

Storage in OpenStack

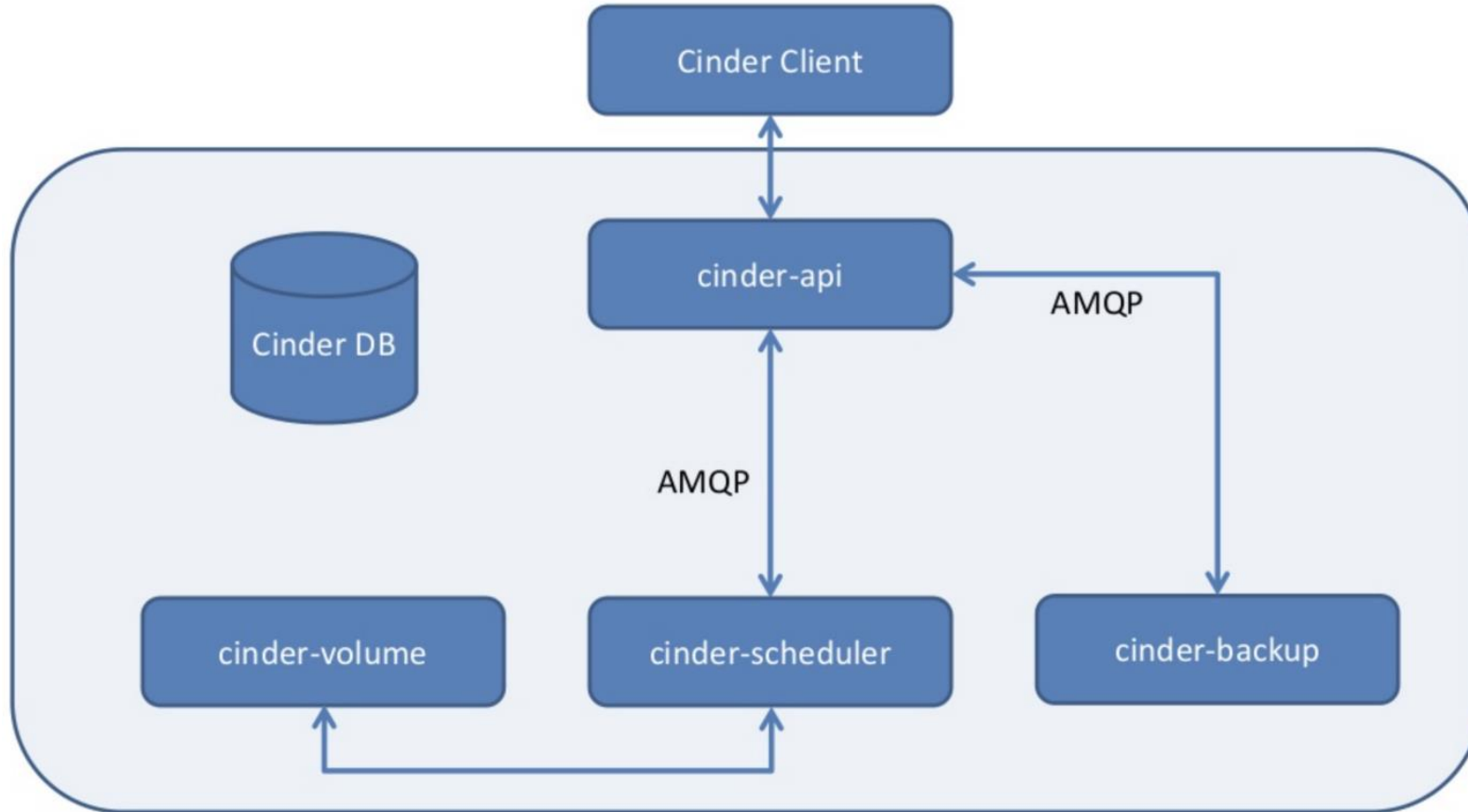
Servizio Openstack	Descrizione
nova	Block storage effimero associato a una istanza. Viene distrutto quando l'istanza viene cancellata
cinder	Block storage persistente
swift	Storage persistente a oggetti

Cinder

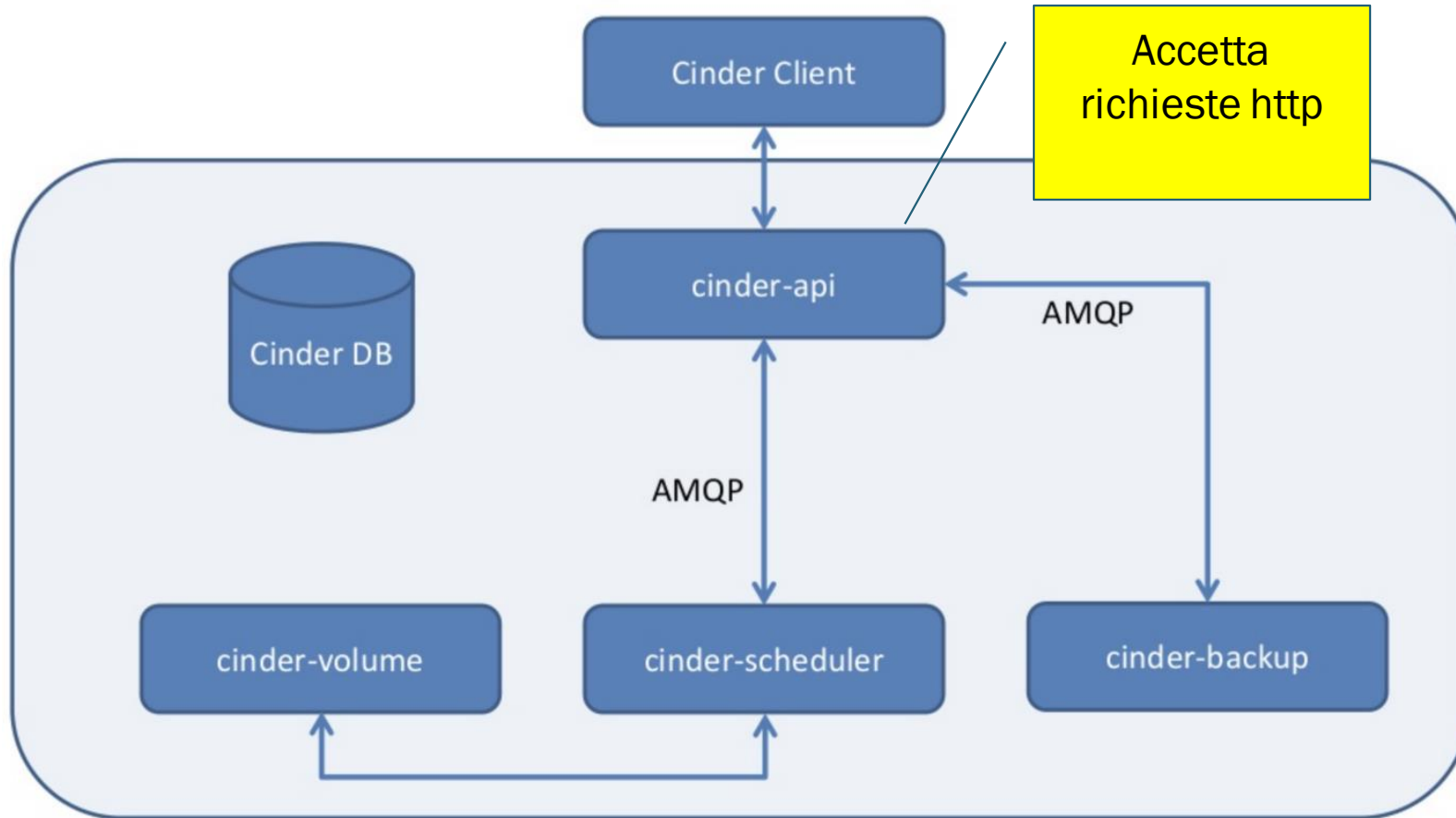
- Fornisce agli utenti un servizio di block storage persistente
 - Creazione/cancellazione di un volume
 - Attach/detach di un volume a/da una istanza
 - Estensione di un volume
 - Backup di un volume
 - Snapshot di un volume
- Un volume cinder non è legato a una specifica istanza
- Persiste nel caso in cui l'istanza a cui è attaccato viene cancellata



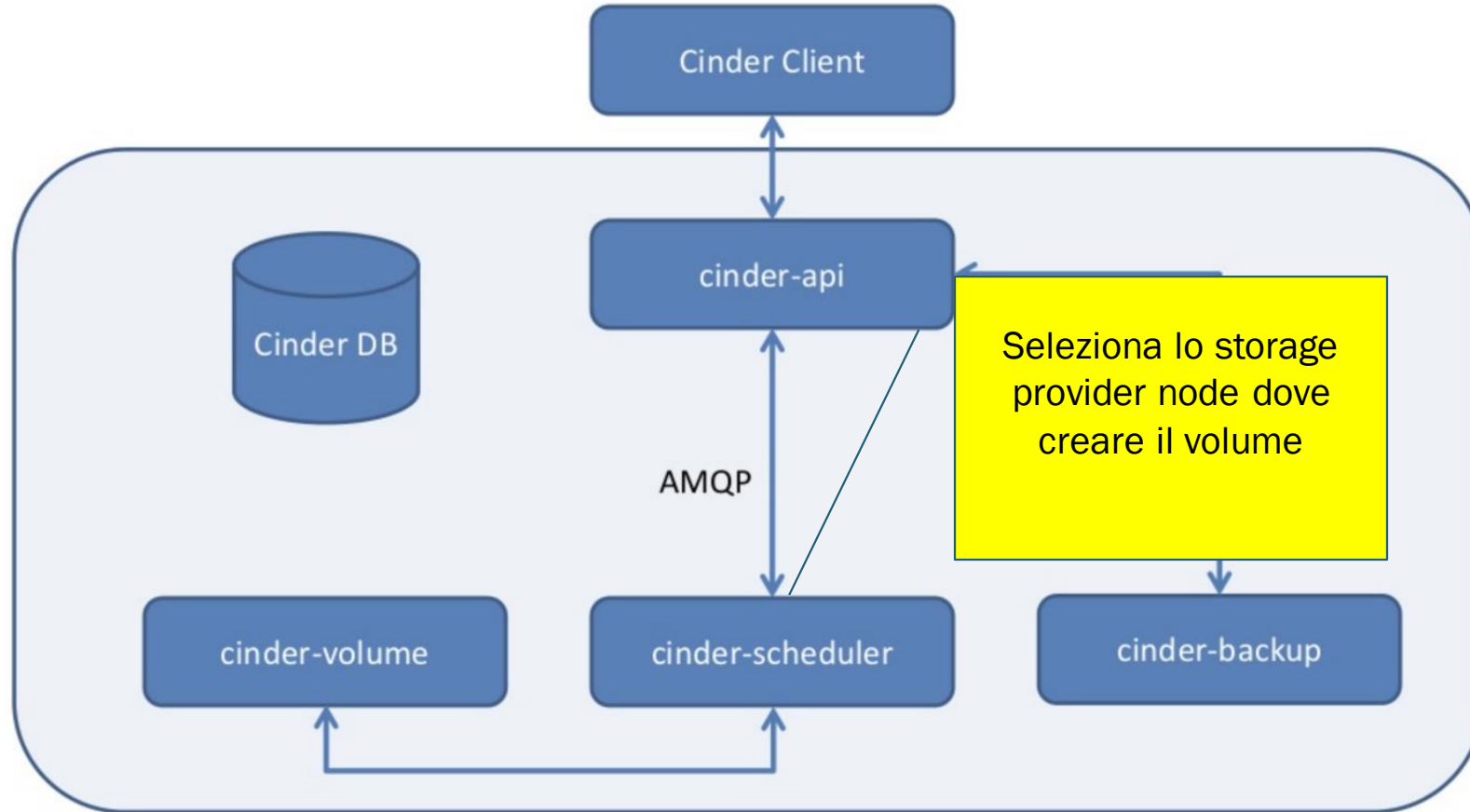
Cinder Architecture



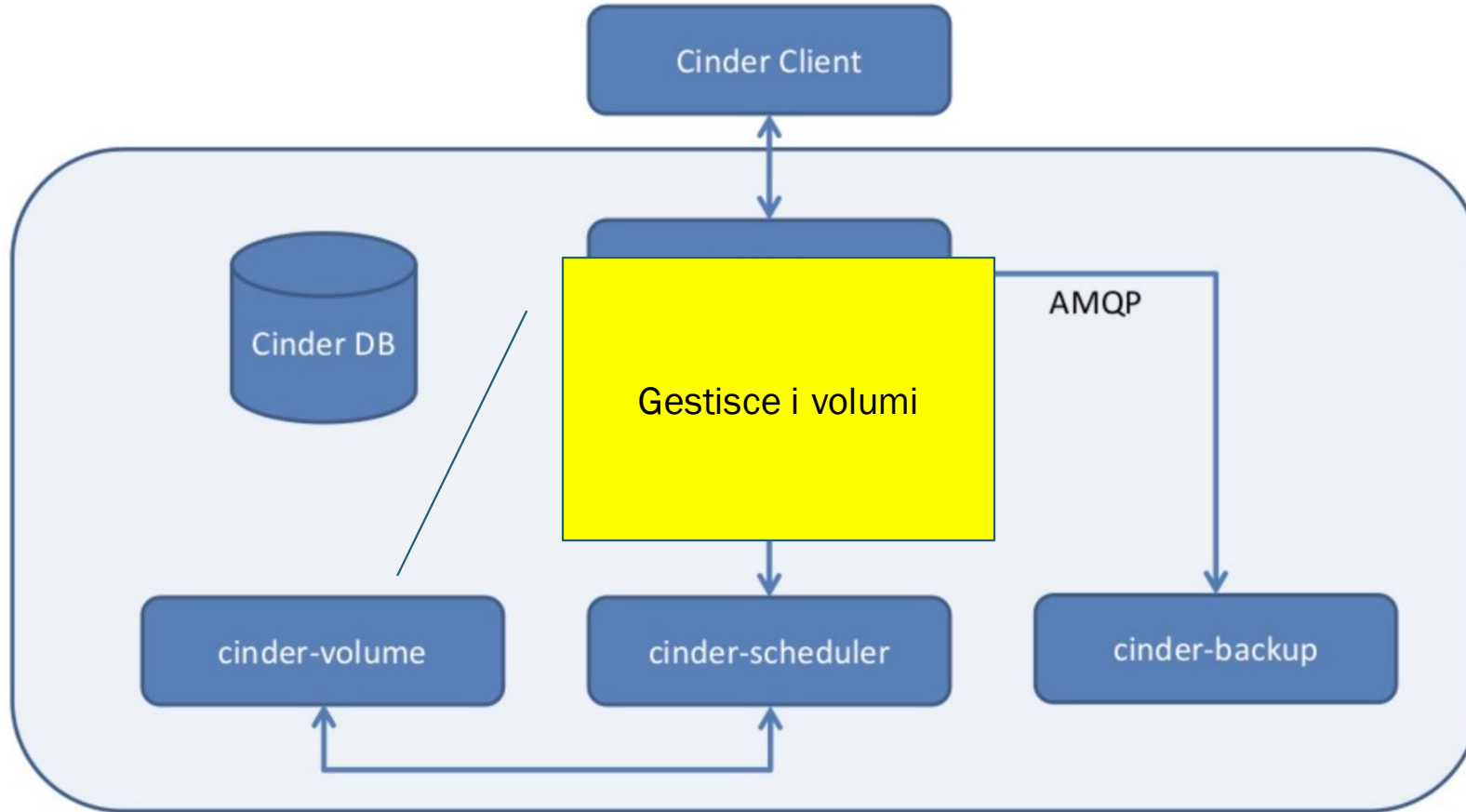
Cinder Architecture



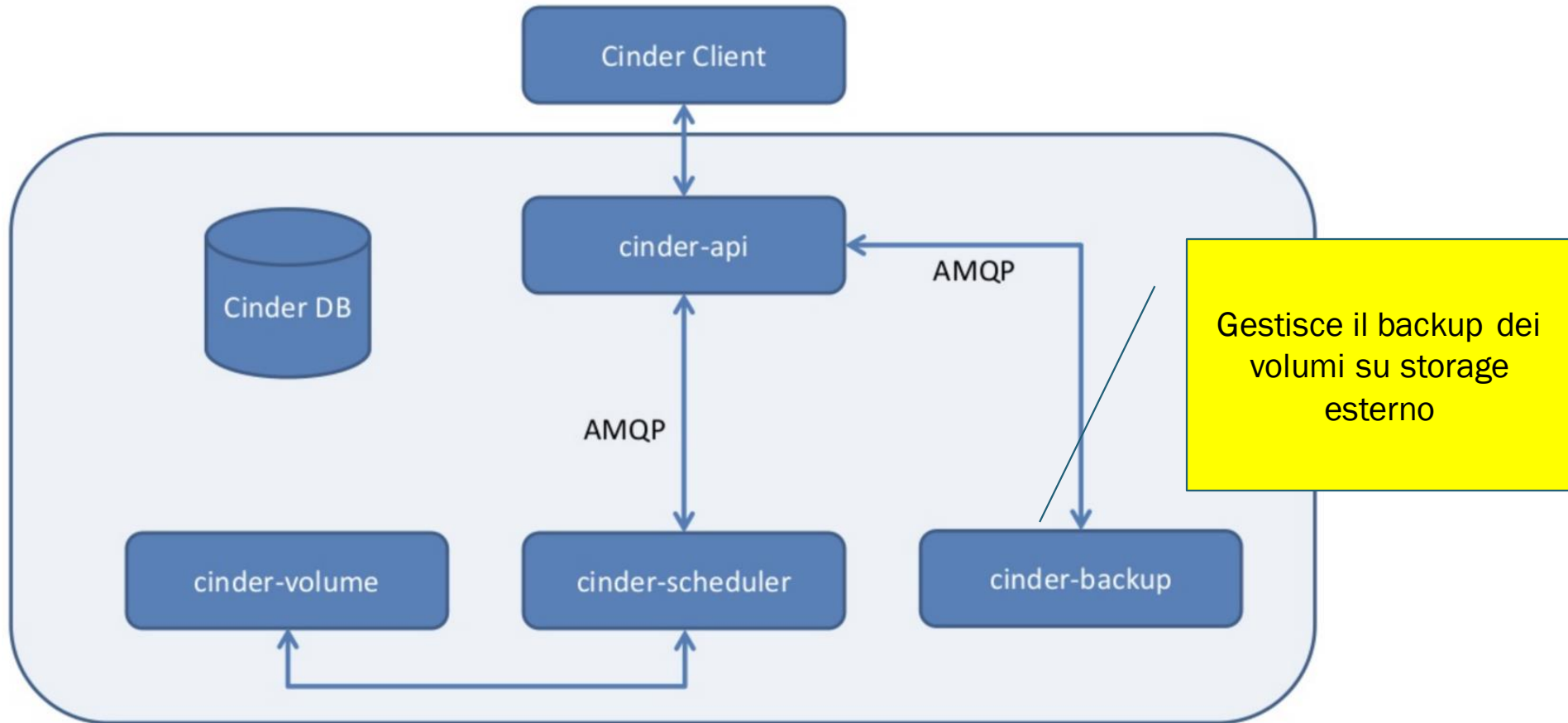
Cinder Architecture



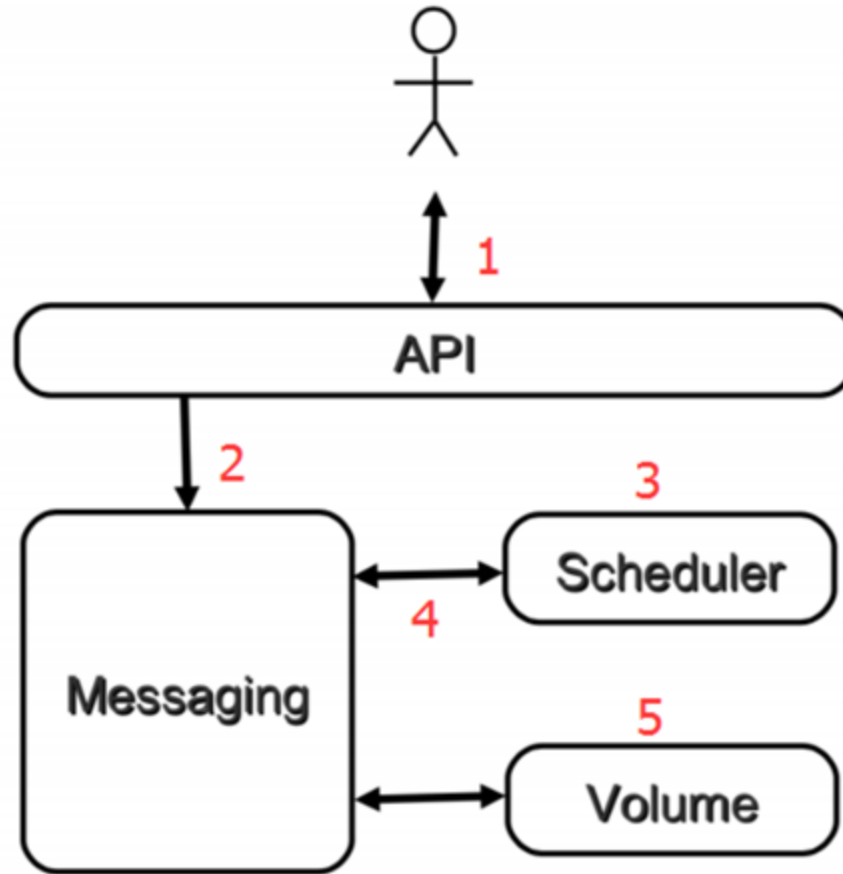
Cinder Architecture



Cinder Architecture



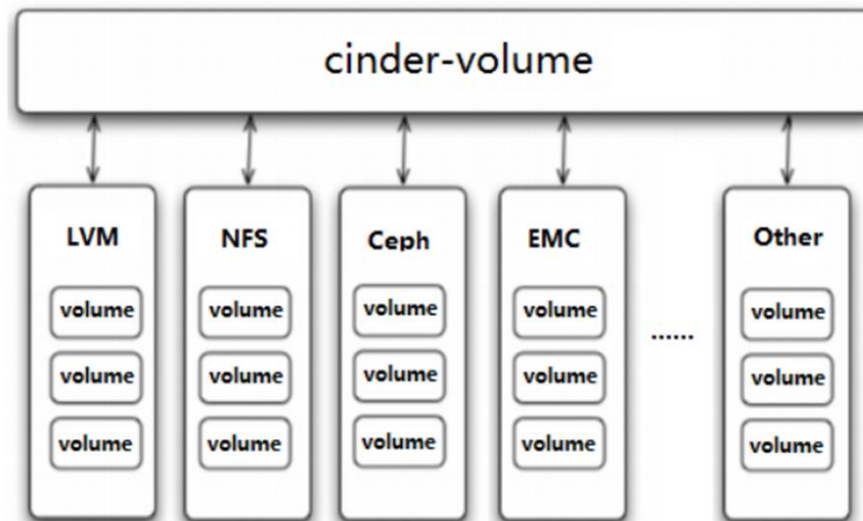
Creazione di un volume



Storage backends

- L'architettura di Cinder prevede la possibilità di integrare diverse implementazioni di storage
 - Interfaccia ben definita che permette agli hardware provider di implementare i necessari cinder driver
- Cinder supporta al momento circa 60 storage backends [*]
- E` possibile usare più backend contemporaneamente

[*] <https://docs.openstack.org/cinder/latest/reference/support-matrix.html>



Cinder backup

- Backend supportati:
 - Ceph
 - GlusterFS
 - NFS
 - Posix file system
 - Swift
 - S3
 - Google cloud storage

Bootable volumes

- Un volume cinder può essere usato anche come root disk di una VM
- A cosa serve ?
 - Per poter cancellare una VM, senza perderne i dati
 - Reliability: spesso lo storage persistente è più affidabile di quello effimero
 - Availability: è in genere più semplice live-migrare una VM con bootable disk da un hypervisor a un altro wrt VM che usano disco effimero

Select Boot Source

Image

Volume Size (GB) *

8

Create New Volume

Yes No

Delete Volume on Instance Delete

Yes No

Snapshot

- Cinder gestisce anche snapshot di volumi
 - La "fotografia" di un volume in un preciso istante
- Più un sistema per clonare volumi (da uno snapshot è possibile creare nuovi volumi) che non per backup
 - Visto che si usa lo stesso backend del volume
 - Per backup è preferibile usare la funzionalità offerta da cinder-backup

Deployment di cinder

- cinder-api e cinder-scheduler su controller node
- cinder-volume su storage node
 - Può anche coincidere con il controller node (dipende dal tipo di storage)
- cinder-backup non è strettamente obbligatorio
 - Si configura e si attiva solo se interessa la funzionalità di backup dei volumi

Installazione: operazioni sul controller node

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

Creazione
database

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

Installazione: operazioni sul controller node

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

Creazione
utente

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

Installazione: operazioni sul controller node (cont.ed)



```
$ openstack service create --name cinderv3 \  
--description "OpenStack Block Storage" volumev3  
  
$ openstack endpoint create --region RegionOne \  
volumev3 public http://controller:8776/v3/%\\$\(project\_id\)s  
$ openstack endpoint create --region RegionOne \  
volumev3 internal http://controller:8776/v3/%\\$\(project\_id\)s  
$ openstack endpoint create --region RegionOne \  
volumev3 admin http://controller:8776/v3/%\\$\(project\_id\)s
```

Creazione
servizio e
endpoint

```
$ yum install openstack-cinder
```


Installazione: operazioni sul controller node (cont.ed)



```
$ openstack service create --name cinderv3 \  
  --description "OpenStack Block Storage" volumev3  
  
$ openstack endpoint create --region RegionOne \  
  volumev3 public http://controller:8776/v3/%\\$\(project\_id\)s  
$ openstack endpoint create --region RegionOne \  
  volumev3 internal http://controller:8776/v3/%\\$\(project\_id\)s  
$ openstack endpoint create --region RegionOne \  
  volumev3 admin http://controller:8776/v3/%\\$\(project\_id\)s
```

```
$ yum install openstack-cinder
```

Installazione
pacchetti

Installazione: operazioni sul controller node (cont.ed)



/etc/cinder/cinder.conf

```
[DEFAULT]
# ...
auth_strategy = keystone
transport_url = rabbit://openstack:RABBIT_PASS@controller
my_ip = MANAGEMENT_INTERFACE_IP_ADDRESS

[database]
# ...
connection = mysql+pymysql://cinder:CINDER_DBPASS@controller/cinder

[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 = cinder
password = CINDER_PASS

[oslo_concurrency]
# ...
lock_path = /var/lib/cinder/tmp
```

Installazione: operazioni sul controller node (cont.ed)



Popolo il
database

```
$ su -s /bin/sh -c "cinder-manage db sync" cinder
```

```
$ systemctl enable openstack-cinder-api.service openstack-cinder-scheduler.service  
$ systemctl start openstack-cinder-api.service openstack-cinder-scheduler.service
```

Installazione: operazioni sul controller node (cont.ed)



```
$ su -s /bin/sh -c "cinder-manage db sync" cinder
```

Start dei servizi
api e scheduler

```
$ systemctl enable openstack-cinder-api.service openstack-cinder-scheduler.service  
$ systemctl start openstack-cinder-api.service openstack-cinder-scheduler.service
```

Installazione: operazioni sullo storage

Installazione
pacchetti

```
$ yum install openstack-cinder
```

/etc/cinder/cinder.conf

```
[DEFAULT]
# ...
auth_strategy = keystone
transport_url = rabbit://openstack:RABBIT_PASS@controller
my_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
glance_api_servers = http://controller:9292

[database]
# ...
connection = mysql+pymysql://cinder:CINDER_DBPASS@controller/cinder

[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 = cinder
password = CINDER_PASS

[oslo_concurrency]
# ...
lock_path = /var/lib/cinder/tmp
```

Installazione: operazioni sullo storage node

```
$ yum install openstack-cinder
```

Configurazione

```
/etc/cinder/cinder.conf
```

```
[DEFAULT]
# ...
auth_strategy = keystone
transport_url = rabbit://openstack:RABBIT_PASS@controller
my_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
glance_api_servers = http://controller:9292

[database]
# ...
connection = mysql+pymysql://cinder:CINDER_DBPASS@controller/cinder

[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 = cinder
password = CINDER_PASS

[oslo_concurrency]
# ...
lock_path = /var/lib/cinder/tmp
```

Installazione: operazioni sullo storage node: LVM

```
$ pvcreate /dev/sdb  
$ vgcreate cinder-volumes /dev/sdb
```

Creazione
volume group
LVM

/etc/cinder/cinder.conf

```
[DEFAULT]  
# ...  
enabled_backends = lvm  
  
[lvm]  
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver  
volume_group = cinder-volumes  
target_protocol = iscsi  
target_helper = lioadm
```

```
$ systemctl enable openstack-cinder-volume.service  
$ systemctl start openstack-cinder-volume.service
```

Installazione: operazioni sullo storage node: LVM

```
$ pvcreate /dev/sdb  
$ vgcreate cinder-volumes /dev/sdb
```

/etc/cinder/cinder.conf

```
[DEFAULT]  
# ...  
enabled_backends = lvm  
  
[lvm]  
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver  
volume_group = cinder-volumes  
target_protocol = iscsi  
target_helper = lioadm
```

Configurazione

```
$ systemctl enable openstack-cinder-volume.service  
$ systemctl start openstack-cinder-volume.service
```


Installazione: operazioni sullo storage node: LVM

```
$ pvcreate /dev/sdb  
$ vgcreate cinder-volumes /dev/sdb
```

/etc/cinder/cinder.conf

```
[DEFAULT]  
# ...  
enabled_backends = lvm  
  
[lvm]  
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver  
volume_group = cinder-volumes  
target_protocol = iscsi  
target_helper = lioadm
```

```
$ systemctl enable openstack-cinder-volume.service  
$ systemctl start openstack-cinder-volume.service
```

Start servizio
cinder-volume

Integrazione di ceph come backend

Creazione pool in
ceph e setting
del relativo
utente

```
$ceph osd pool create volumes
```

```
$ rbd pool init volumes
```

```
$ceph auth get-or-create client.cinder mon 'profile rbd' osd 'profile rbd pool=volumes, mgr 'profile rbd  
pool=volumes'
```

```
$ ceph auth get-or-create client.cinder | ssh {cinder-volume-server} sudo tee /etc/ceph/ceph.client.cinder.keyring  
ssh {cinder-volume-server} sudo chown cinder:cinder /etc/ceph/ceph.client.cinder.keyring
```

- Il secret key di client.cinder va configurato in libvirt dei compute node (necessario per operazioni di attach)

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

Installazione: operazioni sullo storage node: ceph

```
$ yum install ceph-common
```

Installazione
pacchetti

```
/etc/cinder/cinder.conf
```

```
[DEFAULT]
# ...
enabled_backends = ceph

[ceph]
[ceph]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
volume_backend_name = ceph
rbd_pool = volumes
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot = false
rbd_max_clone_depth = 5
rbd_store_chunk_size = 4
rados_connect_timeout = -1
rbd_user = cinder
rbd_secret_uuid = 457eb676-33da-42ec-9a8c-9293d545c337
```

```
$ systemctl enable openstack-cinder-volume.service
$ systemctl start openstack-cinder-volume.service
```

Installazione: operazioni sullo storage node: ceph

```
$ yum install ceph-common
```

```
/etc/cinder/cinder.conf
```

```
[DEFAULT]
# ...
enabled_backends = ceph

[ceph]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
volume_backend_name = ceph
rbd_pool = volumes
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot = false
rbd_max_clone_depth = 5
rbd_store_chunk_size = 4
rados_connect_timeout = -1
rbd_user = cinder
rbd_secret_uuid = 457eb676-33da-42ec-9a8c-9293d545c337
```

Configurazione

```
$ systemctl enable openstack-cinder-volume.service
$ systemctl start openstack-cinder-volume.service
```

Installazione: operazioni sullo storage node: ceph

```
$ yum install ceph-common
```

```
/etc/cinder/cinder.conf
```

```
[DEFAULT]
# ...
enabled_backends = ceph

[ceph]
[ceph]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
volume_backend_name = ceph
rbd_pool = volumes
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot = false
rbd_max_clone_depth = 5
rbd_store_chunk_size = 4
rados_connect_timeout = -1
rbd_user = cinder
rbd_secret_uuid = 457eb676-33da-42ec-9a8c-9293d545c337
```

```
$ systemctl enable openstack-cinder-volume.service
$ systemctl start openstack-cinder-volume.service
```

Start del servizio
cinder-volume

Cinder backup (configurato usando ceph come backend)

/etc/cinder/cinder.conf

Configurazione

```
[DEFAULT]
...
backup_driver = cinder.backup.drivers.ceph.CephBackupDriver
backup_ceph_conf=/etc/ceph/ceph.conf
backup_ceph_user = cinder-backup
backup_ceph_chunk_size = 134217728
backup_ceph_pool = backups
backup_ceph_stripe_unit = 0
backup_ceph_stripe_count = 0
```

```
$ systemctl enable openstack-cinder-backup.service
$ systemctl start openstack-cinder-backup.service
```

Cinder backup (configurato usando ceph come backend)

/etc/cinder/cinder.conf

```
[DEFAULT]
...
backup_driver = cinder.backup.drivers.ceph.CephBackupDriver
backup_ceph_conf=/etc/ceph/ceph.conf
backup_ceph_user = cinder-backup
backup_ceph_chunk_size = 134217728
backup_ceph_pool = backups
backup_ceph_stripe_unit = 0
backup_ceph_stripe_count = 0
```

Start del servizio
cinder-backup

```
$ systemctl enable openstack-cinder-backup.service
$ systemctl start openstack-cinder-backup.service
```

Verifica

```
[root@oa101-ms-ctrl ~(keystone_admin)]# openstack volume service list
```

Binary	Host	Zone	Status	State	Updated At
cinder-scheduler	oa101-ms-ctrl.novalocal	nova	enabled	up	2021-11-27T06:32:42.000000
cinder-volume	oa101-ms-ctrl.novalocal@lvm	nova	enabled	up	2021-11-27T06:32:40.000000
cinder-backup	oa101-ms-ctrl.novalocal	nova	enabled	down	2021-11-25T14:56:55.000000

```
[root@oa101-ms-ctrl ~(keystone_admin)]#
```


Come si interagisce con Cinder

- REST API
- Horizon (web based Dashboard)
- Openstack CLI (*openstack volume ...*)
- Cinder CLI (*cinder ...*)
 - Alcune funzionalità avanzate sono disponibili solo attraverso questa CLI

Dashboard (funzioni amministrative)

The screenshot shows the OpenStack dashboard interface. The top navigation bar includes the 'cloudvms.io' logo and a user profile dropdown labeled 'admin'. The left sidebar contains a navigation menu with 'Admin' highlighted. The main content area is titled 'Volumes' and displays a table of storage volumes. The table has columns for Project, Host, Name, Size, Status, Group, Type, Attached To, Bootable, Encrypted, and Actions. Several volumes are listed, including 'allpix2_vol' (5GiB, Available), 'marah-vol' (50GiB, In-use), 'Fis03teorici' (25GiB, In-use), 'admin' (100GiB, In-use), 'Multi-tenant Services' (10GiB, In-use), and 'QST' (25GiB, In-use). The interface also features a search filter for 'Volume Name', a '+ Manage Volume' button, and a 'Delete Volumes' button.

<input type="checkbox"/>	Project	Host	Name	Size	Status	Group	Type	Attached To	Bootable	Encrypted	Actions
<input type="checkbox"/>	LUXE	cl-d-ctrl-02.cloud.pd.infn.it@ceph-ec#ceph-ec	allpix2_vol	5GiB	Available	-	ceph-ec		No	No	Delete Volume
<input type="checkbox"/>	AbinitioTransport	cl-d-ctrl-02.cloud.pd.infn.it@ceph-ec#ceph-ec	marah-vol	50GiB	In-use	-	ceph-ec	/dev/vdb on M-2020	No	No	Update Volume Status
<input type="checkbox"/>	Fis03teorici	cl-d-ctrl-02.cloud.pd.infn.it@ceph-ec#ceph-ec	71363b95-c3e1-421c-bfee-3964350a9839	25GiB	In-use	-	ceph-ec	/dev/vda on DIB LOCK_SC	Yes	No	Update Volume Status
<input type="checkbox"/>	admin	cl-d-ctrl-02.cloud.pd.infn.it@ceph-ec#ceph-ec	openvas	100GiB	In-use	-	ceph-ec	/dev/vda on scan s	Yes	No	Update Volume Status
<input type="checkbox"/>	Multi-tenant Services	cl-d-ctrl-02.cloud.pd.infn.it@ceph-ec#ceph-ec	openvas	10GiB	In-use	-	ceph-ec	/dev/vdb on test-ov	No	No	Update Volume Status
<input type="checkbox"/>	QST	cl-d-ctrl-02.cloud.pd.infn.it@ceph-ec#ceph-ec	rgbmr-c-vol-bak	25GiB	In-use	-	ceph-ec	/dev/vdc on rgb mrc-vm24	No	No	Update Volume Status

Multiple storage backends

- Cinder può essere configurato in modo da usare contemporaneamente più storage backends
- Con stesso driver o driver diversi

Multiple storage backend (esempio)

/etc/cinder/cinder.conf

```
[DEFAULT]
enabled_backends = ceph,lvm ←
...
[ceph]
volume_group = ceph
volume_backend_name = ceph
volume_driver = cinder.volume.drivers.rbd.RBDDriver
rbd_pool = volumes-prod
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot = false
rbd_store_chunk_size = 4
rados_connect_timeout = -1
rbd_max_clone_depth = 5
rbd_user = cinder-prod
rbd_secret_uuid = d56612fa-2951-471a-88c6-8b473794a987
rbd_exclusive_cinder_pool = true

[lvm]
volume_backend_name=lvm
volume_driver=cinder.volume.drivers.lvm.LVMVolumeDriver
target_ip_address=10.10.0.45
target_helper=lioadm
volume_group=cinder-volumes
volumes_dir=/var/lib/cinder/volumes
```

Volume type

- "Collezione" di capabilities usate per caratterizzare i volumi cinder, es:
 - il backend da usare
 - se usare l'encryption
 - QoS
- Solo l'admin può creare volume type
 - Può ad esempio definire un "premium" volume type per storage ad alte prestazioni, un "basic" volume type, ecc.
 - Volume type possono essere pubblici o esposti solo a specifici progetti
- L'utente seleziona il volume type quando crea un volume
- Lo scheduler sceglie lo storage che soddisfa le capabilities specificate nel volume type

Volume type

```
[root@cld-ctrl-01 ~]# openstack volume type list
+-----+-----+-----+
| ID                | Name          | Is Public |
+-----+-----+-----+
| eef42b10-83eb-4b6a-98c9-f45ac3eefe3f | __DEFAULT__  | False     |
| be70db12-68a3-4de0-8195-977d0eb425af | ceph-ec      | True      |
| 57f3ef77-dfb2-49bf-8bcd-58f7e38fbcf7 | ceph         | True      |
+-----+-----+-----+
[root@cld-ctrl-01 ~]# openstack volume type show ceph-ec
+-----+-----+
| Field              | Value          |
+-----+-----+
| access_project_ids | None           |
| description        | None           |
| id                 | be70db12-68a3-4de0-8195-977d0eb425af |
| is_public          | True           |
| name               | ceph-ec       |
| properties         | volume_backend_name='ceph-ec' ←
| qos_specs_id       | None           |
+-----+-----+
[root@cld-ctrl-01 ~]#
```

- Cinder permette di definire dei QoS e rate limit
- Come si definisce:
 1. Si crea un QoS
 2. Si associa il QoS a un volume type
 3. Si crea un volume usando quel volume type
- La cosa viene poi implementata attraverso libvirt

QoS: attributi supportati

For Fixed IOPS per volume

read_iops_sec

write_iops_sec

total_iops_sec

For Burst IOPS per volume

read_iops_sec_max

write_iops_sec_max

total_iops_sec_max

For Fixed bandwidth per volume

read_bytes_sec

write_bytes_sec

total_bytes_sec

For Burst bandwidth per volume

read_bytes_sec_max

write_bytes_sec_max

total_bytes_sec_max

For burst bucket size

size_iops_sec

QoS esempio:

```
[root@cld-ctrl-01 ~]# openstack volume qos create --consumer "front-end" \  
> --property "read_iops_sec=20000" \  
> --property "write_iops_sec=10000" \  
> high-iops  
+-----+  
| Field      | Value  
+-----+  
| consumer   | front-end  
| id         | 972fadce-52d8-406a-b461-18a2121185a7  
| name       | high-iops  
| properties | read_iops_sec='20000', write_iops_sec='10000'  
+-----+  
[root@cld-ctrl-01 ~]# openstack volume type create silver  
+-----+  
| Field      | Value  
+-----+  
| description | None  
| id         | 91e257b0-4990-4426-a737-289e921b412f  
| is_public  | True  
| name       | silver  
+-----+  
[root@cld-ctrl-01 ~]# openstack volume qos associate 972fadce-52d8-406a-b461-18a2121185a7 91e257b0-4990-4426-a737-289e921b412f  
[root@cld-ctrl-01 ~]# openstack volume type show 91e257b0-4990-4426-a737-289e921b412f  
+-----+  
| Field      | Value  
+-----+  
| access_project_ids | None  
| description        | None  
| id                 | 91e257b0-4990-4426-a737-289e921b412f  
| is_public          | True  
| name                | silver  
| properties          |  
| qos_specs_id       | 972fadce-52d8-406a-b461-18a2121185a7  
+-----+  
[root@cld-ctrl-01 ~]#
```

QoS esempio (cont.ed)

virsh dumpxml di una
istanza con attaccato
un volume di tipo
"silver"

```
</backingStore/>
</backingStore>
<target dev='vda' bus='virtio'/>
<alias name='virtio-disk0'/>
<address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</disk>
<disk type='network' device='disk'>
  <driver name='qemu' type='raw' cache='none' discard='unmap'/>
  <auth username='cinder-prod'>
    <secret type='ceph' uuid='c26612df-2951-4a1a-87c6-8b473794a9c1'/>
  </auth>
  <source protocol='rbd' name='volumes-prod/volume-d5e171cb-896c-4364-a8f5-51ae3720c3a3' index='3'>
    <host name='192.168.61.208' port='6789'/>
    <host name='192.168.61.207' port='6789'/>
    <host name='192.168.61.206' port='6789'/>
  </source>
  <target dev='vdb' bus='virtio'/>
  <iotune>
    <read_iops_sec>20000</read_iops_sec>
    <write_iops_sec>10000</write_iops_sec>
  </iotune>
  <serial>d5e171cb-896c-4364-a8f5-51ae3720c3a3</serial>
  <alias name='virtio-disk1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
</disk>
<controller type='luch' index='0' model='piix3-uhci'>
```

Encryption

- Come abilitare l'encryption dei volumi:
 1. Installare e configurare barbican (come key manager)
 2. Configuro cinder e nova per usare barbican
 3. Creo un volume type per volumi encrypted
 4. Creo volumi encrypted

Ref: <https://docs.openstack.org/cinder/latest/configuration/block-storage/volume-encryption.html>

Configurazione di nova e cinder per usare barbican



/etc/cinder/cinder.conf

```
[key_manager]  
backend = barbican
```

/etc/nova/nova.conf

```
[key_manager]  
backend = barbican
```

```
#systemctl restart openstack-nova-api  
#systemctl restart openstack-cinder-api
```

Encryption (cont.ed)

```
# openstack volume type create --encryption-provider luks \  
  --encryption-cipher aes-xts-plain64 --encryption-key-size 256 \  
  --encryption-control-location front-end LUKS
```

Creo un volume type

```
# openstack volume create --size 1 --type LUKS 'encrypted volume'
```

Encryption (cont.ed)

```
# openstack volume type create --encryption-provider luks \  
--encryption-cipher aes-xts-plain64 --encryption-key-size 256 \  
--encryption-control-location front-end LUKS
```

```
# openstack volume create --size 1 --type LUKS 'encrypted volume'
```

Creo un volume
specificando quel
volume type

Quota

- Cinder permette di definire quote per progetto
 - GB totali, numero massimo di volumi, max GB per volume, ...
- E` possibile definire quote complessive ma anche per singolo volume type
- Purtroppo le quote per volume type sono settabili e visibili solo via command line e non via dashboard

Setting quota via CLI

```
[root@cld-ctrl-01 ~]# cinder quota-update --gigabytes 3000 --volume-type ceph-ec d27fe2becea94a3e980fb9f66e2f291a
```

Property	Value
backup_gigabytes	1000
backups	10
gigabytes	3500
gigabytes__DEFAULT__	-1
gigabytes_ceph	500
gigabytes_ceph-ec	3000
groups	10
per_volume_gigabytes	1000
snapshots	10
snapshots__DEFAULT__	-1
snapshots_ceph	-1
snapshots_ceph-ec	-1
volumes	20
volumes__DEFAULT__	-1
volumes_ceph	-1
volumes_ceph-ec	-1

```
[root@cld-ctrl-01 ~]# cinder quota-usage d27fe2becea94a3e980fb9f66e2f291a
```

Type	In_use	Reserved	Limit	Allocated
backup_gigabytes	0	0	1000	0
backups	0	0	10	0
gigabytes	2478	0	3500	0
gigabytes__DEFAULT__	0	0	-1	0
gigabytes_ceph	228	0	500	0
gigabytes_ceph-ec	2250	0	3000	0
groups	0	0	10	0
per_volume_gigabytes	0	0	1000	0
snapshots	0	0	10	0
snapshots__DEFAULT__	0	0	-1	0
snapshots_ceph	0	0	-1	0
snapshots_ceph-ec	0	0	-1	0
volumes	17	0	20	0
volumes__DEFAULT__	0	0	-1	0
volumes_ceph	4	0	-1	0
volumes_ceph-ec	13	0	-1	0

```
[root@cld-ctrl-01 ~]#
```

d27..91a è l'ID del progetto considerato

Quote per volume type

Create Volume

Volume Name

Description

Volume Source
No source, empty volume

Type
ceph
ceph-ec
ceph

Availability Zone
nova

Group
No group

Description:
Volumes are block devices that can be attached to instances.

Volume Type Description:
ceph
No description available.

Volume Limits

Total Gibibytes 35,705 of 55,000 GiB Used

Number of Volumes 36 of 50 Used

Cancel Create Volume

Anche l'utente via dashboard vede solo la quota complessiva e non per i diversi volume type

Quote per vo

L'utente ha la possibilità di provare a creare un volume su un volume type dove non ha quota. L'operazione ovviamente fallisce

Error: Unable to create volume. ✕

```
[root@cld-ctrl-01 ~]# grep "Quota excee" /var/log/cinder/api.log
2021-11-27 08:32:07.806 5924 WARNING cinder.quota_utils [req-7acceaf5-1d2f-4e00-b587-a0fde1108ea1 e237e43716fb490db5bda4b777835669 0a6d806e32204512b189b1f4d0bda4cf - default default] Quota exceeded for 0a6d806e32204512b189b1f4d0bda4cf, tried to create 100G volume (26680G of 26680G already consumed).: cinder.exception.OverQuota: Quota exceeded for resources: ['gigabytes_ceph']
[root@cld-ctrl-01 ~]#
```

Volume migration

- E` possibile migrare un volume da un backend a un altro (con volume type "compatibile")

```
[root@cld-ctrl-01 ~]# cinder get-pools
+-----+-----+
| Property | Value |
+-----+-----+
| name     | cld-ctrl-02.cloud.pd.infn.it@ceph#ceph |
+-----+-----+
+-----+-----+
| Property | Value |
+-----+-----+
| name     | cld-ctrl-02.cloud.pd.infn.it@ceph-ec#ceph-ec |
+-----+-----+
+-----+-----+
| Property | Value |
+-----+-----+
| name     | cld-ctrl-01.cloud.pd.infn.it@ceph#ceph |
+-----+-----+
+-----+-----+
| Property | Value |
+-----+-----+
| name     | cld-ctrl-01.cloud.pd.infn.it@ceph-ec#ceph-ec |
+-----+-----+
[root@cld-ctrl-01 ~]#
```



Elenco dei backends

Volume migration (cont.ed)

```
[root@cld-ctrl-01 ~]# openstack volume show 75ba2f45-5f2f-4c98-b011-a2a4f0848e66
```

Field	Value
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	None
created_at	2021-11-27T08:48:36.000000
description	
encrypted	False
id	75ba2f45-5f2f-4c98-b011-a2a4f0848e66
migration_status	None
multiattach	False
name	test-volume
os-vol-host-attr:host	cld-ctrl-02.cloud.pd.infn.it@ceph-ec#ceph-ec
os-vol-mig-status-attr:migstat	None
os-vol-mig-status-attr:name_id	None
os-vol-tenant-attr:tenant_id	b38a0dab349e42bdbb469274b20a91b4
properties	
replication_status	None
size	10
snapshot_id	None
source_volid	None
status	available
type	ceph-ec
updated_at	2021-11-27T08:48:36.000000
user_id	e237e43716fb490db5bda4b777835669

```
[root@cld-ctrl-01 ~]# openstack volume migrate --host cld-ctrl-01.cloud.pd.infn.it@ceph-ec#ceph-ec 75ba2f45-5f2f-4c98-b011-a2a4f0848e66
```

```
[root@cld-ctrl-01 ~]#
```

Volume migration (cont.ed)

```
[root@cld-ctrl-01 ~]# openstack volume show 75ba2f45-5f2f-4c98-b011-a2a4f0848e66
```

Field	Value
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	None
created_at	2021-11-27T08:48:36.000000
description	
encrypted	False
id	75ba2f45-5f2f-4c98-b011-a2a4f0848e66
migration_status	success
multiattach	False
name	test volume
os-vol-host-attr:host	cld-ctrl-01.cloud.pd.infn.it@ceph-ec#ceph-ec
os-vol-mig-status-attr:migstat	success
os-vol-mig-status-attr:name_id	None
os-vol-tenant-attr:tenant_id	b38a0dab349e42bdbb469274b20a91b4
properties	
replication_status	None
size	10
snapshot_id	None
source_vol_id	None
status	available
type	ceph-ec
updated_at	2021-11-27T08:53:42.000000
user_id	e237e43716fb490db5bda4b777835669

```
[root@cld-ctrl-01 ~]#
```

Use case: dismissione di un backend



1. Rendo privato e non visibile a tutti i progetti il volume type associato a questo backend (o azzero la quota)
 - In modo che non vengano più creati nuovi volumi su quel backend
2. Migrazione dei volumi che hanno volume type associato a questo backend a un nuovo volume type (associato a un backend diverso)
Due opzioni possibili:
 1. creazione nuovi volumi (dove copio il contenuto di quelli da migrare) e cancellazione dei vecchi volumi
 2. volume retype
3. Cancellazione del vecchio volume type
4. Cancellazione (disabilitandolo dalla configurazione di cinder) del vecchio backend

Volume retype

- Migrazione di un volume da un volume type a un altro

```
# openstack volume set --type <new-type> --retype-policy on-demand <vol-id>
```

- Esperienza personale: il volume retype a volte può dare qualche problema
 - Es. In CloudVeneto lo abbiamo usato per migrare alcuni volumi da un backend gestito dal driver DELL-equallogic a un backend ceph
 - Dopo l'operazione non si riusciva più a fare il mount ("bad geometry: block count exceeds size of device")
 - Risolto allargando un po' il volume

Multi attach

- Alcuni storage backend permettono a un volume di essere attaccato a più istanze contemporaneamente

```
$ cinder type-create multiattach  
$ cinder type-key multiattach set multiattach="<is> True"
```

- Un volume creato usando questo volume type potrà essere attaccato contemporaneamente a più istanze
- **Va usato uno shared-disk file system !!!**

Policies

- Per default non c'è isolamento tra utenti appartenenti allo stesso progetto
 - Posso cancellare il volume di un altro utente
 - Posso attaccare il volume di un altro utente a una mia VM e quindi leggerne i dati
- E` possibile cambiare questo comportamento di default modificando il file `/etc/cinder/policy.yaml`

Ref: <https://docs.openstack.org/cinder/latest/configuration/block-storage/policy.html>

Transfer di ownership di un volume

- Cinder permette il transfer di ownership un volume da un utente a un altro (anche di un progetto diverso)

+ Create Volume ⇌ Accept Transfer 🗑 Delete Volumes

Zone	Bootable	Encrypted	Actions
	Yes	No	Edit Volume

- Extend Volume
- Launch as Instance
- Manage Attachments
- Create Snapshot
- Change Volume Type
- Upload to Image
- Create Transfer**
- Delete Volume

Create Volume Transfer

Transfer Name ^{*}

Description:
Ownership of a volume can be transferred from one project to another. Once a volume transfer is created in a donor project, it then can be "accepted" by a recipient project. This is equivalent to the `cinder transfer-create` command.

Transfer Name

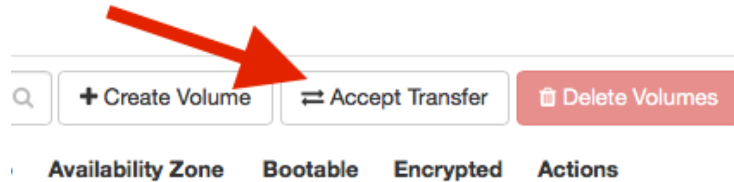
Transfer ID

Authorization Key

Description:
The Transfer ID and the Authorization Key are needed by the recipient in order to accept the transfer. Please capture both the Transfer ID and the Authorization Key and provide them to your transfer recipient.

The Authorization Key will not be available after closing this page, so you must capture it now or download it, or else you will be unable to use the transfer.

Transfer di ownership di un volume (cont.ed)



Accept Volume Transfer ✕

Transfer ID *

Authorization Key *

Description:

Ownership of a volume can be transferred from one project to another. Accepting a transfer requires obtaining the Transfer ID and Authorization Key from the donor. This is equivalent to the `cinder transfer-accept` command.

Backup

```
$ openstack volume backup create [--incremental] [--force] <VOLUME_ID>
```

```
$ openstack volume backup restore <BACKUP_ID> <VOLUME_ID>
```

Ref: <https://docs.openstack.org/cinder/latest/admin/blockstorage-volume-backups.html>

Troubleshooting

- Cinder log files
 - /var/log/cinder/api.log
 - /var/log/cinder/scheduler.log
 - /var/log/cinder/volume.log
 - /var/log/cinder/backup.log
- Compute node Nova log file
 - /var/log/nova/nova-compute.log
 - Per debuggare operazioni relative a attach/detach di un volume a/da una istanza

References

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