cloudera®

Reference Architecture for Deploying Cloudera Enterprise 5.x on Red Hat OpenStack Platform 11 with Red Hat Ceph Storage 2.x

Important Notice

Cloudera, Inc.
395 Page Mill Road
Palo Alto, CA 94306
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com

Release Information

Date: 10/30/2017

Version: 5.13

Table of Contents

Reference Architecture for Deploying Cloudera Enterprise 5.x in Red Hat OpenStack Platform 11 and Red Hat Ceph Storage 2.x

Executive Summary

This document provides a reference architecture for deploying Cloudera Enterprise including CDH on Red Hat's OpenStack Platform (OSP) 11. Much like the Hadoop platform, OpenStack is comprised of a number of related projects to control pools of storage, processing, and networking resources within a data center, and to build a multi-datacenter private cloud infrastructure. The following OpenStack projects are in scope for this release of the reference architecture:

- Compute (Nova): on-demand computing resources from a large network of virtual machines
- Ceph: Software Defined Distributed Storage system which provides Object Store, Remote Block Storage and Shared Filesystem capabilities. This document deals with the Remote Block Storage (RADOS Block Device or RBD) aspect of Ceph.
- Storage Service (Cinder): storage management and provisioning for Cloudera Instances with Remote Software Defined Distributed Storage backend - Ceph RBD.
- Networking (Neutron): flexible models for managing networks and IP addresses (includes Open vSwitch)
- Image service (Glance): discovery, registration, and delivery for disk and virtual machine images
- Identity Management service (Keystone): Manage identity and authorizations for various system users, projects and end-users who will use the OpenStack self-service infrastructure

This release of the reference architecture describes deploying Cloudera's Distribution of Apache Hadoop (CDH) 5.11 on Red Hat OSP 11. This reference architecture articulates a specific design pattern which is recommended to be administrator-driven as opposed to end-user self-service based. The RA will also be applicable for all 5.x releases of CDH subsequent to C 5.11.

Business Objectives

The objective of this reference architecture is to provide safe and reliable design patterns that customers can use to leverage OpenStack to deploy Cloudera EDH IaaS clusters in private cloud environments, while approaching the inherent flexibility of cloud based deployments, with remote block storage.

NOTE:

- **This document extends the scope from the first Reference Architecture for OSP 11 with Locally attached storage to now include Ceph RBD (RADOS Block Device) as a remote block storage tier.**

Cloudera Enterprise

Cloudera is an active contributor to the Apache Hadoop project and provides an enterprise-ready, 100% open-source distribution that includes Hadoop and related projects. The Cloudera distribution bundles the innovative work of a global open-source community, including critical bug fixes and important new features from the public development repository, and applies it to a stable version of the source code. In short, Cloudera integrates the most popular projects related to Hadoop into a single package that is rigorously tested to ensure reliability during production.

Cloudera Enterprise is a revolutionary data-management platform designed specifically to address the opportunities and challenges of big data. The Cloudera subscription offering enables data-driven enterprises to run Apache Hadoop production environments cost-effectively with repeatable success. Cloudera Enterprise combines Hadoop with other open-source projects to create a single, massively scalable system in which you can unite storage with an array of powerful processing and analytic frameworks—the Enterprise Data Hub. By uniting flexible storage and processing under a single management framework and set of system resources, Cloudera delivers the versatility and agility required for modern data management. You can ingest, store, process, explore, and analyze data of any type or quantity without migrating it between multiple specialized systems.

Cloudera Enterprise makes it easy to run open-source Hadoop in production:

Accelerate Time-to-Value

- Speed up your applications with HDFS caching
- Innovate faster with pre-built and custom analytic functions for Cloudera Impala

Maximize Efficiency

- Enable multi-tenant environments with advanced resource management (Cloudera Manager + YARN)
- Centrally deploy and manage third-party applications with Cloudera Manager

Simplify Data Management

- Data discovery and data lineage with Cloudera Navigator
- Protect data with HDFS and HBase snapshots
- Easily migrate data with NFSv3 support

See Cloudera Enterprise for more detailed information.

Cloudera Enterprise can be deployed in a Red Hat OpenStack Platform based infrastructure using the reference architecture described in this document.

About Red Hat

Red Hat is the world's leading provider of open source software solutions, using a community-powered approach to reliable and high-performing cloud, Linux, middleware, storage, and virtualization technologies. Red Hat also

offers award-winning support, training, and consulting services. As a connective hub in a global network of enterprises, partners, and open source communities, Red Hat helps create relevant, innovative technologies that liberate resources for growth and prepare customers for the future of IT.

About Red Hat OpenStack Platform

Red Hat OpenStack Platform allows customers to deploy and scale a secure and reliable private or public OpenStack cloud. By choosing Red Hat OpenStack Platform, companies can concentrate on delivering their cloud applications and benefit from innovation in the OpenStack community, while Red Hat maintains a stable OpenStack and Linux platform for production deployment.

Red Hat OpenStack Platform is based on OpenStack community releases, co-engineered with Red Hat Enterprise Linux 7. It draws on the upstream OpenStack technology and includes enhanced capabilities for a more reliable and dependable cloud platform, including:

- Red Hat OpenStack Platform director, which provides installation, day-to-day management and orchestration, and automated health-check tools, to ensure ease of deployment, long-term stability, and live system upgrades for both core OpenStack services, as well as the director itself.
- High availability for traditional business-critical applications via integrated, automated monitoring and failover services.
- Stronger network security and greater network flexibility with OpenStack Neutron modular layer 2 (ML2), OpenvSwitch (OVS) port security, and IPv6 support.
- Integrated scale-out storage with automated installation and setup of Red Hat Ceph Storage.
- A large OpenStack ecosystem, which offers broad support and compatibility, with more than 350 certified partners for OpenStack compute, storage, networking, and independent software vendor (ISV) applications and services.

About Red Hat Ceph Storage

Red Hat Ceph Storage is an open, cost-effective, software-defined storage solution that:

- Decouples software from hardware to run cost-effectively on industry-standard servers and disks.
- Scales flexibly and massively to support multiple petabyte deployments with consistent performance.
- Provides network block storage for modern use cases, such as cloud infrastructure, media repository, and big data analytics.
- Combines the most stable version of Ceph with a storage management console, deployment tools, and support services.

Target Audience and Scope

This reference architecture is aimed at Datacenter, Cloud, and Hadoop architects who will be deploying Cloudera's Hadoop stack on private OpenStack cloud infrastructure.

Specifically, this document articulates a design pattern that involves using Red Hat Ceph Storage 2.x as a remote storage backend for Red Hat OSP 11, such that all VMs can be provisioned on block devices backed by Ceph. However, only the Ceph RBD (RADOS Block Device) functionality is covered in this document. It articulates a specific design pattern which is recommended to be administrator-driven as opposed to end-user self-service based. The RA will also be applicable for all 5.x releases of CDH subsequent to C 5.13.

Following components of RedHat OSP 11 are in scope for this document --

- Compute (Nova): on-demand computing resources from a large network of virtual machines
- Ceph: Software Defined Distributed Storage system which provides Object Store, Remote Block Storage and Shared Filesystem capabilities. This document deals with the Remote Block Storage (RADOS Block Device or RBD) aspect of Ceph.
- Storage Service (Cinder): storage management and provisioning for Cloudera Instances with Remote Software Defined Distributed Storage backend - Ceph RBD.
- Networking (Neutron): flexible models for managing networks and IP addresses (includes Open vSwitch)
- Image service (Glance): discovery, registration, and delivery for disk and virtual machine images
- Identity Management service (Keystone): Manage identity and authorizations for various system users, projects and end-users who will use the OpenStack self-service infrastructure

Following components of RedHat OSP 11 are NOT in scope for this document --

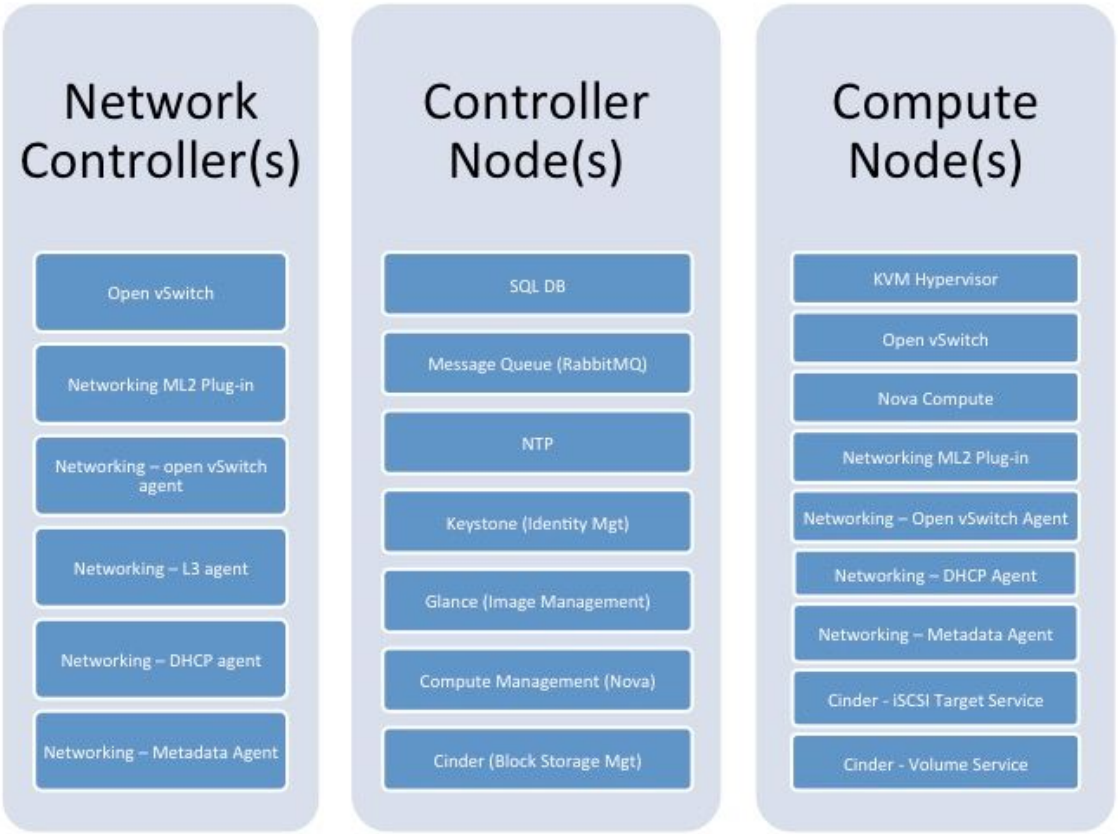
- Telemetry and alerting (Ceilometer)
- Monitoring (Horizon)
- Elastic mapreduce (Sahara)
- Orchestration (Heat)
- Bare metal (Ironic)
- Object Storage services provided by Swift and Ceph.

Future editions may include more information on these OpenStack projects.

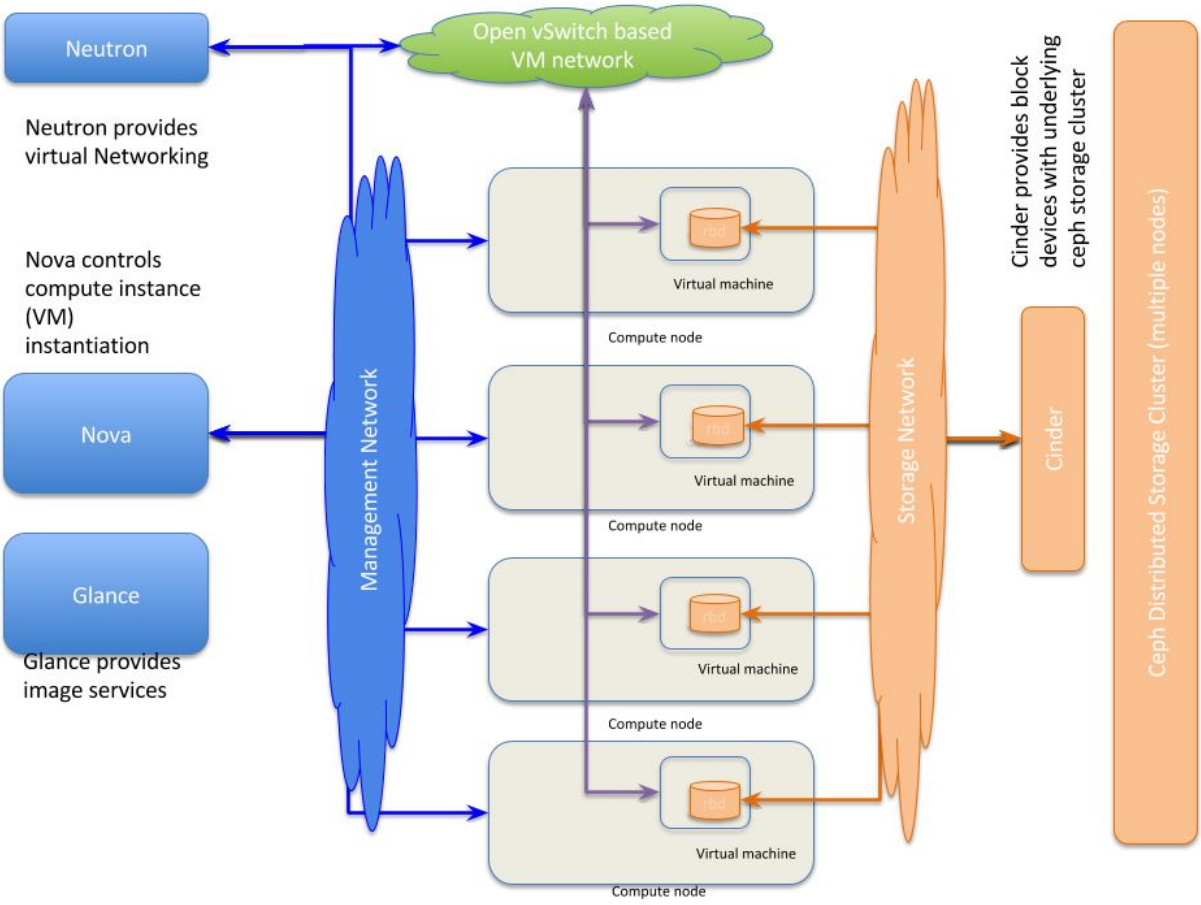
Reference Architecture

Component design

The following diagram illustrates the various components of the OpenStack deployment. Not all the components shown in this high level diagram are covered in this reference architecture document. Please refer to the Target Audience and Scope - it highlights which components are considered in scope and which are considered out of scope for this revision.



High level Red Hat OSP diagram 1



High Level Stack Architecture 2

Component Table

Component Role	Quantity	Component Details
OpenStack Controller	3	● 2 sockets with 6-10 cores per socket ● 128GB RAM ● 2x 10GbE NICs ○ 1 for compute/tenant ○ 1 for management ○ ● 6x 2TB+ internal HDDs ○ 4x drives in RAID-10 for DBs ○ 2x drives in RAID-1 for OS bits
Compute Node	Minimum 8. Max depends on use-case.	● 2 sockets with 6-10 cores per socket ● At least 256GB RAM ● 3x 10GbE NICs ○ 1 for tenant ○ 2x bonded (link aggregation) for management and storage traffic ● 2x 2TB+ internal HDDs ○ 2 x HDDs in RAID-1 for OS bits
Storage Node	Minimum 8.	● 2 sockets with 6-10 cores per socket ● 128GB RAM ● 3x 10GBE NICs ○ 2x 10GbE bonded for storage ○ 1x 10GbE for management ● 24-36 2TB+ internal HDDs ○ 2x HDDs in RAID-1 for OS bits ○ 22-34x HDDs in JBOD mode for storage ○ 3-4x 800GB NVMe SSDs for write journaling

Set up 3 controller nodes in HA configuration. This will ensure that the various key components of the OpenStack deployment will continue to run in case of a hardware failure

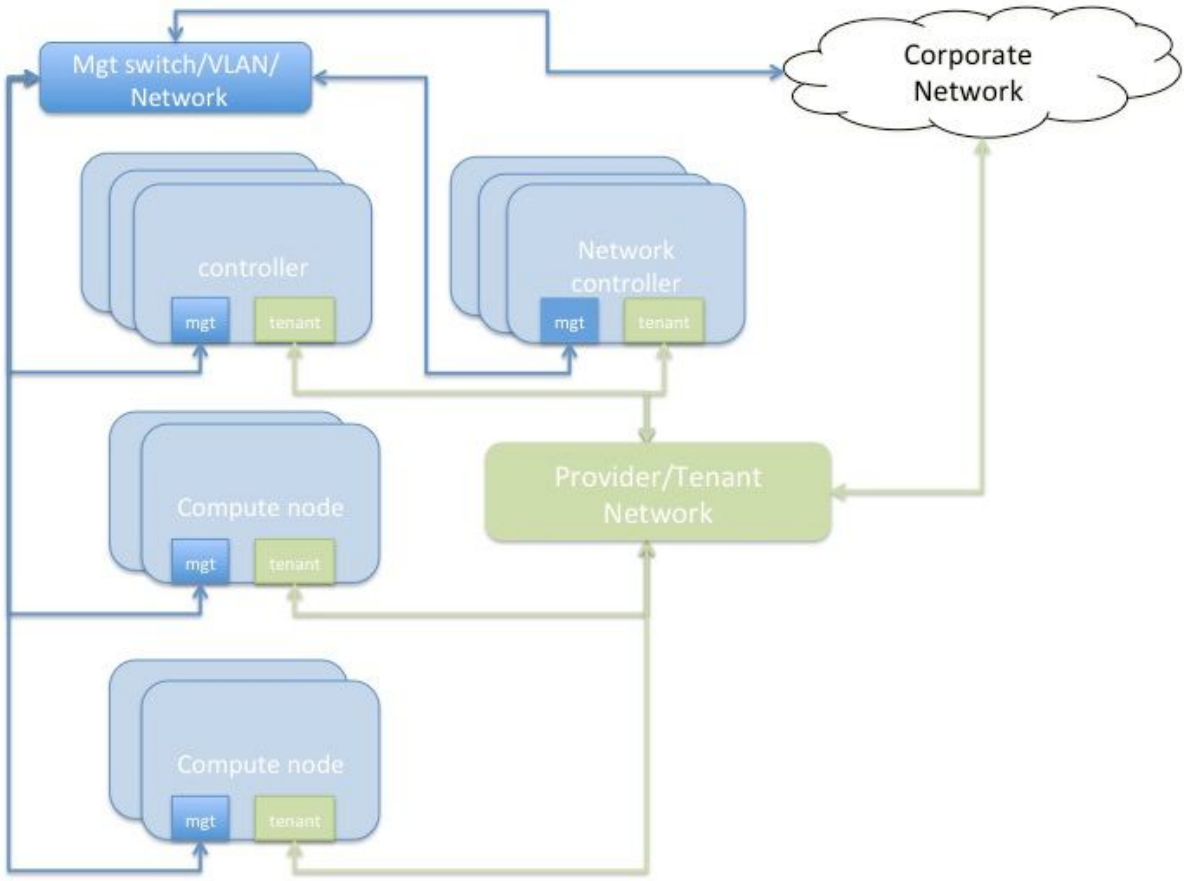
A minimum of 3 master and 5 worker nodes (CDH) are needed to ensure that when HDFS blocks are placed within VMs running on these nodes we have physical disparity to match the 3x replication factor of HDFS. We will use HVE to ensure that duplicate copies of any HDFS block are not placed on the same compute node. But there need to be at least the physical availability of 8 compute nodes.

The storage devices (HDDs) should be set up as JBODs and we should have 1 x HT CPU core driving 1 drive. So on a 12-physical core system (24 HT cores), we should consider having at least 24 physical drives.

The drives should be configured with one single large partition and formatted in XFS. The usable capacity with default Ceph 3x replication is 1:3.2 (so if you want 100TB of usable storage capacity, you have to provision ~ 320TB of raw storage). Recommended minimum HDFS Replication Factor with this is 2. That will yield ~ 50TB of usable HDFS capacity.

Network

This section covers the network topology used in development of this reference architecture, as well as a brief summary of options available in the OpenStack ecosystem in general. A generic guideline for networking would be to advise the customers to pick a model that yields highest network throughput, or at least sufficient network throughput to match the theoretical throughput capabilities of the disks being presented to the VMs on each physical node.



Network topology diagram 2

Controller and compute nodes have 3 x 10GbE NICs each - one will provide the tenant network, the other is a bonded pair of NICS for storage and management traffic which is used for OS provisioning of the physical infrastructure, as well as provide data path for other OpenStack management traffic, and storage traffic.

There are two general flavors of network topology that can be used in an OpenStack based private cloud.

1. Provider Networks -- Provider Networks are essentially physical networks (with physical routers) and are managed by the OpenStack administrators. End-users cannot manage and make changes to these networks. They are the simplest and also the most performant. They entail connecting directly to the physical network infrastructure with minimal SDN (Software Defined Networking) functionality being used.

2. Self-Service networks -- These are networks that can be created and managed by the OpenStack end-users. The underlying physical infrastructure can be provider networks, but there would a virtualized overlay using VXLAN or GRE tunneling. These would typically be private networks which will be routed through a software router hosted on a network controller node.

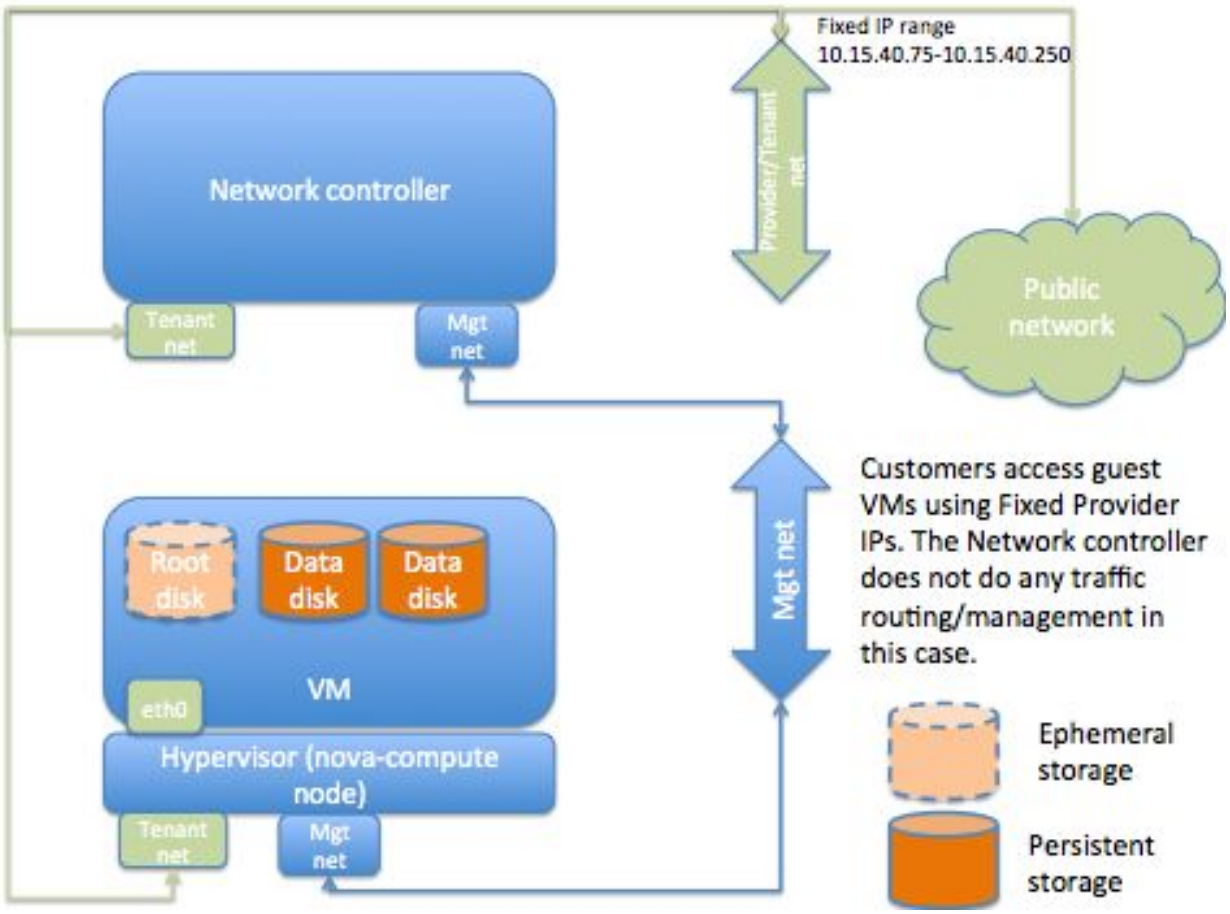
NOTE:

- In our labs, we have used a provider-network based deployment, which provides better network performance for Hadoop workloads, wherein each compute node is able to directly access the physical network infrastructure. This model is however limiting in terms of flexibility in scenarios where self-service capabilities are needed.
- For best network performance, consider using SR-IOV. This will allow the VMs to directly access previously defined virtual functions created on physical NICs. This option is further limiting in terms of flexibility, and the NIC hardware is subject to supportability on Red Hat OSP.
- For a more detailed understanding of the various networking options available in Red Hat OSP 11, refer to the Networking Guide.

Compute (Nova)

The compute nodes' design considerations are as follows -

a. The hypervisor (KVM/QEMU)
b. The instance storage location, either on local ephemeral disks or in a Ceph RBD pool.
    i. If considering local ephemeral storage, there should be sufficient storage capacity in /var/lib/nova/ to house the ephemeral root disks
c. other considerations if applicable - such as appropriate drivers for network and storage, etc for optimal performance.



Logical instance diagram 3

NOTE: Here the storage shown is RBD storage

Over Commitment Ratio

OpenStack's default over-subscription ratio (OSR) of CPU is 16:1 and Memory is 1.5:1. For Hadoop workloads we recommend setting the CPU OSR to 1:1 and Memory OSR to 1:1. Do not over-commit either of the resources. Hadoop workloads are very CPU and memory heavy, besides being IO and Network intensive; they will push the boundaries on all the subcomponents of your infrastructure.

Set the following in /etc/nova/nova.conf on all nodes running Nova-compute --

```
cpu_allocation_ratio = 1
ram_allocation_ratio = 1
```

Instance Types/Flavors

Red Hat OSP 11 does not have instance flavors defined out of the box. Therefore, consider crafting some custom ones that make sense for Hadoop workloads.

We have provided some guidance towards reasonable flavors. These are dependent on the workloads being run on Cloudera EDH.

Instance Flavors Table

Name	RAM (MB)	Disk (GB)	Ephemeral (GB)	VCPUs
rbd-1xl	58,982	100	0	11
rbd-2xl	112,640	100	0	22
rbd-4xl	118,000	100	0	40

The number of vCPUs to allocate will depend on the number of cores per Socket and RAM available in the physical host. Based on the type of hardware, the flavor definitions will vary.

NOTE:

- The flavor configurations are provided here as guidelines. Depending on the use case, the customer should adjust the size of CPUs and Memory. Typically it is recommended to make the instances larger in size and along CPU socket boundaries. Memory sizes will be predicated by the number of applications and types of services that will be running in the cluster.
- The general guidance for CPU allocation is to maintain 1:1 HT core to vCPU ratio. Similarly for RAM, guidance is to maintain 1:1 Physical to Virtual Memory allocation ratio. However, 1-2 cores and about 16GB of RAM should be left reserved for the hypervisor OS.
- Customers are advised to work with their Cloudera Account teams to determine the best instance flavors applicable to their environments, based on their existing or proposed workloads.
    - It is a good idea to keep minimum supportable configurations in mind while defining these flavors. For instance, Cloudera's MPP component - Impala has a minimum requirement for 128GB, and ideally at least 256GB of RAM.

The root disk should be at least 100GB, preferably > 200GB, such that we have sufficient logging space in the "/var" mountpoint/directory.

Set the disks on the VM to use the NOOP io scheduler instead of the default CFQ.

Warning:

It is better to have larger root disks or mount a Cinder volume with sufficient storage capacity to handle multiple copies of the system and various CDH component logs under the /var/log mount point. Cloudera recommends larger root disks and separating /var/log to a dedicated mountpoint.

Guest Image Configuration

If using RBD-based images, the images need to be in raw format as opposed to qcow2. When creating the image, it is important to make sure that you use a node representative of the hardware on which the image is going to run. Also ensure that virtualization features such as Intel VT-x or AMD-V are enabled in the BIOS of the servers where Nova compute will run.

It is a good idea to keep some guest image best practices in mind when creating the images.

In our testing, we followed Red Hat's best practices for VM image tuning --

Following Red Hat best practices of enabling following properties (see references section for more details) --

- `hw_disk_bus='scsi'`
- `hw_scsi_model='virtio-scsi'`
- `hw_vif_multiqueue_enabled='true'`
- `hw_watchdog_action='reset'`
- `os_require_queisce='yes'`
- `hw_qemu_guest_agent='yes'`

These are set as property key-value pairs for a given image.

```
$ openstack image show 967a0ad0-1356-4489-b7c0-7150650ef338
+------------------+------------------------------------------------------------------------------
---| Field            | Value
|
+------------------+------------------------------------------------------------------------------
---
|
| name             | rhel-7.3-rbd-v3
|
| owner            | fcd81cbaa78d490f921a3799a4eaf5b2
|
| properties       |
direct_url='rbd://19c5717d-0db4-4977-835c-7fafabbcf155/images/967a0ad0-1356-4489-b7c0-7150650e
f338/snap', hw_disk_bus='scsi', hw_qemu_guest_agent='yes', hw_scsi_model='virtio-scsi',
|
|                  | hw_vif_multiqueue_enabled='true', hw_watchdog_action='reset',
os_require_queisce='yes'
|
+------------------+------------------------------------------------------------------------------
---
```

NOTE: Output truncated for better readability

Orchestration

We do not have any specific orchestration rules or recommendations for Hadoop instances. There is no benefit to migration, live migration, or storage migration of the instances when using HDFS with replication factor (RF) greater than 1. Our recommendation with this RA is to use HDFS with RF=2 (instead of default 3) along with the Hadoop Virtualization Extensions (HVE) in order to provide better data availability. That will naturally prevent the VMs from being mobile. Each VM will reside within a fixed nodegroup (see Enabling Hadoop Virtual Extensions for more details).

Red Hat OSP director provides automation for the OpenStack platform build (aka the Overcloud), and Red Hat OSP deployments in production are supported only when deployed with the Director.

In order to automate the deployment of Guest VMs and associated virtual infrastructure (such as networks, subnets, etc) as well as the application, various tools can be leveraged.

- The OpenStack ecosystem includes Heat, which allows for templating VMs and automating OSP infrastructure and guest VM deployment.
- There are guides available in the public domain that articulate how to leverage popular tools such as Ansible, Vagrant, Foreman, Chef, and Puppet to fully automate lifecycle management of OpenStack infrastructure as well.

For the Cloudera Enterprise application deployment, the Cloudera Manager API is a very popular option and all Cloudera build automation is done using the CM API. Some relevant URLs are provided in the References section of this document.

A more detailed discussion on this topic is out of scope for the current version of this document.

Storage

Ceph

Ceph is a distributed storage platform that can be implemented on commodity hardware. Ceph provides an object store, block storage as well as a distributed filesystem. For our reference architecture, we will focus on the block storage (RBD) component of Ceph.

The following table summarizes what we strongly recommend as minimum performance characteristics of the various storage components, when all VMs are simultaneously accessing all disks.

Performance Profile Table

Component	Response time/latency (ms)	Minimum Acceptable Throughput (MB/s) [Read or Write]	Recommended (MB/s)
Persistent (Cinder-based Ceph RBD) storage -- per RBD	< 50	30	70
Per Virtual Machine	N/A	120	280
Per Physical Hypervisor	N/A	480	1120

The table above articulates three levels of minimum criteria that Cloudera feels are needed for a reasonable customer experience. Using these parameters as a guideline, customers should be able to formulate minimum requirements, such that they design underlying infrastructure capable of achieving these minimums.

For example, if there is a need for a cluster that can yield a throughput of 5GB/s, the following layers of infrastructure need to be designed in order to achieve that --

- Ceph Backend
- OSP Compute Nodes
- Networking at the Ceph cluster level
- Networking at the OSP compute level
- Inter-networking between Ceph and OSP so there is no bottleneck and 5GB/s of throughput is attainable.

In such a scenario, the Ceph backend itself should be capable of > 5GB/s throughput. That could mean each ceph storage node having at least 2 NVMe SSDs capable of handling 800-900MB/s of sequential write (and usually reads are faster) and sufficient SATA spindles to provide the storage capacity needed.
It would also imply that each storage node have at least 2 x 10GbE NICs in link-aggregated bonded mode, to handle both the client traffic as well as replication traffic.

There should be at least 11 OSP compute nodes, each capable of handling 480MB/s of IO, therefore implying that they have at least two 10GbE NICs - one to handle the N-S IO traffic between OSP and Ceph, and another to handle the E-W traffic between the Cloudera cluster nodes themselves.

NOTE:

- Most NVMe SSDs provide ~ 2000MB/s Read throughput and ~ 1500MB/s Write throughput.

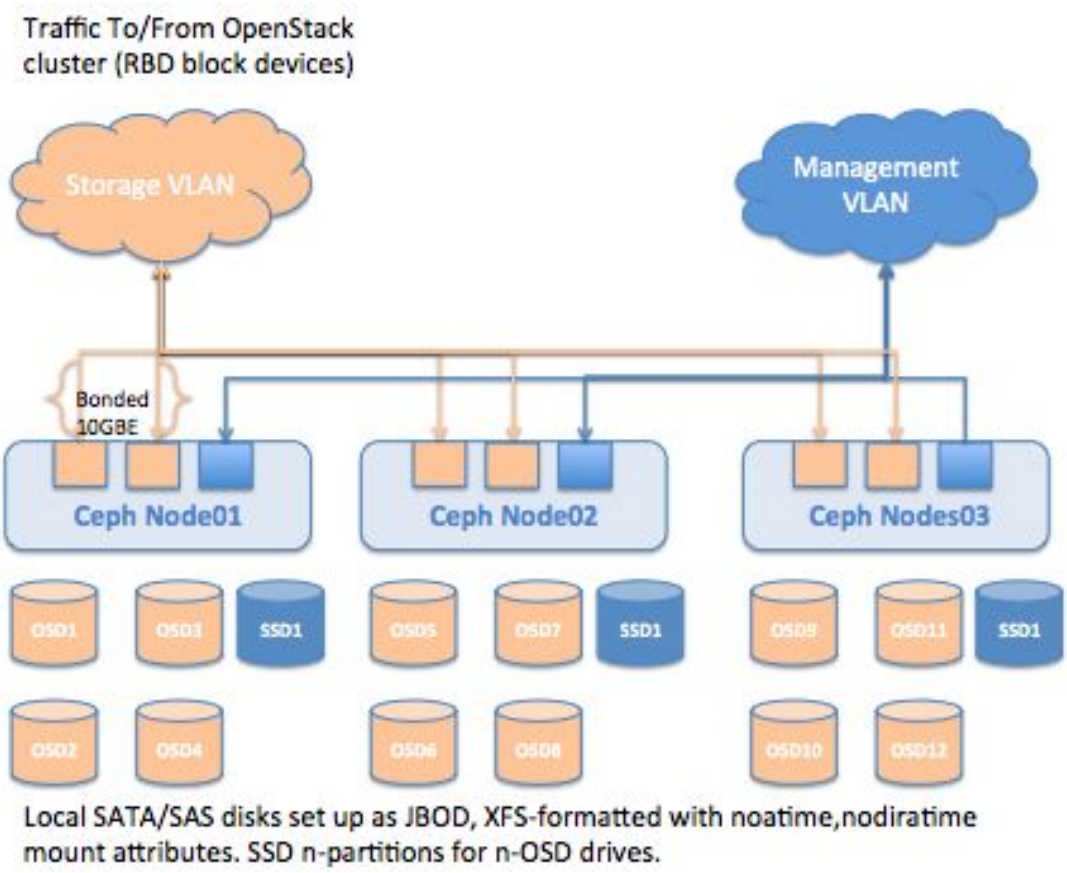
Validating the Ceph Backend Performance

In order to validate that the Ceph backend is adequately performant as well as scalable, we strongly advise that the following test methodology be applied in parallel from multiple client nodes to ensure that multiple clients accessing the Ceph cluster can experience deterministic and reliable throughput and latency responses.

We also strongly recommend re-validating backend performance each time additional Ceph nodes and/or client nodes are added to the environment.

The methodology provided by RedHat will be able to test whether the Ceph backend can provide the required throughput.

Once this is ascertained, then, a CDH cluster built on VMs, that have sufficient storage allocated and appropriate number of volumes provided, can be instantiated and standard MapReduce tests such as TeraGen, TeraSort and TeraValidate run against the cluster, to ascertain whether the CDH cluster is able to yield the required throughput as well.



Ceph Backend Logical Topology Diagram 4

Ceph Design Principles

The performance of the Ceph RBD backend is predicated on the number of SSDs and types of SSDs being used for the journal write acceleration. For instance, if you have a SATA SSD rated to do sequential writes of ~ 380MB/s you can use that to back 3-4 SATA OSD drives, considering we want to get 100-120MB/s sequential write throughput per SATA drive. So in a node with 24 SATA drives, you will need 5-6 SATA SSDs to provide best overall throughput.

If you use NVMe SSDs that are rated for much higher sequential write throughput of say ~ 900MB/s, you can use it to back 8-9 SATA OSD drives.

The following parameters should be considered in the Ceph configuration to allow for larger journal devices --

- Using block devices instead of directories implies that the entire block device gets used as a journal device. A single SSD can be partitioned, for example a 500GB drive can be partitioned into five 100GB partitions and each partition allocated as journal device for an OSD spinning drive.

- Adjust the OSD op threads parameter from default of 2 to larger value. This parameter helps the Ceph OSD daemon multi-thread servicing requests. For instance, in /etc/ceph/ceph.conf file, set : `osd op threads = 30`
- The Ceph journal filestore max sync interval determines the interval at which the journal quiesces writes and synchronizes the filesystem, which creates a consistent commit point. It can then free journal entries up to the commit point. It defaults to 5 seconds and increasing this can improve write and metadata performance. For example, in /etc/ceph/ceph.conf, set : `filestore max sync interval = 15`. For a more detailed understanding on this subject, read the RedHat Ceph Storage Configuration Reference.
- The effective throughput one can achieve from the VM's perspective is also dependent on the network throughput available between the compute nodes and the Ceph storage nodes. Ideally we should have no over-subscription between the OSP Compute and the Ceph Storage nodes.

Consider using SSDs similar to Intel's DC P4600 or DC P4800 series NVMe SSDs for best performance for Ceph Storage nodes.

NOTE:

- Cloudera recommended network architecture is Spine-Leaf topology between Ceph and Compute nodes.
- 10GbE NICs are a minimum.
- There should be no oversubscription between Ceph and OSP Compute nodes.

Following rules of thumb should be considered while designing the Ceph backend for CDH --

- Definitely consider using 3x replication on the Ceph RBD pools. It is to be noted however, that we should still strongly recommend keeping at least 2x replication in place in HDFS for higher availability. With Ceph, 3x replication allows inconsistencies found during integrity checking (scrubbing) to be resolved.
- Separate network for storage
- Ceph storage nodes should have 2x bandwidth as compared to compute nodes. So if compute nodes have 1 x 10gbe NIC for storage, Ceph nodes should have 2 x 10gbe NICs, bonded in link-aggregation mode (and not failover mode).
- For production workloads, have at least 8 storage nodes.
- RAID Controller considerations for the storage nodes - follow best practices. There is no RAID being used at the physical disk layer - just use disks as JBOD. Ceph's RADOS architecture will provide the fault-tolerance and redundancy required.
- Have nodes with 12-36 disks (higher density drives - 4TB at least).
- Consider having flash-based pools for local and intermediate data (OS, YARN local-dirs, impalad scratch dirs)
- Each OSD (disk) should should have 1 GHz of 1 core of CPU and 2GB of RAM (so if node has 24 disks, there should be 24 GHz of Compute (so 24 HT CPUs) and 48GB of RAM (so consider at least a server with 12 physical cores and 64GB of RAM for example).
- 1 NVMe SSD as a journal for every 9-8 OSD spindles or 1 SATA SSD for every 3-4 OSD spindles.
- Recommended filesystem for ceph storage devices is XFS. Follow RedHat's best practices for tuning.

NOTE:

- Please ensure HVE is enabled (see section on HVE) to ensure best data availability in the cluster (prevent data from becoming temporarily unavailable due to underlying infrastructure issues).
- This will also prevent replicas of HDFS blocks from being placed on the same physical hypervisor host.
- To ensure maximum data availability, customers might want to consider setting the HDFS Replication Factor to 3.

The Ceph cluster can be set up either as an independent cluster, or following Red Hat's instructions of setting up a Ceph cluster via OSP director. Setting it up independently gives us better control over the configuration.

Some key steps towards that are to configure the Ansible playbooks for Monitors and OSDs and then executing `ansible-playbook run` to set the cluster up.

Red Hat provides a PG (Placement Group) Calculator to help properly identify the number of placement groups as well as download the commands that should be run to create the storage pools. This tool also provides a good explanation on the relationship between OSDs and PGs per OSD.

After running numbers through the calculator, a commands text file is emitted, that can be then executed as root on the ceph cluster.

```
## Note:
## The 'while' loops below pause between pools to allow all PGs to be created.
## This is a safety mechanism to prevent saturating the Monitor nodes.
##--------------------------------------------------------------------

ceph osd pool create cinder-backup 2048
ceph osd pool set cinder-backup size 3
while [ $(ceph -s | grep creating -c) -gt 0 ]; do echo -n .;sleep 1; done

ceph osd pool create cinder-volumes 4096
ceph osd pool set cinder-volumes size 3
while [ $(ceph -s | grep creating -c) -gt 0 ]; do echo -n .;sleep 1; done

ceph osd pool create ephemeral-vms 2048
ceph osd pool set ephemeral-vms size 3
while [ $(ceph -s | grep creating -c) -gt 0 ]; do echo -n .;sleep 1; done

ceph osd pool create glance-images 512
ceph osd pool set glance-images size 3
while [ $(ceph -s | grep creating -c) -gt 0 ]; do echo -n .;sleep 1; done
```

WARNING:

- It is possible to increase the PG count, but you can NEVER decrease without destroying and recreating the pool.
- Avoid increasing the PG count of a pool -- especially for production clusters, if possible. The goal is to ensure that the PG per OSD ratio does not go high enough to cause problems during Recovery and/or Backfill operations

Integrating Ceph and OSP

Once the ceph backend is configured, the OpenStack controller(s) and the Compute hosts need to be configured to work with the ceph backend. The references section of this document has (links to) instructions on how glance and cinder on the controller node needs to be configured to support ceph. Also provided are details on how to configure the controller and nova compute nodes as ceph clients to handle ceph.

Towards that end, the following things need to be highlighted --

1) Packages needed to set up the ceph clients - python-rbd and ceph-common.
    a) yum install -y python-rbd ceph-common
2) Configuration files/keyring files needed for ceph client configuration
    a. /etc/ceph/ceph.conf
    b. /etc/ceph/*.keyring files (that were configured for cephx authentication to work for specific user ids, or the admin id as per documentation)
    c. the secret.xml and the ceph.client.<user>.key files that will be needed to set up libvirt security on the compute nodes.
    d. /etc/nova/nova.conf file updated with support for rbd.

       The file will contain similar entries under the [libvirt] section --

```
images_type = rbd
images_rbd_pool = vmpool
images_rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_user = cinder
rbd_secret_uuid = 15d2e1d8-502e-4c4e-87db-c0b8930473cf
```

    e. /etc/glance/glance-api.conf

```
[glance_store]
stores = rbd,file
rbd_store_pool = imgpool
rbd_store_user = glance
rbd_store_ceph_conf = /etc/ceph/ceph.conf
rbd_store_chunk_size = 8
```

    f. /etc/cinder/cinder.conf

```
[default]
```

```
enabled_backends=lvm,rbd
backup_driver = cinder.backup.drivers.ceph
backup_ceph_conf = /etc/ceph/ceph.conf
# This ideally is cinder-backup
backup_ceph_user = cinder-backup
backup_ceph_chunk_size = 134217728
backup_ceph_pool = bkuppool
backup_ceph_stripe_unit = 0
backup_ceph_stripe_count = 0
restore_discard_excess_bytes = true
```

Create a new section called [rbd]

```
[rbd]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
volume_backend_name=rbd
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
glance_api_version = 2
# this ideally is cinder
rbd_user = cinder
rbd_secret_uuid = 15d2e1d8-502e-4c4e-87db-c0b8930473cf
```

After making the configuration updates, the respective OpenStack services need to be restarted. In addition to the configuration file changes listed above, run the following from the Controller node --

```
# source ~/keystonerc_admin

[root@vb0324 ~(keystone_admin)]# cinder type-create rbd
[root@vb0324 ~(keystone_admin)]# cinder type-key rbd set
volume_backend_name=rbd
[root@vb0324 ~(keystone_admin)]# cinder type-list
+--------------------------------------+-------+
|                  ID                  |  Name |
+--------------------------------------+-------+
| 525f3c08-0180-4612-bf91-6c94aef6da67 | iscsi |
| aa5627ec-bd28-4c13-8ce9-185921523219 |  lvm  |
| b853fc32-bab8-4d67-9d7f-c855c9d19161 |  rbd  |
+--------------------------------------+-------+
```

Once this is done, you should be able to create rbd volumes via cinder.
This will allow you to attach the volumes to your VMs, format them in an FS format of your choosing, mount them and run various performance benchmarking tests.

After your VMs have been instantiated, you can set up CDH 5.x just like you would a bare-metal cluster. The Cloudera section below has relevant details and caveats for virtualized deployment of the platform.

The standard benchmarking tool - teragen/terasort can be used to baseline performance of your virtualized CDH cluster and help optimize both at the OS level as well at the OpenStack and infrastructure level.

Optimization details are out of scope of the current iteration of this document, but similar considerations as performance tuning for bare-metal clusters may be employed.

Setting up QOS for Deterministic IO performance

Setting up concrete QOS rules ensure that IO patterns are evenly distributed among volumes presented to the various VMs in a given cluster.

Set the QOS rules as follows (this is an example, actual values will have to be set experimentally, depending on the actual workload) --

```
$ openstack volume qos create --property read_bytes_sec=209715200 \
--property write_bytes_sec=209715200 --property read_iops_sec=800 \
--property write_iops_sec=400 --consumer front-end datanode
```

This sets up the QOS rule. The rule is then associated with a cinder volume type.

```
$ openstack volume qos associate datanode rbd
```

This tells cinder that each volume of type “rbd” needs to be set up with the QOS properties defined in the QOS type “datanode”.

The QOS rules show up in the VM xml config as follows (you can generate that by running `virsh dumpxml <domain id>`) --

```
<disk type='network' device='disk'>
    <driver name='qemu' type='raw' cache='none'/>
    <auth username='xxxx'>
      <secret type='ceph' uuid='6f03c880-4ca6-4d6b-997b-e82ad6ed8055'/>
    </auth>
    <source protocol='rbd' name='rbd/volume-dfb3a0d6-a150-4b7f-86ce-776280da47d2'>
      <host name='10.17.204.23' port='6789'/>
      <host name='10.17.204.24' port='6789'/>
      <host name='10.17.206.11' port='6789'/>
    </source>
    <backingStore/>
    <target dev='sdc' bus='scsi'/>
```

```
  <iotune>
    <read_bytes_sec>209715200</read_bytes_sec>
    <write_bytes_sec>209715200</write_bytes_sec>
    <read_iops_sec>800</read_iops_sec>
    <write_iops_sec>400</write_iops_sec>
  </iotune>
  <serial>dfb3a0d6-a150-4b7f-86ce-776280da47d2</serial>
  <alias name='scsi0-0-0-2'/>
  <address type='drive' controller='0' bus='0' target='0' unit='2'/>
</disk>
```

NOTE:

- Any existing volumes associated with the cinder volume type that are already attached to VMs will not exhibit these QOS parameters. They will have to be detached and reattached in order for them to work.

Cloudera Software stack

Guidelines for installing the Cloudera stack on this platform are nearly identical to those for bare-metal. This is addressed in Cloudera's Product Documentation.

Do not allow more than one replica of an HDFS block on any particular physical node. This is enabled with configuring the Hadoop Virtualization Extensions (HVE).

The minimum requirements to build out the cluster are:

- 3x Master Nodes (VMs)
- 5x Worker Nodes (VMs)

The Worker Node count depends on the size of HDFS storage to deploy. The following table identifies service roles for different node types.

Follow the guidelines in the Compute section of this document to provision instance types.

- Ensure that CPU and Memory resources are not overcommitted while provisioning these node instances on the virtualized infrastructure.
- Automated movement of VMs must be disabled. There should be no Migration/Live Migration of VMs allowed in this deployment model.
- Master Nodes should be provisioned on disparate physical hardware, if possible in separate racks or configure anti-affinity between the Master node VMs. In order to do this, a server-group with the 'anti-affinity' policy needs to be created and the master VM instances be booted with the group passed as a hint (`--hint group=<UUID>`).

Logical Component Layout Tables

General Component Layout

Service/Role	Master Node	Master Node2	Master Node3	Worker Node 1..n
ZooKeeper	ZK	ZK	ZK	
HDFS	NN,QJN	NN,QJN	QJN	Data Node
Kudu	Master	Master	Master	Tablet Server
YARN	RM	RM	History Server	Node Manager
Hive			MetaStore, WebHCat, HiveServer2	
Management(misc)	Oozie, CMA	Oozie, CM (standby), Management Services (standby), CMA	Oozie, CM (active), Management Services (active),CMA	CMA
Database		Standby DB	Active DB	
Navigator			Navigator	
HUE			HUE (place on Edge node if exists)	
SEARCH				Solr
Spark	History Server			Runs on YARN
impala			statestore, catalog	impalad
HBASE	HMaster	HMaster	HMaster	Region Servers
Sentry		Sentry	Sentry	

Additional Services Component Layout

Service/Role	Kafka ZooKeeper (1..5)	KeyTrustee Server 1	KeyTrustee Server 2	KMS Proxy 1	KMS Proxy 2	Kafka Brokers	Edge Node (1 per 20 Workers)
Management(misc)	CMA	CMA	CMA	CMA	CMA	CMA	CMA
Kafka (Separate Cluster if doing > 100,000 transactions/second, 3-5 ZK nodes in separate ZK ensemble)	ZK					Kafka Broker	
KeyTrustee Server (Separate Cluster)		KTS (active)	KTS (standby)				
KMS (dedicated Nodes)				KMS (active)	KMS (standby)		
Flume							Flume Agent

NOTE:

- For the various abbreviations used in these tables, please refer to the Glossary of Terms section.

Instance-type Table

Instance Role	Instance Type/Flavor	Comments
Master Nodes	rbd-2xl	The master instances will house components of the cloudera stack as shown in the tables above
Worker Nodes	rbd-4xl	These will have sufficient Compute resources.

NOTE:

- It is advisable to work with a Cloudera SE to determine appropriate instance sizes based on the workloads as well physical resource parameters.

Enabling Hadoop Virtualization Extensions (HVE)

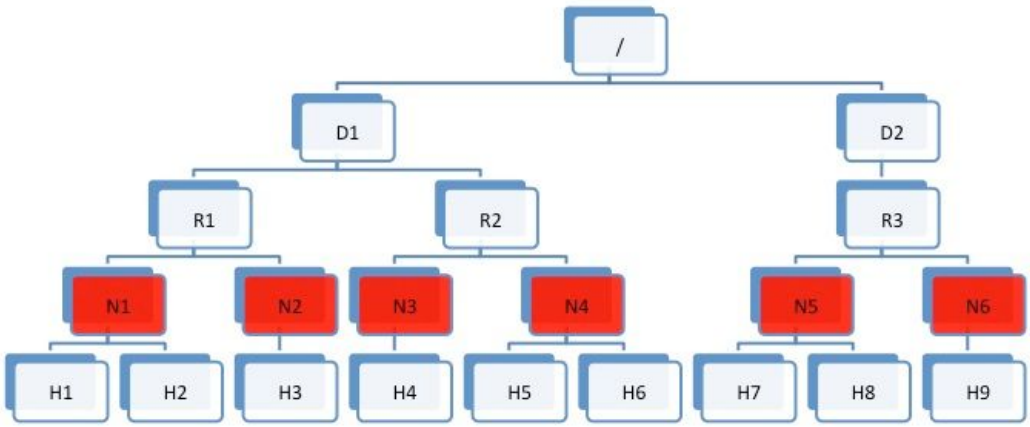
NOTE: While this document refers to hypervisors and virtual machines, this methodology is applicable to all any scenario where a “shared” something is involved. This is a strategy to help mitigate single points of failure, be it a shared power supply, a shared chassis, a shared storage tray, and so on.

Referring to the HDFS-side HVE JIRA (HADOOP-8468), the following are considerations for HVE:

1. VMs on the same physical host are affected by the same hardware failure. In order to match the reliability of a physical deployment, replication of data across two virtual machines on the same host should be avoided.
2. The network between VMs on the same physical host has higher throughput and lower latency and does not consume any physical switch bandwidth.

Thus, we propose to make Hadoop network topology extendable and introduce a new level in the hierarchical topology, a node group level, which maps well onto an infrastructure that is based on a virtualized environment.

The following diagram illustrates the addition of a new layer of abstraction (in red) called NodeGroups. The NodeGroups represent the physical hypervisor on which the nodes (VMs) reside.



HVE

Topology diagram 5

All VMs under the same node group run on the same physical host. With awareness of the node group layer, HVE refines the following policies for Hadoop on virtualization:

Replica Placement Policy

- No duplicated replicas are on the same node or nodes under the same node group.
- First replica is on the local node or local node group of the writer.
- Second replica is on a remote rack of the first replica.
- Third replica is on the same rack as the second replica.
- The remaining replicas are located randomly across rack and node group for minimum restriction.

Replica Choosing Policy

The HDFS client obtains a list of replicas for a specific block sorted by distance, from nearest to farthest: local node, local node group, local rack, off rack.

Balancer Policy

- At the node level, the target and source for balancing follows this sequence: local node group, local rack, off rack.
- At the block level, a replica block is not a good candidate for balancing between source and target node if another replica is on the target node or on the same node group of the target node.

HVE typically supports failure and locality topologies defined from the perspective of virtualization. However, you can use the new extensions to support other failure and locality changes, such as those relating to power supplies, arbitrary sets of physical servers, or collections of servers from the same hardware purchase cycle.

Using Cloudera Manager, configure the following in safety valves:

- HDFS
  - hdfs core-site.xml (Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml/core_site_safety_valve):

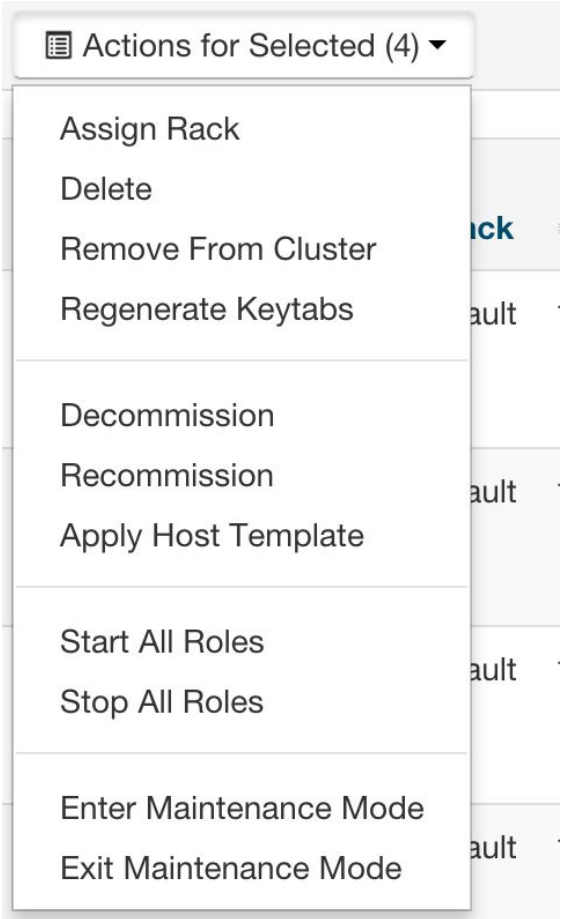
```
<property>
     <name>net.topology.impl</name>
       <value>org.apache.hadoop.net.NetworkTopologyWithNodeGroup<
       /value>
</property>
<property>
     <name>net.topology.nodegroup.aware</name>
     <value>true</value>
</property>
<property>
     <name>dfs.block.replicator.classname</name>
       <value>org.apache.hadoop.hdfs.server.blockmanagement.Block
       PlacementPolicyWithNodeGroup</value>
</property>
```

- In mapred-site.xml, add the following properties and values (this is set using the HDFS Replication Advanced configuration snippet (safety valve) mapred-site.xml (mapreduce_service_replication_config_safety_valve)):

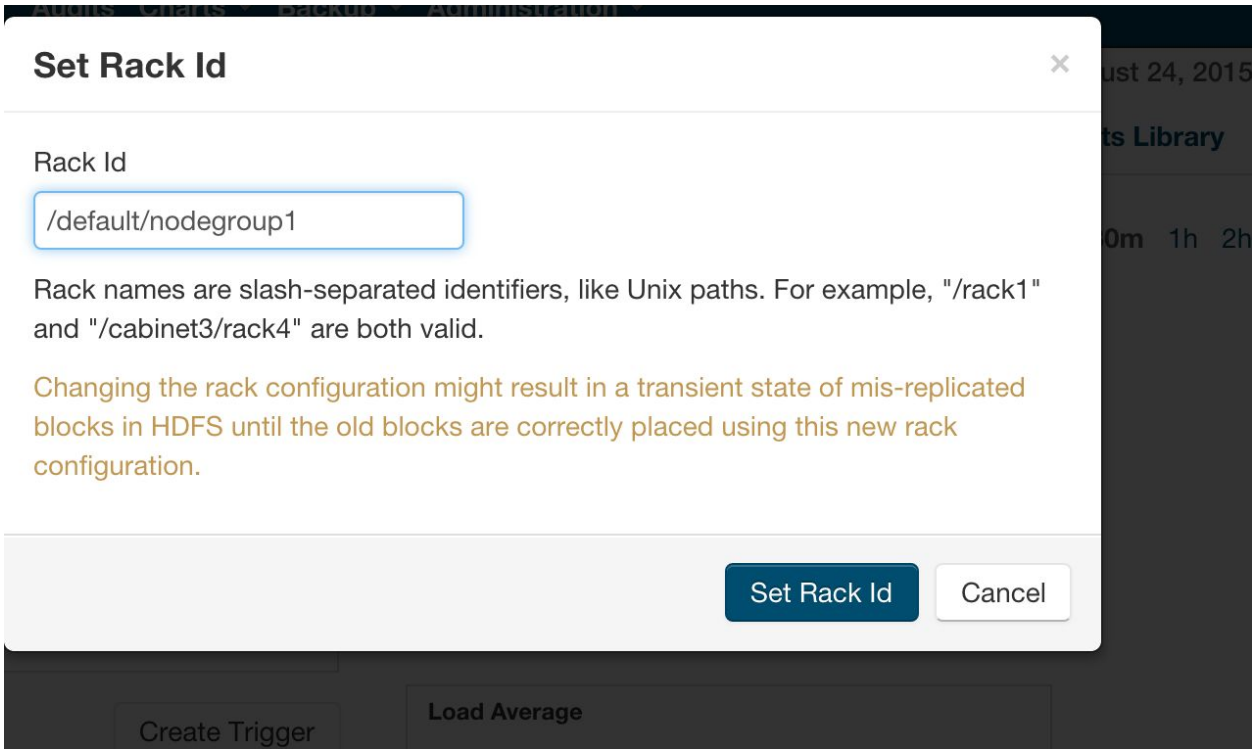
```
<property>
        <name>mapred.jobtracker.nodegroup.aware</name>
        <value>true</value>
</property>
<property>
        <name>mapred.task.cache.levels </name>
        <value>3</value>
</property>
```

Establish the Topology:

Follow the instructions to set rack location of hosts here -- [Specifying Racks for Hosts].
Select all multiple hosts from the Hosts page and then assign rack.



Alternately, In Cloudera manager, you can specify the topology by going into the Hosts/Status page and editing the Rack assignment from /default to /default/nodegroup<id>.



Instructions

The following safety valves need to be applied --

1. HDFS -- Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml
2. YARN - YARN Service MapReduce Advanced Configuration Snippet (Safety Valve) - mapred.xml

Follow this sequence of actions to enable HVE --

- Apply the safety valves
- Assign the rack topology to the nodes
- Stop the cluster
- Deploy client config
- Start ZooKeeper
- Start HDFS
- Start all other services

References

1. Product Documentation for Red Hat OpenStack Platform
2. SR-IOV on Red Hat OSP
3. Red Hat OSP Networking Guide
4. Understanding Red Hat OSP High Availability
5. RedHat Virtualization tuning and Optimization Guide
6. RedHat Ceph Storage 2 Installation Guide
7. Cloudera Product Documentation
8. Cloudera Manager API Documentation
9. Cloudera Manager CM API Python End-to-end Guide
10. HVE - HADOOP-8468

Glossary of Terms

Term	Description
CDH	Cloudera Distributed Hadoop
Ceph	An open-source distributed storage framework (RADOS or Reliable Autonomic Distributed Object Store) that allows a network of commodity hardware to be turned into a shared, distributed storage platform. Ceph natively provides Block Storage (RBD or RADOS Block Device) that are striped across the entire storage cluster, an Object Store as well as a shared filesystem.
Cinder	Cinder is the Storage provisioning and management component of the OpenStack framework.
CM	Cloudera Manager
CMA	Cloudera Manager Agent
DataNode	Worker nodes of the cluster to which the HDFS data is written.
Cloudera EDH	Cloudera Enterprise Data Hub
Ephemeral storage	Storage devices that are locally attached to Nova instances. They persist guest operating system reboots, but are removed when a Nova instance is terminated.
Glance	This is the imaging services component of the OpenStack framework. This maintains images that are used to instantiate Virtual machines in an OpenStack cluster.
HBA	Host bus adapter. An I/O controller that is used to interface a host with storage devices.
HDD	Hard disk drive.

HDFS	Hadoop Distributed File System.
HA/High Availability	Configuration that addresses availability issues in a cluster. In a standard configuration, the NameNode is a single point of failure (SPOF). Each cluster has a single NameNode, and if that machine or process became unavailable, the cluster as a whole is unavailable until the NameNode is either restarted or brought up on a new host. The secondary NameNode does not provide failover capability. High availability enables running two NameNodes in the same cluster: the active NameNode and the standby NameNode. The standby NameNode allows a fast failover to a new NameNode in case of machine crash or planned maintenance.
HVE	Hadoop Virtualization Extensions - this is what enables proper placement of data blocks and scheduling of YARN jobs in a Virtualized Environment wherein, multiple copies of a single block of data or YARN jobs (don't get placed/scheduled on VMs that reside on the same hypervisor host). The YARN component of HVE is still work in progress and won't be supported in CDH 5.4 (YARN-18). The HDFS component is supported in CDH 5.4.
Ironic	Ironic is an OpenStack project which provisions bare metal (as opposed to virtual) machines by leveraging common technologies such as PXE boot and IPMI to cover a wide range of hardware, while supporting pluggable drivers to allow vendor-specific functionality to be added
JBOD	Just a Bunch of Disks (this is in contrast to Disks configured via software or hardware RAID with striping and redundancy mechanisms for data protection)
JHS/Job History Server	Process that archives job metrics and metadata. One per cluster.
LUN	Logical unit number. Logical units allocated from a storage array to a host. This looks like a SCSI

	disk to the host, but it is only a logical volume on the storage array side.
NN/NameNode	The metadata master of HDFS essential for the integrity and proper functioning of the distributed filesystem.
NIC	Network interface card.
Nova	The Compute Scheduling and resource management component of OpenStack.
NodeManager	The process that starts application processes and manages resources on the DataNodes.
Neutron	Neutron is the Network management layer of OpenStack - it incorporates/supports SDN (Software Defined Networking) features, advanced overlay features such as VxLAN and GRE tunneling, and provides a plugin-in architecture to enable support for different technologies.
Open vSwitch/OVS	Open vSwitch is a production quality, multilayer virtual switch licensed under the open source Apache 2.0 license. It is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag). In addition, it is designed to support distribution across multiple physical servers.
PDU	Power distribution unit.
QJM **QJN**	Quorum Journal Manager. Provides a fencing mechanism for high availability in a Hadoop cluster. This service is used to distribute HDFS edit logs to multiple hosts (at least three are required) from the active NameNode. The standby NameNode reads the edits from the JournalNodes and constantly applies them to its own namespace. In case of a failover, the standby NameNode applies all of the edits from the

	JournalNodes before promoting itself to the active state. Quorum JournalNodes. Nodes on which the journal services are installed.
RADOS	Reliable Autonomic Distributed Object Store
RBD	RADOS Block Device
RM	ResourceManager. The resource management component of YARN. This initiates application startup and controls scheduling on the DataNodes of the cluster (one instance per cluster).
SAN	Storage area network.
Sahara	Sahara project aims to provide users with simple means to provision a Hadoop cluster at OpenStack by specifying several parameters like Hadoop version, cluster topology, nodes hardware details and a few more.
SPOF	Single Point of Failure
ToR	Top of rack.
VM/instance	Virtual machine.
ZK/ZooKeeper	ZooKeeper. A centralized service for maintaining configuration information, naming, and providing distributed synchronization and group services.