Manual do usuário

AWS Construtor de rede Telco

AWS Construtor de rede Telco: Manual do usuário

Table of Contents

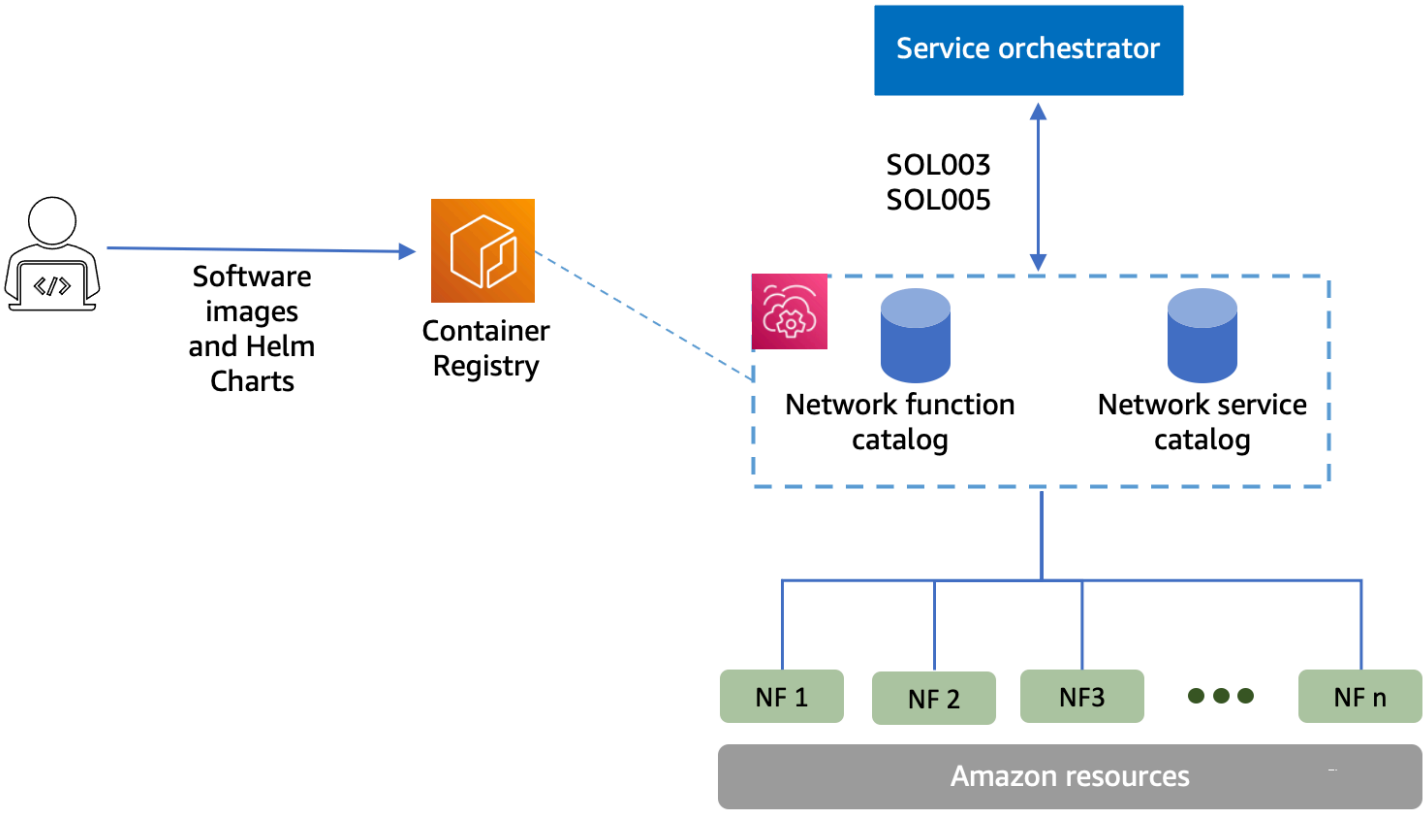
O que é o AWS Telco Network Builder?

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

Com AWS TNB, você implanta redes 5G escaláveis e seguras Nuvem AWS usando uma imagem da sua rede de forma automatizada. Você não precisa aprender novas tecnologias, decidir qual serviço de computação usar ou saber como provisionar e configurar AWS recursos.

Em vez disso, você descreve a infraestrutura de sua rede e fornece as imagens de software das funções de rede de seus parceiros independentes fornecedores de software (ISV). AWS TNBintegra-se a AWS serviços e orquestradores de serviços terceirizados para provisionar automaticamente a AWS infraestrutura 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 os orquestradores de serviços 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 em AWS?

Se você é iniciante em AWS produtos e serviços, comece a aprender mais com os seguintes recursos:

- Introdução à AWS
- Começando com AWS

Para quem é o AWS TNB?

AWS TNBé para CSPs aproveitar a economia, a agilidade e a elasticidade que as Nuvem AWS ofertas oferecem sem escrever e manter scripts e configurações personalizados para projetar, implantar e gerenciar serviços de rede. AWS TNBprovisiona automaticamente a AWS infraestrutura 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 CSP de rede definidos e nas funções de rede que ele deseja implantar. CSP

AWS TNBcaracterísticas

A seguir estão alguns dos motivos pelos quais um CSP gostaria de usar 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

AWS TNBintegra-se a orquestradores de serviços terceirizados populares que estão em conformidade. ETSI

Faz escalonamento

Você pode configurar AWS TNB para escalar AWS os recursos subjacentes 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 AWS

AWS TNBpermite que você inspecione e monitore os AWS recursos que dão suporte à sua rede em um único painel, como Amazon VPCEC2, Amazon e AmazonEKS.

Compatibilidade com modelos de serviço

AWS TNBpermite criar modelos de serviço para todas as cargas de trabalho 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 subjacente de uma implantação de função de rede, por exemplo, alterando o tipo de instância de um tipo de EC2 instância da Amazon, você pode acompanhar 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. Quando você usa AWS TNB para gerenciar o ciclo de vida da sua função de rede, você só faz as alterações nos descritores do serviço de rede que descrevem a função de rede. AWS TNBem seguida, 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 seus RAN aplicativos principais e de rede da mesma forma. IMS

Acessando AWS TNB

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

- AWS TNBconsole — Fornece uma interface web para gerenciar sua rede.
- AWS TNBAPI— Fornece um RESTful API para realizar AWS TNB ações. Para obter mais informações, consulte a AWS TNBAPIReferência
- AWS Command Line Interface (AWS CLI) — Fornece comandos para um amplo conjunto de AWS serviços, incluindo AWS TNB. É compatível com Windows, macOS e Linux. Para obter mais informações, consulte AWS Command Line Interface.
- AWS SDKs— Fornece idiomas específicos APIs e completa muitos dos detalhes da conexão. Por exemplo, cálculo de assinaturas, tratamento de novas tentativas de solicitação e tratamento de erros. Para obter mais informações, consulte AWSSDKs.

Preços para AWS TNB

AWS TNBajuda a CSPs automatizar a implantação e o gerenciamento de suas redes de telecomunicações em. AWS Você paga pelas duas dimensões a seguir ao usar AWS TNB:

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

Você também incorre em cobranças adicionais ao usar outros AWS serviços em conjunto com. AWS TNB Para obter mais informações, consulte AWS TNBPreços.

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 AWS da conta, consulte Faturamento AWS da conta.

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

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

Próximas etapas

Para obter mais informações sobre como começar a usar AWS TNB, consulte os tópicos a seguir:

- Configurando AWS TNB: concluir as etapas de pré-requisitos.
- Começando com AWS TNB— Implante 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.

Como AWS TNB funciona

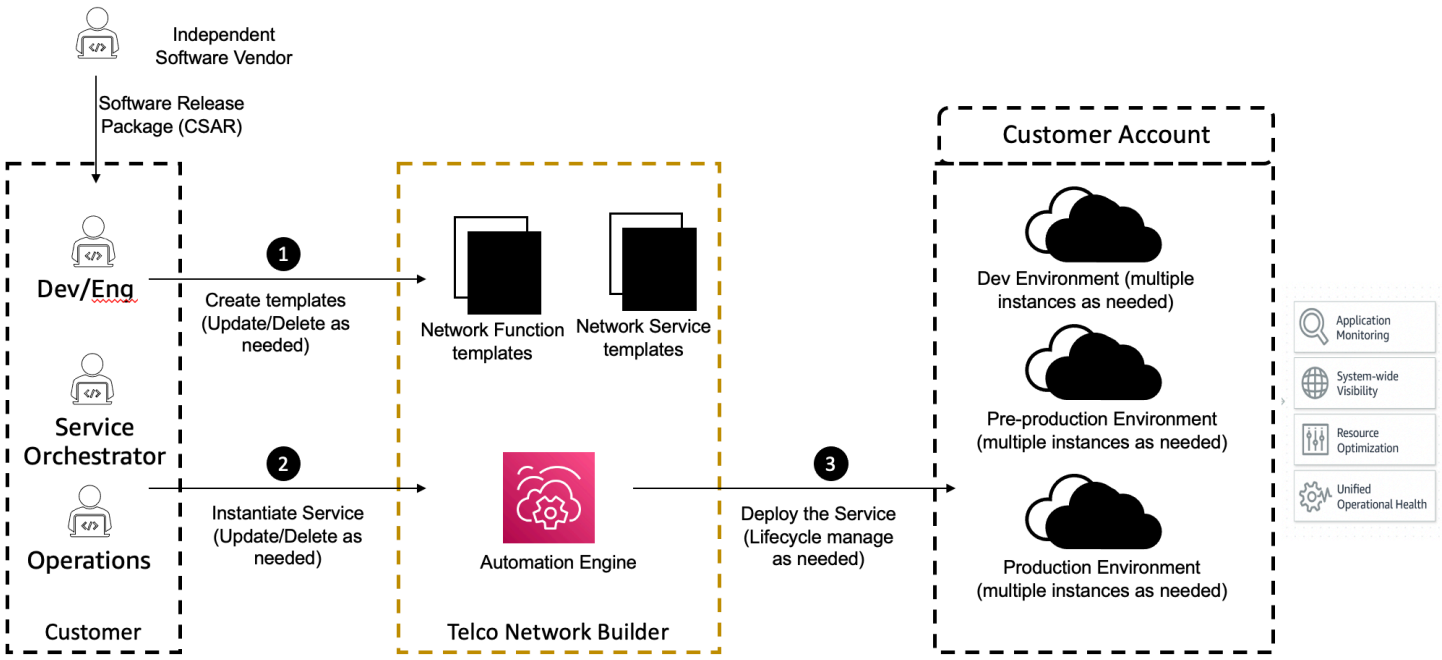
AWS TNBintegra-se com end-to-end orquestradores e AWS recursos padronizados para operar redes 5G completas.

AWS TNBpermite 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 end-to-end orquestrador e integrá-lo ou usá-lo AWS TNB SDKs para criar seu próprio fluxo de automação. AWS TNB APIs Para obter mais informações, consulte AWS TNBarquitetura .

Tópicos

AWS TNBarquitetura

AWS TNBfornece a capacidade de realizar operações de gerenciamento do ciclo de vida por meio do AWS Management Console, AWS CLI AWS TNB RESTAPI, e. SDKs Isso permite que diferentes CSP personalidades, como membros das equipes de Engenharia, Operações e Sistema Programático, aproveitem. AWS TNB Você cria e carrega um pacote de funções de rede como um arquivo Cloud Service Archive (CSAR). O CSAR arquivo contém gráficos 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 AWS infraestrutura. AWS TNBfornece 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 Kubernetes. AWS TNBintegra-se ao seguinte Serviços da AWS como específico APIs para telecomunicações para criar um serviço de rede totalmente operacional:

- Amazon Elastic Container Registry (AmazonECR) para armazenar artefatos de funções de rede de fornecedores independentes de software (ISVs).
- Amazon Elastic Kubernetes Service (AmazonEKS) para configurar clusters.
- Amazon VPC para construções de rede.
- Grupos de segurança usando AWS CloudFormation.
- AWS CodePipeline para alvos de implantação em Regiões da AWS, AWS Locais Zonas AWS Outposts e.
- IAMpara definir funções.
- AWS Organizations para controlar o acesso AWS TNB APIs a.
- AWS Health Dashboard e AWS CloudTrail para monitorar a saúde e as métricas de publicação.

AWS TNBcotas de recursos

Você Conta da AWS tem cotas padrão, anteriormente chamadas de limites, para cada um. AWS service (Serviço da AWS) Salvo indicação em contrário, cada cota é específica para um Região da AWS. Você pode solicitar aumentos para algumas cotas, mas não para todas as cotas.

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

Para solicitar o aumento da quota, consulte Solicitar um aumento de quota no Guia do usuário do Service Quotas.

Você Conta da AWS tem as seguintes cotas relacionadas a. 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 TNBconceitos

Este tópico descreve conceitos essenciais para ajudar você a começar a usar AWS TNB.

Conteúdo

Ciclo de vida de uma função de rede

AWS TNBajuda 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 CSAR pacotes para implantar ou atualizar.
4. Use AWS TNB para carregar o CSAR pacote que define sua função de rede (por exemploAMF, UC eUPF) e integrá-lo a um pipeline de integração contínua e entrega contínua (CI/CD) que pode ajudá-lo a criar novas versões do seu CSAR pacote 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. Ingestão NSDs que define 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 its 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.
- Atualize a instância de rede com um novo pacote de rede e valores de parâmetros. Por exemplo, atualize o EKS `version` parâmetro Amazon no pacote de rede.

Use interfaces padronizadas

AWS TNBintegra-se com 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 AWS TNB SDKsCLI, o ou o APIs para iniciar operações, como instanciar ou atualizar uma função de rede para uma nova versão.

AWS TNBsuporta as seguintes especificações.

Especificação	Versão	Descrição
ETSISOL001	v3.6.1	Define padrões para permitir descritores de funções de rede TOSCA baseados.
ETSISOL002	v3.6.1	Define modelos em torno do gerenciamento de funções de rede.
ETSISOL003	v3.6.1	Define padrões para o gerenciamento do ciclo de vida das funções de rede.
ETSISOL004	v3.6.1	Define CSAR padrões para pacotes de funções de rede.

Especificação	Versão	Descrição
ETSISOL005	v3.6.1	Define padrões para pacotes de serviços de rede e gerenciamento do ciclo de vida do serviço de rede.
ETSISOL007	v3.5.1	Define padrões para permitir descritores de serviços de rede TOSCA baseados em.

Pacotes de funções de rede para AWS TNB

Com AWS TNB, você pode armazenar pacotes de funções de rede que estejam em conformidade com ETSI SOL 001/ SOL 004 em um catálogo de funções. Em seguida, você pode fazer upload de pacotes 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 (AmazonECR) pode atuar como seu 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 OASIS TOSCA padrão e inclui um descritor de rede/serviço que está em conformidade com a especificação. OASIS TOSCA YAML Para obter informações sobre a YAML especificação necessária, consulteTOSCAreferência para 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 serviços de rede para AWS TNB

AWS TNBarmazena descritores de serviços de rede (NSDs) sobre as funções de rede que você deseja implantar e como deseja implantá-las no catálogo. Você pode fazer o upload YAML NSD do seu arquivo (`vnfd.yaml`), conforme descrito por ETSI SOL 007, para incluir as seguintes informações:

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

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

A seguir está um trecho de uma NSD demonstração de como modelar. Serviços da AWS A função de rede será implantada em um EKS cluster da Amazon 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
```

Gerenciamento e operações para AWS TNB

Com AWS TNB, você pode gerenciar sua rede usando operações de gerenciamento padronizadas de acordo com ETSI SOL 003 e SOL 005. Você pode usar o AWS TNB APIs 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.
- Atualizar uma instância de rede instanciada ou atualizada com um novo pacote de rede e valores de parâmetros.
- Gerenciamento de versões de seus pacotes de funções de rede.
- Gerenciando versões do seuNSDs.
- Recuperação de informações sobre suas funções de rede implantadas.

Descritores de serviços de rede para AWS TNB

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

Seu NSD arquivo está dividido nas seguintes partes:

1. TOSCAversão da definição — Essa é a primeira linha do seu NSD YAML arquivo e contém as informações da versão, mostradas no exemplo a seguir.

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

2. VNFD— 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 CSAR pacote de funções de rede.
   - Um nome exclusivo para `namespace`. O nome deve estar associado a uma ID exclusiva para ser referenciado com mais facilidade em todo o NSD YAML arquivo, conforme 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 EKS do Amazon Node Group, ele deve ser criado dentro de um Amazon EKS Cluster. 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 AWS TNB

Configure AWS TNB concluindo as tarefas descritas neste tópico.

Tarefas

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

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e inserir 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 atributos 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.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira sua senha.

   Para obter ajuda ao fazer login usando o usuário raiz, consulte Fazer login como usuário raiz no Guia do usuário do Início de Sessão da AWS .
2. Ative a autenticação multifator (MFA) para seu usuário root.

   Para obter instruções, consulte Habilitar um MFA dispositivo virtual para seu usuário Conta da AWS root (console) no Guia IAM do usuário.

Criar um usuário com acesso administrativo

1. Ative o IAM Identity Center.

   Para obter instruções, consulte Habilitar AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No IAM Identity Center, conceda acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para entrar com seu usuário do IAM Identity Center, use o login URL que foi enviado ao seu endereço de e-mail quando você criou o usuário do IAM Identity Center.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No IAM Identity Center, crie um conjunto de permissões que siga as melhores práticas de aplicação de permissões com privilégios mínimos.

   Para obter instruções, consulte Create a permission set no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Add groups no Guia do usuário do AWS IAM Identity Center .

Escolha uma AWS região

Para ver a lista de regiões disponíveis para AWS TNB, consulte a Lista de serviços AWS regionais. Para ver a lista de endpoints para acesso programático, consulte AWS TNBendpoints 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 FIPS endpoints 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	

Nome da região	Região	Endpoint	Protocolo	
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	
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 AWS TNB usando AWS CLI o. Para começar a usar, consulte o Guia do usuário da AWS Command Line Interface. Para

obter mais informações sobre os comandos para AWS TNB, consulte tnb na Referência de AWS CLI Comandos.

Configurar AWS TNB funções

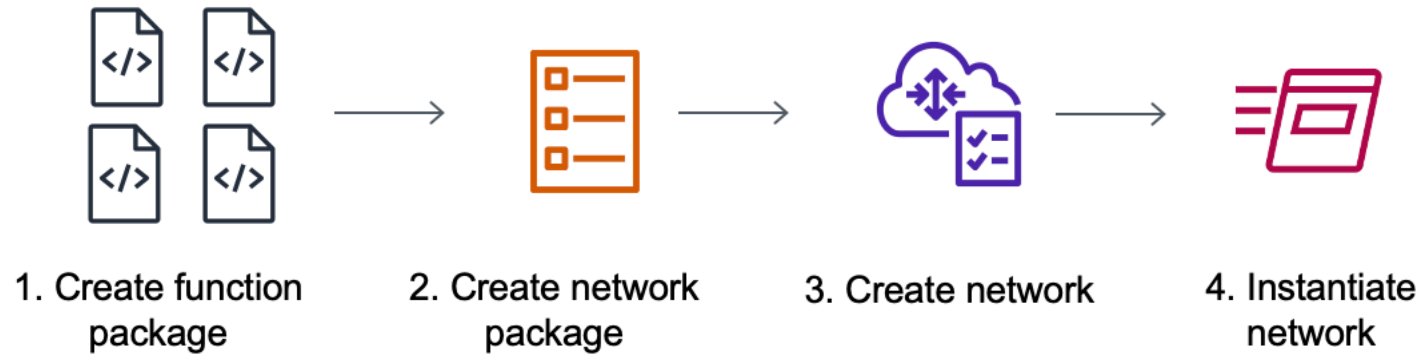
Você deve criar uma função IAM de serviço para gerenciar diferentes partes da sua AWS TNB solução. AWS TNBas funções de serviço podem fazer API chamadas para outros AWS serviços, como AWS CloudFormation AWS CodeBuild, e vários serviços de computação e armazenamento, em seu nome, para instanciar e gerenciar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

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

Começando com AWS TNB

Este tutorial demonstra como você usa 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 5G (). UPF

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 IAM funções.
- Um pacote de função de rede (NF) compatível com ETSI SOL 001/ SOL 004.
- Modelos do Network Service Descriptor (NSD) que estão em conformidade com ETSI SOL 007.

Você pode usar um pacote de funções de amostra ou pacote de rede dos pacotes de amostra para o AWS TNB GitHub site.

Criar um pacote de funções

Um pacote de funções de rede é um arquivo Cloud Service Archive (CSAR). O CSAR arquivo contém gráficos do Helm, imagens de software e um descritor de função de rede ()NFD.

Para criar um pacote de funções

1. Abra o AWS TNB console 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 arquivos e carregue cada CSAR pacote como um `.zip` arquivo. Você pode fazer upload de no máximo 10 arquivos.
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

Um pacote de rede especifica as funções de rede que você deseja implantar e como deseja implantá-las no catálogo.

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 arquivos e carregue cada um NSD como um `.zip` arquivo. Você pode fazer upload de no máximo 10 arquivos.
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

Uma instância de rede é uma única rede criada na AWS TNB qual pode ser implantada. Você deve criar uma instância de rede e instanciá-la. Quando você instancia uma instância de rede, AWS TNB provisiona a AWS infraestrutura necessária, implanta funções de rede em contêineres e configura o gerenciamento de rede e acesso para criar um serviço de rede totalmente operacional.

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. Escolha um pacote de rede. Verifique os detalhes e escolha Avançar.
5. Clique em Criar instância de rede. O estado inicial é `Created`.

   A página Redes é exibida mostrando a nova instância de rede no `Not instantiated` estado.
6. Selecione a instância de rede, escolha Ações e Instanciar.

   A página de instanciação de rede é exibida.
7. Revise os detalhes e atualize os valores dos parâmetros. As atualizações nos valores dos parâmetros se aplicam somente a essa instância de rede. Os parâmetros nos VNFD pacotes NSD e não são alterados.
8. Escolha Instanciar rede.

   A página de status de implantação é exibida.
9. Use o ícone Atualizar para rastrear o status de implantação da sua instância de rede. Você também pode ativar a atualização automática na seção Tarefas de implantação para acompanhar o progresso de cada tarefa.

Limpeza

Agora você pode excluir os recursos que você criou para este tutorial.

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 para 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 telecomunicações ETSI padrão) e um descritor de pacote de funções que usa o TOSCA padrão para descrever como as funções de rede devem ser executadas em sua rede.

Tarefas

- Crie um pacote de funções em AWS TNB
- Exibir um pacote de funções em AWS TNB
- Baixe um pacote de funções em AWS TNB
- Excluir um pacote de funções de AWS TNB

Crie um pacote de funções em AWS TNB

Saiba como criar um pacote de funções no catálogo de funções de AWS TNB rede. Criar um pacote de funções é a primeira etapa para criar uma rede em AWS TNB. Depois de carregar um pacote de funções, você pode criar um pacote de rede.

Console

Para criar um pacote de funções usando o console

1. Abra o AWS TNB console 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 arquivos e carregue cada CSAR pacote como um `.zip` arquivo. Você pode fazer upload de no máximo 10 arquivos.
5. Escolha Próximo.
6. Revise os detalhes do pacote.
7. Escolha Criar pacote de funções.

AWS CLI

Para criar um pacote de funções usando o AWS CLI

1. Use o create-sol-function-packagecomando 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 em 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 AWS TNB console 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 visualizar um pacote de funções usando o AWS CLI

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

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

2. Use o get-sol-function-packagecomando para ver 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
```

Baixe um pacote de funções em AWS TNB

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

Console

Para baixar um pacote de funções usando o console

1. Abra o AWS TNB console 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 o 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 de AWS TNB

Saiba como excluir um pacote de funções do catálogo de funções de AWS TNB rede. 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 AWS TNB console 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 o AWS CLI

1. Use o update-sol-function-packagecomando para desativar um pacote de funções.

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

2. Use o delete-sol-function-packagecomando 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 AWS TNB

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

Tarefas

- Crie um pacote de rede no AWS TNB
- Exibir um pacote de rede em AWS TNB
- Faça o download de um pacote de rede do AWS TNB
- Excluir um pacote de rede do AWS TNB

Crie 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 para 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 da AmazonEKS.

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 AWS TNB console 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 arquivos e carregue cada um NSD como um `.zip` arquivo. Você pode fazer upload de no máximo 10 arquivos.
5. Escolha Próximo.
6. Revise os detalhes do pacote.
7. Escolha Criar pacote de rede.

AWS CLI

Para criar um pacote de rede usando o AWS CLI

1. Use o create-sol-network-packagecomando 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 em 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 AWS TNB console 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 visualizar um pacote de rede usando o AWS CLI

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

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

2. Use o get-sol-network-packagecomando para ver 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
```

Faça o download de um pacote de rede do AWS TNB

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

Console

Para baixar um pacote de rede usando o console

1. Abra o AWS TNB console 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 o AWS CLI

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

```
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 AWS TNB rede. 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 AWS TNB console 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 o AWS CLI

1. Use o update-sol-network-packagecomando para desativar 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 delete-sol-network-packagecomando 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 para AWS TNB

Uma instância de rede é uma única rede criada na AWS TNB qual pode ser implantada.

Tarefas

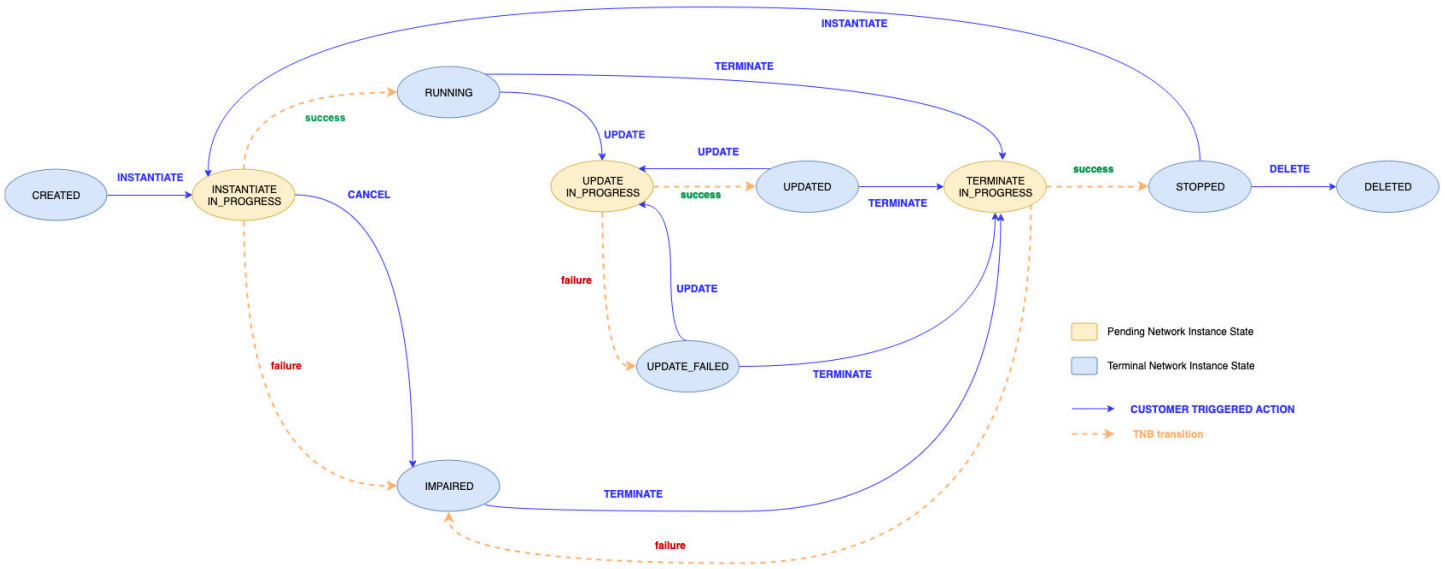
- Operações do ciclo de vida de uma instância de rede
- Crie uma instância de rede usando AWS TNB
- Instancie uma instância de rede usando AWS TNB
- Atualize uma instância de função em AWS TNB
- Atualize uma instância de rede no AWS TNB
- Visualize uma instância de rede em AWS TNB
- Encerre e exclua uma instância de rede do AWS TNB

Operações do ciclo de vida de uma instância de rede

AWS TNBpermite que você gerencie facilmente sua rede usando operações de gerenciamento padronizadas em linha com ETSI SOL 003 e 005. SOL Você pode realizar as seguintes operações de ciclo de vida:

- Crie a rede
- Instancie a rede
- Atualize a função de rede
- Atualizar a instância de rede
- Exibir detalhes e status da rede
- Encerrar a rede

A imagem a seguir mostra as operações de gerenciamento de rede:



Crie 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, instancie-a.

Console

Para criar uma instância de rede usando o console

1. Abra o AWS TNB console 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 o pacote de rede, verifique os detalhes e escolha Avançar.
6. Clique em Criar instância de rede.

   A nova instância de rede aparece na página Redes. Em seguida, instancie essa instância de rede.

AWS CLI

Para criar uma instância de rede usando o AWS CLI

- Use o create-sol-network-instancecomando 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"
```

Em seguida, instancie essa instância de rede.

Instancie uma instância de rede usando AWS TNB

Depois de criar uma instância de rede, você deve instanciá-la. Quando você instancia uma instância de rede, AWS TNB provisiona a AWS infraestrutura necessária, implanta funções de rede em contêineres e configura o gerenciamento de rede e acesso para criar um serviço de rede totalmente operacional.

Console

Para instanciar uma instância de rede usando o console

1. Abra o AWS TNB console em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Redes.
3. Selecione a instância de rede que você deseja instanciar.
4. Escolha Ações e, em seguida, Instanciar.
5. Na página Instanciar rede, revise os detalhes e, opcionalmente, atualize os valores dos parâmetros.

   As atualizações nos valores dos parâmetros se aplicam somente a essa instância de rede. Os parâmetros nos VNFD pacotes NSD e não são alterados.
6. Escolha Instanciar rede.

   A página de status de implantação é exibida.
7. Use o ícone Atualizar para rastrear o status de implantação da sua instância de rede. Você também pode ativar a atualização automática na seção Tarefas de implantação para acompanhar o progresso de cada tarefa.

   Quando o status de implantação muda para`Completed`, a instância de rede é instanciada.

AWS CLI

Para instanciar uma instância de rede usando o AWS CLI

1. Use o instantiate-sol-network-instancecomando para instanciar a instância de rede.

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

2. Em seguida, visualize o status da operação da rede.

Atualize uma instância de função em AWS TNB

Depois que uma instância de rede é instanciada, você pode atualizar um pacote de funções na instância de rede.

Console

Para atualizar uma instância de função usando o console

1. Abra o AWS TNB console em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Redes.
3. Selecione a instância de rede. Você pode atualizar uma instância de rede somente se seu estado for`Instantiated`.

   A página da instância de rede é exibida.
4. Na guia Funções, selecione a instância da função a ser atualizada.
5. Selecione Atualizar.
6. Insira suas substituições de atualização.
7. Selecione Atualizar.

AWS CLI

Use o CLI para atualizar uma instância de função

Use o update-sol-network-instancecomando com o tipo de `MODIFY_VNF_INFORMATION` atualização para atualizar uma instância de função em 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 ...
```

Atualize uma instância de rede no AWS TNB

Depois que uma instância de rede for instanciada, talvez seja necessário atualizar a infraestrutura ou o aplicativo. Para fazer isso, você atualiza o pacote de rede e os valores dos parâmetros da instância de rede e implanta a operação de atualização para aplicar as alterações.

Considerações

- Você pode atualizar uma instância de rede que esteja no `Updated` estado `Instantiated` ou.
- Quando você atualiza uma instância de rede, ele `UpdateSolNetworkService` API usa o novo pacote de rede e valores de parâmetros para atualizar a topologia da instância de rede.
- AWS TNBverifica se o número NSD e VNFD os parâmetros na instância de rede não excedem 200. Esse limite é aplicado para evitar que agentes mal-intencionados transmitam cargas errôneas ou enormes que afetam o serviço.

Parâmetros que você pode atualizar

Você pode atualizar os seguintes parâmetros ao atualizar uma instância de rede instanciada:

Parâmetro	Descrição	Exemplo: Antes	Exem Depo

Versão de EKS cluster da Amazon	Você pode atualizar o valor do `version` parâmetro do plano de controle de EKS cluster da Amazon para a próxima versão secundária. Você não pode fazer o downgrade da versão. Os nós de trabalho não são atualizados.	`EKSCluster:`
		`  type: tosca.nod`
		`es.AWS.Compute.EKS`
		`  properties:`
		`  version: "1.28"`
			`EKSC`
			`r:`
			
			`typ`
			`tos`
			`es.A`
			`mput`

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
			pro s: ver "1.

Parâmetro	Descrição	Exemplo: Antes	Exem Depo

Propriedades de dimensionamento	Você pode atualizar as propriedades de escala dos `EKSSelfManagedNode` TOSCA nós `EKSManagedNode` e.	`EKSNodeGroup01:`
		`  ...`
		`  scaling:`
		`  properties:`
		`  desired_size: 1`
		`  min_size: 1`
		`  max_size: 1`
			`EKSN`
			`oup0`
			`...`
			`sca`
			`pro`
			`s:`
			`des`
			`ize:`

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
			mir max

Parâmetro	Descrição	Exemplo: Antes	Exem Depo

Propriedades do EBS CSI plug-in da Amazon	Você pode ativar ou desativar o EBS CSI plug-in da Amazon em seus EKS clusters da Amazon. Você também pode alterar a versão do plugin.	`EKSCluster:`
		`  capabilities:`
		`  ...`
		`  ebs_csi:`
		`  properties:`
		`  enabled: false`
			`EKSC`
			`r:`
			`cap`
			`ies:`
			`...`
			`ebs`
			`pro`
			`s:`
			`ena`
			`ver`
			`"v1`
			`e`
			`ksbu`
			`"`

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
VNF	Você pode referenciar os VNFs no NSD e implantá-los no cluster criado NSD usando o VNFDeployment TOSCA node. Como parte da atualizaç ão, você poderá adicionar, atualizar e VNFs excluir na rede.	vnfds: - descriptor_id: "43c012fa-2616-41a8-a833-0dfd4c5a049e " namespace: " vnf1" - descriptor_id: "64222f98-ecd6-4871-bf94-7354b53f3ee5 " namespace: "vnf2" // Deleted VNF ... SampleVNF1HelmDeploy: type: tosca.nodes.AWS.Deployment.VNFDeployment requirements: cluster: EKSCluster vnfs: - vnf1.SampleVNF1 - vnf2.SampleVNF2	vnfd - des r_id "59 79e9 - be53 2ad0 " nam : "vr Upd VNF - des r_id "b7 839d -916 a166 " nam : "vr Add VNF .... Sa mple

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
			elmD : typ tos es.A ploy VNFD ment req nts: clu EKS r vnf

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
			- v leVN - v leVN

Parâmetro	Descrição	Exemplo: Antes	Exem Depo

Hooks	Para executar operações de ciclo de vida antes e depois de criar uma função de rede, adicione os `post_create` ganchos `pre_create` e ao `VNFDeployment` nó.
	
	Neste exemplo, o `PreCreate Hook` gancho será executado antes de ser `vnf3.Samp leVNF3` instanciado e o `PostCreateHook` gancho será executado depois de ser `vnf3.SampleVNF3` instanciado.
		`vnfds:`
		`  - descriptor_id:`
		`  "43c012fa-2616-41a8-a833-0dfd4c5a049e  "`
		`  namespace: " vnf1"`
		`  - descriptor_id:`
		`  "64222f98-ecd6-4871-bf94-7354b53f3ee5  "`
		`  namespace: " vnf2"`
		`...`
		`SampleVNF1HelmDeploy:`
		`  type: tosca.nod es.AWS.Deployment. VNFDeployment`
		`  requirements:`
		`  cluster: EKSCluster`
		`  vnfs:`
		`  - vnf1.SampleVNF1`
		`  - vnf2.Samp leVNF2  // Removed during update`
			`vnfd`
			
			`  -`
			`  des r_id`
			`  "43 2616 - a833 d4c5 "`
			
			`  nam :`
			`  "vn`
			
			`  -`
			`  des r_id`
			`  "b7 839d -916 a166 "`
			
			`  nam :`
			`  "vn`
			`....`
			`S ampl Helm y:`
			
			`  typ`
			`  tos`

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
			es.A ploy VNFD ment rec nts: clu EKS r vnf - v leVN No cha to thi fur as the nam and uui rem the sam

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
			- v leVN New VNF as the nam , vnf was not pre y pre int s: Hoo pos te: eHoo pre e: Hook

Parâmetro	Descrição	Exemplo: Antes	Exem Depo

Hooks	Para executar operações de ciclo de vida antes e depois de atualizar uma função de rede, você pode adicionar o `pre_update` gancho e o `post_update` gancho ao `VNFDeployment` nó.
	
	Neste exemplo, `PreUpdate Hook` será executado antes da `vnf1.SampleVNF1` atualização e `PostUpdat eHook` será executado após `vnf1.SampleVNF1` a atualização para o `vnf` pacote indicado pela atualização `uuid` para o namespace vnf1.
		`vnfds:`
		`  - descriptor_id:`
		`  "43c012fa-2616-41a8-a833-0dfd4c5a049e  "`
		`    namespace: " vnf1"`
		`  - descriptor_id:`
		`  "64222f98-ecd6-4871-bf94-7354b53f3ee5  "`
		`    namespace: " vnf2"`
		`...`
		
		`SampleVNF1HelmDeploy:`
		`  type: tosca.nod es.AWS.Deployment. VNFDeployment`
		`  requirements:`
		`    cluster: EKSCluster`
		`    vnfs:`
		`      - vnf1.SampleVNF1`
		`      - vnf2.Samp leVNF2`
			`vnfd`
			
			`  -`
			`  des r_id`
			`  "0e bd87 -`
			`  b8a1 4666`
			`  "`
			
			`  nam :`
			`  "vn`
			
			`  -`
			`  des r_id`
			`  "64 ecd6 -`
			`  bf94 4b53`
			`  "`
			
			`  nam :`
			`  "vn`
			`...`
			`S ampl Helm y:`
			
			`  typ`

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
			tos es.A ploy VNFD ment rec nts: clu EKS r vnf - v leVN A VNF upd as the uui cha for nam "vr - v

Parâmetro	Descrição	Exemplo: Antes	Exem Depo
			leVN No cha to thi fur as nam and uui rem the sam int s: Hoo pre e: Hook pos te: eHoo

Atualização de uma instância de rede

Console

Para atualizar uma instância de rede usando o console

1. Abra o AWS TNB console em https://console.aws.amazon.com/tnb/.
2. No painel de navegação, selecione Redes.
3. Selecione a instância de rede. Você pode atualizar uma instância de rede somente se seu estado for `Instantiated` ou`Updated`.
4. Escolha Ações e atualização.

   A página Atualizar instância aparece com os detalhes da rede e uma lista de parâmetros na infraestrutura atual.
5. Escolha um novo pacote de rede.

   Os parâmetros no novo pacote de rede aparecem na seção Parâmetros atualizados.
6. Opcionalmente, atualize os valores dos parâmetros na seção Parâmetros atualizados. Para obter a lista de valores de parâmetros que você pode atualizar, consulteParâmetros que você pode atualizar.
7. Escolha Atualizar rede.

   AWS TNBvalida a solicitação e inicia a implantação. A página de status de implantação é exibida.
8. Use o ícone Atualizar para rastrear o status de implantação da sua instância de rede. Você também pode ativar a atualização automática na seção Tarefas de implantação para acompanhar o progresso de cada tarefa.

   Quando o status de implantação muda para`Completed`, a instância de rede é atualizada.
9. - Se a validação falhar, a instância de rede permanecerá no mesmo estado em que estava antes de você solicitar a atualização - `Instantiated` ou`Updated`.
   - Se a atualização falhar, o estado da instância da rede será exibido`Update failed`. Escolha o link para cada tarefa que falhou para determinar o motivo.
   - Se a atualização for bem-sucedida, o estado da instância da rede será exibido`Updated`.

AWS CLI

Use o CLI para atualizar uma instância de rede

Use o update-sol-network-instancecomando com o tipo de UPDATE_NS atualização para atualizar uma instância de rede.

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

Visualize uma instância de rede em AWS TNB

Saiba como exibir uma instância de rede.

Console

Para exibir uma instância de rede usando o console

1. Abra o AWS TNB console 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 visualizar uma instância de rede usando o AWS CLI

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

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

2. Use o get-sol-network-instancecomando para ver detalhes sobre uma instância de rede específica.

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

Encerre e exclua 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 AWS TNB console 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 o AWS CLI

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

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

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

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

Operações de rede para 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 AWS TNB de rede
- Cancelar uma operação AWS TNB de rede

Exibir uma operação AWS TNB 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 AWS TNB console 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 visualizar uma operação de rede usando o AWS CLI

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

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

2. Use o get-sol-network-operationcomando para ver 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 AWS TNB de rede

Saiba como cancelar uma operação de rede.

Console

Para cancelar uma operação de rede usando o console

1. Abra o AWS TNB console 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 o AWS CLI

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

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

TOSCAreferência para AWS TNB

A especificação de topologia e orquestração para aplicativos em nuvem (TOSCA) é uma sintaxe declarativa CSPs usada para descrever uma topologia de serviços web baseados em nuvem, seus componentes, relacionamentos e os processos que os gerenciam. CSPsdescreva os pontos de conexão, os links lógicos entre os pontos de conexão e as políticas, como afinidade e segurança, em um TOSCA modelo. CSPsem seguida, faça o upload do modelo AWS TNB que sintetiza os recursos necessários para estabelecer uma rede 5G funcional em todas as zonas de AWS disponibilidade.

Conteúdo

VNFDmodelo

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 TOSCA AWS Nodes. Os nós possíveis são:

- AWS.VNF
- AWS.Artifacts.Helm

AWS.VNF

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

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 doVNFD.

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 doVNFD.

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 AWS Helm Node.

Sintaxe

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

```
  properties:
    implementation: String
```

Propriedades

`implementation`

O diretório local que contém o gráfico do Helm dentro do CSAR pacote.

Obrigatório: sim

Tipo: String

Exemplo

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

Modelo de descritor de serviço de rede

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

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 }
```

VNFDimportar

`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 TOSCA AWS nós possíveis são:

- AWS.NS
- AWS.Computação. EKS
- AWS.Computação. EKS. AuthRole
- AWS.Computação. EKSManagedNode
- AWS.Computação. EKSSelfManagedNode
- 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.Trabalho em rede. ENI
- AWS.HookExecution
- AWS.Trabalho em rede. InternetGateway
- AWS.Trabalho em rede. RouteTable
- AWS.Networking.Subnet
- AWS.Implantação. VNFDeployment

- AWS.Trabalho em rede. VPC
- AWS.Trabalho em rede. 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 doNSD.

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.Computação. 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 você. AWS NFs Os plug-ins da interface de rede de contêineres Multus (CNI) 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 Multus (CNI).

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 Amazon EBS Container Storage Interface (CSI) instalado no EKS cluster da Amazon.

Habilite esse plug-in para usar nós EKS autogerenciados da Amazon em AWS Outposts AWS Locais Zones ou Regiões da AWS. Para obter mais informações, consulte o CSIdriver do Amazon Elastic Block Store no Guia EKS do usuário da Amazon.

`enabled`

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

Obrigatório: não

Tipo: booliano

`version`

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

Obrigatório: não

Tipo: string

Propriedades

`version`

A versão do Kubernetes para o cluster. AWS O Telco Network Builder oferece suporte às versões 1.23 a 1.30 do Kubernetes.

Obrigatório: sim

Tipo: String

Valores possíveis: 1,23 | 1,24 | 1,25 | 1,26 | 1,27 | 1,28 | 1,29 | 1,30

`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: string

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.Computação. EKS. AuthRole

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

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 IAM funções que precisam ser adicionadas ao cluster da AmazonEKS. `aws-auth` `ConfigMap`

`arn`

O ARN do IAM papel.

Obrigatório: sim

Tipo: String

`groups`

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

Obrigatório: Não

Tipo: lista

Requisitos

`clusters`

Um AWS computador. EKSnodo.

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.Computação. EKSManagedNode

AWS TNBoferece suporte a grupos de nós EKS gerenciados para automatizar o provisionamento e o gerenciamento do ciclo de vida dos nós (instâncias da AmazonEC2) para clusters do Amazon Kubernetes. EKS Para criar um grupo de EKS nós, faça o seguinte:

- Escolha as Amazon Machine Images (AMI) para seus nós de trabalho do cluster fornecendo a ID do AMI ou do AMI tipo.
- Forneça um par de EC2 chaves da Amazon para SSH acesso e as propriedades de escalabilidade do seu grupo de nós.
- Certifique-se de que seu grupo de nós esteja associado a um EKS cluster da Amazon.

- Forneça as sub-redes para os nós de trabalho.
- Opcionalmente, anexe 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 EKS gerenciados da Amazon, como tipos de EC2 instância da Amazon e EC2 instância da AmazonAMIs.

`ami_type`

O AMI tipo EKS compatível com a Amazon.

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 doAMI.

Obrigatório: não

Tipo: string

> **Note**
>
> Se ambos `ami_type` `ami_id` forem especificados no modelo, 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 EC2 chaves para permitir o SSH acesso.

Obrigatório: sim

Tipo: String

`root_volume_encryption`

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

Obrigatório: Não

Padrão: True

Tipo: booliano

`root_volume_encryption_key_arn`

O ARN da AWS KMS chave. AWS TNBsuporta chave regularARN, chave multirregional ARN e ARN alias.

Obrigatório: não

Tipo: string

> **Note**
>
> - Se `root_volume_encryption` for falso, não inclua`root_volume_encryption_key_arn`.
> - AWS TNBsuporta criptografia de volume raiz de produtos EBS apoiados pela AMI Amazon.
> - Se o volume raiz AMI do já estiver criptografado, você deverá incluir o `root_volume_encryption_key_arn` for AWS TNB para recriptografar o volume raiz.
> - Se AMI o volume raiz não estiver criptografado, AWS TNB use o `root_volume_encryption_key_arn` para criptografar o volume raiz.
>
> Se você não incluir`root_volume_encryption_key_arn`, AWS TNB usa a chave padrão fornecida por AWS Key Management Service para criptografar o volume raiz.
> - AWS TNBnão decifra nem criptografa. AMI

`scaling`

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

`desired_size`

O número de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: número inteiro

`min_size`

O número mínimo de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: número inteiro

`max_size`

O número máximo de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: número inteiro

Propriedades

`node_role`

A ARN da IAM função que está associada à EC2 instância da Amazon.

Obrigatório: sim

Tipo: String

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: Não

Tipo: lista

Requisitos

`cluster`

Um AWS computador. EKSnodo.

Obrigatório: sim

Tipo: String

`subnets`

Um nó AWS.Networking.Subnet.

Obrigatório: Sim

Tipo: lista

`network_interfaces`

Um AWS.Networking. ENInodo. 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 define`network_interfaces`, AWS TNB obtém a permissão relacionada à ENIs `multus_role` propriedade se você incluiu a `multus` propriedade no AWS.Compute. EKSnodo. Caso contrário, AWS TNB obtém a permissão relacionada à propriedade ENIs 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: string

`user_data`

Um tosca.nodes.AWS.Computação. UserDatareferência de nó. Um script de dados do usuário é passado para as EC2 instâncias da Amazon 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: string

`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.Computação. EKSSelfManagedNode

AWS TNBoferece suporte aos nós EKS autogerenciados da Amazon para automatizar o provisionamento e o gerenciamento do ciclo de vida dos nós (instâncias da AmazonEC2) para clusters do Amazon Kubernetes. EKS Para criar um grupo de EKS nós da Amazon, faça o seguinte:

- Escolha as Amazon Machine Images (AMI) para os nós de trabalho do seu cluster fornecendo a ID doAMI.
- Forneça um par de EC2 chaves da Amazon para SSH acesso.
- Certifique-se de que seu grupo de nós esteja associado a um EKS cluster da Amazon.
- Forneça o tipo de instância e os tamanhos desejados, mínimos e máximos.
- Forneça as sub-redes para os nós de trabalho.

- Opcionalmente, anexe 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 EKS autogerenciados da Amazon, como tipos de EC2 instância da Amazon e EC2 instância AMIs da Amazon.

`ami_id`

O AMI ID usado para iniciar a instância. AWS TNBoferece suporte a instâncias que aproveitamIMDSv2. Para obter mais informações, consulte IMDSversão.

Obrigatório: sim

Tipo: String

`instance_type`

O tamanho da instância.

Obrigatório: sim

Tipo: String

`key_pair`

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

Obrigatório: sim

Tipo: String

`root_volume_encryption`

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

Obrigatório: Não

Padrão: True

Tipo: booliano

`root_volume_encryption_key_arn`

O ARN da AWS KMS chave. AWS TNBsuporta chave regularARN, chave multirregional ARN e ARN alias.

Obrigatório: não

Tipo: string

> **Note**
>
> - Se `root_volume_encryption` for falso, não inclua`root_volume_encryption_key_arn`.

- AWS TNBsuporta criptografia de volume raiz de produtos EBS apoiados pela AMI Amazon.
- Se o volume raiz AMI do já estiver criptografado, você deverá incluir o `root_volume_encryption_key_arn` for AWS TNB para recriptografar o volume raiz.
- Se AMI o volume raiz não estiver criptografado, AWS TNB use o `root_volume_encryption_key_arn` para criptografar o volume raiz.

  Se você não incluir`root_volume_encryption_key_arn`, AWS TNB usa AWS Managed Services para criptografar o volume raiz.
- AWS TNBnão decifra nem criptografa. AMI

`scaling`

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

`desired_size`

O número de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: número inteiro

`min_size`

O número mínimo de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: número inteiro

`max_size`

O número máximo de instâncias neste NodeGroup.

Obrigatório: Sim

Tipo: número inteiro

Propriedades

`node_role`

A ARN da IAM função que está associada à EC2 instância da Amazon.

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 AWS computador. EKSnodo.

Obrigatório: sim

Tipo: String

`subnets`

Um nó AWS.Networking.Subnet.

Obrigatório: Sim

Tipo: lista

`network_interfaces`

Um AWS.Networking. ENInodo. 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 define`network_interfaces`, AWS TNB obtém a permissão relacionada à ENIs `multus_role` propriedade se você incluiu a `multus` propriedade no AWS.Compute. EKSnodo. Caso contrário, AWS TNB obtém a permissão relacionada à propriedade ENIs 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: string

`user_data`

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

Obrigatório: não

Tipo: string

`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 EC2 instâncias da Amazon.

Quando você lança uma nova AmazonEC2instance, o EC2 serviço da Amazon tenta colocar a instância de forma que todas as suas instâncias estejam espalhadas 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 colocar EC2 instâncias da Amazon.

Obrigatório: sim

Tipo: String

Valores possíveis: CLUSTER | PARTITION | SPREAD _ HOST | SPREAD _ RACK

- CLUSTER— agrupa instâncias próximas umas das outras dentro de 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, típico de aplicativos de computação de alto desempenho (). HPC
- PARTITION— distribui suas instâncias em partições lógicas, de forma que grupos de instâncias em uma partição não compartilhem o hardware subjacente com grupos de instâncias em

partições diferentes. Essa estratégia é normalmente usada por grandes workloads distribuídas e replicadas, como Hadoop, Cassandra e Kafka.

- SPREAD_ RACK — coloca um pequeno grupo de instâncias em um 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: número 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 TNBsuporta o lançamento de EC2 instâncias da Amazon com dados personalizados do usuário, por meio do UserData nó no Network Service Descriptor (NSD). Para obter mais

informações sobre dados personalizados do usuário, consulte Dados do usuário e scripts de shell no Guia EC2 do usuário da Amazon.

Durante a instanciação da rede, AWS TNB fornece o registro da EC2 instância Amazon no cluster por meio de um script de dados do usuário. Quando dados personalizados do usuário também são fornecidos, AWS TNB mescla os dois scripts e os transmite como um script multimime para a Amazon. EC2 O script personalizado de dados do usuário é executado antes do script de EKS registro da Amazon.

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 TNBsuporta scripts de dados do usuário de até 7 KB de tamanho.
> - Como 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 AWS CloudFormation regras.

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 TNBoferece suporte a grupos de segurança para automatizar o provisionamento de grupos de EC2segurança da Amazon, 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 AWS.Networking. VPCnodo.

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 TNBsuporta regras de saída de grupos de segurança para automatizar o provisionamento das regras de saída de grupos de segurança da EC2 Amazon, 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 IPv4 endereços em CIDR formato. Você deve especificar um CIDR intervalo que permita o tráfego de saída.

Obrigatório: não

Tipo: string

`cidr_ipv6`

O intervalo de IPv6 endereços em CIDR formato, para tráfego de saída. Você deve especificar um grupo de segurança de destino (`destination_security_groupoudestination_prefix_list`) ou um CIDR intervalo (`cidr_ipoucidr_ipv6`).

Obrigatório: não

Tipo: string

`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: string

`destination_prefix_list`

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

Obrigatório: não

Tipo: string

`from_port`

Se o protocolo for TCP ouUDP, esse será o início do intervalo de portas. Se o protocolo for ICMP ouICMPv6, esse é o número do tipo. Um valor de -1 indica todos os ICMP ICMPv6 /tipos. Se você especificar todos os ICMPv6 tiposICMP/, deverá especificar todos os ICMPv6 códigosICMP//.

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 ouUDP, esse será o fim do intervalo de portas. Se o protocolo for ICMP ouICMPv6, esse é o código. Um valor de -1 indica todos os ICMPv6 códigosICMP//. Se você especificar todos os ICMPv6 tiposICMP/, deverá especificar todos os ICMPv6 códigosICMP//.

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`

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

Obrigatório: não

Tipo: string

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 TNBsuporta regras de entrada de grupos de segurança para automatizar o provisionamento das regras de entrada de grupos de segurança da EC2 Amazon, 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 IPv4 endereços em CIDR formato. Você deve especificar um CIDR intervalo que permita o tráfego de entrada.

Obrigatório: não

Tipo: string

`cidr_ipv6`

O intervalo de IPv6 endereços em CIDR formato, para tráfego de entrada. Você deve especificar um grupo de segurança de origem (`source_security_groupousource_prefix_list`) ou um CIDR intervalo (`cidr_ipoucidr_ipv6`).

Obrigatório: não

Tipo: string

`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: string

`source_prefix_list`

O ID da lista de prefixos de uma lista de prefixos VPC gerenciada existente pela Amazon. 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 gerenciados, consulte Listas de prefixos gerenciadas no Guia VPCdo usuário da Amazon.

Obrigatório: não

Tipo: string

`from_port`

Se o protocolo for TCP ouUDP, esse será o início do intervalo de portas. Se o protocolo for ICMP ouICMPv6, esse é o número do tipo. Um valor de -1 indica todos os ICMP ICMPv6 /tipos. Se você especificar todos os ICMPv6 tiposICMP/, deverá especificar todos os ICMPv6 códigosICMP//.

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 ouUDP, esse será o fim do intervalo de portas. Se o protocolo for ICMP ouICMPv6, esse é o código. Um valor de -1 indica todos os ICMPv6 códigosICMP//. Se você especificar todos os ICMPv6 tiposICMP/, deverá especificar todos os ICMPv6 códigosICMP//.

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

Obrigatório: não

Tipo: string

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 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 AWS TNB.

Obrigatório: Não

Tipo: lista

`resource_id`

O ID do recurso que é importado para 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.Trabalho em rede. ENI

Uma interface de rede é um componente lógico de rede em uma VPC que representa uma placa 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 EC2 instância da Amazon em uma sub-rede, você pode anexar uma interface de rede a ela ou desanexar uma interface de rede dessa instância da Amazon e reconectar-se a outra EC2 instância da Amazon nessa EC2 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: número 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: string

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: string

Valores possíveis: `CODE_BUILD`

Requisitos

`definition`

Um AWS. HookDefinition.Nodo Bash.

Obrigatório: sim

Tipo: String

`vpc`

Um AWS.Networking. VPCnodo.

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 doVPC. Você deve incluir a propriedade `dest_cidr` ou `ipv6_dest_cidr`.

`dest_cidr`

O IPv4 CIDR bloco usado para a partida de 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: string

`ipv6_dest_cidr`

O IPv6 CIDR bloco usado para a partida de destino.

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

Tipo: string

Propriedades

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: Não

Tipo: lista

`egress_only`

Uma propriedade IPv6 específica. 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 AWS.Networking. VPCnodo.

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, chamadas rotas, que determinam para onde o tráfego de rede das sub-redes dentro do seu gateway VPC ou do seu gateway é direcionado. Você deve associar uma tabela de rotas a umaVPC.

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 AWS.Networking. VPCnodo.

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 sua VPC e deve residir inteiramente em uma zona de disponibilidade. Você deve especificar umVPC, um CIDR bloco, 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 IPv4 endereço 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 CIDR bloco da sub-rede.

Obrigatório: não

Tipo: string

`ipv6_cidr_block`

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

Obrigatório: não

Tipo: string

`ipv6_cidr_block_suffix`

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

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

Obrigatório: não

Tipo: string

`outpost_arn`

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

Se você adicionar essa propriedade ao NSD modelo, deverá definir o valor da `availability_zone` propriedade como a Zona de Disponibilidade do AWS Outposts.

Obrigatório: não

Tipo: string

`tags`

As tags a serem anexadas ao recurso.

Obrigatório: Não

Tipo: lista

Requisitos

`vpc`

Um AWS.Networking. VPCnodo.

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.Implantação. VNFDeployment

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

Sintaxe

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

Requisitos

`deployment`

Uma AWS implantação. VNFDeploymentnodo.

Obrigatório: não

Tipo: string

`cluster`

Um AWS computador. EKSnodo.

Obrigatório: sim

Tipo: String

`vnfs`

Um AWS. VNFnodo.

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: string

`post_create`

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

Obrigatório: não

Tipo: string

Exemplo

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

AWS.Trabalho em rede. VPC

Você deve especificar um CIDR bloco 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 alcance IPv4 da rede para oVPC, em CIDR notação.

Obrigatório: sim

Tipo: String

`ipv6_cidr_block`

O IPv6 CIDR bloco usado para criar VPC o.

Valor permitido: `AMAZON_PROVIDED`

Obrigatório: não

Tipo: string

`dns_support`

Indica se as instâncias foram executadas no VPC get DNS hostnames.

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.Trabalho em rede. NATGateway

Você pode definir um nó de NAT gateway público ou privado em uma sub-rede. Para um gateway público, se você não fornecer um ID de alocação de IP elástico, 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: string

`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 IPv4 rotas de destino para o recurso de destino.

Obrigatório: Sim

Tipo: lista

Tipo de membro: string

Propriedades

`nat_gateway`

O AWS.Networking. NATGatewayreferência de nó.

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

Defina nós para NSD VNFD e.

- AWS. HookDefinition.Bash

AWS.HookDefinition.Bash

Define uma AWS HookDefinition entrada`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 AWS TNB reservadas (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 em AWS TNB

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 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.

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

Conteúdo

Proteção de dados em 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 obter mais informações sobre privacidade de dados, consulte Privacidade de dados FAQ. Para obter informações sobre proteção de dados na Europa, consulte o Modelo de Responsabilidade AWS Compartilhada e GDPR a postagem no blog AWS de segurança.

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 a autenticação multifator (MFA) com cada conta.
- UseSSL/TLSpara se comunicar com AWS os recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- 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 FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou umaAPI, use um endpoint. FIPS Para obter mais informações sobre os FIPS endpoints disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com AWS TNB ou Serviços da AWS usa o console,API, AWS CLI, ou AWS SDKs. 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, é altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

Tratamento de dados

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

Criptografia em repouso

AWS TNBsempre 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 TNBprotege 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 TNBos recursos computacionais residem em uma nuvem privada virtual (VPC) compartilhada por todos os clientes. Todo o AWS TNB tráfego interno 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. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar AWS TNB os recursos. IAMé um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Conteúdo

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz AWS TNB.

Usuário do serviço — Se você usar o AWS TNB serviço para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais AWS TNB recursos 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 você não conseguir acessar um recurso no AWS TNB, consulteSolução de problemas de identidade e acesso ao AWS Telco Network Builder.

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

IAMadministrador — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso AWS TNB. Para ver exemplos de políticas AWS TNB baseadas em identidade que você pode usar emIAM, consulte. Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

Autenticando com identidades

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

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do 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 entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. 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 você mesmo assinar solicitações, consulte AWS Signature versão 4 para API solicitações no Guia IAM do usuário.

Independente 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 AWS IAM Identity Center usuário e Autenticação AWS multifator IAM no Guia do IAMusuário.

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 puder executar. Para ver a lista completa de tarefas que exigem que você faça login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do IAM usuário.

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 informações sobre o IAM Identity Center, consulte O que é o IAM Identity Center? no Guia do AWS IAM Identity Center usuário.

Grupos e usuários do IAM

Um IAMusuário é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários 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 IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo no Guia do IAMusuário.

Um IAMgrupo é uma identidade que especifica uma coleção de IAM usuários. 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 IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

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 Casos de uso para IAM usuários no Guia IAM do usuário.

IAMfunções

Uma IAMfunção é uma identidade dentro da sua Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Para assumir temporariamente uma IAM função no AWS Management Console, você pode alternar de um usuário para uma IAM função (console). Você pode assumir uma função chamando uma AWS API operação

AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte Métodos para assumir uma função no Guia IAM do usuário.

IAMfunções 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 informações sobre funções para federação, consulte Criação de uma função para um provedor de identidade terceirizado no Guia IAM do usuário. Se você usa o IAM Identity Center, configura 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 uma função em. IAM Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no Manual do Usuário do AWS IAM Identity Center .
- Permissões temporárias IAM de IAM usuário — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente 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 funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.
- 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 aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FASusa 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. FASas solicitações 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 FAS solicitações, consulte Encaminhar sessões de acesso.

  - Função de serviço — Uma função de serviço é uma IAMfunção que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamenteIAM. Para obter mais informações, consulte Criação de uma função para delegar permissões a uma AWS service (Serviço da AWS) no Guia do IAM usuário.
  - 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 presumir a função 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 IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.
- Aplicativos em execução na Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. Isso é preferível ao armazenamento de chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon no Guia IAM do usuário.

Gerenciando 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 JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte Visão geral das JSON políticas no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas 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 permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar 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 do AWS Management Console AWS CLI, do ou do AWS API.

Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente 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 IAMusuário.

Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso 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 recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

Listas de controle de acesso (ACLs)

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

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber maisACLs, consulte a 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 as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAMusuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso 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 IAM entidades no Guia IAM do usuário.
- Políticas de controle de serviço (SCPs) — SCPs são JSON políticas 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 recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations eSCPs, consulte Políticas de controle de serviços no Guia AWS Organizations do Usuário.
- 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 IAM do usuário.

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 política estão envolvidos, consulte Lógica de avaliação de políticas no Guia IAM do usuário.

Como AWS TNB funciona com IAM

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

IAMrecursos que você pode usar com o AWS Telco Network Builder

IAMrecurso	AWS TNBapoio
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações das políticas	Sim
Atributos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não
ABAC(tags nas políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Não
Funções vinculadas ao serviço	Não

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

Políticas baseadas em identidade para AWS TNB

Compatível com políticas baseadas em identidade: Sim

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia do IAMusuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode usar em uma JSON política, consulte a referência IAM JSON de elementos de política no Guia IAM do usuário.

Exemplos de políticas baseadas em identidade para AWS TNB

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

Políticas baseadas em recursos dentro AWS TNB

Suporte a políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso 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 habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador 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 recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

Ações políticas para AWS TNB

Compatível com ações de políticas: Sim

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

O `Action` elemento de uma JSON política 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 AWS API operação associada. Há algumas exceções, como ações somente de permissão que não têm uma operação correspondente. API Algumas operações também 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 AWS TNB ações, consulte Ações definidas pelo AWS Telco Network Builder na Referência de Autorização de Serviço.

As ações de política 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 AWS TNB baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

Recursos políticos para AWS TNB

Compatível com recursos de políticas: Sim

Os administradores podem usar AWS JSON políticas 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` JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu Amazon Resource Name (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em 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 AWS TNB recursos e seusARNs, consulte Recursos definidos pelo AWS Telco Network Builder na Referência de Autorização de Serviço. Para saber com quais ações você pode especificar cada recurso, consulte Ações definidas pelo AWS Telco Network Builder. ARN

Para ver exemplos de políticas AWS TNB baseadas em identidade, 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 de serviço: Sim

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

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem 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 antes que as permissões da instrução sejam concedidas.

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

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 IAM do usuário.

Para ver uma lista de chaves de AWS TNB condição, 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 AWS TNB baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade do AWS Telco Network Builder

ACLsem AWS TNB

SuportesACLs: Não

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

ABACcom AWS TNB

Suportes ABAC (tags nas políticas): Sim

O controle de acesso baseado em atributos (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 IAM entidades (usuários ou funções) e a muitos AWS recursos. Marcar entidades e recursos é a primeira etapa doABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABACé útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `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 sobreABAC, consulte Definir permissões com ABAC autorização no Guia IAM do usuário. Para ver um tutorial com etapas de configuraçãoABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do IAMusuário.

Usando credenciais temporárias com AWS TNB

Compatível com 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 esse trabalho IAM no Guia do IAM usuário.

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 a troca de funções, consulte Alternar de um usuário para uma IAM função (console) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. 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 obter mais informações, consulte Credenciais de segurança temporárias emIAM.

Permissões principais entre serviços para AWS TNB

Suporta sessões de acesso direto (FAS): Sim

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um

serviço diferente. FASusa 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. FASas solicitações 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 FAS solicitações, consulte Encaminhar sessões de acesso.

Funções de serviço para AWS TNB

Compatível com perfis de serviço: não

Uma função de serviço é uma IAMfunção que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamenteIAM. Para obter mais informações, consulte Criação de uma função para delegar permissões a uma AWS service (Serviço da AWS) no Guia do IAM usuário.

Funções vinculadas a serviços para AWS TNB

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 presumir a função 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 IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas 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 AWS TNB recursos. Eles também não podem realizar tarefas usando o AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte Criar IAM políticas (console) no Guia do IAMusuário.

Para obter detalhes sobre ações e tipos de recursos definidos por AWS TNB, incluindo o formato de 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. ARNs

Conteúdo

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir AWS TNB recursos em sua conta. Essas ações podem incorrer em custos para seus 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 AWS gerenciadas ou políticas AWS gerenciadas para funções de trabalho no Guia IAM do usuário.
- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar 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 IAM como usar para aplicar permissões, consulte Políticas e permissões IAM no Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usandoSSL. 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 IAM JSON da política: Condição no Guia IAM do usuário.

- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM IAMO Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte Validar políticas com o IAM Access Analyzer no Guia do IAMUsuário.
- Exigir autenticação multifatorial (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte APIAcesso seguro com MFA no Guia IAM do usuário.

Para obter mais informações sobre as melhores práticas emIAM, consulte as melhores práticas de segurança IAM no Guia IAM do usuário.

Usando o AWS TNB console

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 AWS TNB recursos 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 AWS CLI o. ou AWS API o. Em vez disso, permita o acesso somente às ações que correspondam à API operação que eles estão tentando realizar.

Exemplos de política de perfil de serviço

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

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

Você cria a função de serviço e sua política de permissão com o IAM serviço. 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 IAM usuário.

AWS TNBfunção de serviço

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

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

O código a seguir mostra uma política AWS TNB de função de serviço:

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

```
            "Effect": "Allow",
            "Sid": "IAMPolicy"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "eks.amazonaws.com",
                        "eks-nodegroup.amazonaws.com"
                    ]
                }
            },
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBAccessSLRPermissions"
        },
        {
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateOrUpdateTags",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteTags",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:DescribeTags",
                "autoscaling:UpdateAutoScalingGroup",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteLaunchTemplateVersions",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeTags",
                "ec2:GetLaunchTemplateData",
                "ec2:RevokeSecurityGroupEgress",
```

```
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RunInstances",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:CreateInternetGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSubnet",
                "ec2:CreateTags",
                "ec2:CreateVpc",
                "ec2:DeleteInternetGateway",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSubnet",
                "ec2:DeleteTags",
                "ec2:DeleteVpc",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeInstances",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySecurityGroupRules",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:AllocateAddress",
                "ec2:AssignIpv6Addresses",
                "ec2:AssociateAddress",
                "ec2:AssociateNatGatewayAddress",
                "ec2:AssociateVpcCidrBlock",
                "ec2:CreateEgressOnlyInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:DeleteEgressOnlyInternetGateway",
                "ec2:DeleteNatGateway",
                "ec2:DescribeAddresses",
                "ec2:DescribeEgressOnlyInternetGateways",
                "ec2:DescribeNatGateways",
```

```
                "ec2:DisassociateAddress",
                "ec2:DisassociateNatGatewayAddress",
                "ec2:DisassociateVpcCidrBlock",
                "ec2:ReleaseAddress",
                "ec2:UnassignIpv6Addresses",
                "ec2:DescribeImages",
                "eks:CreateCluster",
                "eks:ListClusters",
                "eks:RegisterCluster",
                "eks:TagResource",
                "eks:DescribeAddonVersions",
                "events:DescribeRule",
                "iam:GetRole",
                "iam:ListAttachedRolePolicies",
                "iam:PassRole"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBAccessComputePerms"
        },
        {
            "Action": [
                "codebuild:BatchDeleteBuilds",
                "codebuild:BatchGetBuilds",
                "codebuild:CreateProject",
                "codebuild:DeleteProject",
                "codebuild:ListBuildsForProject",
                "codebuild:StartBuild",
                "codebuild:StopBuild",
                "events:DeleteRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "s3:CreateBucket",
                "s3:GetBucketAcl",
                "s3:GetObject",
                "eks:DescribeNodegroup",
                "eks:DeleteNodegroup",
                "eks:AssociateIdentityProviderConfig",
                "eks:CreateNodegroup",
                "eks:DeleteCluster",
                "eks:DeregisterCluster",
                "eks:UpdateAddon",
                "eks:UpdateClusterVersion",
```

```
                "eks:UpdateNodegroupConfig",
                "eks:UpdateNodegroupVersion",
                "eks:DescribeUpdate",
                "eks:UntagResource",
                "eks:DescribeCluster",
                "eks:ListNodegroups",
                "eks:CreateAddon",
                "eks:DeleteAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonVersions",
                "s3:PutObject",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateStack",
                "cloudformation:UpdateTerminationProtection"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/tnb*",
                "arn:aws:codebuild:*:*:project/tnb*",
                "arn:aws:logs:*:*:log-group:/aws/tnb*",
                "arn:aws:s3:::tnb*",
                "arn:aws:eks:*:*:addon/tnb*/*/*",
                "arn:aws:eks:*:*:cluster/tnb*",
                "arn:aws:eks:*:*:nodegroup/tnb*/tnb*/*",
                "arn:aws:cloudformation:*:*:stack/tnb*"
            ],
            "Effect": "Allow",
            "Sid": "TNBAccessInfraResourcePerms"
        },
        {
            "Sid": "CFNTemplatePerms",
            "Effect": "Allow",
            "Action": [
              "cloudformation:GetTemplateSummary"
            ],
            "Resource": "*"
         },
        {
            "Sid": "ImageAMISSMPerms",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
```

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

O código a seguir mostra a política AWS TNB de confiança do serviço:

```
{
  "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 TNBfunção de serviço para o EKS cluster Amazon

Ao criar um EKS recurso da Amazon no seuNSD, você fornece o `cluster_role` atributo para especificar qual função será usada para criar seu EKS cluster da Amazon.

O exemplo a seguir mostra um AWS CloudFormation modelo que cria uma função AWS TNB de serviço para a política de EKS cluster da Amazon.

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBEKSClusterRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBEKSClusterRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
```

```
            Principal:
              Service:
                - eks.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSClusterPolicy"
```

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

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

AWS TNBfunção de serviço para o grupo de EKS nós da Amazon

Ao criar recursos de um grupo de EKS nós da Amazon em seuNSD, você fornece o `node_role` atributo para especificar qual função será usada para criar seu grupo de EKS nós da Amazon.

O exemplo a seguir mostra um AWS CloudFormation modelo que cria uma função AWS TNB de serviço para a política de grupo de EKS nós da Amazon.

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBEKSNodeRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBEKSNodeRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSWorkerNodePolicy"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKS_CNI_Policy"
```

```
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEBSCSIDriverPolicy"
      Policies:
        - PolicyName: EKSNodeRoleInlinePolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "logs:DescribeLogStreams"
                  - "logs:PutLogEvents"
                  - "logs:CreateLogGroup"
                  - "logs:CreateLogStream"
                Resource: "arn:aws:logs:*:*:log-group:/aws/tnb/tnb*"
        - PolicyName: EKSNodeRoleIpv6CNIPolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "ec2:AssignIpv6Addresses"
                Resource: "arn:aws:ec2:*:*:network-interface/*"
```

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

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

AWS TNBfunção de serviço para Multus

Quando você cria um EKS recurso da Amazon no seu NSD e deseja gerenciar o Multus como parte do seu modelo de implantação, você deve fornecer o `multus_role` atributo para especificar qual função será usada para gerenciar o Multus.

O exemplo a seguir mostra um AWS CloudFormation modelo que cria uma função AWS TNB de serviço 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 IAM funções usando o AWS CloudFormation modelo, consulte as seções a seguir no Guia AWS CloudFormation do usuário:

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

AWS TNBfunção de serviço para uma política de gancho de ciclo de vida

Quando seu pacote de funções NSD ou de rede usa um gancho de ciclo de vida, você precisa de uma função de serviço que permita criar um ambiente para a execução de seus ganchos 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 AWS TNB serviço 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 IAM funções usando o AWS CloudFormation modelo, consulte as seções a seguir no Guia AWS CloudFormation do usuário:

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

Permitir que os usuários exibam as próprias permissões

Este exemplo mostra como você pode criar uma política que permita IAM aos usuários visualizar as políticas embutidas e gerenciadas que estão anexadas à identidade do usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou. AWS API

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

```
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

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 AWS TNB e. IAM

Problemas

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

Não estou autorizado a realizar uma ação em 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 exemplo de erro a seguir ocorre quando o `mateojackson` IAM usuário tenta usar o console para ver detalhes sobre um *`my-example-widget`* recurso fictício, mas não tem as permissões fictícias`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 transfira uma função para AWS TNB o.

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 exemplo de erro a seguir ocorre quando um IAM usuário chamado `marymajor` tenta usar o console para realizar 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 Conta da AWS acessem meus AWS TNB recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a 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 é AWS TNB compatível com esses recursos, consulteComo AWS TNB funciona com IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Fornecer acesso a um IAM usuário em outro Conta da AWS de sua propriedade no Guia do IAM usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Fornecer Contas da AWS acesso a terceiros no Guia do IAM usuário.

- Para saber como fornecer acesso por meio da federação de identidades, consulte Fornecendo acesso a usuários autenticados externamente (federação de identidades) no Guia do IAMusuário.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

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 HIPAA segurança e conformidade na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos qualificados.

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

- 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 recursos 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.
- Amazon GuardDuty — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- 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 em AWS TNB

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 aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data centers tradicionais.

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

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

Segurança da infraestrutura em 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 API chamadas AWS publicadas para acessar AWS TNB pela rede. Os clientes devem oferecer suporte para:

- Segurança da camada de transporte (TLS). Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Suítes de criptografia com sigilo direto perfeito (), como (Ephemeral PFS Diffie-Hellman) ou DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. Ou você pode 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 AWS TNB, incluindo:
  - Instâncias de computação (também conhecidas como trabalhadores)
  - Bancos de dados internos
  - Comunicações de rede entre componentes internos
  - A interface AWS TNB de programação de aplicativos (API)
  - 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):
  - IAMusuários, grupos, funções e políticas
  - Buckets S3 que você usa para armazenar seus dados para AWS TNB
  - Outros recursos Serviços da AWS e recursos que você usa para dar suporte ao serviço de rede que você provisionou por meio AWS TNB
  - Código da sua aplicação
  - Conexões entre o serviço de rede que você provisionou AWS TNB e seus clientes

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

Modelo de segurança de conectividade de rede

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

As comunicações entre seus clientes de serviços de rede e seu serviço de rede AWS TNB atravessam a Internet. AWS TNBnão gerencia essas conexões. É sua responsabilidade proteger as conexões de seus clientes.

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

IMDSversão

AWS TNBoferece suporte a instâncias que utilizam o Instance Metadata Service versão 2 (IMDSv2), um método orientado a sessões. IMDSv2inclui maior segurança do queIMDSV1. Para obter mais informações, consulte Adicionar defesa aprofundada contra firewalls abertos, proxies reversos e SSRF vulnerabilidades com aprimoramentos no Amazon Instance Metadata Service. EC2

Ao iniciar sua instância, você deve usarIMDSv2. Para obter mais informações sobreIMDSv2, consulte Use IMDSv2 no Guia do EC2 usuário da Amazon.

Monitoramento AWS TNB

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de AWS TNB suas outras AWS soluções. AWS permite AWS CloudTrail observar AWS TNB, denunciar quando algo está errado e realizar ações automáticas quando apropriado.

Use CloudTrail para capturar informações detalhadas sobre as chamadas feitas para AWS APIs o. Você pode armazenar essas chamadas como arquivos de log no Amazon S3. Você pode usar esses CloudTrail registros para determinar informações como qual chamada foi feita, o endereço IP de origem de onde veio a chamada, quem fez a chamada e quando a chamada foi feita.

Os CloudTrail registros contêm informações sobre as chamadas para API ações do AWS TNB. Eles também contêm informações para chamadas para API ações de serviços como Amazon EC2 e AmazonEBS.

Registrando API chamadas do AWS Telco Network Builder usando AWS CloudTrail

AWS O Telco Network Builder é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou um AWS service (Serviço da AWS). CloudTrail captura todas as API chamadas para AWS TNB eventos. As chamadas capturadas incluem chamadas do AWS TNB console e chamadas de código para as AWS TNB API operações. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita AWS TNB, o endereço IP do qual a solicitação foi feita, quando foi feita e detalhes adicionais.

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 credenciais de usuário.
- Se a solicitação foi feita em nome de um usuário do IAM Identity Center.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS service (Serviço da AWS).

CloudTrail está ativo Conta da AWS quando você cria a conta e você tem acesso automático ao histórico de CloudTrail eventos. O histórico de CloudTrail eventos fornece um registro visível,

pesquisável, baixável e imutável dos últimos 90 dias de eventos de gerenciamento registrados em um. Região da AWS Para obter mais informações, consulte Trabalhando com o histórico de CloudTrail eventos no Guia AWS CloudTrail do usuário. Não há CloudTrail cobrança pela visualização do histórico de eventos.

Para um registro contínuo dos eventos dos Conta da AWS últimos 90 dias, crie uma trilha ou um armazenamento de dados de eventos do CloudTrailLake.

CloudTrail trilhas

Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Todas as trilhas criadas usando o AWS Management Console são multirregionais. Só é possível criar uma trilha de região única ou de várias regiões usando a AWS CLI. É recomendável criar uma trilha multirregional porque você captura todas as atividades Regiões da AWS em sua conta. Se você criar uma trilha de região única, poderá visualizar somente os eventos registrados na Região da AWS da trilha. Para obter mais informações sobre trilhas, consulte Criar uma trilha para a Conta da AWS e Criar uma trilha para uma organização no Guia do usuário do AWS CloudTrail .

Você pode entregar uma cópia dos seus eventos de gerenciamento contínuos para o bucket do Amazon S3 sem nenhum custo CloudTrail criando uma trilha. No entanto, há cobranças de armazenamento do Amazon S3. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços. Para receber informações sobre a definição de preço do Amazon S3, consulte Definição de preço do Amazon S3.

CloudTrail Armazenamentos de dados de eventos em Lake

CloudTrail O Lake permite que você execute consultas SQL baseadas em seus eventos. CloudTrail O Lake converte eventos existentes em JSON formato baseado em linhas para o formato ORCApache. ORCé um formato de armazenamento colunar otimizado para recuperação rápida de dados. Os eventos são agregados em armazenamentos de dados de eventos, que são coleções imutáveis de eventos baseados nos critérios selecionados com a aplicação de seletores de eventos avançados. Os seletores que você aplica a um armazenamento de dados de eventos controlam quais eventos persistem e estão disponíveis para você consultar. Para obter mais informações sobre o CloudTrail Lake, consulte Trabalhando com o AWS CloudTrail Lake no Guia AWS CloudTrail do Usuário.

CloudTrail Os armazenamentos e consultas de dados de eventos em Lake incorrem em custos. Ao criar um armazenamento de dados de eventos, você escolhe a opção de preço que deseja usar para ele. A opção de preço determina o custo para a ingestão e para o armazenamento de

eventos, e o período de retenção padrão e máximo para o armazenamento de dados de eventos. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços.

Exemplos de eventos do AWS TNB

Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a API operação solicitada, a data e a hora da operação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das API chamadas públicas, portanto, os eventos não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra um CloudTrail evento que demonstra a `CreateSolFunctionPackage` operação.


```
{
    "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"
}
```


Para obter informações sobre o conteúdo do CloudTrail registro, consulte o conteúdo do CloudTrail registro no Guia AWS CloudTrail do usuário.

AWS TNBtarefas de implantação

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

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

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 da AmazonEKS.
AppUpdate	nenhuma mudança no nome	Atualiza as funções de rede que já estão instaladas em uma instância de rede.

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
-	ClusterPluginUninstall	Desinstala os plug-ins no cluster da AmazonEKS.
ClusterStorageClassConfiguration	nenhuma mudança no nome	Configura a classe de armazenamento (CSIdriver) em um EKS cluster da Amazon.
FunctionDeletion	nenhuma mudança no nome	Exclui as funções de rede dos AWS TNB recursos.
FunctionInstantiation	FunctionInstall	Implanta funções de rede usandoHELM.
FunctionUninstallation	FunctionUninstall	Desinstala a função de rede de um cluster da AmazonEKS.
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 de. AWS TNB
-	InfrastructureUpdate	Atualiza os AWS recursos provisionados em nome do usuário.
InventoryDeregistration	nenhuma mudança no nome	Cancela o registro de recursos de. AWS AWS TNB
-	InventoryRegistration	Registra os AWS recursos em AWS 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
KubernetesClusterConfiguration	ClusterConfiguration	Configura o cluster Kubernetes e adiciona funções adicionais IAM à 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 plano de controle Amazon EKS e Kubernetes.
-	ValidateNetworkServiceUpdate	Executa as validações antes de atualizar uma instância de rede.

Cotas de serviço para AWS TNB

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

A seguir estão as cotas de serviço para AWS TNB.

Nome	Padrão	Ajustá	Descrição
Operações simultâneas de serviços de rede contínuos	Cada região compatível: 40	Sim	Define o número máximo de operações de serviços de rede simultâneas 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 de serviço de rede em uma região.

Histórico do documento para o guia AWS TNB do usuário

A tabela a seguir descreve as versões de documentação do AWS TNB.

Alteração	Descrição	Data
Versão do Kubernetes para cluster	AWS TNBagora oferece suporte ao Kubernetes versão 1.30 para criar clusters da Amazon. EKS	19 de agosto de 2024

AWS TNBsuporta uma operação adicional para gerenciar o ciclo de vida da rede.	Você pode atualizar uma instância de rede instancia da ou atualizada anteriorm ente com um novo pacote de rede e valores de parâmetros. Consulte:
	• Operações do ciclo de vida
	• Atualizar uma instância de rede
	• AWS TNBexemplo de função de serviço:
	  • Adicione essas EKS ações da Amazon:`eks:Updat eAddon` ,`eks:Updat eClusterV ersion` ,`eks:Updat eNodegrou pConfig` ,`eks:Updat eNodegroupVersion` , `eks:DescribeUpdate`
	  • Adicione esta AWS CloudFormation ação:
		30 de julho de 2024

	`cloudformation:UpdateStack` • Novas tarefas de implantação:`InfrastructureUpdate` ,`Inventory Registration` , `ValidateNetworkServiceUpdate` • APIatualizações: GetSolNetworkOperation, ListSolNetworkOperations, e UpdateSolNetworkInstance	
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 TNBagora oferece suporte ao Kubernetes versão 1.29 para criar clusters da Amazon. EKS	10 de abril de 2024
Support para interface de rede `security_groups`	Você pode anexar grupos de segurança ao AWS.Networking. ENInodo.	2 de abril de 2024
Support para criptografia de volume EBS raiz da Amazon	Você pode ativar a EBS criptografia da Amazon para o volume EBS raiz da Amazon. Para habilitar, adicione as propriedades em AWS.Compute. EKSManagedNodeou AWS.Compute. EKSSelfManagedNodenodo.	2 de abril de 2024

Support para node `labels`	Você pode anexar rótulos de nós ao seu grupo de nós no AWS.Compute. EKSManagedNodeou AWS.Compute. EKSSelfManagedNodenodo.	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 .Networking. AWS ENInodo.	25 de janeiro de 2024
Support para EC2 instâncias da Amazon com dados de usuário personalizados	Você pode iniciar EC2 instâncias da Amazon com dados de usuário personalizados por meio do AWS.Compute. UserData nodo.	16 de janeiro de 2024
Suporte a grupo de segurança	AWS TNBpermite importar 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 AWS.Compute. EKSManagedNodeou AWS.Compute. EKSSelfManagedNodenode, AWS TNB obtém a permissão relacionada à ENIs `multus_role` propriedade, se disponível, ou à `node_role` propriedade.	18 de dezembro de 2023

Suporte a cluster privado	AWS TNBagora 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 TNBagora oferece suporte ao Kubernetes versão 1.28 para criar clusters da Amazon. EKS	11 de dezembro de 2023
AWS TNBsuporta 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 TNBadiciona suporte para IPv6	AWS TNBagora oferece suporte à criação de instâncias de rede com IPv6 infraestrutura. Verifique os nós AWS.Networking. VPC, AWS.Rede.Sub-rede, .Rede.AWS InternetGateway, AWS.Rede. SecurityGroupIngressRule, AWS.Rede. SecurityGroupEgressRule, e AWS.Compute. EKSpara IPv6 configurações. Também adicionamos os nós AWS.Networking. NATGateway e AWS.Networking.Route para configuração. NAT64 Atualizamos a função AWS TNB de serviço e a função AWS TNB de serviço do grupo de EKS nós da Amazon para obter IPv6 permissões. Consulte Service role policy examples.	16 de novembro de 2023
Permissões adicionadas à política AWS TNB de função de serviço	Adicionamos permissões à política de função AWS TNB de serviço do Amazon S3 e para permitir AWS CloudFormation a instancia ção da infraestrutura.	23 de outubro de 2023

AWS TNBlançado em mais regiões	AWS TNBagora 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. EKSSelfManagedNode	AWS TNBagora suporta tags para a definição do `AWS.Compute.EKSSelfManagedNode` nó.	22 de agosto de 2023
AWS TNBsuporta instâncias que aproveitam IMDSv2	Ao iniciar sua instância, você deve usarIMDSv2.	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 TNBagora oferece suporte às versões 1.27 do Kubernetes para criar clusters da Amazon. EKS	25 de julho de 2023
AWS.Computação. EKS. AuthRole	AWS TNBsuportes AuthRole que permitem adicionar IAM funções ao EKS cluster `aws-auth ConfigMap` da Amazon para que os usuários possam acessar o EKS cluster da Amazon usando uma IAM função.	19 de julho de 2023

AWS TNBsuporta grupos de segurança.	Adicionou o AWS.Networking. SecurityGroup, AWS.Trabalho em rede. SecurityGroupEgress sRule, e AWS.Networking. SecurityGroupIngressRulepara o NSD modelo.	18 de julho de 2023
Versão do Kubernetes para cluster	AWS TNBsuporta as versões 1.22 a 1.26 do Kubernetes para criar clusters da Amazon. EKS AWS TNBnão é mais compatível com as versões 1.21 do Kubernetes.	11 de maio de 2023
AWS.Computação. EKSSelfManagedNode	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 AWS TNB do Usuário.	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.