Installazione e configurazione di Cisco Catalyst 8000V su STACKIT Cloud

Sommario

Introduzione

Questo documento descrive i passaggi necessari per installare e configurare Cisco Catalyst 8000V sul cloud STACKIT appartenente a Schwarz Digits.

Introduzione

Cisco Catalyst 8000V è un router virtuale con Cisco IOS XE. Su STACKIT, l'immagine C8000V viene caricata come immagine personalizzata e utilizzata per creare il volume di avvio per il server

virtuale.

Durante la creazione del server, STACKIT posiziona i dati utente forniti su un'unità di configurazione. C800V legge il bootstrap durante l'avvio iniziale. Il contenuto del bootstrap dipende dalla modalità operativa desiderata:

- La modalità autonoma utilizza iosxe_config.txt.
- La modalità controller utilizza ciscosdwan_cloud_init.cfg generato da Cisco SD-WAN Manager.
- La modalità di routing SD utilizza ciscosdwan_cloud_init.cfg generato da Cisco SD-WAN Manager per il routing SD.

Per tutte e tre le modalità viene utilizzata la stessa immagine, lo stesso volume, la stessa interfaccia e la stessa sequenza di creazione server. Il contenuto del bootstrap e la verifica post-distribuzione variano in base alla modalità.

Prerequisiti

Requisiti

- Un progetto STACKIT esistente e un operatore autenticato con autorizzazione a gestire immagini, volumi, interfacce di rete e server.
- Workstation Linux o macOS con una shell compatibile Bash. Gli esempi di comandi riportati in questa guida sono destinati a Linux e macOS.
- Prefissi di rete approvati, controlli di sicurezza, accesso di gestione e un indirizzo IP pubblico opzionale conforme alla progettazione STACKIT.
- Immagine C8000V QCOW2 autorizzata per la versione Cisco IOS XE selezionata.
- Tipo di macchina STACKIT che soddisfa i requisiti di memoria e CPU C8000V pubblicati per la versione e il set di funzionalità selezionati.
- Dimensioni del volume di avvio e impostazione del firmware appropriate per l'immagine C8000V selezionata.
- QEMU installato, inclusa l'utilità qemu-img utilizzata per ispezionare l'immagine sorgente.
- stackit, jq e base64 disponibili nell'ambiente command.
- Bootstrap Day-0 preparato per la modalità autonoma, controller o routing SD.

Il bootstrap può contenere password, certificati o informazioni sull'identità del dispositivo. Archiviarlo in un percorso protetto. La codifica Base64 non crittografa il file.

Versioni software convalidate e tipi di VM

Il flusso di lavoro di distribuzione descritto in questo documento è stato convalidato con le seguenti versioni software e le versioni di STACKIT VM. La presente baseline testata documenta l'ambiente di emulazione; non sostituisce le attuali note di rilascio di Cisco Catalyst 8000V, i requisiti della piattaforma o le linee guida per il dimensionamento specifico delle funzionalità.

Versioni software

- Cisco IOS XE release: 17.18.5
- Cisco SD-WAN Controller release: 20.18.5, utilizzato per la modalità controller (Cisco SD-WAN)

Tipi di VM STACKIT

Le seguenti versioni di STACKIT VM sono state incluse nella convalida:

Dimensioni istanza	vCPU	Memoria (GiB)
c3i.4.	4	8
c3i.8.	8	16
c3i.16	16	32

Selezionare una versione che soddisfi i requisiti di CPU, memoria, throughput e funzionalità per la distribuzione C8000V prevista.

Dettagli dell'hypervisor segnalati dal guest

Il guest C8000V convalidato ha riportato l'ambiente di virtualizzazione STACKIT come segue:

<#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#

Configurazione

1. Installare STACKIT CLI

Questa guida utilizza Homebrew per installare STACKIT CLI. Sono disponibili altri gestori di pacchetti e metodi di installazione. Per ulteriori informazioni, vedere la guida all'installazione di STACKIT CLI.

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

2. Impostazione dell'ambiente

Configurare l'ambiente STACKIT CLI prima di eseguire i comandi di distribuzione.

2.1 Accesso e autenticazione

Accedere all'account STACKIT.

```
stackit auth login
```

Completare l'autenticazione nel browser. Una volta completata l'autenticazione, la CLI conferma che l'accesso è stato eseguito.

2.2 Impostazione del progetto

Impostare il progetto predefinito per i comandi rimanenti.

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

Se non si conosce l'ID del progetto, elencare i progetti disponibili.

```
stackit project list
```

3. Preparare gli input di distribuzione

Raccogliere i valori richiesti dalla distribuzione. Nell'esempio vengono create due reti e due NIC: la prima per la gestione o il trasporto e la seconda per il traffico di servizi o di dati.

```
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>"
```

Usare prefissi non sovrapposti dalla progettazione di rete approvata. Specificare

TRUSTED_SOURCE_CIDR nella notazione CIDR, inclusa la lunghezza del prefisso. Utilizzare /32 per autorizzare un indirizzo IPv4. Gli esempi di regole di gestione riportati di seguito consentono l'uso del protocollo SSH solo da TRUSTED_SOURCE_CIDR.

4. Caricare l'immagine C800V

4.1 Ispezionare l'immagine sorgente

Verificare che il file sia l'immagine autorizzata per la versione desiderata. Registrare il nome file e il checksum nel record di distribuzione.

```
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 Creazione dell'immagine personalizzata STACKIT

Impostare IMAGE_UEFI su true per un'immagine EFI o su false per un'immagine BIOS. Selezionare l'impostazione specificata per l'immagine C8000V scaricata. Impostare le dimensioni minime del disco in base all'immagine selezionata e alla documentazione del prodotto corrente.

```
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 Effettuare un sondaggio fino a quando l'immagine non è disponibile

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

Continuare quando lo stato dell'immagine è DISPONIBILE.

5. Creare il volume di avvio

Creare un nuovo volume di avvio dall'immagine personalizzata. Le dimensioni del volume devono soddisfare i requisiti del disco dell'immagine selezionata.

```
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"
```

Eseguire il polling del volume fino a quando il relativo stato non è DISPONIBILE.

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

6. Creare reti e sicurezza

Utilizzare le reti e i gruppi di sicurezza esistenti quando soddisfano già il progetto approvato. In caso contrario, creare le risorse come illustrato di seguito.

6.1 Creazione o identificazione delle reti

Elenca le reti esistenti prima di crearne di nuove.

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

Se non esistono reti adeguate, creare la rete di gestione o di trasporto e la rete di servizi o di dati.

```
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"
```

Se esistono già reti approvate, impostare MGMT_NETWORK_ID e DATA_NETWORK_ID sui relativi ID.

6.2 Creazione del gruppo di sicurezza di gestione

Creare un gruppo di sicurezza con informazioni sullo stato per la scheda di interfaccia di rete di gestione.

```
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"
```

Consenti SSH da un CIDR di origine approvato

```
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"
```

Aggiungere le regole di controllo e data-plane HTTPS, NETCONF, ICMP o Cisco SD-WAN solo quando richiesto dalla distribuzione. Limitare ogni regola di entrata alla sorgente pratica più stretta.

7. Creare NIC ordinate

Creare le schede NIC prima di creare il server. Collegare il gruppo di sicurezza di gestione alla scheda NIC 1 e inviare gli ID NIC risultanti nell'ordine dell'interfaccia C8000V previsto.

7.1 Creazione della NIC 1 e della 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"
```

Nell'esempio di creazione del server, MGMT_NIC_ID viene inviato prima per Gigabit Ethernet1 e DATA_NIC_ID al secondo per Gigabit Ethernet2. Gli ID NIC aggiuntivi vengono mappati alle interfacce IOS XE successive nell'ordine inviato. Verificare il mapping risultante da IOS XE dopo l'avvio del server.

Registrare l'ID NIC, l'ID di rete, l'indirizzo MAC, l'indirizzo IP assegnato e l'interfaccia IOS XE prevista per ciascuna NIC.

7.2 Associare una IP pubblica facoltativa

Un indirizzo IP pubblico è associato a una scheda NIC specifica. Se è richiesta la gestione esterna, associarla alla scheda NIC 1.

Omettere questo passaggio quando l'accesso alla gestione è fornito tramite un percorso privato.

```
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. Preparare la configurazione del giorno 0

8.1 Selezione del file bootstrap

Per la modalità autonoma, creare iosxe_config.txt con un comando di configurazione IOS XE per riga.

Per la modalità controller, generare un bootstrap Cloud-Init in modalità controller in Cisco SD-WAN Manager e conservare il nome file ciscosdwan_cloud_init.cfg.

Per la modalità di routing SD, generare un bootstrap Cloud-Init in modalità di routing SD in Cisco SD-WAN Manager e conservare il nome file ciscosdwan_cloud_init.cfg.

Non modificare manualmente un bootstrap generato da Cisco SD-WAN Manager. Un file generato contiene l'identità del dispositivo e le informazioni di caricamento e deve essere protetto come elemento portante di credenziali.

8.2 Esempio di modalità autonoma

```
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
```

Impostare il percorso del bootstrap selezionato e limitare l'accesso al file.

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

8.3 Codifica del bootstrap

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

9. Creare l'istanza di C8000V

9.1 Creazione della richiesta di creazione del server

Le impostazioni specifiche di C800V sono configDrive: true, il bootstrap con codifica base64 in userData, il volume di avvio basato su immagini e gli ID NIC ordinati.

```
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
```

Rivedere la richiesta senza visualizzare il bootstrap:

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

9.2 Creazione del server

Utilizzare l'endpoint STACKIT IaaS v2 documentato per l'ambiente di destinazione. Sostituire l'URL dell'API STACKIT in base alle esigenze.

```
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}'
```

Esempio di risposta:

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

Eseguire il polling finché il server non raggiunge lo stato di esecuzione previsto.

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

10. Verificare la distribuzione

10.1 Verifica delle risorse STACKIT

Confermare che:

- il server è in esecuzione nella zona di disponibilità, nella regione e nel progetto previsti;
- il volume di avvio previsto è collegato;
- le schede NIC previste siano collegate;
- l'indirizzo IP pubblico facoltativo è associato alla scheda NIC desiderata; e
- i controlli di accesso applicati corrispondono al progetto approvato.

10.2 Verifica di Cisco IOS XE

Connettersi tramite il percorso di gestione approvato ed eseguire i controlli appropriati per la versione e la modalità selezionate.

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

Verificare che la modalità desiderata sia attiva, che la configurazione per il giorno 0 sia stata applicata e che le interfacce IOS XE corrispondano all'ordine NIC inviato.

Comandi aggiuntivi per la modalità controller:

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

Comandi aggiuntivi per la modalità di routing SD:

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

Dopo la verifica, ruotare le credenziali di provisioning temporanee e rimuovere il file di richiesta locale e la variabile codificata.

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

11. Rimuovere la distribuzione

I comandi di pulizia in questa sezione eliminano definitivamente le risorse. Confermare che la distribuzione non è più necessaria e che ogni ID risorsa appartiene solo a questa distribuzione C8000V. Non eliminare reti esistenti o condivise, gruppi di sicurezza, immagini o altre risorse.

Negli esempi di pulizia vengono mantenute le richieste di conferma interattive. Esaminare la risorsa visualizzata in ogni richiesta di conferma di STACKIT CLI prima di approvare l'operazione.

Per la modalità Controller o Routing SD, completare l'eventuale decommissionamento del controller prima di eliminare le risorse STACKIT.

11.1 Prepararsi alla rimozione

Esaminare gli ID di risorsa registrati prima di continuare.

```
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"
```

Se la distribuzione ha un IP pubblico, scollegarla prima di eliminare il server.

```
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 Eliminare le risorse di distribuzione

Eliminare prima il server. Continuare quando il server non è più visualizzato nell'elenco dei server.

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

```
  --region "$REGION"

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

Eliminare l'IP pubblico opzionale, le due schede NIC ordinate e il volume di avvio.

```
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"
```

Se il gruppo di sicurezza di gestione e le reti sono stati creati solo per questa distribuzione, eliminarli dopo l'eliminazione delle risorse dipendenti.

```
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"
```

Mantenere l'immagine personalizzata quando verrà utilizzata per un'altra distribuzione C8000V. In caso contrario, eliminarla dopo la rimozione di tutti i volumi di avvio dipendenti.

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

```
  --region "$REGION"
```

11.3 Verifica della pulitura

Elencare le risorse rimanenti e confermare che gli ID eliminati non sono più presenti. Le risorse condivise mantenute intenzionalmente dovrebbero comunque essere visualizzate.

```
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"
```

Confermare che tutti gli elementi di bootstrap locali mantenuti dopo la distribuzione siano stati gestiti in base ai criteri di conservazione delle credenziali dell'organizzazione.

Appendice

Comandi STACKIT utili

```
# 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"
```

Comandi C8000V

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

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).