Ceph with CloudStack

Andrija Panic™
Cloud Architect
andrija.panic@shapeblue.com
Twitter: @AndrijaRS

About me / sobre mí / 關於我 / 私について

- **Cloud Architect** @ ShapeBlue
- From Belgrade, Serbia
- Committer and PMC member
- Involved with CloudStack since version 4.0.0-incubating
- Interested in:
  - Cloud infrastructure architecture and engineering.
  - Virtualization, Storage and SDxx
- Downtime:
  - Father to 2 princesses
  - Music, gym and hobby electronic

Quick Ceph intro

Shape Blue The Cloud Specialists

cloudstack

Ceph

"The name Ceph comes from cephalopod, a class of molluscs that includes the octopus and squid… the reasoning had something to do with their high level of intelligence and "many-tentacled", "distributed" physiology."

Sage Weil

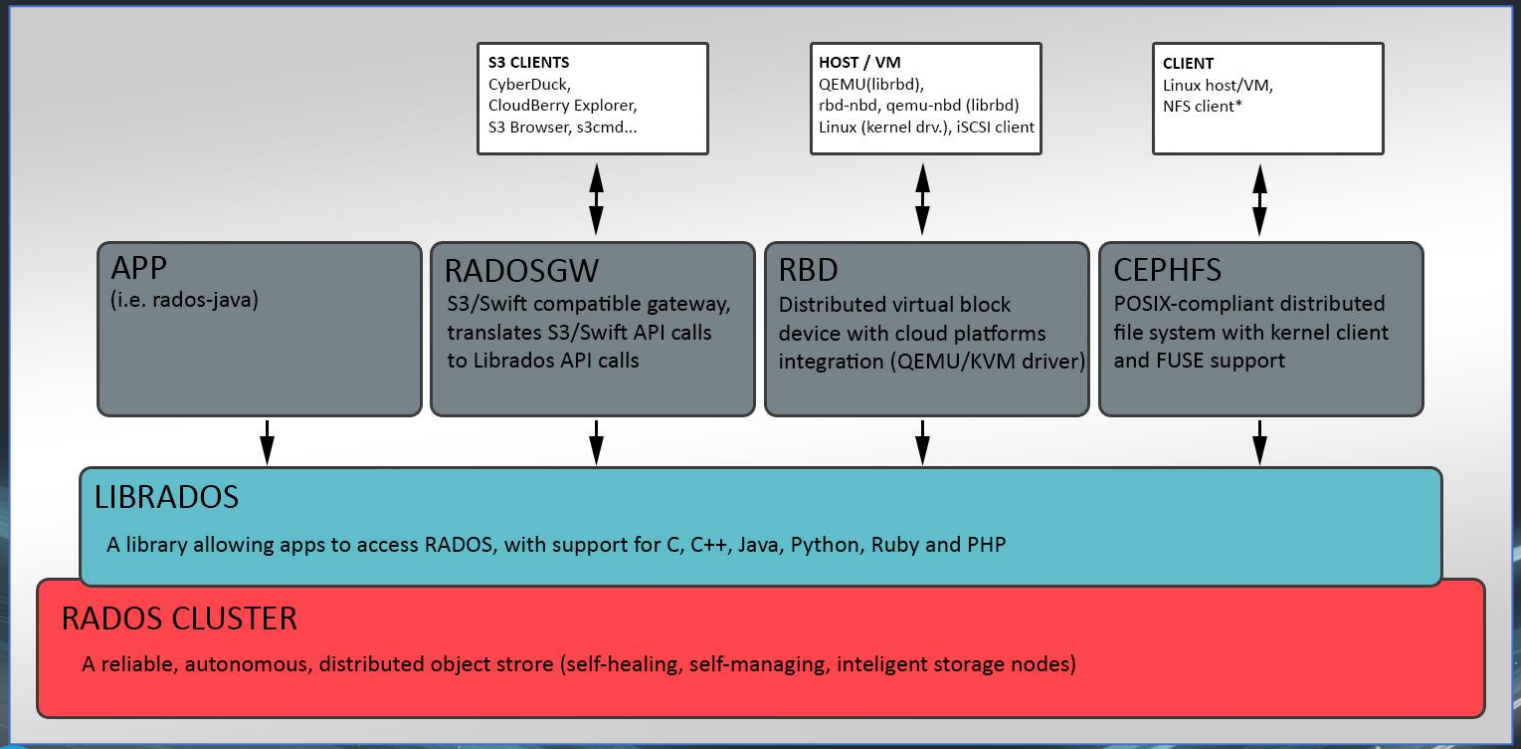
Fun facts:

- Cephalopods have the most complex nervous system of all the invertebrates.
- Some can fly up to 50m through the air, squirting water to help propel themselves.
- Most have special coloured pigments on their skin that are used for camouflage.
- Cephalopods have advanced vision, but most are colour blind.
- They have an ink sac that they squirt into the water to confuse predators

Overview

- Open source SDS solution
- Highly scalable (tens of thousands of nodes)
- No single point of failure
- Hardware agnostic, “runs on commodity hardware”
- Self-managed whenever possible
- Built around the CRUSH algorithm
- Provides multiple access methods:
  - File
  - **Block**
  - Object (S3/Swift)
  - NFS gateway (third-party sw.) for backward compatibility

Architecture



Ceph Storage Cluster

- The Ceph Storage Cluster (RADOS cluster) is the foundation for all Ceph deployments.
- Based upon RADOS, consists of three types of daemons:
  - Ceph Object Storage Daemon (OSD)
  - Ceph Monitor (MON)
  - Ceph Meta Data Server (MDS) - optionally
- A minimal possible system will have at least one Ceph Monitor and two Ceph OSD Daemons for data replication.
- Production system will have at least 3 monitors (redundancy) and minimum 10 OSD nodes (i.e. 80+ OSDs)

Ceph Storage Cluster

Ceph Storage Cluster (RADOS cluster)

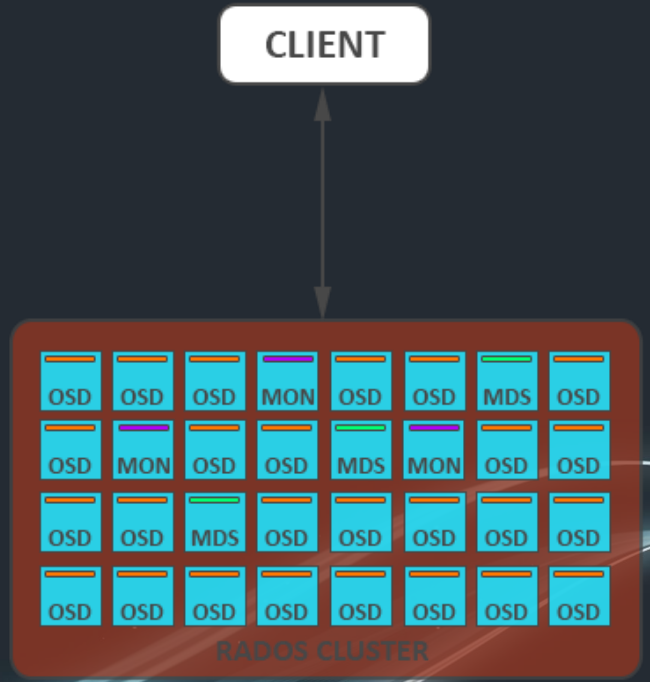
- OSD and MON are mandatory for every cluster
- MDS is required only if using Ceph FS

OSDs:

- 10s to 10000s in a cluster, one per disk (HDD, SSD, NVME)
- Serve stored objects to clients
- Intelligently peer to perform replication/recovery tasks

MONs:

- Maintain a master copy of the Ceph cluster map, cluster membership and state
- Provide consensus for distributed decision-making via PAXOS algorithm
- Small, odd number, do not serve objects to clients



Ceph with CloudStack (PoC)

(1. Preparation)

(2. Installation)

(3. Pool preparation)

(4. Adding Ceph to CloudStack)

Shape Blue The Cloud Specialists

cloudstack

Ceph Storage Cluster – PoC Installation

Preparation

- Make sure the time across all servers is synced with less then 0.05sec of difference! (don’t worry, Ceph will complain if not synced)
- Make sure that “hostname --fqdn” is resolvable between all nodes
- Make sure key-based ssh auth from admin node to all cluster nodes is working (sudo)
- Add proper release repo on the “admin” node, install “ceph-deploy”

Ceph Storage Cluster – PoC Installation (cntd.)

Installation (using ceph-deploy from the admin node)

- mkdir mycluster; cd mycluster;
- ceph-deploy new ceph-node1 ceph-node2 ceph-node3 *(make cluster def.)*
- ceph-deploy install --release nautilus ceph-node1 ceph-node2 ceph-node3 *(install binaries only)*
- ceph-deploy mon create-initial *(create MONs across initially added Ceph nodes)*
- ceph-deploy admin ceph-node1 ceph-node2 ceph-node3 *(copy ceph.conf and the needed keyrings)*
- for n in 1 2 3; do ceph-deploy osd create --data /dev/sdb ceph-node$n; done (*deploy single OSD per node)*

Ceph dashboard (optional but recommended)

- yum install -y ceph-mgr-dashboard
- ceph config set mgr mgr/dashboard/ssl false
- ceph mgr module enable dashboard
- ceph dashboard ac-user-create admin password administrator

Ceph Storage Cluster – PoC Installation (cntd.)

Create a pool for CloudStack

- ceph osd pool create cloudstack 64 replicated
- ceph osd pool set cloudstack size 3
- rbd pool init cloudstack
- ceph auth get-or-create client.cloudstack mon 'profile rbd' osd 'profile rbd pool=cloudstack'*

  Example key:

  [client.cloudstack]

  key = AQAb6M9cY1epJBAAZgzlOlpZSpBcUpYCBWTFrA==

Configure write-back caching on KVM nodes (setup ssh/name resolution from the admin node)

- cat << EOM >> /root/mycluster/ceph.conf

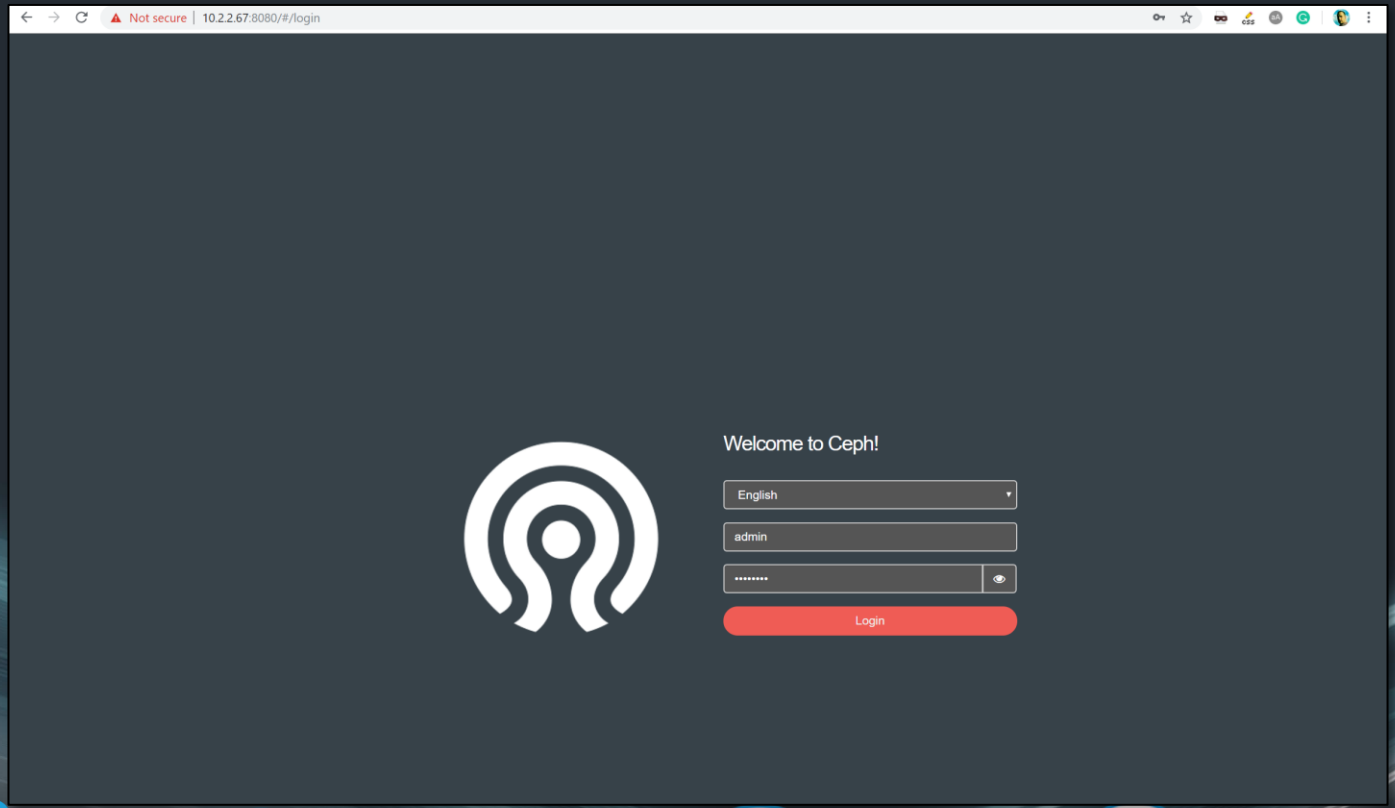
  [client]

  rbd cache = true

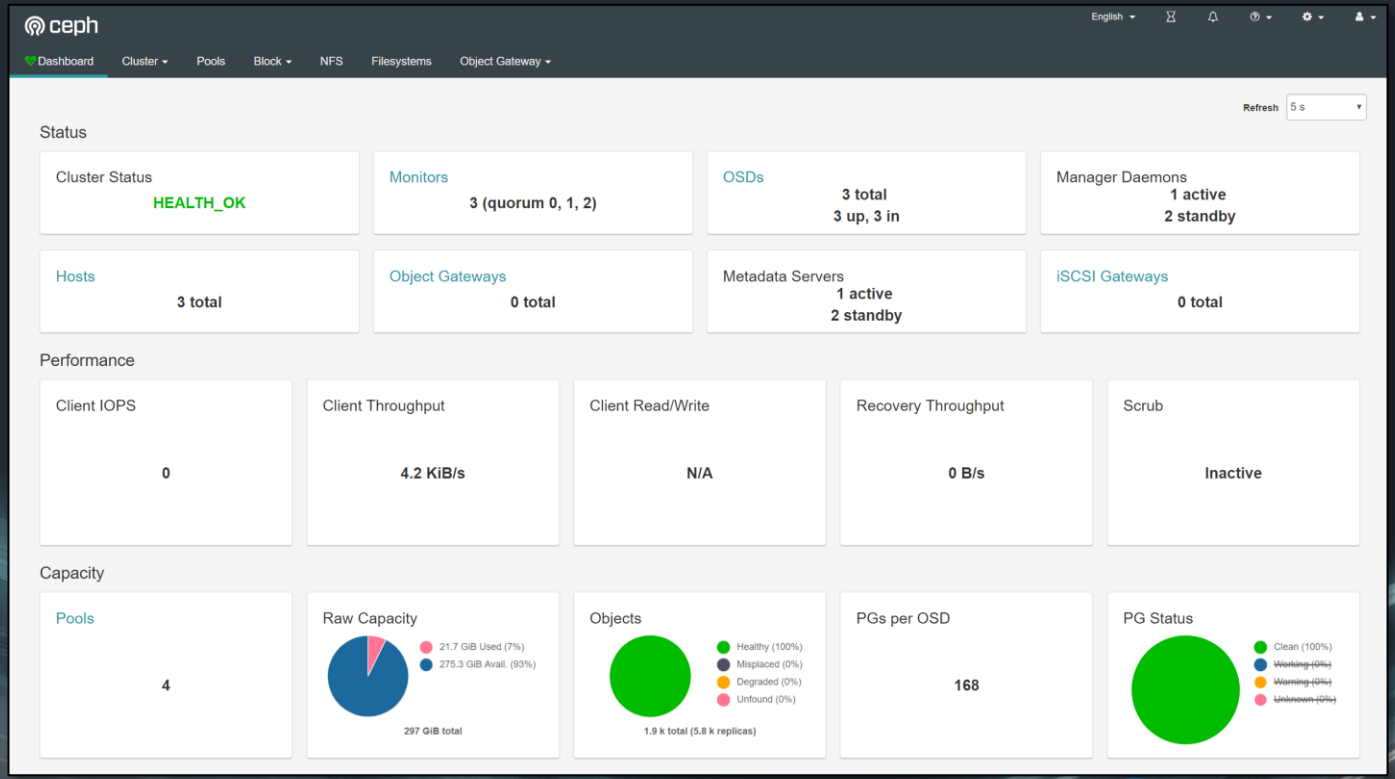
  rbd cache writethrough until flush = true

  EOM
- ceph-deploy --overwrite-conf admin kvm1 kvm2 kvm3

New dashboard – demo



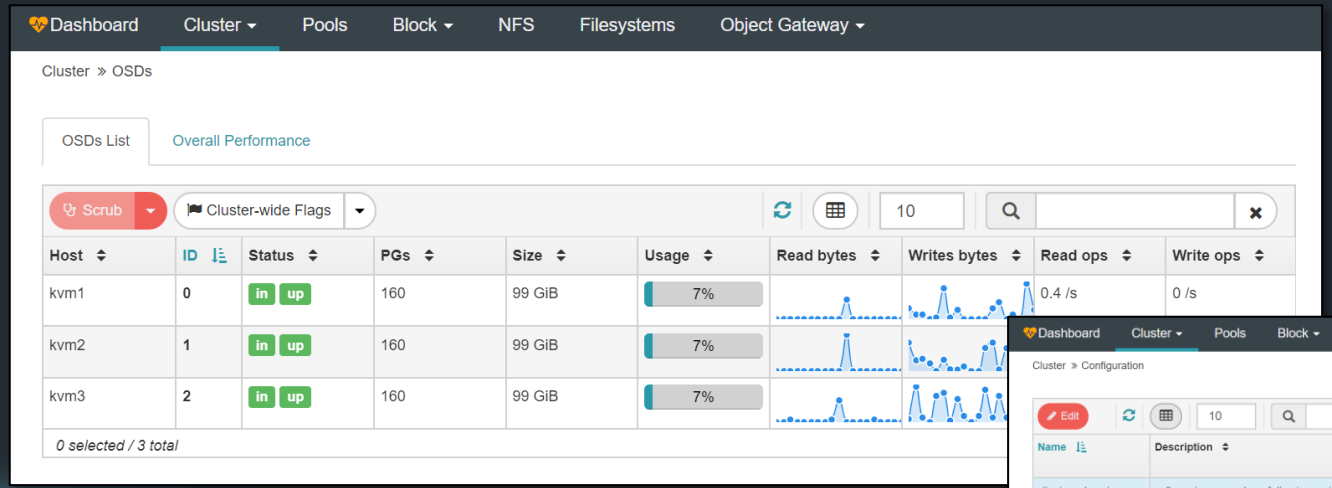
Ceph Storage Cluster – New Dashboard



Shape Blue The Cloud Specialists

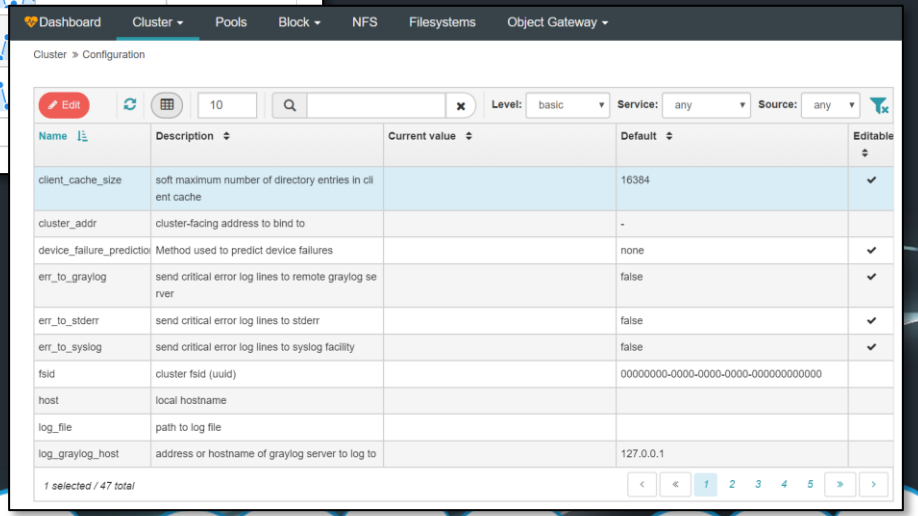
cloudstack

Ceph Storage Cluster – New Dashboard



View/Manage OSDs

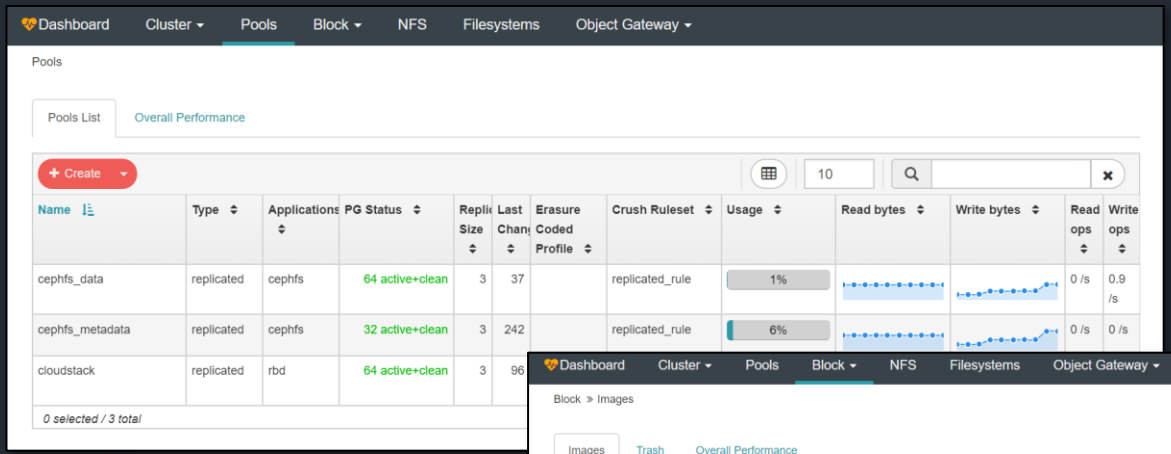
Manage basic cluster configs



Shape Blue The Cloud Specialists

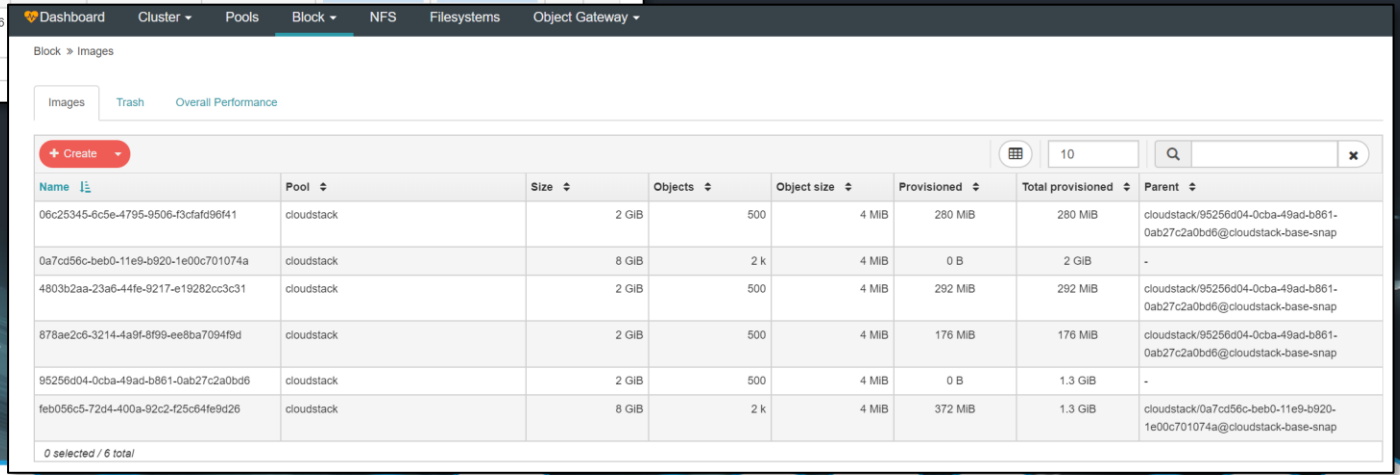
cloudstack

Ceph Storage Cluster – New Dashboard



View/Manage pools

View/Manage RBD images



Shape Blue The Cloud Specialists

cloudstack

Ceph Storage Cluster – New Dashboard

New in Nautilus (based on SUSE's OpenATTIC mostly)

- OSD management (mark as down/out, change OSD settings, recovery profiles)
- Cluster config settings editor
- Ceph Pool management (create/modify/delete)
- ECP management
- RBD mirroring configuration
- Embedded Grafana Dashboards (derived from Ceph Metrics)
- CRUSH map viewer
- NFS Ganesha management
- iSCSI target management (via ceph-iscsi)
- RBD QoS configuration
- Ceph Manager (ceph-mgr) module management
- Prometheus alert Management
- Support for multiple users / roles; SSO (SAMLv2) for user authentication

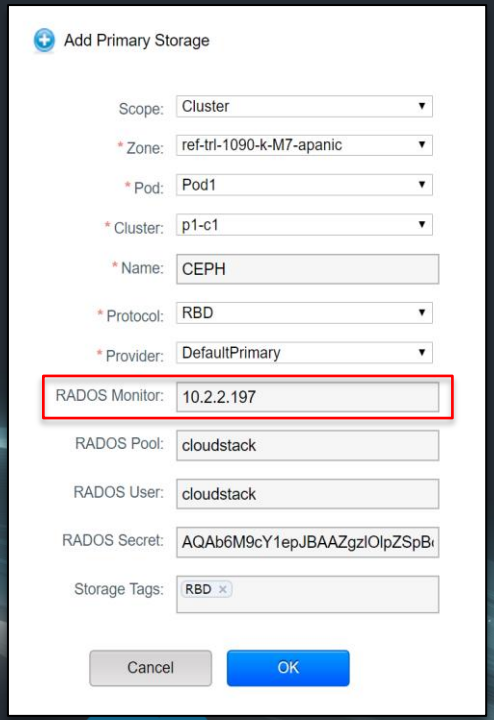
Ceph Storage Cluster – New in Nautilus

(Some) Nautilus improvements:

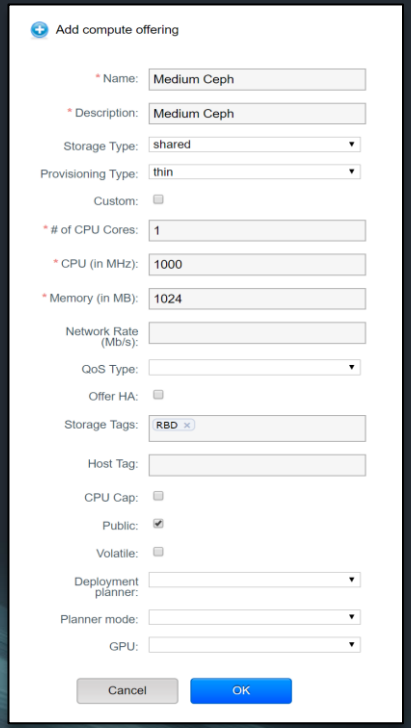
- pg_num can be reduced; can be auto-tuned in the background
- OSD and mon report SMART stats; Failure prediction; Optional automatic migration*
- Mon protocol v2, port 6789 → 3300 (IANA); encryption; dual (v1 and v2) support
- osd_target_memory; NUMA mgmt & OSD pinning; misplaced no more HEALHT_WARN
- S3 tiering policy, bucket versioning
- RBD live image migration (librbd only); rbd-mirror got simpler; rbd top & and rbd CLI;
- CephFS multi-fs support stable; Clustered nfs-ganesha (active/active)
- Run Ceph clusters in Kubernetes (Rook, ceph-ansible)

Ceph Storage Cluster – PoC Installation (cntd.)

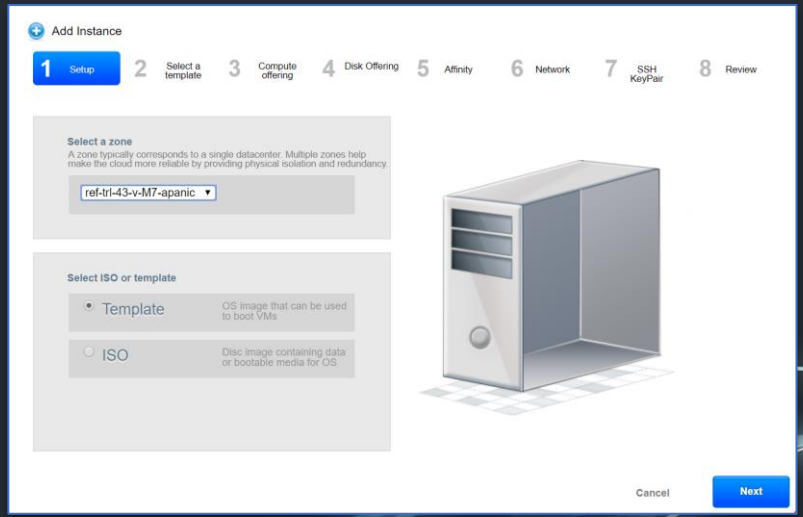
Add Ceph to CloudStack



Create offerings for Ceph



Deploy a VM

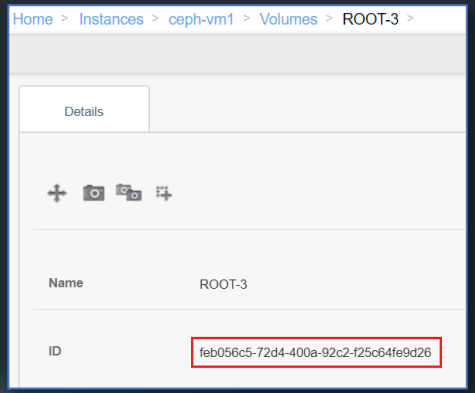


Shape Blue The Cloud Specialists

cloudstack

Finding your way around

Let’s check our ACS volume on Ceph




```
[root@kvm3 ~]# rbd -p cloudstack ls
06c25345-6c5e-4795-9506-f3cfafd96f41
0a7cd56c-beb0-11e9-b920-1e00c701074a
4803b2aa-23a6-44fe-9217-e19282cc3c31
878ae2c6-3214-4a9f-8f99-ee8ba7094f9d
95256d04-0cba-49ad-b861-0ab27c2a0bd6
feb056c5-72d4-400a-92c2-f25c64fe9d26
```

```
[root@kvm3 ~]# rbd info cloudstack/feb056c5-72d4-400a-92c2-f25c64fe9d26
rbd image 'feb056c5-72d4-400a-92c2-f25c64fe9d26':
        size 8 GiB in 2048 objects
        order 22 (4 MiB objects)
        snapshot_count: 2
        id: 38eb1f16e9e8
        block_name_prefix: rbd_data.38eb1f16e9e8
        format: 2
        features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
        op_features:
        flags:
        create_timestamp: Wed Aug 14 22:16:41 2019
        access_timestamp: Wed Aug 14 22:16:41 2019
        modify_timestamp: Wed Aug 14 22:16:41 2019
        parent: cloudstack/0a7cd56c-beb0-11e9-b920-1e00c701074a@cloudstack-base-snap
        overlap: 8 GiB
[root@kvm3 ~]#
```

Finding your way around

Volume provisioning steps:

- Copy template from SS to Ceph: “0a7cd56c-beb0-11e9-b920-1e00c701074a”
- Create a base snapshots and protect it (can’t be deleted): “cloudstack-base-snap”
- Create a VM’s volume as the child (clone) of the snap: “feb056c5-72d4-400a-92c2-f25c64fe9d26”

Find all volumes (children) of specific template (base-snap of the template image)

<root@ceph1># rbd children cloudstack/0a7cd56c-beb0-11e9-b920-1e00c701074a@cloudstack- base-snap
cloudstack/feb056c5-72d4-400a-92c2-f25c64fe9d26
cloudstack/8481fcb1-a91e-4955-a7fc-dd04a44edce5
cloudstack/9b8f978b-74d0-48f7-93f6-5e06b9eb6fd3
cloudstack/3f65da05-268f-41fa-99b2-ce5d4e6d6597
…

Finding your way around

Manually reproducing the ACS behavior:

rbd create -p cloudstack mytemplate --size 100GB (or “qemu-img” convert, or “rbd import”…)

rbd snap create cloudstack/mytemplate@cloudstack-base-snap

rbd snap protect cloudstack/mytemplate@cloudstack-base-snap

rbd clone cloudstack/mytemplate@cloudstack-base-snap cloudstack/**myVMvolume**

…and the cleanup:

[root@ceph1 ~]# rbd rm cloudstack/myVMvolume

Removing image: 100% complete...done.

[root@ceph1 ~]# rbd snap unprotect cloudstack/mytemplate@cloudstack-base-snap

[root@ceph1 ~]# rbd snap rm cloudstack/mytemplate@cloudstack-base-snap

Removing snap: 100% complete...done.

[root@ceph1 ~]# rbd rm cloudstack/mytemplate

Removing image: 100% complete...done.

Finding your way around

“Hacking” the customer’s volume:

- rbd map myPool/myImage (kernel client)
  (will usually fail due to kernel client “rbd.ko” being way behind the cluster version/capabilities)
- rbd-nbd map myPool/myImage (user-space, via librbd)
  (requires “yum install rbd-nbd” and “modprobe nbd max_part=15*”)
- qemu-nbd --connect=/dev/nbd0 rbd:myPool/myImage (user-space, via librbd)
  (requires “modprobe nbd*”)

Qemu-img:

- qemu-img info rbd:cloudstack/47b1cfe5-6bab-4506-87b6-d85b77d9b69c*
- qemu-img info rbd:cloudstack/47b1cfe5-6bab-4506-87b6-d85b77d9b69c:mon_host=10.x.x.y:auth_supported=Cephx:id=cloudstack:key=AQAFSZ……..jEtr/g==

Some limitations

- No support for a full VM snapshot (technically not possible with Ceph/iSCSI/raw block devices)
- No support for the storage heartbeat file (yet…)
- Currently not possible to really restore a volume from a snapshot (old behaviour stays*)
- Two “external” libraries to be aware of – librbd and rados-java

Learning curve

Not your average NFS:

- Ceph can be a rather complex storage system to comprehend
- Make sure you know the storage system well before relying on it in production
- Make sure to excel at troubleshooting, you'll need it sooner or later
- Understand how the things works under the hood
- Understand recovery throttling to avoid high impact on customer IO

Performance considerations

- “Works on commodity hardware”, but don’t expect miracles
- Writing data to primary OSD and replicating that write to another 2 OSDs, takes time
- Latency is very good with NVME (0.5ms-1ms)
- Not so very good with HDD/SSD mix (10ms-30ms)
- Never, ever, ever… use consumer SSDs; bench and test specific enterprise SSD models
- Too many parallel stream end up generating pure random IO pattern on the backend
- Ceph was (unofficially) considered unsuitable for serious random IO workload (2-3y ago)*

Performance considerations (cntd.)

Things have seriously changed last few years (especially with the new BlueStore backend)

- Writing to the raw device (“block”) vs. XFS on FileStore;
- RockDB (“block.db”, “block.wal”) vs. LevelDB
- Now suitable for pure SSD/NVME clusters
- Increased throughput 40-300%*, reduced latency 30-50%* vs. FileStore
- Explicit memory management* (BlueStore runs in user-space)
- Data and metadata checksums; Compression
- Reads still served from Primary OSD only ☹

Additional info

Step by step guide for Ceph with CloudStack (Mimic):

- https://www.shapeblue.com/ceph-and-cloudstack-part-1/
- https://www.shapeblue.com/ceph-and-cloudstack-part-2/
- https://www.shapeblue.com/ceph-and-cloudstack-part-3/

Ceph with

CloudStack

ShapeBlue.com • @ShapeBlue

Andrija Panic, Cloud architect • PMC Apache CloudStack
andrija.panic@shapeblue.com • @AndrijaRS

Shape Blue The Cloud Specialists

cloudstack