



Oracle ZFS Storage - High Availability User Guide

Configuration of an Oracle ZFS Storage - High Availability Instance
in Oracle Cloud Infrastructure (OCI) Using the Deployment Tool

November 2024 | Version 4.15
Copyright © 2024, Oracle and/or its affiliates
Public

CONTENTS

Purpose Statement	4
Disclaimer	4
Introduction	5
Overview of Configuration Steps	6
ZFS-HA Tenancy Architecture Overview	7
Compartment Overview	7
Network Overview Before ZFS-HA	7
Network Overview After ZFS-HA	8
OCI Requirements for ZFS-HA Clusters	8
ZFS Compute Instance and Network Requirements	8
Example - OCI Configuration for a ZFS-HA Cluster	10
First Steps	12
Installation Checklist	13
Run The ZFS Storage Deployment Tool From OCI Marketplace	14
Configure the Deployment Tool Variables	15
Apply the Stack	20
Set a Password for ZFS Administration	21
Connect to the Browser User Interface (BUI)	23
Distribute Shared Resources to Their Default Controllers	23
ZFS-HA Networking Recommendations	24
ZFS-HA Network -> Configuration -> Routing	24
ZFS-HA Network -> Configuration -> Datalinks	24
ZFS-HA Network -> Configuration->Interfaces	24
Share An SMB Filesystem	25
Share An NFS Filesystem	28
Deep Dive - Cluster Configuration Overview	30
High Availability Clustering	30
Active/Passive Clustering	31
Active/Active Clustering	32
Clustered Instance Terminology	33
Clustered Configuration Operation	33
OS8.8.X Documentation Specific to ZFS-HA	34
Shape Migration	34
Manual Shape Migration	34
Automated Shape Migration	36
Automatically Expanding a ZFS-HA Storage Pool	39
Set Automatic Expansion of a pool using the BUI	39
Set Automatic Expansion of a pool using the CLI	39
Set Automatic Expansion of a pool using the RESTful API	40
Manually Expanding a ZFS-HA Storage Pool	41
Expanding Existing Block Volumes	41
Creating New Block Volumes	43
Adding New Block Volumes to a Pool	45
Adding Clustered Interfaces	47
Overview	47
1 - Attach New VNICs In the OCI Console	47
2 - Assign Secondary IP Addresses	49

3 - Reboot Both Controllers	49
4 – Run Takeover/Failback	49
5 – Edit New Interfaces	49
About Devices, Datalinks, and Interfaces	50
6 – Reboot and Rebalance	51
Upgrading Your ZFS-HA Instance	52
ZFS-HA System Notes	52
Networking	52
ZFS-HA Network Routing	52
ZFS-HA Network Datalinks	52
ZFS-HA Network Interfaces	52
Block Storage Notes	52
System Boot Disk	52
Storage Pools	52
Boot and Block Volume Backups	53
Backing Up the ZFS Configuration	53
Documentation and Security References	53
Installation Notes	53
Root User Configuration	53
Known Issues	54

PURPOSE STATEMENT

This document provides step-by-step instructions for configuring an Oracle ZFS Storage - High Availability (ZFS-HA) instance in OCI using the Oracle ZFS-HA Storage Deployment Tool.

The Oracle ZFS-HA Storage Deployment Tool will always create a ZFS-HA system using the ZFS High-Availability Marketplace Image. **The use of this image is not free and will incur a cost of \$1.85 per hour per compute instance.** This cost is in addition to the compute shape and block volume storage charges. There is no charge for the use of the Deployment Tool.

The Deployment Tool cannot be used to deploy the Oracle ZFS Storage image, which is limited in the shapes it supports but which has no image use cost.

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.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

INTRODUCTION

Oracle is uniquely positioned to provide products and services that run 24/7 either on-premises or in the cloud and our expertise helps us optimally operate our engineered products in Oracle's cloud infrastructure.

The Oracle ZFS Storage - High Availability (ZFS-HA) Marketplace Image provides cloud-based NAS storage and replication services to enable on-premises ZFS Storage customers to migrate data and apps from on-premises to OCI. Oracle ZFS Storage - High Availability instances provide both protocol services and performance for data migration, replication, and sharing.

Two OCI Compute instances running the ZFS-HA image can be clustered together to create a highly available operating environment providing file and storage services in the event of a single instance node failure. Both active/active and active/passive modes are supported. Each instance can detect that the peer instance is unavailable and take over servicing the peer's data pools.

This document covers in detail how to provision a ZFS-HA cluster in OCI using the ZFS-HA image and the "Oracle ZFS-HA Storage Deployment Tool". This tool is a Terraform stack which automates the process of creating and configuring the ZFS-HA compute instances, the Virtual Network Interface Cards (VNICs), and IP addressing needed to build a full ZFS-HA cluster.

The Oracle ZFS Storage – High Availability image in OCI can be configured as a Bare Metal (BM) or Virtual Machine (VM) instance to support the following use cases:

- Create a DR site in OCI rather than building out a second on-premises facility by replicating data to a ZFS-HA instance in OCI as a replication target from an on-premises ZFS Storage Appliance and reverse the replication back to on-premises as needed
- Share data from a ZFS-HA instance in OCI over NFS, SMB, or cross protocols back to on-premises
- Migrate and host application storage workloads using similar protocols as your on-premises deployments
- Migrate data to OCI over NFSv3, NFSv4, SMB or cross protocols with AD integration using an Oracle ZFS Storage – High Availability instance as a storage gateway

Sharing data and replicating data can be hosted in the following ways:

- Cloud to Cloud
- On-premises to Cloud
- Cloud to on-premises

Review the following summary of supported shapes and recommended number of NFS and SMB clients to determine the best shape for your requirements.

Network Bandwidth Expectations for NFS/SMB Clients

Shape	Max OCPU	Max Memory	Max Network Bandwidth	Max Client Bandwidth	Typical Sustained Bandwidth	Number of Clients
VM.Standard2.8	N/A	120GB	8.2 Gbps	512 MB/s	384 MB/s	Hundred
VM.Standard2.16	N/A	240GB	16.4 Gbps	1025 MB/s	768 MB/s	Few Hundred
VM.Standard2.24	N/A	320GB	24.6 Gbps	1537 MB/s	1150 MB/s	Hundreds
VM.Standard3.Flex	32	512GB	32 Gbps	2000 MB/s	1500 MB/s	Thousand
VM.Standard.E4.Flex	64	1024GB	40 Gbps	2500 MB/s	1875 MB/s	Thousands
VM.Standard.E5.Flex	94	1049GB	40 Gbps	2500 MB/s	1875 MB/s	Thousands
BM.Standard2.52	N/A	768GB	25x2 Gbps	3125 MB/s	2343 MB/s	Thousands
BM.Standard3.64	N/A	1024GB	50x2 Gbps	6250 MB/s	4687 MB/s	Thousands

Notes:

- The Flex shapes listed require a minimum number of OCPUs to have enough VNICs allocated for High Availability clustering.
 - The Deployment tool will use a default of 16 OCPU and 256 GB of memory. This is the recommended minimum for adequate performance and throughput.
 - The minimum configuration for ZFS-HA instances is 8 OCPU and 128GB of memory.
- Typical sustained values in the above chart are based on a workload mix with 50% read / 50% write.
- Number of clients depends on the desired throughput available to each client. If more throughput is needed per client then fewer clients should be used.
- A bare metal (BM) or virtual machine (VM) instance requires only one volume for operation. You can add more volumes to increase storage capacity for your needs.
- Maximum block volume capacity per cluster is 1024TB based on maximum OCI volumes size of 32TB and the OCI limit of 32 volume attachments.
- Detailed shape specifications are available at [OCI Shapes](#).

Overview of Configuration Steps

This guide describes the steps to configure Oracle ZFS Storage as a compute instance in Oracle's Cloud Infrastructure (OCI) using the Oracle ZFS-HA Storage Deployment Tool and contains the following sections:

- Run the Oracle ZFS-HA Storage Deployment Tool from OCI Marketplace
- Configure the Deployment Tool Variables
- Apply the Deployment Tool stack
- Set a Password for ZFS Administration
- Share an SMB Filesystem
- Share an NFS Filesystem

For more information, see the following references:

- [Oracle ZFS Storage Appliance - Release OS8.8.x](#) - General ZFS Storage administration information
- APIs for ZFS Storage in OCI – The “Oracle ZFS Storage - High Availability API Guide” provides additional management APIs used developed specifically for Oracle ZFS Storage - High Availability version.

ZFS-HA TENANCY ARCHITECTURE OVERVIEW

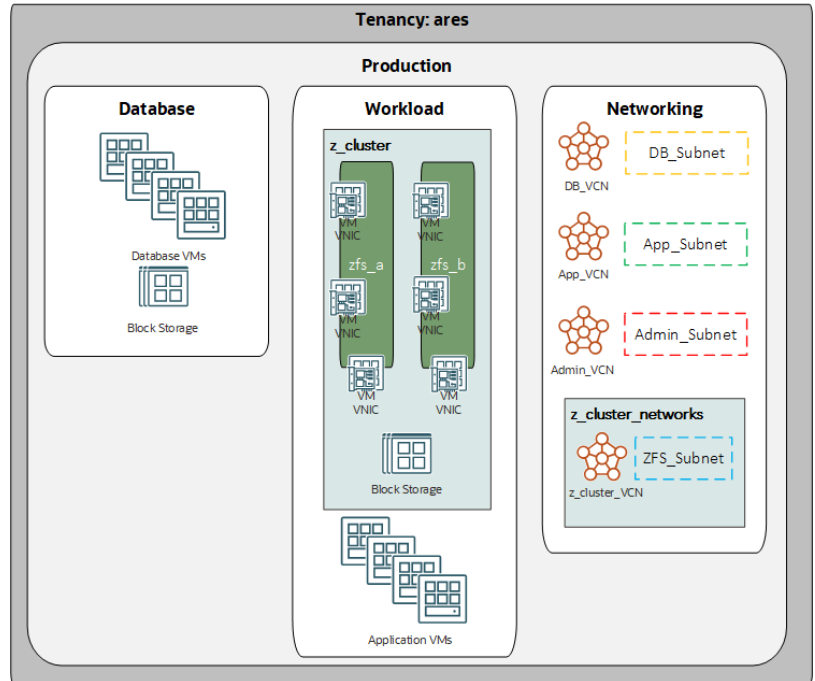
Compartment Overview

The ZFS-HA Deployment Tool creates one or two compute instances and block volumes in a single compartment, referred to in this document as *z_cluster*. A private network must be built to provide I/O to the block volume storage and in a high-availability cluster, I/O between the two controller instances.

In our example tenancy for the Ares corporation, there is a compartment named Production that has three sub-compartments that hold the assets for database VMs, application workload VMs, and Networking.

The ZFS-HA compute and block storage components will be placed in a sub-compartment of Production called *z_cluster*, while the network components will be placed in a sub-compartment of Networking called *z_cluster_networks*.

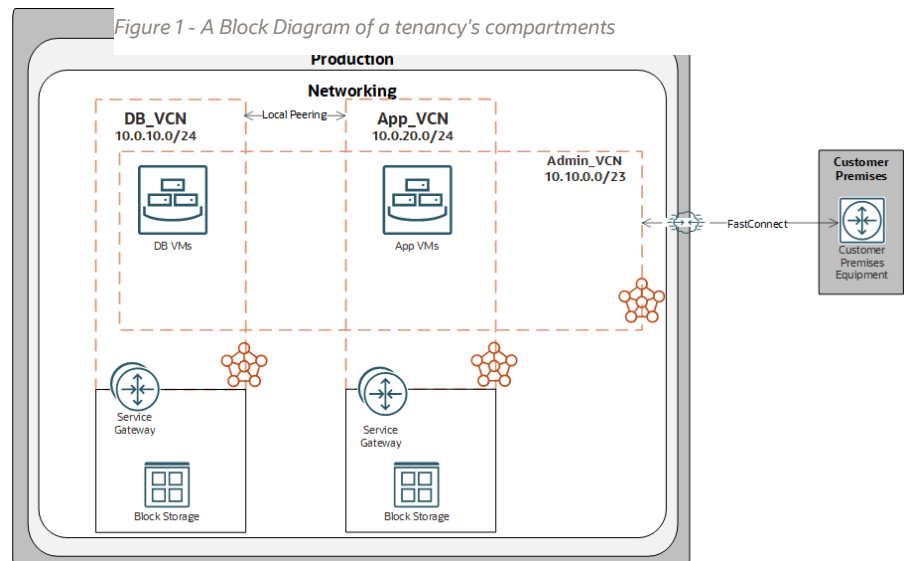
These sub-compartments for ZFS-HA will need to be created before the ZFS-HA deployment tool is run. A VCN must also be created in *z_cluster_networks*. This VCN will be used by the cluster's controller instances to provide the heartbeat communication between the two controllers as well as the I/O to the block volumes. A subnet will be created by the deployment tool for this VCN; do not create it manually.



Network Overview Before ZFS-HA

Figure Two shows the tenancy networking before ZFS-HA is introduced. There are three VCNs, each with an associated subnet.

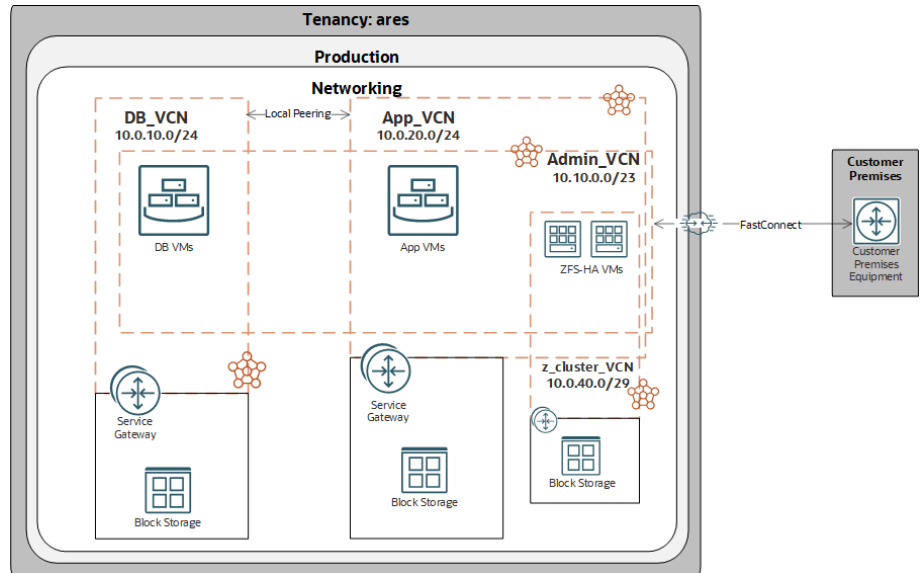
- DB_VCN connects all the database compute instances in the Database compartment for local communication. A Service Gateway provides access to storage resources.
- App_VCN connects all the application compute instances in the Workload compartment for local communication. A Service Gateway provides access to storage resources and a Local Peering Gateway provides communication between App_VCN and DB_VCN.
- Admin_VCN connects a second VNIC on all the database and application compute instances and is connected to the customer's premises using a FastConnect link. This allows the administrators at the customer's location direct access into each of the compute instances.



Network Overview After ZFS-HA

Figure Two shows the tenancy networking after ZFS-HA is installed. There is now a fourth VCN, `z_cluster_VCN`, which is a private VCN that has no external connections other than a Service Gateway for storage I/O.

- `DB_VCN` remains unchanged.
- `Admin_VCN` now includes VNICs on the two ZFS-HA controller instances, which have fixed addresses for direct admin access.
- `App_VCN` now includes two more VNICs from the ZFS-HA instances so that the application instances can mount shares.



OCI Requirements for ZFS-HA Clusters

Many of the requirements for provisioning a ZFS-HA cluster are handled by the Deployment Tool, but some items must be in place before the Stack can be run. Review the following sections to identify requirements for provisioning a ZFS-HA cluster.

ZFS Compute Instance and Network Requirements

- A dynamic group must be created to grant access to resources needed by the ZFS-HA instances to cluster together and connect to various networks within the tenancy.

If the tenancy uses Identity Domains, create this group in the associated identity domain – see [Create a Dynamic Group and Matching Rules](#) for details.

If the tenancy is an older one, it may not use Identity Domains. In this case, create the group in the tenancy as described in [Managing Dynamic Groups](#).

In either case, this dynamic group must be given all access to the compartment that the ZFS-HA compute instances and block volumes will be created in. This will be referred to as the *cluster compartment* and is separate from the cluster networking compartment. The access to the cluster compartment is granted with a rule that points to the cluster compartment's Oracle Cloud Identifier (OCID), such as in the following example. (Note that the example OCID here is much shorter than an actual OCID.)

```
All {instance.compartment.id = 'ocid1.compartment.oc1..aaaa5ldkqq'}
```

Note that OCIDs should always be enclosed in single quotes when used in Dynamic Group or Policy Rule statements.

- Identity Policies must be created to allow the ZFS-HA compute instances in the cluster compartment to manage the resources needed for the cluster processes as well as the network resources needed to connect to VCNs. The rules for these Identity Policies grant access to the dynamic group created in the previous step and thus to the instances in the cluster compartment as well as the Deployment Tool.

The rules may be put in a single policy at the root level, or may be broken into multiple policies, each attached to a compartment which holds a subset of the resources needed. In the example tenancy above, the corporate network

VCNs are in the compartment `Production:Networking`. A policy will be created and attached to that compartment which will hold only the rules that apply to that compartment hierarchy.

If the compartment the rules apply to is not a direct child of the compartment the policy is attached to, then the path down to the compartment referred to must be specified.

There must be a policy attached to the root compartment granting the right to read images within the tenancy:

```
read instance-images in tenancy
```

The policy rules that grant privileges to the cluster compartment are:

```
manage instances in compartment
manage console-histories in compartment
inspect vnic-attachments in compartment
manage volume-attachments in compartment
manage volumes in compartment
```

These rules can be applied to any compartment above the cluster compartment.

Additionally, the following privileges must be given for the compartments that contain each Virtual Cloud Network (VCN) that the instances will use.

```
use vnics in compartment
read private-ips in compartment
- OR -
use private-ips in compartment
```

The compartment for the NAS data client VCN requires a higher level of access to the “private-ips” resource so that it can move IP addresses between the controller instances at a resource takeover or failback event. It must have an identity policy rule to “use private-ips” rather than a rule allowing reads.

These policy rules must be tailored to allow access to the correct compartments and VCNs in your tenancy. Failure to do so will result in issues in creating the cluster and in resource takeover/failback within the cluster. If issues are encountered during deployment, please open a service request with Oracle to have our support team help with troubleshooting the policy rules. See the section [Submitting a Service Request](#) for details on how to route an SR for the fastest response.

- The Deployment Tool automates the attachment of the ZFS-HA’s Virtual Network Interface Cards (VNICs) to VCNs for clustering, administrative, and NAS client connectivity. These VCNs have different requirements, shown here.
 - All VCNs described below and any subnets under them must enable the use of DNS by checking the **“Use DNS hostnames in this VCN/SUBNET”** box. This is the default when creating VCNs or subnets and should not be overridden.
 - Cluster connectivity: The primary VNIC on each ZFS-HA instance is reserved for Block Volume I/O and clustering resources between the two instances. It may also be used for accessing other Oracle services such as ZFS cloud backups to OCI object storage.
 - **There must be an OCI Service Gateway on this VCN for “All <region> Services in Oracle Services Network”.**
 - **The security list for this VCN must have ports 3000 and 3215 open to allow communication between the clustered instances.**
 - **The default Route Table for this VCN must have a rule with the target type of “Service Gateway”, the destination service of “All <region> Services...”, and the target service gateway the name of the service gateway in this VCN.**
 - **Do not create a subnet in this network, but instead identify a subnet range for the tool to create.** The Deployment Tool will create its own subnet in this VCN. This subnet must be a **/28 or larger subnet**.
 - This VCN should **not** allow access to storage administrators or NAS clients. Sharing cluster and block volume traffic on the same VCN as the administrator or data traffic can have a negative effect on the overall performance of the ZFS-HA system.

- NAS client connectivity: This VCN must have appropriate gateways, firewall settings, and routing for the client networks. Any OCI compute instance or on-premises computer that accesses data in an ZFS-HA share is considered a client. A subnet must be created under this VCN before the Deployment Tool is run.
- Administrator access: This VCN must have appropriate gateways, firewall settings, and routing for administrative access to the ZFS-HA controller instances, including CLI, REST, and BUI access. It is recommended that the Admin interface be on a separate VCN for security, but it is not a requirement. A subnet must be created under this VCN before the Deployment Tool is run.
- Replication (optional): A feature of Oracle ZFS Storage Appliances, whether in OCI or on-premises, is the ability to replicate ZFS snapshots, which capture a share or project's data at the specific point in time that the snap is taken. These snapshots can be replicated to other appliances and cloned to be mountable shares on those appliances. While most installations will use the NAS client VCN for replication, it may be advantageous to designate a separate VCN for this traffic to keep it off the VCNs used for client and administrative traffic.

If this is desired, a pair of clustered interfaces can be created and assigned to a new VCN (see [Adding Clustered Interfaces](#) later in this guide). This VCN must have appropriate gateways and routing to reach the other appliances involved in ZFS replication. Note that replication network configuration is not a part of the Deployment Tool. Additional Identity Policy rules may be needed for VNIC and private-ip access if additional VCNs or compartments are used.

Example - OCI Configuration for a ZFS-HA Cluster

Based on our example tenancy for the Ares Corp., the Policies and their rules can be created and attached as follows:

- In the Default Identity Domain (or at the root level for older tenancies not using Identity Domains), the dynamic group *prod_zfs.dg* is created:

```
All{instance.compartment.id = <OCID of z_cluster compartment>
```

- A policy is created at the root level with a single rule:

```
allow dynamic-group prod_zfs.dg to read instance-images in tenancy
```

- A policy is created and is attached to the Workload compartment. It contains the following rules:

```
allow dynamic-group prod_zfs.dg to manage instances in compartment Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage console-histories in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to inspect vnic-attachments in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage volume-attachments in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage volumes in compartment Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage volume-backups in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage backup-policies in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage backup-policy-assignments in compartment \
  Workload:z_cluster
```

- A policy is created and attached to the Networking compartment. It contains the following rules:

```
allow dynamic-group prod_zfs.dg to use private-ips in compartment Networking
allow dynamic-group prod_zfs.dg to use vnics in compartment Networking
allow dynamic-group prod_zfs.dg to read private-ips in compartment \
  Networking:z_cluster_networks
```

A Closer Look At Policy Rules

It's worth taking a closer look at the Identity Policies to break down the rules being used and what they apply to. The rule in the policy attached to the root compartment is the only one global to the entire tenancy. Its purpose is to allow the Deployment Tool to read the ZFS-HA image so that it can be deployed:

```
allow dynamic-group prod_zfs.dg to read instance-images in tenancy
```

The next set of rules are in a policy attached to the `Workload` compartment and give the Deployment Tool permission to create and use block volumes and compute instances in the cluster compartment `Workload:z_cluster`.

```
allow dynamic-group prod_zfs.dg to manage instances in compartment Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage console-histories in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to inspect vnic-attachments in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage volume-attachments in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage volumes in compartment Workload:z_cluster
```

These rules allow the ZFS-HA instances to automatically grow their storage pools as described in the [Automatically Expanding a ZFS-HA Storage Pool](#) section later in this document.

```
allow dynamic-group prod_zfs.dg to manage volume-backups in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage backup-policies in compartment \
  Workload:z_cluster
allow dynamic-group prod_zfs.dg to manage backup-policy-assignments in compartment \
  Workload:z_cluster
```

The final set of rules define the VNIC and IP address privileges to the ZFS-HA networking resources in the various compartments. Each compartment a VNIC is attached to requires two rules. The first grants access to “use vnics”. If a pair of VNICs will have an IP address that floats between the two controllers, the compartment they are in must have a second rule to “use private-ips” in that compartment. For compartments with VNICs that will have fixed addresses, such as the cluster and admin interfaces, the “read private-ips” rule is sufficient.

The VNICs attached to `Admin_VNC` will not have floating IP addresses, but the ones attached to `App_VCN` will. They are both in the `Networking` compartment, so that compartment will need the greater access provided by the “use” verb rather than the “read” verb.

```
allow dynamic-group prod_zfs.dg to use private-ips in compartment Networking
allow dynamic-group prod_zfs.dg to use vnics in compartment Networking
```

Because rules are inherited by a compartment’s children, we only need a single rule to grant the permissions required for the ZFS-HA controllers to reduce the access for `private_ips` in the `z_cluster_networks` compartment of `Networking:z_cluster_networks`:

```
allow dynamic-group prod_zfs.dg to read private-ips in compartment \
  Networking:z_cluster_networks
```

If the `z_cluster_networks` compartment was not in the `Networking` hierarchy, rules would be needed to both “read private-ips” and “use vnics” for the `z_cluster_networks` compartment.

Note that the `Networking` compartment has been given the slightly higher privilege to “use private-ips” so that we can move IP addresses on `App_VCN`, but we have also granted that access for `Admin_VCN`, where it is not needed. In keeping with the principles of least access, the administrator at Ares Corp. might choose to create a new sub-compartment for `App_VCN` so that it can be granted higher access without inadvertently also granting it to `Admin_VCN` and change the rules to match the compartment configuration.

FIRST STEPS

If your organization does not have an Oracle Cloud Infrastructure (OCI) account already, one can be set up at <https://www.oracle.com/cloud/>. Note that the Oracle ZFS Storage – High Availability image is not available as part of the Oracle Cloud Free Tier.

This guide assumes that usable compartments, virtual cloud networks (VCN), and subnets have already been created within the OCI tenancy. An administrator for your OCI tenancy will authorize resources in a specified compartment for you to use.

The following information will be needed to configure the OCI compute instance:

1. OCI Compartment IDs
2. VCN Compartments and Names
3. Subnet Compartments and Names

The Installation Checklist on the next page will help in organizing this information so that it is easily available when it is time to run the Deployment Tool. It is recommended that that page be printed out and filled out as you work through each of the requirements before running the ZRFS-HA Deployment Tool.

You will also need an SSH client to do the initial configuration and know how to configure the SSH client to use ssh key authentication. An SSH key pair must be generated before stating the Stack configuration process.

INSTALLATION CHECKLIST

ZFS-HA Configuration and Placement

OCI Region (Select from OCI console before getting Stack)	
Stack Name	
Cluster Type	Active/Active or Active/Passive
Bare Metal or Virtual Machine	
Shape	
Storage (appliance host) Name	
Compute and Block Storage Compartment	
Availability Domain	
Fault Domain 1	
Fault Domain 2	
SSH Key File Location	

Cluster Network Configuration

Compartment of Cluster Network VCN	
Cluster Network VCN	
Cluster Network subnet CIDR block	

Admin Network Configuration

Compartment of NAS Admin Network VCN	
Subnet in NAS Admin Network	
IP Address for Admin Access VNIC on Primary	*
IP Address for Unused Admin Access VNIC on Primary	*
IP Address for Admin Access VNIC on Secondary	*
IP Address for Unused Admin Access VNIC on Secondary	*

* IP Addresses will be assigned from chosen Subnet range if not defined

NAS Data Network Configuration

Compartment of NAS Data Network VCN	
Subnet in NAS Data Network	
IP Address for Data Access VNIC on Primary	*
IP Address for Pool-a Data IO on Primary (used by the clients)	*
IP Address for Unused Data Access VNIC on Primary	*
IP Address for Data Access VNIC on Secondary	*
IP Address for Pool-b Data IO on Secondary	*
IP Address for Unused Data Access VNIC on Secondary	*

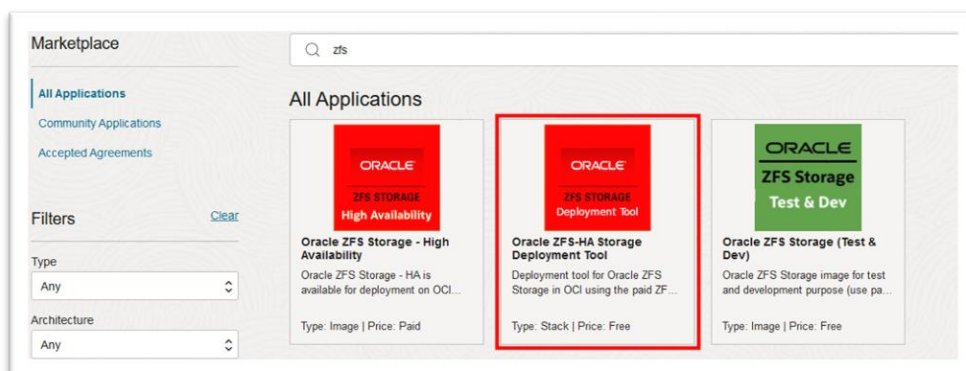
* IP Addresses will be assigned from chosen Subnet range if not defined

Data Volume Configuration

Number of Block Volumes for Primary Storage Pool	(32 Max across all pools)
Number of Block Volumes for Secondary Storage Pool	
Block Volume Size (50GB - 32768GB)	

RUN THE ZFS STORAGE DEPLOYMENT TOOL FROM OCI MARKETPLACE

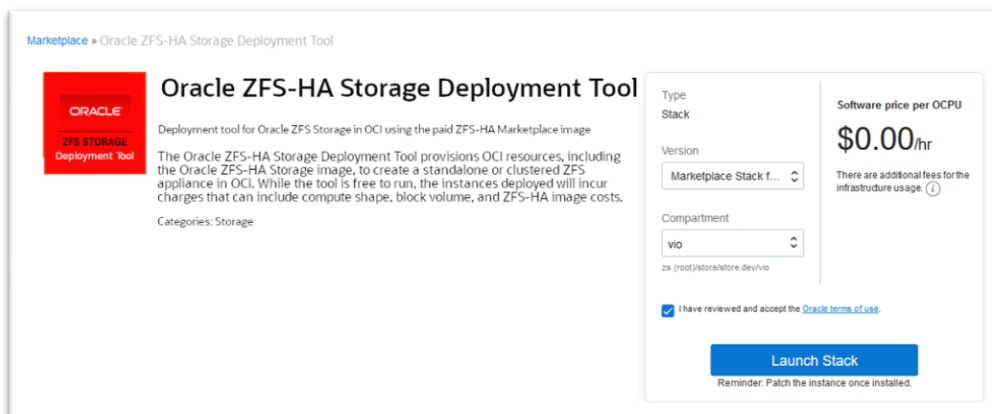
1. Log in to your OCI tenancy and go to the Marketplace and search All Applications for ZFS Storage images. Select the Oracle ZFS-HA Storage Deployment Tool.



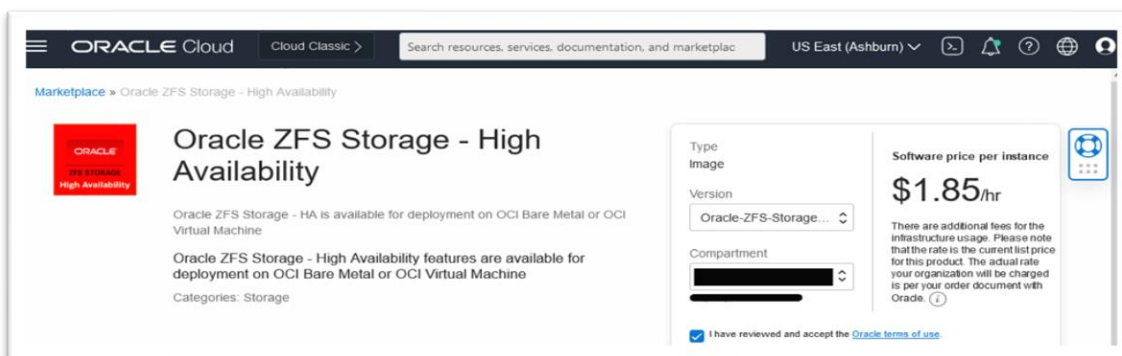
2. Choose the version from the pulldown menu. Except for special cases, the latest version is recommended and is the default choice. As of OS8.8.59, a unified image for both Bare Metal (BM) and Virtual Machine (VM) shapes is provided. In some limited cases a previously released image may be desired, so these images are shown in the pulldown menu. Ensure that the correct type and version of the image is selected for your use case.

Select the appropriate compartment for your tenancy to run the Compute instances in.

Read the Overview and review and accept the terms and conditions, then click “Launch Stack”.



Note that while the Deployment Tool is free to use, it will deploy a pair of instances using the Oracle ZFS Storage – High Availability image. Each ZFS-HA instance has an hourly charge in addition to the Compute shape and Block Storage charges incurred, as shown below.



CONFIGURE THE DEPLOYMENT TOOL VARIABLES

- Enter a name for the Stack and optionally add a description or tags. Neither the Compartment name nor Terraform version can be changed on this screen. Click Next.

Create Stack

1 Stack Information 2 Configure Variables 3 Review

Your application will launch as part of a stack that includes the infrastructure resources required to ensure that the application deploys and runs properly.

Name *Optional*
JH Stack

Description *Optional*

Create in compartment
vio
zs (root)/store/store.dev/vio

Terraform version
1.1.x
0.11.x is no longer supported. [What Terraform versions are supported by Resource Manager?](#)

Tags
Optional tags to organize and track resources in your tenancy. [How do I use tags?](#)

Tag Namespace Tag Key Tag Value
None (add a free-form tag)

Next Cancel

- Configure the variables in the “Storage Configuration and Placement” section.
 - Use the pulldown menu to select the type of cluster desired. An Active/Active cluster will create two storage pools, while an Active/Passive cluster will create only one. Choose SingleHead when high availability is not desired and a single controller is sufficient. Some variables listed here will not appear if SingleHead or an Active/Passive cluster is chosen.
 - Choose the Compute Instance Shape from the pulldown menu. If a Flex shape is chosen, enter the number of OCPUs and the Memory Size in GBs. The recommended shape (E5.Flex) and configuration (8 OCPU/128 GB) is a good starting point for its performance and throughput. When using a Flex shape, the OCPU and memory can be changed even after the ZFS-HA cluster is up and running.

Storage Configuration

Active/Passive

Select a type for storage configuration. HA Solution: Active/Passive or Active/Active, Non-HA Solution: SingleHead.

Compute Instance Shape

VM.Standard.E4.Flex

Compute instance shape to use for ZS OCI instances. Select a shape supported by the image. VM.Standard2.4 shape supports only SingleHead configuration.

Number of OCPUs *Optional*
16
For Active/Active configuration, at least 5 OCPUs is required, 8 OCPUs is recommended.

Memory Size (GBs) *Optional*
256
At least 80 GBs is required, 128 GBs is recommended.

NOTES:

- For Flex shapes, note that the total bandwidth that will be available to the instance is tied to the OCPU count with each OCPU adding 1Gbps to the overall available bandwidth until the maximum bandwidth for the shape is reached as shown in the chart [Network Bandwidth Expectations for NFS/SMB Clients](#).

- An Active/Active cluster must have a minimum of five (5) OCPUs and 80GB of memory; Active/Passive clusters must have a minimum of four (4) OCPUs and 64GB of memory. An Active/Active cluster will not run on a VM.Standard2.4 shape.
 - When an Active/Passive cluster is deployed, the B controller will be the active controller after the cluster is deployed. A Failback action can be taken to move the services back to the A controller.
- In Storage Name, choose a name for the cluster. The two Compute instances will use this name plus “-a” or “-b” appended to it as the hostnames for the instance. As an example, if ‘zfsha’ is entered here, the two Compute instances will be named ‘zfsha-a’ and ‘zfsha-b’.
- In Compartment, choose the compartment in which the Compute instances and block volumes will be placed.
- In Availability Domain, choose the AD in your region to place the compute instances and block volumes. Running the cluster instances in separate Availability Domains is not supported.

Storage Name
jh-zfs
Base hostname for ZS OCI instances and their resources. For cluster, a is appended for primary, b for secondary. Use alphanumeric characters and hyphen("-") only. Cannot end with hyphen.

Compartment
vio
Compartment where to place the storage.

Availability Domain
iZbs:PHX-AD-3
Availability domain where to place the storage.

Fault Domain for Primary
FAULT-DOMAIN-1
Fault domain to place the primary instance.

Fault Domain for Secondary
FAULT-DOMAIN-2
Fault domain to place the secondary instance.

- Choose the Fault Domains in which to run each instance using the pulldown menus. The instances must run in separate Fault Domains.

More on regions and availability domains may be found at <https://docs.oracle.com/en-us/iaas/Content/General/Concepts/regions.htm>.

- Configure the variables in the “Cluster Networking” section.
 1. In the Cluster Networking Configuration section, use the pulldown menus to choose the Compartment and Subnet to be used for the network the iSCSI and VIO clustering traffic will run on.
 2. Enter an unused CIDR block for the Cluster network. A subnet will be created for this block, and the IP addresses used by the Primary VNICS on each Compute Instance will be assigned within this block.
Do not use a CIDR block that has already been assigned to a subnet in this VCN.

Cluster Networking Configuration

Compartment of Cluster Network VCN
store.dev
Compartment where Cluster Network VCN is configured.

Cluster Network VCN (Non-NAS Network)
ZSOCI
Cluster Network VCN (Non-NAS Network) where to create a subnet for cluster network.

Subnet CIDR Block for Cluster Network
10.0.66.0/24
CIDR Block to configure the subnet for cluster network (e.g., 10.0.101.0/24).

- Configure the Variables in the Networking Configurations.
 1. From the pulldown menus, choose the Compartment and Subnet for the Admin network, which is used to access the browser (BUI) and command line (CLI) interfaces of the ZFS-HA instances.

Networking Configuration

☐ Admin Network custom configuration
Check this box to allow users to configure NAS Admin Network access VNICs. Uncheck this box for auto assignment, and to configure from the NAS admin VCN compartment and subnets(This is a default option and recommended).

Compartment of NAS Admin Network VCN

Compartment where NAS Admin Network VCN is configured.

Subnet in NAS Admin Network ⓘ

Subnet where to configure admin access VNICs.

You may optionally enter IP addresses from within the chosen subnet's range for the VNICs that will be created by checking the box for Admin Network Custom Configuration. Any VNICs that do not have an IP address assigned here will have IP addresses automatically assigned from the chosen subnet's range.

Networking Configuration

☒ Admin Network custom configuration
Check this box to allow users to configure NAS Admin Network access VNICs. Uncheck this box for auto assignment, and to configure from the NAS admin VCN compartment and subnets(This is a default option and recommended).

Compartment of NAS Admin Network VCN

Compartment where NAS Admin Network VCN is configured.

Subnet in NAS Admin Network ⓘ

Subnet where to configure admin access VNICs.

IP Address for Admin Access VNIC on Primary *Optional*

Private IP address to assign for admin access VNIC on the primary instance (e.g., 10.0.1.11). Leave blank for auto assignment.

IP Address for Unused Admin Access VNIC on Primary *Optional*

Private IP address to assign for unused admin access VNIC on the primary instance (e.g., 10.0.1.12). Leave blank for auto assignment.

IP Address for Admin Access VNIC on Secondary *Optional*

Private IP address to assign for admin access on the secondary instance (e.g., 10.0.1.13). Leave blank for auto assignment.

IP Address for Unused Admin Access VNIC on Secondary *Optional*

Private IP address to assign for unused admin access on the secondary instance (e.g., 10.0.1.14). Leave blank for auto assignment.

2. From the pulldown menus, choose the Compartment and Subnet for the NAS Data network, which is used by clients to mount shares.

☐ Data Network custom configuration
Check this box to allow users to configure NAS Data Network access VNICs. Uncheck this box for auto assignment, and to configure from the NAS data VCN compartment and subnets(This is a default option and recommended).

Compartment of NAS Data Network VCN

Compartment where NAS Data Network VCN is configured.

Subnet in NAS Data Network ⓘ

Subnet where to configure data access VNICs and IP address for NAS data IO.

As with the Admin network, you may optionally enter IP addresses from within the chosen subnet's range for the VNICS that will be created by checking the box for Data Network Custom Configuration. Any VNICS that do not have an IP address assigned here will have IP addresses automatically assigned from the chosen subnet's range. Note that the "NAS Data IO" address is the one that will be used by the clients and may be moved between the cluster instances depending on which instance is active for the pool the address is associated with.

☒ **Data Network custom configuration**
 Check this box to allow users to configure NAS Data Network access VNICS. Uncheck this box for auto assignment, and to configure from the NAS data VCN compartment and subnets(This is a default option and recommended).

Compartment of NAS Data Network VCN
 Choose...
 Compartment where NAS Data Network VCN is configured.

Subnet in NAS Data Network ⓘ
 There was an error retrieving options.
 Subnet where to configure data access VNICS and IP address for NAS data IO.

IP Address for Data Access VNIC on Primary *Optional*
 Private IP address to assign for data access VNIC on the primary instance (e.g., 10.0.3.11). Leave blank for auto assignment.

IP Address for NAS Data IO on Primary *Optional*
 IP address for NAS data traffic on the primary instance (e.g., 10.0.3.12). Leave blank for auto assignment.

IP Address for Data Access VNIC on Secondary *Optional*
 Private IP address to assign for data access on the secondary instance (e.g., 10.0.3.14). Leave blank for auto assignment.

The screen shots above were taken while setting up a cluster using the default VM.Standard.E4.Flex shape and active/passive cluster configuration. These options will vary when different shapes or cluster types are chosen.

- In the Data Volume Configuration, enter the number and size of the Block Volumes used for each storage pool. In an Active/Active cluster, the two pools do not need to be the same size. Note that there is a limit of 32 volumes across both pools and each pool must have a minimum of two volumes. Only one pool will be created for an Active/Passive pool.

Data Volume Configuration

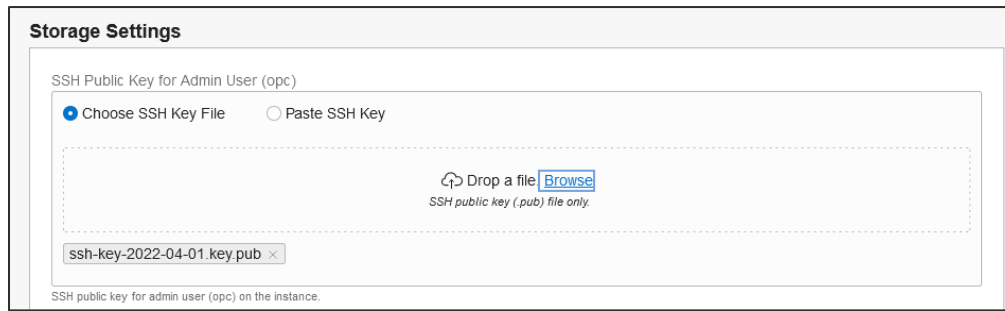
Number of Block Volumes for Primary Storage Pool
 2
 Number of block volumes (data disks) to create the primary storage pool. Max 32 block volumes in total per storage.

Number of Block Volumes for Secondary Storage Pool
 2
 Number of block volumes (data disks) to create the secondary storage pool. The secondary storage pool is created only for Active/Active cluster configuration.

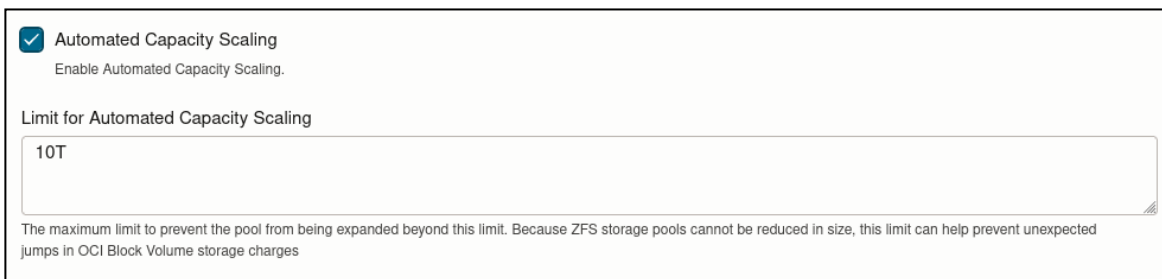
Block Volume Size (GBs)
 50
 Size of each block volume in GBs: 50GB - 32768GB(32TB).

NOTE: The size and number of block volumes affects performance. OCI Block Volumes improve in performance as they grow, and tops out at 1TB. In addition, ZFS storage pool performance increases as the number of block volumes is increased, topping out at around ten volumes. Keep this in mind when determining the number and size of the block volumes used to create a storage pool.

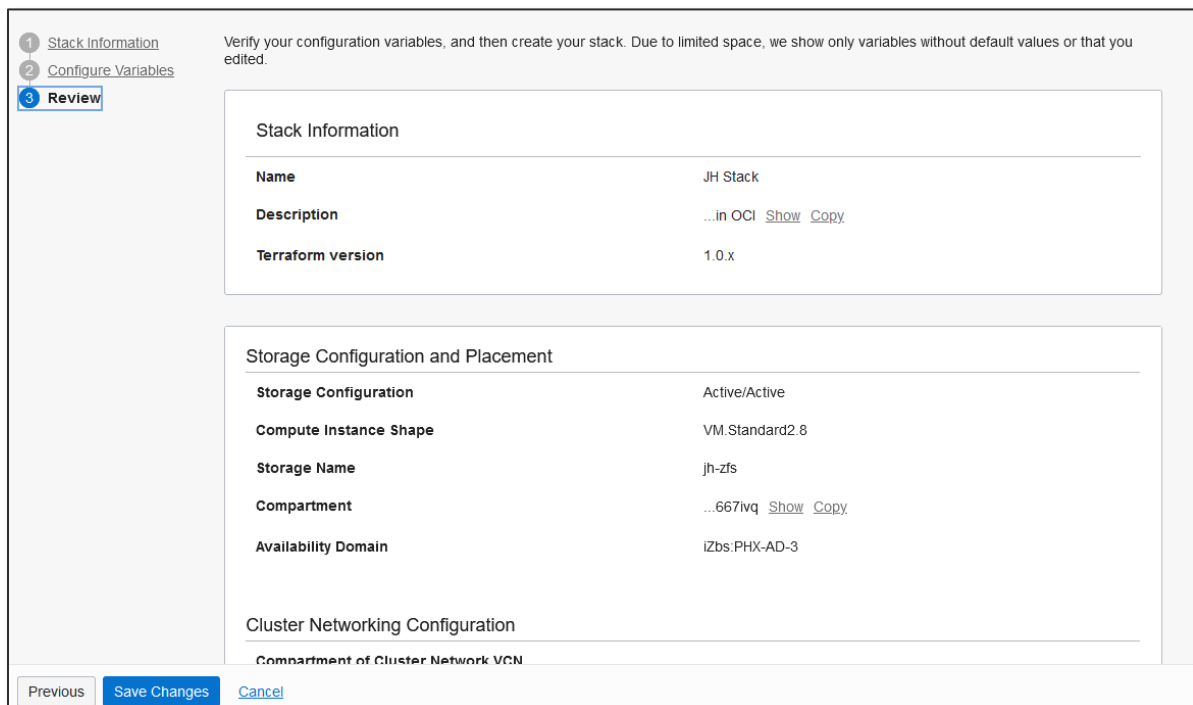
- In the storage section, choose an SSH public key file or paste an SSH public key. There is no need to modify the User Init Data fields. Click Next when complete.



- If pool automated capacity scaling is desired, check the box for `autoscale_option`. This option will automatically expand a pool by existing block volumes and/or add block volumes when a pool is found to be 80% full. Space will be expanded until the pool becomes 60% full. Optionally, a cap may be placed on the automated expansion that prevents growing the pool beyond this size. See the section [Automatically expanding a ZFS-HA Storage Pool](#) later in this document for more detail.

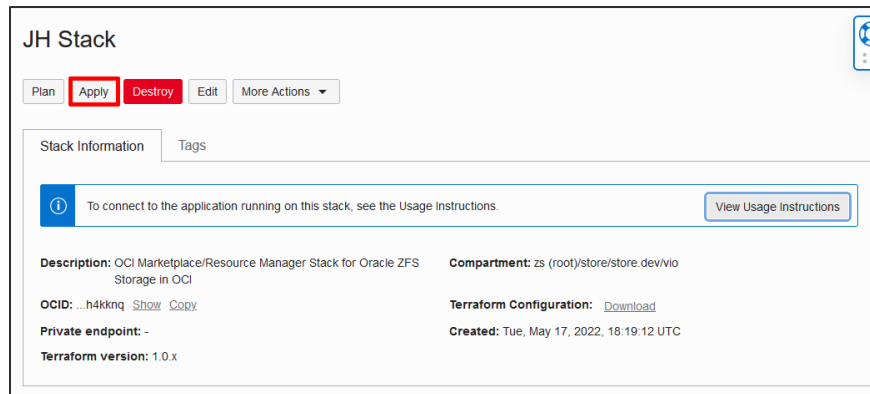


Verify the values that have been entered for the variables. Click Previous if changes need to be made. When complete, click Save changes.



APPLY THE STACK

To begin the process of creating the stack, click the Apply button. If you wish to review the Terraform execution plan before applying the stack, click the Plan button and review the log from that action.



It will take approximately five minutes for the stack to build the ZFS-HA cluster and may take another 5 minutes to complete the booting process the initial clustering configuration.

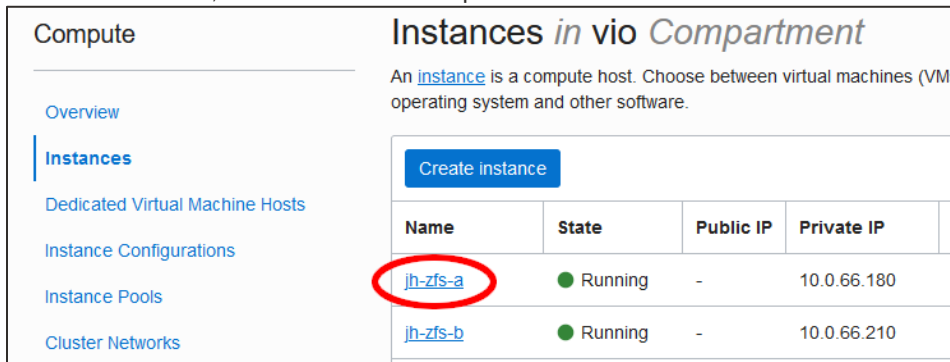
It is important to remember that after the stack has shown as having succeeded some additional time will be needed before the instances can be connected to in order to complete the remaining steps.

SET A PASSWORD FOR ZFS ADMINISTRATION

Once the Stack has been applied and completed successfully, a password must be set to allow administrators to manage the storage on the cluster via either the BUI or CLI. Connect to the primary ZFS-HA instance by using SSH to connect to the Admin VNIC on the primary ZFS-HA Compute instance.

This address may have been configured as the “IP Address for Admin Access VNIC on Primary” variable in the stack. If the address was automatically assigned, it may be found with the following steps:

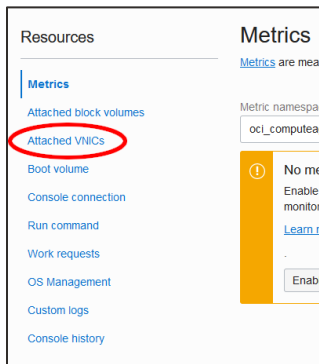
- List the instances in the compartment and select the instance with the name given in the variable configuration with “-a” at the end, as shown in this example:



The screenshot shows the OCI Compute console. On the left, the 'Instances' tab is selected. On the right, a table lists instances in the compartment. The instance 'jh-zfs-a' is circled in red.

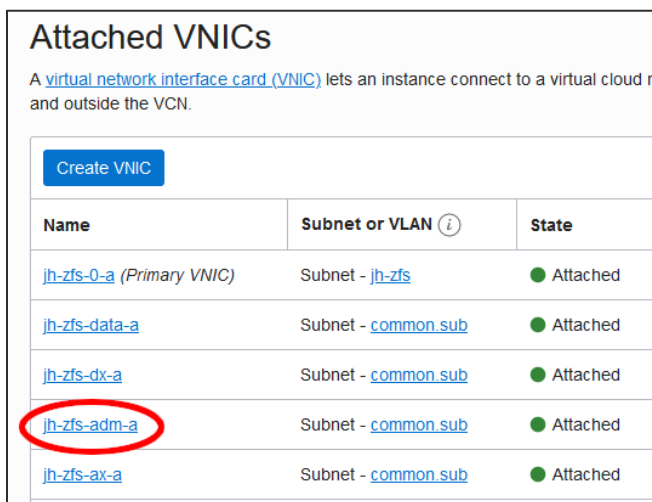
Name	State	Public IP	Private IP
jh-zfs-a	Running	-	10.0.66.180
jh-zfs-b	Running	-	10.0.66.210

- On the Instance details screen, scroll down until the Resources menu is shown on the left side of the screen and choose “Attached VNICS”.



The screenshot shows the OCI Instance details page. On the left, the 'Resources' menu is expanded, and 'Attached VNICS' is highlighted with a red circle.

- In the list of Attached VNICS, find the VNIC that ends with “-adm-a” and select it.



The screenshot shows the OCI Attached VNICS page. A table lists the attached VNICS. The VNIC 'jh-zfs-adm-a' is circled in red.

Name	Subnet or VLAN ⓘ	State
jh-zfs-0-a (Primary VNIC)	Subnet - jh-zfs	Attached
jh-zfs-data-a	Subnet - common.sub	Attached
jh-zfs-dx-a	Subnet - common.sub	Attached
jh-zfs-adm-a	Subnet - common.sub	Attached
jh-zfs-ax-a	Subnet - common.sub	Attached

- Find the VNIC's Private IP address and copy it.



- Using your tool of choice, SSH to the address copied in the above step. Use the private part of the SSH key applied in the Stack variable configuration process and use "opc" as the user.

The instance includes the opc user by default. The opc account has all authorizations enabled and can be used to configure the storage appliance. If root user access is needed, see https://support.oracle.com/knowledge/Sun%20Microsystems/2811414_1.html.

You can transition to a full administrative-capability root account once you have logged in as the opc user if you need full administrative access to the instance.

Run the commands as shown in this example, using your own password where the asterisks are shown:

```
ssh -i .ssh/opc opc@100.104.21.251
```

```
jh-zfs-a:> configuration users
jh-zfs-a:configuration users> select opc
jh-zfs-a:configuration users opc> set initial_password
Enter new initial_password: *****
Re-enter new initial_password: *****
Initial_password - (set) (uncommitted)
jh-zfs-a:configuration users opc> commit
jh-zfs-a:configuration users> exit
```

If you wish to give users the ability to SSH into the cluster controllers using a password rather than with SSH keys, the following commands can be run to enable that before the session is exited:

```
jh-zfs-a:> configuration services ssh
jh-zfs-a:configuration services ssh> set password_authentication=true
password_authentication=true (uncommitted)
jh-zfs-a:configuration services ssh> commit
```

CONNECT TO THE BROWSER USER INTERFACE (BUI)

Log in to the BUI by connecting your browser to https://<primary_admin_address>:215 and using “opc” as the username with the password entered in the previous section. Note that because a self-signed certificate is used for HTTPS encryption, your browser may identify the site as a security risk. You may safely continue to the site.

The BUI may now be used to create shares or perform other ZFS Appliance administration tasks.

Distribute Shared Resources to Their Default Controllers

When the Terraform stack that created the ZFS-HA cluster completed, resources such as the storage pools and VNICs may not be associated with their default controller. On an Active/Active cluster, the secondary ZFS-HA instance will have control of all shared resources. On an Active/Passive cluster, the B node will control all the shared resources.

To assign the resources to the controllers where they belong, perform a Failback action from the BUI of the secondary ZFS-HA instance (node B) by navigating to the Configuration->Network screen and clicking the Failback button.

This example shows an Active/Active cluster that has just been deployed. All shared resources are on the B node.

ConfigurationMaintenanceSharesStatusAnalytics

SERVICESSTORAGENETWORKSANCLUSTERUSERSPREFERENCESSETTINGSALERTS

SETUPUNCONF

FAILBACK

KEOVERREVERTAPPLY

jh-zfs-b

Active (takeover completed)

jh-zfs-a

Ready (waiting for failback)

Active Resources

RESOURCE	OWNER
jh-zfs-a (net/vtionet1) 100.102.221.96	jh-zfs-a
jh-zfs-b (net/vtionet2) 100.102.208.147	jh-zfs-b
jh-zfs-adm-b (net/vtionet4) 100.102.222.73	jh-zfs-b
zfs/pool-a 97.9G	jh-zfs-a
zfs/pool-b 97.9G	jh-zfs-b

Active Resources

No resources are active on this cluster node.

After the Failback has completed, all resources are on the appropriate instances, as shown here.

jh-zfs-b

Active

jh-zfs-a

Active

Active Resources

RESOURCE	OWNER
jh-zfs-b (net/vtionet2) 100.102.208.147	jh-zfs-b
jh-zfs-adm-b (net/vtionet4) 100.102.222.73	jh-zfs-b
zfs/pool-b 97.9G	jh-zfs-b

Active Resources

RESOURCE	OWNER
jh-zfs-a (net/vtionet1) 100.102.221.96	jh-zfs-a
zfs/pool-a	jh-zfs-a

ZFS-HA Networking Recommendations

The following are recommendations that refer to settings within the ZFS-HA images. The Deployment Tool may have configured some or all of these, but it is recommended that they be verified after deployment.

ZFS-HA Network -> Configuration -> Routing

- It is recommended to set the multihoming model to strict.

ZFS-HA Network -> Configuration -> Datalinks


- Link Speed, Link Duplex and Flow Control should all be set to Auto.
- The link speed for VM instances will be reported as 1GB but will actually use the full amount of bandwidth allocated to the instance. (See known issues)
- All network datalinks have their MTU set to 9000 (jumbo frames) by default for best performance. In some cases, especially when data will traverse Wide Area Network links such as when replicating to or from an on-premises ZFS Storage Appliance, this may cause connections to hang if any step of the path is not set to use jumbo frames. In these cases, an MTU of 1500 is recommended.

ZFS-HA Network -> Configuration->Interfaces

- The primary network interface used for iSCSI traffic should not be modified because it can cause a system panic. (See known issues)
- NAS client interfaces should uncheck 'Allow Administration' for enhanced security.

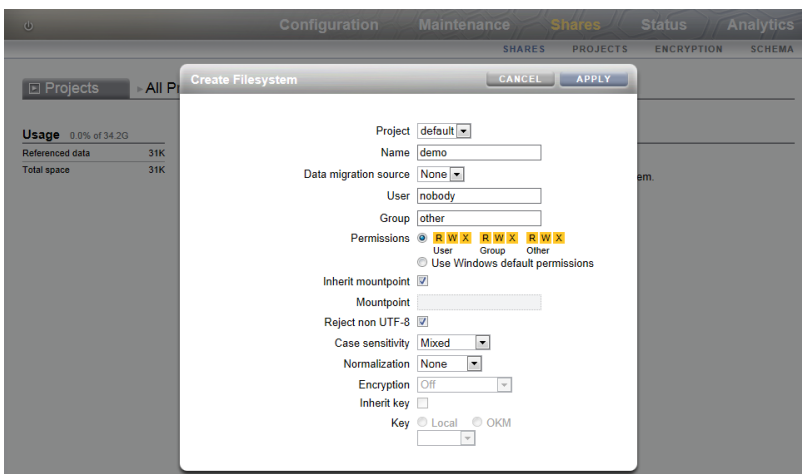
SHARE AN SMB FILESYSTEM


Complete the following steps to set up a simple filesystem share over Server Message Block (SMB) with Windows user access. Begin by logging in to the BUI by connecting your browser to https://<primary_admin_address>:215 and using “opc” as the username with the password entered in the section “Set a Password for ZFS Administration”.

1. Navigate to the Shares screen.
Click the add item icon  next to Filesystems to create a new filesystem.



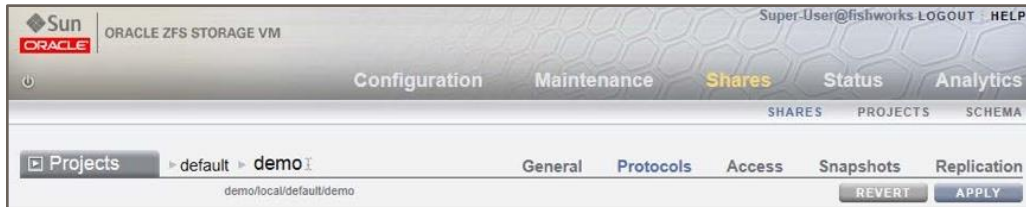
2. Name the filesystem and change the permissions for Group and Other to allow anyone to read, write, and execute on the filesystem. In this example, the filesystem is named demo. The filesystem is part of the default project. Click APPLY to save the changes.



3. In the Shares screen, mouse over the entry for the new filesystem and click the edit icon  to edit the filesystem attributes.

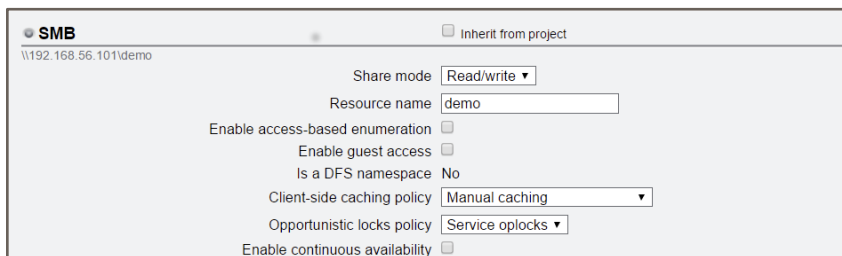


- Click Protocols.



- In the SMB section, clear the checkbox for Inherit from project, select Read/Write Shareable in the Share mode drop-down list, and set the Resource Name.

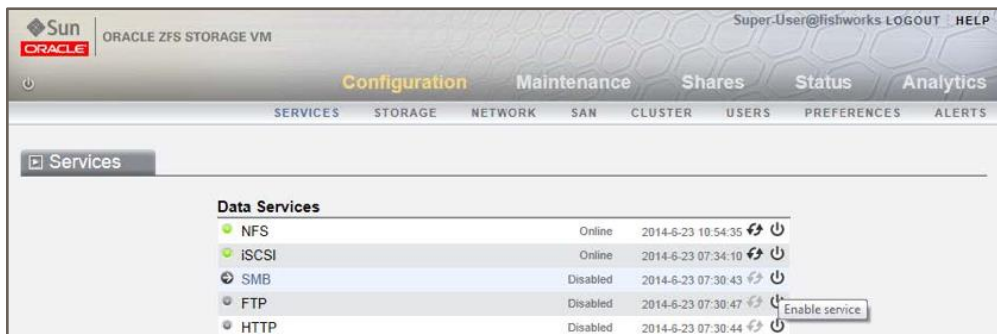
In this example, the Resource Name is demo. Click APPLY to save the changes.



Note: In the SMB section of the Protocol screen, the status of the SMB service is shown with either a green, amber, or grey circle. Beneath that is a possible path to mount the share at, but the IP address shown will be the address of the interface the browser connected to. This is almost certainly incorrect, since the best practice would be to connect the browser through the administrative interface, not the client data interface. You should verify that the correct address is used when accessing the share from a data client.




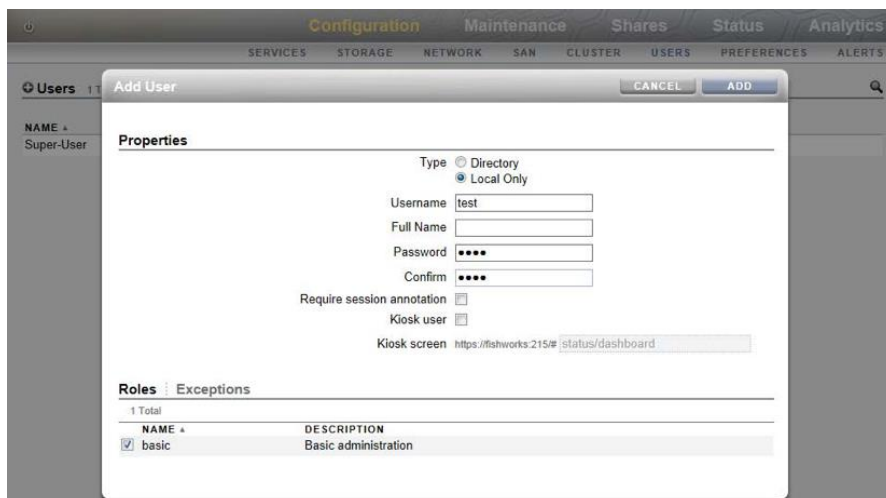
- Select the Configuration tab to access the Configuration Services screen.
- Enable the SMB service by clicking the power icon.



The state will change from Disabled to Online.

8. Configure a user with access to the filesystem share.

- a. Click USERS in the navigation bar, and click the add item icon  next to Users to create a new user.
- b. Select Local Only, set the Username and Password, and click ADD. Log out of the BUI by clicking LOGOUT near the top of the screen.



The screenshot shows the 'Add User' dialog box in the Oracle ZFS Storage BUI. The dialog has a 'Properties' tab and an 'ADD' button. The 'Type' is set to 'Local Only'. The 'Username' field contains 'test'. The 'Full Name' field is empty. The 'Password' and 'Confirm' fields are masked with dots. The 'Require session annotation' checkbox is unchecked. The 'Kiosk user' checkbox is checked. The 'Kiosk screen' field contains the URL 'https://fishworks.215#status/dashboard'. Below the properties, there is a 'Roles' section showing a table with one role: 'basic' with the description 'Basic administration'.

NAME	DESCRIPTION
basic	Basic administration


9. From a Windows client, connect to the NAS Data IO of your ZFS Storage instance, and log in with the credentials you set in step 8 to access the shared filesystem.

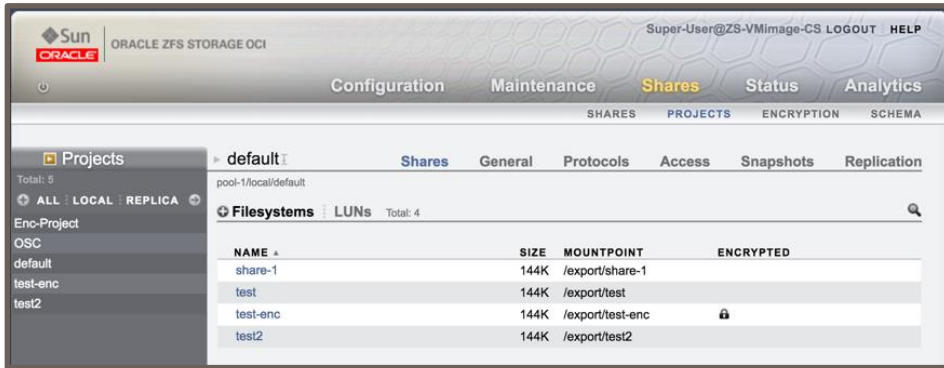
Note: When looking at the Protocol screen of a share in the BUI, the mount point given will use the IP address (or FQDN if it resolves in DNS) of the administrative interface. This is not correct – the NAS client address or FQDN should be substituted.

SHARE AN NFS FILESYSTEM

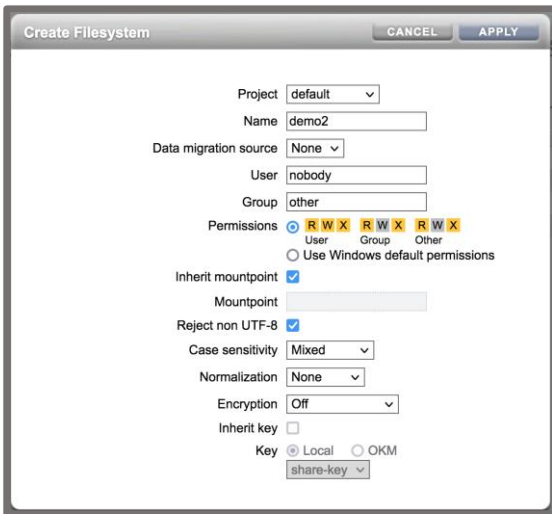
Complete the following steps to set up a simple filesystem share over NFS to share with an NFS client or clients. Begin by logging in to the BUI by connecting your browser to https://<primary_admin_address>:215 and using “opc” as the username with the password entered in the section “Set a Password for ZFS Administration”.


1. Navigate to the Shares screen.

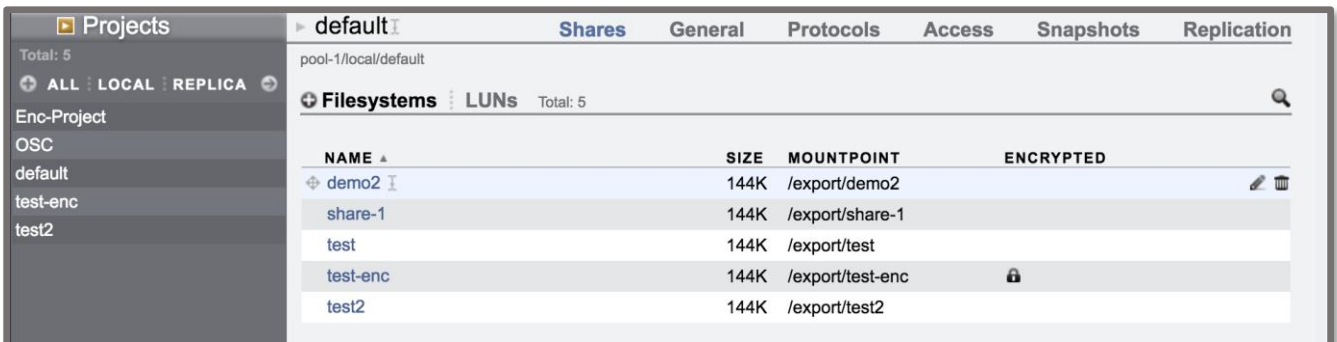
Click the add item icon  next to Filesystems to create a new filesystem. Projects provide an administrative point for filesystems so you can set properties at the project level that are inherited by filesystems within the project. The system includes the default project.



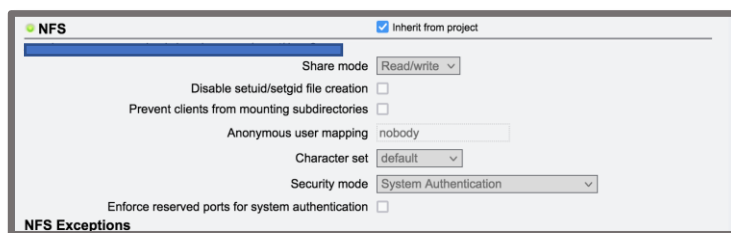
2. Name the filesystem and change the permissions to match the user/group requirements. In this example, the filesystem is named demo2. The filesystem is part of the default project. Click APPLY to save the changes.



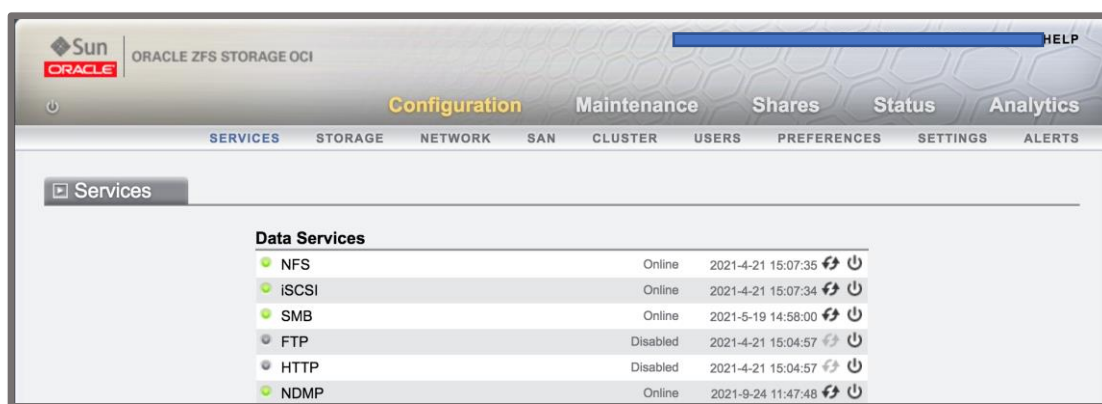
3. In the Shares screen, mouse over the entry for the new filesystem and click the edit icon  to edit the filesystem attributes.



- Click Protocols. In the NFS section, set the Share mode to Read/write in the pulldown menu, if it is not inherited from the project. Click APPLY.



- Select the Configuration tab to access the Configuration Services screen.
- Enable the NFS service by clicking the power icon if it is not already enabled.



- Mount the filesystem over NFS with syntax similar to the following:

```
% mount -t nfs <NAS_Data_IO_address>:/export/demo2 /mnt
```

Use the IP address assigned to the NAS Data IO IP address on the primary ZFS instance.

Note: When looking at the Protocol screen of a share in the BUI, the mount point given will use the IP address (or FQDN if it resolves in DNS) of the administrative interface. This is not correct – the NAS client address or FQDN should be substituted.

DEEP DIVE - CLUSTER CONFIGURATION OVERVIEW

This section gives an overview of how two OCI ZFS-HA instances are connected, either in an Active/Active or Active/Passive configuration. When configured as Active/Passive, one instance is active providing data services and one instance is passive, performing no data operations but available for operation if the active instance becomes unavailable.

Active/Passive configuration behavior:

- The primary data pool or pools are configured and running on the active instance.
- If the active instance fails, the primary data pool(s) are exported and imported on the passive instance and NAS IP addresses are migrated.
- The passive instance becomes the active instance until the active instance is recovered.

Active/Active configuration behavior:

- A minimum of two data pools are required. The total storage for both pools is equal to the number of storage volumes that can be attached to one compute instance. Each node is owner of its own pool(s) and services NAS clients via an IP address tied to those pool(s).
- If the either instance fails, the peer data pool(s) are exported and imported on the working instance and NAS IP addresses are migrated.
- The working instance will now serve both pools from both nodes and may operate in a degraded state since it now must serve those pools with only one node instead of two nodes.

Takeover behavior:

- Estimated failover time between instances is 70-90 seconds
- Orchestration software transitions the following components when takeover occurs back to the active instance:
 - Secondary IP addresses
 - Public IP addresses
 - Storage volumes

High Availability Clustering

A virtual cluster link (VIO) is used to cluster two ZFS Storage High Availability instances. The Primary VNICs on each instance are used for the link over which the cluster heartbeats occur. The cluster quorum is determined by OCI compute instance metadata properties.

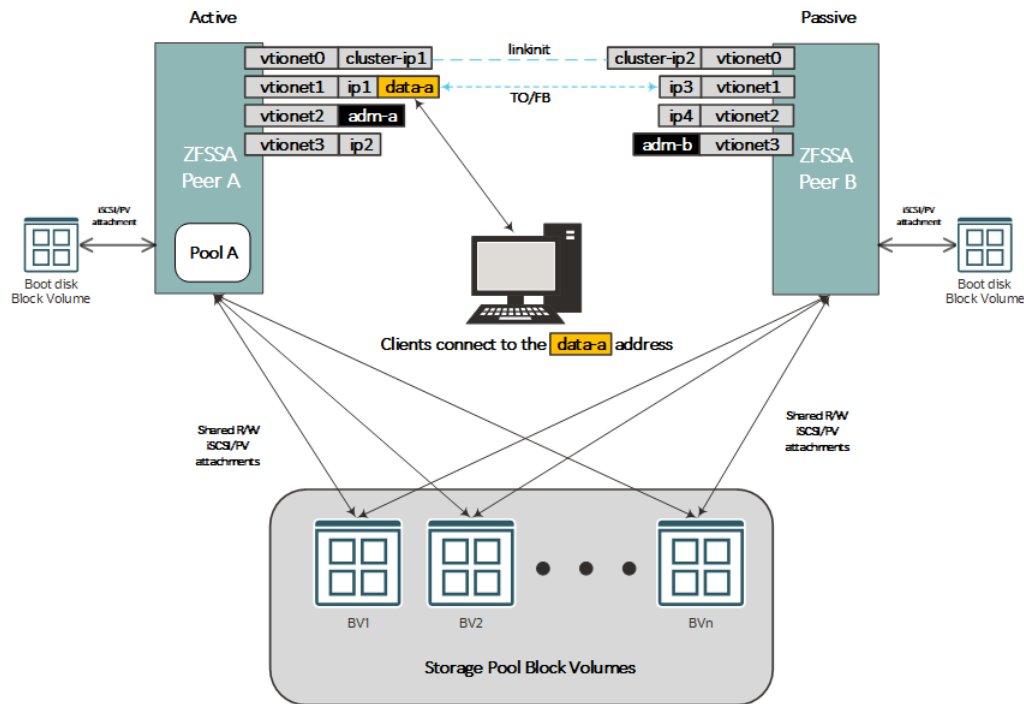
NOTE: The Primary VNIC (Virtual Network Interface Card) is the first network interface in a Compute instance. Additional VNICs may be added and are referred to as Secondary VNICs. Each VNIC is assigned an IP address at its creation and is referred to as the primary IP address. Additional IP address may be assigned to it. These are referred to as the VNIC's secondary IP addresses. Care should be taken not to confuse secondary VNICs with secondary IP addresses.

In a cluster, the following applies to the VNICs:

- The Primary VNIC is used for the VIO link as well as storage volume I/O and OCI API calls. This VNIC is often setup on a private subnet with no access to NAS clients or storage administrators but check with your tenancy administrator for the proper IP subnets and addresses to use.
- Secondary VNICs are configured by the stack to supply access to NAS clients and storage administrators.
- Configuration changes are synchronized across instances.

Active/Passive Clustering

In an Active/Passive cluster, all resources are controlled by a single ZFS-HA compute instance, the Active controller. If the Active controller suffers a failure or an administrator performs a Takeover function, the Passive controller takes over the shared resources such as the storage pool and the nas-ip address.



In an Active/Passive configuration with a single pool, four VNICs are required on each ZFS-HA instance as shown in the table below.

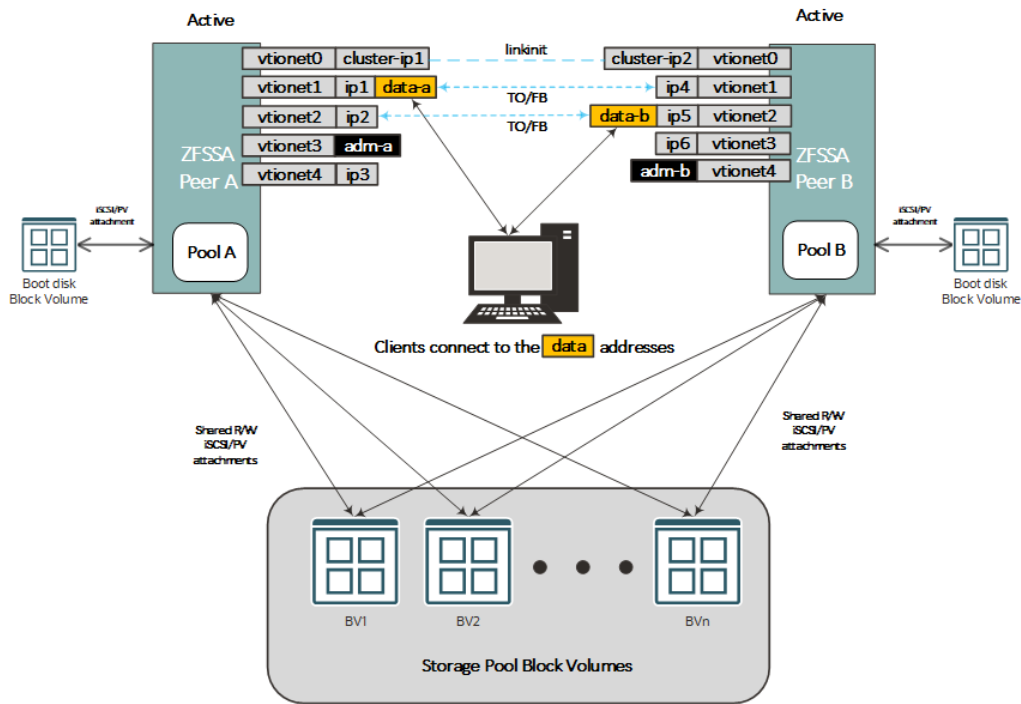
VNIC	ZFS INSTANCE A (ACTIVE)	ZFS INSTANCE B (PASSIVE)	USAGE
vtionet0 (Primary VNIC)	cluster-ip1	cluster-ip2	Used for cluster I/O only
vtionet1 - primary IP	ip1 - Placeholder	ip3 - Placeholder	Primary IP addresses are locked to the instance and cannot move.
vtionet1 - secondary IP	data-a address	unassigned	Floating IP address used for client access to Pool-A. Always assigned to the active controller
vtionet2 – primary IP	adm-a address	ip4 - Placeholder	Private Administrative access to Node A (B unused)
vtionet3 – primary IP	ip2 - Placeholder	adm-b address	Private Administrative access to Node B (A unused)

In this example, a secondary IP address, data-a, is assigned to vtionet1 on Node A, the active controller. In the event that Node B is made active, the data-a address will automatically move to vtionet1 on Node B. Clients attached to the data-a address will have a brief interruption but will continue to be connected to the storage pool when the takeover by Node B is complete.

While only one IP address is used on each controller to connect for management purposes, two VNICs are created for cluster management reasons. One on each controller will always remain unused.

Active/Active Clustering

In an Active/Active cluster, resources are shared across both ZFS-HA compute instances. If either controller suffers a failure or an administrator performs a Takeover function, all shared resources such as the storage pools and the data addresses are moved to the remaining Active controller.



In an Active/Active configuration with two pools, five VNICs are required on each ZFS-HA instance as shown in the table below.

VNIC	ZFS INSTANCE A	ZFS INSTANCE B	USAGE
vtinet0 (Primary VNIC)	cluster-ip1	cluster-ip2	Used for cluster I/O only
vtinet1 - primary IP	ip1 - Placeholder	ip4 - Placeholder	Primary IP addresses are locked to the instance and cannot move.
vtinet1 - secondary IP	data-a address	unassigned	Floating IP address used for client access for Pool-A.
vtinet2 - primary IP	ip2 - Placeholder	ip5 - Placeholder	Primary IP addresses are locked to the instance and cannot move.
vtinet2 - secondary IP	unassigned	data-b address	Floating IP address used for client access for Pool-B.
vtinet3 - primary IP	mgmt-ip1	ip6 - Placeholder	Private Administrative access to Node A (B unused)
vtinet4 - primary IP	ip3 - Placeholder	mgmt-ip2	Private Administrative access to Node B (A unused)

In this example, secondary IP addresses, data-a and data-b, are assigned to vtionet1 on Node A and vtionet2 on Node B. If either node becomes inactive, the data-a and data-b addresses will be assigned to vtionet1 and vtionet2 on the same controller, respectively. The remaining active instance will also control both storage pools. Clients attached to the data-a address will have a brief interruption but will continue to be connected to the storage pool when the takeover by Node B is complete.

Clustered Instance Terminology

A resource is a physical or virtual object that is present and possibly active on one or both cluster heads. Resources are managed by storage administrators who can set which instance owns the resource when clustered.

Term	Description
Resource Type	
Singleton	Known by both instances but only active on one instance. (Storage Pools and NAS IP)
Private	Only available and active on one instance. (Administration Network Interface)
Replicate	Resource known by both heads. (Service configuration)
Symbiote	Follows other resources (Replications actions follow storage pool)
Clustered State	
Unconfigured	Clustering is not configured.
Owner	Clustering is configured. This active instance owns the storage and data resources.
Stripped	Clustering is configured. This passive instance does not control any shared resources.
Clustered	Clustering is configured in an active/active configuration.

Clustered Configuration Operation

- OCI API commands are issued from each clustered ZFS instance to manage OCI compute, storage, and network resources.
- OCI principal authentication is used to issue OCI API commands.
- All ZFS cluster resources must be in the same OCI availability domain and the same dynamic group.
- All storage volumes will be mounted as shareable on both ZFS instances.
- Network interfaces configured as singletons must use secondary IP addresses so they can be migrated.

OS8.8.X DOCUMENTATION SPECIFIC TO ZFS-HA

This section contains information that applies only to ZFS-HA instances. This information cannot be found in the online documentation for the Oracle ZFS Storage Appliance.

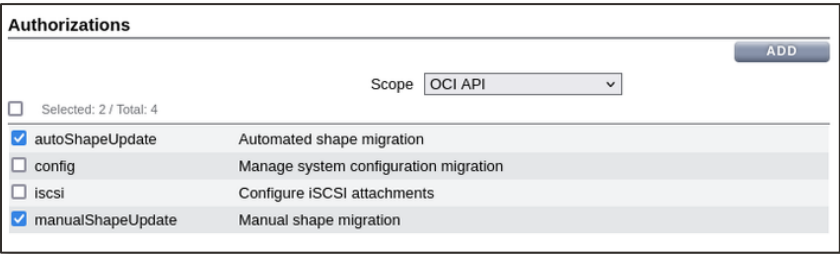
Shape Migration

This feature, added in OS8.8.75, adds the ability for ZFS-HA instances to automatically adjust available CPU, bandwidth, and/or memory on an underlying Flex compute shape or automatically migrate to a different VM.Standard fixed shape according to usage. This has the potential benefits of saving on the supporting infrastructure costs and ensuring consistent performance.

The ability to manually migrate to a different shape is also supported, ensuring a smooth transition when moving a ZFS-HA cluster to a different shape, such as moving from the fixed VM.Standard2.8 shape to VM.Standard.E4.Flex. (Bare Metal shapes are not supported by this feature.)

Note that editing an instance's shape requires a reboot of the instance. On a ZFS-HA cluster, each controller instance will be upgraded and rebooted sequentially to ensure client I/O is maintained. In the case where the underlying shape is being changed, the instance OCID is preserved and all VNICS and Block Volumes attachments to the existing instance will be transparently migrated to the new instance.

In order to perform any shape migration actions, a role associated with the administrative user must have the OCI API authorizations for `autoShapeUpdate` and/or `manualShapeUpdate` set, as shown here in a BUI screenshot.



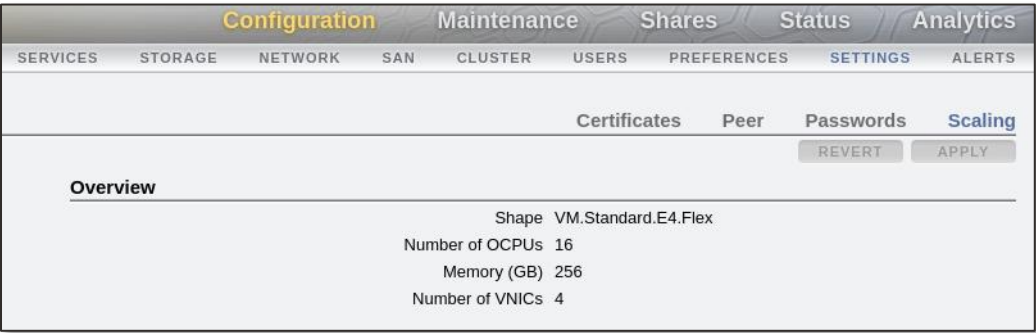
These authorizations may also be set via the CLI or REST interfaces. Refer to the “[Configuring Users](#)” section of the ZFSSA Administration Guide or the “[RESTful API User Service](#)” section of the RESTful API Guide for details.

Manual Shape Migration

The ZFS-HA instances can have their shape changed manually to increase or decrease the performance of the shape as needed. This is triggered through the ZFS Alerts feature - see the [Configuring Alerts](#) section of the ZFSSA documentation for details on this feature. Ensure that the user enabling this feature has a role which has the `manualShapeUpdate` authorization in the OCI scope before continuing.

BUI

The Shape Migration feature is accessed in the BUI at Configuration->Settings->Scaling. The Scaling screen displays the current shape and configuration, as shown here.



This ZFS-HA cluster is running on the E4.Flex shape with 16 OCPU and 256 GB of memory. The screenshot below reflects the desired change – continuing to use the E4.Flex shape but going to 8 OCPU, down from 16. A default value of 16 GB for the Memory is overwritten with the new amount desired – in this case, the same amount as currently used, 256 GB.

Note that when manually changing shapes, moving between Standard and Flex shapes is supported, but not to Bare Metal shapes.

Clicking the Update button will begin the process and reboot both instances sequentially. Once the process has completed, verify that the new shape configuration is being used and that all clustered resources are available on the expected controllers.

CLI

Commands to manually perform a shape migration are accessed from “configuration settings scaling manual”. The commands below walk through the steps needed to change the number of OCPUs used by the instances in our cluster. (Note that the `vnics` property is displayed here but cannot be changed. See the “[Adding Clustered Interfaces](#)” section of this document to learn how to add interfaces to a ZFS-HA cluster.)

```
jh-zfs-a:> configuration settings scaling manual
jh-zfs-a:configuration settings scaling manual> ls
Properties:
                shape = VM.Standard.E4.Flex
                ocpus = 8
                memory = 256G
                vnics = 4
jh-zfs-a:configuration settings scaling manual> set ocpus=6
                ocpus = 6 (uncommitted)
jh-zfs-a:configuration settings scaling manual> commit
Performing a shape update will reboot the appliance. Are you sure? (Y/N)
```

Entering “Y” at this point will begin the process and reboot both heads sequentially. Once the process has completed, verify that the new shape configuration is being used and that all clustered resources are available on the expected controllers.

RESTful API

The current shape may be queried and modified via the RESTful API. Here is the Curl command to retrieve the current manual properties:

```
$ curl -k -u user:pass https://<admin_IP_Address>:215/api/setting/v2/scaling/manual
{
  "Manual scaling settings":
  {
    "href": "/api/setting/v2/scaling/manual",
    "shape": "VM.Standard.E4.Flex",
    "ocpus": 5,
    "memory": 85899345920
  }
}
```

The `shape`, `ocpus`, and/or `memory` settings may be changed. Here is a Curl command to change the current shape's CPU count to 8:

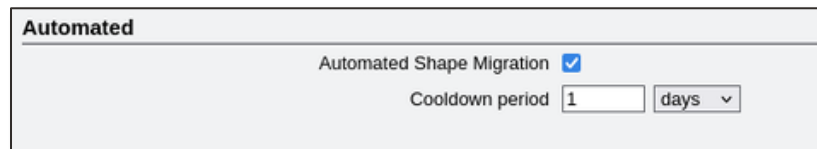
```
$ curl -k -u <user>:<pass> -H "Content-Type: application/json" -d '{"ocpus": 8}' -XPUT \
"https://<admin_IP_address>:215/api/setting/v2/scaling/manual?confirm=true"
```

Automated Shape Migration

The ZFS-HA instances can be set to automatically increase or decrease the performance of the shape as needed. Shape migration is triggered through the ZFS Alerts feature - see the [Configuring Alerts](#) section of the ZFSSA documentation for details on this feature. Ensure that the user enabling this feature has a role which has the `autoShapeUpdate` authorization in the OCI scope before continuing.

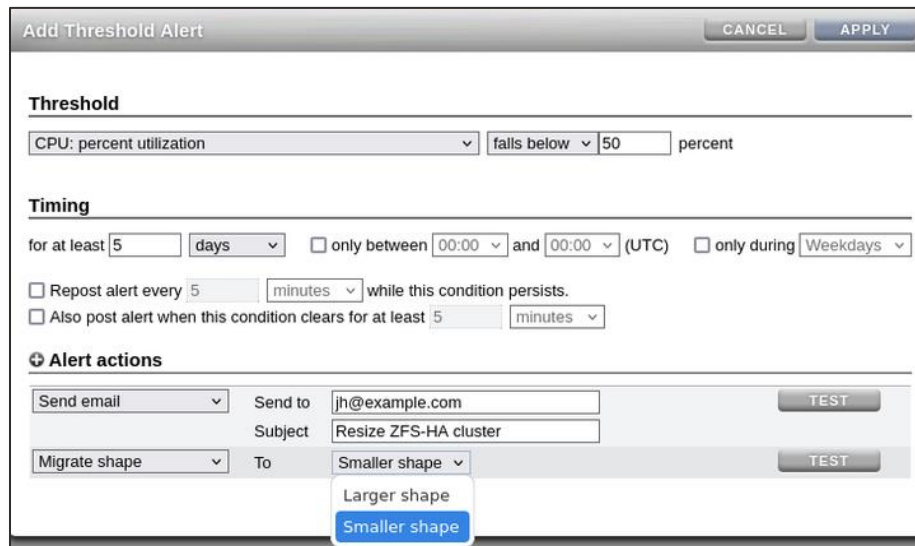
BUI

To prepare for automated scaling, navigate to the Configuration->Settings-Scaling window. Enable automated shape migration by checking the box and set a cooldown period, which sets the minimum amount of time after an automated scaling action before another automated scaling action can be taken. In the example shown below, a cooldown period of one day is selected. If a system is migrated at 9PM Tuesday evening, the shape cannot be automatically migrated again until 9PM Wednesday evening even if an alert would normally resize the shape.



The screenshot shows a window titled "Automated" with a sub-header "Automated Shape Migration" and a checked checkbox. Below it, the "Cooldown period" is set to "1" with a "days" dropdown menu.

In the screenshot below, a threshold alert has been created that will perform two actions if it finds that the system's CPU utilization has stayed below 50% for five days straight. It will send an email with the subject "Resize ZFS-HA Cluster" and it will trigger the Migrate Shape action, in this case, moving to a smaller shape.



The screenshot shows the "Add Threshold Alert" window. The "Threshold" section is set to "CPU: percent utilization" falling below 50 percent. The "Timing" section is set to "for at least 5 days". The "Alert actions" section has two actions: "Send email" with subject "Resize ZFS-HA cluster" and "Migrate shape" to "Smaller shape".

An alert could also be defined to migrate to a larger shape if the CPU was busy for a pre-determined amount of time.

Any of the alert types, either triggered by an action or a threshold, can be used to trigger a shape migration, but the most useful tend to be the threshold alerts for "CPU: percent utilization" and "Network: interface bytes per second".

The network threshold can be important because the shape's overall network bandwidth is tied to the OCPU count at the rate of 1Gbps per OCPU. A shape also has a limit on the number of VNICs it can support, which is the number of OCPU allocated to the shape. When a shape migration requests a smaller shape (with fewer OCPU), if the new OCPU count would be smaller than the number of VNIC attachments, an alert will be generated and the resize will not occur.

Flex shape migrations will retain the same shape but will increase or decrease the number of OCPU by 20%. Memory will also be changed accordingly.

An automated static shape change will use the next available static shape. For example, instances running on VM.Standard2.8 that have a migration to a larger shape triggered will be moved to a VM.Standard2.16 shape. Note that

automated migration between Flex and Standard shapes is not supported but can be done manually. Note that Bare Metal shapes are not supported by the migration feature.

When triggered automatically on a clustered system, the controller instances will both be migrated sequentially to the new shape in order to minimize any effect on clients that have mounted shares. Once the process has completed, verify that the new shape configuration is being used and that all clustered resources are available on the expected controllers.

CLI

Commands to prepare for automated shape migration are accessed from “configuration settings scaling automated”. The commands below walk through getting the current configuration, enabling the feature, and setting a cooldown period of 6 hours. Note that when setting the value for the cooldown period, you must enter a numerical value followed by the time unit. Valid time units here are “seconds”, “minutes”, “hours”, “days”, and “years”. The singular form of these units may also be used.

```
jh-zfs-a:> configuration settings scaling automated
jh-zfs-a:configuration settings scaling automated> ls
Properties:
    shape_update = false
    cooldown = 1 day
jh-zfs-a:configuration settings scaling automated> set shape_update=true
    shape_update = true (uncommitted)
jh-zfs-a:configuration settings scaling automated> set cooldown=6hours
    cooldown = 6 hours (uncommitted)
jh-zfs-a:configuration settings scaling automated> commit
jh-zfs-a:configuration settings scaling automated>
```

Once automated scaling has been enabled, one or more alerts triggering a scaling action must be created. For this example, the ZFS-HA cluster should scale down if the CPU usage remains under 50% for two days. Start by creating the alert as shown below. See the [Configuring Alerts](#) section of the ZFSSA documentation for more details on available statistics and creating alerts in general. Note that by setting `frequency=0` we disable reposting the alert even if the criteria are met, and by setting `minclear=0` we disable sending an “all clear” post.

```
jh-zfs-a:> configuration alerts thresholds
jh-zfs-a:configuration alerts thresholds> create
jh-zfs-a:configuration alerts threshold (uncommitted)> set statname=cpu.utilization
    statname = cpu.utilization (uncommitted)
jh-zfs-a:configuration alerts threshold (uncommitted)> set type=inverted
    type = inverted (uncommitted)
jh-zfs-a:configuration alerts threshold (uncommitted)> set limit=50
    limit = 50 (uncommitted)
jh-zfs-a:configuration alerts threshold (uncommitted)> set minpost=2days
    minpost = 2 days (uncommitted)
jh-zfs-a:configuration alerts threshold (uncommitted)> set frequency=0
    frequency = (uncommitted)
jh-zfs-a:configuration alerts threshold (uncommitted)> set minclear=0
    minclear = (uncommitted)
jh-zfs-a:configuration alerts threshold (uncommitted)> commit
Created watch e8aab8d7-445a-41b2-a446-c1c74df7c0bf
```

When the alert is created, it returns the GUID of the alert. Make note of this, as we need it for the next step, in which the action to be taken is defined. Start by creating an action, and set its category to be “thresholds” and set “thresholdid” to the GUID of the alert created in the last step.

```

jh-zfs-a:configuration alerts thresholds> cd ..
jh-zfs-a:configuration alerts> actions
jh-zfs-a:configuration alerts actions> create
jh-zfs-a:configuration alerts actions (uncommitted)> set category=thresholds
                        category = thresholds
jh-zfs-a:configuration alerts actions (uncommitted)> set thresholdid=e8aab8d7-445a-41b2-
a446-c1c74df7c0bf
                        thresholdid = e8aab8d7-445a-41b2-a446-c1c74df7c0bf (uncommitted)
jh-zfs-a:configuration alerts actions (uncommitted)> commit

```

The last step is to define the handler action this alert will trigger. List the actions; the last one listed should be the action we just created. Select that action, then set its handler as “migrate_shape” and its “scaling” to down:

```

jh-zfs-a:configuration alerts actions> ls
Actions:

ACTIONS      CATEGORY      ACTION      HANDLER
actions-000  thresholds    action-000  syslog
actions-001  thresholds    action-000  syslog
actions-002  thresholds    -           -
jh-zfs-a:configuration alerts actions> select actions-002
jh-zfs-a:configuration alerts actions-002> action
jh-zfs-a:configuration alerts actions-002 action (uncommitted)> set handler=migrate_shape
                        handler = migrate_shape
jh-zfs-a:configuration alerts actions-002 action (uncommitted)> set scaling=down
                        scaling = down (uncommitted)
jh-zfs-a:configuration alerts actions-002 action (uncommitted)> commit

```

With the alert and its action in place, as well as having enabled automated scaling, this cluster will now scale its OCPU count down by 20% when the CPU has under 50% usage for two days continuously.

RESTful API

A Curl command to get the current settings for automated shape migration is given here. Note that the cooldown period is always shown in seconds.

```

$ curl -k -u <user>:<pass> https://<admin_IP_address>:215/api/setting/v2/scaling/automated
{
  "Automated scaling settings": {
    "href": "/api/setting/v2/scaling/automated",
    "shape_update": true,
    "cooldown": 43200
  }
}

```

A Curl command to disable automated shape scaling and to set the cooldown period to one day. A successful operation will return the new current settings:

```

$ curl -k -u <user>:<pass> -H "Content-Type: application/json" -d '{"shape_update": false, \
"cooldown": 86400}' -XPUT https://<admin_IP_address>:215/api/setting/v2/scaling/automated
{
  "Automated scaling settings": {
    "href": "/api/setting/v2/scaling/automated",
    "shape_update": false,
    "cooldown": 86400
  }
}

```

Automatically Expanding a ZFS-HA Storage Pool

A pair of ZFS-HA clustered instances can support up to 32 block volumes of up to 32TB each, for a total of 1PB of usable storage space across both controller instances. Most installations are configured with much smaller storage footprints, and over time may need to expand the pools to provide more storage. This section documents the process of automatically expanding an existing pool.

NOTE: The Automated Capacity Scaling feature described below was released in OS8.8.57. Instances running an older version of the software are encouraged to upgrade by following the steps in the section “[Upgrading Your ZFS-HA Instance](#)”. New instances should be created using an image based on OS8.8.57 or above.

Automatic Capacity Scaling, when enabled, will monitor the pool’s used capacity. If it reaches 80%, the pool will be expanded by a combination of expanding existing block volumes and adding new block volumes to the pool until its used capacity drops to 60%. Optionally, a maximum limit can be set to prevent the pool from being expanded beyond this limit. Because ZFS storage pools cannot be reduced in size, this limit can help prevent unexpected jumps in OCI Block Volume storage charges.

Automatic Capacity Scaling may be disabled at any time. The limit may also be changed but cannot be set below the current pool size.

The Automatic Capacity Scaling feature relies on the Hardware: add-expand-disks authorization. Ensure this is enabled for the opc user in the Configuration->Users screen of the ZFS-HA BUI.

Set Automatic Expansion of a pool using the BUI

For each storage pool listed in the Configuration->Storage screen of the ZFS-HA BUI, Automated Capacity Scaling may be enabled by checking the appropriate box.

The screenshot displays the Oracle ZFS Storage OCI BUI Configuration page. The 'Available Pools' section lists the pool 'jh-zfs-a:pool-a' with a 'Striped' data profile and 'Online' status. The 'Automated Capacity Scaling' checkbox is checked, and the 'Limit' is set to 10. The 'Allocation' section shows a pie chart for 'Data' at 2.93T. The 'Device Status' section at the bottom indicates 'No device faults have been detected in the storage pool.'

Set Automatic Expansion of a pool using the CLI

From the main prompt after logging in with SSH, enter “configuration storage”. At this point you may give the command “list” to show the pools and their settings.

```

jh-zfs-a:> configuration storage
jh-zfs-a:configuration storage> list
Properties:
        pool = pool-a
        status = online
        errors = 0
        owner = jh-zfs-a
        profile = stripe
        log_profile = -
        cache_profile = -
        meta_profile = -
        autoscale = false
        autoscale_limit = 0
        scrub = scrub completed after 0h0m with 0 errors at 2023-8-19 12:23:28
        scrub_schedule = 30 days
        async_destroy_reclaim_space = 0
        encryption = off

```

Use the “select” command to choose a pool to work with if there are multiple pools.
In the example below, the autoscale property is set to true, and a limit that the pool may scale to is set to 10T.

```

jh-zfs-a:configuration storage> set autoscale=true
        autoscale = true (uncommitted)
jh-zfs-a:configuration storage> set autoscale_limit=10T
        autoscale_limit = 10T (uncommitted)
jh-zfs-a:configuration storage> commit

```

Set Automatic Expansion of a pool using the RESTful API

The autoscale and autoscale_limit properties shown in the above section can also be listed and set with calls to the RESTful API as shown in the following examples:

A pool's properties can be listed with a GET call to “api/storage/v1/pools/<poolname>” as shown in this example:

```

curl -XGET -k -u root:password https://<ip_address>:215/api/storage/v1/pools/pool-a | json_pp
{
  "pool": {
    "asn": "3b545811-68cc-46de-8000-a80e869f920a",
    "async_destroy_reclaim_space": 0,
    "autoscale": "True",
    "autoscale_limit": "1099511627776.0",
    [ snip ]
  }
}

```

The pool's properties can be set with a PUT call. An example of turning autolimit on and setting autoscale_limit:

```

curl -XPUT -H "Content-Type: application/json" -k -u root:password \
  https://<ip_address>:215/api/storage/v1/pools/pool-a/edit \
  -d '{"autoscale": true, "autoscale_limit": "10T"}'

```

Full documentation on working with the RESTful API can be found in the [Oracle ZFS Storage Appliance RESTful API Guide, Release OS8.8.x](#)

Manually Expanding a ZFS-HA Storage Pool

Pools may also be expanded manually by creating new OCI block volumes and adding them to the pool, or by expanding existing ones.

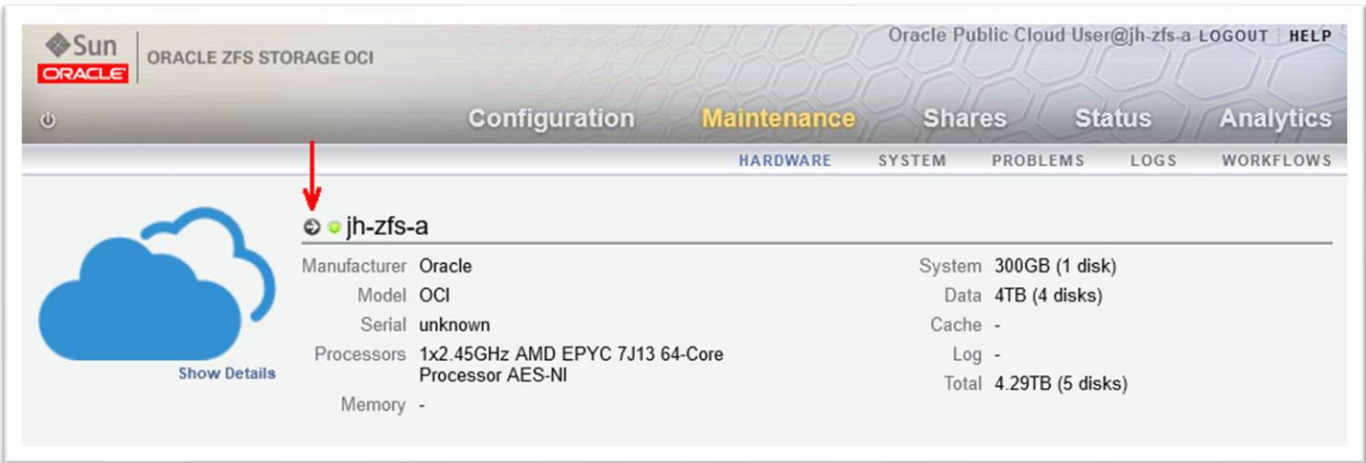
Both can be done on ZFS-HA instances without the need to configure anything through an OCI console.

Expanding Existing Block Volumes

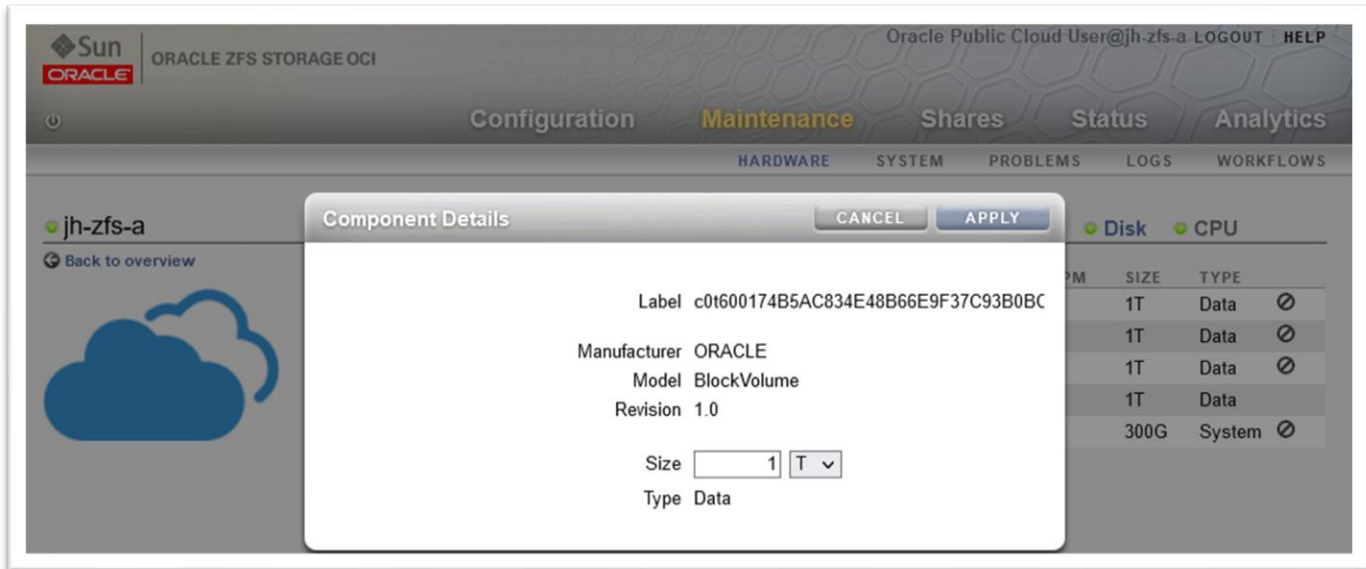
When expanding an existing block volume, the pool will be grown automatically.

Expanding a Block Volume using the BUI

Navigate to the Maintenance → Hardware screen and click on the arrow to the left of the instance name to bring up the detail view.



Click on the green circle of a listed disk to bring up the Component Details window.



Here you may increase the size of a disk up to 32T. When the Apply button is clicked, the volume size is increased and integrated into the pool.

Note that while the boot volume, listed as “System” in the detail view, may be enlarged, there is generally no reason to do so.

Expanding a Block Volume using the CLI

From the main prompt after logging in with SSH, enter “configuration hardware” and then “select chassis-000” to affect the controller, then “select disk”. Then use “list” to list the block volumes connected to the system. Use the select command to choose the volume to resize. Use the “set” command to set the size, which can range from 50G to 32T, then commit the change.

```
jh-zfs-a:maintenance chassis-000 disk> select disk-004
jh-zfs-a:maintenance chassis-000 disk-004> ls
Properties:
    label = c0t60F51204A1A34E099587F38929751240d0
    present = true
    faulted = false
    manufacturer = ORACLE
    model = BlockVolume
    revision = 1.0
    size = 1T
    type = data
    use = data
    device = c0t60F51204A1A34E099587F38929751240d0
    offline = false

jh-zfs-a:maintenance chassis-000 disk-004> set size=2T
    size = 2T (uncommitted)
jh-zfs-a:maintenance chassis-000 disk-004> commit
```

Expanding a Block Volume using the RESTful API

The attached volumes can be listed with a GET call to `https://<ip_address>:215/api/hardware/v1/chassis/chassis-000/disk`. Example:

```
curl -XGET -H "Content-Type: application/json" -k -u root:password \
https://<ip_address>:215/api/hardware/v1/chassis/chassis-000/disk | json_pp
{
  "disk": [
    {
      "label": "c0t600174B5AC834E48B66E9F37C93B0BC8d0",
      "present": true,
      "faulted": false,
      "manufacturer": "ORACLE",
      "model": "BlockVolume",
      "revision": "1.0",
      "size": 1099511627776,
      "type": "data",
      "use": "data",
      "device": "c0t600174B5AC834E48B66E9F37C93B0BC8d0",
      "offline": false,
      "href": "/api/hardware/v1/chassis/chassis-000/disk/disk-000"
    },
    [ snip ]
  ]
}
```

Expand the volume with a PUT call and specify the new size:

```
curl -XPUT -H "Content-Type: application/json" -k -u root:password \
https://<ip_address>:215/api/hardware/v1/chassis/chassis-000/disk/disk-002 -d '{"size": "2T"}' | json_pp
{
  "disk":
  {
    "href": "/api/hardware/v1/chassis/chassis-000/disk/disk-002",
    "label": "c0t6054CEFD72DB49FAAF8A7D41E9AB1D23d0",
    "present": true,
    "faulted": false,
    "manufacturer": "ORACLE",
    "model": "BlockVolume",
    "revision": "1.0",
    "size": 2199023255552,
    "type": "data",
    "device": "c0t6054CEFD72DB49FAAF8A7D41E9AB1D23d0"
  }
}
```

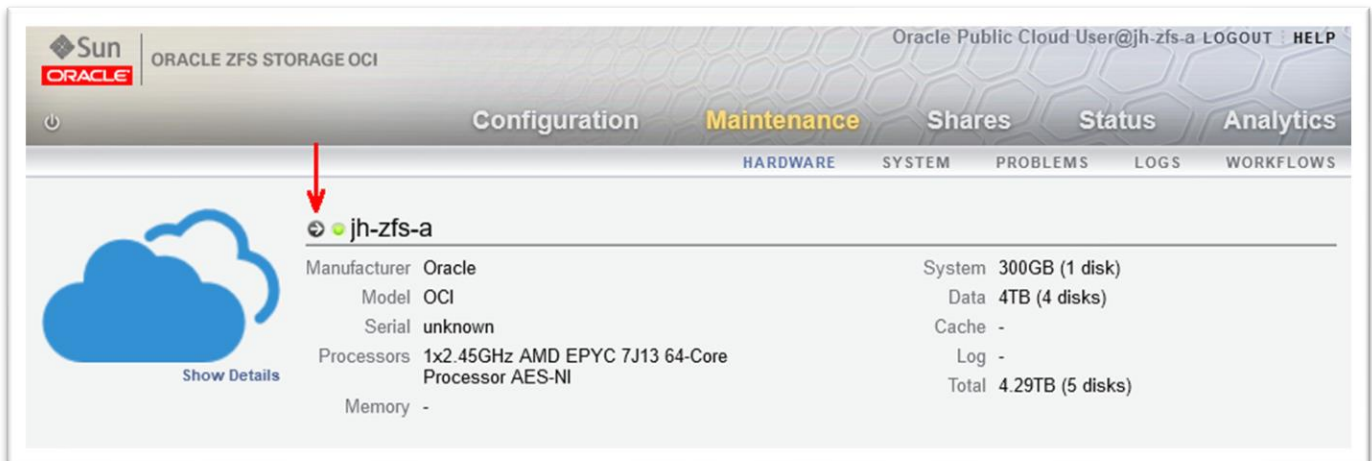
Full documentation on working with the RESTful API can be found in the [Oracle ZFS Storage Appliance RESTful API Guide, Release OS8.8.x](#)

Creating New Block Volumes

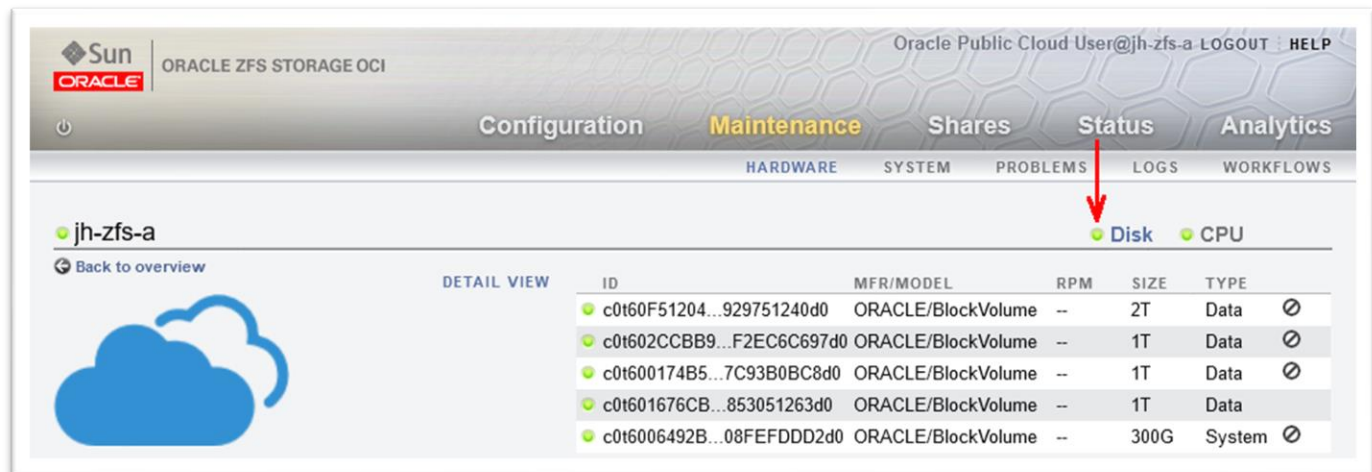
After creating a new block volume, it must be added to the pool manually. This step is explained in the next section.

Creating a Block Volume using the BUI

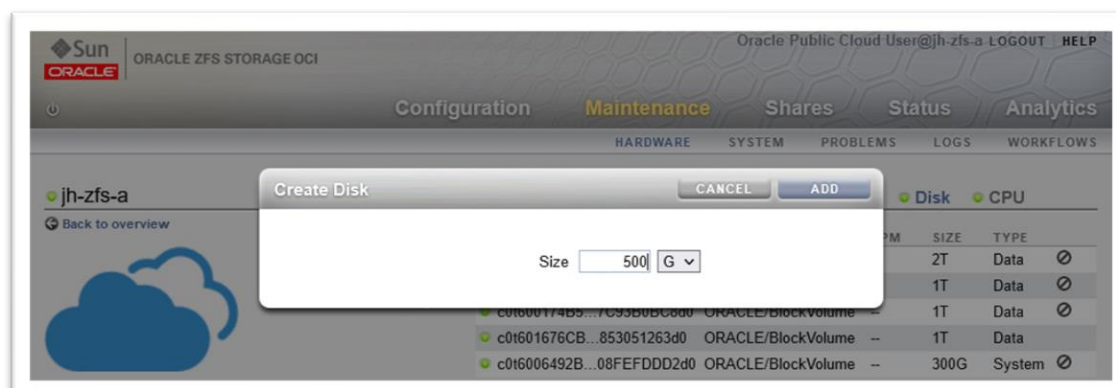
Navigate to the Maintenance → Hardware screen and click on the arrow to the left of the instance name to bring up the detail view.



In the detail view, click on the green circle for Disk to bring up the Add Disk window.



Enter the size of the new volume, then click the Add button to create the disk.



Creating a Block Volume using the CLI

From the main prompt after logging in with SSH, enter “configuration hardware” and then “select chassis-000” to affect the controller, then “select disk”. Use the “create” command to add the new volume, then set the size of the new volume to between 50G and 32T, and commit to add the volume.

```
jh-zfs-a:maintenance chassis-000 disk> create
jh-zfs-a:maintenance chassis-000 disk create (uncommitted)> set size=50G
size = 50G
jh-zfs-a:maintenance chassis-000 disk create (uncommitted)> commit
```

Creating a Block Volume using the RESTful API

Create a new block volume with a POST call and specify the size:

```
curl -XPOST -H "Content-Type: application/json" -k -u root:password \
https://<ip_address>:215/api/hardware/v1/chassis/chassis-000/disk -d '{"size": "50G"}' | json_pp
{
  "disk": [
    {
      "label": "c0t60D08AE0B8D44AD69A00451701055761d0",
      "present": true,
      "faulted": false,
      "manufacturer": "ORACLE",
      "model": "BlockVolume",
      "revision": "1.0",
      "size": 295279001600,
      "type": "data",
      "use": "system",
      "device": "c0t60D08AE0B8D44AD69A00451701055761d0",
      "offline": false,
      "href": "/api/hardware/v1/chassis/chassis-000/disk/disk-000"
    },
    {
      "label": "c0t60036C55D381426FAB5487581748332Bd0",
      "present": true,
      "faulted": false,
      "manufacturer": "ORACLE",
      "model": "BlockVolume",
      "revision": "1.0",
      "size": 53687091200,
      "type": "data",
      "device": "c0t60036C55D381426FAB5487581748332Bd0",
      "href": "/api/hardware/v1/chassis/chassis-000/disk/disk-001"
    }
  ]
}
```

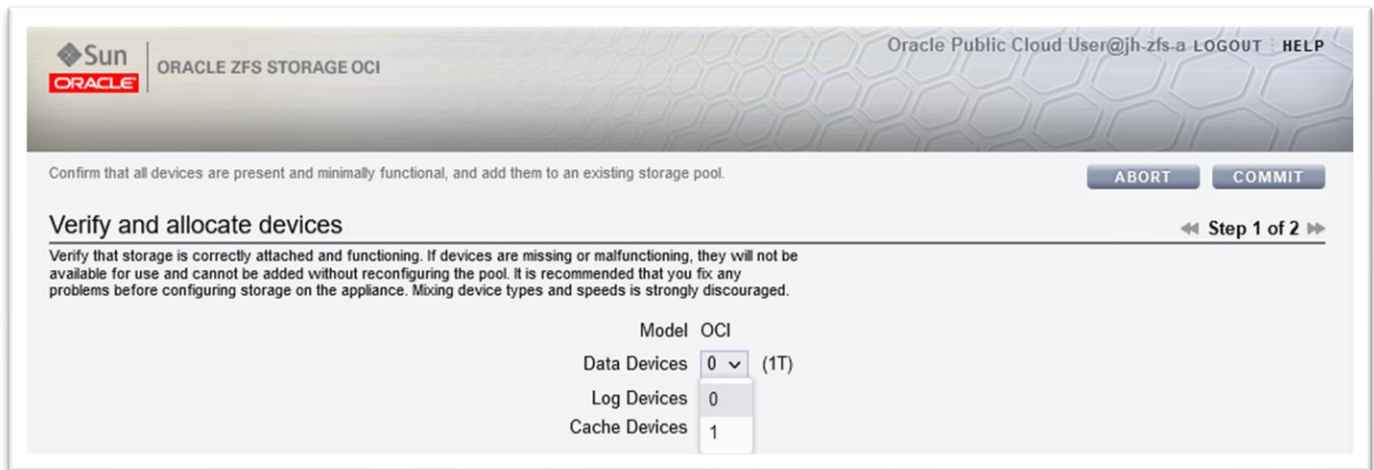
Full documentation on working with the RESTful API can be found in the [Oracle ZFS Storage Appliance RESTful API Guide, Release OS8.8.x](#)

Adding New Block Volumes to a Pool

After creating a new block volume, it must be added to the pool. This section shows how to perform this action. The default storage profile is striped, since Block Volumes are already redundant at the OCI level. Note that if you have a different storage profile, two or more volumes must be added at the same time depending on the profile.

Adding New Block Volumes to a pool using the BUI

In the BUI, navigate to the Configuration → Storage screen and click the Add button. Use the pulldown menu for Data Devices and choose the number of volumes to add to the pool, then click Apply.



Oracle Public Cloud User@jh-zfs-a LOGOUT HELP

Confirm that all devices are present and minimally functional, and add them to an existing storage pool. [ABORT] [COMMIT]

Verify and allocate devices

Verify that storage is correctly attached and functioning. If devices are missing or malfunctioning, they will not be available for use and cannot be added without reconfiguring the pool. It is recommended that you fix any problems before configuring storage on the appliance. Mixing device types and speeds is strongly discouraged.

	Model	OCI
Data Devices	0	(1T)
Log Devices	0	
Cache Devices	1	

Adding New Block Volumes to a pool using the CLI

From the main prompt after logging in with SSH, enter “configuration storage”. If you have a single pool, which is the default ZFS-HA configuration, the pool name is not displayed, but it is selected. If you have multiple pools, a default pool is selected and displayed. If this is not the pool to which you want to add the device, enter `set pool=` and specify another online pool.

```
jh-zfs-a:> configuration storage
jh-zfs-a:configuration storage> show
Properties:
    pool = pool-a
    status = online
    errors = 0
    owner = jh-zfs-a
    profile = stripe
    log_profile = -
    cache_profile = -
    meta_profile = -
    autoscale = false
    autoscale_limit = 0
    scrub = scrub completed after 0h0m with 0 errors at 2023-8-19 12:23:28
    scrub_schedule = 30 days
    async_destroy_reclaim_space = 0
    encryption = off
```

Enter “add”. You can use the command “show” to view the device count available. In this example, we see that zero new data volumes out of a possible one have been allocated to the pool:

```
jh-zfs-a:configuration storage verify> show

System  OCI

State   ok
Data Disks  0 of 1 (1T)
Log Disks   0 of 0
Cache Disks 0 of 0
```

Use the “set” command to specify the number and type of drive to add:

```
jh-zfs-a:configuration storage verify> set data=1
data = 1
```

Enter “done” to complete the process. You may also enter the “abort” command to interrupt the process and stop the new volumes from being added.

Adding New Block Volumes to a pool using the RESTful API

Add the previously created block volumes to a pool with a PUT call and specify the number of volumes to add as data volumes. The new pool status will be returned:

```
curl -XPUT -H "Content-Type: application/json" -k -u
root:zfsisfun! https://<ip_address>:215/api/storage/v1/pools/pool-a/add -d '{"data": 1}' | json_pp

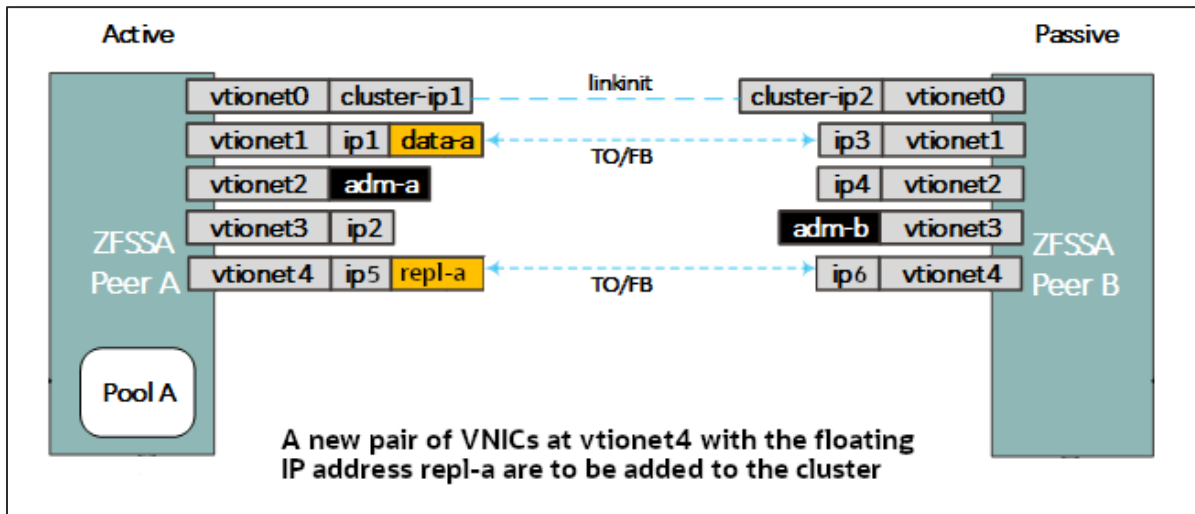
{
  "pool" : {
    "name" : "pool-a",
    "vdev" : [
      {
        "label" : "c0t60F51204A1A34E099587F38929751240d0",
        "type" : "disk",
        "chassis" : "jh-zfs-a",
        "state" : "healthy"
      },
      {
        "type" : "disk",
        "label" : "c0t600174B5AC834E48B66E9F37C93B0BC8d0",
        "chassis" : "jh-zfs-a",
        "state" : "healthy"
      },
      {
        "type" : "disk",
        "label" : "c0t602CCBB9D82843C8B7B5A25F2EC6C697d0",
        "chassis" : "jh-zfs-a",
        "state" : "healthy"
      },
      {
        "state" : "healthy",
        "chassis" : "jh-zfs-a",
        "type" : "disk",
        "label" : "c0t601676CBF59C434B9A44BBB853051263d0"
      }
    ],
    [ pool properties snipped ]
  }
}
```

Full documentation on working with the RESTful API can be found in the [Oracle ZFS Storage Appliance RESTful API Guide, Release OS8.8.x](#)

Adding Clustered Interfaces

When using ZFS Storage in OCI, it may be advantageous to spread workloads across multiple VNICs. One possible use case detailed here is to send or receive ZFS replication traffic across a new set of interfaces. This requires configuration both in the OCI console and on the ZFS-HA instances. Configuration of the ZFS-HA instances will be done using the ZFS Browser User Interface (BUI) which is accessed via the IP address of the admin interface on port 215.

The diagram below shows an active/passive cluster with a new clustered interface on the vtionet4 VNIC on each controller and a new IP address, repl-a, that will float with the pool to the active controller. The process is the same whether the cluster is an active/active or active/passive one.



Overview

1. Attach a new VNIC on both instances in OCI console
2. Add a secondary IP address to the new VNIC on the active node (usually A) in OCI console
3. Reboot passive then active nodes
4. Run Takeover/Failback on node A
5. Edit new interface on active node
 - Change name
 - Change IP address to secondary IP address from step 2
6. Reboot and rebalance the clustered resources

1 - Attach New VNICs In the OCI Console

We start by adding VNICs to the ZFS-HA instances in the OCI cluster. When adding VNICs, use the naming convention <clustername>-<type>-[a|b]. The example cluster name here is jh-zfs and the type is repl (for replication). Your configuration will vary.

Find the A and B instances in the OCI console and open a browser tab for each. Click on Attached VNICs in the Resources sidebar to access the correct screen. Click Create VNIC to add a new VNIC. Each instance will have one interface added.

Add the new VNIC to each controller. On the A node, name it jh-zfs-repl-a and on the B node, name the VNIC jh-zfs-repl-b.

Adding VNICs

Resources

Metrics

Attached block volumes

Attached VNICs

Boot volume

Console connection

Run command

Work requests

OS Management

Custom logs

Console history

Attached VNICs

A [virtual network interface card \(VNIC\)](#) lets an instance connect to a virtual cloud network (VCN) and determines how the endpoints inside and outside the VCN.

Create VNIC

Name	Subnet or VLAN ⁽ⁱ⁾	State	FQDN ⁽ⁱ⁾	VLAN tag
jh-zfs-0-b (Primary VNIC)	Subnet - jh-zfs	● Attached	jh-zfs-0-b... Show Copy	2295
jh-zfs-dx-b	Subnet - common.sub	● Attached	jh-zfs-dx-... Show Copy	1474
jh-zfs-data-b	Subnet - common.sub	● Attached	jh-zfs-dat... Show Copy	2417
jh-zfs-ax-b	Subnet - common.sub	● Attached	jh-zfs-ax-... Show Copy	1108
jh-zfs-adm-b	Subnet - common.sub	● Attached	jh-zfs-adm... Show Copy	2580

Show

When adding VNICs, choose the appropriate compartment/VCN/subnet. These will probably be the same as the data VNICs. You may also set up a separate VCN, but that's outside the scope of this documentation.

If using a separate VCN, it should be in the same compartment as the data VNICs since the new VNICs will have secondary IP addresses that can move between instances at failover/takeover events. This requires slightly elevated permissions for the compartment containing the VNICs. If a different compartment is used, it must use an Identity Policy rule to “use private-ips” rather than just reading them. It is not recommended to use the same compartment as the cluster or admin VCNs as they do not require the elevated access.

VNIC information

Name *Optional*

Select a virtual cloud network in **Networks** [\(Change Compartment\)](#)

Network

Normal setup: subnet

The typical choice when adding a VNIC to an instance. ✓

Advanced setup: VLAN

Only for experienced users who have purchased the Oracle Cloud VMware Solution.

Select a subnet in **Networks** [\(Change Compartment\)](#)

☐ Use network security groups to control traffic (optional) ⁽ⁱ⁾

☐ Skip source/destination check ⁽ⁱ⁾

Assigned IP addresses are usually fine and public IPs are not required. Verify these choices with your OCI network team.

2 - Assign Secondary IP Addresses

Assign a secondary IP address to the new repl-a VNIC on the active controller and note the IP and subnet mask for later.

Resources

IPv4 Addresses

Assign Secondary Private IP Address

Private IP Address	Public IP Address	Fully Qualified Domain Name	Assigned
192.168.216.216 (Primary IP)	(Not Assigned)	-	Thu, Dec 22, 2022, 17:48:50 UTC

Showing 1 item

3 - Reboot Both Controllers

Reboot the nodes to make the new VNICs visible. Reboot one, allow time for it to come back up, then reboot the other. In the case of an active/passive cluster, reboot the passive instance first. There is no preferred order for an active/active cluster.

4 – Run Takeover/Failback

When both have come back up, redistribute the resources if needed using Configuration->Cluster->Failback/Takeover from the ZFS BUI.

ConfigurationMaintenanceSharesStatusAnalytics

SERVICESSTORAGENETWORKSANCLUSTERUSERSPREFERENCESSETTINGSALERTS

SETUPUNCONFIGFAILBACKTAKEOVERREVERTAPPLY

jh-zfs-a

Active (takeover completed)

jh-zfs-b

Ready (waiting for failback)

Active Resources

RESOURCE	OWNER
jh-zfs-a (net/vtinet1)	jh-zfs-a
jh-zfs-b (net/vtinet2)	jh-zfs-b
jh-zfs-adm-a (net/vtinet3)	jh-zfs-a
zfs/pool-a	jh-zfs-a
zfs/pool-b	jh-zfs-b

Active Resources

No resources are active on this cluster node.


5 – Edit New Interfaces

Start by connecting to the BUI of the active node in an active/passive cluster. On an active/active cluster, connect to the node that is controlling the storage pool that should be accessed through the new VNIC. Navigate to the Configuration->Network screen. New devices, datalinks, and interfaces will have been automatically added to this screen corresponding to the VNICs that have been added to the instance.

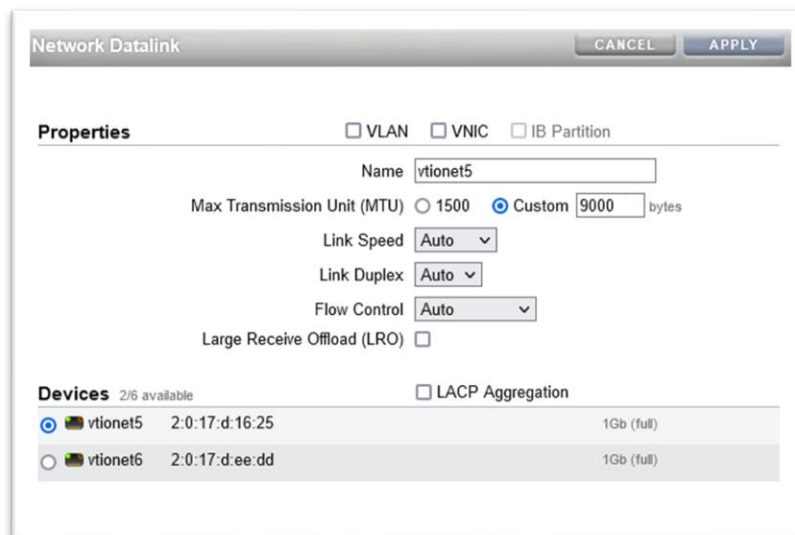
About Devices, Datalinks, and Interfaces



Network devices represent the VNICs that are attached to a VM and which connect to VCNs in your tenancy. Network devices are created by the system and have no configurable settings. The devices will have a *vtionet* label. Note that the devices will all have a listed speed of 1Gb even though they will use the full bandwidth allowed by the compute shape.

Datalinks are Layer 2 objects for sending and receiving packets for specific network devices. Use datalinks to apply settings such as MTU to network devices. Datalinks can correspond 1:1 with a device, or you can define VLAN datalinks composed of other devices and datalinks. Datalinks are required to complete network configuration, even if the datalinks do not apply specific settings to network devices.

Edit the datalink by clicking on the edit icon  and review the MTU size. An MTU of 9000 gives the best performance and is recommended if all the clients are within OCI. Note that if the client is located across a Wide Area Network (WAN), such as when replicating to or from an on-premises Oracle ZFS Storage Appliance, all steps must support jumbo frames or the connection may hang. (See <https://docs.oracle.com/en-us/iaas/Content/Network/Troubleshoot/connectionhang.htm> for more details.) In these cases, an MTU size of 1500 is recommended on both the source and target datalinks.

Press Apply to save the datalink changes.



Properties		
<input type="checkbox"/> VLAN <input type="checkbox"/> VNIC <input type="checkbox"/> IB Partition		
Name: <input type="text" value="vtionet5"/>		
Max Transmission Unit (MTU): <input type="radio"/> 1500 <input checked="" type="radio"/> Custom <input type="text" value="9000"/> bytes		
Link Speed: <input type="text" value="Auto"/>		
Link Duplex: <input type="text" value="Auto"/>		
Flow Control: <input type="text" value="Auto"/>		
Large Receive Offload (LRO): <input type="checkbox"/>		
Devices		
2/6 available <input type="checkbox"/> LACP Aggregation		
<input checked="" type="radio"/>	 vtionet5	2:0:17:d:16:25 1Gb (full)
<input type="radio"/>	 vtionet5	2:0:17:d:ee:dd 1Gb (full)

Interfaces are Layer 3 objects for IP, configuring IP addresses and other properties for datalinks. The interfaces created when the new VNICs were attached to the instance are not configured correctly for high-availability clustering the way the data client network is. Edit the interfaces depending on your cluster type.

Edit the New Interfaces

On the A node, the new interface, usually named *vtionet4*, corresponds to the *repl-a* device. Click on the edit icon for the new interface.

Change the name of the interface to be consistent with our naming convention – here, we use *jh-zfs-repl-a*. Set the static IP Address/Mask to be the same as the **secondary IP address** you assigned to the *repl-a* VNIC in the OCI console – **do not use the primary IP address!** The netmask must be the same as the subnet you chose to add the VNIC to in the OCI console. Refer back to the instance details in the OCI console if needed. Press Apply to save the changes and close the window.

If an MTU change is needed as described in the section “About Devices, Datalinks, and Interfaces”, edit the new datalink to change it. You must change the MTU in the new datalinks on both nodes if a change is needed.

Network Interface

CANCEL

APPLY

Properties

Name

jh-zfs-repl-a

Enable Interface

☒

Allow Administration

☒

☒ Use IPv4 Protocol

Configure with

Static Address List

Address/Mask

192.168.1.2/24

192.168.2.17/24

+

-

Directly Reachable Network(s)

+

☐ Use IPv6 Protocol

Interface Status

Interface State

offline

Datalinks

2/5 available

☐ IP MultiPathing Group

<input type="radio"/>	<div> <div>↔</div> <div>vtionet3</div> <div>Custom MTU(9000), via vtionet3</div> </div>	2:0:17:14:89:11
<input checked="" type="radio"/>	<div> <div>↔</div> <div>vtionet5</div> <div>Custom MTU(9000), via vtionet5</div> </div>	2:0:17:9:48:f7

Finally, click the Apply button on the main Network screen to save both the datalink and interface changes. Do not click Cancel or the modifications will not be saved even if apply had been clicked in the datalink or interface windows.

6 – Reboot and Rebalance

The final step in the process is to reboot both controllers one more time, again starting with the passive node if there is one, followed by performing a failover or takeback action if needed.

On the Configuration->Cluster screen, we will see the new replication interfaces as active on their respective controllers. Configuration is complete!

We can see from the Configuration->Cluster screen that the new network interfaces are shared and use the IP addresses we assigned.

Configuration

Maintenance

Shares

Status

Analytics

SERVICES

STORAGE

NETWORK

SAN

CLUSTER

USERS

PREFERENCES

SETTINGS

ALERTS

SETUP

UNCONFIG

FAILBACK

TAKEOVER

REVERT

APPLY

jh-zfs-a

Active

jh-zfs-b

Active

vtionet0

vtionet0

Active Resources

RESOURCE	OWNER
↔ jh-zfs-a (net/vtionet1)	jh-zfs-a
↔ jh-zfs-adm-a (net/vtionet3)	jh-zfs-a
↔ jh-zfs-repl-a (net/vtionet5)	jh-zfs-a
zfs/pool-a	jh-zfs-a

Active Resources

RESOURCE	OWNER
↔ jh-zfs-b (net/vtionet2)	jh-zfs-b
↔ jh-zfs-repl-b (net/vtionet6)	jh-zfs-b
zfs/pool-b	jh-zfs-b

When setting up ZFS replication on other instances to use this cluster as a target, use the IP addresses of these replication interfaces to identify the appropriate target.

51 Oracle ZFS Storage – High Availability User Guide | Version 4.15
Copyright © 2024, Oracle and/or its affiliates | Public

UPGRADING YOUR ZFS-HA INSTANCE

The ZFS-HA controller instances use the same Operating System (OS) releases as the hardware ZFS Storage Appliances. Your ZFS-HA instances should be updated on a regular basis to enable new features and apply bug fixes.

Note that upgrading the software on a single controller system will incur a few minutes of downtime while the system reboots to activate the new code. This time will be slightly higher on a bare metal instance than on a virtual machine instance.

New versions of the OS are released monthly. Release notes and downloads of the OS as well as upgrade instructions are available at [My Oracle Support Doc ID 2021771.1](#).

Because the releases are the same for both the hardware and cloud based ZFS systems, the documentation may refer to service processor (SP) upgrades or device firmware. These do not apply to ZFS-HA instances in OCI and may be ignored.

Once a system is upgraded, you can choose to apply deferred updates at a time of your choosing. Deferred updates can provide additional features or fixes to the system but they do remove the possibility of rolling back an update. It is generally considered a good idea to run a few days with the OS upgrade before applying the deferred updates. Deferred updates can be applied without any downtime or service interruption.

ZFS-HA SYSTEM NOTES

Networking

ZFS-HA Network Routing

- It is recommended to set the multihoming model to strict. This is the default when the Deployment Tool configures the cluster.

ZFS-HA Network Datalinks

- Link Speed, Link Duplex and Flow Control should all be set to Auto.
- Link speed for VM instances will be reported as 1GB but will actually use the full amount of bandwidth allocated to the instance. (See known issues)
- All network datalinks should have the MTU set to 9000 for best performance.

ZFS-HA Network Interfaces

- The primary network interface used for iSCSI traffic should not be modified because it can cause a system panic. (See known issues)
- NAS client interfaces should uncheck 'Allow Administration' for enhanced security.

Block Storage Notes

System Boot Disk

- System disk contains read only OS image, logs, core dumps and configurations.
- Configuration data can be backed up using 'Maintenance -> System -> Configs'
- Does not include OS image, logs, core dumps, replication or share data.
- Logs and core dumps can be saved using 'Maintenance -> System -> Bundles'

Storage Pools

- Pool disks contain all configuration data under 'Shares'
- All disks in each pool should be same size especially if they are under 800GB.
- All data disks in each pool should have the same performance settings.
- It is suggested to create a volume group containing all data disks for each storage pool.
- For best system resource usage, it is recommended to have only one pool per VM.

- All data disks provided by OCI have multiple copies so striped pools provide data protection. ZFS will detect bit rot but data will have to be restored from backup if bit rot is detected.

Boot and Block Volume Backups

While it is possible to use OCI's Boot and Block Volume Backup services to create snapshots of either the storage pool block or boot volumes, it is not recommended. Using these services will require the entire ZFS-HA cluster be shut down to ensure that the snapshots would be usable.

It is recommended instead that the storage pools be backed up in at least one of two ways:

- Using [ZFS Remote Replication](#) to copy share or project snapshots to another ZFS Storage instance, whether an on-premises ZFS Storage Appliance or other ZFS-HA instances in OCI
- To OCI Object Storage using the ZFS appliance's built-in Cloud service, which leverages ZFS snapshots for object storage backups. See the documentation on [Configuring Cloud Backups](#) for details on enabling and using this service.

It is also recommended that rather than boot volume backups, the ZFS configuration be backed up and downloaded for safekeeping.

Backing Up the ZFS Configuration

It is recommended that after your ZFS Storage in OCI instance is configured, that you create a backup of the configuration with the following steps:

- From the Appliance BUI, go to Maintenance→System.
- Under the Configurations section, click Backup.
- This will create a backup of the Appliance configuration, that can be downloaded and stored separately for recover purposes.

For information about the configuration backup content, especially what is included and what is not included in a configuration backup, see [Backing Up the Configuration](#).

DOCUMENTATION AND SECURITY REFERENCES

For information about setting permissions on shares and recommended security practices, see the following references:

- [Access Control Lists for Filesystems](#)
- [Oracle® ZFS Storage Appliance Security Guide, Release OS8.8.x](#)
- [Oracle ZFS Storage Appliance RESTful API Guide, Release OS8.8.x](#)

INSTALLATION NOTES

Root User Configuration

You will need to configure the root user to perform some tasks such as taking a configuration backup.

To enable root login over ssh, from the Appliance BUI, go to the Configuration tab to reach the Configuration Services screen. Under Remote Access, select the ssh service. From the ssh service screen, enable Permit root login.

For more detailed configuration information, see [My Oracle Support Doc ID 2811414.1](#).

Known Issues

- Virtual Machine instances will show network devices speed as 1Gb even though it will use the full bandwidth allowed by the compute shape. (32749253 - VNICs speed is mentioned as 1G at CLI/BUI though VNIC effective bandwidth is more)
- Destroying the cluster with a saved Terraform stack fails when detaching VNICs from the Compute instances. The workaround is to terminate the cluster nodes from the OCI console and rerun the “Destroy” action.
- If a new OCI VNIC is added to a running ZFS Storage in OCI VM, a reboot is required before the network device can be used. (32518670 - Adding an additional VNIC to the OCI ZFS-HA VM fails)
- If OCI VNICs are added before a VM instance finishes its first boot there is a chance the instance will hang. The workaround is to wait for system to finish booting before adding OCI VNICs and then reboot the instance to pick up the new VNICs. (34045542 - ZFSSA hangs if OCI VNICs are added while booting large VM shapes)
- When data connections are made across Wide Area Networks (WANs) that do not support jumbo frames at each step along the network path, the connections may hang. (See <https://docs.oracle.com/en-us/iaas/Content/Network/Troubleshoot/connectionhang.htm> for more detail.) In these cases it is best to set the MTU on the ZFS-HA datalink used and at the on-premises client to 1500.

CONNECT WITH US

Call +1.800.ORACLE1 or visit [oracle.com](https://www.oracle.com).

Outside North America, find your local office at [oracle.com/contact](https://www.oracle.com/contact).

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2024, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

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

ZFS-HA Quick Start Guide v4.15
November 2024
Author: Joe Hartley

