Oracle Elite
Engineering Exchange

Implementing Application Zones on Oracle SuperCluster

ORACLE WHITE PAPER | JULY 2015

Table of Contents

Introduction

This paper describes how to implement “app zones,” Oracle Solaris Zones within an application domain of Oracle SuperCluster. It addresses architectural implementations and best practices that are specific to the SuperCluster platform, which should be taken into account when configuring app zones in order to retain many of the availability and performance features of SuperCluster.

This paper explains how to create and configure iSCSI LUNs on the internal Oracle ZFS Storage Appliance (Oracle ZFS-SA) upon which the zone rpool will reside, how to identify and replicate the network plumbing from the global zone into the non-global zone that is to be created, and how to optionally apply CPU resource management to a zone.

It also explains how to configure 10 GbE network interfaces utilizing IP Network Multipathing (IPMP), which is the default, or Link Aggregation Control Protocol (LACP) and with or without VLAN tagging.

This paper directly addresses only Oracle Solaris 11 non-global zones within an Oracle Solaris 11 global zone. It does not address Oracle Solaris 10 zones nor zones within an Oracle Solaris 10 global zone. However, many of the concepts are similar and could be indirectly applied.

This paper also does not address database zones within the database domain, nor the use of Oracle Solaris Cluster within app zones.

The concepts in this paper apply equally to Oracle SuperCluster T5-8 and M6-32, although the specific examples are for Oracle SuperCluster T5-8. From the perspective of app zone creation and configuration, the differences between the Oracle SuperCluster T5-8 and M6-32 are minor and strictly related to network interface names and type. These differences are not currently addressed in this paper.

Conceptual Overview

SuperCluster Logical Network Diagram

SuperCluster consists of the following components relevant to app zones:

- Compute nodes (servers), possibly subdivided into domains
- Oracle ZFS-SA (model Oracle ZFS Storage ZS3-ES)
- InfiniBand network interconnecting the compute nodes with storage
- A management network (used for monitoring, maintenance, and patching)
- A 10 GbE network to provide connectivity from the SuperCluster compute nodes out to the data center environment

Additionally, a SuperCluster also contains Oracle Exadata Storage Servers (sometimes called Oracle Exadata storage cells), InfiniBand switches, and power distribution units (PDUs). However, these are not relevant to creating app zones.

While it is possible to create an app zone with no network interfaces, it might not be very practical.

This paper explains how to replicate the network connectivity within the global zone at the app zone level through the use of network virtualization capabilities of Oracle Solaris 11.

Figure 1 presents a high-level logical view of an Oracle SuperCluster T5-8 in an H2-1[1] configuration with one app zone in each app domain.

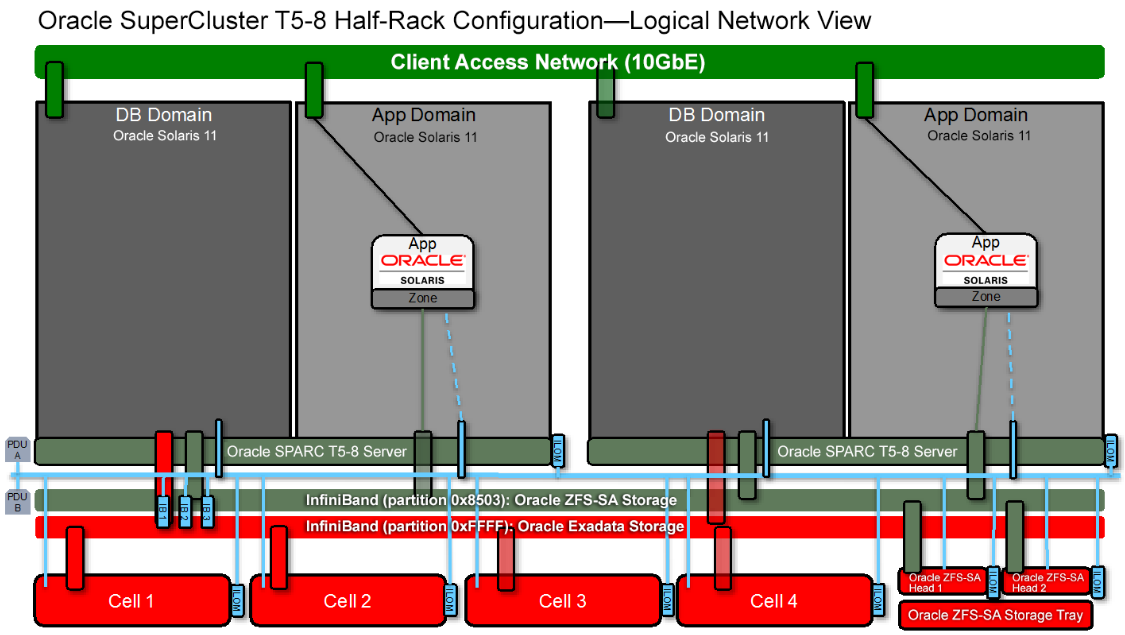


Figure 1. High-level logical view of an Oracle SuperCluster T5-8 in an H2-1 configuration

[1] SuperCluster domain configurations and nomenclature are documented in the *SuperCluster Owner's Guide* in the "Understanding the System" section.

In this example, the app zone is connected to three networks: the 10 GbE client access network, the InfiniBand partition 0x8503, and the internal management network. Note that connectivity to any of these networks is optional, depending on the application. In particular, you might choose not to provide connectivity to the internal management network (because you can always use `zlogin` from the global zone to administer the app zone).

High-Level Steps

The following outlines at a high level the steps to build Oracle Solaris Zones in the app domain (certain details are omitted at this point and are introduced later):

- Create an iSCSI LUN on the Oracle ZFS-SA to hold the Oracle Solaris “rpool” for the zone
- Present the LUN to Oracle Solaris, format it, and create a ZFS “zpool” on it
- Create the zone configuration file, specifying all of the network plumbing
- Install and boot the zone
- Configure network interfaces and services within the zone using a custom script
- (Optional) Create NFS shares on the Oracle ZFS-SA and present them to the zone for additional storage
- (Optional) Assign dedicated CPU resources to the zone

Creating and Configuring iSCSI LUNs

In Oracle Solaris 11, each zone must have its own ZFS root pool or rpool. This rpool can reside on disks within the SuperCluster or on iSCSI LUNs from the Oracle ZFS-SA. It cannot reside on NFS.

This paper explains how to create and configure iSCSI LUNs for the zone’s rpool. It does not address using local storage.

There are two methods for configuring iSCSI LUNs. The recommended best practice is to create one iSCSI LUN per domain (therefore, per global zone) such that each zone that is created gets its own ZFS dataset from that pool for its own rpool.

This method has the advantage that the iSCSI LUN need only be configured once, and there is less administration work to perform on the Oracle ZFS-SA. All zones that are created within that domain can share the space, and if zones are cloned, then ZFS clones can be utilized to save space. If additional space is needed later, the pool can be dynamically expanded.

The other method is to create one iSCSI LUN per zone. This method has the advantage that zones can be replicated individually (for example, for backup) to another Oracle ZFS-SA by replicating the LUN. It also means that a zone can be easily moved to another domain if desired (although the use case for this is rare on SuperCluster). It has the disadvantage of having to manage multiple LUNs and ZFS pools.

Figure 2 and Figure 3 illustrate both scenarios.

SuperCluster: **iSCSI Zone Storage – One LUN per Domain**

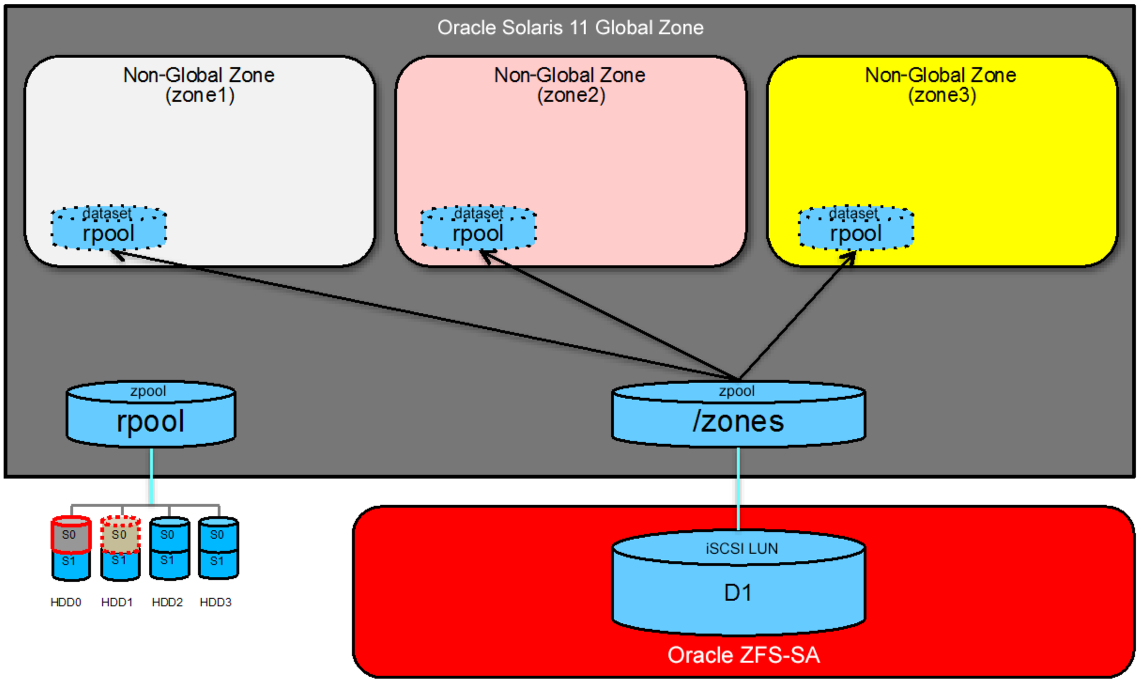


Figure 2. One LUN per domain

SuperCluster: **iSCSI Zone Storage – One LUN per Zone**

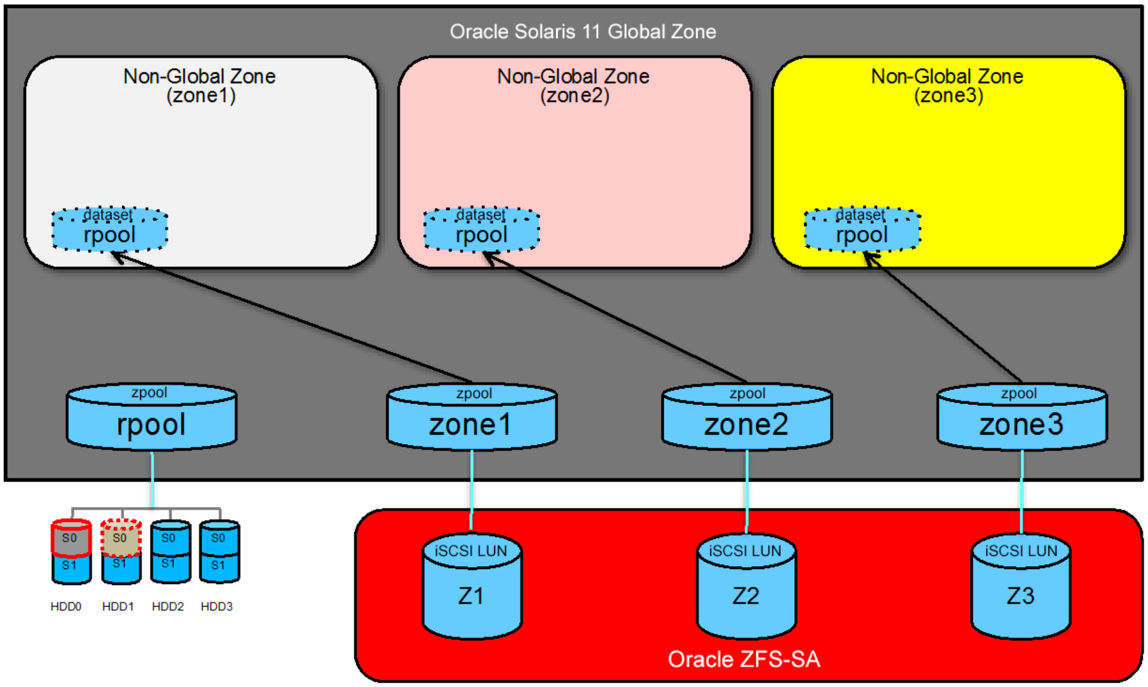


Figure 3. One LUN per zone

As stated previously, the recommended best practice is to create one iSCSI LUN per domain. However, if you plan to create only a small number of zones on the SuperCluster overall, then creating one LUN per zone is a viable alternative.

The procedures for the two scenarios are slightly different. However, both share the first three steps:

- Configure the iSCSI service on each Oracle Solaris 11 domain which is to hold app zones.
- Configure an iSCSI target and target group on Oracle ZFS-SA.
- Configure an iSCSI initiator and iSCSI initiator group on Oracle ZFS-SA for each domain.

The procedure to create one LUN per domain (recommended) is as follows:

- Create a LUN on Oracle ZFS-SA for each domain.
- Associate each LUN with the relevant iSCSI target group and iSCSI initiator group.
- Present the iSCSI LUN(s) to Solaris in the domain(s).
- Format the LUN in each domain.
- Create a zpool on the LUN in each domain.

The procedure to create one LUN per zone (alternative) is as follows:

- Create a LUN on Oracle ZFS-SA for each app zone.
- Associate each LUN with the relevant iSCSI target group and iSCSI initiator group.
- Copy the LUN’s globally unique identifier (GUID) and enter it into the zone configuration file.

The procedure for one LUN per zone utilizes the Oracle Solaris Zones on Shared Storage (ZOSS) feature. In this case, there is no need to manually present and configure the LUN in the domain. This will be done automatically by Oracle Solaris. This is discussed in more detail later in this document.

It is important to emphasize that iSCSI LUNs can only be presented to or used by one domain and cannot be shared across domains. Also, on SuperCluster, iSCSI LUNs are supported only for use by the zone rpool. For these reasons, NFS is utilized for all additional storage as well as for shared storage requirements.

Note that iSCSI LUNs are created before the zone is configured and built, while NFS shares are presented and mounted within the zone only after it is built, configured, and up and running (although they can be created on Oracle ZFS-SA at any time in advance).

Creating and Configuring NFS Shares

In order to present an NFS share to the zone, the InfiniBand network must be plumbed to and accessible by the zone.

The procedure to create, configure, and present an NFS share to one or more zones is as follows:

- Create the NFS share on Oracle ZFS-SA.
- For each zone in which you wish to present the NFS share, perform the following:
  - Edit the `/etc/dfstab` file to add the NFS share with the recommended mount options.
  - Mount the NFS share.

Figure 4 illustrates two NFS shares. One share is mounted to the first two zones while the second is mounted only in the third zone. In practice, NFS shares can be mounted in many zones spread across one or more domains.

SuperCluster: **Zone Storage (NFS)**

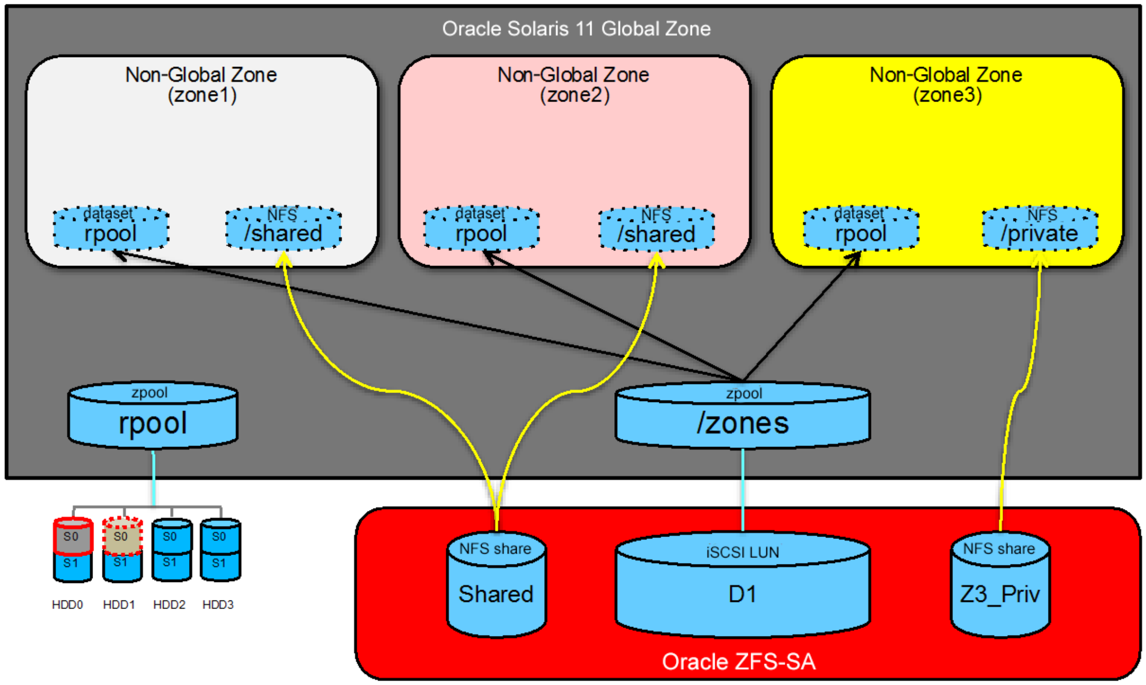


Figure 4. Zone storage for two NFS shares

In the case of using one iSCSI LUN per zone, the diagram is identical from an NFS perspective (see Figure 5).

SuperCluster: **Zone Storage (NFS)**

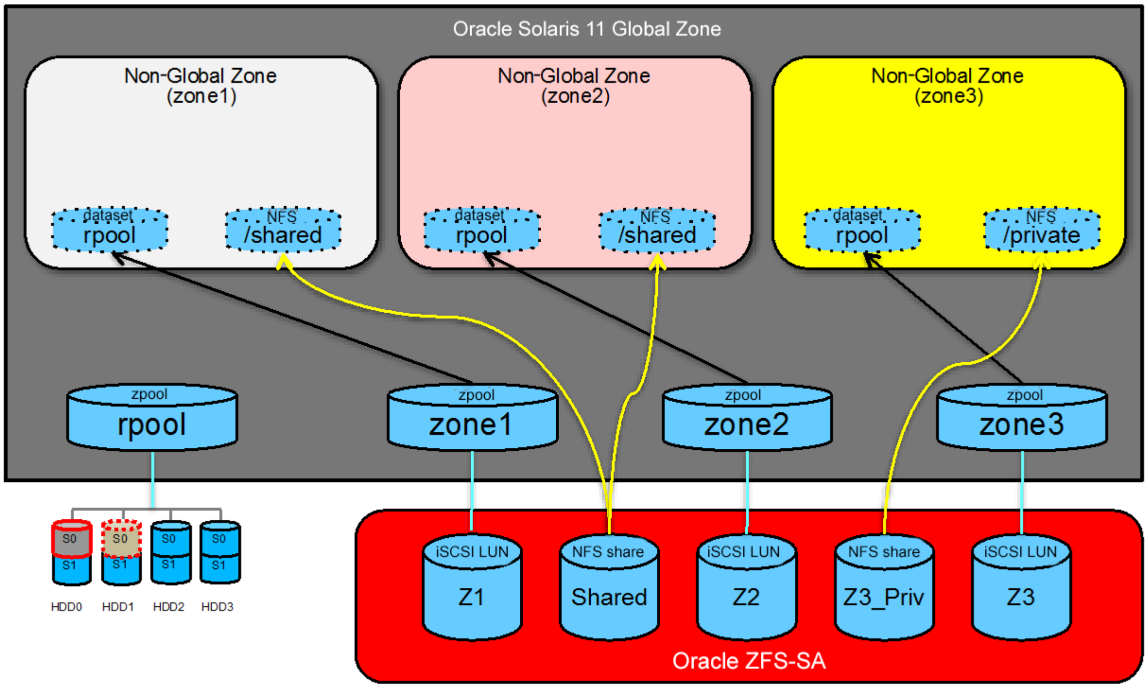


Figure 5. One iSCSI LUN per zone

Creating and Configuring Zones

Once the underlying shared storage is configured, the zone itself can be configured and built.

The following is a high-level summary of the process:

- Identify underlying network interfaces that are required in order to present the desired networks to the zone.
- Create a zone configuration file.
- Create a `sysconfig` template for installing the zone.
- Create a network configuration shell script.
- Configure the zone with `zonecfg`, using the zone configuration file.
- Install (or clone) the zone with `zoneadm`, using the `sysconfig` template.
- Boot the zone.
- Run the network configuration shell script within the zone.
- (Optional) Present and mount NFS shares.
- (Optional) Dedicate CPU resources.

At this point, the zone will be fully functional and ready for additional customization or installation of applications. This is a highly repeatable process that, once scripted, can be used to create new zones within a few minutes.

Additionally, once a zone is built and configured it can be cloned to create additional zones with a similar configuration. The cloning process does not preserve all of the configuration of the zone—such as IP addresses and network services configuration, which must still be configured after the zone is cloned—but it can be very useful for cloning zones that have other custom configuration such as customer-specific packages and applications installed.

Cloning is a process of replicating the same zone with the global zone of a domain. However, it is also possible to replicate the zone for cloning in other domains. This process will be discussed in more detail later.

Configuring Networking in App Zones

One of the areas in which creating app zones might differ somewhat from traditional (non-engineered) Oracle Solaris systems is how the network interfaces are configured. In SuperCluster, all network interfaces are redundant[2]. For the InfiniBand and management network, this redundancy is always implemented via IPMP. The 10 GbE client access network (and the optional backup network) is also usually implemented via IPMP but can be changed to LACP.

These network constructs must be taken into account when plumbing the networking into the zone. For IPMP, you must create virtual network interfaces (VNICs) on the underlying interfaces and plumb those into the zone. The IPMP group is then re-created within the zone. However, for LACP you merely create a VNIC directly on top of the LACP aggregation and plumb that into the zone with no need to re-create the aggregation within the zone.

These two concepts and their logical network plumbing are illustrated in Figure 6 and Figure 7. Note that these examples also include an optional 10 GbE backup network.

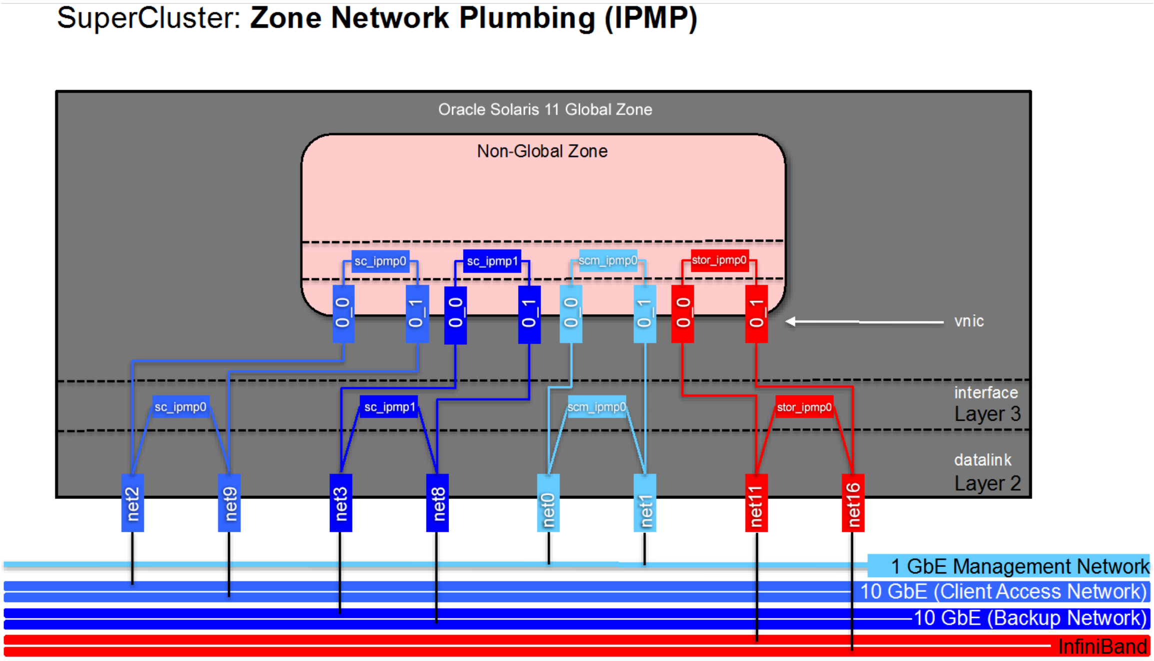


Figure 6. Zone network plumbing (ICMP)

[2] The management network and InfiniBand networks are always implemented with redundancy. However, the 10 GbE "client access network" does not need to be implemented with redundancy if you choose not to. Best practice, however, is to always implement redundancy.

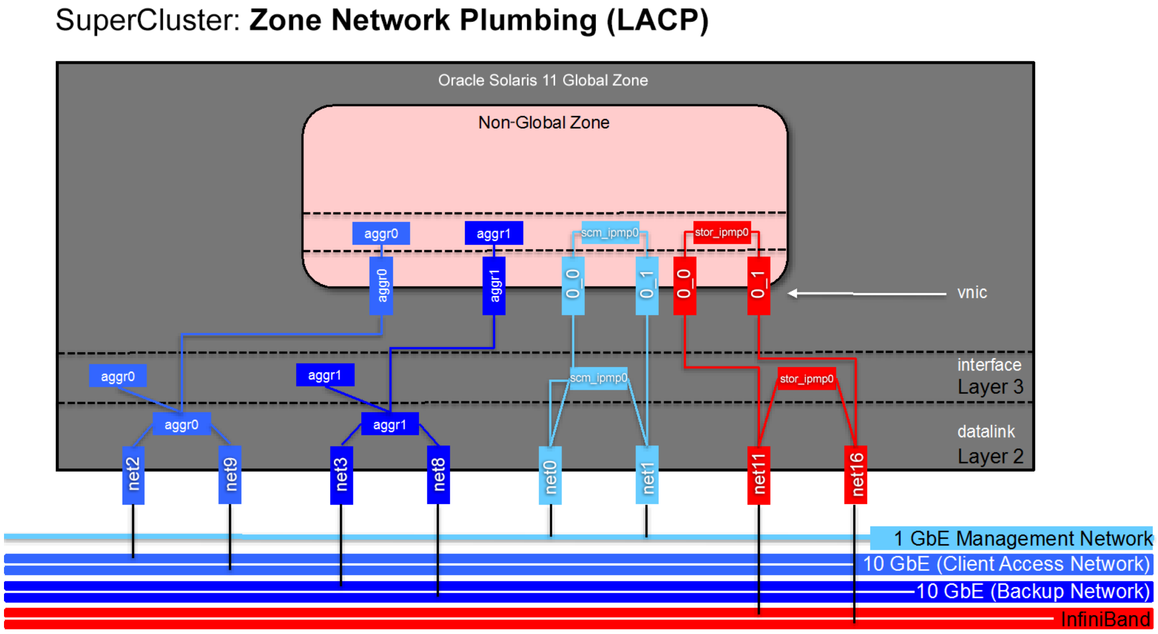


Figure 7. Zone network plumbing (LACP)

VLAN tagging can also be implemented such that each zone can (if desired) be on separate VLANs. This is discussed in more detail later in the paper.

It is also important to emphasize that in Oracle Solaris 11, each zone has a full network stack completely independent of other zones or the global zone it is hosted in. A zone can be on completely different networks (physical or logical), and it can have its own routing table, name services configuration, and security model.

In the case of using VLAN tags on the 10 GbE network, it is as simple as specifying the VLAN tag when creating the VNIC(s) for the zone. Figure 8 and Figure 9 illustrate this for both IPMP and LACP.

SuperCluster: **Zone Network Plumbing (IPMP with VLANS)**

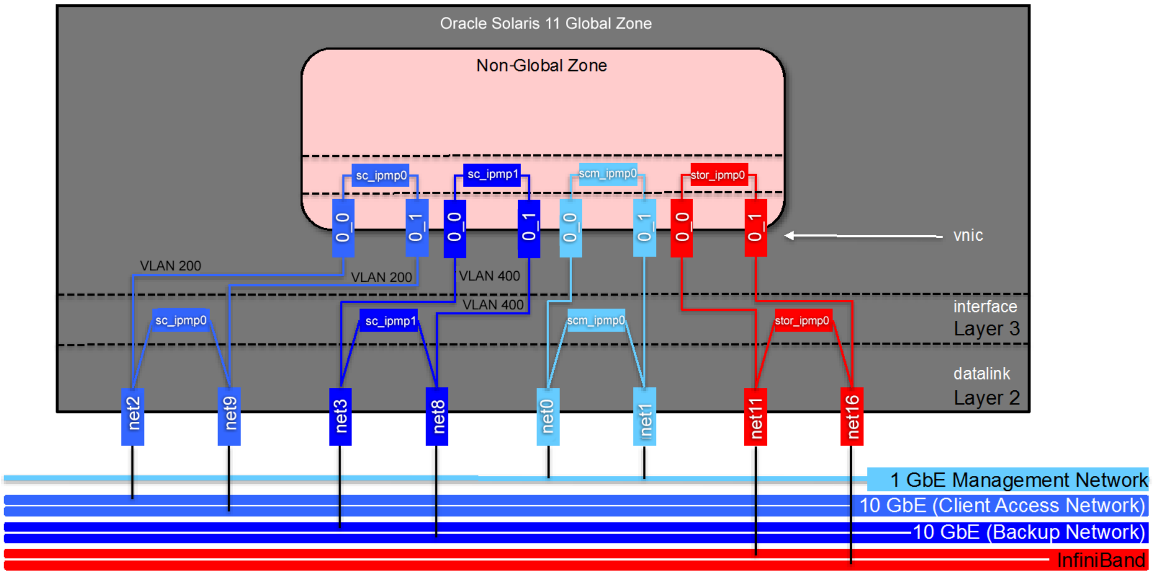


Figure 8. Zone network plumbing (IPMP with VLANs)

SuperCluster: **Zone Network Plumbing (LACP with VLANS)**

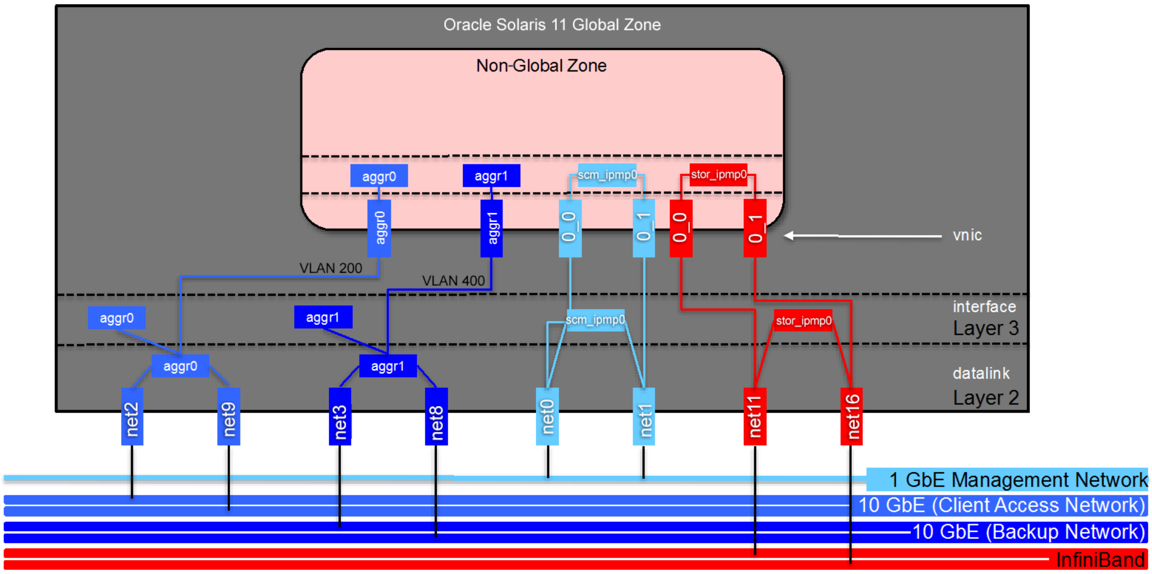


Figure 9. Zone network plumbing (LACP with VLANs)

Installation Guide

Planning for Zones

Before you begin, you should be familiar with the infrastructure of your SuperCluster and the configuration of all the relevant components. An excellent document to begin with is the *SuperCluster Deployment Summary Report* delivered by Oracle Advanced Customer Support upon the completion of the initial installation of your SuperCluster. This document should contain all the configuration details of your SuperCluster, in particular, IP addresses and passwords at the time when the Oracle Advanced Customer Support team completed the installation.

Another important document to read and be familiar with is the *Oracle SuperCluster Owner's Guide* (Oracle SuperCluster T5-8 or M6-32 version). Ask your Oracle representative for a copy of this document if you do not have it (it is not available externally).

Understand what level of patches is currently installed on your SuperCluster. Ensure that you have applied a currently supported version of the Quarterly Full Stack Download Patch (QFSDP) for SuperCluster. See My Oracle Support (MOS) Note 1567979.1 *SuperCluster T5-8 Supported Software Versions.*

Review and have handy for reference the Oracle documentation for creating and managing Oracle Solaris Zones. Ensure that you reference the documentation for your currently installed version of Oracle Solaris.

Determine which networks you will present to your zones. Remember to include the InfiniBand network if your zone needs to communicate within the SuperCluster[3]. Assign IP addresses for each network, and note the netmasks. Ensure that these addresses are in your domain name service (DNS)[4]. Determine what will be your default router. Ensure that your DNS server is reachable via the chosen networks.

Determine how large to make the iSCSI LUN for your zone's rpool. For a minimal installation, it can be as small as 20 GB[5]. However, it might be more practical to choose a larger size, such as 80 GB or even 150 GB. Note that Oracle ZFS-SA has approximately 32 TB of available capacity when initially configured. Decide if any application binaries or data will live within the rpool or will be on separate shares via NFS.

Creating and Configuring iSCSI LUNs

Overview

In Oracle Solaris 11, each zone must have its own zone path on ZFS. This path resides on a ZFS data set within a ZFS pool in the global zone. Multiple zones can share a single ZFS pool by having their own data sets within it, or each zone can have its own ZFS pool.

This ZFS pool can either reside on disks internal to the Oracle SuperCluster T5-8 compute node, or on the Oracle ZFS-SA via iSCSI LUNs. Best practice is to utilize iSCSI LUNs. Each iSCSI LUN has exactly one ZFS pool on top of it. So, in practice, this translates into either multiple zones within a domain sharing one iSCSI LUN or each zone having its own iSCSI LUN.

[3] It is not required to present the InfiniBand network within the zone. The zone's rpool may still reside on an iSCSI LUN via Oracle ZFS-SA. This LUN will be presented via the global zone's InfiniBand network.

[4] The IP addresses used on the InfiniBand network are not typically placed in DNS.

[5] It is possible to dynamically grow the size of this LUN later on.

Regardless of which method is chosen, files created within the zone's path are visible only within that zone and to the root user in the global zone.

Best practice on SuperCluster is to have one iSCSI LUN per domain and have all the zones share this one LUN. However, individual iSCSI LUNs for each zone are an acceptable alternative when there are a small number of zones and there is a desire to replicate or migrate individual zones (via their LUNs).

There are two initial one-time steps required to configure the global zone and the Oracle ZFS-SA:

1. Configure the iSCSI initiator service for Oracle Solaris in the global zone.

2. Configure the iSCSI target and target group on the Oracle ZFS-SA.

Step #1 will need to be performed once on each domain where you wish to present iSCSI LUNs. On a typical Oracle SuperCluster T5-8 with one app domain per SPARC T5-8 node, this step would need to be performed once on each app domain of each node (so a total of two times).

Step #2 needs to be performed only once on the Oracle ZFS-SA.

Once these initial two steps have been performed, the following steps can be repeated for each domain where you wish you present the iSCSI LUN(s):

3. Create and configure an iSCSI LUN on the Oracle ZFS-SA.

4. Present the iSCSI LUN to Oracle Solaris in the global zone.

5. From Oracle Solaris, label the iSCSI LUN as an Oracle Solaris partition.

6. Create a ZFS pool zpool on the LUN.

If you plan to use one iSCSI LUN per domain, you will need to perform all four steps (#3 to #6). If, however, you plan to use one iSCSI LUN per zone, you need to perform only step #3 (to create the LUN). If there is a 1:1 relationship between the zone and iSCSI LUN, then a feature of Oracle Solaris 11 known as Zones On Shared Storage (ZOSS) is utilized whereby the iSCSI LUN's GUID is specified directly within the zone configuration and Oracle Solaris will perform steps #4, #5 and #6 automatically, as needed[6].

Finally, zones may then be created which will reside on the LUN(s).

Configure iSCSI Initiator Service in the Global Zone

Repeat these steps on every application domain where you wish to present iSCSI LUNs.

1. Check whether the iSCSI service is enabled:

```
# svcs iscsi/initiator
STATE          STIME    FMRI
disabled       16:01:12 svc:/network/iscsi/initiator:default
```

2. If necessary, enable the service:

```
# svcadm enable iscsi/initiator
```

[6] Whenever the zone is installed, cloned, or attached.

3. Verify:

```
# svcs iscsi/initiator
STATE          STIME    FMRI
online         16:02:40 svc:/network/iscsi/initiator:default
```

4. Obtain the host IQN. You will need this IQN later when configuring the iSCSI initiator on the Oracle ZFS-SA:

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:e00000000000.506e8d5f
Initiator node alias: solaris
      [ … ]
```

5. Configure the iSCSI initiator. You will need to know the IP address of the Oracle ZFS-SA on the InfiniBand network. The SuperCluster default is 192.168.28.1. However, you can verify your specific IP address by looking for the host name containing “storIB” in your hosts file.

```
# iscsiadm add discovery-address 192.168.28.1
# iscsiadm modify discovery -t enable
```

Configure iSCSI Target and Target Group on the Oracle ZFS-SA

The following steps must be performed from a web browser connected to the active controller of the Oracle ZFS-SA. If you can run the web browser from the SuperCluster and remote display back, then you can simply connect to the IP address of the Oracle ZFS-SA on InfiniBand, which will automatically connect to the active controller.

Otherwise, you must connect via the management network using the specific IP address for the active controller.

1. Connect to the management browser-based user interface (BUI) of the Oracle ZFS-SA via HTTPS to the IP address on port 215.

For example: https://192.168.28.1:215/

The login screen will look like this:

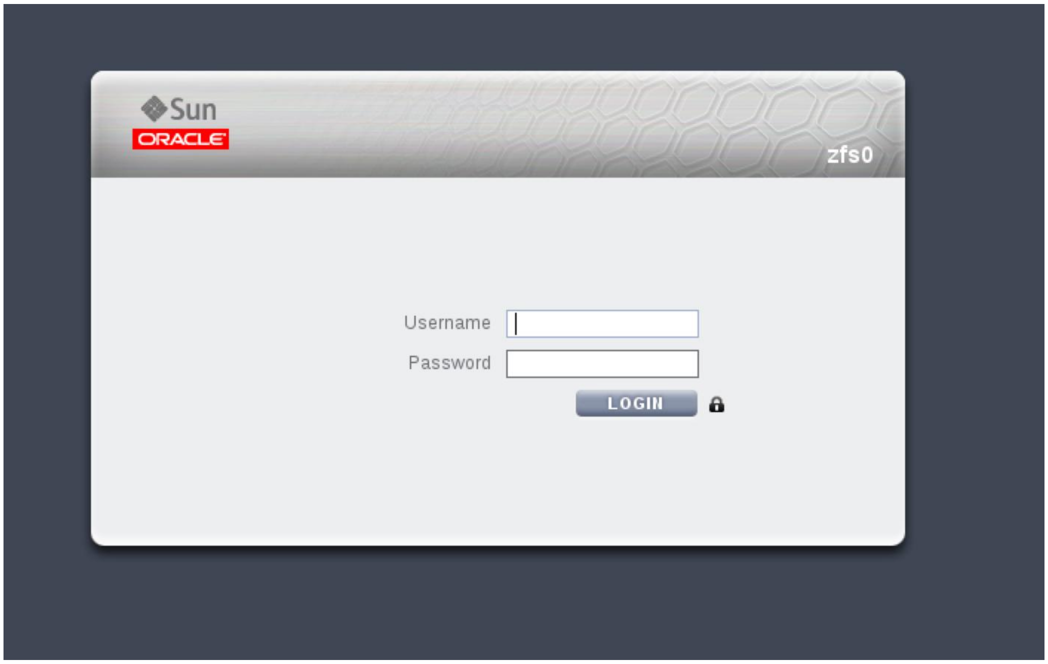
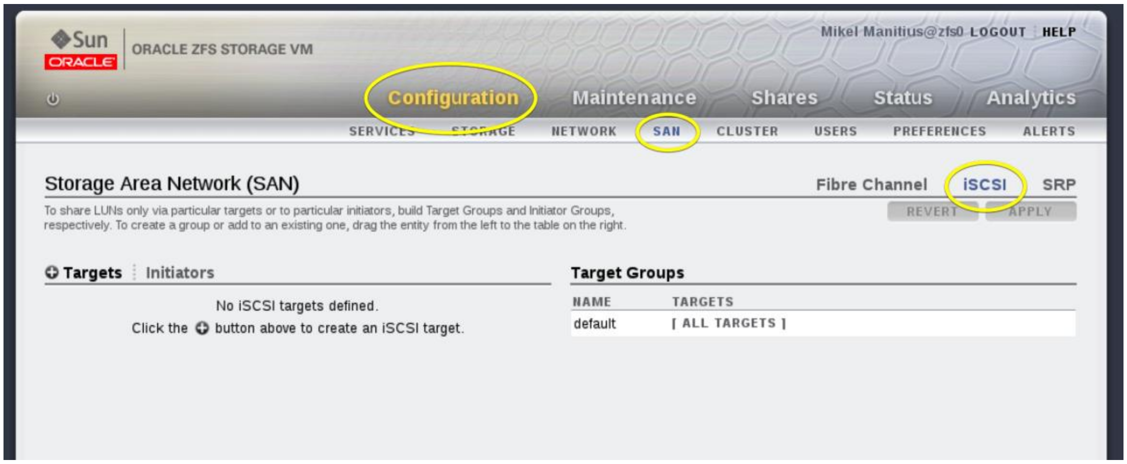


Figure 10. Login screen

Log in as `root` with the password (initially provided in the *SuperCluster Deployment Summary Report*).

2. Navigate to the iSCSI section within the SAN tab under Configuration; see the circled selections in the screenshot below.



Storage Area Network (SAN)

To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

No iSCSI targets defined.

Click the ⊕ button above to create an iSCSI target.

Target Groups

NAME	TARGETS
default	[ALL TARGETS]

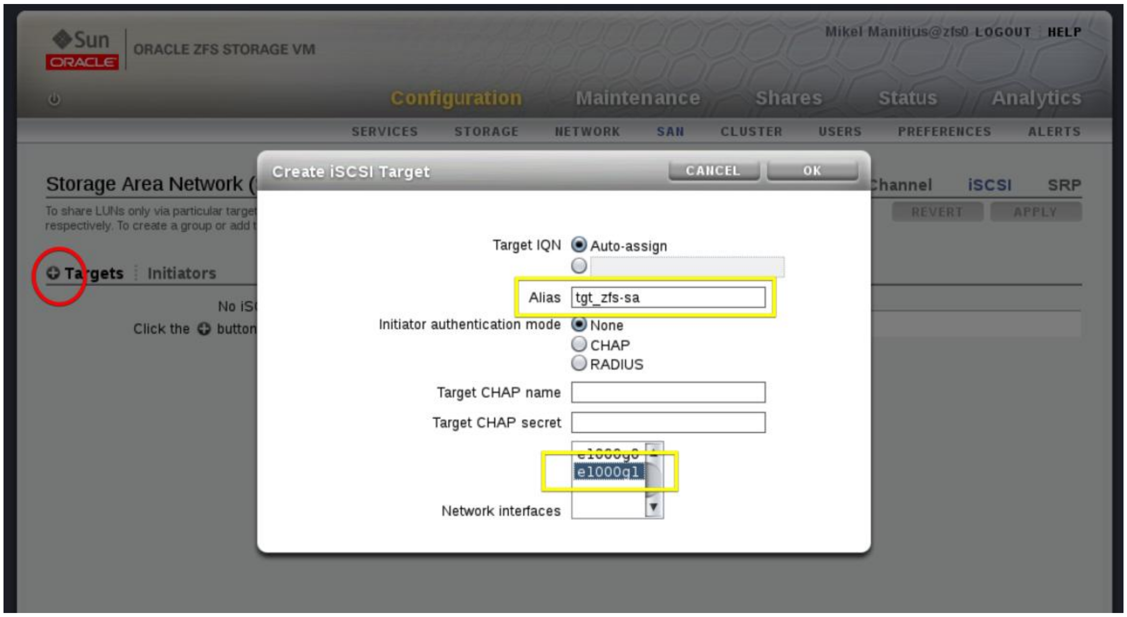
Figure 11. iSCSI section within the SAN tab

3. Click the **+** sign (circled in red in the screenshot below) to the left of the word “Targets.”

A dialog box will appear to configure and create the iSCSI target.

Give it the name `tgt_zfs-sa`.

Make sure to select the network interface that corresponds to the IPMP group for InfiniBand.



Target IQN ● Auto-assign ○

Alias tgt_zfs-sa

Initiator authentication mode ● None ○ CHAP ○ RADIUS

Target CHAP name

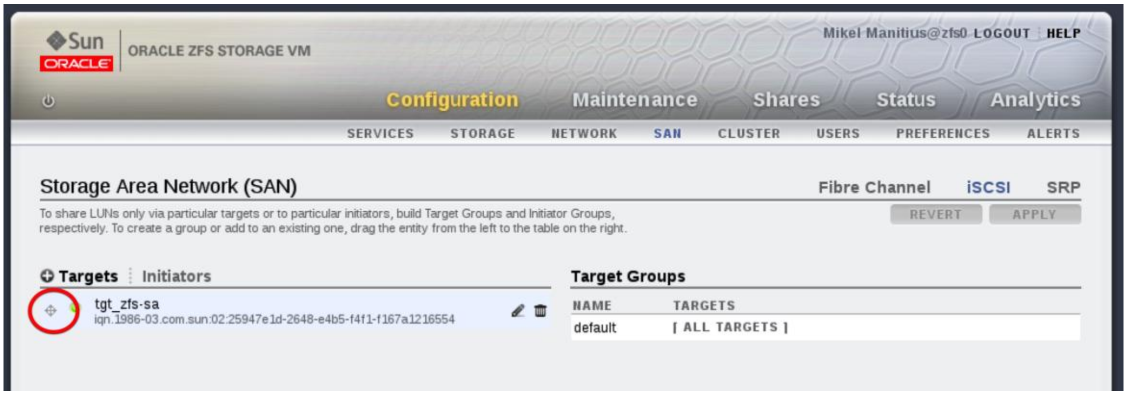
Target CHAP secret

Network interfaces e1000g0 e1000g1

Figure 12. Create iSCSI Target dialog box

Click **OK** and a new iSCSI target is created.

4. Create a new iSCSI target group by dragging the newly created iSCSI target entry on the left across to the group section on the right. When you hover your mouse cursor over the iSCSI target, a set of grab arrows will appear on the very left.



Storage Area Network (SAN)

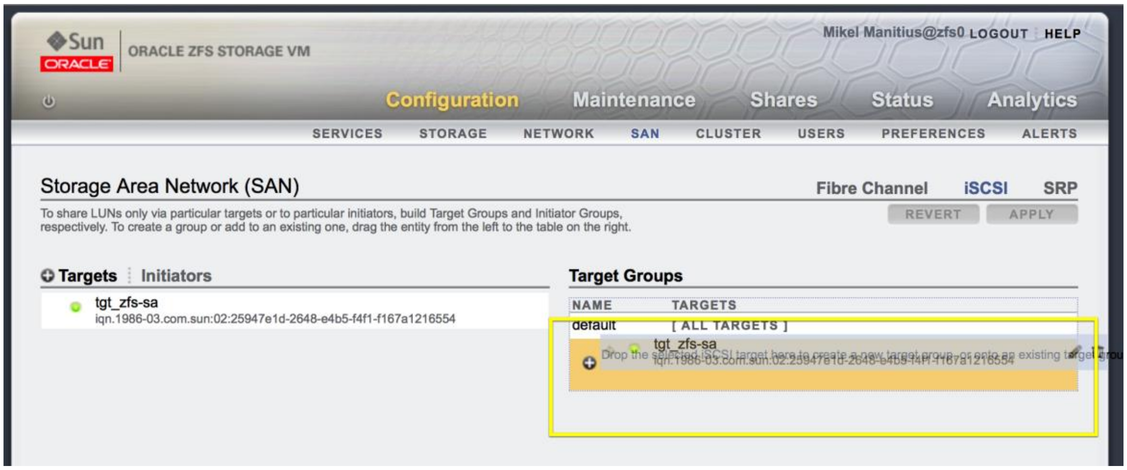
To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

tgt_zfs-sa
iqn.1986-03.com.sun:02:25947e1d-2648-e4b5-f4f1-f167a1216554

Target Groups

NAME	TARGETS
default	[ALL TARGETS]

Figure 13. Screenshot showing the grab arrows circled



Storage Area Network (SAN)

To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

Targets | Initiators

tgt_zfs-sa
iqn.1986-03.com.sun:02:25947e1d-2648-e4b5-f4f1-f167a1216554

Target Groups

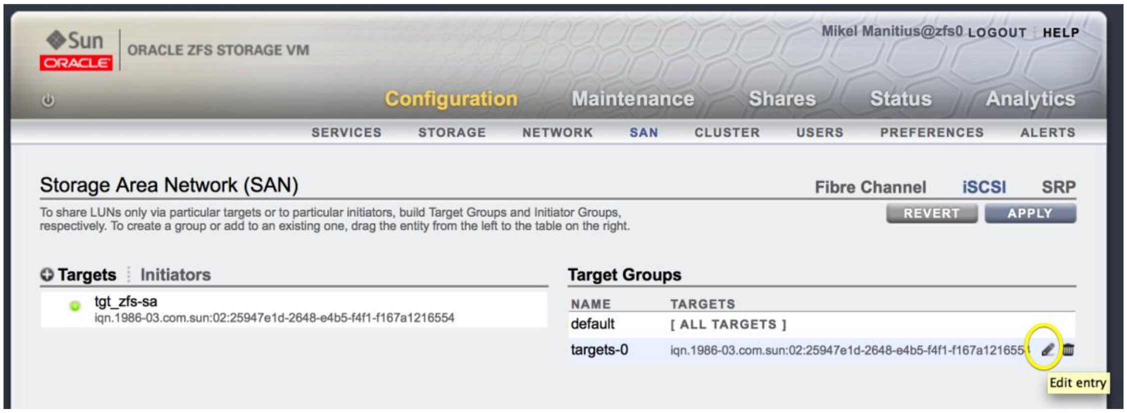
NAME	TARGETS
default	[ALL TARGETS]

tgt_zfs-sa
iqn.1986-03.com.sun:02:25947e1d-2648-e4b5-f4f1-f167a1216554

Drop the selected iSCSI target here to create a new target group, or onto an existing target group

Figure 14. Screenshot showing the target dragged over to the drop area under Target Groups

5. Rename the newly created target group. To open the dialog box in which you can rename the target group, you must hover your mouse cursor over the target group entry until a pencil icon appears and then click it.



Storage Area Network (SAN)

To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

Targets | Initiators

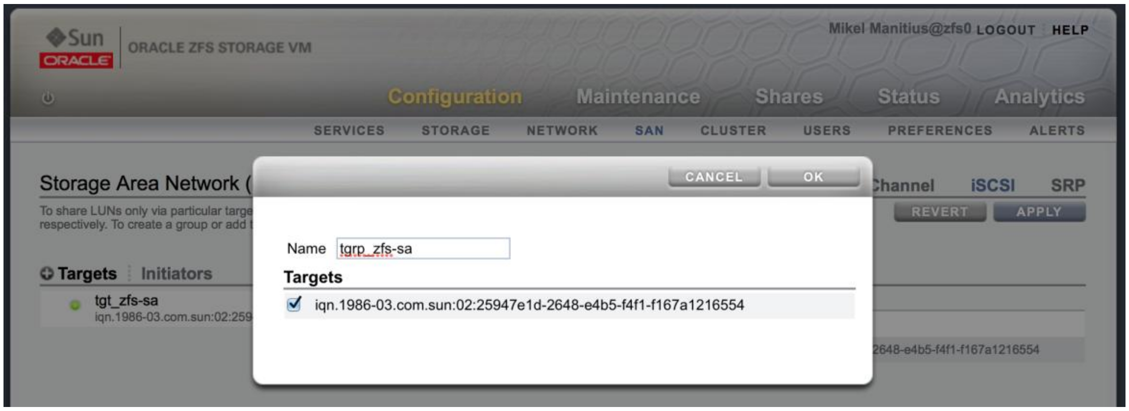
tgt_zfs-sa
iqn.1986-03.com.sun:02:25947e1d-2648-e4b5-f4f1-f167a1216554

Target Groups

NAME	TARGETS
default	[ALL TARGETS]
targets-0	iqn.1986-03.com.sun:02:25947e1d-2648-e4b5-f4f1-f167a1216554

Figure 15. Screenshot showing the pencil icon

Change the name to `tgrp_zfs-sa` and click **OK**.



Name: tgrp_zfs-sa

Targets

☑ iqn.1986-03.com.sun:02:25947e1d-2648-e4b5-f4f1-f167a1216554

Figure 16. Dialog box for renaming the target group

6. Click the **APPLY** button in the upper right. The iSCSI target and target groups are now created.

This concludes the initial configuration of the Oracle ZFS-SA.

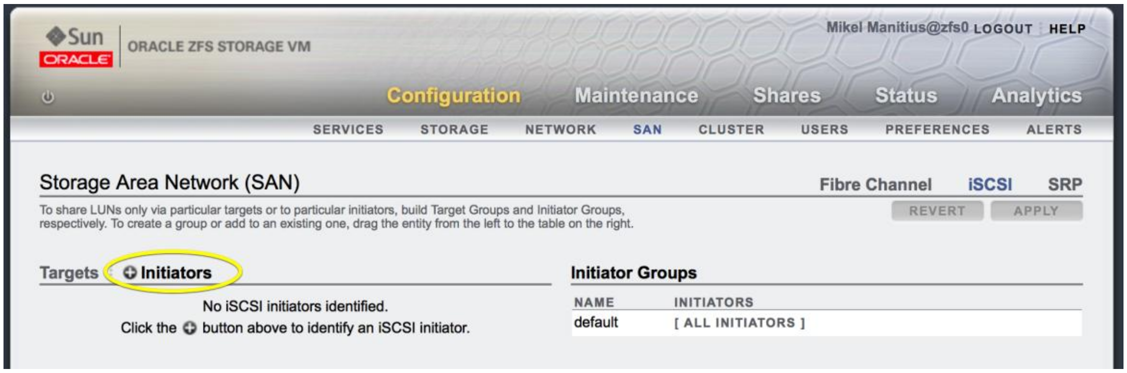
Configure the iSCSI Initiator on the Oracle ZFS-SA

This section explains how to configure an iSCSI initiator on the Oracle ZFS-SA. Each global zone (domain) to which you wish to present iSCSI LUN(s) must have an iSCSI initiator configured on the Oracle ZFS-SA, because the iSCSI LUN(s) created later will be directly associated with the iSCSI initiator of the domain to which they will be presented.

Repeat this procedure for each domain to which you wish to present iSCSI LUNs.

1. Create the iSCSI initiator. You will require the IQN of the global Zone (domain) obtained previously when configuring the iSCSI initiator in the domain.

Click the Initiators link next to Targets and then click the + sign to create a new initiator.



To share LUNs only via particular targets or to particular initiators, build Target Groups and Initiator Groups, respectively. To create a group or add to an existing one, drag the entity from the left to the table on the right.

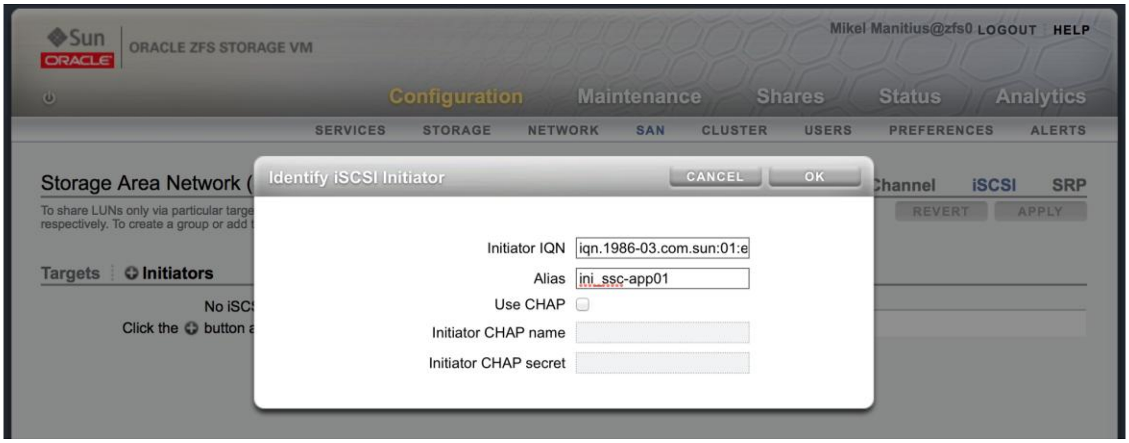
No iSCSI initiators identified.

Click the ⊕ button above to identify an iSCSI initiator.

NAME	INITIATORS
default	[ALL INITIATORS]

Figure 17. Screenshot showing the Initiators link

2. Enter (or paste) the IQN of the domain, and for the alias use the domain's host name prefixed with `ini_` (for example, `ini_ssc-app01`).



Initiator IQN: iqn.1986-03.com.sun:01:e

Alias: ini_ssc-app01

Use CHAP: □

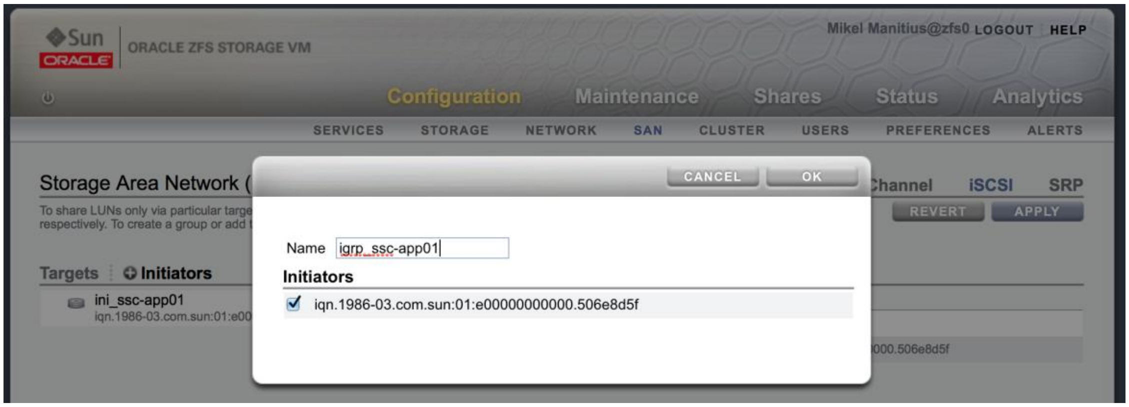
Initiator CHAP name:

Initiator CHAP secret:

Figure 18. Identify iSCSI Initiator dialog box

Click **OK**.

3. Drag and drop the newly created initiator to the Initiator Groups area (same process as was used to create a target group earlier) and then edit the initiator group to rename it just as you did with the initiator, but now using the prefix `igrp_` with the host name (for example, `igrp_ssc-app01`).



Name igrp_ssc-app01

Initiators

☑ iqn.1986-03.com.sun:01:e00000000000.506e8d5f

Figure 19. Dialog box for renaming the initiator group

Click **OK**. And then **APPLY**.

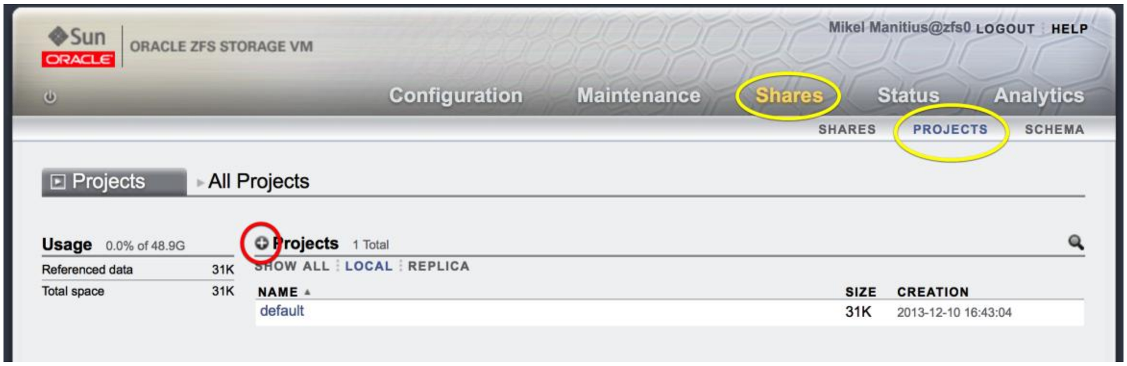
The initiator and initiator group for this domain are now configured. Repeat this process for any other domains to which you wish to create iSCSI LUNs.

Create and Configure iSCSI LUN on the Oracle ZFS-SA

This section explains how to create an iSCSI LUN on the Oracle ZFS-SA. Repeat this procedure for each LUN to be created. However, note that each iSCSI LUN created is specifically assigned to the iSCSI initiator of the host (domain) where it will be presented. When creating multiple iSCSI LUNs, care must be taken to assign each LUN to the appropriate initiator.

1. Create a Project. A Project is a template of settings and logical groupings that can be used for actions such as replication.

At the top, select the **Shares** tab followed by the **Projects** subtab, and then click the **+** sign next to Projects (circled in red).



Usage 0.0% of 48.9G

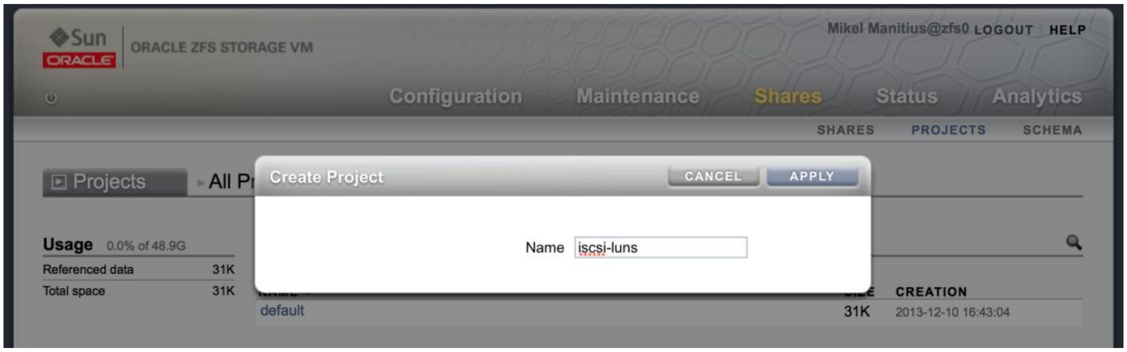
Referenced data	31K
Total space	31K

NAME	SIZE	CREATION
default	31K	2013-12-10 16:43:04

Figure 20. Preparing to create a project

You should decide how you want to group your LUNs into projects. You can just call your project `iscsi-luns` and associate all LUNs with it. Or you can create one project per domain. This decision can play a role if you later plan to use the replication feature of the Oracle ZFS-SA, because replication can be specified on a per-LUN or per-project basis.

In this example, we call the new project `iscsi-luns`. Click **OK**.



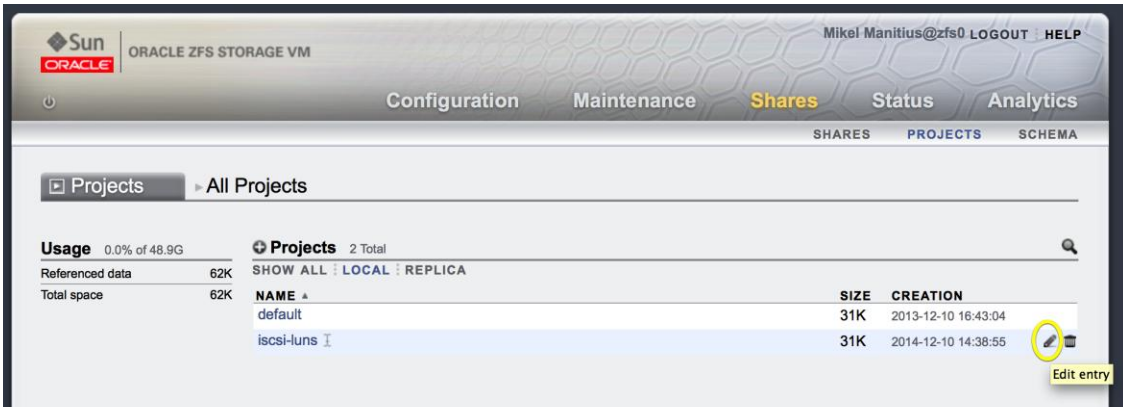
Name iscsi-luns

Referenced data	31K
Total space	31K

NAME	SIZE	CREATION
default	31K	2013-12-10 16:43:04

Figure 21. Create Project dialog box

2. Hover your mouse cursor over the newly created project to expose the pencil icon on the right side, and click it to enter the project.

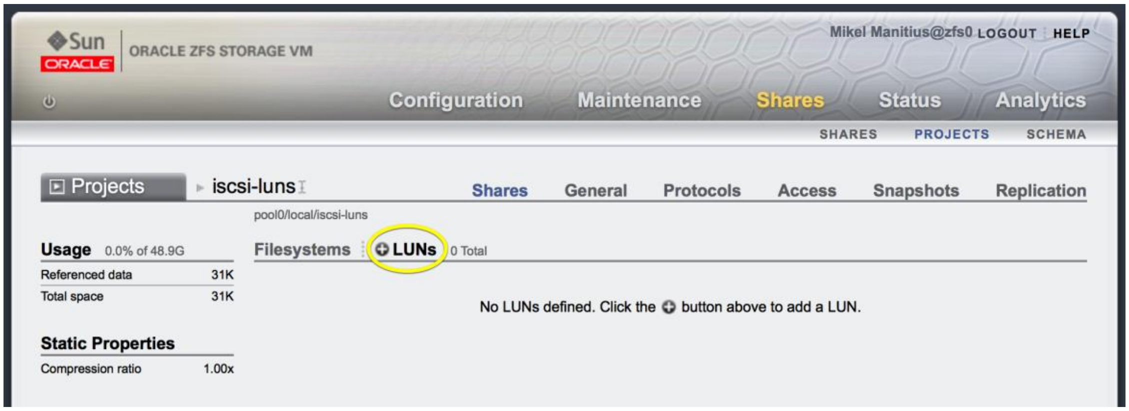


Referenced data	62K
Total space	62K

NAME ▴	SIZE	CREATION
default	31K	2013-12-10 16:43:04
iscsi-luns	31K	2014-12-10 14:38:55

Figure 22. Screenshot showing the pencil icon

3. Click the **+** sign next to “LUNs” to begin creating a new LUN.



pool0/local/iscsi-luns

Referenced data	31K
Total space	31K
Compression ratio	1.00x

No LUNs defined. Click the ⊕ button above to add a LUN.

Figure 23. Screenshot showing the + icon

4. Give the LUN a name. A good naming convention is the host name of the domain to which the LUN will be assigned followed by a reference to the zone for which it is for, for example, `ssc-app01-z1`.

Give the volume (LUN) a size. This example shows 80 G. If this LUN is to hold multiple zones, you might wish to choose a larger size. The LUN and associated ZFS pool can also be dynamically expanded later.

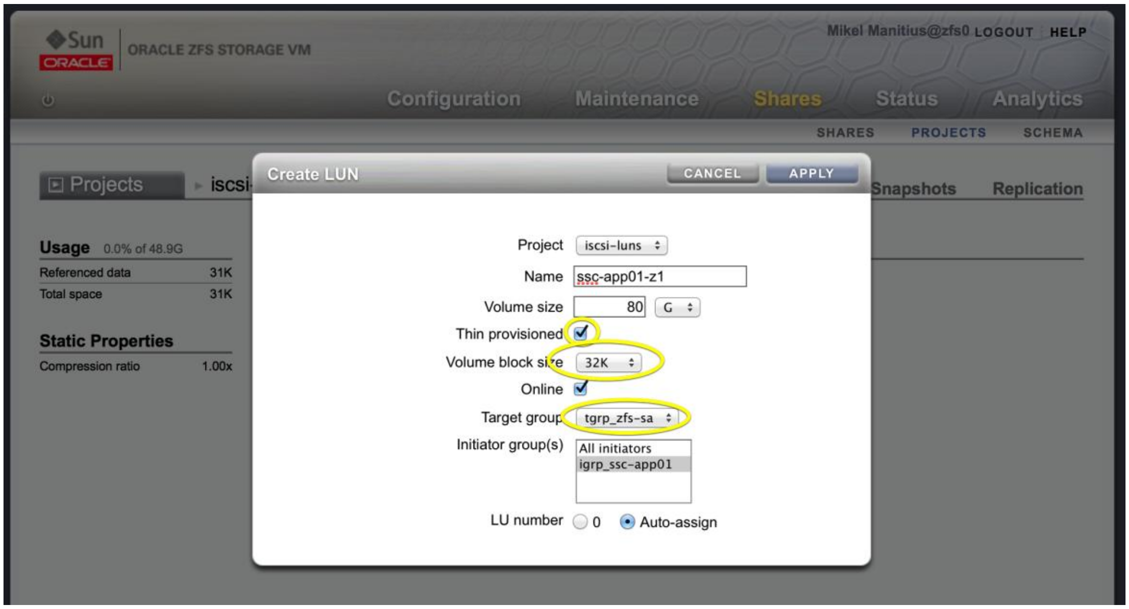
If you select the **Thin provisioned** option, then space will not be reserved in the Oracle ZFS-SA[7].

Change the block size to 32K.

Note: You can set the above as defaults within your project and when you select that project during LUN creation, these will be the defaults.

Select the previously created target group.

Select the initiator group of the target domain to which this iSCSI LUN is to be presented.



Project: iscsi-luns

Name: ssc-app01-z1

Volume size: 80 G

Thin provisioned: ☑

Volume block size: 32K

Online: ☑

Target group: tgrp_zfs-sa

Initiator group(s): All initiators
igrp_ssc-app01

LU number: ○ 0 ● Auto-assign

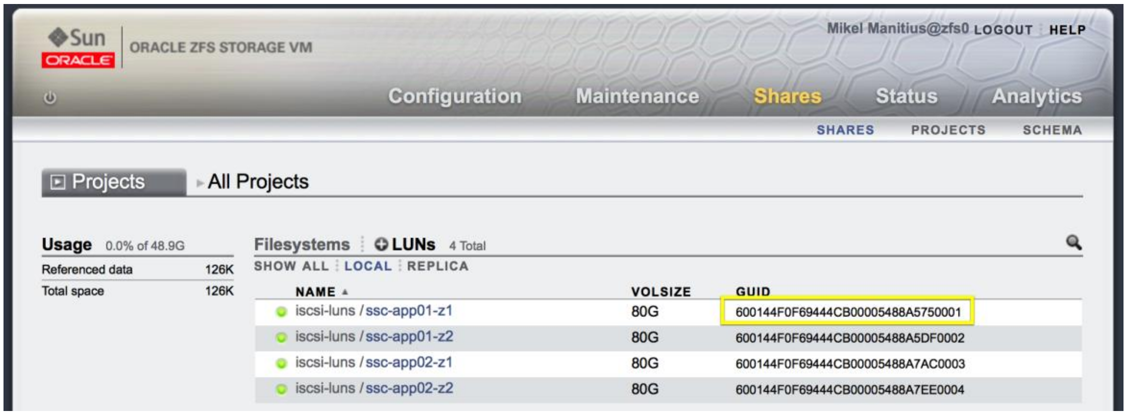
Figure 24. Create LUN dialog box

Click **APPLY**.

5. Repeat this process to create as many LUNs as you need. Take care to associate them with the correct initiator group.

Once you have created all of the LUNs for all of the zones on all of the domains that you need, your screen might look like the following screenshot.

[7] One drawback of thin-provisioned LUNs is that if you run out of space on your Oracle ZFS-SA, all thin-provisioned LUNs will fail to write new blocks; thus, potentially causing a severe disruption to all their zones. Use of this option should be carefully considered.



Usage 0.0% of 48.9G

Referenced data	126K
Total space	126K

NAME	VOLSIZE	GUID
iscsi-luns / ssc-app01-z1	80G	600144F0F69444CB00005488A5750001
iscsi-luns / ssc-app01-z2	80G	600144F0F69444CB00005488A5DF0002
iscsi-luns / ssc-app02-z1	80G	600144F0F69444CB00005488A7AC0003
iscsi-luns / ssc-app02-z2	80G	600144F0F69444CB00005488A7EE0004

Figure 25. Screenshot showing the created LUNs

Note that each LUN has an associated GUID. Either write these down or remember how to get back to this screen, because you will later use these GUIDs to identify LUNs presented to the host (domain) with their names on the Oracle ZFS-SA.

If you plan to use one zone per LUN, you can skip presenting the LUN(s) in the next section, because this will be done automatically by Oracle Solaris when using the ZOSS configuration syntax. If you are using ZOSS, then skip to “Configuring (Defining) App Zones” section.

Present the iSCSI LUN to Oracle Solaris in the Global Zone (Domain)

This section assumes that you will create one iSCSI LUN per domain for all zones to share.

This procedure will create device entries in Oracle Solaris for the iSCSI LUN(s) presented to it and will format them to place an Oracle Solaris label on each LUN.

1. As `root` in the global zone, run the following command. This will cause the system to find iSCSI devices presented to it and create device entries for them.

```
# devfsadm –c iscsi
```

This example shows the output for the `format` command before and after running the command above.

Note the strings highlighted in bold. These should match the GUIDs on the Oracle ZFS-SA (shown in the earlier iSCSI LUN creation step) and can be used to identify which LUNs these are.

```
# format
AVAILABLE DISK SELECTIONS:
       0. c7d0 <VBOX HAR-418909f6-abdac37-0001-62.50GB>
          /pci@0,0/pci-ide@1,1/ide@0/cmdk@0,0
       1. c8d0 <VBOX HAR-0edd1465-be99119-0001-100.00GB>
          /pci@0,0/pci-ide@1,1/ide@1/cmdk@0,0
Specify disk (enter its number): ^C
# devfsadm -c iscsi
# format

c0t600144F0F69444CB00005488A5750001d0: configured with capacity of
79.92GB

AVAILABLE DISK SELECTIONS:
       0. c0t600144F0F69444CB00005488A5750001d0 <SUN-ZFS Storage 7320-
1.0 cyl 2598 alt 2 hd 254 sec 254>
          /scsi_vhci/ssd@600144f0f69444cb00005488a5750001
       1. c1d0 <SUN-DiskSlice-418GB cyl 18232 alt 2 hd 255 sec 189>
          /virtual-devices@100/channel-devices@200/disk@0
       2. c1d1 <SUN-DiskSlice-418GB cyl 18232 alt 2 hd 255 sec 189>
          /virtual-devices@100/channel-devices@200/disk@1
Specify disk (enter its number): 0
selecting c0t600144F0F69444CB00005488A5750001d0
[disk formatted]
Disk not labeled.  Label it now? Y

format>
```

2. The first time that you run `format` after presenting the new LUN(s), you will be prompted to label the new LUN(s). See the example above.

The iSCSI LUN has now been presented to Oracle Solaris. The next step is to create a ZFS pool “zpool” on the LUN.

Create a ZFS Pool “zpool” on the iSCSI LUN

Note: ZFS prefers to manage the entire disk rather than individual slices. In this example, we provide the address of the entire disk without specifying any slice.

1. Identify the desired device from the output of the `format` command in the previous step.

Create the zpool on the device as follows:

```
# zpool create -m /zones zones c0t600144F0F69444CB00005488A5750001d0
```

This will cause a pool named `zones` to be created on the new iSCSI LUN. The root mount point for all data sets created within the pool will be /`zones`.

This assumes that all zones within the domain will share this ZFS pool (so individual zone paths will be `/zones/zone1, /zones/zone2, /zones/zone3`, and so on).

2. Repeat this process for every LUN on each host (domain).

At this point, your iSCSI LUNs are ready and you can proceed with zone creation.

Configuring (Defining) App Zones

Overview

Conceptually, configuring an Oracle Solaris Zone means providing the information necessary to establish what virtual resources will exist within the zone and what underlying resources within the global zone they will be based on. This is essentially just metadata.

It might be easier to think of this process as “defining” the zone, because the actual configuration of IP addresses and network services within the zone will occur later once the zone has been installed and booted. However, this paper will continue to use the term “configure” here, because that is the term used in the Oracle Solaris documentation.

The following types of information will be specified:

- The name of the zone[8].
- The location (path) of the zone’s root ZFS pool (rpool).
- Whether the zone should boot automatically.
- The type of IP interfaces used within the zone (these will always be of type *exclusive*).
- A list of network interfaces to plumb into the zone (discussed in more detail later). Note that strictly speaking, a zone is not required to have any network interfaces. However, in practice it usually has at least one. Any of the following networks can be presented to the zone:
    - InfiniBand (ZFS partition)
    - Management network
    - 10 GbE client access network
    - 10 GbE backup network
    - Other 10 GbE network
- (Optional) Name of the processor set assigned to this zone for CPU resource management.

The process of configuring a zone is performed by providing input to the `zonecfg` command. While this can be performed interactively from the command line, it is generally easier and more manageable to simply create a zone configuration file that contains all of these commands, and then specify its name as a parameter to the `zonecfg` command. This file can then be easily edited and cloned and edited again as needed for additional zones.

[8] This can be and often is different from the Oracle Solaris host name within the zone.

Basic Zone Configuration

The first portion of the zone configuration file identifies where the zone root will reside.

If you are using one iSCSI LUN per domain, then this is all you need at this point. The ZFS pool has already been created and mounted as `/zones`.

```
create -b
set zonepath=/zones/zone1
set autoboot=true
set ip-type=exclusive
```

If, however, you will be using one iSCSI LUN per zone, then you must also add the following additional syntax to associate the zone with the iSCSI LUN directly:

```
add rootzpool
add storage
iscsi://192.168.28.1:3620/luname.naa.600144F0F69444CB00005488A5750001
end
```

Note the two items in bold: the IP address of the Oracle ZFS-SA on the InfiniBand network and the GUID of the previously created iSCSI LUN.

When you use this notation, you are specifying within the zone configuration that this zone has its own iSCSI LUN. In this case, whenever you install, clone, or attach the zone, Oracle Solaris will automatically do the following, as necessary:

- Start the iSCSI initiator service in Oracle Solaris.
- Configure the discovery address and enable static LUN discovery.
- Present the LUN as a device.
- Create a ZFS pool on top of the LUN (or import it, in the case of attaching a LUN).

Likewise, when the zone is detached or uninstalled, Oracle Solaris will export the ZFS pool, and if this is the last LUN presented to this domain, it will also remove the discovery address and disable static LUN discovery.

This provides for a very clean way to associate zones with iSCSI LUNs in a way that is easily portable across domains.

However, this feature is available only when using one zone per iSCSI LUN. If multiple zones are to share an iSCSI LUN, that LUN must be manually presented to Oracle Solaris and a ZFS pool must be manually created upon it as previously discussed.

Identify Network Interfaces

In this section, you will specify the network interfaces for the zone.

You will first need to identify all of the relevant network interfaces in the global zone. Because redundancy is created using IPMP, this means that for every network to present to the zone there will be two underlying network interfaces to locate in the global zone and to present individually into the new zone. The IPMP group will then be re-created within the new zone.

In Oracle Solaris, the Layer 2 elements are called “datalinks” and the Layer 3 elements are called “interfaces.” Datalinks can be physical or virtual (VNIC). Interfaces are created on top of datalinks. Only interfaces have IP addresses (because IP is a Layer 3 protocol). IPMP groups are also considered to be an “interface” and have an assigned IP address.

In the case where LACP is being used for the 10 GbE network, there will be only one interface to locate and present to the new zone.

In this example, we will identify all of the standard networks in SuperCluster and plumbed into the zone. However, in practice you can choose which networks you would like to present.

Identify IPMP Groups

Because all networks use IPMP groups, the easiest way to discover them is to simply list the IPMP groups on a system:

```
# ipmpstat -g
GROUP        GROUPNAME    STATE  FDT        INTERFACES
scm_ipmp0    scm_ipmp0    ok     --         net2 (net3)
sc_ipmp0     sc_ipmp0     ok     --         net0 (net1)
stor_ipmp0   stor_ipmp0   ok     --         stor_ipmp0_0 (stor_ipmp0_1)
#
```

So, on this system we have the following three networks:

- The management network, which has the IPMP group name of `scm_ipmp` and is composed of two datalinks: `net2` and `net3`. This IPMP group is in an active/passive configuration with the active datalink being `net2` and the passive one being `net3` (because it is in parentheses).
- The 10 GbE client access network, which has the IPMP group name of `sc_ipmp0` and is composed of two datalinks: `net0` and `net1`. This IPMP group is in an active/passive configuration[9] with the active datalink being `net0` and the passive one being `net1`.
- The ZFS partition on the InfiniBand network, which has the IPMP group name of `stor_ipmp0` and is composed of two datalinks: `stor_ipmp0_0` and `stor_ipmp0_1`. This IPMP groups is in an active/passive configuration with the active datalink being `stor_ipmp0_0` and the passive one being `stor_ipmp0_1`.

Depending on the configuration of your particular app domain, you could have additional 10 GbE interfaces (for example, a backup network) or even additional InfiniBand partitions (for example, for Oracle Solaris Cluster).

[9] Only active/passive IPMP groups are utilized on SuperCluster.

Note the names of the IPMP groups and their underlying associated datalinks. You will need these when creating the zone configuration file as well as when re-creating the IPMP groups within the zone.

If you look carefully at the IP addresses assigned in your domain, you will notice that the IP addresses are actually assigned directly to the IPMP groups and not to the datalinks themselves.

```
# ipadm show-addr
ADDROBJ            TYPE     STATE        ADDR
lo0/v4             static   ok           127.0.0.1/8
stor_ipmp0/v4      static   ok           192.168.28.4/22
sc_ipmp0/v4        static   ok           10.129.118.102/20
scm_ipmp0/v4       static   ok           10.129.108.112/20
lo0/v6             static   ok           ::1/128
#
```

It is also important to understand the underlying interfaces. Examine the output of the following commands:

```
# dladm show-link
LINK                  CLASS     MTU    STATE    OVER
net1                  phys      1500   up       --
net4                  phys      65520  up       --
net2                  phys      1500   up       --
net5                  phys      65520  up       --
net0                  phys      1500   up       --
net3                  phys      1500   up       --
stor_ipmp0_0          part      65520  up       net5
stor_ipmp0_1          part      65520  up       net4

# dladm show-phys
LINK                MEDIA                STATE      SPEED  DUPLEX    DEVICE
net1                Ethernet             up         10000  full      ixgbe1
net4                Infiniband           up         32000  unknown   ibp0
net2                Ethernet             up         0      unknown   vnet0
net5                Infiniband           up         32000  unknown   ibp1
net0                Ethernet             up         10000  full      ixgbe0
net3                Ethernet             up         0      unknown   vnet1

# dladm show-part
LINK          PKEY  OVER          STATE    FLAGS
stor_ipmp0_0 8503  net5          up       f---
stor_ipmp0_1 8503  net4          up       f---
#
```

There are several important observations to be made from this output that will be relevant to configuring a zone.

- It is possible to identify which interfaces belong to the 10 GbE, InfiniBand, and management networks (see the `MEDIA` and `SPEED` columns in the output of the dladm `show-phys` command).
- Note that the management network has a speed of 0 with interface names `of vnet0 and vnet1`. This is because this is a "middle" domain[10] with virtualized interfaces for the management network. We will need this information later.
- Note that the two datalinks that are identified as being members of the `stor_ipmp0` IPMP group do not show up as physical datalinks. Instead they show up as InfiniBand partitions belonging to datalinks `net4` and `net5`, which are in fact the real InfiniBand datalinks. This information will also be required later.
- Note the InfiniBand partition key (pkey) in the output of the `dladm show-part` command. This is synonymous to a VLAN tag on Ethernet, and we will need this partition key when creating interfaces for the zones.

Creating VNICs

We cannot simply "hand over" to the new zone the IP interfaces used for communication in the global zone, because they are already in use. Instead we will create new virtual network interfaces or, more precisely, we will create new virtual datalinks (known as VNICs) on top of the underlying physical datalinks.

A special note about InfiniBand datalinks: It is not possible to simply create a VNIC on top of an Infiniband datalink. Instead we must create a *partition* datalink by specifying the pkey. In practice, we achieve the same thing (a partition datalink functions like a VNIC) but the nomenclature and syntax are slightly different.

We could manually create all of the VNICs we need in the global zone using the commands dladm `create-vnic` and `dladm create-part` and then specify them in the respective zone configuration files. However, we would then have to manually manage the association of all of these with their respective domains.

Instead, Oracle Solaris makes this very simple for us with the introduction of the "anet" resource in Oracle Solaris 11.

When we specify an anet resource in a zone configuration, Oracle Solaris will automatically create the required VNIC (or InfiniBand partition) when the zone boots and destroy it when the zone shuts down or is halted.

Here is an example of the network portion of the zone configuration file for the 10 GbE client access network. Note that because this example is for IPMP, there are two datalinks to plumb into the zone. The IPMP group will be created within the zone later.

```
add anet
set linkname=sc_ipmp0_0
set lower-link=net0
end
add anet
set linkname=sc_ipmp0_1
set lower-link=net1
end
```

[10] A "middle" domain exists only on systems that have more than two domains. On such systems, only the "outer" two domains (the left and right domains) have physical NICs for the management network and boot disks. All other domains are in the "middle" as are their management network interfaces and boot disks.

The value specified for `linkname` is what the datalink will be named inside the zone. The value specified for `lower-link` is the name of the underlying datalink in the global zone that a VNIC will be created on top of in order to create the new datalink inside the zone.

If you are using VLAN tagging, then specify the VLAN tag here in the zone configuration file, and the new VNIC will automatically be on that VLAN. No further configuration with regard to VLAN tagging will be required in the zone itself. Note that you must specify the same VLAN tag for both underlying interfaces of the IPMP group.

```
add anet
set linkname=sc_ipmp0_0
set lower-link=net0
set vlan-id=200
end
add anet
set linkname=sc_ipmp0_1
set lower-link=net1
set vlan-id=200
end
```

If you have additional 10 GbE networks (for example, for backup), they can be handled exactly the same way. Simply replace the values to match the associated IPMP group settings.

Note that at this stage, the IPMP group has not been specified nor have we specified any IP addresses. All of this will be performed inside the zone once it is installed and booted.

If you are using LACP instead of IPMP on the 10 GbE network, the configuration is simpler because only one VNIC needs to be created on top of the aggregation.

```
add anet
set linkname=aggr0
set lower-link=aggr0
end
```

Likewise, if you plan to use VLAN tags with LACP, the example looks like this:

```
add anet
set linkname=aggr0
set lower-link=aggr0
set vlan-id=200
end
```

Here an example of the configuration for the management network (which always uses IPMP).

```
add anet
set linkname=scm_ipmp0_0
set lower-link=net2
set mac-address=auto
end
add anet
set linkname=scm_ipmp0_1
set lower-link=net3
set mac-address=auto
end
```

Note that in this case, we have added `set mac-address=auto`. This is required in Oracle Solaris 11.1 because the underlying interface is a VNET.[11] In Oracle Solaris 11.1, the default value for `mac-address` in a zone configuration is random, which is incompatible with creating a VNIC on a VNET.

If you are creating a zone in an "outer" domain—that is, one where the management interface is a physical NIC and not a VNET—you do not need to add this extra setting. Also, in Oracle Solaris 11.2, it should[12] no longer be necessary to specify this, because the default value in Oracle Solaris 11.2 is `auto`.

In summary, the handling of a virtual interface (VNIC) on top of a virtual interface (VNET) in Oracle Solaris 11.1 requires that the assignment of MAC addresses happens in a particular way, which `mac-address=auto` handles properly.

Prior to Oracle Solaris 11.1, it was not possible to create a VNIC on top of a VNET, and if you are creating zones in such an older environment, you must create additional VNETs for each zone[13].

Specifying the IPMP group for the InfiniBand partitions is similar, but the partition key must be specified:

```
add anet
set linkname=stor_ipmp0_0
set lower-link=net4
set pkey=0x8503
end
add anet
set linkname=stor_ipmp0_1
set lower-link=net5
set pkey=0x8503
end
```

Note that the value for `lower-link` is derived through the additional steps explained earlier to match the InfiniBand partition with the underlying datalinks.

[11] A VNET is a virtual network interface provided to this domain as a service from one of the "outer" domains in SuperCluster that actually own the physical NICs for the management network.

[12] It was not possible to test and verify this on SuperCluster prior to the completion of this paper.

[13] This procedure is documented in the document entitled *SuperCluster T5-8 Zones with Oracle Database on Database Domains*, which is not available externally. Ask your Oracle representative for a copy.

In the example above, an InfiniBand partition datalink with the specified key will be created automatically in the global zone when this zone boots and will subsequently be removed when the zone is halted or shut down. Because this process is automatic, there are no “dangling” datalinks to handle.

Putting It All Together

Create a file with the desired zone configuration, as discussed previously. You can name this file using the name of the zone and a `.cfg` suffix, for example, `zone1.cfg`.

This file should look something like this (if you are creating multiple zones per iSCSI LUN):

```
create -b
set zonepath=/zones/zone1
set autoboot=true
set ip-type=exclusive

add anet
set linkname=sc_ipmp0_0
set lower-link=net0
end
add anet
set linkname=sc_ipmp0_1
set lower-link=net1
end

add anet
set linkname=scm_ipmp0_0
set lower-link=net2
set mac-address=auto
end
add anet
set linkname=scm_ipmp0_1
set lower-link=net3
set mac-address=auto
end

add anet
set linkname=stor_ipmp0_0
set lower-link=net4
set pkey=0x8503
end
add anet
set linkname=stor_ipmp0_1
set lower-link=net5
set pkey=0x8503
end
```

Note: If you are using LACP for your 10 GbE network(s), replace that section of the above file with the example provided earlier using `aggr0`.

Note: If you are using VLAN tagging, then specify the VLAN tags per the examples above.

If you are creating one zone per iSCSI LUN, then your zone configuration file will look like this:

```
create -b
set zonepath=/zones/zone1
set autoboot=true
set ip-type=exclusive

add rootzpool
add storage
iscsi://192.168.28.1:3620/luname.naa.600144F0F69444CB00005488A575
0001
end

add anet
set linkname=sc_ipmp0_0
set lower-link=net0
end
add anet
set linkname=sc_ipmp0_1
set lower-link=net1
end

add anet
set linkname=scm_ipmp0_0
set lower-link=net2
set mac-address=auto
end
add anet
set linkname=scm_ipmp0_1
set lower-link=net3
set mac-address=auto
end

add anet
set linkname=stor_ipmp0_0
set lower-link=net4
set pkey=0x8503
end
add anet
set linkname=stor_ipmp0_1
set lower-link=net5
set pkey=0x8503
end
```

Configuring the Zone

Finally, configure the zone in Oracle Solaris using the above `zone1.cfg` file.

```
# zonecfg -z zone1 -f zone1.cfg
```

The above example creates and configures a zone named `zone1`.

The configuration is read from the file `zone1.cfg.`

```
# zoneadm list -icv
ID NAME              STATUS       PATH           BRAND      IP
0  global            running      /              solaris    shared
1  zone1             configured   /zones/zone1   solaris    excl
```

Note that at this point, the zone is only configured (that is, it has been defined). The next steps will be to install it, boot it, and configure the networking and services within it.

Installing App Zones

Overview

The process of installing a zone consists of either installing all of the Oracle Solaris packages specified by the zone manifest (or the default manifest, if none is specified) or cloning an existing zone.

This process performs the work necessary to make the zone “bootable.”

Network Interfaces

A note about configuring network interfaces: During the configuration process, we specified all of the Layer 2 network datalinks that are to be created and presented within the zone when the zone boots. Note that actually configuring the Layer 3 interfaces on top of these datalinks is something that occurs within Oracle Solaris when the zone is booted and running.

Zone Configuration

When a new zone is created (or cloned), it is in the “unconfigured”[14] state. The first time the zone is booted, it must be “configured ” using the `sysconfig`(1M) tool. This can either be performed interactively via the console of the zone when it boots or automatically by specifying a system configuration profile XML file when the zone is installed.

[14] This refers to the configuration of services within the actual zone, rather than to the configuration of the zone itself within the global zone (or domain), which was discussed in the previous section. While both are referred to as “configuration,” it might be easier to think of the earlier process as “defining” the zone and this part as actually configuring services within the zone.

In either case, it is not possible to specify the more-complex network configurations, such as IPMP groups used in SuperCluster, during the `sysconfig`(1M) process. The `sysconfig`(1M) process is capable of only very basic network configuration. For this reason, this guide presents a procedure whereby no Layer 3 network interfaces or services are configured during the `sysconfig`(1M) process. All of this configuration will be performed by a custom script that will be run after the zone is up and running, past the `sysconfig`(1M) step.[15]

So you have two choices. Either create the `sysconfig`(1M) XML file and specify it during zone installation or manually log in to the zone via console of the zone when it boots (using the `-C` option of the `zlogin` command) and walk through the configuration manually, specifying no networks. The latter might be simpler if you only have one or two zones to create. Using the XML file scales better when many zones need to be created.

In this example, we will create the `sysconfig`(1M) XML file and make some edits to it manually.

Create the System Configuration XML File

Run the following command to begin the process of creating the `sysconfig`(1M) XML file.

Note: The output location specified with `-o` is the location where the output will be saved. Prior to Oracle Solaris 11.1, this is the name of a directory where the XML file will be created. So, in this case the output will go to `./zone1/sc_profile.xml`. Since Oracle Solaris 11.1, this is simply the name of the output file itself.

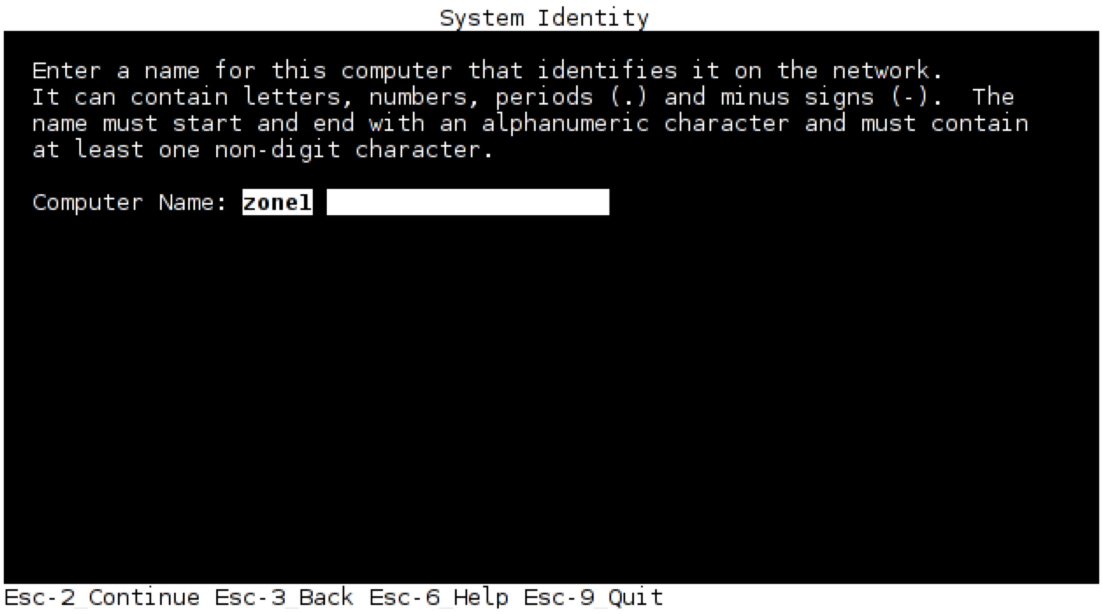
It is best to run this command in an 80x24 terminal window, because this is a character-based user interface.

```
# sysconfig create-profile -o zone1
```

Note: When the first screen comes up, you might be prompted to press F2 to continue. If your keyboard does not have an F2 function key or if it does not work, you may also use ESC-2 (and ESC-3 and so on) as well as the arrow keys to navigate.

Specify the host name for your zone.

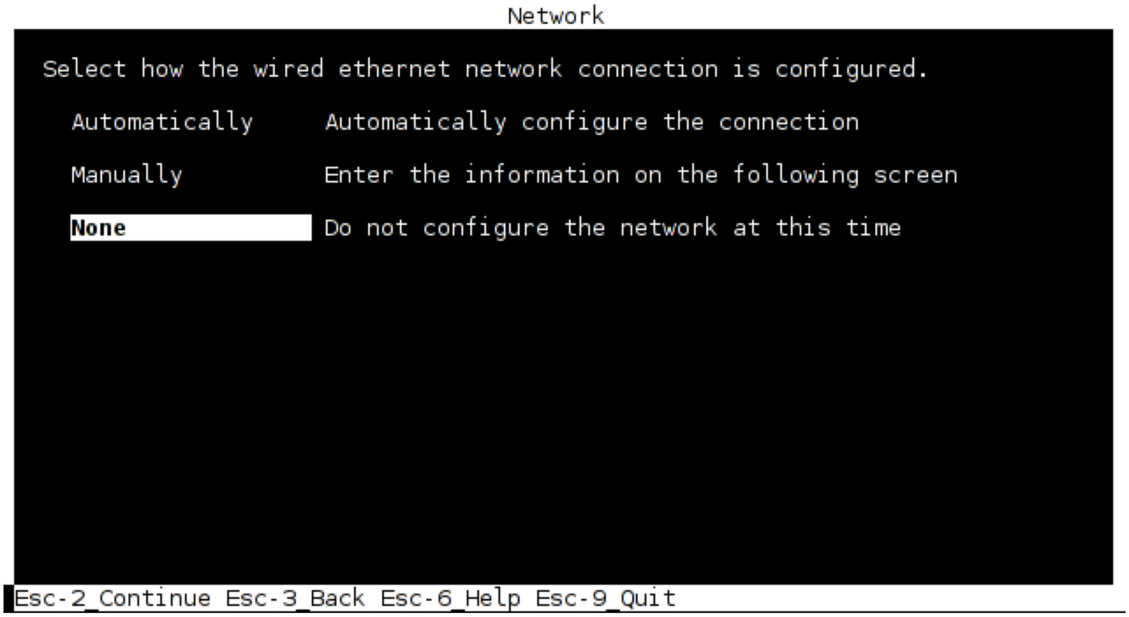
[15] This procedure configures things such as the host name, time zone, root password, and IP addresses for network interfaces, router, name service, and so on.



```
                        System Identity

 Enter a name for this computer that identifies it on the network.
 It can contain letters, numbers, periods (.) and minus signs (-).  The
 name must start and end with an alphanumeric character and must contain
 at least one non-digit character.

 Computer Name: zone1

Esc-2_Continue Esc-3_Back Esc-6_Help Esc-9_Quit
```

Figure 26. System Identity screen

Specify that this zone will not have any network interfaces configured at this point (they will be configured by a custom script later).



```
                            Network

 Select how the wired ethernet network connection is configured.

   Automatically     Automatically configure the connection

   Manually          Enter the information on the following screen

   None              Do not configure the network at this time

Esc-2_Continue Esc-3_Back Esc-6_Help Esc-9_Quit
```

Figure 27. Network screen

On the subsequent screens, you will be prompted to enter your time zone, language, root password, and information for a user account to be created.

Complete the configuration process. The command will terminate indicating the location of the newly created XML file. In this case, it is `./zone1/sc_profile.xml.`

If you will be creating multiple zones, you can simply make copies of this file and edit each one in order to change the value specified for `nodename` for each one. Here’s the relevant section of the file:

```
<service version="1" type="service" name="system/identity">
    <instance enabled="true" name="node">
      <property_group type="application" name="config">
        <propval type="astring" name="nodename" value="zone1"/>
      </property_group>
    </instance>
  </service>
```

Simply change `value=”zone1”` to `value=”zone2”`, `value=”zone3”`, and so on.

Now, install the zone using the newly created XML profile.

```
# zoneadm -z zone1 install -c $PWD/zone1/sc_profile.xml
```

Note: You must specify a full path to the XML file; otherwise, `zoneadm` will produce an error[16].

Note: It is not necessary to maintain the `zone1/sc_profile.xml` file structure or name. You can simply rename your XML files `zone1.xml`, `zone2.xml`, `zone3.xml`, and so on. Just remember to specify the full path to `zoneadm` when using them.

Alternatively, if you choose not to create a `sysconfig`(1M) XML file, install the zone without one.

```
# zoneadm -z zone1 install
```

Once the process has completed, your zone status should change to `installed`.

```
# zoneadm list -icv
ID NAME              STATUS        PATH            BRAND       IP
0  global            running       /               solaris     shared
1  zone1             installed     /zones/zone1    solaris     excl
```

Repeat this process for each zone that is to be created. Note that you do not need to rerun `sysconfig`(1M) for each zone. Simply copy and edit the XML file as discussed previously.

[16] This has been fixed in Oracle Solaris 11.2.

Booting and Configuring App Zones

Overview

At this point, the zone has been installed and is ready to boot and have its services and network interfaces configured.

Note that the process of “booting” a zone isn’t really the same as booting the Oracle Solaris kernel. Because a zone shares the same kernel with the global zone, what actually happens during this “booting” phase is that all of the necessary Oracle Solaris services are started and the environment is configured.

Initial Boot

For the first time the zone is booted, it is helpful to watch the process from the zone’s console in case any problems arise. Therefore, two separate windows are recommended for this procedure: one for the console and one to initiate the boot process from the global zone.

In the first window, open a console for the zone.

```
# zlogin -C zone1
```

In the second window, boot the zone.

```
# zoneadm -z zone1 boot
```

The output in the first window should look something like this, with no errors:

```
# zlogin -C zone1
[Connected to zone 'zone1' console]

[NOTICE: Zone booting up]

SunOS Release 5.11 Version 11.2 64-bit
Copyright (c) 1983, 2014, Oracle and/or its affiliates. All rights
reserved.
Loading smf(5) service descriptions: 134/134
Hostname: zone1

zone1 console login: ~.
[Connection to zone ’zone1’ console closed]
```

However, if you did not specify a `sysconfig(1M)` XML file when installing the zone, you would at this point also be prompted for the same information that is requested when creating the XML file. If so, complete all of the steps and make sure to not specify any network interfaces.

At this point, your zone is up and running and you are ready to configure the network interfaces and services.

Configure Zone Network Interfaces and Services

Look at the configuration of the datalinks and interfaces within the zone. You should see that all of the datalinks specified in the zone configuration are present but that no interfaces have been created upon them.

```
# dladm show-link
LINK                CLASS     MTU    STATE    OVER
stor_ipmp0_0        part      65520  up       ?
stor_ipmp0_1        part      65520  up       ?
sc_ipmp0_0          vnic      1500   up       ?
sc_ipmp0_1          vnic      1500   up       ?
scm_ipmp0_0         vnic      1500   up       ?
scm_ipmp0_1         vnic      1500   up       ?
# ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
```

We will now create a custom shell script to configure the desired networks and services within the zone. We will create this script in the global zone and keep it with our zone configuration and `sysconfig`(1M) XML files. Once the script is ready, it can be copied directly into the zone.

Note that once the zone has been booted, its root ZFS pool and file systems are visible to the root user with the global zone. We can use this mechanism to easily transfer files (such as our configuration script) into the zone.

If you specified `zoneroot` as `/zones/zone1` when creating the zone, then `/var/tmp` within the zone will be visible as `/zones/zone1/root/var/tmp` from the global zone (but only to `root`). This is how we will copy the script from the global zone. This will allow us to have a different configuration script for each zone and quickly copy it into the zone when needed. We will then use the `zlogin` command to execute the script within the zone.

Create and Configure IPMP Groups

In order to create and configure an IPMP group within the zone, the following steps must be performed.

1. Layer 3 interfaces must be created on top of each Layer 2 datalink.

2. The IPMP group must be created based on the two interfaces.

3. One of the interfaces must be set to standby mode in order to create an active/passive IPMP group. Take care to note which interface is passive in the global zone and make the same interface passive in the zone.

4. The IP address and netmask must be assigned to the IPMP group.

Here is an example of the commands for performing these steps for the management network:

```
ipadm create-ip scm_ipmp0_0
ipadm create-ip scm_ipmp0_1
ipadm create-ipmp -i scm_ipmp0_0,scm_ipmp0_1 scm_ipmp0
ipadm set-ifprop -p standby=on -m ip scm_ipmp0_1
ipadm create-addr -a local=10.129.108.120/20 scm_ipmp0/v4
```

You can repeat this block of commands for each IPMP group that you need to define, changing the interface names, IP address, and netmask as necessary.

Create and Configure LACP Interfaces

If you are using LACP for the 10 GbE network, then you simply have one VNIC plumbed into your zone from the global zone. This VNIC is automatically created on top of the aggregation, so the configuration is simpler.

```
ipadm create-ip aggr0
ipadm create-addr -a local=10.129.118.110/24 aggr0/v4
```

Additional network configuration and services need to be defined. These include setting the default router, configuring DNS, and enabling the NFS client (if you plan to use NFS).

Note: The NFS client’s Oracle Solaris Service Management Facility (SMF) service would normally already be enabled. However, because we initially configured the zone with no network interfaces, it was not enabled when the zone was initially booted.

Putting It All Together (Example Script)

An example network configuration script is shown below. Note that the exact networks, IP addresses, and DNS information will be different for your environment. This particular script also includes a backup network, just to show what configuring one might look like. This script example is for IPMP.

```
#
# SuperCluster App Zone IP Configuration
#
# Mikel Manitius
# Thu Oct  9 02:26:46 PDT 2014
#
# Note: standby interface in IPMP group should match that of the Global Zone
#

## Management Network
ipadm create-ip scm_ipmp0_0
ipadm create-ip scm_ipmp0_1
ipadm create-ipmp -i scm_ipmp0_0,scm_ipmp0_1 scm_ipmp0
ipadm set-ifprop -p standby=on -m ip scm_ipmp0_1
ipadm create-addr -a local=10.196.16.242/20 scm_ipmp0/v4

## Client Access Network
ipadm create-ip sc_ipmp0_0
ipadm create-ip sc_ipmp0_1
ipadm create-ipmp -i sc_ipmp0_0,sc_ipmp0_1 sc_ipmp0
ipadm set-ifprop -p standby=on -m ip sc_ipmp0_1
ipadm create-addr -a local=192.168.100.242/24 sc_ipmp0/v4

## Backup Network (optional)
ipadm create-ip sc_ipmp1_0
ipadm create-ip sc_ipmp1_1
ipadm create-ipmp -i sc_ipmp1_0,sc_ipmp1_1 sc_ipmp1
ipadm set-ifprop -p standby=on -m ip sc_ipmp1_1
ipadm create-addr -a local=10.10.10.107/24 sc_ipmp1/v4


## ZFS IB Network
ipadm create-ip stor_ipmp0_0
ipadm create-ip stor_ipmp0_1
ipadm create-ipmp -i stor_ipmp0_0,stor_ipmp0_1 stor_ipmp0
ipadm set-ifprop -p standby=on -m ip stor_ipmp0_1
ipadm create-addr -a local=192.168.66.153/20 stor_ipmp0

## Default Router
route -p add default 10.196.16.1

## Configure DNS
svccfg -s dns/client setprop config/nameserver = net_address: "(10.196.48.21
10.196.48.22)"

svccfg -s dns/client setprop config/domain = astring: "osc.us.oracle.com"
svccfg -s name-service/switch setprop config/host = astring: \"files dns\"
svcadm refresh name-service/switch
svcadm refresh dns/client
nscfg export svc:/network/dns/client:default
svcadm enable nfs/client
```

Here is the same script example for LACP on the 10 GbE networks.

```
#
# SuperCluster App Zone IP Configuration
#
# Mikel Manitius
# Thu Oct  9 02:29:33 PDT 2014
#
# Note: standby interface in IPMP group should match that of the Global Zone
#

## Management Network
ipadm create-ip scm_ipmp0_0
ipadm create-ip scm_ipmp0_1
ipadm create-ipmp -i scm_ipmp0_0,scm_ipmp0_1 scm_ipmp0
ipadm set-ifprop -p standby=on -m ip scm_ipmp0_1
ipadm create-addr -a local=10.196.16.242/20 scm_ipmp0/v4

## Client Access Network
ipadm create-ip aggr0
ipadm create-addr -a local=192.168.100.242/24 aggr0/v4

## Backup Network (Optional)
ipadm create-ip aggr1
ipadm create-addr -a local=10.10.10.107/24 aggr1/v4


## ZFS IB Network
ipadm create-ip stor_ipmp0_0
ipadm create-ip stor_ipmp0_1
ipadm create-ipmp -i stor_ipmp0_0,stor_ipmp0_1 stor_ipmp0
ipadm set-ifprop -p standby=on -m ip stor_ipmp0_1
ipadm create-addr -a local=192.168.66.152/20 stor_ipmp0

## Default Router
route -p add default 10.196.16.1

## Configure DNS
svccfg -s dns/client setprop config/nameserver = net_address: "(10.196.48.21
10.196.48.22)"

svccfg -s dns/client setprop config/domain = astring: "osc.us.oracle.com"
svccfg -s name-service/switch setprop config/host = astring: \"files dns\"
svcadm refresh name-service/switch
svcadm refresh dns/client
nscfg export svc:/network/dns/client:default
svcadm enable nfs/client
```

Run the Network Configuration Script

Save your script to a file and give it a name, for example, `zone1_net.sh`.

Ensure that the script is executable and then copy it into the zone and execute it.

```
# chmod 755 zone1_net.sh
# cp zone1_net.sh /zones/zone1/root/var/tmp
# zlogin zone1 /var/tmp/zone1_net.sh
```

The zone's network interfaces and services are now configured. This configuration will be retained across reboots.

As a final step, you can update the `/etc/hosts` file within the zone to reflect the IP addresses and host names for the newly created interfaces. An alternative approach would be to update the `/etc/hosts` file in the global zone and simply copy it into the newly created zone(s).

CPU Resource Controls

Overview

Unless otherwise configured, zones share CPU resources with the global zone.

It is possible to dedicate CPU resources to a zone. When this is done, the CPU resources are "taken away" from the global zone and dedicated to the specified zone.

Note that you cannot assign all of the CPU resources to non-global zones. The global zone requires some cores to handle I/O operations and other system tasks. The minimum number of cores required in the global zone is two if you have one InfiniBand host channel adapter (HCA) in the domain, and it is four or more if you have more than one InfiniBand HCA in the domain. Consult the *SuperCluster Owner's Guide*[17] for current guidelines and details.

Threads Versus Cores

It is important to understand that Oracle Solaris CPU allocation is based on vCPUs, which are threads. The S3 core in Oracle's SPARC T5 and M6 processors has 8 threads per core (and 16 cores per SPARC T5 processor socket or 12 cores per SPARC M6 processor socket). While a deeper discussion about performance of threads versus cores is beyond the scope of this paper, best practice for performance is to allocate whole cores (meaning multiples of 8 vCPUs).

Software License Boundaries

When properly configured, zones with configured CPU resource controls are considered valid "hard partitioning" license boundaries for Oracle software.

Further details can be obtained from the "Hard Partitioning with Oracle Solaris Zones" white paper:

http://www.oracle.com/technetwork/server-storage/solaris11/technologies/os-zones-hard-partitioning-2347187.pdf

[17] Request a copy from your Oracle representative. It is not available for download on oracle.com.

Dedicating CPU Resources to a Zone

The simplest mechanism is to specify the amount of CPU resources dedicated to a zone within its configuration using the `zonecfg` command.

```
# zonecfg -z zone1
zonecfg:zone1> add dedicated-cpu
zonecfg:zone1:dedicated-cpu> set ncpus=32
zonecfg:zone1:dedicated-cpu> end
zonecfg:zone1> commit
zonecfg:zone1> exit
```

The syntax above assigns 32 vCPUs (threads) for exclusive use by this zone. While the syntax specifies vCPUs, in practice when using multiples of 8, Oracle Solaris will generally assign threads on whole core boundaries[18], though this behavior is not guaranteed.

When this zone boots, a temporary processor set will be created. The specified number of cores will be transferred from the default processor set belonging to the global zone and assigned to a temporary processor set created for this zone. When the zone is shut down or halted, the temporary processor set will be destroyed and its vCPUs will be returned to the default processor set in the global zone.

If the specified resources are not available (for example, if they are assigned to another zone), then the zone will fail to boot.

Care must be taken to not starve the global zone of its minimum required number of cores. These minimum requirements are documented in the *SuperCluster Owner's Guide*.

For most situations, this approach offers a good mix of ease of administration, flexibility, and performance. However, Oracle Solaris 11.2 introduces additional mechanisms for assigning CPU resources. The `cores` and `sockets` keywords can be utilized to specify individual cores and sockets, guaranteeing whole core or socket boundaries.

The trade-off for this approach is increased administrative complexity. In this approach, individual cores or sockets must be identified and specified. You will need to keep track of which specific cores or sockets are in use by which zone. If CPU boundaries are later reallocated across domains, a zone might fail to boot because the specific cores it is requesting are no longer assigned to that domain.

The procedure for assigning specific cores consists of first discovering the processor topology of the domain and then selecting and specifying individual cores.

[18] The S3 core shared by the SPARC T5 and SPARC M6 processor has 8 threads per core.

The following command shows the topology of the processors assigned to the domain.

```
# psrinfo -t
socket: 12
  core: 201457665
    cpus: 1536-1543
  core: 201654273
    cpus: 1544-1551
  core: 201850881
    cpus: 1552-1559
  core: 202047489
    cpus: 1560-1567
  core: 202244097
    cpus: 1568-1575
  core: 202440705
    cpus: 1576-1583
  core: 202637313
    cpus: 1584-1591
  core: 202833921
    cpus: 1592-1599
  core: 203030529
    cpus: 1600-1607
  core: 203227137
    cpus: 1608-1615
  core: 203423745
    cpus: 1616-1623
  core: 203620353
    cpus: 1624-1631
socket: 65548
  core: 203816961
    cpus: 1632-1639
  core: 204013569
    cpus: 1640-1647


[ … truncated … ]
```

To assign four specific cores to the zone, use the following command:

```
# zonecfg -z zone1
zonecfg:zone1> add dedicated-cpu
zonecfg:zone1:dedicated-cpu> set
cores=201457665,201654273,201850881,202047489
zonecfg:zone1:dedicated-cpu> end
zonecfg:zone1> commit
zonecfg:zone1> exit
```

To assign whole sockets instead of cores, simply use the syntax `set sockets=` in the example above. Again, care must be taken to not starve the global zone of the minimum required number of cores.

The changes using either `ncpus=`, `cores=`, or `sockets=` in the examples above will not apply until the zone is booted (or rebooted if it is running).

However, beginning in Oracle Solaris 11.2, these changes can be applied to the zone dynamically, as follows:

```
# zoneadm -z zone1 apply
```

Advanced CPU Resource Allocation

The techniques presented thus far all assign dedicated CPU resources to a specific zone.

However, more-complex configurations are possible and might be desirable for more-flexible license management. One such approach is to manually create a processor set with a set number of cores and have multiple zones share these resources.

Let’s take the example of a SuperCluster used for both disaster recovery and nonproduction test and development work. Given a constraint of a limited number of available software licenses (say, 10 cores worth of licenses), you might want to dedicate a small number of cores (say, 4 cores) to the DR zone while the production SuperCluster is operating normally and the remaining 6 cores to the test and/or development zones.

If a DR event occurs, the test/development zones could be shut down or minimized, thus maximizing the CPU resources available to the DR zone.

This type of configuration can be accomplished by manually creating a processor set with 10 cores assigned to it and associating each zone with this processor set. In this configuration, each zone would share these 10 cores. However, further resource allocation or restriction can then be performed with the additional use of `ncpus=` to limit how many of those 10 cores from that processor set are available to each zone.

Reallocating cores from that processor set across zones assigned to it can be accomplished by simply changing the `ncpus=` value for each zone configuration, all the while never exceeding the 10 assigned to the governing processor set.

Implementation details are documented in the “Hard Partitioning with Oracle Solaris Zones” white paper:

http://www.oracle.com/technetwork/server-storage/solaris11/technologies/os-zones-hard-partitioning-2347187.pdf

Also, refer to the Oracle Solaris 11.2 documentation for *Creating and Using Oracle Solaris Zones*.

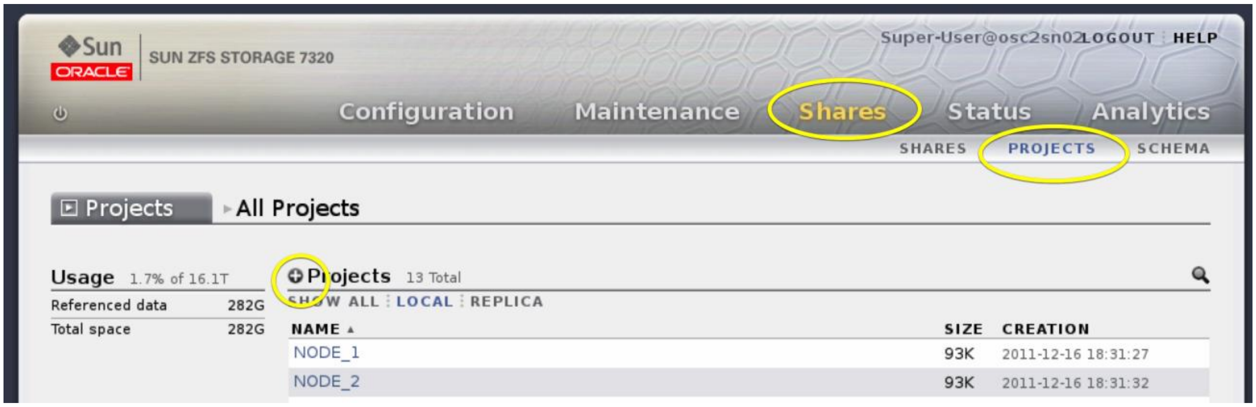
Configuring NFS Shares

In SuperCluster, iSCSI LUNs are only used for zone rpools. All general-purpose storage should use NFS. NFS shares can be created on the Oracle ZFS-SA and mounted by one or more zones. In order to mount an NFS share from the Oracle ZFS-SA, a zone must have available to it a network interface on the 0x8503 InfiniBand partition, as described earlier. It is also possible to mount NFS shares from elsewhere via your 10 GbE network.

One exception to this rule is if the NFS shares are to be used to hold database backups for use with Oracle Recovery Manager (Oracle RMAN) or for database files of a nonproduction database in the app zone. These are special cases where use of Oracle Database's DNFS client is preferable for performance. However, describing this configuration is beyond the scope of this paper.

1. Begin by creating a project on the Oracle ZFS-SA. This project will define the basic security settings that shares which are associated with it will inherit.

Log in to the Oracle ZFS-SA (as explained earlier in the section on creating iSCSI LUNs), navigate to the **Shares** tab and the **Projects** subtab. Click the **+** sign next to "Projects" to create a new project.



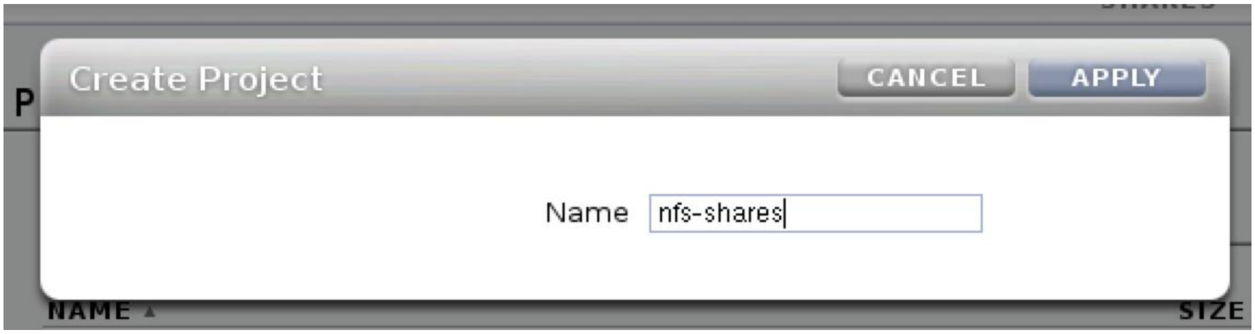
Usage 1.7% of 16.1T

Referenced data	282G
Total space	282G

NAME	SIZE	CREATION
NODE_1	93K	2011-12-16 18:31:27
NODE_2	93K	2011-12-16 18:31:32

Figure 28. Creating a new project

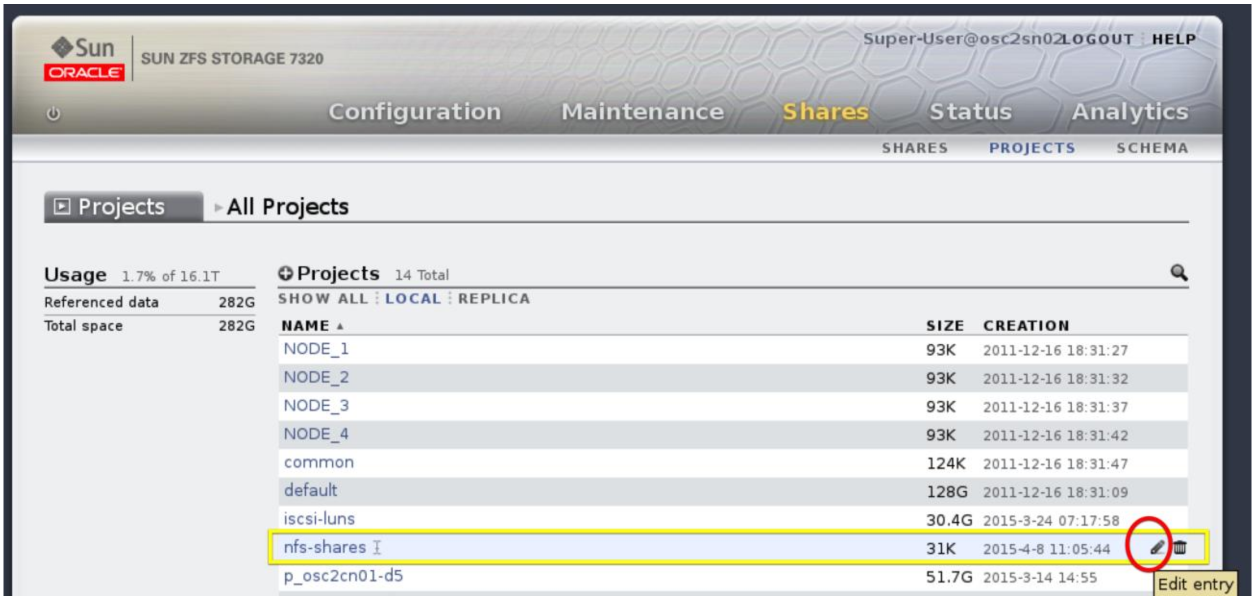
2. Give the project a name. In this example we use the name `nfs-shares`. In practice, you might wish to create different projects if you will be basing replication settings on them.



Name: nfs-shares

Figure 29. Naming the project

3. Once the project has been created, hover your mouse cursor over the project name. On the right side, a small edit icon will appear (circled in red below). Click this icon to edit the project's settings.



Usage 1.7% of 16.1T

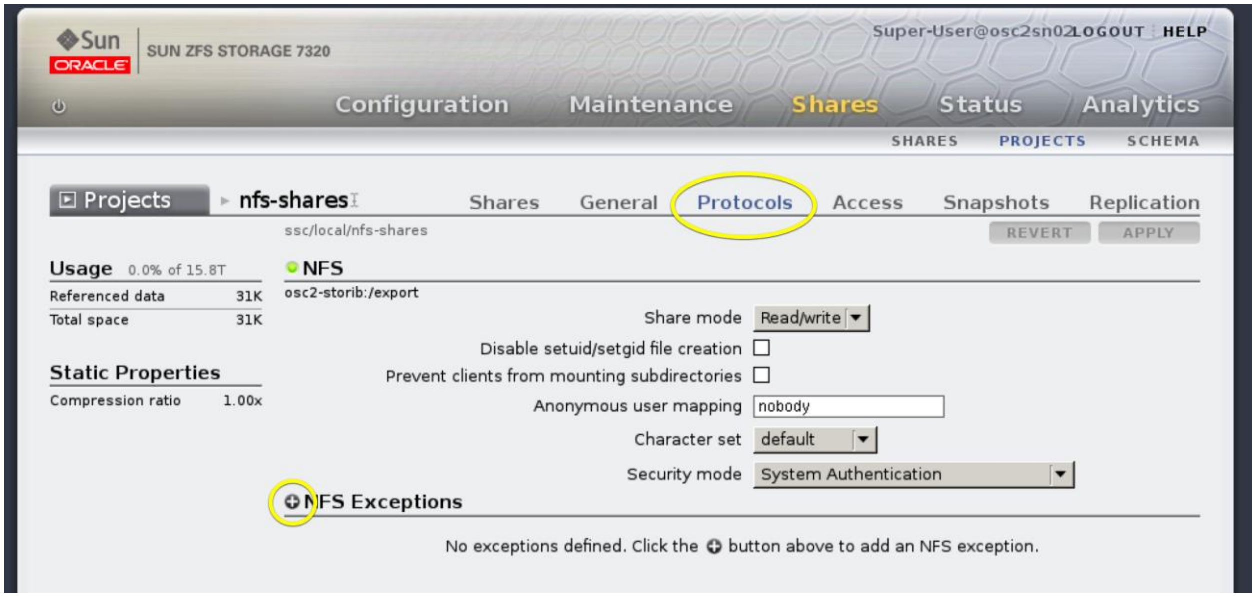
Referenced data	282G
Total space	282G

Projects 14 Total

NAME	SIZE	CREATION
NODE_1	93K	2011-12-16 18:31:27
NODE_2	93K	2011-12-16 18:31:32
NODE_3	93K	2011-12-16 18:31:37
NODE_4	93K	2011-12-16 18:31:42
common	124K	2011-12-16 18:31:47
default	128G	2011-12-16 18:31:09
iscsi-luns	30.4G	2015-3-24 07:17:58
nfs-shares	31K	2015-4-8 11:05:44
p_osc2cn01-d5	51.7G	2015-3-14 14:55

Figure 30. Editing the project settings

4. Navigate to the **Protocols** tab within the project and click the **+** sign to the left of “NFS Exceptions” to specify the NFS security settings.



ssc/local/nfs-shares

Usage 0.0% of 15.8T

Referenced data	31K
Total space	31K

Static Properties

Compression ratio	1.00x

NFS

osc2-storib:/export

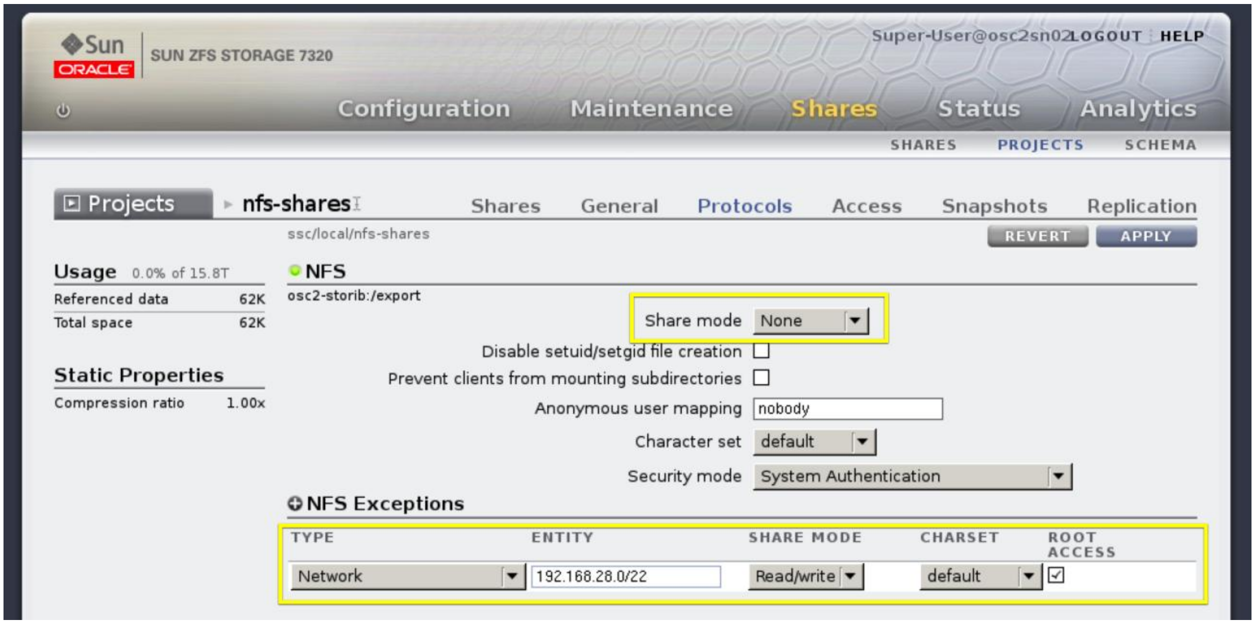
Share mode	Read/write
Disable setuid/setgid file creation	□
Prevent clients from mounting subdirectories	□
Anonymous user mapping	nobody
Character set	default
Security mode	System Authentication

⊕ NFS Exceptions

No exceptions defined. Click the ⊕ button above to add an NFS exception.

Figure 31. Specifying NFS security settings

The NFS Exceptions settings should appear below.



Usage 0.0% of 15.8T

Referenced data	62K
Total space	62K

Static Properties

Compression ratio	1.00x

NFS

osc2-storib:/export

Share mode: None

Disable setuid/setgid file creation ☐

Prevent clients from mounting subdirectories ☐

Anonymous user mapping: nobody

Character set: default

Security mode: System Authentication

NFS Exceptions

TYPE	ENTITY	SHARE MODE	CHARSET	ROOT ACCESS
Network	192.168.28.0/22	Read/write	default	☑

Figure 32. NFS exceptions settings

5. For the Share mode, select **None**. The NFS exceptions will specify what networks can see the share.

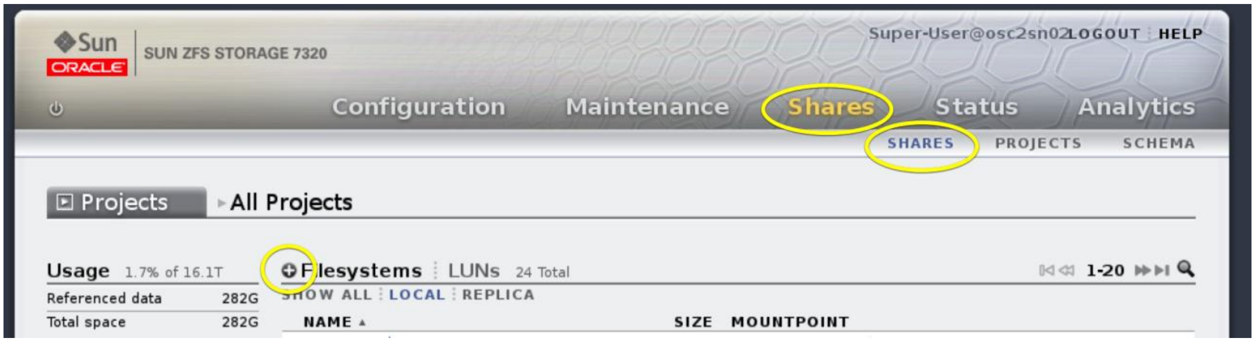
Ensure that the type is selected as **Network**.

In the **ENTITY** field, enter the subnet mask and CDIR of your InfiniBand network. This will ensure that the NFS share is visible only over that network (and not over the management network, for example).

Set SHARE MODE to **Read/Write**.

Enable **ROOT ACCESS** and then click the **APPLY** button.

6. Once the project has been configured, navigate to the **Shares** submenu and click the **+** sign next to "Filesystems."

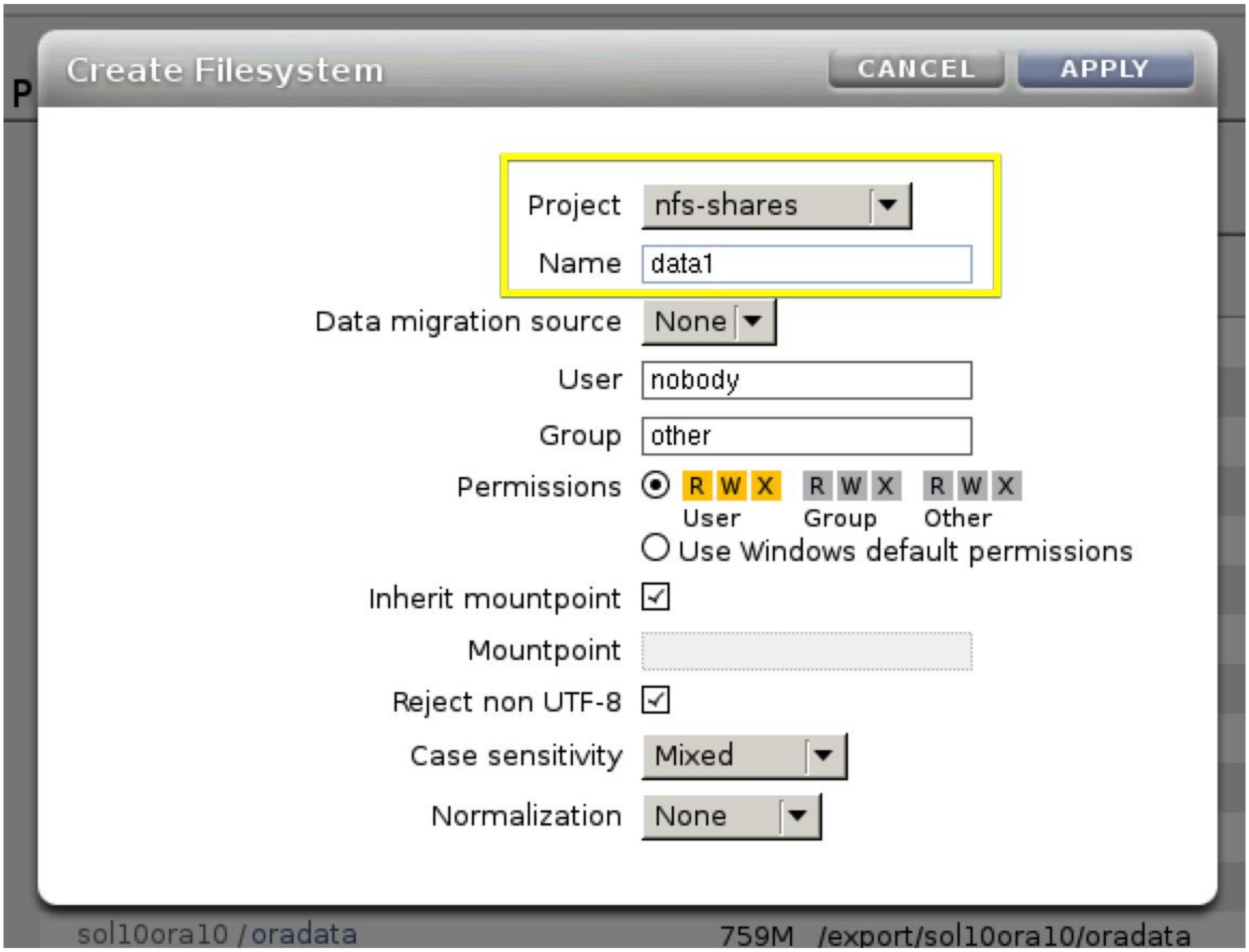


Usage 1.7% of 16.1T

Referenced data	282G
Total space	282G

Figure 33. Shares submenu

The Create Filesystem dialog box will appear.



Project: nfs-shares ▾
Name: data1
Data migration source: None ▾
User: nobody
Group: other
Permissions: ● User R W X, Group R W X, Other R W X
○ Use Windows default permissions
Inherit mountpoint: ☑
Mountpoint:
Reject non UTF-8: ☑
Case sensitivity: Mixed ▾
Normalization: None ▾

Figure 34. Create Filesystem dialog box

7. Ensure that the share is associated with the correct project and give your share a name.

Click the **APPLY** button.

At this point your NFS share has been created. By default your NFS share has no quota set; therefore, all of the remaining capacity in the Oracle ZFS-SA is available to this share. If you'd like to set a quota, you can do this either at the individual share level or for the project that the share inherits.

8. The remaining steps are to mount the NFS share inside your zone.

You can verify that the share is visible to your zone by using the following command:

```
# showmount -e 192.168.28.1
export list for 192.168.28.1:
/export/data1                 @192.168.28.0/22

[ … truncated … ]
```

Use the IP address of the Oracle ZFS-SA on the InfiniBand network. This is 192.168.28.1 by default and is usually[19] associated with the name `storIB` in the `/etc/hosts` file within the global zone.

Edit the `/etc/vfstab` file to create an entry for the NFS share. Use the following NFS mount options, which are best practice on SuperCluster.

```
rw,bg,hard,nointr,rsize=131072,wsize=131072,proto=tcp,vers=3
```

The block size is set to 128k, which is the default used by the Oracle ZFS-SA. NFS version 3 is recommended due to faster failover times in the case of a cluster head failover on the Oracle ZFS-SA.

In this example an entry for this share might look like this:

```
# tail -1 /etc/vfstab
192.168.28.1:/export/data1 -  /export/data1   nfs     -       yes
rw,bg,hard,nointr,rsize=131072,wsize=131072,proto=tcp,vers=3
```

Ensure the NFS client service is enabled in the zone.

```
# svcs nfs/client
STATE          STIME    FMRI
disabled       Apr_02   svc:/network/nfs/client:default
# svcadm enable -r nfs/client
```

Finally, create the mount point and mount the share

```
# mkdir /export/data1
# mount /export/data1
# df -h /export/data1
Filesystem             Size   Used  Available Capacity  Mounted on
192.168.28.1:/export/data1
                        16T    31K        16T     1%    /export/data1
```

9. Repeat this process for each zone where you want to mount this NFS share.

[19] You can change this default during the configuration planning process prior to installation. If you have chosen to change these values, use the ones you've specified. Use the *SuperCluster Deployment Summary Report* as a reference.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

Implementing Application Zones on Oracle SuperCluster
July 2015
Version 1.0
Author: Mikel Manitius