Installation et configuration de Catalyst 8000V sur le cloud STACKIT

Table des matières

- Introduction
- Conditions préalables
  - Exigences
  - Composants utilisés
- Informations générales
- Versions logicielles validées et types de VM
  - Versions logicielles
  - Saveurs VM STACKIT
  - Détails de l'hyperviseur signalés par les invités
- Configurer
  - 1. Installation de l'interface CLI STACKIT
  - 2. Définir l'environnement
  - 3. Préparer les entrées de déploiement
  - 4. Télécharger l'image C8000V
  - 5. Créer le volume de démarrage
  - 6. Création de réseaux et sécurité
  - 7. Créer des cartes réseau commandées
  - 8. Préparation de la configuration du jour 0
  - 9. Créez l'instance C8000V
  - 10. Vérification du déploiement
  - 11. Supprimer le déploiement
- Annexe
  - Commandes STACKIT utiles
  - Commandes C8000V utiles

Introduction

Ce document décrit les étapes nécessaires pour installer et activer Cisco Catalyst 8000V sur le cloud STACKIT appartenant à Schwarz Digits.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Un projet STACKIT existant et un opérateur authentifié autorisé à gérer des images, des volumes, des interfaces réseau et des serveurs.
- Une station de travail Linux ou macOS avec un shell compatible Bash. Les exemples de commandes de ce guide sont destinés à Linux et à macOS.
- Préfixes réseau approuvés, contrôles de sécurité, accès de gestion et IP publique facultative qui suit la conception STACKIT.
- Image C8000V QCOW2 autorisée pour la version sélectionnée de Cisco IOS® XE.
- Type de machine STACKIT qui répond aux exigences publiées en matière de CPU et de mémoire pour la version et l'ensemble de fonctionnalités sélectionnés.
- Taille du volume de démarrage et paramètre du microprogramme appropriés pour l'image C8000V sélectionnée.
- QEMU installé, y compris l'utilitaire qemu-img utilisé pour inspecter l'image source.
- stackit, jq et base64 sont disponibles dans l'environnement de commande.
- Un bootstrap du jour 0 préparé pour le mode autonome, contrôleur ou routage SD.

Le bootstrap peut contenir des mots de passe, des certificats ou des informations d'identité de périphérique. Stockez-le dans un endroit protégé. Le codage Base64 ne chiffre pas le fichier.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions logicielles suivantes :

- Version de Cisco IOS XE : 17.18.5
- Version des contrôleurs SD-WAN Cisco : 20.18.5, utilisé pour le mode Contrôleur (Cisco SD-WAN)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Cisco Catalyst 8000V est un routeur virtuel qui exécute Cisco IOS XE. Sur STACKIT, l'image C8000V est téléchargée en tant qu'image personnalisée et utilisée pour créer le volume de démarrage du serveur virtuel.

Lors de la création du serveur, STACKIT place les données utilisateur fournies sur un lecteur de configuration. C8000V lit le bootstrap pendant le démarrage initial. Le contenu du bootstrap dépend du mode de fonctionnement prévu :

- Le mode autonome utilise iosxe_config.txt.
- Le mode contrôleur utilise ciscosdwan_cloud_init.cfg généré par Cisco SD-WAN Manager.
- Le mode de routage SD utilise ciscosdwan_cloud_init.cfg généré par Cisco SD-WAN Manager pour le routage SD.

La même image, le même volume, la même interface et la même séquence de création de serveur sont utilisés pour les trois modes. Le contenu du bootstrap et la vérification post-déploiement diffèrent selon le mode.

Versions logicielles validées et types de VM

Le workflow de déploiement présenté dans ce document a été validé avec ces versions logicielles et les saveurs de machine virtuelle STACKIT. Cette ligne de base testée documente l'environnement des travaux pratiques ; il ne remplace pas les notes de version actuelles de Cisco Catalyst 8000V, les exigences de la plate-forme ou les conseils de dimensionnement spécifiques aux fonctionnalités.

Versions logicielles

- Version de Cisco IOS XE : 17.18.5
- Version des contrôleurs SD-WAN Cisco : 20.18.5, utilisé pour le mode Contrôleur (Cisco SD-WAN)

Saveurs VM STACKIT

Les saveurs de machine virtuelle STACKIT suivantes ont été incluses dans la validation :

Taille d'instance	vCPU	Mémoire (Go)
C3i.4	4	8
C3i.8	8	16

Taille d'instance	vCPU	Mémoire (Go)
C3i.16	16	32

Sélectionnez une version qui répond aux exigences en matière de processeur, de mémoire, de débit et de fonctionnalités pour le déploiement C8000V prévu.

Détails de l'hyperviseur signalés par les invités

L'invité C8000V validé a signalé l'environnement de virtualisation STACKIT :

<#root>

```
Router#

show platform software system hypervisor


Hypervisor:

KVM


Manufacturer:

STACKIT Cloud


Product Name:

OpenStack Nova


Serial Number: <INSTANCE_UUID>
UUID: <INSTANCE_UUID>
Image Variant: None

Router#
```

Configurer

1. Installation de l'interface CLI STACKIT

Ce guide utilise Homebrew pour installer l'interface de ligne de commande STACKIT. D'autres gestionnaires de paquets et méthodes d'installation sont disponibles. Pour plus d'informations, consultez le guide d'installation de STACKIT CLI.

```
brew tap stackitcloud/tap
brew install --cask stackit
```

2. Définir l'environnement

Configurez l'environnement CLI STACKIT avant d'exécuter les commandes de déploiement.

2.1 Connexion et authentification

Connectez-vous à votre compte STACKIT.

```
stackit auth login
```

Effectuez l'authentification dans votre navigateur. Une fois l'authentification réussie, l'interface de ligne de commande confirme que vous êtes connecté.

2.2 Définition du projet

Définissez le projet par défaut pour les commandes restantes.

```
stackit config set --project-id xxxxxxxx-yyyy-zzzz-aaaa-bbbbbbbbbbbb
```

Si vous ne connaissez pas l'ID de projet, répertoriez les projets disponibles.

```
stackit project list
```

3. Préparer les entrées de déploiement

Collectez les valeurs requises par le déploiement. L'exemple crée deux réseaux et deux cartes réseau : la première pour la gestion ou le transport et la seconde pour le service ou le trafic de données.

```
export PROJECT_ID="<PROJECT_ID>"
export REGION="<REGION>"
export AVAILABILITY_ZONE="<AVAILABILITY_ZONE>"
export MACHINE_TYPE="<MACHINE_TYPE>"

export SERVER_NAME="<C8000V_NAME>"
export IMAGE_NAME="<C8000V_IMAGE_NAME>"
export IMAGE_FILE="/path/to/<C8000V_IMAGE>.qcow2"
export IMAGE_MIN_DISK_SIZE="<MINIMUM_DISK_GB>"
export BOOT_VOLUME_SIZE="<BOOT_VOLUME_GB>"
export IMAGE_UEFI="<true_or_false>"

export MGMT_NETWORK_NAME="${SERVER_NAME}-mgmt"
export DATA_NETWORK_NAME="${SERVER_NAME}-data"
export MGMT_IPV4_PREFIX="<MANAGEMENT_IPV4_PREFIX>"
export DATA_IPV4_PREFIX="<DATA_IPV4_PREFIX>"
export TRUSTED_SOURCE_CIDR="<TRUSTED_SOURCE_IP_OR_NETWORK>/<PREFIX_LENGTH>"
```

Utilisez des préfixes non chevauchants provenant de la conception de réseau approuvée. Spécifiez TRUSTED_SOURCE_CIDR dans la notation CIDR, y compris la longueur du préfixe. Utilisez /32 pour autoriser une adresse IPv4. Les exemples de règles de gestion répertoriés autorisent SSH uniquement à partir de TRUSTED_SOURCE_CIDR.F.

4. Télécharger l'image C8000V

4.1 Inspection de l'image source

Vérifiez que le fichier est l'image autorisée pour la version prévue. Enregistrez le nom du fichier et la somme de contrôle dans l'enregistrement de déploiement.

```
qemu-img info "$IMAGE_FILE"

if command -v shasum >/dev/null 2>&1; then
  shasum -a 256 "$IMAGE_FILE"
else
  sha256sum "$IMAGE_FILE"
fi
```

4.2 Création de l'image personnalisée STACKIT

Définissez IMAGE_UEFI sur true pour une image EFI ou false pour une image BIOS. Sélectionnez le paramètre spécifié pour l'image C8000V téléchargée. Définissez la taille minimale du disque en fonction de l'image sélectionnée et de la documentation actuelle du produit.

```
IMAGE_RESPONSE="$(
  stackit image create \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --name "$IMAGE_NAME" \
    --disk-format qcow2 \
    --uefi="$IMAGE_UEFI" \
    --min-disk-size "$IMAGE_MIN_DISK_SIZE" \
    --local-file-path "$IMAGE_FILE" \
    --output-format json
)"

export IMAGE_ID="$(printf '%s' "$IMAGE_RESPONSE" | jq -r '.id')"
printf 'IMAGE_ID=%s\n' "$IMAGE_ID"
```

4.3 Sondage jusqu'à ce que l'image soit disponible

```
stackit image describe "$IMAGE_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION" \
  --output-format json |
  jq '{id, name, status, diskFormat}'
```

Continuez lorsque l'état de l'image est DISPONIBLE.

5. Créer le volume de démarrage

Créez un nouveau volume de démarrage à partir de l'image personnalisée. La taille du volume doit répondre aux exigences de disque de l'image sélectionnée.

```
BOOT_VOLUME_RESPONSE="$(
  stackit volume create \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --availability-zone "$AVAILABILITY_ZONE" \
    --name "${SERVER_NAME}-boot" \
    --source-id "$IMAGE_ID" \
    --source-type image \
    --size "$BOOT_VOLUME_SIZE" \
    --output-format json
)"
```

```
export BOOT_VOLUME_ID="$(printf '%s' "$BOOT_VOLUME_RESPONSE" | jq -r '.id')"
printf 'BOOT_VOLUME_ID=%s\n' "$BOOT_VOLUME_ID"
```

Interrogez le volume jusqu'à ce que son état soit DISPONIBLE.

```
stackit volume describe "$BOOT_VOLUME_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION" \
  --output-format json |
  jq '{id, name, status, size}'
```

6. Création de réseaux et sécurité

Utiliser les réseaux et les groupes de sécurité existants lorsqu'ils répondent déjà à la conception approuvée. Sinon, créez les ressources comme indiqué.

6.1 Création ou identification des réseaux

Répertoriez les réseaux existants avant d’en créer de nouveaux.

```
stackit network list \
  --project-id "$PROJECT_ID" \
  --region "$REGION"
```

S'il n'existe pas de réseaux appropriés, créez le réseau de gestion ou de transport et le réseau de service ou de données.

```
MGMT_NETWORK_RESPONSE="$(
  stackit network create \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --name "$MGMT_NETWORK_NAME" \
    --ipv4-prefix "$MGMT_IPV4_PREFIX" \
    --output-format json
)"

DATA_NETWORK_RESPONSE="$(
  stackit network create \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --name "$DATA_NETWORK_NAME" \
    --ipv4-prefix "$DATA_IPV4_PREFIX" \
```

```
    --output-format json
)"

export MGMT_NETWORK_ID="$(printf '%s' "$MGMT_NETWORK_RESPONSE" | jq -r '.id')"
export DATA_NETWORK_ID="$(printf '%s' "$DATA_NETWORK_RESPONSE" | jq -r '.id')"
printf 'MGMT_NETWORK_ID=%s\nDATA_NETWORK_ID=%s\n' "$MGMT_NETWORK_ID" "$DATA_NETWORK_ID"
```

Si des réseaux approuvés existent déjà, définissez MGMT_NETWORK_ID et DATA_NETWORK_ID sur leurs ID respectifs.

6.2 Création du groupe de sécurité de gestion

Créez un groupe de sécurité avec état pour la carte réseau de gestion.

```
SECURITY_GROUP_RESPONSE="$(
  stackit security-group create \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --name "${SERVER_NAME}-mgmt" \
    --description "Management access for ${SERVER_NAME}" \
    --stateful \
    --output-format json
)"

export MGMT_SECURITY_GROUP_ID="$(printf '%s' "$SECURITY_GROUP_RESPONSE" | jq -r '.id')"
printf 'MGMT_SECURITY_GROUP_ID=%s\n' "$MGMT_SECURITY_GROUP_ID"
```

Autoriser SSH depuis un CIDR source approuvé

```
stackit security-group rule create \
  --project-id "$PROJECT_ID" \
  --region "$REGION" \
  --security-group-id "$MGMT_SECURITY_GROUP_ID" \
  --direction ingress \
  --ether-type IPv4 \
  --protocol-name tcp \
  --port-range-min 22 \
  --port-range-max 22 \
  --ip-range "$TRUSTED_SOURCE_CIDR" \
  --description "SSH from approved source"
```

Ajoutez le contrôle HTTPS, NETCONF, ICMP ou Cisco SD-WAN et les règles du plan de données uniquement lorsque le déploiement l'exige. Limitez chaque règle d'entrée à la source pratique la plus étroite.

7. Créer des cartes réseau commandées

Créez les cartes réseau avant de créer le serveur. Connectez le groupe de sécurité de gestion à la carte réseau 1 et soumettez les ID de carte réseau résultants dans l'ordre d'interface C8000V prévu.

7.1 Création des cartes réseau 1 et 2

```
MGMT_NIC_RESPONSE="$(
  stackit network-interface create \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --network-id "$MGMT_NETWORK_ID" \
    --name "${SERVER_NAME}-mgmt" \
    --nic-security \
    --security-groups "$MGMT_SECURITY_GROUP_ID" \
    --output-format json
)"

DATA_NIC_RESPONSE="$(
  stackit network-interface create \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --network-id "$DATA_NETWORK_ID" \
    --name "${SERVER_NAME}-data" \
    --nic-security \
    --output-format json
)"

export MGMT_NIC_ID="$(printf '%s' "$MGMT_NIC_RESPONSE" | jq -r '.id')"
export DATA_NIC_ID="$(printf '%s' "$DATA_NIC_RESPONSE" | jq -r '.id')"
printf 'MGMT_NIC_ID=%s\nDATA_NIC_ID=%s\n' "$MGMT_NIC_ID" "$DATA_NIC_ID"
```

L'exemple de création de serveur soumet MGMT_NIC_ID en premier pour GigabitEthernet1 et DATA_NIC_ID en second pour GigabitEthernet2. Les ID de carte réseau supplémentaires correspondent aux interfaces IOS XE suivantes dans l'ordre soumis. Vérifiez le mappage résultant de l’IOS XE après le démarrage du serveur. Enregistrez l’ID de carte réseau, l’ID réseau, l’adresse MAC, l’adresse IP attribuée et l’interface IOS XE prévue pour chaque carte réseau.

7.2 Association d'une adresse IP publique facultative

Une adresse IP publique est associée à une carte réseau spécifique. Lorsque la gestion externe est requise, associez-la à la carte réseau 1.

Omettez cette étape lorsque l'accès à la gestion est fourni via un chemin privé.

```
PUBLIC_IP_RESPONSE="$(
  stackit public-ip create \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --associated-resource-id "$MGMT_NIC_ID" \
    --output-format json
)"

export PUBLIC_IP_ID="$(printf '%s' "$PUBLIC_IP_RESPONSE" | jq -r '.id')"
printf '%s\n' "$PUBLIC_IP_RESPONSE" | jq '{id, ip, associatedResourceId}'
```

8. Préparation de la configuration du jour 0

8.1 Sélection du fichier d'amorçage

Pour le mode autonome, créez iosxe_config.txt avec une commande de configuration IOS XE par ligne.

Pour le mode contrôleur, générez un bootstrap Cloud-Init en mode contrôleur dans Cisco SD-WAN Manager et conservez le nom de fichier ciscosdwan_cloud_init.cfg.

Pour le mode de routage SD, générez un bootstrap de mode de routage SD Cloud-Init dans Cisco SD-WAN Manager et conservez le nom de fichier ciscosdwan_cloud_init.cfg.

Ne modifiez pas manuellement un bootstrap généré par Cisco SD-WAN Manager. Un fichier généré contient des informations d'intégration et d'identité du périphérique et doit être protégé en tant qu'artefact portant des informations d'identification.

8.2 Exemple de mode autonome

```
hostname <ROUTER_HOSTNAME>
!
ip domain name <DOMAIN_NAME>
!
username <ADMIN_USER> privilege 15 secret <ADMIN_SECRET>
enable secret <ENABLE_SECRET>
!
interface GigabitEthernet1
 description Management
 ip address dhcp
 no shutdown
!
interface GigabitEthernet2
 description Service
```

```
 no ip address
 no shutdown
!
crypto key generate rsa modulus 2048
ip ssh version 2
!
line vty 0 4
 login local
 transport input ssh
!
end
```

Définissez le chemin d'accès au bootstrap sélectionné et limitez l'accès au fichier.

```
export BOOTSTRAP_FILE="/path/to/<BOOTSTRAP_FILE>"
chmod 600 "$BOOTSTRAP_FILE"
```

8.3 Encodage des données d'amorçage

```
export USER_DATA_B64="$(base64 < "$BOOTSTRAP_FILE" | tr -d '\r\n')"
```

9. Créez l'instance C8000V

9.1 Création de la demande de création de serveur

Les paramètres spécifiques au C8000V sont configDrive : true, le bootstrap encodé en base64 dans userData, le volume d'amorçage basé sur l'image et les ID de carte réseau commandés.

```
jq -n \
  --arg name "$SERVER_NAME" \
  --arg machineType "$MACHINE_TYPE" \
  --arg availabilityZone "$AVAILABILITY_ZONE" \
  --arg bootVolumeId "$BOOT_VOLUME_ID" \
  --arg mgmtNicId "$MGMT_NIC_ID" \
  --arg dataNicId "$DATA_NIC_ID" \
  --arg userData "$USER_DATA_B64" \
  '{
    name: $name,
    machineType: $machineType,
    availabilityZone: $availabilityZone,
    configDrive: true,
    userData: $userData,
    bootVolume: {
```

```
      source: {
        type: "volume",
        id: $bootVolumeId
      }
    },
    networking: {
      nicIds: [$mgmtNicId, $dataNicId]
    },
    labels: {
      product: "cisco-c8000v"
    }
  }' > c8000v-server.json
```

Vérifiez la demande sans afficher le bootstrap :

```
jq 'del(.userData)' c8000v-server.json
```

9.2 Création du serveur

Utilisez le terminal STACKIT IaaS v2 actuel documenté pour l'environnement cible. Remplacez l'URL de l'API STACKIT si nécessaire.

```
SERVER_RESPONSE="$(
  stackit curl \
    -X POST \
    "https://<STACKIT API URL>/v2/projects/${PROJECT_ID}/regions/${REGION}/servers" \
    -H "Content-Type: application/json" \
    --data @c8000v-server.json \
    --fail
)"

export SERVER_ID="$(printf '%s' "$SERVER_RESPONSE" | jq -r '.id')"
printf '%s\n' "$SERVER_RESPONSE" | jq '{id, name, status, configDrive}'
```

Exemple de réponse :

```
{
  "id": "<SERVER_ID>",
  "name": "<C8000V_NAME>",
  "status": "CREATING",
  "configDrive": true
}
```

Effectuez un sondage jusqu'à ce que le serveur atteigne l'état d'exécution attendu.

```
stackit server describe "$SERVER_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION" \
  --output-format json |
  jq '{id, name, status, powerStatus}'
```

10. Vérification du déploiement

10.1 Vérification des ressources STACKIT

Confirmez que :

- Le serveur s'exécute dans le projet, la région et la zone de disponibilité prévus.
- Le volume de démarrage prévu est joint.
- Les cartes réseau attendues sont connectées.
- L'adresse IP publique facultative est associée à la carte réseau voulue.
- Les contrôles d'accès appliqués correspondent à la conception approuvée.

10.2 Vérification de Cisco IOS XE

Connectez-vous via le chemin de gestion approuvé et exécutez les vérifications appropriées à la version et au mode sélectionnés.

```
show version
show platform software vnic-if interface-mapping
show ip interface brief
show ip route
```

Vérifiez que le mode prévu est actif, que la configuration Day-0 a été appliquée et que les interfaces IOS XE correspondent à la commande de carte réseau envoyée.

Commandes supplémentaires pour le mode contrôleur :

```
show sdwan control connections
show sdwan control local-properties
show sdwan system status
```

Commandes supplémentaires pour le mode de routage SD :

```
show sd-routing control connections summary
show sd-routing control local-properties summary
show sd-routing system status
```

Après vérification, faites pivoter les informations d'identification de mise en service temporaire et supprimez le fichier de demande local et la variable codée.

```
rm -f c8000v-server.json
unset USER_DATA_B64
```

11. Supprimer le déploiement

Les commandes de nettoyage de cette section suppriment définitivement des ressources. Vérifiez que le déploiement n'est plus nécessaire et que chaque ID de ressource appartient uniquement à ce déploiement C8000V. Ne supprimez pas les réseaux, groupes de sécurité, images ou autres ressources existants ou partagés.

Les exemples de nettoyage conservent des invites de confirmation interactives. Vérifiez la ressource affichée dans chaque invite de confirmation de l'interface de ligne de commande STACKIT avant d'approuver l'opération.

Pour le mode contrôleur ou routage SD, effectuez toute mise hors service côté contrôleur requise avant de supprimer les ressources STACKIT.

11.1 Préparation du retrait

Vérifiez les ID de ressource enregistrés avant de continuer.

```
printf 'SERVER_ID=%s\n' "$SERVER_ID"
printf 'PUBLIC_IP_ID=%s\n' "${PUBLIC_IP_ID:-not-created}"
printf 'MGMT_NIC_ID=%s\nDATA_NIC_ID=%s\n' "$MGMT_NIC_ID" "$DATA_NIC_ID"
printf 'BOOT_VOLUME_ID=%s\n' "$BOOT_VOLUME_ID"
printf 'MGMT_SECURITY_GROUP_ID=%s\n' "$MGMT_SECURITY_GROUP_ID"
printf 'MGMT_NETWORK_ID=%s\nDATA_NETWORK_ID=%s\n' "$MGMT_NETWORK_ID" "$DATA_NETWORK_ID"
printf 'IMAGE_ID=%s\n' "$IMAGE_ID"
```

Si le déploiement a une adresse IP publique, détachez-la avant de supprimer le serveur.

```
if [ -n "${PUBLIC_IP_ID:-}" ]; then
  stackit server public-ip detach "$PUBLIC_IP_ID" \
    --server-id "$SERVER_ID" \
    --project-id "$PROJECT_ID" \
    --region "$REGION"
fi
```

11.2 Suppression des ressources de déploiement

Supprimez d'abord le serveur. Continuer lorsque le serveur n'apparaît plus dans la liste des serveurs.

```
stackit server delete "$SERVER_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION"

stackit server list \
  --project-id "$PROJECT_ID" \
  --region "$REGION"
```

Supprimez l'adresse IP publique facultative, les deux cartes réseau ordonnées et le volume de démarrage.

```
if [ -n "${PUBLIC_IP_ID:-}" ]; then
  stackit public-ip delete "$PUBLIC_IP_ID" \
    --project-id "$PROJECT_ID" \
    --region "$REGION"
fi

stackit network-interface delete "$MGMT_NIC_ID" \
  --network-id "$MGMT_NETWORK_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION"

stackit network-interface delete "$DATA_NIC_ID" \
  --network-id "$DATA_NETWORK_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION"

stackit volume delete "$BOOT_VOLUME_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION"
```

Si le groupe de sécurité de gestion et les réseaux ont été créés uniquement pour ce déploiement, supprimez-les une fois que leurs ressources dépendantes ont disparu.

```
stackit security-group delete "$MGMT_SECURITY_GROUP_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION"

stackit network delete "$MGMT_NETWORK_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION"

stackit network delete "$DATA_NETWORK_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION"
```

Conservez l'image personnalisée lorsqu'elle peut être utilisée pour un autre déploiement C800V. Sinon, supprimez-le après la suppression de tous les volumes de démarrage dépendants.

```
stackit image delete "$IMAGE_ID" \
  --project-id "$PROJECT_ID" \
  --region "$REGION"
```

11.3 Vérification du nettoyage

Répertoriez les ressources restantes et confirmez que les ID supprimés ne sont plus présents. Les ressources partagées qui ont été conservées intentionnellement peuvent toujours apparaître.

```
stackit server list --project-id "$PROJECT_ID" --region "$REGION"
stackit public-ip list --project-id "$PROJECT_ID" --region "$REGION"
stackit volume list --project-id "$PROJECT_ID" --region "$REGION"
stackit security-group list --project-id "$PROJECT_ID" --region "$REGION"
stackit network list --project-id "$PROJECT_ID" --region "$REGION"
stackit image list --project-id "$PROJECT_ID" --region "$REGION"
```

Vérifiez que tous les artefacts de bootstrap locaux conservés après le déploiement ont été traités conformément à la stratégie de rétention des informations d'identification de l'organisation.

Annexe

Commandes STACKIT utiles

```
# List servers
stackit server list --project-id "$PROJECT_ID" --region "$REGION"

# List volumes
stackit volume list --project-id "$PROJECT_ID" --region "$REGION"

# List custom images
stackit image list --project-id "$PROJECT_ID" --region "$REGION"

# List network interfaces
stackit network-interface list --project-id "$PROJECT_ID" --region "$REGION"
```

Commandes C8000V utiles

```
show version
show platform software vnic-if interface-mapping
show ip interface brief
show ip route
show logging
```

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.