

UCS上的RHOSP部署指南，適用於斷開模式下移動工作負載

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[瞭解RHOSP](#)

[RHOSP架構](#)

[Undercloud](#)

[Undercloud元件](#)

[過雲](#)

[控制器](#)

[計算](#)

[儲存](#)

[本地儲存庫 \(REPO伺服器 \)](#)

[RHOSP網路](#)

[RHOSP物理連線](#)

[RHOSP邏輯連線](#)

[硬體引數調整](#)

[虛擬機器監控程式 — KVM安裝和網路建立](#)

[REPO和導向器VM建立](#)

[建立REPO和Director VM的前提條件](#)

[建立回購VM](#)

[Director VM建立](#)

[建立本地離線回購](#)

[RHOSP雲部署](#)

[必要條件](#)

[更新輸入檔案](#)

[雲下部署](#)

[超雲部署](#)

[訪問水平儀表板](#)

[RHOSP群集的運行狀況檢查](#)

[摘要](#)

簡介

本文檔介紹在C220 M6 UCS伺服器上部署RHOSP以支援Cisco VPC-DI的框架。

必要條件

需求

Cisco建議您瞭解Red Hat OpenStack Platform(RHOSP)，並熟悉Red Hat Enterprise Linux(RHEL)的強大技能。此外，還需要對虛擬化和網路概念有深入的瞭解。

採用元件

本文件所述內容不限於特定軟體和硬體版本。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

本指南概述了RHOSP與統一計算系統(UCS)基礎設施的整合，重點在於可擴充性、可靠性和效能最佳化。

它詳細說明了最佳實踐，並使用基於指令碼的自動化來部署OpenStack TripleO（包括Undercloud和Overcloud架構）。

通過使用本部署指南，組織可以實現為基於思科虛擬資料包核心 — 分散式例項(VPC-DI)的移動虛擬網路功能(VNF)量身定製的強大高效的RHOSP雲基礎設施。

瞭解RHOSP

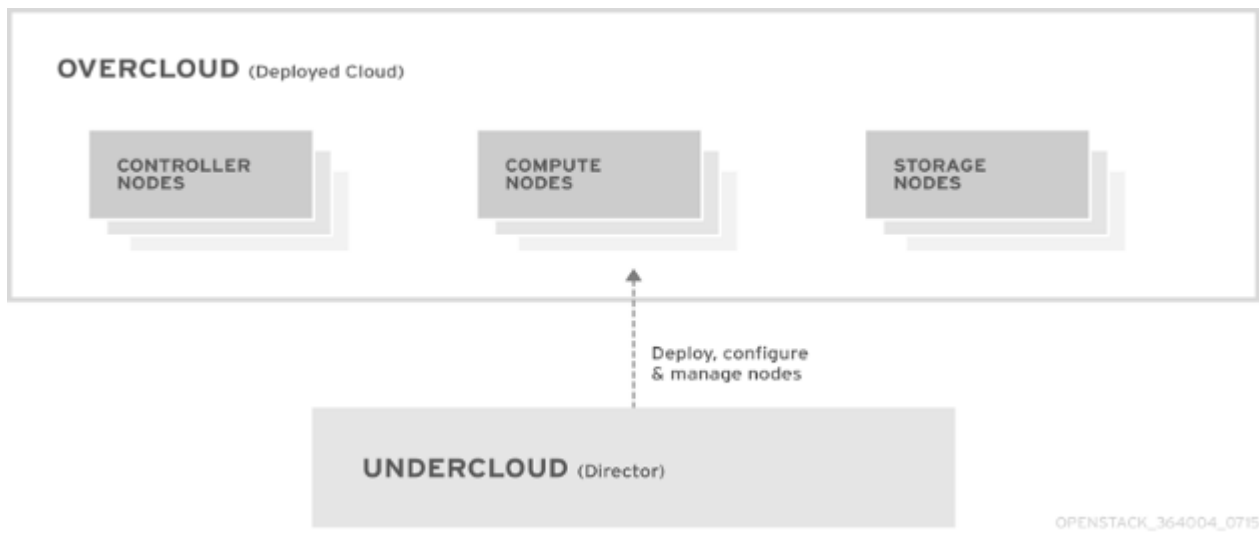
RHOSP是在開源OpenStack專案基礎上構建的企業級私有雲解決方案，由Red Hat整合和支援。它允許組織按需部署和管理虛擬機器(VM)、網路和儲存的基礎架構即服務(IaaS)。

它提供高可用性(HA)、網路功能虛擬化和可定製部署等功能。

RHOSP架構

RHOSP主要基於OpenStack TripleO專案。Openstack使用Director，它用作安裝和管理完整RHOSP環境的工具集。

RHOSP旨在提供可擴展且靈活的雲基礎架構。其體系結構由兩個主要元件組成：Undercloud和Overcloud。



Undercloud

Undercloud是包含RHOSP導向器工具集的主管理節點。它是單系統OpenStack安裝，包括用於調配和管理OpenStack節點(Overcloud)的元件。

Undercloud元件

Undercloud使用OpenStack元件作為其基本工具集。每個元件在底層雲上的獨立容器中運行：

- OpenStack身份(Keystone) — 為控制器元件提供身份驗證和授權
- OpenStack裸機 (諷刺) — 管理裸機節點
- OpenStack網路 (中子) 和開放式vSwitch — 裸機節點的控制網路
- OpenStack協調 (短暫散熱) — 在控制器將覆蓋雲映像寫入磁碟後提供節點協調

過雲

重疊雲是使用undercloud建立的結果RHOSP環境。這包括根據客戶要建立的OpenStack平台(OSP)環境定義的不同節點角色。

控制器

控制器節點為OpenStack環境提供管理、網路和HA。建議的OpenStack環境在HA群集中同時包含三個控制器節點。

計算

計算節點為OpenStack環境提供計算資源。隨著時間的推移，計算節點可以根據網路需求進行擴展/擴展。預設計算節點包含以下提及的元件：

- OpenStack計算 (新星)
- 基於核心的虛擬機器(KVM)/快速模擬器(QEMU)
- 開啟vSwitch

儲存

儲存節點為OpenStack環境提供儲存。



附註：在客戶網路中部署Undercloud/Director和Offline Repository(REPO)有多種方法 — 可以直接在裸機節點上部署，也可以作為VM部署在KVM虛擬機器監控程式之上。在當前部署指南中，Director UCS伺服器託管KVM (虛擬機器監控程式) 以在頂部部署多個VM。RHOSP Director節點和Offline-REPO節點部署為KVM虛擬機器監控程式上的虛擬機器。

本地儲存庫 (REPO伺服器)

Redhat提供了一個名為reposync的實用程式，可用於從內容交付網路([Content Delivery Network](#), CDN)下載包。為了從特定頻道下載所有程式包，系統必須訂閱該頻道。如果系統未訂閱所需的通道，則reposync無法在本地系統上下載並同步這些包。

在/etc/yum.repos.d/path中，儲存庫由以.repo副檔名結尾的文件進行配置。可以在同一檔案中定義多個儲存庫。

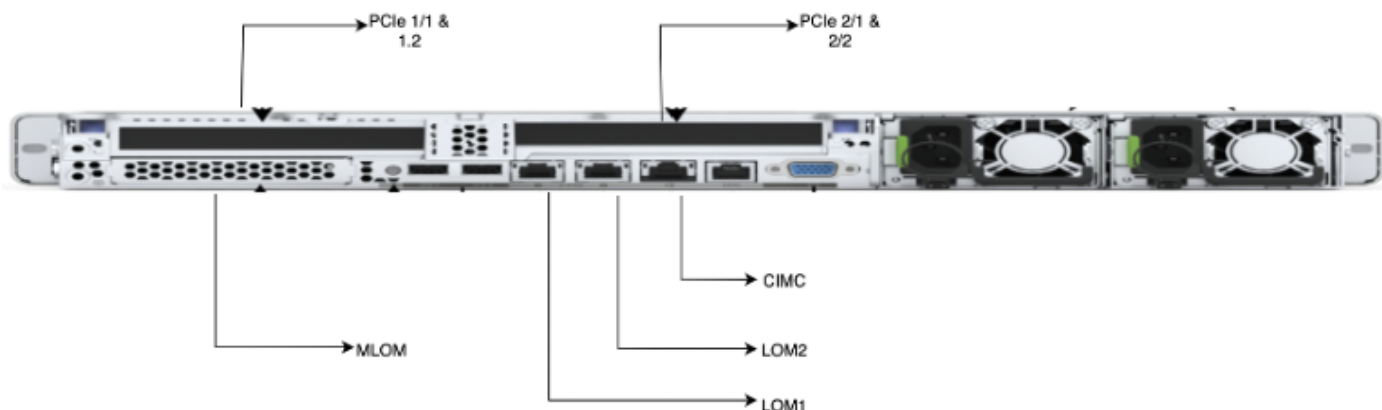
RHOSP網路

網路服務(neutron)是RHOSP的軟體定義網路(SDN)元件。RHOSP網路服務管理進出虛擬機器例項的內部和外部流量，並提供路由、分段、DHCP和後設資料等核心服務。它為虛擬網路功能以及交換機、路由器、埠和防火牆的管理提供API。

RHOSP控制器將OpenStack服務對映到不同的隔離網路。承載每種流量型別的網路包括：Cisco Integrated Management Controller(CIMC)、Provisioning、Internal API、Storage Data、Storage Management、Tenant and External(Secure Shell , SSH)和Operations、Administration and Maintenance(OAM)。

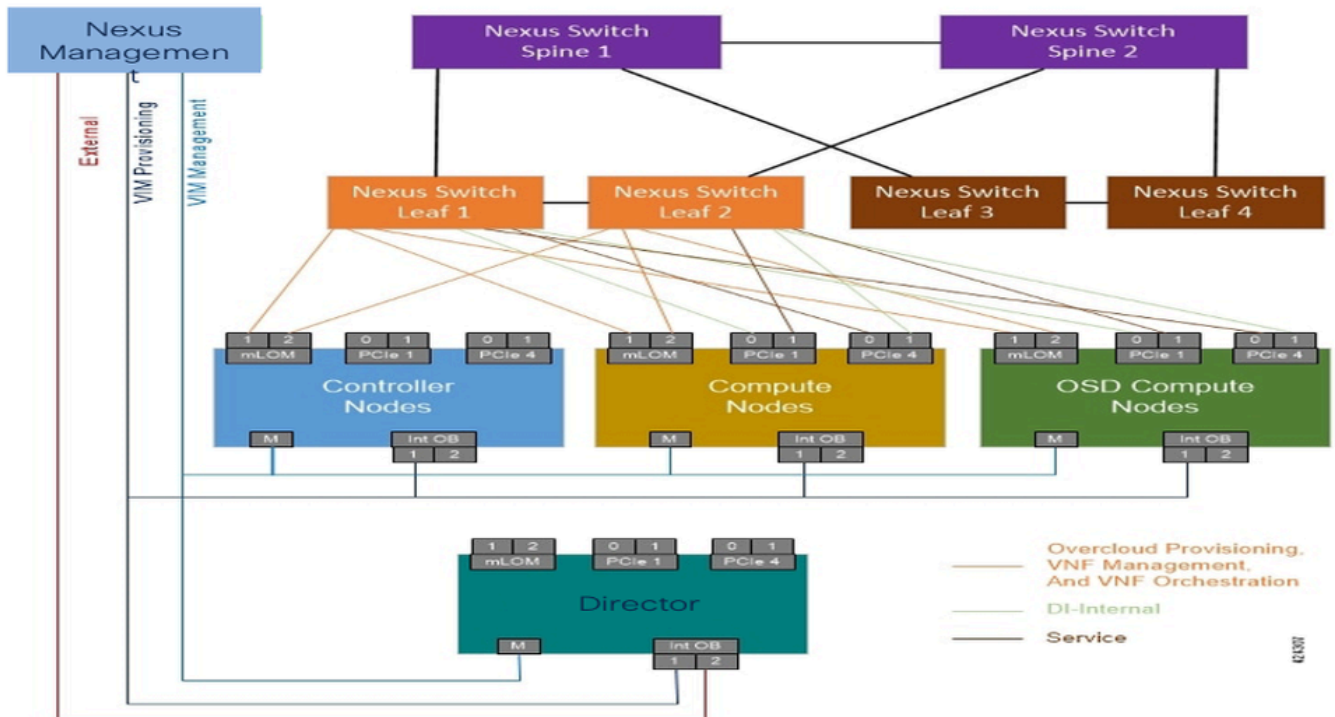
RHOSP物理連線

RHOSP部署使用Cisco UCS C220 M6伺服器的不同物理埠，用於不同的連線目的。



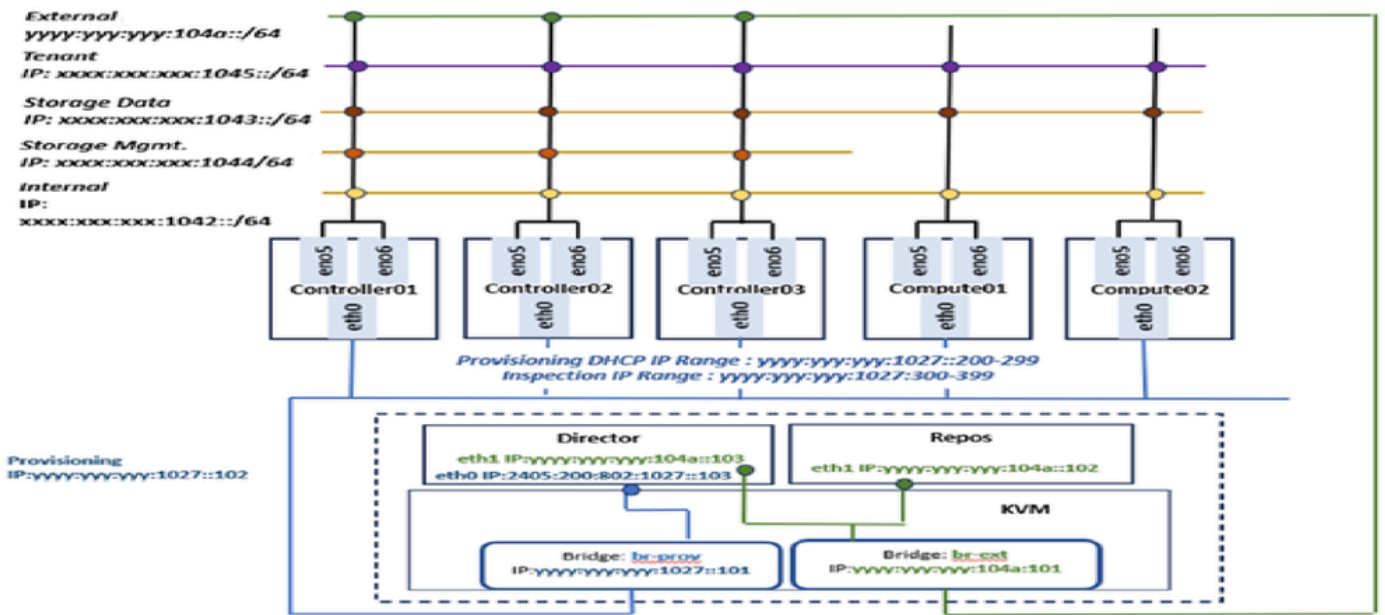
序列號	實體連線埠	詳細資料
1.	CIMC	CIMC為伺服器調配和管理提供帶外連線。
2.	單根I/O虛擬化(SR-IOV)/外圍元件互連Express(PCIe)	PCIe網路介面卡(NIC)用於DI內部網路的計算節點和VNF的服務網路。
3.	主機板上的模組化Lan(MLOM)	MLOM埠配置為繫結。 osp_external、osp_internal、osp_tenant、osp_external、osp_storage_data、osp_storage_mgmt使用MLOM埠進行內部通訊。
4.	Lan on MotherBoard(LOM)	指揮交換機使用LOM1和LOM2埠，而電腦和控制器僅使用LOM1埠。 LOM1用於在所有伺服器上部署或調配Openstack。 LOM2用作控制器上的OAM (外部網路)。

該圖顯示了與伺服器的物理連線。



RHOSP邏輯連線

RHOSP網路有多個子網，用於滿足雲中的不同服務。



- OSP_CIMC:

CIMC是智慧程式設計管理介面(IPMI)，控制所有UCS伺服器的管理。此CIMC網路配置在所有UCS伺服器的獨立CIMC埠上。

- OSP_Provisioning:

此網路負責在Overcloud部署期間對電腦和控制器伺服器進行調配和預引導執行環境(PXE)引導管理，並負責獲取DHCP IP。為了簡單和相容，調配網路在所有UCS伺服器的LOM1埠上配置為本徵VLAN。此調配網路負責在所有伺服器上部署雲。

由於導向器伺服器上的虛擬化，需要在KVM上建立橋接網路，以便導向器虛擬機器與其他伺服器通訊。

- OSP_Internal:

內部API網路用於在OpenStack服務（如neutron、nova、keystone等）之間進行通訊。

OSP_Internal網路在控制器和計算節點上的繫結MLOM埠上配置。

- OSP_Tenant:

租戶網路預設情況下在雲專案中建立，用於VNF管理。在當前設定中，僅為VNF部署建立單個Openstack專案。

OSP_Tenant網路在控制器和計算節點上的繫結MLOM埠上配置。

- OSP_External:

外部網路用於所有外部訪問（如SSH）和API網路。

OSP_External網路在控制器節點的LOM2埠上配置，並在控制器節點和計算節點上的繫結MLOM埠上配置。

- OSP_Storage_Data:

OSP_Storage網路用於與訪問儲存相關的所有操作。對於需要訪問儲存的CEPH服務和VNF之間的通訊，這是必需的。它被控制器、計算節點和CEPH使用。

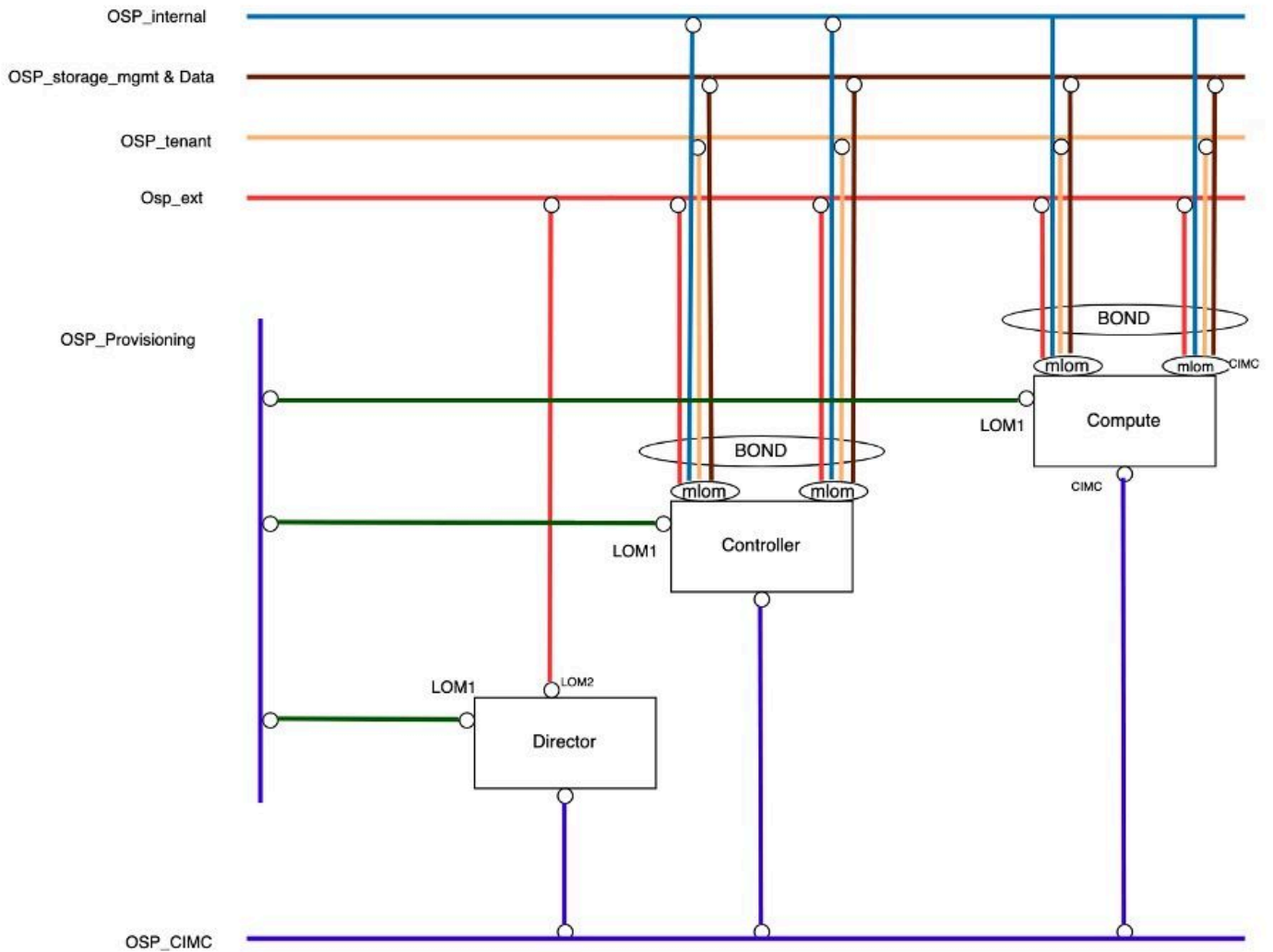
OSP_Storage_Data網路配置在控制器和計算節點上的繫結MLOM埠上。

- OSP_Storage_Management:

OpenStack對象儲存使用此網路在儲存群集中參與副本節點之間（在控制器 — 計算節點之間形成）同步資料對象。

OSP_Storage_Mgmt網路在控制器和計算節點上的繫結MLOM埠上配置。

該圖顯示了雲下邏輯網路如何連線到RHOSP群集中的每種型別的節點。



硬體引數調整

- 根據設計，確保啟用了IP並配置了正確的管理地址。
- 確保管理員使用者及其密碼對於所有CIMC登入訪問必須相同。
- 確保將Bootstrap(Boot)模式設定為Unified Extensible Firmware Interface(UEFI)。
- 確保使用推薦的引數設定BIOS設定。
- 將Boot-Order設定為：

1. LOM-PXE(eno1)
2. 引導硬碟驅動器(HDD)

- 設定LOM1(eno1)的mac地址。
- 確保所有UCS/CIMC伺服器均升級為最新的推薦韌體版本。

虛擬機器監控程式 — KVM安裝和網路建立

在客戶網路中部署Undercloud/Director和Offline REPO有多種方法。這些虛擬機器可以直接部署在裸機節點上，也可以作為KVM虛擬機器監控程式上運行的VM進行部署。

在當前部署指南中，Director UCS伺服器配置為託管KVM虛擬機器監控程式，這有助於建立多個VM。RHOSP Director節點和離線回購(Offline REPO)節點部署為此KVM虛擬機器監控程式上的VM。



附註：必須執行標準的RHEL KVM安裝步驟才能部署KVM虛擬機器監控程式。

- KVM啟動後，更新主機名：
`hostnamectl set-hostname <hostname> —static`
- 配置外部網橋和調配網橋並繫結介面。

br-prov :eth0

br-ext :eth1

必須通過Network Manager Text User Interface(NMTUI)GUI建立這些網橋。

- 預設情況下，在KVM中為物理LOM1和LOM2埠建立eno1和eno2埠，使用CIMC日誌記錄中的MAC地址交叉驗證這些埠。
- 建立網橋網路並通過對映正確的MAC地址新增從屬埠。
- 網橋網路建立後，確保可從KVM訪問調配和外部網關。

REPO和導向器VM建立

建立REPO和Director VM的前提條件

```
# dnf install qemu-kvm libvirt virt-install virt-manager virt-viewer libguestfs-tools
```

- 在KVM中安裝所需軟體包以建立REPO和Director VM。

```
! command not found...
[root@RH01KVM01 /]#
[root@RH01KVM01 /]# dnf install qemu-kvm libvirt virt-install virt-manager virt-viewer libguestfs-tools
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.

Red Hat Enterprise Linux 8.4.0 Appstream                               |
Red Hat Enterprise Linux 8.4.0 BaseOS                               | 564 MB/s | 6.8 MB  00:00
Package qemu-kvm-15:4.2.0-48.module+el8.4.0+10360+630e003b.x86_64 is already installed.
Package libvirt-6.0.0-35.module+el8.4.0+10230+7a9b21e4.x86_64 is already installed.
Package virt-install-2.2.1-4.el8.noarch is already installed.
Package virt-manager-2.2.1-4.el8.noarch is already installed.
Package virt-viewer-9.0-9.el8.x86_64 is already installed.
Dependencies resolved.
-----
Package                               Archite
```

- 在KVM上建立所需的目錄。

```
# mkdir /data # mkdir /data/offlineRepos # mkdir /data/isoImages # mkdir /data/qcow2Images # mkdir /data/images
```

- 將檔案複製到目錄。

```
# scp -r root@[remote-IP]:/root/rhel-8.4-x86_64-dvd.iso /data/isoImages/
# scp -r root@[remote-IP]:/root/offlineRepos/RHEL8.4 /data/offlineRepos/
# scp -r root@[remote-IP]:/etc/yum.repos.d/offlinedvd.repo /etc/yum.repos.d/
# scp -r root@[remote-IP]:/root/rhel-8.4-x86_64-kvm.qcow2 /data/qcow2Images/
# scp -r root@[remote-IP]:/root/OSREPO_RHEL_84.qcow2 /data/images/
# scp -r root@[remote-IP]:/root/OSREPO_DIRECTOR_84.qcow2 /data/images/
```

- 將ISO安裝到/mnt/iso。

```
# mount -t iso9660 -o loop /data/isoImages/rhel-8.4-x86_64-dvd.iso /mnt/iso
```

- 在/etc/yum.repos.d path處建立REPO檔案。

```
# cat /etc/yum.repos.d/offlinedvd.repo

[RHEL8.4_Appstream]
name=Red Hat Enterprise Linux 8.4.0 Appstream
mediaid=None
metadata_expire=-1
gpgcheck=0
enabled=1
```

```
baseurl=file:///data/offlineRepos/RHEL8.4/AppStream/
```

```
[RHEL8.4_BaseOS]
```

```
name=Red Hat Enterprise Linux 8.4.0 BaseOS
```

```
mediaid=None
```

```
metadata_expire=-1
```

```
gpgcheck=0
```

```
enabled=1
```

```
baseurl=file:///data/offlineRepos/RHEL8.4/BaseOS/
```

- 檢查重新輪詢清單並確保對映了Appstream和Baseos REPO。

```
# dnf repolist
```

```
[root@MUMBKVMC01 /]#  
[root@MUMBKVMC01 /]#  
[root@MUMBKVMC01 /]# dnf repolist  
Updating Subscription Management repositories.  
Unable to read consumer identity  
  
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.  
  
repo id                                repo name  
RHEL8.4_Appstream                      Red Hat Enterprise Linux 8.4.0 Appstream  
RHEL8.4_BaseOS                         Red Hat Enterprise Linux 8.4.0 BaseOS  
[root@MUMBKVMC01 /]#  
[root@MUMBKVMC01 /]#  
[root@MUMBKVMC01 /]#
```

建立回購VM

```
$ cd /var/lib/libvirt/images/
```

```
$ export LIBGUESTFS_BACKEND=direct
```

```
$ virt-customize -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 --root-password password:Cisco@12
```

```
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# virt-customize -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 --root-password password:Cisco@123  
[ 0.0] Examining the guest ...  
[ 6.6] Setting a random seed  
[ 6.7] Setting the machine ID in /etc/machine-id  
[ 6.7] Setting passwords  
[ 7.7] Finishing off  
[root@MUMBKVMC01 images]#
```

```
$ virt-filesystems --long -h --all -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-filesystems --long -h --all -a /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2
Name      Type      VFS      Label  MBR      Size      Parent
/dev/sda1 filesystem unknown  -      -        1.0M     -
/dev/sda2 filesystem vfat    -      -        100M    -
/dev/sda3 filesystem xfs     root   -        9.9G    -
/dev/sda1 partition -       -      -        1.0M    /dev/sda
/dev/sda2 partition -       -      -        100M    /dev/sda
/dev/sda3 partition -       -      -        9.9G    /dev/sda
/dev/sda  device   -       -        10G     -
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

- 建立REPO服务器的映像。

```
$ qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_osprepo.qcow2 500G
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_osprepo.qcow2 300G
Formatting '/var/lib/libvirt/images/rhel_84_osprepo.qcow2', fmt=qcow2 size=322122547200 cluster_size=65536 lazy_refcounts=off refcount_bits=16
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

```
$ virt-resize --expand /dev/sda3 /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 /var/lib/libvirt/ima
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_osprepo.qcow2 300G
Formatting '/var/lib/libvirt/images/rhel_84_osprepo.qcow2', fmt=qcow2 size=322122547200 cluster_size=65536 lazy_refcounts=off refcount_bits=16
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-resize --expand /dev/sda3 /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 /var/lib/libvirt/images/rhel_84_osprepo.qcow2
[  9.0] Examining /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2
*****
Summary of changes:
/dev/sda1: This partition will be left alone.
/dev/sda2: This partition will be left alone.
/dev/sda3: This partition will be resized from 9.9G to 299.9G. The
filesystem xfs on /dev/sda3 will be expanded using the 'xfs_growfs'
method.
*****
[  2.0] Setting up initial partition table on /var/lib/libvirt/images/rhel_84_osprepo.qcow2
[ 12.7] Copying /dev/sda1
[ 12.7] Copying /dev/sda2
[ 12.8] Copying /dev/sda3
100% |
[ 20.8] Expanding /dev/sda3 using the 'xfs_growfs' method
00:00
Resize operation completed with no errors. Before deleting the old disk,
carefully check that the resized disk boots and works correctly.
[root@MUMBKVMC01 images]#
```

```
$ qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_osprepo.qcow2 -F qcow2 /data/images/OSP
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_osprepo.qcow2 -F qcow2 /data/images/OSP_REPO_RHEL_84.qcow2
Formatting '/data/images/OSP_REPO_RHEL_84.qcow2', fmt=qcow2 size=322122547200 backing_file=/var/lib/libvirt/images/rhel_84_osprepo.qcow2 backing_fmt=qcow2 cluster_size=65536 lazy_re
fcounts=off refcount_bits=16
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

```
$ guestfish -a /data/images/OSP_REPO_RHEL_84.qcow2 -i ln-sf /dev/null /etc/systemd/system/cloud-init.ser
```

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# guestfish -a /data/images/OSPREPO_RHEL_84.qcow2 -t ln-sf /dev/null /etc/systemd/system/cloud-init.service
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

\$ osinfo-query os | grep rhel8

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# osinfo-query os | grep rhel8
** (osinfo-query:49279): WARNING **: 17:12:33.856: Entity http://pcisig.com/pci/1835/0100 referenced but not defined
rhel8-unknown | Red Hat Enterprise Linux 8 Unknown | 8-unknown | http://redhat.com/rhel/8-unknown
rhel8.0 | Red Hat Enterprise Linux 8.0 | 8.0 | http://redhat.com/rhel/8.0
rhel8.1 | Red Hat Enterprise Linux 8.1 | 8.1 | http://redhat.com/rhel/8.1
rhel8.2 | Red Hat Enterprise Linux 8.2 | 8.2 | http://redhat.com/rhel/8.2
rhel8.3 | Red Hat Enterprise Linux 8.3 | 8.3 | http://redhat.com/rhel/8.3
rhel8.4 | Red Hat Enterprise Linux 8.4 | 8.4 | http://redhat.com/rhel/8.4
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

\$ virt-install --cpu host --memory 32768 --vcpus 16 --os-variant rhel8.4 --disk path=/data/images/OSPREPO_RHEL_84.qcow2

```
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virt-install --cpu host --memory 32768 --vcpus 16 --os-variant rhel8.4 --disk path=/data/images/OSPREPO_RHEL_84.qcow2,device=disk,bus=virtio,format=qcow2
--import --noautoconsole --vnc --network bridge:br-ext --name OSPREPO_RHEL_84
** (process:49296): WARNING **: 17:13:15.813: Entity http://pcisig.com/pci/1835/0100 referenced but not defined
Starting install...
Domain creation completed.
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

\$ virsh list --all

```
root@MUMBKVMC01 images]#
root@MUMBKVMC01 images]# virsh list --all
Id Name State
-----
1 OSPREPO_RHEL_84 running
root@MUMBKVMC01 images]#
root@MUMBKVMC01 images]#
root@MUMBKVMC01 images]# virsh list --all
Id Name State
-----
1 OSPREPO_RHEL_84 running
root@MUMBKVMC01 images]#
```

Director VM建立

- 建立Director伺服器的映像。

```
# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 500G
```

```
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 200G  
Formatting '/var/lib/libvirt/images/rhel_84_ospdirector.qcow2', fmt=qcow2 size=214748364800 cluster_size=65536 lazy_refcounts=off refcount_bits=16  
[root@MUMBKVMC01 images]#
```

```
# virt-resize --expand /dev/sda3 /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 /var/lib/libvirt/ima
```

```
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# virt-resize --expand /dev/sda3 /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2 /var/lib/libvirt/images/rhel_84_ospdirector.qcow2  
[ 0.0] Examining /var/lib/libvirt/images/rhel-8.4-x86_64-kvm.qcow2  
*****  
Summary of changes:  
  
/dev/sda1: This partition will be left alone.  
  
/dev/sda2: This partition will be left alone. |  
  
/dev/sda3: This partition will be resized from 9.9G to 199.9G. The  
filesystem xfs on /dev/sda3 will be expanded using the 'xfs_growfs'  
method.  
*****  
[ 2.0] Setting up initial partition table on /var/lib/libvirt/images/rhel_84_ospdirector.qcow2  
[ 12.7] Copying /dev/sda1  
[ 12.7] Copying /dev/sda2  
[ 12.7] Copying /dev/sda3  
100% [ 20.6] Expanding /dev/sda3 using the 'xfs_growfs' method 00:00  
Resize operation completed with no errors. Before deleting the old disk,  
carefully check that the resized disk boots and works correctly.  
[root@MUMBKVMC01 images]#
```

```
# qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 -F qcow2 /data/images/0
```

```
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# qemu-img create -f qcow2 -b /var/lib/libvirt/images/rhel_84_ospdirector.qcow2 -F qcow2 /data/images/OSPDIRECTOR_RHEL_84.qcow2  
Formatting '/data/images/OSPDIRECTOR_RHEL_84.qcow2', fmt=qcow2 size=214748364800 backing_file=/var/lib/libvirt/images/rhel_84_ospdirector.qcow2 backing_fmt=qcow2 cluster_size=65536  
lazy_refcounts=off refcount_bits=16  
[root@MUMBKVMC01 images]#
```

```
# guestfish -a /data/images/OSPDIRECTOR_RHEL_84.qcow2 -i ln-sf /dev/null /etc/systemd/system/cloud-init  
# virt-install --cpu host --memory 131072 --vcpus 32 --os-variant rhel8.4 --disk path=/data/images/OSPD
```

```
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# virt-install --cpu host --memory 131072 --vcpus 32 --os-variant rhel8.4 --disk path=/data/images/OSPDIRECTOR_RHEL_84.qcow2,device=disk,bus=virtio,format=qcow2 --import --noautoconsole --vnc --network bridge:br-prov --network bridge:br-ext --name OSPDIRECTOR_RHEL_84  
** (process:49762): WARNING **: 17:15:52.006: Entity http://pcisig.com/pci/1836/0100 referenced but not defined  
  
Starting install...  
Domain creation completed.  
[root@MUMBKVMC01 images]#
```

```
# virsh list --all
```

```
Domain creation completed.
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]# virsh list --all
Id   Name                               State
-----
 1   OSPREPO_RHEL_84                    running
 2   OSPDIRECTOR_RHEL_84                running

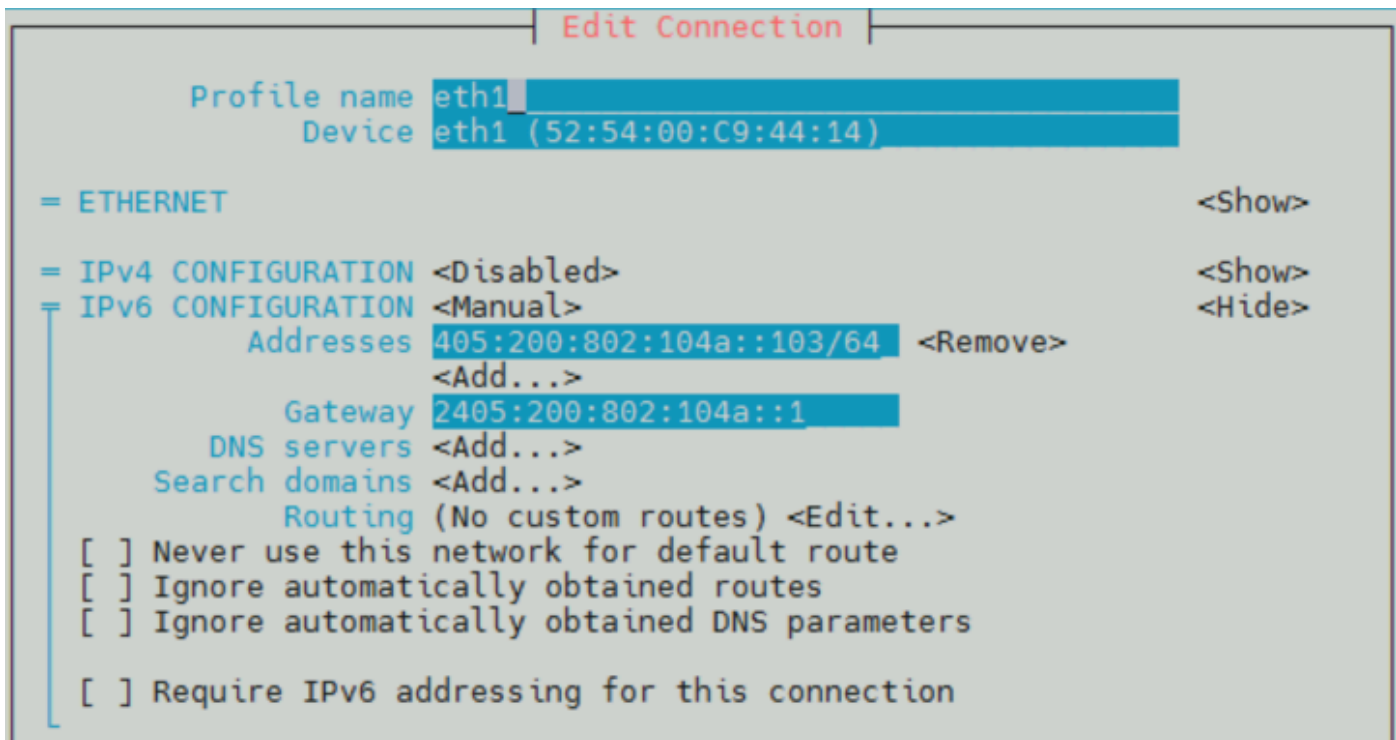
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
[root@MUMBKVMC01 images]#
```

```
[root@MAGKVMC02 images]# virt-install --name OSPDIRECTORCL02_RHEL --description "director" --os-variant rhel8.4 --disk path=/data/images/OSPDIRECTORCL02_RHEL_84.qcow2,size=500,device=disk,bus=virtio,format=qcow2 --cpu host --memory 131072 --vcpus 32 --location /data/isoImages/rhel-8.4-x86_64-dvd.iso --network bridge:br-ext --network bridge:br-prov --extra-args console=ttyS0 --boot uefi
** (process:193050): WARNING **: 15:12:45.915: Entity http://pcisig.com/pci/1836/0100 referenced but not defined
Starting install...
Retrieving file vmlinuz... | 9.6 MB 00:00:00
Retrieving file initrd.img... | 72 MB 00:00:00
Allocating 'OSPDIRECTORCL02_RHEL_84.qcow2' | 500 GB 00:00:00
(virt-viewer:193156): GLib-GIO-CRITICAL **: 15:12:48.336: g_dbus_proxy_new_sync: assertion 'G_IS_DBUS_CONNECTION (connection)' failed
(virt-viewer:193156): GSpice-WARNING **: 15:12:49.080: PulseAudio context failed Connection refused
(virt-viewer:193156): GSpice-WARNING **: 15:12:49.080: pa_context_connect() failed: Connection refused
(virt-viewer:193156): GSpice-WARNING **: 15:12:49.213: Could not create org.gnome.SessionManager dbus proxy: Could not connect: Connection refused
(virt-viewer:193156): GSpice-WARNING **: 15:12:49.213: Warning no automount-inhibiting implementation available
(virt-viewer:193156): GLib-GObject-WARNING **: 15:13:08.438: value "64" of type "gint" is invalid or out of range for property 'desktop-width' of type 'gint'
(virt-viewer:193156): GLib-GObject-WARNING **: 15:13:08.438: value "64" of type "gint" is invalid or out of range for property 'desktop-height' of type 'gint'
```

- 開啟新視窗並進入VM的控制檯登入。

```
# virsh list --all
# virsh console <domain-id>
```

- 在導向器VM部署後，將建立兩個eth介面eth0和eth1。
 - eth0用於OSP_provisioning網路，其中br-ctlplane網路在Undercloud部署期間建立。
 - eth1用於SSH訪問的外部連線。因此，分配了OSP_external子網IP地址。



建立本地離線回購

REPO伺服器必須向Redhat CDN註冊，並且必須擁有部署所需的所有可用的RHOSP 16.2軟體包的儲存庫。必須使用代理將RHEL RPM軟體包和RHOSP容器映像下載到REPO VM。

RHOSP雲部署

RHOSP 16.2通過自動化部署在客戶網路中。Ansible指令碼用於自動部署Undercloud和Overcloud。

必要條件

開始實際雲部署之前要遵循的步驟：

1. 確保KVM節點與OSP_EXT網路上的本地回購VM和導向器VM的連線。
2. 確保將所有可能的指令碼和自動化指令碼上傳到指定資料夾的KVM主機。
3. 建立名稱為「cisco」和「automation」的資料夾，然後放置可觸發指令碼的跳板。

```
# cd /home
# mkdir cisco
# cd /home/cisco
# mkdir automation
```

```
# cd /home/cisco/automation
```

tarball將包含三個資料夾目錄結構，名稱為：

- 指令碼：包含用於裸機節點配置的shell指令碼。
- rpms:包括英特爾乙太網連線(ICE)驅動程式、英特爾驅動程式和Red-hat Package Manager(RPM)軟體包。
- ansible:包含輸入vars檔案、用於部署的ansible yami檔案和jinja-templates。

4.安裝sshpass軟體包。sshpass是一個命令列實用程式，用於以非互動方式為ssh提供密碼。它主要用於手動輸入密碼不可行的指令碼或自動化方案。

- 從Internet/現有伺服器下載sshpass包sshpass.tar.gz。
- 安裝GNU編譯器集合(GCC)軟體包。

```
# yum install gcc
```

- 安裝make程式包。

```
# yum install make
```

- 開啟sshpass軟體包並安裝SSH pass。

```
# tar -xvzf sshpass.tar.gz
# cd sshpass-1.10/
# ./configure
# sudo make install
# sshpass -V
```

5.控制器安裝過程需要非根使用者執行命令。必須在具有sudo訪問許可權的導向器VM中建立「堆疊」使用者。

```
# useradd stack
# passwd stack
```

```
Disable password requirements for the 'stack' user when using sudo.
# echo "stack ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/stack
# chmod 0440 /etc/sudoers.d/stack
```

6.將rootCA.crt檔案從回購伺服器複製到指定路徑上的導向器VM和KVM。此外，更新信任清單中的回購VM證書。

```
# /etc/pki/ca-trust/source/anchors
# update ca-trust
```

7.在/etc/hosts檔案中更新導向器VM和KVM中的本地REPO服務器主機名詳細資訊。

8.在KVM和Director VM上，安裝其他軟體包（如python、ansible等）以執行ansible自動化指令碼。

```
# dnf install python3 python3-devel ansible httpd -y
# update-alternatives --set python /usr/bin/python3
```

9.必須從導向器的調配網路訪問CIMC子網，以便在雲部署期間啟用調配。如果需要，請為同一路由新增靜態路由。

```
# ip -6 route add <CIMC Subnet> via <Provisioning Subnet>
```

10.在KVM和導向器VM中，在/ansible資料夾下建立主機檔案，並根據需要新增堆疊特定詳細資訊。

```
[ospd]
```

```
# <PODNAME> ansible_host=<OSPD IP> ansible_ssh_user=stack ansible_ssh_pass='<STACKPASSWD>' ansible_ssh_
```

```
<podname> - Stack Name of the Cloud.
```

```
<OSPD IP> - Baremetal OSPD Node IP Address
```

```
<STACKPASSWD> - OSPD Node password for 'stack' user
```

11.確保所有可能的行動手冊和輸入檔案必須儲存在導向器VM的/home/stack資料夾下方。

更新輸入檔案

有一個輸入變數檔案，其中包含必須針對雲部署準備的客戶網路特定詳細資訊。

路徑：/home/cisco/automation/ansible/podvars

檔名 : <stack-name>_vars.yml

根據站點特定的IP計畫/低級設計文檔更新突出顯示的引數。



附註：虛擬IP地址僅用於表示目的。

```
<#root>
```

```
# #####  
# XR21 Specific Variables  
# #####
```

```
# =====  
# Common Variables  
# =====
```

```
# UCS hardware type: 'm4/m5/m6'  
hardware: m6
```

```
# Platform type: 'epc/pcrf'  
platform: epc
```

```
# RHEL version  
rhel: { version: 84, tag: 8.4 }
```

```
# Openstack version  
osp: { version: 16, major: 2 }
```

```
# Container version  
container: { tag: 16.2, tools: 3.0 }
```

```
# Overcloud stack name
```

```
stack_name: '
```

```
,
```

```
# OSPD full hostname
```

```
fqdn_hostname: '
```

```
.epdg.ap.hamb.a6.cloud.com'
```

```
# OSPD host login
```

```
ospd_host: { ip:
```

```
'2405:XXXX:089:1054::11'
```

```
, username: 'stack', password: '*****' }
```

```
# OSPD cimc login
```

```
ospd_cimc: { ip:
```

```
'2405:XXXX:089:1054::11'
```

```
, username: 'admin', password: '*****' }
```

```
# CIMC username and password must be same across all Overcloud nodes
```

```
cimc: { username: 'admin', password: '*****', ip_pool: '2405:XXXX:089:1055::/64' }
```

```
# Undercloud-Overcloud provision
```

```
internal_network: {
```

```
  ip_type: 'v6',
```

```
  local_interface: 'eth0',
```

```
  local_ip: '2405:XXXX:089:1041::103',
```

```
  undercloud_public_host: '2405:XXXX:089:1041::105',
```

```
  undercloud_admin_host: '2405:XXXX:089:1041::104',
```

```
  cidr: '2405:XXXX:089:1041::/64',
```

```
  dhcp_start: '2405:XXXX:089:1041::200',
```

```
  dhcp_end: '2405:XXXX:089:1041::299',
```

```
gateway: `2405:XXXX:089:1041::199`,
```

```
# nexthop: `2405:XXXX:089:1041::1`,
```

```
inspection_iprange_start: `2405:XXXX:089:1041::300`,
```

```
inspection_iprange_end: `2405:XXXX:089:1041::399`,
```

```
}
```

```
# DNS
```

```
dns_ips: [ '2405:YYYY:a10:f100::1' ]
```

```
dns_search_domains: [ 'cloud.com' ]
```

```
# NTP
```

```
ntp_ips: [ '2405:YYYY:801:700::afa', '2405:YYYY:801:700::afb' ]
```

```
# Deployment type: 'offline/online'
```

```
repos: { rhel: 'offline', container: 'offline' }
```

```
# Offline details if repos is 'offline'
```

```
offline: {
```

```
    environment: 'v01_00',
```

```
    deliverymedia: '/home/stack/deliverymedia/'
```

```
}
```

```
# Satellite details if repos is 'online'
```

```
satellite: {
```

```
    fqdn_name: 'rh-satellite2.mitg-bxb300.cisco.com',
```

```
    ip: '10.XX.XX.XX',
```

```
    org: 'MITG',
```

```
    user: 'admin',
```

```
    password: '*****',
```

```
    environment: 'production',
```

```
    activation_key: 'ak-rhel{{rhel.version}}-osp{{osp.version}}{{osp.major}}',
```

```
    repos_file: 'rhel{{rhel.version}}osp{{osp.version}}{{osp.major}}.yaml'
```

```
}
```

```
# Offline container registry details
```

```
offline_registry: {
```

```
    ip: '2405:XXXX:089:1055::100',
```

```
    name: '
```

```
,
```

```
port: '5000',  
container_tag: '16.2.6',  
user: 'ciscoadmin',  
password: '*****'  
}
```

```
# Custom cloud domain details  
domain_name: {  
  domain: '  

```

```
,
```

```
cloudshortname: 'n1'  
}
```

```
# Container images namespace  
container_namespace: 'mitg-{{satellite.environment}}-cv-rhel{{rhel.version}}-osp{{osp.version}}{{osp.ma
```

```
# List of cimc IPs
```

```
ctrl_cimc_ip:
```

```
- 2405:xxxx:yyyy:1036::12
```

```
- 2405:xxxx:yyyy:1036::13
```

```
- 2405:xxxx:yyyy:1036::14
```

```
osdc_cimc_ip:
```

cmpt_cimc_ip:

- 2405:XXXX:YYYY:1037::17

- 2405:XXXX:YYYY:1038::18

- 2405:XXXX:YYYY:1038::19

- 2405:XXXX:YYYY:1038::20

mgmt_cimc_ip:

- 2405:XXXX:YYYY:1051::15

- 2405:XXXX:YYYY:1051::16

=====
Hardware Specific Variables
=====

Isolcpu for cpu pinning
isolcpus: { osdc: '4-31,36-63', cmpt: '2-31,34-63', mgmt: '2-31,34-63' }

Hugepages in 1G Pages
hugepages: { osdc: 428, cmpt: 448, mgmt: 448 }

Reserved host memory in MB
reserved_host_memory: { osdc: 84000, cmpt: 64000, mgmt: 64000 }

Number of VFs per SR-IOV port
sriov_vfs_per_port: 16

List of SR-IOV ports
sriov_port_list: [ens1f0, ens1f1, ens9f0, ens9f1]

```
# List of OVS bonding interface
ovs_bond_interface: [eno5, eno6]

# Physical networks
physical_network: [phys_pcie1_0, phys_pcie1_1, phys_pcie2_0, phys_pcie2_1]

# Boot disk size
boot_disk_mb_size: { ctrl: 761985, osdc: 761985, cmpt: 761985, mgmt: 1524925 }

# Boot disk PD slot number
boot_disk_pd_slot: { ctrl: [1,2], osdc: [1,2], cmpt: [1,2], mgmt: [1,2] }

# Boot disk VD slot number
boot_disk_vd_slot: { ctrl: 237, osdc: 235, cmpt: 239, mgmt: 239 }

# Storage backend 'swift' or 'ceph'
storage_backend: 'swift'

# Storage disk size
storage_disk_mb_size: { swift: 761985, ceph: 914573, journal: 0 }

# Storage disk PD slot number
storage_disk_pd_slot: { swift: [6,7], ceph: [3,4,5,6], journal: [0] }

# Storage disk VD slot number
storage_disk_vd_slot: { swift: [238,239], ceph: [236,237,238,239], journal: [0] }

# Firmware version 'yes' or 'no' ???
firmware: { check: 'no', bios_version: '4.2.3c', cimc_version: '4.2(3e)' }

# =====
# OSP Specific Variables
# =====

# Timezone for overcloud nodes

timezone: 'Asia/Kolkata'

# Overcloud node count to deploy

node_count: { ctrl: 3, osdc: 0, cmpt: 11, mgmt: 2 }

local_network: {
  ip_type: 'v6',
  tenant_vlan_id: 1045,
  tenant_net_cidr: '240f:ppp:rr:1045::/64',
  tenant_alloc_pools_start: '240f:ppp:rr:1045::10',
  tenant_alloc_pools_end: '240f:ppp:rr:1045:ffff:ffff:ffff:fffe',
```

```
storage_vlan_id: 1043,  
storage_net_cidr: '240f:ppp:rr:1043::/64',  
storage_alloc_pools_start: '240f:ppp:rr:1043::10',  
storage_alloc_pools_end: '240f:ppp:rr:1043:ffff:ffff:ffff:fffe',  
  
storage_mgmt_vlan_id: 1044,  
storage_mgmt_net_cidr: '240f:ppp:rr:1044::/64',  
storage_mgmt_alloc_pools_start: '240f:ppp:rr:1044::10',  
storage_mgmt_alloc_pools_end: '240f:ppp:rr:1044:ffff:ffff:ffff:fffe',  
  
internal_api_vlan_id: 1042,  
internal_api_net_cidr: '240f:ppp:rr:1042::/64',  
internal_api_alloc_pools_start: '240f:ppp:rr:1042::10',  
internal_api_alloc_pools_end: '240f:ppp:rr:1042:ffff:ffff:ffff:fffe'  
}
```

```
# External VLAN and IP configs
```

```
external_network: {
```

```
    ip_type: 'v6',
```

```
    vlan_id: 1046,
```

```
    default_route: '2405:XXXX:YYYY:1055::1',
```

```
    network_cidr: '2405:XXXX:YYYY:1055::/64',
```

```
    alloc_pool_start: '2405:XXXX:YYYY:1055::100',
```

```
    alloc_pool_end: '2405:XXXX:YYYY:1055::200',
```

```
    horizon_ip: '2405:XXXX:YYYY:1055::107'
```

```
}
```

```
# Neutron mechanism driver 'ovs' or 'ovn'
```

```
neutron: {
```

```
    driver: 'ovs',
```

```
    dvr: false,
```

```
    datacenter_vlan_start: 1050,
```

```
    datacenter_vlan_end: 1070
```

```
}
```

```

# =====
# OS Specific Variables
# =====

# RHEL kernel version

kernelversion: '4.18.0-305.88.1.el8_4.x86_64'

# E810 ICE driver
ice_driver: { check: 'yes', version: 1.12.6, intel_aux_version: 1.0.1 }

# ENIC and FNIC verison
nic_version: { enic: '2.3.0.53', fnic: '1.6.0.53' }

# IPMI watchdog timer config
watchdog: { action: enabled, version: '2.0.31-3.el8.x86_64', timer: 250 }

# StorCLI raid management
storcliver: '007.2612.0000.0000-1.noarch'

# =====
# Platform Specific Variables
# =====

# Buffer pool size based on platform type
innodb_buffer_pool_size: 1610
# #####
# END - XR21 Specific Variables
# #####

```

雲下部署

Undercloud使用ansible指令碼在七個步驟中進行部署。所有步驟都必須從充當跳轉主機的KVM主機執行。

步驟	Tag	說明	實戰手冊YAML
步驟 1.	checkfiles	在Openstack Platform Director(OSPD)上驗證所需的手冊、指令碼和RPM。	osp16_published_playbooks_verify.yml
步驟 2.	真普達瓦爾	生成與硬體、RHEL等相關的POD特定變數檔案，如common_vars.yml (硬體、軟體、網路詳細資訊)、hbm6_vars.yml (CPU、記憶體、	osp16_generate_pod_specific_vars.yml

		Hugepages、磁碟、NIC等)、 rhel_84_vars.yml (RHEL、核心、 ICE驅動程式、NIC版本)、 pf_esc_vars.yml(彈性服務控制器 (ESC)詳細資訊)、 osp_16_vars.ymlOSP版本、時區、 IP型別、VLAN ID、IP、中子詳細 資訊)。	
步驟 3.	preuc部署	配置完全限定域名(FQDN)、網路時間協定(NTP)，並更新控制器節點上的所有包。	osp16_pre_undercloud_deploy.yml
步驟 4.	firstreboot	在較早的配置和軟體包安裝後首次重新啟動Undercloud控制器節點。	osp16_undercloud_deploy.yml
步驟 5.	ucdeploy	在控制器上安裝Undercloud堆疊	osp16_undercloud_tuning.yml
步驟 6.	cstate	配置控制器節點上的BIOS CPU C狀態設定。	osp16_cstate.yml
步驟 7.	secondreboot	在BIOS更改後，在Undercloud director上第二次重新啟動。	不適用

名為osp16_auto_undercloud_deploy.yml的檔案是可在單個迭代中運行的主要可玩手冊，但建議使用不同標籤分步執行該手冊，以便在出現任何部署問題時輕鬆進行故障排除。

```
<#root>
```

```
#
```

```
cd /home/stack/ansible/
```

```
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=
```

```
TAG
```

```
For Ex -
```

```
#
ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=checkfiles
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=genpodvars
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=preucdeploy
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=firstreboot
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=ucdeploy
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=cstate
# ansible-playbook -i hosts osp16_auto_undercloud_deploy.yml -e podname=<> --tags=secondreboot
```

Note :-

Deployment Logs would be generated in “/home/stack/autologs” in Director-VM.

Post-Checks for Verification of Undercloud Deployment.
“stackrc” & “undercloud.conf” file must be generated in /home/stack folder.

```
# sudo podman ps -a
# source stackrc
# openstack stack list
# openstack stack show <stack-name> --fit
# openstack server list
# openstack network list
# openstack subnet list
```

超雲部署

Overcloud部署至少三個控制器處於HA模式，一個計算。使用Ansible指令碼在17個步驟中部署了Overcloud。所有步驟都必須從在此例中充當Jump-host的Director-VM執行。

步驟	Tag	說明	實戰手冊YAML
步驟 1.	真普達瓦爾	生成與硬體、RHEL等相關的Overcloud特定的POD變數檔案，如common_vars.yml (硬體、軟體、網路詳細資訊)、hw_m6_vars.yml (CPU、記憶體、Hugepages、磁碟、NIC等)、rhel_84_vars.yml (RHEL、核心、ICE驅動程式、NIC版本)、pf_esc_vars.yml (ESC詳細資訊)、osp_16_vars.yml (OSP版本) 時區、IP型別、VLAN ID、IP、中子詳細資訊)。	osp16_generate_pod_specific_vars.yml

步驟 2.	geninstack	<p>從上一步建立的 /var/common_vars.yml生成Instackenv JSON檔案。</p> <p>指揮交換機需要節點定義模板，該模板是手動建立的。此檔案instackenv.json使用JSON格式，包含節點的所有硬體和電源管理詳細資訊。此步驟還會在生成檔案之前驗證UCS伺服器上的硬體配置。</p>	osp16_generate_instackenv.yml
步驟 3.	cimcvd	<p>參考common_vars.yml、hw_m6_vars.yaml和rhel_84_vars.yml，在每個伺服器上配置CIMC設定和虛擬磁碟(VD)。</p>	osp16_cimc_vd_configure.yml
步驟 4.	preocdeploy	<p>此步驟執行部署超雲的所有先決條件。它設定FQDN、NTP並更新所有包，將映像推送到路徑以進行部署。</p>	osp16_pre_overcloud_deploy.yml
步驟 5	匯入節點	<p>在此步驟中，將內省伺服器CPU、記憶體、NIC以及介面和網路交換機上的埠。對所有控制器和電腦的已連線網路交換機執行內檢。</p>	osp16_import_ironic_nodes.yml
步驟 6.	gentemplates	<p>為控制器和電腦生成自定義模板檔案。在自定義模板中，為其上運行的所有服務定義控制器和計算角色。它還通過應用證書、路由等執行系統強化。</p>	osp16_generate_custom_templates.yml
步驟 7.	ocdeploy	<p>在此步驟中，完成Openstack Overcloud部署。運行Red Hat提供的deploy.sh，以進行RHOSP部署。</p>	osp16_overcloud_deploy.yml
步驟 8	geninventory	<p>在此步驟中，生成供ansible使用的清單yml檔案，其中預配IP、IPMI(CIMC)IP和憑證與控制器一起儲存和對映，並計算自動登入系統並執行進一步的步驟。</p>	osp16_build_inventory_v3.py
步驟	offlinerepo	<p>在檔案/etc/yum.repo.d/offline.repo中配置Overcloud Offline REPO，使其指向</p>	osp16_config_offline_repo.yml

9.		通過外部網路的回送伺服器。	
步驟 10.	圍欄	使用「Shoot The Other Node In The Head」(HA群集中的一種隔離技術) (STONITH)在所有控制器節點上配置隔離功能。	osp16_config_fencing.yml
步驟 11.	raidcache	為所有控制器和電腦配置Raid快取設定，並配置SWIFT儲存設定。	osp16_raid_cache_tuning.yml
步驟 12.	dnfupdate	對所有節點上的所有包運行DNF更新。	dnf_update_all_packages.yml
步驟 13.	setiplink	在此步驟中，為演化分組資料網關 (EPDG)內部和資料流量啟用SR-IOV埠的信任模式控制。對SR-IOV埠的支援可在neutron中獲得，並允許VM通過SR-IOV虛擬功能訪問網路。	osp16_setIpLink.yml
步驟 14.	監視程式	在此步驟中，控制器節點上的IPMI設定配置為通過帶外連線管理所有伺服器上的任務。	osp16_config_ipmi_watchdog.yml
步驟 15.	冰河	將外圍裝置元件互連(PCI)卡的英特爾E810 ICE驅動程式更新為EPDG的1.12.6版，以使用英特爾NIC埠作為SR-IOV。	osp16_ice_driver_install.yml
步驟 16.	重新引導	執行前面的步驟後，重新啟動所有重疊雲節點。	osp16_reboot_overcloud_hosts.yml
步驟 17.	verifyrhosp	驗證RHOSP部署配置和運行狀況。	osp16_rhosp_verify.yml

對於Overcloud節點的調配，Undercloud使用「overcloud-stronized-uefi-full.qcow2」。因此，開始超雲部署之前，映像必須儲存在欠雲/控制器中的指定路徑中。

從遠端站點複製Overcloud qcow2檔案。

```
<#root>
```

```
# su - stack
# cd /home/stack
# mkdir deliverymedia
# cd deliverymedia
```

```
### Copy overcloud-hardened-uefi-full.qcow2 to deliverymedia ###
```

```
# scp overcloud-hardened-uefi-full.qcow2 stack@[Director-IP]:/home/stack/deliverymedia
[stack@[stack@ Undercloud ~]$ cd /home/stack/ansible/
[stack@[stack@ Undercloud ansible]$ ansible-playbook osp16_auto_overcloud_deploy.yml -e podname=POD_NAME
```

For Ex -

```
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=genpodvars
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=geninstack
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=cimcvd
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=preocdeploy
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=importnodes
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=gentemplates
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=ocdeploy
```

```
#### Push & Update the rootCA.pem in all the Controllers & Computes ####
```

```
# for node in $(nova list | grep -i active | awk '{print $12}' | awk -F "=" '{print $2}'); do scp -o S
```

```
#### Append the Director Entry in "/etc/hosts" file ####
```

```
# for node in $(nova list | grep -i active | awk '{print $12}' | awk -F "=" '{print $2}'); do ssh -o S
```

```
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=geninventory
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=offlinerepo
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=fencing
```

```
### In case of Fencing Failures, please check the reachability of CIMC Subnet from Controllers ####
```

```
## If CIMC Subnet is not pinging, Do add the static Route ##
```

```
# ip -6 route add <CIMC Subnet> via <Provisioning Subnet>
Ex: ip -6 route add 2405:XXXX:YYY:9999::/64 via 2405:XXXX:YYY:9999:1

# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=raidcache
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=dnfupdate
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=setiplink
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=watchdog
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=icedriver
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=reboot
# ansible-playbook -i hosts osp16_auto_overcloud_deploy.yml -e podname=<> --tags=verifyrhaps
```

為了監視部署日誌，請使用最新的日誌檔案。

```
# tail -F </home/stack/autologs/osp16_auto_overcloud_deploy_*.log>
```

確保所有17個步驟均已通過。

檢查失敗=>計數必須為00。

日誌=> /home/stack/autologs/osp16_rhosp_verify.yml_20200703T042257.log

```
#####
```

```
#步驟 | 標籤 | 說明 | 攻略
```

```
#####
```

```
# step1 | genpodvars | 生成POD特定的變數檔案 |
osp16_generate_pod_specific_vars.yml -e podname=
```

```
# step2 | geninstack | 生成Instackenv JSON檔案 |
osp16_generate_instackenv.yml -e podname=
```

```
# step3 | cimcvd | 配置CIMC VD | osp16_cimc_vd_configure.yml
```

```
#步驟4 | preocdeploy | 配置預超雲部署 | osp16_pre_overcloud_deploy.yml
```

```
#第5步 | 匯入節點 | 匯入Openstack裸機Ironic節點 | osp16_import_ironic_nodes.yml
```

```
#第6步 | gentemplates | 生成自定義模板 |
```

osp16_generate_custom_templates.yml

#第7步 | ocdeploy | Openstack Overcloud Deploy |
osp16_overcloud_deploy.yml

step8 | geninventory | 生成清單檔案 | osp16_build_inventory_v3.py
—ipmipass

#步驟9 | offlinerepo | 從離線TAR檔案配置Overcloud離線回購 | osp16_config_offline_repo.yml

step10 | 圍欄 | 重新引導前部署MOP — 配置隔離功能 | osp16_config_fencing.yml

step11 | raidcache | 重新引導前部署MOP - Raid快取和PR調整 | osp16_raid_cache_tuning.yml

#步驟12 | dnfupdate | 重新引導前部署MOP - Dnf更新包 | dnf_update_all_packages.yml

#步驟13 | setiplink | 重新引導前部署MOP — 將VF IP鏈路信任設定為ON | osp16_setIpLink.yml

#步驟14 | 監視程式 | 在重新啟動之前部署MOP — 配置IPMI監視程式 |
osp16_config_ipmi_watchdog.yml

step15 | icedriver | 重新引導前部署MOP — 更新E810 ICE驅動程式 |
osp16_ice_driver_install.yml

#步驟16 | 重新啟動 | 重新啟動所有覆蓋雲節點 |
osp16_reboot_overcloud_hosts.yml

#步驟17 | verifyrhosp | 驗證RHOSP部署配置和運行狀況 | osp16_rhosp_verify.yml -e
podname=

#=====

所有主機均可訪問

=====

檢查完成=> 17

通過支票=> 17

檢查失敗=> 00

=====
總體狀態=>通過!!
=====

成功部署Overcloud後，確保可以訪問Horizon控制面板。

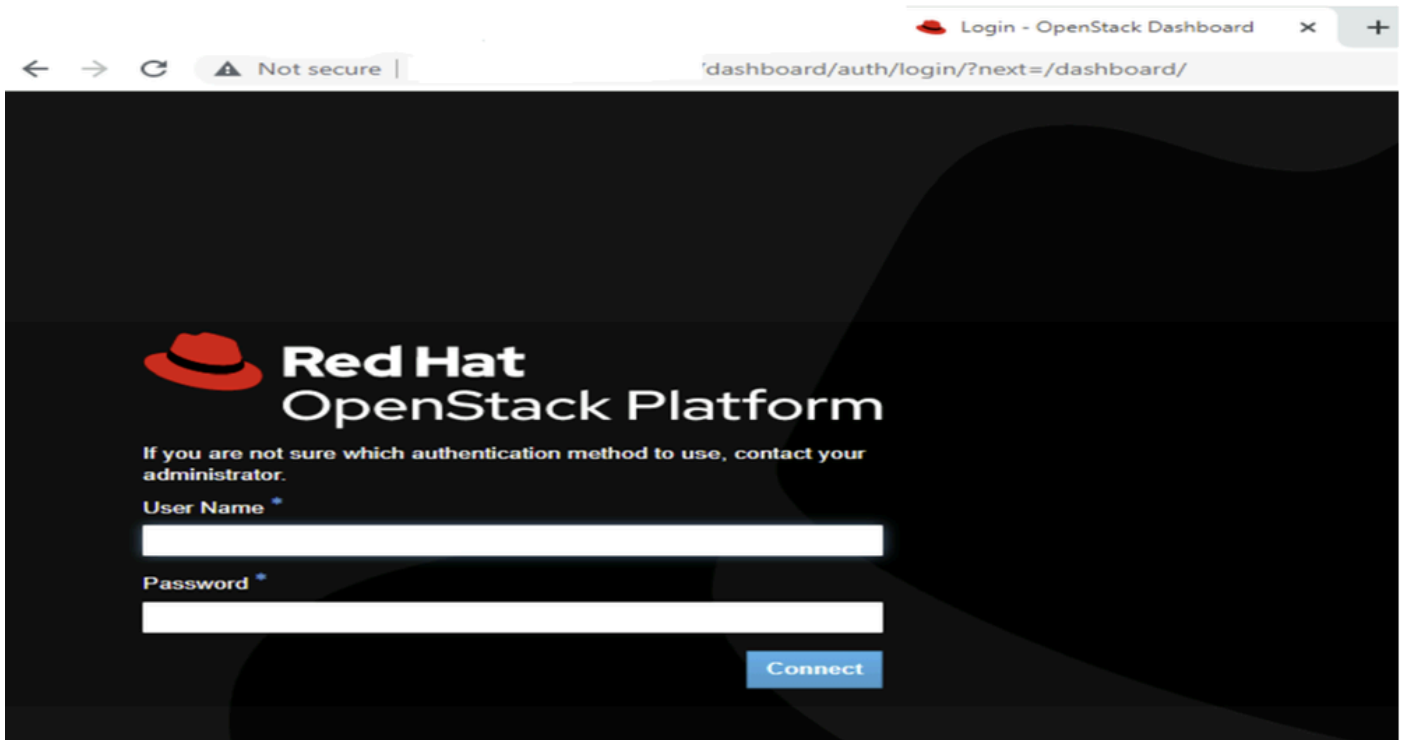
訪問水平儀表板

對於水平儀表板URL，請使用「overcloudc」中的「OS_AUTH URL」。

```
[stack@MUMBMUMBTCUDR201C0-ospd ~]$ cat MUMBMUMBTCUCL200C0rc
# Clear any old environment that may conflict.
for key in $( set | awk 'FS="=" {print $1}' ); do unset $key ; done
export NOVA_VERSION=1.1
export COMPUTE_API_VERSION=1.1
export OS_USERNAME=admin
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_NO_CACHE=True
export OS_CLOUDNAME=MUMBMUMBTCUCL200C0
export no_proxy='[,.]'
export PYTHONWARNINGS='ignore:Certificate has no, ignore:A true SSLContext object is not available'
export OS_AUTH_TYPE=password
export OS_PASSWORD=openstack
export OS_AUTH_URL='http://[redacted]:5000'
export OS_IDENTITY_API_VERSION=3
export OS_COMPUTE_API_VERSION=2.latest
export OS_IMAGE_API_VERSION=2
export OS_VOLUME_API_VERSION=3
export OS_REGION_NAME=regionOne

# Add OS_CLOUDNAME to PS1
if [ -z "${CLOUDPROMPT_ENABLED:-}" ]; then
  export PS1=${PS1:-""}
  export PS1=\${OS_CLOUDNAME:+"(\${OS_CLOUDNAME})"}\ $PS1
  export CLOUDPROMPT_ENABLED=1
fi
```

Horizon控制面板：



RHOSP群集的運行狀況檢查

<#root>

```
### Check OpenStack Services Status ###
```

```
# openstack compute service list  
# openstack network agent list  
# openstack volume service list  
# openstack orchestration service list  
# openstack identity service list  
# openstack endpoint list  
# openstack server list  
# openstack image list
```

摘要

RHOSP 16.2部署指南提供全面的分步說明，說明如何使用Red Hat提供的經驗證的工具和方法部署可擴展且生產就緒的OpenStack雲環境。本指南專為系統管理員和雲架構師量身定製，重點介紹如何使用基於TripleO(OpenStack on OpenStack)的OpenStack控制器部署RHOSP 16.2。

該指南涵蓋部署的所有關鍵階段，包括：

- 基礎設施規劃和前提條件
- 環境準備和網路配置
- Undercloud安裝和配置
- 超雲部署和部署後步驟
- 高可用性、儲存和服務擴展選項

對於尋求具有生態系統整合和Red Hat支援的可靠企業級雲平台的團隊而言，本指南至關重要。

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）。