

# Guía de implementación de RHOSP en UCS para cargas de trabajo de movilidad en modo desconectado

## Contenido

---

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Introducción a RHOSP](#)

[Arquitectura RHOSP](#)

[Subnube](#)

[Componentes bajo la nube](#)

[Nublar](#)

[Controlador](#)

[Informática](#)

[Almacenamiento](#)

[Repositorio local \(servidor de REPO\)](#)

[Redes RHOSP](#)

[Conectividad física RHOSP](#)

[Conectividad lógica RHOSP](#)

[Ajuste de parámetros de hardware](#)

[Instalación y creación de red KVM con hipervisor](#)

[REPO y creación de VM de Director](#)

[Prerequisites de REPO y creación de VM de Director](#)

[REPO VM Creation](#)

[Creación de VM de Director](#)

[Creación de REPO sin conexión local](#)

[Implementación de RHOSP Cloud](#)

[Prerequisites](#)

[Actualización de archivos de entrada](#)

[Implementación subnube](#)

[Implementación en la nube](#)

[Acceso al panel de Horizon](#)

[Comprobación del estado del clúster RHOSP](#)

[Summary](#)

---

## Introducción

Este documento describe un marco para implementar RHOSP en servidores C220 M6 UCS para admitir Cisco VPC-DI.

## Prerequisites

### Requirements

Cisco recomienda que usted tenga conocimiento de Red Hat OpenStack Platform (RHOSP) y posea fuertes habilidades en Red Hat Enterprise Linux (RHEL). Además, se requiere una comprensión sólida de los conceptos de virtualización y redes.

### Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

## Antecedentes

Esta guía describe la integración de RHOSP con la infraestructura de Unified Computing System (UCS), haciendo hincapié en la escalabilidad, la fiabilidad y la optimización del rendimiento.

Detalla las prácticas recomendadas y utiliza la automatización basada en scripts pasible para implementar OpenStack TripleO, que incluye la arquitectura Undercloud y Overcloud.

Con esta guía de implementación, las organizaciones pueden lograr una infraestructura de nube RHOSP sólida y eficiente adaptada para prestar servicio a las funciones de red virtual (VNF) de movilidad basadas en instancia distribuida (VPC-DI) de Cisco Virtual Packet Core.

## Introducción a RHOSP

RHOSP es una solución de nube privada de clase empresarial basada en el proyecto OpenStack de código abierto, integrada y compatible con Red Hat. Permite a las organizaciones implementar

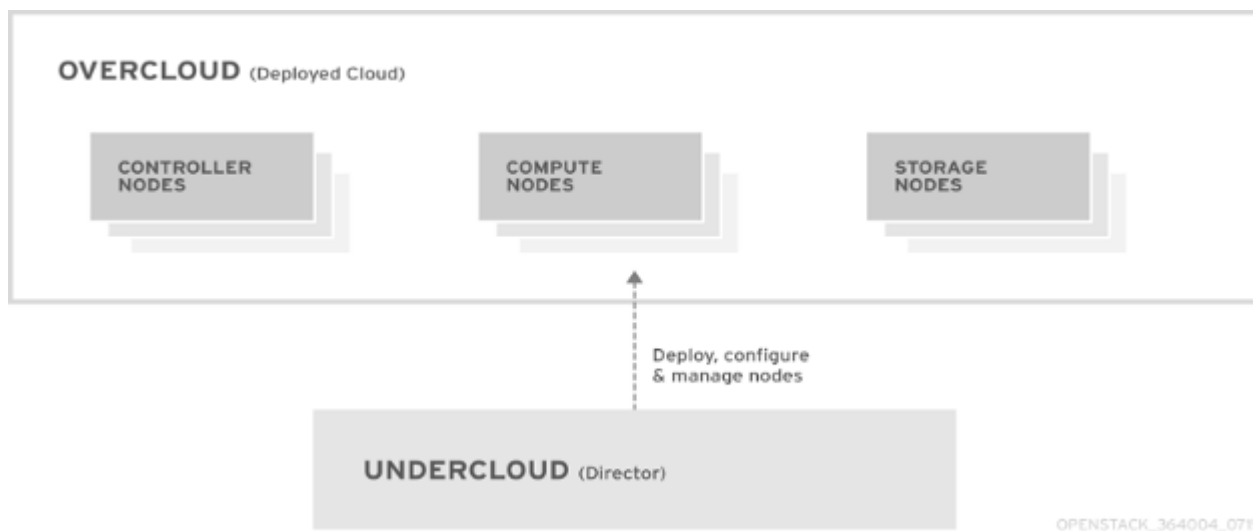
y gestionar la infraestructura como servicio (IaaS) para máquinas virtuales (VM), redes y almacenamiento a demanda.

Proporciona funciones como alta disponibilidad (HA), virtualización de funciones de red e implementaciones personalizables.

## Arquitectura RHOSP

El RHOSP se basa principalmente en el proyecto TripleO de OpenStack. Openstack utiliza Director, que actúa como un conjunto de herramientas para instalar y administrar un entorno RHOSP completo.

RHOSP está diseñado para proporcionar una infraestructura de nube escalable y flexible. Su arquitectura consta de dos componentes principales: Nube insuficiente y nube excesiva.



## Subnube

La nube inferior es el nodo de gestión principal que contiene el conjunto de herramientas del director RHOSP. Se trata de una instalación de OpenStack de un solo sistema que incluye componentes para el aprovisionamiento y la gestión de los nodos de OpenStack que forman el entorno de OpenStack (la nube excesiva).

## Componentes bajo la nube

La nube subyacente utiliza componentes de OpenStack como conjunto de herramientas básico. Cada componente funciona dentro de un contenedor separado en la nube inferior:

- OpenStack Identity (keystone): proporciona autenticación y autorización para los componentes de director
- OpenStack Bare Metal (irónico): gestiona nodos de metal desnudo
- OpenStack Networking (neutrón) y Open vSwitch: red de control para nodos sin software específico
- Orquestación de OpenStack (calor efímero): proporciona la orquestación de nodos después de que el director escriba la imagen de la nube en el disco

## Nublar

La nube superpuesta es el entorno RHOSP resultante creado con la nube subterránea. Esto incluye diferentes funciones de nodo que se definen en función del entorno OpenStack Platform (OSP) que el cliente pretende crear.

## Controlador

Los nodos del controlador proporcionan administración, redes y HA para el entorno OpenStack. Un entorno OpenStack recomendado contiene tres nodos Controller juntos en un clúster HA.

## Informática

Los nodos informáticos proporcionan recursos informáticos para el entorno OpenStack. Los nodos informáticos pueden ampliarse/reducirse en función de los requisitos de la red a lo largo del tiempo. Un nodo de cálculo predeterminado contiene estos componentes mencionados:

- Informática OpenStack (nova)
- VM basada en kernel (KVM)/emulador rápido (QEMU)
- vSwitch abierto

## Almacenamiento

Los nodos de almacenamiento proporcionan almacenamiento para el entorno OpenStack.



Nota: Existen varios enfoques para la implementación de Undercloud/Director y Offline Repository (REPO) en la red del cliente; se pueden implementar directamente en un nodo sin software específico o como VM encima del hipervisor KVM. En la guía de implementación actual, el servidor Director UCS aloja KVM (hipervisor) para implementar

---

---

varias VM en la parte superior. El nodo RHOSP Director y los nodos Offline-REPO se implementan como VM en el hipervisor KVM.

---

## Repositorio local (servidor de REPO)

Redhat proporciona una utilidad llamada `reposync` que se puede utilizar para descargar los paquetes desde la [Red de Entrega de Contenido](#) (CDN). Para descargar todos los paquetes de un canal específico, el sistema debe estar suscrito a ese canal. Si el sistema no está suscrito al canal requerido, `reposync` no puede descargar y sincronizar esos paquetes en el sistema local.

Los repositorios se configuran en `/etc/yum.repos.d/` path por archivos que terminan con la extensión `.repo`. Se pueden definir varios repositorios en el mismo archivo.

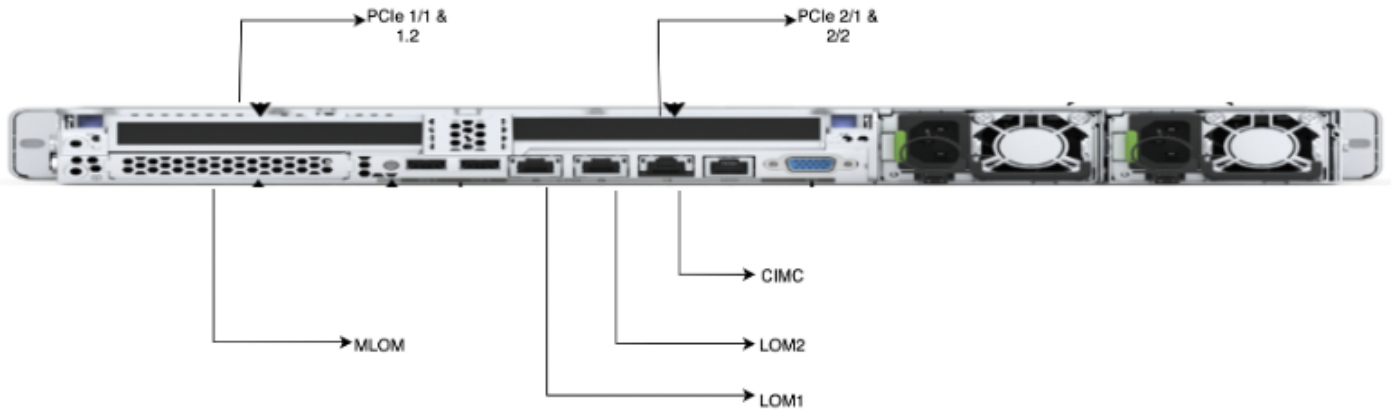
## Redes RHOSP

El servicio de red (neutrón) es el componente de red definida por software (SDN) de RHOSP. El servicio RHOSP Networking gestiona el tráfico interno y externo hacia y desde las instancias de VM y proporciona servicios centrales como routing, segmentación, DHCP y metadatos. Proporciona la API para las capacidades de red virtual y la gestión de switches, routers, puertos y firewalls.

RHOSP Director asigna los servicios de OpenStack a diferentes redes aisladas. Las redes que transportan cada tipo de tráfico son: Cisco Integrated Management Controller (CIMC), aprovisionamiento, API interna, datos de almacenamiento, gestión de almacenamiento, arrendatario y externa (Secure Shell [SSH] y operaciones, administración y mantenimiento [OAM]).

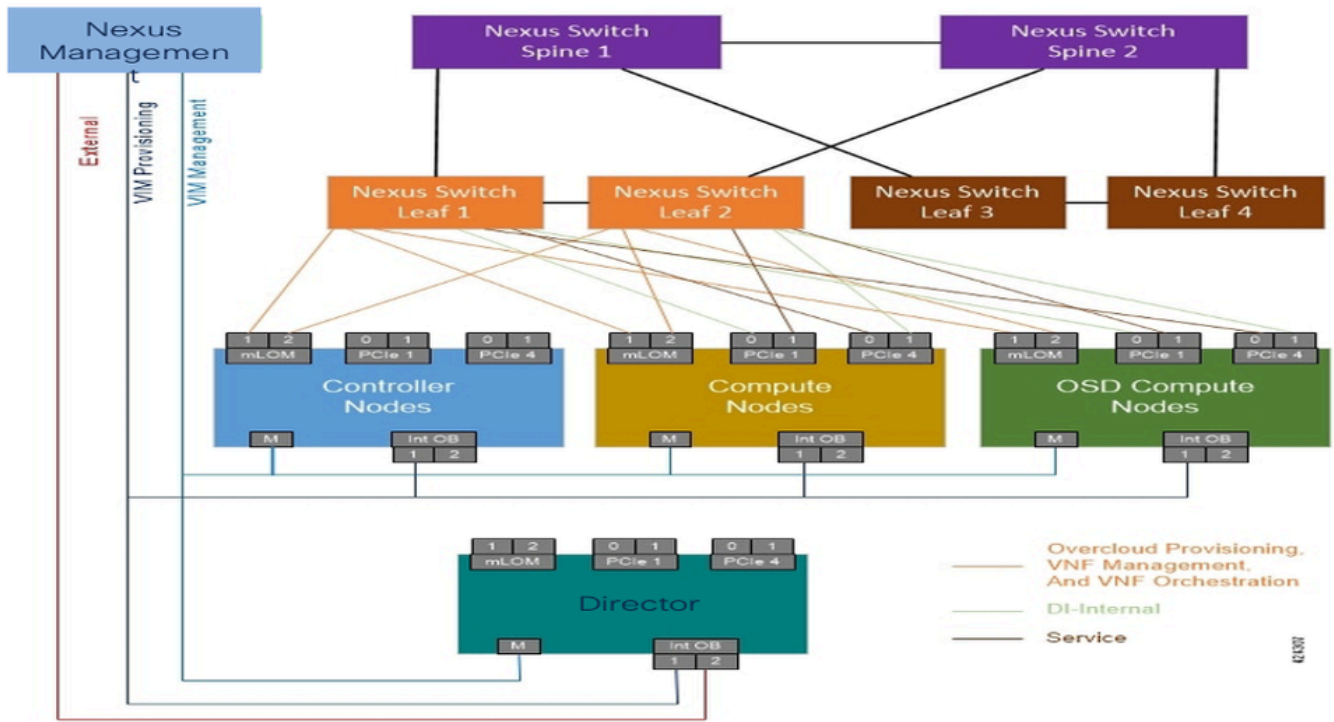
## Conectividad física RHOSP

La implementación de RHOSP utiliza diferentes puertos físicos de los servidores Cisco UCS C220 M6 para diferentes fines de conectividad.



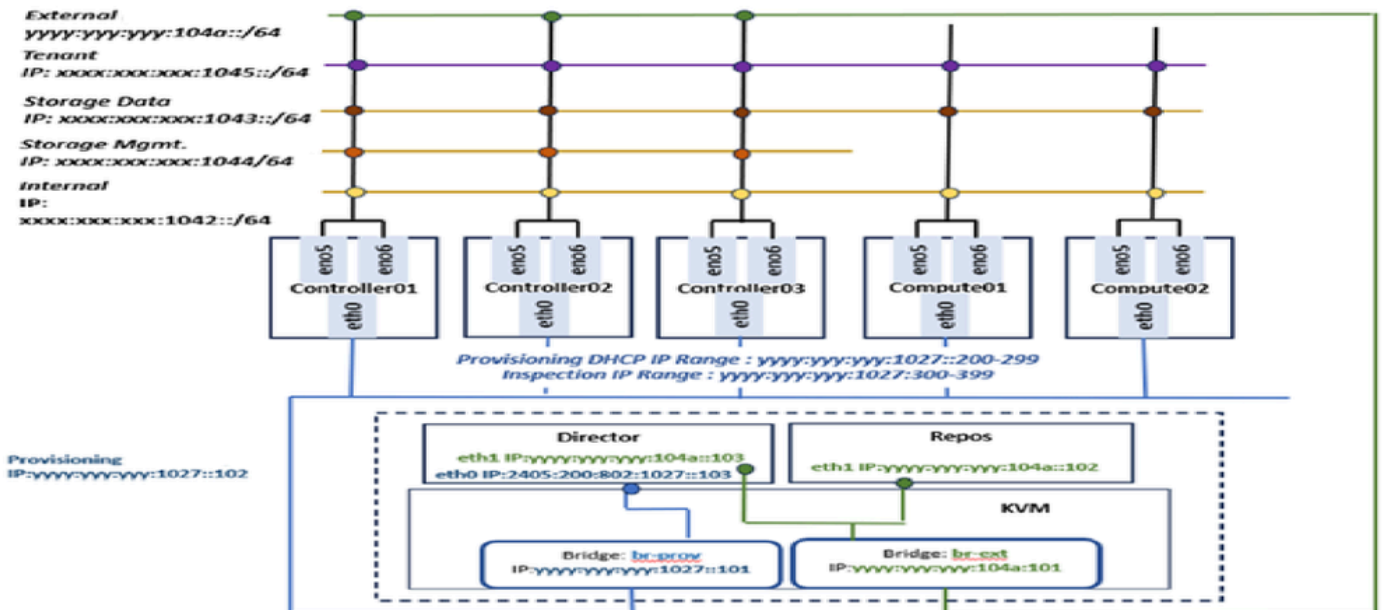
Serial Number	Puertos físicos	Detalles
1.	CIMC	CIMC proporciona conectividad fuera de banda para la administración y el aprovisionamiento de servidores.
2.	Virtualización de E/S de raíz única (SR-IOV)/interconexión de componentes periféricos Express (PCIe)	La tarjeta de interfaz de red (NIC) PCIe se utiliza en los nodos informáticos para las redes internas de ID y de servicio para la VNF.
3.	Lan modular en placa base (MLOM)	Los puertos MLOM se configuran como enlaces. osp_external, osp_internal, osp_tenant, osp_external, osp_storage_data y osp_storage_mgmt utiliza el puerto MLOM para la comunicación interna.
4.	Lan en placa base (LOM)	El director utiliza los puertos LOM1 y LOM2, mientras que los equipos y los controladores solo utilizan el puerto LOM1.  LOM1 se utiliza para implementar o aprovisionar Openstack en todos los servidores.  LOM2 se utiliza como OAM (red externa) en el director.

El diagrama muestra la conectividad física con los servidores.



### Conectividad lógica RHOSP

La red RHOSP cuenta con varias subredes que se encargan de diferentes servicios dentro de la nube.



- OSP\_CIMC:

CIMC es la Intelligent Programming Management Interface (IPMI) que controla la gestión de todos

los servidores UCS. Esta red CIMC se configura en el puerto CIMC independiente de todos los servidores UCS.

- OSP\_Provisioning:

Esta red es responsable del aprovisionamiento y de la gestión del arranque del entorno de ejecución previa al arranque (PXE) de los ordenadores y los servidores controladores durante la implementación de la nube excesiva, así como de obtener la IP de DHCP. La red de aprovisionamiento se configura como VLAN nativa en el puerto LOM1 de todos los servidores UCS para ofrecer simplicidad y compatibilidad. Esta red de aprovisionamiento es responsable de la implementación de la nube en todos los servidores.

Debido a la virtualización en el servidor Director, era necesario crear una red puente en el KVM para que la VM Director se comunicara con los otros servidores.

- OSP\_Internal:

La red API interna se utiliza para la comunicación entre los servicios de OpenStack como neutrones, nova, keystone, etc.

La red OSP\_Internal se configura en los puertos MLOM asociados en los nodos Controlador e Informática.

- OSP\_Arrendatario:

La red de arrendatarios se crea de forma predeterminada en los proyectos de nube para la gestión de VNF. En la configuración actual, sólo se crea un proyecto OpenStack para la implementación de VNF.

La red OSP\_Tenant se configura en los puertos MLOM asociados en los nodos de controlador e informática.

- OSP\_Externo:

La red externa se utiliza para todas las redes de acceso externo (como SSH) y API.

La red OSP\_External se configura en el puerto LOM2 del nodo Director y en los puertos MLOM asociados de los nodos Controller y Compute.

- OSP\_Storage\_Data:

La red OSP\_Storage se utiliza para todas las operaciones relacionadas con el acceso al almacenamiento. Esto es necesario para la comunicación entre el servicio CEPH y VNF que necesitan acceder al almacenamiento. Es utilizado por los nodos Controller, Compute y CEPH.

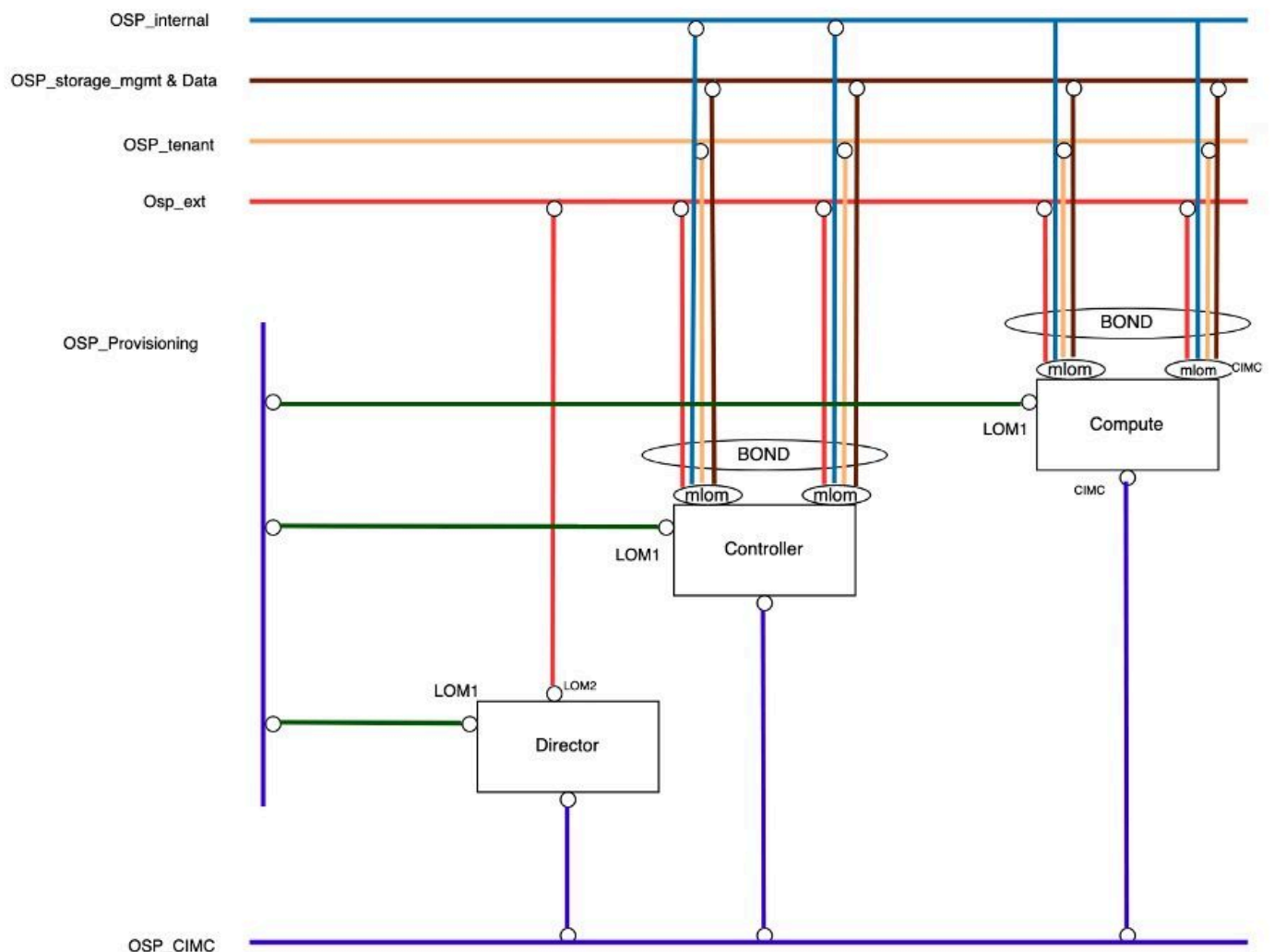
La red OSP\_Storage\_Data se configura en los puertos MLOM asociados en los nodos de controlador e informática.

- OSP\_Storage\_Management:

OpenStack Object Storage utiliza esta red para sincronizar objetos de datos entre los nodos de réplica participantes en un clúster de almacenamiento, formado entre nodos controller-Compute.

La red OSP\_Storage\_Mgmt se configura en los puertos MLOM asociados de los nodos de controlador y cálculo.

El diagrama muestra cómo las redes lógicas de la subnube están conectadas a cada tipo de nodos en el clúster RHOSP.



## Ajuste de parámetros de hardware

- Asegúrese de que la IP esté habilitada y configurada con la dirección de administración correcta según el diseño.
- Asegúrese de que el usuario administrador y su contraseña sean iguales para todos los accesos de inicio de sesión de CIMC.
- Asegúrese de que el modo Bootstrap (Boot) está establecido en Unified Extensible Firmware Interface (UEFI).
- Asegúrese de establecer la configuración del BIOS con los parámetros recomendados.
- Establezca el orden de arranque en:
  1. LOM-PXE (eno1)
  2. Unidad de disco duro de arranque (HDD)
- Establezca la dirección MAC de LOM1 (eno1).
- Asegúrese de que todos los servidores UCS/CIMC están actualizados con la última versión de firmware recomendada.

## Instalación y creación de red KVM con hipervisor

Existen varios métodos para implementar Undercloud/Director y Offline REPO en la red del cliente. Se pueden implementar directamente en un nodo sin software específico o como VM ejecutadas en un hipervisor KVM.

En la guía de implementación actual, el servidor Director UCS está configurado para alojar el hipervisor KVM, lo que facilita la creación de varias VM. El nodo RHOSP Director y el nodo Offline REPO se implementan como VM en este hipervisor KVM.



Nota: Se deben observar los pasos de instalación KVM RHEL estándar para implementar el hipervisor KVM.

---

- Una vez que el KVM esté activo, actualice el nombre de host:  
`hostnamectl set-hostname <hostname> —static`
- Configure los puentes externos y de aprovisionamiento y enlace las interfaces.

br-prov: eth0

br-ext: eth1

Estos puentes deben crearse mediante la GUI de la interfaz de usuario de texto del administrador de red (NMTUI).

- De forma predeterminada, los puertos eno1 y eno2 se crean en KVM para los puertos LOM1 y LOM2 físicos, y verifican los mismos de forma cruzada mediante la dirección MAC del registro CIMC.
- Cree una red puente y agregue puertos esclavos en ella asignando la dirección MAC correcta.
- Creación de red posterior al puente, asegúrese de que el aprovisionamiento y la puerta de enlace externa sean accesibles desde KVM.

## REPO y creación de VM de Director

### Prerrequisitos de REPO y creación de VM de Director

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

- Instale los paquetes necesarios en KVM para REPO y Director VM Creation.

```
! command not found...
[root@RH08KVM01 ~]#
[root@RH08KVM01 ~]# 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 |
Package qemu-kvm-15:4.2.0-48.module+el8.4.0+10260+630e003b.x86_64 is already installed.
Package libvirt-9.0.0-35.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
```

- Cree los directorios necesarios en KVM.

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

- Copie los archivos en los directorios.

```
# 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/
```

- Monte el ISO en /mnt/iso.

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

- Cree el archivo REPO en la ruta /etc/yum.repos.d.

```
# 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/
```

- Verifique el repolist y asegúrese de que los REPOs de Appstream y de Base estén asignados.

```
# dnf repolist
```

```
[root@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]# 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@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]#
```

## REPO VM Creation

```
$ 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@123
```

```
[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]#
```

- Cree la imagen del servidor de REPO.

```
$ 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/images/rhel_84_osprepo.qcow2
```

```
[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  
[ 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 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  
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  
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.service
```

```
[root@MUMBKVMC01 images]#  
[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.service  
[root@MUMBKVMC01 images]#  
[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/1836/0100 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/OSP
```

```
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# virt-install --cpu host --memory 32768 --vcpus 16 --os-variant rhel8.4 --disk path=/data/images/OSP_REPO_RHEL_84.qcow2,device=disk,bus=virtio,format=qcow2  
--import --noautoconsole --vnc --network bridge:br-ext --name OSP_REPO_RHEL_84  
  
** (process:49296): WARNING **: 17:13:15.813: Entity http://pcisig.com/pci/1836/0100 referenced but not defined  
  
Starting install...  
Domain creation completed.  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]#
```

```
$ virsh list --all
```



```
# 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
```

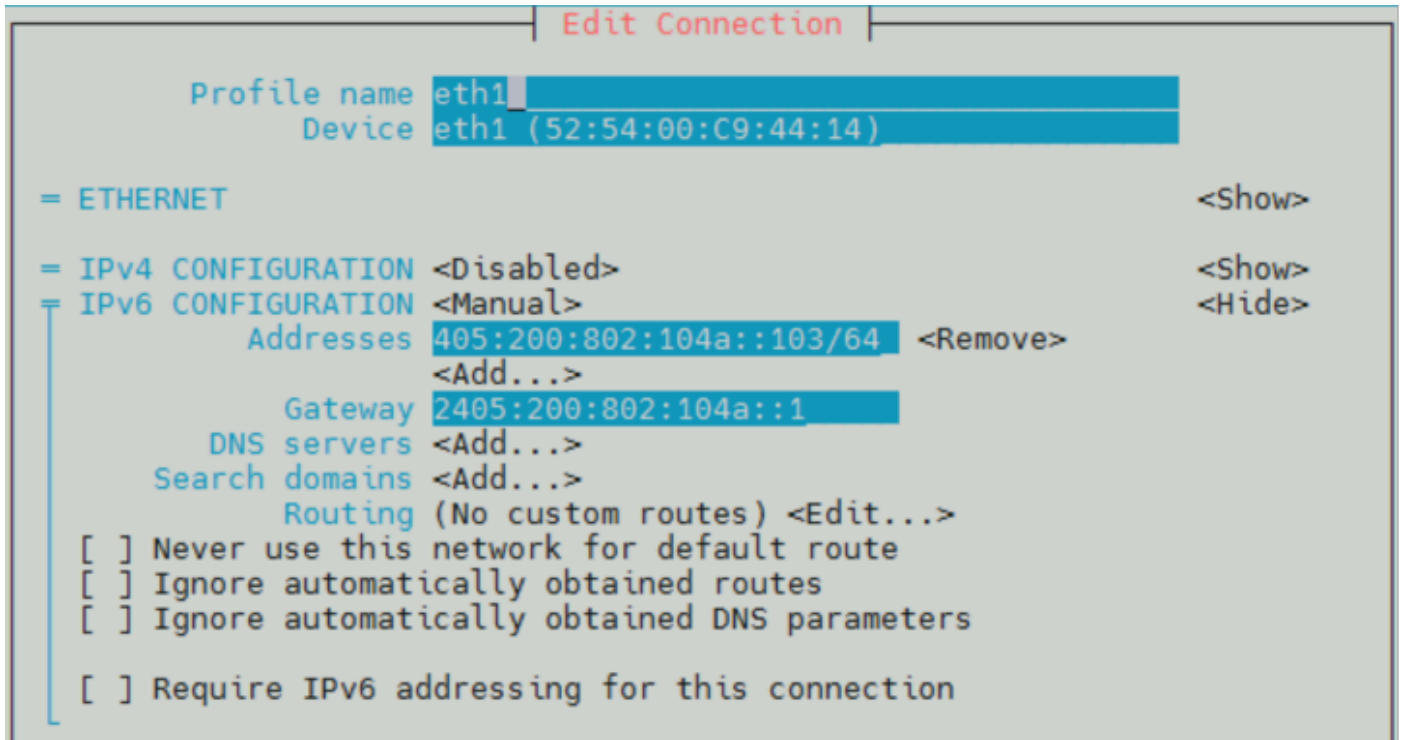
```
Domain creation completed.  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# virsh list --all  
Id Name State  
-----  
1 OSPREPO_RHEL_84 running  
2 OSPDIRECTOR_RHEL_84 running  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]#
```

```
[root@MAGKVMC02 images]#  
[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 **: 15:13:00.438: value "64" of type "gint" is invalid or out of range for property 'desktop-width' of type 'gint'  
(virt-viewer:193156): GLib-GObject-WARNING **: 15:13:00.438: value "64" of type "gint" is invalid or out of range for property 'desktop-height' of type 'gint'
```

- Abra una ventana nueva y inicie sesión en la consola de la máquina virtual.

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

- Implementación de VM de Post Director, se crean dos interfaces eth0 y eth1.
  - eth0 se utiliza para la red OSP\_provisioning, en la que durante la red br-ctlplane se crean durante la implementación de Undercloud.
  - eth1 se utiliza para la conectividad externa para el acceso SSH. Por lo tanto, se asigna la dirección IP de subred externa OSP.



## Creación de REPO sin conexión local

El servidor REPO debe estar registrado con Redhat CDN y debe tener el repositorio de todos los paquetes disponibles de RHOSP 16.2 necesarios para la implementación. Los paquetes RHEL RPM y las imágenes del contenedor RHOSP se deben descargar en la VM de REPO mediante proxy.

## Implementación de RHOSP Cloud

RHOSP 16.2 se implementa en la red del cliente mediante la automatización. Los scripts de Ansible se utilizan para automatizar la implementación en la nube y en la nube.

### Prerequisites

Pasos que deben seguirse antes de iniciar la implementación real de la nube:

1. Garantizar la conectividad del nodo KVM con la máquina virtual de REPO local y la máquina virtual de Director en la red OSP\_EXT.
2. Asegúrese de que todos los scripts ansible y scripts de automatización cargados en el host KVM se encuentran en la carpeta designada.
3. Cree carpetas con nombres como 'cisco' y 'automation' y coloque el tarball de scripts reutilizables.

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

El archivo .tarball constará de tres estructuras de directorio de carpetas, denominadas como:

- scripts: consta de scripts de shell utilizados para la configuración de nodos sin software específico.
- rpms: consta de controladores Intel Ethernet Connection (ICE), controladores Intel y paquetes Red-Hat Package Manager (RPM).
- ansible: consta de un archivo vars de entrada, archivos yaml ansible para implementación y plantillas jinja.

4. Instale el paquete shpass. sshpass es una utilidad de línea de comandos que se utiliza para proporcionar contraseñas a ssh de manera no interactiva. Se utiliza principalmente en escenarios de scripting o automatización en los que no es posible introducir manualmente la contraseña.

- Descargue el paquete sshpass sshpass.tar.gz desde Internet/servidor existente.
- Instale el paquete de la Colección de compiladores de GNU (GCC).

```
# yum install gcc
```

- Instale make packages.

```
# yum install make
```

- Descomprima el paquete shpass e instale el pase SSH.

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

5. El proceso de instalación de director requiere que un usuario no raíz ejecute comandos. El usuario 'Stack' debe crearse en la VM de Director con acceso sudo.

```
# 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. Copie el archivo rootCA.crt del servidor REPO a la VM y KVM de Director en la ruta dada. Además, actualice el certificado de VM de REPO en la lista de confianza.

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

7. Actualice los detalles del nombre de host del servidor REPO local en Director VM y KVM en el archivo /etc/hosts.

8. En KVM y Director VM, instale paquetes adicionales como python, ansible, etc. para ejecutar scripts de automatización ansible.

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

9. La subred CIMC debe ser accesible desde la red de aprovisionamiento del director para permitir el aprovisionamiento durante la implementación de la nube. Si es necesario, agregue una ruta estática para la misma.

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

10. En KVM y Director VM, cree un archivo host en la carpeta /ansible y agregue detalles específicos de la pila según sea necesario.

```
[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. Asegúrese de que todos los cuadernos reproducibles y los archivos de entrada se deben mantener en la carpeta /home/stack de Director VM.

## Actualización de archivos de entrada

Hay un archivo de variable de entrada que consta de los detalles específicos de la red del cliente que se deben preparar para la implementación en la nube.

Ruta: /home/cisco/automation/ansible/podvars

Nombre de Archivo: <stack-name>\_vars.yml

Actualice los parámetros resaltados según el plan IP específico del sitio/documento de diseño de bajo nivel.



Nota: Las direcciones IP ficticias se utilizan únicamente con fines de representación.

---

<#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::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 version
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
# #####

```

## Implementación subnube

La subnube se implementa mediante scripts reutilizables en siete pasos. Todos los pasos deben ejecutarse desde el host KVM que actúa como host de salto en este caso.

Paso	Etiqueta	Descripción	YAMLs de campaña
------	----------	-------------	------------------

Paso 1.	checkfiles	Verifique los cuadernos, scripts y RPM necesarios en Openstack Platform Director (OSPD).	osp16_publish_playbooks_verify.yml
Paso 2.	genpodvar	Generar archivos variables específicos de POD relacionados con hardware, RHEL, etc., como common_vars.yml (hardware, software, detalles de red), hw_m6_vars.yaml (CPU, memoria, páginas enormes, disco, NIC, etc.), rhel_84_vars.yml (RHEL, kernel, controlador ICE, versión NIC), pf_esc_vars.yml (detalles del controlador de servicios elásticos (ESC)), osp_16_vars.yml (versión de OSP, zona horaria, tipo de IP, ID de VLAN, IP, detalles de neutrones).	osp16_generate_pod_specific_vars.yml
Paso 3.	predesplegar	Configure el nombre de dominio completo (FQDN), el protocolo de tiempo de la red (NTP) y actualice todos los paquetes en el nodo de director.	osp16_pre_undercloud_deploy.yml
Paso 4.	arranque inicial	Realiza el primer reinicio del nodo de director de Undercloud después de la instalación del paquete y la configuración anteriores.	osp16_undercloud_deploy.yml
Paso 5.	ucdeploy	Instalar la pila de Undercloud en el director	osp16_undercloud_tuning.yml
Paso 6.	cstate	Configure los parámetros de estado C de la CPU del BIOS en el nodo de director.	osp16_cstate.yml
Paso 7.	arranque secundario	Segundo reinicio en Undercloud director después de los cambios en el BIOS.	N/A

El archivo con el nombre osp16\_auto\_undercloud\_deploy.yml es el principal cuaderno de campaña que se puede ejecutar en una sola iteración, pero se recomienda ejecutar el cuaderno de campaña paso a paso utilizando diferentes etiquetas para facilitar la resolución de problemas en caso de problemas de implementación.

```
<#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
```

## Implementación en la nube

La nube excesiva se implementa con un mínimo de tres controladores en modo HA y un ordenador. La nube excesiva se implementa mediante scripts reutilizables en 17 pasos. Todos los pasos se deben ejecutar desde Director-VM que actúa como Jump-host en este caso.

Paso	Etiqueta	Descripción	YAMLs de campaña
Paso 1.	genpodvar	Generar archivos variables específicos de POD para Overcloud relacionados con hardware, RHEL, etc., como common_vars.yml (hardware, software, detalles de red), hw_m6_vars.yml (CPU, memoria, páginas enormes, disco, NIC, etc.), rhel_84_vars.yml (RHEL, kernel, controlador ICE, versión NIC), pf_esc_vars.yml (detalles ESC), osp_16_vars.yml (versión de OSP, zona horaria, tipo de IP, ID de VLAN, IP, detalles de neutrones).	osp16_generate_pod_specific_vars.yml
Paso 2.	geninstack	Genere el archivo Instackenv JSON desde /var/common_vars.yml creado en el paso anterior.  El director requiere una plantilla de definición de nodo, que se crea manualmente. Este archivo instackenv.json utiliza el formato JSON y contiene todos los detalles de administración de energía y hardware para los nodos. Este paso también valida la configuración de hardware en el servidor UCS antes de generar el archivo.	osp16_generate_instackenv.yml
Paso 3.	cimcvd	Configure la configuración de CIMC y los discos virtuales (VD) en cada servidor haciendo referencia a common_vars.yml, hw_m6_vars.yml, y	osp16_cimc_vd_configure.yml

		rhel_84_vars.yml.	
Paso 4.	predespliegue	<p>Este paso realiza todos los requisitos previos para implementar la nube excesiva.</p> <p>Establece el FQDN, NTP, y actualiza todos los paquetes, empuja la imagen a la ruta para la implementación.</p>	osp16_pre_overcloud_deploy.yml
Paso 5	importnodes	<p>En este paso, se realiza una inspección interna de la CPU del servidor, la memoria, la NIC junto con la interfaz y los puertos en los switches de red. La introspección se realiza en los switches de red conectados para todos los controladores y equipos.</p>	osp16_import_ironic_nodes.yml
Paso 6.	gentemplates	<p>Generar archivos de plantilla personalizados para controladores y equipos. En la plantilla personalizada, defina las funciones de controlador e informática para todos los servicios que se ejecuten en ellas. También realiza la consolidación del sistema mediante la aplicación de certificados, rutas, etc.</p>	osp16_generate_custom_templates.yml
Paso 7.	ocdeploy	<p>En este paso, se realiza una implementación de OpenStack Overcloud. Ejecute deploy.sh proporcionado por Red Hat para la implementación de RHOSP.</p>	osp16_overcloud_deploy.yml
Paso 8	geninventariar	<p>En este paso, se genera un archivo yml de inventario para que lo utilice ansible, donde se almacenan IP de aprovisionamiento, IP IPMI (CIMC) y credenciales, y se asignan con el controlador y los</p>	osp16_build_Inventory_v3.py

		equipos para que la automatización inicie sesión en el sistema y ejecute los pasos siguientes.	
Paso 9.	offlinerepo	Configure Overcloud Offline REPO en el archivo /etc/yum.repo.d/offline.repo señale al servidor de repo a través de la red externa.	osp16_config_offline_repo.yml
Paso 10.	esgrima	Configure la valla en todos los nodos del controlador con Disparar al otro nodo en la cabeza (una técnica de valla en clústeres HA) (STONITH).	osp16_config_fencing.yml
Paso 11.	raidcache	Configure los ajustes de caché RAID para todos los controladores y equipos y también configure los ajustes de almacenamiento de SWIFT.	osp16_raid_cache_tuning.yml
Paso 12.	dnfupdate	Ejecute las actualizaciones de DNF para todos los paquetes en todos los nodos.	dnf_update_all_packages.yml
Paso 13.	setiplink	En este paso, se habilita el control de modo de confianza para los puertos SR-IOV para el tráfico interno y de datos de Evolved Packet Data Gateway (EPDG). La compatibilidad con los puertos SR-IOV está disponible en neutrones y permite a las VM acceder a la red mediante funciones virtuales SR-IOV.	osp16_setIpLink.yml
Paso 14.	perro guardián	En este paso, la configuración de IPMI en el nodo de director se configura para la tarea de administración en todos los	osp16_config_ipmi_watchdog.yml

		servidores a través de una conexión fuera de banda.	
Paso 15.	hielero	Actualice el controlador Intel E810 ICE para tarjetas de interconexión de componentes periféricos (PCI) a la versión 1.12.6 para EPDG para utilizar los puertos NIC de Intel como SR-IOV.	osp16_ice_driver_install.yml
Paso 16.	reiniciar	Reinicie todos los nodos Overcloud después de la ejecución de los pasos anteriores.	osp16_reboot_overcloud_hosts.yml
Paso 17.	verifyrhosp	Verifique la configuración y el estado de la implementación de RHOSP.	osp16_rhosp_verify.yml

Para el aprovisionamiento de nodos Overcloud, Undercloud utiliza 'overcloud-hardened-uefi-full.qcow2'. Por lo tanto, antes de comenzar la implementación de la nube excesiva, la imagen debe almacenarse en una ruta designada en la nube insuficiente/director.

Copie el archivo Overcloud qcow2 del sitio remoto.

```
<#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
```

Para supervisar los registros de implementación, utilice el archivo de registro más reciente.

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

Asegúrese de que han pasado los 17 pasos.

Fallo en las comprobaciones => El recuento debe ser 00.

Logs => /home/stack/autologs/osp16\_rhosp\_verify.yml \_20200703T042257.log

#=====

# PASO	ETIQUETA	DESCRIPCIÓN	CUADERNO
--------	----------	-------------	----------

#=====

# step1	genpodvars	Generar archivos de variables específicas de POD	
osp16_generate_pod_specific_vars.yml -e podname=			

# paso2	geninstack	Generar archivo JSON de Instackenv	
osp16_generate_instackenv.yml -e podname=			

# paso3	cimc vd	Configurar VD de CIMC	
osp16_cimc_vd_configure.yml			

# paso4	preocdeploy	Configurar implementación previa de nube excesiva	
osp16_pre_overcloud_deploy.yml			

# paso5	importnodes	Importar Openstack Baremetal Ironic Nodes	
osp16_import_ironic_nodes.yml			

# paso6	gentemplates	Generar plantillas personalizadas	
osp16_generate_custom_templates.yml			

# step7	ocdeploy	Openstack Overcloud Implementar	
osp16_overcloud_deploy.yml			

# paso8	genInventory	Generar archivo de inventario	
osp16_build_Inventory_v3.py --ipmipass			

# paso9	offlinerepo	Configurar Overcloud Offline Repo desde archivo TAR sin conexión	
osp16_config_offline_repo.yml			

# paso10	cercado	Post Deploy MOP Before Reboot - Configurar delimitación	
osp16_config_fencing.yml			

# paso11	raidcache	Post Deploy MOP Before Reboot - Raid Cache and PR Tuning	
----------	-----------	--	--

osp16\_raid\_cache\_tuning.yml

# paso12 | dnfupdate | Post Deploy MOP Before Reboot - Paquetes de actualización de Dnf  
| dnf\_update\_all\_packages.yml

# paso13 | setiplink | Post Deploy MOP Before Reboot - Set VF IP Link Trust ON |  
osp16\_setIpLink.yml

# paso14 | perro guardián | Post Deploy MOP Before Reboot - Config IPMI Watchdog |  
osp16\_config\_ipmi\_watchdog.yml

# paso15 | icedriver | Post Deploy MOP Before Reboot - Actualización del controlador E810  
ICE | osp16\_ice\_driver\_install.yml

# paso16 | reboot | Reiniciar todos los nodos en la nube |  
osp16\_reboot\_overcloud\_hosts.yml

# paso17 | verifyrhosp | Verificar configuración y estado de implementación de RHOSP |  
osp16\_rhosp\_verify.yml -e podname=

#=====

TODOS LOS HOSTS ACCESIBLES

=====

COMPROBACIONES REALIZADAS => 17

COMPROBACIONES PASADAS => 17

COMPROBACIONES FALLIDAS => 0

=====

ESTADO GENERAL => SUPERADO !!

=====

Tras la correcta implementación de la nube excesiva, asegúrese de que el panel de Horizon esté accesible.

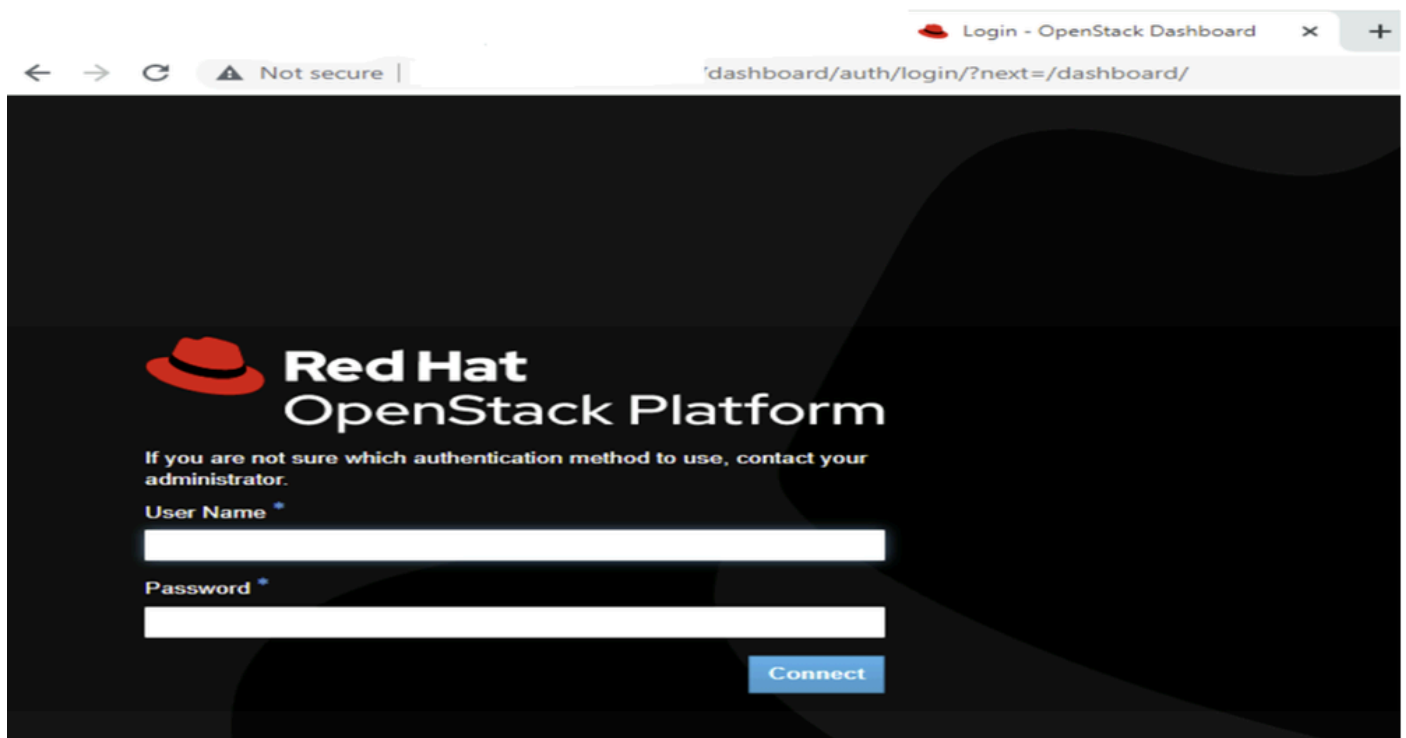
## Acceso al panel de Horizon

Para la URL de panel de horizonte, use 'OS\_AUTH\_URL'from 'overcloudrc'.

```
[stack@MUMBMUMBTCUDR201C0-ospd ~]$ cat MUMBMUMBTCUCL200C0rc
# Clear any old environment that may conflict.
for key in $( set | awk '{FS="="} /^OS_/ {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='[REDACTED],[REDACTED]',[REDACTED]
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
```

Panel de Horizon:



# Comprobación del estado del clúster RHOSP

<#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
```

## Summary

La guía de implementación de RHOSP 16.2 proporciona instrucciones detalladas y completas para implementar un entorno de nube OpenStack escalable y preparado para la producción usando las herramientas y metodologías probadas de Red Hat. Esta guía está adaptada para administradores de sistemas y arquitectos de nube y se centra en la implementación de RHOSP 16.2 mediante el director de OpenStack, que se basa en TripleO (OpenStack en OpenStack).

La guía abarca todas las fases críticas de la implementación, incluidas las siguientes:

- Requisitos previos y planificación de infraestructura
- Preparación del entorno y configuración de la red
- Instalación y configuración bajo la nube
- Pasos posteriores a la implementación y la implementación en la nube
- Opciones de ampliación de servicios, almacenamiento y alta disponibilidad

Esta guía es esencial para los equipos que buscan una plataforma de nube fiable y de clase empresarial con la integración del ecosistema y la compatibilidad de Red Hat.

## 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).