



Guida per l'utente

# AWS Costruttore di reti di telecomunicazioni



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

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

---

# Table of Contents

Che cos'è AWS TNB? .....	1
AWS Sei nuovo a? .....	2
A chi si AWS rivolge TNB? .....	2
AWS Caratteristiche di TNB .....	2
Accedere a AWS TNB .....	3
AWS Prezzi per TNB .....	4
Cosa c'è dopo .....	4
Come funziona AWS TNB .....	5
Architecture .....	5
Integrazione .....	6
Quote .....	7
AWS Concetti TNB .....	8
Ciclo di vita di una funzione di rete .....	8
Usa interfacce standardizzate .....	9
Pacchetto di funzioni .....	10
Pacchetto di rete .....	11
Descrittori dei servizi di rete .....	11
Gestione e operazioni .....	14
Configurazione di TNB AWS .....	16
Registrati per un Account AWS .....	16
Crea un utente con accesso amministrativo .....	17
Scegli una AWS regione .....	18
Annota l'endpoint del servizio .....	18
(Facoltativo) Installa AWS CLI .....	20
Configura i ruoli TNB AWS .....	20
Iniziare con AWS TNB .....	21
Prerequisiti .....	21
Crea un pacchetto di funzioni .....	22
Crea un pacchetto di rete .....	22
Crea e crea un'istanza di rete .....	23
Eliminazione .....	23
Pacchetti di funzioni .....	25
Crea .....	22
Vista .....	26

Scarica un pacchetto .....	27
Eliminazione di un pacchetto .....	28
AWS Pacchetti di rete TNB .....	29
Crea .....	22
Vista .....	30
Scarica .....	31
Elimina .....	32
Rete .....	33
Operazioni del ciclo di vita .....	33
Crea .....	23
Istanziare .....	35
Aggiorna un'istanza di funzione .....	36
Aggiorna un'istanza di rete .....	37
Considerazioni .....	37
Parametri che è possibile aggiornare .....	37
Aggiornamento di un'istanza di rete .....	70
Vista .....	71
Termina ed elimina .....	72
Operazioni di rete .....	74
Vista .....	74
Annulla .....	75
Riferimento TOSCA .....	76
Modello VNFD .....	76
Sintassi .....	76
Modello di topologia .....	76
AWS.VNF .....	77
AWS.Artifacts.Helm .....	78
Modello NSD .....	79
Sintassi .....	79
Utilizzo di parametri definiti .....	80
Importazione VNFD .....	80
Modello di topologia .....	81
AWS.NS .....	82
AWS.Compute.eks .....	83
AWS.Compute.eks. AuthRole .....	87
AWS.Calcola. EKSMangedNode .....	88

AWS.Calcola. EKSSelfManagedNode .....	96
AWS.Calcola. PlacementGroup .....	103
AWS.Calcola. UserData .....	105
AWS.Rete. SecurityGroup .....	106
AWS.Rete. SecurityGroupEgressRule .....	108
AWS.Rete. SecurityGroupIngressRule .....	111
AWS.Risorsa. Importazione .....	114
AWS.Networking.eni .....	115
AWS.HookExecution .....	117
AWS.Rete. InternetGateway .....	118
AWS.Rete. RouteTable .....	121
AWS.Networking.Subnet .....	122
AWS.Implementazione. VNFDeployment .....	125
AWS.Networking.vpc .....	127
AWS.Rete. NATGateway .....	128
AWS.Rete. Percorso .....	130
AWS.Negoziio. SSMPParameters .....	131
Nodi comuni .....	133
AWS.HookDefinition.Bash .....	133
Sicurezza .....	136
Protezione dei dati .....	137
Gestione dei dati .....	138
Crittografia dei dati a riposo .....	138
Crittografia dei dati in transito .....	138
Riservatezza del traffico inter-rette .....	138
Gestione dell'identità e degli accessi .....	138
Destinatari .....	139
Autenticazione con identità .....	139
Gestione dell'accesso tramite policy .....	140
Come funziona AWS TNB con IAM .....	142
Esempi di policy basate su identità .....	147
Risoluzione dei problemi .....	162
Convalida della conformità .....	164
Resilienza .....	165
Sicurezza dell'infrastruttura .....	165
Modello di sicurezza della connettività di rete .....	166

---

Versione IMDS .....	167
Monitoraggio .....	168
CloudTrail registri .....	168
AWS Esempi di eventi TNB .....	170
Attività di distribuzione .....	171
Quote .....	174
Cronologia dei documenti .....	175
.....	clxxxiv

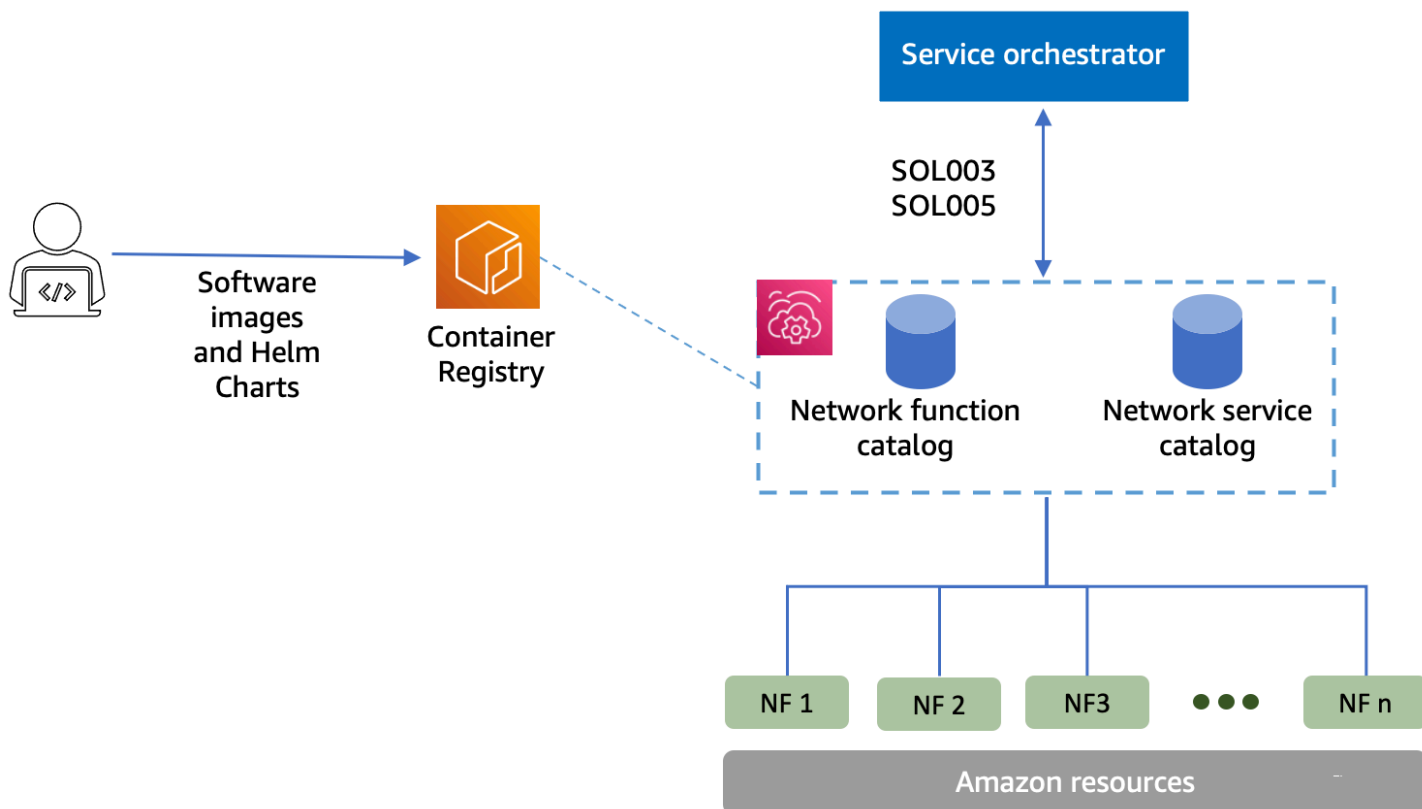
# Cos'è AWS Telco Network Builder?

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 partner ISV (Independent Software Vendor). AWS TNB si 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 delle reti 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?](#)
- [A chi si AWS rivolge TNB?](#)
- [AWS Caratteristiche di TNB](#)
- [Accedere a AWS TNB](#)
- [AWS Prezzi per TNB](#)
- [Cosa c'è dopo](#)

## AWS Sei nuovo a?

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

- [Introduzione a AWS](#)
- [Iniziare con AWS](#)

## A chi si AWS rivolge TNB?

AWS TNB è pensato per CSPs per sfruttare l'efficienza in termini di costi, l'agilità e l'elasticità delle Cloud AWS offerte senza scrivere e mantenere script e configurazioni personalizzati per progettare, implementare e gestire i servizi di rete. AWS TNB fornisce 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 definiti dal CSP e sulle funzioni di rete che il CSP desidera implementare.

## AWS Caratteristiche di TNB

Di seguito sono riportati alcuni dei motivi per cui 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 TNB si integra con i più diffusi orchestratori di servizi di terze parti conformi a ETSI.

## Bilance

Puoi configurare AWS TNB per scalare AWS le 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 TNB ti consente di ispezionare e monitorare le AWS risorse che supportano la tua rete su un'unica dashboard, come Amazon VPC EC2, Amazon e Amazon EKS.

### Supporta modelli di servizio

AWS TNB consente 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 TNB apporterà 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. È inoltre possibile gestire le applicazioni RAN, Core, IMS e di rete allo stesso modo.

## Accedere a AWS TNB

Puoi creare, accedere e gestire le tue risorse AWS TNB utilizzando una delle seguenti interfacce:

- AWS Console TNB: fornisce un'interfaccia web per la gestione della rete.
- AWS API TNB: fornisce un' RESTful API per eseguire azioni AWS TNB. Per ulteriori informazioni, consulta [AWS TNB API Reference](#)

- **AWS Command Line Interface (AWS CLI)** — Fornisce comandi per un'ampia gamma di AWS servizi, incluso AWS TNB. È supportato su Windows, macOS e Linux. Per ulteriori informazioni, consulta la [Guida per l'utente 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, consulta [AWS SDKs](#).

## AWS Prezzi per TNB

AWS TNB aiuta ad CSPs automatizzare l'implementazione e la gestione delle proprie reti di telecomunicazioni su. AWS Quando utilizzi TNB, paghi per le seguenti due dimensioni: AWS

- Per elemento funzionale di rete gestito (MNF) ore.
- Per numero di richieste API.

Inoltre, l'utilizzo di altri AWS servizi in collaborazione con TNB comporta costi aggiuntivi. AWS [Per ulteriori informazioni, consulta TNB Pricing.AWS](#)

Per visualizzare la tua fattura, passa al Pannello di controllo di gestione fatturazione e costi nella [console Gestione dei costi e fatturazione AWS](#). 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 a usare AWS TNB, consulta i seguenti argomenti:

- [Configurazione di AWS TNB](#)— Completare i passaggi preliminari.
- [Guida introduttiva a AWS TNB](#)— Implementa la tua prima funzione di rete, come Centralized Unit (CU), Access and Mobility Management Function (AMF), User Plane Function (UPF) o un 5G Core completo.

# Come funziona AWS TNB

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

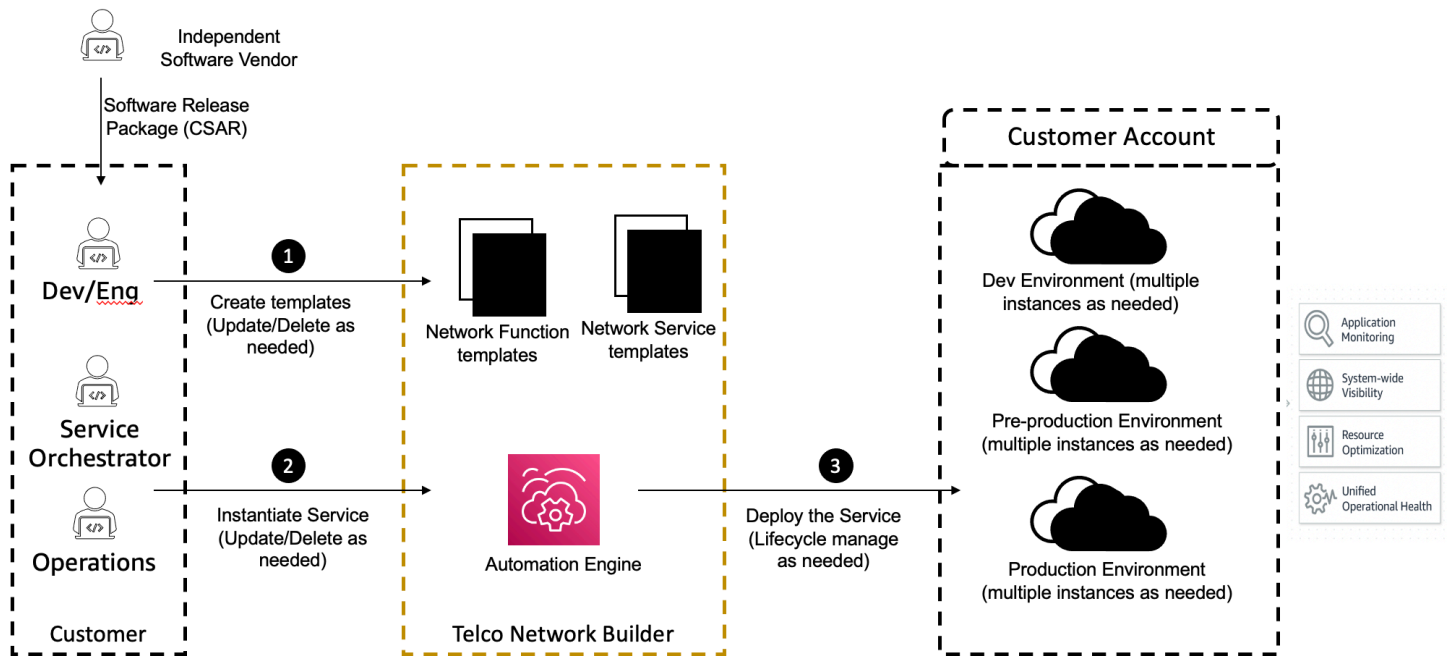
AWS TNB consente di importare pacchetti di funzioni di rete e descrittori di servizi di rete (NSDs) e fornisce il motore di automazione per gestire le reti. Puoi utilizzare il tuo end-to-end orchestratore e integrarlo con AWS TNB oppure utilizzare AWS SDKs TNB APIs per creare il tuo flusso di automazione. Per ulteriori informazioni, consulta [AWS Architettura TNB](#).

## Argomenti

- [AWS Architettura TNB](#)
- [Integrazione con Servizi AWS](#)
- [AWS quote di risorse TNB](#)

## AWS Architettura TNB

AWS TNB offre la possibilità di eseguire operazioni di gestione del ciclo di vita tramite l' Console di gestione AWS API REST di AWS TNB e AWS CLI SDKs. Ciò consente alle diverse personalità del CSP, come i membri dei team di ingegneria, operazioni e sistema programmatico, di trarre vantaggio da TNB. AWS È possibile creare e caricare un pacchetto di funzioni di rete come file Cloud Service Archive (CSAR). Il file CSAR contiene grafici Helm, immagini software e un Network Function Descriptor (NFD). È possibile utilizzare 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 TNB ti 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 TNB si integra con quanto segue Servizi AWS come specifico per le telecomunicazioni per creare un servizio di rete completamente operativo: APIs

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

## AWS quote di risorse TNB

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 AWS È 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, scegli e seleziona Servizi AWSTNB.AWS

Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas.

Hai Account AWS le seguenti quote relative a TNB. AWS

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

# AWS Concetti TNB

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

## Indice

- [Ciclo di vita di una funzione di rete](#)
- [Usa interfacce standardizzate](#)
- [Pacchetto di funzioni](#)
- [Pacchetto di rete](#)
- [Gestione e operazioni per TNB AWS](#)

## Ciclo di vita di una funzione di rete

AWS TNB ti 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 pacchetti CSAR da distribuire o aggiornare.
4. Usa AWS TNB per caricare il pacchetto CSAR che definisce la tua funzione di rete (ad esempio, CU AMF e UPF) e esegui l'integrazione con una pipeline di integrazione e distribuzione continua (CI/CD) che può aiutarti a creare nuove versioni del tuo pacchetto CSAR man mano che 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. Utilizza le informazioni per creare il tuo 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.

### Istanziamento

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

2. Crea un'istanza (o fornisci) la funzione di rete come definita nel relativo 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 `version` parametro Amazon EKS nel pacchetto di rete.

## Usa interfacce standardizzate

AWS TNB si integra con gli orchestratori di servizi conformi allo European Telecommunications Standards Institute (ETSI), consentendovi di semplificare l'implementazione dei servizi di rete. Gli orchestratori di servizi possono utilizzare AWS TNB, SDKs la CLI o il APIs per avviare operazioni, come l'istanziamento o l'aggiornamento di una funzione di rete a una nuova versione.

AWS TNB supporta le seguenti specifiche.

Specifiche	Versione	Description
ETSI SOL001	<a href="#">versione 3.6.1</a>	Definisce gli standard per consentire i descrittori di funzioni di rete basati su Tosca.
ETSI SOL002	<a href="#">versione 3.6.1</a>	Definisce i modelli relativi alla gestione delle funzioni di rete.
ETSI SOL003	<a href="#">versione 3.6.1</a>	Definisce gli standard per la gestione del ciclo di vita delle funzioni di rete.
ETSI SOL004	<a href="#">versione 3.6.1</a>	Definisce gli standard CSAR per i pacchetti di funzioni di rete.
ETSI SOL005	<a href="#">versione 3.6.1</a>	Definisce gli standard per i pacchetti di servizi di rete e la gestione del ciclo di vita dei servizi di rete.

Specifiche	Versione	Description
ETSI SOL007	<a href="#">versione 3.5.1</a>	Definisce gli standard per consentire i descrittori di servizi di rete basati su Tosca.

## Pacchetto di funzioni

Con AWS TNB, è possibile archiviare pacchetti di funzioni conformi a ETSI SOL001/SOL004 in un catalogo di funzioni. Quindi, puoi caricare pacchetti Cloud Service Archive (CSAR) che contengono artefatti che descrivono la funzione della tua rete virtuale.

- **Descrittore di funzioni di rete virtuale:** definisce i metadati per l'onboarding dei pacchetti e la gestione delle funzioni di rete virtuale. È necessario assegnare un nome a questo file. `vnfd.yaml`
- **Immagini del software:** fa riferimento alla funzione di rete virtuale Container Images. Amazon Elastic Container Registry (Amazon ECR) può fungere da archivio di immagini delle funzioni di rete virtuale.
- **File aggiuntivi:** da utilizzare per gestire la funzione di rete virtuale, ad esempio script e grafici Helm.

Il CSAR è un pacchetto definito dallo standard OASIS TOSCA e include un descrittore di rete/servizio conforme alla specifica YAML di OASIS TOSCA. Per informazioni sulla specifica YAML richiesta, vedere [Riferimento TOSCA per AWS TNB](#)

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

```
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"
```

## Pacchetto di rete

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

Il pacchetto di rete contiene i seguenti file:

- Un file descrittore di rete (`nsd.yaml`) in formato TOSCA come descritto da ETSI SOL007.

Il `nsd.yaml` file contiene riferimenti ai [pacchetti di funzioni caricati con il relativo descrittore](#). IDs

- Eventuali script di dati utente.
- Eventuali script Lifecycle Hook.
- Eventuali file di `values.yaml` configurazione dei plugin.

AWS TNB supporta gli standard ETSI per la modellazione di risorse, come rete, servizi e funzioni, nel linguaggio TOSCA. AWS TNB ne rende più efficiente l'utilizzo Servizi AWS modellandoli in modo che il vostro orchestratore di servizi conforme all'ETSI possa comprendere.

## AWS Descrittori di servizi di rete per TNB

Un descrittore di servizi di rete (NSD) è un `.yaml` file contenuto in un pacchetto di rete che utilizza lo standard TOSCA 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 l'NSD e configurare le risorse sottostanti e le operazioni del ciclo di vita della rete, è necessario comprendere lo schema NSD TOSCA supportato da TNB. AWS

Il file NSD è suddiviso nelle seguenti parti:

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

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

2. VNFD — L'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 pacchetto CSAR della funzione di rete.
- Un nome univoco per `namespace`. Il nome deve essere associato a un ID univoco per facilitarne il riferimento all'interno del file YAML NSD, 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 cluster Amazon EKS. 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
```

## Esempio NSD

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

```
tosca_definitions_version: tnb_simple_yaml_1_0

SampleNFEKS:
  type: tosa.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: toscanodes.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 TNB AWS

Con AWS TNB, è possibile gestire la rete utilizzando operazioni di gestione standardizzate in conformità con ETSI SOL003 e SOL005. È possibile utilizzare il AWS TNB APIs per eseguire operazioni del ciclo di vita come:

- Istanziamento 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 tuo NSDs.

- **Recupero di informazioni sulle funzioni di rete distribuite.**

# Configurazione di AWS TNB

Configura AWS TNB completando le attività descritte in questo argomento.

## Processi

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

## 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 o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene 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 [Console di gestione AWS](#) come 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 [Accedere come utente root](#) nella Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta [Abilitare un dispositivo MFA virtuale per l'utente Account AWS root \(console\)](#) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita il Centro identità IAM.

Per istruzioni, consulta [Abilitazione del AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center .

2. Nel Centro identità IAM, assegna 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 l'impostazione predefinita IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere come utente del Centro identità IAM, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente del Centro identità IAM.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta [AWS Accedere al portale di accesso](#) nella Guida per l'Accedi ad AWS utente.

## Assegnazione dell'accesso ad altri utenti

1. Nel Centro identità IAM, crea un set di autorizzazioni conforme alla best practice per l'applicazione di autorizzazioni con il privilegio minimo.

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 AWS regione

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 endpoint [AWS TNB](#) nel. 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 endpoint FIPS in regioni selezionate. Per ulteriori informazioni, consulta [Endpoint del servizio AWS](#).

Nome della regione	Regione	Endpoint	Protocollo
Stati Uniti orientali (Virginia settentrionale)	us-east-1	tnb.us-east-1.amazonaws.com	HTTPS
Stati Uniti occidentali (Oregon)	us-west-2	tnb.us-west-2.amazonaws.com	HTTPS

Nome della regione	Regione	Endpoint	Protocollo
Asia Pacifico (Seoul)	ap-northeast-2	tnb.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacifico (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 a 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 Command Reference.AWS CLI

## Configura i ruoli TNB AWS

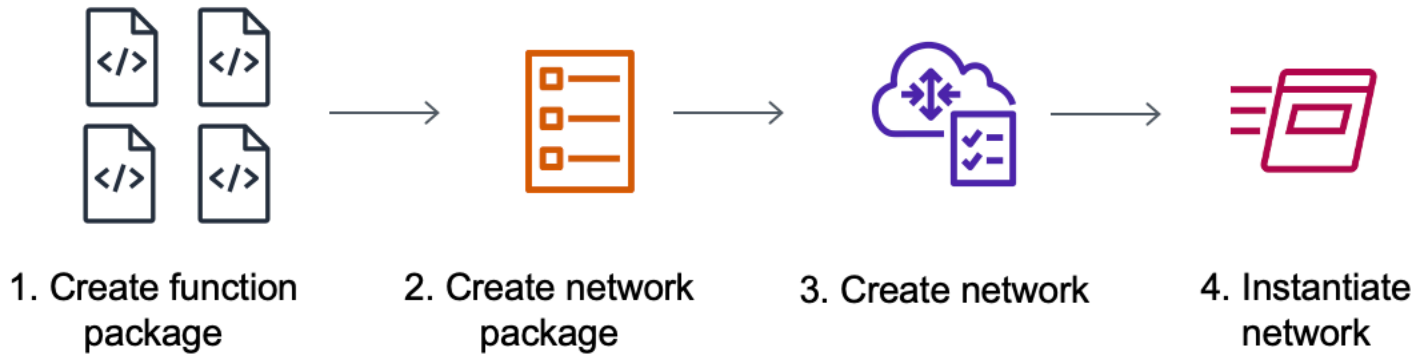
È necessario creare un ruolo di servizio IAM per gestire diverse parti della soluzione AWS TNB. AWS I ruoli di servizio TNB possono effettuare chiamate API ad altri AWS servizi, come AWS CloudFormation AWS CodeBuild, e a vari servizi di elaborazione e archiviazione, per tuo conto, per creare istanze e gestire le risorse per la tua implementazione.

Per ulteriori informazioni sul ruolo del servizio AWS TNB, vedere. [Gestione delle identità e degli accessi per TNB AWS](#)

# Guida introduttiva a AWS TNB

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

Il diagramma seguente illustra il processo di implementazione:



## Attività

- [Prerequisiti](#)
- [Crea 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 ruoli IAM.
- Un [pacchetto Network Function \(NF\)](#) conforme a ETSI SOL001/SOL004.
- Modelli NSD ([Network Service Descriptor](#)) conformi a ETSI SOL007.

È possibile utilizzare un pacchetto di funzioni di esempio o un pacchetto di rete dal sito [Sample packages](#) for TNB. AWS GitHub

## Crea un pacchetto di funzioni

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

Per creare un pacchetto di funzioni

1. Apri la console AWS TNB 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, scegli Scegli file e carica ogni pacchetto CSAR 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 ogni 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 implementata. È necessario creare un'istanza di rete e crearne un'istanza. Quando si crea un'istanza di rete, AWS TNB fornisce l'AWS infrastruttura necessaria, implementa funzioni di rete containerizzate e 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 stato `Not instantiated`.

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 pacchetti NSD e VNFD non cambiano.
8. Scegli Instantiated 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 dello stato di 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 standard ETSI) e un descrittore di pacchetti di funzioni che utilizza lo standard TOSCA per descrivere come le funzioni di rete dovrebbero funzionare sulla rete.

## Processi

- [Crea un pacchetto di funzioni in TNB AWS](#)
- [Visualizza un pacchetto di funzioni in AWS TNB](#)
- [Scarica un pacchetto di funzioni da AWS TNB](#)
- [Eliminare un pacchetto di funzioni da AWS TNB](#)

## Crea un pacchetto di funzioni in TNB AWS

Scopri come creare un pacchetto di funzioni nel catalogo delle funzioni di rete AWS TNB. 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, puoi creare un pacchetto di rete.

## Console

Per creare un pacchetto di funzioni utilizzando la console

1. Apri la console AWS TNB 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 pacchetto CSAR 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-package](#) comando 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. 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. Apri la console AWS TNB 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-packages](#) comando per elencare i pacchetti di funzioni.

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

- Utilizzate il [get-sol-function-package](#) comando 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
```

## Scarica un pacchetto di funzioni da AWS TNB

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

### Console

Per scaricare un pacchetto di funzioni utilizzando la console

- Apri la console AWS TNB all'indirizzo <https://console.aws.amazon.com/tnb/>.
- Nel riquadro di navigazione sul lato sinistro della console, scegli Pacchetti di funzioni.
- Usa la casella di ricerca per trovare il pacchetto di funzioni
- Scegli il pacchetto di funzioni
- 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 rete AWS TNB. Per eliminare un pacchetto di funzioni, il pacchetto deve essere disabilitato.

## Console

Per eliminare un pacchetto di funzioni utilizzando la console

1. Apri la console AWS TNB 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-package](#) comando 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-package](#) comando 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). Definisce i pacchetti di funzioni che desideri distribuire e l' AWS infrastruttura su cui desideri distribuirli.

Il pacchetto di rete contiene i seguenti file:

- Un file descrittore di rete (nsd.yaml) in formato TOSCA come descritto da ETSI. SOL007

Il nsd.yaml file contiene riferimenti ai [pacchetti di funzioni](#) caricati con il relativo descrittore. IDs

- Eventuali script di dati utente.
- Eventuali script Lifecycle Hook.
- Eventuali file di values.yaml configurazione dei plugin.

## Processi

- [Crea un pacchetto di rete in TNB AWS](#)
- [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 TNB AWS

Un pacchetto di rete è costituito da un file NSD (Network Service Descriptor) (obbligatorio) e da qualsiasi file aggiuntivo (opzionale), come gli script specifici per le vostre esigenze. Ad esempio, se nel pacchetto di rete sono presenti più pacchetti di funzioni, è possibile utilizzare l'NSD per definire quali funzioni di rete devono essere eseguite in determinate VPCs sottoreti o cluster Amazon EKS.

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. Apri la console AWS TNB 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 ogni 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-package](#) comando 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. 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. Apri la console AWS TNB 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 AWS CLI

1. Utilizzate il [list-sol-network-packages](#) comando per elencare i pacchetti di rete.

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

2. Utilizzate il [get-sol-network-package](#) comando 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 rete AWS TNB.

### Console

Per scaricare un pacchetto di rete utilizzando la console

1. Apri la console AWS TNB 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}$ \  
--region us-west-2
```

```
--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 rete AWS TNB. Per eliminare un pacchetto di rete, il pacchetto deve essere disabilitato.

### Console

Per eliminare un pacchetto di rete utilizzando la console

1. Apri la console AWS TNB 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 AWS CLI

1. Utilizzare il [update-sol-network-package](#) comando 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-package](#) comando 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 implementata.

## Processi

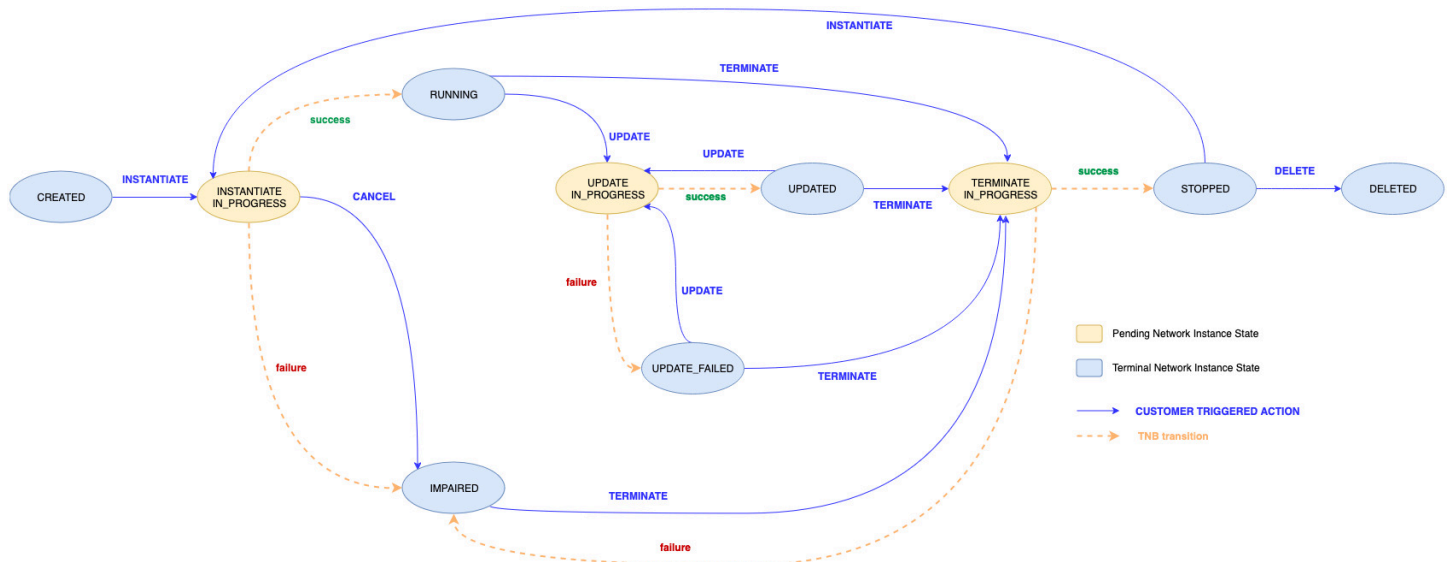
- [Operazioni del ciclo di vita di un'istanza di rete](#)
- [Crea un'istanza di rete utilizzando TNB AWS](#)
- [Crea un'istanza di rete utilizzando TNB AWS](#)
- [Aggiorna un'istanza di funzione in AWS TNB](#)
- [Aggiorna un'istanza di rete in AWS TNB](#)
- [Visualizza un'istanza di rete in AWS TNB](#)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

## Operazioni del ciclo di vita di un'istanza di rete

AWS TNB consente di gestire facilmente la rete utilizzando operazioni di gestione standardizzate in linea con ETSI e. SOL003 SOL005 È possibile eseguire le seguenti operazioni relative al 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 TNB AWS

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 console AWS TNB 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 AWS CLI

- Utilizzare il [create-sol-network-instance](#) comando 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 TNB AWS

Dopo aver creato un'istanza di rete, è necessario crearne un'istanza. Quando si crea un'istanza di rete, AWS TNB fornisce l'AWS infrastruttura necessaria, implementa funzioni di rete containerizzate e 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 console AWS TNB 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 pacchetti NSD e VNFD 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 inCompleted, viene creata un'istanza dell'istanza di rete.

## AWS CLI

Per creare un'istanza di rete utilizzando AWS CLI

1. Utilizzare il [instantiate-sol-network-instance](#) comando 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. Apri la console AWS TNB 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. Scegliere Aggiorna.
6. Inserisci le sostituzioni di aggiornamento.
7. Scegliere Aggiorna.

## AWS CLI

Usa la CLI per aggiornare un'istanza di funzione

Utilizzate il [update-sol-network-instance](#) comando 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, l'UpdateSolNetworkServiceAPI utilizza il nuovo pacchetto di rete e i valori dei parametri per aggiornare la topologia dell'istanza di rete.
- AWS TNB verifica che il numero di parametri NSD e VNFD nell'istanza di rete non superi 200. Questo limite viene applicato per proteggere i malintenzionati dal trasferimento di 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	Description	Esempio: Prima	Esempio: Secondo
Versione del cluster Amazon EKS	Puoi aggiornare il valore per il <code>version</code> parametro del piano di controllo del cluster Amazon EKS alla versione secondari a successiva. Non è possibile effettuare il downgrade della versione.	<pre>EKSCluster:   type: toska.nod es.AWS.Compute.EKS properties:   version: "1.28"</pre>	<pre>EKSCLUSTER:   type:   toska.nod es.A mput</pre>

Parametro	Description	Esempio: Prima

Esem  
dopo

pro  
s:

ver  
"1.

Parametro	Description	Esempio: Prima	Esem dopo
<p>Nodi di lavoro Amazon EKS</p>	<p>Puoi aggiornare il valore del <code>EKSManagedNode</code> <code>kubernetes_version</code> parametro per aggiornare il tuo gruppo di nodi a una versione più recente di Amazon EKS oppure puoi aggiornare il <code>ami_id</code> parametro per aggiornare il tuo gruppo di nodi all'ultima AMI ottimizzata per EKS.</p> <p>Puoi aggiornare l'ID AMI per <code>EKSSelfManagedNode</code>. La versione Amazon EKS dell'AMI deve essere uguale o inferiore a due versioni rispetto alla versione del cluster Amazon EKS. Ad esempio, se la versione del cluster Amazon EKS è 1.31, la versione dell'AMI Amazon EKS deve essere 1.31, 1.30 o 1.29.</p>	<pre>EKSManagedNodeGroup01:   ...   properties:     kubernetes_version: " 1.28"     EKSSelfManagedNodeGroup01:       compute:         compute:           properties:             ami_id:               "ami-1231230LD "</pre>	<p>EKSM dNoc p01: ... pro s:  kub s_ve : "1.  EKS nage 01:  com</p>

Parametro	Description	Esempio: Prima

Esem  
dopo

com

pro  
s:

am  
"am  
3NEW

Parametro	Description	Esempio: Prima	Esem dopo
Gruppi di nodi Amazon EKS	<p>Puoi aggiungere o rimuovere gruppi di nodi in base alle tue esigenze di elaborazione.</p> <p>Quando elimini gruppi di nodi esistenti e ne aggiungi e di nuovi, assicurati che i nuovi gruppi di nodi siano diversi da quelli eliminati, altrimenti l'operazione verrà considerata come una modifica del gruppo di nodi anziché un'eliminazione e un'aggiunta. Nota che per i gruppi di nodi esistenti, è possibile aggiornare solo un set limitato di parametri. Scorri questa tabella per vedere quali parametri puoi aggiornare.</p>	<pre>Free5GCEKSNODE01:   type: tosca.nod es.AWS.Compute.EKS ManagedNode ...   scaling:     properties:       desired_size: 1       min_size: 1       max_size: 1 ... Free5GCEKSNODE02 : # Deleted Nodegroup   type: tosca.nod es.AWS.Compute.EKS ManagedNode ...   scaling:     properties:       desired_size: 1       min_size: 1       max_size: 1 ... Free5GCEKSNODE03 : # Deleted Nodegroup   type: tosca.nod es.AWS.Compute.EKS SelfManagedNode ...   scaling:     properties:       desired_size: 1       min_size: 1       max_size: 1 ...</pre>	<p>Free SNODE typ tos es.A mput Mana de ... sca pro s: des ize: 1</p>

Parametro	Description	Esempio: Prima

Esem  
dopo

min  
1

max  
1

...

*Free*  
*SNo*

#

New

No

typ

tos

es.A

mput

Self

edNo

...

sca

pro

s:

Parametro	Description	Esempio: Prima

Esem  
dopo

des  
ize:  
1

mir  
1

max  
1

...  
*Free*  
*SNo*  
#  
New  
Noc

typ  
tos  
es.A  
mput  
Mana  
de

Parametro	Description	Esempio: Prima	Esem dopo
			... sca pro s: des ize: 1 mir 1 max 1

Parametro	Description	Esempio: Prima	Esem dopo ...
			...

Parametro	Description	Esempio: Prima	Esem dopo
Proprietà di ridimensionamento	È possibile aggiornare le proprietà di scala dei nodi EKSMANAGEDNode e EKSSelfManagedNode TOSCA.	<pre> EKSNodeGroup01:   ...   scaling:     properties:       desired_size: 1       min_size: 1       max_size: 1 </pre>	EKSNodeGroup01: ... scaling: ... desired_size: 1 min_size: 1 max_size: 1



Parametro	Description	Esempio: Prima
<p>Proprietà del plug-in Amazon EBS CSI</p>	<p>Puoi abilitare o disabilitare il plug-in Amazon EBS CSI sui tuoi cluster Amazon EKS. Puoi anche modificare la versione del plugin.</p>	<pre> EKSCluster:   capabilities:     ...     ebs_csi:       properties:         enabled: <i>false</i>                     </pre>

Esem  
dopo

EKS  
r:  
cap  
ies:  
...  
ebs  
pro  
s:  
ena  
ver  
"v1  
e  
ksbu  
"

Parametro	Description	Esempio: Prima	Esem dopo
Dimensione del volume root	È possibile aggiungere, rimuovere o aggiornare la proprietà della dimensione del volume principale dei nodi EKSManged Node e EKSSelf ManagedNode TOSCA.	<pre>Free5GCEKSN01:   ...   capabilities:     compute:       properties:         root_volu me_size: 50</pre>	Free SNoc ... cap ies: com pro s:

Parametro	Description	Esempio: Prima	Esem dopo
			roo me_s

Parametro	Description	Esempio: Prima	Esempio: dopo
<p>VNF</p>	<p>È possibile fare riferimento a essi VNFs nell'NSD e distribuirli nel cluster creato in NSD utilizzando il nodo TOSCA. VNFDeployment Come parte dell'aggiornamento, potrai aggiungere, aggiornare ed eliminare VNFs dalla rete.</p>	<pre> vnfds:   - descriptor_id:     "43c012fa-2616-41a8-     a833-0dfd4c5a049e "     namespace: " vnf1"   - descriptor_id:     "64222f98-ecd6-4871-     bf94-7354b53f3ee5 "     namespace:     "vnf2" // Deleted VNF ... SampleVNF1HelmDeploy:   type: toska.nod es.AWS.Deployment. VNFDeployment   requirements:     cluster:       EKSCluster       vnfs:         - vnf1.Samp leVNF1         - vnf2.Samp leVNF2         </pre>	<pre> vnfd - des r_id "55 79e5 - be53 2ad0 " nam : "vr Upd VNF - des r_id "b7 839c -916 a166 " nam : "vr Add VNF .... Sa mple         </pre>

Parametro	Description	Esempio: Prima

Esem  
dopo

elmd  
:

typ  
tos  
es.A  
ploy  
VNFD  
ment

rec  
nts:

clu  
EKS  
r

vnf

Parametro	Description	Esempio: Prima

Esem  
dopo

- v  
LeVM

- v  
LeVM

Parametro	Description	Esempio: Prima	Esem dopo
Hook	<p>Per eseguire le operazioni del ciclo di vita prima e dopo la creazione di una funzione di rete, aggiungi gli <code>post_create</code> hook <code>pre_create</code> and al <code>VNFDeployment</code> nodo.</p> <p>In questo esempio, l'<code>PreCreateHook</code> hook verrà eseguito prima dell'<code>vnf3.SampleVNF3</code> istanziazione e l'<code>PostCreateHook</code> hook verrà eseguito dopo <code>vnf3.SampleVNF3</code> l'istanza .</p>	<pre> vnfds:   - descriptor_id:     "43c012fa-2616-41a8-     a833-0dfd4c5a049e "     namespace: " vnf1"   - descriptor_id:     "64222f98-ecd6-4871-     bf94-7354b53f3ee5 "     namespace: " vnf2"   ... SampleVNF1HelmDeploy:   type: toscanod es.AWS.Deployment. VNFDeployment   requirements:     cluster: EKSCluster   vnfs:     - vnf1.SampleVNF1     - vnf2.Samp leVNF2 // Removed during update </pre>	<pre> vnfd - des r_id "43 2616 - a833 d4c5 " nam : "vr - des r_id "b7 839c -916 a166 " nam : "vr .... S ampL Helm y: typ tos </pre>

Parametro	Description	Esempio: Prima

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

Parametro	Description	Esempio: Prima

Esem  
dopo

- v  
*LeVM*  
New  
VNF  
as  
the  
nam

,  
vnt  
was  
not  
pre  
y  
pre

int  
s:

Ho

pos  
te:  
*eHoo*

pre  
e:  
*Hook*

Parametro	Description	Esempio: Prima	Esem dopo
Hook	<p>Per eseguire le operazioni del ciclo di vita prima e dopo l'aggiornamento di una funzione di rete, è possibile aggiungere l'<code>pre_update</code> hook e l'<code>post_update</code> hook al nodo. <code>VNFDeployment</code></p> <p>In questo esempio, <code>PreUpdateHook</code> verrà eseguito prima dell'<code>vnf1.SampleVNF1</code> aggiornamento e <code>PostUpdateHook</code> verrà eseguito dopo l'<code>vnf1.SampleVNF1</code> aggiornamento al vnf pacchetto indicato dall'aggiornamento <code>uuid</code> per il namespace <code>vnf1</code>.</p>	<pre>vnfds:   - descriptor_id:     "43c012fa-2616-41a8-     a833-0dfd4c5a049e "     namespace: " vnf1"   - descriptor_id:     "64222f98-ecd6-4871-     bf94-7354b53f3ee5 "     namespace: " vnf2"   ...  SampleVNF1HelmDeploy:   type: tosca.nod es.AWS.Deployment. VNFDeployment   requirements:     cluster: EKSCluster   vnfs:     - vnf1.SampleVNF1     - vnf2.Sample VNF2</pre>	<pre>vnfd - des r_id "0e bd87 - b8a1 4666 "  nam : "vr - des r_id "64 ecd6 - bf94 4b53 "  nam : "vr ... S amp1 Helm y:  typ</pre>

Parametro	Description	Esempio: Prima

Esem  
dopo

tos  
es.A  
ploy  
VNFD  
ment

rec  
nts:

clu  
EKS  
r

vnf

- v  
LeVN  
A  
VNF  
upc  
as  
the  
uui  
cha  
fo  
nam  
"vr

- v

Parametro	Description	Esempio: Prima

Esem  
dopo

*leVM*  
No  
cha  
to  
thi  
fur  
as  
nam  
and  
uui  
rem  
the  
sam

int  
s:

Hoc

pre  
e:  
*Hook*

pos  
te:  
*eHoc*

Parametro	Description	Esempio: Prima	Esem dopo
Sottoreti	È possibile aggiungere ed eliminare sottoreti dalla rete. Prima di eliminare una sottorete, verificate che la sottorete non venga utilizzata da alcuna risorsa della rete.	<pre>Free5GCSubnet01 : #Deleted Subnet   type: toscanetworking. es.AWS.Networking. Subnet   properties:     type: "PUBLIC"     availability_zone: { get_input: subnet_01 _az }     cidr_block: { get_input: subnet_01 _cidr_block }   requirements:     route_table: Free5GCRouteTable     vpc: Free5GCVPC</pre>	<pre>Free bnet #Ne Sub typ tos es.A two Subr pro s: typ "PU ava ity_ { g : sub _az cid k: { g : sub _cid ck } reo nts:</pre>

Parametro	Description	Esempio: Prima

Esem  
dopo

rou  
le:  
Fre  
uteT

vpo  
Fre  
C

Parametro	Description	Esempio: Prima	Esem dopo
Gruppi di sicurezza	È possibile aggiungere ed eliminare gruppi di sicurezza dalla rete. Prima di eliminare un gruppo di sicurezza , verifica che il gruppo di sicurezza non venga utilizzato da alcuna risorsa della rete.	<pre> Free5GCSecurityGroup01 : #Deleted Security Group   type: tosca.nodes.AWS.Networking.SecurityGroup   properties:     description: "SecurityGroup for Free5GC cluster"     name: "Free5GCSecurityGroup01"     tags:       - "Name=Free5GCAdditionalSecurityGroup"     requirements:       vpc: Free5GCVPC  Free5GCSecurityGroupEgressRule01 : #Deleted Security Group Egress Node   type: tosca.nodes.AWS.Networking.SecurityGroupEgressRule   properties:     ip_protocol: "tcp"     from_port: 8000     to_port: 9000     description: "Egress Rule for free5GC cluster"     cidr_ip : "172.10.10.1/24"     requirements:       security_group: Free5GCSecurityGroup01 </pre>	<pre> Free5GCSecurityGroup02 : #New Security Group   type: tosca.nodes.AWS.Networking.SecurityGroup   properties:     description: "SecurityGroup for Free5GC cluster"     name: "Free5GCSecurityGroup02"     tags:       - "Name=Free5GCAdditionalSecurityGroup"     requirements:       vpc: Free5GCVPC  Free5GCSecurityGroupEgressRule02 : #New Security Group Egress Node   type: tosca.nodes.AWS.Networking.SecurityGroupEgressRule   properties:     ip_protocol: "tcp"     from_port: 8000     to_port: 9000     description: "Egress Rule for free5GC cluster"     cidr_ip : "172.10.10.1/24"     requirements:       security_group: Free5GCSecurityGroup02 </pre>

Parametro	Description	Esempio: Prima	Esem dopo
		<pre> Free5GCSecurityGro upIngressRule01 : #Deleted Security Group Ingress Node   type: toasca.nod es.AWS.Networking. SecurityGroupIngre ssRule   properties:     ip_protocol: "tcp"     from_port: 8000     to_port: 9000     description:       "Ingress Rule for free5GC cluster"       cidr_ip: "172.10.1 0.1/24"     requirements:       security_group: Free5GCSecurityGro up01                     </pre>	<p>- "Na e5GC diti ecur oup"  rec nts:  vpo Fre C  <i>Free</i> <i>curi</i> <i>upEg</i> <i>ule0</i> #Ne Sec Gro Egr Noc  typ tos es.A twor Secu roup sRuL  pro s:</p>

Parametro	Description	Esempio: Prima

Esem  
dopo

ip\_  
ol:  
"to

fro  
:  
800

to\_  
900

des  
on:  
"Eg  
RUL  
for  
fre  
clu

cic  
"17  
0.1/

rec  
nts:

sec  
grou  
Fre

Parametro	Description	Esempio: Prima

Esem  
dopo

curi  
up02

*Free*  
*curi*  
*upIn*  
*Rule*

#Ne  
Sec  
Gro  
Ing  
Noc

typ  
tos  
es.A  
twor  
Secu  
roup  
ssRu

pro  
s:

ip\_  
ol:  
"to

fro  
:  
800

to\_  
900

Parametro	Description	Esempio: Prima

Esem  
dopo

des  
on:  
"In  
RuL  
for  
fre  
clu

cio  
"17  
0.1/

rec  
nts:

sec  
grou  
Fre  
curi  
up02

Parametro	Description	Esempio: Prima	Esem dopo
Interfacce di rete	È possibile aggiungere, modificare ed eliminare ENIs dalla rete.	<pre>Free5GCENI01: #Modified ENI   type: tosca.nod es.AWS.Networking.ENI   properties:     device_index: 2   requirements:     subnet: <i>Free5GCENI</i> <i>ISubnet01</i>   security_groups:     - Free5GCSe curityGroup01  Free5GCENI02: #Modified ENI   type: tosca.nod es.AWS.Networking.ENI   properties:     device_index: 3     source_dest_check: true   requirements:     subnet: Free5GCENI ISubnet01  <i>Free5GCENI04</i> : #Deleted ENI   type: tosca.nod es.AWS.Networking.ENI   properties:     device_index: 4     source_dest_check: true   requirements:     subnet: Free5GCENI ISubnet01</pre>	<pre>Free I01: #Mo ENI typ tos es.A two ENI pro s:  dev dex: 2 rec nts:  sub <i>ISub</i>  sec grou  - Fre</pre>

Parametro	Description	Esempio: Prima

Esem  
dopo

curi  
up01  
Fre  
e5G  
:  
#Mo  
ENI

typ  
tos  
es.A  
twor  
ENI

pro  
s:

dev  
dex:  
3

sou  
st\_c  
tru

rec  
nts:

sub  
Fre  
ISub

se  
grou

Parametro	Description	Esempio: Prima

Esem  
dopo

-  
Fre  
curi  
up01  
Free  
I03  
#Ne  
ENI

typ  
tos  
es.A  
twor  
ENI

pro  
s:

dev  
dex:  
3

rec  
nts:

sub  
Fre  
bnet

sec  
grou

Parametro	Description	Esempio: Prima	Esem dopo
			- Fre curi up01

## Aggiornamento di un'istanza di rete

### Console

Per aggiornare un'istanza di rete utilizzando la console

1. Apri la console AWS TNB 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, vedere [Parametri che è possibile aggiornare](#).
7. Scegli Aggiorna rete.

AWS TNB convalida la richiesta e avvia la distribuzione. Viene visualizzata la pagina di 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

Usa la CLI per aggiornare un'istanza di rete

Usa il [update-sol-network-instance](#) comando 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 console AWS TNB 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 AWS CLI

1. Usa il [list-sol-network-instances](#) comando per elencare le tue istanze di rete.

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

2. Usa il [get-sol-network-instance](#) comando 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 console AWS TNB 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-instance](#) comando 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-instance](#) comando 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'istanziamento o la chiusura di un'istanza di rete.

## Attività

- [Visualizza un' AWS operazione di rete TNB](#)
- [Annulla un'operazione di rete AWS TNB](#)

## Visualizza un' AWS operazione di rete 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 console AWS TNB 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-operations](#) comando per elencare tutte le operazioni di rete.

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

2. Utilizzare il [get-sol-network-operation](#) comando 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}$
```

# Annulla un'operazione di rete AWS TNB

Scopri come annullare un'operazione di rete.

## Console

Per annullare un'operazione di rete utilizzando la console

1. Aprire la console AWS TNB 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-operation](#) comando per annullare un'operazione di rete.

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

# Riferimento TOSCA per AWS TNB

Topology and Orchestration Specification for Cloud Applications (TOSCA) è una sintassi dichiarativa che viene utilizzata per descrivere una topologia dei servizi Web basati sul cloud, i relativi componenti, le relazioni e i processi che li gestiscono. TOSCA descrive 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. TOSCA quindi carica il modello su AWS TNB che sintetizza le risorse necessarie per stabilire una rete 5G funzionante tra le zone di disponibilità. AWS

## Indice

- [Modello VNFD](#)
- [Modello di descrittore del servizio di rete](#)
- [Nodi comuni](#)

## Modello VNFD

Definisce un modello VNFD (Virtual Network Function Descriptor).

## 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 nodi TOSCA AWS . I nodi possibili sono:

- [AWS.VNF](#)
- [AWS.Artefatti.Elmo](#)

## 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

L'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 del 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 del 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: toska.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 pacchetto CSAR.

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 NSD (Network Service Descriptor).

## 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 blocco CIDR per il nodo VPC, è possibile utilizzare la `{ get_input: input-parameter-name }` sintassi e definire i parametri nel modello NSD. 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 }
```

## Importazione VNFD

### descriptor\_id

L'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}

namespace

Il nome univoco.

Campo obbligatorio: sì

Tipo: stringa

## Modello di topologia

node\_templates

I possibili AWS nodi TOSCA sono:

- [AWS.NS](#)
- [AWS.Compute.eks](#)
- [AWS.Compute.eks. AuthRole](#)
- [AWS.Calcola. EKSMangedNodo](#)
- [AWS.Calcola. EKSSelfManagedNode](#)
- [AWS.Calcola. PlacementGroup](#)
- [AWS.Calcola. UserData](#)
- [AWS.Rete. SecurityGroup](#)
- [AWS.Rete. SecurityGroupEgressRule](#)
- [AWS.Rete. SecurityGroupIngressRule](#)
- [AWS.Risorsa. Importazione](#)
- [AWS.Networking.eni](#)
- [AWS.HookExecution](#)
- [AWS.Rete. InternetGateway](#)
- [AWS.Rete. RouteTable](#)
- [AWS.Networking.Subnet](#)
- [AWS.Distribuzione. VNFDeployment](#)

- [AWS.Networking.vpc](#)
- [AWS.Rete. NATGateway](#)
- [AWS.Rete. Percorso](#)

## AWS.NS

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

### Sintassi

```
tosca.nodes.AWS.NS:  
  properties:  
    descriptor\_id: String  
    descriptor\_version: String  
    descriptor\_name: String
```

### Proprietà

#### descriptor\_id

L'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 dell'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: toska.nodes.AWS.NS
  properties:
    descriptor_id: "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    descriptor_version: "1.0.0"
    descriptor_name: "Test NS Template"
```

## AWS.compute.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 (CNI) sono abilitati. 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

```
toska.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 includi `multus`, specifica le proprietà `enabled` and `multus_role`.

#### `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 cluster Amazon EKS.

Abilita questo plug-in per utilizzare i nodi autogestiti di Amazon EKS su AWS Outposts, AWS Local Zones o Regioni AWS. Per ulteriori informazioni, consulta il [driver CSI di Amazon Elastic Block Store](#) nella Guida per l'utente di Amazon EKS.

#### `enabled`

Indica se è installato il driver Amazon EBS CSI predefinito.

Campo obbligatorio: no

Tipo: Booleano

## version

La versione del componente aggiuntivo del driver CSI di Amazon EBS. La versione deve corrispondere a una delle versioni restituite dall'azione. `DescribeAddonVersions` Per ulteriori informazioni, [DescribeAddonVersions](#) consulta 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 1.25 a 1.32.

Campo obbligatorio: sì

Tipo: stringa

Valori possibili: 1,25 | 1,26 | 1,27 | 1,28 | 1,29 | 1,30 | 1,31 | 1,32

### 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: toasca.nodes.AWS.Compute.EKS
  properties:
    version: "1.26"
    access: "ALL"
    cluster_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
    ip_family: "IPv6"
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  capabilities:
    multus:
      properties:
        enabled: true
        multus_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/MultusRole"
    ebs_csi:
      properties:
```

```
    enabled: true
    version: "v1.16.0-eksbuild.1"
  requirements:
    subnets:
      - SampleSubnet01
      - SampleSubnet02
```

## AWS.Compute.eks. AuthRole

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

### 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 i ruoli IAM che devono essere aggiunti al cluster Amazon EKS. `aws-auth ConfigMap`

`arn`

L'ARN del 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 nodo .compute.eks.AWS](#)

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. EKSMangedNodo

AWS TNB supporta i gruppi di nodi gestiti EKS per automatizzare il provisioning e la gestione del ciclo di vita dei nodi (istanze Amazon EC2 ) per i cluster Amazon EKS Kubernetes. Per creare un gruppo di nodi EKS, procedi come segue:

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

- 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
        root_volume_size: Integer
    scaling:
      properties:
        desired_size: Integer
        min_size: Integer
        max_size: Integer
  properties:
    node_role: String
    tags: List
    kubernetes_version: String
  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 gestiti di Amazon EKS, ad esempio i tipi di EC2 istanze Amazon e le EC2 istanze Amazon AMLs.

## ami\_type

Il tipo di AMI supportato da Amazon EKS.

Campo obbligatorio: sì

Tipo: stringa

Valori possibili: AL2\_x86\_64 | AL2\_x86\_64\_GPU | AL2\_ARM\_64 | AL2023\_x86\_64 | AL2023\_ARM\_64 | AL2023\_x86\_64\_NVIDIA | AL2023\_x86\_64\_NEURON | CUSTOM | BOTTLEROCKET\_ARM\_64 | BOTTLEROCKET\_x86\_64 | BOTTLEROCKET\_ARM\_64\_NVIDIA | BOTTLEROCKET\_x86\_64\_NVIDIA

## ami\_id

L'ID dell'AMI.

Required: No

Tipo: stringa

### Note

Se entrambi `ami_type` e `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 l'accesso SSH.

Campo obbligatorio: sì

Tipo: stringa

## root\_volume\_encryption

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

Campo obbligatorio: no

Impostazione predefinita: true

Tipo: Booleano

## root\_volume\_encryption\_key\_arn

L'ARN della chiave. AWS KMS AWS TNB supporta ARN a chiave normale, ARN a chiave multiregione e alias ARN.

Required: No

Tipo: stringa

### Note

- Se `root_volume_encryption` è falso, non includerlo.  
`root_volume_encryption_key_arn`
- AWS TNB supporta la crittografia del volume principale delle AMI supportate da Amazon EBS.
- Se il volume root dell'AMI è già crittografato, devi includere il codice `root_volume_encryption_key_arn` per consentire a AWS TNB di ricrittografare il volume root.
- Se il volume root dell'AMI non è crittografato, AWS TNB utilizza il `root_volume_encryption_key_arn` per crittografare il volume root.  
  
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 root.
- AWS TNB non decrittografa un AMI crittografato.

## root\_volume\_size

La dimensione del volume root di Amazon Elastic Block Store in GiBs.

Campo obbligatorio: no

Impostazione predefinita: 20

Tipo: integer

Valori possibili: da 1 a 16.384

## scaling

Proprietà che definiscono i parametri di scalabilità per il gruppo di nodi gestiti Amazon EKS, 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

L'ARN del ruolo IAM collegato all'istanza Amazon EC2 .

Campo obbligatorio: sì

Tipo: stringa

tags

I tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

kubernetes\_version

La versione Kubernetes per il gruppo Managed Node. AWS TNB supporta le versioni di Kubernetes da 1.25 a 1.32. Considera i seguenti aspetti:

- `kubernetes_versionami_id` Specificare o. Non specificare entrambi.
- `kubernetes_version` Deve essere minore o uguale a `AWS.Compute.EKSManagedVersion` del nodo.
- Può esserci una differenza di 3 versioni tra `AWS.Compute.EKSManagedVersion` del nodo e `kubernetes_version`
- Se non viene specificato nessuno dei due `kubernetes_version` o `ami_id`, AWS TNB utilizzerà l'AMI più recente della `AWS.Compute.EKSManagedNode` versione per creare `EKSManagedNode`

Required: No

Tipo: stringa

Valori possibili: 1,25 | 1,26 | 1,27 | 1,28 | 1,29 | 1,30 | 1,31 | 1,32

## Requisiti

cluster

[AWS Un](#) nodo .compute.eks.

Campo obbligatorio: sì

Tipo: stringa

## subnets

[Un AWS nodo .Networking.Subnet.](#)

Campo obbligatorio: sì

Tipo: List

## network\_interfaces

Un nodo [AWS.Networking.eni](#). Assicurati che le interfacce di rete e le sottoreti siano impostate sulla stessa zona di disponibilità o l'istanziamento avrà esito negativo.

[Quando si impostanetwork\\_interfaces, AWS TNB ottiene l'autorizzazione relativa alla ENIs multus\\_role proprietà se è stata inclusa la proprietà nel nodo AWS.Compute.eks. multus](#)  
[Altrimenti, AWS TNB ottiene l'autorizzazione relativa alla proprietà node\\_role. ENIs](#)

Campo obbligatorio: no

Tipo: List

## security\_groups

[Un .Networking.AWS SecurityGroupnodo.](#)

Campo obbligatorio: no

Tipo: List

## placement\_group

Un [tosca.nodes.AWS.Calcola. PlacementGroup](#)nodo.

Required: No

Tipo: stringa

## user\_data

Un [tosca.nodes.AWS.Calcola. UserData](#) riferimento 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

```
SampleEKSMangedNode:
  type: tosca.nodes.AWS.Compute.EKSMangedNode
  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"
        root_volume_size: 1500
      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"
kubernetes_version:
  - "1.30"
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 TNB supporta i nodi autogestiti di Amazon EKS per automatizzare il provisioning e la gestione del ciclo di vita dei nodi (istanze Amazon EC2 ) per i cluster Amazon EKS Kubernetes. Per creare un gruppo di nodi Amazon EKS, procedi come segue:

- Scegli Amazon Machine Images (AMI) per i tuoi nodi cluster worker fornendo l'ID dell'AMI.
- Fornisci una coppia di EC2 chiavi Amazon per l'accesso SSH.
- Assicurati che il tuo gruppo di nodi sia associato a un cluster Amazon EKS.
- 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
root_volume_size: Integer
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 autogestiti di Amazon EKS, ad esempio i tipi di EC2 istanze Amazon e le EC2 istanze AMIs Amazon.

#### ami\_id

L'ID AMI utilizzato per avviare l'istanza. AWS TNB supporta istanze che sfruttano IMDSv2 Per ulteriori informazioni, consulta [Versione IMDS](#).

#### Note

Puoi aggiornare l'ID AMI per `EKSSelfManagedNode`. La versione Amazon EKS dell'AMI deve essere uguale o inferiore a due versioni rispetto alla versione del cluster Amazon EKS. Ad esempio, se la versione del cluster Amazon EKS è 1.31, la versione dell'AMI Amazon EKS deve essere 1.31, 1.30 o 1.29.

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 l'accesso SSH.

Campo obbligatorio: sì

Tipo: stringa

`root_volume_encryption`

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

Campo obbligatorio: no

Impostazione predefinita: true


Tipo: Booleano

`root_volume_encryption_key_arn`

L'ARN della chiave. AWS KMS AWS TNB supporta ARN a chiave normale, ARN a chiave multiregione e alias ARN.

Required: No

Tipo: stringa

 Note

- Se `root_volume_encryption` è falso, non includerlo.  
`root_volume_encryption_key_arn`

- AWS TNB supporta la crittografia del volume principale delle AMI supportate da Amazon EBS.
- Se il volume root dell'AMI è già crittografato, devi includere il codice `root_volume_encryption_key_arn` per consentire a AWS TNB di ricrittografare il volume root.
- Se il volume root dell'AMI non è crittografato, AWS TNB utilizza il `root_volume_encryption_key_arn` per crittografare il volume root.

Se non lo includi `root_volume_encryption_key_arn`, AWS TNB lo utilizza AWS Managed Services per crittografare il volume root.

- AWS TNB non decrittografa un AMI crittografato.

## `root_volume_size`

La dimensione del volume root di Amazon Elastic Block Store in GiBs.

Campo obbligatorio: no

Impostazione predefinita: 20

Tipo: integer

Valori possibili: da 1 a 16.384

## ***scaling***

Proprietà che definiscono i parametri di scalabilità per i nodi autogestiti di Amazon EKS, 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

L'ARN del ruolo IAM collegato all'istanza Amazon EC2 .

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 nodo [AWS.compute.eks](#).

Campo obbligatorio: sì

Tipo: stringa

## subnets

[Un AWS nodo .Networking.Subnet.](#)

Campo obbligatorio: sì

Tipo: List

## network\_interfaces

Un nodo [AWS.Networking.eni](#). Assicurati che le interfacce di rete e le sottoreti siano impostate sulla stessa zona di disponibilità o l'istanziamento avrà esito negativo.

[Quando si impostanetwork\\_interfaces, AWS TNB ottiene l'autorizzazione relativa alla ENIs multus\\_role proprietà se è stata inclusa la proprietà nel nodo AWS.Compute.eks. multus](#)  
[Altrimenti, AWS TNB ottiene l'autorizzazione relativa alla proprietà node\\_role. ENIs](#)

Campo obbligatorio: no

Tipo: List

## security\_groups

[Un .Networking.AWS SecurityGroupnodo.](#)

Campo obbligatorio: no

Tipo: List

## placement\_group

Un [tosca.nodes.AWS.Calcola. PlacementGroup](#)nodo.

Required: No

Tipo: stringa

## user\_data

Un [tosca.nodes.AWS.Calcola. UserData](#) riferimento 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"
        root_volume_size: 1500
      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 avvia un nuovo Amazon EC2 instance, 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 HPC (High Performance Computing).
- **PARTIZIONE:** 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:** posiziona un piccolo gruppo di istanze su hardware sottostante distinto per ridurre i guasti correlati.
- **SPREAD\_HOST** — utilizzato solo con i gruppi di posizionamento 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: toscanodes.AWS.Compute.PlacementGroup
  properties:
    strategy: "PARTITION"
    partition_count: 5
    tags:
      - tag_key=tag_value
```

## AWS.Calcola. UserData

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

Durante la creazione di istanze di rete, AWS TNB fornisce la registrazione dell' EC2 istanza 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 registrazione Amazon EKS.

Per utilizzare variabili personalizzate nello script dei dati utente, aggiungi un punto esclamativo ! dopo la parentesi riccia aperta. { Ad esempio, per utilizzarle MyVariable nello script, inserisci: { ! MyVariable }

### Note

- AWS TNB supporta script di dati utente di dimensioni fino a 7 KB.
- Poiché AWS TNB utilizza CloudFormation per elaborare e rendere lo script multimime dei dati utente, assicurati che lo script rispetti tutte le regole. 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 TNB supporta 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 nodo [AWS.networking.vpc](#).

Campo obbligatorio: sì

Tipo: stringa

## Esempio

```
SampleSecurityGroup001:
  type: toscanodes.AWS.Networking.SecurityGroup
  properties:
    description: "Sample Security Group for Testing"
    name: "SampleSecurityGroup"
    tags:
      - "Name=SecurityGroup"
      - "Environment=Testing"
```

```
requirements:  
  vpc: SampleVPC
```

## AWS.Rete. SecurityGroupEgressRule

AWS TNB supporta le regole di uscita dei gruppi di sicurezza per automatizzare il provisioning delle EC2 Amazon 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'intervallo di indirizzi in formato CIDR. IPv4 È necessario specificare un intervallo CIDR che consenta il traffico in uscita.

Required: No

Tipo: stringa

#### `cidr_ipv6`

L'intervallo di IPv6 indirizzi in formato CIDR, per il traffico in uscita. Devi specificare un gruppo di sicurezza di destinazione (`destination_security_group` o `destination_prefix_list`) o un intervallo CIDR (`cidr_ip` o `cidr_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 gestiti Amazon VPC 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 o UDP, questo è l'inizio dell'intervallo di porte. Se il protocollo è ICMP o ICMPv6, questo è il tipo di numero. Il valore -1 indica tutti e ICMP/ICMPv6 types. If you specify all ICMP/ICMPv6 types, you must specify all ICMP/ICMPv 6 i codici.

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 o UDP, questa è la fine dell'intervallo di porte. Se il protocollo è ICMP o ICMPv6, questo è il codice. Il valore -1 indica tutti e ICMP/ICMPv6 codes. If you specify all ICMP/ICMPv6 types, you must specify all ICMP/ICMPv 6 i codici.

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

Required: No

Tipo: stringa

## Esempio

```
SampleSecurityGroupEgressRule:
  type: toska.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 TNB supporta le regole di ingresso dei gruppi di sicurezza per automatizzare il provisioning delle EC2 Amazon 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'intervallo di indirizzi in formato CIDR. IPv4 È necessario specificare un intervallo CIDR che consenta il traffico in ingresso.

Required: No

Tipo: stringa

#### cidr\_ipv6

L'intervallo di IPv6 indirizzi in formato CIDR, per il traffico in ingresso. È necessario specificare un gruppo di sicurezza di origine (source\_security\_group o source\_prefix\_list) o un intervallo CIDR (cidr\_ip o cidr\_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 gestiti Amazon VPC 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 o UDP, questo è l'inizio dell'intervallo di porte. Se il protocollo è ICMP o ICMPv6, questo è il tipo di numero. Il valore -1 indica tutti e ICMP/ICMPv6 types. If you specify all ICMP/ICMPv6 types, you must specify all ICMP/ICMPv 6 i codici.

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 o UDP, questa è la fine dell'intervallo di porte. Se il protocollo è ICMP o ICMPv6, questo è il codice. Il valore -1 indica tutti e ICMP/ICMPv6 codes. If you specify all ICMP/ICMPv6 types, you must specify all ICMP/ICMPv 6 i codici.

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

Required: No

Tipo: stringa

## Esempio

```
SampleSecurityGroupIngressRule:
  type: toska.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 che viene importata in AWS TNB.

Campo obbligatorio: no

Tipo: List

#### resource\_id

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

Campo obbligatorio: no

Tipo: List

### Esempio

```
SampleImportedVPC:
  type: toasca.nodes.AWS.Resource.Import
```

```
properties:
  resource_type: "tosca.nodes.AWS.Networking.VPC"
  resource_id: "vpc-123456"
```

## AWS.Networking.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' EC2 istanza 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.SecurityGroup](#) nodo.

Required: No

Tipo: stringa

## Esempio

```
SampleENI:
  type: toska.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 nodo [AWS.Networking.vpc](#).

Campo obbligatorio: sì

Tipo: stringa

## Esempio

```
SampleHookExecution:
  type: toska.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 del VPC. È necessario includere la proprietà `dest_cidr` o `ipv6_dest_cidr`.

### `dest_cidr`

Il blocco IPv4 CIDR 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 blocco IPv6 CIDR 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 nodo [AWS.networking.vpc](#).

Campo obbligatorio: sì

Tipo: stringa

### route\_table

Un [AWS.Networking.RouteTable](#) nodo.

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: Free5GCPriateRouteTable
    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 VPC o del gateway. È necessario associare una tabella di routing a un 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 nodo [AWS.networking.vpc](#).

Campo obbligatorio: sì

Tipo: stringa

### Esempio

```
SampleRouteTable:  
  type: toasca.nodes.AWS.Networking.RouteTable  
  properties:  
    tags:
```

```
- "Name=SampleVPC"
- "Environment=Testing"
requirements:
  vpc: SampleVPC
```

## AWS.Networking.Subnet

Una sottorete è un intervallo di indirizzi IP nel tuo VPC e deve risiedere interamente all'interno di una zona di disponibilità. È necessario specificare un VPC, un blocco CIDR, 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 `west-2-lax-1a`.

Campo obbligatorio: sì

Tipo: stringa

`cidr_block`

Il blocco CIDR per la sottorete.

Required: No

Tipo: stringa

`ipv6_cidr_block`

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

`ipv6_cidr_block_suffix`

Required: No

Tipo: stringa

`ipv6_cidr_block_suffix`

Il suffisso esadecimale a 2 cifre del blocco IPv6 CIDR 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`

L'ARN in AWS Outposts cui verrà creata la sottorete. Aggiungi questa proprietà al modello NSD se desideri avviare nodi autogestiti di Amazon EKS su. AWS Outposts Per ulteriori informazioni, consulta [Amazon EKS AWS Outposts nella](#) Guida per l'utente di Amazon EKS.

Se aggiungi questa proprietà al modello NSD, 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 nodo [AWS.networking.vpc](#).

Campo obbligatorio: sì

Tipo: stringa

route\_table

Un [AWS.Networking.RouteTable](#) nodo.

Campo obbligatorio: sì

Tipo: stringa

## Esempio

```
SampleSubnet01:
  type: toasca.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: toska.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.Implementazione. VNFDeployment

Le implementazioni NF sono modellate fornendo l'infrastruttura e l'applicazione ad essa associate. L'attributo [cluster](#) specifica il cluster EKS su cui ospitare il tuo. NFs L'attributo [vnfs](#) specifica le funzioni di rete per la distribuzione. Puoi anche fornire operazioni opzionali di lifecycle hook di tipo [pre\\_create e post\\_create per eseguire istruzioni specifiche per la tua implementazione, ad esempio richiamare un'API](#) del sistema di gestione dell'inventario.

### Sintassi

```
toska.nodes.AWS.Deployment.VNFDeployment:
```

```
requirements:
```

```
deployment: String
```

```
cluster: String
```

```
vnfs: List
```

```
interfaces:
```

```
Hook:
```

```
pre\_create: String
```

```
post\_create: String
```

### Requisiti

deployment

[Un .Deployment.AWS VNFDeployment](#)nodo.

Required: No

Tipo: stringa

## cluster

Un nodo [AWS.compute.eks.](#)

Campo obbligatorio: sì

Tipo: stringa

## vnfs

[Un AWS nodo .VNF.](#)

Campo obbligatorio: sì

Tipo: stringa

## Interfacce

### Ganci

Definisce la fase in cui vengono eseguiti i lifecycle hook.

### pre\_create

[Un.AWS HookExecution](#)nodo. Questo hook viene eseguito prima della distribuzione del VNFDeployment nodo.

Required: No

Tipo: stringa

### post\_create

[Un AWS. HookExecution](#)nodo. 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.Networking.vpc

È necessario specificare un blocco CIDR 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 per il VPC, in notazione CIDR.

Campo obbligatorio: sì

Tipo: stringa

#### ipv6\_cidr\_block

Il blocco IPv6 CIDR utilizzato per creare il VPC.

Valore consentito: AMAZON\_PROVIDED

Required: No

Tipo: stringa

## dns\_support

Indica se le istanze avviate nel VPC ricevono nomi host DNS.

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 Gateway NAT pubblico o privato su una sottorete. Per un gateway pubblico, se non fornisci un ID di allocazione IP elastico, AWS TNB assegnerà 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. InternetGateway](#) riferimento 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

```
Free5GNatGateway01:
  type: toska.nodes.AWS.Networking.NATGateway
  requirements:
    subnet: Free5GSubnet01
    internet_gateway: Free5GCIGW
  properties:
    type: PUBLIC
    eip_allocation_id: eipalloc-12345
```

## AWS.Rete. Percorso

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

## Sintassi

```
toska.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

## Requisiti

### nat\_gateway

L'[AWS.Networking.NATGateway](#) riferimento al nodo.

Campo obbligatorio: sì

Tipo: stringa

### route\_table

L'[AWS.Networking.RouteTable](#) riferimento al nodo.

Campo obbligatorio: sì

Tipo: stringa

## Esempio

```
Free5GCRoute:
  type: toasca.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
```

## AWS.Negoziio. SSMPParameters

È possibile creare parametri SSM tramite TNB. AWS I parametri SSM che crei vengono creati in SSM e preceduti dall'ID dell'istanza di rete TNB. AWS Ciò impedisce che i valori dei parametri vengano sostituiti quando più istanze vengono istanziate e aggiornate utilizzando lo stesso modello NSD.

## Sintassi

```
tosca.nodes.AWS.Store.SSMPParameters
  properties:
    parameters:
      name: String
      value: String
```

[tags](#): List

## Proprietà

### Parametri

#### name

Il nome della proprietà ssm. Utilizza il seguente formato: `^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`

Il nome di ogni parametro deve contenere meno di 256 caratteri.

Campo obbligatorio: sì

Tipo: stringa

#### value

Il valore della proprietà ssm. Utilizza uno dei seguenti formati:

- Per valori senza riferimenti: `^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`
- Per riferimenti statici: `^\$\{[a-zA-Z0-9]+\.(properties|capabilities|requirements)(\.[a-zA-Z0-9\-\_]+)\}\$`
- Per riferimenti dinamici: `^\$\{[a-zA-Z0-9]+\.(name|id|arn)\}\$`

Il valore di ogni parametro deve essere inferiore a 4 KB.

Campo obbligatorio: sì

Tipo: stringa

#### tags

I tag che puoi allegare a una proprietà SSM.

Campo obbligatorio: no

Tipo: List

## Esempio

```
SampleSSM
```

```
type: toska.nodes.AWS.Store.SSMPParameters
properties:
  parameters:
    - name: "Name1"
      value: "Value1"
    - name: "EKS_VERSION"
      value: "${SampleEKS.properties.version}"
    - name: "VPC_ID"
      value: "${SampleVPC.id}"
    - name: "REGION"
      value: "${AWS::Region}"
  tags:
    - "tagKey=tagValue"
```

## Nodi comuni

Definire i nodi per NSD e VNFD.

- [AWS. HookDefinition.Bash](#)

## AWS.HookDefinition.Bash

Definisce un AWS HookDefinition inbash.

### Sintassi

```
toska.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 i seguenti modelli regex:

- Per valori senza riferimenti: `^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`
- Per riferimenti statici: `^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=\$\{[a-zA-Z0-9]+\.(properties|capabilities|requirements)(\.[a-zA-Z0-9\-\_]+)\}\}$`
- Per riferimenti dinamici: `^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=\$\{[a-zA-Z0-9]+\.(name|id|arn)\}\}$`

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 riservate AWS TNB (senza distinzione tra maiuscole e minuscole):
  - CODEBUILD
  - TNB
  - CASA
  - 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**
- Ogni variabile di ambiente (each **envName =envValue**) deve contenere meno di 128 caratteri.

Esempio: `A123-45xYz=Example_789`

Campo obbligatorio: no

Tipo: List

`execution_role`

Il ruolo per l'esecuzione degli hook.

Campo obbligatorio: sì

Tipo: stringa

## Esempio

```
SampleHookScript:
  type: toska.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. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi AWS TNB. I seguenti argomenti mostrano come configurare AWS TNB per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse AWS TNB.

## Indice

- [Protezione dei dati in AWS TNB](#)
- [Gestione delle identità e degli accessi per TNB AWS](#)
- [Convalida della conformità per AWS TNB](#)
- [Resilienza in AWS TNB](#)
- [Sicurezza dell'infrastruttura in TNB AWS](#)
- [Versione IMDS](#)

# 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 maggiori informazioni sulla privacy dei dati, consulta le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [AWS Modello di responsabilità condivisa e GDPR](#) nel AWS Blog sulla sicurezza.

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

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire 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 moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#).

Ti consigliamo 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 con AWS TNB o altri utenti Servizi AWS utilizzando la console, l'API o. AWS CLI 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. Quando si fornisce un URL a un server esterno,

suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al 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, AWS TNB elimina definitivamente i tuoi dati. AWS TNB chiude inoltre le reti ed elimina i pacchetti di funzioni e i pacchetti di rete.

## Crittografia dei dati a riposo

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

## Crittografia dei dati in transito

AWS TNB protegge 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 inter-rete

AWS Le risorse di calcolo TNB risiedono in un cloud privato virtuale (VPC) condiviso da tutti i clienti. Tutto il traffico AWS TNB 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 TNB AWS

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. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse TNB. AWS IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Indice

- [Destinatari](#)
- [Autenticazione con identità](#)

- [Gestione dell'accesso tramite policy](#)
- [Come funziona AWS TNB con IAM](#)
- [Esempi di policy basate sull'identità per Telco Network Builder AWS](#)
- [Risoluzione dei problemi relativi all'identità e all'accesso di AWS Telco Network Builder](#)

## Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta [Risoluzione dei problemi relativi all'identità e all'accesso di AWS Telco Network Builder](#))
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta [Come funziona AWS TNB con IAM](#))
- Amministratore IAM: scrivi policy per gestire l'accesso (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. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/Facebook Per ulteriori informazioni sull'accesso, consulta [Come accedere all' Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente di IAM.

## Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

## Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta [Che cos'è il Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

## Utenti e gruppi IAM

Un [utente IAM](#) è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente di IAM.

## Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

## Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste

politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

## Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scegliere tra policy gestite e policy in linea](#) nella Guida per l'utente di IAM.

## Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#).

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

## Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- **Limiti delle autorizzazioni:** imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- **Politiche di controllo del servizio (SCPs):** specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in AWS Organizations. Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations.
- **Politiche di controllo delle risorse (RCPs):** imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- **Policy di sessione:** policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

## Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

## Come funziona AWS TNB con IAM

Prima di utilizzare IAM per gestire l'accesso a AWS TNB, scopri quali funzionalità IAM sono disponibili per l'uso con TNB. AWS

Funzionalità IAM che puoi utilizzare con AWS Telco Network Builder

Funzionalità IAM	AWS Supporto TNB
<a href="#">Policy basate sull'identità</a>	Sì
<a href="#">Policy basate su risorse</a>	No
<a href="#">Operazioni di policy</a>	Sì
<a href="#">Risorse relative alle policy</a>	Sì
<a href="#">Chiavi di condizione delle policy</a>	Sì
<a href="#">ACLs</a>	No

Funzionalità IAM	AWS Supporto TNB
<a href="#">ABAC (tag nelle policy)</a>	Sì
<a href="#">Credenziali temporanee</a>	Sì
<a href="#">Autorizzazioni del principale</a>	Sì
<a href="#">Ruoli di servizio</a>	No
<a href="#">Ruoli collegati al servizio</a>	No

Per avere una visione di alto livello di come AWS TNB e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWS i servizi che funzionano con IAM nella IAM User Guide](#).

## Politiche basate sull'identità per TNB AWS

Supporta le policy basate sull'identità: sì

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

### Esempi di politiche basate sull'identità per TNB AWS

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

## Politiche basate sulle risorse all'interno di TNB AWS

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy di bucket

Amazon S3. 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 operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#). I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

## Azioni politiche per TNB AWS

Supporta le operazioni di policy: si

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

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni di eseguire l'operazione associata.

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

Le azioni politiche in AWS TNB 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 basate sull'identità di AWS TNB, vedere. [Esempi di policy basate sull'identità per Telco Network Builder AWS](#)

## Risorse politiche per TNB AWS

Supporta le risorse relative alle policy: sì

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

L'elemento `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (\*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"

```

Per visualizzare un elenco dei tipi di risorse AWS TNB e relativi ARNs, vedere [Risorse definite da AWS Telco Network Builder](#) nel Service Authorization Reference. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, vedere [Azioni definite da AWS Telco Network Builder](#).

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

## Chiavi relative alle condizioni delle politiche per TNB AWS

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

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

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile 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. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Per visualizzare un elenco di chiavi di condizione AWS TNB, consulta 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'identità di AWS TNB, vedere. [Esempi di policy basate sull'identità per Telco Network Builder AWS](#)

## ACLs AWS in TNB

Supporti ACLs: 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. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

## ABAC con TNB AWS

Supporta ABAC (tag nelle policy): sì

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire 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 Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta [Definizione delle autorizzazioni con autorizzazione ABAC](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

## Utilizzo di credenziali temporanee con TNB AWS

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono un accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza temporanee in IAM](#) e [Servizi AWS compatibili con IAM](#) nella Guida per l'utente IAM.

## Autorizzazioni principali multiservizio per TNB AWS

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso diretto (FAS) utilizzano le autorizzazioni del principale chiamante an Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).

## Ruoli di servizio per TNB AWS

Supporta i ruoli di servizio: no

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.

## Ruoli collegati ai servizi per TNB AWS

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'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

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

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare risorse AWS TNB. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM \(console\)](#) nella Guida per l'utente di IAM.

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](#)

- [Utilizzo della console TNB AWS](#)
- [Esempi di policy relative al ruolo del servizio](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)

## Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare AWS le risorse TNB nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire 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 maggiori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente di IAM.

- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare [Protezione dell'accesso API con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

## Utilizzo della console TNB AWS

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 AWS TNB presenti nel 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 consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, è opportuno concedere l'accesso solo alle azioni che corrispondono all'operazione API che stanno cercando di eseguire.

## Esempi di policy relative al ruolo del servizio

In qualità di amministratore, possiedi e gestisci le risorse create da AWS TNB come definito dall'ambiente e dai modelli di servizio. È necessario collegare i ruoli di servizio IAM al proprio account per consentire a AWS TNB di creare risorse per la gestione del ciclo di vita della rete.

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

Puoi creare il ruolo del servizio e le sue policy di autorizzazione con il servizio IAM. Per ulteriori informazioni sulla creazione di un ruolo di servizio, consulta [Creating a role to delegate permissions to an AWS service](#) nella IAM User Guide.

## AWS Ruolo di servizio TNB

In qualità di membro del team della piattaforma, puoi in qualità di amministratore creare un ruolo di servizio AWS TNB e fornirlo a AWS TNB. Questo ruolo consente a AWS TNB di effettuare chiamate ad altri servizi come Amazon Elastic Kubernetes CloudFormation Service 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 il seguente ruolo IAM e la politica di fiducia per il tuo ruolo di servizio TNB. AWS Quando definisci le autorizzazioni relative a questa policy, tieni presente che AWS TNB potrebbe fallire con gli errori Access Denied relativi alle risorse escluse dalla tua policy.

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

JSON

```
{
  "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"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "TNBAccessComputePerms"
},
{
    "Resource": "*",
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "ec2.amazonaws.com",
                "eks.amazonaws.com",
                "eks-nodegroup.amazonaws.com",
                "events.amazonaws.com",
                "autoscaling.amazonaws.com",
                "codebuild.amazonaws.com"
            ]
        }
    }
},
{
    "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:ListStackResources",
    "cloudformation:UpdateStack",
    "cloudformation:UpdateTerminationProtection",
    "ssm:PutParameter",
    "ssm:GetParameters",
    "ssm:GetParameter",
    "ssm>DeleteParameter",
    "ssm:AddTagsToResource",
    "ssm:ListTagsForResource",
    "ssm:RemoveTagsFromResource"
  ],
  "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**",
        "arn:aws:ssm:*:*:parameter/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 politica di attendibilità del servizio AWS TNB:

JSON

```
{
  "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 Ruolo del servizio TNB per il cluster Amazon EKS

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

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

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

- [AWS::IAM::Role](#)
- [Selezione di un modello di stack](#)

## AWS Ruolo del servizio TNB per il gruppo di nodi Amazon EKS

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

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

```

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

- [AWS::IAM::Role](#)
- [Selezione di un modello di stack](#)

### AWS Ruolo di servizio TNB per Multus

Quando crei una risorsa Amazon EKS nel tuo NSD e desideri gestire Multus come parte del 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 CloudFormation modello che crea un ruolo di servizio AWS TNB 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 ruoli IAM che utilizzano il AWS CloudFormation modello, consulta le seguenti sezioni della Guida per l'AWS CloudFormation utente:

- [AWS::IAM::Role](#)
- [Selezione di un modello di stack](#)

AWS Ruolo di servizio TNB per una policy di gestione del ciclo di vita

Quando l'NSD o il 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 CloudFormation modello che crea un ruolo di servizio AWS TNB 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 ruoli IAM che utilizzano il AWS CloudFormation modello, consulta le seguenti sezioni della Guida per l'AWS CloudFormation utente:

- [AWS::IAM::Role](#)
- [Selezione di un modello di stack](#)

## Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa policy include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l' AWS CLI API o. AWS

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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 aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AWS TNB e IAM.

### Problemi

- [Non sono autorizzato a eseguire un'azione in TNB AWS](#)

- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse AWS TNB](#)

## Non sono autorizzato a eseguire un'azione in TNB AWS

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` IAM prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa `my-example-widget` fittizia ma non dispone di autorizzazioni `tnb:GetWidget` fittizie.

```
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 aiuto, contatta il tuo AWS amministratore. 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'azione `iam:PassRole`, 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.

Il seguente errore di esempio si verifica quando un utente IAM 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 trasmettere 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 aiuto, contatta il tuo AWS 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 risorse AWS TNB

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 maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS TNB supporta queste funzionalità, consulta [Come funziona AWS TNB con IAM](#)
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà](#) nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(federazione delle identità\)](#) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente di 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 [Scope by Compliance Program Servizi AWS](#) 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. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la [Documentazione AWS sulla sicurezza](#).

## Resilienza in AWS TNB

L'infrastruttura AWS globale è costruita attorno a Regioni AWS zone 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 è possibile 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 TNB esegue il servizio di rete su cluster EKS in un cloud privato virtuale (VPC) nella AWS regione scelta.

## Sicurezza dell'infrastruttura in TNB AWS

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.

Utilizzate chiamate API AWS pubblicate per accedere a AWS TNB attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

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 delle applicazioni (API) AWS TNB
- 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):
  - Utenti, gruppi, ruoli e politiche IAM
  - Bucket S3 che usi per archiviare i tuoi dati per TNB AWS
  - Altro Servizi AWS e risorse che utilizzi per supportare il servizio di rete che hai fornito tramite TNB AWS
  - Il codice dell'applicazione
  - Connessioni tra il servizio di rete fornito tramite AWS TNB e i suoi client

#### Important

Sei responsabile dell'implementazione di un piano di disaster recovery in grado di ripristinare efficacemente un servizio di rete fornito tramite TNB. AWS

## Modello di sicurezza della connettività di rete

I servizi di rete forniti tramite AWS TNB vengono eseguiti su istanze di calcolo all'interno di un cloud privato virtuale (VPC) situato in una AWS regione selezionata. Un 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 tramite AWS TNB per tutti i clienti che operano nella stessa regione condividono lo stesso VPC. I servizi di rete forniti tramite AWS TNB per clienti diversi utilizzano istanze di elaborazione separate all'interno dello stesso VPC.

Le comunicazioni tra i client dei servizi di rete e il servizio di rete in AWS TNB attraversano Internet. AWS TNB non gestisce queste connessioni. È tua responsabilità proteggere le connessioni dei tuoi clienti.

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

## Versione IMDS

AWS TNB supporta istanze che sfruttano Instance Metadata Service versione 2 (IMDSv2), un metodo orientato alla sessione. IMDSv2 include IMDSV1 una sicurezza superiore a. Per ulteriori informazioni, consulta [Aggiungere una difesa approfondita contro firewall aperti, reverse proxy e vulnerabilità SSRF con miglioramenti al](#) servizio di metadati delle istanze Amazon EC2.

Quando IMDSv2 avvii l'istanza, devi usare. Per ulteriori informazioni IMDSv2, consulta [Use IMDSv2](#) in the Amazon EC2 User Guide.

# Monitoraggio AWS TNB

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di AWS TNB e delle altre AWS soluzioni. AWS consente AWS CloudTrail di guardare AWS TNB, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario.

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 log contengono informazioni sulle chiamate alle azioni API per TNB. AWS Contengono anche informazioni per le chiamate alle azioni API da servizi come Amazon EC2 e Amazon EBS.

## Registrazione delle chiamate AWS API Telco Network Builder utilizzando 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 chiamate API per AWS TNB come eventi. Le chiamate acquisite includono chiamate dalla console AWS TNB e chiamate in codice alle operazioni dell'API AWS TNB. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata a 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 del Centro identità IAM.
- 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 AWS Per 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 Console di gestione AWS sono multiregionali. È possibile creare un trail per una singola Regione o per più Regioni tramite AWS CLI. La creazione di un percorso multiregionale è consigliata in quanto consente di registrare l'intera attività del proprio Regioni AWS account. Se si crea un trail per una singola Regione, è possibile visualizzare solo gli eventi registrati nella Regione AWS del trail. Per ulteriori informazioni sui trail, consulta [Creating a trail for your Account AWS](#) e [Creating a trail for an organization](#) nella Guida per l'utente di AWS CloudTrail .

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 basate su SQL sui tuoi eventi. CloudTrail [Lake converte gli eventi esistenti in formato JSON basato su righe in formato Apache ORC](#). ORC è un formato di archiviazione a colonne 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 Prezzi.AWS CloudTrail](#)

## AWS Esempi di eventi TNB

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'operazione API 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 chiamate API pubbliche, quindi gli eventi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra un CloudTrail evento che dimostra l'CreateSolFunctionPackageoperazione.

```
{
  "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 Attività di implementazione TNB

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

La tabella seguente elenca le attività di implementazione di AWS TNB:

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 Amazon EKS.
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.
ClusterStorageClassesConfiguration	nessuna modifica del nome	Configura la classe di storage (driver CSI) su un cluster Amazon EKS.

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 risorse AWS TNB.
FunctionInstantiation	FunctionInstall	Implementa le funzioni di rete utilizzando HELM.
FunctionUninstallation	FunctionUninstall	Disinstalla la funzione di rete da un cluster Amazon EKS.
HookExecution	nessuna modifica del nome	Esegue gli hook del ciclo di vita come definito nell'NSD.
InfrastructureCancellation	nessuna modifica del nome	Annulla un servizio di rete.
InfrastructureInstantiation	nessuna modifica del nome	AWS Fornisce risorse per conto dell'utente.
InfrastructureTermination	nessuna modifica del nome	Deprovisioning delle AWS risorse richiamate tramite AWS TNB.
-	InfrastructureUpdate	Aggiorna le AWS risorse fornite per conto dell'utente.
InventoryDeregistration	nessuna modifica del nome	Annulla la registrazione AWS delle risorse da TNB. AWS
-	InventoryRegistration	Registra le risorse in TNB. AWS AWS
KubernetesClusterConfiguration	ClusterConfiguration	Configura il cluster Kubernetes e aggiunge ruoli IAM aggiuntivi ad Amazon EKS AuthMap come definito nell'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 Amazon EKS e il piano di controllo 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	Adatta e	Description
Operazioni di servizio di rete in corso e simultanee	Ogni regione supportata: 40	<a href="#">Sì</a>	Il numero massimo di operazioni di servizio di rete in corso e simultanee in una regione.
Pacchetti di funzioni	Ogni Regione supportata: 200	<a href="#">Sì</a>	Il numero massimo di pacchetti di funzioni in una regione.
Pacchetti di rete	Ogni regione supportata: 40	<a href="#">Sì</a>	Il numero massimo di pacchetti di rete in una regione.
Istanze di servizi di rete	Ogni regione supportata: 800	<a href="#">Sì</a>	Il numero massimo di istanze di servizi di rete in una regione.

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

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

Modifica	Descrizione	Data
<a href="#">Aggiornamenti alla configurazione di rete del gruppo di nodi Amazon EKS</a>	Aggiungi ed elimina sottoreti e gruppi di sicurezza. Aggiungere, modificare ed eliminare ENIs dalla rete. Per ulteriori informazioni, consulta <a href="#">Parametri che è possibile aggiornare</a> .	10 settembre 2025
<a href="#">Aggiunta ed eliminazione di gruppi di nodi Amazon EKS in cluster esistenti</a>	AWS TNB ora supporta l'aggiunta di nuovi gruppi di nodi e la rimozione di gruppi di nodi esistenti dai cluster Amazon EKS. Per ulteriori informazioni, consulta <a href="#">Parametri che è possibile aggiornare</a> .	4 giugno 2025
<a href="#">Dimensione del volume della radice</a>	Puoi specificare la dimensione e del volume root Amazon EBS sottostante dei tuoi nodi di lavoro Amazon EKS tramite il <code>root_volume_size</code> campo in <a href="#">AWS.Compute.EKSManagedNode</a> e <a href="#">.Compute.AWS EKSSelfManagedNode</a> Nodi TOSCA.	19 maggio 2025
<a href="#">Risorse di riferimento negli script</a>	<a href="#">Puoi fare riferimento alle risorse create da AWS TNB</a>	2 maggio 2025

	<a href="#">per configurarle negli script Lifecycle Hook e negli script dei dati utente.</a>	
<a href="#">La versione 1.32 di Kubernetes ora è supportata per i nodi Amazon EKS e i gruppi di nodi gestiti.</a>	<a href="#">AWS TNB supporta la versione 1.32 di Kubernetes per .compute.eks e .Compute.AWSAWS EKSMangedNodo.</a>	24 aprile 2025
<a href="#">La versione 1.24 di Kubernetes non è più supportata per i nodi Amazon EKS e i gruppi di nodi gestiti</a>	<a href="#">AWS TNB non supporta più la versione 1.24 di Kubernetes per .compute.eks e .Compute.AWSAWS EKSMangedNodo.</a>	17 aprile 2025
<a href="#">AL2023 Supporto AMI per nodi gestiti Amazon EKS</a>	<a href="#">AWS TNB supporta i tipi di AL2023 AMI per <u>AWS.Compute.EKSManaged</u>Nodo.</a>	17 aprile 2025
<a href="#">La versione 1.23 di Kubernetes non è più supportata per i nodi Amazon EKS e i gruppi di nodi gestiti</a>	<a href="#">AWS TNB non supporta più la versione 1.23 di Kubernetes per .compute.eks e .Compute.AWSAWS EKSMangedNodo.</a>	4 aprile 2025
<a href="#">L'ID AMI può essere aggiornato</a>	<a href="#">Ora puoi aggiornare il campo ami_id durante una chiamata UpdateSolNetworkService API.</a>	31 marzo 2025
<a href="#">La versione 1.31 di Kubernetes ora è supportata per i nodi Amazon EKS e i gruppi di nodi gestiti.</a>	<a href="#">AWS TNB supporta la versione 1.31 di Kubernetes per .compute.eks e .Compute.AWSAWS EKSMangedNodo.</a>	18 febbraio 2025

[Versione Kubernetes per .Compute. AWS EKSMangedNodo](#)

AWS TNB supporta le versioni di Kubernetes da 1.23 a 1.30 per creare un gruppo di nodi gestiti da Amazon EKS.

28 gennaio 2025

[Versione Kubernetes per cluster](#)

AWS TNB ora supporta la versione 1.30 di Kubernetes per creare cluster Amazon EKS.

19 agosto 2024

[AWS TNB supporta un'operazione aggiuntiva per gestire il ciclo di vita della rete.](#)

30 luglio 2024

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 Esempio di ruolo del servizio TNB:](#)
  - Aggiungi queste azioni Amazon EKS:eks:UpdateAddon ,eks:UpdateClusterVersion ,eks:UpdateNodegroupConfig ,eks:UpdateNodegroupVersion , eks:DescribeUpdate
  - Aggiungi questa CloudFormation azione: cloudformation:UpdateStack
- Nuove [attività di distribuzione](#): InfrastructureUpdate Inventory Registration , ValidateNetworkServiceUpdate
- Aggiornamenti delle API: [GetSolNetworkOperationListSolNetworkOper](#)

<a href="#">ations</a> , e <a href="#">UpdateSolNetworkInstance</a>		
<a href="#">Nuova attività e nuovi nomi di attività per attività esistenti</a>	È disponibile una nuova attività. A partire dal 7 marzo 2024, alcune attività esistenti hanno nuovi nomi per motivi di chiarezza.	7 maggio 2024
<a href="#">Versione Kubernetes per cluster</a>	AWS TNB ora supporta la versione 1.29 di Kubernetes per creare cluster Amazon EKS.	10 aprile 2024
<a href="#">Support per l'interfaccia di rete security_groups</a>	È possibile collegare gruppi di sicurezza al nodo AWS.Networking.eni.	2 aprile 2024
<a href="#">Support per la crittografia dei volumi root di Amazon EBS</a>	Puoi abilitare la crittografia Amazon EBS per il volume root di Amazon EBS. <a href="#">Per abilitarla, aggiungi le proprietà in AWS.Compute.EKSManagedNode o AWS.Compute.EKSSelfManagedNodenodo.</a>	2 aprile 2024
<a href="#">Support per node labels</a>	Puoi allegare etichette di nodi al tuo gruppo di nodi in <a href="#">AWS.Compute.EKSManagedNode o AWS.Compute.EKSSelfManagedNodenodo.</a>	19 marzo 2024
<a href="#">Support per l'interfaccia di rete source_dest_check</a>	È possibile indicare se si desidera abilitare o disabilitare il source/destination controllo dell'interfaccia di rete tramite il nodo AWS.Networking.eni.	25 gennaio 2024

<a href="#">Support per istanze Amazon EC2 con dati utente personalizzati</a>	Puoi avviare istanze Amazon EC2 con dati utente personalizzati tramite <code>.Compute.AWSUserData</code> nodo.	16 gennaio 2024
<a href="#">Support for Security Group</a>	AWS TNB consente di importare la AWS risorsa Security Group.	8 gennaio 2024
<a href="#">Descrizione aggiornata di <code>network_interfaces</code></a>	Quando la <code>network_interfaces</code> proprietà è inclusa in <a href="#">AWS.Compute.EKSManagedNode</a> o <a href="#">AWS.Compute.EKSSelfManagedNode</a> node, AWS TNB ottiene l'autorizzazione relativa ENIs dalla <code>multus_role</code> proprietà, se disponibile, o dalla proprietà <code>node_role</code>	18 dicembre 2023
<a href="#">Support per cluster privati</a>	AWS TNB ora supporta i cluster privati. Per indicare un cluster privato, imposta la <code>access</code> proprietà su <code>PRIVATE</code>	11 dicembre 2023
<a href="#">Versione Kubernetes per cluster</a>	AWS TNB ora supporta la versione 1.28 di Kubernetes per creare cluster Amazon EKS.	11 dicembre 2023
<a href="#">AWS TNB supporta il gruppo di collocamento</a>	È stato aggiunto un gruppo di posizionamento per <a href="#">AWS.Compute.EKSManagedNode</a> le definizioni dei <a href="#">AWS.Compute.EKSSelfManagedNode</a> nodi.	11 dicembre 2023

## [AWS TNB aggiunge il supporto per IPv6](#)

AWS TNB ora supporta la creazione di istanze di rete con infrastruttura. IPv6

[Controlla i nodi AWS.networking.vpc, .Networking.Subnet, .Networking.AWSAWS](#)  
<https://docs.aws.amazon.com/tnb/latest/ug/node-internet-gateway.html>[InternetGatewayAWS, .Rete. SecurityGroupIngressRule, AWS.Reti. SecurityGroupEgressRulee](#)  
[AWS.compute.eks](#) per le configurazioni. IPv6

[Abbiamo anche aggiunto i nodi .Networking.AWS NATGatewaye](#)  
[AWS.Networking.Route](#) per la configurazione. NAT64 Abbiamo aggiornato il ruolo del servizio AWS TNB e il ruolo del servizio AWS TNB per il gruppo di nodi Amazon EKS per le IPv6 autorizzazioni. Vedi esempi di policy [relative ai ruoli di servizio](#).

16 novembre 2023

## [Sono state aggiunte le autorizzazioni alla politica del ruolo di servizio AWS TNB](#)

Abbiamo aggiunto le autorizzazioni alla policy del ruolo del servizio AWS TNB per Amazon S3 CloudFormation e per abilitare l'istanziamento dell'infrastruttura.

23 ottobre 2023

<a href="#">AWS TNB è stato lanciato in più regioni</a>	AWS TNB è ora disponibile nelle regioni Asia Pacifico (Seoul), Canada (Centrale), Europa (Spagna), Europa (Stoccolma) e Sud America (San Paolo).	27 settembre 2023
<a href="#">Tag per.Compute. AWS EKSSelfManagedNode</a>	AWS TNB ora supporta i tag per la definizione del <code>AWS.Compute.EKSSelfManagedNode</code> nodo.	22 agosto 2023
<a href="#">AWS TNB supporta istanze che sfruttano IMDSv2</a>	All'avvio dell'istanza, è necessario utilizzare. IMDSv2	14 agosto 2023
<a href="#">Autorizzazioni aggiornate per MultusRoleInlinePolicy</a>	MultusRoleInlinePolicy Ora include <code>ec2:DeleteNetworkInterface</code> autorizzazione.	7 agosto 2023
<a href="#">Versione Kubernetes per cluster</a>	AWS TNB ora supporta le versioni 1.27 di Kubernetes per creare cluster Amazon EKS.	25 luglio 2023
<a href="#">AWS.Compute.eks. AuthRole</a>	AWS TNB supporta AuthRole che consente di aggiungere ruoli IAM al cluster Amazon EKS <code>aws-auth ConfigMap</code> in modo che gli utenti possano accedere al cluster Amazon EKS utilizzando un ruolo IAM.	19 luglio 2023

---

<a href="#">AWS TNB supporta i gruppi di sicurezza.</a>	Aggiunto l' <a href="#">AWS.Networking.SecurityGroup</a> , <a href="#">AWS.Networking.SecurityGroupEgressRule</a> e <a href="#">AWS.Networking.SecurityGroupIngressRule</a> al modello NSD.	18 luglio 2023
<a href="#">Versione Kubernetes per cluster</a>	AWS TNB supporta le versioni di Kubernetes da 1.22 a 1.26 per creare cluster Amazon EKS. AWS TNB non supporta più le versioni 1.21 di Kubernetes.	11 maggio 2023
<a href="#">AWS.Calcolo. EKSSelfManagedNode</a>	Puoi creare nodi di lavoro autogestiti su aree geografiche, AWS Local Zones e AWS Outposts	29 marzo 2023
<a href="#">Versione iniziale</a>	Questa è la prima versione della AWS TNB User Guide.	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.