# Instalación y configuración de Cisco Catalyst 8000V en STACKIT Cloud

## Contenido



## Introducción

Este documento describe los pasos necesarios para instalar y activar Cisco Catalyst 8000V en la nube STACKIT perteneciente a Schwarz Digits.

## Background

Cisco Catalyst 8000V es un router virtual que ejecuta Cisco IOS XE. En STACKIT, la imagen de C8000V se carga como una imagen personalizada y se utiliza para crear el volumen de arranque

para el servidor virtual.

Durante la creación del servidor, STACKIT coloca los datos de usuario suministrados en una unidad de configuración. C8000V lee la secuencia de inicio durante el arranque inicial. El contenido de bootstrap depende del modo de funcionamiento esperado:

- El modo autónomo utiliza iosxe_config.txt.
- El modo de controlador utiliza ciscosdwan_cloud_init.cfg generado por Cisco SD-WAN Manager.
- El modo de enrutamiento SD utiliza ciscosdwan_cloud_init.cfg generado por Cisco SD-WAN Manager para el enrutamiento SD.

Se utiliza la misma imagen, volumen, interfaz y secuencia de creación de servidor para los tres modos. El contenido de bootstrap y la verificación posterior a la implementación difieren según el modo.

# Prerequisites

## Requirements

- Un proyecto STACKIT existente y un operador autenticado con permiso para administrar imágenes, volúmenes, interfaces de red y servidores.
- Una estación de trabajo Linux o macOS con un shell compatible con Bash. Los ejemplos de comandos de esta guía están pensados para Linux y macOS.
- Prefijos de red aprobados, controles de seguridad, acceso a la gestión y una IP pública opcional que siguen el diseño STACKIT.
- Una imagen de C8000V QCOW2 autorizada para la versión seleccionada de Cisco IOS XE.
- Tipo de máquina STACKIT que cumple los requisitos de memoria y CPU publicados de C8000V para la versión seleccionada y el conjunto de funciones.
- Un tamaño de volumen de inicio y una configuración de firmware adecuados para la imagen de C8000V seleccionada.
- QEMU instalado, incluida la utilidad qemu-img utilizada para inspeccionar la imagen de origen.
- stackit, jq y base64 están disponibles en el entorno de comandos.
- Bootstrap de día 0 preparado para el modo autónomo, controlador o routing SD.

El bootstrap puede contener contraseñas, certificados o información de identidad del dispositivo. Guárdelo en una ubicación protegida. La codificación Base64 no cifra el archivo.

# Versiones de software validadas y tipos de VM

El flujo de trabajo de implementación de este documento se validó con las siguientes versiones de software y tipos de VM STACKIT. Esta línea de base probada documenta el entorno de laboratorio; no sustituye a las notas de la versión de Cisco Catalyst 8000V, los requisitos de la plataforma ni la orientación sobre el tamaño específico de las funciones actuales.

## Versiones de Software

- Versión de Cisco IOS XE: 17.18.5
- Versión de Cisco SD-WAN Controllers: 20.18.5, se utiliza para el modo de controlador (Cisco SD-WAN)

## Sabores de STACKIT VM

En la validación se incluyeron los siguientes tipos de VM de STACKIT:

| Tamaño de instancia | vCPU | Memoria (GiB) |
|---|---|---|
| c3i.4 | 4 | 8 |
| c3i.8 | 8 | 16 |
| c3i.16 | 16 | 32 |

Seleccione un tipo que cumpla los requisitos de CPU, memoria, rendimiento y funciones para la implementación prevista de C8000V.

## Detalles del hipervisor notificados por el invitado

El invitado validado de C8000V informó del entorno de virtualización STACKIT de la siguiente manera:

<#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#

# Configurar

## 1. Instale la CLI STACKIT

Esta guía utiliza Homebrew para instalar la CLI STACKIT. Hay disponibles otros administradores de paquetes y métodos de instalación. Para obtener más información, consulte la guía de instalación de STACKIT CLI.

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

## 2. Establecer el entorno

Configure el entorno CLI STACKIT antes de ejecutar los comandos de implementación.

### 2.1 Inicio de sesión y autenticación

Inicie sesión en su cuenta STACKIT.

```
stackit auth login
```

Complete la autenticación en su navegador. Una vez que la autenticación se haya realizado correctamente, la CLI confirma que ha iniciado sesión.

### 2.2 Establecer el proyecto

Establezca el proyecto predeterminado para los comandos restantes.

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

Si no conoce el Id. del proyecto, enumere los proyectos disponibles.

```
stackit project list
```

## 3. Preparar los insumos para el despliegue

Recopile los valores requeridos por la implementación. El ejemplo crea dos redes y dos NIC: el primero para la gestión o el transporte y el segundo para el tráfico de servicios o datos.

```
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>"
```

Utilice prefijos no solapados del diseño de red aprobado. Especifique TRUSTED_SOURCE_CIDR

en notación CIDR, incluida la longitud del prefijo. Utilice /32 para permitir una dirección IPv4. Los siguientes ejemplos de reglas de administración permiten SSH sólo desde TRUSTED_SOURCE_CIDR..

## 4. Cargue la imagen de C8000V

### 4.1 Inspeccionar la imagen de origen

Verifique que el archivo sea la imagen autorizada para la versión deseada. Registre el nombre de archivo y la suma de comprobación en el registro de implementación.

```
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 Crear la imagen personalizada STACKIT

Establezca IMAGE_UEFI en true para una imagen EFI o en false para una imagen BIOS. Seleccione la configuración especificada para la imagen C8000V descargada. Establezca el tamaño mínimo de disco según la imagen seleccionada y la documentación del producto actual.

```
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 Encuesta hasta que la imagen esté disponible

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

Continúe cuando el estado de la imagen sea DISPONIBLE.

## 5. Cree el volumen de arranque

Cree un nuevo volumen de arranque a partir de la imagen personalizada. El tamaño del volumen debe cumplir los requisitos de disco de la imagen seleccionada.

```
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"
```

Sondee el volumen hasta que su estado sea DISPONIBLE.

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

## 6. Crear redes y seguridad

Utilice las redes y los grupos de seguridad existentes cuando ya cumplan el diseño aprobado. De lo contrario, cree los recursos como se muestra a continuación.

6.1 Crear o identificar las redes

Enumera las redes existentes antes de crear otras nuevas.

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

Si no existen redes adecuadas, cree la red de gestión o transporte y la red de servicio o de datos.

```
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 ya existen redes aprobadas, establezca MGMT_NETWORK_ID y DATA_NETWORK_ID en sus ID.

6.2 Crear el grupo de seguridad de administración

Cree un grupo de seguridad con estado para la NIC de administración.

```
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"
```

Permitir SSH desde un origen aprobado 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"
```

Agregue reglas de plano de datos y control HTTPS, NETCONF, ICMP o Cisco SD-WAN solo cuando la implementación lo requiera. Limite cada regla de ingreso a la fuente práctica más estrecha.

## 7. Crear NIC ordenadas

Cree las NIC antes de crear el servidor. Adjunte el grupo de seguridad de administración a NIC 1 y envíe los ID de NIC resultantes en el orden de interfaz C8000V previsto.

### 7.1 Creación de NIC 1 y 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"
```

El ejemplo de creación de servidor envía MGMT_NIC_ID primero para GigabitEthernet1 y DATA_NIC_ID segundo para GigabitEthernet2. Los ID de NIC adicionales se asignan a las interfaces IOS XE subsiguientes en el orden enviado. Verifique la asignación resultante de IOS XE después de que se inicie el servidor.

Registre el ID de NIC, el ID de red, la dirección MAC, la dirección IP asignada y la interfaz IOS XE prevista para cada NIC.

### 7.2 Asociar una IP pública opcional

Una IP pública está asociada a una NIC específica. Cuando se requiera gestión externa, asóciela a NIC 1.

Omita este paso cuando el acceso a la gestión se proporcione a través de una ruta privada.

```
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. Prepare la configuración de día 0

### 8.1 Seleccione el archivo de bootstrap

Para el modo autónomo, cree iosxe_config.txt con un comando de configuración IOS XE por

línea.

Para el modo de controlador, genere un bootstrap de Cloud-Init en modo de controlador en Cisco SD-WAN Manager y conserve el nombre de archivo ciscosdwan_cloud_init.cfg.

Para el modo de ruteo SD, genere un bootstrap de Cloud-Init en modo de ruteo SD en Cisco SD-WAN Manager y conserve el nombre de archivo ciscosdwan_cloud_init.cfg.

No edite manualmente un bootstrap generado por Cisco SD-WAN Manager. Un archivo generado contiene la identidad del dispositivo y la información de incorporación, y debe protegerse como un artefacto con credenciales.

8.2 Ejemplo de modo autónomo

```
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
```

Establezca la ruta de acceso al programa previo seleccionado y restrinja el acceso al archivo.

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

### 8.3 Codificación del bootstrap

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

## 9. Crear la instancia de C8000V

### 9.1 Crear la solicitud de creación de servidor

Los parámetros específicos de C8000V son configDrive: true, la secuencia de inicio codificada en base64 en userData, el volumen de inicio basado en imagen y los ID de NIC ordenados.

```
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
```

Revise la solicitud sin mostrar el bootstrap:

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

### 9.2 Crear el servidor

Utilice el terminal STACKIT IaaS v2 actual documentado para el entorno de destino. Sustituya la URL de la API STACKIT según sea necesario.

```
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}'
```

Respuesta de ejemplo:

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

Sondear hasta que el servidor alcance el estado de ejecución esperado.

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

## 10. Compruebe la implementación

### 10.1 Verificación de los recursos STACKIT

Confirme que:

- el servidor se está ejecutando en el proyecto, la región y la zona de disponibilidad previstos;
- el volumen de arranque previsto está conectado;
- se adjuntan las NIC esperadas;
- la dirección IP pública opcional está asociada a la NIC deseada; y
- los controles de acceso aplicados coinciden con el diseño aprobado.

## 10.2 Verificación de Cisco IOS XE

Conéctese a través de la ruta de gestión aprobada y ejecute las comprobaciones correspondientes a la versión y el modo seleccionados.

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

Confirme que el modo deseado esté activo, que se haya aplicado la configuración de día 0 y que las interfaces IOS XE coincidan con el pedido de NIC enviado.

Comandos adicionales para el modo de controlador:

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

Comandos adicionales para el modo de ruteo SD:

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

Después de la verificación, gire las credenciales de aprovisionamiento temporales y quite el archivo de solicitud local y la variable codificada.

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

## 11. Elimine la implementación

Los comandos de limpieza de esta sección eliminan recursos de forma permanente. Confirme que la implementación ya no es necesaria y que cada ID de recurso pertenece únicamente a esta implementación de C8000V. No elimine redes, grupos de seguridad, imágenes u otros recursos existentes o compartidos.

Los ejemplos de limpieza conservan los mensajes de confirmación interactivos. Revise el recurso mostrado en cada mensaje de confirmación de STACKIT CLI antes de aprobar la operación.

Para el modo de controlador o routing SD, complete cualquier retirada del controlador antes de eliminar los recursos STACKIT.

### 11.1 Preparación para la retirada

Revise los ID de recursos registrados antes de continuar.

```
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 la implementación tiene una dirección IP pública, sepárela antes de eliminar el servidor.

```
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 Eliminación de los recursos de implementación

Elimine el servidor primero. Continuar cuando el servidor ya no aparezca en la lista de servidores.

```
stackit server delete "$SERVER_ID" \
```

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

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

Elimine la IP pública opcional, las dos NIC ordenadas y el volumen de arranque.

```
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 el grupo de seguridad de administración y las redes se crearon únicamente para esta implementación, elimínelos una vez que hayan desaparecido sus recursos dependientes.

```
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"
```

Conserve la imagen personalizada cuando se vaya a utilizar para otra implementación de C8000V. De lo contrario, elimínelo después de quitar todos los volúmenes de arranque dependientes.

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

### 11.3 Verificar la limpieza

Enumere los recursos restantes y confirme que los identificadores eliminados ya no están presentes. Los recursos compartidos que se retuvieron intencionadamente deben seguir apareciendo.

```
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"
```

Confirme que todos los artefactos de bootstrap locales retenidos después de la implementación se hayan gestionado de acuerdo con la política de retención de credenciales de la organización.

# Appendix

## Comandos STACKIT útiles

```
# 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"
```

## Comandos útiles de C8000V

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

```
show ip route
show logging
```

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).