Virtual Machine Access to the Internal ZS7-2 in Private Cloud Appliance 2.4.3

Enabling direct software defined network access to the internal ZFS Storage Appliance

March, 2021, Version 1.1

Public

Purpose statement

This document explores a new capability delivered with Private Cloud Appliance software release 2.4.3, whereby virtual machines may access the internal ZFS Storage Appliance directly over the network. This approach is distinguished from existing methods presenting storage as virtual block devices.

This document is intended to explain software design decisions related to this capability and to provide illustrative examples of its usage.

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

List of images

List of tables

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 virtualized guest operating system ("OS").

Beginning with the PCA X8-2 hardware release, the original 40Gb Infiniband network was replaced by a 100Gb Ethernet network, employing standard datacenter networking protocols to enable PCA's software defined networking. PCA software release 2.4.3 utilizes this updated technology stack to provide additional flexibility in connecting to the internal ZFS Storage Appliance ("ZFSSA"). Whereas in previous releases, the internal storage could only be consumed in the guest OS via virtual block devices, customers now have the option of choosing additional storage protocols utilizing direct peer-to-peer networking connectivity between the VM and the internal ZFSSA.

In this document, we will explore the implementation of these new storage networking capabilities with the aim of understanding the underlying design philosophy. We will also walk through several examples related to common use cases.

Software & Hardware Requirements

- PCA X8-2 or newer hardware revision, with Ethernet based networking
- PCA 2.4.3 or newer software release

Accessing Storage from Virtual Machines

With PCA prior to the 2.4.3 software release, access to the internal storage is provided through the use of physical and virtual disks. Figure 1 depicts these access patterns.

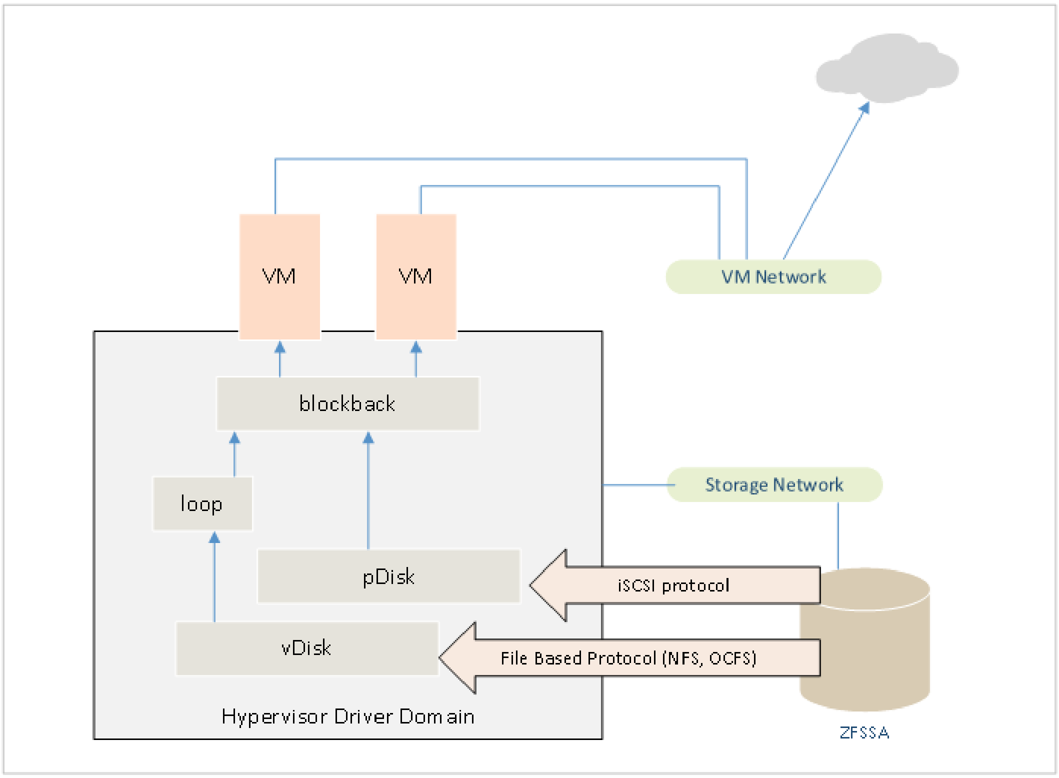


Figure 1 - VMs with Block Attached Storage

ZFSSA storage is exposed to the hypervisor layer via the Storage Network using either NFS or iSCSI protocol. Virtual Machine ("VM") access is then provided through the blockfront / blockback driver, such that the guest only sees a virtual block device. With this pattern, there is no direct network connectivity between the VM and the ZFSSA.

This existing access pattern is still fully supported. The new approaches discussed in this document are complementary and should in no way be construed as to imply deprecation of the physical / virtual disk access patterns. On the contrary, the use of virtual disks or physical disks is *required* for VM boot and will remain an optimal approach for many use cases.

Beginning with PCA software release 2.4.3, for the first time, it is also possible to run NFS and iSCSI protocols directly in the VM to access shares on the internal ZFSSA via direct peer-to-peer network connectivity. This new capability is enabled on PCA software release 2.4.3 with both fresh installations and with upgrades. The new access pattern is depicted in Figure 2.

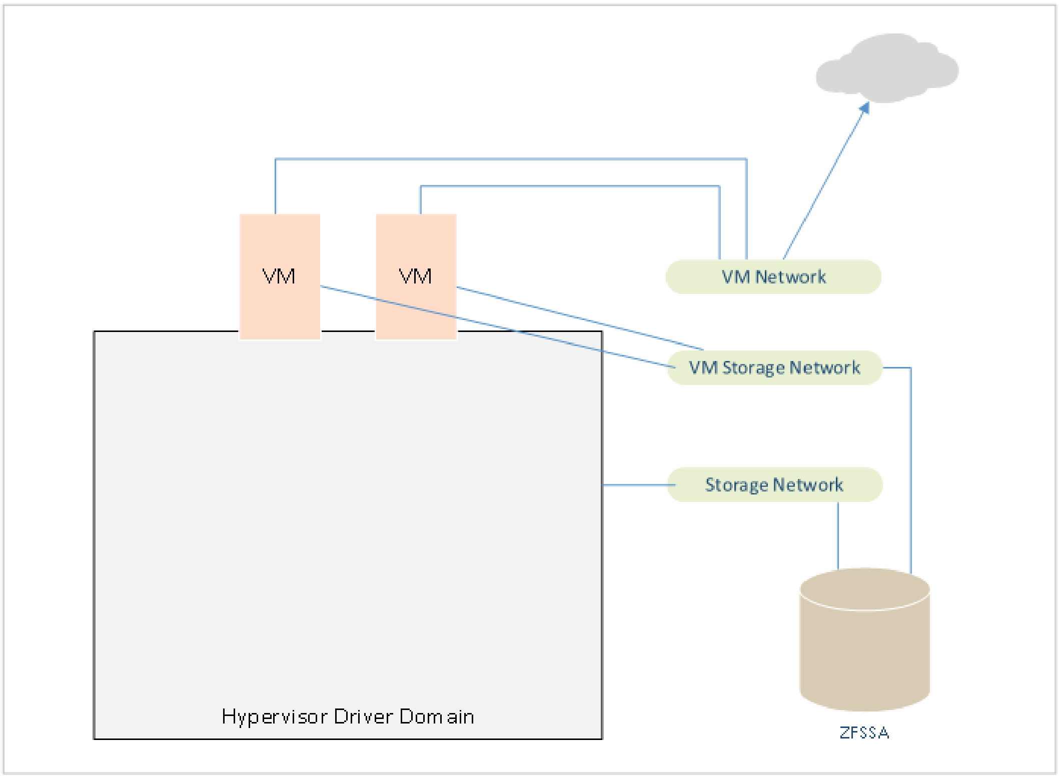


Figure 2 - VMs with Network Attached Storage

This storage access is enabled through the creation of a new network type in pca-admin, called the VM Storage Network. A **VM Storage Network** is a network with the Virtual Machine networking role assigned in Oracle VM Manager ("OVMM"), but which also exposes an interface on the ZFSSA. The intended use case is to provide for non-routed backend networks dedicated solely for accessing the internal ZFSSA. With PCA software release 2.4.3, up to 16 VM Storage Networks can be created.

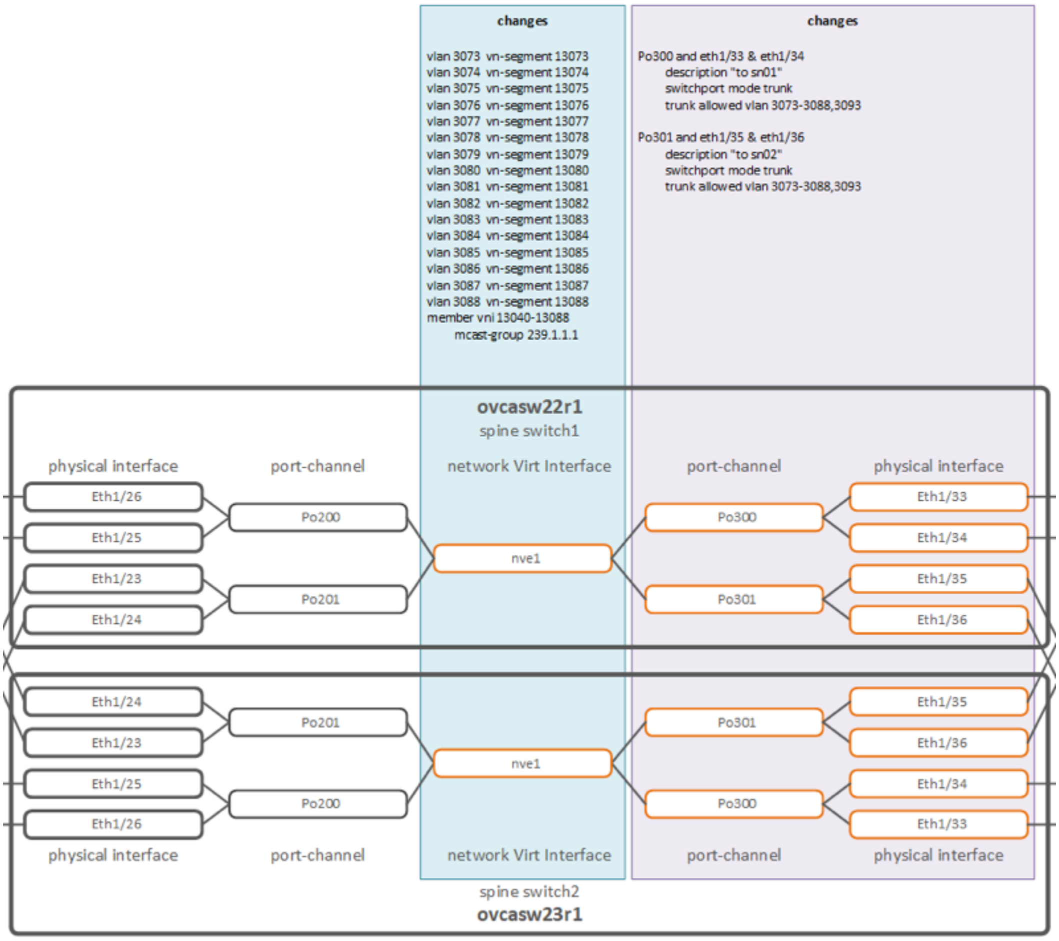
The steps a PCA administrator will undertake to configure this access are:

- Create a VM Storage Network
- Add the VM Storage Network to a Tenant Group
- Create an NFS or iSCSI share
- Attach one or more VMs to the VM Storage Network
- Allow one or more VMs to access the share

Command syntax and examples are provided in the following sections, but first we'll examine the implementation.

Network Changes from Previous Software Releases

Within the Ethernet switch configuration, 16 VXLAN / VLAN pairs have been preallocated as part of the release 2.4.3 installation or upgrade. Figure 3 shows the configuration changes implemented on the PCA spine switches.



changes

```
vlan 3073  vn-segment 13073
vlan 3074  vn-segment 13074
vlan 3075  vn-segment 13075
vlan 3076  vn-segment 13076
vlan 3077  vn-segment 13077
vlan 3078  vn-segment 13078
vlan 3079  vn-segment 13079
vlan 3080  vn-segment 13080
vlan 3081  vn-segment 13081
vlan 3082  vn-segment 13082
vlan 3083  vn-segment 13083
vlan 3084  vn-segment 13084
vlan 3085  vn-segment 13085
vlan 3086  vn-segment 13086
vlan 3087  vn-segment 13087
vlan 3088  vn-segment 13088
member vni 13040-13088
      mcast-group 239.1.1.1
```

changes

```
Po300 and eth1/33 & eth1/34
      description "to sn01"
      switchport mode trunk
      trunk allowed vlan 3073-3088,3093

Po301 and eth1/35 & eth1/36
      description "to sn02"
      switchport mode trunk
      trunk allowed vlan 3073-3088,3093
```

Figure 3 - PCA Spine Switch Configuration

In addition to preallocating the 16 VXLAN / VLAN pairs, the switchport mode is set to trunk on the port-channel connected to the ZFSSA, and the 16 VLANs are enabled on that trunk.

Figure 4 illustrates the VM Storage Network implementation on the compute nodes. The implementation is the same as for other VM networks.

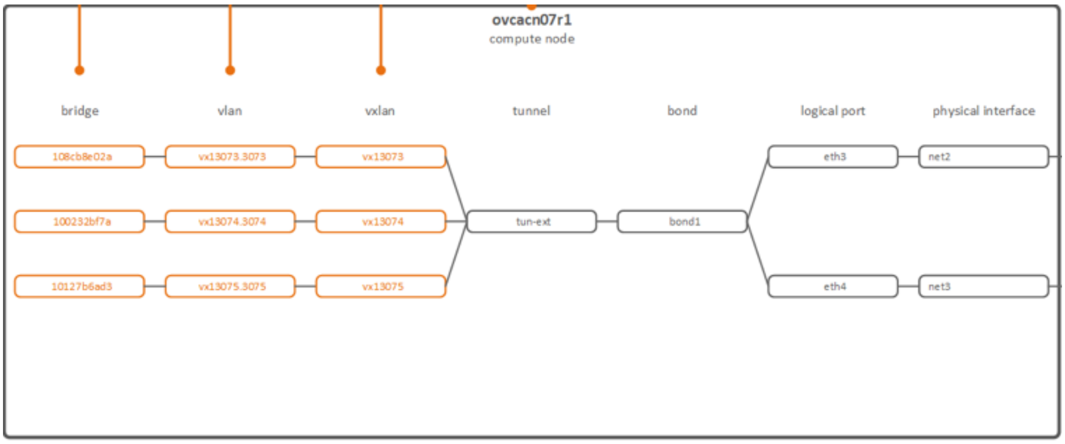


Figure 4 - Compute Node Network Configuration

A strict one to one mapping exists between VM Storage Network VXLAN / VLAN pairs. Additional VLANs should not be created on these VXLAN interfaces. With other network types, a PCA administrator can, if desirable, create many VLANs on top of each VXLAN interface, but in the case of VM storage networks, that configuration would not be meaningful.

The reason for this strict one to one mapping can more easily be seen in Figure 5 below, depicting the network configuration on the ZFSSA.

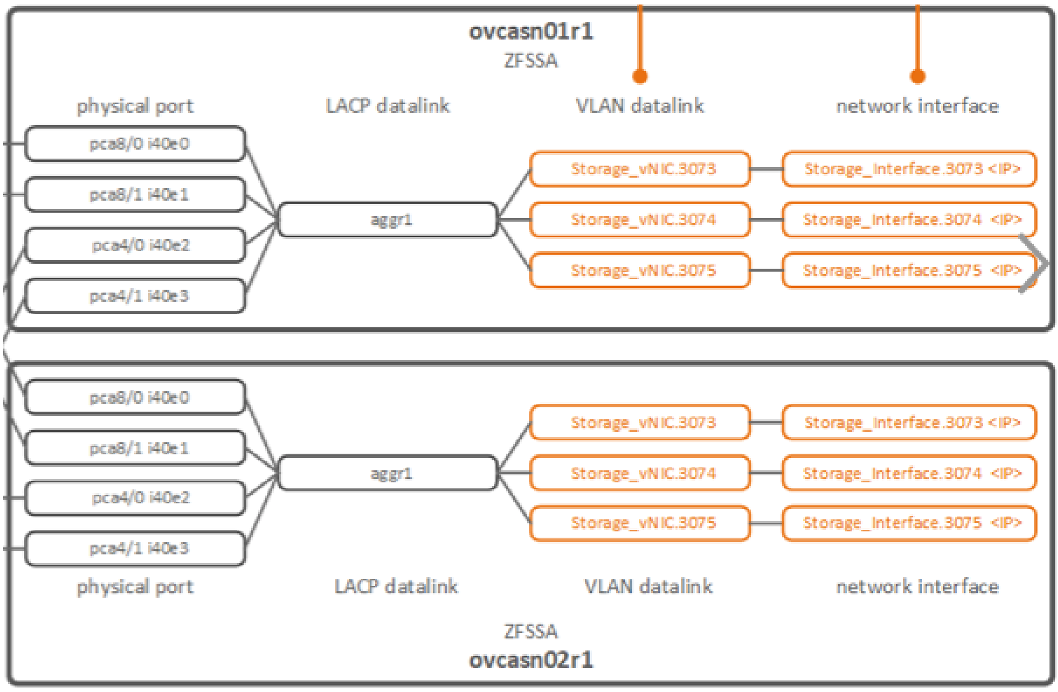
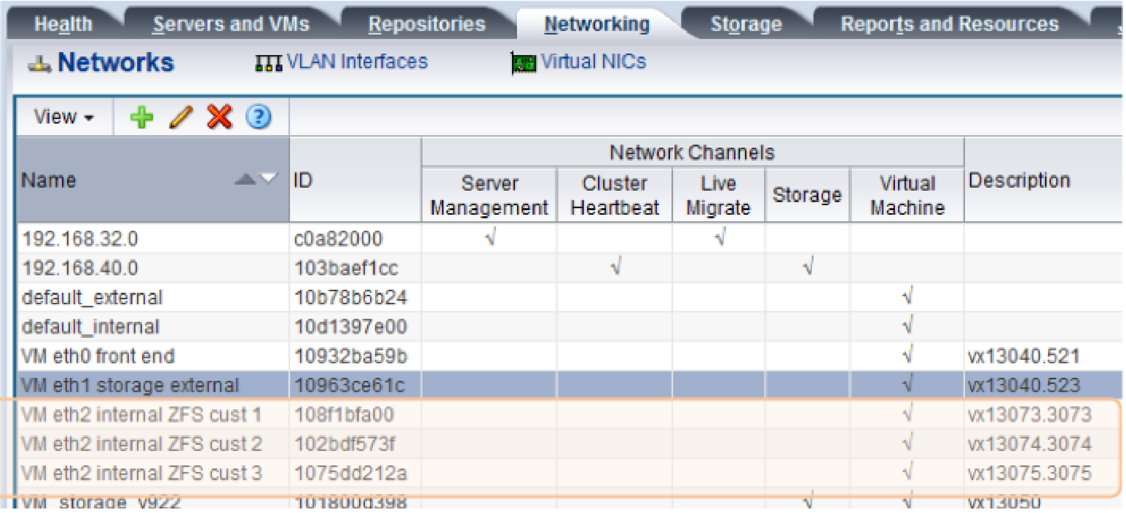


Figure 5 - ZFSSA Network Configuration

Note the absence of any VXLAN interfaces in the ZFSSA network configuration. There are no VXLAN interfaces on the ZFSSA because the VXLAN tunnel is terminated on the spine switch. The spine switch decapsulates VXLAN and communicates with the ZFSSA using only standard 802.1q tagged Ethernet. Without any encapsulation or Q-in-Q configured, any double tagged frames would just be dropped.

Introducing VM Storage Networks

In OVMM, VM Storage Networks are assigned the Virtual Machine role and can be used like any other Virtual Machine network. They differ from other Virtual Machine networks in that an internal ZFSSA interface is configured and attached to it, but that fact is only discoverable through pca-admin. With respect to OVMM, VM Storage Networks appear exactly as any other Virtual Machine network.



Name	ID	Network Channels					Description
		Server Management	Cluster Heartbeat	Live Migrate	Storage	Virtual Machine	
192.168.32.0	c0a82000	√		√			
192.168.40.0	103baef1cc		√		√		
default_external	10b78b6b24					√	
default_internal	10d1397e00					√	
VM eth0 front end	10932ba59b					√	vx13040.521
VM eth1 storage external	10963ce61c					√	vx13040.523
VM eth2 internal ZFS cust 1	108f1bfa00					√	vx13073.3073
VM eth2 internal ZFS cust 2	102bdf573f					√	vx13074.3074
VM eth2 internal ZFS cust 3	1075dd212a					√	vx13075.3075
VM storage v922	101800d398				√	√	vx13050

Figure 6 - OVMM Network Listing

Figure 6 displays a sample list of networks in OVMM. All of the highlighted networks are VM Storage Networks assigned to the Virtual Machine role. This configuration is done automatically by pca-admin when a VM Storage Network is created and should not be modified after the fact. In particular, it should be noted that the Storage role in OVMM should **never** be selected for a VM Storage Network, notwithstanding the fact that pca-admin itself confusingly employs the keyword `storage_network` for their creation.

Note that the selected interface labelled "VM eth1 storage external" depicts an approach using a VLAN on the `default_external` interface to provide connectivity to external storage, whereas the highlighted interfaces show how the capability described in this document can be used for connectivity to internal storage.

Preparing the ZFSSA

Prior to creating any VM Storage Networks or creating any LUNs or shares, it is necessary to modify the DNS resolver configuration on the ZFSSA. By default, DNS resolution is not configured for the ZFSSA, which can result in performance degradation for VM NFS clients as described in Oracle KM Note 2755549.1.

The management network for PCA compute nodes is not exposed externally, so the active management node will be used as the DNS nameserver. Log in to the ZFSSA active node and issue the following commands:

```
ovcasn01r1:> configuration services nfs
ovcasn01r1:configuration services nfs> set mapid_domain=localdomain
                  mapid_domain = localdomain (uncommitted)
ovcasn01r1:configuration services nfs> commit
ovcasn01r1:configuration services nfs> up dns
ovcasn01r1:configuration services dns> set search=localdomain
                        search = localdomain (uncommitted)
ovcasn01r1:configuration services dns> set servers=192.168.4.216
                       servers = 192.168.4.216 (uncommitted)
ovcasn01r1:configuration services dns> commit
ovcasn01r1:configuration services dns> exit
```

Managing VM Storage Networks

New pca-admin commands are available to create and manage VM Storage Networks. When a VM Storage Network is created using the pca-admin command line interface ("CLI"), the system will associate one of the preallocated VXLAN / VLAN pairs, assign the requested address space, and configure a new network interface on the ZFSSA.

A VM Storage Network should only be created and managed using pca-admin. Do not attempt to create them using OVMM or the ZFSSA UI.

Creating a VM Storage Network

The pca-admin CLI syntax to create a VM Storage Network is straightforward:

```
create network <network_name> storage_network <prefix> <netmask> <zfs_ip>
```

<network_name>	Desired name for the network.
<prefix>	Network prefix, must be three octets.
<netmask>	Netmask in dotted decimal form.
<zfs_ip>	Desired IPv4 address for the ZFSSA interface.

Example:

```
create network vmstor1_net storage_network 10.10.1 255.255.255.0 10.10.1.200
```

Caveat

Orabug 32054739 affects network prefix specification for the pca-admin create network subcommand. The parser expects exactly three octets to be specified for the prefix, regardless of the netmask. In some cases, the bug is a mild nuisance, but in others it presents a problem for which there is no workaround. For example, it is currently not possible to specify the prefix 172.31.0.128/25, where the fourth octet is non-zero. Until this bug is fixed, PCA administrators may wish to avoid using any prefixes smaller than /24 when creating VM Storage Networks.

Adding a VM Storage Network to Compute Nodes

After creating the VM Storage Network, it must be added to any compute nodes where it should be available for use by VMs. The CLI syntax is as follows:

```
add network <network_name> <compute_node>
```

```
add network-to-tenant-group <network_name> <tenant_group>
```

<network_name>	Name of the VM Storage Network.
<compute_node>	Name of the compute node.
<tenant_group>	Name of the tenant group.

Examples:

```
add network vmstor1_net ovcacn08r1
```

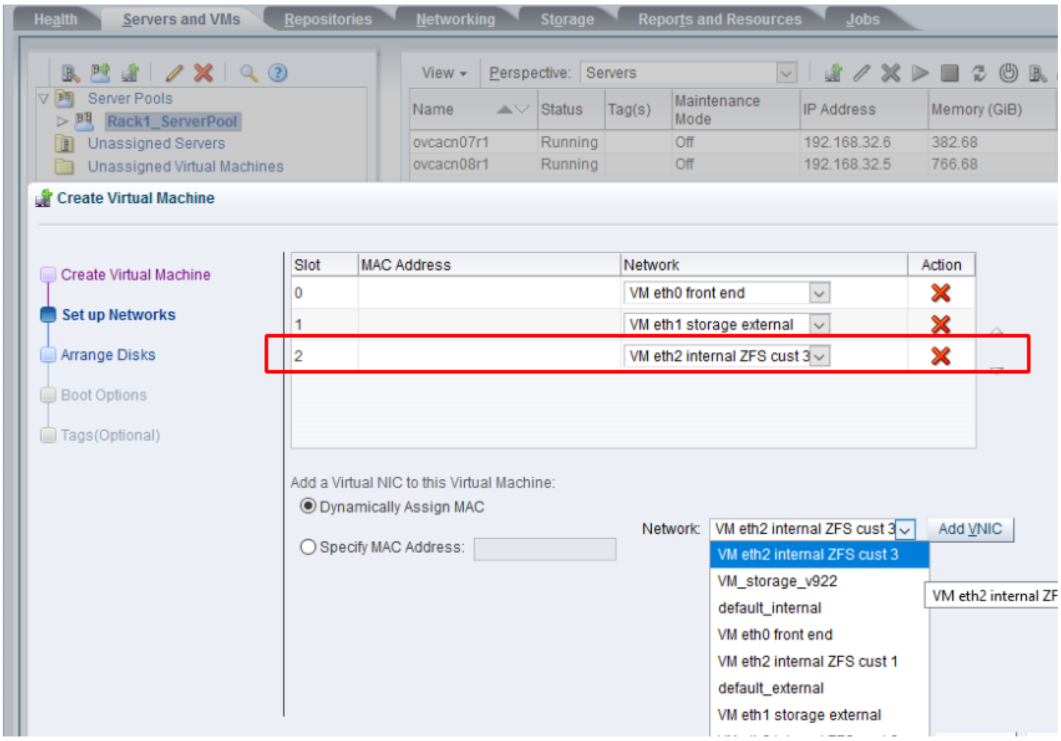
```
add network-to-tenant-group vmstor1_net Rack1_ServerPool
```

In general, the network-to-tenant-group variant is preferred, since it will cause the network to be automatically added to any compute nodes that are subsequently added to the tenant group.

Invocation of this command will cause the system to create the VXLAN interface on the compute node, then refresh the node so that OVMM sees the interface on the node, and finally associate that interface with the network object in OVMM.

Using VM Storage Networks

Once a VM Storage Network is created and added to compute nodes, simply select them for use in the VM configuration, as would be done for any other network. It is recommended to configure a routed network on the first interface and as many non-routed storage subnets as are needed.



Name	Status	Tag(s)	Maintenance Mode	IP Address	Memory (GiB)
ovcacn07r1	Running		Off	192.168.32.6	382.68
ovcacn08r1	Running		Off	192.168.32.5	766.68

Create Virtual Machine

Slot	MAC Address	Network	Action
0		VM eth0 front end	✗
1		VM eth1 storage external	✗
2		VM eth2 internal ZFS cust 3	✗

Add a Virtual NIC to this Virtual Machine:

● Dynamically Assign MAC

○ Specify MAC Address:

Network: VM eth2 internal ZFS cust 3

Figure 7 - VM Network Configuration

Figure 7 shows VM network configuration in progress. This particular PCA is equipped with compute nodes with extra interfaces to access external storage. On this VM, the user has configured a frontend network, a second interface for the external storage, and finally a third interface connected to a VM Storage Network.

Additional VM Storage Network Commands

The following commands are also relevant for VM Storage Networks:

- list network
- show network <network_name>
- remove network <network_name> <compute_node>
- remove network-from-tenant-group <network_name> <tenant_group>
- delete network <network_name>

Managing Storage Shares

NFS shares and iSCSI LUNs on the internal ZFSSA are also managed with pca-admin, but only in the default OVCA_POOL. LUNs and NFS shares in any additional pools must be created using the ZFSSA UI, and not with pca-admin. Any such LUNs or shares created directly on the ZFSSA can still use VM Storage Networks. Simply export them to the appropriate IP addresses or CIDRs using standard ZFSSA procedures.

During 2.4.3 installation or upgrade, a project called `VMINTERNAL` is created on the default OVCA_POOL on the internal ZFSSA. The project level reservation and quota are set to 60TB, and all NFS shares and iSCSI LUNs created with pca-admin will be placed into this project. This arrangement is intended to ensure that customer data does not grow beyond 60TB in the default OVCA_POOL. If customer data were to fill up the default OVCA_POOL, critical PCA services such as Oracle VM Manager could be adversely affected. If customers require more than 60TB, or if commingling customer shares into a single VMINTERNAL project is unworkable, it is recommended to purchase additional disk shelves and create a separate storage pool. The flowchart in Figure X illustrates when pca-admin should be used versus when the ZFSSA UI is more appropriate.

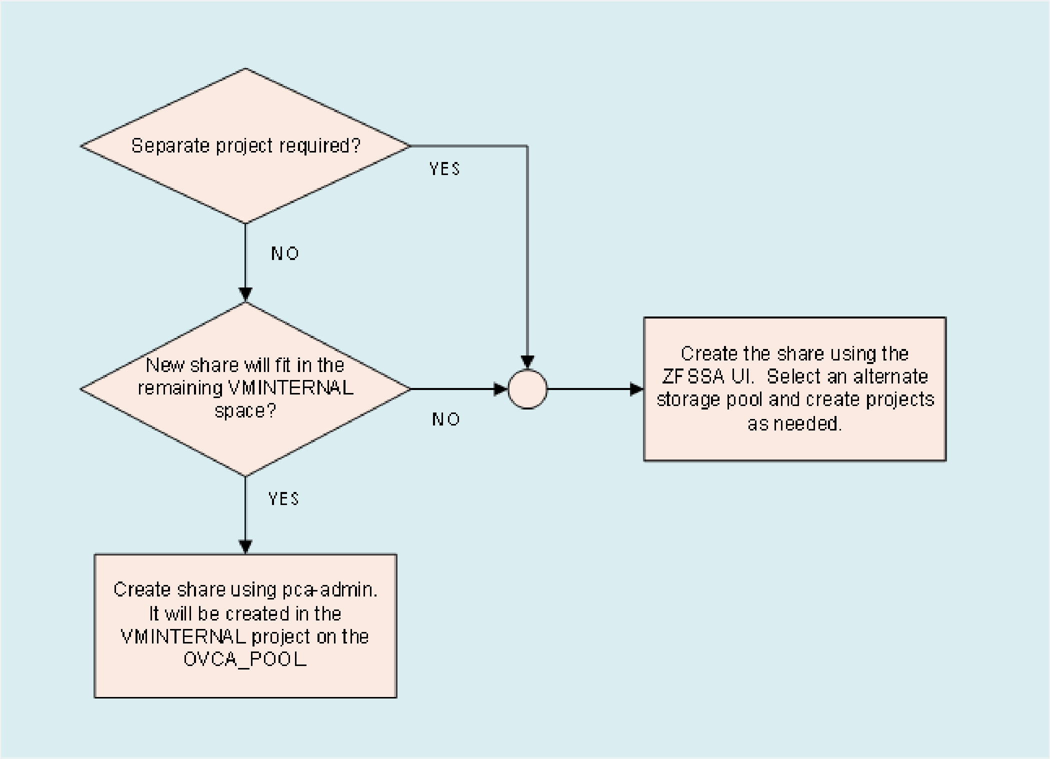

Separate project required?
YES
NO
New share will fit in the remaining VMINTERNAL space?
NO
YES
Create the share using the ZFSSA UI. Select an alternate storage pool and create projects as needed.
Create share using pca-admin. It will be created in the VMINTERNAL project on the OVCA_POOL.

Figure 8 - Share / LUN Creation Flowchart

Storage Profiles

A storage profile is a predefined collection of share properties. The storage profile can be referenced during share or LUN creation to easily configure for common workloads. In PCA software release 2.4.3, three storage profiles are defined:

Storage Profile	NFS Properties	iSCSI Properties

general (default)	compression=lz4
	logbias=latency
	recordsize=128k
		compression=lz4
		logbias=latency
		volblocksize=128k

dbms_oracle	compression=lz4
	logbias=latency
	recordsize=32k
		compression=lz4
		logbias=latency
		volblocksize=32k

bkup_basic	compression=lz4
	logbias=throughput
	recordsize=1M
		compression=lz4
		logbias=throughput
		volblocksize=1M

Table 1 - Storage Profiles

Storage profiles are not configurable via the pca-admin CLI. If you need to create a share using other share properties, you must create it manually using the ZFSSA UI rather than with pca-admin. In particular, if encryption is required, you must use the ZFSSA UI to create the share, as encryption can only be enabled at the time of initial share creation.

Managing NFS Shares

To create an NFS share with pca-admin, the syntax is as follows:

```
create nfs-storage <share_name> <network_name> <size> [profile_name]
```

<share_name>	Desired name for the NFS share.
<network_name>	Name of the VM Storage Network.
<size>	Desired size of the share.
[profile_name]	Optional storage profile name. Defaults to **general**.

Examples:

```
create nfs-storage images vmstor1_net 100g
```

```
create nfs-storage PACMEX_oradata vmstor1_net 50g dbms_oracle
```

Note that <share_name> must be unique across the entire ZFSSA, so PCA reserved names such as MGMT_ROOT cannot be used.

NFS Exceptions

After initially creating a share, although it's associated with a particular subnet, it is not actually exported to any hosts, and therefore it cannot be mounted by any VM. Adding exports is accomplished by adding so-called NFS exceptions, terminology derived from the ZFSSA UI. ZFSSA has a top level access policy per share which pca-admin configures to deny all access, and then access is granted by creating NFS exceptions to that top level policy.

The syntax for adding an NFS exception is as follows:

```
add nfs-exception <share_name> <exception>
```

<share_name>	Desired name for the NFS share.
<exception>	IPv4 address or CIDR range of allowed NFS clients.

Examples:

```
add nfs-exception images 10.10.1.11
```

```
add nfs-exception html 10.10.1.12/30
```

Recall that when initially creating an NFS share with pca-admin, it's required to specify an existing VM Storage Network. The system will record the network specified during creation, and pca-admin will refuse to add NFS exceptions for any NFS client which is not in that network. This restriction may be relaxed in a future release. For now, if it's required to mount a share from more than one VM Storage Network, simply edit the NFS exceptions directly in the ZFSSA UI rather than through pca-admin. This practice is safe, insofar as pca-admin is careful not to overwrite manual changes made to NFS exceptions via the ZFSSA UI. Manual changes will not be reflected, however, in the output of pca-admin commands such as "show nfs-storage".

Additional NFS Share Commands

The following commands are also relevant for managing NFS shares:

- remove nfs-exception <share_name> <exception>
- delete nfs-storage <share_name>

Managing iSCSI LUNs

To create an iSCSI LUN, the CLI syntax is as follows:

create iscsi-storage <lun_name> <network_name> <size> [profile_name]

<lun_name>	Desired name for the LUN.
<network_name>	Name of the existing VM Storage Network.
<size>	Desired size for the LUN.
[profile_name]	Optional storage profile name. Defaults to **general**.

Example:

```
create iscsi-storage documents vmstor1_net 100G general
```

As with NFS shares, <lun_name> must be unique, and the VM Storage Network specified by <network_name> must exist.

Adding iSCSI Initiators

To associate iSCSI initiators with an iSCSI LUN, the CLI syntax is as follows:

```
add initiator <initiator_iqn> <lun_name>
```

<initiator_iqn>	Existing initiator iqn, configured in a VM.
<lun_name>	Name of the iSCSI LUN.

Example:

```
add initiator iqn.1988-12.com.oracle:bddc42b9b14 documents
```

Organization of Initiator Groups

Whenever a new iSCSI volume is created with pca-admin, a new initiator group is also created and immediately associated with the new volume, so that a one to one relationship exists between volumes and initiator groups on the ZFSSA. When `pca-admin add initiator` is later invoked, the specified initiator is simply added into the pre-existing group for the corresponding volume.

This approach is not aligned with published ZFSSA best practices, but LUN conflicts are avoided through the use of a "fake" initiator, aptly named `fakeinitiator`. Each time a new iSCSI volume and initiator group are created, `fakeinitiator` is added to the group prior to its association with the volume. This association causes a unique LUN to be assigned for all iSCSI volumes which are created using pca-admin.

Should the need ever arise to create an iSCSI volume directly on the ZFSSA UI rather than with pca-admin, it is recommended that this same approach be followed. Create a dedicated initiator group, then add `fakeinitiator` to the group, and finally associate the new volume with the group. Following this same approach should ensure that no LUN conflicts occur when storage clients need to attach some volumes created via pca-admin and also others created directly in the ZFSSA UI.

Additional iSCSI LUN Commands

The following commands are also relevant for managing iSCSI LUNs:

- remove initiator <initiator_iqn> <lun_name>
- delete iscsi-storage <lun_name>

Upgrade

It should be noted that an entirely new step was added in the upgrade process to enable VM Storage Network functionality. The upgrade step is referenced by the "storage-network" keyword, and the syntax is as follows:

- Verify mode: pca_upgrader -V -t storage-network
- Upgrade mode: pca_upgrader -U -t storage-network

Upgrade Sequence

An overview of the upgrade procedure is provided below. Note that the new storage network upgrade step is inserted near the end of the process, just before the upgrade of the compute nodes.

Note also there is a bug which will cause the storage network upgrade to fail consistently. The workaround is simple, consisting of deleting or renaming log files. It can be found in the support note referenced below.

Upgrade Procedure Overview

1. Upgrade both management nodes to PCA software release 2.4.3
2. Upgrade firmware:
   a. System ILOM/BIOS: management, then compute nodes
   b. Switches: leaf, spine, then management
   c. Internal ZFSSA
3. Storage network workaround
   ref: Orabug 31811656, MOS Note 2710199.1
4. Storage network upgrade
5. Compute node upgrade

Upgrade Pitfalls

- Never upload the new spine switch config file that ships with release 2.4.3 prior to storage-network upgrade
  - Management and Compute nodes will lose connectivity to internal ZFSSA and will fence
- Never upload the backed up config file to the spine switches after the storage-network upgrade
  - Management and Compute nodes will lose connectivity to internal ZFSSA and will fence

- To resolve above two situations, run the storage-network upgrade

Upgrade Locks

Locks are placed at various times during upgrade. If PCA administrators are trying to provision resources while the upgrade is in progress, certain requests may block or even fail until the upgrade proceeds far enough to remove the lock. The following locks are employed during the upgrade:

- all_provisioning.LOCK
  - To prevent provisioning and de-provisioning of compute nodes during storage-network upgrade
- fw_upgrade.LOCK
  - Placed after completion of first management node upgrade, cleared after successful completion of storage-network upgrade
- storage_network_upgrade.LOCK
  - Placed at the start of storage-network upgrade, prevents changes to configs in spine and leaf switches during upgrade

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com   facebook.com/oracle   twitter.com/oracle

Disclaimer: This document is for informational purposes. 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, timing, and pricing of any features or