

在STACKIT雲上安裝並配置Cisco Catalyst 8000V

目錄

[簡介](#)

[背景](#)

[必要條件](#)

[需求](#)

[經驗證的軟體版本和VM型別](#)

[軟體版本](#)

[STACKIT VM風格](#)

[訪客報告的虛擬機器監控程式詳細資訊](#)

[設定](#)

[1. 安裝STACKIT CLI](#)

[2. 設定環境](#)

[3. 準備部署投入](#)

[4. 上傳C8000V映像](#)

[5. 建立引導卷](#)

[6. 建立網路和安全性](#)

[7. 建立有序的NIC](#)

[8. 準備第0天配置](#)

[9. 建立C8000V例項](#)

[10. 驗證部署](#)

[11. 刪除部署](#)

[附錄](#)

[有用的STACK命令](#)

[有用的C8000V命令](#)

簡介

本文檔介紹在屬於Schwarz Digits的[STACKIT雲上安裝](#)和啟用Cisco [Catalyst](#) 8000V所需的步驟。

背景

Cisco Catalyst 8000V是執行Cisco IOS XE的虛擬路由器。在STACKIT上，C8000V映像作為自定義映像上傳，並用於為虛擬伺服器建立引導卷。

在建立伺服器期間，STACKIT會將提供的使用者資料放在配置驅動器上。C8000V在初始引導期間讀取載入程式。載入程式內容取決於預期操作模式：

- 自主模式使用iosxe_config.txt。
- 控制器模式使用由Cisco SD-WAN Manager生成的ciscosdwan_cloud_init.cfg。
- SD-Routing模式使用由Cisco SD-WAN Manager為SD-Routing生成的ciscosdwan_cloud_init.cfg。

所有三種模式使用相同的映像、卷、介面和伺服器建立順序。載入程式內容和部署後驗證因模式而異。

必要條件

需求

- 現有STACKIT專案和具有管理映像、卷、網路介面和伺服器的許可權的已驗證操作員。
- 具有Bash相容外殼的Linux或macOS工作站。本指南中的命令示例適用於Linux和macOS。
- 遵循STACKIT設計的經批准的網路字首、安全控制、管理訪問和可選公用IP。
- 所選思科IOS XE版本的授權C8000V QCOW2映像。
- 滿足所選版本和功能集已發佈的C8000V CPU和記憶體要求的STACKIT電腦型別。
- 適用於所選C8000V映像的啟動卷大小和韌體設定。
- 已安裝QEMU，包括用於檢查源映像的qemu-img實用程式。
- stackit、jq和base64在命令環境中可用。
- 為自治、控制器或SD-Routing模式準備的第0天載入程式。

引導可以包含密碼、證書或裝置身份資訊。將其儲存在受保護位置。Base64編碼不加密檔案。

經驗證的軟體版本和VM型別

本文檔中的部署工作流程已使用以下軟體版本和STACKIT VM風格進行驗證。該測試基線記錄實驗室環境；它不會取代當前的Cisco Catalyst 8000V版本說明、平台要求或特定功能的規模確定指南。

軟體版本

- Cisco IOS XE版本：17.18.5
- Cisco SD-WAN控制器版本：20.18.5，用於控制器(Cisco SD-WAN)模式

STACKIT VM風格

驗證中包括以下STACKIT VM風格：

例項大小	vCPU	記憶體(GiB)
c3i.4	4	8
c3i.8	8	16
c3i.16	16	32

選擇滿足目標C8000V部署的CPU、記憶體、吞吐量 and 功能要求的風格。

訪客報告的虛擬機器監控程式詳細資訊

經過驗證的C8000V訪客報告了STACKIT虛擬化環境，如下所示：

```
<#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#
```

設定

1. 安裝STACKIT CLI

本指南使用Homebrew安裝STACKIT CLI。還提供其他軟體包管理器和安裝方法。有關詳細資訊，請參閱《STACKIT CLI安裝指南》。

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

2. 設定環境

在運行部署命令之前配置STACKIT CLI環境。

2.1 登入和身份驗證

登入到您的STACKIT帳戶。

```
stackit auth login
```

在瀏覽器中完成身份驗證。身份驗證成功後，CLI會確認您已登入。

2.2 設定專案

為其餘命令設定預設專案。

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

如果您不知道專案ID，請列出可用的專案。

```
stackit project list
```

3.準備部署投入

收集部署所需的值。此示例建立兩個網路和兩個NIC:第一個用於管理或傳輸，第二個用於服務或資料流量。

```
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>"
```

使用已批准網路設計中的非重疊字首。在CIDR表示法中指定TRUSTED_SOURCE_CIDR，包括字首長度。使用/32允許一個IPv4地址。以下管理規則示例僅允許來自TRUSTED_SOURCE_CIDR的SSH。

4.上傳C8000V映像

4.1檢查源映像

驗證該檔案是否為目標版本的授權映像。在部署記錄中記錄檔名和校驗和。

```
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 建立STACKIT自定義映像

將EFI映像的IMAGE_UEFI設定為true，或將BIOS映像的false。選擇為下載的C8000V映像指定的設定。根據選定的映像和當前產品文檔設定最小磁碟大小。

```
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 輪詢，直到映像可用

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

映像狀態為AVAILABLE時繼續。

5. 建立引導卷

從自定義映像建立新的引導卷。卷大小必須符合所選映像的磁碟要求。

```
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"
```

輪詢卷直到其狀態為AVAILABLE。

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

6.建立網路和安全性

當現有網路和安全組已滿足批准的設計時，使用它們。否則，請建立如下所示的資源。

6.1建立或確定網路

請在建立新網路之前列出現有網路。

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

如果不存在合適的網路，請建立管理或傳輸網路以及服務或資料網路。

```
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"

```

如果批准的網路已存在，請將MGMT_NETWORK_ID和DATA_NETWORK_ID設定為它們的ID。

6.2建立管理安全組

為管理NIC建立狀態安全組。

```

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"

```

允許來自已批准的源CIDR的SSH。

```

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"

```

僅當部署需要時，才新增HTTPS、NETCONF、ICMP或Cisco SD-WAN控制和資料平面規則。將每個輸入規則限制在最窄的實際源上。

7.建立有序的NIC

建立伺服器之前建立NIC。將管理安全組連線到NIC 1，並按照預期的C8000V介面順序提交生成的NIC ID。

7.1建立NIC 1和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"
```

server-create示例首先為GigabitEthernet1提交MGMT_NIC_ID，其次為GigabitEthernet2提交DATA_NIC_ID。其他NIC ID按提交順序對映到後續IOS XE介面。在伺服器啟動後驗證IOS XE生成的對映。

記錄每個NIC的NIC ID、網路ID、MAC地址、分配的IP地址和預定的IOS XE介面。

7.2關聯可選的公共IP

公共IP與特定NIC關聯。如果需要外部管理，請將其與NIC 1關聯。

在通過私有路徑提供管理訪問時忽略此步驟。

```
PUBLIC_IP_RESPONSE="$(  
  stackit public-ip create \  
    --project-id "$PROJECT_ID" \  
    --region "$REGION" \  
    --output-format json  
)"
```

```

        --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.準備第0天配置

8.1選擇引導檔案

對於自主模式，使用每行一個IOS XE配置命令建立iosxe_config.txt。

對於控制器模式，在Cisco SD-WAN Manager中生成控制器模式Cloud-Init bootstrap並保留檔名ciscosdwan_cloud_init.cfg。

對於SD-Routing模式，在Cisco SD-WAN Manager中生成SD-Routing-mode Cloud-Init bootstrap並保留檔名ciscosdwan_cloud_init.cfg。

請勿手動編輯Cisco SD-WAN Manager生成的載入程式。生成的檔案包含裝置標識和登入資訊，必須作為帶有憑據的對象進行保護。

8.2自主模式示例

```

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
```

設定所選引導的路徑並限制對檔案的訪問。

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

8.3對載入程式進行編碼

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

9.建立C8000V例項

9.1生成伺服器建立請求

C8000V特定的設定是configDrive:true、userData中採用base64編碼的bootstrap、基於映像的啟動卷以及有序的NIC ID。

```
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

```

在不顯示引導的情況下檢視請求：

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

9.2 建立伺服器

使用針對目標環境記錄的當前STACKIT IaaS v2終結點。根據需要替換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}'

```

響應示例：

```

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

```

輪詢直到伺服器達到預期的運行狀態。

```

stackit server describe "$SERVER_ID" \
    --project-id "$PROJECT_ID" \
    --region "$REGION" \
    --output-format json |

```

```
jq '{id, name, status, powerStatus}'
```

10. 驗證部署

10.1 檢驗STACKIT資源

確認：

- 伺服器正在目標專案、區域和可用區域中運行；
- 已連線預期的啟動卷；
- 已連線預期的NIC；
- 可選的公共IP與預定的NIC關聯；和
- 應用的訪問控制與批准的設計相匹配。

10.2 檢驗Cisco IOS XE

通過批准的管理路徑進行連線，並運行與所選版本和模式相應的檢查。

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

確認目標模式處於活動狀態，已應用第0天配置，並且IOS XE介面與提交的NIC順序匹配。

控制器模式的其他命令：

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

SD-Routing模式的其他命令：

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

```
show sd-routing system status
```

驗證後，輪換臨時設定憑據並刪除本地請求檔案和編碼變數。

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

11. 刪除部署

本節中的清理命令會永久刪除資源。確認不再需要部署，並且每個資源ID僅屬於此C8000V部署。請勿刪除現有或共用網路、安全組、映像或其他資源。

清除示例會保留互動式確認提示。在批准操作之前，檢視每個STACKIT CLI確認提示中顯示的資源。

對於控制器或SD-Routing模式，請在刪除STACKIT資源之前完成所需的控制器端停用。

11.1 準備搬遷

在繼續操作之前，請檢視記錄的資源ID。

```
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"
```

如果部署具有公共IP，請在刪除伺服器之前將其分離。

```
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 刪除部署資源

首先刪除伺服器。在伺服器清單中不再出現伺服器後繼續操作。

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

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

刪除可選的公共IP、兩個有序的NIC和啟動卷。

```
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"
```

如果僅為此部署建立了管理安全組和網路，則在其從屬資源消失後將其刪除。

```
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"
```

將自定義映像用於其他C8000V部署時，請保留該映像。否則，請在刪除所有相關的啟動卷後將其刪除。

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

11.3 驗證清除

列出其餘資源並確認刪除的ID不再存在。故意保留的共用資源仍應顯示。

```
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"
```

確認已根據組織的憑據保留策略處理部署後保留的所有本地引導專案。

附錄

有用的STACK命令

```
# 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"
```

有用的C8000V命令

```
show version
```



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

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。