使用者指南

AWS 電信網路建置器

AWS 電信網路建置器: 使用者指南

Table of Contents

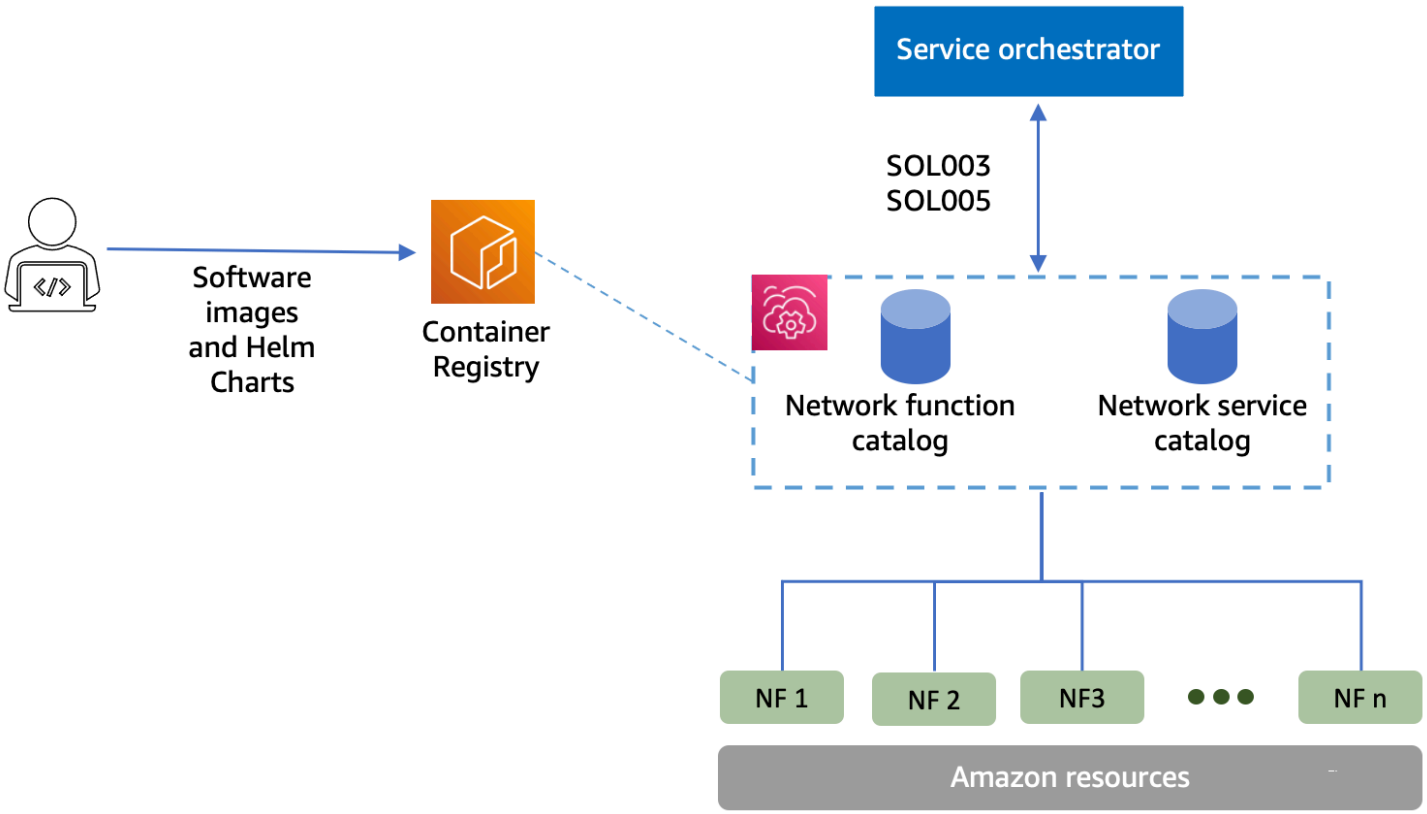
什麼是 AWS Telco Network Builder？

AWS Telco Network Builder (AWS TNB) 是一項 AWS 服務，可提供通訊服務供應商 (CSPs) 在基礎設施上 AWS 部署、管理和擴展 5G 網路的有效方式。

使用 AWS TNB，您可以自動化方式 AWS 雲端 使用網路映像在 中部署可擴展且安全的 5G 網路。您不需要學習新技術、決定要使用的運算服務，或知道如何佈建和設定 AWS 資源。

反之，您會描述網路的基礎設施，並提供獨立軟體廠商 (ISV) 合作夥伴的網路功能軟體映像。 AWS TNB 與第三方服務協調人員 AWS 和服務整合，以自動佈建必要的 AWS 基礎設施、部署容器化網路功能，以及設定聯網和存取管理，以建立完全運作的網路服務。

下圖說明 AWS TNB 和服務協調器之間的邏輯整合，以使用歐洲電信標準協會 (ETSI) 型標準介面來部署網路函數。



主題

- 初次使用 AWS？
- AWS TNB 適用於誰？
- AWS TNB 功能

- 存取 AWS TNB
- AWS TNB 定價
- 下一步

初次使用 AWS？

如果您是初次使用 AWS 產品和服務，請使用下列資源開始進一步了解：

- 簡介 AWS
- 入門 AWS

AWS TNB 適用於誰？

AWS TNB 適用於希望利用成本效益的 CSPs， 敏捷性、 和 彈性 AWS 雲端 提供，無需撰寫和維護自訂指令碼和組態，即可設計、 部署、 和管理網路服務。 AWS TNB 會自動佈建必要的 AWS 基礎設施， 部署容器化網路函數、 和 設定聯網和存取管理，以根據 CSP 定義的網路服務描述項建立完全運作的網路服務， 和 CSP 想要部署的網路函數。

AWS TNB 功能

以下是 CSP 想要使用 AWS TNB 的一些原因：

有助於簡化任務

為您的網路操作提供更高的效率，例如部署新服務、更新和升級網路功能，以及變更網路基礎設施拓撲。

與協調程式整合

AWS TNB 與 ETSI 相容的熱門第三方服務協調程式整合。

規模

您可以設定 AWS TNB 來擴展基礎 AWS 資源以滿足流量需求、更有效率地執行網路函數更新、推出網路基礎設施拓撲變更，並將新 5G 服務的部署時間從幾天縮短為幾小時。

檢查和監控 AWS 資源

AWS TNB 可讓您在單一儀表板上檢查和監控支援網路 AWS 的資源，例如 Amazon VPC、Amazon EC2 和 Amazon EKS。

支援服務範本

AWS TNB 可讓您為所有電信工作負載 (RAN、Core、IMS) 建立服務範本。您可以建立新的服務定義、重複使用現有的範本，或與持續整合和持續交付 (CI/CD) 管道整合，以發佈新的定義。

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，您可以以可重複且可擴展的方式追蹤變更。手動執行此操作需要管理網路狀態、建立和刪除資源，以及注意所需的變更順序。當您使用 AWS TNB 來管理網路函數的生命週期時，您只會對描述網路函數的網路服務描述項進行變更。然後， AWS TNB 會自動以正確的順序進行必要的變更。

簡化網路函數生命週期

您可以管理網路函數的第一個和所有後續版本，並指定升級的時間。您也可以以相同方式管理您的 RAN、Core、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 AWS TNB 資源：

- AWS TNB 主控台 — 提供用於管理網路的 Web 界面。
- AWS TNB API — 提供執行 AWS TNB 動作的 RESTful API。如需詳細資訊，請參閱 AWS TNB API 參考
- AWS Command Line Interface (AWS CLI) — 為廣泛的 AWS 服務提供命令，包括 AWS TNB。Windows、macOS 和 Linux 支援此功能。如需詳細資訊，請參閱《AWS Command Line Interface 使用者指南》https://docs.aws.amazon.com/cli/latest/userguide/。
- AWS SDKs – 提供語言特定的 APIs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。

AWS TNB 定價

AWS TNB 可協助 CSPs 上自動化電信網路的部署和管理 AWS。使用 AWS TNB 時，您需要支付以下兩個維度的費用：

- 依受管網路函數項目 (MNFI) 時數。
- 依 API 請求數。

當您使用其他 AWS 服務搭配 AWS TNB 時，也會產生額外費用。如需詳細資訊，請參閱 AWS TNB 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。

如果您對 AWS 帳單、帳戶和事件有任何疑問，請聯絡 AWS Support。

AWS Trusted Advisor 是一項服務，可用來協助最佳化 AWS 環境的成本、安全性和效能。如需詳細資訊，請參閱 AWS Trusted Advisor。

下一步

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。
- 入門 AWS TNB – 部署您的第一個網路函數，例如集中式單元 (CU)、存取和行動性管理函數 (AMF)、使用者平面函數 (UPF) 或完整的 5G 核心。

運作方式 AWS TNB

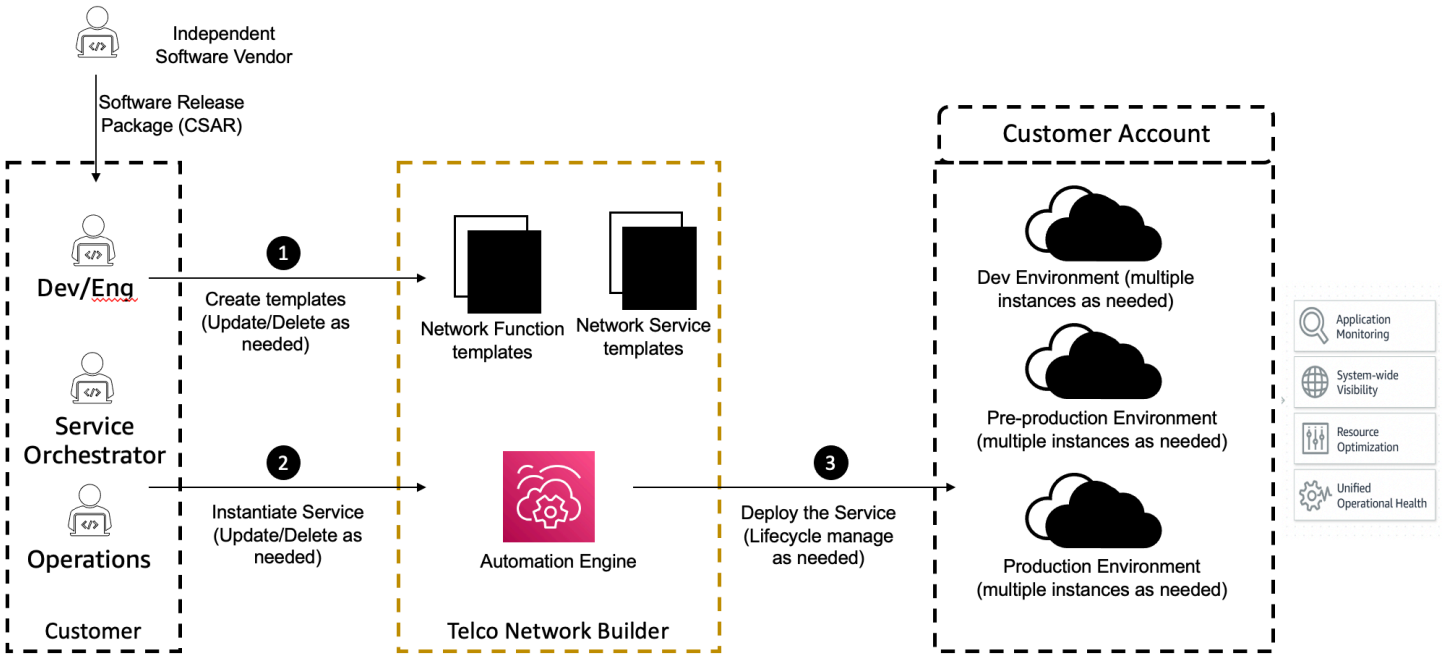
AWS TNB 與標準化 end-to-end 協調程式 AWS 和資源整合，以操作完整的 5G 網路。

AWS TNB 可讓您擷取網路函數套件和網路服務描述符（NSDs），並提供自動化引擎來操作網路。您可以使用協調 end-to-end器並與 整合 AWS TNBAPIs，或使用 AWS TNB SDKs 建置自己的自動化流程。如需詳細資訊，請參閱AWS TNB 架構 。

主題

AWS TNB 架構

AWS TNB 可讓您透過 AWS Management Console、 AWS TNBAPI、 AWS CLI REST 和 執行生命週期管理操作SDKs。這可讓工程、操作和程式設計系統團隊成員等不同的CSP角色利用 AWS TNB。您建立並上傳網路函數套件做為 Cloud Service Archive（CSAR）檔案。CSAR 檔案包含 Helm Chart、軟體映像和 Network Function Descriptor（NFD）。您可以使用 範本重複部署該套件的多個組態。您可以建立網路服務範本，定義要部署的基礎設施和網路函數。您可以使用參數覆寫在不同位置部署不同的組態。然後，您可以使用 範本來實例化網路，並在 AWS 基礎設施上部署網路功能。 AWS TNB 為您提供部署的可見性。



與 整合 AWS 服務

5G 網路由一組互連的容器化網路函數組成，這些函數部署在數千個 Kubernetes 叢集上。 AWS TNB 整合了下列 AWS 服務 電信特定功能APIs，以建立完全運作的網路服務：

- Amazon Elastic Container Registry （ Amazon ECR ） 儲存獨立軟體供應商 （ ISVs ） 網路函數成品。
- Amazon Elastic Kubernetes Service （ Amazon EKS ） 來設定叢集。
- Amazon VPC for networking constructs。
- 使用 的安全群組 AWS CloudFormation。
- AWS CodePipeline 適用於跨 AWS 區域、 AWS 本機區域和 的部署目標 AWS Outposts。
- IAM 來定義角色。
- AWS Organizations 以控制對 的 AWS TNB存取APIs。
- AWS Health Dashboard 和 AWS CloudTrail 來監控運作狀態和張貼指標。

AWS TNB 資源配額

您的 AWS 帳戶 具有每個 的預設配額，先前稱為限制 AWS 服務。除非另有說明，否則每個配額都是特有的 AWS 區域。您可以要求提高某些配額，但並非所有配額都能提高。

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

若要請求提高配額，請參閱《Service Quotas 使用者指南》中的請求提高配額。

您的 AWS 帳戶 具有與 相關的下列配額 AWS TNB。

資源配額	描述	預設值	是否可調整？
網路服務執行個體	一個區域中的網路服務執行個體數量上限。	800	是
並行的持續網路服務操作	一個區域中並行進行中網路服務操作的最大數量。	40	是
網路套件	一個區域中的網路套件數量上限。	40	是

資源配額	描述	預設值	是否可調整？
函數套件	一個區域中的函數套件數目上限。	200	是

AWS TNB 概念

本主題說明協助您開始使用 的基本概念 AWS TNB。

目錄

網路函數的生命週期

AWS TNB 在網路函數的整個生命週期中都協助您。網路函數生命週期包括下列階段和活動：

規劃

1. 透過識別要部署的網路函數來規劃您的網路。
2. 將網路函數軟體映像放入容器映像儲存庫。
3. 建立要部署或升級的CSAR套件。
4. 使用 AWS TNB 上傳定義網路函數的CSAR套件 （ 例如 CU AMF和 UPF ），並與連續整合和連續交付 （ CI/CD ） 管道整合，以協助您建立新的CSAR套件版本，作為新的網路函數軟體映像或客戶指令碼可用。

組態

1. 識別部署所需的資訊，例如運算類型、網路函數版本、IP 資訊和資源名稱。
2. 使用 資訊建立您的網路服務描述符 （ NSD ）。
3. NSDs 定義網路函數的擷取，以及網路函數具實例化所需的資源。

即時化

1. 建立網路函數所需的基礎設施。
2. 依其定義，即時化 （ 或佈建 ） 網路函數，NSD並開始承載流量。
3. 驗證資產。

生產

在網路函數的生命週期內，您將完成生產操作，例如：

- 更新網路函數組態，例如，更新已部署網路函數中的值。
- 使用新的網路套件和參數值更新網路執行個體。例如，更新網路套件中的 Amazon EKS `version` 參數。

使用標準化介面

AWS TNB 與歐洲電信標準協會（ETSI）相容服務協調器整合，可讓您簡化網路服務的部署。服務協調人員可以使用 AWS TNB SDKs、 CLI或 APIs來啟動操作，例如初始化或將網路函數升級至新版本。

AWS TNB 支援下列規格。

規格	發行版本	描述
ETSI SOL001	v3.6.1	定義允許 TOSCA型網路函數描述符的標準。
ETSI SOL002	v3.6.1	定義網路函數管理的模型。
ETSI SOL003	v3.6.1	定義網路函數生命週期管理的標準。
ETSI SOL004	v3.6.1	定義網路函數套件CSAR的標準。
ETSI SOL005	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。
ETSI SOL007	v3.5.1	定義允許 TOSCA型網路服務描述符的標準。

的網路函數套件 AWS TNB

使用 AWS TNB，您可以將符合 ETSI SOL001/SOL004 的網路函數套件存放在函數目錄中。然後，您可以上傳包含描述網路函數成品的 Cloud Service Archive（CSAR）套件。

- 網路函數描述詞 – 定義套件加入和網路函數管理的中繼資料
- 軟體映像 – 參考網路函數容器映像。Amazon Elastic Container Registry（Amazon ECR）可以充當網路函數映像儲存庫。
- 其他檔案 – 用於管理網路函數，例如指令碼和 Helm Chart。

CSAR 是由 OASISTOSCA標準定義的套件，包含符合 OASISTOSCAYAML規格的網路/服務描述詞。如需所需YAML規格的資訊，請參閱 TOSCA 的參考 AWS TNB。

以下是範例網路函數描述符。

```
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"
```

的網路服務描述符 AWS TNB

AWS TNB 會將網路服務描述符 （NSDs） 存放在您要部署的網路函數，以及您要如何將它們部署到目錄中。您可以上傳YAMLNSD檔案 （`vnfd.yaml`），如 ETSI SOL007 所述，包含下列資訊：

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

AWS TNB 支援以TOSCA語言建立資源模型ETSI的標準，例如網路、服務和函數。 AWS TNB AWS 服務 可讓您以 ETSI合規服務協調者可以理解的方式建立資源模型，進而更有效率地使用它們。

以下是 的程式碼片段，NSD示範如何建立 模型 AWS 服務。網路函數將部署在具有 Kubernetes 1.27 版的 Amazon EKS叢集上。應用程式子網路為 Subnet01 和 Subnet02。然後，您可以使用 Amazon Machine Image （ AMI ）、執行個體類型和自動擴展組態 NodeGroups 來定義應用程式的 。

```
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
```

的管理和操作 AWS TNB

透過 AWS TNB，您可以使用標準化管理操作來管理網路，方法為 SOL003 ETSI 和 SOL005。您可以使用 AWS TNBAPIs來執行生命週期操作，例如：

- 讓您的網路函數變得具現化。
- 終止您的網路函數。
- 更新您的網路函數以覆寫 Helm 部署。
- 使用新的網路套件和參數值更新具現化或更新的網路執行個體。
- 管理網路函數套件的版本。
- 管理 的版本NSDs。
- 擷取已部署網路函數的相關資訊。

的網路服務描述符 AWS TNB

網路服務描述詞 （NSD） 是網路套件中的.yaml檔案，使用 TOSCA標準來描述您要部署的網路函數，以及 AWS 您要部署網路函數的基礎設施。若要定義您的 NSD並設定基礎資源和網路生命週期操作，您必須了解 支援的NSDTOSCA結構描述 AWS TNB。

您的NSD檔案分為下列部分：

1. TOSCA 定義版本 – 這是NSDYAML檔案的第一行，包含版本資訊，如下列範例所示。

   ```
   tosca_definitions_version: tnb_simple_yaml_1_0
   ```

2. VNFD – NSD包含要在其中執行生命週期操作的網路函數定義。每個網路函數都必須以下列值識別：
   - 的唯一 IDdescriptor_id。ID 必須與網路函數CSAR套件中的 ID 相符。
   - 的唯一名稱namespace。名稱必須與唯一 ID 相關聯，以便更輕鬆地參考整個NSDYAML檔案，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、網路函數部署，以及任何自訂指令碼，例如生命週期掛鉤。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 – 每個建模資源都有屬性和需求的區段。屬性描述資源的選用或必要屬性，例如 版本。這些要求描述必須作為引數提供的相依性。例如，若要建立 Amazon EKS Node Group Resource，必須在 Amazon EKS Cluster 中建立。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

完成本主題中所述的任務來設定 AWS TNB。

任務

- 註冊 AWS 帳戶
- 建立具有管理存取權的使用者
- 選擇 AWS 區域
- 請注意服務端點
- （選用）安裝 AWS CLI
- 設定 AWS TNB 角色

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個 。

註冊 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 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 會在註冊程序完成後傳送確認電子郵件給您。您可以隨時登錄 https://aws.amazon.com/ 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理存取權的使用者

註冊 後 AWS 帳戶，請保護 AWS 帳戶根使用者、啟用 AWS IAM Identity Center和建立管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者AWS Management Console身分登入 。在下一頁中，輸入您的密碼。

   如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

   如需說明，請參閱《IAM 使用者指南》中的為您的 AWS 帳戶 根使用者 （主控台） 啟用虛擬 MFA 裝置。

建立具有管理存取權的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的啟用 AWS IAM Identity Center。

2. 在 IAM Identity Center 中，將管理存取權授予使用者。

   如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center《 使用者指南》中的使用預設值設定使用者存取權 IAM Identity Center 目錄。

以具有管理存取權的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

  如需使用 IAM Identity Center 使用者登入的說明，請參閱AWS 登入 《 使用者指南》中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的建立許可集。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的新增群組。

選擇 AWS 區域

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

請注意服務端點

若要以程式設計方式連線至 AWS 服務，請使用 端點。除了標準 AWS 端點之外，某些 AWS 服務還在所選區域中提供 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	

區域名稱	區域	端點	通訊協定	
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	

（選用）安裝 AWS CLI

AWS Command Line Interface (AWS CLI) 為廣泛的 AWS 產品提供命令，並在 Windows、macOS 和 Linux 上支援 。您可以使用 存取 AWS TNB AWS CLI。若要開始使用，請參閱《AWS Command Line Interface 使用者指南》。如需 AWS TNB 命令的詳細資訊，請參閱 AWS CLI 命令參考中的 tnb。

設定 AWS TNB 角色

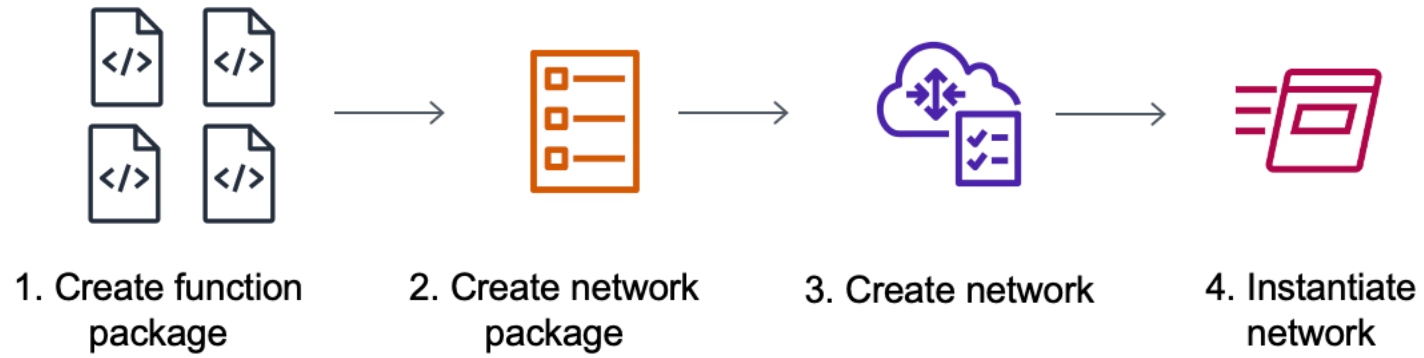
您必須建立 IAM 服務角色來管理 AWS TNB 解決方案的不同部分。 AWS TNB 服務角色可以代您對其他 AWS 服務 AWS CloudFormation AWS CodeBuild，例如 和各種運算和儲存服務進行 API 呼叫，以執行個體化和管理部署的資源。

如需 AWS TNB 服務角色的詳細資訊，請參閱 AWS TNB 的身分和存取管理。

入門 AWS TNB

本教學課程示範如何使用 AWS TNB 來部署網路函數，例如集中式單位 （CU）、存取與行動管理函數（AMF） 或 5G 使用者平面函數 （ ）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
- 建立和實例化網路執行個體
- 清除

必要條件

在執行成功部署之前，您必須具備下列項目：

- AWS 商業支援計劃。
- 透過IAM角色的許可。
- 符合 SOL001/0SOL04 的網路函數 （NF） 套件。 ETSI
- 符合 SOL007 的網路服務描述符 （NSD） 範本。 ETSI

您可以從網站的範例套件中使用範例函數套件或網路套件 AWS TNB GitHub 。

建立函數套件

網路函數套件是 Cloud Service Archive （CSAR） 檔案。CSAR 檔案包含 Helm Chart、軟體映像和 Network Function Descriptor （NFD）。

若要建立函數套件

1. 在 開啟 AWS TNB主控台https://console.aws.amazon.com/tnb/。
2. 在導覽窗格中，選擇函數套件 。
3. 選擇建立函數套件 。
4. 在上傳函數套件 下，選擇選擇檔案 ，然後將每個CSAR套件上傳為.zip檔案。您最多可以上傳 10 個檔案。
5. （選用） 在標籤 下，選擇新增標籤並輸入索引鍵和值。您可以使用標籤來搜尋和篩選資源或追蹤 AWS 成本。
6. 選擇 Next (下一步)。
7. 檢閱套件詳細資訊，然後選擇建立函數套件 。

建立網路套件

網路套件會指定您要部署的網路函數，以及您要如何將函數部署到目錄中。

若要建立網路套件

1. 在導覽窗格中，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 ，然後將每個檔案NSD上傳為.zip檔案。您最多可以上傳 10 個檔案。
4. （選用） 在標籤 下，選擇新增標籤並輸入索引鍵和值。您可以使用標籤來搜尋和篩選資源或追蹤 AWS 成本。
5. 選擇 Next (下一步)。
6. 選擇建立網路套件 。

建立和實例化網路執行個體

網路執行個體是在 中 AWS TNB建立的單一網路，可以部署。您必須建立網路執行個體並加以實例化。當您實例化網路執行個體時， 會 AWS TNB佈建必要的 AWS 基礎設施、部署容器化網路函數，以及設定聯網和存取管理，以建立完全運作的網路服務。

建立和實例化網路執行個體

1. 在導覽窗格中，選擇 Networks 。
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. 選擇網路套件。驗證詳細資訊，然後選擇下一步。
5. 選擇建立網路執行個體 。初始狀態為 `Created`。

   Networks 頁面會顯示 `Not instantiated` 狀態的新網路執行個體。
6. 選取網路執行個體，選擇動作 和 Instantiate 。

   網路實例化頁面隨即出現。
7. 檢閱詳細資訊並更新參數值。參數值的更新僅適用於此網路執行個體。NSD 和 VNFD套件中的參數不會變更。
8. 選擇 Instantiate 網路 。

   隨即顯示部署狀態頁面。
9. 使用重新整理圖示來追蹤網路執行個體的部署狀態。您也可以在部署任務區段中啟用自動重新整理，以追蹤每個任務的進度。

清除

您現在可以刪除為本教學課程建立的資源。

清除您的資源

1. 在導覽窗格中，選擇 Networks 。
2. 選擇網路的 ID，然後選擇終止 。
3. 出現確認提示時，輸入網路 ID，然後選擇終止 。
4. 使用重新整理圖示來追蹤網路執行個體的狀態。

5. （選用）選取網路，然後選擇刪除。

AWS TNB 的函數套件

函數套件是 CSAR (Cloud Service Archive) 格式的 .zip 檔案，其中包含網路函數 (ETSI 標準電信應用程式） 和函數套件描述詞，其使用 TOSCA 標準來描述網路函數應如何在您的網路上執行。

任務

- 在 AWS TNB 中建立函數套件
- 在 AWS TNB 中檢視函數套件
- 從 AWS TNB 下載函數套件
- 從 AWS TNB 刪除函數套件

在 AWS TNB 中建立函數套件

了解如何在 AWS TNB 網路函數目錄中建立函數套件。在 AWS TNB 中建立網路的第一步是建立函數套件。上傳函數套件之後，您可以建立網路套件。

Console

使用主控台建立函數套件

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇函數套件。
3. 選擇建立函數套件。
4. 選擇選擇檔案，並將每個 CSAR 套件上傳為 `.zip` 檔案。您最多可以上傳 10 個檔案。
5. 選擇 Next (下一步)。
6. 檢閱套件詳細資訊。
7. 選擇建立函數套件。

AWS CLI

使用 建立函數套件 AWS CLI

1. 使用 create-sol-function-package 命令來建立新的函數套件：

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

2. 使用 put-sol-function-package-content 命令上傳函數套件內容。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 中檢視函數套件

了解如何檢視函數套件的內容。

Console

使用主控台檢視函數套件

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇函數套件。
3. 使用搜尋方塊尋找函數套件

AWS CLI

使用 檢視函數套件 AWS CLI

1. 使用 list-sol-function-packages 命令來列出函數套件。

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

2. 使用 get-sol-function-package 命令來檢視函數套件的詳細資訊。

```
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 AWS TNB 下載函數套件

了解如何從 AWS TNB 網路函數目錄下載函數套件。

Console

使用主控台下載函數套件

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在主控台左側的導覽窗格中，選擇函數套件。
3. 使用搜尋方塊尋找函數套件
4. 選擇函數套件
5. 選擇動作、下載。

AWS CLI

使用 下載函數套件 AWS CLI

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

```
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 刪除函數套件

了解如何從 AWS TNB 網路函數目錄中刪除函數套件。若要刪除函數套件，套件必須處於停用狀態。

Console

使用主控台刪除函數套件

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇函數套件。
3. 使用搜尋方塊尋找函數套件。

4. 選擇函數套件。
5. 選擇 Actions (動作)、Disable (停用)。
6. 選擇 動作、刪除。

AWS CLI

使用 刪除函數套件 AWS CLI

1. 使用 update-sol-function-package 命令來停用函數套件。

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

2. 使用 delete-sol-function-package 命令來刪除函數套件。

```
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
```

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，可定義您要部署的函數套件，以及您要部署它們的 AWS 基礎設施。

任務

- 在 AWS TNB 中建立網路套件
- 在 AWS TNB 中檢視網路套件
- 從 AWS TNB 下載網路套件
- 從 AWS TNB 刪除網路套件

在 AWS TNB 中建立網路套件

網路套件包含網路服務描述項 (NSD) 檔案 （必要） 和任何其他檔案 （選用），例如針對您需求的指令碼。例如，如果您的網路套件中有多個函數套件，您可以使用 NSD 來定義哪些網路函數應該在某些 VPCs、子網路或 Amazon EKS 叢集中執行。

在建立函數套件之後建立網路套件。建立網路套件後，您需要建立網路執行個體。

Console

使用主控台建立網路套件

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇網路套件。
3. 選擇建立網路套件。
4. 選擇選擇檔案，並將每個 NSD 上傳為`.zip`檔案。您最多可以上傳 10 個檔案。
5. 選擇 Next (下一步)。
6. 檢閱套件詳細資訊。
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 命令來建立網路套件。

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

2. 使用 put-sol-network-package-content 命令上傳網路套件內容。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 中檢視網路套件

了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 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇網路套件。
3. 使用搜尋方塊尋找網路套件。

AWS CLI

使用 檢視網路套件 AWS CLI

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

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

2. 使用 get-sol-network-package 命令來檢視網路套件的詳細資訊。

```
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

了解如何從 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 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，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 命令下載網路套件。

```
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

了解如何從 AWS TNB 網路服務目錄中刪除網路套件。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 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，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 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 命令來停用網路套件。

```
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 命令來刪除網路套件。

```
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
```

AWS TNB 的網路執行個體

網路執行個體是在 AWS TNB 中建立的單一網路，可以部署。

任務

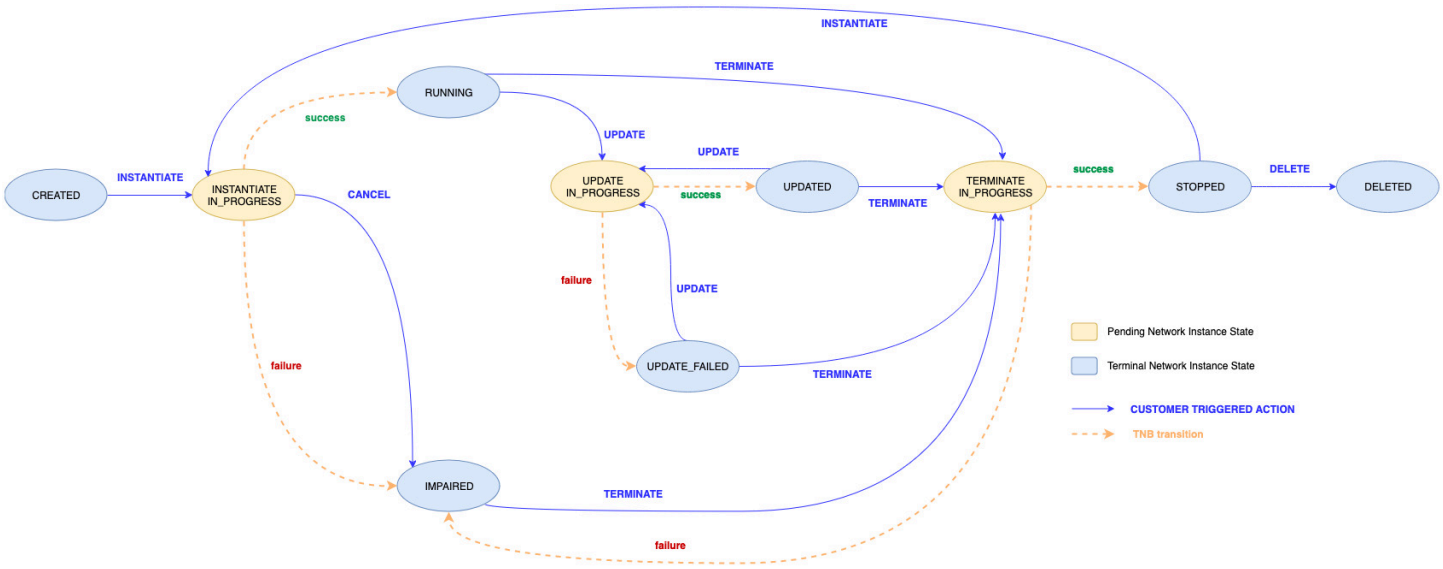
- 網路執行個體的生命週期操作
- 使用 AWS TNB 建立網路執行個體
- 使用 AWS TNB 實例化網路執行個體
- 在 AWS TNB 中更新函數執行個體
- 在 AWS TNB 中更新網路執行個體
- 在 AWS TNB 中檢視網路執行個體
- 從 AWS TNB 終止和刪除網路執行個體

網路執行個體的生命週期操作

AWS TNB 可讓您使用與 ETSI SOL003 和 SOL005 整合的標準化管理操作，輕鬆管理網路。您可以執行下列生命週期操作：

- 建立網路
- 執行個體化網路
- 更新網路函數
- 更新網路執行個體
- 檢視網路詳細資訊和狀態
- 終止網路

下圖顯示網路管理操作：



使用 AWS TNB 建立網路執行個體

您在建立網路套件後建立網路執行個體。建立網路執行個體之後，請將其執行個體化。

Console

使用主控台建立網路執行個體

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
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描述，然後選擇下一步。
5. 選取網路套件，驗證詳細資訊，然後選擇下一步。
6. 選擇建立網路執行個體。

新的網路執行個體會出現在 Networks 頁面上。接著，執行個體化此網路執行個體。

AWS CLI

使用 建立網路執行個體 AWS CLI

- 使用 create-sol-network-instance 命令來建立網路執行個體。

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

接著，執行個體化此網路執行個體。

使用 AWS TNB 實例化網路執行個體

建立網路執行個體之後，您必須將其執行個體化。當您執行個體化網路執行個體時， AWS TNB 會佈建必要的 AWS 基礎設施、部署容器化網路函數，以及設定聯網和存取管理，以建立完全運作的網路服務。

Console

使用主控台執行個體化網路執行個體

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇網路。
3. 選取您要執行個體化的網路執行個體。
4. 選擇動作，然後選擇 Instantiate。
5. 在 Instantiate 網路頁面上，檢閱詳細資訊，並選擇性地更新參數值。

   參數值的更新僅適用於此網路執行個體。NSD 和 VNFD 套件中的參數不會變更。
6. 選擇 Instantiate 網路。

   部署狀態頁面隨即出現。
7. 使用重新整理圖示來追蹤網路執行個體的部署狀態。您也可以在部署任務區段中啟用自動重新整理，以追蹤每個任務的進度。

   當部署狀態變更為 時`Completed`，網路執行個體會執行個體化。

AWS CLI

使用 執行個體化網路執行個體 AWS CLI

1. 使用 instantiate-sol-network-instance 命令來執行個體化網路執行個體。

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

2. 接著，檢視網路操作狀態。

在 AWS TNB 中更新函數執行個體

執行個體化網路執行個體後，您可以在網路執行個體中更新函數套件。

Console

使用主控台更新函數執行個體

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇網路。
3. 選取網路執行個體。只有在網路執行個體的狀態為 時，您才能更新網路執行個體Instantiated。

   網路執行個體頁面隨即出現。
4. 從函數索引標籤中，選取要更新的函數執行個體。
5. 選擇更新。
6. 輸入您的更新覆寫。
7. 選擇更新。

AWS CLI

使用 CLI 更新函數執行個體

使用 update-sol-network-instance 命令搭配MODIFY_VNF_INFORMATION更新類型來更新網路執行個體中的函數執行個體。

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

在 AWS TNB 中更新網路執行個體

執行個體化網路執行個體之後，您可能需要更新基礎設施或應用程式。若要這麼做，請更新網路執行個體的網路套件和參數值，並部署更新操作以套用變更。

考量事項

- 您可以更新處於 Instantiated或 Updated 狀態的網路執行個體。
- 當您更新網路執行個體時，UpdateSolNetworkServiceAPI 會使用新的網路套件和參數值來更新網路執行個體的拓撲。
- AWS TNB 會驗證網路執行個體中的 NSD 和 VNFD 參數數量不超過 200。強制執行此限制，以防止不良行為者傳遞影響服務的錯誤或巨型承載。

您可以更新的參數

您可以在更新執行個體化網路執行個體時更新下列參數：

參數	描述	範例：之前	範例：之後

Amazon EKS 叢集版本	您可以將 Amazon EKS 叢集控制平面version參數的值更新為下一個次要版本。您無法降級版本。工作者節點不會更新。	`EKSCluster:`
		`  type: tosca.nodes.AWS.Compute.EKS`
		`  properties:`
		`  version: "1.28"`
			`EKSC`
			`r:`
			`typ`
			`tos`
			`es.A`
			`mput`

參數	描述	範例：之前	範例：之後
			pro s: ver "1.

參數	描述	範例：之前	範例：之後

擴展屬性	您可以更新 EKSManagedNode 和 EKSSelfManagedNode TOSCA 節點的擴展屬性。	`EKSNodeGroup01:`
		`  ...`
		`  scaling:`
		`  properties:`
		`  desired_size: 1`
		`  min_size: 1`
		`  max_size: 1`
			`EKSN`
			`oup0`
			`...`
			`sca`
			`pro`
			`s:`
			`des`
			`ize:`

參數	描述	範例：之前	範例：之後
			mir max

參數	描述	範例：之前	範例：之後

Amazon EBS CSI 外掛程式屬性	您可以在 Amazon EKS 叢集上啟用或停用 Amazon EBS CSI 外掛程式。您也可以變更外掛程式版本。	`EKSCluster:`
		`  capabilities:`
		`  ...`
		`  ebs_csi:`
		`  properties:`
		`  enabled: false`
			`EKSC`
			`r:`
			`cap`
			`ies:`
			`...`
			`ebs`
			`pro`
			`s:`
			`ena`
			`ver`
			`"v1`
			`e`

參數	描述	範例：之前	範例：之後
			ksbu "

參數	描述	範例：之前	範例：之後
VNF	您可以參考 NSD 中的 VNFs VNFDeployment TOSCA 節點將其部署到 NSD 中建立的叢集。作為更新的一部分，您將能夠新增、更新和刪除網路的 VNFs。	vnfds: - descriptor_id: "*43c012fa-2616-41a8-a833-0dfd4c5a049e* " namespace: " *vnf1*" - descriptor_id: "*64222f98-ecd6-4871-bf94-7354b53f3ee5* " namespace: "*vnf2*" *// Deleted VNF* ... SampleVNF1HelmDeploy: type: tosca.nodes.AWS.Deployment.VNFDeployment requirements: cluster: EKSCluster vnfs: - *vnf1.SampleVNF1* - *vnf2.SampleVNF2*	vnfd - des r_id "59 79e9 - be53 2ad0 " nam : "vr Upd VNF - des r_id "b7 839d -916 a166 " nam : "vr Add VNF ....

參數	描述	範例：之前	範例：之後
			Sa mple elmD : typ tos es.A ploy VNFD ment rec nts: clu EKS r

參數	描述	範例：之前	範例：之後
			vnf - v leVN - v leVN

參數	描述	範例：之前	範例：之後

勾點	若要在建立網路函數之前和之後執行生命週期操作，請將 pre_create 和 post_create 掛鉤新增至VNFDeployment 節點。
	
	在此範例中，勾PreCreateHook 點會在 vnf3.SampleVNF3 執行個體化之前執行，而勾PostCreateHook 點會在 vnf3.SampleVNF3 執行個體化之後執行。
		vnfds:
		  - descriptor_id: "*43c012fa-2616-41a8-a833-0dfd4c5a049e* "
		  namespace: " *vnf1*"
		  - descriptor_id: "*64222f98-ecd6-4871-bf94-7354b53f3ee5* "
		  namespace: " *vnf2*"
		...
		SampleVNF1HelmDeploy:
		  type: tosca.nodes.AWS.Deployment.VNFDeployment
		  requirements:
		  cluster: EKSCluster
		  vnfs:
		  - *vnf1.SampleVNF1*
		  - *vnf2.SampleVNF2 // Removed during update*
			vnfd
			
			- des
			r_id
			"43
			2616
			-
			a833
			d4c5
			"
			
			nam
			:
			"vr
			
			-
			des
			r_id
			"b7
			839d
			-916
			a166
			"
			
			nam
			:
			"vr
			....
			S
			ampl
			Helm
			y:

參數	描述	範例：之前	範例：之後
			typ tos es.A ploy VNFD ment req nts: clu EKS r vnf - v leVN No cha to thi fur as the nam and uui rem

參數	描述	範例：之前	範例：之後
			the sam - v leVN New VNF as the nam , vnf was not pre y pre int s: Hoo pos te: eHoo

參數	描述	範例：之前	範例：之後
			pre e: *Hook*

參數	描述	範例：之前	範例：之後
勾點	若要在更新網路函數之前和之後執行生命週期操作，您可以將pre_update 勾點和post_update 勾點新增至VNFDeployment 節點。 在此範例中， PreUpdate Hook 將在更新 vnf1.SampleVNF1 之前執行，PostUpdateHook 並在更新 之後執行vnf1.SampleVNF1 ，此vnf套件由 uuid 為命名空間 vnf1 更新的 所指示。	vnfds: - descriptor_id: "*43c012fa-2616-41a8-a833-0dfd4c5a049e* " namespace: " *vnf1*" - descriptor_id: "*64222f98-ecd6-4871-bf94-7354b53f3ee5* " namespace: " *vnf2*" ... SampleVNF1HelmDeploy: type: tosca.nodes.AWS.Deployment.VNFDeployment requirements: cluster: EKSCluster vnfs: - *vnf1.SampleVNF1* - *vnf2.SampleVNF2*	vnfd - des r_id "0e bd87 - b8a1 4666 " nam : "vn - des r_id "64 ecd6 - bf94 4b53 " nam : "vn ... S ampl

參數	描述	範例：之前	範例：之後
			Helmy: typ tos es.A ploy VNFD ment req nts: clu EKS r vnf - v leVN A VNF upd as the uui cha for

參數	描述	範例：之前	範例：之後
			nam "vr - v leVN No cha to thi fur as nam and uui rem the sam int s: Hoo pre e: Hook

參數	描述	範例：之前	範例：之後
			pos te: eHoc

更新網路執行個體

Console

使用主控台更新網路執行個體

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇網路。
3. 選取網路執行個體。只有在網路執行個體的狀態為 Instantiated或 時，您才能更新網路執行個體Updated。
4. 選擇動作和更新。

   更新執行個體頁面會顯示網路詳細資訊，以及目前基礎設施中的參數清單。
5. 選擇新的網路套件。

   新網路套件中的參數會出現在更新後的參數區段中。
6. 或者，更新更新參數區段中的參數值。如需您可以更新的參數值清單，請參閱 您可以更新的參數。
7. 選擇更新網路。

   AWS TNB 會驗證請求並啟動部署。隨即出現部署狀態頁面。
8. 使用重新整理圖示來追蹤網路執行個體的部署狀態。您也可以在部署任務區段中啟用自動重新整理，以追蹤每個任務的進度。

   當部署狀態變更為 時Completed，會更新網路執行個體。

9.
   - 如果驗證失敗，網路執行個體會保持與請求更新之前相同的狀態 - Instantiated或Updated。
   - 如果更新失敗，網路執行個體狀態會顯示 Update failed。選擇每個失敗任務的連結，以判斷原因。
   - 如果更新成功，網路執行個體狀態會顯示 Updated。

AWS CLI

使用 CLI 更新網路執行個體

使用 update-sol-network-instance 命令搭配UPDATE_NS更新類型來更新網路執行個體。

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

在 AWS TNB 中檢視網路執行個體

了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 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，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 list-sol-network-instances 命令來列出您的網路執行個體。

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

2. 使用 get-sol-network-instance 命令來檢視特定網路執行個體的詳細資訊。

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

從 AWS TNB 終止和刪除網路執行個體

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 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇網路。
3. 選取網路執行個體的 ID。
4. 選擇終止。
5. 出現確認提示時，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. （選用）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 terminate-sol-network-instance 命令來終止網路執行個體。

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

2. （選用）使用 delete-sol-network-instance 命令來刪除網路執行個體。

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

AWS TNB 的網路操作

網路操作是對網路執行的任何操作，例如網路執行個體執行個體實例化或終止。

任務

- 檢視 AWS TNB 網路操作
- 取消 AWS TNB 網路操作

檢視 AWS TNB 網路操作

檢視網路操作的詳細資訊，包括網路操作中涉及的任務，以及任務的狀態。

Console

使用主控台檢視網路操作

1. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇網路執行個體。
3. 使用搜尋方塊尋找網路執行個體。
4. 在部署索引標籤上，選擇網路操作。

AWS CLI

使用 檢視網路操作 AWS CLI

1. 使用 list-sol-network-operations 命令列出所有網路操作。

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

2. 使用 get-sol-network-operation 命令來檢視網路操作的詳細資訊。

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

取消 AWS TNB 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. 在 https：//https://console.aws.amazon.com/tnb/ 開啟 AWS TNB 主控台。
2. 在導覽窗格中，選擇網路。
3. 選取網路的 ID 以開啟其詳細資訊頁面。
4. 在部署索引標籤上，選擇網路操作。
5. 選擇取消操作。

AWS CLI

使用 取消網路操作 AWS CLI

使用 cancel-sol-network-operation 命令取消網路操作。

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

TOSCA 的參考 AWS TNB

雲端應用程式的拓撲和協調規格（TOSCA）是一種宣告性語法，CSPs用於描述雲端型 Web 服務、其元件、關係和管理它們的程序的拓撲。CSPs 描述連線點、連線點之間的邏輯連結，以及TOSCA範本中的親和性和安全性等政策。CSPs 然後上傳範本，將跨 AWS 可用區域建立正常運作 5G 網路所需的資源合成到 AWS TNB該範本。

目錄

VNFD 範本

定義虛擬網路函數描述符（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`

TOSCA AWS 節點。可能的節點包括：

- AWS.VNF

- AWS.Artifacts.Helm

AWS.VNF

定義 AWS 虛擬網路函數（ 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`

描述符UUID的 。

必要：是

類型：字串

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

`descriptor_version`

的版本VNFD。

必要：是

類型：字串

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

`descriptor_name`

描述符的名稱。

必要：是

類型：字串

`provider`

的作者VNFD。

必要：是

類型：字串

要求

`helm`

定義容器成品的 Helm 目錄。這是 AWS.Artifacts.Helm 的參考。

必要：是

類型：字串

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 AWS Helm Node。

語法

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

```
    implementation: String
```

屬性

`implementation`

包含CSAR套件內 Helm Chart 的本機目錄。

必要：是

類型：字串

範例

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

網路服務描述符範本

定義網路服務描述符（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，您可以使用{ get_input: *input-parameter-name* }語法並在NSD範本中定義參數。然後重複使用相同NSD範本中的 參數。

下列範例示範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 }
```

VNFD 匯入

`descriptor_id`

描述符UUID的 。

必要：是

類型：字串

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

`namespace`

唯一名稱。

必要：是

類型：字串

拓撲範本

`node_templates`

可能的TOSCA AWS 節點包括：

- AWS.NS
- AWS.Compute。EKS
- AWS.ComputeEKS.AuthRole
- AWS.Compute。EKSManagedNode
- AWS.Compute。EKSSelfManagedNode
- AWS.Compute。PlacementGroup
- AWS.Compute。UserData
- AWS.Networking。SecurityGroup
- AWS.Networking。SecurityGroupEgressRule
- AWS.Networking。SecurityGroupIngressRule
- AWS.Resource.Import
- AWS.Networking。ENI
- AWS.HookExecution
- AWS.Networking。InternetGateway
- AWS.Networking。RouteTable
- AWS.Networking.Subnet
- AWS.部署。VNFDeployment
- AWS.Networking。VPC
- AWS.Networking。NATGateway

- AWS.Networking.Route

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`

描述符UUID的 。

必要：是

類型：字串

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

`descriptor_version`

的版本NSD。

必要：是

類型：字串

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

`descriptor_name`

描述符的名稱。

必要：是

類型：字串

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.Compute。EKS

提供叢集的名稱、所需的 Kubernetes 版本，以及允許 Kubernetes 控制平面管理 所需的 AWS 資源的角色NFs。Multus 容器網路介面 （CNI） 外掛程式已啟用。您可以連接多個網路介面，並將進階網路組態套用至以 Kubernetes 為基礎的網路函數。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 Multus 容器網路介面 （CNI） 用量的屬性。

如果您包含 multus，請指定 enabled和 multus_role 屬性。

enabled

指示是否已啟用預設 Multus 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：字串

ebs_csi

定義安裝在 Amazon EKS叢集中的 Amazon EBS Container Storage Interface （CSI） 驅動程式的屬性。

啟用此外掛程式以在 AWS Outposts、 AWS 本機區域或 上使用 Amazon EKS自我管理節點 AWS 區域。如需詳細資訊，請參閱 Amazon 使用者指南 中的 Amazon Elastic Block Store CSI驅動程式。EKS

enabled

指示是否已安裝預設 Amazon EBSCSI驅動程式。

必要：否

類型：布林值

version

Amazon EBSCSI驅動程式附加元件的版本。版本必須與 DescribeAddonVersions動作傳回的其中一個版本相符。如需詳細資訊，請參閱 DescribeAddonVersions Amazon EKSAPI參考中的

必要：否

類型：字串

屬性

`version`

叢集的 Kubernetes 版本。 AWS Telco Network Builder 支援 Kubernetes 版本 1.23 到 1.30。

必要：是

類型：字串

可能的值：1.23 | 1.24 | 1.25 | 1.26 | 1.27 | 1.28 | 1.29 | 1.30

`access`

叢集端點存取。

必要：是

類型：字串

可能的值：`PRIVATE` | `PUBLIC` | `ALL`

`cluster_role`

叢集管理的角色。

必要：是

類型：字串

`tags`

要附加至資源的標籤。

必要：否

類型：清單

`ip_family`

指示叢集中服務和 Pod 地址的 IP 系列。

允許的值：`IPv4`、 `IPv6`

預設值：IPv4

必要：否

類型：字串

要求

subnets

AWS.Networking.Subnet 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.ComputeEKS.AuthRole

AuthRole 可讓您將IAM角色新增至 Amazon EKS叢集，aws-authConfigMap讓使用者可以使用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 aws-auth EKS叢集 之IAM角色的映射清單ConfigMap。

`arn`

IAM 角色ARN的 。

必要：是

類型：字串

`groups`

要指派給 中定義角色的 Kubernetes 群組arn。

必要：否

類型：清單

要求

`clusters`

AWS.Compute.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.Compute。EKSManagedNode

AWS TNB 支援 EKS Managed Node 群組自動化 Amazon Kubernetes 叢集節點 （Amazon EC2執行個體） EKS 的佈建和生命週期管理。若要建立EKS節點群組，請執行下列動作：

- 提供 AMI或 AMI類型的 ID，為您的叢集工作者節點選擇 Amazon Machine Images （AMI）。
- 提供 Amazon EC2金鑰對以供SSH存取，以及節點群組的擴展屬性。
- 確保您的節點群組與 Amazon EKS叢集相關聯。
- 提供工作者節點的子網路。
- 或者，將安全群組、節點標籤和置放群組連接至節點群組。

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AMIs。

ami_type

Amazon EKS支援的AMI類型。

必要：是

類型：字串

可能的值： AL2_x86_64 | AL2_x86_64_GPU | | AL2_ARM_64 | CUSTOM BOTTLEROCKET_ARM_64 | BOTTLEROCKET_x86_64 | | BOTTLEROCKET_ARM_64_NVIDIA | BOTTLEROCKET_x86_64_NVIDIA

`ami_id`

的 IDAMI。

必要：否

類型：字串

> **Note**
> 如果範本中同時`ami_id`指定了 `ami_type`和 ，AWS TNB則 只會使用 `ami_id`值來建立 `EKSManagedNode`。

`instance_types`

執行個體大小。

必要：是

類型：清單

`key_pair`

啟用SSH存取的EC2金鑰對。

必要：是

類型：字串

`root_volume_encryption`

啟用 Amazon EBS根磁碟區的 Amazon EBS加密。如果未提供此屬性，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ARN的 。 AWS TNB 支援一般金鑰 ARN、多區域金鑰ARN和別名 ARN。

必要：否

類型：字串

> **Note**
>
> - 如果 `root_volume_encryption` 為 false，請勿包含 `root_volume_encryption_key_arn`。
> - AWS TNB 支援 Amazon EBS後端 AMI的根磁碟區加密。
> - 如果 AMI的根磁碟區已加密，您必須包含 `root_volume_encryption_key_arn` 的 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。
>
>   如果您不包含 `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執行個體數目，以及節點群組中 Amazon EC2執行個體數目的下限和上限。

`desired_size`

此 中的執行個體數目 NodeGroup。

必要：是

類型：整數

`min_size`

此 中的執行個體數目下限 NodeGroup。

必要：是

類型：整數

`max_size`

此 中的執行個體數目上限 NodeGroup。

必要：是

類型：整數

屬性

`node_role`

連接至 Amazon EC2執行個體IAM的角色ARN的 。

必要：是

類型：字串

`tags`

要連接至資源的標籤。

必要：否

類型：清單

要求

`cluster`

AWS.Compute.EKS 節點。

必要：是

類型：字串

`subnets`

AWS.Networking.Subnet 節點。

必要：是

類型：清單

`network_interfaces`

AWS.Networking.ENI 節點。確保網路介面和子網路設定為相同的可用區域，否則實例會失敗。

當您設定 時`network_interfaces`， AWS TNB如果您在 AWS.Compute. EKS節點中包含 `multus` 屬性， ENIs 會從 `multus_role` 屬性取得與 相關的許可。否則， ENIs 會從 node_role 屬性 AWS TNB取得與 相關的許可。

必要：否

類型：清單

`security_groups`

AWS.Networking.SecurityGroup 節點。

必要：否

類型：清單

`placement_group`

tosca.nodes AWS.Compute.PlacementGroup 節點。

必要：否

類型：字串

`user_data`

tosca.nodes AWS.Compute.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許可新增至傳遞至節點群組的 node_role。

必要：否

類型：字串

`labels`

節點標籤清單。節點標籤必須具有名稱和值。使用以下條件建立標籤：

- 名稱和值必須以 分隔=。
- 名稱和值的長度上限為 63 個字元。
- 標籤可包含字母（A-Z、a-z、）、數字（0-9）和下列字元：`[-, _, ., *, ?]`

- 名稱和值必須以英數字元、 ?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.Compute。EKSSelfManagedNode

AWS TNB 支援 Amazon EKS自我管理節點，以自動化 Amazon Kubernetes 叢集節點 （Amazon EC2 執行個體） EKS 的佈建和生命週期管理。若要建立 Amazon EKS節點群組，請執行下列動作：

- 提供 的 ID，為您的叢集工作者節點選擇 Amazon Machine Images （AMI）AMI。
- 提供 Amazon EC2金鑰對以供SSH存取。
- 確保您的節點群組與 Amazon EKS叢集相關聯。
- 提供執行個體類型和所需、最小和最大大小。
- 提供工作者節點的子網路。
- 或者，將安全群組、節點標籤和置放群組連接至節點群組。

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AMIs。

`ami_id`

用來啟動執行個體的 AMI ID。 AWS TNB 支援使用 的執行個體IMDSv2。如需詳細資訊，請參閱IMDS 版本。

必要：是

類型：字串

`instance_type`

執行個體大小。

必要：是

類型：字串

`key_pair`

啟用SSH存取的 Amazon EC2金鑰對。

必要：是

類型：字串

`root_volume_encryption`

啟用 Amazon EBS根磁碟區的 Amazon EBS加密。如果未提供此屬性， 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ARN的 。 AWS TNB 支援一般金鑰 ARN、多區域金鑰ARN和別名 ARN。

必要：否

類型：字串

> **Note**
>
> - 如果 `root_volume_encryption` 為 false，請勿包含 `root_volume_encryption_key_arn`。
> - AWS TNB 支援 Amazon EBS後端 AMI的根磁碟區加密。
> - 如果 AMI的根磁碟區已加密，您必須包含 `root_volume_encryption_key_arn` 的 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。
>
>   如果您不包含 `root_volume_encryption_key_arn`， AWS TNB 會使用 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執行個體數量的下限和上限。

`desired_size`

此 中的執行個體數目 NodeGroup。

必要：是

類型：整數

`min_size`

此 中的執行個體數目下限 NodeGroup。

必要：是

類型：整數

`max_size`

此 中的執行個體數目上限 NodeGroup。

必要：是

類型：整數

屬性

`node_role`

連接至 Amazon EC2執行個體IAM的角色ARN的 。

必要：是

類型：字串

`tags`

要連接至資源的標籤。標籤將傳播到資源建立的執行個體。

必要：否

類型：清單

要求

`cluster`

AWS.Compute.EKS 節點。

必要：是

類型：字串

`subnets`

AWS.Networking.Subnet 節點。

必要：是

類型：清單

`network_interfaces`

AWS.Networking.ENI 節點。確保網路介面和子網路設定為相同的可用區域，否則實例會失敗。

當您設定 時`network_interfaces`， AWS TNB如果您在 AWS.Compute. EKS節點中包含 `multus` 屬性， ENIs 會從 `multus_role` 屬性取得 的相關許可。否則， ENIs 會從 node_role 屬性 AWS TNB取得與 相關的許可。

必要：否

類型：清單

`security_groups`

AWS.Networking.SecurityGroup 節點。

必要：否

類型：清單

`placement_group`

tosca.nodes AWS.Compute.PlacementGroup 節點。

必要：否

類型：字串

`user_data`

tosca.nodes AWS.Compute.UserData 節點參考。使用者資料指令碼會傳遞至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許可新增至傳遞至節點群組的 node_role。

必要：否

類型：字串

`labels`

節點標籤清單。節點標籤必須具有名稱和值。使用以下條件建立標籤：

- 名稱和值必須以 分隔=。
- 名稱和值的長度上限為 63 個字元。
- 標籤可包含字母（A-Z、a-z、）、數字（0-9）和下列字元：`[-, _, ., *, ?]`
- 名稱和值必須以英數字元、?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.Compute。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 時EC2instance，Amazon EC2服務會嘗試以將您的所有執行個體分散到基礎硬體的方式放置執行個體，以將相關故障降至最低。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：字串

可能的值： CLUSTER | PARTITION | SPREAD_HOST | SPREAD_RACK

- CLUSTER – 將執行個體封裝在可用區域內。此策略可讓工作負載達成高效能運算 （HPC） 應用程式典型的緊密耦合 node-to-node 通訊所需的低延遲網路效能。

- PARTITION – 會將執行個體分散至邏輯分割區，讓一個分割區中的執行個體群組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 – 跨不同的基礎硬體放置一小群執行個體，以減少關聯的故障。
- SPREAD_HOST – 僅用於 Outpost 置放群組。跨不同的基礎硬體放置一小群執行個體，以減少關聯的故障。

`partition_count`

分割區數。

必要：只有在 `strategy` 設定為 時才需要PARTITION。

類型：整數

可能的值：1 | 2 |3 | 4 | 5 | 6 | 7

`tags`

您可以連接到置放群組資源的標籤。

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.Compute。UserData

AWS TNB 支援透過 UserData Network Service Descriptor（ ）中的節點啟動具有自訂使用者資料的Amazon EC2執行個體NSD。如需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 Shell 指令碼。

在網路實例化期間， AWS TNB 會透過使用者資料指令碼將 Amazon EC2執行個體註冊提供給叢集。同時提供自訂使用者資料時， AWS TNB 會合併兩個指令碼，並將其作為多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，請在開啟的捲曲支架 ! 之後新增驚嘆號{。例如，若要在指令碼MyVariable中使用 ，請輸入： {!MyVariable}

> **Note**
>
> - AWS TNB 支援大小上限為 7 KB 的使用者資料指令碼。
> - 由於 AWS TNB 用於 AWS CloudFormation 處理和轉譯multimime使用者資料指令碼，因此請確保指令碼遵守所有 AWS CloudFormation 規則。

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。格式必須為： `./scripts/script_name.sh`

必要：是

類型：字串

`content_type`

使用者資料指令碼的內容類型。

必要：是

類型：字串

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.Networking。SecurityGroup

AWS TNB 支援安全群組自動佈建 Amazon EC2 安全群組，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）、空格和下列特殊字元：._-：/（）#、@【】+=&；{}！$*

必要：是

類型：字串

`name`

安全群組的名稱。名稱最多可使用 255 個字元。您只能包含字母（A-Z 和 a-z）、數字（0-9）、空格和下列特殊字元：._-：/（）#、@【】+=&；{}！$*

必要：是

類型：字串

`tags`

您可以連接至安全群組資源的標籤。

必要：否

類型：清單

要求

`vpc`

AWS.Networking.VPC 節點。

必要：是

類型：字串

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.Networking。SecurityGroupEgressRule

AWS TNB 支援安全群組輸出規則，以自動佈建可連接至 AWS.Networking 的 Amazon EC2 安全群組輸出規則SecurityGroup。請注意，您必須提供 cidr_ip/destination_security_group/destination_prefix_list 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`

IPv4 地址範圍的CIDR格式。您必須指定允許輸出流量CIDR的範圍。

必要：否

類型：字串

`cidr_ipv6`

輸出流量的地址IPv6範圍CIDR，格式為 。您必須指定目的地安全群組（`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受管字首清單的字首清單 ID。這是來自與安全群組相關聯的節點群組執行個體的目的地。如需受管字首清單的詳細資訊，請參閱 Amazon VPC使用者指南 中的受管字首清單。

必要：否

類型：字串

`from_port`

如果通訊協定為 TCP或 UDP，則這是連接埠範圍的起點。如果通訊協定為 ICMP或 ICMPv6，則這是類型編號。值 -1 表示所有 ICMP/ICMPv6 類型。如果您指定所有 ICMP/ICMPv6 類型，則必須指定所有 ICMP/ICMPv6 代碼。

必要：否

類型：整數

`ip_protocol`

IP 通訊協定名稱（tcp、udp、icmp、icmpv6）或通訊協定編號。使用 -1 指定所有通訊協定。授權安全群組規則時，不論您指定的連接埠範圍為何，指定 tcp、udp、icmp 或 icmpv6 以外的 -1 或通訊協定編號，都允許所有連接埠上的流量。對於 tcp、udp 和 icmp，您必須指定連接埠範圍。對於 icmpv6，連接埠範圍是選用的；如果您省略連接埠範圍，則允許所有類型和代碼的流量。

必要：是

類型：字串

`to_port`

如果通訊協定為 TCP或 UDP，則這是連接埠範圍的結尾。如果通訊協定為 ICMP或 ICMPv6，則這是程式碼。值 -1 表示所有 ICMP/ICMPv6 碼。如果您指定所有 ICMP/ICMPv6 類型，則必須指定所有 ICMP/ICMPv6 代碼。

必要：否

類型：整數

要求

`security_group`

要新增此規則的安全群組 ID。

必要：是

類型：字串

`destination_security_group`

允許輸出流量的目的地安全群組 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.Networking。SecurityGroupIngressRule

AWS TNB 支援安全群組傳入規則，以自動佈建可連接至 AWS.Networking 的 Amazon EC2 安全群組傳入規則SecurityGroup。請注意，您必須提供 cidr_ip/source_security_group/source_prefix_list 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`

IPv4 地址範圍的CIDR格式。您必須指定允許輸入流量CIDR的範圍。

必要：否

類型：字串

`cidr_ipv6`

輸入流量的地址IPv6範圍CIDR，格式為 。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`

傳入（傳入）安全群組規則的說明。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受管字首清單的字首清單 ID。這是允許與安全群組相關聯的節點群組執行個體接收流量的來源。如需受管字首清單的詳細資訊，請參閱 Amazon VPC使用者指南 中的受管字首清單。

必要：否

類型：字串

`from_port`

如果通訊協定為 TCP或 UDP，則這是連接埠範圍的起點。如果通訊協定為 ICMP或 ICMPv6，則這是類型編號。值 -1 表示所有 ICMP/ICMPv6 類型。如果您指定所有 ICMP/ICMPv6 類型，則必須指定所有 ICMP/ICMPv6 代碼。

必要：否

類型：整數

`ip_protocol`

IP 通訊協定名稱（tcp、udp、icmp、icmpv6）或通訊協定編號。使用 -1 指定所有通訊協定。授權安全群組規則時，不論您指定的連接埠範圍為何，指定 tcp、udp、icmp 或 icmpv6 以外的 -1 或通訊協定編號，都允許所有連接埠上的流量。對於 tcp、udp 和 icmp，您必須指定連接埠範圍。對於 icmpv6，連接埠範圍是選用的；如果您省略連接埠範圍，則允許所有類型和代碼的流量。

必要：是

類型：字串

`to_port`

如果通訊協定為 TCP或 UDP，則這是連接埠範圍的結尾。如果通訊協定為 ICMP或 ICMPv6，則這是程式碼。值 -1 表示所有 ICMP/ICMPv6 碼。如果您指定所有 ICMP/ICMPv6 類型，則必須指定所有 ICMP/ICMPv6 代碼。

必要：否

類型：整數

要求

`security_group`

要新增此規則的安全群組 ID。

必要：是

類型：字串

`source_security_group`

允許輸入流量的來源安全群組 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.Resource.Import

您可以將下列 AWS 資源匯入 AWS TNB：

- VPC
- 子網路
- 路由表
- 網際網路閘道
- 安全群組

語法

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

屬性

`resource_type`

匯入 的資源類型 AWS TNB。

必要：否

類型：清單

`resource_id`

匯入 的資源 ID AWS TNB。

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.Networking。ENI

網路介面是 中代表虛擬網路卡VPC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 地址。在子網路中部署 Amazon EC2執行個體後，您可以將網路介面連接至該執行個體，或將網路介面與該 Amazon EC2執行個體分離，然後重新連接至該子網路中的另一個 Amazon EC2執行個體。裝置索引會以連接順序識別位置。

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`

指示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： true、 false

預設：true

必要：否

類型：布林值

`tags`

要連接至資源的標籤。

必要：否

類型：清單

要求

`subnet`

AWS.Networking.Subnet 節點。

必要：是

類型：字串

`security_groups`

AWS.Networking.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

生命週期掛鉤可讓您執行自己的指令碼，作為基礎設施和網路實例的一部分。

語法

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

功能

execution

執行掛鉤指令碼之掛鉤執行引擎的屬性。

`type`

掛鉤執行引擎類型。

必要：否

類型：字串

可能的值： `CODE_BUILD`

要求

`definition`

AWS.HookDefinition.Bash 節點。

必要：是

類型：字串

`vpc`

AWS.Networking.VPC 節點。

必要：是

類型：字串

範例

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

AWS.Networking。InternetGateway

定義 AWS 網際網路閘道節點。

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

dest_cidr

用於目的地比對的IPv4CIDR區塊。此屬性用於在 中建立路由，RouteTable其值會用作 DestinationCidrBlock。

必要：如果您包含 ipv6_dest_cidr 屬性，則否。

類型：字串

ipv6_dest_cidr

用於目的地比對的IPv6CIDR區塊。

必要：如果您包含 dest_cidr 屬性，則否。

類型：字串

屬性

tags

要連接至資源的標籤。

必要：否

類型：清單

egress_only

IPv6特定屬性。指示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`

AWS.Networking.VPC 節點。

必要：是

類型：字串

`route_table`

AWS.Networking.RouteTable 節點。

必要：是

類型：字串

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.Networking。RouteTable

路由表包含一組稱為路由的規則，用於決定來自 VPC或閘道內子網路的網路流量導向位置。您必須將路由表與 建立關聯VPC。

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`

AWS.Networking.VPC 節點。

必要：是

類型：字串

範例

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

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

AWS.Networking.Subnet

子網路是 中 IP 地址的範圍VPC，且必須完全位於一個可用區域中。您必須為子網路指定 VPC、 CIDR 區塊、可用區域和路由表。您也必須定義子網路是私有還是公有。

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接收公有IPv4地址。

必要：是

類型：字串

可能的值：PUBLIC | PRIVATE

`availability_zone`

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

必要：是

類型：字串

`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`

透過 Amazon 建立之子網路的 IPv6CIDR區塊的 2 位數十六進位尾碼VPC。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 AWS Outposts 其中建立子網路ARN的 。如果您想要在 上啟動 Amazon EKS自我管理節點，請將此屬性新增至NSD範本 AWS Outposts。如需詳細資訊，請參閱 Amazon 使用者指南 EKS AWS Outposts 中的 Amazon。 EKS

如果您將此屬性新增至NSD範本，則必須將`availability_zone`屬性的值設定為 的可用區域 AWS Outposts。

必要：否

類型：字串

`tags`

要連接至資源的標籤。

必要：否

類型：清單

要求

vpc

AWS.Networking.VPC 節點。

必要：是

類型：字串

route_table

AWS.Networking.RouteTable 節點。

必要：是

類型：字串

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。VNFDeployment

NF 部署的模型是提供基礎設施和與其相關聯的應用程式。叢集屬性會指定託管 的EKS叢集NFs。vnfs 屬性會指定部署的網路函數。您也可以提供類型 pre_create 和 post_create 的選用生命週期掛鉤操作，以執行 部署的特定指示，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.Deployment.VNFDeployment 節點。

必要：否

類型：字串

`cluster`

AWS.Compute.EKS 節點。

必要：是

類型：字串

vnfs

AWS.VNF 節點。

必要：是

類型：字串

介面

掛鉤

定義執行生命週期掛鉤的階段。

pre_create

AWS.HookExecution 節點。此掛鉤會在VNFDeployment節點部署之前執行。

必要：否

類型：字串

post_create

AWS.HookExecution 節點。此掛鉤會在VNFDeployment節點部署後執行。

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.Networking。VPC

您必須為虛擬私有雲端（ ）指定CIDR區塊VPC。

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表示CIDR法中 IPv4的網路範圍。

必要：是

類型：字串

`ipv6_cidr_block`

用來建立 的IPv6CIDR區塊VPC。

允許的值： `AMAZON_PROVIDED`

必要：否

類型：字串

`dns_support`

指示在 中啟動的執行個體是否VPC取得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.Networking。NATGateway

您可以透過子網路定義公有或私有NAT閘道節點。對於公有閘道，如果您不提供彈性 IP 配置 ID，AWS TNB將為您的帳戶配置彈性 IP，並將其與閘道建立關聯。

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.Networking.Subnet 節點參考。

必要：是

類型：字串

`internet_gateway`

AWS.Networking.InternetGateway 節點參考。

必要：是

類型：字串

屬性

`type`

指示閘道是公有還是私有。

允許的值：PUBLIC、 PRIVATE

必要：是

類型：字串

`eip_allocation_id`

代表彈性 IP 地址配置的 ID。

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.Networking.Route

您可以定義路由節點，將目的地路由關聯至NAT閘道作為目標資源，並將路由新增至關聯的路由表。

語法

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

屬性

`dest_cidr_blocks`

目標資源的目的地IPv4路由清單。

必要：是

類型：清單

成員類型：字串

屬性

`nat_gateway`

AWS.Networking.NATGateway 節點參考。

必要：是

類型：字串

`route_table`

AWS.Networking.RouteTable 節點參考。

必要：是

類型：字串

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
```

常見節點

定義 NSD和 的節點VNFD。

- AWS.HookDefinition.Bash

AWS.HookDefinition.Bash

在 AWS HookDefinition 中定義 bash。

語法

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

屬性

`implementation`

掛鉤定義的相對路徑。格式必須為： `./hooks/script_name.sh`

必要：是

類型：字串

`environment_variables`

hook bash 指令碼的環境變數。使用下列格式：**envName**=**envValue**搭配下列 regex： `^[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**值符合下列條件：

- 請勿使用空格。
- 從字母（A-Z 或 a-z）或數字（0-9）**envName**開始。
- 請勿使用下列 AWS TNB預留關鍵字（不區分大小寫）啟動環境變數名稱：
  - CODEBUILD
  - TNB
  - HOME
  - AWS
- 您可以使用任意數量的字母（A-Z 或 a-z）、數字（0-9）和特殊字元-，以及 **_ envName**和**envValue**。

範例：`A123-45xYz=Example_789`

必要：否

類型：清單

`execution_role`

掛鉤執行的角色。

必要：是

類型：字串

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 TNB 中的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是 AWS 和 之間的共同責任。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：

- 雲端的安全性 – AWS 負責保護在 中執行 AWS 服務的基礎設施 AWS 雲端。 AWS 也提供您可以安全使用的服務。第三方稽核人員會定期測試和驗證我們安全的有效性，做為AWS 合規計畫的一部分。若要了解適用於 AWS Telco Network Builder 的合規計劃，請參閱AWS 合規計劃範圍內的服務。
- 雲端安全性 – 您的責任取決於您使用 AWS 的服務。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。

本文件可協助您了解如何在使用 AWS TNB 時套用共同責任模型。下列主題說明如何設定 AWS TNB 以符合您的安全和合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS TNB 資源。

目錄

AWS TNB 中的資料保護

AWS 共同責任模型適用於 Telco Network Builder AWS 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 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，建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 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。如需使用 CloudTrail 線索擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 線索。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 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 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，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) 140-3。

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 AWS TNB 或其他 AWS 服務 使用主控台、API AWS CLI或 AWS SDKs時。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 會標記您的資料以供刪除，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。120 天後， AWS TNB 會永久刪除您的資料。 AWS TNB 也會終止您的網路，並刪除函數套件和網路套件。

靜態加密

AWS TNB 一律會加密存放在服務中的所有靜態資料，而不需要任何額外的組態。此加密會透過 自動進行 AWS Key Management Service。

傳輸中加密

AWS TNB 使用 Transport Layer Security (TLS) 1.2 保護傳輸中的所有資料。

您有責任加密模擬代理程式與其用戶端之間的資料。

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端 (VPC) 中。所有內部 AWS TNB 流量都停留在 AWS 網路中，不會周遊網際網路。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 的身分和存取管理

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證 （登入） 和授權 （具有許可） 來使用 AWS TNB 資源。IAM 是 AWS 服務 您可以免費使用的 。

目錄

目標對象

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

服務使用者 – 如果您使用 AWS TNB 服務來執行您的任務，則您的管理員會為您提供所需的登入資料和許可。當您使用更多 AWS TNB 功能來執行工作時，您可能需要額外的許可。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 中的功能，請參閱 對 AWS Telco Network Builder 身分和存取進行故障診斷。

服務管理員 – 如果您在公司負責 AWS TNB 資源，您可能可以完整存取 AWS TNB。您的任務是判斷服務使用者應存取的 AWS TNB 功能和資源。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搭配 AWS TNB 使用 IAM，請參閱AWS TNB 如何與 IAM 搭配使用。

IAM 管理員 – 如果您是 IAM 管理員，建議您了解撰寫政策以管理 AWS TNB 存取的詳細資訊。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 Telco Network Builder 的身分型政策範例。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （ 登入 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您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。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

視您身分的使用者類型而定，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的登入資料以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。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》中的適用於 API 請求的AWS Signature 第 4 版。

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) 來提高帳戶的安全性。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 IAM 中的AWS 多重要素驗證。

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 帳戶中的所有 AWS 服務 和資源。此身分稱為 AWS 帳戶 Theroot 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

最佳實務是， 要求人類使用者，包括需要管理員存取權的使用者，使用臨時登入資料 AWS 服務 來使用與身分提供者的聯合來存取 。

聯合身分是來自您的企業使用者目錄、Web 身分提供者、 AWS Directory Service、身分中心目錄，或是 AWS 服務 使用透過身分來源提供的憑證存取的任何使用者。當聯合身分存取時 AWS 帳戶，它們會擔任角色，而角色會提供臨時憑證。

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，或者您可以連接並同步到您自己的身分來源中的一組使用者 AWS 帳戶 和群

組，以便在所有 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，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 中具有單一個人或應用程式特定許可 AWS 帳戶 的身分。建議您盡可能依賴臨時憑證，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，但角色僅提供臨時憑證。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。若要暫時在 中擔任 IAM 角色 AWS Management Console，您可以從使用者切換至 IAM 角色（主控台）。您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需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 (聯合) 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 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以直接將政策連接到資源（而不是使用角色做為代理）。如需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 中的跨帳戶資源存取。

- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。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。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在 中執行動作時 AWS，您會被視為委託人。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 使用呼叫 的委託人許可 AWS 服務，並結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 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。如需詳細資訊，請參閱「IAM 使用者指南」中的建立角色以委派許可權給 AWS 服務。
  - 服務連結角色 – 服務連結角色是連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。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 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。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，並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。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臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接至身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件，當與身分或資源建立關聯時， AWS 會定義其許可。當委託人（使用者、根使用者或角色工作階段）發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 的形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，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。具有該政策的使用者可以從 AWS Management Console AWS CLI、 或 API AWS 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。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

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。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策之間選擇，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。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 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。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 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政策類型。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。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。

- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 的多個的服務。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 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括支援 RCPs AWS 服務 的 清單，請參閱AWS Organizations 《 使用者指南》中的資源控制政策 RCPs)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。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。如需詳細資訊，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。若要了解如何 AWS 在涉及多種政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS TNB 如何與 IAM 搭配使用

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

您可以搭配 Telco Network Builder AWS 使用的 IAM 功能

IAM 功能	AWS TNB 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是

IAM 功能	AWS TNB 支援
政策條件索引鍵	是
ACL	否
ABAC (政策中的標籤)	是
暫時性憑證	是
主體許可	是
服務角色	否
服務連結角色	否

若要深入了解 AWS TNB 和其他 AWS 服務如何搭配大多數 IAM 功能運作，請參閱《AWS IAM 使用者指南》中的搭配 IAM 運作的服務。

AWS TNB 的身分型政策

支援身分型政策：是

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。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

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。

AWS TNB 的身分型政策範例

若要檢視 AWS TNB 身分型政策的範例，請參閱 AWS Telco Network Builder 的身分型政策範例。

AWS TNB 中的資源型政策

支援資源型政策：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。主體可以包含帳戶、使用者、角色、聯合身分使用者，或 AWS 服務。

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，做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。當委託人和資源位於不同的位置時 AWS 帳戶，信任帳戶中的 IAM 管理員也必須授予委託人實體（使用者或角色）存取資源的許可。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 中的快帳戶資源存取。

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。政策動作通常具有與相關聯 AWS API 操作相同的名稱。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中的 AWS Telco Network Builder 定義的動作。

AWS TNB 中的政策動作在動作之前使用以下字首：

```
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 身分型政策的範例，請參閱 AWS Telco Network Builder 的身分型政策範例。

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": "*"
```

若要查看 AWS TNB 資源類型及其 ARNs的清單，請參閱服務授權參考中的 AWS Telco Network Builder 定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 Telco Network Builder AWS 定義的動作。

若要檢視 AWS TNB 身分型政策的範例，請參閱 AWS Telco Network Builder 的身分型政策範例。

AWS TNB 的政策條件索引鍵

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， 會使用邏輯OR操作 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的條件金鑰。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 AWS TNB 條件金鑰清單，請參閱服務授權參考中的 AWS Telco Network Builder 的條件金鑰。若要了解您可以使用條件金鑰的動作和資源，請參閱 Telco Network Builder AWS 定義的動作。

若要檢視 AWS TNB 身分型政策的範例，請參閱 AWS Telco Network Builder 的身分型政策範例。

AWS TNB ACLs

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。

ABAC 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。您可以將標籤連接至 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。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，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》中的使用 ABAC 授權定義許可。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 使用臨時憑證

支援臨時憑證：是

當您使用臨時登入資料登入時，有些 AWS 服務 無法使用。如需詳細資訊，包括哪些 AWS 服務 使用臨時登入資料，請參閱《AWS 服務 IAM 使用者指南》中的使用 IAM 的 。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入 ，則會使用臨時登入資料。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取 時，該程序會自動建立臨時登入資料。

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》中的從使用者切換至 IAM 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 建議使用您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

AWS TNB 的跨服務主體許可

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，您會被視為委託人。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 使用呼叫 的委託人許可 AWS 服務，並結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 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。如需詳細資訊，請參閱「IAM 使用者指南」中的建立角色以委派許可權給 AWS 服務。

AWS TNB 的服務連結角色

支援服務連結角色：否

服務連結角色是連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

AWS Telco Network Builder 的身分型政策範例

根據預設，使用者和角色沒有建立或修改 AWS TNB 資源的許可。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行任務。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。

如需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 (主控台)。

如需 AWS TNB 定義的動作和資源類型的詳細資訊，包括每個資源類型的 ARNs 格式，請參閱服務授權參考中的 AWS Telco Network Builder 的動作、資源和條件索引鍵。

目錄

政策最佳實務

以身分為基礎的政策會判斷是否有人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 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用 AWS 受管政策，將許可授予許多常見使用案例。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。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。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。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。如果透過特定 使用服務動作 AWS 服務，您也可以使用條件來授予存取，例如 AWS CloudFormation。如需詳細資訊，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，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA)：如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以增加安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_configure-api-require.html中的透過 MFA 的安全 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 Network Builder 主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 AWS TNB 資源的詳細資訊 AWS 帳戶。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。

對於僅對 AWS CLI 或 AWS API 進行呼叫的使用者，您不需要允許最低主控台許可。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

身為管理員，您擁有和管理 AWS TNB 建立的資源，如環境和服務範本所定義。您必須將 IAM 服務角色連接至您的帳戶，以允許 AWS TNB 為您的網路生命週期管理建立資源。

IAM 服務角色允許 AWS TNB 代您呼叫資源，以執行個體化和管理網路。如果您指定服務角色， AWS TNB 會使用該角色的登入資料。

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

AWS TNB 服務角色

身為平台團隊的成員，您可以身為管理員建立 AWS TNB 服務角色，並將其提供給 AWS TNB。此角色允許 AWS TNB 呼叫其他 服務，例如 Amazon Elastic Kubernetes Service AWS CloudFormation ，並為您的網路佈建所需的基礎設施，以及如 NSD 中所定義的佈建網路函數。

建議您針對 AWS TNB 服務角色使用下列 IAM 角色和信任政策。縮小此政策的許可範圍時，請記住 TNB AWS 可能會失敗，導致存取遭拒錯誤導致從您的政策中剔除資源。

下列程式碼顯示 AWS TNB 服務角色政策：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:GetCallerIdentity"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "AssumeRole"
        },
        {
```

```
            "Action": [
                "tnb:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBPolicy"
        },
        {
            "Action": [
                "iam:AddRoleToInstanceProfile",
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:TagInstanceProfile",
                "iam:UntagInstanceProfile"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "IAMPolicy"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "eks.amazonaws.com",
                        "eks-nodegroup.amazonaws.com"
                    ]
                }
            },
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBAccessSLRPermissions"
        },
        {
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateOrUpdateTags",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteTags",
                "autoscaling:DescribeAutoScalingGroups",
```

```
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:DescribeTags",
                "autoscaling:UpdateAutoScalingGroup",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:CreateSecurityGroup",
                "ec2:DeleteLaunchTemplateVersions",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeTags",
                "ec2:GetLaunchTemplateData",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RunInstances",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:CreateInternetGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSubnet",
                "ec2:CreateTags",
                "ec2:CreateVpc",
                "ec2:DeleteInternetGateway",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSubnet",
                "ec2:DeleteTags",
                "ec2:DeleteVpc",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeInstances",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeSubnets",
```

```
                "ec2:DescribeVpcs",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySecurityGroupRules",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:AllocateAddress",
                "ec2:AssignIpv6Addresses",
                "ec2:AssociateAddress",
                "ec2:AssociateNatGatewayAddress",
                "ec2:AssociateVpcCidrBlock",
                "ec2:CreateEgressOnlyInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:DeleteEgressOnlyInternetGateway",
                "ec2:DeleteNatGateway",
                "ec2:DescribeAddresses",
                "ec2:DescribeEgressOnlyInternetGateways",
                "ec2:DescribeNatGateways",
                "ec2:DisassociateAddress",
                "ec2:DisassociateNatGatewayAddress",
                "ec2:DisassociateVpcCidrBlock",
                "ec2:ReleaseAddress",
                "ec2:UnassignIpv6Addresses",
                "ec2:DescribeImages",
                "eks:CreateCluster",
                "eks:ListClusters",
                "eks:RegisterCluster",
                "eks:TagResource",
                "eks:DescribeAddonVersions",
                "events:DescribeRule",
                "iam:GetRole",
                "iam:ListAttachedRolePolicies",
                "iam:PassRole"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "TNBAccessComputePerms"
        },
        {
            "Action": [
                "codebuild:BatchDeleteBuilds",
                "codebuild:BatchGetBuilds",
                "codebuild:CreateProject",
                "codebuild:DeleteProject",
```

```
                "codebuild:ListBuildsForProject",
                "codebuild:StartBuild",
                "codebuild:StopBuild",
                "events:DeleteRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "s3:CreateBucket",
                "s3:GetBucketAcl",
                "s3:GetObject",
                "eks:DescribeNodegroup",
                "eks:DeleteNodegroup",
                "eks:AssociateIdentityProviderConfig",
                "eks:CreateNodegroup",
                "eks:DeleteCluster",
                "eks:DeregisterCluster",
                "eks:UpdateAddon",
                "eks:UpdateClusterVersion",
                "eks:UpdateNodegroupConfig",
                "eks:UpdateNodegroupVersion",
                "eks:DescribeUpdate",
                "eks:UntagResource",
                "eks:DescribeCluster",
                "eks:ListNodegroups",
                "eks:CreateAddon",
                "eks:DeleteAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonVersions",
                "s3:PutObject",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateStack",
                "cloudformation:UpdateTerminationProtection"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/tnb*",
                "arn:aws:codebuild:*:*:project/tnb*",
                "arn:aws:logs:*:*:log-group:/aws/tnb*",
                "arn:aws:s3:::tnb*",
                "arn:aws:eks:*:*:addon/tnb*/*/*",
                "arn:aws:eks:*:*:cluster/tnb*",
                "arn:aws:eks:*:*:nodegroup/tnb*/tnb*/*",
```

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

下列程式碼顯示 AWS TNB 服務信任政策：

```
{
  "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

當您在 NSD 中建立 Amazon EKS 資源時，請提供 `cluster_role` 屬性來指定要用來建立 Amazon EKS 叢集的角色。

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

```
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"
```

如需使用 AWS CloudFormation 範本之 IAM 角色的詳細資訊，請參閱 AWS CloudFormation 使用者指南中的下列章節：

- AWS::IAM::Role
- 選取堆疊範本

AWS Amazon EKS 節點群組的 TNB 服務角色

當您在 NSD 中建立 Amazon EKS 節點群組資源時，請提供 `node_role` 屬性來指定要用來建立 Amazon EKS 節點群組的角色。

下列範例顯示為 Amazon EKS 節點群組政策建立 AWS TNB 服務角色的 AWS CloudFormation 範本。

```
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/*"
```

如需使用 AWS CloudFormation 範本之 IAM 角色的詳細資訊，請參閱 AWS CloudFormation 使用者指南中的下列章節：

- AWS::IAM::Role
- 選取堆疊範本

AWS Multus 的 TNB 服務角色

當您在 NSD 中建立 Amazon EKS 資源，且您想要在部署範本中管理 Multus 時，必須提供 `multus_role` 屬性來指定要用於管理 Multus 的角色。

下列範例顯示為 Multus 政策建立 AWS TNB 服務角色的 AWS CloudFormation 範本。

```
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: "*"
```

如需使用 AWS CloudFormation 範本之 IAM 角色的詳細資訊，請參閱 AWS CloudFormation 使用者指南中的下列章節：

- AWS::IAM::Role
- 選取堆疊範本

AWS 生命週期掛鉤政策的 TNB 服務角色

當您的 NSD 或網路函數套件使用生命週期掛鉤時，您需要一個服務角色，以允許您建立環境來執行生命週期掛鉤。

> **Note**
>
> 您的生命週期掛鉤政策應根據您的生命週期掛鉤嘗試執行的操作而定。

下列範例顯示為生命週期掛鉤政策建立 AWS TNB 服務角色的 AWS CloudFormation 範本。

```
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"
```

如需使用 AWS CloudFormation 範本之 IAM 角色的詳細資訊，請參閱 AWS CloudFormation 使用者指南中的下列章節：

- 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。此政策包含在主控台上完成此動作的許可，或使用 AWS CLI 或 AWS API 以程式設計方式完成此動作的許可。

```
{
    "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": "*"
        }
    ]
}
```

對 AWS Telco Network Builder 身分和存取進行故障診斷

使用下列資訊來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

- 我無權在 AWS TNB 中執行動作
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許 以外的人員 AWS 帳戶 存取我的 AWS TNB 資源

我無權在 AWS TNB 中執行動作

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 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，表示您無權執行`iam:PassRole`動作，則必須更新您的政策，以允許您將角色傳遞至 AWS TNB。

有些 AWS 服務 可讓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 `marymajor` 使用者嘗試使用主控台在 AWS TNB 中執行動作時，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 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取我的 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：

- 若要了解 AWS TNB 是否支援這些功能，請參閱 AWS TNB 如何與 IAM 搭配使用。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個資源中提供存取權給 IAM 使用者。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的提供存取權給第三方 AWS 帳戶 擁有。
- 如需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)。
- 如需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 中的跨帳戶資源存取。

AWS TNB 的合規驗證

若要了解 是否 AWS 服務 在特定合規計劃的範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS Compliance Programs。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱在 中下載報告 AWS Artifact。

使用 時的合規責任 AWS 服務 取決於資料的敏感度、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源以協助合規：

- 安全合規與治理 - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- Amazon Web Services 上的 HIPAA 安全與合規架構 – 本白皮書說明公司如何使用 AWS 來建立符合 HIPAA 資格的應用程式。

> **Note**
>
> 並非所有 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 – 此工作手冊和指南的集合可能適用於您的產業和位置。
- AWS 客戶合規指南 – 透過合規的角度了解共同責任模型。本指南摘要說明保護 的最佳實務， AWS 服務 並將指南映射到跨多個架構的安全控制 （ 包括國家標準和技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO))。
- AWS Config 開發人員指南中的使用規則評估資源 – AWS Config 服務會評估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 可讓您全面檢視其中的安全狀態 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 控制參考」。
- Amazon GuardDuty – 這可透過監控您的環境是否有可疑和惡意活動，來 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可滿足特定合規架構所規定的入侵偵測需求，以協助您因應 PCI DSS 等各種不同的合規需求。
- AWS Audit Manager – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險的方式，以及符合法規和產業標準的方式。

AWS TNB 中的彈性

AWS 全域基礎設施是以 AWS 區域 和 可用區域為基礎建置。 AWS 區域 提供多個實體分隔和隔離的可用區域，這些區域與低延遲、高輸送量和高度備援聯網連接。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 全球基礎設施。

AWS TNB 會在您選擇的 AWS 區域中的虛擬私有雲端 (VPC) 中執行 Network Service on EKS 叢集。

AWS TNB 中的基礎設施安全性

身為受管服務， AWS Telco Network Builder 受到 AWS 全球網路安全的保護。如需 AWS 安全服務及如何 AWS 保護基礎設施的相關資訊，請參閱AWS 雲端安全。若要使用基礎設施安全性的最佳實務來設計您的 AWS 環境，請參閱 Security Pillar AWS Well-Architected Framework 中的基礎設施保護。

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範例：

- AWS 負責保護支援 AWS TNB 的元件，包括：
  - 運算執行個體 （ 也稱為工作者)
  - 內部資料庫
  - 內部元件之間的網路通訊
  - AWS TNB 應用程式程式設計界面 (API)
  - AWS 軟體開發套件 (SDK)
- 您有責任保護對 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
  - 用來存放 AWS TNB 資料的 S3 儲存貯體
  - 您用來支援您透過 AWS TNB 佈建之網路服務的其他 AWS 服務 和資源
  - 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**
>
> 您有責任實作災難復原計劃，以有效復原您透過 AWS TNB 佈建的網路服務。

網路連線安全模型

您透過 AWS TNB 佈建的網路服務，會在虛擬私有雲端 (VPC) 內的運算執行個體上執行，該虛擬私有雲端位於您選取的 AWS 區域中。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設施。VPCs內的運算執行個體之間的通訊會保留在 AWS 網路中，不會透過網際網路傳輸。有些內部服務通訊會跨網際網路，並會加密。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使用相同 VPC 中的個別運算執行個體。

您的網路服務用戶端與 AWS TNB 中的網路服務之間的通訊周遊網際網路。 AWS TNB 不會管理這些連線。您有責任保護用戶端連線的安全。

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

IMDS 版本

AWS TNB 支援使用執行個體中繼資料服務第 2 版 (IMDSv2) 的執行個體，這是一種工作階段導向的方法。IMDSv2 的安全性高於 IMDSV1。如需詳細資訊，請參閱使用 Amazon EC2 執行個體中繼資料服務的增強功能，對開放防火牆、反向代理和 SSRF 漏洞新增深度防禦。

啟動執行個體時，您必須使用 IMDSv2。如需 IMDSv2 的詳細資訊，請參閱《Amazon EC2 使用者指南》中的使用 IMDSv22。 Amazon EC2

監控 AWS TNB

監控是維護 和其他 AWS 解決方案可靠性、可用性和效能 AWS TNB的重要部分。 AWS 提供 AWS CloudTrail 觀看 AWS TNB、報告錯誤，並在適當時採取自動動作。

使用 CloudTrail 擷取對 進行呼叫的詳細資訊 AWS APIs。您可以將這些呼叫儲存為 Amazon S3 中的日誌檔案。您可以使用這些 CloudTrail 日誌來判斷該資訊，例如進行何種呼叫、呼叫來源 IP 地址、進行呼叫的人員，以及進行呼叫的時間。

CloudTrail 日誌包含 API動作呼叫的相關資訊 AWS TNB。它們也包含 Amazon EC2和 Amazon 等服務對API動作的呼叫資訊EBS。

使用 AWS 記錄 Telco Network Builder API呼叫 AWS CloudTrail

AWS Telco Network Builder 與 整合AWS CloudTrail，此服務提供使用者、角色或 所採取動作的記錄 AWS 服務。 CloudTrail 會將 的所有API呼叫 AWS TNB擷取為事件。擷取的 AWS TNB呼叫包括從主控台呼叫，以及對 API 操作的 AWS TNB程式碼呼叫。使用 收集的資訊 CloudTrail，您可以判斷所提出的請求 AWS TNB、提出請求的 IP 地址、提出時間，以及其他詳細資訊。

每一筆事件或日誌專案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：

- 該請求是使用根使用者還是使用者憑證提出。
- 是否代表 IAM Identity Center 使用者提出請求。
- 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 服務服務提出。

CloudTrail 當您建立 帳戶 AWS 帳戶 時， 會在 中處於作用中狀態，而且您會自動存取 CloudTrail 事件歷史記錄。 CloudTrail TheEvent 歷史記錄提供過去 90 天內記錄的管理事件的可檢視、可搜尋、可下載和不可變記錄 AWS 區域。如需詳細資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 事件歷史記錄。檢視事件歷史記錄無需 CloudTrail付費。

如需 AWS 帳戶 過去 90 天內持續記錄的事件，請建立追蹤或 CloudTrail Lake 事件資料存放區。

CloudTrail 線索

線索可讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。使用 建立的所有線索 AWS Management Console 都是多區域。您可以使用 建立單一區域或多區域追蹤 AWS CLI。建議您建

立多區域追蹤，因為您擷取 AWS 區域 帳戶中所有 的活動。如果您建立單一區域追蹤，您只能檢視記錄於追蹤的事件 AWS 區域。如需追蹤的詳細資訊，請參閱AWS CloudTrail 《 使用者指南》中的為您的 建立追蹤 AWS 帳戶和為組織建立追蹤。

您可以透過 CloudTrail 建立線索，免費將一份持續管理事件的副本交付至 Amazon S3 儲存貯體，但需要支付 Amazon S3 儲存費用。如需 CloudTrail 定價的詳細資訊，請參閱AWS CloudTrail 定價。如需 Amazon S3 定價的相關資訊，請參閱 Amazon S3 定價。

CloudTrail Lake 事件資料存放區

CloudTrail Lake 可讓您在事件上執行SQL以 為基礎的查詢。 CloudTrail Lake 會以資料列為基礎的JSON格式將現有事件轉換為 Apache ORC 格式。 ORC 是一種欄式儲存格式，已針對快速擷取資料進行最佳化。系統會將事件彙總到事件資料存放區中，事件資料存放區是事件的不可變集合，其依據為您透過套用進階事件選取器選取的條件。套用於事件資料存放區的選取器控制哪些事件持續存在並可供您查詢。如需 CloudTrail Lake 的詳細資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 AWS CloudTrail Lake。

CloudTrail Lake 事件資料存放區和查詢會產生成本。建立事件資料存放區時，您可以選擇要用於事件資料存放區的定價選項。此定價選項將決定擷取和儲存事件的成本，以及事件資料存放區的預設和最長保留期。如需 CloudTrail 定價的詳細資訊，請參閱AWS CloudTrail 定價。

AWS TNB 事件範例

事件代表來自任何來源的單一請求，並包含請求API操作、操作日期和時間、請求參數等相關資訊。CloudTrail log 檔案不是公開API呼叫的排序堆疊追蹤，因此事件不會以任何特定順序顯示。

下列範例顯示示範 `CreateSolFunctionPackage`操作 CloudTrail 的事件。

```
{
    "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"
}
```


如需 CloudTrail 記錄內容的資訊，請參閱 AWS CloudTrail 使用者指南中的CloudTrail記錄內容。

AWS TNB 部署任務

了解部署任務，以有效地監控部署並更快地採取行動。

下表列出 AWS TNB部署任務：

2024 年 3 月 7 日之前開始的部署任務名稱	2024 年 3 月 7 日及之後開始的部署任務名稱	Task description (任務描述)
AppInstallation	ClusterPluginInstall	在 Amazon EKS叢集上安裝 Multus 外掛程式。
AppUpdate	名稱沒有變更	更新已在網路執行個體中安裝的網路函數。
-	ClusterPluginUninstall	在 Amazon EKS叢集上解除安裝外掛程式。
ClusterStorageClassConfiguration	名稱沒有變更	在 Amazon EKS叢集上設定儲存類別 （CSI 驅動程式）。
FunctionDeletion	名稱沒有變更	從 AWS TNB資源中刪除網路函數。
FunctionInstantiation	FunctionInstall	使用 部署網路函數HELM。
FunctionUninstallation	FunctionUninstall	從 Amazon EKS叢集解除安裝網路函數。
HookExecution	名稱沒有變更	執行 中定義的生命週期掛鉤NSD。
InfrastructureCancellation	名稱沒有變更	取消網路服務。
InfrastructureInstantation	名稱沒有變更	代表使用者佈建 AWS 資源。
InfrastructureTermination	名稱沒有變更	取消佈建透過 叫用 AWS 的資源 AWS TNB。
-	InfrastructureUpdate	更新代表使用者佈建 AWS 的資源。
InventoryDeregistration	名稱沒有變更	從 取消註冊 AWS 資源 AWS TNB。
-	InventoryRegistration	在 中註冊 AWS 資源 AWS TNB。
KubernetesClusterConfiguration	ClusterConfiguration	設定 Kubernetes 叢集，並將其他IAM角色新增至 Amazon，EKS AuthMap 如 中所定義NSD。

2024 年 3 月 7 日之前開始的部署任務名稱	2024 年 3 月 7 日及之後開始的部署任務名稱	Task description (任務描述)
NetworkServiceFinalization	名稱沒有變更	完成網路服務並提供成功或失敗狀態更新。
NetworkServiceInstantiation	名稱沒有變更	初始化網路服務。
SelfManagedNodesConfiguration	名稱沒有變更	使用 Amazon EKS和 Kubernetes 控制平面啟動自我管理節點。
-	ValidateNetworkServiceUpdate	在更新網路執行個體之前執行驗證。

的服務配額 AWS TNB

服務配額也稱為限制，是 AWS 您的帳戶的服務資源或操作數量上限。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 的服務配額 AWS TNB。

名稱	預設	可調整	描述
並行的持續網路服務操作	每個受支援的區域：40	是	一個區域中並行進行中網路服務操作的數量上限。
函數套件	每個受支援的區域：200	是	一個區域中的函數套件數目上限。
網路套件	每個受支援的區域：40	是	一個區域中的網路套件數量上限。
網路服務執行個體	每個支援的區域：800	是	一個區域中的網路服務執行個體數量上限。

使用者指南的文件歷史記錄 AWS TNB

下表說明 的文件版本 AWS TNB。

變更	描述	日期
叢集的 Kubernetes 版本	AWS TNB 現在支援 Kubernetes 1.30 版來建立 Amazon EKS 叢集。	2024 年 8 月 19 日

AWS TNB 支援管理網路生命週期的額外操作。	您可以使用新的網路套件和參數值來更新現化或先前更新的網路執行個體。請參閱：
	• 生命週期操作
	• 更新網路執行個體
	• AWS TNB 服務角色範例：
	  • 新增這些 Amazon EKS 動作：`eks:UpdateAddon` 、`eks:UpdateClusterVersion` 、`eks:UpdateNodegroupConfig` 、`eks:UpdateNodegroupVersion` 、`eks:DescribeUpdate`
	  • 新增此 AWS CloudFormation 動作： `cloudformation:UpdateStack`
	• 新的部署任務：`InfrastructureUpdate` 、`Inventory` `Registration` 、
		2024 年 7 月 30 日

	`ValidateNetworkSer viceUpdate` • API 更新：GetSolNet workOperation、 ListSolNe tworkOperations 和 UpdateSolNetworkInstance	
現有任務的新任務和新任務名稱	有新的任務可用。截至 2024 年 3 月 7 日，為了清楚起見，某些現有任務具有新的名稱。	2024 年 5 月 7 日
叢集的 Kubernetes 版本	AWS TNB 現在支援 Kubernete s 1.29 版來建立 Amazon EKS 叢集。	2024 年 4 月 10 日
支援網路介面 `security_ groups`	您可以將安全群組連接至 AWS.Networking. ENI節點。	2024 年 4 月 2 日
支援 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，請在 AWS.Compu te.EKSManagedNode 或 AWS.Compute.EKSSel fManagedNode 節點中新增 屬性。	2024 年 4 月 2 日
節點支援 `labels`	您可以在 AWS.Compu te.EKSManagedNode 或 AWS.Compute.EKSSel fManagedNode 節點中將節點標籤連接至節點群組。	2024 年 3 月 19 日
支援網路介面 `source_de st_check`	您可以指出是否要透過 AWS.Networking.ENI 節點啟用或停用網路介面來源/目的地檢查。	2024 年 1 月 25 日

支援具有自訂使用者資料的 Amazon EC2執行個體	您可以透過 AWS.Compute.UserData node 啟動具有自訂使用者資料的 Amazon EC2 執行個體。	2024 年 1 月 16 日
支援安全群組	AWS TNB 可讓您匯入安全群組 AWS 資源。	2024 年 1 月 8 日
已更新 的描述 network_interfaces	當 network_interfaces 屬性包含在 AWS.Compute.EKSManagedNode 或 AWS.Compute. EKSSelfManagedNode節點中時， AWS TNB 會在可用時ENIs從 multus_role 屬性取得與相關的許可，或從 node_role 屬性取得 的相關許可。	2023 年 12 月 18 日
支援私有叢集	AWS TNB 現在支援私有叢集。若要指示私有叢集，請將 access 屬性設定為 PRIVATE。	2023 年 12 月 11 日
叢集的 Kubernetes 版本	AWS TNB 現在支援 Kubernetes 1.28 版來建立 Amazon EKS 叢集。	2023 年 12 月 11 日
AWS TNB 支援置放群組	已新增 AWS.Compute.EKSManagedNode 和 AWS.Compute.EKSSelfManagedNode 節點定義的置放群組。	2023 年 12 月 11 日

AWS TNB 新增 的支援 IPv6	AWS TNB 現在支援使用IPv6基礎設施建立網路執行個體。檢查節點 AWS.Networking.VPC、AWS.Networking.Subnet、AWS.Networking.InternetGateway、AWS.Networking.SecurityGroupIngressRule、AWS.Networking.SecurityGroupEgressRule 和 AWS.Compute.EKS 的IPv6組態。我們也新增了節點 AWS.Networking.NATGateway 和 AWS.Networking.Route 進行NAT64組態。我們更新 AWS TNB了Amazon EKS節點群組的服務角色和服務 AWS TNB角色以取得IPv6許可。請參閱服務角色政策範例 。	2023 年 11 月 16 日
新增服務角色政策的 AWS TNB許可	我們已將 AWS TNB許可新增至 Amazon S3 的服務角色政策 AWS CloudFormation ，並啟用基礎設施實例化。	2023 年 10 月 23 日
AWS TNB 在更多 區域中啟動	AWS TNB 現可於亞太區域（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）區域使用。	2023 年 9 月 27 日
AWS.Compute 的標籤。EKSSelfManagedNode	AWS TNB 現在支援AWS.Compute.EKSSelfManagedNode 節點定義的標籤。	2023 年 8 月 22 日

AWS TNB 支援利用 的執行個體 IMDSv2	啟動執行個體時，您必須使用 IMDSv2。	2023 年 8 月 14 日
已更新 的許可 MultusRoleInlinePolicy	現在 MultusRoleInlinePolicy 包含 `ec2:DeleteNetworkInterface` 許可。	2023 年 8 月 7 日
叢集的 Kubernetes 版本	AWS TNB 現在支援 Kubernetes 1.27 版來建立 Amazon EKS 叢集。	2023 年 7 月 25 日
AWS.ComputeEKS.AuthRole	AWS TNB 支援 ， AuthRole 可讓您將IAM角色新增至 Amazon EKS叢集，`aws-authConfigMap` 讓使用者可以使用IAM角色存取 Amazon EKS叢集。	2023 年 7 月 19 日
AWS TNB 支援安全群組。	已將 AWS.Networking.SecurityGroup、AWS.Networking.SecurityGroupEgressRule 和 AWS.Networking.SecurityGroupIngressRule 新增至 NSD 範本。	2023 年 7 月 18 日
叢集的 Kubernetes 版本	AWS TNB 支援 Kubernetes 1.22 版到 1.26 版來建立 Amazon EKS叢集。 AWS TNB 不再支援 Kubernetes 1.21 版。	2023 年 5 月 11 日
AWS.Compute。EKSSelfManagedNode	您可以在區域內、 AWS 本機區域和 上建立自我管理的工作者節點 AWS Outposts。	2023 年 3 月 29 日
初始版本	這是使用者指南的第一個版本 AWS TNB。	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。