



Guida per l'utente

# AWS Costruttore di reti di telecomunicazioni



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

Copyright © 2024 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 delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

# Table of Contents

Che cos'è ilAWS TNB? .....	1
Non hai mai usato AWS? .....	2
A chi siAWS rivolge TNB? .....	2
Perché usareAWS TNB? .....	2
Accedere aAWS TNB .....	4
Prezzi perAWS TNB .....	4
Che cosa è .....	5
Come funziona .....	6
Architettura .....	6
Integration .....	7
Quote .....	8
Concetti .....	9
Ciclo di vita di una funzione di rete .....	9
Usa interfacce standardizzate .....	10
Pacchetti NF .....	11
Descrizione del servizio NF .....	12
Gestione e operazioni .....	13
Descrittori dei servizi di rete .....	14
Configurazione .....	16
Registrazione ad AWS .....	16
Scelta di una Regione AWS .....	17
Annota l'endpoint del servizio .....	17
(Facoltativo) Installazione dell'AWS CLI .....	18
Creazione di un utente IAM .....	18
Configura i ruoli TNB AWS .....	19
Nozioni di base .....	20
Prerequisiti .....	20
Creazione di un pacchetto di funzioni .....	21
Creazione di un pacchetto di rete .....	21
Creare e istanziare un'istanza di rete .....	21
Elimina .....	22
Pacchetti di funzioni .....	23
Crea .....	21
Vista .....	24

Scarica un pacchetto .....	25
Eliminazione di un pacchetto .....	26
Pacchetti di rete .....	27
Crea .....	21
Vista .....	28
Scarica .....	29
Eliminazione .....	30
Rete .....	31
Istanziare .....	31
Vista .....	32
Aggiornamento .....	32
Termina ed elimina .....	33
Operazioni di rete .....	35
Vista .....	35
Annulla .....	36
Riferimento ai .....	37
Modello VNFD .....	37
Sintassi .....	37
Modello di topologia .....	37
AWS.VNF .....	38
AWS.Artifacts.Helm .....	39
Modello NSD .....	40
Sintassi .....	40
Utilizzo di parametri definiti .....	41
Importazione VNFD .....	41
Modello di topologia .....	42
AWS.NS .....	43
AWS.Compute.eks .....	44
AWS.Compute.eks. AuthRole .....	48
AWS.Compute.eks ManagedNode .....	49
AWS.Compute.eks SelfManagedNode .....	56
AWS.Calcola. PlacementGroup .....	62
AWS.Calcola. UserData .....	64
AWS.Rete. SecurityGroup .....	65
AWS.Rete. SecurityGroupEgressRule .....	67
AWS.Rete. SecurityGroupIngressRule .....	70

AWS.Risorsa. Importazione .....	73
AWS.Networking.eni .....	74
AWS.HookExecution .....	76
AWS.Rete. InternetGateway .....	77
AWS.Rete. RouteTable .....	80
AWS.Networking.Subnet .....	81
AWS.deployment.vnf Distribuzione .....	84
AWS.Networking.vpc .....	86
AWS.Networking.natGateway .....	88
AWS.Rete. Percorso .....	89
Nodi comuni .....	91
AWS.HookDefinition.Bash .....	91
Sicurezza .....	93
Protezione dei dati .....	94
Gestione dei dati .....	95
Crittografia a riposo .....	95
Crittografia in transito .....	95
Riservatezza del traffico Internet .....	95
Gestione dell'identità e degli accessi .....	95
Destinatari .....	96
Autenticazione con identità .....	96
Gestione dell'accesso con policy .....	100
Come funziona AWS Telco Network Builder con IAM .....	103
Esempi di policy basate su identità .....	109
Risoluzione dei problemi .....	124
Convalida della conformità .....	126
Resilienza .....	127
Sicurezza dell'infrastruttura .....	127
Modello di sicurezza della connettività di rete .....	129
Versione IMDS .....	129
Monitoraggio .....	130
CloudTrail registri .....	130
AWSInformazioni TNB in CloudTrail .....	130
Informazioni sulle voci dei file di registro diAWS TNB .....	131
Attività di distribuzione .....	133
Quote .....	135

---

Cronologia dei documenti .....	136
.....	cxli

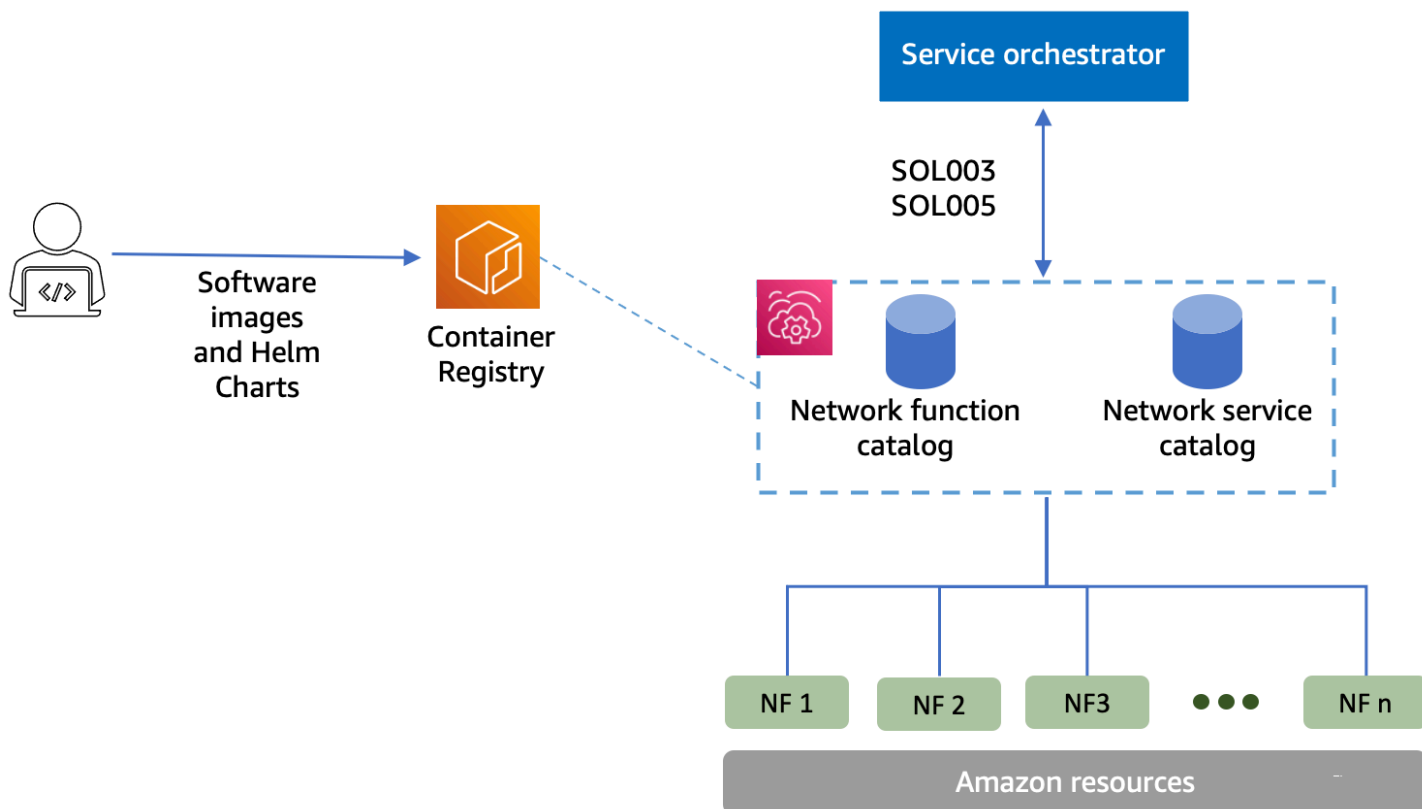
# Cos'è AWS Telco Network Builder?

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

Con AWS TNB, implementate reti 5G scalabili e sicure Cloud AWS utilizzando immagini software containerizzate in modo automatizzato. Non è necessario apprendere nuove tecnologie, decidere quale servizio di elaborazione utilizzare o sapere come eseguire il provisioning e la configurazione delle risorse AWS.

Invece, descrivi l'infrastruttura della tua rete e fornisci le immagini software delle funzioni di rete dei tuoi 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 della rete e degli accessi per creare un servizio di rete completamente operativo.

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



## Argomenti

- [Non hai mai usato AWS?](#)
- [A chi siAWS rivolge TNB?](#)
- [Perché usareAWS TNB?](#)
- [Accedere aAWS TNB](#)
- [Prezzi perAWS TNB](#)
- [Che cosa è](#)

## Non hai mai usato AWS?

Se non hai familiarità con i prodotti e i servizi AWS, consulta le risorse seguenti per informazioni di base.

- [Introduzione a AWS](#)
- [Nozioni di base suAWS](#)

## A chi siAWS rivolge TNB?

AWSTNB è destinato ai CSP che desiderano sfruttare l'efficienza in termini di costi, l'agilità e l'elasticità delleCloud AWS offerte senza scrivere e mantenere script e configurazioni personalizzati per progettare, distribuire e gestire i servizi di rete. AWS TNB fornisce automaticamente l'AWSinfrastruttura necessaria, implementa funzioni di rete containerizzate e configura la gestione della rete 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.

## Perché usareAWS TNB?

Di seguito sono riportati alcuni dei motivi per cui un CSP vorrebbe utilizzareAWS TNB:

Aiuta a semplificare le attività

Offri maggiore efficienza alle operazioni di rete, ad esempio l'implementazione di nuovi servizi, l'aggiornamento e l'aggiornamento delle funzioni di rete e la modifica delle topologie dell'infrastruttura di rete.



## Si integra con gli orchestratori

AWSTNB si integra con i più diffusi orchestratori di servizi di terze parti conformi allo standard ETSI.

## bilance

È possibile configurare AWS TNB per scalare 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

AWSTNB ti consente di ispezionare e monitorare le risorse che supportano la tua rete su un'unica dashboard, come Amazon VPC, Amazon EC2 e Amazon EKS.

## Supporta modelli di servizio

AWSTNB 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 integrarlo con una pipeline CI/CD (Continuous Integration and Continuous Delivery) per pubblicare una nuova definizione.

## Tiene traccia delle modifiche alle implementazioni di rete

Quando modifichi la configurazione sottostante dell'implementazione di una funzione di rete, ad esempio cambiando il tipo di istanza di un tipo di istanza Amazon EC2, puoi tenere traccia delle modifiche in modo ripetibile e scalabile. Farlo manualmente richiederebbe la gestione dello stato della rete, la creazione e l'eliminazione delle risorse e l'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 e tutte le versioni successive di una funzione di rete e specificare quando eseguire l'aggiornamento. Puoi anche gestire le tue applicazioni RAN, Core, IMS e di rete allo stesso modo.

## Accedere aAWS TNB

È possibile creare, accedere e gestire le risorseAWS TNB utilizzando una qualsiasi delle seguenti interfacce:

- AWSConsole TNB: fornisce un'interfaccia web per la gestione della rete.
- AWSAPI TNB: fornisce un'API RESTful per eseguire azioniAWS TNB. Per ulteriori informazioni, consulta [AWSTNB API Reference](#)
- AWS Command Line Interface(AWS CLI) — Fornisce comandi per un'ampia gamma diAWS servizi, inclusoAWS TNB. ed è supportata su Windows, macOS e Linux. Per ulteriori informazioni, consulta [AWS Command Line Interface](#).
- AWSSDK: forniscono API specifiche per ogni linguaggio e completa molti dettagli della connessione. Questi includono il calcolo delle firme e la gestione di errori e di nuovi tentativi di richiesta. Per ulteriori informazioni, consulta [SDK AWS](#).

## Prezzi perAWS TNB

AWSTNB aiuta i CSP ad automatizzare l'implementazione e la gestione delle proprie reti di telecomunicazioniAWS. Paghi per le seguenti due dimensioni quando usiAWS TNB:

- In base alle ore di funzionamento della rete gestita (MNFI).
- Per numero di richieste API.

Inoltre, l'utilizzo di altriAWS servizi in combinazione conAWS TNB comporta costi aggiuntivi. Per ulteriori informazioni, [AWSconsulta Prezzi](#) di.

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

Per domande su fatturazione, account ed eventi AWS, [contatta il supporto di AWS](#).

AWS Trusted Advisorè un servizio che puoi utilizzare per ottimizzare costi, sicurezza e prestazioni dell'AWSambiente. Per ulteriori informazioni, consulta [AWSTrusted Advisor](#).

# Che cosa è

Per ulteriori informazioni sulleAWS nozioni di base delle

- [Configurazione di AWS TNB](#)— Completa i passaggi prerequisiti.
- [Guida di base conAWS TNB](#)— Implementa la tua prima funzione di rete, come l'unità centralizzata (CU), la funzione di gestione degli accessi e della mobilità (AMF), la funzione User Plane (UPF) o un core 5G completo.

# Come funziona AWS TNB

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

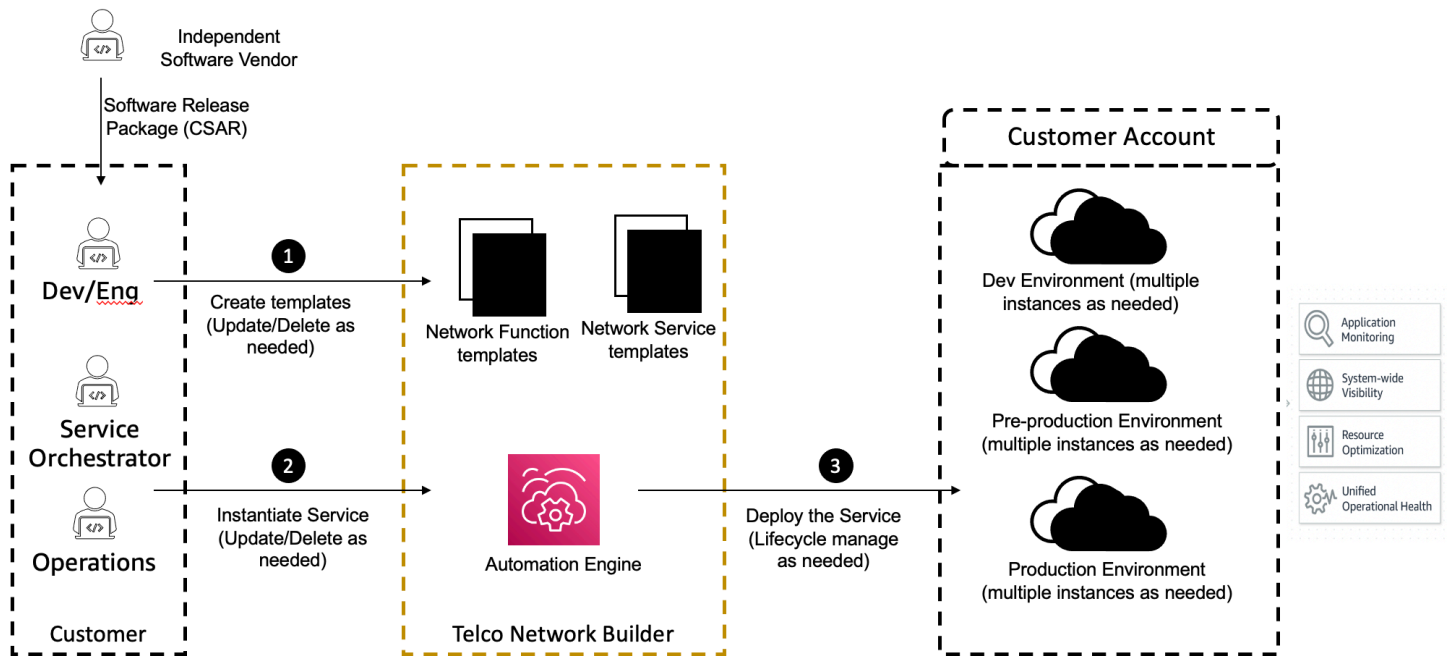
AWS TNB consente di inserire pacchetti di funzioni di rete e descrittori dei servizi di rete (NSD) e fornisce il motore di automazione per il funzionamento delle reti. Puoi utilizzare il tuo end-to-end orchestratore e integrarlo con le API AWS TNB oppure utilizzare gli SDK AWS TNB per creare il tuo flusso 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'AWS Management Console, API REST AWS TNB e gli SDK AWS CLI. Ciò consente ai diversi personaggi del CSP, come i membri dei team di ingegneria, operazioni e sistema programmatico, di sfruttare AWS TNB. Crei e carichi 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 i modelli per distribuire ripetutamente più configurazioni di quel pacchetto. Si creano modelli di servizi di rete che definiscono l'infrastruttura e le funzioni di rete che si desidera implementare. È possibile utilizzare le sostituzioni dei parametri per distribuire configurazioni diverse in posizioni diverse. È quindi possibile creare istanze di rete, utilizzando i modelli e distribuire le funzioni di rete sull'AWS infrastruttura. AWS TNB ti offre la visibilità delle tue implementazioni.



## Integrazione con Servizi AWS

Una rete 5G è costituita da un insieme di funzioni di rete containerizzate interconnesse distribuite su migliaia di cluster Kubernetes. AWS TNB si integra con i Servizi AWS seguenti API specifiche per le telecomunicazioni per creare un servizio di rete completamente operativo:

- Amazon Elastic Container Registry (Amazon ECR) per archiviare gli artefatti delle funzioni di rete dei fornitori di software indipendenti (ISV).
- Amazon Elastic Kubernetes Service (Amazon EKS) per configurare i cluster.
- Amazon VPC per strutture di rete.
- Gruppi di sicurezza che utilizzano AWS CloudFormation.
- AWS CodePipeline per obiettivi di distribuzione in Regioni AWS, AWS Local Zones e AWS Outposts.
- IAM per definire i ruoli.
- AWS Organizations per controllare l'accesso alle API AWS TNB.
- AWS Health Dashboard e AWS CloudTrail per monitorare le metriche sanitarie e postali.

## AWSQuote di risorse TNB

Il tuoAccount AWS dispone di quote di default, precedentemente definite limiti, per ciascuna di esseServizio AWS. Salvo dove diversamente specificato, ogni quota si applica a unRegione AWS. Se per alcune quote è possibile richiedere aumenti, non per tutte le quote.

Per visualizzare le quote perAWS TNB, aprire la [console Service Quotas](#). Nel riquadro di navigazione, selezionare Servizi AWS, quindi selezionare AWSTNB.

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

Il tuoAccount AWS dispone delle seguenti quote in relazione alAWS TNB.

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

# AWS Concetti TNB

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

## Indice

- [Ciclo di vita di una funzione di rete](#)
- [Usa interfacce standardizzate](#)
- [Pacchetti di funzioni di rete per TNB AWS](#)
- [Descrittori di servizio delle funzioni di rete per TNB AWS](#)
- [Gestione e operazioni per TNB AWS](#)
- [Descrittori dei servizi di rete per TNB AWS](#)

## Ciclo di vita di una funzione di rete

AWSTNB 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 elaborazione, 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. Inserisci NSD che definiscono 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.
- Sostituisci o disattiva la funzione di rete.

## Usa interfacce standardizzate

AWSTNB 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 gli SDK AWS TNB, la CLI o le API per avviare operazioni, come l'istanziamento o l'aggiornamento di una funzione di rete a una nuova versione.

AWSTNB supporta le seguenti specifiche.

Specifiche	Versione	Descrizione
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.



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

## Pacchetti di funzioni di rete per TNB AWS

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

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

Un pacchetto CSAR è un file zip conforme alla specifica YAML OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA).

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

```
tosca_definitions_version: tnb_simple_yaml_1_0

topology_template:

  node_templates:

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

```
    provider: "SampleNF"
  requirements:
    helm: HelmChart

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

## Descrittori di servizio delle funzioni di rete per TNB AWS

AWSTNB memorizza i descrittori dei servizi di rete (NSD) sulle funzioni di rete che si desidera implementare e su come implementarle nel catalogo. È possibile caricare il file YAML NSD, come descritto da ETSI SOL007, per includere:

- NF che desideri implementare
- Istruzioni di rete
- Istruzioni di calcolo
- Lifecycle Hooks (script personalizzati)

AWSTNB 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 a ETSI possa comprendere.

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: tosca.nodes.AWS.Compute.EKS
  properties:
    version: "1.27"
    access: "ALL"
    cluster_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleClusterRole"
  capabilities:
```

```
multus:
  properties:
    enabled: true
requirements:
  subnets:
    - Subnet01
    - Subnet02

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

## Gestione e operazioni per 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 le API AWS TNB 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.

- Gestione delle versioni dei pacchetti di funzioni di rete.
- Gestione delle versioni dei tuoi NSD.
- Recupero di informazioni sulle funzioni di rete distribuite.

## Descrittori dei servizi di rete per TNB AWS

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
```

# Configurazione di AWS TNB

Configura AWS TNB completando le attività descritte in questo argomento.

## Processi

- [Registrazione ad AWS](#)
- [Scelta di una Regione AWS](#)
- [Annota l'endpoint del servizio](#)
- [\(Facoltativo\) Installazione dell'AWS CLI](#)
- [Creazione di un utente IAM](#)
- [Configura i ruoli TNB AWS](#)

## Registrazione ad AWS

Quando ti iscrivi ad Amazon Web Services, ti iscrivi Account AWS automaticamente a tutti i servizi inAWS, incluso AWS TNB. Ti vengono addebitati solo i servizi che utilizzi.

Se disponi già di un Account AWS, passa all'operazione successiva. Se non disponi di un Account AWS, utilizza la seguente procedura per crearne uno.

Per creare un Account AWS

1. Aprire la pagina all'indirizzo <https://portal.aws.amazon.com/billing/signup>.
2. Seguire le istruzioni online.

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

Durante la registrazione di un Account AWS, viene creato un Utente root dell'account AWS. L'utente root ha accesso a tutte le risorse e i Servizi AWS in tale account. Come best practice di sicurezza, [assegnare l'accesso amministrativo a un utente amministrativo](#) e utilizzare solo l'utente root per eseguire [attività che richiedono l'accesso di un utente root](#).

## Scelta di una Regione AWS

Per visualizzare l'elenco delle regioni disponibili per AWS TNB, consulta l'elenco dei [servizi AWS regionali](#). Per visualizzare l'elenco degli endpoint per l'accesso programmatico, consulta gli endpoint [AWSTNB](#) nel. Riferimenti generali di AWS

## Annota l'endpoint del servizio

Per connettersi a livello di programmazione a un servizio AWS, è possibile utilizzare un endpoint. Oltre agli endpoint AWS standard, alcuni servizi AWS offrono endpoint FIPS in determinate Regioni. 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
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

Nome della regione	Regione	Endpoint	Protocollo
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) Installazione dell'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

## Creazione di un utente IAM

AWS Identity and Access Management (IAM) è un servizio Web che consente di controllare in modo sicuro l'accesso alle risorse AWS. Crea un ruolo utente IAM per utilizzare credenziali a breve termine per l'accesso. AWS

Per creare il ruolo, segui le istruzioni in Guida [introduttiva](#) nella Guida per l'AWS IAM Identity Center utente.



Puoi anche configurare l'accesso programmatico [configurando l'uso AWS IAM Identity Center nella AWS CLI Guida per l'AWS Command Line Interface](#) utente.

## Configura i ruoli TNB AWS

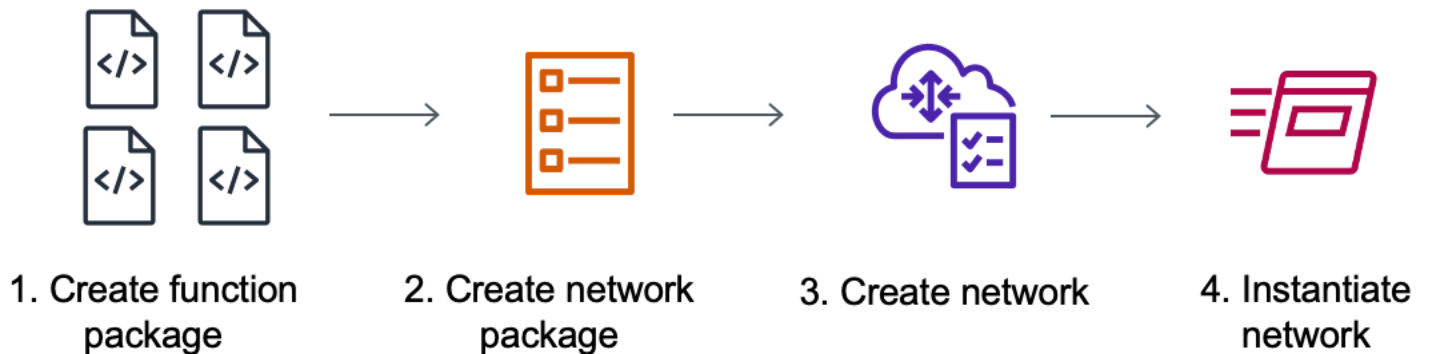
È necessario creare un ruolo di servizio IAM per gestire diverse parti della soluzione AWS TNB. AWS I ruoli di servizio TNB possono effettuare chiamate API ad altri AWS servizi, come AWS CloudFormationAWS CodeBuild, e vari servizi di elaborazione e archiviazione, per tuo conto, per creare istanze e gestire 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 di base conAWS TNB

Questo tutorial dimostra come utilizzareAWS TNB per implementare una funzione di rete, ad esempio la CU (unità centralizzata), la funzione di gestione degli accessi e della mobilità (AMF) o la funzione del piano utente 5G (UPF).

Il diagramma seguente illustra il processo di distribuzione:



## Processi

- [Prerequisiti](#)
- [Creazione di un pacchetto di funzioni](#)
- [Creazione di un pacchetto di rete](#)
- [Creare e istanziare un'istanza di rete](#)
- [Elimina](#)

## Prerequisiti

- È necessario disporre di un pianoAWS di Support aziendale.
- Configura le tue autorizzazioni utilizzando i ruoli IAM.
- Un [Package NF](#) conforme a ETSI SOL001/SOL004.
- [Modelli NSD](#) conformi a ETSI SOL007.

## Creazione di un pacchetto di funzioni

Per creare un pacchetto di funzioni

1. Apri la console AWS TNB all'[indirizzo https://console.aws.amazon.com/tnb/](https://console.aws.amazon.com/tnb/).
2. Nel riquadro di navigazione, seleziona Pacchetti di funzione.
3. Scegli Crea pacchetto di funzioni.
4. Scegli Scegli file e carica il tuo pacchetto CSAR.
5. Seleziona Successivo.
6. Controlla i dettagli del pacchetto, quindi scegli Crea pacchetto di funzioni.

## Creazione di un pacchetto di rete

Per creare un pacchetto di rete

1. Nel riquadro di navigazione, seleziona Pacchetti di rete.
2. Scegli Crea pacchetto di rete.
3. Scegli Scegli file e carica il tuo NSD.
4. Seleziona Successivo.
5. Scegli Crea pacchetto di rete.

## Creare e istanziare un'istanza di rete

Per creare e istanziare un'istanza di rete

1. Nel riquadro di navigazione, seleziona Reti.
2. Scegli Crea istanza di rete.
3. Immetti un nome e seleziona Avanti.
4. Selezionare il proprio NSD. Verifica i dettagli, quindi scegli Avanti.
5. Scegli Crea istanza di rete. Lo stato iniziale è Created.
6. Scegli l'ID dell'istanza di rete, quindi scegli Istanza.
7. Scegli Instantiate network.
8. Aggiorna per tenere traccia dello stato dell'istanza di rete.

# Elimina

Per eliminare le risorse

1. Nel riquadro di navigazione, seleziona Reti.
2. Scegliere l'ID di rete e quindi scegliere Elimina.
3. Quando viene richiesta la conferma, immettere l'ID di rete e quindi scegliere Elimina.
4. 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 TNB. Dopo aver caricato un pacchetto di funzioni, è necessario 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/](https://console.aws.amazon.com/tnb/).
2. Nel riquadro di navigazione, scegli Pacchetti di funzioni.
3. Scegliete Crea pacchetto di funzioni.
4. Scegli il file e carica il pacchetto CSAR di NF.
5. Seleziona 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. Ad 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/](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/](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/](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. Scegliere 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) che definisce i pacchetti di funzioni che desideri distribuire e l'AWS infrastruttura su cui desideri distribuirli.

## 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 eventuali file aggiuntivi (facoltativi), 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 determinati VPC, 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/](https://console.aws.amazon.com/tnb/).
2. Nel riquadro di navigazione, scegli Pacchetti di rete.
3. Scegli Crea pacchetto di rete.
4. Scegli il file e carica il tuo pacchetto CSAR.
5. Seleziona 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. Ad 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/](https://console.aws.amazon.com/tnb/).
2. Nel riquadro di navigazione, scegli Pacchetti di rete.
3. Usa la casella di ricerca per trovare il pacchetto di rete.

### AWS CLI

Per visualizzare un pacchetto di rete utilizzando il AWS CLI

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

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

- 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

- Apri la console AWS TNB all'[indirizzo https://console.aws.amazon.com/tnb/](https://console.aws.amazon.com/tnb/).
- Nel riquadro di navigazione, scegli Pacchetti di rete.
- Usa la casella di ricerca per trovare il pacchetto di rete
- Scegli il pacchetto di rete.
- Scegli Azioni, Scarica.

### AWS CLI

Per scaricare un pacchetto di rete utilizzando AWS CLI

- Utilizzare il comando [get-sol-network-package-content](#) per scaricare un pacchetto di rete.

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

# Eliminare un pacchetto di rete da AWS TNB

Scopri come eliminare un pacchetto di rete dal catalogo dei servizi di 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/](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. Scegliere Operazioni, Elimina.

## AWS CLI

Per eliminare un pacchetto di rete utilizzando il 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

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

## 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, è necessario crearne un'istanza. Quando si crea un'istanza di rete, AWS TNB implementa le funzioni di rete in base alle specifiche del descrittore del servizio di rete.

## Console

Per creare e istanziare un'istanza di rete utilizzando la console

1. [Apri la console AWS TNB all'indirizzo https://console.aws.amazon.com/tnb/](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 tuo NSD. Verifica i dettagli, quindi scegli Avanti.
6. Scegli Crea istanza di rete.
7. Scegli Instantiate.
8. Scegli Instantiate network.
9. Aggiorna per tenere traccia dello stato dell'istanza di rete.

## AWS CLI

Per creare e istanziare un'istanza di rete utilizzando AWS CLI

1. 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"
```

- Utilizzate 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}$
```

## Visualizza un'istanza di rete in TNB AWS

Scopri come visualizzare un'istanza di rete.

### Console

Per visualizzare un'istanza di rete utilizzando la console

- Apri la console AWS TNB all'[indirizzo https://console.aws.amazon.com/tnb/](https://console.aws.amazon.com/tnb/).
- Nel riquadro di navigazione, scegli Istanze di rete.
- Usa la casella di ricerca per trovare l'istanza di rete.

### AWS CLI

Per visualizzare un'istanza di rete utilizzando AWS CLI

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

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

- Usa il [get-sol-network-instance](#) comando per visualizzare i dettagli su un'istanza di rete.

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

## Aggiorna un'istanza di rete in AWS TNB

Scopri come aggiornare 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/](https://console.aws.amazon.com/tnb/).
2. Nel riquadro di navigazione, scegli Reti.
3. Seleziona l'ID dell'istanza di rete.
4. Nella scheda Funzioni, selezionare l'istanza della funzione da aggiornare.
5. Scegli Update (Aggiorna).
6. Inserisci le sostituzioni di aggiornamento per confermare l'aggiornamento.
7. Scegli Update (Aggiorna).
8. Aggiorna per tenere traccia dello stato dell'istanza di rete.

## AWS CLI

Usa la CLI per aggiornare un'istanza di rete

Usa il [update-sol-network-instance](#) comando per aggiornare 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 ...
```

## 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/](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 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 perAWS TNB

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

## Processi

- [Visualizzazione di un processo di rete](#)
- [Annullamento di un'operazione di rete](#)

## Visualizzazione di un processo di rete

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 mediante la console

1. Apri la consoleAWS TNB all'[indirizzo https://console.aws.amazon.com/tnb/](https://console.aws.amazon.com/tnb/).
2. Nel riquadro di navigazione, scegliere Istanze di rete.
3. Utilizza la casella di ricerca per trovare l'istanza di rete.
4. Nella scheda Distribuzioni, scegli Operazione di rete.

### AWS CLI

Per visualizzare un'operazione di rete utilizzando ilAWS CLI

1. Usa il [list-sol-network-operations](#) comando per elencare tutte le operazioni di rete.

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

2. Usa 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}$
```

# Annullamento di un'operazione di rete

Informazioni su come annullare un'operazione di rete.

## Console

Come annullare una utilizzando la console

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

## AWS CLI

Per annullare un'operazione di rete utilizzando AWS CLI

Utilizzate 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 perAWS TNB

La specifica di topologia e orchestrazione per applicazioni cloud (TOSCA) è una sintassi dichiarativa utilizzata dai CSP per descrivere una topologia dei servizi Web basati sul cloud, i loro componenti, le relazioni e i processi che li gestiscono. I CSP descrivono i punti di connessione, i collegamenti logici tra i punti di connessione e le politiche come affinità e sicurezza in un modello TOSCA. I CSP caricano quindi il modello suAWS TNB che sintetizza le risorse necessarie per stabilire una rete 5G funzionante nelle zone diAWS disponibilità.

## Indice

- [Modello VNFD](#)
- [Modello NSD](#)
- [Nodi comuni](#)

## Modello VNFD

Definisce un modello di Virtual Network Function Descriptor (VNFD).

## Sintassi

```
tosca_definitions_version: tnb_simple_yaml_1_0

topology_template:

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

  node\_templates:
    SampleNode1: tosca.nodes.AWS.VNF
```

## Modello di topologia

### node\_templates

La TOSCAAWSNodi. I nodi possibili sono:

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

## AWS.VNF

Definisce unAWSnodo di funzione di rete 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: String

provider

L'autore del VNFD.

Campo obbligatorio: sì

Tipo: String

## Requisiti

helm

La directory Helm che definisce gli artefatti del contenitore. Questo è un riferimento a [AWS.Artefatti.Elmo](#).

Campo obbligatorio: sì

Tipo: String

## 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 unAWSNodo Helm.

## Sintassi

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

```
properties:  
  implementation: String
```

## Proprietà

### implementation

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

Campo obbligatorio: sì

Tipo: String

## Esempio

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

## Modello NSD

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: String

## Modello di topologia

node\_templates

I possibili AWS nodi TOSCA sono:

- [AWS.NS](#)
- [AWS.Compute.eks](#)
- [AWS.Compute.eks. AuthRole](#)
- [AWS.Compute.eks ManagedNode](#)
- [AWS.Compute.eks SelfManagedNode](#)
- [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.deployment.vnf](#)



- [AWS.Networking.vpc](#)
- [AWS.Networking.NAT Gateway](#)
- [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: String

## 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 di Kubernetes di gestire le risorse necessarie per i tuoi NF. AWS 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: String

### **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 1.23 a 1.29.

Campo obbligatorio: sì

Tipo: String

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

### access

L'accesso agli endpoint del cluster.

Campo obbligatorio: sì

Tipo: String

Valori possibili: PRIVATE | PUBLIC | ALL

### cluster\_role

Il ruolo della gestione dei cluster.

Campo obbligatorio: sì

Tipo: String

### tags

Tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

ip\_family

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

Valore consentito: IPv4, IPv6

Valore predefinito: IPv4

Required: No

▪Tipo: stringa

## Requisiti

subnets

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

Campo obbligatorio: sì

Tipo: List

## Esempio

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

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

## AWS.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: String

`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.compute.eks ManagedNode

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, devi scegliere Amazon Machine Images (AMI) per i tuoi nodi di lavoro del cluster fornendo l'ID dell'AMI o il tipo di AMI. Fornisci anche una coppia di chiavi Amazon EC2 per l'accesso SSH e le proprietà di scalabilità per il tuo gruppo di nodi. Il tuo gruppo di nodi deve essere associato a un cluster EKS. È necessario fornire le sottoreti per i nodi di lavoro.

Facoltativamente, puoi allegare gruppi di sicurezza, etichette di nodi e un gruppo di posizionamento al tuo gruppo di nodi.

## Sintassi

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

## Funzionalità

### compute

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

#### ami\_type

Il tipo di AMI supportato da Amazon EKS.

Campo obbligatorio: sì

Tipo: String



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

### ami\_id

L'ID 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 chiavi EC2 per abilitare l'accesso SSH.

Campo obbligatorio: sì

Tipo: String

### 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.

## scaling

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

### `tags`

I tag da allegare alla risorsa.

Campo obbligatorio: no

Tipo: List

## Requisiti

### `cluster`

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

Campo obbligatorio: sì

Tipo: String

## 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 imposti network\\_interfaces, AWS TNB ottiene l'autorizzazione relativa agli ENI dalla multus\\_role proprietà se hai incluso la proprietà nel nodo AWS.Compute.eks. multus](#)  
[Altrimenti, AWS TNB ottiene l'autorizzazione relativa agli ENI dalla proprietà node\\_role.](#)

Campo obbligatorio: no

Tipo: List

## security\_groups

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

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 istanze Amazon EC2 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: tosa.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"
      scaling:
        properties:
          desired_size: 1
          min_size: 1
          max_size: 1
    properties:
      node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
      tags:
        - "Name=SampleVPC"
```

```

- "Environment=Testing"
requirements:
  cluster: SampleEKS
  subnets:
    - SampleSubnet
  network_interfaces:
    - SampleENI01
    - SampleENI02
  security_groups:
    - SampleSecurityGroup01
    - SampleSecurityGroup02
  placement_group: SamplePlacementGroup
  user_data: CustomUserData
  labels:
    - "sampleLabelName001=sampleLabelValue001"
    - "sampleLabelName002=sampleLabelValue002"

```

## AWS.Compute.eks SelfManagedNode

AWS TNB 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, devi scegliere Amazon Machine Images (AMI) per i tuoi nodi di lavoro del cluster fornendo l'ID dell'AMI. Facoltativamente, fornisci una coppia di chiavi Amazon EC2 per l'accesso SSH. È inoltre necessario fornire il tipo di istanza e le dimensioni desiderate, minime e massime. Il tuo gruppo di nodi deve essere associato a un cluster Amazon EKS. È necessario fornire le sottoreti per i nodi di lavoro.

Facoltativamente, puoi allegare gruppi di sicurezza, etichette di nodi e un gruppo di posizionamento al tuo gruppo di nodi.

### Sintassi

```

tosca.nodes.AWS.Compute.EKSSelfManagedNode:
  capabilities:
    compute:
      properties:
        ami\_id: String
        instance\_type: String
        key\_pair: String
        root\_volume\_encryption: Boolean
        root\_volume\_encryption\_key\_arn: String
    scaling:

```

```
properties:
  desired\_size: Integer
  min\_size: Integer
  max\_size: Integer
properties:
  node\_role: String
  tags: List
requirements:
  cluster: String
  subnets: List
  network\_interfaces: List
  security\_groups: List
  placement\_group: String
  user\_data: String
  labels: List
```

## Funzionalità

### ***compute***

Proprietà che definiscono i parametri di calcolo per i nodi autogestiti di Amazon EKS, ad esempio i tipi di istanze Amazon EC2 e le AMI di istanze Amazon EC2.

#### `ami_id`

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

Campo obbligatorio: sì

Tipo: String

#### `instance_type`

La dimensione dell'istanza.

Campo obbligatorio: sì

Tipo: String

#### `key_pair`

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

Campo obbligatorio: sì

Tipo: String

`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.



## ***scaling***

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

### `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: String

### 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 imposti network\\_interfaces, AWS TNB ottiene l'autorizzazione relativa agli ENI dalla multus\\_role proprietà se hai incluso la proprietà nel nodo AWS.Compute.eks. multus](#)  
[Altrimenti, AWS TNB ottiene l'autorizzazione relativa agli ENI dalla proprietà node\\_role.](#)

Campo obbligatorio: no

Tipo: List

### security\_groups

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

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 istanze Amazon EC2 lanciate dal gruppo di nodi autogestito. Aggiungi le autorizzazioni necessarie per l'esecuzione di dati utente personalizzati al `node_role` passato al gruppo di nodi.

Required: No

▪Tipo: stringa

labels

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

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

Ad esempio, `myLabelName1=*NodeLabelValue1`

Campo obbligatorio: no

Tipo: List

## Esempio

```
SampleEKSSelfManagedNode:
  type: tosca.nodes.AWS.Compute.EKSSelfManagedNode
  capabilities:
    compute:
      properties:
        ami_id: "ami-123123EXAMPLE"
        instance_type: "c5.large"
        key_pair: "SampleKeyPair"
        root_volume_encryption: true
        root_volume_encryption_key_arn: "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
```

```
scaling:
  properties:
    desired_size: 1
    min_size: 1
    max_size: 1
properties:
  node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleNodeRole"
  tags:
    - "Name=SampleVPC"
    - "Environment=Testing"
requirements:
  cluster: SampleEKSCluster
  subnets:
    - SampleSubnet
  network_interfaces:
    - SampleNetworkInterface01
    - SampleNetworkInterface02
  security_groups:
    - SampleSecurityGroup01
    - SampleSecurityGroup02
  placement_group: SamplePlacementGroup
  user_data: CustomUserData
  labels:
    - "sampleLabelName001=sampleLabelValue001"
    - "sampleLabelName002=sampleLabelValue002"
```

## AWS.Calcola. PlacementGroup

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

Quando avvii una nuova istanza Amazon EC2, il servizio Amazon EC2 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 istanze Amazon EC2.

Campo obbligatorio: sì

Tipo: String

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

## AWS.Calcola. UserData

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

Durante l'istanza di rete, AWS TNB fornisce la registrazione dell'istanza Amazon EC2 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 [multimime](#) ad Amazon EC2. Lo script personalizzato per i dati utente viene eseguito prima dello script di registrazione di 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 AWS CloudFormation per elaborare e rendere lo script multimime dei dati utente, assicurati che lo script rispetti tutte le regole. AWS CloudFormation

## Sintassi

```
toska.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: String

### content\_type

Tipo di contenuto dello script di dati utente.

Campo obbligatorio: sì

Tipo: String

Valori possibili: `x-shellscript`

## Esempio

```
ExampleUserData:
  type: toasca.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 dei gruppi di [sicurezza Amazon EC2 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: String

### 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: String

### 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: String



## Esempio

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

## AWS.Rete. SecurityGroupEgressRule

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

Required: No

▪Tipo: stringa

`cidr_ipv6`

L'intervallo di indirizzi IPv6 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 numero del tipo. Il valore -1 indica tutti i tipi ICMP/ICMPv6. Se specifichi tutti i tipi ICMP/ICMPv6, devi specificare tutti i codici ICMP/ICMPv6.

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: String

## 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 i codici ICMP/ICMPv6. Se specifichi tutti i tipi ICMP/ICMPv6, devi specificare tutti i codici ICMP/ICMPv6.

Campo obbligatorio: no

Tipo: integer

## Requisiti

### security\_group

L'ID del gruppo di sicurezza a cui deve essere aggiunta questa regola.

Campo obbligatorio: sì

Tipo: String

### 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: toasca.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 regole di ingresso dei gruppi di sicurezza di Amazon EC2 che possono essere collegate 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 IPv4 in formato CIDR. È necessario specificare un intervallo CIDR che consenta il traffico in ingresso.

Required: No

▀Tipo: stringa

### `cidr_ipv6`

L'intervallo di indirizzi IPv6 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 numero del tipo. Il valore -1 indica tutti i tipi ICMP/ICMPv6. Se specifichi tutti i tipi ICMP/ICMPv6, devi specificare tutti i codici ICMP/ICMPv6.

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: String

`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 i codici ICMP/ICMPv6. Se specifichi tutti i tipi ICMP/ICMPv6, devi specificare tutti i codici ICMP/ICMPv6.

Campo obbligatorio: no

Tipo: integer

## Requisiti

`security_group`

L'ID del gruppo di sicurezza a cui deve essere aggiunta questa regola.

Campo obbligatorio: sì

Tipo: String

`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 risorse in TNB: AWS AWS

- 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: tosca.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'istanza Amazon EC2 in una sottorete, puoi collegare un'interfaccia di rete ad essa oppure scollegare un'interfaccia di rete da quell'istanza Amazon EC2 e ricollegarla a un'altra 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: String

### 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: String

### vpc

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

Campo obbligatorio: sì

Tipo: String

## 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 CIDR IPv4 utilizzato per la corrispondenza di destinazione. Questa proprietà viene utilizzata per creare un percorso in `RouteTable` e il relativo valore viene utilizzato come `DestinationCidrBlock`.

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

▪Tipo: stringa

#### `ipv6_dest_cidr`

Il blocco CIDR IPv6 utilizzato per la corrispondenza 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 specifica per IPv6. 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: String

### route\_table

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

Campo obbligatorio: sì

Tipo: String

## 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: toska.nodes.AWS.Networking.InternetGateway
  properties:
    egress_only: true
  capabilities:
    routing:
      properties:
        ipv6_dest_cidr: ":::/0"
  requirements:
    route_table: Free5GCPrivateRouteTable
    vpc: Free5GCVPC
```

## AWS.Rete. RouteTable

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

### Sintassi

```
toska.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: String

## Esempio

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

## AWS.Networking.Subnet

Una sottorete è un intervallo di indirizzi IP 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 IPv4 pubblico.

Campo obbligatorio: sì

Tipo: String

Valori possibili: PUBLIC | PRIVATE

`availability_zone`

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

Campo obbligatorio: sì

Tipo: String

`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 CIDR IPv6 per la sottorete creata su Amazon VPC.

Utilizza il seguente formato: *2-digit hexadecimal* `::/subnetMask`

Se includi questa proprietà, non `ipv6_cidr_block` includerla.

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: String

### route\_table

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

Campo obbligatorio: sì

Tipo: String

## Esempio

```
SampleSubnet01:
```

```

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

```

```

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

```

## AWS.deployment.vnf

Le implementazioni NF sono modellate fornendo l'infrastruttura e l'applicazione ad esse associate. L'attributo [cluster](#) specifica il cluster EKS per ospitare i tuoi NF. 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

```

tosca.nodes.AWS.Deployment.VNFDeployment:
  requirements:
    deployment: String
    cluster: String
    vnfs: List
  interfaces:
    Hook:

```

```
pre_create: String
post_create: String
```

## Requisiti

### deployment

[Un nodo .deployment.vnfDeployment.AWS](#)

Required: No

▪Tipo: stringa

### cluster

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

Campo obbligatorio: sì

Tipo: String

### vnfs

[Un AWS nodo .VNF.](#)

Campo obbligatorio: sì

Tipo: String

## 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 rete IPv4 per il VPC, in notazione CIDR.

Campo obbligatorio: sì

Tipo: String

ipv6\_cidr\_block

Il blocco CIDR IPv6 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: toska.nodes.AWS.Networking.VPC
  properties:
    cidr_block: "10.100.0.0/16"
    ipv6_cidr_block: "AMAZON_PROVIDED"
    dns_support: true
  tags:
    - "Name=SampleVPC"
    - "Environment=Testing"
```

## AWS.Networking.NATGateway

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: String

#### internet\_gateway

Il [AWS file .Networking.InternetGateway](#) riferimento al nodo.

Campo obbligatorio: sì

Tipo: String

### Proprietà

#### type

Indica se il gateway è pubblico o privato.

Valore consentito: PUBLIC, PRIVATE

Campo obbligatorio: sì

Tipo: String

`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

```
tosca.nodes.AWS.Networking.Route:
  properties:
    dest\_cidr\_blocks: List
  requirements:
```

```
nat\_gateway: String  
route\_table: String
```

## Proprietà

### dest\_cidr\_blocks

L'elenco delle rotte IPv4 di destinazione verso la risorsa di destinazione.

Campo obbligatorio: sì

Tipo: List

Tipo di membro: String

## Proprietà

### nat\_gateway

Il riferimento al [AWS nodo .networking.natGateway](#).

Campo obbligatorio: sì

Tipo: String

### route\_table

Il [AWS file .Networking.RouteTable](#) riferimento al nodo.

Campo obbligatorio: sì

Tipo: String

## Esempio

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



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

## Nodi comuni

Definisce i nodi da utilizzare in NSD e VNFD.

- [SEGHE.HookDefinition.Colpo](#)

## AWS.HookDefinition.Bash

Definisce unAWS HookDefinitionnelbash.

### Sintassi

```
tosca.nodes.AWS.HookDefinition.Bash:
  properties:
    implementation: String
    environment\_variables: List
    execution\_role: String
```

### Proprietà

#### implementation

Il percorso relativo alla definizione del gancio. Il formato deve essere: ./  
hooks/*script\_name*.sh

Campo obbligatorio: sì

Tipo: String

#### environment\_variables

Le variabili di ambiente per lo script hook bash. Utilizza il seguente

formato:**envName=envValue**con la seguente espressione regolare: $^{[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+\$}$

Assicurarsi che**envName=envValue**il valore soddisfa i seguenti criteri:

- Non utilizzare spazi.

- Inizio **envName** con una lettera (A-Z o a-z) o un numero (0-9).
- Non avviare il nome della variabile di ambiente con quanto segue **AWSP** parole chiave riservate TNB (maiuscole e minuscole):
  - COMPILAZIONE DEL CODICE
  - TNB
  - CASA
  - AWS
- Puoi usare un numero qualsiasi di lettere (A-Z o a-z), numeri (0-9) e caratteri speciali - e \_ per **envName** e **envValue**.

Esempio: A123-45xYz=Example\_789

Campo obbligatorio: no

Tipo: List

`execution_role`

Il ruolo dell'esecuzione degli hook.

Campo obbligatorio: sì

Tipo: String

## Esempio

```
SampleHookScript:
  type: tosa.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 Telco Network Builder

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

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

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

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa 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. Questi contenuti includono la configurazione della protezione e le attività di gestione per i servizi Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [Modello di responsabilità condivisa AWS e GDPR](#) nel Blog sulla sicurezza AWS .

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. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- 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 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con AWS TNB o altri utenti Servizi AWS utilizzando la console, l'API o gli SDK. AWS CLI AWS 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 fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua 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 interrompe inoltre le reti ed elimina i pacchetti di funzioni e i pacchetti di rete.

## Crittografia 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 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 Internet

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 software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS 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 con policy](#)
- [Come funziona AWS Telco Network Builder 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 a seconda del lavoro svolto in AWS TNB.

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

Amministratore del servizio: se sei responsabile delle risorse AWS TNB della tua azienda, probabilmente hai pieno accesso a AWS TNB. È tuo compito determinare a quali funzionalità e risorse AWS TNB devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con AWS TNB, consulta [Come funziona AWS Telco Network Builder con IAM](#)

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere politiche per gestire l'accesso a AWS TNB. Per visualizzare esempi di policy basate sull'identità AWS TNB che puoi utilizzare in IAM, 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 essere autenticato (aver effettuato l'accesso root dell'account AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center (precedentemente AWS Single Sign-On), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

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

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [Signing AWS API request](#) nella IAM User Guide.

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

## Account AWS utente root

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

## Identità federata

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

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

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di

utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni sul Centro identità IAM, consulta [Cos'è Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

## Utenti e gruppi IAM

Un [utente IAM](#) è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente di IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato Amministratori IAM e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Quando creare un utente IAM \(invece di un ruolo\)](#) nella Guida per l'utente di IAM.

## Ruoli IAM

Un [ruolo IAM](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console [cambiando ruolo](#). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene



autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Creazione di un ruolo per un provider di identità di terza parte](#) nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .

- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.
- **Accesso a più servizi:** alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
- **Sessioni di accesso diretto (FAS):** quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).
- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire azioni 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 [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di IAM.
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli

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

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta [Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2](#) nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente di IAM.

## Gestione dell'accesso con policy

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

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

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

Le policy IAM definiscono le autorizzazioni relative a un'azione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'AWS CLI o dall' AWS API.

## Policy basate su identità

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

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente di IAM.

## Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

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

## Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta [Panoramica delle liste di controllo degli accessi \(ACL\)](#) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

## Altri tipi di policy

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

- **Limiti delle autorizzazioni:** un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- **Politiche di controllo dei servizi (SCP):** le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- **Policy di sessione:** le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente di IAM.

## Più tipi di policy

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

## Come funziona AWS Telco Network Builder 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 su identità</a>	Sì
<a href="#">Policy basate su risorse</a>	No
<a href="#">Azioni di policy</a>	Sì
<a href="#">Risorse relative alle policy</a>	Sì
<a href="#">Chiavi di condizione delle policy</a>	Sì
<a href="#">Liste di controllo degli accessi (ACL)</a>	No
<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 su identità	Sì
---------------------------------------	----

Le policy basate su 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 azioni che utenti e ruoli

possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Con le policy basate su 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. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per 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 di 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. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

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

## Azioni politiche per AWS TNB

Supporta le azioni di policy

Sì

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

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

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

Per visualizzare un elenco di 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 di policy Sì

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

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

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

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

Per visualizzare un elenco dei tipi di risorse AWS TNB e dei relativi ARN, consulta [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 Sì  
specifiche del servizio

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

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



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

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente di IAM.

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

## ACL in TNB AWS

Supporta le ACL	No
-----------------	----

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

## ABAC con TNB AWS

Supporta ABAC (tag nelle policy)	Sì
----------------------------------	----

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag

a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

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

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

Per ulteriori informazioni su ABAC, consulta [Che cos'è 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 AWS TNB

Supporta le credenziali temporanee	Sì
------------------------------------	----

Alcune Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla [Servizi AWS compatibilità con IAM nella IAM User Guide](#).

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta [Cambio di un ruolo \(console\)](#) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API o AWS CLI. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza provvisorie in IAM](#).

## Autorizzazioni principali multiservizio per TNB AWS

Supporta sessioni di accesso diretto (FAS)      Sì

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

## Ruoli di servizio per AWS TNB

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 [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di 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'azione 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 ai servizi, ma non modificarle.

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

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse TNB. AWS. Inoltre, non possono eseguire attività utilizzando AWS Management

Console, AWS Command Line Interface (AWS CLI) o l' AWS API. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

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

Per i dettagli sulle azioni e sui tipi di risorse definiti da AWS TNB, incluso il formato degli ARN 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 le risorse TNB nel tuo account. AWS Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo 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 puoi aggiungere una condizione alle tue 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 AWS CloudFormation. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.

- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per IAM Access Analyzer](#) 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, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta [Configurazione dell'accesso alle API protetto 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, concedi l'accesso solo alle operazioni 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 AWS 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:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:GetCallerIdentity"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AssumeRole"
    },
    {
      "Action": [
        "tnb:*"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "TNBPolicy"
    },
    {
      "Action": [
        "iam:AddRoleToInstanceProfile",
```

```

        "iam:CreateInstanceProfile",
        "iam>DeleteInstanceProfile",
        "iam:GetInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:TagInstanceProfile",
        "iam:UntagInstanceProfile"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "IAMPolicy"
},
{
    "Condition": {
        "StringEquals": {
            "iam:AWSServiceName": [
                "eks.amazonaws.com",
                "eks-nodegroup.amazonaws.com"
            ]
        }
    },
    "Action": [
        "iam:CreateServiceLinkedRole"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "TNBAccessSLRPermissions"
},
{
    "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:CreateOrUpdateTags",
        "autoscaling>DeleteAutoScalingGroup",
        "autoscaling>DeleteTags",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DescribeTags",
        "autoscaling:UpdateAutoScalingGroup",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreateSecurityGroup",
        "ec2>DeleteLaunchTemplateVersions",

```

```
"ec2:DescribeLaunchTemplates",
"ec2:DescribeLaunchTemplateVersions",
"ec2>DeleteLaunchTemplate",
"ec2>DeleteSecurityGroup",
"ec2:DescribeSecurityGroups",
"ec2:DescribeTags",
"ec2:GetLaunchTemplateData",
"ec2:RevokeSecurityGroupEgress",
"ec2:RevokeSecurityGroupIngress",
"ec2:RunInstances",
"ec2:AssociateRouteTable",
"ec2:AttachInternetGateway",
"ec2>CreateInternetGateway",
"ec2>CreateNetworkInterface",
"ec2>CreateRoute",
"ec2>CreateRouteTable",
"ec2>CreateSubnet",
"ec2:CreateTags",
"ec2>CreateVpc",
"ec2>DeleteInternetGateway",
"ec2>DeleteNetworkInterface",
"ec2>DeleteRoute",
"ec2>DeleteRouteTable",
"ec2>DeleteSubnet",
"ec2>DeleteTags",
"ec2>DeleteVpc",
"ec2:DetachNetworkInterface",
"ec2:DescribeInstances",
"ec2:DescribeInternetGateways",
"ec2:DescribeKeyPairs",
"ec2:DescribeNetworkInterfaces",
"ec2:DescribeRouteTables",
"ec2:DescribeSecurityGroupRules",
"ec2:DescribeSubnets",
"ec2:DescribeVpcs",
"ec2:DetachInternetGateway",
"ec2:DisassociateRouteTable",
"ec2:ModifySecurityGroupRules",
"ec2:ModifySubnetAttribute",
"ec2:ModifyVpcAttribute",
"ec2:AllocateAddress",
"ec2:AssignIpv6Addresses",
"ec2:AssociateAddress",
"ec2:AssociateNatGatewayAddress",
```



```

        "ec2:AssociateVpcCidrBlock",
        "ec2:CreateEgressOnlyInternetGateway",
        "ec2:CreateNatGateway",
        "ec2>DeleteEgressOnlyInternetGateway",
        "ec2>DeleteNatGateway",
        "ec2:DescribeAddresses",
        "ec2:DescribeEgressOnlyInternetGateways",
        "ec2:DescribeNatGateways",
        "ec2:DisassociateAddress",
        "ec2:DisassociateNatGatewayAddress",
        "ec2:DisassociateVpcCidrBlock",
        "ec2:ReleaseAddress",
        "ec2:UnassignIpv6Addresses",
        "ec2:DescribeImages",
        "eks:CreateCluster",
        "eks:ListClusters",
        "eks:RegisterCluster",
        "eks:TagResource",
        "eks:DescribeAddonVersions",
        "events:DescribeRule",
        "iam:GetRole",
        "iam:ListAttachedRolePolicies",
        "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "TNBAccessComputePerms"
},
{
    "Action": [
        "codebuild:BatchDeleteBuilds",
        "codebuild:BatchGetBuilds",
        "codebuild:CreateProject",
        "codebuild>DeleteProject",
        "codebuild>ListBuildsForProject",
        "codebuild:StartBuild",
        "codebuild:StopBuild",
        "events>DeleteRule",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "s3:CreateBucket",
        "s3:GetBucketAcl",
        "s3:GetObject",
    ]
}

```

```

        "eks:DescribeNodegroup",
        "eks>DeleteNodegroup",
        "eks:AssociateIdentityProviderConfig",
        "eks:CreateNodegroup",
        "eks>DeleteCluster",
        "eks:DeregisterCluster",
        "eks:UntagResource",
        "eks:DescribeCluster",
        "eks:ListNodegroups",
        "eks:CreateAddon",
        "eks>DeleteAddon",
        "eks:DescribeAddon",
        "eks:DescribeAddonVersions",
        "s3:PutObject",
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStacks",
        "cloudformation:UpdateTerminationProtection"
    ],
    "Resource": [
        "arn:aws:events:*:*:rule/tnb*",
        "arn:aws:codebuild:*:*:project/tnb*",
        "arn:aws:logs:*:*:log-group:/aws/tnb*",
        "arn:aws:s3:::tnb*",
        "arn:aws:eks:*:*:addon/tnb*/**/*",
        "arn:aws:eks:*:*:cluster/tnb*",
        "arn:aws:eks:*:*:nodegroup/tnb*/tnb*/**",
        "arn:aws:cloudformation:*:*:stack/tnb*"
    ],
    "Effect": "Allow",
    "Sid": "TNBAccessInfraResourcePerms"
},
{
    "Sid": "CFNTemplatePerms",
    "Effect": "Allow",
    "Action": [
        "cloudformation:GetTemplateSummary"
    ],
    "Resource": "*"
},
{
    "Sid": "ImageAMISSMPerms",
    "Effect": "Allow",

```

```

    "Action": [
      "ssm:GetParameters"
    ],
    "Resource": [
      "arn:aws:ssm::*:parameter/aws/service/eks/optimized-ami/*",
      "arn:aws:ssm::*:parameter/aws/service/bottlerocket/*"
    ]
  },
  {
    "Action": [
      "tag:GetResources"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "TaggingPolicy"
  },
  {
    "Action": [
      "outposts:GetOutpost"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "OutpostPolicy"
  }
]
}

```

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

```

{
  "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 AWS 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 AWS 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 AWS CloudFormation modello che crea un ruolo di servizio AWS TNB per una policy Life-Cycle Hook.

```
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:ListGroupsWithUser",
        "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 disponi dell'autorizzazione per eseguire un'operazione, le tue policy devono essere aggiornate in modo che ti sei consentito eseguire tale operazione.

L'errore di esempio seguente si verifica quando l'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'`iam:PassRole` azione, le tue politiche devono essere aggiornate per consentirti di assegnare un ruolo a AWS TNB.

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

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 passare il ruolo al servizio.

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

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

Se hai bisogno di 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 servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS TNB supporta queste funzionalità, consulta. [Come funziona AWS Telco Network Builder 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 di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente 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. AWS fornisce le seguenti risorse per contribuire alla conformità:

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

### Note

Non Servizi AWS tutte sono idonee all'HIPAA. Per ulteriori informazioni, consulta la sezione [Riferimenti sui servizi conformi ai requisiti HIPAA](#).

- [AWS Risorse per la conformità](#): questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.

- [AWS Guide alla conformità dei clienti](#): comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- [Valutazione delle risorse con regole](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- [AWS Security Hub](#)— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina [Documentazione di riferimento sui controlli della Centrale di sicurezza](#).
- [AWS Audit Manager](#)— Ciò Servizio AWS consente di verificare continuamente AWS l'utilizzo per semplificare la gestione dei rischi e la conformità alle normative e agli standard di settore.

## Resilienza in AWS TNB

L'infrastruttura AWS globale è costruita attorno Regioni AWS a 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, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

[Per ulteriori informazioni sulle zone di disponibilità, vedere Global Regioni AWS Infrastructure.AWS](#)

AWS 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 globale. AWS 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.

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

Ecco alcuni esempi di responsabilità condivise:

- AWS è responsabile della protezione dei componenti che supportano AWS TNB, tra cui:
  - Istanze di calcolo (note anche come worker)
  - Database interni
  - Comunicazioni di rete tra componenti interni
  - L'interfaccia di programmazione 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 all'interno dei VPC 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 clienti 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 AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDK sono crittografate.

## Versione IMDS

AWS TNB supporta istanze che utilizzano Instance Metadata Service versione 2 (IMDSv2), un metodo orientato alla sessione. IMDSv2 include una sicurezza maggiore rispetto a IMDSv1. 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 avvii l'istanza, devi usare IMDSv2. Per ulteriori informazioni su IMDSv2, consulta [Utilizzare IMDSv2](#) nella Guida per l'utente di Amazon EC2 per istanze Linux.

# MonitoraggioAWSTNB

Il monitoraggio è una parte importante del mantenimento dell'affidabilità, della disponibilità e delle prestazioni diAWSTNB e altriAWSsoluzioni.AWSfornisceAWS CloudTrailguardareAWSTNB, segnala quando qualcosa non va e intraprendi azioni automatiche quando necessario.

UsaCloudTrailper acquisire informazioni dettagliate sulle chiamate effettuate aAWSAPI. Puoi archiviare queste chiamate come file di registro in Amazon S3. Puoi usare questiCloudTrailregistri 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.

LaCloudTraili log contengono informazioni sulle chiamate alle azioni API perAWSTNB. Contengono inoltre informazioni per le chiamate alle azioni API da servizi come Amazon EC2 e Amazon EBS.

## Registrazione delle chiamate APIAWS Telco Network Builder utilizzandoAWS CloudTrail

AWSTelco Network Builder è integrato conAWS CloudTrail, un servizio che offre un registro delle operazioni eseguite da un utente, un ruolo o unAWS servizio inAWS TNB. CloudTrail acquisisce tutte le chiamate API perAWS TNB come eventi. Le chiamate acquisite includono le chiamate dalla consoleAWS TNB e le chiamate di codice alle operazioni delle APIAWS TNB. Se si crea un trail, è possibile abilitare la distribuzione continua di CloudTrail eventi in un bucket Amazon S3, inclusi gli eventi perAWS TNB. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console in Cronologia degli eventi. Le informazioni raccolte da permettono CloudTrail di determinare la richiesta effettuata aAWS TNB, l'indirizzo IP da cui è stata effettuata la richiesta, l'autore della richiesta, il momento in cui è stata effettuata e altri dettagli.

Per ulteriori informazioni CloudTrail, consulta la [Guida perAWS CloudTrail l'utente](#).

## AWSInformazioni TNB in CloudTrail

CloudTrail è abilitato sul tuo alAccount AWS momento della sua creazione. Quando si verifica un'attività inAWS TNB, questa viene registrata in un CloudTrail evento insieme ad altri eventi diAWS servizio nella cronologia degli eventi di servizio. Puoi visualizzare, cercare e scaricare gli eventi recenti nell'Account AWS. Per ulteriori informazioni, consulta [Visualizzazione di eventi mediante la cronologia CloudTrail eventi](#).



Per una registrazione continua degli eventi che includa gli eventi per AWS TNB, crea un trail. Account AWS Un trail consente di CloudTrail distribuire i file di registro in un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il percorso registra gli eventi di tutte le Regioni nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, è possibile configurare altri AWS servizi per analizzare con maggiore dettaglio e usare i dati raccolti nei CloudTrail log. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [Ricezione di file di CloudTrail registro da più regioni e Ricezione di file di registro da](#)

Tutte le azioni AWS TNB vengono registrate CloudTrail e documentate nel [AWS Telco Network Builder API Reference](#). Ad esempio, le chiamate a `CreateSolNetworkInstance` e `CreateSolNetworkPackage` le azioni generano voci nei file di CloudTrail registro. `CreateSolFunctionPackage`

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 credenziali utente root o AWS Identity and Access Management (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.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

## Informazioni sulle voci dei file di registro di AWS TNB

Un trail è una configurazione che consente la distribuzione di eventi come i file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro possono contenere una o più voci di registro. Un evento rappresenta una singola richiesta da un'origine e include informazioni sull'operazione richiesta, data e ora dell'operazione, parametri della richiesta e così via. CloudTrail

i file di registro non sono una traccia stack ordinata delle chiamate pubbliche dell'API, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra 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"
}

```

## AWSAttività 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:

Attività	Cosa succede durante questa attività
AppInstallation	Installa il plug-in Multus sul cluster Amazon EKS.
AppUpdate	Aggiorna le funzioni di rete già installate in un'istanza di rete.
ClusterStorageClassConfiguration	Configura la classe di storage (driver CSI) su un cluster Amazon EKS.
FunctionDeletion	Elimina le funzioni di rete dalle AWS risorse TNB.
FunctionInstantiation	Implementa le funzioni di rete utilizzando HELM.
FunctionUninstallation	Disinstalla la funzione di rete da un cluster Amazon EKS.
HookExecution	Esegue gli hook del ciclo di vita come definito nell'NSD.
InfrastructureCancellation	Annulla un servizio di rete.
InfrastructureInstantation	AWSFornisce risorse per conto dell'utente.
InfrastructureTermination	Deprovisioning delle AWS risorse richiamate tramite AWS TNB.
InventoryDeregistration	Annulla la registrazione delle risorse da TNBAWS. AWS

Attività	Cosa succede durante questa attività
KubernetesClusterConfigurati on	Configura il cluster Kubernetes e aggiunge ruoli IAM aggiuntivi ad Amazon EKS come definito nell'NSD. AuthMap
NetworkServiceFinalization	Finalizza il servizio di rete e fornisce un aggiornamento dello stato con esito positivo o negativo.
NetworkServiceInstantiation	Inizializza il servizio di rete.
SelfManagedNodesConfigurati on	Avvia i nodi autogestiti con il piano di controllo di Amazon EKS e Kubernetes.

## Quote di servizio per AWS Telco Network Builder

Le quote di servizio, a cui si fa riferimento anche come limiti, rappresentano il numero massimo di risorse di servizio o operazioni 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	Default	Adattate	Descrizione
Operazioni simultanee di servizi di rete in corso	Ogni regione supportata: 40	<a href="#">Sì</a>	Il numero massimo di operazioni simultanee di servizi di rete in corso 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 dei servizi di rete in una regione.

# Cronologia dei documenti per la AWS TNB User Guide

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

Modifica	Descrizione	Data
<a href="#">Versione Kubernetes per cluster</a>	AWS TNB ora supporta le versioni 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 abilitare, aggiungi le proprietà nel nodo AWS.Compute.eks o AWS.Compute.eksManagedNode.SelfManagedNode</a>	2 aprile 2024
<a href="#">Support per node labels</a>	<a href="#">Puoi allegare etichette di nodi al tuo gruppo di nodi nel nodo AWS.Compute.eks o AWS.Compute.eksManagedNode.SelfManagedNode</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 controllo della sorgente/destinazione dell'interfaccia di rete tramite il nodo .Networking.eni. AWS	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 nel SelfManagedNode nodo <a href="#">AWS.Compute.eksManagedNode</a> o <a href="#">AWS.Compute.eks</a> , AWS TNB ottiene l'autorizzazione relativa agli ENI dalla proprietà, se disponibile, o dalla proprietà. <code>multus_role</code> <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. PRIVATE	11 dicembre 2023
<a href="#">Versione Kubernetes per cluster</a>	AWS TNB ora supporta le versioni 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.SecurityGroupEgressRule](#) e [AWS.compute.eks](#) per configurazioni IPv6. [Abbiamo anche aggiunto i nodi AWS.networking.natGateway](#) e [.Networking.Route](#) per la [configurazione NAT64.AWS](#). Abbiamo aggiornato il ruolo di servizio AWS TNB e il ruolo di servizio AWS TNB per il gruppo di nodi Amazon EKS per le autorizzazioni IPv6. [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 AWS CloudFormation e per abilitare l'istanza 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.eks AWS SelfManagedNode</a>	AWS TNB ora supporta i tag per la definizione del nodo. <code>AWS.Compute.EKSSelfManagedNode</code>	22 agosto 2023
<a href="#">AWS TNB supporta le istanze che sfruttano IMDSv2</a>	All'avvio dell'istanza, è necessario utilizzare IMDSv2.	14 agosto 2023
<a href="#">Autorizzazioni aggiornate per MultusRoleInlinePolicy</a>	<code>MultusRoleInlinePolicy</code> 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 <code>AuthRole</code> 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.compute.eks SelfManagedNode</a>	È possibile 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.