

Glance

- Glance: OpenStack Image Service
- Permette di:
 - Creare immagini
 - Fare discovery di immagini
 - Fare il retrieving di immagini
- **Immagine**: singolo file che contiene un virtual disk dove è installato un sistema operativo
- L'immagine viene usata dal servizio NOVA quando viene istanziata una virtual machine



Immagine e istanze

Launch Instance ×

[Details](#)
[Access & Security](#)
[Networking](#)
[Network Ports](#)
[Post-Creation](#)
[Advanced Options](#)

Availability Zone
nova

Instance Name

Flavor
cldareapd.tiny

Number of Instances
1

Instance Boot Source
Boot from image

Image Name
Select Image

Specify the details for launching an instance.
The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	cldareapd.tiny
VCPUs	1
Root Disk	1 GB
Ephemeral Disk	0 GB
Total Disk	1 GB
RAM	512 MB

Project Limits

Number of Instances	25 of 50 Used
Number of VCPUs	186 of 248 Used
Total RAM	331,776 of 507,904 MB Used
Number of Volumes	36 of 50 Used

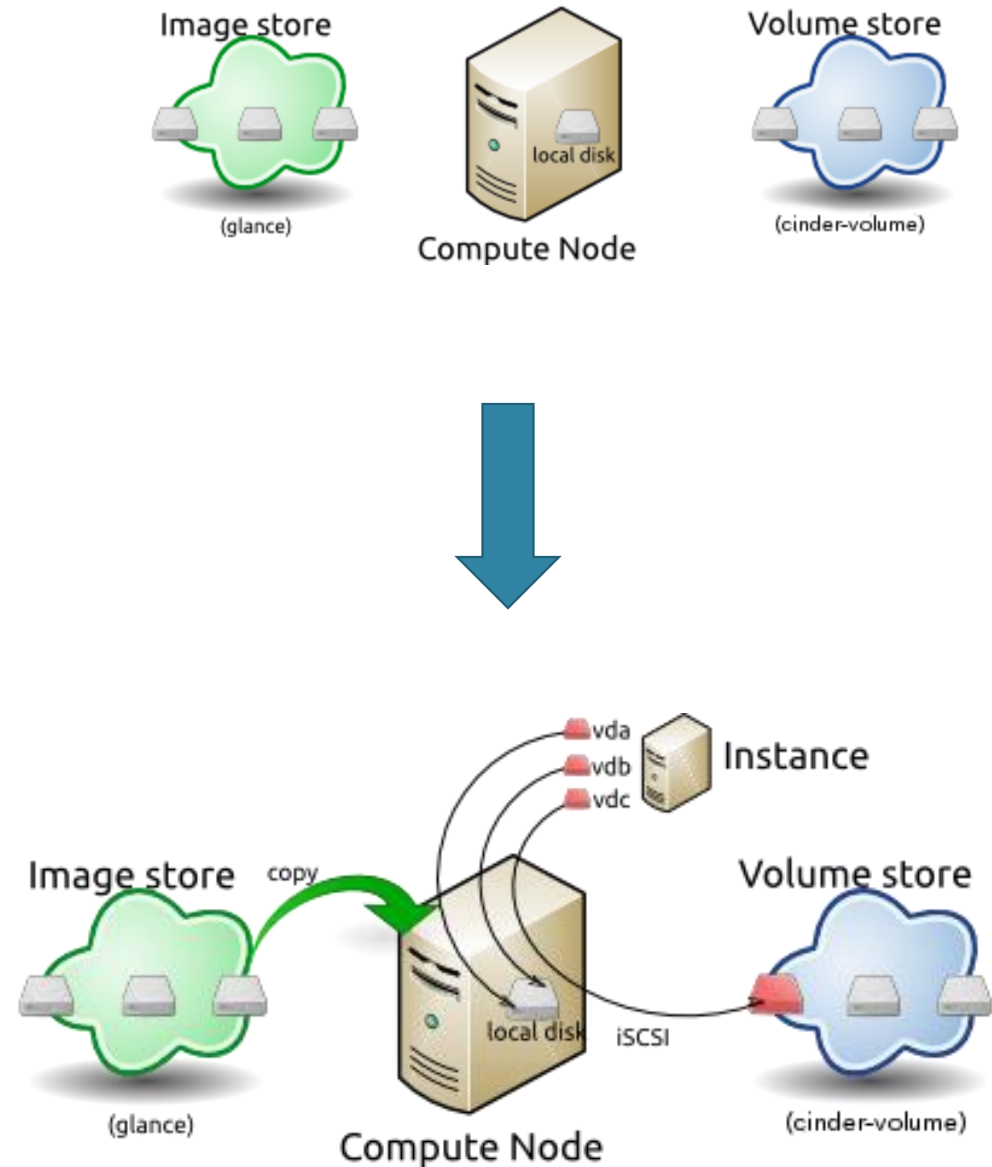
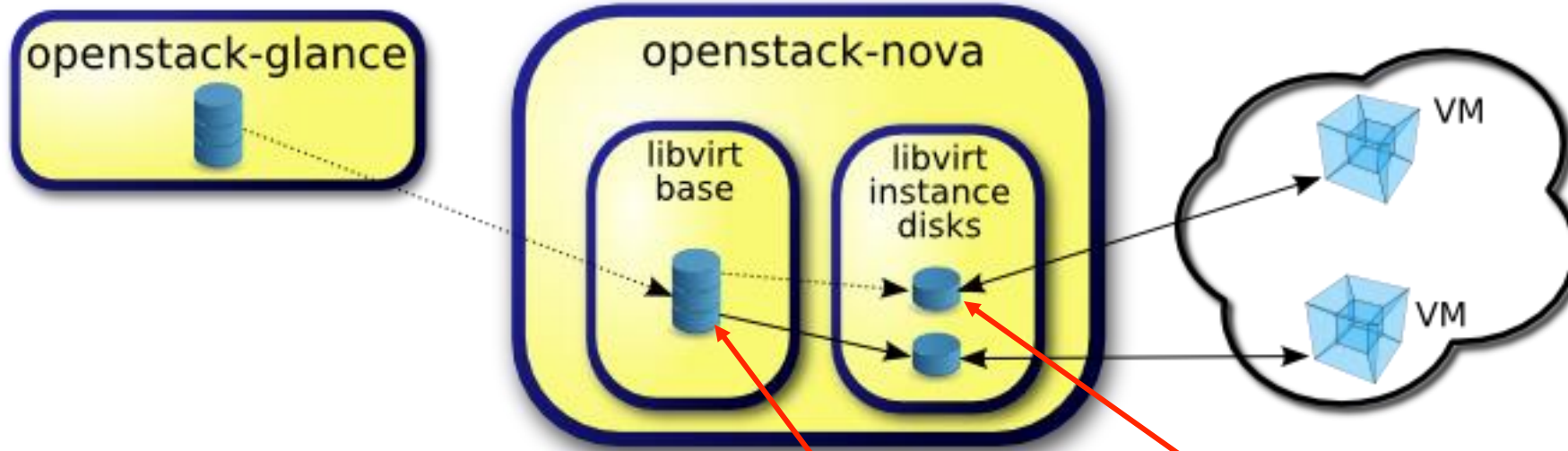


Immagine e istanze (libvirt)



```
[root@cld-np-12 ~]# qemu-img info /var/lib/nova/instances/15adf8e4-7cb7-4204-881f-a08c196a3d89/disk
image: /var/lib/nova/instances/15adf8e4-7cb7-4204-881f-a08c196a3d89/disk
file format: qcow2
virtual size: 20 GiB (21474836480 bytes)
disk size: 3.55 GiB
cluster_size: 65536
backing file: /var/lib/nova/instances/_base/f2a2b305b11572144acfc6b6cf4cb9850c6af760
Format specific information:
  compat: 1.1
  lazy refcounts: false
  refcount bits: 16
  corrupt: false
```

Immagini e istanze (boot from volume)

Launch Instance

Details *

Source *

Flavor *

Networks *

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source ¹

Image

Create New Volume ²

Yes No

Volume Size (GB) ³

1

Delete Volume on Instance Delete ⁴

Yes No

Allocated

Name	Updated	Size	Type	Visibility
Select an item from Available items below				

Available ¹⁷ Select one

Click here for filters.

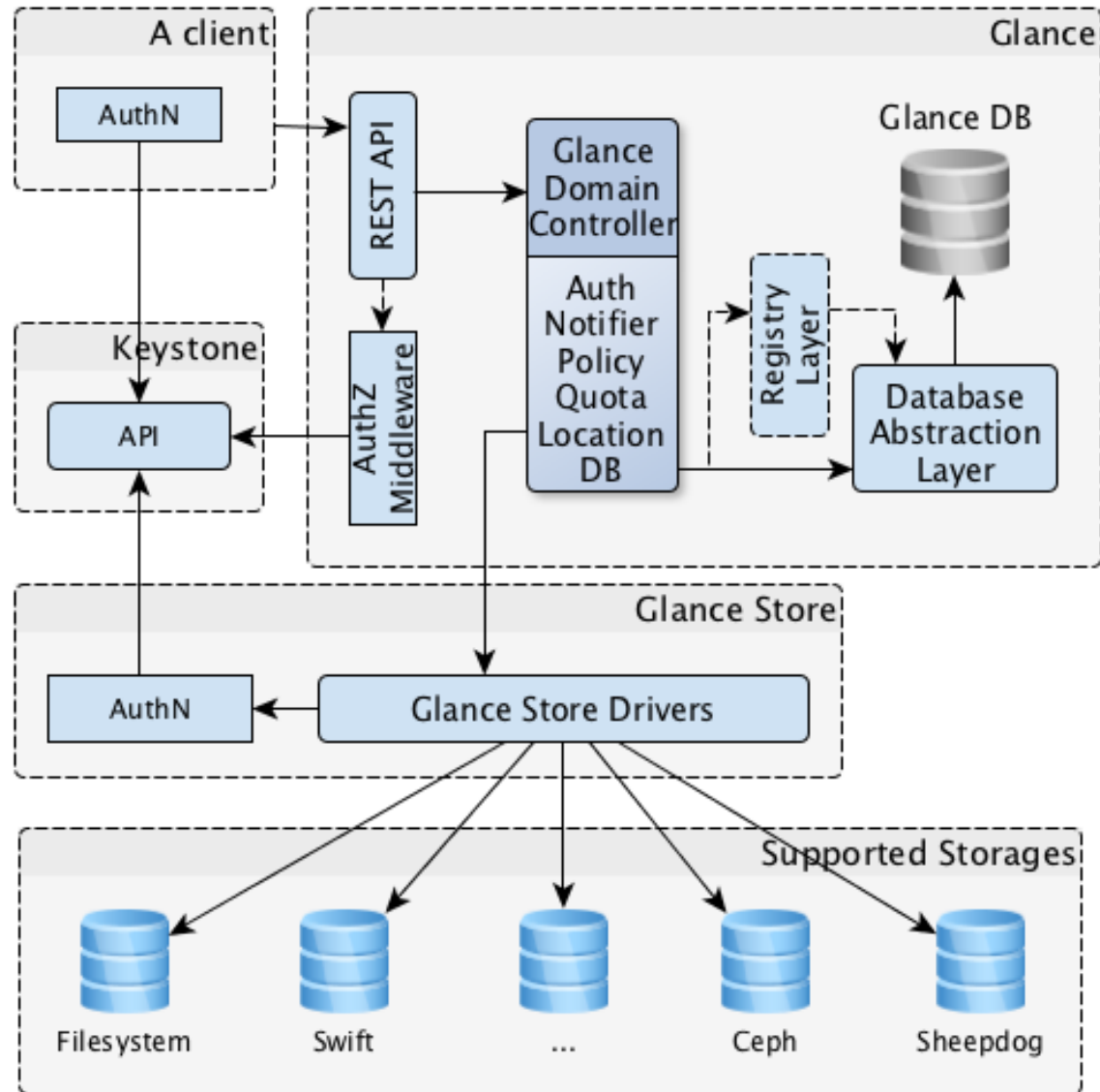
Name	Updated	Size	Type	Visibility
> murano-openvpn-server-image	3/11/19 2:21 PM	2.66 GB	raw	Public
> OpenVPN Server	2/8/19 2:27 PM	3.13 GB	raw	Public
> Ubuntu 18.04 LTS 2018-12-07	12/7/18 8:32 PM	2.62 GB	raw	Public
> OSGeo 11.0 2018-10-09	10/9/18 8:20 PM	16.02 GB	raw	Public
> Ubuntu 16.04 LTS 2018-08-14	8/14/18 9:34 PM	5.00 GB	raw	Public
> CentOS 6 2018-08-01	8/1/18 3:47 AM	5.00 GB	raw	Public
> Ubuntu 14.04 LTS 2018-07-31	7/31/18 2:36 AM	5.00 GB	raw	Public
> CentOS 7 2018-07-18	7/18/18 3:08 AM	5.00 GB	raw	Public
> App Catalog Image	10/24/18 12:20 PM	2.50 GB	raw	Public

Metodo alternativo per creare le istanze

L'immagine può essere usata per creare un bootable volume

Da quel volume viene creata una istanza

Architettura di Glance



Glance Storage Backends

- Glance supporta diversi backend:
 - Filesystem
 - HTTP (read-only)
 - Swift
 - S3
 - RBD (Ceph)
 - Cinder
 - VMWare

Installazione

Glance si installa in genere sul controller node

```
$ mysql -u root -p
MariaDB [(none)]> CREATE DATABASE glance;
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' \
  IDENTIFIED BY 'GLANCE_DBPASS';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' \
  IDENTIFIED BY 'GLANCE_DBPASS';
```

```
$ openstack user create --domain default --password-prompt glance
$ openstack role add --project service --user glance admin
```



Database

Installazione

```
$ mysql -u root -p
MariaDB [(none)]> CREATE DATABASE glance;
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' \
  IDENTIFIED BY 'GLANCE_DBPASS';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' \
  IDENTIFIED BY 'GLANCE_DBPASS';
```

Creazione utente
glance

```
$ openstack user create --domain default --password-prompt glance
$ openstack role add --project service --user glance admin
```


Installazione

```
$ openstack service create --name glance \  
  --description "OpenStack Image" image  
  
$ openstack endpoint create --region RegionOne \  
  image public http://controller:9292  
$ openstack endpoint create --region RegionOne \  
  image internal http://controller:9292  
$ openstack endpoint create --region RegionOne \  
  image admin http://controller:9292
```

Creazione servizio e
endpoint

```
# yum install openstack-glance
```

Installazione

```
$ openstack service create --name glance \  
  --description "OpenStack Image" image \  
  
$ openstack endpoint create --region RegionOne \  
  image public http://controller:9292 \  
$ openstack endpoint create --region RegionOne \  
  image internal http://controller:9292 \  
$ openstack endpoint create --region RegionOne \  
  image admin http://controller:9292
```

Installazione pacchetti

```
# yum install openstack-glance
```

Configurazione

/etc/glance/glance-api.conf

```
[database]
# ...
connection = mysql+pymysql://glance:GLANCE_DBPASS@controller/glance
[keystone_authtoken]
# ...
www_authenticate_uri = http://controller:5000
auth_url = http://controller:5000
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password = GLANCE_PASS

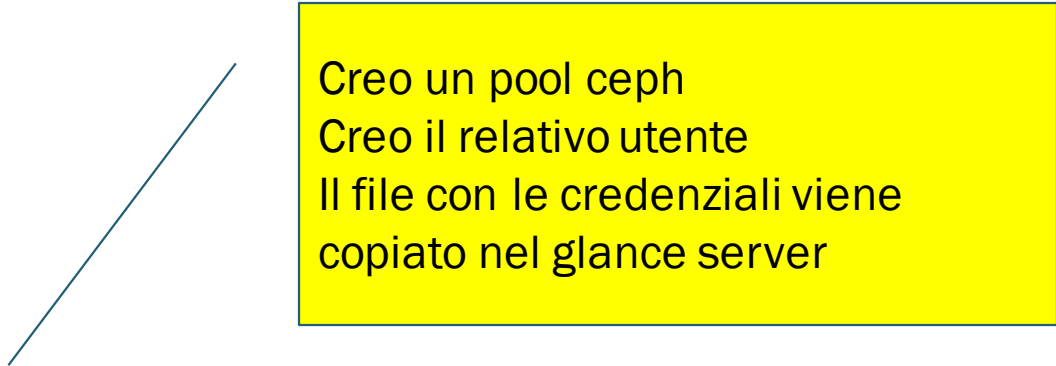
[paste_deploy]
# ...
flavor = keystone
```

Configurazione (filesystem backend)

[/etc/glance/glance-api.conf](#)

```
[glance_store]
# ...
stores = file
default_store = file
filesystem_store_datadir = /var/lib/glance/images/
```

Ceph backend (operazioni sul ceph node)



Creo un pool ceph
Creo il relativo utente
Il file con le credenziali viene
copiato nel glance server

```
$ ceph osd pool create images  
$ rbd pool init images  
$ ceph auth get-or-create client.glance mon 'profile rbd' osd 'profile rbd pool=images' mgr \  
'profile rbd pool=images'  
$ ceph auth get-or-create client.glance | ssh {your-glance-api-server} \  
$ sudo tee /etc/ceph/ceph.client.glance.keyring
```

Configurazione ceph backend (op. sul controller node)

```
$ yum install python-rbd
```

/etc/glance/glance-api.conf

```
[glance_store]
# ...
stores = rbd
default_store = rbd
rbd_store_ceph_conf=/etc/ceph/ceph.conf
rbd_store_user=glance
rbd_store_pool=images
rbd_store_chunk_size=8
```

Installazione: step finali

Popolo il database

```
# su -s /bin/sh -c "glance-manage db_sync" glance
```

```
# systemctl enable openstack-glance-api.service  
# systemctl start openstack-glance-api.service
```

Installazione: step finali

```
# su -s /bin/sh -c "glance-manage db_sync" glance
```



Start del servizio

```
# systemctl enable openstack-glance-api.service  
# systemctl start openstack-glance-api.service
```


Installazione: step finali

```
# su -s /bin/sh -c "glance-manage db_sync" glance
```



Start del servizio

```
# systemctl enable openstack-glance-api.service  
# systemctl start openstack-glance-api.service
```

In versioni più vecchie di openstack i servizi da configurare erano due: glance-api e glance-registry

Registrazione di una immagine

- Si fornisce il file immagine
- Eventualmente si forniscono dei metadati associati all'immagine
- L'immagine viene memorizzata nel backend
- Viene creata una entry nel database
 - Location del file immagine nel backend
 - Metadati forniti
 - Altre informazioni

Metadati

- Forniscono informazioni riguardo l'immagine (es. architettura, OS name, OS version, etc.)
- Sono memorizzati nel record associato all'immagine
 - Creati quando l'immagine viene registrata e/o successivamente
 - Possono essere modificati
- Vengono usati (per discovery):
 - da altri servizi OpenStack
 - Es. NOVA puo` usare l'informazione per scegliere l'hypervisor dove creare una VM istanziata usando questa immagine
 - Deve essere abilitato *ImagePropertiesFilter* nello scheduler di NOVA
 - da altri consumer (in maniera programmatica o manuale)
 - es. il middleware di INFN-Cloud (v. presentazione di venerdì)

Snapshot

- Glance viene anche usato per memorizzare snapshot di Virtual Machine
- Snapshot servono:
 - per backup
 - per "spostare" una istanza in un altro ambiente
 - come meccanismo per creare nuove immagini da usare per istanziare altre VM
- Per una snapshot sono definiti i seguenti metadati:
 - *image_type*: snapshot
 - *instance_uuid*: UUID della VM di cui si è fatto lo snapshot
 - *base_image_ref*: UUID della immagine che è stata usata per istanziare la VM di cui si è fatto lo snapshot

- Glance supporta diversi formati delle immagini
- Due formati da considerare:
 - Formato del container
 - Il file dell'immagine può essere un container che contiene il virtual disk
 - Il formato del virtual disk

Formati dei container supportati

- bare (no container)
- ovf (Open Virtualization Format)
- aki (Amazon kernel Image)
- ari (Amazon ramdisk image)
- ami (Amazon machine image)
- ova (OVF file contained in a single tar file)
- docker (docker tar archive of the docker filesystem)
- compressed (non è specificato il formato usato per la compressione: sta al consumer capirlo)

Formati dei virtual disk supportati

- raw: raw disk image (i.e. il dd del root disk su un file)
- qcow2: qemu emulator format (KVM/Xen)
- vhd: Microsoft virtual disk (Hyper-V/Azure)
- vhdx: enhanced vdh
- vmdk: VMWare disk image format
- vdi: VirtualBox disk image format
- iso: CD/DVD image format
- docker: docker images
- aki/ari/ami: Amazon Kernel/Ramdisk/Machine images

Tipi di immagini

- **Pubblica**
 - Fornita dall'amministratore della Cloud
 - Visibile a tutti gli utenti di tutti i progetti
- **Privata**
 - Appartiene a uno specifico progetto
 - Usabile da tutti e soli gli utenti di quel progetto
- **Shared**
 - Immagine privata che viene condivisa con specifici altri progetti
- **Community**
 - Immagine creata da un progetto
 - Messa a disposizione di tutti gli altri utenti
 - Senza definirne la lista dei "consumer"

Sharing di una immagine

```
$ . Prj1-openrc.sh  
$ openstack image add project <image-id> <Prj2-id>
```

Prj1 condivide una immagine con il progetto Prj2

```
$. Prj2-openrc.sh  
$ openstack image set --accept <image-id>
```

Sharing di una immagine

```
# . Prj1-openrc.sh  
# openstack image add project <image-id> <Prj2-id>
```

Prj2 deve accettare
l'immagine prima di
poterla usare

```
# . Prj2-openrc.sh  
# openstack image set --accept <image-id>
```

Accesso alle funzionalità di Glance



- API
- Dashboard
- openstack command line (*openstack image*)
 - uniform command line per tutti i servizi OpenStack
- glance command line (*glance ...*)
 - Alcune funzionalità sono disponibili solo attraverso questa CLI

Dashboard (accesso amministrativo)

The screenshot shows the OpenStack dashboard administrative interface. The top header displays the user 'admin' in two locations, both circled in red. The left sidebar contains navigation options, with 'Admin' and 'Images' also circled in red. The main content area shows the 'Images' page with a search bar and a table of images.

Admin / Compute / Images

Images

Click here for filters or full text search. + Create Image Delete Images

Displaying 20 items | [Next >](#)

<input type="checkbox"/>	Owner	Name [▲]	Type	Status	Visibility	Protected	Disk Format	Size	
<input type="checkbox"/>	> admin	185fed6a-6e96-4eda-a7ff-a4ea1fb92f7b	Image		Shared	No	-	0 bytes	Delete Image
<input type="checkbox"/>	> NSR	200404_4cores64gb	Snapshot	Active	Image from Other Project - Non-Public	No	QCOW2	8.51 GB	Launch ▼
<input type="checkbox"/>	> NSR	201006_NSR05_CentOS7_8cores40GB20+500GB1TitanXp	Snapshot	Active	Image from Other Project - Non-Public	No	QCOW2	12.10 GB	Launch ▼
<input type="checkbox"/>	> CMS	40MHz_Dask_2020-11-02	Snapshot	Active	Image from Other Project - Non-Public	No	QCOW2	23.70 GB	Launch ▼
<input type="checkbox"/>	> GAMMA	88bc2c46-e8e6-4776-9dce-185d44baa3c8	Image		Image from Other Project - Non-Public	No	RAW	0 bytes	Delete Image
<input type="checkbox"/>	> Fis03teorici	ActivePolymers_withlammps	Snapshot	Active	Image from Other Project - Non-Public	No	QCOW2	0 bytes	Launch ▼

Creazione immagine via dashboard

Create Image ✕

?

Image Details
Specify an image to upload to the Image Service.

Image Name

Image Description

Image Source

File

Format

Image Requirements

Kernel

Ramdisk

Architecture

Minimum Disk (GB) **Minimum RAM (MB)**

Image Sharing

Visibility

Protected

Creazione immagine via CLI

```
openstack image create \  
--public \  
--container-format bare \  
--disk-format qcow2 \  
--min-disk 2 \  
--min-ram 512 \  
--file /home/images/C7/CentOS7-cloudimage.qcow2 \  
Centos7-x86_64
```

Requirement per immagini Linux

- Root partition resizable
 - La dimensione del disco di una VM viene scelto dall'utente (scegliendo un opportuno flavour)
 - cloud-init / cloud-tools deve essere installato nell'immagine
- No Indirizzi MAC
 - Rimuoverli dai network scripts
 - Assicurarasi che i file `/etc/udev/rules.d/70-persistent-net.rules` e `/lib/udev/rules.d/75-persistent-net-generator.rules` siano vuoti
- SSH installato e abilitato al boot, configurato per permettere accesso via chiave

Requirement per immagini Linux (cont.ed)

- SSH injection key abilitato
 - cloud-init deve essere installato
- Capacità di processare gli user data
 - cloud-init deve essere installato
- Attenzione al firewall
 - Consigliato disabilitare firewall nell'immagine e gestire le regole di accesso delle istanze attraverso security group

Dove trovo immagini da caricare in Glance ?

- Immagini "ufficiali" disponibili per diverse distribuzioni
 - CentOS
 - <http://cloud.centos.org/centos/7/images/>
 - <https://cloud.centos.org/centos/8/>
 - <https://cloud.centos.org/centos/8-stream/>
 - Ubuntu
 - <https://cloud-images.ubuntu.com/>
 - Debian
 - <http://cdimage.debian.org/cdimage/openstack/>
 - Fedora
 - <https://alt.fedoraproject.org/cloud/>
 - ...

E se servono customizzazioni wrt le immagini "ufficiali" ?

- Varie opzioni possibili
 - Implemento le customizzazioni quando l'istanza viene creata
 - Vendor data (configurazione di nova)
 - Definito dal cloud admin: vale per tutte le VM che vengono istanziate
 - Sfrutto il meccanismo dello snapshot
 - Creo da zero una immagine/modifico un'immagine esistente, usando uno dei tool disponibili
 - guestfish
 - OZ
 - virt-customize
 - virt-builder
 - ...

Vendor-data

Abilito il meccanismo dei vendor data

/etc/nova/nova.conf

```
[api]
vendordata_providers = StaticJSON
vendordata_jsonfile_path = /etc/nova/vendor-data.json
```

/etc/nova/vendor-data.json

```
{"cloud-init": "#cloud-boothook\n#!/bin/bash\n/usr/bin/curl -m20 -s http://cld-vmlog.pd.infn.it/cloud-vendordata-script.sh | /bin/bash\n"}
```

cloud-vendordata-script.sh

```
#!/bin/bash
export PATH=/usr/sbin:/usr/bin:/sbin:/bin
export instance_id="`cat /var/lib/cloud/data/instance-id`"
logger "Vendor data: configure rsyslog"
export LOGSERVER=cld-vmlog.pd.infn.it
echo "\$template CloudFormat, \"%TIMESTAMP% $instance_id %syslogtag%%msg:::sp-if-no-1st-sp%%msg:::drop-last-1f%\n\" > /etc/rsyslog.d/99-cloudveneto.conf
echo "*.* @${LOGSERVER};CloudFormat" >> /etc/rsyslog.d/99-cloudveneto.conf
# Restart rsyslog if needed
pidof rsyslogd && service rsyslog restart
logger "Vendor data injected on $LINUX_DISTRIBUTION host"
```

Vendor-data

/etc/nova/nova.conf

```
[api]
vendordata_providers = StaticJSON
vendordata_jsonfile_path = /etc/nova/vendor-data.json
```

Definisco cosa deve essere fatto per ogni istanza che viene creata (in questo caso l'esecuzione di uno script disponibile su un server web)

/etc/nova/vendor-data.json

```
{"cloud-init": "#cloud-boothook\n#!/bin/bash\n/usr/bin/curl -m20 -s http://cld-vmlog.pd.infn.it/cloud-vendordata-script.sh | /bin/bash\n"}
```

cloud-vendordata-script.sh

```
#!/bin/bash
export PATH=/usr/sbin:/usr/bin:/sbin:/bin
export instance_id="`cat /var/lib/cloud/data/instance-id`"
logger "Vendor data: configure rsyslog"
export LOGSERVER=cld-vmlog.pd.infn.it
echo "\$template CloudFormat, \"\%TIMESTAMP% $instance_id %syslogtag%%msg:::sp-if-no-1st-sp%%msg:::drop-last-1f%\n\" > /etc/rsyslog.d/99-cloudveneto.conf
echo "*.* @${LOGSERVER};CloudFormat" >> /etc/rsyslog.d/99-cloudveneto.conf
# Restart rsyslog if needed
pidof rsyslogd && service rsyslog restart
logger "Vendor data injected on $LINUX_DISTRIBUTION host"
```

Vendor-data

/etc/nova/nova.conf

```
[api]
vendordata_providers = StaticJSON
vendordata_jsonfile_path = /etc/nova/vendor-data.json
```

/etc/nova/vendor-data.json

```
{"cloud-init": "#cloud-boothook\n#!/bin/bash\n/usr/bin/curl -m20 -s http://cloudveneto.infn.it/cloud-init/script.sh | /bin/bash\n"}
```

Script (nel server web): in questo esempio configura rsyslog per le istanze

cloud-vendordata-script.sh

```
#!/bin/bash
export PATH=/usr/sbin:/usr/bin:/sbin:/bin
export instance_id="`cat /var/lib/cloud/data/instance-id`"
logger "Vendor data: configure rsyslog"
export LOGSERVER=cld-vmlog.pd.infn.it
echo "\$template CloudFormat, \"%TIMESTAMP% $instance_id %syslogtag%%msg:::sp-if-no-1st-sp%%msg:::drop-last-1f%\n\" > /etc/rsyslog.d/99-cloudveneto.conf
echo "*.* @${LOGSERVER};CloudFormat" >> /etc/rsyslog.d/99-cloudveneto.conf
# Restart rsyslog if needed
pidof rsyslogd && service rsyslog restart
logger "Vendor data injected on $LINUX_DISTRIBUTION host"
```

Snapshotting per customizzazione

1. Creo una istanza partendo da una immagine
 2. Implemento le customizzazioni desiderate su questa istanza
 3. Faccio uno snapshot della istanza
 4. Uso questo snapshot come immagine per far partire altre VM
- E` il metodo più semplice ma:
 - L'immagine file prodotto è in genere grande
 - Le istanze che verranno fatte partire dalla nuova immagine dovranno avere root disk \geq root disk della VM di cui si è fatta la snapshot

virt-builder

Esempio:

```
virt-builder -v -x centos-7.7 \  
  --uninstall "chrony" \  
  --install "ntp,cloud-init,cloud-utils,cloud-utils-growpart,gdisk" \  
  --timezone Europe/Rome" \  
  --write '/etc/ntp.conf:server ntp.pd.infn.it' \  
  --run-command 'systemctl enable ntpd' \  
  --edit '/etc/resolv.conf:s/nameserver 10.0.2.3//' \  
  --output c7.7-test-grow.qcow2 --format qcow2 --selinux-relabel
```

- Ref: <https://libguestfs.org/virt-builder.1.html>

oz-install-d 4 -t 10000 -u -a centos7x.kickstart -p centos7x_padova-oz.xml

```
<template>
<name>centos7x-INFNpadova</name>
<description>CentOS 7.x x86_64 for Padova</description>
<disk>
  <size>4G</size>
</disk>
<os>
  <name>CentOS-7</name>
  <version>4</version>
  <arch>x86_64</arch>
  <install type='iso'>
    <iso>file:///Images/centos/7/isos/x86_64/CentOS-7-x86_64-DVD-2003.iso</iso>
  </install>
</os>
<commands>
  <command name="install-epel-7">
rpm --import http://dl.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL
rpm --import http://dl.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-7
rpm -ivh http://dl.fedoraproject.org/pub/epel/7/x86\_64/Packages/e/epel-release-7-13.noarch.rpm
  </command>
...
...
```


Disattivazione di una immagine

- Meglio non cancellare immagini se ci sono istanze attive create da questa immagine
 - Ci potrebbero essere problemi con operazioni quali resize/migration
- Meglio disattivare l'immagine in questi casi
 - Non sarà più possibile creare nuove istanze da quella immagine

Per disattivare una immagine:

```
openstack image set --deactivate<image-id>
```

Per controllare se ci sono istanze che stanno usando una certa immagine:

```
openstack server list --all --image <image-id>
```

Cancellazione di una immagine

- Una immagine può essere protetta per prevenire cancellazioni accidentali

```
openstack image set --protected <image-id>
```

Per cancellare una immagine:

```
openstack image delete <image-id>
```

Quota

- Glance non supporta quote per progetto
- C'è solo la possibilità di definire delle global quota

`image_size_cap=SIZE`

Optional. Default: 1099511627776 (1 TB)

Maximum image size, in bytes, which can be uploaded through the Glance API server.

`user_storage_quota=SIZE`

Optional. Default: 0 (Unlimited).

This value specifies the maximum amount of storage that each user can use across all storage systems

- *image_size_cap* setta la dimensione massima anche per le snapshot
- Es. Settare questo valore a 50GB vuol dire che VM con root disk > 50GB NON possono essere snapshottabili

Policies

- Una immagine privata appartiene a un progetto, non a un utente
- Un utente può cancellare/modificare l'immagine creata da un altro utente dello stesso progetto
- Per default tutti possono creare/modificare/cancellare immagini
 - Solo admin può farlo per quelle pubbliche
- E` possibile definire delle policies per restringere certe operazioni a certi insiemi di utenti

Ref: <https://docs.openstack.org/glance/latest/admin/policies.html>

Multi store

- E` possibile configurare glance perché usi contemporaneamente più backend
- E` possibile specificare quale backend usare quando si crea una immagine
- E` possibile specificare un default backend

Multi-store

```
[DEFAULT]
enabled_backends = fast:rd, reliable:file
...
[glance_store]
default_backend = fast
...

[reliable]
filesystem_store_datadir = /opt/stack/data/glance/reliable
store_description = "Reliable filesystem backend"

[fast]
store_description = "Fast rbd backend"
rbd_store_chunk_size = 8
rbd_store_pool = images
rbd_store_user = admin
rbd_store_ceph_conf = /etc/ceph/ceph.conf
```

```
# glance image-create -store ...
```

Migrazione da un backend a un altro

1. Aggiungo in glance il nuovo store
2. Setto come default il nuovo store
3. Per ogni immagine presente nel vecchio store
 1. Copio il relativo file nel nuovo store
 2. Modifico il "location" dell'immagine aggiungendo il "path" nel nuovo store
4. Quando voglio dismettere il vecchio store
 1. Per le immagini in cui avevo definito la doppia location, elimino quella dello store da dismettere
 2. Elimino il vecchio store, "togliendolo" dal file di configurazione di nova

Download di una immagine/snapshot

- E` possibile fare il download di una immagine registrata in glance
 - Immagine o snapshot
 - A cosa serve ?
 - per backup
 - per migrare l'immagine in un altro tenant o in un'altra cloud

```
# openstack image save -file <file> <image-id>
```


Troubleshooting

```
# systemctl status openstack-glance-api
```

```
# less /var/log/glance/api.log
```

Alcune best practices

- Usare storage affidabile (opportunamente replicato) per il backend
- Fare backup del database
- Fornire immagini pubbliche con password authentication disabilitata
 - Autenticazione solo via chiave SSH, con fingerprint della chiave loggato
- Aggiornare regolarmente le immagini pubbliche
 - In particolare quando ci sono vulnerability fix

Riferimenti

- <https://docs.openstack.org/glance/latest/>
- <https://docs.ceph.com/en/latest/rbd/rbd-openstack/>