Guía del usuario

AWS Creador de redes de telecomunicaciones

AWS Creador de redes de telecomunicaciones: Guía del usuario

Table of Contents

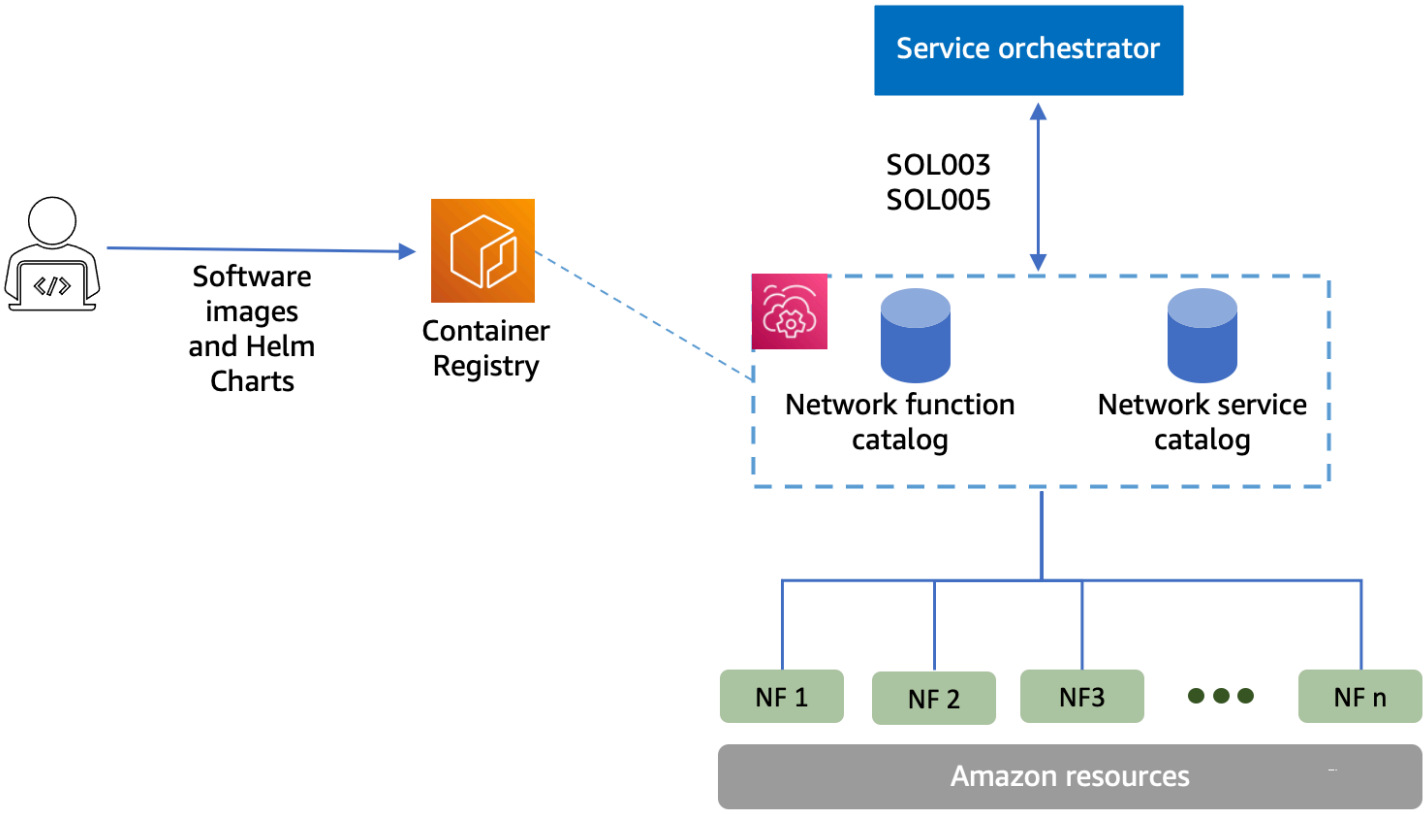
¿Qué es AWS Telco Network Builder?

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

Con AWS TNB, puede implementar redes 5G escalables y seguras en la Nube de AWS mediante el uso de imágenes de software en contenedores de forma automatizada. No es necesario aprender nuevas tecnologías, decidir qué servicio informático utilizar o saber cómo aprovisionar y configurar los recursos de AWS.

En su lugar, describe la infraestructura de su red y proporciona las imágenes de software de las funciones de la red de sus socios proveedores de software independientes (ISV). AWS TNB se integra con orquestadores de servicios de terceros y servicios AWS para aprovisionar automáticamente la infraestructura AWS necesaria, implementar funciones de red en contenedores y configurar la administración de redes y acceso para crear un servicio de red totalmente operativo.

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



Temas

¿Es la primera vez que utiliza AWS?

Si es nuevo en AWS productos y servicios, empiece por obtener más información con los siguientes recursos:

- Introducción a AWS
- Introducción a AWS

¿Para quién es AWS TNB?

AWS TNB es para los CSP que buscan aprovechar la rentabilidad, la agilidad y la elasticidad que Nube de AWS ofrece sin tener que escribir ni mantener guiones y configuraciones personalizados para diseñar, implementar y administrar los servicios de red. AWS TNB aprovisiona automáticamente la infraestructura AWS necesaria, implementa funciones de red en contenedores y configura la administración de redes y acceso para crear servicios de red totalmente operativos basados en los descriptores de servicios de red definidos por el CSP y las funciones de red que el CSP desea implementar.

¿Por qué usar AWS TNB?

Los siguientes son algunos de los motivos por los que un CSP desearía utilizar 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 TNB se integra con los populares orquestadores de servicios de terceros que cumplen normas ETSI.

Escalas

Puede configurar AWS TNB para escalar recursos AWS 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 TNB le permite inspeccionar y supervisar los recursos AWS que dan soporte a su red en un único panel, como Amazon VPC, Amazon EC2 y Amazon EKS.

Admite plantillas de servicio

AWS TNB le permite crear plantillas de servicio 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 una implementación de funciones de red, por ejemplo, al cambiar el tipo de instancia de un tipo de instancia de Amazon EC2, 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 utiliza AWS TNB para administrar el ciclo de vida de la función de la red, solo realiza los cambios en los descriptores de los servicios de red que describen la función de la red. AWS A continuación, TNB 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 aplicaciones RAN, Core, IMS y de red de la misma manera.

Acceso a AWS TNB

Puede crear, acceder y administrar los recursos de AWS TNB desde cualquiera de las siguientes interfaces:

- Consola AWS TNB: proporciona una interfaz web para administrar la red.
- API de AWS TNB: proporciona una API RESTful para realizar acciones de AWS TNB. Para obtener más información, consulte Referencia de la API de AWS.
- AWS Command Line Interface (AWS CLI): proporciona comandos para un amplio conjunto de servicios de AWS, incluido AWS TNB. Es compatible con Windows, macOS y Linux. Para obtener más información, consulte AWS Command Line Interface.
- SDK de AWS: proporciona API específicas de cada lenguaje 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 AWS SDKs.

Precios de AWS TNB

AWS TNB ayuda a los CSP a automatizar la implementación y administración de sus redes de telecomunicaciones en AWS. Al utilizar AWS TNB, paga las dos dimensiones siguientes:

- Por horas de elementos de función de red gestionados (MNFI).
- Por número de solicitudes de API.

También soportará cargos adicionales al utilizar otros servicios AWS junto con AWS TNB. Para obtener más información, consulte Precios de AWS TNB.

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 acerca de la facturación de las cuentas de AWS, consulte Facturación de cuentas de AWS.

Si tiene alguna pregunta sobre los eventos, las cuentas y la facturación de AWS, póngase 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 del entorno de AWS. Para obtener más información, consulte AWS Trusted Advisor.

Siguientes pasos

Para obtener información acerca de cómo comenzar a utilizar AWS TNB, consulte los siguientes temas:

- Configuración de AWS TNB: completar los pasos de requisitos previos.
- Introducción a 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 funciona AWS TNB

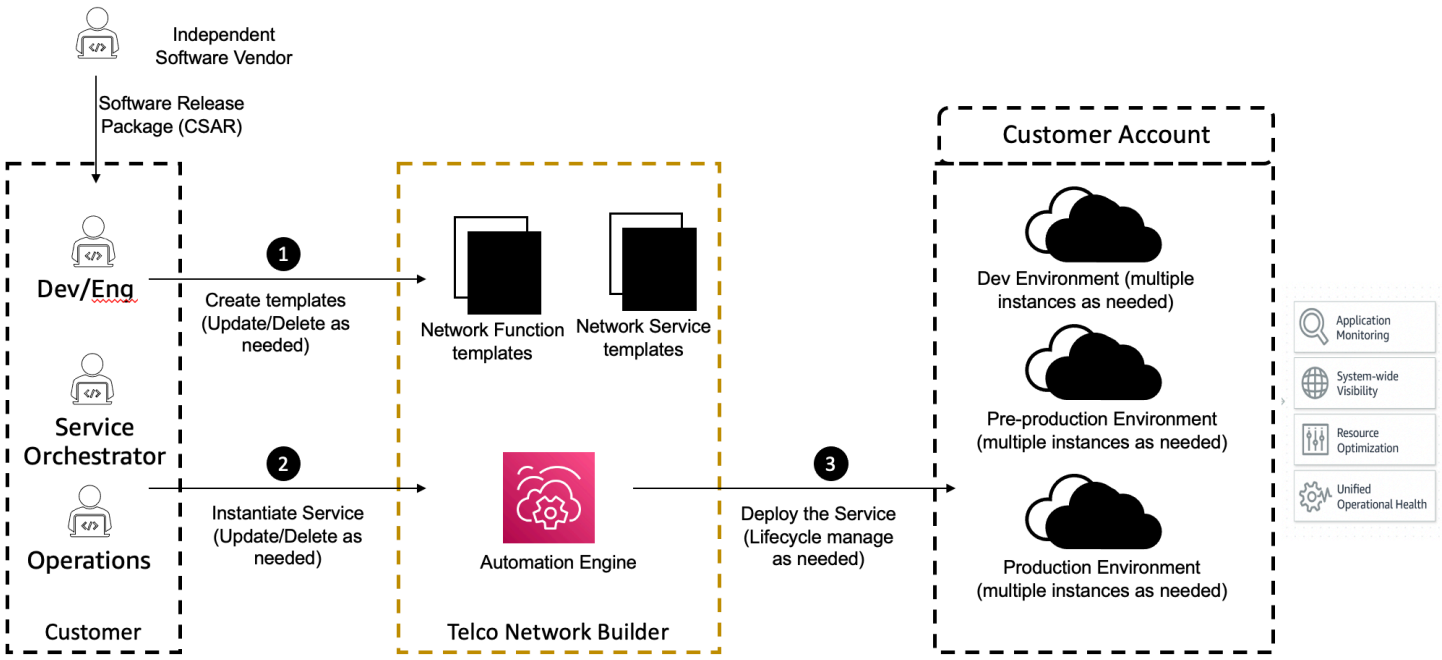
TNB AWS se integra con orquestadores y recursos AWS estandarizados de extremo a extremo para operar redes 5G completas.

AWS TNB le permite incorporar paquetes de funciones de red y descriptores de servicios de red (NSD) y le proporciona el motor de automatización necesario para operar sus redes. Puede usar su orquestador de extremo a extremo e integrarlo con las API de AWS TNB, o usar los SDK de TNB AWS para crear su propio flujo de automatización. Para obtener más información, consulte Arquitectura de AWS TNB.

Temas

Arquitectura de AWS TNB

AWS TNB le brinda la capacidad de realizar operaciones de administración del ciclo de vida a través de la AWS Management Console, AWS CLI, API de REST de AWS y los SDK. Esto permite a las distintas personas de CSP, como los miembros de los equipos de ingeniería, operaciones y sistemas programáticos, aprovechar las ventajas de AWS TNB. Puede crear y subir un paquete de funciones de red como un archivo de servicios en la nube (CSAR). El archivo CSAR 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 TNB le proporciona la visibilidad de sus implementaciones.



Integración con Servicios de AWS

Una red 5G se compone de un conjunto de funciones de red en contenedores interconectadas que se implementan en miles de clústeres de Kubernetes. AWS TNB se integra con los siguientes Servicios de AWS como API específicas de telecomunicaciones para crear un servicio de red totalmente operativo:

- Amazon Elastic Container Registry (Amazon ECR) para almacenar artefactos de funciones de red de proveedores de software independientes (ISV).
- Amazon Elastic Kubernetes Service (Amazon EKS) para configurar clústeres.
- Amazon VPC para constructos de redes.
- Grupos de seguridad con AWS CloudFormation.
- AWS CodePipeline para objetivos de implementación en todas las Regiones de AWS, Zonas Locales AWS y AWS Outposts.
- IAM para definir roles.
- AWS Organizations para controlar el acceso a las API de AWS TNB.
- AWS Health Dashboard y AWS CloudTrail para monitorear el estado y las métricas posteriores.

Cuotas de recursos de AWS TNB

Su Cuenta de AWS tiene cuotas predeterminadas, anteriormente conocidas como “límites”, para cada Servicio de AWS. A menos que se indique otra cosa, cada cuota es específica de una Región de AWS. Puede solicitar el aumento de algunas cuotas, pero no de todas.

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

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

La Cuenta de AWS incluye las siguientes cuotas en relación 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

Conceptos de AWS TNB

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

Contenido

Ciclo de vida de una función de red

AWS TNB le ayuda durante todo el 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 paquetes CSAR que vaya a implementar o actualizar.
4. Utilice AWS TNB para subir el paquete CSAR que define la función de su red (por ejemplo, CU, AMF y UPF) e intégrelo con una canalización de integración y entrega continuas (CI/CD) que pueda ayudarle a crear nuevas versiones del paquete CSAR a medida que estén disponibles nuevas imágenes de software de funciones de red o guiones 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 NSD 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. Cree una instancia (o aprovisione) la función de red tal como se define en su NSD y comience a transportar tráfico.
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.
- Sustituir o desactivar la función de red.

Utilizar interfaces estandarizadas

AWS TNB se integra con orquestadores de servicios compatibles con el Instituto Europeo de Normas de Telecomunicaciones (ETSI), lo que le permite simplificar la implementación de sus servicios de red. Los orquestadores de servicios pueden usar los SDK de AWS TNB, la CLI o las API para iniciar operaciones, como crear instancias o actualizar una función de red a una nueva versión.

AWS TNB admite las siguientes especificaciones.

Especificación	Release	Descripción
ETSI SOL001	v3.6.1	Define los estándares para permitir los descriptores de funciones de red basados en TOSCA.
ETSI SOL002	v3.6.1	Define modelos en torno a la gestión de funciones de red.
ETSI SOL003	v3.6.1	Define los estándares para la gestión del ciclo de vida de las funciones de red.
ETSI SOL004	v3.6.1	Define los estándares CSAR para los paquetes de funciones de red.

Especificación	Release	Descripción
ETSI SOL005	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.
ETSI SOL007	v3.5.1	Define los estándares para permitir los descriptores de servicios de red basados en TOSCA.

Paquetes de funciones de red para AWS TNB

Con AWS TNB, puede almacenar paquetes de funciones de red que cumplan la norma ETSI SOL001/SOL004 en un catálogo de funciones. A continuación, puede subir paquetes de Cloud Service Archive (CSAR) que contengan artefactos que describan el funcionamiento de su 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 (Amazon ECR) puede actuar como su repositorio de imágenes de funciones de red.
- Archivos adicionales: utilícelos para administrar la función de red; por ejemplo, guiones y gráficos de Helm.

Un paquete CSAR es un archivo zip que cumple la especificación YAML de la especificación de topología y orquestación para aplicaciones en la nube (TOSCA) de OASIS.

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 funciones de red para AWS TNB

TNB AWS almacena los descriptores de servicios de red (NSD) sobre las funciones de red que desea implementar y cómo desea implementarlas en el catálogo. Puede subir su archivo NSD de YAML, tal como lo describe ETSI SOL007, para incluir:

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

TNB AWS es compatible con los estándares de la ETSI para el modelado de recursos, como redes, servicios y funciones, en el lenguaje TOSCA. AWS TNB hace que sea más eficiente utilizar Servicios de AWS al modelarlos de una manera que su orquestador de servicios compatible con la ETSI pueda entender.

El siguiente es un fragmento de un NSD que muestra cómo modelar Servicios de AWS. La función de red se implementará en un clúster de Amazon EKS con la versión 1.27 de Kubernetes. Las subredes de las aplicaciones son Subnet01 y Subnet02. A continuación, puede definir los 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 de AWS TNB

Con AWS TNB, puede gestionar su red mediante operaciones de gestión estandarizadas de acuerdo con las normas ETSI SOL003 y SOL005. Puede utilizar las API de AWS TNB 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.
- Administrar las versiones de sus paquetes de funciones de red.
- Administrar versiones de sus NSD.
- 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 archivo `.yaml` de un paquete de red que utiliza el estándar TOSCA para describir las funciones de red que desea implementar y la infraestructura AWS en la que desea implementar las funciones de red. Para definir su NSD y configurar sus recursos subyacentes y las operaciones del ciclo de vida de la red, debe comprender el esquema TOSCA del NSD compatible con AWS TNB.

El archivo NSD está dividido en las partes siguientes:

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

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

2. VNFD: el NSD contiene la definición de la función de red en la que se deben 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 identificador debe coincidir con el identificador del paquete CSAR de la función de red.
   - Un nombre exclusivo para `namespace`. El nombre debe estar asociado a un identificador único para poder consultarlo más fácilmente en todo el archivo NSD YAML, 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 nodos de Amazon EKS, debe crearse dentro de un clúster de Amazon EKS. 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 de AWS TNB

Configure AWS TNB realizando las tareas descritas en este tema.

Tareas

- Inscripción en AWS
- Elegir una región de AWS
- Observe el punto de conexión de servicio
- (Opcional) Instale AWS CLI
- Creación de un usuario de IAM
- Configure los roles de AWS TNB

Inscripción en AWS

Al inscribirse en Amazon Web Services, su Cuenta de AWS se inscribe automáticamente en todos los servicios de AWS, incluido AWS TNB. Solo se le cobrará por los servicios que utilice.

Si ya dispone de una Cuenta de AWS, vaya a la siguiente tarea. Si no dispone de una Cuenta de AWS, utilice el siguiente procedimiento para crear una.

Para crear un Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones en línea.

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

   Al registrase en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tiene acceso a todos los recursos y Servicios de AWS de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario administrativo y utilice únicamente el usuario raíz para realizar la ejecución tareas que requieren acceso de usuario raíz.

Elegir una región de AWS

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

Observe el punto de conexión de servicio

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

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

Nombre de la región	Región	Punto de enlace	Protocolo	
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 AWS CLI

La AWS Command Line Interface (AWS CLI) ofrece comandos para un amplio conjunto de productos de AWS y es compatible con Windows, Mac y Linux. Puede obtener acceso a AWS TNB mediante la AWS CLI. Para empezar, consulte la Guía de usuario de AWS Command Line Interface. Para obtener más información sobre los comandos de AWS TNB, consulte tnb en Referencia de comandos de la AWS CLI.

Creación de un usuario de IAM

AWS Identity and Access Management (IAM) es un servicio web que lo ayuda a controlar de forma segura el acceso a los recursos de AWS. Cree un rol de usuario de IAM para usar credenciales a corto plazo para acceder a AWS.

Para crear el rol, siga las instrucciones en Introducción en la Guía del usuario de AWS IAM Identity Center.

También puede configurar el acceso programático mediante Configuración de la AWS CLI para usar AWS IAM Identity Center en la Guía del usuario de AWS Command Line Interface.

Configure los roles de AWS TNB

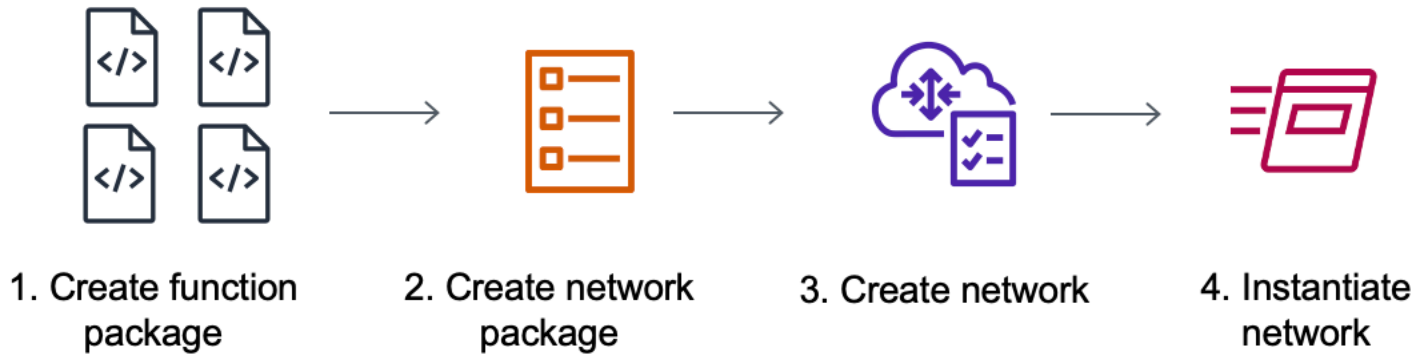
Debe crear un rol de servicio de IAM para administrar diferentes partes de su solución AWS TNB. AWS Los roles de servicio de TNB pueden realizar llamadas a la API a otros servicios AWS, como AWS CloudFormation, AWS CodeBuild y a varios servicios de cómputo y almacenamiento, en su nombre, para crear instancias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

Para obtener más información sobre el rol de servicio AWS TNB, consulte Gestión de identidad y acceso para TNB AWS.

Introducción a AWS TNB

En este tutorial se muestra cómo se utiliza AWS TNB para implementar una función de red, por ejemplo, la CU (unidad centralizada), la función de gestión 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
- Eliminar recursos

Requisitos previos

- Debe tener un plan AWS Business Support.
- Configure sus permisos mediante los roles de IAM.
- Un paquete NF que cumple la norma ETSI SOL001/SOL004.
- Plantillas NSD que cumplen la ETSI SOL007.

Cree un paquete de funciones

Para crear un paquete de funciones

1. Abra la consola de AWS TNB en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de funciones en el panel de navegación.
3. Elija Crear paquete de funciones.
4. Seleccione Elegir archivo y suba el paquete CSAR.
5. Elija Siguiente.
6. Revise los detalles del paquete y, a continuación, seleccione Crear paquete de funciones.

Crear un paquete de red

Para crear un paquete de red

1. Seleccione Paquetes de red en el panel de navegación.
2. Seleccione Crear paquete de red.
3. Seleccione Elegir archivo y suba el NSD.
4. Elija Siguiente.
5. Seleccione Crear paquete de red.

Crear e instanciar una instancia de red

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. Seleccione el NSD. Verifique los detalles y, a continuación, seleccione Siguiente.
5. Elija Crear instancia de red. El estado inicial es `Created`.
6. Elija el identificador de la instancia de red y, a continuación, elija Instanciar.
7. Elija Instanciar red.
8. Actualice para realizar un seguimiento del estado de la instancia de red.

Eliminar recursos

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. Actualice 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 estándar de la ETSI) y un descriptor de paquete de funciones que utiliza el estándar TOSCA para describir cómo deben ejecutarse las funciones en su red.

Tareas

- Cree un paquete de funciones en AWS TNB
- Ver un paquete de funciones en AWS TNB
- Descargue un paquete de funciones de AWS TNB
- Eliminar 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 de funciones de red de AWS TNB. Crear un paquete de funciones es el primer paso para crear una red en TNB. Una vez que haya cargado un paquete de funciones, tendrá que crear un paquete de red.

Console

Cree un paquete de funciones mediante la consola

1. Abra la consola de AWS TNB en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de funciones en el panel de navegación.
3. Elija Crear paquete de funciones.
4. Elija Elegir archivo y suba el paquete CSAR de su NF.
5. Elija Siguiente.
6. Revise los detalles del paquete.
7. Elija Crear paquete de funciones.

AWS CLI

Para crear un paquete de funciones mediante la AWS CLI

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

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

2. Utilice el comando put-sol-function-package-content para subir 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
```

Ver 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 consola de AWS TNB 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 la AWS CLI

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

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

2. Utilice el comando get-sol-function-package 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 red de AWS TNB.

Console

Para descargar un paquete de funciones mediante la consola

1. Abra la consola de AWS TNB 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 la AWS CLI

Utilice el comando get-sol-function-package-content para descargar el contenido del 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
```

Eliminar un paquete de funciones de AWS TNB

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

Console

Para eliminar un paquete de funciones mediante la consola

1. Abra la consola de AWS TNB 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 la AWS CLI

1. Utilice el comando update-sol-function-package para desactivar un paquete de funciones.

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

2. Utilice el comando delete-sol-function-package 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 infraestructura AWS en la que desea implementarlos.

Tareas

- Crear un paquete de red en AWS TNB
- Ver un paquete de red en AWS TNB
- Descargue un paquete de red de AWS TNB
- Elimine un paquete de red de AWS TNB

Crear 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 guiones 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 VPC, subredes o clústeres de Amazon EKS.

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 consola de AWS TNB en https://console.aws.amazon.com/tnb/.
2. Seleccione Paquetes de red en el panel de navegación.
3. Seleccione Crear paquete de red.
4. Seleccione Elegir archivo y suba el paquete CSAR.
5. Elija Siguiente.
6. Revise los detalles del paquete.
7. Seleccione Crear paquete de red.

AWS CLI

Para crear un paquete de red con la AWS CLI

1. Use el comando create-sol-network-package para crear un paquete de red.

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

2. Utilice el comando put-sol-network-package-content para subir 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
```

Ver 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 consola de AWS TNB 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 con la AWS CLI

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

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

2. Use el comando get-sol-network-package 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 red de AWS TNB.

Console

Para descargar un paquete de red con la consola

1. Abra la consola de AWS TNB 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 con la AWS CLI

- Utilice el comando get-sol-network-package-content para descargar el contenido del 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 red de AWS TNB. Para eliminar un paquete de red, el paquete debe estar desactivado.

Console

Para eliminar un paquete de red con la consola

1. Abra la consola de AWS TNB 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 con la AWS CLI

1. Use el comando update-sol-network-package 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. Use el comando delete-sol-network-package 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 AWS TNB que se puede implementar.

Tareas

- Cree una instancia de red mediante AWS TNB
- Ver una instancia de red en AWS TNB
- Actualizar una instancia de red en AWS TNB
- Finalice y elimine una instancia de red de AWS TNB

Cree una instancia de red mediante AWS TNB

Una instancia de red se crea después de crear un paquete de red. Una vez que haya creado una instancia de red, debe instanciarla. Al crear una instancia de red, AWS TNB implementa las funciones de red según las especificaciones del descriptor del servicio de red.

Console

Para crear e instanciar una instancia de red con la consola

1. Abra la consola de AWS TNB 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 NSD. Verifique los datos y, a continuación, seleccione Siguiente.
6. Elija Crear instancia de red.
7. Elija Instanciar.
8. Elija Instanciar red.
9. Actualice para realizar un seguimiento del estado de la instancia de red.

AWS CLI

Para crear e instanciar una instancia de red con la AWS CLI

1. Use el comando create-sol-network-instance 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"
```

2. Use el comando instantiate-sol-network-instance para instanciar la instancia de red.

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

Ver 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 consola de AWS TNB 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 con la AWS CLI

1. Use el comando list-sol-network-instances para enumerar las instancias de red.

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

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

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

Actualizar una instancia de red en AWS TNB

Obtenga información sobre cómo actualizar una instancia de red.

Console

Para actualizar una instancia de red con la consola

1. Abra la consola de AWS TNB 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. En la pestaña de funciones, seleccione la instancia de función que desee actualizar.
5. Elija Actualizar.
6. Introduzca las anulaciones de actualización para confirmar la actualización.
7. Elija Actualizar.
8. Actualice para realizar un seguimiento del estado de la instancia de red.

AWS CLI

Utilice la CLI para actualizar una instancia de red

Use el comando update-sol-network-instance para actualizar 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 ...
```

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 consola de AWS TNB 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 con la AWS CLI

1. Use el comando terminate-sol-network-instance para finalizar una instancia de red.

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

2. (Opcional) Use el comando delete-sol-network-instance 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 de red
- Cancelación de una operación de red

Ver una operación 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 consola de AWS TNB 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, elija la operación de red.

AWS CLI

Para ver una operación de red con la AWS CLI

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

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

2. Use el comando get-sol-network-operation 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}$
```

Cancelación de una operación 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 consola de AWS TNB 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 con la AWS CLI

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

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

Referencia TOSCA para AWS TNB

La especificación de topología y orquestación para aplicaciones en la nube (TOSCA) es una sintaxis declarativa que los CSP utilizan para describir una topología de servicios web basados en la nube, sus componentes, relaciones y los procesos que los gestionan. Los CSP describen 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. A continuación, los CSP suben la plantilla en AWS TNB, que sintetiza los recursos necesarios para establecer una red 5G que funcione en todas las zonas de disponibilidad de AWS.

Contenido

plantilla VNFD

Define una plantilla de descriptor de funciones 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 nodos TOSCA AWS. Los posibles nodos son:

- AWS.VNF
- AWS.Artifacts.Helm

AWS.VNF

Define un nodo de función de red virtual AWS (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: String

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 del VNFD.

Obligatorio: sí

Tipo: String

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

`descriptor_name`

El nombre del descriptor.

Obligatorio: sí

Tipo: String

`provider`

El autor del VNFD.

Obligatorio: sí

Tipo: String

Requisitos

`helm`

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

Obligatorio: sí

Tipo: String

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 Helm AWS.

Sintaxis

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

Propiedades

`implementation`

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

Obligatorio: sí

Tipo: String

Ejemplo

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

Plantilla de NSD

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

Cuando desee transferir dinámicamente un parámetro, como el bloque de CIDR para el nodo de VPC, puede usar la sintaxis `{ get_input: input-parameter-name }` y definir los parámetros en la plantilla de NSD. A continuación, reutilice el parámetro en la misma plantilla de NSD.

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 }
```

Importación de VNFD

`descriptor_id`

El UUID del descriptor.

Obligatorio: sí

Tipo: String

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 posibles AWS nodos TOSCA son:

- AWS.NS
- AWS.Compute.EKS
- AWS.compute.EKS. AuthRole
- AWS.compute.EKS ManagedNode
- AWS.compute.eks SelfManagedNode
- AWS.Compute. PlacementGroup
- AWS.Computar. UserData
- AWS.Redes. SecurityGroup
- AWS.Redes. SecurityGroupEgressRule
- AWS.Redes. SecurityGroupIngressRule
- AWS.Resource.Import
- AWS.Networking.ENI
- AWS.HookExecution

- AWS.Redes. InternetGateway
- AWS.Redes. RouteTable
- AWS.Networking.Subnet
- AWS.Deployment.VNFDeployment
- AWS.Networking.VPC
- AWS.Networking.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: String

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 del NSD.

Obligatorio: sí

Tipo: String

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 deseada y una función que permita al plano de control de Kubernetes gestionar los AWS recursos necesarios para sus NF. Los complementos de la interfaz de red de contenedores (CNI) de Multus están habilitados. 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 (CNI) de Multus.

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 de la interfaz de almacenamiento de contenedores (CSI) de Amazon EBS instalado en el clúster de Amazon EKS.

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

`enabled`

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

Obligatorio: no

Tipo: booleano

`version`

La versión del complemento de controlador de CSI de Amazon EBS. La versión debe coincidir con una de las versiones devueltas por la DescribeAddonVersionsacción. Para obtener más información, consulte la referencia DescribeAddonVersionsde la API de Amazon EKS

Obligatorio: no

Tipo: String

Propiedades

`version`

La versión de Kubernetes para el clúster. AWS Telco Network Builder es compatible con las versiones 1.23 a 1.29 de Kubernetes.

Obligatorio: sí

Tipo: cadena

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

`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: String

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.Compute.eks. AuthRole

An AuthRole le permite añadir funciones de IAM al clúster de Amazon EKS para que los usuarios puedan acceder al clúster `aws-auth ConfigMap` de Amazon EKS mediante una función de IAM.

Sintaxis

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

Propiedades

`role_mappings`

Lista de asignaciones que definen los roles de IAM que deben añadirse al clúster de Amazon EKS `aws-auth ConfigMap`.

`arn`

El ARN del rol de IAM.

Obligatorio: sí

Tipo: cadena

`groups`

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

Obligatorio: no

Tipo: lista

Requisitos

`clusters`

Un nodo AWS.Compute.EKS.

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.eks ManagedNode

AWS TNB es compatible con los grupos de nodos gestionados por EKS para automatizar el aprovisionamiento y la administración del ciclo de vida de los nodos (instancias de Amazon EC2) para los clústeres de Amazon EKS Kubernetes. Para crear un grupo de nodos EKS, debe elegir Imágenes de máquina de Amazon (AMI) para los nodos de trabajo del clúster proporcionando el identificador de la AMI o el tipo de AMI. También proporciona un par de claves de Amazon EC2 para el acceso SSH y las propiedades de escalado de su grupo de nodos. El grupo de nodos debe estar asociado a un clúster de EKS. Debe proporcionar las subredes de los nodos de trabajo.

Si lo desea, puede adjuntar 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 de cálculo del grupo de nodos gestionado por Amazon EKS, como los tipos de instancias de Amazon EC2 y las AMI de instancias de Amazon EC2.

`ami_type`

El tipo de AMI compatible con Amazon EKS.

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`

Es el ID de la AMI.

Obligatorio: no

Tipo: String

> **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 claves EC2 para habilitar el acceso SSH.

Obligatorio: sí

Tipo: cadena

`root_volume_encryption`

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

Obligatorio: no

Predeterminado: true

Tipo: Booleano

`root_volume_encryption_key_arn`

El ARN de la AWS KMS clave. AWS TNB admite el ARN de clave normal, el ARN de clave multirregional y el ARN de alias.

Obligatorio: no

Tipo: String

> **Note**
>
> - Si `root_volume_encryption` es falso, no lo incluya. `root_volume_encryption_key_arn`
> - AWS TNB admite el cifrado del volumen raíz de las AMI respaldadas por Amazon EBS.
> - Si el volumen raíz de la AMI ya está cifrado, debe incluirlo `root_volume_encryption_key_arn` para que AWS TNB vuelva a cifrar el volumen raíz.
> - Si el volumen raíz de la AMI no está cifrado, AWS TNB lo utiliza `root_volume_encryption_key_arn` para cifrar el volumen raíz.
>
>   Si no la incluye`root_volume_encryption_key_arn`, AWS TNB utilizará la clave predeterminada proporcionada por AWS Key Management Service para cifrar el volumen raíz.
> - AWS TNB no descifra una AMI cifrada.

`scaling`

Propiedades que definen los parámetros de escalado del grupo de nodos gestionado por Amazon EKS, como el número deseado de instancias de Amazon EC2 y el número mínimo y máximo de instancias de Amazon EC2 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 del rol de IAM asociado a la instancia de Amazon EC2.

Obligatorio: sí

Tipo: cadena

`tags`

Las etiquetas que deben asociarse al recurso.

Obligatorio: no

Tipo: lista

Requisitos

`cluster`

Un nodo AWS.Compute.EKS.

Obligatorio: sí

Tipo: cadena

`subnets`

Un nodo AWS.Networking.Subnet.

Obligatorio: sí

Tipo: lista

`network_interfaces`

Un nodo AWS.Networking.ENI. 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.

Al configurarlo`network_interfaces`, AWS TNB obtiene el permiso relacionado con los ENI de la `multus_role` propiedad si la incluyó en el nodo `multus` AWS.Compute.EKS. De lo contrario, AWS TNB obtiene el permiso relacionado con las ENI 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: String

`user_data`

Un tosca.nodes.AWS.Compute. UserDatareferencia de nodo. Un script de datos de usuario se pasa a las instancias de Amazon EC2 lanzadas por el grupo de nodos administrados. Agregue los

permisos necesarios para ejecutar datos de usuario personalizados al node_role pasado al grupo de nodos.

Obligatorio: no

Tipo: String

`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 una longitud máxima de 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.compute.eks SelfManagedNode

AWS TNB es compatible con los nodos autogestionados de Amazon EKS para automatizar el aprovisionamiento y la administración del ciclo de vida de los nodos (instancias de Amazon EC2) para los clústeres de Kubernetes de Amazon EKS. Para crear un grupo de nodos de Amazon EKS, debe elegir las Imágenes de máquina de Amazon (AMI) para los nodos de trabajo del clúster proporcionando el identificador de la AMI o el tipo de AMI. Si lo desea, proporcione un par de claves Amazon EC2 para el acceso a SSH. También debe proporcionar el tipo de instancia y los tamaños deseado, mínimo y máximo. El grupo de nodos debe estar asociado a un clúster de Amazon EKS. Debe proporcionar las subredes de los nodos de trabajo.

Si lo desea, puede adjuntar 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 de cálculo de los nodos autogestionados de Amazon EKS, como los tipos de instancias de Amazon EC2 y las AMI de instancias de Amazon EC2.

`ami_id`

El ID de AMI utilizado para lanzar la instancia. AWS TNB admite instancias que utilizan IMDSv2. Para obtener más información, consulte Versión IMDS.

Obligatorio: sí

Tipo: cadena

`instance_type`

El tamaño de la instancia.

Obligatorio: sí

Tipo: cadena

`key_pair`

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

Obligatorio: sí

Tipo: cadena

`root_volume_encryption`

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

Obligatorio: no

Predeterminado: true

Tipo: Booleano

`root_volume_encryption_key_arn`

El ARN de la AWS KMS clave. AWS TNB admite el ARN de clave normal, el ARN de clave multirregional y el ARN de alias.

Obligatorio: no

Tipo: String

> **Note**
>
> - Si `root_volume_encryption` es falso, no lo incluya. `root_volume_encryption_key_arn`
> - AWS TNB admite el cifrado del volumen raíz de las AMI respaldadas por Amazon EBS.
> - Si el volumen raíz de la AMI ya está cifrado, debe incluirlo `root_volume_encryption_key_arn` para que AWS TNB vuelva a cifrar el volumen raíz.
> - Si el volumen raíz de la AMI no está cifrado, AWS TNB lo utiliza `root_volume_encryption_key_arn` para cifrar el volumen raíz.
>
>   Si no lo incluye`root_volume_encryption_key_arn`, AWS TNB lo utiliza AWS Managed Services para cifrar el volumen raíz.

- AWS TNB no descifra una AMI cifrada.

scaling

Propiedades que definen los parámetros de escalado de los nodos autogestionados de Amazon EKS, como el número deseado de instancias de Amazon EC2 y el número mínimo y máximo de instancias de Amazon EC2 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 del rol de IAM asociado a la instancia de Amazon EC2.

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 nodo AWS.Compute.EKS.

Obligatorio: sí

Tipo: cadena

`subnets`

Un nodo AWS.Networking.Subnet.

Obligatorio: sí

Tipo: lista

`network_interfaces`

Un nodo AWS.Networking.ENI. 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.

Al configurarlo`network_interfaces`, AWS TNB obtiene el permiso relacionado con los ENI de la `multus_role` propiedad si la incluyó en el nodo `multus` AWS.Compute.EKS. De lo contrario, AWS TNB obtiene el permiso relacionado con las ENI 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: String

`user_data`

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

Obligatorio: no

Tipo: String

`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 una longitud máxima de 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 las instancias de Amazon EC2.

Cuando se lanza una nueva instancia de Amazon EC2, el servicio de Amazon EC2 intenta colocar la instancia de forma que todas las instancias se distribuyan en el hardware subyacente para minimizar los errores 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 instancias de Amazon EC2.

Obligatorio: sí

Tipo: cadena

Valores posibles: CLUSTER | PARTITION | SPREAD_HOST | SPREAD_RACK

- CLUSTER: agrupa las instancias unas cerca de otras 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 que es típica de las aplicaciones de computación de alto rendimiento (HPC).
- PARTITION: distribuye las instancias entre las particiones lógicas de modo que los grupos de instancias de una partición no compartan el hardware subyacente con los grupos de instancias de las demás particiones. Esta estrategia suelen utilizarla grandes cargas de trabajo distribuidas y replicadas, como Hadoop, Cassandra y Kafka.
- SPREAD_RACK: coloca estrictamente un pequeño grupo de instancias en distintos equipos de hardware subyacentes para reducir los fallos correlacionados.
- SPREAD_HOST: solo puede usar con 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.Computación. UserData

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

Durante la instanciación de la red, AWS TNB proporciona el registro de la instancia de Amazon EC2 al clúster mediante un script de datos de usuario. Cuando también se proporcionan datos de usuario personalizados, AWS TNB combina 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 registro de Amazon EKS.

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 TNB admite scripts de datos de usuario de hasta 7 KB de tamaño.

- Como AWS TNB procesa y renderiza el script de `multimime` datos de usuario, asegúrese de que el script cumpla con todas las reglas. AWS CloudFormation 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 TNB admite grupos de seguridad para automatizar el aprovisionamiento de los grupos de seguridad de Amazon EC2 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 nodo AWS.Networking.VPC.

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 TNB admite reglas de salida de grupos de seguridad para automatizar el aprovisionamiento de las reglas de salida de grupos de seguridad de Amazon EC2 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 intervalo de direcciones IPv4 en formato CIDR. Debe especificar un rango de CIDR que permita el tráfico de salida.

Obligatorio: no

Tipo: String

`cidr_ipv6`

El intervalo de direcciones IPv6, en formato CIDR, para tráfico de salida. Debe especificar un grupo de seguridad de destino (`destination_security_group` o `destination_prefix_list`) o un rango de CIDR (`cidr_ip` o `cidr_ipv6`).

Obligatorio: no

Tipo: String

`description`

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

Obligatorio: no

Tipo: String

`destination_prefix_list`

El identificador de lista de prefijos de una lista de prefijos gestionada por Amazon VPC existente. Este es el destino de las instancias del grupo de nodos asociado al grupo de seguridad. Para obtener más información sobre listas de prefijos administrados, consulte la Lista de prefijos administrados en la Guía del usuario de Amazon VPC.

Obligatorio: no

Tipo: String

`from_port`

Si el protocolo es TCP o UDP, es el inicio del rango de puertos. Si el protocolo es ICMP o ICMPv6, este es el número de tipo. Un valor -1 indica que todos son tipos ICMP/ICMPv6. Si especifica que todos son tipos ICMP/ICMPv6, debe también especificar los códigos de ICMP/ICMPv6.

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 o UDP, es el final del rango de puertos. Si el protocolo es ICMP o ICMPv6, este es el código. Un valor -1 indica que son todos códigos ICMP/ICMPv6. Si especifica que todos son tipos ICMP/ICMPv6, debe también especificar los códigos de ICMP/ICMPv6.

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

Obligatorio: no

Tipo: String

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 TNB admite las reglas de entrada de grupos de seguridad para automatizar el aprovisionamiento de las reglas de entrada de grupos de seguridad de Amazon EC2 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 intervalo de direcciones IPv4 en formato CIDR. Debe especificar un rango de CIDR que permita el tráfico de entrada.

Obligatorio: no

Tipo: String

`cidr_ipv6`

El intervalo de direcciones IPv6, en formato CIDR, para tráfico de entrada. Debe especificar un grupo de seguridad de origen (`source_security_group` o `source_prefix_list`) o un rango de CIDR (`cidr_ip` o `cidr_ipv6`).

Obligatorio: no

Tipo: String

`description`

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

Obligatorio: no

Tipo: String

`source_prefix_list`

El identificador de lista de prefijos de una lista de prefijos gestionada por Amazon VPC 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 listas de prefijos administrados, consulte la Lista de prefijos administrados en la Guía del usuario de Amazon VPC.

Obligatorio: no

Tipo: String

`from_port`

Si el protocolo es TCP o UDP, es el inicio del rango de puertos. Si el protocolo es ICMP o ICMPv6, este es el número de tipo. Un valor -1 indica que todos son tipos ICMP/ICMPv6. Si especifica que todos son tipos ICMP/ICMPv6, debe también especificar los códigos de ICMP/ICMPv6.

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 o UDP, es el final del rango de puertos. Si el protocolo es ICMP o ICMPv6, este es el código. Un valor -1 indica que son todos códigos ICMP/ICMPv6. Si especifica que todos son tipos ICMP/ICMPv6, debe también especificar los códigos de ICMP/ICMPv6.

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

Obligatorio: no

Tipo: String

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 que se importa a AWS TNB.

Obligatorio: no

Tipo: lista

`resource_id`

El identificador del recurso que se importa a 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.Networking.ENI

Una interfaz de red es un componente de red lógico en una 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 instancia de Amazon EC2 en una subred, puede adjuntarle una interfaz de red o desconectar una interfaz de red de esa instancia de Amazon EC2 y volver a conectarla a otra 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: String

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: String

Valores posibles: `CODE_BUILD`

Requisitos

`definition`

Un AWS. HookDefinitionNodo .Bash.

Obligatorio: sí

Tipo: cadena

`vpc`

Un nodo AWS.Networking.VPC.

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 de la VPC. Debe incluir la propiedad `dest_cidr` o `ipv6_dest_cidr`.

`dest_cidr`

El bloque de CIDR IPv4 utilizado para la coincidencia del destino. 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 bloque de CIDR IPv6 utilizado para la coincidencia del 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 específica de IPv6. 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`

Un nodo AWS.Networking.VPC.

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

Las tablas de enrutamiento contienen conjuntos de reglas, denominadas rutas, que determinan hacia dónde se dirige el tráfico de red desde las subredes de la VPC o puerta de enlace. Debe asociar una tabla de enrutamiento a una 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 nodo AWS.Networking.VPC.

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 en su VPC, y debe residir completamente en una zona de disponibilidad. Debe especificar una VPC, un bloque de CIDR, 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 IPv4 pública.

Obligatorio: sí

Tipo: cadena

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

`availability_zone`

La zona de disponibilidad de la subred. Este campo admite zonas de AWS disponibilidad dentro de una AWS región, por ejemplo `us-west-2` (US West (Oregon)). 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 bloque de CIDR de la subred.

Obligatorio: no

Tipo: String

`ipv6_cidr_block`

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

Obligatorio: no

Tipo: String

`ipv6_cidr_block_suffix`

El sufijo hexadecimal de 2 dígitos del bloque de CIDR IPv6 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: String

`outpost_arn`

El ARN en el AWS Outposts que se creará la subred. Añada esta propiedad a la plantilla de NSD si desea lanzar nodos autogestionados de Amazon EKS en AWS Outposts. Para obtener más información, consulte Amazon EKS en AWS Outposts en la Guía del usuario de Amazon EKS.

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

Obligatorio: no

Tipo: String

`tags`

Las etiquetas que deben asociarse al recurso.

Obligatorio: no

Tipo: lista

Requisitos

`vpc`

Un nodo AWS.Networking.VPC.

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.Deployment.VNFDeployment

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

Sintaxis

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

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

Requisitos

`deployment`

Un nodo AWS.Deployment.VNFDeployment.

Obligatorio: no

Tipo: String

`cluster`

Un nodo AWS.Compute.EKS.

Obligatorio: sí

Tipo: cadena

`vnfs`

Un nodo AWS.VNF.

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: String

`post_create`

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

Obligatorio: no

Tipo: String

Ejemplo

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

AWS.Networking.VPC

Debe especificar un bloque de CIDR 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 red IPv4 para la VPC en la notación de CIDR.

Obligatorio: sí

Tipo: cadena

`ipv6_cidr_block`

El bloque de IPv6 CIDR que se utiliza para crear la VPC.

Valor permitido: `AMAZON_PROVIDED`

Obligatorio: no

Tipo: String

`dns_support`

Especifica si las instancias lanzadas en la VPC obtienen nombres de host de DNS.

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.Networking.NATGateway

Puede definir un nodo de puerta de enlace de NAT 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: String

`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 puerta de enlace NAT 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 rutas IPv4 de destino al recurso de destino.

Obligatorio: sí

Tipo: lista

Tipo de miembro: cadena

Propiedades

`nat_gateway`

La referencia del nodo AWS.Networking.NATGateway

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 que se vayan a utilizar en NSD y VNFD.

- AWS.HookDefinition.bash

AWS.HookDefinition.Bash

Define una AWS HookDefinition en `bash`.

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: String

`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 reservadas de AWS TNB (no distingue entre mayúsculas y minúsculas):
  - CODEBUILD
  - TNB
  - INICIO
  - 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: String

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"
```

La seguridad en AWS Telco Network Builder

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 al utilizar AWS TNB. Los siguientes temas muestran cómo configurar 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 recursos de AWS TNB.

Contenido

Protección de datos en TNB AWS

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 las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS.

Con fines de protección de 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 autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- 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 módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

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 u otro tipo Servicios de AWS mediante la consola, la API o los SDK. AWS CLI AWS 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, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Gestión de datos

Cuando cierras 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 restaurará tus datos. Transcurridos 120 días, AWS TNB borra tus datos de forma permanente. AWS TNB también cierra sus redes y elimina sus paquetes de funciones y sus paquetes de red.

Cifrado en reposo

AWS TNB siempre cifra todos los datos almacenados en el servicio en reposo sin requerir ninguna configuración adicional. Este cifrado es totalmente automático. AWS Key Management Service

Cifrado en tránsito

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

Gestión de identidad y acceso para TNB AWS

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

Contenido

- Administración de acceso mediante políticas
- Cómo funciona AWS Telco Network Builder 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 que realices en AWS TNB.

Usuario del servicio: si utiliza el servicio AWS TNB para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más funciones de AWS TNB 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 característica en AWS TNB, consulte Solución de problemas de AWS identidad y acceso a Telco Network Builder.

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

Administrador de IAM: si es administrador de IAM, puede que desee obtener más información sobre cómo redactar políticas para administrar el acceso a TNB. AWS Para ver ejemplos de políticas basadas en la identidad de AWS TNB que puede utilizar en IAM, 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 con sus AWS credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. 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 (Centro de identidades

de IAM), 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, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

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 proporciona 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 Firmar las solicitudes de la AWS API en la Guía del usuario de IAM.

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 multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWSen la Guía del usuario de IAM.

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 las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

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 de su empresa, 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 Single Sign-On. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. 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 de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

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 permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada.

Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM 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 acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS Single Sign-On.
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común 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 usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS 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 sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al 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 administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar una AWS función a una instancia EC2 y ponerla 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 a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

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 documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON 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 permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen 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 la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de 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 roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM 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 puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de 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 característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). 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 Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita 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 acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations.
- 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 más información, consulte Políticas de sesión en la Guía del usuario de IAM.

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 AWS determinar 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 usuario de IAM.

Cómo funciona AWS Telco Network Builder con IAM

Antes de usar IAM para administrar el acceso a AWS TNB, conozca qué funciones de IAM están disponibles para su uso con TNB. AWS

Funciones de IAM que puede utilizar con Telco Network Builder AWS

Característica de IAM	AWS Soporte TNB
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de políticas	Sí
ACL	No
ABAC (etiquetas en 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 funcionan el AWS TNB y otros AWS servicios con la mayoría de las funciones de IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas de TNB basadas en la identidad AWS

Compatibilidad con las políticas basadas en identidades	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, 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 más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, 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 más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para TNB AWS

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

Políticas basadas en los recursos dentro de TNB AWS

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM 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 toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de 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 administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso 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 más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones políticas para TNB AWS

Admite acciones de política	Sí

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

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. 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 las acciones de AWS TNB, consulte las acciones definidas por AWS Telco Network Builder en la Referencia de autorización de servicios.

Las acciones políticas en 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 de 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 TNB AWS

Admite recursos de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON 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` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el 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 recursos de AWS TNB y sus ARN, consulte los recursos definidos por AWS Telco Network Builder en la Referencia de autorización de servicios. Para saber con qué

acciones puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS Telco Network Builder.

Para ver ejemplos de políticas de 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 TNB AWS

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden usar las políticas de AWS JSON 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 un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

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 usuario de IAM.

Para ver una lista de las claves de condición AWS TNB, consulte las claves de condición de AWS Telco Network Builder en la Referencia de autorización de servicios. 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 de AWS TNB basadas en la identidad, consulte. Ejemplos de políticas basadas en identidad para Creador de redes de telecomunicaciones de AWS

ACL en TNB AWS

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con TNB AWS

Admite ABAC (etiquetas en las políticas)	Sí

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

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta 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 información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con TNB AWS

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante 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 más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS 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 más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales entre servicios para TNB AWS

Admite Forward access sessions (FAS)	Sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en él AWS, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS 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 sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio de AWS TNB

Compatible con funciones de servicio	No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

Funciones vinculadas al servicio para TNB AWS

Compatible con roles vinculados al servicio	No

Un rol vinculado al 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 administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

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 los recursos de AWS TNB. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles, y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por AWS TNB, incluido el formato de los ARN para cada uno de los tipos de recursos, 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

Las políticas basadas en la identidad determinan si alguien puede crear, acceder o eliminar los recursos de AWS TNB de su cuenta. Estas acciones pueden generar costos 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. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, 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. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. 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 Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas

recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.

- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

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

Uso de la consola TNB AWS

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 recursos de AWS TNB de su propiedad. Cuenta de AWS 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 solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Ejemplos de políticas de roles de servicio

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

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

Puede crear el rol de servicio y su política de permisos con el servicio de IAM. 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 usuario de IAM.

AWS Función de servicio TNB

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

Se recomienda utilizar el siguiente rol de IAM y la política de confianza para el rol de servicio de AWS TNB. Al determinar el alcance de los permisos de esta política, tenga en cuenta que AWS TNB puede fallar y provocar errores de acceso denegado en relación con los recursos no incluidos en su política.

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

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

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

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

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

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

```
            ]
        },
        {
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TaggingPolicy"
        },
        {
            "Action": [
                "outposts:GetOutpost"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "OutpostPolicy"
        }
    ]
}
```

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "tnb.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

AWS Función de servicio TNB para el clúster Amazon EKS

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

El siguiente ejemplo muestra una AWS CloudFormation plantilla que crea un rol de servicio AWS TNB para la política de clústeres de Amazon EKS.

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBEKSClusterRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBEKSClusterRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - eks.amazonaws.com
            Action:
```

```
              - "sts:AssumeRole"
      Path: /
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSClusterPolicy"
```

Para obtener más información sobre los roles de IAM que utilizan AWS CloudFormation plantillas, consulte las siguientes secciones de la Guía del AWS CloudFormation usuario:

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

AWS Función de servicio TNB para el grupo de nodos Amazon EKS

Al crear recursos de un grupo de nodos de Amazon EKS en su NSD, proporciona el atributo `node_role` para especificar qué rol se utilizará para crear su grupo de nodos de Amazon EKS.

El siguiente ejemplo muestra una AWS CloudFormation plantilla que crea un rol de servicio AWS TNB para la política de grupo de nodos de Amazon EKS.

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBEKSNodeRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBEKSNodeRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSWorkerNodePolicy"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKS_CNI_Policy"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEBSCSIDriverPolicy"
```

```
      Policies:
        - PolicyName: EKSNodeRoleInlinePolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "logs:DescribeLogStreams"
                  - "logs:PutLogEvents"
                  - "logs:CreateLogGroup"
                  - "logs:CreateLogStream"
                Resource: "arn:aws:logs:*:*:log-group:/aws/tnb/tnb*"
        - PolicyName: EKSNodeRoleIpv6CNIPolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "ec2:AssignIpv6Addresses"
                Resource: "arn:aws:ec2:*:*:network-interface/*"
```

Para obtener más información sobre los roles de IAM que utilizan AWS CloudFormation plantillas, consulte las siguientes secciones de la Guía del AWS CloudFormation usuario:

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

AWS Función de servicio TNB para Multus

Cuando cree un recurso de Amazon EKS en su NSD y desee administrar Multus como parte de su plantilla de implementación, debe proporcionar el atributo `multus_role` para especificar qué rol se utilizará para administrar Multus.

El siguiente ejemplo muestra una AWS CloudFormation plantilla que crea un rol de servicio de AWS TNB 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 las funciones de IAM que utilizan AWS CloudFormation plantillas, consulte las siguientes secciones de la Guía del AWS CloudFormation usuario:

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

AWS Función de servicio de TNB para una política vinculada al ciclo de vida

Cuando su NSD o paquete de funciones 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 servicio de AWS TNB para una política de enlace de ciclo de vida.

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

Para obtener más información sobre las funciones de IAM que utilizan AWS CloudFormation plantillas, consulte las siguientes secciones de la Guía del AWS CloudFormation usuario:

- AWS::IAM::Role

- Selección de una plantilla de pila

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la AWS CLI API o. AWS

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

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

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

Problemas

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

No estoy autorizado a realizar ninguna acción en TNB AWS

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

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `tnb:`*`GetWidget`*.

```
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 su AWS 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 transferir una función a AWS TNB.

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.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en 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 su administrador. AWS 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 recursos de AWS TNB

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 admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si AWS TNB admite estas funciones, consulte. Cómo funciona AWS Telco Network Builder con IAM
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Validación de conformidad para TNB AWS

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.
- Diseño de arquitectura para garantizar la seguridad y el cumplimiento de la HIPAA en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar AWS las empresas para crear aplicaciones aptas para la HIPAA.

  > **Note**
  >
  > No Servicios de AWS todas cumplen los requisitos de la HIPAA. Para más información, consulte la Referencia de servicios compatibles con HIPAA.

- 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 el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan 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.
- 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 Regiones de AWS en zonas 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 TNB ejecuta su servicio de red en clústeres de EKS en una nube privada virtual (VPC) en AWS la región que elija.

Seguridad de la infraestructura en TNB AWS

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

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.

- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o 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 estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de 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 compatibles con 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 de programación de aplicaciones (API) de AWS TNB
  - AWS Kits de desarrollo de software (SDK)
- Usted es responsable de proteger el acceso a sus AWS recursos y a los componentes de la carga de trabajo, incluidos (entre otros):
  - Usuarios, grupos, roles y políticas de IAM
  - Depósitos de S3 que utiliza para almacenar sus datos para TNB AWS
  - Otros Servicios de AWS recursos que utiliza para respaldar el servicio de red que aprovisionó a través de TNB AWS
  - Su código de la aplicación
  - Conexiones entre el servicio de red que aprovisionó a través de 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 que haya aprovisionado a través de TNB. AWS

Modelo de seguridad de la conectividad de red

Los servicios de red que aprovisiona a través de AWS TNB se ejecutan en instancias de procesamiento dentro de una nube privada virtual (VPC) ubicada en AWS la región que seleccione. Una 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 de la VPC permanecen dentro de la red de AWS y no circulan por Internet. Algunas comunicaciones de los servicios internos se transmiten por Internet y están cifradas. Los servicios de red aprovisionados a través de AWS TNB para todos los clientes que se ejecutan en la misma región comparten la misma VPC. Los servicios de red aprovisionados a través de AWS TNB para diferentes clientes utilizan instancias informáticas independientes dentro de la misma VPC.

Las comunicaciones entre sus clientes de servicio de red y su servicio de red en AWS TNB atraviesan Internet. AWS TNB no gestiona estas conexiones. Es su responsabilidad proteger las conexiones de sus clientes.

Sus conexiones a AWS TNB a través de los AWS SDK AWS Management Console, AWS Command Line Interface (AWS CLI) y están cifradas.

Versión IMDS

AWS TNB admite instancias que utilizan la versión 2 del Instance Metadata Service (IMDSv2), un método orientado a la sesión. IMDSv2 incluye una mayor seguridad que IMDSV1. Para obtener más información, consulte Agregar defensa en profundidad contra firewalls abiertos, proxies inversos y vulnerabilidades SSRF con mejoras en el servicio de metadatos de instancias EC2 de Amazon.

Al lanzar la instancia, debe usar IMDSv2. Para obtener más información sobre IMDSv2, consulte Use IMDSv2 (Utilizar IMDSv2) en la Amazon EC2 User Guide for Linux Instances (Guía del usuario de Amazon EC2 para instancias de Linux).

Supervisión de AWS TNB

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS TNB y de sus soluciones de AWS. AWS ofrece AWS CloudTrail para vigilar AWS TNB, informar cuando algo no va bien y tomar medidas automáticamente cuando proceda.

Utilice CloudTrail para recopilar información detallada sobre las llamadas realizadas a la API de AWS. Puede almacenar estas llamadas como archivos de registro en Amazon S3. Puede utilizar estos registros de CloudTrail para determinar esta información como qué llamada se hizo, la dirección IP de origen de la que procede la llamada, quién la ha realizado y cuándo se hizo.

Los registros de CloudTrail contienen información sobre las llamadas a las acciones de la API para AWS TNB. También contienen información sobre las llamadas a las acciones de la API desde servicios como Amazon EC2 y Amazon EBS.

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

AWS Telco Network Builder se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones que realiza un usuario, rol o servicio de AWS en AWS TNB. CloudTrail captura las llamadas a la API de AWS TNB como eventos. Las llamadas capturadas incluyen llamadas desde la consola de AWS TNB y las llamadas desde el código a las operaciones de la API de AWS TNB. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos para AWS TNB. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Historial de eventos. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a AWS TNB, la dirección IP desde la que se realizó, quién la realizó y cuándo, etc.

Para obtener más información acerca de CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

Información de AWS TNB en CloudTrail

CloudTrail se habilita en su Cuenta de AWS cuando la crea. Cuando se produce una actividad en AWS TNB, esa actividad se registra en un evento de CloudTrail junto con otros eventos de servicio de AWS en Historial de eventos. Puede ver, buscar y descargar los últimos eventos de la Cuenta de AWS. Para obtener más información, consulte Ver eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de eventos en la Cuenta de AWS, incluidos los eventos de AWS TNB, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail
- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de registro de CloudTrail de varias regiones y Recibir archivos de registro de CloudTrail de varias cuentas

Todas las acciones de AWS TNB las registra CloudTrail y se documentan en la Referencia de la API de AWS Telco Network Builder. Por ejemplo, las llamadas a las acciones `CreateSolFunctionPackage`, `CreateSolNetworkInstance` y `CreateSolNetworkPackage` generan entradas en los archivos de registros de CloudTrail.

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

- Si la solicitud se realizó con credenciales de usuario AWS Identity and Access Management (IAM) o credenciales de usuario raíz.
- 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.

Para obtener más información, consulte el elemento userIdentity de CloudTrail.

Definición de entradas de archivos de registro de AWS TNB

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registros en un bucket de Amazon S3 que especifique. Los archivos de registro de CloudTrail pueden

contener una o varias entradas de registro. Un evento representa una solicitud específica realizada desde un origen y contiene información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. Los archivos de registro de CloudTrail no rastrean el orden en la pila de las llamadas públicas a la API, por lo que estas no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateSolFunctionPackage`.


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

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

Tareas de implementación de AWS TNB

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

En la siguiente tabla se enumeran tareas de implementación de AWS TNB:

Tarea	Lo que ocurre durante esta tarea
AppInstallation	Instala el complemento Multus en el clúster de Amazon EKS.
AppUpdate	Actualiza las funciones de red que ya están instaladas en una instancia de red.
ClusterStorageClassConfiguration	Configura la clase de almacenamiento (controlador CSI) en un clúster de Amazon EKS.
FunctionDeletion	Elimina las funciones de red de recursos de AWS TNB.
FunctionInstantiation	Implementa funciones de red mediante HELM.
FunctionUninstallation	Desinstala la función de red de un clúster de Amazon EKS.
HookExecution	Ejecuta los enlaces de ciclo de vida tal como se define en el NSD.
InfrastructureCancellation	Cancela un servicio de red.
InfrastructureInstantation	Concede recursos AWS en nombre del usuario.

Tarea	Lo que ocurre durante esta tarea
InfrastructureTermination	Desaprovisiona los recursos AWS invocados a través de AWS TNB.
InventoryDeregistration	Anula el registro de recursos AWS de AWS TNB.
KubernetesClusterConfiguration	Configura el clúster de Kubernetes y añade roles de IAM adicionales al AuthMap de Amazon EKS, tal como se define en el NSD.
NetworkServiceFinalization	Finaliza el servicio de red y proporciona una actualización del estado de éxito o error.
NetworkServiceInstantiation	Inicia el servicio de red.
SelfManagedNodesConfiguration	Arranca los nodos autogestionados con Amazon EKS y el plano de control de Kubernetes.

Service Quotas para AWS Telco Network Builder

Las Service Quotas, que también se denominan límites, establecen el número máximo de recursos u operaciones de servicio que puede haber en su cuenta de AWS. Para obtener más información, consulte Cuotas de servicio de AWS en la Referencia general de Amazon Web Services.

A continuación se incluyen las service quotas de 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	Número máximo de paquetes de funciones en una región.
Paquetes de red	Cada región admitida: 40	Sí	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 del usuario de AWS TNB

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

Cambio	Descripción	Fecha
Versión de Kubernetes para clúster	AWS TNB ahora es compatible con las versiones 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 nodo AWS.Networking.ENI.	2 de abril de 2024
Support para el cifrado de volúmenes raíz de Amazon EBS	Puede habilitar el cifrado de Amazon EBS para el volumen raíz de Amazon EBS. Para habilitarlo, añada las propiedades en el nodo AWS.Compute.EKS o ManagedNodeAWS.Compute.EKS. SelfManagedNode	2 de abril de 2024
Support for node `labels`	Puede adjuntar etiquetas de nodo a su grupo de nodos en el nodo AWS.Compute.EKS o AWS.Compute.EKS. ManagedNode SelfManagedNode	19 de marzo de 2024
Support para interfaz de red `source_dest_check`	Puede indicar si desea activar o desactivar la comprobación de origen/destino de la interfaz de red a través del nodo AWS.Networking.ENI.	25 de enero de 2024

Compatibilidad con instancias de Amazon EC2 con datos de usuario personalizados	Puede lanzar instancias de Amazon EC2 con datos de usuario personalizados a través de .Compute. AWS UserData nodo.	16 de enero de 2024
Compatibilidad con grupo de seguridad	AWS TNB le permite importar el AWS recurso del grupo de seguridad.	8 de enero de 2024
Descripción actualizada de `network_interfaces`	Cuando la `network_interfaces` propiedad se incluye en el SelfManagedNode nodo AWS.Compute.eks ManagedNode o AWS.Compute.eks, AWS TNB obtiene el permiso relacionado con los ENI de la propiedad, si está disponible, o de la propiedad. `multus_role` `node_role`	18 de diciembre de 2023
Compatibilidad con clúster privado	AWS TNB ahora admite clústeres privados. Para indicar un clúster privado, establezca la propiedad `access` en `PRIVATE`.	11 de diciembre de 2023
Versión de Kubernetes para clúster	AWS TNB ahora es compatible con las versiones 1.28 de Kubernetes para crear clústeres de Amazon EKS.	11 de diciembre de 2023

AWS TNB admite 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 TNB añade soporte para IPv6	AWS TNB ahora admite la creación de instancias de red con infraestructura IPv6. Compruebe los nodos AWS.Networking.VPC, .Networking.Subnet, .Networking.AWSAWS https://docs.aws.amazon.com/tnb/latest/ug/node-internet-gateway.htmlInternetGatewayAWS, .Redes. SecurityGroupIngressRule, AWS.Redes. SecurityGroupEgressRuley AWS.compute.EKS para configuraciones de IPv6. También añadimos los nodos AWS.Networking.NATGateway y AWS.Networking.Route para la configuración de NAT64. Hemos actualizado el rol de servicio AWS TNB y el rol de servicio AWS TNB para el grupo de nodos Amazon EKS para los permisos de IPv6. Consulte Ejemplos de políticas de roles de servicio.	16 de noviembre de 2023

Se agregaron permisos a la política de roles de servicio de AWS TNB	Hemos añadido permisos a la política de funciones de servicio de AWS TNB para Amazon S3 y AWS CloudFormation para permitir la instanciación de la infraestructura.	23 de octubre de 2023
AWS TNB se lanzó en más regiones	AWS TNB ya 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 .compute.eks AWS SelfManagedNode	AWS TNB ahora admite etiquetas para la definición del nodo. `AWS.Compute.EKSSelfManagedNode`	22 de agosto de 2023
AWS TNB admite instancias que utilizan IMDSv2	Al lanzar la instancia, debe usar IMDSv2.	14 de agosto de 2023
Se actualizaron los permisos para MultusRoleInlinePolicy	MultusRoleInlinePolicy Ahora incluye el `ec2:DeleteNetworkInterface` permiso.	7 de agosto de 2023
Versión de Kubernetes para clúster	AWS TNB ahora 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 TNB admite AuthRole que le permite añadir funciones de IAM al clúster de Amazon EKS para que los usuarios puedan acceder al clúster `aws-auth` `ConfigMap` de Amazon EKS mediante una función de IAM.	19 de julio de 2023
AWS TNB admite grupos de seguridad.	Se agregó el archivo AWS.Networking. SecurityGroup, AWS.Networking. SecurityGroupEgressRuley AWS.Networking. SecurityGroupIngressRulea la plantilla NSD.	18 de julio de 2023
Versión de Kubernetes para clúster	AWS TNB admite las versiones 1.22 a 1.26 de Kubernetes para crear clústeres de Amazon EKS. AWS TNB ya no es compatible con las versiones 1.21 de Kubernetes.	11 de mayo de 2023
AWS.Compute.eks SelfManagedNode	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 usuario de AWS TNB.	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.