Benutzerhandbuch

AWS Telco Network Builder

AWS Telco Network Builder: Benutzerhandbuch

Table of Contents

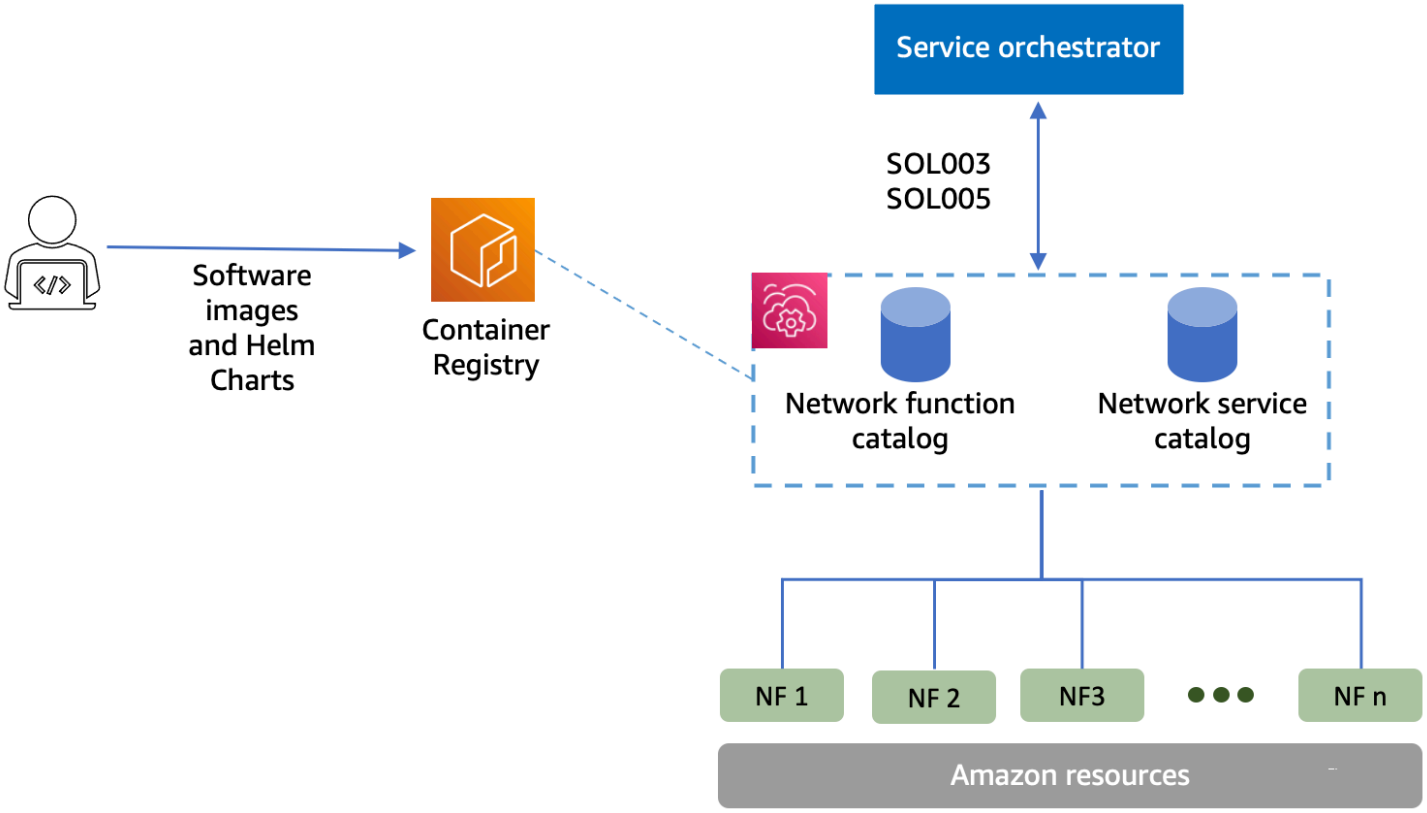
Was istAWS Telco Network Builder?

AWSTelco Network Builder (AWSTNB) ist einAWS Dienst, der Kommunikationsdienstanbietern (CSPs) eine effiziente Möglichkeit bietet, 5G-Netzwerke in derAWS Infrastruktur bereitzustellen, zu verwalten und zu skalieren.

MitAWS TNB stellen Sie skalierbare und sichere 5G-Netzwerke bereit,AWS Cloud indem Sie containerisierte Software-Images automatisiert verwenden. Sie müssen keine neuen Technologien erlernen, entscheiden, welchen Compute-Service Sie verwenden möchten, oder wissen, wie Sie bereitstellen und konfigurierenAWSRessourcen schätzen.

Stattdessen beschreiben Sie die Infrastruktur Ihres Netzwerks und stellen die Software-Images der Netzwerkfunktionen von Ihren unabhängigen Softwareanbieter-Partnern (ISV) zur Verfügung. AWS TNB lässt sich in Service-Orchestratoren undAWS Dienste von Drittanbietern integrieren, um automatisch die erforderlicheAWS Infrastruktur bereitzustellen, containerisierte Netzwerkfunktionen bereitzustellen und das Netzwerk- und Zugriffsmanagement zu konfigurieren, um einen voll funktionsfähigen Netzwerkdienst zu schaffen.

Das folgende Diagramm veranschaulicht die logischen Integrationen zwischenAWS TNB und Service Orchestrator zur Bereitstellung von Netzwerkfunktionen mithilfe von Standardschnittstellen, die auf dem European Telecommunications Standards Institute (ETSI) basieren.



Themen

Neu bei AWS?

Wenn Sie mit AWS-Produkten und -Dienstleistungen noch nicht vertraut sind, beginnen Sie mit den folgenden Ressourcen, um mehr darüber zu erfahren:

- Einführung in AWS
- Erste Schritte mit AWS

Für wen istAWS TNB?

AWSTNB richtet sich an CSPs, die die Vorteile der Kosteneffizienz, Agilität und Elastizität nutzen möchten, ohne benutzerdefinierte Skripts und Konfigurationen für denAWS Cloud Entwurf, die Bereitstellung und Verwaltung von Netzwerkdiensten schreiben und verwalten zu müssen. AWS TNB stellt automatisch die erforderlicheAWS Infrastruktur bereit, stellt containerisierte Netzwerkfunktionen bereit und konfiguriert das Netzwerk- und Zugriffsmanagement, um voll funktionsfähige Netzwerkdienste auf der Grundlage der vom CSP definierten Netzwerkdienstdeskriptoren und der Netzwerkfunktionen, die der CSP bereitstellen möchte, zu erstellen.

WarumAWS TNB verwenden?

Im Folgenden sind einige der Gründe aufgeführt, warum ein CSPAWS TNB verwenden möchte:

Hilft, Aufgaben zu vereinfachen

Sorgen Sie für mehr Effizienz Ihres Netzwerkbetriebs, z. B. durch die Bereitstellung neuer Dienste, die Aktualisierung und Aktualisierung von Netzwerkfunktionen und die Änderung der Netzwerkinfrastrukturtopologien.

Lässt sich in Orchestratoren integrieren

AWSTNB lässt sich in beliebte Service-Orchestratoren von Drittanbietern integrieren, die ETSI-konform sind.

Waagen

Sie könnenAWS TNB so konfigurieren, dass die zugrunde liegendenAWS Ressourcen skaliert werden, um den Datenverkehrsanforderungen gerecht zu werden, Netzwerkfunktionsupdates effizienter durchzuführen, Änderungen der Netzwerkinfrastruktur einzuführen und die Bereitstellungszeit neuer 5G-Dienste von Tagen auf Stunden zu reduzieren.

Inspiziert und überwachtAWS Ressourcen

AWSMit TNB können Sie dieAWS Ressourcen, die Ihr Netzwerk unterstützen, auf einem einzigen Dashboard überprüfen und überwachen, z. B. Amazon VPC, Amazon EC2 und Amazon EKS.

Unterstützt Service-Vorlagen

AWSMit TNB können Sie Servicevorlagen für alle Telekommunikations-Workloads (RAN, Core, IMS) erstellen. Sie können eine neue Servicedefinition erstellen, eine vorhandene Vorlage

wiederverwenden oder sie in eine CI/CD-Pipeline (Continuous Integration and Continuous Delivery) integrieren, um eine neue Definition zu veröffentlichen.

Verfolgt Änderungen an Netzwerkbereitstellungen

Wenn Sie die zugrunde liegende Konfiguration einer Netzwerkfunktionsbereitstellung ändern, z. B. indem Sie den Instanztyp eines Amazon EC2-Instanztyps ändern, können Sie die Änderungen auf wiederholbare und skalierbare Weise verfolgen. Um dies manuell zu tun, müssten Sie den Status des Netzwerks verwalten, Ressourcen erstellen und löschen und auf die Reihenfolge der erforderlichen Änderungen achten. Wenn SieAWS TNB verwenden, um den Lebenszyklus Ihrer Netzwerkfunktion zu verwalten, nehmen Sie nur die Änderungen an Ihren Netzwerkdienstdeskriptoren vor, die die Netzwerkfunktion beschreiben. AWS TNB nimmt dann automatisch die erforderlichen Änderungen in der richtigen Reihenfolge vor.

Vereinfacht den Lebenszyklus von Netzwerkfunktionen

Sie können die erste und alle nachfolgenden Versionen einer Netzwerkfunktion verwalten und angeben, wann ein Upgrade durchgeführt werden soll. Auf die gleiche Weise können Sie auch Ihre RAN-, Core-, IMS- und Netzwerkanwendungen verwalten.

Zugriff aufAWS TNB

Sie können die folgenden Schnittstellen verwenden, um IhreAWS TNB-Ressourcen zu erstellen, auf sie zuzugreifen und sie zu verwalten:

- AWSTNB-Konsole — Bietet eine Weboberfläche für die Verwaltung Ihres Netzwerks.
- AWSTNB-API — Stellt eine RESTful-API für die Ausführung vonAWS TNB-Aktionen bereit. Weitere Informationen finden Sie in der AWSTNB-API-Referenz
- AWS Command Line Interface(AWS CLI) — Stellt Befehle für eine Vielzahl vonAWS Diensten bereit, einschließlichAWS TNB. Sie wird unter Windows, macOS und Linux unterstützt. Weitere Informationen finden Sie unter AWS Command Line Interface.
- AWS-SDKs — Stellt sprachspezifische APIs bereit und vervollständigt viele der Verbindungsdetails. Dazu gehören das Berechnen von Signaturen, das Behandeln von Wiederholungsversuchen und das Behandeln von Fehlern. Weitere Informationen finden Sie unter AWS SDKs.

Preisgestaltung fürAWS TNB

AWSTNB hilft CSPs dabei, den Einsatz und die Verwaltung ihrer Telekommunikationsnetzwerke zu automatisierenAWS. Sie zahlen für die folgenden zwei Dimensionen, wenn SieAWS TNB verwenden:

- Nach Stunden des verwalteten Netzwerkfunktionselements (MNFI).
- Nach Anzahl der API-Anfragen.

Es fallen auch zusätzliche Gebühren an, wenn Sie andereAWS Dienste in Verbindung mitAWS TNB nutzen. Weitere Informationen finden Sie unter AWSTNB-Preise.

Um Ihre Rechnung anzuzeigen, navigieren Sie zum Fakturierungs- und Kostenverwaltungs-Dashboard in der AWS Billing and Cost Management-Konsole. Ihre Abrechnung enthält Links zu Nutzungsberichten mit zusätzlichen Details zu Ihrer Abrechnung. Weitere Informationen zurAWS Kontoabrechnung finden Sie unter AWSKontoabrechnung.

Wenden Sie sich bei Fragen zu AWS-Abrechnungen, -Konten und -Vorfällen an AWS Support.

AWS Trusted Advisorist ein Service, mit dem Sie den Preis, die Sicherheit und die Leistung IhrerAWS -Umgebung optimieren können. Weitere Informationen finden Sie unter AWSTrusted Advisor.

Was ist?

Weitere Informationen zu den ersten Schritten mitAWS TNB finden Sie in den folgenden Themen:

- AWS TNB einrichten— Führen Sie die erforderlichen Schritte aus.
- Erste Schritte mit AWS TNB— Stellen Sie Ihre erste Netzwerkfunktion bereit, z. B. eine zentrale Einheit (CU), eine Access and Mobility Management Function (AMF), eine Benutzerplane-Funktion (UPF) oder einen kompletten 5G-Kern.

So funktioniertAWS TNB

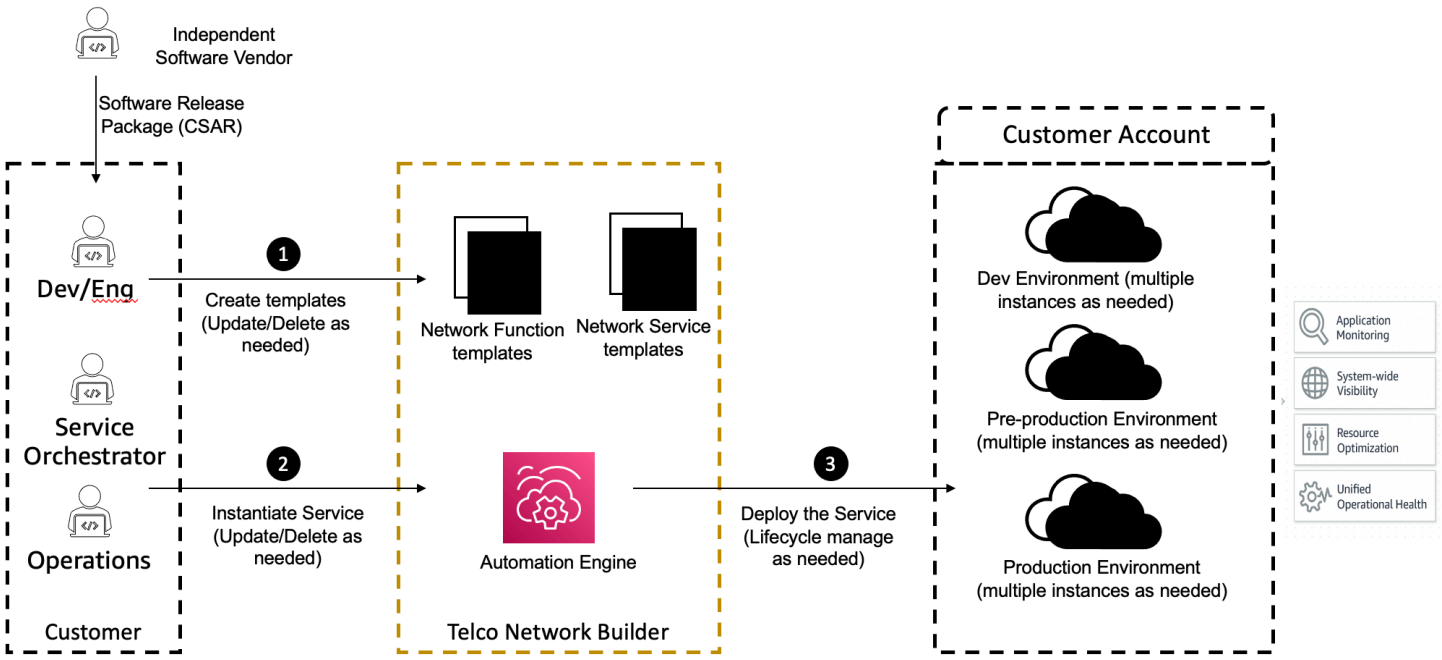
AWSTNB lässt sich in standardisierte end-to-end Orchestratoren undAWS Ressourcen integrieren, um vollständige 5G-Netzwerke zu betreiben.

AWSTNB ermöglicht Ihnen die Aufnahme von Netzwerkfunktionspaketen und Netzwerkdienstdeskriptoren (NSDs) und stellt Ihnen die Automatisierungs-Engine für den Betrieb Ihrer Netzwerke zur Verfügung. Sie können Ihren end-to-end Orchestrator verwenden und inAWS TNB-APIs integrieren oderAWS TNB-SDKs verwenden, um Ihren eigenen Automatisierungsablauf zu erstellen. Weitere Informationen finden Sie unter AWSTNB-Architektur.

Themen

AWSTNB-Architektur

AWSTNB bietet Ihnen die Möglichkeit, Lebenszyklusmanagement-Operationen über dieAWS Management ConsoleAWS CLI,AWS TNB-REST-API und SDKs durchzuführen. Auf diese Weise können die verschiedenen CSP-Personas, z. B. Mitglieder der Teams Engineering, Operations und Programmatic System, die Vorteile vonAWS TNB nutzen. Sie erstellen ein Netzwerkfunktionspaket und laden es als Cloud Service Archive (CSAR) -Datei hoch. Die CSAR-Datei enthält Helm-Diagramme, Software-Images und einen Network Function Descriptor (NFD). Sie können Vorlagen verwenden, um mehrere Konfigurationen dieses Pakets wiederholt bereitzustellen. Sie erstellen Netzwerkdienstvorlagen, die die Infrastruktur und die Netzwerkfunktionen definieren, die Sie bereitstellen möchten. Sie können Parameter-Overrides verwenden, um verschiedene Konfigurationen an verschiedenen Orten bereitzustellen. Anschließend können Sie mithilfe der Vorlagen ein Netzwerk instanziieren und Ihre Netzwerkfunktionen auf derAWS Infrastruktur bereitstellen. AWS TNB bietet Ihnen den Überblick über Ihre Implementierungen.



Integration in AWS-Services

Ein 5G-Netzwerk besteht aus einer Reihe miteinander verbundener containerisierter Netzwerkfunktionen, die in Tausenden von Kubernetes-Clustern eingesetzt werden. AWS TNB lässt sich in die folgendenAWS-Services telekommunikationsspezifischen APIs integrieren, um einen voll funktionsfähigen Netzwerkdienst zu schaffen:

- Amazon Elastic Container Registry (Amazon ECR) zum Speichern von Netzwerkfunktions-Artefakten
- Amazon Elastic Kubernetes Service(Amazon EKS) zum Einrichten von Clustern.
- Amazon VPC für Netzwerkkonstrukte.
- Sicherheitsgruppen verwendenAWS CloudFormation.
- AWS CodePipelinefür Einsatzziele in allenAWS-Regionen,AWS Local Zones undAWS Outposts.
- IAM, um Rollen zu definieren.
- AWS Organizationsum den Zugriff aufAWS TNB-APIs zu kontrollieren.
- AWS Health DashboardundAWS CloudTrail um den Gesundheitszustand zu überwachen und Kennzahlen zu veröffentlichen.

AWSTNB-Ressourcenquoten

IhrAWS-Konto verfügt über StandardKontingent, die früher als Grenzwerte bezeichnet wurdenAWS-Service. Sofern nicht anders angegeben, gilt jedes Kontingent spezifischAWS-Region Sie können Erhöhungen für einige Konting

Um die Kontingente fürAWS TNB anzuzeigen, öffnen Sie die Service-Quotas-Konsole. Wählen AWS-ServicesSie im Navigationsbereich AWSTNB aus und wählen Sie

Informationen zur Erhöhung eines Kontingents finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

IhrAWS-Konto umfasst die folgenden KontingenteAWS

Kon	Beschreibung	Standardwert	Anpassbar?
Netzwerkdienstinstanzen	Die maximale Anzahl von NetzwerkDienst-Instanzen in einer Region.	800	Ja
Gleichzeitiger laufender Netzwerkdienstbetrieb	Die maximale Anzahl der gleichzeitigen laufenden NetzwerkDienstvorfälle in einer Region.	40	Ja
Netzwerkpakete	Die maximale Anzahl von Netzwerkpaketen in einer Region.	40	Ja
Funktionspakete	Die maximale Anzahl der Funktionpaketen in einer Region.	200	Ja

AWS TNB-Konzepte

In diesem Thema werden grundlegende Konzepte beschrieben, die Ihnen den Einstieg in die Verwendung von AWS TNB erleichtern sollen.

Inhalt

Lebenszyklus einer Netzwerkfunktion

AWS TNB unterstützt Sie während des gesamten Lebenszyklus Ihrer Netzwerkfunktionen. Der Lebenszyklus von Netzwerkfunktionen umfasst die folgenden Phasen und Aktivitäten:

Planung

1. Planen Sie Ihr Netzwerk, indem Sie die bereitzustellenden Netzwerkfunktionen identifizieren.
2. Speichern Sie die Software-Images für Netzwerkfunktionen in einem Container-Image-Repository.
3. Erstellen Sie die CSAR-Pakete, die bereitgestellt oder aktualisiert werden sollen.
4. Verwenden Sie AWS TNB, um das CSAR-Paket hochzuladen, das Ihre Netzwerkfunktion definiert (z. B. CU AMF und UPF), und integrieren Sie es in eine CI/CD-Pipeline (Continuous Integration and Continuous Delivery), die Ihnen helfen kann, neue Versionen Ihres CSAR-Pakets zu erstellen, sobald neue Netzwerkfunktions-Softwareimages oder Kundenskripte verfügbar sind.

Konfiguration

1. Identifizieren Sie die für die Bereitstellung erforderlichen Informationen, z. B. den Berechnungstyp, die Version der Netzwerkfunktion, IP-Informationen und Namen der Ressourcen.
2. Verwenden Sie diese Informationen, um Ihren Network Service Descriptor (NSD) zu erstellen.

3. Erfassen Sie NSDs, die Ihre Netzwerkfunktionen und die Ressourcen definieren, die für die Instanziierung der Netzwerkfunktion erforderlich sind.

Instanziierung

1. Erstellen Sie die Infrastruktur, die für die Netzwerkfunktionen erforderlich ist.
2. Instanziieren (oder stellen Sie sie bereit), wie in ihrer NSD definiert, und beginnen Sie mit der Übertragung des Datenverkehrs.
3. Validieren Sie die Ressourcen.

Produktion

Während des Lebenszyklus der Netzwerkfunktion werden Sie Produktionsvorgänge abschließen, wie z. B.:

- Aktualisieren Sie die Konfiguration der Netzwerkfunktion, indem Sie beispielsweise einen Wert in der bereitgestellten Netzwerkfunktion aktualisieren.
- Ersetzen Sie die Netzwerkfunktion oder nehmen Sie sie außer Betrieb.

Verwenden Sie standardisierte Schnittstellen

AWS TNB lässt sich in ETSI-konforme Service-Orchestratoren integrieren, sodass Sie die Bereitstellung Ihrer Netzwerkdienste vereinfachen können. Service-Orchestratoren können AWS TNB-SDKs, die CLI oder die APIs verwenden, um Operationen wie die Instanziierung oder Aktualisierung einer Netzwerkfunktion auf eine neue Version zu initiieren.

AWS TNB unterstützt die folgenden Spezifikationen.

Spezifikation	Veröffentlichung	Beschreibung
ETSI SOL001	v3.6.1	Definiert Standards für die Zulassung von TOSCA-basierten Netzwerkfunktionsdeskriptoren.
ETSI SOL002	v3.6.1	Definiert Modelle rund um das Netzwerkfunktionsmanagement.
ETSI SOL003	v3.6.1	Definiert Standards für das Lebenszyklusmanagement von Netzwerkfunktionen.

Spezifikation	Veröffentlichung	Beschreibung
ETSI SOL004	v3.6.1	Definiert CSAR-Standards für Netzwerkf unktionspakete.
ETSI SOL005	v3.6.1	Definiert Standards für das Netzwerk-Servicepa ket und das Lebenszyklusmanagement von Netzwerkdiensten.
ETSI SOL007	v3.5.1	Definiert Standards für die Zulassung von TOSCA-basierten Netzwerkdienstdeskriptoren.

Netzwerk-Funktionspakete für TNB AWS

Mit AWS TNB können Sie Netzwerkfunktionspakete, die ETSI SOL001/SOL004 entsprechen, in einem Funktionskatalog speichern. Anschließend können Sie Cloud Service Archive (CSAR) -Pakete hochladen, die Artefakte enthalten, die Ihre Netzwerkfunktion beschreiben.

- Deskriptor für Netzwerkfunktionen — Definiert Metadaten für das Onboarding von Paketen und die Verwaltung von Netzwerkfunktionen
- Software-Images — Verweist auf Container-Images für Netzwerkfunktionen. Amazon Elastic Container Registry (Amazon ECR) kann als Repository für Netzwerkfunktions-Images dienen.
- Zusätzliche Dateien — werden zur Verwaltung der Netzwerkfunktion verwendet, z. B. für Skripts und Helm-Diagramme.

Das CSAR ist ein Paket, das durch den OASIS-TOSCA-Standard definiert ist und einen Netzwerk-/Dienstdeskriptor enthält, der der OASIS TOSCA YAML-Spezifikation entspricht. Hinweise zur erforderlichen YAML-Spezifikation finden Sie unter. TOSCA Referenz fürAWS TNB

Im Folgenden finden Sie ein Beispiel für einen Deskriptor für Netzwerkfunktionen.

```
tosca_definitions_version: tnb_simple_yaml_1_0

topology_template:

  node_templates:

    SampleNF:
```

```
      type: tosca.nodes.AWS.VNF
      properties:
        descriptor_id: "SampleNF-descriptor-id"
        descriptor_version: "2.0.0"
        descriptor_name: "NF 1.0.0"
        provider: "SampleNF"
      requirements:
        helm: HelmChart

    HelmChart:
      type: tosca.nodes.AWS.Artifacts.Helm
      properties:
        implementation: "./SampleNF"
```

Dienstdeskriptoren für Netzwerkfunktionen für TNB AWS

AWS TNB speichert Netzwerkdienstbeschreibungen (NSDs) über die Netzwerkfunktionen, die Sie bereitstellen möchten, und darüber, wie Sie sie bereitstellen möchten, im Katalog. Sie können Ihre YAML-NSD-Datei, wie von ETSI SOL007 beschrieben, hochladen und Folgendes enthalten:

- NF, die Sie bereitstellen möchten
- Anweisungen zum Netzwerk
- Anweisungen berechnen
- Lifecycle-Hooks (benutzerdefinierte Skripte)

AWS TNB unterstützt ETSI-Standards für die Modellierung von Ressourcen wie Netzwerk, Dienst und Funktion in der TOSCA-Sprache. AWS TNB macht die Nutzung für Sie effizienter, AWS-Services indem es sie so modelliert, dass Ihr ETSI-konformer Service Orchestrator sie verstehen kann.

Im Folgenden finden Sie einen Ausschnitt aus einem NSD, der zeigt, wie man modelliert. AWS-Services Die Netzwerkfunktion wird auf einem Amazon EKS-Cluster mit Kubernetes Version 1.27 bereitgestellt. Die Subnetze für die Anwendungen sind Subnet01 und Subnet02. Anschließend können Sie das NodeGroups für Ihre Anwendungen mit einem Amazon Machine Image (AMI), einem Instance-Typ und einer Autoscaling-Konfiguration definieren.

```
tosca_definitions_version: tnb_simple_yaml_1_0

SampleNFEKS:
  type: tosca.nodes.AWS.Compute.EKS
```

```
  properties:
    version: "1.27"
    access: "ALL"
    cluster_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleClusterRole"
  capabilities:
    multus:
      properties:
        enabled: true
  requirements:
    subnets:
      - Subnet01
      - Subnet02

SampleNFEKSNode01:
  type: tosca.nodes.AWS.Compute.EKSManagedNode
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleNodeRole"
  capabilities:
    compute:
      properties:
        ami_type: "AL2_x86_64"
        instance_types:
          - "t3.xlarge"
        key_pair: "SampleKeyPair"
    scaling:
      properties:
        desired_size: 3
        min_size: 2
        max_size: 6
  requirements:
    cluster: SampleNFEKS
    subnets:
      - Subnet01
    network_interfaces:
      - ENI01
      - ENI02
```

Verwaltung und Betrieb für TNB AWS

Mit AWS TNB können Sie Ihr Netzwerk mithilfe standardisierter Verwaltungsoperationen gemäß ETSI SOL003 und SOL005 verwalten. Sie können die AWS TNB-APIs verwenden, um Lebenszyklusvorgänge durchzuführen, wie z. B.:

- Instanziieren Ihrer Netzwerkfunktionen.
- Beenden Ihrer Netzwerkfunktionen.
- Aktualisierung Ihrer Netzwerkfunktionen, um Helm-Bereitstellungen außer Kraft zu setzen.
- Verwaltung von Versionen Ihrer Netzwerkfunktionspakete.
- Versionen Ihrer NSDs verwalten.
- Abrufen von Informationen über Ihre bereitgestellten Netzwerkfunktionen.

Netzwerkdienstdeskriptoren für TNB AWS

Ein Network Service Descriptor (NSD) ist eine `.yaml` Datei in einem Netzwerkpaket, die den TOSCA-Standard verwendet, um die Netzwerkfunktionen, die Sie bereitstellen möchten, und die AWS Infrastruktur, auf der Sie die Netzwerkfunktionen bereitstellen möchten, zu beschreiben. Um Ihre NSD zu definieren und Ihre zugrunde liegenden Ressourcen und Netzwerklebenszyklusoperationen zu konfigurieren, müssen Sie das von TNB unterstützte NSD-TOSCA-Schema verstehen. AWS

Ihre NSD-Datei ist in die folgenden Teile unterteilt:

1. TOSCA-Definitionsversion — Dies ist die erste Zeile Ihrer NSD-YAML-Datei und enthält die Versionsinformationen, wie im folgenden Beispiel gezeigt.

   ```
   tosca_definitions_version: tnb_simple_yaml_1_0
   ```

2. VNFD — Die NSD enthält die Definition der Netzwerkfunktion, auf der Lebenszyklusoperationen ausgeführt werden sollen. Jede Netzwerkfunktion muss anhand der folgenden Werte identifiziert werden:

   - Eine eindeutige ID für`descriptor_id`. Die ID muss mit der ID im CSAR-Paket der Netzwerkfunktion übereinstimmen.
   - Ein eindeutiger Name für`namespace`. Der Name muss mit einer eindeutigen ID verknüpft werden, damit er in Ihrer NSD-YAML-Datei leichter referenziert werden kann, wie im folgenden Beispiel gezeigt.

   ```
   vnfds:
     - descriptor_id: "61465757-cb8f-44d8-92c2-b69ca0de025b"
       namespace: "amf"
   ```

3. Topologievorlage — Definiert die bereitzustellenden Ressourcen, die Bereitstellung von Netzwerkfunktionen und alle benutzerdefinierten Skripts, wie z. B. Lifecycle-Hooks. Dies wird im folgenden Beispiel veranschaulicht.

```
topology_template:

  node_templates:

    SampleNS:
      type: tosca.nodes.AWS.NS
      properties:
        descriptor_id: "<Sample Identifier>"
        descriptor_version: "<Sample nversion>"
        descriptor_name: "<Sample name>"
```

4. Zusätzliche Knoten — Jede modellierte Ressource hat Abschnitte für Eigenschaften und Anforderungen. Die Eigenschaften beschreiben optionale oder obligatorische Attribute für eine Ressource, z. B. die Version. Die Anforderungen beschreiben Abhängigkeiten, die als Argumente angegeben werden müssen. Um beispielsweise eine Amazon EKS Node Group-Ressource zu erstellen, muss sie innerhalb eines Amazon EKS-Clusters erstellt werden. Dies wird im folgenden Beispiel veranschaulicht.

```
SampleEKSNode:
  type: tosca.nodes.AWS.Compute.EKSManagedNode
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
  capabilities:
    compute:
      properties:
        ami_type: "AL2_x86_64"
        instance_types:
          - "t3.xlarge"
        key_pair: "SampleKeyPair"
    scaling:
      properties:
        desired_size: 1
        min_size: 1
        max_size: 1
  requirements:
    cluster: SampleEKS
    subnets:
      - SampleSubnet
```

```
    network_interfaces:
      - SampleENI01
      - SampleENI02
```

AWS TNB einrichten

Richten Sie AWS TNB ein, indem Sie die in diesem Thema beschriebenen Aufgaben ausführen.

Aufgaben

- Melden Sie sich an für AWS
- Wählen Sie eine AWS Region
- Notieren Sie sich den Service-Endpunkt
- (Optional) Installieren Sie AWS CLI
- Erstellen eines IAM-Benutzers
- Richten Sie AWS TNB-Rollen ein

Melden Sie sich an für AWS

Wenn Sie sich für Amazon Web Services registrieren, AWS-Konto sind Sie automatisch für alle Dienste angemeldet AWS, einschließlich AWS TNB. Berechnet werden Ihnen aber nur die Services, die Sie nutzen.

Wenn Sie AWS-Konto bereits eine haben, fahren Sie mit der nächsten Aufgabe fort. Wenn Sie kein AWS-Konto haben, führen Sie die folgenden Schritte zum Erstellen eines Kontos aus.

Um eine zu erstellen AWS-Konto

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

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

Wählen Sie eine AWS Region

Eine Liste der verfügbaren Regionen für AWS TNB finden Sie in der Liste der AWS regionalen Dienste. Eine Liste der Endpunkte für den programmatischen Zugriff finden Sie unter AWS TNB-Endpunkte in der. Allgemeine AWS-Referenz

Notieren Sie sich den Service-Endpunkt

Um programmgesteuert eine Verbindung zu einem AWS Dienst herzustellen, verwenden Sie einen Endpunkt. Zusätzlich zu den AWS Standardendpunkten bieten einige AWS Dienste FIPS-Endpunkte in ausgewählten Regionen. Weitere Informationen finden Sie unter AWS -Service-Endpunkte.

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
USA Ost (Nord-Virginia)	us-east-1	tnb.us-east-1.amazonaws.com	HTTPS	
USA West (Oregon)	us-west-2	tnb.us-west-2.amazonaws.com	HTTPS	
Asien-Pazifik (Seoul)	ap-northeast-2	tnb.ap-northeast-2.amazonaws.com	HTTPS	
Asien-Pazifik (Sydney)	ap-southeast-2	tnb.ap-southeast-2.amazonaws.com	HTTPS	
Kanada (Zentral)	ca-central-1	tnb.ca-central-1.amazonaws.com	HTTPS	
Europa (Frankfurt)	eu-central-1	tnb.eu-central-1.amazonaws.com	HTTPS	

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
Europa (Paris)	eu-west-3	tnb.eu-west-3.amazonaws.com	HTTPS	
Europa (Spanien)	eu-south-2	tnb.eu-south-2.amazonaws.com	HTTPS	
Europa (Stockholm)	eu-north-1	tnb.eu-north-1.amazonaws.com	HTTPS	
Südamerika (São Paulo)	sa-east-1	tnb.sa-east-1.amazonaws.com	HTTPS	

(Optional) Installieren Sie AWS CLI

Die AWS Command Line Interface (AWS CLI) bietet Befehle für eine Vielzahl von AWS Produkten und wird unter Windows, MacOS und Linux unterstützt. Sie können mit dem auf AWS TNB zugreifen. AWS CLI Informationen zu den ersten Schritten finden Sie im AWS Command Line Interface - Benutzerhandbuch. Weitere Informationen zu den Befehlen für AWS TNB finden Sie unter tnb in der AWS CLI Befehlsreferenz.

Erstellen eines IAM-Benutzers

AWS Identity and Access Management (IAM) ist ein Webdienst, mit dem Sie den Zugriff auf Ressourcen sicher kontrollieren können. AWS Erstellen Sie eine IAM-Benutzerrolle, um kurzfristige Anmeldeinformationen für den Zugriff zu verwenden. AWS

Folgen Sie den Anweisungen unter Erste Schritte im AWS IAM Identity Center Benutzerhandbuch, um die Rolle zu erstellen.

Sie können den programmatischen Zugriff auch konfigurieren, indem Sie den AWS CLI zu AWS IAM Identity Center verwendenden im AWS Command Line Interface Benutzerhandbuch konfigurieren.

Richten Sie AWS TNB-Rollen ein

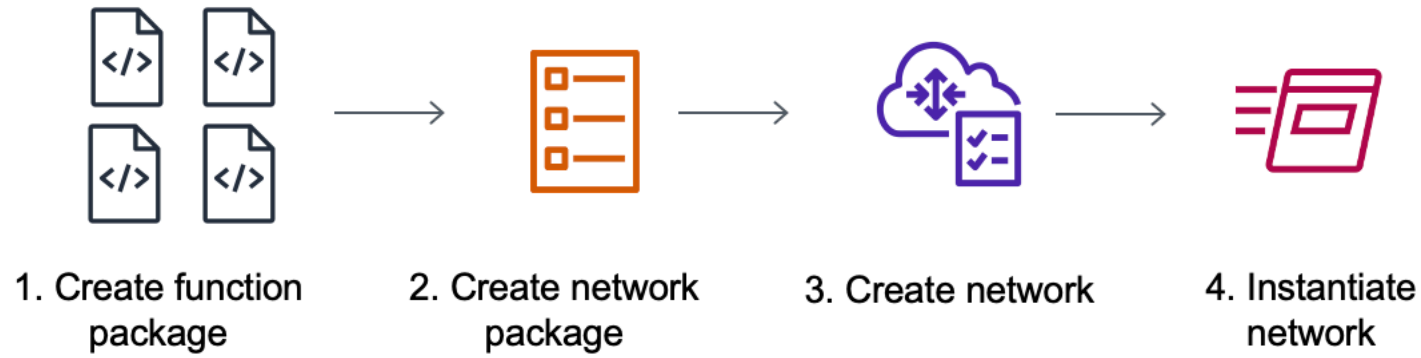
Sie müssen eine IAM-Servicerolle erstellen, um verschiedene Teile Ihrer AWS TNB-Lösung zu verwalten. AWS TNB-Servicerollen können in Ihrem Namen API-Aufrufe an andere AWS Dienste wie AWS CloudFormation AWS CodeBuild, und verschiedene Rechen- und Speicherdienste richten, um Ressourcen für Ihre Bereitstellung zu instanziieren und zu verwalten.

Weitere Informationen zur AWS TNB-Servicerolle finden Sie unter. Identitäts- und Zugriffsmanagement für TNB AWS

Erste Schritte mit AWS TNB

In diesem Tutorial wird gezeigt, wie Sie AWS TNB verwenden, um eine Netzwerkfunktion bereitzustellen, z. B. die Centralized Unit (CU), die Access and Mobility Management Function (AMF) oder die 5G User Plane Function (UPF).

Das folgende Diagramm veranschaulicht den Bereitstellungsprozess:



Aufgaben

- Voraussetzungen
- Erstellen Sie ein Funktionspaket
- Erstellen Sie ein Netzwerkpaket
- Erstellen und instanziieren Sie eine Netzwerkinstanz
- Bereinigen

Voraussetzungen

Bevor Sie eine erfolgreiche Bereitstellung durchführen können, müssen Sie über Folgendes verfügen:

- Ein AWS Business Support-Plan.
- Berechtigungen durch IAM-Rollen.
- Ein Network Function (NF) -Paket, das ETSI SOL001/SOL004 entspricht.
- NSD-Vorlagen (Network Service Descriptor), die ETSI SOL007 entsprechen.

Sie können ein Beispielfunktionspaket oder ein Netzwerkpaket von der Website Beispielpakete für TNB verwenden. AWS GitHub

Erstellen Sie ein Funktionspaket

Um ein Funktionspaket zu erstellen

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich die Option Funktionspakete aus.
3. Wählen Sie Funktionspaket erstellen.
4. Wählen Sie unter Funktionspaket hochladen die Option Datei auswählen aus und laden Sie Ihr CSAR-Paket als `.zip` Datei hoch.
5. (Optional) Wählen Sie unter Tags die Option Neues Tag hinzufügen aus und geben Sie einen Schlüssel und einen Wert ein. Sie können Tags verwenden, um Ihre Ressourcen zu suchen und zu filtern oder Ihre AWS Kosten zu verfolgen.
6. Wählen Sie Weiter aus.
7. Überprüfen Sie die Paketdetails und wählen Sie dann Funktionspaket erstellen.

Erstellen Sie ein Netzwerkpaket

Um ein Netzwerkpaket zu erstellen

1. Wählen Sie im Navigationsbereich Netzwerkpakete aus.
2. Wählen Sie Netzwerkpaket erstellen aus.
3. Wählen Sie unter Netzwerkpaket hochladen die Option Datei auswählen aus und laden Sie Ihre NSD als `.zip` Datei hoch.
4. (Optional) Wählen Sie unter Tags die Option Neues Tag hinzufügen aus und geben Sie einen Schlüssel und einen Wert ein. Sie können Tags verwenden, um Ihre Ressourcen zu suchen und zu filtern oder Ihre AWS Kosten zu verfolgen.
5. Wählen Sie Weiter aus.
6. Wählen Sie Netzwerkpaket erstellen.

Erstellen und instanziieren Sie eine Netzwerkinstanz

Um eine Netzwerkinstanz zu erstellen und zu instanziieren

1. Wählen Sie im Navigationsbereich Netzwerke aus.
2. Wählen Sie Netzwerkinstanz erstellen aus.
3. Geben Sie einen Namen und eine Beschreibung für das Netzwerk ein und wählen Sie dann Weiter.
4. Wählen Sie Ihr NSD aus. Überprüfen Sie die Details und wählen Sie dann Weiter.
5. Wählen Sie Netzwerkinstanz erstellen aus. Der Ausgangszustand ist`Created`.
6. Wählen Sie die ID der Netzwerkinstanz und dann Instantiieren aus.
7. Wählen Sie Netzwerk instanziieren.
8. Verwenden Sie das Aktualisierungssymbol, um den Status Ihrer Netzwerkinstanz zu verfolgen.

Bereinigen

So bereinigen Sie Ihre Ressourcen

1. Wählen Sie im Navigationsbereich Netzwerke aus.
2. Wählen Sie die ID des Netzwerks und dann Terminate aus.
3. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie die Netzwerk-ID ein und wählen Sie dann Terminate.
4. Verwenden Sie das Aktualisierungssymbol, um den Status Ihrer Netzwerkinstanz zu verfolgen.
5. (Optional) Wählen Sie das Netzwerk aus und klicken Sie auf Löschen.

Funktionspakete für AWS TNB

Ein Funktionspaket ist eine ZIP-Datei im CSAR-Format (Cloud Service Archive), die eine Netzwerkfunktion (eine ETSI-Standard-Telekommunikationsanwendung) und einen Funktionspaketdeskriptor enthält, der den TOSCA-Standard verwendet, um zu beschreiben, wie die Netzwerkfunktionen in Ihrem Netzwerk ausgeführt werden sollen.

Aufgaben

- Erstellen Sie ein Funktionspaket in TNB AWS
- Sehen Sie sich ein Funktionspaket in TNB an AWS
- Laden Sie ein Funktionspaket von AWS TNB herunter
- Löscht ein Funktionspaket aus TNB AWS

Erstellen Sie ein Funktionspaket in TNB AWS

Erfahren Sie, wie Sie ein Funktionspaket im AWS TNB-Netzwerkfunktionskatalog erstellen. Das Erstellen eines Funktionspakets ist der erste Schritt zur Erstellung eines Netzwerks in TNB. Sobald Sie ein Funktionspaket hochgeladen haben, müssen Sie ein Netzwerkpaket erstellen.

Console

Um ein Funktionspaket mit der Konsole zu erstellen

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich die Option Funktionspakete aus.
3. Wählen Sie Funktionspaket erstellen.
4. Wählen Sie Datei auswählen und laden Sie das CSAR-Paket Ihrer NF hoch.
5. Wählen Sie Weiter aus.
6. Überprüfen Sie die Paketdetails.
7. Wählen Sie Funktionspaket erstellen.

AWS CLI

Um ein Funktionspaket zu erstellen, verwenden Sie AWS CLI

1. Verwenden Sie den create-sol-function-packageBefehl, um ein neues Funktionspaket zu erstellen:

```
aws tnb create-sol-function-package
```

2. Verwenden Sie den Befehl put-sol-function-package-content, um den Inhalt des Funktionspakets hochzuladen. Beispiele:

```
aws tnb put-sol-function-package-content \
--vnf-pkg-id ^fp-[a-f0-9]{17}$ \
--content-type application/zip \
--file "fileb://valid-free5gc-udr.zip" \
--endpoint-url "https://tnb.us-west-2.amazonaws.com" \
--region us-west-2
```

Sehen Sie sich ein Funktionspaket in TNB an AWS

Erfahren Sie, wie Sie den Inhalt eines Funktionspakets einsehen können.

Console

So zeigen Sie ein Funktionspaket mit der Konsole an

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich die Option Funktionspakete aus.
3. Verwenden Sie das Suchfeld, um das Funktionspaket zu finden

AWS CLI

Um ein Funktionspaket anzuzeigen, verwenden Sie AWS CLI

1. Verwenden Sie den list-sol-function-packagesBefehl, um Ihre Funktionspakete aufzulisten.

```
aws tnb list-sol-function-packages
```

2. Verwenden Sie den get-sol-function-packageBefehl, um Details zu einem Funktionspaket anzuzeigen.

```
aws tnb get-sol-function-package \
--vnf-pkg-id ^fp-[a-f0-9]{17}$ \
--endpoint-url "https://tnb.us-west-2.amazonaws.com" \
--region us-west-2
```

Laden Sie ein Funktionspaket von AWS TNB herunter

Erfahren Sie, wie Sie ein Funktionspaket aus dem AWS TNB-Netzwerkfunktionskatalog herunterladen.

Console

So laden Sie ein Funktionspaket über die Konsole herunter

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole die Option Funktionspakete aus.
3. Verwenden Sie das Suchfeld, um das Funktionspaket zu finden
4. Wählen Sie das Funktionspaket
5. Wählen Sie Aktionen, Herunterladen.

AWS CLI

Um ein Funktionspaket herunterzuladen, verwenden Sie AWS CLI

Verwenden Sie den Befehl get-sol-function-package-content, um ein Funktionspaket herunterzuladen.

```
aws tnb get-sol-function-package-content \
--vnf-pkg-id ^fp-[a-f0-9]{17}$ \
--accept "application/zip" \
--endpoint-url "https://tnb.us-west-2.amazonaws.com" \
--region us-west-2
```

Löscht ein Funktionspaket aus TNB AWS

Erfahren Sie, wie Sie ein Funktionspaket aus dem AWS TNB-Netzwerkfunktionskatalog löschen. Um ein Funktionspaket zu löschen, muss sich das Paket in einem deaktivierten Zustand befinden.

Console

Um ein Funktionspaket mit der Konsole zu löschen

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich die Option Funktionspakete aus.
3. Verwenden Sie das Suchfeld, um das Funktionspaket zu finden.
4. Wählen Sie ein Funktionspaket.
5. Wählen Sie Aktionen, deaktivieren.
6. Wählen Sie Actions (Aktionen), Delete (Löschen) aus.

AWS CLI

Um ein Funktionspaket zu löschen, verwenden Sie AWS CLI

1. Verwenden Sie den update-sol-function-packageBefehl, um ein Funktionspaket zu deaktivieren.

```
aws tnb update-sol-function-package --vnf-pkg-id ^fp-[a-f0-9]{17}$ ---
operational-state DISABLED
```

2. Verwenden Sie den delete-sol-function-packageBefehl, um ein Funktionspaket zu löschen.

```
aws tnb delete-sol-function-package \
--vnf-pkg-id ^fp-[a-f0-9]{17}$ \
--endpoint-url "https://tnb.us-west-2.amazonaws.com" \
--region us-west-2
```

Netzwerkpakete für AWS TNB

Ein Netzwerkpaket ist eine ZIP-Datei im CSAR-Format (Cloud Service Archive), die die Funktionspakete definiert, die Sie bereitstellen möchten, und die AWS Infrastruktur, auf der Sie sie bereitstellen möchten.

Aufgaben

- Erstellen Sie ein Netzwerkpaket in TNB AWS
- Sehen Sie sich ein Netzwerkpaket in TNB an AWS
- Laden Sie ein Netzwerkpaket von AWS TNB herunter
- Löscht ein Netzwerkpaket aus TNB AWS

Erstellen Sie ein Netzwerkpaket in TNB AWS

Ein Netzwerkpaket besteht aus einer NSD-Datei (Network Service Descriptor) (erforderlich) und allen zusätzlichen Dateien (optional), z. B. Skripten, die auf Ihre Bedürfnisse zugeschnitten sind. Wenn Ihr Netzwerkpaket beispielsweise mehrere Funktionspakete enthält, können Sie mithilfe der NSD definieren, welche Netzwerkfunktionen in bestimmten VPCs, Subnetzen oder Amazon EKS-Clustern ausgeführt werden sollen.

Erstellen Sie ein Netzwerkpaket, nachdem Sie Funktionspakete erstellt haben. Sobald Sie ein Netzwerkpaket erstellt haben, müssen Sie eine Netzwerkinstanz erstellen.

Console

Um ein Netzwerkpaket mit der Konsole zu erstellen

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Netzwerkpakete aus.
3. Wählen Sie Netzwerkpaket erstellen aus.
4. Wählen Sie Datei auswählen und laden Sie Ihr CSAR-Paket hoch.
5. Wählen Sie Weiter aus.
6. Überprüfen Sie die Paketdetails.
7. Wählen Sie Netzwerkpaket erstellen.

AWS CLI

Um ein Netzwerkpaket mit dem zu erstellen AWS CLI

1. Verwenden Sie den create-sol-network-packageBefehl, um ein Netzwerkpaket zu erstellen.

```
aws tnb create-sol-network-package
```

2. Verwenden Sie den Befehl put-sol-network-package-content, um den Inhalt eines Netzwerkpakets hochzuladen. Beispiele:

```
aws tnb put-sol-network-package-content \
--nsd-info-id ^np-[a-f0-9]{17}$ \
--content-type application/zip \
--file "fileb://free5gc-core-1.0.9.zip" \
--endpoint-url "https://tnb.us-west-2.amazonaws.com" \
--region us-west-2
```

Sehen Sie sich ein Netzwerkpaket in TNB an AWS

Erfahren Sie, wie Sie den Inhalt eines Netzwerkpakets anzeigen.

Console

So zeigen Sie ein Netzwerkpaket mit der Konsole an

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Netzwerkpakete aus.
3. Verwenden Sie das Suchfeld, um das Netzwerkpaket zu finden.

AWS CLI

Um ein Netzwerkpaket anzuzeigen, verwenden Sie AWS CLI

1. Verwenden Sie den list-sol-network-packagesBefehl, um Ihre Netzwerkpakete aufzulisten.

```
aws tnb list-sol-network-packages
```

2. Verwenden Sie den get-sol-network-packageBefehl, um Details zu einem Netzwerkpaket anzuzeigen.

```
aws tnb get-sol-network-package \
--nsd-info-id ^np-[a-f0-9]{17}$ \
--endpoint-url "https://tnb.us-west-2.amazonaws.com" \
--region us-west-2
```

Laden Sie ein Netzwerkpaket von AWS TNB herunter

Erfahren Sie, wie Sie ein Netzwerkpaket aus dem AWS TNB-Netzwerkdienstkatalog herunterladen.

Console

So laden Sie ein Netzwerkpaket über die Konsole herunter

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Netzwerkpakete aus.
3. Verwenden Sie das Suchfeld, um das Netzwerkpaket zu finden
4. Wählen Sie das Netzwerkpaket aus.
5. Wählen Sie Aktionen, Herunterladen.

AWS CLI

Um ein Netzwerkpaket herunterzuladen, verwenden Sie AWS CLI

- Verwenden Sie den Befehl get-sol-network-package-content, um ein Netzwerkpaket herunterzuladen.

```
aws tnb get-sol-network-package-content \
--nsd-info-id ^np-[a-f0-9]{17}$ \
--accept "application/zip" \
--endpoint-url "https://tnb.us-west-2.amazonaws.com" \
--region us-west-2
```

Löscht ein Netzwerkpaket aus TNB AWS

Erfahren Sie, wie Sie ein Netzwerkpaket aus dem AWS TNB-Netzwerkdienstkatalog löschen. Um ein Netzwerkpaket zu löschen, muss sich das Paket im deaktivierten Zustand befinden.

Console

Um ein Netzwerkpaket mit der Konsole zu löschen

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Netzwerkpakete aus.
3. Verwenden Sie das Suchfeld, um das Netzwerkpaket zu finden
4. Wählen Sie ein Netzwerkpaket
5. Wählen Sie Aktionen, deaktivieren.
6. Wählen Sie Actions (Aktionen), Delete (Löschen) aus.

AWS CLI

Um ein Netzwerkpaket mit dem zu löschen AWS CLI

1. Verwenden Sie den update-sol-network-packageBefehl, um ein Netzwerkpaket zu deaktivieren.

```
aws tnb update-sol-network-package --nsd-info-id ^np-[a-f0-9]{17}$ --nsd-
operational-state DISABLED
```

2. Verwenden Sie den delete-sol-network-packageBefehl, um ein Netzwerkpaket zu löschen.

```
aws tnb delete-sol-network-package \
--nsd-info-id ^np-[a-f0-9]{17}$ \
--endpoint-url "https://tnb.us-west-2.amazonaws.com" \
--region us-west-2
```

Netzwerkinstanzen für AWS TNB

Eine Netzwerkinstanz ist ein einzelnes Netzwerk, das in AWS TNB erstellt wurde und bereitgestellt werden kann.

Aufgaben

- Instanziieren Sie eine Netzwerkinstanz mithilfe von TNB AWS
- Eine Netzwerkinstanz in TNB anzeigen AWS
- Aktualisieren Sie eine Netzwerkinstanz in AWS TNB
- Beenden und löschen Sie eine Netzwerkinstanz aus AWS TNB

Instanziieren Sie eine Netzwerkinstanz mithilfe von TNB AWS

Sie erstellen eine Netzwerkinstanz, nachdem Sie ein Netzwerkpaket erstellt haben. Sobald Sie eine Netzwerkinstanz erstellt haben, müssen Sie sie instanziieren. Wenn Sie eine Netzwerkinstanz instanziieren, stellt AWS TNB die Netzwerkfunktionen gemäß den Spezifikationen im Netzwerkdienstdeskriptor bereit.

Console

Um eine Netzwerkinstanz mithilfe der Konsole zu erstellen und zu instanziieren

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Netzwerke aus.
3. Wählen Sie Netzwerkinstanz erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Instanz ein und wählen Sie dann Weiter.
5. Wählen Sie Ihr NSD aus. Überprüfen Sie die Angaben und wählen Sie dann Weiter.
6. Wählen Sie Netzwerkinstanz erstellen aus.
7. Wählen Sie Instanziieren.
8. Wählen Sie Netzwerk instanziieren.
9. Aktualisieren Sie, um den Status Ihrer Netzwerkinstanz zu verfolgen.

AWS CLI

Um eine Netzwerkinstanz mit dem zu erstellen und zu instanziieren AWS CLI

1. Verwenden Sie den create-sol-network-instanceBefehl, um eine Netzwerkinstanz zu erstellen.

```
aws tnb create-sol-network-instance --nsd-info-id ^np-[a-f0-9]{17}$ --ns-name
 "SampleNs" --ns-description "Sample"
```

2. Verwenden Sie den instantiate-sol-network-instanceBefehl, um die Netzwerkinstanz zu instanziieren.

```
aws tnb instantiate-sol-network-instance --ns-instance-id ^ni-[a-f0-9]{17}$
```

Eine Netzwerkinstanz in TNB anzeigen AWS

Erfahren Sie, wie Sie eine Netzwerkinstanz anzeigen.

Console

So zeigen Sie eine Netzwerkinstanz mithilfe der Konsole an

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Netzwerkinstanzen aus.
3. Verwenden Sie das Suchfeld, um die Netzwerkinstanz zu finden.

AWS CLI

Um eine Netzwerkinstanz mit dem anzuzeigen AWS CLI

1. Verwenden Sie den list-sol-network-instancesBefehl, um Ihre Netzwerkinstanzen aufzulisten.

```
aws tnb list-sol-network-instances
```

2. Verwenden Sie den get-sol-network-instanceBefehl, um Details zu einer Netzwerkinstanz anzuzeigen.

```
aws tnb get-sol-network-instance --ns-instance-id ^ni-[a-f0-9]{17}$
```

Aktualisieren Sie eine Netzwerkinstanz in AWS TNB

Erfahren Sie, wie Sie eine Netzwerkinstanz aktualisieren.

Console

Um die Netzwerkinstanz mithilfe der Konsole zu aktualisieren

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Netzwerke aus.
3. Wählen Sie die ID der Netzwerkinstanz aus.
4. Wählen Sie auf der Registerkarte Funktionen die Funktionsinstanz aus, die aktualisiert werden soll.
5. Wählen Sie Aktualisieren aus.
6. Geben Sie Ihre Update-Overrides ein, um das Update zu bestätigen.
7. Wählen Sie Aktualisieren aus.
8. Aktualisieren Sie, um den Status Ihrer Netzwerkinstanz zu verfolgen.

AWS CLI

Verwenden Sie die CLI, um eine Netzwerkinstanz zu aktualisieren

Verwenden Sie den update-sol-network-instanceBefehl, um eine Netzwerkinstanz zu aktualisieren.

```
aws tnb update-sol-network-instance --ns-instance-id ^ni-[a-f0-9]{17}$ --update-type
 MODIFY_VNF_INFORMATION --modify-vnf-info ...
```

Beenden und löschen Sie eine Netzwerkinstanz aus AWS TNB

Um eine Netzwerkinstanz zu löschen, muss sich die Instanz in einem beendeten Zustand befinden.

Console

Um eine Netzwerkinstanz mithilfe der Konsole zu beenden und zu löschen

1. Öffnen Sie die AWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.

2. Wählen Sie im Navigationsbereich Netzwerke aus.
3. Wählen Sie die ID der Netzwerkinstanz aus.
4. Wähen Sie Beenden.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie die ID ein und wählen Sie Terminate.
6. Aktualisieren Sie, um den Status Ihrer Netzwerkinstanz zu verfolgen.
7. (Optional) Wählen Sie die Netzwerkinstanz aus und klicken Sie auf Löschen.

AWS CLI

Um eine Netzwerkinstanz zu beenden und zu löschen, verwenden Sie AWS CLI

1. Verwenden Sie den terminate-sol-network-instanceBefehl, um eine Netzwerkinstanz zu beenden.

```
aws tnb terminate-sol-network-instance --ns-instance-id ^ni-[a-f0-9]{17}$
```

2. (Optional) Verwenden Sie den delete-sol-network-instanceBefehl, um eine Netzwerkinstanz zu löschen.

```
aws tnb delete-sol-network-instance --ns-instance-id ^ni-[a-f0-9]{17}$
```

Netzwerkbetrieb fürAWS TNB

Ein Netzwerkvorgang ist jeder Vorgang, der an Ihrem Netzwerk ausgeführt wird, z. B. die Instanziierung oder Terminierung einer Netzwerkinstanz.

Aufgaben

- Anzeigen eines Netzwerkvorgangs
- Abbrechen eines Netzwerkvorgangs

Anzeigen eines Netzwerkvorgangs

Sehen Sie sich die Details eines Netzwerkvorgangs an, einschließlich der mit dem Netzwerkbetrieb verbundenen Aufgaben und des Status der Aufgaben.

Console

Anhalten eines Netzwerkvorgangs mit der

1. Öffnen Sie dieAWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Network aus.
3. Verwenden Sie das Suchfeld, um die Netzwerkinstanz zu suchen.
4. Wählen Sie auf der Registerkarte Deployments die Option Network Operation aus.

AWS CLI

Um einen Netzwerkvorgang mit dem anzuzeigenAWS CLI

1. Verwenden Sie den list-sol-network-operationsBefehl, um alle Netzwerkvorgänge aufzulisten.

```
aws tnb list-sol-network-operations
```

2. Verwenden Sie den get-sol-network-operationBefehl, um Details zu einem Netzwerkvorgang anzuzeigen.

```
aws tnb get-sol-network-operation --ns-lcm-op-occ-id ^no-[a-f0-9]{17}$
```

Abbrechen eines Netzwerkvorgangs

Erfahren Sie, wie Sie einen Netzwerkvorgang abbrechen.

Console

So stornieren Sie eine mithilfe der

1. Öffnen Sie dieAWS TNB-Konsole unter https://console.aws.amazon.com/tnb/.
2. Wählen Sie im Navigationsbereich Networks aus.
3. Wählen Sie die ID des Netzwerks aus, um seine Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Deployments die Option Network Operation aus.
5. Wählen Sie Vorgang abbrechen.

AWS CLI

Um einen Netzwerkvorgang abzubrechen, verwenden Sie denAWS CLI

Verwenden Sie den cancel-sol-network-operationBefehl, um einen Netzwerkvorgang abzubrechen.

```
aws tnb cancel-sol-network-operation --ns-lcm-op-occ-id ^no-[a-f0-9]{17}$
```

TOSCA Referenz fürAWS TNB

Topology and Orchestration Specification for Cloud Applications (TOSCA) ist eine deklarative Syntax, die CSPs verwenden, um eine Topologie von cloudbasierten Webdiensten, ihren Komponenten, Beziehungen und den Prozessen, die sie verwalten, zu beschreiben. CSPs beschreiben die Verbindungspunkte, die logischen Verknüpfungen zwischen den Verbindungspunkten und die Richtlinien wie Affinität und Sicherheit in einer TOSCA-Vorlage. CSPs laden die Vorlage dann aufAWS TNB hoch, wodurch die Ressourcen synthetisiert werden, die für den Aufbau eines funktionierenden 5G-Netzwerks in denAWS Availability Zones erforderlich sind.

Inhalt

VNFD-Vorlage

Definiert eine VNFD-Vorlage (Virtual Network Function Descriptor).

Syntax

```
tosca_definitions_version: tnb_simple_yaml_1_0

topology_template:

  inputs:
    SampleInputParameter:
      type: String
      description: "Sample parameter description"
      default: "DefaultSampleValue"

  node_templates:
    SampleNode1: tosca.nodes.AWS.VNF
```

Topologie-Vorlage

`node_templates`

Die TOSCAAWSKnoten. Die möglichen Knoten sind:

- AWS.VNF
- AWS.Artefakte.Helm

AWS.VNF

Definiert eineAWSKnoten für virtuelle Netzwerkfunktionen (VNF).

Syntax

```
tosca.nodes.AWS.VNF:
  properties:
    descriptor_id: String
    descriptor_version: String
    descriptor_name: String
    provider: String
  requirements:
    helm: String
```

Eigenschaften

`descriptor_id`

Die UUID des Deskriptors.

Erforderlich: Ja

Typ: Zeichenfolge

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

`descriptor_version`

Die Version des VNFD.

Erforderlich: Ja

Typ: Zeichenfolge

Pattern: `^[0-9]{1,5}\\.[0-9]{1,5}\\.[0-9]{1,5}.*`

`descriptor_name`

Der Name des Deskriptors.

Erforderlich: Ja

Typ: Zeichenfolge

`provider`

Der Autor des VNFD.

Erforderlich: Ja

Typ: Zeichenfolge

Voraussetzungen

`helm`

Das Helm-Verzeichnis, das Container-Artefakte definiert. Dies ist ein Hinweis auf[AWS.Artefakte.Helm](#).

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
SampleVNF:
  type: tosca.nodes.AWS.VNF
  properties:
    descriptor_id: "6a792e0c-be2a-45fa-989e-5f89d94ca898"
    descriptor_version: "1.0.0"
    descriptor_name: "Test VNF Template"
    provider: "Operator"
  requirements:
```

```
    helm: SampleHelm
```

AWS.Artifacts.Helm

Definiert eineAWSHelmknoten.

Syntax

```
tosca.nodes.AWS.Artifacts.Helm:
  properties:
    implementation: String
```

Eigenschaften

`implementation`

Das lokale Verzeichnis, das das Helm-Diagramm im CSAR-Paket enthält.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
SampleHelm:
  type: tosca.nodes.AWS.Artifacts.Helm
  properties:
    implementation: "./vnf-helm"
```

NSD-Vorlage

Definiert eine NSD-Vorlage (Network Service Descriptor).

Syntax

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

```
vnfds:
  - descriptor_id: String
    namespace: String

topology_template:

  inputs:
    SampleInputParameter:
      type: String
      description: "Sample parameter description"
      default: "DefaultSampleValue"

  node_templates:
    SampleNode1: tosca.nodes.AWS.NS
```

Definierte Parameter werden verwendet

Wenn Sie einen Parameter dynamisch übergeben möchten, z. B. den CIDR-Block für den VPC-Knoten, können Sie die `{ get_input: input-parameter-name }` Syntax verwenden und die Parameter in der NSD-Vorlage definieren. Verwenden Sie den Parameter dann in derselben NSD-Vorlage erneut.

Das folgende Beispiel zeigt, wie Parameter definiert und verwendet werden:

```
tosca_definitions_version: tnb_simple_yaml_1_0

topology_template:

  inputs:
    cidr_block:
      type: String
      description: "CIDR Block for VPC"
      default: "10.0.0.0/24"

  node_templates:
    ExampleSingleClusterNS:
      type: tosca.nodes.AWS.NS
      properties:
        descriptor_id: "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
         .....

    ExampleVPC:
      type: tosca.nodes.AWS.Networking.VPC
```

```
      properties:
        cidr_block: { get_input: cidr_block }
```

VNFD-Import

`descriptor_id`

Die UUID des Deskriptors.

Erforderlich: Ja

Typ: Zeichenfolge

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

`namespace`

Der eindeutige Name.

Erforderlich: Ja

Typ: Zeichenfolge

Topologie-Vorlage

`node_templates`

Die möglichen AWS TOSCA-Knoten sind:

- AWS.NS
- AWS.compute.eks
- AWS.Compute.EKS. AuthRole
- AWS.eks berechnen ManagedNode
- AWS.compute.EKS SelfManagedNode
- AWS. Berechnen. PlacementGroup
- AWS.Berechnen. UserData
- AWS.Netzwerke. SecurityGroup
- AWS.Netzwerke. SecurityGroupEgressRule

- AWS.Netzwerke. SecurityGroupIngressRule
- AWS.Ressource.Import
- AWS.Netzwerking.ENI
- AWS.HookExecution
- AWS.Netzwerke. InternetGateway
- AWS.Netzwerke. RouteTable
- AWS.Netzwerk.Subnetz
- AWS.Bereitstellung.VNF-Bereitstellung
- AWS.Netzwerk.VPC
- AWS.networking.NAT-Gateway
- AWS.Netzwerkung.Route

AWS.NS

Definiert einen AWS Netzwerkdienstknoten (NS).

Syntax

```
tosca.nodes.AWS.NS:
  properties:
    descriptor_id: String
    descriptor_version: String
    descriptor_name: String
```

Eigenschaften

`descriptor_id`

Die UUID des Deskriptors.

Erforderlich: Ja

Typ: Zeichenfolge

Pattern: `[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}`

`descriptor_version`

Die Version des NSD.

Erforderlich: Ja

Typ: Zeichenfolge

Pattern: `^[0-9]{1,5}\\.[0-9]{1,5}\\.[0-9]{1,5}.*`

`descriptor_name`

Der Name des Deskriptors.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
SampleNS:
  type: tosca.nodes.AWS.NS
  properties:
    descriptor_id: "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    descriptor_version: "1.0.0"
    descriptor_name: "Test NS Template"
```

AWS.compute.EKS

Geben Sie den Namen des Clusters, die gewünschte Kubernetes-Version und eine Rolle an, die es der Kubernetes-Steuerebene ermöglicht, die für Ihre NFs erforderlichen Ressourcen zu verwalten. AWS Multus Container Network Interface (CNI) -Plugins sind aktiviert. Sie können mehrere Netzwerkschnittstellen anhängen und eine erweiterte Netzwerkkonfiguration auf die Kubernetes-basierten Netzwerkfunktionen anwenden. Sie geben auch den Cluster-Endpunktzugriff und die Subnetze für Ihren Cluster an.

Syntax

```
tosca.nodes.AWS.Compute.EKS:
  capabilities:
```

```
    multus:
      properties:
        enabled: Boolean
        multus_role: String
    ebs_csi:
      properties:
        enabled: Boolean
        version: String
  properties:
    version: String
    access: String
    cluster_role: String
    tags: List
    ip_family: String
  requirements:
    subnets: List
```

Funktionen

`multus`

Optional. Eigenschaften, die die Verwendung des Multus Container Network Interface (CNI) definieren.

Wenn Sie einschließen`multus`, geben Sie die Eigenschaften `enabled` und `multus_role` an.

`enabled`

Gibt an, ob die standardmäßige Multus-Funktion aktiviert ist.

Erforderlich: Ja

Typ: Boolesch

`multus_role`

Die Rolle für die Multus-Netzwerkschnittstellenverwaltung.

Erforderlich: Ja

Typ: Zeichenfolge

`ebs_csi`

Eigenschaften, die den im Amazon EKS-Cluster installierten Amazon EBS Container Storage Interface (CSI) -Treiber definieren.

Aktivieren Sie dieses Plugin, um selbstverwaltete Amazon EKS-Nodes auf AWS Outposts, AWS Local Zones oder AWS-Regionen zu verwenden. Weitere Informationen finden Sie unter Amazon Elastic Block Store CSI-Treiber im Amazon EKS-Benutzerhandbuch.

`enabled`

Zeigt an, ob der standardmäßige Amazon EBS CSI-Treiber installiert ist.

Erforderlich: Nein

Typ: Boolesch

`version`

Die Version des Amazon EBS CSI-Treiber-Add-ons. Die Version muss mit einer der von der DescribeAddonVersionsAktion zurückgegebenen Versionen übereinstimmen. Weitere Informationen finden Sie DescribeAddonVersionsin der Amazon EKS API-Referenz

Erforderlich: Nein

Typ: Zeichenfolge

Eigenschaften

`version`

Die Kubernetes-Version für den Cluster. AWS Telco Network Builder unterstützt die Kubernetes-Versionen 1.23 bis 1.29.

Erforderlich: Ja

Typ: Zeichenfolge

Mögliche Werte: 1,23 | 1,24 | 1,25 | 1,26 | 1,27 | 1,28 | 1,29

`access`

Der Cluster-Endpunktzugriff.

Erforderlich: Ja

Typ: Zeichenfolge

Mögliche Werte: `PRIVATE` | `PUBLIC` | `ALL`

`cluster_role`

Die Rolle für die Clusterverwaltung.

Erforderlich: Ja

Typ: Zeichenfolge

`tags`

Tags, die an die Ressource angehängt werden sollen.

Erforderlich: Nein

Typ: Liste

`ip_family`

Gibt die IP-Familie für Service- und Pod-Adressen im Cluster an.

Zulässiger Wert:`IPv4`, `IPv6`

Standardwert: `IPv4`

Erforderlich: Nein

Typ: Zeichenfolge

Voraussetzungen

`subnets`

Ein AWS.Networking.Subnet-Knoten.

Erforderlich: Ja

Typ: Liste

Beispiel

```
SampleEKS:
  type: tosca.nodes.AWS.Compute.EKS
  properties:
    version: "1.23"
    access: "ALL"
    cluster_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
    ip_family: "IPv6"
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  capabilities:
    multus:
      properties:
        enabled: true
        multus_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/MultusRole"
    ebs_csi:
      properties:
        enabled: true
        version: "v1.16.0-eksbuild.1"
  requirements:
    subnets:
    - SampleSubnet01
    - SampleSubnet02
```

AWS.compute.EKS. AuthRole

An AuthRole ermöglicht es Ihnen, dem Amazon EKS-Cluster IAM-Rollen hinzuzufügen, `aws-auth` `ConfigMap` sodass Benutzer mithilfe einer IAM-Rolle auf den Amazon EKS-Cluster zugreifen können.

Syntax

```
tosca.nodes.AWS.Compute.EKS.AuthRole:
  properties:
    role_mappings: List
      arn: String
      groups: List
  requirements:
    clusters: List
```

Eigenschaften

`role_mappings`

Liste der Zuordnungen, die IAM-Rollen definieren, die dem Amazon EKS-Cluster hinzugefügt werden müssen. `aws-auth ConfigMap`

`arn`

Der ARN der IAM-Rolle.

Erforderlich: Ja

Typ: Zeichenfolge

`groups`

Kubernetes-Gruppen, die der in definierten Rolle zugewiesen werden sollen. `arn`

Erforderlich: Nein

Typ: Liste

Voraussetzungen

`clusters`

Ein AWS.compute.EKS-Knoten.

Erforderlich: Ja

Typ: Liste

Beispiel

```
EKSAuthMapRoles:
    type: tosca.nodes.AWS.Compute.EKS.AuthRole
    properties:
        role_mappings:
        - arn: arn:aws:iam::${AWS::TNB::AccountId}:role/TNBHookRole1
          groups:
          - system:nodes
```

```
          - system:bootstrappers
        - arn: arn:aws:iam::${AWS::TNB::AccountId}:role/TNBHookRole2
          groups:
          - system:nodes
          - system:bootstrappers
    requirements:
         clusters:
         - Free5GCEKS1
         - Free5GCEKS2
```

AWS.compute.EKS ManagedNode

AWS TNB unterstützt EKS Managed Node-Gruppen, um die Bereitstellung und das Lebenszyklusmanagement von Knoten (Amazon EC2 EC2-Instances) für Amazon EKS Kubernetes-Cluster zu automatisieren. Um eine EKS-Knotengruppe zu erstellen, müssen Sie die Amazon Machine Images (AMI) für Ihre Cluster-Worker-Knoten auswählen, indem Sie entweder die ID des AMI oder den AMI-Typ angeben. Sie stellen auch ein Amazon EC2 EC2-Schlüsselpaar für den SSH-Zugriff und die Skalierungseigenschaften für Ihre Knotengruppe bereit. Ihre Knotengruppe muss einem EKS-Cluster zugeordnet sein. Sie müssen die Subnetze für die Worker-Knoten bereitstellen.

Optional können Sie Ihrer Knotengruppe Sicherheitsgruppen, Knotenbezeichnungen und eine Platzierungsgruppe hinzufügen.

Syntax

```
tosca.nodes.AWS.Compute.EKSManagedNode:
  capabilities:
    compute:
      properties:
        ami_type: String
        ami_id: String
        instance_types: List
        key_pair: String
        root_volume_encryption: Boolean
        root_volume_encryption_key_arn: String
    scaling:
      properties:
        desired_size: Integer
        min_size: Integer
        max_size: Integer
  properties:
    node_role: String
```

```
    tags: List
  requirements:
    cluster: String
    subnets: List
    network_interfaces: List
    security_groups: List
    placement_group: String
    user_data: String
    labels: List
```

Funktionen

`compute`

Eigenschaften, die die Rechenparameter für die von Amazon EKS verwaltete Knotengruppe definieren, z. B. Amazon EC2 EC2-Instance-Typen und Amazon EC2 EC2-Instance-AMIs.

`ami_type`

Der von Amazon EKS unterstützte AMI-Typ.

Erforderlich: Ja

Typ: Zeichenfolge

Mögliche Werte: `AL2_x86_64` | | `AL2_x86_64_GPU` | `AL2_ARM_64` | | `CUSTOM` | `BOTTLEROCKET_ARM_64` | `BOTTLEROCKET_x86_64` `BOTTLEROCKET_ARM_64_NVIDIA` `BOTTLEROCKET_x86_64_NVIDIA`

`ami_id`

Die ID des AMI.

Erforderlich: Nein

Typ: Zeichenfolge

> **Note**
>
> Wenn `ami_type` sowohl als auch in der Vorlage angegeben `ami_id` sind, verwendet AWS TNB nur den `ami_id` Wert für die Erstellung`EKSManagedNode`.

`instance_types`

Die Instanzgröße.

Erforderlich: Ja

Typ: Liste

`key_pair`

Das EC2-Schlüsselpaar zur Aktivierung des SSH-Zugriffs.

Erforderlich: Ja

Typ: Zeichenfolge

`root_volume_encryption`

Aktiviert die Amazon EBS-Verschlüsselung für das Amazon EBS-Root-Volume. Wenn diese Eigenschaft nicht angegeben wird, verschlüsselt AWS TNB standardmäßig Amazon EBS-Root-Volumes.

Erforderlich: Nein

Standard: true

Typ: Boolesch

`root_volume_encryption_key_arn`

Der ARN des AWS KMS Schlüssels. AWS TNB unterstützt reguläre Schlüssel-ARN, Multi-Region-Schlüssel-ARN und Alias-ARN.

Erforderlich: Nein

Typ: Zeichenfolge

> **Note**
>
> - Wenn der Wert falsch `root_volume_encryption` ist, schließen Sie ihn nicht ein. `root_volume_encryption_key_arn`
> - AWS TNB unterstützt die Root-Volume-Verschlüsselung von Amazon EBS-gestützten AMIs.

- Wenn das Root-Volume des AMI bereits verschlüsselt ist, müssen Sie das `root_volume_encryption_key_arn` für AWS TNB hinzufügen, um das Root-Volume erneut zu verschlüsseln.
- Wenn das Root-Volume des AMI nicht verschlüsselt ist, verwendet AWS TNB das, `root_volume_encryption_key_arn` um das Root-Volume zu verschlüsseln.

  Wenn Sie dies nicht angeben`root_volume_encryption_key_arn`, verwendet AWS TNB den von bereitgestellten Standardschlüssel, um das Root-Volume AWS Key Management Service zu verschlüsseln.
- AWS TNB entschlüsselt kein verschlüsseltes AMI.

`scaling`

Eigenschaften, die die Skalierungsparameter für die von Amazon EKS verwaltete Knotengruppe definieren, z. B. die gewünschte Anzahl von Amazon EC2 EC2-Instances und die Mindest- und Höchstanzahl von Amazon EC2 EC2-Instances in der Knotengruppe.

`desired_size`

Die Anzahl der Instances in dieser. NodeGroup

Erforderlich: Ja

Typ: Ganzzahl

`min_size`

Die Mindestanzahl von Instanzen in diesem Bereich NodeGroup.

Erforderlich: Ja

Typ: Ganzzahl

`max_size`

Die maximale Anzahl von Instanzen in diesem Bereich NodeGroup.

Erforderlich: Ja

Typ: Ganzzahl

Eigenschaften

`node_role`

Der ARN der IAM-Rolle, die der Amazon EC2 EC2-Instance zugeordnet ist.

Erforderlich: Ja

Typ: Zeichenfolge

`tags`

Die Tags, die an die Ressource angehängt werden sollen.

Erforderlich: Nein

Typ: Liste

Voraussetzungen

`cluster`

Ein AWS.compute.EKS-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

`subnets`

Ein .Networking.Subnet-Knoten.AWS

Erforderlich: Ja

Typ: Liste

`network_interfaces`

Ein .Networking.ENI-Knoten.AWS Stellen Sie sicher, dass die Netzwerkschnittstellen und Subnetze auf dieselbe Availability Zone eingestellt sind. Andernfalls schlägt die Instanziierung fehl.

Wenn Sie diese Einstellung vornehmen`network_interfaces`, bezieht AWS TNB die Berechtigungen für ENIs aus der Eigenschaft, sofern Sie die `multus_role` Eigenschaft in den

AWS.Compute.eks-Knoten aufgenommen haben. `multus` Andernfalls erhält AWS TNB die Berechtigungen für ENIs aus der Eigenschaft node_role.

Erforderlich: Nein

Typ: Liste

`security_groups`

Ein .Networking.AWS SecurityGroupKnoten.

Erforderlich: Nein

Typ: Liste

`placement_group`

Ein tosca.nodes.AWS. Berechnen. PlacementGroupKnoten.

Erforderlich: Nein

Typ: Zeichenfolge

`user_data`

Ein tosca.nodes.AWS. Berechnen. UserDataKnotenreferenz. Ein Benutzerdatenskript wird an die Amazon EC2 EC2-Instances übergeben, die von der verwalteten Knotengruppe gestartet wurden. Fügen Sie der an die Knotengruppe übergebenen node_role die Berechtigungen hinzu, die für die Ausführung benutzerdefinierter Benutzerdaten erforderlich sind.

Erforderlich: Nein

Typ: Zeichenfolge

`labels`

Eine Liste von Knotenbezeichnungen. Eine Knotenbezeichnung muss einen Namen und einen Wert haben. Erstellen Sie ein Label anhand der folgenden Kriterien:

- Der Name und der Wert müssen durch getrennt werden=.
- Der Name und der Wert können jeweils bis zu 63 Zeichen lang sein.
- Das Etikett kann Buchstaben (A-Z, a-z,), Zahlen (0-9) und die folgenden Zeichen enthalten: `[-, _, ., *, ?]`
- Der Name und der Wert müssen mit einem alphanumerischen Zeichen oder beginnen und enden. `? *`

Beispiel: `myLabelName1=*NodeLabelValue1`

Erforderlich: Nein

Typ: Liste

Beispiel

```
SampleEKSManagedNode:
  type: tosca.nodes.AWS.Compute.EKSManagedNode
  capabilities:
    compute:
      properties:
        ami_type: "AL2_x86_64"
        instance_types:
          - "t3.xlarge"
        key_pair: "SampleKeyPair"
        root_volume_encryption: true
        root_volume_encryption_key_arn: "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    scaling:
      properties:
        desired_size: 1
        min_size: 1
        max_size: 1
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  requirements:
    cluster: SampleEKS
    subnets:
      - SampleSubnet
    network_interfaces:
      - SampleENI01
      - SampleENI02
    security_groups:
      - SampleSecurityGroup01
      - SampleSecurityGroup02
    placement_group: SamplePlacementGroup
    user_data: CustomUserData
    labels:
```

```
      - "sampleLabelName001=sampleLabelValue001"
      - "sampleLabelName002=sampleLabelValue002"
```

AWS.compute.EKS SelfManagedNode

AWS TNB unterstützt selbstverwaltete Amazon EKS-Knoten, um die Bereitstellung und das Lebenszyklusmanagement von Knoten (Amazon EC2 EC2-Instances) für Amazon EKS Kubernetes-Cluster zu automatisieren. Um eine Amazon EKS-Knotengruppe zu erstellen, müssen Sie die Amazon Machine Images (AMI) für Ihre Cluster-Worker-Knoten auswählen, indem Sie die AMI-ID angeben. Geben Sie optional ein Amazon EC2 EC2-Schlüsselpaar für den SSH-Zugriff an. Sie müssen auch den Instance-Typ und die gewünschte Mindest- und Maximalgröße angeben. Ihre Knotengruppe muss mit einem Amazon EKS-Cluster verknüpft sein. Sie müssen die Subnetze für die Worker-Knoten bereitstellen.

Optional können Sie Ihrer Knotengruppe Sicherheitsgruppen, Knotenbezeichnungen und eine Platzierungsgruppe hinzufügen.

Syntax

```
tosca.nodes.AWS.Compute.EKSSelfManagedNode:
  capabilities:
    compute:
      properties:
        ami_id: String
        instance_type: String
        key_pair: String
        root_volume_encryption: Boolean
        root_volume_encryption_key_arn: String
    scaling:
      properties:
        desired_size: Integer
        min_size: Integer
        max_size: Integer
  properties:
    node_role: String
    tags: List
  requirements:
    cluster: String
    subnets: List
    network_interfaces: List
    security_groups: List
    placement_group: String
```

```
    user_data: String
    labels: List
```

Funktionen

compute

Eigenschaften, die die Rechenparameter für die selbstverwalteten Amazon EKS-Knoten definieren, z. B. Amazon EC2 EC2-Instance-Typen und Amazon EC2 EC2-Instance-AMIs.

`ami_id`

Die AMI-ID, die zum Starten der Instance verwendet wurde. AWS TNB unterstützt Instances, die IMDSv2 nutzen. Weitere Informationen finden Sie unter IMDS-Version.

Erforderlich: Ja

Typ: Zeichenfolge

`instance_type`

Die Instanzgröße.

Erforderlich: Ja

Typ: Zeichenfolge

`key_pair`

Das Amazon EC2 EC2-Schlüsselpaar zur Aktivierung des SSH-Zugriffs.

Erforderlich: Ja

Typ: Zeichenfolge

`root_volume_encryption`

Aktiviert die Amazon EBS-Verschlüsselung für das Amazon EBS-Root-Volume. Wenn diese Eigenschaft nicht angegeben wird, verschlüsselt AWS TNB standardmäßig Amazon EBS-Root-Volumes.

Erforderlich: Nein

Standard: true

Typ: Boolesch

`root_volume_encryption_key_arn`

Der ARN des AWS KMS Schlüssels. AWS TNB unterstützt reguläre Schlüssel-ARN, Multi-Region-Schlüssel-ARN und Alias-ARN.

Erforderlich: Nein

Typ: Zeichenfolge

> **Note**
>
> - Wenn der Wert falsch `root_volume_encryption` ist, schließen Sie ihn nicht ein. `root_volume_encryption_key_arn`
> - AWS TNB unterstützt die Root-Volume-Verschlüsselung von Amazon EBS-gestützten AMIs.
> - Wenn das Root-Volume des AMI bereits verschlüsselt ist, müssen Sie das `root_volume_encryption_key_arn` für AWS TNB hinzufügen, um das Root-Volume erneut zu verschlüsseln.
> - Wenn das Root-Volume des AMI nicht verschlüsselt ist, verwendet AWS TNB das, `root_volume_encryption_key_arn` um das Root-Volume zu verschlüsseln.
>
>   Wenn Sie dies nicht angeben`root_volume_encryption_key_arn`, verwendet AWS TNB das AWS Managed Services Root-Volume zur Verschlüsselung.
> - AWS TNB entschlüsselt kein verschlüsseltes AMI.

`scaling`

Eigenschaften, die die Skalierungsparameter für die selbstverwalteten Amazon EKS-Knoten definieren, z. B. die gewünschte Anzahl von Amazon EC2 EC2-Instances und die Mindest- und Höchstanzahl von Amazon EC2 EC2-Instances in der Knotengruppe.

`desired_size`

Die Anzahl der Instances in dieser. NodeGroup

Erforderlich: Ja

Typ: Ganzzahl

`min_size`

Die Mindestanzahl von Instanzen in diesem Bereich NodeGroup.

Erforderlich: Ja

Typ: Ganzzahl

`max_size`

Die maximale Anzahl von Instanzen in diesem Bereich NodeGroup.

Erforderlich: Ja

Typ: Ganzzahl

Eigenschaften

`node_role`

Der ARN der IAM-Rolle, die der Amazon EC2 EC2-Instance zugeordnet ist.

Erforderlich: Ja

Typ: Zeichenfolge

`tags`

Die Tags, die an die Ressource angehängt werden sollen. Die Tags werden an die von der Ressource erstellten Instanzen weitergegeben.

Erforderlich: Nein

Typ: Liste

Voraussetzungen

`cluster`

Ein AWS.compute.EKS-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

`subnets`

Ein .Networking.Subnet-Knoten.AWS

Erforderlich: Ja

Typ: Liste

`network_interfaces`

Ein .Networking.ENI-Knoten.AWS Stellen Sie sicher, dass die Netzwerkschnittstellen und Subnetze auf dieselbe Availability Zone eingestellt sind. Andernfalls schlägt die Instanziierung fehl.

Wenn Sie diese Einstellung vornehmen`network_interfaces`, bezieht AWS TNB die Berechtigungen für ENIs aus der Eigenschaft, sofern Sie die `multus_role` Eigenschaft in den AWS.Compute.eks-Knoten aufgenommen haben. `multus` Andernfalls erhält AWS TNB die Berechtigungen für ENIs aus der Eigenschaft node_role.

Erforderlich: Nein

Typ: Liste

`security_groups`

Ein .Networking.AWS SecurityGroupKnoten.

Erforderlich: Nein

Typ: Liste

`placement_group`

Ein tosca.nodes.AWS. Berechnen. PlacementGroupKnoten.

Erforderlich: Nein

Typ: Zeichenfolge

`user_data`

Ein tosca.nodes.AWS. Berechnen. UserDataKnotenreferenz. Ein Benutzerdatenskript wird an die Amazon EC2 EC2-Instances übergeben, die von der selbstverwalteten Knotengruppe gestartet wurden. Fügen Sie der an die Knotengruppe übergebenen node_role die Berechtigungen hinzu, die für die Ausführung benutzerdefinierter Benutzerdaten erforderlich sind.

Erforderlich: Nein

Typ: Zeichenfolge

`labels`

Eine Liste von Knotenbezeichnungen. Eine Knotenbezeichnung muss einen Namen und einen Wert haben. Erstellen Sie ein Label anhand der folgenden Kriterien:

- Der Name und der Wert müssen durch getrennt werden=.
- Der Name und der Wert können jeweils bis zu 63 Zeichen lang sein.
- Die Bezeichnung kann Buchstaben (A-Z, a-z,), Zahlen (0-9) und die folgenden Zeichen enthalten: `[-, _, ., *, ?]`
- Der Name und der Wert müssen mit einem alphanumerischen Zeichen oder beginnen und enden. `? *`

Beispiel: `myLabelName1=*NodeLabelValue1`

Erforderlich: Nein

Typ: Liste

Beispiel

```
SampleEKSSelfManagedNode:
  type: tosca.nodes.AWS.Compute.EKSSelfManagedNode
  capabilities:
    compute:
      properties:
        ami_id: "ami-123123EXAMPLE"
        instance_type: "c5.large"
        key_pair: "SampleKeyPair"
        root_volume_encryption: true
        root_volume_encryption_key_arn: "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    scaling:
      properties:
        desired_size: 1
        min_size: 1
        max_size: 1
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleNodeRole"
    tags:
      - "Name=SampleVPC"
```

```
      - "Environment=Testing"
  requirements:
    cluster: SampleEKSCluster
    subnets:
      - SampleSubnet
    network_interfaces:
      - SampleNetworkInterface01
      - SampleNetworkInterface02
    security_groups:
      - SampleSecurityGroup01
      - SampleSecurityGroup02
    placement_group: SamplePlacementGroup
    user_data: CustomUserData
    labels:
      - "sampleLabelName001=sampleLabelValue001"
      - "sampleLabelName002=sampleLabelValue002"
```

AWS. Berechne. PlacementGroup

Ein PlacementGroup Knoten unterstützt verschiedene Strategien zum Platzieren von Amazon EC2 EC2-Instances.

Wenn Sie eine neue Amazon EC2-Instance starten, versucht der Amazon EC2-Service, die Instance so zu platzieren, dass alle Ihre Instances auf die zugrunde liegende Hardware verteilt sind, um korrelierte Ausfälle zu minimieren. Mithilfe von Placement-Gruppen können Sie die Platzierung einer Gruppe von untereinander abhängigen Instances beeinflussen, um den Anforderungen Ihres Workloads nachzukommen.

Syntax

```
tosca.nodes.AWS.Compute.PlacementGroup
  properties:
    strategy: String
    partition_count: Integer
    tags: List
```

Eigenschaften

`strategy`

Die Strategie, mit der Amazon EC2 EC2-Instances platziert werden sollen.

Erforderlich: Ja

Typ: Zeichenfolge

Mögliche Werte: CLUSTER | PARTITION | SPREAD_HOST | SPREAD_RACK

- CLUSTER — fasst Instanzen nahe beieinander in einer Availability Zone zusammen. Diese Strategie ermöglicht es Workloads, die Netzwerkleistung mit niedriger Latenz zu erreichen, die für eng gekoppelte node-to-node Kommunikation erforderlich ist, wie sie für HPC-Anwendungen (High Performance Computing) typisch ist.
- PARTITION — verteilt Ihre Instances auf logische Partitionen, sodass Gruppen von Instanzen in einer Partition die zugrunde liegende Hardware nicht gemeinsam mit Gruppen von Instanzen in verschiedenen Partitionen nutzen. Diese Strategie wird in der Regel für große verteilte und replizierte Workloads wie Hadoop, Cassandra und Kafka verwendet.
- SPREAD_RACK — platziert eine kleine Gruppe von Instanzen auf unterschiedlicher zugrunde liegender Hardware, um korrelierte Ausfälle zu reduzieren.
- SPREAD_HOST — wird nur mit Outpost-Platzierungsgruppen verwendet. Platziert eine kleine Gruppe von Instances auf unterschiedlicher zugrundeliegender Hardware, um korrelierte Ausfälle zu reduzieren.

`partition_count`

Die Anzahl an Partitionen.

Erforderlich: Nur erforderlich, wenn auf `PARTITION` gesetzt `strategy` ist.

Typ: Ganzzahl

Mögliche Werte: 1 | 2 | 3 | 4 | 5 | 6 | 7

`tags`

Die Tags, die Sie der Platzierungsgruppenressource zuordnen können.

Erforderlich: Nein

Typ: Liste

Beispiel

```
ExamplePlacementGroup:
```

```
  type: tosca.nodes.AWS.Compute.PlacementGroup
  properties:
    strategy: "PARTITION"
    partition_count: 5
    tags:
      - tag_key=tag_value
```

AWS. Berechnen. UserData

AWS TNB unterstützt das Starten von Amazon EC2 EC2-Instances mit benutzerdefinierten Benutzerdaten über den UserData Knoten im Network Service Descriptor (NSD). Weitere Informationen zu benutzerdefinierten Benutzerdaten finden Sie unter Benutzerdaten und Shell-Skripts im Amazon EC2 EC2-Benutzerhandbuch.

Während der Netzwerkinstanziierung stellt AWS TNB die Amazon EC2 EC2-Instance-Registrierung für den Cluster über ein Benutzerdatenskript bereit. Wenn auch benutzerdefinierte Benutzerdaten bereitgestellt werden, führt AWS TNB beide Skripts zusammen und gibt sie als Multimime-Skript an Amazon EC2 weiter. Das benutzerdefinierte Benutzerdatenskript wird vor dem Amazon EKS-Registrierungsskript ausgeführt.

Um benutzerdefinierte Variablen im Benutzerdatenskript zu verwenden, fügen Sie `!` nach der geöffneten geschweiften Klammer ein Ausrufezeichen hinzu. `{` Um es beispielsweise `MyVariable` im Skript zu verwenden, geben Sie Folgendes ein: `{!MyVariable}`

> **Note**
>
> - AWS TNB unterstützt Benutzerdatenskripten mit einer Größe von bis zu 7 KB.
> - Da AWS TNB das `multimime` Benutzerdatenskript verarbeitet und rendert, sollten Sie sicherstellen, dass das Skript alle Regeln einhält. AWS CloudFormation AWS CloudFormation

Syntax

```
tosca.nodes.AWS.Compute.UserData:
  properties:
    implementation: String
    content_type: String
```

Eigenschaften

`implementation`

Der relative Pfad zur Benutzerdatenskriptdefinition. Das Format muss wie folgt sein: `./scripts/script_name.sh`

Erforderlich: Ja

Typ: Zeichenfolge

`content_type`

Inhaltstyp des Benutzerdatenskripts.

Erforderlich: Ja

Typ: Zeichenfolge

Mögliche Werte: `x-shellscript`

Beispiel

```
ExampleUserData:
  type: tosca.nodes.AWS.Compute.UserData
  properties:
    content_type: "text/x-shellscript"
    implementation: "./scripts/customUserData.sh"
```

AWS.Netzwerke. SecurityGroup

AWS TNB unterstützt Sicherheitsgruppen, um die Bereitstellung von Amazon EC2-Sicherheitsgruppen zu automatisieren, die Sie Amazon EKS Kubernetes-Cluster-Knotengruppen zuordnen können.

Syntax

```
tosca.nodes.AWS.Networking.SecurityGroup
  properties:
    description: String
    name: String
```

```
    tags: List
  requirements:
    vpc: String
```

Eigenschaften

`description`

Die Beschreibung der Sicherheitsgruppe. Sie können bis zu 255 Zeichen verwenden, um die Gruppe zu beschreiben. Sie können nur Buchstaben (A-Z und a-z), Zahlen (0-9), Leerzeichen und die folgenden Sonderzeichen verwenden: ._-:/() #, @ [] +=&; {}! $*

Erforderlich: Ja

Typ: Zeichenfolge

`name`

Ein Name für die Sicherheitsgruppe. Sie können bis zu 255 Zeichen für den Namen verwenden. Sie können nur Buchstaben (A-Z und a-z), Zahlen (0-9), Leerzeichen und die folgenden Sonderzeichen verwenden: ._-:/() #, @ [] +=&; {}! $*

Erforderlich: Ja

Typ: Zeichenfolge

`tags`

Die Tags, die Sie an die Sicherheitsgruppenressource anhängen können.

Erforderlich: Nein

Typ: Liste

Voraussetzungen

`vpc`

Ein AWS.networking.VPC-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
SampleSecurityGroup001:
  type: tosca.nodes.AWS.Networking.SecurityGroup
  properties:
        description: "Sample Security Group for Testing"
        name: "SampleSecurityGroup"
        tags:
          - "Name=SecurityGroup"
          - "Environment=Testing"
      requirements:
        vpc: SampleVPC
```

AWS.Netzwerke. SecurityGroupEgressRule

AWS TNB unterstützt Ausgangsregeln für Sicherheitsgruppen, um die Bereitstellung von Amazon EC2 Security Group Egress Rules zu automatisieren, die an .Networking angehängt werden können. AWS SecurityGroup. Beachten Sie, dass Sie eine cidr_ip/destination_security_group/ destination_prefix_list als Ziel für ausgehenden Datenverkehr angeben müssen.

Syntax

```
AWS.Networking.SecurityGroupEgressRule
  properties:
    ip_protocol: String
    from_port: Integer
    to_port: Integer
    description: String
    destination_prefix_list: String
    cidr_ip: String
    cidr_ipv6: String
  requirements:
    security_group: String
    destination_security_group: String
```

Eigenschaften

`cidr_ip`

Der IPv4-Adressbereich im CIDR-Format. Sie müssen einen CIDR-Bereich angeben, der ausgehenden Datenverkehr zulässt.

Erforderlich: Nein

Typ: Zeichenfolge

`cidr_ipv6`

Der IPv6-Adressbereich im CIDR-Format für ausgehenden Verkehr. Sie müssen eine Zielsicherheitsgruppe (`destination_security_group` oder `destination_prefix_list`) oder einen CIDR-Bereich (`cidr_ip` oder `cidr_ipv6`) angeben.

Erforderlich: Nein

Typ: Zeichenfolge

`description`

Die Beschreibung einer Sicherheitsgruppenregel für ausgehenden Datenverkehr. Sie können bis zu 255 Zeichen verwenden, um die Regel zu beschreiben.

Erforderlich: Nein

Typ: Zeichenfolge

`destination_prefix_list`

Die Präfixlisten-ID einer bestehenden von Amazon VPC verwalteten Präfixliste. Dies ist das Ziel von Knotengruppen-Instances, die der Sicherheitsgruppe zugeordnet sind. Weitere Informationen zu verwalteten Präfixlisten finden Sie unter Verwaltete Präfixlisten im Amazon VPC-Benutzerhandbuch.

Erforderlich: Nein

Typ: Zeichenfolge

`from_port`

Wenn das Protokoll TCP oder UDP ist, ist dies der Anfang des Portbereichs. Wenn das Protokoll ICMP oder ICMPv6 ist, ist dies die Typnummer. Ein Wert von -1 gibt alle ICMP-/ICMPv6-Typen an. Wenn Sie alle ICMP-/ICMPv6-Typen angeben, müssen Sie alle ICMP-/ICMPv6-Codes angeben.

Erforderlich: Nein

Typ: Ganzzahl

`ip_protocol`

Der IP-Protokollname (tcp, udp, icmp, icmpv6) oder die Protokollnummer. Verwenden Sie -1, um alle Protokolle anzugeben. Bei der Autorisierung von Sicherheitsgruppenregeln ermöglicht die Angabe von -1 oder einer anderen Protokollnummer als tcp, udp, icmp oder icmpv6 den Verkehr auf allen Ports, unabhängig vom angegebenen Portbereich. Für tcp, udp und icmp müssen Sie einen Portbereich angeben. Für icmpv6 ist der Portbereich optional. Wenn Sie den Portbereich weglassen, ist Verkehr für alle Typen und Codes zulässig.

Erforderlich: Ja

Typ: Zeichenfolge

`to_port`

Wenn das Protokoll TCP oder UDP ist, ist dies das Ende des Portbereichs. Wenn das Protokoll ICMP oder ICMPv6 ist, ist dies der Code. Ein Wert von -1 gibt alle ICMP-/ICMPv6-Codes an. Wenn Sie alle ICMP-/ICMPv6-Typen angeben, müssen Sie alle ICMP-/ICMPv6-Codes angeben.

Erforderlich: Nein

Typ: Ganzzahl

Voraussetzungen

`security_group`

Die ID der Sicherheitsgruppe, zu der diese Regel hinzugefügt werden soll.

Erforderlich: Ja

Typ: Zeichenfolge

`destination_security_group`

Die ID oder TOSCA-Referenz der Zielsicherheitsgruppe, zu der ausgehender Datenverkehr zugelassen ist.

Erforderlich: Nein

Typ: Zeichenfolge

Beispiel

```
SampleSecurityGroupEgressRule:
      type: tosca.nodes.AWS.Networking.SecurityGroupEgressRule
      properties:
        ip_protocol: "tcp"
        from_port: 8000
        to_port: 9000
        description: "Egress Rule for sample security group"
        cidr_ipv6: "2600:1f14:3758:ca00::/64"
      requirements:
        security_group: SampleSecurityGroup001
        destination_security_group: SampleSecurityGroup002
```

AWS.Netzwerke. SecurityGroupIngressRule

AWS TNB unterstützt Ingress-Regeln für Sicherheitsgruppen, um die Bereitstellung von Amazon EC2 Security Group Ingress Rules zu automatisieren, die an .Networking angehängt werden können. AWS SecurityGroup. Beachten Sie, dass Sie eine cidr_ip/source_security_group/source_prefix_list als Quelle für eingehenden Datenverkehr angeben müssen.

Syntax

```
AWS.Networking.SecurityGroupIngressRule
  properties:
    ip_protocol: String
    from_port: Integer
    to_port: Integer
    description: String
    source_prefix_list: String
    cidr_ip: String
    cidr_ipv6: String
  requirements:
    security_group: String
    source_security_group: String
```

Eigenschaften

`cidr_ip`

Der IPv4-Adressbereich im CIDR-Format. Sie müssen einen CIDR-Bereich angeben, der eingehenden Datenverkehr zulässt.

Erforderlich: Nein

Typ: Zeichenfolge

`cidr_ipv6`

Der IPv6-Adressbereich im CIDR-Format für eingehenden Datenverkehr. Sie müssen eine Quellsicherheitsgruppe (`source_security_group` oder `source_prefix_list`) oder einen CIDR-Bereich (`cidr_ip` oder `cidr_ipv6`) angeben.

Erforderlich: Nein

Typ: Zeichenfolge

`description`

Die Beschreibung einer Sicherheitsgruppenregel für eingehenden (eingehenden) Datenverkehr. Sie können bis zu 255 Zeichen verwenden, um die Regel zu beschreiben.

Erforderlich: Nein

Typ: Zeichenfolge

`source_prefix_list`

Die Präfixlisten-ID einer bestehenden von Amazon VPC verwalteten Präfixliste. Dies ist die Quelle, von der Knotengruppen-Instances, die der Sicherheitsgruppe zugeordnet sind, Datenverkehr empfangen dürfen. Weitere Informationen zu verwalteten Präfixlisten finden Sie unter Verwaltete Präfixlisten im Amazon VPC-Benutzerhandbuch.

Erforderlich: Nein

Typ: Zeichenfolge

`from_port`

Wenn das Protokoll TCP oder UDP ist, ist dies der Anfang des Portbereichs. Wenn das Protokoll ICMP oder ICMPv6 ist, ist dies die Typnummer. Ein Wert von -1 gibt alle ICMP-/ICMPv6-Typen

an. Wenn Sie alle ICMP-/ICMPv6-Typen angeben, müssen Sie alle ICMP-/ICMPv6-Codes angeben.

Erforderlich: Nein

Typ: Ganzzahl

`ip_protocol`

Der IP-Protokollname (tcp, udp, icmp, icmpv6) oder die Protokollnummer. Verwenden Sie -1, um alle Protokolle anzugeben. Bei der Autorisierung von Sicherheitsgruppenregeln ermöglicht die Angabe von -1 oder einer anderen Protokollnummer als tcp, udp, icmp oder icmpv6 den Verkehr auf allen Ports, unabhängig vom angegebenen Portbereich. Für tcp, udp und icmp müssen Sie einen Portbereich angeben. Für icmpv6 ist der Portbereich optional. Wenn Sie den Portbereich weglassen, ist Verkehr für alle Typen und Codes zulässig.

Erforderlich: Ja

Typ: Zeichenfolge

`to_port`

Wenn das Protokoll TCP oder UDP ist, ist dies das Ende des Portbereichs. Wenn das Protokoll ICMP oder ICMPv6 ist, ist dies der Code. Ein Wert von -1 gibt alle ICMP-/ICMPv6-Codes an. Wenn Sie alle ICMP-/ICMPv6-Typen angeben, müssen Sie alle ICMP-/ICMPv6-Codes angeben.

Erforderlich: Nein

Typ: Ganzzahl

Voraussetzungen

`security_group`

Die ID der Sicherheitsgruppe, zu der diese Regel hinzugefügt werden soll.

Erforderlich: Ja

Typ: Zeichenfolge

`source_security_group`

Die ID oder TOSCA-Referenz der Quellsicherheitsgruppe, von der eingehender Datenverkehr zugelassen werden soll.

Erforderlich: Nein

Typ: Zeichenfolge

Beispiel

```
SampleSecurityGroupIngressRule:
      type: tosca.nodes.AWS.Networking.SecurityGroupIngressRule
      properties:
        ip_protocol: "tcp"
        from_port: 8000
        to_port: 9000
        description: "Ingress Rule for free5GC cluster on IPv6"
        cidr_ipv6: "2600:1f14:3758:ca00::/64"
      requirements:
        security_group: SampleSecurityGroup1
        source_security_group: SampleSecurityGroup2
```

AWS.Ressource.Import

Sie können die folgenden AWS Ressourcen in TNB importieren: AWS

- VPC
- Subnetz
- Routing-Tabelle
- Internet Gateway
- Sicherheitsgruppe

Syntax

```
tosca.nodes.AWS.Resource.Import
  properties:
    resource_type: String
    resource_id: String
```

Eigenschaften

`resource_type`

Der Ressourcentyp, der in AWS TNB importiert wird.

Erforderlich: Nein

Typ: Liste

`resource_id`

Die Ressourcen-ID, die in AWS TNB importiert wird.

Erforderlich: Nein

Typ: Liste

Beispiel

```
SampleImportedVPC
  type: tosca.nodes.AWS.Resource.Import
  properties:
    resource_type: "tosca.nodes.AWS.Networking.VPC"
    resource_id: "vpc-123456"
```

AWS.networking.ENI

Eine Netzwerkschnittstelle ist eine logische Netzwerkkomponente in einer VPC, die eine virtuelle Netzwerkkarte darstellt. Einer Netzwerkschnittstelle wird anhand ihres Subnetzes entweder automatisch oder manuell eine IP-Adresse zugewiesen. Nachdem Sie eine Amazon EC2 EC2-Instance in einem Subnetz bereitgestellt haben, können Sie ihr eine Netzwerkschnittstelle hinzufügen oder eine Netzwerkschnittstelle von dieser Amazon EC2 EC2-Instance trennen und erneut eine Verbindung zu einer anderen Amazon EC2 EC2-Instance in diesem Subnetz herstellen. Der Geräteindex identifiziert die Position in der Reihenfolge der Dateianhänge.

Syntax

```
tosca.nodes.AWS.Networking.ENI:
  properties:
    device_index: Integer
```

```
    source_dest_check: Boolean
    tags: List
  requirements:
    subnet: String
    security_groups: List
```

Eigenschaften

`device_index`

Der Geräteindex muss größer als Null sein.

Erforderlich: Ja

Typ: Ganzzahl

`source_dest_check`

Gibt an, ob die Netzwerkschnittstelle eine Quell-/Zielüberprüfung durchführt. Der Wert `true` bedeutet, dass die Prüfung aktiviert ist, und der Wert `false` bedeutet, dass die Prüfung deaktiviert ist.

Zulässiger Wert: wahr, falsch

Standard: true

Erforderlich: Nein

Typ: Boolesch

`tags`

Die Tags, die an die Ressource angehängt werden sollen.

Erforderlich: Nein

Typ: Liste

Voraussetzungen

`subnet`

Ein AWS.Networking.Subnet-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

`security_groups`

Ein AWS.Networking. SecurityGroupKnoten.

Erforderlich: Nein

Typ: Zeichenfolge

Beispiel

```
SampleENI:
  type: tosca.nodes.AWS.Networking.ENI
  properties:
    device_index: 5
    source_dest_check: true
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  requirements:
    subnet: SampleSubnet
    security_groups:
      - SampleSecurityGroup01
      - SampleSecurityGroup02
```

AWS.HookExecution

Ein Lifecycle-Hook bietet Ihnen die Möglichkeit, Ihre eigenen Skripts als Teil Ihrer Infrastruktur und Netzwerkinstanziierung auszuführen.

Syntax

```
tosca.nodes.AWS.HookExecution:
  capabilities:
    execution:
      properties:
        type: String
  requirements:
    definition: String
```

```
    vpc: String
```

Funktionen

`execution`

Eigenschaften für die Hook-Ausführungs-Engine, die die Hook-Skripte ausführt.

`type`

Der Typ der Hook-Ausführungs-Engine.

Erforderlich: Nein

Typ: Zeichenfolge

Mögliche Werte: `CODE_BUILD`

Voraussetzungen

`definition`

Ein AWS. HookDefinition.Bash-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

`vpc`

Ein AWS.Networking.VPC-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
SampleHookExecution:
  type: tosca.nodes.AWS.HookExecution
  requirements:
    definition: SampleHookScript
    vpc: SampleVPC
```

AWS.Netzwerke. InternetGateway

Definiert einen AWS Internet-Gateway-Knoten.

Syntax

```
tosca.nodes.AWS.Networking.InternetGateway:
  capabilities:
    routing:
      properties:
        dest_cidr: String
        ipv6_dest_cidr: String
  properties:
    tags: List
    egress_only: Boolean
  requirements:
    vpc: String
    route_table: String
```

Funktionen

`routing`

Eigenschaften, die die Routing-Verbindung innerhalb der VPC definieren. Sie müssen entweder die `ipv6_dest_cidr` Eigenschaft `dest_cidr` oder angeben.

`dest_cidr`

Der IPv4-CIDR-Block, der für die Zielabgleiche verwendet wird. Diese Eigenschaft wird verwendet, um eine Route in zu erstellen, `RouteTable` und ihr Wert wird als`DestinationCidrBlock`.

Erforderlich: Nein, wenn Sie die `ipv6_dest_cidr` Eigenschaft angegeben haben.

Typ: Zeichenfolge

`ipv6_dest_cidr`

Der IPv6-CIDR-Block, der für die Zielabgleiche verwendet wird.

Erforderlich: Nein, wenn Sie die `dest_cidr` Immobilie aufgenommen haben.

Typ: Zeichenfolge

Eigenschaften

`tags`

Die Tags, die an die Ressource angehängt werden sollen.

Erforderlich: Nein

Typ: Liste

`egress_only`

Eine IPv6-spezifische Eigenschaft. Gibt an, ob das Internet-Gateway nur für die ausgehende Kommunikation vorgesehen ist oder nicht. Wenn wahr `egress_only` ist, müssen Sie die `ipv6_dest_cidr` Eigenschaft definieren.

Erforderlich: Nein

Typ: Boolesch

Voraussetzungen

`vpc`

Ein AWS.networking.VPC-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

`route_table`

Ein AWS.Networking. RouteTableKnoten.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
Free5GCIGW:
  type: tosca.nodes.AWS.Networking.InternetGateway
  properties:
```

```
    egress_only: false
  capabilities:
    routing:
      properties:
        dest_cidr: "0.0.0.0/0"
        ipv6_dest_cidr: "::/0"
  requirements:
    route_table: Free5GCRouteTable
    vpc: Free5GCVPC
Free5GCEGW:
  type: tosca.nodes.AWS.Networking.InternetGateway
  properties:
    egress_only: true
  capabilities:
    routing:
      properties:
        ipv6_dest_cidr: "::/0"
  requirements:
    route_table: Free5GCPrivateRouteTable
    vpc: Free5GCVPC
```

AWS. Netzwerke. RouteTable

Eine Routentabelle enthält eine Reihe von Regeln, die als Routen bezeichnet werden und bestimmen, wohin der Netzwerkverkehr von Subnetzen innerhalb Ihrer VPC oder Ihres Gateways geleitet wird. Sie müssen einer VPC eine Routing-Tabelle zuordnen.

Syntax

```
tosca.nodes.AWS.Networking.RouteTable:
  properties:
    tags: List
  requirements:
    vpc: String
```

Eigenschaften

`tags`

Tags, die an die Ressource angehängt werden sollen.

Erforderlich: Nein

Typ: Liste

Voraussetzungen

`vpc`

Ein AWS.networking.VPC-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
SampleRouteTable:
  type: tosca.nodes.AWS.Networking.RouteTable
  properties:
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  requirements:
    vpc: SampleVPC
```

AWS.Networking.Subnet

Ein Subnetz ist ein Bereich von IP-Adressen in Ihrer VPC, der sich vollständig innerhalb einer Availability Zone befinden muss. Sie müssen eine VPC, einen CIDR-Block, eine Availability Zone und eine Routing-Tabelle für Ihr Subnetz angeben. Sie müssen auch definieren, ob Ihr Subnetz privat oder öffentlich ist.

Syntax

```
tosca.nodes.AWS.Networking.Subnet:
  properties:
    type: String
    availability_zone: String
    cidr_block: String
    ipv6_cidr_block: String
    ipv6_cidr_block_suffix: String
    outpost_arn: String
    tags: List
```

```
  requirements:
    vpc: String
    route_table: String
```

Eigenschaften

`type`

Gibt an, ob in diesem Subnetz gestartete Instances eine öffentliche IPv4-Adresse erhalten.

Erforderlich: Ja

Typ: Zeichenfolge

Mögliche Werte: `PUBLIC` | `PRIVATE`

`availability_zone`

Die Availability Zone für das Subnetz. Dieses Feld unterstützt AWS Availability Zones innerhalb einer AWS Region, zum Beispiel `us-west-2` (USA West (Oregon)). Es unterstützt beispielsweise auch AWS Local Zones innerhalb der Availability Zone`us-west-2-lax-1a`.

Erforderlich: Ja

Typ: Zeichenfolge

`cidr_block`

Der CIDR-Block für das Subnetz.

Erforderlich: Nein

Typ: Zeichenfolge

`ipv6_cidr_block`

Der CIDR-Block, der zur Erstellung des IPv6-Subnetzes verwendet wurde. Wenn Sie diese Eigenschaft angeben, schließen Sie sie nicht ein. `ipv6_cidr_block_suffix`

Erforderlich: Nein

Typ: Zeichenfolge

`ipv6_cidr_block_suffix`

Das zweistellige hexadezimale Suffix des IPv6-CIDR-Blocks für das über Amazon VPC erstellte Subnetz. Verwenden Sie das folgende Format: *`2-digit hexadecimal`*`::/`*`subnetMask`*

Wenn Sie diese Eigenschaft angeben, schließen Sie sie nicht ein. `ipv6_cidr_block`

Erforderlich: Nein

Typ: Zeichenfolge

`outpost_arn`

Der ARN, in dem AWS Outposts das Subnetz erstellt wird. Fügen Sie diese Eigenschaft zur NSD-Vorlage hinzu, wenn Sie selbstverwaltete Amazon EKS-Knoten auf starten möchten. AWS Outposts Weitere Informationen finden Sie unter Amazon EKS on AWS Outposts im Amazon EKS-Benutzerhandbuch.

Wenn Sie diese Eigenschaft zur NSD-Vorlage hinzufügen, müssen Sie den Wert für die `availability_zone` Eigenschaft auf die Availability Zone von festlegen. AWS Outposts

Erforderlich: Nein

Typ: Zeichenfolge

`tags`

Die Tags, die an die Ressource angehängt werden sollen.

Erforderlich: Nein

Typ: Liste

Voraussetzungen

`vpc`

Ein AWS.networking.VPC-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

`route_table`

Ein AWS.Networking. RouteTableKnoten.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
SampleSubnet01:
  type: tosca.nodes.AWS.Networking.Subnet
  properties:
    type: "PUBLIC"
    availability_zone: "us-east-1a"
    cidr_block: "10.100.50.0/24"
    ipv6_cidr_block_suffix: "aa::/64"
    outpost_arn: "arn:aws:outposts:region:accountId:outpost/op-11223344EXAMPLE"
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  requirements:
    vpc: SampleVPC
    route_table: SampleRouteTable

SampleSubnet02:
  type: tosca.nodes.AWS.Networking.Subnet
  properties:
    type: "PUBLIC"
    availability_zone: "us-west-2b"
    cidr_block: "10.100.50.0/24"
    ipv6_cidr_block: "2600:1f14:3758:ca00::/64"
  requirements:
    route_table: SampleRouteTable
    vpc: SampleVPC
```

AWS.deployment.vnfDeployment

NF-Bereitstellungen werden modelliert, indem die Infrastruktur und die damit verbundene Anwendung bereitgestellt werden. Das Clusterattribut gibt den EKS-Cluster an, der Ihre NFs hostet. Das vnfs-Attribut gibt die Netzwerkfunktionen für Ihre Bereitstellung an. Sie können auch optionale Lifecycle-Hook-Operationen vom Typ pre_create und post_create bereitstellen, um Anweisungen auszuführen, die für Ihre Bereitstellung spezifisch sind, z. B. das Aufrufen einer API für das Inventarverwaltungssystem.

Syntax

```
tosca.nodes.AWS.Deployment.VNFDeployment:
  requirements:
    deployment: String
```

```
    cluster: String
    vnfs: List
  interfaces:
    Hook:
      pre_create: String
      post_create: String
```

Voraussetzungen

`deployment`

Ein .deployment.vnfDeployment-Knoten.AWS

Erforderlich: Nein

Typ: Zeichenfolge

`cluster`

Ein AWS.compute.EKS-Knoten.

Erforderlich: Ja

Typ: Zeichenfolge

`vnfs`

Ein .VNF-Knoten.AWS

Erforderlich: Ja

Typ: Zeichenfolge

Schnittstellen

Haken

Definiert die Phase, in der Lifecycle-Hooks ausgeführt werden.

`pre_create`

Ein AWS. HookExecutionKnoten. Dieser Hook wird ausgeführt, bevor der `VNFDeployment` Knoten bereitgestellt wird.

Erforderlich: Nein

Typ: Zeichenfolge

`post_create`

Ein AWS. HookExecutionKnoten. Dieser Hook wird ausgeführt, nachdem der `VNFDeployment` Knoten bereitgestellt wurde.

Erforderlich: Nein

Typ: Zeichenfolge

Beispiel

```
SampleHelmDeploy:
  type: tosca.nodes.AWS.Deployment.VNFDeployment
  requirements:
    deployment: SampleHelmDeploy2
    cluster: SampleEKS
    vnfs:
      - vnf.SampleVNF
  interfaces:
    Hook:
      pre_create: SampleHook
```

AWS.networking.VPC

Sie müssen einen CIDR-Block für Ihre Virtual Private Cloud (VPC) angeben.

Syntax

```
tosca.nodes.AWS.Networking.VPC:
  properties:
    cidr_block: String
    ipv6_cidr_block: String
    dns_support: String
    tags: List
```

Eigenschaften

`cidr_block`

Der IPv4-Netzwerkbereich für die VPC in CIDR-Notation.

Erforderlich: Ja

Typ: Zeichenfolge

`ipv6_cidr_block`

Der IPv6-CIDR-Block, der zur Erstellung der VPC verwendet wurde.

Zulässiger Wert: `AMAZON_PROVIDED`

Erforderlich: Nein

Typ: Zeichenfolge

`dns_support`

Gibt an, ob die in VPC gestarteten Instances DNS-Hostnamen erhalten.

Erforderlich: Nein

Typ: Boolesch

Standard: `false`

`tags`

Tags, die an die Ressource angehängt werden sollen.

Erforderlich: Nein

Typ: Liste

Beispiel

```
SampleVPC:
  type: tosca.nodes.AWS.Networking.VPC
  properties:
    cidr_block: "10.100.0.0/16"
    ipv6_cidr_block: "AMAZON_PROVIDED"
    dns_support: true
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
```

AWS.networking.NatGateway

Sie können einen öffentlichen oder privaten NAT-Gateway-Knoten über ein Subnetz definieren. Wenn Sie bei einem öffentlichen Gateway keine Elastic IP-Zuweisungs-ID angeben, weist AWS TNB Ihrem Konto eine Elastic IP zu und ordnet diese dem Gateway zu.

Syntax

```
tosca.nodes.AWS.Networking.NATGateway:
  requirements:
    subnet: String
    internet_gateway: String
  properties:
    type: String
    eip_allocation_id: String
    tags: List
```

Eigenschaften

`subnet`

Die Node-Referenz AWS.Networking.Subnet.

Erforderlich: Ja

Typ: Zeichenfolge

`internet_gateway`

Das AWS.Networking. InternetGatewayKnotenreferenz.

Erforderlich: Ja

Typ: Zeichenfolge

Eigenschaften

`type`

Gibt an, ob das Gateway öffentlich oder privat ist.

Zulässiger Wert:`PUBLIC`, `PRIVATE`

Erforderlich: Ja

Typ: Zeichenfolge

`eip_allocation_id`

Die ID, die die Zuweisung der Elastic IP-Adresse darstellt.

Erforderlich: Nein

Typ: Zeichenfolge

`tags`

Tags, die an die Ressource angehängt werden sollen.

Erforderlich: Nein

Typ: Liste

Beispiel

```
Free5GCNatGateway01:
  type: tosca.nodes.AWS.Networking.NATGateway
    requirements:
       subnet: Free5GCSubnet01
       internet_gateway: Free5GCIGW
    properties:
       type: PUBLIC
       eip_allocation_id: eipalloc-12345
```

AWS.Netzwerking.Route

Sie können einen Routenknoten definieren, der die Zielroute dem NAT-Gateway als Zielressource zuordnet und die Route der zugehörigen Routentabelle hinzufügt.

Syntax

```
tosca.nodes.AWS.Networking.Route:
  properties:
    dest_cidr_blocks: List
  requirements:
```

```
    nat_gateway: String
    route_table: String
```

Eigenschaften

`dest_cidr_blocks`

Die Liste der IPv4-Zielrouten zur Zielressource.

Erforderlich: Ja

Typ: Liste

Mitgliedstyp: Zeichenfolge

Eigenschaften

`nat_gateway`

Die AWS Knotenreferenz .networking.NatGateway.

Erforderlich: Ja

Typ: Zeichenfolge

`route_table`

Das AWS.Networking. RouteTableKnotenreferenz.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
Free5GCRoute:
  type: tosca.nodes.AWS.Networking.Route
  properties:
    dest_cidr_blocks:
      - 0.0.0.0/0
      - 10.0.0.0/28
  requirements:
```

```
    nat_gateway: Free5GCNatGateway01
    route_table: Free5GCRouteTable
```

Gemeinsame Knoten

Definiert Knoten, die in NSD und VNFD verwendet werden sollen.

- AWS.HookDefinition.Bash

AWS.HookDefinition.Bash

Definiert eineAWS HookDefinitioninbash.

Syntax

```
tosca.nodes.AWS.HookDefinition.Bash:
  properties:
    implementation: String
    environment_variables: List
    execution_role: String
```

Eigenschaften

`implementation`

Der relative Pfad zur Hook-Definition. Das Format muss wie folgt lauten:`./hooks/`*`script_name`*`.sh`

Erforderlich: Ja

Typ: Zeichenfolge

`environment_variables`

Die Umgebungsvariablen für das Hook-Bash-Skript. Verwenden Sie das folgende Format:**`envName`**=**`envValue`**mit dem folgenden Regex:`^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`

Stellen Sie sicher, dass die**`envName`**=**`envValue`**Wert erfüllt die folgenden Kriterien:

- Verwenden Sie keine Leerzeichen.

- Beginne**envName**mit einem Buchstaben (A-Z oder a-z) oder einer Zahl (0-9).
- Starten Sie den Namen der Umgebungsvariablen nicht mit dem folgendenAWSReservierte TNB-Schlüsselwörter (Groß- und Kleinschreibung wird nicht berücksichtigt):
  - CODEBUILD
  - TNB
  - ZUHAUSE
  - AWS
- Sie können eine beliebige Anzahl von Buchstaben (A-Z oder a-z), Zahlen (0-9) und Sonderzeichen verwenden-und_zum**envName**und**envValue**.

Beispiel: `A123-45xYz=Example_789`

Required: No

Typ: Liste

`execution_role`

Die Rolle der Hook-Ausführung.

Erforderlich: Ja

Typ: Zeichenfolge

Beispiel

```
SampleHookScript:
  type: tosca.nodes.AWS.HookDefinition.Bash
  properties:
    implementation: "./hooks/myhook.sh"
    environment_variables:
      - "variable01=value01"
      - "variable02=value02"
    execution_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleHookPermission"
```

Sicherheit in AWS Telco Network Builder

Cloud-Sicherheit hat AWS höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für AWS Telco Network Builder gelten, finden Sie unter AWS Services im Bereich nach Compliance-Programm AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung von AWS TNB anwenden können. In den folgenden Themen erfahren Sie, wie Sie AWS TNB konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer AWS TNB-Ressourcen unterstützen.

Inhalt

Datenschutz in TNB AWS

Das Modell der AWS gemeinsamen Verantwortung und gilt für den Datenschutz in AWS Telco Network Builder. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Systeme laufen. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit AWS TNB oder anderen Geräten arbeiten und die Konsole, die AWS-Services API oder SDKs verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Umgang mit Daten

Wenn Sie Ihr AWS Konto schließen, markiert AWS TNB Ihre Daten zum Löschen und entfernt sie für jegliche Verwendung. Wenn Sie Ihr AWS Konto innerhalb von 90 Tagen reaktivieren, stellt AWS TNB Ihre Daten wieder her. Nach 120 Tagen löscht AWS TNB Ihre Daten dauerhaft. AWS TNB terminiert auch Ihre Netzwerke und löscht Ihre Funktionspakete und Ihre Netzwerkpakete.

Verschlüsselung im Ruhezustand

AWS TNB verschlüsselt immer alle im Dienst gespeicherten Daten im Ruhezustand, ohne dass eine zusätzliche Konfiguration erforderlich ist. Diese Verschlüsselung erfolgt automatisch. AWS Key Management Service

Verschlüsselung während der Übertragung

AWS TNB schützt alle Daten während der Übertragung mit Transport Layer Security (TLS) 1.2.

Es liegt in Ihrer Verantwortung, Daten zwischen Ihren Simulationsagenten und deren Clients zu verschlüsseln.

Datenschutz für den Datenverkehr zwischen Netzwerken

AWS TNB-Rechenressourcen befinden sich in einer Virtual Private Cloud (VPC), die von allen Kunden gemeinsam genutzt wird. Der gesamte interne AWS TNB-Verkehr blieb im AWS Netzwerk und durchquert nicht das Internet. Verbindungen zwischen Ihren Simulationsagenten und ihren Clients werden über das Internet geleitet.

Identitäts- und Zugriffsmanagement für TNB AWS

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um TNB-Ressourcen zu verwenden AWS . IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Inhalt

- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- So funktioniert AWS Telco Network Builder mit IAM
- Beispiele für identitätsbasierte Richtlinien für AWS Telco Network Builder
- Fehlerbehebung bei Identität und Zugriff bei AWS Telco Network Builder

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in AWS TNB ausführen.

Dienstbenutzer — Wenn Sie den AWS TNB-Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr AWS TNB-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Fuktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie in AWS TNB nicht auf eine Funktion zugreifen können, finden Sie weitere Informationen unter. Fehlerbehebung bei Identität und Zugriff bei AWS Telco Network Builder

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die AWS TNB-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff AWS auf TNB. Es ist Ihre Aufgabe, zu bestimmen, auf welche AWS TNB-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 darüber, wie Ihr Unternehmen IAM mit AWS TNB nutzen kann, finden Sie unter. So funktioniert AWS Telco Network Builder mit IAM

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf TNB schreiben können. AWS Beispiele für identitätsbasierte AWS TNB-Richtlinien, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Telco Network Builder

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.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise 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 könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
- Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
- Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

- 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 Zugriff auf was 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.

So funktioniert AWS Telco Network Builder mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf AWS TNB verwenden, sollten Sie sich darüber informieren, welche IAM-Funktionen mit TNB verwendet werden können. AWS

IAM-Funktionen, die Sie mit Telco Network Builder verwenden können AWS

IAM-Feature	AWS TNB-Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein

IAM-Feature	AWS TNB-Unterstützung
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie AWS TNB und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im AWS IAM-Benutzerhandbuch unter Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für TNB AWS

Unterstützt Richtlinien auf Identitätsbasis.	Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet

ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für TNB AWS

Beispiele für identitätsbasierte AWS TNB-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Telco Network Builder

Ressourcenbasierte Richtlinien innerhalb von TNB AWS

Unterstützt ressourcenbasierte Richtlinien	Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden im IAM-Benutzerhandbuch.

Politische Maßnahmen für TNB AWS

Unterstützt Richtlinienaktionen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der AWS TNB-Aktionen finden Sie unter Von AWS Telco Network Builder definierte Aktionen in der Service Authorization Reference.

Richtlinienaktionen in AWS TNB verwenden vor der Aktion das folgende Präfix:

```
tnb
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "tnb:CreateSolFunctionPackage",
      "tnb:DeleteSolFunctionPackage"
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `List` beginnen, einschließlich der folgenden Aktion:

```
"Action": "tnb:List*"
```

Beispiele für identitätsbasierte AWS TNB-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Telco Network Builder

Politische Ressourcen für TNB AWS

Unterstützt Richtlinienressourcen	Ja

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": "*"
```

Eine Liste der AWS TNB-Ressourcentypen und ihrer ARNs finden Sie unter Von AWS Telco Network Builder definierte Ressourcen in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von AWS Telco Network Builder definierte Aktionen.

Beispiele für identitätsbasierte AWS TNB-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Telco Network Builder

Bedingungsschlüssel für Richtlinien für TNB AWS

Unterstützt servicespezifische Richtlini enbedingungsschlüssel	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der AWS TNB-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für AWS Telco Network Builder in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von AWS Telco Network Builder definierte Aktionen.

Beispiele für identitätsbasierte AWS TNB-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Telco Network Builder

ACLs in TNB AWS

Unterstützt ACLs	Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit TNB AWS

Unterstützt ABAC (Tags in Richtlinien)	Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie

können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Was ist ABAC? im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit TNB AWS

Unterstützt temporäre Anmeldeinformationen	Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln zu einer Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden

AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipalberechtigungen für TNB AWS

Unterstützt Forward Access Sessions (FAS)	Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für TNB AWS

Unterstützt Servicerollen	Nein

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

Mit Diensten verknüpfte Rollen für TNB AWS

Unterstützt serviceverknüpfte Rollen	Nein

Eine dienstgebundene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Beispiele für identitätsbasierte Richtlinien für AWS Telco Network Builder

Standardmäßig sind Benutzer und Rollen nicht berechtigt, TNB-Ressourcen zu erstellen oder zu ändern AWS . Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Einzelheiten zu den von AWS TNB definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden Ressourcentyp, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Telco Network Builder in der Service Authorization Reference.

Inhalt

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AWS TNB-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden

Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung 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 TNB-Konsole AWS

Um auf die AWS Telco Network Builder-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS TNB-Ressourcen in Ihrem aufzulisten und anzuzeigen. AWS-Konto Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Beispiele für Richtlinien für Servicerollen

Als Administrator besitzen und verwalten Sie die Ressourcen, die AWS TNB gemäß den Umgebungs- und Dienstvorlagen erstellt. Sie müssen Ihrem Konto IAM-Servicerollen zuordnen, damit AWS TNB Ressourcen für Ihr Netzwerk-Lebenszyklusmanagement erstellen kann.

Eine IAM-Servicerolle ermöglicht es AWS TNB, in Ihrem Namen Ressourcen aufzurufen, um Ihre Netzwerke zu instanziieren und zu verwalten. Wenn Sie eine Servicerolle angeben, verwendet AWS TNB die Anmeldeinformationen dieser Rolle.

Sie erstellen die Service-Rolle und die Berechtigungsrichtlinie mit dem IAM-Service. Weitere Informationen zum Erstellen einer Servicerolle finden Sie unter Creating a role to delegate permissions to an AWS service im IAM-Benutzerhandbuch.

AWS TNB-Servicerolle

Als Mitglied des Plattformteams können Sie als Administrator eine AWS TNB-Servicerolle erstellen und sie TNB zur Verfügung stellen. AWS Diese Rolle ermöglicht es AWS TNB, andere Dienste wie Amazon Elastic Kubernetes Service aufzurufen und AWS CloudFormation die erforderliche Infrastruktur für Ihr Netzwerk bereitzustellen und Netzwerkfunktionen bereitzustellen, wie in Ihrer NSD definiert.

Wir empfehlen Ihnen, die folgende IAM-Rollen- und Vertrauensrichtlinie für Ihre TNB-Servicerolle zu verwenden. AWS Denken Sie bei der Einschränkung der Zugriffsrechte für diese Richtlinie daran, dass AWS TNB bei Ressourcen, die nicht in Ihrer Richtlinie enthalten sind, möglicherweise mit dem Fehler „Zugriff verweigert“ fehlschlägt.

Der folgende Code zeigt eine AWS TNB-Servicerollenrichtlinie:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:GetCallerIdentity"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "AssumeRole"
        },
        {
            "Action": [
                "tnb:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBPolicy"
        },
        {
            "Action": [
                "iam:AddRoleToInstanceProfile",
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:TagInstanceProfile",
                "iam:UntagInstanceProfile"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "IAMPolicy"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "eks.amazonaws.com",
                        "eks-nodegroup.amazonaws.com"
                    ]
                }
            },
            "Action": [
```

```
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBAccessSLRPermissions"
        },
        {
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateOrUpdateTags",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteTags",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:DescribeTags",
                "autoscaling:UpdateAutoScalingGroup",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteLaunchTemplateVersions",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeTags",
                "ec2:GetLaunchTemplateData",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RunInstances",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:CreateInternetGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSubnet",
                "ec2:CreateTags",
                "ec2:CreateVpc",
                "ec2:DeleteInternetGateway",
                "ec2:DeleteNetworkInterface",
```

```
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSubnet",
                "ec2:DeleteTags",
                "ec2:DeleteVpc",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeInstances",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySecurityGroupRules",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:AllocateAddress",
                "ec2:AssignIpv6Addresses",
                "ec2:AssociateAddress",
                "ec2:AssociateNatGatewayAddress",
                "ec2:AssociateVpcCidrBlock",
                "ec2:CreateEgressOnlyInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:DeleteEgressOnlyInternetGateway",
                "ec2:DeleteNatGateway",
                "ec2:DescribeAddresses",
                "ec2:DescribeEgressOnlyInternetGateways",
                "ec2:DescribeNatGateways",
                "ec2:DisassociateAddress",
                "ec2:DisassociateNatGatewayAddress",
                "ec2:DisassociateVpcCidrBlock",
                "ec2:ReleaseAddress",
                "ec2:UnassignIpv6Addresses",
                "ec2:DescribeImages",
                "eks:CreateCluster",
                "eks:ListClusters",
                "eks:RegisterCluster",
                "eks:TagResource",
                "eks:DescribeAddonVersions",
                "events:DescribeRule",
                "iam:GetRole",
```

```
                "iam:ListAttachedRolePolicies",
                "iam:PassRole"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBAccessComputePerms"
        },
        {
            "Action": [
                "codebuild:BatchDeleteBuilds",
                "codebuild:BatchGetBuilds",
                "codebuild:CreateProject",
                "codebuild:DeleteProject",
                "codebuild:ListBuildsForProject",
                "codebuild:StartBuild",
                "codebuild:StopBuild",
                "events:DeleteRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "s3:CreateBucket",
                "s3:GetBucketAcl",
                "s3:GetObject",
                "eks:DescribeNodegroup",
                "eks:DeleteNodegroup",
                "eks:AssociateIdentityProviderConfig",
                "eks:CreateNodegroup",
                "eks:DeleteCluster",
                "eks:DeregisterCluster",
                "eks:UntagResource",
                "eks:DescribeCluster",
                "eks:ListNodegroups",
                "eks:CreateAddon",
                "eks:DeleteAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonVersions",
                "s3:PutObject",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateTerminationProtection"
            ],
            "Resource": [
```

```
                "arn:aws:events:*:*:rule/tnb*",
                "arn:aws:codebuild:*:*:project/tnb*",
                "arn:aws:logs:*:*:log-group:/aws/tnb*",
                "arn:aws:s3:::tnb*",
                "arn:aws:eks:*:*:addon/tnb*/*/*",
                "arn:aws:eks:*:*:cluster/tnb*",
                "arn:aws:eks:*:*:nodegroup/tnb*/tnb*/*",
                "arn:aws:cloudformation:*:*:stack/tnb*"
            ],
            "Effect": "Allow",
            "Sid": "TNBAccessInfraResourcePerms"
        },
        {
            "Sid": "CFNTemplatePerms",
            "Effect": "Allow",
            "Action": [
              "cloudformation:GetTemplateSummary"
            ],
            "Resource": "*"
         },
        {
            "Sid": "ImageAMISSMPerms",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": [
                "arn:aws:ssm:*::parameter/aws/service/eks/optimized-ami/*",
                "arn:aws:ssm:*::parameter/aws/service/bottlerocket/*"
            ]
        },
        {
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TaggingPolicy"
        },
        {
            "Action": [
                "outposts:GetOutpost"
            ],
            "Resource": "*",
```

```
            "Effect": "Allow",
            "Sid": "OutpostPolicy"
        }
    ]
}
```

Der folgende Code zeigt die Vertrauensrichtlinie für den AWS TNB-Dienst:

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

```
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

AWS TNB-Servicerolle für Amazon EKS-Cluster

Wenn Sie eine Amazon EKS-Ressource in Ihrem NSD erstellen, geben Sie das `cluster_role` Attribut an, um anzugeben, welche Rolle zur Erstellung Ihres Amazon EKS-Clusters verwendet wird.

Das folgende Beispiel zeigt eine AWS CloudFormation Vorlage, die eine AWS TNB-Servicerolle für die Amazon EKS-Cluster-Richtlinie erstellt.

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBEKSClusterRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBEKSClusterRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - eks.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSClusterPolicy"
```

Weitere Informationen zu IAM-Rollen mithilfe von AWS CloudFormation Vorlagen finden Sie in den folgenden Abschnitten des AWS CloudFormation Benutzerhandbuchs:

- AWS::IAM::Role
- Auswählen einer Stack-Vorlage

AWS TNB-Servicerolle für die Amazon EKS-Knotengruppe

Wenn Sie eine Amazon EKS-Knotengruppenressource in Ihrer NSD erstellen, geben Sie das `node_role` Attribut an, um anzugeben, welche Rolle zur Erstellung Ihrer Amazon EKS-Knotengruppe verwendet wird.

Das folgende Beispiel zeigt eine AWS CloudFormation Vorlage, die eine AWS TNB-Servicerolle für die Amazon EKS-Knotengruppenrichtlinie erstellt.

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBEKSNodeRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBEKSNodeRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSWorkerNodePolicy"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKS_CNI_Policy"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEBSCSIDriverPolicy"
      Policies:
        - PolicyName: EKSNodeRoleInlinePolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "logs:DescribeLogStreams"
                  - "logs:PutLogEvents"
                  - "logs:CreateLogGroup"
                  - "logs:CreateLogStream"
```

```
                Resource: "arn:aws:logs:*:*:log-group:/aws/tnb/tnb*"
        - PolicyName: EKSNodeRoleIpv6CNIPolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "ec2:AssignIpv6Addresses"
                Resource: "arn:aws:ec2:*:*:network-interface/*"
```

Weitere Informationen zu IAM-Rollen mithilfe von AWS CloudFormation Vorlagen finden Sie in den folgenden Abschnitten des AWS CloudFormation Benutzerhandbuchs:

- AWS::IAM::Role
- Auswählen einer Stack-Vorlage

AWS TNB-Servicerolle für Multus

Wenn Sie eine Amazon EKS-Ressource in Ihrem NSD erstellen und Multus als Teil Ihrer Bereitstellungsvorlage verwalten möchten, müssen Sie das `multus_role` Attribut angeben, um anzugeben, welche Rolle für die Verwaltung von Multus verwendet werden soll.

Das folgende Beispiel zeigt eine AWS CloudFormation Vorlage, die eine AWS TNB-Servicerolle für eine Multus-Richtlinie erstellt.

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBMultusRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBMultusRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - events.amazonaws.com
            Action:
              - "sts:AssumeRole"
          - Effect: Allow
```

```
            Principal:
              Service:
                - codebuild.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /
      Policies:
        - PolicyName: MultusRoleInlinePolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "codebuild:StartBuild"
                  - "logs:DescribeLogStreams"
                  - "logs:PutLogEvents"
                  - "logs:CreateLogGroup"
                  - "logs:CreateLogStream"
                Resource:
                  - "arn:aws:codebuild:*:*:project/tnb*"
                  - "arn:aws:logs:*:*:log-group:/aws/tnb/*"
              - Effect: Allow
                Action:
                  - "ec2:CreateNetworkInterface"
                  - "ec2:ModifyNetworkInterfaceAttribute"
                  - "ec2:AttachNetworkInterface"
                  - "ec2:DeleteNetworkInterface"
                  - "ec2:CreateTags"
                  - "ec2:DetachNetworkInterface"
                Resource: "*"
```

Weitere Informationen zu IAM-Rollen mithilfe von AWS CloudFormation Vorlagen finden Sie in den folgenden Abschnitten des AWS CloudFormation Benutzerhandbuchs:

- AWS::IAM::Role
- Auswählen einer Stack-Vorlage

AWS TNB-Servicerolle für eine Life-Cycle-Hook-Richtlinie

Wenn Ihr NSD- oder Netzwerkfunktionspaket einen Life-Cycle-Hook verwendet, benötigen Sie eine Service-Rolle, damit Sie eine Umgebung für die Ausführung Ihrer Life-Cycle-Hooks einrichten können.

> **Note**
>
> Ihre Lifecycle-Hook-Richtlinie sollte auf dem basieren, was Ihr Life-Cycle-Hook zu tun versucht.

Das folgende Beispiel zeigt eine AWS CloudFormation Vorlage, die eine AWS TNB-Servicerolle für eine Life-Cycle-Hook-Richtlinie erstellt.

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBHookRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBHookRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - codebuild.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AdministratorAccess"
```

Weitere Informationen zu IAM-Rollen mithilfe von AWS CloudFormation Vorlagen finden Sie in den folgenden Abschnitten des AWS CloudFormation Benutzerhandbuchs:

- AWS::IAM::Role
- Auswählen einer Stack-Vorlage

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI OR-API. 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": "*"
        }
    ]
}
```

Fehlerbehebung bei Identität und Zugriff bei AWS Telco Network Builder

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS TNB und IAM auftreten können.

Problembereiche

- Ich bin nicht berechtigt, eine Aktion in TNB durchzuführen AWS
- Ich bin nicht berechtigt, iam durchzuführen: PassRole

- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS TNB-Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion in TNB durchzuführen AWS

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer fiktiven *`my-example-widget`*-Ressource zu verwenden, jedoch nicht über `tnb:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 tnb:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Mateo-Richtlinie aktualisiert werden, damit er mit der `tnb:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam durchzuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an AWS TNB übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS TNB 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 von mir den Zugriff AWS-Konto auf meine AWS TNB-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 Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob AWS TNB diese Funktionen unterstützt, finden Sie unter. So funktioniert AWS Telco Network Builder 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.

Konformitätsvalidierung für TNB AWS

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 Implementierung 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.

Resilienz bei AWS TNB

Die AWS globale Infrastruktur basiert AWS-Regionen auf Availability Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

AWS TNB führt Ihren Netzwerkdienst auf EKS-Clustern in einer Virtual Private Cloud (VPC) in der von Ihnen AWS ausgewählten Region aus.

Sicherheit der Infrastruktur in TNB AWS

Als verwalteter Dienst ist AWS Telco Network Builder durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf AWS TNB 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.

Hier sind einige Beispiele für gemeinsame Verantwortlichkeiten:

- AWS ist verantwortlich für die Sicherung von Komponenten, die AWS TNB unterstützen, darunter:
  - Recheninstanzen (auch bekannt als Worker)
  - Interne Datenbanken
  - Netzwerkkommunikation zwischen internen Komponenten
  - Die AWS TNB-Anwendungsprogrammierschnittstelle (API)
  - AWS Softwareentwicklungskits (SDK)
- Sie sind dafür verantwortlich, Ihren Zugriff auf Ihre AWS Ressourcen und Ihre Workload-Komponenten zu sichern, einschließlich (aber nicht beschränkt auf):
  - IAM-Benutzer, -Gruppen, -Rollen und -Richtlinien
  - S3-Buckets, die Sie zum Speichern Ihrer Daten für TNB verwenden AWS
  - Andere Ressourcen AWS-Services und Ressourcen, die Sie zur Unterstützung des Netzwerkdienstes verwenden, den Sie über TNB bereitgestellt haben AWS
  - Ihr Anwendungscode
  - Verbindungen zwischen dem Netzwerkdienst, den Sie über AWS TNB bereitgestellt haben, und seinen Clients

> **Important**
>
> Sie sind für die Implementierung eines Notfallwiederherstellungsplans verantwortlich, mit dem ein Netzwerkdienst, den Sie über TNB bereitgestellt haben, effektiv wiederhergestellt werden kann. AWS

Sicherheitsmodell für Netzwerkkonnektivität

Die Netzwerkdienste, die Sie über AWS TNB bereitstellen, werden auf Recheninstanzen innerhalb einer Virtual Private Cloud (VPC) ausgeführt, die sich in einer von Ihnen AWS ausgewählten Region befindet. Eine VPC ist ein virtuelles Netzwerk in der AWS Cloud, das die Infrastruktur nach Arbeitslast oder organisatorischer Einheit isoliert. Die Kommunikation zwischen Recheninstanzen innerhalb von VPCs bleibt im AWS Netzwerk und wird nicht über das Internet übertragen. Ein Teil der internen Dienstkommunikation erfolgt über das Internet und ist verschlüsselt. Netzwerkdienste, die über AWS TNB für alle Kunden bereitgestellt werden, die in derselben Region laufen, teilen sich dieselbe VPC. Netzwerkdienste, die über AWS TNB für verschiedene Kunden bereitgestellt werden, verwenden separate Recheninstanzen innerhalb derselben VPC.

Die Kommunikation zwischen Ihren Netzwerkdienstclients und Ihrem Netzwerkdienst in AWS TNB erfolgt über das Internet. AWS TNB verwaltet diese Verbindungen nicht. Es liegt in Ihrer Verantwortung, Ihre Kundenverbindungen zu sichern.

Ihre Verbindungen zu AWS TNB über die AWS SDKs AWS Management Console, AWS Command Line Interface (AWS CLI) und sind verschlüsselt.

IMDS-Version

AWS TNB unterstützt Instances, die Instance Metadata Service Version 2 (IMDSv2), eine sitzungsorientierte Methode, nutzen. IMDSv2 bietet eine höhere Sicherheit als IMDSV1. Weitere Informationen finden Sie unter Umfassender Schutz vor offenen Firewalls, Reverse-Proxys und SSRF-Schwachstellen mit Verbesserungen am Amazon EC2 Instance Metadata Service.

Wenn Sie Ihre Instance starten, müssen Sie IMDSv2 verwenden. Weitere Informationen zu IMDSv2 finden Sie unter Verwenden von IMDSv2 im Amazon EC2 EC2-Benutzerhandbuch.

ÜberwachungAWSTNB

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung vonAWSTNB und dein andererAWSLösungen.AWSbietetAWS CloudTrailzum ZuschauenAWSTNB, melde, wenn etwas nicht stimmt, und ergreife gegebenenfalls automatische Maßnahmen.

BenutzeCloudTrailum detaillierte Informationen über die Anrufe zu erfassenAWSAPIs.
Sie können diese Aufrufe als Protokolldateien in Amazon S3 speichern. Sie können diese verwendenCloudTrailProtokolle, um Informationen zu ermitteln, z. B. welcher Anruf getätigt wurde, von welcher Quell-IP-Adresse der Anruf kam, wer den Anruf getätigt hat und wann der Anruf getätigt wurde.

DieCloudTrailDie Protokolle enthalten Informationen über die Aufrufe von API-Aktionen fürAWSTNB. Sie enthalten auch Informationen für Aufrufe von API-Aktionen von Diensten wie Amazon EC2 und Amazon EBS.

Protokollieren vonAWS Telco Network Builder-API-Aufrufen mitAWS CloudTrail

AWSTelco Network Builder ist mit integriertAWS CloudTrail, einem Dienst, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einemAWS Dienst inAWS TNB durchgeführten Aktionen liefert. CloudTrail erfasst alle API-Aufrufe fürAWS TNB als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von derAWS TNB-Konsole und Code-Aufrufe derAWS TNB-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon-S3-Bucket, einschließlich Ereignisse fürAWS TNB aktivieren. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail Konsole trotzdem in Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie feststellen CloudTrail, welche Anforderung anAWS TNB gesendet wurde, die IP-Adresse, von der die Anforderung gesendet wurde, den Absender und den Zeitpunkt der Anforderung sowie weitere Details.

Weitere Informationen CloudTrail finden Sie im AWS CloudTrailBenutzerhandbuch.

AWSTNB-Informationen in CloudTrail

CloudTrail wirdAWS-Konto beim Erstellen Ihres für Sie aktiviert. Wenn eine Aktivität in TNB eine Aktivität inAWS TNB in TNB in TNB eine Aktivität in TNB eine Aktivität in TNB eine CloudTrail Aktivität in TNB eine Aktivität inAWS TNB eine Aktivität in TNB eine Aktivität in TNB Sie können die neusten Ereignisse in Ihr(em) AWS-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail -Ereignisverlauf.

Um die Ereignisse in IhremAWS-Konto System einschließlich Ereignissen fürAWS TNB kontinuierlich aufzuzeichnen, erstellen Sie einen Trail. 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 Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon S3 Bucket bereit. Darüber hinaus können Sie andereAWS -Services konfigurieren, um die in den CloudTrail -Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail -Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail -Protokolldateien aus mehreren Konten

AlleAWS TNB-Aktionen werden von der AWSTelco Network Builder API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe von`CreateSolNetworkInstance` und`CreateSolNetworkPackage` Aktionen Einträge in den CloudTrail Protokolldateien.`CreateSolFunctionPackage`

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Anhand der Identitätsinformationen zur Benutzeridentität können Sie Folgendes bestimmen:

- Ob die Anfrage mit Stammbenutzer- oder AWS Identity and Access Management (IAM)-Anmeldeinformationen ausgeführt wurde.
- Ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer ausgeführt wurde.
- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter CloudTrail -Element userIdentity.

Grundlagen zu den Einträgen inAWS TNB-Protokolldateieinträgen

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, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail Protokolldateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher in keiner bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail -Protokolleintrag, der die`CreateSolFunctionPackage` Aktion demonstriert.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:example",
        "arn": "arn:aws:sts::111222333444:assumed-role/example/user",
        "accountId": "111222333444",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111222333444:role/example",
                "accountId": "111222333444",
                "userName": "example"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-02T01:42:39Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-02-02T01:43:17Z",
    "eventSource": "tnb.amazonaws.com",
    "eventName": "CreateSolFunctionPackage",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "XXX.XXX.XXX.XXX",
```

```
    "userAgent": "userAgent",
    "requestParameters": null,
    "responseElements": {
        "vnfPkgArn": "arn:aws:tnb:us-east-1:111222333444:function-package/
fp-12345678abcEXAMPLE",
        "id": "fp-12345678abcEXAMPLE",
        "operationalState": "DISABLED",
        "usageState": "NOT_IN_USE",
        "onboardingState": "CREATED"
    },
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111222333444",
    "eventCategory": "Management"
}
```


AWS TNB-Bereitstellungsaufgaben

Machen Sie sich mit den Bereitstellungsaufgaben vertraut, um Bereitstellungen effektiv zu überwachen und schneller Maßnahmen zu ergreifen.

In der folgenden Tabelle sind die AWS TNB-Bereitstellungsaufgaben aufgeführt:

Aufgabenname für Bereitstellungen, die vor dem 7. März 2024 gestartet wurden	Aufgabenname für Bereitstellungen, die am und nach dem 7. März 2024 gestartet wurden	Task description (Aufgabenbeschreibung)
AppInstallation	ClusterPluginInstall	Installiert das Multus-Plugin auf dem Amazon EKS-Cluster.
AppUpdate	keine Änderung des Namens	Aktualisiert die Netzwerkfunktionen, die bereits in einer Netzwerkinstanz installiert sind.
-	ClusterPluginUninstall	Deinstalliert die Plugins auf dem Amazon EKS-Cluster.

Aufgabenname für Bereitstellungen, die vor dem 7. März 2024 gestartet wurden	Aufgabenname für Bereitstellungen, die am und nach dem 7. März 2024 gestartet wurden	Task description (Aufgabenbeschreibung)
ClusterStorageClassConfiguration	keine Änderung des Namens	Konfiguriert die Speicherklasse (CSI-Treiber) auf einem Amazon EKS-Cluster.
FunctionDeletion	keine Änderung des Namens	Löscht Netzwerkfunktionen aus AWS TNB-Ressourcen.
FunctionInstantiation	FunctionInstall	Stellt Netzwerkfunktionen mithilfe von HELM bereit.
FunctionUninstallation	FunctionUninstall	Deinstalliert die Netzwerkfunktion von einem Amazon EKS-Cluster.
HookExecution	keine Änderung des Namens	Führt Lifecycle-Hooks aus, wie in der NSD definiert.
InfrastructureCancellation	keine Änderung des Namens	Bricht einen Netzwerkdienst ab.
InfrastructureInstantation	keine Änderung des Namens	Stellt AWS Ressourcen im Namen des Benutzers bereit.
InfrastructureTermination	keine Änderung des Namens	Stellt über TNB aufgerufene AWS Ressourcen ab. AWS
InventoryDeregistration	keine Änderung des Namens	Meldet AWS Ressourcen von TNB ab AWS .
KubernetesClusterConfiguration	ClusterConfiguration	Konfiguriert den Kubernetes-Cluster und fügt dem Amazon EKS zusätzliche IAM-Rollen hinzu, AuthMap wie im NSD definiert.
NetworkServiceFinalization	keine Änderung des Namens	Schließt den Netzwerkdienst ab und informiert über den Status eines Erfolgs oder Fehlers.

Aufgabenname für Bereitstellungen, die vor dem 7. März 2024 gestartet wurden	Aufgabenname für Bereitstellungen, die am und nach dem 7. März 2024 gestartet wurden	Task description (Aufgabenbeschreibung)
NetworkServiceInstantiation	keine Änderung des Namens	Initialisiert den Netzwerkdienst.
SelfManagedNodesConfiguration	keine Änderung des Namens	Bootstrappt selbstverwaltete Knoten mit Amazon EKS- und Kubernetes-Steuerebene.

Servicekontingente für AWS Telco Network Builder

Servicekontingente, auch als Limits bezeichnet, sind die maximale Anzahl von Serviceressourcen oder -vorgängen für Ihr AWS-Konto. Weitere Informationen finden Sie unter AWS Servicekontingente im Allgemeine Amazon Web Services-Referenz.

Im Folgenden sind die Servicekontingente für AWS TNB aufgeführt.

Name	Standard	Anpas	Beschreibung
Gleichzeitiger laufender Netzwerkdienstbetrieb	Jede unterstützte Region: 40	Ja	Die maximale Anzahl gleichzeitiger laufender Netzwerkdienstoperationen in einer Region.
Funktionspakete	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Funktionspaketen in einer Region.
Netzwerk-Pakete	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Netzwerkpaketen in einer Region.
Netzwerkdienstinstanzen	Jede unterstützte Region: 800	Ja	Die maximale Anzahl von Netzwerkdienstinstanzen in einer Region.

Dokumentenverlauf für das AWS TNB-Benutzerhandbuch

In der folgenden Tabelle werden die Dokumentationsversionen für AWS TNB beschrieben.

Änderung	Beschreibung	Datum
Neue Aufgabe und neue Aufgabennamen für bestehende Aufgaben	Eine neue Aufgabe ist verfügbar. Seit dem 7. März 2024 haben einige bestehende Aufgaben aus Gründen der Übersichtlichkeit neue Namen.	7. Mai 2024
Kubernetes-Version für Cluster	AWS TNB unterstützt jetzt Kubernetes-Versionen 1.29 zur Erstellung von Amazon EKS-Clustern.	10. April 2024
Support für Netzwerkschnittstellen `security_groups`	Sie können Sicherheitsgruppen an den Knoten AWS.networking.ENI anhängen.	2. April 2024
Support für Amazon EBS-Root-Volume-Verschlüsselung	Sie können die Amazon EBS-Verschlüsselung für das Amazon EBS-Root-Volume aktivieren. Fügen Sie zur Aktivierung die Eigenschaften im Knoten AWS.Compute.eks oder ManagedNodeAWS.Compute.Eks hinzu. SelfManagedNode	2. April 2024
Support für Node `labels`	Sie können Ihrer Knotengruppe im Knoten AWS.Compute.eks oder ManagedNodeAWS.Compute.Eks	19. März 2024

	Knotenbezeichnungen hinzufügen. SelfManagedNode	
Support für Netzwerkschnittstellen `source_dest_check`	Sie können über den Knoten AWS.networking.ENI angeben, ob Sie die Quell-/Zielüberprüfung der Netzwerkschnittstelle aktivieren oder deaktivieren möchten.	25. Januar 2024
Support für Amazon EC2 EC2-Instances mit benutzerdefinierten Benutzerdaten	Sie können Amazon EC2 EC2-Instances mit benutzerdefinierten Benutzerdaten über AWS.Compute starten. UserData Knoten.	16. Januar 2024
Support für Security Group	AWS TNB ermöglicht es Ihnen, die Security AWS Group-Ressource zu importieren.	8. Januar 2024
Die Beschreibung von wurde aktualisiert `network_interfaces`	Wenn die `network_interfaces` Eigenschaft im SelfManagedNode Knoten AWS.Compute.Eks ManagedNode oder AWS.Compute.Eks enthalten ist, ruft AWS TNB die Berechtigungen für ENIs von der Eigenschaft ab, falls verfügbar, oder von der Eigenschaft. `multus_role` `node_role`	18. Dezember 2023

Support für private Cluster	AWS TNB unterstützt jetzt private Cluster. Um einen privaten Cluster anzugeben, setzen Sie die `access` Eigenschaft auf`PRIVATE`.	11. Dezember 2023
Kubernetes-Version für Cluster	AWS TNB unterstützt jetzt Kubernetes-Versionen 1.28 zur Erstellung von Amazon EKS-Clustern.	11. Dezember 2023
AWS TNB unterstützt Platzierungsgruppen	Platzierungsgruppe für die `AWS.Compute.EKSSelfManagedNode` Knotendefinitionen `AWS.Compute.EKSManagedNode` und hinzugefügt.	11. Dezember 2023

AWS TNB fügt Unterstützung für IPv6 hinzu	AWS TNB unterstützt jetzt die Erstellung von Netzwerkinstanzen mit IPv6-Infrastruktur. Überprüfen Sie die Knoten AWS.Networking.VPC, .Networking.Subnet, .Networking.AWSAWS https://docs.aws.amazon.com/tnb/latest/ug/node-internet-gateway.htmlInternetGatewayAWS, .Netzwerke. SecurityGroupIngressRule, AWS.Netzwerke. SecurityGroupEgressRule, und AWS.compute.EKS für IPv6-Konfigurationen. Wir haben auch die Knoten AWS.networking.NatGateway und .Networking.Route für die NAT64-Konfiguration hinzugefügt.AWS Wir haben die AWS TNB-Servicerolle und die AWS TNB-Servicerolle für die Amazon EKS-Knotengruppe für IPv6-Berechtigungen aktualisiert. Sehen Sie sich Beispiele für Richtlinien für Servicerollen an.	16. November 2023
Der AWS TNB-Servicerollenrichtlinie wurden Berechtigungen hinzugefügt	Wir haben der AWS TNB-Servicerollenrichtlinie für Amazon S3 und AWS CloudFormation zur Aktivierung der Infrastrukturinstanziierung Berechtigungen hinzugefügt.	23. Oktober 2023

AWS TNB wurde in mehr Regionen eingeführt	AWS TNB ist jetzt in den Regionen Asien-Pazifik (Seoul), Kanada (Zentral), Europa (Spanien), Europa (Stockholm) und Südamerika (São Paulo) verfügbar.	27. September 2023
Tags für .compute.EKS AWS SelfManagedNode	AWS TNB unterstützt jetzt Tags für die Knotendefinition. `AWS.Compute.EKSSelfManagedNode`	22. August 2023
AWS TNB unterstützt Instances, die IMDSv2 nutzen	Wenn Sie Ihre Instance starten, müssen Sie IMDSv2 verwenden.	14. August 2023
Aktualisierte Berechtigungen für MultusRoleInlinePolicy	Das beinhaltet MultusRoleInlinePolicy jetzt die `ec2:DeleteNetworkInterface` Erlaubnis.	7. August 2023
Kubernetes-Version für Cluster	AWS TNB unterstützt jetzt Kubernetes-Versionen 1.27 zur Erstellung von Amazon EKS-Clustern.	25. Juli 2023
AWS.compute.EKS. AuthRole	AWS TNB unterstützt AuthRole das, sodass Sie dem Amazon EKS-Cluster IAM-Rollen hinzufügen können, `aws-auth ConfigMap` sodass Benutzer über eine IAM-Rolle auf den Amazon EKS-Cluster zugreifen können.	19. Juli 2023

AWS TNB unterstützt Sicherheitsgruppen.	AWS.Networking wurde hinzugefügt. SecurityGroup, AWS.Netzwerke.SecurityGroupEgressRule, und AWS.Networking.SecurityGroupIngressRulezur NSD-Vorlage.	18. Juli 2023
Kubernetes-Version für Cluster	AWS TNB unterstützt die Kubernetes-Versionen 1.22 bis 1.26 zur Erstellung von Amazon EKS-Clustern. AWS TNB unterstützt die Kubernetes-Versionen 1.21 nicht mehr.	11. Mai 2023
AWS.compute.eks SelfManagedNode	Sie können selbstverwaltete Worker-Knoten in regionalen, AWS Local Zones und erstellen. AWS Outposts	29. März 2023
Erstversion	Dies ist die erste Version des AWS TNB-Benutzerhandbuchs.	21. Februar 2023

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.