使用者指南

AWS 電信網絡生成器

AWS 電信網絡生成器: 使用者指南

Table of Contents

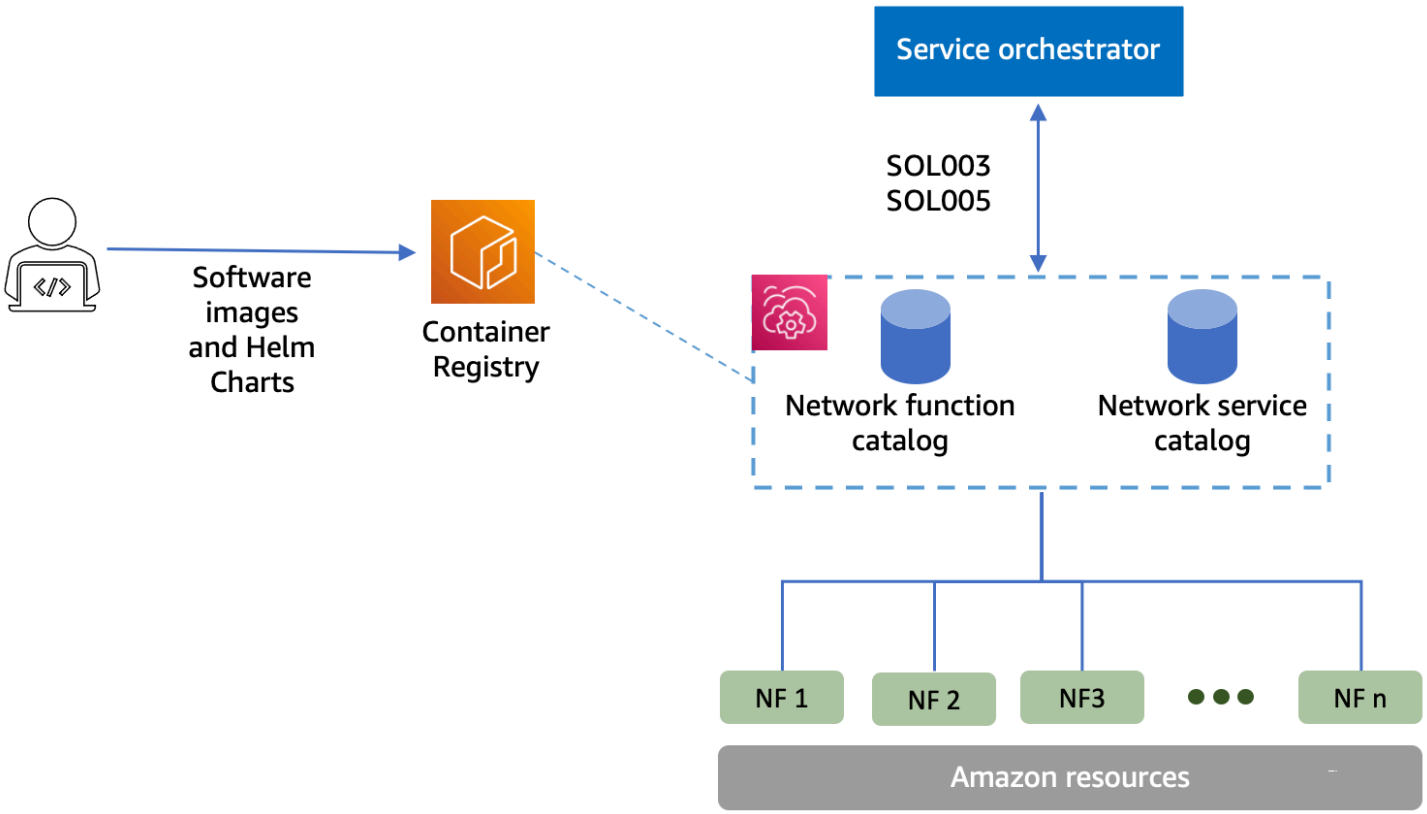
什麼是AWS電信網絡生成器？

AWS電信網絡生成器（AWSTNB）是一種AWS服務，可為通信服務提供商（CSP）提供有效的方式來部署，管理和擴展AWS基礎設施上的 5G 網絡。

使用AWS TNB，您可以自動化的方式在AWS 雲端使用容器化軟件映像中部署可擴展且安全的 5G 網絡。您不需要學習新技術、決定要使用哪種運算服務，也無需了解如何佈建和設定 AWS。

相反地，您可以描述網路的基礎架構，並提供來自獨立軟體廠商 (ISV) 合作夥伴的網路功能軟體影像。AWSTNB 與第三方服務協調器和服AWS務整合，可自動佈建必要的AWS基礎架構、部署容器化網路功能，並設定網路和存取管理，以建立完全可運作的網路服務。

下圖說明AWS TNB 與服務協調器之間的邏輯整合，以利用歐洲電信標準協會 (ETSI) 為基礎的標準介面部署網路功能。



主題

- AWS 新手？
- AWSTNB 適用於誰？
- 為什麼要使用AWS TNB？

- 存取AWS TNB
- AWSTNB 的定價
- 什是？

AWS 新手？

如果您剛接觸 AWS 的產品與服務，請利用下列資源開始深入了解：

- 簡介 AWS
- AWS 入門

AWSTNB 適用於誰？

AWSTNB 適用於希望利用產品的成本效益、敏捷性和彈性的 CSP，而無需撰寫和維護自訂指令碼和組態來設計、部署和管理網路服務。AWS 雲端 AWSTNB 會自動佈建必要的AWS基礎結構、部署容器化網路功能，並設定網路和存取管理，以根據 CSP 定義的網路服務描述元以及 CSP 想要部署的網路功能，建立完全可運作的網路服務。

為什麼要使用AWS TNB？

下列是 CSP 想要使用AWS TNB 的一些原因：

協助簡化工作

為您的網路作業提供更高的效率，例如部署新服務、更新和升級網路功能，以及變更網路基礎架構拓撲。

與協調器整合

AWSTNB 與符合 ETSI 標準的熱門協力廠商服務協調器整合。

秤

您可以設定AWS TNB 來擴充基礎AWS資源以滿足流量需求、更有效率地執行網路功能更新、推出網路基礎架構拓撲變更，並將新 5G 服務的部署時間從數天縮短到數小時。

檢查和監控AWS資源

AWSTNB 可讓您在單一儀表板上檢查和監控支援網路的AWS資源，例如亞馬遜 VPC、Amazon EC2 和亞馬遜 EKS。

支援服務範本

AWSTNB 可讓您為所有電信工作負載 (RAN、核心、IMS) 建立服務範本。您可以建立新的服務定義、重複使用現有範本，或與持續整合和持續傳遞 (CI/CD) 管線整合以發佈新定義。

追蹤網路部署的變更

當您變更網路函數部署的基礎組態時，例如變更 Amazon EC2 執行個體類型的執行個體類型，您可以透過可重複且可擴展的方式追蹤變更。手動執行此作業需要管理網路狀態、建立和刪除資源，以及注意所需變更的順序。當您使用AWS TNB 來管理網路功能的生命週期時，您只會對描述網路功能的網路服務描述元進行變更。 AWS然後，TNB 將自動以正確的順序進行所需的更改。

簡化網路功能生命週期

您可以管理網路功能的第一個和所有後續版本，並指定升級的時間。您也可以用相同的方式管理 RAN、核心、IMS 和網路應用程式。

存取AWS TNB

您可以使用下列任一界面來建立、存取和管理您的AWS TNB 資源：

- AWSTNB 主控台 — 提供用於管理網路的 Web 介面。
- AWSTNB API — 提供用於執行AWS TNB 操作的 REST 風格 API。如需詳細資訊，請參閱 AWSTNB API 參考資料
- AWS Command Line Interface(AWS CLI) — 為包括AWS TNB 在內的廣泛AWS服務集提供命令。Windows、macOS 和 Linux 都支援。如需詳細資訊，請參閱AWS Command Line Interface。
- AWSSDK-提供語言特定 API，並完成許多連線詳細資訊。包括計算簽章、處理請求重試和錯誤處理。如需詳細資訊，請參閱 AWS 開發套件。

AWSTNB 的定價

AWSTNB 可協助通訊服務供應商自動化其電信網路的部署和管理AWS。使用AWS TNB 時，您需要支付下列兩個維度的費用：

- 通過託管網絡功能項（MNFI）小時。
- 按 API 請求的數量。

與AWS TNB 一起使用其他AWS服務時，也會產生額外費用。如需詳細資訊，請參閱 AWSTNB 定價。

若要檢視您的帳單，請前往 AWS Billing and Cost Management 主控台中的帳單與成本管理儀表板。您的帳單內含用量報告的連結，可提供帳單的其他詳細資訊。如需AWS帳戶帳單的詳細資訊，請參閱AWS帳戶帳單。

如果您有關於 AWS 帳單、帳戶和事件的任何問題，請聯絡 AWS Support。

AWS Trusted Advisor是一種可協助您將AWS環境的成本、安全性與效能最佳化的服務。如需詳細資訊，請參閱 AWSTrusted Advisor。

什是？

如需如何開始使用AWS TNB 的詳細資訊，請參閱下列主題：

- 設定 AWS TNB— 完成先決條件步驟。
- 開始使用 AWS TNB— 部署您的第一個網絡功能，例如集中式單元（CU），訪問和移動管理功能（AMF），用戶平面功能（UPF）或完整的 5G 核心。

AWSTNB 的工作原理

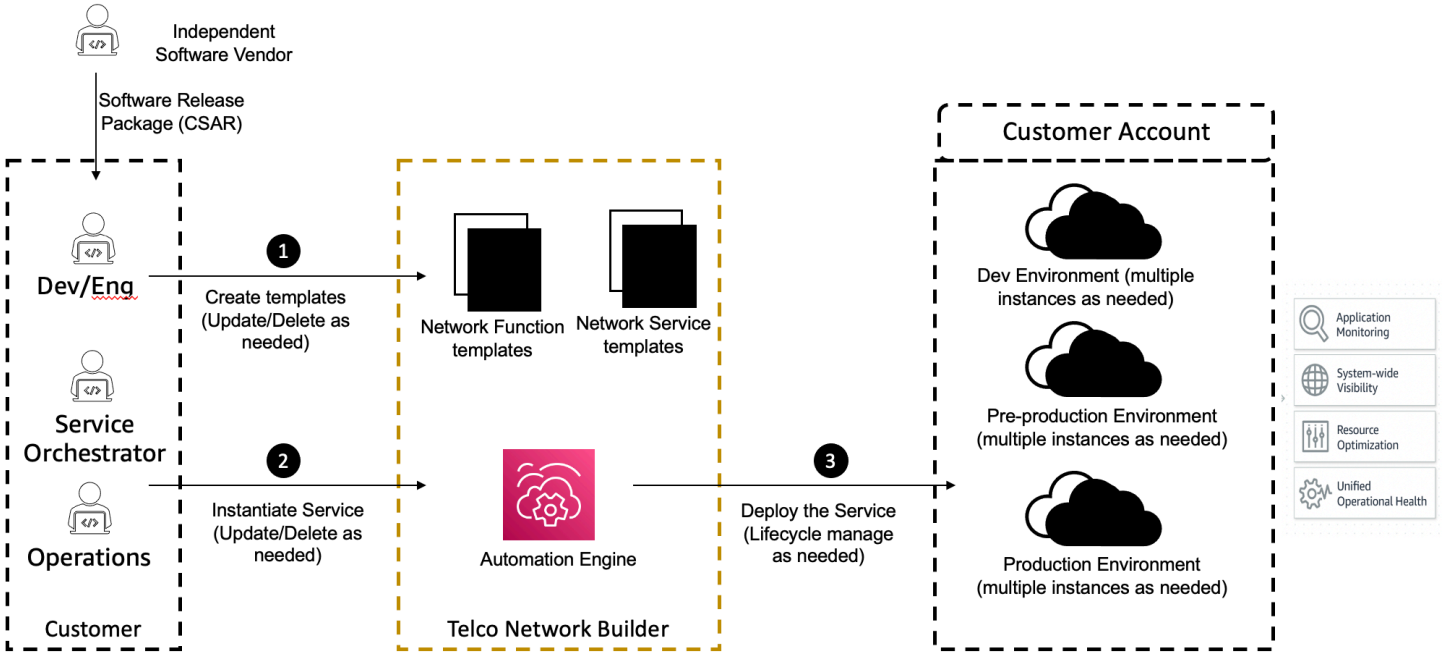
AWSTNB 與標準化 end-to-end 協調器和AWS資源整合，以運作完整的 5G 網路。

AWSTNB 可讓您擷取網路功能套件和網路服務描述元 (NSDs)，並提供自動化引擎來操作您的網路。您可以使用 end-to-end 協調器並與AWS TNB API 整合，或使用AWS TNB SDK 建立自己的自動化流程。如需更多詳細資訊，請參閱 AWSTNB 架構。

主題

AWSTNB 架構

AWSTNB 讓您能夠透過AWS Management Console、AWS CLI、AWS TNB REST API 和 SDK 執行生命週期管理作業。這可讓不同的 CSP 角色 (例如工程、作業和程式化系統小組的成員) 利用AWS TNB。您可以建立並上傳網路功能套件做為雲端服務封存 (CSAR) 檔案。該 CSAR 文件包含頭盔圖表，軟件映像和網絡功能描述符（NFD）。您可以使用範本重複部署該套件的多個設定。您可以建立定義您要部署的基礎結構和網路功能的網路服務範本。您可以使用參數覆寫，在不同的位置部署不同的組態。然後，您可以使用範本實例化網路，並在AWS基礎架構上部署網路功能。 AWSTNB 為您提供部署的可見性。



與 AWS 服務 的整合

5G 網路由一組互連的容器化網路功能組成，部署在數千個 Kubernetes 叢集中。 AWSTNB 與以下項目整合AWS 服務為電信專用 API，以建立完全可運作的網路服務：

- Amazon Elastic Container Registry (Amazon ECR) 可儲存獨立軟體廠商 (ISV) 網路功能的網路功能。
- Amazon Elastic Kubernetes Service（亞馬遜 EKS）來設置集群。
- 適用於聯網建構的亞馬遜 VPC。
- 使用AWS CloudFormation.
- AWS CodePipeline適用於跨區域AWS 區域、AWS Local Zones 和AWS Outposts.
- IAM 來定義角色。
- AWS Organizations以控制對AWS TNB API 的存取。
- AWS Health Dashboard並AWS CloudTrail監控健康狀況和發布指標。

AWSTNB 資源配額

您的AWS 帳戶有預設配額，先前稱為限額，每個配額AWS 服務。除非另有說明，否則每個配額都是AWS 區域. 您可以要求提高某些配額，但不是所有配額。

若要檢視AWS TNB 的配額，請開啟 Service Quotas 主控台。在導覽窗格中，選擇 AWS 服務，然後選取 AWSTNB。

若要請求提高配額，請參閱《Service Quotas 使用者指南》https://docs.aws.amazon.com/servicequotas/latest/userguide/request-quota-increase.html中的請求提高配額。

您的 TNBAWS 帳戶 具有下列與AWS TNB 相關的配額。

資源配額	描述	預設值	是否可調整？
網路服務實例	一個區域內的網路服務執行個體數數服務執行個體數	800	是
同時進行的網路服務作業	在一個區域中，同時進行的網路服務數服務數服務數作業數數	40	是

資源配額	描述	預設值	是否可調整？
網路套件	一個區域中限制網路套件數數數數套件數數數數	40	是
功能套件	一個區域中限制功能套件數數數數套件數數數數	200	是

AWS TNB 概念

本主題說明協助您開始使用 AWS TNB 的基本概念。

目錄

網路功能的生命週期

AWS TNB 可以幫助您在整個網絡功能的生命週期。網絡功能生命週期包括以下階段和活動：

規劃

1. 透過識別要部署的網路功能來規劃您的網路。
2. 將網絡功能軟件映像放入容器映像存儲庫中。
3. 建立要部署或升級的 CSAR 套件。
4. 使用 AWS TNB 上傳定義網路功能 (例如 CU AMF 和 UPF) 的 CSAR 套件，並與持續整合和持續傳遞 (CI/CD) 管線整合，以協助您在新的網路功能軟體影像或客戶指令碼時建立 CSAR 套件的新版本。

組態

1. 識別部署所需的資訊，例如運算類型、網路功能版本、IP 資訊和資源名稱。
2. 使用這些資訊來建立您的網路服務描述元 (NSD)。
3. 內嵌定義網路功能的 NSD，以及網路功能實例化所需的資源。

實例化

1. 建立網路功能所需的基礎架構。
2. 實例化（或提供）其 NSD 中定義的網絡功能，並開始承載流量。
3. 驗證資產。

生產

在網路功能的生命週期中，您將完成生產作業，例如：

- 更新網路功能組態，例如，更新已部署網路功能中的值。
- 更換或解除使用網路功能。

使用標準化介面

AWS TNB 與歐洲電信標準協會 (ETSI) 相容的服務協調器整合，可讓您簡化網路服務的部署。服務協調器可以使用 AWS TNB SDK、CLI 或 API 來啟動作業，例如實體化或升級網路功能至新版本。

AWS TNB 支持以下規格。

規格	發行版本	描述
等等	v3.6.1	定義允許以 TOSCAA 為基礎的網路函數描述元的標準。
等	v3.6.1	定義網路功能管理的模型。
等	v3.6.1	定義網路功能生命週期管理的標準。
等	v3.6.1	定義網路函數套件的 CSAR 標準。
等	v3.6.1	定義網路服務套件和網路服務生命週期管理的標準。
等	v3.5.1	定義允許以 TOSCAA 為基礎之網路服務描述元的標準。

適用於 AWS TNB 的網路功能套件

使用 AWS TNB，您可以將符合 ETSI SOL001/SOL004 的網路功能套件儲存到功能目錄中。然後，您可以上傳包含描述您網路功能之成品的雲端服務封存 (CSAR) 套件。

- 網路功能描述元 — 定義套件上架和網路功能管理的中繼資料
- 軟件映像 — 參考網絡功能容器映像。Amazon Elastic Container Registry (Amazon ECR) 可以充當您的網絡功能映像存儲庫。

- 其他檔案 — 用於管理網路功能；例如指令碼和 Helm 圖表。

CSAR 是由 OASIS TOSCA 標準定義的套件，其中包含符合 OASIS TOSCA YAML 規格的網路/服務描述元。如需所需 YAML 規格的相關資訊，請參閱AWSTNB 的托斯卡參考。

以下是網路函數描述元的範例。

```
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"
```

TNB 的 AWS 網路功能服務描述元

AWS TNB 會儲存您想要部署的網路功能，以及如何將它們部署到目錄的網路服務描述元 (NSDs)。您可以上傳 YAML NSD 檔案，如 ETSI SOL007 所述，以包含下列項目：

- 您要部署的 NF
- 網路指示
- 運算指示
- 生命週期掛鉤 (自訂指令碼)

AWS TNB 支援 ETSI 標準，用於 TOSCA 語言中的資源建模，例如網路、服務和功能。 AWS TNB 讓您以符合 ETSI 規範的服務協調器可以 AWS 服務 理解的方式建模，讓您更有效率地使用它們。

以下是 NSD 的片段，展示如何建模 AWS 服務。網路功能將部署在具有 1.27 版本的 Amazon EKS 叢集上。應用程式的子網路是子網路 01 和子網路 02。然後，您可以使用 Amazon 機器映像 (AMI)、執行個體類型和自動調度資源組態 NodeGroups 為應用程式定義。

```
tosca_definitions_version: tnb_simple_yaml_1_0

SampleNFEKS:
  type: tosca.nodes.AWS.Compute.EKS
  properties:
    version: "1.27"
    access: "ALL"
    cluster_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleClusterRole"
  capabilities:
    multus:
      properties:
        enabled: true
  requirements:
    subnets:
      - Subnet01
      - Subnet02

SampleNFEKSNode01:
  type: tosca.nodes.AWS.Compute.EKSManagedNode
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleNodeRole"
  capabilities:
    compute:
      properties:
        ami_type: "AL2_x86_64"
        instance_types:
          - "t3.xlarge"
        key_pair: "SampleKeyPair"
    scaling:
      properties:
        desired_size: 3
        min_size: 2
        max_size: 6
  requirements:
    cluster: SampleNFEKS
    subnets:
```

```
      - Subnet01
    network_interfaces:
      - ENI01
      - ENI02
```

AWS TNB 的管理及營運

透過 AWS TNB，您可以使用符合 ETSI SOL003 和 SOL005 的標準化管理作業來管理您的網路。您可以使用 AWS TNB API 來執行生命週期作業，例如：

- 實例化您的網絡功能。
- 終止您的網路功能。
- 更新您的網路功能以覆寫 Helm 部署。
- 管理網路功能套件的版本。
- 管理您的 NSD 版本。
- 擷取已部署網路功能的相關資訊。

TNB 的 AWS 網路服務描述元

網路服務描述元 (NSD) 是網路套.yaml件中的檔案，它使用 TOSCA 標準來描述您要部署的網路功能，以及您要部署網路功能的 AWS 基礎結構。若要定義您的 NSD 並設定您的基礎資源和網路生命週期作業，您必須瞭解 TNB 支援的 NSD TOSCA 結構描述。 AWS

您的 NSD 文件分為以下幾部分：

1. TOSCA 定義版本 — 這是 NSD YAML 檔案的第一行，其中包含版本資訊，如下列範例所示。

```
tosca_definitions_version: tnb_simple_yaml_1_0
```

2. VNFD — NSD 包含要在其上執行生命週期作業之網路功能的定義。每個網路功能必須由下列值來識別：
   - 的唯一識別碼descriptor_id。識別碼必須與網路函數 CSAR 套件中的識別碼相符。
   - 的唯一名稱namespace。名稱必須與唯一 ID 相關聯，才能更輕鬆地在整個 NSD YAML 檔案中參照，如下列範例所示。

```
vnfds:
```

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

3. 拓撲範本 — 定義要部署的資源、網路功能部署，以及任何自訂指令碼，例如生命週期掛接。如以下範例所示。

```
topology_template:

  node_templates:

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

4. 其他節點 — 每個已建模的資源都有屬性和需求的區段。屬性描述資源的選擇性或必要屬性，例如版本。需求描述了必須作為參數提供的依賴關係。例如，若要建立 Amazon EKS 節點群組資源，必須在 Amazon EKS 叢集中建立該資源。如以下範例所示。

```
SampleEKSNode:
  type: tosca.nodes.AWS.Compute.EKSManagedNode
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
  capabilities:
    compute:
      properties:
        ami_type: "AL2_x86_64"
        instance_types:
          - "t3.xlarge"
        key_pair: "SampleKeyPair"
    scaling:
      properties:
        desired_size: 1
        min_size: 1
        max_size: 1
  requirements:
    cluster: SampleEKS
    subnets:
      - SampleSubnet
    network_interfaces:
      - SampleENI01
```

```
      - SampleENI02
```

設定 AWS TNB

透過完成本主題中描述的工作來設定 AWS TNB。

任務

- 註冊成為 AWS
- 選擇一個 AWS 地區
- 注意服務端點
- (選擇性) 安裝 AWS CLI
- 建立 IAM 使用者
- 設定 AWS TNB 角色

註冊成為 AWS

當您註冊 Amazon Web Services 時，系統會自動註冊您 AWS 帳戶 的所有服務 AWS，包括 AWS TNB。您只需支付實際使用服務的費用。

如果您已 AWS 帳戶 經擁有，請跳到下一個任務。如果您還沒有 AWS 帳戶，請使用下列程序建立新帳戶。

若要建立 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

   當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

選擇一個 AWS 地區

若要檢視 AWS TNB 的可用區域清單，請參閱區AWS 域服務清單。若要檢視用於程式設計方式存取的端點清單，請參閱AWS . AWS 一般參考

注意服務端點

若要以程式設計方式連線到 AWS 服務，請使用端點。除了標準 AWS 端點之外，某些 AWS 服務還在選取的區域提供 FIPS 端點。如需詳細資訊，請參閱 AWS 服務端點。

區域名稱	區域	端點	通訊協定	
美國東部 (維吉尼亞北部)	us-east-1	tnb.us-east-1.amazonaws.com	HTTPS	
美國西部 (奧勒岡)	us-west-2	tnb.us-west-2.amazonaws.com	HTTPS	
亞太區域 (首爾)	ap-northeast-2	tnb.ap-northeast-2.amazonaws.com	HTTPS	
亞太區域 (悉尼)	ap-southeast-2	tnb.ap-southeast-2.amazonaws.com	HTTPS	
加拿大 (中部)	ca-central-1	tnb.ca-central-1.amazonaws.com	HTTPS	
歐洲 (法蘭克福)	eu-central-1	tnb.eu-central-1.amazonaws.com	HTTPS	
歐洲 (巴黎)	eu-west-3	tnb.eu-west-3.amazonaws.com	HTTPS	
歐洲 (西班牙)	eu-south-2	tnb.eu-south-2.amazonaws.com	HTTPS	
歐洲 (斯德哥爾摩)	eu-north-1	tnb.eu-north-1.amazonaws.com	HTTPS	
南美洲 (聖保羅)	sa-east-1	tnb.sa-east-1.amazonaws.com	HTTPS	

(選擇性) 安裝 AWS CLI

AWS Command Line Interface (AWS CLI) 提供多種 AWS 產品的指令，並且在視窗、macOS 和 Linux 上都支援這些指令。您可以使用存取 AWS TNB。 AWS CLI若要開始使用，請參閱《 使用者指南AWS Command Line Interface》。若要取得有關 AWS TNB 指令的更多資訊，請參閱《指AWS CLI 令參考》中的 tnb。

建立 IAM 使用者

AWS Identity and Access Management (IAM) 是可協助您安全地控制 AWS 資源存取的 Web 服務。建立 IAM 使用者角色以使用短期登入資料進行存取 AWS。

要創建角色，請按照「AWS IAM Identity Center 用戶指南」中的「入門」中的說明進行操作。

您也可以 AWS IAM Identity Center在《使用AWS Command Line Interface 者指南》中將設定 AWS CLI 為使用，以設定程式設計方式存取。

設定 AWS TNB 角色

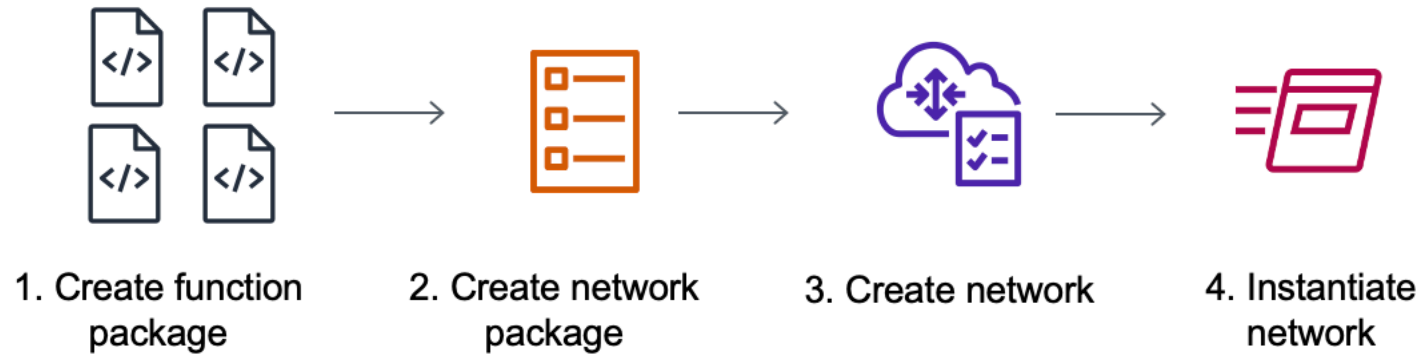
您必須建立 IAM 服務角色來管理 AWS TNB 解決方案的不同部分。 AWS TNB 服務角色可以代表您對其他 AWS 服務 (例如 AWS CloudFormation AWS CodeBuild、以及各種計算和儲存服務) 進行 API 呼叫，以實體化和管理部署的資源。

如需 AWS TNB 服務角色的詳細資訊，請參閱AWS TNB 的身分識別與存取管理。

開始使用 AWS TNB

本教學課程示範如何使用 AWS TNB 部署網路功能，例如集中式單元 (CU)、存取和行動管理功能 (AMF) 或 5G 使用者平面功能 (UPF)。

下圖說明部署程序：



任務

- 必要條件
- 建立函數套件
- 建立網路套件
- 建立並實體化網路執行個體
- 清除

必要條件

您必須具備下列項目，才能執行成功部署：

- AWS 業務 Support 計劃。
- 透過 IAM 角色提供的許可。
- 符合 ETSI SOL001 的網路功能 (NF) 封裝。
- 符合 ETSI SOL007 規範的網路服務描述元 (NSD) 範本。

您可以從 AWS TNB 網 GitHub 站的示例軟件包中使用示例函數包或網絡包。

建立函數套件

若要建立函數套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇 [函式套件]。
3. 選擇 [建立功能套件]。
4. 在 [上傳功能套件] 下，選擇 [選擇檔案]，然後將 CSAR 套件上傳為.zip檔案。
5. (選擇性) 在「標籤」下，選擇「新增標籤」，然後輸入金鑰和值。您可以使用標籤來搜尋和篩選資源或追蹤 AWS 成本。
6. 選擇下一步。
7. 檢閱套件詳細資料，然後選擇 [建立功能套件]。

建立網路套件

建立網路套件

1. 在瀏覽窗格中，選擇 [網路套件]。
2. 選擇「建立網路套件」。
3. 在 [上傳網路套件] 下，選擇 [選擇檔案]，然後將您的 NSD 上傳為.zip檔案。
4. (選擇性) 在「標籤」下，選擇「新增標籤」，然後輸入金鑰和值。您可以使用標籤來搜尋和篩選資源或追蹤 AWS 成本。
5. 選擇下一步。
6. 選擇「建立網路套件」。

建立並實體化網路執行個體

若要建立和具現化網路執行個體

1. 在導覽窗格中，選擇 [網路]。
2. 選擇建立網路執行個體。
3. 輸入網路的名稱和描述，然後選擇 [下一步]。
4. 選擇您的 NSD。確認詳細資料，然後選擇 [下一步]。

5. 選擇建立網路執行個體。初始狀態為Created。
6. 選擇網路執行個體的 ID，然後選擇「實例化」。
7. 選擇「實例化網絡」。
8. 使用「重新整理」圖示追蹤網路執行個體的狀態。

清除

清除您的資源

1. 在導覽窗格中，選擇 [網路]。
2. 選擇網路的 ID，然後選擇 [終止]。
3. 出現確認提示時，請輸入網路 ID，然後選擇 [終止]。
4. 使用「重新整理」圖示追蹤網路執行個體的狀態。
5. (選擇性) 選取網路，然後選擇「刪除」。

適用於 AWS TNB 的功能套件

函數套件是 CSAR (雲端服務封存) 格式的 .zip 檔案，其中包含網路功能 (ETSI 標準電信應用程式) 和使用 TOSCA 標準來描述網路功能應如何在您的網路上執行的函式封裝描述器。

任務

- 在 AWS TNB 中創建一個函數包
- 在 AWS TNB 中查看功能包
- 從 TNB 下載功能AWS包
- 從 AWS TNB 刪除函數包

在 AWS TNB 中創建一個函數包

瞭解如何在 AWS TNB 網路功能目錄中建立函數套件。創建函數包是在 TNB 中創建網絡的第一步。一旦你上傳了一個功能包，你需要創建一個網絡包。

Console

若要使用主控台建立函數套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇 [函式套件]。
3. 選擇 [建立功能套件]。
4. 選擇選擇文件並上傳您的 NF 的 CSAR 軟件包。
5. 選擇下一步。
6. 檢閱封裝詳細資料。
7. 選擇 [建立功能套件]。

AWS CLI

若要使用建立函數套件 AWS CLI

1. 使用指create-sol-function-package令建立新的函式套件：

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

2. 使用 put-sol-function-package-content 指令來上傳函數套件內容。例如：

```
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
```

在 AWS TNB 中查看功能包

瞭解如何檢視函式套件的內容。

Console

若要使用主控台檢視函式套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇 [函式套件]。
3. 使用搜索框查找功能包

AWS CLI

若要使用檢視函式套件 AWS CLI

1. 使用指list-sol-function-packages令列出您的函數套件。

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

2. 使用指get-sol-function-package令可檢視有關函數套件的詳細資料。

```
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
```

從 TNB 下載功能AWS包

瞭解如何從 AWS TNB 網路功能目錄下載功能套件。

Console

使用控制台下載功能套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在主控台左側的瀏覽窗格中，選擇 [功能套件]。
3. 使用搜索框查找功能包
4. 選擇功能套件
5. 選擇 [動作]、[下載]

AWS CLI

若要使用下載函數套件 AWS CLI

使用 get-sol-function-package-content 指令下載函數套件。

```
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
```

從 AWS TNB 刪除函數包

瞭解如何從 AWS TNB 網路功能目錄刪除功能套件。若要刪除函式套件，套件必須處於停用狀態。

Console

若要使用主控台刪除函數套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇 [函式套件]。
3. 使用搜尋方塊尋找功能套件。

4. 選擇功能套件。
5. 選擇 Actions (動作)、Disable (停用)。
6. 選擇 動作 、刪除 。

AWS CLI

若要使用刪除函數套件 AWS CLI

1. 使用指update-sol-function-package令來停用功能套件。

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

2. 使用delete-sol-function-package指令刪除函數套件。

```
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
```

適用於 AWS TNB 的網路套件

網路套件是 CSAR (雲端服務封存) 格式的 .zip 檔案，會定義您要部署的函式套件，以及您要部署它們的AWS基礎結構。

任務

- 在 AWS TNB 中創建一個網絡包
- 在 AWS TNB 中查看網絡包
- 從 AWS TNB 下載網路套件
- 從 AWS TNB 刪除網路套件

在 AWS TNB 中創建一個網絡包

網路套件包含網路服務描述元 (NSD) 檔案 (必要) 和任何其他檔案 (選用)，例如特定於您需求的指令碼。例如，如果您的網路套件中有多個函數套件，您可以使用 NSD 定義應在特定 VPC、子網路或 Amazon EKS 叢集中執行的網路函數。

建立函數套件後建立網路套件。建立網路套件之後，您需要建立網路執行個體。

Console

使用控制台建立網路套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在瀏覽窗格中，選擇 [網路套件]。
3. 選擇「建立網路套件」。
4. 選擇「選擇檔案」 並上傳您的 CSAR 套件。
5. 選擇下一步。
6. 檢閱封裝詳細資料。
7. 選擇「建立網路套件」。

AWS CLI

使用建立網路套件 AWS CLI

1. 使用create-sol-network-package指令建立網路套件。

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

2. 使用 put-sol-network-package-content 指令來上傳網路套件內容。例如：

```
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
```

在 AWS TNB 中查看網絡包

瞭解如何檢視網路套件的內容。

Console

使用主控台檢視網路套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在瀏覽窗格中，選擇 [網路套件]。
3. 使用搜尋方塊來尋找網路套件。

AWS CLI

使用檢視網路套件 AWS CLI

1. 使用指list-sol-network-packages令列出您的網路套件。

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

2. 使用此命get-sol-network-package令可檢視有關網路套件的詳細資料。

```
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
```

從 AWS TNB 下載網路套件

瞭解如何從 AWS TNB 網路服務目錄下載網路套件。

Console

使用主控台下載網路套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在瀏覽窗格中，選擇 [網路套件]。
3. 使用搜尋方塊尋找網路套件
4. 選擇網路套件。
5. 選擇「動作」、「下載」

AWS CLI

使用下載網路套件 AWS CLI

- 使用 get-sol-network-package-content 指令下載網路套件。

```
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
```

從 AWS TNB 刪除網路套件

瞭解如何從 AWS TNB 網路服務目錄刪除網路套件。若要刪除網路套件，套件必須處於停用狀態。

Console

使用主控台刪除網路套件

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在瀏覽窗格中，選擇 [網路套件]。
3. 使用搜尋方塊尋找網路套件

4. 選擇網絡包
5. 選擇 Actions (動作)、Disable (停用)。
6. 選擇 動作 、刪除 。

AWS CLI

使用刪除網路套件 AWS CLI

1. 使用update-sol-network-package指令停用網路套件。

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

2. 使用delete-sol-network-package指令刪除網路套件。

```
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
```

AWSTNB 的網路執行個體

網路執行個體是在 AWS TNB 中建立的可部署的單一網路。

任務

- 使AWS用 TNB 實例化網絡實例
- 檢視 AWS TNB 中的網路執行個體
- 更新 TNB 中的網絡實AWS例
- 終止並從 AWS TNB 刪除網絡實例

使AWS用 TNB 實例化網絡實例

您可以在建立網路套件之後建立網路執行個體。建立網路執行個體之後，您必須將其具現化。當您具現化網路執行個體時，AWSTNB 會根據網路服務描述元中的規格部署網路功能。

Console

若要使用主控台建立和具現化網路執行個體

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇 [網路]。
3. 選擇建立網路執行個體。
4. 輸入執行個體的名稱和說明，然後選擇 [下一步]。
5. 選擇您的 NSD。驗證詳細資料，然後選擇 [下一步]。
6. 選擇建立網路執行個體。
7. 選擇「實例化」。
8. 選擇「實例化網絡」。
9. 重新整理以追蹤網路執行個體的狀態。

AWS CLI

若要使用建立和例證化網路執行個體 AWS CLI

1. 使用create-sol-network-instance指令建立網路例證。

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

2. 使用指instantiate-sol-network-instance令來實例化網路例證。

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

檢視 AWS TNB 中的網路執行個體

瞭解如何檢視網路執行個體。

Console

使用主控台檢視網路執行個體

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在瀏覽窗格中，選擇 [網路執行個體]。
3. 使用搜尋方塊尋找網路執行個體。

AWS CLI

使用檢視網路執行個體 AWS CLI

1. 使用指list-sol-network-instances令列出您的網路執行個體。

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

2. 使用指get-sol-network-instance令檢視有關網路執行個體的詳細資料。

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

更新 TNB 中的網絡實AWS例

瞭解如何更新網路執行個體。

Console

使用主控台更新網路執行個體

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇 [網路]。
3. 選取網路執行個體的 ID。
4. 在 [函數] 索引標籤上，選取要更新的函數執行個體。
5. 選擇 Update (更新)。
6. 輸入更新覆寫以確認更新。
7. 選擇 Update (更新)。
8. 重新整理以追蹤網路執行個體的狀態。

AWS CLI

使用 CLI 更新網路執行個體

使用update-sol-network-instance指令更新網路例證。

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

終止並從 AWS TNB 刪除網絡實例

若要刪除網路執行個體，執行個體必須處於終止狀態。

Console

使用主控台終止和刪除網路執行個體

1. 在以下位置開啟 AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇 [網路]。
3. 選取網路執行個體的 ID。
4. 選擇 Terminate (終止)。
5. 當系統提示您確認時，請輸入 ID 並選擇「終止」。
6. 重新整理以追蹤網路執行個體的狀態。

7. (選擇性) 選取網路執行個體，然後選擇刪除。

AWS CLI

使用終止和刪除網路執行個體 AWS CLI

1. 使用指terminate-sol-network-instance令終止網路執行個體。

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

2. (選擇性) 使用delete-sol-network-instance指令刪除網路例證。

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

適用於AWS TNB 的網路作業

網絡操作是對您的網絡進行的任何操作，例如網絡實例實例化或終止。

任務

- 檢視網路
- 取消網路操作

檢視網路

檢視網路作業的詳細資訊，包括網路作業所涉及的作業和作業狀態。

Console

使用主控台檢視網路

1. 請在以下位置開啟AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇網路主控台。
3. 使用搜尋方塊尋找網路執行個體。
4. 在「部署」索引標籤上，選擇「網路作業」。

AWS CLI

若要使用檢視網路作業AWS CLI

1. 使用指list-sol-network-operations令列出所有網路作業。

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

2. 使用指get-sol-network-operation令可檢視有關網路作業的詳細資料。

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

取消網路操作

了解如何取消網路操作。

Console

使用主控台取消網路操作

1. 請在以下位置開啟AWS TNB 主控台。 https://console.aws.amazon.com/tnb/
2. 在導覽窗格中，選擇網路
3. 選取網路介面 ID，開啟其詳細資訊頁面。
4. 在「部署」索引標籤上，選擇「網路作業」。
5. 選擇「取消作業」。

AWS CLI

若要使用取消網路作業AWS CLI

使用cancel-sol-network-operation指令取消網路作業。

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

AWSTNB 的托斯卡參考

雲端應用程式的拓撲和協調流程規格 (TOSCA) 是一種宣告式語法，CSP 用來描述雲端 Web 服務的拓撲、其元件、關係以及管理這些服務的程序。CSP 描述連線點、連線點之間的邏輯連結，以及 TOSCA 範本中的相似性和安全性等原則。CSP 然後將範本上傳到AWS TNB，該 TNB 會合成跨AWS可用區域建立正常運作的 5G 網路所需的資源。

內容

VNFD 模板

定義虛擬網路函數描述元 (VNFD) 範本。

語法

```
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
```

拓撲範本

`node_templates`

托斯卡AWS節點。可能的節點是：

- AWS.VNF

- AWS. 人工. 頭盔

AWS.VNF

定義一個AWS虛擬網路功能 (VNF) 節點。

語法

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

屬性

`descriptor_id`

描述元的 UUID。

必要：是

類型：String

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

`descriptor_version`

VNFD 的版本。

必要：是

類型：String

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

`descriptor_name`

描述元的名稱。

必要：是

類型：String

`provider`

VNFD 的作者。

必要：是

類型：String

請求

`helm`

Helm 目錄定義容器工件。這是一個參考AWS. 人工. 頭盔。

必要：是

類型：String

範例

```
SampleVNF:
  type: tosca.nodes.AWS.VNF
  properties:
    descriptor_id: "6a792e0c-be2a-45fa-989e-5f89d94ca898"
    descriptor_version: "1.0.0"
    descriptor_name: "Test VNF Template"
    provider: "Operator"
  requirements:
    helm: SampleHelm
```

AWS.Artifacts.Helm

定義一個AWS頭盔節點。

語法

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

屬性

`implementation`

CSAR 套件中包含頭盔圖表的本機目錄。

必要：是

類型：String

範例

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

NSD 模板

定義網路服務描述元 (NSD) 範本。

語法

```
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
```

使用定義的參數

當您想要動態傳遞參數 (例如 VPC 節點的 CIDR 區塊) 時，可以使用{ get_input: *input-parameter-name* }語法並在 NSD 範本中定義參數。然後在相同的 NSD 範本中重複使用該參數。

下面的例子演示了如何定義和使用參數：

```
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 }
```

越南變頻器導入

`descriptor_id`

描述元的 UUID。

必要：是

類型：String

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

`namespace`

唯一的名稱。

必要：是

類型：String

拓撲範本

`node_templates`

可能的托斯卡 AWS 節點是：

- AWS. NS
- AWS. 計算機.
- AWS. 計算機. AuthRole
- AWS. 計算機. ManagedNode
- AWS. 計算機. SelfManagedNode
- AWS計算。 PlacementGroup
- AWS計算。 UserData
- AWS. 網路。 SecurityGroup
- AWS. 網路。 SecurityGroupEgressRule
- AWS. 網路。 SecurityGroupIngressRule
- AWS. 資源. 匯入
- AWS. 網絡.
- AWS.HookExecution
- AWS. 網路。 InternetGateway
- AWS. 網路。 RouteTable
- AWS. 網路. 子網路
- AWS. 部署. 虛擬部署
- AWS. 網路. VPC
- AWS. 網路.
- AWS. 網路. 路線

AWS. NS

定義 AWS 網路服務 (NS) 節點。

語法

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

屬性

`descriptor_id`

描述元的 UUID。

必要：是

類型：String

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

`descriptor_version`

NSD 的版本。

必要：是

類型：String

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

`descriptor_name`

描述元的名稱。

必要：是

類型：String

範例

```
SampleNS:
  type: tosca.nodes.AWS.NS
  properties:
    descriptor_id: "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

```
    descriptor_version: "1.0.0"
    descriptor_name: "Test NS Template"
```

AWS. 計算機.

提供叢集的名稱、所需的 Kubernetes 版本，以及允許 Kubernetes 控制平面管理 NF 所需資源的角色。 AWS 多容器網絡接口（CNI）插件已啟用。您可以附加多個網路介面，並將進階網路設定套用至 Kubernetes 型網路功能。您也可以指定叢集端點存取和叢集的子網路。

語法

```
tosca.nodes.AWS.Compute.EKS:
  capabilities:
    multus:
      properties:
        enabled: Boolean
        multus_role: String
    ebs_csi:
      properties:
        enabled: Boolean
        version: String
  properties:
    version: String
    access: String
    cluster_role: String
    tags: List
    ip_family: String
  requirements:
    subnets: List
```

功能

`multus`

選用。定義多容器網路介面 (CNI) 用法的屬性。

如果包括`multus`，請指定`enabled`和`multus_role`性質。

`enabled`

指出是否啟用預設多項功能。

必要：是

類型：布林值

`multus_role`

Multus 網路介面管理的角色。

必要：是

類型：String

ebs_csi

定義 Amazon EKS 叢集中安裝之 Amazon EBS 容器儲存界面 (CSI) 驅動程式的屬性。

啟用此外掛程式以在 AWS Outposts、L AWS ocal Zones 或上使用 Amazon EKS 自我管理節點。AWS 區域如需詳細資訊，請參閱 Amazon EKS 使用者指南中的 Amazon 彈性區塊存放區 CSI 驅動程式。

`enabled`

指示是否已安裝預設的 Amazon EBS CSI 驅動程式。

必要：否

類型：布林值

`version`

Amazon EBS CSI 驅動程式附加元件的版本。版本必須符合DescribeAddonVersions動作傳回的其中一個版本。如需詳細資訊，請DescribeAddonVersions參閱 Amazon EKS API 參考

必要：否

類型：字串

屬性

`version`

叢集的 Kubernetes 版本。 AWS 電信網絡生成器支持庫伯尼特版本 1.23 到 1.29。

必要：是

類型：String

可能的值：

`access`

叢集端點存取。

必要：是

類型：String

可能的值：`PRIVATE` | `PUBLIC` | `ALL`

`cluster_role`

叢集管理的角色。

必要：是

類型：String

`tags`

要附加至資源的標籤。

必要：否

類型：清單

`ip_family`

指出叢集中服務和網繭位址的 IP 系列。

允許的值：`IPv4`，`IPv6`

預設值：`IPv4`

必要：否

類型：字串

要求

`subnets`

AWS. 網路. 子網路節點。

必要：是

類型：清單

範例

```
SampleEKS:
  type: tosca.nodes.AWS.Compute.EKS
  properties:
    version: "1.23"
    access: "ALL"
    cluster_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
    ip_family: "IPv6"
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  capabilities:
    multus:
      properties:
        enabled: true
        multus_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/MultusRole"
    ebs_csi:
      properties:
        enabled: true
        version: "v1.16.0-eksbuild.1"
  requirements:
    subnets:
    - SampleSubnet01
    - SampleSubnet02
```

AWS. 計算機. AuthRole

可 AuthRole 讓您將 IAM 角色新增至 Amazon EKS 叢集，以`aws-authConfigMap`便使用者可以使用 IAM 角色存取 Amazon EKS 叢集。

語法

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

```
    clusters: List
```

屬性

`role_mappings`

定義需要新增至 Amazon EKS 叢集`aws-authConfigMap`的 IAM 角色的對應清單。

`arn`

IAM 角色的 ARN。

必要：是

類型：String

`groups`

要指派給中定義之角色的 Kubernetes 群組。`arn`

必要：否

類型：清單

要求

`clusters`

一個 AWS.計算.eks 節點。

必要：是

類型：清單

範例

```
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. 計算機. ManagedNode

AWS TNB 支援 EKS 受管節點群組，以自動化 Amazon EKS Kubernetes 叢集節點 (Amazon EC2 執行個體) 的佈建和生命週期管理。若要建立 EKS 節點群組，您必須提供 AMI 的識別碼或 AMI 類型，為叢集工作者節點選擇 Amazon 機器映像 (AMI)。您也提供 Amazon EC2 key pair 以供 SSH 存取使用，以及節點群組的擴展屬性。您的節點群組必須與 EKS 叢集相關聯。您必須提供 Worker 節點的子網路。

或者，您可以將安全性群組、節點標籤和放置群組貼附至節點群組。

語法

```
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
```

功能

compute

定義 Amazon EKS 受管節點群組運算參數的屬性，例如 Amazon EC2 執行個體類型和 Amazon EC2 執行個體 AMI。

ami_type

Amazon EKS 支持的 AMI 類型。

必要：是

類型：String

可能的值：AL2_x86_64AL2_x86_64_GPUAL2_ARM_64CUSTOM| BOTTLEROCKET_ARM_64 | BOTTLEROCKET_x86_64 | BOTTLEROCKET_ARM_64_NVIDIA | BOTTLEROCKET_x86_64_NVIDIA

ami_id

AMI 的 ID。

必要：否

類型：字串

> **Note**
>
> 如果在模板中指定了ami_type和ami_id， AWS TNB 將僅使用ami_id值來創建EKSManagedNode。

instance_types

執行個體大小。

必要：是

類型：清單

`key_pair`

用於啟用 SSH 存取的 EC2 金鑰對。

必要：是

類型：String

`root_volume_encryption`

為 Amazon EBS 根磁碟區啟用 Amazon EBS 加密。如果未提供此內容， AWS TNB 會依預設加密 Amazon EBS 根磁碟區。

必要：否

預設：true

類型：布林值

`root_volume_encryption_key_arn`

AWS KMS 金鑰的 ARN。 AWS TNB 支持常規鍵 ARN，多區域鍵 ARN 和別名 ARN。

必要：否

類型：字串

> **Note**
>
> - 如果`root_volume_encryption`是假的，請不要包括`root_volume_encryption_key_arn`。
> - AWS TNB 支援 Amazon EBS 支援 AMI 的根磁碟區加密。
> - 如果 AMI 的根磁碟區已經加密，您必須包含 `root_volume_encryption_key_arn` for AWS TNB 才能重新加密根磁碟區。
> - 如果 AMI 的根磁碟區未加密， AWS TNB 會使用`root_volume_encryption_key_arn`來加密根磁碟區。
>
>   如果不包含`root_volume_encryption_key_arn` ， AWS TNB 會使用提供的預設金鑰 AWS Key Management Service 來加密根磁碟區。

- AWS TNB 不會解密加密的 AMI。

scaling

定義 Amazon EKS 受管節點群組擴展參數的屬性，例如所需的 Amazon EC2 執行個體數量，以及節點群組中 Amazon EC2 執行個體的最小和最大數量。

`desired_size`

其中的執行個體數目 NodeGroup。

必要：是

類型：整數

`min_size`

在此執行個體的最小數目 NodeGroup。

必要：是

類型：整數

`max_size`

在此執行個體的最大數目 NodeGroup。

必要：是

類型：整數

屬性

`node_role`

附加至亞馬遜 EC2 執行個體的 IAM 角色的 ARN。

必要：是

類型：String

`tags`

要附加到資源的標籤。

必要：否

類型：清單

要求

`cluster`

一個 AWS.計算.eks 節點。

必要：是

類型：String

`subnets`

AWS. 網路. 子網路節點。

必要：是

類型：清單

`network_interfaces`

一個 AWS. 網路 .eni 節點。請確定網路介面和子網路設定為相同的可用區域，否則建立會失敗。

當您設定時`network_interfaces`，如果您在 AWS.Compute.eks 節點中包含屬性，則 AWS TNB 會從`multus_role`屬性取得與 ENI 相關的權限。`multus`否則， AWS TNB 會從節點角色屬性取得與 ENI 相關的權限。

必要：否

類型：清單

`security_groups`

一個 AWS. 網路。 SecurityGroup節點。

必要：否

類型：清單

`placement_group`

一個節點。AWS計算。 PlacementGroup節點。

必要：否

類型：字串

`user_data`

一個節點。AWS計算。 UserData節點參考。使用者資料指令碼會傳遞至受管節點群組啟動的 Amazon EC2 執行個體。將執行自訂使用者資料所需的權限新增至傳遞至節點群組的 node_role。

必要：否

類型：字串

`labels`

節點標籤的列表。節點標籤必須具有名稱和值。使用下列條件建立標籤：

- 名稱和值必須以分隔=。
- 名稱和值的長度最多可為 63 個字元。
- 標籤可包含字母 (A-Z、a-z、)、數字 (0-9) 及下列字元：`[-, _, ., *, ?]`
- 名稱和值必須以字母數字?、或*字元開頭和結尾。

例如：`myLabelName1=*NodeLabelValue1`

必要：否

類型：清單

範例

```
SampleEKSManagedNode:
  type: tosca.nodes.AWS.Compute.EKSManagedNode
  capabilities:
    compute:
      properties:
        ami_type: "AL2_x86_64"
        instance_types:
          - "t3.xlarge"
        key_pair: "SampleKeyPair"
        root_volume_encryption: true
        root_volume_encryption_key_arn: "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    scaling:
```

```
      properties:
        desired_size: 1
        min_size: 1
        max_size: 1
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleRole"
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  requirements:
    cluster: SampleEKS
    subnets:
      - SampleSubnet
    network_interfaces:
      - SampleENI01
      - SampleENI02
    security_groups:
      - SampleSecurityGroup01
      - SampleSecurityGroup02
    placement_group: SamplePlacementGroup
    user_data: CustomUserData
    labels:
      - "sampleLabelName001=sampleLabelValue001"
      - "sampleLabelName002=sampleLabelValue002"
```

AWS. 計算機. SelfManagedNode

AWS TNB 支援 Amazon EKS 自我管理節點，以自動化 Amazon EKS Kubernetes 叢集節點 (Amazon EC2 執行個體) 的佈建和生命週期管理。若要建立 Amazon EKS 節點群組，您必須提供 AMI 的識別碼，為叢集工作者節點選擇 Amazon 機器映像 (AMI)。選擇性地提供亞 Amazon EC2 key pair 以進行 SSH 存取。您還必須提供執行個體類型以及所需的大小、最小和最大大小。您的節點群組必須與 Amazon EKS 叢集相關聯。您必須提供 Worker 節點的子網路。

或者，您可以將安全性群組、節點標籤和放置群組貼附至節點群組。

語法

```
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
```

功能

compute

定義 Amazon EKS 自我管理節點運算參數的屬性，例如 Amazon EC2 執行個體類型和 Amazon EC2 執行個體 AMI。

`ami_id`

用來啟動執行個體的 AMI 識別碼。 AWS TNB 支援利用 IMDSv2 的執行個體。如需詳細資訊，請參閱 法定版本。

必要：是

類型：String

`instance_type`

執行個體大小。

必要：是

類型：String

`key_pair`

用於啟用 SSH 存取的 Amazon EC2 key pair。

必要：是

類型：String

`root_volume_encryption`

為 Amazon EBS 根磁碟區啟用 Amazon EBS 加密。如果未提供此內容， AWS TNB 會依預設加密 Amazon EBS 根磁碟區。

必要：否

預設：true

類型：布林值

`root_volume_encryption_key_arn`

AWS KMS 金鑰的 ARN。 AWS TNB 支持常規鍵 ARN，多區域鍵 ARN 和別名 ARN。

必要：否

類型：字串

> **Note**
>
> - 如果`root_volume_encryption`是假的，請不要包括`root_volume_encryption_key_arn`。
> - AWS TNB 支援 Amazon EBS 支援 AMI 的根磁碟區加密。
> - 如果 AMI 的根磁碟區已經加密，您必須包含 `root_volume_encryption_key_arn` for AWS TNB 才能重新加密根磁碟區。
> - 如果 AMI 的根磁碟區未加密， AWS TNB 會使用`root_volume_encryption_key_arn`來加密根磁碟區。
>
>   如果不包含`root_volume_encryption_key_arn`， AWS TNB 會用 AWS Managed Services 來加密根磁碟區。
> - AWS TNB 不會解密加密的 AMI。

scaling

定義 Amazon EKS 自我管理節點擴展參數的屬性，例如所需的 Amazon EC2 執行個體數量，以及節點群組中 Amazon EC2 執行個體的最小和最大數量。

`desired_size`

其中的執行個體數目 NodeGroup。

必要：是

類型：整數

`min_size`

在此執行個體的最小數目 NodeGroup。

必要：是

類型：整數

`max_size`

在此執行個體的最大數目 NodeGroup。

必要：是

類型：整數

屬性

`node_role`

附加至亞馬遜 EC2 執行個體的 IAM 角色的 ARN。

必要：是

類型：String

`tags`

要附加到資源的標籤。標籤會傳播至資源建立的執行個體。

必要：否

類型：清單

要求

`cluster`

一個 AWS.計算.eks 節點。

必要：是

類型：String

`subnets`

AWS. 網路. 子網路節點。

必要：是

類型：清單

`network_interfaces`

一個 AWS. 網路 .eni 節點。請確定網路介面和子網路設定為相同的可用區域，否則建立會失敗。

當您設定時`network_interfaces`，如果您在 AWS.Compute.eks 節點中包含屬性，則 AWS TNB 會從`multus_role`屬性取得與 ENI 相關的權限。`multus`否則， AWS TNB 會從節點角色屬性取得與 ENI 相關的權限。

必要：否

類型：清單

`security_groups`

一個 AWS. 網路。 SecurityGroup節點。

必要：否

類型：清單

`placement_group`

一個節點。AWS計算。 PlacementGroup節點。

必要：否

類型：字串

`user_data`

一個節點。AWS計算。 UserData節點參考。使用者資料指令碼會傳遞至由自我管理節點群組啟動的 Amazon EC2 執行個體。將執行自訂使用者資料所需的權限新增至傳遞至節點群組的 node_role。

必要：否

類型：字串

`labels`

節點標籤的列表。節點標籤必須具有名稱和值。使用下列條件建立標籤：

- 名稱和值必須以分隔=。
- 名稱和值的長度最多可為 63 個字元。
- 標籤可以包含字母 (A-Z、a-z、)、數字 (0-9) 和下列字元：`[-, _, ., *, ?]`
- 名稱和值必須以字母數字?、或*字元開頭和結尾。

例如：`myLabelName1=*NodeLabelValue1`

必要：否

類型：清單

範例

```
SampleEKSSelfManagedNode:
  type: tosca.nodes.AWS.Compute.EKSSelfManagedNode
  capabilities:
    compute:
      properties:
        ami_id: "ami-123123EXAMPLE"
        instance_type: "c5.large"
        key_pair: "SampleKeyPair"
        root_volume_encryption: true
        root_volume_encryption_key_arn: "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    scaling:
      properties:
        desired_size: 1
        min_size: 1
        max_size: 1
```

```
  properties:
    node_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleNodeRole"
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
  requirements:
    cluster: SampleEKSCluster
    subnets:
      - SampleSubnet
    network_interfaces:
      - SampleNetworkInterface01
      - SampleNetworkInterface02
    security_groups:
      - SampleSecurityGroup01
      - SampleSecurityGroup02
    placement_group: SamplePlacementGroup
    user_data: CustomUserData
    labels:
      - "sampleLabelName001=sampleLabelValue001"
      - "sampleLabelName002=sampleLabelValue002"
```

AWS計算。 PlacementGroup

PlacementGroup 節點支援放置 Amazon EC2 執行個體的不同策略。

當您啟動新的 Amazon EC2Instance 時，Amazon EC2 服務會嘗試以將所有執行個體分散到基礎硬體的方式放置執行個體，以將相關故障降到最低。不過，您可以使用 置放群組 來影響一組 互相依存 執行個體的置放，以符合您的工作負載需求。

語法

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

屬性

`strategy`

用來放置 Amazon EC2 執行個體的策略。

必要：是

類型：String

可能的值：群集 | 分區 | 展開主機 | 展開

- 叢集 — 將執行個體封裝在可用區域內。此策略可讓工作負載達到高效能運算 (HPC) 應用程式典型之緊密結合 node-to-node 通訊所需的低延遲網路效能。
- PAR TITION — 將執行個體分散到邏輯分割區，這樣一個分割區中的執行個體群組就不會與不同分割區中的執行個體群組共用基礎硬體。大量分散和複寫的工作負載 (例如 Hadoop、Cassandra 和 Kafka) 通常採取此策略。
- SPREAD_RACK — 在不同的基礎硬體上放置一小組執行個體，以減少相關的故障。
- 「展示 _ 主機」 — 僅與「前哨」放置群組搭配使用。在不同的基礎硬體上放置一小組執行個體，以減少相關的故障。

`partition_count`

分割區數。

必要：僅當設定`strategy`為時才需要`PARTITION`。

類型：整數

可能的值：1 | 2 | 4

`tags`

您可以附加至放置群組資源的標籤。

必要：否

類型：清單

範例

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

```
      - tag_key=tag_value
```

AWS計算。 UserData

AWS TNB 支援透過網路服務描述元 (NSD) 中的 UserData節點啟動含有自訂使用者資料的 Amazon EC2 執行個體。如需有關自訂使用者資料的詳細資訊，請參閱 Amazon EC2 使用者指南中的使用者資料和殼層指令碼。

在網路實例化期間， AWS TNB 會透過使用者資料指令碼向叢集提供 Amazon EC2 執行個體註冊。當還提供自訂使用者資料時， AWS TNB 會合併這兩個指令碼，並將它們作為多重 MIME 指令碼傳遞給 Amazon EC2。自訂使用者資料指令碼會在 Amazon EKS 註冊指令碼之前執行。

若要在使用者資料指令碼中使用自訂變數，請在左{大括號!後加上驚嘆號。例如，若要在指令集MyVariable中使用，請輸入：{!MyVariable}

> **Note**
>
> - AWS TNB 支持最大 7 KB 的用戶數據腳本。
> - 由於 AWS TNB 用 AWS CloudFormation 來處理和呈現使用multimime者資料指令碼，因此請確保指令碼遵守所有規則。 AWS CloudFormation

語法

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

屬性

`implementation`

使用者資料指令碼定義的相對路徑。格式必須是：`./scripts/script_name.sh`

必要：是

類型：String

`content_type`

使用者資料指令碼的內容類型。

必要：是

類型：String

可能的值：`x-shellscript`

範例

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

AWS. 網路。 SecurityGroup

AWS TNB 支援安全群組以自動化 Amazon EC2 安全群組的佈建，您可以將這些群組連接到 Amazon EKS Kubernetes 叢集節點群組。

語法

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

屬性

`description`

安全性群組的說明。您最多可以使用 255 個字元來描述群組。您只能包含字母 (A-Z 和 a-z)、數字 (0-9)、空格及下列特殊字元：._-:/() #, @ [] +=&; {}! $*

必要：是

類型：String

`name`

安全性群組的名稱。您最多可以使用 255 個字元作為名稱。您只能包含字母 (A-Z 和 a-z)、數字 (0-9)、空格及下列特殊字元：._-:/() #, @ [] +=&; {}! $*

必要：是

類型：String

`tags`

您可以附加至安全性群組資源的標籤。

必要：否

類型：清單

要求

`vpc`

一個 AWS. 網路 .vPC 節點。

必要：是

類型：String

範例

```
SampleSecurityGroup001:
  type: tosca.nodes.AWS.Networking.SecurityGroup
  properties:
        description: "Sample Security Group for Testing"
        name: "SampleSecurityGroup"
        tags:
          - "Name=SecurityGroup"
          - "Environment=Testing"
      requirements:
        vpc: SampleVPC
```

AWS. 網路。 SecurityGroupEgressRule

AWS TNB 支援安全群組輸出規則，以自動化可連接至 .Network 的 Amazon EC2 安全群組輸出規則的佈建。 AWS SecurityGroup。請注意，您必須提供一個 cidr_ip/ 目的地安全組/目的地前綴清單作為輸出流量的目的地。

語法

```
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
```

屬性

`cidr_ip`

採用 CIDR 格式的 IPv4 位址範圍。您必須指定允許輸出流量的 CIDR 範圍。

必要：否

類型：字串

`cidr_ipv6`

以 CIDR 格式表示的 IPv6 位址範圍，適用於輸出流量。您必須指定目標安全群組 (`destination_security_group` 或 `destination_prefix_list`) 或 CIDR 範圍 (`cidr_ip` 或 `cidr_ipv6`)。

必要：否

類型：字串

`description`

輸出 (傳出) 安全群組規則的描述。您最多可以使用 255 個字元來描述規則。

必要：否

類型：字串

`destination_prefix_list`

現有 Amazon VPC 受管前置詞清單的前置詞清單識別碼。這是來自與安全性群組相關聯之節點群組執行個體的目的地。如需受管前置詞清單的詳細資訊，請參閱 Amazon VPC 使用者指南中的受管前置詞清單。

必要：否

類型：字串

`from_port`

如果通訊協定是 TCP 或 UDP，這是連接埠範圍的開頭。如果通訊協定是 ICMP 或 ICMPv6，這是類型編號。值 -1 表示所有 ICMP/ICMPv6 類型。若您指定所有 ICMP/ICMPv6 類型，您必須指定所有 ICMP/ICMPv6 代碼。

必要：否

類型：整數

`ip_protocol`

IP 通訊協定名稱 (TCP、UDP、ICMP、icmpv6) 或通訊協定號碼。使用 -1 指定所有通訊協定。授權安全性群組規則時，指定 -1 或 tcp、udp、icmp 或 icmpv6 以外的通訊協定號碼會允許所有連接埠上的流量，無論您指定的任何連接埠範圍為何。對於 tcp、udp 和 icmp，您必須指定連接埠範圍。對於 icmpv6，連接埠範圍是選擇性的；如果省略連接埠範圍，則允許所有類型和代碼的流量。

必要：是

類型：String

`to_port`

如果通訊協定是 TCP 或 UDP，這是連接埠範圍的結尾。如果通訊協定是 ICMP 或 ICMPv6，這是代碼。值 -1 表示所有 ICMP/ICMPv6 代碼。若您指定所有 ICMP/ICMPv6 類型，您必須指定所有 ICMP/ICMPv6 代碼。

必要：否

類型：整數

要求

`security_group`

要新增此規則的安全性群組識別碼。

必要：是

類型：String

`destination_security_group`

允許輸出流量之目的地安全性群組的 ID 或 TOSCA 參考。

必要：否

類型：字串

範例

```
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. 網路。 SecurityGroupIngressRule

AWS TNB 支援安全群組輸入規則，以自動化可連接至 .Network 的 Amazon EC2 安全群組入口規則的佈建。 AWS SecurityGroup。請注意，您必須提供一個來源作為輸入流量的來源。

語法

```
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
```

屬性

`cidr_ip`

採用 CIDR 格式的 IPv4 位址範圍。您必須指定允許輸入流量的 CIDR 範圍。

必要：否

類型：字串

`cidr_ipv6`

IPv6 位址範圍 (以 CIDR 格式表示)，適用於輸入流量。您必須指定來源安全群組 (`source_security_group` 或 `source_prefix_list`) 或 CIDR 範圍 (`cidr_ip` 或 `cidr_ipv6`)。

必要：否

類型：字串

`description`

輸入 (輸入) 安全性群組規則的說明。您最多可以使用 255 個字元來描述規則。

必要：否

類型：字串

`source_prefix_list`

現有 Amazon VPC 受管前置詞清單的前置詞清單識別碼。這是允許與安全性群組相關聯的節點群組執行個體接收流量的來源。如需受管前置詞清單的詳細資訊，請參閱 Amazon VPC 使用者指南中的受管前置詞清單。

必要：否

類型：字串

`from_port`

如果通訊協定是 TCP 或 UDP，這是連接埠範圍的開頭。如果通訊協定是 ICMP 或 ICMPv6，這是類型編號。值 -1 表示所有 ICMP/ICMPv6 類型。若您指定所有 ICMP/ICMPv6 類型，您必須指定所有 ICMP/ICMPv6 代碼。

必要：否

類型：整數

`ip_protocol`

IP 通訊協定名稱 (TCP、UDP、ICMP、icmpv6) 或通訊協定號碼。使用 -1 指定所有通訊協定。授權安全性群組規則時，指定 -1 或 tcp、udp、icmp 或 icmpv6 以外的通訊協定號碼會允許所有連接埠上的流量，無論您指定的任何連接埠範圍為何。對於 tcp、udp 和 icmp，您必須指定連接埠範圍。對於 icmpv6，連接埠範圍是選擇性的；如果省略連接埠範圍，則允許所有類型和代碼的流量。

必要：是

類型：String

`to_port`

如果通訊協定是 TCP 或 UDP，這是連接埠範圍的結尾。如果通訊協定是 ICMP 或 ICMPv6，這是代碼。值 -1 表示所有 ICMP/ICMPv6 代碼。若您指定所有 ICMP/ICMPv6 類型，您必須指定所有 ICMP/ICMPv6 代碼。

必要：否

類型：整數

要求

`security_group`

要新增此規則的安全性群組識別碼。

必要：是

類型：String

`source_security_group`

允許輸入流量之來源安全性群組的 ID 或 TOSCA 參考。

必要：否

類型：字串

範例

```
SampleSecurityGroupIngressRule:
      type: tosca.nodes.AWS.Networking.SecurityGroupIngressRule
      properties:
        ip_protocol: "tcp"
        from_port: 8000
        to_port: 9000
        description: "Ingress Rule for free5GC cluster on IPv6"
        cidr_ipv6: "2600:1f14:3758:ca00::/64"
      requirements:
        security_group: SampleSecurityGroup1
        source_security_group: SampleSecurityGroup2
```

AWS. 資源. 匯入

您可以將下列 AWS 資源匯入 AWS TNB：

- VPC
- 子網路
- 路由表
- Internet Gateway
- 安全群組

語法

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

```
    resource_id: String
```

屬性

`resource_type`

匯入至 AWS TNB 的資源類型。

必要：否

類型：清單

`resource_id`

匯入至 AWS TNB 的資源識別碼。

必要：否

類型：清單

範例

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

AWS. 網絡. 埃尼

網路介面是 VPC 中代表虛擬網路卡的邏輯網路元件。網路介面會根據其子網路自動或手動指派 IP 位址。在子網路中部署 Amazon EC2 執行個體後，您可以將網路界面附加到該執行個體，或從該 Amazon EC2 執行個體分離網路界面，然後重新連接到該子網路中的另一個 Amazon EC2 執行個體。裝置索引可識別附件順序中的位置。

語法

```
tosca.nodes.AWS.Networking.ENI:
  properties:
    device_index: Integer
```

```
    source_dest_check: Boolean
    tags: List
  requirements:
    subnet: String
    security_groups: List
```

屬性

device_index

裝置索引必須大於零。

必要：是

類型：整數

source_dest_check

指出網路介面是否執行來源/目的地檢查。true 值表示啟用檢查，false 值表示停用檢查。

允許的值：真，假

預設：true

必要：否

類型：布林值

tags

要附加到資源的標籤。

必要：否

類型：清單

要求

subnet

AWS. 網路. 子網路節點。

必要：是

類型：String

`security_groups`

一個 AWS. 網路。 SecurityGroup節點。

必要：否

類型：字串

範例

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

AWS.HookExecution

生命週期勾點可讓您執行自己的指令碼，做為基礎結構和網路實例化的一部分。

語法

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

```
    vpc: String
```

功能

execution

執行勾點指令碼之勾點執行引擎的屬性。

type

勾點執行引擎類型。

必要：否

類型：字串

可能的值：CODE_BUILD

要求

definition

一個 AWS. HookDefinition. 巴什節點。

必要：是

類型：String

vpc

一個 AWS. 網路 .vPC 節點。

必要：是

類型：String

範例

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

```
    vpc: SampleVPC
```

AWS. 網路。 InternetGateway

定義 AWS Internet Gateway 節點。

語法

```
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
```

功能

routing

定義 VPC 內路由連線的屬性。您必須包含dest_cidr或ipv6_dest_cidr屬性。

dest_cidr

用於目的地比對的 IPv4 CIDR 區塊。此屬性用於在中建立路由RouteTable，其值用作DestinationCidrBlock。

必要：否，如果您包含該ipv6_dest_cidr屬性。

類型：字串

ipv6_dest_cidr

用於目的地比對的 IPv6 CIDR 區塊。

必要：否，如果您包含該dest_cidr屬性。

類型：字串

屬性

`tags`

要附加到資源的標籤。

必要：否

類型：清單

`egress_only`

一個 IPv4 特定的屬性。指出網際網路閘道是否僅用於輸出通訊。當`egress_only`為 true 時，您必須定義`ipv6_dest_cidr`屬性。

必要：否

類型：布林值

要求

`vpc`

一個 AWS. 網路 .vPC 節點。

必要：是

類型：String

`route_table`

一個 AWS. 網路。 RouteTable節點。

必要：是

類型：String

範例

```
Free5GCIGW:
  type: tosca.nodes.AWS.Networking.InternetGateway
  properties:
    egress_only: false
```

```
  capabilities:
    routing:
      properties:
        dest_cidr: "0.0.0.0/0"
        ipv6_dest_cidr: "::/0"
  requirements:
    route_table: Free5GCRouteTable
    vpc: Free5GCVPC
Free5GCEGW:
  type: tosca.nodes.AWS.Networking.InternetGateway
  properties:
    egress_only: true
  capabilities:
    routing:
      properties:
        ipv6_dest_cidr: "::/0"
  requirements:
    route_table: Free5GCPrivateRouteTable
    vpc: Free5GCVPC
```

AWS. 網路。 RouteTable

路由表包含一組稱為路由的規則，用於確定來自 VPC 或閘道內子網路的網路流量導向的位置。您必須將路由表與 VPC 相關聯。

語法

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

屬性

`tags`

要附加至資源的標籤。

必要：否

類型：清單

要求

vpc

一個 AWS. 網路 .vPC 節點。

必要：是

類型：String

範例

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

AWS. 網路. 子網路

子網路是 VPC 中的一系列 IP 位址，而且必須完全位於一個可用區域內。您必須為子網路指定 VPC、CIDR 區塊、可用區域和路由表。您還必須定義子網路是私人還是公用子網路。

語法

```
tosca.nodes.AWS.Networking.Subnet:
  properties:
    type: String
    availability_zone: String
    cidr_block: String
    ipv6_cidr_block: String
    ipv6_cidr_block_suffix: String
    outpost_arn: String
    tags: List
  requirements:
    vpc: String
    route_table: String
```

屬性

`type`

指示在此子網路中啟動的執行個體是否會收到公有 IPv4 地址。

必要：是

類型：String

可能的值：PUBLIC | PRIVATE

`availability_zone`

子網路的可用區域。此欄位支援區 AWS 域內的可用 AWS 區域，例如 us-west-2 (美國西部 (奧勒岡))。例如，它也 AWS 支援可用區域內的本機區域us-west-2-lax-1a。

必要：是

類型：String

`cidr_block`

子網路的 CIDR 區塊。

必要：否

類型：字串

`ipv6_cidr_block`

用來建立 IPv6 子網路的 CIDR 區塊。如果您包含此屬性，請勿包含ipv6_cidr_block_suffix。

必要：否

類型：字串

`ipv6_cidr_block_suffix`

IPv6 CIDR 區塊的 2 位數十六進位字尾，適用於透過 Amazon VPC 建立的子網路。使用下列格式：*2-digit hexadecimal*::/*subnetMask*

如果您包含此屬性，請勿包含ipv6_cidr_block。

必要：否

類型：字串

`outpost_arn`

將在中建立 AWS Outposts 子網路的 ARN。如果您想要在上啟動 Amazon EKS 自我管理節點，請將此內容新增至 NSD 範本。 AWS Outposts有關更多信息，請參閱 Amazon EKS 用戶指南 AWS Outposts中的 Amazon EKS。

如果您將此內容新增至 NSD 範本，則必須將`availability_zone`內容的值設定為的可用區域。 AWS Outposts

必要：否

類型：字串

`tags`

要附加到資源的標籤。

必要：否

類型：清單

要求

`vpc`

一個 AWS. 網路 .vPC 節點。

必要：是

類型：String

`route_table`

一個 AWS. 網路。 RouteTable節點。

必要：是

類型：String

範例

```
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. 部署. 虛擬部署

NF 部署透過提供與其相關聯的基礎結構及應用程式建立模型。叢集屬性指定 EKS 叢集來託管您的 NF。vnfs 屬性會指定部署的網路功能。您也可以提供 pre_create 和 post_create 類型的選擇性生命週期勾點作業，以執行部署特定的指示，例如呼叫庫存管理系統 API。

語法

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

要求

`deployment`

AWS部署 .vnf 部署節點。

必要：否

類型：字串

`cluster`

一個 AWS.計算.eks 節點。

必要：是

類型：String

`vnfs`

一個 AWS.VNF 節點。

必要：是

類型：String

介面

掛鉤

定義執行生命週期掛接的階段。

`pre_create`

一個 AWS. HookExecution節點。此勾點會在`VNFDeployment`節點部署之前執行。

必要：否

類型：字串

`post_create`

一個 AWS. HookExecution節點。此勾點會在`VNFDeployment`節點部署之後執行。

必要：否

類型：字串

範例

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

AWS. 網路. VPC

您必須為虛擬私有雲 (VPC) 指定 CIDR 區塊。

語法

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

屬性

`cidr_block`

VPC 的 IPv4 網路範圍 (以 CIDR 表示法表示)。

必要：是

類型：String

`ipv6_cidr_block`

用來建立虛擬私人雲端的 IPv6 CIDR 區塊。

允許的值：AMAZON_PROVIDED

必要：否

類型：字串

dns_support

指示 VPC 中啟動的執行個體是否會收到 DNS 主機名稱。

必要：否

類型：布林值

預設：false

tags

要附加至資源的標籤。

必要：否

類型：清單

範例

```
SampleVPC:
  type: tosca.nodes.AWS.Networking.VPC
  properties:
    cidr_block: "10.100.0.0/16"
    ipv6_cidr_block: "AMAZON_PROVIDED"
    dns_support: true
    tags:
      - "Name=SampleVPC"
      - "Environment=Testing"
```

AWS. 網路.

您可以透過子網路定義公用或私人 NAT 閘道節點。對於公共閘道，如果您沒有提供彈性 IP 配置 ID，AWS TNB 會為您的帳戶配置彈性 IP，並將其關聯到閘道。

語法

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

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

屬性

`subnet`

. AWS網路. 子網路節點參考。

必要：是

類型：String

`internet_gateway`

該 AWS. 網路。 InternetGateway節點參考。

必要：是

類型：String

屬性

`type`

指出閘道是公用還是私人。

允許的值：`PUBLIC` , `PRIVATE`

必要：是

類型：String

`eip_allocation_id`

代表彈性 IP 位址配置的識別碼。

必要：否

類型：字串

`tags`

要附加至資源的標籤。

必要：否

類型：清單

範例

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

AWS. 網路. 路線

您可以定義將目的路由路由與 NAT 閘道相關聯作為目標資源的路由節點，並將路由新增至相關聯的路由表。

語法

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

屬性

`dest_cidr_blocks`

目的地 IPv4 路由到目標資源的清單。

必要：是

類型：清單

成員類型：字串

屬性

`nat_gateway`

AWS. 網路. 納入閘道節點參考。

必要：是

類型：String

`route_table`

該 AWS. 網路。 RouteTable節點參考。

必要：是

類型：String

範例

```
Free5GCRoute:
  type: tosca.nodes.AWS.Networking.Route
  properties:
    dest_cidr_blocks:
      - 0.0.0.0/0
      - 10.0.0.0/28
  requirements:
    nat_gateway: Free5GCNatGateway01
    route_table: Free5GCRouteTable
```

共同節點

定義要在 NSD 和 VNFD 中使用的節點。

- AWS。HookDefinition. 巴什

AWS.HookDefinition.Bash

定義一個AWS HookDefinition在bash。

語法

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

屬性

`implementation`

掛接定義的相對路徑。格式必須是：`./hooks/script_name.sh`

必要：是

類型：String

`environment_variables`

鉤子 bash 腳本的環境變量。請使用下列格式：**envName**=**envValue**使用以下正則表達式：`^[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+=[a-zA-Z0-9]+[a-zA-Z0-9\-\_]*[a-zA-Z0-9]+$`

確保**envName**=**envValue**值符合以下條件：

- 請勿使用空格。
- 開始**envName**帶有一個字母（A-Z 或 a-z）或數字（0-9）。
- 請勿以下列項目啟動環境變數名稱AWSTNB 保留的關鍵字（不區分大小寫）：
  - 代碼生成
  - TNB
  - 家
  - AWS
- 您可以使用任意數量的字母（A-Z 或 a-z），數字（0-9）和特殊字符-和_為了**envName**和**envValue**。

範例：A123-45xYz=Example_789

必要：否

類型：清單

`execution_role`

勾點執行的角色。

必要：是

類型：String

範例

```
SampleHookScript:
  type: tosca.nodes.AWS.HookDefinition.Bash
  properties:
    implementation: "./hooks/myhook.sh"
    environment_variables:
      - "variable01=value01"
      - "variable02=value02"
    execution_role: "arn:aws:iam::${AWS::TNB::AccountId}:role/SampleHookPermission"
```

AWS 電信網絡構建器中的安全性

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，這些架構是為了滿足對安全性最敏感的組織的需求而建置的。

安全是 AWS 與您之間共同承擔的責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端安全性 — AWS 負責保護中執行 AWS 服務的基礎架構 AWS 雲端。 AWS 還為您提供可以安全使用的服務。若要瞭解適用於 AWS Telco 網路產生器的法規遵循方案，請參閱遵循規範計劃的AWS 服務範圍內的AWS 服務 (遵循)。
- 雲端安全性 — 您的責任取決於您使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您瞭解如何在使用 AWS TNB 時套用共同責任模型。下列主題說明如何設定 AWS TNB 以符合安全性與合規性目標。您還將學習如何使用其他 AWS 服務來幫助您監控和保護 AWS TNB 資源。

目錄

AWS TNB 中的資料保護

AWS 共同責任模型適用於 AWS 電信網路建置器中的資料保護。如此模型所述， AWS 負責保護執行所有 AWS 雲端. 您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用主控台、API 或 AWS SDK AWS 服務 使用 AWS TNB 或其他工作時。 AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

標籤處理

當您關閉 AWS 帳戶時， AWS TNB 會將您的資料標記為要刪除，並將其從任何使用中移除。如果您在 90 天內重新啟用 AWS 帳戶， AWS TNB 會還原您的資料。在 120 天之後， AWS TNB 會永久刪除您的資料。 AWS TNB 還會終止您的網絡並刪除您的功能包和網絡包。

靜態加密

AWS TNB 始終加密存儲在靜態服務中的所有數據，而不需要任何額外的配置。此加密是自動透過 AWS Key Management Service。

傳輸中加密

AWS TNB 使用傳輸層安全性 (TLS) 1.2 保護傳輸中的所有資料。

您有責任在模擬代理程式與其用戶端之間加密資料。

網際網路流量隱私權

AWS TNB 運算資源位於所有客戶共用的虛擬私有雲 (VPC) 中。所有內部 AWS TNB 流量都保留在 AWS 網絡中，並且不遍歷互聯網。您的模擬代理程式與其用戶端之間的連線會透過網際網路路由。

AWS TNB 的身分識別與存取管理

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。 AWS 服務 IAM 管理員控制哪些人可以驗證 (登入) 和授權 (具有權限) 以使用 AWS TNB 資源。IAM 是您可以使用的 AWS 服務 ，無需額外付費。

目錄

物件

根據您在 AWS TNB 中執行的工作，使用方式 AWS Identity and Access Management (IAM) 會有所不同。

服務使用者 — 如果您使用 AWS TNB 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 AWS TNB 功能來完成工作時，您可能需要額外的權限。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取 AWS TNB 中的特徵，請參閱AWS 電信網路建置器身分與存取疑難排解。

服務管理員 — 如果您負責公司的 AWS TNB 資源，您可能擁有 AWS TNB 的完整存取權。判斷服務使用者應存取哪些 AWS TNB 功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何將 IAM 與 AWS TNB 搭配使用，請參閱AWS 電信網絡構建器如何與 IAM 配合使用。

IAM 管理員 — 如果您是 IAM 管理員，您可能想要瞭解如何撰寫政策來管理 AWS TNB 存取權的詳細資訊。若要檢視您可以在 IAM 中使用的 AWS TNB 身分型政策範例，請參閱。AWS 電信網路建置器的身分識別原則範例

使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。 AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。 AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》 AWS 帳戶中的如何登入您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署您的要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的簽署 AWS API 請求。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的多重要素驗證和 IAM 使用者指南中的在 AWS中使用多重要素驗證 (MFA)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時認證 AWS 服務 來存取。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identi ty Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。 AWS Directory Service同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的相關資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

IAM 角色是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以切換角色，在中暫時擔任 IAM 角色。 AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的使用 IAM 角色。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源（而不是使用角色作為代理）。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的 IAM 角色與資源類型政策的差異。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。

- 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
- 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務。
- 服務連結角色 — 服務連結角色是連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的建立 IAM 角色 (而非使用者) 的時機。

使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS ，當與身分識別或資源相關聯時，會定義其權限。 AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶. 受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。 AWS WAF若要進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政

策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 實體許可範圍。

- 服務控制策略（SCP）— SCP 是 JSON 策略，用於指定中組織或組織單位（OU）的最大權限。AWS Organizations AWS Organizations 是一種用於分組和集中管理您企業擁 AWS 帳戶 有的多個服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的 SCP 運作方式。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS 電信網絡構建器如何與 IAM 配合使用

在您使用 IAM 管理 AWS TNB 的存取權限之前，請先了解哪些 IAM 功能可與 AWS TNB 搭配使用。

您可以搭配 AWS 電信網路建置器使用的 IAM 功能

IAM 功能	AWS 支援 TNB
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是

IAM 功能	AWS 支援 TNB
主體許可	是
服務角色	否
服務連結角色	否

若要深入瞭解 AWS TNB 和其他 AWS 服務如何搭配大多數 IAM 功能運作，請參閱 IAM 使用者指南中的搭配 IAM 使用的AWS 服務。

TNB 的身分識別型原則 AWS

支援身分型政策	是

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

TNB 的以身分識別為基礎的原則範例 AWS

若要檢視 AWS TNB 身分型原則的範例，請參閱。AWS 電信網路建置器的身分識別原則範例

TNB 內 AWS 以資源為基礎的政策

支援以資源基礎的政策	否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

若要啟用跨帳戶存取，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 角色與資源型政策有何差異。

AWS TNB 的政策動作

支援政策動作	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS TNB 動作清單，請參閱服務授權參考資料中的 T AWS elco 網路建置器定義的動作。

AWS TNB 中的策略動作在動作之前使用以下前綴：

```
tnb
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

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

您也可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 `List` 文字的所有動作，請包含以下動作：

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

若要檢視 AWS TNB 身分型原則的範例，請參閱。AWS 電信網路建置器的身分識別原則範例

適用於 AWS TNB 的政策資源

支援政策資源	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

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

要查看 AWS TNB 資源類型及其 ARN 的列表，請參閱服務授權參考中的 AWS 電信網絡生成器定義的資源。若要瞭解可以使用哪些動作指定每個資源的 ARN，請參閱AWS 電信網路建置器定義的動作。

若要檢視 AWS TNB 身分型原則的範例，請參閱。AWS 電信網路建置器的身分識別原則範例

TNB 的政策條件 AWS 金鑰

支援服務特定政策條件金鑰	是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯OR運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的AWS 全域條件內容金鑰。

若要查看 AWS TNB 條件金鑰清單，請參閱服務授權參考資料中的 T AWS elco 網路建置器的條件金鑰。若要瞭解您可以使用條件金鑰的動作和資源，請參閱 AWS Telco 網路建置器定義的動作。

若要檢視 AWS TNB 身分型原則的範例，請參閱。AWS 電信網路建置器的身分識別原則範例

在 TNB 中的 AWS ACL

支援 ACL	否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

阿巴克與 AWS TNB

支援 ABAC (政策中的標籤)	是

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

若要根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件金鑰，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的什麼是 ABAC?。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的使用屬性型存取控制 (ABAC)。

搭配 AWS TNB 使用臨時登入資料

支援臨時憑證	是

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料搭配AWS 服務 使用，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的切換至角色 (主控台)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而非使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

TNB 的 AWS 跨服務主體權限

支援轉寄存取工作階段 (FAS)	是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。

AWS TNB 的服務角色

支援服務角色	否

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務。

TNB 的 AWS 服務連結角色

支援服務連結角色。	否

服務連結角色是一種連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

AWS 電信網路建置器的身分識別原則範例

根據預設，使用者和角色沒有建立或修改 AWS TNB 資源的權限。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行工作。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

有關 AWS TNB 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARN 格式，請參閱服務授權參考中的 T AWS elco 網路 Builder 的動作、資源和條件索引鍵。

目錄

政策最佳實務

以身分識別為基礎的原則會決定某人是否可以在您的帳戶中建立、存取或刪除 AWS TNB 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始將權限授與使用者和工作負載，請使用可授與許多常見使用案例權限的AWS 受管理原則。它們在您的 AWS 帳戶. 建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定 AWS 服務) 使用 AWS CloudFormation。如需更多資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access

Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 IAM Access Analyzer 政策驗證。

- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。若要在呼叫 API 作業時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 IAM 使用者指南中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

使用 AWS TNB 主控台

若要存取 AWS Telco 網路建置器主控台，您必須擁有最低限度的權限集。這些權限必須允許您 AWS 帳戶列出和檢視有關. AWS 如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

服務角色政策範例

身為系統管理員，您擁有並管理 AWS TNB 所建立的資源，如環境和服務範本所定義。您必須將 IAM 服務角色附加到您的帳戶，以允許 AWS TNB 為網路生命週期管理建立資源。

IAM 服務角色允許 AWS TNB 代表您呼叫資源，以實例化和管理您的網路。如果您指定服務角色，AWS TNB 會使用該角色的認證。

您使用 IAM 服務建立服務角色及其許可政策。如需有關建立服務角色的詳細資訊，請參閱《IAM 使用者指南》中的建立角色以將權限委派給 AWS 服務。

AWS TNB 服務角色

身為平台小組的成員，您可以以管理員身分建立 AWS TNB 服務角色，並將其提供給 AWS TNB。此角色可讓 AWS TNB 呼叫其他服務 (例如 Amazon Elastic Kubernetes Service)，並 AWS CloudFormation 為您的網路佈建所需的基礎設施，以及佈建 NSD 中定義的網路功能。

建議您針對 AWS TNB 服務角色使用下列 IAM 角色和信任政策。設定此原則的權限限定範圍時，請記住， AWS TNB 可能會失敗，並顯示拒絕存取錯誤，導致您的原則中限制的資源。

下列程式碼顯示 AWS TNB 服務角色原則：

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Action": [
                "sts:GetCallerIdentity"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "AssumeRole"
        },
        {
            "Action": [
                "tnb:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBPolicy"
        },
        {
            "Action": [
                "iam:AddRoleToInstanceProfile",
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:TagInstanceProfile",
                "iam:UntagInstanceProfile"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "IAMPolicy"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "eks.amazonaws.com",
                        "eks-nodegroup.amazonaws.com"
                    ]
                }
            },
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
```

```
            "Effect": "Allow",
            "Sid": "TNBAccessSLRPermissions"
        },
        {
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateOrUpdateTags",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteTags",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:DescribeTags",
                "autoscaling:UpdateAutoScalingGroup",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteLaunchTemplateVersions",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeTags",
                "ec2:GetLaunchTemplateData",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RunInstances",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:CreateInternetGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSubnet",
                "ec2:CreateTags",
                "ec2:CreateVpc",
                "ec2:DeleteInternetGateway",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSubnet",
```

```
                "ec2:DeleteTags",
                "ec2:DeleteVpc",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeInstances",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySecurityGroupRules",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:AllocateAddress",
                "ec2:AssignIpv6Addresses",
                "ec2:AssociateAddress",
                "ec2:AssociateNatGatewayAddress",
                "ec2:AssociateVpcCidrBlock",
                "ec2:CreateEgressOnlyInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:DeleteEgressOnlyInternetGateway",
                "ec2:DeleteNatGateway",
                "ec2:DescribeAddresses",
                "ec2:DescribeEgressOnlyInternetGateways",
                "ec2:DescribeNatGateways",
                "ec2:DisassociateAddress",
                "ec2:DisassociateNatGatewayAddress",
                "ec2:DisassociateVpcCidrBlock",
                "ec2:ReleaseAddress",
                "ec2:UnassignIpv6Addresses",
                "ec2:DescribeImages",
                "eks:CreateCluster",
                "eks:ListClusters",
                "eks:RegisterCluster",
                "eks:TagResource",
                "eks:DescribeAddonVersions",
                "events:DescribeRule",
                "iam:GetRole",
                "iam:ListAttachedRolePolicies",
                "iam:PassRole"
            ],
```

```
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBAccessComputePerms"
        },
        {
            "Action": [
                "codebuild:BatchDeleteBuilds",
                "codebuild:BatchGetBuilds",
                "codebuild:CreateProject",
                "codebuild:DeleteProject",
                "codebuild:ListBuildsForProject",
                "codebuild:StartBuild",
                "codebuild:StopBuild",
                "events:DeleteRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "s3:CreateBucket",
                "s3:GetBucketAcl",
                "s3:GetObject",
                "eks:DescribeNodegroup",
                "eks:DeleteNodegroup",
                "eks:AssociateIdentityProviderConfig",
                "eks:CreateNodegroup",
                "eks:DeleteCluster",
                "eks:DeregisterCluster",
                "eks:UntagResource",
                "eks:DescribeCluster",
                "eks:ListNodegroups",
                "eks:CreateAddon",
                "eks:DeleteAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonVersions",
                "s3:PutObject",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateTerminationProtection"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/tnb*",
                "arn:aws:codebuild:*:*:project/tnb*",
                "arn:aws:logs:*:*:log-group:/aws/tnb*",
```

```
                "arn:aws:s3:::tnb*",
                "arn:aws:eks:*:*:addon/tnb*/*/*",
                "arn:aws:eks:*:*:cluster/tnb*",
                "arn:aws:eks:*:*:nodegroup/tnb*/tnb*/*",
                "arn:aws:cloudformation:*:*:stack/tnb*"
            ],
            "Effect": "Allow",
            "Sid": "TNBAccessInfraResourcePerms"
        },
        {
            "Sid": "CFNTemplatePerms",
            "Effect": "Allow",
            "Action": [
              "cloudformation:GetTemplateSummary"
            ],
            "Resource": "*"
         },
        {
            "Sid": "ImageAMISSMPerms",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": [
                "arn:aws:ssm:*::parameter/aws/service/eks/optimized-ami/*",
                "arn:aws:ssm:*::parameter/aws/service/bottlerocket/*"
            ]
        },
        {
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TaggingPolicy"
        },
        {
            "Action": [
                "outposts:GetOutpost"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "OutpostPolicy"
        }
```

```
    ]
}
```

下列程式碼顯示 AWS TNB 服務信任原則：

```
{
  "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 Amazon EKS 叢集的 TNB 服務角色

在 NSD 中建立 Amazon EKS 資源時，您需要提供`cluster_role`屬性來指定將使用哪個角色來建立 Amazon EKS 叢集。

下列範例顯示為 Amazon EKS 叢集政策建立 AWS TNB 服務角色的範 AWS CloudFormation 本。

```
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"
```

如需有關使用 AWS CloudFormation 範本的 IAM 角色的詳細資訊，請參閱使用AWS CloudFormation者指南中的以下各節：

- AWS::IAM::Role
- 選取堆疊範本

AWS Amazon EKS 節點群組的 TNB 服務角色

在 NSD 中建立 Amazon EKS 節點群組資源時，您需要提供`node_role`屬性以指定將使用哪個角色來建立 Amazon EKS 節點群組。

下列範例顯示為 Amazon EKS 節點群組原則建立 AWS TNB 服務角色的範 AWS CloudFormation 本。

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  TNBEKSNodeRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: "TNBEKSNodeRole"
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - "sts:AssumeRole"
      Path: /
      ManagedPolicyArns:
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKSWorkerNodePolicy"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/AmazonEKS_CNI_Policy"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly"
        - !Sub "arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEBSCSIDriverPolicy"
      Policies:
        - PolicyName: EKSNodeRoleInlinePolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "logs:DescribeLogStreams"
                  - "logs:PutLogEvents"
                  - "logs:CreateLogGroup"
                  - "logs:CreateLogStream"
                Resource: "arn:aws:logs:*:*:log-group:/aws/tnb/tnb*"
        - PolicyName: EKSNodeRoleIpv6CNIPolicy
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "ec2:AssignIpv6Addresses"
                Resource: "arn:aws:ec2:*:*:network-interface/*"
```

如需有關使用 AWS CloudFormation 範本的 IAM 角色的詳細資訊，請參閱使用AWS CloudFormation者指南中的以下各節：

- AWS::IAM::Role
- 選取堆疊範本

AWS 穆圖斯的 TNB 服務角色

當您在 NSD 中建立 Amazon EKS 資源，並希望將 Multus 作為部署範本的一部分進行管理時，必須提供`multus_role`屬性以指定將用於管理 Mults 的角色。

下列範例顯示為 Multus 原則建立 AWS TNB 服務角色的範 AWS CloudFormation 本。

```
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: "*"
```

如需有關使用 AWS CloudFormation 範本的 IAM 角色的詳細資訊，請參閱使用AWS CloudFormation者指南中的以下各節：

- AWS::IAM::Role
- 選取堆疊範本

AWS 生命週期勾點原則的 TNB 服務角色

當您的 NSD 或網路功能套件使用生命週期勾點時，您需要一個服務角色來建立執行生命週期勾點的環境。

> **Note**
>
> 您的生命週期勾點原則應以您的生命週期勾點嘗試執行的動作為基礎。

下列範例顯示為生命週期勾點原則建立 AWS TNB 服務角色的範 AWS CloudFormation 本。

```
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"
```

如需有關使用 AWS CloudFormation 範本的 IAM 角色的詳細資訊，請參閱使用AWS CloudFormation者指南中的以下各節：

- AWS::IAM::Role
- 選取堆疊範本

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

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

```
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 電信網路建置器身分與存取疑難排解

使用下列資訊可協助您診斷和修正使用 AWS TNB 和 IAM 時可能遇到的常見問題。

問題

- 我沒有在 AWS TNB 中執行操作的授權
- 我沒有授權執行 iam：PassRole
- 我想允許我以外的人訪問我 AWS 帳戶 的 AWS TNB 資源

我沒有在 AWS TNB 中執行操作的授權

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 tnb:*GetWidget* 許可時發生。

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

在此情況下，必須更新 Mateo 政策，允許他使用 tnb:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我沒有授權執行 iam：PassRole

如果您收到未獲授權執行`iam:PassRole`動作的錯誤訊息，則必須更新您的原則，以允許您將角色傳遞給 AWS TNB。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM 使用者`marymajor`嘗試使用主控台在 AWS TNB 中執行動作時，就會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

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

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我想允許我以外的人訪問我 AWS 帳戶 的 AWS TNB 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要瞭解 AWS TNB 是否支援這些功能，請參閱AWS 電信網絡構建器如何與 IAM 配合使用。
- 若要了解如何提供您所擁有資源 AWS 帳戶 的存取權，請參閱《IAM 使用者指南》中的另一個您擁有 AWS 帳戶 的 IAM 使用者提供存取權限。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的提供第三方 AWS 帳戶 擁有的存取權。
- 若要了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 IAM 使用者指南中的 IAM 角色與資源型政策的差異。

適用於 AWS TNB 的合規性驗證

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱AWS 服務 遵循規範計劃方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱AWS 規範計劃AWS。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全性與合規性快速入門指南 — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- 在 Amazon Web Services 上架構 HIPAA 安全性與合規性 — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 用程式。

> **Note**
>
> 並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 HIPAA 資格服務參照。

- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- AWS 客戶合規指南 — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- 使用AWS Config 開發人員指南中的規則評估資源 — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 Security Hub controls reference。
- Amazon GuardDuty — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。 GuardDuty 可協助您因應各種合規性需求，例如 PCI DSS，滿足特定合規性架構所規定的入侵偵測需求。
- AWS Audit Manager— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

AWS TNB 的韌性

AWS 全球基礎架構是圍繞 AWS 區域 和可用區域建立的。 AWS 區域 提供多個實體分離和隔離的可用區域，這些區域透過低延遲、高輸送量和高度備援的網路連線。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱AWS 全域基礎結構。

AWS TNB 會在您選擇的 AWS 區域中的虛擬私有雲 (VPC) 中的 EKS 叢集上執行網路服務。

AWS TNB 中的基礎架構安全性

作為託管服務， AWS 電信網絡構建器受到全球網絡安 AWS 全的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱AWS 雲端安全 若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱安全性支柱架構良 AWS 好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫透過網路存取 AWS TNB。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

以下是一些共同責任的例子：

- AWS 負責保護支援 AWS TNB 的元件，包括：
  - 運算執行個體 (也稱為背景工作)
  - 內部資料庫
  - 內部元件之間的網路通訊
  - AWS TNB 應用程式開發介面 (API)
  - AWS 軟體開發套件 (SDK)
- 您有責任保護您對 AWS 資源和工作負載元件的存取，包括 (但不限於)：
  - IAM 使用者、群組、角色和政策
  - 用於存放 AWS TNB 資料的 S3 儲存貯體

- 您用來支援透過 AWS TNB 佈建之網路服務的其 AWS 服務 他資源
- 您的應用程式碼
- 您透過 AWS TNB 佈建的網路服務與其用戶端之間的連線

> **Important**
>
> 您必須負責實作災難復原計畫，以有效地復原透過 AWS TNB 佈建的網路服務。

網路連線安全性模型

您透過 AWS TNB 佈建的網路服務，會在位於您所選 AWS 區域的虛擬私有雲端 (VPC) 內的運算執行個體上執行。VPC 是 AWS 雲端中的虛擬網路，可依工作負載或組織實體隔離基礎結構。VPC 中運算執行個體之間的通訊會保留在 AWS 網路內，而且不會透過網際網路傳輸。一些內部服務通信穿過互聯網，並且是加密的。透過 AWS TNB 為在相同區域中執行的所有客戶佈建的網路服務共用相同的 VPC。透過 AWS TNB 為不同客戶佈建的網路服務會在同一個 VPC 內使用個別的運算執行個體。

您的網路服務用戶端與您在 AWS TNB 中的網路服務之間的通訊會遍歷網際網路。 AWS TNB 不會管理這些連線。保護您的客戶連接是您的責任。

您透過 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和 AWS SDK 與 AWS TNB 的連線會經過加密。

法定版本

AWS TNB 支援利用執行個體中繼資料服務版本 2 (IMDSv2) (工作階段導向方法) 的執行個體。包含比 IMDSV1 更高的安全性。如需詳細資訊，請參閱增強 Amazon EC2 執行個體中繼資料服務，針對開放式防火牆、反向代理和 SSRF 弱點新增深度防禦。

啟動執行個體時，您必須使用 IMDSv2。如需有關 IMDSv2 的詳細資訊，請參閱 Amazon EC2 使用者指南中的使用 IMDSv2。

監控AWSTNB

監控是維持可靠性，可用性和性能的重要組成部分AWSTNB 和您的其他AWS解決方案。AWS提供 AWS CloudTrail觀看AWSTNB, 報告時出現錯誤, 並在適當時採取自動操作.

使用CloudTrail捕獲有關呼叫的詳細信息AWSAPI。您可以將這些呼叫做為日誌檔存放在 Amazon S3 中。你可以使用這些CloudTrail記錄以確定撥打哪個呼叫的信息，呼叫來自的源 IP 地址，誰撥打電話以及何時進行呼叫。

該CloudTrail記錄檔包含呼叫 API 動作的相關資訊AWSTNB。它們也包含從 Amazon EC2 和亞馬遜 EBS 等服務呼叫 API 動作的相關資訊。

使用記錄AWS電信網絡生成器 API 調用AWS CloudTrail

AWSTelco 網絡生成器已與整合AWS CloudTrail，這項服務可提供由用戶、角色或AWS TNB 中AWS 服務所採取之動作的記錄。 CloudTrail 擷取AWS TNB 的 API 呼叫當作事件。擷取的呼叫包括從 AWS TNB 主控台進行的呼叫，以及對AWS TNB API 操作的程式碼呼叫。如果您建立追蹤，就可以將 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括AWS TNB 的事件。即使未設定追蹤，您依然可以在 CloudTrail 主控台中檢視最新的事件。您可以使用收集的資訊來 CloudTrail判斷對AWS TNB 提出的請求、提出請求的 IP 地址、提出請求的對象、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱使AWS CloudTrail用者指南。

AWS中的 TNB 資訊 CloudTrail

CloudTrail 當您建立帳戶AWS 帳戶時，系統即會在中啟用。此外，AWSTNB 中發生活動時，系統便會將該活動記錄至 CloudTrail 事件，並將其他AWS服務事件記錄到事件中。您可以檢視、搜尋和下載 AWS 帳戶 的最新事件。如需詳細資訊，請參閱使用 CloudTrail 事件歷史記錄檢視事件。

若要持續記錄您的事件AWS 帳戶，包括AWS TNB 的事件，請建立線索。線索能 CloudTrail 將日誌檔案交付至 Amazon S3 通知 依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他AWS服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務與整合
- 設定的 Amazon SNS 通知 CloudTrail

- 接收 CloudTrail 多個區域的 CloudTrail 日誌檔案

所有AWS TNB 動作都由記錄， CloudTrail 並記錄在AWS電信網路產生器 API 參考中。例如，呼叫CreateSolNetworkInstance和CreateSolNetworkPackage動作會CreateSolFunctionPackage在 CloudTrail 記錄檔中產生項目。

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否透過根或 AWS Identity and Access Management (IAM) 使用者憑證來提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解AWS TNB 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付至您指定的 Amazon S3 儲存貯體。 CloudTrail 日誌檔案包含一個或多個日誌項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。 CloudTrail 日誌檔案並非依公有 API 呼叫的堆疊追蹤，因此不會以任何特定順序出現。

以下範例顯示的是展示CreateSolFunctionPackage動作的 CloudTrail 日誌項目。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:example",
        "arn": "arn:aws:sts::111222333444:assumed-role/example/user",
        "accountId": "111222333444",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111222333444:role/example",
                "accountId": "111222333444",
                "userName": "example"
            },
            "webIdFederationData": {},
```

```
            "attributes": {
                "creationDate": "2023-02-02T01:42:39Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-02-02T01:43:17Z",
    "eventSource": "tnb.amazonaws.com",
    "eventName": "CreateSolFunctionPackage",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "XXX.XXX.XXX.XXX",
    "userAgent": "userAgent",
    "requestParameters": null,
    "responseElements": {
        "vnfPkgArn": "arn:aws:tnb:us-east-1:111222333444:function-package/
fp-12345678abcEXAMPLE",
        "id": "fp-12345678abcEXAMPLE",
        "operationalState": "DISABLED",
        "usageState": "NOT_IN_USE",
        "onboardingState": "CREATED"
    },
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111222333444",
    "eventCategory": "Management"
}
```


AWS TNB 部署工作

瞭解部署工作以有效監控部署並加快採取行動。

下表列出 AWS TNB 部署作業：

在 2024 年 3 月 7 日之前開始的部署工作名稱	2024 年 3 月 7 日以後開始的部署工作名稱	Task description (任務描述)
AppInstallation	ClusterPluginInstall	在 Amazon EKS 集群上安裝多個插件。

在 2024 年 3 月 7 日之前開始的部署工作名稱	2024 年 3 月 7 日以後開始的部署工作名稱	Task description (任務描述)
AppUpdate	名稱沒有變化	更新網路執行個體中已安裝的網路功能。
-	ClusterPluginUninstall	解除安裝 Amazon EKS 叢集上的外掛程式。
ClusterStorageClassConfiguration	名稱沒有變化	在 Amazon EKS 叢集上設定儲存類別 (CSI 驅動程式)。
FunctionDeletion	名稱沒有變化	從 AWS TNB 資源刪除網路功能。
FunctionInstantiation	FunctionInstall	使用 HELM 部署網路功能。
FunctionUninstallation	FunctionUninstall	從 Amazon EKS 叢集解除安裝網路功能。
HookExecution	名稱沒有變化	執行 NSD 中定義的生命週期掛接。
InfrastructureCancellation	名稱沒有變化	取消網路服務。
InfrastructureInstantation	名稱沒有變化	代表使用者佈建 AWS 資源。
InfrastructureTermination	名稱沒有變化	取消佈建透過 AWS T AWS NB 呼叫的資源。
InventoryDeregistration	名稱沒有變化	從 AWS TNB 取消註冊 AWS 資源。
KubernetesClusterConfiguration	ClusterConfiguration	設定 Kubernetes 叢集，並將其他 IAM 角色新增至 Amazon EKS AuthMap (如 NSD 中所定義)。
NetworkServiceFinalization	名稱沒有變化	完成網路服務並提供成功或失敗狀態更新。
NetworkServiceInstantiation	名稱沒有變化	初始化網路服務。

在 2024 年 3 月 7 日之前開始的部署工作名稱	2024 年 3 月 7 日以後開始的部署工作名稱	Task description (任務描述)
SelfManagedNodesConfiguration	名稱沒有變化	透過 Amazon EKS 和 Kubernetes 控制平面啟動自我管理節點。

AWS電信網絡生成器的服務配額

服務配額 (也稱為限制) 是您AWS帳戶的服務資源或作業數目上限。如需詳細資訊，請參閱《AWS》中的 Amazon Web Services 一般參考 服務配額。

以下是 AWS TNB 的服務配額。

名稱	預設	可調整	描述
同時進行的網路服務作業	每個受支援的區域：40	是	一個區域中同時進行的網路服務作業數目上限。
功能套件	每個支持地區：200	是	一個區域中功能包的最大數量。
網路套件	每個受支援的區域：40	是	一個區域中網路套件的最大數量。
網路服務實例	每個支持的地區：800	是	一個區域中的網路服務執行個體數目上限。

AWS TNB 使用者指南的文件歷史記錄

下表說明 AWS TNB 的文件發行版本。

變更	描述	日期
現有任務的新任務和新任務名稱	有新工作可用。從 2024 年 3 月 7 日起，為了清楚起見，一些現有任務具有新名稱。	2024年5月7日
叢集的庫伯內特斯版本	AWS TNB 現在支援庫伯尼特斯 1.29 版以建立 Amazon EKS 叢集。	2024年4月10日
Support 網路介面 `security_groups`	您可以將安全性群組附加至 AWS.NETWORK .eni 節點。	2024年4月2日
Support Amazon EBS 根磁碟區加密	您可以為 Amazon EBS 根磁碟區啟用 Amazon EBS 加密。若要啟用此功能，請在 AWS.Compute.EKS 或 AWS.Compute. ManagedNode eks 節點中新增屬性。SelfManagedNode	2024年4月2日
Support 節點 `labels`	您可以將節點標籤附加到 AWS.Compute.EKS 或 AWS.Compute. ManagedNode eks 節點中的節點群組。SelfManagedNode	2024年3月19 日
Support 網路介面 `source_dest_check`	您可以透過 .networking.eni 節點指示是否要啟用或停用網路介面來源/目的地檢查 AWS。	2024年1月25日
Support 具有自訂使用者資料的 Amazon EC2 執行個體	您可以透過. AWS Compute 啟動具有自訂使用者資料的	2024年1月16日

	Amazon EC2 執行個體。 UserData 節點。	
Support 安全性群組	AWS TNB 可讓您匯入安全性群組 AWS 資源。	2024 年 1 月 8 日
更新的描述 network_interfaces	當network_interfaces 屬性包含在 AWS.Compute.eks ManagedNode 或 AWS.Compute.eks SelfManagedNode 節點中時， AWS TNB 會從屬性取得與 ENI 相關的權限 (如果有的話)，或從屬性取得與 ENI 相關的權限。multus_role node_role	2023 年 12 月 18 日
Support 私人叢集	AWS TNB 現在支援私有叢集。若要指示私用叢集，請將內access容設定為PRIVATE。	2023 年 12 月 11 日
叢集的庫伯內特斯版本	AWS TNB 現在支援庫伯尼特斯 1.28 版以建立 Amazon EKS 叢集。	2023 年 12 月 11 日
AWS TNB 支持放置組	已新增AWS.Compute.EKSManagedNode 和AWS.Compute.EKSSelfManagedNode 節點定義的放置群組。	2023 年 12 月 11 日

AWS TNB 增加了對 IPv6 的支持	AWS TNB 現在支援建立具有 IPv6 基礎架構的網路執行個體。檢查節點AWS. 網路 .vPC,. 網路. 子網路,AWS.AWS InternetGateway，AWS. 網路。 SecurityGroupIngressRule，AWS. 網路。 SecurityGroupEgressRule和 AWS.Compute.E K 適用於 IPv6 組態。我們還添加AWS了節點。網絡。AWS NAT64 我們更新了 AWS 適用於 IPv6 許可的 Amazon EKS 節點群組的 AWS TNB 服務角色和 TNB 服務角色。請參閱服務角色原則範例。	2023 年 11 月 16 日
新增 AWS TNB 服務角色原則的權限	我們為 Amazon S3 的 AWS TNB 服務角色政策新增了許可，並 AWS CloudFormation 啟用基礎設施實例化。	2023 年 10 月 23 日
AWS TNB 在更多地區推出	AWS TNB 現已在亞太區域 (首爾)、加拿大 (中部)、歐洲 (西班牙)、歐洲 (斯德哥爾摩) 和南美洲 (聖保羅) 區域推出。	2023 年 9 月 27 日
對於 AWS. 計算機的標籤 SelfManagedNode	AWS TNB 現在支持`AWS.Compute.EKSSelfManagedNode` 節點定義的標籤。	2023 年 8 月 22 日
AWS TNB 支援利用 IMDSv2 的執行個體	啟動執行個體時，您必須使用 IMDSv2。	2023 年 8 月 14 日

已更新的權限 MultusRoleInlinePolicy	MultusRoleInlinePolicy 現在包括`ec2:DeleteNetworkInterface` 權限。	2023 年 8 月 7 日
叢集的庫伯內特斯版本	AWS TNB 現在支援庫伯尼特斯 1.27 版以建立 Amazon EKS 叢集。	2023 年 7 月 25 日
AWS. 計算機. AuthRole	AWS TNB 支援可 AuthRole 讓您將 IAM 角色新增至 Amazon EKS 叢集，讓使用者可`aws-authConfigMap` 以使用 IAM 角色存取 Amazon EKS 叢集。	2023 年 7 月 19 日
AWS TNB 支持安全組。	添加了 AWS. 網絡。SecurityGroup，AWS. 網路。SecurityGroupEgressRule和 AWS. 網路。 SecurityGroupIngressRule到 NSD 模板。	2023 年 7 月 18 日
叢集的庫伯內特斯版本	AWS TNB 支援 1.22 至 1.26 版的庫伯尼特斯，以建立 Amazon EKS 叢集。 AWS TNB 不再支持庫伯尼特版本 1.21。	2023 年 5 月 11 日
AWS. 計算機. SelfManagedNode	您可以在區域內、L AWS ocal Zones 和. AWS Outposts	2023 年 3 月 29 日
初始版本	這是 AWS TNB 用戶指南的第一個發行版本。	2023 年 2 月 21 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。