Guide de l’utilisateur

AWS Générateur de réseaux de télécommunications

AWS Générateur de réseaux de télécommunications: Guide de l’utilisateur

Table of Contents

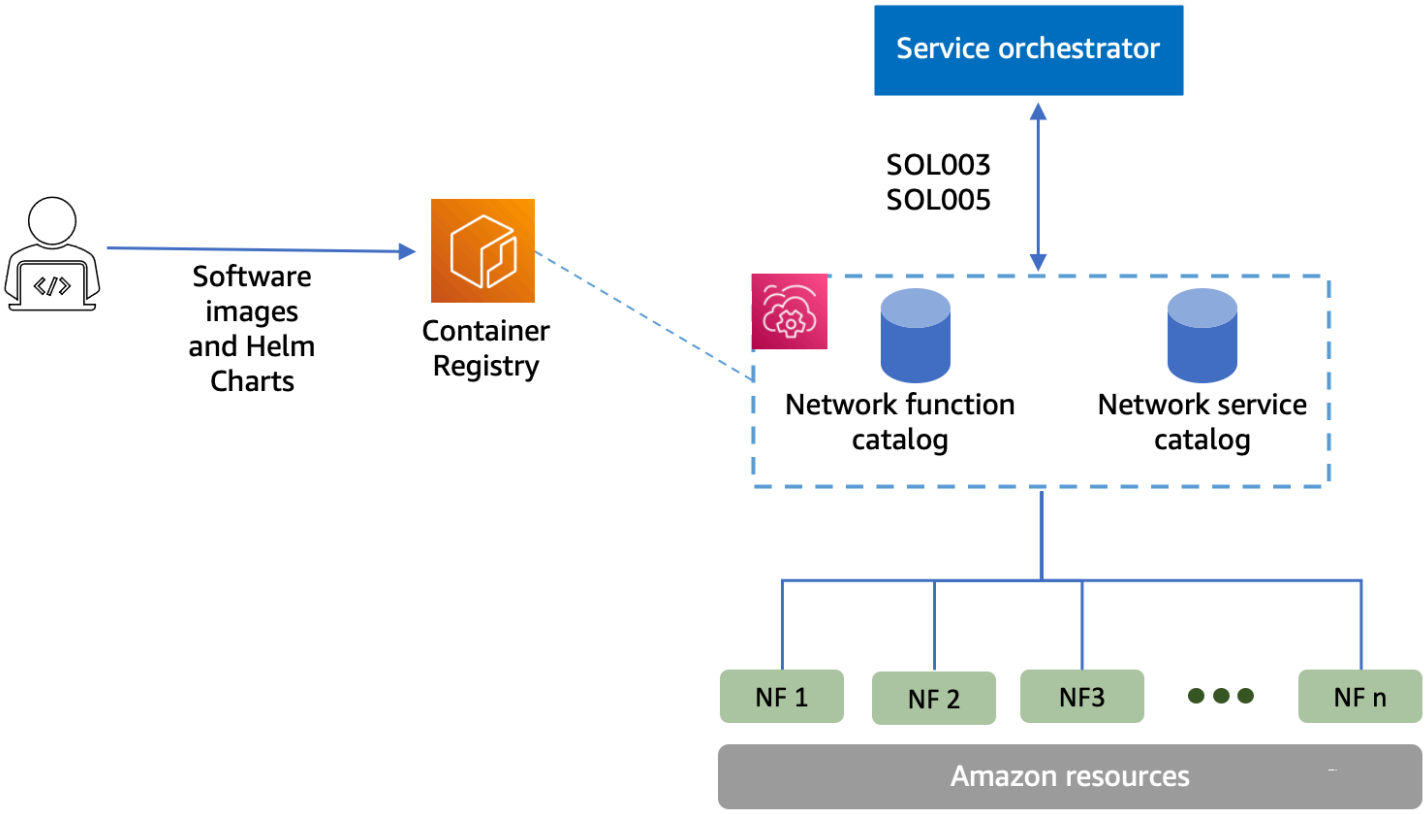
Qu'est-ce queAWS Telco Network Builder ?

AWSTelco Network Builder (AWSTNB) est unAWS service qui fournit aux fournisseurs de services de communication (CSP) un moyen efficace de déployer, de gérer et de faire évoluer les réseaux 5G sur lesAWS infrastructures.

AvecAWS TNB, vous déployez des réseaux 5G évolutifs et sécurisés enAWS Cloud utilisant des images logicielles conteneurisées de manière automatisée. Vous n'avez pas besoin de vous former à de nouvelles technologies, de décider du service de calcul à utiliser ou de savoir comment approvisionner et configurer des ressources AWS.

Vous décrivez plutôt l'infrastructure de votre réseau et fournissez les images logicielles des fonctions réseau fournies par vos partenaires fournisseurs de logiciels indépendants (ISV). AWS TNB s'intègre à des orchestrateurs deAWS services et à des services tiers pour fournir automatiquement l'AWSinfrastructure nécessaire, déployer des fonctions réseau conteneurisées et configurer la gestion de la mise en réseau et des accès afin de créer un service réseau pleinement opérationnel.

Le schéma suivant illustre les intégrations logiques entre leAWS TNB et les orchestrateurs de services pour déployer des fonctions réseau à l'aide d'interfaces standard basées sur l'ETSI (European Telecommunications Standards Institute).



Rubriques

Vous ne connaissez pas encore AWS ?

Si vous n'êtes pas encore familiarisé avec les produits et services AWS, commencez par prendre connaissance des ressources suivantes :

- Introduction à AWS
- Démarrer avec AWS

À qui s'AWSadresse TNB ?

AWSTNB est destiné aux CSP qui souhaitent tirer parti de la rentabilité, de l'agilité et de l'élasticité qu'ilsAWS Cloud offrent sans écrire ni gérer des scripts et des configurations personnalisés pour concevoir, déployer et gérer des services réseau. AWS TNB fournit automatiquement l'AWSinfrastructure nécessaire, déploie des fonctions réseau conteneurisées et configure la gestion du réseau et des accès afin de créer des services réseau pleinement opérationnels basés sur les descripteurs de services réseau définis par le CSP et les fonctions réseau que le CSP souhaite déployer.

Pourquoi utiliserAWS TNB ?

Voici quelques-unes des raisons pour lesquelles un CSP souhaiterait utiliser leAWS TNB :

Aide à simplifier les tâches

> Améliorez l'efficacité de vos opérations réseau, notamment en déployant de nouveaux services, en mettant à jour et en mettant à niveau les fonctions réseau et en modifiant les topologies de l'infrastructure réseau.

S'intègre aux orchestrateurs

AWSTNB s'intègre à des orchestrateurs de services tiers populaires conformes à l'ETSI.

Balances

Vous pouvez configurerAWS TNB pour adapter lesAWS ressources sous-jacentes afin de répondre à la demande de trafic, effectuer plus efficacement des mises à jour des fonctions réseau, déployer des modifications topologiques de l'infrastructure réseau et réduire le temps de déploiement des nouveaux services 5G de quelques jours à quelques heures.

Inspecte et surveille lesAWS ressources

AWSTNB vous permet d'inspecter et de surveiller lesAWS ressources qui prennent en charge votre réseau sur un tableau de bord unique, telles qu'Amazon VPC, Amazon EC2 et Amazon EKS.

Supporte les modèles de services

AWSTNB vous permet de créer des modèles de service pour toutes les charges de travail télécoms (RAN, Core, IMS). Vous pouvez créer une nouvelle définition de service, réutiliser un modèle existant ou intégrer un pipeline d'intégration et de fourniture continues (CI/CD) pour publier une nouvelle définition.

Suit les modifications apportées aux déploiements réseau

Lorsque vous modifiez la configuration sous-jacente d'un déploiement de fonctions réseau, par exemple en modifiant le type d'instance d'un type d'instance Amazon EC2, vous pouvez suivre les modifications de manière reproductible et évolutive. Pour ce faire manuellement, il faudrait gérer l'état du réseau, créer et supprimer des ressources et prêter attention à l'ordre des modifications nécessaires. Lorsque vous utilisezAWS TNB pour gérer le cycle de vie de votre fonction réseau, vous n'apportez les modifications qu'aux descripteurs de service réseau décrivant la fonction réseau. AWS TNB effectuera alors automatiquement les modifications requises dans le bon ordre.

Simplifie le cycle de vie des fonctions du réseau

Vous pouvez gérer la première version et toutes les versions suivantes d'une fonction réseau et spécifier quand effectuer la mise à niveau. Vous pouvez également gérer vos applications RAN, Core, IMS et réseau de la même manière.

Accès àAWS TNB

Vous pouvez créer vos ressourcesAWS TNB, y accéder et les gérer à l'aide des interfaces suivantes :

- AWSConsole TNB : fournit une interface Web pour gérer votre réseau.
- AWSAPI TNB — Fournit une API RESTful pour effectuer des actionsAWS TNB. Pour plus d'informations, voir AWSTNB API Reference
- AWS Command Line Interface(AWS CLI) — Fournit des commandes pour un large éventail deAWS services, y comprisAWS TNB. Elle est prise en charge sur Windows, macOS et Linux. Pour plus d'informations, veuillez consulter AWS Command Line Interface.
- AWSSDK : fournit des API spécifiques à chaque langue et complète de nombreux détails de connexion. Ces outils incluent le calcul des signatures, la gestion des nouvelles tentatives de demande et la gestion des erreurs. Pour de plus amples informations, veuillez consulter SDK AWS.

Tarification duAWS TNB

AWSTNB aide les CSP à automatiser le déploiement et la gestion de leurs réseaux de télécommunications surAWS. Vous payez pour les deux dimensions suivantes lorsque vous utilisezAWS TNB :

- Par heures d'élément de fonction réseau géré (MNFI).
- Par nombre de demandes d'API.

Vous êtes également soumis à des frais supplémentaires lorsque vous utilisez d'autresAWS services conjointement avecAWS TNB. Pour plus d'informations, consultez Tarification deAWS TNB.

Pour consulter votre facture, accédez au Tableau de bord de gestion des coûts et de la facturation dans la console AWS Billing and Cost Management. Votre facture contient des liens vers les rapports d'utilisation qui fournissent des détails supplémentaires sur votre facture. Pour plus d'informations sur la facturation desAWS comptes, consultez la section Facturation desAWS comptes.

Pour toute question relative à la facturation, aux comptes et aux événements AWS, contactez AWS Support.

AWS Trusted Advisorest un service que vous pouvez utiliser pour optimiser les coûts, la sécurité et les performances de votreAWS environnement. Pour plus d'informations, consultez AWSTrusted Advisor.

Présentation de

Pour plus d'informations sur la façon de commencer àAWS utiliser TNB, consultez les rubriques suivantes :

- Configuration du AWS TNB— Effectuez les étapes préalables.
- Débuter avec AWS TNB— Déployez votre première fonction réseau, telle que l'unité centralisée (CU), la fonction de gestion de l'accès et de la mobilité (AMF), la fonction de plan utilisateur (UPF) ou un cœur 5G complet.

Comment fonctionneAWS TNB

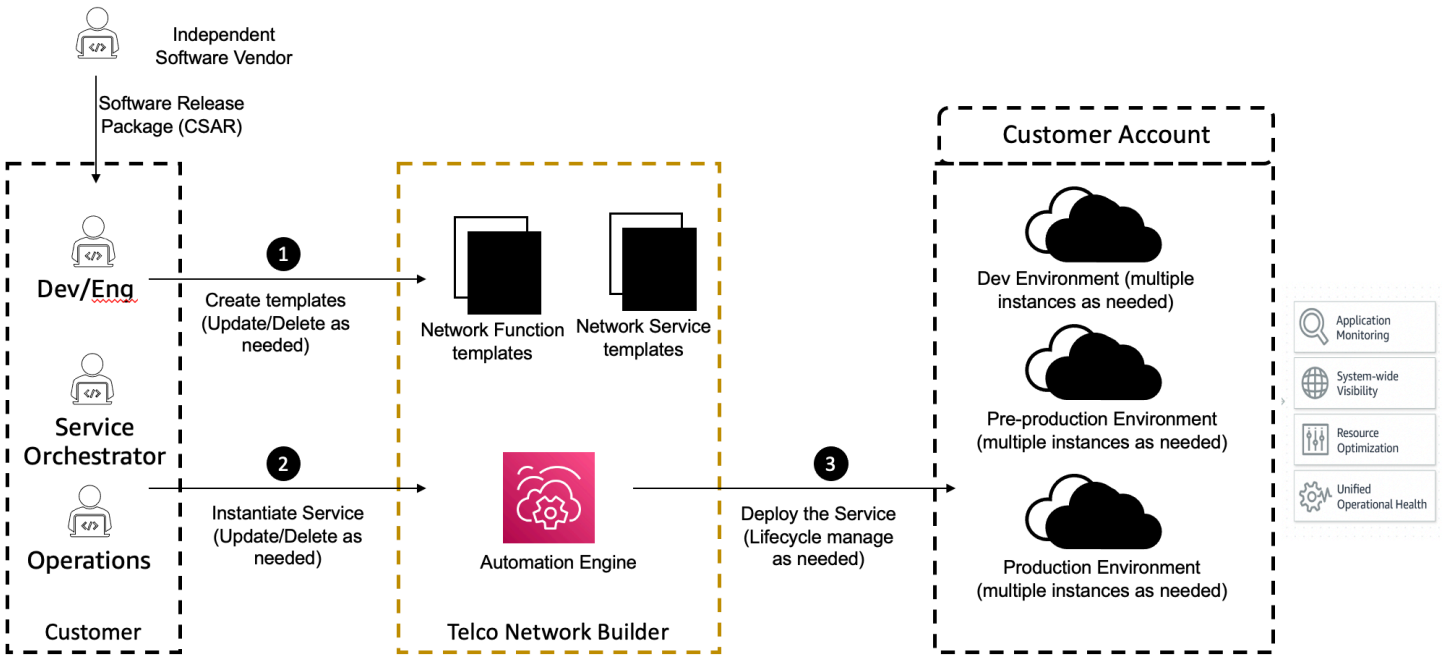
AWSTNB s'intègre à des end-to-end orchestrateurs et à desAWS ressources standardisés pour exploiter des réseaux 5G complets.

AWSTNB vous permet d'ingérer des packages de fonctions réseau et des descripteurs de services réseau (NSD) et vous fournit le moteur d'automatisation nécessaire au fonctionnement de vos réseaux. Vous pouvez utiliser votre end-to-end orchestrateur et l'intégrer aux APIAWS TNB, ou utiliser les SDKAWS TNB pour créer votre propre flux d'automatisation. Pour plus d'informations, veuillez consulter AWSArchitecture TNB.

Rubriques

AWSArchitecture TNB

AWSTNB vous permet d'effectuer des opérations de gestion du cycle de vie via l'AWS Management ConsoleAPI RESTAWS TNB et les SDK.AWS CLI Cela permet aux différents acteurs du CSP, tels que les membres des équipes d'ingénierie, des opérations et des systèmes programmatiques, de tirer parti duAWS TNB. Vous créez et chargez un package de fonctions réseau sous la forme d'un fichier Cloud Service Archive (CSAR). Le fichier CSAR contient des diagrammes Helm, des images logicielles et un descripteur de fonction réseau (NFD). Vous pouvez utiliser des modèles pour déployer plusieurs configurations de ce package à plusieurs reprises. Vous créez des modèles de service réseau qui définissent l'infrastructure et les fonctions réseau que vous souhaitez déployer. Vous pouvez utiliser des remplacements de paramètres pour déployer différentes configurations à différents emplacements. Vous pouvez ensuite instancier un réseau à l'aide des modèles et déployer vos fonctions réseau surAWS l'infrastructure. AWS TNB vous offre la visibilité de vos déploiements.



Intégration à Services AWS

Un réseau 5G est composé d'un ensemble de fonctions réseau conteneurisées interconnectées déployées sur des milliers de clusters Kubernetes. AWS TNB s'intègre aux APIServices AWS suivantes, spécifiques aux télécommunications, afin de créer un service réseau pleinement opérationnel :

- Amazon Elastic Container Registry (Amazon ECR) pour stocker les artefacts de fonctions réseau des fournisseurs de logiciels indépendants (ISV).
- Amazon Elastic Kubernetes Service(Amazon EKS) pour configurer des clusters.
- Amazon VPC pour les structures de mise en réseau.
- Groupes de sécurité utilisantAWS CloudFormation.
- AWS CodePipelinepour les cibles de déploiement dansRégions AWSAWS Local Zones etAWS Outposts.
- IAM pour définir les rôles.
- AWS Organizationspour contrôler l'accès aux APIAWS TNB.
- AWS Health DashboardetAWS CloudTrail pour surveiller l'état de santé et publier des statistiques.

AWSQuotas de ressources TNB

VotreCompte AWS dispose de quotas par défaut, anciennement appelés quotas par défaut, pour chacun d'entre euxService AWS. Sauf indication contraire, chaque quota est spécifique à unRégion AWS. Vous pouvez demander des quotas pour certains quotas, mais pas pour tous les quotas.

Pour afficher les quotas pourAWS TNB, ouvrez la console Service Quotas de quotas de quotas de quotas de quotas pour TNB. Dans le panneau de navigation Services AWS, choisissez et sélectionnez AWSTNB.

Pour demander une augmentation de quota, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

Voici les quotas de quotas de quotas de quotas de quotas concernant leAWS TNB.Compte AWS

Quotas de ressources	Description	Valeur par défaut	Ajustable?
Instances de service réseau	Le nombre maximal d'instances de service réseau dans une région.	800	Oui
Opérations de service réseau continues simultanées	Le nombre maximal d'opérations de service réseau en cours de service réseau en cours dans une région.	40	Oui
Packages réseau	Le nombre maximal de packages réseau dans une région.	40	Oui
Packages de fonctions	Le nombre maximal de packages de fonctions dans une région.	200	Oui

AWS Concepts du TNB

Cette rubrique décrit les concepts essentiels pour vous aider à commencer à utiliser le AWS TNB.

Table des matières

Cycle de vie d'une fonction réseau

AWS TNB vous aide tout au long du cycle de vie des fonctions de votre réseau. Le cycle de vie des fonctions réseau comprend les étapes et activités suivantes :

Planification

1. Planifiez votre réseau en identifiant les fonctions réseau à déployer.
2. Placez les images du logiciel de fonction réseau dans un référentiel d'images de conteneur.
3. Créez les packages CSAR à déployer ou à mettre à niveau.
4. Utilisez AWS TNB pour télécharger le package CSAR qui définit votre fonction réseau (par exemple, CU AMF et UPF) et intégrez-le à un pipeline d'intégration et de livraison continues (CI/CD) qui peut vous aider à créer de nouvelles versions de votre package CSAR à mesure que de nouvelles images du logiciel de fonction réseau ou des scripts client sont disponibles.

Configuration

1. Identifiez les informations requises pour le déploiement, telles que le type de calcul, la version de la fonction réseau, les informations IP et les noms des ressources.
2. Utilisez ces informations pour créer votre descripteur de service réseau (NSD).
3. Ingérez des NSD qui définissent les fonctions de votre réseau et les ressources nécessaires à l'instanciation de la fonction réseau.

Instanciation

1. Créez l'infrastructure requise par les fonctions du réseau.

2. Instanciez (ou provisionnez) la fonction réseau telle que définie dans son NSD et commencez à transporter le trafic.
3. Validez les actifs.

Production

Au cours du cycle de vie de la fonction réseau, vous effectuerez des opérations de production, telles que :

- Mettez à jour la configuration de la fonction réseau, par exemple, mettez à jour une valeur dans la fonction réseau déployée.
- Remplacez ou désactivez la fonction réseau.

Utiliser des interfaces standardisées

AWS TNB s'intègre aux orchestrateurs de services conformes à la norme ETSI (European Telecommunications Standards Institute), ce qui vous permet de simplifier le déploiement de vos services réseau. Les orchestrateurs de services peuvent utiliser les SDK AWS TNB, la CLI ou les API pour lancer des opérations, telles que l'instanciation ou la mise à niveau d'une fonction réseau vers une nouvelle version.

AWS TNB prend en charge les spécifications suivantes.

Spécification de	Version	Description
ETSI SOL001	v3.6.1	Définit les normes permettant d'autoriser les descripteurs de fonctions réseau basés sur TOSCA.
ETSI SOL002	v3.6.1	Définit les modèles relatifs à la gestion des fonctions réseau.
ETSI SOL003	v3.6.1	Définit les normes pour la gestion du cycle de vie des fonctions réseau.
ETSI SOL004	v3.6.1	Définit les normes CSAR pour les packages de fonctions réseau.

Spécification de	Version	Description
ETSI SOL005	v3.6.1	Définit les normes relatives aux packages de services réseau et à la gestion du cycle de vie des services réseau.
ETSI SOL007	v3.5.1	Définit les normes permettant d'autoriser les descripteurs de service réseau basés sur TOSCA.

Packages de fonctions réseau pour AWS TNB

Avec AWS TNB, vous pouvez stocker des packages de fonctions réseau conformes aux normes ETSI SOL001/SOL004 dans un catalogue de fonctions. Vous pouvez ensuite télécharger des packages Cloud Service Archive (CSAR) contenant des artefacts décrivant le fonctionnement de votre réseau.

- Descripteur de fonction réseau — Définit les métadonnées pour l'intégration des packages et la gestion des fonctions réseau
- Images logicielles — Fait référence à la fonction réseau Container Images. Amazon Elastic Container Registry (Amazon ECR) peut faire office de référentiel d'images de fonctions réseau.
- Fichiers supplémentaires : à utiliser pour gérer le fonctionnement du réseau ; par exemple, les scripts et les diagrammes Helm.

Le CSAR est un package défini par la norme OASIS TOSCA et inclut un descripteur de réseau/service conforme à la spécification OASIS TOSCA YAML. Pour plus d'informations sur la spécification YAML requise, consultezRéférence TOSCA pourAWS TNB.

Voici un exemple de descripteur de fonction réseau.

```
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"
```

Descripteurs de service de fonction réseau pour TNB AWS

AWS TNB stocke les descripteurs de service réseau (NSD) relatifs aux fonctions réseau que vous souhaitez déployer et à la manière dont vous souhaitez les déployer dans le catalogue. Vous pouvez télécharger votre fichier YAML NSD, comme décrit par ETSI SOL007 pour inclure les éléments suivants :

- NF que vous souhaitez déployer
- Instructions de mise en réseau
- Instructions de calcul
- Hooks du cycle de vie (scripts personnalisés)

AWS TNB prend en charge les normes ETSI pour la modélisation des ressources, telles que le réseau, le service et la fonction, dans le langage TOSCA. AWS TNB vous permet de les utiliser plus efficacement en les Services AWS modélisant de manière à ce que votre orchestrateur de services conforme à la norme ETSI puisse les comprendre.

Ce qui suit est un extrait d'un NSD montrant comment modéliser. Services AWS La fonction réseau sera déployée sur un cluster Amazon EKS avec Kubernetes version 1.27. Les sous-réseaux des applications sont Subnet01 et Subnet02. Vous pouvez ensuite définir le NodeGroups pour vos applications à l'aide d'une Amazon Machine Image (AMI), d'un type d'instance et d'une configuration de mise à l'échelle automatique.

```
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
```

Gestion et opérations pour AWS TNB

Avec AWS TNB, vous pouvez gérer votre réseau à l'aide d'opérations de gestion standardisées conformément aux normes ETSI SOL003 et SOL005. Vous pouvez utiliser les API AWS TNB pour effectuer des opérations de cycle de vie telles que :

- Instanciation des fonctions de votre réseau.
- Mettre fin aux fonctions de votre réseau.
- Mettre à jour les fonctions de votre réseau pour annuler les déploiements Helm.
- Gestion des versions de vos packages de fonctions réseau.
- Gestion des versions de vos NSD.
- Récupération d'informations sur les fonctions de votre réseau déployé.

Descripteurs de service réseau pour TNB AWS

Un descripteur de service réseau (NSD) est un `.yaml` fichier d'un package réseau qui utilise la norme TOSCA pour décrire les fonctions réseau que vous souhaitez déployer et l' AWS infrastructure sur laquelle vous souhaitez déployer les fonctions réseau. Pour définir votre NSD et configurer vos ressources sous-jacentes et les opérations du cycle de vie du réseau, vous devez comprendre le schéma NSD TOSCA pris en charge par TNB. AWS

Votre fichier NSD est divisé en plusieurs parties :

1. Version de définition TOSCA — Il s'agit de la première ligne de votre fichier NSD YAML et contient les informations de version, illustrées dans l'exemple suivant.

   ```
   tosca_definitions_version: tnb_simple_yaml_1_0
   ```

2. VNFD — Le NSD contient la définition de la fonction réseau sur laquelle effectuer les opérations du cycle de vie. Chaque fonction réseau doit être identifiée par les valeurs suivantes :
   - Un identifiant unique pour`descriptor_id`. L'identifiant doit correspondre à celui du package CSAR de la fonction réseau.
   - Un nom unique pour`namespace`. Le nom doit être associé à un identifiant unique afin de pouvoir le référencer plus facilement dans l'ensemble de votre fichier NSD YAML, comme illustré dans l'exemple suivant.

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

3. Modèle de topologie : définit les ressources à déployer, le déploiement des fonctions réseau et tous les scripts personnalisés, tels que les hooks du cycle de vie. Voici un exemple :

```
topology_template:

  node_templates:

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

4. Nœuds supplémentaires : chaque ressource modélisée comporte des sections pour les propriétés et les exigences. Les propriétés décrivent les attributs facultatifs ou obligatoires d'une ressource, tels que la version. Les exigences décrivent les dépendances qui doivent être fournies en tant qu'arguments. Par exemple, pour créer une ressource de groupe de nœuds Amazon EKS, celle-ci doit être créée au sein d'un cluster Amazon EKS. Voici un exemple :

```
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
```

Configuration du AWS TNB

Configurez AWS TNB en effectuant les tâches décrites dans cette rubrique.

Tâches

- Inscrivez-vous pour AWS
- Choisissez une AWS région
- Notez le point de terminaison du service
- (Facultatif) Installez le AWS CLI
- Créer un utilisateur IAM
- Configurer les rôles AWS TNB

Inscrivez-vous pour AWS

Lorsque vous vous inscrivez à Amazon Web Services, vous êtes automatiquement Compte AWS inscrit à tous les services AWS, y compris AWS TNB. Seuls les services que vous utilisez vous sont facturés.

Si vous en avez Compte AWS déjà un, passez à la tâche suivante. Si vous n'avez pas de Compte AWS, utilisez la procédure suivante pour en créer un.

Pour créer un Compte AWS

1. Ouvrez https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

Choisissez une AWS région

Pour consulter la liste des régions disponibles pour AWS TNB, consultez la liste des services AWS régionaux. Pour consulter la liste des points de terminaison pour un accès programmatique, voir les points de terminaison AWS TNB dans le. Références générales AWS

Notez le point de terminaison du service

Pour vous connecter par programmation à un AWS service, vous utilisez un point de terminaison. Outre les points de AWS terminaison standard, certains AWS services proposent des points de terminaison FIPS dans certaines régions. Pour plus d'informations, consultez Points de terminaison du service AWS.

Nom de la région	Région	Point de terminaison	Protocole	
US East (Virginie du Nord)	us-east-1	tnb.us-east-1.amazonaws.com	HTTPS	
USA Ouest (Oregon)	us-west-2	tnb.us-west-2.amazonaws.com	HTTPS	
Asia Pacific (Seoul)	ap-northeast-2	tnb.ap-northeast-2.amazonaws.com	HTTPS	
Asie-Pacifique (Sydney)	ap-southeast-2	tnb.ap-southeast-2.amazonaws.com	HTTPS	
Canada (Centre)	ca-central-1	tnb.ca-central-1.amazonaws.com	HTTPS	
Europe (Francfort)	eu-central-1	tnb.eu-central-1.amazonaws.com	HTTPS	

Nom de la région	Région	Point de terminaison	Protocole	
Europe (Paris)	eu-west-3	tnb.eu-west-3.amazonaws.com	HTTPS	
Europe (Espagne)	eu-south-2	tnb.eu-south-2.amazonaws.com	HTTPS	
Europe (Stockholm)	eu-north-1	tnb.eu-north-1.amazonaws.com	HTTPS	
Amérique du Sud (São Paulo)	sa-east-1	tnb.sa-east-1.amazonaws.com	HTTPS	

(Facultatif) Installez le AWS CLI

Le AWS Command Line Interface (AWS CLI) fournit des commandes pour un large éventail de AWS produits et est pris en charge sous Windows, macOS et Linux. Vous pouvez accéder à AWS TNB à l'aide du AWS CLI. Consultez le AWS Command Line Interface Guide de l'utilisateur pour démarrer. Pour plus d'informations sur les commandes pour AWS TNB, voir tnb dans le manuel de référence des AWS CLI commandes.

Créer un utilisateur IAM

AWS Identity and Access Management (IAM) est un service Web qui vous permet de contrôler en toute sécurité l'accès aux AWS ressources. Créez un rôle d'utilisateur IAM pour utiliser des informations d'identification à court terme pour y accéder AWS.

Pour créer le rôle, suivez les instructions de la section Mise en route du guide de AWS IAM Identity Center l'utilisateur.

Vous pouvez également configurer l'accès par programmation en configurant le AWS CLI à utiliser AWS IAM Identity Center dans le guide de l'AWS Command Line Interface utilisateur.

Configurer les rôles AWS TNB

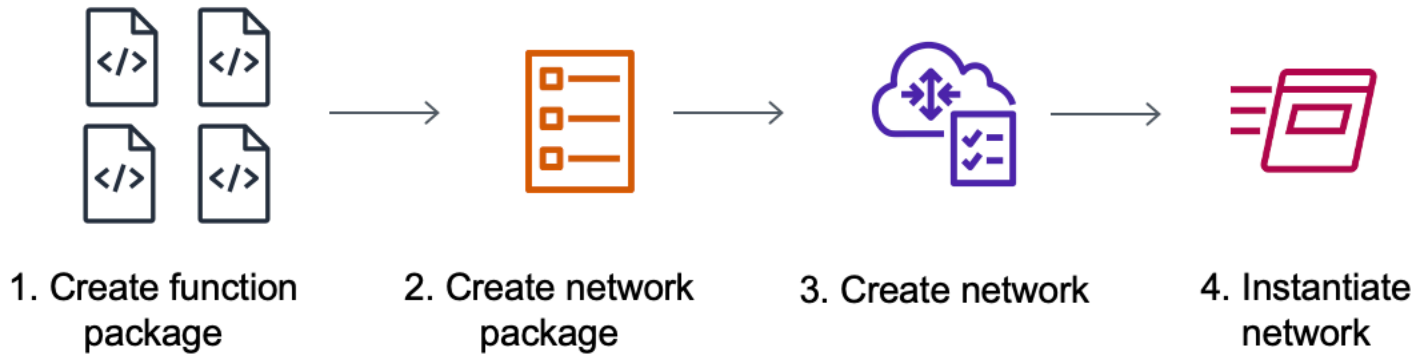
Vous devez créer un rôle de service IAM pour gérer les différentes parties de votre solution AWS TNB. AWS Les rôles de service TNB peuvent effectuer des appels d'API vers d'autres AWS services, tels que AWS CloudFormation AWS CodeBuild, et divers services de calcul et de stockage, en votre nom, afin d'instancier et de gérer les ressources pour votre déploiement.

Pour plus d'informations sur le rôle de service AWS TNB, consultezGestion des identités et des accès pour AWS TNB.

Débuter avec AWS TNB

Ce didacticiel explique comment utiliser le AWS TNB pour déployer une fonction réseau, par exemple l'unité centralisée (CU), la fonction de gestion de l'accès et de la mobilité (AMF) ou la fonction de plan utilisateur 5G (UPF).

Le schéma suivant illustre le processus de déploiement :



Tâches

- Prérequis
- Création d'un package de fonctions
- Création d'un package réseau
- Création et instanciation d'une instance réseau
- Nettoyage

Prérequis

Avant de pouvoir effectuer un déploiement réussi, vous devez disposer des éléments suivants :

- Un plan de Support aux AWS entreprises.
- Autorisations via les rôles IAM.
- Un package de fonctions réseau (NF) conforme à la norme ETSI SOL001/SOL004.
- Modèles de descripteur de service réseau (NSD) conformes à la norme ETSI SOL007.

Création d'un package de fonctions

Pour créer un package de fonctions

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, choisissez Function packages.
3. Choisissez Créer un package de fonctions.
4. Sous Télécharger le package de fonctions, choisissez Choisir un fichier et téléchargez votre package CSAR sous forme de `.zip` fichier.
5. (Facultatif) Sous Balises, choisissez Ajouter une nouvelle balise et entrez une clé et une valeur. Vous pouvez utiliser des balises pour rechercher et filtrer vos ressources ou suivre vos AWS coûts.
6. Choisissez Suivant.
7. Vérifiez les détails du package, puis choisissez Create function package.

Création d'un package réseau

Pour créer un package réseau

1. Dans le volet de navigation, sélectionnez Network packages.
2. Choisissez Créer un package réseau.
3. Sous Télécharger le package réseau, choisissez Choisir un fichier et téléchargez votre NSD sous forme de `.zip` fichier.
4. (Facultatif) Sous Balises, choisissez Ajouter une nouvelle balise et entrez une clé et une valeur. Vous pouvez utiliser des balises pour rechercher et filtrer vos ressources ou suivre vos AWS coûts.
5. Choisissez Suivant.
6. Choisissez Créer un package réseau.

Création et instanciation d'une instance réseau

Pour créer et instancier une instance réseau

1. Dans le volet de navigation, sélectionnez Networks.

2. Choisissez Create network instance.
3. Entrez un nom et une description pour le réseau, puis choisissez Next.
4. Sélectionnez votre NSD. Vérifiez les informations, puis choisissez Next.
5. Choisissez Create network instance. L'état initial est`Created`.
6. Choisissez l'ID de l'instance réseau, puis choisissez Instantiate.
7. Choisissez Instancier le réseau.
8. Utilisez l'icône Actualiser pour suivre l'état de votre instance réseau.

Nettoyage

Pour nettoyer vos ressources

1. Dans le volet de navigation, sélectionnez Networks.
2. Choisissez l'ID du réseau, puis cliquez sur Terminate.
3. Lorsque vous êtes invité à confirmer, entrez l'ID réseau, puis choisissez Terminate.
4. Utilisez l'icône Actualiser pour suivre l'état de votre instance réseau.
5. (Facultatif) Sélectionnez le réseau, puis choisissez Supprimer.

Packages de fonctions pour AWS TNB

Un package de fonctions est un fichier .zip au format CSAR (Cloud Service Archive) qui contient une fonction réseau (une application de télécommunication standard ETSI) et un descripteur de package de fonctions qui utilise la norme TOSCA pour décrire comment les fonctions réseau doivent s'exécuter sur votre réseau.

Tâches

- Créez un package de fonctions dans AWS TNB
- Afficher un package de fonctions dans AWS TNB
- Téléchargez un package de fonctions depuis AWS TNB
- Supprimer un package de fonctions de AWS TNB

Créez un package de fonctions dans AWS TNB

Découvrez comment créer un package de fonctions dans le catalogue des fonctions réseau AWS TNB. La création d'un package de fonctions est la première étape pour créer un réseau dans TNB. Une fois que vous avez chargé un package de fonctions, vous devez créer un package réseau.

Console

Pour créer un package de fonctions à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, choisissez Function packages.
3. Choisissez Créer un package de fonctions.
4. Choisissez Choisir un fichier et téléchargez le package CSAR de votre NF.
5. Choisissez Suivant.
6. Vérifiez les détails du package.
7. Choisissez Créer un package de fonctions.

AWS CLI

Pour créer un package de fonctions à l'aide du AWS CLI

1. Utilisez la create-sol-function-packagecommande pour créer un nouveau package de fonctions :

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

2. Utilisez la commande put-sol-function-package-content pour télécharger le contenu du package de fonctions. Par exemple :

```
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
```

Afficher un package de fonctions dans AWS TNB

Découvrez comment afficher le contenu d'un package de fonctions.

Console

Pour afficher un package de fonctions à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, choisissez Function packages.
3. Utilisez le champ de recherche pour trouver le package de fonctions

AWS CLI

Pour afficher un package de fonctions à l'aide du AWS CLI

1. Utilisez la list-sol-function-packagescommande pour répertorier vos packages de fonctions.

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

2. Utilisez la get-sol-function-packagecommande pour afficher les détails d'un package de fonctions.

```
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
```

Téléchargez un package de fonctions depuis AWS TNB

Découvrez comment télécharger un package de fonctions à partir du catalogue de fonctions réseau AWS TNB.

Console

Pour télécharger un package de fonctions à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation sur le côté gauche de la console, choisissez Function packages.
3. Utilisez le champ de recherche pour trouver le package de fonctions
4. Choisissez le package de fonctions
5. Choisissez Actions, puis Télécharger.

AWS CLI

Pour télécharger un package de fonctions à l'aide du AWS CLI

Utilisez la commande get-sol-function-package-content pour télécharger un package de fonctions.

```
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
```

Supprimer un package de fonctions de AWS TNB

Découvrez comment supprimer un package de fonctions du catalogue de fonctions réseau AWS TNB. Pour supprimer un package de fonctions, celui-ci doit être désactivé.

Console

Pour supprimer un package de fonctions à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, choisissez Function packages.
3. Utilisez le champ de recherche pour trouver le package de fonctions.
4. Choisissez un pack de fonctions.
5. Choisissez Actions, Désactiver .
6. Sélectionnez Actions, Delete (Supprimer).

AWS CLI

Pour supprimer un package de fonctions à l'aide du AWS CLI

1. Utilisez la update-sol-function-packagecommande pour désactiver un package de fonctions.

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

2. Utilisez la delete-sol-function-packagecommande pour supprimer un package de fonctions.

```
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
```

Packages réseau pour AWS TNB

Un package réseau est un fichier .zip au format CSAR (Cloud Service Archive) qui définit les packages de fonctions que vous souhaitez déployer et l'AWSinfrastructure sur laquelle vous souhaitez les déployer.

Tâches

- Créez un package réseau dans AWS TNB
- Afficher un package réseau dans AWS TNB
- Téléchargez un package réseau auprès de AWS TNB
- Supprimer un package réseau de AWS TNB

Créez un package réseau dans AWS TNB

Un package réseau se compose d'un fichier descripteur de service réseau (NSD) (obligatoire) et de tout fichier supplémentaire (facultatif), tel que des scripts spécifiques à vos besoins. Par exemple, si votre package réseau contient plusieurs packages de fonctions, vous pouvez utiliser le NSD pour définir les fonctions réseau qui doivent être exécutées dans certains VPC, sous-réseaux ou clusters Amazon EKS.

Créez un package réseau après avoir créé des packages de fonctions. Une fois que vous avez créé un package réseau, vous devez créer une instance réseau.

Console

Pour créer un package réseau à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, sélectionnez Network packages.
3. Choisissez Créer un package réseau.
4. Choisissez Choisir un fichier et téléchargez votre package CSAR.
5. Choisissez Suivant.
6. Vérifiez les détails du package.
7. Choisissez Créer un package réseau.

AWS CLI

Pour créer un package réseau à l'aide du AWS CLI

1. Utilisez la create-sol-network-packagecommande pour créer un package réseau.

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

2. Utilisez la commande put-sol-network-package-content pour télécharger le contenu du package réseau. Par exemple :

```
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
```

Afficher un package réseau dans AWS TNB

Découvrez comment afficher le contenu d'un package réseau.

Console

Pour afficher un package réseau à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, sélectionnez Network packages.
3. Utilisez le champ de recherche pour trouver le package réseau.

AWS CLI

Pour consulter un package réseau à l'aide du AWS CLI

1. Utilisez la list-sol-network-packagescommande pour répertorier vos packages réseau.

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

2. Utilisez la get-sol-network-packagecommande pour afficher les détails d'un package réseau.

```
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
```

Téléchargez un package réseau auprès de AWS TNB

Découvrez comment télécharger un package réseau à partir du catalogue de services réseau AWS TNB.

Console

Pour télécharger un package réseau à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, sélectionnez Network packages.
3. Utilisez le champ de recherche pour trouver le package réseau
4. Choisissez le package réseau.
5. Choisissez Actions, puis Télécharger.

AWS CLI

Pour télécharger un package réseau à l'aide du AWS CLI

- Utilisez la commande get-sol-network-package-content pour télécharger un package réseau.

```
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
```

Supprimer un package réseau de AWS TNB

Découvrez comment supprimer un package réseau du catalogue de services réseau AWS TNB. Pour supprimer un package réseau, celui-ci doit être dans un état désactivé.

Console

Pour supprimer un package réseau à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, sélectionnez Network packages.
3. Utilisez le champ de recherche pour trouver le package réseau
4. Choisissez un package réseau
5. Choisissez Actions, Désactiver .
6. Sélectionnez Actions, Delete (Supprimer).

AWS CLI

Pour supprimer un package réseau à l'aide du AWS CLI

1. Utilisez la update-sol-network-packagecommande pour désactiver un package réseau.

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

2. Utilisez la delete-sol-network-packagecommande pour supprimer un package réseau.

```
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
```

Instances réseau pour AWS TNB

Une instance réseau est un réseau unique créé dans AWS TNB qui peut être déployé.

Tâches

- Instancier une instance réseau à l'aide de TNB AWS
- Afficher une instance réseau dans AWS TNB
- Mettre à jour une instance réseau dans AWS TNB
- Mettre fin à une instance réseau et la supprimer de AWS TNB

Instancier une instance réseau à l'aide de TNB AWS

Vous créez une instance réseau après avoir créé un package réseau. Une fois que vous avez créé une instance réseau, vous devez l'instancier. Lorsque vous instanciez une instance réseau, AWS TNB déploie les fonctions réseau conformément aux spécifications du descripteur de service réseau.

Console

Pour créer et instancier une instance réseau à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, sélectionnez Networks.
3. Choisissez Create network instance.
4. Entrez un nom et une description pour l'instance, puis choisissez Next.
5. Sélectionnez votre NSD. Vérifiez les informations, puis choisissez Next.
6. Choisissez Create network instance.
7. Choisissez Instancier.
8. Choisissez Instancier le réseau.
9. Actualisez pour suivre l'état de votre instance réseau.

AWS CLI

Pour créer et instancier une instance réseau à l'aide du AWS CLI

1. Utilisez la create-sol-network-instancecommande pour créer une instance réseau.

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

2. Utilisez la instantiate-sol-network-instancecommande pour instancier l'instance réseau.

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

Afficher une instance réseau dans AWS TNB

Découvrez comment afficher une instance réseau.

Console

Pour afficher une instance réseau à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, sélectionnez Network instances.
3. Utilisez le champ de recherche pour trouver l'instance réseau.

AWS CLI

Pour afficher une instance réseau à l'aide du AWS CLI

1. Utilisez la list-sol-network-instancescommande pour répertorier vos instances réseau.

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

2. Utilisez la get-sol-network-instancecommande pour afficher les détails d'une instance réseau.

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

Mettre à jour une instance réseau dans AWS TNB

Découvrez comment mettre à jour une instance réseau.

Console

Pour mettre à jour une instance réseau à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, sélectionnez Networks.
3. Sélectionnez l'ID de l'instance réseau.
4. Dans l'onglet Fonctions, sélectionnez l'instance de fonction à mettre à jour.
5. Choisissez Mettre à jour.
6. Entrez vos annulations de mise à jour pour confirmer la mise à jour.
7. Choisissez Mettre à jour.
8. Actualisez pour suivre l'état de votre instance réseau.

AWS CLI

Utiliser la CLI pour mettre à jour une instance réseau

Utilisez la update-sol-network-instancecommande pour mettre à jour une instance réseau.

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

Mettre fin à une instance réseau et la supprimer de AWS TNB

Pour supprimer une instance réseau, celle-ci doit être dans un état terminé.

Console

Pour mettre fin à une instance réseau et la supprimer à l'aide de la console

1. Ouvrez la console AWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, sélectionnez Networks.
3. Sélectionnez l'ID de l'instance réseau.
4. Sélectionnez Terminer.
5. Lorsque vous êtes invité à confirmer, entrez l'ID et choisissez Terminate.
6. Actualisez pour suivre l'état de votre instance réseau.

7. (Facultatif) Sélectionnez l'instance réseau et choisissez Supprimer.

AWS CLI

Pour mettre fin à une instance réseau et la supprimer à l'aide du AWS CLI

1. Utilisez la terminate-sol-network-instancecommande pour mettre fin à une instance réseau.

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

2. (Facultatif) Utilisez la delete-sol-network-instancecommande pour supprimer une instance réseau.

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

Exploitation du réseau pourAWS TNB

Une opération réseau est toute opération effectuée sur votre réseau, telle que l'instanciation ou la résiliation d'une instance réseau.

Tâches

- Affichage d'opérations réseau d'd'd'd'd'
- Annulation d'opération de réseau Annulation d'

Affichage d'opérations réseau d'd'd'd'd'

Afficher les détails d'une opération réseau, y compris les tâches impliquées dans le fonctionnement du réseau et l'état des tâches.

Console

Pour afficher les opérations réseau à l'aide de la console de la console, pour afficher

1. Ouvrez la consoleAWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, choisissez Network Instances.
3. Utilisez la zone de recherche pour trouver l'instance réseau.
4. Dans l'onglet Déploiements, choisissez Network Operation.

AWS CLI

Pour visualiser une opération réseau à l'aide duAWS CLI

1. Utilisez la list-sol-network-operationscommande pour répertorier toutes les opérations réseau.

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

2. Utilisez la get-sol-network-operationcommande pour afficher les détails d'une opération réseau.

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

Annulation d'opération de réseau Annulation d'

Découvrez comment annuler l'opération du réseau.

Console

Pour annuler l'opération réseau à l'aide de la console de la console, pour annuler

1. Ouvrez la consoleAWS TNB à l'adresse https://console.aws.amazon.com/tnb/.
2. Dans le volet de navigation, choisissez Networks.
3. Sélectionnez l'ID du réseau pour ouvrir sa page de détails.
4. Dans l'onglet Déploiements, choisissez Network Operation.
5. Choisissez Annuler l'opération.

AWS CLI

Pour annuler une opération réseau à l'aide duAWS CLI

Utilisez la cancel-sol-network-operationcommande pour annuler une opération réseau.

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

Référence TOSCA pourAWS TNB

La spécification de topologie et d'orchestration pour les applications cloud (TOSCA) est une syntaxe déclarative utilisée par les CSP pour décrire la topologie des services Web basés sur le cloud, leurs composants, leurs relations et les processus qui les gèrent. Les CSP décrivent les points de connexion, les liens logiques entre les points de connexion et les politiques telles que l'affinité et la sécurité dans un modèle TOSCA. Les CSP téléchargent ensuite le modèle surAWS TNB, qui synthétise les ressources nécessaires pour établir un réseau 5G fonctionnel dans les zones deAWS disponibilité.

Table des matières

Modèle VNFD

Définit un modèle de descripteur de fonction réseau virtuel (VNFD).

Syntaxe

```
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
```

Modèle de topologie

`node_templates`

La TOSCAAWSNœuds. Les nœuds possibles sont les suivants :

- AWS.VNF
- AWS. Artefacts. Casque

AWS.VNF

Définit unAWSnœud de fonction réseau virtuelle (VNF).

Syntaxe

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

Propriétés

`descriptor_id`

L'UUID du descripteur.

Obligatoire : oui

Type : String

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

`descriptor_version`

La version du VNFD.

Obligatoire : oui

Type : String

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

`descriptor_name`

Le nom du descripteur.

Obligatoire : oui

Type : String

`provider`

L'auteur du VNFD.

Obligatoire : oui

Type : String

Prérequis

`helm`

Le répertoire Helm qui définit les artefacts du conteneur. Il s'agit d'une référence àAWS. Artefacts. Casque.

Obligatoire : oui

Type : String

Exemple

```
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

Définit unAWSNœud de barre.

Syntaxe

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

Propriétés

`implementation`

Le répertoire local qui contient le graphique Helm dans le package CSAR.

Obligatoire : oui

Type : String

Exemple

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

Modèle NSD

Définit un modèle de descripteur de service réseau (NSD).

Syntaxe

```
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
```

Utilisation de paramètres définis

Lorsque vous souhaitez transmettre dynamiquement un paramètre, tel que le bloc CIDR pour le nœud VPC, vous pouvez utiliser `{ get_input: input-parameter-name }` la syntaxe et définir les paramètres dans le modèle NSD. Réutilisez ensuite le paramètre dans le même modèle NSD.

L'exemple suivant montre comment définir et utiliser des paramètres :

```
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 }
```

Importation VNFD

`descriptor_id`

L'UUID du descripteur.

Obligatoire : oui

Type : String

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

`namespace`

Le nom unique.

Obligatoire : oui

Type : String

Modèle de topologie

`node_templates`

Les AWS nœuds TOSCA possibles sont les suivants :

- AWS N.S.
- AWS.Computer.EKS
- AWS.Computer.EKS. AuthRole
- AWS.Computer.EKS ManagedNode
- AWS.Computer.EKS SelfManagedNode
- AWS.Calculez. PlacementGroup
- AWS.Calculez. UserData
- AWS.Réseautage. SecurityGroup
- AWS.Réseautage. SecurityGroupEgressRule

- AWS.Réseautage. SecurityGroupIngressRule
- AWS.Ressource.Importer
- AWS.Networking.eni
- AWS.HookExecution
- AWS.Réseautage. InternetGateway
- AWS.Réseautage. RouteTable
- AWS.Réseau.Sous-réseau
- AWS.Deployment.VNF Déploiement
- AWS.Réseau.VPC
- AWS Passerelle .Networking.NAT
- AWS.Mise en réseau.Route

AWS N.S.

Définit un nœud de service AWS réseau (NS).

Syntaxe

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

Propriétés

`descriptor_id`

L'UUID du descripteur.

Obligatoire : oui

Type : String

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

`descriptor_version`

Version du NSD.

Obligatoire : oui

Type : String

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

`descriptor_name`

Le nom du descripteur.

Obligatoire : oui

Type : String

Exemple

```
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.Computer.EKS

Indiquez le nom du cluster, la version de Kubernetes souhaitée et un rôle permettant au plan de contrôle Kubernetes de gérer les ressources requises pour vos NF. AWS Les plugins CNI (Multus Container Network Interface) sont activés. Vous pouvez associer plusieurs interfaces réseau et appliquer une configuration réseau avancée aux fonctions réseau basées sur Kubernetes. Vous spécifiez également l'accès au point de terminaison du cluster et les sous-réseaux de votre cluster.

Syntaxe

```
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
```

Fonctionnalités

`multus`

Facultatif. Propriétés qui définissent l'utilisation de l'interface réseau de conteneurs (CNI) Multus.

Si vous incluez`multus`, spécifiez les `multus_role` propriétés `enabled` et.

`enabled`

Indique si la fonctionnalité Multus par défaut est activée.

Obligatoire : oui

Type : booléen

`multus_role`

Le rôle de la gestion de l'interface réseau Multus.

Obligatoire : oui

Type : String

`ebs_csi`

Propriétés qui définissent le pilote Amazon EBS Container Storage Interface (CSI) installé dans le cluster Amazon EKS.

Activez ce plugin pour utiliser les nœuds autogérés Amazon EKS sur AWS Outposts les Zones AWS Locales ou Régions AWS. Pour plus d'informations, consultez le pilote Amazon Elastic Block Store CSI dans le guide de l'utilisateur Amazon EKS.

`enabled`

Indique si le pilote Amazon EBS CSI par défaut est installé.

Obligatoire : non

Type : booléen

`version`

Version du module complémentaire de pilote Amazon EBS CSI. La version doit correspondre à l'une des versions renvoyées par l'DescribeAddonVersionsaction. Pour plus d'informations, consultez DescribeAddonVersionsle manuel Amazon EKS API Reference

Obligatoire : non

Type : chaîne

Propriétés

`version`

Version de Kubernetes pour le cluster. AWS Telco Network Builder prend en charge les versions 1.23 à 1.29 de Kubernetes.

Obligatoire : oui

Type : String

Valeurs possibles : 1.23 | 1.24 | 1.25 | 1.26 | 1.27 | 1.28 | 1.29

`access`

L'accès au point de terminaison du cluster.

Obligatoire : oui

Type : String

Valeurs possibles : `PRIVATE` | `PUBLIC` | `ALL`

`cluster_role`

Le rôle de la gestion des clusters.

Obligatoire : oui

Type : String

`tags`

Balises à associer à la ressource.

Obligatoire : non

Type: liste

`ip_family`

Indique la famille d'adresses IP pour les adresses de service et de pod dans le cluster.

Valeur autorisée :`IPv4`, `IPv6`

Valeur par défaut : `IPv4`

Obligatoire : non

Type : chaîne

Prérequis

`subnets`

Un nœud AWS.Networking.Subnet.

Obligatoire : oui

Type: liste

Exemple

```
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.Computer.EKS. AuthRole

An vous AuthRole permet d'ajouter des rôles IAM au cluster Amazon EKS `aws-auth` `ConfigMap` afin que les utilisateurs puissent accéder au cluster Amazon EKS à l'aide d'un rôle IAM.

Syntaxe

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

Propriétés

`role_mappings`

Liste des mappages qui définissent les rôles IAM qui doivent être ajoutés au cluster Amazon EKS. `aws-auth` `ConfigMap`

`arn`

ARN du rôle IAM.

Obligatoire : oui

Type : String

`groups`

Groupes Kubernetes à attribuer au rôle défini dans. `arn`

Obligatoire : non

Type: liste

Prérequis

`clusters`

Un nœud AWS.Compute.EKS.

Obligatoire : oui

Type: liste

Exemple

```
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.Computer.EKS ManagedNode

AWS TNB prend en charge les groupes de nœuds gérés par EKS pour automatiser le provisionnement et la gestion du cycle de vie des nœuds (instances Amazon EC2) pour les clusters Amazon EKS Kubernetes. Pour créer un groupe de nœuds EKS, vous devez choisir les Amazon Machine Images (AMI) pour les nœuds de travail de votre cluster en fournissant soit l'ID de l'AMI, soit le type d'AMI. Vous fournissez également une paire de clés Amazon EC2 pour l'accès SSH et les propriétés de dimensionnement de votre groupe de nœuds. Votre groupe de nœuds doit être associé à un cluster EKS. Vous devez fournir les sous-réseaux pour les nœuds de travail.

Vous pouvez éventuellement associer des groupes de sécurité, des étiquettes de nœuds et un groupe de placement à votre groupe de nœuds.

Syntaxe

```
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
```

Fonctionnalités

`compute`

Propriétés qui définissent les paramètres informatiques du groupe de nœuds gérés par Amazon EKS, tels que les types d'instances Amazon EC2 et les AMI d'instance Amazon EC2.

`ami_type`

Type d'AMI compatible avec Amazon EKS.

Obligatoire : oui

Type : String

Valeurs possibles : `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`

ID de l'AMI.

Obligatoire : non

Type : chaîne

> **Note**
>
> Si `ami_type` les deux `ami_id` sont spécifiés dans le modèle, AWS TNB n'utilisera que la `ami_id` valeur pour créer`EKSManagedNode`.

`instance_types`

Taille de l'instance.

Obligatoire : oui

Type: liste

`key_pair`

La paire de clés EC2 pour activer l'accès SSH.

Obligatoire : oui

Type : String

`root_volume_encryption`

Active le chiffrement Amazon EBS pour le volume racine Amazon EBS. Si cette propriété n'est pas fournie, AWS TNB chiffre les volumes racine Amazon EBS par défaut.

Obligatoire : non

Valeur par défaut : true

Type : booléen

`root_volume_encryption_key_arn`

L'ARN de la AWS KMS clé. AWS TNB prend en charge l'ARN clé standard, l'ARN clé multirégional et l'ARN alias.

Obligatoire : non

Type : chaîne

> **Note**
>
> - Si `root_volume_encryption` c'est faux, ne l'incluez pas`root_volume_encryption_key_arn`.
> - AWS TNB prend en charge le chiffrement du volume racine des AMI soutenues par Amazon EBS.
> - Si le volume racine de l'AMI est déjà chiffré, vous devez inclure le AWS TNB `root_volume_encryption_key_arn` pour le rechiffrer.
> - Si le volume racine de l'AMI n'est pas chiffré, AWS TNB utilise le `root_volume_encryption_key_arn` pour chiffrer le volume racine.
>
>   Si vous ne l'incluez pas`root_volume_encryption_key_arn`, AWS TNB utilise la clé par défaut fournie par AWS Key Management Service pour chiffrer le volume racine.

- AWS TNB ne déchiffre pas une AMI chiffrée.

`scaling`

Propriétés qui définissent les paramètres de dimensionnement pour le groupe de nœuds géré par Amazon EKS, tels que le nombre souhaité d'instances Amazon EC2 et le nombre minimum et maximum d'instances Amazon EC2 dans le groupe de nœuds.

`desired_size`

Le nombre d'instances qu'il contient NodeGroup.

Obligatoire : oui

Type : entier

`min_size`

Le nombre minimum d'instances dans ce cas NodeGroup.

Obligatoire : oui

Type : entier

`max_size`

Le nombre maximum d'instances dans ce cas NodeGroup.

Obligatoire : oui

Type : entier

Propriétés

`node_role`

L'ARN du rôle IAM attaché à l'instance Amazon EC2.

Obligatoire : oui

Type : String

`tags`

Les balises à associer à la ressource.

Obligatoire : non

Type: liste

Prérequis

`cluster`

Un nœud AWS.Compute.EKS.

Obligatoire : oui

Type : String

`subnets`

Un nœud AWS.Networking.Subnet.

Obligatoire : oui

Type: liste

`network_interfaces`

Un nœud AWS.Networking.ENI. Assurez-vous que les interfaces réseau et les sous-réseaux sont définis sur la même zone de disponibilité, sinon l'instanciation échouera.

Lorsque vous définissez`network_interfaces`, AWS TNB obtient l'autorisation relative aux ENI auprès de la `multus_role` propriété si vous avez inclus la `multus` propriété dans le nœud AWS.compute.EKS. Sinon, AWS TNB obtient l'autorisation relative aux ENI à partir de la propriété node_role.

Obligatoire : non

Type: liste

`security_groups`

Un AWS.Networking. SecurityGroupnœud.

Obligatoire : non

Type: liste

`placement_group`

Un tosca.nodes.AWS.Calculez. PlacementGroupnœud.

Obligatoire : non

Type : chaîne

`user_data`

Un tosca.nodes.AWS.Calculez. UserDataréférence de nœud. Un script de données utilisateur est transmis aux instances Amazon EC2 lancées par le groupe de nœuds gérés. Ajoutez les autorisations requises pour exécuter des données utilisateur personnalisées au node_role transmis au groupe de nœuds.

Obligatoire : non

Type : chaîne

`labels`

Liste des étiquettes de nœuds. L'étiquette d'un nœud doit avoir un nom et une valeur. Créez une étiquette en utilisant les critères suivants :

- Le nom et la valeur doivent être séparés par=.
- Le nom et la valeur peuvent chacun comporter jusqu'à 63 caractères.
- L'étiquette peut inclure des lettres (A-Z, a-z), des chiffres (0-9) et les caractères suivants : `[-, _, ., *, ?]`
- Le nom et la valeur doivent commencer et se terminer par un * caractère alphanumérique ou. ?

Par exemple, `myLabelName1=*NodeLabelValue1`

Obligatoire : non

Type: liste

Exemple

```
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.Computer.EKS SelfManagedNode

AWS TNB prend en charge les nœuds autogérés Amazon EKS pour automatiser le provisionnement et la gestion du cycle de vie des nœuds (instances Amazon EC2) pour les clusters Amazon EKS Kubernetes. Pour créer un groupe de nœuds Amazon EKS, vous devez choisir les Amazon Machine Images (AMI) pour vos nœuds de travail de cluster en fournissant l'ID de l'AMI. Vous pouvez

éventuellement fournir une paire de clés Amazon EC2 pour l'accès SSH. Vous devez également indiquer le type d'instance ainsi que les tailles souhaitées, minimales et maximales. Votre groupe de nœuds doit être associé à un cluster Amazon EKS. Vous devez fournir les sous-réseaux pour les nœuds de travail.

Vous pouvez éventuellement associer des groupes de sécurité, des étiquettes de nœuds et un groupe de placement à votre groupe de nœuds.

Syntaxe

```
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
```

Fonctionnalités

compute

Propriétés qui définissent les paramètres de calcul pour les nœuds autogérés Amazon EKS, tels que les types d'instances Amazon EC2 et les AMI d'instance Amazon EC2.

`ami_id`

ID d'AMI utilisé pour lancer l'instance. AWS TNB prend en charge les instances qui exploitent IMDSv2. Pour plus d'informations, consultez Version IMDS.

Obligatoire : oui

Type : String

`instance_type`

Taille de l'instance.

Obligatoire : oui

Type : String

`key_pair`

La paire de clés Amazon EC2 pour activer l'accès SSH.

Obligatoire : oui

Type : String

`root_volume_encryption`

Active le chiffrement Amazon EBS pour le volume racine Amazon EBS. Si cette propriété n'est pas fournie, AWS TNB chiffre les volumes racine Amazon EBS par défaut.

Obligatoire : non

Valeur par défaut : true

Type : booléen

`root_volume_encryption_key_arn`

L'ARN de la AWS KMS clé. AWS TNB prend en charge l'ARN clé standard, l'ARN clé multirégional et l'ARN alias.

Obligatoire : non

Type : chaîne

> **Note**
>
> - Si `root_volume_encryption` c'est faux, ne l'incluez pas`root_volume_encryption_key_arn`.
> - AWS TNB prend en charge le chiffrement du volume racine des AMI soutenues par Amazon EBS.
> - Si le volume racine de l'AMI est déjà chiffré, vous devez inclure le AWS TNB `root_volume_encryption_key_arn` pour le rechiffrer.
> - Si le volume racine de l'AMI n'est pas chiffré, AWS TNB utilise le `root_volume_encryption_key_arn` pour chiffrer le volume racine.
>
> Si vous ne l'incluez pas`root_volume_encryption_key_arn`, AWS TNB l'utilise AWS Managed Services pour chiffrer le volume racine.
> - AWS TNB ne déchiffre pas une AMI chiffrée.

`scaling`

Propriétés qui définissent les paramètres de dimensionnement pour les nœuds autogérés Amazon EKS, tels que le nombre souhaité d'instances Amazon EC2 et le nombre minimum et maximum d'instances Amazon EC2 dans le groupe de nœuds.

`desired_size`

Le nombre d'instances qu'il contient NodeGroup.

Obligatoire : oui

Type : entier

`min_size`

Le nombre minimum d'instances dans ce cas NodeGroup.

Obligatoire : oui

Type : entier

`max_size`

Le nombre maximum d'instances dans ce cas NodeGroup.

Obligatoire : oui

Type : entier

Propriétés

`node_role`

L'ARN du rôle IAM attaché à l'instance Amazon EC2.

Obligatoire : oui

Type : String

`tags`

Les balises à associer à la ressource. Les balises seront propagées aux instances créées par la ressource.

Obligatoire : non

Type: liste

Prérequis

`cluster`

Un nœud AWS.Compute.EKS.

Obligatoire : oui

Type : String

`subnets`

Un nœud AWS.Networking.Subnet.

Obligatoire : oui

Type: liste

`network_interfaces`

Un nœud AWS.Networking.ENI. Assurez-vous que les interfaces réseau et les sous-réseaux sont définis sur la même zone de disponibilité, sinon l'instanciation échouera.

Lorsque vous définissez`network_interfaces`, AWS TNB obtient l'autorisation relative aux ENI auprès de la `multus_role` propriété si vous avez inclus la `multus` propriété dans le nœud AWS.compute.EKS. Sinon, AWS TNB obtient l'autorisation relative aux ENI à partir de la propriété node_role.

Obligatoire : non

Type: liste

`security_groups`

Un AWS.Networking. SecurityGroupnœud.

Obligatoire : non

Type: liste

`placement_group`

Un tosca.nodes.AWS.Calculez. PlacementGroupnœud.

Obligatoire : non

Type : chaîne

`user_data`

Un tosca.nodes.AWS.Calculez. UserDataréférence de nœud. Un script de données utilisateur est transmis aux instances Amazon EC2 lancées par le groupe de nœuds autogéré. Ajoutez les autorisations requises pour exécuter des données utilisateur personnalisées au node_role transmis au groupe de nœuds.

Obligatoire : non

Type : chaîne

`labels`

Liste des étiquettes de nœuds. L'étiquette d'un nœud doit avoir un nom et une valeur. Créez une étiquette en utilisant les critères suivants :

- Le nom et la valeur doivent être séparés par=.
- Le nom et la valeur peuvent chacun comporter jusqu'à 63 caractères.
- L'étiquette peut inclure des lettres (A-Z, a-z), des chiffres (0-9) et les caractères suivants : `[-, _, ., *, ?]`

- Le nom et la valeur doivent commencer et se terminer par un * caractère alphanumérique ou. ?

Par exemple, `myLabelName1=*NodeLabelValue1`

Obligatoire : non

Type: liste

Exemple

```
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.Calculez. PlacementGroup

Un PlacementGroup nœud prend en charge différentes stratégies pour placer des instances Amazon EC2.

Lorsque vous lancez une nouvelle instance Amazon EC2, le service Amazon EC2 tente de placer l'instance de telle sorte que toutes vos instances soient réparties sur le matériel sous-jacent afin de minimiser les défaillances corrélées. Vous pouvez utiliser des groupes de placement pour influencer le placement d'un groupe d'instances interdépendantes afin de répondre aux besoins de votre charge de travail.

Syntaxe

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

Propriétés

`strategy`

La stratégie à utiliser pour placer les instances Amazon EC2.

Obligatoire : oui

Type : String

Valeurs possibles : CLUSTER | PARTITION | SPREAD_HOST | SPREAD_RACK

- CLUSTER : regroupe les instances à proximité les unes des autres au sein d'une zone de disponibilité. Cette stratégie permet aux charges de travail d'atteindre les performances réseau à faible latence nécessaires aux node-to-node communications étroitement couplées, typiques des applications de calcul haute performance (HPC).
- PARTITION : répartit vos instances sur des partitions logiques de telle sorte que les groupes d'instances d'une partition ne partagent pas le matériel sous-jacent avec des groupes

d'instances situés dans différentes partitions. Cette stratégie est généralement utilisée par les grandes charges de travail distribuées et répliquées telles que Hadoop, Cassandra, et Kafka.

- SPREAD_RACK — place un petit groupe d'instances sur un matériel sous-jacent distinct afin de réduire les défaillances corrélées.
- SPREAD_HOST : utilisé uniquement avec les groupes de placement Outpost. Place un petit groupe d'instances sur un matériel sous-jacent distinct afin de réduire les défaillances corrélées.

`partition_count`

Nombre de partitions.

Obligatoire : obligatoire uniquement lorsque `strategy` ce paramètre est défini sur`PARTITION`.

Type : entier

Valeurs possibles : 1 | 2 | 3 | 4 | 5 | 6 | 7

`tags`

Les balises que vous pouvez associer à la ressource du groupe de placement.

Obligatoire : non

Type: liste

Exemple

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

AWS.Calculez. UserData

AWS TNB prend en charge le lancement d'instances Amazon EC2 avec des données utilisateur personnalisées, via UserData le nœud du Network Service Descriptor (NSD). Pour plus d'informations sur les données utilisateur personnalisées, consultez la section Données utilisateur et scripts shell dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Lors de l'instanciation du réseau, AWS TNB fournit l'enregistrement de l'instance Amazon EC2 au cluster via un script de données utilisateur. Lorsque des données utilisateur personnalisées sont également fournies, AWS TNB fusionne les deux scripts et les transmet en tant que script multimime à Amazon EC2. Le script de données utilisateur personnalisé est exécuté avant le script d'enregistrement Amazon EKS.

Pour utiliser des variables personnalisées dans le script de données utilisateur, ajoutez un point d'exclamation `!` après l'accolade ouverte. `{` Par exemple, pour l'utiliser `MyVariable` dans le script, entrez : `{!MyVariable}`

> **Note**
>
> - AWS TNB prend en charge les scripts de données utilisateur d'une taille maximale de 7 Ko.
> - AWS TNB étant utilisé AWS CloudFormation pour traiter et afficher le script de `multimime` données utilisateur, assurez-vous que le script respecte toutes les règles. AWS CloudFormation

Syntaxe

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

Propriétés

`implementation`

Le chemin relatif vers la définition du script de données utilisateur. Le format doit être le suivant : `./scripts/script_name.sh`

Obligatoire : oui

Type : String

`content_type`

Type de contenu du script de données utilisateur.

Obligatoire : oui

Type : String

Valeurs possibles : `x-shellscript`

Exemple

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

AWS.Réseautage. SecurityGroup

AWS TNB prend en charge les groupes de sécurité pour automatiser le provisionnement des groupes de sécurité Amazon EC2 que vous pouvez associer aux groupes de nœuds du cluster Amazon EKS Kubernetes.

Syntaxe

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

Propriétés

`description`

Description du groupe de sécurité. Vous pouvez utiliser jusqu'à 255 caractères pour décrire le groupe. Vous ne pouvez inclure que des lettres (A-Z et a-z), des chiffres (0-9), des espaces et les caractères spéciaux suivants : ._- :/() #, @ [] +=& ; {} ! $*

Obligatoire : oui

Type : String

`name`

Nom du groupe de sécurité. Vous pouvez utiliser jusqu'à 255 caractères pour le nom. Vous ne pouvez inclure que des lettres (A-Z et a-z), des chiffres (0-9), des espaces et les caractères spéciaux suivants : ._- :/() #, @ [] +=& ; {} ! $*

Obligatoire : oui

Type : String

`tags`

Les balises que vous pouvez associer à la ressource du groupe de sécurité.

Obligatoire : non

Type: liste

Prérequis

`vpc`

Un nœud AWS.Networking.VPC.

Obligatoire : oui

Type : String

Exemple

```
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.Réseautage. SecurityGroupEgressRule

AWS TNB prend en charge les règles de sortie des groupes de sécurité afin d'automatiser le provisionnement des règles de sortie des groupes de sécurité Amazon EC2 qui peuvent être associées à .Networking. AWS SecurityGroup. Notez que vous devez fournir un cidr_ip/destination_security_group/destination_prefix_list comme destination pour le trafic de sortie.

Syntaxe

```
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
```

Propriétés

`cidr_ip`

La plage d'adresses IPv4 au format CIDR. Vous devez spécifier une plage CIDR qui autorise le trafic sortant.

Obligatoire : non

Type : chaîne

`cidr_ipv6`

La plage d'adresses IPv6 au format CIDR, pour le trafic sortant. Vous devez spécifier un groupe de sécurité de destination (`destination_security_group` ou `destination_prefix_list`) ou une plage d'adresses CIDR (`cidr_ip` ou `cidr_ipv6`).

Obligatoire : non

Type : chaîne

`description`

Description d'une règle de groupe de sécurité pour le trafic entrant (sortant). Vous pouvez utiliser jusqu'à 255 caractères pour décrire la règle.

Obligatoire : non

Type : chaîne

`destination_prefix_list`

L'ID de liste de préfixes d'une liste de préfixes gérée par Amazon VPC existante. Il s'agit de la destination à partir des instances de groupes de nœuds associées au groupe de sécurité. Pour plus d'informations sur les listes de préfixes gérées, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

Obligatoire : non

Type : chaîne

`from_port`

Si le protocole est TCP ou UDP, il s'agit du début de la plage de ports. Si le protocole est ICMP ou ICMPv6, il s'agit du numéro de type. La valeur -1 indique tous les types ICMP/ICMPv6. Si vous spécifiez tous les types ICMP/ICMPv6, vous devez spécifier tous les codes ICMP/ICMPv6.

Obligatoire : non

Type : entier

`ip_protocol`

Nom du protocole IP (tcp, udp, icmp, icmpv6) ou numéro de protocole. Utilisez -1 pour spécifier tous les protocoles. Lorsque vous autorisez les règles du groupe de sécurité, la spécification de -1 ou d'un numéro de protocole autre que TCP, UDP, ICMP ou ICMPv6 autorise le trafic sur tous les ports, quelle que soit la plage de ports que vous spécifiez. Pour TCP, UDP et ICMP, vous devez spécifier une plage de ports. Pour icmpv6, la plage de ports est facultative ; si vous omettez la plage de ports, le trafic est autorisé pour tous les types et codes.

Obligatoire : oui

Type : String

`to_port`

Si le protocole est TCP ou UDP, il s'agit de la fin de la plage de ports. Si le protocole est ICMP ou ICMPv6, il s'agit du code. La valeur -1 indique tous les codes ICMP/ICMPv6. Si vous spécifiez tous les types ICMP/ICMPv6, vous devez spécifier tous les codes ICMP/ICMPv6.

Obligatoire : non

Type : entier

Prérequis

`security_group`

ID du groupe de sécurité auquel cette règle doit être ajoutée.

Obligatoire : oui

Type : String

`destination_security_group`

ID ou référence TOSCA du groupe de sécurité de destination vers lequel le trafic de sortie est autorisé.

Obligatoire : non

Type : chaîne

Exemple

```
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.Réseautage. SecurityGroupIngressRule

AWS TNB prend en charge les règles d'entrée des groupes de sécurité afin d'automatiser le provisionnement des règles d'entrée des groupes de sécurité Amazon EC2 qui peuvent être associées à .Networking. AWS SecurityGroup. Notez que vous devez fournir un cidr_ip/source_security_group/source_prefix_list comme source pour le trafic entrant.

Syntaxe

```
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
```

Propriétés

`cidr_ip`

La plage d'adresses IPv4 au format CIDR. Vous devez spécifier une plage CIDR qui autorise le trafic entrant.

Obligatoire : non

Type : chaîne

`cidr_ipv6`

La plage d'adresses IPv6 au format CIDR, pour le trafic entrant. Vous devez spécifier un groupe de sécurité source (`source_security_group` ou `source_prefix_list`) ou une plage d'adresses CIDR (`cidr_ip` ou `cidr_ipv6`).

Obligatoire : non

Type : chaîne

`description`

Description d'une règle de groupe de sécurité d'entrée (entrante). Vous pouvez utiliser jusqu'à 255 caractères pour décrire la règle.

Obligatoire : non

Type : chaîne

`source_prefix_list`

L'ID de liste de préfixes d'une liste de préfixes gérée par Amazon VPC existante. Il s'agit de la source à partir de laquelle les instances du groupe de nœuds associées au groupe de sécurité seront autorisées à recevoir du trafic. Pour plus d'informations sur les listes de préfixes gérées, consultez la section Listes de préfixes gérées dans le guide de l'utilisateur Amazon VPC.

Obligatoire : non

Type : chaîne

`from_port`

Si le protocole est TCP ou UDP, il s'agit du début de la plage de ports. Si le protocole est ICMP ou ICMPv6, il s'agit du numéro de type. La valeur -1 indique tous les types ICMP/ICMPv6. Si vous spécifiez tous les types ICMP/ICMPv6, vous devez spécifier tous les codes ICMP/ICMPv6.

Obligatoire : non

Type : entier

`ip_protocol`

Nom du protocole IP (tcp, udp, icmp, icmpv6) ou numéro de protocole. Utilisez -1 pour spécifier tous les protocoles. Lorsque vous autorisez les règles du groupe de sécurité, la spécification de -1 ou d'un numéro de protocole autre que TCP, UDP, ICMP ou ICMPv6 autorise le trafic sur tous les ports, quelle que soit la plage de ports que vous spécifiez. Pour TCP, UDP et ICMP, vous devez spécifier une plage de ports. Pour icmpv6, la plage de ports est facultative ; si vous omettez la plage de ports, le trafic est autorisé pour tous les types et codes.

Obligatoire : oui

Type : String

`to_port`

Si le protocole est TCP ou UDP, il s'agit de la fin de la plage de ports. Si le protocole est ICMP ou ICMPv6, il s'agit du code. La valeur -1 indique tous les codes ICMP/ICMPv6. Si vous spécifiez tous les types ICMP/ICMPv6, vous devez spécifier tous les codes ICMP/ICMPv6.

Obligatoire : non

Type : entier

Prérequis

`security_group`

ID du groupe de sécurité auquel cette règle doit être ajoutée.

Obligatoire : oui

Type : String

`source_security_group`

ID ou référence TOSCA du groupe de sécurité source à partir duquel le trafic entrant doit être autorisé.

Obligatoire : non

Type : chaîne

Exemple

```
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.Ressource.Importer

Vous pouvez importer les AWS ressources suivantes dans AWS TNB :

- VPC
- Sous-réseau
- Table de routage
- Internet Gateway
- Security Group

Syntaxe

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

Propriétés

`resource_type`

Type de ressource importé dans AWS TNB.

Obligatoire : non

Type: liste

`resource_id`

ID de ressource importé dans AWS TNB.

Obligatoire : non

Type: liste

Exemple

```
SampleImportedVPC
  type: tosca.nodes.AWS.Resource.Import
```

```
  properties:
    resource_type: "tosca.nodes.AWS.Networking.VPC"
    resource_id: "vpc-123456"
```

AWS.Networking.eni

Une interface réseau est un composant réseau logique d'un VPC qui représente une carte réseau virtuelle. Une adresse IP est attribuée à une interface réseau automatiquement ou manuellement en fonction de son sous-réseau. Après avoir déployé une instance Amazon EC2 dans un sous-réseau, vous pouvez y associer une interface réseau ou détacher une interface réseau de cette instance Amazon EC2 et la rattacher à une autre instance Amazon EC2 de ce sous-réseau. L'index de l'appareil identifie la position dans l'ordre de fixation.

Syntaxe

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

Propriétés

`device_index`

L'indice de l'appareil doit être supérieur à zéro.

Obligatoire : oui

Type : entier

`source_dest_check`

Indique si l'interface réseau effectue la vérification de la source/de la destination. La valeur `true` signifie que la vérification est activée, tandis que la valeur `false` signifie qu'elle est désactivée.

Valeur autorisée : vrai, faux

Valeur par défaut : true

Obligatoire : non

Type : booléen

`tags`

Les balises à associer à la ressource.

Obligatoire : non

Type: liste

Prérequis

`subnet`

Un nœud AWS.Networking.Subnet.

Obligatoire : oui

Type : String

`security_groups`

Un AWS.Networking. SecurityGroupnœud.

Obligatoire : non

Type : chaîne

Exemple

```
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 hook de cycle de vie vous permet d'exécuter vos propres scripts dans le cadre de votre infrastructure et de l'instanciation de votre réseau.

Syntaxe

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

Fonctionnalités

`execution`

Propriétés du moteur d'exécution du hook qui exécute les scripts du hook.

`type`

Type de moteur d'exécution du hook.

Obligatoire : non

Type : chaîne

Valeurs possibles : `CODE_BUILD`

Prérequis

`definition`

Un AWS. HookDefinition.Nœud Bash.

Obligatoire : oui

Type : String

`vpc`

Un nœud AWS.Networking.VPC.

Obligatoire : oui

Type : String

Exemple

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

AWS.Réseautage. InternetGateway

Définit un nœud AWS Internet Gateway.

Syntaxe

```
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
```

Fonctionnalités

routing

Propriétés qui définissent la connexion de routage au sein du VPC. Vous devez inclure la `ipv6_dest_cidr` propriété `dest_cidr` ou.

`dest_cidr`

Bloc d'adresse CIDR IPv4 utilisé pour la correspondance de destination. Cette propriété est utilisée pour créer un itinéraire dans `RouteTable` et sa valeur est utilisée comme`DestinationCidrBlock`.

Obligatoire : Non si vous avez inclus la `ipv6_dest_cidr` propriété.

Type : chaîne

`ipv6_dest_cidr`

Bloc d'adresse CIDR IPv6 utilisé pour la correspondance de destination.

Obligatoire : Non si vous avez inclus la `dest_cidr` propriété.

Type : chaîne

Propriétés

`tags`

Les balises à associer à la ressource.

Obligatoire : non

Type: liste

`egress_only`

Propriété spécifique à IPv6. Indique si la passerelle Internet est uniquement destinée à la communication de sortie ou non. Lorsque `egress_only` c'est vrai, vous devez définir la `ipv6_dest_cidr` propriété.

Obligatoire : non

Type : booléen

Prérequis

`vpc`

Un nœud AWS.Networking.VPC.

Obligatoire : oui

Type : String

`route_table`

Un AWS.Networking. RouteTablenœud.

Obligatoire : oui

Type : String

Exemple

```
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.Réseautage. RouteTable

Une table de routage contient un ensemble de règles, appelées routes, qui déterminent la direction du trafic réseau provenant des sous-réseaux de votre VPC ou de votre passerelle. Vous devez associer une table de routage à un VPC.

Syntaxe

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

Propriétés

`tags`

Balises à associer à la ressource.

Obligatoire : non

Type: liste

Prérequis

`vpc`

Un nœud AWS.Networking.VPC.

Obligatoire : oui

Type : String

Exemple

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

```
      - "Environment=Testing"
  requirements:
    vpc: SampleVPC
```

AWS.Réseau.Sous-réseau

Un sous-réseau est une plage d'adresses IP de votre VPC, qui doit résider entièrement dans une seule zone de disponibilité. Vous devez spécifier un VPC, un bloc CIDR, une zone de disponibilité et une table de routage pour votre sous-réseau. Vous devez également définir si votre sous-réseau est privé ou public.

Syntaxe

```
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
```

Propriétés

`type`

Indique si les instances qui sont lancées dans ce sous-réseau reçoivent une adresse IPv4 publique.

Obligatoire : oui

Type : String

Valeurs possibles : `PUBLIC` | `PRIVATE`

`availability_zone`

Zone de disponibilité du sous-réseau. Ce champ prend en charge les zones de AWS disponibilité au sein d'une AWS région, par exemple `us-west-2` (USA Ouest (Oregon)). Il prend également

en charge les zones AWS locales au sein de la zone de disponibilité, par exemple`us-west-2-lax-1a`.

Obligatoire : oui

Type : String

`cidr_block`

Le bloc CIDR pour le sous-réseau.

Obligatoire : non

Type : chaîne

`ipv6_cidr_block`

Le bloc CIDR utilisé pour créer le sous-réseau IPv6. Si vous incluez cette propriété, ne l'incluez pas`ipv6_cidr_block_suffix`.

Obligatoire : non

Type : chaîne

`ipv6_cidr_block_suffix`

Suffixe hexadécimal à 2 chiffres du bloc d'adresse CIDR IPv6 pour le sous-réseau créé via Amazon VPC. Utilisez le format suivant : *`2-digit hexadecimal`*`::/`*`subnetMask`*

Si vous incluez cette propriété, ne l'incluez pas`ipv6_cidr_block`.

Obligatoire : non

Type : chaîne

`outpost_arn`

L'ARN dans AWS Outposts lequel le sous-réseau sera créé. Ajoutez cette propriété au modèle NSD si vous souhaitez lancer des nœuds autogérés Amazon EKS sur. AWS Outposts Pour plus d'informations, consultez Amazon EKS AWS Outposts dans le guide de l'utilisateur Amazon EKS.

Si vous ajoutez cette propriété au modèle NSD, vous devez définir la valeur de la `availability_zone` propriété sur la zone de disponibilité du AWS Outposts.

Obligatoire : non

Type : chaîne

`tags`

Les balises à associer à la ressource.

Obligatoire : non

Type: liste

Prérequis

`vpc`

Un nœud AWS.Networking.VPC.

Obligatoire : oui

Type : String

`route_table`

Un AWS.Networking. RouteTablenœud.

Obligatoire : oui

Type : String

Exemple

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

```
      - "Environment=Testing"
  requirements:
    vpc: SampleVPC
    route_table: SampleRouteTable

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

AWS.Deployment.VNF Déploiement

Les déploiements NF sont modélisés en fournissant l'infrastructure et l'application qui y sont associées. L'attribut cluster indique le cluster EKS qui hébergera vos NF. L'attribut vnfs spécifie les fonctions réseau pour votre déploiement. Vous pouvez également fournir des opérations d'accroche du cycle de vie facultatives de type pre_create et post_create pour exécuter des instructions spécifiques à votre déploiement, telles que l'appel d'une API du système de gestion des stocks.

Syntaxe

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

Prérequis

`deployment`

Un nœud AWS.Deployment.VNFDeployment.

Obligatoire : non

Type : chaîne

`cluster`

Un nœud AWS.Compute.EKS.

Obligatoire : oui

Type : String

`vnfs`

Un nœud AWS.VNF.

Obligatoire : oui

Type : String

Interfaces

Hooks

Définit l'étape au cours de laquelle les hooks du cycle de vie sont exécutés.

`pre_create`

Un AWS. HookExecutionnœud. Ce hook est exécuté avant le déploiement du `VNFDeployment` nœud.

Obligatoire : non

Type : chaîne

`post_create`

Un AWS. HookExecutionnœud. Ce hook est exécuté après le déploiement du `VNFDeployment` nœud.

Obligatoire : non

Type : chaîne

Exemple

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

AWS.Réseau.VPC

Vous devez spécifier un bloc CIDR pour votre cloud privé virtuel (VPC).

Syntaxe

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

Propriétés

`cidr_block`

La gamme de réseau IPv4 pour le VPC, en notation CIDR.

Obligatoire : oui

Type : String

`ipv6_cidr_block`

Le bloc d'adresse CIDR IPv6 utilisé pour créer le VPC.

Valeur autorisée : `AMAZON_PROVIDED`

Obligatoire : non

Type : chaîne

`dns_support`

Indique si les instances lancées dans le VPC ont obtenu des noms d'hôte DNS.

Obligatoire : non

Type : booléen

Par défaut : `false`

`tags`

Balises à associer à la ressource.

Obligatoire : non

Type: liste

Exemple

```
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 Passerelle .Networking.NAT

Vous pouvez définir un nœud de passerelle NAT public ou privé sur un sous-réseau. Pour une passerelle publique, si vous ne fournissez pas d'identifiant d'allocation d'adresse IP élastique, AWS TNB attribuera une adresse IP élastique à votre compte et l'associera à la passerelle.

Syntaxe

```
tosca.nodes.AWS.Networking.NATGateway:
```

```
  requirements:
    subnet: String
    internet_gateway: String
  properties:
    type: String
    eip_allocation_id: String
    tags: List
```

Propriétés

`subnet`

La référence du AWS nœud .Networking.Subnet.

Obligatoire : oui

Type : String

`internet_gateway`

Le AWS.Networking. InternetGatewayréférence de nœud.

Obligatoire : oui

Type : String

Propriétés

`type`

Indique si la passerelle est publique ou privée.

Valeur autorisée :`PUBLIC`, `PRIVATE`

Obligatoire : oui

Type : String

`eip_allocation_id`

L'ID qui représente l'allocation de l'adresse IP élastique.

Obligatoire : non

Type : chaîne

`tags`

Balises à associer à la ressource.

Obligatoire : non

Type: liste

Exemple

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

AWS.Mise en réseau.Route

Vous pouvez définir un nœud de route qui associe la route de destination à la passerelle NAT en tant que ressource cible et ajoute la route à la table de routage associée.

Syntaxe

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

Propriétés

`dest_cidr_blocks`

La liste des routes IPv4 de destination vers la ressource cible.

Obligatoire : oui

Type: liste

Type de membre : Chaîne

Propriétés

`nat_gateway`

La référence du AWS nœud .Networking.NatGateway.

Obligatoire : oui

Type : String

`route_table`

Le AWS.Networking. RouteTableréférence de nœud.

Obligatoire : oui

Type : String

Exemple

```
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
```

Nœuds communs

Définit les nœuds à utiliser dans NSD et VNFD.

- AWS.HookDefinition.Bash

AWS.HookDefinition.Bash

Définit unAWS HookDefinitiondansbash.

Syntaxe

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

Propriétés

`implementation`

Le chemin relatif vers la définition du hook. Le format doit être le suivant :`./hooks/`*`script_name`*`.sh`

Obligatoire : oui

Type : String

`environment_variables`

Les variables d'environnement pour le script hook bash. Utilisez le format suivant :**`envName`**`=`**`envValue`**avec la regex suivante :`^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`

Assurez-vous que**`envName`**`=`**`envValue`**la valeur répond aux critères suivants :

- N'utilisez pas d'espaces.
- Démarrer**`envName`**avec une lettre (A-Z ou a-z) ou un chiffre (0-9).
- Ne commencez pas le nom de la variable d'environnement par ce qui suitAWSMots clés réservés TNB (distinction majuscules/minuscules) :
  - CRÉATION DE CODE
  - TNB
  - MAISON
  - AWS

- Vous pouvez utiliser n'importe quel nombre de lettres (A-Z ou a-z), de chiffres (0-9) et de caractères spéciaux-et_pour**envName**et**envValue**.

Exemple : `A123-45xYz=Example_789`

Obligatoire : non

Type: liste

`execution_role`

Le rôle de l'exécution du hook.

Obligatoire : oui

Type : String

Exemple

```
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"
```

Sécurité dans AWS Telco Network Builder

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Telco Network Builder, voir AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation du AWS TNB. Les rubriques suivantes expliquent comment configurer le AWS TNB pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources AWS TNB.

Table des matières

Protection des données au sein de AWS TNB

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans AWS Telco Network Builder. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec AWS TNB ou d'autres utilisateurs à Services AWS l'aide de la console, de l'API ou des AWS SDK. AWS CLI Toutes les données

que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d’informations d’identification dans l’adresse URL permettant de valider votre demande adressée à ce serveur.

Manipulation des données

Lorsque vous fermez votre AWS compte, AWS TNB marque vos données pour suppression et les supprime de toute utilisation. Si vous réactivez votre AWS compte dans les 90 jours, AWS TNB restaure vos données. Après 120 jours, AWS TNB supprime définitivement vos données. AWS TNB met également fin à vos réseaux et supprime vos packages de fonctions et vos packages réseau.

Chiffrement au repos

AWS TNB chiffre toujours toutes les données stockées dans le service au repos sans nécessiter de configuration supplémentaire. Ce cryptage est automatique via AWS Key Management Service.

Chiffrement en transit

AWS TNB sécurise toutes les données en transit à l'aide du protocole TLS (Transport Layer Security) 1.2.

Il est de votre responsabilité de chiffrer les données entre vos agents de simulation et leurs clients.

Confidentialité du trafic inter-réseaux

AWS Les ressources informatiques du TNB résident dans un cloud privé virtuel (VPC) partagé par tous les clients. Tout le trafic interne du AWS TNB est resté sur le AWS réseau et ne transite pas par Internet. Les connexions entre vos agents de simulation et leurs clients sont acheminées via Internet.

Gestion des identités et des accès pour AWS TNB

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources AWS TNB. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Table des matières

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans AWS TNB.

Utilisateur du service : si vous utilisez le service AWS TNB pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités du AWS TNB pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans AWS TNB, consultezRésolution des problèmes d'identité et d'accès à AWS Telco Network Builder.

Administrateur du service — Si vous êtes responsable des ressources du AWS TNB dans votre entreprise, vous avez probablement un accès complet au AWS TNB. C'est à vous de déterminer les fonctionnalités et les ressources du AWS TNB auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec AWS TNB, consultez. Comment AWS Telco Network Builder fonctionne avec IAM

Administrateur IAM — Si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à AWS TNB. Pour consulter des exemples de politiques basées sur l'identité AWS TNB que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité pour AWS Telco Network Builder

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section Signature des demandes AWS d'API dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas

utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez Tâches nécessitant des informations d'identification d'utilisateur root dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations

pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console changeant de rôle. Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez Sessions de transmission d'accès.
  - Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
  - Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre

une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principals (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Vue d'ensemble des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour

une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.

- Politiques de contrôle des services (SCP) — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans. AWS Organizations AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations .
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez politiques de séance dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment AWS Telco Network Builder fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS TNB, découvrez quelles fonctionnalités IAM peuvent être utilisées avec TNB. AWS

Fonctionnalités IAM que vous pouvez utiliser avec AWS Telco Network Builder

Fonction IAM	AWS Assistance TNB
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition d'une politique	Oui
ACL	Non
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Fonctions du service	Non
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble du fonctionnement du AWS TNB et des autres AWS services avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le guide de l'utilisateur IAM.

Politiques basées sur l'identité pour le TNB AWS

Prend en charge les politiques basées sur l'identité	Oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles

ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour le TNB AWS

Pour consulter des exemples de politiques basées sur l'identité du AWS TNB, voir. Exemples de politiques basées sur l'identité pour AWS Telco Network Builder

Politiques basées sur les ressources au sein de TNB AWS

Prend en charge les politiques basées sur les ressources	Non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources

accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez Différence entre les rôles IAM et les politiques basées sur une ressource dans le Guide de l'utilisateur IAM.

Actions politiques pour le AWS TNB

Prend en charge les actions de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions AWS TNB, voir Actions définies par AWS Telco Network Builder dans la référence d'autorisation de service.

Les actions politiques dans AWS TNB utilisent le préfixe suivant avant l'action :

```
tnb
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

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

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `List`, incluez l'action suivante :

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

Pour consulter des exemples de politiques basées sur l'identité du AWS TNB, voir. Exemples de politiques basées sur l'identité pour AWS Telco Network Builder

Ressources politiques pour le AWS TNB

Prend en charge les ressources de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de ressources AWS TNB et de leurs ARN, consultez la section Ressources définies par AWS Telco Network Builder dans la référence d'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, voir Actions définies par AWS Telco Network Builder.

Pour consulter des exemples de politiques basées sur l'identité du AWS TNB, voir. Exemples de politiques basées sur l'identité pour AWS Telco Network Builder

Clés de conditions de politique pour AWS TNB

Prend en charge les clés de condition de politique spécifiques au service	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition AWS TNB, consultez la section Clés de condition pour AWS Telco Network Builder dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, voir Actions définies par AWS Telco Network Builder.

Pour consulter des exemples de politiques basées sur l'identité du AWS TNB, voir. Exemples de politiques basées sur l'identité pour AWS Telco Network Builder

ACL en TNB AWS

Prend en charge les listes ACL	Non

Les listes de contrôle d'accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux

politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec TNB AWS

Prend en charge ABAC (étiquettes dans les politiques)	Oui

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, consultez Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) ? dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec AWS TNB

Prend en charge les informations d'identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Changement de rôle (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Autorisations principales interservices pour TNB AWS

Prend en charge les sessions d'accès direct (FAS)	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Rôles de service pour AWS TNB

Prend en charge les fonctions de service	Non

Une fonction du service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

Rôles liés aux services pour TNB AWS

Prend en charge les rôles liés à un service	Non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Exemples de politiques basées sur l'identité pour AWS Telco Network Builder

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources AWS TNB. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par AWS TNB, y compris le format des ARN pour chacun des types de ressources, voir Actions, ressources et clés de condition pour AWS Telco Network Builder dans la référence d'autorisation de service.

Table des matières

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources AWS TNB dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue.

Compte AWS Pour exiger le MFA lorsque des opérations d’API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d’informations, consultez Configuration de l’accès aux API protégé par MFA dans le Guide de l’utilisateur IAM.

Pour plus d’informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l’utilisateur IAM.

Utilisation de la AWS console TNB

Pour accéder à la console AWS Telco Network Builder, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails des ressources AWS TNB de votre Compte AWS. Si vous créez une stratégie basée sur l’identité qui est plus restrictive que l’ensemble minimum d’autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l’accès à uniquement aux actions qui correspondent à l’opération d’API qu’ils tentent d’effectuer.

Exemples de politiques relatives aux rôles de service

En tant qu'administrateur, vous possédez et gérez les ressources créées par AWS TNB, telles que définies par les modèles d'environnement et de service. Vous devez associer des rôles de service IAM à votre compte pour permettre à AWS TNB de créer des ressources pour la gestion du cycle de vie de votre réseau.

Un rôle de service IAM permet à AWS TNB d'appeler des ressources en votre nom afin d'instancier et de gérer vos réseaux. Si vous spécifiez un rôle de service, AWS TNB utilise les informations d'identification de ce rôle.

Le rôle de service et sa politique d'autorisation sont créés à partir du service IAM. Pour plus d'informations sur la création d'un rôle de service, consultez la section Création d'un rôle pour déléguer des autorisations à un AWS service dans le Guide de l'utilisateur IAM.

AWS Rôle de service TNB

En tant que membre de l'équipe de la plateforme, vous pouvez, en tant qu'administrateur, créer un rôle de service AWS TNB et le fournir à AWS TNB. Ce rôle permet à AWS TNB de passer des appels vers d'autres services tels qu'Amazon Elastic Kubernetes AWS CloudFormation Service, de fournir

l'infrastructure requise pour votre réseau et de fournir des fonctions réseau telles que définies dans votre NSD.

Nous vous recommandons d'utiliser le rôle IAM et la politique de confiance suivants pour votre rôle de service AWS TNB. Lorsque vous délimitez les autorisations relatives à cette politique, gardez à l'esprit que AWS TNB peut échouer en cas d'erreurs d'accès refusé vers des ressources exclues de votre politique.

Le code suivant illustre une politique de rôle de service 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"
        }
    ]
}
```

Le code suivant illustre la politique de confiance du service 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 Rôle de service TNB pour le cluster Amazon EKS

Lorsque vous créez des ressources Amazon EKS dans votre NSD, vous fournissez l'`cluster_role`attribut pour spécifier le rôle qui sera utilisé pour créer votre cluster Amazon EKS.

L'exemple suivant montre un AWS CloudFormation modèle qui crée un rôle de service AWS TNB pour la politique de cluster Amazon EKS.

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

Pour plus d'informations sur les rôles IAM utilisant AWS CloudFormation un modèle, consultez les sections suivantes du guide de l'AWS CloudFormation utilisateur :

- AWS::IAM::Role
- Sélection d'un modèle de pile

AWS Rôle de service TNB pour le groupe de nœuds Amazon EKS

Lorsque vous créez des ressources de groupe de nœuds Amazon EKS dans votre NSD, vous fournissez l'`node_role`attribut permettant de spécifier le rôle qui sera utilisé pour créer votre groupe de nœuds Amazon EKS.

L'exemple suivant montre un AWS CloudFormation modèle qui crée un rôle de service AWS TNB pour la politique de groupe de nœuds 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/*"
```

Pour plus d'informations sur les rôles IAM utilisant AWS CloudFormation un modèle, consultez les sections suivantes du guide de l'AWS CloudFormation utilisateur :

- AWS::IAM::Role
- Sélection d'un modèle de pile

AWS Rôle de service TNB pour Multus

Lorsque vous créez une ressource Amazon EKS dans votre NSD et que vous souhaitez gérer Multus dans le cadre de votre modèle de déploiement, vous devez fournir l'`multus_role`attribut pour spécifier le rôle qui sera utilisé pour gérer Multus.

L'exemple suivant montre un AWS CloudFormation modèle qui crée un rôle de service AWS TNB pour une politique 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: "*"
```

Pour plus d'informations sur les rôles IAM utilisant AWS CloudFormation un modèle, consultez les sections suivantes du guide de l'AWS CloudFormation utilisateur :

- AWS::IAM::Role
- Sélection d'un modèle de pile

AWS Rôle du service TNB dans le cadre d'une politique d'accrochage du cycle de vie

Lorsque votre NSD ou votre package de fonctions réseau utilise un hook de cycle de vie, vous avez besoin d'un rôle de service vous permettant de créer un environnement pour l'exécution de vos hooks de cycle de vie.

> **Note**
>
> Votre politique d'accrochage du cycle de vie doit être basée sur ce que tente de faire votre crochet du cycle de vie.

L'exemple suivant montre un AWS CloudFormation modèle qui crée un rôle de service AWS TNB pour une politique d'accrochage du cycle de vie.

```
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"
```

Pour plus d'informations sur les rôles IAM utilisant AWS CloudFormation un modèle, consultez les sections suivantes du guide de l'AWS CloudFormation utilisateur :

- AWS::IAM::Role
- Sélection d'un modèle de pile

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or 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": "*"
        }
    ]
}
```

Résolution des problèmes d'identité et d'accès à AWS Telco Network Builder

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS TNB et IAM.

Problèmes

Je ne suis pas autorisé à effectuer une action dans AWS TNB

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `tnb:`*`GetWidget`* fictives.

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

Dans ce cas, la stratégie de Mateo doit être mise à jour pour l'autoriser à accéder à la ressource *`my-example-widget`* à l'aide de l'action `tnb:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à AWS TNB.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans AWS TNB. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

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

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes ressources AWS TNB

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si AWS TNB prend en charge ces fonctionnalités, consultezComment AWS Telco Network Builder fonctionne avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.

- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

Validation de conformité pour AWS TNB

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides de démarrage rapide sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur AWS la sécurité et la conformité.
- Architecture axée sur la sécurité et la conformité HIPAA sur Amazon Web Services : ce livre blanc décrit comment les entreprises peuvent créer des applications AWS conformes à la loi HIPAA.

> **Note**
>
> Tous ne Services AWS sont pas éligibles à la loi HIPAA. Pour plus d'informations, consultez le HIPAA Eligible Services Reference.

- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST),

le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).

- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.
- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience au AWS TNB

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section Infrastructure AWS globale.

AWS TNB exécute votre service réseau sur des clusters EKS dans un cloud privé virtuel (VPC) dans AWS la région de votre choix.

Sécurité de l'infrastructure à AWS TNB

En tant que service géré, AWS Telco Network Builder est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure,

consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à AWS TNB via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Voici quelques exemples de responsabilités partagées :

- AWS est chargé de sécuriser les composants compatibles avec le AWS TNB, notamment :
  - Instances de calcul (également appelées « travailleurs »)
  - Bases de données internes
  - Communications réseau entre les composants internes
  - L'interface de programmation d'applications (API) AWS TNB
  - AWS Kits de développement logiciel (SDK)
- Vous êtes responsable de la sécurisation de votre accès à vos AWS ressources et aux composants de votre charge de travail, notamment (mais sans s'y limiter) :
  - Utilisateurs, groupes, rôles et politiques IAM
  - Buckets S3 que vous utilisez pour stocker vos données pour TNB AWS
  - Autres ressources Services AWS et ressources que vous utilisez pour prendre en charge le service réseau que vous avez fourni via TNB AWS
  - Le code de votre application
  - Connexions entre le service réseau que vous avez fourni via AWS TNB et ses clients

> **Important**
>
> Vous êtes responsable de la mise en œuvre d'un plan de reprise après sinistre capable de restaurer efficacement un service réseau que vous avez fourni via AWS TNB.

Modèle de sécurité de connectivité réseau

Les services réseau que vous fournissez via AWS TNB s'exécutent sur des instances de calcul au sein d'un cloud privé virtuel (VPC) situé dans AWS une région que vous sélectionnez. Un VPC est un réseau virtuel dans le AWS cloud qui isole l'infrastructure par charge de travail ou entité organisationnelle. Les communications entre les instances de calcul au sein des VPC restent au sein du AWS réseau et ne transitent pas par Internet. Certaines communications internes du service transitent par Internet et sont cryptées. Les services réseau fournis via AWS TNB pour tous les clients opérant dans la même région partagent le même VPC. Les services réseau fournis via AWS TNB pour différents clients utilisent des instances de calcul distinctes au sein du même VPC.

Les communications entre vos clients de service réseau et votre service réseau dans AWS TNB passent par Internet. AWS TNB ne gère pas ces connexions. Il est de votre responsabilité de sécuriser les connexions avec vos clients.

Vos connexions à AWS TNB via le AWS Management Console, AWS Command Line Interface (AWS CLI) et les AWS SDK sont cryptées.

Version IMDS

AWS TNB prend en charge les instances qui exploitent le service de métadonnées d'instance version 2 (IMDSv2), une méthode orientée session. IMDSv2 inclut un niveau de sécurité supérieur à celui de l'IMDSV1. Pour plus d'informations, consultez Renforcer la défense contre les pare-feux ouverts, les proxys inverses et les vulnérabilités SSRF grâce aux améliorations apportées au service de métadonnées d'instance Amazon EC2.

Lorsque vous lancez votre instance, vous devez utiliser IMDSv2. Pour plus d'informations sur IMDSv2, consultez Utiliser IMDSv2 dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

SurveillanceAWSTNB

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances deAWSTNB et votre autreAWSsolutions.AWSfournitAWS CloudTrailà regarderAWSTNB, signalez tout problème et prenez des mesures automatiques le cas échéant.

UtiliserCloudTrailpour recueillir des informations détaillées sur les appels passés àAWSAPI. Vous pouvez enregistrer ces appels sous forme de fichiers journaux dans Amazon S3. Vous pouvez les utiliserCloudTraildes journaux pour déterminer des informations telles que l'appel a été effectué, l'adresse IP source d'où provient l'appel, l'auteur de l'appel et la date de l'appel.

LeCloudTrailles journaux contiennent des informations sur les appels aux actions d'API pourAWSTNB. Ils contiennent également des informations relatives aux appels aux actions d'API provenant de services tels qu'Amazon EC2 et Amazon EBS.

Journalisation des appels d'APIAWS Telco Network Builder à l'aide deAWS CloudTrail

AWSTelco Network Builder est intégré avecAWS CloudTrail, service qui enregistre les actions effectuées par un utilisateur, un rôle ou unAWS service dansAWS TNB. CloudTrail capture les appels d'API pourAWS TNB en tant qu'événements. Les appels capturés incluent les appels de la consoleAWS TNB et les appels de code adressés aux opérations d'APIAWS TNB. Si vous créez un journal de suivi, vous pouvez activer l'envoi en continu d' CloudTrail événements à un compartiment Amazon S3, notamment d'événements pourAWS TNB. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée àAWS TNB, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur de la demande, la date de la demande, ainsi que d'autres informations.

Pour en savoir plus CloudTrail, consultez le Guide deAWS CloudTrail l'utilisateur.

AWSInformations TNB dans CloudTrail

CloudTrail est activé dans votreCompte AWS lors de la création de ce dernier. Lorsqu'une activité a lieu dansAWS TNB, cette activité est enregistrée dans un CloudTrail événement avec d'autres

événementsAWS de service dans Historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre Compte AWS. Pour plus d'informations, veuillez consulter Affichage d'événements avec Historique des CloudTrail événements.

Pour obtenir un enregistrement continu dans votreCompte AWS, y compris les événements pourAWS TNB, créez un journal de suivi. Un journal de suivi permet CloudTrail de livrer des fichiers journaux vers un compartiment Amazon S3 vers un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal d'activité consigne les événements de toutes les Régions dans la partition AWS et livre les fichiers journaux dans le compartiment Simple Storage Service (Amazon S3) de votre choix. En outre, vous pouvez configurer d'autresAWS services pour analyser plus en profondeur les données d'événement collectées dans CloudTrail journaux et prendre les mesures nécessaires pour agir sur celles-ci. Pour en savoir plus, consultez les ressources suivantes :

- Présentation de la création d'un journal d'activité
- CloudTrail services et intégrations pris en charge
- Configuration Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et Réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les actionsAWS TNB sont enregistrées CloudTrail et documentées dans la référence de l'APIAWS Telco Network Builder. Par exemple, les appels à`CreateSolNetworkInstance` et les`CreateSolFunctionPackage``CreateSolNetworkPackage` actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail .

Présentation des entrées journauxAWS TNB

Un journal Amazon S3 que vous spécifiez permet la remise d'événements sous forme de fichiers journaux vers un compartiment Amazon S3 que vous spécifiez. CloudTrail Les fichiers journaux peuvent contenir une ou plusieurs entrée de journal. Un événement représente une demande individuelle émise à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail Les fichiers journaux ne constituent pas une pile ordonnée retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre l'`CreateSolFunctionPackage`action.


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

```
    "requestParameters": null,
    "responseElements": {
        "vnfPkgArn": "arn:aws:tnb:us-east-1:111222333444:function-package/
fp-12345678abcEXAMPLE",
        "id": "fp-12345678abcEXAMPLE",
        "operationalState": "DISABLED",
        "usageState": "NOT_IN_USE",
        "onboardingState": "CREATED"
    },
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111222333444",
    "eventCategory": "Management"
}
```


AWS Tâches de déploiement TNB

Comprenez les tâches de déploiement pour surveiller efficacement les déploiements et agir plus rapidement.

Le tableau suivant répertorie les tâches de déploiement du AWS TNB :

Nom de la tâche pour les déploiements commencés avant le 7 mars 2024	Nom de la tâche pour les déploiements commencés le 7 mars 2024 ou après	Description de la tâche
AppInstallation	ClusterPluginInstall	Installe le plug-in Multus sur le cluster Amazon EKS.
AppUpdate	aucun changement de nom	Met à jour les fonctions réseau déjà installées dans une instance réseau.
-	ClusterPluginUninstall	Désinstalle les plug-ins sur le cluster Amazon EKS.

Nom de la tâche pour les déploiements commencés avant le 7 mars 2024	Nom de la tâche pour les déploiements commencés le 7 mars 2024 ou après	Description de la tâche
ClusterStorageClassConfiguration	aucun changement de nom	Configure la classe de stockage (pilote CSI) sur un cluster Amazon EKS.
FunctionDeletion	aucun changement de nom	Supprime les fonctions réseau des ressources AWS TNB.
FunctionInstantiation	FunctionInstall	Déploie les fonctions réseau à l'aide de HELM.
FunctionUninstallation	FunctionUninstall	Désinstalle la fonction réseau d'un cluster Amazon EKS.
HookExecution	aucun changement de nom	Exécute les hooks du cycle de vie tels que définis dans le NSD.
InfrastructureCancellation	aucun changement de nom	Annule un service réseau.
InfrastructureInstantation	aucun changement de nom	Fournit AWS des ressources pour le compte de l'utilisateur.
InfrastructureTermination	aucun changement de nom	Déprovisionne les AWS ressources invoquées via AWS TNB.
InventoryDeregistration	aucun changement de nom	Désenregistre les AWS ressources du TNB. AWS
KubernetesClusterConfiguration	ClusterConfiguration	Configure le cluster Kubernetes et ajoute des rôles IAM supplémentaires à Amazon EKS, AuthMap comme défini dans le NSD.
NetworkServiceFinalization	aucun changement de nom	Finalise le service réseau et fournit une mise à jour de l'état de réussite ou d'échec.

Nom de la tâche pour les déploiements commencés avant le 7 mars 2024	Nom de la tâche pour les déploiements commencés le 7 mars 2024 ou après	Description de la tâche
NetworkServiceInstantiation	aucun changement de nom	Initialise le service réseau.
SelfManagedNodesConfiguration	aucun changement de nom	Démarre les nœuds autogérés avec le plan de contrôle Amazon EKS et Kubernetes.

Quotas de service pour AWS Telco Network Builder

Les quotas de service, également appelés limites, correspondent au nombre maximum de ressources ou d'opérations de service pour votre AWS compte. Pour plus d'informations, consultez la section Quotas du service AWS dans le Référence générale d'Amazon Web Services.

Les quotas de service pour AWS TNB sont les suivants.

Nom	Par défaut	Ajusta	Description
Opérations de service réseau continues simultanées	Chaque Région prise en charge : 40	Oui	Nombre maximal d'opérations de service réseau simultanées en cours dans une région.
Packages de fonctions	Chaque région prise en charge : 200	Oui	Le nombre maximum de packages de fonctions dans une région.
Packages réseau	Chaque Région prise en charge : 40	Oui	Le nombre maximum de packages réseau dans une région.
Instances de service réseau	Chaque région prise en charge : 800	Oui	Le nombre maximum d'instances de service réseau dans une région.

Historique du document pour le guide de l'utilisateur du AWS TNB

Le tableau suivant décrit les versions de documentation pour AWS TNB.

Modification	Description	Date
Nouvelle tâche et nouveaux noms de tâches pour les tâches existantes	Une nouvelle tâche est disponible. Depuis le 7 mars 2024, certaines tâches existantes portent de nouveaux noms pour des raisons de clarté.	7 mai 2024
Version Kubernetes pour cluster	AWS TNB prend désormais en charge les versions 1.29 de Kubernetes pour créer des clusters Amazon EKS.	10 avril 2024
Support pour l'interface réseau `security_groups`	Vous pouvez associer des groupes de sécurité au nœud AWS.Networking.ENI.	2 avril 2024
Support pour le chiffrement du volume racine Amazon EBS	Vous pouvez activer le chiffrement Amazon EBS pour le volume racine Amazon EBS. Pour l'activer, ajoutez les propriétés dans le nœud AWS.Compute.EKS ou AWS.Compute.EKS. ManagedNode SelfManagedNode	2 avril 2024
Support pour le nœud `labels`	Vous pouvez associer des étiquettes de nœud à votre groupe de nœuds dans le nœud AWS.Compute.EKS	19 mars 2024

	ou AWS.Compute.EKS. ManagedNode SelfManag edNode	
Support pour l'interface réseau `source_dest_check`	Vous pouvez indiquer si vous souhaitez activer ou désactive r le contrôle source/destination de l'interface réseau via le nœud AWS.Networking.ENI.	25 janvier 2024
Support pour les instances Amazon EC2 avec données utilisateur personnalisées	Vous pouvez lancer des instances Amazon EC2 avec des données utilisateur personnalisées via le AWS fichier .Compute. UserData nœud.	16 janvier 2024
Support pour le groupe de sécurité	AWS TNB vous permet d'importer la AWS ressource Security Group.	8 janvier 2024
Description mise à jour de `network_interfaces`	Lorsque la `network_i nterfaces` propriété est incluse dans le SelfManag edNode nœud AWS.Compu te.eks ManagedNode ou AWS.Compute.EKS, AWS TNB obtient l'autorisation relative aux ENI auprès de la propriété si elle est disponibl e, ou auprès de la propriété. `multus_role` `node_role`	18 décembre 2023
Support pour les clusters privés	AWS TNB prend désormais en charge les clusters privés. Pour indiquer un cluster privé, définissez la `access` propriété sur`PRIVATE`.	11 décembre 2023

Version Kubernetes pour cluster	AWS TNB prend désormais en charge les versions 1.28 de Kubernetes pour créer des clusters Amazon EKS.	11 décembre 2023
AWS TNB soutient un groupe de placement	Ajout d'un groupe de placement pour les définitions `AWS.Compute.EKSManagedNode` des `AWS.Compute.EKSSelfManagedNode` nœuds et.	11 décembre 2023

AWS TNB ajoute le support pour IPv6	AWS TNB prend désormais en charge la création d'instances réseau avec une infrastructure IPv6. Vérifiez les nœuds AWS.Networking.VPC, .Networking.Subnet ,AWS.Networking.AWS InternetGateway, AWS.Réseautage. SecurityGroupIngressRule, AWS.Réseautage. SecurityGroupEgressRule, et AWS.compute.EKS pour les configurations IPv6. Nous avons également ajouté les nœuds AWS.Networking.NatGateway et .Networking.Route pour la configuration AWS NAT64. Nous avons mis à jour le rôle de service AWS TNB et le rôle de service AWS TNB pour le groupe de nœuds Amazon EKS pour les autorisations IPv6. Consultez les exemples de politiques relatives aux rôles de service.	16 novembre 2023
Autorisations ajoutées à la politique des rôles de service AWS TNB	Nous avons ajouté des autorisations à la politique des rôles de service AWS TNB pour Amazon S3 et pour AWS CloudFormation permettre l'instanciation de l'infrastructure.	23 octobre 2023

AWS TNB est lancé dans d'autres régions	AWS TNB est désormais disponible dans les régions Asie-Pacifique (Séoul), Canada (centre), Europe (Espagne), Europe (Stockholm) et Amérique du Sud (São Paulo).	27 septembre 2023
Tags pour AWS.compute.EKSSelfManagedNode	AWS TNB prend désormais en charge les balises pour la définition du `AWS.Compute.EKSSelfManagedNode` nœud.	22 août 2023
AWS TNB prend en charge les instances qui exploitent IMDSv2	Lorsque vous lancez votre instance, vous devez utiliser IMDSv2.	14 août 2023
Autorisations mises à jour pour MultusRoleInlinePolicy	Cela inclut MultusRoleInlinePolicy désormais l'`ec2:DeleteNetworkInterface` autorisation.	7 août 2023
Version Kubernetes pour cluster	AWS TNB prend désormais en charge les versions 1.27 de Kubernetes pour créer des clusters Amazon EKS.	25 juillet 2023
AWS.Computer.EKS.AuthRole	AWS TNB vous permet d'ajouter des rôles IAM au cluster Amazon EKS `aws-auth ConfigMap` afin que les utilisateurs puissent accéder au cluster Amazon EKS à l'aide d'un rôle IAM. AuthRole	19 juillet 2023

AWS TNB prend en charge les groupes de sécurité.	Ajout du AWS.Networking. SecurityGroup, AWS.Réseautage. SecurityGroupEgressRule, et AWS.Networking. SecurityGroupIngressRuleau modèle NSD.	18 juillet 2023
Version Kubernetes pour cluster	AWS TNB prend en charge les versions 1.22 à 1.26 de Kubernetes pour créer des clusters Amazon EKS. AWS TNB ne prend plus en charge les versions 1.21 de Kubernetes.	11 mai 2023
AWS.Computer.EKS SelfManagedNode	Vous pouvez créer des nœuds de travail autogérés dans la région, dans les Zones AWS Locales et. AWS Outposts	29 mars 2023
Première version	Il s'agit de la première version du guide de l'utilisateur du AWS TNB.	21 février 2023

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.