

Cisco Catalyst 8000V installeren en configureren op STACKIT Cloud

Inhoud

[Inleiding](#)

[Achtergrond](#)

[Voorwaarden](#)

[Vereisten](#)

[Gevalideerde softwarereleases en VM-typen](#)

[Softwarereleases](#)

[STACKIT VM-smaken](#)

[Gastgerapporteerde hypervisorgegevens](#)

[Configureren](#)

[1. Installeer de STACKIT CLI](#)

[2. Stel het milieu in](#)

[3. Bereid de inputs voor de implementatie voor](#)

[4. Upload het C8000V-image](#)

[5. Maak het opstartvolume](#)

[6. Netwerken en beveiliging maken](#)

[7. Bestelde NIC's maken](#)

[8. Bereid de configuratie van Dag 0 voor](#)

[9. Maak de C8000V-instantie](#)

[10. Controleer de implementatie](#)

[11. Verwijder de implementatie](#)

[Bijlage](#)

[Nuttige STACKIT-opdrachten](#)

[Nuttige C8000V-opdrachten](#)

Inleiding

In dit document worden de stappen beschreven die nodig zijn om [Cisco Catalyst 8000V](#) op de [STACKIT](#)-cloud van Schwarz Digits te installeren en te installeren.

Achtergrond

Cisco Catalyst 8000V is een virtuele router die Cisco IOS XE draait. Op STACKIT wordt het C8000V-image geüpload als een aangepaste image en gebruikt om het opstartvolume voor de

virtuele server te maken.

Tijdens het maken van de server plaatst STACKIT de meegeleverde gebruikersgegevens op een configuratiestation. C8000V leest de bootstrap tijdens het opstarten. De inhoud van de bootstrap is afhankelijk van de beoogde bedrijfsmodus:

- In de autonome modus wordt `iosxe_config.txt` gebruikt.
- In de Controllermodus wordt `ciscosdwan_cloud_init.cfg` gebruikt, gegenereerd door Cisco SD-WAN Manager.
- SD-Routing modus maakt gebruik van `ciscosdwan_cloud_init.cfg` gegenereerd door Cisco SD-WAN Manager voor SD-Routing.

Voor alle drie de modi worden dezelfde sequentie voor het maken van images, volumes, interfaces en servers gebruikt. De bootstrap-inhoud en verificatie na implementatie verschillen per modus.

Voorwaarden

Vereisten

- Een bestaand STACKIT-project en een geverifieerde operator met toestemming om images, volumes, netwerkinterfaces en servers te beheren.
- Een Linux- of macOS-werkstation met een Bash-compatibele shell. De opdrachtvoorbeelden in deze handleiding zijn bedoeld voor Linux en macOS.
- Goedgekeurde netwerkvoorvoegsels, beveiligingscontroles, beheertoegang en een optioneel openbaar IP die het STACKIT-ontwerp volgen.
- Een geautoriseerd C8000V QCOW2-image voor de geselecteerde Cisco IOS XE-release.
- Een STACKIT-machinetype dat voldoet aan de gepubliceerde C8000V-CPU en geheugenvereisten voor de geselecteerde release- en functieset.
- Een grootte van het opstartvolume en een firmware-instelling die geschikt is voor het geselecteerde C8000V-image.
- QEMU geïnstalleerd, inclusief het `qemu-img`-hulpprogramma dat wordt gebruikt om het bronimage te inspecteren.
- Stackit, JQ en Base64 beschikbaar in de opdrachtomgeving.
- Een Day-0 bootstrap die is voorbereid voor de modus Autonomous, Controller of SD-Routing.

De bootstrap kan wachtwoorden, certificaten of apparaatidentiteitsgegevens bevatten. Bewaar het op een beschermde locatie. Base64-codering codeert het bestand niet.

Gevalideerde software releases en VM-typen

De implementatieworkflow in dit document is gevalideerd met de volgende software releases en STACKIT VM-smaken. Deze geteste basislijn documenteert de laboratoriumomgeving; deze vervangt niet de huidige Cisco Catalyst 8000V-releasenota's, platformvereisten of functiespecifieke maatrichtlijnen.

Software releases

- Cisco IOS XE release: 17.18.5
- Cisco SD-WAN Controllers release: 20.18.5, gebruikt voor Controller (Cisco SD-WAN) modus

STACKIT VM-smaken

De volgende STACKIT VM-smaken zijn opgenomen in de validatie:

Instantiegrootte	vCPU's	Geheugen (GiB)
C3I.4	4	8
C3I.8	8	16
C3I.16	16	32

Selecteer een smaak die voldoet aan de vereisten voor CPU, geheugen, doorvoer en functionaliteit voor de beoogde C8000V-implementatie.

Gastgerapporteerde hypervisorgegevens

De gevalideerde C8000V-gast rapporteerde de STACKIT-virtualisatieomgeving als volgt:

```
<#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#

Configureren

1. Installeer de STACKIT CLI

Deze handleiding gebruikt Homebrew om de STACKIT CLI te installeren. Andere pakketbeheerders en installatiemethoden zijn beschikbaar. Zie de installatiehandleiding van de STACKIT CLI voor meer informatie.

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

2. Stel het milieu in

Configureer de STACKIT CLI-omgeving voordat u de implementatieopdrachten uitvoert.

2.1 Log in en verifieer

Log in op je STACKIT account.

```
stackit auth login
```

Voltooi de authenticatie in uw browser. Nadat de verificatie is voltooid, bevestigt de CLI dat u bent ingelogd.

2.2 Stel het project in

Stel het standaardproject in voor de resterende opdrachten.

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

Als u de project-ID niet kent, geeft u de beschikbare projecten op.

```
stackit project list
```

3. Bereid de inputs voor de implementatie voor

Verzamel de waarden die vereist zijn voor de implementatie. In het voorbeeld worden twee netwerken en twee NIC's gemaakt: de eerste voor beheer of transport en de tweede voor service- of dataverkeer.

```
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>"
```

Gebruik niet-overlappende voorvoegsels uit het goedgekeurde netwerkontwerp. Geef TRUSTED_SOURCE_CIDR op in de CIDR-notatie, inclusief de lengte van het voorvoegsel. Gebruik /32 om één IPv4-adres toe te staan. De onderstaande voorbeelden van beheerregels staan alleen SSH van TRUSTED_SOURCE_CIDR toe.

4. Upload het C8000V-image

4.1 Inspecteer de bronafbeelding

Controleer of het bestand de geautoriseerde afbeelding is voor de beoogde release. Noteer de bestandsnaam en de controlesom in het implementatierecord.

```
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 De aangepaste STACKIT-afbeelding maken

Stel IMAGE_UEFI in op true voor een EFI-afbeelding of false voor een BIOS-afbeelding. Selecteer de instelling die is opgegeven voor het gedownloade C8000V-image. Stel de minimale schijfgrootte in op basis van de geselecteerde afbeelding en de huidige productdocumentatie.

```
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 Poll totdat de afbeelding beschikbaar is

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

Ga door wanneer de status van het image BESCHIKBAAR is.

5. Maak het opstartvolume

Maak een nieuw opstartvolume aan in de aangepaste afbeelding. De volumegrootte moet voldoen aan de schijfvereisten van de geselecteerde afbeelding.

```
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"
```

Poll het volume totdat de status BESCHIKBAAR is.

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

6. Netwerken en beveiliging maken

Gebruik bestaande netwerken en beveiligingsgroepen wanneer ze al voldoen aan het goedgekeurde ontwerp. Maak anders de bronnen aan zoals hieronder wordt weergegeven.

6.1 De netwerken aanmaken of identificeren

Maak een lijst van de bestaande netwerken voordat u nieuwe maakt.

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

Als er geen geschikte netwerken bestaan, maakt u het beheer- of transportnetwerk en het service- of datanetwerk.

```
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"
```

Als er al goedgekeurde netwerken bestaan, stelt u in plaats daarvan MGMT_NETWORK_ID en DATA_NETWORK_ID in op hun ID's.

6.2 De beveiligingsgroep voor beheer maken

Maak een stateful security group voor de beheer-NIC.


```
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"
```

SSH toestaan van een goedgekeurde bron CIDR.

```
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"
```

Voeg HTTPS-, NETCONF-, ICMP- of Cisco SD-WAN-besturings- en gegevensvlakregels alleen toe wanneer dit vereist is voor de implementatie. Beperk elke ingangsregel tot de smalste praktische bron.

7. Bestelde NIC's maken

Maak de NIC's voordat u de server maakt. Sluit de beheerbeveiligingsgroep aan op NIC 1 en dien de resulterende NIC-ID's in de beoogde C8000V-interfacevolgorde in.

7.1 Maak NIC 1 en NIC 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"

```

Het voorbeeld voor het maken van de server dient eerst MGMT_NIC_ID in voor GigabitEthernet1 en DATA_NIC_ID seconde voor GigabitEthernet2. Extra NIC-ID's koppelen aan volgende IOS XE-interfaces in de opgegeven volgorde. Controleer de resulterende toewijzing van IOS XE nadat de server is opgestart.

Noteer de NIC-ID, het netwerk-ID, het MAC-adres, het toegewezen IP-adres en de beoogde IOS XE-interface voor elke NIC.

7.2 Een optioneel openbaar IP koppelen

Een openbare IP is gekoppeld aan een specifieke NIC. Als extern beheer vereist is, koppelt u dit aan NIC 1.

Laat deze stap weg wanneer beheertoegang via een privépad wordt geboden.

```

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. Bereid de configuratie van Dag 0 voor

8.1 Selecteer het bootstrap-bestand

Voor de autonome modus maakt u iosxe_config.txt met één IOS XE-configuratieopdracht per regel.

Voor de Controllermodus genereert u een Controller-modus Cloud-Init-bootstrap in Cisco SD-WAN Manager en behoudt u de bestandsnaam ciscosdwan_cloud_init.cfg.

Voor SD-Routing modus, genereer een SD-Routing-modus Cloud-Init bootstrap in Cisco SD-WAN Manager en behoud de bestandsnaam ciscosdwan_cloud_init.cfg.

Bewerk een door Cisco SD-WAN Manager gegenereerde bootstrap niet handmatig. Een gegenereerd bestand bevat informatie over de identiteit van het apparaat en onboarding en moet worden beschermd als een artefact met een referentie.

8.2 Voorbeeld van de autonome modus

```
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
```

Stel het pad in naar de geselecteerde bootstrap en beperk de toegang tot het bestand.

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

8.3 Codeer de bootstrap

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

9. Maak de C8000V-instantie

9.1 Stel het verzoek voor het maken van de server samen

De C8000V-specifieke instellingen zijn configDrive: true, de base64-gecodeerde bootstrap in userData, het op afbeeldingen gebaseerde opstartvolume en de bestelde NIC-ID's.

```
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
```

Controleer het verzoek zonder de bootstrap weer te geven:

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

9.2 De server maken

Gebruik het huidige STACKIT IaaS v2-eindpunt dat is gedocumenteerd voor de doelomgeving. Vervang indien nodig de STACKIT API URL.

```
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}'
```

Voorbeeld van een reactie:

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

Poll totdat de server de verwachte actieve status heeft bereikt.

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

10. Controleer de implementatie

10.1 Controleer de STACKIT-middelen

Bevestig dat:

- de server draait in de beoogde project-, regio- en beschikbaarheidszone;
- het beoogde opstartvolume is aangesloten;
- de verwachte NIC's zijn bijgevoegd;
- het optionele openbare IP is gekoppeld aan de beoogde NIC; en
- De toegepaste toegangscontroles komen overeen met het goedgekeurde ontwerp.

10.2 Controleer Cisco IOS XE

Maak verbinding via het goedgekeurde beheerpad en voer de controles uit die passen bij de geselecteerde release- en releasemodus.

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

Bevestig dat de beoogde modus actief is, dat de Day-0-configuratie is toegepast en dat de IOS XE-interfaces overeenkomen met de opgegeven NIC-volgorde.

Extra opdrachten voor Controllermodus:

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

Extra opdrachten voor SD-routeringsmodus:

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

Na verificatie roteert u de tijdelijke provisioningreferenties en verwijdt u het lokale aanvraagbestand en de gecodeerde variabele.

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

```
unset USER_DATA_B64
```

11. Verwijder de implementatie

De opruimopdrachten in deze sectie verwijderen bronnen permanent. Bevestig dat de implementatie niet langer vereist is en dat elke resource-ID alleen tot deze C8000V-implementatie behoort. Verwijder geen bestaande of gedeelde netwerken, beveiligingsgroepen, images of andere bronnen.

In de opruimvoorbeelden blijven interactieve bevestigingsverzoeken behouden. Controleer de bron die wordt weergegeven in elke bevestigingsprompt van de STACKIT CLI voordat u de bewerking goedkeurt.

Voor de Controller- of SD-routeringsmodus moet u de vereiste uitbedrijfsname aan de controllerzijde voltooien voordat u de STACKIT-bronnen verwijderd.

11.1 Voorbereiden op verwijdering

Controleer de opgenomen resource-ID's voordat u doorgaat.

```
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"
```

Als de implementatie een openbaar IP-adres heeft, maakt u het los voordat u de server verwijderd.

```
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 Verwijder de implementatiemiddelen

Verwijder eerst de server. Doorgaan nadat de server niet meer in de serverlijst wordt weergegeven.

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

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

Verwijder de optionele openbare IP, de twee bestelde NIC's en het opstartvolume.

```
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"
```

Als de beheerbeveiligingsgroep en -netwerken alleen voor deze implementatie zijn gemaakt, verwijdert u deze nadat de afhankelijke bronnen zijn verdwenen.

```
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"
```


Bewaar het aangepaste image wanneer het wordt gebruikt voor een andere C8000V-implementatie. Anders verwijdert u deze nadat alle afhankelijke opstartvolumes zijn verwijderd.

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

11.3 Controleer de schoonmaak

Maak een lijst van de resterende bronnen en bevestig dat de verwijderde ID's niet meer aanwezig zijn. Gedeelde middelen die opzettelijk werden behouden, moeten nog steeds verschijnen.

```
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"
```

Bevestig dat alle lokale bootstrap-artefacten die na de implementatie zijn bewaard, zijn verwerkt volgens het beleid voor het bewaren van referenties van de organisatie.

Bijlage

Nuttige STACKIT-opdrachten

```
# 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"
```

Nuttige C8000V-opdrachten

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

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.