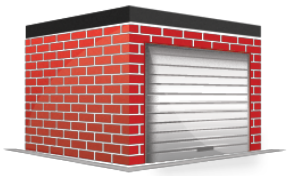
`



BRICKSTOR

# RackTop BrickStor Appliance

Guide

This guide applies to BrickStorOS version 18.12 and later.

Rev 2.0.0

`

# Table of Contents

`

`

`

## Purpose

This document describes the overarching architecture of BrickStor and some of the most critical administrative features and functions to get the system online. It includes examples of common administrative tasks, configuration options, and explanations of system settings. This information contained herein is intended for anyone who wants to configure or administer the BrickStor, and is written for individuals familiar with storage area network terminology.

If you find incorrect information within this manual, please email errata@racktopsystems.com with the subject “Documentation Errata”.

`

## Galaxy Series Appliance Platform

The RackTop BrickStor™ Appliance leverages the RackTop Galaxy (3000) series hardware platform.

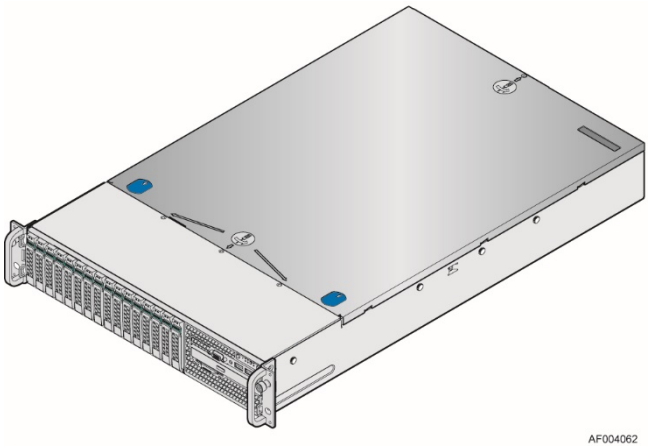



Figure 1 - RackTop Galaxy Series Appliance

## Appliance Components

This section helps you identify the components of your server system. If you are near the system, you can also use the Quick Reference Label provided on the inside of the chassis cover to assist in identifying components.

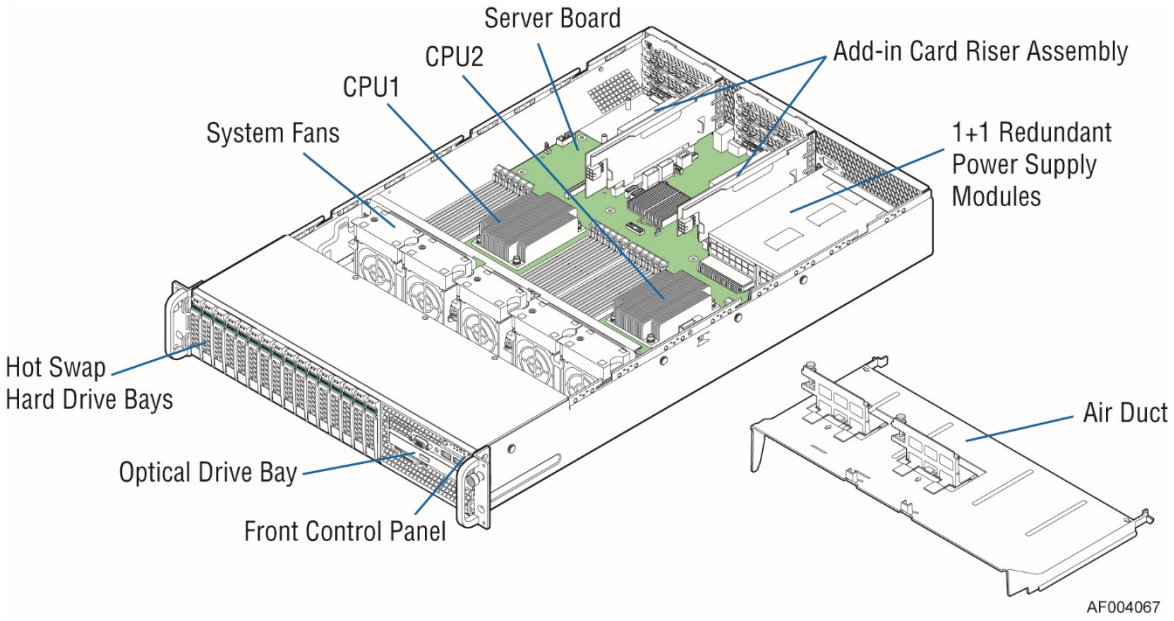



Figure 2 - RackTop Galaxy Series Appliance Components

`

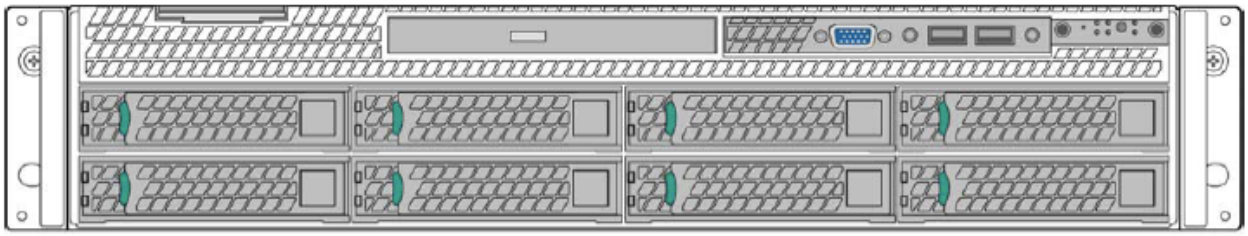
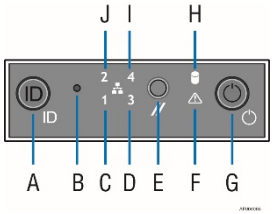


Figure 3 - Front Drive Configuration

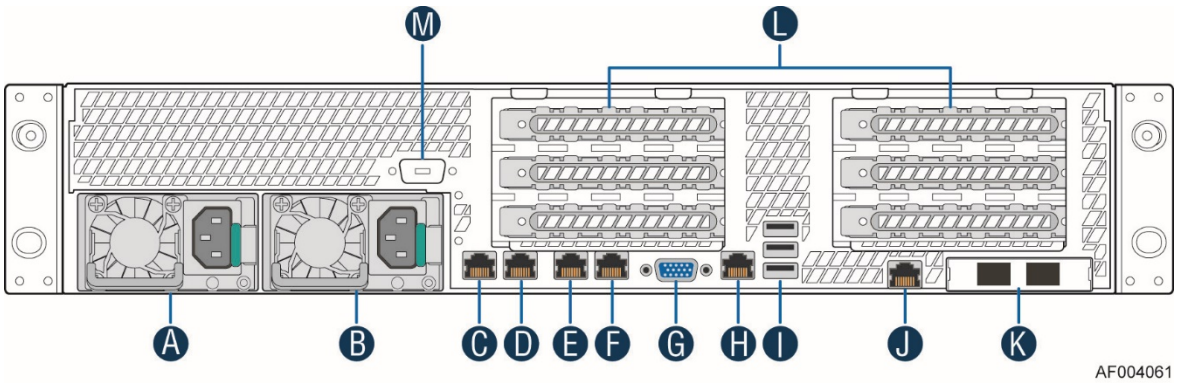
## Front Panel




| Label | Description | Label | Description |
|---|---|---|---|
| A | System ID Button w/Integrated LED | F | System Status LED |
| B | NMI Button (recessed, tool required for use) | G | Power Button w/Integrated LED |
| C | NIC-1 Activity LED | H | Hard Drive Activity LED |
| D | NIC-3 Activity LED | I | NIC-4 Activity LED |
| E | System Cold Reset Button | J | NIC-2 Activity LED |

Figure 4 - Front Panel Options

## Back Panel (TI31XX)

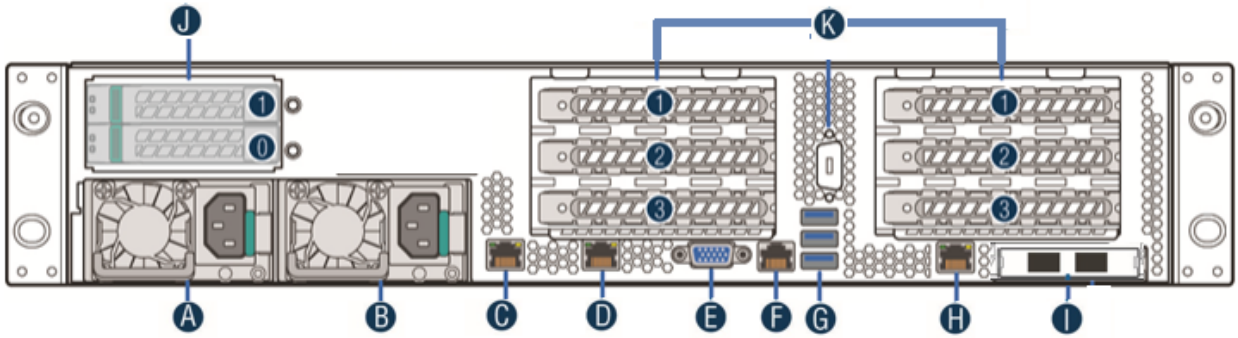



| | | | |
|---|---|---|---|
| A | Power Supply Module #1 | H | RJ45 Serial-A Port |
| B | Power Supply Module #2 | I | USB Ports |
| C | NIC – 1 | J | RMM4 NIC Port |
| D | NIC – 2 | K | 10G IO Module |
| E | NIC – 3 | L | Add-in adapter slots via Riser Card 1 and Riser Card 2 |
| F | NIC – 4 | M | Serial-B Port (Optional) |
| G | Video Connector | | |

Figure 5 - Back Panel

`

## Back Panel (TI32XX)




| | | | |
|---|---|---|---|
| A | Power Supply Module #1 | G | USB |
| B | Power Supply Module #2 | H | RMM Out of Band Management |
| C | NIC – 1 10Gb BaseT | I | 10GB or 40GB Network Module |
| D | NIC – 2 10Gb BaseT | J | Mirrored OS Bays |
| E | Video Connector | K | PCI Bays |
| F | RJ45 Serial-A Port | | |

Figure 6 - Back Panel

`

# Physical Components of BrickStor

## Controller / Head

A controller is sometimes referred to as the head or the compute part of a BrickStor system. It contains the operating system and is the gateway interface into managing your BrickStor, as well as the part of the system exposing storage services, providing content management, security, auditing, etc. A typical controller is equipped with multi-core Intel CPUs and 256GB or larger memory. This memory is used for caching discussed in greater detail later in this document. Networking is provided via onboard interfaces with typical system containing two 10GbE Ethernet interfaces onboard, and two or more 10GbE or faster Ethernet interfaces as add-on components. Component redundancy is provided wherever possible, including power, cooling, storage used by operating system, etc. While controllers are field serviceable, a lot of effort is dedicated to eliminating the need for this service in the first place.

## Storage Enclosure / JBOD

A storage enclosure, often referred to as a JBOD is at its essence a box with redundant components, which just like the controller is engineered to be fault-tolerant and keep functioning in various degraded states. A JBOD is either fully or partially populated with mechanical and/or solid state drives. A typical configuration is what we refer to as a '*hybrid storage'* system, concept discussed in some detail later in this document. These drives are the primary storage for your BrickStor and organized into logical groupings referred to as pools, another concept discussed in more detail later in this document. Special cache and write optimized '*journal'* devices are frequently also installed in this enclosure. Their purpose is discussed later in the document. A typical configuration consists of at least a single JBOD and a single controller, with some number of drives in the JBOD. There are high availability options available in addition to this 'basic standard' configuration. High availability is a configuration which includes two controllers and one or more JBODs with shared access between these controllers. The basic premise is high availability to some degree protects from catastrophic physical failure, or failure in operating system on a controller. Because storage is common between the controllers, high availability configuration is not meant to provide increased protection for storage, instead storage is protected through mirroring or a parity scheme such as RAID. This is discussed later in the document.

Enclosures are attached to controller(s) via dual SAS host controllers, and utilize SAS drives, which permit dual pathing throughout the system. This is another feature which adds to redundancy of the system. Loss of path to storage may cause a pause, while system recovers from the loss and continues operating with a single remaining path. Whenever possible, RackTop recommends having dual pathing throughout. Diagrams provided at installation time have necessary detail about recommended configuration.

## Drives

While in some instances special purpose drives used for caching or journaling are installed in the controller, in a typical configuration mechanical and solid state drives are installed in the enclosure. Both types of drives use SAS interface, which possesses dual-ported capability and enables dual pathing as described in the last section. Enterprise grade drives are a standard feature in all systems and selected to fit a specific configuration both in terms of capacity and parity scheme or mirroring.

`

## Logical Components of BrickStor

### BrickStorOS

BrickStorOS is the Operating System running on your BrickStor appliance. It is not a general-purpose operating system, on the contrary it is for all intents and purposes part of an embedded system, which in combination with RackTop compute hardware becomes a BrickStor storage appliance.

Like most appliances, there is a console mode, and there is shell access, restricted as well as unrestricted, but these exist for supporting very low-level functionality such as configuration of certain things, optimization, troubleshooting and other diagnostic functions. Take caution when attempting to perform actions within the OS that are not documented or recommended by RackTop as it may result in system instability, loss of data and violation of the terms of the system's maintenance contract.

### VDEV

A '*vdev'* is a virtual device which can be a single disk, two or more disks that are mirrored, or a group of disks with a parity scheme such as RAID-5. The idea of a vdev is something that abstracts away some a some unit of storage, which may or may not have any redundancy. One can think of a vdev as a building block in pools, a concept that we address next. Usually, when you hear this term from someone of RackTop it is used to mean a group of disks, and could usually be replaced with word stripe, which will have roughly same meaning in terms of how BrickStorOS implements redundanacy in the storage.

### Hybrid Pool

A *'hybrid pool'* is the name for a collection of drives, optionally with dedicated read-optimized cache devices and/or write optimized journal devices. All pools are hybrid pools because they are a combination of in-memory read cache as well as actual high capacity persistent storage and optionally read and write cache devices. The high capacity data drives are organized in virtual devices frequently referred to as vdevs. Pools are groups of virtual devices usually with some data protection scheme, such as RAID or mirroring, on top of which filesystems and raw block devices are provisioned. A typical pool is what RackTop refers to as a hybrid pool is a mix of mechanical drives and solid state drives. In such a pool data is redundantly stored on large capacity, slower, typically mechanical devices, arranged into a parity scheme that satisfies data protection as well as capacity and IOPS requirements, while high bandwidth, low latency solid state drives are used for the purposes of caching to accelerate reads and for the purposes of handling synchronous writes, enabling a much better cost to performance ratio over traditional purely mechanical, or purely solid state configurations.

One or more data pools must exist on a system in order to present storage to consumers via AFP, NFS, SMB, etc. While there is no hard limit on number of pools a system could have, usually fewer than 4 pools are configured on any given system. Under normal circumstances the burden of designing and configuring pools is not on the customer, but in the instances where a system is no longer satisfying previously prescribed requirements, RackTop strongly recommends that customer contacts support before any changes are made to configuration of any pool.

`

From a system administrator's point of view a pool is a logical organization of independent drives and contains all information about the devices comprising it, structure, filesystems, raw volumes, replication target if any, etc., encoded within its metadata, which makes it possible to easily migrate pools between systems. Critically, this property means that loss of the controller does not in any way compromise data. A replacement controller is all that's necessary to return to normal operations. This feature also enables RackTop's high availability product, which moves pools as well as related network configuration between nodes in the cluster.

`

## Adaptive Replacement Cache (ARC)

The '*ARC*' is a portion of memory in the controller dedicated to caching recently accessed data. The ARC caches both recently written data, with assumption that this data may be read soon after being written as well as recently read data, with assumption that this data is potentially going to be read again. Depending on popularity of data it may remain in the cache for a long time, or be evicted in favor of other data, based on criteria which both the user as well as system can optimize for.

## Level 2 Adaptive Replacement Cache (L2ARC) - Read Cache

The '*L2ARC*' is an optional SSD Cache device that can be used to extend the amount of data that is cached for Reads. There must be enough space in RAM for the block pointers required to address the L2ARC. When data is evicted from the ARC it will potentially move to the L2ARC (based up on user configuration settings). Data read from L2ARC will be moved back into ARC.

## Journal

A journal is both a software concept and a core physical component, a write ahead log that is used to reduce latency on storage when synchronous writes are issued by clients. RackTop frequently refers to journal as a ZIL, an intent log or a log device. In synchronous write cases, writes are committed to this journal and periodically pushed to primary storage. Journal guarantees that data is protected from loss on power failure due to being in cache before cache is flushed to stable storage.

A log device and is normally only ever written to and never read from. A log device i.e. journal is present to protect the system from unexpected interruptions, such as power loss, a system crash, loss of storage connectivity, etc. In rare instances where due to power loss or other catastrophe, recovery is necessary, journal is read from in order to recreate a consistent state of the pool, which may require rolling back some transactions, but results in restoring pool to a consistent state, unlike traditional storage systems where only best effort is promised. RackTop recommends mirroring journal devices as a means of preventing loss of journal device, which has performance and potential availability impact. All pools configured at the factory prior to system shipping, the journal, if present, will be mirrored.

`

## Data Protection Schemes

BrickStor is not a traditional RAID system and should not be compared to one. Unlike a traditional system where a RAID controller is a piece of hardware with severely restricted processing power and caching abilities, specifically designed to support one of a number of possible RAID schemes, a BrickStor implements this in software and benefits from the full power and capability of a purpose-engineered operating system, massively powerful processors and huge cache, which in combination allow for things such as encryption, data reduction by means of compression and in certain situations deduplication, end-to-end data integrity by means of check-summing and storing multiple copies of checksums of each data and metadata block elsewhere within a given pool. This is also integrated with a notion of snapshots, which leverage the same underlying building blocks, and made even more useful by read/write snapshots referred to as clones.

This in software implementation allows for various parity schemes as well as mirroring configurations. The following are schemes currently supported by RackTop:

- No Parity - fast, but with only minimal protection, and total loss if any single device is lost, useful for scratch-only data
- Mirrored - Equivalent to RAID 10 / RAID 1+0, aka a stripe of mirrors, where two or more drives in a mirror are possible, offers highest availability with a capacity trade-off
- RAIDZ1 (single parity) - Equivalent to RAID 50 / RAID 5+0, which allows for loss of a single drive in each group (vdev)
- RAIDZ2 (double parity) - Equivalent to RAID 60 / RAID 6+0, which allows for loss of two drives in each group (vdev)
- RAIDZ3 (triple parity) - simiar to RAIDZ2, but with even more parity protection, allowing for loss of three drives in each group (vdev)

*See the section about performance to understand the implications of each RAID scheme.*

## Resilvering

Resilvering is the process of rebuilding a disk within a vdev after a device has been removed or failed. ZFS has no fsck repair tool equivalent, common on Unix filesystems. Instead, ZFS has a repair tool called "scrub" which examines and repairs silent corruption and other problems. Scrub can run while the volume is online; scrub checks everything, including metadata and the data. This process works from the top down and only writes data to the disk that is needed. If a disk was temporarily offline it would only have to rebuild the data that was missed while the device was offline. Resilvering does not occur upon removal of disk but happens once disk has been replaced.

`

# Initial Setup and Quick Start

## *Cabling and Connections*

All BrickStor and Shared storage shelves come with Redundant and hot swappable power supplies. Both power supplies should be connected to reduce the risk of outage. For a basic configuration with one storage shelf and one BrickStor connect a SAS Cable from one external mini-SAS port of the BrickStor to port A on the top expander of the shelf and the second mini-SAS port on the BrickStor to port A on the bottom expander of the shelf.

There is a lights-out remote management Ethernet port (Intel RMM4 LAN) to the right of the USB ports when observing the rear of the BrickStor (labeled **MGMT)**. This specialized Ethernet port can be configured via the BIOS. To enter the BIOS press F2 during system boot.

1. Navigate to the **Server Management** tab and then scroll down to **BMC LAN Configuration**. Press "Enter".
2. Scroll down to Intel RMM4 LAN Configuration **IP Source.** And then select either **Static** or **Dynamic**. Set the IP as needed.
3. Select the User ID and Enable or disable the account as needed.
    a. By Default the User ID **root** with password **racktop** is configured.

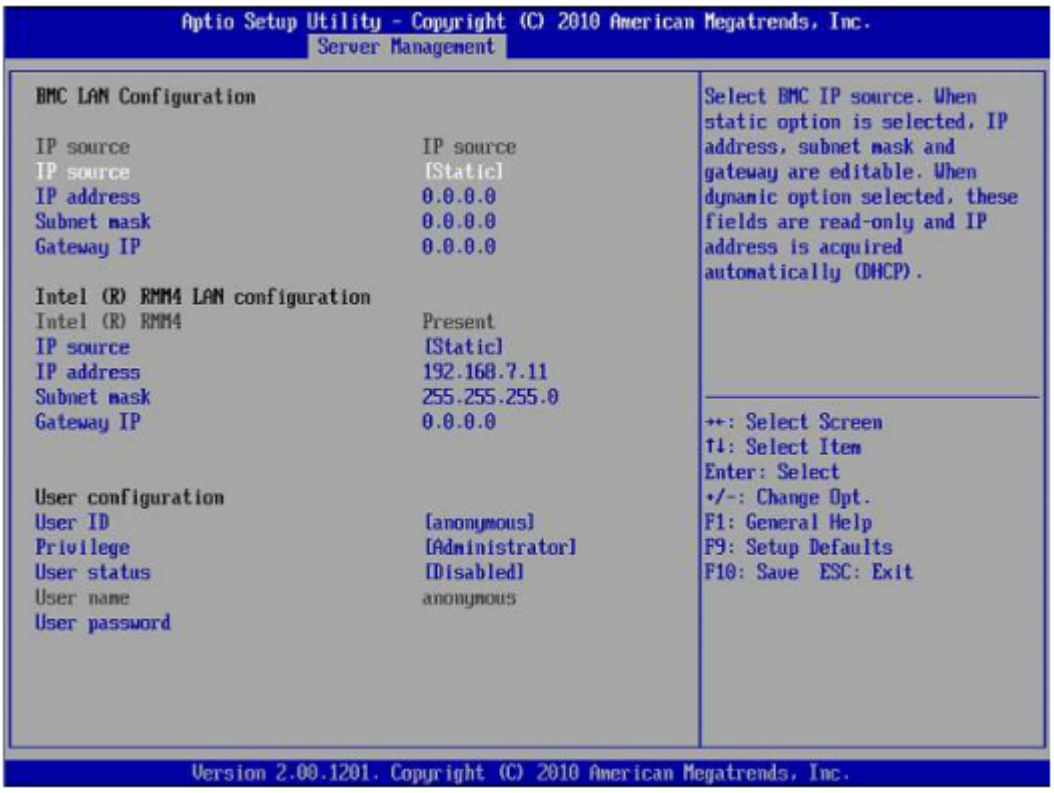



Figure 7 - RMM IP Address Settings

`

`

`

# BrickStor Quick Start

## *Default Accounts*

BrickStor ships with a default administrative account for configuring the system. The root account has system wide superuser permissions within BrickStorOS.

## *Default Passwords*

The default password for root account is "**racktop**". This password is well known and should be changed immediately.

## *Configuring Ethernet Address on Physical Interfaces*

Network configuration model of any BrickStor appliance is essentially identical and as follows. All systems have an even number, two or four onboard 1GbE or 10GbE RJ45 interfaces clustered together to the right of the power supplies. Onboard 10GbE, 25GbE, etc. fiber or copper interfaces are typically included with every system and either reside on one of the two riser cards or installed into a special interface on the system board, on the far-right edge of the chassis. Under normal circumstances the leftmost Ethernet interface is automatically configured for the purposes of management access, which includes '*ssh'* for some initial setup and for diagnostic as well as some feature configuration not currently accessible via the graphical management interface, as well as for access via the graphical management interface.

All physical interfaces are abstracted with one or multiple virtual interfaces, which do not typically share the MAC address of their underlying physical interface. Relationship of virtual interfaces to physical is many to one, meaning a single physical interface may possess one or multiple virtual interfaces. Multiple virtual interface over a single physical interface are common in VLAN tagging scenarios, which we discuss later in the document.

It is also possible to bond links into what we commonly refer to as an aggregate, more commonly known as a LAG (Link Aggregation Group), with or without LACP capability. Due to various complexities of configuration and the wide range of possible configurations we will not discuss the details of this configuration in this guide. If this is a requirement, please contact RackTop support for details.

Multiple virtual interfaces do not share a MAC address, instead each is assigned a randomly generated address, unless one is explicitly provided. We will not discuss the details of this customization here. If this is a requirement, please contact RackTop support for details.

`

The appliance ships with at least one already existing virtual interface named ‘admin0’, configured to automatically obtain an IP address via DHCP to ease initial configuration whenever possible. This is the *primary* management interface, meaning this is the interface used to manage the machine as an administrator, not meant for data traffic normally.

At least one *data* interface is required to expose files via one or more supported protocols: AFP, NFS and SMB. For all interfaces other than management, which typically will already exist and will not need to be created or re-created, use the naming convention ‘data**XX**’, where ‘**XX**’ is a non-negative numeric value starting with 0. All physical interfaces are suitable candidates, sans the first interface used for management as discussed previously. Needs vary, but a typical configuration will use 10GbE, 25GbE, and similar high bandwidth interfaces for all data access.

Virtual interfaces on BrickStorOS are configured via ‘dladm’, which must be created before a physical link can be used. After a system has been connected to network equipment, information about state of physical interfaces can be seen with a command in the following example. A typical output follows this general appearance:

```
# dladm show-phys
LINK         MEDIA                STATE      SPEED  DUPLEX    DEVICE
ixgbe0       Ethernet             down       0      unknown   ixgbe0
ixgbe1       Ethernet             down       0      unknown   ixgbe1
igb0         Ethernet             up         1000   full      igb0
igb1         Ethernet             unknown    0      half      igb1
igb2         Ethernet             unknown    0      half      igb2
igb3         Ethernet             unknown    0      half      igb3
```

In the above example it can be seen that link named ‘igb0’ is up and configured at 1GbE. This is the physical interface on which virtual interface ‘admin0’ is provisioned. Typically, data interfaces will follow the naming convention prescribed earlier and use high speed interfaces, commonly identified as `ixgbe**XX**`, where ‘**XX**’ is a non-negative numeric value starting with 0. Following is an example of Testablishing such a data interface over physical interface called ‘ixgbe0’.

```
# dladm create-vnic -l ixgbe0 data0
```

Once a virtual interface has been created, an IP address must be assigned to this interface. IP interfaces on BrickStorOS are configured via ‘ipadm’. The default ‘admin0’ IP interface cannot be modified since it is a temporary interface from an ipadm standpoint, instead it needs to be created persistently if a static IP address assignment is required. If you need to create a static IP, perform the following either via ssh, while connected via an IP address assigned to another interface, or directly via console of virtual console:

```
# ipadm delete-if admin0
# ipadm create-if admin0
# ipadm create-addr –T static –a local=x.x.x.x/24 admin0/v4
```

Where in the last command ‘x.x.x.x/24’ is the IP address/CIDR and ‘admin0/v4’ is the interface name and

`

IP version (4 or 6). Upon creation of an IP interface, two addresses are configured, IPv4 and IPv6. For all intents and purposes IPv6 interface should be ignored usually.

### VLAN Tagging

If VLAN tagging is setup on the port for trunking, you can create an interface like shown:

```
# dladm show-link
```

This will give you a list of available links for the next step, which is :

```
# dladm create-vlan -l ixgbe0 -v 10 vlan10
```

Replace ixgbe0 with an appropriate physical interface from your system and vlan10 with the name for your vlan. Note: vlan name must lead with a letter and also contain at least one number.

**Link Aggregation (Bonding)**

If link aggregation is required, first create an aggregate and then create a vnic on top of it:

```
# dladm create-aggr -l ixgbe0 –l ixgbe1 0
```

Where '0' denotes the number that will be placed in the name 'aggr0'. After that, create a vnic on top of the aggregate:

```
# dladm create-vnic –l aggr0 data0
```

## *Configuring Default Gateway*

If, in the previous steps, you have deleted an interface, you may not have a default gateway if the interface that was deleted was the only one on its subnet. You can find your default gateway by using:

```
# netstat -rn

Routing Table: IPv4
  Destination           Gateway           Flags  Ref     Use     Interface
-------------------- -------------------- ----- ----- ---------- ---------
default              10.1.12.254          UG        3       5761
10.1.12.0            10.1.12.196          U         7    2008782 admin0
127.0.0.1            127.0.0.1            UH        2         70 lo0
```

`

```
Routing Table: IPv6

  Destination/Mask            Gateway                   Flags Ref   Use    If
--------------------------- --------------------------- ----- --- ------- ---
--

::1                         ::1                         UH      2      10 lo0
```

From there, under the flags column you are looking for a ‘G’, which stands for gateway and a ‘default’ designation under the ‘Destination’ column. You can add a new permanent default route using the following:

```
# route –p add default x.x.x.x
```

Where x.x.x.x is your default gateway. You can now see your default route in ‘netstat –rn’

## *Time Zone Setup*

Set the timezone of BrickStor through the command line by editing the following file.

```
# tzselect
```

Then follow the prompts.

```
# tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
 1) Africa
 2) Americas
 3) Antarctica
 4) Arctic Ocean
 5) Asia
 6) Atlantic Ocean
 7) Australia
 8) Europe
 9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the POSIX TZ format.
#?
```

`

```
Please select a country or region.
 1) Anguilla                          28) Haiti
 2) Antigua & Barbuda                 29) Honduras
 3) Argentina                         30) Jamaica
 4) Aruba                             31) Martinique
 5) Bahamas                           32) Mexico
 6) Barbados                          33) Montserrat
 7) Belize                            34) Nicaragua
 8) Bolivia                           35) Panama
 9) Brazil                            36) Paraguay
10) Canada                            37) Peru
11) Caribbean NL                      38) Puerto Rico
12) Cayman Islands                    39) St Barthelemy
13) Chile                             40) St Kitts & Nevis
14) Colombia                          41) St Lucia
15) Costa Rica                        42) St Maarten (Dutch)
16) Cuba                              43) St Martin (French)
17) Curaçao                          44) St Pierre & Miquelon
18) Dominica                          45) St Vincent
19) Dominican Republic                46) Suriname
20) Ecuador                           47) Trinidad & Tobago
21) El Salvador                       48) Turks & Caicos Is
22) French Guiana                     49) United States
23) Greenland                         50) Uruguay
24) Grenada                           51) Venezuela
25) Guadeloupe                        52) Virgin Islands (UK)
26) Guatemala                         53) Virgin Islands (US)
27) Guyana
#?
```

`

```
Please select one of the following time zone regions.
 1) Eastern (most areas)
 2) Eastern - MI (most areas)
 3) Eastern - KY (Louisville area)
 4) Eastern - KY (Wayne)
 5) Eastern - IN (most areas)
 6) Eastern - IN (Da, Du, K, Mn)
 7) Eastern - IN (Pulaski)
 8) Eastern - IN (Crawford)
 9) Eastern - IN (Pike)
10) Eastern - IN (Switzerland)
11) Central (most areas)
12) Central - IN (Perry)
13) Central - IN (Starke)
14) Central - MI (Wisconsin border)
15) Central - ND (Oliver)
16) Central - ND (Morton rural)
17) Central - ND (Mercer)
18) Mountain (most areas)
19) Mountain - ID (south); OR (east)
20) MST - Arizona (except Navajo)
21) Pacific
22) Alaska (most areas)
23) Alaska - Juneau area
24) Alaska - Sitka area
25) Alaska - Annette Island
26) Alaska - Yakutat
27) Alaska (west)
28) -
29) Hawaii
#?
```

```
The following information has been given:

        United States
        Eastern (most areas)

Therefore TZ='America/New_York' will be used.
Local time is now:      Wed Nov 28 12:34:29 EST 2018
Universal Time is now:  Wed Nov 28 17:34:29 UTC 2018
Is the above information OK?
1) Yes
2) No
#? Yes
Please enter 1 for Yes, or 2 for No.
#? 1
Reboot to switch from current timezone to America/New_York
# _
```

A reboot is required for the changes to take effect.

## *NTP Setup*

### Preparing to Setup and Sync Time

First disable the NTP service so that you can synchronize time for the system to the NTP server. By default, the NTP service is configured to get time from the pool.ntp.org service.

You can enable from the command line or the GUI. To enable by command line:

`

```
# svcadm disable ntp
```

Next run the *'ntpdate'* command to synchronize time. This should show a current offset .

*Note: ntp service must be disabled for ntpdate to work*

```
# ntpdate <IP of Time Server>
```

If the offset was very large you can run the ntpdate command again to verify that clock was adjusted accordingly and offset now should be very small.

**Example:**

```
# ntpdate pool.ntp.org

10 Sep 08:30:08 ntpdate[7063]: step time server 129.6.15.28 offset -
17971.406299 sec

# ntpdate pool.ntp.org

10 Sep 08:30:31 ntpdate[7064]: adjust time server 129.6.15.29 offset 0.002656
sec
```

*Problems with SMB authentication or AD join may be related to BrickStor's time being 5 minutes or more out of sync with Active Directory time.*

## *Hosts Entries*

### Setting up hosts entries

Most of the time this should not be necessary, but in the exceptional cases where host name resolution is required and cannot be accomplished via DNS, static entries may be added to allow for local resolution. This activity is accomplished via *'bsradm'* as follows, where *'192.168.0.1'* is the address and *'othernode'* is name resolving to this address:

```
# bsradm hosts add --ip 192.168.0.1 --names othernode
```

*Note: this may be a required step if DNS is not setup and you are connecting to an NFS datastore from ESXi.*

`

## *RMM (Remote Terminal) IP Address*

Your BrickStor storage appliance comes equipped with a Remote Management Module frequently abbreviated to RMM. RackTop recommends connecting this Ethernet interface as well as the *'admin0'* management Ethernet interface to a dedicated management network, if one is available. Separation of management and administration concerns from data access is a recommended best practice. This enables you to access the appliance as if you were standing in front of it with a crash cart or KVM, even when services such as SSH are down. You can use RMM to power cycle the machine or see and use the console. If RMM is already configured, you can find the IP address with this command from the terminal:

```
# bsradm hw rmm

IpSource: DHCP Address

IpAddress: 192.168.0.101

SubnetMask: 255.255.255.0

MacAddress: 00:1e:67:50:c7:c1

SnmpCommunityString: public

DefaultGateway: 192.168.0.1

Vlan: 0
```

Once you have the IP address, you can login to RMM via your browser. You will need to use Java to access the console and this will most likely require adding a security exception for the IP address in the Java control panel.

## *Creating Local Accounts*

As root you can create local accounts that can be used for controlled access to shares as well as providing access to administrative functions such as the ability to manage BrickStor with the myRack Manager.

```
#useradd <username>

To set the user's password:

#passwd <username>
```

## *Add Local Accounts to Bsradmins Group*

To allow a given local account administrative access of a BrickStor appliance via myRack Manager, this account must be in the *'bsradmins'* group of the appliance. To add a user to the group, run the following command, replacing username placeholder with actual local account name:

```
# usermod -G bsradmins <username>
```

`

## *Adding and removing e-mail addresses from Notification List*

To add e-mail addresses to receive notifications from the BrickStor appliance, use the following command format at the terminal:

```
# bsradm notify add <email address> –all
```

Other options besides the “all” notifications options are:

```
--system     Add to system notification list

--reports    Add to reports notification list

--faults     Add to faults notification list
```

To list users and their notification types, use:

```
# bsradm notify show
```

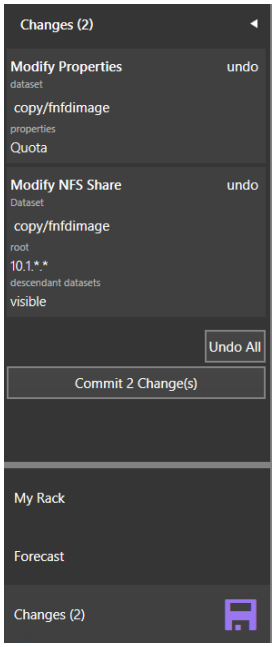
And to remove users from their notification, use:

```
# bsradm notify remove <email address> --all
```

`

## *General Conventions for BrickStor*

### Using the GUI




The GUI is designed to allow you to make changes and see the results of your changes before they are committed. You can make a series of changes in any order and the changes will appear in the change on the left side of the GUI.

The system will automatically order the changes to optimally commit the changes to the system. You can undo all changes with the Undo All button or specific changes by clicking undo on individual changes. The Purple Icon on the bottom right denotes there are changes pending. Additionally, a “Changes” watermark will appear in the main management area of the GUI.

The system will force you to acknowledge change that will destroy or take shares and volumes offline.

### Data Layout

The Pool is the top level for data resources and represents a collection of physical devices (Write Cache, Read Cache, and data disks). A storage profile is applied to the pool. The pool has a 10% data reservation with a maximum size of 100GB. This ensures that the pool doesn’t unexpectedly run out of capacity. This reservation can be removed or resized to enable more space for data. But at this time the admin can procure more capacity or remove data to free space before it is too late and the system is physically out of space.

An appropriate container should be created for different types of data under the pool. All data sets should be stored under the appropriate container. Global for global datasets shared as a file protocol and vbd for all volumes shared via a block protocol within the global container.

Format:

File Protocol Share <pool name>/global/<data set name>

Block Protocol Share <pool name>/global/vbd/<LUN Name>

Containers are created in a similar way to a new dataset but a container is chosen as the type.

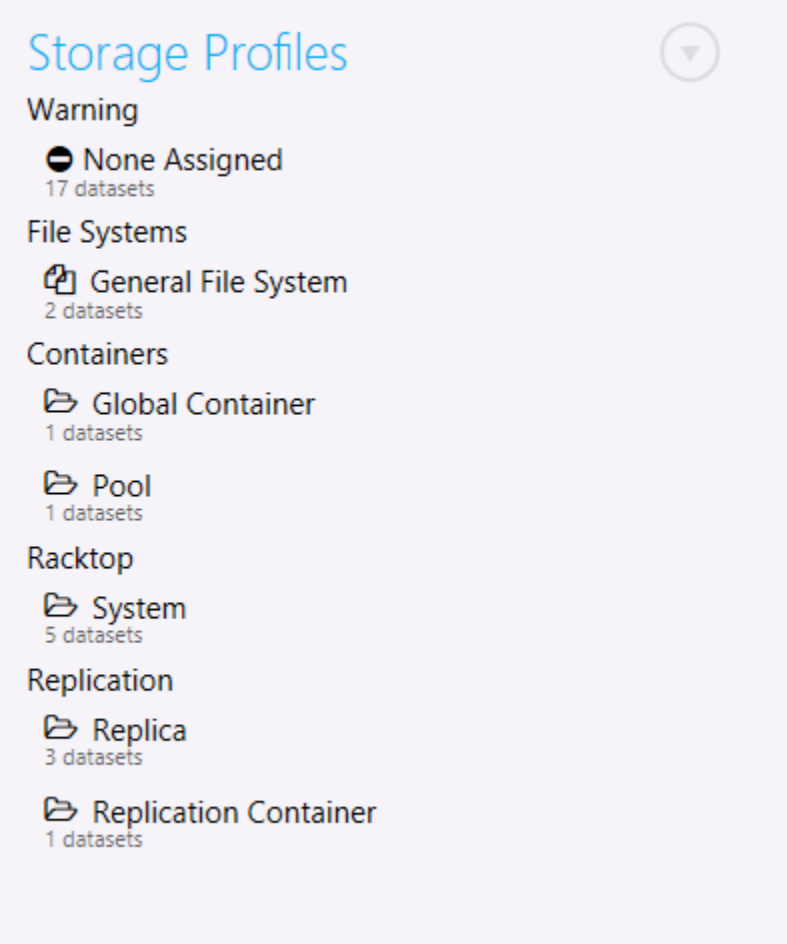
### Data Protection Policies

Data protection policies determine the frequency and replication schedule for datasets and volumes. Each storage profile as an associated protection policy. However, the system allows users to set custom protection policies for each data set or volume as well. Data Protection policies should be used for any type of scheduled data protection to achieve business driven Recovery Point Objectives (RPO).

`

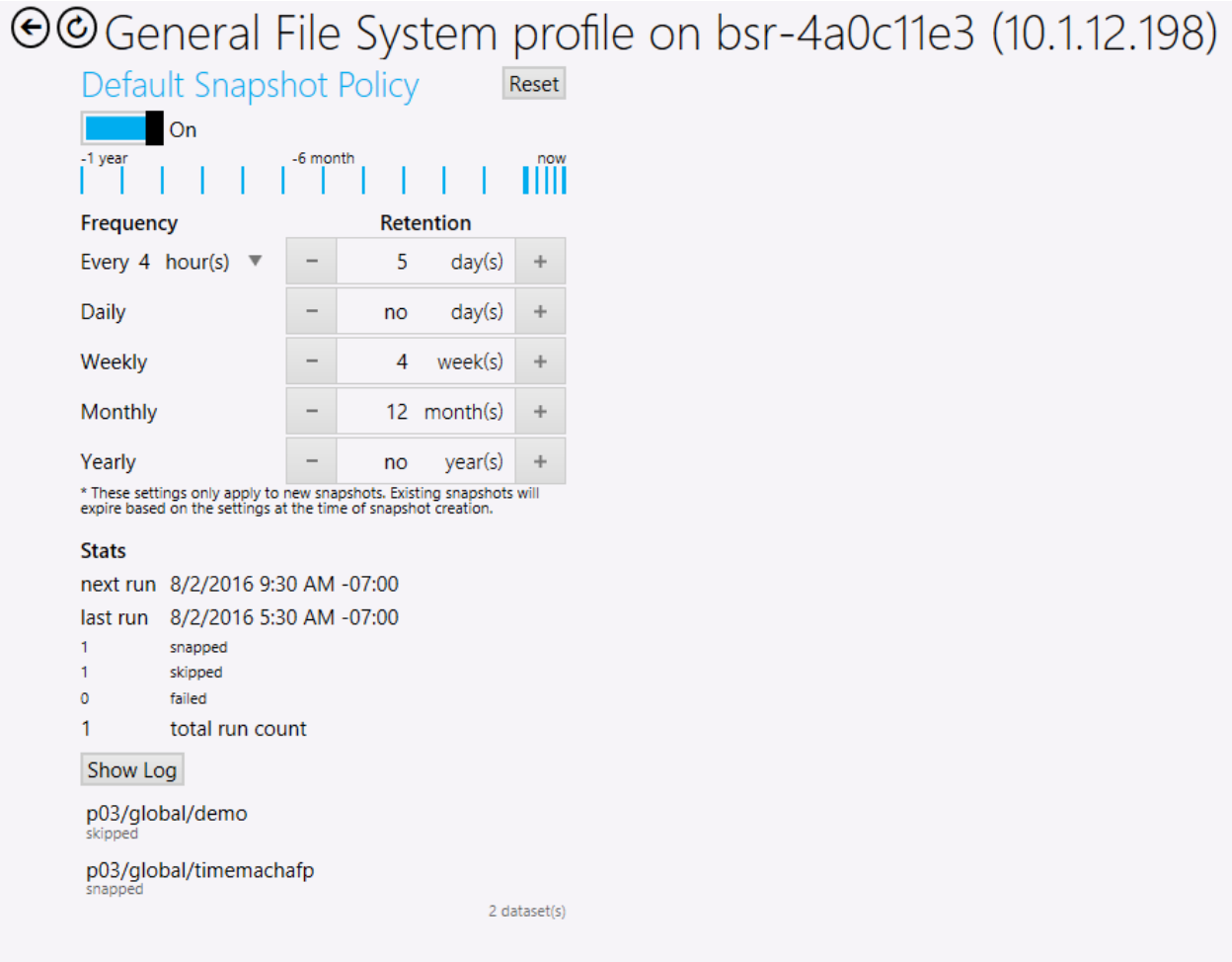
### *Changing the default Snapshot Frequency and Retention for a Storage Profiles Data Protection Policy*

From the main page of the myRack manager you can change the default policy for a storage profiles data protection policy. It will apply to all snapshots going forward. It will also show you from the screen what datasets have that storage profile applied and will be affected.




Once you click on the profile you would like to edit you can change the policy.

`



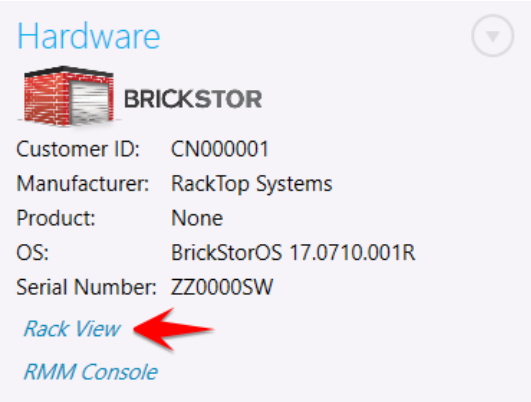

## *Rack View*

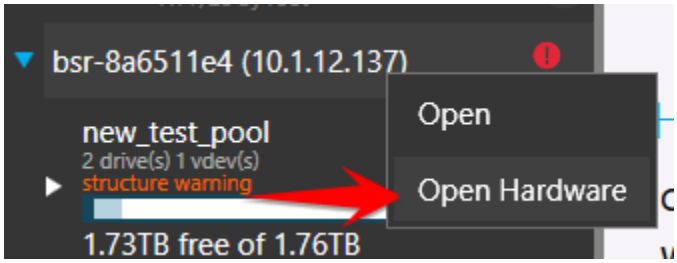
myRack Manager features the capability to easily view and modify your appliance hardware called Rack View. Rack View allows users to add or modify pools and vdevs, and gives visuals that allow users to see what changes will occur to the system's hardware prior to committing them.

### Accessing Rack View

To access Rack View, simply click the Rack View button under the Hardware section of a system or right click the system from the My Rack tab and select Open Hardware.

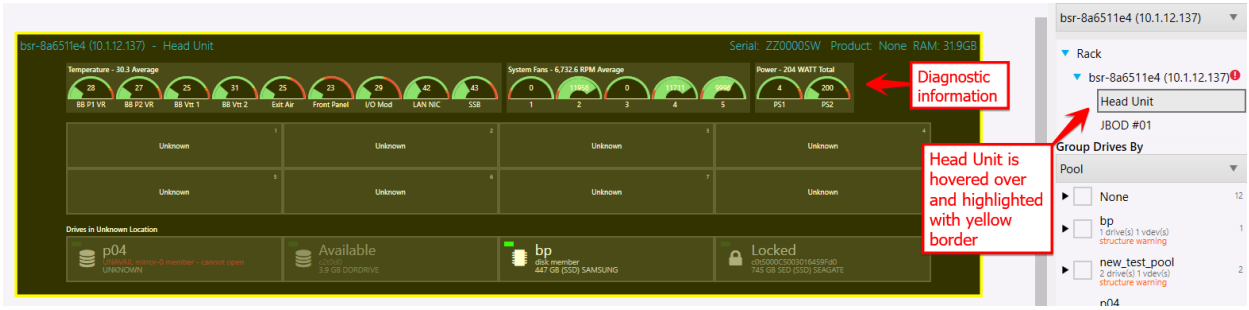
`



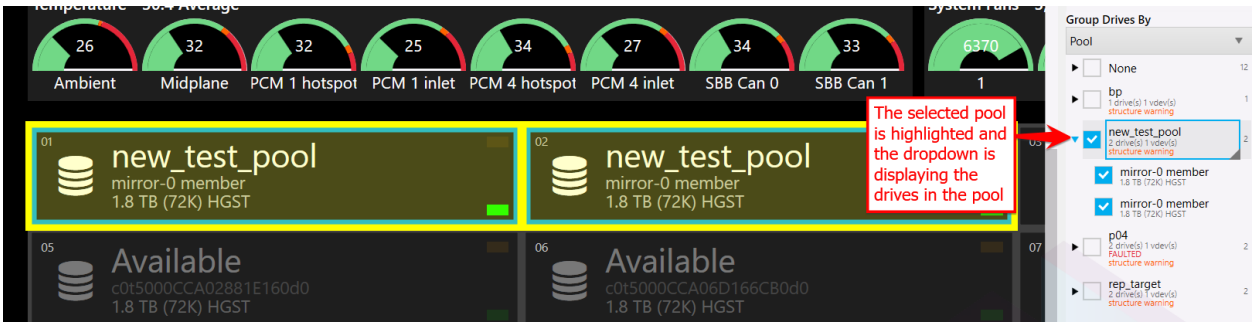




## The Rack View Interface

Rack View will display the current hardware that is on your system including the head unit, JBODs, and any drives that are within these appliances. It will also display various diagnostic information such as the values of temperature sensors in the system and the fan speeds. On the upper right hand side you can select which appliance you want to zoom to. The appliance will be highlighted in yellow when the mouse is hovered over it and left clicking will zoom to the appliance.



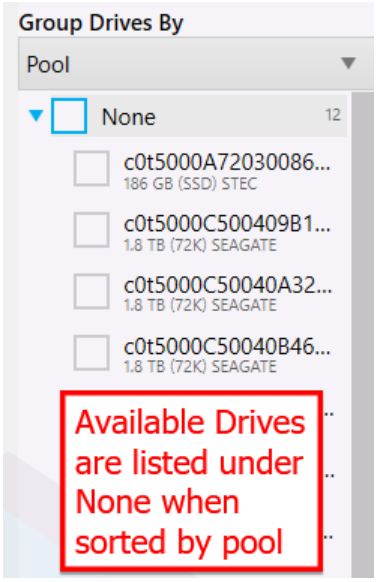

The right-hand side of Rack View also allows you to group the drives in the appliances based on certain properties such as pool, make, and vdev type. To change the grouping type, select the dropdown under Group Drives By and then select how you want to group them. When hovering over one of these groups, affiliated drives will be highlighted and left clicking will zoom to the drives. You can also expand these groups with the arrow and select individual drives that are a part of the group.

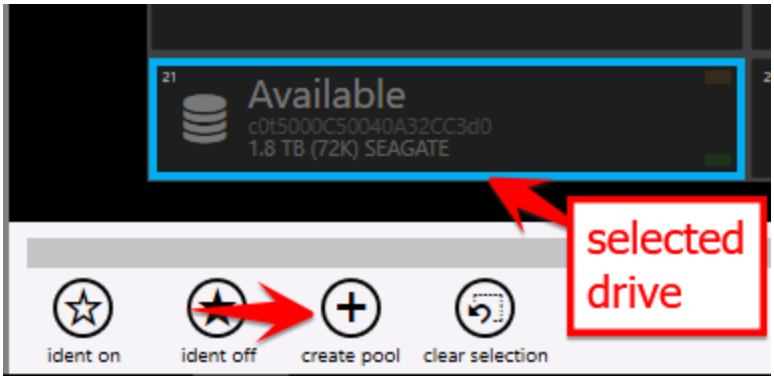


`

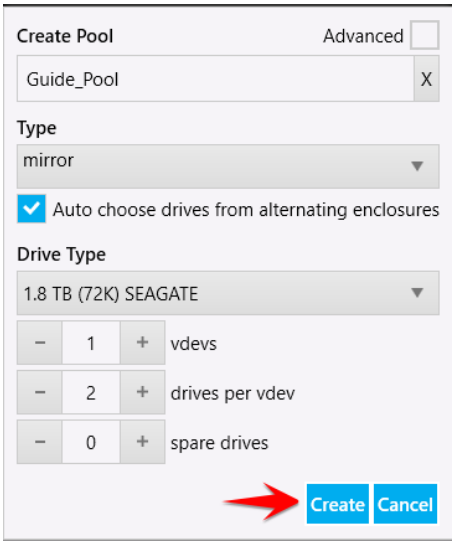
## Creating a Pool within the Rack View

To create a pool, first select an available drive by left clicking a drive labeled as Available or by selecting a drive from the right hand dropdown of None when sorted by Pool. The selected drive will have a blue border and the icon create pool will appear at the bottom of the screen.



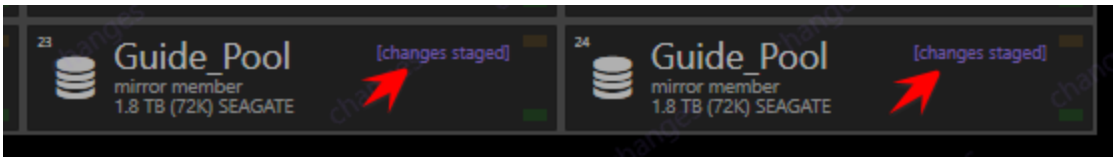




Clicking create pool will open the Create Pool dialog box where you must set a pool name and can change the type of vdev, the number of vdevs, how many drives are in each vdev, as well as how many spares you want the pool to have. By default it will choose drives from alternating enclosures but you can uncheck this box to select specific drives for the pool. When everything is configured, click create to queue the changes.




Rack View will display the queued changes and any pool that will be affected by changes will have the [changes staged] indicator on it.

`




To finalize the creation of the pool, go to the Changes tab on the left hand side of myRack Manager and click Commit Change(s).

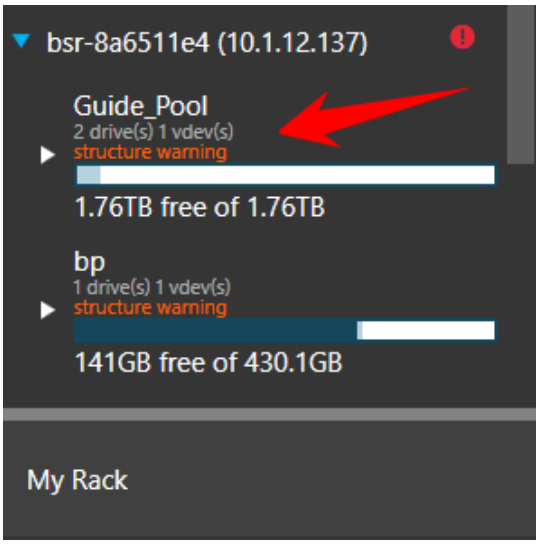
## Modifying an Existing Pool

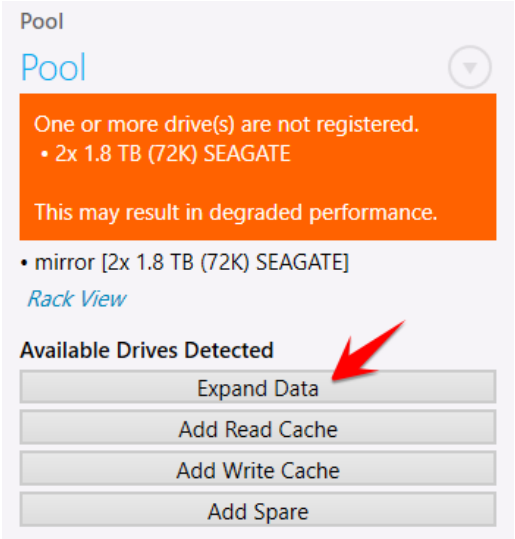
myRack Manager features a number of ways to modify pools that are currently on the system.

### *Expanding a Pool*

To expand a pool, either select the pool in Rack View and click the expand pool button at the bottom of the screen,

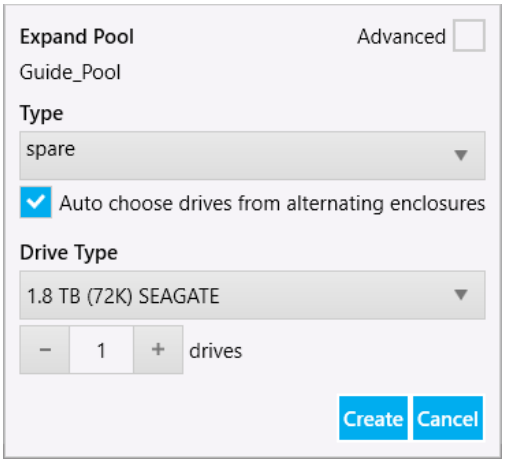

Or select the pool from the My Rack tab on the left-hand side of myRack Manager and click either the Expand Data, Add Read Cache, Add Write Cache, or Add Spare button under the Pool heading, depending on what you would like to add to expand the pool (will only appear if the correct types of drives are available).

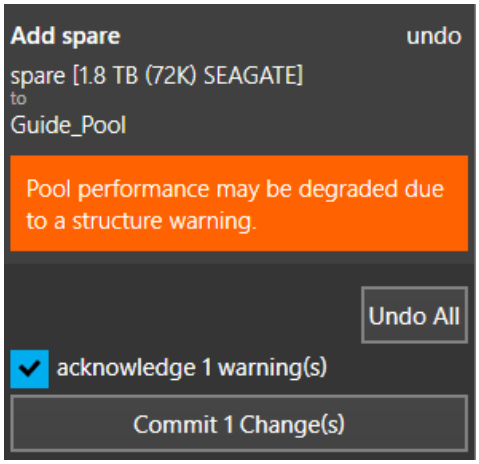


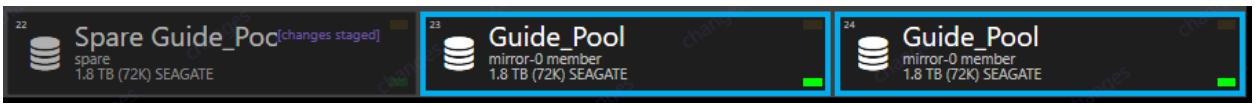




This will bring up the Expand Pool dialog box where you can choose to expand the pool by adding more vdevs, read and write caches, or spares. When the desired setting have been configured, click create to queue the change.

`



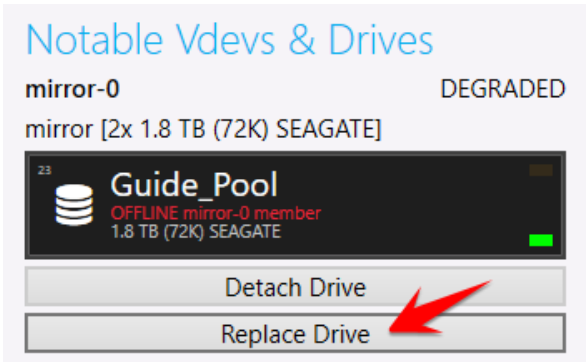




All changes in the queue will be indicated in Rack View and must be committed using the changes tab on the left hand side of myRack Manager.

### *Replacing a Drive*

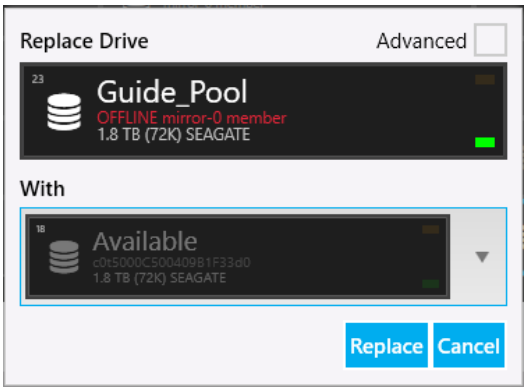
If a drive becomes disabled or faulted it may be necessary to replace the drive with another available drive in the system. Select the drive you wish to replace in Rack View and click the replace button at the bottom of the screen.


Or, if the drive is offline, you can navigate to the degraded pool in the My Rack tab on the left hand side of the screen and click the Replace Drive button under the Notable Vdevs and Drives heading.



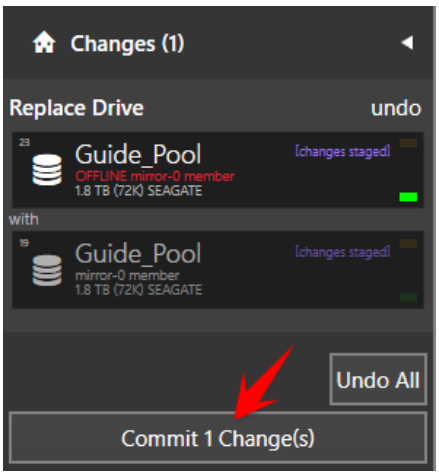

This will bring up the Replace Drive dialog box where you can select the drive to use as the replacement then click the Replace button to queue the change.

`



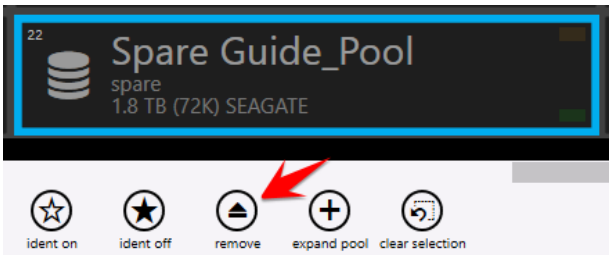


The change will be indicated in Rack View and will not be committed until the Commit Changes button is clicked on the Changes tab.



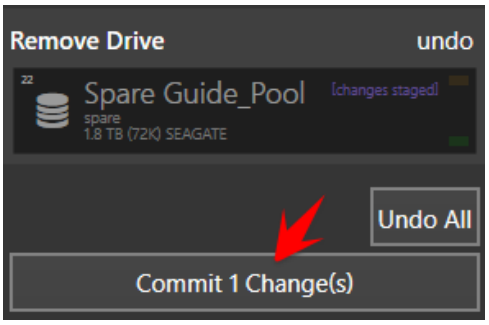

### *Removing a Spare*

If a pool has a spare drive that no longer requires one, it can be removed to free up the drive by selecting the spare in the Rack View and clicking the remove button.




The change will be indicated in Rack View and will not be committed until you click the Commit Changes button in the Changes tab on the left-hand side.

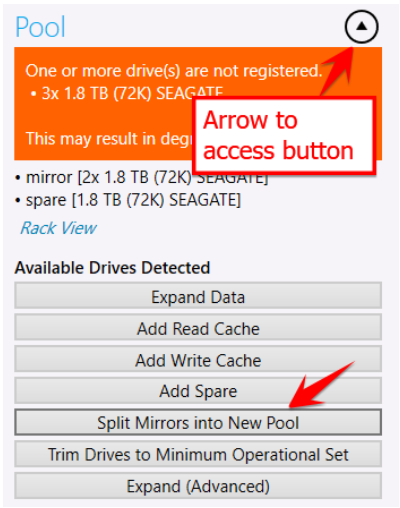
`

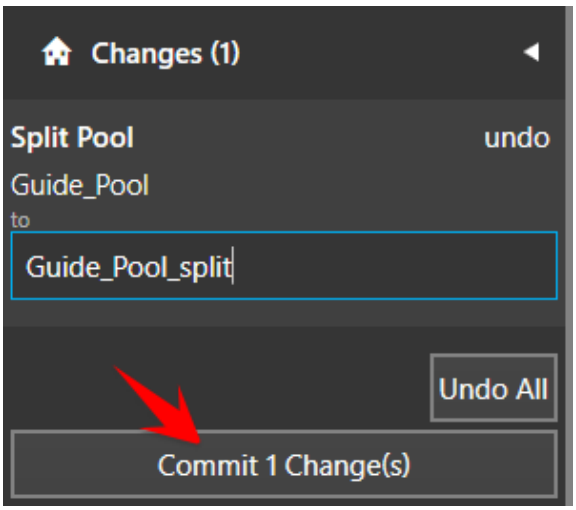



## *Splitting a Mirrored Pool*

A pool consisting of mirror vdevs can be split into two pools with no redundancy that contain the same data. **Note that this is only recommended in certain scenarios as the lack of redundancy increases the risk of data loss.** To split a mirrored pool, select the pool on Rack View and click the split pool button,


Or navigate to the pool from the My Rack tab on the left hand side and click the Split Mirrors into New Pool button under the Pool heading (you will need to click the arrow button to the right of the Pool heading to access this).



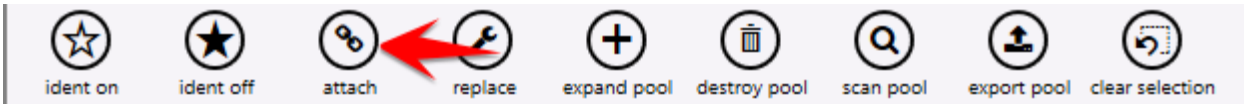




From the changes tab on the left hand side you can change the name of the new pool that will result from the split and commit the changes with the Commit Changes button (by default the new pool created this way will be exported).

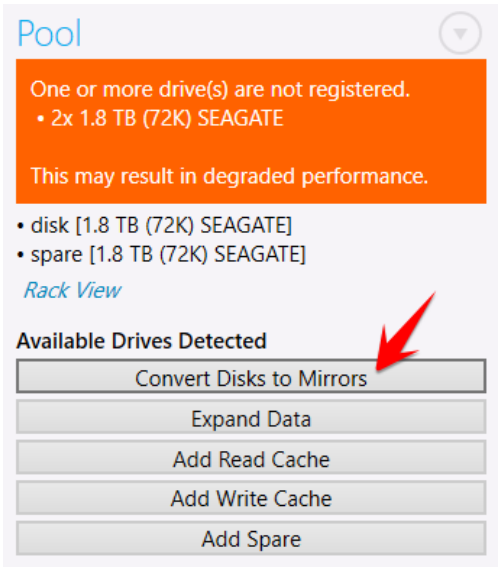
## *Attaching a Drive to a Pool*

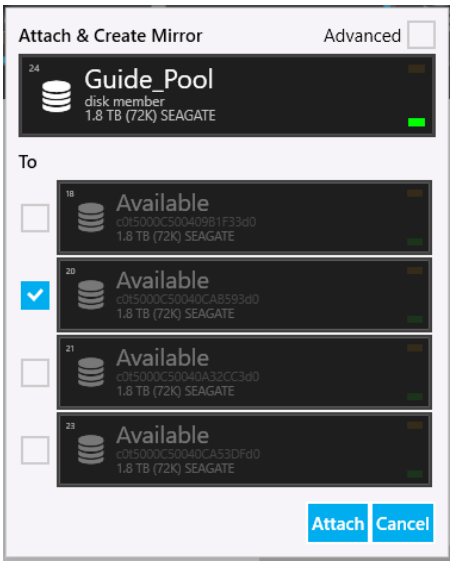
A pool with no redundancy can be converted to a mirrored pool, if there are enough available drives, in order to reduce the risk of data loss. To do this, select the pool in Rack View and click the Attach button,

`

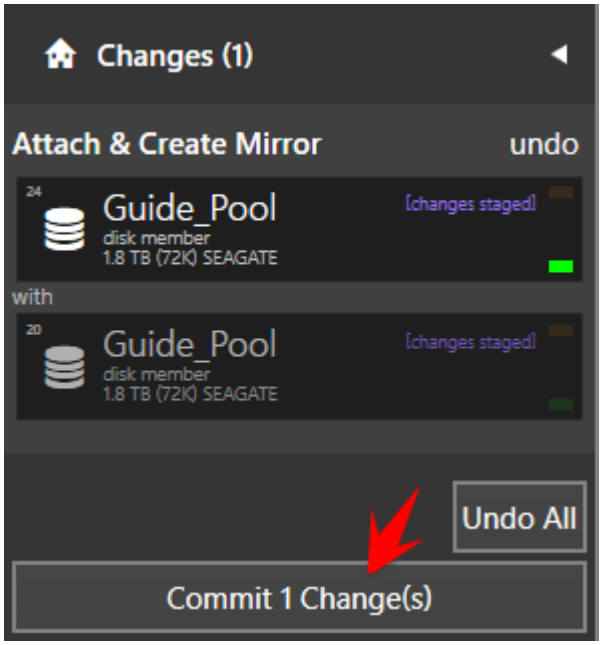


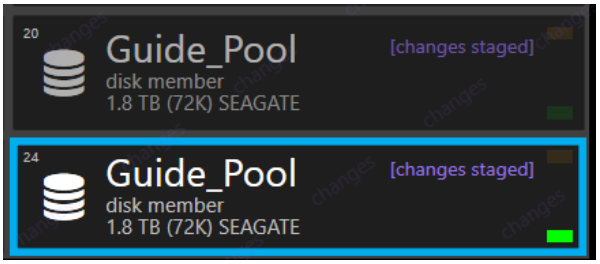

Or navigate to the pool from the My Rack tab on the left hand side and click the Convert Disks to Mirrors button under the Pool heading.




If done through Rack View, you will need to select the drive to attach yourself. When done through the pool's page it will select a drive for you automatically. The change will be indicated in Rack View and will not be committed until you click the Commit Changes button in the Changes tab on the left hand side.



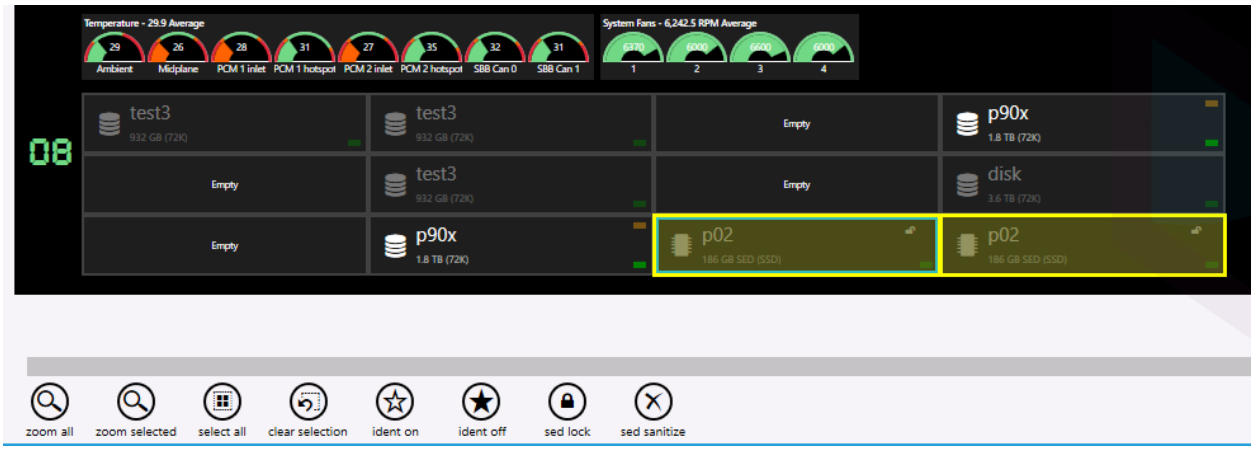



`

### *Locking and Unlocking TCG Drives*

Rack View allows you to lock and unlock drives in an exported pool. You can observe whether a drive is locked or unlocked by the lock indicator on the top right corner of the drive in rack view. This drive is unlocked and has the open lock icon.

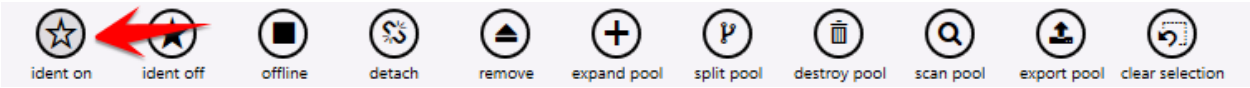

Drives will automatically lock when power is removed from the drive. Using the buttons in the bottom of the Rack View you can lock and sanitize drives.




When you sanitize a drive using the button in the GUI it is changing the data encryption key (DEK) for the data band. This means that data on the band is no longer readable because there is no copy of the DEK. This erase method meets NIST 800-88 purge standards.

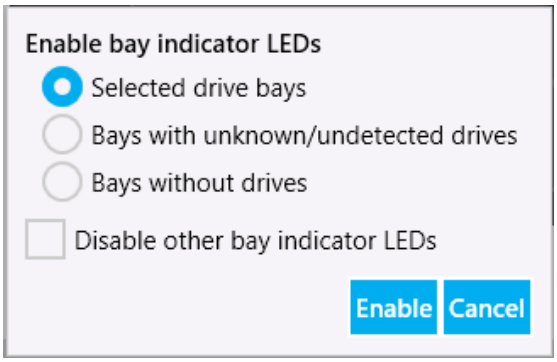
### *Toggling Identifying Lights*

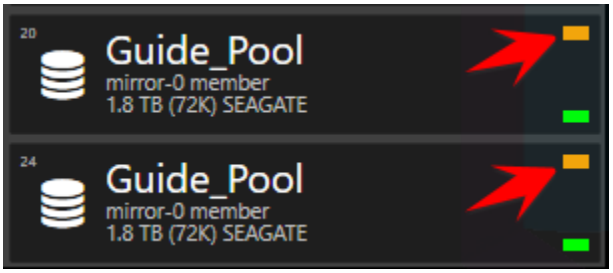
Rack View allows you to toggle a physical indicating light on each drive to assist with identifying the correct drives on the machine. You can either select one drive by clicking directly on in it Rack View, or multiple drives using the Group Drives By interface on the right hand side. Once the appropriate drives have been selected click the ident on button at the bottom of the screen.




This will bring up the Enable bay indicator LEDs dialog box, where you can turn on the lights for either the selected bays, bays with unknown drives, or bays without drives. You can also choose to disable all other indicator lights to ensure only the desired drives have their lights enabled.

`

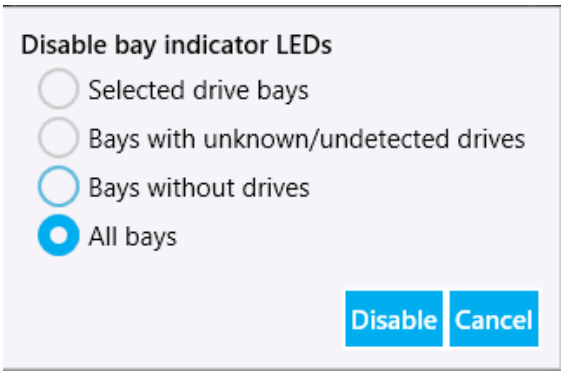



Drives with their indicating LEDs enabled will have a blinking orange indicator on Rack View as well as on the physical drive on the appliance.

To disable the identifying lights, select the desired drives like before and click the ident off button.

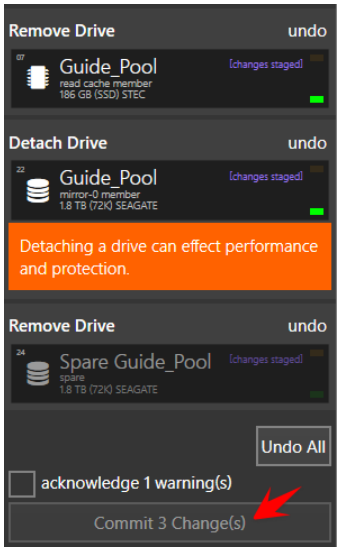

This will bring up the Disable bay indicator LEDs dialog box where you can turn off the lights on either the selected bays, bays with unknown drives, bays without drives, or all bays in general.




### *Trimming a Pool*

If a pool is going to be retired or is no longer necessary and to be removed, it can be trimmed to the minimum operational set of drives. **This will remove all redundancy and additional data protection and should only be done in specific scenarios.** To trim a pool, navigate to the pool from the My Rack tab on the left hand side and click the Trim Drives to Minimum Operational Set button under the Pool heading (you will need to click the arrow button to the right of the Pool heading to access this).

`

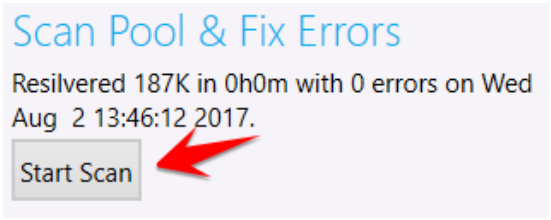



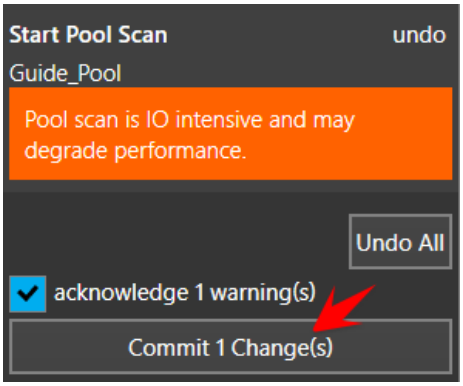

The steps it will take to trim the pool will be listed in the changes tab on the left hand side and no changes will take effect until the Commit Changes button is clicked.

## Scanning and Repairing a Pool

A pool can be checked for faults or problems and corrected using the scan pool feature. To scan a pool for potential faults, either select the pool in Rack View and click the scan pool button at the bottom of the screen.


Or navigate to the pool from the My Rack tab on the left hand side and click the Start Scan button under the Scan Pool & Fix Errors heading.



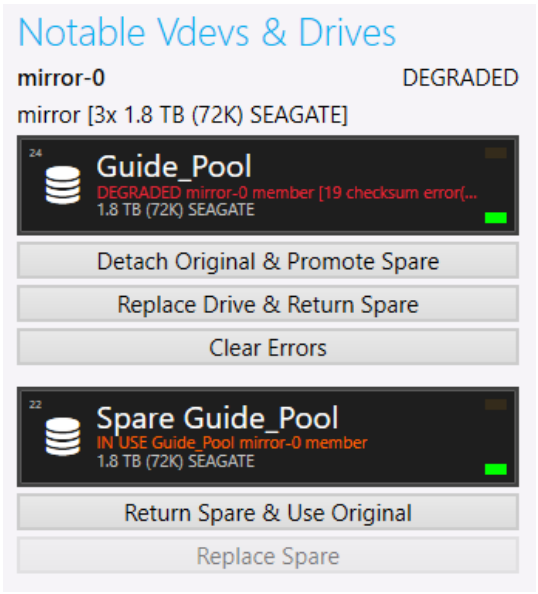



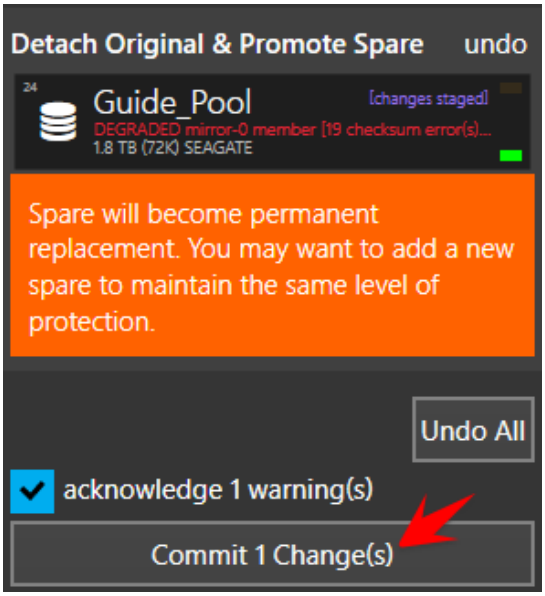

The scan will not be started until you click the Commit Changes button in the Changes tab on the left-hand side.

If the scan detects a faulty drive in the pool, it will mark the drive as degraded and replace it with a spare drive if one is available.

`


From the pool's screen on the My Rack tab, the faulted drive will appear under Notable Vdevs & Drives. You can choose to promote the spare drive and detach the faulted drive from the pool, replace the faulted drive with another available drive on the system and return the spare to being a spare for the pool, or you can clear the errors on the drive if the problem has been corrected and return the spare. These options can also be found at the bottom of the screen in Rack View.


Each of these changes will require you to click the Commit Changes button in the Changes tab on the left-hand side to complete the action.

## *Starting Services*

### NFS File Share

If you are going to be using NFS, enable the NFS server from the GUI or the command line.

In the GUI check the enable box next to NFS under Services in the appliance page of the GUI.

```
# svcadm enable nfs/server
```

Then check that the NFS server is online:

```
# svcs nfs/server

STATE          STIME    FMRI

online         13:27:51 svc:/network/nfs/server:default
```

`

## iSCSI Target

If you are going to be using iSCSI,  enable the iSCSI services from the GUI or the command line:

```
# svcadm enable stmf

# svcadm enable iscsi/target
```

Then check that they are online:

```
# svcs stmf iscsi/target

STATE          STIME    FMRI

online         14:07:34 svc:/system/stmf:default

online         14:07:34 svc:/network/iscsi/target:default
```

## SMB File Share

If you are going to be using SMB, enable the SMB server from the GUI or the command line.

In the GUI check the enable box next to SMB under Services in the appliance page of the GUI.

```
# svcadm enable idmap

# svcadm enable smb/server

# svcadm enable security/ktkt_warn
```

Then check that the services are online:

```
# svcs smb/server idmap ktkt_warn

STATE          STIME    FMRI

online         16:08:26 svc:/system/idmap:default

online         16:08:26 svc:/network/smb/server:default

online         16:08:26 svc:/network/security/ktkt_warn:default
```

`

## *SMB/CIFS Share Configuration*

## *Joining Active Directory*

The first step for making a CIFS share available for users is to join Active Directory, which requires several configuration steps before joining the domain will be possible. A machine account will be created for a BrickStor upon successful domain join operation. This machine account will enable users to passthrough authenticate and be either permitted or denied access to shares without requiring separate authentication against the BrickStor. In other words, once users are logged into Active Directory, their authentication information is stored on their system and in Active Directory, and no further authnetication prompts are necessary in order to access shares on a domain-joined BrickStor.

Active Directory requires certain attributes of name resolution, which usually means the BrickStor must be configured to resolve names against domain in the given instance of Active Directory to which it will be bound. BrickStor's domain setting must also be set to name of domain being joined.

First, validate what is currently configured, because no change may be necessary. Check currently configured domain with the following command:

```
# bsradm dns domain get
```

If the value reported is correct, that is, it matches the Active Directory domain name, no change is necessary. If however a modification is necessary, change should be made with the following command, replacing placeholder 'domain.tld' with actual fully qualified Active Directory domain name:

```
# bsradm dns domain set <domain.tld>
```

Next, confirm that correct DNS resolvers are configured, and if not, make necessary changes. In most environments at least two DNS servers will be configured and BrickStor must point to these resolvers, which in typical Active Directory configurations will be domain controllers also, or commonly member servers with a dedicated DNS function.

First, validate what is currently configured, because no change may be necessary. Check currently configured domain name resolution servers with the following command:

```
# bsradm dns ns show
```

If values reported are correct, no further resolver changes should be necessary. If however DNS servers need changing, use the following commands to add/remove entries, replacing placeholder 'address' with IP address of system being added or removed.

```
# bsradm dns ns add <address>
```

```
# bsradm dns ns remove <address>
```

*Note: NTP must be correctly configured with accurate synchronized timing with the Domain Controller before you can join the Domain Successfully*

`

The command for joining the storage appliance to the domain is:

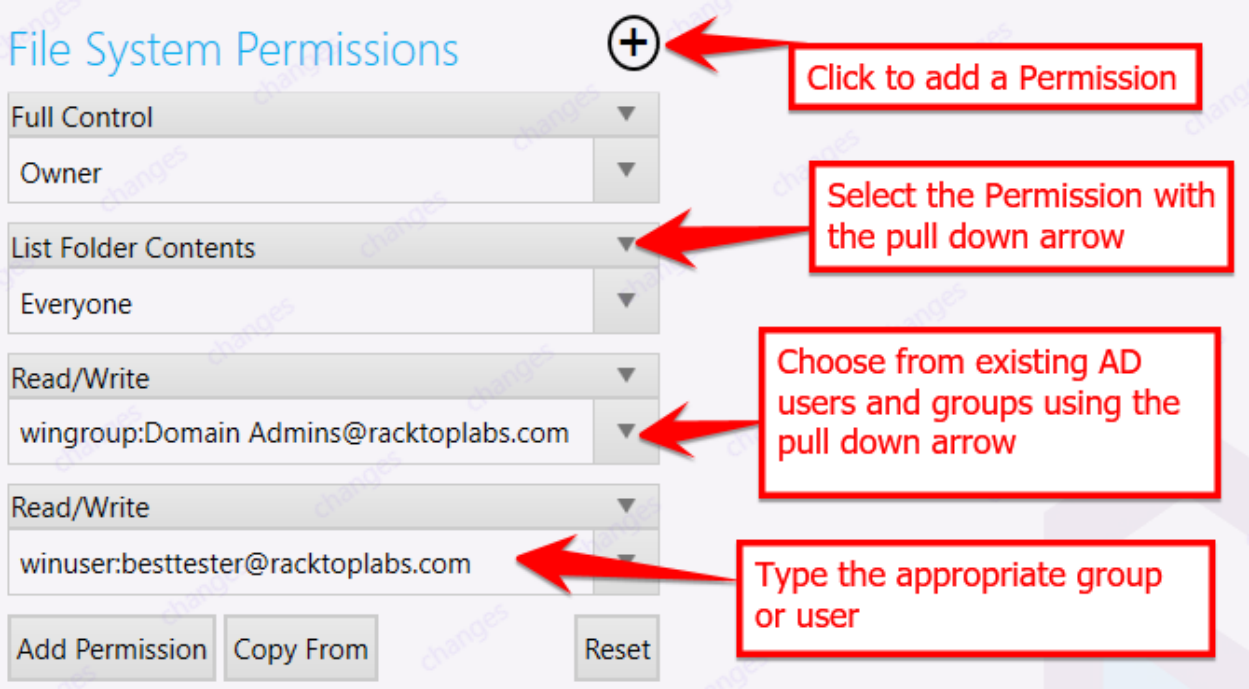
```
# smbadm join -y -u Administrator domain.tld
```

Where ‘Administrator’ is the name of the user you want to use to join the domain. This account is only used to create the computer object and does not need to be a service account. You will be prompted for a password. If the join fails, please double check your username and password and the settings in /etc/resolv.conf.

`

## Share Permissions

The GUI is designed to handle permissions management and simplify settings. It allows you to copy the permissions settings of another dataset as well as assign ACL groups and users.




**The appropriate format for adding a Windows AD user is**

winuser:<fully qualified name>

Ex: winuser:btest@racktoplabs.com

**The appropriate format for adding a Windows AD group is**

wingroup:<group name@domain>

Ex: wingroup:All Domain Users@racktoplabs.com

**The appropriate format for adding a local user is**

unixuser:<username>

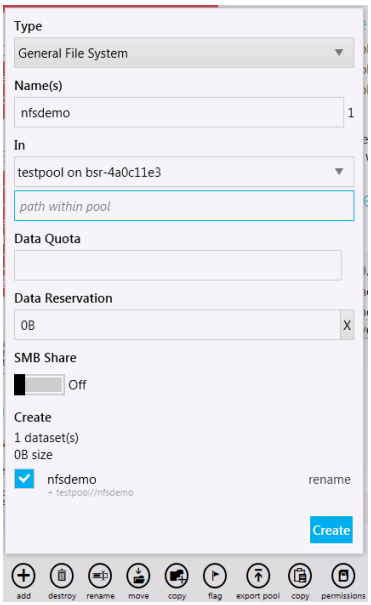
Ex: unixuser:winadm

`

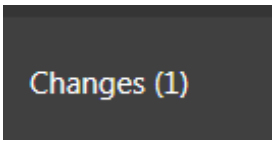
## *NFS Share Configuration*

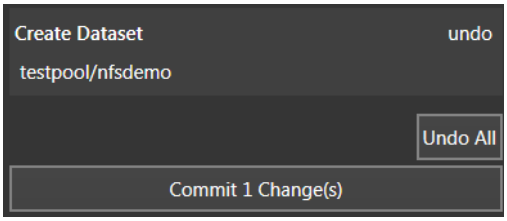
### Creating an NFS dataset (using MyRack Manager)

The first step in the GUI to create a NFS share is using the “Add” button near the bottom of the interface. This button will prompt you for information on the name of the share, location of the share (which pool, which existing dataset if that applies), and Data Quota and Reservation size.

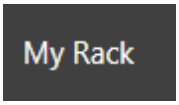



Once you click the “Create” button here, you need to click “Changes” to review and apply the changes as seen below:

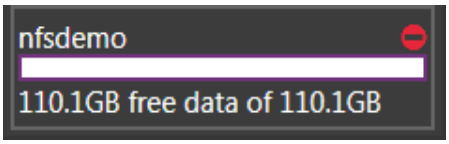


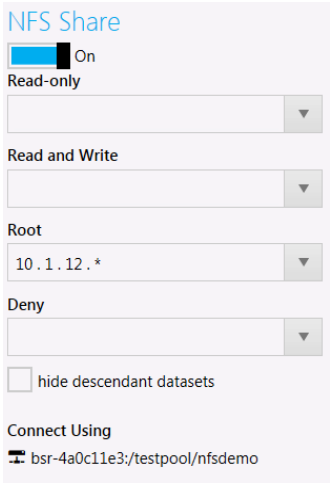

Once this is done, click on “My Rack” in the left pane in order to get back to your new dataset. The dataset is not yet shared via NFS, so we must click on the dataset and set properties as it applies to us.

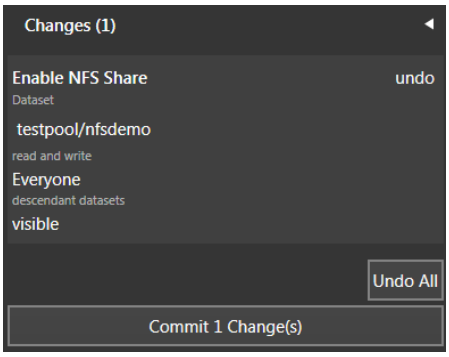


`




In the dataset properties on the right side of the MyRack Manager GUI, look for the “NFS Share” properties and make changes as required here. The first required change is to toggle the slider on the top to “On”. The ‘Connect Using’ shows you the path you can use to mount the dataset from clients.




## Creating an NFS dataset (command line)

The first step in creating an NFS share is creating the dataset. You can see a list of datasets you currently have on your system with:

```
# zfs list
```

If you want to see a list of datasets that use NFS already and their properties, use:

```
# zfs get sharenfs
```

The third column over will tell you the properties of each dataset’s NFS share properties. By default, this will be ‘off’. When you set a dataset to be shared, you may use something like this:

`

```
# zfs set sharenfs="sec=sys,rw=*,root=@10.1.11.0/24" storage/global/exampledataset
```

In order to restrict root access by IP and allow read and write access. Once you have set this option, your share will be available from the client(s) mounting the dataset, assuming the NFS daemon is enabled on the BrickStor appliance. The available options for the share are:

**share=***name***=***sharename*

Identifies an NFS or SMB share name. Maximum length of a share name is 80 characters.

**path=***pathname*

Identifies the physical path of the dataset to be shared or a subdirectory within the dataset to be shared.

**prot=smb | nfs**

Identifies the NFS or SMB file sharing protocol.

The following share properties are optional:

**desc=***description*

Identifies a text string that describes the share resource. Spaces or commas in the description must be enclosed in quotation marks (“ “).

**ro= | rw=**

Identifies whether the share is available as read/write or read-only to all clients. You can also specify a colon-separated list that includes hostnames, IP addresses, or a netgroup.

**root=**

Identifies a root user from a specified host or list of hosts that have root access. By default, no host has root access.

**sec=**

Identifies an NFS server security mode, such as sys, dh, krb5, and so on. For supported security mode information, see nfssec(5).

## *iSCSI Share Configuration*

### Creating a Default Target and Target Portal Group

Create a target portal group to restrict the target to your data0 (data, not management) IP address:

```
# itadm create-tpg global <x.x.x.x>
```

Where <x.x.x.x> is the IP address associated with data0. Next, we need to create the default target. To create the target, type the following:

```
# itadm create-target
```

Now check the status of your targets to make sure everything is okay:

`

```
# itadm list-target –v

TARGET NAME                                                     STATE          SESSIONS
iqn.2010-03.com.racktopsystems:02:c434c8d7-5643-6364-af5d-cb0bae33d531   online         0
      alias:            -
      auth:             none (defaults)
      targetchapuser:       -
      targetchapsecret:     unset
      tpg-tags:         default
```

Next, modify your target to be part of the target portal group:

```
# itadm modify-target –t global <iqn>
```

Where <iqn> is the target listed in the previous step. From here, you should be able to manage the rest from the MyRack Manager GUI.

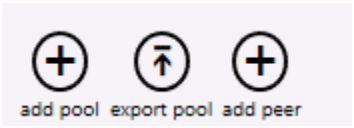
**Example:**

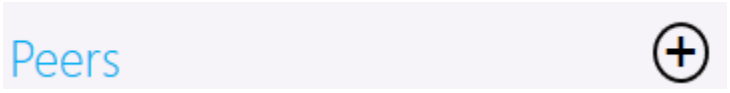
```
# itadm modify-target –t global iqn.2010-03.com.racktopsystems:02:c434c8d7-5643-6364-af5d-cb0bae33d531

# itadm list-target -v
TARGET NAME                                                     STATE    SESSIONS
iqn.2010-03.com.racktopsystems:02:c434c8d7-5643-6364-af5d-cb0bae33d531  online   0
      alias:            -
      auth:             none (defaults)
      targetchapuser:       -
      targetchapsecret:     unset
      tpg-tags:         global = 2
```

## Data Replication

BrickStor supports block replication between two or more pools within the same system or across systems. In order to set up replication between two systems you must establish a peer relationship with the target system from the origin system. Once the peer relationship is created you can set up replication between pools on a per data set basis.

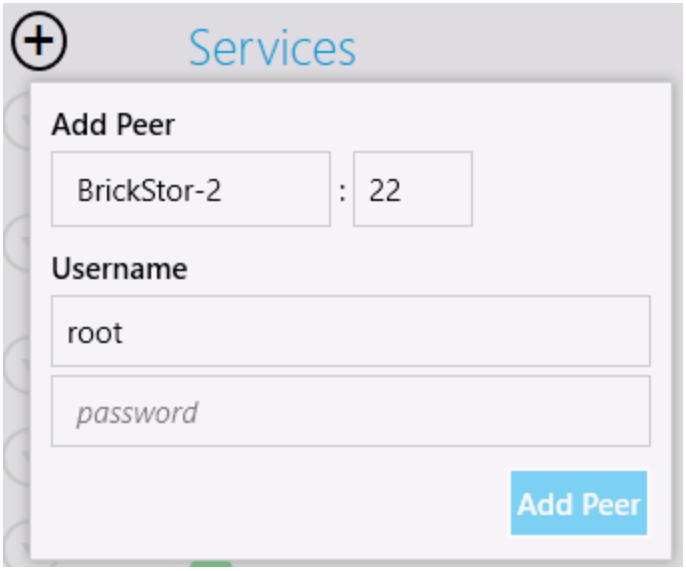
### *Configuring a Peer Relationship*

Click on the Add Peer Button at the bottom left of the main panel 


Or the Plus Button next to Peers 


In the next dialogue simply enter the IP address and host name of the BrickStor you wish to add and then click the Add Peer button.

`

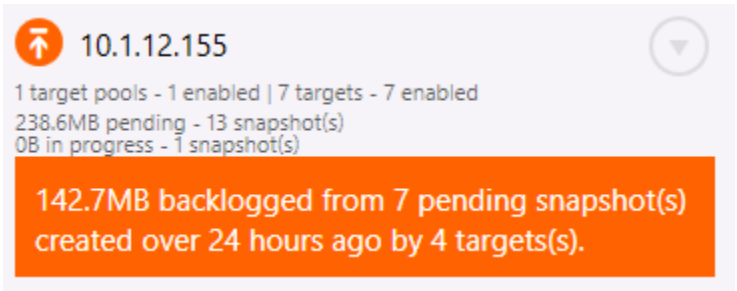



Now the Peer will appear in the list of Peers on the main screen. The Peer will be grey until you have added a target to that peer. You must repeat this process in order to replicate in the reverse direction on the other host.

## Peer Status Symbols

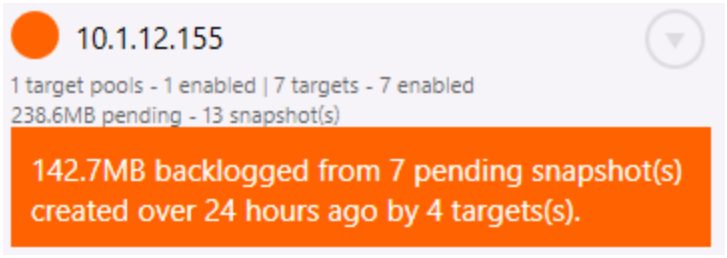
**Healthy No Backlog**


**Backlogged with Transfer in Progress**

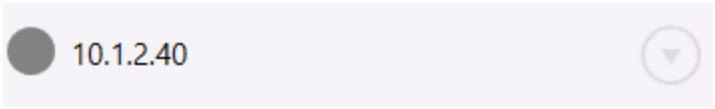



**Backlogged No Transfer in Progress**

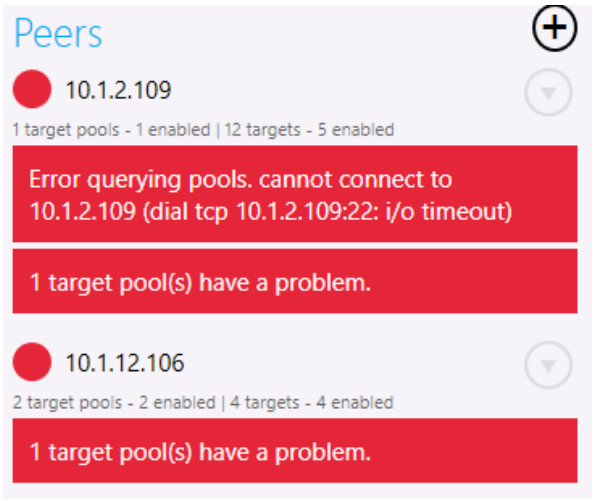
`




**Peer Configured without replication targets enabled for Peer**

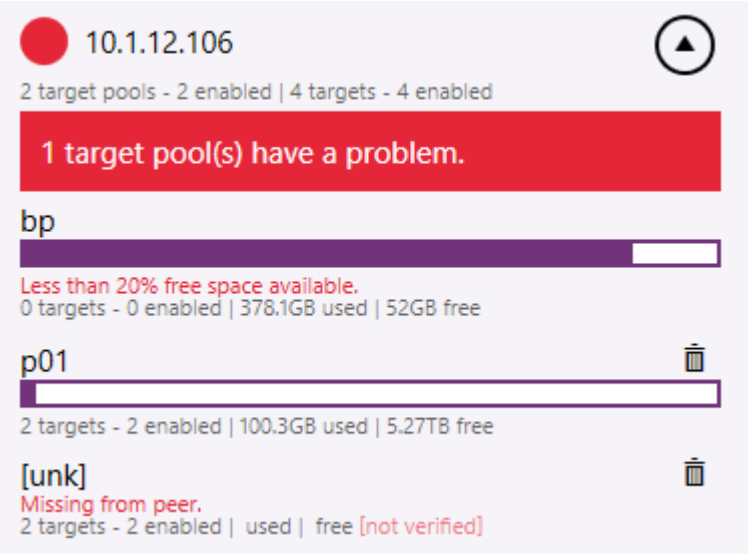



**Peer has a Problem**



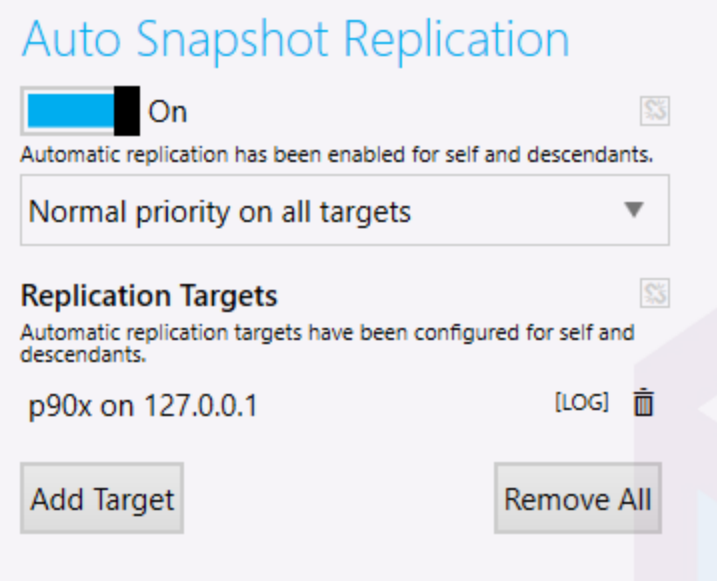

The reason for a Red Peer symbol is that the Peer is unreachable, the target pool is not imported and will show up as [unk] or the target pool is out of space.

`




## Configuring Replication on a Data Set or Volume

Once the peer is established you can replicate the data set or volume to any pool on that peer with at least 20% free space. You can select a replication target at any level of the pool hierarchy and have it inherited so that any data set or volume with snapshots will replicate to the same target or set of targets. By configuring replication on the global container and turning it on anything below inheriting those properties so that they automatically replicate in the same manner. In the example below all volumes and data sets will replicate their snapshots to P90x.




## Replication Hierarchy

Data will be replicated to the target pool under the Replication Container in a Replication Container with the name of the Serial Number of Source BrickStor.

`

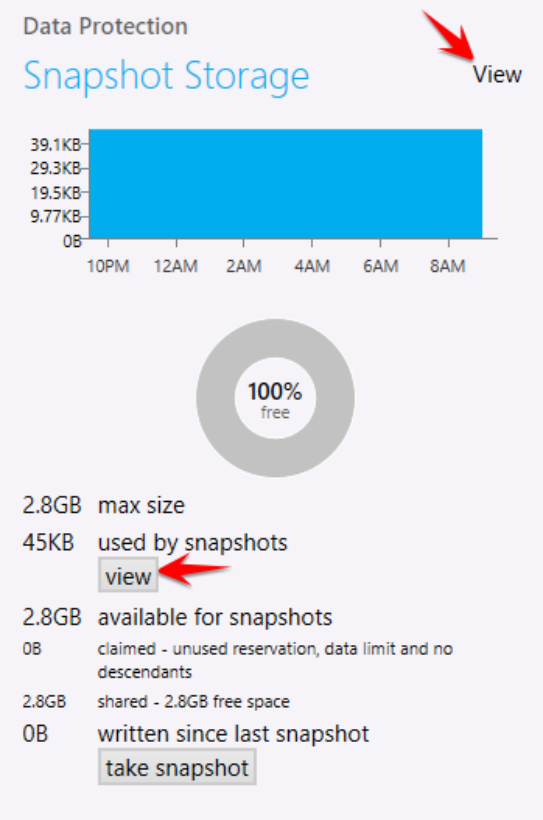
**Hierarchy**

<Pool Name>

- Replication
    - <Serial Number of Source BrickStor>
        - Data Set GUID of Source Data Set

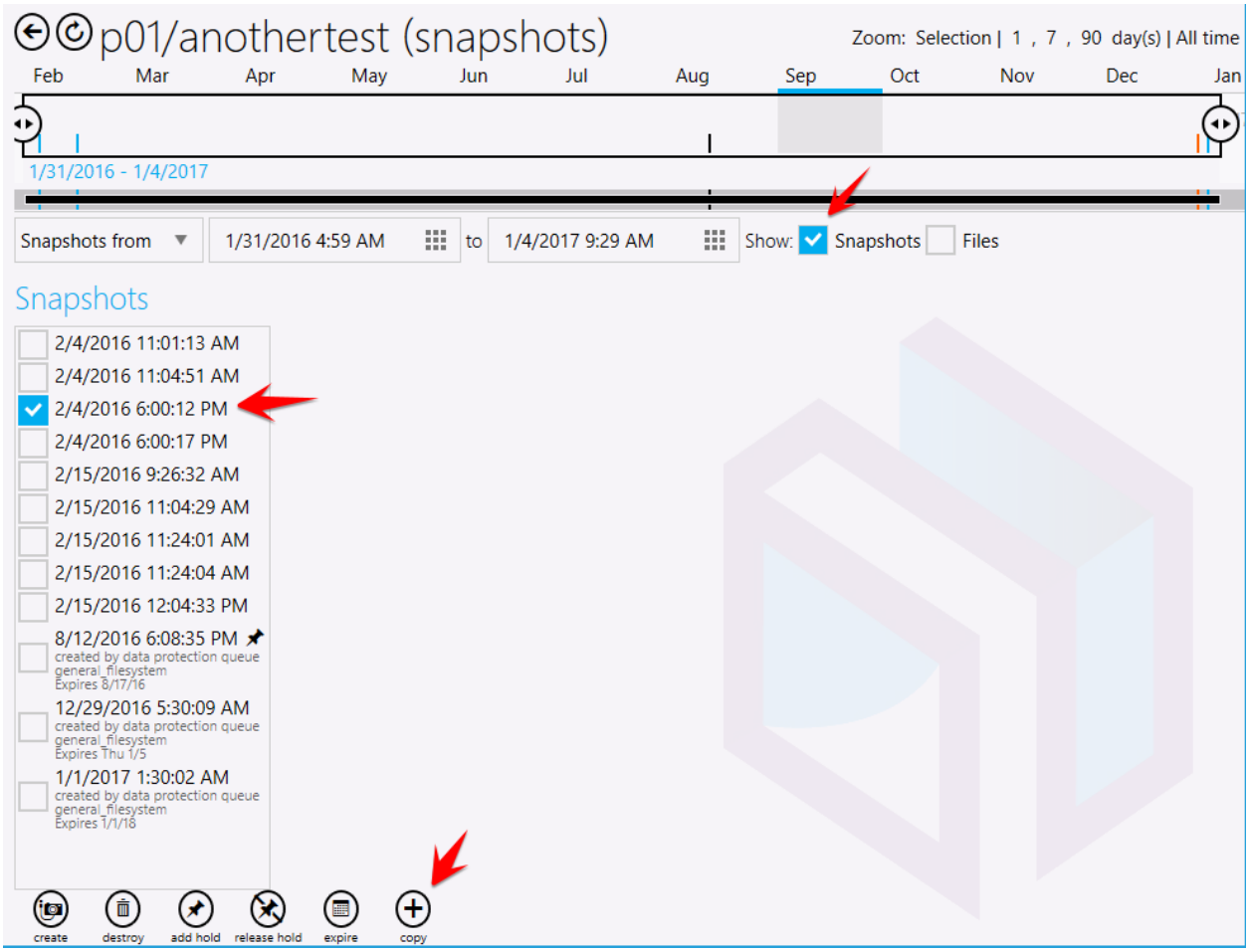
## Restoring Snapshots

To restore a snap for use simply go the data set you wish to restore and click on one of the view buttons.



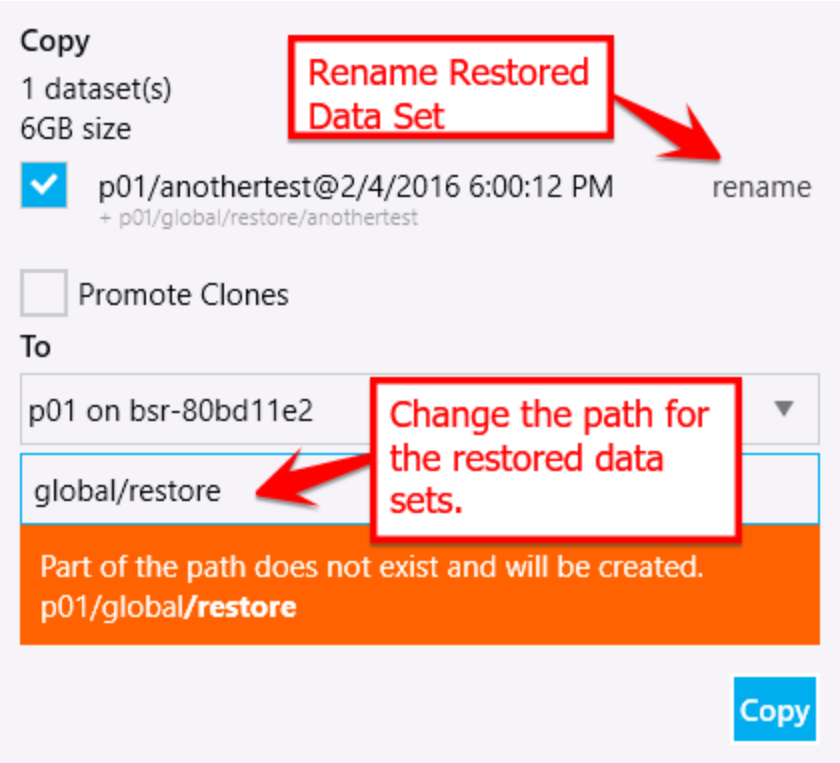
`

In the next screen you will have the option to choose the snapshot you would like to restore. Ensure Show Snapshots is checked which will provide the list of snapshots in the main screen. Click the snaphot you wish to restore and then press the copy button on the bottom.




In the next screen you will have the option to set the path on the restored data set as well as the name of the restored data set. There is an option to promote the clone by checking Promote Clones. If you do this for the case where you want to have a second writeable copy of the same data set or volume and check promote clones all of the snapshots prior to the snapshot you copy will go with the new clone. And the original data set will only have snapshots from after the snapshot you copy. This is not reversible.

`



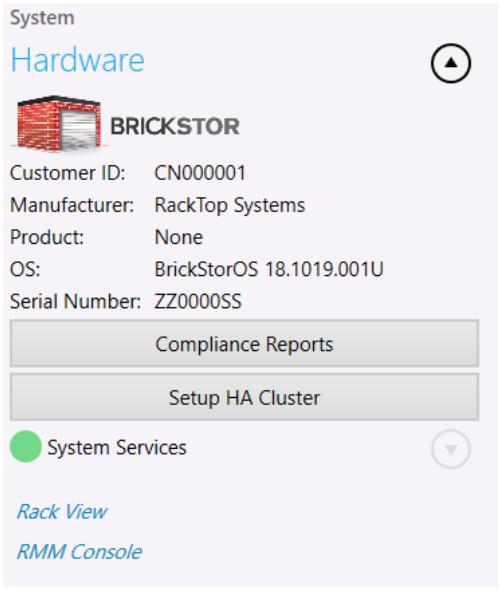

## High Availability Cluster Setup

### *Requirements*

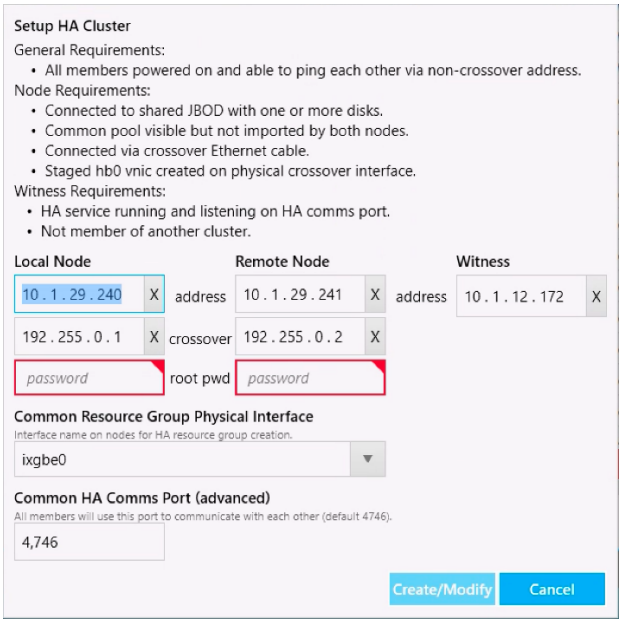
- Follow the HA Guide Cabling Requirements – All data and cache disks must be in a shelf that is accessible by both nodes in the cluster, a heartbeat network cable must be connected directly between the two nodes, and the RMM ports for each node must be configured and connected to the network.
- Create a vNIC labeled hb0 on both nodes on the physical port with the directly connected heartbeat cable. (A true crossover cable is not ne
- Must be able to ping admin0 on both nodes from the other node
- Must have a valid login and password for both RMMs. A new account will be created for the cluster management.
- SSH from each node to the other node. Verify you can connect and certificates have been exchanged and trusted.
- A witness server must be configured (Windows or Linux) with the witness binaries and services running.

To configure HA click on the Setup HA Cluster button after you expand the System Hardware pane on one of the systems you wish to cluster.

`



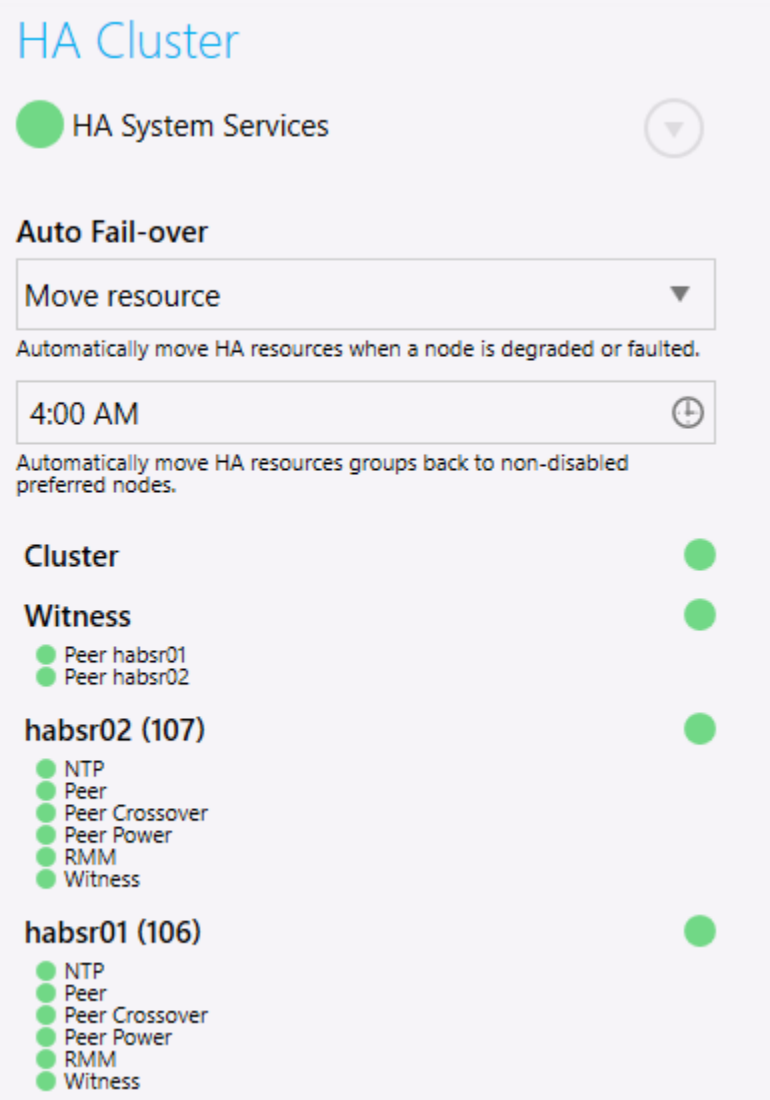

- Add the addresses of the Remote Node and Witness
- Provide the root pwd for both BrickStor nodes
- Choose the physical interface for data (This is the interface the resource group will place the vNIC on) this interface can be an aggregate that is created prior to the HA Cluster Setup.
- Use the default port 4746 unless the witness is configured differently.




Click Create/Modify to configure HA.

`

Once the cluster is created it will appear in the general tab of myRack manager on each node.




Auto Failover can be configured to be disabled or require a manual failover, automatically move resource groups but not disable the failing node, or move the resource groups and disable the failing node which will require an admin to enable the node before it can be configured to accept pools.

When healthy the system will show all green and if there is an error or problem will indicate the failing check or service.

Note that if the term “Peer” is in the status that is referring to the health of the peer. For example If under habsr02 “Peer Power” is Orange it means that habsr01 is experiencing the power issue.

`

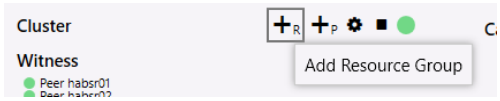
## *Create a Resource Group*

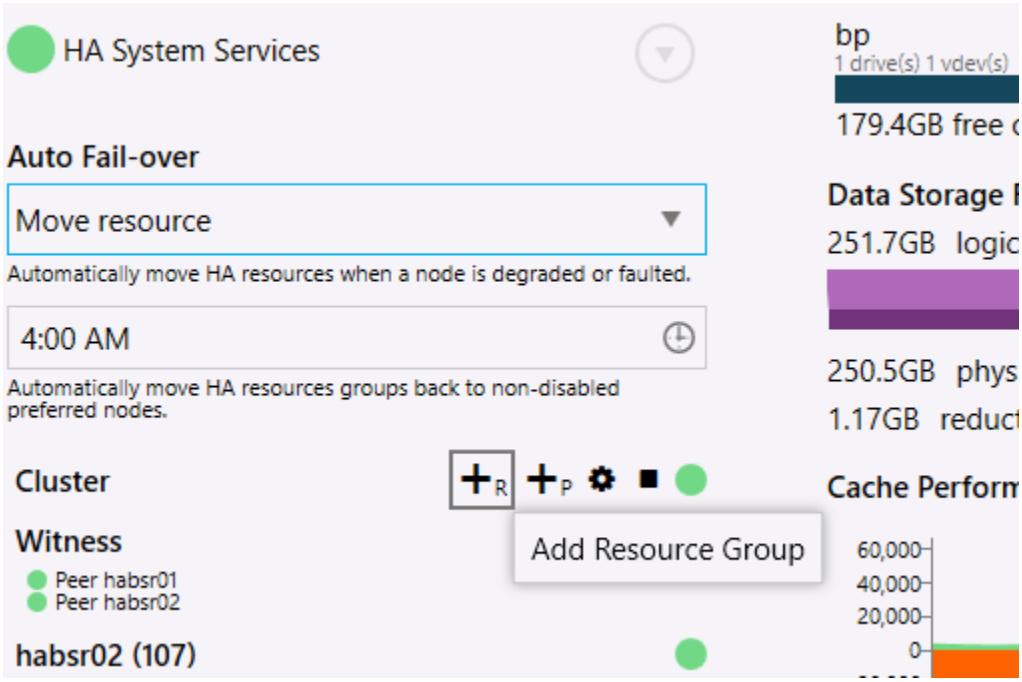
Resource groups are the combination of a vNIC and a Pool. The resource group travels from node to node. During a move/failover the vNIC is removed from the original node and created on the new node while the pool is exported from the original node and imported on the new node.

**Creating A Resource Group**

First create a pool on one of the nodes.

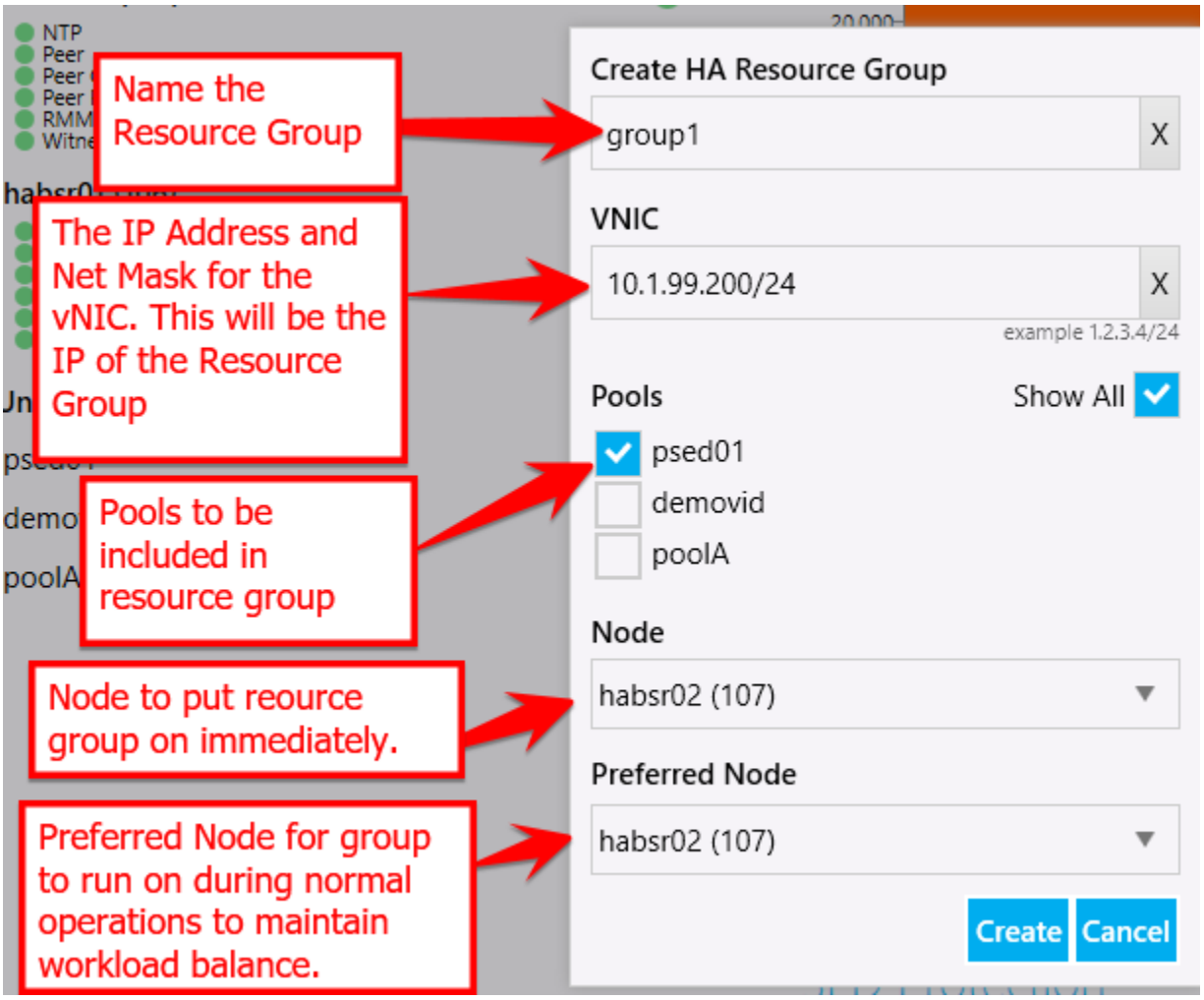
Click the +R Symbol to add a resource group in the cluster.



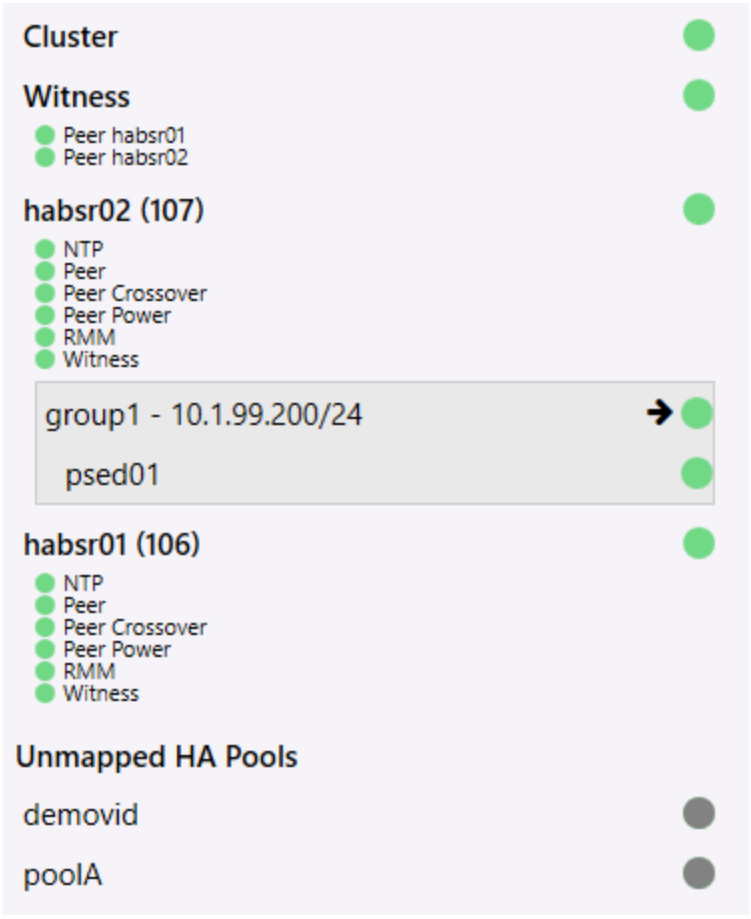
`

Configure the resource group.

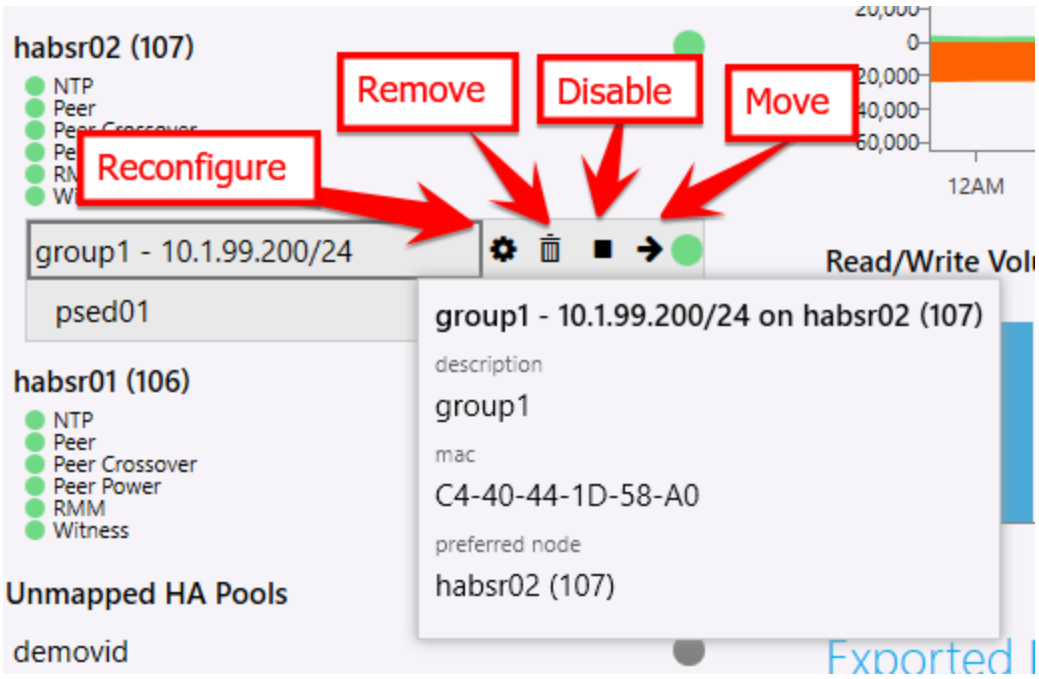



Once the resource group is created you will see it appear on the node.

`



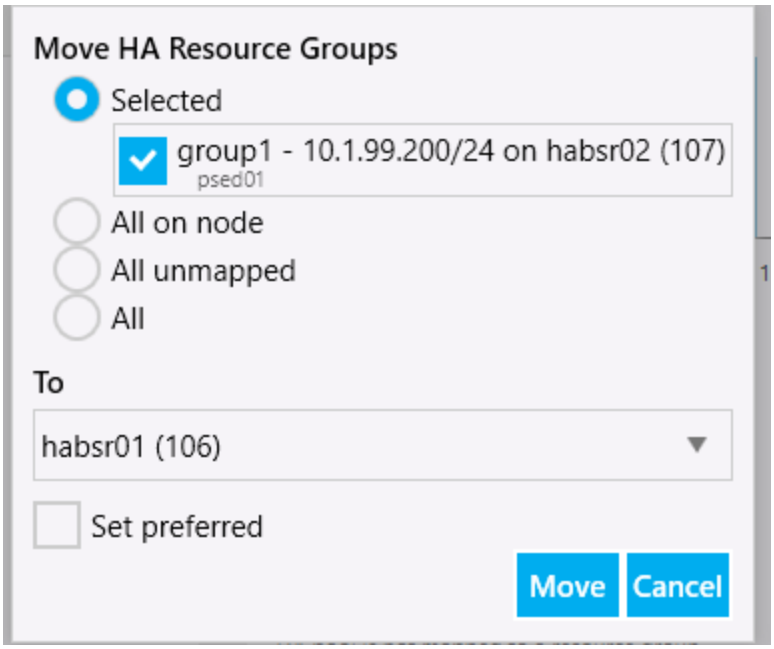

**Managing Resource Groups**



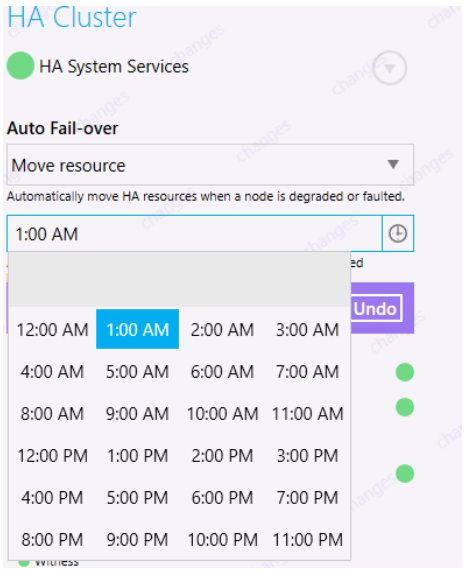
`

Moving the cursor over the resource group name will give the admin options to:

- Reconfigure the resource group
- Remove the resource group from the cluster
- Disable the resource group – temporarily removes the vNIC and exports the pool making it unavailable.
- Move the resource group – move the resource group to the other node.




**Auto-rebalance**




Auto-rebalance will move resource groups to their preferred node if not disabled at a specific time each day.

`

## Self-Encrypting Drive Management

BrickStor can leverage TCG FIPS 140-2 certified self-encrypting drives for increased security. To manage the keys and disks within BrickStorOS does require a special license from RackTop and appropriate FIPS drives. TCG licensed systems come with drives encrypted using a factory generated key. Self-Encrypting Drives placed in a system that are not licensed will not lock when power is removed.

### *Key Manager*

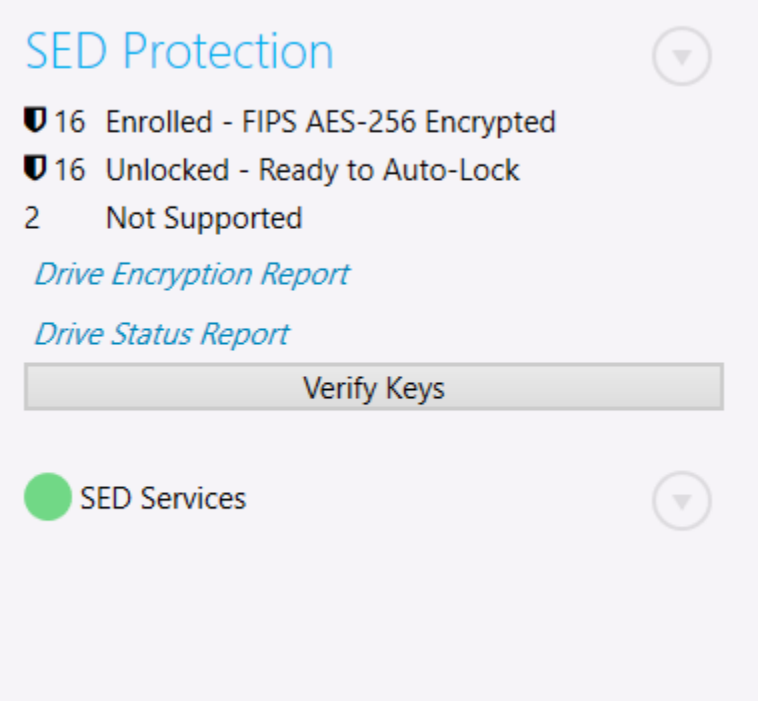
BrickStor supports the use of an internal key manager for non-HA systems and an enterprise KMIP compliant key manager for single node or High Availability BrickStor systems.

From the CLI the admin can configure the external key manager or local key management configuration by editing /etc/keymgrd/keymgrd.conf

Future versions of the UI will enable configuration of the key manager in myRack manager.

Please contact support for specific instructions to set up the external key manager based on your key management solution.

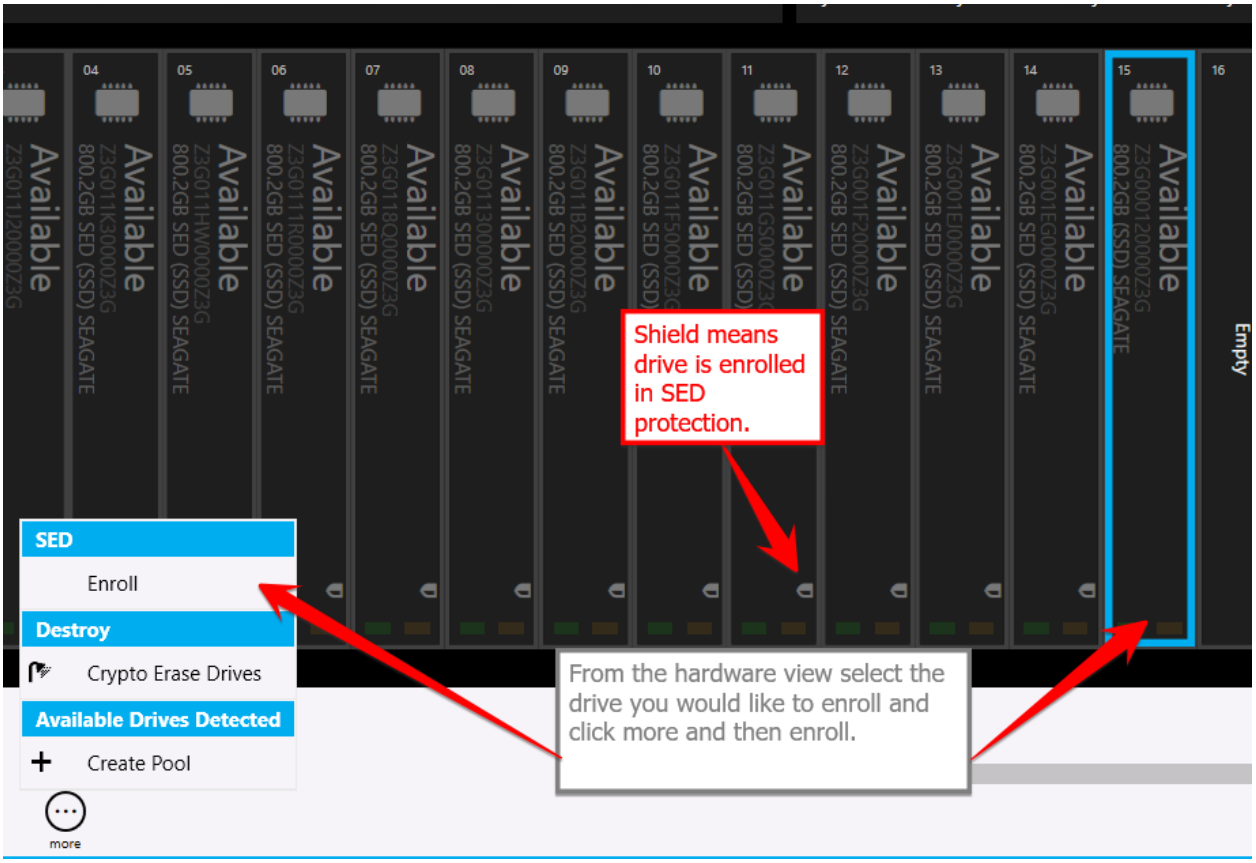
Before enrolling drives ensure all the SED services in the GUI are running on the main details pane.



`

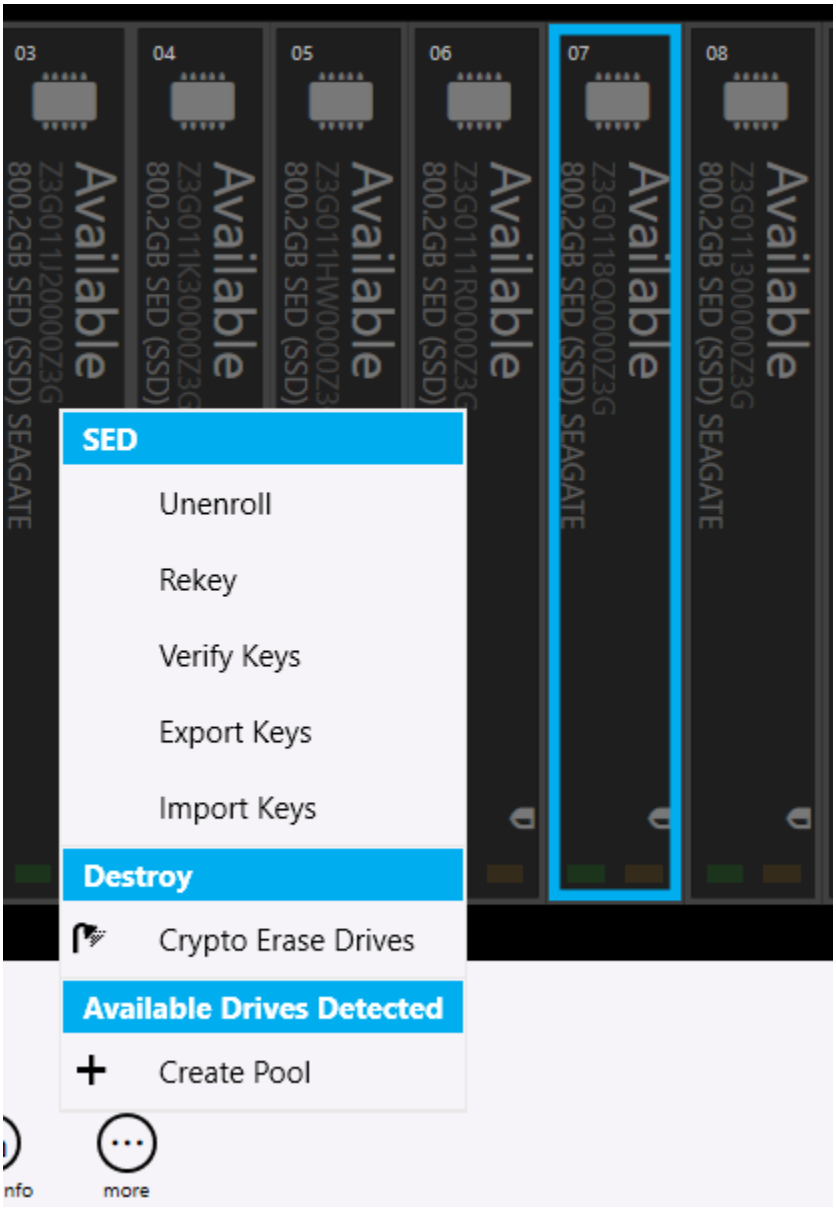
## *Drive Enrollment*

Once the key manager is configured drives can be enrolled in the system. Each drive will receive a unique key used to unlock the self-encrypting drive known as the key encryption key (KEK) from the key manager and configure the drive to auto lock when power is removed from the drive. To enroll drives or a pool in the system go to the hardware view page of the UI. If you select a drive that is not in a pool you can select multiple drives and enroll the ones you choose to enroll. If you select a drive that is already a member of a pool it will enroll all drives that are a member of that pool.



`

### *Other Self Encrypting Drive Operations*




Unenroll – Removes drive from SED management and sets the drive to default PIN and sets the drive to stay unlocked.

Rekey –Requests a new key from the key manager and changes the KEK PIN on the drive.

Verify Key – Verify the KEK unlocks the drive and is available from the key management service

Export Keys – Will provide a password protected file with the KEK PINS that can be imported later for backup purposes or to another node so that the other node can unlock the drives. This is required in HA using the internal key management service.

`

Import Keys – Allows you to import keys that were exported from the same node or another node into the internal key management database. This is performed for HA nodes to share keys between the heads. This can also be used to import keys to a replacement head node.

See the BrickStor SED Usage guide for more details related to Self-Encrypting Systems.

## *Exporting and Backing Up Keys*

When using the BrickStor internal key manager it is important to backup the keys and store them in an alternate location.

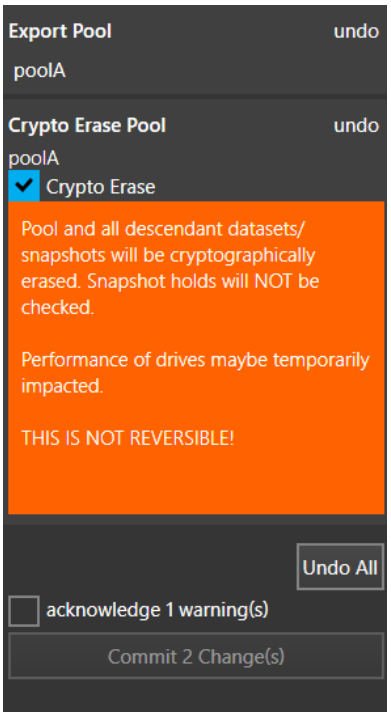
The /etc/racktop/keymgrd.conf file allows users to set the location of the internal key file.

The configuration file also allows users to configure the BrickStor to rotate KEKs on a scheduled internal. This is only recommended when using an external key manager in order to ensure you have backup copies of the keys.

## *Cryptographically Erasing SEDs*

Users can Crypto Erase SEDs which will reset the pins and put them in an unenrolled state. To manage the drive again just enroll the drive.

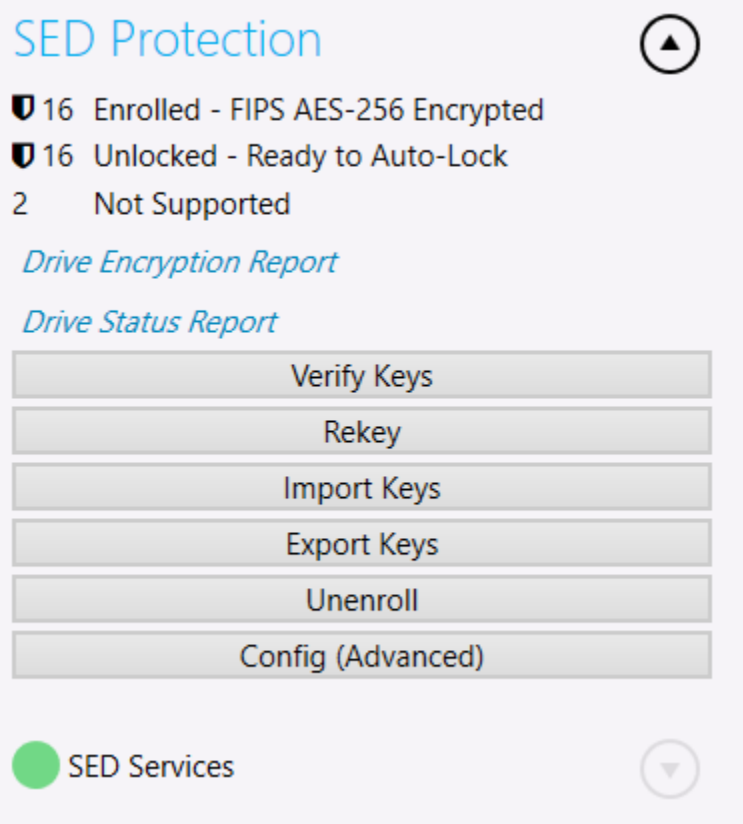
As part of a pool destroy users can select the crypto erase option. This Option is irreversible. Data is permanently destroyed and unrecoverable. However, if you don't select the crypto erase option the data is potentially recoverable in the future off each drive.



`

**If the KEK PIN has been lost for a drive a crypto erase is the only option to put the drive back into a usable state because the drive will become erased and unlocked.**

### *SED Protection on the Main Pane*




Under the general tab of myRack Manager users can perform various SED configuration options as well review reports about which drives are enrolled in SED management and the current status of each drive.

## Configuration & Performance Implications

### *RAID Performance*

BrickStor uses mirrors and RAID-Z for disk level redundancy within vdevs.

#### RAIDZ

RAID-Z vdevs are a variant of RAID-5 and RAID-6:

- You can choose the number of data disks and the number of parity disks. Today, the number of parity disks is limited to 3 (RAID-Z3).
- Each data block that is handed over to ZFS is split up into its own stripe of multiple disk blocks at the disk level, across the RAID-Z vdev. This is important to keep in mind: Each individual I/O

`

operation at the file system level will be mapped to multiple, parallel and smaller I/O operations across members of the RAID-Z vdev.

- When writing to a RAID-Z vdev, ZFS will use a best fit algorithm when the vdev is less than 90% full.
- Write transactions in ZFS are always atomic, even when using RAID-Z: Each write operation is only finished if the überblock has been successfully written to disk. This means there's no possibility to suffer from the traditional RAID-5 write hole, in which a power-failure can cause a partially (and therefore broken) written RAID-5 set of blocks.
- Due to the copy-on-write nature of ZFS, there's no read-modify-write cycle for changing blocks on disk: ZFS writes are always full stripe writes to free blocks. This allows ZFS to choose blocks that are in sequence on the disk, essentially turning random writes into sequential writes, maximizing disk write capabilities.

Just like traditional RAID-5 and RAID-6, you can lose up to 1 disk or 2 disks respectively without losing any data using RAID-Z1 and RAID-Z2. And just like ZFS mirroring, for each block at the file system level, ZFS can try to reconstruct data out of partially working disks, as long as it can find a critical number of blocks to reconstruct the original RAID-Z group.

### Performance of RAIDZ

When the system writes to a pool it writes to the vdevs in a stripe. A Vdev in a RAID-Z configuration will have the IOPS and performance characteristics of the single slowest disk in that vdev (it will not be a summation of the disks). This is because a read from disk requires a piece of data from every disk in the vdev to complete the read. So a pool with 3 vdevs in a RAID-Z1 with 5 disks per vDEV will have the raw IOPS performance of 3 disks. You may see better performance than this through caching but this is the most amount of raw IOPS the pool can deliver from disk. The more vdev's in the pool the better the performance.

### Performance of Mirrors

When the vdev's are configured as mirrors the configuration of the pool is equivalent to RAID-10. A pool with mirrored vdev's will always outperform other configurations. A read from disk only needs data from one disk in the mirror. As with RAID-Z, the more vdevs the better performance will be. Resilver times with mirrored vdevs will be faster than with RAID-Z and will have less of a performance impact on the overall system during resilvering.

**RackTop recommends the use of mirrored vdevs in environments with high random IO such as virtualization because it provides the highest performance.**

## *Compression*

Compression is performed inline and at the block level. It is transparent to all other layers of the storage system. Each block is compressed independently and all-zero blocks are converted into file holes. To prevent "inflation" of already-compressed or incompressible blocks, BrickStor maintains a 12.5% compression ratio threshold below which blocks are written in uncompressed format.

`

BrickStor supports compression via the LZJB, GZIP (levels 1-9), LZE, and LZ4. RackTop finds that LZ4 works very well, balancing speed and compression performance. It is common to realize a 1.3 to 1.6 compression ration with highly compressible data which not only optimizes storage density but also improves write performance due to the reduction in disk IO.

**RackTop recommends always using compression because any CPU penalty is typically outweighed by the savings in storage and bandwidth to the disk.**

## *Deduplication*

Deduplication is performed inline and at the block level, Also like compression, deduplication is transparent to all other layers of the storage system. For deduplication to work as expected the blocks written to the system must be aligned. Deduplication even when turned off will not reverse the deduplication of blocks already written to the system. This can only be accomplished through copying or moving the data. Deduplication negatively impacts the system performance because a deduplication table must be stored in RAM. This takes up space that could otherwise be used for metadata and caching. Should the deduplication not all fit in RAM then system performance will degrade sharply because every read and write operation will require the system to reread the dedup table from disk.

Unless your system is configured with all Solid State Drives, RackTop does not recommend turning on deduplication for high performance environments. The performance impact to the system when utilizing spinning disks negates the storage capacity savings in almost all cases. Instead RackTop recommends the use of ZFS clones to accomplish a similar outcome but with added performance.

## *Clones*

ZFS clones create an active version of a snapshot. By creating a snapshot of a base VM and using clones of that same snapshot you can have an unlimited number of copies of the same base virtual machine without taking up more storage capacity. The only increased storage footprint will come from the deltas or differences between clones. Additionally since each VM will reference the same set of base data blocks the system and user will benefit from caching since all VM's will be utilizing the same blocks of data.

## *Imbalance of vdev Capacity*

If you wish to grow the capacity of a volume by adding another vdev you should do so by adding a vdev of equivalent size to the other vdevs in the pool. If the other vdevs are already past 90% capacity they will still be slow because data will not automatically balance or spread across all vdevs after the additional capacity is added. To force a rebalance in a VMware environment you can perform a vmotion or storage migration. With the Copy On Write Characteristics of ZFS, the pool will automatically rebalance across all vdevs.