

Guida alla distribuzione di RHOSP su UCS for Mobility Workload in modalità disconnessa

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Informazioni su RHOSP](#)

[Architettura RHOSP](#)

[Undercloud](#)

[Componenti undercloud](#)

[Overcloud](#)

[Controller](#)

[Calcola](#)

[Storage](#)

[Repository locale \(server REPO\)](#)

[RHOSP Networking](#)

[Connettività fisica RHOSP](#)

[Connettività logica RHOSP](#)

[Regolazione parametri hardware](#)

[Installazione e creazione della rete KVM dell'ipervisor](#)

[Creazione di VM di REPO e Director](#)

[Prerequisiti per la creazione di VM di REPO e Director](#)

[Creazione VM REPO](#)

[Creazione VM Director](#)

[Creazione locale di REPO offline](#)

[Implementazione RHOSP Cloud](#)

[Prerequisiti](#)

[Aggiornamento dei file di input](#)

[Distribuzione undercloud](#)

[Distribuzione cloud](#)

[Accesso al dashboard orizzonte](#)

[Verifica dello stato del cluster RHOSP](#)

[Riepilogo](#)

Introduzione

Questo documento descrive un framework per l'implementazione di RHOSP su server UCS C220 M6 per il supporto di VPC-DI Cisco.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza di Red Hat OpenStack Platform (RHOSP) e la conoscenza di Red Hat Enterprise Linux (RHEL). Inoltre, è necessaria una solida conoscenza dei concetti di virtualizzazione e rete.

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Questa guida descrive l'integrazione di RHOSP con l'infrastruttura UCS (Unified Computing System), sottolineando la scalabilità, l'affidabilità e l'ottimizzazione delle prestazioni.

Descrive le best practice e utilizza un'automazione basata su script flessibile per l'installazione di OpenStack TripleO, che comprende l'architettura Undercloud e Overcloud.

Utilizzando questa guida all'installazione, le organizzazioni possono ottenere un'infrastruttura cloud ROSP robusta ed efficiente personalizzata per le funzioni VNF (Virtual Packet Core - Distributed Instance) basate su Cisco Virtual Packet Core (VPC-DI).

Informazioni su RHOSP

RHOSP è una soluzione di cloud privato di livello enterprise basata sul progetto open-source OpenStack, integrata e supportata da Red Hat. Consente alle organizzazioni di installare e gestire l'infrastruttura IaaS (Infrastructure-as-a-Service) per macchine virtuali (VM), reti e storage su

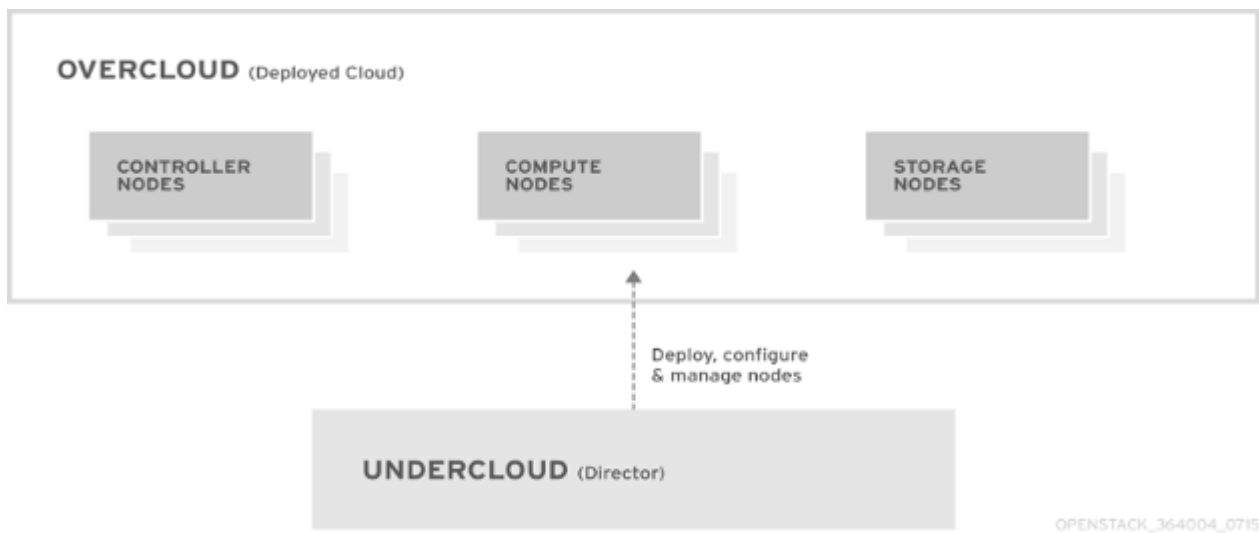
richiesta.

Offre funzionalità quali alta disponibilità (HA, High Availability), virtualizzazione delle funzioni di rete e installazioni personalizzabili.

Architettura RHOSP

Il RHOSP è basato principalmente sul progetto TripleO OpenStack. Openstack utilizza Director come set di strumenti per l'installazione e la gestione di un ambiente RHOSP completo.

RHOSP è progettato per fornire un'infrastruttura cloud scalabile e flessibile. La sua architettura è costituita da due componenti principali: Undercloud e Overcloud.



Undercloud

L'undercloud è il nodo di gestione principale che contiene il set di strumenti del director RHOSP. Si tratta di un'installazione OpenStack a sistema singolo che include componenti per il provisioning e la gestione dei nodi OpenStack che formano l'ambiente OpenStack (overcloud).

Componenti undercloud

L'undercloud utilizza i componenti OpenStack come set di strumenti di base. Ciascun componente opera all'interno di un contenitore separato sul subcloud:

- OpenStack Identity (keystone) - Fornisce autenticazione e autorizzazione per i componenti director

- OpenStack Bare Metal (ironico) - Gestisce i nodi bare metal
- OpenStack Networking (neutron) e Open vSwitch - Controllo della rete per nodi bare metal
- OpenStack Orchestration (Ephemeral Heat) - Fornisce l'orchestrazione dei nodi dopo che il director ha scritto l'immagine di overcloud su disco

Overcloud

L'overcloud è l'ambiente RHOSP risultante creato utilizzando l'undercloud. Sono inclusi diversi ruoli dei nodi definiti in base all'ambiente OpenStack Platform (OSP) che il cliente intende creare.

Controller

I nodi di controller forniscono amministrazione, rete e disponibilità elevata per l'ambiente OpenStack. Un ambiente OpenStack consigliato contiene tre nodi Controller in un cluster HA.

Calcola

I nodi di elaborazione forniscono risorse di elaborazione per l'ambiente OpenStack. I nodi di elaborazione possono essere scalabili in base ai requisiti di rete nel tempo. Un nodo di calcolo predefinito contiene i seguenti componenti:

- OpenStack Compute (nova)
- VM basata su kernel (KVM)/emulatore rapido (QEMU)
- Apri vSwitch

Storage

I nodi di storage forniscono storage per l'ambiente OpenStack.



Nota: Sono disponibili diversi approcci per l'installazione di Undercloud/Director e Repository offline (REPO) nella rete del Cliente; è possibile installarli direttamente su un nodo baremetal o come VM su un hypervisor KVM. Nella guida all'installazione corrente, il server UCS Director ospita KVM (Hypervisor) per l'installazione di più VM. Il nodo RHOSP Director e i nodi Offline-REPO vengono implementati come VM su hypervisor KVM.

Repository locale (server REPO)

Redhat fornisce un'utilità chiamata `reposync` che può essere utilizzata per scaricare i pacchetti dal [Content Delivery Network](#) (CDN). Per scaricare tutti i pacchetti da un canale specifico, il sistema deve essere abbonato a quel canale. Se il sistema non è sottoscritto al canale richiesto, `reposync` non è in grado di scaricare e sincronizzare tali pacchetti sul sistema locale.

I repository sono configurati in `/etc/yum.repos.d/` percorso da file che terminano con l'estensione `.repo`. È possibile definire più repository nello stesso file.

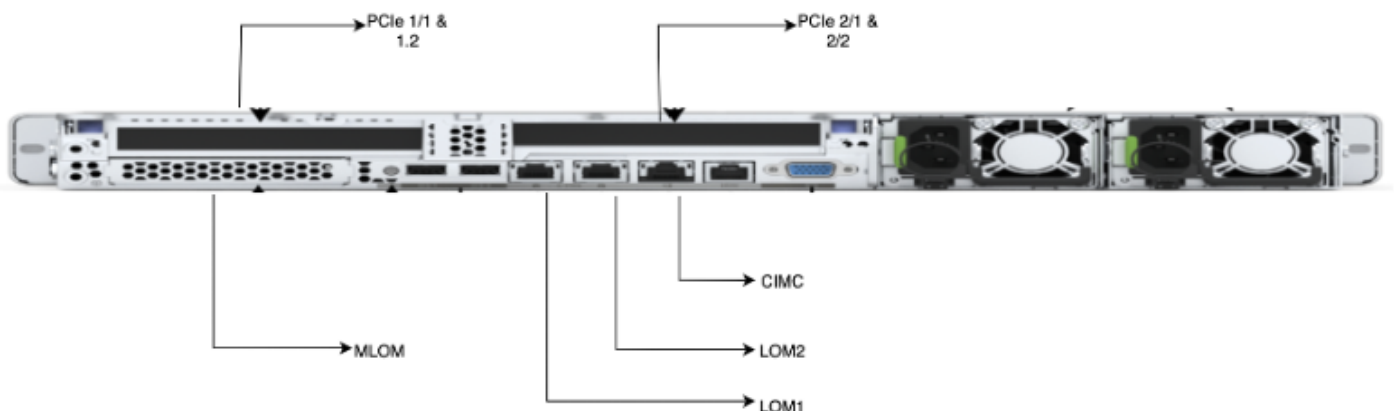
RHOSP Networking

Il servizio di rete (`neutron`) è il componente di rete definito dal software (SDN) di RHOSP. Il servizio di rete RHOSP gestisce il traffico interno ed esterno da e verso le istanze VM e fornisce servizi di base come routing, segmentazione, DHCP e metadati. Fornisce l'API per le funzionalità di rete virtuale e la gestione di switch, router, porte e firewall.

Il director RHOSP mappa i servizi OpenStack su diverse reti isolate. Le reti che trasportano ciascun tipo di traffico sono: Cisco Integrated Management Controller (CIMC), provisioning, API interna, dati di archiviazione, gestione dello storage, tenant e esterno (SSH (Secure Shell) e operazioni, amministrazione e manutenzione (OAM)).

Connettività fisica RHOSP

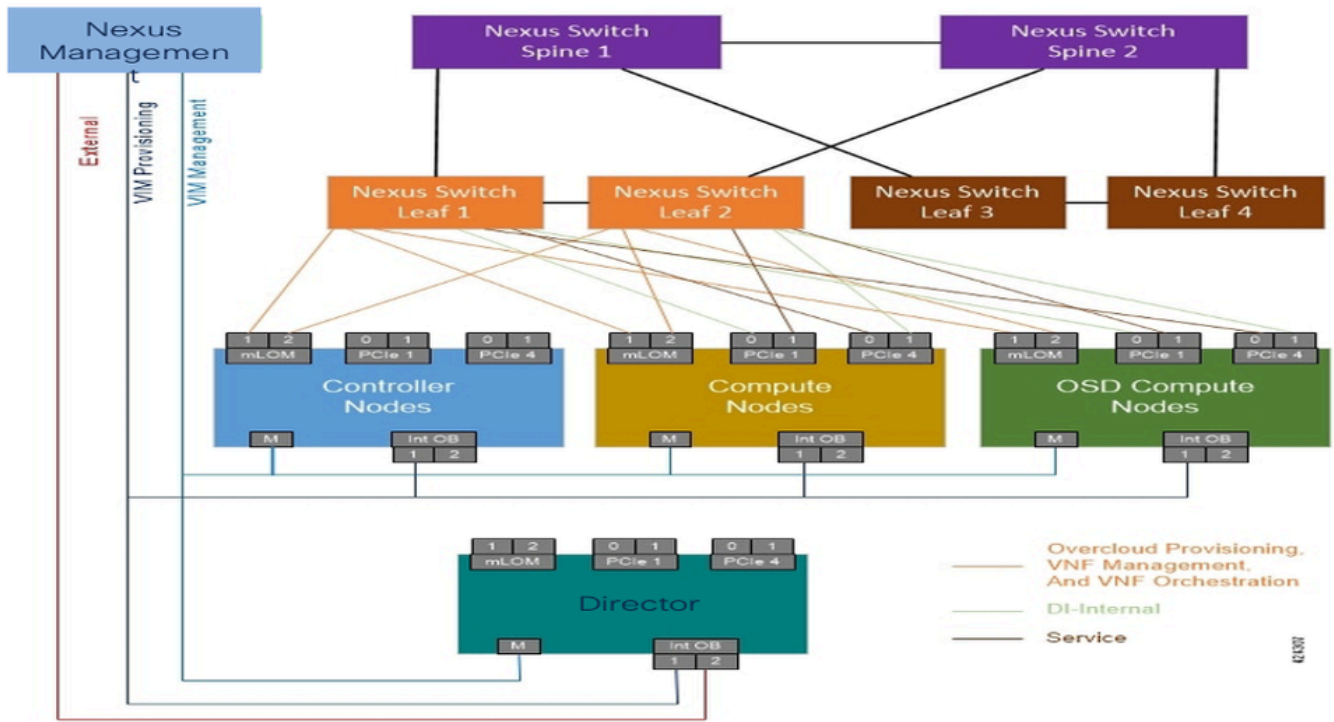
L'implementazione RHOSP utilizza diverse porte fisiche dei server Cisco UCS C220 M6 per diversi scopi di connettività.



Numero di serie	Porte fisiche	Dettagli

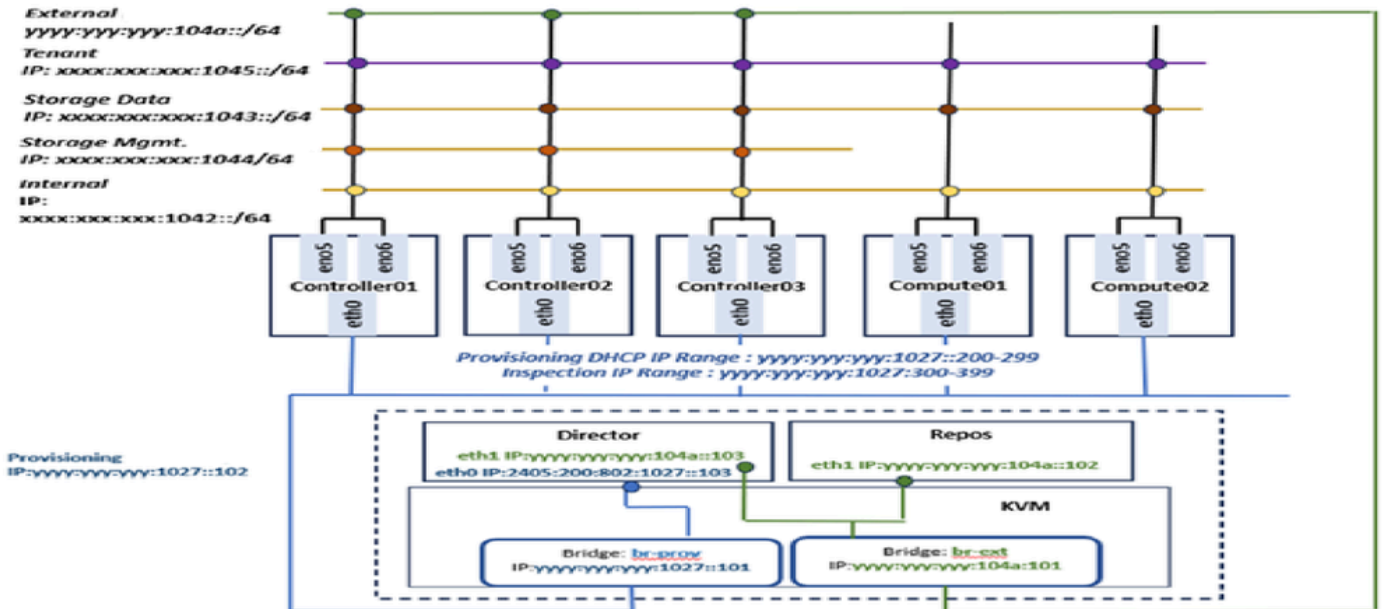
1.	CIMC	CIMC fornisce connettività fuori banda per il provisioning e la gestione dei server.
2.	Single Root I/O Virtualization (SR-IOV)/Peripheral Component Interconnect Express (PCIe)	La scheda di interfaccia di rete (NIC, Network Interface Card) PCIe viene utilizzata sui nodi di elaborazione per le reti DI-internal e di servizio per il VNF.
3.	Lan modulare su scheda madre (MLOM)	Le porte MLOM sono configurate come collegamenti. osp_external, osp_internal, osp_tenant, osp_external, osp_storage_data, osp_storage_mgmt utilizza la porta MLOM per le comunicazioni interne.
4.	Lan on MotherBoard (LOM)	Il director utilizza le porte LOM1 e LOM2, mentre i computer e i controller utilizzano solo la porta LOM1. LOM1 viene utilizzato per distribuire o effettuare il provisioning di Openstack su tutti i server. LOM2 viene utilizzato come OAM (External Network) sul director.

Il diagramma mostra la connettività fisica con i server.



Connettività logica RHOSP

La rete RHOSP dispone di più subnet che forniscono servizi diversi all'interno del cloud.



- OSP_CIMC:

CIMC è l'interfaccia IPMI (Intelligent Programming Management Interface) che controlla la gestione di tutti i server UCS. Questa rete CIMC è configurata sulla porta CIMC autonoma di tutti i

server UCS.

- OSP_Provisioning:

Questa rete è responsabile del provisioning e della gestione dell'avvio PXE (Preboot Execution Environment) dei computer e dei server controller durante l'installazione di Overcloud, nonché per ottenere l'indirizzo IP DHCP. Per semplicità e compatibilità, la rete di provisioning è configurata come VLAN nativa sulla porta LOM1 di tutti i server UCS. Questa rete di provisioning è responsabile dell'installazione del cloud su tutti i server.

A causa della virtualizzazione sul server Director, è necessario creare una rete bridge sullo switch KVM affinché la VM Director possa comunicare con gli altri server.

- OSP_Internal:

La rete API interna viene utilizzata per la comunicazione tra i servizi OpenStack come neutron, nova, keystone e così via.

La rete OSP_Internal è configurata su porte MLOM collegate sui nodi Controller e Compute.

- OSP_Tenant:

La rete tenant viene creata per impostazione predefinita nei progetti cloud per la gestione VNF. Nell'installazione corrente, per la distribuzione VNF viene creato solo il progetto OpenStack singolo.

La rete OSP_Tenant è configurata su porte MLOM collegate sui nodi Controller e Compute.

- OSP_External:

La rete esterna viene utilizzata per tutte le reti di accesso esterno (ad esempio SSH) e API.

La rete OSP_External è configurata sulla porta LOM2 sul nodo Director e sulle porte MLOM collegate sui nodi Controller e Compute.

- OSP_Storage_Data:

La rete OSP_Storage viene utilizzata per tutte le operazioni correlate all'accesso allo storage. Questo è necessario per la comunicazione tra il servizio CEPH e VNF che hanno bisogno di

accedere allo storage. Viene utilizzato da Controller, Nodi di calcolo e CEPH.

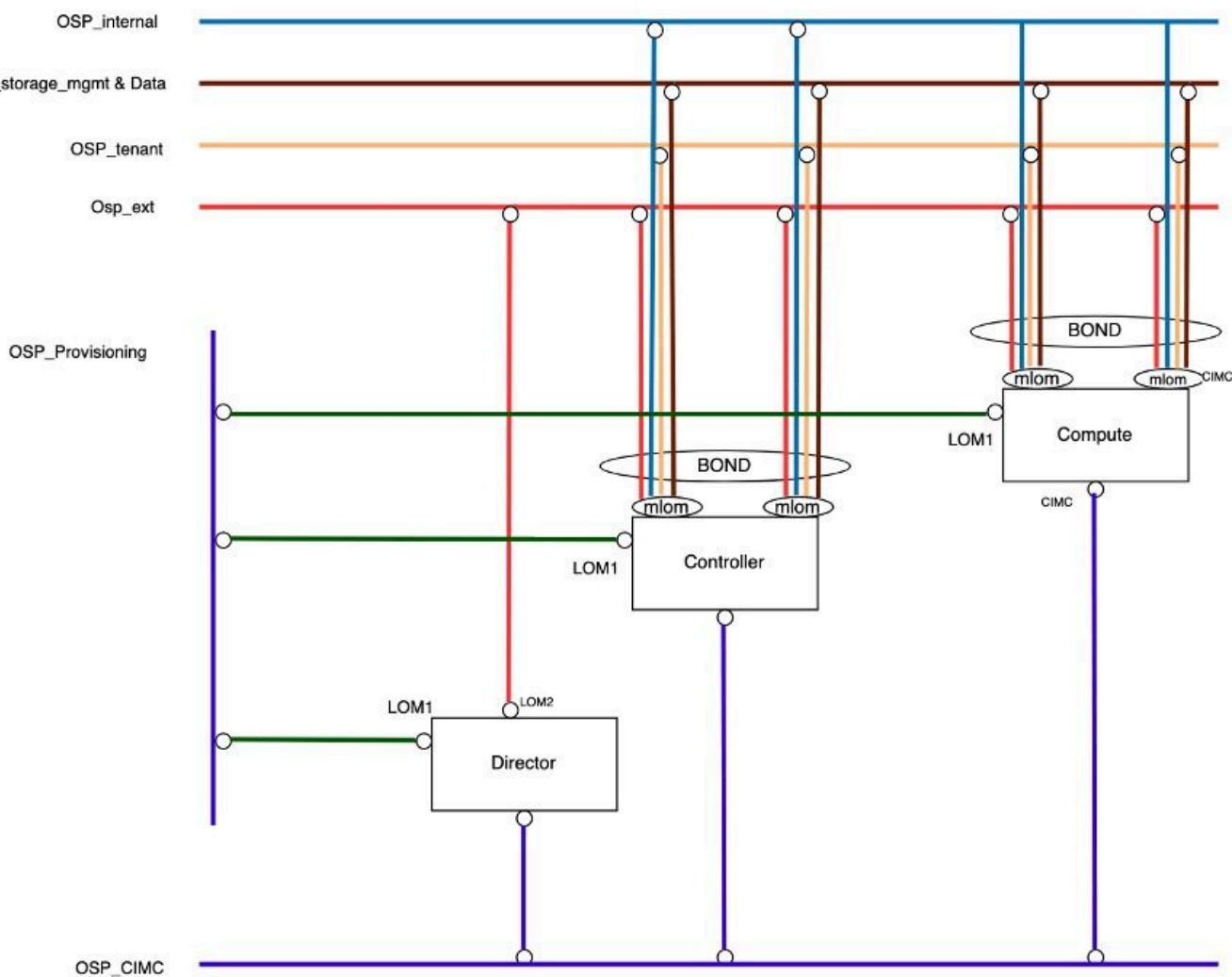
La rete OSP_Storage_Data è configurata su porte MLOM collegate sui nodi Controller e Compute.

- OSP_Storage_Management:

OpenStack Object Storage utilizza questa rete per sincronizzare gli oggetti dati tra i nodi di replica partecipanti nel cluster di archiviazione, formati tra i nodi di elaborazione dei controller.

La rete OSP_Storage_Mgmt è configurata su porte MLOM collegate sui nodi Controller e Compute.

Il diagramma mostra come le reti logiche subcloud sono connesse a ciascun tipo di nodi nel cluster RHOSP.



Regolazione parametri hardware

- Verificare che l'indirizzo IP sia abilitato e configurato con l'indirizzo di gestione corretto in base alla progettazione.
- Accertarsi che l'utente admin e la relativa password siano gli stessi per tutti gli accessi CIMC.
- Verificare che la modalità bootstrap (avvio) sia impostata su Unified Extensible Firmware Interface (UEFI).
- Assicurarsi di impostare il BIOS con i parametri consigliati.
- Imposta ordine di avvio su:
 1. LOM-PXE (eno1)
 2. Unità disco rigido di avvio

- Impostare l'indirizzo MAC di LOM1 (eno1).
- Accertarsi che tutti i server UCS/CIMC siano aggiornati con la versione firmware più recente consigliata.

Installazione e creazione della rete KVM dell'hypervisor

Esistono diversi metodi per l'installazione di Undercloud/Director e Offline REPO nella rete del cliente. Questi possono essere installati direttamente su un nodo bare-metal o come VM in esecuzione su un hypervisor KVM.

Nella guida all'installazione corrente, il server UCS Director è configurato per ospitare l'hypervisor KVM, che semplifica la creazione di più VM. Il nodo RHOSP Director e il nodo Offline REPO vengono implementati come VM su questo hypervisor KVM.



Nota: Per installare l'hypervisor KVM è necessario seguire le procedure di installazione KVM RHEL standard.

-
- Una volta attivato il KVM, aggiornare il nome host:
`hostnamectl set-hostname <hostname> - statico`

 - Configurare i bridge esterni e di provisioning e associare le interfacce.

`br-prov : eth0`

`br-ext eth1`

I bridge devono essere creati tramite l'interfaccia utente di testo (NMTUI) di Network Manager.

- Per impostazione predefinita, le porte eno1 e eno2 vengono create in KVM per le porte LOM1 e LOM2 fisiche. Eseguire la verifica incrociata utilizzando l'indirizzo MAC della registrazione CIMC.
- Creare una rete bridge e aggiungervi porte slave mappando l'indirizzo MAC corretto.
- Creazione di una rete post-bridge, verificare che il provisioning e il gateway esterno siano raggiungibili da KVM.

Creazione di VM di REPO e Director

Prerequisiti per la creazione di VM di REPO e Director

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

- Installare i pacchetti necessari in KVM per la creazione di VM di REPO e Director.

```

: command not found...
[root@MUMBKVM01 ~]#
[root@MUMBKVM01 ~]# 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+10368+630e803b.x86_64 is already installed.
Package libvirt-5.0.0-35.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

```

- Creare le directory necessarie su KVM.

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

- Copiare i file nelle directory.

```

# 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/

```

- Montare l'ISO su /mnt/iso.

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

- Creare il file REPO nel percorso /etc/yum.repos.d.

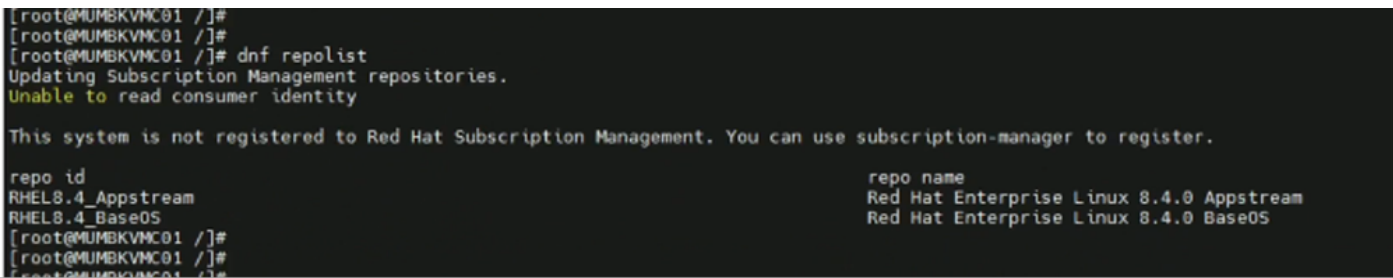
```
# cat /etc/yum.repos.d/offlinedvd.repo
```

```
[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/
```

- Controllare il repolist e assicurarsi che i REPO Appstream e Baseos siano mappati.

```
# 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 ~]#
```

Creazione VM REPO

```
$ 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]#
```

- Creare l'immagine del server REPO.

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

```
[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
[ 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/OSPRE

```
[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.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]#
```

\$ 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_REPO_RHEL_84.qcow2,device=disk,bus=virtio,format=qcow2

```
[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
```

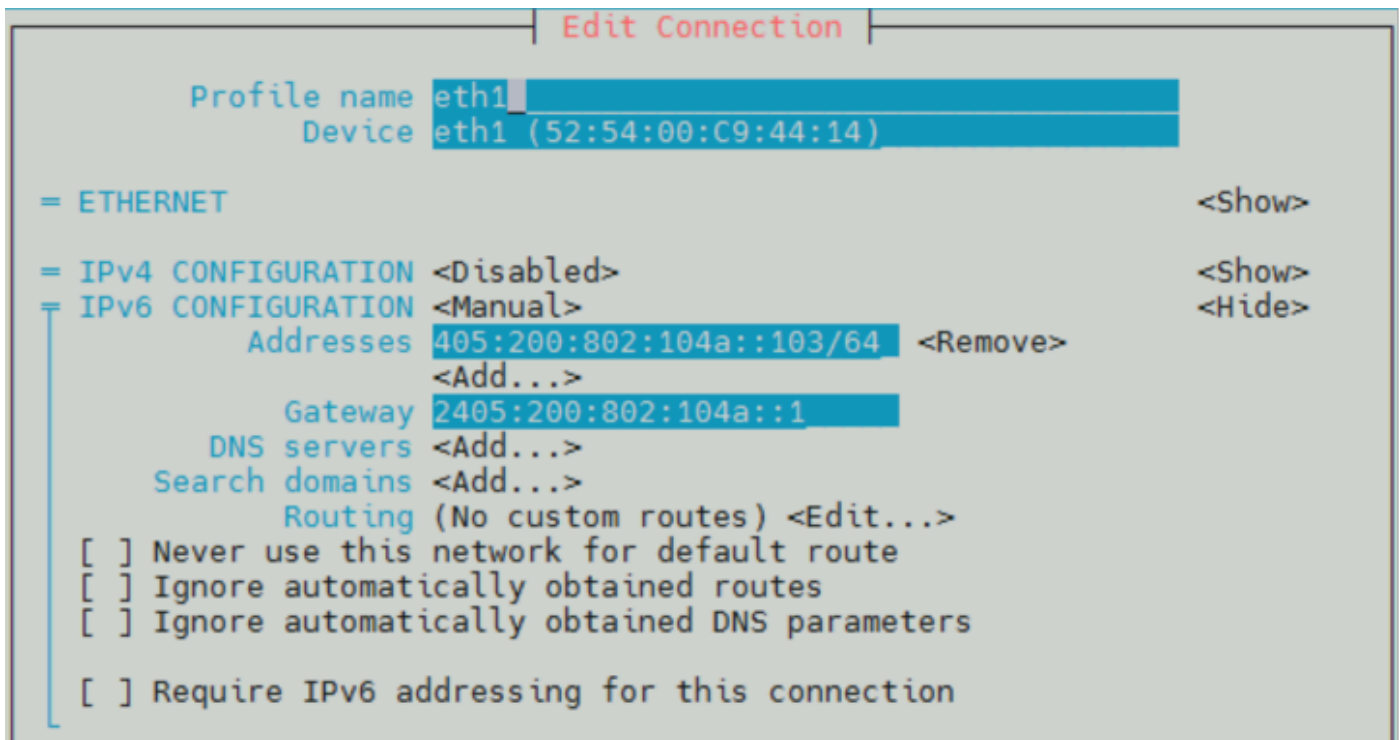
```
Domain creation completed.  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]# virsh list --all  
Id Name State  
-----  
1 OSPREPO_RHEL_84 running  
2 OSPDIRECTOR_RHEL_84 running  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]#  
[root@MUMBKVMC01 images]#
```

```
[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'
```

- Aprire una nuova finestra e accedere alla console della VM.

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

- Post Director Installazione VM, vengono create due interfacce eth0 e eth1.
 - eth0 viene utilizzato per la rete OSP_provisioning, in cui durante l'installazione di Undercloud viene creata la rete br-ctlplane.
 - eth1 viene utilizzato per la connettività esterna per l'accesso SSH. Viene quindi assegnato l'indirizzo IP della subnet esterna OSP_external.



Creazione locale di REPO offline

Il server REPO deve essere registrato con la rete CDN Redhat e deve disporre del repository di tutti i pacchetti disponibili di RHOSP 16.2 necessari per la distribuzione. I pacchetti RHEL RPM e le immagini contenitore RHOSP devono essere scaricate nella VM REPO utilizzando il proxy.

Implementazione RHOSP Cloud

RHOSP 16.2 viene implementato nella rete del cliente tramite automazione. Gli script ansible vengono utilizzati per automatizzare l'installazione di Undercloud e Overcloud.

Prerequisiti

Passaggi da seguire prima di avviare l'effettiva distribuzione cloud:

1. Garantire la connettività del nodo KVM con VM Local REPO e Director VM sulla rete OSP_EXT.
2. Verificare che tutti gli script eseguibili e gli script di automazione caricati nell'host KVM si trovino nella cartella specificata.
3. Creare cartelle denominate 'cisco' e 'automation' e inserire il tallone di script ansible.

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

La funzione Tarball è costituita da tre strutture di directory delle cartelle, denominate come:

- script: è costituito da script della shell utilizzati per la configurazione dei nodi baremetal.
- rpm: è costituito da driver Intel Ethernet Connection (ICE), driver Intel e pacchetti Red-Hat Package Manager (RPM).
- andibile: è costituito da file vars di input, file yaml andibili per la distribuzione e modelli jinja.

4. Installare il pacchetto sshpass. sshpass è un'utilità della riga di comando utilizzata per fornire password al protocollo ssh in modo non interattivo. Viene utilizzato principalmente in scenari di script o automazione in cui non è possibile immettere manualmente la password.

- Scaricare il pacchetto sshpass sshpass.tar.gz da Internet/server esistente.
- Installare il pacchetto GCC (GNU Compiler Collection).

```
# yum install gcc
```

- Installare i pacchetti Make.

```
# yum install make
```

- Annullare il pacchetto sshpass e installare il passaggio SSH.

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

5. Il processo di installazione della directory richiede un utente non root per eseguire i comandi. L'utente 'Stack' deve essere creato in Director VM con accesso 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. Copiare il file rootCA.crt dal server REPO alla macchina virtuale Director e allo switch KVM nel percorso specificato. Inoltre, aggiornare il certificato della macchina virtuale REPO nell'elenco di attendibilità.

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

7. Aggiornare i dettagli relativi al nome host del server REPO locale nel file in /etc/hosts di Director VM e KVM.

8. Su KVM e Director VM, installare pacchetti aggiuntivi come python, ansible e così via per eseguire script di automazione andibili.

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

9. La subnet CIMC deve essere raggiungibile dalla rete di provisioning di Director per abilitare il provisioning durante l'installazione del cloud. Se necessario, aggiungere la route statica per la stessa.

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

10. In KVM e Director VM, creare un file host nella cartella /ansible e aggiungere i dettagli specifici dello stack, se necessario.

```
[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. Verificare che tutti i playbook e i file di input analizzabili siano conservati nella cartella /home/stack di Director VM.

Aggiornamento dei file di input

È presente un file di variabile di input costituito da dettagli specifici della rete del cliente che devono essere preparati per l'installazione del cloud.

Percorso: /home/cisco/automation/ansible/podvars

Nome file: <nome-stack>_vars.yml

Aggiornare i parametri evidenziati in base al piano IP specifico del sito/documento di progettazione di basso livello.



Nota: Gli indirizzi IP fittizi vengono utilizzati solo a scopo di rappresentazione.

```
<#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
# #####

```

Distribuzione undercloud

Undercloud viene distribuito utilizzando gli script ansibili in sette passaggi. Tutti i passaggi devono essere eseguiti da un host KVM che in questo caso funge da host di collegamento.

Passaggio	Contrassegno	Descrizione	YAML playbook
-----------	--------------	-------------	---------------

Passaggio 1.	file di controllo	Verificare playbook, script e RPM richiesti su OpenStack Platform Director (OSPD).	osp16_published_playbooks_verifica.yml
Passaggio 2.	genpodvar	Generare file di variabili specifiche del POD correlati a hardware, RHEL e così via, ad esempio common_vars.yml (hardware, software, dettagli di rete), hw_m6_vars.yml (CPU, memoria, hugepages, disco, NIC e così via), rhel_84_vars.yml (RHEL, kernel, driver ICE, versione NIC), pf_esc_vars.yml (dettagli ESC (Elastic Services Controller)), osp_16_vars.yml (versione OSP, fuso orario, tipo IP, ID VLAN, IP, dettagli neutroni).	osp16_generate_pod_specific_vars.yml
Passaggio 3.	preucdeploy	Configurare FQDN (Fully Qualified Domain Name), NTP (Network Time Protocol) e aggiornare tutti i pacchetti nel nodo della directory.	osp16_pre_undercloud_deploy.yml
Passaggio 4.	primo riavvio	Esegue il primo riavvio del nodo della directory Undercloud dopo la configurazione e l'installazione del pacchetto precedenti.	osp16_undercloud_deploy.yml
Passaggio 5.	disinstallazione	Installa stack undercloud su director	osp16_undercloud_tuning.yml
Passaggio 6.	cstate	Configurare le impostazioni dello stato C della CPU del BIOS sul nodo della directory.	osp16_cstate.yml

Passaggio 7.	riavvio secondario	Secondo riavvio dopo modifiche del BIOS sul director Undercloud.	N/D
-----------------	-----------------------	--	-----

Il file con nome osp16_auto_undercloud_deploy.yml è un playbook principale che può essere eseguito in una singola iterazione, ma è consigliabile eseguire il playbook in modo graduale utilizzando tag diversi per facilitare la risoluzione dei problemi in caso di problemi di distribuzione.

```
<#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
```

Distribuzione cloud

Overcloud viene implementato con almeno tre controller in modalità HA e un computer. Overcloud viene distribuito utilizzando script ansibili in 17 passaggi. Tutti i passi devono essere eseguiti da Director-VM che, in questo caso, funge da host di collegamento.

Passaggio	Contrassegno	Descrizione	YAML playbook
Passaggio 1.	genpodvar	Generare file di variabili specifiche del POD per Overcloud relativi a hardware, RHEL e così via, come <code>common_vars.yml</code> (hardware, software, dettagli di rete), <code>hw_m6_vars.yml</code> (CPU, memoria, hugepages, disco, NIC e così via), <code>rhel_84_vars.yml</code> (RHEL, kernel, driver ICE, versione NIC), <code>pf_esc_vars.yml</code> (dettagli ESC), <code>osp_16_vars.yml</code> (versione OSP timezone, tipo di IP, ID VLAN, IP, dettagli neutroni).	<code>osp16_generate_pod_specific_vars.yml</code>
Passaggio 2.	geninstack	Generare il file JSON <code>Instackenv</code> da <code>/var/common_vars.yml</code> creato nel passaggio precedente. La directory richiede un modello di definizione del nodo, che viene creato manualmente. Il file <code>instackenv.json</code> utilizza il formato JSON e contiene tutti i dettagli relativi all'hardware e al risparmio energia per i nodi. Questa procedura consente inoltre di convalidare la configurazione hardware sul server UCS prima di generare	<code>osp16_generate_instackenv.yml</code>

		il file.	
Passaggio 3.	cimcvd	Configurare l'impostazione CIMC e i dischi virtuali (VD) su ciascun server che fa riferimento a common_vars.yml, hw_m6_vars.yml e rhel_84_vars.yml.	osp16_cimc_vd_configure.yml
Passaggio 4.	predistribuzione	In questo passaggio vengono eseguiti tutti i prerequisiti per la distribuzione di Overcloud. Imposta l'FQDN, l'NTP e aggiorna tutti i pacchetti, quindi esegue il push dell'immagine nel percorso per la distribuzione.	osp16_pre_overcloud_deploy.yml
Passaggio 5	importonodi	In questo passaggio vengono introdotti la CPU del server, la memoria, la scheda NIC e l'interfaccia e le porte sugli switch di rete. L'introspezione viene eseguita sugli switch di rete collegati per tutti i controller e i computer.	osp16_import_ironic_nodes.yml
Passaggio 6.	modelli	Generare file modello personalizzati per controller e computer. Nel modello personalizzato, definire i ruoli di controller e calcolo per tutti i servizi in esecuzione su di essi. Consente inoltre di rendere più sicuro il sistema applicando certificati, route e così via.	osp16_generate_custom_templates.yml
Passaggio 7.	ocdeploy	In questo passaggio viene eseguita una distribuzione di Openstack Overcloud. Eseguire deploy.sh fornito da	osp16_overcloud_deploy.yml

		Red Hat per la distribuzione di RHOSP.	
Passaggio 8	geninventario	In questo passaggio viene generato un file con il codice di inventario utilizzabile da ansible in cui l'indirizzo IP di provisioning, l'indirizzo IP di IPMI (CIMC) e le credenziali vengono archiviati e mappati con il controller e i computer per consentire l'accesso automatico al sistema ed eseguire i passaggi successivi.	osp16_build_inventory_v3.py
Passaggio 9.	offlinerepo	Configurare Overcloud Offline REPO nel file /etc/yum.repo.d/offline.repo che punta al server repo tramite la rete esterna.	osp16_config_offline_repo.yml
Passaggio 10.	recinzione	Configurare la schermatura su tutti i nodi di controller con Shoot The Other Node In The Head (una tecnica di schermatura nei cluster HA) (STONITH).	osp16_config_recinzione.yml
Passaggio 11.	raidcache	Configurare l'impostazione della cache RAID per tutti i controller e i computer e configurare anche le impostazioni di archiviazione SWIFT.	osp16_raid_cache_tuning.yml
Passaggio 12.	dnfupdate	Eseguire aggiornamenti DNF per tutti i pacchetti in tutti i nodi.	dnf_update_all_packages.yml
Passaggio 13.	setiplink	In questo passaggio, il controllo della modalità di	osp16_setlpLink.yml

		attendibilità per le porte SR-IOV è abilitato per il traffico dati e interno di Evolved Packet Data Gateway (EPDG). Il supporto per le porte SR-IOV è disponibile in neutroni e consente alle VM di accedere alla rete tramite le funzioni virtuali SR-IOV.	
Passaggio 14.	watchdog	In questo passaggio, l'impostazione IPMI sul nodo del director è configurata per l'attività di gestione su tutti i server tramite connessione fuori banda.	osp16_config_ipmi_watchdog.yml
Passaggio 15.	icedriver	Aggiornare il driver Intel E810 ICE per schede PCI (Peripheral Component Interconnect) alla versione 1.12.6 per EPDG in modo da utilizzare le porte NIC Intel come SR-IOV.	osp16_ice_driver_install.yml
Passaggio 16.	riavviare	Riavviare tutti i nodi di overcloud dopo l'esecuzione dei passaggi precedenti.	osp16_reboot_overcloud_hosts.yml
Passaggio 17.	verifica	Verificare la configurazione e l'integrità della distribuzione RHOSP.	osp16_rosopo_verifica.yml

Per il provisioning dei nodi Overcloud, Undercloud utilizza 'overcloud-hardened-uefi-full.qow2'. Pertanto, prima di iniziare l'installazione di Overcloud, l'immagine deve essere archiviata nel percorso indicato in `undercloud/director`.

Copiare il file `qws2` Overcloud dal sito 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=verifyrhosp
```

Per monitorare i log di distribuzione, utilizzare il file di log più recente.

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

Verificare che tutti i 17 passaggi siano stati completati.

Controlli non riusciti => Il conteggio deve essere 0.

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

```
#=====
```

N. FASE	TAG	DESCRIZIONE	PLAYBOOK
---------	-----	-------------	----------

```
#=====
```

# passaggio1	genpodvar	Genera file di variabili specifiche del POD	
osp16_generate_pod_specific_vars.yml -e nomepod=			

# passaggio2	geninstack	Genera file JSON Instackenv	
osp16_generate_instackenv.yml -e nomepod=			

# passaggio 3	cimcvd	Configurazione di VD CIMC	
osp16_cimc_vd_configure.yml			

# passaggio4	prelocalizzazione	Configura distribuzione pre-overcloud	
osp16_pre_overcloud_deploy.yml			

# passaggio 5	importnodi	Importa Nodi Ironici Baremetrici Openstack	
osp16_import_ironic_nodes.yml			

# passaggio6	modelli gentili	Genera modelli personalizzati	
osp16_generate_custom_templates.yml			

passaggio7 | ocdeploy | Distribuzione Openstack Overcloud |
osp16_overcloud_deploy.yml

passaggio8 | geninventario | Genera file di inventario |
osp16_build_inventory_v3.py —ipmipass

passaggio9 | eropoffline | Configurare il repository offline di Overcloud dal file TAR offline |
osp16_config_offline_repo.yml

n. passaggio10 | recinzioni | Post-distribuire il MOP prima del riavvio - Configurare la
restrizione | osp16_config_recinzione.yml

n. passaggio11 | raidcache | Post-distribuire MOP prima del riavvio - Cache Raid e tuning PR |
osp16_raid_cache_tuning.yml

n. passaggio12 | dnfupdate | Post Deploy MOP Before Reboot - Pacchetti di aggiornamento
DNF | dnf_update_all_packages.yml

n. passaggio13 | setiplink | Post Deploy MOP Before Reboot - Imposta attendibilità
collegamento IP VF ON | osp16_setIpLink.yml

n. passaggio14 | watchdog | Post-distribuire MOP prima del riavvio - Configura watchdog
IPMI | osp16_config_ipmi_watchdog.yml

n. passaggio15 | fiume icedriver | Post Deploy MOP Before Reboot - Aggiorna il driver ICE
E810 | osp16_ice_driver_install.yml

n. passaggio16 Riavvio | | Riavvia tutti i nodi di overcloud |
osp16_reboot_overcloud_hosts.yml

n. passaggio17 | verifirroso | Verifica della configurazione e dell'integrità dell'installazione di
RHOSP | osp16_rhosp_verify.yml -e nomepod=

#=====

TUTTI GLI HOST RAGGIUNGIBILI

=====

CONTROLLI ESEGUITI => 17

CONTROLLI SUPERATI => 17

CONTROLLI NON RIUSCITI => 0

=====
STATO COMPLESSIVO => SUPERATO!!
=====

Dopo la corretta distribuzione di OpenStack, verificare che il dashboard Orizzonte sia accessibile.

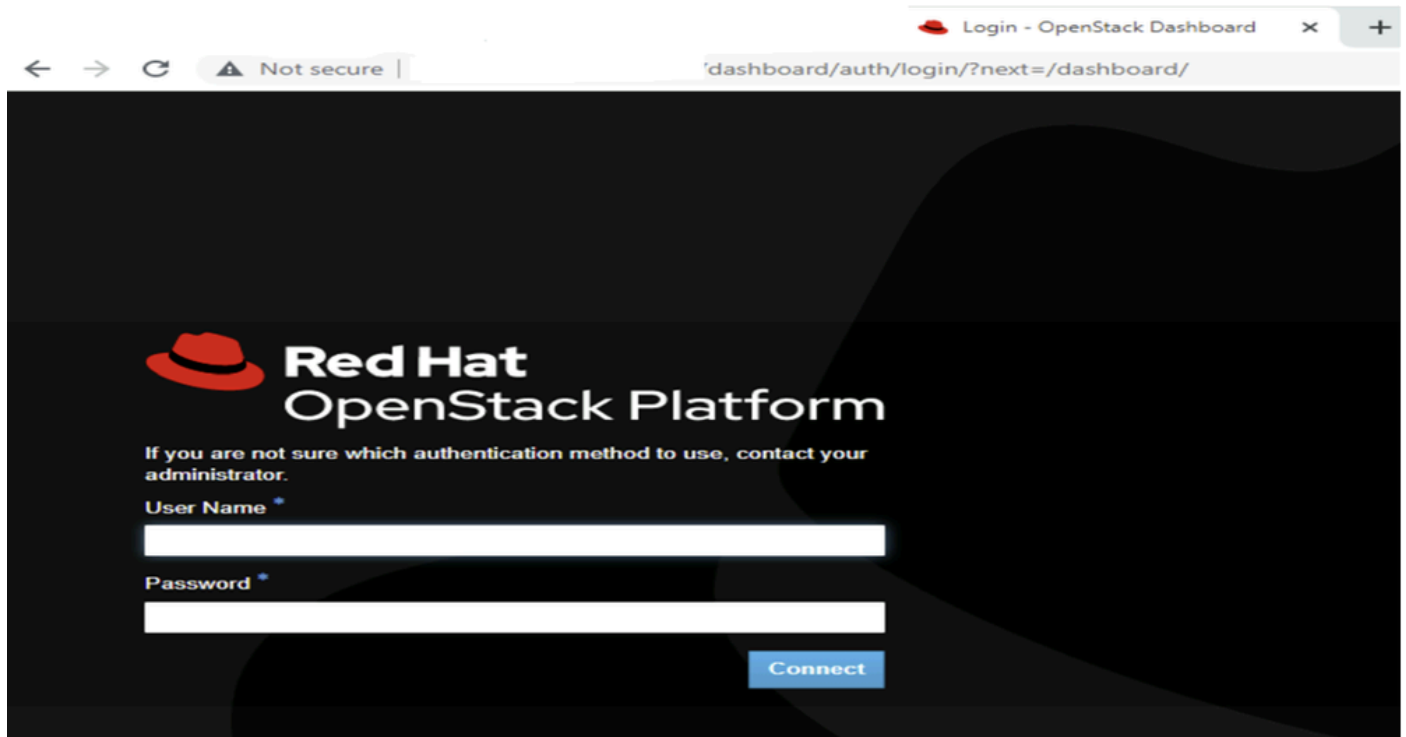
Accesso al dashboard orizzonte

Per l'URL del dashboard dell'orizzonte, utilizzare 'OS_AUTH_URL' da 'overcloud'.

```
[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]'
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
```

Dashboard orizzonte:



Verifica dello stato del cluster 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
```

Riepilogo

La guida all'installazione di RHOSP 16.2 fornisce istruzioni complete e dettagliate per l'installazione di un ambiente cloud OpenStack scalabile e pronto per la produzione utilizzando gli strumenti e le metodologie collaudate di Red Hat. Questa guida è personalizzata per gli amministratori di sistema e gli architetti cloud e si concentra sull'installazione di RHOSP 16.2 mediante il director OpenStack, basato su TripleO (OpenStack su OpenStack).

La guida copre tutte le fasi critiche dell'installazione, tra cui:

- Pianificazione dell'infrastruttura e prerequisiti
- Preparazione dell'ambiente e configurazione della rete
- Installazione e configurazione nel cloud
- Fasi di installazione e post-distribuzione cloud
- Opzioni di scalabilità di HA, storage e servizi

Questa guida è essenziale per i team alla ricerca di una piattaforma cloud affidabile di classe enterprise con l'integrazione dell'ecosistema e il supporto di Red Hat.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).