

# GPFS backend per OpenStack

Alessandro Brunengo

Mirko Corosu

INFN-Genova

# Contenuto

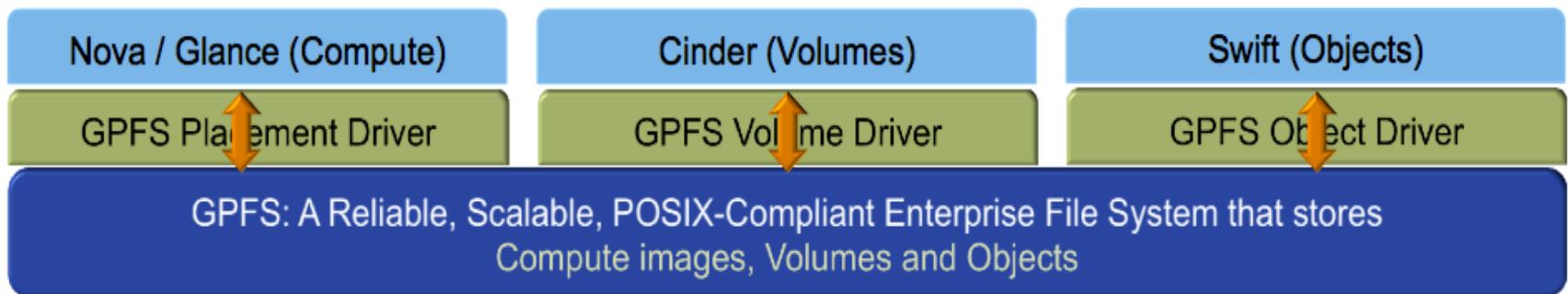
---

- o Overview
- o GPFS backend per Glance
- o GPFS backend per Cinder
- o GPFS backend per Nova

# Overview

---

# Introduzione



**Glance, Cinder e Nova** di OpenStack

- Cinder e' strutturato in modo da poter usare un **pluggable driver** specifico per uno storage di backend
- OpenStack include un driver per GPFS dalla release Kilo
- Il driver e' realizzato per permettere a Cinder di sfruttare tutte le **feature di GPFS** per l'affidabilita', la flessibilita' e le performance

# Features: overview

---

- Usando GPFS come backend si possono sfruttare le features:
  - Live migration
  - Snapshot di volumi
  - Supporto per consistency groups
  - Shared storage per VM, in un unico sistema di storage (tramite protocol access, o remote cluster access)
  - Tiering
- Usando **lo stesso independent fileset** per Glance, Cinder e Nova backend, si ottiene
  - Deploy istantaneo e uso efficiente di disco per bootable volume da immagine (cloning)
  - Deploy istantaneo di volumi da volumi (cloning)

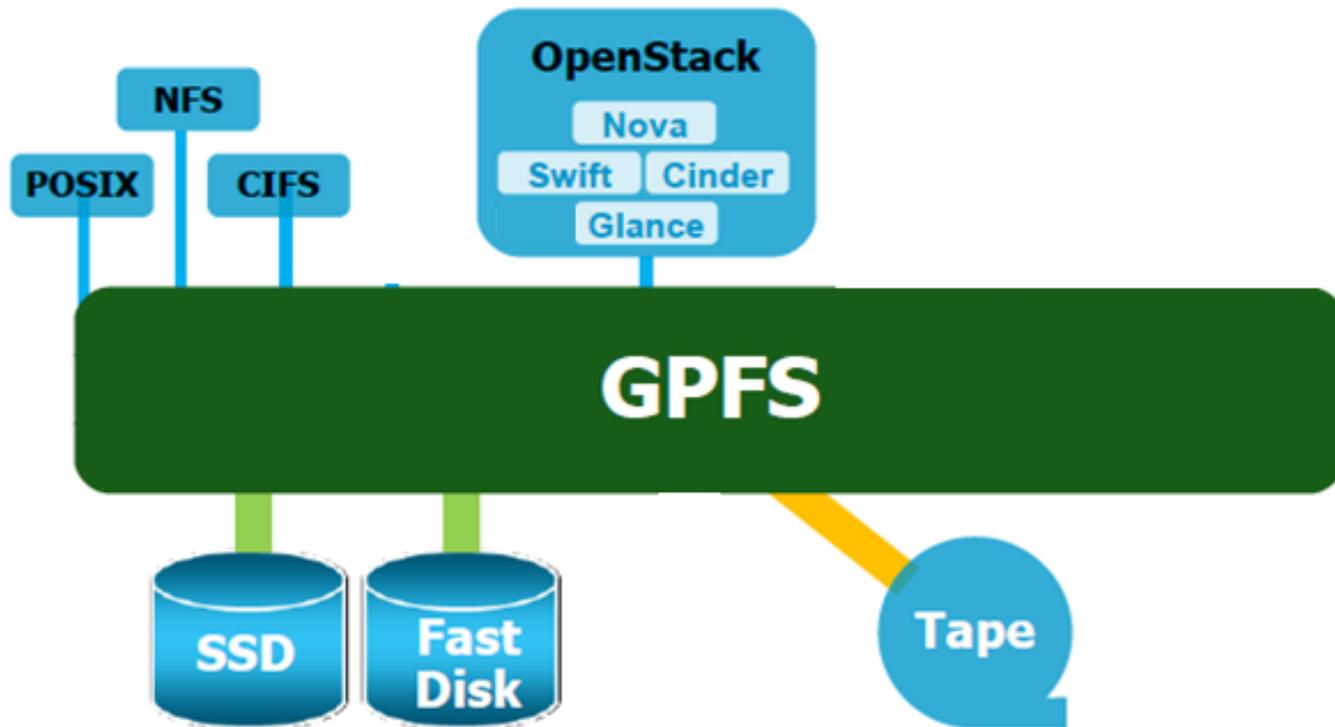
# Features: overview

<b>Volume and Image (Cinder and Glance)</b>	
<b>Volume Creation and Deletion</b>	<b>Yes</b>
<b>Volume Attach and Detach to a VM instance</b>	<b>Yes</b>
<b>Volume Snapshot management</b>	<b>Yes</b>
<b>Volume Creation from Snapshot</b>	<b>Yes</b>
<b>Efficient cloning of Volumes</b>	<b>Yes</b>
<b>Extend Volumes</b>	<b>Yes</b>
<b>Copy Image to Volume and Copy Volume to Image</b>	<b>Yes</b>
<b>Instantaneous Boot Volume Create From Glance Repo</b>	<b>Yes</b>
<b>Live migration of Instances</b>	<b>Yes</b>
<b>Backup of Volumes</b>	<b>Yes</b>
<b>Sharing of Data between instances (with file system support)</b>	<b>Yes</b>
<b>Quality of service that uses multitier storage (with Flash support)</b>	<b>Yes</b>
<b>Encryption of Volumes Yes</b>	<b>Yes</b>
<b>Tiering of Volumes Yes</b>	<b>Yes</b>
<b>Compression of Volumes Yes</b>	<b>Yes</b>

<b>Object Store Feature (Swift)</b>	
<b>In-Place Analytics with Hadoop compatibility</b>	<b>Yes</b>
<b>High Performance and High Availability</b>	<b>Yes</b>
<b>Object Compression</b>	<b>Yes</b>
<b>Object Encryption</b>	<b>Yes</b>
<b>Multi-Region Support</b>	<b>Yes</b>
<b>WAN caching with Active File Management (AFM)</b>	<b>Yes</b>
<b>Policy-based information lifecycle management</b>	<b>Yes</b>
<b>Easy installation, configuration, and management</b>	<b>Yes</b>
<b>Integrated monitoring</b>	<b>Yes</b>
<b>Swift and S3 API Support</b>	<b>Yes</b>
<b>Large Object Support (5 TB)</b>	<b>Yes</b>
<b>Support for OpenStack Swift object store features</b>	<b>Yes</b>
<b>Unified File and Object support</b>	<b>Yes</b>

# Features: overview

---



# Preconfigurazione

---

- Al fine di utilizzare GPFS come backend, i nodi che esportano i servizi Glance, Cinder e Nova **devono essere configurati come client** nel cluster GPFS
  - In questo modo avranno accesso al file system in modo nativo, e potranno sfruttarne le funzionalità
- E' opportuno **dedicare un independent fileset** per ospitare i dati di tutti e tre servizi OpenStack:

```
# mmcrfileset gpfs_dev openstack_fs -inode-space new
# mmlinkfileset gpfs_dev openstack_fs -J /gpfs/openstack
# chmod 755 /gpfs/openstack
```
- Questi comandi possono essere eseguiti dall'account root di un qualunque nodo del cluster GPFS

# GPFS backend per Glance

---

# Glance: configurazione

---

- o Creazione di un subtree per ospitare i volumi di Glance

```
# mkdir -p /gpfs/openstack/glance/images  
# chmod -R 755 /gpfs/openstack/glance
```

- o Assegnare permission ed ownership opportune

```
# chown -R <glance-user> /gpfs/openstack/glance
```

- o l'ownership corretta dipende dalla distribuzione di OpenStack installata: verificare l'ownership dello user sotto cui girano i processi di Glance

# Glance: configurazione (cont.)

---

- Modificare la sessione `[glance_store]` del file `/etc/glance/glance-api.conf`:

```
stores = file[...]  
default_store = file
```

- Se si vuole utilizzare **solo il backend GPFS**:

```
filesystem_store_datadir =  
/gpfs/openstack/glance/images
```

- Se si vuole **aggiungere** questo store ad altri:

```
filesystem_store_datadir =  
filesystem_store_datadirs =  
<prev_configured_dir>:100  
filesystem_store_datadirs =  
/gpfs/openstack/glance/images:200
```

- Far quindi ripartire il servizio `openstack-glance-api`

# GPFS backend per Cinder

---

# Cinder: il driver GPFS

---

- Il driver GPFS supporta:
  - Volume Create/Delete
  - Volume Attach/Detach
  - Snapshot Create/Delete
  - Consistency Group (su fileset dedicato)
  - Create Volume from Snapshot
  - Copy Image to Volume/Volume to Image
  - Clone Volume
  - Extend Volume
  - Volume Stats
- Le funzionalita' evidenziate vengono eseguite tramite la funzione di cloning (copy on write) di GPFS
  - provisioning quasi istantanea
  - ottimizzazione dello spazio occupato

# Cinder: parametri del driver

---

- o Parametri di configurazione del driver:
  - o `volume_driver` = `cinder.volume.drivers.ibm.gpfs.GPFSDriver`
  - o `gpfs_mount_point_base`: path della directory GPFS in cui Cinder mettera' i suoi dati (volumi/snapshot)
  - o `gpfs_image_dir`: path del repository di Glance su GPFS ("None" se Glance non e' configurato su GPFS)
  - o `gpfs_image_share_mode`: `copy` (se si vuole creare copia da immagine) o `copy_on_write` (se si vuole usare cloning)
    - o solo se `gpfs_image_dir` e' nello stesso fileset di `gpfs_mount_point_base`
  - o `gpfs_max_clone_depth`: limite superiore al cloning depth di GPFS (o: illimitato)
  - o `gpfs_sparse_volumes`: "True" per creare sparse file alla creazione dei volumi
  - o `gpfs_storage_pool`: specifica lo storage pool di GPFS a cui i file corrispondenti ai volumi Cinder sono assegnati (default: "system")
  - o `volume_backend_name`: stringa per identificare il backend name

# Cinder: configurazione

---

- Inserire il nodo Cinder nel cluster GPFS (se non già membro)
- Creare una directory per lo store dei dati di Cinder nel fileset appositamente creato

```
# mkdir -p /gpfs/openstack/cinder/volumes  
# chmod -R 755 /gpfs/openstack/cinder
```

- Assegnare l'ownership opportuna

```
# chown -R <cinder-user> /gpfs/openstack/cinder
```

- l'ownership corretta dipende dalla distribuzione di OpenStack installata: verificare l'ownership dello user sotto cui girano i processi di Cinder

# Cinder: configurazione

---

Nel file `/etc/cinder/cinder.conf`:

- Creare una sessione `[gpfs]` in cui inserire le variabili del driver già viste:

```
[gpfs]
volume_driver = cinder.volume.drivers.ibm.gpfs.GPFSDriver
gpfs_mount_point_base = /gpfs/openstack/cinder/volumes/
gpfs_images_dir = /gpfs/openstack/glance/images/
gpfs_images_share_mode = copy_on_write
gpfs_max_clone_depth = 3
gpfs_sparse_volumes = True
gpfs_storage_pool = system
volume_backend_name = GPFS
```

- Nella sessione `[DEFAULT]`  
`enabled_backends = [...,] gpfs`  
`default_volume_type = gpfs`

# Cinder: restart

---

- Dopo la modifica dei file di configurazione, far ripartire i servizi di Cinder
  - cinder-api
  - cinder-volume
  - cinder-schedule

# Cinder: multiple backend e volume type

---

- Se Cinder utilizza piu' backend, e' necessario creare un **volume type** per il backend GPFS appena definito
- Tramite l'interfaccia di Horizon
  - login come admin
  - 'Admin'->'Volumes'->'Volume types'->'Create Volume Type'
    - specificare nome 'gpfs'
    - Selezionare 'View Extra Specs' del volume type creato, e crearne una
      - key: volume\_backend\_name
      - value: GPFS
- Tramite command line:

```
# cinder type-create gpfs
```

```
# cinder type-key gpfs set volume_backend_name=GPFS
```

# GPFS backend per Nova

---

# Nova: configurazione

---

- Inserire i nodi Nova nel cluster GPFS (se non già membro)
- Creare una directory per lo store delle immagini effimere di Nova nel fileset appositamente creato per i servizi

```
# mkdir -p /gpfs/openstacknova/instances  
# chmod -R 755 /gpfs/openstack/nova
```

- Assegnare l'ownership opportuna

```
# chown -R <nova-user> /gpfs/openstack/nova
```

- l'ownership corretta dipende dalla distribuzione di OpenStack installata: verificare l'ownership dello user sotto cui girano i processi di Nova
- su tutti i compute nodes l'account per nova deve avere **stessi UID e GID**

# Nova: configurazione (cont.)

---

- Modificare nel file `/etc/nova/nova.conf`, sezione `[DEFAULT]`, il parametro:

```
instances_path = /gpfs/openstack/nova/instances
```

- Fare restart dei servizi Nova
  - `nova-api`
  - `nova-compute`
  - `nova-scheduler`
- Queste operazioni devono essere fatte su **tutti i compute nodes**

# Vantaggio di uno shared filesystem

---

- L'utilizzo di GPFS come backend per i volumi effimeri di Nova permette di supportare la live migration di VM
  - il file system GPFS e' **shared** tra i compute nodes
  - tutti i compute nodes utilizzano e vedono lo **stesso path per i volumi effimeri** e per i volumi Cinder

# Esercitazione

---

- [https://wiki.ge.infn.it/calcolo/index.php/Corso\\_Cloud\\_Storage\\_Es4](https://wiki.ge.infn.it/calcolo/index.php/Corso_Cloud_Storage_Es4)