

Installation und Konfiguration von Cisco Catalyst 8000V in STACKIT Cloud

Inhalt

[Einleitung](#)

[Hintergrund](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Validierte Softwareversionen und VM-Typen](#)

[Softwareversionen](#)

[STACKIT VM-Produktvarianten](#)

[Vom Gast gemeldete Hypervisordetails](#)

[Konfigurieren](#)

[1. Installation der STACKIT-CLI](#)

[2. Umgebung einstellen](#)

[3. Vorbereiten der Bereitstellungsdaten](#)

[4. Laden Sie das C8000V-Image hoch.](#)

[5. Boot-Volume erstellen](#)

[6. Erstellen Sie Netzwerke und Sicherheit](#)

[7. Erstellen Sie die bestellten NICs.](#)

[8. Vorbereiten der Day-0-Konfiguration](#)

[9. Erstellen Sie die C8000V-Instanz](#)

[10. Überprüfen der Bereitstellung](#)

[11. Entfernen Sie die Bereitstellung](#)

[Anhang](#)

[Nützliche STACKIT-Befehle](#)

[Nützliche C8000V-Befehle](#)

Einleitung

Dieses Dokument beschreibt die erforderlichen Schritte zur Installation und Aktivierung von [Cisco Catalyst 8000V](#) in der [STACKIT](#)-Cloud, die zu Schwarz Digits gehört.

Hintergrund

Cisco Catalyst 8000V ist ein virtueller Router, auf dem Cisco IOS XE ausgeführt wird. Auf STACKIT wird das C8000V-Image als benutzerdefiniertes Image hochgeladen und zum Erstellen

des Boot-Volumes für den virtuellen Server verwendet.

Während der Servererstellung legt STACKIT die bereitgestellten Benutzerdaten auf einem Konfigurationslaufwerk ab. C8000V liest das Bootstrap während des ersten Bootvorgangs. Der Bootstrap-Inhalt hängt von der beabsichtigten Betriebsart ab:

- Im autonomen Modus wird `iosxe_config.txt` verwendet.
- Im Controller-Modus wird `ciscosdwan_cloud_init.cfg` verwendet, das vom Cisco SD-WAN Manager generiert wird.
- Im SD-Routing-Modus wird `ciscosdwan_cloud_init.cfg` verwendet, das vom Cisco SD-WAN Manager für SD-Routing generiert wurde.

Für alle drei Modi werden das gleiche Bild, die gleiche Lautstärke, die gleiche Schnittstelle und die gleiche Sequenz zum Erstellen von Servern verwendet. Der Bootstrap-Inhalt und die Überprüfung nach der Bereitstellung unterscheiden sich je nach Modus.

Voraussetzungen

Anforderungen

- Ein bestehendes STACKIT-Projekt und ein authentifizierter Operator mit der Berechtigung zur Verwaltung von Images, Volumes, Netzwerkschnittstellen und Servern.
- Eine Linux- oder MacOS-Workstation mit einer Bash-kompatiblen Shell. Die Befehlsbeispiele in diesem Leitfaden sind für Linux und MacOS vorgesehen.
- Genehmigte Netzwerkpräfixe, Sicherheitskontrollen, Verwaltungszugriff und eine optionale öffentliche IP, die dem STACKIT-Design entspricht.
- Ein autorisiertes C8000V QCOW2-Image für die ausgewählte Cisco IOS XE-Version.
- Ein STACKIT-Maschinentyp, der die veröffentlichten CPU- und Speicheranforderungen des C8000V für die ausgewählte Version und den ausgewählten Funktionssatz erfüllt.
- Eine Boot-Volume-Größe und Firmware-Einstellung, die für das ausgewählte C8000V-Image geeignet sind.
- QEMU wurde installiert, einschließlich des Dienstprogramms `qemu-img`, das zum Überprüfen des Quell-Images verwendet wird.
- `stackit`, `jq` und `base64` sind in der Befehlsumgebung verfügbar.
- Ein Day-0-Bootstrap, der für den Autonomous-, Controller- oder SD-Routing-Modus vorbereitet wurde.

Der Bootstrap kann Kennwörter, Zertifikate oder Informationen zur Geräteidentität enthalten. Bewahren Sie es an einem geschützten Ort auf. Die Base64-Codierung verschlüsselt die Datei nicht.

Validierte Softwareversionen und VM-Typen

Der in diesem Dokument beschriebene Bereitstellungs-Workflow wurde mit den folgenden Softwareversionen und STACKIT VM-Versionen validiert. Diese getestete Baseline dokumentiert die Laborumgebung. Sie ersetzt nicht die aktuellen Versionshinweise, Plattformanforderungen oder funktionsspezifischen Dimensionierungsleitfäden für Cisco Catalyst 8000V.

Softwareversionen

- Cisco IOS XE-Version: 17.18.5
- Version der Cisco SD-WAN Controller: 20.18.5 für Controller-Modus (Cisco SD-WAN)

STACKIT VM-Produktvarianten

Die folgenden STACKIT VM-Varianten wurden in die Validierung einbezogen:

Instanzgröße	vCPU	Speicher (GiB)
c3i4	4	8
c3i8	8	16
c3i.16	16	32

Wählen Sie eine Variante, die die CPU-, Arbeitsspeicher-, Durchsatz- und Funktionsanforderungen für die beabsichtigte C8000V-Bereitstellung erfüllt.

Vom Gast gemeldete Hypervisor details

Der validierte Gast des C8000V meldete die STACKIT-Virtualisierungsumgebung wie folgt:

```
<#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#

Konfigurieren

1. Installation der STACKIT-CLI

In dieser Anleitung wird Homebrew zur Installation der STACKIT-CLI verwendet. Weitere Paketmanager und Installationsmethoden sind verfügbar. Weitere Informationen finden Sie in der Installationsanleitung für die STACKIT-CLI.

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

2. Umgebung einstellen

Konfigurieren Sie die STACKIT-CLI-Umgebung, bevor Sie die Bereitstellungsbefehle ausführen.

2.1 Anmeldung und Authentifizierung

Melden Sie sich bei Ihrem STACKIT-Konto an.

```
stackit auth login
```

Schließen Sie die Authentifizierung in Ihrem Browser ab. Nach erfolgreicher Authentifizierung bestätigt die CLI, dass Sie angemeldet sind.

2.2 Festlegen des Projekts

Legen Sie das Standardprojekt für die verbleibenden Befehle fest.

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

Wenn Sie die Projekt-ID nicht kennen, listen Sie die verfügbaren Projekte auf.

```
stackit project list
```

3. Vorbereiten der Bereitstellungsdaten

Sammeln Sie die für die Bereitstellung erforderlichen Werte. Im Beispiel werden zwei Netzwerke und zwei Netzwerkkarten erstellt: die erste für die Verwaltung oder den Verkehr und die zweite für den Dienst- oder Datenverkehr.

```
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>"
```

Verwenden Sie nicht überlappende Präfixe aus dem genehmigten Netzwerkdesign. Geben Sie TRUSTED_SOURCE_CIDR in der CIDR-Notation einschließlich der Präfixlänge an. Verwenden Sie /32, um eine IPv4-Adresse zuzulassen. Die folgenden Beispiele für Verwaltungsregeln erlauben SSH nur von TRUSTED_SOURCE_CIDR.

4. Laden Sie das C8000V-Image hoch.

4.1 Überprüfen des Quellbilds

Überprüfen Sie, ob es sich bei der Datei um das autorisierte Image für die beabsichtigte Version handelt. Notieren Sie den Dateinamen und die Prüfsumme im Bereitstellungsdatensatz.

```
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 Erstellen des benutzerdefinierten STACKIT-Images

Legen Sie IMAGE_UEFI für ein EFI-Image auf true oder false für ein BIOS-Image fest. Wählen Sie die für das heruntergeladene C8000V-Image angegebene Einstellung aus. Legen Sie die minimale Festplattengröße entsprechend dem ausgewählten Image und der aktuellen Produktdokumentation fest.

```
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 Umfrage durchführen, bis das Bild verfügbar ist

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

Fahren Sie fort, wenn der Bildstatus VERFÜGBAR ist.

5. Boot-Volume erstellen

Erstellen Sie ein neues Boot-Volume aus dem benutzerdefinierten Image. Die Volume-Größe muss die Festplattenanforderungen des ausgewählten Images erfüllen.

```
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"
```

Fragen Sie das Volume ab, bis sein Status VERFÜGBAR ist.

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

6. Erstellen Sie Netzwerke und Sicherheit

Verwenden Sie vorhandene Netzwerke und Sicherheitsgruppen, wenn diese das genehmigte Design bereits erfüllen. Erstellen Sie andernfalls die Ressourcen wie unten gezeigt.

6.1 Erstellen oder Identifizieren von Netzwerken

Listen Sie die vorhandenen Netzwerke auf, bevor Sie neue erstellen.

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

Wenn keine geeigneten Netzwerke vorhanden sind, erstellen Sie das Management- oder Transportnetzwerk und das Dienst- oder Datennetzwerk.

```
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"
```

Wenn bereits genehmigte Netzwerke vorhanden sind, legen Sie stattdessen MGMT_NETWORK_ID und DATA_NETWORK_ID auf ihre IDs fest.

6.2 Erstellen der Management-Sicherheitsgruppe

Erstellen Sie eine Stateful-Security-Gruppe für die Management-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 von einem genehmigten Quell-CIDR zulassen.

```
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"
```

Fügen Sie HTTPS, NETCONF, ICMP oder Cisco SD-WAN-Steuerungs- und Datenebenenregeln nur dann hinzu, wenn dies für die Bereitstellung erforderlich ist. Begrenzen Sie jede Eingangsregel auf die engste praktische Quelle.

7. Erstellen Sie die bestellten NICs.

Erstellen Sie die Netzwerkkarten, bevor Sie den Server erstellen. Verknüpfen Sie die Management-Sicherheitsgruppe mit NIC 1, und übermitteln Sie die resultierenden NIC-IDs in der beabsichtigten Schnittstellenreihenfolge vom Typ C8000V.

7.1 Erstellen von NIC 1 und 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"

```

Im Beispiel zum Erstellen eines Servers wird MGMT_NIC_ID zuerst für GigabitEthernet1 und DATA_NIC_ID anschließend für GigabitEthernet2 gesendet. Zusätzliche NIC-IDs werden den nachfolgenden IOS XE-Schnittstellen in der angegebenen Reihenfolge zugeordnet. Überprüfen Sie die resultierende Zuordnung von IOS XE nach dem Serverstart.

Notieren Sie sich die NIC-ID, die Netzwerk-ID, die MAC-Adresse, die zugewiesene IP-Adresse und die beabsichtigte IOS XE-Schnittstelle für jede NIC.

7.2 Zuordnen einer optionalen öffentlichen IP-Adresse

Eine öffentliche IP ist mit einer bestimmten Netzwerkkarte verknüpft. Wenn eine externe Verwaltung erforderlich ist, verknüpfen Sie diese mit NIC 1.

Lassen Sie diesen Schritt aus, wenn der Managementzugriff über einen privaten Pfad bereitgestellt wird.

```

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. Vorbereiten der Day-0-Konfiguration

8.1 Auswählen der Bootstrap-Datei

Für den Autonomous-Modus erstellen Sie `iosxe_config.txt` mit einem IOS XE-Konfigurationsbefehl pro Zeile.

Generieren Sie für den Controller-Modus einen Cloud-Init-Bootstrap im Cisco SD-WAN Manager im Controller-Modus, und behalten Sie den Dateinamen `ciscosdwan_cloud_init.cfg` bei.

Generieren Sie für den SD-Routing-Modus einen Cloud-Init-Bootstrap im SD-WAN Manager im SD-Routing-Modus, und behalten Sie den Dateinamen `ciscosdwan_cloud_init.cfg` bei.

Bearbeiten Sie einen Bootstrap, der vom Cisco SD-WAN Manager generiert wurde, nicht manuell. Eine generierte Datei enthält Informationen zur Geräteidentität und zum Onboarding und muss als Artefakt geschützt werden, das die Anmeldeinformationen trägt.

8.2 Beispiel für Autonomous-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
```

Legen Sie den Pfad zum ausgewählten Bootstrap fest, und beschränken Sie den Zugriff auf die Datei.

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

8.3 Codieren des Bootstrap

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

9. Erstellen Sie die C8000V-Instanz

9.1 Erstellen der Server-Erstellungsanforderung

Die C8000V-spezifischen Einstellungen sind configDrive: true, das Base64-codierte Bootstrap in userData, das Image-basierte Boot-Volume und die bestellten NIC-IDs.

```
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
```

Überprüfen Sie die Anfrage, ohne den Bootstrap anzuzeigen:

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

9.2 Erstellen des Servers

Verwenden Sie den aktuellen, für die Zielumgebung dokumentierten STACKIT IaaS v2-Endpoint. Ersetzen Sie ggf. die 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}'
```

Beispielantwort:

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

Abfragen, bis der Server den erwarteten Ausführungszustand erreicht.

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

10. Überprüfen der Bereitstellung

10.1 Überprüfen der STACKIT-Ressourcen

Bestätigen Sie Folgendes:

- der Server in der vorgesehenen Projekt-, Region- und Verfügbarkeitszone ausgeführt wird;
- das vorgesehene Bootvolume ist angeschlossen;
- die erwarteten NICs angeschlossen sind;
- Die optionale öffentliche IP ist mit der beabsichtigten Netzwerkkarte verknüpft. und
- Die angewendeten Zugriffskontrollen stimmen mit dem genehmigten Design überein.

10.2 Überprüfen von Cisco IOS XE

Stellen Sie eine Verbindung über den genehmigten Verwaltungspfad her, und führen Sie die Prüfungen für die ausgewählte Version und den ausgewählten Modus aus.

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

Vergewissern Sie sich, dass der beabsichtigte Modus aktiv ist, die Day-0-Konfiguration angewendet wurde und die IOS XE-Schnittstellen mit der übermittelten NIC-Bestellung übereinstimmen.

Zusätzliche Befehle für den Controller-Modus:

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

Zusätzliche Befehle für den SD-Routing-Modus:

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

Nach der Überprüfung können Sie die Anmeldeinformationen für die temporäre Bereitstellung rotieren und die lokale Anforderungsdatei und die codierte Variable entfernen.

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

11. Entfernen Sie die Bereitstellung

Mit den Aufräumbefehlen in diesem Abschnitt werden Ressourcen endgültig gelöscht. Bestätigen Sie, dass die Bereitstellung nicht mehr erforderlich ist und dass jede Ressourcen-ID nur zu dieser C8000V-Bereitstellung gehört. Löschen Sie keine vorhandenen oder freigegebenen Netzwerke, Sicherheitsgruppen, Images oder anderen Ressourcen.

In den Bereinigungsbeispielen werden die interaktiven Bestätigungsaufforderungen beibehalten. Überprüfen Sie vor der Genehmigung des Vorgangs die in jeder Bestätigungsaufforderung der STACKIT-CLI angezeigte Ressource.

Führen Sie im Controller- oder SD-Routing-Modus die erforderliche Außerbetriebnahme des Controllers durch, bevor Sie die STACKIT-Ressourcen löschen.

11.1 Bereiten Sie das Entfernen vor.

Überprüfen Sie die aufgezeichneten Ressourcen-IDs, bevor Sie fortfahren.

```
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"
```

Wenn die Bereitstellung über eine öffentliche IP verfügt, trennen Sie sie, bevor Sie den Server löschen.

```
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 Löschen der Bereitstellungsressourcen

Löschen Sie zuerst den Server. Fahren Sie fort, nachdem der Server nicht mehr in der Serverliste angezeigt wird.

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

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

Löschen Sie die optionale öffentliche IP-Adresse, die beiden geordneten NICs und das Boot-Volume.

```
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"
```

Wenn die Sicherheitsgruppe und die Netzwerke nur für diese Bereitstellung erstellt wurden, löschen Sie sie, nachdem die abhängigen Ressourcen verschwunden sind.

```
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"
```



```
--project-id "$PROJECT_ID" \  
--region "$REGION"
```

Behalten Sie das benutzerdefinierte Image bei, wenn es für eine andere C8000V-Bereitstellung verwendet wird. Andernfalls löschen Sie es, nachdem alle abhängigen Boot-Volumes entfernt wurden.

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

11.3 Überprüfen der Bereinigung

Listen Sie die verbleibenden Ressourcen auf und vergewissern Sie sich, dass die gelöschten IDs nicht mehr vorhanden sind. Freigegebene Ressourcen, die absichtlich beibehalten wurden, sollten weiterhin angezeigt werden.

```
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"
```

Bestätigen Sie, dass alle lokalen Bootstrap-Artefakte, die nach der Bereitstellung aufbewahrt wurden, gemäß der Richtlinie zur Aufbewahrung von Anmeldeinformationen behandelt wurden.

Anhang

Nützliche STACKIT-Befehle

```
# 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"
```

Nützliche C8000V-Befehle

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

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.