



Guide de l'utilisateur

# AWS Générateur de réseaux de télécommunications



# AWS Générateur de réseaux de télécommunications: Guide de l'utilisateur

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

Qu'est-ce que c'est AWS TNB ? .....	1
Nouveau AWS ? .....	2
A quoi sert AWS TNB ? .....	2
AWS TNBfonctionnalités .....	2
Accès AWS TNB .....	4
Tarification pour AWS TNB .....	4
Quelle est la prochaine étape .....	5
Comment AWS TNB fonctionne .....	6
Architecture .....	6
Intégration .....	7
Quotas .....	8
AWS TNBconcepts .....	9
Cycle de vie d'une fonction réseau .....	9
Utiliser des interfaces standardisées .....	10
Packages de fonctions réseau .....	11
AWS TNBdescripteurs de services réseau .....	12
Gestion et opérations .....	13
Descripteurs de services réseau .....	14
Configuration AWS TNB .....	17
Inscrivez-vous pour un Compte AWS .....	17
Création d'un utilisateur doté d'un accès administratif .....	18
Choisissez une AWS région .....	19
Notez le point de terminaison du service .....	19
(Facultatif) Installez le AWS CLI .....	21
Configuration des AWS TNB rôles .....	21
Commencer avec AWS TNB .....	22
Prérequis .....	22
Création d'un package de fonctions .....	23
Création d'un package réseau .....	23
Création et instanciation d'une instance réseau .....	24
Nettoyage .....	25
Packages de fonctions .....	26
Création .....	23
Vue .....	27

Téléchargez un package .....	28
Supprimer un package .....	29
AWS TNBpackages réseau .....	30
Création .....	23
Vue .....	31
Téléchargement .....	32
Suppression .....	32
Réseau .....	34
Opérations liées au cycle de vie .....	34
Création .....	24
Instancier .....	36
Mettre à jour une instance de fonction .....	37
Mettre à jour une instance réseau .....	38
Considérations .....	38
Paramètres que vous pouvez mettre à jour .....	38
Mettre à jour une instance réseau .....	52
Vue .....	53
Résilier et supprimer .....	54
Opérations du réseau .....	55
Vue .....	55
Annuler .....	56
TOSCARéférence .....	57
VNFDmodèle .....	57
Syntaxe .....	57
Modèle de topologie .....	57
AWS.VNF .....	58
AWS.Artifacts.Helm .....	59
NSDmodèle .....	60
Syntaxe .....	60
Utilisation de paramètres définis .....	61
VNFDimportation .....	61
Modèle de topologie .....	62
AWS N.S. ....	63
AWS.Calculez. EKS .....	64
AWS.Calculez. EKS. AuthRole .....	68
AWS.Calculez. EKSMANAGEDNode .....	69

AWS.Calculez. EKSSelfManagedNode .....	76
AWS.Calculez. PlacementGroup .....	82
AWS.Calculez. UserData .....	84
AWS.Réseautage. SecurityGroup .....	86
AWS.Réseautage. SecurityGroupEgressRule .....	87
AWS.Réseautage. SecurityGroupIngressRule .....	90
AWS.Ressource.Importer .....	93
AWS.Réseautage. ENI .....	94
AWS.HookExecution .....	96
AWS.Réseautage. InternetGateway .....	98
AWS.Réseautage. RouteTable .....	100
AWS.Réseau.Sous-réseau .....	101
AWS.Déploiement. VNFDeployment .....	104
AWS.Réseautage. VPC .....	106
AWS.Réseautage. NATGateway .....	108
AWS.Mise en réseau.Route .....	109
Nœuds communs .....	111
AWS.HookDefinition.Bash .....	111
Sécurité .....	113
Protection des données .....	114
Manipulation des données .....	115
Chiffrement au repos .....	115
Chiffrement en transit .....	115
Confidentialité du trafic inter-réseaux .....	115
Gestion des identités et des accès .....	115
Public ciblé .....	116
Authentification par des identités .....	116
Gestion des accès à l'aide de politiques .....	120
Comment AWS TNB fonctionne avec IAM .....	123
Exemples de politiques basées sur l'identité .....	130
Résolution des problèmes .....	145
Validation de conformité .....	147
Résilience .....	148
Sécurité de l'infrastructure .....	149
Modèle de sécurité de connectivité réseau .....	150
IMDSversion .....	151

---

Surveillance .....	152
CloudTrail journaux .....	152
AWS TNBexemples d'événements .....	154
Tâches de déploiement .....	155
Quotas .....	158
Historique de la documentation .....	159
.....	clxvi

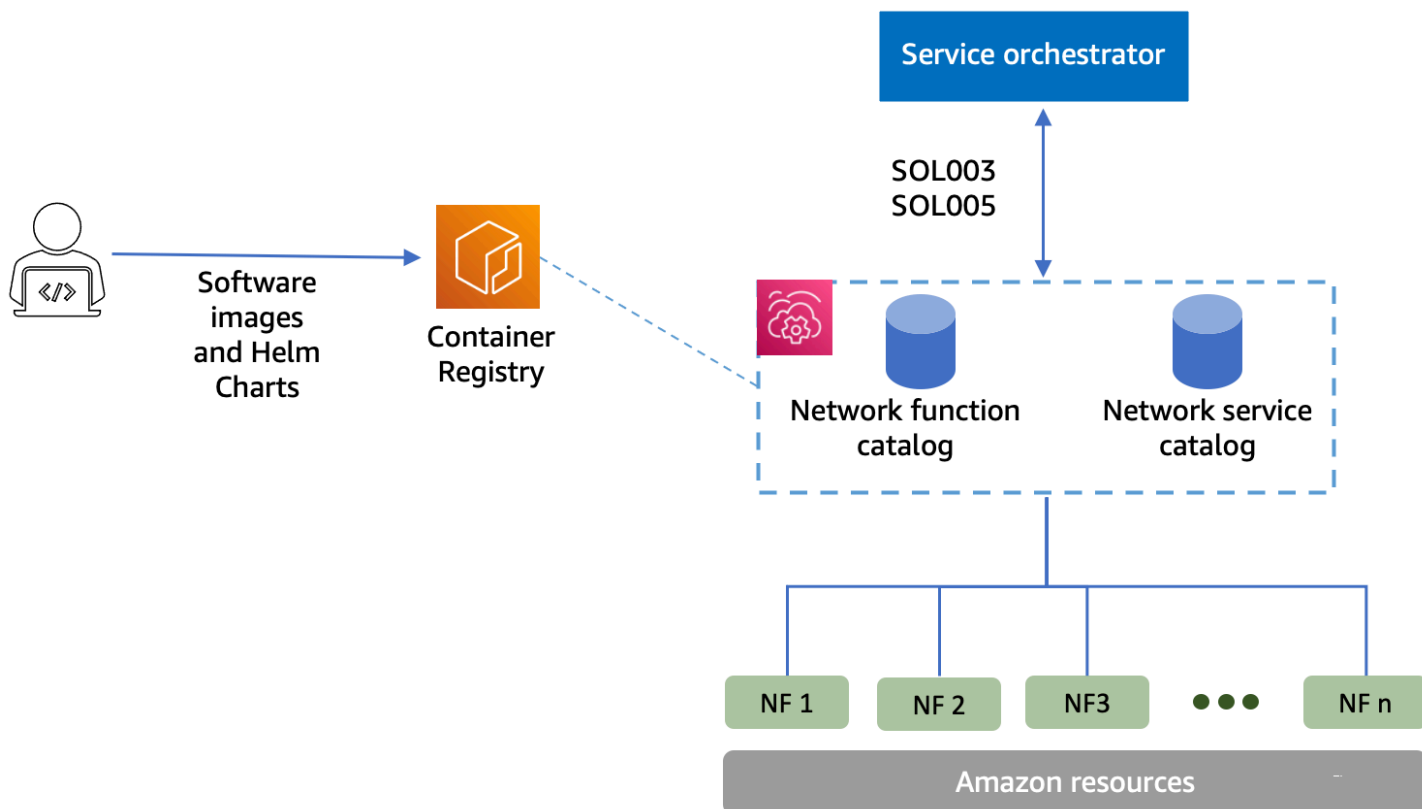
## Qu'est-ce que AWS Telco Network Builder ?

AWS Telco Network Builder (AWS TNB) est un AWS service qui fournit aux fournisseurs de services de communication (CSPs) un moyen efficace de déployer, de gérer et de faire évoluer les réseaux 5G sur AWS l'infrastructure.

Avec AWS TNB, vous déployez des réseaux 5G évolutifs et sécurisés en AWS Cloud utilisant une image de votre réseau de manière automatisée. Vous n'avez pas besoin d'apprendre de nouvelles technologies, de choisir le service informatique à utiliser ou de savoir comment approvisionner et configurer les AWS ressources.

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

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



## Rubriques

- [Nouveau AWS ?](#)
- [A quoi sert AWS TNB ?](#)
- [AWS TNBfonctionnalités](#)
- [Accès AWS TNB](#)
- [Tarification pour AWS TNB](#)
- [Quelle est la prochaine étape](#)

## Nouveau AWS ?

Si vous débutez dans le domaine des AWS produits et services, commencez à en apprendre davantage à l'aide des ressources suivantes :

- [Introduction à AWS](#)
- [Commencer avec AWS](#)

## A quoi sert AWS TNB ?

AWS TNB vise à CSPs tirer parti de la rentabilité, de l'agilité et de l'élasticité qu'il AWS Cloud offre sans écrire ni gérer de scripts et de configurations personnalisés pour concevoir, déployer et gérer des services réseau. AWS TNB provisionne automatiquement l' AWS infrastructure nécessaire, déploie les fonctions réseau conteneurisées et configure le réseau et la gestion des accès afin de créer des services réseau entièrement opérationnels basés sur les descripteurs de services réseau CSP définis et les fonctions réseau que le client souhaite déployer. CSP

## AWS TNBfonctionnalités

Voici quelques-unes des raisons qu'un utilisateur CSP souhaiterait utiliser AWS TNB :

### Aide à simplifier les tâches

Améliorez l'efficacité des opérations de votre réseau, telles que le déploiement de nouveaux services, la mise à jour et la mise à niveau des fonctions réseau et la modification des topologies de l'infrastructure réseau.



## S'intègre aux orchestrateurs

AWS TNBs'intègre aux orchestrateurs de services tiers les plus populaires qui sont conformes ETSI.

## Balances

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

## Inspecte et surveille les ressources AWS

AWS TNBvous permet d'inspecter et de surveiller les AWS ressources qui soutiennent votre réseau sur un tableau de bord unique, comme Amazon VPCEC2, Amazon et AmazonEKS.

## Supporte les modèles de service

AWS TNBvous permet de créer des modèles de service pour toutes les charges de travail des télécommunications (RAN, Core,IMS). Vous pouvez créer une nouvelle définition de service, réutiliser un modèle existant ou intégrer un pipeline d'intégration et de livraison 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'EC2instance Amazon, 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 faire attention à l'ordre des modifications nécessaires. Lorsque vous gérez le cycle de vie de votre fonction réseau, vous apportez AWS TNB uniquement les modifications aux descripteurs de service réseau décrivant la fonction réseau. AWS TNBeffectuera 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 le moment de la mise à niveau. Vous pouvez également gérer vos RAN applications principales et réseau de la même manière. IMS

## Accès AWS TNB

Vous pouvez créer, accéder et gérer vos AWS TNB ressources à l'aide de l'une des interfaces suivantes :

- **AWS TNBconsole** — Fournit une interface Web pour gérer votre réseau.
- **AWS TNBAPI**— Permet d'effectuer RESTful API des AWS TNB actions. Pour plus d'informations, voir [AWS TNBAPIRéférence](#)
- **AWS Command Line Interface (AWS CLI)** — Fournit des commandes pour un large éventail de AWS services, notamment AWS TNB. Il est compatible avec Windows, macOS et Linux. Pour de plus amples informations, veuillez consulter [AWS Command Line Interface](#).
- **AWS SDKs**— Fournit des informations spécifiques à la langue APIs 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 plus d'informations, consultez [AWSSDKs](#).

## Tarification pour AWS TNB

AWS TNBpermet CSPs d'automatiser le déploiement et la gestion de leurs réseaux de télécommunications sur AWS. Vous payez pour les deux dimensions suivantes lorsque vous utilisez AWS TNB :

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

Vous devez également payer des frais supplémentaires lorsque vous utilisez d'autres AWS services conjointement avec AWS TNB. Pour plus d'informations, consultez la section [AWS TNBTarification](#).

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 AWS du compte, consultez la section [Facturation AWS du compte](#).

Si vous avez des questions concernant la AWS facturation, les comptes et les événements, [contactez le AWS Support](#).

AWS Trusted Advisor est un service que vous pouvez utiliser pour optimiser les coûts, la sécurité et les performances de votre AWS environnement. Pour plus d'informations, consultez [AWS Trusted Advisor](#).

## Quelle est la prochaine étape

Pour plus d'informations sur la façon de démarrer AWS TNB, consultez les rubriques suivantes :

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

# Comment AWS TNB fonctionne

AWS TNBs'intègre à des end-to-end orchestrateurs et à des AWS ressources standardisés pour exploiter des réseaux 5G complets.

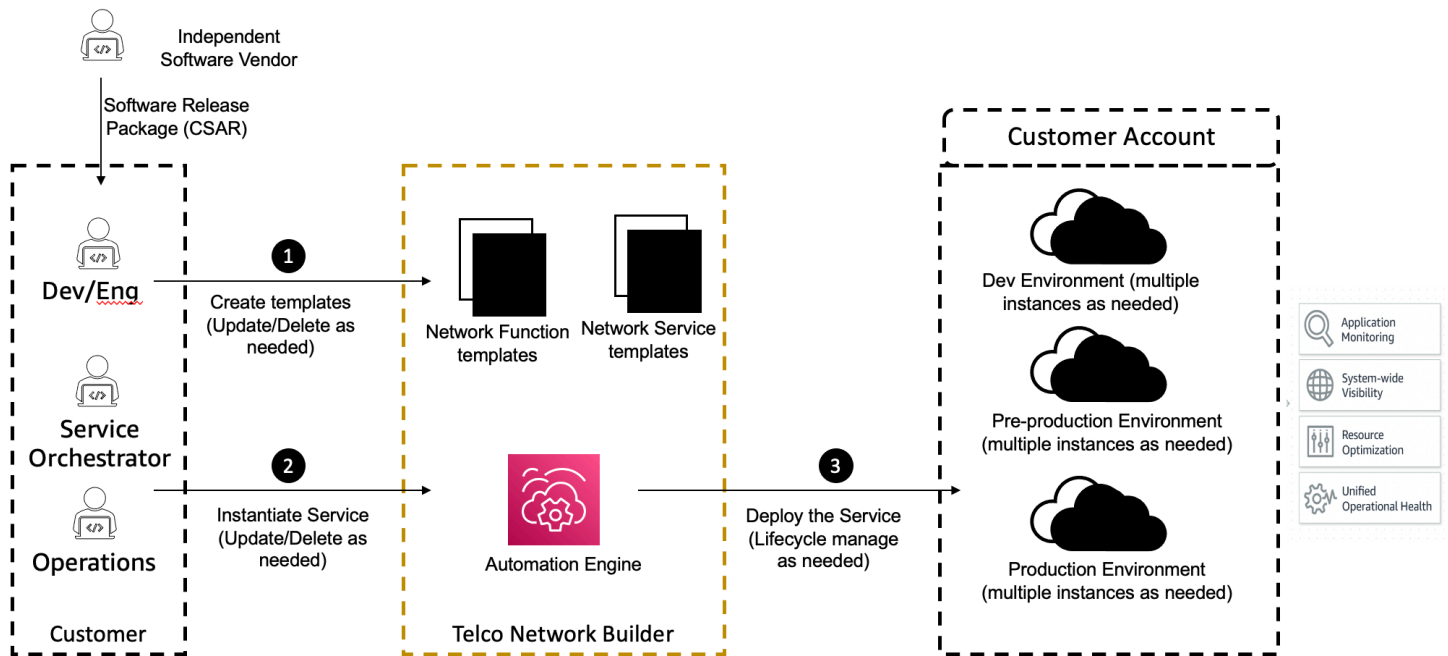
AWS TNBvous permet d'ingérer des packages de fonctions réseau et des descripteurs de services réseau (NSDs) et vous fournit le moteur d'automatisation nécessaire au fonctionnement de vos réseaux. Vous pouvez utiliser votre end-to-end orchestrateur et l'intégrer ou AWS TNB APIs l'utiliser AWS TNB SDKs pour créer votre propre flux d'automatisation. Pour de plus amples informations, veuillez consulter [AWS TNBarchitecture](#) .

## Rubriques

- [AWS TNBarchitecture](#)
- [Intégration avec Services AWS](#)
- [AWS TNBquotas de ressources](#)

## AWS TNBarchitecture

AWS TNBvous permet d'effectuer des opérations de gestion du cycle de vie via AWS Management Console AWS CLI, AWS TNB RESTAPI, etSDKs. Cela permet aux différentes CSP personnes, telles que les membres des équipes d'ingénierie, des opérations et des systèmes programmatiques, d'en tirer parti. AWS TNB Vous créez et téléchargez un package de fonctions réseau sous forme de fichier Cloud Service Archive (CSAR). Le CSAR fichier contient des diagrammes Helm, des images logicielles et un descripteur de fonction réseau (NFD). Vous pouvez utiliser des modèles pour déployer à plusieurs reprises plusieurs configurations de ce package. Vous créez des modèles de services 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 sur AWS l'infrastructure. AWS TNBvous offre la visibilité de vos déploiements.



## Intégration avec Services AWS

Un réseau 5G est constitué d'un ensemble de fonctions réseau conteneurisées interconnectées déployées sur des milliers de clusters Kubernetes. AWS TNBs'intègre aux éléments suivants, spécifiques Services AWS aux télécoms, APIs afin de créer un service réseau entièrement opérationnel :

- Amazon Elastic Container Registry (AmazonECR) pour stocker les artefacts des fonctions réseau des fournisseurs de logiciels indépendants (ISVs).
- Amazon Elastic Kubernetes Service (AmazonEKS) pour configurer des clusters.
- Amazon VPC pour les constructions de réseaux.
- Groupes de sécurité utilisant AWS CloudFormation.
- AWS CodePipeline pour les cibles de déploiement dans Régions AWS les Zones Locales et AWS Outposts.
- IAM pour définir les rôles.
- AWS Organizations pour contrôler l'accès à AWS TNB APIs.
- AWS Health Dashboard et AWS CloudTrail pour surveiller l'état de santé et publier des indicateurs.

## AWS TNBquotas de ressources

Vous Compte AWS disposez de quotas par défaut, anciennement appelés limites, pour chacun d'entre eux Service AWS. Sauf indication contraire, chaque quota est spécifique à un Région AWS. Vous pouvez demander l'augmentation de certains quotas, mais pas de tous les quotas.

Pour consulter les quotas pour AWS TNB, ouvrez la [console Service Quotas](#). Dans le volet de navigation, choisissez Services AWS, puis sélectionnez AWS TNB.

Pour demander une augmentation de quota, consultez [Demande d'augmentation de quota](#) dans le Guide de l'utilisateur Service Quotas.

Vous Compte AWS disposez des quotas suivants relatifs à AWS TNB.

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

# AWS TNB concepts

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

## Table des matières

- [Cycle de vie d'une fonction réseau](#)
- [Utiliser des interfaces standardisées](#)
- [Packages de fonctions réseau pour AWS TNB](#)
- [Descripteurs de service réseau pour AWS TNB](#)
- [Gestion et opérations pour AWS TNB](#)
- [Descripteurs de service réseau pour AWS TNB](#)

## 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 du 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 CSAR packages à déployer ou à mettre à niveau.
4. AWS TNB à utiliser pour télécharger le CSAR package qui définit votre fonction réseau (par exemple, CUAMF, etUPF) et pour l'intégrer à un pipeline d'intégration et de livraison continues (CI/CD) qui peut vous aider à créer de nouvelles versions de votre CSAR package à mesure que de nouvelles images logicielles de fonction réseau, ou des scripts clients, 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 NSDs 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 le sien 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.
- Mettez à jour l'instance réseau avec un nouveau package réseau et de nouvelles valeurs de paramètres. Par exemple, mettez à jour le `EKS version` paramètre Amazon dans le package réseau.

## Utiliser des interfaces standardisées

AWS TNBs'intègre aux orchestrateurs de services conformes aux normes de l'Institut européen des normes de télécommunications (ETSI), ce qui vous permet de simplifier le déploiement de vos services réseau. Les orchestrateurs de services peuvent utiliser AWS TNB SDKsCLI, le ou APIs pour lancer des opérations, telles que l'instanciation ou la mise à niveau d'une fonction réseau vers une nouvelle version.

AWS TNBprend en charge les spécifications suivantes.

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



Spécification de	Version	Description
ETSI SOL005	<a href="#">v3.6.1</a>	Définit les normes relatives aux packages de services réseau et à la gestion du cycle de vie des services réseau.
ETSI SOL007	<a href="#">v3.5.1</a>	Définit les normes pour autoriser les descripteurs de service réseau TOSCA basés sur l'autorisation.

## Packages de fonctions réseau pour AWS TNB

Avec AWS TNB, vous pouvez stocker des packages de fonctions réseau conformes à la norme ETSI SOL 001/ SOL 004 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 la fonction réseau ; par exemple, les scripts et les diagrammes Helm.

CSAR II s'agit d'un package défini par la OASIS TOSCA norme et inclut un descripteur de réseau/ service conforme à la spécification. OASIS TOSCA YAML Pour plus d'informations sur les YAML spécifications requises, consultez [TOSCA référence pour AWS 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 réseau pour AWS TNB

AWS TNB stocke les descripteurs de service réseau (NSDs) relatifs aux fonctions réseau que vous souhaitez déployer et à la manière dont vous souhaitez les déployer dans le catalogue. Vous pouvez télécharger votre YAML NSD fichier (`vnfd.yaml`), comme décrit par ETSI SOL 007 pour inclure les informations suivantes :

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

Ce qui suit est un extrait d'un document NSD montrant comment modéliser. Services AWS La fonction réseau sera déployée sur un EKS cluster Amazon 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: tosa.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 normalisées conformément aux ETSI SOL normes 003 et SOL 005. Vous pouvez utiliser le AWS TNB APIs 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.
- Mettre à jour une instance réseau instanciée ou mise à jour avec un nouveau package réseau et de nouvelles valeurs de paramètres.
- Gestion des versions de vos packages de fonctions réseau.
- Gestion des versions de votre NSDs.
- Récupération d'informations sur les fonctions de votre réseau déployé.

## Descripteurs de service réseau pour AWS TNB

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

Votre NSD dossier est divisé en plusieurs parties :

1. TOSCA version de définition : il s'agit de la première ligne de votre NSD YAML fichier. Elle contient les informations de version, comme indiqué dans l'exemple suivant.

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

2. VNFd— NSD Il 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 figurant dans le CSAR package de fonctions 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 NSD YAML fichier, comme le montre 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: toska.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 Amazon EKS Node, celle-ci doit être créée au sein d'un EKS cluster Amazon. Voici un exemple :

```
SampleEKSNODE:
  type: toska.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 AWS TNB

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

## Tâches

- [Inscrivez-vous pour un Compte AWS](#)
- [Création d'un utilisateur doté d'un accès administratif](#)
- [Choisissez une AWS région](#)
- [Notez le point de terminaison du service](#)
- [\(Facultatif\) Installez le AWS CLI](#)
- [Configuration des AWS TNB rôles](#)

## Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas un Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'<https://portal.aws.amazon.com/billing/inscription>.
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 AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les [tâches nécessitant un accès utilisateur racine](#).

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à <https://aws.amazon.com/> et en choisissant Mon compte.

# Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en [AWS Management Console](#) tant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez [Connexion en tant qu'utilisateur racine](#) dans le Guide de l'utilisateur Connexion à AWS .

2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur root.

Pour obtenir des instructions, voir [Activer un MFA périphérique virtuel pour votre utilisateur Compte AWS root \(console\)](#) dans le guide de IAM l'utilisateur.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

Pour obtenir des instructions, consultez [Activation d' AWS IAM Identity Center](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Dans IAM Identity Center, accordez un accès administratif à un utilisateur.

Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir [Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center](#) dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL identifiant envoyé à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.



Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section [Connexion au portail AWS d'accès](#) dans le guide de Connexion à AWS l'utilisateur.

### Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme à la meilleure pratique consistant à appliquer les autorisations du moindre privilège.

Pour obtenir des instructions, consultez [Création d'un ensemble d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

Pour obtenir des instructions, consultez [Ajout de groupes](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

## 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 [AWS TNB points de terminaison](#) 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 terminaison standard AWS, certains 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	

Nom de la région	Région	Point de terminaison	Protocole
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
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 y accéder AWS TNB en utilisant le AWS CLI. Consultez le [AWS Command Line Interface Guide de l'utilisateur](#) pour démarrer. Pour plus d'informations sur les commandes pour AWS TNB, consultez [tnb](#) dans la référence des AWS CLI commandes.

## Configuration des AWS TNB rôles

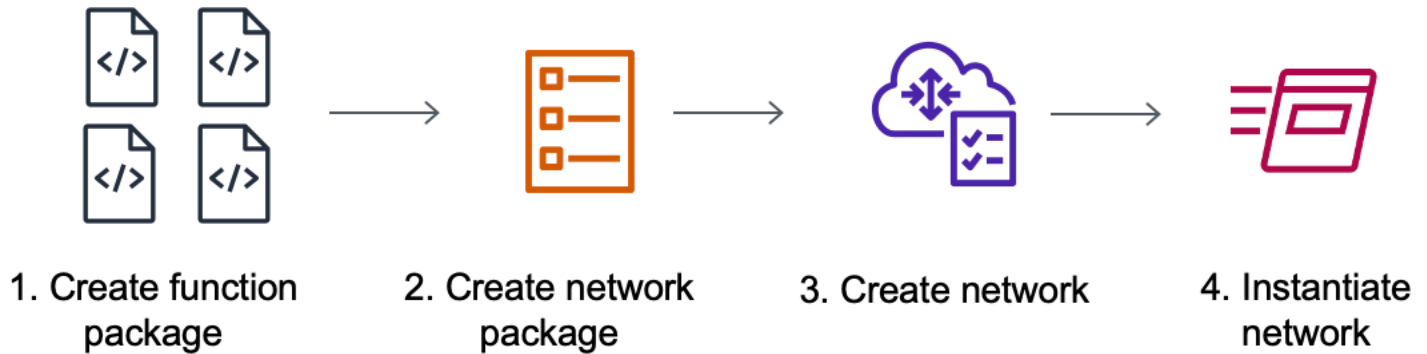
Vous devez créer un rôle IAM de service pour gérer les différentes parties de votre AWS TNB solution. AWS TNBles rôles de service peuvent API appeler d'autres AWS services, tels que AWS CloudFormation AWS CodeBuild, et divers services de calcul et de stockage, en votre nom, afin d'instancier et de gérer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

Pour plus d'informations sur le rôle AWS TNB de service, consultez [Gestion des identités et des accès pour AWS TNB](#).

# Commencer avec AWS TNB

Ce didacticiel explique comment AWS TNB 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 instantiation 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 IAM les rôles.
- Un [package de fonctions réseau \(NF\)](#) conforme à la norme ETSI SOL 001/ 004SOL.
- [Modèles de descripteur de service réseau \(NSD\)](#) conformes à ETSI SOL 007.

Vous pouvez utiliser un exemple de package de fonctions ou de package réseau à partir de la section [Exemples de packages pour AWS TNB](#) GitHub le site.

## Création d'un package de fonctions

Un package de fonctions réseau est un fichier Cloud Service Archive (CSAR). Le CSAR fichier contient des diagrammes Helm, des images logicielles et un descripteur de fonction réseau (NFD).

Pour créer un package de fonctions

1. Ouvrez la AWS TNB console 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 les fichiers, puis téléchargez chaque CSAR package sous forme de .zip fichier. Vous pouvez télécharger un maximum de 10 fichiers.
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

Un package réseau indique les fonctions réseau que vous souhaitez déployer et la manière dont vous souhaitez les déployer dans le catalogue.

Pour créer un package réseau

1. Dans le volet de navigation, sélectionnez Network packages.
2. Choisissez Créer un package réseau.
3. Sous Télécharger un package réseau, choisissez Choisir des fichiers, puis chargez chacun d'entre eux NSD sous forme de .zip fichier. Vous pouvez télécharger un maximum de 10 fichiers.

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

Une instance réseau est un réseau unique créé et AWS TNB qui peut être déployé. Vous devez créer une instance réseau et l'instancier. Lorsque vous instanciez une instance réseau, que vous AWS TNB provisionnez l' AWS infrastructure nécessaire, que vous déployez des fonctions réseau conteneurisées et que vous configurez le réseau et la gestion des accès pour créer un service réseau entièrement opérationnel.

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. Choisissez un package réseau. Vérifiez les informations et choisissez Next.
5. Choisissez Create network instance. L'état initial est Created.

La page Réseaux apparaît et indique la nouvelle instance de réseau dans son Not instantiated état actuel.

6. Sélectionnez l'instance réseau, choisissez Actions et Instanciation.

La page Network instanciate apparaît.

7. Vérifiez les détails et mettez à jour les valeurs des paramètres. Les mises à jour des valeurs des paramètres s'appliquent uniquement à cette instance réseau. Les paramètres des VNFD packages NSD et ne changent pas.
8. Choisissez Instancier le réseau.

La page État du déploiement apparaît.

9. Utilisez l'icône Actualiser pour suivre l'état de déploiement de votre instance réseau. Vous pouvez également activer l'actualisation automatique dans la section Tâches de déploiement pour suivre la progression de chaque tâche.

# Nettoyage

Vous pouvez désormais supprimer les ressources que vous avez créées pour ce didacticiel.

Pour nettoyer vos ressources

1. Dans le volet de navigation, sélectionnez Networks.
2. Choisissez l'ID du réseau, puis sélectionnez 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 ETSI standard) et un descripteur de package de fonctions qui utilise la TOSCA norme pour décrire comment les fonctions réseau doivent s'exécuter sur votre réseau.

## Tâches

- [Créer un package de fonctions dans AWS TNB](#)
- [Afficher un package de fonctions dans AWS TNB](#)
- [Télécharger un package de fonctions sur AWS TNB](#)
- [Supprimer un package de fonctions de AWS TNB](#)

## Créer un package de fonctions dans AWS TNB

Découvrez comment créer un package de fonctions dans le catalogue des fonctions AWS TNB réseau. La création d'un package de fonctions est la première étape pour créer un réseau dans AWS TNB. Après avoir chargé un package de fonctions, vous pouvez créer un package réseau.

## Console

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

1. Ouvrez la AWS TNB console 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 des fichiers et chargez chaque CSAR package sous forme de .zip fichier. Vous pouvez télécharger un maximum de 10 fichiers.
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-package](#) commande 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 AWS TNB console 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-packages](#) commande pour répertorier vos packages de fonctions.

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

2. Utilisez la [get-sol-function-package](#) commande 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 sur AWS TNB

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

### Console

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

1. Ouvrez la AWS TNB console 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 AWS TNB réseau. Pour supprimer un package de fonctions, celui-ci doit être dans un état désactivé.

## Console

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

1. Ouvrez la AWS TNB console 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, Supprimer.

## AWS CLI

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

1. Utilisez la [update-sol-function-package](#) commande 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-package](#) commande 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' AWS infrastructure 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 sur 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 VPCs sous-réseaux ou EKS clusters Amazon.

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 AWS TNB console 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 des fichiers et téléchargez-les NSD sous forme de .zip fichier. Vous pouvez télécharger un maximum de 10 fichiers.
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-package](#) commande 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 AWS TNB console 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-packages](#) commande pour répertorier vos packages réseau.

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

2. Utilisez la [get-sol-network-package](#) commande 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 sur AWS TNB

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

### Console

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

1. Ouvrez la AWS TNB console 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 AWS TNB réseau. 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 AWS TNB console 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, Supprimer.

## AWS CLI

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

1. Utilisez la [update-sol-network-package](#) commande 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-package](#) commande 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 et AWS TNB qui peut être déployé.

## Tâches

- [Opérations du cycle de vie d'une instance réseau](#)
- [Créez une instance réseau à l'aide de AWS TNB](#)
- [Instancier une instance réseau à l'aide de AWS TNB](#)
- [Mettre à jour une instance de fonction dans AWS TNB](#)
- [Mettre à jour une instance réseau dans AWS TNB](#)
- [Afficher une instance réseau dans AWS TNB](#)
- [Mettre fin à une instance réseau et la supprimer de AWS TNB](#)

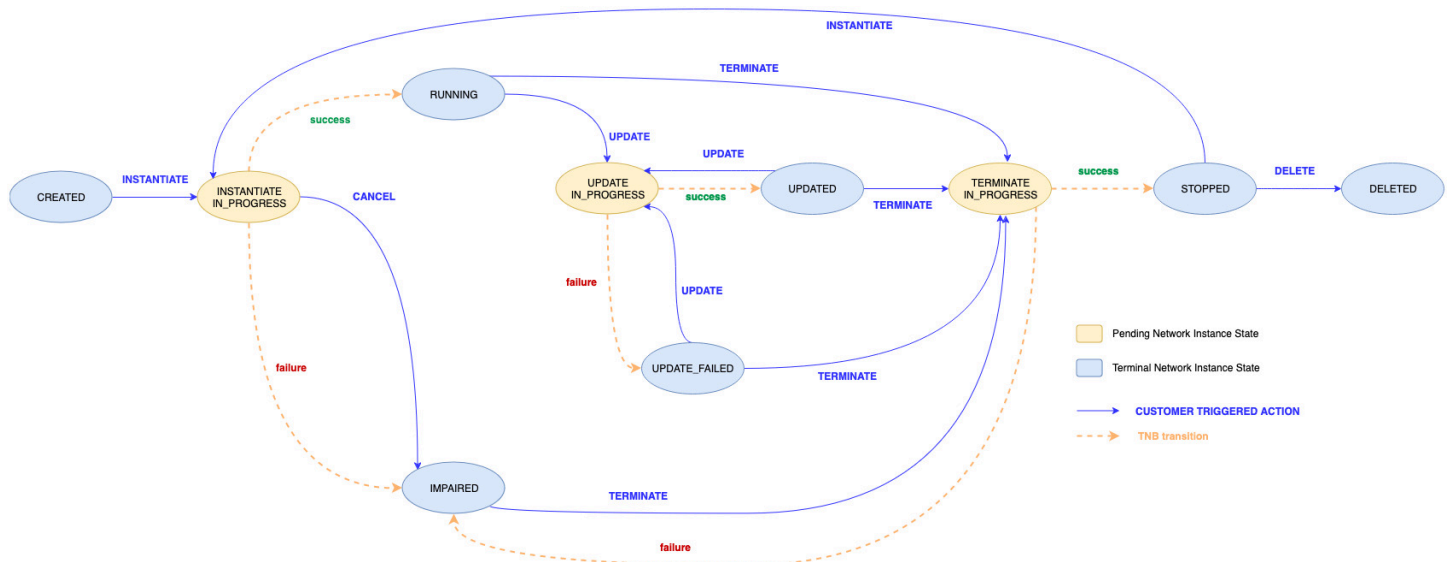
## Opérations du cycle de vie d'une instance réseau

AWS TNB vous permet de gérer facilement votre réseau à l'aide d'opérations de gestion standardisées conformes aux normes ETSI SOL 003 et SOL 005. Vous pouvez effectuer les opérations de cycle de vie suivantes :

- Créez le réseau
- Instancier le réseau
- Mettre à jour la fonction réseau
- Mettre à jour l'instance réseau
- Afficher les détails et l'état du réseau
- Mettre fin au réseau

L'image suivante montre les opérations de gestion du réseau :





## Créez une instance réseau à l'aide de AWS TNB

Vous créez une instance réseau après avoir créé un package réseau. Après avoir créé une instance réseau, instanciez-la.

### Console

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

1. Ouvrez la AWS TNB console 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 le package réseau, vérifiez les informations, puis choisissez Next.
6. Choisissez Create network instance.

La nouvelle instance réseau apparaît sur la page Réseaux. Ensuite, instanciez cette instance réseau.

### AWS CLI

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

- Utilisez la [create-sol-network-instance](#) commande 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"
```

Ensuite, instanciez cette instance réseau.

## Instancier une instance réseau à l'aide de AWS TNB

Après avoir créé une instance réseau, vous devez l'instancier. Lorsque vous instanciez une instance réseau, AWS TNB provisionne l' AWS infrastructure nécessaire, déploie des fonctions réseau conteneurisées et configure le réseau et la gestion des accès pour créer un service réseau entièrement opérationnel.

### Console

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

1. Ouvrez la AWS TNB console à l'adresse <https://console.aws.amazon.com/tnb/>.
2. Dans le volet de navigation, sélectionnez Networks.
3. Sélectionnez l'instance réseau que vous souhaitez instancier.
4. Choisissez Actions, puis Instancier.
5. Sur la page Instancier le réseau, passez en revue les détails et, éventuellement, mettez à jour les valeurs des paramètres.

Les mises à jour des valeurs des paramètres s'appliquent uniquement à cette instance réseau. Les paramètres des VNFD packages NSD et ne changent pas.

6. Choisissez Instancier le réseau.

La page État du déploiement s'affiche.

7. Utilisez l'icône Actualiser pour suivre l'état de déploiement de votre instance réseau. Vous pouvez également activer l'actualisation automatique dans la section Tâches de déploiement pour suivre la progression de chaque tâche.

Lorsque l'état du déploiement passe à `Completed`, l'instance réseau est instanciée.

## AWS CLI

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

1. Utilisez la [instantiate-sol-network-instance](#) commande pour instancier l'instance réseau.

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

2. Ensuite, consultez l'état de fonctionnement du réseau.

## Mettre à jour une instance de fonction dans AWS TNB

Après l'instanciation d'une instance réseau, vous pouvez mettre à jour un package de fonctions dans l'instance réseau.

### Console

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

1. Ouvrez la AWS TNB console à l'adresse <https://console.aws.amazon.com/tnb/>.
2. Dans le volet de navigation, sélectionnez Networks.
3. Sélectionnez l'instance réseau. Vous ne pouvez mettre à jour une instance réseau que si son état est `Instantiated`.

La page de l'instance réseau s'affiche.

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.
7. Choisissez Mettre à jour.

## AWS CLI

Utilisez le CLI pour mettre à jour une instance de fonction

Utilisez la [update-sol-network-instance](#) commande avec le type de `MODIFY_VNF_INFORMATION` mise à jour pour mettre à jour une instance de fonction dans 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 à jour une instance réseau dans AWS TNB

Après l'instanciation d'une instance réseau, vous devrez peut-être mettre à jour l'infrastructure ou l'application. Pour ce faire, vous devez mettre à jour le package réseau et les valeurs des paramètres de l'instance réseau et déployer l'opération de mise à jour pour appliquer les modifications.

### Considérations

- Vous pouvez mettre à jour une instance réseau à l'Updated état Instantiated ou.
- Lorsque vous mettez à jour une instance réseau, le nouveau package réseau et les nouvelles valeurs de paramètres sont UpdateSolNetworkService API utilisés pour mettre à jour la topologie de l'instance réseau.
- AWS TNB vérifie que le nombre NSD et VNFD les paramètres de l'instance réseau ne dépassent pas 200. Cette limite est appliquée pour empêcher les acteurs malveillants de transmettre des charges utiles erronées ou énormes qui affectent le service.

### Paramètres que vous pouvez mettre à jour

Vous pouvez mettre à jour les paramètres suivants lorsque vous mettez à jour une instance réseau instanciée :

Paramètre	Description	Exemple : Avant	Exemple : Après
Version EKS du cluster Amazon	Vous pouvez mettre à jour la valeur du version paramètre du plan de contrôle du EKS cluster Amazon vers la version mineure suivante. Vous ne pouvez pas rétrograder la version. Les nœuds de travail ne sont pas mis à jour.	<pre>EKScluster:   type: tosca.nodes.AWS.Compute.EKS   properties:     version: "1.28"</pre>	<pre>EKScluster:   type: tosca.nodes.AWS.Compute.EKS   properties:     version: "1.28"</pre>

Paramètre	Description	Exemple : Avant

Exem  
Après  
es.A  
mput  
pro  
s:  
ver  
"1.

Paramètre	Description	Exemple : Avant	Exem Après
Propriétés de dimensionnement	Vous pouvez mettre à jour les propriétés de mise à l'échelle EKSMANagedNode des EKSSelfManagedNode TOSCA nœuds et.	<pre> EKSNodeGroup01:   ...   scaling:     properties:       desired_s size: 1       min_size: 1       max_size: 1 </pre>	<pre> EKSM oup0 ... sca  pro s:  des ize: </pre>



Paramètre	Description	Exemple : Avant	Exem Après
Propriétés EBS CSI du plugin Amazon	Vous pouvez activer ou désactiver le EBS CSI plug-in Amazon sur vos EKS clusters Amazon. Vous pouvez également modifier la version du plugin.	<pre> EKSCluster:   capabilities:     ...     ebs_csi:       properties:         enabled: <i>false</i> </pre>	<pre> EKSCLu r:   cap ies:   ...  ebs  pro s:  ena  ver "v1 e ksbu "</pre>



Paramètre	Description	Exemple : Avant	Exem Après
VNF	<p>Vous pouvez les référence r VNFs dans le NSD et les déployer sur le cluster créé à NSD l'aide VNFDeployment TOSCA du nœud. Dans le cadre de la mise à jour, vous pourrez ajouter, mettre à jour et VNFs supprimer des informations sur le réseau.</p>	<pre> vnfds:   - descriptor_id:     "43c012fa-2616-41a8-     a833-0dfd4c5a049e "     namespace: " vnf1"   - descriptor_id:     "64222f98-ecd6-4871-     bf94-7354b53f3ee5 "     namespace:     "vnf2" // Deleted VNF ... SampleVNF1HelmDeploy:   type: toska.nod es.AWS.Deployment. VNFDeployment   requirements:     cluster:       EKSCluster       vnfs:         - vnf1.Samp leVNF1         - vnf2.Samp leVNF2                     </pre>	<pre> vnfd - des r_id "55 79e9 - be53 2ad0 "  nam : "vr Upd VNF  - des r_id "b7 839c -916 a166 "  nam : "vr Add VNF .... Sa mple                     </pre>

Paramètre	Description	Exemple : Avant

Exem  
Après

eImD  
:

typ  
tos  
es.A  
ploy  
VNFD  
ment

rec  
nts:

clu  
EKS  
r

vnf

Paramètre	Description	Exemple : Avant

Exem  
Après

- v  
LeVM

- v  
LeVM

Paramètre	Description	Exemple : Avant	Exem Après
Hooks	<p>Pour exécuter des opérations de cycle de vie avant et après la création d'une fonction réseau, ajoutez les <code>post_create</code> crochets <code>pre_create</code> et au <code>VNFDeployment</code> nœud.</p> <p>Dans cet exemple, le <code>PreCreateHook</code> hook s'exécutera avant d'être instancié et le <code>PostCreateHook</code> hook <code>vnf3.SampleVNF3</code> s'exécutera après <code>vnf3.SampleVNF3</code> l'instanciation.</p>	<pre>vnfds:   - descriptor_id:     "43c012fa-2616-41a8-     a833-0dfd4c5a049e "     namespace: " vnf1"   - descriptor_id:     "64222f98-ecd6-4871-     bf94-7354b53f3ee5 "     namespace: " vnf2"   ... SampleVNF1HelmDeploy:   type: tosca.nod es.AWS.Deployment. VNFDeployment   requirements:     cluster: EKSCluster   vnfs:     - vnf1.SampleVNF1     - vnf2.Samp leVNF2 // Removed during update</pre>	<pre>vnfd - des r_id "43 2616 - a833 d4c5 " nam : "vr - des r_id "b7 839c -916 a166 " nam : "vr .... S ampL Helm y: typ tos</pre>

Paramètre	Description	Exemple : Avant

Exem  
Après  
es.A  
ploy  
VNFD  
ment  
rec  
nts:  
clu  
EKS  
r  
vnf  
- v  
leVM  
No  
cha  
to  
thi  
fur  
as  
the  
nam  
and  
uui  
rem  
the  
sam

Paramètre	Description	Exemple : Avant

Exem  
Après

- v  
*LeVM*

New

VNF

as

the

nam

'  
vnt

was

not

pre

y  
pre

int

s:

Ho

pos

te:

*eHo*

pre

e:

*Hook*

Paramètre	Description	Exemple : Avant	Exem Après
Hooks	<p>Pour exécuter des opérations de cycle de vie avant et après la mise à jour d'une fonction réseau, vous pouvez ajouter le <code>pre_update</code> <code>post_update</code> hook et le hook au VNFDeployment nœud.</p> <p>Dans cet exemple, <code>PreUpdateHook</code> sera exécuté avant <code>vnf1.SampleVNF1</code> la mise à jour et <code>PostUpdateHook</code> exécutera après <code>vnf1.SampleVNF1</code> la mise à jour vers le vnf package indiqué par la mise à jour <code>uuid</code> pour l'espace de noms <code>vnf1</code>.</p>	<pre> vnfds:   - descriptor_id:     "43c012fa-2616-41a8-     a833-0dfd4c5a049e "     namespace: " vnf1"   - descriptor_id:     "64222f98-ecd6-4871-     bf94-7354b53f3ee5 "     namespace: " vnf2"   ...  SampleVNF1HelmDeploy:   type: tosca.nod es.AWS.Deployment. VNFDeployment   requirements:     cluster: EKSCluster   vnfs:     - vnf1.SampleVNF1     - vnf2.Samp leVNF2 </pre>	<pre> vnfd - des r_id "0e bd87 - b8a1 4666 "  nam : "vr ... S ampl Helm y:  typ </pre>

Paramètre	Description	Exemple : Avant

Exem  
Après

tos  
es.A  
ploy  
VNFD  
ment

rec  
nts:

clu  
EKS  
r

vnf

- v  
LeVM  
A  
VNF  
upc  
as  
the  
uui  
cha  
for  
nam  
"vr

- v



Paramètre	Description	Exemple : Avant

Exem  
Après

*LeVM*  
No  
cha  
to  
thi  
fur  
as  
nam  
and  
uui  
rem  
the  
sam

int  
s:

Hoc

pre  
e:  
*Hook*

pos  
te:  
*eHoc*

# Mettre à jour une instance réseau

## Console

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

1. Ouvrez la AWS TNB console à l'adresse <https://console.aws.amazon.com/tnb/>.
2. Dans le volet de navigation, sélectionnez Networks.
3. Sélectionnez l'instance réseau. Vous ne pouvez mettre à jour une instance réseau que si son état est `Instantiated` ou `Updated`.
4. Choisissez Actions et Mettre à jour.

La page Mettre à jour l'instance apparaît avec les détails du réseau et une liste des paramètres de l'infrastructure actuelle.

5. Choisissez un nouveau package réseau.

Les paramètres du nouveau package réseau apparaissent dans la section Paramètres mis à jour.

6. Vous pouvez éventuellement mettre à jour les valeurs des paramètres dans la section Paramètres mis à jour. Pour la liste des valeurs de paramètres que vous pouvez mettre à jour, consultez [Paramètres que vous pouvez mettre à jour](#).
7. Choisissez Mettre à jour le réseau.

AWS TNB valide la demande et lance le déploiement. La page État du déploiement apparaît.

8. Utilisez l'icône Actualiser pour suivre l'état de déploiement de votre instance réseau. Vous pouvez également activer l'actualisation automatique dans la section Tâches de déploiement pour suivre la progression de chaque tâche.

Lorsque l'état du déploiement passe à `Completed`, l'instance réseau est mise à jour.

9.
  - Si la validation échoue, l'instance réseau reste dans le même état qu'avant que vous ne demandiez la mise à jour, `Instantiated` soit `Updated`.
  - Si la mise à jour échoue, l'état de l'instance réseau s'affiche `Update failed`. Choisissez le lien correspondant à chaque tâche ayant échoué pour en déterminer la raison.
  - Si la mise à jour réussit, l'état de l'instance réseau s'affiche `Updated`.

## AWS CLI

Utilisez le CLI pour mettre à jour une instance réseau

Utilisez la [update-sol-network-instance](#) commande avec le type de UPDATE\_NS mise à jour pour mettre à jour une instance réseau.

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

## 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 AWS TNB console 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-instances](#) commande pour répertorier vos instances réseau.

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

2. Utilisez la [get-sol-network-instance](#) commande pour afficher les détails d'une instance réseau spécifique.

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

# 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 AWS TNB console 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 Résilier.
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-instance](#) commande 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-instance](#) commande pour supprimer une instance réseau.

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

# Opérations réseau pour AWS TNB

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

## Tâches

- [Afficher une opération AWS TNB réseau](#)
- [Annuler une opération AWS TNB réseau](#)

## Afficher une opération AWS TNB réseau

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

### Console

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

1. Ouvrez la AWS TNB console 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.
4. Dans l'onglet Déploiements, choisissez le fonctionnement du réseau.

### AWS CLI

Pour visualiser une opération réseau à l'aide du AWS CLI

1. Utilisez la [list-sol-network-operations](#) commande pour répertorier toutes les opérations réseau.

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

2. Utilisez la [get-sol-network-operation](#) commande pour afficher les détails d'une opération réseau.

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

# Annuler une opération AWS TNB réseau

Découvrez comment annuler une opération réseau.

## Console

Pour annuler une opération réseau à l'aide de la console

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

## AWS CLI

Pour annuler une opération réseau à l'aide du AWS CLI

Utilisez la [cancel-sol-network-operation](#) commande pour annuler une opération réseau.

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

# TOSCA référence pour AWS TNB

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

## Table des matières

- [VNFD modèle](#)
- [Modèle de descripteur de service réseau](#)
- [Nœuds communs](#)

## VNFD modèle

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

Les TOSCA AWS nœuds. Les nœuds possibles sont les suivants :

- [AWS.VNF](#)
- [AWS.Artefacts. Casque](#)

## AWS.VNF

Définit un nœud de fonction réseau AWS 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

Le 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 duVNFD.

Obligatoire : oui

Type : String

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

#### descriptor\_name

Nom du descripteur.



Obligatoire : oui

Type : String

provider

L'auteur duVNFD.

Obligatoire : oui

Type : String

## Prérequis

helm

Le répertoire Helm définissant les artefacts du conteneur. Il s'agit d'une référence à [AWS.Artifacts.Helm](#).

Obligatoire : oui

Type : String

## Exemple

```
SampleVNF:
  type: toska.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 un nœud AWS Helm.

## Syntaxe

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

```
properties:  
  implementation: String
```

## Propriétés

### implementation

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

Obligatoire : oui

Type : String

## Exemple

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

## Modèle de descripteur de service réseau

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 CIDR bloc du VPC nœud, vous pouvez utiliser la `{ get_input: input-parameter-name }` syntaxe et définir les paramètres dans le NSD modèle. Réutilisez ensuite le paramètre dans le même NSD modèle.

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 }
```

## VNFDimportation

### descriptor\_id

Le 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 TOSCA AWS nœuds possibles sont les suivants :

- [AWS N.S.](#)
- [AWS.Calculez. EKS](#)
- [AWS.Calculez. EKS. AuthRole](#)
- [AWS.Calculez. EKSMangedNode](#)
- [AWS.Calculez. EKSSelfManagedNode](#)
- [AWS.Calculez. PlacementGroup](#)
- [AWS.Calculez. UserData](#)
- [AWS.Réseautage. SecurityGroup](#)
- [AWS.Réseautage. SecurityGroupEgressRule](#)
- [AWS.Réseautage. SecurityGroupIngressRule](#)
- [AWS.Ressource.Importer](#)
- [AWS.Réseautage. ENI](#)
- [AWS.HookExecution](#)
- [AWS.Réseautage. InternetGateway](#)
- [AWS.Réseautage. RouteTable](#)
- [AWS.Réseau.Sous-réseau](#)
- [AWS.Déploiement. VNFDeployment](#)

- [AWS.Réseautage. VPC](#)
- [AWS.Réseautage. NATGateway](#)
- [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

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

## AWS.Calculez. 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 votre. AWS NFs Les plugins Multus Container Network Interface (CNI) 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

```
toska.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 du conteneur Multus (CNI).

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 EKS cluster Amazon.

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

#### `enabled`

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

Obligatoire : non

Type : booléen

## version

Version du module complémentaire Amazon EBS CSI Driver. La version doit correspondre à l'une des versions renvoyées par l'DescribeAddonVersionsaction. Pour plus d'informations, consultez [DescribeAddonVersions](#)le Amazon EKS API Reference

Obligatoire : non

Type : String

## Propriétés

### version

Version de Kubernetes pour le cluster. AWS Telco Network Builder prend en charge les versions 1.23 à 1.30 de Kubernetes.

Obligatoire : oui

Type : String

Valeurs possibles : 1,23 | 1,24 | 1,25 | 1,26 | 1,27 | 1,28 | 1,29 | 1,30

### 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 : String

## Prérequis

subnets

Un nœud [AWS.Networking.Subnet](#).

Obligatoire : oui

Type: liste

## Exemple

```
SampleEKS:
  type: tosa.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.Calculez. EKS. AuthRole

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

### 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 IAM les rôles qui doivent être ajoutés au EKS cluster aws-auth ConfigMap Amazon.

arn

Le ARN IAM rôle.

Obligatoire : oui

Type : String

groups

Groupes Kubernetes à attribuer au rôle défini dans. arn

Obligatoire : non

Type: liste

## Prérequis

### clusters

Un [AWS.Compute.EKS](#) nœud.

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.Calculez. EKSMangedNode

AWS TNBprend en charge les groupes de nœuds EKS gérés pour automatiser le provisionnement et la gestion du cycle de vie des nœuds (EC2instances Amazon) pour les clusters Amazon EKS Kubernetes. Pour créer un groupe de EKS nœuds, procédez comme suit :

- Choisissez les Amazon Machine Images (AMI) pour les nœuds de travail de votre cluster en fournissant l'ID AMI ou le AMI type.
- Fournissez une paire de EC2 clés Amazon pour SSH l'accès et les propriétés de dimensionnement de votre groupe de nœuds.
- Assurez-vous que votre groupe de nœuds est associé à un EKS cluster Amazon.

- Fournissez 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 EKS gérés par Amazon, tels que les types d'EC2 instances Amazon et les EC2 instances AmazonAMIs.

## ami\_type

Le AMI type EKS pris en charge par Amazon.

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

L'identifiant du AMI.

Obligatoire : non

Type : String

### Note

Si ami\_type les deux ami\_id sont spécifiés dans le modèle, AWS TNB il utilisera uniquement la ami\_id valeur pour créerEKSMangedNode.

## instance\_types

Taille de l'instance.

Obligatoire : oui

Type: liste

## key\_pair

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

Obligatoire : oui

Type : String

## root\_volume\_encryption

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

Obligatoire : non

Valeur par défaut : true


Type : booléen

`root_volume_encryption_key_arn`

ARNLa AWS KMS clé. AWS TNBprend en charge la clé normaleARN, la clé multirégionale ARN et l'aliasARN.

Obligatoire : non

Type : String

 Note

- Si `root_volume_encryption` c'est faux, ne l'incluez pas `root_volume_encryption_key_arn`.
- AWS TNBprend en charge le chiffrement du volume racine des fichiers EBS soutenus par AMI Amazon.
- Si le volume racine AMI est déjà chiffré, vous devez inclure le `root_volume_encryption_key_arn` for AWS TNB pour rechiffrer le volume racine.
- Si le AMI volume racine 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 TNBne déchiffre pas un chiffréAMI.

## scaling

Propriétés qui définissent les paramètres de dimensionnement pour le groupe de nœuds EKS géré par Amazon, tels que le nombre souhaité d'EC2instances Amazon et le nombre minimum et maximum d'EC2instances Amazon 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`

Le ARN IAM rôle qui est attaché à l'EC2instance Amazon.

Obligatoire : oui

Type : String

`tags`

Les balises à associer à la ressource.

Obligatoire : non

Type: liste

## Prérequis

`cluster`

Un [AWS.Compute. EKS](#)nœud.

Obligatoire : oui

Type : String

## subnets

Un nœud [AWS.Networking.Subnet](#).

Obligatoire : oui

Type: liste

## network\_interfaces

Un [AWS.Networking.ENI](#) nœud. 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 associée à ENIs la `multus_role` propriété si vous l'avez incluse dans le `multus` [AWSfichier .Compute.EKS](#) nœud. Sinon, AWS TNB obtient l'autorisation associée à ENIs partir de la propriété [node\\_role](#).

Obligatoire : non

Type: liste

## security\_groups

Un [AWS.Networking.SecurityGroup](#) nœud.

Obligatoire : non

Type: liste

## placement\_group

Un [tosca.nodes.AWS.Calculez.PlacementGroup](#) nœud.

Obligatoire : non

Type : String

## user\_data

Un [tosca.nodes.AWS.Calculez.UserData](#) référence de nœud. Un script de données utilisateur est transmis aux EC2 instances Amazon 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 : String

## 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

```
SampleEKSMangedNode:
  type: toscanodes.AWS.Compute.EKSMangedNode
  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.Calculez. EKSSelfManagedNode

AWS TNBprend en charge les nœuds EKS autogérés Amazon pour automatiser le provisionnement et la gestion du cycle de vie des nœuds (EC2instances Amazon) pour les clusters Amazon EKS Kubernetes. Pour créer un groupe de EKS nœuds Amazon, procédez comme suit :

- Choisissez les Amazon Machine Images (AMI) pour les nœuds de travail de votre cluster en fournissant l'un ou l'autre des ID duAMI.
- Fournissez une paire de EC2 clés Amazon pour SSH y accéder.
- Assurez-vous que votre groupe de nœuds est associé à un EKS cluster Amazon.
- Indiquez le type d'instance et les tailles souhaitées, minimales et maximales.
- Fournissez 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 EKS autogérés par Amazon, tels que les types d'EC2instances Amazon et les EC2 instances AMIs Amazon.

#### ami\_id

AMIID utilisé pour lancer l'instance. AWS TNBprend en charge les instances qui tirent parti deIMDSv2. Pour de plus amples informations, veuillez consulter [IMDSversion](#).

Obligatoire : oui

Type : String

#### instance\_type

Taille de l'instance.

Obligatoire : oui

Type : String

## key\_pair

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

Obligatoire : oui

Type : String

## root\_volume\_encryption

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

Obligatoire : non

Valeur par défaut : true

Type : booléen

## root\_volume\_encryption\_key\_arn

ARNLa AWS KMS clé. AWS TNBprend en charge la clé normaleARN, la clé multirégionale ARN et l'aliasARN.

Obligatoire : non

Type : String

### Note

- Si `root_volume_encryption` c'est faux, ne l'incluez pas `root_volume_encryption_key_arn`.
- AWS TNBprend en charge le chiffrement du volume racine des fichiers EBS soutenus par AMI Amazon.
- Si le volume racine AMI est déjà chiffré, vous devez inclure le `root_volume_encryption_key_arn` for AWS TNB pour rechiffrer le volume racine.
- Si le AMI volume racine n'est pas chiffré, AWS TNB utilise le `root_volume_encryption_key_arn` pour chiffrer le volume racine.

Si vous n'incluez pas `root_volume_encryption_key_arn`, AWS TNB utilise AWS Managed Services pour chiffrer le volume racine.

- AWS TNBne déchiffre pas un chiffréAMI.

## ***scaling***

Propriétés qui définissent les paramètres de dimensionnement pour les nœuds EKS autogérés par Amazon, tels que le nombre souhaité d'EC2instances Amazon et le nombre minimum et maximum d'EC2instances Amazon 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`

Le ARN IAM rôle qui est attaché à l'EC2instance Amazon.

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 [AWS.Compute. EKS](#) nœud.

Obligatoire : oui

Type : String

### subnets

Un nœud [AWS.Networking.Subnet](#).

Obligatoire : oui

Type: liste

### network\_interfaces

Un [AWS.Networking. ENI](#) nœud. 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 associée à ENIs la `multus_role` propriété si vous l'avez incluse dans le `multus` [AWSfichier .Compute. EKS](#) nœud. Sinon, AWS TNB obtient l'autorisation associée à ENIs partir de la propriété [node\\_role](#).

Obligatoire : non

Type: liste

### security\_groups

Un [AWS.Networking. SecurityGroup](#) nœud.

Obligatoire : non

Type: liste

## placement\_group

Un [tosca.nodes.AWS.Calculez.PlacementGroup](#) nœud.

Obligatoire : non

Type : String

## user\_data

Un [tosca.nodes.AWS.Calculez.UserData](#) référence de nœud. Un script de données utilisateur est transmis aux EC2 instances Amazon 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 : String

## 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: toasca.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.Calculiez. PlacementGroup

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

Lorsque vous lancez un nouvel AmazonEC2instance, le EC2 service Amazon 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 des EC2 instances Amazon.

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 des partitions différentes. 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 surPARTITION.

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

## AWS.Calculez. UserData

AWS TNBprend en charge le lancement d'EC2instances Amazon avec des données utilisateur personnalisées, via le UserData nœud dans 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 EC2 l'utilisateur Amazon.

Lors de l'instanciation du réseau, AWS TNB fournit l'enregistrement de l'EC2instance Amazon 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 EKS d'enregistrement Amazon.

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 TNBprend en charge les scripts de données utilisateur d'une taille maximale de 7 Ko.

- Dans la mesure où il est AWS TNB utilisé AWS CloudFormation pour traiter et afficher le script de multimime données utilisateur, assurez-vous que le script respecte toutes AWS CloudFormation les règles.

## 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: toasca.nodes.AWS.Compute.UserData  
  properties:  
    content_type: "text/x-shellscript"  
    implementation: "./scripts/customUserData.sh"
```

## AWS.Réseautage. SecurityGroup

AWS TNBprend en charge les groupes de sécurité pour automatiser le provisionnement des [groupes de EC2 sécurité Amazon](#) 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 [AWS.Networking.VPC](#) nœud.

Obligatoire : oui

Type : String

## Exemple

```
SampleSecurityGroup001:
  type: toscanodes.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 EC2 de sécurité Amazon 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'IPv4adresses au CIDR format. Vous devez spécifier une CIDR plage qui autorise le trafic sortant.

Obligatoire : non

Type : String

### `cidr_ipv6`

La plage d'IPv6adresses au CIDR format, pour le trafic sortant. Vous devez spécifier un groupe de sécurité de destination (`destination_security_group`ou`destination_prefix_list`) ou une CIDR plage (`cidr_ip`ou`cidr_ipv6`).

Obligatoire : non

Type : String

### `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 : String

### `destination_prefix_list`

L'ID de liste de préfixes d'une liste de préfixes VPC gérée par Amazon 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 VPC l'utilisateur Amazon.

Obligatoire : non

Type : String

`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 ICMPv6 types ICMP /. Si vous spécifiez tous les ICMPv6 types ICMP/, vous devez spécifier tous les ICMPv6 codes ICMP /.

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, voici le code. La valeur -1 indique tous les ICMPv6 codes ICMP /. Si vous spécifiez tous les ICMPv6 types ICMP/, vous devez spécifier tous les ICMPv6 codes ICMP /.

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

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

Obligatoire : non

Type : String

## 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 EC2 de sécurité Amazon 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'IPv4adresses au CIDR format. Vous devez spécifier une CIDR plage qui autorise le trafic entrant.

Obligatoire : non

Type : String

### cidr\_ipv6

La plage d'IPv6adresses au CIDR format, pour le trafic entrant. Vous devez spécifier un groupe de sécurité source (`source_security_group`ou`source_prefix_list`) ou une CIDR plage (`cidr_ip`ou`cidr_ipv6`).

Obligatoire : non

Type : String

### 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 : String

### source\_prefix\_list

L'ID de liste de préfixes d'une liste de préfixes VPC gérée par Amazon 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 VPC l'utilisateur Amazon.

Obligatoire : non

Type : String

`from_port`

Si le protocole est TCP ouUDP, il s'agit du début de la plage de ports. Si le protocole est ICMP ouICMPv6, il s'agit du numéro de type. La valeur -1 indique tous les ICMPv6 types ICMP /. Si vous spécifiez tous les ICMPv6 typesICMP/, vous devez spécifier tous les ICMPv6 codes ICMP /.

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 ouUDP, il s'agit de la fin de la plage de ports. Si le protocole est ICMP ouICMPv6, voici le code. La valeur -1 indique tous les ICMPv6 codes ICMP /. Si vous spécifiez tous les ICMPv6 typesICMP/, vous devez spécifier tous les ICMPv6 codes ICMP /.

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

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

Obligatoire : non

Type : String

## Exemple

```
SampleSecurityGroupIngressRule:
  type: toska.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 vers AWS TNB.

Obligatoire : non

Type: liste

### resource\_id

L'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.Réseautage. 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 EC2 instance Amazon dans un sous-réseau, vous pouvez y associer une interface réseau ou détacher une interface réseau de cette EC2 instance Amazon et la rattacher à une autre EC2 instance Amazon 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. SecurityGroup](#) nœud.

Obligatoire : non

Type : String

## Exemple

```
SampleENI:
  type: toska.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 : String

Valeurs possibles : CODE\_BUILD

### Prérequis

#### definition

Un [AWS. HookDefinition. Nœud Bash](#).

Obligatoire : oui

Type : String

#### vpc

Un [AWS.Networking. VPC](#)nœud.

Obligatoire : oui

Type : String

### Exemple

```
SampleHookExecution:
  type: toasca.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`

Le IPv4 CIDR bloc utilisé pour le match 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 : String

#### `ipv6_dest_cidr`

Le IPv6 CIDR bloc utilisé pour le match de destination.

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

Type : String



## Propriétés

### tags

Les balises à associer à la ressource.

Obligatoire : non

Type: liste

### egress\_only

Une propriété IPv6 spécifique. 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 [AWS.Networking.VPC](#) nœud.

Obligatoire : oui

Type : String

### route\_table

Un [AWS.Networking.RouteTable](#) nœud.

Obligatoire : oui

Type : String

## Exemple

```
Free5GCIGW:  
  type: toasca.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: toska.nodes.AWS.Networking.InternetGateway
  properties:
    egress_only: true
capabilities:
  routing:
    properties:
      ipv6_dest_cidr: "::/0"
requirements:
  route_table: Free5GCPriateRouteTable
  vpc: Free5GCVPC
```

## AWS.Réseautage. RouteTable

Une table de routage contient un ensemble de règles, appelées routes, qui déterminent où est dirigé le trafic réseau provenant des sous-réseaux de votre passerelle VPC ou de votre passerelle. Vous devez associer une table de routage à unVPC.

### 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 [AWS.Networking.VPC](#) nœud.

Obligatoire : oui

Type : String

## Exemple

```
SampleRouteTable:
  type: toska.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 dans votre VPC, et il doit résider entièrement dans une seule zone de disponibilité. Vous devez spécifier un VPC, un CIDR bloc, 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

```
toska.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 lancées dans ce sous-réseau reçoivent une IPv4 adresse 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 CIDR bloc du sous-réseau.

Obligatoire : non

Type : String

### ipv6\_cidr\_block

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

Obligatoire : non

Type : String

### ipv6\_cidr\_block\_suffix

Le suffixe hexadécimal à 2 chiffres du IPv6 CIDR bloc pour le sous-réseau créé sur 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 : String

`outpost_arn`

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

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

Obligatoire : non

Type : String

`tags`

Les balises à associer à la ressource.

Obligatoire : non

Type: liste

## Prérequis

`vpc`

Un [AWS.Networking.VPC](#) nœud.

Obligatoire : oui

Type : String

`route_table`

Un [AWS.Networking.RouteTable](#) nœud.

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.Déploiement. VNFDeployment

Les déploiements NF sont modélisés en fournissant l'infrastructure et l'application qui y sont associées. L'attribut [cluster](#) indique le EKS cluster qui hébergera votre NFs. 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'un système de gestion des stocks. API

## Syntaxe

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

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

## Prérequis

### deployment

Un [AWS.Deployment. VNFDeployment](#)nœud.

Obligatoire : non

Type : String

### cluster

Un [AWS.Compute. EKS](#)nœud.

Obligatoire : oui

Type : String

### vnfs

Un [AWS. VNF](#)nœud.

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. HookExecution](#)nœud. Ce hook est exécuté avant le déploiement du VNFDeployment nœud.

Obligatoire : non

Type : String

post\_create

Un [AWS.HookExecution](#) nœud. Ce hook est exécuté après le déploiement du VNFDeployment nœud.

Obligatoire : non

Type : String

## Exemple

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

## AWS.Réseautage. VPC

Vous devez spécifier un CIDR bloc 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 plage de IPv4 réseau pour leVPC, en CIDR notation.



Obligatoire : oui

Type : String

ipv6\_cidr\_block

Le IPv6 CIDR bloc utilisé pour créer leVPC.

Valeur autorisée : AMAZON\_PROVIDED

Obligatoire : non

Type : String

dns\_support

Indique si les instances ont été lancées dans le VPC get DNS hostnames.

Obligatoire : non

Type : booléen

Par défaut : false

tags

Balises à associer à la ressource.

Obligatoire : non

Type: liste

## Exemple

```
SampleVPC:
  type: toska.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.Réseautage. NATGateway

Vous pouvez définir un nœud de NAT passerelle public ou privé sur un sous-réseau. Pour une passerelle publique, si vous ne fournissez pas d'identifiant d'allocation IP élastique, vous AWS TNB attribuerez une adresse IP élastique à votre compte et l'associez 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. InternetGateway](#) référence 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 : String

`tags`

Balises à associer à la ressource.

Obligatoire : non

Type: liste

## Exemple

```
Free5GNatGateway01:
  type: toska.nodes.AWS.Networking.NATGateway
  requirements:
    subnet: Free5GSubnet01
    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 NAT passerelle 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 IPv4 itinéraires de destination vers la ressource cible.

Obligatoire : oui

Type: liste

Type de membre : Chaîne

## Propriétés

### nat\_gateway

Le [AWS.Networking. NATGateway](#) référence de nœud.

Obligatoire : oui

Type : String

### route\_table

Le [AWS.Networking. RouteTable](#) référence de nœud.

Obligatoire : oui

Type : String

## Exemple

```
Free5GCRoute:  
  type: toasca.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

Définissez les nœuds pour le NSD etVNFD.

- [AWS. HookDefinition.Bash](#)

### AWS.HookDefinition.Bash

Définit une AWS HookDefinition entréebash.

#### Syntaxe

```
tosca.nodes.AWS.HookDefinition.Bash:
  properties:
    implementation: String
    environment\_variables: List
    execution\_role: String
```

#### Propriétés

##### implementation

Le chemin relatif vers la définition du crochet. 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 l'expression régulière 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 la **envName=envValue** valeur répond aux critères suivants :

- N'utilisez pas d'espaces.

- **envName** Commencez par une lettre (A-Z ou a-z) ou un chiffre (0-9).
- Ne commencez pas le nom de la variable d'environnement par les mots clés AWS TNB réservés suivants (sans distinction majuscules/majuscules) :
  - CODEBUILD
  - TNB
  - HOME
  - AWS
- Vous pouvez utiliser n'importe quel nombre de lettres (A-Z ou a-z), de chiffres (0-9), de caractères spéciaux et pour - et\_. **envName 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: tosa.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 TNB

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 son utilisation AWS TNB. Les rubriques suivantes expliquent comment procéder à la configuration AWS TNB pour atteindre 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 AWS TNB ressources.

## Table des matières

- [Protection des données dans AWS TNB](#)
- [Gestion des identités et des accès pour AWS TNB](#)
- [Validation de conformité pour AWS TNB](#)
- [Résilience dans AWS TNB](#)
- [Sécurité de l'infrastructure dans AWS TNB](#)
- [IMDSversion](#)

# Protection des données dans 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 la section [Confidentialité des données FAQ](#). Pour plus d'informations sur la protection des données en Europe, consultez le [modèle de responsabilité AWS partagée et le billet de GDPR blog sur le blog sur la AWS sécurité](#).

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) pour chaque compte.
- Utilisez SSL/TLS pour communiquer avec les AWS ressources. Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Configuration API et journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section [Utilisation des CloudTrail sentiers](#) dans le guide de AWS CloudTrail l'utilisateur.
- 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 FIPS 140 à 3 modules cryptographiques validés pour accéder AWS via une interface de ligne de commande ou un API, utilisez un point de terminaison. FIPS Pour plus d'informations sur les FIPS points de terminaison disponibles, voir [Federal Information Processing Standard \(FIPS\) 140-3](#).

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 Nom. Cela inclut lorsque vous travaillez avec AWS TNB ou d'autres Services AWS utilisateurs de la console API, AWS CLI, ou AWS SDKs. 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 un URL à un serveur externe, nous vous recommandons vivement de ne pas inclure d'informations d'identification dans le URL afin de valider votre demande auprès de 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, vos données AWS TNB seront restaurées. Supprimez AWS TNB définitivement vos données au bout de 120 jours. 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 de Transport Layer Security (TLS) 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 TNB Les ressources informatiques se trouvent dans un cloud privé virtuel (VPC) partagé par tous les clients. Tout AWS TNB le trafic interne est resté sur le AWS réseau et n'a pas transité 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. IAM les administrateurs contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les AWS TNB ressources. IAM est un Service AWS ventilateur que vous pouvez utiliser sans frais supplémentaires.

### Table des matières

- [Public ciblé](#)

- [Authentification par des identités](#)
- [Gestion des accès à l'aide de politiques](#)
- [Comment AWS TNB fonctionne avec IAM](#)
- [Exemples de politiques basées sur l'identité pour AWS Telco Network Builder](#)
- [Résolution des problèmes d'identité et d'accès à AWS Telco Network Builder](#)

## Public ciblé

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

Utilisateur du service : si vous utilisez le AWS TNB service 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 nouvelles AWS TNB fonctionnalités 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, consultez [Résolution des problèmes d'identité et d'accès à AWS Telco Network Builder](#).

Administrateur du service — Si vous êtes responsable des AWS TNB ressources de votre entreprise, vous avez probablement un accès complet à AWS TNB. C'est à vous de déterminer les AWS TNB fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite envoyer des demandes à votre IAM administrateur pour modifier les autorisations des utilisateurs de votre service. Consultez les informations de cette page pour comprendre les concepts de base de IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec AWS TNB, voir [Comment AWS TNB fonctionne avec IAM](#).

IAM administrateur — Si vous êtes IAM administrateur, vous souhaitez 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 AWS TNB basées sur l'identité 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 que Utilisateur racine d'un compte AWS, en tant qu'IAM utilisateur ou en assumant un IAM rôle.

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 en tant qu'identité fédérée, votre administrateur a préalablement configuré la fédération d'identité à l'aide de IAM rôles. 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 [version 4 de AWS Signature pour les API demandes](#) dans le guide de IAM l'utilisateur.

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, voir [Authentification multifactorielle](#) dans le guide de l'AWS IAM Identity Center utilisateur et [Authentification AWS multifactorielle IAM dans](#) le guide de l'IAMutilisateur.

## 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 nécessitent que vous vous connectiez en tant qu'utilisateur root, consultez la section [Tâches nécessitant des informations d'identification utilisateur root](#) dans le Guide de IAM l'utilisateur.

## 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 plus d'informations sur IAM Identity Center, consultez [Qu'est-ce qu'IAM Identity Center ?](#) dans le guide de AWS IAM Identity Center l'utilisateur.

## Utilisateurs et groupes IAM

Un [IAMutilisateur](#) est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des IAM utilisateurs dotés d'informations d'identification à long terme, telles que des mots de passe et des clés d'accès. Toutefois, si vous avez des cas d'utilisation spécifiques qui nécessitent des informations d'identification à long terme auprès des IAM utilisateurs, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, voir [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification à long terme](#) dans le Guide de IAM l'utilisateur.

Un [IAMgroupe](#) est une identité qui définit un ensemble d'IAMutilisateurs. 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 nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer IAM des ressources.

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 la section [Cas d'utilisation pour IAM les utilisateurs](#) dans le Guide de IAM l'utilisateur.

## IAM rôles

Un [IAM rôle](#) est une identité au sein de Compte AWS vous dotée d'autorisations spécifiques. Il est similaire à un IAM utilisateur, mais n'est pas associé à une personne en particulier. Pour assumer temporairement un IAM rôle dans le AWS Management Console, vous pouvez [passer d'un rôle d'utilisateur à un IAM rôle \(console\)](#). Vous pouvez assumer un rôle en appelant une AWS API opération AWS CLI or ou en utilisant une option personnalisée URL. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section [Méthodes pour assumer un rôle](#) dans le Guide de IAM l'utilisateur.

IAM les rôles dotés d'informations d'identification temporaires sont utiles dans les situations suivantes :

- 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 plus d'informations sur les rôles pour la fédération, voir [Création d'un rôle pour un fournisseur d'identité tiers](#) dans le guide de IAM l'utilisateur. Si vous utilisez IAM Identity Center, vous configurez un ensemble d'autorisations. Pour contrôler les accès auxquels vos identités peuvent accéder après leur authentification, IAM Identity Center met en corrélation l'ensemble d'autorisations avec un rôle dans IAM. Pour plus d'informations sur les jeux d'autorisations, consultez [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations IAM utilisateur temporaires : un IAM utilisateur ou un rôle peut assumer un IAM rôle afin d'obtenir temporairement différentes autorisations pour une tâche spécifique.
- Accès entre comptes : vous pouvez utiliser un IAM rôle pour autoriser une personne (un mandant fiable) d'un autre compte à 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 connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux [ressources entre comptes IAM dans le guide](#) de l'IAM utilisateur.
- 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 transmises (FAS) — Lorsque vous utilisez un IAM utilisateur ou un rôle 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 au Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FAS les demandes 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 les politiques relatives FAS aux demandes, consultez la section [Transférer les sessions d'accès](#).
- Rôle de service — Un rôle de service est un [IAM rôle](#) qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieur IAM. Pour plus d'informations, consultez [la section Création d'un rôle auquel déléguer des autorisations Service AWS](#) dans le Guide de IAM l'utilisateur.
- 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 IAM administrateur peut consulter, mais pas modifier les autorisations pour les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un IAM rôle pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui font AWS CLI des AWS API demandes. Cela est préférable au stockage des clés d'accès dans l'EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour 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 exécutés sur l'EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez la section [Utilisation d'un IAM rôle pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon](#) dans le Guide de IAM l'utilisateur.

## 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 JSON documents. Pour plus d'informations sur la structure et le contenu des documents de JSON politique, voir [Présentation des JSON politiques](#) dans le guide de IAM l'utilisateur.

Les administrateurs peuvent utiliser AWS JSON des politiques 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 autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

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

## Politiques basées sur l'identité

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir [Définir des IAM autorisations personnalisées avec des politiques gérées par le client](#) dans le Guide de l'IAM utilisateur.

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 savoir comment choisir entre une politique gérée ou une politique intégrée, voir [Choisir entre les politiques gérées et les politiques intégrées dans le Guide](#) de l'IAM utilisateur.

## Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux

compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. 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 de politiques AWS gérées depuis une IAM stratégie basée sur les ressources.

## Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

Amazon S3 et Amazon VPC sont des exemples de services compatibles ACLs. AWS WAF Pour en savoir plus ACLs, consultez la [présentation de la liste 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.

- Limites d'autorisations — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité (IAM utilisateur ou rôle). 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 annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez la section Limites d'[autorisations pour les IAM entités](#) dans le Guide de IAM l'utilisateur.
- Politiques de contrôle des services (SCPs) : SCPs JSON politiques 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 des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Les SCP limites d'autorisations pour les entités présentes dans les comptes des membres, y compris chacune d'entre elles Utilisateur racine d'un compte AWS. Pour plus d'informations sur les Organisations et consultez SCPs les [politiques de contrôle des services](#) dans le Guide de AWS Organizations l'utilisateur.

- **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 la section [Politiques de session](#) dans le guide de IAM l'utilisateur.

## 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 IAM l'utilisateur.

## Comment AWS TNB fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS TNB, découvrez quelles IAM fonctionnalités sont disponibles AWS TNB.

IAM fonctionnalités que vous pouvez utiliser avec AWS Telco Network Builder

IAM fonctionnalité	AWS TNB soutien
<a href="#">Politiques basées sur l'identité</a>	Oui
<a href="#">Politiques basées sur les ressources</a>	Non
<a href="#">Actions de politique</a>	Oui
<a href="#">Ressources de politique</a>	Oui
<a href="#">Clés de condition de politique</a>	Oui

IAM fonctionnalité	AWS TNB soutien
<a href="#">ACLs</a>	Non
<a href="#">ABAC(balises dans les politiques)</a>	Oui
<a href="#">Informations d'identification temporaires</a>	Oui
<a href="#">Autorisations de principal</a>	Oui
<a href="#">Fonctions du service</a>	Non
<a href="#">Rôles liés à un service</a>	Non

Pour obtenir une vue d'ensemble du fonctionnement de la plupart des IAM fonctionnalités AWS TNB et des autres AWS services, reportez-vous à la section [AWS Services compatibles IAM](#) dans le Guide de IAM l'utilisateur.

## Politiques basées sur l'identité pour AWS TNB

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir [Définir des IAM autorisations personnalisées avec des politiques gérées par le client](#) dans le Guide de l'IAM utilisateur.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier les actions et les 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 en savoir plus sur tous les éléments que vous pouvez utiliser dans une JSON politique, consultez la [référence aux éléments de IAM JSON politique](#) dans le Guide de IAM l'utilisateur.

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

Pour consulter des exemples de politiques AWS TNB basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS Telco Network Builder](#)

## Politiques basées sur les ressources au sein de AWS TNB

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. 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 activer l'accès entre comptes, vous pouvez spécifier un compte entier ou IAM des entités d'un autre compte comme 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 IAM administrateur du compte de confiance 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 [la section Accès aux ressources entre comptes IAM dans](#) le Guide de IAM l'utilisateur.

## Actions politiques pour AWS TNB

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser AWS JSON des politiques 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'Actionélément d'une JSON politique décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès dans une politique. Les actions de stratégie portent généralement le même nom que l' AWS APIopération associée. Il existe certaines exceptions, telles que les actions avec autorisation uniquement qui n'ont pas d'opération correspondante. API Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

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

Pour consulter la liste des AWS TNB actions, reportez-vous à la section [Actions définies par AWS Telco Network Builder](#) dans la référence d'autorisation de service.

Les actions de politique en AWS TNB cours 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 AWS TNB basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS Telco Network Builder](#)

## Ressources politiques pour AWS TNB

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser AWS JSON des politiques 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 `Resource` JSON de stratégie indique le ou les objets auxquels s'applique l'action. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de spécifier une ressource en utilisant 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 AWS TNB ressources et leurs caractéristiques ARNs, 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 pour chaque ressource, voir [Actions définies par AWS Telco Network Builder](#). ARN

Pour consulter des exemples de politiques AWS TNB basées sur l'identité, consultez. [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 AWS JSON des politiques 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 autoriser un IAM utilisateur à accéder à une ressource uniquement si celle-ci est étiquetée avec son nom IAM d'utilisateur. Pour plus d'informations, consultez [IAM la section Éléments de politique : variables et balises](#) dans le Guide de IAM l'utilisateur.

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 contextuelles de condition AWS globales](#) dans le guide de IAM l'utilisateur.

Pour consulter la liste des clés de AWS TNB condition, 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 AWS TNB basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS Telco Network Builder](#)

## ACLs dans AWS TNB

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

## ABAC avec AWS TNB

Supports ABAC (balises dans les politiques) : Oui

Le contrôle d'accès basé sur les attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises à IAM des entités (utilisateurs ou rôles) et à de nombreuses AWS ressources. Le balisage des entités et des ressources est la première étape de ABAC. Vous concevez ensuite des ABAC politiques pour autoriser les opérations lorsque le tag du principal correspond à celui de la ressource à laquelle il essaie d'accéder.

ABAC est utile dans les environnements qui se développent rapidement et aide dans les situations 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 ABAC, voir [Définir des autorisations avec ABAC autorisation](#) dans le Guide de IAM l'utilisateur. Pour consulter un didacticiel présentant les étapes de configuration ABAC, voir [Utiliser le contrôle d'accès basé sur les attributs \(ABAC\)](#) dans le guide de l'IAM utilisateur.

## 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 IAM](#) dans le Guide de IAM l'utilisateur.

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, voir [Passer d'un utilisateur à un IAM rôle \(console\)](#) dans le guide de IAM l'utilisateur.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide du AWS CLI ou AWS API. 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 la section Informations [d'identification de sécurité temporaires dans IAM](#).

## Autorisations principales interservices pour AWS TNB

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

Lorsque vous utilisez un IAM utilisateur ou un rôle 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 an Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FAS les demandes 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 les politiques relatives FAS aux demandes, consultez la section [Transférer les sessions d'accès](#).

## Rôles de service pour AWS TNB

Supporte les rôles de service : Non

Un rôle de service est un [IAMrôle](#) qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieurIAM. Pour plus d'informations, consultez [la section Création d'un rôle auquel déléguer des autorisations Service AWS](#) dans le Guide de IAM l'utilisateur.

## Rôles liés à un service pour AWS TNB

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 IAM administrateur peut consulter, mais pas modifier les autorisations pour 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 AWS TNB des ressources. Ils ne peuvent pas non plus effectuer de tâches en utilisant le AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

Pour savoir comment créer une politique IAM basée sur l'identité à l'aide de ces exemples de documents de JSON stratégie, voir [Créer des IAM politiques \(console\)](#) dans le guide de l'IAMutilisateur.

Pour plus de détails sur les actions et les types de ressources définis par AWS TNB, y compris le format de ARNs 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](#)
- [Utilisation de la AWS TNB console](#)
- [Exemples de politiques relatives aux rôles de service](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)

## Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer AWS TNB des ressources 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 [les politiques AWS gérées ou les politiques AWS gérées pour les fonctions professionnelles](#) dans le Guide de IAM l'utilisateur.
- Appliquer les autorisations du moindre privilège : lorsque vous définissez des autorisations à IAM l'aide de politiques, accordez uniquement les autorisations nécessaires à l'exécution d'une 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 IAM pour appliquer des autorisations, consultez la section [Politiques et autorisations IAM dans](#) le guide de IAM l'utilisateur.
- Utilisez des conditions dans IAM les politiques pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques pour limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez rédiger une condition de politique pour spécifier que toutes les demandes doivent être envoyées en utilisant 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, voir [Éléments IAM JSON de politique : Condition](#) dans le guide de IAM l'utilisateur.
- Utilisez IAM Access Analyzer pour valider vos IAM politiques afin de garantir des autorisations sécurisées et fonctionnelles. IAM Access Analyzer valide les politiques nouvelles et existantes afin qu'elles respectent le langage des politiques (JSON) et IAM les IAM meilleures pratiques. IAM Access Analyzer fournit plus de 100 vérifications des politiques et des recommandations

exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez la section [Valider les politiques avec IAM Access Analyzer](#) dans le guide de l'IAMUtilisateur.

- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des IAM utilisateurs ou un utilisateur root Compte AWS, activez-le MFA pour une sécurité supplémentaire. Pour exiger le MFA moment où les API opérations sont appelées, ajoutez MFA des conditions à vos politiques. Pour plus d'informations, consultez la section [APIAccès sécurisé avec MFA](#) dans le guide de IAM l'utilisateur.

Pour plus d'informations sur les meilleures pratiques en matière de [sécuritéIAM](#), consultez la section [Bonnes pratiques en matière](#) de sécurité IAM dans le Guide de IAM l'utilisateur.

## Utilisation de la AWS TNB console

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 AWS TNB des ressources de votre Compte AWS. Si vous créez une politique 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 politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement le AWS CLI ou le AWS API. Au lieu de cela, autorisez uniquement l'accès aux actions correspondant à l'APIopération qu'ils tentent d'effectuer.

## Exemples de politiques relatives aux rôles de service

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

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

Vous créez le rôle de service et sa politique d'autorisation avec le IAM service. 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 IAM l'utilisateur.

## AWS TNBrôle de service

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

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

Le code suivant illustre une politique AWS TNB de rôle de service :

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

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

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

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

```

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

```

```

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

```

Le code suivant illustre la politique AWS TNB de confiance du service :

```

{
  "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 TNBrôle de service pour le EKS cluster Amazon

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

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

```

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

```

```

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

```

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

- [AWS : IAM : Rôle](#)
- [Sélection d'un modèle de pile](#)

### AWS TNBrôle de service pour le groupe de EKS nœuds Amazon

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

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

```

AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBEKSNodeRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBEKSNodeRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /

```

```

ManagedPolicyArns:
  - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSEKSWorkerNodePolicy"
  - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSCNI_Policy"
  - !Sub "arn:${AWS::Partition}:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly"
  - !Sub "arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEBSCSIDriverPolicy"
Policies:
  - PolicyName: EKSEKSNodeRoleInlinePolicy
    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: EKSEKSNodeRoleIpv6CNIPolicy
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: Allow
          Action:
            - "ec2:AssignIpv6Addresses"
          Resource: "arn:aws:ec2:*:*:network-interface/*"

```

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

- [AWS : IAM : Rôle](#)
- [Sélection d'un modèle de pile](#)

## AWS TNBrôle de service pour Multus

Lorsque vous créez une EKS ressource Amazon 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 AWS TNB service 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 IAM les rôles utilisant AWS CloudFormation un modèle, consultez les sections suivantes du guide de l'AWS CloudFormation utilisateur :

- [AWS: : IAM : :Rôle](#)
- [Sélection d'un modèle de pile](#)

AWS TNBrôle de service pour une politique d'accrochage du cycle de vie

Lorsque votre package de fonctions réseau 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 AWS TNB service 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 IAM les rôles utilisant AWS CloudFormation un modèle, consultez les sections suivantes du guide de l'AWS CloudFormation utilisateur :

- [AWS: : IAM : :Rôle](#)
- [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 IAM utilisateurs de consulter les politiques intégrées et gérées associées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide du AWS CLI ou. AWS API

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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](#)
- [Je ne suis pas autorisé à effectuer iam : PassRole](#)
- [Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes AWS TNB ressources](#)

### 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 lorsque l'utilisateur IAM `mateojacksonIAMutilisateur` essaie d'utiliser la console pour afficher les détails d'une `my-example-widget` ressource fictive mais ne dispose pas des `tnb:GetWidget` autorisations 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 AWS TNB.

Certains 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 IAM utilisateur 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 AWS TNB ressources

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 (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises AWS TNB en charge, consultez [Comment AWS TNB fonctionne avec IAM](#).
- Pour savoir comment donner accès à vos ressources sur un site Comptes AWS qui vous appartient, consultez la section [Fournir l'accès à un IAM utilisateur dans un autre site Compte AWS que vous possédez](#) dans le Guide de IAM l'utilisateur.



- 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 IAM l'utilisateur.
- Pour savoir comment fournir un accès via la fédération d'identité, consultez la section [Fournir un accès aux utilisateurs authentifiés de manière externe \(fédération d'identité\)](#) dans le guide de l'IAMutilisateur.
- Pour connaître la différence entre l'utilisation de rôles et de politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux [ressources entre comptes IAM dans le guide](#) de l'IAMutilisateur.

## 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 fournissent des étapes pour déployer des environnements de base axés sur AWS la sécurité et la conformité.
- [Architecture axée sur la HIPAA sécurité et la conformité sur Amazon Web Services](#) : ce livre blanc décrit comment les entreprises peuvent AWS créer HIPAA des applications éligibles.

### Note

Tous ne Services AWS sont pas HIPAA éligibles. Pour plus d'informations, consultez la [référence des services HIPAA éligibles](#).

- 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 résumant les meilleures pratiques en matière de sécurisation Services AWS et reprennent 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](#).
- [Amazon GuardDuty](#) — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité PCIDSS, par exemple en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- [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 dans 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 EKS des clusters dans un cloud privé virtuel (VPC) dans la AWS région de votre choix.

## Sécurité de l'infrastructure dans 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 API les appels AWS publiés pour accéder AWS TNB via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Sécurité de la couche de transport (TLS). Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Des suites de chiffrement parfaitement confidentielles (PFS) telles que (Ephemeral Diffie-Hellman) ou DHE 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 identifiant de clé d'accès et d'une clé d'accès secrète associés à un IAM principal. 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 qui prennent en charge AWS TNB, notamment :
  - Instances de calcul (également appelées « travailleurs »)
  - Bases de données internes
  - Communications réseau entre les composants internes
  - L'interface de programmation de l' AWS TNB application (API)
  - 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) :
  - IAM utilisateurs, groupes, rôles et politiques
  - Les compartiments S3 que vous utilisez pour stocker vos données AWS TNB
  - Autres ressources Services AWS et ressources que vous utilisez pour prendre en charge le service réseau que vous avez fourni via AWS TNB
  - Le code de votre application
  - Connexions entre le service réseau que vous avez fourni 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 par le biais AWS TNB de ce dernier.

## Modèle de sécurité de connectivité réseau

Les services réseau que vous fournissez s'exécutent sur des instances de calcul au sein d'un cloud privé virtuel (VPC) situé dans une AWS région que vous sélectionnez. AWS TNB A 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 VPCs internes 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 à tous AWS TNB les clients opérant dans la même région partagent les mêmes VPC services. Les services réseau fournis par le biais AWS TNB de différents clients utilisent des instances de calcul distinctes au sein d'une même VPC instance.

Les communications entre les clients de votre service réseau et votre service réseau 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 AWS TNB via le AWS Management Console, AWS Command Line Interface (AWS CLI) et AWS SDKs sont cryptées.

## IMDSversion

AWS TNB prend en charge les instances qui exploitent le service de métadonnées d'instance version 2 (IMDSv2), une méthode axée sur les sessions. IMDSv2 inclut un niveau de sécurité supérieur à IMDSv1. Pour plus d'informations, consultez Renforcer [la défense contre les pare-feux ouverts, les proxys inverses et les SSRF vulnérabilités grâce aux améliorations apportées au service Amazon EC2 Instance Metadata](#).

Lorsque vous lancez votre instance, vous devez utiliser IMDSv2. Pour plus d'informations IMDSv2, consultez la section [Utilisation IMDSv2](#) dans le guide de EC2 l'utilisateur Amazon.

# Surveillance AWS TNB

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité AWS TNB et des performances de vos autres AWS solutions. AWS permet AWS CloudTrail de surveiller AWS TNB, de signaler en cas de problème et de prendre des mesures automatiques le cas échéant.

CloudTrail À utiliser pour capturer des informations détaillées sur les appels passés à AWS APIs. Vous pouvez stocker ces appels sous forme de fichiers journaux dans Amazon S3. Vous pouvez utiliser ces CloudTrail 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.

Les CloudTrail journaux contiennent des informations sur les appels à l'APIaction pour AWS TNB. Ils contiennent également des informations relatives aux appels à l'APIaction émanant de services tels qu'Amazon EC2 et AmazonEBS.

## Enregistrement des API appels AWS Telco Network Builder à l'aide de AWS CloudTrail

AWS Telco Network Builder est intégré à [AWS CloudTrail](#) un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS. CloudTrail capture tous les API appels AWS TNB sous forme d'événements. Les appels capturés incluent des appels provenant de la AWS TNB console et des appels de code vers les AWS TNB API opérations. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite AWS TNB, l'adresse IP à partir de laquelle la demande a été faite, la date à laquelle elle a été faite et des informations supplémentaires.

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 :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou d'utilisateur root.
- Si la demande a été faite au nom d'un utilisateur de IAM l'Identity Center.
- 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.

CloudTrail est actif dans votre compte Compte AWS lorsque vous créez le compte et vous avez automatiquement accès à l'historique des CloudTrail événements. L'historique des CloudTrail événements fournit un enregistrement consultable, consultable, téléchargeable et immuable des 90 derniers jours des événements de gestion enregistrés dans un. Région AWS Pour plus d'informations, consultez la section [Utilisation de l'historique des CloudTrail événements](#) dans le guide de AWS CloudTrail l'utilisateur. La consultation de CloudTrail l'historique des événements est gratuite.

Pour un enregistrement continu des événements de vos 90 Compte AWS derniers jours, créez un magasin de données sur les événements de Trail ou [CloudTrailLake](#).

## CloudTrail sentiers

Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Tous les sentiers créés à l'aide du AWS Management Console sont multirégionaux. Vous pouvez créer un parcours à région unique ou multirégionale à l'aide du. AWS CLI Il est recommandé de créer un parcours multirégional, car vous capturez l'activité dans l'ensemble Régions AWS de votre compte. Si vous créez un parcours à région unique, vous ne pouvez voir que les événements enregistrés dans le parcours. Région AWS Pour plus d'informations sur les sentiers, consultez les [sections Création d'un sentier pour votre organisation](#) Compte AWS et [Création d'un sentier pour une organisation](#) dans le guide de AWS CloudTrail l'utilisateur.

Vous pouvez envoyer une copie de vos événements de gestion en cours dans votre compartiment Amazon S3 gratuitement CloudTrail en créant un journal. Toutefois, des frais de stockage Amazon S3 sont facturés. Pour plus d'informations sur la CloudTrail tarification, consultez la section [AWS CloudTrail Tarification](#). Pour obtenir des informations sur la tarification Amazon S3, consultez [Tarification Amazon S3](#).

## CloudTrail Stockages de données sur les événements du lac

CloudTrail Lake vous permet d'exécuter SQL des requêtes basées sur vos événements. CloudTrail Lake convertit les événements existants au JSON format basé sur les lignes au ORC format [Apache](#). ORCest un format de stockage en colonnes optimisé pour une extraction rapide des données. Les événements sont agrégés dans des magasins de données d'événement. Ceux-ci constituent des collections immuables d'événements basées sur des critères que vous sélectionnez en appliquant des [sélecteurs d'événements avancés](#). Les sélecteurs que vous appliquez à un magasin de données d'événement contrôlent les événements qui persistent et que vous pouvez interroger. Pour plus d'informations sur CloudTrail Lake, consultez la section [Travailler avec AWS CloudTrail Lake](#) dans le guide de AWS CloudTrail l'utilisateur.

CloudTrail Les stockages et requêtes de données sur les événements de Lake entraînent des coûts. Lorsque vous créez un magasin de données d'événement, vous choisissez l'[option de tarification](#) que vous voulez utiliser pour le magasin de données d'événement. L'option de tarification détermine le coût d'ingestion et de stockage des événements, ainsi que les périodes de conservation par défaut et maximale pour le magasin de données d'événement. Pour plus d'informations sur la CloudTrail tarification, consultez la section [AWS CloudTrail Tarification](#).

## AWS TNBexemples d'événements

Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'APIopération demandée, la date et l'heure de l'opération, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des API appels publics, de sorte que les événements n'apparaissent pas dans un ordre spécifique.

L'exemple suivant montre un CloudTrail événement illustrant l'CreateSolFunctionPackageopération.

```
{
  "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"
}

```

Pour plus d'informations sur le contenu des CloudTrail enregistrements, voir [le contenu des CloudTrail enregistrements](#) dans le Guide de AWS CloudTrail l'utilisateur.

## AWS TNBTâches de déploiement

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

Le tableau suivant répertorie les tâches AWS TNB de déploiement :

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 plugin Multus sur le cluster AmazonEKS.

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
AppUpdate	aucun changement de nom	Met à jour les fonctions réseau déjà installées dans une instance réseau.
-	ClusterPluginUninstall	Désinstalle les plugins sur le cluster AmazonEKS.
ClusterStorageClassesConfiguration	aucun changement de nom	Configure la classe de stockage (CSIpilote) sur un EKS cluster Amazon.
FunctionDeletion	aucun changement de nom	Supprime les fonctions réseau des AWS TNB ressources.
FunctionInstantiation	FunctionInstall	Déploie les fonctions réseau à l'aide deHELM.
FunctionUninstallation	FunctionUninstall	Désinstalle la fonction réseau d'un cluster AmazonEKS.
HookExecution	aucun changement de nom	Exécute les hooks du cycle de vie tels que définis dans leNSD.
InfrastructureCancellation	aucun changement de nom	Annule un service réseau.
InfrastructureInstantiation	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.
-	InfrastructureUpdate	Met à jour les AWS ressources mises en service pour le compte de l'utilisateur.
InventoryDeregistration	aucun changement de nom	Désenregistre les ressources de. AWS 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
-	InventoryRegistration	Enregistre les AWS ressources dans AWS TNB.
KubernetesClusterConfiguration	ClusterConfiguration	Configure le cluster Kubernetes et ajoute des rôles supplémentaires IAM 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.
NetworkServiceInstantiation	aucun changement de nom	Initialise le service réseau.
SelfManagedNodesConfiguration	aucun changement de nom	Bootstrap les nœuds autogérés avec le plan de contrôle Amazon EKS et Kubernetes.
-	ValidateNetworkServiceUpdate	Exécute les validations avant de mettre à jour une instance réseau.

## Quotas de service pour AWS TNB

Les quotas de service, également appelés limites, correspondent au nombre maximal 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.

Voici les quotas de service pour AWS TNB.

Nom	Par défaut	Ajuste	Description
Opérations de service réseau continues simultanées	Chaque Région prise en charge : 40	<a href="#">Oui</a>	Le nombre maximum d'opérations de service réseau en cours simultanées dans une région.
Packages de fonctions	Chaque région prise en charge : 200	<a href="#">Oui</a>	Le nombre maximum de packages de fonctions dans une région.
Packages réseau	Chaque Région prise en charge : 40	<a href="#">Oui</a>	Le nombre maximum de packages réseau dans une région.
Instances de service réseau	Chaque région prise en charge : 800	<a href="#">Oui</a>	Le nombre maximum d'instances de service réseau dans une région.

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

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

Modification	Description	Date
<a href="#">Version Kubernetes pour cluster</a>	AWS TNB prend désormais en charge la version 1.30 de Kubernetes pour créer des clusters Amazon. EKS	19 août 2024
<a href="#">AWS TNB prend en charge une opération supplémentaire pour gérer le cycle de vie du réseau.</a>	<p>Vous pouvez mettre à jour une instance réseau instanciée ou précédemment mise à jour avec un nouveau package réseau et de nouvelles valeurs de paramètres. Consultez :</p> <ul style="list-style-type: none"> <li>• <a href="#">Opérations liées au cycle</a></li> <li>• <a href="#">Mettre à jour une instance réseau</a></li> <li>• <a href="#">AWS TNB exemple de rôle de service</a> : <ul style="list-style-type: none"> <li>• Ajoutez les EKS actions Amazon suivantes : <code>eks:UpdateAddon</code> <code>eks:UpdateClusterVersion</code> <code>eks:UpdateNodegroupConfig</code> <code>eks:UpdateNodegroupVersion</code> , <code>eks:DescribeUpdate</code></li> </ul> </li> </ul>	30 juillet 2024

- Ajoutez cette AWS CloudFormation action : `cloudformation:UpdateStack`
- Nouvelles [tâches de déploiement](#) : `InfrastructureUpdate`, `InventoryRegistration`, `ValidateNetworkServiceUpdate`
- API mises à jour : [GetSolNetworkOperationListSolNetworkOperations](#), et [UpdateSolNetworkInstance](#)

### [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 la version 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 `AWS.Networking.ENI`.

2 avril 2024

<a href="#">Support pour le chiffrement du volume EBS racine Amazon</a>	Vous pouvez activer le EBS chiffrement Amazon pour le volume EBS racine Amazon. Pour l'activer, ajoutez les propriétés dans le <a href="#">AWSfichier .Compute. EKSMangedNode</a> ou <a href="#">AWS.Compute. EKSSelfManagedNode</a> œud.	2 avril 2024
<a href="#">Support pour le nœud labels</a>	Vous pouvez associer des étiquettes de nœud à votre groupe de nœuds dans le <a href="#">AWSfichier .Compute. EKSMangedNode</a> ou <a href="#">AWS.Compute. EKSSelfManagedNode</a> œud.	19 mars 2024
<a href="#">Support pour l'interface réseau source_dest_check</a>	Vous pouvez indiquer si vous souhaitez activer ou désactiver le contrôle source/destination de l'interface réseau via le AWS fichier .Networking. ENInœud.	25 janvier 2024
<a href="#">Support pour les EC2 instances Amazon avec des données utilisateur personnalisées</a>	Vous pouvez lancer des EC2 instances Amazon avec des données utilisateur personnalisées via le AWS fichier .Compute. UserData nœud.	16 janvier 2024
<a href="#">Support pour le groupe de sécurité</a>	AWS TNBvous permet d'importer la AWS ressource Security Group.	8 janvier 2024

<a href="#">Description mise à jour de <code>network_interfaces</code></a>	Lorsque la <code>network_interfaces</code> propriété est incluse dans le <a href="#">AWS fichier <code>.Compute.EKSManagedNode</code></a> ou <a href="#">AWS <code>.Compute.EKSSelfManagedNode</code></a> , AWS TNB obtient l'autorisation associée à ENIs partir de la <code>multus_role</code> propriété si elle est disponible, ou à partir de la <code>node_role</code> propriété.	18 décembre 2023
<a href="#">Support pour les clusters privés</a>	AWS TNB prend désormais en charge les clusters privés. Pour indiquer un cluster privé, définissez la <code>access</code> propriété sur <code>PRIVATE</code> .	11 décembre 2023
<a href="#">Version Kubernetes pour cluster</a>	AWS TNB prend désormais en charge la version 1.28 de Kubernetes pour créer des clusters Amazon. EKS	11 décembre 2023
<a href="#">AWS TNB soutient le groupe de placement</a>	Ajout d'un groupe de placement pour les définitions <a href="#">AWS <code>.Compute.EKSManagedNode</code></a> des <a href="#">AWS <code>.Compute.EKSSelfManagedNode</code></a> nœuds et.	11 décembre 2023



## [AWS TNBajoute le support pour IPv6](#)

AWS TNBprend désormais en charge la création d'instances réseau avec IPv6 infrastructure. Vérifiez les nœuds [AWS.Networking.VPC](#), [AWS.Réseau.Sous-réseau](#), [.Mise en réseau.AWS InternetGateway](#), [AWS.Réseau. utage.SecurityGroupIngressRule](#), [AWS.Réseautage.SecurityGroupEgressRule](#), et [AWS.Compute.EKS](#)pour les IPv6 configurations. Nous avons également ajouté les nœuds [AWS.Networking.NATGateway](#)et [AWS.Networking.Route](#) pour la configuration. NAT64 Nous avons mis à jour le AWS TNB rôle de AWS TNB service et le rôle de service du groupe de EKS nœuds Amazon pour IPv6 les autorisations. Consultez les [exemples de politiques relatives aux rôles de service](#).

16 novembre 2023

## [Autorisations ajoutées à la politique des rôles de AWS TNB service](#)

Nous avons ajouté des autorisations à la politique des rôles de AWS TNB service pour Amazon S3 et pour AWS CloudFormation permettre l'instanciation de l'infrastructure.

23 octobre 2023

<a href="#">AWS TNBlancé dans un plus grand nombre de régions</a>	AWS TNBest désormais disponible dans les régions Asie-Pacifique (Séoul), Canada (centre), Europe (Espagne), Europe (Stockholm) et Amérique du Sud (São Paulo).	27 septembre 2023
<a href="#">Balises pour AWS.Compute.EKSSelfManagedNode</a>	AWS TNBprend désormais en charge les balises pour la définition du AWS .Compute .EKSSelfManagedNode nœud.	22 août 2023
<a href="#">AWS TNBprend en charge les instances qui tirent parti IMDSv2</a>	Lorsque vous lancez votre instance, vous devez utiliser IMDSv2.	14 août 2023
<a href="#">Autorisations mises à jour pour MultusRoleInlinePolicy</a>	Cela inclut MultusRoleInlinePolicy désormais l'ec2:DeleteNetworkInterface autorisation.	7 août 2023
<a href="#">Version Kubernetes pour cluster</a>	AWS TNBprend désormais en charge les versions 1.27 de Kubernetes pour créer des clusters Amazon. EKS	25 juillet 2023
<a href="#">AWS.Calculez. EKS. AuthRole</a>	AWS TNBsupports AuthRole qui vous permettent d'ajouter IAM des rôles au EKS cluster Amazon aws-auth ConfigMap afin que les utilisateurs puissent accéder au EKS cluster Amazon à l'aide d'un IAM rôle.	19 juillet 2023

---

<a href="#">AWS TNBprend en charge les groupes de sécurité.</a>	Ajout du <a href="#">AWS.Networking.SecurityGroup</a> , <a href="#">AWS.Réseau. SecurityGroupEgressRule</a> , et <a href="#">AWS.Networking.SecurityGroupIngressRule</a> au NSD modèle.	18 juillet 2023
<a href="#">Version Kubernetes pour cluster</a>	AWS TNBprend en charge les versions 1.22 à 1.26 de Kubernetes pour créer des clusters Amazon. EKS AWS TNBne prend plus en charge les versions 1.21 de Kubernetes.	11 mai 2023
<a href="#">AWS.Calculez. EKSSelfManagedNode</a>	Vous pouvez créer des nœuds de travail autogérés dans la région, dans les Zones AWS Locales et. AWS Outposts	29 mars 2023
<a href="#">Première version</a>	Il s'agit de la première version du guide de AWS TNB l'utilisateur.	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.