



Storage Guide

for RSA NetWitness® Platform 11.x



Copyright © 1994-2020 Dell Inc. or its subsidiaries. All Rights Reserved.

Contact Information

RSA Link at <https://community.rsa.com> contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

For a list of RSA trademarks, go to <https://www.rsa.com/en-us/company/rsa-trademarks>.

License Agreement

This software and the associated documentation are proprietary and confidential to Dell, are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by Dell.

Third-Party Licenses

This product may include software developed by parties other than RSA. By using this product, a user of this product agrees to be fully bound by terms of the license agreements applicable to third-party software in this product.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Dell believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

August 2020

Contents

Storage Overview	5
Storage Requirements	6
Drive Specifications	6
Required NetWitness Platform Storage Volumes	6
Performance Recommendations	7
Input/Output Operations Per Second	8
General Description of How NetWitness Platform Hosts Store Data	8
Prepare Virtual or Cloud Storage	9
Configure Storage Using the REST API	10
REST API Storage Configuration Commands	10
Storage Configuration Tasks	11
Task 1 - Attach Storage to the Host and Access the REST API Storage Commands	11
Task 2 - (Conditional) RAID Configuration for PowerVault and DACs	13
Task 3 - Allocate Block Devices to Partitions, Volume Groups, and Logical Volumes	15
Task 4 - Allocate Volume Groups to NetWitness Services - srvAlloc	17
Task 5 - (Optional) Reconfigure Storage Configuration for 10G Capture	17
Prepare Unity Storage	20
Task 1 - Access Unisphere User Interface (UI)	21
Task 2 - Create Pools	22
Task 3 - Create LUNS	25
Task 4 - Register Hosts	27
Task 5 - Assign LUNS to Hosts	29
Task 6 - Install PowerPath	31
Migrate Data to Another Storage Type	33
Migrate Data Using the Warm and Hot Tier Option	33
Stop the Service	33
Set Up PowerVault	33
Configure The Mount Points	34
Set up Warm and Hot Tiers	35
Decommission the DAC	37
Move Data From DAC to PowerVault	38
Data on PowerVault After Move from DAC	40
Appendix A. How NetWitness Platform Hosts Store Data	42
Decoder Hosts	42
Concentrator Host	42

Archiver Host	43
Hybrid Hosts	43
Options for SAN Configurations	43
Performance Recommendations	43
Appendix B. Encrypt a Series 6E Core or Hybrid Host (encryptSedVd.py)	44
Appendix C. Troubleshooting	47
Reconfigure Pre-Configured DAC Attached to Decoder Using REST API	47
Appendix D. Sample Storage Configuration Scenarios	48
Configure Storage for Archiver	48
Configure Storage for Network (Packet) Decoder	51
Configure Storage for Network Concentrator	61
Configure Storage for Log Decoder Hybrid	67

Storage Overview

This guide provides you with storage requirements and the instructions on how to allocate storage for physical (DACs, PowerVaults, Unity) and virtual storage devices for RSA NetWitness Platform. It also includes the following topics.

- Detect Encryption on Existing PowerVault
- Migrate Data to Another Device

Refer to the following Hardware Setup Guides for information on how to connect these device to RSA NetWitness Platform Core and Hybrid physical hosts:

- PowerVault (MD 1400) Setup Guide (see the "Enclosure Options" section of "Hardware Description") - RSA Link <https://community.rsa.com/docs/DOC-94091>
- 60-Drive DAC Setup Guide - RSA Link <https://community.rsa.com/docs/DOC-44956>
- 15-Drive DAC Setup Guide - RSA Link <https://community.rsa.com/docs/DOC-44957>

Storage Requirements

This section contains all the storage requirements needed to successfully attach storage to your NetWitness Platform deployment host systems. It contains the required drive types, appropriate volumes, and performance IOPS that are needed.

Drive Specifications

General specifications for core NetWitness Platform Hosts are:

- IO size 490/Dec
- Response/Latency < 20ms
- Decoder 10/90 read/write (low random I/O)
- Concentrator 50/50 read/write (high random I/O)

RAID Group	Suitable Volumes
NL-SAS or 10K SAS	All Packet Decoder volumes All Log Decoder volumes All Archiver volumes Concentrator meta volume
SSD	Concentrator index volume

Required NetWitness Platform Storage Volumes

Service Volume Names

Service	Volume Name	File Systems Created
Network Decoder	decoder	packetdb
Network Decoder	decodersmall	decoder root, index, sessiondb, metadb
Log Decoder	logdecoder	packetdb
Log Decoder	logdecodersmall	logdecoder root, index, sessiondb, metadb
Concentrator	concentrator	concentrator root, metadb, sessiondb
Concentrator	index	index
Archiver	archiver	database

Volume Sizing

The volume sizes below are automatically created when using the NetWitness Platform storage tool, described in [Configure Storage Using the REST API](#).

Volume	Filesystem	Mount Point	Size
decodersmall	decoroot	/var/netwitness/decoder	10 GB
decodersmall	index	/var/netwitness/decoder/index	30 GB
decodersmall	sessiondb	/var/netwitness/decoder/sessiondb	600 GB
decodersmall	metadb	/var/netwitness/decoder/metadb	100% of free space on decodersmall volume
decoder	packetdb	/var/netwitness/decoder/packetdb	100% of free space on decoder volume
logdecodersmall	decoroot	/var/netwitness/logdecoder	10 GB
logdecodersmall	index	/var/netwitness/logdecoder/index	30 GB
logdecodersmall	sessiondb	/var/netwitness/logdecoder/sessiondb	600 GB
logdecodersmall	metadb	/var/netwitness/logdecoder/metadb	100% of free space on logdecodersmall volume
logdecoder	packetdb	/var/netwitness/logdecoder/packetdb	100% of free space on logdecoder volume
concentrator	root	/var/netwitness/concentrator	30 GB
concentrator	sessiondb	/var/netwitness/concentrator/sessiondb	10% of free space on concentrator volume
concentrator	metadb	/var/netwitness/concentrator/metadb	100% of free space on concentrator volume
index	index	/var/netwitness/concentrator/index	100% of free space on index volume
archiver	database	/var/netwitness/archiver/database	100% of free space on archiver volume

Performance Recommendations

RSA recommends that Packet and Log Decoders receive two LUNs or Block Devices, one for Packet data, the other for all other databases. This allows you to segregate the high-bandwidth Packet Database from the other databases so they do not compete for I/O bandwidth with other activity.

Concentrators require a separate SSD-based index volume for best performance. You must house this index volume on a different RAID group than the Concentrator Meta database volume, which you can store on NL-SAS. Archivers can use a single large NL-SAS storage volume per appliance.

Input/Output Operations Per Second

The following table lists the IOPS requirements for the Decoder and Concentrator hosts.

Logs	Log Decoder	Concentrator
10K EPS	400	8,000
20K EPS	550	10,300
25K EPS	1,200	10,800

Packets	Network Decoder	Concentrator
1Gbps	600	6,050
2 Gbps	950	8,300
4 Gbps	1,650	12,800
6 Gbps	2,400	17,300
8 Gbps	3,200	21,800

General Description of How NetWitness Platform Hosts Store Data

For information about how NetWitness Platform hosts store data, see [Appendix A. How NetWitness Platform Hosts Store Data](#).

Prepare Virtual or Cloud Storage

Virtual or Cloud NetWitness hosts need block storage attached. Make sure that the allocated storage meets all of the [Storage Requirements](#). Specifically, make sure that the required storage volumes are created (see "Required NetWitness Platform Storage Volumes" in [Storage Requirements](#)), and:

- At least two Block Devices are created for Decoders (meta /session and packet volumes)
- At least two block devices are created for Concentrators (index and meta volumes)
- Ensure that block devices can meet the minimum IOPS for expected ingestion rates

Attach the allocated storage to the NetWitness host by following the hosting platforms native procedure.

- VmWare – Vsphere Console (add disk to VM)
- Hyper-V – Manager Console (add disk to VM)
- Azure – Add Managed Disks to virtual instance
- AWS – Add EBS Storage to virtual instance
- Google Cloud Platform (GCP) - Add storage to virtual instance

After the storage is attached to the virtual host, proceed to "Task 3 - Allocate Block Devices to Partitions, Volume Groups, and Logical Volumes" in [Configure Storage Using the REST API](#).

Configure Storage Using the REST API

In NetWitness Platform 11.3 and later releases, you use the REST API for all storage configuration operations. For information about how to use the REST API, see the *RESTful API User Guide*. Go to the [Master Table of Contents](#) to find all RSA NetWitness Platform 11.x documents.

REST API Storage Configuration Commands

Each of the commands listed below has built-in help that describes their function and usage. If you are using the REST interface, select the command from the drop-down menu to see the help text. For examples of REST API storage configuration commands, see [Appendix D. Sample Storage Configuration Scenarios](#).

Commands for Direct-Attached RAID Volumes

- `raidList` - List the RAID controllers and direct-attach enclosures that are present on this host.
- `raidNew` - Allocate direct-attached enclosures to block devices.

Commands for Allocating Block Devices as Storage

- `devlist` - List available block devices on the host.
- `partNew` - Allocate partitions on a block device and create volume groups.
- `vgs` - Summarize how block devices are organized into volume groups.

Commands for Allocating Storage to Services

- `srvList` - List services on the host and their allocated storage paths.
- `srvAlloc` - Allocate a volume group to a service.
- `srvFree` - Remove a volume group from a service.

Command to Reconfigure Services to Detect and Use All of the New Storage

- `reconfig` - After configuring new storage, detect and use new storage on the associated service and database.

Storage Configuration Tasks

Task 1 - Attach storage to the host and access the REST API storage configuration commands.

Task 2 - (Conditional) Configure RAID if necessary.

Task 3 - Allocate block devices to partitions, volume groups, and logical volumes.

Task 4 - Allocate volume groups to NetWitness services.

Task 5 - Reconfigure services and databases to detect and appropriately use new storage.

Task 1 - Attach Storage to the Host and Access the REST API Storage

Commands

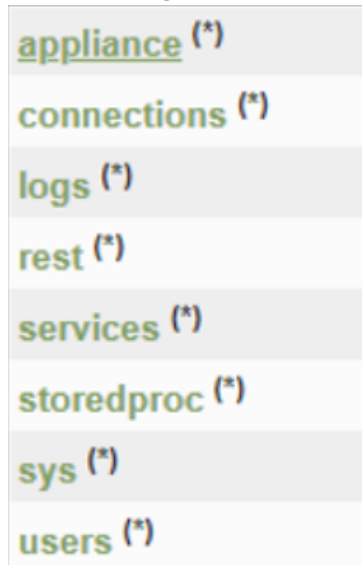
Complete the following steps to attach an external storage device to a host and access the storage configuration commands available through the REST API.

1. Attach the storage and make it available to this host.
 - To attach PV storage, refer to the [PowerVault \(Dell MD 1400\) Setup Guide](#).
 - For third-party storage, create the RAID groups to match the volumes listed in [Storage Requirements](#)
2. There are two ways that you can access the REST API storage commands: from a Browser, or from the **Services > Explore** view from the User Interface.

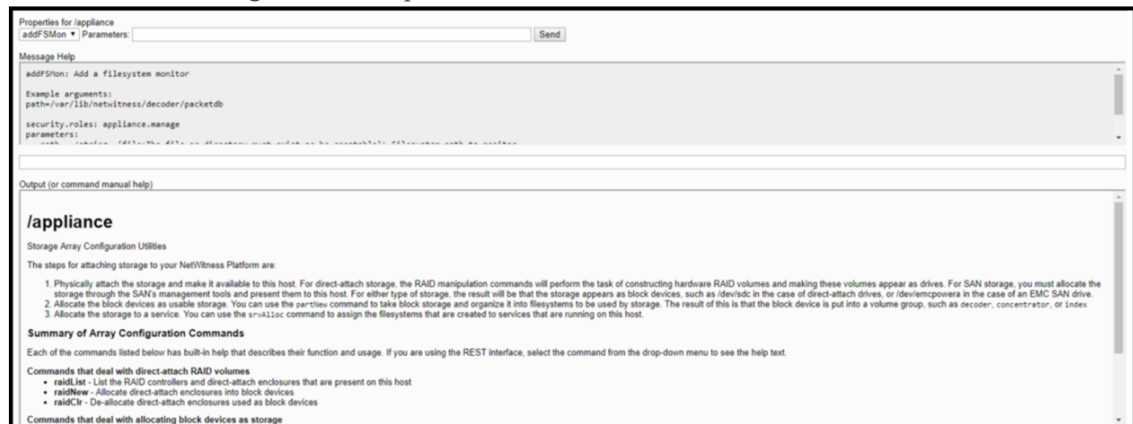
Note: Once you have accessed the REST API, the steps that you perform are the same, no matter which method you used to access it.


- From a Browser.
 - a. Open a Browser and specify the ip-address of the host with port **50106**.
The following example is the Decoder, but you need to use port 50106 for any host hardware for which you are configuring storage using the REST API.
`https://<decoder-ip-address>:50106`

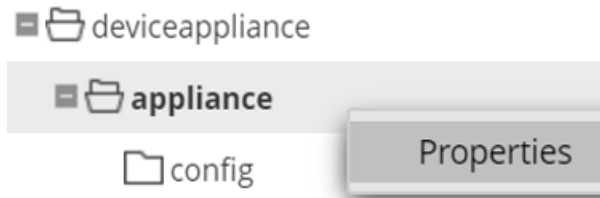
- b. Log in with the `admin` account credentials.
The following REST API menu is displayed.



- c. Click on the `(*)` next to `appliance` to access the REST command set.
The **Properties for /appliance** dialog is displayed under the initial REST menu. The **Output (or command manual help)** section describes the commands that the REST API can send to the device, their usage, and their parameters.



- From the user interface.
 - a. Go to **Admin > Services**.
 - b. Select the service (for example, a Concentrator).
 - c. Under  (actions), select **View > Explore**.
 - d. Navigate to **deviceappliance/appliance**, right click, and click **Properties**.



You can now access the storage commands from the **Properties** dialog.

3. Proceed to:
 - [Task 2](#) if you need to configure RAID for PowerVault or DACs.
 - [Task 3](#) if you do not need to configure RAID and already have a block device available.

Task 2 - (Conditional) RAID Configuration for PowerVault and DACs

NetWitness Platform hardware uses direct-attached SAS drives for storage. These drives are housed in a SAS enclosure. SAS enclosures are shelves of drives attached to the NetWitness node by a cable connected to the SAS host bus adapter.

SAS enclosures are also known as other names, such as "DAC" (Direct-Attached Capacity), or "JBOD" (Jumbo Box of Disks), or "Dell PowerVault".

NetWitness Platform utilizes Dell PERC SAS host bus adapters. NetWitness Platform devices typically include two SAS host bus adapters. One is used for controller drives that are internal to the NetWitness Node, and another is used for controlling drives attached to the SAS enclosures. The internal controller and drives are configured when the node is built, but the external SAS enclosures are not. You execute the `raidList` and `raidNew` commands to identify and configure the external SAS enclosures.

These commands work with the following SAS enclosure types:

- EMC ESAS 15-drive enclosures
- EMC ESAS 60-drive enclosures
- Dell PowerVault 12-drive enclosures

Note: EMC 60-drive enclosures are logically organized as four separate 15-drive sub-enclosures. They behave as if there are four 15-drive enclosures, each of which can be configured independently.

The `raidList` and `raidNew` commands operate on entire enclosures. Execute `raidList` to identify the enclosures. execute `raidNew` to configure an enclosure to perform one of the pre-determined roles within a NetWitness Platform node.

After you attach storage to the host and access the REST API storage commands, complete the following steps to create RAID if required.

1. Execute the `raidList` command to identify the controllers and enclosures that are attached to the system.

In the following example, Controller 1 does not display any block devices. This indicates the array is not configured.

Properties for /appliance

raidList ▾ Parameters: Send

Message Help

```
raidList: list drive shelves attached to this appliance
security.roles: appliance.manage
```

/appliance?msg=raidList&force-content-type=text/plain

Output (or command manual help)

```
Controller 0, Enclosure 32
  Vendor: DP
  Model: BP13G+EXP
  In Use: true
  Drives: 931.511 GB x 2
         1.818 TB x 2
  Devices: sda
          sdb

Controller 1, Enclosure 82
  Vendor: DELL
  Model: MD1400
  In Use: false
  Drives: 10.691 TB x 12
  Devices:

Controller 1, Enclosure 13
  Vendor: DELL
  Model: MD1400
  In Use: false
  Drives: 10.691 TB x 12
  Devices:
```

2. Select a RAID layout scheme for the Enclosure.
The following tables show you the supported allocation schemes.

Note: For RAID configuration, when the Decoder is configured for 10G capture, use the `decoder` scheme for the **first two enclosures** and the `archiver` scheme for subsequent enclosures. When you are not configuring for 10G capture, use the `decoder` scheme for the **first enclosure** and the `archiver` scheme for subsequent enclosures. These configurations will maximize storage capacity and performance.

Scheme	Drives Required	Allocation
decoder	12 or 15 HDDs	3x drives in RAID 5 for decodersmall, all remaining drives in RAID 5 for decoder
logdecoder	12 or 15 HDDs	Same as decoder
archiver	12 or 15 HDDs	All drives in RAID 6 for archiver or decoder database volume

Scheme	Drives Required	Allocation
networkhybrid	12 or 15 HDDs	3x drives in RAID 5 for meta expansion, all remaining drives in RAID 5 for packet expansion
loghybrid	12 or 15 HDDs	Half of the drives in RAID 5 for meta expansion, half the drives in RAID 5 for packet expansion
concentrator	3 or more SSDs, 3 or more HDDs	All SSDs in RAID 5 for index, all HDDs in RAID 6 for meta

- After the controller, enclosure, and scheme are identified, execute the `raidNew` command to create RAID Volumes. For example:

```
send /appliance raidNew controller=1 enclosure=82 scheme=decoder
preferSecure=false
```

Add the `commit=1` parameter to actually execute this operation. Execute the `raidList` command to list the created block devices.
- (Optional) Configure SEDs (Self-Encrypting Drives). If the `raidNew` command detects self-encrypting drives and a security key has been set on the controller, the `raidNew` command will attempt to create a secure array. To set a security key on the controller, execute the `raidKey` command. For example:

```
send /appliance raidKey controller=1 key=myPasssphrase keyId=1
```

 - To create a secured (that is, encrypted) array on physical devices attached to a controller with a security key set, specify `preferSecure=true` when using `raidNew`
 - To create an unsecured (that is, unencrypted) array on physical devices attached to a controller with a security key set, specify `preferSecure=false` when using `raidNew`.
- Go to [Task 3 - Allocate Block Devices to Partitions, Volume Groups, and Logical Volumes](#), after you create RAID volumes.

Task 3 - Allocate Block Devices to Partitions, Volume Groups, and Logical Volumes

The `partNew` command prepares a storage device to use in NetWitness Platform. It performs the following tasks.

- Creates the partition table on the block device.
- Creates the Linux Volume Manager physical device partition.
- Creates a volume group containing the physical device.
- Creates logical volumes in the volume group.
- Creates XFS filesystems on each logical volume.
- Creates `/etc/fstab` entries for each logical volume.
- Mounts each logical volume.

Complete the following steps to allocate block devices to partitions, volume groups, and logical volumes.

1. Run the `devlist` command to locate unused block devices. The following example shows the `devlist` command output.

Output (or command manual help)

```
sda: vendor=DELL model="PERC H730P Mini" size="931 GB" used=1
sdb: vendor=DELL model="PERC H730P Mini" size="1.81 TB" used=1
sdc: vendor=DELL model="PERC H830 Adp" size="21.38 TB" used=1
sdd: vendor=DELL model="PERC H830 Adp" size="85.53 TB" used=1
```

You must provide a name for the service that will be used with the storage, for example, **decoder** for the Network Decoder service, or **concentrator** for the Concentrator service. You have the option of providing the volume type. The default volume type has the same name as the service.

2. Execute the `partNew` command to allocate block devices to partitions, volume groups, and logical volumes.

By default, the `partNew` command does not make changes. It displays the actions that will be taken if you commit the command string. To actually make the changes to the system, add the `commit=true` parameter to the command.

For example, to assign devices `sdd` and `sde` to **Decoder**:

```
send /appliance partNew name=sdc service=decoder volume=decodersmall
commit=true
send /appliance partNew name=sdd service=decoder volume=decoder commit=true
```

Caution: For the **decoder** and **concentrator** services, you must create storage volumes in a specific order.

- The **decoder** has the **decodersmall** and **decoder** volumes. Create the **decodersmall** volume before the **decoder** volume because **decodersmall** contains the small filesystem mounted at `/var/netwitness/decoder`.

- The **concentrator** has the **concentrator** and **index** volumes. Create the **concentrator** volume before **index** volume or it will fail and you receive the following message.

```
Failed to process message partNew for /appliance
com.rsa.netwitness.carlos.transport.TransportException: Volumes for index
require mount point /var/netwitness/concentrator to be created and
mounted first.
```

3. Execute the `vgs` command to validate that the `partNew` command created the correct Logical Volumes.

The output of this command:

- Enumerates all the volume groups on this host.
- Displays the physical volumes that the volume group consists of, and the logical volumes within the volume group.

4. Go to [Task 4 - Allocate Volume Groups to NetWitness Services- `srvAlloc`](#).

Task 4 - Allocate Volume Groups to NetWitness Services - `srvAlloc`

The `srvAlloc` command configures services on a host to use storage in a volume group. You must provide the name of the service to configure and the volume group to assign to the service (the service you provide must be installed on the host). For information about NetWitness Platform service volumes, see "NetWitness Platform Service Volume Reference" in [Storage Requirements](#).

Allocate services in the following order:

- For the Decoder, allocate `decodersmall` first then the `decoder`
- For a Concentrator, allocate `concentrator` first then `index`.


Note: By default, the `srvAlloc` command does not make changes. You must append the `commit=true` parameter to the command string to actually make the changes to the system and restart the specified service after making changes.

1. Execute the `srvLst` command to see a list of services installed on this host.
The `srvLst` command communicates with the service through the SSL port. You install a Category on a host. A Category can be a single service, or multiple related services, located on the same host.
2. Execute the `srvAlloc` command to configure a service on a host to use storage in a volume group.
For example:

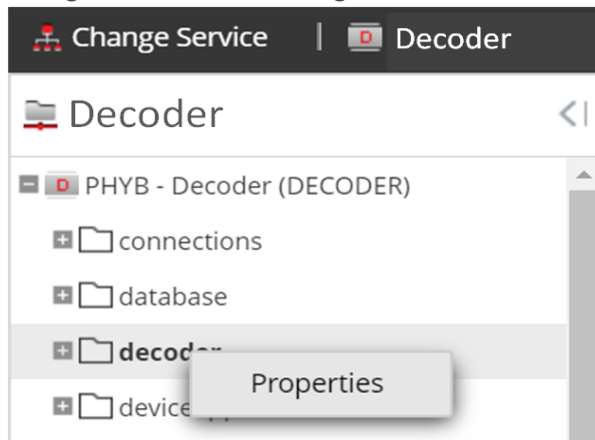
```
service=concentrator volume=index commit=1  
service=concentrator volume=concentrator commit=1
```
3. Go to Task 5 - Reconfigure Services and Databases to Detect and Appropriately Use New Storage.

Task 5 - (Optional) Reconfigure Storage Configuration for 10G Capture

You need to reconfigure the Decoder service and databases for 10G capture. Complete the following steps so that the Network Decoder service and its database detect and use new free space.

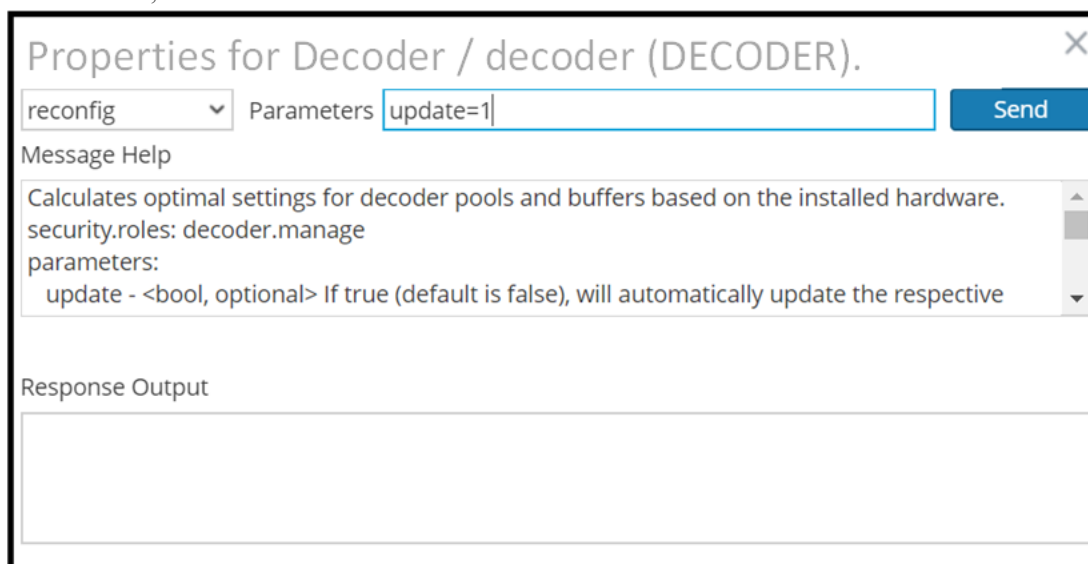
1. Go to **Admin > Services**.
The **Services** view is displayed.
2. Select the **decoder**.
3. Under  (actions), select **View > Explore**.
The **Explore** tree for the service is displayed.

4. Reconfigure space on the **decoder** service.
 - a. Navigate to the **decoder**, right click, and click **Properties**.

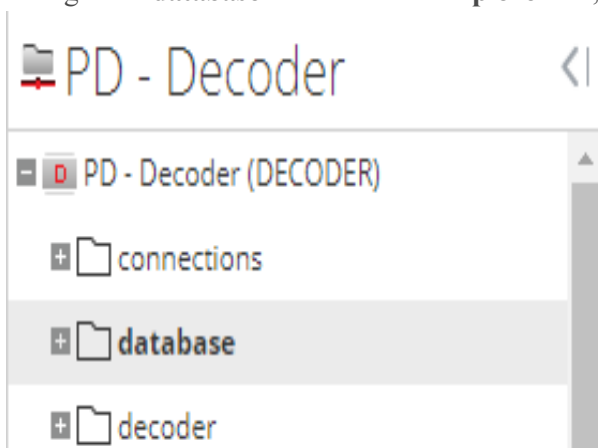


The **Properties** dialog is displayed.

- b. Execute the `reconfig` command by selecting it from the drop-down list, specify `update=1` in **Parameters**, and click **Send**.

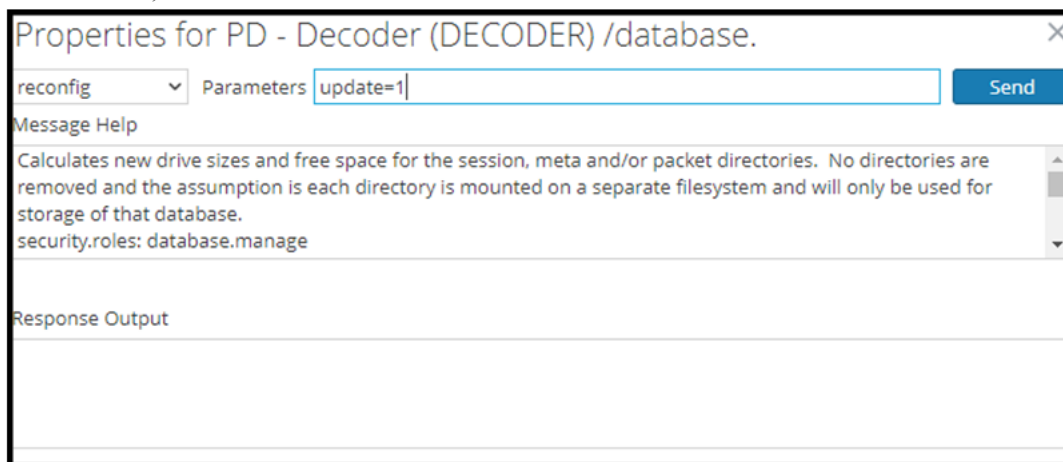


5. Reconfigure space on the database.
 - a. Navigate to **database** in the service **Explore** tree, right click, and click **Properties**.



The **Properties** dialog is displayed.

- b. Execute the `reconfig` command by selecting it from the drop-down list, specify `update=1` in **Parameters**, and click **Send**.



Prepare Unity Storage

You must work with your Dell EMC Storage Engineer to allocate storage within your Unity environment for the RSA NetWitness Platform and ensure the allocated storage meets all of the RSA NetWitness Platform Storage Requirements. Specifically, make sure that:

- You have at least two LUNS created for Decoders (meta /session and packet volumes).
- You have at least two LUNS created for Concentrators (index and meta volumes).
- Ensure block devices can meet the minimum IOPS for expected ingestion rates.

You must add every RSA NetWitness host that uses the Unity storage as a host within the Unity interface. After you create hosts and LUNs, you must assign the LUNs to the hosts. Assigning the LUNs to hosts makes the storage visible to the hosts so they can locate the storage through the host-based Dell EMC PowerPath software.

Note: A Dell EMC engineer will configure the following Unity Array.

You need to perform the following tasks to prepare Unity Storage.

[Task 1 - Access Unisphere User Interface \(UI\)](#)

[Task 2 - Create Pools](#)

[Task 3 - Create LUNS](#)

[Task 4 - Register Hosts](#)

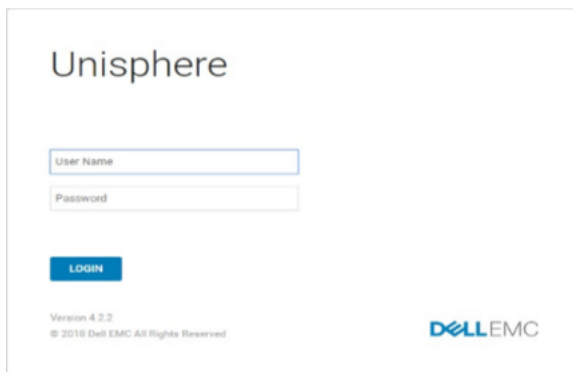
[Task 5 - Assign LUNS to Hosts](#)

[Task 6 - Install PowerPath](#)

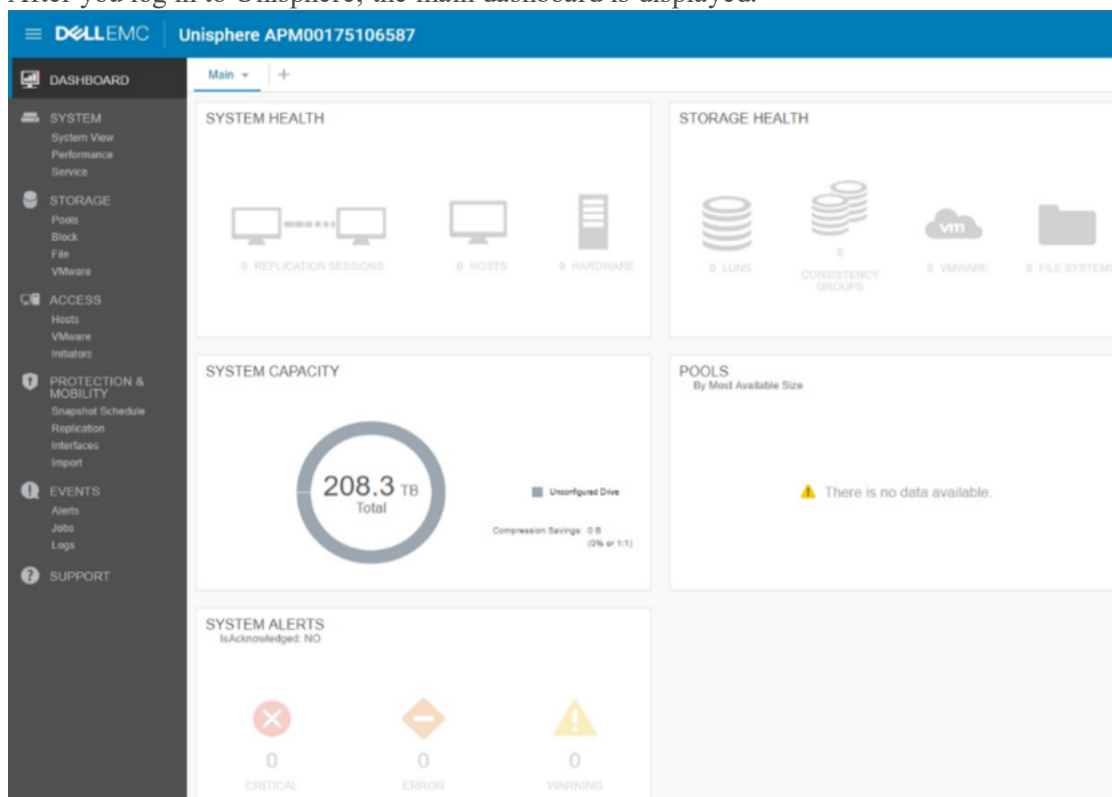
Task 1 - Access Unisphere User Interface (UI)

1. Connect your workstation on the same subnet as the UNITY.
2. Open a browser and go to **http://<unisphereIP>** to connect to the Unisphere UI.
3. Log in with the credentials provided by the DellEMC CE. The default credentials are **admin/Password123#**.

Note: Unisphere will ask you to change password the first time log in. It also asks you to install the license before you can configure array (DellEMC CE may do this for you. You must get the new admin password from them).



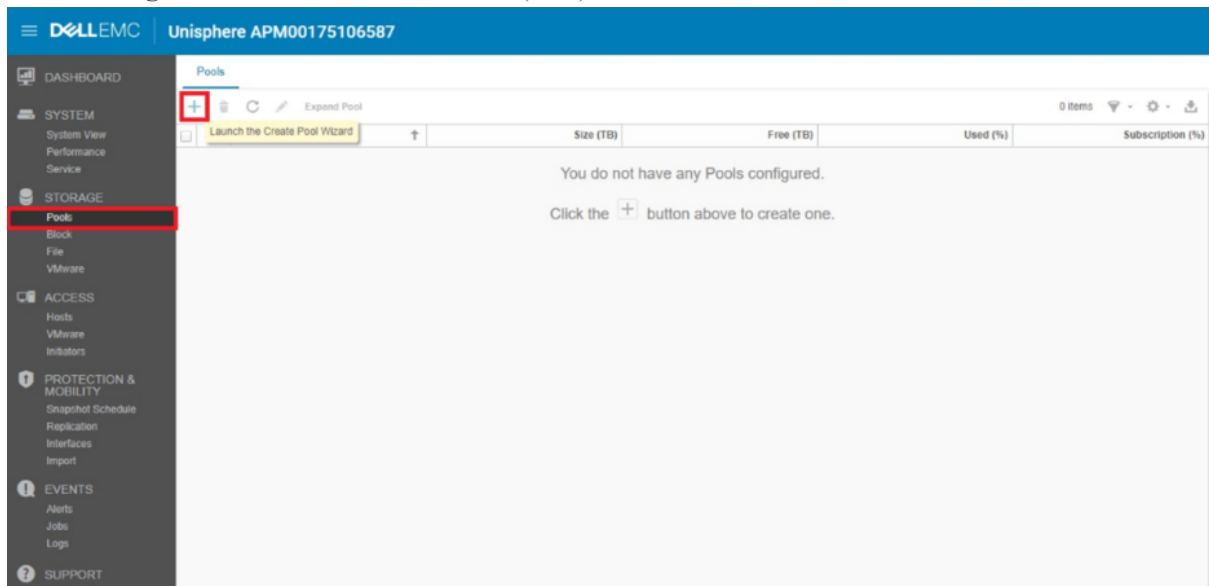
After you log in to Unisphere, the main dashboard is displayed.



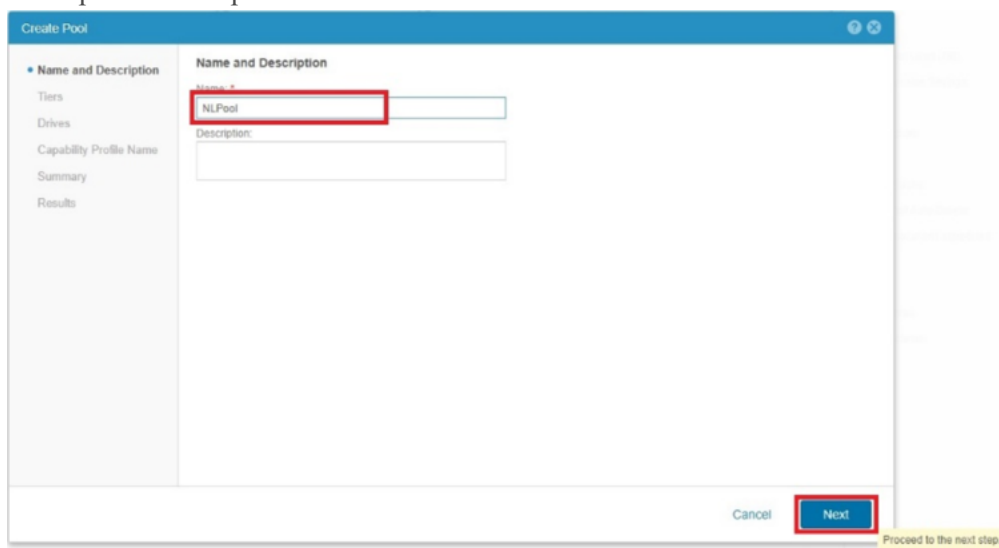
Task 2 - Create Pools

The NetWitness configuration consists of two different pools. One pool is dedicated to the NL-SAS drives and the other pool is dedicated to the SSDs.

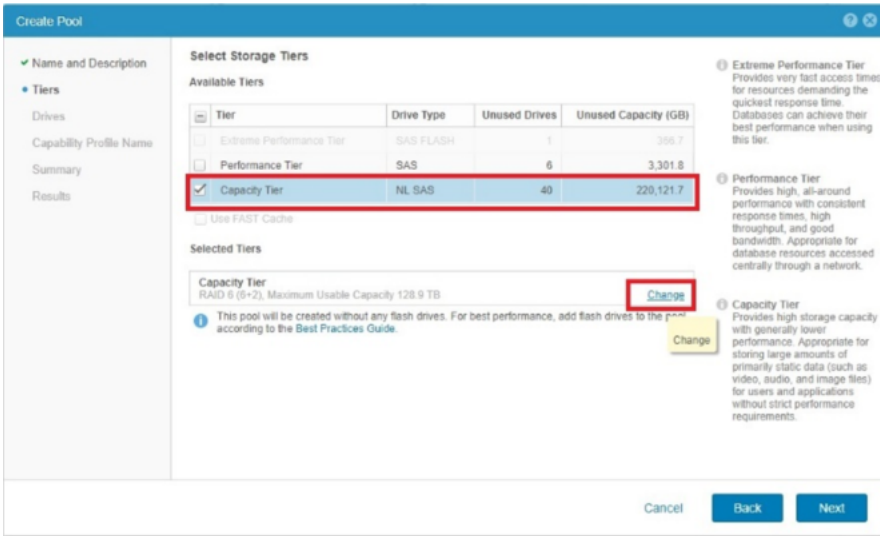
1. From **Storage Section**, click > **Pools** > **+** (Add) to launch the Create Pool Wizard.



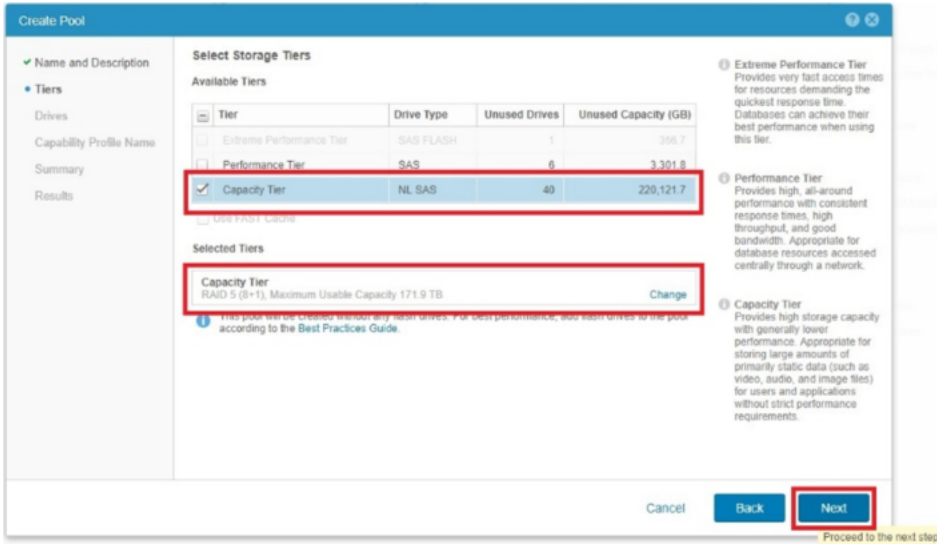
2. Enter in a name for the pool (for example, **NLPool**) and click **Next**. Optionally, you can also enter a description for the pool.



3. Select **Capacity Tier** under **Tier** for the tier type (drive type) and click **Change**.



4. Choose the RAID type and from the drop down and select the RAID size.
The RAID type and size are a customer preference. The only requirement is to make sure you have enough IOPS within the pool to accommodate the log or packet capture and queries. In the following example, a **RAID 5 (8+1)** configuration is selected, however some customers may prefer a **RAID 6 (10+2 or 12+2)**.
5. Make sure you have the correct Raid type and size selected.

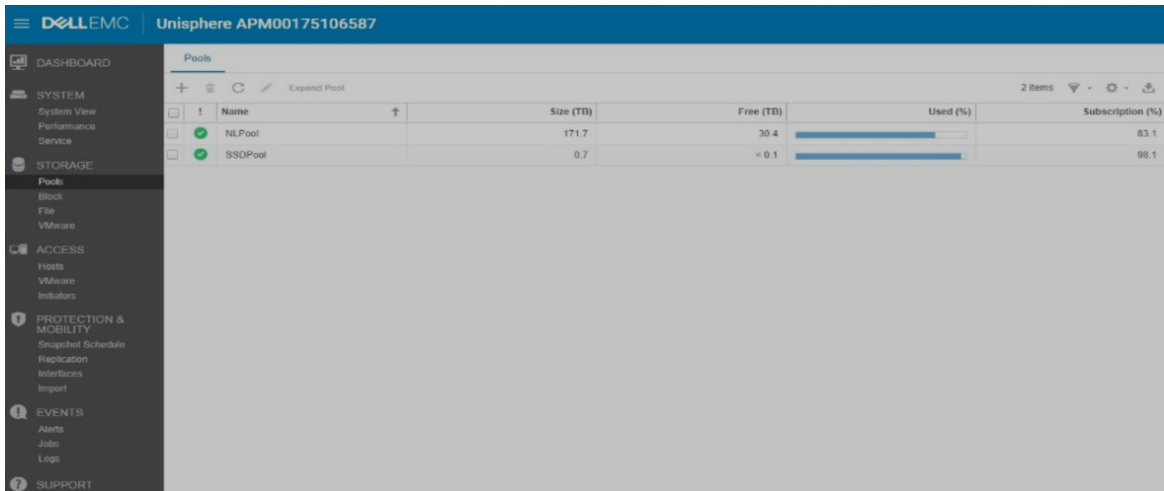


6. Choose the number of drives you want to add into the pool and click **Next**.

7. Skip the **VMware Capability** section and click **Next**.

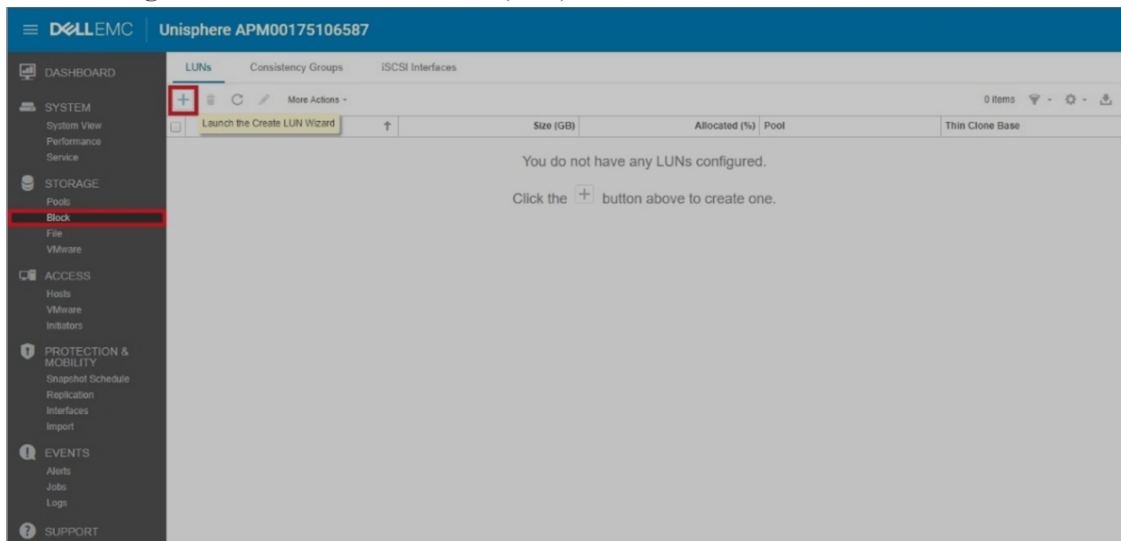
8. Make sure that everything is correct on the Summary tab, and click **Finish**.
9. Create another pool for the SSDs using steps 2 – 8.
- Enter in a name for the other pool (for example, **SDDPool**) and click **Next**. Optionally, you can also enter a description for the pool.
 - Select **Extreme Performance Tier** under **Tier** for the tier type (drive type) and click **Change**.
 - Choose the RAID type and from the drop down, select the RAID size, and click **OK**.

Note: Raid 5 (4+1) RAID Configuration is different then Capacity Tier.



Task 3 - Create LUNS

1. From **Storage** section, click **Block** > **+** (Add) to launch the **Create LUN Wizard**.



The table below list all of the possible LUNS you may need to create. The ConIndex is the only LUN you need to assign to the SSD Pool. Make sure that the LUN sizes do not exceed what is listed below.

DecoderLarge01	75 TB orLess	NL Pool	No
DecoderSmall01	20 TB or Less	NL Pool	No
Concentrator01	15 TB or Less	NL Pool	No
Archiver01	75 TB or Less	NL Pool	No
ConIndex01	3 TB or Less	SSD Pool	No

2. Enter the LUN Name from the list. Optionally, you can enter a description of LUN.
3. Select the appropriate pool from the list on the drop-down menu.
4. Deselect the **Thin** checkbox (These will be fully provisioned LUNs).
5. Select **Next** to proceed to the next menu.

The screenshot shows the 'Create LUNs' configuration window, specifically the 'Configure LUN(s)' step. The interface includes a sidebar with navigation options: Configure, Access, Snapshot, Replication, Summary, and Results. The main configuration area is titled 'Configure LUN(s)' and contains the following fields and options:

- Number of LUNs:** Set to 1.
- Name:** 'DecoderLarge01' (highlighted with a red box).
- Description:** (Empty text field).
- Pool:** 'NLPool (Capacity Tier, 171.9 TB free)' (highlighted with a red box).
- Tiering Policy:** 'Start High Then Auto-Tier'.
- Size:** '20 TB' (highlighted with a red box).
- Thin:** Deselected (checkbox highlighted with a red box).
- Compression:** Deselected (checkbox).
- Host I/O Limit:** 'No Limit'.

At the bottom right, there are 'Cancel' and 'Next' buttons, with 'Next' highlighted by a red box. A yellow tooltip at the bottom right says 'Proceed to the next step.'

6. Click **Next** until you get to the summary section.
7. Verify that the **Name**, **Pool**, **Size** and **Thin** selections are all correct.
8. Click **Finish** to complete LUN creation.

The screenshot shows the 'Create LUNs' configuration window, specifically the 'Summary' step. The sidebar on the left shows the navigation options, with 'Summary' selected. The main configuration area is titled 'Summary' and contains the following information:

- Name and Description:** Name: 'DecoderLarge01' (highlighted with a red box).
- LUN Configuration:** Pool: 'NLPool', Size: '20.0 TB', Thin: 'No' (all three items in this section are highlighted with red boxes).
- Compression:** No.
- Tiering Policy:** Start High Then Auto-Tier.
- Host I/O Limit:** No Limit.
- Host Access:** Access has not been configured for any hosts.
- Snapshot:** No snapshot schedule will be assigned to this LUN.
- Replication:** No replication is being created.

At the bottom right, there are 'Cancel', 'Back', and 'Finish' buttons, with 'Finish' highlighted by a red box.

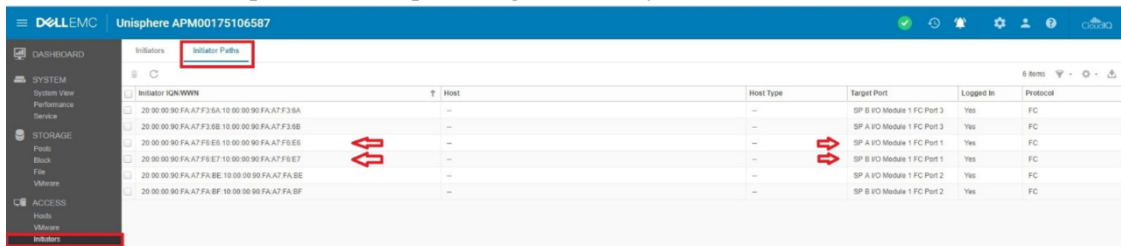
- Repeat steps 2- 8 for the remaining LUN creations.

Task 4 - Register Hosts

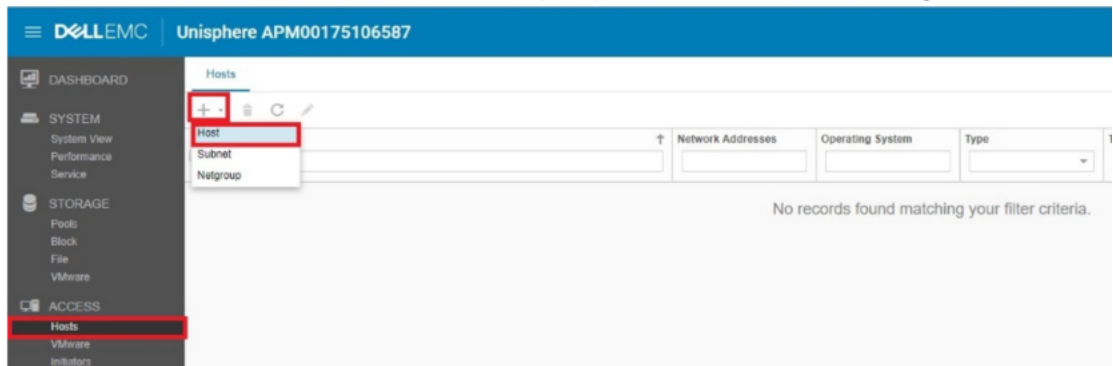
Before proceeding, record the hostname and IP address of the Head Unit and make sure that the HBAs in the head unit are properly cabled to the UNITY.

- From the Access section, click **Initiators**.
- Under the **Initiator Paths** tab, make sure that the correct HBAs are selected that you will use to register the Head Unit.

You should see two initiators per Head Unit. This represents the fiber connection from port 1 to SPA and port 1 to SPB. If you have multiple head units, the easiest method is to power each down and then power them up and register one by one.



- From the **Access** section, click **Hosts** > **+** (Add) > **Host** to add a host configuration.



- Enter the Hostname of the Head Unit.
- Under **Operating System**, select **Linux** from the-drop down menu.
- Enter the IP address of the Head Unit.

7. Click **Next** to proceed to the next section.

Add a Host

Specify a Name and Additional Information

Name: * 95Decoder

Description:

Operating System: Linux

Network Address: 10.25.66.32

Tenant: Select or enter a tenant

While the host operating system information is not required, providing it will allow for more specific setup and troubleshooting instructions.

In order to customize access to NFS shares, the Network Address (name or IP address) is required. Port information is not allowed.

Network Address examples:
 IPv4 address: 192.168.1.2
 IPv6 address: FE80:3202:B3FF:FE1E:8329
 Host name: hostname

Tenant information is not required. To create a tenant, select the Tenants tab for a file system.

Cancel **Next**

Proceed to the next step.

8. In the Initiators section, select the two initiators that correspond to the correct port associated with the Head Unit and click **Next** to proceed.

Add a Host

Select Discovered Initiators or Manually Add Initiators

Automatically Discovered Initiators

Initiator IQN/WWN	Connected To
<input checked="" type="checkbox"/> 20:00:00:90:FA:A7:F5:E6:10:00:00:90:FA:A7:F5:E6	SP A iVO Module 1 FC Port 1
<input type="checkbox"/> 20:00:00:90:FA:A7:FA:BF:10:00:00:90:FA:A7:FA:BF	SP B iVO Module 1 FC Port 2
<input type="checkbox"/> 20:00:00:90:FA:A7:F3:6A:10:00:00:90:FA:A7:F3:6A	SP B iVO Module 1 FC Port 3
<input checked="" type="checkbox"/> 20:00:00:90:FA:A7:F3:6A:10:00:00:90:FA:A7:F3:6A	SP B iVO Module 1 FC Port 3

Manually Added Initiators

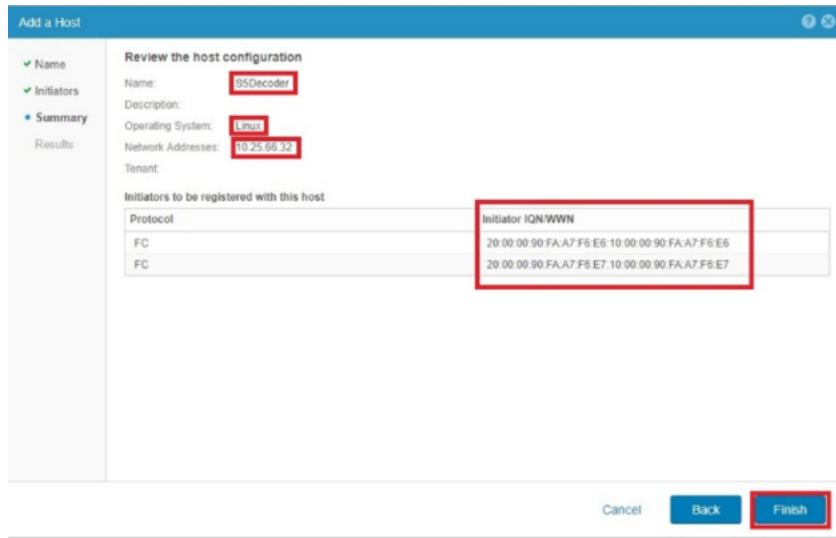
No initiators have been manually added yet. Click the + button to manually add an initiator.

The host uses initiator(s) to access the storage resources. Select from the list of initiators the system has auto-discovered or click the "+" button to manually add an initiator if they are not connected yet.

Cancel **Back** **Next**


Proceed to the next step.

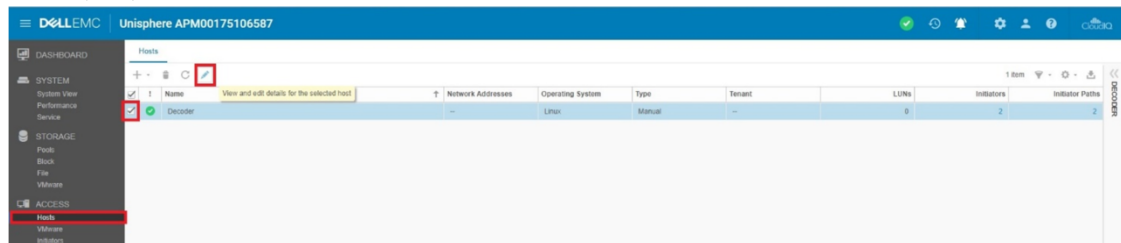
- Make sure that the **Name**, **OS**, **IP** and **WWNs** are correct and click **Finish**.



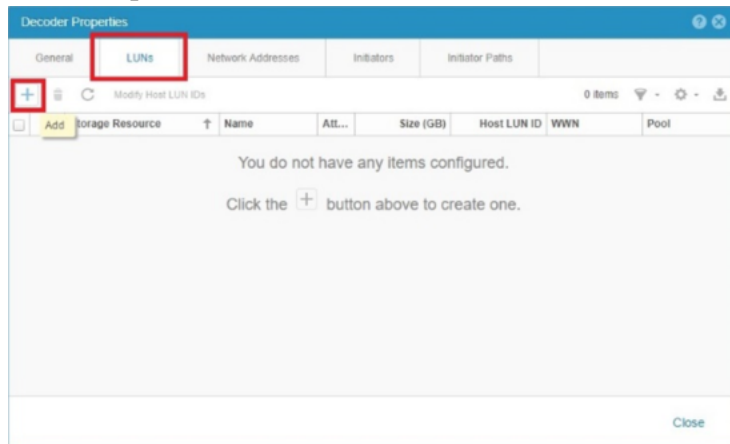
- Repeat steps 2-9 for all Head Units.
- In the Initiators section, select the two initiators that correspond to the correct port associated with the Head Unit. Then click “Next” to proceed.

Task 5 - Assign LUNS to Hosts

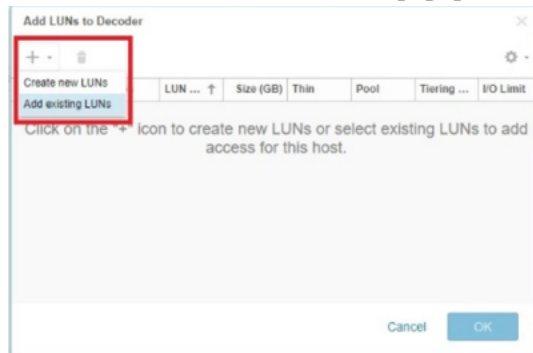
- From the **Access** section, click **Hosts**, select the head unit (for example, **Decoder**) and click  (edit) to view and edit details for the selected host.



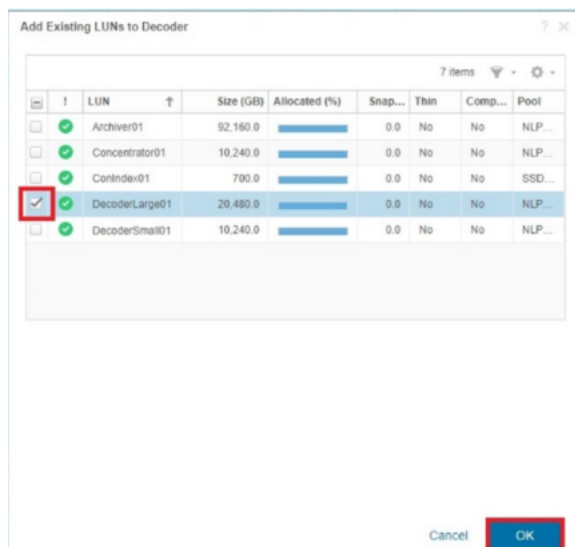
2. In the **Properties** section, select the **LUNS** tab and click  (Add icon).



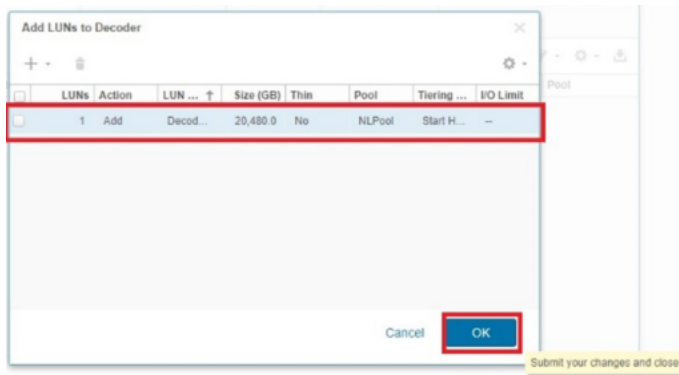
3. From the **Add LUNs to <Host>** popup, click  > **Add existing LUNs**.



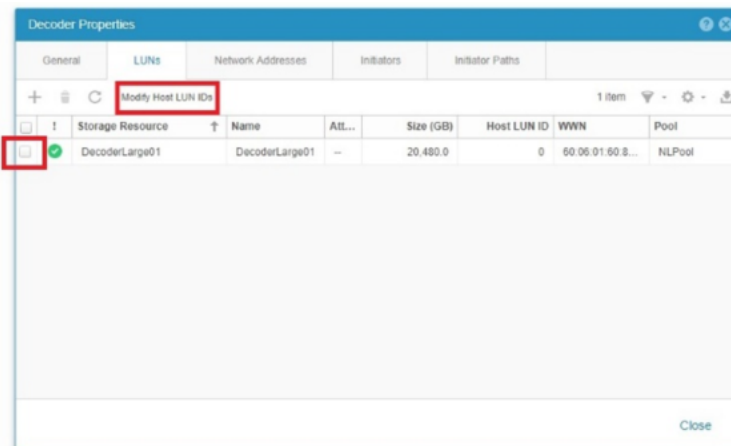
4. Select the LUN to add to the Head Unit and **OK**.



5. Make sure that the correct LUN was added to the host and click **OK**.



6. (OPTIONAL) If you need to modify the HLU (Host LUNN Unique ID):
 - a. Select the LUN you want to change.
 - b. Click **Modify Host LUN IDs**.



7. Click  (edit), change the HLU to the number you want, and click **OK**.

Task 6 - Install PowerPath

1. Make sure that the Emulex ports on the Decoder host are attached to the Unity.
2. Log in to `root` on the Decoder attached to the Unity with the `admin` credentials.
3. Install PowerPath and register the Dell EMC PowerPath licenses for Unity hardware.

```
yum install DelleMCPower.LINUX-6.4.0.00.00-95.RHEL7.x86_64.rpm
```

Note: When you purchase an RSA Provided Unity, PowerPath licenses are sent to you. You can download PowerPath at support.dell.com.

Note: It is possible that the RPM downloaded from Dell EMC is not signed with a cert that the RSA device has available, which can cause the installation to fail with the package not signed error. Run the yum install with the `--ngpgcheck` option to enable the software to install.

- Make sure that all the PowerPath connections are correct.

```
powermt display dev=all
```

The following output is an example of valid PowerPath connections.

```
===== Host ----- - Stor - -- I/O Path -- -- Stats ---
### HW Path          I/O Paths   Interf.  Mode   State  Q-IOs Errors
=====
   15 lpfc            sde       SP A6   active  alive   0      0
   18 lpfc            sdg       SP B6   active  alive   0      0

Pseudo name=emcpowerb
Unity ID=APM00174407815 [Host_62]
Logical device ID=600601609D9046006996745A46B60AB6 [DecoderSmall101]
state=alive; policy=CLAROpt; queued-IOs=0
Owner: default=SP A, current=SP A      Array failover mode: 4
=====
----- Host ----- - Stor - -- I/O Path -- -- Stats ---
### HW Path          I/O Paths   Interf.  Mode   State  Q-IOs Errors
=====
   15 lpfc            sdd       SP A6   active  alive   0      0
   18 lpfc            sdf       SP B6   active  alive   0      0
```

- Verify that the PowerPath license is installed using the `emcprep` command.


```
[root@NWAPPLIANCE24932 ~]# emcprep -list
Key BQPO-DB4M-VFC2-Q24R-ML9Z-EQTU
Product: PowerPath
Capabilities: Al
```
- Add the following string to the `/etc/lvm/lvm.conf` file to filter the LVM (Logical Volume Manager) so that it ignores duplicate volumes.


```
filter = [ "a|^/dev/sda2$|", "a|^/dev/sdb1$|",
"a|^/dev/emcpower.*|", "r|.*/|" ]
```
- Run the following commands in this order:
 - `systemctl enable PowerPath.service`
 - `systemctl start PowerPath.service`
- Reboot the Decoder.
- Complete the instructions in [Configure Storage Using the REST API](#) to complete storage configuration.

Migrate Data to Another Storage Type

This section provides two options for moving data from DACs to PowerVaults:

[Migrate Data Using the Warm and Hot Tier Option](#)

[Move Data From DAC to PowerVault](#)

Refer to the Hardware Setup Guides on RSA Link (<https://community.rsa.com/community/products/netwitness/hardware-setup-guides>) for detailed instructions for setting up RSA NetWitness Platform host and storage hardware.



Migrate Data Using the Warm and Hot Tier Option

In this procedure, you configure a warm tier for the DAC's, so that they do not write any new data. The warm tier continues to be available for analyst operations. You configure the PowerVaults as a hot tier, where new data can be written and available for analysts. When the required data retention is available on the hot tier, the warm tier can be decommissioned.

To set up the warm and hot tiers, perform the following tasks:

- [Stop the Service](#)
- [Set Up PowerVault](#)
- [Configure The Mount Points](#)
- [Set up Warm and Hot Tiers](#)
- [Decommission the DAC](#)

Stop the Service

1. Log in to the NetWitness Platform user interface.
2. Go to **Admin > Services** and select the service (for example, Log Decoder).
3. Click   > **View > Config**, and under Log Decoder Configuration, clear the **Capture Autostart** checkbox, and then click **Apply**.
4. In the menu bar, click the down arrow next to **Config**, select **System**, and at the top of the panel, click **Stop Capture**.
5. From the command line interface in NwConsole, stop the service by running the following command:

```
systemctl stop nwlogdecoder
```

Set Up PowerVault

1. Go to the REST API for the service by entering the IP address of the service, in this example, the Log Decoder. For example, 172.16.0.1:50106.
2. Click the asterisk (*) next to the service. for example, **decoder (*)**.

3. Under **Properties for /decoder**, click the down arrow, select **RaidNew** and enter the following parameters, entering the name of the service for scheme. In this example, we use logdecoder.
`controller=1 enclosure=75 scheme=logdecoder commit=1`
4. Click **Send**.
5. To configure the partitions, click the down arrow again, select **PartNew**, and enter the following parameters,
`name=sde service=logdecoder volume=logdecoderssmall commit=1`
6. Click **Send**.
7. With **PartNew** still selected, enter the following parameters:
`name=sdf service=logdecoder volume=logdecoder commit=1`



Note: To validate the partition definitions before committing them, you can enter these parameters without `commit=1`, and click **Send**. After you validate the parameters, add `#commit=1` and then click **Send** to commit the parameter settings.

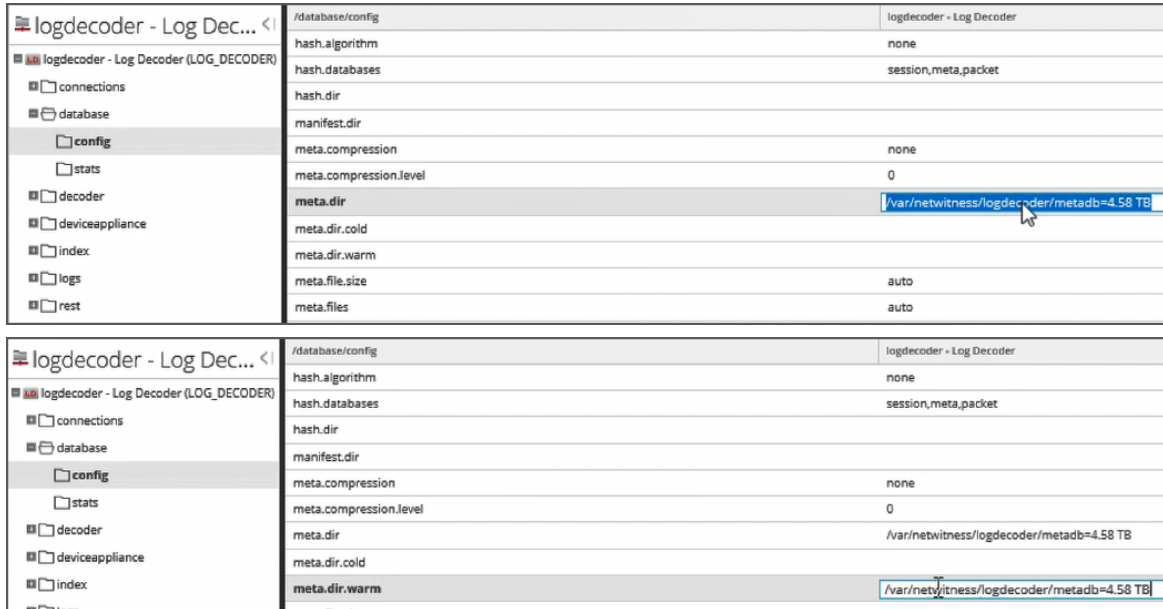
Configure The Mount Points

1. On the NwConsole at the root level of the service (for example, the Log Decoder), run `df -h`.
A list of mounted partitions is displayed.
2. Unmount all of the old storage points of the DAC and copy all the data to the Log Decoder. At the root level, run the `umount` command and the path name of each partition. You can concatenate the path names, for example:
`umount /var/netwitness/logdecoder/index
/var/netwitness/logdecoder/sessiondb /var/netwitness/logdecoder/metadb
/var/netwitness/logdecoder/packetdb /var/netwitness/logdecoder/index0
/var/netwitness/logdecoder/sessiondb0 /var/netwitness/logdecoder/metadb0
/var/netwitness/logdecoder/packetdb0`
3. Temporarily mount the petitions in the `decoroot` folder in the `/mnt` directory in order to access the files. For example:
`mount /dev/mapper/logdecoderssmall-decoroot /mnt/decoroot/`
4. Copy the contents of `decoroot` from `/mnt` to `/var/netwitness/logdecoder`, answering Y (yes) to the prompts:
`cp -R statdb /var/netwitness/logdecoder/`
5. Unmount `/mnt/decoroot`.
`umount /mnt/decoroot`
6. Comment out `decoroot` from `/etc/fstab`, as this was on the DAC and the DAC will be decommissioned.
`#/dev/logdecoderssmall/decoroot
/var/netwitness/logdecoder/xfs/noatime,nosuid 1 2`
7. Mount all the remaining file systems.
`mount -a`
8. Start the `nwlogdecoder` service (with capture still disabled).
`systemctl start nwlogdecoder`

Set up Warm and Hot Tiers

Caution: Before you set up warm and hot tiers, be sure that you know the right warm and hot tier entries for each collection so that you can set them up accurately.

1. Go to **Admin > Services** and select the service (for example, Log Decoder).
2. For the Log Decoder service, click   > **View > Explore**, and go to **database > config**.
 - a. Copy the contents of `meta.dir` and paste them to `meta.dir.warm` as shown in the following example:



logdecoder - Log Decoder (LOG_DECODER)	/database/config	logdecoder - Log Decoder
connections	hash.algorithm	none
database	hash.databases	session,meta,packet
config	hash.dir	
stats	manifest.dir	
decoder	meta.compression	none
deviceappliance	meta.compression.level	0
index	meta.dir	/var/netwitness/logdecoder/metadb=4.58 TB
logs	meta.dir.cold	
rest	meta.dir.warm	
	meta.file.size	auto
	meta.files	auto

logdecoder - Log Decoder (LOG_DECODER)	/database/config	logdecoder - Log Decoder
connections	hash.algorithm	none
database	hash.databases	session,meta,packet
config	hash.dir	
stats	manifest.dir	
decoder	meta.compression	none
deviceappliance	meta.compression.level	0
index	meta.dir	/var/netwitness/logdecoder/metadb=4.58 TB
	meta.dir.cold	
	meta.dir.warm	/var/netwitness/logdecoder/metadb=4.58 TB

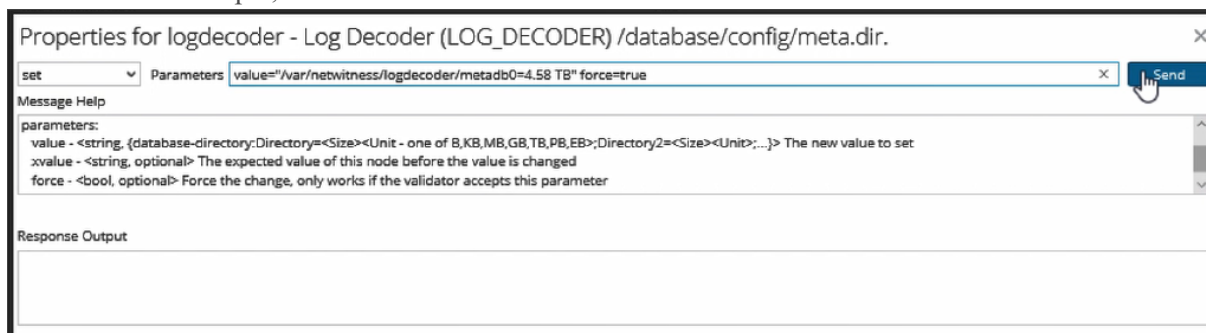
- b. In the same way, copy the packet database in `packet.dir` to `packet.dir.warm`.
 - c. Copy the session database in `session.dir` to `session.dir.warm`.
3. Go to **index > config** and copy `index.dir` to `index.dir.warm`.

Note that the new volumes end in 0, so PowerVault will write to the directories ending in 0, for example:

```
[root@logdecoder ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/netwitness_vg00-root 30G    3.3G   27G  11% /
devtmpfs                  63G         0   63G   0% /dev
tmpfs                     63G    12K    63G   1% /dev/shm
tmpfs                     63G    34M    63G   1% /run
tmpfs                     63G         0   63G   0% /sys/fs/cgroup
/dev/sdal                 1019M    96M   924M  10% /boot
/dev/mapper/netwitness_vg00-nwhome 3.3T    1.2G   3.3T   1% /var/netwitness
/dev/mapper/netwitness_vg00-usrhome 10G     33M    10G   1% /home
/dev/mapper/netwitness_vg00-varlog 10G    1.5G   8.6G  15% /var/log
tmpfs                    13G         0    13G   0% /run/user/0
/dev/mapper/logdecodersmall-index 30G     54M    30G   1% /var/netwitness/logdecoder/index
/dev/mapper/logdecodersmall-sessiondb 600G   733M   599G   1% /var/netwitness/logdecoder/sessiondb
/dev/mapper/logdecodersmall-metadb 4.9T    11G   4.9T   1% /var/netwitness/logdecoder/metadb
/dev/mapper/logdecoder-packetdb 31T     12G    31T   1% /var/netwitness/logdecoder/packetdb
/dev/mapper/logdecodersmall0-index 30G     33M    30G   1% /var/netwitness/logdecoder/index0
/dev/mapper/logdecodersmall0-sessiondb 600G    34M   600G   1% /var/netwitness/logdecoder/sessiondb0
/dev/mapper/logdecodersmall0-metadb 21T     34M    21T   1% /var/netwitness/logdecoder/metadb0
/dev/mapper/logdecoder0-packetdb 86T     35M    86T   1% /var/netwitness/logdecoder/packetdb0
[root@logdecoder ~]#
```

Update the Decoder configuration with the path to the PowerVault mount by adding a 0 to the path.

1. In the `/database/config` column, right-click **meta.dir** and click **Properties**.
2. In **Properties for logdecoder**, select **set**, and in **Parameters**, enter `value="/var/netwitness/logdecoder/metadb0=4.58 TB" force=true`, as shown in this example, and then click **Send**.



3. Repeat step 2 for **session.dir**, **packet.dir**, and **index.dir**. Do not be concerned if the size is the same as the DAC in `"=xx GB"`. This will be updated in the next step.

Note: We are only putting the PowerVault paths into the `*.dir` values.

4. Update the sizes for the live PowerVault volumes.
 - a. In the Log Decoder Explore view, in the left panel, right-click **database** and click **Properties**.
 - b. Select **reconfig** and in **Parameters**, enter `update=1` and click **Send**.
 - c. Repeat steps a and b for **index**.



5. Restart the service.

```
systemctl restart nwlogdecoder
```



6. Go to **Admin > Services**, select the Log Decoder service, and click   > **View > System**.

7. Click **Start Capture**.
8. Go to the **Config** view, select **Capture Autostart**, and click **Apply**.
9. Reboot the host.

Decommission the DAC



When the DAC data has aged, you should go back into the Explore view and remove all of the *.dir.warm configurations for session, meta, packet and index. You can determine when the DAC data has aged by going to the Log Decoder   > **View** Explore view. Since we have a hot and warm tier, there are two sets of configuration stats that you need to be aware of. For example, for a packet Decoder, when you look at the packet oldest time in `packet.oldest.file.time`, look at the `packet.oldest.file.time.hot` value and if you see that your DAC had storage up until 30 days ago you can take your DAC offline and decommission it.

These are the basic steps for decommissioning a DAC. RSA recommends that you work with your Customer Support representative when you decommission your DACs.

1. Go to **Admin** > **Services** and select the service (for example, Log Decoder).
2. Click   > **View** > **Config**, and under Log Decoder Configuration, clear the **Capture Autostart** checkbox, and then click **Apply**.
3. In the menu bar, click the down arrow next to **Config**, select **System**, and at the top of the panel, click **Stop Capture**.
4. From the commandline interface in NwConsole, stop the service by running the following command:

```
systemctl stop nwlogdecoder
```
5. Unmount the warm tier. At the root level, run the `umount` command and the path name of each partition. You can concatenate the path names, for example:

```
umount /var/netwitness/logdecoder/index  
/var/netwitness/logdecoder/sessiondb /var/netwitness/logdecoder/metadb  
/var/netwitness/logdecoder/packetdb /var/netwitness/logdecoder/index0  
/var/netwitness/logdecoder/sessiondb0 /var/netwitness/logdecoder/metadb0  
/var/netwitness/logdecoder/packetdb0
```
6. Comment out all the old DAC dbs from `/etc/fstab`, so that only the PowerVault dbs remain.
7. Start the service.

```
systemctl start nwlogdecoder
```
8. In the user interface, go to **Admin** > **Services** and select the Log Decoder service.
9. Click   > **View** > **Explore** and remove the warm tier configurations:
 - a. In **database** > **config**, delete the content for `meta.dir.warm`, `packet.dir.warm`, `session.dir.warm`.
 - b. In **index** > **config**, delete the content for `index.dir.warm`.
 - c. Go to the **Config** view, select **Capture Autostart**, and click **Apply**
 - d. Go to the **System** view and click **Start Capture**.

- Restart the service.

```
systemctl restart nwlogdecoder
```

The DAC is now unmounted, and is no longer configured in the Decoder for warm storage and is ready to be wiped clean.

- Remove the logical volume. Run `lvscan` to get a list of the logical volumes.
- Run `lvremove` on the old logical volumes, for example:


```
/dev/logdecodersmall/decoroot /dev/lvremove /dev/logdecodersmall/index
/dev/logdecodersmall/sessiondb /dev/logdecodersmall/metadb
/dev/logdecodersmall/packetdb
```
- Remove the volume groups. Run `vgscan` to get a list of volume groups.
- Run `vgremove` on the old volume groups (be careful not to remove any volume groups that end in 0, as they are PowerVault).
- Run `pvs` to view block devices that are freed up.
- When the DAC has been successfully removed, reboot the host.

Move Data From DAC to PowerVault

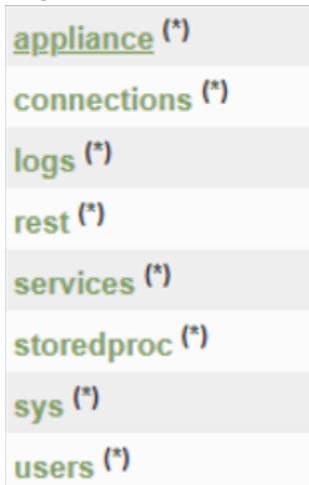
The following procedure describes how to move data from DAC to PowerVault. Before you move data from 2 DACs to 2 PowerVaults, a table, similar to the following table, is displayed if you run the `pvs` (Physical Volume Size) command from the Decoder Linux console (or SSH to the Decoder) with 2 DACs attached and configured to the Decoder. The column headings are Physical Volume (PV), Volume Group (VG), Linux Format (Fmt), Linux Attribute (Attr), Physical Volume Size (PSize), and Physical Volume Free Space (PFree).

PV	VG	Fmt	Attr	PSize	PFree
/dev/sda2	netwitness_vg00	lvm2	a--	<930.00g	0
/dev/sdb1	netwitness_vg00	lvm2	a--	<1.82t	0
/dev/sdc	decodersmall	lvm2	a--	<5.46t	0
/dev/sdd	decoder	lvm2	a--	<27.29t	0
/dev/sde	decodersmall0	lvm2	a--	<5.46t	0
/dev/sdf	decoder0	lvm2	a--	<27.29t	0

Complete the following steps to move data from a DAC to a PowerVault.

- Attach two PowerVaults to a separate PERC controller on the Decoder.
- Create the devices.
 - Open a Browser and specify the ip-address of the Network Decoder and port **50106** to access the REST tool.

- b. Log in with the `admin` account credentials.



- c. Click on the (*) next to **appliance** to access the REST command set.
- d. Run `raidList` to display the Controller/Enclosure combination with the new PowerVault enclosures.
In the following example, the output shows `dev/sdg` and `/dev/sdh` on **Controller 2, Enclosure 246**.

```
Controller 2, Enclosure 246
Vendor:  DELL
Model:   MD1400
In Use:  true
Drives: 10.691 TB x 12
Devices: sdg
        sdh
```

- e. Under **Properties for /appliance**, select `raidNew`, specify `controller=<PowerVault-controller-id>` `enclosure=<PowerVault-enclosure-id>` `scheme=decoder` `preferSecure=false`, and click **Send**.

Note: You specify `preferSecure=false` if the PowerVault drives are not SED drives. If PowerVault drives are SED drives and you do not want to encrypt them you specify `preferSecure=false`. You must specify `preferSecure=true` if PowerVault drives are SED drives and you want to encrypt them.

3. Go to the Decoder Linux console or SSH to the Decoder and run the following commands.

```
parted -s /dev/sdg mklabel gpt
parted -s -a optimal /dev/sdg mkpart LVM 0% 100%
pvcreate -f /dev/sdg
parted -s /dev/sdh mklabel gpt
parted -s -a optimal /dev/sdh mkpart LVM 0% 100%
pvcreate -f /dev/sdh
```

If the volume is created successfully, the following message is displayed.

```
Physical volume "/dev/sdg" successfully created
```

Note: Repeat this step for every block device. The block device names may be different depending on how many enclosures per card slot.

- Run the following command strings to extend the DAC volume group (**decoder**, **decodersmall**) to the PowerVault Physical volume.

```
vgextend decoder /dev/sdg
vgextend decodersmall /dev/sdh
```

- Run the following command strings to move the data from the DAC to the PowerVault. In this following command string, the DAC is **/dev/sdc** and the PowerVault is **/dev/sdg**.

```
pvmove /dev/sdc /dev/sdg
pvmove /dev/sdd /dev/sdh
```

Note: 1.) The `pvmove` command synchronizes data across volumes so that NetWitness can continue ingesting or aggregating data while the migration is executing. You can run the `pvmove` command multiple times if it fails.
2.) Depending on the amount of data on the drives, the move can take a long time complete depending on the amount of data. For example, in a test, it took four hours to move one TB of data.

- After the move is complete, run the following commands to reduce and remove the DAC drive.

```
vgreduce decoder /dev/sdc
pvremove /dev/sdc
vgreduce decodersmall /dev/sdd
pvremove /dev/sdd
```

- Detach the physical connections from the DACs to the host.
- Verify that the Physical volumes are moved from the DACs to the PowerVaults.
 - Reboot the host.

```
reboot
```
 - Verify that the `/etc/fstab` file is correct.
 - Run the `pvs` command and make sure that the **PSize** and **PFree** values are correct on the PowerVault.

```
root@k1t1f1r1cdecoder:~# pvs
  PU          VG          Fmt Attr PSize   PFree
  /dev/sda2   netwitness_vg00  lvm2 a--  <930.00g    0
  /dev/sdb1   netwitness_vg00  lvm2 a--   <1.82t     0
  /dev/sdc1   decodersmall     lvm2 a--   21.38t <15.93t
  /dev/sdd1   decoder          lvm2 a--  <85.54t  58.25t
```

Data on PowerVault After Move from DAC

After you move data from 2 DACs to 2 PowerVaults, a table, similar to the following table, is displayed if you run the `pvs` (Physical Volume Size) command from the Decoder Linux console (or SSH to the Decoder) with 2 PowerVaults attached and configured to the Decoder. The column headings are Physical Volume (PV), Volume Group (VG), Linux Format (Fmt), Linux Attribute (Attr), Physical Volume Size (PSize), and Physical Volume Free Space (PFree).

PV	VG	Fmt	Attr	PSize	PFree
/dev/sda2	netwitness_vg00	lvm2	a--	<930.00g	0
/dev/sdb1	netwitness_vg00	lvm2	a--	<1.82t	0
/dev/sdc1	decodersmall	lvm2	a--	21.38t	<15.93t
/dev/sdd1	decoder	lvm2	a--	<85.54t	58.25t

Appendix A. How NetWitness Platform Hosts Store Data

In most deployments, NetWitness Platform Decoders, Log Decoders, Concentrators, Archivers, and Hybrid hosts require external storage to house their data. Each host uses the external storage in different ways and with different expectations on throughput and performance of the external storage. Some hosts have a higher occurrence of sequential writes and some hosts have a higher occurrence of random reads and writes.

Decoder Hosts

Log Decoders and Network Decoders capture data and parse meta. The difference between these two hosts is in the type of data they capture:

- Log Decoder captures logs.
- Network Decoder captures packets.

Both Log Decoders and Network Decoders parse out meta data from the raw captured traffic. The meta data is then aggregated to a Concentrator for indexing. The host requires storage to house the raw payload data (raw packets or raw logs) and a cache for the meta extracted during data capture for Concentrator aggregation.

Your retention requirements is a key factor in determining the amount of storage you need for the raw packets or raw logs. In most deployments, you add storage over time based on increased retention requirements and increased capture rates. The storage for the raw data must support a high amount of sequential writes with random reads. Especially in the case of higher speed Network Decoder environments, it is recommended to have a minimum of two partitions exposed to the host to support the throttling between partitions for reads and writes.

The meta cache on a Decoder is generally fixed in size but you can expand it to support additional cache the possible loss of connectivity between the Decoder and a corresponding Concentrator. The meta cache must support a random IOPS rate for sustained writes from the Decoder of meta extracted and the corresponding reads from the Concentrator as meta is aggregated to a Concentrator.

Concentrator Host

A Concentrator aggregates and indexes the meta data from a Decoder. Both the meta and index storage needs are scaled based on your NetWitness Platform deployment retention requirements. Similar to raw data stored on the Decoders, you may need to increase the storage for both meta data and index data over time to meet your retention requirements.

The meta storage houses all meta data extracted from either a Network Decoder or Log Decoder. Although the ratio of how much meta is extracted may change, the expectations for performance against meta storage is the same for both packet capture and log capture environments. The meta storage must support a sustained amount of sequential writes with random reads of meta data.

The index storage houses the live index generated from the meta data aggregated to a Concentrator. The size of the index is directly related to the size of the meta store. In addition to supporting IOPS for sustained writes, the index also needs to support a much higher rate IOPS for reads than meta based on interactive queries run through analyst interaction and reports and alerts.

Archiver Host

The Archiver host requires a single partition for both meta and raw log storage. The storage pool deals primarily with sequential writes for long term data written from a Log Decoder or Network Decoder and random reads for reports and analysis.

Hybrid Hosts

A Hybrid hosts two or more services on a single host. For example:

- A Network Hybrid hosts both the Decoder and Concentrator services handling packets exclusively. It captures packet data and indexes this data to the Concentrator service. Expectations for storage performance match what is outlined for a dedicated Network Decoder host and dedicated Concentrator host.
- A Log Hybrid hosts both the Log Decoder and Concentrator services handling logs exclusively. It captures log data and indexes the data to a Concentrator service. Expectations for performance match what is outlined for a dedicated Log Decoder and dedicated Concentrator.
- An Endpoint Log Hybrid hosts the Endpoint Server, Log Decoder, Concentrator, Log Collector, and Endpoint Broker services. It collects and manages endpoint (host) data from Windows, Mac, and Linux hosts, collects log files and Windows logs from Windows hosts, and generates metadata to correlate endpoint data with sessions from other events sources, such as logs and packets.

Options for SAN Configurations

If you want to use a Storage Area Network (SAN) , use the same basic drive groups and partition organization that you use for the other RSA storage devices. Depending on the SAN configuration and overhead, SAN configurations may require more enclosures and drives to operate with the same performance as on PowerVault or DAC. When deciding whether to use SAN, PowerVault, or DAC, any additional overhead on the SAN will be important to determine the minimum storage required.

Performance Recommendations

RSA recommends that Packet and Log Decoders receive two LUNs or Block Devices, one for Packet data, the other for all other databases. This allows you to segregate the high-bandwidth Packet Database from the other databases so they do not compete for I/O bandwidth with other activity.

Concentrators require a separate SSD-based index volume for best performance. You must house this index volume on a different RAID group than the Concentrator Meta database volume, which you can stored on NL-SAS. Archivers can use a single large NL-SAS storage volume per appliance.

Appendix B. Encrypt a Series 6E Core or Hybrid Host (encryptSedVd.py)

RSA Series 6E Core and Hybrid hosts have Self-Encrypting Drives (SED). The `encryptSedVd.py` script:

- Validates that the Series 6E host has the correct setup for encryption.
- Encrypts unencrypted drives.

Note: For external storage devices such as PowerVault, refer to "[Configure Storage Using the REST API](#)" under "Using the REST API to Configure Storage" for instructions on how to encrypt their SED drives.

The following scenarios are examples of why you would use the `encryptSedVd.py` script.

- You want to know if a physical host has encryption. In this case, if the script determines that the device does not have encryption, it gives you the opportunity to encrypt it.
- You set up a device without encryption and you want to encrypt it.

You will find this script in the `rsa-sa-tools` directory for releases 11.4.0.0 and later. The following directory is for 11.4.0.0.

```
rsa-sa-tools-11.4.0.0-<needBuildNumberFromMark>.noarch.rpm
```

The following procedure illustrates how to use the script.

1. Log in as `root`.
2. Change the directory to the `rsa-sa-tools` RPM base directory:

```
cd /opt/rsa/saTools/supportScript/
```

3. Execute the following command:

```
OWB_ALLOW_NON_FIPS=1 ./encryptSedVd.py
```

The script tells you if the disks are encrypted or not encrypted.

- If the drives are encrypted, the script displays the following message.
No unencrypted RAID virtual drives with SED physical drives found.
- If the drives are not encrypted, the script identifies the unencrypted drives as shown in the following example.

```
Detected unencrypted RAID Virtual Drives with SED Physical Disks
Please select the drives to encrypt
Navigation: <Tab><Up/Down Arrow> move vertical
<Esc> Quit, <Enter> Save, <Space> Select/Deselect, <A> Select All, <D> Deselect All

  ID VD  DG  RAID  SIZE  HBA
(■) 0  0  0   RAID1 1.1TB PERC H740P Mini
( ) 0  1  1   RAID1 2.2TB PERC H740P Mini
```


4. If the drives are not encrypted and you want to encrypt them:
 - a. Select the drives you want to encrypt with the space bar and press **Enter**.

The following prompt is displayed.

```
Please enter a passphrase for the PERC H740P Mini security key, minimum length 8 characters, maximum 32
The passphrase must contain a mix of lowercase, uppercase, numeric and non-alphanumeric characters
Optionally enter a key identifier, a default id will be created if not specified

Editing: <Backspace> clear cursor left, <Delete> clear cursor right
Navigation: <Tab><Up/Down Arrow> move vertical, <Left/Right Arrow> move horizontal
<Esc> quit without saving, <Enter> save, trailing spaces are ignored

Enter Passphrase:
[ ]

Verify Passphrase:
[ ]

Key ID (optional):
[ ]
```

- b. In the **Enter Passphrase** text box, type the <passphrase>, for example nFreDaW\$792, and press **Tab**.
- c. In the **Verify Passphrase** text box, re-enter passphrase again for validation.
- d. In the **Key ID (optional)** text box, enter an optional ID string for the security key less than 256 characters or press Enter for none.

The following prompt is displayed.

```
The Passphrase for the security key *Must* be securely backed up in case of PERC adapter hardware
failure and/or replacement, without it the data on all encrypted disks will be unrecoverable.

Entered Passphrase('Quoted'): 'Testing$123'
Entered KeyId('Quoted'): '1'

( ) I understand the risks and have added the passphrase to my organization's permanent record
<Esc> Cancel, <Y> Acknowledge Backup, <D> Decline Backup, <Enter> Save
```

- e. Select <Y> and press **Enter** to confirm that you added the Passphrase.
- f. Submit the following command string to verify that the SED drives are encrypted.

```
/opt/MegaRAID/perccli/perccli64 /c0 show more
```

The following information is displayed. You can see that all four SED drives are encrypted (that is, Y is displayed for each drive in the SED column).

```
Physical Drives = 4

PD LIST :
=====

-----
EID:SlT DID State DG      Size Intf Med SED PI SeSz Model      Sp
-----
64:0      0 Onln  0 1.090 TB SAS  HDD Y   N  512B ST1200MM0069  U
64:1      1 Onln  0 1.090 TB SAS  HDD Y   N  512B ST1200MM0069  U
64:2      2 Onln  1 2.182 TB SAS  HDD Y   N  512B ST2400MM0149  U
64:3      3 Onln  1 2.182 TB SAS  HDD Y   N  512B ST2400MM0149  U
-----
```

You will find detailed information on `perccli` commands in the Dell PowerEdge RAID

Controller CLI Reference Guide (http://14u-00.jinr.ru/pub/misc/h-w/LSI/dell-sas-hba-12gbps_reference-guide_en-us.pdf).

Appendix C. Troubleshooting

This section contains instructions on how to resolve various storage tasks using the REST API.

Reconfigure Pre-Configured DAC Attached to Decoder Using REST API

This scenario covers how to reconfigure a DAC using the REST API that was configured using another tool and clear any pre-existing data (if no longer need or backed up to another storage device).

The following information describes the state of the host and storage hardware prior to the attempt to reconfigure the storage devices using the REST API.

When the DAC was added, it had old data and was configured (but not using the REST API). This prevented the REST API from executing the `raidNew` command and returned the `Physical disk does not have appropriate attributes` error message.

The following steps describe the scenario and with its resolution.

1. From the Decoder Linux console (or SSH to Decoder), submitted the following command string.

```
/opt/MegaRAID/perccli/perccli64 /c2/fall del
```

You will find detailed information on `perccli` commands in the **Dell PowerEdge RAID Controller CLI Reference Guide** (https://topics-cdn.dell.com/pdf/dell-sas-hba-12gbps_reference-guide_en-us.pdf).

This deleted all foreign configuration from controller 2 and cleared all data from the DAC.
2. Tried to partition the DAC, but the `partNew` command failed because that information was already defined on the DAC. `partNew` displayed that you must use one an available device, but `devList` showed it in use.
3. Assuming that the partitions were defined, tried to allocate the storage devices, but this did not work because the DAC was not mounted.
4. Tried to mount the DAC from the command line, but received `mount failed: structure needs to be cleaned` error message.
5. There was no data that needed to be preserved on the DAC, so submitted the following command strings to clean the structure.

```
mkfs.xfs -f /dev/decoder0/packetdb  
mkfs.xfs -f /dev/decoder1/packetdb
```
6. Mounted devices to their appropriate locations in `/var/netwitness/decoder`.
7. Completed the remainder of the applicable steps as described in [Configure Storage Using the REST API](#) to reconfigure the DAC

Appendix D. Sample Storage Configuration Scenarios

This appendix illustrates the following example of how to configure storage on two non-encrypted 15-drive DAC external storage devices.

- [Configure Storage for Archiver](#)
- [Configure Storage for Network \(Packet\) Decoder](#)
- [Configure Storage for Network Concentrator](#)
- [Configure Storage for Log Decoder Hybrid](#)

Configure Storage for Archiver

The following scenario configures storage on one, non-encrypted, 15-Drive DAC for an Archiver physical host.

1. Execute the `raidList` command.
 - a. Record the Controller Number, Enclosure Number, In Use, Drives, and Devices.
You should see the following information.
In Use: FALSE
Devices: <empty>
 - b. Verify the Drive Count, Size, and Vendor.
The following example illustrates what you should see before you create a RAID array.

The screenshot shows a web-based storage management interface. On the left is a navigation tree with folders like 'archiver', 'connections', 'deviceappliance', and 'appliance'. The main area displays 'Properties for NWHOST2100 – Archiver (ARCHIVER)/deviceappliance/appliance.' A dropdown menu is set to 'raidList'. Below this, there is a 'Message Help' section with the text 'list drive shelves attached to this appliance' and 'security.roles: appliance.manage'. The 'Response Output' section shows details for two controllers:

```

Controller 0, Enclosure 32
Vendor: DP
Model: BP13G+EXP
In Use: true
Drives: 931.511 GB x 2
      1.818 TB x 2
Devices: sda
       sdb

Controller 1, Enclosure 0
Vendor: EMC
Model: ESES Enclosure
In Use: false
Drives: 3.637 TB x 15
Devices:
  
```

2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
`controller=1 enclosure=0 scheme=archiver commit=1`

The following example illustrates what you should see after you create a RAID array.

Properties for NWHOST2100 – Archiver (ARCHIVER)/deviceappliance/appliance.

raidNew Parameters controller=1 enclosure=0 scheme=archiver commit=1

Message Help

enclosure - <uint32, (enum-one:32.0)> Enclosure number of the shelf to clear
 scheme - <string, (enum-one:decoder|logdecoder|concentrator|archiver|network-hybrid|log-hybrid)> Type of RAID volumes to allocate
 preferSecure - <bool, optional, (bool:0,1.yes,no,true,false,on,off)> Prefer creation of a secure array given compatible physical drives and a controller with a security key set
 commit - <bool, optional> commit changes

Response Output

```
/opt/MegaRAID/perccli/perccli64 /c1 add vd r6 drives=0:0:0:1,0:2,0:3,0:4,0:5,0:6,0:7,0:8,0:9,0:10,0:11,0:12,0:13,0:14 wb ra cached Strip=128
Controller = 1
Status = Success
Description = Add VD Succeeded
```

- Execute the `raidList` command to verify the new RAID array.

You should now see the following information.

In Use: TRUE

Devices: <device> (for example, `sdc`)

Properties for NWHOST2100 – Archiver (ARCHIVER)/deviceappliance/appliance.

raidList Parameters

Message Help

list drive shelves attached to this appliance
 security.roles: appliance.manage

Response Output

```
Controller 0, Enclosure 32
Vendor: DP
Model: BP13G+EXP
In Use: true
Drives: 931.511 GB x 2
      1.818 TB x 2
Devices: sda
        sdb

Controller 1, Enclosure 0
Vendor: EMC
Model: ESES Enclosure
In Use: true
Drives: 3.637 TB x 15
Devices: sdc
```

- Execute the `partNew` command with the following parameters to create partitions and mount points in the `etc/fstab` file.
`name=<device>` (for example, `sdc`) `service=archiver` `volume=archiver` `commit=1`
- Execute the `srvAlloc` command with the following parameters to allocate the space to the archiver service. This adds storage to the archiver service configuration and restarts the service every time it is executed.
`service=archiver` `volume=archiver0` `commit=1`

Properties for NWHOST2100 – Archiver (ARCHIVER)/device/appliance/appliance.

srvAlloc ▼ Parameters `service=archiver volume=archiver0 commit=1`

Message Help

```
service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
volume - <string, {enum-one:archiver0|netwitness_vg00}> volume group name
commit - <bool, optional> commit changes
```

Change Service | NWHOST2100 - Archiver | System ⌵

Start Aggregation Stop Aggregation Host Tasks Shutdown Service

Archiver Service Information

Name NWHOST2100 (Archiver)
 Version 11.3.0.0 (Rev null)
 Memory Usage 30016 KB (0.02% of 126 GB)
 CPU 0%
 Running Since 2019-Jun-12 13:12:17
 Uptime **1 minute 10 seconds**
 Current Time 2019-Jun-12 13:13:27

6. Confirm the “Hot Storage” in “Data Retention”.

Change Service | NWHOST2100 - Archiver | Config ⌵

General **Data Retention** Files Appliance Service Configuration

Configure the rollover criteria for removing database records from primary storage using an age-based threshold, and schedule the timing for checking if t

1. Configure hot, warm and cold storage
2. Configure collections
3. Define retention rules

Total Hot Storage **47.29 TB** ⚙️ Total Warm Storage Not Configured ⚙️ Cold Storage Not Configured ⚙️

1 Mount Point

Collections

+ - ✎ | ↻

<input type="checkbox"/>	Collection	Usage / Hot Storage	Usage / Warm Storage	Cold Storage	Retention
<input type="checkbox"/>	default	0 B / 44.93 TB (95%)	Disabled	○	No Limit
Total Storage		0 B / 44.93 TB	0 B / 0 B		

Retention Rules

+ - ✎ | ↻ | ⬆ Move Up ⬇ Move Down | Apply ↻ Revert

<input type="checkbox"/>	Order ^	Rule Name	Condition
<input type="checkbox"/>		default	*

7. Reconfigure the following Archiver service to detect and take advantage of all of the free space as described in [Task 5 - \(Optional\) Reconfigure Storage Configuration for 10G Capture](#).

Configure Storage for Network (Packet) Decoder

The following scenario configures storage on two, non-encrypted, 15-Drive DACs for a Network Decoder for 10G Capture physical host.

1. Execute the `raidList` command.
 - a. Record the Controller Number, Enclosure Number, In Use, Drives, and Devices.

You should see the following information.

```
In Use: FALSE
Devices: <empty>
```

- b. Verify the Drive Count, Size, and Vendor.

The following example illustrates what you should see before you create a RAID array.

Properties for NWHOST2100 –

raidList Parameters

Message Help

list drive shelves attached to this appliance
security.roles: appliance.manage

Response Output

```
Drives: 931.511 GB x 2
1.818 TB x 2
Devices: sda
sdb

Controller 1, Enclosure 0
Vendor: EMC
Model: ESES Enclosure
In Use: false
Drives: 3.637 TB x 15
Devices:

Controller 1, Enclosure 2
Vendor: EMC
Model: ESES Enclosure
In Use: false
Drives: 3.637 TB x 15
Devices:
```

2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
 - Parameters for the first enclosure:
`controller=1 enclosure=0 scheme=decoder commit=1`

raidNew	Parameters	controller=1 enclosure=0 scheme=decoder commit=1
Message Help		
parameters: controller - <int32, {enum-one:0,1}> Controller the shelf is attached to enclosure - <uint32, {enum-one:32,0,2}> Enclosure number of the shelf to clear scheme - <string, {enum-one:decoder logdecoder concentrator archiver network-hybrid log-hybrid}> Type of RAID volumes to allocate		
Response Output		
/opt/MegaRAID/perccli/perccli64 /c1 add vd r5 drives=0:0,0:1,0:2 wb ra cached Strip=128 Controller = 1 Status = Success Description = Add VD Succeeded		
/opt/MegaRAID/perccli/perccli64 /c1 add vd r5 drives=0:3,0:4,0:5,0:6,0:7,0:8,0:9,0:10,0:11,0:12,0:13,0:14 wb ra cached Strip=128 Controller = 1 Status = Success Description = Add VD Succeeded		

- Parameters for the second enclosure:

```
controller=1 enclosure=2 scheme=decoder commit=1
```

raidList Parameters

Message Help

list drive shelves attached to this appliance
security.roles: appliance.manage

Response Output

```
Devices: sda
         sdb

Controller 1, Enclosure 0
Vendor: EMC
Model: ESES Enclosure
In Use: true
Drives: 3.637 TB x 15
Devices: sdc
         sdd

Controller 1, Enclosure 2
Vendor: EMC
Model: ESES Enclosure
In Use: true
Drives: 3.637 TB x 15
Devices: sde
         sdf
```

- Use the `raidList` command to display block devices for enclosures so you can verify `In Use: TRUE`.

- SSH to the Network Decoder and use the `lsblk` command to confirm sizes for **decodersmall**.

```
[root@NWHOST2000 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0  931G  0 disk
├─sda1                               8:1    0    1G  0 part /boot
├─sda2                               8:2    0  930G  0 part
│   ├─netwitness_vg00-root           253:0  0   30G  0 lvm  /
│   ├─netwitness_vg00-swap           253:1  0    4G  0 lvm  [SWAP]
│   ├─netwitness_vg00-nwhome         253:2  0  2.7T  0 lvm  /var/netwitness
│   ├─netwitness_vg00-varlog         253:3  0   10G  0 lvm  /var/log
│   └─netwitness_vg00-usrhome         253:4  0   10G  0 lvm  /home
sdb                                  8:16   0  1.8T  0 disk
├─sdb1                               8:17   0  1.8T  0 part
└─netwitness_vg00-nwhome             253:2  0  2.7T  0 lvm  /var/netwitness
sdc                                  8:32   0  7.3T  0 disk
sdd                                  8:48   0   40T  0 disk
sde                                  8:64   0  7.3T  0 disk
sdf                                  8:80   0   40T  0 disk
```

Note: For RAID configuration, when you use the decoder for 10G Capture you use **decoder** for both enclosures for performance reasons. When you do not use the **decoder** for 10G Capture, you use the decoder and archiver for the enclosures to maximize storage for because the second enclosure is a single RAID under the **archiver** configuration.

- Execute the `partNew` command to create the **decodersmall** partition first (decoder dir, index, metadb, sessiondb) (First Enclosure, SDC, SDD) with the following parameters.
`name=sdC service=decoder volume=decodersmall commit=1`

partNew Parameters `name=sdcsdd service=decoder volume=decoderssmall commit=1`

Message Help

name - <string, (enum-one:sdcsdd,sde,sdf)> block device name
 service - <string, (enum-one:archiver|concentrator|decoder|logdecoder)> service that will use storage
 volume - <string, optional, (enum-one:index|concentrator|decoderssmall|decoder|logdecoderssmall|logdecoder|archiver)> volume to create
 commit - <bool, optional> commit changes

Response Output

```
Logical volume "decoroot" created.
/sbin/mkfs.xfs /dev/decoderssmall/decoroot
meta-data=/dev/decoderssmall/decoroot isize=512  agcount=4, agsize=655360 blks
         =             sectsz=512  attr=2, projid32bit=1
         =             crc=1      finobt=0, sparse=0
data     =             bsize=4096  blocks=2621440, imaxpct=25
         =             sunit=0    swidth=0 blks
naming   =version 2      bsize=4096  ascii-ci=0 ftype=1
log      =internal log   bsize=4096  blocks=2560, version=2
         =             sectsz=512  sunit=0 blks, lazy-count=1
realtime =none          extsz=4096  blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/decoder
/bin/mount /var/netwitness/decoder
/sbin/lvcreate -y -n index -L 30G decoderssmall
Logical volume "index" created.
/sbin/mkfs.xfs /dev/decoderssmall/index
meta-data=/dev/decoderssmall/index isize=512  agcount=4, agsize=1966080 blks
         =             sectsz=512  attr=2, projid32bit=1
         =             crc=1      finobt=0, sparse=0
data     =             bsize=4096  blocks=7864320, imaxpct=25
```

```
[root@NWHOST2000 ~]# df -h
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/netwitness_vg00-root          30G   2.5G   28G   9% /
devtmpfs                                  63G    0    63G   0% /dev
tmpfs                                      63G   12K   63G   1% /dev/shm
tmpfs                                      63G   26M   63G   1% /run
tmpfs                                      63G    0    63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-nwhome        2.7T   98M   2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-varlog        10G   49M   10G   1% /var/log
/dev/mapper/netwitness_vg00-usrhome        10G   33M   10G   1% /home
/dev/sda1                                  1014M  88M   927M   9% /boot
tmpfs                                      13G    0    13G   0% /run/user/0
/dev/mapper/decoderssmall-decoroot         10G   33M   10G   1% /var/netwitness/decoder
/dev/mapper/decoderssmall-index           30G   33M   30G   1% /var/netwitness/decoder/index
/dev/mapper/decoderssmall-sessiondb       600G   33M   600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decoderssmall-metadb          6.7T   33M   6.7T   1% /var/netwitness/decoder/metadb
```

- Execute the `partNew` command to create the decoder volume (packetdb) (First Enclosure, SDC, SDD) with the following parameters.

```
name==sdd service=decoder volume=decoder commit=1
```

partNew Parameters name=**sdd** service=**decoder** volume=**decoder** commit=1

Message Help

name - <string, {enum-one:sd, **sdd**, sde, sdf}> block device name
 service - <string, {enum-one:archiver | concentrator | **decoder** | logdecoder}> service that will use storage
 volume - <string, optional, {enum-one:index | concentrator | decodersmall | **decoder** | logdecodersmall | logdecoder | archiver}> volume to create
 commit - <bool, optional> commit changes

Response Output

```
/sbin/parted -s /dev/sdd mklabel gpt
/sbin/parted -s -a optimal /dev/sdd mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdd1
Physical volume "/dev/sdd1" successfully created.
/sbin/vgcreate -f decoder /dev/sdd1
Volume group "decoder" successfully created
/sbin/lvcreate -y -n packetdb -l 100%FREE decoder
Logical volume "packetdb" created.
/sbin/mkfs.xfs /dev/decoder/packetdb
meta-data=/dev/decoder/packetdb isize=512 agcount=41, agsize=268435455 blks
=          sectsz=512 attr=2, projid32bit=1
=          crc=1 finobt=0, sparse=0
data =          bsize=4096 blocks=10742791168, imaxpct=5
=          sunit=0 swidth=0 blks
naming =version 2          bsize=4096 ascii-ci=0 ftype=1
log  =internal log        bsize=4096 blocks=521728, version=2
=          sectsz=512 sunit=0 blks, lazy-count=1
realtime=none            extsz=4096 blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/decoder/packetdb
/bin/mount /var/netwitness/decoder/packetdb
```

```
[root@NWHOST2000 ~]# df -h
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/netwitness_vg00-root           30G   2.5G   28G   9% /
devtmpfs                                  63G    0    63G   0% /dev
tmpfs                                       63G   12K   63G   1% /dev/shm
tmpfs                                       63G   26M   63G   1% /run
tmpfs                                       63G    0    63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-nwhome         2.7T   98M   2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-varlog         10G   50M   10G   1% /var/log
/dev/mapper/netwitness_vg00-usrhome        10G   33M   10G   1% /home
/dev/sda1                                  1014M  88M   927M   9% /boot
tmpfs                                       13G    0    13G   0% /run/user/0
/dev/mapper/decodersmall-decoroot          10G   33M   10G   1% /var/netwitness/decoder
/dev/mapper/decodersmall-index             30G   33M   30G   1% /var/netwitness/decoder/index
/dev/mapper/decodersmall-sessiondb        600G   33M   600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decodersmall-metadb           6.7T   33M   6.7T   1% /var/netwitness/decoder/metadb
/dev/mapper/decoder-packetdb              41T   34M   41T   1% /var/netwitness/decoder/packetdb
```

In the following example, the following partions are created for SDC, SDD (Enclosure 0).

```
[root@NWHOST2000 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0      0  931G  0 disk
├─sda1                               8:1      0    1G  0 part /boot
├─sda2                               8:2      0  930G  0 part
│   ├─netwitness_vg00-root           253:0    0   30G  0 lvm /
│   ├─netwitness_vg00-swap           253:1    0    4G  0 lvm [SWAP]
│   ├─netwitness_vg00-nwhome         253:2    0  2.7T  0 lvm /var/netwitness
│   ├─netwitness_vg00-varlog         253:3    0   10G  0 lvm /var/log
│   └─netwitness_vg00-usrhome        253:4    0   10G  0 lvm /home
sdb                                  8:16     0  1.8T  0 disk
├─sdb1                               8:17     0  1.8T  0 part
│   └─netwitness_vg00-nwhome         253:2    0  2.7T  0 lvm /var/netwitness
sdc                                  8:32     0  7.3T  0 disk
├─sdc1                               8:33     0  7.3T  0 part
│   ├─decodersmall-decoroot          253:5    0   10G  0 lvm /var/netwitness/decoder
│   ├─decodersmall-index             253:6    0   30G  0 lvm /var/netwitness/decoder/index
│   ├─decodersmall-sessiondb         253:7    0  600G  0 lvm /var/netwitness/decoder/sessiondb
│   └─decodersmall-metadb            253:8    0  6.7T  0 lvm /var/netwitness/decoder/metadb
sdd                                  8:48     0   40T  0 disk
├─sdd1                               8:49     0   40T  0 part
│   └─decoder-packetdb               253:9    0   40T  0 lvm /var/netwitness/decoder/packetdb
sde                                  8:64     0  7.3T  0 disk
sdf                                  8:80     0   40T  0 disk
```

At this point, you add the second DAC enclosure.

- Execute the `partNew` command to create the `decodersmall` partition first (Second Enclosure, SDE, SDF) with the following parameters.

```
name=sde service=decoder volume=decodersmall commit=1
```

Properties for 11 mtlInxwpacket01 - Decoder (DECODER) /deviceappliance/appliance.

partNew Parameters name=sde service=decoder volume=decodersmall commit=1

Message Help

name - <string, {enum-one:sdc,sdd,sde,sdf}> block device name
 service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
 volume - <string, optional, {enum-one:index|concentrator|decodersmall|decoder|logdecodersmall|logdecoder|archiver}> volume to create
 commit - <bool, optional> commit changes

Response Output

```
/sbin/parted -s /dev/sde mklabel gpt
/sbin/parted -s -a optimal /dev/sde mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sde1
Physical volume "/dev/sde1" successfully created.
/sbin/vgcreate -f decodersmall0 /dev/sde1
Volume group "decodersmall0" successfully created
/sbin/lvcreate -y -n index -L 30G decodersmall0
Logical volume "index" created.
/sbin/mkfs.xfs /dev/decodersmall0/index
meta-data=/dev/decodersmall0/index isize=512  agcount=4, agsize=1966080 blks
=          sectsz=512  attr=2, projid32bit=1
=          crc=1      finobt=0, sparse=0
data      =          bsize=4096  blocks=7864320, imaxpct=25
=          sunit=0   swidth=0 blks
naming    =version 2      bsize=4096  ascii-ci=0  ftype=1
log       =internal log  bsize=4096  blocks=3840, version=2
=          sectsz=512  sunit=0 blks, lazy-count=1
realtime  =none         extsz=4096  blocks=0, rtextents=0
/sbin/mkdir -p /var/netwitness/decoder/index0
/bin/mount /var/netwitness/decoder/index0
```

```
[root@NWHOST2000 ~]# df -h
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/netwitness_vg00-root          30G  2.5G   28G   9% /
devtmpfs                                  63G    0   63G   0% /dev
tmpfs                                      63G  12K   63G   1% /dev/shm
tmpfs                                      63G  26M   63G   1% /run
tmpfs                                      63G    0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-nwhome        2.7T   98M  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-varlog         10G   50M   10G   1% /var/log
/dev/mapper/netwitness_vg00-usrhome        10G   33M   10G   1% /home
/dev/sda1                                  1014M  88M  927M   9% /boot
tmpfs                                      13G    0   13G   0% /run/user/0
/dev/mapper/decodersmall-decoroot         10G   33M   10G   1% /var/netwitness/decoder
/dev/mapper/decodersmall-index            30G   33M   30G   1% /var/netwitness/decoder/index
/dev/mapper/decodersmall-sessiondb        600G   33M  600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decodersmall-metadb           6.7T   33M  6.7T   1% /var/netwitness/decoder/metadb
/dev/mapper/decoder-packetdb              41T   34M   41T   1% /var/netwitness/decoder/packetdb
```

8. Execute the `partNew` command to create the `packetdb` decoder volume (Second Enclosure SDE, SDF) with the following parameters.

```
name=sdf service=decoder volume=decoder commit=1
```

partNew Parameters name=sdf service=decoder volume=decoder commit=1

Message Help

```
name - <string, {enum-one:sdc,sdd,sde,sdf}> block device name
service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
volume - <string, optional, {enum-one:index|concentrator|decodersmall|decoder|logdecodersmall|logdecoder|archiver}> volume to create
commit - <bool, optional> commit changes
```

Response Output

```
/sbin/parted -s /dev/sdf mklabel gpt
/sbin/parted -s -a optimal /dev/sdf mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdf1
Physical volume "/dev/sdf1" successfully created.
/sbin/vgcreate -f decoder0 /dev/sdf1
Volume group "decoder0" successfully created
/sbin/lvcreate -y -n packetdb -l 100%FREE decoder0
Logical volume "packetdb" created.
/sbin/mkfs.xfs /dev/decoder0/packetdb
meta-data=/dev/decoder0/packetdb isize=512  agcount=41, agsize=268435455 blks
=          sectsz=512  attr=2, projid32bit=1
=          crc=1      finobt=0, sparse=0
data =          bsize=4096  blocks=10742791168, imaxpct=5
=          sunit=0    swidth=0 blks
naming  =version 2          bsize=4096  ascii-ci=0  ftype=1
log     =internal log      bsize=4096  blocks=521728, version=2
=          sectsz=512   sunit=0 blks, lazy-count=1
realtime=none          extsz=4096  blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/decoder/packetdb0
/bin/mount /var/netwitness/decoder/packetdb0
```



```
[root@NWHOST2000 ~]# df -h
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/netwitness_vg00-root          30G  2.5G   28G   9% /
devtmpfs                                  63G    0   63G   0% /dev
tmpfs                                      63G  12K   63G   1% /dev/shm
tmpfs                                      63G  27M   63G   1% /run
tmpfs                                      63G    0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-nwhome        2.7T  98M  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-varlog        10G  50M   10G   1% /var/log
/dev/mapper/netwitness_vg00-usrhome       10G  33M   10G   1% /home
/dev/sda1                                 1014M  88M  927M   9% /boot
tmpfs                                      13G    0   13G   0% /run/user/0
/dev/mapper/decodersmall-decoroot         10G  33M   10G   1% /var/netwitness/decoder
/dev/mapper/decoderssmall-index           30G  33M   30G   1% /var/netwitness/decoder/index
/dev/mapper/decoderssmall-sessiondb       600G  33M  600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decoderssmall-metadb          6.7T  33M  6.7T   1% /var/netwitness/decoder/metadb
/dev/mapper/decoder-packetdb              41T  34M   41T   1% /var/netwitness/decoder/packetdb
/dev/mapper/decoderssmall0-index           30G  33M   30G   1% /var/netwitness/decoder/index0
/dev/mapper/decoderssmall0-sessiondb      600G  33M  600G   1% /var/netwitness/decoder/sessiondb0
/dev/mapper/decoderssmall0-metadb         6.7T  33M  6.7T   1% /var/netwitness/decoder/metadb0
/dev/mapper/decoder0-packetdb             41T  34M   41T   1% /var/netwitness/decoder/packetdb0
```

```
[root@NWHOST2000 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0  931G  0 disk
├─sda1                               8:1    0    1G  0 part /boot
├─sda2                               8:2    0  930G  0 part
│   ├─netwitness_vg00-root            253:0  0    30G  0 lvm /
│   ├─netwitness_vg00-swap            253:1  0    4G  0 lvm [SWAP]
│   ├─netwitness_vg00-nwhome          253:2  0  2.7T  0 lvm /var/netwitness
│   ├─netwitness_vg00-varlog          253:3  0   10G  0 lvm /var/log
│   └─netwitness_vg00-usrhome          253:4  0   10G  0 lvm /home
sdb                                  8:16   0  1.8T  0 disk
├─sdb1                               8:17   0  1.8T  0 part
└─┬─netwitness_vg00-nwhome            253:2  0  2.7T  0 lvm /var/netwitness
sdc                                  8:32   0  7.3T  0 disk
├─sdc1                               8:33   0  7.3T  0 part
│   ├─decoderssmall-decoroot          253:5  0   10G  0 lvm /var/netwitness/decoder
│   ├─decoderssmall-index             253:6  0   30G  0 lvm /var/netwitness/decoder/index
│   ├─decoderssmall-sessiondb         253:7  0  600G  0 lvm /var/netwitness/decoder/sessiondb
│   └─decoderssmall-metadb            253:8  0  6.7T  0 lvm /var/netwitness/decoder/metadb
sdd                                  8:48   0   40T  0 disk
├─sdd1                               8:49   0   40T  0 part
└─┬─decoder-packetdb                 253:9  0   40T  0 lvm /var/netwitness/decoder/packetdb
sde                                  8:64   0   7.3T  0 disk
├─sde1                               8:65   0   7.3T  0 part
│   ├─decoderssmall10-index           253:10 0   30G  0 lvm /var/netwitness/decoder/index0
│   ├─decoderssmall10-sessiondb       253:11 0  600G  0 lvm /var/netwitness/decoder/sessiondb0
│   └─decoderssmall10-metadb          253:12 0  6.7T  0 lvm /var/netwitness/decoder/metadb0
sdf                                  8:80   0   40T  0 disk
├─sdf1                               8:81   0   40T  0 part
└─┬─decoder0-packetdb                253:13 0   40T  0 lvm /var/netwitness/decoder/packetdb0
```

9. Execute the `srvAlloc` command with the following parameters to add the storage information into the Service Configuration settings.

- `service=decoder volume=decodersmall commit=1`
- `service=decoder volume=decoderssmall0 commit=1`
- `service=decoder volume=decoder commit=1`
- `service=decoder volume=decoder0 commit=1`

srvAlloc ▾ Parameters service=decoder commit=1 volume=decoder0

Message Help

service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
 volume - <string, {enum-one:decoder,decoder0,decodersmall,decodersmall0,netwitness_vg00}> volume group name
 commit - <bool, optional> commit changes

Response Output

Set /database/config/packet.dir to /var/netwitness/decoder/packetdb==38 TB:/var/netwitness/decoder/packetdb0==38.01 TB

/database/config	NWHOST2000 - Concentrator
meta.compression	none
meta.compression.level	0
meta.dir	/var/netwitness/decoder/metadb==6.3 TB;/var/netwitness/decoder/metadb0==6.32 TB
meta.dir.cold	
meta.dir.warm	
meta.file.size	auto
meta.files	auto
meta.free.space.min	23 GB
meta.index.fidelity	4
meta.integrity.flush	sync
meta.write.block.size	64 KB
packet.compression	none
packet.compression.level	0
packet.dir	/var/netwitness/decoder/packetdb==38 TB;/var/netwitness/decoder/packetdb0==38.01 TB
packet.dir.cold	
packet.dir.warm	
packet.file.size	auto
packet.file.type	pcapng
packet.files	auto
packet.free.space.min	23 GB
packet.index.fidelity	1
packet.integrity.flush	sync
packet.write.block.size	64 KB
session.dir	/var/netwitness/decoder/sessiondb==569.71 GB;/var/netwitness/decoder/sessiondb0==569.72 GB
session.dir.cold	

10. Reconfigure the following Network Decoder service and its database to detect and take advantage of all of the free space as described in [Task 5 - \(Optional\) Reconfigure Storage Configuration for 10G Capture](#).

Configure Storage for Network Concentrator

The following scenario configures storage on one, non-encrypted, 15-Drive DAC for a Network Concentrator physical host.

1. Execute the `raidList` command.

`raidList`

Message Help

```
list drive shelves attached to this appliance
security.roles: appliance.manage
```

Response Output

```
Controller 0, Enclosure 32
Vendor: DP
Model: BP13G+EXP
In Use: true
Drives: 931.511 GB x 2
        1.818 TB x 2
Devices: sda
         sdb
```

```
Controller: 1, Enclosure 6
Vendor: EMC
Model: ESES Enclosure
In Use: false
Drives: 186.309 GB x 6
        3.637 TB x 9
Devices:
```

- Execute the `raidNew` command with the following parameters.

`controller=1 enclosure=6 scheme=concentrator`

raidNew Parameters `controller=1 enclosure=6 scheme=concentrator commit=1`

Message Help

parameters:
 controller - <int32, {enum-one:0,1}> Controller the shelf is attached to
 enclosure - <uint32, {enum-one:32,6}> Enclosure number of the shelf to clear
 scheme - <string, {enum-one:decoder|logdecoder|concentrator|archiver|network-hybrid|log-hybrid}> Type of RAID volumes to allocate

Response Output

```
/opt/MegaRAID/perccli/perccli64 /c1 add vd r5 drives=6:0,6:1,6:2,6:3,6:4,6:5 wb ra cached Strip=128
Controller = 1
Status = Success
Description = Add VD Succeeded

/opt/MegaRAID/perccli/perccli64 /c1 add vd r6 drives=6:6,6:7,6:8,6:9,6:10,6:11,6:12,6:13,6:14 wb ra cached Strip=128
Controller = 1
Status = Success
Description = Add VD Succeeded
```

```
[root@NWHOST1500 ~]# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   931G  0 disk
├─sda1                               8:1    0     1G  0 part /boot
└─sda2                               8:2    0   930G  0 part
   ├─netwitness_vg00-root             253:0    0    30G  0 lvm /
   ├─netwitness_vg00-swap             253:1    0     4G  0 lvm [SWAP]
   ├─netwitness_vg00-nwhome           253:2    0   2.7T  0 lvm /var/netwitness
   ├─netwitness_vg00-varlog           253:3    0    10G  0 lvm /var/log
   └─netwitness_vg00-usrhome           253:4    0    10G  0 lvm /home
sdb                                  8:16   0   1.8T  0 disk
├─sdb1                               8:17   0   1.8T  0 part
└─netwitness_vg00-nwhome             253:2    0   2.7T  0 lvm /var/netwitness
sdc                                  8:32   0  928.8G  0 disk
sdd                                  8:48   0   25.5T  0 disk
[root@NWHOST1500 ~]#
```

- Execute the `partNew` command to create the **concentrator** partition first with the following parameters. You must create the **concentrator** volume before **index** volume or it will fail.

```
name=sdd service=concentrator volume=concentrator commit=1
```

partNew Parameters name=sdd service=concentrator volume=concentrator commit=1

Message Help

parameters:
 name - <string, {enum-one:sdc,sdd}> block device name
 service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
 volume - <string, optional, {enum-one:index|concentrator|decodersmall|decoder|logdecodersmall|logdecoder|archiver}> volume to create

Response Output

```
/sbin/parted -s /dev/sdd mklabel gpt
/sbin/parted -s -a optimal /dev/sdd mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdd1
Physical volume "/dev/sdd1" successfully created.
/sbin/vgcreate -f concentrator /dev/sdd1
Volume group "concentrator" successfully created
/sbin/lvcreate -y -n root -L 30G concentrator
Logical volume "root" created.
/sbin/mkfs.xfs /dev/concentrator/root
meta-data=/dev/concentrator/root isize=512 agcount=4, agsize=1966080 blks
=          sectsz=512 attr=2, projid32bit=1
=          crc=1  finobt=0, sparse=0
data      =          bsize=4096 blocks=7864320, imaxpct=25
=          sunit=0  swidth=0 blks
naming    =version 2          bsize=4096 ascii-ci=0 ftype=1
log       =internal log      bsize=4096 blocks=3840, version=2
=          sectsz=512  sunit=0 blks, lazy-count=1
realtime  =none             extsz=4096  blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/concentrator
/bin/mount /var/netwitness/concentrator
```

```
[root@NWHOST1500 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0      0  931G  0 disk
├─sda1                               8:1      0    1G  0 part /boot
└─sda2                               8:2      0  930G  0 part
   ├─netwitness_vg00-root             253:0    0   30G  0 lvm /
   ├─netwitness_vg00-swap             253:1    0    4G  0 lvm [SWAP]
   ├─netwitness_vg00-nwhome           253:2    0  2.7T  0 lvm /var/netwitness
   ├─netwitness_vg00-varlog           253:3    0   10G  0 lvm /var/log
   └─netwitness_vg00-usrhome           253:4    0   10G  0 lvm /home
sdb                                  8:16     0   1.8T  0 disk
├─sdb1                               8:17     0   1.8T  0 part
└─netwitness_vg00-nwhome             253:2    0  2.7T  0 lvm /var/netwitness
sdc                                  8:32     0  928.8G  0 disk
sdd                                  8:48     0  25.5T  0 disk
├─sdd1                               8:49     0  25.5T  0 part
│  ├─concentrator-root               253:5    0   30G  0 lvm /var/netwitness/concentrator
│  ├─concentrator-sessiondb          253:6    0   600G  0 lvm /var/netwitness/concentrator/sessiondb
│  └─concentrator-metadb              253:7    0  24.9T  0 lvm /var/netwitness/concentrator/metadb
```

- Execute the partNew command with the following parameters to create an index on SSDs.

```
name=sdc service=concentrator volume=index commit=1
```

partNew Parameters name=sdc service=concentrator volume=index commit=1

Message Help

parameters:

name - <string, {enum-one:sdc,sdd}> block device name
 service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
 volume - <string, optional, {enum-one:index|concentrator|decodersmall|decoder|logdecodersmall|logdecoder|archiver}> volume to create

Response Output

```
/sbin/parted -s /dev/sdc mklabel gpt
/sbin/parted -s -a optimal /dev/sdc mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdc1
Physical volume "/dev/sdc1" successfully created.
/sbin/vgcreate -f index /dev/sdc1
Volume group "index" successfully created
/sbin/lvcreate -y -n index -l 100%FREE index
Wiping xfs signature on /dev/index/index.
Logical volume "index" created.
/sbin/mkfs.xfs /dev/index/index
meta-data=/dev/index/index  isize=512  agcount=4, agsize=60866304 blks
          =                  sectsz=4096  attr=2, projid32bit=1
          =                  crc=1      finobt=0, sparse=0
data      =                  bsize=4096  blocks=243465216, imaxpct=25
          =                  sunit=0    swidth=0 blks
naming    =version 2        bsize=4096  ascii-ci=0 ftype=1
log       =internal log    bsize=4096  blocks=118879, version=2
          =                  sectsz=4096  sunit=1 blks, lazy-count=1
realtime  =none           extsz=4096  blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/concentrator/index
```

```
[root@NWHOST1500 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0  931G  0 disk
├─sda1                                8:1    0    1G  0 part /boot
└─sda2                                8:2    0  930G  0 part
   ├─netwitness_vg00-root              253:0  0    30G  0 lvm  /
   ├─netwitness_vg00-swap              253:1  0     4G  0 lvm  [SWAP]
   ├─netwitness_vg00-nwhome            253:2  0  2.7T  0 lvm  /var/netwitness
   ├─netwitness_vg00-varlog            253:3  0    10G  0 lvm  /var/log
   └─netwitness_vg00-usrhome            253:4  0    10G  0 lvm  /home
bdb                                  8:16   0   1.8T  0 disk
├─sdb1                                8:17   0   1.8T  0 part
└─netwitness_vg00-nwhome              253:2  0  2.7T  0 lvm  /var/netwitness
sdc                                  8:32   0  928.8G  0 disk
├─sdc1                                8:33   0  928.8G  0 part
└─index-index                         253:8   0  928.8G  0 lvm  /var/netwitness/concentrator/index
sdd                                  8:48   0  25.5T  0 disk
├─sdd1                                8:49   0  25.5T  0 part
├─concentrator-root                  253:5   0    30G  0 lvm  /var/netwitness/concentrator
├─concentrator-sessiondb             253:6   0   600G  0 lvm  /var/netwitness/concentrator/sessiondb
└─concentrator-metadb                 253:7   0   24.9T  0 lvm  /var/netwitness/concentrator/metadb
```

```
[root@NWHOST1500 ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/netwitness_vg00-root  30G    2.1G    28G   7% /
devtmpfs                  63G         0    63G   0% /dev
tmpfs                     63G     12K    63G   1% /dev/shm
tmpfs                     63G     10M    63G   1% /run
tmpfs                     63G         0    63G   0% /sys/fs/cgroup
/dev/sda1                 1014M     91M    924M   9% /boot
/dev/mapper/netwitness_vg00-varlog  10G     52M    10G   1% /var/log
/dev/mapper/netwitness_vg00-usrhome  10G     33M    10G   1% /home
/dev/mapper/netwitness_vg00-nwhome  2.7T     98M    2.7T   1% /var/netwitness
tmpfs                    13G         0    13G   0% /run/user/0
/dev/mapper/concentrator-root      30G     33M    30G   1% /var/netwitness/concentrator
/dev/mapper/concentrator-sessiondb 600G     33M   600G   1% /var/netwitness/concentrator/sessiondb
/dev/mapper/concentrator-metadb     25T     33M    25T   1% /var/netwitness/concentrator/metadb
/dev/mapper/index-index           929G     33M   929G   1% /var/netwitness/concentrator/index
```

- Execute the `srvAlloc` command with the following parameters.

```
service=concentrator volume=index commit=1
```

Parameters

Message Help

parameters:

service - <string, {enum-one:archiver | concentrator | decoder | logdecoder}> service that will use storage

volume - <string, {enum-one:concentrator,index,netwitness_vg00}> volume group name

commit - <bool, optional> commit changes

Response Output

```
Set /index/config/index.dir to /var/netwitness/concentrator/index==881.87 GB
```

NWHOST1500 - Concentrator		Explore
NWHOST1500 - Concentrator	/index/config	NWHOST1500 - Concentrator (CONC)
NWHOST1500 - Concentrator (CONC)	index.dir	/var/netwitness/concentrator/index==881.87 GB
concentrator	index.dir.cold	
connections	index.dir.warm	
database	index.slices.open	42
deviceappliance	page.compression	huffhybrid
index	reindex.enable	true
config	save.session.count	auto

- Execute the `srvAlloc` command with the following parameters.

```
service=concentrator volume=concentrator commit=1
```

srvAlloc ▾ Parameters `service=concentrator volume=concentrator commit=1`

Message Help

parameters:

- service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
- volume - <string, {enum-one:concentrator,index,netwitness_vg00}> volume group name
- commit - <bool, optional> commit changes

Response Output

Set /database/config/meta.dir to /var/netwitness/concentrator/metadb==23.6 TB
Set /database/config/session.dir to /var/netwitness/concentrator/sessiondb==569.72 GB

NWHOST1500 - Concentrator (CONCENTRAT)		Explore
NWHOST1500 - Concentrator		
NWHOST1500 - Concentrator (CONC)		
concentrator		
connections		
database		
config		
stats		
deviceappliance		
index		
logs		
rest		
sdk		
services		
storedproc		
sys		
users		
/database/config		NWHOST1500 - Concentrator (CONC)
hash.algorithm		none
hash.databases		session,meta
hash.dir		
manifest.dir		
meta.compression		none
meta.compression.level		0
meta.dir		/var/netwitness/concentrator/metadb==23.6 TB
meta.dir.cold		
meta.dir.warm		
meta.file.size		auto
meta.files		auto
meta.free.space.min		23 GB
meta.index.fidelity		4
meta.integrity.flush		sync
meta.write.block.size		64 KB
session.dir		/var/netwitness/concentrator/sessiondb==569.72 GB

7. Reconfigure the following Network Concentrator service and its database to detect and take advantage of all of the free space as described in [Task 5 - \(Optional\) Reconfigure Storage Configuration for 10G Capture](#).

Configure Storage for Log Decoder Hybrid

The following scenario configures storage on one, non-encrypted, 15-Drive DAC for a Log Decoder Hybrid physical host.

1. Execute the `raidList` command.

`raidList`

Message Help

```
list drive shelves attached to this appliance
security.roles: appliance.manage
```

Response Output

```
Controller 0, Enclosure 32
```

```
Vendor: DP
Model: BP13G+EXP
In Use: true
Drives: 745.21 GB x 2
        931.511 GB x 4
        5.457 TB x 8
```

```
Devices: sda
```

```
sdb
sdc
sdd
sde
```

```
Controller 1, Enclosure 31
```

```
Vendor: EMC
Model: ESES Enclosure
In Use: false
Drives: 3.637 TB x 15
Devices:
```

2. Execute the `raidNew` command with the following parameters.
`controller=1 enclosure=31 scheme=log-hybrid commit=1`

raidNew Parameters controller=1 enclosure=31 scheme=log-hybrid commit=1

Message Help

controller - <int32, {enum-one:0,1}> Controller the shelf is attached to
 enclosure - <uint32, {enum-one:32,31}> Enclosure number of the shelf to clear
 scheme - <string, {enum-one:decoder|logdecoder|concentrator|archiver|network-hybrid|log-hybrid}> Type of RAID volumes to allocate
 preferSecure - <bool, optional, {bool:0,1,yes,no,true,false,on,off}> Prefer creation of a secure array given compatible physical drives and a controller with a security key set

Response Output

```
/opt/MegaRAID/perccli/perccli64 /c1 add vd r5 drives=31:0,31:1,31:2,31:3,31:4,31:5,31:6 wb ra cached Strip=128
Controller = 1
Status = Success
Description = Add VD Succeeded
```

```
/opt/MegaRAID/perccli/perccli64 /c1 add vd r5 drives=31:7,31:8,31:9,31:10,31:11,31:12,31:13,31:14 wb ra cached Strip=128
Controller = 1
Status = Success
Description = Add VD Succeeded
```

```
[root@NWHOST1700 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   931G  0 disk
├─sda1                                8:1    0     1G  0 part /boot
├─sda2                                8:2    0   930G  0 part
│   ├─netwitness_vg00-root            253:0    0    30G  0 lvm /
│   ├─netwitness_vg00-swap           253:1    0     4G  0 lvm [SWAP]
│   ├─netwitness_vg00-nwhome         253:11   0   876G  0 lvm /var/netwitness
│   ├─netwitness_vg00-varlog         253:12   0    10G  0 lvm /var/log
│   └─netwitness_vg00-usrhome         253:13   0    10G  0 lvm /home
sdb                                  8:16   0   931G  0 disk
├─sdb1                                8:17   0   931G  0 part
├─decodermeta-vlnwdm                 253:9    0   931G  0 lvm /var/netwitness/decoder/metadb
sdc                                  8:32   0   16.4T  0 disk
├─sdc1                                8:33   0   16.4T  0 part
│   ├─decoderpacket-vlnwdp           253:2    0   16.2T  0 lvm /var/netwitness/decoder/packetdb
│   ├─decoderpacket-vlnwds           253:3    0   100G  0 lvm /var/netwitness/decoder/sessiondb
│   ├─decoderpacket-vlnwdi           253:4    0    50G  0 lvm /var/netwitness/decoder/index
│   └─decoderpacket-vlnwd            253:5    0    30G  0 lvm /var/netwitness/decoder
sdd                                  8:48   0   16.4T  0 disk
├─sdd1                                8:49   0   16.4T  0 part
│   ├─concentrator-vlnwcm            253:6    0   14.9T  0 lvm /var/netwitness/concentrator/metadb
│   ├─concentrator-vlnwcs            253:7    0    1.5T  0 lvm /var/netwitness/concentrator/sessiondb
│   └─concentrator-vlnwc             253:8    0    30G  0 lvm /var/netwitness/concentrator
sde                                  8:64   0  744.6G  0 disk
├─sde1                                8:65   0  744.6G  0 part
├─index-vlnwci                       253:10   0  744.6G  0 lvm /var/netwitness/concentrator/index
sdf                                  8:80   0   21.8T  0 disk
sdg                                  8:96   0   25.5T  0 disk
```


3. Execute the `partNew` command with the following parameters with the following parameters.

- `name=sdf service=concentrator volume=concentrator commit=1`

partNew `name=sdf service=concentrator volume=concentrator commit=1`

Message Help

name - <string, {enum-one:sdf,sdg}> block device name
 service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
 volume - <string, optional, {enum-one:index|concentrator|decodersmall|decoder|logdecodersmall|logdecoder|archiver}> volume to create
 commit - <bool, optional> commit changes

Response Output

```
/sbin/parted -s /dev/sdf mklabel gpt
/sbin/parted -s -a optimal /dev/sdf mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdf1
Physical volume "/dev/sdf1" successfully created.
/sbin/vgcreate -f concentrator0 /dev/sdf1
Volume group "concentrator0" successfully created
```

```
[root@NWHOST1700 ~]# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0  931G  0 disk
├─sda1                               8:1    0    1G  0 part /boot
├─sda2                               8:2    0  930G  0 part
│   ├─netwitness_vg00-root           253:0    0   30G  0 lvm  /
│   ├─netwitness_vg00-swap          253:1    0    4G  0 lvm  [SWAP]
│   ├─netwitness_vg00-nwhome        253:11   0  876G  0 lvm  /var/netwitness
│   ├─netwitness_vg00-varlog        253:12   0   10G  0 lvm  /var/log
│   └─netwitness_vg00-usrhome       253:13   0   10G  0 lvm  /home
sdb                                  8:16   0  931G  0 disk
├─sdb1                               8:17   0  931G  0 part
│   └─decodermeta-vlnwdm            253:9    0  931G  0 lvm  /var/netwitness/decoder/metadb
sdc                                  8:32   0  16.4T  0 disk
├─sdc1                               8:33   0  16.4T  0 part
│   ├─decoderpacket-vlnwdp         253:2    0  16.2T  0 lvm  /var/netwitness/decoder/packetdb
│   ├─decoderpacket-vlnwds         253:3    0   100G  0 lvm  /var/netwitness/decoder/sessiondb
│   ├─decoderpacket-vlnwdi         253:4    0    50G  0 lvm  /var/netwitness/decoder/index
│   └─decoderpacket-vlnwd         253:5    0    30G  0 lvm  /var/netwitness/decoder
sdd                                  8:48   0  16.4T  0 disk
├─sdd1                               8:49   0  16.4T  0 part
│   ├─concentrator-vlnwcm          253:6    0  14.9T  0 lvm  /var/netwitness/concentrator/metadb
│   ├─concentrator-vlnwcs          253:7    0    1.5T  0 lvm  /var/netwitness/concentrator/sessiondb
│   └─concentrator-vlnwc           253:8    0    30G  0 lvm  /var/netwitness/concentrator
sde                                  8:64   0  744.6G  0 disk
├─sde1                               8:65   0  744.6G  0 part
│   └─index-vlnwci                 253:10   0  744.6G  0 lvm  /var/netwitness/concentrator/index
sdf                                  8:80   0  21.8T  0 disk
├─sdf1                               8:81   0  21.8T  0 part
│   ├─concentrator0-sessiondb      253:14   0   600G  0 lvm  /var/netwitness/concentrator/sessiondb0
│   └─concentrator0-metadb         253:15   0   21.2T  0 lvm  /var/netwitness/concentrator/metadb0
sdg                                  8:96   0  25.5T  0 disk
```

- name=sdg service=logdecoder volume=logdecoder commit=1

partNew

Message Help

name - <string, {enum-one:sdf,sdg}> block device name
 service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
 volume - <string, optional, {enum-one:index|concentrator|decodersmall|decoder|logdecodersmall|logdecoder|archiver}> volume to create
 commit - <bool, optional> commit changes

Response Output

```
/sbin/parted -s /dev/sdg mklabel gpt
/sbin/parted -s -a optimal /dev/sdg mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdg1
Physical volume "/dev/sdg1" successfully created.
/sbin/vgcreate -f logdecoder0 /dev/sdg1
Volume group "logdecoder0" successfully created
```

```
[root@NWHOST1700 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                  8:0    0   931G  0 disk
├─sda1                               8:1    0     1G  0 part  /boot
├─sda2                               8:2    0   930G  0 part
│   ├─netwitness_vg00-root           253:0  0     30G  0 lvm    /
│   ├─netwitness_vg00-swap           253:1  0      4G  0 lvm    [SWAP]
│   ├─netwitness_vg00-nwhome         253:11 0   876G  0 lvm    /var/netwitness
│   ├─netwitness_vg00-varlog         253:12 0    10G  0 lvm    /var/log
│   └─netwitness_vg00-usrhome         253:13 0    10G  0 lvm    /home
sdb                                  8:16   0   931G  0 disk
├─sdb1                               8:17   0   931G  0 part
│   └─decodermeta-vlnwdm             253:9  0   931G  0 lvm    /var/netwitness/decoder/metadb
sdc                                  8:32   0  16.4T  0 disk
├─sdc1                               8:33   0  16.4T  0 part
│   ├─decoderpacket-vlnwdp           253:2  0  16.2T  0 lvm    /var/netwitness/decoder/packetdb
│   ├─decoderpacket-vlnwds           253:3  0   100G  0 lvm    /var/netwitness/decoder/sessiondb
│   ├─decoderpacket-vlnwdi           253:4  0    50G  0 lvm    /var/netwitness/decoder/index
│   └─decoderpacket-vlnwd            253:5  0    30G  0 lvm    /var/netwitness/decoder
sdd                                  8:48   0  16.4T  0 disk
├─sdd1                               8:49   0  16.4T  0 part
│   ├─concentrator-vlnwcm            253:6  0   14.9T 0 lvm    /var/netwitness/concentrator/metadb
│   ├─concentrator-vlnwcs            253:7  0    1.5T 0 lvm    /var/netwitness/concentrator/sessiondb
│   └─concentrator-vlnwc             253:8  0    30G  0 lvm    /var/netwitness/concentrator
sde                                  8:64   0  744.6G 0 disk
├─sde1                               8:65   0  744.6G 0 part
│   └─index-vlnwci                   253:10 0  744.6G 0 lvm    /var/netwitness/concentrator/index
sdf                                  8:80   0  21.8T  0 disk
├─sdf1                               8:81   0  21.8T  0 part
│   ├─concentrator0-sessiondb        253:14 0   600G 0 lvm    /var/netwitness/concentrator/sessiondb0
│   └─concentrator0-metadb           253:15 0   21.2T 0 lvm    /var/netwitness/concentrator/metadb0
sdg                                  8:96   0  25.5T  0 disk
├─sdg1                               8:97   0  25.5T  0 part
│   └─logdecoder0-packetdb           253:16 0  25.5T 0 lvm    /var/netwitness/decoder/packetdb0
```

4. Execute the `srvAlloc` command with the following parameters.

- `service=concentrator volume=concentrator0 commit=1`

Parameters

Message Help

```

service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
volume - <string, {enum-one:concentrator,concentrator0,decodermeta,decoderpacket,index,logdecoder0,netwitness_vg00}> volume group name
commit - <bool, optional> commit changes
    
```

Response Output

```

Set /database/config/meta.dir to /var/netwitness/concentrator/metadb=14.08 TB;/var/netwitness/concentrator/metadb0==20.17 TB
Set /database/config/session.dir to /var/netwitness/concentrator/sessiondb=1.41 TB;/var/netwitness/concentrator/sessiondb0==569.72 GB
    
```

Parameter	Value
hash.algorithm	none
hash.databases	session,meta
hash.dir	
manifest.dir	
meta.compression	none
meta.compression.level	0
meta.dir	/var/netwitness/concentrator/metadb=14.08 TB;/var/netwitness/concentrator/metadb0==20.17 TB
meta.dir.cold	
meta.dir.warm	
meta.file.size	auto
meta.files	auto
meta.free.space.min	132 GB
meta.index.fidelity	4
meta.integrity.flush	sync
meta.write.block.size	64 KB
session.dir	/var/netwitness/concentrator/sessiondb=1.41 TB;/var/netwitness/concentrator/sessiondb0==569.72 GB

- `service=logdecoder volume=logdecoder0 commit=1`

5. Reconfigure the following Log Decoder service and its database to detect and take advantage of all of the free space as described in [Task 5 - \(Optional\) Reconfigure Storage Configuration for 10G Capture](#).