Guida per l'utente

AWS Costruttore di reti di telecomunicazioni

AWS Costruttore di reti di telecomunicazioni: Guida per l'utente

Table of Contents

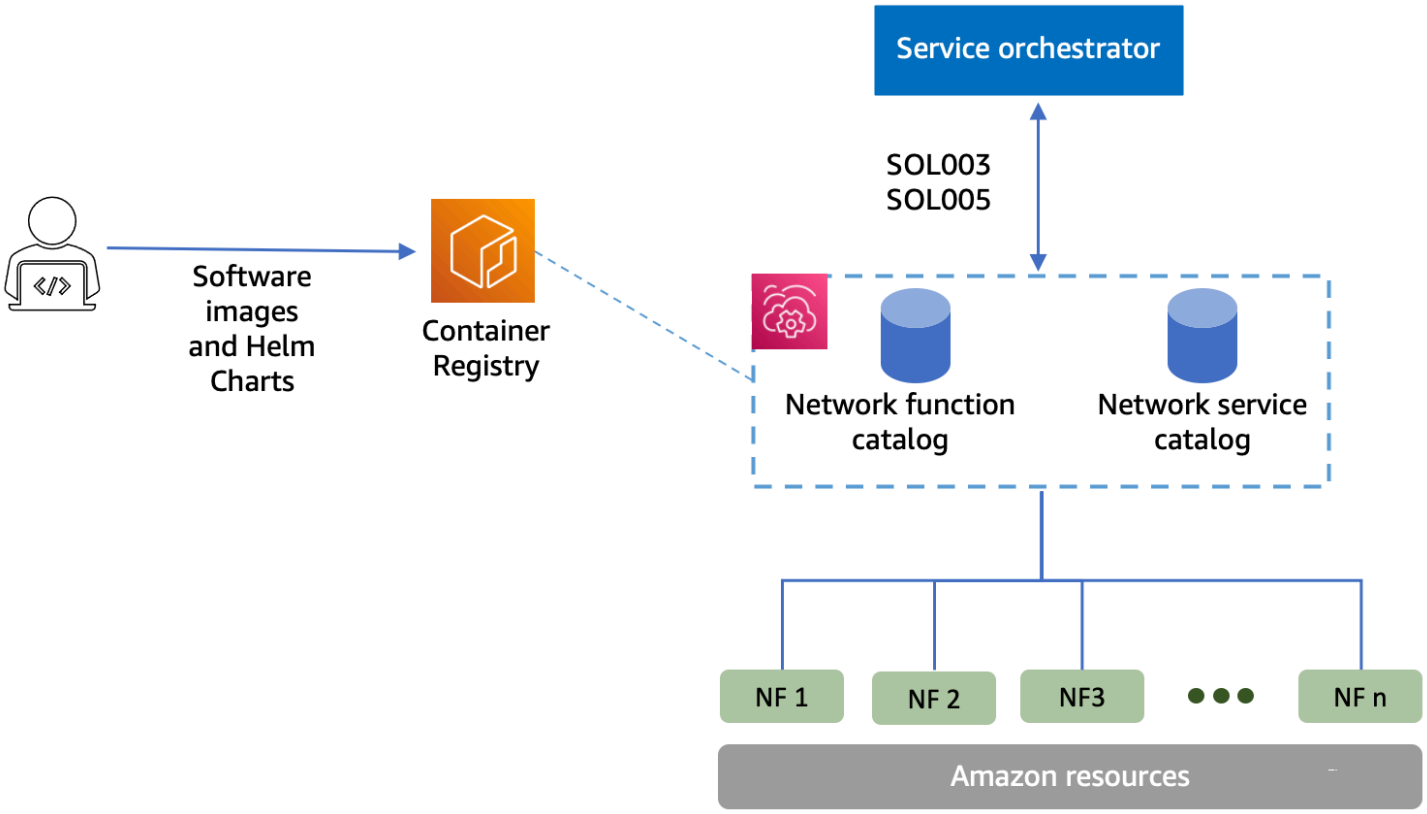
Cos'è AWS Telco Network Builder?

AWS Telco Network Builder (AWS TNB) è un AWS servizio che fornisce ai fornitori di servizi di comunicazione (CSPs) un modo efficiente per implementare, gestire e scalare le reti 5G sull'infrastruttura. AWS

Con AWS TNB, implementate reti 5G scalabili e sicure Cloud AWS utilizzando un'immagine della rete in modo automatizzato. Non è necessario apprendere nuove tecnologie, decidere quale servizio di elaborazione utilizzare o sapere come fornire e configurare le risorse. AWS

Dovrete invece descrivere l'infrastruttura di rete e fornire le immagini software delle funzioni di rete fornite dai fornitori di software indipendenti (ISV) partner. AWS TNBsi integra con orchestratori di AWS servizi e servizi di terze parti per fornire automaticamente l' AWS infrastruttura necessaria, implementare funzioni di rete containerizzate e configurare la gestione della rete e degli accessi per creare un servizio di rete completamente operativo.

Il diagramma seguente illustra le integrazioni logiche tra AWS TNB e gli orchestratori di servizi per implementare le funzioni di rete utilizzando interfacce standard basate sull'European Telecommunications Standards Institute (). ETSI



Argomenti

AWS Sei nuovo a?

Se non conosci AWS prodotti e servizi, inizia a saperne di più con le seguenti risorse:

- Introduction to AWS
- Guida introduttiva con AWS

Per chi è AWS TNB?

AWS TNBserve per CSPs sfruttare l'efficienza in termini di costi, l'agilità e l'elasticità Cloud AWS offerte senza scrivere e mantenere script e configurazioni personalizzati per progettare, implementare e gestire i servizi di rete. AWS TNBfornisce automaticamente l' AWS infrastruttura necessaria, implementa funzioni di rete containerizzate e configura la gestione delle reti e degli accessi per creare servizi di rete completamente operativi basati sui descrittori dei servizi di rete CSP definiti e sulle funzioni di rete che desidera implementare. CSP

AWS TNBcaratteristiche

Di seguito sono riportati alcuni dei motivi che un CSP vorrebbe utilizzare AWS TNB:

Aiuta a semplificare le attività

Offrite maggiore efficienza alle operazioni di rete, ad esempio implementando nuovi servizi, aggiornando e aggiornando le funzioni di rete e modificando le topologie dell'infrastruttura di rete.

Si integra con gli orchestratori

AWS TNBsi integra con i più diffusi orchestratori di servizi di terze parti conformi. ETSI

Scale

Puoi configurare AWS TNB la scalabilità AWS delle risorse sottostanti per soddisfare la domanda di traffico, eseguire in modo più efficiente gli aggiornamenti delle funzioni di rete, implementare le modifiche alla topologia dell'infrastruttura di rete e ridurre i tempi di implementazione dei nuovi servizi 5G da giorni a ore.

Ispeziona e monitora le risorse AWS

AWS TNBti consente di ispezionare e monitorare le AWS risorse che supportano la tua rete su un'unica dashboard, come Amazon VPCEC2, Amazon e AmazonEKS.

Supporta modelli di servizio

AWS TNBconsente di creare modelli di servizio per tutti i carichi di lavoro di telecomunicazione (RAN, Core,IMS). È possibile creare una nuova definizione di servizio, riutilizzare un modello esistente o effettuare l'integrazione con una pipeline di integrazione e distribuzione continua (CI/CD) per pubblicare una nuova definizione.

Tiene traccia delle modifiche alle implementazioni di rete

Quando modifichi la configurazione sottostante di un'implementazione di funzioni di rete, ad esempio cambiando il tipo di istanza di un tipo di EC2 istanza Amazon, puoi tenere traccia delle modifiche in modo ripetibile e scalabile. Per farlo manualmente sarebbe necessario gestire lo stato della rete, creare ed eliminare risorse e prestare attenzione all'ordine delle modifiche necessarie. Quando si utilizza AWS TNB per gestire il ciclo di vita della funzione di rete, si apportano solo le modifiche ai descrittori dei servizi di rete che descrivono la funzione di rete. AWS TNBapporterà quindi automaticamente le modifiche richieste nell'ordine corretto.

Semplifica il ciclo di vita delle funzioni di rete

È possibile gestire la prima versione e tutte le versioni successive di una funzione di rete e specificare quando eseguire l'aggiornamento. Puoi anche gestire RAN le tue applicazioni Core e di rete nello stesso modo. IMS

Accedendo AWS TNB

È possibile creare, accedere e gestire le AWS TNB risorse utilizzando una delle seguenti interfacce:

- AWS TNBconsole: fornisce un'interfaccia web per la gestione della rete.
- AWS TNBAPI— Fornisce un RESTful API strumento per eseguire AWS TNB azioni. Per ulteriori informazioni, vedere AWS TNBAPIReference.

- AWS Command Line Interface (AWS CLI) — Fornisce comandi per un'ampia gamma di AWS servizi, tra cui AWS TNB. È supportato su Windows, macOS e Linux. Per ulteriori informazioni, consulta AWS Command Line Interface.
- AWS SDKs— Fornisce informazioni specifiche per la lingua APIs e completa molti dettagli di connessione. Questi includono il calcolo delle firme e la gestione di errori e di nuovi tentativi di richiesta. Per ulteriori informazioni, vedere. AWSSDKs

Prezzi per AWS TNB

AWS TNBaiuta ad CSPs automatizzare l'implementazione e la gestione delle proprie reti di telecomunicazioni su. AWS Paghi per le seguenti due dimensioni quando utilizzi: AWS TNB

- Per funzione di rete gestita item (MNFI) ore.
- Per numero di API richieste.

Inoltre, l'utilizzo di altri AWS servizi in combinazione con. AWS TNB Per ulteriori informazioni, consulta la pagina Prezzi.AWS TNB

Per visualizzare la tua fattura, passa al Pannello di controllo di gestione fatturazione e costi nella console AWS Billing and Cost Management. La fattura contiene collegamenti per passare ai report di utilizzo, che consentono di visualizzare i dettagli della fattura. Per ulteriori informazioni sulla fatturazione AWS dell'account, consulta Fatturazione AWS dell'account.

In caso di domande relative alla AWS fatturazione, agli account e agli eventi, contatta l' AWS assistenza.

AWS Trusted Advisor è un servizio che puoi utilizzare per ottimizzare i costi, la sicurezza e le prestazioni del tuo AWS ambiente. Per ulteriori informazioni, vedere AWS Trusted Advisor.

Cosa c'è dopo

Per ulteriori informazioni su come iniziare AWS TNB, consulta i seguenti argomenti:

- Configurazione AWS TNB— Completare i passaggi preliminari.
- Guida introduttiva con AWS TNB— Implementa la tua prima funzione di rete, come l'unità centralizzata (CU), la funzione di gestione degli accessi e della mobilità (AMF), la funzione User Plane (UPF) o un core 5G completo.

Come AWS TNB funziona

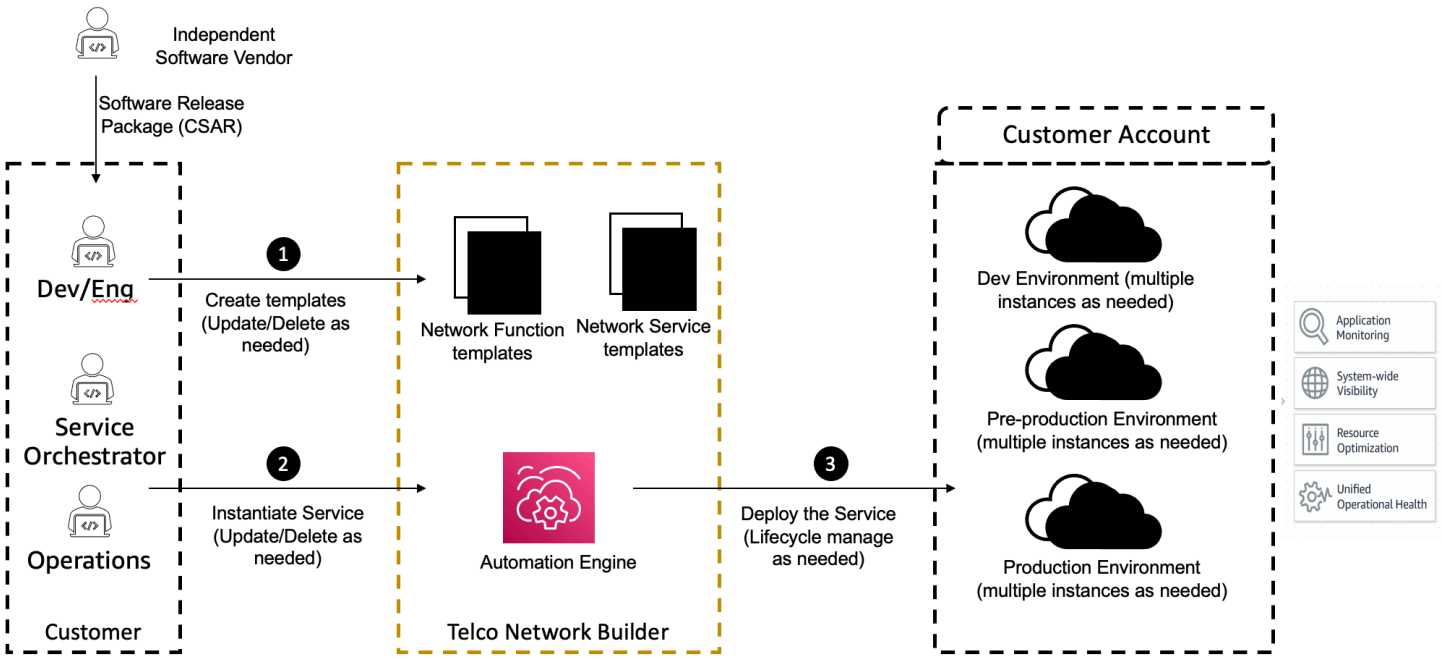
AWS TNBsi integra con end-to-end orchestratori e AWS risorse standardizzati per gestire reti 5G complete.

AWS TNBconsente di importare pacchetti di funzioni di rete e descrittori di servizi di rete (NSDs) e fornisce il motore di automazione per gestire le reti. Puoi usare il tuo end-to-end orchestratore e integrarlo con AWS TNBAPIs, o usarlo AWS TNB SDKs per creare il tuo flusso di automazione. Per ulteriori informazioni, consulta AWS TNBarchitettura .

Argomenti

AWS TNBarchitettura

AWS TNBoffre la possibilità di eseguire operazioni di gestione del ciclo di vita tramite AWS Management Console, AWS CLI AWS TNB RESTAPI, e. SDKs Ciò consente a diverse CSP personalità, come i membri dei team di ingegneria, operazioni e sistema programmatico, di trarne vantaggio. AWS TNB È possibile creare e caricare un pacchetto di funzioni di rete come file Cloud Service Archive (CSAR). Il CSAR file contiene grafici Helm, immagini software e un Network Function Descriptor ()NFD. È possibile utilizzare i modelli per distribuire ripetutamente più configurazioni di quel pacchetto. Si creano modelli di servizi di rete che definiscono l'infrastruttura e le funzioni di rete che si desidera implementare. È possibile utilizzare le sostituzioni dei parametri per distribuire configurazioni diverse in posizioni diverse. È quindi possibile creare un'istanza di rete, utilizzando i modelli e distribuire le funzioni di rete sull'infrastruttura. AWS AWS TNBti offre la visibilità delle tue implementazioni.



Integrazione con Servizi AWS

Una rete 5G è costituita da una serie di funzioni di rete containerizzate interconnesse distribuite su migliaia di cluster Kubernetes. AWS TNBsi integra con quanto segue, specifico per le telecomunicazioni, per creare un servizio di rete completamente Servizi AWS operativo: APIs

- Amazon Elastic Container Registry (AmazonECR) per archiviare gli artefatti delle funzioni di rete di Independent Software Vendors (ISVs).
- Amazon Elastic Kubernetes Service (AmazonEKS) per configurare i cluster.
- Amazon VPC per costrutti di rete.
- Gruppi di sicurezza che utilizzano AWS CloudFormation.
- AWS CodePipeline per gli obiettivi di distribuzione tra Regioni AWS AWS Local Zones e AWS Outposts.
- IAMper definire i ruoli.
- AWS Organizations per controllare l'accesso a AWS TNBAPIs.
- AWS Health Dashboard e AWS CloudTrail per monitorare le metriche relative alla salute e ai post.

AWS TNBquote di risorse

Your Account AWS ha delle quote predefinite, precedentemente denominate limiti, per ciascuna di esse. Servizio AWS Salvo diversa indicazione, ogni quota è specifica per un. Regione AWSÈ possibile richiedere un aumento per alcune quote, ma non per tutte le quote.

Per visualizzare le quote per AWS TNB, apri la console Service Quotas. Nel riquadro di navigazione Servizi AWS, scegli e seleziona. AWS TNB

Per richiedere un aumento delle quote, consultare Richiesta di aumento delle quote nella Guida dell'utente di Service Quotas.

La tua Account AWS ha le seguenti quote relative a AWS TNB.

Quota di risorse	Descrizione	Valore predefinito	Modificabile?
Istanze di servizi di rete	Il numero massimo di istanze di servizi di rete in una regione.	800	Sì
Operazioni simultanee di servizi di rete in corso	Il numero massimo di operazioni di servizio di rete in corso e simultanee in una regione.	40	Sì
Pacchetti di rete	Il numero massimo di pacchetti di rete in una regione.	40	Sì
Pacchetti di funzioni	Il numero massimo di pacchetti di funzioni in una regione.	200	Sì

AWS TNBconcetti

Questo argomento descrive i concetti essenziali per aiutarti a iniziare a usare AWS TNB.

Indice

Ciclo di vita di una funzione di rete

AWS TNBti aiuta durante tutto il ciclo di vita delle tue funzioni di rete. Il ciclo di vita delle funzioni di rete include le seguenti fasi e attività:

Pianificazione

1. Pianifica la tua rete identificando le funzioni di rete da implementare.
2. Inserisci le immagini del software per le funzioni di rete in un archivio di immagini container.
3. Crea i CSAR pacchetti da distribuire o aggiornare.
4. AWS TNBUtilizzatelo per caricare il CSAR pacchetto che definisce la funzione di rete (ad esempio, CU AMF eUPF) e effettuate l'integrazione con una pipeline di integrazione e distribuzione continua (CI/CD) che può aiutarvi a creare nuove versioni del CSAR pacchetto non appena sono disponibili nuove immagini software per le funzioni di rete o script per i clienti.

Configurazione

1. Identifica le informazioni necessarie per l'implementazione, come il tipo di calcolo, la versione della funzione di rete, le informazioni IP e i nomi delle risorse.
2. Utilizzate le informazioni per creare il descrittore del servizio di rete ()NSD.
3. Inserimento NSDs che definisce le funzioni di rete e le risorse necessarie per la creazione di istanze da parte della funzione di rete.

Istanziazione

1. Crea l'infrastruttura richiesta dalle funzioni di rete.

2. Crea un'istanza (o fornisci) la funzione di rete come definita nella sua NSD e inizia a trasportare traffico.
3. Convalida gli asset.

Produzione

Durante il ciclo di vita della funzione di rete, completerai le operazioni di produzione, come:

- Aggiorna la configurazione della funzione di rete, ad esempio aggiorna un valore nella funzione di rete distribuita.
- Aggiorna l'istanza di rete con un nuovo pacchetto di rete e i valori dei parametri. Ad esempio, aggiorna il EKS `version` parametro Amazon nel pacchetto di rete.

Usa interfacce standardizzate

AWS TNBsi integra con gli orchestratori di servizi conformi allo European Telecommunications Standards Institute (ETSI) e consente di semplificare l'implementazione dei servizi di rete. Gli orchestratori di servizi possono utilizzare AWS TNBSDKs, the o the APIs per avviare operazioni, come l'CLIistanziazione o l'aggiornamento di una funzione di rete a una nuova versione.

AWS TNBsupporta le seguenti specifiche.

Specifiche	Versione	Descrizione
ETSISOL001	versione 3.6.1	Definisce gli standard per consentire descrittori di funzioni di rete TOSCA basati su di essa.
ETSISOL002	versione 3.6.1	Definisce i modelli relativi alla gestione delle funzioni di rete.
ETSISOL003	versione 3.6.1	Definisce gli standard per la gestione del ciclo di vita delle funzioni di rete.
ETSISOL004	versione 3.6.1	Definisce CSAR gli standard per i pacchetti di funzioni di rete.
ETSISOL005	versione 3.6.1	Definisce gli standard per i pacchetti di servizi di rete e la gestione del ciclo di vita dei servizi di rete.

Specifiche	Versione	Descrizione
ETSISOL007	versione 3.5.1	Definisce gli standard per consentire i descritto ri TOSCA di servizi di rete basati.

Pacchetti di funzioni di rete per AWS TNB

Con AWS TNB, è possibile archiviare pacchetti di funzioni di rete conformi a ETSI SOL 001/ SOL 004 in un catalogo di funzioni. Quindi, puoi caricare pacchetti Cloud Service Archive (CSAR) che contengono artefatti che descrivono la tua funzione di rete.

- Descrittore delle funzioni di rete: definisce i metadati per l'onboarding dei pacchetti e la gestione delle funzioni di rete
- Immagini software: riferimenti alla funzione di rete Container Images. Amazon Elastic Container Registry (AmazonECR) può fungere da archivio di immagini delle funzioni di rete.
- File aggiuntivi: da utilizzare per gestire la funzione di rete, ad esempio script e grafici Helm.

CSARÈ un pacchetto definito dallo OASIS TOSCA standard e include un descrittore di rete/servizio conforme alle specifiche. OASIS TOSCA YAML Per informazioni sulle specifiche richieste, vedere. YAML TOSCAriferimento per AWS TNB

Di seguito è riportato un esempio di descrittore di funzioni di rete.

```
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"
```

Descrittori di servizi di rete per AWS TNB

AWS TNBmemorizza i descrittori dei servizi di rete (NSDs) sulle funzioni di rete che si desidera implementare e su come distribuirle nel catalogo. È possibile caricare il YAML NSD file (`vnfd.yaml`), come descritto da ETSI SOL 007, per includere le seguenti informazioni:

- Funzione di rete che si desidera implementare
- Istruzioni di rete
- Istruzioni di calcolo
- Lifecycle Hooks (script personalizzati)

AWS TNBsupporta ETSI gli standard per la modellazione di risorse, come rete, servizi e funzioni, nel linguaggio. TOSCA AWS TNBne rende più efficiente l'utilizzo Servizi AWS modellandole in modo che il service orchestrator di servizi ETSI conforme sia in grado di comprenderle.

Di seguito è riportato un frammento di una dimostrazione di come modellare. NSD Servizi AWS La funzione di rete verrà implementata su un EKS cluster Amazon con Kubernetes versione 1.27. Le sottoreti per le applicazioni sono Subnet01 e Subnet02. Puoi quindi definire NodeGroups le tue applicazioni con Amazon Machine Image (AMI), tipo di istanza e configurazione con scalabilità automatica.

```
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
```

Gestione e operazioni per AWS TNB

Con AWS TNB, è possibile gestire la rete utilizzando operazioni di gestione standardizzate in conformità con ETSI SOL 003 e SOL 005. È possibile utilizzarlo AWS TNB APIs per eseguire operazioni sul ciclo di vita come:

- Istanziazione delle funzioni di rete.
- Interruzione delle funzioni di rete.
- Aggiornamento delle funzioni di rete per sostituire le implementazioni di Helm.
- Aggiornamento di un'istanza di rete istanziata o aggiornata con un nuovo pacchetto di rete e valori dei parametri.

- Gestione delle versioni dei pacchetti di funzioni di rete.
- Gestione delle versioni del tuoNSDs.
- Recupero di informazioni sulle funzioni di rete distribuite.

Descrittori dei servizi di rete per AWS TNB

Un descrittore di servizi di rete (NSD) è un `.yaml` file contenuto in un pacchetto di rete che utilizza lo TOSCA standard per descrivere le funzioni di rete che si desidera implementare e l' AWS infrastruttura su cui si desidera implementare le funzioni di rete. Per definire NSD e configurare le risorse sottostanti e le operazioni del ciclo di vita della rete, è necessario comprendere lo schema supportato da. NSD TOSCA AWS TNB

Il NSD file è suddiviso nelle seguenti parti:

1. TOSCAversione della definizione: questa è la prima riga del NSD YAML file e contiene le informazioni sulla versione, mostrate nell'esempio seguente.

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

2. VNFD— NSD Contiene la definizione della funzione di rete su cui eseguire le operazioni del ciclo di vita. Ogni funzione di rete deve essere identificata dai seguenti valori:
   - Un ID univoco per`descriptor_id`. L'ID deve corrispondere all'ID nel CSAR pacchetto di funzioni di rete.
   - Un nome univoco per`namespace`. Il nome deve essere associato a un ID univoco per facilitarne il riferimento in tutto il NSD YAML file, come illustrato nell'esempio seguente.

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

3. Modello di topologia: definisce le risorse da distribuire, l'implementazione delle funzioni di rete ed eventuali script personalizzati, come i lifecycle hook. Questo viene mostrato nell'esempio seguente.

```
topology_template:

  node_templates:
```

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

4. Nodi aggiuntivi: ogni risorsa modellata presenta sezioni per proprietà e requisiti. Le proprietà descrivono gli attributi facoltativi o obbligatori di una risorsa, ad esempio la versione. I requisiti descrivono le dipendenze che devono essere fornite come argomenti. Ad esempio, per creare una risorsa Amazon EKS Node Group, deve essere creata all'interno di un Amazon EKS Cluster. Questo viene mostrato nell'esempio seguente.

```
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
```

Configurazione AWS TNB

Esegui la configurazione AWS TNB completando le attività descritte in questo argomento.

Attività

- Registrati per un Account AWS
- Crea un utente con accesso amministrativo
- Scegli una regione AWS
- Annota l'endpoint del servizio
- (Facoltativo) Installa AWS CLI
- Imposta i ruoli AWS TNB

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Attiva l'autenticazione a più fattori (MFA) per il tuo utente root.

   Per istruzioni, consulta Abilitare un MFA dispositivo virtuale per l'utente Account AWS root (console) nella Guida per l'IAMutente.

Crea un utente con accesso amministrativo

1. Abilita IAM Identity Center.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con i valori predefiniti IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente dell'IAMIdentity Center, utilizza l'accesso URL che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso tramite un utente di IAM Identity Center, consulta Accesso al portale di AWS accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Scegli una regione AWS

Per visualizzare l'elenco delle regioni disponibili per AWS TNB, consulta l'elenco dei servizi AWS regionali. Per visualizzare l'elenco degli endpoint per l'accesso programmatico, consulta gli AWS TNBendpoint in. Riferimenti generali di AWS

Annota l'endpoint del servizio

Per connettersi a livello di codice a un AWS servizio, si utilizza un endpoint. Oltre agli AWS endpoint standard, alcuni AWS servizi offrono FIPS endpoint in regioni selezionate. Per ulteriori informazioni, consulta Endpoint del servizio AWS.

Nome della regione	Regione	Endpoint	Protocollo	
US East (N. Virginia)	us-east-1	tnb.us-east-1.amazonaws.com	HTTPS	
US West (Oregon)	us-west-2	tnb.us-west-2.amazonaws.com	HTTPS	
Asia Pacifico (Seoul)	ap-northeast-2	tnb.ap-northeast-2.amazonaws.com	HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Asia Pacific (Sydney)	ap-southeast-2	tnb.ap-southeast-2.amazonaws.com	HTTPS	
Canada (Centrale)	ca-central-1	tnb.ca-central-1.amazonaws.com	HTTPS	
Europa (Francoforte)	eu-central-1	tnb.eu-central-1.amazonaws.com	HTTPS	
Europa (Parigi)	eu-west-3	tnb.eu-west-3.amazonaws.com	HTTPS	
Europa (Spagna)	eu-south-2	tnb.eu-south-2.amazonaws.com	HTTPS	
Europa (Stoccolma)	eu-north-1	tnb.eu-north-1.amazonaws.com	HTTPS	
Sud America (São Paulo)	sa-east-1	tnb.sa-east-1.amazonaws.com	HTTPS	

(Facoltativo) Installa AWS CLI

Il AWS Command Line Interface (AWS CLI) fornisce comandi per un'ampia gamma di AWS prodotti ed è supportato su Windows, macOS e Linux. È possibile accedere AWS TNB utilizzando. AWS CLI Per iniziare, consulta la AWS Command Line Interface Guida per l'utente di . Per ulteriori informazioni sui comandi per AWS TNB, vedere tnb nel AWS CLI Command Reference.

Imposta i ruoli AWS TNB

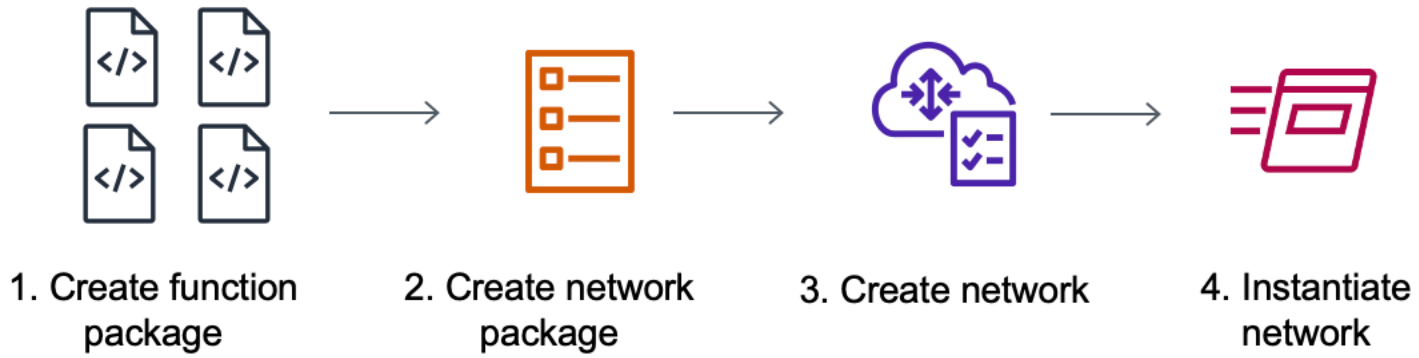
È necessario creare un ruolo di IAM servizio per gestire diverse parti della AWS TNB soluzione. AWS TNBi ruoli di servizio possono effettuare API chiamate ad altri AWS servizi, ad esempio AWS CloudFormation AWS CodeBuild, e a vari servizi di elaborazione e archiviazione, per conto dell'utente, per creare istanze e gestire le risorse per l'implementazione.

Per ulteriori informazioni sul ruolo di AWS TNB servizio, vedere. Gestione delle identità e degli accessi per AWS TNB

Guida introduttiva con AWS TNB

Questo tutorial illustra come utilizzare AWS TNB per implementare una funzione di rete, ad esempio Centralized Unit (CU), Access and Mobility Management Function () o 5G User Plane Function (AMF). UPF

Il diagramma seguente illustra il processo di implementazione:



Attività

- Prerequisiti
- Create un pacchetto di funzioni
- Crea un pacchetto di rete
- Crea e crea un'istanza di rete
- Eliminazione

Prerequisiti

Prima di poter eseguire una distribuzione corretta, è necessario disporre di quanto segue:

- Un piano AWS Business Support.
- Autorizzazioni tramite IAM ruoli.
- Un pacchetto Network Function (NF) conforme ETSI SOL a 001/ 004. SOL
- Modelli Network Service Descriptor (NSD) conformi a 007. ETSI SOL

È possibile utilizzare un pacchetto di funzioni di esempio o un pacchetto di rete contenuto in Pacchetti di esempio per il AWS TNB GitHub sito.

Create un pacchetto di funzioni

Un pacchetto di funzioni di rete è un file Cloud Service Archive (CSAR). Il CSAR file contiene grafici Helm, immagini software e un Network Function Descriptor ()NFD.

Per creare un pacchetto di funzioni

1. Apri la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Pacchetti di funzioni.
3. Scegliete Crea pacchetto di funzioni.
4. In Carica pacchetto di funzioni, scegliete Scegli file e carica ogni CSAR pacchetto come `.zip` file. Puoi caricare un massimo di 10 file.
5. (Facoltativo) In Tag, scegli Aggiungi nuovo tag e inserisci una chiave e un valore. Puoi utilizzare i tag per cercare e filtrare le tue risorse o tenere traccia AWS dei costi.
6. Scegli Next (Successivo).
7. Esamina i dettagli del pacchetto, quindi scegli Crea pacchetto di funzioni.

Crea un pacchetto di rete

Un pacchetto di rete specifica le funzioni di rete che si desidera implementare e come distribuirle nel catalogo.

Per creare un pacchetto di rete

1. Nel riquadro di navigazione, scegli Pacchetti di rete.
2. Scegli Crea pacchetto di rete.
3. In Carica pacchetto di rete, scegli Scegli file e carica ciascuno NSD come `.zip` file. Puoi caricare un massimo di 10 file.
4. (Facoltativo) In Tag, scegli Aggiungi nuovo tag e inserisci una chiave e un valore. Puoi utilizzare i tag per cercare e filtrare le tue risorse o tenere traccia AWS dei costi.
5. Scegli Next (Successivo).

6. Scegli Crea pacchetto di rete.

Crea e crea un'istanza di rete

Un'istanza di rete è una singola rete creata in AWS TNB che può essere distribuita. È necessario creare un'istanza di rete e crearne un'istanza. Quando si crea un'istanza di rete, si effettua il AWS TNB provisioning dell' AWS infrastruttura necessaria, si distribuiscono funzioni di rete containerizzate e si configura la gestione della rete e degli accessi per creare un servizio di rete completamente operativo.

Per creare e istanziare un'istanza di rete

1. Nel riquadro di navigazione, scegli Reti.
2. Scegli Crea istanza di rete.
3. Inserisci un nome e una descrizione per la rete, quindi scegli Avanti.
4. Scegli un pacchetto di rete. Verifica i dettagli e scegli Avanti.
5. Scegli Crea istanza di rete. Lo stato iniziale è`Created`.

   Viene visualizzata la pagina Reti che mostra la nuova istanza di rete nello `Not instantiated` stato.
6. Seleziona l'istanza di rete, scegli Azioni e Crea istanza.

   Viene visualizzata la pagina di istanza di rete.
7. Rivedi i dettagli e aggiorna i valori dei parametri. Gli aggiornamenti ai valori dei parametri si applicano solo a questa istanza di rete. I parametri nei VNFD pacchetti NSD and non vengono modificati.
8. Scegli Instantiate network.

   Viene visualizzata la pagina sullo stato della distribuzione.
9. Utilizza l'icona Aggiorna per tenere traccia dello stato di distribuzione dell'istanza di rete. È inoltre possibile abilitare l'aggiornamento automatico nella sezione Attività di distribuzione per tenere traccia dell'avanzamento di ciascuna attività.

Eliminazione

Ora puoi eliminare le risorse che hai creato per questo tutorial.

Per eliminare le risorse

1. Nel riquadro di navigazione, scegli Reti.
2. Scegli l'ID della rete, quindi scegli Termina.
3. Quando viene richiesta la conferma, inserisci l'ID di rete, quindi scegli Termina.
4. Usa l'icona Aggiorna per tenere traccia dello stato dell'istanza di rete.
5. (Facoltativo) Seleziona la rete e scegli Elimina.

Pacchetti di funzioni per AWS TNB

Un pacchetto di funzioni è un file.zip in formato CSAR (Cloud Service Archive) che contiene una funzione di rete (un'applicazione di telecomunicazione ETSI standard) e un descrittore di pacchetti di funzioni che utilizza TOSCA lo standard per descrivere come le funzioni di rete devono essere eseguite sulla rete.

Attività

- Crea un pacchetto di funzioni in AWS TNB
- Visualizza un pacchetto di funzioni in AWS TNB
- Scaricate un pacchetto di funzioni da AWS TNB
- Eliminare un pacchetto di funzioni da AWS TNB

Crea un pacchetto di funzioni in AWS TNB

Scopri come creare un pacchetto di funzioni nel catalogo delle funzioni di AWS TNB rete. La creazione di un pacchetto di funzioni è il primo passo per creare una rete in AWS TNB. Dopo aver caricato un pacchetto di funzioni, è possibile creare un pacchetto di rete.

Console

Per creare un pacchetto di funzioni utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Pacchetti di funzioni.
3. Scegliete Crea pacchetto di funzioni.
4. Scegli i file e carica ogni CSAR pacchetto come `.zip` file. Puoi caricare un massimo di 10 file.
5. Scegli Next (Successivo).
6. Controlla i dettagli del pacchetto.
7. Scegli Crea pacchetto di funzioni.

AWS CLI

Per creare un pacchetto di funzioni utilizzando AWS CLI

1. Utilizzate il create-sol-function-packagecomando per creare un nuovo pacchetto di funzioni:

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

2. Utilizzate il comando put-sol-function-package-content per caricare il contenuto del pacchetto di funzioni. Per esempio:

```
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
```

Visualizza un pacchetto di funzioni in AWS TNB

Scopri come visualizzare il contenuto di un pacchetto di funzioni.

Console

Per visualizzare un pacchetto di funzioni utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Pacchetti di funzioni.
3. Usa la casella di ricerca per trovare il pacchetto di funzioni

AWS CLI

Per visualizzare un pacchetto di funzioni utilizzando il AWS CLI

1. Utilizzate il list-sol-function-packagescomando per elencare i pacchetti di funzioni.

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

2. Utilizzate il get-sol-function-packagecomando per visualizzare i dettagli su un pacchetto di funzioni.

```
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
```

Scaricate un pacchetto di funzioni da AWS TNB

Scoprite come scaricare un pacchetto di funzioni dal catalogo delle funzioni di AWS TNB rete.

Console

Per scaricare un pacchetto di funzioni utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione sul lato sinistro della console, scegli Pacchetti di funzioni.
3. Usa la casella di ricerca per trovare il pacchetto di funzioni
4. Scegli il pacchetto di funzioni
5. Scegli Azioni, Scarica.

AWS CLI

Per scaricare un pacchetto di funzioni utilizzando AWS CLI

Utilizzate il comando get-sol-function-package-content per scaricare un pacchetto di funzioni.

```
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
```

Eliminare un pacchetto di funzioni da AWS TNB

Scopri come eliminare un pacchetto di funzioni dal catalogo delle funzioni di AWS TNB rete. Per eliminare un pacchetto di funzioni, il pacchetto deve essere disabilitato.

Console

Per eliminare un pacchetto di funzioni utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Pacchetti di funzioni.
3. Usa la casella di ricerca per trovare il pacchetto di funzioni.
4. Scegliete un pacchetto di funzioni.
5. Scegliere Actions (Operazioni), Disable (Disabilita).
6. Scegli Operazioni > Elimina.

AWS CLI

Per eliminare un pacchetto di funzioni utilizzando AWS CLI

1. Utilizzate il update-sol-function-packagecomando per disabilitare un pacchetto di funzioni.

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

2. Utilizzate il delete-sol-function-packagecomando per eliminare un pacchetto di funzioni.

```
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
```

Pacchetti di rete per AWS TNB

Un pacchetto di rete è un file.zip in formato CSAR (Cloud Service Archive) che definisce i pacchetti di funzioni che desideri distribuire e l' AWS infrastruttura su cui desideri distribuirli.

Attività

- Crea un pacchetto di rete in AWS TNB
- Visualizza un pacchetto di rete in AWS TNB
- Scarica un pacchetto di rete da AWS TNB
- Eliminare un pacchetto di rete da AWS TNB

Crea un pacchetto di rete in AWS TNB

Un pacchetto di rete è costituito da un file descrittore del servizio di rete (NSD) (obbligatorio) e da qualsiasi file aggiuntivo (opzionale), ad esempio script specifici per le esigenze dell'utente. Ad esempio, se nel pacchetto di rete sono presenti più pacchetti di funzioni, è possibile utilizzare il NSD per definire quali funzioni di rete devono essere eseguite in determinate VPCs sottoreti o cluster AmazonEKS.

Crea un pacchetto di rete dopo aver creato i pacchetti di funzioni. Dopo aver creato un pacchetto di rete, è necessario creare un'istanza di rete.

Console

Per creare un pacchetto di rete utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Pacchetti di rete.
3. Scegli Crea pacchetto di rete.
4. Scegli i file e carica ciascuno NSD come `.zip` file. Puoi caricare un massimo di 10 file.
5. Scegli Next (Successivo).
6. Controlla i dettagli del pacchetto.
7. Scegli Crea pacchetto di rete.

AWS CLI

Per creare un pacchetto di rete utilizzando AWS CLI

1. Utilizzare il create-sol-network-packagecomando per creare un pacchetto di rete.

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

2. Utilizzate il comando put-sol-network-package-content per caricare il contenuto del pacchetto di rete. Per esempio:

```
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
```

Visualizza un pacchetto di rete in AWS TNB

Scopri come visualizzare il contenuto di un pacchetto di rete.

Console

Per visualizzare un pacchetto di rete utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Pacchetti di rete.
3. Usa la casella di ricerca per trovare il pacchetto di rete.

AWS CLI

Per visualizzare un pacchetto di rete utilizzando il AWS CLI

1. Utilizzate il list-sol-network-packagescomando per elencare i pacchetti di rete.

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

2. Utilizzate il get-sol-network-packagecomando per visualizzare i dettagli su un pacchetto di rete.

```
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
```

Scarica un pacchetto di rete da AWS TNB

Scopri come scaricare un pacchetto di rete dal catalogo dei servizi di AWS TNB rete.

Console

Per scaricare un pacchetto di rete utilizzando la console

1. Apri la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Pacchetti di rete.
3. Usa la casella di ricerca per trovare il pacchetto di rete
4. Scegli il pacchetto di rete.
5. Scegli Azioni, Scarica.

AWS CLI

Per scaricare un pacchetto di rete utilizzando AWS CLI

- Utilizzare il comando get-sol-network-package-content per scaricare un pacchetto di rete.

```
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
```

Eliminare un pacchetto di rete da AWS TNB

Scopri come eliminare un pacchetto di rete dal catalogo dei servizi di AWS TNB rete. Per eliminare un pacchetto di rete, il pacchetto deve essere disabilitato.

Console

Per eliminare un pacchetto di rete utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Pacchetti di rete.
3. Usa la casella di ricerca per trovare il pacchetto di rete
4. Scegli il pacchetto di rete
5. Scegliere Actions (Operazioni), Disable (Disabilita).
6. Scegli Operazioni > Elimina.

AWS CLI

Per eliminare un pacchetto di rete utilizzando il AWS CLI

1. Utilizzare il update-sol-network-packagecomando per disabilitare un pacchetto di rete.

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

2. Utilizzare il delete-sol-network-packagecomando per eliminare un pacchetto di rete.

```
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
```

Istanze di rete per AWS TNB

Un'istanza di rete è una singola rete creata in AWS TNB che può essere distribuita.

Attività

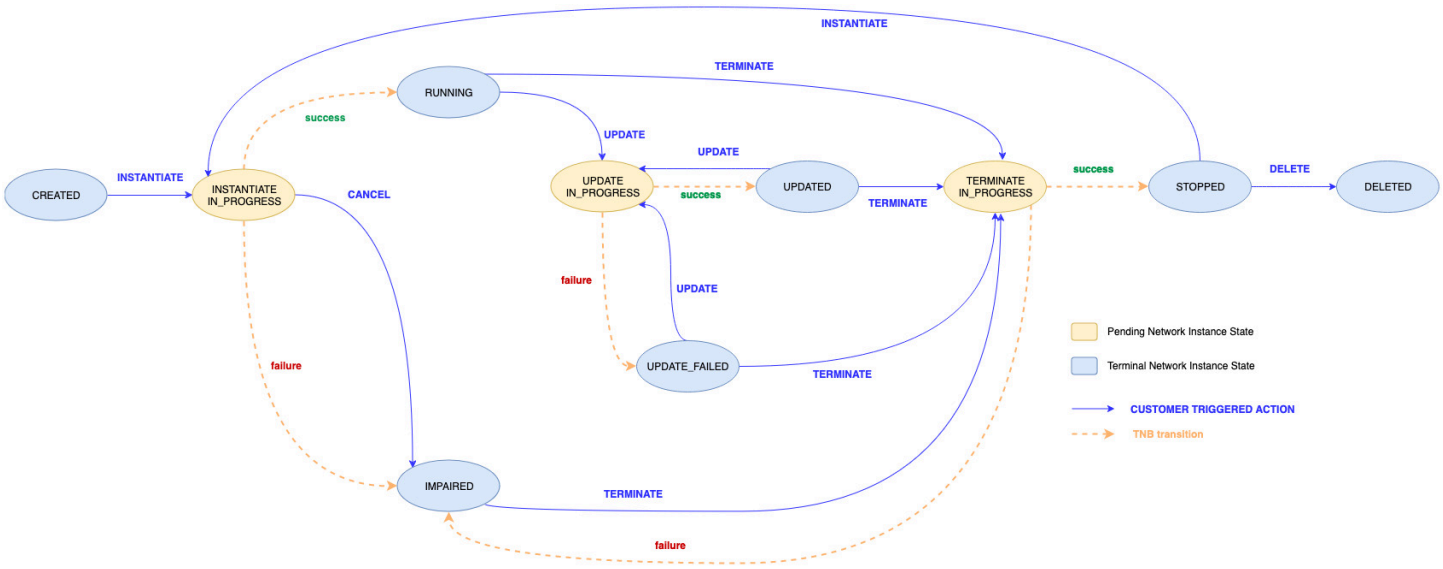
- Operazioni del ciclo di vita di un'istanza di rete
- Crea un'istanza di rete utilizzando AWS TNB
- Crea un'istanza di rete utilizzando AWS TNB
- Aggiorna un'istanza di funzione in AWS TNB
- Aggiorna un'istanza di rete in AWS TNB
- Visualizza un'istanza di rete in AWS TNB
- Termina ed elimina un'istanza di rete da AWS TNB

Operazioni del ciclo di vita di un'istanza di rete

AWS TNBconsente di gestire facilmente la rete utilizzando operazioni di gestione standardizzate in linea con ETSI SOL 003 e SOL 005. È possibile eseguire le seguenti operazioni del ciclo di vita:

- Creare la rete
- Crea un'istanza della rete
- Aggiorna la funzione di rete
- Aggiorna l'istanza di rete
- Visualizza i dettagli e lo stato della rete
- Termina la rete

L'immagine seguente mostra le operazioni di gestione della rete:



Crea un'istanza di rete utilizzando AWS TNB

Si crea un'istanza di rete dopo aver creato un pacchetto di rete. Dopo aver creato un'istanza di rete, creane un'istanza.

Console

Per creare un'istanza di rete utilizzando la console

1. Apri la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Reti.
3. Scegli Crea istanza di rete.
4. Inserisci un nome e una descrizione per l'istanza, quindi scegli Avanti.
5. Seleziona il pacchetto di rete, verifica i dettagli e scegli Avanti.
6. Scegli Crea istanza di rete.

   La nuova istanza di rete viene visualizzata nella pagina Reti. Successivamente, crea un'istanza di questa istanza di rete.

AWS CLI

Per creare un'istanza di rete utilizzando il AWS CLI

- Utilizzare il create-sol-network-instancecomando per creare un'istanza di rete.

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

Quindi, crea un'istanza di questa istanza di rete.

Crea un'istanza di rete utilizzando AWS TNB

Dopo aver creato un'istanza di rete, è necessario crearne un'istanza. Quando si crea un'istanza di rete, si effettua il AWS TNB provisioning dell' AWS infrastruttura necessaria, si distribuiscono funzioni di rete containerizzate e si configura la gestione della rete e degli accessi per creare un servizio di rete completamente operativo.

Console

Per creare un'istanza di rete utilizzando la console

1. Apri la AWS TNB console all'indirizzo. https://console.aws.amazon.com/tnb/
2. Nel riquadro di navigazione, scegli Reti.
3. Seleziona l'istanza di rete di cui desideri creare un'istanza.
4. Scegli Azioni, quindi Crea istanza.
5. Nella pagina Instantiate network, rivedi i dettagli e, facoltativamente, aggiorna i valori dei parametri.

   Gli aggiornamenti ai valori dei parametri si applicano solo a questa istanza di rete. I parametri nei VNFD pacchetti NSD and non cambiano.
6. Scegli Instantiate network.

   Viene visualizzata la pagina sullo stato della distribuzione.
7. Utilizza l'icona Aggiorna per tenere traccia dello stato di distribuzione dell'istanza di rete. È inoltre possibile abilitare l'aggiornamento automatico nella sezione Attività di distribuzione per tenere traccia dell'avanzamento di ciascuna attività.

   Quando lo stato di distribuzione cambia in`Completed`, viene creata un'istanza dell'istanza di rete.

AWS CLI

Per creare un'istanza di rete utilizzando il AWS CLI

1. Utilizzare il instantiate-sol-network-instancecomando per creare un'istanza di rete.

```
aws tnb instantiate-sol-network-instance --ns-instance-id ^ni-[a-f0-9]{17}$ --
additional-params-for-ns "{\"param1\": \"value1\", \"param2\": \"value2\"}"
```

2. Quindi, visualizza lo stato del funzionamento della rete.

Aggiorna un'istanza di funzione in AWS TNB

Dopo aver creato un'istanza di rete, è possibile aggiornare un pacchetto di funzioni nell'istanza di rete.

Console

Per aggiornare un'istanza di funzione utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Reti.
3. Seleziona l'istanza di rete. È possibile aggiornare un'istanza di rete solo se il relativo stato è`Instantiated`.

   Viene visualizzata la pagina dell'istanza di rete.
4. Dalla scheda Funzioni, selezionare l'istanza della funzione da aggiornare.
5. Scegli Aggiorna.
6. Inserisci le sostituzioni di aggiornamento.
7. Scegli Aggiorna.

AWS CLI

Utilizzate il CLI per aggiornare un'istanza di funzione

Utilizzate il update-sol-network-instancecomando con il tipo di `MODIFY_VNF_INFORMATION` aggiornamento per aggiornare un'istanza di funzione in un'istanza di rete.

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

Aggiorna un'istanza di rete in AWS TNB

Dopo aver creato un'istanza di rete, potrebbe essere necessario aggiornare l'infrastruttura o l'applicazione. A tale scopo, si aggiornano il pacchetto di rete e i valori dei parametri per l'istanza di rete e si distribuisce l'operazione di aggiornamento per applicare le modifiche.

Considerazioni

- È possibile aggiornare un'istanza di rete che si trova nello `Updated` stato `Instantiated` or.
- Quando si aggiorna un'istanza di rete, `UpdateSolNetworkService` API utilizza il nuovo pacchetto di rete e i valori dei parametri per aggiornare la topologia dell'istanza di rete.
- AWS TNBverifica che il numero di VNFD parametri NSD e nell'istanza di rete non superi 200. Questo limite viene applicato per evitare che i malintenzionati trasmettano payload errati o ingenti che influiscono sul servizio.

Parametri che è possibile aggiornare

È possibile aggiornare i seguenti parametri quando si aggiorna un'istanza di rete istanziata:

Parametro	Descrizione	Esempio: Prima	Esem dopo

Versione Amazon EKS cluster	Puoi aggiornare il valore del `version` parametro del piano di controllo del EKS cluster Amazon alla versione secondaria successiva. Non puoi effettuare il downgrade della versione. I nodi di lavoro non vengono aggiornati.	`EKSCluster:`
		`  type: tosca.nod`
		`es.AWS.Compute.EKS`
		`  properties:`
		`  version: "1.28"`
			`EKSC`
			`r:`
			
			`typ`
			`tos`
			`es.A`
			`mput`

Parametro	Descrizione	Esempio: Prima	Esem dopo
			pro s: ver "1.

Parametro	Descrizione	Esempio: Prima	Esem dopo

Proprietà di ridimensionamento	È possibile aggiornare le proprietà di ridimensionamento dei nodi `EKSManagedNode` and `EKSSelfManagedNode` TOSCA.	`EKSNodeGroup01:`
		`  ...`
		`  scaling:`
		`  properties:`
		`  desired_size: 1`
		`  min_size: 1`
		`  max_size: 1`
			`EKSN oup0`
			`...`
			`sca`
			`pro s:`
			`des ize:`

Parametro	Descrizione	Esempio: Prima	Esempio dopo
			mir max

Parametro	Descrizione	Esempio: Prima	Esem dopo

Proprietà EBS CSI del plugin Amazon	Puoi abilitare o disabilitare il EBS CSI plug-in Amazon sui tuoi EKS cluster Amazon. Puoi anche modificare la versione del plugin.	`EKSCluster:`
		`  capabilities:`
		`  ...`
		`  ebs_csi:`
		`  properties:`
		`  enabled: false`
			`EKSC`
			`r:`
			`cap`
			`ies:`
			`...`
			`ebs`
			`pro`
			`s:`
			`ena`
			`ver`
			`"v1`
			`e`
			`ksbu`
			`"`

Parametro	Descrizione	Esempio: Prima	Esem dopo

VNF	È possibile fare riferimento VNFs a tali NSD file e distribuirli nel cluster creato NSD utilizzando VNFDeployment TOSCA node. Come parte dell'aggiornamento, potrai aggiungere, aggiornare ed VNFs eliminare dalla rete.	`vnfds:`
		`  - descriptor_id: "43c012fa-2616-41a8-a833-0dfd4c5a049e  "`
		`    namespace: " vnf1"`
		`  - descriptor_id: "64222f98-ecd6-4871-bf94-7354b53f3ee5  "`
		`    namespace: "vnf2" // Deleted VNF`
		`...`
		`SampleVNF1HelmDeploy:`
		`      type: tosca.nodes.AWS.Deployment.VNFDeployment`
		`      requirements:`
		`        cluster: EKSCluster`
		`        vnfs:`
		`          - vnf1.SampleVNF1`
		`          - vnf2.SampleVNF2`
			`vnfd`
			` -`
			` des`
			`r_id`
			` "59`
			`79e9`
			`-`
			`be53`
			`2ad0`
			` "`
			` nam`
			`:`
			` "vn`
			` Upd`
			` VNF`
			` -`
			` des`
			`r_id`
			` "b7`
			`839d`
			`-916`
			`a166`
			` "`
			` nam`
			`:`
			` "vn`
			` Add`
			` VNF`
			`....`
			`Sa`
			`mple`

Parametro	Descrizione	Esempio: Prima	Esem dopo
			elmD : typ tos es.A ploy VNFD ment rec nts: clu EKS r vnf

Parametro	Descrizione	Esempio: Prima	Esempio: dopo
			- v leVN - v leVN

Parametro	Descrizione	Esempio: Prima	Esem dopo

Ganci	Per eseguire le operazioni del ciclo di vita prima e dopo la creazione di una funzione di rete, aggiungete gli `post_create` hook `pre_create` and al `VNFDeployment` nodo.
	
	In questo esempio, l'`PreCreateHook` hook verrà eseguito prima dell'`vnf3.SampleVNF3` istanziazione e l'`PostCreateHook` hook verrà eseguito dopo `vnf3.SampleVNF3` l'istanza.
		`vnfds:`
		`  - descriptor_id: "43c012fa-2616-41a8-a833-0dfd4c5a049e"`
		`    namespace: " vnf1"`
		`  - descriptor_id: "64222f98-ecd6-4871-bf94-7354b53f3ee5"`
		`    namespace: " vnf2"`
		`...`
		`SampleVNF1HelmDeploy:`
		`  type: tosca.nodes.AWS.Deployment.VNFDeployment`
		`  requirements:`
		`    cluster: EKSCluster`
		`    vnfs:`
		`      - vnf1.SampleVNF1`
		`      - vnf2.SampleVNF2  // Removed during update`
			`vnfd`
			`-`
			`des`
			`r_id`
			`"43`
			`2616`
			`-`
			`a833`
			`d4c5`
			`"`
			`nam`
			`:`
			`"vn`
			`-`
			`des`
			`r_id`
			`"b7`
			`839d`
			`-916`
			`a166`
			`"`
			`nam`
			`:`
			`"vn`
			`....`
			`S`
			`ampl`
			`Helm`
			`y:`
			`typ`
			`tos`

Parametro	Descrizione	Esempio: Prima	Esem dopo
			es.A ploy VNFD ment rec nts: clu EKS r vnf - v leVN No cha to thi fur as the nam and uui rem the sam

Parametro	Descrizione	Esempio: Prima	Esem dopo

Parametro	Descrizione	Esempio: Prima	Esempio dopo

Ganci	Per eseguire le operazioni del ciclo di vita prima e dopo l'aggiornamento di una funzione di rete, è possibile aggiungere l'`pre_update` hook e l'`post_update` hook al `VNFDeployment` nodo.
	
	In questo esempio, `PreUpdateHook` verrà eseguito prima dell'`vnf1.SampleVNF1` aggiornamento e `PostUpdateHook` verrà eseguito dopo l'`vnf1.SampleVNF1` aggiornamento al `vnf` pacchetto indicato dall'aggiornamento `uuid` per il namespace vnf1.
		`vnfds:`
		`  - descriptor_id: "43c012fa-2616-41a8-a833-0dfd4c5a049e"`
		`    namespace: "vnf1"`
		`  - descriptor_id: "64222f98-ecd6-4871-bf94-7354b53f3ee5"`
		`    namespace: "vnf2"`
		`...`
		
		`SampleVNF1HelmDeploy:`
		`  type: tosca.nodes.AWS.Deployment.VNFDeployment`
		`  requirements:`
		`    cluster: EKSCluster`
		`    vnfs:`
		`      - vnf1.SampleVNF1`
		`      - vnf2.SampleVNF2`
			`vnfd`
			
			`-`
			`des`
			`r_id`
			`"0e`
			`bd87`
			`-`
			`b8a1`
			`4666`
			`"`
			
			`nam`
			`:`
			`"vn`
			
			`-`
			`des`
			`r_id`
			`"64`
			`ecd6`
			`-`
			`bf94`
			`4b53`
			`"`
			
			`nam`
			`:`
			`"vn`
			`...`
			`S`
			`ampl`
			`Helm`
			`y:`
			
			`typ`

Parametro	Descrizione	Esempio: Prima	Esempio dopo
			tos es.A ploy VNFD ment rec nts: clu EKS r vnf - v leVN A VNF upd as the uui cha for nam "vr - v

Parametro	Descrizione	Esempio: Prima	Esem dopo
			leVN No cha to thi fur as nam and uui rem the sam int s: Hoo pre e: Hook pos te: eHoo

Aggiornamento di un'istanza di rete

Console

Per aggiornare un'istanza di rete utilizzando la console

1. Apri la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Reti.
3. Seleziona l'istanza di rete. È possibile aggiornare un'istanza di rete solo se il relativo stato è `Instantiated` o`Updated`.
4. Scegli Azioni e aggiorna.

   Viene visualizzata la pagina Aggiorna istanza con i dettagli della rete e un elenco di parametri nell'infrastruttura corrente.
5. Scegli un nuovo pacchetto di rete.

   I parametri del nuovo pacchetto di rete vengono visualizzati nella sezione Parametri aggiornati.
6. Facoltativamente, aggiorna i valori dei parametri nella sezione Parametri aggiornati. Per l'elenco dei valori dei parametri che è possibile aggiornare, vedereParametri che è possibile aggiornare.
7. Scegli Aggiorna rete.

   AWS TNBconvalida la richiesta e avvia la distribuzione. Viene visualizzata la pagina sullo stato della distribuzione.
8. Utilizza l'icona Aggiorna per tenere traccia dello stato di distribuzione dell'istanza di rete. È inoltre possibile abilitare l'aggiornamento automatico nella sezione Attività di distribuzione per tenere traccia dell'avanzamento di ciascuna attività.

   Quando lo stato di distribuzione cambia in`Completed`, l'istanza di rete viene aggiornata.
9. - Se la convalida fallisce, l'istanza di rete rimane nello stesso stato in cui si trovava prima della richiesta dell'aggiornamento, `Instantiated` oppure`Updated`.
   - Se l'aggiornamento fallisce, viene visualizzato `Update failed` lo stato dell'istanza di rete. Scegli il link per ogni operazione non riuscita per determinarne il motivo.
   - Se l'aggiornamento ha esito positivo, viene visualizzato `Updated` lo stato dell'istanza di rete.

AWS CLI

Utilizzare il CLI per aggiornare un'istanza di rete

Utilizzate il update-sol-network-instancecomando con il tipo di UPDATE_NS aggiornamento per aggiornare un'istanza di rete.

```
aws tnb update-sol-network-instance --ns-instance-id ^ni-[a-f0-9]{17}$ --
update-type UPDATE_NS --update-ns "{\"nsdInfoId\":\"^np-[a-f0-9]{17}$\",
 \"additionalParamsForNs\": {\"param1\": \"value1\"}}"
```

Visualizza un'istanza di rete in AWS TNB

Scopri come visualizzare un'istanza di rete.

Console

Per visualizzare un'istanza di rete utilizzando la console

1. Apri la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Istanze di rete.
3. Usa la casella di ricerca per trovare l'istanza di rete.

AWS CLI

Per visualizzare un'istanza di rete utilizzando il AWS CLI

1. Usa il list-sol-network-instancescomando per elencare le tue istanze di rete.

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

2. Usa il get-sol-network-instancecomando per visualizzare i dettagli su un'istanza di rete specifica.

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

Termina ed elimina un'istanza di rete da AWS TNB

Per eliminare un'istanza di rete, l'istanza deve trovarsi in uno stato terminato.

Console

Per terminare ed eliminare un'istanza di rete utilizzando la console

1. Apri la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Reti.
3. Seleziona l'ID dell'istanza di rete.
4. Scegliere Terminate (Termina).
5. Quando viene richiesta la conferma, inserisci l'ID e scegli Termina.
6. Aggiorna per tenere traccia dello stato dell'istanza di rete.
7. (Facoltativo) Seleziona l'istanza di rete e scegli Elimina.

AWS CLI

Per terminare ed eliminare un'istanza di rete utilizzando il AWS CLI

1. Utilizzare il terminate-sol-network-instancecomando per terminare un'istanza di rete.

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

2. (Facoltativo) Utilizzate il delete-sol-network-instancecomando per eliminare un'istanza di rete.

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

Operazioni di rete per AWS TNB

Un'operazione di rete è qualsiasi operazione eseguita sulla rete, ad esempio l'istanziazione o la chiusura di un'istanza di rete.

Attività

- Visualizza un'operazione di rete AWS TNB
- Annullare un'operazione AWS TNB di rete

Visualizza un'operazione di rete AWS TNB

Visualizza i dettagli di un'operazione di rete, comprese le attività coinvolte nel funzionamento della rete e lo stato delle attività.

Console

Per visualizzare un'operazione di rete utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Istanze di rete.
3. Usa la casella di ricerca per trovare l'istanza di rete.
4. Nella scheda Distribuzioni, scegli l'operazione di rete.

AWS CLI

Per visualizzare un'operazione di rete utilizzando il AWS CLI

1. Utilizzare il list-sol-network-operationscomando per elencare tutte le operazioni di rete.

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

2. Utilizzare il get-sol-network-operationcomando per visualizzare i dettagli su un'operazione di rete.

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

Annullare un'operazione AWS TNB di rete

Scopri come annullare un'operazione di rete.

Console

Per annullare un'operazione di rete utilizzando la console

1. Aprire la AWS TNB console all'indirizzo https://console.aws.amazon.com/tnb/.
2. Nel riquadro di navigazione, scegli Reti.
3. Seleziona l'ID della rete per aprirne la pagina dei dettagli.
4. Nella scheda Distribuzioni, scegli Funzionamento di rete.
5. Scegli Annulla operazione.

AWS CLI

Per annullare un'operazione di rete utilizzando il AWS CLI

Utilizzare il cancel-sol-network-operationcomando per annullare un'operazione di rete.

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

TOSCAriferimento per AWS TNB

Topology and Orchestration Specification for Cloud Applications (TOSCA) è una sintassi dichiarativa che viene CSPs utilizzata per descrivere una topologia dei servizi Web basati sul cloud, i relativi componenti, le relazioni e i processi che li gestiscono. CSPsdescrivi i punti di connessione, i collegamenti logici tra i punti di connessione e le politiche come l'affinità e la sicurezza in un modello. TOSCA CSPsquindi carica il modello in AWS TNB cui sintetizza le risorse necessarie per stabilire una rete 5G funzionante tra AWS le zone di disponibilità.

Indice

VNFDmodello

Definisce un modello di descrittore di funzioni di rete virtuale (VNFD).

Sintassi

```
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
```

Modello di topologia

`node_templates`

I TOSCA AWS nodi. I nodi possibili sono:

- AWS.VNF
- AWS.Artifacts.Helm

AWS.VNF

Definisce un nodo di funzione di rete AWS virtuale (VNF).

Sintassi

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

Proprietà

`descriptor_id`

Il UUID del descrittore.

Campo obbligatorio: sì

Tipo: stringa

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

`descriptor_version`

La versione di. VNFD

Campo obbligatorio: sì

Tipo: stringa

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

`descriptor_name`

Il nome del descrittore.

Campo obbligatorio: sì

Tipo: stringa

`provider`

L'autore di. VNFD

Campo obbligatorio: sì

Tipo: stringa

Requisiti

`helm`

La directory Helm che definisce gli artefatti del contenitore. Questo è un riferimento a .Artifacts.Helm.AWS

Campo obbligatorio: sì

Tipo: stringa

Esempio

```
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

Definisce un nodo AWS Helm.

Sintassi

```
tosca.nodes.AWS.Artifacts.Helm:
```

```
  properties:
    implementation: String
```

Proprietà

`implementation`

La directory locale che contiene il grafico Helm all'interno del CSAR pacchetto.

Campo obbligatorio: sì

Tipo: stringa

Esempio

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

Modello di descrittore del servizio di rete

Definisce un modello di descrittore di servizio di rete (NSD).

Sintassi

```
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
```

Utilizzo di parametri definiti

Quando si desidera passare dinamicamente un parametro, ad esempio il CIDR blocco per il VPC nodo, è possibile utilizzare la `{ get_input: input-parameter-name }` sintassi e definire i parametri nel NSD modello. Quindi riutilizzate il parametro sullo stesso modello. NSD

L'esempio seguente mostra come definire e utilizzare i parametri:

```
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 }
```

VNFDimportare

`descriptor_id`

Il UUID descrittore.

Campo obbligatorio: sì

Tipo: stringa

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

`namespace`

Il nome univoco.

Campo obbligatorio: sì

Tipo: stringa

Modello di topologia

`node_templates`

I TOSCA AWS nodi possibili sono:

- AWS.NS
- AWS.Calcola. EKS
- AWS.Calcola. EKS. AuthRole
- AWS.Calcola. EKSManagedNode
- AWS.Calcola. EKSSelfManagedNode
- AWS.Calcola. PlacementGroup
- AWS.Calcola. UserData
- AWS.Rete. SecurityGroup
- AWS.Rete. SecurityGroupEgressRule
- AWS.Rete. SecurityGroupIngressRule
- AWS.Risorsa. Importazione
- AWS.Rete. ENI
- AWS.HookExecution
- AWS.Rete. InternetGateway
- AWS.Rete. RouteTable
- AWS.Networking.Subnet
- AWS.Distribuzione. VNFDeployment

- AWS.Rete. VPC
- AWS.Rete. NATGateway
- AWS.Rete. Percorso

AWS.NS

Definisce un nodo di servizio di AWS rete (NS).

Sintassi

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

Proprietà

`descriptor_id`

Il UUID del descrittore.

Campo obbligatorio: sì

Tipo: stringa

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

`descriptor_version`

La versione di. NSD

Campo obbligatorio: sì

Tipo: stringa

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

`descriptor_name`

Il nome del descrittore.

Campo obbligatorio: sì

Tipo: stringa

Esempio

```
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.Calcola. EKS

Fornisci il nome del cluster, la versione di Kubernetes desiderata e un ruolo che consenta al piano di controllo Kubernetes di gestire le risorse necessarie per il tuo. AWS NFs I plugin Multus Container Network Interface () sono abilitati. CNI È possibile collegare più interfacce di rete e applicare una configurazione di rete avanzata alle funzioni di rete basate su Kubernetes. È inoltre necessario specificare l'accesso agli endpoint del cluster e le sottoreti per il cluster.

Sintassi

```
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
```

Funzionalità

`multus`

Facoltativo. Proprietà che definiscono l'utilizzo dell'interfaccia di rete Multus Container (). CNI

Se include`multus`, specificate le `multus_role` proprietà `enabled` and.

`enabled`

Indica se la funzionalità Multus predefinita è abilitata.

Campo obbligatorio: sì

Tipo: Booleano

`multus_role`

Il ruolo della gestione dell'interfaccia di rete Multus.

Campo obbligatorio: sì

Tipo: stringa

`ebs_csi`

Proprietà che definiscono il driver Amazon EBS Container Storage Interface (CSI) installato nel EKS cluster Amazon.

Abilita questo plug-in per utilizzare i nodi EKS autogestiti di Amazon su AWS Outposts, AWS Local Zones o Regioni AWS. Per ulteriori informazioni, consulta il CSIdriver Amazon Elastic Block Store nella Amazon EKS User Guide.

`enabled`

Indica se è installato il EBS CSI driver Amazon predefinito.

Campo obbligatorio: no

Tipo: Booleano

`version`

La versione del componente aggiuntivo Amazon EBS CSI driver. La versione deve corrispondere a una delle versioni restituite dall'DescribeAddonVersionsazione. Per ulteriori informazioni, DescribeAddonVersionsconsulta Amazon EKS API Reference

Required: No

Tipo: stringa

Proprietà

`version`

La versione Kubernetes per il cluster. AWS Telco Network Builder supporta le versioni di Kubernetes da 1.23 a 1.30.

Campo obbligatorio: sì

Tipo: stringa

Valori possibili: 1,23 | 1,24 | 1,25 | 1,26 | 1,27 | 1,28 | 1,29 | 1,30

`access`

L'accesso agli endpoint del cluster.

Campo obbligatorio: sì

Tipo: stringa

Valori possibili: `PRIVATE` | `PUBLIC` | `ALL`

`cluster_role`

Il ruolo della gestione dei cluster.

Campo obbligatorio: sì

Tipo: stringa

`tags`

Tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

`ip_family`

Indica la famiglia IP per gli indirizzi di servizio e pod nel cluster.

Valore consentito:`IPv4`, `IPv6`

Valore predefinito: `IPv4`

Required: No

Tipo: stringa

Requisiti

`subnets`

Un nodo AWS.Networking.Subnet.

Campo obbligatorio: sì

Tipo: List

Esempio

```
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.Computare. EKS. AuthRole

An AuthRole consente di aggiungere IAM ruoli al EKS cluster Amazon `aws-auth` `ConfigMap` in modo che gli utenti possano accedere al EKS cluster Amazon utilizzando un IAM ruolo.

Sintassi

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

Proprietà

`role_mappings`

Elenco di mappature che definiscono IAM i ruoli da aggiungere al cluster AmazonEKS. `aws-auth` `ConfigMap`

`arn`

Il ARN ruolo. IAM

Campo obbligatorio: sì

Tipo: stringa

`groups`

Gruppi Kubernetes da assegnare al ruolo definito in. `arn`

Campo obbligatorio: no

Tipo: List

Requisiti

`clusters`

Un AWS.Compute. EKSnodo.

Campo obbligatorio: sì

Tipo: List

Esempio

```
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.Calcola. EKSManagedNode

AWS TNBsupporta i gruppi di nodi EKS gestiti per automatizzare il provisioning e la gestione del ciclo di vita dei nodi (EC2istanze Amazon) per i cluster Amazon Kubernetes. EKS Per creare un gruppo di nodi, procedi come segue: EKS

- Scegli Amazon Machine Images (AMI) per i tuoi nodi di lavoro del cluster fornendo l'ID del AMI o il AMI tipo.
- Fornisci una coppia di EC2 chiavi Amazon per SSH l'accesso e le proprietà di scalabilità per il tuo gruppo di nodi.
- Assicurati che il tuo gruppo di nodi sia associato a un EKS cluster Amazon.

- Fornisci le sottoreti per i nodi di lavoro.
- Facoltativamente, allega gruppi di sicurezza, etichette di nodi e un gruppo di posizionamento al tuo gruppo di nodi.

Sintassi

```
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
```

Funzionalità

`compute`

Proprietà che definiscono i parametri di calcolo per il gruppo di nodi EKS gestiti da Amazon, ad esempio i tipi di EC2 istanze Amazon e le EC2 istanze AmazonAMIs.

`ami_type`

Il AMI tipo EKS supportato da Amazon.

Campo obbligatorio: sì

Tipo: stringa

Valori possibili: `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`

L'ID diAMI.

Required: No

Tipo: stringa

> **Note**
>
> Se entrambi `ami_type` `ami_id` sono specificati nel modello, AWS TNB utilizzerà solo il `ami_id` valore da creare`EKSManagedNode`.

`instance_types`

La dimensione dell'istanza.

Campo obbligatorio: sì

Tipo: List

`key_pair`

La coppia di EC2 chiavi per abilitare SSH l'accesso.

Campo obbligatorio: sì

Tipo: stringa

`root_volume_encryption`

Abilita EBS la crittografia Amazon per il volume EBS root di Amazon. Se questa proprietà non viene fornita, AWS TNB crittografa i volumi EBS root di Amazon per impostazione predefinita.

Campo obbligatorio: no

Impostazione predefinita: true

Tipo: Booleano

`root_volume_encryption_key_arn`

La ARN AWS KMS chiave. AWS TNBsupporta chiavi normaliARN, chiavi multiregionali ARN e aliasARN.

Required: No

Tipo: stringa

> **Note**
>
> - Se `root_volume_encryption` è falso, non includerlo. `root_volume_encryption_key_arn`
> - AWS TNBsupporta la crittografia del volume root di Amazon EBS -backedAMI.
> - Se il volume root AMI è già crittografato, è necessario includere il modulo `root_volume_encryption_key_arn` per AWS TNB ricrittografare il volume root.
> - Se il AMI volume principale non è crittografato, AWS TNB utilizza il `root_volume_encryption_key_arn` per crittografare il volume principale.
>
> Se non lo includi`root_volume_encryption_key_arn`, AWS TNB utilizza la chiave predefinita fornita da AWS Key Management Service per crittografare il volume principale.
> - AWS TNBnon decrittografa un file crittografato. AMI

`scaling`

Proprietà che definiscono i parametri di scalabilità per il gruppo di nodi EKS gestiti da Amazon, ad esempio il numero desiderato di EC2 istanze Amazon e il numero minimo e massimo di EC2 istanze Amazon nel gruppo di nodi.

`desired_size`

Il numero di istanze incluse in questo file. NodeGroup

Campo obbligatorio: sì

Tipo: integer

`min_size`

Il numero minimo di istanze in questo campo. NodeGroup

Campo obbligatorio: sì

Tipo: integer

`max_size`

Il numero massimo di istanze in questo campo. NodeGroup

Campo obbligatorio: sì

Tipo: integer

Proprietà

`node_role`

Il ARN IAM ruolo associato all'EC2istanza Amazon.

Campo obbligatorio: sì

Tipo: stringa

`tags`

I tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

Requisiti

`cluster`

Un AWS.Compute. EKSnodo.

Campo obbligatorio: sì

Tipo: stringa

`subnets`

Un nodo AWS.Networking.Subnet.

Campo obbligatorio: sì

Tipo: List

`network_interfaces`

Un AWS.Networking. ENInodo. Assicurati che le interfacce di rete e le sottoreti siano impostate sulla stessa zona di disponibilità o l'istanziazione avrà esito negativo.

Quando si impostanetwork_interfaces, AWS TNB ottiene l'autorizzazione relativa alla ENIs multus_role proprietà se è stata inclusa la proprietà in .Compute. multus AWS EKSnodo. Altrimenti, AWS TNB ottiene l'autorizzazione relativa alla ENIs proprietà node_role.

Campo obbligatorio: no

Tipo: List

`security_groups`

Un .Networking.AWS SecurityGroupnodo.

Campo obbligatorio: no

Tipo: List

`placement_group`

Un tosca.nodes.AWS.Calcola. PlacementGroupnodo.

Required: No

Tipo: stringa

`user_data`

Un tosca.nodes.AWS.Calcola. UserDatariferimento al nodo. Uno script di dati utente viene passato alle EC2 istanze Amazon lanciate dal gruppo di nodi gestiti. Aggiungi le autorizzazioni necessarie per eseguire dati utente personalizzati al node_role passato al gruppo di nodi.

Required: No

Tipo: stringa

`labels`

Un elenco di etichette di nodi. L'etichetta di un nodo deve avere un nome e un valore. Crea un'etichetta utilizzando i seguenti criteri:

- Il nome e il valore devono essere separati da=.
- Il nome e il valore possono avere ciascuno una lunghezza massima di 63 caratteri.
- L'etichetta può includere lettere (A-Z, a-z), numeri (0-9) e i seguenti caratteri: `[-, _, ., *, ?]`
- Il nome e il valore devono iniziare e terminare con un carattere alfanumerico o. `? *`

Ad esempio, `myLabelName1=*NodeLabelValue1`.

Campo obbligatorio: no

Tipo: List

Esempio

```
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.Calcola. EKSSelfManagedNode

AWS TNBsupporta i nodi EKS autogestiti di Amazon per automatizzare il provisioning e la gestione del ciclo di vita dei nodi (istanze AmazonEC2) per i cluster Amazon Kubernetes. EKS Per creare un gruppo di EKS nodi Amazon, procedi come segue:

- Scegli Amazon Machine Images (AMI) per i tuoi nodi di lavoro del cluster fornendo l'ID diAMI.
- Fornisci una coppia di EC2 chiavi Amazon per SSH l'accesso.
- Assicurati che il tuo gruppo di nodi sia associato a un EKS cluster Amazon.
- Fornisci il tipo di istanza e le dimensioni desiderate, minime e massime.
- Fornisci le sottoreti per i nodi di lavoro.
- Facoltativamente, allega gruppi di sicurezza, etichette di nodi e un gruppo di posizionamento al tuo gruppo di nodi.

Sintassi

```
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
```

Funzionalità

compute

Proprietà che definiscono i parametri di calcolo per i nodi EKS autogestiti di Amazon, come i tipi di EC2 istanze Amazon e le EC2 istanze AMIs Amazon.

`ami_id`

L'AMIID utilizzato per avviare l'istanza. AWS TNBsupporta istanze che IMDSv2 sfruttano. Per ulteriori informazioni, consulta IMDSversione.

Campo obbligatorio: sì

Tipo: stringa

`instance_type`

La dimensione dell'istanza.

Campo obbligatorio: sì

Tipo: stringa

`key_pair`

La coppia di EC2 chiavi Amazon per abilitare SSH l'accesso.

Campo obbligatorio: sì

Tipo: stringa

`root_volume_encryption`

Abilita EBS la crittografia Amazon per il volume EBS root di Amazon. Se questa proprietà non viene fornita, AWS TNB crittografa i volumi EBS root di Amazon per impostazione predefinita.

Campo obbligatorio: no

Impostazione predefinita: true

Tipo: Booleano

`root_volume_encryption_key_arn`

La ARN AWS KMS chiave. AWS TNBsupporta chiavi normaliARN, chiavi multiregionali ARN e aliasARN.

Required: No

Tipo: stringa

> **Note**
>
> - Se `root_volume_encryption` è falso, non includerlo. `root_volume_encryption_key_arn`
> - AWS TNBsupporta la crittografia del volume root di Amazon EBS -backedAMI.
> - Se il volume root AMI è già crittografato, è necessario includere il modulo `root_volume_encryption_key_arn` per AWS TNB ricrittografare il volume root.
> - Se il AMI volume principale non è crittografato, AWS TNB utilizza il `root_volume_encryption_key_arn` per crittografare il volume principale.
>
>   Se non lo includi`root_volume_encryption_key_arn`, AWS TNB viene utilizzato AWS Managed Services per crittografare il volume root.
> - AWS TNBnon decrittografa un file crittografato. AMI

scaling

Proprietà che definiscono i parametri di scalabilità per i nodi EKS autogestiti di Amazon, ad esempio il numero desiderato di EC2 istanze Amazon e il numero minimo e massimo di EC2 istanze Amazon nel gruppo di nodi.

`desired_size`

Il numero di istanze in esso contenute. NodeGroup

Campo obbligatorio: sì

Tipo: integer

`min_size`

Il numero minimo di istanze in questo campo. NodeGroup

Campo obbligatorio: sì

Tipo: integer

`max_size`

Il numero massimo di istanze in questo campo. NodeGroup

Campo obbligatorio: sì

Tipo: integer

Proprietà

`node_role`

Il ARN IAM ruolo associato all'EC2istanza Amazon.

Campo obbligatorio: sì

Tipo: stringa

`tags`

I tag da allegare alla risorsa. I tag verranno propagati alle istanze create dalla risorsa.

Campo obbligatorio: no

Tipo: List

Requisiti

`cluster`

Un AWS.Compute. EKSnodo.

Campo obbligatorio: sì

Tipo: stringa

`subnets`

Un nodo AWS.Networking.Subnet.

Campo obbligatorio: sì

Tipo: List

`network_interfaces`

Un AWS.Networking. ENInodo. Assicurati che le interfacce di rete e le sottoreti siano impostate sulla stessa zona di disponibilità o l'istanziazione avrà esito negativo.

Quando si impostanetwork_interfaces, AWS TNB ottiene l'autorizzazione relativa alla ENIs multus_role proprietà se è stata inclusa la proprietà in .Compute. multus AWS EKSnodo. Altrimenti, AWS TNB ottiene l'autorizzazione relativa alla ENIs proprietà node_role.

Campo obbligatorio: no

Tipo: List

`security_groups`

Un .Networking.AWS SecurityGroupnodo.

Campo obbligatorio: no

Tipo: List

`placement_group`

Un tosca.nodes.AWS.Calcola. PlacementGroupnodo.

Required: No

Tipo: stringa

`user_data`

Un tosca.nodes.AWS.Calcola. UserDatariferimento al nodo. Uno script di dati utente viene passato alle EC2 istanze Amazon lanciate dal gruppo di nodi autogestito. Aggiungi le autorizzazioni necessarie per l'esecuzione di dati utente personalizzati al node_role passato al gruppo di nodi.

Required: No

Tipo: stringa

`labels`

Un elenco di etichette di nodi. L'etichetta di un nodo deve avere un nome e un valore. Crea un'etichetta utilizzando i seguenti criteri:

- Il nome e il valore devono essere separati da=.
- Il nome e il valore possono avere ciascuno una lunghezza massima di 63 caratteri.
- L'etichetta può includere lettere (A-Z, a-z), numeri (0-9) e i seguenti caratteri: `[-, _, ., *, ?]`
- Il nome e il valore devono iniziare e terminare con un carattere alfanumerico o. `? *`

Ad esempio, `myLabelName1=*NodeLabelValue1`.

Campo obbligatorio: no

Tipo: List

Esempio

```
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.Calcola. PlacementGroup

Un PlacementGroup nodo supporta diverse strategie per posizionare le EC2 istanze Amazon.

Quando avvii un nuovo AmazonEC2instance, il EC2 servizio Amazon tenta di collocare l'istanza in modo tale che tutte le istanze siano distribuite sull'hardware sottostante per ridurre al minimo i guasti correlati. I gruppi di collocamento consentono comunque di influire sul collocamento di un gruppo di istanze interdipendenti per soddisfare le esigenze del carico di lavoro.

Sintassi

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

Proprietà

`strategy`

La strategia da utilizzare per posizionare le EC2 istanze Amazon.

Campo obbligatorio: sì

Tipo: stringa

Valori possibili: CLUSTER | PARTITION | SPREAD _ HOST | SPREAD _ RACK

- CLUSTER— raggruppa le istanze ravvicinate all'interno di una zona di disponibilità. Questa strategia consente ai carichi di lavoro di raggiungere le prestazioni di rete a bassa latenza necessarie per node-to-node comunicazioni strettamente accoppiate, tipiche delle applicazioni di elaborazione ad alte prestazioni (). HPC
- PARTITION— distribuisce le istanze su partizioni logiche in modo che i gruppi di istanze in una partizione non condividano l'hardware sottostante con gruppi di istanze in partizioni diverse. Questa strategia di solito viene utilizzata in grandi carichi di lavoro distribuiti e replicati, come Hadoop, Cassandra e Kafka.
- SPREAD_ RACK — colloca un piccolo gruppo di istanze su hardware sottostante distinto per ridurre i guasti correlati.
- SPREAD_ HOST — utilizzato solo con i gruppi di collocamento Outpost. Posiziona un piccolo gruppo di istanze su hardware sottostante distinto per ridurre i guasti correlati.

`partition_count`

Il numero di partizioni.

Obbligatorio: richiesto solo quando `strategy` è impostato su. `PARTITION`

Tipo: integer

Valori possibili: 1 | 2 | 3 | 4 | 5 | 6 | 7

`tags`

I tag che potete allegare alla risorsa del gruppo di collocamento.

Campo obbligatorio: no

Tipo: List

Esempio

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

AWS.Calcola. UserData

AWS TNBsupporta l'avvio di EC2 istanze Amazon con dati utente personalizzati, tramite il UserData nodo in Network Service Descriptor (). NSD Per ulteriori informazioni sui dati utente personalizzati, consulta Dati utente e script di shell nella Amazon EC2 User Guide.

Durante l'istanza di rete, AWS TNB fornisce la registrazione dell'EC2istanza Amazon al cluster tramite uno script di dati utente. Quando vengono forniti anche dati utente personalizzati, AWS TNB unisce entrambi gli script e li trasmette come script multimima ad Amazon. EC2 Lo script personalizzato per i dati utente viene eseguito prima dello script di EKS registrazione Amazon.

Per utilizzare variabili personalizzate nello script userdata, aggiungi un punto esclamativo `!` dopo il corsetto riccio aperto. `{` Ad esempio, per utilizzarle `MyVariable` nello script, inserisci: `{! MyVariable}`

> **Note**
>
> - AWS TNBsupporta script di dati utente di dimensioni fino a 7 KB.
> - Poiché AWS TNB AWS CloudFormation gli utenti elaborano e renderizzano lo script `multimime` dei dati utente, assicuratevi che lo script rispetti tutte le regole. AWS CloudFormation

Sintassi

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

```
    content_type: String
```

Proprietà

`implementation`

Il percorso relativo alla definizione dello script dei dati utente. Il formato deve essere: `./scripts/script_name.sh`

Campo obbligatorio: sì

Tipo: stringa

`content_type`

Tipo di contenuto dello script di dati utente.

Campo obbligatorio: sì

Tipo: stringa

Valori possibili: `x-shellscript`

Esempio

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

AWS.Rete. SecurityGroup

AWS TNBsupporta i gruppi di sicurezza per automatizzare il provisioning di Amazon EC2 Security Groups che puoi collegare ai gruppi di nodi del cluster Amazon EKS Kubernetes.

Sintassi

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

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

Proprietà

`description`

La descrizione del gruppo di sicurezza. È possibile utilizzare fino a 255 caratteri per descrivere il gruppo. È possibile includere solo lettere (A-Z e a-z), numeri (0-9), spazi e i seguenti caratteri speciali: ._-:/() #, @ [] +=&; {}! $*

Campo obbligatorio: sì

Tipo: stringa

`name`

Un nome per il gruppo di sicurezza. È possibile utilizzare fino a 255 caratteri per il nome. È possibile includere solo lettere (A-Z e a-z), numeri (0-9), spazi e i seguenti caratteri speciali: ._-:/() #, @ [] +=&; {}! $*

Campo obbligatorio: sì

Tipo: stringa

`tags`

I tag che puoi allegare alla risorsa del gruppo di sicurezza.

Campo obbligatorio: no

Tipo: List

Requisiti

`vpc`

Un AWS.Networking. VPCnodo.

Campo obbligatorio: sì

Tipo: stringa

Esempio

```
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.Rete. SecurityGroupEgressRule

AWS TNBsupporta le regole di uscita dei gruppi di sicurezza per automatizzare il provisioning delle Amazon EC2 Security Group Egress Rules che possono essere collegate a .Networking. AWS SecurityGroup. Tieni presente che devi fornire un cidr_ip/destination_security_group/ destination_prefix_list come destinazione per il traffico in uscita.

Sintassi

```
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
```

Proprietà

`cidr_ip`

L'IPv4intervallo di indirizzi in formato. CIDR È necessario specificare un CIDR intervallo che consenta il traffico in uscita.

Required: No

Tipo: stringa

`cidr_ipv6`

L'intervallo di IPv6 indirizzi in CIDR formato, per il traffico in uscita.
È necessario specificare un gruppo di sicurezza di destinazione (`destination_security_groupodestination_prefix_list`) o un CIDR intervallo (`cidr_ipocidr_ipv6`).

Required: No

Tipo: stringa

`description`

La descrizione di una regola in uscita del gruppo di sicurezza. È possibile utilizzare fino a 255 caratteri per descrivere la regola.

Required: No

Tipo: stringa

`destination_prefix_list`

L'ID dell'elenco di prefissi di un elenco di prefissi VPC gestito da Amazon esistente. Questa è la destinazione delle istanze del gruppo di nodi associate al gruppo di sicurezza. Per ulteriori informazioni sugli elenchi di prefissi gestiti, consulta Managed prefix lists nella Amazon VPC User Guide.

Required: No

Tipo: stringa

`from_port`

Se il protocollo è TCP oUDP, questo è l'inizio dell'intervallo di porte. Se il protocollo è ICMP oICMPv6, questo è il tipo di numero. Il valore -1 indica tutti i ICMPv6 tipi ICMP /. Se si specificano tutti i ICMPv6 tipiICMP/, è necessario specificare tutti i ICMPv6 codici ICMP /.

Campo obbligatorio: no

Tipo: integer

`ip_protocol`

Il nome del protocollo IP (tcp, udp, icmp, icmpv6) o il numero di protocollo. Usare -1 per specificare tutti i protocolli. Quando si autorizzano le regole del gruppo di sicurezza, specificando -1 o un numero di protocollo diverso da tcp, udp, icmp o icmpv6 si consente il traffico su tutte le porte, indipendentemente dall'intervallo di porte specificato. Per tcp, udp e icmp, è necessario specificare un intervallo di porte. Per icmpv6, l'intervallo di porte è facoltativo; se si omette l'intervallo di porte, è consentito il traffico per tutti i tipi e codici.

Campo obbligatorio: sì

Tipo: stringa

`to_port`

Se il protocollo è TCP oUDP, questa è la fine dell'intervallo di porte. Se il protocollo è ICMP oICMPv6, questo è il codice. Il valore -1 indica tutti i ICMPv6 codici ICMP /. Se si specificano tutti i ICMPv6 tipiICMP/, è necessario specificare tutti i ICMPv6 codici ICMP /.

Campo obbligatorio: no

Tipo: integer

Requisiti

`security_group`

L'ID del gruppo di sicurezza a cui aggiungere questa regola.

Campo obbligatorio: sì

Tipo: stringa

`destination_security_group`

L'ID o il TOSCA riferimento del gruppo di sicurezza di destinazione a cui è consentito il traffico in uscita.

Required: No

Tipo: stringa

Esempio

```
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.Rete. SecurityGroupIngressRule

AWS TNBsupporta le regole di ingresso dei gruppi di sicurezza per automatizzare il provisioning di Amazon EC2 Security Group Ingress Rules che possono essere allegate a .Networking. AWS SecurityGroup. Nota che devi fornire un cidr_ip/source_security_group/source_prefix_list come fonte per il traffico in ingresso.

Sintassi

```
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
```

Proprietà

`cidr_ip`

L'IPv4intervallo di indirizzi in formato. CIDR È necessario specificare un CIDR intervallo che consenta il traffico in ingresso.

Required: No

Tipo: stringa

`cidr_ipv6`

L'intervallo di IPv6 indirizzi in CIDR formato, per il traffico in ingresso. È necessario specificare un gruppo di sicurezza di origine (`source_security_grouposource_prefix_list`) o un CIDR intervallo (`cidr_ipocidr_ipv6`).

Required: No

Tipo: stringa

`description`

La descrizione di una regola del gruppo di sicurezza in ingresso (in entrata). È possibile utilizzare fino a 255 caratteri per descrivere la regola.

Required: No

Tipo: stringa

`source_prefix_list`

L'ID dell'elenco di prefissi di un elenco di prefissi VPC gestito da Amazon esistente. Questa è la fonte da cui le istanze del gruppo di nodi associate al gruppo di sicurezza potranno ricevere traffico. Per ulteriori informazioni sugli elenchi di prefissi gestiti, consulta Managed prefix lists nella Amazon VPC User Guide.

Required: No

Tipo: stringa

`from_port`

Se il protocollo è TCP oUDP, questo è l'inizio dell'intervallo di porte. Se il protocollo è ICMP oICMPv6, questo è il tipo di numero. Il valore -1 indica tutti i ICMPv6 tipi ICMP /. Se si specificano tutti i ICMPv6 tipiICMP/, è necessario specificare tutti i ICMPv6 codici ICMP /.

Campo obbligatorio: no

Tipo: integer

`ip_protocol`

Il nome del protocollo IP (tcp, udp, icmp, icmpv6) o il numero di protocollo. Usare -1 per specificare tutti i protocolli. Quando si autorizzano le regole del gruppo di sicurezza, specificando -1 o un numero di protocollo diverso da tcp, udp, icmp o icmpv6 si consente il traffico su tutte le porte, indipendentemente dall'intervallo di porte specificato. Per tcp, udp e icmp, è necessario specificare un intervallo di porte. Per icmpv6, l'intervallo di porte è facoltativo; se si omette l'intervallo di porte, è consentito il traffico per tutti i tipi e codici.

Campo obbligatorio: sì

Tipo: stringa

`to_port`

Se il protocollo è TCP oUDP, questa è la fine dell'intervallo di porte. Se il protocollo è ICMP oICMPv6, questo è il codice. Il valore -1 indica tutti i ICMPv6 codici ICMP /. Se si specificano tutti i ICMPv6 tipiICMP/, è necessario specificare tutti i ICMPv6 codici ICMP /.

Campo obbligatorio: no

Tipo: integer

Requisiti

`security_group`

L'ID del gruppo di sicurezza a cui aggiungere questa regola.

Campo obbligatorio: sì

Tipo: stringa

`source_security_group`

L'ID o il TOSCA riferimento del gruppo di sicurezza di origine da cui deve essere consentito il traffico in ingresso.

Required: No

Tipo: stringa

Esempio

```
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.Risorsa. Importazione

È possibile importare le seguenti AWS risorse in AWS TNB:

- VPC
- Sottorete
- Tabella di routing
- Internet Gateway
- Gruppo di sicurezza

Sintassi

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

Proprietà

`resource_type`

Il tipo di risorsa in cui viene importata AWS TNB.

Campo obbligatorio: no

Tipo: List

`resource_id`

L'ID della risorsa in cui viene importata AWS TNB.

Campo obbligatorio: no

Tipo: List

Esempio

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

AWS.Rete. ENI

Un'interfaccia di rete è un componente di rete logico in un VPC che rappresenta una scheda di rete virtuale. A un'interfaccia di rete viene assegnato un indirizzo IP automaticamente o manualmente in base alla relativa sottorete. Dopo aver distribuito un'EC2istanza Amazon in una sottorete, puoi collegare un'interfaccia di rete ad essa oppure scollegare un'interfaccia di rete da quell'istanza Amazon e ricollegarla a un'altra EC2 istanza Amazon EC2 in quella sottorete. L'indice dei dispositivi identifica la posizione nell'ordine degli allegati.

Sintassi

```
tosca.nodes.AWS.Networking.ENI:
  properties:
    device_index: Integer
    source_dest_check: Boolean
    tags: List
  requirements:
    subnet: String
    security_groups: List
```

Proprietà

`device_index`

L'indice del dispositivo deve essere maggiore di zero.

Campo obbligatorio: sì

Tipo: integer

`source_dest_check`

Indica se l'interfaccia di rete esegue il controllo di origine/destinazione. Un valore di `true` indica che il controllo è abilitato, mentre `false` indica che il controllo è disabilitato.

Valore consentito: vero, falso

Impostazione predefinita: true

Campo obbligatorio: no

Tipo: Booleano

`tags`

I tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

Requisiti

`subnet`

Un nodo AWS.Networking.Subnet.

Campo obbligatorio: sì

Tipo: stringa

`security_groups`

Un AWS.Networking. SecurityGroupnodo.

Required: No

Tipo: stringa

Esempio

```
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

Un lifecycle hook ti offre la possibilità di eseguire i tuoi script come parte dell'infrastruttura e della creazione di istanze di rete.

Sintassi

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

Funzionalità

`execution`

Proprietà del motore di esecuzione degli hook che esegue gli script hook.

`type`

Il tipo di motore di esecuzione degli hook.

Required: No

Tipo: stringa

Valori possibili: `CODE_BUILD`

Requisiti

`definition`

Un AWS. HookDefinition.nodo Bash.

Campo obbligatorio: sì

Tipo: stringa

`vpc`

Un AWS.Networking. VPCnodo.

Campo obbligatorio: sì

Tipo: stringa

Esempio

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

AWS.Rete. InternetGateway

Definisce un nodo AWS Internet Gateway.

Sintassi

```
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
```

Funzionalità

`routing`

Proprietà che definiscono la connessione di routing all'interno diVPC. È necessario includere la `ipv6_dest_cidr` proprietà `dest_cidr` o.

`dest_cidr`

Il IPv4 CIDR blocco utilizzato per la partita di destinazione. Questa proprietà viene utilizzata per creare un percorso in `RouteTable` e il suo valore viene utilizzato come`DestinationCidrBlock`.

Obbligatorio: No se hai incluso la `ipv6_dest_cidr` proprietà.

Tipo: stringa

`ipv6_dest_cidr`

Il IPv6 CIDR blocco utilizzato per la partita di destinazione.

Obbligatorio: No se hai incluso la `dest_cidr` proprietà.

Tipo: stringa

Proprietà

`tags`

I tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

`egress_only`

Una proprietà IPv6 specifica. Indica se il gateway Internet serve solo per le comunicazioni in uscita o meno. Quando `egress_only` è vero, è necessario definire la `ipv6_dest_cidr` proprietà.

Campo obbligatorio: no

Tipo: Booleano

Requisiti

`vpc`

Un AWS.Networking. VPCnodo.

Campo obbligatorio: sì

Tipo: stringa

`route_table`

Un AWS.Networking. RouteTablenodo.

Campo obbligatorio: sì

Tipo: stringa

Esempio

```
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.Rete. RouteTable

Una tabella di routing contiene un insieme di regole, chiamate route, che determinano dove viene diretto il traffico di rete proveniente dalle sottoreti all'interno del gateway VPC o del gateway. È necessario associare una tabella di routing a. VPC

Sintassi

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

Proprietà

`tags`

Tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

Requisiti

`vpc`

Un AWS.Networking. VPCnodo.

Campo obbligatorio: sì

Tipo: stringa

Esempio

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

AWS.Networking.Subnet

Una sottorete è un intervallo di indirizzi IP presenti nell'utente VPC e deve risiedere interamente all'interno di una zona di disponibilità. È necessario specificare unaVPC, un CIDR blocco, una zona di disponibilità e una tabella di routing per la sottorete. È inoltre necessario definire se la sottorete è privata o pubblica.

Sintassi

```
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
```

Proprietà

`type`

Indica se le istanze avviate in questa sottorete ricevono un indirizzo pubblico. IPv4

Campo obbligatorio: sì

Tipo: stringa

Valori possibili: `PUBLIC` | `PRIVATE`

`availability_zone`

La zona di disponibilità per la sottorete. Questo campo supporta le zone di AWS disponibilità all'interno di una AWS regione, ad esempio `us-west-2` (Stati Uniti occidentali (Oregon)). Supporta anche AWS Local Zones all'interno della Availability Zone, ad esempio`us-west-2-lax-1a`.

Campo obbligatorio: sì

Tipo: stringa

`cidr_block`

Il CIDR blocco per la sottorete.

Required: No

Tipo: stringa

`ipv6_cidr_block`

Il CIDR blocco usato per creare la IPv6 sottorete. Se includi questa proprietà, non `ipv6_cidr_block_suffix` includerla.

Required: No

Tipo: stringa

`ipv6_cidr_block_suffix`

Il suffisso esadecimale a 2 cifre del IPv6 CIDR blocco per la sottorete creata su Amazon. VPC Utilizza il seguente formato: *`2-digit hexadecimal`*`::/`*`subnetMask`*

Se includi questa proprietà, non includerla. `ipv6_cidr_block`

Required: No

Tipo: stringa

`outpost_arn`

In ARN AWS Outposts che modo verrà creata la sottorete. Aggiungi questa proprietà al NSD modello se desideri avviare nodi Amazon EKS autogestiti su AWS Outposts. Per ulteriori informazioni, consulta Amazon EKS on AWS Outposts nella Amazon EKS User Guide.

Se aggiungi questa proprietà al NSD modello, devi impostare il valore della `availability_zone` proprietà nella zona di disponibilità di AWS Outposts.

Required: No

Tipo: stringa

`tags`

I tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

Requisiti

`vpc`

Un AWS.Networking. VPCnodo.

Campo obbligatorio: sì

Tipo: stringa

`route_table`

Un AWS.Networking. RouteTablenodo.

Campo obbligatorio: sì

Tipo: stringa

Esempio

```
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.Distribuzione. VNFDeployment

Le implementazioni NF sono modellate fornendo l'infrastruttura e l'applicazione ad essa associate. L'attributo cluster specifica il cluster su cui ospitare il tuoEKS. NFs L'attributo vnfs specifica le funzioni di rete per la distribuzione. È inoltre possibile fornire operazioni opzionali di lifecycle hook di tipo pre_create e post_create per eseguire istruzioni specifiche per la distribuzione, ad esempio richiamare un sistema di gestione dell'inventario. API

Sintassi

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

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

Requisiti

`deployment`

Un .Deployment.AWS VNFDeploymentnodo.

Required: No

Tipo: stringa

`cluster`

Un AWS.Compute. EKSnodo.

Campo obbligatorio: sì

Tipo: stringa

`vnfs`

Un AWS. VNFnodo.

Campo obbligatorio: sì

Tipo: stringa

Interfacce

Ganci

Definisce la fase in cui vengono eseguiti i lifecycle hook.

`pre_create`

Un.AWS HookExecutionnodo. Questo hook viene eseguito prima della distribuzione del `VNFDeployment` nodo.

Required: No

Tipo: stringa

`post_create`

Un AWS. HookExecutionnodo. Questo hook viene eseguito dopo la distribuzione del `VNFDeployment` nodo.

Required: No

Tipo: stringa

Esempio

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

AWS.Rete. VPC

È necessario specificare un CIDR blocco per il cloud privato virtuale (VPC).

Sintassi

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

Proprietà

`cidr_block`

L'intervallo di IPv4 rete perVPC, in CIDR notazione.

Campo obbligatorio: sì

Tipo: stringa

`ipv6_cidr_block`

Il IPv6 CIDR blocco utilizzato per creare ilVPC.

Valore consentito: `AMAZON_PROVIDED`

Required: No

Tipo: stringa

`dns_support`

Indica se le istanze sono state avviate in VPC get DNS hostnames.

Campo obbligatorio: no

Tipo: Booleano

Impostazione predefinita: `false`

`tags`

Tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

Esempio

```
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.Rete. NATGateway

È possibile definire un nodo NAT Gateway pubblico o privato su una sottorete. Per un gateway pubblico, se non fornisci un ID di allocazione IP elastico, AWS TNB allocherà un IP elastico per il tuo account e lo assocerà al gateway.

Sintassi

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

Proprietà

`subnet`

Il riferimento al nodo AWS.Networking.Subnet.

Campo obbligatorio: sì

Tipo: stringa

`internet_gateway`

Il AWS file .Networking. InternetGatewayriferimento al nodo.

Campo obbligatorio: sì

Tipo: stringa

Proprietà

`type`

Indica se il gateway è pubblico o privato.

Valore consentito:`PUBLIC`, `PRIVATE`

Campo obbligatorio: sì

Tipo: stringa

`eip_allocation_id`

L'ID che rappresenta l'allocazione dell'indirizzo IP elastico.

Required: No

Tipo: stringa

`tags`

Tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

Esempio

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

AWS.Rete. Percorso

È possibile definire un nodo di route che associ la route di destinazione al NAT Gateway come risorsa di destinazione e aggiunga la route alla tabella di route associata.

Sintassi

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

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

Proprietà

`dest_cidr_blocks`

L'elenco dei IPv4 percorsi di destinazione verso la risorsa di destinazione.

Campo obbligatorio: sì

Tipo: List

Tipo di membro: String

Proprietà

`nat_gateway`

L'AWS.Networking. NATGatewayriferimento al nodo.

Campo obbligatorio: sì

Tipo: stringa

`route_table`

L'AWS.Networking. RouteTableriferimento al nodo.

Campo obbligatorio: sì

Tipo: stringa

Esempio

```
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
```

Nodi comuni

Definire i nodi per la NSD eVNFD.

- AWS. HookDefinition.Bash

AWS.HookDefinition.Bash

Definisce un AWS HookDefinition in`bash`.

Sintassi

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

Proprietà

`implementation`

Il percorso relativo alla definizione del gancio. Il formato deve essere: `./hooks/`*`script_name`*`.sh`

Campo obbligatorio: sì

Tipo: stringa

`environment_variables`

Le variabili di ambiente per lo script hook bash. Usa il seguente formato: **`envName`**`=`**`envValue`** con la seguente espressione regolare: `^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`

Assicuratevi che il **`envName`**`=`**`envValue`** valore soddisfi i seguenti criteri:

- Non utilizzate spazi.

- Inizia **envName** con una lettera (A-Z o a-z) o un numero (0-9).
- Non iniziate il nome della variabile di ambiente con le seguenti parole chiave AWS TNB riservate (senza distinzione tra maiuscole e minuscole):
  - CODEBUILD
  - TNB
  - HOME
  - AWS
- È possibile utilizzare un numero qualsiasi di lettere (A-Z o a-z), numeri (0-9) e caratteri speciali e per e. **_ envName envValue**

Esempio: `A123-45xYz=Example_789`

Campo obbligatorio: no

Tipo: List

`execution_role`

Il ruolo dell'esecuzione degli hook.

Campo obbligatorio: sì

Tipo: stringa

Esempio

```
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"
```

Sicurezza in AWS TNB

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità che si applicano a AWS Telco Network Builder, consulta AWS Services in Scope by Compliance Program by Compliance Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo AWS TNB. I seguenti argomenti mostrano come eseguire la configurazione AWS TNB per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere AWS TNB le tue risorse.

Indice

Protezione dei dati in AWS TNB

Il modello di responsabilità AWS condivisa modello di di si applica alla protezione dei dati in AWS Telco Network Builder. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i. Cloud AWS L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta la sezione Privacy dei dati FAQ. Per informazioni sulla protezione dei dati in Europa, consulta il Modello di responsabilitàAWS condivisa e GDPR il post sul blog sulla AWS sicurezza.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e di configurare i singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- UsaSSL/TLSper comunicare con AWS le risorse. Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Configurazione API e registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'uso dei CloudTrail percorsi per registrare AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di FIPS 140-3 moduli crittografici convalidati per accedere AWS tramite un'interfaccia a riga di comando o unAPI, usa un endpoint. FIPS Per ulteriori informazioni sugli FIPS endpoint disponibili, vedere Federal Information Processing Standard () 140-3. FIPS

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori AWS TNB o Servizi AWS utilizzi in altro modo la console, API AWS CLI, o. AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Se fornisci un URL a un server esterno, ti consigliamo vivamente di non includere le informazioni sulle credenziali URL per convalidare la tua richiesta a quel server.

Gestione dei dati

Quando chiudi il tuo AWS account, AWS TNB contrassegna i tuoi dati per l'eliminazione e li rimuove da qualsiasi utilizzo. Se riattivi il tuo AWS account entro 90 giorni, AWS TNB ripristina i tuoi dati. Dopo 120 giorni, elimina AWS TNB definitivamente i dati. AWS TNBinoltre chiude le reti ed elimina i pacchetti di funzioni e i pacchetti di rete.

Crittografia a riposo

AWS TNBcrittografa sempre tutti i dati archiviati nel servizio a riposo senza richiedere alcuna configurazione aggiuntiva. Questa crittografia è automatica tramite AWS Key Management Service.

Crittografia in transito

AWS TNBprotegge tutti i dati in transito utilizzando Transport Layer Security (TLS) 1.2.

È tua responsabilità crittografare i dati tra i tuoi agenti di simulazione e i loro clienti.

Riservatezza del traffico Internet

AWS TNBle risorse di elaborazione risiedono in un cloud privato virtuale (VPC) condiviso da tutti i clienti. Tutto AWS TNB il traffico interno è rimasto all'interno della AWS rete e non attraversa Internet. Le connessioni tra i tuoi agenti di simulazione e i loro clienti vengono instradate su Internet.

Gestione delle identità e degli accessi per AWS TNB

AWS Identity and Access Management (IAM) è un programma Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. IAMgli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. AWS TNB IAMè un dispositivo Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Indice

- Risoluzione dei problemi relativi all'identità e all'accesso di AWS Telco Network Builder

Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che svolgi. AWS TNB

Utente del servizio: se utilizzi il AWS TNB servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più AWS TNB funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in AWS TNB, consultaRisoluzione dei problemi relativi all'identità e all'accesso di AWS Telco Network Builder.

Amministratore del servizio: se sei responsabile delle AWS TNB risorse della tua azienda, probabilmente hai pieno accesso a AWS TNB. È tuo compito determinare a quali AWS TNB funzionalità e risorse devono accedere gli utenti del servizio. È quindi necessario inviare richieste all'IAMamministratore per modificare le autorizzazioni degli utenti del servizio. Consulta le informazioni contenute in questa pagina per comprendere i concetti di base diIAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con AWS TNB, consultaCome AWS TNB funziona con IAM.

IAMamministratore: se sei un IAM amministratore, potresti voler conoscere i dettagli su come scrivere politiche a cui gestire l'accesso AWS TNB. Per visualizzare esempi di policy AWS TNB basate sull'identità che puoi utilizzare inIAM, consulta. Esempi di policy basate sull'identità per Telco Network Builder AWS

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. È necessario autenticarsi (accedere a AWS) come Utente root dell'account AWS, come IAM utente o assumendo un ruolo. IAM

È possibile accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAMIdentity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Quando accedi come identità federata, l'amministratore aveva precedentemente configurato la federazione delle identità utilizzando i ruoli. IAM Quando si accede AWS utilizzando la federazione, si assume indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando () per firmare crittograficamente le tue richieste utilizzando le tue credenziali. CLI Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta AWS Signature Version 4 per API le richieste nella Guida per l'IAMutente.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'AWS IAM Identity Center utente e Autenticazione a AWS più fattori IAM nella Guida per l'IAMutente.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per l'elenco completo delle attività che richiedono l'accesso come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'IAMutente.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per

informazioni su IAM Identity Center, vedi Cos'è IAM Identity Center? nella Guida AWS IAM Identity Center per l'utente.

IAM users and groups

Un IAMutente è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Laddove possibile, consigliamo di fare affidamento su credenziali temporanee anziché creare IAM utenti con credenziali a lungo termine come password e chiavi di accesso. Tuttavia, se hai casi d'uso specifici che richiedono credenziali a lungo termine con IAM gli utenti, ti consigliamo di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta Ruotare regolarmente le chiavi di accesso per i casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente. IAM

Un IAMgruppo è un'identità che specifica un insieme di utenti. IAM Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile assegnare un nome a un gruppo IAMAdminse concedere a tale gruppo le autorizzazioni per IAM amministrare le risorse.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Casi d'uso per IAM gli utenti nella Guida per l'IAMutente.

IAMruoli

Un IAMruolo è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un IAM utente, ma non è associato a una persona specifica. Per assumere temporaneamente un IAM ruolo in AWS Management Console, puoi passare da un utente a un IAM ruolo (console). È possibile assumere un ruolo chiamando un' AWS APIoperazione AWS CLI or o utilizzando un'operazione personalizzataURL. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere Metodi per assumere un ruolo nella Guida per l'IAMutente.

IAMi ruoli con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per informazioni

sui ruoli per la federazione, vedere Creazione di un ruolo per un provider di identità di terze parti nella Guida per l'IAMutente. Se utilizzi IAM Identity Center, configuri un set di autorizzazioni. Per controllare a cosa possono accedere le identità dopo l'autenticazione, IAM Identity Center correla il set di autorizzazioni a un ruolo in. IAM Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

- Autorizzazioni IAM utente temporanee: un IAM utente o un ruolo può assumere il IAM ruolo di assumere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso su più account: puoi utilizzare un IAM ruolo per consentire a qualcuno (un responsabile fidato) di un altro account di accedere alle risorse del tuo account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle risorse su più account IAM nella Guida per l'utente. IAM
- Accesso tra servizi: alcuni Servizi AWS utilizzano funzionalità in altri. Servizi AWS Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto (FAS): quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FASutilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. FASIe richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'internoIAM. Per ulteriori informazioni, vedere Creazione di un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAMutente.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un IAM ruolo per gestire le credenziali temporanee per le applicazioni in esecuzione su un'EC2istanza e che effettuano AWS CLI o richiedono AWS API. Ciò è preferibile alla memorizzazione delle chiavi di accesso all'interno dell'EC2istanza. Per assegnare un AWS ruolo a un'EC2istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull'EC2istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta Usare un IAM ruolo per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella Guida per l'IAMutente.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come JSON documenti. Per ulteriori informazioni sulla struttura e il contenuto dei documenti relativi alle JSON politiche, vedere Panoramica delle JSON politiche nella Guida per l'IAMutente.

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM politiche. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

IAMle politiche definiscono le autorizzazioni per un'azione indipendentemente dal metodo utilizzato per eseguire l'operazione. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale criterio può ottenere informazioni sul ruolo da AWS Management Console, da o da. AWS CLI AWS API

Policy basate su identità

I criteri basati sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per

informazioni su come creare una politica basata sull'identità, consulta Definire le IAM autorizzazioni personalizzate con le politiche gestite dal cliente nella Guida per l'utente. IAM

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli all'interno del tuo. Account AWS Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una politica gestita o una politica in linea, consulta Scegliere tra politiche gestite e politiche in linea nella Guida per l'IAMutente.

Policy basate su risorse

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di policy basate sulle risorse sono le policy di IAM role trust e le policy di Amazon S3 bucket. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le politiche AWS gestite IAM in una politica basata sulle risorse.

Elenchi di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLssono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioniACLs, consulta la panoramica di Access control list (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite di autorizzazioni è una funzionalità avanzata in cui si impostano le autorizzazioni massime che una politica basata sull'identità può concedere a un'entità

(utente o ruolo). IAM IAM È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità nella Guida per l'utente. IAM IAM

- Politiche di controllo del servizio (SCPs): SCPs sono JSON politiche che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. SCPLimita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna Utente root dell'account AWS. Per ulteriori informazioni su Organizations andSCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta le politiche di sessione nella Guida IAM per l'utente.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per informazioni su come AWS determinare se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle politiche nella Guida per l'IAMutente.

Come AWS TNB funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a AWS TNB, scopri con AWS TNB quali IAM funzionalità è possibile utilizzare.

IAMfunzionalità che puoi usare con AWS Telco Network Builder

IAMcaratteristica	AWS TNBsupporto
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
ACLs	No
ABAC(tag nelle politiche)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	No
Ruoli collegati al servizio	No

Per avere una panoramica generale del funzionamento AWS TNB e degli altri AWS servizi con la maggior parte delle IAM funzionalità, consulta AWS i servizi che funzionano con la maggior parte delle funzionalità IAM nella Guida per l'IAMutente.

Politiche basate sull'identità per AWS TNB

Supporta le policy basate su identità: sì

Le politiche basate sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta Definire le IAM autorizzazioni personalizzate con le politiche gestite dal cliente nella Guida per l'utente. IAM

Con le politiche IAM basate sull'identità, puoi specificare azioni e risorse consentite o negate, nonché le condizioni in base alle quali le azioni sono consentite o negate. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per ulteriori informazioni su tutti gli elementi che è possibile utilizzare in una JSON politica, vedere il riferimento agli elementi IAM JSON della politica nella Guida per l'IAMutente.

Esempi di policy basate sull'identità per AWS TNB

Per visualizzare esempi di politiche basate sull' AWS TNBidentità, vedere. Esempi di policy basate sull'identità per Telco Network Builder AWS

Politiche basate sulle risorse all'interno AWS TNB

Supporta le policy basate su risorse: no

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di policy basate sulle risorse sono le policy di IAM role trust e le policy di Amazon S3 bucket. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per abilitare l'accesso tra più account, puoi specificare un intero account o IAM entità in un altro account come principale in una politica basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un IAM amministratore dell'account fidato deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta la sezione Cross Account Resource Access IAM nella Guida IAM per l'utente.

Azioni politiche per AWS TNB

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare AWS JSON le policy per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'`Action`elemento di una JSON policy descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a una policy. Le azioni politiche in genere hanno lo stesso nome dell' AWS APIoperazione associata. Esistono alcune eccezioni, come le azioni basate solo sulle autorizzazioni che non hanno un'operazione corrispondente. API Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di AWS TNB azioni, vedere Azioni definite da AWS Telco Network Builder nel Service Authorization Reference.

Le azioni politiche in AWS TNB uso utilizzano il seguente prefisso prima dell'azione:

```
tnb
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

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

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `List`, includi la seguente azione:

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

Per visualizzare esempi di politiche AWS TNB basate sull'identità, vedere. Esempi di policy basate sull'identità per Telco Network Builder AWS

Risorse politiche per AWS TNB

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare AWS JSON le policy per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Resource` JSON policy specifica l'oggetto o gli oggetti a cui si applica l'azione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best

practice, specifica una risorsa utilizzando il relativo Amazon Resource Name (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di AWS TNB risorse e relativiARNs, consulta Resources defined by AWS Telco Network Builder nel Service Authorization Reference. Per sapere con quali azioni è possibile specificare il tipo ARN di ciascuna risorsa, vedere Azioni definite da AWS Telco Network Builder.

Per visualizzare esempi di politiche basate sull' AWS TNBidentità, vedere. Esempi di policy basate sull'identità per Telco Network Builder AWS

Chiavi relative alle condizioni delle politiche per AWS TNB

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica`OR`. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile concedere a un IAM utente l'autorizzazione ad accedere a una risorsa solo se è contrassegnata con il suo nome IAM utente. Per ulteriori informazioni, consulta gli elementi IAM della politica: variabili e tag nella Guida IAM per l'utente.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'IAMutente.

Per visualizzare un elenco di chiavi di AWS TNB condizione, vedere Chiavi di condizione per AWS Telco Network Builder nel Service Authorization Reference. Per sapere con quali azioni e risorse è possibile utilizzare una chiave di condizione, vedere Azioni definite da AWS Telco Network Builder.

Per visualizzare esempi di politiche basate sull' AWS TNBidentità, vedere. Esempi di policy basate sull'identità per Telco Network Builder AWS

ACLsin AWS TNB

SupportiACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLssono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

ABACcon AWS TNB

Supporti ABAC (tag nelle politiche): Sì

Il controllo degli accessi basato sugli attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. È possibile allegare tag a IAM entità (utenti o ruoli) e a molte AWS risorse. L'etichettatura di entità e risorse è il primo passo diABAC. Quindi si progettano ABAC politiche per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa a cui sta tentando di accedere.

ABACè utile in ambienti in rapida crescita e aiuta in situazioni in cui la gestione delle politiche diventa complicata.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni in meritoABAC, vedere Definizione delle autorizzazioni con ABAC autorizzazione nella Guida per l'IAMutente. Per visualizzare un tutorial con i passaggi per la

configurazioneABAC, consulta Use Attribute-based access control (ABAC) nella Guida per l'utente. IAM

Utilizzo di credenziali temporanee con AWS TNB

Supporta le credenziali temporanee: sì

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione Servizi AWS relativa alla funzionalità IAM nella Guida per l'IAMutente.

Si utilizzano credenziali temporanee se si accede AWS Management Console utilizzando qualsiasi metodo tranne il nome utente e la password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-on (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sul cambio di ruolo, consulta Passare da un utente a un IAM ruolo (console) nella Guida per l'IAMutente.

È possibile creare manualmente credenziali temporanee utilizzando AWS CLI o AWS API. È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, vedere Credenziali di sicurezza temporanee in. IAM

Autorizzazioni principali per più servizi per AWS TNB

Supporta sessioni di accesso diretto (FAS): Sì

Quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FASutilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. FASle richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta Forward access sessions.

Ruoli di servizio per AWS TNB

Supporta i ruoli di servizio: No

Un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'internoIAM. Per ulteriori informazioni, vedere Creazione di un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAMutente.

Ruoli collegati ai servizi per AWS TNB

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.

Esempi di policy basate sull'identità per Telco Network Builder AWS

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare AWS TNB risorse. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS API. Per concedere agli utenti il permesso di eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM policy. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

Per informazioni su come creare una politica IAM basata sull'identità utilizzando questi documenti di esempio, consulta Create JSON IAM policy (console) nella Guida per l'IAMutente.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da AWS TNB, incluso il formato di ARNs per ciascun tipo di risorsa, vedere Azioni, risorse e chiavi di condizione per AWS Telco Network Builder nel Service Authorization Reference.

Indice

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare AWS TNB risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta le politiche AWS gestite o le politiche AWS gestite per le funzioni lavorative nella Guida per l'IAMutente.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le IAM politiche, concedi solo le autorizzazioni necessarie per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo per applicare le autorizzazioni, consulta Politiche e autorizzazioni nella Guida IAM per l'utente. IAM IAM
- Utilizza le condizioni nelle IAM politiche per limitare ulteriormente l'accesso: puoi aggiungere una condizione alle tue politiche per limitare l'accesso ad azioni e risorse. Ad esempio, puoi scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzandoSSL. È inoltre possibile utilizzare condizioni per concedere l'accesso alle azioni di servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta Elementi IAM JSON della politica: Condizione nella Guida IAM per l'utente.
- Usa IAM Access Analyzer per convalidare IAM le tue policy e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio delle IAM policy () e alle best practice. JSON IAM IAMAccess Analyzer fornisce più di 100 controlli delle politiche e consigli pratici per aiutarti a creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle politiche con IAM Access Analyzer nella Guida per l'utente. IAM
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede l'utilizzo di IAM utenti o di un utente root Account AWS, attiva questa opzione MFA per una maggiore sicurezza. Per richiedere MFA quando vengono richiamate API le operazioni, aggiungi MFA delle condizioni alle tue politiche. Per ulteriori informazioni, consulta Secure API access with MFA nella Guida IAM per l'utente.

Per ulteriori informazioni sulle best practice inIAM, consulta la sezione Procedure consigliate in materia di sicurezza IAM nella Guida IAM per l'utente.

Utilizzo della AWS TNB console

Per accedere alla console AWS Telco Network Builder, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse del AWS TNB tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario concedere autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso il AWS CLI o il. AWS API Consenti invece l'accesso solo alle azioni che corrispondono all'APIoperazione che stanno cercando di eseguire.

Esempi di policy relative al ruolo di servizio

In qualità di amministratore, possiedi e gestisci le risorse AWS TNB create secondo quanto definito dall'ambiente e dai modelli di servizio. È necessario assegnare ruoli IAM di servizio al proprio account per consentire la creazione AWS TNB di risorse per la gestione del ciclo di vita della rete.

Un ruolo di IAM servizio consente di AWS TNB effettuare chiamate alle risorse per conto dell'utente per creare istanze e gestire le reti. Se specifichi un ruolo di servizio, AWS TNB utilizza le credenziali di quel ruolo.

Il ruolo di servizio e la relativa politica di autorizzazione vengono creati con il IAM servizio. Per ulteriori informazioni sulla creazione di un ruolo di servizio, vedere Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'IAMutente.

AWS TNBruolo di servizio

In qualità di membro del team della piattaforma, in qualità di amministratore puoi creare un ruolo di AWS TNB servizio e assegnarlo a AWS TNB. Questo ruolo consente di AWS TNB effettuare chiamate ad altri servizi come Amazon Elastic Kubernetes Service AWS CloudFormation e di fornire l'infrastruttura richiesta per la tua rete e fornire le funzioni di rete come definito nel tuo. NSD

Ti consigliamo di utilizzare la seguente politica di IAM ruolo e fiducia per il tuo AWS TNB ruolo di servizio. Quando definisci l'ambito delle autorizzazioni relative a questa politica, tieni presente che AWS TNB potrebbe fallire a causa degli errori di accesso negato relativi alle risorse contemplate dalla tua politica.

Il codice seguente mostra una politica relativa ai ruoli AWS TNB di servizio:

```
{
    "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:UpdateAddon",
                "eks:UpdateClusterVersion",
                "eks:UpdateNodegroupConfig",
                "eks:UpdateNodegroupVersion",
                "eks:DescribeUpdate",
                "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:UpdateStack",
                "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"
        }
    ]
}
```

Il codice seguente mostra la policy di attendibilità del AWS TNB servizio:

```
{
  "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 TNBruolo di servizio per il EKS cluster Amazon

Quando crei una EKS risorsa Amazon nel tuoNSD, fornisci l'`cluster_role`attributo per specificare quale ruolo verrà utilizzato per creare il tuo EKS cluster Amazon.

L'esempio seguente mostra un AWS CloudFormation modello che crea un ruolo AWS TNB di servizio per la policy del EKS cluster Amazon.

```
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"
```

Per ulteriori informazioni sui IAM ruoli che utilizzano il AWS CloudFormation modello, consulta le seguenti sezioni della Guida AWS CloudFormation per l'utente:

- AWS::IAM: :Ruolo
- Selezione di un modello di stack

AWS TNBruolo di servizio per il gruppo di EKS nodi Amazon

Quando crei un gruppo di EKS nodi Amazon nelle tue risorseNSD, fornisci l'`node_role`attributo per specificare quale ruolo verrà utilizzato per creare il tuo gruppo di EKS nodi Amazon.

L'esempio seguente mostra un AWS CloudFormation modello che crea un ruolo AWS TNB di servizio per la policy del gruppo di EKS nodi Amazon.

```
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/*"
```

Per ulteriori informazioni sui IAM ruoli che utilizzano il AWS CloudFormation modello, consulta le seguenti sezioni della Guida AWS CloudFormation per l'utente:

- AWS::IAM: :Ruolo
- Selezione di un modello di stack

AWS TNBruolo di servizio per Multus

Quando crei una EKS risorsa Amazon nel tuo NSD e desideri gestire Multus come parte del tuo modello di distribuzione, devi fornire l'`multus_role`attributo per specificare quale ruolo verrà utilizzato per la gestione di Multus.

L'esempio seguente mostra un AWS CloudFormation modello che crea un ruolo di AWS TNB servizio per una policy Multus.

```
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: "*"
```

Per ulteriori informazioni sui IAM ruoli che utilizzano il AWS CloudFormation modello, consulta le seguenti sezioni della Guida per l'AWS CloudFormation utente:

- AWS::IAM: :Ruolo
- Selezione di un modello di stack

AWS TNBruolo di servizio per una politica di aggancio del ciclo di vita

Quando il proprio NSD pacchetto di funzioni di rete utilizza un hook del ciclo di vita, è necessario un ruolo di servizio che consenta di creare un ambiente per l'esecuzione degli hook del ciclo di vita.

> **Note**
>
> La vostra policy relativa al ciclo di vita dovrebbe basarsi su ciò che il vostro Life-Cycle Hook sta cercando di fare.

L'esempio seguente mostra un AWS CloudFormation modello che crea un ruolo di AWS TNB servizio per una policy di hook del ciclo di vita.

```
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"
```

Per ulteriori informazioni sui IAM ruoli che utilizzano il AWS CloudFormation modello, consulta le seguenti sezioni della Guida per l'AWS CloudFormation utente:

- AWS::IAM: :Ruolo
- Selezione di un modello di stack

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra come è possibile creare una politica che consenta IAM agli utenti di visualizzare le politiche in linea e gestite allegate alla propria identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando o a livello di codice. AWS CLI AWS API

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Risoluzione dei problemi relativi all'identità e all'accesso di AWS Telco Network Builder

Utilizza le seguenti informazioni per diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AWS TNB eIAM.

Problemi

Non sono autorizzato a eseguire alcuna azione in AWS TNB

Se ricevi un errore che indica che non disponi dell'autorizzazione per eseguire un'operazione, le tue policy devono essere aggiornate in modo che ti sei consentito eseguire tale operazione.

L'errore di esempio seguente si verifica quando l'`mateojackson`IAMutente tenta di utilizzare la console per visualizzare i dettagli su una *`my-example-widget`* risorsa fittizia ma non dispone delle autorizzazioni fittizie`tnb:`*`GetWidget`*.

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

In questo caso, la policy deve essere aggiornata in modo che Mateo possa accedere alla risorsa *`my-example-widget`* mediante l'operazione `tnb:`*`GetWidget`*.

Se hai bisogno di assistenza, contatta l'amministratore. AWS L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di assegnare un ruolo a AWS TNB.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un IAM utente denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in. AWS TNB Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di assistenza, contatta AWS l'amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie AWS TNB risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS TNB supporta queste funzionalità, consulta. Come AWS TNB funziona con IAM
- Per informazioni su Account AWS come fornire l'accesso alle risorse di tua proprietà, consulta Fornire l'accesso a un IAM utente di un altro Account AWS utente di tua proprietà nella Guida per l'IAMutente.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a persone Account AWS di proprietà di terzi nella Guida per l'IAMutente.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso agli utenti autenticati esternamente (federazione delle identità) nella Guida per l'IAMutente.
- Per conoscere la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle risorse tra account nella Guida per l'utente. IAM IAM

Convalida della conformità per AWS TNB

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Ambito per programma di conformitàServizi AWS di conformità e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide introduttive su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla AWS sicurezza e la conformità.
- Architettura per la HIPAA sicurezza e la conformità su Amazon Web Services: questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee. HIPAA

  > **Note**
  >
  > Non tutte sono idonee. Servizi AWS HIPAA Per ulteriori informazioni, consulta la Guida ai servizi HIPAA idonei.

- AWS Risorse per AWS per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe riguardare il tuo settore e la tua località.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization ()). ISO
- Evaluating Resources with Rules nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e

verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.

- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, ad esempio PCI DSS soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente AWS l'utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Resilienza in AWS TNB

L'infrastruttura AWS globale è costruita attorno a zone Regioni AWS di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni sulle zone di disponibilità, vedere Global Regioni AWS Infrastructure.AWS

AWS TNBesegue il servizio di rete su EKS cluster in un cloud privato virtuale (VPC) nella AWS regione prescelta.

Sicurezza dell'infrastruttura in AWS TNB

In quanto servizio gestito, AWS Telco Network Builder è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi di AWS sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Si utilizzano API chiamate AWS pubblicate per accedere tramite AWS TNB la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). Richiediamo TLS 1.2 e consigliamo TLS 1.3.

- Suite di cifratura con Perfect Forward Secrecy (PFS) come (Ephemeral Diffie-Hellman) o DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale. IAM O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Ecco alcuni esempi di responsabilità condivise:

- AWS è responsabile della protezione dei componenti che supportano AWS TNB, tra cui:
  - Istanze di calcolo (note anche come worker)
  - Database interni
  - Comunicazioni di rete tra componenti interni
  - L'interfaccia di programmazione dell' AWS TNBapplicazione (API)
  - AWS Kit di sviluppo software () SDK
- L'utente è responsabile della protezione dell'accesso alle AWS risorse e ai componenti del carico di lavoro, tra cui (a titolo esemplificativo ma non esaustivo):
  - IAMutenti, gruppi, ruoli e politiche
  - bucket S3 per i quali memorizzi i tuoi dati AWS TNB
  - Altro Servizi AWS e risorse che utilizzi per supportare il servizio di rete tramite cui hai effettuato il provisioning AWS TNB
  - Il codice dell'applicazione
  - Connessioni tra il servizio di rete tramite il quale hai effettuato il provisioning AWS TNB e i relativi client

> **Important**
>
> Sei responsabile dell'implementazione di un piano di disaster recovery in grado di ripristinare efficacemente un servizio di rete tramite il quale hai effettuato il provisioning. AWS TNB

Modello di sicurezza della connettività di rete

I servizi di rete tramite i quali effettui il AWS TNB provisioning vengono eseguiti su istanze di calcolo all'interno di un cloud privato virtuale (VPC) situato in una AWS regione selezionata dall'utente. A VPC è una rete virtuale nel AWS cloud, che isola l'infrastruttura in base al carico di lavoro o all'entità organizzativa. La comunicazione tra le istanze di elaborazione interne VPCs rimane all'interno della AWS rete e non viaggia su Internet. Alcune comunicazioni di servizio interno attraversano Internet e sono crittografate. I servizi di rete forniti a tutti AWS TNB i clienti che operano nella stessa regione condividono le stesse informazioni. VPC I servizi di rete forniti a clienti diversi utilizzano istanze di elaborazione separate all'interno della stessa. AWS TNB VPC

Comunicazioni tra i client dei servizi di rete e il servizio di rete tramite Internet AWS TNB. AWS TNBnon gestisce queste connessioni. È tua responsabilità proteggere le connessioni dei tuoi client.

Le tue connessioni AWS TNB tramite AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDKs sono crittografate.

IMDSversione

AWS TNBsupporta istanze che utilizzano Instance Metadata Service versione 2 (IMDSv2), un metodo orientato alla sessione. IMDSv2include una sicurezza superiore a. IMDSV1 Per ulteriori informazioni, consulta Aggiungere una difesa approfondita contro firewall aperti, reverse proxy e SSRF vulnerabilità con miglioramenti ad Amazon Instance Metadata Service. EC2

Quando avvii l'istanza, devi usare. IMDSv2 Per ulteriori informazioniIMDSv2, consulta Use IMDSv2 in the Amazon EC2 User Guide.

Monitoraggio AWS TNB

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle AWS TNB altre AWS soluzioni. AWS consente AWS CloudTrail di osservare AWS TNB, segnalare quando qualcosa non va e intraprendere azioni automatiche laddove opportuno.

CloudTrail Utilizzato per acquisire informazioni dettagliate sulle chiamate effettuate a AWS APIs. È possibile archiviare queste chiamate come file di log in Amazon S3. È possibile utilizzare questi CloudTrail registri per determinare informazioni come la chiamata effettuata, l'indirizzo IP di origine da cui proviene la chiamata, chi ha effettuato la chiamata e quando è stata effettuata la chiamata.

I CloudTrail registri contengono informazioni sulle chiamate all'APIazione per. AWS TNB Contengono anche informazioni per gli inviti all'APIazione da parte di servizi come Amazon EC2 e AmazonEBS.

Registrazione delle chiamate AWS Telco Network Builder utilizzando API AWS CloudTrail

AWS Telco Network Builder è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o un. Servizio AWS CloudTrail acquisisce tutte le API chiamate come eventi. AWS TNB Le chiamate acquisite includono chiamate dalla AWS TNB console e chiamate in codice alle AWS TNB API operazioni. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata effettuata AWS TNB, l'indirizzo IP da cui è stata effettuata la richiesta, quando è stata effettuata e ulteriori dettagli.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente root o utente.
- Se la richiesta è stata effettuata per conto di un utente di IAM Identity Center.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro Servizio AWS.

CloudTrail è attivo nel tuo account Account AWS quando crei l'account e hai automaticamente accesso alla cronologia degli CloudTrail eventi. La cronologia CloudTrail degli eventi fornisce un

record visualizzabile, ricercabile, scaricabile e immutabile degli ultimi 90 giorni di eventi di gestione registrati in un. Regione AWSPer ulteriori informazioni, consulta Lavorare con la cronologia degli CloudTrail eventi nella Guida per l'utente.AWS CloudTrail Non sono CloudTrail previsti costi per la visualizzazione della cronologia degli eventi.

Per una registrazione continua degli eventi degli Account AWS ultimi 90 giorni, crea un trail o un data store di eventi CloudTrailLake.

CloudTrail sentieri

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Tutti i percorsi creati utilizzando il AWS Management Console sono multiregionali. È possibile creare un percorso a regione singola o multiregione utilizzando. AWS CLI La creazione di un percorso multiregionale è consigliata in quanto consente di registrare l'intera attività del proprio account Regioni AWS . Se crei un percorso a regione singola, puoi visualizzare solo gli eventi registrati nel percorso. Regione AWS Per ulteriori informazioni sui percorsi, consulta Creazione di un percorso per te Account AWS e Creazione di un percorso per un'organizzazione nella Guida per l'AWS CloudTrail utente.

Puoi inviare gratuitamente una copia dei tuoi eventi di gestione in corso al tuo bucket Amazon S3 CloudTrail creando un percorso, tuttavia ci sono costi di storage di Amazon S3. Per ulteriori informazioni sui CloudTrail prezzi, consulta la pagina Prezzi.AWS CloudTrail Per informazioni sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

CloudTrail Archivi di dati sugli eventi di Lake

CloudTrail Lake ti consente di eseguire query SQL basate sui tuoi eventi. CloudTrail Lake converte gli eventi esistenti in JSON formato basato su righe in formato Apache. ORC ORCè un formato di archiviazione colonnare ottimizzato per il recupero rapido dei dati. Gli eventi vengono aggregati in archivi di dati degli eventi, che sono raccolte di eventi immutabili basate sui criteri selezionati applicando i selettori di eventi avanzati. I selettori applicati a un archivio di dati degli eventi controllano quali eventi persistono e sono disponibili per l'esecuzione della query. Per ulteriori informazioni su CloudTrail Lake, consulta Working with AWS CloudTrail Lake nella Guida per l'utente.AWS CloudTrail

CloudTrail Gli archivi e le richieste di dati sugli eventi di Lake comportano dei costi. Quando crei un datastore di eventi, scegli l'opzione di prezzo da utilizzare per tale datastore. L'opzione di prezzo determina il costo per l'importazione e l'archiviazione degli eventi, nonché il periodo di conservazione predefinito e quello massimo per il datastore di eventi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione Prezzi.AWS CloudTrail

AWS TNBesempi di eventi

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'APIoperazione richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia stack ordinata delle API chiamate pubbliche, quindi gli eventi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra un CloudTrail evento che dimostra l'`CreateSolFunctionPackage`operazione.

```
{
    "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"
}
```

Per informazioni sul contenuto dei CloudTrail record, consultate il contenuto dei CloudTrail record nella Guida per l'AWS CloudTrail utente.

AWS TNBattività di distribuzione

Comprendi le attività di implementazione per monitorare efficacemente le implementazioni e agire più rapidamente.

La tabella seguente elenca le attività di AWS TNB distribuzione:

Nome dell'attività per le distribuzioni iniziate prima del 7 marzo 2024	Nome dell'attività per le distribuzioni iniziate a partire dal 7 marzo 2024	Task description (Descrizione attività)
AppInstallation	ClusterPluginInstall	Installa il plug-in Multus sul cluster AmazonEKS .
AppUpdate	nessuna modifica del nome	Aggiorna le funzioni di rete già installate in un'istanza di rete.
-	ClusterPluginUninstall	Disinstalla i plugin sul cluster Amazon. EKS
ClusterStorageClassConfiguration	nessuna modifica del nome	Configura la classe di storage (CSIdriver) su un EKS cluster Amazon.

Nome dell'attività per le distribuzioni iniziate prima del 7 marzo 2024	Nome dell'attività per le distribuzioni iniziate a partire dal 7 marzo 2024	Task description (Descrizione attività)
FunctionDeletion	nessuna modifica del nome	Elimina le funzioni di rete dalle AWS TNB risorse.
FunctionInstantiation	FunctionInstall	Implementa le funzioni di rete utilizzando. HELM
FunctionUninstallation	FunctionUninstall	Disinstalla la funzione di rete da un cluster AmazonEKS.
HookExecution	nessuna modifica del nome	Esegue gli hook del ciclo di vita come definito in. NSD
InfrastructureCancellation	nessuna modifica del nome	Annulla un servizio di rete.
InfrastructureInstantation	nessuna modifica del nome	AWS Fornisce risorse per conto dell'utente.
InfrastructureTermination	nessuna modifica del nome	Defornisce le AWS risorse richiamate tramite. AWS TNB
-	InfrastructureUpdate	Aggiorna le AWS risorse fornite per conto dell'utente.
InventoryDeregistration	nessuna modifica del nome	Annulla la registrazione delle risorse da. AWS AWS TNB
-	InventoryRegistration	Registra le risorse in. AWS AWS TNB
KubernetesClusterConfiguration	ClusterConfiguration	Configura il cluster Kubernetes e aggiunge ruoli aggiuntivi IAM ad Amazon EKS AuthMap come definito in. NSD

Nome dell'attività per le distribuzioni iniziate prima del 7 marzo 2024	Nome dell'attività per le distribuzioni iniziate a partire dal 7 marzo 2024	Task description (Descrizione attività)
NetworkServiceFinalization	nessuna modifica del nome	Finalizza il servizio di rete e fornisce un aggiornamento dello stato di esito positivo o negativo.
NetworkServiceInstantiation	nessuna modifica del nome	Inizializza il servizio di rete.
SelfManagedNodesConfiguration	nessuna modifica del nome	Avvia i nodi autogestiti con il piano di controllo Amazon EKS e Kubernetes.
-	ValidateNetworkServiceUpdate	Esegue le convalide prima di aggiornare un'istanza di rete.

Quote di servizio per AWS TNB

Le quote di servizio, note anche come limiti, sono il numero massimo di risorse o operazioni di servizio per l'account AWS . Per ulteriori informazioni, consulta Service Quotas di AWS nella Riferimenti generali di Amazon Web Services.

Di seguito sono riportate le quote di servizio per. AWS TNB

Nome	Predefinita	Adattabile	Descrizione
Operazioni di servizio di rete in corso e simultanee	Ogni regione supportata: 40	Sì	Il numero massimo di operazioni di servizio di rete in corso e simultanee in una regione.
Pacchetti di funzioni	Ogni Regione supportata: 200	Sì	Il numero massimo di pacchetti di funzioni in una regione.
Pacchetti di rete	Ogni regione supportata: 40	Sì	Il numero massimo di pacchetti di rete in una regione.
Istanze di servizi di rete	Ogni regione supportata: 800	Sì	Il numero massimo di istanze di servizi di rete in una regione.

Cronologia dei documenti per la guida per AWS TNB l'utente

La tabella seguente descrive le versioni della documentazione per. AWS TNB

Modifica	Descrizione	Data
Versione Kubernetes per cluster	AWS TNBora supporta la versione 1.30 di Kubernetes per creare cluster Amazon. EKS	19 agosto 2024

AWS TNBsupporta un'operazione aggiuntiva per gestire il ciclo di vita della rete.	È possibile aggiornare un'istanza di rete istanziata o precedentemente aggiornata con un nuovo pacchetto di rete e valori dei parametri. Vedere:
	• Operazioni del ciclo di vita
	• Aggiornare un'istanza di rete
	• AWS TNBesempio di ruolo di servizio:
	  • Aggiungi queste EKS azioni Amazon:`eks:UpdateAddon` ,`eks:UpdateClusterVersion` ,`eks:UpdateNodegroupConfig` ,`eks:UpdateNodegroupVersion` , `eks:DescribeUpdate`
	  • Aggiungi questa AWS CloudFormation azione: `cloudformation:UpdateStack`
		30 luglio 2024

	• Nuove attività di distribuzione: `InfrastructureUpdate Inventory Registration , ValidateNetworkServiceUpdate` • APIaggiornamenti: GetSolNetworkOperationListSolNetworkOperations, e UpdateSolNetworkInstance	
Nuova attività e nuovi nomi di attività per attività esistenti	È disponibile una nuova attività. A partire dal 7 marzo 2024, alcune attività esistenti hanno nuovi nomi per motivi di chiarezza.	7 maggio 2024
Versione Kubernetes per cluster	AWS TNBora supporta la versione 1.29 di Kubernetes per creare cluster Amazon. EKS	10 aprile 2024
Support per l'interfaccia di rete `security_groups`	È possibile collegare gruppi di sicurezza a AWS.Networking. ENInodo.	2 aprile 2024
Support per la crittografia dei volumi EBS root di Amazon	Puoi abilitare la EBS crittografia Amazon per il volume EBS root Amazon. Per abilitarla, aggiungi le proprietà in AWS.Compute. EKSManagedNodeo AWS .Compute. EKSSelfManagedNodenodo.	2 aprile 2024

Support per node `labels`	Puoi allegare etichette di nodi al tuo gruppo di nodi in AWS.Compute. EKSManagedNodeo AWS .Compute. EKSSelfManagedNodenodo.	19 marzo 2024
Support per l'interfaccia di rete `source_dest_check`	È possibile indicare se si desidera abilitare o disabilitare il controllo della sorgente/destinazione dell'interfaccia di rete tramite .Networking. AWS ENInodo.	25 gennaio 2024
Support per EC2 istanze Amazon con dati utente personalizzati	Puoi avviare EC2 istanze Amazon con dati utente personalizzati tramite AWS.Compute. UserData nodo.	16 gennaio 2024
Support for Security Group	AWS TNBconsente di importare la AWS risorsa Security Group.	8 gennaio 2024
Descrizione aggiornata di `network_interfaces`	Quando la `network_interfaces` proprietà è inclusa in AWS.Compute. EKSManagedNodeo AWS .Compute. EKSSelfManagedNodenode, AWS TNB ottiene l'autorizzazione relativa ENIs dalla `multus_role` proprietà, se disponibile, o dalla `node_role` proprietà.	18 dicembre 2023

Support per cluster privati	AWS TNBora supporta i cluster privati. Per indicare un cluster privato, imposta la `access` proprietà su`PRIVATE`.	11 dicembre 2023
Versione Kubernetes per cluster	AWS TNBora supporta la versione 1.28 di Kubernetes per creare cluster Amazon. EKS	11 dicembre 2023
AWS TNBsupporta il gruppo di collocamento	È stato aggiunto un gruppo di posizionamento per `AWS.Compute.EKSManagedNode` le definizioni dei `AWS.Compute.EKSSelfManagedNode` nodi.	11 dicembre 2023

AWS TNBaggiunge il supporto per IPv6	AWS TNBora supporta la creazione di istanze di rete con IPv6 infrastruttura. Controlla i nodi AWS.Networking. VPC, AWS.Networking.Subnet, .Networking.AWS InternetGateway, AWS.Reti. SecurityGroupIngressRule, AWS.Reti. SecurityGroupEgressRulee AWS.Compute. EKSper IPv6 le configurazioni. Abbiamo anche aggiunto i nodi AWS.Networking. NATGatewaye AWS.Networking.Route per la configurazione. NAT64 Abbiamo aggiornato il ruolo AWS TNB di servizio e il ruolo di AWS TNB servizio per il gruppo di EKS nodi Amazon per quanto riguarda IPv6 le autorizzazioni. Vedi esempi di policy relative ai ruoli di servizio.	16 novembre 2023
Autorizzazioni aggiunte alla policy del ruolo AWS TNB di servizio	Abbiamo aggiunto le autorizzazioni alla policy del ruolo AWS TNB di servizio per Amazon S3 AWS CloudFormation e per abilitare l'istanziazione dell'infrastruttura.	23 ottobre 2023

AWS TNBlanciato in più regioni	AWS TNBè ora disponibile nelle regioni Asia Pacifico (Seoul), Canada (Centrale), Europa (Spagna), Europa (Stoccolma) e Sud America (San Paolo).	27 settembre 2023
Tag per .Compute AWS. EKSSelfManagedNode	AWS TNBora supporta i tag per la definizione del `AWS.Compute.EKSSelfManagedNode` nodo.	22 agosto 2023
AWS TNBsupporta istanze che sfruttano IMDSv2	Quando avvii l'istanza, devi usare. IMDSv2	14 agosto 2023
Autorizzazioni aggiornate per MultusRoleInlinePolicy	MultusRoleInlinePolicy Ora include l'`ec2:DeleteNetworkInterface` autorizzazione.	7 agosto 2023
Versione Kubernetes per cluster	AWS TNBora supporta le versioni 1.27 di Kubernetes per creare cluster Amazon. EKS	25 luglio 2023
AWS.Calcolo. EKS. AuthRole	AWS TNBsupporti AuthRole che consentono di aggiungere IAM ruoli al EKS cluster Amazon `aws-auth ConfigMap` in modo che gli utenti possano accedere al EKS cluster Amazon utilizzando un IAM ruolo.	19 luglio 2023

AWS TNBsupporta gruppi di sicurezza.	È stato aggiunto il AWS file .Networking. SecurityGroup, AWS.Networking. SecurityGroupEgressRulee AWS.Networking. SecurityGroupIngressRuleal NSD modello.	18 luglio 2023
Versione Kubernetes per cluster	AWS TNBsupporta le versioni di Kubernetes da 1.22 a 1.26 per creare cluster Amazon. EKS AWS TNBnon supporta più le versioni 1.21 di Kubernetes.	11 maggio 2023
AWS.Calcolo. EKSSelfManagedNode	È possibile creare nodi di lavoro autogestiti su aree geografiche, AWS Local Zones e. AWS Outposts	29 marzo 2023
Versione iniziale	Questa è la prima versione della Guida per l' AWS TNButente.	21 febbraio 2023

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.