Manual do usuário

AWS Construtor de rede Telco

AWS Construtor de rede Telco: Manual do usuário

Table of Contents

O que é o AWS TNB? 1
Novo no AWS? 2
Para quem é o AWS TNB? 2
Por que usar o AWS TNB? 2
Acesso ao AWS TNB 4
Definição de preço para o AWS TNB 4
Próximas etapas 5
Como funciona 6
Arquitetura 6
Integração 7
Cotas 8
Conceitos 9
Ciclo de vida de uma função de rede 9
Use interfaces padronizadas 10
Pacotes de NF 11
Descrição de serviço do NF 12
Gerenciamento e operações 13
Descritores de serviços de rede 14
Configuração 16
Inscreva-se para AWS 16
Escolha uma AWS região 17
Observar o endpoint do serviço 17
(Opcional) Instale o AWS CLI 18
Criar um usuário do IAM 18
Configurar funções AWS do TNB 19
Conceitos básicos 20
Pré-requisitos 20
Criar um pacote de funções 21
Criar um pacote de rede 21
Criar e instanciar uma instância de rede 21
Limpeza 22
Pacotes de funções 23
Criar 21
Exibir 24

Baixar um pacote ... 25
Excluir um pacote ... 25
Pacotes de rede ... 27
Criar ... 21
Exibir ... 28
Baixar ... 29
Excluir ... 29
Rede ... 31
Instanciar ... 31
Exibir ... 32
Atualizar ... 32
Encerrar e excluir ... 33
Operações de rede ... 35
Exibir ... 35
Cancelar ... 36
Referência TOSCA ... 37
Modelo de VNFD ... 37
Sintaxe ... 37
Modelo de topologia ... 37
AWS.VNF ... 38
AWS.Artifacts.Helm ... 39
Modelo de NSD ... 40
Sintaxe ... 40
Uso de parâmetros definidos ... 41
Importação de VNFD ... 41
Modelo de 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.Computação. PlacementGroup ... 62
AWS.Computação. UserData ... 64
AWS.Trabalho em rede. SecurityGroup ... 66
AWS.Trabalho em rede. SecurityGroupEgressRule ... 67
AWS.Trabalho em rede. SecurityGroupIngressRule ... 70

AWS.Resource.Import 73
AWS.Networking.ENI 74
AWS.HookExecution 76
AWS.Trabalho em rede. InternetGateway 78
AWS.Trabalho em rede. RouteTable 80
AWS.Networking.Subnet 81
AWS.Deployment.VNFDeployment 84
AWS.Networking.VPC 86
AWS.Networking.NATGateway 88
AWS.Networking.Route 89
Nós comuns 91
AWS.HookDefinition.Bash 91
Segurança 93
Proteção de dados 94
Tratamento de dados 95
Criptografia inativa 95
Criptografia em trânsito 95
Privacidade do tráfego entre redes 95
Gerenciamento de identidade e acesso 95
Público 96
Autenticando com identidades 96
Gerenciamento do acesso usando políticas 100
Como o AWS Telco Network Builder funciona com o IAM 103
Exemplos de políticas baseadas em identidade 110
Solução de problemas 124
Validação de conformidade 126
Resiliência 128
Segurança da infraestrutura 128
Modelo de segurança de conectividade de rede 129
Versão do IMDS 130
Monitoramento 131
Logs do CloudTrail 131
Informações do AWS TNB no CloudTrail 131
Noções básicas das entradas de arquivos de log do AWS TNB 132
Tarefas de implantação 134
Cotas 136

Histórico do documento 137
..... cxlii

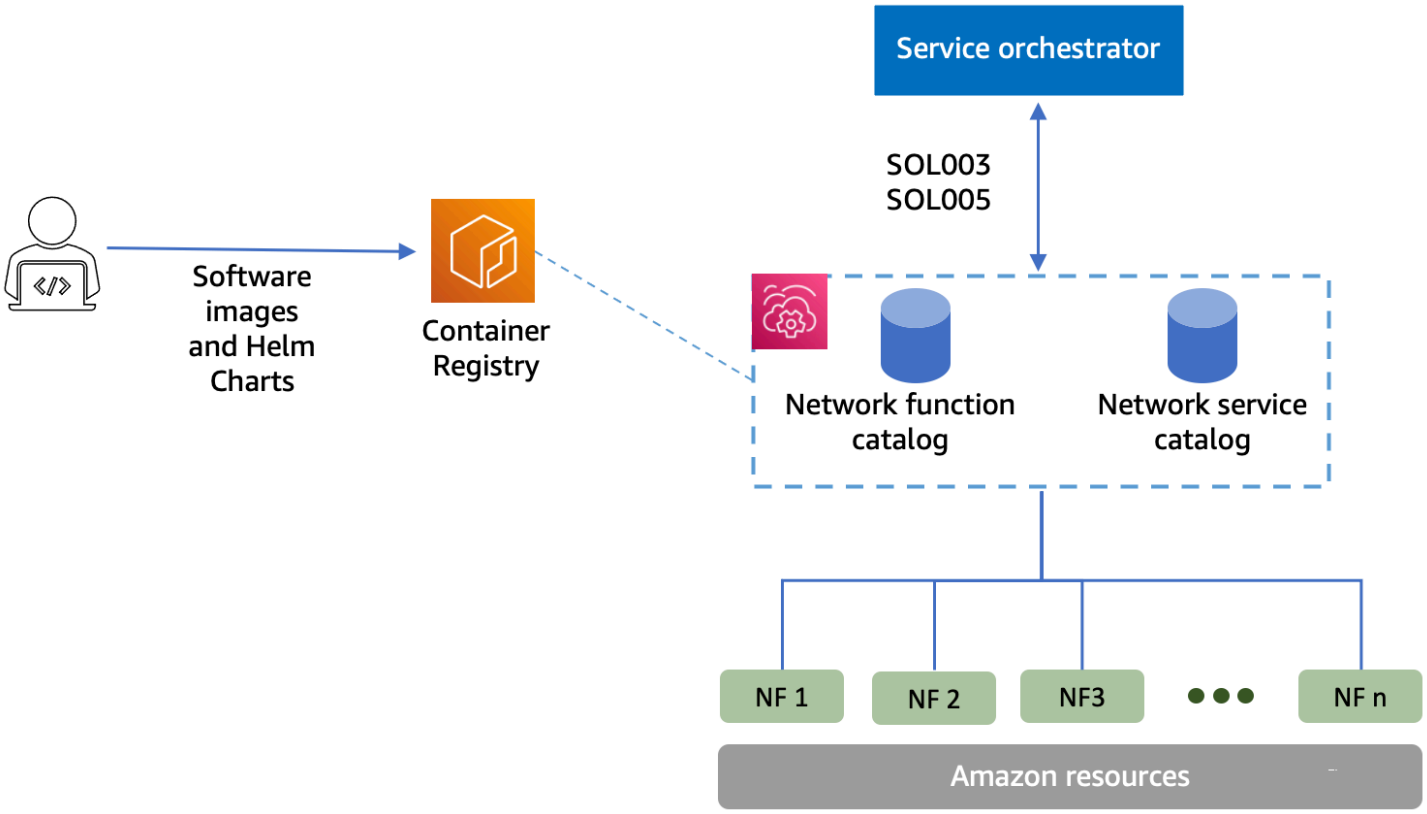
O que é o AWS Telco Network Builder?

O AWS Telco Network Builder (AWS TNB) é um serviço da AWS que fornece aos provedores de serviços de comunicação (CSPs) uma maneira eficiente de implantar, gerenciar e escalar redes 5G na infraestrutura da AWS.

Com o AWS TNB, você implanta redes 5G escaláveis e seguras na Nuvem AWS usando imagens de software em contêineres de maneira automatizada. Você não precisa aprender novas tecnologias, decidir qual serviço de computação usar nem saber como provisionar e configurar os recursos da AWS.

Em vez disso, você descreve a infraestrutura de sua rede e fornece as imagens de software das funções de rede de seus parceiros provedores de software independentes (ISVs). O AWS TNB se integra a serviços da AWS e orquestradores de serviços terceirizados para provisionar automaticamente a infraestrutura da AWS necessária, implantar funções de rede em contêineres e configurar o gerenciamento de rede e acesso para criar um serviço de rede totalmente operacional.

O diagrama a seguir ilustra as integrações lógicas entre o AWS TNB e os orquestradores de serviços para implantar funções de rede usando interfaces padrão baseadas no Instituto Europeu de Padrões de Telecomunicações (ETSI).



Tópicos

- Novo no AWS?
- Para quem é o AWS TNB?
- Por que usar o AWS TNB?
- Acesso ao AWS TNB
- Definição de preço para o AWS TNB
- Próximas etapas

Novo no AWS?

Caso esteja começando a usar os produtos e serviços da AWS, passe a saber mais com os seguintes recursos:

- Introdução à AWS
- Conceitos básicos da AWS

Para quem é o AWS TNB?

O AWS TNB é para CSPs que buscam aproveitar a economia, a agilidade e a elasticidade que as ofertas da Nuvem AWS oferecem sem escrever e manter scripts e configurações personalizados para projetar, implantar e gerenciar serviços de rede. O AWS TNB provisiona automaticamente a infraestrutura da AWS necessária, implanta funções de rede em contêineres e configura o gerenciamento de rede e acesso para criar serviços de rede totalmente operacionais com base nos descritores de serviços de rede definidos pelo CSP e nas funções de rede que o CSP deseja implantar.

Por que usar o AWS TNB?

A seguir estão alguns dos motivos pelos quais um CSP gostaria de usar o AWS TNB:

Ajuda a simplificar tarefas

Forneça mais eficiência às suas operações de rede, como implantação de novos serviços, atualização e upgrade de funções de rede e alteração de topologias de infraestrutura de rede.

Integra-se com orquestradores

O AWS TNB se integra a orquestradores de serviços terceirizados populares que são compatíveis com ETSI.

Faz escalonamento

Você pode configurar o AWS TNB para escalar os recursos da AWS para atender à demanda de tráfego, realizar atualizações de funções de rede com mais eficiência, implementar alterações na topologia da infraestrutura de rede e reduzir o tempo de implantação de novos serviços 5G de dias para horas.

Inspeciona e monitora recursos da AWS

O AWS TNB permite que você inspecione e monitore os recursos da AWS que dão suporte à sua rede em um único painel, como Amazon VPC, Amazon EC2 e Amazon EKS.

Compatibilidade com modelos de serviço

O AWS TNB permite criar modelos de serviço para todos os workloads de telecomunicações (RAN, Core, IMS). Você pode criar uma nova definição de serviço, reutilizar um modelo existente ou integrar-se a um pipeline de integração contínua e entrega contínua (CI/CD) para publicar uma nova definição.

Rastreia as alterações nas implantações de rede

Quando você altera a configuração de uma implantação de função de rede, por exemplo, alterando o tipo de uma instância do Amazon EC2, você pode rastrear as alterações de forma repetível e escalável. Fazer isso manualmente exigiria gerenciar o estado da rede, criar e excluir recursos e prestar atenção à ordem das alterações necessárias. Ao usar o AWS TNB para gerenciar o ciclo de vida da função de rede, você só faz as alterações nos descritores de serviço de rede que descrevem a função de rede. O AWS TNB fará automaticamente as alterações necessárias na ordem correta.

Simplifica o ciclo de vida da função de rede

Você pode gerenciar a primeira e todas as versões subsequentes de uma função de rede e especificar quando fazer upgrade. Você também pode gerenciar suas aplicações RAN, Core, IMS e de rede da mesma forma.

Acesso ao AWS TNB

Você pode criar, acessar e gerenciar seus recursos do AWS TNB usando qualquer uma das seguintes interfaces:

- Console do AWS TNB: fornece uma interface da web para gerenciar sua rede.
- API do AWS TNB: fornece uma API RESTful para realizar ações do AWS TNB. Para obter mais informações, consulte Referência de API do AWS TNB
- AWS Command Line Interface (AWS CLI): fornece comandos para um amplo conjunto de serviços da AWS, inclusive o AWS TNB. É compatível com Windows, macOS e Linux. Para obter mais informações, consulte AWS Command Line Interface.
- SDKs da AWS: fornecem APIs específicas de idioma e preenche muitos dos detalhes da conexão. Por exemplo, cálculo de assinaturas, tratamento de novas tentativas de solicitação e tratamento de erros. Para mais informações, consulte SDKs da AWS.

Definição de preço para o AWS TNB

O AWS TNB ajuda os CSPs a automatizar a implantação e o gerenciamento de suas redes de telecomunicações na AWS. Você paga pelas duas dimensões a seguir ao usar o AWS TNB:

- Por horas de item de função de rede gerenciada (MNFI).
- Por número de solicitações de API.

Você também incorre em cobranças adicionais ao usar outros serviços da AWS em conjunto com o AWS TNB. Para obter mais informações, consulte Definição de preço do AWS.

Para exibir sua fatura, acesse o Painel do Billing and Cost Management no console do AWS Billing and Cost Management. Sua fatura contém links para relatórios de uso que fornecem mais detalhes da fatura. Para obter mais informações sobre o faturamento de contas da AWS, consulte Faturamento de contas da AWS.

Se tiver dúvidas sobre faturamento, contas e eventos da AWS, entre em contato com o Suporte da AWS.

O AWS Trusted Advisor é um serviço que você pode usar para ajudar a otimizar os custos, a segurança e a performance do ambiente da AWS. Para obter mais informações, consulte AWS Trusted Advisor.

Próximas etapas

Consulte os tópicos a seguir para obter informações sobre os conceitos básicos do AWS TNB.

- Configurando o AWS TNB: concluir as etapas de pré-requisitos.
- Começando com o AWS TNB: implantar sua primeira função de rede, como Unidade Centralizada (UC), Função de Gerenciamento de Acesso e Mobilidade (AMF), Função de Plano de Usuário (UPF) ou um núcleo 5G completo.

Funcionamento do AWS TNB

O AWS TNB se integra a orquestradores e recursos padronizados da AWS de ponta a ponta para operar redes 5G completas.

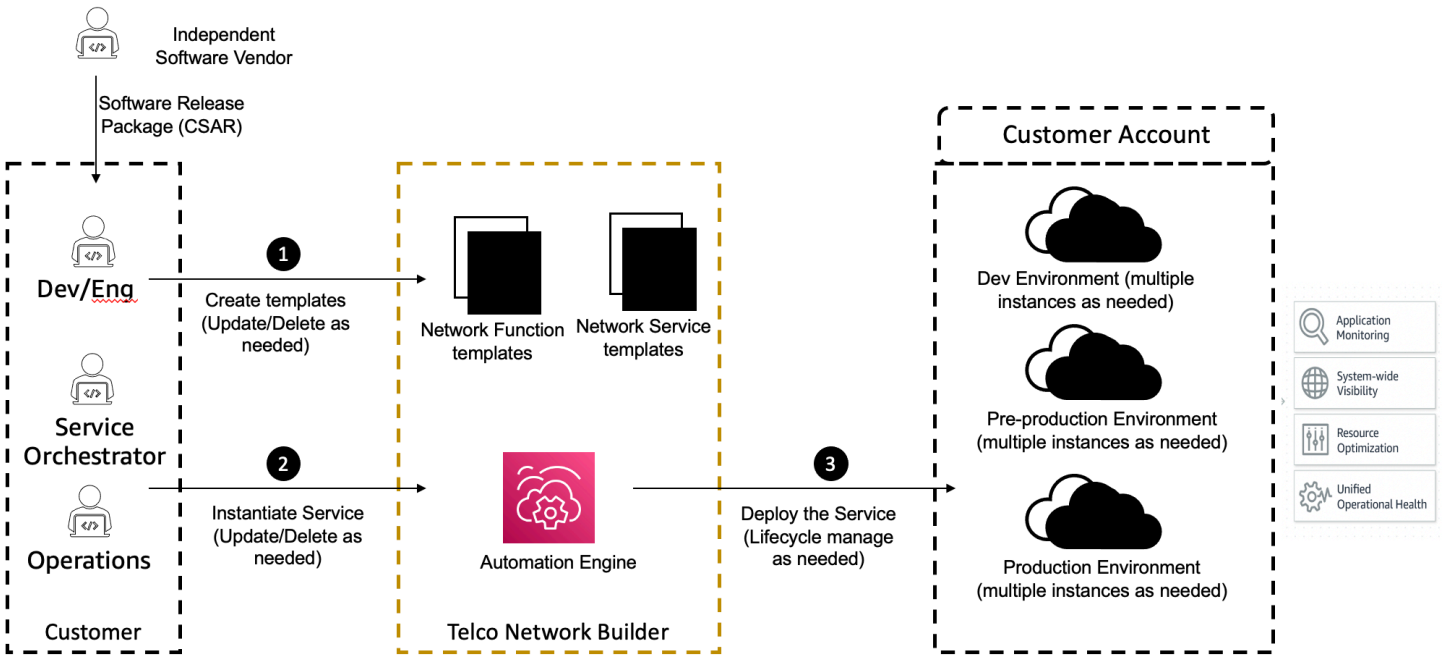
O AWS TNB permite ingerir pacotes de funções de rede e descritores de serviços de rede (NSDs) e fornece o mecanismo de automação para operar suas redes. Você pode usar seu orquestrador de ponta a ponta e integrar-se às APIs do AWS TNB ou usar os SDKs do AWS TNB para criar seu próprio fluxo de automação. Para obter mais informações, consulte Arquitetura do AWS TNB.

Tópicos

- Arquitetura do AWS TNB
- Integração com Serviços da AWS
- Cotas de recursos do AWS TNB

Arquitetura do AWS TNB

O AWS TNB oferece a capacidade de realizar operações de gerenciamento do ciclo de vida por meio do AWS Management Console, da AWS CLI, da API REST do AWS TNB e dos SDKs. Isso permite que as diferentes personalidades do CSP, como membros das equipes de Engenharia, Operações e Sistema Programático, tirem proveito do AWS TNB. Você cria e carrega um pacote de funções de rede como um arquivo Cloud Service Archive (CSAR). O arquivo CSAR contém charts do Helm, imagens de software e um descritor de função de rede (NFD). Você pode usar modelos para implantar repetidamente várias configurações desse pacote. Você cria modelos de serviço de rede definindo a infraestrutura e as funções de rede que você deseja implantar. Você pode usar substituições de parâmetros para implantar configurações diferentes em locais diferentes. Em seguida, você pode instanciar uma rede usando os modelos e implantar suas funções de rede na infraestrutura da AWS. O AWS TNB fornece a visibilidade de suas implantações.



Integração com Serviços da AWS

Uma rede 5G é composta por um conjunto de funções de rede em contêineres interconectadas implantadas em milhares de clusters do Kubernetes. O AWS TNB se integra aos seguintes Serviços da AWS como APIs específicas de telecomunicações para criar um serviço de rede totalmente operacional:

- Amazon Elastic Container Registry (Amazon ECR) para armazenar artefatos de funções de rede de provedores de software independente (ISVs).
- Amazon Elastic Kubernetes Service (Amazon EKS) para configurar clusters.
- Amazon VPC para estruturas de rede.
- Grupos de segurança usando AWS CloudFormation.
- AWS CodePipeline para alvos de implantação em Regiões da AWS, zonas locais da AWS e AWS Outposts.
- IAM para definir perfis.
- AWS Organizations para controlar o acesso às APIs do AWS TNB.
- AWS Health Dashboard e AWS CloudTrail para monitorar a integridade e as métricas de publicação.

Cotas de recursos do AWS TNB

Sua Conta da AWS tem cotas padrão, anteriormente chamadas de limites, para cada AWS service (Serviço da AWS). A menos que especificado de outra forma, cada cota é específica a uma Região da AWS. Você pode solicitar aumentos para algumas cotas, mas não para todas.

Para exibir todas as cotas do AWS TNB, abra o console do Service Quotas. No painel de navegação, escolha Serviços da AWS e selecione AWS TNB.

Para solicitar o aumento da cota, consulte Requesting a Quota Increase (Solicitar um aumento de cota) no Guia do usuário do Service Quotas.

A Conta da AWS tem as seguintes cotas relacionadas ao AWS TNB.

Cota de recurso	Descrição	Valor padrão	Ajustável?
Instâncias do serviço de rede	O número máximo de instâncias do serviço de rede em uma região.	800	Sim
Operações simultâneas de serviços de rede contínuas	O número máximo de operações simultâneas de serviços de rede contínuas em uma região.	40	Sim
Pacotes de rede	O número máximo de pacotes de rede em uma região.	40	Sim
Pacotes de funções	O número máximo de pacotes de funções em uma região.	200	Sim

AWS Conceitos do TNB

Este tópico descreve conceitos essenciais para ajudá-lo a começar a usar o AWS TNB.

Conteúdo

- Ciclo de vida de uma função de rede
- Use interfaces padronizadas
- Pacotes de funções de rede para AWS TNB
- Descritores de serviço de função de rede para TNB AWS
- Gestão e operações da AWS TNB
- Descritores de serviços de rede para TNB AWS

Ciclo de vida de uma função de rede

AWS O TNB ajuda você em todo o ciclo de vida de suas funções de rede. O ciclo de vida da função de rede inclui os seguintes estágios e atividades:

Planejamento

1. Planeje sua rede identificando as funções de rede a serem implantadas.
2. Coloque as imagens do software de função de rede em um repositório de imagens de contêiner.
3. Crie os pacotes CSAR para implantar ou fazer upgrade.
4. Use o AWS TNB para carregar o pacote CSAR que define sua função de rede (por exemplo, CU AMF e UPF) e integre com um pipeline de integração contínua e entrega contínua (CI/CD) que pode ajudá-lo a criar novas versões do seu pacote CSAR à medida que novas imagens de software de função de rede ou scripts de clientes estiverem disponíveis.

Configuração

1. Identifique as informações necessárias para a implantação, como tipo de computação, versão da função de rede, informações de IP e nomes dos recursos.
2. Use as informações para criar seu descritor de serviço de rede (NSD).
3. Ingira NSDs que definem suas funções de rede e os recursos necessários para que a função de rede seja instanciada.

Instanciação

1. Crie a infraestrutura exigida pelas funções de rede.
2. Instancie (ou provisione) a função de rede conforme definida em seu NSD e comece a transportar tráfego.
3. Valide os ativos.

Produção

Durante o ciclo de vida da função de rede, você concluirá as operações de produção, como:

- Atualize a configuração da função de rede, por exemplo, atualize um valor na função da rede implantada.
- Substitua ou desative a função de rede.

Use interfaces padronizadas

AWS O TNB se integra aos orquestradores de serviços compatíveis com o Instituto Europeu de Padrões de Telecomunicações (ETSI), permitindo que você simplifique a implantação de seus serviços de rede. Os orquestradores de serviços podem usar os SDKs do AWS TNB, a CLI ou as APIs para iniciar operações, como instanciar ou atualizar uma função de rede para uma nova versão.

AWS O TNB suporta as seguintes especificações.

Especificação	Versão	Descrição
ETSI SOL001	v3.6.1	Define padrões para permitir descritores de funções de rede baseados em TOSCA.
ETSI SOL002	v3.6.1	Define modelos em torno do gerenciamento de funções de rede.
ETSI SOL003	v3.6.1	Define padrões para o gerenciamento do ciclo de vida das funções de rede.
ETSI SOL004	v3.6.1	Define padrões CSAR para pacotes de funções de rede.

Especificação	Versão	Descrição
ETSI SOL005	v3.6.1	Define padrões para pacotes de serviços de rede e gerenciamento do ciclo de vida do serviço de rede.
ETSI SOL007	v3.5.1	Define padrões para permitir descritores de serviços de rede baseados em TOSCA.

Pacotes de funções de rede para AWS TNB

Com o AWS TNB, você pode armazenar pacotes de funções de rede que estejam em conformidade com o ETSI SOL001/SOL004 em um catálogo de funções. Em seguida, você pode carregar pacotes do Cloud Service Archive (CSAR) que contêm artefatos que descrevem sua função de rede.

- Descritor de função de rede: define metadados para a integração de pacotes e o gerenciamento de funções de rede
- Imagens de software: referencia as imagens de contêiner de funções de rede. O Amazon Elastic Container Registry (Amazon ECR) pode atuar como um repositório de imagens de funções de rede.
- Arquivos adicionais: use para gerenciar a função de rede; por exemplo, scripts e charts do Helm.

O CSAR é um pacote definido pelo padrão OASIS TOSCA e inclui um descritor de rede/serviço que está em conformidade com a especificação OASIS TOSCA YAML. Para obter informações sobre a especificação YAML necessária, consulteReferência TOSCA para o AWS TNB.

Veja a seguir um exemplo de descritor de função de rede.

```
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"
```

Descritores de serviço de função de rede para TNB AWS

AWS O TNB armazena descritores de serviços de rede (NSDs) sobre as funções de rede que você deseja implantar e como deseja implantá-las no catálogo. Você pode carregar seu arquivo YAML NSD, conforme descrito pelo ETSI SOL007, para incluir:

- O NF que você deseja implantar
- Instruções de rede
- Instruções de computação
- Ganchos do ciclo de vida (scripts personalizados)

AWS O TNB suporta os padrões ETSI para a modelagem de recursos, como rede, serviço e função, na linguagem TOSCA. AWS O TNB torna seu uso mais eficiente, Serviços da AWS modelando-os de uma forma que seu orquestrador de serviços compatível com ETSI possa entender.

A seguir está um trecho de um NSD mostrando como modelar. Serviços da AWS A função de rede será implantada em um cluster do Amazon EKS com o Kubernetes versão 1.27. As sub-redes dos aplicativos são Subnet01 e Subnet02. Em seguida, você pode definir o NodeGroups para seus aplicativos com uma Amazon Machine Image (AMI), tipo de instância e configuração de escalonamento automático.

```
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
```

Gestão e operações da AWS TNB

Com o AWS TNB, você pode gerenciar sua rede usando operações de gerenciamento padronizadas de acordo com ETSI SOL003 e SOL005. Você pode usar as APIs do AWS TNB para realizar operações de ciclo de vida, como:

- Instanciação das suas funções de rede.
- Encerramento das suas funções de rede.

- Atualização das suas funções de rede para substituir as implantações do Helm.
- Gerenciamento de versões de seus pacotes de funções de rede.
- Gerenciamento de versões de seus NSDs.
- Recuperação de informações sobre suas funções de rede implantadas.

Descritores de serviços de rede para TNB AWS

Um descritor de serviço de rede (NSD) é um arquivo `.yaml` em um pacote de rede que usa o padrão TOSCA para descrever as funções de rede que você quer implantar e a infraestrutura da AWS na qual deseja implantá-las. Para definir seu NSD e configurar seus recursos subjacentes e operações do ciclo de vida da rede, você deve entender o esquema NSD TOSCA suportado pelo TNB. AWS

Seu arquivo NSD é dividido nas seguintes partes:

1. Versão da definição TOSCA: é a primeira linha do seu arquivo NSD YAML e contém as informações da versão, mostradas no exemplo a seguir.

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

2. VNFD: o NSD contém a definição da função de rede na qual realizar operações de ciclo de vida. Cada função de rede deve ser identificada pelos seguintes valores:
   - Um ID exclusivo para `descriptor_id`. O ID deve corresponder ao ID no pacote CSAR de funções de rede.
   - Um nome exclusivo para `namespace`. O nome precisa estar associado a um ID exclusivo para ser referenciado com mais facilidade em todo o arquivo NSD YAML, mostrado no exemplo a seguir.

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

3. Modelo de topologia: define os recursos a serem implantados, a implantação da função de rede e quaisquer scripts personalizados, como ganchos do ciclo de vida. Isso é mostrado no exemplo a seguir.

```
topology_template:

  node_templates:
```

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

4. Nós adicionais: cada recurso modelado tem seções para propriedades e requisitos. As propriedades descrevem atributos opcionais ou obrigatórios de um recurso, como a versão. Os requisitos descrevem dependências que precisam ser fornecidas como argumentos. Por exemplo, para criar um recurso de grupo de nós do Amazon EKS, ele precisa ser criado dentro de um cluster do Amazon EKS. Isso é mostrado no exemplo a seguir.

```
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
```

Configurando o AWS TNB

Configure o AWS TNB concluindo as tarefas descritas neste tópico.

Tarefas

- Inscreva-se para AWS
- Escolha uma AWS região
- Observar o endpoint do serviço
- (Opcional) Instale o AWS CLI
- Criar um usuário do IAM
- Configurar funções AWS do TNB

Inscreva-se para AWS

Quando você se inscreve na Amazon Web Services, você Conta da AWS se inscreve automaticamente em todos os serviços AWS, incluindo AWS TNB. Você será cobrado apenas pelos serviços que usar.

Se você Conta da AWS já tiver um, vá para a próxima tarefa. Se você não tem uma Conta da AWS, siga o procedimento abaixo para criar uma.

Para criar um Conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções on-line.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

Escolha uma AWS região

Para ver a lista de regiões disponíveis para AWS TNB, consulte a Lista de serviços AWS regionais. Para exibir a lista de endpoints para acesso programático, consulte Endpoints do AWS TNB no Referência geral da AWS.

Observar o endpoint do serviço

Para se conectar programaticamente a um AWS serviço, você usa um endpoint. Além dos AWS endpoints padrão, alguns AWS serviços oferecem endpoints FIPS em regiões selecionadas. Para obter mais informações, consulte Endpoints de serviço da AWS.

Nome da região	Região	Endpoint	Protocolo	
Leste dos EUA (Norte da Virgínia)	us-east-1	tnb.us-east-1.amazonaws.com	HTTPS	
Oeste dos EUA (Oregon)	us-west-2	tnb.us-west-2.amazonaws.com	HTTPS	
Ásia-Pacífico (Seul)	ap-northeast-2	tnb.ap-northeast-2.amazonaws.com	HTTPS	
Ásia-Pacífico (Sydney)	ap-southeast-2	tnb.ap-southeast-2.amazonaws.com	HTTPS	
Canadá (Central)	ca-central-1	tnb.ca-central-1.amazonaws.com	HTTPS	
Europa (Frankfurt)	eu-central-1	tnb.eu-central-1.amazonaws.com	HTTPS	

Nome da região	Região	Endpoint	Protocolo	
Europa (Paris)	eu-west-3	tnb.eu-west-3.amazonaws.com	HTTPS	
Europa (Espanha)	eu-south-2	tnb.eu-south-2.amazonaws.com	HTTPS	
Europa (Estocolmo)	eu-north-1	tnb.eu-north-1.amazonaws.com	HTTPS	
América do Sul (São Paulo)	sa-east-1	tnb.sa-east-1.amazonaws.com	HTTPS	

(Opcional) Instale o AWS CLI

O AWS Command Line Interface (AWS CLI) fornece comandos para um amplo conjunto de AWS produtos e é compatível com Windows, macOS e Linux. Você pode acessar o AWS TNB usando o. AWS CLI Para começar a usar, consulte o Guia do usuário da AWS Command Line Interface. Para obter mais informações sobre os comandos do AWS TNB, consulte tnb na Referência de AWS CLI Comandos.

Criar um usuário do IAM

AWS Identity and Access Management (IAM) é um serviço web que ajuda você a controlar com segurança o acesso aos AWS recursos. Crie um perfil de usuário do IAM para usar credenciais de curto prazo para acessar a AWS.

Para criar o perfil, siga as instruções em Conceitos básicos no Guia do usuário do AWS IAM Identity Center .

Você também pode configurar o acesso programático configurando o AWS CLI para uso AWS IAM Identity Center no Guia do AWS Command Line Interface usuário.

Configurar funções AWS do TNB

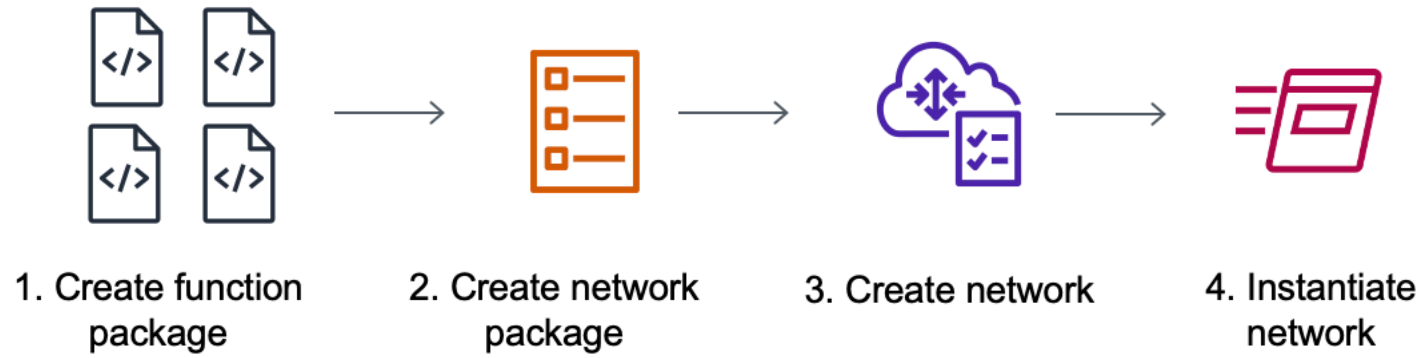
Você deve criar uma função de serviço do IAM para gerenciar diferentes partes da sua solução AWS TNB. AWS As funções de serviço do TNB podem fazer chamadas de API para outros AWS serviços, como AWS CloudFormation AWS CodeBuild, e vários serviços de computação e armazenamento, em seu nome, para instanciar e gerenciar recursos para sua implantação.

Para obter mais informações sobre a função de serviço AWS TNB, consulteGerenciamento de identidade e acesso para AWS TNB.

Começando com o AWS TNB

Este tutorial demonstra como você usa o AWS TNB para implantar uma função de rede, por exemplo, a Unidade Centralizada (UC), a Função de Gerenciamento de Acesso e Mobilidade (AMF) ou a Função de Plano de Usuário (UPF) 5G.

O diagrama a seguir ilustra o processo de implantação:



Tarefas

- Pré-requisitos
- Criar um pacote de funções
- Criar um pacote de rede
- Criar e instanciar uma instância de rede
- Limpeza

Pré-requisitos

Antes de realizar uma implantação bem-sucedida, você deve ter o seguinte:

- Um plano AWS de Business Support.
- Permissões por meio de funções do IAM.
- Um pacote de função de rede (NF) compatível com o ETSI SOL001/SOL004.
- Modelos de Network Service Descriptor (NSD) que estão em conformidade com o ETSI SOL007.

Criar um pacote de funções

Para criar um pacote de funções

1. Abra o console AWS do TNB em https://console.aws.amazon.com/tnb/.
2. Selecione Pacotes de funções no painel de navegação.
3. Escolha Criar pacote de funções.
4. Em Carregar pacote de funções, escolha Escolher arquivo e carregue seu pacote CSAR como um `.zip` arquivo.
5. (Opcional) Em Tags, escolha Adicionar nova tag e insira uma chave e um valor. Você pode usar tags para pesquisar e filtrar seus recursos ou monitorar seus AWS custos.
6. Escolha Próximo.
7. Revise os detalhes do pacote e escolha Criar pacote de funções.

Criar um pacote de rede

Para criar um pacote de rede

1. No painel de navegação, selecione Pacotes de rede.
2. Escolha Criar pacote de rede.
3. Em Carregar pacote de rede, escolha Escolher arquivo e carregue seu NSD como um `.zip` arquivo.
4. (Opcional) Em Tags, escolha Adicionar nova tag e insira uma chave e um valor. Você pode usar tags para pesquisar e filtrar seus recursos ou monitorar seus AWS custos.
5. Escolha Próximo.
6. Escolha Criar pacote de rede.

Criar e instanciar uma instância de rede

Para criar e instanciar uma instância de rede

1. No painel de navegação, selecione Redes.
2. Clique em Criar instância de rede.
3. Insira um nome e uma descrição para a rede e escolha Próximo.

4. Selecione seu NSD. Verifique os detalhes e escolha Próximo.
5. Clique em Criar instância de rede. O estado inicial é `Created`.
6. Escolha o ID da instância de rede e, em seguida, escolha Instanciar.
7. Escolha Instanciar rede.
8. Use o ícone Atualizar para rastrear o status da sua instância de rede.

Limpeza

Para limpar recursos

1. No painel de navegação, selecione Redes.
2. Escolha o ID da rede e, em seguida, escolha Encerrar.
3. Quando a confirmação for solicitada, insira o ID da rede e escolha Encerrar.
4. Use o ícone Atualizar para rastrear o status da sua instância de rede.
5. (Opcional) Selecione a rede e escolha Excluir.

Pacotes de funções do AWS TNB

Um pacote de funções é um arquivo .zip no formato CSAR (Cloud Service Archive) que contém uma função de rede (um aplicativo de telecomunicação padrão ETSI) e um descritor de pacote de funções que usa o padrão TOSCA para descrever como as funções de rede devem ser executadas em sua rede.

Tarefas

- Crie um pacote de funções no AWS TNB
- Exibir um pacote de funções no AWS TNB
- Baixar um pacote de funções do AWS TNB
- Excluir um pacote de funções do AWS TNB

Crie um pacote de funções no AWS TNB

Saiba como criar um pacote de funções no catálogo de funções de rede do AWS TNB. A criação de um pacote de funções é a primeira etapa para criar uma rede no TNB. Depois de carregar um pacote de funções, você precisa criar um pacote de rede.

Console

Para criar um pacote de funções usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. Selecione Pacotes de funções no painel de navegação.
3. Escolha Criar pacote de funções.
4. Escolha Escolher arquivo e carregue o pacote CSAR do seu NF.
5. Escolha Próximo.
6. Revise os detalhes do pacote.
7. Escolha Criar pacote de funções.

AWS CLI

Para criar um pacote de funções usando a AWS CLI

1. Use o comando create-sol-function-package para criar um novo pacote de funções:

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

2. Use o comando put-sol-function-package-content para carregar o conteúdo do pacote de funções. Por exemplo:

```
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
```

Exibir um pacote de funções no AWS TNB

Saiba como exibir o conteúdo de um pacote de funções.

Console

Para exibir um pacote de funções usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. Selecione Pacotes de funções no painel de navegação.
3. Use a caixa de pesquisa para encontrar o pacote de funções

AWS CLI

Para exibir um pacote de funções usando a AWS CLI

1. Use o comando list-sol-function-packages para listar seus pacotes de funções.

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

2. Use o comando get-sol-function-package para exibir detalhes sobre um pacote de funções.

```
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
```

Baixar um pacote de funções do AWS TNB

Saiba como baixar um pacote de funções do catálogo de funções de rede do AWS TNB.

Console

Para baixar um pacote de funções usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, no lado esquerdo do console, escolha Pacotes de funções.
3. Use a caixa de pesquisa para encontrar o pacote de funções
4. Escolha o pacote de funções
5. Em Ações, escolha Baixar.

AWS CLI

Para baixar um pacote de funções usando a AWS CLI

Use o comando get-sol-function-package-content para baixar um pacote de funções.

```
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
```

Excluir um pacote de funções do AWS TNB

Saiba como excluir um pacote de funções do catálogo de funções de rede do AWS TNB. Para excluir um pacote de funções, é preciso que ele esteja desabilitado.

Console

Para excluir um pacote de funções usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. Selecione Pacotes de funções no painel de navegação.
3. Use a caixa de pesquisa para encontrar o pacote de funções.
4. Escolha um pacote de funções.
5. Escolha Ações, Desabilitar.
6. Escolha Ações, Excluir.

AWS CLI

Para excluir um pacote de funções usando a AWS CLI

1. Use o comando update-sol-function-package para desabilitar um pacote de funções.

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

2. Use o comando delete-sol-function-package para excluir um pacote de funções.

```
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
```

Pacotes de rede para o AWS TNB

Um pacote de rede é um arquivo .zip no formato CSAR (Cloud Service Archive) que define os pacotes de funções que você quer implantar e a infraestrutura da AWS na qual pretender implantá-los.

Tarefas

- Criar um pacote de rede no AWS TNB
- Exibir um pacote de rede no AWS TNB
- Baixar um pacote de rede do AWS TNB
- Excluir um pacote de rede do AWS TNB

Criar um pacote de rede no AWS TNB

Um pacote de rede consiste em um arquivo descritor de serviço de rede (NSD) (obrigatório) e quaisquer arquivos adicionais (opcionais), como scripts específicos às suas necessidades. Por exemplo, se você tiver vários pacotes de funções em seu pacote de rede, poderá usar o NSD para definir quais funções de rede devem ser executadas em determinadas VPCs, sub-redes ou clusters do Amazon EKS.

Crie um pacote de rede depois de criar pacotes de funções. Depois de criar um pacote de rede, você precisa criar uma instância de rede.

Console

Para criar um pacote de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Pacotes de rede.
3. Escolha Criar pacote de rede.
4. Escolha Escolher arquivo e carregue o seu pacote CSAR.
5. Escolha Próximo.
6. Revise os detalhes do pacote.
7. Escolha Criar pacote de rede.

AWS CLI

Para criar um pacote de rede usando a AWS CLI

1. Use o comando create-sol-network-package para criar um pacote de rede.

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

2. Use o comando put-sol-network-package-content para carregar o conteúdo do pacote de rede. Por exemplo:

```
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
```

Exibir um pacote de rede no AWS TNB

Saiba como exibir o conteúdo de um pacote de rede.

Console

Para exibir um pacote de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Pacotes de rede.
3. Use a caixa de pesquisa para encontrar o pacote de rede.

AWS CLI

Para exibir um pacote de rede usando a AWS CLI

1. Use o comando list-sol-network-packages para listar seus pacotes de rede.

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

2. Use o comando get-sol-network-package para exibir detalhes sobre um pacote de rede.

```
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
```

Baixar um pacote de rede do AWS TNB

Saiba como baixar um pacote de rede do catálogo de serviços de rede do AWS TNB.

Console

Para baixar um pacote de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Pacotes de rede.
3. Use a caixa de pesquisa para encontrar o pacote de rede
4. Escolha o pacote de rede.
5. Em Ações, escolha Baixar.

AWS CLI

Para baixar um pacote de rede usando a AWS CLI

- Use o comando get-sol-network-package-content para baixar o conteúdo de um pacote.

```
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
```

Excluir um pacote de rede do AWS TNB

Saiba como excluir um pacote de rede do catálogo de serviços de rede do AWS TNB. Para excluir um pacote de rede, é preciso que ele esteja desabilitado.

Console

Para excluir um pacote de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Pacotes de rede.
3. Use a caixa de pesquisa para encontrar o pacote de rede
4. Escolher o pacote de rede
5. Escolha Ações, Desabilitar.
6. Escolha Ações, Excluir.

AWS CLI

Para excluir um pacote de rede usando a AWS CLI

1. Use o comando update-sol-network-package para desabilitar um pacote de rede.

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

2. Use o comando delete-sol-network-package para excluir um pacote de rede.

```
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
```

Instâncias de rede do AWS TNB

Uma instância de rede é uma rede única criada no AWS TNB que pode ser implantada.

Tarefas

- Instancie uma instância de rede usando AWS TNB
- Exibir uma instância de rede no AWS TNB
- Atualizar uma instância de rede no AWS TNB
- Encerrar e excluir uma instância de rede do AWS TNB

Instancie uma instância de rede usando AWS TNB

Você cria uma instância de rede depois de criar um pacote de rede. Depois de criar uma instância de rede, é preciso instanciá-la. Quando você instancia uma instância de rede, o AWS TNB implanta as funções de rede de acordo com as especificações no descritor do serviço de rede.

Console

Para criar e instanciar uma instância de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Redes.
3. Clique em Criar instância de rede.
4. Insira um nome e uma descrição para a instância e, em seguida, escolha Próximo.
5. Selecione seu NSD. Verifique os detalhes e escolha Próximo.
6. Clique em Criar instância de rede.
7. Escolha Instanciar.
8. Escolha Instanciar rede.
9. Atualize para rastrear o status da instância de rede.

AWS CLI

Para criar e instanciar uma instância de rede usando a AWS CLI

1. Use o comando create-sol-network-instance para criar uma instância de rede.

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

2. Use o comando instantiate-sol-network-instance para instanciar a instância de rede.

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

Exibir uma instância de rede no AWS TNB

Saiba como exibir uma instância de rede.

Console

Para exibir uma instância de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, escolha Interfaces de rede.
3. Use a caixa de pesquisa para encontrar a instância de rede.

AWS CLI

Para exibir uma instância de rede usando a AWS CLI

1. Use o comando list-sol-network-instances para listar suas instâncias de rede.

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

2. Use o comando get-sol-network-instance para exibir detalhes sobre uma instância de rede.

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

Atualizar uma instância de rede no AWS TNB

Saiba como atualizar uma instância de rede.

Console

Para atualizar a instância de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Redes.
3. Selecione o ID da instância de rede.
4. Na guia Funções, selecione a instância da função a ser atualizada.
5. Escolha Atualizar.
6. Insira suas substituições de atualização para confirmar a atualização.
7. Escolha Atualizar.
8. Atualize para rastrear o status da instância de rede.

AWS CLI

Usar a CLI para atualizar uma instância de rede

Use o comando update-sol-network-instance para atualizar uma instância de rede.

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

Encerrar e excluir uma instância de rede do AWS TNB

Para excluir uma instância de rede, é preciso que ela esteja encerrada.

Console

Para encerrar e excluir uma instância de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Redes.
3. Selecione o ID da instância de rede.
4. Escolha Encerrar.
5. Quando receber a solicitação de confirmação, insira o ID e escolha Encerrar.
6. Atualize para rastrear o status da instância de rede.

7. (Opcional) Selecione a instância de rede e escolha Excluir.

AWS CLI

Para encerrar e excluir uma instância de rede usando a AWS CLI

1. Use o comando terminate-sol-network-instance para encerrar uma instância de rede.

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

2. (Opcional) Use o comando delete-sol-network-instance para excluir uma instância de rede.

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

Operações de rede para o AWS TNB

Uma operação de rede é qualquer operação feita em sua rede, como instanciação ou encerramento de instância de rede.

Tarefas

- Exibir uma operação de rede
- Cancelar uma operação de rede

Exibir uma operação de rede

Exiba os detalhes de uma operação de rede, incluindo as tarefas envolvidas na operação de rede e o status das tarefas.

Console

Para exibir uma operação de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, escolha Interfaces de rede.
3. Use a caixa de pesquisa para encontrar a instância de rede.
4. Na guia Implantações, escolha a operação de rede.

AWS CLI

Para exibir uma operação de rede usando a AWS CLI

1. Use o comando list-sol-network-operations para listar todas as operações de rede.

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

2. Use o comando get-sol-network-operation para exibir detalhes sobre uma operação de rede.

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

Cancelar uma operação de rede

Saiba como cancelar uma operação de rede.

Console

Para cancelar uma operação de rede usando o console

1. Abra o console do AWS TNB em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Redes.
3. Selecione o ID da rede para abrir sua página de detalhes.
4. Na guia Implantações, escolha a operação de rede.
5. Escolha Cancelar operação.

AWS CLI

Para cancelar uma operação de rede usando a AWS CLI

Use o comando cancel-sol-network-operation para cancelar uma operação de rede.

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

Referência TOSCA para o AWS TNB

Topology and Orchestration Specification for Cloud Applications (TOSCA) é uma sintaxe declarativa que os CSPs usam para descrever uma topologia de serviços web baseados em nuvem, seus componentes, relacionamentos e os processos que os gerenciam. Os CSPs descrevem os pontos de conexão, os links lógicos entre os pontos de conexão e as políticas, como afinidade e segurança, em um modelo TOSCA. Os CSPs carregam o modelo para o AWS TNB, que sintetiza os recursos necessários para estabelecer uma rede 5G funcional em todas as zonas de disponibilidade da AWS.

Índice

- Modelo de VNFD
- Modelo de NSD
- Nós comuns

Modelo de VNFD

Define um modelo de descritor de função de rede virtual (VNFD).

Sintaxe

```
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
```

Modelo de topologia

`node_templates`

Os nós TOSCA da AWS. Os nós possíveis são:

- AWS.VNF
- AWS.Artifacts.Helm

AWS.VNF

Define um nó de função de rede virtual (VNF) da AWS.

Sintaxe

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

Propriedades

`descriptor_id`

O UUID do descritor.

Obrigatório: sim

Tipo: string

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

`descriptor_version`

A versão do VNFD.

Obrigatório: sim

Tipo: string

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

`descriptor_name`

O nome do descritor.

Obrigatório: sim

Tipo: string

`provider`

O autor do VNFD.

Obrigatório: sim

Tipo: string

Requisitos

`helm`

O diretório Helm que define artefatos de contêiner. Essa é uma referência a AWS.Artifacts.Helm.

Obrigatório: sim

Tipo: string

Exemplo

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

AWS.Artifacts.Helm

Define um nó Helm da AWS.

Sintaxe

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

```
  properties:
    implementation: String
```

Propriedades

`implementation`

O diretório local que contém o chart do Helm no pacote CSAR.

Obrigatório: sim

Tipo: string

Exemplo

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

Modelo de NSD

Define um modelo de descritor de serviço de rede (NSD).

Sintaxe

```
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
```

Uso de parâmetros definidos

Quando quiser passar um parâmetro dinamicamente, como o bloco CIDR para o nó VPC, você pode usar a sintaxe `{ get_input: input-parameter-name }` e definir os parâmetros no modelo de NSD. Em seguida, reutilize o parâmetro no mesmo modelo de NSD.

O exemplo a seguir mostra como definir e usar parâmetros:

```
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 }
```

Importação de VNFD

`descriptor_id`

O UUID do descritor.

Obrigatório: sim

Tipo: string

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

`namespace`

O nome exclusivo.

Obrigatório: sim

Tipo: string

Modelo de topologia

`node_templates`

Os possíveis AWS nós do TOSCA são:

- AWS.NS
- AWS.Compute.EKS
- AWS.compute.eks. AuthRole
- AWS.compute.eks ManagedNode
- AWS.compute.eks SelfManagedNode
- AWS.Computação. PlacementGroup
- AWS.Computação. UserData
- AWS.Trabalho em rede. SecurityGroup
- AWS.Trabalho em rede. SecurityGroupEgressRule
- AWS.Trabalho em rede. SecurityGroupIngressRule
- AWS.Resource.Import
- AWS.Networking.ENI
- AWS.HookExecution
- AWS.Trabalho em rede. InternetGateway
- AWS.Trabalho em rede. RouteTable
- AWS.Networking.Subnet
- AWS.Deployment.VNFDeployment

- AWS.Networking.VPC
- AWS.Networking.NATGateway
- AWS.Networking.Route

AWS.NS

Define um nó de serviço de AWS rede (NS).

Sintaxe

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

Propriedades

`descriptor_id`

O UUID do descritor.

Obrigatório: sim

Tipo: string

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

`descriptor_version`

A versão do NSD.

Obrigatório: sim

Tipo: string

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

`descriptor_name`

O nome do descritor.

Obrigatório: sim

Tipo: String

Exemplo

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

AWS.Compute.EKS

Forneça o nome do cluster, a versão desejada do Kubernetes e uma função que permita que o plano de controle do Kubernetes gerencie os recursos necessários para seus NFs. AWS Os plug-ins da interface de rede de contêineres (CNI) Multus estão habilitados. Você pode conectar várias interfaces de rede e aplicar configurações de rede avançadas às funções de rede baseadas no Kubernetes. Você também especifica o acesso ao endpoint do cluster e as sub-redes do seu cluster.

Sintaxe

```
tosca.nodes.AWS.Compute.EKS:
  capabilities:
    multus:
      properties:
        enabled: Boolean
        multus_role: String
    ebs_csi:
      properties:
        enabled: Boolean
        version: String
  properties:
    version: String
    access: String
    cluster_role: String
    tags: List
    ip_family: String
  requirements:
```

```
    subnets: List
```

Capacidades

`multus`

Opcional. Propriedades que definem o uso da interface de rede de contêineres (CNI) Multus.

Se você incluir `multus`, especifique as propriedades `enabled` e `multus_role`.

`enabled`

Indica se o recurso Multus padrão está habilitado.

Obrigatório: Sim

Tipo: booliano

`multus_role`

O perfil do gerenciamento da interface de rede Multus.

Obrigatório: sim

Tipo: String

`ebs_csi`

Propriedades que definem o driver da CSI (Container Storage Interface) do Amazon EBS instalado no cluster do Amazon EKS.

Habilite esse plug-in para usar os nós autogerenciados do Amazon EKS em AWS Outposts AWS Locais Zones ou Regiões da AWS. Para obter mais informações, consulte Driver da CSI do Amazon EBS no Guia do usuário do Amazon EKS.

`enabled`

Indica se o driver padrão da CSI do Amazon EBS está instalado.

Obrigatório: não

Tipo: booliano

`version`

A versão do complemento do driver da CSI do Amazon EBS. A versão deve corresponder a uma das versões retornadas pela DescribeAddonVersionsação. Para obter mais informações, consulte DescribeAddonVersionsa Referência da API Amazon EKS

Obrigatório: não

Tipo: sequência

Propriedades

`version`

A versão do Kubernetes para o cluster. AWS O Telco Network Builder oferece suporte às versões 1.23 a 1.29 do Kubernetes.

Obrigatório: sim

Tipo: String

Valores possíveis: 1,23 | 1,24 | 1,25 | 1,26 | 1,27 | 1,28 | 1,29

`access`

Acesso ao endpoint do cluster.

Obrigatório: sim

Tipo: String

Valores possíveis: `PRIVATE` | `PUBLIC` | `ALL`

`cluster_role`

O perfil do gerenciamento de clusters.

Obrigatório: sim

Tipo: String

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: não

Tipo: lista

`ip_family`

Indica a família de IP para endereços de serviço e pod no cluster.

Valor permitido: `IPv4`, `IPv6`

Valor padrão: `IPv4`

Obrigatório: não

Tipo: sequência

Requisitos

`subnets`

Um nó AWS.Networking.Subnet.

Obrigatório: Sim

Tipo: lista

Exemplo

```
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

Um AuthRole permite que você adicione funções do IAM ao cluster `aws-auth ConfigMap` do Amazon EKS para que os usuários possam acessar o cluster do Amazon EKS usando uma função do IAM.

Sintaxe

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

Propriedades

`role_mappings`

Lista de mapeamentos que definem perfis do IAM que precisam ser adicionadas ao cluster `aws-auth ConfigMap` do Amazon EKS.

`arn`

O ARN do perfil do IAM.

Obrigatório: sim

Tipo: String

`groups`

Grupos do Kubernetes a serem atribuídos ao perfil definido em `arn`.

Obrigatório: não

Tipo: lista

Requisitos

`clusters`

Um nó AWS.Compute.EKS.

Obrigatório: Sim

Tipo: lista

Exemplo

```
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 O TNB oferece suporte a grupos de nós gerenciados do EKS para automatizar o provisionamento e o gerenciamento do ciclo de vida dos nós (instâncias do Amazon EC2) para clusters do Amazon EKS Kubernetes. Para criar um grupo de nós do EKS, você precisa escolher as imagens de máquina da Amazon (AMI) para os nós de processamento do cluster fornecendo o ID ou o tipo da AMI. Você também fornece um par de chaves do Amazon EC2 para acesso SSH e as propriedades de escalabilidade para seu grupo de nós. Seu grupo de nós precisa estar associado a um cluster do EKS. Você precisa fornecer as sub-redes dos nós de processamento.

Opcionalmente, você pode anexar grupos de segurança, rótulos de nós e um grupo de posicionamento ao seu grupo de nós.

Sintaxe

```
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
```

Capacidades

`compute`

Propriedades que definem os parâmetros de computação para o grupo de nós gerenciados do Amazon EKS, como tipos de instância do Amazon EC2 e AMIs de instância do Amazon EC2.

`ami_type`

O tipo de AMI compatível com o Amazon EKS.

Obrigatório: sim

Tipo: String

Valores possíveis: `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`

O ID da AMI.

Obrigatório: não

Tipo: sequência

> **Note**
>
> Se ambos `ami_type` `ami_id` forem especificados no modelo, o AWS TNB usará somente o `ami_id` valor para criar`EKSManagedNode`.

`instance_types`

O tamanho da instância.

Obrigatório: Sim

Tipo: lista

`key_pair`

O par de chaves do EC2 para habilitar o acesso SSH.

Obrigatório: sim

Tipo: String

`root_volume_encryption`

Ativa a criptografia do Amazon EBS para o volume raiz do Amazon EBS. Se essa propriedade não for fornecida, o AWS TNB criptografará os volumes raiz do Amazon EBS por padrão.

Obrigatório: não

Padrão: True

Tipo: booliano

`root_volume_encryption_key_arn`

O ARN da chave. AWS KMS AWS O TNB suporta ARN de chave regular, ARN de chave multirregional e ARN de alias.

Obrigatório: não

Tipo: sequência

> **Note**
>
> - Se `root_volume_encryption` for falso, não inclua`root_volume_encryption_key_arn`.
> - AWS O TNB suporta criptografia de volume raiz de AMIs suportadas pelo Amazon EBS.
> - Se o volume raiz da AMI já estiver criptografado, você deverá incluir o `root_volume_encryption_key_arn` para que o AWS TNB recriptografe o volume raiz.
> - Se o volume raiz da AMI não estiver criptografado, o AWS TNB usará o `root_volume_encryption_key_arn` para criptografar o volume raiz.
>
>   Se você não incluir`root_volume_encryption_key_arn`, o AWS TNB usa a chave padrão fornecida por AWS Key Management Service para criptografar o volume raiz.
> - AWS O TNB não descriptografa uma AMI criptografada.

`scaling`

Propriedades que definem os parâmetros de escalabilidade para o grupo de nós gerenciados do Amazon EKS, como o número desejado de instâncias do Amazon EC2 e os números mínimo e máximo de instâncias do Amazon EC2 no grupo de nós.

`desired_size`

O número de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: inteiro

`min_size`

O número mínimo de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: inteiro

`max_size`

O número máximo de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: inteiro

Propriedades

`node_role`

O ARN do perfil do IAM anexado à instância do Amazon EC2.

Obrigatório: sim

Tipo: String

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: não

Tipo: lista

Requisitos

`cluster`

Um nó AWS.Compute.EKS.

Obrigatório: sim

Tipo: String

`subnets`

Um nó AWS.Networking.Subnet.

Obrigatório: Sim

Tipo: lista

`network_interfaces`

Um nó AWS.Networking.ENI. Certifique-se de que as interfaces de rede e sub-redes estejam definidas com a mesma zona de disponibilidade, senão a instanciação falhará.

Quando você configura`network_interfaces`, o AWS TNB obtém a permissão relacionada aos ENIs da `multus_role` propriedade se você incluiu a `multus` propriedade no nó aws.compute.eks. Caso contrário, o AWS TNB recebe a permissão relacionada a ENIs da propriedade node_role.

Obrigatório: não

Tipo: lista

`security_groups`

Um AWS.Networking. SecurityGroupnodo.

Obrigatório: não

Tipo: lista

`placement_group`

Um tosca.nodes.AWS.Computação. PlacementGroupnodo.

Obrigatório: não

Tipo: sequência

`user_data`

Um tosca.nodes.AWS.Computação. UserDatareferência de nó. Um script de dados de usuário é transmitido às instâncias do Amazon EC2 iniciadas pelo grupo de nós gerenciados. Adicione as permissões necessárias para executar dados de usuário personalizados no node_role transmitido ao grupo de nós.

Obrigatório: não

Tipo: sequência

`labels`

Uma lista de rótulos de nós. Um rótulo de nó deve ter um nome e um valor. Crie um rótulo usando os seguintes critérios:

- O nome e o valor devem ser separados por=.
- O nome e o valor podem ter, cada um, até 63 caracteres.
- O rótulo pode incluir letras (A-Z, a-z), números (0-9) e os seguintes caracteres: `[-, _, ., *, ?]`
- O nome e o valor devem começar e terminar com um caractere alfanumérico ou `*` caractere. `?`

Por exemplo, `myLabelName1=*NodeLabelValue1`.

Obrigatório: não

Tipo: lista

Exemplo

```
SampleEKSManagedNode:
  type: tosca.nodes.AWS.Compute.EKSManagedNode
  capabilities:
    compute:
      properties:
        ami_type: "AL2_x86_64"
        instance_types:
          - "t3.xlarge"
        key_pair: "SampleKeyPair"
        root_volume_encryption: true
        root_volume_encryption_key_arn: "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    scaling:
      properties:
        desired_size: 1
        min_size: 1
        max_size: 1
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
```

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

AWS.compute.eks SelfManagedNode

AWS O TNB oferece suporte aos nós autogerenciados do Amazon EKS para automatizar o provisionamento e o gerenciamento do ciclo de vida dos nós (instâncias do Amazon EC2) para clusters do Amazon EKS Kubernetes. Para criar um grupo de nós do Amazon EKS, você precisa escolher as imagens de máquina da Amazon (AMI) para os nós de processamento do cluster fornecendo o ID da AMI. Opcionalmente, forneça um par de chaves do Amazon EC2 para acesso SSH. Você também precisa fornecer o tipo de instância e os tamanhos mínimo e máximo desejados. O grupo de nós precisa estar associado a um cluster do Amazon EKS. Você precisa fornecer as sub-redes dos nós de processamento.

Opcionalmente, você pode anexar grupos de segurança, rótulos de nós e um grupo de posicionamento ao seu grupo de nós.

Sintaxe

```
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
```

Capacidades

compute

Propriedades que definem os parâmetros de computação para os nós autogerenciados do Amazon EKS, como tipos de instância do Amazon EC2 e AMIs de instância do Amazon EC2.

`ami_id`

O ID da AMI usado para iniciar a instância. AWS O TNB oferece suporte a instâncias que utilizam o IMDSv2. Para ter mais informações, consulte Versão do IMDS.

Obrigatório: sim

Tipo: String

`instance_type`

O tamanho da instância.

Obrigatório: sim

Tipo: String

`key_pair`

O par de chaves do Amazon EC2 para permitir o acesso SSH.

Obrigatório: sim

Tipo: String

`root_volume_encryption`

Ativa a criptografia do Amazon EBS para o volume raiz do Amazon EBS. Se essa propriedade não for fornecida, o AWS TNB criptografará os volumes raiz do Amazon EBS por padrão.

Obrigatório: não

Padrão: True

Tipo: booliano

`root_volume_encryption_key_arn`

O ARN da chave. AWS KMS AWS O TNB suporta ARN de chave regular, ARN de chave multirregional e ARN de alias.

Obrigatório: não

Tipo: sequência

> **Note**
>
> - Se `root_volume_encryption` for falso, não inclua`root_volume_encryption_key_arn`.
> - AWS O TNB suporta criptografia de volume raiz de AMIs suportadas pelo Amazon EBS.
> - Se o volume raiz da AMI já estiver criptografado, você deverá incluir o `root_volume_encryption_key_arn` para que o AWS TNB recriptografe o volume raiz.
> - Se o volume raiz da AMI não estiver criptografado, o AWS TNB usará o `root_volume_encryption_key_arn` para criptografar o volume raiz.
>
>   Se você não incluir`root_volume_encryption_key_arn`, o AWS TNB usa AWS Managed Services para criptografar o volume raiz.

- AWS O TNB não descriptografa uma AMI criptografada.

scaling

Propriedades que definem os parâmetros de escalabilidade para os nós autogerenciados do Amazon EKS, como o número desejado de instâncias do Amazon EC2 e os números mínimo e máximo de instâncias do Amazon EC2 no grupo de nós.

`desired_size`

O número de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: inteiro

`min_size`

O número mínimo de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: inteiro

`max_size`

O número máximo de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: inteiro

Propriedades

`node_role`

O ARN do perfil do IAM anexado à instância do Amazon EC2.

Obrigatório: sim

Tipo: String

`tags`

As tags a serem anexadas ao recurso. As tags serão propagadas para as instâncias criadas pelo recurso.

Obrigatório: não

Tipo: lista

Requisitos

`cluster`

Um nó AWS.Compute.EKS.

Obrigatório: sim

Tipo: String

`subnets`

Um nó AWS.Networking.Subnet.

Obrigatório: Sim

Tipo: lista

`network_interfaces`

Um nó AWS.Networking.ENI. Certifique-se de que as interfaces de rede e sub-redes estejam definidas com a mesma zona de disponibilidade, senão a instanciação falhará.

Quando você configura`network_interfaces`, o AWS TNB obtém a permissão relacionada aos ENIs da `multus_role` propriedade se você incluiu a `multus` propriedade no nó aws.compute.eks. Caso contrário, o AWS TNB recebe a permissão relacionada a ENIs da propriedade node_role.

Obrigatório: não

Tipo: lista

`security_groups`

Um AWS.Networking. SecurityGroupnodo.

Obrigatório: não

Tipo: lista

`placement_group`

Um tosca.nodes.AWS.Computação. PlacementGroupnodo.

Obrigatório: não

Tipo: sequência

`user_data`

Um tosca.nodes.AWS.Computação. UserDatareferência de nó. Um script de dados de usuário é transmitido às instâncias do Amazon EC2 iniciadas pelo grupo de nós autogerenciados. Adicione as permissões necessárias para executar dados de usuário personalizados no node_role transmitido ao grupo de nós.

Obrigatório: não

Tipo: sequência

`labels`

Uma lista de rótulos de nós. Um rótulo de nó deve ter um nome e um valor. Crie um rótulo usando os seguintes critérios:

- O nome e o valor devem ser separados por=.
- O nome e o valor podem ter, cada um, até 63 caracteres.
- O rótulo pode incluir letras (A-Z, a-z,), números (0-9) e os seguintes caracteres: `[-, _, ., *, ?]`
- O nome e o valor devem começar e terminar com um caractere alfanumérico ou `*` caractere. `?`

Por exemplo, `myLabelName1=*NodeLabelValue1`.

Obrigatório: não

Tipo: lista

Exemplo

```
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.Computação. PlacementGroup

Um PlacementGroup nó oferece suporte a diferentes estratégias para colocar instâncias do Amazon EC2.

Ao executar uma nova instância do Amazon EC2, o serviço do Amazon EC2 tenta posicionar a instância de forma que todas as instâncias sejam distribuídas pelo hardware subjacente para minimizar falhas correlacionadas. É possível usar grupos de posicionamento para influenciar o

posicionamento de um grupo de instâncias interdependentes para atender às necessidades de sua workload.

Sintaxe

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

Propriedades

`strategy`

A estratégia a ser usada para posicionar instâncias do Amazon EC2.

Obrigatório: sim

Tipo: String

Valores possíveis: CLUSTER | PARTITION | SPREAD_HOST | SPREAD_RACK

- CLUSTER: agrupa as instâncias em uma zona de disponibilidade. Essa estratégia permite que as cargas de trabalho alcancem o desempenho de rede de baixa latência necessário para uma node-to-node comunicação fortemente acoplada, típica dos aplicativos de computação de alto desempenho (HPC).
- PARTITION: distribui as instâncias entre partições lógicas, de tal modo que as instâncias em uma partição não compartilhem o hardware subjacente com os grupos de instâncias em outras partições. Essa estratégia é normalmente usada por grandes workloads distribuídas e replicadas, como Hadoop, Cassandra e Kafka.
- SPREAD_RACK: posiciona um pequeno grupo de instâncias no hardware subjacente distinto para reduzir falhas correlacionadas.
- SPREAD_HOST: usado somente com grupos de posicionamento do Outpost. Posiciona um pequeno grupo de instâncias no hardware subjacente distinto para reduzir falhas correlacionadas.

`partition_count`

O número de partições.

Obrigatório: obrigatório somente quando `strategy` é definido como `PARTITION`.

Tipo: inteiro

Valores possíveis: 1 | 2 | 3 | 4 | 5 | 6 | 7

`tags`

As tags que você pode anexar ao recurso de grupo de posicionamento.

Obrigatório: não

Tipo: lista

Exemplo

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

AWS.Computação. UserData

AWS O TNB oferece suporte ao lançamento de instâncias do Amazon EC2 com dados personalizados do usuário, por meio do nó UserData no Network Service Descriptor (NSD). Consulte mais informações sobre dados de usuário personalizados em Dados de usuário e scripts de shell no Guia do usuário do Amazon EC2 para instâncias do Linux.

Durante a instanciação da rede, o AWS TNB fornece o registro da instância do Amazon EC2 para o cluster por meio de um script de dados do usuário. Quando dados personalizados do usuário também são fornecidos, o AWS TNB mescla os dois scripts e os transmite como um script multimime para o Amazon EC2. O script de dados de usuário personalizado é executado antes do script de registro do Amazon EKS.

Para usar variáveis personalizadas no script de dados de usuário, adicione um ponto de exclamação `!` após o colchete aberto `{`. Por exemplo, para usar `MyVariable` no script, insira: `{!MyVariable}`

> **Note**
>
> - AWS O TNB suporta scripts de dados do usuário de até 7 KB de tamanho.

- Como o AWS TNB usa AWS CloudFormation para processar e renderizar o script de `multimime` dados do usuário, certifique-se de que o script cumpra todas as regras. AWS CloudFormation

Sintaxe

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

Propriedades

`implementation`

O caminho relativo para a definição do script de dados de usuário. O formato precisa ser: `./scripts/script_name.sh`

Obrigatório: sim

Tipo: String

`content_type`

Tipo de conteúdo do script de dados de usuário.

Obrigatório: sim

Tipo: String

Valores possíveis: `x-shellscript`

Exemplo

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

AWS.Trabalho em rede. SecurityGroup

AWS O TNB oferece suporte a grupos de segurança para automatizar o provisionamento de grupos de segurança do Amazon EC2, que você pode anexar aos grupos de nós do cluster Amazon EKS Kubernetes.

Sintaxe

```
tosca.nodes.AWS.Networking.SecurityGroup
  properties:
    description: String
    name: String
    tags: List
  requirements:
    vpc: String
```

Propriedades

`description`

Descrição do grupo de segurança. Podem ser usados até 255 caracteres para descrever o grupo. Só é possível incluir letras (A-Z e a-z), números (0-9), espaços e os seguintes caracteres especiais: _-:/()#,@[]+=&;{}!$*

Obrigatório: sim

Tipo: String

`name`

Um nome para o grupo de segurança. Você pode usar até 255 caracteres para o nome. Só é possível incluir letras (A-Z e a-z), números (0-9), espaços e os seguintes caracteres especiais: _-:/()#,@[]+=&;{}!$*

Obrigatório: sim

Tipo: String

`tags`

As tags que você pode anexar ao recurso de grupo de segurança.

Obrigatório: não

Tipo: lista

Requisitos

`vpc`

Um nó AWS.Networking.VPC.

Obrigatório: sim

Tipo: String

Exemplo

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

AWS.Trabalho em rede. SecurityGroupEgressRule

AWS O TNB suporta regras de saída de grupos de segurança para automatizar o provisionamento das regras de saída de grupos de segurança do Amazon EC2, que podem ser anexadas ao .Networking. AWS SecurityGroup. Observe que você precisa fornecer um cidr_ip/ destination_security_group/destination_prefix_list como destino para o tráfego de saída.

Sintaxe

```
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
```

Propriedades

`cidr_ip`

O intervalo de endereços IPv4, no formato CIDR. Você precisa especificar um intervalo CIDR que permita o tráfego de saída.

Obrigatório: não

Tipo: sequência

`cidr_ipv6`

O intervalo de endereços IPv6 no formato CIDR, para tráfego de saída. Você deve especificar um grupo de segurança de destino (`destination_security_group` ou `destination_prefix_list`) ou um intervalo de CIDR (`cidr_ip` ou `cidr_ipv6`).

Obrigatório: não

Tipo: sequência

`description`

A descrição de uma regra de saída de grupos de segurança. Podem ser usados até 255 caracteres para descrever a regra.

Obrigatório: não

Tipo: sequência

`destination_prefix_list`

O ID da lista de prefixos de uma lista de prefixos gerenciada do Amazon VPC. Esse é o destino das instâncias do grupo de nós associadas ao grupo de segurança. Para obter mais informações sobre listas de prefixos gerenciadas, consulte Listas de prefixos gerenciados no Guia do usuário da Amazon VPC.

Obrigatório: não

Tipo: sequência

`from_port`

Se o protocolo for TCP ou UDP, esse será o início do intervalo de portas. Se o protocolo for ICMP ou ICMPv6, esse será o número do tipo. Um valor -1 indica todos os tipos de ICMP/ICMPv6. Se você especificar todos os tipos de ICMP/ICMPv6, deverá especificar todos os códigos ICMP/ICMPv6.

Obrigatório: não

Tipo: inteiro

`ip_protocol`

O nome do protocolo IP (tcp, udp, icmp, icmpv6) ou o número do protocolo. Use -1 para especificar todos os protocolos. Ao autorizar regras de grupo de segurança, especificar -1 ou um número de protocolo diferente de tcp, udp, icmp ou icmpv6 permitirá o tráfego em todas as portas, seja qual for o intervalo de portas especificado. Para tcp, udp e icmp, você precisa especificar um intervalo de portas. Para icmpv6, o intervalo de portas é opcional. Se você omiti-lo, o tráfego de todos os tipos e códigos será permitido.

Obrigatório: sim

Tipo: String

`to_port`

Se o protocolo for TCP ou UDP, esse será o fim do intervalo de portas. Se o protocolo for ICMP ou ICMPv6, esse será o código. Um valor -1 indica todos os códigos ICMP/ICMPv6. Se você especificar todos os tipos de ICMP/ICMPv6, deverá especificar todos os códigos ICMP/ICMPv6.

Obrigatório: não

Tipo: inteiro

Requisitos

`security_group`

O ID do grupo de segurança ao qual essa regra deve ser adicionada.

Obrigatório: sim

Tipo: String

`destination_security_group`

O ID ou referência TOSCA do grupo de segurança de destino para o qual o tráfego de saída é permitido.

Obrigatório: não

Tipo: sequência

Exemplo

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

AWS.Trabalho em rede. SecurityGroupIngressRule

AWS O TNB suporta regras de entrada de grupos de segurança para automatizar o provisionamento das regras de entrada de grupos de segurança do Amazon EC2, que podem ser anexadas ao .Networking. AWS SecurityGroup. Você precisa fornecer um cidr_ip/source_security_group/ source_prefix_list como origem do tráfego de entrada.

Sintaxe

```
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
```

Propriedades

`cidr_ip`

O intervalo de endereços IPv4, no formato CIDR. Você precisa especificar um intervalo CIDR que permita o tráfego de entrada.

Obrigatório: não

Tipo: sequência

`cidr_ipv6`

O intervalo de endereços IPv6 no formato CIDR, para tráfego de entrada. Você deve especificar um grupo de segurança de origem (`source_security_group` ou `source_prefix_list`) ou um intervalo de CIDR (`cidr_ip` ou `cidr_ipv6`).

Obrigatório: não

Tipo: sequência

`description`

A descrição de uma regra de entrada de grupo de segurança. Podem ser usados até 255 caracteres para descrever a regra.

Obrigatório: não

Tipo: sequência

`source_prefix_list`

O ID da lista de prefixos de uma lista de prefixos gerenciada do Amazon VPC. Essa é a fonte da qual as instâncias do grupo de nós associadas ao grupo de segurança poderão receber tráfego. Para obter mais informações sobre listas de prefixos gerenciadas, consulte Listas de prefixos gerenciados no Guia do usuário da Amazon VPC.

Obrigatório: não

Tipo: sequência

`from_port`

Se o protocolo for TCP ou UDP, esse será o início do intervalo de portas. Se o protocolo for ICMP ou ICMPv6, esse será o número do tipo. Um valor -1 indica todos os tipos de ICMP/ICMPv6. Se você especificar todos os tipos de ICMP/ICMPv6, deverá especificar todos os códigos ICMP/ICMPv6.

Obrigatório: não

Tipo: inteiro

`ip_protocol`

O nome do protocolo IP (tcp, udp, icmp, icmpv6) ou o número do protocolo. Use -1 para especificar todos os protocolos. Ao autorizar regras de grupo de segurança, especificar -1 ou um número de protocolo diferente de tcp, udp, icmp ou icmpv6 permitirá o tráfego em todas as portas, seja qual for o intervalo de portas especificado. Para tcp, udp e icmp, você precisa especificar um intervalo de portas. Para icmpv6, o intervalo de portas é opcional. Se você omiti-lo, o tráfego de todos os tipos e códigos será permitido.

Obrigatório: sim

Tipo: String

`to_port`

Se o protocolo for TCP ou UDP, esse será o fim do intervalo de portas. Se o protocolo for ICMP ou ICMPv6, esse será o código. Um valor -1 indica todos os códigos ICMP/ICMPv6. Se você especificar todos os tipos de ICMP/ICMPv6, deverá especificar todos os códigos ICMP/ICMPv6.

Obrigatório: não

Tipo: inteiro

Requisitos

`security_group`

O ID do grupo de segurança ao qual essa regra deve ser adicionada.

Obrigatório: sim

Tipo: String

`source_security_group`

O ID ou referência TOSCA do grupo de segurança de origem do qual o tráfego de entrada deve ser permitido.

Obrigatório: não

Tipo: sequência

Exemplo

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

AWS.Resource.Import

Você pode importar os seguintes AWS recursos para o AWS TNB:

- VPC
- Sub-rede
- Tabela de rotas
- Gateway da Internet
- Grupo de segurança

Sintaxe

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

Propriedades

`resource_type`

O tipo de recurso que é importado para o AWS TNB.

Obrigatório: não

Tipo: lista

`resource_id`

O ID do recurso que é importado para o AWS TNB.

Obrigatório: não

Tipo: lista

Exemplo

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

AWS.Networking.ENI

Uma interface de rede é um componente lógico de redes em uma VPC que representa uma cartão de rede virtual. Um endereço IP é atribuído a uma interface de rede de forma automática ou manual, com base em sua sub-rede. Depois de implantar uma instância do Amazon EC2 em uma sub-rede, você pode anexar uma interface de rede a ela ou separar uma interface de rede dessa instância do Amazon EC2 e reconectar-se a outra instância do Amazon EC2 nessa sub-rede. O índice do dispositivo identifica a posição na ordem do anexo.

Sintaxe

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

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

Propriedades

`device_index`

O índice do dispositivo precisa ser maior que zero.

Obrigatório: Sim

Tipo: inteiro

`source_dest_check`

Indica se a interface de rede executa a verificação de origem/destino. O valor `true` significa que a verificação está habilitada e `false` significa que a verificação está desabilitada.

Valor permitido: verdadeiro, falso

Padrão: True

Obrigatório: não

Tipo: booliano

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: não

Tipo: lista

Requisitos

`subnet`

Um nó AWS.Networking.Subnet.

Obrigatório: sim

Tipo: String

`security_groups`

Um AWS.Networking. SecurityGroupnodo.

Obrigatório: não

Tipo: sequência

Exemplo

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

AWS.HookExecution

Um gancho do ciclo de vida fornece a capacidade de executar seus próprios scripts como parte de sua infraestrutura e instanciação de rede.

Sintaxe

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

Capacidades

`execution`

Propriedades do mecanismo de execução de hook que executa os scripts de hook.

`type`

O tipo de mecanismo de execução de hook.

Obrigatório: não

Tipo: sequência

Valores possíveis: `CODE_BUILD`

Requisitos

`definition`

Um AWS. HookDefinition.Modo Bash.

Obrigatório: sim

Tipo: String

`vpc`

Um nó AWS.Networking.VPC.

Obrigatório: sim

Tipo: String

Exemplo

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

AWS.Trabalho em rede. InternetGateway

Define um nó do AWS Internet Gateway.

Sintaxe

```
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
```

Capacidades

`routing`

Propriedades que definem a conexão de roteamento dentro da VPC. Você deve incluir a propriedade `dest_cidr` ou `ipv6_dest_cidr`.

`dest_cidr`

O bloco CIDR IPv4 usado para a correspondência do destino. Essa propriedade é usada para criar uma rota em `RouteTable` e seu valor é usado como `DestinationCidrBlock`.

Obrigatório: não se você incluiu a propriedade `ipv6_dest_cidr`.

Tipo: sequência

`ipv6_dest_cidr`

O bloco CIDR IPv6 usado para a correspondência do destino.

Obrigatório: não se você incluiu a propriedade `dest_cidr`.

Tipo: sequência

Propriedades

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: não

Tipo: lista

`egress_only`

Uma propriedade específica de IPv6. Indica se o gateway da Internet serve apenas para comunicação de saída ou não. Quando `egress_only` é verdadeiro, você deve definir a propriedade `ipv6_dest_cidr`.

Obrigatório: não

Tipo: booliano

Requisitos

`vpc`

Um nó AWS.Networking.VPC.

Obrigatório: sim

Tipo: String

`route_table`

Um AWS.Networking. RouteTablenodo.

Obrigatório: sim

Tipo: String

Exemplo

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

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

AWS.Trabalho em rede. RouteTable

Uma tabela de rotas contém um conjunto de regras, chamado de rotas, que determina para onde o tráfego de rede de sua sub-rede ou gateway é direcionado. Você precisa associar uma tabela de rotas a uma VPC.

Sintaxe

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

Propriedades

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: não

Tipo: lista

Requisitos

`vpc`

Um nó AWS.Networking.VPC.

Obrigatório: sim

Tipo: String

Exemplo

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

AWS.Networking.Subnet

Uma sub-rede é um intervalo de endereços IP na VPC que precisa residir inteiramente em uma zona de disponibilidade. Você precisa especificar uma VPC, um bloco CIDR, uma zona de disponibilidade e uma tabela de rotas para sua sub-rede. Você também precisa definir se sua sub-rede é privada ou pública.

Sintaxe

```
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
```

Propriedades

`type`

Indica se as instâncias executadas nessa sub-rede recebem um endereço IPv4 público.

Obrigatório: sim

Tipo: String

Valores possíveis: `PUBLIC` | `PRIVATE`

`availability_zone`

A zona de disponibilidade da sub-rede. Esse campo é compatível com zonas de AWS disponibilidade em uma AWS região, por exemplo `us-west-2` (Oeste dos EUA (Oregon)). Ele também suporta Zonas AWS Locais dentro da Zona de Disponibilidade, por exemplo`us-west-2-lax-1a`.

Obrigatório: sim

Tipo: String

`cidr_block`

O bloco CIDR da sub-rede.

Obrigatório: não

Tipo: sequência

`ipv6_cidr_block`

O bloco CIDR usado para criar a sub-rede IPv6. Se você incluir essa propriedade, não inclua `ipv6_cidr_block_suffix`.

Obrigatório: não

Tipo: sequência

`ipv6_cidr_block_suffix`

O sufixo hexadecimal de dois dígitos do bloco CIDR IPv6 para a sub-rede criada na Amazon VPC. Use o seguinte formato: *`2-digit hexadecimal`*`::/`*`subnetMask`*.

Se você incluir essa propriedade, não inclua `ipv6_cidr_block`.

Obrigatório: não

Tipo: sequência

`outpost_arn`

O ARN em AWS Outposts que a sub-rede será criada. Adicione essa propriedade ao modelo de NSD se quiser executar nós autogerenciados do Amazon EKS no AWS Outposts. Para obter mais informações, consulte Amazon EKS no AWS Outposts no Guia do usuário do Amazon EKS.

Se você adicionar essa propriedade ao modelo de NSD, precisará definir o valor da propriedade `availability_zone` como a zona de disponibilidade do AWS Outposts.

Obrigatório: não

Tipo: sequência

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: não

Tipo: lista

Requisitos

`vpc`

Um nó AWS.Networking.VPC.

Obrigatório: sim

Tipo: String

`route_table`

Um AWS.Networking. RouteTablenodo.

Obrigatório: sim

Tipo: String

Exemplo

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

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

AWS.Deployment.VNFDeployment

As implantações de NF são modeladas fornecendo a infraestrutura e o aplicativo associado a ele. O atributo cluster especifica o cluster do EKS que vai hospedar seus NFs. O atributo vnfs especifica as funções de rede da sua implantação. Você também pode fornecer operações opcionais de ganchos do ciclo de vida do tipo pre_create e post_create para executar instruções específicas da sua implantação, como chamar uma API do sistema de gerenciamento de inventário.

Sintaxe

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

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

Requisitos

`deployment`

Um nó AWS.Deployment.VNFDeployment.

Obrigatório: não

Tipo: sequência

`cluster`

Um nó AWS.Compute.EKS.

Obrigatório: sim

Tipo: String

`vnfs`

Um nó AWS.VNF.

Obrigatório: sim

Tipo: String

Interfaces

Hooks

Define o estágio em que os ganchos do ciclo de vida são executados.

`pre_create`

Um AWS. HookExecutionnodo. Esse hook é executado antes da implantação do nó `VNFDeployment`.

Obrigatório: não

Tipo: sequência

`post_create`

Um AWS. HookExecutionnodo. Esse hook é executado após a implantação do nó `VNFDeployment`.

Obrigatório: não

Tipo: sequência

Exemplo

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

AWS.Networking.VPC

Você precisa especificar um bloco CIDR para sua nuvem privada virtual (VPC).

Sintaxe

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

Propriedades

`cidr_block`

O intervalo de rede do IPv4 para a VPC, na notação CIDR.

Obrigatório: sim

Tipo: String

`ipv6_cidr_block`

O bloco CIDR IPv6 usado para criar a VPC.

Valor permitido: `AMAZON_PROVIDED`

Obrigatório: não

Tipo: sequência

`dns_support`

Indica se as instâncias executadas na VPC obtêm nomes de host DNS.

Obrigatório: não

Tipo: booliano

Padrão: `false`

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: não

Tipo: lista

Exemplo

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

AWS.Networking.NATGateway

É possível definir um nó público ou privado do NAT Gateway em uma sub-rede. Para um gateway público, se você não fornecer um ID de alocação de IP elástico, o AWS TNB alocará um IP elástico para sua conta e o associará ao gateway.

Sintaxe

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

Propriedades

`subnet`

A referência do nó AWS.Networking.Subnet.

Obrigatório: sim

Tipo: String

`internet_gateway`

O AWS.Networking. InternetGatewayreferência de nó.

Obrigatório: sim

Tipo: String

Propriedades

`type`

Indica se o gateway é público ou privado.

Valor permitido: `PUBLIC`, `PRIVATE`

Obrigatório: sim

Tipo: String

`eip_allocation_id`

O ID que representa a alocação do endereço IP elástico.

Obrigatório: não

Tipo: sequência

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: não

Tipo: lista

Exemplo

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

AWS.Networking.Route

Você pode definir um nó de rota que associe a rota de destino ao NAT Gateway como o recurso de destino e adicione a rota à tabela de rotas associada.

Sintaxe

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

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

Propriedades

`dest_cidr_blocks`

A lista de rotas IPv4 de destino para o recurso de destino.

Obrigatório: Sim

Tipo: lista

Tipo de membro: string

Propriedades

`nat_gateway`

A referência do nó AWS.Networking.NATGateway.

Obrigatório: sim

Tipo: String

`route_table`

O AWS.Networking. RouteTablereferência de nó.

Obrigatório: sim

Tipo: String

Exemplo

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

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

Nós comuns

Define os nós a serem usados no NSD e no VNFD.

- AWS.HookDefinition.Bash

AWS.HookDefinition.Bash

Define uma AWS HookDefinition no `bash`.

Sintaxe

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

Propriedades

`implementation`

O caminho relativo para a definição do hook. O formato precisa ser: `./hooks/`*`script_name`*`.sh`

Obrigatório: sim

Tipo: string

`environment_variables`

As variáveis de ambiente para o script bash do hook. Use o seguinte formato: **`envName`**`=`**`envValue`** com o seguinte regex: `^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`

Certifique-se de que o valor **`envName`**`=`**`envValue`** atenda aos seguintes critérios:

- Não use espaços.
- Comece **`envName`** com uma letra (A-Z ou a-z) ou número (0-9).

- Não inicie o nome da variável de ambiente com as seguintes palavras-chave reservadas do AWS TNB (sem distinção entre maiúsculas e minúsculas):
  - CODEBUILD
  - TNB
  - HOME
  - AWS
- Você pode usar qualquer número de letras (A-Z ou a-z), números (0-9) e os caracteres especiais - e _ para **envName** e **envValue**.

Exemplo: A123-45xYz=Example_789

Obrigatório: não

Tipo: lista

`execution_role`

O perfil da execução do hook.

Obrigatório: sim

Tipo: string

Exemplo

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

Segurança no AWS Telco Network Builder

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como a segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam ao AWS Telco Network Builder, consulte AWS Serviços no escopo do programa de conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o AWS TNB. Os tópicos a seguir mostram como configurar o AWS TNB para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos AWS do TNB.

Conteúdo

- Proteção de dados no AWS TNB
- Gerenciamento de identidade e acesso para AWS TNB
- Validação de conformidade para AWS TNB
- Resiliência no TNB AWS
- Segurança de infraestrutura no AWS TNB
- Versão do IMDS

Proteção de dados no AWS TNB

O modelo de responsabilidade AWS compartilhada se aplica à proteção de dados no AWS Telco Network Builder. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a AWS postagem do blog Shared Responsibility Model and GDPR no AWS Blog de segurança da.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso inclui quando você trabalha com o AWS TNB ou outro Serviços da AWS usando o console, a API ou os AWS SDKs. AWS CLI Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de

diagnóstico. Se você fornecer um URL para um servidor externo, recomendemos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Tratamento de dados

Quando você fecha sua AWS conta, o AWS TNB marca seus dados para exclusão e os remove de qualquer uso. Se você reativar sua AWS conta dentro de 90 dias, o AWS TNB restaurará seus dados. Após 120 dias, o AWS TNB exclui permanentemente seus dados. AWS O TNB também encerra suas redes e exclui seus pacotes de funções e seus pacotes de rede.

Criptografia inativa

AWS O TNB sempre criptografa todos os dados armazenados no serviço em repouso sem exigir nenhuma configuração adicional. Essa criptografia é automática por meio de AWS Key Management Service.

Criptografia em trânsito

AWS O TNB protege todos os dados em trânsito usando o Transport Layer Security (TLS) 1.2.

É sua responsabilidade criptografar os dados entre seus agentes de simulação e os clientes deles.

Privacidade do tráfego entre redes

AWS Os recursos computacionais da TNB residem em uma nuvem privada virtual (VPC) compartilhada por todos os clientes. Todo o tráfego interno AWS do TNB permaneceu na AWS rede e não atravessa a Internet. As conexões entre seus agentes de simulação e os clientes deles são roteadas pela Internet.

Gerenciamento de identidade e acesso para AWS TNB

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos AWS do TNB. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Conteúdo

- Público
- Autenticando com identidades
- Gerenciamento do acesso usando políticas
- Como o AWS Telco Network Builder funciona com o IAM
- Exemplos de políticas baseadas em identidade do AWS Telco Network Builder
- Solução de problemas de identidade e acesso ao AWS Telco Network Builder

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz no AWS TNB.

Usuário do serviço — Se você usar o serviço AWS TNB para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões necessárias. À medida que você usa mais recursos do AWS TNB para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no AWS TNB, consulte Solução de problemas de identidade e acesso ao AWS Telco Network Builder.

Administrador de serviços — Se você é responsável pelos recursos do AWS TNB em sua empresa, provavelmente tem acesso total ao AWS TNB. É seu trabalho determinar quais recursos e recursos AWS do TNB seus usuários do serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como sua empresa pode usar o IAM com o AWS TNB, consulteComo o AWS Telco Network Builder funciona com o IAM.

Administrador do IAM — Se você for administrador do IAM, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso ao AWS TNB. Para ver exemplos de políticas baseadas em identidade do AWS TNB que você pode usar no IAM, consulte. Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

Autenticando com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Assinatura de solicitações de AWS API no Guia do usuário do IAM.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia do usuário do AWS IAM Identity Center . Usar a autenticação multifator (MFA) na AWS no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele pode executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o . AWS IAM Identity Center Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte "What is IAM Identity Center?" (O que é o Centro de Identidade do IAM?) noAWS IAM Identity Center Guia do usuário do .

Grupos e usuários do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos depender de credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e atribuir a esse grupo permissões para administrar atributos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de um perfil) no Guia do usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Você

pode assumir temporariamente uma função do IAM no AWS Management Console trocando de funções. Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para o uso de perfis, consulte Usar perfis do IAM no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidades de terceiros no Guia do usuário do IAM. Se você usar o IAM Identity Center, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de permissões noAWS IAM Identity Center Guia do usuário do .
- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre perfis e políticas baseadas em atributo para acesso entre contas, consulte Como os perfis do IAM diferem das políticas baseadas em atributo no Guia do usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal de chamada, usando um perfil de serviço ou uma função vinculada ao serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões

para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

- Perfil de serviço: um perfil de serviço é um perfil do IAMhttps://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do usuário do IAM.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

- Aplicativos em execução no Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e fazendo AWS CLI solicitações de API. AWS É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Usar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se deseja usar as funções do IAM, consulte Quando criar um perfil do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM a perfis, e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em atributos são políticas em linha que estão localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (perfil ou usuário do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade e dos seus limites de permissões. As políticas baseadas em atributo que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em. AWS Organizations AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte Como os SCPs funcionam no Guia do usuário do AWS Organizations .
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas

substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o AWS Telco Network Builder funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao AWS TNB, saiba quais recursos do IAM estão disponíveis para uso com o AWS TNB.

Recursos do IAM que você pode usar com o AWS Telco Network Builder

Atributo do IAM	AWS Suporte TNB
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações de políticas	Sim
atributos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não
ABAC (etiquetas em políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Não
Funções vinculadas ao serviço	Não

Para ter uma visão de alto nível de como o AWS TNB e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte AWS os serviços que funcionam com o IAM no Guia do usuário do IAM.

Políticas baseadas em identidade para TNB AWS

É compatível com políticas baseadas em identidade	Sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou atributos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Não é possível especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexado. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos da política JSON do IAM no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para TNB AWS

Para ver exemplos de políticas baseadas em identidade do AWS TNB, consulte. Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

Políticas baseadas em recursos dentro do TNB AWS

Oferece suporte a políticas baseadas em recursos	Não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico.

Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em atributo. Adicionar uma entidade principal entre contas à política baseada em atributo é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em atributo conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Ações políticas para a AWS TNB

Oferece suporte a ações de políticas	Sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações do AWS TNB, consulte Ações definidas pelo AWS Telco Network Builder na Referência de Autorização de Serviço.

As ações de política no AWS TNB usam o seguinte prefixo antes da ação:

```
tnb
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

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

Você também pode especificar várias ações usando caracteres-curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `List`, inclua a seguinte ação:

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

Para ver exemplos de políticas baseadas em identidade do AWS TNB, consulte. Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

Recursos políticos para AWS TNB

Oferece suporte a atributos de políticas	Sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando Nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de atributo específico, conhecido como permissões em nível de atributo.

Para ações não compatíveis com permissões no nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de recursos AWS TNB e seus ARNs, consulte Recursos definidos pelo AWS Telco Network Builder na Referência de Autorização de Serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo AWS Telco Network Builder.

Para ver exemplos de políticas baseadas em identidade do AWS TNB, consulte. Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

Chaves de condição de política para AWS TNB

Compatível com chaves de condição de política específicas do serviço	Sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou `Condition` bloco de) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista das chaves de condição AWS TNB, consulte Chaves de condição para o AWS Telco Network Builder na Referência de Autorização de Serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Ações definidas pelo AWS Telco Network Builder.

Para ver exemplos de políticas baseadas em identidade do AWS TNB, consulte. Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

ACLs em TNB AWS

Oferece suporte a ACLs	Não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com TNB AWS

Oferece suporte a ABAC (tags em políticas)	Sim

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do atributo que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial.

Para obter mais informações sobre o ABAC, consulte O que é ABAC? no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Usando credenciais temporárias com AWS o TNB

Oferece suporte a credenciais temporárias	Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS "Trabalhe com o IAM" no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar perfis, consulte Alternar para um perfil (console) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte Credenciais de segurança temporárias no IAM.

Permissões principais entre serviços para TNB AWS

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

Perfis de serviço do AWS TNB

Oferece suporte a perfis de serviço	Não

O perfil de serviço é um perfil do IAMhttps://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html que um serviço assume para realizar ações em seu nome. Um administrador do IAM

pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do usuário do IAM.

Funções vinculadas a serviços para TNB AWS

É compatível com perfis vinculados ao serviço	Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos AWS do TNB. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis, e os usuários podem assumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte Criação de políticas do IAM no Guia do Usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pelo AWS TNB, incluindo o formato dos ARNs para cada um dos tipos de recursos, consulte Ações, recursos e chaves de condição para o AWS Telco Network Builder na Referência de Autorização de Serviço.

Conteúdo

- Melhores práticas de política
- Usando o console AWS TNB
- Exemplos de política de perfil de serviço
- Permitir que os usuários exibam as próprias permissões

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos AWS do TNB em sua conta. Essas ações podem incorrer em custos para a Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para perfis de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e atributos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos de política JSON do IAM: condições no Manual do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações acionáveis para ajudar você a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir a MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do usuário do IAM.

Para mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usando o console AWS TNB

Para acessar o console do AWS Telco Network Builder, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do AWS TNB em seu Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam a operação de API que estiverem tentando executar.

Exemplos de política de perfil de serviço

Como administrador, você possui e gerencia os recursos que o AWS TNB cria, conforme definido pelos modelos de ambiente e serviço. Você deve anexar funções de serviço do IAM à sua conta para permitir que o AWS TNB crie recursos para o gerenciamento do ciclo de vida da sua rede.

Uma função de serviço do IAM permite que o AWS TNB faça chamadas para recursos em seu nome para instanciar e gerenciar suas redes. Se você especificar uma função de serviço, o AWS TNB usará a credencial dessa função.

Você cria o perfil de serviço e a respectiva política de permissão com o serviço do IAM. Para obter mais informações sobre a criação de uma função de serviço, consulte Criação de uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

AWS Função de serviço da TNB

Como membro da equipe da plataforma, você pode, como administrador, criar uma função de serviço da AWS TNB e fornecê-la à AWS TNB. Essa função permite que o AWS TNB faça chamadas para outros serviços, como o Amazon Elastic Kubernetes AWS CloudFormation Service, provisione a infraestrutura necessária para sua rede e provisione funções de rede conforme definido em seu NSD.

Recomendamos que você use o seguinte perfil do IAM e a política de confiança para seu perfil de serviço do AWS TNB. Ao definir o escopo da permissão nesta política, lembre-se de que o AWS TNB pode falhar com erros de acesso negado em relação a recursos decodificados de sua política.

O código a seguir mostra uma política de função de serviço do 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"
        }
    ]
}
```

O código a seguir mostra a política de confiança do serviço 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 Função de serviço TNB para o cluster Amazon EKS

Ao criar um recurso do Amazon EKS em seu NSD, você fornece o atributo `cluster_role` para especificar qual perfil será usado para criar seu cluster do Amazon EKS.

O exemplo a seguir mostra um AWS CloudFormation modelo que cria uma função de serviço AWS TNB para a política de cluster do 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"
```

Para obter mais informações sobre as funções do IAM usando o AWS CloudFormation modelo, consulte as seções a seguir no Guia AWS CloudFormation do usuário:

- AWS::IAM::Role
- Seleção de um modelo de pilha

AWS Função de serviço TNB para o grupo de nós Amazon EKS

Ao criar recursos de um grupo de nós do Amazon EKS em seu NSD, você fornece o atributo `node_role` para especificar qual perfil será usado para criar seu grupo de nós do Amazon EKS.

O exemplo a seguir mostra um AWS CloudFormation modelo que cria uma função de serviço AWS TNB para a política de grupo de nós do 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/*"
```

Para obter mais informações sobre as funções do IAM usando o AWS CloudFormation modelo, consulte as seções a seguir no Guia AWS CloudFormation do usuário:

- AWS::IAM::Role
- Seleção de um modelo de pilha

AWS Função de serviço TNB para Multus

Ao criar um recurso do Amazon EKS em seu NSD, se você quiser gerenciar o Multus como parte do seu modelo de implantação, deverá fornecer o atributo `multus_role` para especificar qual perfil será usado para gerenciar o Multus.

O exemplo a seguir mostra um AWS CloudFormation modelo que cria uma função de serviço AWS TNB para uma política 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: "*"
```

Para obter mais informações sobre as funções do IAM usando o AWS CloudFormation modelo, consulte as seções a seguir no Guia AWS CloudFormation do usuário:

- AWS::IAM::Role
- Seleção de um modelo de pilha

AWS Função de serviço da TNB para uma política de gancho de ciclo de vida

Quando seu pacote de perfis de rede ou NSD usa um hook de ciclo de vida, você precisa de um perfil de serviço que permita criar um ambiente para a execução de seus hooks de ciclo de vida.

> **Note**
>
> Sua política de gancho do ciclo de vida deve ser baseada no que seu gancho de ciclo de vida está tentando fazer.

O exemplo a seguir mostra um AWS CloudFormation modelo que cria uma função de serviço AWS TNB para uma política de gancho de ciclo de vida.

```
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"
```

Para obter mais informações sobre as funções do IAM usando o AWS CloudFormation modelo, consulte as seções a seguir no Guia AWS CloudFormation do usuário:

- AWS::IAM::Role
- Seleção de um modelo de pilha

Permitir que os usuários exibam as próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

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

Solução de problemas de identidade e acesso ao AWS Telco Network Builder

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com o AWS TNB e o IAM.

Problemas

- Não estou autorizado a realizar uma ação no AWS TNB

- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha acessem meus Conta da AWS recursos AWS do TNB

Não estou autorizado a realizar uma ação no AWS TNB

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, é preciso atualizar suas políticas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para exibir detalhes sobre um recurso do *`my-example-widget`* fictício, mas não tem as permissões fictícias do `tnb:GetWidget`.

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

Nesse caso, a política de Mateo deve ser atualizada para permitir que ele tenha acesso ao recurso *`my-example-widget`* usando a ação `tnb:GetWidget`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você passe uma função para a AWS TNB.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro de exemplo a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no AWS TNB. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha acessem meus Conta da AWS recursos AWS do TNB

Você pode criar uma função que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços compatíveis com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o AWS TNB oferece suporte a esses recursos, consulteComo o AWS Telco Network Builder funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Validação de conformidade para AWS TNB

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

> **Note**
>
> Nem todos Serviços da AWS são elegíveis para a HIPAA. Para obter mais informações, consulte a Referência dos serviços qualificados pela HIPAA.

- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Guias de conformidade do cliente — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os atributos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.
- AWS Audit Manager— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência no TNB AWS

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte Infraestrutura AWS global.

AWS O TNB executa seu serviço de rede em clusters EKS em uma nuvem privada virtual (VPC) AWS na região que você escolher.

Segurança de infraestrutura no AWS TNB

Como um serviço gerenciado, o AWS Telco Network Builder é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o AWS TNB pela rede. Os clientes devem ser compatíveis com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com sigilo de encaminhamento perfeito (perfect forward secrecy, ou PFS) como DHE (Ephemeral Diffie-Hellman, ou Efêmero Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman, ou Curva elíptica efêmera Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, são compatíveis com esses modos.

Além disso, as solicitações devem ser assinadas utilizando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Estes são alguns exemplos de responsabilidades compartilhadas:

- AWS é responsável por proteger os componentes que suportam o AWS TNB, incluindo:
  - Instâncias de computação (também conhecidas como trabalhadores)
  - Bancos de dados internos
  - Comunicações de rede entre componentes internos
  - A interface de programação de aplicativos (API) AWS TNB
  - AWS Kits de desenvolvimento de software (SDK)
- Você é responsável por proteger seu acesso aos seus AWS recursos e aos componentes da carga de trabalho, incluindo (mas não se limitando a):
  - Usuários, grupos, perfis e políticas do IAM
  - Buckets S3 que você usa para armazenar seus dados para TNB AWS
  - Outros recursos Serviços da AWS e recursos que você usa para oferecer suporte ao serviço de rede que você provisionou por meio do TNB AWS
  - Código da sua aplicação
  - Conexões entre o serviço de rede que você provisionou por meio do AWS TNB e seus clientes

> **Important**
>
> Você é responsável por implementar um plano de recuperação de desastres que possa efetivamente recuperar um serviço de rede que você provisionou por meio do AWS TNB.

Modelo de segurança de conectividade de rede

Os serviços de rede que você provisiona por meio AWS do TNB são executados em instâncias de computação em uma nuvem privada virtual (VPC) localizada em uma AWS região selecionada por você. Uma VPC é uma rede virtual na AWS nuvem, que isola a infraestrutura por carga de trabalho ou entidade organizacional. A comunicação entre instâncias de computação nas VPCs permanece dentro da rede AWS e não trafega pela Internet. Algumas comunicações internas de serviços cruzam a Internet e são criptografadas. Os serviços de rede provisionados por meio AWS do TNB para todos os clientes que operam na mesma região compartilham a mesma VPC. Os serviços de rede provisionados por meio AWS do TNB para diferentes clientes usam instâncias de computação separadas na mesma VPC.

As comunicações entre seus clientes de serviços de rede e seu serviço de rede no AWS TNB atravessam a Internet. AWS O TNB não gerencia essas conexões. É sua responsabilidade proteger as conexões de seus clientes.

Suas conexões com o AWS TNB por meio de AWS Management Console, AWS Command Line Interface (AWS CLI) e AWS SDKs são criptografadas.

Versão do IMDS

AWS O TNB oferece suporte a instâncias que utilizam o Instance Metadata Service versão 2 (IMDSv2), um método orientado a sessões. O IMDSv2 inclui maior segurança do que o IMDSV1. Para obter mais informações, consulte Adicionar defesa profunda contra firewalls abertos, proxies reversos e vulnerabilidades SSRF com melhorias no Serviço de metadados da instância do Amazon EC2.

Ao executar sua instância, você precisa usar o IMDSv2. Para obter mais informações sobre o IMDSv2, consulte Usar o IMDSv2, no Guia do usuário do Amazon EC2 para instâncias Linux.

Monitoramento do AWS TNB

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do AWS TNB e das outras soluções da AWS. A AWS fornece o AWS CloudTrail para observar o AWS TNB, informar quando algo está errado e realizar ações automáticas quando apropriado.

Use o CloudTrail para capturar informações detalhadas sobre as chamadas feitas às AWS APIs. Você pode armazenar essas chamadas como arquivos de log no Amazon S3. Você pode usar esses logs do CloudTrail para determinar informações do tipo: quais chamadas foram feitas, o endereço IP de origem da chamada, quem fez a chamada e quando ela foi feita.

Os logs do CloudTrail contêm informações sobre as chamadas às ações de API do AWS TNB. Eles também contêm informações para chamadas a ações de API de serviços como Amazon EC2 e Amazon EBS.

Registro de chamadas de API do AWS Telco Network Builder usando a AWS CloudTrail

O AWS Telco Network Builder é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, perfil ou serviço da AWS no AWS TNB. O CloudTrail captura as chamadas de API do AWS TNB como eventos. As chamadas capturadas incluem as chamadas do console do AWS TNB e as chamadas de código às operações da API do AWS TNB. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o AWS TNB. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Histórico de eventos. Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o AWS TNB, o endereço IP no qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, além de outros detalhes.

Para saber mais sobre o CloudTrail, consulte o Guia do usuário do AWS CloudTrail.

Informações do AWS TNB no CloudTrail

O CloudTrail é habilitado em sua Conta da AWS quando ela é criada. Quando ocorre uma atividade no AWS TNB, ela é registrada em um evento do CloudTrail junto com outros eventos de serviço da AWS em Histórico de eventos. Você pode exibir, pesquisar e baixar eventos recentes em sua Conta

da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos na sua Conta da AWS, incluindo eventos para o AWS TNB, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros serviços da AWS para analisar mais ainda mais e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- Serviços e integrações compatíveis com o CloudTrail
- Configurar notificações do Amazon SNS para o CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões e Receber arquivos de log do CloudTrail de várias contas

Todas as ações do AWS TNB são registradas pelo CloudTrail e estão documentadas na Referência de API do AWS Telco Network Builder. Por exemplo, as chamadas às ações `CreateSolFunctionPackage`, `CreateSolNetworkInstance` e `CreateSolNetworkPackage` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail.

Noções básicas das entradas de arquivos de log do AWS TNB

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log.

Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateSolFunctionPackage`.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:example",
        "arn": "arn:aws:sts::111222333444:assumed-role/example/user",
        "accountId": "111222333444",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111222333444:role/example",
                "accountId": "111222333444",
                "userName": "example"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-02T01:42:39Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-02-02T01:43:17Z",
    "eventSource": "tnb.amazonaws.com",
    "eventName": "CreateSolFunctionPackage",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "XXX.XXX.XXX.XXX",
    "userAgent": "userAgent",
    "requestParameters": null,
    "responseElements": {
        "vnfPkgArn": "arn:aws:tnb:us-east-1:111222333444:function-package/
fp-12345678abcEXAMPLE",
        "id": "fp-12345678abcEXAMPLE",
        "operationalState": "DISABLED",
```

```
        "usageState": "NOT_IN_USE",
        "onboardingState": "CREATED"
    },
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111222333444",
    "eventCategory": "Management"
}
```


AWS Tarefas de implantação do TNB

Entenda as tarefas de implantação para monitorar efetivamente as implantações e agir com mais rapidez.

A tabela a seguir lista as tarefas de implantação AWS do TNB:

Nome da tarefa para implantações iniciadas antes de 7 de março de 2024	Nome da tarefa para implantações iniciadas em e após 7 de março de 2024	Task description
AppInstallation	ClusterPluginInstall	Instala o plug-in Multus no cluster do Amazon EKS.
AppUpdate	nenhuma mudança no nome	Atualiza as funções de rede que já estão instaladas em uma instância de rede.
-	ClusterPluginUninstall	Desinstala os plug-ins no cluster Amazon EKS.
ClusterStorageClassConfiguration	nenhuma mudança no nome	Configura a classe de armazenamento (driver CSI) em um cluster do Amazon EKS.
FunctionDeletion	nenhuma mudança no nome	Exclui funções de rede dos recursos do AWS TNB.
FunctionInstantiation	FunctionInstall	Implanta funções de rede usando o HELM.

Nome da tarefa para implantações iniciadas antes de 7 de março de 2024	Nome da tarefa para implantações iniciadas em e após 7 de março de 2024	Task description
FunctionUninstallation	FunctionUninstall	Desinstala a função de rede de um cluster do Amazon EKS.
HookExecution	nenhuma mudança no nome	Executa ganchos do ciclo de vida conforme definido no NSD.
InfrastructureCancellation	nenhuma mudança no nome	Cancela um serviço de rede.
InfrastructureInstantation	nenhuma mudança no nome	AWS Provisiona recursos em nome do usuário.
InfrastructureTermination	nenhuma mudança no nome	Desprovisiona AWS recursos invocados por meio do TNB AWS .
InventoryDeregistration	nenhuma mudança no nome	Cancela o registro de AWS recursos do TNB. AWS
KubernetesClusterConfiguration	ClusterConfiguration	Configura o cluster Kubernetes e adiciona funções adicionais do IAM ao Amazon EKS, AuthMap conforme definido no NSD.
NetworkServiceFinalization	nenhuma mudança no nome	Finaliza o serviço de rede e fornece uma atualização do status de sucesso ou falha.
NetworkServiceInstantiation	nenhuma mudança no nome	Inicializa o serviço de rede.
SelfManagedNodesConfiguration	nenhuma mudança no nome	Inicializa nós autogerenciados com o Amazon EKS e o ambiente de gerenciamento do Kubernetes.

Cotas de serviço do AWS Telco Network Builder

As cotas de serviço, também chamadas de limites, correspondem ao número máximo de recursos ou operações de serviço para sua conta da AWS. Para obter mais informações, consulte Service Quotas do AWS em Referência geral da Amazon Web Services.

Veja a seguir as cotas de serviço do AWS TNB.

Nome	Padrão	Ajustá	Descrição
Operações simultâneas de serviços de rede contínuas	Cada região compatível: 40	Sim	O número máximo de operações simultâne as de serviços de rede contínuas em uma região.
Pacotes de funções	Cada região compatível: 200	Sim	O número máximo de pacotes de funções em uma região.
Pacotes de rede	Cada região compatível: 40	Sim	O número máximo de pacotes de rede em uma região.
Instâncias do serviço de rede	Cada região compatível: 800	Sim	O número máximo de instâncias do serviço de rede em uma região.

Histórico de documentos do Guia do usuário do AWS TNB

A tabela a seguir descreve os lançamentos da documentação AWS do TNB.

Alteração	Descrição	Data
Nova tarefa e novos nomes de tarefas para tarefas existentes	Uma nova tarefa está disponível. Em 7 de março de 2024, algumas tarefas existentes têm novos nomes para maior clareza.	7 de maio de 2024
Versão do Kubernetes para cluster	AWS O TNB agora oferece suporte às versões 1.29 do Kubernetes para criar clusters do Amazon EKS.	10 de abril de 2024
Support para interface de rede `security_groups`	Você pode anexar grupos de segurança ao nó AWS.networking.ENI.	2 de abril de 2024
Support para criptografia de volume raiz do Amazon EBS	Você pode habilitar a criptografia do Amazon EBS para o volume raiz do Amazon EBS. Para habilitar, adicione as propriedades no nó aws.compute.eks ou aws.compute.eks. ManagedNode SelfManagedNode	2 de abril de 2024
Support para node `labels`	Você pode anexar rótulos de nós ao seu grupo de nós no nó AWS.compute.eks ou AWS.compute.eks. ManagedNode SelfManagedNode	19 de março de 2024

Support para interface de rede `source_dest_check`	Você pode indicar se deseja ativar ou desativar a verificação de origem/destino da interface de rede por meio do nó .networking.ENI. AWS	25 de janeiro de 2024
Suporte a instâncias do Amazon EC2 com dados de usuário personalizados	Você pode iniciar instâncias do Amazon EC2 com dados personalizados do usuário por meio do AWS.Compute. UserData nodo.	16 de janeiro de 2024
Suporte a grupo de segurança	AWS O TNB permite que você importe o AWS recurso do Grupo de Segurança.	8 de janeiro de 2024
Descrição de `network_interfaces` atualizada	Quando a `network_interfaces` propriedade é incluída no SelfManagedNode nó aws.compute.eksManagedNode ou aws.compute.eks, o AWS TNB obtém a permissão relacionada aos ENIs da propriedade, se disponível, ou da propriedade. `multus_role` `node_role`	18 de dezembro de 2023
Suporte a cluster privado	AWS O TNB agora oferece suporte a clusters privados. Para indicar um cluster privado, defina a propriedade `access` como `PRIVATE`.	11 de dezembro de 2023
Versão do Kubernetes para cluster	AWS O TNB agora oferece suporte às versões 1.28 do Kubernetes para criar clusters do Amazon EKS.	11 de dezembro de 2023

AWS TNB apoia grupo de colocação	Grupo de posicionamento adicionado para as definições do nó `AWS.Compute.EKSManagedNode` e `AWS.Compute.EKSSelfManagedNode`.	11 de dezembro de 2023
AWS TNB adiciona suporte para IPv6	AWS O TNB agora oferece suporte à criação de instâncias de rede com infraestrutura IPv6. Verifique os nós AWS.Networking.VPC, .Networking.Subnet, .Networking.AWSAWS InternetGateway, AWS.Rede. SecurityGroupIngressRule, AWS.Rede. SecurityGroupEgressRulee AWS.compute.eks para configurações de IPv6. Também adicionamos os nós AWS.Networking.NATGateway e AWS.Networking.Route para a configuração de NAT64. Atualizamos a função de serviço AWS TNB e a função de serviço AWS TNB para o grupo de nós Amazon EKS para permissões IPv6. Consulte Service role policy examples.	16 de novembro de 2023

Permissões adicionadas à política de função de serviço do AWS TNB	Adicionamos permissões à política de função de serviço do AWS TNB para o Amazon S3 AWS CloudFormation e para permitir a instanciação da infraestrutura.	23 de outubro de 2023
AWS TNB lançado em mais regiões	AWS O TNB agora está disponível nas regiões Ásia-Pacífico (Seul), Canadá (Central), Europa (Espanha), Europa (Estocolmo) e América do Sul (São Paulo).	27 de setembro de 2023
Etiquetas para AWS.compute.eks SelfManagedNode	AWS O TNB agora suporta tags para a definição do `AWS.Compute.EKSSelfManagedNode` nó.	22 de agosto de 2023
AWS O TNB oferece suporte a instâncias que utilizam o IMDSv2	Ao executar sua instância, você precisa usar o IMDSv2.	14 de agosto de 2023
Permissões atualizadas para o MultusRoleInlinePolicy	O MultusRoleInlinePolicy agora inclui a `ec2:DeleteNetworkInterface` permissão.	7 de agosto de 2023
Versão do Kubernetes para cluster	AWS O TNB agora oferece suporte às versões 1.27 do Kubernetes para criar clusters do Amazon EKS.	25 de julho de 2023

AWS.compute.eks. AuthRole	AWS O TNB oferece suporte para AuthRole que você adicione funções do IAM ao cluster `aws-auth` `ConfigMap` do Amazon EKS para que os usuários possam acessar o cluster do Amazon EKS usando uma função do IAM.	19 de julho de 2023
AWS O TNB oferece suporte a grupos de segurança.	Adicionou o AWS.Networking. SecurityGroup, AWS.Rede. SecurityGroupEgressRule, e AWS.Networking. SecurityG roupIngressRulepara o modelo NSD.	18 de julho de 2023
Versão do Kubernetes para cluster	AWS O TNB oferece suporte às versões 1.22 a 1.26 do Kubernetes para criar clusters do Amazon EKS. AWS O TNB não é mais compatível com as versões 1.21 do Kubernetes.	11 de maio de 2023
AWS.compute.eks SelfManag edNode	Você pode criar nós de trabalho autogerenciados na região, nas Zonas AWS Locais e. AWS Outposts	29 de março de 2023
Lançamento inicial	Esta é a primeira versão do Guia do Usuário do AWS TNB.	21 de fevereiro de 2023

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.