Guía del usuario

AWS Creador de redes de telecomunicaciones

AWS Creador de redes de telecomunicaciones: Guía del usuario

Table of Contents

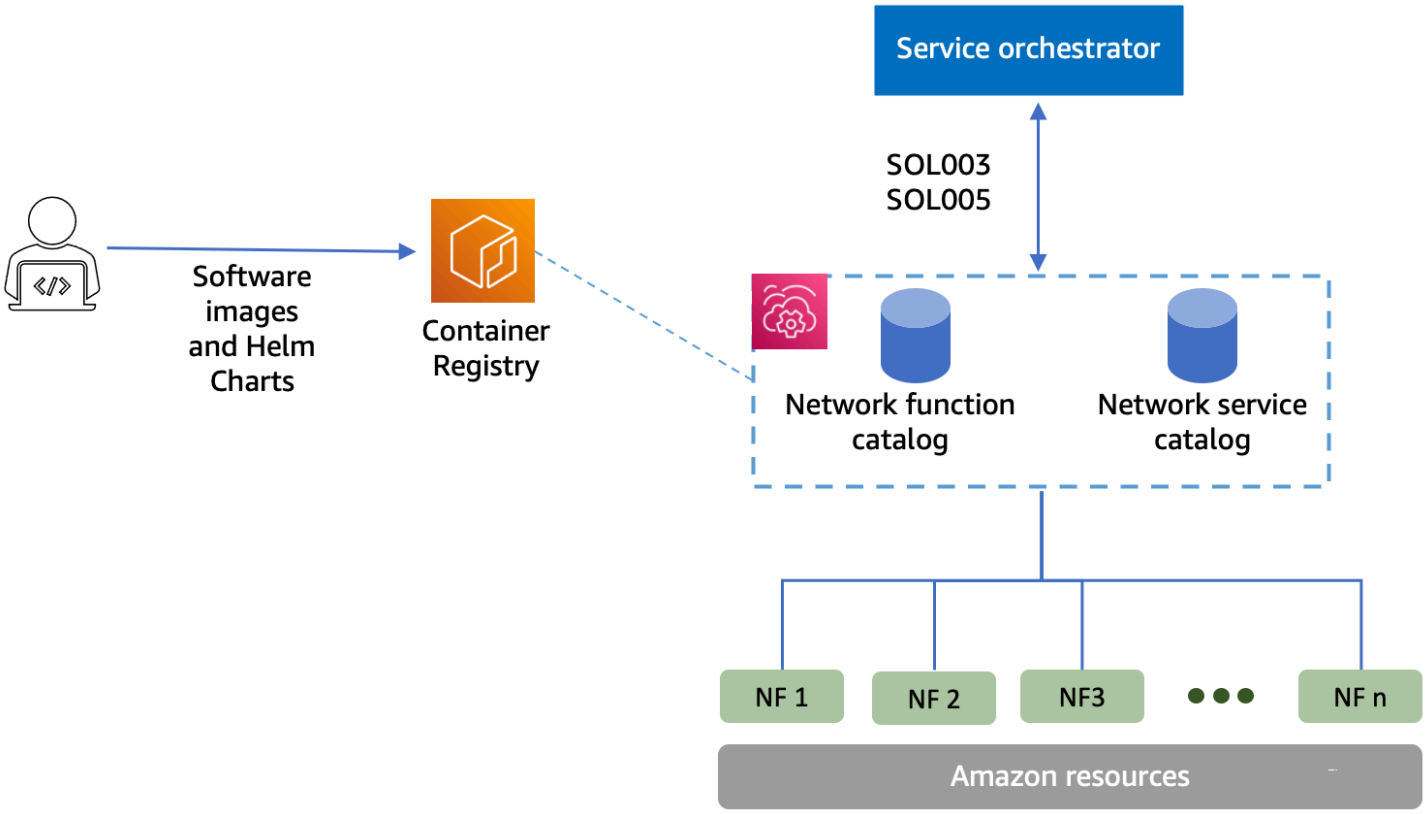
¿Qué es AWS Telco Network Builder?

AWS Telco Network Builder (AWS TNB) es un AWS servicio que proporciona a los proveedores de servicios de comunicación (CSPs) una forma eficiente de implementar, administrar y escalar redes 5G en la AWS infraestructura.

Con AWS TNB él, usted despliega redes 5G escalables y seguras Nube de AWS utilizando una imagen de su red de forma automatizada. No necesita aprender nuevas tecnologías, decidir qué servicio de cómputo usar ni saber cómo aprovisionar y configurar AWS los recursos.

En su lugar, debe describir la infraestructura de su red y proporcionar las imágenes de software de las funciones de la red de sus socios proveedores de software independientes. ISV AWS TNBse integra con orquestadores de AWS servicios y servicios de terceros para aprovisionar automáticamente la AWS infraestructura necesaria, implementar funciones de red en contenedores y configurar la administración de redes y accesos para crear un servicio de red totalmente operativo.

El siguiente diagrama ilustra las integraciones lógicas entre los orquestadores de servicios AWS TNB y los que implementan las funciones de red mediante interfaces estándar basadas en el Instituto Europeo de Normas de Telecomunicación (). ETSI



Temas

¿Nuevo en? AWS

Si es la primera vez que AWS conoce los productos y servicios, comience a aprender más con los siguientes recursos:

- Introducción a AWS
- Cómo empezar con AWS

¿Para quién es AWS TNB?

AWS TNBCSPsbusca aprovechar la rentabilidad, la agilidad y la elasticidad que Nube de AWS ofrece sin tener que escribir ni mantener scripts y configuraciones personalizados para diseñar, implementar y administrar los servicios de red. AWS TNBaprovisiona automáticamente la AWS infraestructura necesaria, despliega funciones de red en contenedores y configura la administración de redes y accesos para crear servicios de red totalmente operativos basados en los descriptores de servicios de red CSP definidos y en las funciones de red que se desean implementar. CSP

AWS TNBcaracterísticas

Las siguientes son algunas de las razones por las que a CSP querría utilizarlas AWS TNB:

Ayuda a simplificar tareas

> Proporcione una mayor eficiencia a sus operaciones de red, como la implementación de nuevos servicios, la actualización y mejora de las funciones de la red y el cambio de las topologías de la infraestructura de red.

Se integra con los orquestadores

AWS TNBse integra con los populares orquestadores de servicios de terceros que cumplen ETSI con los requisitos.

Escalas

Puede configurarlo AWS TNB para escalar AWS los recursos subyacentes a fin de satisfacer la demanda de tráfico, actualizar las funciones de la red de manera más eficiente, implementar cambios en la topología de la infraestructura de red y reducir el tiempo de implementación de los nuevos servicios 5G de días a horas.

Inspecciona y monitorea los recursos AWS

AWS TNBle permite inspeccionar y supervisar los AWS recursos que dan soporte a su red en un único panel, como Amazon VPCEC2, Amazon y AmazonEKS.

Admite plantillas de servicio

AWS TNBle permite crear plantillas de servicios para todas las cargas de trabajo de telecomunicaciones (RAN, Core,IMS). Puede crear una nueva definición de servicio, reutilizar una plantilla existente o integrarla con una canalización de integración y entrega continuas (CI/CD) para publicar una nueva definición.

Realiza un seguimiento de los cambios en las implementaciones de red

Al cambiar la configuración subyacente de un despliegue de funciones de red, por ejemplo, al cambiar el tipo de instancia de un tipo de EC2 instancia de Amazon, puede realizar un seguimiento de los cambios de forma repetible y escalable. Hacerlo manualmente requeriría administrar el estado de la red, crear y eliminar recursos y prestar atención al orden de los cambios necesarios. Cuando se utiliza AWS TNB para gestionar el ciclo de vida de la función de red, solo se realizan los cambios en los descriptores de los servicios de red que describen la función de red. AWS TNBentonces realizará automáticamente los cambios necesarios en el orden correcto.

Simplifica el ciclo de vida de las funciones de red

Puede administrar la primera versión y todas las versiones posteriores de una función de red y especificar cuándo realizar la actualización. También puede administrar sus RAN aplicaciones principales y de red de la misma manera. IMS

Accediendo AWS TNB

Puede crear, acceder y administrar sus AWS TNB recursos mediante cualquiera de las siguientes interfaces:

- AWS TNBconsola: proporciona una interfaz web para administrar la red.
- AWS TNBAPI— Proporciona una RESTful API forma de realizar AWS TNB acciones. Para obtener más información, consulte AWS TNBAPIla Referencia
- AWS Command Line Interface (AWS CLI): proporciona comandos para un amplio conjunto de AWS servicios, que incluyen AWS TNB. Es compatible con Windows, macOS y Linux. Para obtener más información, consulte AWS Command Line Interface.
- AWS SDKs— Proporciona un idioma específico APIs y completa muchos de los detalles de la conexión. Incluyen cálculos de firmas, control de reintentos de solicitud y control de errores. Para obtener más información, consulte. AWSSDKs

Precios para AWS TNB

AWS TNBayuda a CSPs automatizar el despliegue y la administración de sus redes de telecomunicaciones en AWS. Usted paga por las dos dimensiones siguientes cuando las utiliza AWS TNB:

- Por elemento de función de red gestionada (MNFI) horas.
- Por número de API solicitudes.

También incurrirá en cargos adicionales al utilizar otros AWS servicios junto con AWS TNB. Para obtener más información, consulta los AWS TNBprecios.

Para ver su factura, vaya al Panel de Billing and Cost Management en la consola de AWS Billing and Cost Management. La factura contiene enlaces a informes de uso que ofrecen detalles sobre la cuenta. Para obtener más información sobre la facturación de la AWS cuenta, consulta Facturación de la AWS cuenta.

Si tienes preguntas sobre la AWS facturación, las cuentas y los eventos, ponte en contacto con AWS Support.

AWS Trusted Advisor es un servicio que puede utilizar para ayudar a optimizar los costes, la seguridad y el rendimiento de su AWS entorno. Para obtener más información, consulte AWS Trusted Advisor.

Siguientes pasos

Para obtener más información sobre cómo empezar AWS TNB, consulte los siguientes temas:

- Configuración AWS TNB: completar los pasos de requisitos previos.
- Empezar con AWS TNB— Implemente su primera función de red, como la unidad centralizada (CU), la función de gestión del acceso y la movilidad (AMF), la función de plano de usuario (UPF) o un núcleo 5G completo.

Cómo AWS TNB funciona

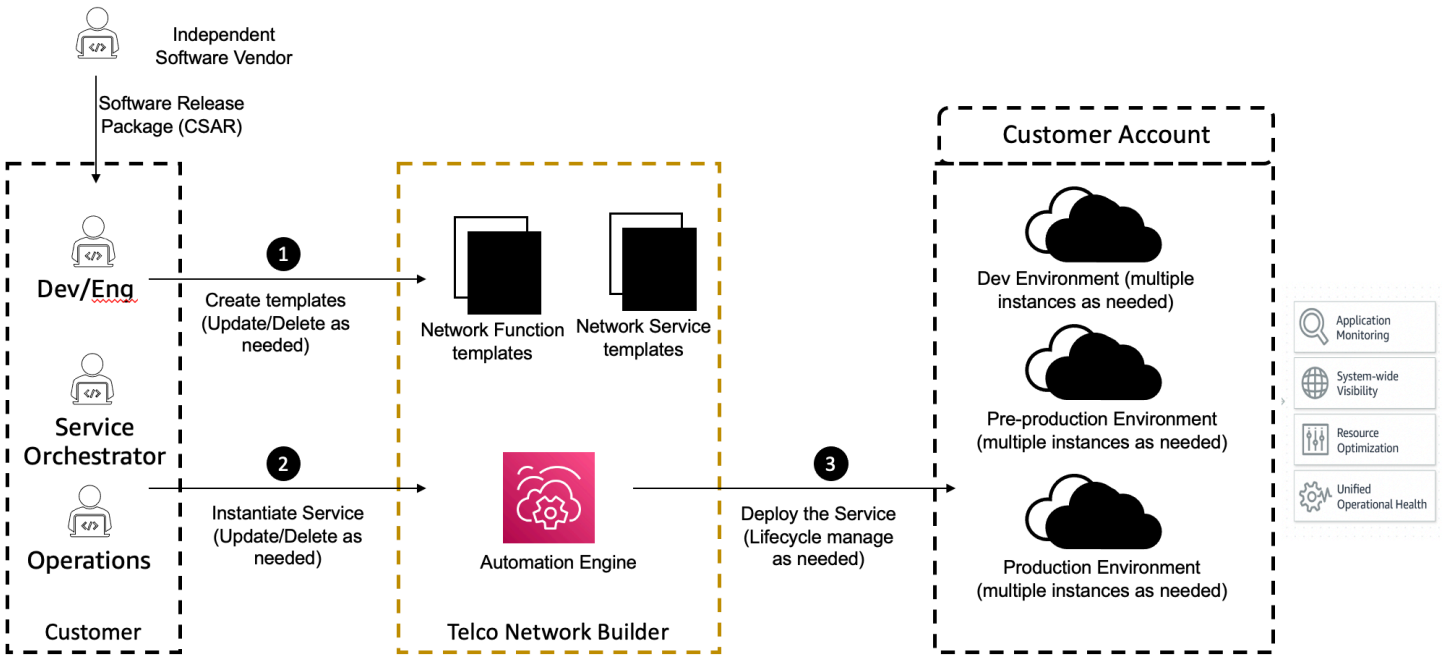
AWS TNBse integra con end-to-end orquestadores y AWS recursos estandarizados para operar redes 5G completas.

AWS TNBle permite incorporar paquetes de funciones de red y descriptores de servicios de red (NSDs) y le proporciona el motor de automatización necesario para operar sus redes. Puede usar su end-to-end orquestador e integrarlo o usarlo AWS TNB SDKs para crear su propio flujo de automatización. AWS TNB APIs Para obtener más información, consulte AWS TNBarquitectura .

Temas

AWS TNBarquitectura

AWS TNBle proporciona la capacidad de realizar operaciones de administración del ciclo de vida a través de AWS Management Console AWS CLI, AWS TNB RESTAPI, ySDKs. Esto permite que las diferentes CSP personas, como los miembros de los equipos de ingeniería, operaciones y sistemas programáticos, lo aprovechen. AWS TNB Puede crear y cargar un paquete de funciones de red como un archivo de Cloud Service Archive (CSAR). El CSAR archivo contiene gráficos de Helm, imágenes de software y un descriptor de funciones de red (NFD). Puede utilizar plantillas para implementar varias configuraciones de ese paquete de forma repetida. Puede crear plantillas de servicios de red que definen la infraestructura y las funciones de red que desea implementar. Puede utilizar las anulaciones de parámetros para implementar diferentes configuraciones en diferentes ubicaciones. A continuación, puede crear instancias de una red mediante las plantillas e implementar las funciones de la red en la infraestructura. AWS AWS TNBle proporciona la visibilidad de sus despliegues.



Integración con Servicios de AWS

Una red 5G se compone de un conjunto de funciones de red en contenedores interconectadas que se despliegan en miles de clústeres de Kubernetes. AWS TNBse integra con lo siguiente Servicios de AWS como elemento específico de las telecomunicaciones para crear un servicio de red totalmente operativo: APIs

- Amazon Elastic Container Registry (AmazonECR) para almacenar artefactos de funciones de red de proveedores de software independientes (ISVs).
- Amazon Elastic Kubernetes Service (AmazonEKS) para configurar clústeres.
- Amazon VPC para construcciones de redes.
- Grupos de seguridad que utilizan AWS CloudFormation.
- AWS CodePipeline para objetivos de despliegue en todas Regiones de AWS las Zonas AWS Locales y AWS Outposts.
- IAMpara definir las funciones.
- AWS Organizations para controlar el acceso a AWS TNBAPIs.
- AWS Health Dashboard y AWS CloudTrail para monitorear el estado y publicar las métricas.

AWS TNBcuotas de recursos

Cuenta de AWS Tiene cuotas predeterminadas, anteriormente denominadas límites, para cada uno de ellos Servicio de AWS. A menos que se indique lo contrario, cada cuota es específica de un Región de AWS. Puede solicitar el aumento de algunas cuotas, pero no de todas.

Para ver las cuotas AWS TNB, abra la consola Service Quotas. En el panel de navegación Servicios de AWS, elija y seleccione AWS TNB.

Para solicitar un aumento de cuota, consulte Solicitud de un aumento de cuota en la Guía de usuario de Service Quotas.

Cuenta de AWS Tiene las siguientes cuotas relacionadas con AWS TNB.

Cuota de recursos	Descripción	Valor predeterminado	¿Ajustable?
Instancias de servicios de red	El número máximo de instancias de servicios de red por región.	800	Sí
Operaciones de servicios de red simultáneas y continuas	Establece el número máximo de operaciones de red simultáneas en curso en una región.	40	Sí
Paquetes de red	Establece el número máximo de paquetes de red en una región.	40	Sí
Paquetes de funciones	El número máximo de paquetes de funciones en una región.	200	Sí

AWS TNBconceptos

En este tema se describen los conceptos esenciales que le ayudarán a empezar a utilizarlos AWS TNB.

Contenido

Ciclo de vida de una función de red

AWS TNBle ayuda a lo largo del ciclo de vida de las funciones de su red. El ciclo de vida de las funciones de red incluye las siguientes etapas y actividades:

Planificación

1. Planifique su red identificando las funciones de red que se van a implementar.
2. Coloque las imágenes del software de funciones de red en un repositorio de imágenes contenedor.
3. Cree los CSAR paquetes para implementarlos o actualizarlos.
4. AWS TNBUtilícelo para cargar el CSAR paquete que define su función de red (por ejemploAMF, CU yUPF) e intégrelo con una canalización de integración y entrega continuas (CI/CD) que le ayude a crear nuevas versiones del CSAR paquete a medida que estén disponibles nuevas imágenes de software de funciones de red o scripts de clientes.

Configuración

1. Identifique la información necesaria para la implementación, como el tipo de cómputo, la versión de la función de red, la información de IP y los nombres de los recursos.
2. Utilice la información para crear el descriptor de servicio de red (). NSD
3. Ingiera los NSDs datos que definan las funciones de la red y los recursos necesarios para la creación de instancias de la función de red.

Instanciación

1. Cree la infraestructura que requieren las funciones de la red.
2. Instancie (o aprovisione) la función de red tal como se define en ella y comience a transportar tráfico. NSD
3. Valide los activos.

Producción

Durante el ciclo de vida de la función de red, completará operaciones de producción tales como:

- Actualice la configuración de la función de red, por ejemplo, actualice un valor en la función de red implementada.
- Actualice la instancia de red con un nuevo paquete de red y valores de parámetros. Por ejemplo, actualice el EKS `version` parámetro Amazon en el paquete de red.

Utilizar interfaces estandarizadas

AWS TNBse integra con los orquestadores de servicios compatibles con el Instituto Europeo de Normas de Telecomunicación (ETSI), lo que le permite simplificar el despliegue de sus servicios de red. Los orquestadores de servicios pueden utilizar AWS TNB SDKs el CLI o el APIs para iniciar operaciones, como crear instancias o actualizar una función de red a una nueva versión.

AWS TNBadmite las siguientes especificaciones.

Especificación	Release	Descripción
ETSISOL001	v3.6.1	Define los estándares para permitir los descriptores de funciones de red TOSCA basados en datos.
ETSISOL002	v3.6.1	Define modelos en torno a la gestión de funciones de red.
ETSISOL003	v3.6.1	Define los estándares para la gestión del ciclo de vida de las funciones de red.
ETSISOL004	v3.6.1	Define CSAR los estándares para los paquetes de funciones de red.

Especificación	Release	Descripción
ETSISOL005	v3.6.1	Define los estándares para la gestión del paquete de servicios de red y del ciclo de vida de los servicios de red.
ETSISOL007	v3.5.1	Define los estándares para permitir los descriptores de servicios de red TOSCA basados en datos.

Paquetes de funciones de red para AWS TNB

Con él AWS TNB, puede almacenar paquetes de funciones de red que cumplan con los requisitos ETSI SOL 001/ SOL 004 en un catálogo de funciones. Luego, puedes cargar paquetes de Cloud Service Archive (CSAR) que contengan artefactos que describan la función de tu red.

- Descriptor de funciones de red: define los metadatos para la incorporación de paquetes y la administración de las funciones de red
- Imágenes de software: hace referencia a las imágenes del contenedor de funciones de red. Amazon Elastic Container Registry (AmazonECR) puede actuar como repositorio de imágenes de funciones de red.
- Archivos adicionales: utilícelos para administrar la función de red; por ejemplo, guiones y gráficos de Helm.

CSARSe trata de un paquete definido por la OASIS TOSCA norma e incluye un descriptor de red/servicio que cumple con la especificación. OASIS TOSCA YAML Para obtener información sobre la YAML especificación requerida, consulte. TOSCAreferencia para AWS TNB

A continuación se muestra un ejemplo de un descriptor de función de red.

```
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"
```

Descriptores de servicios de red para AWS TNB

AWS TNBalmacena los descriptores de servicios de red (NSDs) sobre las funciones de red que desea implementar y cómo desea implementarlas en el catálogo. Puede cargar el YAML NSD archivo (`vnfd.yaml`), tal como lo describe ETSI SOL 007, para incluir la siguiente información:

- Función de red que desea implementar
- Instrucciones de red
- Instrucciones de cálculo
- Enlaces de ciclo de vida (guiones personalizados)

AWS TNBadmite ETSI los estándares para el modelado de recursos, como la red, el servicio y la función, en el TOSCA lenguaje. AWS TNBhace que su uso sea más eficiente al modelarlos Servicios de AWS de una manera que su orquestador de servicios que ETSI cumpla con las normas pueda entenderlos.

El siguiente es un fragmento de una NSD muestra de cómo modelar. Servicios de AWS La función de red se implementará en un EKS clúster de Amazon con la versión 1.27 de Kubernetes. Las subredes de las aplicaciones son Subnet01 y Subnet02. Luego, puede definir las NodeGroups para sus aplicaciones con una imagen de máquina de Amazon (AMI), un tipo de instancia y una configuración de escalado 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
```

Administración y operaciones para AWS TNB

Con AWS TNB, puede administrar su red mediante operaciones de administración estandarizadas de acuerdo con los ETSI SOL códigos 003 y SOL 005. Puede utilizarla AWS TNB APIs para realizar operaciones del ciclo de vida, tales como:

- Creación de instancias de las funciones de su red.
- Finalización de las funciones de su red.
- Actualizar las funciones de su red para anular implementaciones de Helm.
- Actualizar una instancia de red instanciada o actualizada con un nuevo paquete de red y valores de parámetros.
- Administrar las versiones de sus paquetes de funciones de red.
- Administrar versiones de su. NSDs
- Recuperar información sobre las funciones de red implementadas.

Descriptores de servicios de red para AWS TNB

Un descriptor de servicio de red (NSD) es un `.yaml` archivo de un paquete de red que utiliza el TOSCA estándar para describir las funciones de red que desea implementar y la AWS infraestructura en la que desea implementar las funciones de red. Para definir NSD y configurar los recursos subyacentes y las operaciones del ciclo de vida de la red, debe comprender el NSD TOSCA esquema en el que se basa. AWS TNB

NSDEl archivo se divide en las siguientes partes:

1. TOSCAversión de definición: es la primera línea del NSD YAML archivo y contiene la información de la versión, que se muestra en el siguiente ejemplo.

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

2. VNFD— NSD Contiene la definición de la función de red en la que se van a realizar las operaciones del ciclo de vida. Cada función de la red debe identificarse mediante los siguientes valores:
   - Un identificador único para `descriptor_id`. El ID debe coincidir con el ID del CSAR paquete de funciones de red.
   - Un nombre exclusivo para `namespace`. El nombre debe estar asociado a un identificador único para poder consultarlo más fácilmente en todo el NSD YAML archivo, como se muestra en el siguiente ejemplo.

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

3. Plantilla de topología: define los recursos que se van a implementar, la implementación de la función de red y cualquier guion personalizado, como los enlaces de ciclo de vida. Esto se muestra en el siguiente ejemplo.

```
topology_template:

  node_templates:

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

4. Nodos adicionales: cada recurso modelado tiene secciones para propiedades y requisitos. Las propiedades describen los atributos opcionales u obligatorios de un recurso, como la versión. Los requisitos describen las dependencias que se deben proporcionar como argumentos. Por ejemplo, para crear un recurso de grupo de EKS nodos de Amazon, debe crearse dentro de un Amazon EKS Cluster. Esto se muestra en el siguiente ejemplo.

```
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
```

Configuración AWS TNB

Realice la configuración AWS TNB completando las tareas descritas en este tema.

Tareas

- Inscríbase en una Cuenta de AWS
- Creación de un usuario con acceso administrativo
- Elige una AWS región
- Observe el punto de conexión de servicio
- (Opcional) Instale el AWS CLI
- Configure los roles AWS TNB

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abrir https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Activa la autenticación multifactorial (MFA) para tu usuario root.

   Para obtener instrucciones, consulte Habilitar un MFA dispositivo virtual para el usuario Cuenta de AWS root (consola) en la Guía del IAM usuario.

Creación de un usuario con acceso administrativo

1. Habilite IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre cómo usar el Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con su usuario de IAM Identity Center, utilice el inicio de sesión URL que se envió a su dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario de IAM Identity Center, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos con privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Elige una AWS región

Para ver la lista de regiones disponibles AWS TNB, consulte la Lista de servicios AWS regionales. Para ver la lista de puntos finales para el acceso programático, consulte los AWS TNBpuntos finales en el. Referencia general de AWS

Observe el punto de conexión de servicio

Para conectarse mediante programación a un AWS servicio, se utiliza un punto final. Además de los puntos de conexión estándar AWS , algunos AWS servicios ofrecen puntos de conexión en FIPS determinadas regiones. Para obtener más información, consulte puntos de conexión de servicio de AWS.

Nombre de la región	Región	Punto de conexión	Protocolo	
Este de EE. UU. (Norte de Virginia)	us-east-1	tnb.us-east-1.amazonaws.com	HTTPS	

Nombre de la región	Región	Punto de conexión	Protocolo	
Oeste de EE. UU. (Oregón)	us-west-2	tnb.us-west-2.amazonaws.com	HTTPS	
Asia-Pacífico (Seúl)	ap-northeast-2	tnb.ap-northeast-2.amazonaws.com	HTTPS	
Asia-Pacífico (Sídney)	ap-southeast-2	tnb.ap-southeast-2.amazonaws.com	HTTPS	
Canadá (centro)	ca-central-1	tnb.ca-central-1.amazonaws.com	HTTPS	
Europa (Fráncfort)	eu-central-1	tnb.eu-central-1.amazonaws.com	HTTPS	
Europa (París)	eu-west-3	tnb.eu-west-3.amazonaws.com	HTTPS	
Europa (España)	eu-south-2	tnb.eu-south-2.amazonaws.com	HTTPS	
Europa (Estocolmo)	eu-north-1	tnb.eu-north-1.amazonaws.com	HTTPS	
América del Sur (São Paulo)	sa-east-1	tnb.sa-east-1.amazonaws.com	HTTPS	

(Opcional) Instale el AWS CLI

El AWS Command Line Interface (AWS CLI) proporciona comandos para un amplio conjunto de AWS productos y es compatible con Windows, macOS y Linux. Puede acceder AWS TNB mediante AWS CLI. Para empezar, consulte la AWS Command Line Interface Guía del usuario de . Para obtener más información sobre los comandos de AWS TNB, consulte tnb en la Referencia de AWS CLI comandos.

Configure los roles AWS TNB

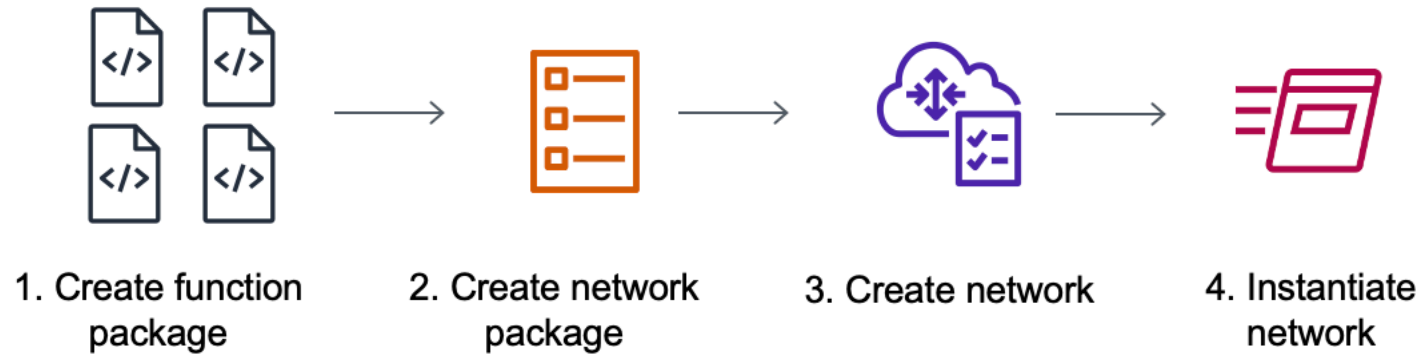
Debe crear un rol IAM de servicio para administrar las distintas partes de la AWS TNB solución. AWS TNBLos roles de servicio pueden realizar API llamadas a otros AWS servicios, como AWS CloudFormation AWS CodeBuild, y a varios servicios de cómputo y almacenamiento, en su nombre, para crear instancias y administrar los recursos para su implementación.

Para obtener más información sobre la función de AWS TNB servicio, consulte. Administración de identidad y acceso para AWS TNB

Empezar con AWS TNB

En este tutorial se muestra cómo AWS TNB implementar una función de red, por ejemplo, la unidad centralizada (CU), la función de administración de acceso y movilidad (AMF) o la función de plano de usuario de 5G (UPF).

En el diagrama siguiente se ilustra el proceso de implementación:



Tareas

- Requisitos previos
- Cree un paquete de funciones
- Crear un paquete de red
- Crear e instanciar una instancia de red
- Limpieza

Requisitos previos

Para poder realizar una implementación correcta, debe disponer de lo siguiente:

- Un plan AWS Business Support.
- Permisos mediante IAM funciones.
- Un paquete de funciones de red (NF) que cumple con la norma ETSI SOL 001/ 004SOL.
- Plantillas de descriptores de servicios de red (NSD) que cumplen con la norma 007. ETSI SOL

Puede utilizar un paquete de funciones o un paquete de red de ejemplo del AWS TNB GitHub sitio Paquetes de muestra.

Cree un paquete de funciones

Un paquete de funciones de red es un archivo Cloud Service Archive (CSAR). El CSAR archivo contiene gráficos de Helm, imágenes de software y un descriptor de funciones de red (NFD).

Para crear un paquete de funciones

1. Abra la AWS TNB consola en. https://console.aws.amazon.com/tnb/
2. Seleccione Paquetes de funciones en el panel de navegación.
3. Elija Crear paquete de funciones.
4. En el paquete de funciones de carga, selecciona Elegir archivos y carga cada CSAR paquete como un `.zip` archivo. Puede cargar un máximo de 10 archivos.
5. (Opcional) En Etiquetas, selecciona Añadir nueva etiqueta e introduce una clave y un valor. Puede usar etiquetas para buscar y filtrar sus recursos o realizar un seguimiento de sus AWS costos.
6. Elija Next (Siguiente).
7. Revise los detalles del paquete y, a continuación, seleccione Crear paquete de funciones.

Crear un paquete de red

Un paquete de red especifica las funciones de red que desea implementar y cómo desea implementarlas en el catálogo.

Para crear un paquete de red

1. Seleccione Paquetes de red en el panel de navegación.
2. Seleccione Crear paquete de red.
3. En Cargar paquete de red, elija Elegir archivos y cargue cada uno NSD como un `.zip` archivo. Puedes cargar un máximo de 10 archivos.
4. (Opcional) En Etiquetas, selecciona Añadir nueva etiqueta e introduce una clave y un valor. Puede usar etiquetas para buscar y filtrar sus recursos o realizar un seguimiento de sus AWS costos.

5. Elija Next (Siguiente).
6. Seleccione Crear paquete de red.

Crear e instanciar una instancia de red

Una instancia de red es una red única creada en la AWS TNB que se puede implementar. Debe crear una instancia de red e instanciarla. Al crear una instancia de red, se AWS TNB aprovisiona la AWS infraestructura necesaria, se despliegan funciones de red en contenedores y se configura la administración de redes y accesos para crear un servicio de red totalmente operativo.

Para crear e instanciar una instancia de red

1. En el panel de navegación, elija Redes.
2. Elija Crear instancia de red.
3. Introduzca un nombre y una descripción para la red y, a continuación, elija Siguiente.
4. Elija un paquete de red. Compruebe los detalles y seleccione Siguiente.
5. Elija Crear instancia de red. El estado inicial es `Created`.

   Aparece la página Redes que muestra la nueva instancia de red en el `Not instantiated` estado.
6. Seleccione la instancia de red, elija Acciones e Instanciar.

   Aparece la página de creación de instancias de red.
7. Revise los detalles y actualice los valores de los parámetros. Las actualizaciones de los valores de los parámetros solo se aplican a esta instancia de red. Los parámetros de los VNFD paquetes NSD y no cambian.
8. Elija Instanciar red.

   Aparece la página de estado del despliegue.
9. Utilice el icono de actualización para realizar un seguimiento del estado de despliegue de la instancia de red. También puede activar la actualización automática en la sección Tareas de despliegue para realizar un seguimiento del progreso de cada tarea.

Limpieza

Ahora puede eliminar los recursos que creó para este tutorial.

Para limpiar los recursos

1. En el panel de navegación, elija Redes.
2. Elija el identificador de la red y, a continuación, elija Finalizar.
3. Cuando se le solicite confirmación, introduzca el identificador de red y, a continuación, elija Finalizar.
4. Utilice el icono de actualización para realizar un seguimiento del estado de la instancia de red.
5. (Opcional) Seleccione la red y elija Eliminar.

Paquetes de funciones para AWS TNB

Un paquete de funciones es un archivo.zip en formato CSAR (Cloud Service Archive) que contiene una función de red (una aplicación de telecomunicaciones ETSI estándar) y un descriptor de paquete de funciones que utiliza el TOSCA estándar para describir cómo deben ejecutarse las funciones de la red en la red.

Tareas

- Cree un paquete de funciones en AWS TNB
- Vea un paquete de funciones en AWS TNB
- Descargue un paquete de funciones de AWS TNB
- Elimine un paquete de funciones de AWS TNB

Cree un paquete de funciones en AWS TNB

Aprenda a crear un paquete de funciones en el catálogo AWS TNB de funciones de red. Crear un paquete de funciones es el primer paso para crear una red en AWS TNB. Después de cargar un paquete de funciones, puede crear un paquete de red.

Console

Cree un paquete de funciones mediante la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de funciones en el panel de navegación.
3. Elija Crear paquete de funciones.
4. Selecciona Elegir archivos y carga cada CSAR paquete como un `.zip` archivo. Puedes subir un máximo de 10 archivos.
5. Elija Next (Siguiente).
6. Revise los detalles del paquete.
7. Elija Crear paquete de funciones.

AWS CLI

Para crear un paquete de funciones mediante el AWS CLI

1. Utilice el create-sol-function-packagecomando para crear un nuevo paquete de funciones:

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

2. Utilice el comando put-sol-function-package-content para cargar el contenido del paquete de funciones. Por ejemplo:

```
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
```

Vea un paquete de funciones en AWS TNB

Aprenda a ver el contenido de un paquete de funciones.

Console

Para ver un paquete de funciones mediante la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de funciones en el panel de navegación.
3. Utilice el cuadro de búsqueda para encontrar el paquete de funciones

AWS CLI

Para ver un paquete de funciones mediante el AWS CLI

1. Utilice el list-sol-function-packagescomando para enumerar los paquetes de funciones.

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

2. Utilice el get-sol-function-packagecomando para ver los detalles de un paquete de funciones.

```
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
```

Descargue un paquete de funciones de AWS TNB

Aprenda a descargar un paquete de funciones del catálogo de funciones de AWS TNB red.

Console

Para descargar un paquete de funciones mediante la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. En el panel de navegación del lado izquierdo de la consola, elija Paquetes de funciones.
3. Utilice el cuadro de búsqueda para encontrar el paquete de funciones
4. Elija el paquete de funciones
5. En Acciones, elija Descargar.

AWS CLI

Para descargar un paquete de funciones mediante el AWS CLI

Utilice el comando get-sol-function-package-content para descargar un paquete de funciones.

```
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
```

Elimine un paquete de funciones de AWS TNB

Aprenda a eliminar un paquete de funciones del catálogo de funciones de AWS TNB red. Para eliminar un paquete de funciones, el paquete debe estar desactivado.

Console

Para eliminar un paquete de funciones mediante la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de funciones en el panel de navegación.
3. Utilice el cuadro de búsqueda para encontrar el paquete de funciones.
4. Elija un paquete de funciones.
5. Elija Acciones, Desactivar.
6. Elija Acciones, Eliminar.

AWS CLI

Para eliminar un paquete de funciones mediante el AWS CLI

1. Utilice el update-sol-function-packagecomando para deshabilitar un paquete de funciones.

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

2. Utilice el delete-sol-function-packagecomando para eliminar un paquete de funciones.

```
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
```

Paquetes de red para AWS TNB

Un paquete de red es un archivo.zip en formato CSAR (Cloud Service Archive) que define los paquetes de funciones que desea implementar y la AWS infraestructura en la que desea implementarlos.

Tareas

- Cree un paquete de red en AWS TNB
- Vea un paquete de red en AWS TNB
- Descargue un paquete de red de AWS TNB
- Elimine un paquete de red de AWS TNB

Cree un paquete de red en AWS TNB

Un paquete de red consta de un archivo descriptor del servicio de red (NSD) (obligatorio) y cualquier archivo adicional (opcional), como los scripts específicos que se adapten a sus necesidades. Por ejemplo, si tiene varios paquetes de funciones en su paquete de red, puede usar el NSD para definir qué funciones de red deben ejecutarse en determinadas VPCs subredes o EKS clústeres de Amazon.

Cree un paquete de red después de crear los paquetes de funciones. Una vez que haya creado un paquete de red, debe crear una instancia de red.

Console

Para crear un paquete de red con la consola

1. Abra la AWS TNB consola en. https://console.aws.amazon.com/tnb/
2. Seleccione Paquetes de red en el panel de navegación.
3. Seleccione Crear paquete de red.
4. Selecciona Elegir archivos y carga cada uno NSD como un `.zip` archivo. Puedes subir un máximo de 10 archivos.
5. Elija Next (Siguiente).
6. Revise los detalles del paquete.
7. Seleccione Crear paquete de red.

AWS CLI

Para crear un paquete de red mediante AWS CLI

1. Utilice el create-sol-network-packagecomando para crear un paquete de red.

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

2. Utilice el comando put-sol-network-package-content para cargar el contenido del paquete de red. Por ejemplo:

```
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
```

Vea un paquete de red en AWS TNB

Aprenda a ver el contenido de un paquete de red.

Console

Para ver un paquete de red con la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de red en el panel de navegación.
3. Utilice el cuadro de búsqueda para encontrar el paquete de red.

AWS CLI

Para ver un paquete de red mediante AWS CLI

1. Utilice el list-sol-network-packagescomando para enumerar los paquetes de red.

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

2. Utilice el get-sol-network-packagecomando para ver los detalles de un paquete de red.

```
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
```

Descargue un paquete de red de AWS TNB

Aprenda a descargar un paquete de red del catálogo de servicios de AWS TNB red.

Console

Para descargar un paquete de red con la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de red en el panel de navegación.
3. Utilice el cuadro de búsqueda para encontrar el paquete de red
4. Seleccione el paquete de red.
5. En Acciones, elija Descargar.

AWS CLI

Para descargar un paquete de red mediante AWS CLI

- Utilice el comando get-sol-network-package-content para descargar un paquete de red.

```
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
```

Elimine un paquete de red de AWS TNB

Aprenda a eliminar un paquete de red del catálogo de servicios de AWS TNB red. Para eliminar un paquete de red, el paquete debe estar desactivado.

Console

Para eliminar un paquete de red con la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de red en el panel de navegación.
3. Utilice el cuadro de búsqueda para encontrar el paquete de red
4. Seleccione el paquete de red
5. Elija Acciones, Desactivar.
6. Elija Acciones, Eliminar.

AWS CLI

Para eliminar un paquete de red mediante el AWS CLI

1. Utilice el update-sol-network-packagecomando para deshabilitar un paquete de red.

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

2. Utilice el delete-sol-network-packagecomando para eliminar un paquete de red.

```
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
```

Instancias de red para AWS TNB

Una instancia de red es una red única creada en la AWS TNB que se puede implementar.

Tareas

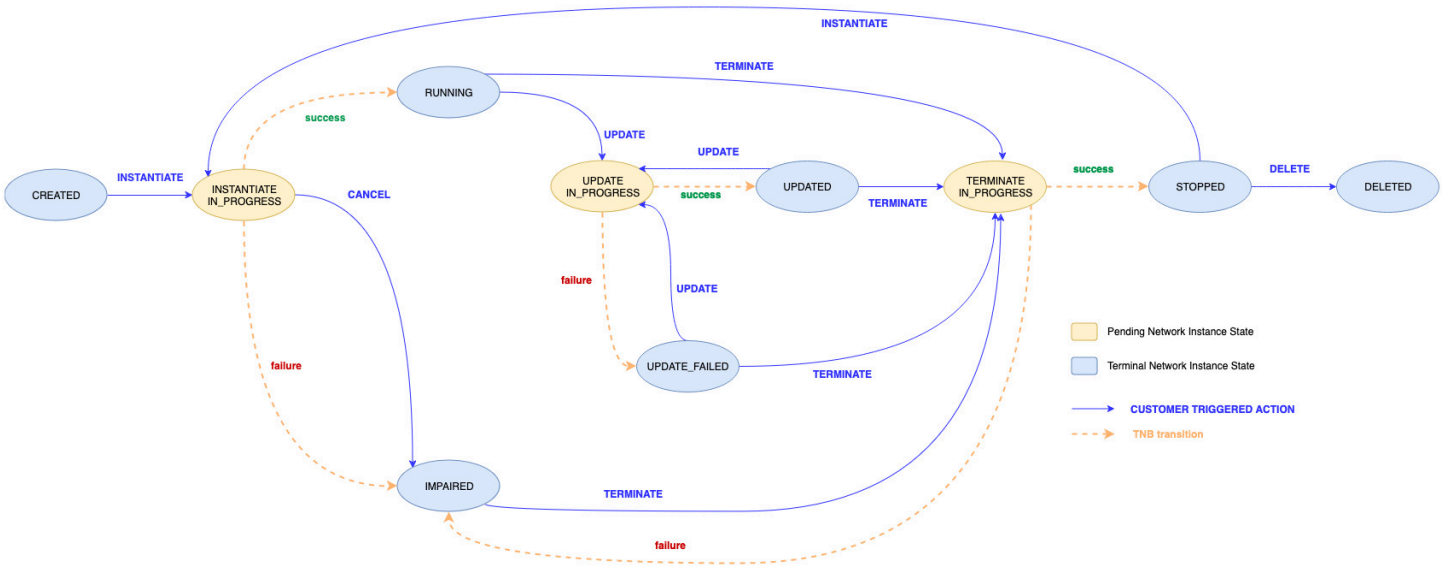
- Operaciones del ciclo de vida de una instancia de red
- Cree una instancia de red mediante AWS TNB
- Cree una instancia de red mediante AWS TNB
- Actualice una instancia de función en AWS TNB
- Actualice una instancia de red en AWS TNB
- Vea una instancia de red en AWS TNB
- Finalice y elimine una instancia de red de AWS TNB

Operaciones del ciclo de vida de una instancia de red

AWS TNBle permite administrar fácilmente su red mediante operaciones de administración estandarizadas en línea con ETSI SOL 003 y SOL 005. Puede realizar las siguientes operaciones del ciclo de vida:

- Cree la red
- Cree una instancia de la red
- Actualice la función de red
- Actualice la instancia de red
- Vea los detalles y el estado de la red
- Termine la red

La siguiente imagen muestra las operaciones de administración de la red:



Cree una instancia de red mediante AWS TNB

Una instancia de red se crea después de crear un paquete de red. Después de crear una instancia de red, instanciela.

Console

Para crear una instancia de red mediante la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. En el panel de navegación, elija Redes.
3. Elija Crear instancia de red.
4. Introduzca un nombre y una descripción para la instancia y, a continuación, elija Siguiente.
5. Seleccione el paquete de red, compruebe los detalles y pulse Siguiente.
6. Elija Crear instancia de red.

   La nueva instancia de red aparece en la página Redes. A continuación, cree una instancia de esta instancia de red.

AWS CLI

Para crear una instancia de red mediante AWS CLI

- Utilice el create-sol-network-instancecomando para crear una instancia de red.

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

A continuación, cree una instancia de esta instancia de red.

Cree una instancia de red mediante AWS TNB

Después de crear una instancia de red, debe instanciarla. Al crear una instancia de red, se AWS TNB aprovisiona la AWS infraestructura necesaria, se despliegan funciones de red en contenedores y se configura la administración de redes y accesos para crear un servicio de red totalmente operativo.

Console

Para crear una instancia de red mediante la consola

1. Abra la AWS TNB consola en. https://console.aws.amazon.com/tnb/
2. En el panel de navegación, elija Redes.
3. Seleccione la instancia de red de la que desee crear una instancia.
4. Elija Acciones y, a continuación, Crear una instancia.
5. En la página Instanciar la red, revise los detalles y, si lo desea, actualice los valores de los parámetros.

   Las actualizaciones de los valores de los parámetros solo se aplican a esta instancia de red. Los parámetros de los VNFD paquetes NSD y no cambian.
6. Elija Instanciar red.

   Aparece la página de estado del despliegue.
7. Utilice el icono de actualización para realizar un seguimiento del estado de despliegue de la instancia de red. También puede activar la actualización automática en la sección Tareas de despliegue para realizar un seguimiento del progreso de cada tarea.

   Cuando el estado de la implementación cambia a`Completed`, se crea una instancia de red.

AWS CLI

Para crear una instancia de red mediante el AWS CLI

1. Usa el instantiate-sol-network-instancecomando para crear una instancia de red.

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

2. A continuación, consulte el estado de funcionamiento de la red.

Actualice una instancia de función en AWS TNB

Después de crear una instancia de red, puede actualizar un paquete de funciones en la instancia de red.

Console

Para actualizar una instancia de función mediante la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. En el panel de navegación, elija Redes.
3. Seleccione la instancia de red. Puede actualizar una instancia de red solo si su estado es`Instantiated`.

   Aparece la página de la instancia de red.
4. En la pestaña Funciones, seleccione la instancia de la función que desee actualizar.
5. Elija Actualizar.
6. Introduzca las anulaciones de actualización.
7. Elija Actualizar.

AWS CLI

Utilícela CLI para actualizar una instancia de función

Utilice el update-sol-network-instancecomando con el tipo de `MODIFY_VNF_INFORMATION` actualización para actualizar una instancia de función en una instancia de red.

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

Actualice una instancia de red en AWS TNB

Después de crear una instancia de red, es posible que tengas que actualizar la infraestructura o la aplicación. Para ello, debe actualizar el paquete de red y los valores de los parámetros de la instancia de red e implementar la operación de actualización para aplicar los cambios.

Consideraciones

- Puede actualizar una instancia de red que esté en el `Updated` estado `Instantiated` o.
- Al actualizar una instancia de red, `UpdateSolNetworkService` API utiliza el nuevo paquete de red y los valores de los parámetros para actualizar la topología de la instancia de red.
- AWS TNBverifica que el número NSD y los VNFD parámetros de la instancia de red no superen los 200. Este límite se aplica para evitar que personas malintencionadas transmitan cargas erróneas o enormes que afecten al servicio.

Parámetros que puede actualizar

Puede actualizar los siguientes parámetros al actualizar una instancia de red instanciada:

Parámetro	Descripción	Ejemplo: antes	Ejemplo: Después

Versión de Amazon EKS Cluster	Puedes actualizar el valor del `version` parámetro del plano de control del EKS clúster de Amazon a la siguiente versión secundaria. No puedes cambiar la versión a una versión inferior. Los nodos de trabajo no se actualizan.	`EKSCluster:`
		`  type: tosca.nodes.AWS.Compute.EKS`
		`  properties:`
		`  version: "1.28"`
			`EKSC`
			`r:`
			
			`typ`
			`tos`
			`es.A`
			`mput`

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu
			pro s: ver "1.

Parámetro	Descripción	Ejemplo: antes	Ejemplo: Despu

Propiedades de escalado	Puede actualizar las propiedades de escalado de los `EKSSelfManagedNode` TOSCA nodos `EKSManagedNode` y.	`EKSNodeGroup01:`
		`  ...`
		`  scaling:`
		`  properties:`
		`  desired_size: 1`
		`  min_size: 1`
		`  max_size: 1`
			`EKSNoup0`
			`...`
			`sca`
			`pros:`
			`desize:`

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu
			mir max

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu

Propiedades del EBS CSI plugin de Amazon	Puedes activar o desactivar el EBS CSI plugin de Amazon en tus EKS clústeres de Amazon. También puedes cambiar la versión del plugin.	`EKSCluster:`
		`  capabilities:`
		`  ...`
		`  ebs_csi:`
		`  properties:`
		`  enabled: false`
			EKSC
			r:
			cap
			ies:
			...
			ebs
			pro
			s:
			ena
			ver
			"v1
			e
			ksbu
			"

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu
VNF	Puedes hacer referencia a ellos NSD e implementarlos VNFs en el clúster creado al NSD usar el VNFDeployment TOSCA nodo. Como parte de la actualización, podrás añadirlos, actualizarlos y VNFs eliminarlos en la red.	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

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu
			elmD : typ tos es.A ploy VNFD ment rec nts: clu EKS r vnf

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu

			`- v`
			`leVN`
			
			`- v`
			`leVN`

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu

Enlaces	Para ejecutar las operaciones del ciclo de vida antes y después de crear una función de red, añada los `post_create` ganchos `pre_create` y al `VNFDeployment` nodo.
	
	En este ejemplo, el `PreCreateHook` gancho se ejecutará antes de crear `vnf3.SampleVNF3` una instancia y se ejecutará después de crearla`vnf3.SampleVNF3` . `PostCreateHook`
		`vnfds:`
		`  - descriptor_id: "43c012fa-2616-41a8-a833-0dfd4c5a049e "`
		`    namespace: " vnf1"`
		`  - descriptor_id: "64222f98-ecd6-4871-bf94-7354b53f3ee5 "`
		`    namespace: " vnf2"`
		`...`
		`SampleVNF1HelmDeploy:`
		`  type: tosca.nodes.AWS.Deployment.VNFDeployment`
		`  requirements:`
		`    cluster: EKSCluster`
		`    vnfs:`
		`      - vnf1.SampleVNF1`
		`      - vnf2.SampleVNF2  // Removed during update`
			`vnfd`
			` -`
			` des`
			`r_id`
			` "43`
			`2616`
			`-`
			`a833`
			`d4c5`
			` "`
			` nam`
			`:`
			` "vn`
			` -`
			` des`
			`r_id`
			` "b7`
			`839d`
			`-916`
			`a166`
			` "`
			` nam`
			`:`
			` "vn`
			`....`
			`S`
			`ampl`
			`Helm`
			`y:`
			` typ`
			` tos`

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu
			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

Parámetro	Descripción	Ejemplo: antes	Ejemplo: Después
			- v leVN New VNF as the nam , vnf was not pre y pre int s: Hoo pos te: eHoo pre e: Hook

Parámetro	Descripción	Ejemplo: antes	Ejemplo: Después

Enlaces	Para ejecutar las operaciones del ciclo de vida antes y después de actualizar una función de red, puede añadir el `pre_update` gancho y el `post_update` gancho al nodo. `VNFDeployment`
	
	En este ejemplo, `PreUpdate Hook` se ejecutará antes de la actualización y `vnf1.SampleVNF1` se `PostUpdateHook` ejecutará después de que `vnf1.SampleVNF1` se actualice en el `vnf` paquete indicado por la actualización `uuid` para el espacio de nombres vnf1.
		`vnfds:`
		`  - descriptor_id: "43c012fa-2616-41a8-a833-0dfd4c5a049e "`
		`    namespace: " vnf1"`
		`  - descriptor_id: "64222f98-ecd6-4871-bf94-7354b53f3ee5 "`
		`    namespace: " vnf2"`
		`...`
		
		`SampleVNF1HelmDeploy:`
		`  type: tosca.nodes.AWS.Deployment.VNFDeployment`
		`  requirements:`
		`    cluster: EKSCluster`
		`    vnfs:`
		`      - vnf1.SampleVNF1`
		`      - vnf2.SampleVNF2`
			`vnfd`
			
			`  -`
			`  des`
			`r_id`
			`  "0e`
			`bd87`
			`  -`
			`b8a1`
			`4666`
			`  "`
			
			`  nam`
			`:`
			`  "vn`
			
			`  -`
			`  des`
			`r_id`
			`  "64`
			`ecd6`
			`  -`
			`bf94`
			`4b53`
			`  "`
			
			`  nam`
			`:`
			`  "vn`
			`...`
			`S`
			`ampl`
			`Helm`
			`y:`
			
			`  typ`

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu
			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

Parámetro	Descripción	Ejemplo: antes	Ejemp Despu
			leVN No cha to thi fur as nam and uui rem the sam int s: Hoo pre e: Hook pos te: eHoo

Actualización de una instancia de red

Console

Para actualizar una instancia de red mediante la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. En el panel de navegación, elija Redes.
3. Seleccione la instancia de red. Puede actualizar una instancia de red solo si su estado es `Instantiated` o`Updated`.
4. Seleccione Acciones y Actualizar.

   Aparece la página de actualización de la instancia con los detalles de la red y una lista de parámetros de la infraestructura actual.
5. Elige un paquete de red nuevo.

   Los parámetros del nuevo paquete de red aparecen en la sección Parámetros actualizados.
6. Si lo desea, actualice los valores de los parámetros en la sección Parámetros actualizados. Para ver la lista de valores de parámetros que puede actualizar, consulteParámetros que puede actualizar.
7. Seleccione Actualizar red.

   AWS TNBvalida la solicitud e inicia el despliegue. Aparece la página de estado del despliegue.
8. Utilice el icono de actualización para realizar un seguimiento del estado de despliegue de la instancia de red. También puede activar la actualización automática en la sección Tareas de despliegue para realizar un seguimiento del progreso de cada tarea.

   Cuando el estado de la implementación cambia a`Completed`, la instancia de red se actualiza.
9. - Si se produce un error en la validación, la instancia de red permanece en el mismo estado en el que estaba antes de solicitar la actualización, ya sea `Instantiated` o no`Updated`.
   - Si se produce un error en la actualización, se muestra el estado de la instancia de red`Update failed`. Elija el enlace de cada tarea fallida para determinar el motivo.
   - Si la actualización se realiza correctamente, se muestra `Updated` el estado de la instancia de red.

AWS CLI

Use el CLI para actualizar una instancia de red

Utilice el update-sol-network-instancecomando con el tipo de UPDATE_NS actualización para actualizar una instancia de red.

```
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\"}}"
```

Vea una instancia de red en AWS TNB

Obtenga información sobre cómo ver una instancia de red.

Console

Para ver una instancia de red con la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. Seleccione Instancias de red en el panel de navegación.
3. Utilice el cuadro de búsqueda para encontrar la instancia de red.

AWS CLI

Para ver una instancia de red mediante AWS CLI

1. Utilice el list-sol-network-instancescomando para enumerar las instancias de red.

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

2. Use el get-sol-network-instancecomando para ver los detalles de una instancia de red específica.

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

Finalice y elimine una instancia de red de AWS TNB

Para eliminar una instancia de red, la instancia debe tener estado finalizado.

Console

Para finalizar y eliminar una instancia de red con la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. En el panel de navegación, elija Redes.
3. Seleccione el identificador de la instancia de red.
4. Elija Finalizar.
5. Cuando se le solicite confirmación, introduzca el identificador y elija Finalizar.
6. Actualice para realizar un seguimiento del estado de la instancia de red.
7. (Opcional) Seleccione la instancia de red y elija Eliminar.

AWS CLI

Para finalizar y eliminar una instancia de red mediante el AWS CLI

1. Utilice el terminate-sol-network-instancecomando para terminar una instancia de red.

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

2. (Opcional) Utilice el delete-sol-network-instancecomando para eliminar una instancia de red.

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

Operaciones de red para AWS TNB

Una operación de red es cualquier operación que se realiza en la red, como la instanciación o la finalización de una instancia de red.

Tareas

- Ver una operación AWS TNB de red
- Cancela una operación AWS TNB de red

Ver una operación AWS TNB de red

Vea los detalles de una operación de red, incluidas las tareas implicadas en la operación de la red y el estado de las tareas.

Console

Para ver una operación de red con la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. Seleccione Instancias de red en el panel de navegación.
3. Utilice el cuadro de búsqueda para encontrar la instancia de red.
4. En la pestaña Implementaciones, seleccione la operación de red.

AWS CLI

Para ver una operación de red mediante el AWS CLI

1. Utilice el list-sol-network-operationscomando para enumerar todas las operaciones de red.

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

2. Utilice el get-sol-network-operationcomando para ver los detalles de una operación de red.

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

Cancela una operación AWS TNB de red

Obtenga información acerca de cómo cancelar una operación de red.

Console

Para cancelar una operación de red con la consola

1. Abra la AWS TNB consola en https://console.aws.amazon.com/tnb/.
2. En el panel de navegación, elija Redes.
3. Seleccione el identificador de la red para abrir su página de detalles.
4. En la pestaña Implementaciones, elija la operación de red.
5. Seleccione Cancelar operación.

AWS CLI

Para cancelar una operación de red mediante el AWS CLI

Utilice el cancel-sol-network-operationcomando para cancelar una operación de red.

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

TOSCAreferencia para AWS TNB

La especificación de topología y organización para aplicaciones en la nube (TOSCA) es una sintaxis declarativa que se CSPs utiliza para describir una topología de servicios web basados en la nube, sus componentes, relaciones y los procesos que los gestionan. CSPsdescriba los puntos de conexión, los enlaces lógicos entre los puntos de conexión y las políticas, como la afinidad y la seguridad, en una plantilla. TOSCA CSPsa continuación, cargue la plantilla en la AWS TNB que se sintetizan los recursos necesarios para establecer una red 5G que funcione en todas las zonas de AWS disponibilidad.

Contenido

VNFDplantilla

Define una plantilla de descriptor de función de red virtual (VNFD).

Sintaxis

```
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
```

Plantilla de topología

`node_templates`

Los TOSCA AWS nodos. Los posibles nodos son:

- AWS.VNF
- AWS.Artifacts.Helm

AWS.VNF

Define un nodo de función de red AWS virtual (VNF).

Sintaxis

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

Propiedades

`descriptor_id`

El UUID del descriptor.

Obligatorio: sí

Tipo: cadena

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

`descriptor_version`

La versión delVNFD.

Obligatorio: sí

Tipo: cadena

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

`descriptor_name`

El nombre del descriptor.

Obligatorio: sí

Tipo: cadena

`provider`

El autor delVNFD.

Obligatorio: sí

Tipo: cadena

Requisitos

`helm`

El directorio Helm que define los artefactos del contenedor. Esta es una referencia a AWS.Artifacts.Helm.

Obligatorio: sí

Tipo: cadena

Ejemplo

```
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 un nodo AWS Helm.

Sintaxis

```
tosca.nodes.AWS.Artifacts.Helm:
  properties:
    implementation: String
```

Propiedades

`implementation`

El directorio local que contiene el gráfico de Helm dentro del CSAR paquete.

Obligatorio: sí

Tipo: cadena

Ejemplo

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

Plantilla de descriptor de servicio de red

Define una plantilla de descriptor de servicio de red (NSD).

Sintaxis

```
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 parámetros definidos

Si desea transferir un parámetro de forma dinámica, como el CIDR bloque del VPC nodo, puede utilizar la `{ get_input: input-parameter-name }` sintaxis y definir los parámetros de la NSD plantilla. A continuación, reutilice el parámetro en la misma NSD plantilla.

En el siguiente ejemplo se muestra cómo definir y utilizar parámetros:

```
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`

El UUID del descriptor.

Obligatorio: sí

Tipo: cadena

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

`namespace`

El nombre único.

Obligatorio: sí

Tipo: cadena

Plantilla de topología

`node_templates`

Los TOSCA AWS nodos posibles son:

- AWS.NS
- AWS.Compute. EKS
- AWS.Computar. EKS. AuthRole
- AWS.Computar. EKSManagedNode
- AWS.Computar. EKSSelfManagedNode
- AWS.Computar. PlacementGroup
- AWS.Computar. UserData
- AWS.Redes. SecurityGroup
- AWS.Redes. SecurityGroupEgressRule

- AWS.Redes. SecurityGroupIngressRule
- AWS.Resource.Import
- AWS.Redes. ENI
- AWS.HookExecution
- AWS.Redes. InternetGateway
- AWS.Redes. RouteTable
- AWS.Networking.Subnet
- AWS.Despliegue. VNFDeployment
- AWS.Redes. VPC
- AWS.Redes. NATGateway
- AWS.Networking.Route

AWS.NS

Define un nodo de servicio de AWS red (NS).

Sintaxis

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

Propiedades

`descriptor_id`

El UUID del descriptor.

Obligatorio: sí

Tipo: cadena

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

`descriptor_version`

La versión delNSD.

Obligatorio: sí

Tipo: cadena

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

`descriptor_name`

El nombre del descriptor.

Obligatorio: sí

Tipo: cadena

Ejemplo

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

AWS.Compute. EKS

Proporcione el nombre del clúster, la versión de Kubernetes que desee y una función que permita al plano de control de Kubernetes gestionar los recursos que necesita. AWS NFs Los complementos de la interfaz de red de contenedores Multus () están habilitados. CNI Puede conectar varias interfaces de red y aplicar una configuración de red avanzada a las funciones de red basadas en Kubernetes. También debe especificar el acceso al punto de conexión del clúster y las subredes del clúster.

Sintaxis

```
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. Propiedades que definen el uso de la interfaz de red de contenedores Multus (CNI).

Si incluye `multus`, especifique las propiedades `enabled` y `multus_role`.

`enabled`

Indica si la capacidad de Multus predeterminada está habilitada.

Obligatorio: sí

Tipo: Booleano

`multus_role`

La función de administración de la interfaz de red de Multus.

Obligatorio: sí

Tipo: cadena

`ebs_csi`

Propiedades que definen el controlador Amazon EBS Container Storage Interface (CSI) instalado en el EKS clúster de Amazon.

Habilita este complemento para usar nodos EKS autogestionados de Amazon en AWS Outposts, Zonas AWS Locales o Regiones de AWS. Para obtener más información, consulte el CSIcontrolador Amazon Elastic Block Store en la Guía del EKS usuario de Amazon.

`enabled`

Indica si el EBS CSI controlador de Amazon predeterminado está instalado.

Obligatorio: no

Tipo: booleano

`version`

La versión del complemento de EBS CSI controladores de Amazon. La versión debe coincidir con una de las versiones devueltas por la DescribeAddonVersionsacción. Para obtener más información, consulta DescribeAddonVersionsla EKSAPIreferencia de Amazon

Obligatorio: no

Tipo: cadena

Propiedades

`version`

La versión de Kubernetes para el clúster. AWS Telco Network Builder es compatible con las versiones 1.23 a 1.30 de Kubernetes.

Obligatorio: sí

Tipo: cadena

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

`access`

El acceso al punto de conexión del clúster.

Obligatorio: sí

Tipo: cadena

Valores posibles: `PRIVATE` | `PUBLIC` | `ALL`

`cluster_role`

El rol de administración de clústeres.

Obligatorio: sí

Tipo: cadena

`tags`

Etiquetas que deben asociarse a este recurso.

Obligatorio: no

Tipo: lista

`ip_family`

Indica la familia de IP de las direcciones de servicio y pod del clúster.

Valor permitido: `IPv4`, `IPv6`

Valor predeterminado: `IPv4`

Obligatorio: no

Tipo: cadena

Requisitos

`subnets`

Un nodo AWS.Networking.Subnet.

Obligatorio: sí

Tipo: lista

Ejemplo

```
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.Calcular. EKS. AuthRole

A AuthRole le permite añadir IAM roles al EKS clúster `aws-auth` `ConfigMap` de Amazon para que los usuarios puedan acceder al EKS clúster de Amazon mediante un IAM rol.

Sintaxis

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

Propiedades

`role_mappings`

Lista de mapeos que definen las IAM funciones que deben añadirse al clúster de AmazonEKS. `aws-auth` `ConfigMap`

`arn`

El ARN del rol. IAM

Obligatorio: sí

Tipo: cadena

`groups`

Grupos de Kubernetes para asignarlos a la función definida en `arn`.

Obligatorio: no

Tipo: lista

Requisitos

`clusters`

Un AWS.Compute. EKSnodo.

Obligatorio: sí

Tipo: lista

Ejemplo

```
EKSAuthMapRoles:
    type: tosca.nodes.AWS.Compute.EKS.AuthRole
    properties:
        role_mappings:
        - arn: arn:aws:iam::${AWS::TNB::AccountId}:role/TNBHookRole1
          groups:
          - system:nodes
          - system:bootstrappers
        - arn: arn:aws:iam::${AWS::TNB::AccountId}:role/TNBHookRole2
          groups:
          - system:nodes
          - system:bootstrappers
    requirements:
         clusters:
         - Free5GCEKS1
```

```
        - Free5GCEKS2
```

AWS.Compute. EKSManagedNode

AWS TNBadmite grupos de nodos EKS gestionados para automatizar el aprovisionamiento y la gestión del ciclo de vida de los nodos (EC2instancias de Amazon) para los clústeres de Amazon EKS Kubernetes. Para crear un grupo de EKS nodos, haga lo siguiente:

- Elija Amazon Machine Images (AMI) para los nodos de trabajadores de su clúster proporcionando el ID del AMI o el AMI tipo.
- Proporcione un EC2 key pair de Amazon para el SSH acceso y las propiedades de escalado de su grupo de nodos.
- Asegúrese de que su grupo de nodos esté asociado a un EKS clúster de Amazon.
- Proporcione las subredes de los nodos de trabajo.
- Si lo desea, adjunte grupos de seguridad, etiquetas de nodos y un grupo de ubicación a su grupo de nodos.

Sintaxis

```
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`

Propiedades que definen los parámetros informáticos del grupo de nodos EKS gestionado por Amazon, como los tipos de EC2 instancias de Amazon y la EC2 instancia de AmazonAMIs.

`ami_type`

El AMI tipo EKS compatible con Amazon.

Obligatorio: sí

Tipo: cadena

Valores posibles: `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`

El ID delAMI.

Obligatorio: no

Tipo: cadena

> **Note**
>
> Si ambos `ami_type` y `ami_id` se especifican en la plantilla, AWS TNB utilizará solo el `ami_id` valor para crear`EKSManagedNode`.

`instance_types`

El tamaño de la instancia.

Obligatorio: sí

Tipo: lista

`key_pair`

El par de EC2 claves para permitir el SSH acceso.

Obligatorio: sí

Tipo: cadena

`root_volume_encryption`

Habilita el EBS cifrado de Amazon para el volumen EBS raíz de Amazon. Si no se proporciona esta propiedad, AWS TNB cifra los volúmenes EBS raíz de Amazon de forma predeterminada.

Obligatorio: no

Predeterminado: true

Tipo: Booleano

`root_volume_encryption_key_arn`

El ARN de la AWS KMS clave. AWS TNBadmite la clave normalARN, la clave multirregional ARN y el aliasARN.

Obligatorio: no

Tipo: cadena

> **Note**
>
> - Si `root_volume_encryption` es falso, no lo incluya`root_volume_encryption_key_arn`.
> - AWS TNBadmite el cifrado del volumen raíz de los EBS respaldados por AMI Amazon.
> - Si el AMI volumen raíz ya está cifrado, debe incluir el formulario `root_volume_encryption_key_arn` para volver AWS TNB a cifrarlo.
> - Si el AMI volumen raíz no está cifrado, AWS TNB utiliza el `root_volume_encryption_key_arn` para cifrarlo.

Si no lo incluye`root_volume_encryption_key_arn`, AWS TNB utiliza la clave predeterminada proporcionada por AWS Key Management Service para cifrar el volumen raíz.

- AWS TNBno descifra un cifrado. AMI

`scaling`

Propiedades que definen los parámetros de escalado del grupo de nodos EKS gestionado por Amazon, como el número deseado de EC2 instancias de Amazon y el número mínimo y máximo de EC2 instancias de Amazon en el grupo de nodos.

`desired_size`

El número de instancias que contiene NodeGroup.

Obligatorio: sí

Tipo: entero

`min_size`

El número mínimo de instancias que contiene NodeGroup.

Obligatorio: sí

Tipo: entero

`max_size`

El número máximo de instancias que contiene NodeGroup.

Obligatorio: sí

Tipo: entero

Propiedades

`node_role`

El ARN IAM rol asociado a la EC2 instancia de Amazon.

Obligatorio: sí

Tipo: cadena

`tags`

Las etiquetas que deben asociarse al recurso.

Obligatorio: no

Tipo: lista

Requisitos

`cluster`

Un AWS.Compute. EKSnodo.

Obligatorio: sí

Tipo: cadena

`subnets`

Un nodo AWS.Networking.Subnet.

Obligatorio: sí

Tipo: lista

`network_interfaces`

Un AWS.Networking. ENInodo. Asegúrese de que las interfaces de red y las subredes estén configuradas en la misma zona de disponibilidad o se producirá un error en la instanciación.

Cuando lo configuras`network_interfaces`, AWS TNB obtiene el permiso correspondiente ENIs de la `multus_role` propiedad si la has incluido en el archivo `multus` AWS.Compute. EKSnodo. De lo contrario, AWS TNB obtiene el permiso correspondiente ENIs de la propiedad node_role.

Obligatorio: no

Tipo: lista

`security_groups`

Un .Networking.AWS SecurityGroupnodo.

Obligatorio: no

Tipo: lista

`placement_group`

Un tosca.nodes.AWS.Compute. PlacementGroupnodo.

Obligatorio: no

Tipo: cadena

`user_data`

Un tosca.nodes.AWS.Compute. UserDatareferencia de nodo. Se pasa un script de datos de usuario a las EC2 instancias de Amazon lanzadas por el grupo de nodos gestionado. Agregue los permisos necesarios para ejecutar datos de usuario personalizados al node_role pasado al grupo de nodos.

Obligatorio: no

Tipo: cadena

`labels`

Una lista de etiquetas de nodos. Una etiqueta de nodo debe tener un nombre y un valor. Cree una etiqueta con los siguientes criterios:

- El nombre y el valor deben estar separados por=.
- El nombre y el valor pueden tener hasta 63 caracteres cada uno.
- La etiqueta puede incluir letras (A-Z, a-z), números (0-9) y los siguientes caracteres: `[-, _, ., *, ?]`
- El nombre y el valor deben empezar y terminar con un alfanumérico o un carácter. ? *

Por ejemplo, `myLabelName1=*NodeLabelValue1` .

Obligatorio: no

Tipo: lista

Ejemplo

```
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.Calcular. EKSSelfManagedNode

AWS TNBadmite los nodos EKS autogestionados de Amazon para automatizar el aprovisionamiento y la gestión del ciclo de vida de los nodos (EC2instancias de Amazon) para los clústeres de Amazon EKS Kubernetes. Para crear un grupo de EKS nodos de Amazon, haga lo siguiente:

- Elija Amazon Machine Images (AMI) para los nodos de trabajadores de su clúster proporcionando el ID delAMI.
- Proporcione un EC2 key pair de Amazon para SSH acceder.
- Asegúrese de que su grupo de nodos esté asociado a un EKS clúster de Amazon.
- Proporciona el tipo de instancia y los tamaños deseado, mínimo y máximo.
- Proporcione las subredes de los nodos de trabajo.
- Si lo desea, adjunte grupos de seguridad, etiquetas de nodos y un grupo de ubicación a su grupo de nodos.

Sintaxis

```
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

Propiedades que definen los parámetros informáticos de los nodos EKS autogestionados de Amazon, como los tipos de instancias de Amazon y la EC2 instancia de AmazonEC2. AMIs

`ami_id`

El AMI ID utilizado para lanzar la instancia. AWS TNBadmite instancias que aprovechanIMDSv2. Para obtener más información, consulte IMDSversión.

Obligatorio: sí

Tipo: cadena

`instance_type`

El tamaño de la instancia.

Obligatorio: sí

Tipo: cadena

`key_pair`

El par de EC2 claves de Amazon para permitir el SSH acceso.

Obligatorio: sí

Tipo: cadena

`root_volume_encryption`

Habilita el EBS cifrado de Amazon para el volumen EBS raíz de Amazon. Si no se proporciona esta propiedad, AWS TNB cifra los volúmenes EBS raíz de Amazon de forma predeterminada.

Obligatorio: no

Predeterminado: true

Tipo: Booleano

`root_volume_encryption_key_arn`

El ARN de la AWS KMS clave. AWS TNBadmite la clave normalARN, la clave multirregional ARN y el aliasARN.

Obligatorio: no

Tipo: cadena

> **Note**
>
> - Si `root_volume_encryption` es falso, no lo incluya`root_volume_encryption_key_arn`.
> - AWS TNBadmite el cifrado del volumen raíz de los EBS respaldados por AMI Amazon.
> - Si el AMI volumen raíz ya está cifrado, debe incluir el formulario `root_volume_encryption_key_arn` para volver AWS TNB a cifrarlo.
> - Si el AMI volumen raíz no está cifrado, AWS TNB utiliza el `root_volume_encryption_key_arn` para cifrarlo.
>
> Si no lo incluye`root_volume_encryption_key_arn`, se AWS TNB utiliza AWS Managed Services para cifrar el volumen raíz.
> - AWS TNBno descifra un cifrado. AMI

`scaling`

Propiedades que definen los parámetros de escalado de los nodos EKS autogestionados de Amazon, como el número deseado de EC2 instancias de Amazon y el número mínimo y máximo de EC2 instancias de Amazon en el grupo de nodos.

`desired_size`

El número de instancias que contiene. NodeGroup

Obligatorio: sí

Tipo: entero

`min_size`

El número mínimo de instancias que contiene NodeGroup.

Obligatorio: sí

Tipo: entero

`max_size`

El número máximo de instancias que contiene NodeGroup.

Obligatorio: sí

Tipo: entero

Propiedades

`node_role`

El ARN IAM rol asociado a la EC2 instancia de Amazon.

Obligatorio: sí

Tipo: cadena

`tags`

Las etiquetas que deben asociarse al recurso. Las etiquetas se propagarán a las instancias creadas por el recurso.

Obligatorio: no

Tipo: lista

Requisitos

`cluster`

Un AWS.Compute. EKSnodo.

Obligatorio: sí

Tipo: cadena

`subnets`

Un nodo AWS.Networking.Subnet.

Obligatorio: sí

Tipo: lista

`network_interfaces`

Un AWS.Networking. ENInodo. Asegúrese de que las interfaces de red y las subredes estén configuradas en la misma zona de disponibilidad o se producirá un error en la instanciación.

Cuando lo configuras`network_interfaces`, AWS TNB obtiene el permiso correspondiente ENIs de la `multus_role` propiedad si la has incluido en el archivo `multus` AWS.Compute. EKSnodo. De lo contrario, AWS TNB obtiene el permiso correspondiente ENIs de la propiedad node_role.

Obligatorio: no

Tipo: lista

`security_groups`

Un .Networking.AWS SecurityGroupnodo.

Obligatorio: no

Tipo: lista

`placement_group`

Un tosca.nodes.AWS.Compute. PlacementGroupnodo.

Obligatorio: no

Tipo: cadena

`user_data`

Un tosca.nodes.AWS.Compute. UserDatareferencia de nodo. Se pasa un script de datos de usuario a las EC2 instancias de Amazon lanzadas por el grupo de nodos autogestionado. Agregue los permisos necesarios para ejecutar datos de usuario personalizados al node_role pasado al grupo de nodos.

Obligatorio: no

Tipo: cadena

`labels`

Una lista de etiquetas de nodos. Una etiqueta de nodo debe tener un nombre y un valor. Cree una etiqueta con los siguientes criterios:

- El nombre y el valor deben estar separados por=.
- El nombre y el valor pueden tener hasta 63 caracteres cada uno.
- La etiqueta puede incluir letras (A-Z, a-z), números (0-9) y los siguientes caracteres: `[-, _, ., *, ?]`
- El nombre y el valor deben empezar y terminar con un alfanumérico o un carácter. `? *`

Por ejemplo, `myLabelName1=*NodeLabelValue1` .

Obligatorio: no

Tipo: lista

Ejemplo

```
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.Calcular. PlacementGroup

Un PlacementGroup nodo admite diferentes estrategias para colocar EC2 instancias de Amazon.

Cuando lanzas un Amazon nuevoEC2instance, el EC2 servicio de Amazon intenta colocar la instancia de forma que todas las instancias estén distribuidas en el hardware subyacente para minimizar los fallos correlacionados. Sin embargo, los grupos de ubicación influyen en la ubicación de un grupo de instancias interdependientes para satisfacer las necesidades de la carga de trabajo.

Sintaxis

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

Propiedades

`strategy`

La estrategia que se utilizará para colocar las EC2 instancias de Amazon.

Obligatorio: sí

Tipo: cadena

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

- CLUSTER— empaqueta las instancias muy juntas dentro de una zona de disponibilidad. Esta estrategia permite que las cargas de trabajo alcancen el rendimiento de red de baja latencia necesario para una node-to-node comunicación estrechamente acoplada, típica de las aplicaciones informáticas de alto rendimiento (). HPC
- PARTITION— distribuye las instancias entre particiones lógicas, de forma que los grupos de instancias de una partición no compartan el hardware subyacente con grupos de instancias de distintas particiones. Esta estrategia suelen utilizarla grandes cargas de trabajo distribuidas y replicadas, como Hadoop, Cassandra y Kafka.
- SPREAD_ RACK — coloca un pequeño grupo de instancias en distintos tipos de hardware subyacentes para reducir los errores correlacionados.
- SPREAD_ HOST: se usa solo con los grupos de ubicación de Outpost. Coloca un pequeño grupo de instancias en distintos equipos de hardware subyacentes para reducir los fallos correlacionados.

`partition_count`

El número de particiones.

Obligatorio: obligatorio solo cuando `strategy` está establecido en `PARTITION`.

Tipo: entero

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

`tags`

Las etiquetas que puede adjuntar al recurso de grupo con ubicación.

Obligatorio: no

Tipo: lista

Ejemplo

```
ExamplePlacementGroup:
```

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

AWS.Compute. UserData

AWS TNBadmite el lanzamiento de EC2 instancias de Amazon con datos de usuario personalizados, a través del UserData nodo del Network Service Descriptor (NSD). Para obtener más información sobre los datos de usuario personalizados, consulte Datos de usuario y scripts de shell en la Guía del EC2 usuario de Amazon.

Durante la instanciación de la red, AWS TNB proporciona el registro de la EC2 instancia de Amazon al clúster mediante un script de datos de usuario. Cuando también se proporcionan datos de usuario personalizados, AWS TNB fusiona ambos scripts y los pasa como un script multimime a Amazon. EC2 El script de datos de usuario personalizado se ejecuta antes que el script de EKS registro en Amazon.

Para utilizar variables personalizadas en el script de datos de usuario, añada un signo de exclamación `!` después de la llave abierta `{`. Por ejemplo, para utilizar `MyVariable` en el script, introduzca: `{!MyVariable}`

> **Note**
>
> - AWS TNBadmite scripts de datos de usuario de hasta 7 KB de tamaño.
> - Como se AWS TNB utiliza AWS CloudFormation para procesar y renderizar el script de `multimime` datos de usuario, asegúrese de que el script cumpla con todas las reglas. AWS CloudFormation

Sintaxis

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

Propiedades

`implementation`

La ruta relativa a la definición del script de datos de usuario. El formato debe ser: `./scripts/script_name.sh`

Obligatorio: sí

Tipo: cadena

`content_type`

Tipo de contenido del script de datos de usuario.

Obligatorio: sí

Tipo: cadena

Valores posibles: `x-shellscript`

Ejemplo

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

AWS.Redes. SecurityGroup

AWS TNBadmite grupos de seguridad para automatizar el aprovisionamiento de grupos de EC2seguridad de Amazon que puede adjuntar a los grupos de nodos del clúster de Amazon EKS Kubernetes.

Sintaxis

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

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

Propiedades

`description`

La descripción del grupo de seguridad. Puede usar hasta 255 caracteres para describir el grupo. Solo puede incluir letras (A-Z y a-z), números (0-9), espacios y los siguientes caracteres especiales: ._-:/()#,@[]+=&;{}!$*

Obligatorio: sí

Tipo: cadena

`name`

Un nombre para el grupo de seguridad. Puede utilizar hasta 255 caracteres para el nombre. Solo puede incluir letras (A-Z y a-z), números (0-9), espacios y los siguientes caracteres especiales: ._-:/()#,@[]+=&;{}!$*

Obligatorio: sí

Tipo: cadena

`tags`

Las etiquetas que puede adjuntar al recurso de grupo de seguridad.

Obligatorio: no

Tipo: lista

Requisitos

`vpc`

Un .Networking.AWS VPCnodo.

Obligatorio: sí

Tipo: cadena

Ejemplo

```
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.Redes. SecurityGroupEgressRule

AWS TNBadmite reglas de salida de grupos de seguridad para automatizar el aprovisionamiento de las reglas de salida de grupos EC2 de seguridad de Amazon que se pueden adjuntar a .Networking. AWS SecurityGroup. Tenga en cuenta que debe proporcionar un cidr_ip/destination_security_group/ destination_prefix_list como destino del tráfico de salida.

Sintaxis

```
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
```

Propiedades

`cidr_ip`

El rango de IPv4 direcciones en CIDR formato. Debe especificar un CIDR rango que permita el tráfico de salida.

Obligatorio: no

Tipo: cadena

`cidr_ipv6`

El rango IPv6 de direcciones en CIDR formato, para el tráfico de salida. Debe especificar un grupo (`destination_security_groupodestination_prefix_list`) de seguridad de destino o un CIDR rango (`cidr_ipocidr_ipv6`).

Obligatorio: no

Tipo: cadena

`description`

Descripción de una regla del grupo de seguridad de salida. Puede usar hasta 255 caracteres para describir la regla.

Obligatorio: no

Tipo: cadena

`destination_prefix_list`

El ID de lista de prefijos de una lista de prefijos VPC gestionada por Amazon existente. Este es el destino de las instancias del grupo de nodos asociado al grupo de seguridad. Para obtener más información sobre las listas de prefijos gestionadas, consulta Listas de prefijos gestionadas en la Guía VPCdel usuario de Amazon.

Obligatorio: no

Tipo: cadena

`from_port`

Si el protocolo es TCP oUDP, este es el inicio del rango de puertos. Si el protocolo es ICMP oICMPv6, este es el número de tipo. Un valor de -1 indica todos los ICMPv6 tipos deICMP/. Si especifica todos los ICMPv6 tiposICMP/, debe especificar todos los ICMPv6 códigosICMP/.

Obligatorio: no

Tipo: entero

`ip_protocol`

El nombre del protocolo IP (tcp, udp, icmp, icmpv6) o el número de protocolo. Utilice -1 para especificar todos los protocolos. Al autorizar las reglas del grupo de seguridad, si especifica -1 o un número de protocolo que no sea tcp, udp, icmp o icmpv6 permite el tráfico en todos los puertos, independientemente del rango de puertos que especifique. Para tcp, udp e icmp debe especificar un rango de puertos. Para icmpv6, el rango de puertos es opcional; si se omite el rango de puertos, se permite el tráfico para todos los tipos y códigos.

Obligatorio: sí

Tipo: cadena

`to_port`

Si el protocolo es TCP oUDP, este es el final del rango de puertos. Si el protocolo es ICMP oICMPv6, este es el código. Un valor de -1 indica todos los ICMPv6 códigosICMP/. Si especifica todos los ICMPv6 tiposICMP/, debe especificar todos los ICMPv6 códigosICMP/.

Obligatorio: no

Tipo: entero

Requisitos

`security_group`

El identificador del grupo de seguridad al que se añade esta regla.

Obligatorio: sí

Tipo: cadena

`destination_security_group`

El ID o la TOSCA referencia del grupo de seguridad de destino al que se permite el tráfico de salida.

Obligatorio: no

Tipo: cadena

Ejemplo

```
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.Redes. SecurityGroupIngressRule

AWS TNBadmite reglas de entrada de grupos de seguridad para automatizar el aprovisionamiento de las reglas de entrada de grupos EC2 de seguridad de Amazon que se pueden adjuntar a .Networking. AWS SecurityGroup. Tenga en cuenta que debe proporcionar un cidr_ip/ source_security_group/source_prefix_list como origen del tráfico de entrada.

Sintaxis

```
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
```

Propiedades

`cidr_ip`

El rango de IPv4 direcciones en CIDR formato. Debe especificar un CIDR rango que permita la entrada de tráfico.

Obligatorio: no

Tipo: cadena

`cidr_ipv6`

El rango IPv6 de direcciones en CIDR formato, para el tráfico de entrada. Debe especificar un grupo (`source_security_grouposource_prefix_list`) de seguridad de origen o un CIDR rango (`cidr_ipocidr_ipv6`).

Obligatorio: no

Tipo: cadena

`description`

La descripción de una regla del grupo de seguridad de entrada. Puede usar hasta 255 caracteres para describir la regla.

Obligatorio: no

Tipo: cadena

`source_prefix_list`

El ID de lista de prefijos de una lista de prefijos VPC gestionada por Amazon existente. Esta es la fuente desde la que se permitirá recibir tráfico a las instancias del grupo de nodos asociadas al grupo de seguridad. Para obtener más información sobre las listas de prefijos gestionadas, consulta Listas de prefijos gestionadas en la Guía VPCdel usuario de Amazon.

Obligatorio: no

Tipo: cadena

`from_port`

Si el protocolo es TCP oUDP, este es el inicio del rango de puertos. Si el protocolo es ICMP oICMPv6, este es el número de tipo. Un valor de -1 indica todos los ICMPv6 tipos deICMP/. Si especifica todos los ICMPv6 tiposICMP/, debe especificar todos los ICMPv6 códigosICMP/.

Obligatorio: no

Tipo: entero

`ip_protocol`

El nombre del protocolo IP (tcp, udp, icmp, icmpv6) o el número de protocolo. Utilice -1 para especificar todos los protocolos. Al autorizar las reglas del grupo de seguridad, si especifica -1 o un número de protocolo que no sea tcp, udp, icmp o icmpv6 permite el tráfico en todos los puertos, independientemente del rango de puertos que especifique. Para tcp, udp e icmp debe especificar un rango de puertos. Para icmpv6, el rango de puertos es opcional; si se omite el rango de puertos, se permite el tráfico para todos los tipos y códigos.

Obligatorio: sí

Tipo: cadena

`to_port`

Si el protocolo es TCP oUDP, este es el final del rango de puertos. Si el protocolo es ICMP oICMPv6, este es el código. Un valor de -1 indica todos los ICMPv6 códigosICMP/. Si especifica todos los ICMPv6 tiposICMP/, debe especificar todos los ICMPv6 códigosICMP/.

Obligatorio: no

Tipo: entero

Requisitos

`security_group`

El identificador del grupo de seguridad al que se añade esta regla.

Obligatorio: sí

Tipo: cadena

`source_security_group`

El ID o la TOSCA referencia del grupo de seguridad de origen desde el que se va a permitir el tráfico de entrada.

Obligatorio: no

Tipo: cadena

Ejemplo

```
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

Puede importar los siguientes AWS recursos a AWS TNB:

- VPC
- Subred
- Tabla de enrutamiento
- Puerta de enlace de Internet
- Security Group

Sintaxis

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

Propiedades

`resource_type`

El tipo de recurso al que se importa AWS TNB.

Obligatorio: no

Tipo: lista

`resource_id`

El identificador del recurso al que se importa AWS TNB.

Obligatorio: no

Tipo: lista

Ejemplo

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

AWS.Redes. ENI

Una interfaz de red es un componente de red lógico en un VPC que representa una tarjeta de red virtual. A una interfaz de red se le asigna una dirección IP automática o manualmente en función de su subred. Tras implementar una EC2 instancia de Amazon en una subred, puede adjuntarle una interfaz de red o desconectar una interfaz de red de esa instancia de Amazon y volver a conectarla a otra EC2 instancia de Amazon EC2 de esa subred. El índice del dispositivo identifica la posición en el orden en que se adjunta.

Sintaxis

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

Propiedades

`device_index`

El índice del dispositivo debe ser mayor que cero.

Obligatorio: sí

Tipo: entero

`source_dest_check`

Indica si la interfaz de red comprueba el origen y el destino. Un valor de `true` significa que la comprobación está habilitada y un valor de `false` significa que la comprobación está deshabilitada.

Valor permitido: verdadero, falso

Predeterminado: true

Obligatorio: no

Tipo: booleano

`tags`

Las etiquetas que deben asociarse al recurso.

Obligatorio: no

Tipo: lista

Requisitos

`subnet`

Un nodo AWS.Networking.Subnet.

Obligatorio: sí

Tipo: cadena

`security_groups`

Un AWS.Networking. SecurityGroupnodo.

Obligatorio: no

Tipo: cadena

Ejemplo

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

AWS.HookExecution

Un enlace de ciclo de vida le permite ejecutar sus propios scripts como parte de la instanciación de su infraestructura y red.

Sintaxis

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

Capacidades

`execution`

Propiedades del motor de ejecución de enlaces que ejecuta los guiones de enlace.

`type`

Tipo de motor de ejecución de enlaces.

Obligatorio: no

Tipo: cadena

Valores posibles: `CODE_BUILD`

Requisitos

`definition`

Un AWS. HookDefinitionNodo .Bash.

Obligatorio: sí

Tipo: cadena

`vpc`

Un AWS.Networking. VPCnodo.

Obligatorio: sí

Tipo: cadena

Ejemplo

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

AWS.Redes. InternetGateway

Define un nodo AWS de Internet Gateway.

Sintaxis

```
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`

Propiedades que definen la conexión de enrutamiento dentro delVPC. Debe incluir la propiedad `dest_cidr` o `ipv6_dest_cidr`.

`dest_cidr`

El IPv4 CIDR bloque utilizado para el destino coincide. Esta propiedad se utiliza para crear una ruta en `RouteTable` y su valor se utiliza como `DestinationCidrBlock`.

Obligatorio: no si se ha incluido la propiedad `ipv6_dest_cidr`.

Tipo: cadena

`ipv6_dest_cidr`

El IPv6 CIDR bloque utilizado para la coincidencia de destino.

Obligatorio: no si se ha incluido la propiedad `dest_cidr`.

Tipo: cadena

Propiedades

`tags`

Las etiquetas que deben asociarse al recurso.

Obligatorio: no

Tipo: lista

`egress_only`

Una propiedad IPv6 específica. Indica si la puerta de enlace de Internet es solo para la comunicación de salida o no. Si `egress_only` es verdadero, debe definir la propiedad `ipv6_dest_cidr`.

Obligatorio: no

Tipo: booleano

Requisitos

`vpc`

Una AWS.Networking. VPCnodo.

Obligatorio: sí

Tipo: cadena

`route_table`

Un AWS.Networking. RouteTablenodo.

Obligatorio: sí

Tipo: cadena

Ejemplo

```
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.Redes. RouteTable

Una tabla de rutas contiene un conjunto de reglas, denominadas rutas, que determinan hacia dónde se dirige el tráfico de red procedente de las subredes de su puerta de enlace VPC o de su puerta de enlace. Debe asociar una tabla de rutas a. VPC

Sintaxis

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

Propiedades

`tags`

Etiquetas que deben asociarse a este recurso.

Obligatorio: no

Tipo: lista

Requisitos

`vpc`

Un AWS.Networking. VPCnodo.

Obligatorio: sí

Tipo: cadena

Ejemplo

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

AWS.Networking.Subnet

Una subred es un rango de direcciones IP de su VPC propiedad y debe residir completamente dentro de una zona de disponibilidad. Debe especificar un CIDR bloqueVPC, una zona de disponibilidad y una tabla de enrutamiento para la subred. También debe definir si su subred es privada o pública.

Sintaxis

```
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
```

Propiedades

`type`

Indica si las instancias lanzadas en esta subred reciben una dirección públicaIPv4.

Obligatorio: sí

Tipo: cadena

Los valores posibles son: `PUBLIC` | `PRIVATE`

`availability_zone`

La zona de disponibilidad de la subred. Este campo admite las zonas de AWS disponibilidad de una AWS región, por ejemplo `us-west-2` (EE.UU. Oeste (Oregón)). También es compatible con las Zonas AWS Locales dentro de la Zona de Disponibilidad, por ejemplo`us-west-2-lax-1a`.

Obligatorio: sí

Tipo: cadena

`cidr_block`

El CIDR bloque de la subred.

Obligatorio: no

Tipo: cadena

`ipv6_cidr_block`

El CIDR bloque utilizado para crear la IPv6 subred. Si se incluye esta propiedad, no incluya `ipv6_cidr_block_suffix`.

Obligatorio: no

Tipo: cadena

`ipv6_cidr_block_suffix`

El sufijo hexadecimal de 2 dígitos del IPv6 CIDR bloque para la subred creada a través de Amazon. VPC Use el siguiente formato *`2-digit hexadecimal`*`::/`*`subnetMask`*

Si se incluye esta propiedad, no incluya `ipv6_cidr_block`.

Obligatorio: no

Tipo: cadena

`outpost_arn`

En ARN el AWS Outposts que se creará la subred. Añade esta propiedad a la NSD plantilla si quieres lanzar nodos EKS autogestionados de Amazon en AWS Outposts ella. Para obtener más información, consulta Amazon EKS on AWS Outposts en la Guía del EKS usuario de Amazon.

Si añade esta propiedad a la NSD plantilla, debe establecer el valor de la `availability_zone` propiedad en la zona de disponibilidad del AWS Outposts.

Obligatorio: no

Tipo: cadena

`tags`

Las etiquetas que deben asociarse al recurso.

Obligatorio: no

Tipo: lista

Requisitos

`vpc`

Un AWS.Networking. VPCnodo.

Obligatorio: sí

Tipo: cadena

`route_table`

Un AWS.Networking. RouteTablenodo.

Obligatorio: sí

Tipo: cadena

Ejemplo

```
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.Despliegue. VNFDeployment

Las implementaciones de NF se modelan proporcionando la infraestructura y la aplicación asociadas a ellas. El atributo de clúster especifica el EKS clúster que alojará suNFs. El atributo vnfs especifica las funciones de red de su implementación. También puedes proporcionar operaciones opcionales de enlaces de ciclo de vida del tipo pre_create y post_create para ejecutar instrucciones específicas de tu implementación, como llamar a un sistema de gestión de inventario. API

Sintaxis

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

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

Requisitos

`deployment`

Un .Deployment.AWS VNFDeploymentnodo.

Obligatorio: no

Tipo: cadena

`cluster`

Un AWS.Compute. EKSnodo.

Obligatorio: sí

Tipo: cadena

`vnfs`

Un AWS. VNFnodo.

Obligatorio: sí

Tipo: cadena

Interfaces

Enlaces

Define la etapa en la que se ejecutan los enlaces del ciclo de vida.

`pre_create`

Un AWS. HookExecutionnodo. Este enlace se ejecuta antes de que se implemente el nodo `VNFDeployment`.

Obligatorio: no

Tipo: cadena

`post_create`

Un AWS. HookExecutionnodo. Este enlace se ejecuta después de la implementación del nodo `VNFDeployment`.

Obligatorio: no

Tipo: cadena

Ejemplo

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

AWS.Redes. VPC

Debe especificar un CIDR bloque para su nube privada virtual (VPC).

Sintaxis

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

Propiedades

`cidr_block`

El rango de IPv4 redes para elVPC, en CIDR notación.

Obligatorio: sí

Tipo: cadena

`ipv6_cidr_block`

El IPv6 CIDR bloque utilizado para crear elVPC.

Valor permitido: `AMAZON_PROVIDED`

Obligatorio: no

Tipo: cadena

`dns_support`

Indica si las instancias se lanzaron en los DNS nombres de host VPC get.

Obligatorio: no

Tipo: booleano

Valor predeterminado: `false`

`tags`

Etiquetas que deben asociarse a este recurso.

Obligatorio: no

Tipo: lista

Ejemplo

```
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.Redes. NATGateway

Puede definir un nodo de NAT puerta de enlace público o privado a través de una subred. En el caso de una puerta de enlace pública, si no proporciona un identificador de asignación de IP elástica, AWS TNB asignará una IP elástica a su cuenta y la asociará a la puerta de enlace.

Sintaxis

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

Propiedades

`subnet`

La referencia del nodo AWS.Networking.Subnet.

Obligatorio: sí

Tipo: cadena

`internet_gateway`

El AWS.Networking. InternetGatewayreferencia de nodo.

Obligatorio: sí

Tipo: cadena

Propiedades

`type`

Indica si la puerta de enlace es pública o privada.

Valor permitido: `PUBLIC`, `PRIVATE`

Obligatorio: sí

Tipo: cadena

`eip_allocation_id`

El ID que representa la asignación de la dirección IP elástica.

Obligatorio: no

Tipo: cadena

`tags`

Etiquetas que deben asociarse a este recurso.

Obligatorio: no

Tipo: lista

Ejemplo

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

AWS.Networking.Route

Puede definir un nodo de ruta que asocie la ruta de destino a la NAT puerta de enlace como recurso de destino y agregue la ruta a la tabla de rutas asociada.

Sintaxis

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

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

Propiedades

`dest_cidr_blocks`

La lista de IPv4 rutas de destino al recurso de destino.

Obligatorio: sí

Tipo: lista

Tipo de miembro: cadena

Propiedades

`nat_gateway`

El AWS.Networking. NATGatewayreferencia de nodo.

Obligatorio: sí

Tipo: cadena

`route_table`

El AWS.Networking. RouteTablereferencia de nodo.

Obligatorio: sí

Tipo: cadena

Ejemplo

```
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
```

Nodos comunes

Defina los nodos para NSD yVNFD.

- AWS. HookDefinition.Bash

AWS.HookDefinition.Bash

Define una AWS HookDefinition pulgadabash.

Sintaxis

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

Propiedades

`implementation`

La ruta relativa a la definición del enlace. El formato debe ser: `./hooks/`*`script_name`*`.sh`

Obligatorio: sí

Tipo: cadena

`environment_variables`

Las variables de entorno del guion bash de enlace. Utilice el siguiente formato: **`envName`**`=`**`envValue`** con la siguiente expresión regular: `^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`

Asegúrese de que el valor **`envName`**`=`**`envValue`** cumpla los siguientes criterios:

- No utilice espacios.
- Comience **`envName`** con una letra (A-Z o a-z) o un número (0-9).

- No inicie el nombre de la variable de entorno con las siguientes palabras clave AWS TNB reservadas (no distingue entre mayúsculas y minúsculas):
  - CODEBUILD
  - TNB
  - HOME
  - AWS
- Puede utilizar cualquier número de letras (A-Z o a-z), números (0-9) y caracteres especiales `-` y `_` para **`envName`** y **`envValue`**.

Ejemplo: `A123-45xYz=Example_789`

Obligatorio: no

Tipo: lista

`execution_role`

El rol de ejecución de enlaces.

Obligatorio: sí

Tipo: cadena

Ejemplo

```
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"
```

Seguridad en AWS TNB

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento que se aplican a AWS Telco Network Builder, consulte AWS Servicios incluidos en el ámbito de aplicación del programa AWS Servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS TNB. Los siguientes temas muestran cómo configurarlo AWS TNB para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus AWS TNB recursos.

Contenido

Protección de datos en AWS TNB

El modelo de se aplica a protección de datos en AWS Telco Network Builder. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte la sección Privacidad de datos FAQ. Para obtener información sobre la protección de datos en Europa, consulte el modelo de responsabilidad AWS compartida y la entrada del GDPR blog sobre AWS seguridad.

Para proteger los datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactorial (MFA) con cada cuenta.
- UseSSL/TLSpara comunicarse con AWS los recursos. Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Configure API y registre la actividad del usuario con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita entre FIPS 140 y 3 módulos criptográficos validados para acceder a AWS través de una interfaz de línea de comandos o unaAPI, utilice un FIPS terminal. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la Norma federal de procesamiento de información () FIPS 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS TNB o Servicios de AWS utiliza la consola, API AWS CLI, o. AWS SDKs Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si

proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya información sobre las credenciales URL para validar su solicitud a ese servidor.

Gestión de datos

Cuando cierres tu AWS cuenta, AWS TNB marca tus datos para eliminarlos y los elimina para que no se puedan usar. Si reactivas tu AWS cuenta en un plazo de 90 días, AWS TNB restaura tus datos. Transcurridos 120 días, borra tus datos de AWS TNB forma permanente. AWS TNBtambién cierra las redes y elimina los paquetes de funciones y los paquetes de red.

Cifrado en reposo

AWS TNBsiempre cifra todos los datos almacenados en el servicio en reposo sin necesidad de ninguna configuración adicional. Este cifrado es totalmente automático. AWS Key Management Service

Cifrado en tránsito

AWS TNBprotege todos los datos en tránsito mediante Transport Layer Security (TLS) 1.2.

Es su responsabilidad cifrar los datos entre sus agentes de simulación y sus clientes.

Privacidad del tráfico entre redes

AWS TNBlos recursos de cómputo residen en una nube privada virtual (VPC) compartida por todos los clientes. Todo AWS TNB el tráfico interno permanece dentro AWS de la red y no atraviesa Internet. Las conexiones entre sus agentes de simulación y sus clientes se enrutan a través de Internet.

Administración de identidad y acceso para AWS TNB

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. IAMlos administradores controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar AWS TNB los recursos. IAMes una Servicio de AWS que puede utilizar sin coste adicional.

Contenido

- Administración de acceso mediante políticas
- ¿Cómo AWS TNB funciona con IAM
- Ejemplos de políticas basadas en identidad para Creador de redes de telecomunicaciones de AWS
- Solución de problemas de AWS identidad y acceso a Telco Network Builder

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se realice AWS TNB.

Usuario del servicio: si utiliza el AWS TNB servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más AWS TNB funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una función en AWS TNB, consulteSolución de problemas de AWS identidad y acceso a Telco Network Builder.

Administrador de servicios: si está a cargo de AWS TNB los recursos de su empresa, probablemente tenga acceso total a ellos AWS TNB. Su trabajo consiste en determinar a qué AWS TNB funciones y recursos deben acceder los usuarios del servicio. A continuación, debe enviar solicitudes a su IAM administrador para cambiar los permisos de los usuarios del servicio. Revise la información de esta página para comprender los conceptos básicos deIAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con AWS TNB, consulte¿Cómo AWS TNB funciona con IAM.

IAMadministrador: si es IAM administrador, puede que desee obtener más información sobre cómo puede redactar políticas para administrar el acceso a ellas AWS TNB. Para ver ejemplos de políticas AWS TNB basadas en la identidad que puede utilizarIAM, consulte. Ejemplos de políticas basadas en identidad para Creador de redes de telecomunicaciones de AWS

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como IAM usuario o asumiendo un IAM rol. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAMIdentity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son

ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, el administrador configuró previamente la federación de identidades mediante roles. IAM Cuando accede AWS mediante la federación, asume indirectamente un rol.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS incluye un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte la versión 4 de la AWS firma para ver API las solicitudes en la Guía del IAM usuario.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactorial (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactorial en la Guía del AWS IAM Identity Center usuario y Autenticación AWS multifactorial IAM en la Guía del IAM usuario.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de tareas que requieren que inicie sesión como usuario root, consulte Tareas que requieren credenciales de usuario root en la Guía del IAM usuario.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios empresarial, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de

identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones Cuentas de AWS . Para obtener información sobre IAM Identity Center, consulte ¿Qué es IAM Identity Center? en la Guía AWS IAM Identity Center del usuario.

Usuarios y grupos de IAM

Un IAMusuario es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte Rotar las claves de acceso con regularidad para los casos de uso que requieran credenciales de larga duración en la Guía del IAM usuario.

Un IAMgrupo es una identidad que especifica un conjunto de IAM usuarios. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar IAM los recursos.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Casos de uso para IAM usuarios en la Guía del IAM usuario.

IAMroles

Un IAMrol es una identidad dentro de ti Cuenta de AWS que tiene permisos específicos. Es similar a un IAM usuario, pero no está asociado a una persona específica. Para asumir temporalmente un IAM rol en la AWS Management Console, puede cambiar de un IAM rol de usuario a uno (consola). Puede asumir un rol llamando a una AWS API operación AWS CLI o o utilizando una operación personalizadaURL. Para obtener más información sobre los métodos de uso de roles, consulte Métodos para asumir un rol en la Guía del IAM usuario.

IAMlos roles con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información sobre los roles para la federación, consulte Creación de un rol para un proveedor de identidad externo en la Guía del IAM usuario. Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el conjunto de permisos con un rol en. IAM Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos IAM de usuario temporales: un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.
- Acceso multicuenta: puedes usar un IAM rol para permitir que alguien (un responsable de confianza) de una cuenta diferente acceda a los recursos de tu cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para conocer la diferencia entre las funciones y las políticas basadas en recursos para el acceso multicuenta, consulta el tema sobre el acceso a los recursos entre cuentas IAM en la Guía del IAM usuario.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros. Servicios de AWS Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un IAM usuario o un rol para realizar acciones en AWS ellas, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FASutiliza los permisos del principal que llama a an Servicio de AWS, junto con los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar las sesiones de acceso.
  - Función de servicio: una función de servicio es una IAMfunción que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentroIAM. Para obtener más información, consulte Crear un rol para delegar permisos Servicio de AWS en un rol en el IAMManual del usuario.

  - Función vinculada a un servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.
- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y que realizan AWS CLI o AWS API solicitan. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Uso de un IAM rol para conceder permisos a aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del IAM usuario.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como JSON documentos. Para obtener más información sobre la estructura y el contenido de los documentos de JSON políticas, consulte Descripción general de JSON las políticas en la Guía del IAM usuario.

Los administradores pueden usar AWS JSON las políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

IAMlas políticas definen los permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de AWS Management Console AWS CLI, el o el AWS API.

Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Definir IAM permisos personalizados con políticas administradas por el cliente en la Guía del usuario. IAM

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y funciones de su empresa. Cuenta de AWS Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para saber cómo elegir entre una política gestionada o una política integrada, consulte Elegir entre políticas gestionadas y políticas integradas en la Guía del IAM usuario.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puede usar políticas AWS administradas desde una política IAM basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Amazon S3 AWS WAF y Amazon VPC son ejemplos de servicios compatiblesACLs. Para obtener más informaciónACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAMusuario o rol). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte los límites de los permisos para IAM las entidades en la Guía del IAMusuario.
- Políticas de control de servicios (SCPs): SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU) AWS Organizations. AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. SCPLimita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre OrganizationsSCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte las políticas de sesión en la Guía del IAM usuario.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del IAM usuario.

¿Cómo AWS TNB funciona con IAM

Antes de utilizar IAM para gestionar el acceso a AWS TNB, infórmese sobre las IAM funciones disponibles para su uso AWS TNB.

IAMfunciones que puede utilizar con AWS Telco Network Builder

IAMcaracterística	AWS TNBapoyo
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACLs	No
ABAC(etiquetas en las políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	No
Roles vinculados al servicio	No

Para obtener una visión general de cómo AWS TNB funcionan otros AWS servicios con la mayoría de las IAM funciones, consulte AWS los servicios con los que funcionan IAM en la Guía del IAM usuario.

Políticas basadas en la identidad para AWS TNB

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas

políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Definir IAM permisos personalizados con políticas administradas por el cliente en la Guía del usuario. IAM

Con las políticas IAM basadas en la identidad, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para obtener más información sobre todos los elementos que puede utilizar en una JSON política, consulte la referencia sobre los elementos de la IAM JSON política en la Guía del IAMusuario.

Ejemplos de políticas basadas en la identidad para AWS TNB

Para ver ejemplos de políticas AWS TNB basadas en la identidad, consulte. Ejemplos de políticas basadas en identidad para Creador de redes de telecomunicaciones de AWS

Políticas basadas en recursos incluidas AWS TNB

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar una cuenta completa o IAM entidades de otra cuenta como principales en una política basada en recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el IAM administrador de la cuenta de confianza también debe conceder permiso a la entidad principal (usuario o rol) para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Acceso a recursos entre cuentas IAM en la Guía del IAM usuario.

Acciones políticas para AWS TNB

Compatibilidad con las acciones de política: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de política suelen tener el mismo nombre que la AWS API operación asociada. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de AWS TNB acciones, consulte las acciones definidas por AWS Telco Network Builder en la Referencia de autorización de servicios.

Las acciones políticas AWS TNB utilizan el siguiente prefijo antes de la acción:

```
tnb
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `List`, incluya la siguiente acción:

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

Para ver ejemplos de políticas AWS TNB basadas en la identidad, consulte. Ejemplos de políticas basadas en identidad para Creador de redes de telecomunicaciones de AWS

Recursos de políticas para AWS TNB

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso mediante su nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de AWS TNB recursos y sus respectivos tiposARNs, consulte los recursos definidos por AWS Telco Network Builder en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar cada recurso, consulte Acciones definidas por AWS Telco Network Builder. ARN

Para ver ejemplos de políticas AWS TNB basadas en la identidad, consulte. Ejemplos de políticas basadas en identidad para Creador de redes de telecomunicaciones de AWS

Claves de condición de la política para AWS TNB

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder a un IAM usuario permiso para acceder a un recurso solo si está etiquetado con su nombre de IAM usuario. Para obtener más información, consulte los elementos IAM de la política: variables y etiquetas en la Guía del IAM usuario.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del IAMusuario.

Para ver una lista de claves de AWS TNB condición, consulte las claves de condición de AWS Telco Network Builder en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte Acciones definidas por AWS Telco Network Builder.

Para ver ejemplos de políticas AWS TNB basadas en la identidad, consulte. Ejemplos de políticas basadas en identidad para Creador de redes de telecomunicaciones de AWS

ACLsen AWS TNB

SoportesACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

ABACcon AWS TNB

Soportes ABAC (etiquetas en las políticas): Sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a IAM entidades (usuarios o roles) y a muchos AWS recursos. Etiquetar entidades y recursos es el primer paso deABAC. Luego, diseñe ABAC políticas para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso al que está intentando acceder.

ABACes útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más informaciónABAC, consulte Definir permisos con ABAC autorización en la Guía del IAM usuario. Para ver un tutorial con los pasos de configuraciónABAC, consulte Usar el control de acceso basado en atributos (ABAC) en la Guía del IAMusuario.

Uso de credenciales temporales con AWS TNB

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta la sección Servicios de AWS Cómo trabajar con credenciales temporales IAM en la Guía del IAM usuario.

Está utilizando credenciales temporales si inicia sesión AWS Management Console con cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de rol, consulte Cambiar de un rol de usuario a un IAM rol (consola) en la Guía del IAMusuario.

Puede crear credenciales temporales manualmente con la tecla AWS CLI o AWS API. A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales entre servicios para AWS TNB

Admite sesiones de acceso directo (FAS): Sí

Cuando utilizas un IAM usuario o un rol para realizar acciones en AWSél, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FASutiliza los permisos del principal que llama a una Servicio de AWS, junto con los que solicita, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para

realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar las sesiones de acceso.

Roles de servicio para AWS TNB

Compatible con roles de servicio: No

Una función de servicio es una IAMfunción que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentroIAM. Para obtener más información, consulte Crear un rol para delegar permisos Servicio de AWS en un rol en el IAMManual del usuario.

Funciones vinculadas a servicios para AWS TNB

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.

Ejemplos de políticas basadas en identidad para Creador de redes de telecomunicaciones de AWS

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar AWS TNB recursos. Tampoco pueden realizar tareas con AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS API. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

Para obtener información sobre cómo crear una política IAM basada en la identidad mediante estos documentos de JSON política de ejemplo, consulte Crear IAM políticas (consola) en la Guía del IAMusuario.

Para obtener más información sobre las acciones y los tipos de recursos definidos por cada uno de ellos AWS TNB, incluido el ARNs formato de cada uno de ellos, consulte las claves de condición, recursos y acciones de AWS Telco Network Builder en la Referencia de autorización de servicios.

Contenido

- Prácticas recomendadas sobre las políticas
- Uso de la AWS TNB consola
- Ejemplos de políticas de roles de servicio
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear AWS TNB recursos de su cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costes adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Para obtener más información, consulte las políticas AWS gestionadas o las políticas AWS gestionadas para las funciones laborales en la Guía del IAM usuario.
- Aplique permisos con privilegios mínimos: cuando establezca permisos con IAM políticas, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Para obtener más información sobre cómo IAM aplicar permisos, consulte Políticas y permisos IAM en la IAM Guía del usuario.
- Utilice las condiciones en IAM las políticas para restringir aún más el acceso: puede añadir una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de política para especificar que todas las solicitudes deben enviarse medianteSSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte los elementos IAM JSON de la política: Condición en la Guía del IAM usuario.
- Utilice IAM Access Analyzer para validar sus IAM políticas y garantizar permisos seguros y funcionales: IAM Access Analyzer valida las políticas nuevas y existentes para que se ajusten al lenguaje de las políticas (JSON) y IAM a las IAM mejores prácticas. IAMAccess Analyzer proporciona más de 100 comprobaciones de políticas y recomendaciones prácticas para ayudarle

a crear políticas seguras y funcionales. Para obtener más información, consulte Validar políticas con IAM Access Analyzer en la Guía del IAM usuario.

- Requerir autenticación multifactorial (MFA): si se encuentra en una situación en la que se requieren IAM usuarios o un usuario raíz Cuenta de AWS, actívela MFA para aumentar la seguridad. Para solicitarlo MFA cuando se convoque a API las operaciones, añada MFA condiciones a sus políticas. Para obtener más información, consulte APIAcceso seguro con MFA en la Guía del IAM usuario.

Para obtener más información sobre las prácticas recomendadasIAM, consulte las prácticas recomendadas de seguridad IAM en la Guía del IAM usuario.

Uso de la AWS TNB consola

Para acceder a la consola de AWS Telco Network Builder, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los AWS TNB recursos de su Cuenta de AWS cuenta. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que realicen llamadas únicamente al AWS CLI o al AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la API operación que están intentando realizar.

Ejemplos de políticas de roles de servicio

Como administrador, usted es el propietario y administrador de los recursos que AWS TNB crea, tal y como se definen en las plantillas de servicio y entorno. Debe asignar funciones de IAM servicio a su cuenta para poder crear recursos AWS TNB para la administración del ciclo de vida de la red.

Un rol IAM de servicio le permite AWS TNB realizar llamadas a los recursos en su nombre para crear instancias y administrar sus redes. Si especificas un rol de servicio, AWS TNB usa la credencial de ese rol.

El rol de servicio y su política de permisos se crean con el IAM servicio. Para obtener más información sobre la creación de un rol de servicio, consulte Crear un rol para delegar permisos a un AWS servicio en la Guía del IAM usuario.

AWS TNBrol de servicio

Como miembro del equipo de la plataforma, como administrador, puede crear un rol de AWS TNB servicio y asignárselo AWS TNB. Esta función le permite realizar llamadas AWS TNB a otros servicios, como Amazon Elastic Kubernetes Service, y aprovisionar la infraestructura necesaria para su red AWS CloudFormation y aprovisionar las funciones de red tal como se definen en su. NSD

Le recomendamos que utilice la siguiente política de IAM rol y confianza para su AWS TNB función de servicio. Al determinar el alcance de los permisos de esta política, tenga en cuenta que AWS TNB pueden fallar si se producen errores de acceso denegado en relación con los recursos no incluidos en su política.

El código siguiente muestra una política de roles de AWS TNB servicio:

```
{
    "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"
        }
    ]
}
```

El siguiente código muestra la política AWS TNB de confianza del servicio:

```
{
  "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 TNBrol de servicio para el EKS clúster de Amazon

Cuando crea un EKS recurso de Amazon en suNSD, proporciona el `cluster_role` atributo para especificar qué rol se utilizará para crear su EKS clúster de Amazon.

El siguiente ejemplo muestra una AWS CloudFormation plantilla que crea un rol de AWS TNB servicio para la política de EKS clústeres de 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 obtener más información sobre IAM los roles que utilizan AWS CloudFormation plantillas, consulte las siguientes secciones de la Guía del AWS CloudFormation usuario:

- AWS:IAM: :Rol
- Selección de una plantilla de pila

AWS TNBrol de servicio para el grupo de EKS nodos de Amazon

Cuando crea un grupo de EKS nodos de Amazon en sus recursosNSD, proporciona el `node_role` atributo para especificar qué rol se utilizará para crear su grupo de EKS nodos de Amazon.

El siguiente ejemplo muestra una AWS CloudFormation plantilla que crea un rol de AWS TNB servicio para la política de grupo de EKS nodos de 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 obtener más información sobre IAM los roles que utilizan AWS CloudFormation plantillas, consulte las siguientes secciones de la Guía del AWS CloudFormation usuario:

- AWS:IAM: :Rol
- Selección de una plantilla de pila

AWS TNBfunción de servicio para Multus

Cuando crea un EKS recurso de Amazon en su plantilla de despliegue NSD y desea administrar Multus como parte de su plantilla de despliegue, debe proporcionar el `multus_role` atributo para especificar qué función se utilizará para administrar Multus.

El siguiente ejemplo muestra una AWS CloudFormation plantilla que crea un rol de AWS TNB servicio para una política de 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 obtener más información sobre IAM los roles que utilizan AWS CloudFormation plantillas, consulte las siguientes secciones de la Guía del AWS CloudFormation usuario:

- AWS:IAM: :Rol
- Selección de una plantilla de pila

AWS TNBfunción de servicio para una política vinculada al ciclo de vida

Cuando su paquete de funciones NSD o de red utiliza un enlace de ciclo de vida, necesita un rol de servicio que le permita crear un entorno para la ejecución de sus enlaces de ciclo de vida.

> **Note**
>
> Su política de enlace de ciclo de vida debe basarse en lo que intente hacer su enlace de ciclo de vida.

El siguiente ejemplo muestra una AWS CloudFormation plantilla que crea un rol de AWS TNB servicio para una política de enlace del 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 obtener más información sobre IAM los roles que utilizan AWS CloudFormation plantillas, consulte las siguientes secciones de la Guía del AWS CloudFormation usuario:

- AWS:IAM: :Rol
- Selección de una plantilla de pila

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo se puede crear una política que permita a IAM los usuarios ver las políticas integradas y administradas asociadas a su identidad de usuario. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la tecla o. AWS CLI AWS API

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

```
            "Resource": "*"
        }
    ]
}
```

Solución de problemas de AWS identidad y acceso a Telco Network Builder

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con AWS TNB yIAM.

Problemas

- No estoy autorizado a realizar ninguna acción en AWS TNB
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis AWS TNB recursos

No estoy autorizado a realizar ninguna acción en AWS TNB

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

El siguiente ejemplo de error se produce cuando el `mateojackson` IAM usuario intenta usar la consola para ver detalles sobre un *`my-example-widget`* recurso ficticio pero no tiene los `tnb:`*`GetWidget`* permisos ficticios.

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

En este caso, la política de Mateo se debe actualizar para permitirle acceder al recurso *`my-example-widget`* mediante la acción `tnb:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con AWS el administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibes un mensaje de error que indica que no estás autorizado a realizar la `iam:PassRole` acción, debes actualizar tus políticas para que puedas transferirle AWS TNB una función.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

El siguiente ejemplo de error se produce cuando un IAM usuario denominado `marymajor` intenta utilizar la consola para realizar una acción en ella. AWS TNB Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con AWS el administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis AWS TNB recursos

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que respaldan las políticas basadas en recursos o las listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para más información, consulte lo siguiente:

- Para saber si AWS TNB es compatible con estas funciones, consulte. ¿Cómo AWS TNB funciona con IAM
- Para obtener información sobre cómo proporcionar acceso a los recursos de su propiedad, consulte Proporcionar acceso a un IAM usuario en otro Cuenta de AWS de su propiedad en la Guía del IAM usuario. Cuentas de AWS
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo permitir el acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del IAM usuario.

- Para obtener información sobre cómo proporcionar acceso mediante la federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (federación de identidades) en la Guía del IAM usuario.
- Para saber la diferencia entre el uso de roles y políticas basadas en recursos para el acceso entre cuentas, consulte el acceso a recursos entre cuentas IAM en la Guía del usuario. IAM

Validación de conformidad para AWS TNB

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.
- Diseñando una arquitectura basada en la HIPAA seguridad y el cumplimiento en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar las empresas AWS para crear HIPAA aplicaciones aptas.

> **Note**
>
> No todos son aptos. Servicios de AWS HIPAA Para obtener más información, consulta la Referencia de servicios HIPAA aptos.

- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde la perspectiva del cumplimiento. En las guías se resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y se orientan a los controles de seguridad en varios

marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).

- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, por ejemplo PCIDSS, cumpliendo con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS consumo para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en AWS TNB

La infraestructura AWS global se basa en zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global.AWS

AWS TNBejecuta su servicio de red en EKS clústeres de una nube privada virtual (VPC) en la AWS región que elija.

Seguridad de la infraestructura en AWS TNB

Como servicio gestionado, AWS Telco Network Builder está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las API llamadas AWS publicadas para acceder a AWS TNB través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Cifre suites con perfecto secreto (PFS), como (Ephemeral Diffie-Hellman) o DHE ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben firmarse con un identificador de clave de acceso y una clave de acceso secreta asociada a un director. IAM También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

A continuación, se muestran algunos ejemplos de responsabilidades compartidas:

- AWS es responsable de proteger los componentes que dan soporte AWS TNB, entre los que se incluyen:
  - Instancias de cómputo (también conocidas como trabajadores)
  - Base de datos interna
  - Comunicaciones de red entre componentes internos
  - La interfaz AWS TNB de programación de aplicaciones (API)
  - AWS Kits de desarrollo de software (SDK)
- Usted es responsable de proteger el acceso a sus AWS recursos y a los componentes de su carga de trabajo, incluidos (entre otros):
  - IAMusuarios, grupos, funciones y políticas
  - Depósitos de S3 para los que se almacenan los datos AWS TNB
  - Otros Servicios de AWS recursos que utiliza para respaldar el servicio de red a través del cual aprovisionó AWS TNB
  - Su código de la aplicación

- Conexiones entre el servicio de red a través del cual aprovisionó AWS TNB y sus clientes

> **Important**
>
> Usted es responsable de implementar un plan de recuperación ante desastres que pueda recuperar de manera efectiva un servicio de red a través del cual ha aprovisionado. AWS TNB

Modelo de seguridad de la conectividad de red

Los servicios de red que aprovisiona AWS TNB se ejecutan en instancias informáticas dentro de una nube privada virtual (VPC) ubicada en AWS la región que seleccione. A VPC es una red virtual en la AWS nube que aísla la infraestructura por carga de trabajo o entidad organizativa. La comunicación entre las instancias informáticas VPCs internas permanece dentro de la AWS red y no se transmite a través de Internet. Algunas comunicaciones de los servicios internos se transmiten por Internet y están cifradas. Los servicios de red aprovisionados AWS TNB para todos los clientes que se ejecutan en la misma región comparten las mismas VPC características. Los servicios de red aprovisionados AWS TNB para diferentes clientes utilizan instancias informáticas independientes dentro de la misma. VPC

Las comunicaciones entre los clientes del servicio de red y el servicio de red AWS TNB atraviesan Internet. AWS TNBno gestiona estas conexiones. Es su responsabilidad proteger las conexiones de sus clientes.

Sus conexiones AWS TNB pasan por AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDKs están cifradas.

IMDSversión

AWS TNBadmite instancias que utilizan la versión 2 (IMDSv2) de Instance Metadata Service, un método orientado a la sesión. IMDSv2incluye una seguridad superior a. IMDSV1 Para obtener más información, consulte Mejore la defensa contra firewalls abiertos, proxies inversos y SSRF vulnerabilidades con mejoras en el Amazon EC2 Instance Metadata Service.

Al lanzar la instancia, debe usar. IMDSv2 Para obtener más informaciónIMDSv2, consulte Uso IMDSv2 en la Guía del EC2 usuario de Amazon.

Monitorización AWS TNB

La supervisión es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de AWS TNB sus demás AWS soluciones. AWS permite AWS CloudTrail observar AWS TNB, informar cuando algo anda mal y tomar medidas automáticas cuando sea apropiado.

Se utiliza CloudTrail para capturar información detallada sobre las llamadas realizadas a AWS APIs. Puede almacenar estas llamadas como archivos de registro en Amazon S3. Puede usar estos CloudTrail registros para determinar información como qué llamada se realizó, la dirección IP de origen de la llamada, quién hizo la llamada y cuándo se realizó la llamada.

Los CloudTrail registros contienen información sobre las llamadas a la API acción de AWS TNB. También contienen información sobre las llamadas a la API acción de servicios como Amazon EC2 y AmazonEBS.

Registro de API llamadas de AWS Telco Network Builder mediante AWS CloudTrail

AWS Telco Network Builder está integrado con AWS CloudTrailun servicio que proporciona un registro de las acciones realizadas por un usuario, rol o un Servicio de AWS. CloudTrail captura todas las API llamadas AWS TNB como eventos. Las llamadas capturadas incluyen las llamadas desde la AWS TNB consola y las llamadas en código a las AWS TNB API operaciones. Con la información recopilada CloudTrail, puede determinar el destinatario de la solicitud AWS TNB, la dirección IP desde la que se realizó la solicitud, cuándo se realizó y detalles adicionales.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales del usuario raíz o del usuario.
- Si la solicitud se realizó en nombre de un usuario de IAM Identity Center.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro Servicio de AWS.

CloudTrail está activa en su cuenta Cuenta de AWS al crear la cuenta y automáticamente tiene acceso al historial de CloudTrail eventos. El historial de CloudTrail eventos proporciona un

registro visible, consultable, descargable e inmutable de los últimos 90 días de eventos de gestión registrados en un. Región de AWSPara obtener más información, consulte Uso del historial de CloudTrail eventos en la Guía del usuario.AWS CloudTrail La visualización del historial de eventos no conlleva ningún CloudTrail cargo.

Para tener un registro continuo de los eventos de Cuenta de AWS los últimos 90 días, crea un almacén de datos de eventos de senderos o CloudTraillagos.

CloudTrail senderos

Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. Todos los senderos creados con él AWS Management Console son multirregionales. Puede crear un registro de seguimiento de una sola región o de varias regiones mediante la AWS CLI. Se recomienda crear un sendero multirregional, ya que puedes capturar toda la actividad de tu Regiones de AWS cuenta. Si crea un registro de seguimiento de una sola región, solo podrá ver los eventos registrados en la Región de AWS del registro de seguimiento. Para obtener más información acerca de los registros de seguimiento, consulte Creación de un registro de seguimiento para su Cuenta de AWS y Creación de un registro de seguimiento para una organización en la Guía del usuario de AWS CloudTrail .

Puede enviar una copia de sus eventos de administración en curso a su bucket de Amazon S3 sin coste alguno CloudTrail mediante la creación de una ruta; sin embargo, hay cargos por almacenamiento en Amazon S3. Para obtener más información sobre CloudTrail los precios, consulte AWS CloudTrail Precios. Para obtener información acerca de los precios de Amazon S3, consulte Precios de Amazon S3.

CloudTrail Almacenes de datos de eventos en Lake

CloudTrail Lake le permite ejecutar consultas SQL basadas en sus eventos. CloudTrail Lake convierte los eventos existentes en JSON formato basado en filas al ORC formato Apache. ORCes un formato de almacenamiento en columnas que está optimizado para una rápida recuperación de datos. Los eventos se agregan en almacenes de datos de eventos, que son recopilaciones inmutables de eventos en función de criterios que se seleccionan aplicando selectores de eventos avanzados. Los selectores que se aplican a un almacén de datos de eventos controlan los eventos que perduran y están disponibles para la consulta. Para obtener más información sobre CloudTrail Lake, consulte Cómo trabajar con AWS CloudTrail Lake en la Guía del AWS CloudTrail usuario.

CloudTrail Los almacenes de datos y las consultas sobre eventos de Lake conllevan costes. Cuando crea un almacén de datos de eventos, elige la opción de precios que desea utilizar

para él. La opción de precios determina el costo de la incorporación y el almacenamiento de los eventos, así como el periodo de retención predeterminado y máximo del almacén de datos de eventos. Para obtener más información sobre CloudTrail los precios, consulte AWS CloudTrail Precios.

Ejemplos de eventos de AWS TNB

Un evento representa una solicitud única de cualquier fuente e incluye información sobre la API operación solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las API llamadas públicas, por lo que los eventos no aparecen en ningún orden específico.

El siguiente ejemplo muestra un CloudTrail evento que demuestra la `CreateSolFunctionPackage` operación.


```
{
    "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 obtener información sobre el contenido de los CloudTrail registros, consulte el contenido de los CloudTrail registros en la Guía del AWS CloudTrail usuario.

AWS TNBtareas de despliegue

Comprenda las tareas de implementación para supervisar las implementaciones de manera efectiva y tomar medidas más rápido.

En la siguiente tabla se enumeran las tareas AWS TNB de despliegue:

Nombre de la tarea para las implement aciones iniciadas antes del 7 de marzo de 2024	Nombre de la tarea para las implement aciones que se iniciaron a partir del 7 de marzo de 2024	Task description
AppInstallation	ClusterPluginInstall	Instala el complemento Multus en el clúster de AmazonEKS.

Nombre de la tarea para las implement aciones iniciadas antes del 7 de marzo de 2024	Nombre de la tarea para las implement aciones que se iniciaron a partir del 7 de marzo de 2024	Task description
AppUpdate	sin cambios de nombre	Actualiza las funciones de red que ya están instaladas en una instancia de red.
-	ClusterPluginUninstall	Desinstala los complementos en el clúster de AmazonEKS.
ClusterStorageClas sConfiguration	sin cambios de nombre	Configura la clase de almacenamiento (CSIcontrolador) en un EKS clúster de Amazon.
FunctionDeletion	sin cambios de nombre	Elimina las funciones de red de AWS TNB los recursos.
FunctionInstantiation	FunctionInstall	Despliega funciones de red mediante. HELM
FunctionUninstallation	FunctionUninstall	Desinstala la función de red de un clúster de AmazonEKS.
HookExecution	sin cambios de nombre	Ejecuta los enlaces del ciclo de vida tal como se define en. NSD
InfrastructureCanc ellation	sin cambios de nombre	Cancela un servicio de red.
InfrastructureInst antation	sin cambio de nombre	Aprovisiona AWS recursos en nombre del usuario.
InfrastructureTerm ination	sin cambios de nombre	Desaprovisiona AWS los recursos invocados mediante AWS TNB.
-	InfrastructureUpdate	Actualiza los AWS recursos aprovisionados en nombre del usuario.

Nombre de la tarea para las implement aciones iniciadas antes del 7 de marzo de 2024	Nombre de la tarea para las implement aciones que se iniciaron a partir del 7 de marzo de 2024	Task description
InventoryDeregistr ation	sin cambios de nombre	Anula el registro de los AWS recursos de. AWS TNB
-	InventoryRegistration	Registra los AWS recursos en. AWS TNB
KubernetesClusterC onfiguration	ClusterConfiguration	Configura el clúster de Kubernetes y añade funciones adicionales IAM a Amazon EKS AuthMap tal y como se define en. NSD
NetworkServiceFina lization	sin cambios de nombre	Finaliza el servicio de red y proporciona una actualización del estado de éxito o error.
NetworkServiceInst antiation	sin cambio de nombre	Inicia el servicio de red.
SelfManag edNodesConfiguration	sin cambio de nombre	Impulse los nodos autogestionados con el plano de control de Amazon EKS y Kubernetes.
-	ValidateNetworkSer viceUpdate	Ejecuta las validaciones antes de actualizar una instancia de red.

Cuotas de servicio para AWS TNB

Las cuotas de servicio, también denominadas límites, son la cantidad máxima de recursos u operaciones de servicio para su AWS cuenta. Para obtener más información, consulte el artículo sobre AWS Service Quotas en la Referencia general de Amazon Web Services.

A continuación se indican las cuotas de servicio para AWS TNB.

Nombre	Valor predeterminado	Ajusta	Descripción
Operaciones simultáneas y continuas de servicios de red	Cada región admitida: 40	Sí	Número máximo de operaciones simultáneas de servicios de red en curso en una región.
Paquetes de funciones	Cada región admitida: 200	Sí	El número máximo de paquetes de funciones en una región.
Paquetes de red	Cada región admitida: 40	Sí	El número máximo de paquetes de red en una región.
Instancias de servicios de red	Cada región admitida: 800	Sí	Número máximo de instancias de servicios de red en una región.

Historial de documentos de la guía AWS TNB del usuario

En la siguiente tabla se describen las versiones de la documentación de. AWS TNB

Cambio	Descripción	Fecha
Versión de Kubernetes para clúster	AWS TNBahora es compatible con la versión 1.30 de Kubernetes para crear clústeres de Amazon. EKS	19 de agosto de 2024
AWS TNBadmite una operación adicional para administrar el ciclo de vida de la red.	Puede actualizar una instancia de red instanciada o previamente actualizada con un nuevo paquete de red y valores de parámetros. Consulte: • Operaciones del ciclo de vida • Actualizar una instancia de red • AWS TNBejemplo de rol de servicio: • Agrega estas EKS acciones de Amazon: `eks:UpdateAddon` `eks:UpdateClusterVersion` ,`eks:UpdateNodegroupConfig` ,`eks:UpdateNodegroupVersion` , `eks:DescribeUpdate`	30 de julio de 2024

	• Agrega esta AWS CloudFormation acción: `cloudformation:UpdateStack` • Nuevas tareas de despliegue:`InfrastructureUpdate` ,`Inventory Registration` , `ValidateNetworkServiceUpdate` • APIactualizaciones: GetSolNetworkOperationListSolNetworkOperations, y UpdateSolNetworkInstance	
Nueva tarea y nuevos nombres de tareas para las tareas existentes	Hay una nueva tarea disponible. A partir del 7 de marzo de 2024, algunas tareas existentes tienen nuevos nombres para mayor claridad.	7 de mayo de 2024
Versión de Kubernetes para clúster	AWS TNBahora es compatible con la versión 1.29 de Kubernetes para crear clústeres de Amazon. EKS	10 de abril de 2024
Support para interfaz de red `security_groups`	Puede adjuntar grupos de seguridad al AWS.Networking.ENInodo.	2 de abril de 2024

Support para el cifrado de volúmenes EBS raíz de Amazon	Puede activar el EBS cifrado de Amazon para el volumen EBS raíz de Amazon. Para habilitarlo, añada las propiedades en el archivo AWS.Compute. EKSManagedNodeo AWS.Compute. EKSSelfManagedNodenodo.	2 de abril de 2024
Support for node `labels`	Puedes adjuntar etiquetas de nodo a tu grupo de nodos en el archivo AWS.Compute. EKSManagedNodeo AWS.Compute. EKSSelfManagedNodenodo.	19 de marzo de 2024
Support para interfaz de red `source_dest_check`	Puede indicar si desea habilitar o deshabilitar la comprobación de origen/destino de la interfaz de red a través de .Networking. AWS ENInodo.	25 de enero de 2024
Support para EC2 instancias de Amazon con datos de usuario personalizados	Puede lanzar EC2 instancias de Amazon con datos de usuario personalizados a través de AWS.Compute. UserData nodo.	16 de enero de 2024
Compatibilidad con grupo de seguridad	AWS TNBpermite importar el AWS recurso del grupo de seguridad.	8 de enero de 2024

Descripción actualizada de `network_interfaces`	Cuando la `network_interfaces` propiedad está incluida en el archivo AWS.Compute. EKSManagedNodeo AWS.Compute. EKSSelfManagedNodenode, AWS TNB obtiene el permiso correspondiente ENIs de la `multus_role` propiedad , si está disponible, o de la `node_role` propiedad.	18 de diciembre de 2023
Compatibilidad con clúster privado	AWS TNBahora es compatible con clústeres privados. Para indicar un clúster privado, establezca la propiedad `access` en `PRIVATE`.	11 de diciembre de 2023
Versión de Kubernetes para clúster	AWS TNBahora es compatible con la versión 1.28 de Kubernetes para crear clústeres de Amazon. EKS	11 de diciembre de 2023
AWS TNBadmite grupos de ubicación	Se agregó un grupo de ubicación para las definiciones de nodo `AWS.Compute.EKSManagedNode` y `AWS.Compute.EKSSelfManagedNode` .	11 de diciembre de 2023

AWS TNBañade soporte para IPv6	AWS TNBahora admite la creación de instancias de red con IPv6 infraestructura. Compruebe los nodos AWS.Networking.VPC, AWS.Networking.Subnet, .Networking.AWS InternetGateway, AWS.Redes. SecurityGroupIngressRule, AWS.Redes.SecurityGroupEgressRuley AWS.Compute. EKSpara IPv6 configuraciones. También hemos añadido los nodos AWS.Networking. NATGatewayy AWS.Networking.Route para la configuración. NAT64 Hemos actualizado el rol AWS TNB de servicio y el rol AWS TNB de servicio del grupo de EKS nodos de Amazon para obtener IPv6 permisos. Consulte Ejemplos de políticas de roles de servicio.	16 de noviembre de 2023
Se han añadido permisos a la política AWS TNB de roles de servicio	Añadimos permisos a la política de roles de AWS TNB servicio para Amazon S3 y AWS CloudFormation para habilitar la instanciación de la infraestructura.	23 de octubre de 2023

AWS TNBlanzado en más regiones	AWS TNBya está disponible en las regiones de Asia Pacífico (Seúl), Canadá (Centro), Europa (España), Europa (Estocolmo) y Sudamérica (São Paulo).	27 de septiembre de 2023
Etiquetas para AWS.Compute. EKSSelfManagedNode	AWS TNBahora admite etiquetas para la definición del `AWS.Compute.EKSSelfManagedNode` nodo.	22 de agosto de 2023
AWS TNBadmite instancias que aprovechan IMDSv2	Al lanzar la instancia, debe usarIMDSv2.	14 de agosto de 2023
Permisos actualizados para MultusRoleInlinePolicy	MultusRoleInlinePolicy Ahora incluye el `ec2:DeleteNetworkInterface` permiso.	7 de agosto de 2023
Versión de Kubernetes para clúster	AWS TNBahora es compatible con las versiones 1.27 de Kubernetes para crear clústeres de Amazon. EKS	25 de julio de 2023
AWS.Compute. EKS. AuthRole	AWS TNBadmite AuthRole que le permite añadir IAM roles al EKS clúster `aws-auth ConfigMap` de Amazon para que los usuarios puedan acceder al EKS clúster de Amazon mediante un IAM rol.	19 de julio de 2023

AWS TNBadmite grupos de seguridad.	Se agregó el archivo AWS.Networking. SecurityGroup, AWS.Networking. SecurityGroupEgressRuley AWS.Networking. SecurityGroupIngressRulea la NSD plantilla.	18 de julio de 2023
Versión de Kubernetes para clúster	AWS TNBadmite las versiones 1.22 a 1.26 de Kubernetes para crear clústeres de Amazon. EKS AWS TNBya no es compatible con las versiones 1.21 de Kubernetes.	11 de mayo de 2023
AWS.Compute. EKSSelfManagedNode	Puede crear nodos de trabajo autogestionados en la región, en las Zonas AWS Locales y. AWS Outposts	29 de marzo de 2023
Versión inicial	Esta es la primera versión de la Guía del AWS TNB usuario.	21 de febrero de 2023

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.