

Oracle Private Cloud Appliance X8-2 Configuration for Cloudera Infrastructure

Provisioning Oracle Cloud Appliance X8-2
compute, networking, and storage resources
for Cloudera

Purpose statement

This document illustrates the creation of a high-performance compute, networking, and storage infrastructure for a PCA based Cloudera environment.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Table of contents

Purpose statement	2
Disclaimer	2
Introduction	5
Oracle Private Cloud Appliance X8-2	5
Software & Hardware Requirements	5
Creating Virtual Machines for Cloudera Infrastructure	6
Planning CPU Requirements	7
Planning Memory Requirements	7
Planning Network Requirements	7
Building the Cloudera Infrastructure	9
Defining Networks	9
ETH0 Network	9
ETH1 Network	9
ETH2-ETH5 Networks	10
Defining VMs	14
Customize Oracle Linux 7	17
Initial Customization of Oracle Linux 7 iSCSI Storage Services for Cloudera Data	19
Create iSCSI Definitions and iSCSI LUNs on the Internal ZFS Storage Appliance	22
Discover iSCSI LUNs, Format Devices, Create Filesystems on each Cloudera Node	26
Proceed with Cloudera Installation	32
Appendix A – Accessing the Administrative Interface for the PCA X8-2 Internal Oracle ZFS Storage Appliance	33
Appendix B – Add Additional iSCSI LUNs to a Cloudera Node	34

List of images

Figure 1: 'pca-admin' commands to create storage networks	10
Figure 2: 'pca-admin' commands to add storage networks to PCA X8-2 compute nodes	10
Figure 3: Displaying a storage network after creation	11
Figure 4: OVM Manager Networks Pane showing storage networks assigned to Virtual Machines Role ONLY	12
Figure 5: Example Network configuration on an internal ZFS ZS7-2 MR within an PCA X8-2	13
Figure 6: Oracle VM Manager Create VM Wizard – page 1	14
Figure 7: Oracle VM Manager Create VM Wizard – page 2	15
Figure 8: Oracle VM Manager Create VM Wizard – page 3 – create vDisk	15

Figure 9: Oracle VM Manager Create VM Wizard – page 3 – select CDROM / ISO	16
Figure 10: Oracle VM Manager Create VM Wizard – page 4	16
Figure 11: : ZFS Storage Appliance ZS7-2 MR iSCSI Initiator setup	22
Figure 12: ZFS Storage Appliance ZS7-2 MR iSCSI Initiator Group setup	23
Figure 13 : ZFS Storage Appliance ZS7-2 MR ZFS Project create	24
Figure 14:: ZFS Storage Appliance ZS7-2 MR ZFS Project setup	25
Figure 15 : ZFS Storage Appliance ZS7-2 MR iSCSI LUN create	26
Figure 16: ZFS storage appliance ZS7-2 MR iSCSI target and target group	27
Figure 17: List of Partition after Creation	30
Figure 18: Filesystem creation	30
Figure 19: Cloudera filesystem mounts	31
Figure 20: Cloudera Configuration	32
Figure 21: Brower Proxy Settings	33

List of tables

Table 1: Cloudera Configuration Parameters	Error! Bookmark not defined.
--	-------------------------------------

Introduction

Oracle Private Cloud Appliance ("PCA") is an integrated infrastructure system engineered to enable rapid deployment of converged compute, network, and storage technologies for hosting applications or workloads on a guest operating system ("OS").

PCA X8-2 provides a fast, flexible virtual and physical infrastructure for Cloudera Big Data workloads. Cloudera cluster nodes can be assigned the exact amount of CPU, memory, network, and storage resources necessary for optimal performance and availability.

In this document, we will primarily illustrate how to provision the resources necessary for a Cloudera cluster. The specific sizing of virtual machines, memory, networking and storage required for any given customer's Cloudera cluster will depend on many factors beyond the scope of this document. General recommendations will be given here, and specific values are to be determined by the customer based on their projected workloads using guidance from Cloudera.

Oracle Private Cloud Appliance X8-2

Oracle Private Cloud Appliance is an easy-to-deploy, "turnkey" converged infrastructure solution that integrates compute, network, and storage resources in a software-defined fabric.

Oracle Private Cloud Appliance rack consists of the following main hardware components:

- **Compute Nodes.** Compute nodes include Oracle Server X8-2 systems powered by two Intel® Xeon® Processor with 24 cores per socket. The X8-2 compute nodes can be ordered in three different memory configurations – 384GB, 768GB and 1.5 TB. Each compute node runs Oracle VM Server for x86 to provide server virtualization. Compute nodes may be added or removed from the Oracle Private Cloud Appliance configurations without any downtime. A Private Cloud Appliance rack can support up to 1,200 compute cores.
- **Switches.** Ethernet switches used for the data network and management network in a Private Cloud Appliance. The different types of switches used are:
 - **Leaf Switches** - (2) 36 port 100GbE switches used for high-speed internal communication between the internal hardware components (Compute Nodes, system disk, management servers) in a Private Cloud Appliance solution
 - **Spine Switches** - (2) 36 port 100GbE switches used for high-speed communication between the Private Cloud Appliance and other Engineered Systems, storage or the data center network. The Spine switches form the backbone of the network and perform routing tasks.
 - **Management Switch** - (1) 48 port switch used to provide easy management of all internal hardware components (Compute Nodes, system disk, fabric interconnects, management servers) in a Private Cloud Appliance. High speed low latency SDN is implemented on top of 100GbE leaf and spine switches. These offer 100GbE connectivity for all communication between internal-rack components and allow for flexible 10/25/40 or 100 GbE connectivity to customer datacenter.
- **Integrated Storage.** Oracle Private Cloud Appliance features a fully integrated, enterprise-grade Oracle ZFS Storage Appliance ZS7-2 MR ("ZFSSA") for providing extreme performance and superior efficiency required by demanding enterprise applications running in VMs. This storage subsystem is designed to be fully redundant for maximum fault tolerance and serviceability in production. The Oracle Private Cloud Appliance X8-2 storage subsystem is loaded with high-performance DIMM and flash memory for optimal read/write performance under the most demanding file storage workloads. The storage capacity of Oracle Private Cloud Appliance X8-2 can be expanded beyond the initial configuration by adding storage trays. Storage can also be expanded by adding data center racks containing external Oracle ZFS Storage Appliances.

Software & Hardware Requirements

The following components are considered mandatory to this solution: -

- PCA X8-2 or newer hardware revision, with Ethernet based networking
- PCA 2.4.3 or newer PCA X8 software release
- Oracle Linux 7
- OVM Manager and pca-admin facilities available in the PCA with admin/root access

Creating the infrastructure for a Cloudera cluster involves multiple steps. After the infrastructure is created, the Cloudera software must be installed and configured.

Because the installation and configuration of the Cloudera software is very customer and workload dependent, those activities are beyond the scope of this document. This document will show how to

1. Create VMs for each Cloudera node
2. Create networks and add network resources to the VMs
3. Create storage resources and assign them to the VMs
4. Configure the Linux OS to prepare it for installation of Cloudera software.

Creating Virtual Machines for Cloudera Infrastructure

The PCA X8-2 software environment is based on Oracle VM (“OVM”), a Xen-based virtualization engine. PCA X8-2 has two management nodes and a variable number of compute nodes, as well as a ZFS Storage Appliance ZS7-2 MR (“ZFSSA”), and a Cisco-based 100GbE network.

Management nodes are configured in an HA environment and run the OVM Manager, which provides web-based, CLI, and REST interfaces to manage and control the OVM virtual environments.

Compute nodes run the Oracle VM Server (“OVS”) software that allows virtual machines to be created, managed, and controlled on an individual compute node.

The OVM Manager communicates with the OVS on each compute node. In addition to the OVM Manager, there is a command-line facility called pca-admin, which is used from the management nodes to configure storage and network resources as well as manage other aspects of the PCA X8-2.

Optionally, Oracle Enterprise Manager can be used to assist in managing and monitoring the PCA X8-2.

PCA X8-2 Compute Nodes have two Intel® Xeon® Processors providing 24 cores per socket. The X8-2 Compute Nodes can be ordered in three different memory configurations – 384GB, 768GB and 1.5 TB. PCA X8-2 Compute Nodes are connected to each other, and to the management nodes, over 100GbE links. Private internal networks can be created to connect together VMs running on the same compute node or on other compute nodes. Storage networks can be created to connect to the internal Oracle ZFS Storage Appliance. External networks can be created to connect to the customer data center network using the Cisco 100GbE spine switches at speeds of 10GbE, 25GbE, 40GbE and 100GbE. External storage or other systems are connected to PCA X8-2-resident VMs by using an external network definition. Exadata database machines can be directly connected to the PCA X8-2 through the PCA X8-2 spine switches.

This document assumes that the internal ZFSSA storage will be used for Cloudera HDFS, and all Cloudera nodes and the necessary databases are resident in the PCA X8-2.

Before creating the VMs, consult Cloudera documentation or use past experience to determine how many nodes will reside in the cluster, including the Cloudera Manager nodes as well as Cloudera worker nodes. Size the nodes appropriately as to the number of CPU's, the amount of memory, and the amount of storage required for the operating system, the Cloudera software roles, and the HDFS storage. The distribution of Cloudera workload across a cluster is largely determined by the Cloudera roles that are assigned to each node. The roles assigned to any individual node in a cluster will determine what CPU, memory, network, and storage resources are needed for

that node. The PCA X8-2 is very flexible in allocation of these resources. Resources can be easily added or removed from a VM after it is defined.

Planning CPU Requirements

24 vCPUs per Cloudera node VM is a good general starting point, and the Cloudera Manager dashboard can help fine tune this number, as it has a dashboard element showing both overall CPU utilization for a cluster as well as CPU utilization for individual nodes. Refer to the Oracle solution brief: “Optimizing Oracle VM Server for x86 Performance” for an in-depth description of all of the considerations around CPU allocation within OVM on PCA. Each PCA X8-2 Compute Node provides 96 vCPUs across 48 cores provided by two physical sockets (24 cores per physical socket). Two Cloudera nodes per PCA X8-2 Compute Node, each with 24 vCPUs, will encourage OVM to assign each vCPU to its own core for maximum performance, but more than two VMs can be assigned to a PCA X8-2 Compute node. The number of VMs assigned to a PCA X8-2 Compute Node will vary depending on VM size, and the amount of memory physically installed in each PCA X8-2 Compute Node. The Cloudera roles assigned to each Cloudera node will dictate the VM size necessary. Always assign an even number of vCPUs to each VM. The virtual machines should be in PVHVM mode. The Cloudera VMs do not have to be the exclusive workloads on the PCA X8 Compute Nodes; CPU priorities can be adjusted for lower priority VMs running other workloads; any additional VMs on a PCA Compute Node will require that CPU, memory and network resources be assigned to the non-Cloudera VMs, and the user must maintain awareness of **all** workloads running in the PCA when resolving issues. The Cloudera performance dashboards will not see non-Cloudera workloads running on shared PCA Compute Nodes or storage.

Planning Memory Requirements

The amount of memory assigned to each Cloudera node depends on the Cloudera roles assigned to each node. During Cloudera configuration, roles are assigned to each node in the cluster, and within each role are dozens, or even hundreds of configuration options which can affect the amount of RAM needed to service the role. Different workloads will also determine the memory requirements, for example, Spark workloads are very memory intensive. The aggregate of the operating system requirements, the Cloudera role requirements, and workloads, will determine the memory requirements for an individual VM. The amount of RAM installed in the PCA X8-2 Compute Node will also limit how much RAM can be assigned to the VMs running on that PCA X8-2 Compute Node. Memory can be added or removed from a VM very easily, but the VM must be stopped before adding or removing memory. In an environment exclusively running Cloudera, start by proportioning the available RAM on the PCA X8-2 Compute Nodes evenly to each VM, keeping in mind that approximately 32GB per PCA X8-2 Compute Node will be required for use by the underlying OVS hypervisor. Use the Cloudera dashboards to track the amount of memory used by each node. If a particular Cloudera node is not using all of its memory during peak workloads, and other nodes are memory constrained, reassign memory from the lower utilization nodes to the higher utilization nodes while those nodes are down. VMs may be migrated from one PCA Compute Node to another, and this is another way to balance memory utilization as well as CPU utilization. Avoid memory swapping, as this severely degrades performance.

Planning Network Requirements

Cloudera uses network connections between each node to distribute data for processing, and for data protection. Having a very fast network between each Cloudera node is critical for maintaining cluster performance. Each PCA X8-2 Compute Node has two, 100GbE connections into a 100GbE backbone network used for communication between VMs, as well as communication with the internal Oracle ZFS Storage Appliance, and for uplinks into the customer data center network. This tremendous bandwidth helps Cloudera workloads perform at their peak. The preferred configuration for Cloudera on PCA X8-2 uses an external network for user access and data transfer to and from the Cloudera nodes, a non-routable, private network for cluster communication, four Storage Networks for access to the internal Oracle ZFS Storage Appliance, and optionally, an additional external network for connection to Exadata or another external database server. If the entire Cloudera environment is contained within a PCA X8-2, the only cabling required will be to provide connection to the customer network. This connection to the customer network can be 10GbE for simple user access, or 25GbE, 40GbE, or 100GbE for fast data ingest and export. Refer to the Oracle Solution Brief : “Networking the Oracle Private Cloud Appliance and

Oracle Private Cloud at Customer” for details about connecting the PCA X8-2 to the customer datacenter networks.

Planning Storage Requirements

PCA X8-2 contains an Oracle ZFS Storage Appliance ZS7-2 MR for shared storage among all PCA X8-2 VMs, and for internal use in running the PCA X8-2. The ZS7-2 MR is used as network storage accessed through the 100GbE backbone network. There are two storage controllers for redundancy, and each has identical capabilities. These capabilities include 1TB of DRAM, dual SAS-3 HBAs, fast CPUs, and two, two-port 40GbE network interface cards (4x40GbE in each controller). The controllers are configured in an active/passive cluster, and the same performance can be achieved regardless of which controller is the current primary controller. The ZS7-2 MR supports encryption, compression, replication to remote or local destinations, snapshot/clone, Snapshot to Oracle Cloud Infrastructure, and many other features.

The default disk configuration of the ZS7-2 MR is a single, high capacity disk enclosure containing 20 data disks of 14TB each, two write flash disks, and two read flash disks. The data disks are configured in a mirrored pool for performance and redundancy, for a total of 110TB of usable capacity. It is not recommended to use the default disk enclosure for intensive Cloudera production workloads. Test workloads from a small Cloudera cluster can be accommodated with care, but storage for production clusters should be segregated onto additional storage enclosures attached to the PCA X8-2 internal ZFS Storage Appliance controllers. Depending on the number of PCA X8-2 Compute Nodes in the PCA X8-2 rack, up to four additional storage enclosures can be installed in the native PCA X8-2 rack, and installation is non-disruptive. Additional storage enclosures up to a total of 24 can be installed in an additional rack close to the PCA X8-2. High performance All Flash disk enclosures are an option, but standard high capacity storage enclosures with standard disks are capable of handling the Cloudera workload. The additional enclosures added to the PCA X8-2 to support the Cloudera workload should be configured into a single storage pool and the pool should have at least two write log devices. Write log devices are not necessary for every storage enclosure, but the overall storage pool should have at least two write log cache devices available. Read log cache devices are not necessary for a Cloudera workload.

Customers running extremely large and busy clusters should consider an external ZFS Storage Appliance ZS7-2. The internal ZFS Storage Appliance has been shown to stream up to 11GB/sec in a Cloudera environment. The variables of data compressibility and data access density and size make too difficult to give a rote answer as to “how big” a cluster can be in the internal ZFS Storage Appliance. If your aggregate write and read bandwidth requirements approach or exceed 10GB/sec, it is wise to consider an external ZFS Storage Appliance. Growth is inevitable and starting with a workload approaching the limits of the internal storage may cause issues later.

Because the PCA X8-2 100GbE networking can be extended outside of the base rack, the performance of external storage does not suffer, and if the workload justifies it, a ZS7-2 HE (High End) can be used for external storage, and will provide even higher performance than the internal ZS7-2 MR (Mid-Range). Available ZS7-2 Racked Systems provide an easy path for attaching external storage to the PCA X8-2.

OVM in the PCA X8-2 has several methods for provisioning storage to VM’s. Shared OVM repositories are available by default. These repositories provide storage for OS images as well as being used for virtual disks to be assigned to VM’s. vDisks are very flexible and easy to manage and are suitable for the operating system and the Cloudera binaries in a Cloudera installation. Physical iSCSI OVM pDisks can be created and managed by OVM, these have higher performance at the cost of somewhat less flexibility. Native iSCSI disks or NFS shares can be created outside of OVM control using PCA utilities, and these provide the highest performance. PCA X8-2 utilities can create custom storage networks to be used exclusively for accessing native iSCSI LUNs or shares.

For Cloudera, OVM vDisks are recommended for the basic operating system directories, as well as the directories where Cloudera binaries are installed. For busy databases or Cloudera data files, native iSCSI disks accessed over PCA X8-2 custom Storage Networks are the best choice. NFS shares are not supported by Cloudera.

A traditional Cloudera installation on bare metal servers with local disks may have four to twelve local drives on each server. Once a server is deployed, it is very difficult or impossible to change the number of disks or reassign

disks between server nodes. PCA X8-2 can match or exceed the performance of local disk and the number of drives assigned to a VM is completely flexible and configurable.

Building the Cloudera Infrastructure

We will illustrate an example of the infrastructure for a Cloudera cluster with 10 worker nodes, plus one Cloudera Manager node. All nodes will be running Oracle Linux 7.9. The necessary Cloudera database will reside on one of the Cloudera nodes.

The “Oracle Private Cloud Appliance Administrator's Guide for Release 2.4.3” (2.4.3 or higher is required) should be referred to for command structure and other details. The Oracle Solution Brief “Networking the Oracle Private Cloud Appliance and Oracle Private Cloud at Customer” provides in depth details about PCA X8-2 networking.

It is assumed that the user has familiarity with both the OVM Manager and the PCA X8-2 management node logins, and has the proper admin and root credentials to create networks, storage, and VMs. It is best to start by defining network and storage resources, then move on to creating the compute resources. It is assumed that the user has already loaded an .iso or an Oracle VM Template for the Oracle Linux 7 version desired. Refer to the “Oracle VM Manager User's Guide for Release 3.4” for details on loading an .iso or VM Template into an OVM Repository. The repository used in these examples is “Rack1_Repository”.

Defining Networks

Each Cloudera node will have six network interfaces.

1. ETH0: One network interface to connect to the data center network for user access and perhaps, data ingest and export. There is always a PCA default 10GbE network defined called default_external providing a portal into the customer data center. A custom network utilizing separate switch ports could also be used for greater bandwidth.
2. ETH1: A network interface to a non-routable internal network to be used for the inter-node communication. The pre-defined standard network called default_internal can be used, or a custom network can be defined.
3. ETH2-5: Four storage network interfaces connecting to four custom storage networks, providing four data paths to the ZFS Storage Appliance from each VM. PCA release 2.4.3 and higher can use the pca-admin command from a PCA X8-2 management node to define these custom storage networks. pca-admin should be for the storage network definitions, because in addition to creating the network on the PCA OVM environment, pca-admin will create the necessary network definitions on the internal ZFS Storage Appliance.

Jumbo frames should be used on all networks. Use the pca-admin command (documented in “Oracle Private Cloud Appliance Administrator's Guide for Release 2.4.3”) to define **all** networks. Do **not use** the OVM Manager to create networks. Once a network has been created by pca-admin, it will appear in the OVM Manager in the Networking tab and can be assigned to a VM. Assign any custom network to be used by Cloudera to all PCA X8-2 Compute Nodes where Cloudera VMs can run.

ETH0 Network

In this example a custom external network called vm_external_vlan_40g_v565 was already defined on our PCA. The network called default_external could have been used if 10GbE connections to the data center were sufficient. Use default_external where possible. If a custom external network is necessary, refer to “Oracle Private Cloud Appliance Administrator's Guide for Release 2.4.3” and the Oracle Solution Brief “Networking the Oracle Private Cloud Appliance and Oracle Private Cloud at Customer” for information about custom networks. If a custom network is created for ETH0, it should be an external network related to an uplink port group.

ETH1 Network

In this example, the default network for communication between VMs, called default_internal, is used. Since this network is always available on all PCA X8-2 Compute Nodes by default, it does not need to be created and added to PCA X8-2 Compute Nodes.

ETH2-ETH5 Networks

Four custom storage networks will be created that will be data paths to the ZFS Storage Appliance. These networks are created using the `pca-admin` command from the active PCA X8-2 management node.

To create the custom Storage Networks:

1. Login to the root user on the current active PCA X8-2 Management Node.
2. Choose four non-routable subnets to assign, one to each network. In this example we chose 10.10.67.0/24, 10.10.68.0/24, 10.10.69.0/24, and 10.10.70.0/24. The gateway will be defined as the .1 IP address in the subnet. The network subnets must be unique within the internal PCA infrastructure. Choose the IP address within the subnet which you want to use to communicate with the ZFS Storage Appliance. In this case, we decided to use .100, so to reach the ZFS Storage Appliance on subnet 10.10.67.0/24, we will use 10.10.67.100 as the ZFSSA IP address, on subnet 10.10.68.0/24, we will use 10.10.68.100 as the ZFSSA IP address, etc.
3. Use the `pca-admin` command to create the four storage networks, one at a time, as follows:

```
root@ovcamn05r1-pca# pca-admin create network cloudera1_net storage_network 10.10.67 255.255.255.0 10.10.67.100
                                <pca-admin command> <network name> <network type> <subnet> <subnet mask> <ZFS Storage
IP>
root@ovcamn05r1-pca# pca-admin create network cloudera2_net storage_network 10.10.68 255.255.255.0 10.10.68.100
root@ovcamn05r1-pca# pca-admin create network cloudera3_net storage_network 10.10.69 255.255.255.0 10.10.69.100
root@ovcamn05r1-pca# pca-admin create network cloudera4_net storage_network 10.10.70 255.255.255.0 10.10.70.100
```

Figure 1: 'pca-admin' commands to create storage networks

4. After creating the storage networks, add **each** storage network to **each** physical PCA X8-2 Compute Node where Cloudera nodes will reside.

```
root@ovcamn05r1-pca# pca-admin add network cloudera1_net ovcacn07r1
root@ovcamn05r1-pca# pca-admin add network cloudera1_net ovcacn08r1
root@ovcamn05r1-pca# pca-admin add network cloudera1_net ovcacn09r1
.
.
root@ovcamn05r1-pca# pca-admin add network cloudera2_net ovcacn07r1
root@ovcamn05r1-pca# pca-admin add network cloudera2_net ovcacn08r1
root@ovcamn05r1-pca# pca-admin add network cloudera2_net ovcacn09r1
.
.
.
root@ovcamn05r1-pca# pca-admin add network cloudera3_net ovcacn07r1
root@ovcamn05r1-pca# pca-admin add network cloudera3_net ovcacn08r1
root@ovcamn05r1-pca# pca-admin add network cloudera3_net ovcacn09r1
.
.
.
root@ovcamn05r1-pca# pca-admin add network cloudera4_net ovcacn07r1
root@ovcamn05r1-pca# pca-admin add network cloudera4_net ovcacn08r1
root@ovcamn05r1-pca# pca-admin add network cloudera4_net ovcacn09r1
```

Figure 2: 'pca-admin' commands to add storage networks to PCA X8-2 compute nodes

5. Verify each network definition with the show network command. Be sure the network prefix, the netmask, the PCA X8-2 Compute Nodes, and the ZFSSA storage IP for the storage networks are as expected, as well as verifying the ETH0 and ETH1 networks.

Here is an example of showing the first storage network, cloudera1_net, the other three networks should be similarly verified:

```
root@ovcamn05r1-pca# pca-admin show network cloudera1_net
-----
Network_Name      cloudera1_net
Trunkmode         None
Description       None
Ports            None
vNICs            None
Status           ready
Network_Type      storage_network
Compute_Nodes     ovcacn12r1, ovcacn11r1, ovcacn09r1, ovcacn10r1, ovcacn08r1, ovcacn07r1
Prefix           10.10.67
Netmask          255.255.255.0
Route_Destination None
Route_Gateway     None
Storage_IP       10.10.67.100
-----

Status: Success
```

Figure 3: Displaying a storage network after creation

- Verify that each Storage Network appears in the Networking tab in OVM Manager, with a “Y” in the Virtual Machine column. All six networks should have a “Y” **ONLY** in the Virtual Machine column, **do not alter anything** in the Networking tab in OVM.

The screenshot shows the Oracle VM Manager interface. The 'Networking' tab is active, displaying a table of networks. The table has columns for Name, ID, Intra-Network Server, and Network Channels (Server Management, Cluster Heartbeat, Live Migrate, Storage, Virtual Machine, and Description). The 'Virtual Machine' column contains checkmarks for several networks, including 'mg_k8s_net' and 'oracle-database-storage'.

Name	ID	Intra-Network Server	Network Channels					Description
			Server Management	Cluster Heartbeat	Live Migrate	Storage	Virtual Machine	
cloudera1_net	102e073365						√	
cloudera2_net	1009d0aaa5						√	
cloudera3_net	105eb5eb36						√	
cloudera4_net	10a252dc89						√	
default_external	10a7903108						√	
default_internal	105f170519						√	
demo_vm_internal	1089e759b2						√	
lab1_net	10a4c16d84						√	
mg_k8s_net	104d6150b9						√	
oracle-database-storage	10d605871b						√	
test_vm_internal1	10713bb504						√	
test_vm_internal2	1036623cde						√	
test_vm_internal3	102e9bdb0d						√	
test_vm_internal4	10f1f3e285						√	
vm_external_vlan_40g_v565	1081ced857						√	

Rows Selected: 1

Job Summary: 0 Total Jobs, 0 Pending, 0 In Progress, 0 Failed, 0 Aborted, 0 Complete

Description: No data to display

Figure 4: OVM Manager Networks Pane showing storage networks assigned to Virtual Machines Role **ONLY**

- In the internal ZFS Storage Appliance, creating Storage Networks will cause network interfaces and datalinks to be built on top of the four 40GbE physical network connections that connect into the PCA X8-2 network infrastructure. **DO NOT** modify the ZFSSA network configuration manually. Externally attached ZFS Storage Appliances can be manually configured in a similar manner to the way internal Storage Networks are automatically created on the internal ZFS Storage Appliance.

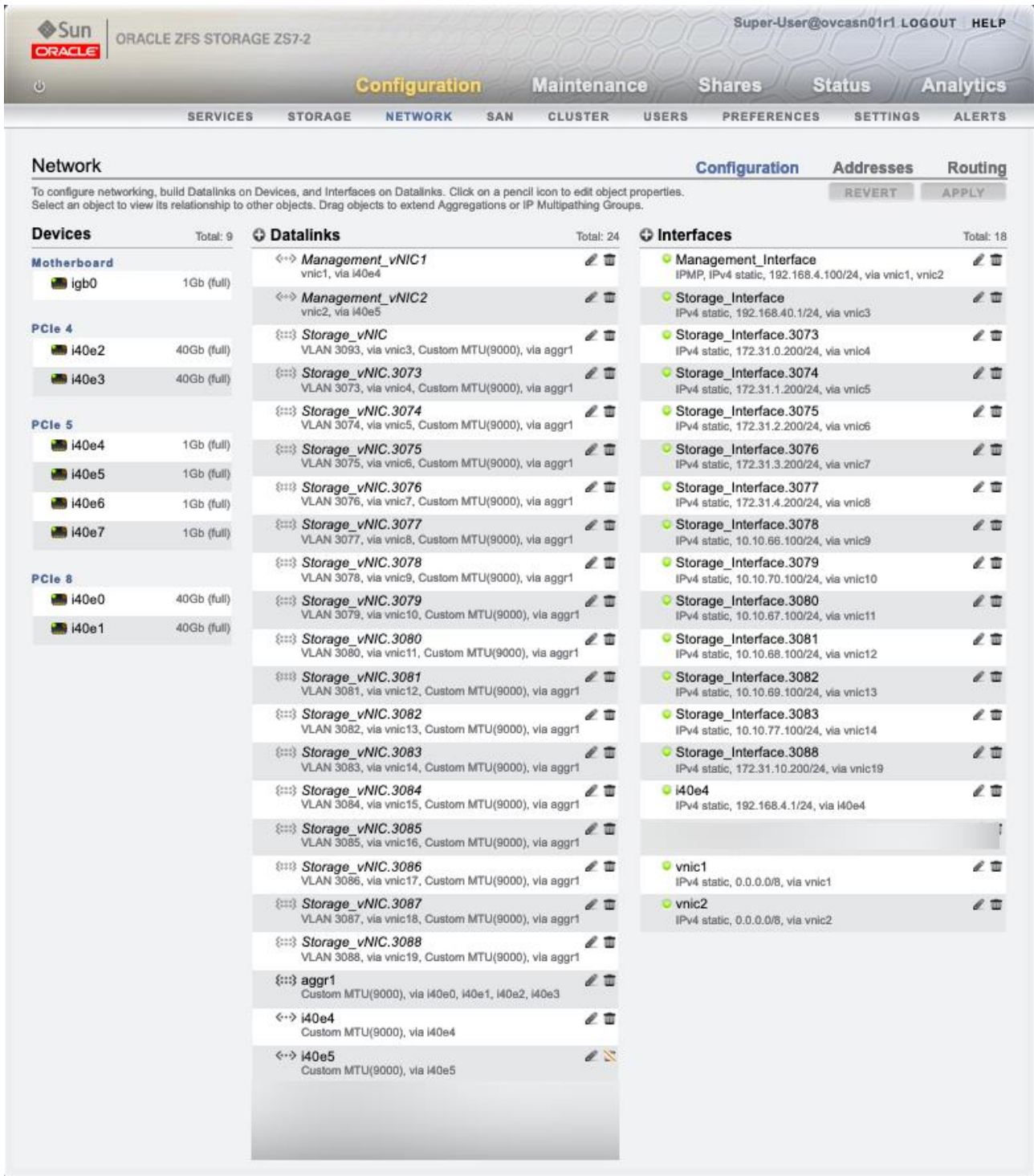


Figure 5: Example Network configuration on an internal ZFS ZS7-2 MR within an PCA X8-2

There are now six networks defined on the PCA X8-2 for use by the Cloudera VMs as ETH0 through ETH5.

Defining VMs

After the networks are defined, VMs that will run the Oracle Linux 7 OS and Cloudera can be defined. The VMs will use a single 160GB sparse vDisk for all storage needed by the OS and the Cloudera binaries. vDisks are created from shared OVM repositories containing many vDisks belonging to many VMs. A sparse vDisk only uses space in the repository when data is written to the disk. A new, separate repository could be created on either an iSCSI LUN or an NFS share to hold only the vDisks related to Cloudera, but in this example, we will use the default repository for the PCA X8-2, which is called Rack1_Repository. All six of the networks we discussed previously will be assigned to each VM. This example will use an Oracle Linux .iso file already loaded into a repository as the base for the OS build. Refer to the “Oracle VM Manager User’s Guide for Release 3.4” for details on loading an .iso or VM Template into an OVM Repository.

The OVM Manager Browser User interface will be used to create the VMs. We will illustrate the creation of one VM, the other ten will be identical.

1. Login to the OVM BUI using the admin user.
2. Click the “Servers and VMs” tab and then expand “Server Pools”. Click Rack1_ServerPool. In the action bar just above “Rack1_ServerPool”, click the “Create Virtual Machine..” icon. Hovering over the icons will give you hints.
3. On the “Create Virtual Machine” screen, click “Create a new VM” and then “Next”.
4. On the “Create Virtual Machine” details pane, change the following from defaults:
 - Repository: Set to “Rack1_Repository”
 - Name: Give the VM a descriptive name
 - Operating System: Set to your OS (Oracle Linux 7)
 - Domain Type: Set to Xen HVM PV Drivers
 - Set Max. Memory, Memory, Max. Processors and Processors as discussed previously and click “Next”.

* Server Pool:	Rack1_ServerPool	Operating System:	Oracle Linux 7
Server:	Any	<input type="checkbox"/> Enable Viridian	
* Repository:	Rack1-Repository	Mouse Device Type:	OS Default
* Name:	cloudera1	Keymap:	en-us (English, United States)
<input type="checkbox"/> Enable High Availability		Domain Type:	Xen HVM PV Drivers
<input type="checkbox"/> Enable Huge Pages		Start Policy:	Current Server
Description:		Max. Memory (MB):	220000
		Memory (MB):	220000
		Max. Processors:	24
		Processors:	24
		Priority:	50
		Processor Cap %:	100
		Restart Action On Crash:	Restart

Figure 6: Oracle VM Manager Create VM Wizard – page 1

- On the “Set up Networks” pane, create VNICs to be associated with the networks created in the section “Defining Networks”. The order in which the VNICs are listed will determine which ETHx interface is used to address them in the Oracle Linux OS, the first VNIC will be ETH0, the second ETH1, and so on. After the networks have been assigned in the proper order, click “Next”.

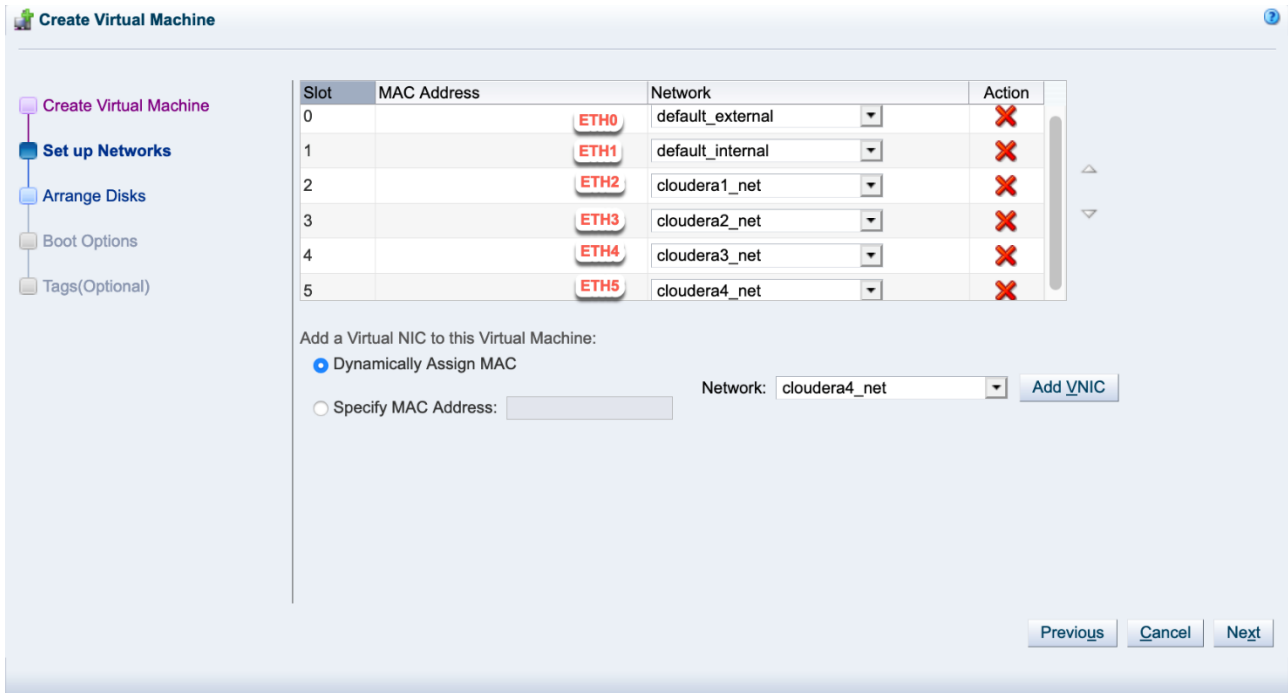


Figure 7: Oracle VM Manager Create VM Wizard – page 2

- In the “Arrange Disks” pane, select a Disk Type of “Virtual Disk”. Click the “+” that appears after you specify “Virtual Disk” in the dropdown. Select “Rack1_Repository”, give the disk a descriptive name, and select the size. 160GB should be big enough for both the OS and the Cloudera binaries. Leave the allocation type as “Sparse Allocation”. Click “OK”, then Click “Next”.

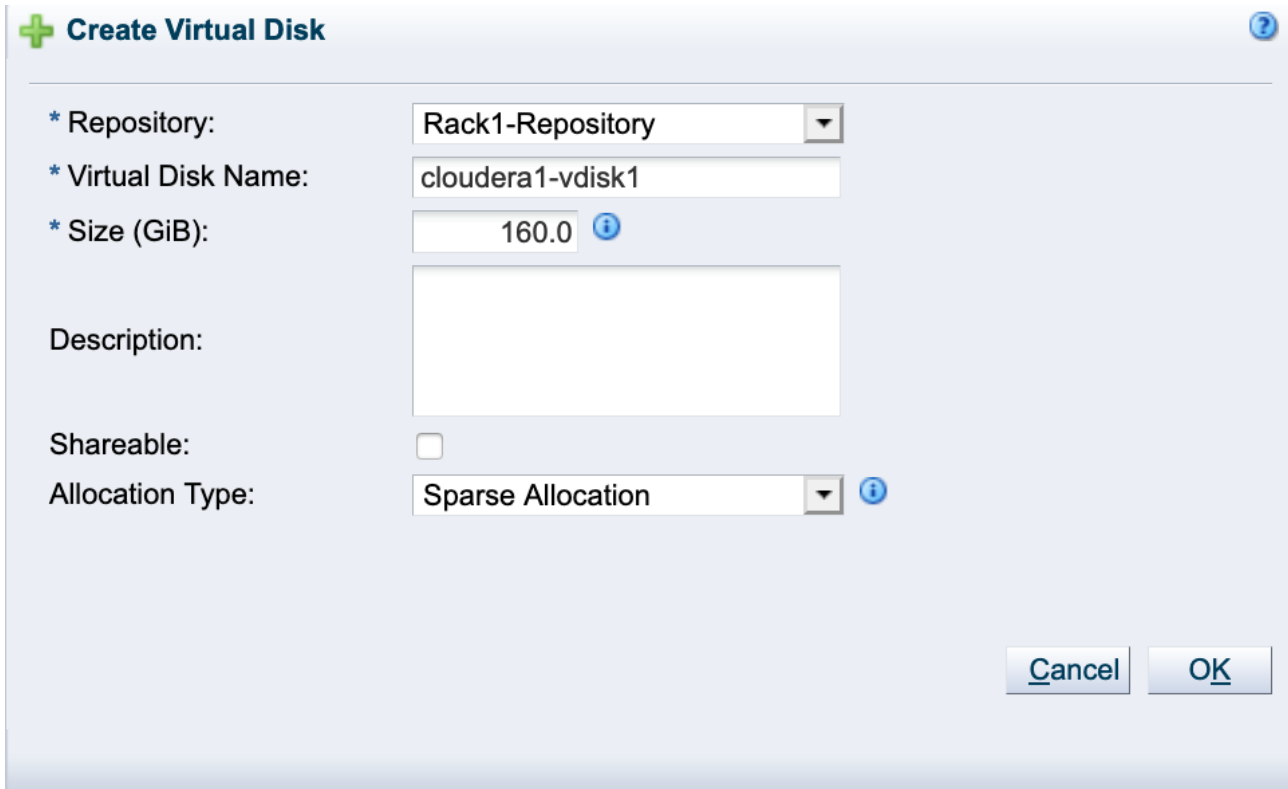


Figure 8: Oracle VM Manager Create VM Wizard – page 3 – create vDisk

- If you are going to build from an .iso, for slot 1, select CDROM, and then use the “Actions” icons to choose an .iso you have loaded into a repository. Click “Next”.

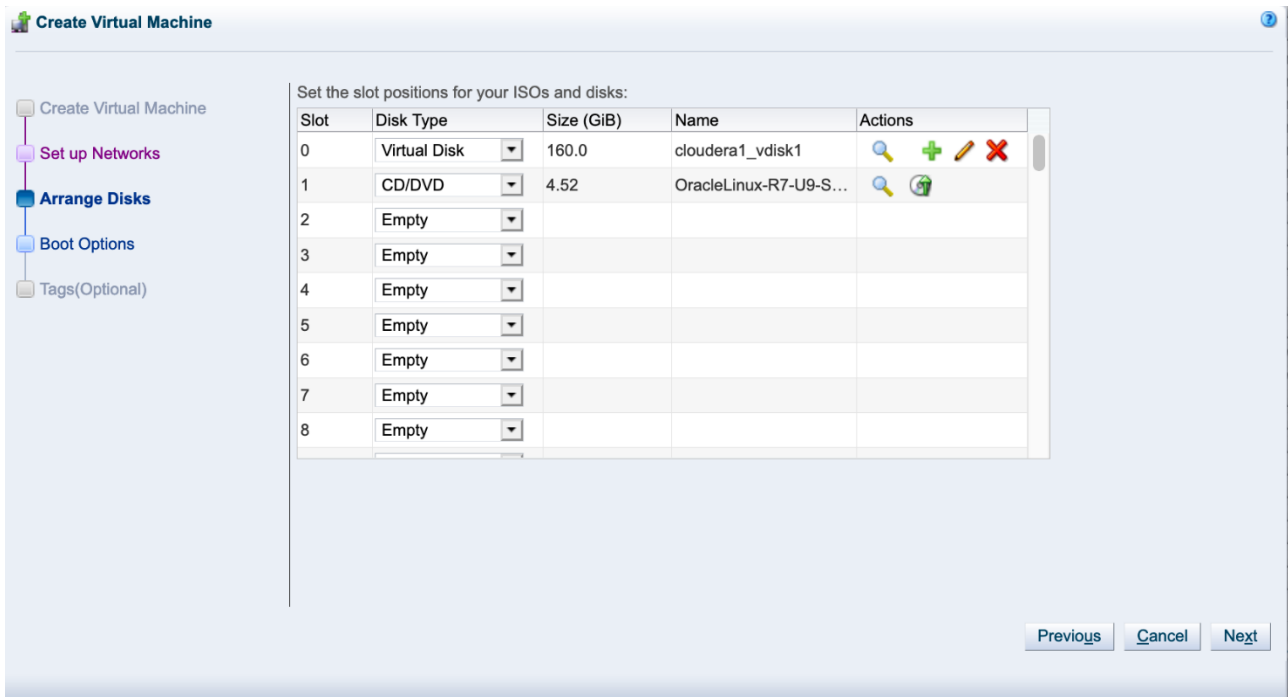


Figure 9: Oracle VM Manager Create VM Wizard – page 3 – select CDROM / ISO

- In the “Boot Options” pane, select CDROM as the first boot option to boot from CDROM as the initial boot media. Choose “Disk” as the second boot option. Click “Finish”. The VM will be created and will be in “Stopped” state. After the system is built, return to this screen and remove “CDROM” as the first option.

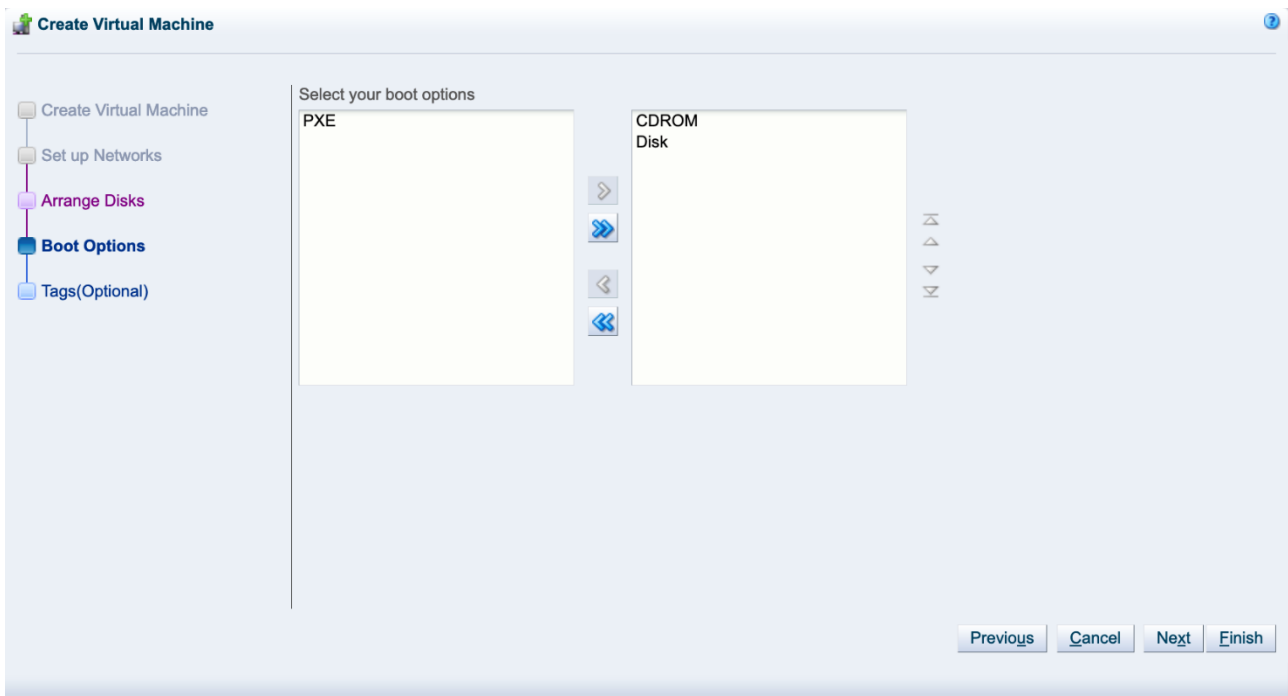


Figure 10: Oracle VM Manager Create VM Wizard – page 4

- Repeat the Build VM process for each VM needed.
- The next step is to boot from the selected .iso and customize Linux.

Customize Oracle Linux 7

It is assumed that the user is familiar with creating and configuring Oracle Linux 7; this will not be covered here. We will discuss changes that are necessary for the Oracle Linux 7 to run well on the PCA X8-2 as a base for a Cloudera Cluster.

Boot the Oracle Linux 7 .iso CDRROM image chosen and build and configure the Oracle Linux systems to be used as the nodes in your Cloudera cluster to your liking. The Oracle Linux 7 server types of Infrastructure Server, or Server with GUI, will install most required packages. Refer to the Cloudera installation documentation for a full list of package requirements. Once the systems are built, return to Oracle VM and change the boot order so that the Oracle Linux .iso is no longer the first boot option. Refer to the “Oracle VM Manager User’s Guide for Release 3.4” for details. During system configuration, you can configure ETH0.

It is assumed that:

1. You have root access.
2. You have configured ETH0 to be able to access the OS from your data center network, and any desired DNS and NTP settings were specified at install time.
3. You have configured yum repositories for Oracle Linux.
4. It will be necessary to inspect the distribution of the VMs across the PCA X8-2 Compute Nodes. When the VMs are created, OVM will choose a PCA X8-2 Compute Node, but the OVM Manager “Migrate or Move” function should be used to balance the distribution as needed. If you have worker nodes with more roles than others in the cluster, try to move those to PCA X8-2 Compute Nodes running worker nodes with fewer roles.
5. The internode communication network on ETH1 must be configured on each Cloudera node. Edit `/etc/sysconfig/network-scripts/ifcfg-eth1`. MTU=9000 is advised. Any non-routable subnet can be chosen for this network, which is connected to the PCA X8-2 default internal communication network called `default_internal`. However, you need to be sure you aren’t using IP addresses already used by other VMs also communicating on `default_internal`. In this example we have chosen 192.168.99.0/24 as the internal communication subnet. Each node in the cluster must have ETH1 connected to `default_internal`, and must have an IP address in the 192.168.99.0/24 subnet. If other workloads will be running on the PCA X8-2, it may be preferable to create a custom internal network for the Cloudera traffic. This example uses `default_internal`.

Here is an example `ifcfg-eth1` for one of the Cloudera nodes. You will need to change `ifcfg-eth1` on each node so that each node has a unique IP address on the `default_internal` network.

```
TYPE=Ethernet
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
NAME=eth1
DEVICE=eth1
ONBOOT=yes
IPADDR=192.168.99.nn ← Specify unique IP
PREFIX=24
MTU=9000
ETHTOOL_OPTS="autoneg on"
```

- Verify that each node can ping the other on the 192.168.99.0/24 network (or the network you have chosen).
- In order for Cloudera to use the ETH1 default_internal network, you will need to configure /etc/hosts to have a hostname related to each of the IP addresses of the Cloudera nodes.

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.99.24 cloudera-3.us.oracle.com cloudera-3
192.168.99.22 cloudera-1.us.oracle.com cloudera-1
192.168.99.23 cloudera-2.us.oracle.com cloudera-2
192.168.99.25 cloudera-4.us.oracle.com cloudera-4
192.168.99.26 cloudera-5.us.oracle.com cloudera-5
192.168.99.27 cloudera-6.us.oracle.com cloudera-6
192.168.99.28 cloudera-7.us.oracle.com cloudera-7
192.168.99.29 cloudera-8.us.oracle.com cloudera-8
192.168.99.30 cloudera-9.us.oracle.com cloudera-9
192.168.99.31 cloudera-10.us.oracle.com cloudera-10
192.168.99.32 cloudera-11.us.oracle.com cloudera-11
```

- After configuring the Oracle Linux 7 system, proceed to create and configure the iSCSI storage environment prior to installing the Cloudera software.

There are a group of Oracle Linux tunings that can help with intense workloads.

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag
echo "50" > /proc/sys/net/core/busy_poll
sysctl -w net.core.rmem_max=268435456
sysctl -w net.core.wmem_max=268435456
sysctl -w net.core.optmem_max=134217728
sysctl -w net.ipv4.tcp_rmem="4096 87380 134217728"
sysctl -w net.ipv4.tcp_wmem="4096 65536 134217728"
sysctl -w net.core.netdev_max_backlog=300000
sysctl -w net.ipv4.tcp_no_metrics_save=1
sysctl -w net.ipv4.tcp_timestamps=0
sysctl -w net.ipv4.tcp_sack=1
sysctl -w net.ipv4.tcp_low_latency=1
sysctl -w net.ipv4.tcp_congestion_control=htcp
sysctl -w net.ipv4.tcp_mtu_probing=1
```

Initial Customization of Oracle Linux 7 iSCSI Storage Services for Cloudera Data

The Cloudera data storage used in this example will reside on iSCSI LUNs accessed using the four Storage Networks created earlier, using the dm-multipath facility of Linux. dm_multipath can be installed from the standard Oracle Linux 7 yum repositories by issuing this command:

```
yum install device-mapper-multipath
```

After installing dm-multipath, a parameter file must be modified in /etc/multipath.conf to select the parameters applicable to the LUNs we will create on the Oracle ZFS Storage Appliance. Assuming there are no other multipath devices being used by your Cloudera VMs, here is the stanza necessary for multipath to operate on iSCSI LUNs on the ZFS Storage Appliance:

```
devices {  
    device {  
        vendor                "SUN"  
        product               "ZFS Storage.*"  
        prio                  alua  
        hardware_handler      "1 alua"  
        path_grouping_policy  group_by_prio  
        path_selector         "round-robin 0"  
        failback              immediate  
        no_path_retry         600  
        rr_min_io_rq          100  
        path_checker          tur  
        rr_weight             uniform  
        features               "0"  
    }  
}
```

dm_multipath runs as a service called multipathd . After configuring the /etc/multipath.conf file, issue the following commands to start multipathd and enable it to be started at system boot time:

```
systemctl start multipathd
```

```
systemctl enable multipathd
```

Our example will create 12 iSCSI LUNs for each of the ten worker nodes and the one Cloudera Manager node. Using many LUNs increases the throughput of the Cloudera workloads. The Cloudera Manager node does not necessarily need 12 iSCSI LUNs as long as it remains only running the Cloudera Manager, but creating the LUNs allows that node to be converted to a worker node at a later point if desired. This example will show use of the ZFS Storage Appliance Browser User Interface (BUI). Appendix A gives an overview of how to access the BUI by tunnelling through a PCA X8-2 Management Node. The ZFS Storage Appliance CLI or REST API can also be used.

There are multiple steps involved in configuring Oracle Linux 7, and the storage. It is helpful to open a window with the ZFS Storage Appliance BUI, and an Oracle Linux command line logged into root. The following steps are required on **each** Cloudera node.

1. Configure `dm_multipath` as shown above.
2. Generate an iSCSI iqn, which is the initiator iSCSI Qualified Name. This is a unique identifier for each system's iSCSI initiator. The iqn belongs to each Oracle Linux 7 host, and is not connected to an IP address. To generate a unique iqn, login to the host :
 - Issue the following commands:

```
cp /etc/iscsi/initiatorname.iscsi /tmp/initiatorname.iscsi.old
echo "InitiatorName=`/sbin/iscsi-iname`" > /etc/iscsi/initiatorname.iscsi
```
 - The above command will generate an iqn string and put it into `/etc/iscsi/initiatorname.iscsi` . Only run this command **once**. `/etc/iscsi/initiatorname.iscsi` should look something like this:

```
cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1988-12.com.oracle:ad22aa8bfd3a
```
3. Configure each of the four Storage Network interfaces (ETH2, ETH3, ETH4, ETH5) to communicate to the ZFS Storage Appliance on the Storage Networks created in the Defining Networks section previously. It is important to configure the interfaces with the proper subnets. When you defined each of the Storage Networks using the `pca-admin` command on the PCA X8-2 management node, you specified a subnet and an address for the ZFS Storage Appliance. You then assigned vNICs to each of the four Storage Networks assigned to each of the Cloudera VMs, in a specific order. Create network interface definitions using those subnets. In our example, the subnets created in this example are 10.10.67.0/24, 10.10.68.0/24, 10.10.69.0/24, and 10.10.70.0/24. One ETHx definition must address each subnet. Each node must have a unique IP in each subnet, and must not use the IP assigned to the ZFS Storage Appliance specified when you defined the Storage Networks. In this example, the ZFS Storage appliance is .100 on each of the above subnets. Each of our Cloudera nodes needs to be assigned a different IP suffix. In this example the first Cloudera node uses 10.10.67.22 for ETH2, 10.10.68.22 for ETH3, 10.10.69.22 for ETH4 and 10.10.70.22 for ETH5. The second Cloudera node uses 10.10.67.23 for ETH2, 10.10.68.23 for ETH3, 10.10.69.23 for ETH4 and 10.10.70.23 for ETH5, and so on. Continue until all Storage Network IPs are configured on each node. A worksheet may be helpful.

Here are example `/etc/sysconfig/network-scripts/ifcfg-eth(x)` files for ETH2 through ETH5 on the first node:

```
ifcfg-eth2
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
NAME=eth2
DEVICE=eth2
ONBOOT=yes
MTU=9000
IPADDR=10.10.67.22
PREFIX=24
GATEWAY=10.10.67.1
```

```
ifcfg-eth3
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
NAME=eth3
DEVICE=eth3
ONBOOT=yes
MTU=9000
IPADDR=10.10.68.22
PREFIX=24
GATEWAY=10.10.68.1
```

```
ifcfg-eth4
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
NAME=eth4
DEVICE=eth4
ONBOOT=yes
MTU=9000
IPADDR=10.10.69.22
PREFIX=24
GATEWAY=10.10.69.1
```

```
ifcfg-eth5
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
NAME=eth5
DEVICE=eth5
ONBOOT=yes
MTU=9000
IPADDR=10.10.70.22
PREFIX=24
GATEWAY=10.10.70.1
```

After configuring the four Storage network interfaces, be sure the ZFS Storage Appliance is reachable on each subnet:

```
ping -c3 10.10.67.100
```

```
ping -c3 10.10.68.100
```

```
ping -c3 10.10.69.100
```

```
ping -c3 10.10.70.100
```

Create iSCSI Definitions and iSCSI LUNs on the Internal ZFS Storage Appliance

CREATE TARGETS, TARGET GROUPS, PROJECTS, AND LUNS, THEN RUN `iscsadm` COMMANDS TO MAKE ALL FOUR INTERFACES ELIGIBLE AND NAME THEM. `/dev/mpath` devices should ensue. FORMAT AND CREATE EXT4 FILESYSTEMS. CREATE MOUNTPOINTS AND `fstab` entries, THEN MOUNT. DONE!

The internal ZFS Storage Appliance must be configured to present LUNs over each Storage network to each Cloudera node. In this example, we will present twelve LUNs to each node. Spreading the I/O workload among many nodes helps to alleviate operating system queuing at the device level. Each LUN will be formatted with a single partition, and an ext4 filesystem will be created on each LUN. During Cloudera configuration, Cloudera will build an HDFS file system using each LUN on each data worker node.

We will illustrate using the ZFS Storage Appliance BUI to create initiators, initiator groups, and LUNs. See Appendix A for instructions about accessing the BUI.

1. Define in the ZFSSA BUI the initiator for each Cloudera node. For EACH Oracle Linux 7 Node:
 - Login to the ZFSSA as root.
 - Login to the Cloudera node.
 - In Step 2 of the section “Initial Customization of Oracle Linux 7 for iSCSI Storage for Cloudera Data”, an iSCSI IQN was generated and stored in `/etc/iscsi/initiatorname.iscsi`. For each node, the IQN string (which begins with “iqn.”) will be needed. Copy the IQN (which begins with “iqn.”).
 - In the ZFSSA BUI, click SAN->iSCSI->Initiators.
 - In the window labelled “Identify iSCSI Initiator: click “+Initiators”. Paste or type the IQN for the node into the field labelled “Initiator IQN”. Type a descriptive name relating to the node into the field labelled “Alias”. Click “OK”.

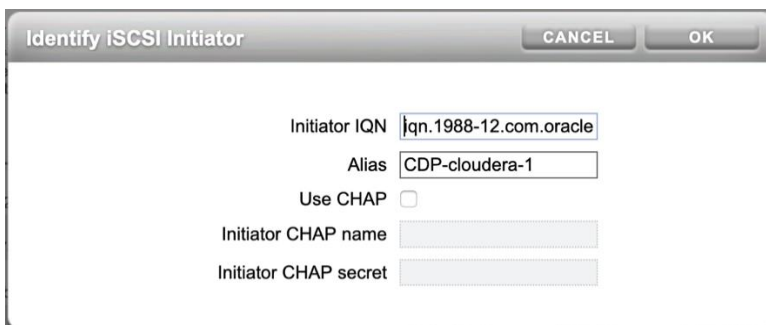


Figure 11: ZFS Storage Appliance ZS7-2 MR iSCSI Initiator setup

2. Repeat the above for each Cloudera node, obtaining the unique IQN for each node and creating a separate initiator definition on the ZFSSA for each node.
3. Create an Initiator Group for each Cloudera node containing the unique initiator IQN for that node, and a special entry for “fakeinitiator”, which prevents duplicate LUN numbers.

Find the entry called “FAKE_INITIATOR” in the initiators column. Put your mouse arrow just to the left of the entry and a cross of arrows should appear, indicating that the entry can be dragged and dropped. Drag this entry to the bottom of the list of initiator groups and drop it. A group called “initiators-0” will be created. Now, drag and drop

the initiator you created in Step 2 on top of the group “initiators-0”. Be precise. An initiator group now exists that is called “initiators-0” containing two initiators, the “fakeinitiator” and the initiator for the Cloudera node. Highlight “initiators-0” and click the pencil icon to the right to edit “initiator-0”. Give the initiator group a new, descriptive name relating to the Cloudera node, and click “OK”.

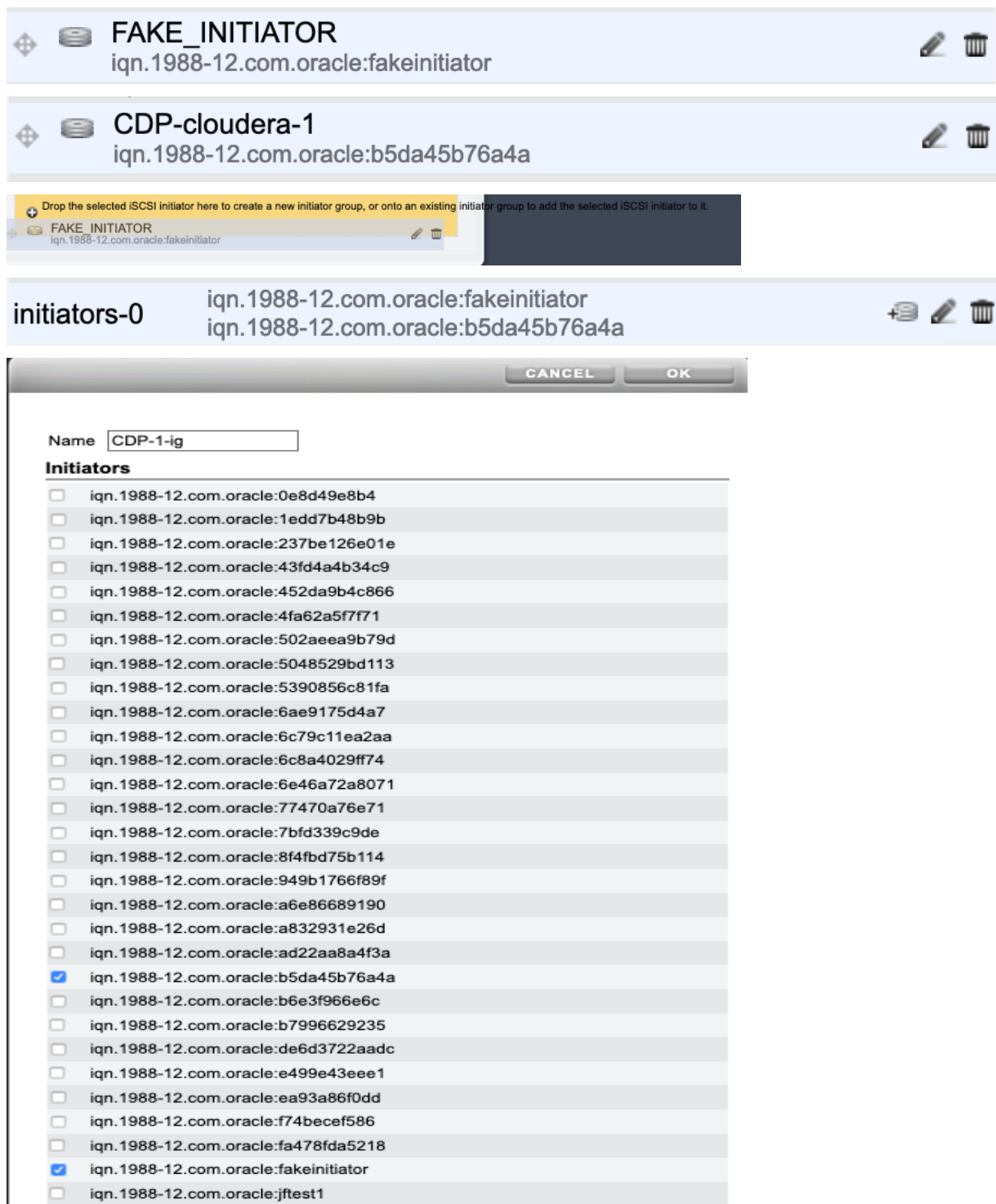


Figure 12: ZFS Storage Appliance ZS7-2 MR iSCSI Initiator Group setup

4. Repeat 3 above for each Cloudera node until you have an initiator and an initiator group for each Cloudera node.
5. Once initiators and initiator groups are created, LUNs need to be created belonging to each initiator group. The amount of space available to HDFS on each node will depend on the number of LUNs and the size of each LUN. The LUNs do not necessarily need to be the same size, but there is no reason to have some LUNs bigger than others, since all LUNs will be aggregated under an HDFS filesystem by Cloudera.

LUN's should belong to a ZFSSA Project. This will make it easier to define the LUNs by allowing the LUNs to inherit common attributes during definition. This ensures that all LUNs being used for Cloudera data are symmetric.

- Create Project(s) for Cloudera data LUNs. You can create a Project for each node, or one Project for all LUNs. The advantage of creating a Project for each node is that Snapshot and Replication actions can be done at a Project level. If all LUNs belong to the same Project, all LUNs must be snapped or replicated together if those functions are to be used in the future. If there is a Project for each node, LUNs can be managed at a node level.
- To create a Project, in the ZFSSA BUI, click Shares. In the upper left corner, Click Pools and select the Pool in which your Cloudera data will reside. As mentioned earlier, it is ****not**** recommended to use the pool named OVCA_POOL for Cloudera data for any workload other than casual test. If a production workload is expected, install additional storage enclosures and create a new, pool to contain the Cloudera data.
- After selecting the proper Pool, expand the Projects pane under the Pool selection, and click the “+”. Enter a name for the pool, and select whether encryption will be used. Encryption in ZFSSA is efficient but does extract a 10% or higher I/O penalty, higher levels of encryption “cost” more in performance. Refer to the Oracle Solution Brief “Best Practices for Deploying Encryption and Managing Its Keys on Oracle ZFS Storage Appliance” for more details about encryption on the ZFSSA.

Figure 13 : ZFS Storage Appliance ZS7-2 MR ZFS Project create

- After creating the Project, it will appear in the Projects column. Click on the Project name to display the Project attribute screen. Click on the General tab to adjust defaults for the Project.

There are specific LUN attributes that will improve performance for Cloudera workloads. HDFS workloads categorize as “Large block, streaming”, in general. These attributes should be specified in the Project so that the attributes will be automatically inherited by any LUNs created in the Project. Important attributes for Cloudera LUNs that are not defaults are:

- Data Compression – Data compression is highly recommended, and LZ4 compression is the recommended setting
- Synchronous write bias – Throughput
- Database record size/Volume block size – 128k
- Volume size – set to your preferred default volume size for each LUN, each LUN will be created at this size unless the size is overridden when the LUN is created

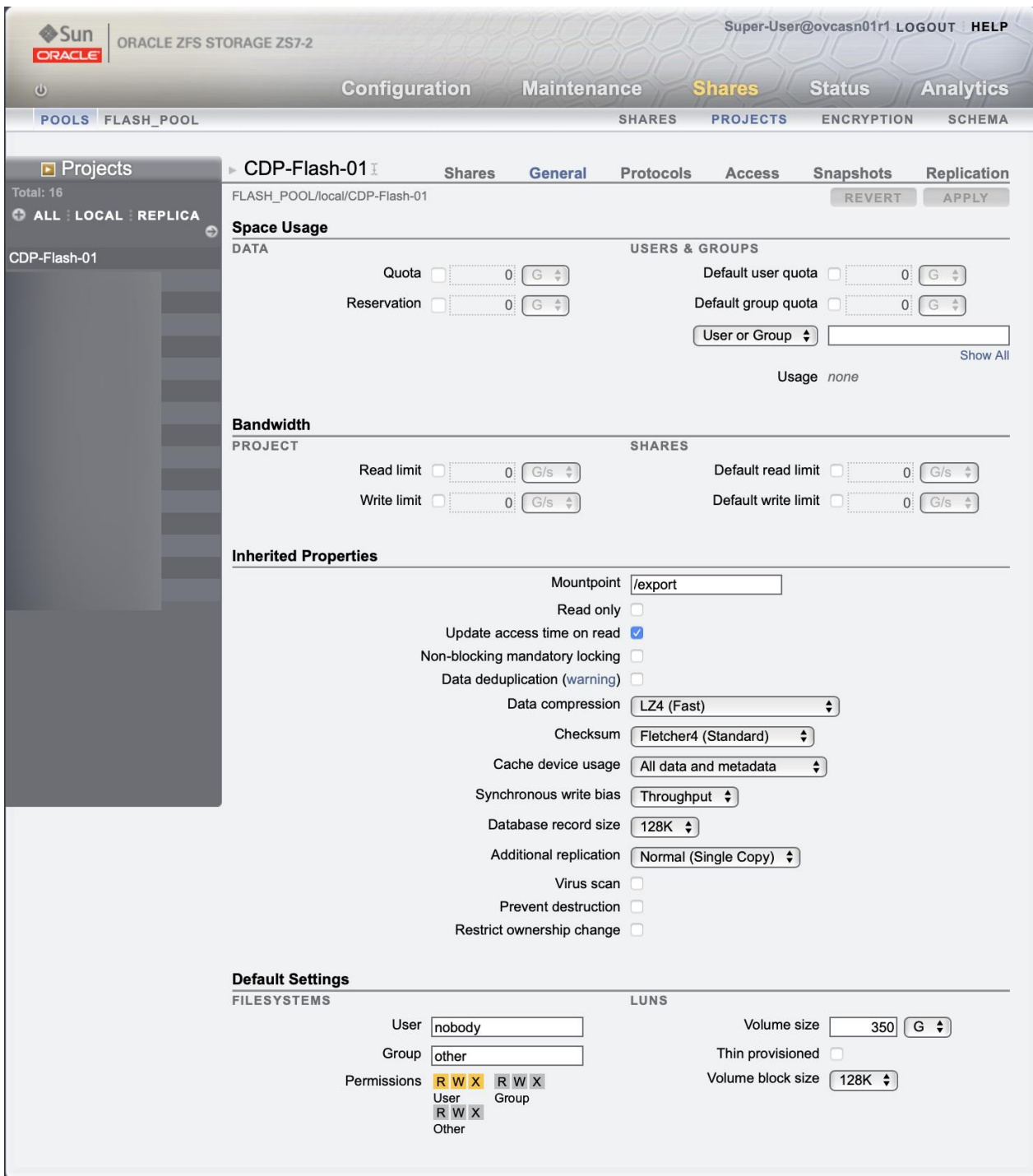


Figure 14:: ZFS Storage Appliance ZS7-2 MR ZFS Project setup

6. Once the Projects have been created, continue to create the desired number of LUNs for each node. In our configuration, we used 12 LUNs for each node. 1000 total LUNs per ZFS Storage Appliance controller is a soft limit.
7. To create LUNs, return to Configuration->SAN->iSCSI->Initiators. Click an initiator group that you created. To the right, an icon of a “+” sign and a disk drive will appear, click this icon. The “Create LUN” screen will appear. Assuming that the Project defaults were adjusted, the Volume size, and the Volume block size will be pre-seeded. Give the volume a descriptive name that relates it to the initiator group and node. Be sure “Online” is checked. The proper initiator group should be chosen for you since you are adding directly to the initiator group, but verify that the correct initiator group is chosen. The Target group should be “All targets” to allow all four Storage Networks to operate on the LUN. Auto-assign should be indicated so that a unique LUN number is created.

Create LUN
CANCEL
APPLY

Project

Name

Volume size

Thin provisioned

Volume block size

Online

Target group

Initiator group(s)

LU number 0 Auto-assign

Encryption

Inherit key

Key Local OKM

Figure 15 : ZFS Storage Appliance ZS7-2 MR iSCSI LUN create

8. Continue creating LUNs until all LUNs are created in each initiator group for all Cloudera nodes.

Discover iSCSI LUNs, Format Devices, Create Filesystems on each Cloudera Node

Four PCA X8-2 Storage Networks were created earlier. iSCSI initiators, initiator groups, and LUNs were created to provide storage for each Cloudera node. Each Cloudera node Oracle Linux 7 system must now attach to the storage, format volumes and create and mount filesystems. Finally, multipathing must be verified.

1. Login as root to the first Cloudera node.
2. The iscsiadm command is necessary to configure iSCSI on Oracle Linux 7. iscsiadm is part of the package iscsi-initiator-utils. This can be installed with the following command:

```
yum install iscsi-initiator-utils
```

3. Each Storage Network interface must be identified to iSCSI as being eligible for traffic:

```
iscsiadm -m iface -I eth2
iscsiadm -m iface -I eth3
iscsiadm -m iface -I eth4
iscsiadm -m iface -I eth5
```

4. Name each iSCSI interface:

```
iscsiadm -m iface -I eth2 -o update -n iface.net_ifacename -v eth2
iscsiadm -m iface -I eth3 -o update -n iface.net_ifacename -v eth3
iscsiadm -m iface -I eth4 -o update -n iface.net_ifacename -v eth4
iscsiadm -m iface -I eth5 -o update -n iface.net_ifacename -v eth5
```

```
iscsiadm -m iface
```

```
default tcp,<empty>,<empty>,<empty>,<empty>
iser iser,<empty>,<empty>,<empty>,<empty>
eth2 tcp,<empty>,<empty>,eth2,<empty>
eth3 tcp,<empty>,<empty>,eth3,<empty>
eth4 tcp,<empty>,<empty>,eth4,<empty>
eth5 tcp,<empty>,<empty>,eth5,<empty>
```

5. Discover the iSCSI initiators on each interface (Use the four IP addresses for the ZFS Storage Appliance specified when you created the Storage Networks using the `pca-admin` command previously.)

```
iscsiadm -m discovery -t st -p 10.10.67.100 -I eth2
iscsiadm -m discovery -t st -p 10.10.68.100 -I eth3
iscsiadm -m discovery -t st -p 10.10.69.100 -I eth4
iscsiadm -m discovery -t st -p 10.10.70.100 -I eth5
```

When the Storage Networks were created by the `pca-admin` command, an iSCSI target and iSCSI target group were created on the ZFS Storage Appliance for each Storage Network. The iSCSI target IQN is required for the Cloudera node to login to the target and access the LUNs.

To find the target IQN in the ZFSSA BUI, navigate to Configuration->SAN->iSCSI->Targets. Targets with the format “OVM-iscsi.nnnn” and target groups with the format of OVM.nnnn should be seen. T

he targets were created in ascending order when created. If you need to figure which target corresponds to a Storage Network subnet, the “nnnn” suffix appears as “Storage_Interface.nnnn” for each ZFSSA IP address in Configuration->Network.

In our example, a Network Interface called “Storage_Interface.3080” with an IP address of 10.10.67.100 was created. This corresponds to an iSCSI target called OVM-iscsi.3080. We want to login to the target IQN ending in f59d3d4ff23f using IP address 10.10.67.100.

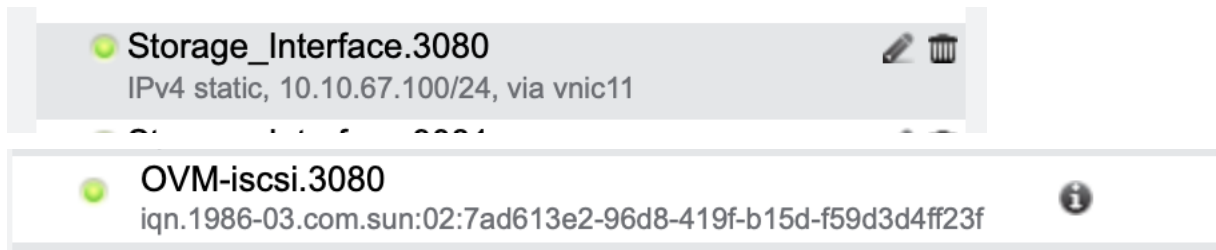


Figure 16: ZFS storage appliance ZS7-2 MR iSCSI target and target group

```
iscsiadm -m node --targetname iqn.1986-03.com.sun:02:7ad613e2-96d8-419f-b15d-
f59d3d4ff23f --portal 10.10.67.100:3260 -login
```

Each Storage Network interface must login to its target on each node. There should be four `iscsiadm` commands to login issued on each node, one for each Storage Network interface/target pair.

After the four logins, you should verify that there are four target logins on each node with the following `iscsiadm` command. These logins will be remembered across reboots.

```
iscsiadm -m session
```

```
tcp: [1] 10.10.70.100:3260,2 iqn.1986-03.com.sun:02:2647063d-bbb0-4ec0-b70c-  
ba15d6580444 (non-flash)
```

```
tcp: [2] 10.10.68.100:3260,4 iqn.1986-03.com.sun:02:2bb7232a-4b80-4f70-97dd-  
e923b827b2f5 (non-flash)
```

```
tcp: [3] 10.10.69.100:3260,5 iqn.1986-03.com.sun:02:3bb1c464-2176-498b-ad0f-  
d725206db1d8 (non-flash)
```

```
tcp: [4] 10.10.67.100:3260,3 iqn.1986-03.com.sun:02:7ad613e2-96d8-419f-b15d-  
f59d3d4ff23f (non-flash)
```

6. Discover and login should find the LUNs associated with the node's initiator group and should build a `/dev` entry for each path to each device. If you have 12 LUNs and four paths, there should be 48 `/dev/sd*` entries created in `/dev`. Now, assuming the proper configuration of `/etc/multipath.conf` as discussed earlier, restarting the multipath service should build a singular `/dev/mapper/mpath(x)` entry for each LUN.

```
systemctl restart multipathd
```

```
multipath -ll
```

```

mpathe (3600144f09030b5990000601b3f4f00a7) dm-23 SUN      ,ZFS Storage 7370
size=350G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
`-- policy='round-robin 0' prio=50 status=active
  |- 2:0:0:221 sdn  8:208  active ready running
  |- 3:0:0:221 sdt  65:48  active ready running
  |- 4:0:0:221 sdan 66:112 active ready running
  `-- 5:0:0:221 sdbd 67:112 active ready running
mpathd (3600144f09030b5990000601b3f4c00a5) dm-10 SUN      ,ZFS Storage 7370
size=350G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
`-- policy='round-robin 0' prio=50 status=active
  |- 2:0:0:217 sdi  8:128  active ready running
  |- 3:0:0:217 sdp  8:240  active ready running
  |- 4:0:0:217 sdal 66:80  active ready running
  `-- 5:0:0:217 sdbb 67:80  active ready running
mpathp (3600144f09030b5990000606f72ca0007) dm-7 SUN      ,ZFS Storage 7370
size=2.0T features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
`-- policy='round-robin 0' prio=50 status=active
  |- 2:0:0:407 sdab 65:176 active ready running
  |- 3:0:0:407 sdae 65:224 active ready running
  |- 4:0:0:407 sdaw 67:0   active ready running
  `-- 5:0:0:407 sdbk 67:224 active ready running
mpathc (3600144f09030b5990000601b3f5300a9) dm-25 SUN      ,ZFS Storage 7370
size=350G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
`-- policy='round-robin 0' prio=50 status=active
  |- 2:0:0:225 sds  65:32  active ready running
  |- 3:0:0:225 sdx  65:112 active ready running
  |- 4:0:0:225 sdap 66:144 active ready running
  `-- 5:0:0:225 sdbf 67:144 active ready running
mpatho (3600144f09030b5990000606f733f0008) dm-12 SUN      ,ZFS Storage 7370
size=2.0T features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
`-- policy='round-robin 0' prio=50 status=active
  |- 2:0:0:409 sdac 65:192 active ready running
  |- 3:0:0:409 sdah 66:16  active ready running
  |- 4:0:0:409 sday 67:32  active ready running
  `-- 5:0:0:409 sdbl 67:240 active ready running
mpathb (3600144f09030b5990000601b3f4a00a4) dm-16 SUN      ,ZFS Storage 7370
size=350G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
`-- policy='round-robin 0' prio=50 status=active
  |- 2:0:0:215 sdg  8:96   active ready running
  |- 3:0:0:215 sdm  8:192  active ready running
  |- 4:0:0:215 sdak 66:64  active ready running
  `-- 5:0:0:215 sdba 67:64  active ready running
.
.
.

```

Each LUN can now be partitioned and formatted for use by using the /dev/mapper/mpath(x) handle. Create a single partition encompassing the entire LUN:

```
fdisk -l /dev/mapper/mpatha

Disk /dev/mapper/mpatha: 375.8 GB, 375809638400 bytes, 734003200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 131072 bytes / 131072 bytes
Disk label type: dos
Disk identifier: 0x23561f7d
```

Device	Boot	Start	End	Blocks	Id	System
/dev/mapper/mpatha1		2048	734003199	367000576	83	Linux

Figure 17: List of Partition after Creation

7. After formatting, there will be a /dev/mapper/mpath(x)1 entry for each partition created, for example, /dev/mapper/mpatha1, /dev/mapper/mpathb1, /dev/mapper/mpathc1 and so on. These are the partitions where the ext4 (recommended) file system should be created. Create an ext4 filesystem on each LUN:

```
mkfs.ext4 /dev/mapper/mpatha1
mkfs.ext4 /dev/mapper/mpathb1
mkfs.ext4 /dev/mapper/mpathc1
.
.
```

Figure 18: Filesystem creation

8. For each LUN, we need to create a mountpoint directory and add an entry for each LUN to /etc/fstab. Then, for each LUN, we need to issue a mount command to mount the LUN on the mountpoint designated in /etc/fstab. The LUNs will be mounted automatically at boot time through the fstab entries. They can be mounted the first time via a mount command:

```
mkdir /cdp01
mkdir /cdp02
mkdir /cdp03
.
.
.
vi /etc/fstab

/dev/mapper/mpatha1 /cdp01 ext4 defaults 0 0
/dev/mapper/mpathb1 /cdp02 ext4 defaults 0 0
/dev/mapper/mpathc1 /cdp03 ext4 defaults 0 0
.
.
.
```

```
mount /cdp01
mount /cdp02
mount /cdp03
.
.
```

Figure 19: Cludera filesystem mounts

it is a good idea to reboot the node at this point to be sure that the LUNs will come online with multipath enabled after a reboot. Use the mount command and the multipath -ll command to verify after reboot.

Proceed with Cloudera Installation

The CPU/Memory/Network, and Storage infrastructure are now prepared for Cloudera installation. The installation of Cloudera is beyond the scope of this document but there are some key elements that will help the installation be successful:

1. When building the Cloudera Cluster, use the host names specified in /etc/hosts that relate to the default_internal network that were created in Step 5 of the section “Customize Oracle Linux 7”.
2. Spread all of the Cloudera data across all of the LUNs created previously. Great I/O bandwidth is a key component of a high performing Cloudera cluster. During the install process, the configuration options `dfs.datanode.data.dir` and `dfs.namenode.name.dir` are populated during the wizard installation. All mountpoints for the LUNs that were created should be specified. Cloudera will by default create subdirectories named `<mountpoint>dfs/dn` and `<mountpoint>/dfs/nn`.

The screenshot displays a configuration window with two main sections: 'DataNode Data Directories' and 'NameNode Data Directories'. Each section has a 'Default Group' dropdown menu and a list of input fields for specifying directory paths. Each input field is accompanied by a blue expand/collapse icon.

Section	Property	Default Group	Input Fields
DataNode Data Directories <code>dfs.datanode.data.dir</code>	DataNode Default Group	?	<code>/cdp01/dfs/dn</code>
			<code>/cdp02/dfs/dn</code>
			<code>/cdp03/dfs/dn</code>
			<code>/cdp04/dfs/dn</code>
			<code>/cdp05/dfs/dn</code>
			<code>/cdp06/dfs/dn</code>
			<code>/cdp07/dfs/dn</code>
			<code>/cdp08/dfs/dn</code>
			<code>/cdp09/dfs/dn</code>
			<code>/cdp10/dfs/dn</code>
			<code>/cdp11/dfs/dn</code>
			<code>/cdp12/dfs/dn</code>
NameNode Data Directories <code>dfs.namenode.name.dir</code>	NameNode Default Group	?	<code>/cdp01/dfs/nn</code>
			<code>/cdp02/dfs/nn</code>

Figure 20: Cloudera Configuration

Appendix A – Accessing the Administrative Interface for the PCA X8-2 Internal Oracle ZFS Storage Appliance

The administrative interfaces to the PCA X8-2 internal Oracle ZFS Storage Appliance are only accessible through the PCA X8-2 active Management Node. The static admin interface for ovcasn01r1 is always 192.168.4.1 and ovcasn02r1 is 192.168.4.2. The storage nodes may be reached from the Management Nodes using their respective names, ovcasn01r1 and ovcasn02r1. The IP address of the currently active Oracle ZFS Storage Appliance controller is always presented to the management nodes on IP address 192.168.4.100.

If the Appliance Command Line Interface (CLI) is to be used, use ssh to login to root on the currently active PCA X8-2 Management Node, then ssh from that command line to login to root on the ZFS Storage Appliance node on which you want to operate. You will be presented with ZFS Storage Appliance CLI prompts, and ZFS Storage Appliance CLI commands may be issued from there.

If the Oracle ZFS Storage Appliance Browser User Interface (BUI) is to be used, a ssh tunnel is necessary to relay port 215 traffic to the workstation that is running the browser.

1. Login to the PCA X8-2 active management node and establish a tunnel to 192.168.4.100:215 and pass traffic back to the local session on port 2215.

```
user $ ssh root@pca-vip -L 2215:192.168.4.100:215
```

```
root@pca-vip's password: *****
```

```
Last login: Mon Nov 9 15:33:40 2020 from dhcp-11-11-11-75.vpn.oracle.com
```

```
root@ovcamn06r1 #
```

Set manual Proxy settings in your browser to SOCKS Host 127.0.0.1:2215, then connect to <https://localhost:2215> to access the Oracle ZFS Storage Appliance BUI. In Firefox, these settings are in Preferences->Network Settings. Your browser will be unable to access any websites other than the ZFSSA BUI while these manual proxy settings are active.

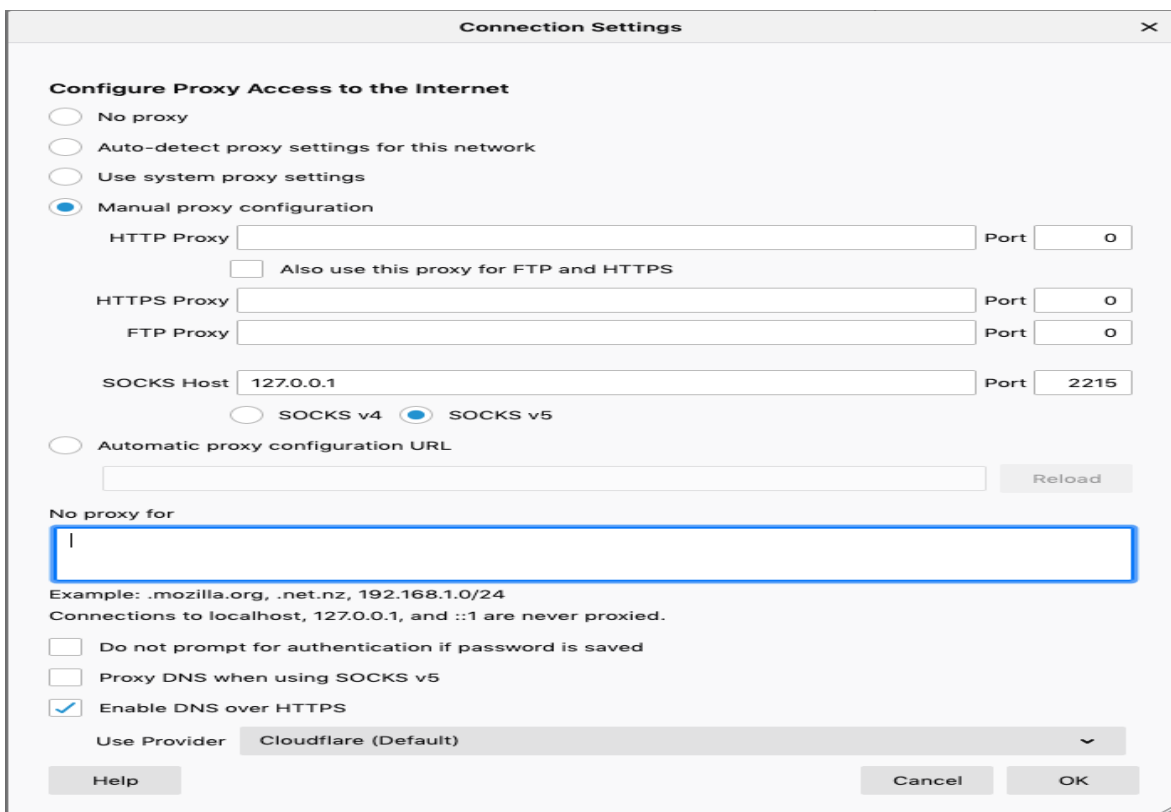
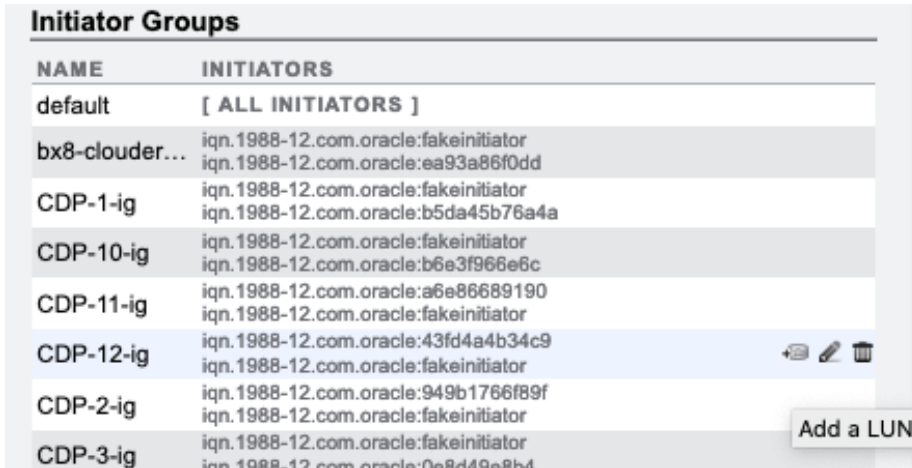


Figure 21: Browser Proxy Settings

Appendix B – Add Additional iSCSI LUNs to a Cloudera Node

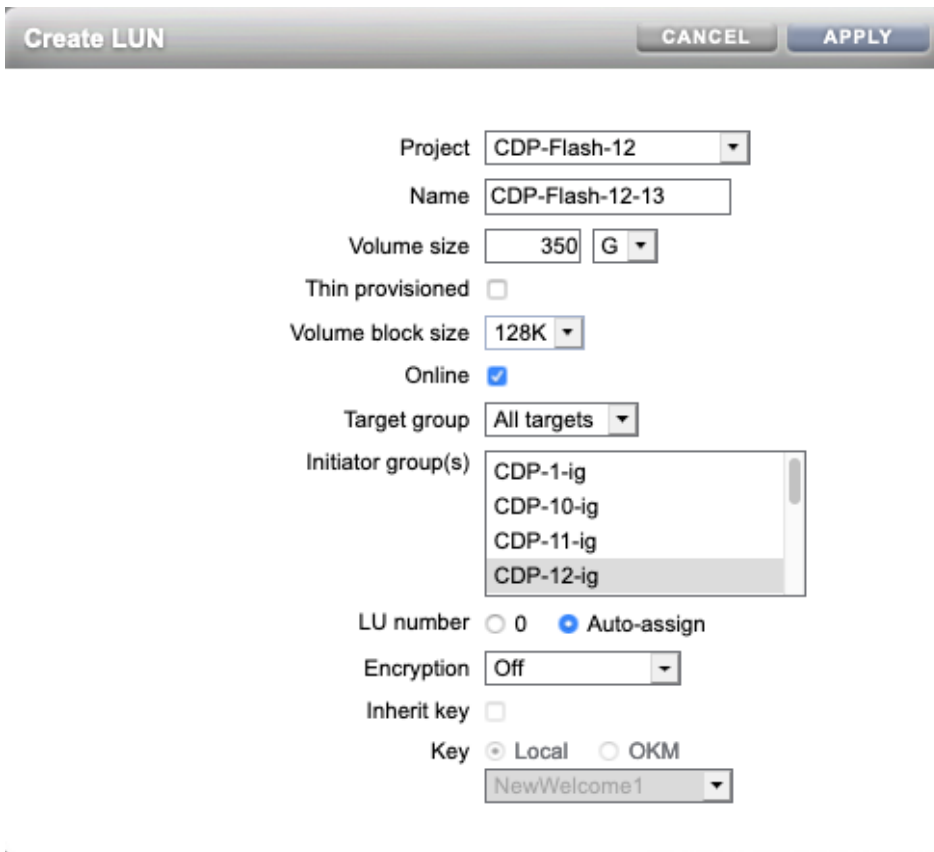
iSCSI LUNs can be added to an existing Oracle Linux 7 VM without rebooting.

1. Login to the ZFS Storage Appliance BUI and navigate to Configuration->SAN-iSCSI->Initiators
2. Select the Initiator Group for the VM to which LUNs will be added
3. Click the “+” next to the disk icon.



NAME	INITIATORS
default	[ALL INITIATORS]
bx8-clouder...	iqn.1988-12.com.oracle:fakeinitiator iqn.1988-12.com.oracle:ea93a86f0dd
CDP-1-ig	iqn.1988-12.com.oracle:fakeinitiator iqn.1988-12.com.oracle:b5da45b76a4a
CDP-10-ig	iqn.1988-12.com.oracle:fakeinitiator iqn.1988-12.com.oracle:b6e3f966e6c
CDP-11-ig	iqn.1988-12.com.oracle:a6e86689190 iqn.1988-12.com.oracle:fakeinitiator
CDP-12-ig	iqn.1988-12.com.oracle:43fd4a4b34c9 iqn.1988-12.com.oracle:fakeinitiator
CDP-2-ig	iqn.1988-12.com.oracle:949b1766f89f iqn.1988-12.com.oracle:fakeinitiator
CDP-3-ig	iqn.1988-12.com.oracle:0e8d49a8b4

4. The “Create LUN” dialog box will open. Specify parameters, ensuring that the correct Project to own the LUN is selected. Click “Apply” when finished. The LUN will be created.



Create LUN [CANCEL] [APPLY]

Project: CDP-Flash-12

Name: CDP-Flash-12-13

Volume size: 350 G

Thin provisioned:

Volume block size: 128K

Online:

Target group: All targets

Initiator group(s): CDP-1-ig, CDP-10-ig, CDP-11-ig, CDP-12-ig

LU number: 0 Auto-assign

Encryption: Off

Inherit key:

Key: Local OKM

NewWelcome1

5. Login as root to the Oracle Linux 7 VM that will own the LUN. List the /dev/mapper directory to know which LUNs are already mapped to dm-multipath.

```
[root@cloudera-12 ~]# ls /dev/mapper/mpath*
```

```
/dev/mapper/mpatha /dev/mapper/mpathb1 /dev/mapper/mpathd /dev/mapper/mpathe1  
/dev/mapper/mpathg /dev/mapper/mpathh1 /dev/mapper/mpathj /dev/mapper/mpathk1  
/dev/mapper/mpathm /dev/mapper/mpathp /dev/mapper/mpatha1 /dev/mapper/mpathc  
/dev/mapper/mpathd1 /dev/mapper/mpathf /dev/mapper/mpathg1 /dev/mapper/mpathi  
/dev/mapper/mpathj1 /dev/mapper/mpathl /dev/mapper/mpathn /dev/mapper/mpathp1  
/dev/mapper/mpathb /dev/mapper/mpathc1 /dev/mapper/mpathe /dev/mapper/mpathf1  
/dev/mapper/mpathh /dev/mapper/mpathi1 /dev/mapper/mpathk /dev/mapper/mpathl1  
/dev/mapper/mpatho
```

6. Rescan the iSCSI bus to find the new LUN:

```
[root@cloudera-12 ~]# iscsiadm -m session --rescan
```

```
Rescanning session [sid: 1, target: iqn.1986-03.com.sun:02:2647063d-bbb0-4ec0-b70c-  
ba15d6580444, portal: 10.10.70.100,3260]
```

```
Rescanning session [sid: 2, target: iqn.1986-03.com.sun:02:2bb7232a-4b80-4f70-97dd-  
e923b827b2f5, portal: 10.10.68.100,3260]
```

```
Rescanning session [sid: 3, target: iqn.1986-03.com.sun:02:3bb1c464-2176-498b-ad0f-  
d725206db1d8, portal: 10.10.69.100,3260]
```

```
Rescanning session [sid: 4, target: iqn.1986-03.com.sun:02:7ad613e2-96d8-419f-b15d-  
f59d3d4ff23f, portal: 10.10.67.100,3260]
```

7. List the /dev/mapper directory again to find the new /dev/mpath[x] entry created for the new LUN:

```
[root@cloudera-12 ~]# ls /dev/mapper/mpath*
```

```
/dev/mapper/mpatha /dev/mapper/mpathb1 /dev/mapper/mpathd /dev/mapper/mpathe1  
/dev/mapper/mpathg /dev/mapper/mpathh1 /dev/mapper/mpathj /dev/mapper/mpathk1  
/dev/mapper/mpathm /dev/mapper/mpathp  
  
/dev/mapper/mpatha1 /dev/mapper/mpathc /dev/mapper/mpathd1 /dev/mapper/mpathf  
/dev/mapper/mpathg1 /dev/mapper/mpathi /dev/mapper/mpathj1 /dev/mapper/mpathl  
/dev/mapper/mpathn /dev/mapper/mpathp1  
  
/dev/mapper/mpathb /dev/mapper/mpathc1 /dev/mapper/mpathe /dev/mapper/mpathf1  
/dev/mapper/mpathh /dev/mapper/mpathi1 /dev/mapper/mpathk /dev/mapper/mpathl1  
/dev/mapper/mpatho /dev/mapper/mpathq
```

8. Use fdisk to create a partition on the new LUN:

```
[root@cloudera-12 ~]# fdisk /dev/mapper/mpathq
```

```
Welcome to fdisk (util-linux 2.23.2).
```

```
Changes will remain in memory only, until you decide to write them.
```

```
Be careful before using the write command.
```

```
Device does not contain a recognized partition table
```

```
Building a new DOS disklabel with disk identifier 0x1ec1250d.
```

```
Command (m for help):
```

```
Command (m for help): n
```

```
Partition type:
```

```
  p   primary (0 primary, 0 extended, 4 free)
```

```
  e   extended
```

```
Select (default p):
```

```
Using default response p
```

```
Partition number (1-4, default 1):
```

```
First sector (2048-734003199, default 2048):
```

```
Using default value 2048
```

```
Last sector, +sectors or +size{K,M,G} (2048-734003199, default 734003199):
```

```
Using default value 734003199
```

```
Partition 1 of type Linux and of size 350 GiB is set
```

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
WARNING: Re-reading the partition table failed with error 22: Invalid argument.
```

```
The kernel still uses the old table. The new table will be used at
```

```
the next reboot or after you run partprobe(8) or kpartx(8)
```

```
Syncing disks.
```

9. Use the `kpartx` command to ensure that `dm-multipath` finds the new partition, then list `/dev/mapper` again to be sure the new partition is listed:

```
[root@cloudera-12 ~]# kpartx -u /dev/mapper/mpathq
```

```
[root@cloudera-12 ~]# ls /dev/mapper/mpath*
```

```
/dev/mapper/mpatha /dev/mapper/mpathb1 /dev/mapper/mpathd /dev/mapper/mpathe1  
/dev/mapper/mpathg /dev/mapper/mpathh1 /dev/mapper/mpathj /dev/mapper/mpathk1  
/dev/mapper/mpathm /dev/mapper/mpathp /dev/mapper/mpathq1  
  
/dev/mapper/mpatha1 /dev/mapper/mpathc /dev/mapper/mpathd1 /dev/mapper/mpathf  
/dev/mapper/mpathg1 /dev/mapper/mpathi /dev/mapper/mpathj1 /dev/mapper/mpathl  
/dev/mapper/mpathn /dev/mapper/mpathp1  
  
/dev/mapper/mpathb /dev/mapper/mpathc1 /dev/mapper/mpathe /dev/mapper/mpathf1  
/dev/mapper/mpathh /dev/mapper/mpathi1 /dev/mapper/mpathk /dev/mapper/mpathl1  
/dev/mapper/mpatho /dev/mapper/mpathq
```

10. Use mkfs to create a new filesystem on the partition:

```
[root@cloudera-12 ~]# mkfs.ext4 /dev/mapper/mpathq1
```

11. Create a directory, add the new filesystem to /etc/fstab, and mount:

```

[root@cloudera-12 ~]# mkdir /cdp13
[root@cloudera-12 ~]# vi /etc/fstab

# /etc/fstab
# Created by anaconda on Tue Jun  2 08:50:49 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/ol-root    /                    xfs     defaults    0 0
UUID=adf16f1d-f170-410a-9b87-ff8d2ddaac91 /boot                xfs     defaults    0
/dev/mapper/ol-hadoop /scratch             ext4    defaults    0 0
/dev/mapper/ol-home   /home                ext4    defaults    1 2
/dev/mapper/ol-swap   swap                 swap    defaults    0 0
.
.
.
/dev/mapper/mpathh1   /cdp08               ext4    defaults    0 0
/dev/mapper/mpathi1   /cdp09               ext4    defaults    0 0
/dev/mapper/mpathj1   /cdp10               ext4    defaults    0 0
/dev/mapper/mpathk1   /cdp11               ext4    defaults    0 0
/dev/mapper/mpathl1   /cdp12               ext4    defaults    0 0
/dev/mapper/mpathq1   /cdp13               ext4    defaults    0 0

[root@cloudera-12 ~]# mount /cdp13
[root@cloudera-12 ~]# ls /cdp13

```

Connect with us

Call +1.800.ORACLE1 or visit [oracle.com](https://www.oracle.com). Outside North America, find your local office at: [oracle.com/contact](https://www.oracle.com/contact).

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.