionieren, bis sie ablaufen. Diese umfasst die folgenden Funktionen:

- Vertrauensverteilung: keine. Es gibt kein Standardvertrauen oder eine Standardverteilung für diesen Unterzeichner in einem Kubernetes-Cluster.
- Zulässige Themen: beliebig
- Zulässige x509-Erweiterungen: Besitzer- subjectAltName und Schlüsselnutzungserweiterungen und verwirft andere Erweiterungen
- Zulässige Schlüsselnutzung: Darf keine anderen Nutzungen als [„Schlüsselverschlüsselung“, „digitale Signatur“, „Serverauth“] enthalten

  > **Note**
  >
  > Das Signieren von Clientzertifikaten wird nicht unterstützt.

- Ablauf/Zertifikatslebensdauer: 1 Jahr (Standard und Maximum)
- CA-Bit zulässig/unzulässig: unzulässig

Beispiel CSR-Erstellung mit signerName

Diese Schritte veranschaulichen, wie Sie ein Bereitstellungszertifikat für DNS-Namen `myserver.default.svc` mit `signerName: beta.eks.amazonaws.com/app-serving` erstellen. Verwenden Sie dies als Leitfaden für Ihre eigene Umgebung.

1. Führen Sie den Befehl `openssl genrsa -out myserver.key 2048` aus, um einen privaten RSA-Schlüssel zu erzeugen.

```
openssl genrsa -out myserver.key 2048
```

2. Führen Sie den folgenden Befehl aus, um eine Zertifikatsanforderung zu erstellen.

```
openssl req -new -key myserver.key -out myserver.csr -subj "/
CN=myserver.default.svc"
```

3. Generieren Sie einen `base64`-Wert für die CSR-Variablen für die Verwendung in einem späteren Schritt.

```
base_64=$(cat myserver.csr | base64 -w 0 | tr -d "\n")
```

4. Führen Sie den folgenden Befehl aus, um eine Datei mit dem Namen `mycsr.yaml` zu erstellen. Im folgenden Beispiel ist `beta.eks.amazonaws.com/app-serving` der `signerName`.

```
cat >mycsr.yaml <<EOF
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: myserver
spec:
  request: $base_64
  signerName: beta.eks.amazonaws.com/app-serving
  usages:
    - digital signature
    - key encipherment
    - server auth
EOF
```

5. Reichen Sie die CSR ein.

```
kubectl apply -f mycsr.yaml
```

6. Genehmigen Sie das Bereitstellungszertifikat.

```
kubectl certificate approve myserver
```

7. Stellen Sie sicher, dass das Zertifikat ausgestellt wurde.

```
kubectl get csr myserver
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME       AGE     SIGNERNAME                                  REQUESTOR
 CONDITION
myserver   3m20s   beta.eks.amazonaws.com/app-serving   kubernetes-admin
 Approved,Issued
```

8. Exportieren Sie das ausgestellte Zertifikat.

```
kubectl get csr myserver -o jsonpath='{.status.certificate}'| base64 -d
 > myserver.crt
```

Überlegungen zur Zertifikatssignierung vor dem Upgrade Ihres Clusters auf Kubernetes 1.24

In Kubernetes `1.23` und älter werden `kubelet`-Bereitstellungszertifikate mit nicht verifizierbaren IP- und DNS-Subject Alternative Names (SANs) automatisch mit nicht verifizierbaren SANs ausgestellt. Diese SANs werden im bereitgestellten Zertifikat weggelassen. In `1.24`- und neuere Clustern werden keine `kubelet`-Bereitstellungszertifikate ausgestellt, wenn ein SAN nicht verifiziert werden kann. Dadurch wird verhindert, dass die `kubectl exec`- und `kubectl logs`-Befehle funktionieren.

Stellen Sie vor dem Upgrade Ihres Clusters auf `1.24` fest, ob Ihr Cluster Zertifikatsignierungsanforderungen (CSR) hat, die nicht genehmigt wurden, indem Sie die folgenden Schritte ausführen:

1. Führen Sie den folgenden Befehl aus.

```
kubectl get csr -A
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME        AGE   SIGNERNAME                        REQUESTOR
                        REQUESTEDDURATION   CONDITION
```

```
csr-7znmf   90m   kubernetes.io/kubelet-serving
 system:node:ip-192-168-42-149.region.compute.internal      <none>
 Approved
csr-9xx5q   90m   kubernetes.io/kubelet-serving
 system:node:ip-192-168-65-38.region.compute.internal      <none>
 Approved, Issued
```

Wenn die zurückgegebene Ausgabe eine CSR mit einem `kubernetes.io/kubelet-serving`-Unterzeichner anzeigt, der `Approved`, aber nicht `Issued` für einen Knoten ist, müssen Sie die Anforderung genehmigen.

2. Genehmigen Sie die CSR manuell. Ersetzen Sie `csr-7znmf` durch Ihren eigenen Wert.

```
kubectl certificate approve csr-7znmf
```

Um CSRs in Zukunft automatisch zu genehmigen, empfehlen wir, dass Sie einen genehmigenden Controller schreiben, der CSRs automatisch validieren und genehmigen kann, die IP- oder DNS-SANs enthalten, die Amazon EKS nicht verifizieren kann.

Identitäts- und Zugriffsverwaltung für Amazon EKS

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert (mit Berechtigungen ausgestattet) werden kann, um Amazon-EKS-Ressourcen zu nutzen. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon EKS ausführen.

Service-Benutzer – Wenn Sie den Amazon–EKS-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Je mehr Amazon-EKS-Funktionen Sie für Ihre Arbeit nutzen, desto mehr Berechtigungen benötigen Sie möglicherweise. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Unter Fehlersuche bei IAM finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Funktion in Amazon EKS haben.

Service-Administrator – Wenn Sie in Ihrem Unternehmen für die Amazon-EKS-Ressourcen zuständig sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon EKS. Ihre Aufgabe besteht darin, zu bestimmen, auf welche Amazon-EKS-Funktionen und -Ressourcen Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit Amazon EKS verwenden kann, finden Sie unter Funktionsweise von Amazon EKS 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 EKS erstellen können. Beispiele für identitätsbasierte Amazon-EKS-Richtlinien, die Sie in IAM verwenden können, finden Sie unter Beispiele für identitätsbasierte Amazon-EKS-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, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. 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 Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, 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.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb von Ihnen 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 einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten 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 Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen 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 Verwenden von IAM-Rollen 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 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 So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. 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 können Sie Aktionen ausführen, die dann eine andere Aktion in einem anderen Service auslösen. 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.

- Anwendungen, die auf Amazon EC2 ausgeführt werden — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

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. Berechtigungen in den Richtlinien bestimmen, ob die 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 bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

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 Erstellen von IAM-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 eingebundenen Richtlinie wählen, finden Sie unter Auswahl 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.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

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.

- Service Control Policies (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 innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs 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 EKS mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon EKS zu verwalten, sollten Sie verstehen, welche IAM-Funktionen für die Verwendung mit Amazon EKS verfügbar sind. Einen allgemeinen Überblick darüber, wie Amazon EKS und andere AWS Services mit IAM zusammenarbeiten, finden Sie im IAM-Benutzerhandbuch unter AWS Services, die mit IAM funktionieren.

Themen

- Identitätsbasierte Amazon-EKS-Richtlinien

- Ressourcenbasierte Amazon-EKS-Richtlinien
- Autorisierung auf der Basis von Amazon-EKS-Tags
- Amazon EKS IAM-Rollen

Identitätsbasierte Amazon-EKS-Richtlinien

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. Amazon EKS unterstützt bestimmte Aktionen, Ressourcen und Zustandsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Aktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `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 keine passende API-Operation gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen in Amazon EKS verwenden das folgende Präfix vor der Aktion: `eks:`. Um jemandem beispielsweise die Berechtigung zum Abrufen von Informationen zu einem Amazon-EKS-Cluster zu erteilen, fügen Sie die Aktion `DescribeCluster` in seine Richtlinie ein. Richtlinienanweisungen müssen entweder ein `Action`- oder ein `NotAction`-Element enthalten.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": ["eks:action1", "eks:action2"]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "eks:Describe*"
```

Eine Liste der Amazon-EKS-Aktionen finden Sie unter Von Amazon Elastic Kubernetes Service definierte Aktionen in der Serviceautorisierungsreferenz.

Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen 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": "*"
```

Die Amazon-EKS-Cluster-Ressource verfügt über den folgenden ARN:

```
arn:aws:eks:region-code:account-id:cluster/cluster-name
```

Weitere Informationen zum Format von ARNs finden Sie unter Amazon-Ressourcennamen (ARNs) und AWS Service-Namespaces.

Um beispielsweise den Cluster mit dem Namen *`my-cluster`* in Ihrer Anweisung anzugeben, verwenden Sie den folgenden ARN:

```
"Resource": "arn:aws:eks:region-code:111122223333:cluster/my-cluster"
```

Um alle Cluster anzugeben, die zu einem bestimmten Konto gehören AWS-Region, und verwenden Sie den Platzhalter (*):

```
"Resource": "arn:aws:eks:region-code:111122223333:cluster/*"
```

Einige Amazon-EKS-Aktionen, z. B. das Erstellen von Ressourcen, können für bestimmte Ressourcen nicht ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

```
"Resource": "*"
```

Eine Liste der Amazon-EKS-Ressourcentypen und ihrer ARNs finden Sie unter Von Amazon Elastic Kubernetes Service definierte Ressourcen in der Service-Autorisierungsreferenz. Um zu erfahren, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, lesen Sie von Amazon Elastic Kubernetes Service definierte Aktionen.

Bedingungsschlüssel

Amazon EKS definiert seinen eigenen Satz von Konditionsschlüsseln und unterstützt auch die Verwendung einiger globaler Konditionsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter AWS Globale Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Sie können Bedingungsschlüssel festlegen, wenn Sie Ihrem Cluster einen OpenID Connect-Anbieter zuordnen. Weitere Informationen finden Sie unter Beispiel für eine IAM-Richtlinie.

Alle Amazon EC2-Aktionen unterstützen die Bedingungsschlüssel `aws:RequestedRegion` und `ec2:Region`. Weitere Informationen finden Sie unter Beispiel: Beschränken des Zugriffs auf ein bestimmtes Objekt. AWS-Region

Eine Liste der Amazon EKS-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Elastic Kubernetes Service in der Service-Autorisierungsreferenz. Um zu erfahren, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, lesen Sie von Amazon Elastic Kubernetes Service definierte Aktionen.

Beispiele

Beispiele für identitätsbasierte Amazon-EKS-Richtlinien finden Sie unter Beispiele für identitätsbasierte Amazon-EKS-Richtlinien.

Wenn Sie einen Amazon-EKS-Cluster erstellen, werden dem IAM-Prinzipal, der den Cluster erstellt, automatisch `system:masters`-Berechtigungen in der rollenbasierten Zugriffssteuerungs(RBAC)-Konfiguration des Clusters in der Amazon-EKS-Steuerebene erteilt. Dieser Prinzipal wird in keiner sichtbaren Konfiguration angezeigt. Achten Sie daher darauf, welcher Prinzipal den Cluster ursprünglich erstellt hat. Um zusätzlichen IAM-Prinzipals die Möglichkeit zu geben, mit Ihrem Cluster zu interagieren, müssen Sie die `aws-auth ConfigMap` innerhalb von Kubernetes bearbeiten und ein Kubernetes `rolebinding` oder `clusterrolebinding` mit dem Namen einer `group` erstellen, den Sie in der `aws-auth ConfigMap` angeben.

Weitere Informationen zur Arbeit mit dem finden Sie ConfigMap unterZugriff auf Kubernetes APIs gewähren .

Ressourcenbasierte Amazon-EKS-Richtlinien

Amazon EKS unterstützt keine ressourcenbasierten Richtlinien.

Autorisierung auf der Basis von Amazon-EKS-Tags

Sie können Tags an Amazon-EKS-Ressourcen anhängen oder Tags in einer Anforderung an Amazon EKS übergeben. Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder `aws:TagKeys`Bedingung verwenden. Weitere Informationen über die Markierung von Amazon-EKS-Ressourcen finden Sie unter Kennzeichnen Ihrer Amazon EKS-Ressourcen . Weitere Informationen darüber, mit welchen Aktionen Sie Tags in Bedingungsschlüsseln verwenden können, finden Sie unter Von Amazon EKS definierte Aktionen in der Service-Authorization-Referenz.

Amazon EKS IAM-Rollen

Eine IAM-Rolle ist eine Entität innerhalb Ihres AWS Kontos, die über bestimmte Berechtigungen verfügt.

Verwenden temporärer Anmeldeinformationen mit Amazon EKS

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie AWS STS API-Operationen wie AssumeRoleoder GetFederationTokenaufrufen.

Amazon EKS unterstützt die Verwendung temporärer Anmeldeinformationen.

Serviceverknüpfte Rollen

Mit dienstbezogenen Rollen können AWS Dienste auf Ressourcen in anderen Diensten zugreifen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto angezeigt und gehören zum Service. Ein Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, jedoch nicht bearbeiten.

Amazon EKS unterstützt Service-verknüpfte Rollen. Details zum Erstellen oder Verwalten von serviceverknüpften Amazon-EKS-Rollen finden Sie unter Verwendung von serviceverknüpften Rollen für Amazon EKS.

Servicerollen

Dieses Feature ermöglicht einem Service das Annehmen einer Servicerolle in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem IAM-Konto angezeigt und gehören zum Konto. Dies bedeutet, dass ein IAM-Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Services beeinträchtigen.

Amazon EKS unterstützt Servicerollen. Weitere Informationen finden Sie unter Amazon-EKS-Cluster-IAM-Rolle und Amazon-EKS-Knoten-IAM-Rolle.

Auswählen einer IAM-Rolle in Amazon EKS

Wenn Sie eine Cluster-Ressource in Amazon EKS erstellen, müssen Sie eine Rolle auswählen, damit Amazon EKS in Ihrem Namen auf mehrere andere AWS Ressourcen zugreifen kann. Wenn Sie zuvor eine Servicerolle erstellt haben, stellt Ihnen Amazon EKS eine Liste der Rollen bereit, aus denen Sie auswählen können. Es ist wichtig, eine Rolle auszuwählen, an die die von Amazon EKS verwalteten Richtlinien angefügt sind. Weitere Informationen finden Sie unter Überprüfen, ob eine Clusterrolle vorhanden ist und Nach einer vorhandenen Knotenrolle suchen.

Beispiele für identitätsbasierte Amazon-EKS-Richtlinien

Standardmäßig haben IAM-Benutzer und -Rollen nicht die Berechtigung, Amazon-EKS-Ressourcen zu erstellen oder zu ändern. 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 IAM-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.

Wenn Sie einen Amazon-EKS-Cluster erstellen, werden dem IAM-Prinzipal, der den Cluster erstellt, automatisch `system:masters`-Berechtigungen in der rollenbasierten Zugriffssteuerungs(RBAC)-Konfiguration des Clusters in der Amazon-EKS-Steuerebene erteilt. Dieser Prinzipal wird in keiner sichtbaren Konfiguration angezeigt. Achten Sie daher darauf, welcher Prinzipal den Cluster ursprünglich erstellt hat. Um zusätzlichen IAM-Prinzipals die Möglichkeit zu geben, mit Ihrem Cluster zu interagieren, müssen Sie die `aws-auth ConfigMap` innerhalb von Kubernetes bearbeiten und ein Kubernetes `rolebinding` oder `clusterrolebinding` mit dem Namen einer `group` erstellen, den Sie in der `aws-auth ConfigMap` angeben.

Weitere Informationen zur Arbeit mit dem ConfigMap finden Sie unterZugriff auf Kubernetes APIs gewähren .

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der Amazon-EKS-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für IAM-Benutzer
- Erstellen eines Kubernetes-Clusters in der AWS Cloud
- Erstellen Sie einen lokalen Kubernetes-Cluster auf einem Outpost
- Aktualisieren eines Kubernetes-Clusters
- Auflisten oder Beschreiben aller Cluster

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien können festlegen, ob jemand Amazon-EKS-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder daraus löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen,

die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Amazon-EKS-Konsole

Um auf die Amazon EKS-Konsole zugreifen zu können, muss ein IAM-Prinzipal über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen ermöglichen es dem Principal, Details zu den Amazon EKS-Ressourcen in Ihrem AWS Konto aufzulisten und einzusehen. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die restriktiver als die mindestens erforderlichen Berechtigungen ist, funktioniert die Konsole für Prinzipals, denen diese Richtlinie zugewiesen ist, nicht wie vorgesehen.

Um sicherzustellen, dass Ihre IAM-Prinzipals die Amazon EKS-Konsole weiterhin verwenden können, erstellen Sie eine Richtlinie mit Ihrem eigenen eindeutigen Namen, z. B. `AmazonEKSAdminPolicy`. Hängen Sie die Richtlinie an die Prinzipale an. Informationen finden Sie im Abschnitt Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

> **Important**
>
> Mit der folgenden Beispielrichtlinie kann ein Prinzipal Informationen auf der Registerkarte Konfiguration in der Konsole anzeigen. Um auf Informationen auf den Registerkarten Übersicht und Ressourcen in der AWS Management Console zuzugreifen, benötigt der Prinzipal zudem Kubernetes-Berechtigungen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "eks:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "eks.amazonaws.com"
```

```
                }
            }
        }
    ]
}
```

Sie müssen Prinzipalen, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die Sie ausführen möchten.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für IAM-Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI API oder. AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
```

```
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Erstellen eines Kubernetes-Clusters in der AWS Cloud

AWS-Region Sie können das durch das AWS-Region ersetzen, in dem Sie einen AWS-Region Cluster erstellen möchten. Wenn in der AWS Management Console die Warnung The actions in your policy do not support resource-level permissions and require you to choose **`All resources`** (Die Aktionen in Ihrer Richtlinie unterstützen keine Berechtigungen auf Ressourcenebene und erfordern die Auswahl von) angezeigt wird, können Sie diese ignorieren. Wenn Ihr Konto die *`AWSServiceRoleForAmazonEKS`*Rolle bereits besitzt, können Sie die `iam:CreateServiceLinkedRole` Aktion aus der Richtlinie entfernen. Wenn Sie jemals einen Amazon-EKS-Cluster in Ihrem Konto erstellt haben, ist diese Rolle bereits vorhanden, sofern Sie sie nicht gelöscht haben.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "eks:CreateCluster",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-cluster"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
eks.amazonaws.com/AWSServiceRoleForAmazonEKS",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iam:AWSServiceName": "eks"
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/cluster-role-name"
        }
    ]
}
```

Erstellen Sie einen lokalen Kubernetes-Cluster auf einem Outpost

AWS-Region Sie können das durch das ersetzen, in AWS-Region dem Sie einen Cluster AWS-Region erstellen möchten. Wenn in der AWS Management Console die Warnung The actions in your policy do not support resource-level permissions and require you to choose **`All resources`** (Die Aktionen in Ihrer Richtlinie unterstützen keine Berechtigungen auf Ressourcenebene und erfordern die Auswahl von) angezeigt wird, können Sie diese ignorieren. Wenn Ihr Konto bereits über die `AWSServiceRoleForAmazonEKSLocalOutpost`-Rolle verfügt, können Sie die `iam:CreateServiceLinkedRole`-Aktion aus der Richtlinie entfernen. Wenn Sie jemals einen lokalen Amazon-EKS-Cluster auf einem Outpost in Ihrem Konto erstellt haben, ist diese Rolle bereits vorhanden, sofern Sie sie nicht gelöscht haben.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "eks:CreateCluster",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-cluster"
        },
        {
            "Action": [
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "iam:GetRole"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/outposts.eks-
local.amazonaws.com/AWSServiceRoleForAmazonEKSLocalOutpost"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "iam:PassRole",
                "iam:ListAttachedRolePolicies"
            ]
            "Resource": "arn:aws:iam::111122223333:role/cluster-role-name"
        },
        {
            "Action": [
                "iam:CreateInstanceProfile",
                "iam:TagInstanceProfile",
                "iam:AddRoleToInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile"
            ],
            "Resource": "arn:aws:iam::*:instance-profile/eks-local-*",
            "Effect": "Allow"
        },
    ]
}
```

Aktualisieren eines Kubernetes-Clusters

AWS-Region

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "eks:UpdateClusterVersion",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-cluster"
        }
    ]
}
```

Auflisten oder Beschreiben aller Cluster

Diese Beispielrichtlinie enthält die benötigten Mindestberechtigungen zum Auflisten und Beschreiben aller Cluster in Ihrem Konto. Ein IAM-Prinzipal muss in der Lage sein, Cluster aufzulisten und zu beschreiben, um den Befehl verwenden zu können. `update-kubeconfig` AWS CLI

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "eks:DescribeCluster",
                "eks:ListClusters"
            ],
            "Resource": "*"
        }
    ]
}
```

Verwendung von serviceverknüpften Rollen für Amazon EKS

Amazon Elastic Kubernetes Service verwendet AWS Identity and Access Management (IAM) service-verknüpfte Rollen. Eine servicegebundene Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EKS verknüpft ist. Servicebezogene Rollen sind von Amazon EKS vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Themen

- Verwenden von Rollen für Amazon-EKS-Cluster
- Verwenden von Rollen für Amazon-EKS-Knotengruppen
- Verwenden von Rollen für Amazon-EKS-Fargate-Profile
- Verbinden eines Kubernetes-Clusters mithilfe von Rollen mit Amazon EKS
- Verwenden von Rollen für lokale Amazon-EKS-Cluster in Outpost

Verwenden von Rollen für Amazon-EKS-Cluster

Amazon Elastic Kubernetes Service verwendet AWS Identity and Access Management (IAM) service-verknüpfte Rollen. Eine servicegebundene Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EKS verknüpft ist. Servicebezogene Rollen sind von Amazon EKS vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon EKS einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon EKS definiert die Berechtigungen seiner mit dem Service verbundenen Rollen, und sofern nicht anders definiert, kann nur Amazon EKS seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Amazon-EKS-Ressourcen, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Amazon EKS

Amazon EKS verwendet die serviceverknüpfte Rolle namens `AWSServiceRoleForAmazonEKS` – Die Rolle ermöglicht Amazon EKS, Cluster in Ihrem Konto zu verwalten. Die angehängten Richtlinien ermöglichen es der Rolle, die folgenden Ressourcen zu verwalten: Netzwerkschnittstellen, Sicherheitsgruppen, Protokolle und VPCs.

> **Note**
>
> Die serviceverknüpfte `AWSServiceRoleForAmazonEKS`-Rolle unterscheidet sich von der Rolle, die für die Clustererstellung erforderlich ist. Weitere Informationen finden Sie unter Amazon-EKS-Cluster-IAM-Rolle.

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonEKS` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `eks.amazonaws.com`

Die Richtlinie für Rollenberechtigungen erlaubt Amazon EKS, die folgenden Aktionen auf den angegebenen Ressourcen durchzuführen:

- `AmazonEKSServiceRolePolicy`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon EKS

Sie müssen eine servicegebundene Rolle nicht manuell erstellen. Wenn Sie einen Cluster in der AWS Management Console, der oder der AWS API erstellen AWS CLI, erstellt Amazon EKS die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen Cluster anlegen, legt Amazon EKS die servicegebundene Rolle wieder für Sie an.

Bearbeiten einer serviceverknüpften Rolle für Amazon EKS

Amazon EKS verhindert die Bearbeitung der serviceverknüpften Rolle `AWSServiceRoleForAmazonEKS`. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer servicegebundenen Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon EKS

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle 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 manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst alle von der Rolle verwendeten Ressourcen löschen.

> **Note**
>
> Wenn der Amazon-EKS-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt der Löschvorgang möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um Amazon EKS-Ressourcen zu löschen, die von der **`AWSServiceRoleForAmazonEKS`**-Rolle verwendet werden.

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wenn Ihr Cluster über Knotengruppen oder Fargate-Profile verfügt, müssen Sie diese löschen, bevor Sie den Cluster löschen können. Weitere Informationen finden Sie unter Löschen einer verwalteten Knotengruppe und Löschen eines Fargate-Profils.
4. Wählen Sie auf der Seite Cluster den Cluster aus, den Sie löschen möchten, und wählen Sie Löschen.
5. Geben Sie den Namen des Clusters im Bestätigungsfenster zum Löschen ein, und wählen Sie dann Bestätigen.
6. Wiederholen Sie diesen Vorgang für alle anderen Cluster in Ihrem Konto. Warten Sie, bis alle Löschoperationen abgeschlossen sind.

Manuelles Löschen der -serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForAmazonEKS` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für Amazon EKS serviceverknüpfte Rollen

Amazon EKS unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter Amazon-EKS-Endpunkte und -Kontingente.

Verwenden von Rollen für Amazon-EKS-Knotengruppen

Amazon EKS verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine servicegebundene Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EKS verknüpft ist. Servicebezogene Rollen sind von Amazon EKS vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon EKS einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon EKS definiert die Berechtigungen seiner mit dem Service verbundenen Rollen, und sofern nicht anders definiert,

kann nur Amazon EKS seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Amazon-EKS-Ressourcen, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Amazon EKS

Amazon EKS verwendet die serviceverknüpfte Rolle namens `AWSServiceRoleForAmazonEKSNodegroup` – Die Rolle ermöglicht Amazon EKS, Knotengruppen in Ihrem Konto zu verwalten. Die angehängten Richtlinien ermöglichen es der Rolle, die folgenden Ressourcen zu verwalten: Auto-Scaling-Gruppen, Sicherheitsgruppen, Startvorlagen und IAM-Instance-Profile.

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonEKSNodegroup` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `eks-nodegroup.amazonaws.com`

Die Richtlinie für Rollenberechtigungen erlaubt Amazon EKS, die folgenden Aktionen auf den angegebenen Ressourcen durchzuführen:

- `AWSServiceRoleForAmazonEKSNodegroup`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon EKS

Sie müssen eine servicegebundene Rolle nicht manuell erstellen. Wenn Sie CreateNodegroup in der AWS Management Console, der oder der AWS CLI AWS API sind, erstellt Amazon EKS die serviceverknüpfte Rolle für Sie.

> **Important**
>
> Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Wenn Sie den Amazon EKS-Service vor dem 1. Januar 2017 genutzt haben, als er begann, serviceverknüpfte Rollen zu unterstützen, hat Amazon EKS die AWSServiceRoleForAmazonEKSNodegroup Rolle in Ihrem Konto erstellt. Weitere Informationen finden Sie unter In meinem IAM-Konto wird eine neue Rolle angezeigt.

Eine serviceverknüpfte Rolle in Amazon EKS (AWS API) erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine verwaltete Knotengruppe in der AWS Management Console, der oder der AWS CLI AWS API erstellen, erstellt Amazon EKS die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine andere verwaltete Knotengruppe erstellen, erstellt Amazon EKS die serviceverknüpfte Rolle erneut für Sie.

Bearbeiten einer serviceverknüpften Rolle für Amazon EKS

Amazon EKS verhindert die Bearbeitung der serviceverknüpften Rolle `AWSServiceRoleForAmazonEKSNodegroup`. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer servicegebundenen Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon EKS

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle 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 manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst alle von der Rolle verwendeten Ressourcen löschen.

> **Note**
>
> Wenn der Amazon-EKS-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt der Löschvorgang möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um Amazon EKS-Ressourcen zu löschen, die von der **`AWSServiceRoleForAmazonEKSNodegroup`**-Rolle verwendet werden.

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.
4. Wählen Sie im Abschnitt Node groups (Knotengruppen) die zu löschende Knotengruppe aus.
5. Geben Sie den Namen der Knotengruppe im Bestätigungsfenster zum Löschen ein und wählen Sie dann Bestätigen.
6. Wiederholen Sie diesen Vorgang für alle anderen Knotengruppen im Cluster. Warten Sie, bis alle Löschoperationen abgeschlossen sind.

Manuelles Löschen der -serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForAmazonEKSNodegroup` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für Amazon EKS serviceverknüpfte Rollen

Amazon EKS unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter Amazon-EKS-Endpunkte und -Kontingente.

Verwenden von Rollen für Amazon-EKS-Fargate-Profile

Amazon EKS verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine servicegebundene Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EKS verknüpft ist. Servicebezogene Rollen sind von Amazon EKS vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon EKS einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon EKS definiert die Berechtigungen seiner mit dem Service verbundenen Rollen, und sofern nicht anders definiert, kann nur Amazon EKS seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Amazon-EKS-Ressourcen, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Amazon EKS

Amazon EKS verwendet die serviceverknüpfte Rolle namens `AWSServiceRoleForAmazonEKSForFargate` – Die Rolle ermöglicht Amazon EKS Fargate, das für Fargate-Pods erforderliche VPC-Netzwerk zu konfigurieren. Die angehängten Richtlinien ermöglichen es der Rolle, Elastic Network Interfaces zu erstellen und zu löschen und Elastic Network Interfaces und Ressourcen zu beschreiben.

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonEKSForFargate` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `eks-fargate.amazonaws.com`

Die Richtlinie für Rollenberechtigungen erlaubt Amazon EKS, die folgenden Aktionen auf den angegebenen Ressourcen durchzuführen:

- `AmazonEKSForFargateServiceRolePolicy`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon EKS

Sie müssen eine servicegebundene Rolle nicht manuell erstellen. Wenn Sie ein Fargate-Profil in der AWS Management Console, der oder der AWS API erstellen AWS CLI, erstellt Amazon EKS die serviceverknüpfte Rolle für Sie.

> **Important**
>
> Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Wenn Sie den Amazon EKS-Service vor dem 13. Dezember 2019 genutzt haben, als er begann, serviceverknüpfte Rollen zu unterstützen, hat Amazon EKS die AWSServiceRoleForAmazonEKSForFargate Rolle in Ihrem Konto erstellt. Weitere Informationen finden Sie unter Eine neue Rolle ist in meinem IAM-Konto erschienen.

Eine serviceverknüpfte Rolle in Amazon EKS (AWS API) erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie ein Fargate-Profil in der AWS Management Console, der oder der AWS API erstellen AWS CLI, erstellt Amazon EKS die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine andere verwaltete Knotengruppe erstellen, erstellt Amazon EKS die serviceverknüpfte Rolle erneut für Sie.

Bearbeiten einer serviceverknüpften Rolle für Amazon EKS

Amazon EKS verhindert die Bearbeitung der serviceverknüpften Rolle `AWSServiceRoleForAmazonEKSForFargate`. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer servicegebundenen Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon EKS

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle 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 manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst alle von der Rolle verwendeten Ressourcen löschen.

> **Note**
>
> Wenn der Amazon-EKS-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt der Löschvorgang möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um Amazon EKS-Ressourcen zu löschen, die von der **`AWSServiceRoleForAmazonEKSForFargate`**-Rolle verwendet werden.

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie auf der Seite Cluster (Cluster) den Cluster aus.
4. Wählen Sie die Registerkarte Compute (Datenverarbeitung) aus.
5. Wenn im Abschnitt Fargate profiles (Fargate-Profile) Fargate-Profile vorhanden sind, wählen Sie jedes einzeln aus und wählen Sie dann Delete (Löschen).
6. Geben Sie den Namen des Profils im Bestätigungsfenster zum Löschen ein, und wählen Sie dann Bestätigen.
7. Wiederholen Sie diesen Vorgang für alle anderen Fargate-Profile im Cluster und für alle anderen Cluster in Ihrem Konto.

Manuelles Löschen der -serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die serviceverknüpfte Rolle zu löschen. AWSServiceRoleForAmazonEKSForFargate Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für Amazon EKS serviceverknüpfte Rollen

Amazon EKS unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter Amazon-EKS-Endpunkte und -Kontingente.

Verbinden eines Kubernetes-Clusters mithilfe von Rollen mit Amazon EKS

Amazon EKS verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine servicegebundene Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EKS verknüpft ist. Servicebezogene Rollen sind von Amazon EKS vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon EKS einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon EKS definiert die Berechtigungen seiner mit dem Service verbundenen Rollen, und sofern nicht anders definiert, kann nur Amazon EKS seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Amazon-EKS-Ressourcen, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Amazon EKS

Amazon EKS verwendet die serviceverknüpfte Rolle namens `AWSServiceRoleForAmazonEKSConnector` – Die Rolle ermöglicht Amazon EKS, Kubernetes-Cluster zu verbinden. Die angehängten Richtlinien ermöglichen es der Rolle, die erforderlichen Ressourcen für die Verbindung mit Ihrem registrierten Kubernetes-Cluster zu verwalten.

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonEKSConnector` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `eks-connector.amazonaws.com`

Die Richtlinie für Rollenberechtigungen erlaubt Amazon EKS, die folgenden Aktionen auf den angegebenen Ressourcen durchzuführen:

- `AmazonEKSConnectorServiceRolePolicy`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon EKS

Sie müssen keine dienstverknüpfte Rolle manuell erstellen, um eine Verbindung mit einem Cluster herzustellen. Wenn Sie einen Cluster in der AWS Management Console, oder der AWS API verbinden AWS CLI`eksctl`, erstellt Amazon EKS die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen Cluster verbinden, legt Amazon EKS die servicegebundene Rolle wieder für Sie an.

Bearbeiten einer serviceverknüpften Rolle für Amazon EKS

Amazon EKS verhindert die Bearbeitung der serviceverknüpften Rolle `AWSServiceRoleForAmazonEKSConnector`. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer servicegebundenen Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon EKS

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle 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 manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst alle von der Rolle verwendeten Ressourcen löschen.

> **Note**
>
> Wenn der Amazon-EKS-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt der Löschvorgang möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um Amazon EKS-Ressourcen zu löschen, die von der **`AWSServiceRoleForAmazonEKSConnector`**-Rolle verwendet werden.

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie auf der Seite Cluster (Cluster) den Cluster aus.
4. Wählen Sie die Registerkarte Abmelden und dann die Registerkarte OK.

Manuelles Löschen der -serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die AWSServiceRoleForAmazonEKSConnector serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Verwenden von Rollen für lokale Amazon-EKS-Cluster in Outpost

Amazon Elastic Kubernetes Service verwendet AWS Identity and Access Management (IAM) service-verknüpfte Rollen. Eine servicegebundene Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EKS verknüpft ist. Servicebezogene Rollen sind von Amazon EKS vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon EKS einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon EKS definiert die Berechtigungen seiner mit dem Service verbundenen Rollen, und sofern nicht anders definiert, kann nur Amazon EKS seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Amazon-EKS-Ressourcen, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Amazon EKS

Amazon EKS verwendet die serviceverknüpfte Rolle namens `AWSServiceRoleForAmazonEKSLocalOutpost` – Die Rolle ermöglicht Amazon EKS, lokale Cluster in Ihrem Konto zu verwalten. Die angehängten Richtlinien ermöglichen es der Rolle, die folgenden Ressourcen zu verwalten: Netzwerkschnittstellen, Sicherheitsgruppen, Protokolle und Amazon-EC2-Instances.

> **Note**
>
> Die serviceverknüpfte `AWSServiceRoleForAmazonEKSLocalOutpost`-Rolle unterscheidet sich von der Rolle, die für die Clustererstellung erforderlich ist. Weitere Informationen finden Sie unter Amazon-EKS-Cluster-IAM-Rolle.

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonEKSLocalOutpost` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `outposts.eks-local.amazonaws.com`

Die Richtlinie für Rollenberechtigungen erlaubt Amazon EKS, die folgenden Aktionen auf den angegebenen Ressourcen durchzuführen:

- `AmazonEKSServiceRolePolicy`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon EKS

Sie müssen eine servicegebundene Rolle nicht manuell erstellen. Wenn Sie einen Cluster in der AWS Management Console, der oder der AWS API erstellen AWS CLI, erstellt Amazon EKS die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen Cluster anlegen, legt Amazon EKS die servicegebundene Rolle wieder für Sie an.

Bearbeiten einer serviceverknüpften Rolle für Amazon EKS

Amazon EKS verhindert die Bearbeitung der serviceverknüpften Rolle `AWSServiceRoleForAmazonEKSLocalOutpost`. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon EKS

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle 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 manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst alle von der Rolle verwendeten Ressourcen löschen.

> **Note**
>
> Wenn der Amazon-EKS-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt der Löschvorgang möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um Amazon EKS-Ressourcen zu löschen, die von der **`AWSServiceRoleForAmazonEKSLocalOutpost`**-Rolle verwendet werden.

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich Clusters (Cluster) für Amazon EKS aus.
3. Wenn Ihr Cluster über Knotengruppen oder Fargate-Profile verfügt, müssen Sie diese löschen, bevor Sie den Cluster löschen können. Weitere Informationen finden Sie unter Löschen einer verwalteten Knotengruppe und Löschen eines Fargate-Profils.
4. Wählen Sie auf der Seite Cluster den Cluster aus, den Sie löschen möchten, und wählen Sie Löschen.

5. Geben Sie den Namen des Clusters im Bestätigungsfenster zum Löschen ein, und wählen Sie dann Bestätigen.
6. Wiederholen Sie diesen Vorgang für alle anderen Cluster in Ihrem Konto. Warten Sie, bis alle Löschoperationen abgeschlossen sind.

Manuelles Löschen der -serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForAmazonEKSLocalOutpost` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für Amazon EKS serviceverknüpfte Rollen

Amazon EKS unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter Amazon-EKS-Endpunkte und -Kontingente.

Amazon-EKS-Cluster-IAM-Rolle

Die Amazon-EKS-Cluster-IAM-Rolle ist für jeden Cluster erforderlich. Von Amazon EKS verwaltete Kubernetes-Cluster verwenden diese Rolle zur Verwaltung von Knoten, und der ältere Cloud-Anbieter verwendet diese Rolle, um Load Balancer mit Elastic Load Balancing für Services zu erstellen.

Bevor Sie Amazon-EKS-Cluster erstellen können, müssen Sie eine IAM-Rolle mit einer der folgenden IAM-Richtlinien erstellen:

- `AmazonEKSClusterPolicy`
- Eine benutzerdefinierten IAM-Richtlinie. Die folgenden Mindestberechtigungen ermöglichen es dem Kubernetes-Cluster, Knoten zu verwalten, erlauben es dem alten Cloud-Anbieter jedoch nicht, Load Balancer mit Elastic Load Balancing zu erstellen. Ihre benutzerdefinierte IAM-Richtlinie muss mindestens folgende Berechtigungen haben:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
```

```
      ],
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "aws:TagKeys": "kubernetes.io/cluster/*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeAvailabilityZones",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> Vor dem 3. Oktober 2023 ClusterPolicy war AmazonEKS für die IAM-Rolle für jeden Cluster erforderlich.
> Vor dem 16. April 2020 waren AmazonEKS ServicePolicy und AmazonEKS erforderlich, und der vorgeschlagene Name für die Rolle ClusterPolicy lautete. `eksServiceRole`
> Mit der `AWSServiceRoleForAmazonEKS` serviceverknüpften Rolle ist die ServicePolicyAmazonEKS-Richtlinie für Cluster, die am oder nach dem 16. April 2020 erstellt wurden, nicht mehr erforderlich.

Überprüfen, ob eine Clusterrolle vorhanden ist

Mit dem folgenden Verfahren können Sie feststellen, ob Ihr Konto bereits über die Amazon-EKS-Clusterrolle verfügt.

So prüfen Sie **`eksClusterRole`** in der IAM-Konsole

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Suchen Sie in der Liste der Rollen nach `eksClusterRole`. Wenn keine Rolle mit `eksClusterRole` vorhanden ist, informieren Sie sich unter Erstellen der Amazon-EKS-Cluster-Rolle, wie Sie die Rolle erstellen können. Wenn eine Rolle mit `eksClusterRole` vorhanden ist, wählen Sie die Rolle aus, um die angefügten Richtlinien anzuzeigen.
4. Wählen Sie Permissions (Berechtigungen).
5. Stellen Sie sicher, dass die von AmazonEks ClusterPolicy verwaltete Richtlinie mit der Rolle verknüpft ist. Wenn die Richtlinie angefügt ist, ist Ihre Amazon-EKS-Clusterrolle korrekt konfiguriert.
6. Wählen Sie Trust Relationships (Vertrauensstellungen) und dann Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.
7. Überprüfen Sie, dass die Vertrauensstellung die folgende Richtlinie enthält. Wenn die Vertrauensstellung mit der folgenden Richtlinie übereinstimmt, wählen Sie Cancel (Abbrechen) aus. Andernfalls kopieren Sie die Richtlinie in das Fenster Edit trust policy (Vertrauensrichtlinie bearbeiten) und wählen Sie Update policy (Richtlinie aktualisieren) aus.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Erstellen der Amazon-EKS-Cluster-Rolle

Sie können das AWS Management Console oder das verwenden AWS CLI , um die Cluster-Rolle zu erstellen.

AWS Management Console

So erstellen Sie Ihre Amazon EKS-Clusterrolle in der IAM-Konsole

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie Roles (Rollen) und anschließend Create Role (Rolle erstellen) aus.
3. Wählen Sie unter Trusted entity type (Typ der vertrauenswürdigen Entität) die Option AWS - Service aus.
4. Wählen Sie in der Dropdown-Liste Use cases for other AWS-Services (Anwendungsfälle für andere) die Option EKS aus.
5. Wählen Sie EKS - Cluster (EKS – Cluster) für Ihren Anwendungsfall und dann Next (Weiter) aus.
6. Wählen Sie auf der Registerkarte Add permissions (Berechtigungen hinzufügen) die Option Next (Weiter) aus.
7. Geben Sie unter Role name (Rollenname) einen eindeutigen Namen für die Rolle ein, z. B. **`eksClusterRole`**.
8. Geben Sie unter Description (Beschreibung) einen beschreibenden Text wie **`Amazon EKS - Cluster role`** ein.
9. Wählen Sie Create role (Rolle erstellen) aus.

AWS CLI

1. Kopieren Sie den folgenden Inhalt in eine Datei namens *`cluster-trust-policy.json`*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Erstellen Sie die -Rolle. Sie können ***eksClusterRole*** mit einem beliebigen Namen ersetzen, den Sie wählen.

```
aws iam create-role \
  --role-name eksClusterRole \
  --assume-role-policy-document file://"cluster-trust-policy.json"
```

3. Fügen Sie der Rolle die erforderliche IAM-Richtlinie an.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy \
  --role-name eksClusterRole
```

Amazon-EKS-Knoten-IAM-Rolle

Der Amazon EKS Node `kubelet` Daemon ruft in Ihrem Namen AWS APIs auf. Knoten erhalten über ein IAM-Instance-Profil und zugehörige Richtlinien Berechtigungen für diese API-Aufrufe. Bevor Sie Knoten starten und in einem Cluster registrieren können, müssen Sie eine IAM-Rolle erstellen, die diese Knoten beim Start verwenden können. Diese Anforderung gilt für Knoten, die mit dem für Amazon EKS optimierten AMI von Amazon oder mit anderen Knoten-AMIs gestartet werden, die Sie verwenden möchten. Darüber hinaus gilt diese Anforderung sowohl für verwaltete Knotengruppen als auch für selbstverwaltete Knoten.

> **Note**
>
> Sie können nicht dieselbe Rolle verwenden, die zum Erstellen von Clustern verwendet wurde.

Bevor Sie Knoten erstellen, müssen Sie eine IAM-Rolle mit folgenden Berechtigungen erstellen:

- Berechtigungen für das `kubelet` zum Beschreiben von Amazon-EC2-Ressourcen in der VPC, wie etwa durch die `AmazonEKSWorkerNodePolicy`-Richtlinie bereitgestellt. Diese Richtlinie stellt auch die Berechtigungen für den Amazon EKS Pod Identity Agent bereit.
- Berechtigungen für das `kubelet` zum Verwenden von Container-Images aus Amazon Elastic Container Registry (Amazon ECR), wie etwa durch die Richtlinie `AmazonEC2ContainerRegistryReadOnly` bereitgestellt. Die Berechtigungen zur Verwendung von Container-Images aus Amazon Elastic Container Registry (Amazon ECR) sind erforderlich, da

die integrierten Add-Ons für Netzwerke Pods ausführen, die Container-Images von Amazon ECR verwenden.

- (Optional) Berechtigungen für den Amazon EKS Pod Identity Agent zum Verwenden der Aktion `eks-auth:AssumeRoleForPodIdentity`, um Anmeldeinformationen für Pods abzurufen. Wenn Sie AmazonEKS nicht verwendenWorkerNodePolicy, müssen Sie diese Berechtigung zusätzlich zu den EC2-Berechtigungen für die Verwendung von EKS Pod Identity bereitstellen.
- (Optional) Wenn Sie IRSA oder EKS Pod Identity nicht verwenden, um Berechtigungen für die VPC-CNI-Pods zu erteilen, müssen Sie Berechtigungen für das VPC CNI in der Instance-Rolle bereitstellen. Sie können entweder die von `AmazonEKS_CNI_Policy` verwaltete Richtlinie verwenden (wenn Sie Ihren Cluster mit der `IPv4`-Produktfamilie erstellt haben) oder aber eine von Ihnen erstellte IPv6-Richtlinie (wenn Sie Ihren Cluster mit der `IPv6`-Produktfamilie erstellt haben). Anstatt die Richtlinie jedoch an diese Rolle anzuhängen, empfehlen wir Ihnen, die Richtlinie an eine separate Rolle anzuhängen, die speziell für das Amazon VPC CNI-Add-on verwendet wird. Weitere Informationen zum Erstellen einer separaten Rolle für das Amazon VPC CNI-Add-on finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

> **Note**
>
> Vor dem 3. Oktober 2023 waren `AmazonEKSWorkerNodePolicy` und `AmazonEC2ContainerRegistryReadOnly` in der IAM-Rolle für jede verwaltete Knotengruppe erforderlich.
> Die Amazon-EC2-Knotengruppen müssen eine andere IAM-Rolle haben als das Fargate-Profil. Weitere Informationen finden Sie unter IAM-Rolle zur Ausführung von Amazon-EKS-Pod.

Nach einer vorhandenen Knotenrolle suchen

Mit dem folgenden Verfahren können Sie feststellen, ob Ihr Konto bereits über die Amazon-EKS-Knotenrolle verfügt.

So prüfen Sie **`eksNodeRole`** in der IAM-Konsole

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.

3. Suchen Sie in der Liste der Rollen nach `eksNodeRole`, `AmazonEKSNodeRole` oder `NodeInstanceRole`. Wenn eine Rolle mit einem dieser Namen nicht existiert, erfahren Sie unter Erstellen der Amazon-EKS-Knoten-IAM-Rolle, wie Sie die Rolle erstellen. Wenn eine Rolle mit `eksNodeRole`, `AmazonEKSNodeRole` oder `NodeInstanceRole` vorhanden ist, wählen Sie die Rolle aus, um die angehängten Richtlinien anzuzeigen.
4. Wählen Sie Permissions (Berechtigungen).
5. Stellen Sie sicher, dass die von AmazonEKS WorkerNodePolicy und AmazonEC2 ContainerRegistryReadOnly verwalteten Richtlinien an die Rolle angehängt sind oder dass eine benutzerdefinierte Richtlinie mit den Mindestberechtigungen angehängt ist.

> **Note**
>
> Wenn die AmazonEKS_CNI_Policy-Richtlinie an die Rolle angehängt ist, empfehlen wir, sie zu entfernen und einer IAM-Rolle anzuhängen, die stattdessen dem `aws-node`-Kubernetes-Servicekonto zugeordnet ist. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

6. Wählen Sie Trust Relationships (Vertrauensstellungen) und dann Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.
7. Überprüfen Sie, dass die Vertrauensstellung die folgende Richtlinie enthält. Wenn die Vertrauensstellung mit der folgenden Richtlinie übereinstimmt, wählen Sie Cancel (Abbrechen) aus. Andernfalls kopieren Sie die Richtlinie in das Fenster Edit trust policy (Vertrauensrichtlinie bearbeiten) und wählen Sie Update policy (Richtlinie aktualisieren) aus.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Erstellen der Amazon-EKS-Knoten-IAM-Rolle

Sie können die Node-IAM-Rolle mit dem oder dem erstellen. AWS Management Console AWS CLI

AWS Management Console

So erstellen Sie Ihre Amazon EKS-Knotenrolle in der IAM-Konsole

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Klicken Sie auf der Seite Roles (Rollen) auf Create role (Rolle erstellen).
4. Gehen Sie auf der Seite Select trusted entity (Vertrauenswürdige Entität auswählen) wie folgt vor:
   a. Wählen Sie im Abschnitt Trusted entity type (Typ der vertrauenswürdigen Entität) die Option AWS service (-Service) aus.
   b. Wählen Sie unter Use case (Anwendungsfall) die Option EC2 aus.
   c. Wählen Sie Next (Weiter).
5. Führen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) die folgenden Schritte aus:
   a. Geben Sie im Feld Filter policies (Filterrichtlinien) **`AmazonEKSWorkerNodePolicy`** ein.
   b. Aktivieren Sie WorkerNodePolicy in den Suchergebnissen das Kontrollkästchen links neben AmazonEks.
   c. Wählen Sie Clear filters (Filter löschen) aus.
   d. Geben Sie im Feld Filter policies (Filterrichtlinien) **`AmazonEC2ContainerRegistryReadOnly`** ein.
   e. Aktivieren Sie in den Suchergebnissen das Kontrollkästchen links neben AmazonEC2 ContainerRegistryReadOnly.

      Die verwaltete Richtlinie AmazonEKS_CNI_Policy oder eine selbst erstellte IPv6-Richtlinie muss ebenfalls an diese Rolle oder an eine andere Rolle angehängt werden, die dem `aws-node`-Servicekonto von Kubernetes zugeordnet ist. Wir empfehlen, die Richtlinie der Rolle zuzuweisen, die dem Kubernetes-Servicekonto zugeordnet ist, anstatt sie dieser Rolle zuzuweisen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

f. Wählen Sie Next (Weiter).

6. Gehen Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) wie folgt vor:

   a. Geben Sie unter Role name (Rollenname) einen eindeutigen Namen für die Rolle ein, z. B. **`AmazonEKSNodeRole`**.

   b. Ersetzen Sie unter Description (Beschreibung) den aktuellen Text durch beschreibenden Text wie beispielsweise **`Amazon EKS - Node role`**.

   c. Fügen Sie der Rolle unter Tags hinzufügen (optional) Metadaten hinzu, indem Sie Tags als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.

   d. Wählen Sie Rolle erstellen aus.

AWS CLI

1. Führen Sie den folgenden Befehl aus, um die Datei `node-role-trust-relationship.json` zu erstellen.

```
cat >node-role-trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

2. Erstellen Sie die IAM-Rolle.

```
aws iam create-role \
  --role-name AmazonEKSNodeRole \
  --assume-role-policy-document file://"node-role-trust-relationship.json"
```

3. Hängen Sie die beiden erforderlichen verwalteten IAM-Richtlinien an die IAM-Rolle an.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name AmazonEKSNodeRole
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name AmazonEKSNodeRole
```

4. Hängen Sie eine der folgenden IAM-Richtlinien an die IAM-Rolle an, je nachdem, mit welcher IP-Familie Sie Ihren Cluster erstellt haben. Die Richtlinie muss an diese Rolle oder an eine Rolle angehängt werden, die dem Kubernetes `aws-node`-Servicekonto zugeordnet ist, das für das Amazon VPC CNI plugin for Kubernetes verwendet wird. Wir empfehlen, die Richtlinie der Rolle zuzuweisen, die dem Kubernetes-Servicekonto zugeordnet ist. Um die Richtlinie der Rolle zuzuweisen, die dem Kubernetes-Servicekonto zugeordnet ist, siehe Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

   - IPv4

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
  --role-name AmazonEKSNodeRole
```

   - IPv6

     1. Kopieren Sie den folgenden Text und speichern Sie ihn in einer Datei mit dem Namen *vpc-cni-ipv6-policy*.json.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssignIpv6Addresses",
                "ec2:DescribeInstances",
                "ec2:DescribeTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeInstanceTypes"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ]
        }
    ]
}
```

2. Erstellen Sie die IAM-Richtlinie.

```
aws iam create-policy --policy-name AmazonEKS_CNI_IPv6_Policy --policy-
document file://vpc-cni-ipv6-policy.json
```

3. Fügen Sie die IAM-Richtlinie an die IAM-Rolle an. Ersetzen Sie *111122223333* durch Ihre Konto-ID.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::111122223333:policy/AmazonEKS_CNI_IPv6_Policy \
  --role-name AmazonEKSNodeRole
```

IAM-Rolle zur Ausführung von Amazon-EKS-Pod

Die Amazon Pod EKS-Ausführungsrolle ist für die Ausführung in Pods der AWS Fargate Infrastruktur erforderlich.

Wenn Ihr Cluster Pods auf einer AWS Fargate Infrastruktur erstellt wird, müssen die Komponenten, die auf der Fargate-Infrastruktur ausgeführt werden, in Ihrem Namen AWS API-Aufrufe tätigen. Auf diese Weise können sie Aktionen wie das Abrufen von Container-Images aus Amazon ECR oder das Weiterleiten von Protokollen an andere AWS Dienste ausführen. Die Amazon-EKS-Pod-Ausführungsrolle stellt die entsprechenden IAM-Berechtigungen bereit.

Wenn Sie ein Fargate-Profil erstellen, müssen Sie eine Pod-Ausführungsrolle für die Amazon-EKS-Komponenten angeben, die auf der Fargate-Infrastruktur mit dem Profil ausgeführt werden. Diese Rolle wird zur rollenbasierten Kubernetes-Zugriffssteuerung (RBAC) des Clusters zur Autorisierung hinzugefügt. Auf diese Weise kann sich das `kubelet`, das in der Fargate-Infrastruktur ausgeführt

wird, bei Ihrem Amazon-EKS-Cluster registrieren, sodass es als Knoten in Ihrem Cluster angezeigt werden kann.

> **Note**
>
> Das Fargate-Profil muss eine andere IAM-Rolle als Amazon-EC2-Knotengruppen haben.

> **Important**
>
> Die im Fargate-Pod ausgeführten Container können nicht die IAM-Berechtigungen annehmen, die einer Pod-Ausführungsrolle zugeordnet sind. Um den Containern in Ihrem Fargate Pod Berechtigungen für den Zugriff auf andere AWS Dienste zu erteilen, müssen Sie verwendenIAM-Rollen für Servicekonten.

Bevor Sie ein Fargate-Profil erstellen, müssen Sie eine IAM-Rolle mit `AmazonEKSFargatePodExecutionRolePolicy` erstellen.

Auf eine korrekt konfigurierte Pod-Ausführungsrolle prüfen

Mit dem folgenden Verfahren können Sie feststellen, ob Ihr Konto bereits über eine korrekt konfigurierte Amazon-EKS-Pod-Ausführungsrolle verfügt. Um Confused-Deputy-Sicherheitsprobleme zu vermeiden, ist es wichtig, dass die Rolle den Zugriff basierend auf `SourceArn` beschränkt. Sie können die Ausführungsrolle nach Bedarf ändern, um Fargate-Profile auch auf anderen Clustern zu unterstützen.

Nach einer Amazon-EKS-Pod-Ausführungsrolle in der IAM-Konsole suchen

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Suchen Sie auf der Seite Rollen in der Rollenliste für AmazonEKS FargatePodExecutionRole. Wenn die Rolle nicht vorhanden ist, finden Sie unter Erstellen der Amazon-EKS-Pod-Ausführungsrolle Informationen zum Erstellen der Rolle. Wenn die Rolle vorhanden ist, wählen Sie sie aus.
4. Gehen Sie auf der FargatePodExecutionRoleAmazonEKS-Seite wie folgt vor:

    a. Wählen Sie Permissions (Berechtigungen).

b. Stellen Sie sicher, dass die von FargatePodExecutionRolePolicyAmazonEKS verwaltete Amazon-Richtlinie mit der Rolle verknüpft ist.

c. Wählen Sie Trust Relationships (Vertrauensbeziehungen) aus.

d. Wählen Sie Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.

5. Prüfen Sie auf der Seite Edit trust policy (Vertrauensrichtlinie bearbeiten), dass die Vertrauensbeziehung die folgende Richtlinie und eine Zeile für Fargate-Profile in Ihrem Cluster enthält. Wenn ja, wählen Sie Cancel (Abbrechen) aus.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Condition": {
         "ArnLike": {
            "aws:SourceArn": "arn:aws:eks:region-
code:111122223333:fargateprofile/my-cluster/*"
         }
      },
      "Principal": {
        "Service": "eks-fargate-pods.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Wenn die Richtlinie übereinstimmt, aber keine Zeile mit den Fargate-Profilen in Ihrem Cluster enthält, können Sie die folgende Zeile oben im `ArnLike`-Objekt hinzufügen. Ersetzen Sie *`region-code`* durch die AWS-Region , in der sich Ihr Cluster befindet, *`111122223333`* durch die ID Ihres Kontos und *`my-cluster`* durch den Namen Ihres Clusters.

```
"aws:SourceArn": "arn:aws:eks:region-code:111122223333:fargateprofile/my-cluster/
*",
```

Wenn die Richtlinie nicht übereinstimmt, kopieren Sie die vollständige vorherige Richtlinie in das Formular und wählen Sie Update policy (Richtlinie aktualisieren) aus. Ersetzen Sie *`region-code`* durch die AWS-Region , in der sich Ihr Cluster befindet. * Ersetzen Sie *`111122223333`*

durch Ihre Konto-ID und *my-cluster* durch den Namen Ihres Clusters. Wenn Sie dieselbe Rolle für alle Cluster in Ihrem Konto verwenden möchten, ersetzen Sie *my-cluster* durch *.

Erstellen der Amazon-EKS-Pod-Ausführungsrolle

Wenn Sie noch nicht über die Amazon Pod EKS-Ausführungsrolle für Ihren Cluster verfügen, können Sie die AWS Management Console oder verwenden, AWS CLI um sie zu erstellen.

AWS Management Console

So erstellen Sie eine AWS Fargate-Pod-Ausführungsrolle mit der AWS Management Console

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Klicken Sie auf der Seite Roles (Rollen) auf Create role (Rolle erstellen).
4. Gehen Sie auf der Seite Select trusted entity (Vertrauenswürdige Entität auswählen) wie folgt vor:
   a. Wählen Sie im Abschnitt Trusted entity type (Typ der vertrauenswürdigen Entität) die Option AWS service (-Service) aus.
   b. Wählen Sie in der Dropdown-Liste Use cases for other AWS-Services (Anwendungsfälle für andere) die Option EKS aus.
   c. Wählen Sie EKS - Fargate-Pod aus.
   d. Wählen Sie Next (Weiter).
5. Wählen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) die Option Next (Weiter) aus.
6. Gehen Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) wie folgt vor:
   a. Geben Sie unter Role name (Rollenname) einen eindeutigen Namen für die Rolle ein, z. B. **AmazonEKSFargatePodExecutionRole**.
   b. Fügen Sie der Rolle unter Tags hinzufügen (optional) Metadaten hinzu, indem Sie Tags als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.
   c. Wählen Sie Rolle erstellen aus.

7. Suchen Sie auf der Seite Rollen in der Rollenliste für AmazonEKS FargatePodExecutionRole. Wählen Sie die Rolle aus.
8. Gehen Sie auf der FargatePodExecutionRoleAmazonEKS-Seite wie folgt vor:
   a. Wählen Sie Trust Relationships (Vertrauensbeziehungen) aus.
   b. Wählen Sie Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.
9. Führen Sie auf der Seite Edit trust policy (Vertrauensrichtlinie bearbeiten) die folgenden Schritte aus:
   a. Kopieren Sie die folgenden Inhalte in das Formular unter Edit trust policy (Vertrauensrichtlinie bearbeiten). Ersetzen Sie den *Regionalcode durch den Regionscode*, in dem AWS-Region sich Ihr Cluster befindet. * Ersetzen Sie *111122223333* durch Ihre Konto-ID und *my-cluster* durch den Namen Ihres Clusters. Wenn Sie dieselbe Rolle für alle Cluster in Ihrem Konto verwenden möchten, ersetzen Sie *my-cluster* durch *.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Condition": {
         "ArnLike": {
            "aws:SourceArn": "arn:aws:eks:region-
code:111122223333:fargateprofile/my-cluster/*"
         }
      },
      "Principal": {
        "Service": "eks-fargate-pods.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

   b. Wählen Sie Update policy.

AWS CLI

Um eine AWS FargatePod Ausführungsrolle zu erstellen mit AWS CLI

1. Kopieren Sie den folgenden Inhalt in eine Datei namens *pod-execution-role-trust-policy*.json. Ersetzen Sie den *Regionscode us-iso-east* durch den Regionscode AWS-Region , in dem sich Ihr Cluster befindet. * Ersetzen Sie *111122223333* durch Ihre Konto-ID und *my-cluster* durch den Namen Ihres Clusters. Wenn Sie dieselbe Rolle für alle Cluster in Ihrem Konto verwenden möchten, ersetzen Sie *my-cluster* durch *.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Condition": {
         "ArnLike": {
            "aws:SourceArn": "arn:aws:eks:region-
code:111122223333:fargateprofile/my-cluster/*"
         }
      },
      "Principal": {
        "Service": "eks-fargate-pods.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Erstellen einer IAM-Pod-Ausführungsrolle

```
aws iam create-role \
  --role-name AmazonEKSFargatePodExecutionRole \
  --assume-role-policy-document file://"pod-execution-role-trust-policy.json"
```

3. Hängen Sie die erforderliche von Amazon EKS verwaltete IAM-Richtlinie an die Rolle an.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSFargatePodExecutionRolePolicy \
  --role-name AmazonEKSFargatePodExecutionRole
```

Amazon-EKS-Konnektor-IAM-Rolle

Sie können Kubernetes Cluster verbinden, um sie in Ihrem anzuzeigen AWS Management Console. Um eine Verbindung zu einem Kubernetes-Cluster herzustellen, erstellen Sie eine IAM-Rolle.

Überprüfen auf eine vorhandene EKS-Connector-Rolle

Mit dem folgenden Verfahren können Sie feststellen, ob Ihr Konto bereits über die Amazon-EKS-Konnektorrolle verfügt.

So prüfen Sie **`AmazonEKSConnectorAgentRole`** in der IAM-Konsole

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Suchen Sie in der Liste der Rollen nach `AmazonEKSConnectorAgentRole`. Wenn keine Rolle mit `AmazonEKSConnectorAgentRole` vorhanden ist, informieren Sie sich unter Erstellen der Rolle des Amazon-EKS-Connector-Agenten, wie Sie die Rolle erstellen können. Wenn eine Rolle mit `AmazonEKSConnectorAgentRole` vorhanden ist, wählen Sie die Rolle aus, um die angefügten Richtlinien anzuzeigen.
4. Wählen Sie Permissions (Berechtigungen).
5. Stellen Sie sicher, dass die von AmazonEks ConnectorAgentPolicy verwaltete Richtlinie mit der Rolle verknüpft ist. Wenn die Richtlinie angefügt ist, ist Ihre Amazon-EKS-Connector-Rolle ordnungsgemäß konfiguriert.
6. Wählen Sie Trust Relationships (Vertrauensstellungen) und dann Edit trust policy (Vertrauensrichtlinie bearbeiten) aus.
7. Überprüfen Sie, dass die Vertrauensstellung die folgende Richtlinie enthält. Wenn die Vertrauensstellung mit der folgenden Richtlinie übereinstimmt, wählen Sie Cancel (Abbrechen) aus. Andernfalls kopieren Sie die Richtlinie in das Fenster Edit trust policy (Vertrauensrichtlinie bearbeiten) und wählen Sie Update policy (Richtlinie aktualisieren) aus.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm.amazonaws.com"
```

```
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Erstellen der Rolle des Amazon-EKS-Connector-Agenten

Sie können das AWS Management Console oder verwenden AWS CloudFormation , um die Connector-Agent-Rolle zu erstellen.

AWS CLI

1. Erstellen Sie eine Datei mit dem Namen `eks-connector-agent-trust-policy.json`, die die folgende JSON enthält, die für die IAM-Rolle verwendet werden soll.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. Erstellen Sie eine Datei mit dem Namen `eks-connector-agent-policy.json`, die die folgende JSON enthält, die für die IAM-Rolle verwendet werden soll.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SsmControlChannel",
            "Effect": "Allow",
```

```
            "Action": [
                "ssmmessages:CreateControlChannel"
            ],
            "Resource": "arn:aws:eks:*:*:cluster/*"
        },
        {
            "Sid": "ssmDataplaneOperations",
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenDataChannel",
                "ssmmessages:OpenControlChannel"
            ],
            "Resource": "*"
        }
    ]
}
```

3. Erstellen Sie die Agentenrolle von Amazon EKS Connector mithilfe der Vertrauensrichtlinie und Richtlinie, die Sie in den vorherigen Listenelementen erstellt haben.

```
aws iam create-role \
     --role-name AmazonEKSConnectorAgentRole \
     --assume-role-policy-document file://eks-connector-agent-trust-policy.json
```

4. Hängen Sie die Richtlinie an Ihre Amazon-EKS-Connector-Agentenrolle an.

```
aws iam put-role-policy \
     --role-name AmazonEKSConnectorAgentRole \
     --policy-name AmazonEKSConnectorAgentPolicy \
     --policy-document file://eks-connector-agent-policy.json
```

AWS CloudFormation

Um Ihre Amazon EKS-Connector-Agent-Rolle mit zu erstellen AWS CloudFormation.

1. Speichern Sie die folgende AWS CloudFormation Vorlage in einer Textdatei auf Ihrem lokalen System.

> **Note**
>
> Diese Vorlage erstellt auch die mit dem Service verknüpfte Rolle, die andernfalls beim Aufruf der `registerCluster`-API erstellt würde. Details dazu finden Sie unter Verbinden eines Kubernetes-Clusters mithilfe von Rollen mit Amazon EKS.

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: 'Provisions necessary resources needed to register clusters in EKS'
Parameters: {}
Resources:
  EKSConnectorSLR:
    Type: AWS::IAM::ServiceLinkedRole
    Properties:
      AWSServiceName: eks-connector.amazonaws.com

  EKSConnectorAgentRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action: [ 'sts:AssumeRole' ]
            Principal:
              Service: 'ssm.amazonaws.com'

  EKSConnectorAgentPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: EKSConnectorAgentPolicy
      Roles:
        - {Ref: 'EKSConnectorAgentRole'}
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: 'Allow'
            Action: [ 'ssmmessages:CreateControlChannel' ]
            Resource:
            - Fn::Sub: 'arn:${AWS::Partition}:eks:*:*:cluster/*'
```

```
          - Effect: 'Allow'
            Action: [ 'ssmmessages:CreateDataChannel',
 'ssmmessages:OpenDataChannel', 'ssmmessages:OpenControlChannel' ]
            Resource: "*"
Outputs:
  EKSConnectorAgentRoleArn:
    Description: The agent role that EKS connector uses to communicate with AWS-
Services.
    Value: !GetAtt EKSConnectorAgentRole.Arn
```

2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Klicken Sie auf Erstellen eines Stacks (entweder mit neuen Ressourcen oder vorhandenen Ressourcen.
4. Wählen Sie für Specify template (Vorlage festlegen) Upload a template file (Vorlagendatei hochladen) aus und wählen Sie dann Choose file (Datei wählen).
5. Wählen Sie die zuvor erstellte Datei und klicken Sie dann auf Next (Weiter).
6. Geben Sie für Stack name (Stack-Name) einen Namen für Ihre Rolle ein, wie z. B. `eksConnectorAgentRole`. Klicken Sie dann auf Next (Weiter).
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
8. Überprüfen Sie auf der Seite Überprüfen Ihre Informationen, bestätigen Sie, dass der Stack IAM-Ressourcen erstellen kann, und wählen Sie dann Erstellen aus.

AWS verwaltete Richtlinien für Amazon Elastic Kubernetes Service

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

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 kundenverwaltete 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.

AWS verwaltete Richtlinie: Amazoneks_CNI_Policy

Sie können die `AmazonEKS_CNI_Policy` an Ihre IAM-Entitäten anhängen. Bevor Sie eine Amazon-EC2-Knotengruppe erstellen, muss diese Richtlinie entweder an die Knoten-IAM-Rolle oder an eine IAM-Rolle angehängt werden, die speziell vom Amazon VPC CNI plugin for Kubernetes verwendet wird. Dies dient dazu, Aktionen in Ihrem Namen auszuführen. Wir empfehlen, dass Sie die Richtlinie an eine Rolle anhängen, die nur vom Plugin verwendet wird. Weitere Informationen finden Sie unter Arbeiten mit dem Amazon VPC CNI plugin for Kubernetes-Amazon-EKS-Add-on und Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

Details zu Berechtigungen

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen:

- **`ec2:*NetworkInterface`**und **`ec2:*PrivateIpAddresses`** — Ermöglicht dem Amazon VPC CNI-Plugin, Aktionen wie die Bereitstellung von Elastic Network Interfaces und IP-Adressen durchzuführen, Pods um Netzwerke für Anwendungen bereitzustellen, die in Amazon EKS ausgeführt werden.
- **`ec2`**Aktionen lesen — Ermöglicht dem Amazon VPC CNI-Plugin, Aktionen wie das Beschreiben von Instances und Subnetzen durchzuführen, um die Anzahl der freien IP-Adressen in Ihren Amazon VPC-Subnetzen zu sehen. Das VPC CNI kann die freien IP-Adressen in jedem Subnetz verwenden, um die Subnetze mit den meisten freien IP-Adressen auszuwählen, die bei der Erstellung einer elastischen Netzwerkschnittstelle verwendet werden sollen.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS_CNI_Policy im Managed Policy Reference Guide. AWS

AWS verwaltete Richtlinie: AmazonEKS ClusterPolicy

Sie können `AmazonEKSClusterPolicy` an Ihre IAM-Entitäten anhängen. Bevor Sie einen Cluster erstellen, müssen Sie über eine Cluster-IAM-Rolle verfügen, der diese Richtlinie beigefügt ist. KubernetesCluster, die von Amazon EKS verwaltet werden, rufen in Ihrem Namen andere AWS Services auf. Sie tun dies, um die Ressourcen zu verwalten, die Sie mit dem Service verwenden.

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen:

- **`autoscaling`** – Lesen und aktualisieren Sie die Konfiguration einer Auto-Scaling-Gruppe. Diese Berechtigungen werden von Amazon EKS nicht verwendet, verbleiben jedoch aus Gründen der Abwärtskompatibilität in der Richtlinie.
- **`ec2`** – Arbeiten Sie mit Volumes und Netzwerkressourcen, die Amazon EC2-Knoten zugeordnet sind. Dies ist erforderlich, damit die Kubernetes-Steuerebene Instances zu einem Cluster verbinden und von persistenten Kubernetes-Volumes angeforderte Amazon-EBS-Volumes dynamisch bereitstellen und verwalten kann.
- **`elasticloadbalancing`** – Arbeiten Sie mit Elastic Load Balancern und fügen Sie ihnen Knoten als Ziele hinzu. Dies ist erforderlich, damit die Kubernetes-Steuerebene von Kubernetes-Diensten angeforderte Elastic Load Balancer dynamisch bereitstellen kann.
- **`iam`** – Erstellen einer serviceverknüpften Rolle. Dies ist erforderlich, damit die Kubernetes-Steuerebene von Kubernetes-Diensten angeforderte Elastic Load Balancer dynamisch bereitstellen kann.
- **`kms`** – Lesen Sie einen Schlüssel von AWS KMS. Dies ist erforderlich, damit die Kubernetes-Steuerebene die Secrets-Verschlüsselung von Kubernetes-Secrets unterstützt, die in `etcd` gespeichert sind.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS ClusterPolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AmazonEKS FargatePodExecutionRolePolicy

Sie können `AmazonEKSFargatePodExecutionRolePolicy` an Ihre IAM-Entitäten anhängen. Bevor Sie ein Fargate-Profil erstellen können, müssen Sie eine Fargate-Pod-Ausführungsrolle erstellen und diese Richtlinie an diese anhängen. Weitere Informationen finden Sie unter Erstellen einer Fargate-Pod-Ausführungsrolle und AWS Fargate Profil.

Diese Richtlinie gewährt der Rolle die Berechtigungen, die den Zugriff auf andere AWS Serviceressourcen ermöglichen, die für die Ausführung von Amazon EKS Pods auf Fargate erforderlich sind.

Details zu Berechtigungen

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen:

- **`ecr`** – Ermöglicht Pods, die auf Fargate ausgeführt werden, Container-Images abzurufen, die in Amazon ECR gespeichert sind.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS FargatePodExecutionRolePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AmazonEKS ForFargateServiceRolePolicy

Sie können `AmazonEKSForFargateServiceRolePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist mit einer servicegebundenen Rolle verknüpft, die es Amazon EKS ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter AWSServiceRoleforAmazonEKSForFargate.

Diese Richtlinie erteilt Amazon EKS die erforderlichen Berechtigungen zum Ausführen von Fargate-Aufgaben. Die Richtlinie wird nur verwendet, wenn Sie über Fargate-Knoten verfügen.

Details zu Berechtigungen

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen.

- **`ec2`** – Erstellen und löschen Sie Elastic Network Interfaces und beschreiben Sie Elastic Network Interfaces und Ressourcen. Dies ist erforderlich, damit der Amazon-EKS-Fargate-Service das für Fargate-Pods erforderliche VPC-Netzwerk konfigurieren kann.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS ForFargateServiceRolePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AmazonEKS ServicePolicy

Sie können `AmazonEKSServicePolicy` an Ihre IAM-Entitäten anhängen. Für Cluster, die vor dem 16. April 2020 erstellt wurden, mussten Sie eine IAM-Rolle erstellen und diese Richtlinie anhängen. Für Cluster, die am oder nach dem 16. April 2020 erstellt wurden, müssen Sie keine Rolle erstellen und diese Richtlinie nicht zuweisen. Wenn Sie einen Cluster mithilfe eines IAM-Prinzipals erstellen, der über die `iam:CreateServiceLinkedRole` entsprechende Berechtigung verfügt, wird die dienstbezogene AWS ServiceRoleforAmazonEKS-Rolle automatisch für Sie erstellt. An die serviceverknüpfte Rolle ist das AWS verwaltete Richtlinie: AmazonEKS ServiceRolePolicy angehängt.

Diese Richtlinie ermöglicht Amazon EKS, die erforderlichen Ressourcen für den Betrieb von Amazon-EKS-Clustern zu erstellen und zu verwalten.

Details zu Berechtigungen

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen.

- **`eks`** – Aktualisieren Sie die Kubernetes-Version Ihres Clusters, nachdem Sie ein Update initiiert haben. Diese Berechtigung wird von Amazon EKS nicht verwendet, verbleibt jedoch aus Gründen der Abwärtskompatibilität in der Richtlinie.
- **`ec2`** – Arbeiten Sie mit Elastic Network Interfaces und anderen Netzwerkressourcen und Tags. Dies wird von Amazon EKS benötigt, um ein Netzwerk zu konfigurieren, das die Kommunikation zwischen Knoten und der Kubernetes-Steuerebene erleichtert.
- **`route53`** – Verknüpfen Sie eine VPC mit einer gehosteten Zone. Dies wird von Amazon EKS benötigt, um ein privates Endpunktnetzwerk für Ihren Kubernetes-Cluster-API-Server zu aktivieren.
- **`logs`** – Protokollereignisse Dies ist erforderlich, damit Amazon EKS die Protokolle der Kubernetes Kontrollebene an versenden kann CloudWatch.
- **`iam`** – Erstellen einer serviceverknüpften Rolle. Dies ist erforderlich, damit Amazon EKS die serviceverknüpfte `AWSServiceRoleForAmazonEKS`-Rolle in Ihrem Namen erstellt.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS ServicePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AmazonEKS ServiceRolePolicy

Sie können `AmazonEKSServiceRolePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist mit einer servicegebundenen Rolle verknüpft, die es Amazon EKS ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon EKS. Wenn Sie einen Cluster mithilfe eines IAM-Prinzipals erstellen, der über die `iam:CreateServiceLinkedRole` entsprechende Berechtigung verfügt, wird die mit dem Dienst verknüpfte AWS ServiceRoleforAmazonEKS-Rolle automatisch für Sie erstellt, und diese Richtlinie wird ihr zugeordnet.

Diese Richtlinie ermöglicht es der serviceverknüpften Rolle, AWS Dienste in Ihrem Namen aufzurufen.

Details zu Berechtigungen

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen.

- **`ec2`** – Erstellen und Beschreiben der Elastic Network Interfaces und Amazon-EC2-Instances, der Cluster-Sicherheitsgruppe und der VPC, die für die Cluster-Erstellung erforderlich sind.
- **`iam`** – Listen Sie alle verwalteten Richtlinien auf, die einer IAM-Rolle zugeordnet sind. Dies ist erforderlich, damit Amazon EKS alle verwalteten Richtlinien und Berechtigungen auflisten und validieren kann, die zum Erstellen von Clustern erforderlich sind.
- Eine VPC mit einer gehosteten Zone verknüpfen – Dies wird von Amazon EKS benötigt, um ein privates Endpunktnetzwerk für Ihren Kubernetes-Cluster-API-Server zu aktivieren.
- Ereignis protokollieren — Dies ist erforderlich, damit Amazon EKS die Protokolle der Kubernetes Kontrollebene an senden kann CloudWatch.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS ServiceRolePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AmazonEksVPC ResourceController

Sie können die `AmazonEKSVPCResourceController`-Richtlinie an Ihre IAM-Identitäten anfügen. Wenn Sie Sicherheitsgruppen für Pods verwenden, müssen Sie diese Richtlinie an Ihr Amazon-EKS-Cluster-IAM-Rolle anhängen, um Aktionen in Ihrem Namen auszuführen.

Diese Richtlinie gewährt der Clusterrolle Berechtigungen zum Verwalten von Elastic Network Interfaces und IP-Adressen für Knoten.

Details zu Berechtigungen

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen:

- **`ec2`** – Verwalten Sie Elastic Network Interfaces und IP-Adressen, um Pod-Sicherheitsgruppen und Windows-Knoten zu unterstützen.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEksVPC ResourceController im Managed Policy Reference Guide. AWS

AWS verwaltete Richtlinie: AmazonEKS WorkerNodePolicy

Sie können die `AmazonEKSWorkerNodePolicy` an Ihre IAM-Entitäten anhängen. Sie müssen diese Richtlinie an eine Knoten-IAM-Rolle anhängen, die Sie angeben, wenn Sie Amazon EC2-Knoten erstellen, die es Amazon EKS ermöglichen, Aktionen in Ihrem Namen auszuführen. Wenn Sie eine Knotengruppe mit `eksctl` erstellen, wird die Knoten-IAM-Rolle erstellt und diese Richtlinie automatisch an die Rolle angehängt.

Diese Richtlinie gewährt Amazon EKS Amazon EC2-Knoten Berechtigungen zum Herstellen einer Verbindung mit Amazon-EKS-Clustern.

Details zu Berechtigungen

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen:

- **`ec2`** – Lesen Sie Informationen zum Instance-Volumen und zum Netzwerk. Dies ist erforderlich, damit Kubernetes-Knoten Informationen zu Amazon-EC2-Ressourcen beschreiben können, die für den Knoten erforderlich sind, um dem Amazon-EKS-Cluster beizutreten.
- **`eks`** – Beschreiben Sie optional den Cluster als Teil des Knoten-Bootstrappings.
- **`eks-auth:AssumeRoleForPodIdentity`** – Erlauben Sie das Abrufen von Anmeldeinformationen für EKS-Workloads auf dem Knoten. Dies ist erforderlich, damit EKS Pod Identity ordnungsgemäß funktioniert.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS WorkerNodePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AWSServiceRoleForAmazonEKSNodegroup

Sie können `AWS ServiceRoleForAmazonEKSNodegroup` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist mit einer servicegebundenen Rolle verknüpft, die es Amazon EKS ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon EKS.

Diese Richtlinie gewährt der `AWS ServiceRoleForAmazonEKSNodegroup`-Rolle Berechtigungen, die es ihr ermöglichen, Amazon EC2-Knotengruppen in Ihrem Konto zu erstellen und zu verwalten.

Details zu Berechtigungen

Diese Richtlinie enthält die folgenden Berechtigungen, die es Amazon EKS ermöglichen, die folgenden Aufgaben auszuführen:

- **`ec2`** – Arbeiten Sie mit Sicherheitsgruppen, Tags und Startvorlagen. Dies ist für von Amazon EKS verwaltete Knotengruppen erforderlich, um die Fernzugriffskonfiguration zu aktivieren. Darüber hinaus erstellen von Amazon EKS verwaltete Knotengruppen in Ihrem Namen eine Startvorlage. Dies dient zum Konfigurieren der Amazon EC2 Auto Scaling-Gruppe, die jede verwaltete Knotengruppe unterstützt.
- **`iam`** – Erstellen einer serviceverknüpften Rolle und Übergeben einer Rolle. Dies ist für von Amazon EKS verwaltete Knotengruppen erforderlich, um Instance-Profile für die Rolle zu verwalten, die beim Erstellen einer verwalteten Knotengruppe übergeben wird. Dieses Instance-Profil wird von Amazon-EC2-Instances verwendet, die als Teil einer verwalteten Knotengruppe gestartet werden. Amazon EKS muss serviceverknüpfte Rollen für andere Services wie Amazon EC2 Auto Scaling-Gruppen erstellen. Diese Berechtigungen werden bei der Erstellung einer verwalteten Knotengruppe verwendet.
- **`autoscaling`** – Arbeiten Sie mit Sicherheitsgruppen für Auto Scaling. Dies ist für von Amazon EKS verwaltete Knotengruppen erforderlich, um die Amazon EC2 Auto Scaling-Gruppe zu verwalten, die jede verwaltete Knotengruppe unterstützt. Es wird auch verwendet, um Funktionen wie das Entfernen von Pods zu unterstützen, wenn Knoten während Knotengruppenaktualisierungen beendet oder recycelt werden.

Die neueste Version des JSON-Richtliniendokuments finden Sie AWSServiceRoleForAmazonEKSNodegroupim AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AmazonEBSCSI DriverPolicy

Die `AmazonEBSCSIDriverPolicy`-Richtlinie ermöglicht es dem Amazon-EBS-Container-Storage-Interface-Treiber (CSI), Volumes in Ihrem Namen zu erstellen, zu ändern, anzuhängen, zu trennen und zu löschen. Es gewährt dem EBS CSI-Treiber auch Berechtigungen zum Erstellen und Löschen von Snapshots sowie zum Auflisten Ihrer Instances, Volumes und Snapshots.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEBSCSI DriverServiceRolePolicy im Managed Policy Reference Guide. AWS

AWS verwaltete Richtlinie: AmazonEFSCSI DriverPolicy

Die `AmazonEFSCSIDriverPolicy`-Richtlinie ermöglicht es dem Amazon EFS Container Storage Interface (CSI), Zugriffspunkte in Ihrem Namen zu erstellen und zu löschen. Es gewährt dem Amazon-EFS-CSI-Treiber außerdem Berechtigungen zum Auflisten Ihrer Zugriffspunkte, Dateisysteme, Mount-Ziele und Amazon-EC2-Verfügbarkeitszonen.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEFSCSI DriverServiceRolePolicy im Managed Policy Reference Guide. AWS

AWS verwaltete Richtlinie: AmazonEKS LocalOutpostClusterPolicy

Sie können diese Richtlinie mit IAM-Entitäten verknüpfen. Bevor Sie einen lokalen Cluster erstellen, müssen Sie diese Richtlinie an Ihre Clusterrolle anhängen. KubernetesCluster, die von Amazon EKS verwaltet werden, rufen in Ihrem Namen andere AWS Services auf. Sie tun dies, um die Ressourcen zu verwalten, die Sie mit dem Service verwenden.

Die `AmazonEKSLocalOutpostClusterPolicy` umfasst folgende Berechtigungen.

- **`ec2`** – Erforderliche Berechtigungen für Amazon-EC2-Instances, um dem Cluster erfolgreich als Steuerebene-Instances beizutreten.
- **`ssm`** – Ermöglicht Amazon EC2 Systems Manager eine Verbindung mit der Instance auf Steuerebene, die von Amazon EKS für die Kommunikation und Verwaltung des lokalen Clusters in Ihrem Konto verwendet wird.
- **`logs`**— Ermöglicht Instances, Logs an Amazon zu übertragen CloudWatch.
- **`secretsmanager`**— Ermöglicht Instances das sichere Abrufen und Löschen von AWS Secrets Manager Bootstrap-Daten für Instances auf der Kontrollebene.
- **`ecr`** – Ermöglicht Pods und Containern, die auf Instances auf der Steuerebene ausgeführt werden, das Abrufen von Container-Images, die in Amazon Elastic Container Registry gespeichert sind.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS LocalOutpostClusterPolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AmazonEKS LocalOutpostServiceRolePolicy

Sie können diese Richtlinie nicht mit Ihren IAM-Entitäten verknüpfen. Wenn Sie einen Cluster mit einem IAM-Prinzipal erstellen, der über die `iam:CreateServiceLinkedRole`-Berechtigung verfügt, erstellt Amazon EKS automatisch die serviceverknüpfte Rolle `AWSServiceRoleforAmazonEKSLocalOutpost` für Sie und fügt diese Richtlinie daran an. Diese Richtlinie ermöglicht es der serviceverknüpften Rolle, in Ihrem Namen AWS Dienste für lokale Cluster aufzurufen.

Die `AmazonEKSLocalOutpostServiceRolePolicy` umfasst folgende Berechtigungen.

- **`ec2`** – Ermöglicht Amazon EKS die Zusammenarbeit mit Sicherheits-, Netzwerk- und anderen Ressourcen, um Instances der Steuerebene in Ihrem Konto erfolgreich zu starten und zu verwalten.
- **`ssm`** – Ermöglicht Amazon EC2 Systems Manager eine Verbindung mit der Instances auf Steuerebene, die von Amazon EKS für die Kommunikation und Verwaltung des lokalen Clusters in Ihrem Konto verwendet wird.
- **`iam`** – Ermöglicht Amazon EKS die Verwaltung des Instance-Profils, das den Instances auf Steuerebene zugeordnet ist.
- **`secretsmanager`**— Ermöglicht Amazon EKS, Bootstrap-Daten für die Instances auf der Kontrollebene zu speichern, AWS Secrets Manager sodass sie beim Instance-Bootstrapping sicher referenziert werden können.
- **`outposts`** – Ermöglicht Amazon EKS, Outpost-Informationen von Ihrem Konto abzurufen, um einen lokalen Cluster in einem Outpost erfolgreich zu starten.

Die neueste Version des JSON-Richtliniendokuments finden Sie unter AmazonEKS LocalOutpostServiceRolePolicy im AWS Managed Policy Reference Guide.

Amazon EKS-Updates für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon EKS an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen. Wenn Sie automatisch über

Änderungen an dieser Seite benachrichtigt werden möchten, abonnieren Sie den RSS-Feed auf der Dokumentverlaufsseite von Amazon EKS.

Änderung	Beschreibung	Datum
Amazoneks_CNI_Policy — Aktualisierung einer bestehenden Richtlinie	Amazon EKS hat neue `ec2:DescribeSubnets` Berechtigungen hinzugefügt, damit Amazon VPC CNI plugin for Kubernetes sie die Anzahl der freien IP-Adressen in Ihren Amazon VPC-Subnetzen sehen können. Das VPC CNI kann die freien IP-Adressen in jedem Subnetz verwenden, um die Subnetze mit den meisten freien IP-Adressen auszuwählen, die bei der Erstellung einer elastischen Netzwerkschnittstelle verwendet werden sollen.	4. März 2024
AmazonEKS WorkerNodePolicy — Aktualisierung einer bestehenden Richtlinie	Amazon EKS wurden neue Berechtigungen hinzugefügt, die EKS-Pod-Identitäten zulassen. Der Amazon EKS Pod Identity-Agent verwendet die Knotenrolle.	26. November 2023
DriverPolicyAmazonEFSCSI eingeführt.	AWS führte das ein. `AmazonEFSCSIDriverPolicy`	26. Juli 2023
Berechtigungen zu AmazonEks ClusterPolicy hinzugefügt.	Die `ec2:DescribeAvailabilityZones` -Berechtigung wurde hinzugefügt, damit Amazon EKS die AZ-Details während der automatischen Subnetzerkennung beim Erstellen von Load Balancern abrufen kann.	07. Februar 2023

Änderung	Beschreibung	Datum
Die Richtlinienbedingungen in DriverPolicyAmazonEBSCSI wurden aktualisiert.	Ungültige Richtlinienbedingungen mit Platzhalterzeichen im `StringLike` -Schlüsselfeld wurden entfernt. Außerdem wurde eine neue Bedingung `ec2:ResourceTag/kubernetes.io/created-for/pvc/name: "*"` zu `ec2:DeleteVolume` hinzugefügt, die es dem EBS-CSI-Treiber ermöglicht, vom In-Tree-Plugin erstellte Volumes zu löschen.	17. November 2022
Berechtigungen zu AmazonEks LocalOutpostServiceRolePolicy hinzugefügt.	`ec2:DescribeVPCAttribute` , `ec2:GetConsoleOutput` und `ec2:DescribeSecret` wurden hinzugefügt, um eine bessere Validierung der Voraussetzungen und eine bessere Kontrolle des Lebenszyklus zu ermöglichen. Außerdem wurden `ec2:DescribePlacementGroups` und `"arn:aws:ec2:*:*:placement-group/*"` zu `ec2:RunInstances` hinzugefügt, um die Platzierungskontrolle der Steuerebene von Amazon-EC2-Instances auf Outposts zu unterstützen.	24. Oktober 2022

Änderung	Beschreibung	Datum
Aktualisieren Sie die Amazon Elastic Container Registry-Berechtigungen in AmazonEKS LocalOutpostClusterPolicy.	Die Aktion `ecr:GetDownloadUrlForLayer` wurde von allen Ressourcenabschnitten in einen Bereich mit begrenztem Umfang verschoben. Ressource `arn:aws:ecr:*:*:repository/eks/*` wurde hinzugefügt. Entfernen Sie die Ressource `arn:aws:ecr:*:*:repository/eks/eks-certificates-controller-public` . Diese Ressource wird durch die hinzugefügte `arn:aws:ecr:*:*:repository/eks/*` - Ressource abgedeckt.	20. Oktober 2022
Berechtigungen zu AmazonEks LocalOutpostClusterPolicy hinzugefügt.	Das `arn:aws:ecr:*:*:repository/kubelet-config-updater` Repository der Amazon-Elastic-Container-Registry wurde hinzugefügt, damit die Cluster-Steuerebenen-Instances einige `kubelet`-Argumente aktualisieren können.	31. August 2022
Einführung von AmazonEks LocalOutpostClusterPolicy.	AWS führte das ein. `AmazonEKSLocalOutpostClusterPolicy`	24. August 2022
Einführung von AmazonEks LocalOutpostServiceRolePolicy.	AWS führte das ein. `AmazonEKSLocalOutpostServiceRolePolicy`	23. August 2022
Einführung von AmazonEBS CSI DriverPolicy.	AWS führte das ein. `AmazonEBSCSIDriverPolicy`	4. April 2022

Änderung	Beschreibung	Datum
Berechtigungen zu AmazonEks WorkerNodePolicy hinzugefügt.	`ec2:DescribeInstanceTypes` hinzugefügt, um Amazon-EKS-optimierte AMIs zu ermöglichen, die Eigenschaften auf Ebene von Instances automatisch erkennen können.	21. März 2022
Es wurden Berechtigungen für hinzugefügt. AWSServiceRoleForAmazonEKSNodegroup	`autoscaling:EnableMetricsCollection` -Berechtigung hinzugefügt, um Amazon EKS die Aktivierung der Kennzahlenerfassung zu ermöglichen.	13. Dezember 2021
Berechtigungen zu AmazonEks ClusterPolicy hinzugefügt.	`ec2:DescribeAccountAttributes` -, `ec2:DescribeAddresses` - und `ec2:DescribeInternetGateways` -Berechtigungen hinzugefügt, damit Amazon EKS eine serviceverknüpfte Rolle für einen Network Load Balancer erstellen kann.	17. Juni 2021
Amazon EKS hat mit der Nachverfolgung von Änderungen begonnen.	Amazon EKS hat damit begonnen, Änderungen für seine AWS verwalteten Richtlinien nachzuverfolgen.	17. Juni 2021

Fehlersuche bei IAM

In diesem Thema werden einige häufige Fehler bei der Verwendung von Amazon EKS mit IAM sowie deren Lösung behandelt.

AccessDeniedException

Wenn Sie `AccessDeniedException` beim Aufrufen eines AWS API-Vorgangs eine Meldung erhalten, verfügen die von Ihnen verwendeten IAM-Prinzipalanmeldedaten nicht über die erforderlichen Berechtigungen, um diesen Aufruf durchzuführen.

```
An error occurred (AccessDeniedException) when calling the DescribeCluster operation:
User: arn:aws:iam::111122223333:user/user_name is not authorized to perform:
```

```
eks:DescribeCluster on resource: arn:aws:eks:region:111122223333:cluster/my-cluster
```

In der vorigen Beispielmeldung hat der Benutzer keine Berechtigung zum Aufruf der API-Operation `DescribeCluster` von Amazon EKS. Um einem IAM-Prinzipal Amazon-EKS-Administratorrechte zu erteilen, lesen Sie Beispiele für identitätsbasierte Amazon-EKS-Richtlinien.

Weitere allgemeine Informationen zu IAM finden Sie unterSteuern des Zugriffs mit RichtlinienimIAM User Guideaus.

Sie können keine Nodes (Knoten) auf der Registerkarte Compute (Datenverarbeitung) oder irgendetwas auf der Registerkarte Resources (Ressourcen) sehen und in der AWS Management Console wird ein Fehler angezeigt

Möglicherweise sehen Sie eine Konsolenfehlermeldung, die folgendermaßen lautet: `Your current user or role does not have access to Kubernetes objects on this EKS cluster`. Stellen Sie sicher, dass der IAM-Prinzipalbenutzer, den Sie AWS Management Console mit verwenden, über die erforderlichen Berechtigungen verfügt. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

Die aws-auth-`ConfigMap` gewährt keinen Zugriff auf den Cluster

Der AWS IAM-Authenticator lässt keinen Pfad in dem Rollen-ARN zu, der in der `ConfigMap` verwendet wird. Entfernen Sie daher den Pfad, bevor Sie `rolearn` angeben. Sie können beispielsweise `arn:aws:iam::111122223333:role/team/developers/eks-admin` in `arn:aws:iam::111122223333:role/eks-admin` ändern.

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 EKS 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 Service.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon EKS 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 EKS-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. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffssteuerungslisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Amazon EKS diese Features unterstützt, finden Sie unter Funktionsweise von Amazon EKS mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Pod-Container erhalten folgenden Fehler: **`An error occurred (SignatureDoesNotMatch) when calling the GetCallerIdentity operation: Credential should be scoped to a valid region`**

Ihre Container erhalten diesen Fehler, wenn Ihre Anwendung explizit Anfragen an den AWS STS globalen Endpunkt (`https://sts.amazonaws`) sendet und Ihr Kubernetes Dienstkonto für die Verwendung eines regionalen Endpunkts konfiguriert ist. Sie können das Problem mit einer der folgenden Optionen beheben:

- Aktualisieren Sie Ihren Anwendungscode, um explizite Aufrufe an den AWS STS globalen Endpunkt zu entfernen.
- Aktualisieren Sie Ihren Anwendungscode, um explizite Anrufe an regionale Endpunkte wie `https://sts.us-west-2.amazonaws.com` zu senden. Ihre Anwendung sollte über Redundanz verfügen, um bei Ausfall des Service in der AWS-Region eine andere AWS-Region auszuwählen. Weitere Informationen finden Sie unter Verwalten von AWS STS in einer AWS-Region im IAM-Benutzerhandbuch.
- Konfigurieren Sie Ihre Servicekonten so, dass sie den globalen Endpunkt verwenden. Alle Versionen vor `1.22` verwenden standardmäßig den globalen Endpunkt, aber Cluster ab der Version `1.22` verwenden standardmäßig den regionalen Endpunkt. Weitere Informationen finden Sie unter Den AWS Security Token Service Endpunkt für ein Dienstkonto konfigurieren.

Standardmäßig erstellte Kubernetes-Amazon-EKS-Rollen und -Benutzer

Wenn Sie einen Kubernetes-Cluster erstellen, werden für das ordnungsgemäße Funktionieren von Kubernetes mehrere Kubernetes Standardidentitäten auf diesem Cluster erstellt. Amazon EKS erstellt Kubernetes-Identitäten für jede seiner Standardkomponenten. Die Identitäten stellen Kubernetes rollenbasierte Autorisierungssteuerung (RBAC) für die Cluster-Komponenten bereit. Weitere Informationen finden Sie unter Using RBAC authorization in der Kubernetes-Dokumentation.

Wenn Sie optionale Add-ons zu Ihrem Cluster installieren, werden Ihrem Cluster möglicherweise zusätzliche Kubernetes-Identitäten hinzugefügt. Weitere Informationen zu Identitäten, die in diesem Thema nicht behandelt werden, finden Sie in der Dokumentation des Add-Ons.

Sie können die Liste der von Amazon EKS erstellten Kubernetes-Identitäten auf Ihrem Cluster mit dem AWS Management Console - oder `kubectl`-Befehlszeilentool anzeigen. Alle Benutzeridentitäten erscheinen in den `kube` Audit-Logs, die Ihnen über Amazon CloudWatch zur Verfügung stehen.

AWS Management Console

Voraussetzung

Der IAM-Prinzipal, den Sie verwenden, muss über die in Erforderliche Berechtigungen beschriebenen Berechtigungen verfügen.

Um von Amazon EKS erstellte Identitäten anzuzeigen, verwenden Sie den AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie in der Liste Clusters (Cluster) den Cluster aus, der die anzuzeigenden Identitäten enthält.
3. Wählen Sie die Registerkarte Resources (Ressourcen) aus.
4. Wählen Sie unter Resource types (Ressourcentypen) die Option Authorization (Autorisierung) aus.
5. Wählen Sie ClusterRoles, ClusterRoleBindings, Rollen oder RoleBindings. Alle Ressourcen, denen eks vorangestellt ist, wurden von Amazon EKS erstellt. Weitere von Amazon EKS erstellte Identitätsressourcen sind:

   - Der ClusterRoleund ClusterRoleBindingbenannte aws-node. Die aws-node-Ressourcen unterstützen das Amazon VPC CNI plugin for Kubernetes, das Amazon EKS auf allen Clustern installiert.
   - Ein ClusterRolebenannter vpc-resource-controller-roleund ein benannter. ClusterRoleBindingvpc-resource-controller-rolebinding Diese Ressourcen unterstützen den Amazon-VPC-Ressourcencontroller, den Amazon EKS auf allen Clustern installiert.

   Zusätzlich zu den Ressourcen, die Sie in der Konsole sehen, sind die folgenden speziellen Benutzeridentitäten in Ihrem Cluster vorhanden, obwohl sie in der Konfiguration des Clusters nicht sichtbar sind:

   - **`eks:cluster-bootstrap`** – Wird für `kubectl`-Operationen während des Cluster-Bootstraps verwendet.
   - **`eks:support-engineer`** – Wird für Cluster-Management-Operationen verwendet.
6. Wählen Sie eine bestimmte Ressource aus, um Details dazu anzuzeigen. Standardmäßig werden Ihnen Informationen in der Structured view (strukturierten Ansicht) angezeigt. In der

oberen rechten Ecke der Detailseite können Sie die Raw view (Rohansicht) auswählen, um alle Informationen für die Ressource anzuzeigen.

Kubectl

Voraussetzung

Die Entität, mit der Sie AWS Identity and Access Management ((IAM) oder OpenID Connect (OIDC)) die Kubernetes Ressourcen im Cluster auflisten, muss von IAM oder Ihrem OIDC Identitätsanbieter authentifiziert werden. Der Entität müssen Berechtigungen zur Verwendung der Verben Kubernetes `get` und `list` für die `Role`-, `ClusterRole`-, `RoleBinding`-, and `ClusterRoleBinding`-Ressourcen in Ihrem Cluster gewährt werden, mit denen die Entität arbeiten soll. Weitere Informationen zum Gewähren von Zugriff auf IAM-Entitäten auf Ihren Cluster finden Sie unter the section called “Gewähren Sie Zugriff auf Kubernetes-APIs”. Weitere Informationen zum Gewähren von Zugriff auf Ihren Cluster durch Entitäten, die von Ihrem eigenen OIDC-Anbieter authentifiziert wurden, finden Sie unterr Authentifizieren Sie Benutzer für Ihren Cluster von einem OpenID Connect Identitätsanbieter.

So zeigen Sie von Amazon EKS erstellte Identitäten mit dem **`kubectl`** an

Führen Sie den Region aus, die Sie anzeigen möchten. Alle zurückgegebenen Ressourcen, denen eks vorangestellt ist, wurden von Amazon EKS erstellt. Zusätzlich zu den Ressourcen, die in der Ausgabe der Befehle zurückgegeben werden, sind die folgenden speziellen Benutzeridentitäten in Ihrem Cluster vorhanden, obwohl sie in der Konfiguration des Clusters nicht sichtbar sind:

- **`eks:cluster-bootstrap`** – Wird für `kubectl`-Operationen während des Cluster-Bootstraps verwendet.
- **`eks:support-engineer`** – Wird für Cluster-Management-Operationen verwendet.

ClusterRoles— `ClusterRoles` sind auf Ihren Cluster beschränkt, sodass alle einer Rolle erteilten Berechtigungen für Ressourcen in jedem Kubernetes Namespace des Clusters gelten.

Der folgende Befehl gibt alle von Amazon EKS erstellten Kubernetes `ClusterRoles` auf Ihrem Cluster zurück.

```
kubectl get clusterroles | grep eks
```

Zusätzlich zu dem in der Ausgabe zurückgegebenen `ClusterRoles`, dem etwas vorangestellt ist, sind die folgenden `ClusterRoles` vorhanden.

- **`aws-node`** – Diese `ClusterRole` unterstützt die Amazon VPC CNI plugin for Kubernetes, die Amazon EKS auf allen Clustern installiert.
- **`vpc-resource-controller-role`** – Diese `ClusterRole` unterstützt den Amazon-VPC-Ressourcencontroller, den Amazon EKS auf allen Clustern installiert.

Um die Spezifikation für ein `ClusterRole` anzuzeigen, ersetzen Sie *`eks:k8s-metrics`* im folgenden Befehl durch eine `ClusterRole`, die in der Ausgabe des vorherigen Befehls zurückgegeben wird. Das folgende Beispiel gibt die Spezifikation für die *`eks:k8s-metrics`* `ClusterRole` zurück.

```
kubectl describe clusterrole eks:k8s-metrics
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:         eks:k8s-metrics
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources         Non-Resource URLs  Resource Names  Verbs
  ---------         -----------------  --------------  -----
                    [/metrics]         []              [get]
  endpoints         []                 []              [list]
  nodes             []                 []              [list]
  pods              []                 []              [list]
  deployments.apps  []                 []              [list]
```

ClusterRoleBindings— `ClusterRoleBindings` sind auf Ihren Cluster beschränkt.

Der folgende Befehl gibt alle von Amazon EKS erstellten Kubernetes `ClusterRoleBindings` auf Ihrem Cluster zurück.

```
kubectl get clusterrolebindings | grep eks
```

Zusätzlich zu dem in der Ausgabe zurückgegebenen `ClusterRoleBindings`, sind die folgenden `ClusterRoleBindings` vorhanden.

- **aws-node** – Diese `ClusterRoleBinding` unterstützt die Amazon VPC CNI plugin for Kubernetes, die Amazon EKS auf allen Clustern installiert.
- **vpc-resource-controller-rolebinding** – Diese `ClusterRoleBinding` unterstützt den Amazon-VPC-Ressourcencontroller, den Amazon EKS auf allen Clustern installiert.

Um die Spezifikation für ein `ClusterRoleBinding` anzuzeigen, ersetzen Sie *`eks:k8s-metrics`* im folgenden Befehl durch eine `ClusterRoleBinding`, die in der Ausgabe des vorherigen Befehls zurückgegeben wird. Das folgende Beispiel gibt die Spezifikation für die *`eks:k8s-metrics`* `ClusterRoleBinding` zurück.

```
kubectl describe clusterrolebinding eks:k8s-metrics
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:         eks:k8s-metrics
Labels:       <none>
Annotations:  <none>
Role:
  Kind:  ClusterRole
  Name:  eks:k8s-metrics
Subjects:
  Kind  Name             Namespace
  ----  ----             ---------
  User  eks:k8s-metrics
```

Rollen – `Roles` sind auf einen Kubernetes-Namespace beschränkt. Alle mit `Roles` erstellten Amazon EKS sind auf den `kube-system`-Namespace beschränkt.

Der folgende Befehl gibt alle von Amazon EKS erstellten Kubernetes `Roles` auf Ihrem Cluster zurück.

```
kubectl get roles -n kube-system | grep eks
```

Um die Spezifikation für ein `Role` anzuzeigen, ersetzen Sie *`eks:k8s-metrics`* im folgenden Befehl durch den Namen eines `Role`, das in der Ausgabe des vorherigen Befehls zurückgegeben wird. Das folgende Beispiel gibt die Spezifikation für die *`eks:k8s-metrics`* `Role` zurück.

```
kubectl describe role eks:k8s-metrics -n kube-system
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:         eks:k8s-metrics
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources         Non-Resource URLs  Resource Names             Verbs
  ---------         -----------------  --------------             -----
  daemonsets.apps   []                 [aws-node]                 [get]
  deployments.apps  []                 [vpc-resource-controller]  [get]
```

RoleBindings— `RoleBindings` sind auf einen Namespace beschränkt. Kubernetes Alle mit `RoleBindings` erstellten Amazon EKS sind auf den `kube-system`-Namespace beschränkt.

Der folgende Befehl gibt alle von Amazon EKS erstellten Kubernetes `RoleBindings` auf Ihrem Cluster zurück.

```
kubectl get rolebindings -n kube-system | grep eks
```

Um die Spezifikation für ein `RoleBinding` anzuzeigen, ersetzen Sie *`eks:k8s-metrics`* im folgenden Befehl durch eine `RoleBinding`, die in der Ausgabe des vorherigen Befehls zurückgegeben wird. Das folgende Beispiel gibt die Spezifikation für die *`eks:k8s-metrics`* `RoleBinding` zurück.

```
kubectl describe rolebinding eks:k8s-metrics -n kube-system
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:         eks:k8s-metrics
Labels:       <none>
Annotations:  <none>
Role:
  Kind:  Role
  Name:  eks:k8s-metrics
Subjects:
  Kind  Name             Namespace
  ----  ----             ---------
  User  eks:k8s-metrics
```

Compliance-Validierung für Amazon Elastic Kubernetes Service

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- Architecting for HIPAA Security and Compliance on Amazon Web Services — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

  > **Note**
  >
  > AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der Referenz für HIPAA-berechtigte Services.

- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.

- AWS Security Hub— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Ausfallsicherheit bei Amazon EKS

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit geringer Latenz, hohem Durchsatz und hoher Redundanz 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.

Die Kubernetes-Steuerebene wird von Amazon EKS über mehrere AWS-Availability-Zones hinweg ausgeführt und skaliert, um eine hohe Verfügbarkeit zu gewährleisten. Amazon EKS skaliert Steuerebene-Instances automatisch basierend auf der Last, erkennt und ersetzt fehlerhafte Steuerebene-Instances und patcht die Steuerebene automatisch. Nachdem Sie ein Versionsupdate eingeleitet haben, aktualisiert Amazon EKS Ihre Steuerebene für Sie und behält die hohe Verfügbarkeit der Steuerebene während des Updates bei.

Diese Steuerebene besteht aus mindestens zwei API-Server-Instances und drei `etcd`-Instances, die über drei Availability Zones innerhalb einer AWS-Region ausgeführt werden. Amazon EKS

- überwacht aktiv die Last auf Steuerebenen-Instances und skaliert sie automatisch, um eine hohe Leistung sicherzustellen;

- Erkennt und ersetzt automatisch fehlerhafte Instances der Steuerebene und startet sie bei Bedarf in den Availability Zones innerhalb der AWS-Region neu.
- Nutzt die Architektur von AWS-Regionen, um eine hohe Verfügbarkeit zu gewährleisten. Deshalb kann Amazon EKS ein SLA für die Verfügbarkeit von API-Server-Endpunkten anbieten.

Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter AWSGlobale Infrastruktur.

Infrastruktursicherheit in Amazon EKS

Als verwalteter Service ist Amazon Elastic Kubernetes Service durch die AWS globale Netzwerksicherheit von geschützt. Informationen zu AWS Sicherheitsservices und wie die Infrastruktur AWS schützt, finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung mit den bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastrukturschutz in Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon EKS 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.

Wenn Sie einen Amazon-EKS-Cluster erstellen, geben Sie die VPC-Subnetze an, die Ihr Cluster verwenden soll. Amazon EKS erfordert Subnetze in mindestens zwei Availability Zones. Wir empfehlen eine VPC mit öffentlichen und privaten Subnetzen, damit Kubernetes öffentliche Load Balancer in den öffentlichen Subnetzen erstellen kann, die den Datenverkehr auf Pods ausgleichen, die auf Knoten in privaten Subnetzen laufen.

Weitere Informationen zu Überlegungen bezüglich der VPC finden Sie unter Amazon EKS: VPC- und Subnetz-Anforderungen und -Überlegungen.

Wenn Sie Ihre VPC und Knotengruppen mit den in der Erste Schritte mit Amazon EKS Anleitung bereitgestellten AWS CloudFormation Vorlagen erstellen, werden Ihre Steuerebene und Knotensicherheitsgruppen mit unseren empfohlenen Einstellungen konfiguriert.

Weitere Informationen zu Überlegungen bezüglich Sicherheitsgruppen finden Sie unter Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe.

Wenn Sie einen neuen Cluster erstellen, erstellt Amazon EKS einen Endpunkt für den verwalteten Kubernetes-API-Server, über den Sie mit Ihrem Cluster kommunizieren (mit Kubernetes-Verwaltungswerkzeugen wie `kubectl`). Standardmäßig ist dieser API-Server-Endpunkt öffentlich im Internet zugänglich und der Zugriff auf den API-Server wird durch eine Kombination aus AWS Identity and Access Management (IAM) und nativer Kubernetesrollenbasierter Zugriffskontrolle (RBAC) gesichert.

Sie können den privaten Zugriff auf den Kubernetes-API-Server aktivieren, sodass die gesamte Kommunikation zwischen Ihren Knoten und dem API-Server innerhalb Ihrer VPC bleibt. Sie können die IP-Adressen einschränken, die über das Internet auf Ihren API-Server zugreifen können, oder den Internetzugriff auf den API-Server vollständig deaktivieren.

Weitere Informationen zum Ändern des Zugriffs auf Cluster-Endpunke finden Sie unter Ändern des Cluster-Endpunktzugriffs.

Sie können Kubernetes-Netzwerkrichtlinien mit Amazon VPC CNI oder Tools von Drittanbietern wie Project Calico implementieren. Weitere Informationen zur Verwendung von Amazon VPC CNI für Netzwerkrichtlinien finden Sie unter Konfigurieren Ihres Clusters für Kubernetes-Netzwerkrichtlinien. Project Calico ist ein Open-Source-Projekt eines Drittanbieters. Weitere Informationen finden Sie in der Dokumentation zu Project Calico.

Konfigurations- und Schwachstellenanalyse in Amazon EKS

Sicherheit ist ein wichtiger Aspekt bei der Konfiguration und Wartung von Kubernetes-Clustern und -Anwendungen. Im Folgenden sind Ressourcen aufgeführt, mit denen Sie die Sicherheitskonfiguration Ihrer EKS-Cluster analysieren können, Ressourcen, mit denen Sie nach Sicherheitslücken suchen können, und Integrationen mit AWS Diensten, die diese Analyse für Sie durchführen können.

Der Benchmark des Center for Internet Security (CIS) für Amazon EKS

Der Center for Internet Security (CIS) Kubernetes Benchmark bietet Anleitungen für Amazon EKS-Sicherheitskonfigurationen. Die Benchmark:

- Gilt für Amazon-EC2-Knoten (sowohl verwaltete als auch selbstverwaltete), bei denen Sie für die Sicherheitskonfigurationen von Kubernetes-Komponenten verantwortlich sind.
- Bietet eine von der Community genehmigte Standardmethode, um sicherzustellen, dass Sie Ihren Kubernetes-Cluster und Ihre Knoten sicher konfiguriert haben, wenn Sie Amazon EKS verwenden.
- Besteht aus vier Abschnitten; Protokollierungskonfiguration der Steuerebene, Knotensicherheitskonfigurationen, Richtlinien und verwaltete Services.
- Unterstützt alle Kubernetes-Versionen, die derzeit in Amazon EKS verfügbar sind und können mit kube-bench, ein Standard-Open-Source-Tool zur Überprüfung der Konfiguration mit dem CIS-Benchmark auf Kubernetes-Clustern.

Weitere Informationen hierzu finden Sie unterEinführung des CIS Amazon EKS Benchmarkaus.

Amazon-EKS-Plattformversionen

Die Versionen der Amazon EKS-Plattform stellen die Funktionen der Cluster-Steuerebene dar, einschließlich der aktivierten Kubernetes API-Serverflags und der aktuellen Kubernetes Patch-Version. Neue Cluster werden mit der neuesten Plattformversion bereitgestellt. Details hierzu finden Sie unter Amazon-EKS-Plattformversionen.

Sie können einen Amazon-EKS-Cluster auf neuere Kubernetes-Versionen aktualisieren. Wenn neue Kubernetes-Versionen in Amazon EKS verfügbar werden, empfehlen wir Ihnen, Ihre Cluster proaktiv auf die neueste verfügbare Version zu aktualisieren. Weitere Informationen zu Kubernetes-Versionen in EKS finden Sie unter Kubernetes-Versionen für Amazon EKS.

Liste der Sicherheitslücken im Betriebssystem

Liste der Sicherheitslücken in AL2023

Verfolgen Sie Sicherheits- oder Datenschutzereignisse für Amazon Linux 2023 im Amazon Linux Security Center oder abonnieren Sie den zugehörigen RSS-Feed. Sicherheits- und Datenschutzereignisse enthalten eine Übersicht über das Problem sowie Pakete und Anweisungen zum Aktualisieren Ihrer Instances, um das Problem zu beheben.

Liste der Sicherheitslücken in Amazon Linux 2

Verfolgen Sie Sicherheits- oder Datenschutzereignisse für Amazon Linux 2 im Amazon Linux Security Center oder abonnieren Sie den zugehörigen RSS-Feed. Sicherheits- und Datenschutzereignisse

enthalten eine Übersicht über das Problem sowie Pakete und Anweisungen zum Aktualisieren Ihrer Instances, um das Problem zu beheben.

Knotenerkennung mit Amazon Inspector

Mit Amazon Inspector können Sie eine Prüfung auf nicht gewünschte Netzwerkzugänglichkeit Ihrer Arbeitsknoten und auf Schwachstellen auf diesen Amazon-EC2-Instances vornehmen.

Cluster- und Knotenerkennung mit Amazon GuardDuty

Der GuardDuty Bedrohungserkennungsservice von Amazon, der Ihnen hilft, Ihre Konten, Container, Workloads und die Daten in Ihrer AWS Umgebung zu schützen. GuardDutyBietet unter anderem die folgenden beiden Funktionen zur Erkennung potenzieller Bedrohungen für Ihre EKS-Cluster: EKS-Schutz und Runtime Monitoring.

Weitere Informationen finden Sie unter Erkennen Sie Bedrohungen mit Amazon GuardDuty.

Bewährte Methoden für die Sicherheit in Amazon EKS

Bewährte Methoden für die Amazon-EKS-Sicherheit werden auf Github verwaltet: https://aws.github.io/aws-eks-best-practices/security/docs/

Pod-Sicherheitsrichtlinie

Der Zulassungscontroller für Kubernetes-Pod-Sicherheitsrichtlinien überprüft Pod-Erstellung und Update-Anfragen anhand einer Reihe von Regeln. Standardmäßig werden Amazon-EKS-Cluster mit einer vollständig permissiven Sicherheitsrichtlinie ohne Einschränkungen bereitgestellt. Weitere Informationen finden Sie unter Pod Security Policies in der Kubernetes-Dokumentation.

> **Note**
>
> `PodSecurityPolicy` (PSP) war in der Kubernetes-Version `1.21` veraltet und wurde in Kubernetes `1.25` entfernt. PSPs werden durch Pod Security Admission (PSA) ersetzt, einen integrierten Zulassungscontroller, die die in den Pod Security Standards (PSS) beschriebenen Sicherheitskontrollen implementiert. PSA und PSS haben beide den Beta-Featurestatus erreicht und sind in Amazon EKS standardmäßig aktiviert. Um das Entfernen von PSP in `1.25` anzugehen, empfehlen wir Ihnen, PSS in Amazon EKS zu implementieren.

Weitere Informationen finden Sie unter Implementieren von Pod-Sicherheitsstandards in Amazon EKS im AWS-Blog.

Amazon-EKS-Pod-Standardsicherheitsrichtlinie

Amazon-EKS-Cluster mit Kubernetes-Version `1.13` oder höher verfügen über eine standardmäßige Pod-Sicherheitsrichtlinie mit dem Namen `eks.privileged`. Diese Richtlinie weist keine Einschränkungen hinsichtlich der Art der vom System akzeptierten Pod auf. Dies entspricht der Ausführung von Kubernetes mit deaktiviertem `PodSecurityPolicy`-Controller.

> **Note**
>
> Diese Richtlinie wurde erstellt, um die Rückwärtskompatibilität mit Clustern aufrechtzuerhalten, für die der `PodSecurityPolicy`-Controller nicht aktiviert war. Sie können restriktivere Richtlinien für Ihren Cluster und für einzelne Namespaces und Service-Konten erstellen und dann die Standardrichtlinie löschen, um die restriktiveren Richtlinien zu aktivieren.

Sie können die Standardrichtlinie mit dem folgenden Befehl anzeigen.

```
kubectl get psp eks.privileged
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME              PRIV    CAPS    SELINUX    RUNASUSER    FSGROUP    SUPGROUP
 READONLYROOTFS   VOLUMES
eks.privileged    true    *       RunAsAny   RunAsAny     RunAsAny   RunAsAny    false
      *
```

Um weitere Details anzugeben, können Sie die Richtlinie mit dem folgenden Befehl beschreiben.

```
kubectl describe psp eks.privileged
```

Eine Beispielausgabe sieht wie folgt aus.

```
Name:  eks.privileged
```

```
Settings:
  Allow Privileged:                       true
  Allow Privilege Escalation:             0xc0004ce5f8
  Default Add Capabilities:               <none>
  Required Drop Capabilities:             <none>
  Allowed Capabilities:                   *
  Allowed Volume Types:                   *
  Allow Host Network:                     true
  Allow Host Ports:                       0-65535
  Allow Host PID:                         true
  Allow Host IPC:                         true
  Read Only Root Filesystem:              false
  SELinux Context Strategy: RunAsAny
    User:                                 <none>
    Role:                                 <none>
    Type:                                 <none>
    Level:                                <none>
  Run As User Strategy: RunAsAny
    Ranges:                               <none>
  FSGroup Strategy: RunAsAny
    Ranges:                               <none>
  Supplemental Groups Strategy: RunAsAny
    Ranges:                               <none>
```

Sie können die vollständige YAML-Datei für die `eks.privileged`-Pod-Sicherheitsrichtlinie, ihre Clusterrolle und die Clusterrollenbindung in [Installieren oder Wiederherstellen der standardmäßigen Pod-Sicherheitsrichtlinie] anzeigen.

Löschen der standardmäßigen Amazon-EKS-Pod-Sicherheitsrichtlinie

Wenn Sie restriktivere Richtlinien für Ihre Pods erstellen, können Sie anschließend die standardmäßige `eks.privileged`-Pod-Sicherheitsrichtlinie von Amazon-EKS löschen, um Ihre benutzerdefinierten Richtlinien zu aktivieren.

> **Important**
>
> Wenn Sie Version `1.7.0` oder höher des CNI-Plugins verwenden und dem `aws-node`–KubernetesServicekonto, das für die im Daemonset bereitgestellten `aws-node`-Pods verwendet wird, eine benutzerdefinierte Pod-Sicherheitsrichtlinie zuweisen, muss die Richtlinie `NET_ADMIN` im Abschnitt `allowedCapabilities` zusammen mit `hostNetwork: true` und `privileged: true` in den `spec` der Richtlinie enthalten.

So löschen Sie die Pod-Standardsicherheitsrichtlinie

1. Erstellen Sie eine Datei namens *privileged-podsecuritypolicy.yaml* mit dem Inhalt in der Beispieldatei in Installieren oder Wiederherstellen der standardmäßigen Pod-Sicherheitsrichtlinie.
2. Löschen Sie die YAML-Datei mit dem folgenden Befehl. Dadurch werden die standardmäßige Pod-Sicherheitsrichtlinie, das `ClusterRole` und das damit verknüpfte `ClusterRoleBinding` gelöscht.

```
kubectl delete -f privileged-podsecuritypolicy.yaml
```

Installieren oder Wiederherstellen der standardmäßigen Pod-Sicherheitsrichtlinie

Wenn Sie ein Upgrade von einer früheren Version von Kubernetes durchführen oder die standardmäßige Amazon-EKS-`eks.privileged`-Pod-Sicherheitsrichtlinie geändert oder gelöscht haben, können Sie sie mit den folgenden Schritten wiederherstellen.

So installieren Sie die Standard-Pod-Sicherheitsrichtlinie oder stellen sie wieder her

1. Erstellen Sie eine Datei mit dem Namen *privileged-podsecuritypolicy.yaml* und den folgenden Inhalten.

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: eks.privileged
  annotations:
    kubernetes.io/description: 'privileged allows full unrestricted access to
      Pod features, as if the PodSecurityPolicy controller was not enabled.'
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
  labels:
    kubernetes.io/cluster-service: "true"
    eks.amazonaws.com/component: pod-security-policy
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
```

```
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
  readOnlyRootFilesystem: false

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: eks:podsecuritypolicy:privileged
  labels:
    kubernetes.io/cluster-service: "true"
    eks.amazonaws.com/component: pod-security-policy
rules:
- apiGroups:
  - policy
  resourceNames:
  - eks.privileged
  resources:
  - podsecuritypolicies
  verbs:
  - use

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: eks:podsecuritypolicy:authenticated
  annotations:
    kubernetes.io/description: 'Allow all authenticated users to create privileged
 Pods.'
```

```
  labels:
    kubernetes.io/cluster-service: "true"
    eks.amazonaws.com/component: pod-security-policy
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: eks:podsecuritypolicy:privileged
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: system:authenticated
```

2. Wenden Sie die YAML-Datei mit dem folgenden Befehl an.

```
kubectl apply -f privileged-podsecuritypolicy.yaml
```

Entfernen der Pod-Sicherheitsrichtlinie (PSP) – Häufig gestellte Fragen

Die `PodSecurityPolicy` war in Kubernetes`1.21` veraltet und wurde aus Kubernetes`1.25` entfernt. Wenn Sie PodSecurityPolicy in Ihrem Cluster verwenden, müssen Sie zu den integrierten Kubernetes Pod-Sicherheitsstandards (PSS) oder zu einer policy-as-code Lösung migrieren, bevor Sie Ihren Cluster auf Version aktualisieren**`1.25`**, um Unterbrechungen Ihrer Workloads zu vermeiden. Wählen Sie eine häufig gestellte Frage aus, um mehr zu erfahren.

Was ist ein PSP?

PodSecurityPolicy ist ein integrierter Zulassungscontroller, mit dem ein Cluster-Administrator sicherheitsrelevante Aspekte der Pod Spezifikation steuern kann. Wenn ein Pod die Anforderungen seiner PSP erfüllt, wird der Pod wie gewohnt in den Cluster aufgenommen. Wenn ein Pod die PSP-Anforderungen nicht erfüllt, wird der Pod abgelehnt und kann nicht ausgeführt werden.

Ist das Entfernen der PSP speziell für Amazon EKS oder wird sie im Kubernetes-Upstream entfernt?

Dies ist eine Upstream-Änderung im Kubernetes-Projekt und keine Änderung in Amazon EKS. PSP war in Kubernetes `1.21` veraltet und wurde in Kubernetes `1.25` entfernt. Die Kubernetes-Community hat schwerwiegende Probleme mit der Benutzerfreundlichkeit von PSP identifiziert. Dazu gehörten

die versehentliche Erteilung umfassenderer Genehmigungen als beabsichtigt und Schwierigkeiten bei der Überprüfung, welche PSPs in einer bestimmten Situation anwenden. Diese Probleme konnten nicht behoben werden, ohne grundlegende Änderungen vorzunehmen. Dies ist der Hauptgrund, warum die Kubernetes-Community beschlossen hat, PSP zu entfernen.

Wie kann ich überprüfen, ob ich die PSPs in meinen Clustern von Amazon EKS verwende?

Zur Überprüfung, ob Sie die PSPs in Ihrem Cluster verwenden, könnten Sie den folgenden Befehl ausführen:

```
kubectl get psp
```

Führen Sie den folgenden Befehl aus, um die Pods anzuzeigen, auf die sich die PSPs in Ihrem Cluster auswirken. Dieser Befehl gibt den Pod-Namen, den Namespace und die PSPs aus:

```
kubectl get pod -A -o jsonpath='{range.items[?(@.metadata.annotations.kubernetes
\.io/psp)]}{.metadata.name}{"\t"}{.metadata.namespace}{"\t"}
{.metadata.annotations.kubernetes\.io/psp}{"\n"}'
```

Was kann ich tun, wenn ich die PSPs in meinem Amazon EKS-Cluster verwende?

Bevor Sie Ihren Cluster auf `1.25` aktualisieren, müssen Sie Ihre PSPs auf eine der folgenden Alternativen migrieren:

- Kubernetes PSS.
- P-olicy-as-code Lösungen aus der Kubernetes Umgebung.

Als Reaktion auf die PSP-Einstellung und die anhaltende Notwendigkeit, die Pod-Sicherheit von Anfang an zu kontrollieren, hat die Kubernetes-Community eine integrierte Lösung mit (PSS) und Pod Security Admission (PSA) entwickelt. Der PSA-Webhook implementiert die Steuerelemente, die in der PSS definiert sind.

Die bewährten Methoden für die Migration der PSPs zur integrierten Version der PSS finden Sie im Leitfaden zu bewährten Methoden für Amazon EKS.. Wir empfehlen außerdem, unseren Block zur Implementation von Pod-Sicherheitsstandards in Amazon EKS zu lesen. Weitere Referenzen

umfassen die Migration von PodSecurityPolicy zum integrierten PodSecurity Zulassungscontroller und die Zuordnung PodSecurityPolicies zu Pod-Sicherheitsstandards.

P-olicy-as-code Lösungen bieten Integritätsschutz als Leitfaden für Cluster-Benutzer und verhindern unerwünschte Verhaltensweisen durch vorgeschriebene automatisierte Kontrollen. P-olicy-as-code Lösungen verwenden in der Regel Kubernetes Dynamic Admission Controllers, um den Anforderungsfluss des Kubernetes API-Servers mithilfe eines Webhook-Aufrufs abzufangen. P-olicy-as-code Lösungen mutieren und validieren Anforderungsnutzlasten basierend auf Richtlinien, die als Code geschrieben und gespeichert werden.

Für sind mehrere Open-Source- policy-as-code Lösungen verfügbarKubernetes. Bewährte Methoden für die Migration PSPs zu einer policy-as-code Lösung finden Sie im Abschnitt Policy-as-code der Pod-Sicherheitsseite auf GitHub.

Ich sehe einen PSP-Aufruf von **eks.privileged** in meinem Cluster. Was ist das und was kann ich dagegen tun?

Amazon-EKS-Cluster mit Kubernetes-Version `1.13` oder höher verfügen über eine standardmäßige PSP mit dem Namen `eks.privileged`. Diese Richtlinie wurde in `1.24` und früheren Clustern erstellt. Sie wird nicht in `1.25` und späteren Clustern verwendet. Amazon EKS migriert diese PSP automatisch zu einer PSS-basierten Durchsetzung. Von Ihrer Seite aus ist keine Aktion erforderlich.

Nimmt Amazon EKS Änderungen an PSPs vor, die bereits in meinem bestehenden Cluster vorhanden sind, wenn ich meinen Cluster auf Version **1.25** aktualisiere?

Nein. Neben `eks.privileged`, eine von Amazon EKS erstellte PSP, werden beim Upgrade keine Änderungen an anderen PSPs in Ihrem Cluster vorgenommen, wenn Sie auf `1.25` aktualisieren.

Verhindert Amazon EKS ein Cluster-Update auf Version **1.25**, wenn ich noch nicht von der PSP migriert habe?

Nein. Amazon EKS verhindert ein Cluster-Update auf die Version `1.25` nicht, wenn Sie noch nicht von der PSP migriert haben.

Was passiert, wenn ich vergesse, mein PSPs zu PSS/PSA oder zu einer policy-as-code Lösung zu migrieren, bevor ich meinen Cluster auf Version aktualisiere**1.25**? Kann ich nach der Aktualisierung meines Clusters migrieren?

Wenn ein Cluster mit einer PSP auf Kubernetes-Version `1.25` aktualisiert wird, erkennt der API-Server die PSP-Ressource in `1.25` nicht. Dies kann dazu führen, dass Pods falsche

Sicherheitsbereiche erhalten. Eine vollständige Liste der Auswirkungen finden Sie unter Migrieren von PodSecurityPolicy zum integrierten PodSecurity Zulassungscontroller .

Wie wirkt sich diese Änderung auf die Pod-Sicherheit für Windows-Workloads aus?

Wir erwarten keine spezifischen Auswirkungen auf Windows-Workloads. PodSecurityContext hat ein Feld namens `windowsOptions` in der `PodSpec v1` API für Windows Pods. Dies verwendet PSS in Kubernetes `1.25`. Weitere Informationen und bewährte Methoden zur Durchsetzung PSS von Windows-Workloads finden Sie im Leitfaden zu bewährten Methoden für Amazon EKS und in der Kubernetes-Dokumentation.

Verwenden von AWS Secrets Manager-Secrets mit Kubernetes

Um Secrets aus „Secrets Manager“ und Parameter aus „Parameter Store“ als gemountete Dateien in Amazon EKS Pods anzuzeigen, können Sie den AWS-Secrets and Configuration Provider (ASCP) für die Kubernetes-Secrets-Store-CSI-Treiber verwenden.

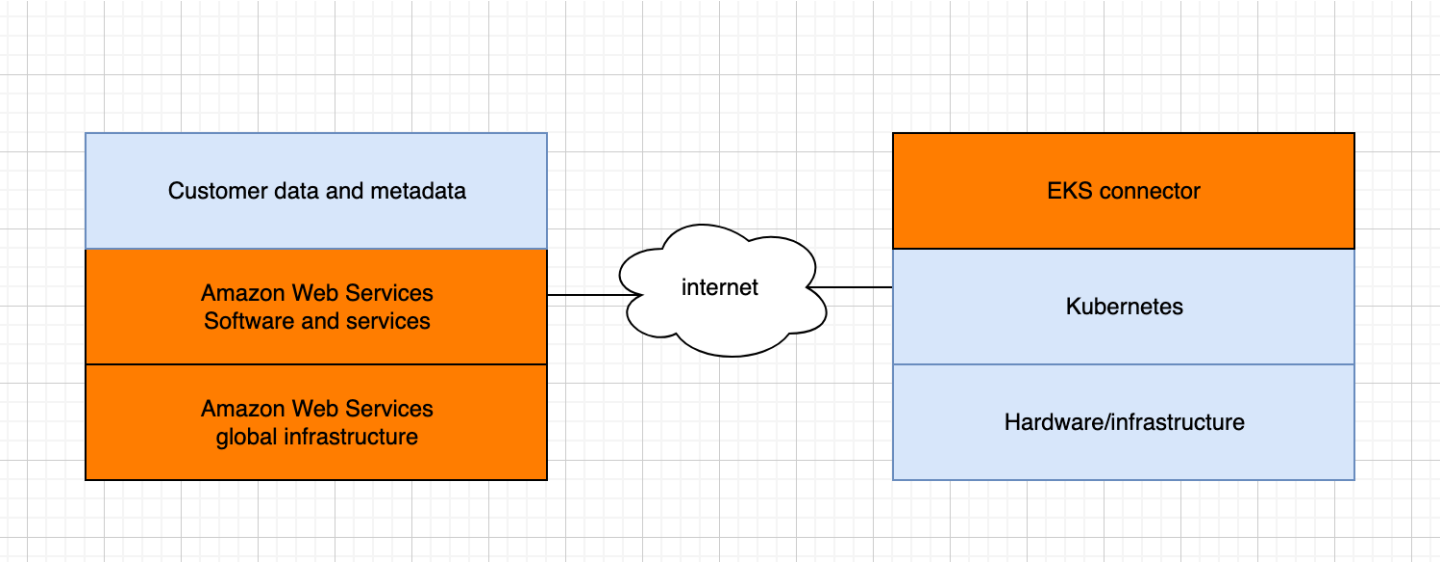
Mit dem ASCP können Sie Ihre Secrets in Secrets Manager speichern und verwalten und diese dann über Ihre auf Amazon EKS ausgeführten Workloads abrufen. Sie können IAM-Rollen und -Richtlinien verwenden, um den Zugriff auf Ihre Secrets auf bestimmte Kubernetes Pods in einem Cluster zu beschränken. Der ASCP ruft die Pod-Identität ab und tauscht die Identität gegen eine IAM-Rolle. Der ASCP übernimmt die IAM-Rolle des Pod und kann dann Secrets aus Secrets Manager abrufen, die für diese Rolle autorisiert sind.

Wenn Sie in Secrets Manager die automatische Rotation für Ihre Secrets verwenden, können Sie auch die Secrets-Store-CSI-Treiberrotation verwenden, um sicherzustellen, dass Sie das neueste Secret aus Secrets Manager abrufen.

Weitere Informationen finden Sie unter Verwenden von Secrets-Manager-Secrets in Amazon EKS im AWS-Secrets Manager-Benutzerhandbuch.

Überlegungen zum Amazon EKS Connector

Der Amazon EKS Connector ist eine Open-Source-Komponente, die auf Ihrem Kubernetes-Cluster ausgeführt wird. Dieser Cluster kann sich außerhalb der AWS-Umgebung befinden. Dies erfordert zusätzliche Überlegungen zur Sicherheitsverantwortung. Das folgende Diagramm verdeutlicht diese Konfiguration. Orange steht für Pflichten von AWS, Blau für Pflichten des Kunden:



In diesem Thema werden die Unterschiede im Verantwortungsmodell beschrieben, wenn sich der verbundene Cluster außerhalb von AWS befindet.

Pflichten von AWS

- Wartung, Erstellung und Bereitstellung von Amazon EKS Connector, eine Open-Source-Komponente, die auf dem Kubernetes-Cluster eines Kunden läuft und mit AWS kommuniziert.
- Aufrechterhaltung der Kommunikationssicherheit auf Transport- und Anwendungsebene zwischen dem verbundenen Kubernetes-Cluster und den AWS-Services.

Pflichten des Kunden

- Kubernetes-Cluster-spezifische Sicherheit, insbesondere in folgenden Bereichen:
  - Kubernetes-Secrets müssen ordnungsgemäß verschlüsselt und geschützt sein.
  - Sperren des Zugriffs auf den `eks-connector`-Namespace.
- Konfigurieren der RBAC-Berechtigungen (rollenbasierte Zugriffskontrolle) zur Verwaltung des Benutzerzugriffs von IAM-Prinzipal aus AWS. Detaillierte Anweisungen finden Sie unter Gewähren des Zugriffs für einen IAM-Prinzipal zum Anzeigen von Kubernetes-Ressource eines Clusters.
- Amazon EKS Connector installieren und aktualisieren.
- Wartung der Hardware, Software und Infrastruktur, die den verbundenen Kubernetes-Cluster unterstützt.

- Sichern ihrer AWS-Konten (z. B. durch Verwendung von sicheren Anmeldeinformationen des Root-Benutzers).

Anzeigen der Kubernetes-Ressourcen

Mit der AWS Management Console können Sie die Kubernetes-Ressourcen anzeigen, die in Ihrem Cluster bereitgestellt sind. Sie können Kubernetes Ressourcen nicht mit dem AWS CLI oder anzeigen `eksctl`. Um Kubernetes-Ressourcen mit einem Befehlszeilentool anzuzeigen, verwenden Sie `kubectl`.

Voraussetzung

Um die Registerkarte Ressourcen und den Abschnitt Knoten auf der Registerkarte Compute in der anzeigen zu können AWS Management Console, muss der von Ihnen verwendete IAM-Prinzipal über bestimmte IAM-Rechte und Kubernetes Berechtigungen verfügen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

Um Kubernetes Ressourcen mit dem anzuzeigen AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie in der Liste Cluster den Cluster aus, der die Kubernetes-Ressourcen enthält, die Sie anzeigen möchten.
3. Wählen Sie die Registerkarte für Resources (Ressourcen).
4. Wählen Sie eine Resource type-Gruppe (Ressourcentyp) aus, für die Sie Ressourcen anzeigen möchten, z. B. Workloads. Ihnen wird eine Liste der Ressourcentypen in dieser Gruppe angezeigt.
5. Wählen Sie einen Ressourcentyp aus, z. B. Deployments (Bereitstellungen) in der Gruppe Workloads. Sie sehen eine Beschreibung des Ressourcentyps, einen Link zur Kubernetes-Dokumentation für weitere Informationen über den Ressourcentyp und eine Liste der Ressourcen dieses Typs, die in Ihrem Cluster bereitgestellt werden. Wenn die Liste leer ist, werden keine Ressourcen dieses Typs in Ihrem Cluster bereitgestellt.
6. Wählen Sie eine Ressource aus, um weitere Informationen dazu anzuzeigen. Probieren Sie die folgenden Beispiele aus:

   - Wählen Sie die Gruppe Workloads aus, dann den Ressourcentyp Deployments (Bereitstellungen) gefolgt von der Ressource coredns. Wenn Sie eine Ressource auswählen, befinden Sie sich standardmäßig in der strukturierten Ansicht. Für einige Ressourcentypen wird in der strukturierten Ansicht der Abschnitt Pods angezeigt. In diesem Abschnitt sind die Pods aufgeführt, die von der Workload verwaltet werden. Sie können jedes beliebige Pod

auswählen, um Informationen über das Pod anzuzeigen. Nicht für alle Ressourcentypen werden Informationen in der strukturierten Ansicht angezeigt. Wenn Sie die Ansicht Raw in der oberen rechten Ecke der Seite für die Ressource auswählen, sehen Sie die vollständige JSON-Antwort von der Kubernetes-API für die Ressource.

- Wählen Sie die Cluster-Gruppe aus und dann den Ressourcentyp Nodes (Knoten). Eine Liste aller Knoten in Ihrem Cluster wird angezeigt. Die Knoten können von jedem Amazon-EKS-Knoten-Typ sein. Das ist die gleiche Liste, die Sie im Abschnitt Nodes (Knoten) sehen, wenn Sie die Registerkarte Compute (Datenverarbeitung) für Ihren Cluster auswählen. Wählen Sie eine Knotenressource aus der Liste aus. In der strukturierten Ansicht sehen Sie ebenfalls einen Abschnitt Pods. In diesem Abschnitt finden Sie alle Pods, die auf dem Knoten ausgeführt werden.

Erforderliche Berechtigungen

Um die Registerkarte Ressourcen und den Abschnitt Knoten auf der Registerkarte Compute in der anzuzeigen AWS Management Console, muss der IAM-Principal, den Sie verwenden, über bestimmte Mindest-IAM-Werte und Kubernetes Berechtigungen verfügen. Führen Sie die folgenden Schritte aus, um Ihren IAM-Prinzipalen die erforderlichen Berechtigungen zuzuweisen.

1. Stellen Sie sicher, dass `eks:AccessKubernetesApi` und andere erforderliche IAM-Berechtigungen zum Anzeigen von Kubernetes-Ressourcen dem von Ihnen verwendeten IAM-Prinzipal zugewiesen sind. Weitere Informationen zum Bearbeiten von Berechtigungen für einen IAM-Prinzipal finden Sie unter Zugriff für Prinzipale steuern und im IAM-Benutzerhandbuch. Weitere Informationen zum Bearbeiten von Rollenberechtigungen finden Sie unter Modifying a role permissions policy (console) (Ändern einer Rollenberechtigungsrichtlinie (Konsole)) im IAM-Benutzerhandbuch.

   Die folgende Beispielrichtlinie enthält die erforderlichen Berechtigungen für einen Prinzipal zum Anzeigen von Kubernetes-Ressourcen für alle Cluster in Ihrem Konto. Ersetzen Sie *111122223333* durch Ihre AWS -Konto-ID.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "eks:ListFargateProfiles",
```

```
                "eks:DescribeNodegroup",
                "eks:ListNodegroups",
                "eks:ListUpdates",
                "eks:AccessKubernetesApi",
                "eks:ListAddons",
                "eks:DescribeCluster",
                "eks:DescribeAddonVersions",
                "eks:ListClusters",
                "eks:ListIdentityProviderConfigs",
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:GetParameter",
            "Resource": "arn:aws:ssm:*:111122223333:parameter/*"
        }
    ]
}
```

Um Knoten in verbundenen Clustern anzuzeigen, sollte die IAM-Rolle des Amazon-EKS-Connectors in der Lage sein, den Prinzipal im Cluster anzunehmen. Dadurch kann der Amazon-EKS-Anschluss den Prinzipal einem Kubernetes-Benutzer zuordnen.

2. Erstellen Sie ein `clusterrolebinding`-Kubernetes oder `rolebinding`, das an ein Kubernetes-`role` oder `clusterrole` gebunden ist, das über die erforderlichen Berechtigungen zum Anzeigen der Kubernetes-Ressourcen verfügt. Informationen zu Rollen und Rollenbindungen in Kubernetes finden Sie unter Using RBAC Authorization (Verwenden der RBAC-Autorisierung) in der Kubernetes-Dokumentation. Sie können eines der folgenden Manifeste auf Ihren Cluster anwenden, die eine `role` und eine `rolebinding` oder eine `clusterrole` und eine `clusterrolebinding` mit den erforderlichen Kubernetes-Berechtigungen erstellen:

   Kubernetes-Ressourcen in allen Namespaces anzeigen

   Der Gruppenname in der Datei lautet `eks-console-dashboard-full-access-group`. Wenden Sie das Manifest mit dem folgenden Befehl auf Ihren Cluster an:

   ```
   kubectl apply -f https://s3.us-west-2.amazonaws.com/amazon-eks/docs/eks-console-full-access.yaml
   ```

Kubernetes-Ressourcen in einem bestimmten Namespace anzeigen

Der Namespace in dieser Datei ist `default`. Der Gruppenname in der Datei lautet `eks-console-dashboard-restricted-access-group`. Wenden Sie das Manifest mit dem folgenden Befehl auf Ihren Cluster an:

```
kubectl apply -f https://s3.us-west-2.amazonaws.com/amazon-eks/docs/eks-console-restricted-access.yaml
```

Wenn Sie den Namen, den Namespace, die Berechtigungen oder eine andere Konfiguration in der Datei der Kubernetes-Gruppe ändern müssen, laden Sie die Datei herunter und bearbeiten Sie sie, bevor Sie sie auf Ihren Cluster anwenden:

1. Laden Sie die Datei mit einem der folgenden Befehle herunter.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/docs/eks-console-full-access.yaml
```

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/docs/eks-console-restricted-access.yaml
```

2. Bearbeiten Sie die Datei nach Bedarf.
3. Wenden Sie das Manifest mit einem der folgenden Befehle auf Ihren Cluster an:

```
kubectl apply -f eks-console-full-access.yaml
```

```
kubectl apply -f eks-console-restricted-access.yaml
```

3. Ordnen Sie den IAM-Prinzipal dem Kubernetes-Benutzer oder der Gruppe in der `aws-auth ConfigMap` zu. Sie können ein Tool wie `eksctl` verwenden, um die `ConfigMap` zu aktualisieren, oder Sie können sie durch manuelle Bearbeitung aktualisieren.

> **Important**
>
> Wir empfehlen die Verwendung von `eksctl` oder einem anderen Tool, um die `ConfigMap` zu bearbeiten. Weitere Informationen zu anderen Tools, die Sie verwenden können, finden Sie unter Verwenden von Tools zur Änderung von `aws-authConfigMap`

in den Best-Practice-Leitfäden zu Amazon EKS. Ist `aws-auth ConfigMap` falsch formatiert, können Sie den Zugriff auf Ihren Cluster verlieren.

eksctl

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

1. Zeigen Sie die aktuellen Mappings in der `ConfigMap` an. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters. Ersetzen Sie es *`region-code`* durch AWS-Region das, in dem sich Ihr Cluster befindet.

   ```
   eksctl get iamidentitymapping --cluster my-cluster --region=region-code
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   ARN
                     USERNAME                                GROUPS
               ACCOUNT
   arn:aws:iam::111122223333:role/eksctl-my-cluster-my-nodegroup-
   NodeInstanceRole-1XLS7754U3ZPA    system:node:{{EC2PrivateDNSName}}
    system:bootstrappers,system:nodes
   ```

2. Fügen Sie ein Mapping für eine Rolle hinzu. In diesem Beispiel wird davon ausgegangen, dass Sie die IAM-Berechtigungen im ersten Schritt einer Rolle namens *`my-console-viewer-role`* zuweisen. Ersetzen Sie *`111122223333`* durch Ihre Konto-ID.

   ```
   eksctl create iamidentitymapping \
       --cluster my-cluster \
       --region=region-code \
       --arn arn:aws:iam::111122223333:role/my-console-viewer-role \
       --group eks-console-dashboard-full-access-group \
       --no-duplicate-arns
   ```

> **Important**
>
> Der Rollen-ARN darf keinen Pfad wie `role/my-team/developers/my-role` enthalten. Das Format des ARN muss `arn:aws:iam::111122223333:role/my-role` sein. In diesem Beispiel muss `my-team/developers/` entfernt werden.

Eine Beispielausgabe sieht wie folgt aus.

```
[...]
2022-05-09 14:51:20 [#]  adding identity "arn:aws:iam::111122223333:role/my-console-viewer-role" to auth ConfigMap
```

3. Fügen Sie ein Mapping für einen Benutzer hinzu. Bewährte Methoden für IAM empfehlen, dass Sie Rollen statt Benutzern Berechtigungen gewähren. In diesem Beispiel wird davon ausgegangen, dass Sie die IAM-Berechtigungen im ersten Schritt einem Benutzer *my-user* zuweisen. Ersetzen Sie *111122223333* durch Ihre Konto-ID.

```
eksctl create iamidentitymapping \
    --cluster my-cluster \
    --region=region-code \
    --arn arn:aws:iam::111122223333:user/my-user \
    --group eks-console-dashboard-restricted-access-group \
    --no-duplicate-arns
```

Eine Beispielausgabe sieht wie folgt aus.

```
[...]
2022-05-09 14:53:48 [#]  adding identity "arn:aws:iam::111122223333:user/my-user" to auth ConfigMap
```

4. Zeigen Sie wieder die Mappings in der `ConfigMap` an.

```
eksctl get iamidentitymapping --cluster my-cluster --region=region-code
```

Eine Beispielausgabe sieht wie folgt aus.

```
ARN
                    USERNAME                                      GROUPS
                      ACCOUNT
arn:aws:iam::111122223333:role/eksctl-my-cluster-my-nodegroup-
NodeInstanceRole-1XLS7754U3ZPA    system:node:{{EC2PrivateDNSName}}
 system:bootstrappers,system:nodes
arn:aws:iam::111122223333:role/my-console-viewer-role
                                                                  eks-console-
dashboard-full-access-group
arn:aws:iam::111122223333:user/my-user
                                                                  eks-console-
dashboard-restricted-access-group
```

Edit ConfigMap manually

Weitere Informationen zum Hinzufügen von Benutzern oder Rollen zur `aws-auth` `ConfigMap` finden Sie unter Hinzufügen von IAM-Prinzipal zu Ihrem Amazon EKS-Cluster.

1. Öffnen Sie die `aws-auth` `ConfigMap` zum Bearbeiten.

   ```
   kubectl edit -n kube-system configmap/aws-auth
   ```

2. Fügen Sie die Mappings zur `aws-auth` `ConfigMap` hinzu, aber ersetzen Sie keine der vorhandenen Mappings. Im folgenden Beispiel werden Mappings zwischen IAM-Prinzipalen mit Berechtigungen hinzugefügt, die im ersten Schritt hinzugefügt wurden, sowie die im vorherigen Schritt erstellten Kubernetes-Gruppen:

   - Die *`my-console-viewer-role`*-Rolle und `eks-console-dashboard-full-access-group`.
   - Der *`my-user`*-Benutzer und `eks-console-dashboard-restricted-access-group`.

   In diesen Beispielen wird davon ausgegangen, dass Sie die IAM-Berechtigungen im ersten Schritt einer Rolle namens *`my-console-viewer-role`* und einem Benutzer namens *`my-user`* zuweisen. *`111122223333`*Ersetzen Sie es durch Ihre AWS Konto-ID.

   ```
   apiVersion: v1
   data:
   mapRoles: |
   ```

```
  - groups:
    - eks-console-dashboard-full-access-group
    rolearn: arn:aws:iam::111122223333:role/my-console-viewer-role
    username: my-console-viewer-role
mapUsers: |
  - groups:
    - eks-console-dashboard-restricted-access-group
    userarn: arn:aws:iam::111122223333:user/my-user
    username: my-user
```

> **Important**
>
> Der Rollen-ARN darf keinen Pfad wie `role/my-team/developers/my-console-viewer-role` enthalten. Das Format des ARN muss `arn:aws:iam::111122223333:role/my-console-viewer-role` sein. In diesem Beispiel muss `my-team/developers/` entfernt werden.

3. Speichern Sie die Datei und beenden Sie den Text-Editor.

Obervability in Amazon EKS

Sie können Ihre Daten in Amazon EKS mit vielen verfügbaren Überwachungs- oder Protokollierungstools beobachten. Ihre Amazon EKS-Protokolldaten können zu AWS-Services oder zu Partner-Tools zur Datenanalyse gestreamt werden. In der sind viele Dienste verfügbar AWS Management Console , die Daten zur Behebung Ihrer Amazon EKS-Probleme bereitstellen. Sie können auch eine AWS-unterstützte Open-Source-Lösung für die Überwachung der Amazon EKS-Infrastruktur verwenden.

Nachdem Sie im linken Navigationsbereich der Amazon-EKS-Konsole auf Cluster geklickt haben, können Sie den Cluster-Zustand und die Details anzeigen, indem Sie den Namen des Clusters auswählen. Informationen zu vorhandenen Kubernetes-Ressourcen, die in Ihrem Cluster bereitgestellt werden, finden Sie unter Anzeigen der Kubernetes-Ressourcen.

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon EKS und Ihren AWS Lösungen. Wir empfehlen Ihnen, Überwachungsdaten aus allen Teilen Ihrer AWS Lösung zu sammeln. Auf diese Weise können Sie Ausfälle, die sich über mehrere Punkte erstrecken, leichter debuggen. Bevor Sie mit der Überwachung von Amazon EKS beginnen, sollten Sie sicherstellen, dass für Ihren Überwachungsplan folgende Fragen beantwortet sind.

- Was sind Ihre Ziele? Benötigen Sie Echtzeit-Benachrichtigungen, wenn Ihre Cluster drastisch skaliert werden?
- Welche Ressourcen müssen beachtet werden?
- Wie oft müssen Sie diese Ressourcen beobachten? Muss Ihr Unternehmen schnell auf Risiken reagieren?
- Welche Tools möchten Sie verwenden? Wenn Sie das Programm bereits im AWS Fargate Rahmen Ihres Starts ausführen, können Sie den integrierten Protokollrouter verwenden.
- Wer soll die Überwachungsaufgaben ausführen?
- An wen sollen Benachrichtigungen gesendet werden, wenn etwas schief geht?

Protokollieren und Überwachen in Amazon EKS

Amazon EKS bietet integrierte Tools für die Protokollierung und Überwachung. Die Protokollierung der Steuerebene zeichnet alle API-Aufrufe zu Ihren Clustern. Diese enthält

Überwachungsinformationen darüber, welche Benutzer welche Aktionen an Ihren Clustern durchgeführt haben, sowie rollenbasierte Informationen. Weitere Informationen finden Sie unter Protokollieren und Überwachen in Amazon EKS in der verbindlichen AWS -Anleitung.

Die Protokollierung auf der Amazon EKS-Kontrollebene stellt Prüf- und Diagnoseprotokolle direkt von der Amazon EKS-Steuerebene zu den CloudWatch Protokollen in Ihrem Konto bereit. Diese Protokolle erleichtern Ihnen die Absicherung und Ausführung Ihrer Cluster. Sie können genau die Protokolltypen auswählen, die Sie benötigen, und die Protokolle werden als Protokollstreams an eine Gruppe für jeden Amazon EKS-Cluster gesendet CloudWatch. Weitere Informationen finden Sie unter Amazon-EKS-Steuerebenen-Protokollierung.

> **Note**
>
> Wenn Sie die Amazon EKS-Authentifikatorprotokolle in Amazon überprüfen CloudWatch, werden die Einträge angezeigt, die Text enthalten, der dem folgenden Beispieltext ähnelt.
>
> ```
> level=info msg="mapping IAM role" groups="[]"
>  role="arn:aws:iam::111122223333:role/XXXXXXXXXXXXXXXXXX-
> NodeManagerRole-XXXXXXXX" username="eks:node-manager"
> ```
>
> Einträge, die diesen Text enthalten, werden erwartet. Das `username` ist eine interne Amazon EKS-Servicerolle, die bestimmte Vorgänge für verwaltete Knotengruppen und Fargate ausführt.
> Für eine detaillierte, anpassbare Protokollierung ist Kubernetes-Protokollierung verfügbar.

Amazon EKS ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon EKS ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für Amazon EKS als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon-EKS-Konsole und Code-Aufrufe der Amazon-EKS-API-Operationen. Weitere Informationen finden Sie unter Protokollieren von Amazon EKS-API-Aufrufen mit AWS CloudTrail.

Der Kubernetes-API-Server stellt eine Reihe von Metriken zur Verfügung, die für die Überwachung und Analyse hilfreich sind. Weitere Informationen finden Sie unter Prometheus-Metriken.

Informationen Fluent Bit zur Konfiguration von benutzerdefinierten Amazon CloudWatch Protokollen finden Sie unter Einrichtung Fluent Bit im CloudWatch Amazon-Benutzerhandbuch.

Protokollierungs- und Überwachungs-Tools von Amazon EKS

Amazon Web Services bietet verschiedene Tools, mit deren Hilfe Sie Amazon EKS überwachen können. Sie können einige Tools konfigurieren, um eine automatische Überwachung einzurichten, einige erfordern jedoch manuelle Anrufe. Wir empfehlen, dass Sie Überwachungsaufgaben so weit automatisieren, wie es Ihre Umgebung und Ihr vorhandenes Toolset zulassen.

Protokollierungs-Tools

Bereiche	Tool	Beschreibung	Setup
Anwendungen	Einblicke in Amazon CloudWatch Container	Sammelt, aggregiert und fasst Metriken und Protokolle von Ihren containerisierten Anwendungen und Microservices zusammen.	Einrichtungsverfahren
Steuerebene	AWS CloudTrail	Protokolliert API-Aufrufe eines Benutzers, einer Rolle oder eines Service.	Einrichtungsverfahren
Mehrere Bereiche für AWS Fargate Instanzen	AWS Fargate Router protokollieren	Beispielsweise AWS Fargate streamt er Protokolle an AWS Dienste oder Partner-Tools. Verwendet AWS für Fluent Bit. Protokolle können an andere Tools	Einrichtungsverfahren

Bereiche	Tool	Beschreibung	Setup
		AWS-Services oder Partner-Tools gestreamt werden.	

Überwachungstools

Bereiche	Tool	Beschreibung	Setup
Anwendungen	CloudWatch Container Insights	CloudWatch Container Insights sammelt, aggregiert und fasst Metriken und Protokolle aus Ihren containerisierten Anwendungen und Microservices zusammen.	Einrichtungsverfahren
Anwendungen	AWS Distro für OpenTelemetry (ADOT)	Es sammelt korrelierte Metriken, Trace-Daten und Metadaten und sendet sie an Überwachungsdienste oder Partner. AWS Es kann über CloudWatch Container Insights	Einrichtungsverfahren

Bereiche	Tool	Beschreibung	Setup
		eingerichtet werden.	
Anwendungen	DevOpsAmazon-Guru	Erkennt betriebliche Leistung und Verfügbarkeit auf Knotenebene.	Einrichtungsverfahren
Anwendungen	AWS X-Ray	Empfängt Trace-Daten über Ihre Anwendung. Diese Trace-Daten enthalten ein- und ausgehende Anfragen und Metadaten zu diesen. Für Amazon EKS erfordert die Implementierung das OpenTelemetry-Add-on.	Einrichtungsverfahren
Anwendungen	Amazon CloudWatch Observability-Betreiber	Der Amazon CloudWatch Observability Operator sammelt Metriken, Protokolle und Trace-Daten. Es sendet sie an Amazon CloudWatch und AWS X-Ray.	Einrichtungsverfahren

Bereiche	Tool	Beschreibung	Setup
Steuerebene	Prometheus	CloudWatch Die Gebühren für die Aufnahme von Protokollen, die Archivierung und das Scannen von Daten gelten für aktivierte Protokolle auf der Kontrolle bene.	Einrichtu ngsverfahren

Prometheus-Metriken

Prometheus ist eine Überwachungs- und Zeitreihendatenbank, die Endpunkte durchsucht. Sie bietet die Möglichkeit, gesammelte Daten abzufragen, zu aggregieren und zu speichern. Sie können sie auch für Warnungen und für die Aggregation von Warnungen verwenden. In diesem Thema wird erklärt, wie Sie Prometheus als verwaltete oder als Open-Source-Option einrichten. Die Überwachung der Metriken der Amazon-EKS-Steuerebene ist ein häufiger Anwendungsfall.

Amazon Managed Service für Prometheus ist ein Prometheus-kompatibler Überwachungs- und Warnungsservice, der die Überwachung von containerisierten Anwendungen und Infrastrukturen in großem Umfang vereinfacht. Es ist ein vollständig verwalteter Service, der die Aufnahme, Speicherung, Abfrage und Warnung Ihrer Metriken automatisch skaliert. Es lässt sich auch in AWS Sicherheitsdienste integrieren, um einen schnellen und sicheren Zugriff auf Ihre Daten zu ermöglichen. Sie können die Open-Source-PromQL-Abfragesprache verwenden, um Ihre Metriken abzufragen und darauf zu warnen.

Weitere Informationen zur Verwendung der aktivierten Prometheus-Metriken finden Sie im Benutzerhandbuch von Amazon Managed Service für Prometheus.

Aktivieren von Prometheus-Metriken beim Erstellen eines Clusters

> **Important**
>
> Die Ressourcen von Amazon Managed Service for Prometheus befinden sich außerhalb des Cluster-Lebenszyklus und müssen unabhängig vom Cluster verwaltet werden. Wenn Sie Ihren Cluster löschen, stellen Sie sicher, dass Sie auch alle entsprechenden Scraper löschen, um die anfallenden Kosten zu stoppen. Weitere Informationen finden Sie unter Suchen und Löschen von Scrapern im Amazon Managed Service for Prometheus Benutzerhandbuch.

Wenn Sie einen neuen Cluster erstellen, können Sie die Option aktivieren, Metriken an Prometheus zu senden. In der befindet AWS Management Console sich diese Option im Schritt Observability konfigurieren beim Erstellen eines neuen Clusters. Weitere Informationen finden Sie unter Erstellen eines Amazon-EKS-Clusters.

Prometheus erkennt und sammelt Metriken aus Ihrem Cluster über ein Pull-basiertes Modell namens Scraping. Scraper sind dafür eingerichtet, Daten aus Ihrer Cluster-Infrastruktur und containerisierten Anwendungen zu sammeln.

Wenn Sie die Option zum Senden von Prometheus-Metriken aktivieren, bietet Amazon Managed Service für Prometheus einen vollständig verwalteten agentenlosen Scraper. Verwenden Sie die folgenden erweiterten Konfigurationsoptionen, um den Standard-Scraper nach Bedarf anzupassen.

Scraper-Alias

(Optional) Geben Sie einen eindeutigen Alias für den Scraper ein.

Bestimmungsort

Wählen Sie einen Workspace für Amazon Managed Service für Prometheus aus. Ein Workspace ist ein logischer Bereich für die Speicherung und Abfrage von Prometheus-Metriken. In diesem Workspace können Sie die Prometheus-Metriken aller Konten einsehen, die Zugriff darauf haben. Die Option Neuen Workspace erstellen weist Amazon EKS an, unter Verwendung des von Ihnen angegebenen Workspace-Alias einen Workspace in Ihrem Namen zu erstellen. Mit der Option Bestehenden Workspace auswählen können Sie einen vorhandenen Workspace aus einer Dropdown-Liste auswählen. Weitere Informationen zu Workspaces finden Sie unter Workspaces verwalten im Benutzerhandbuch von Amazon Managed Service für Prometheus.

Zugriff auf Services

In diesem Abschnitt werden die Berechtigungen zusammengefasst, die Sie beim Senden von Prometheus-Metriken gewähren:

- Zulassen, dass Amazon Managed Service für Prometheus den gescrapten Amazon-EKS-Cluster beschreibt
- Remote-Schreiben in den Workspace von Amazon Managed Prometheus

Wenn `AmazonManagedScraperRole` bereits vorhanden ist, verwendet der Scraper diese Rolle. Wählen Sie den `AmazonManagedScraperRole`-Link aus, um die Berechtigungsdetails anzuzeigen. Falls `AmazonManagedScraperRole` noch nicht vorhanden ist, wählen Sie den Link Berechtigungsdetails anzeigen aus, um die spezifischen Berechtigungen zu sehen, die Sie durch das Senden von Prometheus-Metriken gewähren.

Subnetze

Sehen Sie sich die Subnetze an, die der Scraper übernimmt. Wenn Sie sie ändern müssen, kehren Sie zum Schritt Netzwerk angeben der Cluster-Erstellung zurück.

Sicherheitsgruppen

Sehen Sie sich die Sicherheitsgruppen an, die der Scraper übernimmt. Wenn Sie sie ändern müssen, kehren Sie zum Schritt Netzwerk angeben der Cluster-Erstellung zurück.

Scraper-Konfiguration

Ändern Sie die Scraper-Konfiguration nach Bedarf im YAML-Format. Verwenden Sie dazu das Formular oder laden Sie eine YAML-Ersatzdatei hoch. Weitere Informationen finden Sie unter Scraper-Konfiguration im Benutzerhandbuch von Amazon Managed Service für Prometheus.

Amazon Managed Service für Prometheus bezieht sich auf den agentenlosen Scraper, der zusammen mit dem Cluster als verwalteter AWS -Sammler erstellt wird. Weitere Informationen zu AWS verwalteten Collectors finden Sie unter AWS Managed Collectors im Amazon Managed Service for Prometheus User Guide.

Important

Sie müssen Ihre `aws-auth ConfigMap` so einrichten, dass der Scraper Cluster-interne Berechtigungen erhält. Weitere Informationen finden Sie unter Konfigurieren Ihres Amazon-EKS-Clusters im Benutzerhandbuch von Amazon Managed Service für Prometheus.

Anzeigen von Prometheus-Scraper-Details

Nachdem Sie einen Cluster erstellt haben, bei dem Prometheus-Metriken aktiviert sind, können Sie Ihre Prometheus-Scraper-Details anzeigen. Wenn Sie sich Ihren Cluster im ansehen AWS Management Console, wählen Sie den Tab Observability. Eine Tabelle zeigt eine Liste der Scraper für den Cluster, einschließlich Informationen wie Scraper-ID, Alias, Status und Erstellungsdatum.

Um weitere Details zum Scraper anzuzeigen, wählen Sie einen Scraper-ID-Link aus. Beispielsweise können Sie die Scraper-Konfiguration, den Amazon-Ressourcennamen (ARN), die Remote-Schreib-URL und Netzwerkinformationen anzeigen. Sie können die Scraper-ID als Eingabe für Operationen in der Amazon-Managed-Service-für-Prometheus-API wie `DescribeScraper` und `DeleteScraper` verwenden. Sie können die API auch zum Erstellen weiterer Scraper verwenden.

Weitere Informationen zur Verwendung der Prometheus-API finden Sie in der API-Referenz zu Amazon Managed Service für Prometheus.

Bereitstellen von Prometheus mit Helm

Alternativ können Sie Prometheus mit Helm V3 in Ihrem Cluster bereitstellen. Wenn Sie Helm bereits installiert haben, können Sie Ihre Version mit dem Befehl `helm version` überprüfen. Helm ist ein Paketmanager für Kubernetes-Cluster. Weitere Informationen zu Helm und zur Installation finden Sie unter Verwendung von Helm mit Amazon EKS.

Nachdem Sie Helm für Ihren Amazon-EKS-Cluster konfiguriert haben, können Sie es verwenden, um Prometheus mit den folgenden Schritten bereitzustellen.

Prometheus stellen Sie mit Helm wie folgt bereit:

1. Erstellen Sie einen Prometheus-Namespace.

```
kubectl create namespace prometheus
```

2. Fügen Sie das `prometheus-community`-Diagramm-Repository hinzu.

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

3. Stellen Sie Prometheus bereit.

```
helm upgrade -i prometheus prometheus-community/prometheus \
```

```
    --namespace prometheus \
    --set alertmanager.persistence.storageClass="gp2" \
    --set server.persistentVolume.storageClass="gp2"
```

> **Note**
>
> Wenn Sie bei der Ausführung dieses Befehls die Fehlermeldung `Error: failed to download "stable/prometheus" (hint: running `helm repo update` may help)` erhalten, führen Sie `helm repo update prometheus-community` aus, und versuchen Sie dann erneut, den Befehl aus Schritt 2 auszuführen.
> Wenn Sie die Fehlermeldung `Error: rendered manifests contain a resource that already exists` erhalten, führen Sie `helm uninstall` *`your-release-name`* `-n` *`namespace`* aus, und versuchen Sie dann, den Befehl aus Schritt 3 erneut auszuführen.

4. Stellen Sie sicher, dass alle Pods im `prometheus`-Namespace den Status `READY` haben.

```
kubectl get pods -n prometheus
```

Eine Beispielausgabe sieht wie folgt aus.

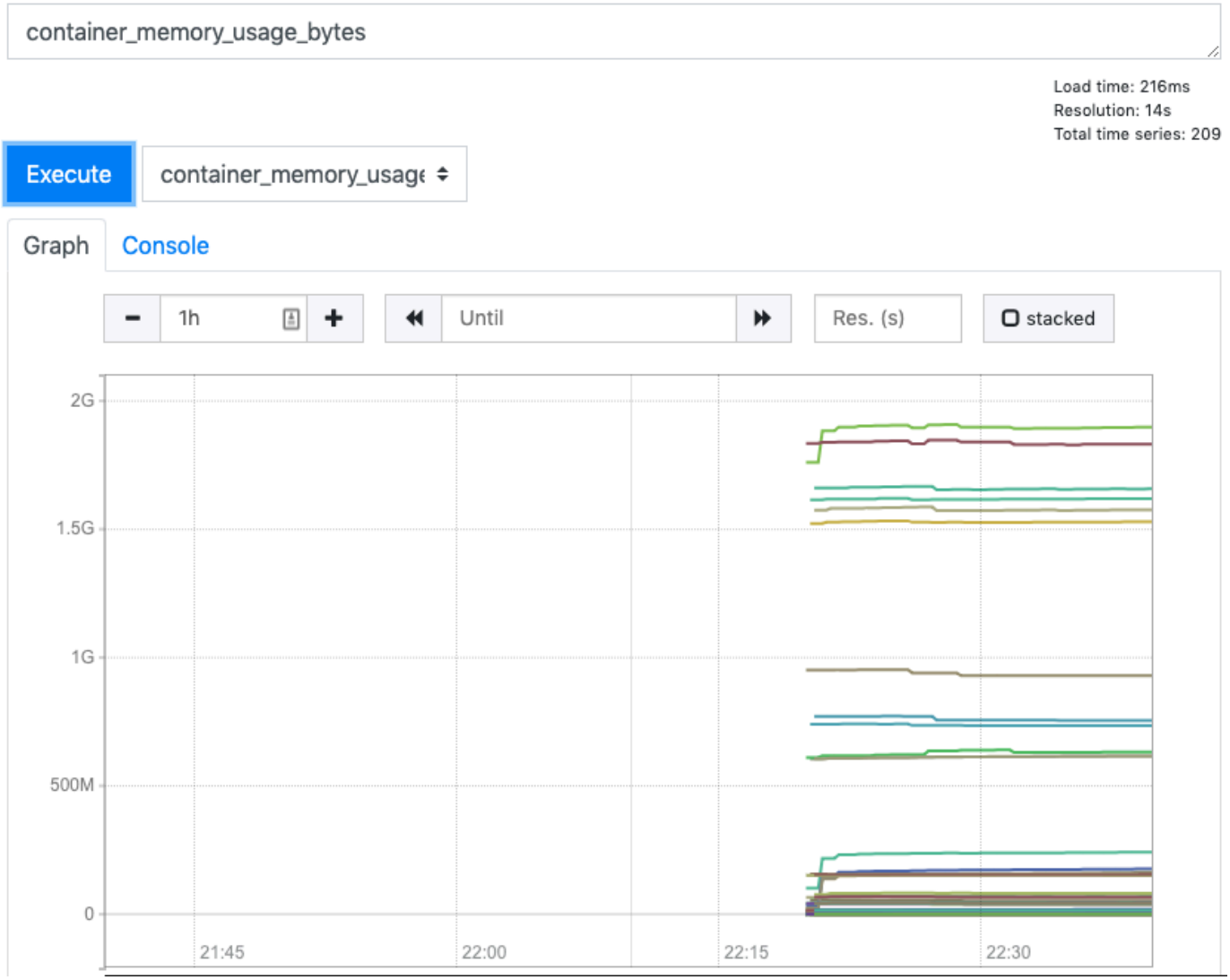
```
NAME                                             READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-59b4c8c744-r7bgp         1/2     Running   0          48s
prometheus-kube-state-metrics-7cfd87cf99-jkz2f   1/1     Running   0          48s
prometheus-node-exporter-jcjqz                   1/1     Running   0          48s
prometheus-node-exporter-jxv2h                   1/1     Running   0          48s
prometheus-node-exporter-vbdks                   1/1     Running   0          48s
prometheus-pushgateway-76c444b68c-82tnw          1/1     Running   0          48s
prometheus-server-775957f748-mmht9               1/2     Running   0          48s
```

5. Verwenden Sie `kubectl`, um die Prometheus-Konsole auf Ihren lokalen Computer weiterzuleiten.

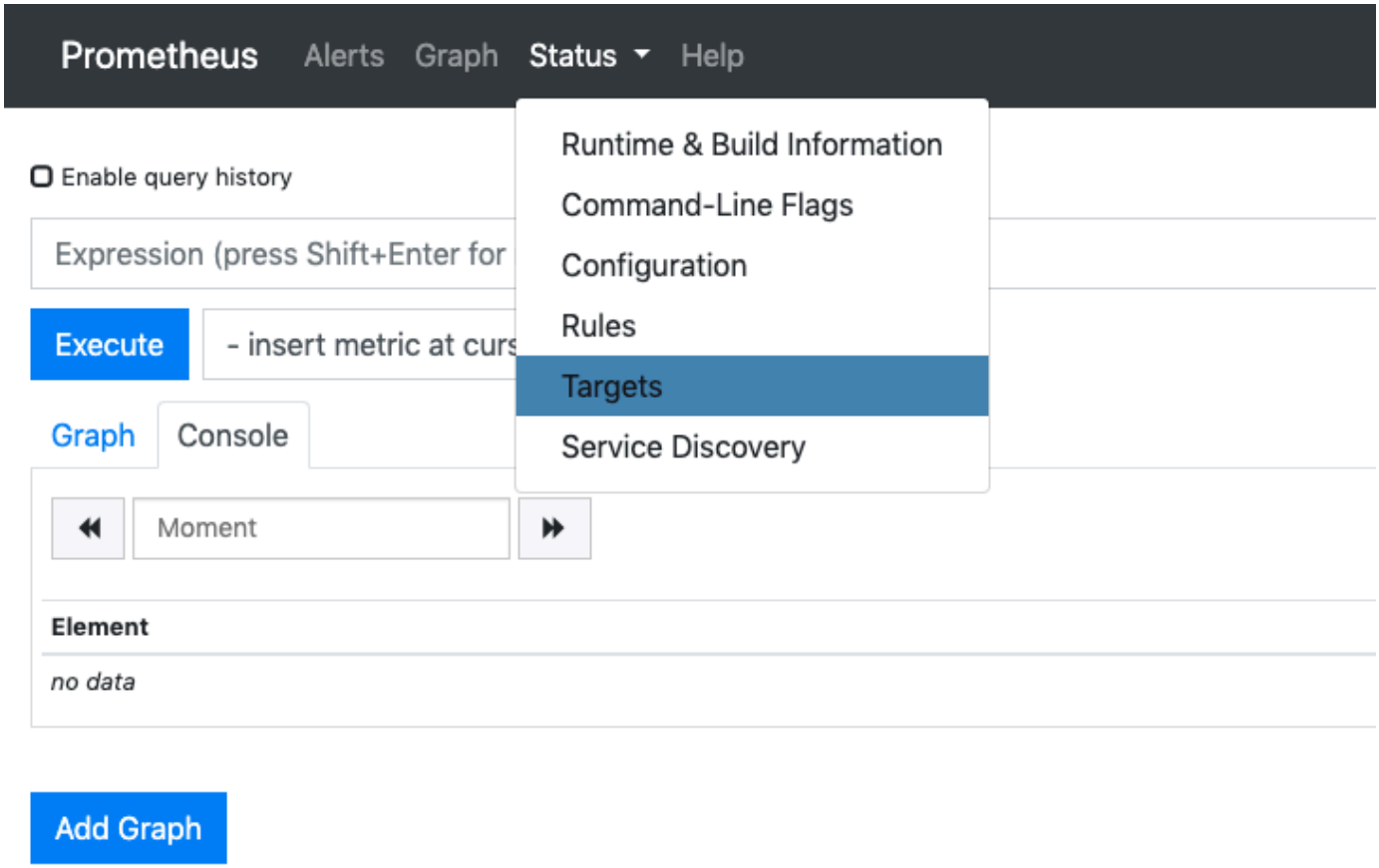
```
kubectl --namespace=prometheus port-forward deploy/prometheus-server 9090
```

6. Richten Sie einen Webbrowser auf `http://localhost:9090`, um die Prometheus-Konsole anzuzeigen.
7. Wählen Sie eine Metrik aus dem Menü - insert metric at cursor (Metrik bei Cursor eingeben) und danach Execute (Ausführen). Wählen Sie die Registerkarte Graph

(Diagramm) aus, um die Metrik im Zeitverlauf anzuzeigen. Das folgende Image zeigt `container_memory_usage_bytes` im Zeitverlauf.



8. Wählen Sie in der oberen Navigationsleiste Status, dann Targets (Ziele).



Es werden alle Kubernetes-Endpunkte angezeigt, die über die Service-Erkennung mit Prometheus verbunden sind.

Anzeigen von Rohmetriken der Steuerebene

Als Alternative zur Bereitstellung von Prometheus gibt der Kubernetes-API-Server eine Reihe von Metriken frei, die in einem Prometheus-Format dargestellt werden. Diese Metriken sind für die Überwachung und Analyse hilfreich. Sie werden intern durch einen Metrikendpunkt freigegeben, der die `/metrics`-HTTP-API referenziert. Wie andere Endpunkte wird auch dieser Endpunkt über die Amazon-EKS-Steuerebene exponiert. Dieser Endpunkt ist in erster Linie nützlich, um sich eine bestimmte Metrik anzusehen. Um Metriken im Laufe der Zeit zu analysieren, empfehlen wir die Bereitstellung von Prometheus.

Um die Ausgabe der Rohmetriken anzuzeigen, verwenden Sie `kubectl` mit dem Flag `--raw`. Mit diesem Befehl können Sie einen beliebigen HTTP-Pfad übergeben und die Rohantwort zurückgeben.

```
kubectl get --raw /metrics
```

Eine Beispielausgabe sieht wie folgt aus.

```
[...]
# HELP rest_client_requests_total Number of HTTP requests, partitioned by status code,
 method, and host.
# TYPE rest_client_requests_total counter
rest_client_requests_total{code="200",host="127.0.0.1:21362",method="POST"} 4994
rest_client_requests_total{code="200",host="127.0.0.1:443",method="DELETE"} 1
rest_client_requests_total{code="200",host="127.0.0.1:443",method="GET"} 1.326086e+06
rest_client_requests_total{code="200",host="127.0.0.1:443",method="PUT"} 862173
rest_client_requests_total{code="404",host="127.0.0.1:443",method="GET"} 2
rest_client_requests_total{code="409",host="127.0.0.1:443",method="POST"} 3
rest_client_requests_total{code="409",host="127.0.0.1:443",method="PUT"} 8
# HELP ssh_tunnel_open_count Counter of ssh tunnel total open attempts
# TYPE ssh_tunnel_open_count counter
ssh_tunnel_open_count 0
# HELP ssh_tunnel_open_fail_count Counter of ssh tunnel failed open attempts
# TYPE ssh_tunnel_open_fail_count counter
ssh_tunnel_open_fail_count 0
```

Diese Rohausgabe zeigt exakt das, was der API-Server bereitstellt. Die verschiedenen Metriken werden zeilenweise aufgelistet, wobei jede Zeile einen Metriknamen, Tags und einen Wert enthält.

```
metric_name{"tag"="value"[,...]}
            value
```

Amazon EKS-Add-On-Unterstützung für Amazon CloudWatch

Amazon CloudWatch Observability erfasst Protokolle, Metriken und Trace-Daten in Echtzeit. Es sendet sie an Amazon CloudWatch und AWS X-Ray. Sie können dieses Add-on installieren, um sowohl CloudWatch Application Signals als auch CloudWatch Container Insights mit verbesserter Observability für Amazon EKS zu aktivieren. Dies hilft Ihnen, den Zustand und die Leistung Ihrer Infrastruktur und containerisierten Anwendungen zu überwachen. Der Amazon CloudWatch Observability Operator dient zur Installation und Konfiguration der erforderlichen Komponenten.

Amazon EKS unterstützt Amazon CloudWatch Observability Operator als Amazon-EKS-Add-on. Das Container Insights Add-on ermöglicht Linux sowohl Windows Worker-Knoten als auch Worker-Knoten im Cluster. Um Container Insights aktiviert zu werdenWindows, muss die Amazon EKS-Add-On-Version `1.5.0` oder höher sein. Derzeit wird CloudWatch Application Signals auf Amazon EKS nicht unterstütztWindows.

In den folgenden Themen wird beschrieben, wie Sie Amazon CloudWatch Observability Operator für Ihren Amazon-EKS-Cluster verwenden.

- Anweisungen zur Installation dieses Add-ons finden Sie unter Installieren des CloudWatch Agenten mithilfe der Amazon EKS-Add-Ons für CloudWatch Observability im CloudWatch Amazon-Benutzerhandbuch.
- Weitere Informationen zu CloudWatch Anwendungssignalen finden Sie unter Anwendungssignale.
- Weitere Informationen zu Container Insights finden Sie unter Verwendung Container Insights im CloudWatch Amazon-Benutzerhandbuch.

Amazon-EKS-Steuerebenen-Protokollierung

Die Amazon-EKS-Steuerebenenprotokollierung stellt Prüfungs- und Diagnoseprotokolle direkt von der Amazon-EKS-Steuerebene zu CloudWatch Protokollen in Ihrem Konto bereit. Diese Protokolle erleichtern Ihnen die Absicherung und Ausführung Ihrer Cluster. Sie können die genauen Protokolltypen auswählen, die Sie benötigen, und Protokolle werden für jeden Amazon-EKS-Cluster in als Protokollstreams an eine Gruppe gesendet CloudWatch. Weitere Informationen finden Sie unter Amazon CloudWatch-Protokollierung.

Sie können mit der Protokollierung der Amazon-EKS-Steuerebene beginnen, indem Sie die für jeden neuen oder bestehenden Amazon-EKS-Cluster zu aktivierenden Protokolltypen auswählen. Sie können die einzelnen Protokolltypen pro Cluster über die AWS Management Console, AWS CLI (Version `1.16.139` oder höher) oder die Amazon-EKS-API aktivieren oder deaktivieren. Wenn diese Option aktiviert ist, werden Protokolle automatisch vom Amazon-EKS-Cluster an CloudWatch Protokolle im selben Konto gesendet.

Wenn Sie die Protokollierung der Amazon-EKS-Steuerebene verwenden, werden Ihnen für jeden betriebenen Cluster die Amazon-EKS-Standardpreise berechnet. Ihnen werden die standardmäßigen Kosten für die Erfassung und Speicherung von CloudWatch Protokollen für Protokolle in Rechnung gestellt, die von Ihren Clustern an CloudWatch Protokolle gesendet werden. Ihnen werden außerdem alle als Teil Ihres Clusters bereitgestellten AWS-Ressourcen (z. B. Amazon-EC2-Instances oder Amazon-EBS-Volumes) berechnet.

Die folgenden Cluster-Steuerebenen-Protokolltypen sind verfügbar. Jeder Protokolltyp entspricht einer Komponente der Kubernetes-Steuerebene. Um mehr über diese Komponenten zu erfahren, lesen Sie Kubernetes-Komponenten in der Kubernetes-Dokumentation.

API-Server (**`api`**)

Der API-Server Ihres Clusters ist die Komponente der Steuerebene, die die Kubernetes-API verfügbar macht. Wenn Sie API-Serverprotokolle beim Starten des Clusters oder kurz danach aktivieren, enthalten die Protokolle API-Server-Markierungen, die zum Starten des API-Servers verwendet wurden. Weitere Informationen finden Sie unter `kube-apiserver` und in der Prüfungsrichtlinie in der Kubernetes-Dokumentation.

Prüfung (**`audit`**)

Kubernetes-Prüfungsprotokolle bieten eine Aufzeichnung der einzelnen Benutzer, Administratoren oder Systemkomponenten, die Ihren Cluster beeinflusst haben. Weitere Informationen finden Sie unter Prüfung in der Kubernetes-Dokumentation.

Authenticator (**`authenticator`**)

Authenticator-Protokolle sind eindeutig für Amazon EKS. Diese Protokolle stellen die Steuerebenenkomponente dar, die Amazon EKS für die Kubernetes-Authentifizierung per rollenbasierter Zugriffssteuerung (RBAC) mit IAM-Anmeldeinformationen verwendet. Weitere Informationen finden Sie unter Clusterverwaltung.

Controller-Manager (**`controllerManager`**)

Der Controller-Manager verwaltet die zentralen Regelkreise, die mit Kubernetes ausgeliefert werden. Weitere Informationen finden Sie unter kube-controller-manager in der Kubernetes-Dokumentation.

Scheduler (**`scheduler`**)

Die Scheduler-Komponente verwaltet, wann und wo Pods in Ihrem Cluster ausgeführt werden. Weitere Informationen finden Sie unter kube-scheduler in der Kubernetes-Dokumentation.

Aktivieren und Deaktivieren von Steuerebenenprotokollen

Standardmäßig werden Protokolle der Cluster-Steuerebene nicht an CloudWatch -Protokolle gesendet. Sie müssen jeden Protokolltyp einzeln aktivieren, um Protokolle für Ihren Cluster zu senden. Für aktivierte Protokolle der Steuerebene gelten die Raten für die Aufnahme von CloudWatch Protokollen, die Archivierung und das Scannen von Daten. Weitere Informationen finden Sie unter -CloudWatch Preise.

Um die Protokollierungskonfiguration der Steuerebene zu aktualisieren, benötigt Amazon EKS bis zu fünf verfügbare IP-Adressen in jedem Subnetz. Wenn Sie einen Protokolltyp aktivieren, werden die Protokolle mit der Protokollausführungsstufe 2 gesendet.

AWS Management Console

So aktivieren oder deaktivieren Sie Steuerebenenprotokolle mit der AWS Management Console:

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters aus, um Ihre Cluster-Informationen anzuzeigen.
3. Wählen Sie den Registerkarte Beobachtbarkeit.
4. Wählen Sie im Abschnitt Steuerebenen-Protokollierung die Option Protokollierung verwalten aus.
5. Wählen Sie für jeden Protokolltyp aus, ob er aktiviert oder deaktiviert sein soll. Standardmäßig sind alle Protokollierungstypen deaktiviert.
6. Wählen Sie Änderungen speichern, um den Vorgang abzuschließen.

AWS CLI

So aktivieren oder deaktivieren Sie Steuerebenenprotokolle mit der AWS CLI:

1. Sie können Ihre AWS CLI-Version mit dem folgenden Befehl überprüfen.

   ```
   aws --version
   ```

   Bei einer AWS CLI-Version vor `1.16.139` müssen Sie zunächst auf die neueste Version aktualisieren. Informationen zum Installieren oder Aktualisieren des AWS CLI finden Sie unter Installieren des AWS Command Line Interface im AWS Command Line Interface-Benutzerhandbuch.
2. Aktualisieren Sie die Exportkonfiguration des Steuerebenenprotokolls Ihres Clusters mit dem folgenden AWS CLI-Befehl. Ersetzen Sie *`my-cluster`* durch den Namen Ihres Clusters und geben Sie die gewünschten Endpunkt-Zugriffswerte ein.

> **Note**
>
> Der folgende Befehl sendet alle verfügbaren Protokolltypen an CloudWatch Logs.

```
aws eks update-cluster-config \
    --region region-code \
    --name my-cluster \
    --logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "update": {
        "id": "883405c8-65c6-4758-8cee-2a7c1340a6d9",
        "status": "InProgress",
        "type": "LoggingUpdate",
        "params": [
            {
                "type": "ClusterLogging",
                "value": "{\"clusterLogging\":[{\"types\":[\"api\",\"audit\",
\"authenticator\",\"controllerManager\",\"scheduler\"],\"enabled\":true}]}"
            }
        ],
        "createdAt": 1553271814.684,
        "errors": []
    }
}
```

3. Überwachen Sie den Status Ihres Protokoll-Konfigurationsupdates mit dem folgenden Befehl unter Verwendung des Cluster-Namens und der Update-ID, die vom vorherigen Befehl zurückgegeben wurden. Ihre Aktualisierung ist abgeschlossen, wenn als Status `Successful` angezeigt wird.

```
aws eks describe-update \
    --region region-code\
    --name my-cluster \
    --update-id 883405c8-65c6-4758-8cee-2a7c1340a6d9
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "update": {
        "id": "883405c8-65c6-4758-8cee-2a7c1340a6d9",
        "status": "Successful",
        "type": "LoggingUpdate",
        "params": [
            {
                "type": "ClusterLogging",
                "value": "{\"clusterLogging\":[{\"types\":[\"api\",\"audit\",
\"authenticator\",\"controllerManager\",\"scheduler\"],\"enabled\":true}]}"
            }
        ],
        "createdAt": 1553271814.684,
        "errors": []
    }
}
```

Anzeigen von Cluster-Steuerebenenprotokollen

Nachdem Sie einen der Protokolltypen der Steuerebene für Ihren Amazon-EKS-Cluster aktiviert haben, können Sie sie in der CloudWatch Konsole anzeigen.

Weitere Informationen zum Anzeigen, Analysieren und Verwalten von Protokollen in CloudWatchfinden Sie im Amazon- CloudWatch Logs-Benutzerhandbuch.

So zeigen Sie Ihre Protokolle der Cluster-Steuerebene in der CloudWatch Konsole an

1. Öffnen Sie die CloudWatch -Konsole. Der Link öffnet die Konsole und zeigt Ihre aktuell verfügbaren Protokollgruppen an und filtert sie mit dem `/aws/eks`-Präfix.
2. Wählen Sie den Cluster aus, für den Sie die Protokolle anzeigen möchten. Das Format für den Namen der Protokollgruppen lautet `/aws/eks/`*`my-cluster`*`/cluster`.
3. Wählen Sie den anzuzeigenden Protokollstream aus. Die folgende Liste beschreibt das Namensformat des Protokollstreams der einzelnen Protokolltypen.

> **Note**
>
> Wenn die Daten des Protokollstreams ansteigen, werden die Namen des Protokollstreams rotiert. Wenn mehrere Protokollstreams für einen bestimmten Protokolltyp vorhanden sind, können Sie den neuesten Protokollstream anzeigen, indem Sie nach dem Namen des Protokollstreams mit der letzten Last event time (Uhrzeit des letzten Ereignisses) suchen.

- Kubernetes-API-Server-Komponentenprotokolle (**`api`**) – `kube-apiserver-`*`1234567890abcdef01234567890abcde`*
- Prüfung (**`audit`**) – `kube-apiserver-audit-`*`1234567890abcdef01234567890abcde`*
- Authentifikator (**`authenticator`**) – `authenticator-`*`1234567890abcdef01234567890abcde`*
- Controller-Manager (**`controllerManager`**) – `kube-controller-manager-`*`1234567890abcdef01234567890abcde`*
- Scheduler (**`scheduler`**) – `kube-scheduler-`*`1234567890abcdef01234567890abcde`*

4. Sehen Sie sich die Ereignisse des Protokollstreams an.

   Sie sollten beispielsweise die anfänglichen API-Server-Flags für den Cluster sehen, wenn Sie den oberen Teil von `kube-apiserver-`*`1234567890abcdef01234567890abcde`* anzeigen.

> **Note**
>
> Wenn die API-Serverprotokolle am Anfang des Protokolldatenstroms nicht angezeigt werden, ist es wahrscheinlich, dass die API-Serverprotokolldatei auf dem Server rotiert wurde, bevor Sie die API-Serverprotokollierung auf dem Server aktiviert haben. Alle Protokolldateien, die rotiert werden, bevor die API-Serverprotokollierung aktiviert ist, können nicht nach exportiert werden CloudWatch.
> Sie können jedoch einen neuen Cluster mit derselben Kubernetes-Version erstellen und beim Erstellen des Clusters die API-Serverprotokollierung aktivieren. Für Cluster mit derselben Plattformversion sind dieselben Flags aktiviert, daher sollten Ihre Flags mit den Flags des neuen Clusters übereinstimmen. Wenn Sie die Anzeige der Flags für den neuen Cluster in abgeschlossen haben CloudWatch, können Sie den neuen Cluster löschen.

Protokollieren von Amazon EKS-API-Aufrufen mit AWS CloudTrail

Amazon EKS ist in AWS CloudTrail integriert. CloudTrail ist ein Service, der eine Aufzeichnung der Aktionen eines Benutzers, einer Rolle oder eines AWS-Services in Amazon EKS bereitstellt. CloudTrail erfasst alle API-Aufrufe für Amazon EKS als Ereignisse. Dazu gehören Aufrufe von der Amazon-EKS-Konsole und von Code-Aufrufen der Amazon-EKS-API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon S3-Bucket aktivieren. Dies umfasst Ereignisse von Amazon EKS. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole trotzdem in Ereignisverlauf anzeigen. Anhand der Informationen, die CloudTrail sammelt, können Sie mehrere Details zu einer Anfrage ermitteln. Sie können zum Beispiel ermitteln, wann eine Anfrage an Amazon EKS gestellt wurde, von welcher IP-Adresse aus die Anfrage gestellt wurde und wer die Anfrage gestellt hat.

Weitere Informationen zu CloudTrail finden Sie im AWS CloudTrail-Benutzerhandbuch.

Themen

Amazon EKS-Informationen in CloudTrail

Wenn Sie Ihr AWS-Konto erstellen, wird CloudTrail ebenfalls in Ihrem AWS-Konto aktiviert. Wenn in Amazon EKS eine Aktivität auftritt, wird sie in einem CloudTrail-Ereignis zusammen mit anderen AWS-Service-Ereignissen im Ereignisverlauf aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS-Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail-API-Ereignisverlauf.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, darunter Ereignisse für Amazon EKS, können Sie einen Pfad (Trail) erstellen. Ein Trail ermöglicht es CloudTrail, Protokolldateien in einem Amazon-S3-Bucket bereitzustellen. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen AWS-Regionen in der AWS-Partition und stellt die Protokolldateien in dem Amazon-S3-Bucket bereit, den Sie angeben. Darüber hinaus können Sie andere AWS-Services konfigurieren, um die in den

CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu handeln. Weitere Informationen finden Sie in den folgenden Ressourcen.

- Übersicht zum Erstellen eines Trails
- In CloudTrail unterstützte Services und Integrationen
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail-Protokolldateien von mehreren Konten.

Alle Amazon EKS-Aktionen werden von CloudTrail protokolliert und sind in der Amazon EKS-API-Referenz dokumentiert. Zum Beispiel werden durch Aufrufe der Abschnitte `CreateCluster`, `ListClusters` und `DeleteCluster` Einträge in den CloudTrail-Protokolldateien generiert.

Jeder Ereignis- oder Protokolleintrag enthält Informationen über die Art der für die Abfrage genutzten IAM-Identität und dazu, welche Anmeldeinformationen verwendet wurden. Wenn temporäre Anmeldeinformationen verwendet wurden, zeigt der Eintrag, wie die Anmeldeinformationen bezogen wurden.

Weitere Informationen finden Sie unter CloudTrail-Element userIdentity.

Erläuterungen der Amazon EKS-Protokolldateieinträge

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion. Dazu gehören Informationen zu Datum und Uhrzeit der Aktion sowie zu den verwendeten Anfrageparameter. CloudTrail-Protokolleinträge sind kein geordnetes Stack-Trace der öffentlichen API-Aufrufe und erscheinen daher in keiner bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail-Protokolleintrag, der die Aktion `CreateCluster` demonstriert.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/username",
```

```
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "username"
  },
  "eventTime": "2018-05-28T19:16:43Z",
  "eventSource": "eks.amazonaws.com",
  "eventName": "CreateCluster",
  "awsRegion": "region-code",
  "sourceIPAddress": "205.251.233.178",
  "userAgent": "PostmanRuntime/6.4.0",
  "requestParameters": {
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-a670c2df",
        "subnet-4f8c5004"
      ]
    },
    "roleArn": "arn:aws:iam::111122223333:role/AWSServiceRoleForAmazonEKS-
CAC1G1VH3ZKZ",
    "clusterName": "test"
  },
  "responseElements": {
    "cluster": {
      "clusterName": "test",
      "status": "CREATING",
      "createdAt": 1527535003.208,
      "certificateAuthority": {},
      "arn": "arn:aws:eks:region-code:111122223333:cluster/test",
      "roleArn": "arn:aws:iam::111122223333:role/AWSServiceRoleForAmazonEKS-
CAC1G1VH3ZKZ",
      "version": "1.10",
      "resourcesVpcConfig": {
        "securityGroupIds": [],
        "vpcId": "vpc-21277358",
        "subnetIds": [
          "subnet-a670c2df",
          "subnet-4f8c5004"
        ]
      }
    }
  },
  "requestID": "a7a0735d-62ab-11e8-9f79-81ce5b2b7d37",
  "eventID": "eab22523-174a-499c-9dd6-91e7be3ff8e3",
  "readOnly": false,
```

```
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Protokolleinträge für serviceverknüpfte Amazon-EKS-Rollen

Die serviceverknüpften Amazon EKS-Rollen führen API-Aufrufe an AWS-Ressourcen aus. CloudTrail-Protokolleinträge mit `username: AWSServiceRoleForAmazonEKS` und `username: AWSServiceRoleForAmazonEKSNodegroup` werden für Aufrufe angezeigt, die von den serviceverknüpften Amazon-EKS-Rollen ausgeführt werden. Weitere Informationen zu Amazon EKS und serviceverknüpften Rollen finden Sie unter Verwendung von serviceverknüpften Rollen für Amazon EKS.

Das folgende Beispiel zeigt einen CloudTrail-Protokolleintrag, der eine `DeleteInstanceProfile`-Aktion der serviceverknüpften Rolle `AWSServiceRoleForAmazonEKSNodegroup` veranschaulicht, die im `sessionContext` vermerkt ist.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA3WHGPEZ7SJ2CW55C5:EKS",
        "arn": "arn:aws:sts::111122223333:assumed-role/
AWSServiceRoleForAmazonEKSNodegroup/EKS",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA3WHGPEZ7SJ2CW55C5",
                "arn": "arn:aws:iam::111122223333:role/aws-service-role/eks-
nodegroup.amazonaws.com/AWSServiceRoleForAmazonEKSNodegroup",
                "accountId": "111122223333",
                "userName": "AWSServiceRoleForAmazonEKSNodegroup"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-02-26T00:56:33Z"
            }
        },
        "invokedBy": "eks-nodegroup.amazonaws.com"
```

```
    },
    "eventTime": "2020-02-26T00:56:34Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "DeleteInstanceProfile",
    "awsRegion": "region-code",
    "sourceIPAddress": "eks-nodegroup.amazonaws.com",
    "userAgent": "eks-nodegroup.amazonaws.com",
    "requestParameters": {
        "instanceProfileName": "eks-11111111-2222-3333-4444-abcdef123456"
    },
    "responseElements": null,
    "requestID": "11111111-2222-3333-4444-abcdef123456",
    "eventID": "11111111-2222-3333-4444-abcdef123456",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Aktivieren der Erfassung von Auto-Scaling-Gruppenmetriken

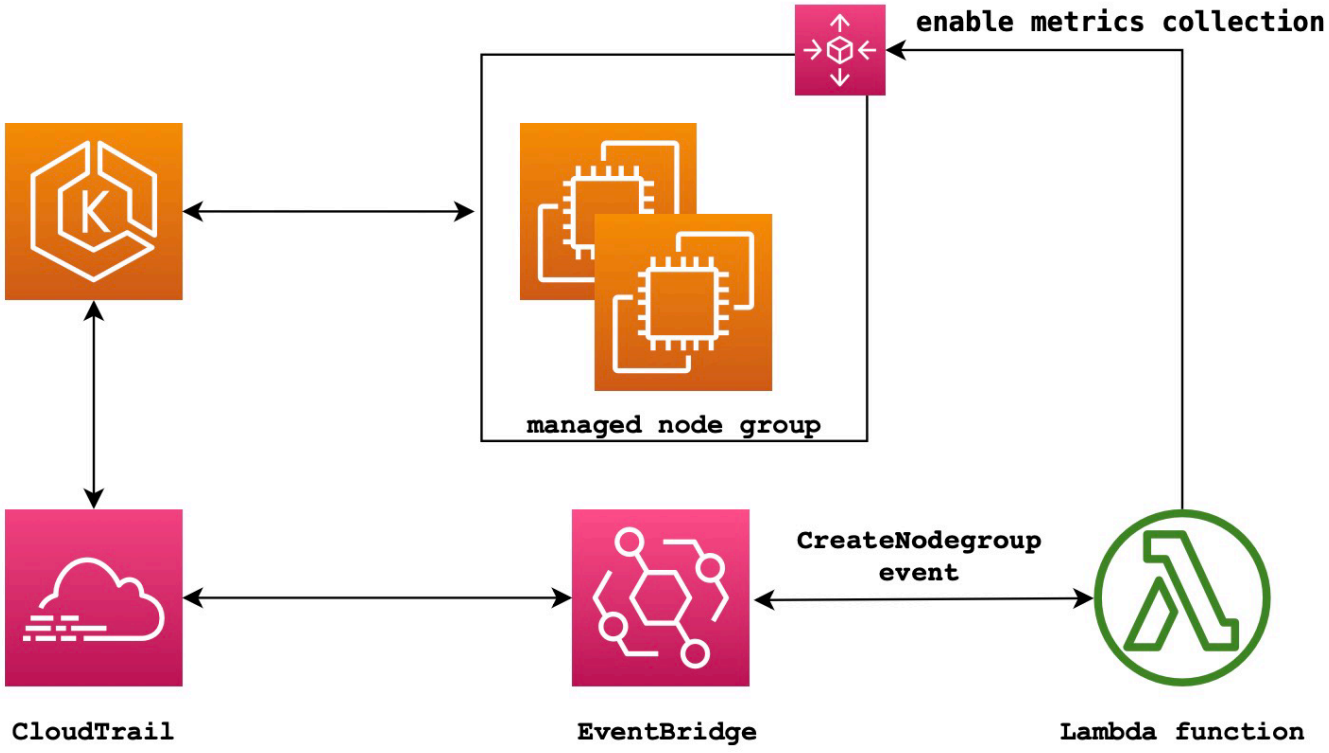
In diesem Thema wird beschrieben, wie Sie die Erfassung von Auto-Scaling-Gruppenmetriken mithilfe von AWS Lambda und AWS CloudTrail aktivieren können. Amazon EKS aktiviert nicht automatisch die Erfassung von Gruppenmetriken für Auto-Scaling-Gruppen, die für verwaltete Knoten erstellt wurden.

Sie können Auto-Scaling-Gruppenmetriken verwenden, um Änderungen in einer Auto-Scaling-Gruppe zu verfolgen und Warnungen für Schwellenwerte festzulegen. Auto Scaling-Gruppenmetriken sind in der Auto Scaling-Konsole oder der Amazon CloudWatch-Konsole verfügbar. Nach der Aktivierung sendet die Auto Scaling-Gruppe CloudWatch jede Minute Stichprobendaten an Amazon. Für die Aktivierung dieser Metriken fallen keine Gebühren an.

Durch Aktivieren der Sammlung von Auto-Scaling-Gruppenmetriken können Sie die Skalierung verwalteter Knotengruppen überwachen. Die Auto-Scaling-Gruppenmetriken zeigen die minimale, maximale und gewünschte Größe einer Auto-Scaling-Gruppe an. Sie können eine Warnung auslösen, wenn die Anzahl der Knoten in einer Knotengruppe unter die Mindestgröße fällt, was auf eine fehlerhafte Knotengruppe hindeuten würde. Das Nachverfolgen der Knotengruppengröße ist auch nützlich, um die maximale Anzahl anzupassen, damit Ihre Datenebene nicht an Kapazität verliert.

Wenn Sie eine verwaltete Knotengruppe erstellen, AWS CloudTrail sendet ein `CreateNodegroup` Ereignis an Amazon EventBridge. Indem Sie eine Amazon- EventBridge Regel erstellen, die dem

`CreateNodegroup` Ereignis entspricht, lösen Sie eine Lambda-Funktion aus, um die Erfassung von Gruppenmetriken für die Auto Scaling-Gruppe zu aktivieren, die der verwalteten Knotengruppe zugeordnet ist.



So aktivieren Sie die Erfassung von Auto-Scaling-Gruppenmetriken

1. Erstellen Sie eine IAM-Rolle für Lambda.

```
LAMBDA_ROLE=$(aws iam create-role \
  --role-name lambda-asg-enable-metrics \
  --assume-role-policy-document '{"Version": "2012-10-17","Statement":
 [{ "Effect": "Allow", "Principal": {"Service": "lambda.amazonaws.com"}, "Action":
 "sts:AssumeRole"}]}' \
  --output text \
  --query 'Role.Arn')
echo $LAMBDA_ROLE
```

2. Erstellen Sie eine Richtlinie, die das Beschreiben von Amazon EKS-Knotengruppen und das Aktivieren der Erfassung von Auto Scaling-Gruppenmetriken ermöglicht.

```
cat > /tmp/lambda-policy.json <<EOF
```

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
            "eks:DescribeNodegroup",
            "autoscaling:EnableMetricsCollection"
            ],
        "Resource": [
            "*"
        ]
      }
    ]
}
EOF
LAMBDA_POLICY_ARN=$(aws iam create-policy \
  --policy-name lambda-asg-enable-metrics-policy \
  --policy-document file:///tmp/lambda-policy.json \
  --output text \
  --query 'Policy.Arn')
echo $LAMBDA_POLICY_ARN
```

3. Fügen Sie die Richtlinie der IAM-Rolle für Lambda an.

```
aws iam attach-role-policy \
  --policy-arn $LAMBDA_POLICY_ARN \
  --role-name lambda-asg-enable-metrics
```

4. Fügen Sie die -`AWSLambdaBasicExecutionRole`verwaltete Richtlinie hinzu, die über die Berechtigungen verfügt, die die Funktion zum Schreiben von Protokollen in CloudWatch - Protokolle benötigt.

```
aws iam attach-role-policy \
  --role-name lambda-asg-enable-metrics \
  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
```

5. Erstellen Sie den Lambda-Code.

```
cat > /tmp/lambda-handler.py <<EOF
import json
import boto3
import time
```

```
import logging

eks = boto3.client('eks')
autoscaling = boto3.client('autoscaling')

logger = logging.getLogger()
logger.setLevel(logging.INFO)

def lambda_handler(event, context):
    ASG_METRICS_COLLLECTION_TAG_NAME = "ASG_METRICS_COLLLECTION_ENABLED"
    initial_retry_delay = 10
    attempts = 0

    #print(event)

    if not event["detail"]["eventName"] == "CreateNodegroup":
        print("invalid event.")
        return -1

    clusterName = event["detail"]["requestParameters"]["name"]
    nodegroupName = event["detail"]["requestParameters"]["nodegroupName"]
    try:
        metricsCollectionEnabled = event["detail"]["requestParameters"]["tags"]
[ASG_METRICS_COLLLECTION_TAG_NAME]
    except KeyError:
        print(ASG_METRICS_COLLLECTION_TAG_NAME, "tag not found.")
        return

    # Check if metrics collection is enabled in tags
    if metricsCollectionEnabled.lower() != "true":
        print("Metrics collection is not enabled in nodegroup tags.")
        return

    # Get the name of the associated autoscaling group
    print("Getting the autoscaling group name for nodegroup=", nodegroupName, ",
 cluster=", clusterName )
    for i in range(0,10):
        try:
            autoScalingGroup =
 eks.describe_nodegroup(clusterName=clusterName,nodegroupName=nodegroupName)
["nodegroup"]["resources"]["autoScalingGroups"][0]["name"]
        except:
            attempts += 1
```

```
            print("Failed to obtain the associated autoscaling group for
 nodegroup", nodegroupName, "Retrying in", initial_retry_delay*attempts,
 "seconds.")
            time.sleep(initial_retry_delay*attempts)
        else:
            break

    print("Enabling metrics collection on autoscaling group ", autoScalingGroup)

    # Enable metrics collection in the autoscaling group
    try:
        enableMetricsCollection =
 autoscaling.enable_metrics_collection(AutoScalingGroupName=autoScalingGroup,Granularity="1
    except:
        print("Unable to enable metrics collection on nodegroup=",nodegroup)
    print("Enabled metrics collection on nodegroup", nodegroupName)
EOF
```

6. Erstellen Sie ein Bereitstellungspaket.

```
cd /tmp
zip function.zip lambda-handler.py
```

7. Erstellen Sie eine Lambda-Funktion.

```
LAMBDA_ARN=$(aws lambda create-function --function-name asg-enable-metrics-
collection \
  --zip-file fileb://function.zip --handler lambda-handler.lambda_handler \
  --runtime python3.9 \
  --timeout 600 \
  --role $LAMBDA_ROLE \
  --output text \
  --query 'FunctionArn')
echo $LAMBDA_ARN
```

8. Erstellen Sie eine - EventBridge Regel.

```
RULE_ARN=$(aws events put-rule --name CreateNodegroupRuleToLambda \
  --event-pattern "{\"source\":[\"aws.eks\"],\"detail-type\":[\"AWS API Call via
 CloudTrail\"],\"detail\":{\"eventName\":[\"CreateNodegroup\"],\"eventSource\":
[\"eks.amazonaws.com\"]}}" \
  --output text \
  --query 'RuleArn')
```

```
echo $RULE_ARN
```

9. Fügen Sie die Lambda-Funktion als Ziel hinzu.

```
aws events put-targets --rule CreateNodegroupRuleToLambda \
 --targets "Id"="1","Arn"="$LAMBDA_ARN"
```

10. Fügen Sie eine Richtlinie hinzu, die es ermöglicht EventBridge , die Lambda-Funktion aufzurufen.

```
aws lambda add-permission \
  --function-name asg-enable-metrics-collection \
  --statement-id CreateNodegroupRuleToLambda \
  --action 'lambda:InvokeFunction' \
  --principal events.amazonaws.com \
  --source-arn $RULE_ARN
```

Die Lambda-Funktion aktiviert die Erfassung von Auto-Scaling-Gruppenmetriken für alle verwalteten Knotengruppen, die Sie mit `ASG_METRICS_COLLLECTION_ENABLED` auf `TRUE` markieren.
Um zu bestätigen, dass die Auto Scaling group metrics collection (Erfassung von Auto-Scaling-Gruppenmetriken) aktiviert ist, navigieren Sie zur zugeordneten Auto-Scaling-Gruppe in der Amazon-EC2-Konsole. Auf der Registerkarte Monitoring (Überwachung) sollten Sie sehen, dass das Kontrollkästchen Enable (Aktivieren) aktiviert ist.

Amazon-EKS-Add-on-Unterstützung für ADOT Operator

Amazon EKS unterstützt die Verwendung der AWS Management Console, AWS CLI sowie der Amazon EKS API zur Installation und Verwaltung von AWS-Distro for OpenTelemetry (ADOT)-Operator. Damit können Sie einfacher aktivieren, dass Ihre Anwendungen, die auf Amazon EKS ausgeführt werden, Metrik- und Trace-Daten an mehrere Überwachungsservices wie Amazon CloudWatch, Prometheus und X-Ray senden.

Weitere Informationen finden Sie unter Erste Schritte mit AWS Distro für OpenTelemetry unter Verwendung von EKS-Add-ons in der Dokumentation zu AWS Distro für OpenTelemetry.

Weitere in Amazon EKS integrierte AWS Services

Zusätzlich zu den in anderen Abschnitten behandelten Services arbeitet Amazon EKS mit weiteren AWS Services zusammen, um zusätzliche Lösungen bereitzustellen. In diesem Thema werden einiger der weiteren Services aufgeführt, die mithilfe von Amazon EKS ihre Funktionalität erweitern, sowie Services, die Amazon EKS zur Ausführung von Aufgaben nutzt.

Themen

Erstellen von Amazon-EKS-Ressourcen mit AWS CloudFormation

Amazon EKS ist in AWS CloudFormation integriert. Dies ist ein Service, der Ihnen hilft, Ihre AWS-Ressourcen zu modellieren und einzurichten, so dass Sie weniger Zeit für die Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur aufwenden müssen. Sie erstellen eine Vorlage, in der alle gewünschten AWS-Ressourcen, wie etwa ein Amazon EKS-Cluster, beschrieben werden, und AWS CloudFormation übernimmt die Bereitstellung und Konfigurierung dieser Ressourcen für Sie.

Wenn Sie AWS CloudFormation verwenden, können Sie Ihre Vorlage wiederverwenden, um Ihre Amazon-EKS-Ressourcen einheitlich und wiederholt einzurichten. Sie beschreiben Ihre Ressourcen dann einmal und können die gleichen Ressourcen dann in mehreren AWS-Konten und -Regionen immer wieder bereitstellen.

Amazon-EKS- und AWS CloudFormation-Vorlagen

Um Ressourcen für Amazon EKS und verwandte Dienstleistungen bereitzustellen und zu konfigurieren, müssen Sie AWS CloudFormation-Vorlagen kennen und verstehen. Vorlagen sind

formatierte Textdateien in JSON oder YAML. Diese Vorlagen beschreiben die Ressourcen, die Sie in Ihren AWS CloudFormation-Stacks bereitstellen möchten. Wenn Sie noch keine Erfahrungen mit JSON oder YAML haben, können Sie AWS CloudFormation Designer verwenden, der den Einstieg in die Arbeit mit AWS CloudFormation-Vorlagen erleichtert. Weitere Informationen finden Sie unter Was ist AWS CloudFormation-Designer? im AWS CloudFormation-Benutzerhandbuch.

Amazon EKS unterstützt das Erstellen von Clustern und Knotengruppen in AWS CloudFormation. Weitere Informationen, einschließlich Beispiele für JSON- und YAML-Vorlagen für Ihre Amazon-EKS-Ressourcen, finden Sie unter Amazon-EKS-Ressourcentypreferenz im AWS CloudFormation-Handbuch.

Weitere Informationen zu AWS CloudFormation

Weitere Informationen zu AWS CloudFormation finden Sie in den folgenden Ressourcen.

- AWS CloudFormation
- AWS CloudFormation-Benutzerhandbuch
- AWS CloudFormation-Benutzerhandbuch für die Befehlszeilenschnittstelle

Amazon EKS und AWS Local Zones

Eine lokale AWS-Zone ist die Erweiterung einer AWS-Region in geografischer Nähe zu Ihren Benutzern. Local Zones haben ihre eigenen Verbindungen mit dem Internet und unterstützen AWS Direct Connect. Ressourcen, die in einer Local Zone erstellt wurden, können von lokalen Benutzern mit sehr latenzarmen Verbindungen genutzt werden. Weitere Informationen finden Sie unter Local Zones.

Amazon EKS unterstützt bestimmte Ressourcen in Local Zones. Dies umfasst selbstverwaltete Amazon-EC2-Knoten, Amazon-EBS-Volumes und Application Load Balancers (ALBs). Es wird empfohlen, Folgendes zu berücksichtigen, wenn Sie Local Zones als Teil Ihres Amazon-EKS-Clusters verwenden.

Knoten

Sie können keine verwalteten Knotengruppen oder Fargate-Knoten in Local Zones mit Amazon EKS erstellen. Sie können jedoch selbstverwaltete Amazon-EC2-Knoten in Local Zones erstellen, indem Sie die Amazon-EC2-API, AWS CloudFormation oder `eksctl` verwenden. Weitere Informationen finden Sie unter Selbstverwaltete Knoten.

-Netzwerkarchitektur

- Die von Amazon EKS verwaltete Kubernetes-Steuerebene wird immer in der AWS-Region ausgeführt. Die von Amazon EKS verwaltete Kubernetes-Steuerungsebene kann nicht in der lokalen Zone ausgeführt werden. Da Local Zones in Ihrer VPC als Subnetz angezeigt werden, sieht Kubernetes Ihre lokalen Zonenressourcen als Teil dieses Subnetzes.
- Der Cluster von Amazon EKS Kubernetes kommuniziert mit den Amazon-EC2-Instances, die Sie in der AWS-Region oder der lokalen Zone mit von Amazon EKS verwalteten Elastic-Network-Schnittstellen ausführen. Weitere Informationen zur Amazon EKS Network-Architektur finden Sie unterAmazon-EKS-Netzwerkaus.
- Im Gegensatz zu regionalen Subnetzen kann Amazon EKS keine Netzwerkschnittstellen in Ihre lokalen Zonen-Subnetze platzieren. Das bedeutet, dass Sie beim Erstellen des Clusters keine Subnetze der lokalen Zone angeben dürfen.

Deep Learning Containers

AWS-Deep-Learning-Containers sind Docker-Images für Trainings- und Servicing-Modelle in TensorFlow auf Amazon EKS und Amazon Elastic Container Service (Amazon ECS). Deep-Learning-Container bieten optimierte Umgebungen mit TensorFlow, NVIDIA CUDA (für GPU-Instances) und Intel-MKL-Bibliotheken (für CPU-Instances). Sie sind in Amazon ECR verfügbar.

Informationen zu den ersten Schritten mit AWS-Deep-Learning-Containers in Amazon EKS finden Sie unter Setup in Amazon EKS im AWSDeep Learning Container-Entwicklerhandbuch.

Amazon VPC Lattice

Amazon VPC Lattice ist ein vollständig verwalteter Anwendungsnetzwerkservice, der direkt in die AWS-Netzwerkinfrastruktur integriert ist und den Sie verwenden können, um Ihre Services über mehrere Konten und Virtual Private Clouds (VPCs) hinweg zu verbinden, zu schützen und zu überwachen. Mit Amazon EKS können Sie Amazon VPC Lattice mithilfe des AWS Gateway API Controller, einer Implementierung der Kubernetes Gateway-API, nutzen. Mit Amazon VPC Lattice können Sie auf einfache und konsistente Weise clusterübergreifende Konnektivität mit Kubernetes-Standardsemantik einrichten. Informationen zum Einstieg in die Verwendung von Amazon VPC Lattice mit Amazon EKS finden Sie im AWS-Gateway-API-Controller-Benutzerhandbuch.

AWS Resilience Hub

AWS Resilience Hub bewertet die Resilienz eines Amazon-EKS-Clusters durch Analyse seiner Infrastruktur. AWS Resilience Hub verwendet die Konfiguration der Kubernetes-rollenbasierten Zugriffskontrolle (RBAC) zur Bewertung der in Ihrem Cluster bereitgestellten Kubernetes-Workloads. Weitere Informationen finden Sie unter Aktivieren des AWS Resilience Hub-Zugriffs auf Ihr Amazon-EKS-Cluster im AWS Resilience Hub-Benutzerhandbuch.

Erkennen Sie Bedrohungen mit Amazon GuardDuty

Amazon GuardDuty ist ein Service zur Bedrohungserkennung, der Ihnen hilft, Ihre Konten, Container, Workloads und die Daten in Ihrer AWS Umgebung zu schützen. Mithilfe von Modellen für maschinelles Lernen (ML) und Funktionen zur Erkennung von Anomalien und Bedrohungen werden GuardDuty kontinuierlich verschiedene Protokollquellen und Laufzeitaktivitäten überwacht, um potenzielle Sicherheitsrisiken und böswillige Aktivitäten in Ihrer Umgebung zu identifizieren und zu priorisieren.

GuardDuty Bietet unter anderem die folgenden beiden Funktionen zur Erkennung potenzieller Bedrohungen für Ihre EKS-Cluster: EKS-Schutz und Runtime Monitoring.

EKS-Schutz

Diese Funktion deckt die Bedrohungserkennung ab und hilft Ihnen, Amazon EKS-Cluster zu schützen, indem die zugehörigen KubernetesAudit-Protokolle überwacht werden. KubernetesAudit-Logs erfassen sequenzielle Aktionen innerhalb Ihres Clusters, einschließlich Aktivitäten von Benutzern, Anwendungen, die die Kubernetes API verwenden, und der Kontrollebene. So GuardDuty kann beispielsweise festgestellt werden, dass APIs, die aufgerufen wurden, um potenziell Ressourcen in einem Kubernetes Cluster zu manipulieren, von einem nicht authentifizierten Benutzer aufgerufen wurden.

Wenn Sie EKS Protection aktivieren, können GuardDuty Sie nur zur kontinuierlichen Erkennung von Bedrohungen auf Ihre Amazon EKS-Auditprotokolle zugreifen. Wenn eine potenzielle Bedrohung für Ihren Cluster GuardDuty identifiziert wird, generiert es ein zugehöriges Kubernetes Audit-Log-Ergebnis eines bestimmten Typs. Weitere Informationen zu den Arten von Ergebnissen, die anhand von Kubernetes Audit-Logs verfügbar sind, finden Sie unter Kubernetes Audit-Logs — Finding Types im GuardDuty Amazon-Benutzerhandbuch.

Weitere Informationen finden Sie unter EKS-Schutz im GuardDuty Amazon-Benutzerhandbuch.

Laufzeit-Überwachung

Diese Funktion überwacht und analysiert Ereignisse auf Betriebssystemebene, Netzwerk- und Dateiereignisse, um Ihnen zu helfen, potenzielle Bedrohungen bei bestimmten AWS Workloads in Ihrer Umgebung zu erkennen.

Wenn Sie Runtime Monitoring aktivieren und den GuardDuty Agenten in Ihren Amazon EKS-Clustern installieren, GuardDuty beginnt die Überwachung der Runtime-Ereignisse, die mit diesem Cluster verknüpft sind. Wenn eine potenzielle Bedrohung für Ihren Cluster GuardDuty identifiziert wird, wird ein zugehöriges Runtime Monitoring-Ergebnis generiert. Beispielsweise kann eine Bedrohung möglicherweise damit beginnen, dass ein einzelner Container kompromittiert wird, auf dem eine anfällige Webanwendung ausgeführt wird. Diese Webanwendung verfügt möglicherweise über Zugriffsberechtigungen für die zugrunde liegenden Container und Workloads. In diesem Szenario könnten falsch konfigurierte Anmeldeinformationen möglicherweise zu einem umfassenderen Zugriff auf das Konto und die darin gespeicherten Daten führen.

Um Runtime Monitoring zu konfigurieren, installieren Sie den GuardDuty Agenten als Amazon EKS-Add-on in Ihrem Cluster. Weitere Informationen zum Add-on finden Sie unterVerfügbare Amazon-EKS-Add-Ons von Amazon EKS.

Weitere Informationen finden Sie unter Runtime Monitoring im GuardDuty Amazon-Benutzerhandbuch.

Verwenden von Amazon Security Lake mit Amazon EKS

Amazon Security Lake ist ein vollständig verwalteter Security Data Lake-Service, mit dem Sie Sicherheitsdaten aus verschiedenen Quellen, einschließlich Amazon EKS, zentralisieren können. Durch die Integration von Amazon EKS in Security Lake können Sie tiefere Einblicke in die Aktivitäten gewinnen, die auf Ihren Kubernetes Ressourcen ausgeführt werden, und die Sicherheitslage Ihrer Amazon EKS-Cluster verbessern.

> **Note**
>
> Weitere Informationen zur Verwendung von Security Lake mit Amazon EKS und zur Einrichtung von Datenquellen finden Sie in der Amazon Security Lake-Dokumentation.

Vorteile der Verwendung von Security Lake mit Amazon Amazon EKS

Zentralisierte Sicherheitsdaten — Security Lake sammelt und zentralisiert automatisch Sicherheitsdaten aus Ihren Amazon EKS-Clustern zusammen mit Daten von anderen AWS Diensten, SaaS-Anbietern, lokalen Quellen und Quellen von Drittanbietern. Dies bietet einen umfassenden Überblick über Ihre Sicherheitslage in Ihrem gesamten Unternehmen.

Standardisiertes Datenformat — Security Lake konvertiert die gesammelten Daten in das Open Cybersecurity Schema Framework (OCSF) -Format, bei dem es sich um ein Standard-Open-Source-Schema handelt. Diese Normalisierung ermöglicht eine einfachere Analyse und Integration mit anderen Sicherheitstools und -diensten.

Verbesserte Bedrohungserkennung — Durch die Analyse der zentralen Sicherheitsdaten, einschließlich der Protokolle der Amazon EKS-Kontrollebene, können Sie potenziell verdächtige Aktivitäten in Ihren Amazon EKS-Clustern effektiver erkennen. Dies hilft dabei, Sicherheitsvorfälle zu identifizieren und umgehend darauf zu reagieren.

Vereinfachtes Datenmanagement — Security Lake verwaltet den Lebenszyklus Ihrer Sicherheitsdaten mit anpassbaren Aufbewahrungs- und Replikationseinstellungen. Dies vereinfacht die Datenverwaltungsaufgaben und stellt sicher, dass Sie die für Compliance- und Auditzwecke erforderlichen Daten aufbewahren.

Security Lake für Amazon EKS aktivieren

Gehen Sie wie folgt vor, um Security Lake mit Amazon EKS zu verwenden:

1. Aktivieren Sie die Protokollierung der Amazon EKS-Kontrollebene für Ihre EKS-Cluster. Ausführliche Anweisungen finden Sie unter Aktivieren und Deaktivieren von Protokollen auf Kontrollebene.
2. Fügen Sie Amazon EKS Audit Logs als Quelle in Security Lake hinzu. Security Lake beginnt dann mit der Erfassung detaillierter Informationen über die Aktivitäten, die auf den Kubernetes-Ressourcen ausgeführt werden, die in Ihren EKS-Clustern ausgeführt werden.
3. Konfigurieren Sie die Aufbewahrungs- und Replikationseinstellungen für Ihre Sicherheitsdaten in Security Lake entsprechend Ihren Anforderungen.
4. Verwenden Sie die in Security Lake gespeicherten normalisierten OCSF-Daten für die Reaktion auf Vorfälle, Sicherheitsanalysen und die Integration mit anderen AWS Diensten oder Tools von Drittanbietern. Beispielsweise können Sie mithilfe von Amazon Ingestion Sicherheitsinformationen aus Amazon Security Lake-Daten generieren. OpenSearch

Analysieren von EKS-Protokollen in Security Lake

Security Lake normalisiert EKS-Protokollereignisse auf das OCSF-Format, wodurch es einfacher wird, die Daten zu analysieren und mit anderen Sicherheitsereignissen zu korrelieren. Sie können verschiedene Tools und Dienste wie Amazon Athena, Amazon oder Sicherheitsanalysetools von Drittanbietern verwenden QuickSight, um die normalisierten Daten abzufragen und zu visualisieren.

Weitere Informationen zur OCSF-Zuordnung für EKS-Protokollereignisse finden Sie in der Zuordnungsreferenz im OCSF-Repository. GitHub

Amazon Detective

Amazon Detective hilft Ihnen, die Ursache von Sicherheitserkenntnissen oder verdächtigen Aktivitäten zu analysieren, zu untersuchen und schnell zu identifizieren. Detective sammelt automatisch Protokolldaten von Ihren AWS Ressourcen. Es verwendet dann Machine Learning, statistische Analysen und die Diagrammtheorie, um Visualisierungen zu erstellen, mit denen Sie effektive Sicherheitsuntersuchungen schneller und effizienter durchführen können. Detective bietet vordefinierte Datenaggregationen, Übersichten und Kontexte, mit denen Sie Art und Ausmaß möglicher Sicherheitsprobleme schnell analysieren und feststellen können. Weitere Informationen finden Sie im Benutzerhandbuch von Amazon Detective.

Detective organisiert Kubernetes und verarbeitet AWS Daten zu Ergebnissen wie:

- Amazon EKS-Clusterdetails, einschließlich der IAM-Identität, mit der der Cluster erstellt wurde, und der Servicerolle des Clusters. Sie können die AWS und die Kubernetes API-Aktivität dieser IAM-Identitäten mit Detective untersuchen.
- Container-Details, wie das Image und der Sicherheitskontext. Sie können sich die Angaben auch für beendete Pods ansehen.
- Kubernetes-API-Aktivität, einschließlich allgemeiner Trends bei der API-Aktivität und Details zu bestimmten API-Aufrufen. Sie können beispielsweise die Anzahl der erfolgreichen und fehlgeschlagenen Kubernetes-API-Aufrufe anzeigen, die in einem ausgewählten Zeitraum ausgegeben wurden. Darüber hinaus kann der Abschnitt über neu beobachtete API-Aufrufe hilfreich sein, um verdächtige Aktivitäten zu identifizieren.

Amazon EKS Audit Logs sind ein optionales Datenquellenpaket, das Ihrem Detective-Verhaltensdiagramm hinzugefügt werden kann. Sie können die verfügbaren optionalen Quellpakete

und deren Status in Ihrem Konto einsehen. Weitere Informationen finden Sie unter Amazon-EKS-Auditprotokolle für Detective im Benutzerhandbuch von Amazon Detective.

Verwenden Sie Amazon Detective mit Amazon EKS

Ergebnisse für einen Amazon-EKS-Cluster überprüfen

Bevor Sie die Ergebnisse überprüfen können, muss Detective für mindestens 48 Stunden in derselben Umgebung aktiviert sein AWS-Region , in der sich Ihr Cluster befindet. Weitere Informationen finden Sie unter Einrichten von Amazon Detective im Benutzerhandbuch von Amazon Detective.

1. Öffnen Sie die Detective-Konsole unter https://console.aws.amazon.com/detective/.
2. Wählen Sie Search (Suche) im linken Navigationsbereich.
3. Wählen Sie Choose type (Typ auswählen) und anschließend EKS cluster (EKS-Cluster) aus.
4. Geben Sie den Clusternamen oder ARN ein und wählen Sie dann Search (Suchen) aus.
5. Wählen Sie in den Suchergebnissen den Namen des Clusters aus, für den Sie Aktivitäten anzeigen möchten. Weitere Informationen darüber, was Sie sich ansehen können, finden Sie unter Allgemeine Kubernetes-API-Aktivität mit einem Amazon EKS-Cluster im Benutzerhandbuch von Amazon Detective.

Amazon-EKS-Fehlerbehebung

In diesem Kapitel werden einige häufige Fehler bei der Verwendung von Amazon EKS sowie deren Lösung behandelt. Wenn Sie Probleme in bestimmten Amazon-EKS-Bereichen beheben müssen, lesen Sie die separaten Themen Fehlersuche bei IAM, Beheben von Problemen in Amazon EKS Connector und Fehlerbehebung für ADOT mithilfe von EKS-Add-ons.

Weitere Informationen zur Fehlerbehebung finden Sie in den Knowledge Center-Inhalten zu Amazon Elastic Kubernetes Service unter AWS re:Post.

Unzureichende Kapazität

Wenn beim Versuch, einen Amazon-EKS-Cluster zu erstellen, der folgende Fehler auftritt, verfügt eine der angegebenen Availability Zones nicht über ausreichende Kapazität, um einen Cluster zu unterstützen.

```
Cannot create cluster 'example-cluster' because region-1d, the targeted
Availability Zone, does not currently have sufficient capacity to support
the cluster. Retry and choose from these Availability Zones: region-1a,
region-1b, region-1c
```

Versuchen Sie erneut, den Cluster mit Subnetzen in Ihrer Cluster-VPC zu erstellen, die in den Availability Zones gehostet werden, die diesen Fehler verursacht haben.

Es gibt Availability Zones, in denen sich kein Cluster befinden darf. Vergleichen Sie die Availability Zones, in denen sich Ihre Subnetze befinden, mit der Liste der Availability Zones unter Subnetz-Anforderungen und -Überlegungen.

Knoten können nicht mit dem Cluster verknüpft werden

Es gibt zwei häufige Gründe, aus denen Knoten nicht mit dem Cluster verknüpft werden können:

- Wenn es sich bei den Knoten um verwaltete Knoten handelt, fügt Amazon EKS Einträge zu `aws-auth ConfigMap` hinzu, wenn Sie die Knotengruppe erstellen. Wenn der Eintrag entfernt oder geändert wurde, müssen Sie ihn erneut hinzufügen. Weitere Informationen erhalten Sie durch Eingabe von **`eksctl create iamidentitymapping --help`** im Terminal. Sie können Ihre aktuellen `aws-auth ConfigMap`-Einträge anzeigen, indem Sie im folgenden Befehl *my-*

cluster durch den Namen Ihres Clusters ersetzen und dann den geänderten Befehl ausführen: **eksctl get iamidentitymapping --cluster *my-cluster***. Der ARN der von Ihnen angegebenen Rolle darf als Pfad nur / enthalten. Wenn der Name Ihrer Rolle also beispielsweise `development/apps/my-role` lautet, müssen Sie ihn bei Angabe des ARN für die Rolle in `my-role` ändern. Achten Sie darauf, den ARN der IAM-Rolle des Knotens (nicht den ARN des Instances-Profils) anzugeben.

Wenn es sich um selbstverwaltete Knoten handelt und Sie keine Zugriffseinträge für den ARN der IAM-Rolle des Knotens erstellt haben, führen Sie die gleichen Befehle aus, die für verwaltete Knoten aufgeführt sind. Wenn Sie einen Zugriffseintrag für den ARN für die IAM-Rolle Ihres Knotens erstellt haben, ist er im Zugriffseintrag möglicherweise nicht richtig konfiguriert. Achten Sie darauf, dass in Ihrem `aws-auth ConfigMap`-Eintrag oder Zugriffseintrag der ARN der IAM-Rolle des Knotens (nicht der ARN des Instances-Profils) als Haupt-ARN angegeben ist. Weitere Informationen zu Zugriffseinträgen finden Sie unter Zugangseinträge verwalten.

- Die Vorlage ClusterNamein Ihrer AWS CloudFormation Knotenvorlage entspricht nicht genau dem Namen des Clusters, dem Ihre Knoten beitreten sollen. Wenn Sie einen falschen Wert an dieses Feld übergeben, führt dies zu Fehlern in der Konfiguration der `/var/lib/kubelet/kubeconfig`-Datei des Knotens und dieser kann nicht mit dem Cluster verknüpft werden.
- Der Knoten ist nicht als Eigentum des Clusters gekennzeichnet. Auf Ihre Worker-Knoten muss die folgende Markierung angewendet werden, wobei *my-cluster* durch den Namen des Clusters ersetzt wird.

Schlüssel	Wert
`kubernetes.io/cluster/` *my-cluster*	owned

- Die Knoten können möglicherweise nicht über eine öffentliche IP-Adresse auf den Cluster zugreifen. Stellen Sie sicher, dass den Knoten, die in öffentlichen Subnetzen bereitgestellt werden, eine öffentliche IP-Adresse zugewiesen wird. Ist dies nicht der Fall, können Sie einem Knoten eine Elastic-IP-Adresse nach dem Start zuordnen. Weitere Informationen dazu finden Sie unter Zuordnen einer Elastic-IP-Adresse zu einer laufenden Instance oder einer Netzwerkschnittstelle. Wenn das öffentliche Subnetz nicht so eingestellt ist, dass es Instances, die es nutzen, automatisch öffentliche IP-Adressen zuweist, empfehlen wir, diese Einstellung zu aktivieren. Weitere Informationen finden Sie unter Ändern des öffentlichen `IPv4`-Adressierungsattributs für Ihr Subnetz. Wenn der Arbeitsknoten in einem privaten Subnetz bereitgestellt wird, muss das Subnetz über eine Route zu einem NAT-Gateway verfügen, dem eine öffentliche IP-Adresse zugewiesen ist.

- Der AWS STS Endpunkt für den AWS-Region , auf dem Sie die Knoten bereitstellen, ist für Ihr Konto nicht aktiviert. Informationen zum Aktivieren der Region finden Sie unter Aktivieren und Deaktivieren AWS STS in einem AWS-Region.
- Der Knoten hat keinen privaten DNS-Eintrag, was dazu führt, dass das `kubelet`-Protokoll einen `node "" not found`-Fehler enthält. Stellen Sie sicher, dass für die VPC, in der der Knoten erstellt wird, Werte für `domain-name` und `domain-name-servers` als `Options` in einem `DHCP options set` festgelegt sind. Die Standard-Werte sind `domain-name:<region>.compute.internal` und `domain-name-servers:AmazonProvidedDNS`. Weitere Informationen finden Sie unter DHCP-Optionssätze im Amazon-VPC-Benutzerhandbuch.
- Wenn die Knoten in der verwalteten Knotengruppe nicht innerhalb von 15 Minuten eine Verbindung zum Cluster herstellen, wird das Problem „NodeCreationFehler“ gemeldet und der Konsolenstatus wird auf `Create failed` gesetzt. Bei Windows AMIs mit langsamen Startzeiten kann dieses Problem mithilfe von Fast Launch behoben werden.

Zur Ermittlung und Behebung häufiger Ursachen, die den Beitritt von Worker-Knoten zu einem Cluster verhindern, können Sie das Runbook `AWSSupport-TroubleshootEKSWorkerNode` verwenden. Weitere Informationen finden Sie unter `AWSSupport-TroubleshootEKSWorkerNode` in der Referenz zum Automation-Runbook für AWS Systems Manager .

Nicht autorisiert oder Zugriff verweigert (**`kubectl`**)

Wenn Sie beim Ausführen von `kubectl`-Befehlen einen der folgenden Fehler erhalten, ist `kubectl` nicht korrekt für Amazon EKS konfiguriert oder die Anmeldeinformationen für den von Ihnen verwendeten IAM-Prinzipal (Rolle oder Benutzer) sind keinem Kubernetes-Benutzernamen mit ausreichenden Berechtigungen für Kubernetes-Objekte in Ihrem Amazon EKS-Cluster zugeordnet.

- `could not get token: AccessDenied: Access denied`
- `error: You must be logged in to the server (Unauthorized)`
- `error: the server doesn't have a resource type "svc"`

Das kann auf eine der folgenden Ursachen zurückzuführen sein:

- Der Cluster wurde mit Anmeldeinformationen für einen bestimmten IAM-Prinzipal erstellt und `kubectl` ist für die Verwendung von Anmeldeinformationen für einen anderen IAM-Prinzipal konfiguriert. Aktualisieren Sie in diesem Fall die Datei `kube config` mit den Anmeldeinformationen, mit denen der Cluster erstellt wurde, um das Problem zu beheben. Weitere

Informationen finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

- Falls Ihr Cluster die Mindestanforderungen an die Plattform erfüllt, die unter Zugangseinträge verwalten im Abschnitt mit den Voraussetzungen angegeben sind, ist kein Zugriffseintrag mit Ihrem IAM-Prinzipal vorhanden. Falls einer vorhanden ist, sind für ihn nicht die erforderlichen Kubernetes-Gruppennamen definiert oder ihm ist nicht die richtige Zugriffsrichtlinie zugeordnet. Weitere Informationen finden Sie unter Zugangseinträge verwalten.
- Falls Ihr Cluster die unter Zugangseinträge verwalten angegebenen Mindestanforderungen an die Plattform nicht erfüllt, ist in `aws-auth ConfigMap` kein Eintrag mit Ihrem IAM-Prinzipal vorhanden. Falls einer vorhanden ist, ist er keinen Kubernetes-Gruppennamen zugeordnet, die an ein Kubernetes-Element vom Typ `Role` oder `ClusterRole` mit den erforderlichen Berechtigungen gebunden sind. Weitere Informationen zu Kubernetes-Objekten für die rollenbasierte Autorisierung (RBAC) finden Sie in der Dokumentation zu Kubernetes unter Using RBAC Authorization. Sie können Ihre aktuellen `aws-auth ConfigMap`-Einträge anzeigen, indem Sie im folgenden Befehl *`my-cluster`* durch den Namen Ihres Clusters ersetzen und dann den geänderten Befehl ausführen: **`eksctl get iamidentitymapping --cluster my-cluster`**. Wenn in `ConfigMap` kein Eintrag mit dem ARN Ihres IAM-Prinzipals enthalten ist, geben Sie in Ihrem Terminal **`eksctl create iamidentitymapping --help`** ein, um zu erfahren, wie Sie einen solchen Eintrag erstellen.

Wenn Sie das installieren und konfigurieren AWS CLI, können Sie die von Ihnen verwendeten IAM-Anmeldeinformationen konfigurieren. Weitere Informationen finden Sie unter Konfigurieren der AWS CLI im AWS Command Line Interface -Leitfaden. Sie können auch `kubectl` für die Verwendung einer IAM-Rolle konfigurieren, wenn Sie für den Zugriff auf Kubernetes-Objekte in Ihrem Cluster eine IAM-Rolle annehmen. Weitere Informationen finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

`hostname doesn't match`

Die Python-Version Ihres Systems muss `2.7.9` oder höher sein. Andernfalls erhalten Sie `hostname doesn't match` Fehler bei AWS CLI Aufrufen von Amazon EKS. Weitere Informationen finden Sie auf der Seite mit häufig gestellten Fragen der Python Requests-Website unter What are “hostname doesn’t match” errors?.

`getsockopt: no route to host`

Docker wird im `172.17.0.0/16`-CIDR-Bereich in Amazon-EKS-Clustern ausgeführt. Wir empfehlen, dass die VPC-Subnetze Ihres Clusters diesen Bereich nicht überschneiden. Andernfalls erhalten Sie den folgenden Fehler:

```
Error: : error upgrading connection: error dialing backend: dial tcp
 172.17.<nn>.<nn>:10250: getsockopt: no route to host
```

`Instances failed to join the Kubernetes cluster`

Wenn Sie den Fehler `Instances failed to join the Kubernetes cluster` in der erhalten, stellen Sie sicher AWS Management Console, dass entweder der private Endpunktzugriff des Clusters aktiviert ist oder dass Sie die CIDR-Blöcke für den Zugriff auf öffentliche Endpunkte korrekt konfiguriert haben. Weitere Informationen finden Sie unter Zugriffskontrolle für den Amazon-EKS-Cluster-Endpunkt.

Fehlercodes bei verwalteten Knotengruppen

Wenn bei Ihrer verwaltete Knotengruppe ein Hardwarezustandsproblem auftritt, gibt Amazon EKS einen Fehlercode zurück, der Ihnen bei der Diagnose des Problems hilft. Diese Zustandsprüfungen erkennen keine Softwareprobleme, da sie auf Amazon-EC2-Zustandsprüfungen basiert sind. Die Fehlercodes sind in der folgenden Liste beschrieben.

AccessDenied

> Amazon EKS oder mindestens einer Ihrer verwalteten Knoten kann nicht mit Ihrem Kubernetes-Cluster-API-Server authentifizieren oder autorisieren. Weitere Informationen zum Beheben einer häufigen Ursache finden Sie unter Behebung einer häufigen Ursache für `AccessDenied`-Fehler bei verwalteten Knotengruppen. Neben der Fehlermeldung `Not authorized for images` können private Windows-AMIs auch diesen Fehlercode verursachen. Weitere Informationen finden Sie unter `Not authorized for images`.

AmiIdNotFound

> Die AAMI-ID, die Ihrer Startvorlage zugeordnet ist, konnte nicht gefunden werden. Stellen Sie sicher, dass das AMI vorhanden ist und für Ihr Konto freigegeben ist.

AutoScalingGroupNotGefunden

Die Auto-Scaling-Gruppe, die der verwalteten Knotengruppe zugeordnet ist, konnte nicht gefunden werden. Möglicherweise können Sie eine Auto-Scaling-Gruppe mit den gleichen Einstellungen zur Wiederherstellung erstellen.

ClusterUnreachable

Amazon EKS oder ein oder mehrere Ihrer verwalteten Knoten kann nicht mit Ihrem Kubernetes-Cluster-API-Server kommunizieren. Dies kann passieren, wenn Netzwerkunterbrechungen auftreten oder wenn API-Server eine Zeitüberschreitung für die Verarbeitung von Anforderungen vornehmen.

Ec2 SecurityGroup NotFound

Die Cluster-Sicherheitsgruppe für den Cluster konnte nicht gefunden werden. Sie müssen Ihren Cluster neu erstellen.

Ec2 SecurityGroup DeletionFailure

Die RAS-Sicherheitsgruppe für Ihre verwaltete Knotengruppe konnte nicht gelöscht werden. Entfernen Sie alle Abhängigkeiten aus der Sicherheitsgruppe.

Ec2 LaunchTemplate NotFound

Die Amazon-EC2-Startvorlage für Ihre verwaltete Knotengruppe konnte nicht gefunden werden. Sie müssen Ihre Knotengruppe neu erstellen, um sie wiederherzustellen.

Ec2 LaunchTemplate VersionMismatch

Die Amazon-EC2-Startvorlagenversion für Ihre verwaltete Knotengruppe stimmt nicht mit der von Amazon EKS erstellten Version überein. Möglicherweise können Sie zu der Version zurückkehren, die Amazon EKS für die Wiederherstellung erstellt hat.

IamInstanceProfileNotGefunden

Das IAM-Instance-Profil für Ihre verwaltete Knotengruppe konnte nicht gefunden werden. Möglicherweise können Sie erneut ein Instance-Profil mit den gleichen Einstellungen zur Wiederherstellung erstellen.

IamNodeRoleNotGefunden

Die IAM-Rolle für Ihre verwaltete Knotengruppe konnte nicht gefunden werden. Möglicherweise können Sie erneut eine IAM-Rolle mit den gleichen Einstellungen zur Wiederherstellung erstellen.

AsgInstanceLaunchFailures

Beim Versuch, Instances zu starten, treten in Ihrer Auto-Scaling-Gruppe Fehlfunktionen auf.

NodeCreationFehlschlag

Ihre gestarteten Instances können sich nicht bei Ihrem Amazon-EKS-Cluster registrieren. Häufige Ursachen für diesen Fehler sind unzureichende Knoten-IAM-Rollen-Berechtigungen oder fehlender ausgehender Internetzugriff für die Knoten. Ihre Knoten müssen eine der folgenden Anforderungen erfüllen:

- Kann über eine öffentliche IP-Adresse auf das Internet zugreifen. Die Sicherheitsgruppe, die mit dem Subnetz verknüpft ist, in dem sich der Knoten befindet, muss die Kommunikation zulassen. Weitere Informationen finden Sie unter Subnetz-Anforderungen und -Überlegungen und Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe.
- Ihre Knoten und Ihre VPC müssen die Anforderungen in Anforderungen an private Cluster erfüllen.

InstanceLimitÜbertroffen

Ihr AWS Konto kann keine weiteren Instances des angegebenen Instance-Typs starten. Möglicherweise können Sie eine Erhöhung des Amazon-EC2-Instance-Limits zur Wiederherstellung anfordern.

InsufficientFreeAdressen

Mindestens eines der Subnetze, die Ihrer verwalteten Knotengruppe zugeordnet sind, verfügt nicht über genügend verfügbare IP-Adressen für neue Knoten.

InternalFailure

Diese Fehler werden normalerweise durch ein serverseitiges Amazon-EKS-Problem verursacht.

Behebung einer häufigen Ursache für **`AccessDenied`**-Fehler bei verwalteten Knotengruppen

Die häufigste Ursache für `AccessDenied`-Fehler beim Ausführen von Vorgängen auf verwalteten Knotengruppen ist das Fehlen von `eks:node-manager`, `ClusterRole` oder `ClusterRoleBinding`. Amazon EKS richtet diese Ressourcen in Ihrem Cluster als Teil des Onboarding mit verwalteten Knotengruppen ein, die für die Verwaltung der Knotengruppen erforderlich sind.

Die `ClusterRole` kann sich im Laufe der Zeit ändern, sollte aber dem folgenden Beispiel ähneln:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: eks:node-manager
rules:
- apiGroups:
  - ''
  resources:
  - pods
  verbs:
  - get
  - list
  - watch
  - delete
- apiGroups:
  - ''
  resources:
  - nodes
  verbs:
  - get
  - list
  - watch
  - patch
- apiGroups:
  - ''
  resources:
  - pods/eviction
  verbs:
  - create
```

Die `ClusterRoleBinding` kann sich im Laufe der Zeit ändern, sollte aber dem folgenden Beispiel ähneln:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: eks:node-manager
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
```

```
  name: eks:node-manager
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: eks:node-manager
```

Stellen Sie sicher, dass die Ressource `eks:node-manager` `ClusterRole` besteht.

```
kubectl describe clusterrole eks:node-manager
```

Wenn bestehend, vergleichen Sie die Ausgabe mit dem vorherigen `ClusterRole`-Beispiel.

Stellen Sie sicher, dass die Ressource `eks:node-manager` `ClusterRoleBinding` besteht.

```
kubectl describe clusterrolebinding eks:node-manager
```

Wenn bestehend, vergleichen Sie die Ausgabe mit dem vorherigen `ClusterRoleBinding`-Beispiel.

Wenn Sie einen fehlenden oder defekten `ClusterRole` oder `ClusterRoleBinding` als Ursache eines `AcessDenied`-Fehlers beim Anfordern verwalteter Knotengruppenoperationen finden, können Sie sie wiederherstellen. Speichern Sie den folgenden Inhalt in einer Datei mit dem Namen *`eks-node-manager-role.yaml`*.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: eks:node-manager
rules:
- apiGroups:
  - ''
  resources:
  - pods
  verbs:
  - get
  - list
  - watch
  - delete
- apiGroups:
  - ''
  resources:
  - nodes
```

```
  verbs:
  - get
  - list
  - watch
  - patch
- apiGroups:
  - ''
  resources:
  - pods/eviction
  verbs:
  - create
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: eks:node-manager
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: eks:node-manager
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: eks:node-manager
```

Wenden Sie die Datei an.

```
kubectl apply -f eks-node-manager-role.yaml
```

Wiederholen Sie den Vorgang der Knotengruppe, um festzustellen, ob das Problem dadurch behoben wurde.

Not authorized for images

Eine mögliche Ursache für die Fehlermeldung `Not authorized for images` ist die Verwendung eines privaten Windows-AMI von Amazon EKS zum Starten verwalteter Windows-Knotengruppen. AWS Macht Windows AMIs, die älter als 4 Monate sind, nach der Veröffentlichung neuer AMIs privat, sodass auf sie nicht mehr zugegriffen werden kann. Wenn Ihre verwaltete Knotengruppe ein privates Windows AMI verwendet, sollten Sie erwägen, Ihre Windows verwaltete Knotengruppe zu aktualisieren. Wir können zwar nicht garantieren, dass wir Zugriff auf AMIs gewähren können, die privat gemacht wurden, aber Sie können den Zugriff beantragen, indem Sie ein Ticket beim AWS

Support einreichen. Weitere Informationen finden Sie unter Patches, Sicherheitsupdates und AMI-IDs im Amazon EC2 EC2-Benutzerhandbuch.

Der Knoten befindet sich im Status **NotReady**

Wenn Ihr Knoten in einen `NotReady` Status wechselt, deutet dies wahrscheinlich darauf hin, dass der Knoten fehlerhaft ist und nicht für die Planung eines neuen Pods Zugriffs verfügbar ist. Dies kann verschiedene Ursachen haben, z. B. wenn dem Knoten nicht genügend Ressourcen für CPU, Arbeitsspeicher oder verfügbaren Festplattenspeicher zur Verfügung stehen.

Für Amazon Windows EKS-optimierte AMIs gibt es keine Reservierung für Rechenressourcen, die standardmäßig in der `kubelet` Konfiguration angegeben sind. Um Ressourcenprobleme zu vermeiden, können Sie Rechenressourcen für Systemprozesse reservieren, indem Sie ihnen Konfigurationswerte für `kube-reserved`und/oder zur Verfügung stellen `system-reserved`. `kubelet` Dazu verwenden Sie den `-KubeletExtraArgs` Befehlszeilenparameter im Bootstrap-Skript. Weitere Informationen finden Sie in der Kubernetes Dokumentation unter Reserve Compute Resources for System Daemons und in diesem Benutzerhandbuch unter Bootstrap-Skript-Konfigurationsparameter.

CNI-Protokollerfassungstool

Das Amazon VPC CNI plugin for -Kubernetes verfügt über ein eigenes Fehlerbehebungsskript, das auf Knoten unter `/opt/cni/bin/aws-cni-support.sh` verfügbar ist. Sie können das Skript verwenden, um Diagnoseprotokolle für Support-Fälle und allgemeine Fehlerbehebung zu sammeln.

Mit dem folgenden Befehl können Sie das Skript auf dem Knoten ausführen:

```
sudo bash /opt/cni/bin/aws-cni-support.sh
```

> **Note**
>
> Wenn das Skript an diesem Speicherort nicht vorhanden ist, konnte der CNI-Container nicht ausgeführt werden. Sie können das Skript mit dem folgenden Befehl manuell herunterladen und ausführen:
>
> ```
> curl -O https://raw.githubusercontent.com/awslabs/amazon-eks-ami/master/log-collector-script/linux/eks-log-collector.sh
> ```

```
sudo bash eks-log-collector.sh
```

Das Skript sammelt die folgenden Diagnoseinformationen: Die bereitgestellte CNI-Version kann jünger als die Skriptversion sein.

```
      This is version 0.6.1. New versions can be found at https://github.com/awslabs/
amazon-eks-ami

Trying to collect common operating system logs...
Trying to collect kernel logs...
Trying to collect mount points and volume information...
Trying to collect SELinux status...
Trying to collect iptables information...
Trying to collect installed packages...
Trying to collect active system services...
Trying to collect Docker daemon information...
Trying to collect kubelet information...
Trying to collect L-IPAMD information...
Trying to collect sysctls information...
Trying to collect networking information...
Trying to collect CNI configuration information...
Trying to collect running Docker containers and gather container data...
Trying to collect Docker daemon logs...
Trying to archive gathered information...

 Done... your bundled logs are located in /var/
log/eks_i-0717c9d54b6cfaa19_2020-03-24_0103-UTC_0.6.1.tar.gz
```

Die Diagnoseinformationen werden gesammelt und gespeichert unter:

```
/var/log/eks_i-0717c9d54b6cfaa19_2020-03-24_0103-UTC_0.6.1.tar.gz
```

Container-Laufzeitnetzwerk nicht bereit

Möglicherweise erhalten Sie einen `Container runtime network not ready`-Fehler und einen Autorisierungsfehler, die den folgenden ähneln:

```
4191 kubelet.go:2130] Container runtime network not ready: NetworkReady=false
 reason:NetworkPluginNotReady message:docker: network plugin is not ready: cni config
 uninitialized
```

```
4191 reflector.go:205] k8s.io/kubernetes/pkg/kubelet/kubelet.go:452: Failed to list
 *v1.Service: Unauthorized
4191 kubelet_node_status.go:106] Unable to register node
 "ip-10-40-175-122.ec2.internal" with API server: Unauthorized
4191 reflector.go:205] k8s.io/kubernetes/pkg/kubelet/kubelet.go:452: Failed to list
 *v1.Service: Unauthorized
```

Dieses Problem kann folgende Ursachen haben:

1. Für Ihren Cluster ist keine `aws-auth` `ConfigMap` vorhanden oder sie enthält keine Einträge für die IAM-Rolle, mit der Sie Ihre Knoten konfiguriert haben.

   Dieser `ConfigMap`-Eintrag ist erforderlich, wenn Ihre Knoten eines der folgenden Kriterien erfüllen:

   - Verwaltete Knoten in einem Cluster mit einer beliebigen Kubernetes- oder Plattformversion.
   - Selbstverwaltete Knoten in einem Cluster, der älter ist als eine der Plattformversionen, die im Abschnitt „Voraussetzungen“ des Themas Zugangseinträge verwalten aufgeführt sind.

   Sehen Sie sich zur Behebung des Problems die vorhandenen Einträge in `ConfigMap` an, indem Sie *`my-cluster`* im folgenden Befehl durch den Namen Ihres Clusters ersetzen und dann den geänderten Befehl ausführen: **`eksctl get iamidentitymapping --cluster`** ***`my-cluster`***. Sollten Sie im Zusammenhang mit dem Befehl eine Fehlermeldung erhalten, verfügt Ihr Cluster möglicherweise nicht über eine `aws-auth` `ConfigMap`. Der folgende Befehl fügt `ConfigMap` einen Eintrag hinzu. Ist `ConfigMap` nicht vorhanden, wird sie durch den Befehl erstellt. Ersetzen Sie *`111122223333`* durch die AWS-Konto ID für die IAM-Rolle und *`myAmazonEKS durch den Namen der Rolle Ihres Knotens NodeRole`*.

   ```
   eksctl create iamidentitymapping --cluster my-cluster \
       --arn arn:aws:iam::111122223333:role/myAmazonEKSNodeRole --group
    system:bootstrappers,system:nodes \
       --username system:node:{{EC2PrivateDNSName}}
   ```

   Der ARN der von Ihnen angegebenen Rolle darf als Pfad nur `/` enthalten. Wenn der Name Ihrer Rolle also beispielsweise `development/apps/my-role` lautet, müssen Sie ihn beim Angeben des ARN der Rolle in `my-role` ändern. Achten Sie darauf, den ARN der IAM-Rolle des Knotens (nicht den ARN des Instances-Profils) anzugeben.
2. Ihre selbstverwalteten Knoten befinden sich in einem Cluster mit einer Plattformversion, die die Mindestversionsanforderung erfüllt, die unter „Voraussetzungen“ des Themas Zugangseinträge

verwalten angegeben ist. Aber `aws-auth ConfigMap` enthält keinen Eintrag für die IAM-Rolle des Knotens (siehe vorheriger Punkt) oder es ist kein Zugriffseintrag für die Rolle vorhanden. Sehen Sie sich zur Behebung des Problems Ihre vorhandenen Zugriffseinträge an, indem Sie *my-cluster* im folgenden Befehl durch den Namen Ihres Clusters ersetzen und dann den geänderten Befehl ausführen: **aws eks list-access-entries --cluster-name *my-cluster***. Der folgende Befehl fügt einen Zugriffseintrag für die IAM-Rolle des Knotens hinzu. *Ersetzen Sie 111122223333 durch die AWS-Konto ID für die IAM-Rolle und myAmazonEKS durch den Namen der Rolle Ihres Knotens. NodeRole* Ersetzen Sie im Falle eines Windows-Knotens *EC2_Linux* durch **EC2_Windows**. Achten Sie darauf, den ARN der IAM-Rolle des Knotens (nicht den ARN des Instances-Profils) anzugeben.

```
aws eks create-access-entry --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/myAmazonEKSNodeRole --type EC2_Linux
```

TLS-Handshake-Zeitüberschreitung

Wenn ein Knoten nicht in der Lage ist, eine Verbindung zum öffentlichen API-Server-Endpunkt herzustellen, erhalten Sie möglicherweise einen Fehler ähnlich dem folgenden.

```
server.go:233] failed to run Kubelet: could not init cloud provider "aws": error
 finding instance i-1111f2222f333e44c: "error listing AWS instances: \"RequestError:
 send request failed\\ncaused by: Post  net/http: TLS handshake timeout\""
```

Der `kubelet`-Prozess wird weiter fortgesetzt und testet den API-Server-Endpunkt. Der Fehler kann auch vorübergehend während jeder Prozedur auftreten, die eine laufende Aktualisierung des Clusters auf der Steuerungsebene durchführt, z. B. eine Konfigurationsänderung oder eine Versionsaktualisierung.

Um das Problem zu beheben, überprüfen Sie die Routing-Tabelle und die Sicherheitsgruppen, um sicherzustellen, dass der Datenverkehr von den Knoten den öffentlichen Endpunkt erreichen kann.

InvalidClientTokenId

Wenn Sie IAM-Rollen für Dienstkonten für einen Cluster in China verwenden Pod oder in einem Cluster in China `DaemonSet` bereitgestellt werden und die `AWS_DEFAULT_REGION` Umgebungsvariable nicht in der Spezifikation festgelegt haben AWS-Region, wird bei oder möglicherweise der folgende Fehler angezeigt: Pod `DaemonSet`

```
An error occurred (InvalidClientTokenId) when calling the GetCallerIdentity operation:
 The security token included in the request is invalid
```

Zur Behebung des Problems müssen Sie die `AWS_DEFAULT_REGION`-Umgebungsvariable Ihrer Pod- oder `DaemonSet`-Spezifikation wie in der folgenden Beispiel-Pod-Spezifikation hinzufügen.

```
apiVersion: v1
kind: Pod
metadata:
  name: envar-demo
  labels:
    purpose: demonstrate-envars
spec:
  containers:
  - name: envar-demo-container
    image: gcr.io/google-samples/node-hello:1.0
    env:
    - name: AWS_DEFAULT_REGION
      value: "region-code"
```

Ablauf des Webhook-Zertifikats für die VPC-Zulassung

Wenn das Zertifikat, das zum Signieren des Webhook für die VPC-Zulassung verwendet wurde, abläuft, bleibt der Status für neue Windows-Pod-Bereitstellungen auf `ContainerCreating`.

Informationen zum Beheben des Problems, wenn Sie Legacy-System-Windows-Support auf Ihrer Datenebene haben, finden Sie unter Erneuerung des VPC-Zulassungs-Webhook-Zertifikats. Wenn Ihre Cluster- und Plattformversion höher ist als eine Version, die unter Windows-Supportvoraussetzungen aufgeführt ist, wird empfohlen, den Legacy-System-Windows-Support auf Ihrer Datenebene zu entfernen und für Ihre Steuerebene zu aktivieren. Sobald Sie dies getan haben, müssen Sie das Webhook-Zertifikat nicht mehr verwalten. Weitere Informationen finden Sie unter Die Windows-Unterstützung für Ihre Amazon-EKS-Cluster aktivieren.

Knotengruppen müssen der Kubernetes-Version entsprechen, bevor ein Upgrade der Steuerebene durchgeführt wird

Bevor Sie für eine Steuerebene ein Upgrade auf eine neue Kubernetes-Version durchführen, muss die Nebenversion der verwalteten Knoten und der Fargate-Knoten in Ihrem Cluster der aktuellen

Version Ihrer Steuerebene entsprechen. Die Amazon EKS-`update-cluster-version`-API akzeptiert keine Anforderungen, bis für alle von Amazon EKS verwalteten Knoten ein Upgrade auf die aktuelle Cluster-Version durchgeführt wurde. Amazon EKS stellt APIs zum Upgraden verwalteter Knoten bereit. Informationen zum Upgraden der Kubernetes-Version einer verwalteten Knotengruppe finden Sie unter Aktualisieren einer verwalteten Knotengruppe. Löschen Sie zum Upgraden der Version eines Fargate-Knotens den pod, der durch den Knoten dargestellt wird, und stellen Sie den pod erneut bereit, nachdem Sie das Upgrade für Ihre Steuerebene durchgeführt haben. Weitere Informationen finden Sie unter Aktualisieren einer Amazon-EKS-Cluster-Kubernetes-Version.

Beim Starten vieler Knoten gibt es **`Too Many Requests`**-Fehler

Wenn Sie viele Knoten gleichzeitig starten, wird möglicherweise die Fehlermeldung `Too Many Requests` in den Ausführungsprotokollen für die Amazon-EC2-Benutzerdaten angezeigt. Dies kann auftreten, weil die Steuerebene mit `describeCluster`-Aufrufen überlastet wird. Die Überladung führt zu einer Drosselung, Knoten, die das Bootstrap-Skript nicht ausführen können, und Knoten, die dem Cluster insgesamt nicht beitreten können.

Stellen Sie sicher, dass die Argumente `--apiserver-endpoint`, `--b64-cluster-ca` und `--dns-cluster-ip` an das Bootstrap-Skript des Knotens übergeben werden. Wenn Sie diese Argumente einschließen, muss das Bootstrap-Skript keinen `describeCluster`-Aufruf ausführen, wodurch verhindert wird, dass die Steuerebene überlastet wird. Weitere Informationen finden Sie unter Stellen Sie Benutzerdaten bereit, um Argumente an die `bootstrap.sh`-Datei zu übergeben, die in einem für Amazon EKS optimierten Linux/Bottlerocket-AMI enthalten ist.

Unautorisierte Fehlerantwort (HTTP 401) bei Kubernetes-API-Serveranfragen

Sie sehen diese Fehler, wenn das Servicekonto-Token eines Pods auf einem Cluster abgelaufen ist.

Der Kubernetes-API-Server Ihres Amazon-EKS-Clusters lehnt Anfragen mit Tokens ab, die älter als 90 Tage sind. In früheren Kubernetes-Versionen hatten Token kein Ablaufdatum. Clients, die diese Tokens verwenden, müssen die Tokens nun innerhalb einer Stunde aktualisieren. Um zu verhindern, dass der Kubernetes-API-Server Ihre Anfrage aufgrund eines ungültigen Tokens ablehnt, muss die von Ihrem Workload verwendete Kubernetes-Client-SDK-Version dieselbe oder höher als die folgenden Versionen sein:

- Go-Version `0.15.7` und höher

- Python-Version `12.0.0` und höher
- Java-Version `9.0.0` und höher
- JavaScript Version `0.10.3` und später
- Ruby `master`-Branch
- Haskell-Version `0.3.0.0`
- C#-Version `7.0.5` und höher

Sie können alle vorhandenen Pods in Ihrem Cluster ermitteln, die veraltete Token verwenden. Weitere Informationen finden Sie unter Kubernetes-Servicekonten.

Die Amazon-EKS-Plattformversion liegt mehr als zwei Versionen hinter der aktuellen Plattformversion

Dies kann passieren, wenn Amazon EKS die Plattformversion Ihres Clusters nicht automatisch aktualisieren kann. Obwohl es dafür viele Ursachen gibt, folgen einige der häufigsten Ursachen. Wenn Ihr Cluster von einem dieser Probleme betroffen ist, funktioniert er möglicherweise immer noch, aber seine Plattformversion wird nicht von Amazon EKS aktualisiert.

Problem

Die Cluster-IAM-Rolle wurde gelöscht – Diese Rolle wurde beim Erstellen des Clusters angegeben. Sie können mit dem folgenden Befehl überprüfen, welche Rolle angegeben wurde. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

```
aws eks describe-cluster --name my-cluster --query cluster.roleArn --output text | cut
 -d / -f 2
```

Eine Beispielausgabe sieht wie folgt aus.

```
eksClusterRole
```

Lösung

Erstellen einer neuen Cluster-IAM-Rolle mit demselben Namen.

Problem

Ein bei der Clustererstellung festgelegtes Subnetz wurde gelöscht – Die mit dem Cluster zu verwendenden Subnetze wurden bei der Clustererstellung angegeben. Sie können mit dem folgenden Befehl überprüfen, welche Subnetze angegeben wurden. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

```
aws eks describe-cluster --name my-cluster --query cluster.resourcesVpcConfig.subnetIds
```

Eine Beispielausgabe sieht wie folgt aus.

```
[
"subnet-EXAMPLE1",
"subnet-EXAMPLE2"
]
```

Lösung

Bestätigen Sie, ob die Subnetz-IDs in Ihrem Konto vorhanden sind.

```
vpc_id=$(aws eks describe-cluster --name my-cluster --query
 cluster.resourcesVpcConfig.vpcId --output text)
aws ec2 describe-subnets --filters "Name=vpc-id,Values=$vpc_id" --query
 "Subnets[*].SubnetId"
```

Eine Beispielausgabe sieht wie folgt aus.

```
[
"subnet-EXAMPLE3",
"subnet-EXAMPLE4"
]
```

Wenn die in der Ausgabe zurückgegebenen Subnetz-IDs nicht mit den Subnetz-IDs übereinstimmen, die bei der Erstellung des Clusters angegeben wurden, müssen Sie die vom Cluster verwendeten Subnetze ändern, wenn Sie möchten, dass Amazon EKS den Cluster aktualisiert. Dies liegt daran, dass Amazon EKS, wenn Sie bei der Erstellung Ihres Clusters mehr als zwei Subnetze angegeben haben, nach dem Zufallsprinzip Subnetze auswählt, die Sie für die Erstellung neuer Elastic-Network-Schnittstellen angegeben haben. Diese Netzwerkschnittstellen ermöglichen es der Steuerebene, mit Ihren Knoten zu kommunizieren. Amazon EKS aktualisiert den Cluster nicht, wenn das von ihm ausgewählte Subnetz nicht existiert. Sie haben keine Kontrolle darüber, in welchem der

Subnetze, die Sie bei der Clustererstellung angegeben haben, Amazon EKS wählt, um eine neue Netzwerkschnittstelle zu erstellen.

Wenn Sie ein Kubernetes-Versionsupdate für Ihren Cluster initiieren, kann das Update aus demselben Grund fehlschlagen.

Problem

Eine bei der Clustererstellung angegebene Sicherheitsgruppe wurde gelöscht – Wenn Sie bei der Clustererstellung Sicherheitsgruppen angegeben haben, können Sie deren IDs mit dem folgenden Befehl anzeigen. Ersetzen Sie *my-cluster* durch den Namen Ihres Clusters.

```
aws eks describe-cluster --name my-cluster --query
 cluster.resourcesVpcConfig.securityGroupIds
```

Eine Beispielausgabe sieht wie folgt aus.

```
[
    "sg-EXAMPLE1"
]
```

Wenn `[]` zurückgegeben wird, wurden beim Erstellen des Clusters keine Sicherheitsgruppen angegeben und eine fehlende Sicherheitsgruppe ist nicht das Problem. Wenn Sicherheitsgruppen zurückgegeben werden, vergewissern Sie sich, dass die Sicherheitsgruppen in Ihrem Konto vorhanden sind.

Lösung

Bestätigen Sie, ob diese Sicherheitsgruppen in Ihrem Konto vorhanden sind.

```
vpc_id=$(aws eks describe-cluster --name my-cluster --query
 cluster.resourcesVpcConfig.vpcId --output text)
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=$vpc_id" --query
 "SecurityGroups[*].GroupId"
```

Eine Beispielausgabe sieht wie folgt aus.

```
[
"sg-EXAMPLE2"
```

```
]
```

Wenn die in der Ausgabe zurückgegebenen Sicherheitsgruppen-IDs nicht mit den Sicherheitsgruppen-IDs übereinstimmen, die bei der Erstellung des Clusters angegeben wurden, müssen Sie die vom Cluster verwendeten Sicherheitsgruppen ändern, wenn Sie möchten, dass Amazon EKS den Cluster aktualisiert. Amazon EKS aktualisiert einen Cluster nicht, wenn die bei der Clustererstellung angegebenen Sicherheitsgruppen-IDs nicht existieren.

Wenn Sie ein Kubernetes-Versionsupdate für Ihren Cluster initiieren, kann das Update aus demselben Grund fehlschlagen.

Andere Gründe, warum Amazon EKS die Plattformversion Ihres Clusters nicht aktualisiert

- Sie haben nicht mindestens sechs verfügbare IP-Adressen (wir empfehlen jedoch 16) in jedem der Subnetze, die Sie bei der Erstellung Ihres Clusters angegeben haben. Wenn Sie nicht über genügend verfügbare IP-Adressen im Subnetz verfügen, müssen Sie entweder IP-Adressen im Subnetz freigeben oder die vom Cluster verwendeten Subnetze ändern, um Subnetze mit genügend verfügbaren IP-Adressen zu verwenden.
- Sie haben die Verschlüsselung von Geheimnissen aktiviert, als Sie Ihren Cluster erstellt haben, und der von Ihnen angegebene AWS KMS Schlüssel wurde gelöscht. Wenn Sie möchten, dass Amazon EKS den Cluster aktualisiert, müssen Sie einen neuen Cluster erstellen

Häufig gestellte Fragen zur Clusterintegrität und Fehlercodes mit Lösungspfaden

Amazon EKS erkennt Probleme mit Ihren EKS-Clustern und der Cluster-Infrastruktur und speichert sie in der Cluster-Integrität. Mithilfe von Cluster-Integritätsinformationen können Sie Cluster-Probleme schneller erkennen, behandeln und beheben. Auf diese Weise können Sie Anwendungsumgebungen erstellen, die sicherer sind und up-to-date. Darüber hinaus kann es aufgrund von Problemen mit der erforderlichen Infrastruktur- oder Clusterkonfiguration sein, dass Sie kein Upgrade auf neuere Versionen von Kubernetes durchführen können oder dass Amazon EKS keine Sicherheitsupdates auf einem beeinträchtigten Cluster installieren kann. Es kann drei Stunden dauern, bis Amazon EKS Probleme erkennt oder feststellt, dass ein Problem behoben wurde.

Amazon EKS und die Benutzer sind gemeinsam für die Integrität eines Amazon EKS-Clusters verantwortlich. Sie selbst sind für die erforderliche Infrastruktur von IAM-Rollen und Amazon VPC-Subnetzen sowie für andere notwendige Infrastrukturkomponenten verantwortlich, die im Voraus

bereitgestellt werden müssen. Amazon EKS erkennt Änderungen an der Konfiguration dieser Infrastruktur und des Clusters.

Sie können über die Amazon EKS-Konsole auf die Integrität Ihres Clusters zugreifen. Suchen Sie hierzu auf der Registerkarte Übersicht der Detailseite des Amazon EKS-Clusters nach dem Abschnitt Integritätsprobleme. Diese Daten sind auch verfügbar, wenn Sie die Aktion `DescribeCluster` der EKS-API aufrufen (beispielsweise innerhalb der AWS Command Line Interface).

Warum sollte ich diese Funktion verwenden?

Sie erhalten einen besseren Einblick in den Zustand Ihres Amazon EKS-Clusters und können Probleme schnell diagnostizieren und beheben, ohne Zeit mit dem Debuggen oder dem Öffnen von AWS Supportanfragen verbringen zu müssen. Beispiel: Sie haben versehentlich ein Subnetz für den Amazon EKS-Cluster gelöscht. Amazon EKS kann keine kontoübergreifenden Netzwerkschnittstellen und Kubernetes AWS CLI Befehle wie `kubectl` Exec oder `kubectl` Logs erstellen. Bei diesen tritt folgender Fehler auf: `Error from server: error dialing backend: remote error: tls: internal error.` Nun wird ein Amazon EKS-Integritätsproblem mit folgendem Hinweis angezeigt: `subnet-da60e280 was deleted: could not create network interface`.

In welcher Beziehung steht diese Funktion zu anderen AWS Diensten oder funktioniert mit ihnen?

IAM-Rollen und Amazon VPC-Subnetze sind zwei Beispiele für erforderliche Infrastrukturkomponenten, für die die Cluster-Integritätsfunktion Probleme erkennt. Diese Funktion gibt detaillierte Informationen zurück, wenn diese Ressourcen nicht ordnungsgemäß konfiguriert sind.

Fallen für Cluster mit Integritätsproblemen Gebühren an?

Ja. Jeder Amazon EKS-Cluster wird zu den Amazon EKS-Standardpreisen abgerechnet. Die Funktion Cluster-Integrität steht ohne Aufpreis zur Verfügung.

Kann diese Funktion mit Amazon EKS-Clustern in AWS Outposts verwendet werden?

Ja, Clusterprobleme wurden für EKS-Cluster in der AWS Cloud erkannt, einschließlich erweiterter Cluster AWS Outposts und eingeschalteter lokaler Cluster AWS Outposts. Die Cluster-Integritätsfunktion erkennt keine Probleme mit Amazon EKS Anywhere oder Amazon EKS Distro (EKS-D).

Kann ich benachrichtigt werden, wenn neue Probleme erkannt werden?

Nein. Sie müssen die Amazon EKS-Konsole überprüfen oder die EKS-API `DescribeCluster` aufrufen.

Werden in der Konsole Warnungen im Zusammenhang mit Integritätsproblemen ausgegeben?

Ja. Cluster mit Problemen verfügen über ein Banner im oberen Bereich der Konsole.

Die ersten beiden Spalten werden für API-Antwortwerte benötigt. Das dritte Feld des ClusterIssueHealth-Objekts istresourceIds, dessen Rückgabe vom Problemtyp abhängt.

Code	Fehlermeldung	ResourceIds	Cluster wiederherstellbar?	
SUBNET_NOT_FOUND	We couldn't find one or more subnets currently associated with your cluster. Call Amazon EKS update-cluster-config API to update subnets.	Subnetz-IDs	Ja	
SECURITY_GROUP_NOT_FOUND	We couldn't find one or more security groups currently associated with your cluster. Rufen Sie die Amazon update-cluster-config EKS-API auf, um Sicherheitsgruppen zu aktualisieren	Sicherheitsgruppen-IDs	Ja	
IP_NOT_AVAILABLE	One or more of the subnets associated with your cluster does not have enough available IP addresses for Amazon EKS to perform cluster management operations. Geben Sie Adressen in den Subnetzen frei oder ordnen Sie Ihrem Cluster mithilfe der Amazon update-cluster-config EKS-API verschiedene Subnetze zu.	Subnetz-IDs	Ja	

Code	Fehlermeldung	ResourceIds	Cluster wiederherstellbar?	
VPC_NOT_FOUND	We couldn't find the VPC associated with your cluster. You must delete and recreate your cluster.	VPC-ID	Nein	
ASSUME_ROLE_ACCESS_DENIED	Ihr Cluster verwendet Amazon EKS nicht service-linked-role. We couldn't assume the role associated with your cluster to perform required Amazon EKS management operations. Check the role exists and has the required trust policy.	Die IAM-Rolle des Clusters	Ja	
PERMISSION_ACCESS_DENIED	Ihr Cluster verwendet Amazon EKS nicht service-linked-role. The role associated with your cluster does not grant sufficient permissions for Amazon EKS to perform required management operations. Check the policies attached to the cluster role and if any separate deny policies are applied.	Die IAM-Rolle des Clusters	Ja	
ASSUME_ROLE_ACCESS_DENIED_USING_SLR	Wir konnten nicht von der Amazon EKS-Clusterverwaltung ausgehen service-linked-role. Check the role exists and has the required trust policy.	Das Amazon EKS service-linked-role	Ja	

Code	Fehlermeldung	ResourceIds	Cluster wiederherstellbar?	
PERMISSION_ACCESS_DENIED_USING_SLR	Die Amazon EKS-Clusterverwaltung gewährt Amazon EKS service-linked-role nicht genügend Berechtigungen, um die erforderlichen Verwaltungsvorgänge durchzuführen. Check the policies attached to the cluster role and if any separate deny policies are applied.	Das Amazon EKS service-linked-role	Ja	
OPT_IN_REQUIRED	Your account doesn't have an Amazon EC2 service subscription. Update your account subscriptions in your account settings page.	N/A	Ja	
STS_REGIONAL_ENDPOINT_DISABLED	The STS regional endpoint is disabled. Enable the endpoint for Amazon EKS to perform required cluster management operations.	N/A	Ja	
KMS_KEY_DISABLED	Der mit Ihrem Cluster verknüpfte AWS KMS Schlüssel ist deaktiviert. Re-enable the key to recover your cluster.	Die KMS Key Arn	Ja	
KMS_KEY_NOT_FOUND	Wir konnten den mit Ihrem Cluster verknüpften AWS KMS Schlüssel nicht finden. You must delete and recreate the cluster.	Die KMS Key ARN	Nein	

Code	Fehlermeldung	ResourceIds	Cluster wiederherstellbar?	
KMS_GRANT_REVOKED	Zuschüsse für den mit Ihrem Cluster verknüpften AWS KMS Schlüssel wurden widerrufen. You must delete and recreate the cluster.	Die KMS Key Arn	Nein	

Amazon-EKS-Anschluss

Sie können den Amazon EKS Connector verwenden, um alle konformen Kubernetes-Cluster zu registrieren und mit AWS zu verbinden und in der Amazon-EKS-Konsole visualisieren. Nachdem ein Cluster verbunden wurde, können Sie den Status, die Konfiguration und die Workloads für diesen Cluster in der Amazon-EKS-Konsole anzeigen. Sie können dieses Feature verwenden, um verbundene Cluster in der Amazon-EKS-Konsole anzuzeigen, aber Sie können sie nicht verwalten. Der Amazon EKS Connector erfordert einen Agent, der ein Open-Source-Projekt auf Github ist. Weitere technische Inhalte, darunter häufig gestellte Fragen und Fehlerbehebung, finden Sie unter Beheben von Problemen in Amazon EKS Connector

Der Amazon EKS Connector kann die folgenden Typen von Kubernetes-Clustern mit Amazon EKS verbinden.

- On-Premises Kubernetes-Cluster
- Selbstverwaltete Cluster, die auf Amazon EC2 ausgeführt werden
- Verwaltete Cluster von anderen Cloud-Anbietern

Überlegungen zum Amazon EKS Connector

Bevor Sie Amazon EKS Connector verwenden, sollten Sie Folgendes verstehen:

- Sie müssen über Administratorrechte für den Kubernetes-Cluster verfügen, um den Cluster mit Amazon EKS zu verbinden.
- Im Kubernetes-Cluster müssen Linux-64-Bit-Worker-Knoten (x86) vorhanden sein, bevor eine Verbindung hergestellt werden kann. ARM-Worker-Knoten werden nicht unterstützt.
- In Ihrem `ssm.`-Cluster müssen Worker-Knoten vorhanden sein, die ausgehenden Zugriff auf die Kubernetes- und `ssmmessages.`-Systems Manager-Endpunkte haben. Weitere Informationen finden Sie unter Systems-Manager-Endpunkte in der Allgemeine AWS-Referenz.
- Standardmäßig können Sie bis zu zehn Cluster in einer Region verbinden. Sie können eine Erhöhung über die Servicekontingentkonsole anfordern. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung.
- Für externe Kubernetes-Cluster werden nur die Amazon EKS `RegisterCluster`-, `ListClusters`-, `DescribeCluster`- und `DeregisterCluster`-APIs unterstützt.
- Sie müssen über die folgenden Berechtigungen verfügen, um einen Cluster zu registrieren:

- eks:RegisterCluster
- ssm:CreateActivation
- ssm:DeleteActivation
- iam: PassRole

- Sie müssen über die folgenden Berechtigungen verfügen, um die Registrierung eines Clusters aufzuheben:
  - eks:DeregisterCluster
  - ssm:DeleteActivation
  - ssm:DeregisterManagedInstance

Erforderliche IAM-Rollen für Amazon EKS Connector

Die Verwendung des Amazon EKS Connectors erfordert die folgenden zwei IAM-Rollen:

- Die serviceverknüpfte Rolle Amazon EKS Connector wird bei der ersten Registrierung eines Clusters erstellt.
- Sie müssen die IAM-Rolle des Amazon-EKS-Connector-Agents erstellen. Details dazu finden Sie unter Amazon-EKS-Konnektor-IAM-Rolle.

Um die Cluster- und Workload-Ansichtsberechtigung für IAM-Prinzipale zu aktivieren, müssen Sie die Clusterrollen `eks-connector` und Amazon EKS Connector auf Ihren Cluster anwenden. Führen Sie die Schritte unter Gewähren des Zugriffs für einen IAM-Prinzipal zum Anzeigen von Kubernetes-Ressource eines Clusters aus.

Verbinden eines externen Clusters

Sie können einen externen Kubernetes-Cluster mit Amazon EKS verbinden, indem Sie mehrere Methoden im folgenden Prozess verwenden. Dieser Prozess umfasst zwei Schritte: die Registrierung des Clusters bei Amazon EKS und die Installation des `eks-connector`-Agents im Cluster.

> **Important**
>
> Der zweite Schritt muss innerhalb von 3 Tagen nach Abschluss des ersten Schritts erfolgen, bevor die Registrierung abläuft.

Connector-Methoden

Nicht alle Methoden zur Installation des Agents können nach jeder der Methoden zur Registrierung des Clusters verwendet werden. In der folgenden Tabelle sind die einzelnen Registrierungsmethoden aufgeführt und welche Methoden zur Installation des Agents verwendet werden können.

Schritt	Methoden		
Registrieren des Clusters	AWS Management Console	AWS Command Line Interface	`eksctl`
Installieren des Agents	Helm, YAML-Manifeste	Helm, YAML-Manifeste	YAML-Manifeste

Voraussetzungen

- Stellen Sie sicher, dass die Agent-Rolle von Amazon EKS Connector erstellt wurde. Führen Sie die Schritte unter Erstellen der Rolle des Amazon-EKS-Connector-Agenten aus.
- Sie müssen über die folgenden Berechtigungen verfügen, um einen Cluster zu registrieren:
  - `eks:RegisterCluster`
  - `ssm:CreateActivation`
  - `ssm:DeleteActivation`
  - `iam:PassRole`

Schritt 1: Registrieren des Clusters

AWS CLI

Voraussetzungen

- AWS CLI muss installiert sein. Informationen zur Installation oder Aktualisierung finden Sie unter Installieren des AWS CLI.

So registrieren Sie Ihren Cluster beim AWS CLI

- Geben Sie für die Connector-Konfiguration Ihre Amazon-EKS-Connector-Agent-IAM-Rolle an. Weitere Informationen finden Sie unter Erforderliche IAM-Rollen für Amazon EKS Connector.

```
aws eks register-cluster \
     --name my-first-registered-cluster \
     --connector-config roleArn=arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole,provider="OTHER" \
     --region aws-region
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "cluster": {
        "name": "my-first-registered-cluster",
        "arn": "arn:aws:eks:region:111122223333:cluster/my-first-registered-
cluster",
        "createdAt": 1627669203.531,
        "ConnectorConfig": {
            "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
            "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
            "activationExpiry": 1627672543.0,
            "provider": "OTHER",
            "roleArn": "arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole"
        },
        "status": "CREATING"
    }
}
```

Sie verwenden die Werte `aws-region`, `activationId` und `activationCode` im nächsten Schritt.

AWS Management Console

Um Ihren Kubernetes-Cluster bei der Konsole zu registrieren.

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.

2. Wählen Sie Add cluster (Cluster hinzufügen) und Register (Registrieren) aus, um die Konfigurationsseite aufzurufen.
3. Füllen Sie im Abschnitt Cluster konfigurieren die folgenden Felder aus:
   - Name – Ein eindeutiger Name für Ihren Cluster.
   - Provider (Provider) – Hiermit zeigen Sie die Dropdown-Liste der Kubernetes-Clusteranbieter an. Wenn Sie den spezifischen Anbieter nicht kennen, wählen Sie Andere.
   - Rolle EKS-Connector – Wählen Sie die Rolle aus, die für die Verbindung mit dem Cluster verwendet werden soll.
4. Wählen Sie Cluster registrieren aus.
5. Die Seite Cluster-Übersicht wird angezeigt. Wenn Sie das Helm-Chart verwenden möchten, kopieren Sie den Befehl `helm install` und fahren Sie mit dem nächsten Schritt fort. Wenn Sie das YAML-Manifest verwenden möchten, wählen Sie YAML-Datei herunterladen aus, um die Manifestdatei auf Ihr lokales Laufwerk herunterzuladen.

   > **Important**
   >
   > - Dies ist Ihre einzige Möglichkeit, den Befehl `helm install` zu kopieren oder diese Datei herunterzuladen. Navigieren Sie nicht von dieser Seite weg, da auf den Link nicht zugegriffen werden kann und Sie den Cluster abmelden und die Schritte von vorne beginnen müssen.
   > - Der Befehl bzw. die Manifestdatei kann nur einmal für den registrierten Cluster verwendet werden. Wenn Sie Ressourcen aus dem Kubernetes-Cluster löschen, müssen Sie den Cluster erneut registrieren und eine neue Manifestdatei abrufen.

   Fahren Sie mit dem nächsten Schritt fort, um die Manifestdatei auf Ihren Kubernetes-Cluster anzuwenden.

eksctl

Voraussetzungen

- `eksctl`-Version `0.68` oder höher muss installiert sein. Informationen zur Installation oder Aktualisierung finden Sie unter Erste Schritte mit Amazon EKS – `eksctl`.

So registrieren Sie Ihren Cluster beim **`eksctl`**

1. Registrieren Sie den Cluster unter Angabe eines Namens, eines Anbieters und einer Region.

```
eksctl register cluster --name my-cluster --provider my-provider --
region region-code
```

Beispielausgabe:

```
2021-08-19 13:47:26 [#]  creating IAM role "eksctl-20210819194112186040"
2021-08-19 13:47:26 [#]  registered cluster "<name>" successfully
2021-08-19 13:47:26 [#]  wrote file eks-connector.yaml to <current directory>
2021-08-19 13:47:26 [#]  wrote file eks-connector-clusterrole.yaml to <current
 directory>
2021-08-19 13:47:26 [#]  wrote file eks-connector-console-dashboard-full-access-
group.yaml to <current directory>
2021-08-19 13:47:26 [!]  note: "eks-connector-clusterrole.yaml" and "eks-
connector-console-dashboard-full-access-group.yaml" give full EKS Console access
 to IAM identity "<aws-arn>", edit if required; read https://eksctl.io/usage/
eks-connector for more info
2021-08-19 13:47:26 [#]  run `kubectl apply -f eks-connector.yaml,eks-connector-
clusterrole.yaml,eks-connector-console-dashboard-full-access-group.yaml` before
 expiry> to connect the cluster
```

Dadurch werden Dateien auf Ihrem lokalen Computer erstellt. Diese Dateien müssen innerhalb von 3 Tagen auf den externen Cluster angewendet werden, sonst läuft die Registrierung ab.

2. Wenden Sie in einem Terminal, das Zugriff auf den Cluster hat, die Datei `eks-connector-binding.yaml` an:

```
kubectl apply -f eks-connector-binding.yaml
```

Schritt 2: Installieren des **`eks-connector`**-Agents

Helm chart

1. Wenn Sie im vorherigen Schritt die AWS CLI verwendet haben, ersetzen Sie `ACTIVATION_CODE` und `ACTIVATION_ID` im folgenden Befehl durch die Werte `activationId` bzw. `activationCode`. Ersetzen Sie die `aws-region` durch die AWS-Region, die Sie im vorherigen Schritt verwendet haben. Führen Sie dann den Befehl aus, um den `eks-connector`-Agent auf dem registrierenden Cluster zu installieren:

   ```
   $ helm install eks-connector \
     --namespace eks-connector \
     oci://public.ecr.aws/eks-connector/eks-connector-chart \
     --set eks.activationCode=ACTIVATION_CODE \
     --set eks.activationId=ACTIVATION_ID \
     --set eks.agentRegion=aws-region
   ```

   Wenn Sie im vorherigen Schritt die AWS Management Console verwendet haben, verwenden Sie den Befehl, den Sie aus dem vorherigen Schritt kopiert haben und in den diese Werte eingetragen sind.
2. Überprüfen Sie den Zustand der installierten `eks-connector`-Bereitstellung und warten Sie, bis der Status des registrierten Clusters in Amazon EKS `ACTIVE` lautet.

YAML manifest

Schließen Sie die Verbindung ab, indem Sie die Amazon-EKS-Connector-Manifestdatei auf Ihren Kubernetes-Cluster anwenden. Dazu müssen Sie die zuvor beschriebenen Methoden verwenden. Wenn das Manifest nicht innerhalb von drei Tagen angewendet wird, läuft die Amazon-EKS-Connector-Registrierung ab. Wenn die Clusterverbindung abläuft, muss die Registrierung des Clusters aufgehoben werden, bevor der Cluster erneut verbunden wird.

1. Laden Sie die Amazon EKS Connector YAML-Datei herunter.

   ```
   curl -O https://amazon-eks.s3.us-west-2.amazonaws.com/eks-connector/manifests/
   eks-connector/latest/eks-connector.yaml
   ```
2. Bearbeiten Sie die YAML-Datei von Amazon EKS Connector und ersetzen Sie alle Verweise auf `%AWS_REGION%`, `%EKS_ACTIVATION_ID%`, `%EKS_ACTIVATION_CODE%` durch die `aws-region`, `activationId` und `activationCode` aus der Ausgabe des vorherigen Schritts.

Der folgende Beispielbefehl kann diese Werte ersetzen.

```
sed -i "s~%AWS_REGION%~$aws-region~g; s~%EKS_ACTIVATION_ID
%~$EKS_ACTIVATION_ID~g; s~%EKS_ACTIVATION_CODE%~$(echo -n $EKS_ACTIVATION_CODE |
 base64)~g" eks-connector.yaml
```

> **Important**
>
> Stellen Sie sicher, dass Ihr Aktivierungscode das base64-Format hat.

3. In einem Terminal, das auf den Cluster zugreifen kann, können Sie jetzt die aktualisierte Manifestdatei mit dem folgenden Befehl anwenden:

```
kubectl apply -f eks-connector.yaml
```

4. Nachdem die YAML-Dateien des Amazon-EKS-Connector-Manifests und der Rollenbindung auf Ihren Kubernetes-Cluster angewendet wurden, bestätigen Sie, dass der Cluster jetzt verbunden ist.

```
aws eks describe-cluster \
     --name "my-first-registered-cluster" \
     --region AWS_REGION
```

   Die Ausgabe sollte `status=ACTIVE` enthalten.

5. (Optional) Fügen Sie Ihrem Cluster Tags hinzu. Weitere Informationen finden Sie unter Kennzeichnen Ihrer Amazon EKS-Ressourcen .

Nächste Schritte

Falls Sie Probleme mit diesen Schritten haben, finden Sie weitere Informationen unter Beheben von Problemen in Amazon EKS Connector.

Um zusätzlichen IAM-Benutzern Zugriff auf die Amazon-EKS-Konsole zu gewähren, um Kubernetes-Ressourcen in einem verbundenen Cluster anzusehen, siehe Gewähren des Zugriffs für einen IAM-Prinzipal zum Anzeigen von Kubernetes-Ressource eines Clusters.

Gewähren des Zugriffs für einen IAM-Prinzipal zum Anzeigen von Kubernetes-Ressource eines Clusters

Gewähren Sie zusätzlichen IAM-Prinzipals Zugriff auf die Amazon EKS-Konsole, um Informationen zu Kubernetes-Ressourcen anzuzeigen, die auf Ihrem verbundenen Cluster ausgeführt werden.

Voraussetzungen

Der IAM-Prinzipal die Sie für den Zugriff auf das AWS Management Console verwenden, muss die folgenden Anforderungen erfüllen:

- Er muss über die `eks:AccessKubernetesApi`-IAM-Berechtigung verfügen.
- Das Amazon-EKS-Connector-Service-Konto kann den IAM-Prinzipal im Cluster imitieren. Dadurch kann der Amazon EKS Connector den IAM-Prinzipal einem Kubernetes-Benutzer zuordnen.

So erstellen Sie die Amazon-EKS-Connector-Clusterrolle und wenden sie an

1. Laden Sie die `eks-connector`-Clusterrollenvorlage herunter.

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/eks-connector/manifests/eks-connector-console-roles/eks-connector-clusterrole.yaml
   ```

2. Bearbeiten Sie die YAML-Datei der Clusterrollen-Vorlage. Ersetzen Sie die Referenzen von `%IAM_ARN%` durch den Amazon-Ressourcennamen (ARN) Ihres IAM-Prinzipals.
3. Wenden Sie die Amazon-EKS-Connector-Clusterrolle YAML auf Ihren Kubernetes-Cluster an.

   ```
   kubectl apply -f eks-connector-clusterrole.yaml
   ```

Damit ein IAM-Prinzipal Kubernetes-Ressourcen auf der Amazon-EKS-Konsole ansehen kann, muss der Prinzipal einer Kubernetes-`role` oder `clusterrole`- mit den nötigen Ressourcen zugeordnet sein, um die Ressourcen zu lesen. Weitere Informationen finden Sie unter Using RBAC authorization in der Kubernetes-Dokumentation.

Konfigurieren Sie einen IAM-Prinzipal für den Zugriff auf den verbundenen Cluster wie folgt

1. Sie können einer dieser Beispielmanifestdateien herunterladen, um eine `clusterrole` und `clusterrolebinding` oder eine `role` und `rolebinding` zu erstellen:

Kubernetes-Ressourcen in allen Namespaces anzeigen

Die `eks-connector-console-dashboard-full-access-clusterrole`-Clusterrolle ermöglicht den Zugriff auf alle Namespaces und Ressourcen, die in der Konsole visualisiert werden können. Sie können den Namen des `role`, `clusterrole` und der entsprechenden Bindung ändern, bevor Sie ihn auf Ihren Cluster anwenden. Verwenden Sie den folgenden Befehl, um eine Beispieldatei herunterzuladen.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/eks-connector/manifests/eks-connector-console-roles/eks-connector-console-dashboard-full-access-group.yaml
```

Kubernetes-Ressourcen in einem bestimmten Namespace anzeigen

Der Namespace in dieser Datei ist `default`, wenn Sie also einen anderen Namespace angeben möchten, bearbeiten Sie die Datei, bevor Sie sie auf Ihren Cluster anwenden. Verwenden Sie den folgenden Befehl, um eine Beispieldatei herunterzuladen.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/eks-connector/manifests/eks-connector-console-roles/eks-connector-console-dashboard-restricted-access-group.yaml
```

2. Bearbeiten Sie die YAML-Datei mit vollem oder eingeschränktem Zugriff, um die Referenzen von `%IAM_ARN%` durch den Amazon-Ressourcennamen (ARN) Ihres IAM-Prinzipals zu ersetzen.
3. Wenden Sie die YAML-Dateien mit vollem oder eingeschränktem Zugriff auf Ihren Kubernetes-Cluster an. Ersetzen Sie den Wert der YAML-Datei durch Ihren eigenen.

```
kubectl apply -f eks-connector-console-dashboard-full-access-group.yaml
```

Informationen zum Anzeigen von Kubernetes-Ressourcen in Ihrem verbundenen Cluster finden Sie unter Anzeigen der Kubernetes-Ressourcen. Daten für einige Ressourcentypen auf der Registerkarte Resources (Ressourcen) sind nicht für verbundene Cluster verfügbar.

Einen Cluster abmelden

Wenn Sie mit der Verwendung eines verbundenen Clusters fertig sind, können Sie die Registrierung aufheben. Nach der Abmeldung ist der Cluster in der Amazon-EKS-Konsole nicht mehr sichtbar.

Sie müssen über die folgenden Berechtigungen verfügen, um die DeregisterCluster-API aufzurufen:

- `eks:DeregisterCluster`
- `ssm:DeleteActivation`
- `ssm:DeregisterManagedInstance`

Dieser Vorgang umfasst zwei Schritte: Abmeldung des Clusters von Amazon EKS und Deinstallation des Agents „eks-connector“ im Cluster.

Aufheben der Registrierung des Kubernetes-Clusters

AWS CLI

Voraussetzungen

- AWS CLI muss installiert sein. Informationen zur Installation oder Aktualisierung finden Sie unter Installieren des AWS CLI.
- Stellen Sie sicher, dass die Amazon-EKS-Connector-Agentenrolle erstellt wurde.

Den verbundenen Cluster deregistrieren.

```
aws eks deregister-cluster \
    --name my-cluster \
    --region region-code
```

AWS Management Console

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie Clusters (Cluster) aus.
3. Wählen Sie auf der Seite Cluster den verbundenen Cluster aus und wählen Sie Deregistrieren.
4. Bestätigen Sie, dass Sie den Cluster deregistrieren möchten.

eksctl

Voraussetzungen

- `eksctl`-Version `0.68` oder höher muss installiert sein. Informationen zur Installation oder Aktualisierung finden Sie unter Erste Schritte mit Amazon EKS – `eksctl`.
- Stellen Sie sicher, dass die Amazon-EKS-Connector-Agentenrolle erstellt wurde.

So melden Sie Ihre Cluster bei **`eksctl`** ab

- Geben Sie für die Connector-Konfiguration Ihre Amazon-EKS-Connector-Agent-IAM-Rolle an. Weitere Informationen finden Sie unter Erforderliche IAM-Rollen für Amazon EKS Connector.

```
eksctl deregister cluster --name my-cluster
```

Bereinigen der Ressourcen in Ihrem Kubernetes-Cluster

Helm

- Führen Sie zur Deinstallation des Agents den folgenden Befehl aus.

```
helm -n eks-connector uninstall eks-connector
```

YAML manifest

1. Löschen Sie die Amazon-EKS-Connector-YAML-Datei aus Ihrem Kubernetes-Cluster.

```
kubectl delete -f eks-connector.yaml
```

2. Wenn Sie `clusterrole` oder `clusterrolebindings` für einen zusätzlichen IAM-Prinzipal für den Zugriff auf den Cluster erstellt haben, stellen Sie sicher, dass Sie ihn aus Ihrem Kubernetes-Cluster löschen.

Beheben von Problemen in Amazon EKS Connector

Dieses Thema behandelt einige der häufigsten Fehler, die bei der Verwendung des Amazon EKS Connectors auftreten können, einschließlich Anleitungen zu deren Behebung und Umgehungen.

Grundlegende Fehlersuche

In diesem Abschnitt werden Schritte zur Diagnose eines nicht eindeutigen Problems beschrieben.

Prüfen Sie den Status von Amazon EKS Connector

Überprüfen Sie den Status von Amazon EKS Connector.

```
kubectl get pods -n eks-connector
```

Überprüfen Sie die Protokolle des Amazon-EKS-Connectors

Der Amazon-EKS-Connector Pod besteht aus drei Containern. Führen Sie die folgenden Befehle aus, um vollständige Protokolle für alle diese Container abzurufen, damit Sie sie überprüfen können:

- `connector-init`

  ```
  kubectl logs eks-connector-0 --container connector-init -n eks-connector
  kubectl logs eks-connector-1 --container connector-init -n eks-connector
  ```

- `connector-proxy`

  ```
  kubectl logs eks-connector-0 --container connector-proxy -n eks-connector
  kubectl logs eks-connector-1 --container connector-proxy -n eks-connector
  ```

- `connector-agent`

  ```
  kubectl exec eks-connector-0 --container connector-agent -n eks-connector -- cat /
  var/log/amazon/ssm/amazon-ssm-agent.log
  kubectl exec eks-connector-1 --container connector-agent -n eks-connector -- cat /
  var/log/amazon/ssm/amazon-ssm-agent.log
  ```

Bringen Sie den effektiven Clusternamen in Erfahrung

Amazon-EKS-Cluster werden von `clusterName` innerhalb eines einzigen AWS -Kontos und einer AWS-Region-Region eindeutig identifiziert. Wenn Sie über mehrere verbundene Cluster in Amazon EKS verfügen, können Sie bestätigen, bei welchem Amazon-EKS-Cluster der aktuelle Kubernetes-Cluster registriert ist. Geben Sie dafür Folgendes ein, um den `clusterName` des aktuellen Clusters herauszufinden.

```
kubectl exec eks-connector-0 --container connector-agent -n eks-connector \
  -- cat /var/log/amazon/ssm/amazon-ssm-agent.log | grep -m1 -oE "eks_c:[a-zA-Z0-9_-]+"
 | sed -E "s/^.*eks_c:([a-zA-Z0-9_-]+)_[a-zA-Z0-9]+.*$/\1/"
kubectl exec eks-connector-1 --container connector-agent -n eks-connector \
  -- cat /var/log/amazon/ssm/amazon-ssm-agent.log | grep -m1 -oE "eks_c:[a-zA-Z0-9_-]+"
 | sed -E "s/^.*eks_c:([a-zA-Z0-9_-]+)_[a-zA-Z0-9]+.*$/\1/"
```

Verschiedene Befehle

Die folgenden Befehle sind nützlich, um Informationen abzurufen, die Sie zur Behebung von Problemen benötigen.

- Verwenden Sie den folgenden Befehl, um Bilder zu sammeln, die von Pods im Amazon EKS Connector verwendet werden.

  ```
  kubectl get pods -n eks-connector -o jsonpath="{.items[*].spec.containers[*].image}"
   | tr -s '[[:space:]]' '\n'
  ```

- Verwenden Sie den folgenden Befehl, um die Knotennamen zu ermitteln, auf denen Amazon-EKS-Connector ausgeführt wird.

  ```
  kubectl get pods -n eks-connector -o jsonpath="{.items[*].spec.nodeName}" | tr -s
   '[[:space:]]' '\n'
  ```

- Führen Sie den folgenden Befehl aus, um Ihre Kubernetes-Client- und Server-Versionen abzurufen.

  ```
  kubectl version
  ```

- Führen Sie den folgenden Befehl aus, um Informationen zu Ihren Knoten abzurufen.

  ```
  kubectl get nodes -o wide --show-labels
  ```

Helm-Ausgabe: 403 Forbidden

Wenn Sie beim Ausführen von Helm-Installationsbefehlen die folgende Fehlermeldung erhalten haben:

```
Error: INSTALLATION FAILED: unexpected status from HEAD request to https://
public.ecr.aws/v2/eks-connector/eks-connector-chart/manifests/0.0.6: 403 Forbidden
```

Sie können die folgende Zeile ausführen, um das Problem zu beheben:

```
docker logout public.ecr.aws
```

Konsolenfehler: Der Cluster steckt im Status „Ausstehend“ fest

Wenn der Cluster in dem `Pending` Status auf der Amazon EKS-Konsole hängen bleibt, nachdem Sie ihn registriert haben, kann das daran liegen, dass der Amazon EKS-Connector den Cluster AWS noch nicht erfolgreich verbunden hat. Für einen registrierten Cluster bedeutet der `Pending`-Zustand, dass die Verbindung nicht erfolgreich hergestellt wurde. Um dieses Problem zu lösen, stellen Sie sicher, dass Sie das Manifest auf den Kubernetes-Ziel-Cluster angewendet haben. Wenn Sie es auf den Cluster angewendet haben, der Cluster sich aber immer noch im `Pending`-Zustand befindet, ist `eks-connector`-StatefulSet möglicherweise fehlerhaft. Informationen zum Beheben dieses Problems finden Sie in diesem Thema unter Amazon EKS-Connector-Pods stürzen laufend ab .

Konsolenfehler: **`User “system:serviceaccount:eks-connector:eks-connector” can't impersonate resource “users” in API group “”`** im Cluster-Bereich

Der Amazon EKS Connector verwendet Kubernetes-Benutzeridentitätswechsel, um im Namen von IAM-Prinzipalen aus der AWS Management Console zu handeln. Jedem Principal, der über das AWS `eks-connector` Dienstkonto auf die Kubernetes API zugreift, muss die Erlaubnis erteilt werden, sich als der entsprechende Kubernetes Benutzer mit einem IAM-ARN als Benutzernamen auszugeben. Kubernetes In den folgenden Beispielen wird der IAM-ARN einem Kubernetes-Benutzer zugeordnet.

- Der IAM-Benutzer *`john`* aus dem AWS Konto *`111122223333`* ist einem Benutzer zugeordnet. Kubernetes Bewährte Methoden für IAM empfehlen, dass Sie Rollen statt Benutzern Berechtigungen gewähren.

  ```
  arn:aws:iam::111122223333:user/john
  ```

- Die IAM-Rolle *`admin`* aus dem AWS Konto *`111122223333`* ist einem Benutzer zugeordnet: Kubernetes

```
arn:aws:iam::111122223333:role/admin
```

Das Ergebnis ist ein IAM-Rollen-ARN anstelle des AWS STS Sitzungs-ARN.

Anweisungen zum Konfigurieren der `ClusterRole` und `ClusterRoleBinding`, um dem `eks-connector`-Servicekonto die Berechtigung zu erteilen, sich als zugeordneter Benutzer auszugeben, finden Sie unter Gewähren des Zugriffs für einen IAM-Prinzipal zum Anzeigen von Kubernetes-Ressource eines Clusters. Stellen Sie sicher, dass `%IAM_ARN%` in der Vorlage durch den IAM-ARN des AWS Management Console -IAM-Prinzipals ersetzt wird.

Konsolenfehler: **`[...] is forbidden: User [...] cannot list resource "[...] in API group"`** im Cluster-Bereich

Betrachten Sie das folgende Problem. Der Amazon EKS-Connector hat erfolgreich die Identität des anfordernden AWS Management Console IAM-Prinzipals im Zielcluster angenommen. Kubernetes Der imitierte Prinzipal hat jedoch keine RBAC-Berechtigung für Kubernetes-API-Vorgänge.

Um dieses Problem zu beheben, gibt es zwei Methoden, um zusätzlichen Benutzern Berechtigungen zu erteilen. Wenn Sie „eks-connector“ zuvor über Helm-Chart installiert haben, können Sie Benutzern ganz einfach Zugriff gewähren, indem Sie den folgenden Befehl ausführen. Ersetzen Sie `userARN1` und `userARN2` durch eine Liste der ARNs der IAM-Rollen, um Zugriff auf die Kubernetes-Ressourcen zu gewähren:

```
helm upgrade eks-connector oci://public.ecr.aws/eks-connector/eks-connector-chart \
    --reuse-values \
    --set 'authentication.allowedUserARNs={userARN1,userARN2}'
```

Als Cluster-Administrator können Sie auch einzelnen Kubernetes-Benutzern die entsprechende Stufe von RBAC-Berechtigungen gewähren. Weitere Informationen und Beispiele finden Sie unter Gewähren des Zugriffs für einen IAM-Prinzipal zum Anzeigen von Kubernetes-Ressource eines Clusters.

Konsolenfehler: Amazon EKS kann nicht mit Ihrem Kubernetes-Cluster-API-Server kommunizieren. Der Cluster muss sich im ACTIVE-Status befinden, um eine erfolgreiche Verbindung zu erreichen. Try again in few minutes. (Versuchen Sie es in ein paar Minuten erneut.)

Wenn der Amazon-EKS-Service nicht mit dem Amazon EKS-Connector im Zie-Cluster kommunizieren kann, kann dies einen der folgenden Gründe haben:

- Der Amazon-EKS-Connector im Ziel-Cluster ist fehlerhaft.
- Schlechte Konnektivität oder eine unterbrochene Verbindung zwischen dem Ziel-Cluster und der AWS-Region.

Um dieses Problem zu beheben, überprüfen Sie die Amazon-EKS-Connector-Protokolle. Wenn kein Fehler für den Amazon-EKS-Connector angezeigt wird, versuchen Sie die Verbindung nach einigen Minuten erneut. Wenn Sie regelmäßig hohe Latenz oder unregelmäßige Konnektivität für den Zielcluster feststellen, sollten Sie erwägen, den Cluster erneut bei einem Cluster zu registrieren, der sich in Ihrer AWS-Region Nähe befindet.

Amazon EKS-Connector-Pods stürzen laufend ab

Es gibt viele Gründe, die dazu führen können, dass ein Amazon-EKS-Connector Pod in den `CrashLoopBackOff`-Status übergeht. Dieses Problem betrifft wahrscheinlich den `connector-init`-Container. Prüfen Sie den Status von Amazon-EKS-Connector Pod.

```
kubectl get pods -n eks-connector
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME              READY   STATUS                  RESTARTS   AGE
eks-connector-0   0/2     Init:CrashLoopBackOff   1          7s
```

Wenn Ihre Ausgabe der vorherigen Ausgabe ähnlich ist, lesen Sie Überprüfen Sie die Protokolle des Amazon-EKS-Connectors, um das Problem zu beheben.

Failed to initiate eks-connector: InvalidActivation

Wenn Sie den Amazon-EKS-Connector zum ersten Mal starten, registriert er ein `activationId` und `activationCode` bei Amazon Web Services. Die Registrierung schlägt möglicherweise fehl, was dazu führen kann, dass der `connector-init`-Container mit einem Fehler ähnlich dem folgenden abstürzt.

```
F1116 20:30:47.261469       1 init.go:43] failed to initiate eks-connector:
 InvalidActivation:
```

Berücksichtigen Sie die folgenden Ursachen und empfohlenen Korrekturen, um dieses Problem zu beheben:

- Die Registrierung ist möglicherweise fehlgeschlagen, da `activationId` und `activationCode` nicht in Ihrer Manifest-Datei enthalten sind. Stellen Sie in diesem Fall sicher, dass es sich um die richtigen Werte handelt, die von der `RegisterCluster`-API-Operation zurückgegeben wurden, und dass sich `activationCode` in der Manifest-Datei befindet. Der `activationCode` wird zu Kubernetes-Geheimnissen hinzugefügt, daher muss er `base64`-codiert sein. Weitere Informationen finden Sie unter Schritt 1: Registrieren des Clusters.
- Die Registrierung ist möglicherweise fehlgeschlagen, da Ihre Aktivierung abgelaufen ist. Dies liegt daran, dass Sie den Amazon-EKS-Connector aus Sicherheitsgründen innerhalb von drei Tagen nach der Registrierung des Clusters aktivieren müssen. Um dieses Problem zu lösen, stellen Sie sicher, dass Sie vor Ablaufdatum und -uhrzeit das Amazon-EKS-Connector-Manifest auf den Kubernetes-Ziel-Cluster angewendet haben. Um das Ablaufdatum der Aktivierung zu bestätigen, rufen Sie die `DescribeCluster`-API-Operation auf.

  ```
  aws eks describe-cluster --name my-cluster
  ```

  In der folgenden Beispielantwort wird das Ablaufdatum und die Uhrzeit als `2021-11-12T22:28:51.101000-08:00` aufgezeichnet.

  ```
  {
      "cluster": {
          "name": "my-cluster",
          "arn": "arn:aws:eks:region:111122223333:cluster/my-cluster",
          "createdAt": "2021-11-09T22:28:51.449000-08:00",
          "status": "FAILED",
  ```

```
        "tags": {
        },
        "connectorConfig": {
            "activationId": "00000000-0000-0000-0000-000000000000",
            "activationExpiry": "2021-11-12T22:28:51.101000-08:00",
            "provider": "OTHER",
            "roleArn": "arn:aws:iam::111122223333:role/my-connector-role"
        }
    }
}
```

Wenn das `activationExpiry` erreicht wurde, melden Sie den Cluster ab und registrieren Sie ihn erneut. Dadurch wird eine neue Aktivierung generiert.

Im Cluster-Knoten fehlt die ausgehende Konnektivität

Um ordnungsgemäß zu funktionieren, benötigt der Amazon-EKS-Connector eine ausgehende Konnektivität zu mehreren AWS -Endpunkten. Sie können einen privaten Cluster ohne ausgehende Konnektivität nicht mit einem Ziel- AWS-Region verbinden. Um dieses Problem zu lösen, müssen Sie die erforderliche ausgehende Konnektivität hinzufügen. Weitere Informationen zu den Anforderungen für Konnektoren finden Sie unter Überlegungen zum Amazon EKS Connector.

Amazon-EKS-Connectors Pods befinden sich im **`ImagePullBackOff`**-Zustand

Wenn Sie den `get pods`-Befehl ausführen und sich Pods im `ImagePullBackOff`-Zustand befinden, können diese nicht ordnungsgemäß funktionieren. Wenn sich die Amazon-EKS-Connectors Pods im `ImagePullBackOff`-Zustand befinden, können diese nicht ordnungsgemäß funktionieren. Prüfen Sie den Status Ihres Amazon-EKS-Connectors Pods.

```
kubectl get pods -n eks-connector
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME              READY   STATUS                  RESTARTS   AGE
eks-connector-0   0/2     Init:ImagePullBackOff   0          4s
```

Die standardmäßige Amazon-EKS-Connector-Manifest-Datei verweist auf Images aus der öffentlichen Amazon-ECR-Galerie. Es ist möglich, dass der Ziel-Cluster Kubernetes keine Images

aus der öffentlichen Amazon-ECR-Galerie abrufen kann. Beheben Sie entweder das Image-Abrufproblem der öffentlichen Amazon-ECR-Galerie oder erwägen Sie eine Spiegelung der Images in der privaten Container-Registry Ihrer Wahl.

Häufig gestellte Fragen

F: Wie funktioniert die zugrunde liegende Technologie hinter dem Amazon-EKS-Connector?

A: Der Amazon-EKS-Connector basiert auf dem AWS Systems Manager (Systems Manager)-Agent. Der Amazon-EKS-Connector wird als `StatefulSet` auf Ihrem Kubernetes-Cluster ausgeführt. Er stellt eine Verbindung her und ermöglicht die Kommunikation zwischen dem API-Server Ihres Clusters und Amazon Web Services. Dadurch werden Clusterdaten in der Amazon EKS-Konsole angezeigt, bis Sie den Cluster von AWS trennen. Der Systems-Manager-Agent ist ein Open-Source-Projekt. Weitere Informationen zu diesem Projekt finden Sie auf der GitHub-Projektseite.

F: Ich habe einen On-Premises-Kubernetes-Cluster, den ich verbinden möchte. Muss ich Firewall-Ports öffnen, um ihn zu verbinden?

A: Nein, Sie müssen keine Firewall-Ports öffnen. Der AWS-Regionen-Cluster benötigt nur eine ausgehende Verbindung zu Kubernetes. AWS-Services greifen niemals auf Ressourcen in Ihrem On-Premises-Netzwerk zu. Der Amazon EKS Connector läuft auf Ihrem Cluster und leitet die Verbindung mit AWS ein. Wenn die Clusterregistrierung abgeschlossen ist, gibt AWS Befehle nur an den Amazon EKS Connector aus, nachdem Sie eine Aktion von der Amazon-EKS-Konsole aus gestartet haben, die Informationen vom Kubernetes-API-Server auf Ihrem Cluster benötigt.

F: Welche Daten werden vom Amazon EKS Connector über meinen Cluster an AWS gesendet?

A: Der Amazon EKS Connector sendet technische Informationen, die erforderlich sind, damit Ihr Cluster in AWS registriert wird. Er sendet auch Cluster- und Workload-Metadaten für die Funktionen der Amazon EKS-Konsole, die Kunden anfordern. Der Amazon-EKS-Connector erfasst oder sendet diese Daten nur, wenn Sie eine Aktion von der Amazon-EKS-Konsole oder der Amazon-EKS-API aus starten, die das Senden der Daten an AWS erfordert. Neben der Kubernetes-Versionsnummer speichert AWS standardmäßig keine Daten. Er speichert Daten nur, wenn Sie dies autorisieren.

F: Kann ich einen Cluster außerhalb einer AWS-Region verbinden?

A: Ja, Sie können einen Cluster von jedem Standort aus mit Amazon EKS verbinden. Darüber hinaus kann sich der Amazon-EKS-Service in jeder AWS-öffentlichen kommerziellen AWS-Region befinden. Dies funktioniert mit einer gültigen Netzwerkverbindung vom Cluster zur Ziel-AWS-Region. Wir

empfehlen Ihnen, eine AWS-Region auszuwählen, die Ihrem Cluster-Standort am nächsten ist, um die Leistungsoptimierung der Benutzeroberfläche zu gewährleisten. Wenn beispielsweise ein Cluster in Tokio ausgeführt wird, verbinden Sie zwecks niedriger Latenz den Cluster mit der AWS-Region in Tokio (also mit der AWS-Region `ap-northeast-1`). Sie können einen Cluster von jedem Standort aus mit Amazon EKS in jedem öffentlichen kommerziellen AWS-Regionen verbinden, mit Ausnahme von China oder GovCloud AWS-Regionen.

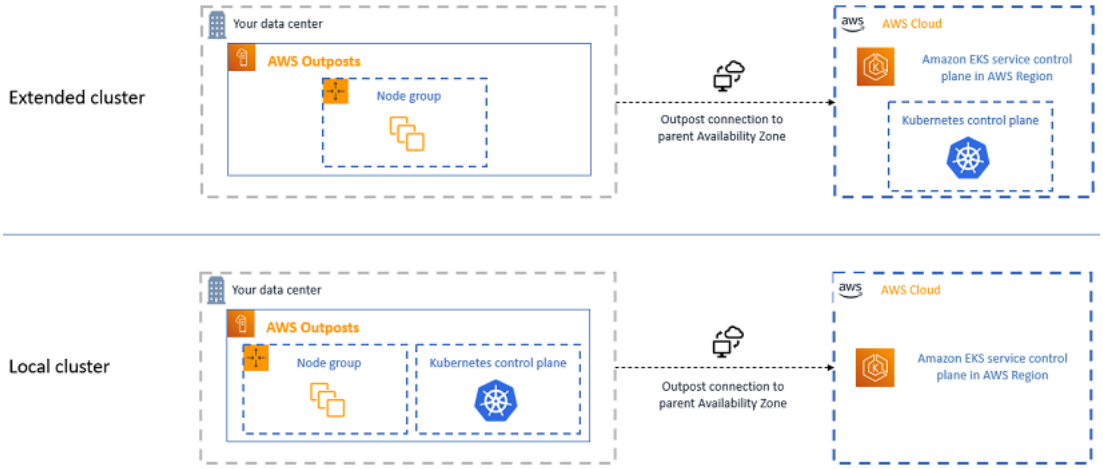
auf Amazon EKSAWS Outposts

Sie können Amazon EKS verwenden, um On-Premises-KubernetesAnwendungen auf AWS Outposts auszuführen. Sie können Amazon EKS auf Outposts folgendermaßen bereitstellen:

- Erweiterte Cluster – Führen Sie die Kubernetes-Steuerebene in einer AWS-Region und Knoten auf Ihrem Outpost aus.
- Lokale Cluster – Führen Sie die Kubernetes-Steuerebene und Knoten auf Ihrem Outpost aus.

Für beide Bereitstellungsoptionen wird die Kubernetes-Steuerebene vollständig von AWS verwaltet. Sie können dieselben Amazon-EKS-APIs, -Tools und -Konsolen verwenden, die Sie in der Cloud zum Erstellen und Ausführen von Amazon EKS auf Outposts verwenden.

Das folgende Diagramm zeigt diese Optionen für die Bereitstellung.



Wann die einzelnen Bereitstellungsoptionen verwendet werden sollten

Sowohl lokale als auch erweiterte Cluster sind allgemeine Bereitstellungsoptionen und können für eine Reihe von Anwendungen verwendet werden.

Mit lokalen Clustern können Sie den gesamten Amazon-EKS-Cluster lokal auf Outposts ausführen. Mit dieser Option können Sie das Risiko von Ausfallzeiten bei Anwendungen verringern, die durch vorübergehende Netzwerkunterbrechungen in der Cloud entstehen können. Diese Netzwerkunterbrechungen können durch Glasfaserausfälle oder Wetterereignisse verursacht werden. Da der gesamte Amazon-EKS-Cluster lokal auf Outposts ausgeführt wird, bleiben Anwendungen

verfügbar. Sie können Cluster-Vorgänge während Netzwerkunterbrechungen mit der Cloud durchführen. Weitere Informationen finden Sie unter Vorbereitungen für Netzwerkunterbrechungen. Wenn Sie Bedenken hinsichtlich der Qualität der Netzwerkverbindung zwischen Ihren Outposts und dem übergeordneten AWS-Region haben und eine hohe Verfügbarkeit durch Netzwerkunterbrechungen benötigen, verwenden Sie die Bereitstellungsoption des lokalen Clusters.

Mit erweiterten Clustern können Sie Kapazität auf Ihrem Outpost sparen, da die Kubernetes-Steuerebene im übergeordneten AWS-Region ausgeführt wird. Diese Option ist geeignet, wenn Sie in eine zuverlässige, redundante Netzwerkverbindung von Ihrem Outpost zum AWS-Region investieren können. Die Qualität der Netzwerkverbindung ist für diese Option entscheidend. Die Art und Weise, wie Kubernetes Netzwerkunterbrechungen zwischen der Kubernetes-Steuerebene und den Knoten handhabt, kann zu Ausfallzeiten der Anwendung führen. Weitere Informationen zum Verhalten von Kubernetes finden Sie unter Planung, Vorkaufsrecht und Bereinigung in der Kubernetes-Dokumentation.

Vergleich der Optionen für die Bereitstellung

Die folgende Tabelle vergleicht die Unterschiede zwischen den beiden Optionen.

Funktion	Erweiterter Cluster	Lokaler Cluster
Kubernetes Position der Steuerebene	AWS-Region	Outpost
Kubernetes-Steuerebene-Konto	AWS-Konto	Ihr Konto
Regionale Verfügbarkeit	Siehe Service-Endpunkte	USA Ost (Ohio), USA Ost (Nord-Virginia),USA West (Nordkalifornien), USA West (Oregon), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Kanada (Zentral), Europa (Frankfurt), Europa (Irland), Europa (London), Naher Osten (Bahrain),

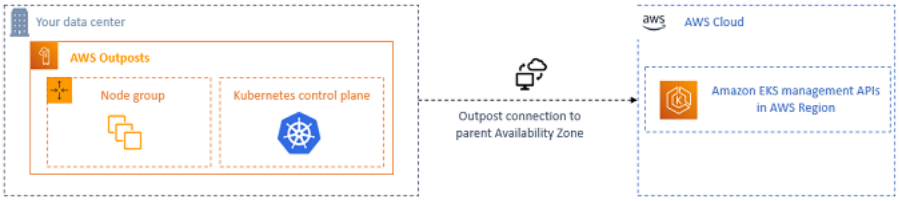
Funktion	Erweiterter Cluster	Lokaler Cluster
		Naher Osten (Bahrain) und Südamerika (São Paulo)
Kubernetes-Nebenversionen	Unterstützte Amazon-EKS-Versionen.	Unterstützte Amazon-EKS-Versionen.
Plattformversionen	Siehe Amazon-EKS-Plattformversionen	Siehe Versionen der lokalen Amazon EKS-Clusterplattform
Outpost-Formfaktoren	Outpost-Racks	Outpost-Racks
Benutzeroberflächen	AWS Management Console, AWS CLI, Amazon-EKS-APIs, `eksctl`, AWS CloudFormation und Terraform	AWS Management Console, AWS CLI, Amazon-EKS-APIs, `eksctl`, AWS CloudFormation und Terraform
Verwaltete Richtlinien	AmazonEKSClusterPolicy und AmazonEKSServiceRolePolicy	AmazonEKSLocalOutpostClusterPolicy und AmazonEKSLocalOutpostServiceRolePolicy
Cluster-VPC und Subnetze	Siehe Amazon EKS: VPC- und Subnetz-Anforderungen und -Überlegungen	Siehe Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen
Cluster-Endpunktzugriff	Öffentlich oder privat oder beides	Nur privat
Kubernetes-API-Serverauthentifizierung	AWS Identity and Access Management (IAM) und OIDC	IAM und `x.509`-Zertifikate
Knotentypen	Nur selbstverwaltet	Nur selbstverwaltet
Knoten-Berechnungstypen	Amazon EC2 On-Demand	Amazon EC2 On-Demand
Knoten-Speichertypen	Amazon EBS `gp2` und lokale NVMe-SSD	Amazon EBS `gp2` und lokale NVMe-SSD

Funktion	Erweiterter Cluster	Lokaler Cluster
Amazon EKS-optimierte AMIs	Amazon Linux, Windows und Bottlerocket	Nur Amazon Linux
IP-Versionen	Nur `IPv4`	Nur `IPv4`
Add-Ons	Amazon-EKS-Add-Ons oder selbstverwaltete Add-Ons	Nur Selbstverwaltete Add-Ons
Standard-Container-Netzwerkschnittstelle	Amazon VPC CNI plugin for Kubernetes	Amazon VPC CNI plugin for Kubernetes
Kubernetes-Steuerebene-Protokolle	Amazon CloudWatch Logs	Amazon CloudWatch Logs
Load Balancing	Verwenden Sie den AWS Load Balancer Controller nur zum Bereitstellen von Application Load Balancern (keine Network Load Balancer)	Verwenden Sie den AWS Load Balancer Controller nur zum Bereitstellen von Application Load Balancern (keine Network Load Balancer)
Secrets-Umschlagverschlüsselung	Siehe Aktivieren der Secret-Verschlüsselung in einem vorhandenen Cluster	Nicht unterstützt
IAM-Rollen für Servicekonten	Siehe IAM-Rollen für Servicekonten	Nicht unterstützt
Fehlerbehebung	Siehe Amazon-EKS-Fehlerbehebung	Siehe Problembehandlung bei lokalen Clustern für Amazon EKS auf AWS Outposts

Lokale Cluster für Amazon EKS auf AWS Outposts

Mit lokalen Clustern können Sie Ihren gesamten Amazon-EKS-Cluster lokal auf AWS Outposts ausführen. Auf diese Weise wird das Risiko von Anwendungsausfällen, die sich aus vorübergehenden Unterbrechungen der Netzwerkverbindung zur Cloud ergeben können,

minimiert. Diese Verbindungsunterbrechungen können durch Glasfaserunterbrechungen oder Wetterereignisse verursacht werden. Weil das gesamte Kubernetes-Cluster lokal auf Outposts läuft, bleiben Anwendungen verfügbar. Sie können Cluster-Vorgänge während Netzwerkunterbrechungen mit der Cloud durchführen. Weitere Informationen finden Sie unter Vorbereitungen für Netzwerkunterbrechungen. Das folgende Diagramm zeigt eine lokale Cluster-Bereitstellung.



Lokale Cluster sind allgemein für die Verwendung mit Outpost-Racks verfügbar.

Unterstützte AWS-Regionen

Sie können lokale Cluster in den folgenden AWS-Regionen erstellen: USA Ost (Ohio), USA Ost (Nord-Virginia),USA West (Nordkalifornien), USA West (Oregon), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Kanada (Zentral), Europa (Frankfurt), Europa (Irland), Europa (London), Naher Osten (Bahrain), Naher Osten (Bahrain) und Südamerika (São Paulo). Weitere Informationen zu unterstützten Funktionen finden Sie unter Vergleich der Optionen für die Bereitstellung.

Themen

- Erstellen eines lokalen Clusters auf einem Outpost
- Versionen der lokalen Amazon EKS-Clusterplattform
- Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen
- Vorbereitungen für Netzwerkunterbrechungen
- Überlegungen zur Kapazität
- Problembehandlung bei lokalen Clustern für Amazon EKS auf AWS Outposts

Erstellen eines lokalen Clusters auf einem Outpost

Dieses Thema bietet einen Überblick darüber, was beim Ausführen eines lokalen Clusters auf einem Outpost zu beachten ist. Das Thema enthält auch Anweisungen zum Bereitstellen eines lokalen Clusters auf einem Outpost.

Überlegungen

> **Important**
>
> - Diese Überlegungen werden in der zugehörigen Amazon-EKS-Dokumentation nicht wiederholt. Wenn andere Themen der Amazon-EKS-Dokumentation mit den hier aufgeführten Überlegungen in Konflikt stehen, befolgen Sie die Überlegungen hier.
> - Diese Überlegungen sind freibleibend und können sich häufig ändern. Wir empfehlen Ihnen daher, dieses Thema regelmäßig zu überprüfen.
> - Viele der Überlegungen unterscheiden sich von den Überlegungen zum Erstellen eines Clusters in der AWS Cloud.

- Lokale Cluster unterstützen nur Outpost-Racks. Ein einzelner lokaler Cluster kann über mehrere physische Outpost-Racks laufen, die aus einem einzigen logischen Outpost bestehen. Ein einzelner lokaler Cluster kann nicht über mehrere logische Outposts hinweg ausgeführt werden. Jeder logische Outpost hat einen einzigen Outpost-ARN.
- Lokale Cluster führen und verwalten die Kubernetes-Steuerebenen in Ihrem Konto auf dem Outpost. Sie können keine Workloads auf den Instances der Kubernetes-Steuerebene ausführen oder die Komponenten der Kubernetes-Steuerebene ändern. Diese Knoten werden vom Amazon-EKS-Service verwaltet. Änderungen an der Kubernetes-Steuerebene bleiben nicht durch automatische Amazon-EKS-Verwaltungsaktionen, wie z. B. Patching, erhalten.
- Lokale Cluster unterstützen selbstverwaltete Add-Ons und selbstverwaltete Amazon Linux-Knotengruppen. Die Amazon VPC CNI plugin for Kubernetes-, `kube-proxy`-, und CoreDNS-Add-Ons werden automatisch auf lokalen Clustern installiert.
- Lokale Cluster erfordern die Verwendung von Amazon EBS auf Outposts. In Ihrem Outpost muss Amazon EBS für die Kubernetes-Aufbewahrung der Steuerebene verfügbar sein.
- Lokale Cluster verwenden Amazon EBS auf Outposts. In Ihrem Outpost muss Amazon EBS für die Kubernetes-Aufbewahrung der Steuerebene verfügbar sein. Outposts unterstützen nur Amazon EBS `gp2`-Volumes.
- Amazon-EBS-gestützte Kubernetes `PersistentVolumes` werden mit dem Amazon-EBS-CSI-Treiber unterstützt.

Voraussetzungen

- Vertrautheit mit den Outposts deployment options (Optionen für die Bereitstellung von Outposts), Überlegungen zur Kapazität, und Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen.
- Ein vorhandener Outpost. Weitere Informationen finden Sie unter Was ist AWS Outposts.
- Das `kubectl`-Befehlszeilen-Tool ist auf Ihrem Computer oder AWS CloudShell installiert. Die Version kann der Kubernetes-Version Ihres Clusters entsprechen oder eine Nebenversion älter oder neuer sein. Wenn Ihre Clusterversion beispielsweise `1.29` ist, können Sie `kubectl`-Version `1.28`, `1.29`, oder `1.30` damit verwenden. Informationen zum Installieren oder Aktualisieren von `kubectl` finden Sie unter Installieren oder Aktualisieren von `kubectl`.
- Version `2.12.3` oder höher oder Version `1.27.160` oder höher von AWS Command Line Interface (AWS CLI), die auf Ihrem Gerät installiert und konfiguriert ist, oder. AWS CloudShell Um Ihre aktuelle Version zu überprüfen, verwenden Sie **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Paket-Manager wie `yum`, `apt-get` oder Homebrew für macOS sind oft mehrere Versionen hinter der neuesten Version von AWS CLI. Informationen zur Installation der neuesten Version von finden Sie unter Installation, Aktualisierung und Deinstallation der AWS CLI und Schnellkonfiguration mit aws configure im AWS Command Line Interface -Benutzerhandbuch. Die AWS CLI Version, in der installiert ist, AWS CloudShell kann auch mehrere Versionen hinter der neuesten Version liegen. Informationen zur Aktualisierung finden Sie unter Installation AWS CLI in Ihrem Home-Verzeichnis im AWS CloudShell Benutzerhandbuch.
- Ein IAM-Prinzipal (Benutzer oder Rolle) mit Berechtigungen zum Erstellen (`create`) und Beschreiben (`describe`) eines Amazon-EKS-Clusters. Weitere Informationen finden Sie unter Erstellen Sie einen lokalen Kubernetes-Cluster auf einem Outpost und Auflisten oder Beschreiben aller Cluster.

Wenn ein lokaler Amazon-EKS-Cluster erstellt wird, wird der IAM-Prinzipal, der den Cluster erstellt, dauerhaft hinzugefügt. Der Prinzipal wird ausdrücklich als Administrator zur Kubernetes-RBAC-Autorisierungstabelle hinzugefügt. Diese Entität hat `system:masters`-Berechtigungen. Die Identität dieser Entität ist in Ihrer Cluster-Konfiguration nicht sichtbar. Daher ist es wichtig, die Entität zu notieren, die den Cluster erstellt hat, und sicherzustellen, dass Sie diese keinesfalls löschen. Anfänglich kann nur der Prinzipal, der den Server erstellt hat, Aufrufe an den Kubernetes-API-Server mit `kubectl` tätigen. Wenn Sie die Konsole zum Erstellen des Clusters verwenden, stellen Sie sicher, dass sich dieselben IAM-Anmeldeinformationen in der AWS SDK-Anmeldeinformationskette befinden, wenn Sie `kubectl` Befehle auf Ihrem Cluster ausführen. Nachdem Ihr Cluster erstellt wurde, können Sie anderen IAM-Prinzipalen Zugriff auf Ihren Cluster gewähren.

Einen lokalen Amazon-EKS-Cluster erstellen

Sie können einen lokalen Cluster mit `eksctl`, der AWS Management Console, der AWS CLI, der Amazon-EKS-API, den AWS -SDKs, AWS CloudFormation oder Terraform erstellen.

1. Erstellen eines lokalen Clusters.

   eksctl

   Voraussetzung

   Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

   So erstellen Sie einen Cluster mit **`eksctl`**

   1. Kopieren Sie den folgenden Inhalt auf Ihr Gerät. Ersetzen Sie die folgenden Werte und führen Sie dann den geänderten Befehl aus, um die `outpost-control-plane.yaml`-Datei zu erstellen:
      - Ersetzen Sie *`region-code`* durch die unterstützte AWS-Region, in der Sie Ihren Cluster bereitstellen möchten.
      - Ersetzen Sie *`my-cluster`* durch Ihren Cluster-Namen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
      - Ersetzen Sie *`vpc-ExampleID1`* und *`subnet-ExampleID1`* durch die IDs Ihrer vorhandenen VPC und Ihres Subnetzes. Die VPC und das Subnetz müssen die Anforderungen in Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen erfüllen.
      - Ersetze es *`uniqueid`* durch die ID deines Outposts.
      - Ersetzen Sie *`m5.large`* durch einen Instance-Typ, der auf Ihrem Outpost verfügbar ist. Bevor Sie einen Instance-Typ auswählen, lesen Sie Überlegungen zur Kapazität. Drei Steuerebenen-Instances werden bereitgestellt. Sie können diese Nummer nicht ändern.

```
cat >outpost-control-plane.yaml <<EOF
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code
  version: "1.24"

vpc:
  clusterEndpoints:
    privateAccess: true
  id: "vpc-vpc-ExampleID1"
  subnets:
    private:
      outpost-subnet-1:
        id: "subnet-subnet-ExampleID1"

outpost:
  controlPlaneOutpostARN: arn:aws:outposts:region-code:111122223333:outpost/
op-uniqueid
  controlPlaneInstanceType: m5.large
EOF
```

Eine vollständige Liste aller verfügbaren Optionen und Standardwerte finden Sie unter AWS Outposts -Support und Config file schema (Schema der Konfigurationsdatei) in der `eksctl`-Dokumentation.

2. Erstellen Sie den Cluster mit der Konfigurationsdatei, die Sie im vorherigen Schritt erstellt haben. `eksctl` erstellt eine VPC und ein Subnetz auf Ihrem Outpost, in dem Sie den Cluster bereitstellen können.

```
eksctl create cluster -f outpost-control-plane.yaml
```

Die Clusterbereitstellung dauert mehrere Minuten. Während der Cluster erstellt wird, werden mehrere Ausgabezeilen angezeigt. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

```
[#]  EKS cluster "my-cluster" in "region-code" region is ready
```

> **Tip**
>
> Sie können die meisten Optionen, die beim Erstellen eines Clusters mit `eksctl` angegeben werden können, über den Befehl **`eksctl create cluster --help`** anzeigen. Verwenden Sie eine `config`-Datei, um alle verfügbaren Optionen anzuzeigen. Weitere Informationen finden Sie unter Verwenden von Config-Dateien und im Config-Datei-Schema in der `eksctl`-Dokumentation. Auf GitHub finden Sie Beispiele für Config-Dateien.

`Eksctl` hat automatisch einen Zugriffseintrag für den IAM-Prinzipal (Benutzer oder Rolle) erstellt, der den Cluster erstellt hat, und dem IAM-Prinzipal Administratorberechtigungen für Kubernetes-Objekte im Cluster erteilt. Wenn der Clusterersteller keinen Administratorzugriff auf Kubernetes-Objekte im Cluster haben soll, fügen Sie der vorherigen Konfigurationsdatei den folgenden Text hinzu: **`bootstrapClusterCreatorAdminPermissions: false`** (auf der gleichen Ebene wie `metadata`, `vpc` und `outpost`). Wenn Sie die Option hinzugefügt haben, müssen Sie nach der Clustererstellung einen Zugriffseintrag für mindestens einen IAM-Prinzipal erstellen. Andernfalls kann kein IAM-Prinzipal auf Kubernetes-Objekte im Cluster zugreifen.

AWS Management Console

Voraussetzung

Eine vorhandene VPC und ein Subnetz, die die Amazon-EKS-Anforderungen erfüllen. Weitere Informationen finden Sie unter Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen.

Um Ihren Cluster mit dem zu erstellen AWS Management Console

1. Wenn Sie bereits eine lokale Cluster-IAM-Rolle haben oder Ihren Cluster mit `eksctl` erstellen, können Sie diesen Schritt überspringen. Standardmäßig erstellt `eksctl` eine Rolle für Sie.

   a. Führen Sie den folgenden Befehl aus, um eine JSON-Datei für eine IAM-Vertrauensrichtlinie zu erstellen.

```
cat >eks-local-cluster-role-trust-policy.json <<EOF
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

b. Erstellen Sie die Amazon-EKS-Cluster-IAM-Rolle. Um eine IAM-Rolle zu erstellen, muss dem IAM-Prinzipal, der die Rolle erstellt, die folgende `iam:CreateRole`-Aktion (Berechtigung) zugewiesen werden.

```
aws iam create-role --role-name myAmazonEKSLocalClusterRole --assume-role-
policy-document file://"eks-local-cluster-role-trust-policy.json"
```

c. Hängen Sie die von Amazon EKS verwaltete Richtlinie `AmazonEKSLocalOutpostClusterPolicy` an die Rolle an. Um eine IAM-Richtlinie an einen IAM-Prinzipal anzuhängen, muss der Prinzipal, der die Richtlinie anhängt, eine der folgenden IAM-Aktionen (Berechtigungen) zugewiesen werden: `iam:AttachUserPolicy` oder `iam:AttachRolePolicy`.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSLocalOutpostClusterPolicy --role-name myAmazonEKSLocalClusterRole
```

2. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
3. Stellen Sie oben im Konsolenbildschirm sicher, dass Sie eine unterstützte AWS-Region gewählt haben.
4. Wählen Sie Cluster hinzufügen und dann Erstellen aus.
5. Füllen Sie auf der Seite Configure cluster (Cluster konfigurieren) die folgenden Felder aus oder wählen Sie sie aus:
   - Standort der Kubernetes-Steuerebene – Wählen Sie AWS Outposts aus.

- Outpost-ID – Wählen Sie die ID des Outposts, auf dem Sie Ihre Steuerebene erstellen möchten.
- Instance Type (Instance-Typ) – Wählen Sie einen Instance-Typ aus. Es werden nur die in Ihrem Outpost verfügbaren Instance-Typen angezeigt. In der Dropdown-Liste beschreibt jeder Instance-Typ, für wie viele Knoten der Instance-Typ empfohlen wird. Bevor Sie einen Instance-Typ auswählen, lesen Sie Überlegungen zur Kapazität. Alle Replikate werden mit demselben Instance-Typ bereitgestellt. Sie können den Instance-Typ nicht mehr ändern, nachdem der Cluster erstellt wurde. Drei Steuerebenen-Instances werden bereitgestellt. Sie können diese Nummer nicht ändern.
- Name – Ein Name für Ihren Cluster. Es muss in Ihrem einzigartig sein AWS-Konto. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
- Kubernetes-Version – wählen Sie die Version von Kubernetes, die Sie für Ihren Cluster verwenden möchten. Wir empfehlen, die frühere Version auszuwählen, es sei denn, Sie müssen eine ältere Version verwenden.
- Cluster-Servicerolle — Wählen Sie die Amazon EKS-Cluster-IAM-Rolle aus, die Sie in einem vorherigen Schritt erstellt haben, damit die Kubernetes Kontrollebene AWS Ressourcen verwalten kann.
- Kubernetes-Cluster-Administratorzugriff: Wenn der IAM-Prinzipal (Rolle oder Benutzer), der den Cluster erstellt, Administratorzugriff auf die Kubernetes-Objekte im Cluster haben soll, übernehmen Sie die Standardeinstellung (Zulassen). Amazon EKS erstellt einen Zugriffseintrag für den IAM-Prinzipal und erteilt Cluster-Administratorberechtigungen für den Zugriffseintrag. Weitere Informationen zu Zugriffseinträgen finden Sie unter Zugangseinträge verwalten.

  Wenn nicht der Prinzipal, der den Cluster erstellt, sondern ein anderer IAM-Prinzipal Administratorzugriff auf Kubernetes-Cluster-Objekte haben soll, wählen Sie die Verweigerungsoption aus. Nach der Clustererstellung kann jeder IAM-Prinzipal, der über IAM-Berechtigungen zum Erstellen von Zugriffseinträgen verfügt, Zugriffseinträge für beliebige IAM-Prinzipale hinzufügen, die Zugriff auf Kubernetes-Cluster-Objekte benötigen. Weitere Informationen zu den erforderlichen IAM-Berechtigungen finden Sie in der Service-Authorization-Referenz unter Von Amazon Elastic Kubernetes

Service definierte Aktionen. Wenn Sie die Verweigerungsoption auswählen und keine Zugriffseinträge erstellen, können keine IAM-Prinzipale auf die Kubernetes-Objekte im Cluster zugreifen.

- Tags – (Optional) Fügen Sie Ihrem Cluster beliebige Tags hinzu. Weitere Informationen finden Sie unter Kennzeichnen Ihrer Amazon EKS-Ressourcen .

Wenn Sie mit dieser Seite fertig sind, wählen Sie Weiter aus.

6. Wählen Sie auf der Seite Specify networking (Netzwerk angeben) die Werte für die folgenden Felder aus:
   - VPC – Wählen Sie eine vorhandene VPC aus. Die VPC muss über eine ausreichende Anzahl von IP-Adressen für den Cluster, alle Knoten und andere Kubernetes-Ressourcen, die Sie erstellen möchten, verfügen. Ihre VPC muss die Anforderungen in VPC-Anforderungen und -Überlegungen erfüllen.
   - Subnetze – Standardmäßig sind alle im vorherigen Feld angegebenen Subnetze in der VPC vorausgewählt. Die ausgewählten Subnetze müssen die Anforderungen in Subnetz-Anforderungen und -Überlegungen erfüllen.

     Security groups (Sicherheitsgruppen) – (Optional) Geben Sie eine oder mehrere Sicherheitsgruppen an, die Amazon EKS den erstellten Netzwerkschnittstellen zuordnen soll. Amazon EKS erstellt automatisch eine Sicherheitsgruppe, die die Kommunikation zwischen Ihrem Cluster und Ihrer VPC ermöglicht. Amazon EKS verknüpft diese Sicherheitsgruppe und alle, die Sie wählen, mit den erstellten Netzwerkschnittstellen. Weitere Informationen zu der Cluster-Sicherheitsgruppe, die Amazon EKS erstellt, finden Sie unter Anforderungen und Überlegungen zur Amazon-EKS-Sicherheitsgruppe. Sie können die Regeln in der von Amazon EKS erstellten Cluster-Sicherheitsgruppe ändern. Wenn Sie Ihre eigenen Sicherheitsgruppen hinzufügen möchten, können Sie die ausgewählten nach der Cluster-Erstellung nicht ändern. Damit On-Premises-Hosts mit dem Cluster-Endpunkt kommunizieren können, müssen Sie eingehenden Datenverkehr von der Cluster-Sicherheitsgruppe zulassen. Bei Clustern, die nicht über eine eingehende und ausgehende Internetverbindung verfügen (auch als private Cluster bezeichnet), müssen Sie eine der folgenden Maßnahmen ergreifen:
     - Fügen Sie die Sicherheitsgruppe hinzu, die mit den erforderlichen VPC-Endpunkten verbunden ist. Weitere Informationen zu den erforderlichen Endpunkten finden Sie unter Schnittstellen-VPC-Endpunkte in Subnetzzugriff auf AWS-Services.
     - Ändern Sie die Sicherheitsgruppe, die Amazon EKS erstellt hat, um Datenverkehr von der Sicherheitsgruppe zuzulassen, die den VPC-Endpunkten zugeordnet ist.

Wenn Sie mit dieser Seite fertig sind, wählen Sie Weiter aus.

7. Auf der Seite Beobachtbarkeit konfigurieren können Sie optional auswählen, welche Metriken und Optionen zur Steuerebenen-Protokollierung Sie aktivieren möchten. Standardmäßig sind alle Protokollierungstypen deaktiviert.
   - Weiteren Informationen zur Prometheus-Metrik-Option finden Sie unter Aktivieren von Prometheus-Metriken beim Erstellen eines Clusters.
   - Weitere Informationen zu den Optionen für die Steuerebenen-Protokollierung finden Sie unter Amazon-EKS-Steuerebenen-Protokollierung.

   Wenn Sie mit dieser Seite fertig sind, wählen Sie Weiter aus.
8. Überprüfen Sie auf der Seite Review and create (Überprüfen und erstellen) die Informationen, die Sie auf den vorherigen Seiten eingegeben oder ausgewählt haben. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie zufrieden sind, klicken Sie auf Create app (Anwendung erstellen). Das Feld Status zeigt CREATING (WIRD ERSTELLT) an, während der Cluster bereitgestellt wird.

   Die Clusterbereitstellung dauert mehrere Minuten.

2. Nachdem Ihr Cluster erstellt wurde, können Sie die erstellten Instances der Amazon-EC2-Steuerebene anzeigen.

```
aws ec2 describe-instances --query 'Reservations[*].Instances[*].{Name:Tags[?
Key==`Name`]|[0].Value}' | grep my-cluster-control-plane
```

Eine Beispielausgabe sieht wie folgt aus.

```
"Name": "my-cluster-control-plane-id1"
"Name": "my-cluster-control-plane-id2"
"Name": "my-cluster-control-plane-id3"
```

Jede Instance ist mit `node-role.eks-local.amazonaws.com/control-plane` gekennzeichnet, sodass auf den Instances der Steuerebene keine Workloads geplant werden. Weitere Informationen zu Taints finden Sie unter Taints und Toleranzen in der Kubernetes-Dokumentation. Amazon EKS überwacht kontinuierlich den Status lokaler Cluster. Wir führen automatische Verwaltungsaktionen durch, z. B. Sicherheits-Patches und das Reparieren fehlerhafter Instances. Wenn lokale Cluster von der Cloud getrennt werden, führen wir Aktionen

durch, um sicherzustellen, dass der Cluster bei der erneuten Verbindung wieder in einen fehlerfreien Zustand versetzt wird.

3. Wenn Sie Ihren Cluster mit `eksctl` erstellt haben, können Sie diesen Schritt überspringen. `eksctl` schließt diesen Schritt für Sie ab. Aktivieren Sie `kubectl`, um mit Ihrem Cluster zu kommunizieren, indem Sie einen neuen Kontext zur Datei `kubectl config` hinzufügen. Anweisungen zum Erstellen und Aktualisieren der Datei finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.

   ```
   aws eks update-kubeconfig --region region-code --name my-cluster
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   Added new context arn:aws:eks:region-code:111122223333:cluster/my-cluster to /home/
   username/.kube/config
   ```

4. Um eine Verbindung zum Kubernetes-API-Server Ihres lokalen Clusters herzustellen, müssen Sie über Zugriff auf das lokale Gateway für das Subnetz verfügen oder eine Verbindung innerhalb der VPC herstellen. Weitere Informationen zum Verbinden eines Outpost-Racks mit Ihrem lokalen Netzwerk finden Sie im Benutzerhandbuch unter So funktionieren lokale Gateways für Racks. AWS Outposts Wenn Sie direktes VPC-Routing verwenden und das Outpost-Subnetz eine Route zu Ihrem lokalen Gateway hat, werden die privaten IP-Adressen der Kubernetes-Instances der Steuerebene automatisch über Ihr lokales Netzwerk übertragen. Der Kubernetes-API-Server-Endpunkt des lokalen Clusters wird in Amazon Route 53 (Route 53) gehostet. Der API-Service-Endpunkt kann von öffentlichen DNS-Servern in die privaten IP-Adressen der Kubernetes API-Server aufgelöst werden.

   Die Instances der Kubernetes-Steuerebene von lokalen Clustern werden mit statischen elastischen Netzwerkoberflächen mit festen privaten IP-Adressen konfiguriert, die sich während des gesamten Cluster-Lebenszyklus nicht ändern. Rechner, die mit dem Kubernetes-API-Server interagieren, verfügen möglicherweise nicht über eine Verbindung zu Route 53, wenn die Netzwerkverbindung unterbrochen ist. In diesem Fall empfehlen wir, `/etc/hosts` mit den statischen privaten IP-Adressen für den weiteren Betrieb zu konfigurieren. Wir empfehlen außerdem, lokale DNS-Server einzurichten und diese mit Ihrem Outpost zu verbinden. Weitere Informationen finden Sie in der AWS Outposts -Dokumentation. Führen Sie den folgenden Befehl aus, um zu bestätigen, dass die Kommunikation mit Ihrem Cluster hergestellt wurde.

   ```
   kubectl get svc
   ```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   28h
```

5. (Optional) Testen Sie die Authentifizierung bei Ihrem lokalen Cluster, wenn er sich in einem vom AWS Cloud getrennten Status befindet. Anweisungen finden Sie unter Vorbereitungen für Netzwerkunterbrechungen.

Interne Ressourcen

Amazon EKS erstellt die folgenden Ressourcen in Ihrem Cluster. Die Ressourcen sind für den internen Gebrauch von Amazon EKS bestimmt. Damit Ihr Cluster ordnungsgemäß funktioniert, sollten Sie diese Ressourcen nicht bearbeiten oder ändern.

- Die folgenden Spiegel-Pods:
  - `aws-iam-authenticator-`*`node-hostname`*
  - `eks-certificates-controller-`*`node-hostname`*
  - `etcd-`*`node-hostname`*
  - `kube-apiserver-`*`node-hostname`*
  - `kube-controller-manager-`*`node-hostname`*
  - `kube-scheduler-`*`node-hostname`*
- Die folgenden selbstverwalteten Add-Ons:
  - `kube-system/coredns`
  - `kube-system/kube-proxy` (wird erst erstellt, wenn Sie Ihren ersten Knoten hinzugefügt haben)
  - `kube-system/aws-node` (wird erst erstellt, wenn Sie Ihren ersten Knoten hinzugefügt haben). Lokale Cluster verwenden das Amazon VPC CNI plugin for Kubernetes-Plugin für Cluster-Netzwerke. Ändern Sie nicht die Konfiguration für Steuerebene-Instances (Pods mit dem Namen `aws-node-controlplane-*`). Es gibt Konfigurationsvariablen, die Sie verwenden können, um den Standardwert zu ändern, wenn das Plugin neue Netzwerkschnittstellen erstellt. Weitere Informationen finden Sie in der Dokumentation zu. GitHub
- Die folgenden Services:
  - `default/kubernetes`

  - `kube-system/kube-dns`
- Eine `PodSecurityPolicy` namens `eks.system`
- Eine `ClusterRole` namens `eks:system:podsecuritypolicy`
- Eine `ClusterRoleBinding` namens `eks:system`
- Eine PodSecurityStandardrichtlinie
- Zusätzlich zur Cluster-Sicherheitsgruppe erstellt Amazon EKS in Ihrer Gruppe eine Sicherheitsgruppe mit AWS-Konto dem Name`eks-local-internal-do-not-use-or-edit-`*`cluster-name`*`-`*`uniqueid`*. Diese Sicherheitsgruppe ermöglicht den freien Datenverkehr zwischen Kubernetes-Komponenten, die auf Instances der Steuerebene ausgeführt werden.

Empfohlene nächste Schritte:

- Erteilen Sie dem IAM-Prinzipal, der den Cluster erstellt hat, die erforderlichen Berechtigungen zum Anzeigen von Kubernetes Ressourcen in AWS Management Console
- Gewähren Sie IAM-Entitäten Zugriff auf Ihren Cluster. Wenn Sie möchten, dass die Entitäten Kubernetes-Ressourcen in der Amazon-EKS-Konsole anzeigen sollen, gewähren Sie den Entitäten Erforderliche Berechtigungen.
- Konfigurieren Sie die Protokollierung für Ihren Cluster
- Machen Sie sich mit dem vertraut, was während Netzwerktrennungen passiert.
- Fügen Sie Knoten zu Ihrem Cluster hinzu
- Erwägen Sie, einen Backup-Plan für Ihr `etcd` zu erstellen. Amazon EKS unterstützt keine automatisierte Sicherung und Wiederherstellung von `etcd` für lokale Cluster. Weitere Informationen finden Sie unter Sichern eines `etcd`-Cluster in der Kubernetes-Dokumentation. Die beiden Hauptoptionen sind die Verwendung von `etcdctl` zur Automatisierung der Erstellung von Snapshots oder die Verwendung der Amazon-EBS-Speicher-Volume-Sicherung.

Versionen der lokalen Amazon EKS-Clusterplattform

Lokale Cluster-Plattformversionen stellen die Fähigkeiten des Amazon-EKS-Clusters auf AWS Outposts dar. Die Versionen enthalten die Komponenten, die auf der Kubernetes-Steuerebene ausgeführt werden, wobei die Kubernetes-API-Server-Flags aktiviert sind. Diese enthalten auch die aktuelle Kubernetes-Patch-Version. Jede Kubernetes-Minor-Version hat eine oder mehrere zugehörige Plattformversionen. Die Plattformversionen für verschiedene Kubernetes-Minor-Versionen

sind voneinander unabhängig. Die Plattformversionen für lokale Cluster und Amazon-EKS-Cluster in der Cloud sind unabhängig.

Wenn eine neue Kubernetes-Minor-Version für lokale Cluster verfügbar ist (z. B. `1.28`), beginnt die erste Plattformversion für diese Kubernetes-Minor-Version bei `eks-local-outposts.1`. Allerdings veröffentlicht Amazon EKS regelmäßig neue Plattformversionen, um neue Einstellungen für die Kubernetes-Steuerebene und Sicherheitsfixes bereitzustellen.

Wenn neue lokale Cluster-Plattformversionen für eine Minor-Version verfügbar werden:

- Die Versionsnummer der -Plattform wird erhöht (`eks-local-outposts.n+1`).
- Amazon EKS aktualisiert automatisch alle vorhandenen lokalen Cluster auf die neueste Plattformversion für ihre entsprechende Kubernetes-Nebenversion. Automatische Aktualisierungen bestehender Plattformversionen werden schrittweise durchgeführt. Der Rollout-Prozess kann einige Zeit in Anspruch nehmen. Wenn Sie die Features der neuesten Plattformversion sofort benötigen, empfehlen wir Ihnen, einen neuen lokalen Cluster zu erstellen.
- Amazon EKS veröffentlicht möglicherweise ein neues Knoten-AMI mit einer entsprechenden Patch-Version. Alle Patch-Versionen sind zwischen der Kubernetes-Steuerebene und den Knoten-AMIs für eine einzelne Kubernetes-Nebenversion kompatibel.

Neue -Plattformversionen führen keine kritischen Änderungen ein. Sie führen nicht zu Service-Unterbrechungen.

Lokale Cluster werden immer mit der neuesten verfügbaren Plattformversion (`eks-local-outposts.n`) für die angegebene Kubernetes-Version erstellt.

Die aktuellen und kürzlichen -Plattformversionen sind in den folgenden Tabellen beschrieben.

Kubernetes-Version **`1.28`**

Die folgenden Zugangscontroller sind für alle `1.28` Versionen der Plattform aktiviert: `CertificateApproval`, `CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `DefaultStorageClass`, `DefaultTolerationSeconds`, `ExtendedResourceToleration`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`, `NodeRestriction`, `PersistentVolumeClaimResize`, `Priority`, `PodSecurity`, `ResourceQuota`, `RuntimeClass`, `ServiceAccount`, `StorageObjectInUseProtection`, `TaintNodesByCondition`, `ValidatingAdmissionPolicy` und `ValidatingAdmissionWebhook`.

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.28.6	eks-local-outposts.5	Die Bottlerocket-Version wurde auf v1.19.3 aktualisiert und enthält die neuesten Bugfixes zur Unterstützung des lokalen Boots in Outposts.	18. April 2024
1.28.6	eks-local-outposts.4	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen. Die Unterstützung oder der lokale Start in Outposts wurde wiederhergestellt. Aus Kompatibilitätsgründen auf Bottlerocket Version herabgestuft. v1.15.1	2. April 2024
1.28.6	eks-local-outposts.3	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	22. März 2024
1.28.6	eks-local-outposts.2	Neue Plattformversion mit Sicherheitskorrekturen und Verbesserungen, auf die Kube-Proxy aktualisiert wurde. v1.28.6 AWS IAM Authenticator wurde aktualisiert auf. v0.6.17 Das Amazon VPC CNI-Plugin für Kubernetes wurde aus Kompatibilitätsgründen auf herabgestuft. v1.13.2 Bottlerocketv1.19.2Aktualisierte Version auf.	8. März 2024

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.28.1	eks-local-outposts.1	Erste Veröffentlichung der Kubernetes-Versionv1.28. für lokale Amazon EKS-Cluster auf Outpost	4. Oktober 2023

Kubernetes-Version **1.27**

Die folgenden Zugangscontroller sind für alle 1.27 Versionen der Plattform aktiviert: CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, ExtendedResourceToleration, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, PersistentVolumeClaimResize, Priority, PodSecurity, ResourceQuota, RuntimeClass, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition, ValidatingAdmissionPolicy und ValidatingAdmissionWebhook.

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.27.10	eks-local-outposts.5	Neue Plattform mit Sicherheitskorrekturen und Verbesserungen.	2. April 2024
1.27.10	eks-local-outposts.4	Neue Plattform mit Sicherheitskorrekturen und Verbesserungen. Kube-Proxy wurde aktualisiert auf. v1.27.10 AWS IAM Authenticator wurde aktualisiert auf. v0.6.17 Aktualisierte Bottlerocket Version auf. v1.19.2	22. März 2024

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.27.3	eks-local-outposts.3	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen. kube-proxy aktualisiert zu v1.27.3. Amazon-VPC-CNI-Plugin für Kubernetes aktualisiert zu v1.13.2.	14. Juli 2023
1.27.1	eks-local-outposts.2	Das CoreDNS-Image wurde aktualisiert auf v1.10.1.	22. Juni 2023
1.27.1	eks-local-outposts.1	Erste Veröffentlichung der Kubernetes-Version 1.27 für lokale Amazon EKS-Cluster auf Outposts.	30. Mai 2023

Kubernetes-Version **1.26**

Die folgenden Zugangscontroller sind für alle 1.26 Versionen der Plattform aktiviert: CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, ExtendedResourceToleration, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, PersistentVolumeClaimResize, Priority, PodSecurity, ResourceQuota, RuntimeClass, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition, ValidatingAdmissionPolicy und ValidatingAdmissionWebhook.

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.26.13	eks-local-outposts.5	Neue Plattformversion mit Sicherheitskorrekturen und Verbesserungen. Kube-	22. März 2024

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
		Proxy wurde aktualisiert auf. `v1.26.13` AWS IAM Authenticator wurde aktualisiert auf. `v0.6.17` Aktualisierte Bottlerocket Version auf. `v1.19.2`	

Kubernetes-Version **`1.25`**

Die folgenden Zugangscontroller sind für alle Versionen der Plattform `1.25` aktiviert: `CertificateApproval`, `CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `DefaultStorageClass`, `DefaultTolerationSeconds`, `ExtendedResourceToleration`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`, `NodeRestriction`, `PersistentVolumeClaimResize`, `Priority`, `PodSecurity`, `ResourceQuota`, `RuntimeClass`, `ServiceAccount`, `StorageObjectInUseProtection`, `TaintNodesByCondition` und `ValidatingAdmissionWebhook`.

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
`1.25.16`	`eks-local-outposts.7`	Neue Plattformversion mit Sicherheitskorrekturen und Verbesserungen. Kube-Proxy wurde aktualisiert auf. `v1.25.16` AWS IAM Authenticator wurde aktualisiert auf. `v0.6.17` Aktualisierte Bottlerocket Version auf. `v1.19.2`	22. März 2024
`1.25.11`	`eks-local-outposts.6`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen. `kube-proxy` aktualisi	14. Juli 2023

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
		ert zu v1.25.11. Amazon-VPC-CNI-Plugin für Kubernetes aktualisiert zu v1.13.2.	
1.25.9	eks-local-outposts.5	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	13. Juli 2023
1.25.6	eks-local-outposts.4	Bottlerocket-Version aktualisiert auf 1.13.2	2. Mai 2023
1.25.6	eks-local-outposts.3	Das Betriebssystem der Amazon EKS Control Plane Instance wurde auf die Bottlerocket-Version v1.13.1 und das Amazon VPC CNI-Plugin für Kubernetes auf Version aktualisiert. v1.12.6	14. April 2023
1.25.6	eks-local-outposts.2	Die Erfassung von Diagnosen für Instanzen der Kubernetes-Steuerebene wurde verbessert.	08. März 2023
1.25.6	eks-local-outposts.1	Erste Veröffentlichung der Kubernetes-Version 1.25 für lokale Amazon EKS-Cluster auf Outposts.	1. März 2023

Kubernetes-Version **1.24**

Die folgenden Zugangscontroller sind für alle Versionen der Plattform 1.24 aktiviert: DefaultStorageClass, DefaultTolerationSeconds, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, ResourceQuota, ServiceAccount, ValidatingAdmissionWebhook,

PodSecurityPolicy, TaintNodesByCondition, StorageObjectInUseProtection, PersistentVolumeClaimResize, ExtendedResourceToleration, CertificateApproval, PodPriority, CertificateSigning, CertificateSubjectRestriction, RuntimeClass und DefaultIngressClass.

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.24.17	eks-local-outposts.7	Neue Plattformversion mit Sicherheitskorrekturen und Verbesserungen. Kube-Proxy wurde aktualisiert auf. v1.25.16 AWS IAM Authenticator aktualisiert. v0.6.17 Aktualisierte Bottlerocket Version auf. v1.19.2	22. März 2024
1.24.15	eks-local-outposts.6	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen. kube-proxy aktualisiert zu v1.24.15. Amazon-VPC-CNI-Plugin für Kubernetes aktualisiert zu v1.13.2.	14. Juli 2023
1.24.13	eks-local-outposts.5	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	13. Juli 2023
1.24.9	eks-local-outposts.4	Bottlerocket-Version aktualisiert auf 1.13.2	2. Mai 2023
1.24.9	eks-local-outposts.3	Das Betriebssystem der Amazon EKS Control Plane Instance wurde auf die Bottlerocket-Version v1.13.1 und das Amazon VPC CNI-	14. April 2023

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
		Plugin für Kubernetes auf Version aktualisiert. `v1.12.6`	
`1.24.9`	`eks-local-outposts.2`	Die Erfassung von Diagnosen für Instanzen der Kubernetes-Steuerebene wurde verbessert.	08. März 2023
`1.24.9`	`eks-local-outposts.1`	Erste Veröffentlichung der Kubernetes-Version `1.24` für lokale Amazon EKS-Cluster auf Outposts.	17. Januar 2023

Kubernetes-Version **`1.23`**

Die folgenden Zugangscontroller sind für alle Versionen der Plattform `1.23` aktiviert: `DefaultStorageClass`, `DefaultTolerationSeconds`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`, `NodeRestriction`, `ResourceQuota`, `ServiceAccount`, `ValidatingAdmissionWebhook`, `PodSecurityPolicy`, `TaintNodesByCondition`, `StorageObjectInUseProtection`, `PersistentVolumeClaimResize`, `ExtendedResourceToleration`, `CertificateApproval`, `PodPriority`, `CertificateSigning`, `CertificateSubjectRestriction`, `RuntimeClass` und `DefaultIngressClass`.

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
`1.23.17`	`eks-local-outposts.6`	Neue Plattformversion mit Sicherheitsfixes und -verbesserungen.	13. Juli 2023

Kubernetes-Version	Amazon-EKS-Plattformversion	Versionshinweise	Datum der Veröffentlichung
1.23.17	eks-local-outposts.5	Neue Plattformversion mit Sicherheitskorrekturen und Verbesserungen. Kube-Proxy wurde aktualisiert auf. v1.23.17 Aktualisierte Bottlerocket Version auf. v1.14.1	6. Juli 2023
1.23.15	eks-local-outposts.4	Das Betriebssystem der Amazon EKS Control Plane Instance wurde auf die Bottlerocket-Version v1.13.1 und das Amazon VPC CNI-Plugin für Kubernetes auf Version aktualisiert. v1.12.6	14. April 2023
1.23.15	eks-local-outposts.3	Die Erfassung von Diagnosen für Instanzen der Kubernetes-Steuerebene wurde verbessert.	08. März 2023
1.23.15	eks-local-outposts.2	Erste Veröffentlichung der Kubernetes-Version 1.23 für lokale Amazon EKS-Cluster auf Outposts.	17. Januar 2023

Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen

Wenn Sie einen lokalen Cluster erstellen, geben Sie eine VPC und mindestens ein privates Subnetz an, das auf Outposts ausgeführt wird. Dieses Thema bietet einen Überblick über die VPC- und Subnetzanforderungen und -überlegungen für Ihren lokalen Cluster.

VPC-Anforderungen und -Überlegungen

Wenn Sie einen lokalen Cluster erstellen, muss die von Ihnen angegebene VPC die folgenden Anforderungen und Überlegungen erfüllen:

- Stellen Sie sicher, dass die VPC über genügend IP-Adressen für den lokalen Cluster, alle Knoten und andere Kubernetes-Ressourcen verfügt, die Sie erstellen möchten. Wenn die VPC, die Sie verwenden möchten, nicht über genügend IP-Adressen verfügt, erhöhen Sie die Anzahl der verfügbaren IP-Adressen. Das ist möglich, indem Sie zusätzliche CIDR-Blöcke (Classless Inter-Domain Routing) mit Ihrer VPC verbinden. Sie können entweder vor oder nach der Erstellung Ihres Clusters private (RFC 1918) und öffentliche (nicht RFC 1918) CIDR-Blöcke mit Ihrer VPC verbinden. Es kann bis zu 5 Stunden dauern, bis ein CIDR-Block, den Sie einem VPC zugeordnet haben, von einem Cluster erkannt wird.
- Der VPC dürfen keine IP-Präfixe oder IPv6-CIDR-Blöcke zugewiesen werden. Aufgrund dieser Einschränkungen gelten die in Erhöhen Sie die Anzahl der verfügbaren IP-Adressen für Ihre Amazon-EC2-Knoten und `IPv6`Adressen für ClusterPods, und services enthaltenen Informationen nicht für Ihre VPC.
- Die VPC verfügt über einen DNS-Hostnamen und die DNS-Auflösung ist aktiviert. Ohne diese Funktionen kann der lokale Cluster nicht erstellt werden, und Sie müssen die Funktionen aktivieren und Ihren Cluster neu erstellen. Weitere Informationen finden Sie unter DNS-Attribute für Ihre VPC im Amazon-VPC-Benutzerhandbuch.
- Um über Ihr lokales Netzwerk auf Ihren lokalen Cluster zuzugreifen, muss die VPC der lokalen Gateway-Routing-Tabelle Ihres Outpost zugeordnet sein. Weitere Informationen finden Sie unter VPC-Zuordnungen im AWS Outposts-Benutzerhandbuch.

Subnetz-Anforderungen und -Überlegungen

Geben Sie beim Erstellen des Clusters mindestens ein privates Subnetz an. Wenn Sie mehr als ein Subnetz angeben, werden die Kubernetes-Steuerebene-Instances gleichmäßig auf die Subnetze verteilt. Wenn mehr als ein Subnetz angegeben wird, müssen die Subnetze auf demselben Outpost vorhanden sein. Darüber hinaus müssen die Subnetze auch über die richtigen Routen und Sicherheitsgruppen-Berechtigungen verfügen, um miteinander kommunizieren zu können. Wenn Sie einen lokalen Cluster erstellen, müssen die von Ihnen angegebenen Subnetze die folgenden Anforderungen erfüllen:

- Die Subnetze befinden sich alle auf demselben logischen Outpost.

- Die Subnetze verfügen zusammen über mindestens drei verfügbare IP-Adressen für die Kubernetes-Instances der Steuerebene. Wenn drei Subnetze angegeben werden, muss jedes Subnetz über mindestens eine verfügbare IP-Adresse verfügen. Wenn zwei Subnetze angegeben werden, muss jedes Subnetz über mindestens zwei verfügbare IP-Adressen verfügen. Wenn ein Subnetz angegeben wird, muss das Subnetz über mindestens drei verfügbare IP-Adressen verfügen.
- Die Subnetze verfügen über eine Route zum lokalen Gateway des Outpost-Racks, um über Ihr lokales Netzwerk auf den Kubernetes-API-Server zuzugreifen. Wenn die Subnetze über keine Route zum lokalen Gateway des Outpost-Racks verfügen, müssen Sie mit Ihrem Kubernetes-API-Server von der VPC aus kommunizieren.
- Die Subnetze müssen eine IP-Adressen-basierte Benennung aufweisen. Die ressourcenbasierte Benennung von Amazon EC2 wird nicht von Amazon EKS unterstützt.

Subnetzzugriff auf AWS-Services

Die privaten Subnetze des lokalen Clusters in Outposts müssen mit regionalen AWS-Services-Services kommunizieren können. Dazu können Sie ein NAT-Gateway für den ausgehenden Internetzugang verwenden oder, wenn Sie den gesamten Datenverkehr innerhalb Ihrer VPC privat halten möchten, Schnittstellen-VPC-Endpunkte verwenden.

Verwenden eines NAT-Gateways

Die privaten Subnetze des lokalen Clusters in Outposts müssen über eine zugeordnete Routing-Tabelle mit einer Route zu einem NAT-Gateway verfügen, das sich in einem öffentlichen Subnetz in der übergeordneten Availability Zone des Outposts befindet. Das öffentliche Subnetz muss über eine Route zu einem Internet-Gateway verfügen. Das NAT-Gateway ermöglicht einen ausgehenden Zugriff auf das Internet und verhindert unerwünschte eingehende Verbindungen aus dem Internet zu Instances im Outpost.

Verwendung von -Schnittstellen-VPC-Endpunkten

Wenn die privaten Subnetze des lokalen Clusters in Outposts keine ausgehende Internetverbindung haben oder wenn Sie den gesamten Datenverkehr innerhalb Ihrer VPC privat halten möchten, müssen Sie die folgenden Schnittstellen-VPC-Endpunkte und den Gateway-Endpunkt in einem regionalen Subnetz erstellen, bevor Sie Ihren Cluster erstellen.

Endpunkt	Endpunkttyp
com.amazonaws. *region-code* .ssm	Schnittstelle
com.amazonaws. *region-code* .ssmmessages	Schnittstelle
com.amazonaws. *region-code* .ec2messages	Schnittstelle
com.amazonaws. *region-code* .ec2	Schnittstelle
com.amazonaws. *region-code* .secretsmanager	Schnittstelle
com.amazonaws. *region-code* .logs	Schnittstelle
com.amazonaws. *region-code* .sts	Schnittstelle
com.amazonaws. *region-code* .ecr.api	Schnittstelle
com.amazonaws. *region-code* .ecr.dkr	Schnittstelle
com.amazonaws. *region-code* .s3	Gateway

Die Endpunkte müssen die folgenden Anforderungen erfüllen:

- Sie müssen in einem privaten Subnetz erstellt werden, das sich in der übergeordneten Availability Zone Ihres Outposts befindet.
- Private DNS-Namen müssen aktiviert sein.
- Sie müssen über eine angefügte Sicherheitsgruppe verfügen, die eingehenden HTTPS-Datenverkehr aus dem CIDR-Bereich des privaten Outpost-Subnetzes zulässt.

Für die Erstellung von Endpunkten fallen Gebühren an. Weitere Informationen finden Sie unter AWS PrivateLink Preise. Wenn Ihre Pods Zugriff auf andere AWS-Services benötigen, müssen Sie zusätzliche Endpunkte erstellen. Eine umfassende Liste der Endpunkte finden Sie unter AWS-Services, die in AWS PrivateLink integriert werden.

Erstellen einer VPC

Sie können eine VPC erstellen, die die vorherigen Anforderungen erfüllt, indem Sie eine der folgenden AWS CloudFormation-Vorlagen verwenden:

- Vorlage 1 – Diese Vorlage erstellt eine VPC mit einem privaten Subnetz im Outpost und einem öffentlichen Subnetz im AWS-Region. Das private Subnetz verfügt über eine Route zum Internet über ein NAT-Gateway, das sich im öffentlichen Subnetz im AWS-Region befindet. Diese Vorlage kann verwendet werden, um einen lokalen Cluster in einem Subnetz mit ausgehendem Internetzugriff zu erstellen.
- Vorlage 2 – Diese Vorlage erstellt eine VPC mit einem privaten Subnetz auf dem Outpost und der Mindestanzahl von VPC-Endpunkten, die erforderlich sind, um einen lokalen Cluster in einem Subnetz zu erstellen, das über keinen eingehenden oder ausgehenden Internetzugriff verfügt (auch als privates Subnetz bezeichnet).

Vorbereitungen für Netzwerkunterbrechungen

Wenn Ihr lokales Netzwerk die Verbindung zum AWS Cloud verloren hat, können Sie Ihren lokalen Amazon-EKS-Cluster weiterhin auf einem Outpost verwenden. In diesem Thema wird beschrieben, wie Sie Ihren lokalen Cluster auf Netzwerkunterbrechungen vorbereiten können, und verwandte Überlegungen.

Überlegungen zur Vorbereitung Ihres lokalen Clusters auf eine Netzwerkunterbrechung:

- Lokale Cluster ermöglichen Stabilität und kontinuierlichen Betrieb bei vorübergehenden, ungeplanten Netzwerkunterbrechungen. AWS Outposts bleibt ein vollständig vernetztes Angebot, das als Erweiterung des AWS Cloud in Ihrem Rechenzentrum fungiert. Bei Netzwerkunterbrechungen zwischen Ihrem Outpost und AWS Cloud, empfehlen wir, zu versuchen, Ihre Verbindung wiederherzustellen. Eine Anleitung dazu finden Sie in der Checkliste zur Fehlerbehebung im AWS Outposts-Rack-Netzwerk im AWS Outposts-Benutzerhandbuch. Weitere Informationen zum Beheben von Problemen mit lokalen Clustern finden Sie unter Problembehandlung bei lokalen Clustern für Amazon EKS auf AWS Outposts.
- Outposts geben eine `ConnectedStatus`-Metrik aus, mit der Sie den Konnektivitätsstatus Ihres Outposts überwachen können. Weitere Informationen finden Sie unter Outposts-Metriken im AWS Outposts-Benutzerhandbuch.
- Lokale Cluster verwenden IAM als standardmäßigen Authentifizierungsmechanismus mit dem AWS Identity and Access Management-Authentifikator für Kubernetes. IAM ist während

Netzwerkunterbrechungen nicht verfügbar. Daher unterstützen lokale Cluster einen alternativen Authentifizierungsmechanismus mithilfe von `x.509`-Zertifikaten, die Sie verwenden können, um eine Verbindung zu Ihrem Cluster herzustellen, wenn die Netzwerkverbindung unterbrochen ist. Informationen zum Abrufen und Verwenden eines `x.509`-Zertifikats für Ihren Cluster finden Sie unter Authentifizierung bei Ihrem lokalen Cluster während einer Netzwerkunterbrechung.

- Wenn Sie bei Netzwerkunterbrechungen nicht auf Route 53 zugreifen können, sollten Sie lokale DNS-Server in Ihrer On-Premises-Umgebung verwenden. Die Kubernetes-Instances der Steuerebene verwenden statische IP-Adressen. Sie können die Hosts, die Sie für die Verbindung mit Ihrem Cluster verwenden, mit dem Hostnamen und den IP-Adressen des Endpunkts als Alternative zur Verwendung lokaler DNS-Server konfigurieren. Weitere Informationen finden Sie unter DNS im AWS Outposts-Benutzerhandbuch.
- Wenn Sie bei Netzwerkunterbrechungen mit einem Anstieg des Anwendungsdatenverkehrs rechnen, können Sie in Ihrem Cluster bei Verbindung mit der Cloud freie Rechenkapazität bereitstellen. Amazon-EC2-Instances sind im Preis von AWS Outposts enthalten. Der Betrieb von Ersatz-Instances hat also keinen Einfluss auf Ihre AWS-Nutzungskosten.
- Während Netzwerkunterbrechungen, die das Erstellen, Aktualisieren und Skalieren von Workloads ermöglichen, müssen die Container-Images Ihrer Anwendung über das lokale Netzwerk zugänglich sein und Ihr Cluster muss über genügend Kapazität verfügen. Lokale Cluster hosten keine Container-Registrierung für Sie. Wenn die Pods zuvor auf diesen Knoten ausgeführt wurden, werden Container-Images auf den Knoten zwischengespeichert. Wenn Sie die Container-Images Ihrer Anwendung normalerweise aus Amazon ECR in die Cloud ziehen, sollten Sie erwägen, einen lokalen Cache oder eine Registrierung auszuführen. Ein lokaler Cache oder eine lokale Registry ist hilfreich, wenn Sie während einer Netzwerkunterbrechung Workload-Ressourcen erstellen, aktualisieren und skalieren müssen.
- Lokale Cluster verwenden Amazon EBS als Standardspeicherklasse für persistente Volumes und den Amazon-EBS-CSI-Treiber, um den Lebenszyklus persistenter Amazon-EBS-Volumes zu verwalten. Während Netzwerkunterbrechungen können Pods, die von Amazon EBS gesichert werden, nicht erstellt, aktualisiert oder skaliert werden. Dies liegt daran, dass diese Vorgänge Aufrufe an die Amazon-EBS-API in der Cloud erfordern. Wenn Sie statusbehaftete Workloads auf lokalen Clustern bereitstellen und während Netzwerktrennungen Vorgänge zum Erstellen, Aktualisieren oder Skalieren benötigen, sollten Sie die Verwendung eines alternativen Speichermechanismus in Betracht ziehen.
- Amazon EBS-Snapshots können nicht erstellt oder gelöscht werden, wenn AWS Outposts kein Zugriff auf die relevanten AWS-APIs in der Region (z. B. die APIs für Amazon EBS oder Amazon S3) möglich ist.

- Bei der Integration von ALB (Ingress) mit AWS Certificate Manager (ACM) werden Zertifikate abgerufen und im Speicher der AWS Outposts-ALB-Compute-Instance gespeichert. Die aktuelle TLS-Terminierung wird im Falle einer Trennung von AWS-Region weiter betrieben. Mutationsvorgänge in diesem Kontext schlagen fehl (z. B. neue Eingangsdefinitionen, neue API-Vorgänge für ACM-basierte Zertifikate, ALB-Rechenskalierung oder Zertifikatsrotation). Weitere Informationen finden Sie unter Problembehandlung bei der Erneuerung verwalteter Zertifikate im AWS Certificate Manager-Benutzerhandbuch.
- Die Amazon-EKS-Steuerebenen-Protokolle werden während Netzwerkunterbrechungen lokal auf Kubernetes-Instances der Steuerebene zwischengespeichert. Nach der erneuten Verbindung werden die Protokolle an CloudWatch Protokolle im übergeordneten gesendetAWS-Region. Sie können Prometheus, Grafana oder Amazon-EKS-Partnerlösungen verwenden, um den Cluster lokal mithilfe des Metrik-Endpunkts des Kubernetes-API-Servers oder mithilfe von Fluent Bit für Protokolle zu überwachen.
- Wenn Sie den AWS Load Balancer Controller auf Outposts für den Anwendungsdatenverkehr verwenden, erhalten vorhandene Pods, denen das AWS Load Balancer Controller vorangestellt ist, während der Netzwerkunterbrechung weiterhin Datenverkehr. Neue Pods, die bei Netzwerkunterbrechungen erstellt wurden, empfangen keinen Datenverkehr, bis der Outpost wieder mit der AWS Cloud verbunden ist. Erwägen Sie, die Replikatanzahl für Ihre Anwendungen festzulegen, während Sie mit der AWS Cloud verbunden sind, um Ihre Skalierungsanforderungen während Netzwerktrennungen zu erfüllen.
- Das Amazon VPC CNI plugin for Kubernetes ist standardmäßig auf den sekundären IP-Modus eingestellt. Es ist mit `WARM_ENI_TARGET = 1` konfiguriert, wodurch das Plugin „eine vollständige elastische Netzwerkschnittstelle“ mit verfügbaren IP-Adressen bereithalten kann. Erwägen Sie, `WARM_ENI_TARGET`-, `WARM_IP_TARGET`- und `MINIMUM_IP_TARGET`-Werte entsprechend Ihren Skalierungsanforderungen während eines getrennten Zustands zu ändern. Weitere Informationen finden Sie in der `readme` Datei für das Plugin auf GitHub. Eine Liste der maximalen Anzahl von Pods, die von jedem Instance-Typ unterstützt werden, finden Sie in der `eni-max-pods.txt` Datei auf GitHub.

Authentifizierung bei Ihrem lokalen Cluster während einer Netzwerkunterbrechung

AWS Identity and Access Management (IAM) ist während Netzwerkunterbrechungen nicht verfügbar. Sie können sich nicht bei Ihrem lokalen Cluster mit IAM-Anmeldeinformationen authentifizieren, während keine Verbindung besteht. Sie können jedoch über Ihr lokales Netzwerk mithilfe von `x509`-Zertifikaten eine Verbindung zu Ihrem Cluster herstellen, wenn die Verbindung getrennt ist. Sie müssen ein Client `X509`-Zertifikat herunterladen und speichern, das Sie während der

Verbindungstrennung verwenden können. In diesem Thema erfahren Sie, wie Sie das Zertifikat erstellen und verwenden, um sich bei Ihrem Cluster zu authentifizieren, wenn dieser sich in einem nicht verbundenen Status befindet.

1. Erstellen einer Zertifikatssignierungsanforderung

   a. Generieren einer Zertifikatssignierungsanforderung.

   ```
   openssl req -new -newkey rsa:4096 -nodes -days 365 \
       -keyout admin.key -out admin.csr -subj "/CN=admin"
   ```

   b. Erstellen einer Zertifikatssignierungsanforderung in Kubernetes.

   ```
   BASE64_CSR=$(cat admin.csr | base64 -w 0)
   cat << EOF > admin-csr.yaml
   apiVersion: certificates.k8s.io/v1
   kind: CertificateSigningRequest
   metadata:
     name: admin-csr
   spec:
     signerName: kubernetes.io/kube-apiserver-client
     request: ${BASE64_CSR}
     usages:
     - client auth
   EOF
   ```

2. Erstellen einer Zertifikatssignierungsanforderung mit `kubectl`.

   ```
   kubectl create -f admin-csr.yaml
   ```

3. Überprüfen Sie den Status der Zertifikatssignierungsanforderung.

   ```
   kubectl get csr admin-csr
   ```

   Eine Beispielausgabe sieht wie folgt aus.

   ```
   NAME       AGE   REQUESTOR                      CONDITION
   admin-csr  11m   kubernetes-admin               Pending
   ```

   Kubernetes hat die Anfrage zum Signieren des Zertifikats erstellt.

4. Genehmigen Sie die Zertifikatssignieranforderung.

```
kubectl certificate approve admin-csr
```

5. Überprüfen Sie erneut den Status der Anfrage zum Signieren des Zertifikats auf Genehmigung.

```
kubectl get csr admin-csr
```

Eine Beispielausgabe sieht wie folgt aus.

```
NAME       AGE   REQUESTOR                     CONDITION
admin-csr  11m   kubernetes-admin              Approved
```

6. Rufen Sie das Zertifikat ab und überprüfen Sie es.

a. Rufen Sie das Zertifikat ab.

```
kubectl get csr admin-csr -o jsonpath='{.status.certificate}' | base64 --decode
 > admin.crt
```

b. Überprüfen Sie das Zertifikat.

```
cat admin.crt
```

7. Erstellen Sie eine Cluster-Rollenbindung für einen `admin`-Benutzer.

```
kubectl create clusterrolebinding admin --clusterrole=cluster-admin \
    --user=admin --group=system:masters
```

8. Generieren Sie eine kubeconfig mit Benutzerbereich für einen getrennten Status.

Sie können eine `kubeconfig`-Datei mit den heruntergeladenen `admin`-Zertifikaten generieren. Ersetzen Sie *`my-cluster`* und *`apiserver-endpoint`* in den folgenden Befehlen.

```
aws eks describe-cluster --name my-cluster \
    --query "cluster.certificateAuthority" \
    --output text | base64 --decode > ca.crt
```

```
kubectl config --kubeconfig admin.kubeconfig set-cluster my-cluster \
    --certificate-authority=ca.crt --server apiserver-endpoint --embed-certs
```

```
kubectl config --kubeconfig admin.kubeconfig set-credentials admin \
    --client-certificate=admin.crt --client-key=admin.key --embed-certs
```

```
kubectl config --kubeconfig admin.kubeconfig set-context admin@my-cluster \
    --cluster my-cluster --user admin
```

```
kubectl config --kubeconfig admin.kubeconfig use-context admin@my-cluster
```

9. Ziegen Sie Ihre `kubeconfig`-Datei an.

```
kubectl get nodes --kubeconfig admin.kubeconfig
```

10. Wenn Sie bereits über Services in Produktion auf Ihrem Outpost verfügen, überspringen Sie diesen Schritt. Wenn Amazon EKS der einzige Service ist, der auf Ihrem Outpost ausgeführt wird, und der Outpost sich nicht in der Produktion befindet, können Sie eine Netzwerkunterbrechung simulieren. Bevor Sie mit Ihrem lokalen Cluster in Produktion gehen, simulieren Sie einen Verbindungsabbruch, um sicherzustellen, dass Sie auf Ihren Cluster zugreifen können, wenn er sich in einem getrennten Status befindet.

    a. Wenden Sie Firewall-Regeln auf die Netzwerkgeräte an, die Ihren Outpost mit dem AWS-Region verbinden. Dadurch wird der Service-Link des Outposts getrennt. Sie können keine neuen Instances erstellen. Aktuell ausgeführte Instances verlieren die Konnektivität zur AWS-Region und zum Internet.

    b. Sie können die Verbindung zu Ihrem lokalen Cluster testen, während die Verbindung getrennt ist, indem Sie das `x509`-Zertifikat verwenden. Stellen Sie sicher, dass Sie Ihr `kubeconfig` in das `admin.kubeconfig` ändern, das Sie in einem vorherigen Schritt erstellt haben. Ersetzen Sie *`my-cluster`* durch den Namen Ihres lokalen Clusters.

        ```
        kubectl config use-context admin@my-cluster --kubeconfig admin.kubeconfig
        ```

    Wenn Sie Probleme mit Ihren lokalen Clustern feststellen, während diese sich im getrennten Status befinden, empfehlen wir Ihnen, ein Support-Ticket zu eröffnen.

Überlegungen zur Kapazität

Dieses Thema enthält Anleitungen zur Auswahl des Instance-Typs der Kubernetes-Steuerebene und (optional) zur Verwendung von Platzierungsgruppen, um Hochverfügbarkeitsanforderungen für Ihren lokalen Amazon-EKS-Cluster auf einem Outpost zu erfüllen.

Bevor Sie einen Instance-Typ (z. B. `m5`, `c5`, oder `r5`)) zur Verwendung für die Kubernetes-Steuerebene Ihres lokalen Clusters in Outposts auswählen, bestätigen Sie die Instance-Typen, die in Ihrer Outpost-Konfiguration verfügbar sind. Nachdem Sie die verfügbaren Instance-Typen identifiziert haben, wählen Sie die Instance-Größe (z. B. `large`, `xlarge`, oder `2xlarge`) basierend auf der Anzahl der Knoten, die Ihr Workload benötigt. Die folgende Tabelle enthält Empfehlungen für die Auswahl einer Instance-Größe.

> **Note**
>
> Die Instance-Größen müssen auf Ihren Outposts eingestellt werden. Stellen Sie sicher, dass Sie über genügend Kapazität für drei Instances der auf Ihren Outposts verfügbaren Größe für die Lebensdauer Ihres lokalen Clusters verfügen. Eine Liste der verfügbaren Amazon EC2 Instanztypen finden Sie in den Abschnitten Compute und Storage in den AWS Outposts Rack-Funktionen.

Anzahl der Knoten	Instance-Größe der Kubernetes-Steuerebene
1–20	`large`
21–100	`xlarge`
101–250	`2xlarge`
251–500	`4xlarge`

Der Speicher für die Kubernetes-Steuerebene erfordert 246 GB Amazon EBS-Speicher für jeden lokalen Cluster, um die erforderlichen IOPS von `etcd` zu erfüllen. Bei der Erstellung des lokalen Clusters werden die Amazon-EBS-Volumes automatisch für Sie bereitgestellt.

Platzierung der Steuerebene

Wenn Sie keine Platzierungsgruppe mit der `OutpostConfig.ControlPlanePlacement.GroupName`-Eigenschaft angeben, erhalten die für Ihre Kubernetes-Steuerebene bereitgestellten Amazon EC2-Instances keine spezifische Durchsetzung der Hardwareplatzierung über die zugrunde liegende Kapazität, die auf Ihrem Outpost verfügbar ist.

Sie können Platzierungsgruppen verwenden, um die Hochverfügbarkeitsanforderungen für Ihren lokalen Amazon EKS-Cluster auf einem Outpost zu erfüllen. Indem Sie bei der Clustererstellung eine Platzierungsgruppe angeben, beeinflussen Sie die Platzierung der Instances der Kubernetes-Steuerebene. Die Instances sind auf unabhängige zugrundeliegende Hardware (Racks oder Hosts) verteilt, wodurch die Auswirkungen korrelierter Instances bei Hardwareausfällen minimiert werden.

Voraussetzungen

Die Art der Verteilung, die Sie konfigurieren können, hängt von der Anzahl der Outpost-Racks in Ihrer Bereitstellung ab.

- Bereitstellungen mit einem oder zwei physischen Racks in einem einzelnen logischen Outpost – Sie müssen über mindestens drei Hosts verfügen, die mit dem Instance-Typ konfiguriert sind, den Sie für Ihre Instances der Kubernetes-Steuerebene auswählen. Eine verteilte Platzierungsgruppe, die eine Verteilung auf Host-Ebene verwendet, stellt sicher, dass alle Instances der Kubernetes-Steuerebene auf verschiedenen Hosts innerhalb der zugrunde liegenden Racks ausgeführt werden, die in Ihrer Outpost-Bereitstellung verfügbar sind.
- Bereitstellungen mit drei oder mehr physischen Racks in einem einzigen logischen Outpost – Sie müssen mindestens drei Hosts mit dem Instance-Typ konfiguriert haben, den Sie für Ihre Instances der Kubernetes-Steuerebene auswählen. Eine verteilte Platzierungsgruppe, die eine Verteilung auf Rack-Ebene verwendet, stellt sicher, dass alle Instances der Kubernetes-Steuerebene auf unterschiedlichen Racks in Ihrer Outpost-Bereitstellung ausgeführt werden. Sie können alternativ die auf Host-Ebene verteilte Platzierungsgruppe verwenden, wie in der vorherigen Option beschrieben.

Sie sind für die Erstellung der gewünschten Platzierungsgruppe verantwortlich. Sie geben die Platzierungsgruppe an, wenn Sie die `CreateCluster`-API aufrufen. Weitere Informationen zu Placement-Gruppen und deren Erstellung finden Sie unter Placement-Gruppen im Amazon EC2 EC2-Benutzerhandbuch.

Überlegungen

- Wenn eine Platzierungsgruppe angegeben wird, muss verfügbare Slot-Kapazität auf Ihrem Outpost vorhanden sein, um erfolgreich einen lokalen Amazon-EKS-Cluster zu erstellen. Die Kapazität variiert je nachdem, ob Sie den Host- oder Rack-Spread-Typ verwenden. Wenn nicht genügend Kapazität vorhanden ist, verbleibt der Cluster im `Creating`-Zustand. Sie können das `Insufficient Capacity Error` auf dem Zustandsfeld der `DescribeCluster`-API-Antwort überprüfen. Sie müssen Kapazität freigeben, damit der Erstellungsprozess fortgesetzt werden kann.
- Während der Plattform- und Versionsaktualisierungen des lokalen Amazon-EKS-Clusters werden die Instances der Kubernetes-Steuerebene aus Ihrem Cluster durch neue Instances ersetzt, die eine fortlaufende Aktualisierungsstrategie verwenden. Während dieses Ersetzungsprozesses wird jede Instance der Steuerebene beendet, wodurch ihr jeweiliger Slot freigegeben wird. Eine neue aktualisierte Instance wird an ihrer Stelle bereitgestellt. Die aktualisierte Instance wird möglicherweise in den freigegebenen Slot platziert. Wenn der Slot von einer anderen unabhängigen Instance verbraucht wird und keine Kapazität mehr vorhanden ist, die die erforderliche verteilte Topologieanforderung erfüllt, verbleibt der Cluster im `Updating`-Zustand. Sie können das entsprechende `Insufficient Capacity Error` im Zustandsfeld der `DescribeCluster`-API-Antwort anzeigen. Sie müssen Kapazitäten freigeben, damit der Aktualisierungsprozess fortschreiten und vorherige Hochverfügbarkeitsniveaus wiederherstellen kann.
- Sie können jeweils AWS-Region maximal 500 Placement-Gruppen pro Konto erstellen. Weitere Informationen finden Sie unter Allgemeine Regeln und Einschränkungen im Amazon EC2 EC2-Benutzerhandbuch.

Problembehandlung bei lokalen Clustern für Amazon EKS auf AWS Outposts

In diesem Thema werden einige häufige Fehler behandelt, die bei der Verwendung lokaler Cluster auftreten können, und wie Sie diese beheben können. Lokale Cluster ähneln Amazon-EKS-Clustern in der Cloud, es gibt jedoch einige Unterschiede in der Art und Weise, wie sie von Amazon EKS verwaltet werden.

API-Verhalten

Lokale Cluster werden über die Amazon-EKS-API erstellt, aber asynchron ausgeführt. Dies bedeutet, dass Anforderungen an die Amazon-EKS-API für lokale Cluster sofort zurückgegeben werden. Diese

Anforderungen können jedoch erfolgreich sein, aufgrund von Eingabevalidierungsfehlern schnell scheitern oder fehlschlagen und beschreibende Validierungsfehler aufweisen. Dieses Verhalten entspricht dem der Kubernetes-API.

Lokale Cluster wechseln nicht in einen `FAILED`-Status. Amazon EKS versucht, den Cluster-Status kontinuierlich mit dem vom Benutzer angeforderten gewünschten Status abzugleichen. Infolgedessen kann ein lokaler Cluster für längere Zeit im `CREATING`-Status verbleiben, bis das zugrunde liegende Problem behoben ist.

Beschreibung des Bereichs Cluster-Integrität

Probleme mit lokalen Clustern können mit dem `describe-cluster` Amazon EKS AWS CLI-Befehl erkannt werden. Probleme mit lokalen Clustern werden durch das `cluster.health`-Feld der Antwort des `describe-cluster`-Befehls angezeigt. Die in diesem Feld enthaltene Meldung enthält einen Fehlercode, eine beschreibende Meldung und zugehörige Ressourcen-IDs. Diese Informationen sind nur über die Amazon-EKS-API und AWS CLI verfügbar. Ersetzen Sie im folgenden Beispiel *`my-cluster`* durch den Namen Ihres lokalen Clusters.

```
aws eks describe-cluster --name my-cluster --query 'cluster.health'
```

Eine Beispielausgabe sieht wie folgt aus.

```
{
    "issues": [
        {
            "code": "ConfigurationConflict",
            "message": "The instance type 'm5.large' is not supported in Outpost 'my-
outpost-arn'.",
            "resourceIds": [
                "my-cluster-arn"
            ]
        }
    ]
}
```

Wenn das Problem nicht behoben werden kann, müssen Sie möglicherweise den lokalen Cluster löschen und einen neuen erstellen. Versuchen Sie beispielsweise, einen Cluster mit einem Instance-Typ bereitzustellen, der auf Ihrem Outpost nicht verfügbar ist. Die folgende Tabelle enthält allgemeine Fehler im Zusammenhang mit der Integrität.

Fehlerszenario	Code	Fehlermeldung	ResourceIds
Die bereitgestellten Subnetze konnten nicht gefunden werden.	`ResourceNotFound`	`The subnet ID subnet-id does not exist`	Alle bereitgestellten Subnetz-IDs
Bereitgestellte Subnetze gehören nicht zur selben VPC.	`ConfigurationConflict`	`Subnets specified must belong to the same VPC`	Alle bereitgestellten Subnetz-IDs
Einige bereitgestellte Subnetze gehören nicht zum angegebenen Outpost.	`ConfigurationConflict`	`Subnet subnet-id expected to be in outpost-arn , but is in other-outpost-arn`	Problematische Subnetz-ID
Einige bereitgestellte Subnetze gehören zu keinem Outpost.	`ConfigurationConflict`	`Subnet subnet-id is not part of any Outpost`	Problematische Subnetz-ID
Einige bereitgestellte Subnetze verfügen nicht über genügend freie Adressen, um elastische Netzwerkschnittstellen für Instances der Steuerebene zu erstellen.	`ResourceLimitExceeded`	`The specified subnet does not have enough free addresses to satisfy the request.`	Problematische Subnetz-ID
Der angegebene Instance-Typ der Steuerebene wird von Ihrem Outpost nicht unterstützt.	`ConfigurationConflict`	`The instance type type is not supported in Outpost outpost-arn`	Cluster-ARN

Fehlerszenario	Code	Fehlermeldung	ResourceIds
Sie haben eine Amazon-EC2-Instanc e auf der Steuerebe ne beendet oder `run-instance` war erfolgreich, aber der beobachte te Status ändert sich zu `Terminate d` . Dies kann für einen bestimmten Zeitraum passieren, nachdem Ihr Outpost die Verbindung wieder hergestellt hat und interne Fehler von Amazon EBS dazu führen, dass ein interner Amazon-EC2-Workflow fehlschlägt.	`InternalFailure`	`EC2 instance state "Terminat ed" is unexpected`	Cluster-ARN
Sie verfügen über unzureichende Kapazität auf Ihrem Outpost. Dies kann auch bei der Erstellun g eines Clusters passieren, wenn ein Outpost vom AWS-Region getrennt wird.	`ResourceL imitExceeded`	`There is not enough capacity on the Outpost to launch or start the instance.`	Cluster-ARN
Ihr Konto hat das Kontingent Ihrer Sicherheitsgruppe überschritten.	`ResourceL imitExceeded`	Von Amazon-EC2-API zurückgegebene Fehlermeldung	Ziel-VPC-ID

Fehlerszenario	Code	Fehlermeldung	ResourceIds
Ihr Konto hat Ihr Kontingent für die elastische Netzwerkschnittstelle überschritten.	`ResourceLimitExceeded`	Von Amazon-EC2-API zurückgegebene Fehlermeldung	Ziel-Subnetz-ID
Instances der Steuerebene konnten nicht über AWS Systems Manager erreicht werden. Informationen zur Lösung finden Sie unter Instances der Steuerebene sind nicht über AWS Systems Manager erreichbar.	`ClusterUnreachable`	Amazon-EKS-Steuerebene-Instances sind nicht über SSM erreichbar. Bitte überprüfen Sie Ihre SSM- und Netzwerkkonfiguration und lesen Sie die Dokumentation zur Fehlerbehebung bei EKS on Outposts.	Amazon-EC2-Instance-IDs
Beim Abrufen von Details für eine verwaltete Sicherheitsgruppe oder eine elastische Netzwerkschnittstelle ist ein Fehler aufgetreten.	Basierend auf dem Amazon-EC2-Client-Fehlercode.	Von Amazon-EC2-API zurückgegebene Fehlermeldung	Alle verwalteten Sicherheitsgruppen-IDs

Fehlerszenario	Code	Fehlermeldung	ResourceIds
Beim Autorisieren oder Widerrufen von Eingangsregeln für Sicherheitsgruppen ist ein Fehler aufgetreten. Dies gilt sowohl für die Sicherheitsgruppen des Clusters als auch der Steuerebene.	Basierend auf dem Amazon-EC2-Client-Fehlercode.	Von Amazon-EC2-API zurückgegebene Fehlermeldung	Problematische Sicherheitsgruppen-ID
Beim Löschen einer elastischen Netzwerkschnittstelle für eine Instance der Steuerebene ist ein Fehler aufgetreten.	Basierend auf dem Amazon-EC2-Client-Fehlercode.	Von Amazon-EC2-API zurückgegebene Fehlermeldung	Problematische ID der Elastic-Network-Schnittstelle

In der folgenden Tabelle sind Fehler von anderen AWS-Services aufgeführt, die im Integritätsfeld der `describe-cluster`-Antwort angezeigt werden.

Amazon-EC2-Fehlercode	Code für Cluster-Integritätsprobleme	Beschreibung
`AuthFailure`	`AccessDenied`	Dieser Fehler kann aus einer Vielzahl von Gründen auftreten. Der häufigste Grund ist, dass Sie versehentlich ein Tag entfernt haben, das der Service verwendet, um die Richtlinie für die mit dem Service verknüpfte Rolle von der Steuerebene herabzustufen. Wenn dies auftritt,

Amazon-EC2-Fehlercode	Code für Cluster-Integrität sprobleme	Beschreibung
		kann Amazon EKS diese AWS-Ressourcen nicht mehr verwalten und überwachen.
`UnauthorizedOperation`	`AccessDenied`	Dieser Fehler kann aus einer Vielzahl von Gründen auftreten. Der häufigste Grund ist, dass Sie versehentlich ein Tag entfernt haben, das der Service verwendet, um die Richtlinie für die mit dem Service verknüpfte Rolle von der Steuerebene herabzustufen. Wenn dies auftritt, kann Amazon EKS diese AWS-Ressourcen nicht mehr verwalten und überwachen.
`InvalidSubnetID.NotFound`	`ResourceNotFound`	Dieser Fehler tritt auf, wenn die Subnetz-ID für die Eingangsregeln einer Sicherheitsgruppe nicht gefunden werden kann.
`InvalidPermission.NotFound`	`ResourceNotFound`	Dieser Fehler tritt auf, wenn die Berechtigungen für die Eingangsregeln einer Sicherheitsgruppe nicht korrekt sind.
`InvalidGroup.NotFound`	`ResourceNotFound`	Dieser Fehler tritt auf, wenn die Gruppe der Eingangsregeln einer Sicherheitsgruppe nicht gefunden werden kann.

Amazon-EC2-Fehlercode	Code für Cluster-Integrität sprobleme	Beschreibung
`InvalidNetworkInterfaceID.NotFound`	`ResourceNotFound`	Dieser Fehler tritt auf, wenn die Netzwerkschnittstellen-ID für die Eingangsregeln einer Sicherheitsgruppe nicht gefunden werden kann.
`InsufficientFreeAddressesInSubnet`	`ResourceLimitExceeded`	Dieser Fehler tritt auf, wenn das Kontingent der Subnetzressourcen überschritten wird.
`InsufficientCapacityOnOutpost`	`ResourceLimitExceeded`	Dieser Fehler tritt auf, wenn das Kapazitätskontingent des Außenpostens überschritten wird.
`NetworkInterfaceLimitExceeded`	`ResourceLimitExceeded`	Dieser Fehler tritt auf, wenn das Kontingent der elastischen Netzwerkschnittstelle überschritten wird.
`SecurityGroupLimitExceeded`	`ResourceLimitExceeded`	Dieser Fehler tritt auf, wenn das Kontingent der Sicherheitsgruppe überschritten wird.

Amazon-EC2-Fehlercode	Code für Cluster-Integrität sprobleme	Beschreibung
`VcpuLimitExceeded`	`ResourceLimitExceeded`	Dies wird beim Erstellen einer Amazon-EC2-Instanc e in einem neuen Konto beobachtet. Die Fehlermel dung könnte ähnlich wie die folgende lauten: „`You have requested more vCPU capacity than your current vCPU limit of 32 allows for the instance bucket that the specified instance type belongs to. Please visit http://aws.amazon.com/contact-us/ec2-request to request an adjustment to this limit."`
`InvalidParameterValue`	`ConfigurationConflict`	Amazon EC2 gibt diesen Fehlercode zurück, wenn der angegebene Instance-Typ auf dem Outpost nicht unterstützt wird.
Alle anderen Fehler	`InternalFailure`	None

Cluster können nicht erstellt oder geändert werden

Lokale Cluster erfordern andere Berechtigungen und Richtlinien als Amazon-EKS-Cluster, die in der Cloud gehostet werden. Wenn ein Cluster nicht erstellt werden kann und einen `InvalidPermissions` Fehler erzeugt, überprüfen Sie erneut, ob der von Ihnen verwendeten Clusterrolle die von AmazonEKSLocalOutpostClusterPolicy verwaltete Richtlinie zugeordnet ist. Alle anderen API-Aufrufe erfordern dieselben Berechtigungen wie Amazon-EKS-Cluster in der Cloud.

Cluster bleibt im **`CREATING`**-Zustand hängen

Die Zeit, die zum Erstellen eines lokalen Clusters benötigt wird, hängt von mehreren Faktoren ab. Zu diesen Faktoren gehören Ihre Netzwerkkonfiguration, die Outpost-Konfiguration und die Konfiguration des Clusters. Im Allgemeinen wird ein lokaler Cluster erstellt und wechselt innerhalb von 15–20 Minuten in den `ACTIVE`-Status. Wenn ein lokaler Cluster im `CREATING`-Status bleibt, können Sie `describe-cluster`aufrufen, um Informationen über die Ursache im `cluster.health`-Ausgabefeld abzurufen.

Die am häufigsten auftretenden Probleme sind:

AWS Systems Manager (Systems Manager) stößt auf die folgenden Probleme:

- Ihr Cluster kann von dem AWS-Region, in dem sich Systems Manager befindet, keine Verbindung zur Instance der Steuerebene herstellen. Sie können dies überprüfen, indem Sie `aws ssm start-session --target` *`instance-id`* von einem Bastion-Host in der Region aufrufen. Wenn dieser Befehl nicht funktioniert, überprüfen Sie, ob Systems Manager auf der Instance der Steuerebene ausgeführt wird. Eine andere Möglichkeit besteht darin, den Cluster zu löschen und ihn dann neu zu erstellen.
- Instances auf der Steuerebene von Systems Manager haben möglicherweise keinen Zugriff auf das Internet. Überprüfen Sie, ob das Subnetz, das Sie beim Erstellen des Clusters angegeben haben, über ein NAT-Gateway und eine VPC mit einem Internet-Gateway verfügt. Verwenden Sie VPC Reachability Analyzer, um zu überprüfen, ob die Instance der Steuerebene das Internet-Gateway erreichen kann. Weitere Informationen finden Sie unter Erste Schritte mit VPC Reachability Analyzer.
- Der von Ihnen angegebene Rollen-ARN enthält keine Richtlinien. Überprüfen Sie, ob das AWS verwaltete Richtlinie: AmazonEKS LocalOutpostClusterPolicy aus der Rolle entfernt wurde. Dies kann auch auftreten, wenn ein AWS CloudFormation-Stack falsch konfiguriert ist.

Beim Erstellen eines Clusters werden mehrere Subnetze falsch konfiguriert und angegeben:

- Alle bereitgestellten Subnetze müssen demselben Outpost zugeordnet sein und sich gegenseitig erreichen. Wenn beim Erstellen eines Clusters mehrere Subnetze angegeben werden, versucht Amazon EKS, die Instances der Steuerebene auf mehrere Subnetze zu verteilen.
- Amazon-EKS-verwaltete Sicherheitsgruppen werden auf die Elastic-Network-Schnittstelle angewendet. Andere Konfigurationselemente wie NACL-Firewall-Regeln könnten jedoch mit den Regeln für die elastische Netzwerkschnittstelle in Konflikt geraten.

VPC- und Subnetz-DNS-Konfiguration ist falsch konfiguriert oder fehlt

Sehen Sie sich Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen an.

Knoten können keinem Cluster hinzugefügt werden

Häufige Ursachen:

- AMI-Probleme:
  - Sie verwenden ein nicht unterstütztes AMI. Sie müssen v20220620 oder höher für das Amazon EKS-optimierte Amazon Linux-AMIs Amazon EKS optimierte Amazon Linux verwenden.
  - Wenn Sie eine AWS CloudFormation-Vorlage zum Erstellen Ihrer Knoten verwendet haben, stellen Sie sicher, dass kein nicht unterstütztes AMI verwendet wurde.
- AWS IAM Authenticator `ConfigMap` fehlt – Wenn es fehlt, müssen Sie es erstellen. Weitere Informationen finden Sie unter Anwenden von `aws-authConfigMap` auf Ihren Cluster.
- Die falsche Sicherheitsgruppe wird verwendet – Stellen Sie sicher, dass Sie `eks-cluster-sg-`*`cluster-name`*`-`*`uniqueid`* für die Sicherheitsgruppe Ihrer Worker-Knoten verwenden. Die ausgewählte Sicherheitsgruppe wird durch AWS CloudFormation geändert, um bei jeder Verwendung des Stacks eine neue Sicherheitsgruppe zuzulassen.
- Unerwartete VPC-Schritte für private Links – Falsche CA-Daten (`--b64-cluster-ca`) oder API-Endpunkt (`--apiserver-endpoint`) sind bestanden.
- Fehlkonfigurierte Pod-Sicherheitsrichtlinie:
  - Die CoreDNS- und Amazon VPC CNI plugin for Kubernetes-Daemonsets müssen auf Knoten ausgeführt werden, damit Knoten dem Cluster beitreten und mit ihm kommunizieren können.
  - Das Amazon VPC CNI plugin for Kubernetes erfordert einige privilegierte Netzwerkfeatures, um ordnungsgemäß zu funktionieren. Sie können die privilegierten Netzwerkfeatures mit dem folgenden Befehl anzeigen: `kubectl describe psp eks.privileged`.

  Wir empfehlen nicht, die standardmäßige Richtlinie für die Pod-Sicherheit zu ändern. Weitere Informationen finden Sie unter Pod-Sicherheitsrichtlinie.

Sammeln von Protokollen

Wenn ein Außenposten von dem AWS-Region getrennt wird, dem er zugeordnet ist, funktioniert der Kubernetes-Cluster wahrscheinlich weiterhin normal. Wenn der Cluster jedoch nicht richtig funktioniert, befolgen Sie die Schritte zur Fehlerbehebung unter Vorbereitungen für

Netzwerkunterbrechungen. Wenn Sie auf andere Probleme stoßen, wenden Sie sich an AWS Support. AWS Support kann Sie beim Herunterladen und Ausführen eines Tools zur Protokollerfassung unterstützen. Auf diese Weise können Sie Protokolle von den Steuerebenen-Instances Ihres Kubernetes-Clusters erfassen und zur weiteren Untersuchung an den AWS Support-Support senden.

Instances der Steuerebene sind nicht über AWS Systems Manager erreichbar

Wenn die Amazon-EKS-Steuerebene-Instances nicht über AWS Systems Manager (Systems Manager) erreichbar sind, zeigt Amazon EKS den folgenden Fehler für Ihren Cluster an.

```
Amazon EKS control plane instances are not reachable through SSM. Please verify your
 SSM and network configuration, and reference the EKS on Outposts troubleshooting
 documentation.
```

Um dieses Problem zu lösen, stellen Sie sicher, dass Ihre VPC und Subnetze die Anforderungen in Amazon EKS lokale Cluster-VPC- und Subnetz-Anforderungen und -Überlegungen erfüllen und dass Sie die Schritte in Einrichten von Session Manager im AWS Systems Manager-Benutzerhandbuch abgeschlossen haben.

Starten selbstverwalteter Amazon Linux-Knoten auf einem Outpost

Dieses Thema beschreibt Hinweise zum Starten von Auto-Scaling-Gruppen von Amazon-Linux-Knoten auf einem Outpost, die mit Ihrem Amazon-EKS-Cluster registriert sind. Der Cluster kann sich auf dem AWS Cloud oder auf einem Außenposten befinden.

Voraussetzungen

- Ein vorhandener Outpost. Weitere Informationen finden Sie unter Was ist AWS Outposts.
- Ein vorhandener Amazon-EKS-Cluster. Informationen zur Bereitstellung eines Clusters auf dem finden Sie AWS Cloud unterErstellen eines Amazon-EKS-Clusters. Informationen zum Bereitstellen eines Clusters in einem Outpost finden Sie unter Lokale Cluster für Amazon EKS auf AWS Outposts.
- Angenommen, Sie erstellen Ihre Knoten in einem Cluster auf dem AWS Cloud und Sie haben Subnetze in dem Bereich, in AWS-Region dem Sie AWS Outposts, AWS Wavelength, oder AWS Local Zones aktiviert haben. Diese Subnetze dürfen dann bei der Erstellung Ihres Clusters nicht angegeben worden sein. Wenn Sie Ihre Knoten in einem Cluster auf einem Outpost erstellen, müssen Sie bei der Erstellung Ihres Clusters ein Outpost-Subnetz angegeben haben.

- (Empfohlen für Cluster auf dem AWS Cloud) Das Amazon VPC CNI plugin for Kubernetes Add-on wurde mit einer eigenen IAM-Rolle konfiguriert, der die erforderliche IAM-Richtlinie zugeordnet ist. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA). Lokale Cluster unterstützen keine IAM-Rollen für Service-Konten.

Sie können eine selbstverwaltete Amazon Linux-Knotengruppe mit `eksctl` oder AWS Management Console (mit einer AWS CloudFormation Vorlage) erstellen. Sie können auch Terraform verwenden.

eksctl

Voraussetzung

Version `0.183.0` oder höher des `eksctl`-Befehlszeilen-Tools, das auf Ihrem Computer oder in der AWS CloudShell installiert ist. Informationen zum Installieren und Aktualisieren von `eksctl` finden Sie in der Dokumentation zu `eksctl` unter Installation.

So starten Sie selbstverwaltete Linux-Knoten mit **`eksctl`**

1. Wenn Ihr Cluster in der AWS Cloud ist und die von AmazonEKS_CNI_Policy verwaltete IAM-Richtlinie an Ihre Amazon-EKS-Knoten-IAM-Rolle angefügt ist, sollten Sie sie stattdessen einer IAM-Rolle zuzuweisen, die Sie dem Kubernetes `aws-node`-Service-Konto zuordnen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA). Wenn sich Ihr Cluster in Ihrem Outpost befindet, muss die Richtlinie an Ihre Knotenrolle angehängt sein.
2. Der folgende Befehl erstellt eine Knotengruppe in einem bestehenden Cluster. Der Cluster muss mit `eksctl` erstellt worden sein. Ersetzen Sie *`al-nodes`* durch einen Namen für Ihre Knotengruppe. Der Knotengruppenname darf nicht länger als 63 Zeichen sein. Er muss mit einem Buchstaben oder einer Ziffer beginnen, kann danach aber auch Bindestriche und Unterstriche enthalten. Ersetzen Sie *`my-cluster`* mit dem Namen Ihres Clusters. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Sie muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein. Wenn Ihr Cluster auf einem Outpost existiert, ersetzen Sie *`id`* mit der ID eines Outpost-Subnetzes. Wenn Ihr Cluster auf dem existiert AWS Cloud, *`id`* ersetzen Sie ihn durch die ID eines Subnetzes, das Sie bei der Erstellung Ihres Clusters nicht angegeben haben. Ersetzen Sie *`instance-type`* durch einen Instance-Typ, der von Ihrem Outpost unterstützt wird. Ersetzen Sie den Rest

der *example values* durch Ihre eigenen Werte. Die Knoten werden standardmäßig mit derselben Kubernetes-Version wie die Steuerebene erstellt.

Ersetzen Sie *instance-type* durch einen Instance-Typ, der auf Ihrem Outpost verfügbar ist.

Ersetzen Sie *my-key* mit dem Namen Ihres Amazon-EC2-Schlüsselpaars oder öffentlichen Schlüssels. Dieser Schlüssel wird für den SSH-Zugriff zu Ihren Knoten verwendet, nachdem diese gestartet wurden. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Schlüsselpaare im Amazon-EC2-Benutzerhandbuch.

Erstellen Sie Ihre Knoten-Gruppe mit dem folgenden Befehl.

```
eksctl create nodegroup --cluster my-cluster --name al-nodes --node-
type instance-type \
    --nodes 3 --nodes-min 1 --nodes-max 4 --managed=false --node-volume-type gp2
 --subnet-ids subnet-id
```

Wenn Ihr Cluster bereitgestellt wird auf AWS Cloud:

- Die von Ihnen bereitgestellte Knotengruppe kann `IPv4`-Adressen an Pods aus einem anderen CIDR-Block als dem der Instance zuweisen. Weitere Informationen finden Sie unter Benutzerdefinierte Netzwerke für Pods.
- Die von Ihnen bereitgestellte Knotengruppe benötigt keinen ausgehenden Internetzugriff. Weitere Informationen finden Sie unter Anforderungen an private Cluster.

Eine vollständige Liste aller verfügbaren Optionen und Standardwerte finden Sie unter AWS Outposts -Support in der `eksctl`-Dokumentation.

Wenn Knoten dem Cluster nicht beitreten können, finden Sie weitere Informationen unter Knoten können nicht mit dem Cluster verknüpft werden in Amazon-EKS-Fehlerbehebung und Knoten können keinem Cluster hinzugefügt werden in Problembehandlung bei lokalen Clustern für Amazon EKS auf AWS Outposts.

Eine Beispielausgabe sieht wie folgt aus. Mehrere Zeilen werden ausgegeben, während die Knoten erstellt werden. Die letzte Ausgabezeile ähnelt der folgenden Beispielzeile.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Optional) Stellen Sie eine Beispielanwendung bereit, um Ihren Cluster und Ihre Linux-Worker-Knoten zu testen.

AWS Management Console

Schritt 1: So starten Sie selbstverwaltete Amazon Linux-Knoten mit dem AWS Management Console

1. Laden Sie die neueste Version der AWS CloudFormation Vorlage herunter.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2022-12-23/amazon-eks-nodegroup.yaml
```

2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie Create stack (Stack erstellen) und dann With new resources (standard) (Mit neuen Ressourcen [Standard]) aus.
4. Wählen Sie für Specify template (Vorlage festlegen) Upload a template file (Vorlagendatei hochladen) aus und wählen Sie dann Choose file (Datei wählen). Wählen Sie die `amazon-eks-nodegroup.yaml`-Datei aus, die Sie in einem vorherigen Schritt heruntergeladen haben, und wählen Sie dann Next (Weiter) aus.
5. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) die folgenden Parameter ein und klicken Sie dann auf Next (Weiter):

   - Stack name (Stack-Name): Wählen Sie einen Stack-Namen für Ihren AWS CloudFormation -Stack aus. Sie können ihn beispielsweise ***al-nodes*** nennen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Es muss mit einem alphanumerischen Zeichen beginnen und darf nicht länger als 100 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
   - ClusterName: Geben Sie den Namen Ihres Clusters ein. Wenn dieser Name nicht mit Ihrem Cluster-Namen übereinstimmt, können Ihre Knoten dem Cluster nicht beitreten.
   - ClusterControlPlaneSecurityGruppe: Wählen Sie den SecurityGroupsWert aus der AWS CloudFormation Ausgabe aus, die Sie bei der Erstellung Ihrer VPC generiert haben.

Die folgenden Schritte zeigen einen Vorgang zum Abrufen der entsprechenden Gruppe.

1. Öffnen Sie die Amazon-EKS-Konsole unterhttps://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie den Namen des Clusters.
3. Wählen Sie die Registerkarte Network (Network) aus.
4. Verwenden Sie den Wert Zusätzliche Sicherheitsgruppen als Referenz, wenn Sie aus der Dropdownliste ClusterControlPlaneSecurityGruppe auswählen.

- NodeGroupName: Geben Sie einen Namen für Ihre Knotengruppe ein. Dieser Name kann zu einem späteren Zeitpunkt zum Identifizieren der Auto-Scaling-Knotengruppe verwendet werden, die für Ihre Knoten erstellt wurde.
- NodeAutoScalingGroupMinSize: Geben Sie die Mindestanzahl von Knoten ein, auf die Ihre Auto Scaling Scaling-Gruppe für Knoten skalieren kann.
- NodeAutoScalingGroupDesiredCapacity: Geben Sie die gewünschte Anzahl von Knoten ein, auf die bei der Erstellung Ihres Stacks skaliert werden soll.
- NodeAutoScalingGroupMaxSize: Geben Sie die maximale Anzahl von Knoten ein, auf die Ihre Auto Scaling Scaling-Gruppe für Knoten skalieren kann.
- NodeInstanceTyp: Wählen Sie einen Instance-Typ für Ihre Knoten. Wenn Ihr Cluster auf dem läuft AWS Cloud, finden Sie weitere Informationen unterAuswählen eines Amazon-EC2-Instance-Typs. Wenn Ihr Cluster auf einem Outpost ausgeführt wird, können Sie nur einen Instance-Typ auswählen, der auf Ihrem Outpost verfügbar ist.
- NodeImageidssmParam: Vorab mit dem Amazon EC2 Systems Manager Manager-Parameter eines kürzlich für Amazon EKS optimierten AMI für eine variable Version gefüllt. Kubernetes Um eine andere Kubernetes-Nebenversion zu verwenden, die von Amazon EKS unterstützt wird, ersetzen Sie *1.XX* durch eine andere unterstützte Version. Wir empfehlen, dieselbe Kubernetes-Version wie Ihr Cluster anzugeben.

  Um das Amazon-EKS-optimierte beschleunigte AMI zu verwenden, ersetzen Sie *amazon-linux-2* durch **amazon-linux-2-gpu**. Um das Amazon-EKS-optimierte Arm-AMI zu verwenden, ersetzen Sie *amazon-linux-2* durch **amazon-linux-2-arm64**.

  > **Note**
  >
  > Das Amazon EKS Node AMI basiert auf Amazon Linux. Sie können Sicherheits- oder Datenschutzereignisse für Amazon Linux 2 im Amazon Linux-

Sicherheitszentrum verfolgen oder den zugehörigen RSS-Feed abonnieren. Sicherheits- oder Datenschutzereignisse enthalten eine Übersicht über das Problem, welche Pakete betroffen sind und wie Sie Ihre Instances aktualisieren, um das Problem zu beheben.

- NodeImageID: (Optional) Wenn Sie Ihr eigenes benutzerdefiniertes AMI (anstelle des für Amazon EKS optimierten AMI) verwenden, geben Sie eine Knoten-AMI-ID für Ihr AWS-Region. Wenn Sie hier einen Wert angeben, überschreibt dieser alle Werte im Feld NodeImageidsSMParam.
- NodeVolumeGröße: Geben Sie eine Root-Volume-Größe für Ihre Knoten in GiB an.
- NodeVolumeTyp: Geben Sie einen Root-Volume-Typ für Ihre Knoten an.
- KeyName: Geben Sie den Namen eines Amazon EC2 SSH-Schlüsselpaars ein, mit dem Sie sich nach dem Start über SSH mit Ihren Knoten verbinden können. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Schlüsselpaare im Amazon-EC2-Benutzerhandbuch.

  Note

  Wenn Sie hier kein key pair angeben, schlägt die AWS CloudFormation Stack-Erstellung fehl.

- BootstrapArguments: Es gibt mehrere optionale Argumente, die Sie an Ihre Knoten übergeben können. Weitere Informationen finden Sie in den Bootstrap-Skript-Nutzungsinformationen auf GitHub. Wenn Sie Knoten zu einem lokalen Amazon EKS-Cluster hinzufügen AWS Outposts (auf dem die Instances der Kubernetes Kontrollebene ausgeführt werden AWS Outposts) und der Cluster keine Eingangs- und Ausgangsinternetverbindung hat (auch als private Cluster bezeichnet), müssen Sie die folgenden Bootstrap-Argumente (als einzelne Zeile) angeben.

  ```
  --b64-cluster-ca ${CLUSTER_CA} --apiserver-endpoint https://
  ${APISERVER_ENDPOINT} --enable-local-outpost true --cluster-id ${CLUSTER_ID}
  ```

- DisableIMDSv1: Standardmäßig unterstützt jeder Knoten die Instance-Metadaten-Service-Version 1 (IMDSv1) und IMDSv2. Sie können IMDSv1 deaktivieren. Um zu verhindern, dass künftige Knoten und Pods in der Knotengruppe IMDSv1 verwenden, legen Sie DisableIMDSv1 (IMDSv1 deaktivieren) auf true (wahr) fest. Weitere Informationen finden

Sie unter Konfiguration des Instance-Metadatenservice. Weitere Informationen zum Einschränken des Zugriffs darauf auf Ihre Knoten finden Sie unter Beschränken Sie den Zugriff auf das Instance-Profil, das dem Worker-Knoten zugewiesen ist.

- VpcId: Geben Sie die ID für die VPC ein, die Sie erstellt haben. Bevor Sie eine VPC auswählen, überprüfen Sie VPC-Anforderungen und -Überlegungen.
- Subnets (Subnetze): Wenn sich Ihr Cluster auf einem Outpost befindet, wählen Sie mindestens ein privates Subnetz in Ihrer VPC. Bevor Sie Subnetze auswählen, überprüfen Sie die Anforderungen und Überlegungen zu Subnetzen. Sie können sehen, welche Subnetze privat sind, indem Sie den jeweiligen Subnetzlink in der Registerkarte Networking (Netzwerk) Ihres Clusters öffnen.

6. Treffen Sie die gewünschte Auswahl auf der Seite Configure stack options (Stackoptionen konfigurieren) und wählen Sie dann Next (Weiter) aus.
7. Aktivieren Sie das Kontrollkästchen links neben I acknowledge that AWS CloudFormation might create IAM resources with custom names („Mir ist bewusst, dass IAM-Ressourcen mit eigenen Namen erstellen kann“) und wählen Sie dann Create Stack (Stack erstellen) aus.
8. Wenn Ihr Stack fertig erstellt wurde, wählen Sie ihn in der Konsole aus und klicken Sie auf Outputs (Ausgänge).
9. Notieren Sie sich die NodeInstanceRolle für die Knotengruppe, die erstellt wurde. Sie benötigen diese, wenn Sie Ihre Amazon-EKS--Arbeitsknoten konfigurieren.

Schritt 2: So aktivieren Sie die Knoten, die Ihrem Cluster beitreten sollen

1. Überprüfen Sie, ob Sie bereits über eine `aws-auth` `ConfigMap` verfügen.

```
kubectl describe configmap -n kube-system aws-auth
```

2. Wenn eine `aws-auth` `ConfigMap` angezeigt wird, aktualisieren Sie sie nach Bedarf.

   a. Öffnen Sie `ConfigMap` zum Bearbeiten.

```
kubectl edit -n kube-system configmap/aws-auth
```

   b. Fügen Sie nach Bedarf einen neuen `mapRoles`-Eintrag hinzu. Setzen Sie den `rolearn` Wert auf den NodeInstanceRollenwert, den Sie im vorherigen Verfahren aufgezeichnet haben.

```
[...]
```

```
data:
  mapRoles: |
    - rolearn: <ARN of instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
[...]
```

c. Speichern Sie die Datei und beenden Sie den Text-Editor.

3. Wenn die Fehlermeldung `Error from server (NotFound): configmaps "aws-auth" not found` angezeigt wird, wenden Sie die standardmäßige `ConfigMap` an.

   a. Laden Sie die Konfigurationszuordnung herunter.

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/aws-auth-cm.yaml
   ```

   b. Stellen Sie in der `aws-auth-cm.yaml` Datei `rolearn` den NodeInstanceRole-Wert ein, den Sie im vorherigen Verfahren aufgezeichnet haben. Hierzu können Sie einen Texteditor verwenden oder *`my-node-instance-role`* ersetzen und den folgenden Befehl ausführen:

   ```
   sed -i.bak -e 's|<ARN of instance role (not instance profile)>|my-node-instance-role|' aws-auth-cm.yaml
   ```

   c. Wenden Sie die Konfiguration an. Die Ausführung dieses Befehls kann einige Minuten dauern.

   ```
   kubectl apply -f aws-auth-cm.yaml
   ```

4. Sehen Sie sich den Status Ihrer Knoten an und warten Sie, bis diese in den `Ready`-Status eintreten.

   ```
   kubectl get nodes --watch
   ```

   Geben Sie `Ctrl+C` ein, um zu einer Shell-Eingabeaufforderung zurückzukehren.

> **Note**
>
> Wenn Sie Autorisierungs- oder Ressourcenfehler erhalten, finden Sie weitere Informationen unter Nicht autorisiert oder Zugriff verweigert (`kubectl`) im Thema zur Fehlerbehebung.

Wenn Knoten dem Cluster nicht beitreten können, finden Sie weitere Informationen unter Knoten können nicht mit dem Cluster verknüpft werden in Amazon-EKS-Fehlerbehebung und Knoten können keinem Cluster hinzugefügt werden in Problembehandlung bei lokalen Clustern für Amazon EKS auf AWS Outposts.

5. Installieren Sie den Amazon-EBS-CSI-Treiber. Weitere Informationen finden Sie unter Installation am GitHub. Stellen Sie im Abschnitt Einrichten der Treiberberechtigung sicher, dass Sie die Anweisungen für die Option Verwenden des IAM-Instance-Profils befolgen. Sie müssen die gp2-Speicherklasse verwenden. Die gp3-Speicherklasse wird nicht unterstützt.

   Führen Sie die folgenden Schritte aus, um eine gp2-Speicherklasse auf Ihrem Cluster zu erstellen.

   1. Führen Sie den folgenden Befehl aus, um die Datei `gp2-storage-class.yaml` zu erstellen.

      ```
      cat >gp2-storage-class.yaml <<EOF
      apiVersion: storage.k8s.io/v1
      kind: StorageClass
      metadata:
        annotations:
          storageclass.kubernetes.io/is-default-class: "true"
        name: ebs-sc
      provisioner: ebs.csi.aws.com
      volumeBindingMode: WaitForFirstConsumer
      parameters:
        type: gp2
        encrypted: "true"
      allowVolumeExpansion: true
      EOF
      ```

   2. Wenden Sie das Manifest auf Ihren Cluster an.

```
kubectl apply -f gp2-storage-class.yaml
```

6. (Nur GPU-Knoten) Wenn Sie einen GPU-Instance-Typ und das mit Amazon EKS optimierte beschleunigte AMI gewählt haben, müssen Sie das NVIDIA-Geräte-Plugin für Kubernetes mit dem folgenden Befehl als DaemonSet auf Ihren Cluster anwenden. Ersetzen Sie *vX.X.X* mit Ihrer gewünschten Nvidia/K8S-Geräte-Plugin-Version, bevor Sie den folgenden Befehl ausführen.

```
kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
```

Schritt 3: Zusätzliche Aktionen

1. (Optional) Stellen Sie eine Beispielanwendung bereit, um Ihren Cluster und Ihre Linux-Worker-Knoten zu testen.
2. Wenn Ihr Cluster auf einem Outpost bereitgestellt wird, überspringen Sie diesen Schritt. Wenn Ihr Cluster auf dem bereitgestellt wird AWS Cloud, sind die folgenden Informationen optional. Wenn die von verwaltete AmazonEKS_CNI_Policy IAM-Richtlinie an Ihre Amazon-EKS-Knoten-IAM-Rolle angefügt ist, sollten Sie sie stattdessen einer IAM-Rolle zuzuweisen, die Sie dem `aws-node`-Servicekonto für Kubernetes zuordnen. Weitere Informationen finden Sie unter Konfigurieren der Amazon VPC CNI plugin for Kubernetes zur Verwendung von IAM-Rollen für Servicekonten (IRSA).

Verwandte Projekte

Diese Open-Source-Projekte erweitern die Funktionalität von Kubernetes-Clustern, einschließlich Amazon-EKS-verwalteten Clustern, die auf oder außerhalb von AWS, ausgeführt werden.

Verwaltungs-Tools

Verwandte Verwaltungs-Tools für Amazon EKS und Kubernetes-Cluster.

eksctl

`eksctl` ist ein einfaches CLI-Tool zum Erstellen von Clustern in Amazon EKS.

- Projekt-URL
- Projekt-Dokumentation
- AWS Open-Source-Blog: eksctl: Amazon-EKS-Cluster mit nur einem Befehl

AWS Controller für Kubernetes

Mit AWS-Controller für Kubernetes können Sie AWS-Ressourcen direkt aus Ihrem Kubernetes-Cluster erstellen und verwalten.

- Projekt-URL
- AWS Open-Source-Blog: AWS Service-Operator für Kubernetes jetzt verfügbar

Flux CD

Flux ist ein Tool, mit dem Sie Ihre Cluster-Konfiguration mit Git verwalten können. Es verwendet einen Operator im Cluster, um Bereitstellungen innerhalb von Kubernetes auszulösen. Weitere Informationen zu Operatoren finden Sie unter OperatorHub.io auf GitHub.

- Projekt-URL
- Projekt-Dokumentation

CDK für Kubernetes

Mit dem CDK für Kubernetes (cdk8s) können Sie Kubernetes Apps und Komponenten mit vertrauten Programmiersprachen definieren. Cdk8s-Apps synthetisieren zu Standard-Kubernetes-Manifests, die auf jeden Kubernetes-Cluster angewendet werden können.

- Projekt-URL
- Projekt-Dokumentation
- AWS Container-Blog: Einführung in cdk8s+: Absichtsgesteuerte APIs für Kubernetes-Objekte

Netzwerk

Verwandte Netzwerkprojekte für Amazon EKS und Kubernetes-Cluster.

Amazon VPC CNI plugin for Kubernetes

Amazon EKS unterstützt native VPC-Netzwerke über Amazon VPC CNI plugin for Kubernetes. Das Plug-In weist jedem Pod eine IP-Adresse von Ihrer VPC zu.

- Projekt-URL
- Projekt-Dokumentation

AWS Load Balancer Controller für Kubernetes

Der AWS Load Balancer Controller verwaltet AWS Elastic Load Balancers für einen Kubernetes-Cluster. Er erfüllt eingehende Kubernetes-Ressourcen durch die Bereitstellung von AWS Application Load Balancers. Er erfüllt Kubernetes-Service-Ressourcen durch die Bereitstellung von AWS-Network-Load-Balancers.

- Projekt-URL
- Projekt-Dokumentation

ExternalDNS

ExternalDNS synchronisiert exponierte Kubernetes-Services und -Zugriffe mit DNS-Anbietern wie Amazon Route 53 und AWS Service Discovery.

- Projekt-URL
- Projekt-Dokumentation

Machine Learning

Verwandte Machine-Learning-Projekte für Amazon EKS und Kubernetes-Cluster.

Kubeflow

Ein Machine-Learning-Toolkit für Kubernetes.

- Projekt-URL
- Projekt-Dokumentation
- AWS Open-Source-Blog: Kubeflow in Amazon EKS

Auto Scaling

Verwandte Auto-Scaling-Projekte für Amazon EKS und Kubernetes-Cluster.

Cluster Autoscaler

Cluster Autoscaler ist ein Tool, das die Größe des Kubernetes-Clusters automatisch an die CPU- und Speicher-Auslastung anpasst.

- Projekt-URL
- Projekt-Dokumentation
- Amazon-EKS-Workshop: https://www.eksworkshop.com/

Escalator

Escalator ist ein Batch- oder Auftrag-optimiertes Tool zur automatischen horizontalen Skalierung für Kubernetes.

- Projekt-URL
- Projekt-Dokumentation

Überwachung

Verwandte Überwachungs-Projekte für Amazon EKS und Kubernetes-Cluster.

Prometheus

Prometheus ist ein Open-Source-Toolkit zur Überwachung und Warnung von Systemen.

- Projekt-URL
- Projekt-Dokumentation
- Amazon-EKS-Workshop: https://eksworkshop.com/intermediate/240_monitoring/

Fortlaufende Integration/Fortlaufende Bereitstellung

Verwandte CI/CD-Projekte (Continuous Integration/Continuous Deployment) für Amazon EKS und Kubernetes-Cluster.

Jenkins X

CI/CD-Lösung für moderne Cloud-Anwendungen auf Amazon EKS und Kubernetes-Clustern.

- Projekt-URL
- Projekt-Dokumentation

Neue Funktionen und Roadmap von Amazon EKS

Sie können mehr über die neuen Funktionen von Amazon EKS erfahren, indem Sie zum Feed „Neuheiten“ auf der Seite Neuheiten bei AWS gehen. Sie können auch die Roadmap auf GitHub ansehen, um sich über bevorstehende Funktionen und Prioritäten zu informieren, damit Sie planen können, wie Sie Amazon EKS in Zukunft nutzen möchten. Sie können uns direktes Feedback zu den Roadmap-Prioritäten geben.

Dokumentverlauf für Amazon EKS

In der folgenden Tabelle werden die wichtigsten Updates und neuen Features für das Amazon-EKS-Benutzerhandbuch beschrieben. Wir aktualisieren die Dokumentation regelmäßig, um das Feedback, das Sie uns senden, einzuarbeiten.

Änderung	Beschreibung	Datum
Kubernetes-Version `1.30`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.30` hinzugefügt.	23. Mai 2024
Versionsupdate für die Amazon EKS-Plattform	Dies ist eine neue Plattform version mit Sicherheitsfixes und -verbesserungen. Dies beinhaltet neue Patch-Versionen von Kubernete s `1.29.41.28.9`, und. `1.27.13`	14. Mai 2024
CoreDNS	CoreDNSAutoscaler passt die Anzahl der Replikate der CoreDNS Bereitstellung in einem EKS-Cluster dynamisch an die Anzahl der Knoten und CPU-Kerne an. Diese Funktion funktioniert für die neueste Plattformversion der EKS-Release-Version CoreDNS `v1.9 1.25` und höher.	14. Mai 2024
für Windows	Das Amazon CloudWatc h Observability Operator Container Insights Add-on ermöglicht jetzt auch Windows Worker-Knoten im Cluster.	10. April 2024

Kubernetes	Neues Thema zu Kubernetes-Konzepten hinzugefügt.	5. April 2024
Restrukturieren Sie Access und IAM-Inhalte	Verschieben Sie bestehende Seiten, die sich auf Access- und IAM-Themen beziehen, wie z. B. Auth Config Map, Zugriffseinträge, Pod-ID und IRSA, in einen neuen Abschnitt. Überarbeiten Sie den Inhalt der Übersicht.	2. April 2024
Bottlerocket Betriebssystemunterstützung für den Amazon S3 CSI-Treiber	Der Mountpoint for Amazon S3 CSI-Treiber ist jetzt kompatibel mit. Bottlerocket	13. März 2024
AWS verwaltete Richtlinienaktualisierungen — Aktualisierung einer bestehenden Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	4. März 2024
2023	Amazon Linux 2023 (AL2023) ist ein neues Linux-basiertes Betriebssystem, das eine sichere, stabile und leistungsstarke Umgebung für Ihre Cloud-Anwendungen bietet.	29. Februar 2024

EKS Pod Identity und IRSA unterstützen Sidecars in Kubernetes1.29	In Kubernetes1.29, Sidecar-Container sind in Amazon EKS-Clustern verfügbar. Sidecar-Container werden mit IAM-Rollen für Dienstkonten oder EKS Pod Identity unterstützt. Weitere Informationen zu Sidecars finden Sie in der Dokumentation unter Sidecar-Container. Kubernetes	26. Februar 2024
Kubernetes-Version 1.29	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version 1.29 hinzugefügt.	23. Januar 2024
Vollständige Version: Amazon EKS Extended Support für Kubernetes Versionen	Der verlängerte Support der Kubernetes-Version ermöglicht es Ihnen, länger als 14 Monate bei einer bestimmten Kubernetes-Version zu bleiben.	16. Januar 2024

Zustandserkennung von Amazon EKS-Clustern in der AWS Cloud	Amazon EKS erkennt Probleme mit Ihren Amazon EKS-Clustern und der Infrastruktur der Cluster-Voraussetzungen in der Cluster-Integrität . Sie können die Probleme mit Ihren EKS-Clustern im AWS Management Console und im Cluster in `health` der EKS-API einsehen. Diese Probleme werden zusätzlich zu den Problemen erkannt, die von der Konsole erkannt und angezeigt werden. Bisher war die Clusterintegrität nur für lokale Cluster verfügbar AWS Outposts.	28. Dezember 2023
AWS-Region	Amazon EKS ist jetzt in der AWS-Region„Kanada West (Calgary)“ (`ca-west-1`) verfügbar.	20. Dezember 2023
in Cluster	Sie können jetzt Empfehlungen für Ihren Cluster erhalten, die auf wiederkehrenden Prüfungen basieren.	20. Dezember 2023

Sie können jetzt mithilfe von Zugriffseinträgen IAM-Rollen und -Benutzern Zugriff auf Ihren Cluster gewähren.	Vor der Einführung von Zugriffseinträgen haben Sie IAM-Rollen und -Benutzern Zugriff auf Ihren Cluster gewährt, indem Sie Einträge zu `aws-auth ConfigMap` hinzugefügt haben. Jetzt verfügt jeder Cluster über einen Zugriffsmodus und Sie können nach Ihrem eigenen Zeitplan auf die Verwendung von Zugriffseinträgen umsteigen. Nachdem Sie zwischen den Modi gewechselt haben, können Sie Benutzer hinzufügen, indem Sie Zugriffseinträge in den AWS CLI SDKs AWS CloudFormation, und hinzufügen. AWS	18. Dezember 2023
Aktualisierung der Amazon-EKS-Plattformversion	Dies ist eine neue Plattformversion mit Sicherheitsfixes und -verbesserungen. Dies schließt neue Patch-Versionen von Kubernetes `1.28.4`, `1.27.8`, `1.26.11` und `1.25.16` mit ein.	12. Dezember 2023
Mountpoint für Amazon-S3-CSI-Treiber	Sie können jetzt den Mountpoint für Amazon-S3-CSI-Treiber auf Amazon-EKS-Clustern installieren.	27. November 2023

Aktivieren von Prometheus-Metriken beim Erstellen eines Clusters	In der können Sie jetzt Prometheus Metriken aktiviere n AWS Management Console, wenn Sie einen Cluster erstellen. Sie können die Prometheus-Scraper-Details auch auf der Registerkarte Beobachtbarkeit anzeigen.	26. November 2023
Amazon-EKS-Pod-Identitäten	Amazon-EKS-Pod-Identitäten verknüpfen eine IAM-Rolle mit einem Kubernetes-Service konto. Mit diesem Feature müssen Sie der Knoten-IA M-Rolle keine erweiterten Berechtigungen mehr bereitste llen. Auf diese Weise können Pods auf diesem Knoten AWS APIs aufgerufen werden. Im Gegensatz zu IAM-Rollen für Servicekonten befinden sich EKS-Pod-Identitäten vollständ ig in EKS. Sie benötigen keinen OIDC-Identitätsanb ieter.	26. November 2023
AWS verwaltete Richtlini enaktualisierungen — Aktualisi erung einer bestehenden Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	26. November 2023
CSI-Snapshot-Controller	Sie können den CSI-Snaps hot-Controller jetzt für die Verwendung mit kompatibl en CSI-Treibern wie dem Amazon-EBS-CSI-Treiber installieren.	17. November 2023

Neuverfassung des Themas „ADOT Operator“	Der Abschnitt zur Amazon-EKS-Add-on-Unterstützung für ADOT Operator wurde durch die Dokumentation zur AWS Distro für OpenTelemetry redundant. Wir haben die verbleibenden wichtigen Informationen auf diese Ressource migriert, um veraltete und inkonsistente Informationen zu reduzieren.	14. November 2023
CoreDNS-EKS-Add-on-Unterstützung für Prometheus-Metriken	Die Versionen `v1.10.1-eksbuild.5` , `v1.9.3-eksbuild.9` und `v1.8.7-eksbuild.8` des EKS-Add-ons für CoreDNS legen den Port, über den CoreDNS Metriken veröffentlicht hat, im `kube-dns`-Service offen. Dies macht es einfacher, die CoreDNS-Metriken in Ihre Überwachungssysteme aufzunehmen.	10. November 2023
Amazon EKS CloudWatch Observability Operator-Add-on	Die Amazon EKS CloudWatch Observability Operator-Seite wurde hinzugefügt.	6. November 2023
Kapazitätsblöcke für selbstverwaltete P5-Instances in USA Ost (Ohio)	In USA Ost (Ohio) können Sie Kapazitätsblöcke jetzt für selbstverwaltete P5-Instances verwenden.	31. Oktober 2023

Cluster unterstützen das Ändern von Subnetzen und Sicherheitsgruppen	Sie können den Cluster aktualisieren, um zu ändern, welche Subnetze und Sicherheitsgruppen der Cluster verwendet. Sie können von der neuesten Version der AWS Management Console, AWS CLI AWS CloudFormation, und Version `v0.164.0-rc.0` oder einer späteren `eksctl` Version aus aktualisieren. Möglicherweise müssen Sie dies tun, um Subnetzen mehr verfügbare IP-Adressen zur Verfügung zu stellen, damit eine Clusterversion erfolgreich aktualisiert werden kann.	24. Oktober 2023

Die Clusterrolle und die Rolle der verwalteten Knotengruppe unterstützen vom Kunden verwaltete AWS Identity and Access Management Richtlinien	Sie können anstelle der `AmazonEKSClusterPolicy` AWS verwalteten Richtlinie eine benutzerdefinierte IAM-Richtlinie für die Clusterrolle verwenden. Außerdem können Sie anstelle der verwalteten Richtlinie eine benutzerdefinierte IAM-Richtlinie für die Knotenrolle in einer verwalteten Knotengruppe verwenden. `AmazonEKSWorkerNodePolicy` AWS Auf diese Weise können Sie eine Richtlinie mit der geringsten Berechtigung erstellen, um strenge Compliance-Anforderungen zu erfüllen.	23. Oktober 2023
Link zur Installation von eksctl korrigiert	Korrigiert den Installationslink für eksctl, nachdem die Seite verschoben wurde.	06. Oktober 2023
Vorversion: Amazon EKS verlängerter Support für Kubernetes-Versionen	Der verlängerte Support der Kubernetes-Version ermöglicht es Ihnen, länger als 14 Monate bei einer bestimmten Kubernetes-Version zu bleiben.	04. Oktober 2023
Verweise auf die Integration AWS App Mesh entfernen	Amazon EKS-Integrationen AWS App Mesh bleiben nur bestehenden Kunden von App Mesh vorbehalten.	29. September 2023

Kubernetes-Version `1.28`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.28` hinzugefügt.	26. September 2023
Bestehende Cluster unterstützen die Durchsetzung von Kubernetes-Netzwerkrichtlinien im Amazon VPC CNI plugin for Kubernetes	Sie können die Kubernetes-Netzwerkrichtlinie in bestehenden Clustern mit dem Amazon VPC CNI plugin for Kubernetes verwenden, anstatt eine Drittanbieterlösung zu benötigen.	15. September 2023
Das Amazon-EKS-Add-on für CoreDNS unterstützt das Ändern von PDB	Sie können das `PodDisruptionBudget` des EKS-Add-ons für CoreDNS in Versionen `v1.9.3-eksbuild.7` und höher sowie `v1.10.1-eksbuild.4` und höher ändern.	15. September 2023
Amazon-EKS-Unterstützung für gemeinsam genutzte Subnetze	Neue Anforderungen und Überlegungen für gemeinsam genutzte Subnetze für die Erstellung von Amazon-EKS-Clustern in gemeinsam genutzten Subnetzen.	07. September 2023
Aktualisierungen an „Was ist Amazon EKS?“	Neue Themen in Häufige Anwendungsfälle und Architektur hinzugefügt. Andere Themen wurden aktualisiert.	6. September 2023

Durchsetzung von Kubernetes-Netzwerkrichtlinien im Amazon VPC CNI plugin for Kubernetes	Sie können die Kubernetes-Netzwerkrichtlinie mit dem Amazon VPC CNI plugin for Kubernetes verwenden, anstatt eine Drittanbieterlösung zu benötigen.	29. August 2023
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in der AWS-Region Israel (Tel Aviv) verfügbar (`il-central-1`).	1. August 2023
Konfigurierbarer flüchtiger Speicher für Fargate	Sie können die Gesamtmenge des kurzlebigen Speichers für jeden Pod erhöhen, der auf Amazon EKS Fargate läuft.	31. Juli 2023
Add-on-Support für Amazon-EFS-CSI-Treiber	Sie können jetzt die API AWS Management Console, und verwenden AWS CLI, um den Amazon EFS CSI-Treiber zu verwalten.	26. Juli 2023
AWS verwaltete Richtlinienaktualisierungen — Neue Richtlinie	Amazon EKS hat eine neue AWS verwaltete Richtlinie hinzugefügt.	26. Juli 2023
KubernetesVersionsupdates für 1.27, 1.26, 1.25 und 1.24 sind jetzt für lokale Cluster verfügbar AWS Outposts	KubernetesVersionsupdates für 1.27.3, 1.26.6, 1.25.11 und 1.24.15 sind jetzt für lokale Cluster auf verfügbar AWS Outposts	20. Juli 2023

Unterstützung von IP-Präfixen für Windows-Knoten	Wenn Sie Ihren Knoten IP-Präfixe zuweisen, können Sie auf Ihren Knoten eine deutlich höhere Anzahl von Pods hosten, als wenn Sie Ihren Knoten einzelne sekundäre IP-Adressen zuweisen.	6. Juli 2023
Amazon FSx für OpenZFS-CSI-Treiber	Sie können jetzt den CSI-Treiber für Amazon FSx für OpenZFS in Amazon-EKS-Clustern installieren.	30. Juni 2023
Pods auf Linux- Knoten in `IPv4`-Clustern können jetzt mit `IPv6`-Endpunkten kommunizieren.	Nachdem Sie Ihrem Knoten eine IPv6-Adresse zugewiesen haben, wird die `IPv4`-Adresse Ihrer Pods in die in die `IPv6`-Adresse des Knotens übersetzt, auf dem sie läuft.	19. Juni 2023
Windowsverwaltete Knotengruppen in AWS GovCloud (US) Regions	In den AWS GovCloud (US) Regions von Amazon EKS verwalteten Knotengruppen können jetzt Windows Container ausgeführt werden.	30. Mai 2023
Kubernetes-Version `1.27`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.27` hinzugefügt.	24. Mai 2023
Kubernetes-Version `1.26`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.26` hinzugefügt.	11. April 2023

Domainlose gMSA	Sie können jetzt domainlose gMSA mit Windows-Pods verwenden.	27. März 2023
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt im asiatisch-pazifischen Raum (Melbourne) verfügbar (`ap-southeast-4`) AWS-Region.	10. März 2023
CSI-Treiber für Amazon File Cache	Sie können jetzt den CSI-Treiber für Amazon File Cache in Amazon-EKS-Clustern installieren.	3. März 2023
KubernetesVersion 1.25 ist jetzt für lokale Cluster verfügbar auf AWS Outposts	Sie können jetzt einen lokalen Amazon-EKS-Cluster auf einem Outpost mit Kubernetes-Versionen erstellen `1.22` – `1.25`.	1. März 2023
Kubernetes-Version `1.25`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.25` hinzugefügt.	22. Februar 2023
AWS verwaltete Richtlinienaktualisierungen — Aktualisierung einer bestehenden Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	07. Februar 2023
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt im asiatisch-pazifischen Raum (Hyderabad`ap-south-2`) (), Europa (Zürich) (`eu-central-2`) und Europa (Spanien) (`eu-south-2`) AWS-Regionen verfügbar.	06. Februar 2023

KubernetesVersionen `1.21` — `1.24` sind jetzt für lokale Cluster auf AWS Outposts verfügbar.	Sie können jetzt einen lokalen Amazon-EKS-Cluster auf einem Outpost mit Kubernetes-Versionen erstellen `1.21` – `1.24`. Bisher war nur die Version `1.21` verfügbar.	17. Januar 2023
Amazon EKS unterstützt jetzt AWS PrivateLink	Sie können eine verwenden AWS PrivateLink , um eine private Verbindung zwischen Ihrer VPC und Amazon EKS herzustellen.	16. Dezember 2022
Windows-Unterstützung für verwaltete Knotengruppen	Sie können jetzt Windows für von Amazon EKS verwaltete Knotengruppen verwenden.	15. Dezember 2022
Amazon-EKS-Add-Ons von unabhängigen Softwareanbietern sind jetzt in AWS Marketplace verfügbar	Sie können jetzt Amazon-EKS-Add-Ons von unabhängigen Softwareanbietern über die AWS Marketplace suchen und abonnieren.	28. November 2022
AWS verwaltete Richtlinienaktualisierungen — Aktualisierung einer bestehenden Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	17. November 2022
Kubernetes-Version `1.24`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.24` hinzugefügt.	15. November 2022
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt im Nahen Osten (VAE) verfügbar (`me-central-1`) AWS-Region.	03. November 2022

AWS verwaltete Richtlinienaktualisierungen — Aktualisierung einer bestehenden Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	24. Oktober 2022
AWS verwaltete Richtlinienaktualisierungen — Aktualisierung einer bestehenden Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	20. Oktober 2022
Lokale Cluster auf AWS Outposts sind jetzt verfügbar	Sie können jetzt einen lokalen Amazon-EKS-Cluster auf einem Outpost erstellen.	19. September 2022
vCPU-basierte Kontingente in Fargate	Fargate wird von Pod-basierten Kontingenten auf vCPU-basierte Kontingente umgestellt.	8. September 2022
AWS verwaltete Richtlinienaktualisierungen — Aktualisierung einer vorhandenen Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	31. August 2022
Kostenüberwachung	Amazon EKS unterstützt jetzt Kubecost, mit dem Sie Ihre Kosten überwachen können, aufgeschlüsselt nach Kubernetes-Ressourcen einschließlich Pods, Knoten, Namespaces und Labels.	24. August 2022
AWS verwaltete Richtlinienaktualisierungen — Neue Richtlinie	Amazon EKS hat eine neue AWS verwaltete Richtlinie hinzugefügt.	24. August 2022

AWS verwaltete Richtlinienaktualisierungen — Neue Richtlinie	Amazon EKS hat eine neue AWS verwaltete Richtlinie hinzugefügt.	23. August 2022
Markieren von Ressourcen für die Fakturierung	Unterstützung für ein von `aws:eks:cluster-name` generiertes Kostenzuweisungs-Tag wurde für alle Cluster hinzugefügt.	16. August 2022
Fargate-Profil-Platzhalter	Unterstützung für Fargate-Profil-Platzhalter in den Selektorkriterien für Namespaces, Labelschlüssel und Labelwerte wurde hinzugefügt.	16. August 2022
Kubernetes-Version `1.23`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.23` hinzugefügt.	11. August 2022
KubernetesRessourcen anzeigen im AWS Management Console	Sie können jetzt Informationen über Kubernetes-Ressourcen, die in Ihrem Cluster bereitgestellt sind, in der AWS Management Console ansehen.	3. Mai 2022
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt im asiatisch-pazifischen Raum (Jakarta) verfügbar (`ap-southeast-3`) AWS-Region.	2. Mai 2022

Seite „Beobachtbarkeit“ und ADOT-Add-on-Unterstützung	Die Observability-Seite und die AWS Distribution für OpenTelemetry (ADOT) wurden hinzugefügt.	21. April 2022
Kubernetes-Version `1.22`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.22` hinzugefügt.	4. April 2022
AWS verwaltete Richtlini enaktualisierungen — Neue Richtlinie	Amazon EKS hat eine neue AWS verwaltete Richtlinie hinzugefügt.	4. April 2022
Fargate-Pod-Patching-Details hinzugefügt	Beim Upgrade von Fargate-Pods versucht Amazon EKS zunächst, Pods basierend auf Ihren Budgets für Pod-Unter brechungen zu vertreiben. Sie können Ereignisregeln erstellen, um auf fehlgesch lagene Bereinigungen zu reagieren, bevor die Pods gelöscht werden.	1. April 2022
Vollständiger Release: Add-on-Support für Amazon-EBS-CSI-Treiber	Sie können jetzt die API AWS Management Console, und verwenden AWS CLI, um den Amazon EBS CSI-Treiber zu verwalten.	31. März 2022
AWS Outposts Aktualisierung des Inhalts	Anweisungen zum Bereitste llen eines Amazon-EKS-Clusters auf AWS Outposts.	22. März 2022

AWS verwaltete Richtlinienaktualisierungen — Aktualisierung einer bestehenden Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	21. März 2022
Windows `containerd` -Unterstützung	Sie können jetzt die `containerd` -Laufzeit für Windows-Knoten auswählen.	14. März 2022
Überlegungen zum Amazon EKS Connector zur Sicherheitsdokumentation hinzugefügt	Beschreibt das Modell der gemeinsamen Verantwortung in Bezug auf verbundene Cluster.	25. Februar 2022
Weisen Sie Ihren Pods und Services `IPv6`-Adressen zu	Sie können jetzt einen Cluster mit `1.21` oder höher erstellen, der Ihren Pods und Services `IPv6`-Adressen zuweist.	6. Januar 2022
AWS verwaltete Richtlinienaktualisierungen — Aktualisierung einer bestehenden Richtlinie	Amazon EKS hat eine bestehende AWS verwaltete Richtlinie aktualisiert.	13. Dezember 2021
Vorversion: Add-on-Support für Amazon-EBS-CSI-Treiber	Sie können jetzt eine Vorschau anzeigen, indem Sie die API AWS Management Console AWS CLI, und verwenden, um den Amazon EBS CSI-Treiber zu verwalten.	9. Dezember 2021
Support von Karpenter Autoscaler	Sie können jetzt das Karpenter -Open-Source-Projekt verwenden, um Ihre Knoten automatisch zu skalieren.	29. November 2021

Support für Fluent-Bit-Kubernetes-Filter bei der Fargate-Protokollierung	Sie können jetzt den Fluent-Bit-Kubernetes-Filter mit Fargate-Protokollierung verwenden.	10. November 2021
Windows-Support in der Steuerebene verfügbar	Windows-Support ist jetzt in Ihrer Steuerebene verfügbar. Sie müssen ihn nicht mehr in Ihrer Datenebene aktivieren.	9. November 2021
Bottlerocket wurde als AMI-Typ für verwaltete Knotengruppen hinzugefügt	Zuvor war Bottlerocket nur als Option für selbstverwaltete Knoten verfügbar. Jetzt kann es als verwaltete Knotengruppe konfiguriert werden, wodurch der Aufwand reduziert wird, der zur Erfüllung der Knoten-Compliance-Anforderungen erforderlich ist.	28. Oktober 2021
Support für DL1-Treiber	Benutzerdefinierte Amazon-Linux-AMIs unterstützen jetzt Deep-Learning-Workloads fürAmazon-Linux-2. Diese Aktivierung ermöglicht eine generische On-Premises- oder Cloud-Baseline-Konfiguration.	25. Oktober 2021
VT1-Videounterstützung	Benutzerdefinierte Amazon-Linux-AMIs unterstützen jetzt VT1 für einige Distributionen. Diese Aktivierung bewirbt Xilinx U30-Geräte in Ihrem Amazon-EKS-Cluster.	13. September 2021

Amazon EKS Connector ist jetzt verfügbar	Sie können Amazon EKS Connector verwenden, um jeden konformen Kubernetes Cluster zu registrieren, eine Verbindung herzustellen AWS und ihn in der Amazon EKS-Konsole zu visualisieren.	8. September 2021
Amazon EKS Anywhere ist jetzt verfügbar	Amazon EKS Anywhere ist eine neue Bereitstellungsoption für Amazon EKS, mit der Sie Kubernetes-Cluster einfach On-Premises erstellen und betreiben können.	8. September 2021
Amazon FSx für NetApp ONTAP CSI-Treiber	Es wurde ein Thema hinzugefügt, das den Amazon FSx for NetApp ONTAP CSI-Treiber zusammenfasst und Links zu anderen Referenzen enthält.	2. September 2021
Verwaltete Knotengruppen berechnen jetzt automatisch die von Amazon EKS empfohlenen maximalen Pods für Knoten	Verwaltete Knotengruppen berechnen jetzt automatisch die maximalen Amazon-EKS-Pods für Knoten, die Sie ohne eine Startvorlage bereitstellen oder mit einer Startvorlage, in der Sie keine AMI-ID angegeben haben.	30. August 2021
Amazon-EKS-Verwaltung der Add-on-Einstellungen entfernen, ohne die Amazon-EKS-Add-on-Software zu entfernen	Sie können jetzt ein Amazon-EKS-Add-on entfernen, ohne die Add-on-Software aus Ihrem Cluster zu entfernen.	20. August 2021

Erstellen mehrfach vernetzter Pods mit Multus	Sie können jetzt mit Multus mehrere Netzwerkschnittstellen zu einem Pod hinzufügen.	2. August 2021
Fügen Sie Ihren Linux Amazon-EC2-Knoten weitere IP-Adressen hinzu	Sie können Ihren Linux Amazon-EC2-Knoten jetzt deutlich mehr IP-Adressen hinzufügen. Dies bedeutet, dass Sie auf jedem Knoten eine höhere Dichte an Pods ausführen können.	27. Juli 2021
`containerd` -Laufzeit-Bootstrap	Das für Amazon EKS optimierte beschleunigte Amazon Linux Amazon Machine Image (AMI) enthält jetzt ein Bootstrap -Flag, um die `containerd` -Laufzeit in Amazon-EKS-optimierten und Bottlerocket-AMIs zu aktivieren. Dieses Flag ist in allen unterstützten Kubernetes-Versionen des AMI verfügbar.	19. Juli 2021
Kubernetes-Version `1.21`	Support für Kubernetes-Version `1.21` hinzugefügt.	19. Juli 2021
Thema zu verwalteten Richtlinien hinzugefügt	Eine Liste aller von Amazon EKS IAM verwalteten Richtlinien und Änderungen, die seit dem 17. Juni 2021 daran vorgenommen wurden.	17. Juni 2021

Verwenden von Sicherheitsgruppen für Pods mit Fargate	Sie können jetzt zusätzlich zur Verwendung mit Amazon-EC2-Knoten Sicherheitsgruppen für Pods mit Fargate verwenden.	1. Juni 2021
CoreDNS- und `kube-proxy` -Amazon-EKS-Add-ons hinzugefügt	Amazon EKS kann Ihnen jetzt bei der Verwaltung der CoreDNS- und `kube-proxy` -Amazon-EKS-Add-ons für Ihren Cluster helfen.	19. Mai 2021
Kubernetes-Version `1.20`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.20` hinzugefügt.	18. Mai 2021
AWS Load Balancer Controller `r2.2.0` veröffentlicht	Sie können jetzt den AWS Load Balancer Controller verwenden, um Elastic Load Balancer mithilfe von Instance- oder IP-Zielen zu erstellen.	14. Mai 2021
Knoten-Taints für verwaltete Knoten-Gruppen	Amazon EKS unterstützt jetzt das Hinzufügen von Notizmarkierungen zu verwalteten Knotengruppen.	11. Mai 2021
Geheimnisverschlüsselung für vorhandene Cluster	Amazon EKS unterstützt jetzt das Hinzufügen von Geheimnis-Verschlüsselung zu vorhandenen Clustern.	26. Februar 2021
Kubernetes-Version `1.19`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.19` hinzugefügt.	16. Februar 2021

Amazon EKS unterstützt jetzt OpenID Connect (OIDC)-Identitätsanbieter als Methode zur Authentifizierung von Benutzern bei Clustern mit Version `1.16` oder höher.	OIDC-Identitätsanbieter können mit oder als Alternative zu AWS Identity and Access Management (IAM) verwendet werden.	12. Februar 2021
Sehen Sie sich Knoten- und Workload-Ressourcen an im AWS Management Console	Sie können jetzt Details zu Ihren verwalteten, selbstverwalteten und Fargate-Knoten und Ihren bereitgestellten Kubernetes-Workloads in der AWS Management Console.	1. Dezember 2020
Spot-Instance-Typen in einer verwalteten Knotengruppe bereitstellen	Sie können nun mehrere Spot- oder On-Demand-Instance -Typen für eine verwaltete Knotengruppe bereitstellen.	1. Dezember 2020
Amazon EKS kann jetzt spezifische Add-ons für Ihren Cluster verwalten	Sie können Add-Ons selbst verwalten oder Amazon EKS erlauben, den Start und die Version eines Add-Ons über die Amazon-EKS-API zu steuern.	1. Dezember 2020
Teilen Sie eine ALB über mehrere Ingresses	Sie können jetzt einen AWS Application Load Balancer (ALB) für mehrere Kubernetes Ingresses gemeinsam nutzen. In der Vergangenheit mussten Sie für jeden Ingress eine separate ALB bereitstellen.	23. Oktober 2020

Unterstützung von NLB-IP-Zielen	Sie können jetzt einen Network Load Balancer (NLB) mit IP-Zielen bereitstellen. Dies bedeutet, dass Sie einen NLB verwenden können, um den Netzwerkverkehr zu Fargate Pods und direkt zu Pods zu verteilen, die auf Amazon-EC2-Knoten ausgeführt werden.	23. Oktober 2020
Kubernetes-Version `1.18`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.18` hinzugefügt.	13. Oktober 2020
Geben Sie einen benutzerdefinierten CIDR-Block für die Zuweisung der Kubernetes-Service-IP-Adresse an.	Sie können jetzt einen benutzerdefinierten CIDR-Block angeben, von dem Kubernetes-Service-IP-Adressen zuweist.	29. September 2020
Weisen Sie einzelnen Pods Sicherheitsgruppen zu	Sie können jetzt einigen der einzelnen Pods, die auf vielen Amazon-EC2-Instance-Typen ausgeführt werden, verschiedene Sicherheitsgruppen zuordnen.	9. September 2020
Bereitstellen von Bottlerocket auf Ihren Knoten	Sie können jetzt Knoten bereitstellen, auf denen Bottlerocket ausgeführt wird.	31. August 2020
Die Möglichkeit, Arm-Knoten zu starten, ist allgemein verfügbar	Sie können jetzt Arm-Knoten in verwalteten und selbstverwalteten Knotengruppen starten.	17. August 2020

Startvorlagen für verwaltete Knotengruppen und benutzerdefiniertes AMI	Sie können jetzt eine verwaltete Knotengruppe bereitstellen, die eine Amazon-EC2-Startvorlage verwendet. Die Startvorlage kann bei Bedarf ein benutzerdefiniertes AMI angeben.	17. August 2020
EFS-Unterstützung für AWS Fargate	Sie können Amazon EFS jetzt mit verwenden AWS Fargate.	17. August 2020
Aktualisierung der Amazon-EKS-Plattformversion	Dies ist eine neue Plattformversion mit Sicherheitsfixes und -verbesserungen. Dies beinhaltet UDP-Unterstützung für Services vom Typ `LoadBalancer` bei Verwendung von Network Load Balancers mit Kubernetes-Version `1.15` oder höher. Weitere Informationen finden Sie unter dem Problem UDP für AWS Network Load Balancer zulassen unterGitHub.	12. August 2020
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in den AWS-Regionen Afrika (Kapstadt) (`af-south-1`) und Europa (Mailand) (`eu-south-1`) verfügbar.	6. August 2020

Fargate-Nutzungskennzahlen	AWS Fargate bietet CloudWatch Nutzungsmetriken, die Aufschluss darüber geben, wie Ihr Konto die Fargate-On-Demand-Ressourcen nutzt.	3. August 2020
Kubernetes-Version `1.17`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.17` hinzugefügt.	10. Juli 2020
Erstellen und Verwalten von App-Mesh-Ressourcen aus Kubernetes mit dem App-Mesh-Controller für Kubernetes	Sie können App-Mesh-Ressourcen in Kubernetes erstellen und verwalten Kubernetes. Der Controller injiziert außerdem automatisch den Envoy-Proxy und die Init-Container in Pods, die Sie bereitstellen.	18. Juni 2020
Amazon EKS unterstützt jetzt Amazon-EC2-Inf1-Knoten	Sie können Amazon-EC2-Inf1-Knoten zu Ihrem Cluster hinzufügen.	4. Juni 2020
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in den Ländern AWS GovCloud (US-Ost) (`us-gov-east-1`) und (US-West) AWS GovCloud (`us-gov-west-1`) verfügbar. AWS-Regionen	13. Mai 2020

Kubernetes1.12 wird auf Amazon EKS nicht mehr unterstützt	Kubernetes-Version 1.12 wird auf Amazon EKS nicht mehr unterstützt. Aktualisieren Sie alle 1.12-Cluster auf Version 1.13 oder höher, um Service-Unterbrechungen zu vermeiden.	12. Mai 2020
Kubernetes-Version 1.16	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version 1.16 hinzugefügt.	30. April 2020
Die AWSServiceRoleForAmazonEKSserviceverknüpfte Rolle wurde hinzugefügt	Die AWSServiceRoleForAmazonEKSserviceverknüpfte Rolle wurde hinzugefügt.	16. April 2020
Kubernetes-Version 1.15	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version 1.15 hinzugefügt.	10. März 2020
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in den AWS-Regionen Peking (cn-north-1) und Ningxia (cn-northwest-1) verfügbar.	26. Februar 2020
FSx-for-Lustre-CSI-Treiber	Thema für die Installation des CSI-Treibers für FSx for Lustre in Amazon-EKS-Clustern für Kubernetes 1.14 hinzugefügt.	23. Dezember 2019

Beschränken des Netzwerkzugriffs auf den öffentlichen Zugriffsendpunkt eines Clusters	Mit diesem Update können Sie Amazon EKS verwenden, um die CIDR-Bereiche einzuschränken, die mit dem öffentlichen Zugriffsendpunkt des Kubernetes-API-Servers kommunizieren können.	20. Dezember 2019
Auflösen der privaten Zugriffsendpunkt-Adresse für einen Cluster von außerhalb einer VPC	Mit diesem Update können Sie Amazon EKS verwenden, um den privaten Zugriffs-Endpunkt des Kubernetes-API-Servers von außerhalb einer VPC aufzulösen.	13. Dezember 2019
(Beta) Amazon EC2 A1 Amazon-EC2-Instance-Knoten	Starten Sie Amazon EC2 A1 Amazon-EC2-Instance-Knoten, die sich bei Ihrem Amazon-EKS-Cluster registrieren.	4. Dezember 2019
Einen Cluster auf AWS Outposts erstellen	Amazon EKS unterstützt jetzt das Erstellen von Clustern auf AWS Outposts.	3. Dezember 2019
AWS Fargate auf Amazon EKS	Amazon-EKS-Kubernetes-Cluster unterstützen jetzt das Ausführen von Pods auf Fargate.	3. Dezember 2019
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in Kanada (Central) (`ca-central-1`) verfügbar AWS-Region.	21. November 2019

Verwaltete Knotengruppen	Von Amazon EKS verwaltete Knotengruppen automatisieren die Bereitstellung und das Lebenszyklusmanagement von Knoten (Amazon-EC2-Instances) für Amazon-EKS-Kubernetes-Cluster.	18. November 2019
Aktualisierung der Amazon-EKS-Plattformversion	Neue Plattformversionen zur Berücksichtigung von `CVE-2019-11253` .	6. November 2019
Kubernetes`1.11` wird auf Amazon EKS nicht mehr unterstützt	Kubernetes-Version `1.11` wird auf Amazon EKS nicht mehr unterstützt. Bitte aktualisieren Sie alle `1.11`-Cluster auf Version `1.12` oder höher, um Service-Unterbrechungen zu vermeiden.	4. November 2019
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in der AWS-Region Südamerika (São Paulo) (`sa-east-1`) verfügbar.	16. Oktober 2019
Windows--Support	Amazon-EKS-Cluster, auf denen Kubernetes-Version `1.14` ausgeführt wird, unterstützen nun Windows-Workloads.	7. Oktober 2019
Auto Scaling	Es wurde ein Kapitel hinzugefügt, in dem einige der verschiedenen Arten von Kubernetes-Auto Scaling behandelt werden, die auf Amazon-EKS-Clustern unterstützt werden.	30. September 2019

Kubernetes-Dashboard-Update	Thema für die Installation des Kubernetes-Dashboards auf Amazon-EKS-Clustern zur Verwendung der Beta `2.0`-Version aktualisiert.	28. September 2019
Amazon EFS-CSI-Treiber	Thema für die Installation des CSI-Treibers für Amazon EFS in Amazon-EKS-Clustern für Kubernetes `1.14` hinzugefügt.	19. September 2019
Amazon EC2 Systems Manager-Parameter für Amazon-EKS-optimierte AMI-ID	Thema zum Abrufen der Amazon-EKS-optimierten AMI-ID mit einem Amazon-EC2-Systems-Manager-Parameter hinzugefügt. Mit dem Parameter müssen Sie keine AMI-IDs mehr abrufen.	18. September 2019
Amazon EKS-Ressourcenmarkierung	Sie können die Markierung Ihrer Amazon-EKS-Cluster verwalten.	16. September 2019
Amazon-EBS-CSI-Treiber	Thema für die Installation des Amazon-EBS-CSI-Treibers auf Kubernetes `1.14`-Amazon-EKS-Clustern hinzugefügt.	9. September 2019
Neues Amazon-EKS-optimiertes AMIs für `CVE-2019-9512` und `CVE-2019-9514` gepatcht	Amazon EKS hat das für Amazon-EKS-optimierte AMI aktualisiert, um `CVE-2019-9512` und `CVE-2019-9514` zu adressieren.	6. September 2019
Ankündigung der Beendigung der Unterstützung von Kubernetes`1.11` in Amazon EKS	Amazon EKS hat den Support für Kubernetes-Version `1.11` am 4. November 2019 eingestellt.	4. September 2019

Kubernetes-Version `1.14`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.14` hinzugefügt.	03. September 2019
IAM-Rollen für Servicekonten	Mit IAM-Rollen für Servicekonten auf Amazon-EKS-Clustern können Sie eine IAM-Rolle mit einem Kubernetes-Servicekonto verknüpfen. Mit diesem Feature müssen Sie der Knoten-IAM-Rolle keine erweiterten Berechtigungen mehr bereitstellen. Auf diese Weise können Pods auf diesem Knoten AWS APIs aufgerufen werden.	03. September 2019
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in der AWS-Region Naher Osten (Bahrain) (`me-south-1`) verfügbar.	29. August 2019
Aktualisierung der Amazon-EKS-Plattformversion	Neue Plattformversionen zur Berücksichtigung von `CVE-2019-9512` und `CVE-2019-9514`.	28. August 2019
Aktualisierung der Amazon-EKS-Plattformversion	Neue Plattformversionen zur Berücksichtigung von `CVE-2019-11247` und `CVE-2019-11249`.	5. August 2019
Erweiterung der Amazon-EKS-Region	Amazon EKS ist jetzt in der AWS-Region Asien-Pazifik (Hongkong) (`ap-east-1`) verfügbar.	31. Juli 2019

Kubernetes1.10 wird auf Amazon EKS nicht mehr unterstützt	Kubernetes-Version 1.10 wird auf Amazon EKS nicht mehr unterstützt. Aktualisieren Sie alle 1.10-Cluster auf Version 1.11 oder höher, um Service-Unterbrechungen zu vermeiden.	30. Juli 2019
Thema über ALB Ingress Controller hinzugefügt	Der AWS ALB Ingress Controller für Kubernetes ist ein Controller, der bewirkt, dass bei der Erstellung von Ingress-Ressourcen ein ALB erstellt wird.	11. Juli 2019
Neues Amazon-EKS-optimiertes AMIs	Entfernen unnötiger kubectl-Binärdateien von AMIs.	3. Juli 2019
Kubernetes-Version 1.13	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version 1.13 hinzugefügt.	18. Juni 2019
Neues Amazon-EKS-optimiertes AMIs für AWS-2019-005 gepatcht	Amazon EKS hat das für Amazon EKS optimierte AMI aktualisiert, um die in AWS-2019-005 beschriebenen Schwachstellen zu beheben.	17. Juni 2019
Ankündigung der Einstellung des Supports von Kubernetes1.10 in Amazon EKS	Amazon EKS hat am 22. Juli 2019 die Unterstützung von Kubernetes-Version 1.10 eingestellt.	21. Mai 2019

Aktualisierung der Amazon-EKS-Plattformversion	Neue Plattformversion für Kubernetes `1.11`- und `1.10`-Cluster zur Unterstützung von benutzerdefinierten DNS-Namen im `kubelet`-Zertifikat und zur Verbesserung der `etcd`-Leistung.	21. Mai 2019
AWS CLI`get-token` -Befehl	Der Befehl `aws eks get-token` wurde zur AWS CLI hinzugefügt. Sie müssen den AWS IAM Authenticator nicht mehr installieren, um Client-Sicherheitstoken für die Kubernetes Cluster-API-Serverkommunikation zu erstellen. Führen Sie ein Upgrade Ihrer AWS CLI Installation auf die neueste Version durch, um diese neue Funktionalität nutzen zu können. Weitere Informationen finden Sie unter Installieren der AWS Command Line Interface im AWS Command Line Interface -Leitfaden.	10. Mai 2019

Erste Schritte mit `eksctl`	In diesem Leitfaden für die ersten Schritte wird beschrieben, wie Sie alle erforderlichen Ressourcen installieren können, um mit Amazon EKS mit `eksctl` zu beginnen. Dies ist ein einfaches Befehlszeilendienstprogramm zum Erstellen und Verwalten von Kubernetes-Clustern auf Amazon EKS.	10. Mai 2019
Aktualisierung der Amazon-EKS-Plattformversion	Neue Plattformversion für Kubernetes `1.12`-Cluster zur Unterstützung von benutzerdefinierten DNS-Namen im `kubelet`-Zertifikat und zur Verbesserung der `etcd`-Leistung. Dadurch wird ein Fehler behoben, durch den bedingt `kubelet`-Daemons von Knoten alle paar Sekunden ein neues Zertifikat angefordert haben.	8. Mai 2019
Prometheus-Tutorial	Thema zum Bereitstellen von Prometheus in Ihrem Amazon EKS-Cluster hinzugefügt.	5. April 2019

Amazon-EKS-Steuerebenen-Protokollierung	Mit diesem Update können Sie Audit- und Diagnoseprotokolle direkt aus dem Amazon-EKS-Steuerungsbereich abrufen. Sie können diese CloudWatch Protokolle in Ihrem Konto als Referenz für die Sicherung und Ausführung von Clustern verwenden.	4. April 2019
Kubernetes-Version `1.12`	Unterstützung für neue Cluster und Versionsupgrades aus Kubernetes-Version `1.12` hinzugefügt.	28. März 2019
Einführungsleitfaden für App Mesh hinzugefügt	Dokumentation für den Einstieg mit App Mesh und Kubernetes hinzugefügt.	27. März 2019
Privater Amazon-EKS-API-Server-Endpunktzugriff	Dokumentation zur Deaktivierung des öffentlichen Zugriffs für den Kubernetes-API-Server-Endpunkt Ihres Amazon-EKS-Clusters hinzugefügt.	19. März 2019
Thema zur Installation des Kubernetes Metrics Servers hinzugefügt.	Der Kubernetes-Metrik-Server ist ein Aggregator für Ressourcenverbrauchsdaten in Ihrem Cluster.	18. März 2019
Liste der zugehörigen Open-Source-Projekte hinzugefügt.	Diese Open-Source-Projekte erweitern die Funktionalität von Kubernetes Clustern AWS, auf denen ausgeführt wird, einschließlich Clustern, die von Amazon EKS verwaltet werden.	15. März 2019

Thema zur lokalen Installation von Helm hinzugefügt.	Der `helm`-Paketmanager für Kubernetes unterstützt Sie bei der Installation und Verwaltung von Anwendungen in Ihrem Kubernetes-Cluster. In diesem Thema wird gezeigt, wie die `helm`- und `tiller`-Binärdateien lokal installiert und ausgeführt werden. Auf diese Weise können Sie Diagramme mit der Helm-CLI auf Ihrem lokalen System installieren und verwalten.	11. März 2019
Aktualisierung der Amazon-EKS-Plattformversion	Neue Plattformversion, die Amazon EKS Kubernetes `1.11`-Cluster auf Patch-Level `1.11.8` aktualisiert, um `CVE-2019-1002100` zu adressieren.	8. März 2019
Erhöhtes Cluster-Limit	Amazon EKS hat die Anzahl der in einer AWS-Region erstellbaren Cluster von 3 auf 50 erhöht.	13. Februar 2019
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in Europa (London) (), Europa (Pariseu-west-2) (`eu-west-3`) und im asiatisch-pazifischen Raum (Mumbai) (`ap-south-1`) verfügbar AWS-Regionen.	13. Februar 2019

Neues Amazon-EKS-optimiertes AMIs für ALAS-2019-1156 gepatcht	Amazon EKS hat das für Amazon EKS optimierte AMI aktualisiert, um die in ALAS-2019-1156 beschriebene Schwachstelle zu beheben.	11. Februar 2019
Neues Amazon-EKS-optimiertes AMIs für ALAS2-2019-1141 gepatcht	Amazon EKS hat das für Amazon EKS optimierte AMI aktualisiert, um die CVEs zu berücksichtigen, auf die in ALAS2-2019-1141 verwiesen wird.	9. Januar 2019
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt im asiatisch-pazifischen Raum (Seoul) verfügbar (ap-northeast-2) AWS-Region.	9. Januar 2019
Erweiterung der Amazon-EKS-Region	Amazon EKS ist jetzt zusätzlich in den folgenden Ländern verfügbar AWS-Regionen: Europa (Frankfurt) (eu-central-1), Asien-Pazifik (Tokio) (ap-northeast-1), Asien-Pazifik (Singapur) (ap-southeast-1) und Asien-Pazifik (Sydney) (ap-southeast-2).	19. Dezember 2018
Amazon EKS-Cluster-Aktualisierungen	Dokumentation für Amazon-EKS-Cluster Kubernetes-Versionsupdates und Knotenaustausch hinzugefügt.	12. Dezember 2018

Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in der AWS-Region Europa (Stockholm) (`eu-north-1`) verfügbar.	11. Dezember 2018
Aktualisierung der Amazon-EKS-Plattformversion	Neue Plattformversion, bei der Kubernetes auf Patch-Level `1.10.11` aktualisiert wird, um `CVE-2018-1002105` zu beheben.	4. Dezember 2018
Support für Version`1.0.0` für den ALB-Controller für eingehenden Datenverkehr hinzugefügt	Der ALB Ingress Controller veröffentlicht eine Version `1.0.0` mit formeller Unterstützung von. AWS	20. November 2018
Unterstützung der CNI-Netzwerkkonfiguration hinzugefügt	Die Amazon VPC CNI plugin for Kubernetes-Version `1.2.1` unterstützt jetzt eine benutzerdefinierte Netzwerkkonfiguration für Netzwerkschnittstellen des sekundären Pod.	16. Oktober 2018
Unterstützung für `MutatingAdmissionWebhook` und `ValidatingAdmissionWebhook` hinzugefügt	Die Amazon-EKS-Plattformversion `1.10-eks.2` unterstützt jetzt die Zulassungscontroller `MutatingAdmissionWebhook` und `ValidatingAdmissionWebhook` .	10. Oktober 2018
Partner-AMI-Informationen hinzugefügt.	Canonical hat in enger Zusammenarbeit mit Amazon EKS Knoten-AMIs erstellt, die Sie in Ihren Clustern verwenden können.	3. Oktober 2018

Ergänzung von Anweisungen für AWS CLIupdate-kubeconfig -Befehl	Amazon EKS hat das hinzugefügtupdate-kubeconfig , AWS CLI um das Erstellen einer kubeconfig Datei für den Zugriff auf Ihren Cluster zu vereinfachen.	21. September 2018
Neues Amazon-EKS-optimiertes AMI	Amazon EKS hat die Amazon-EKS-optimierten AMIs (mit und ohne GPU-Unterstützung) mit verschiedenen Sicherheitsupdates und AMI-Optimierungen aktualisiert.	13. September 2018
Amazon AWS-Region EKS-Erweiterung	Amazon EKS ist jetzt in der Region Europa (Irland) (eu-west-1) verfügbar.	5. September 2018
Aktualisierung der Amazon-EKS-Plattformversion	Neue Plattformversion mit Unterstützung für die Kubernetes-Aggregationsebene und den Horizontal Pod Autoscaler(HPA).	31. August 2018
Neue Amazon-EKS-optimierte AMIs und GPU-Unterstützung	Amazon EKS hat das für Amazon EKS optimierte AMI aktualisiert, sodass es eine neue AWS CloudFormation Knotenvorlage und ein neues Bootstrap-Skript verwendet . Außerdem ist ein neues Amazon-EKS-optimiertes AMI mit GPU-Unterstützung verfügbar.	22. August 2018

Neues Amazon-EKS-optimiertes AMIs für ALAS2-2018-1058 gepatcht	Amazon EKS hat das für Amazon EKS optimierte AMI aktualisiert, um die CVEs zu berücksichtigen, auf die in ALAS2-2018-1058 verwiesen wird.	14. August 2018
Amazon-EKS-optimierte AMI-Build-Skripts	Amazon EKS hat die Build-Skripts, die zum Erstellen des Amazon-EKS-optimierten AMI verwendet werden, als Open-Source-Software zur Verfügung gestellt. Diese Build-Skripts sind ab sofort auf GitHub verfügbar.	10. Juli 2018
Erstveröffentlichung von Amazon EKS	Erste Dokumentation für den Servicestart	5. Juni 2018

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.