Instalar e configurar o Cisco Catalyst 8000V no STACKIT Cloud

Contents

Introdução

Este documento descreve as etapas necessárias para instalar e ativar o Cisco Catalyst 8000V na nuvem STACKIT pertencente aos Schwarz Digits.

Background

O Cisco Catalyst 8000V é um roteador virtual que executa o Cisco IOS XE. No STACKIT, a imagem do C8000V é carregada como uma imagem personalizada e usada para criar o volume

de inicialização do servidor virtual.

Durante a criação do servidor, o STACKIT coloca os dados de usuário fornecidos em uma unidade de configuração. O C8000V lê o bootstrap durante a inicialização inicial. O conteúdo do bootstrap depende do modo de operação pretendido:

- O modo autônomo usa iosxe_config.txt.
- O modo do controlador usa ciscosdwan_cloud_init.cfg gerado pelo Cisco SD-WAN Manager.
- O modo de Roteamento SD usa ciscosdwan_cloud_init.cfg gerado pelo Cisco SD-WAN Manager para Roteamento SD.

A mesma imagem, volume, interface e sequência de criação de servidor é usada para todos os três modos. O conteúdo de bootstrap e a verificação pós-implantação diferem por modo.

Pré-requisitos

Requisitos

- Projeto STACKIT existente e operador autenticado com permissão para gerenciar imagens, volumes, interfaces de rede e servidores.
- Uma estação de trabalho Linux ou macOS com um shell compatível com Bash. Os exemplos de comandos neste guia destinam-se a Linux e macOS.
- Prefixos de rede aprovados, controles de segurança, acesso de gerenciamento e um IP público opcional que seguem o projeto STACKIT.
- Uma imagem autorizada C8000V QCOW2 para a versão XE do Cisco IOS selecionada.
- Um tipo de máquina STACKIT que atende aos requisitos de CPU e memória do C8000V publicados para a versão selecionada e o conjunto de recursos.
- Um tamanho do volume de inicialização e configuração de firmware apropriados para a imagem do C8000V selecionada.
- QEMU instalado, incluindo o utilitário qemu-img usado para inspecionar a imagem de origem.
- stackit, jq e base64 disponíveis no ambiente de comando.
- Um bootstrap de dia 0 preparado para o modo Autônomo, Controlador ou Roteamento SD.

O bootstrap pode conter senhas, certificados ou informações de identidade do dispositivo. Armazene-o em um local protegido. A codificação Base64 não criptografa o arquivo.

Versões de software validadas e tipos de VM

O fluxo de trabalho de implantação neste documento foi validado com as seguintes versões de software e tipos de VM STACKIT. Esta linha de base testada documenta o ambiente de laboratório; ele não substitui as notas de versão atuais do Cisco Catalyst 8000V, requisitos de plataforma ou orientação de dimensionamento específica de recursos.

Necessárias

- Versão do Cisco IOS XE: 17.18.5
- Versão dos controladores SD-WAN da Cisco: 20.18.5, usado para o modo de controlador (Cisco SD-WAN)

Tipos de VM STACKIT

Os seguintes tipos de VM STACKIT foram incluídos na validação:

Tamanho da instância	vCPUS	Memória (GiB)
c3i.4	4	8
c3i.8	8	16
c3i.16	16	32

Selecione um tipo que atenda aos requisitos de CPU, memória, throughput e recursos para a implantação pretendida do C8000V.

Detalhes do hipervisor relatados pelo convidado

O convidado C8000V validado relatou o ambiente de virtualização STACKIT da seguinte maneira:

<#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. Instalar o STACKIT CLI

Este guia usa o Homebrew para instalar o STACKIT CLI. Outros gerenciadores de pacotes e métodos de instalação estão disponíveis. Para obter mais informações, consulte o guia de instalação do STACKIT CLI.

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

2. Definir o ambiente

Configure o ambiente STACKIT CLI antes de executar os comandos de implantação.

2.1 Fazer login e autenticar

Faça login na sua conta STACKIT.

```
stackit auth login
```

Conclua a autenticação em seu navegador. Após a autenticação ser bem-sucedida, a CLI confirma que você está conectado.

2.2 Definir o projeto

Defina o projeto padrão para os comandos restantes.

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

Se você não souber a ID do projeto, liste os projetos disponíveis.

```
stackit project list
```

3. Preparar os dados de implantação

Colete os valores exigidos pela implantação. O exemplo cria duas redes e duas placas de rede: o primeiro para gerenciamento ou transporte e o segundo para serviço ou tráfego de dados.

```
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>"
```

Use prefixos não sobrepostos do projeto de rede aprovado. Especifique

TRUSTED_SOURCE_CIDR na notação CIDR, incluindo o comprimento do prefixo. Use /32 para permitir um endereço IPv4. Os exemplos de regra de gerenciamento abaixo permitem SSH apenas de TRUSTED_SOURCE_CIDR.

4. Carregue a imagem do C8000V

4.1 Inspecione a imagem de origem

Verifique se o arquivo é a imagem autorizada para a versão desejada. Registre o nome do arquivo e a soma de verificação no registro de implantação.

```
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 Criar a imagem personalizada do STACKIT

Defina IMAGE_UEFI como true para uma imagem EFI ou false para uma imagem BIOS. Selecione a configuração especificada para a imagem C8000V baixada. Defina o tamanho mínimo do disco de acordo com a imagem selecionada e a documentação atual do produto.

```
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 Pesquisa até que a imagem esteja disponível

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

Continue quando o status da imagem for AVAILABLE.

5. Criar o volume de inicialização

Crie um novo volume de inicialização a partir da imagem personalizada. O tamanho do volume deve atender aos requisitos de disco da imagem selecionada.

```
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"
```

Faça polling no volume até que seu status seja AVAILABLE.

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

6. Criar redes e segurança

Use as redes e os grupos de segurança existentes quando eles já atenderem ao projeto aprovado. Caso contrário, crie os recursos conforme mostrado abaixo.

6.1 Criar ou identificar as redes

Liste as redes existentes antes de criar novas.

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

Se não existirem redes adequadas, crie a rede de gerenciamento ou transporte e a rede de serviços ou dados.

```
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 já existirem redes aprovadas, defina MGMT_NETWORK_ID e DATA_NETWORK_ID como suas IDs.

6.2 Criar o grupo de segurança de gerenciamento

Crie um grupo de segurança stateful para a NIC de gerenciamento.

```
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 de um CIDR de origem aprovado

```
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"
```

Adicione o controle HTTPS, NETCONF, ICMP ou Cisco SD-WAN e as regras de plano de dados somente quando exigido pela implantação. Limite cada regra de ingresso à fonte prática mais estreita.

7. Criar NICs ordenadas

Crie as NICs antes de criar o servidor. Anexe o grupo de segurança de gerenciamento à NIC 1 e envie as IDs de NIC resultantes na ordem de interface pretendida do C8000V.

7.1 Criar NIC 1 e 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"
```

O exemplo de criação de servidor envia MGMT_NIC_ID primeiro para GigabitEthernet1 e DATA_NIC_ID segundo para GigabitEthernet2. As IDs de NIC adicionais mapeiam para interfaces IOS XE subsequentes na ordem enviada. Verifique o mapeamento resultante do IOS XE após a inicialização do servidor.

Registre a ID da NIC, a ID da rede, o endereço MAC, o endereço IP atribuído e a interface IOS XE pretendida para cada NIC.

7.2 Associar um IP público opcional

Um IP público é associado a uma placa de rede específica. Quando o gerenciamento externo for necessário, associe-o à placa de rede 1.

Omita esta etapa quando o acesso de gerenciamento for fornecido através de um caminho privado.

```
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. Preparar a configuração de Dia 0

8.1 Selecione o arquivo de bootstrap

Para o modo Autônomo, crie iosxe_config.txt com um comando de configuração IOS XE por linha.

Para o modo de controlador, gere um bootstrap Cloud-Init no modo de controlador no Cisco SD-WAN Manager e mantenha o nome de arquivo ciscosdwan_cloud_init.cfg.

Para o modo de roteamento SD, gere um bootstrap de Cloud-Init no modo de roteamento SD no Cisco SD-WAN Manager e mantenha o nome de arquivo ciscosdwan_cloud_init.cfg.

Não edite manualmente um bootstrap gerado pelo Cisco SD-WAN Manager. Um arquivo gerado contém informações de identidade e integração do dispositivo e deve ser protegido como um artefato portador de credenciais.

8.2 Exemplo de modo autônomo

```
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
```

Definir o caminho para o bootstrap selecionado e restringir o acesso ao arquivo.

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

8.3 Codificação do bootstrap

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

9. Crie a instância do C8000V

9.1 Criar a solicitação de criação de servidor

As configurações específicas do C8000V são configDrive: true, o bootstrap codificado na base64 em userData, o volume de inicialização baseado em imagem e as IDs de NIC solicitadas.

```
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 a solicitação sem exibir o bootstrap:

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

9.2 Criar o servidor

Use o endpoint STACKIT IaaS v2 atual documentado para o ambiente de destino. Substitua a URL da API STACKIT conforme necessário.

```
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}'
```

Exemplo de resposta:

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

Faça uma chamada seletiva até que o servidor atinja o estado de execução esperado.

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

10. Verificar a implantação

10.1 Verificar os recursos do STACKIT

Confirme se:

- o servidor está sendo executado no projeto pretendido, região e zona de disponibilidade;
- está ligado o volume de arranque previsto;
- as placas de rede esperadas estão conectadas;
- o IP público opcional está associado à placa de rede pretendida; e
- os controles de acesso aplicados correspondem ao projeto aprovado.

10.2 Verificar o Cisco IOS XE

Conecte-se por meio do caminho de gerenciamento aprovado e execute as verificações apropriadas para a versão e o modo selecionados.

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

Confirme se o modo pretendido está ativo, se a configuração de Dia 0 foi aplicada e se as interfaces do IOS XE correspondem ao pedido de placa de rede enviado.

Comandos adicionais para o modo de controlador:

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

Comandos adicionais para o modo de roteamento SD:

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

Após a verificação, gire as credenciais de provisionamento temporárias e remova o arquivo de solicitação local e a variável codificada.

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

11. Remova a implantação

Os comandos de limpeza nesta seção excluem recursos permanentemente. Confirme se a implantação não é mais necessária e se cada ID de recurso pertence somente a esta implantação do C8000V. Não exclua redes, grupos de segurança, imagens ou outros recursos existentes ou compartilhados.

Os exemplos de limpeza retêm prompts de confirmação interativos. Revise o recurso mostrado em cada prompt de confirmação do STACKIT CLI antes de aprovar a operação.

Para o modo de controlador ou roteamento SD, conclua qualquer descomissionamento do lado do controlador necessário antes de excluir os recursos STACKIT.

11.1 Preparar para remoção

Revise as IDs de recurso registradas 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"
```

Se a implantação tiver um IP público, desanexe-o antes de excluir o 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 Excluir os recursos de implantação

Exclua o servidor primeiro. Continue depois que o servidor não aparecer mais na lista de servidores.

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

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

Exclua o IP público opcional, as duas placas de rede ordenadas e o volume de inicialização.

```
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 o grupo de segurança de gerenciamento e as redes tiverem sido criados somente para esta implantação, exclua-os depois que os recursos dependentes forem removidos.

```
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"
```

Mantenha a imagem personalizada quando ela for usada para outra implantação do C8000V. Caso contrário, exclua-o depois que todos os volumes de inicialização dependentes forem removidos.

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

11.3 Verificar a limpeza

Liste os recursos restantes e confirme se as IDs excluídas não estão mais presentes. Os recursos compartilhados que foram intencionalmente retidos ainda devem aparecer.

```
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 se todos os artefatos de bootstrap locais mantidos após a implantação foram tratados de acordo com a política de retenção de credenciais da organização.

Appendix

Comandos STACKIT úteis

```
# 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 C8000V úteis

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

```
show logging
```

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.