NetWitness® Platform

Version 12.5.1.0

Storage Guide

NETWITNESS

Platform

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA and other trademarks are trademarks of RSA Security LLC or its affiliates ("RSA"). For a list of RSA trademarks, go to https://www.rsa.com/en-us/company/rsa-trademarks. Other trademarks are trademarks of their respective owners.

License Agreement

This software and the associated documentation are proprietary and confidential to RSA Security LLC or its affiliates 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 RSA.

It is advised not to deploy third-party repos or perform any change to the underlying NetWitness Operating System that is not part of the supported NetWitness version. Any such change outside of the NetWitness approved image may result in a service or functionality conflict and require a reimage of the NetWitness system to bring NetWitness back to an optimized functional state. In the event a third-party repo is deployed, or other non-supported change is made by the customer without NetWitness approval, the customer takes full responsibility for any system malfunction until the issue can be remediated through troubleshooting efforts or a reimage of the service.

Third-Party Licenses

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

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

Use, copying, and distribution of any RSA Security LLC or its affiliates ("RSA") software described in this publication requires an applicable software license.

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

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Miscellaneous

This product, this software, the associated documentations as well as the contents are subject to NetWitness' standard Terms and Conditions in effect as of the issuance date of this documentation and which can be found at https://www.netwitness.com/standard-form-agreements/.

November, 2024

Contents

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 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 NetWitness Platform Core and Hybrid physical hosts:

- PowerVault (MD 2412) Setup Guide (see the "Enclosure Options" section of "Hardware Description") - NetWitness Community.
- PowerVault (MD 1400) Setup Guide (see the "Enclosure Options" section of "Hardware Description") - NetWitness Community.
- 60-Drive DAC Setup Guide - NetWitness Community.
- 15-Drive DAC Setup Guide - NetWitness Community.

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.

IMPORTANT: For Series 7, Series 6E, MD2412 and MD1400, all drives are all self-encrypting drives. However, encryption should be turned on explicitly by the customer, so that they can own and manage the encryption key.

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

Service	Volume Name	File Systems Created
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/sessiond	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

Volume	Filesystem	Mount Point	Size
archiver	database	/var/netwitness/archiver/database	100% of free space on archiver volume

Performance Recommendations

NetWitness 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.

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.

Enable Encryption on Series 6E or Series 7 Hosts

All drives in Series6E/Series7 and any configured external storage (MD1400 or MD2412) are self-encrypting (SED). However, encryption has to be turned on explicitly by the customer, so they can generate and manage their own encryption keys. NetWitness is not responsible for key generation and management and will not have access to customer's disk encryption keys. If the customer chooses to enable encryption, NetWitness recommends enabling it after completion of external storage configuration (if any).

Refer to Appendix B. Encrypt a Series 6E or Series 7 Core or Hybrid Host (encryptSedVd.py) for details on enabling encryption.

Configure Drive Pack(s)

Note: The terms 'Meta Disk Kit' and 'Meta Drive Pack' mean the same and are interchangeable.

Benefits of Series 6/6E and Series 7 Drive Pack

Series 6/6E Hosts

You can add additional drives to the Series 6 or 6E appliances to accommodate various use cases. These drives provide the capability for the decoder meta or concentrator index volumes to reside on the appliance. Each Meta Disk Kit has 3 drives. A maximum of 2 Meta Disk Kits can be installed on series 6/6E appliances. The Meta cache or index size determines the number of Meta Disk Kits. A standard Series 6/6E appliance has 4 drives in slots 0,1,2 and 3. The remaining slots, 4 through 9 are empty (highlighted in red in the Series 6/6E Disk Layout image below). These slots are used to install the Meta Disk Kit(s).

Series 7 Hosts

You can add additional drives to the Series 7 to accommodate various use cases. These drives provide the capability for the decoder meta to reside on the appliance. Each Meta Disk Kit has 3 drives. A maximum of 2 Meta Disk Kits can be installed on series 7 appliances. The Meta cache determines the number of Meta Disk Kits. A standard Series 7 appliance has 4 drives in slots 0,1,2 and 3. The remaining slots, 4 through 9 are empty (highlighted in red in the Series 7 Disk Layout image below). These slots are used to install the Meta Disk Kit(s).

NW Host - Meta Disk Kit support Matrix

NW Host	NW Service	External Storage	Meta Disk Kit Applicable (Yes/No)
Series 6/6E	Decoder	MD1400	Yes
	Decoder	MD2412	Yes
	Concentrator	MD1400	Yes
	Concentrator	MD2412	No
Series 7	Decoder	MD2412	Yes
	Concentrator	MD2412	No

Series 6/6E and Series 7 Disk Layout

- **Maximize PowerVault Storage Capacity** - Traditionally, PowerVault storage allocates a volume for the Decoder metadata. This reduces the usable storage on the PowerVault. Drive Packs reduce this issue by providing 20TB of extra usable PV storage.
- **Reduces Cost for Meta Only Use Case** - In Meta Data Only deployments, the use of drive packs can help remove the requirement for the use of a single PowerVault on these Decoders..
- **Enable existing deployments to utilize compression options** - For existing deployments, an SSD index drive pack is required if you need to enable compression. When compressing the packetdb (Decoders) and metadb (Concentrators), additional indexing is needed to support compression of those databases.
- **Provides capability for expanding meta keys and associated indexing** - The index storage needs are scaled based on the NetWitness Platform deployment retention requirements. If additional meta keys are enabled and indexed, it may impact index retention.

Decoder Meta Use Cases

Note: Applicable to Series 6/6E and Series 7 Decoder.

- Meta-Only
- Maximize Power Vault Storage

Three or more 2.4TB 10K SAS SED drives can be added to a Decoder for the decodersmall or logdecodersmall volumes. These volumes are used to store the meta cache on the Decoders.

Both the 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 store a cache for the meta extracted during the data capture for Concentrator aggregation. The meta cache on a Decoder is generally fixed in size, but you can expand it to support additional cache to avoid possible connectivity loss between the Decoder and the corresponding Concentrator.

Typically, the decodersmall or logdecodersmall volumes are stored on the first three drives of the first PowerVault enclosure and, in 10G configurations, second PowerVault enclosure. By utilizing the drive pack option, these three drives can instead be used for the packetdb (maximizing Power Vault storage).

For meta-only scenarios, the decodersmall volume would be stored on the drive pack, therefore eliminating the need for a Power Vault.

Note: The steps noted in the below (Sample Storage Configuration Scenarios for Meta Disk Kit(s) on Series 6/6E Decoder) also applies to Series 7 Meta Disk Kit configuration for both Meta-Only and Maximize PowerVault Storage (MD2412).

Sample Storage Configuration Scenarios for Meta Disk Kit(s) on Series 6/6E Decoder

This section describes how to configure a Meta Disk kit on a Series 6/6E decoder as meta-only and maximized MD1400 PowerVault storage capacity.

Meta-Only (No Externally Attached Storage)

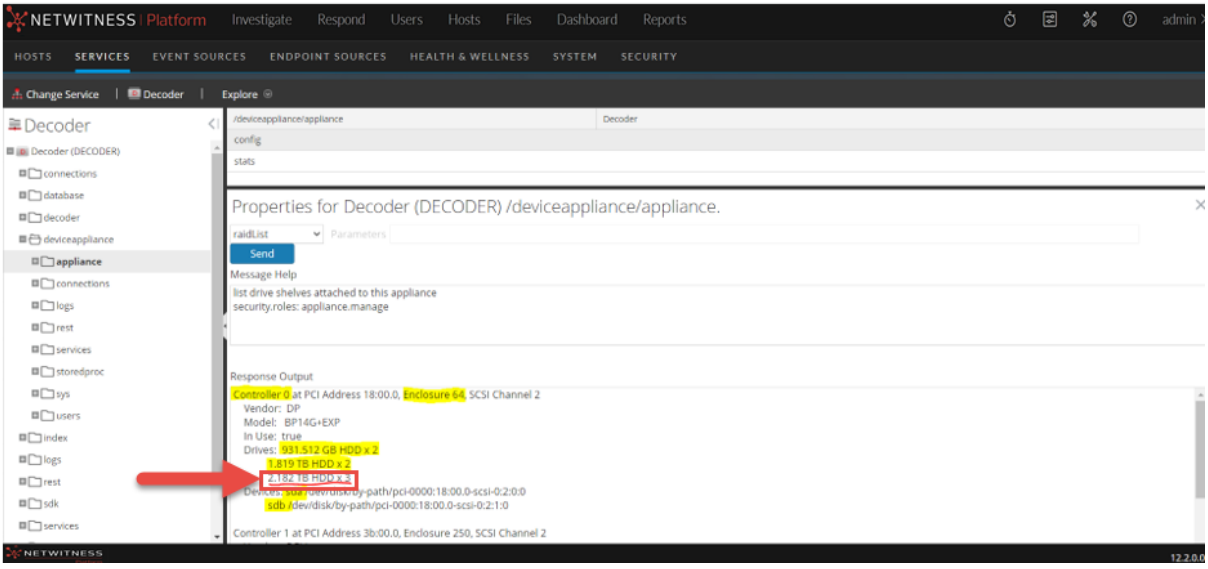
Install and configure one Meta Disk Kit (3 SED Drives configured as RAID 5) on a S6 Core Appliance that has been orchestrated as a Decoder with one attached and unconfigured MD1400 PowerVault.

Note:
1. The configuration for Log Decoder is similar to Network Decoder. Substitute the service and volume names for Log Decoder that would normally be associated with Network Decoder.
2. One Meta Disk pack is configured as RAID5 (3 drives) and two Meta Disk packs (6 drives) are configured as RAID6.
3.When configuring two Meta Disk Packs, the disks are installed in slots 4 through 9 and when adding second Meta Disk Pack, the disks are installed in slots 7 though 9. Refer Series 6/6E and Series 7 Disk Layout for slot details.

On the Series 6 (Dell R640) appliance, the Meta Drive Pack disks are installed in slots 4, 5 and 6. The virtual drive configuration requires identifying the controller ID and Enclosure ID (EID). On Series 6 appliance, the controller ID and Enclosure IDs are 0 and 64, respectively. The nwraidtool.py script, that is installed on every orchestrated server, can help to confirm these ID numbers.

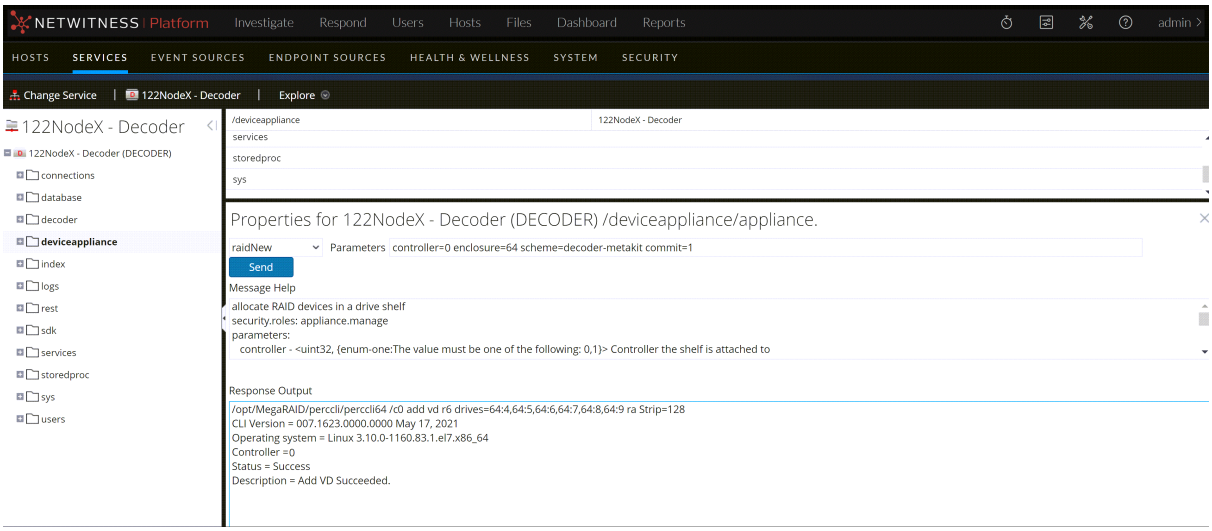
1. Install the three drives in the Drive Pack in slots 4, 5 and 6. Refer Series 6/6E and Series 7 Disk Layout for slot details.
2. Identify the existing block devices using `raidList' property. Login to **NW UI > Hosts > Select the Decoder Host > Actions > View > Explore > deviceAppliance > appliance (Right Click to access properties) > raidList** and click **Send**.

   The existing devices on Controller 0, Enclosure 64 are highlighted in Yellow. The installed Drive Pack in slots 4,5 and 6 is highlighted in Red.

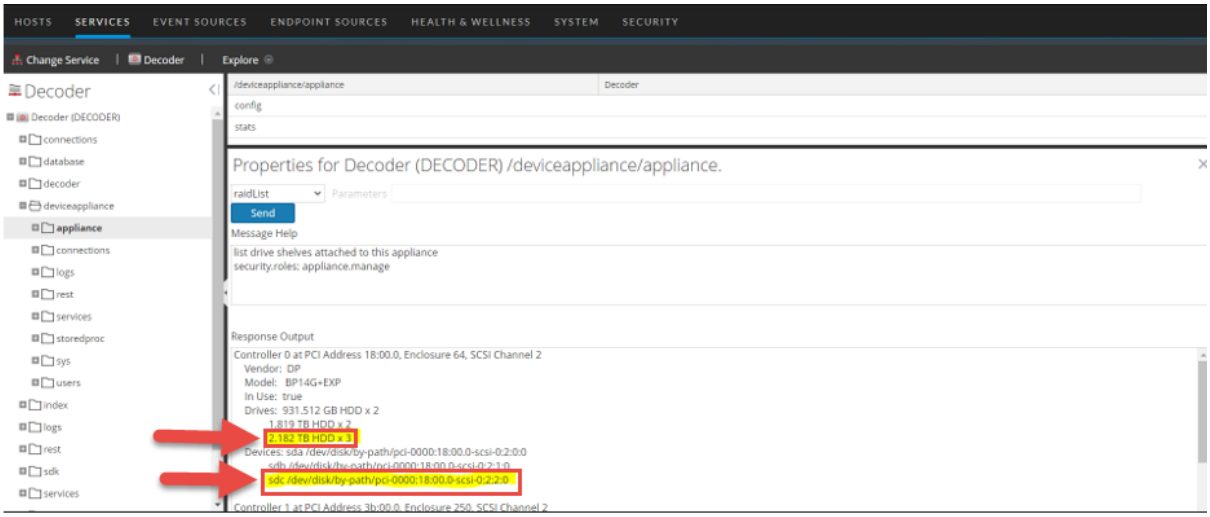


3. Access **NW UI->Hosts->Select the Decoder Host->Actions->View->Explore->deviceAppliance->appliance (Right Click to access properties)->raidNew**, specify the controller, enclosure, and scheme, click **Send**.

Note: For decoder the scheme is decoder-metakit, for logdecoder the scheme is logdecoder-metakit



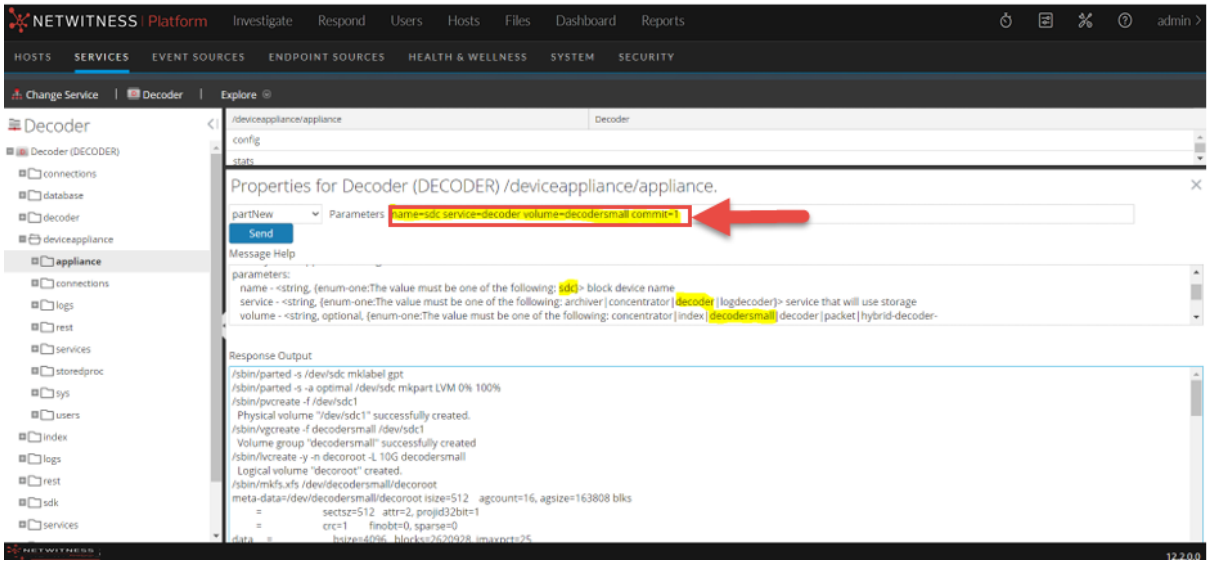
4. Identify the new device (highlighted in Yellow) using raidList command. In this case it is ‘sdc’. Access **NW UI->Hosts->Select the Decoder Host->Actions->View->Explore->deviceAppliance->Appliance (Right Click to access properties)->raidList->Click on Send**.



5. Execute the **partNew** command by selecting it from the Properties drop-down to create the decodersmall partition (decoder dir, index, metadb, sessiondb) with the following parameters.

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

Note: For logdecoder, use the command `name=sdc service=logdecoder volume=logdecodersmall commit=1`

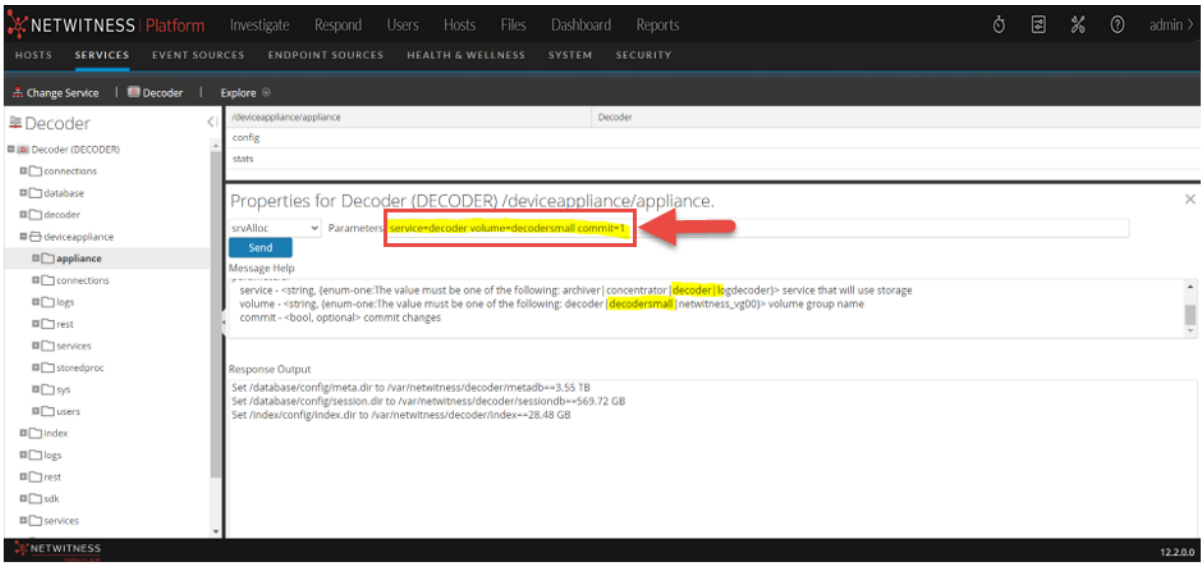



```
[root@s6Core ~]# df -hP
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                             63G     0   63G   0% /dev
tmpfs                                63G  100K   63G   1% /dev/shm
tmpfs                                63G   11M   63G   1% /run
tmpfs                                63G     0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-root     30G  4.1G   26G  14% /
/dev/mapper/netwitness_vg00-nwhome  2.7T  516M  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-varlog   10G   70M   10G   1% /var/log
/dev/mapper/netwitness_vg00-usrhome  10G   33M   10G   1% /home
/dev/sda1                          1014M   91M  924M   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   34M  600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decodersmall-metadb     3.8T   34M  3.8T   1% /var/netwitness/decoder/metadb
[root@s6Core ~]#
```

6. Allocate the decodersmall to Decoder service using ‘srvAlloc’.

   `service=decoder volume=decodersmall commit=1`

> **Note:**
> - For logdecoder, use the command `service=logdecoder volume=logdecodersmall commit=1`
> - If a second Drive Pack is being configured, the volume for decoder would be ‘decodersmall0’. For logdecoder, it is ‘logdecodersmall0’.


NETWITNESS | Platform
Investigate
Respond
Users
Hosts
Files
Dashboard
Reports
admin
HOSTS
SERVICES
EVENT SOURCES
ENDPOINT SOURCES
HEALTH & WELLNESS
SYSTEM
SECURITY
Change Service
Decoder
Explore
Decoder
Decoder (DECODER)
connections
database
decoder
deviceappliance
appliance
connections
logs
rest
services
storedproc
sys
users
index
logs
rest
sdk
services
/deviceappliance/appliance
Decoder
config
stats
Properties for Decoder (DECODER) /deviceappliance/appliance.
srvAlloc
Parameters
service=decoder volume=decodersmall commit=1
Send
Message Help
service - <string, {enum-one:The value must be one of the following: archiver | concentrator | decoder | logdecoder}> service that will use storage
volume - <string, {enum-one:The value must be one of the following: decoder | decodersmall | netwitness_vg00}> volume group name
commit - <bool, optional> commit changes
Response Output
Set /database/config/meta.dir to /var/netwitness/decoder/metadb==3.55 TB
Set /database/config/session.dir to /var/netwitness/decoder/sessiondb==569.72 GB
Set /index/config/index.dir to /var/netwitness/decoder/index==28.48 GB
NETWITNESS
12.2.0.0

Maximize PowerVault Storage Capacity

Installing and configuring a Drive pack on the Decoder appliance as decodersmall frees the first three drives on the attached Power Vault for storing additional packet data.

Best Practice Recommendation: When 1 to 4 PowerVaults are configured, one (1) Meta Disk kit is recommended. When 5 to 8 PowerVaults are configured, two (2) Meta Disk kits are recommended.

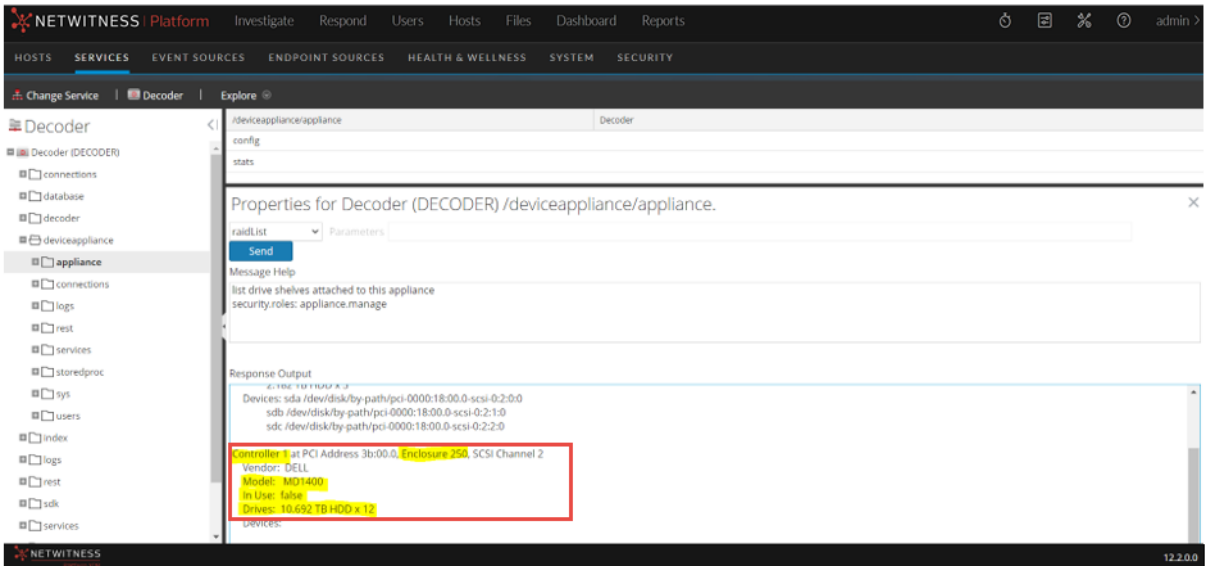
Note: This sample scenario assumes a S6/S6E appliance orchestrated as a Decoder and connected to a PowerVault (but not configured as storage).

1. Follow all the steps in Meta Only: No Externally Attached Storage section to configure one meta disk pack on decoder as the decodersmall partition.
2. Use the **Explorer** view (Login into UI, select **Hosts->Services->Decoder ->Actions->View->Explorer->deviceAppliance->appliance(right click)->properties) -> raidList ->Click on Send**, to identify and confirm the Controller Number, Enclosure Number, In Use, Drives, Devices, Drive Count, Size, and Vendor (highlighted in yellow).

   You should see the following information.

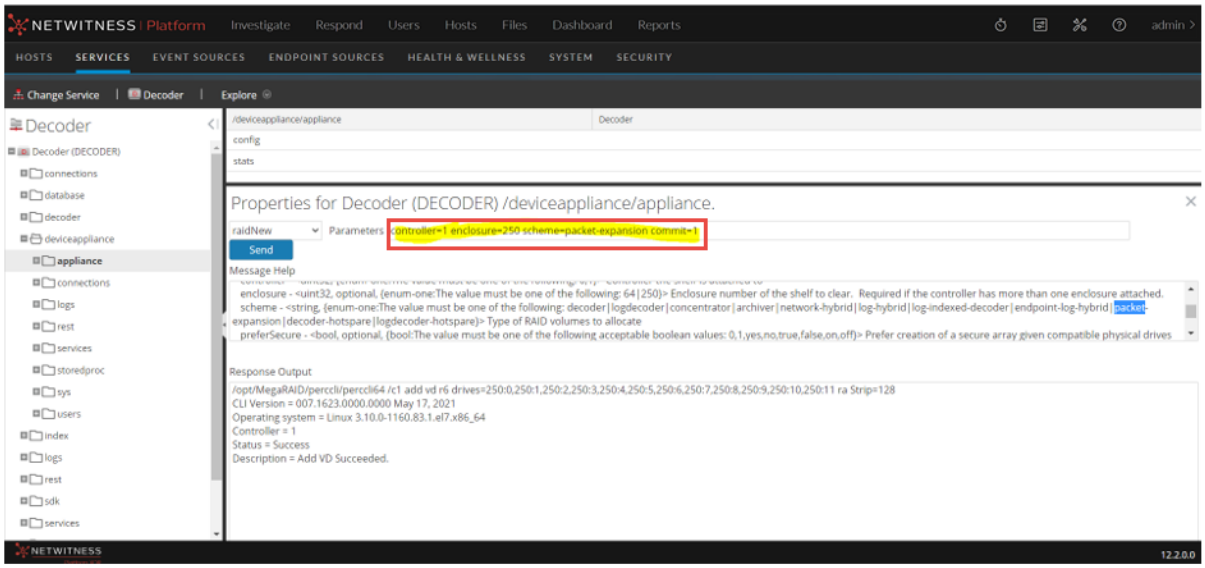
   In Use: FALSE

   Devices: <EMPTY>

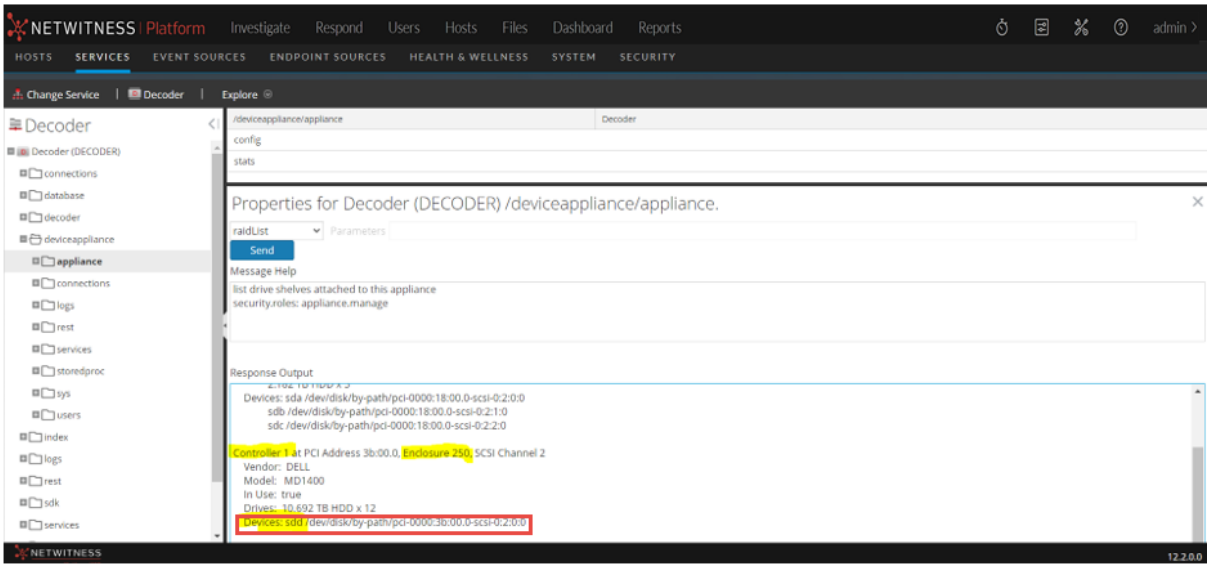


3. Create the raid using ‘raidNew’ from the properties drop-down (passing the below parameters) on the attached PowerVault. Identify the controller and enclosure from ‘raidList’. Encryption can be turned on after configuring storage using the steps listed in Appendix B. Encrypt a Series 6E Core or Hybrid Host.

   ```
   controller=1 enclosure=250 scheme=packet-expansion commit=1
   ```



4. Identify the block device created in the earlier step. Use ‘raidList’ to retrieve all the block devices. In this case, it is ‘sdd’ (highlighted in Yellow).

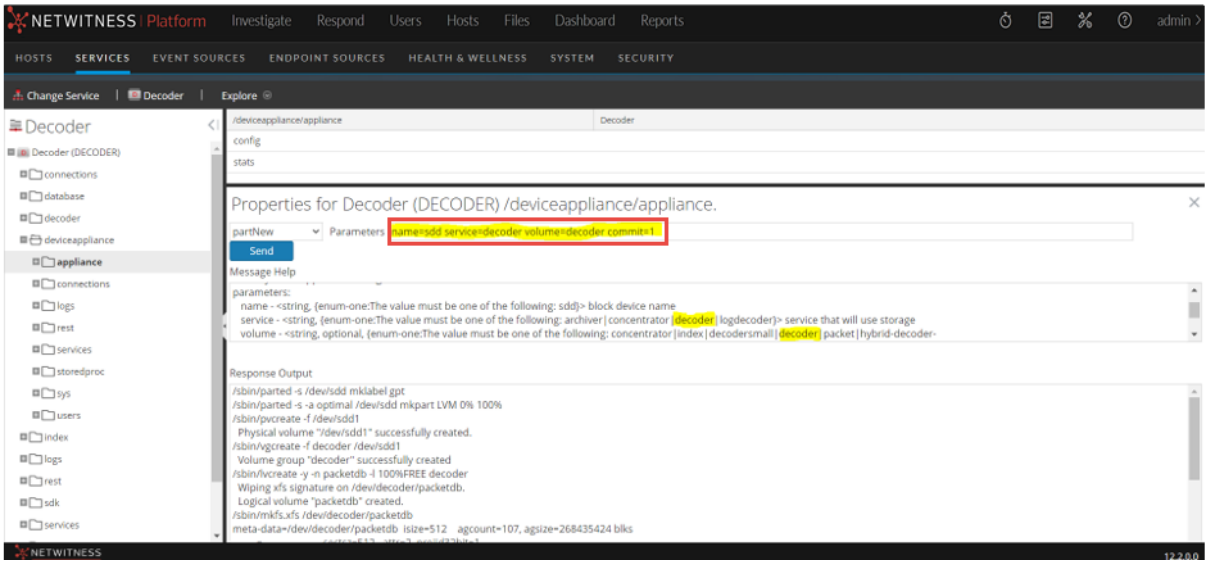


5. Make partitions on the block device (‘sdd’) using ‘partNew’.

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

   For logdecoder, use the following command.

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



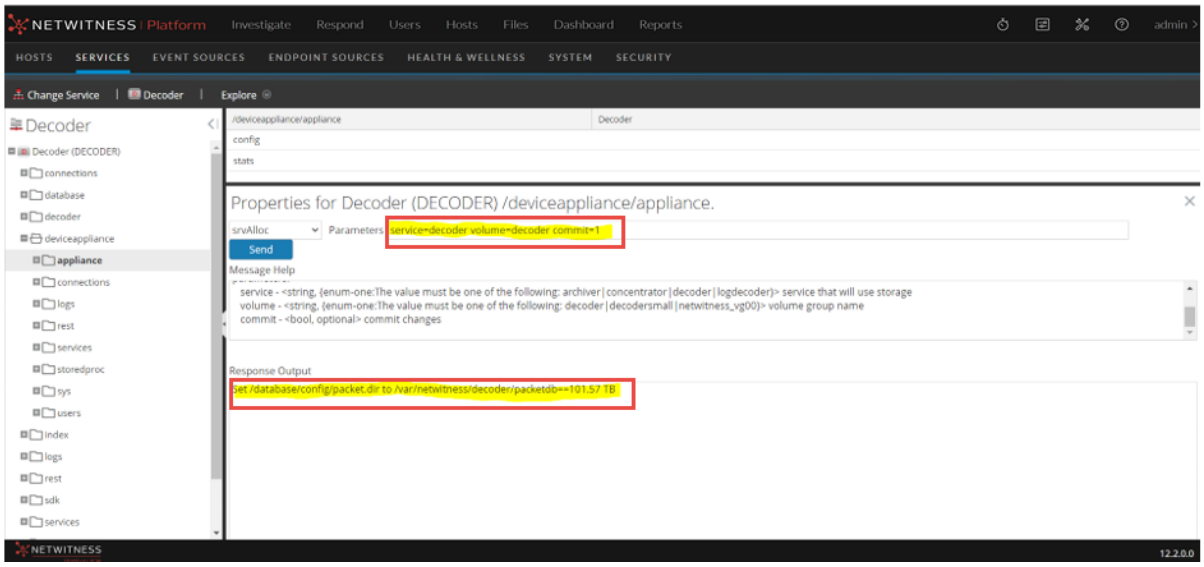

```
[root@s6Core ~]# df -hP
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                             63G     0   63G   0% /dev
tmpfs                                63G   80K   63G   1% /dev/shm
tmpfs                                63G   11M   63G   1% /run
tmpfs                                63G     0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-root     30G  4.1G   26G  14% /
/dev/mapper/netwitness_vg00-nwhome  2.7T  515M  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-varlog   10G   67M   10G   1% /var/log
/dev/mapper/netwitness_vg00-usrhome  10G   33M   10G   1% /home
/dev/sda1                          1014M   91M  924M   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   34M  600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decodersmall-metadb     3.8T   34M  3.8T   1% /var/netwitness/decoder/metadb
/dev/mapper/decoder-packetdb        107T   38M  107T   1% /var/netwitness/decoder/packetdb
[root@s6Core ~]#
```

6. Allocate the decoder volume to Decoder service using ‘srvAlloc’.

   ```
   service=decoder volume=decoder commit=1
   ```

   For logdecoder, use the following command.

   ```
   service=logdecoder volume=logdecoder commit=1
   ```


NETWITNESS | Platform
Investigate
Respond
Users
Hosts
Files
Dashboard
Reports
admin
HOSTS
SERVICES
EVENT SOURCES
ENDPOINT SOURCES
HEALTH & WELLNESS
SYSTEM
SECURITY
Change Service
Decoder
Explore
Decoder
Decoder (DECODER)
connections
database
decoder
deviceappliance
appliance
connections
logs
rest
services
storedproc
sys
users
index
logs
rest
sdk
services
/deviceappliance/appliance
Decoder
config
stats
Properties for Decoder (DECODER) /deviceappliance/appliance.
srvAlloc
Parameters
service=decoder volume=decoder commit=1
Send
Message Help
service - <string, {enum-one:The value must be one of the following: archiver|concentrator|decoder|logdecoder}> service that will use storage
volume - <string, {enum-one:The value must be one of the following: decoder|decodersmall|netwitness_vg00}> volume group name
commit - <bool, optional> commit changes
Response Output
Set /database/config/packet.dir to /var/netwitness/decoder/packetdb==101.57 TB
NETWITNESS
12.2.0.0

Concentrator Index Use Cases

Caution: The Drive Pack(s) for Concentrator Index are supported only on Series 6/6E hosts. These use cases are NOT supported on Series 7 Concentrator hosts. On Series 7 Concentrator hosts, the slots 4 through 10 on the host are reserved for Index database along with MD2412.

The following use cases are supported on Series 6/6E Concentrator Hosts configured with MD1400 Power Vaults.

- Support Additional Meta-Key Indexing
- Capability to Enable compression for Existing Deployments

Three or more 3.84 TB or 3.2 TB or 7.68 TB SSD SED drives can be added to a Concentrator to increase the index volume. The index storage needs are scaled based on the NetWitness Platform deployment retention requirements. If additional meta keys are enabled and indexed, it may impact index retention.

For existing deployments, an SSD index drive pack is required if you need to enable compression. When compressing the packetdb (Decoders) and metadb (Concentrators), additional indexing is needed to support compression of those databases.

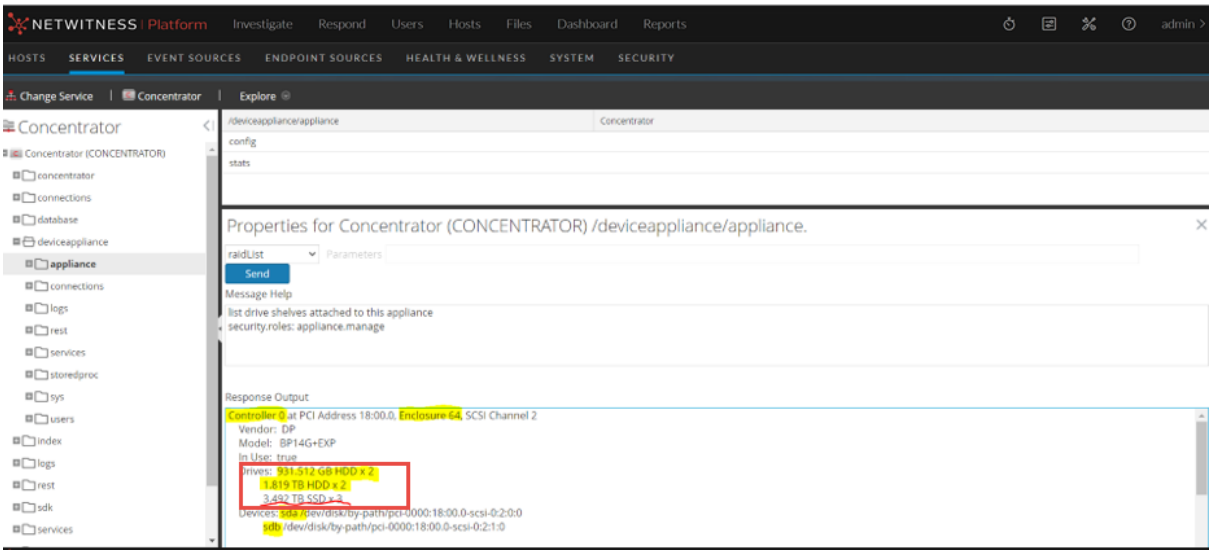
Sample Storage Configuration for Concentrator Index with One Meta Disk Kit (Three SSD's)

This section describes the steps to configure a Meta Disk Kit (Three SSD's) installed on a Series 6/6E appliance orchestrated as Concentrator and **configured with One Power Vault**.

Note:
- Concentrator index usecase supports a maximum of Two Meta Disk Kits. A single Meta Disk pack is configured as RAID5 (3 drives) or two Meta Disk packs (6 drives) are configured as RAID6.
- Even though each Meta Disk Kit consist of 3 SSD's, In rare circumstances when there is no need for very large index expansion and to control costs, users may choose to install only two SSD's (in slots 4 and 5 and configured as RAID1) instead of 3 SSD's (One Meta Disk Kit).
- When configuring two Meta Disk Packs, the disks are installed in slots 4 through 9 and when adding second Meta Disk Pack, the disks are installed in slots 7 though 9. Refer to Series 6/6E and Series 7 Disk Layout for slot details.

On the Series 6 (Dell R640) appliance, the Meta Drive Pack disks are installed in slots 4, 5 and 6. The virtual drive configuration requires identifying the controller ID and Enclosure ID (EID). On Series 6 appliance, the controller ID and Enclosure IDs are 0 and 64. However, the `nwraidtool.py` script that is installed on every server can help to confirm these ID numbers.

1. Install the three SSDs in the Meta Disk kit in slots 4, 5 and 6 on the Concentrator Appliance. Refer to Series 6/6E and Series 7 Disk Layout for slot details.
2. Identify the existing block devices using **raidList** property. Login to **NW UI > Hosts > Select the Concentrator Host > Actions > View > Explore > deviceAppliance > appliance (Right Click to access properties) > raidList** and click **Send**.



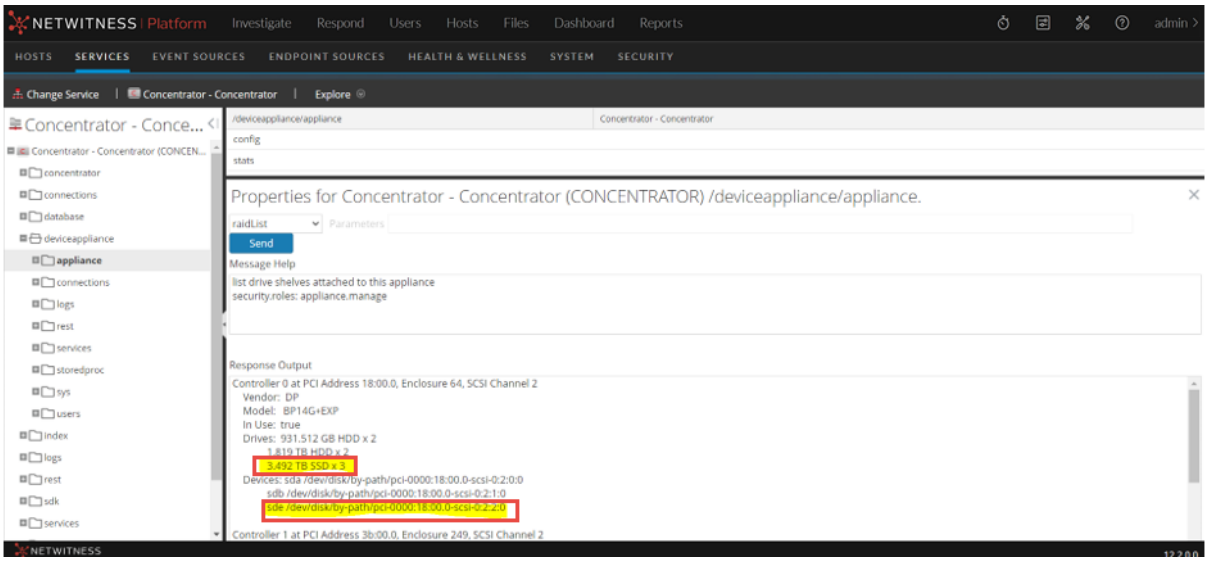
3. SSH to the Concentrator appliance or use iDRAC to connect to the console. Use **perccli** to create the virtual drive with installed Meta Pack kit. Create the Virtual Drive or Drive Group (DG) on the internal controller using the disks in slot 4 through 6 using the below command.

```
/opt/MegaRAID/perccli/perccli64 /c0 add vd type=raid5 drives=64:4-6
strip=128
```

Note:
- To configure two Meta Disk Packs, use the following command.
`/opt/MegaRAID/perccli/perccli64 /c0 add vd type=raid6 drives=64:4-9 strip=128`
- To add a second Meta Disk Pack (First Meta Disk Pack already configured in slots 4,5 and 6), use the following command.
`/opt/MegaRAID/perccli/perccli64 /c0 add vd type=raid5 drives=64:7-9 strip=128`
- To configure only 2 SSD's (instead of 3 SSD's) from a Meta Disk Pack, use the following command. `/opt/MegaRAID/perccli/perccli64 /c0 add vd type=raid1 drives=64:4-5 strip=128`

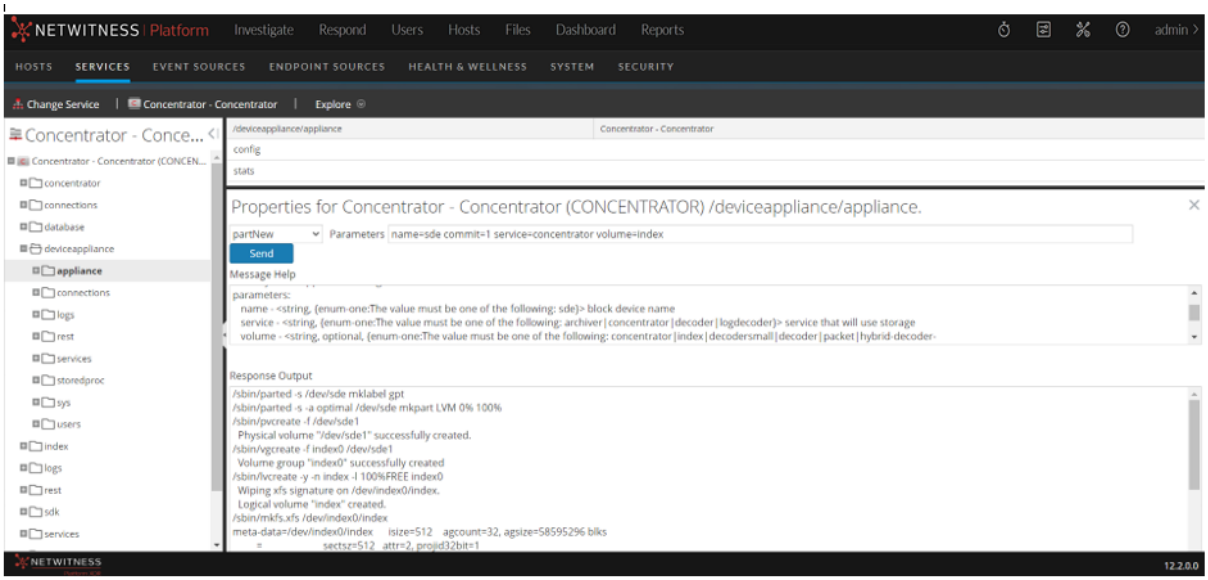
```
[root@S6CoreConc ~]# /opt/MegaRAID/perccli/perccli64 /c0 add vd type=raid5 drives=64:4-6 strip=128
CLI Version = 007.1623.0000.0000 May 17, 2021
Operating system = Linux 3.10.0-1160.83.1.el7.x86_64
Controller = 0
Status = Success
Description = Add VD Succeeded.
```

4. Identify the new device (highlighted in Yellow) using **raidList** command. In this case it is 'sde'. Login to **NW UI->Hosts->Select the Concentrator Host->Actions->View->Explore->deviceAppliance->Appliance (Right Click to access properties)->raidList->Click on Send.**



5. Execute the **partNew** command with the below parameters to create the new index partition on the block device (sde) created in earlier step:

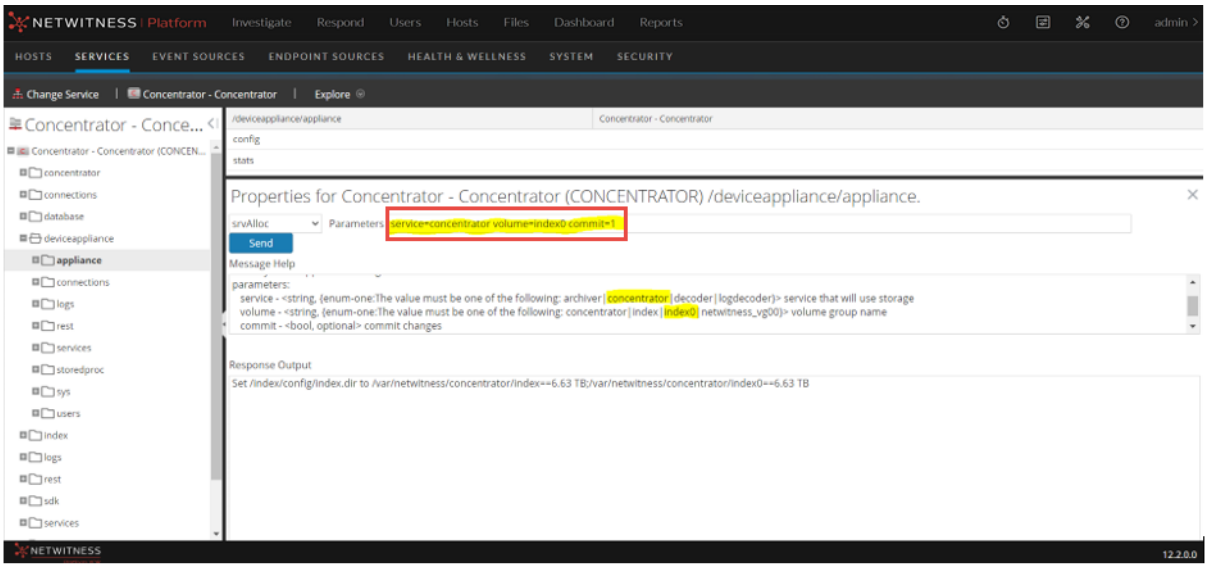
   ```
   name=sde service=concentrator volume=index commit=1
   ```



6. The concentrator appliance is already configured with a Power Vault (an `index` volume is created and configured along with other volumes on the Power Vault) before adding the Meta Disk Kit. The new ‘index’ volume created in the earlier step is named as ‘index0’.

   Allocate the new index volume (index0) using srvAlloc property to concentrator service using the below parameters:

   ```
   service=concentrator volume=index0 commit=1
   ```




```
[root@Concentrator ~]# df -hP
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                             63G     0   63G   0% /dev
tmpfs                                63G   60K   63G   1% /dev/shm
tmpfs                                63G   11M   63G   1% /run
tmpfs                                63G     0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-root     30G  3.7G   27G  13% /
/dev/sda1                          1014M   91M  924M   9% /boot
/dev/mapper/netwitness_vg00-nwhome  2.7T  561M  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-usrhome  10G   33M   10G   1% /home
/dev/mapper/netwitness_vg00-varlog   10G   48M   10G   1% /var/log
tmpfs                                13G     0   13G   0% /run/user/0
/dev/mapper/concentrator-root        30G   34M   30G   1% /var/netwitness/concentrator
/dev/mapper/concentrator-sessiondb  7.5T   34M  7.5T   1% /var/netwitness/concentrator/sessiondb
/dev/mapper/concentrator-metadb      68T   36M   68T   1% /var/netwitness/concentrator/metadb
/dev/mapper/index-index             7.0T   34M  7.0T   1% /var/netwitness/concentrator/index
/dev/mapper/index0-index            7.0T   34M  7.0T   1% /var/netwitness/concentrator/index0
[root@Concentrator ~]#
```

Enable Security on SED Capable Drives

To enable Security on the SED Capable Drive Group on Series 6E (PERC H740 Mini, PERC H750, PERC H840 Adaptors) or Series 7 Hosts (PERC H965i, PERC H965e Adaptors), seeEnable Encryption on Series 6E or Series 7 Hosts.

Prepare Virtual or Cloud Storage

This section describes how to set up virtual or cloud storage for the following types of component hosts:

- Decoder, Log Decoder, Concentrator, Archiver
- NW Server, ESA Primary, ESA Secondary and Malware Analysis
- Log Collector
- Endpoint Log Hybrid
- Additional Endpoint Log Hybrid Partitions
- UEBA

Decoder, Log Decoder, Concentrator, Archiver

Virtual or Cloud NetWitness hosts for Decoders, Log Decoders, Concentrators, and Archivers 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.

NW Server, ESA Primary, ESA Secondary and Malware Analysis

For an extension of `/var/netwitness/` partition, attach an external volume.

Run `lsblk` to get the physical volume name and run the following commands:

1. `pvcreate <pv_name>` (for example, `pv_name` is `/dev/sdc`)
2. `vgextend netwitness_vg00 /dev/sdc`

3. `lvresize --resizefs --extents +100%FREE /dev/netwitness_vg00/nwhome`
4. `xfs_growfs /dev/netwitness_vg00/nwhome`

NetWitness recommends the following partition definitions. However, you can change these values based on the retention days.

LVM	Folder	Block Storage
/dev/netwitness_vg00/nwhome	/var/netwitness/	Refer to the Cloud Provider Block Storage setup (storage) tables.

Log Collector

For an extension of `/var/netwitness/` partition, attach an external volume

Run `lsblk` to get the physical volume name and run the following commands:

1. `pvcreate <pv_name>` (for example, `pv_name` is `dev/sdc`)
2. `vgextend netwitness_vg00 /dev/sdc`
3. `lvresize --resizefs --extents +100%FREE /dev/netwitness_vg00/nwhome`
4. `xfs_growfs /dev/netwitness_vg00/nwhome`

NetWitness recommends the following partition definitions. However, you can change these values based on the retention days.

LVM	Folder	Block Storage
/dev/netwitness_ vg00/nwhome	/var/netwitness/	Refer to the Cloud Provider Block Storage setup (storage) tables.

Endpoint Log Hybrid

The total disk size required depends on the data retention period. You can use the below per day disk usage indicative values to calculate the required disk size for your deployment. For example, to retain 30 days of data, multiply the below per day disk usage values with 30.

The following table provides disk usage for one full scan. The full scan disk usage values are based on the below event count:

- Files count -1100
- Processes count -100
- Dlls count - 500
- Drivers count -150
- Services count - 500
- Tasks count -100

Endpoint Log Hybrid(50K Advance Agents - Disk usage per full scan)					
	MetaDB	PacketDB	SessionDB	Index	Total
Log Decoder	220 GB	12 GB	5 GB	NA	237 GB
Concentrator	230 GB	NA	5 GB	6 GB	241 GB
MongoDB	NA	NA	NA	NA	35 GB (First full scan) 30 GB (Subsequent per scan increase)

The following tables provide per day disk usage for tracking data. The total tracking events per agent per day is 29000.

Endpoint Log Hybrid (50K Advance Agents - Tracking data without Expanded Network Visibility)					
	MetaDB	PacketDB	SessionDB	Index	Total
Log Decoder	1500 GB	140 GB	46 GB	NA	1,686 GB
Concentrator	1600 GB	NA	46 GB	30 GB	1,676 GB
MongoDB	NA	NA	NA	NA	35 GB (First full scan) 1.5 GB (Tracking data per day increase)

The following tables provide per day disk usage for tracking data. Total tracking events per agent per day is 33000

Endpoint Log Hybrid (50K Advance Agents - Tracking data with Expanded Network Visibility)					
	MetaDB	PacketDB	SessionDB	Index	Total
Log Decoder	1800 GB	152 GB	55 GB	NA	2007 GB
Concentrator	1900 GB	NA	55 GB	36 GB	1991 GB
MongoDB	NA	NA	NA	NA	35 GB (First full scan) 1.5 GB (Tracking data per day increase)

The following table provides per day disk usage for insight agents. The total tracking data per agent per day is 10800 plus 1 full scan daily.

Endpoint Log Hybrid (50K Insights Agents with Expanded Network Visibility)					
	MetaDB	PacketDB	SessionDB	Index	Total
Log Decoder	500 GB	52 GB	18 GB	NA	570 GB

Endpoint Log Hybrid (50K Insights Agents with Expanded Network Visibility)					
Concentrator	600 GB	NA	18 GB	13 GB	631 GB
MongoDB	NA	NA	NA	NA	35 GB (First full scan) 30 GB (Subsequent per scan increase)

The following table provides Endpoint Agents sizing based on the feature.

Feature	Description	Agent or Endpoint Server
Endpoint Only	Only scan and tracking data	Maximum 50K Endpoint Agents only
Windows Logs Only	Only Windows Logs from agents. Assuming 20K events per second supported by Hybrid.	Maximum 20K Agents: • Generates 20K log events per second
File Collection Only	Only File Collection from agents. Assuming 20K events per second supported by Hybrid	Maximum 20K Agents : • Generates 20K log events per second
Endpoint and Windows Logs	Event per second per agent • (For Windows Logs) 1 event sent by 1 agent every second • (For Tracking Events) 0.4 event sent by 1 agent every second • 20K events per second supported by Hybrid **Note:** Total agents should be calculated as below: Hybrid events per second/ (Windows Logs Endpoint Server of 1 agent + Tracking Event Endpoint Server for 1 agent) For example, 20000 / (1.0 + 0.4)	Maximum 15K (approximately) Agents: • Generates 15K (approximately) Windows log events Plus • Generates 15K (approximately) Agents EDR data

Feature	Description	Agent or Endpoint Server
Endpoint, Windows Logs and File Collection	Event per second per agent: • (For Windows Logs) 1 event sent by 1 agent every second • (For Tracking Events) 0.4 event sent by 1 agent every second • (For File Collection) 1 event sent by 1 agent every second • 20, 000 events per second supported by Hybrid **Note:** Total agents should be calculated as below: Hybrid events per second/ (Windows Logs Endpoint Server of 1 agent + Tracking Event Endpoint Server for 1 agent + File Collection) For example, 20000 / (1.0 + 1.0 + 0.4)	Maximum 10K (approximately) Agents: • Generates 10K (approximately) Windows log events Plus • Generates 10K (approximately) Endpoint Agents data Plus • Generates 10K (approximately) Agents File Collection data

Extending File Systems

For Endpoint Server, attach external disk for extension of `/var/netwitness/` partition, create an external disk with suffix as `nwhome`.

Follow these steps:

1. Ensure you have added a new disk. For more information, see "Task 1. Add New Disk" in the *Virtual Hosts Installation Guide for NetWitness Platform*. Go to the NetWitness All Versions Documents page and find NetWitness Platform guides to troubleshoot issues.
2. Execute `lsblk` and get the physical volume name
3. `pvcreate <pv_name>` suppose the PV name is `/dev/sdc`
4. `vgextend netwitness_vg00 /dev/sdc`
5. `lvresize --resizefs --extents +100%FREE /dev/netwitness_vg00/nwhome`
6. `xfs_growfs /dev/mapper/netwitness_vg00-nwhome`

NetWitness recommended partition for Endpoint Server (can be changed based on the retention days).

LVM	Folder	Size	Disk Type
`/dev/netwitness_vg00/nwhome`	`/var/netwitness/`	6TB	HDD

For Mongo DB, attach external disk for extension of `/var/netwitness/mongo` partition, create an external disk with suffix as `nwhome`.

Follow these steps:

1. Ensure you have added a new disk. For more information, see "Task 1. Add New Disk" in the *Virtual Hosts Installation Guide*. Go to the NetWitness All Versions Documents page and find NetWitness Platform guides to troubleshoot issues.
2. Execute `lsblk` and get the physical volume name
3. `pvcreate <pv_name>` suppose the PV name is `/dev/sdc1`
4. `vgextend hybrid /dev/sdc1`
5. `lvresize --resizefs --extents +100%FREE /dev/hybrid-vlmng`
6. `xfs_growfs /dev/mapper/hybrid-vlmng`

NetWitness recommended partition for Mongo DB (Can be changed based on the retention days). Minimum recommended size for var/netwitness is 500 GB.

LVM	Folder	Size	Disk Type
`/dev/hybrid-vlmng`	`/var/netwitness/mongo`	6TB	HDD

Additional Endpoint Log Hybrid Partitions

The following partition should be on the volume group endpoint and should be in a single RAID 0 array.

Folder	LVM	Volume Group
/var/netwitness/mongo	hybrid-mongo	endpoint
/var/netwitness/concentrator	concentrator-concroot	endpoint
/var/netwitness/concentrator/index	hybrid-concinde	endpoint
/var/netwitness/logdecoder	hybrid-ldecroot	endpoint

Run `lsblk` to get the physical volume name and run the following commands:

1. `pvcreate /dev/md0`
2. `vgcreate -s 32 endpoint /dev/md0`
3. `lvcreate -L <disk_size> -n <lvm_name> endpoint`
4. `mkfs.xfs /dev/ endpoint /<lvm_name>`
5. Repeat the above steps for all the LVMs mentioned.

NetWitness recommends the following partitions. However, you can change these values based on the retention days.

LVM	Folder	Block Storage
/dev/netwitness_vg00/nwhome	/var/netwitness/	Refer to the Cloud Provider Block Storage setup (storage) tables.

LVM	Folder	Block Storage
/dev/endpoint/hybridmongo	/var/netwitness/mongo	Refer to the Cloud Provider Block Storage setup (storage) tables.
/dev/endpoint/concentratorconcroot	/var/netwitness/concentrator	Refer to the Cloud Provider Block Storage setup (storage) tables.
/dev/endpoint/hybridconcinde	/var/netwitness/concentrator/index	Refer to the Cloud Provider Block Storage setup (storage) tables.
/dev/endpoint/hybridldecroot	/var/netwitness/logdecoder	Refer to the Cloud Provider Block Storage setup (storage) tables.

UEBA

The following procedure attaches an external disk and extends the `/var/netwitness/` partition. You must use `nwhome` as the eternal disk suffix. This procedure illustrates how to add a 2TB disk.

Note: `/var/netwitness` is the only partition that can reside on this volume.

1. List the physical volume name.
   `lsblk` (for example, `dev/mapper/sdc`)
2. Extend the `/var/netwitness/` partition.
   ```
   pvcreate <pv_name>where pv_name is dev/mapper/sdc
   vgextend netwitness_vg00 /dev/mapper/sdc
   lvresize --resizefs --extents +100%FREE /dev/mapper/netwitness_vg00/nwhome
   xfs_growfs /dev/mapper/netwitness_vg00-nwhome
   ```

This partition is the NetWitness recommended partition for UEBA. You can change it based on retention days.

Configure Storage Using the REST API

you can 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 NetWitness All Versions Documents page and find NetWitness Platform guides to troubleshoot issues.

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 for 15-Drive DACs.

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.
- `multipath-II` : To verify if SAN devices are attached.

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 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 MD1400) Setup Guide and PowerVault (Dell MD2412) 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.

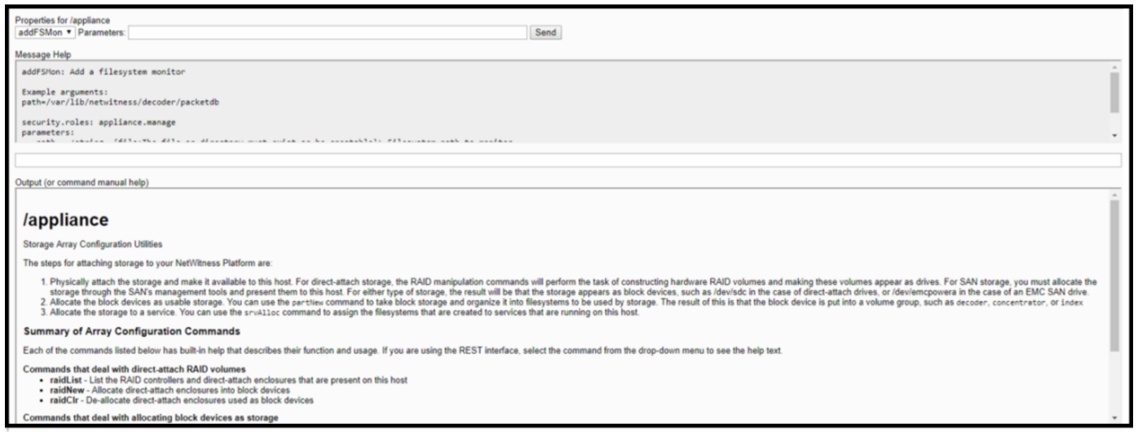
   Note: Once you have accessed the REST API, the steps that you perform are the same, no matter which method you used to access it.

- From a Browser.

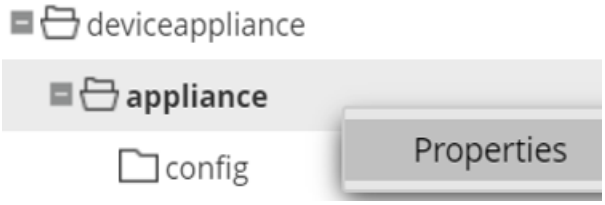
  a. Open a Browser and specify the ip-address of the host with port **50106**.
  The following example is the Decoder, but you need to use port 50106 for any host hardware for which you are configuring storage using the REST API.
  `https://<decoder-ip-address>:50106`

  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. In the **NetWitness** menu, 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:

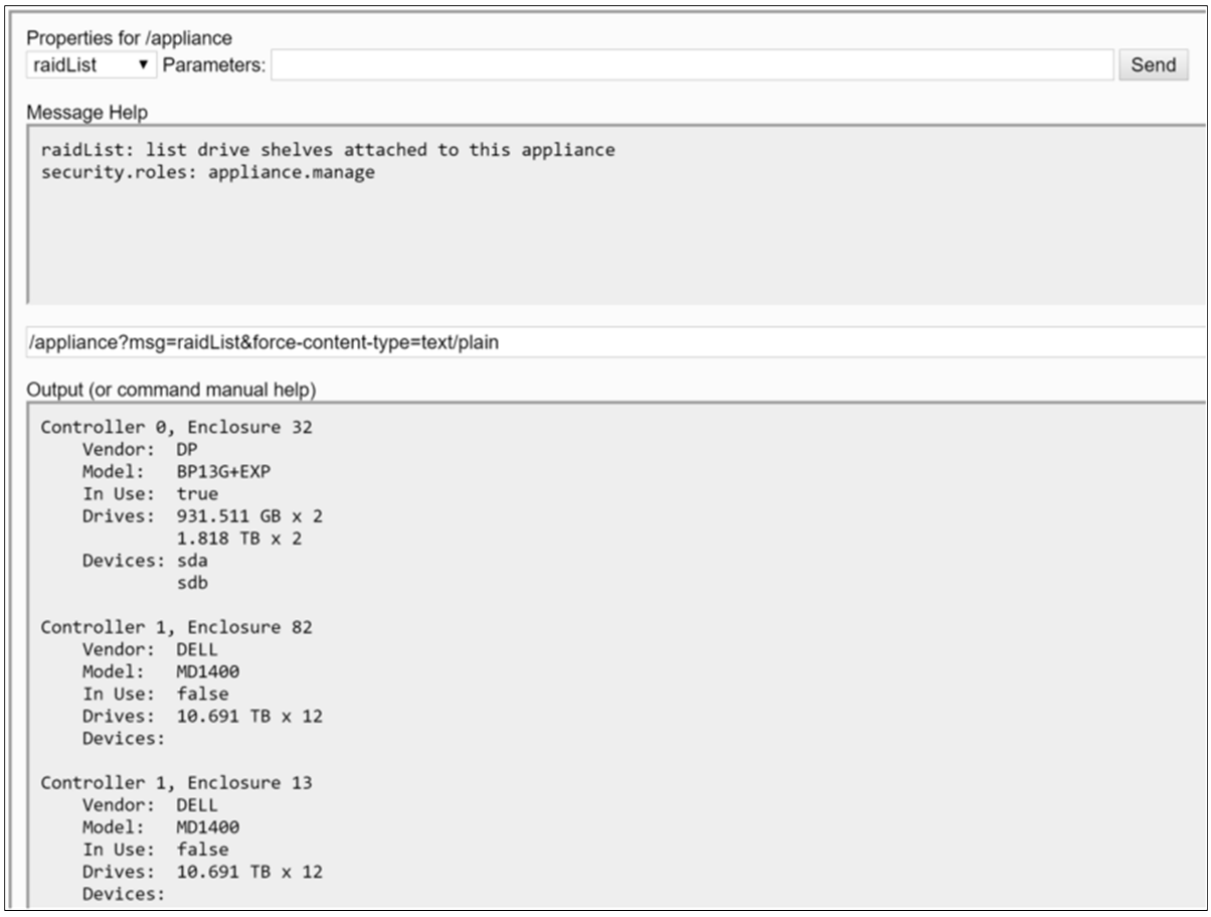
- EMC ESAS 15-drive enclosures
- EMC ESAS 60-drive enclosures
- Dell PowerVault 12-drive enclosures
- Dell PowerVault 8-drive enclosures

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 enclosues. 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.



2. Select a RAID layout scheme for the Enclosure.

The following tables list the PV to Supported Hosts Mapping.

Type	SKU	Specification	Supported Hosts
High Density	NW-PV-A	Dell Storage MD1400 12 x 12 TB NL-SAS SED	Decoder, LogDecoder, Archiver, Log Hybrid, Network Hybrid, Endpoint Log Hybrid
High Density	NW-PV-B	Dell Storage MD 1400 8 x 12TB NL-SAS SED	Decoder, LogDecoder, Archiver, Log Hybrid, Network Hybrid, Endpoint Log Hybrid
High Performance	NW-PV-C	Dell Storage MD 1400 6 x 12TB NL-SAS SED, 2 x 3.8TB SSD SED	Concentrator

Type	SKU	Specification	Supported Hosts
High Performance	NW-PV-D	Dell Storage MD 1400 9 x 12TB NL-SAS SED, 3 x 3.8TB SSD SED	Concentrator

Type	SKU	Specification	Supported Hosts
High Density	192TB (NW-PV-A-N)	Dell Storage MD1400 12 x 16 TB NL-SAS SED	Decoder, LogDecoder, Archiver, Log Hybrid, Network Hybrid, Endpoint Log Hybrid
High Density	128TV (NW-PV-B-N)	Dell Storage MD 1400 8 x 16TB NL-SAS SED	Decoder, LogDecoder, Archiver, Log Hybrid, Network Hybrid, Endpoint Log Hybrid
High Performance	103TB (NW-PV-C-N)	Dell Storage MD 1400 6 x 16TB NL-SAS SED, 2 x 3.8TB SSD SED	Concentrator
High Performance	155TB (NW-PV-D-N)	Dell Storage MD 1400 9 x 16TB NL-SAS SED, 3 x 3.8TB SSD SED	Concentrator
High Density	S7-PV-HD-192	Dell Storage MD2412 12 x 16 TB SAS SED	Decoder, LogDecoder, Archiver, Log Hybrid, Network Hybrid, Endpoint Log Hybrid
High Density	S7-PV-HD-192 AND S7-PE-CON-SDF-3 OR S7-PV-HD-192 AND S7-PE-CON-SDF-7	Dell Storage MD2412 12 x 16 TB SAS SED One or more SSD Drive Pack (s)	Concentrator

The following tables show you the supported allocation schemes.

Note:
- On a Series 6 or Series 7 Network Decoder or newer with multiple PowerVault storage trays, use the decoder-hotspare RAID scheme for the first enclosure and the packet-expansion RAID scheme for subsequent enclosures.

Scheme	Drives Required	Allocation
decoder-hotspare	8 or 12 or 15 HDDs	2x Drives in RAID 1 for decoder small, 1 drive as hotspare, all remaining drives in RAID 5 for decoder

Scheme	Drives Required	Allocation
logdecoder-hotspare	8 or 12 or 15 HDDs	Same as decoder-hotspare configuration
archiver	8 or 12 or 15 HDDs	All drives in RAID 6 for archiver or decoder database volume
network-hybrid	8 or 12 or 15 HDDs	3x drives in RAID 5 for meta expansion, all remaining drives in RAID 5 for packet expansion
log-hybrid **Note:** log-hybrid scheme is **also** used to configure a **PowerVault** for Endpoint Log Hybrid host.	8 or 12 or 15 HDDs	Half of the drives in RAID 5 for meta expansion, half the drives in RAID 5 for packet expansion
concentrator	2 or more SSDs, 4 or more HDDs	All SSDs in RAID 1 or RAID 5 for index, all HDDs in RAID 6 for meta
packet-expansion	8 or 12 or 15 HDDs	All drives in RAID 6 for decoder volume, no drives allocated for decodersmall
decoder-metakit	1 metakit (3 HDDs) or 2 metakits (6 HDDs)	3x drives in RAID 5 or 6x drives in RAID 6 for meta
logdecoder-metakit	1 metakit (3 HDDs) or 2 metakits (6 HDDs)	3x drives in RAID 5 or 6x drives in RAID 6 for meta
concentrator-metakit	1 metakit (3 SDDs) or 2 metakits (6 SDDs)	3x drives in RAID 5 or 6x drives in RAID 6 for index. If two drive configuration, then 2x drives in RAID 1 for index.
decoder or logdecoder **Note:** The decoder and logdecoder scheme has been deprecated in favour of decoder-hotspare and logdecoder-hotspare.	8 or 12 or 15 HDDs	3x drives in RAID 5 for decodersmall or logdecodersmall, all remaining drives in RAID 5

3. 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-hotspare preferSecure=false`
   Add the `commit=1` parameter to actually execute this operation. Execute the `raidList` command to

list the created block devices.

4. (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`.
5. 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.

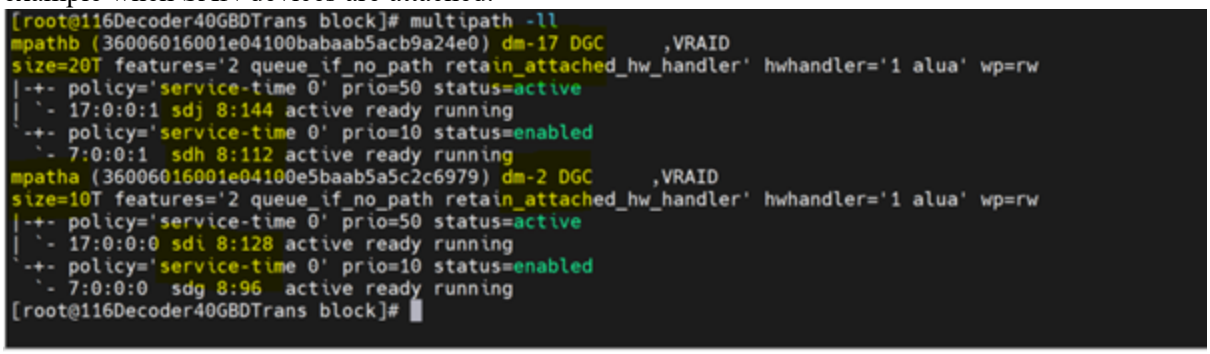
Output (or command manual help)

```
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
```

   Also,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.

Note: Run the `devlist` command to see if the multipath user-friendly names are listed correctly.

2. Run the `multipath_II` command to make sure that SAN devices are attached. The following is an example when SAN devices are attached.



Note: Block devices should be configured with a user-friendly name such as mpatha, mpathb etc.

3. 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.
```

4. 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.
5. 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=1` parameter to the command string to actually make the changes to the system and restart the specified service after making changes.

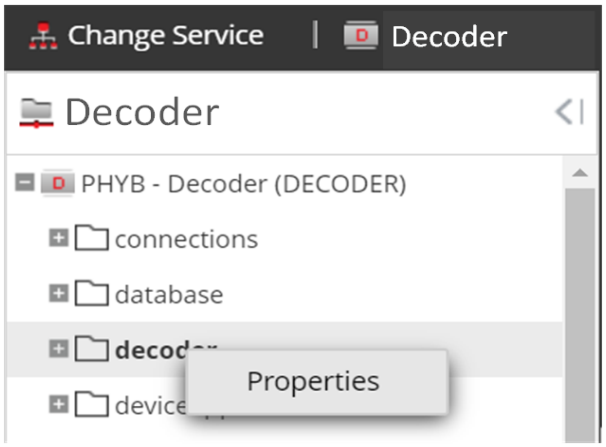
1. Execute the `srvList` command to see a list of services installed on this host.
   The `srvList` command communicates with the service through the SSL port. You install a Category on a host. A Category can be a single service, or multiple related services, located on the same host.
2. Execute the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

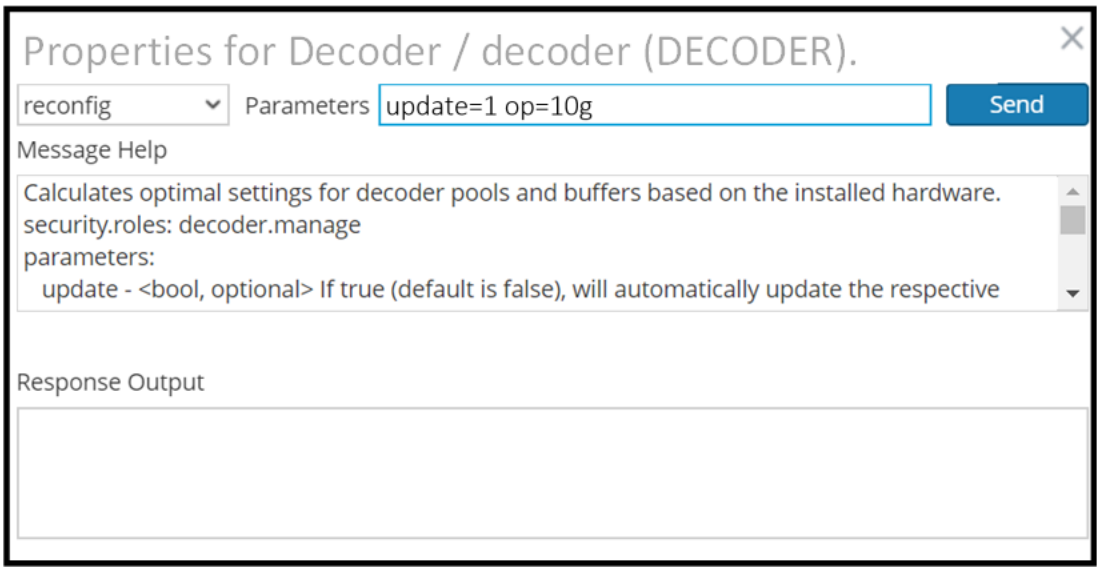
1. In the **NetWitness** menu, go to  (Admin) > **SERVICES**.
   The **SERVICES** view is displayed.
2. Select the **decoder**.
3. Under  (actions), select **View** > **Explore**.
   The **Explore** tree for the service is displayed.

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

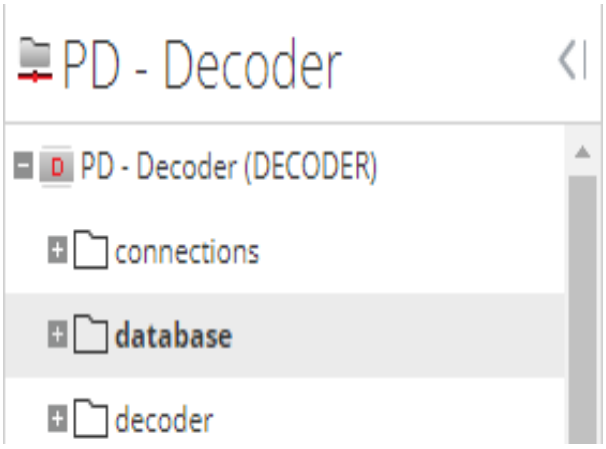


   The **Properties** dialog is displayed.

   b. Execute the `reconfig` command by selecting it from the drop-down list, specifiy `update=1 op=10g` in **Parameters**, and click **Send**.

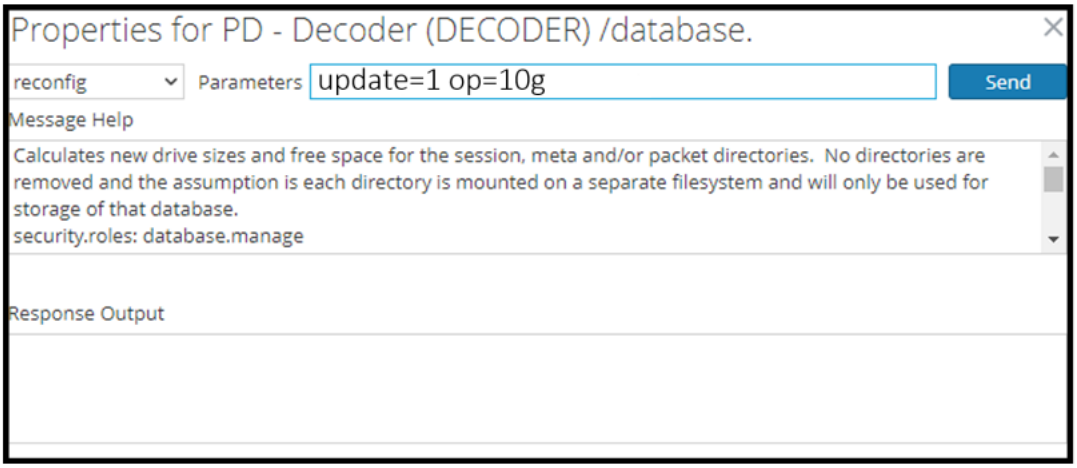


5. Reconfigure space on the database.
   a. Navigate to **database** in the service **Explore** tree, right click, and click **Properties**.



   The **Properties** dialog is displayed.

   b. Execute the `reconfig` command by selecting it from the drop-down list, specifiy `update=1 op=10g` 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 NetWitness Platform and ensure the allocated storage meets all of the 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 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

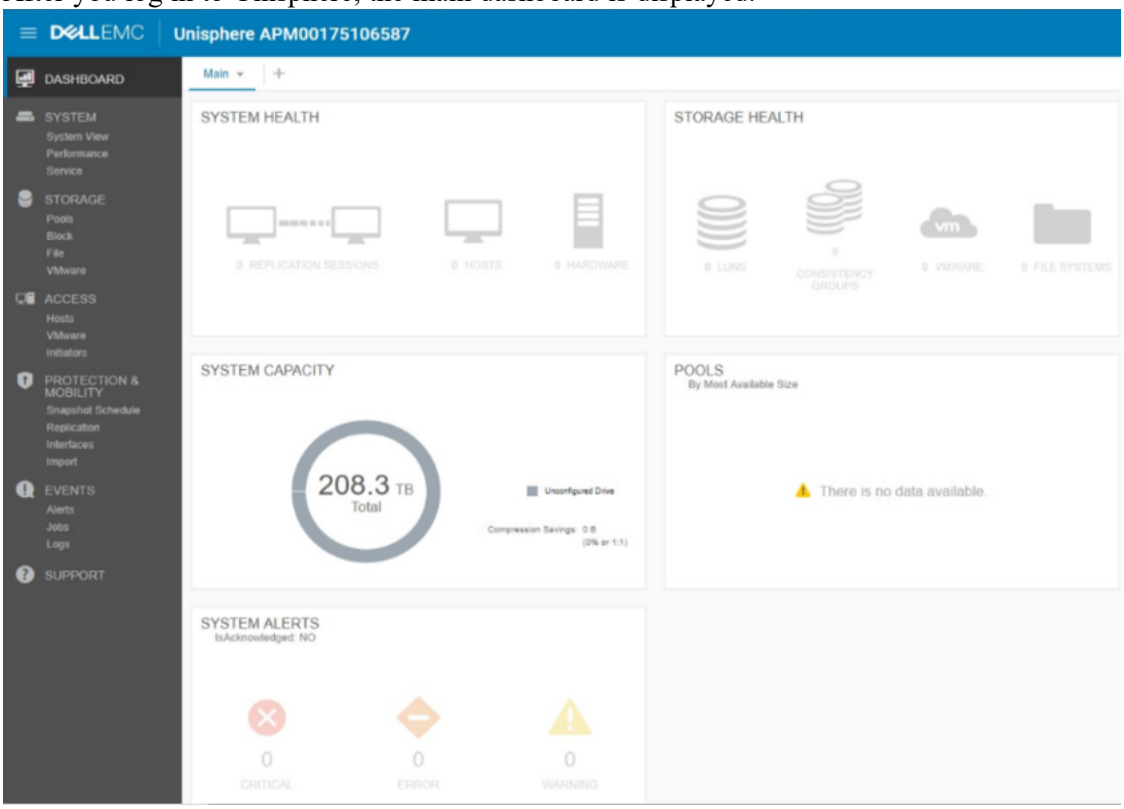
Task 6 - Install PowerPath

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).

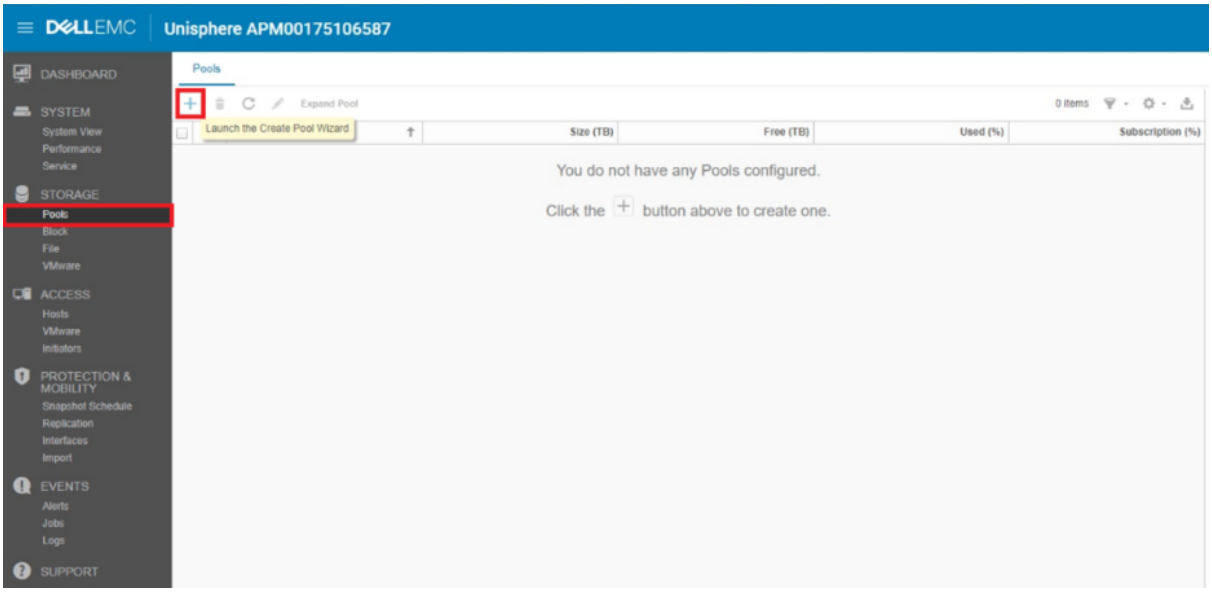
After you log in to Unisphere, the main dashboard is displayed.



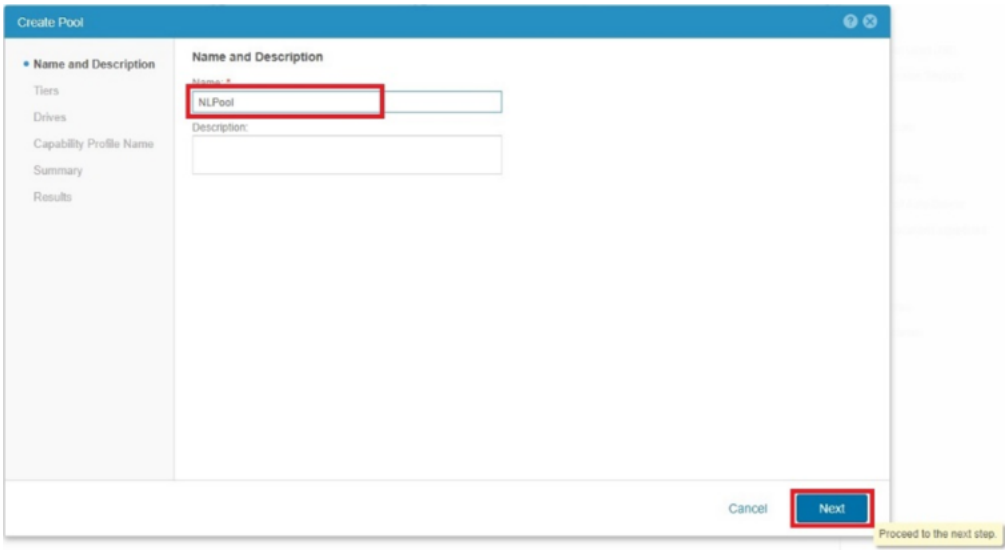
Task 2 - Create Pools

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.

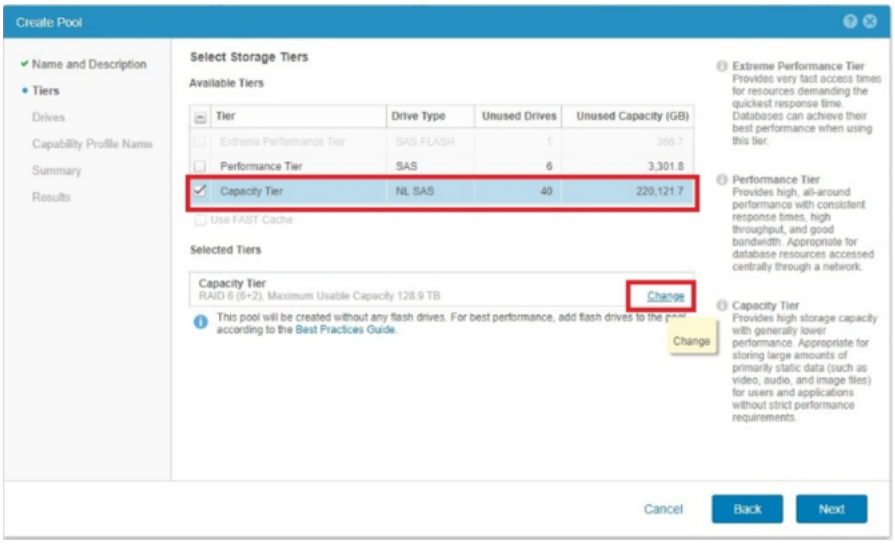
1. From **Storage Section**,click > **Pools** > (Add) to launch the Create Pool Wizard.



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

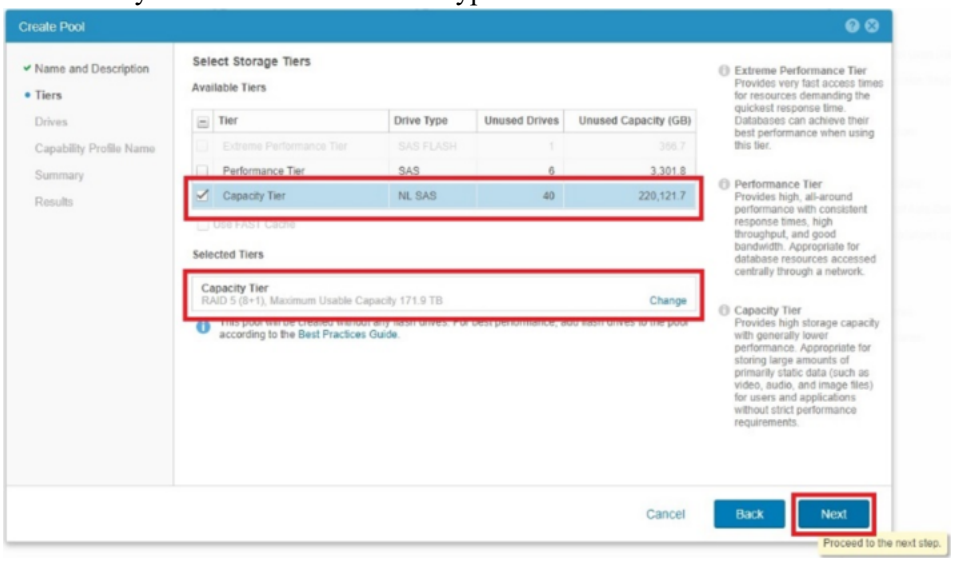


3. Select **Capacity Tier** under **Tier** for the tier type (drive type) and click **Change**.

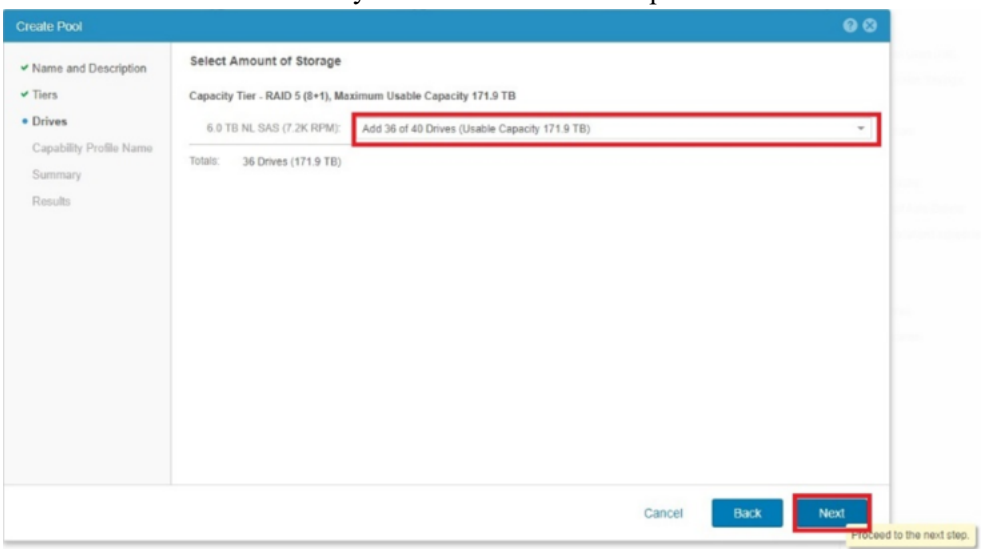


4. Choose the RAID type and from the drop down and select the RAID size.
   The RAID type and size are a customer preference. The only requirement is to make sure you have enough IOPS within the pool to accommodate the log or packet capture and queries. In the following example, a **RAID 5 (8+1)** configuration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

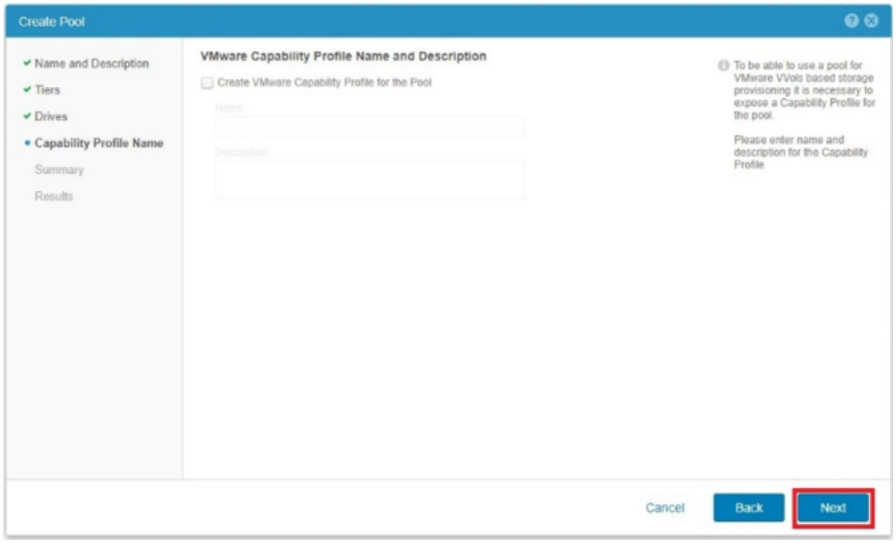
5. Make sure you have the correct Raid type and size selected.



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

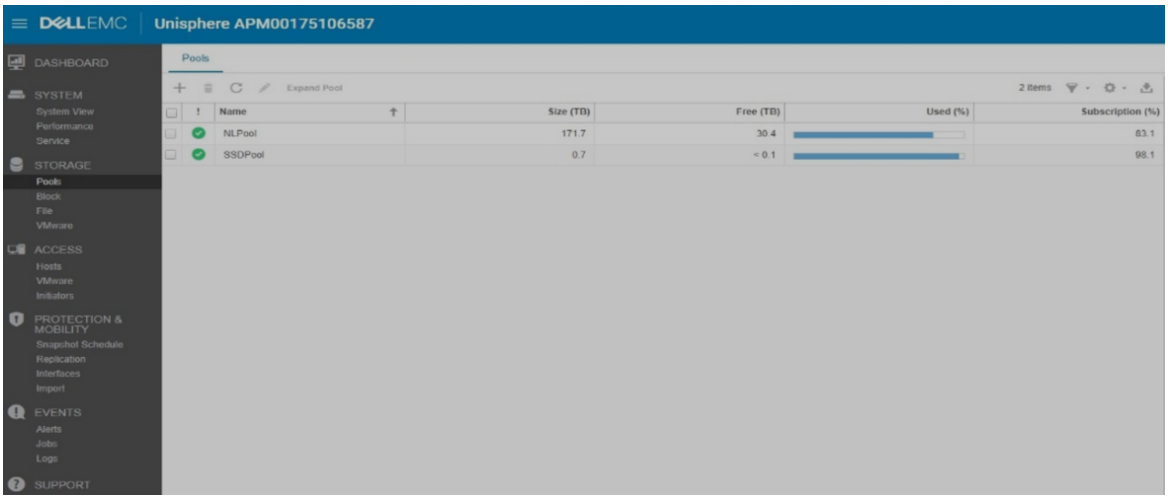


7. Skip the **VMware Capability** section and click **Next**.



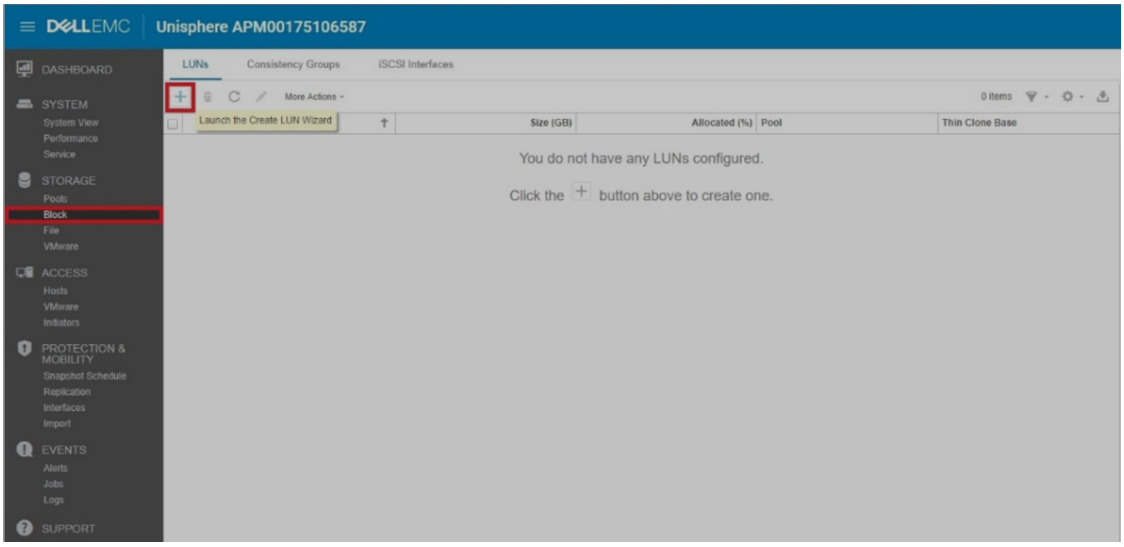
8. Make sure that everything is correct on the Summary tab, and click **Finish**.
9. Create another pool for the SSDs using steps 2 – 8.
   a. Enter in a name for theother pool (for example, **SDDPool**) and click **Next**. Optionally, you can also enter a description for the pool.
   b. Select **Extreme Performance Tier** under **Tier** for the tier type (drive type) and click **Change**.
   c. 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

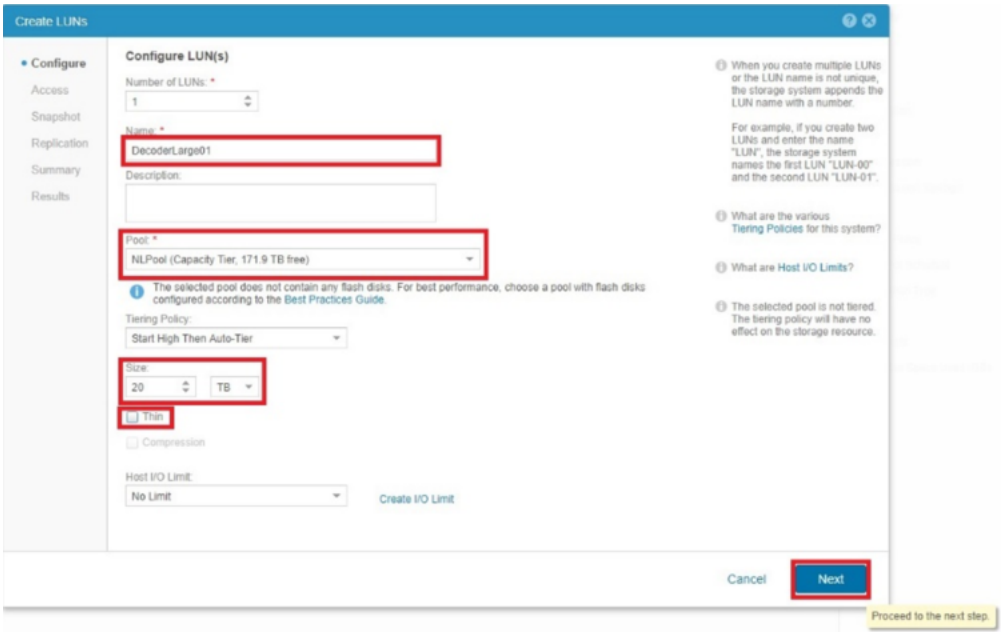
1. From **Storage** section, click **Block** > (Add) to launch the **Create LUN Wizard**.



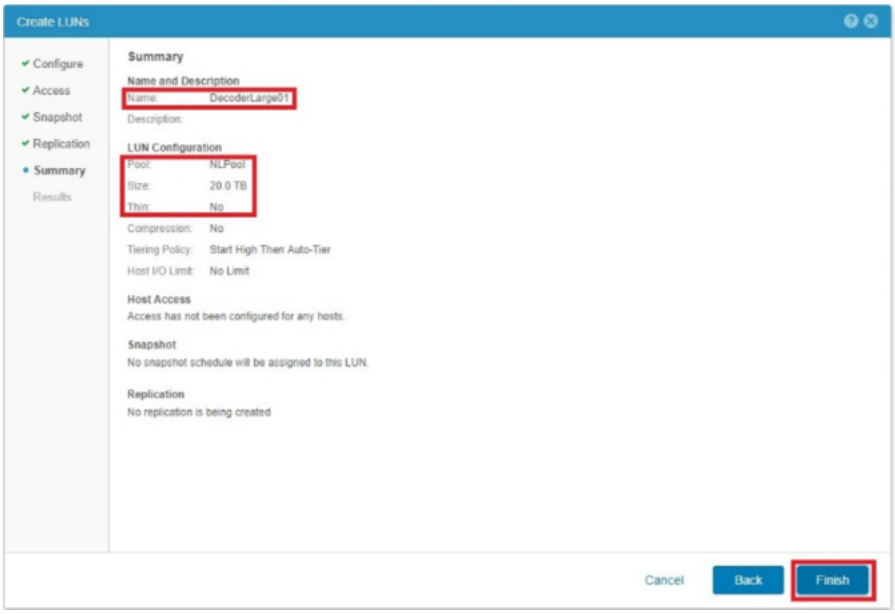
The table below list all of the possible LUNS you may need to create. The ConIndex is the only LUN you need to assign to the SSD Pool. Make sure that the LUN sizes do not exceed what is listed below.

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.



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.

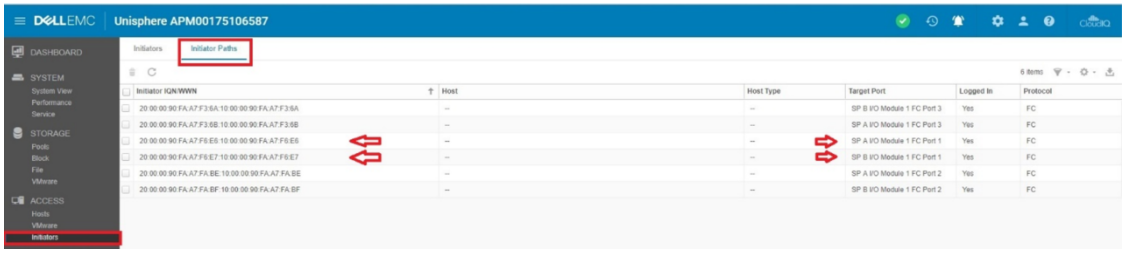


9. Repeat steps 2- 8 for the remaining LUN creations.

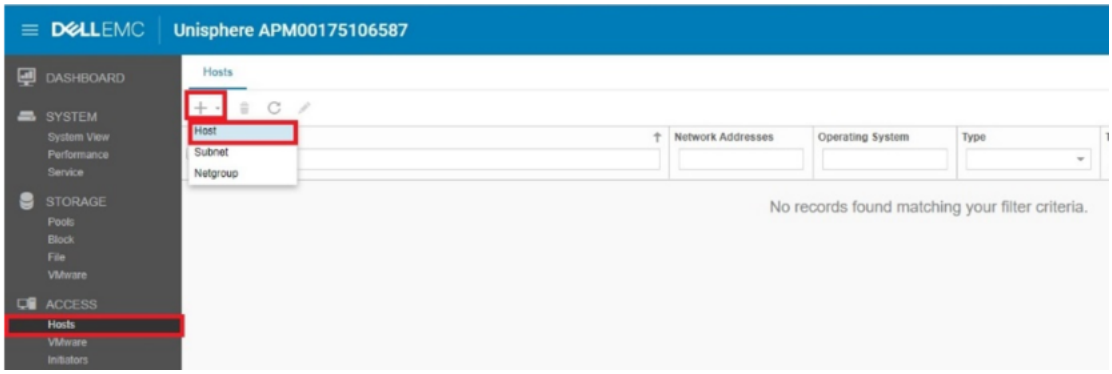
Task 4 - Register Hosts

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.

1. From the Access section, click **Initiators**.
2. 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.

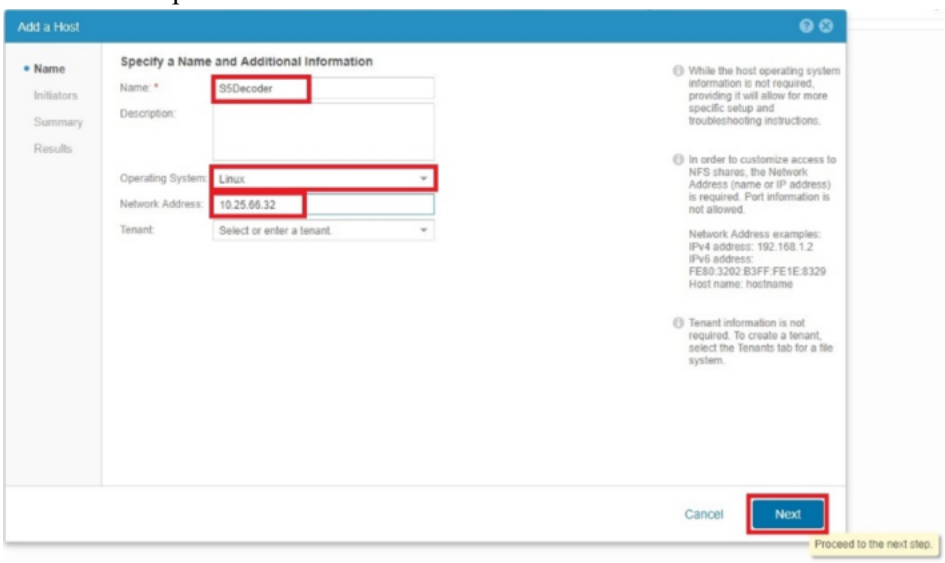


3. From the **Access** section, click **Hosts** > (Add) > **Host** to add a host configuration.

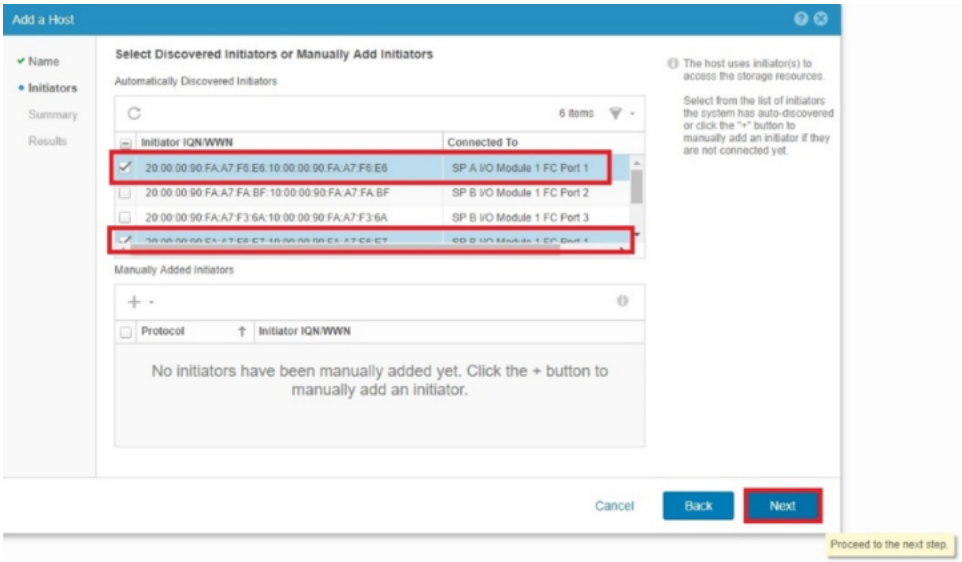


4. Enter the Hostname of the Head Unit.
5. Under **Operating System**, select **Linix** from the-drop down menu.
6. Enter the IP address of the Head Unit.

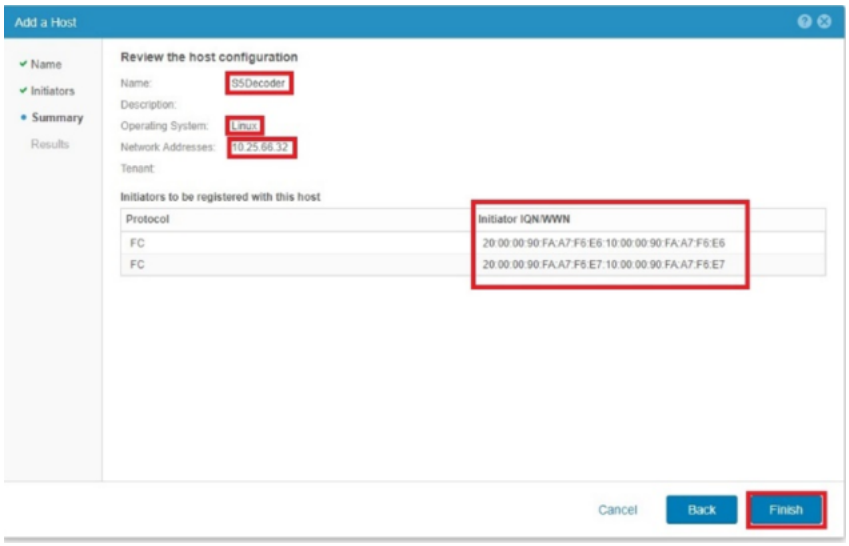
7. Click **Next** to proceed to the next section.



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.



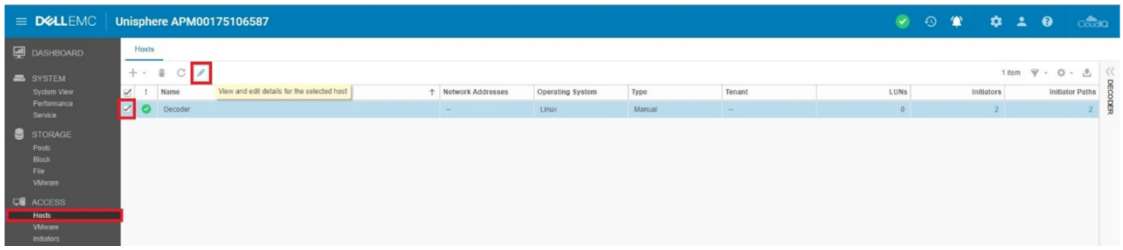
9. Make sure that the **Name**, **OS**, **IP** and **WWNs** are correct and click **Finish**.



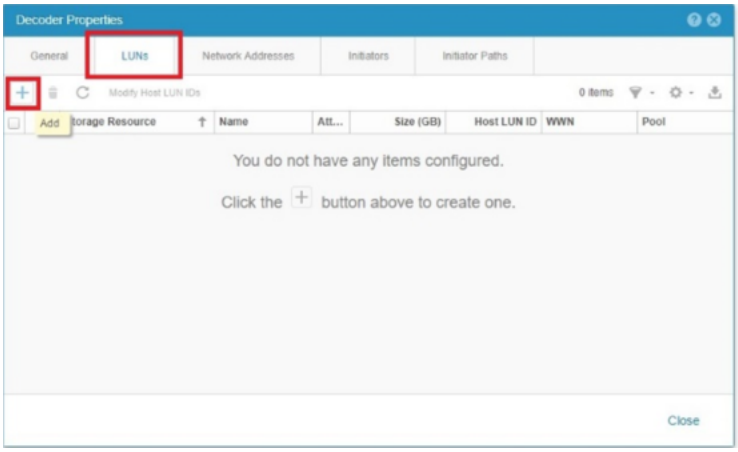
10. Repeat steps 2-9 for all Head Units.
11. In the Initiators section, select the two initiators that correspond to the correct port associated with the Head Unit. Then click “Next” to proceed.

Task 5 - Assign LUNS to Hosts

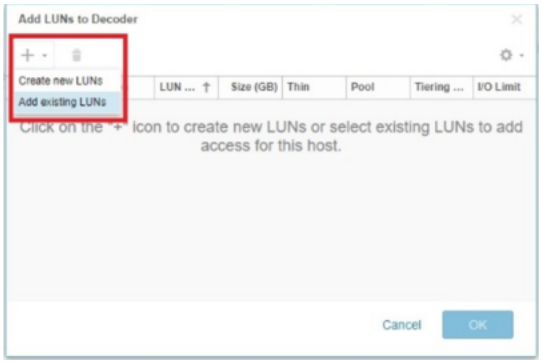
1. From the **Access** section, click **Hosts**, select the head unit (for example, **Decoder**) and click  (edit) to view and edit details for the selected host.



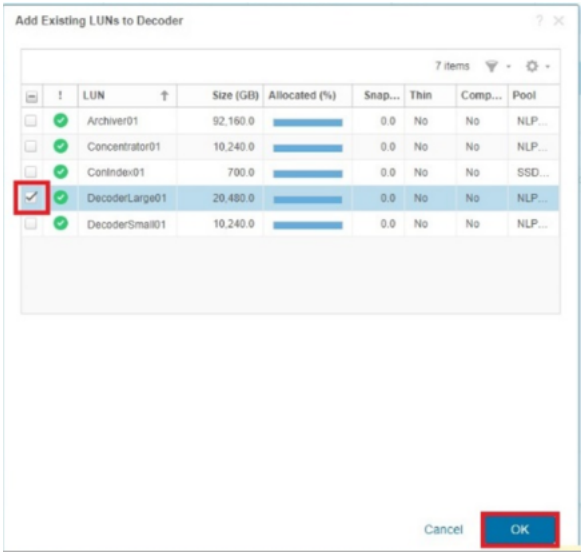
2. In the **Properties** section, select the **LUNS** tab and click + (Add icon).



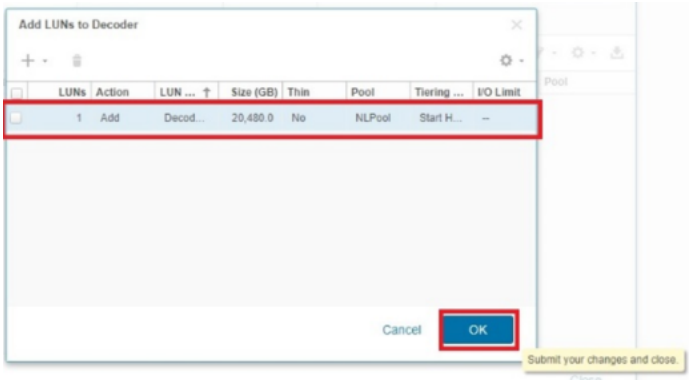
3. From the **Add LUNs to <Host>** popup, click + > **Add existing LUNs**.



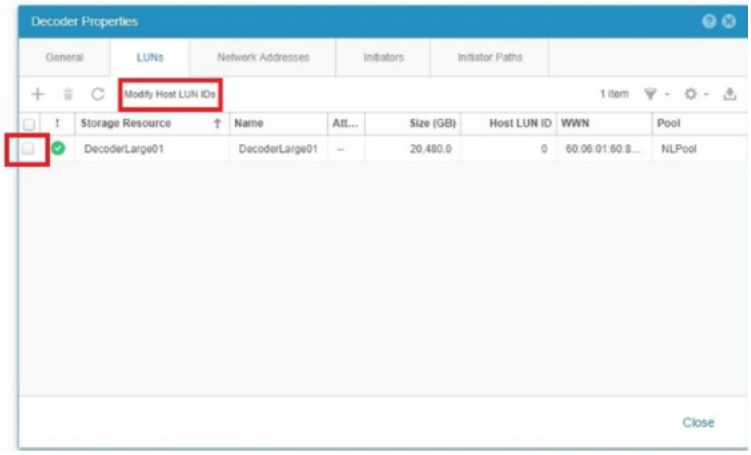
4. Select the LUN to add to the Head Unit and **OK**.



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 NetWitness 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 NetWitness 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.

4. 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=600601609D90460069967 45A46B60AB6 [DecoderSmall01]
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
```

5. Verify that the PowerPath license is installed using the `emcpreg` command.
   ```
   [root@NWAPPLIANCE24932 ~]# emcpreg -list
   Key BQPO-DB4M-VFC2-Q24R-ML9Z-EQTU
    Product: PowerPath
    Capabilities: Al
   ```

6. 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|.*/|" ]
   ```

7. Run the following commands in this order:
   a. `systemctl enable PowerPath.service`
   b. `systemctl start PowerPath.service`

8. Reboot the Decoder.

9. 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 NetWitness Community for detailed instructions for setting up 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
- Decommision 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-hotspare 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=logdecodersmall 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 concatonate 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/logdecodersmall-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 decomissioned.
   ```
   #/dev/logdecodersmall/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:

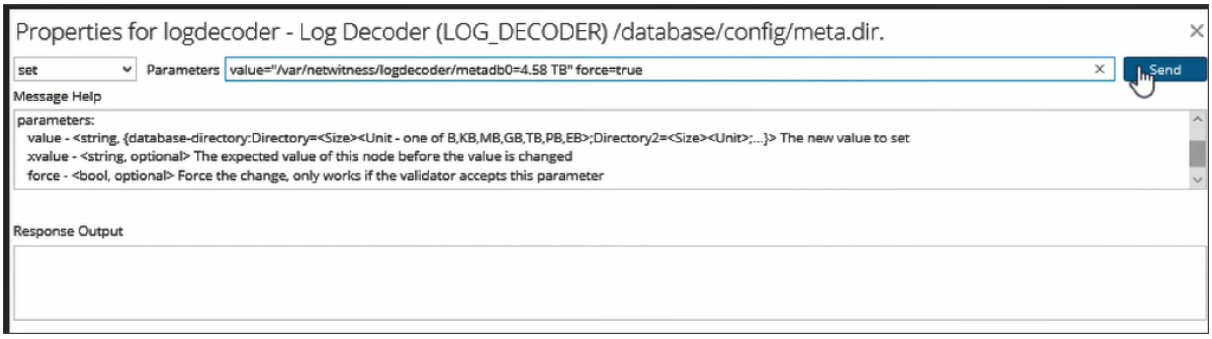
   b. In the same way, copy the packet database in `packet.dir` to `packet.dir.warm`.
   c. Copy the session database in `session.dir` to `session.dir.warm`.
3. Go to **index** > **config** and copy `index.dir` to `index.dir.warm`.

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/sda1                              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' and add 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.

Decommision 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. NetWitness 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 concatonate 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**.

10. 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.

1. Remove the logical volume. Run `lvscan` to get a list of the logical volumes.
2. Run `lvremove` on the old logical volumes, for example:
   `/dev/logdecodersmall/decoroot /devlvremove /dev/logdecodersmall/index /dev/logdecodersmall/sessiondb /dev/logdecodersmall/metadb /dev/logdecodersmall/packetdb`
3. Remove the volume groups. Run `vgscan` to get a list of volume groups.
4. Run `vgremove` on the old volume groups (be careful not to remove any volume groups that end in 0, as they are PowerVault).
5. Run `pvscan` to view block devices that are freed up.
6. 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.

1. Attach two PowerVaults to a separate PERC controller on the Decoder.
2. Create the devices.
   a. 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-hotspare 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 perc card slot.

4. 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
```

5. 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.

6. 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
```

7. Detach the physical connections from the DACs to the host.

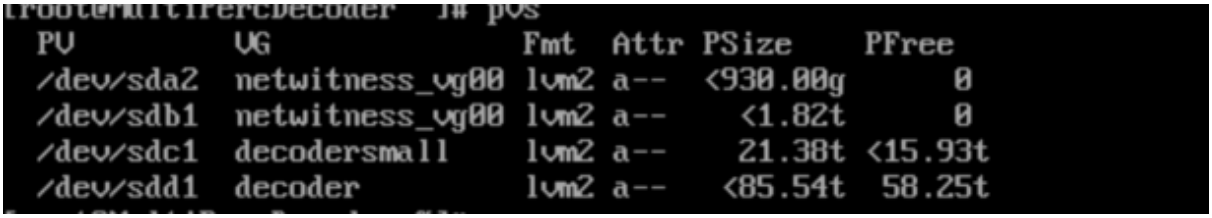
8. Verify that the Physical volumes are moved from the DACs to the PowerVaults.

   a. Reboot the host.

   ```
   reboot
   ```

   b. Verify that the **/etc/fstab** file is correct.

   c. Run the `pvs` command and make sure that the **PSize** and **PFree** values are correct on the PowerVault.



```
  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
```

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

SASE Node-x (Decoder/Concentrator) - GCP Persistent Disk (PD) Storage Configuration

This section contains:

Introduction

The SASE node-x (decoder/concentrator) storage configuration is dependent on a few configuration attributes defined in */opt/rsa/saTools/cloud/host-models.yml* (referred to as host-models file) and */opt/rsa/saTools/cloud/sase-deployment-models.yml* (referred to as sase-deployment-models file).

The configuration files are installed on *Admin Server/Node-z* during orchestration (*Refer to SASE Installation Guide for details*). When the SASE deployment script (*nw-create-cloud-hybrid*) is run for the first time, it copies both the configuration files (*/opt/rsa/saTools/cloud/host-models.yml* and */opt/rsa/saTools/cloud/sase-deployment-models.yml*) to /root/.sase/ directory on the Admin Server/Node-z. Any subsequent SASE deployments refer to */root/.sase/host-models.yml* and */root/.sase/sase-deployment-models.yml*. All subsequent updates such as changing a model-name must be made to */root/.sase/sase-deployment-models.yml*.

The SASE node's disk specifications such as disk_name, disk_type, disk_size for each node along with gcp virtual machine type (machine_type) is collectively referred to as a model. Three models: *c1r6m30*, *c1r12m60* and *c1r23m120* are defined in the *host-models.yml*. One of these models is assigned as model-name attribute value in *sase-deployment-models.yml*.

The SASE node(s) deployment and storage configuration are tightly coupled and storage needs should be considered before installation or deployment. The SASE Installation guide should be cross referenced for complete attribute definition in *host-models* and *sase-deployment-models* files. As a best practice, the storage requirements should be identified and *host-models.yml* and *sase-deployment-models* files updated before deploying or installation of the nodes.

The below outlines the steps for successful installation and storage configuration of SASE nodes in GCP.

1. Identify the storage requirements for each of the SASE services - decoder and concentrator.
2. Identify or define the storage model: Identify a predefined model or define a new custom model. If defining a custom model, copy the */opt/rsa/saTools/cloud/host-models.yml* and */opt/rsa/saTools/cloud/sase-deployment-models.yml* to */root/.sase/* and update the *model-name* attribute for every node in */root/.sase/sase-deployment-models.yml*.

3. Deploy the SASE GCP nodes: Execute the nw-create-cloud-hybrid script on admin node or node-z to complete the installation. *Refer to SASE Installation Guide for details.*
4. Complete storage configuration using **Explore** view of the Decode or Concentrator by logging into *Admin Server* or *UI*.

Identify Storage Requirements

This section covers step 1 defined in Introduction.

Based on storage needs, an appropriate model-name (*c1r6m30*, *c1r12m60* or *c1r23m120*) is assigned to *model_name* attibute in */root/.sase/sase-deployment-models.yml* file. During SASE Node-x deployment the setup script*(/usr/bin/nw-create-cloud-hybrid*) refers to additional_disks attribute value in sase-deployment-models.yml to determine wheather storage disks must be created or not. When the additional_disks is set to true for a node, the corresponding model_name attribute value is used to identify the model specification (such as disk_name, disk_type and disk_size) in */root/.sase/host-models.yml* and the corresponding disks are created and assigned (but not configured i.e these disks are ready for storing NW databases) to the SASE node.

When the predefined models don't satisfy the customer requirements, custom models (*Refer to Define a custom storage model*) can be defined. Please contact Professional Services/ NW Support for details on defining custom models.

Example: Sample content of sase-deployment-models.yml identifying *model_name* and *additional_storage* attribute values:

```
# Define model name to configure host and storage
model_name: c1r12m60
additional_storage: true
```

Note:
- The storage_class and warm_retention attributes are not supported in NW 12.4. These are developmental features and should be ignored.
- The disk_size value is always defined in GB (Gigabytes).
- c1default is for testing purposes only.

Model	Description
c1r6m30	• Creates and attaches persistent disks to decoder service that are capable of 6 days of packet retention at 1gbps line rate (capture) with 100% utilization. • Creates and attaches persistent disks to concentrator service (concentrator volume) and pd-ssd disks for index that are capable of 30 days of concentrator meta retention at 1gbps line rate (capture) with 100% utilization.

Model	Description
c1r12m60	• Creates and attaches persistent disks to decoder service that are capable of 12 days of packet retention at 1gbps line rate (capture) with 100% utilization. • Creates and attaches persistent disks to concentrator service (concentrator volume) and pd-ssd disks for index that are capable of 60 days of concentrator meta retention at 1gbps line rate (capture) with 100% utilization.
c1r23m120	• Creates and attaches persistent disks to decoder service that are capable of 23 days of packet retention at 1gbps line rate (capture) with 100% utilization. • Creates and attaches persistent disks to concentrator service (concentrator volume) and pd-ssd disks for index that are capable of 120 days of concentrator meta retention at 1gbps line rate (capture) with 100% utilization.

Identify or Define Storage Model

This section covers step 2 defined in the Introduction section.

Every predefined model defines *machine_type* attribute for boot disk and a minimum of two disks (one set) for storage (referred to as initial disks or default storage disks or default disks). The default storage disks for decoder service are named *(disk_name* attribute in *host-models.yml*) as *decoder* and *decodersmall*. The default disks for concentrator are index and concentrator. The disk naming convention is also tied to subsequent configuration steps such a creation of partitions, volume allocation to service. It also helps to associate the corresponding partition scheme that is created on that particular disk when using NW REST API to complete storage configuration

Each node (*Decoder* or *Concentrator*) has its own storage attributes defined under disks (*Refer to screen shot below*). The attributes are further classified into two groups for each service with each group corresponding to a specific logical volume name. For decoder service, the two groupings refer to the logical volume names: *decoder* and *decodersmall* and for concentrator service these are *concentrator* and *index*.

Note: The grouping names (*decoder* and *decodersmall* for decoder and *concentrator* and *index* for concentrator) under *disks* match the *disk_name* attribute values that they define. This relationship (underlined in red below) MUST be maintained when defining a Custom storage model since the disk creation process requires this 1-1 mapping of grouping name to *disk_name*.

host-models.yml snippet highlighting the disk groupings and disk_name relationship (underlined in red):

```
c1r12m60:
  Decoder:
    machine_type: n2-standard-32
    storage_class: STANDARD
    # retention size in TB
    warm_retention: 1
    disks:
      # allocate to decoder root, index, sessiondb, metadb
      decodersmall:
        disk_name: decodersmall
        disk_type: pd-standard
        disk_size: 3000
      # allocate to packetdb
      decoder:
        disk_name: decoder
        disk_type: pd-standard
        disk_size: 65000
      # allocate to decoder root, index, sessiondb, metadb
      decodersmall0:
        disk_name: decodersmall0
        disk_type: pd-standard
        disk_size: 3000
      # allocate to packetdb
      decoder0:
        disk_name: decoder0
        disk_type: pd-standard
        disk_size: 65000
```

```
Concentrator:
  machine_type: n2-standard-32
  storage_class: STANDARD
  # retention size in TB
  warm_retention: 1
  disks:
    # allocate to concentrator root, metadb, sessiondb
    concentrator:
      disk_name: concentrator
      disk_type: pd-standard
      disk_size: 40000
    # allocate to index
    index:
      disk_name: index
      disk_type: pd-ssd
      disk_size: 2000
    # allocate to concentrator root, metadb, sessiondb
    concentrator0:
      disk_name: concentrator0
      disk_type: pd-standard
      disk_size: 40000
    # allocate to index
    index0:
      disk_name: index0
      disk_type: pd-ssd
      disk_size: 2000
```

Each disk attached to a SASE Decoder node is configured as a separate logical volume to host the service's database(s) such as *packetdb*, *metadb*, *sessiondb* and *index*. The disk named *(disk_name* attribute in *host-models* file) as decoder corresponds to decoder volume and hosts packetdb and the disk named (*disk_name* attribute in *host-models* file) as *decodersmall* corresponds to *decodersmall volume* and *host sessiondb*, *metadb* and *index*. If decoder storage requirements mandate larger disks (i.e 12 days of Packet retention versus 6 days of packet retention or 23 days of packet retention versus 12 days or 6 days) then multiple sets of disks are defined and created.

If a model defines two sets of disks, the first is the default disks named (*disk_name*) as *decoder* and *decodersmall* and the second set is named as *decoder0* and *decodersmall0*, likewise when a model defines four sets of disks ex: *c1r23m120*, first set is default disks (named *decoder* and *decodersmall*) , second set of disk named as *decoder0* and *decodersmall0* and the third set of disks are named as *decoder1* and *decodersmall1* and fourth set is *decoder2* and *decodersmall2*. Any additional disk set follow the similar naming convention as described above. i.e decoder<COUNTER> and decodersmall<COUNTER> where COUNTER is incremented by 1 for any new set of disks starting with 0. The first part of the name (decoder , decodersmall) helps in associating the disk with volume name/type that is created later. The volume name is used to identify the appropriate Decoder service's database during service allocation.

Similar pattern is followed with *concentrator disks*. The default disk set is *index* and *concentrator*. The concentrator is *pd-standard* (*disk_type*) and index is disks are *pd-balanced*.The *index disk* is used to host *index database* and *concentrator disk* is used to *host root*, *sessiondb* and *metadb*. When more than one disk set is created, these are named as *index<COUNTER>* and *concentrator<COUNTER>* where COUNTER starts with 0 and increments by 1 (similar pattern as decoder disks described above). *Refer to above screen shots for pattern identification for c1r12m60.*

The *additional_storage* attribute value in *sase-deployment-models.yml* determines whether the storage disks are created or not. A value of true creates the disks and false skips the disk creation. The default value is false. Updating the *additional_storage* to *true* and re-running the *nw-create-cloud-hybrid* creates the storage disks. These disks are not deleted by re-running *nw-create-cloud-hybrid* after setting *additional_storage* to *false*.

The storage requirements must be identified to select the appropriate storage model. The retention days along with capture rate is used to identify the appropriate model. After the model is identified, the *model_name* value for every SASE node must be updated in the *sase-deployment-models.yml* file. If the multiple nodes are deployed, then each node's *model_name* must be assigned with appropriate value.

For both predefined and custom models, after identifying the model (*Refer to Identify a Pre-defined Storage Model or Define a Custom Storage Model section(s) below*) the **model_name** attribute value (in *sase-deployment-models.yml*) is set to the identified model name.

Refer Appendix B for SASE Decoder Storage Configuration and Appendix C for SASE Concentrator Storage Configuration.

Deploy SASE Node(s)

This section covers step 3 defined in the Introduction section.

Assumptions:

- The *model_name* is updated with correct host-model and *additional_storage* attributes set to *true* in *sase-deployment-models.yml*
- When opting for custom storage model, the */root/.sase/host-model.yml* is updated with the custom model definition.

Execute the *nw-create-cloud-hybrid* script on *node-z* or *Admin Server* to complete the installation of SASE node-x. The storage disks are created and attached to the SASE nodes but not configured to host the NW service's database.

The *gcp SASE instance name* (SASE Decoder deployed) is a combination of *nw-<name>-<region_name >-<zone_suffix>*. The *name*, *region_name* and *zone_suffix* attributes are defined in *sase-deployment-model* file for the decoder node. Every SASE node follows the similar naming convention.

The deployed SASE node can be accessed by logging into *GCP account → Compute Engines → Virtual Machines-> Search* for the SASE instance using the name.

Refer to SASE Installation Guide for details on installation.

Configure SASE Node(s) Storage

This section covers step 4 defined in the Introduction section.

Assumptions:

- The SASE decoder node is successfully bootstrap and orchestrated in gcp.
- All the storage disks associated with the node's model are created. As noted earlier, the disks are not created when an incorrect *model_name* is used or the *additional_disks* attribute is set to false. Incorrect *model_name* requires uninstalling the node and recreating it. If incorrect *additional_disks* value (false) is used, set the value to true and re-run the *nw-create-cloud-hybrid* script.

```
nw-create-cloud-hybrid --enable-cloud-sase
```

Configure SASE Decoder Storage

Refer to Appendix B: Sample scenario for Configuring SASE Decoder Storage to complete the configuration of the storage disks created during installation.

Configure SASE Concentrator Storage

Refer to Appendix C: Sample scenario for Configuring SASE Concentrator Storage to complete the configuration of the storage disks created during installation.

Extend Storage for SASE Node

Note: Only pre-defined host-models can be extended. Custom host models can not be extended.

Customer storage requirements such as higher retention for packet and meta data may change over time after initial storage configuration. SASE storage models support additional storage allocations. The first column in *Storage Extension Matrix* table below lists the current host model deployed and the second column shows the available host-models that the initial model can be extended.

Storage Extension Matrix:

Current Model (model_name)	Supported Storage Extensions (model_name)
c1r6m30	c1r12m60 c1r23m120
c1r12m60	c1r23m120
c1r23m120	N/A (cannot be extended)

Extend Decoder or Concentrator Storage

Extending decoder storage involves the following steps on the Admin Server or Node-z:

1. Identify the new model in *host-models.yml* and update the node's *model_name* value with the new model, *additional_storage* to *true* and click **Save**.
2. Execute the below command to extend the storage.

   ```
   nw-create-cloud-hybrid --enable-cloud-sase
   ```

3. Complete the storage configuration using NW REST API by logging into Admin Server or UI and Use REST API utility to create partitions and service allocations (Navigate to **decoder's explorer view ->deviceappliance->properties->Right Click** and select **Properties** drop-down).
4. Complete the storage configuration of new disks added in the above step using REST API. *Refer to Configure Storage Using the REST API section for details. Refer to Appendix B - Sample Scenario for Configuring SASE Decoder Storageand Appendix C - Sample Scenario for Configuring SASE Concentrator Storage for sample scenarios.*

Appendix

This section contains:

Appendix A - Defining a Custom Host Model

Custom storage models can be created subject to Google Cloud Persistent Disk restrictions.

The below steps must be executed in sequence for using custom model before starting the SASE installation.

1. Copy */opt/rsa/saTools/cloud/sase-deployment-models.yml* and */opt/rsa/saTools/cloud/host-models.yml* to */root/.sase/*. Update */root/.sase/host-models.yml* with the custom model specification and **Save**.
2. Update the *model_name* attribute value with the above custom model name in */root/.sase/sase-deployment-models.yml* and **Save**.
3. Complete the SASE node deployment. *Refer to SASE Installation Guide*.

All the attributes must be assigned a valid value. It is recommended to append custom model definition to existing content in */root/.sase/host-models.yml*. *Refer to the SASE Installation Guide for details on the attribute description.*

Sample Custom Model Definition for Decoder

The sample custom configuration assumes defining a model capable of 1gbps capture with 10 days of packet retention and 40 days of concentrator meta retention : ***c1r10m40*** , the *machine_type* as *n2-standard-32* (Refer to Google Cloud documentation for details on all the available VM types) and disk_type as *pd-standard* and the storage requirements of ***98000 GB*** for packet db and ***8000 GB*** for other databases (metadb, sessiondb and index).

Due to gcp restrictions on the disk size, the model requires two sets of disks. Following the naming convention described earlier, the two disk sets are named (disk_name) as ***decoder***, ***decodersmall***, ***decoder0*** and ***decodersmall0***.

Custom model definition for model_name: ***c1r10m40***.

host-models.yml:

```
c1r10m40:
    machine_type: n2-standard-32
    disks:
      Decoder:
        decodersmall:
          disk_name: decodersmall
          disk_type: pd-standard
          disk_size: 4000
        decoder:
          disk_name: decoder
          disk_type: pd-standard
          disk_size: 47000
        decodersmall0:
          disk_name: decodersmall0
          disk_type: pd-standard
          disk_size: 4000
        decoder0:
          disk_name: decoder0
          disk_type: pd-standard
          disk_size: 47000
```

Partial Content of sase-deployment-models.yml updated with custom model_name Value:

```
#Container element defining all nodes that will be created within this region.
nw_nodes:

  # First Node to be created. This element is just an arbitrary name for the type of
  # node to be created and provisioned in this region's subnet.
  decoder:

    # Name of the instance known to the nw-ppn network.
    name: decoder_w

    # This node's ip in C.I.D.R format. This address MUST be within the ppn_cidr range.
    ppn_cidr_ip: 172.30.30.6/24

    # This value is concatenated with the region to define the zone
    # that the nw node will be installed into on GCP.
    zone_suffix: '-b'

    # The size and type of boot disk attached to nw node when it is created.
    boot_disk_size: 196
    boot_disk_type: pd-standard

    # Define model name to configure host and storage
    model_name: c1r10m40
    additional_storage: true

    # Used to determine if the calling script will automatically bootstrap and accept
    # the node keys in the Admin Server. This allows for either automated or
    # manual orchestration of a NetWitness Category to the node.
    bootstrap: true

    # Used to determine if the calling script will automatically orchestrate
    # a NetWitness Category to the node.
    orchestrate: true

    # The NetWitness Category to be orchestrated. Must be an exact value (Case Sensitive
    category: Decoder
```

Sample Custom Model Definition for Concentrator

The sample custom configuration model (*c1r10m40*) assumes the *machine_type* as *n2-standard-32* (Refer to Google Cloud documentation for details on all the available VM types) and concentrator meta retention of 40 days. The concentrator disk_type is *pd-standard*, index disk_type is *pd-balanced* and the storage requirements for metadb, sessiondb, root is ***24000 GB*** (for 40 days) and index database is ***6000 GB*** (for 40 days).

Since these sizes fall with in the GCP limitations on per disk disk maximum size, only one set (default name) is needed: *concentrator*, *index*.

Custom model definition for model_name: c1r10m40

host-models.yml:

```
c1r10m40:
    machine_type: n2-standard-32
    disks:
      Concentrator:
        concentrator:
          disk_name: concentrator
          disk_type: pd-standard
          disk_size: 24000
        index:
          disk_name: index
          disk_type: pd-ssd
          disk_size: 6000
```

Partial Content of sase-deployment-models.yml updated with custom model_name Value:

```
#Container element defining all nodes that will be created within this region
nw_nodes:

  # First Node to be created. This element is just an arbitrary name for the
  # node to be created and provisioned in this region's subnet.
  concentrator:

    # Name of the instance known to the nw-ppn network.
    name: concentrator

    # This node's ip in C.I.D.R format. This address MUST be within the ppn_c
    ppn_cidr_ip: 172.30.30.6/24

    # This value is concatenated with the region to define the zone
    # that the nw node will be installed into on GCP.
    zone_suffix: '-b'

    # The size and type of boot disk attached to nw node when it is created.
    boot_disk_size: 196
    boot_disk_type: pd-standard

    # Define model name to configure host and storage
    model_name: c1r10m40
    additional_storage: true
    # Used to determine if the calling script will automatically bootstrap ar
    # the node keys in the Admin Server. This allows for either automated or
    # manual orchestration of a NetWitness Category to the node.
    bootstrap: true

    # Used to determine if the calling script will automatically orchestrate
    # a NetWitness Category to the node.
    orchestrate: true

    # The NetWitness Category to be orchestrated. Must be an exact value (Cas
    category: Concentrator
```

Appendix B - Sample Scenario for Configuring SASE Decoder Storage

The sample storage configuration in this section uses REST API utility to complete the configuration of the storage disks created during installation.

Refer Configure Storage Using the REST API section for details on REST API usage.

Assumptions:

- The SASE Decoder node is deployed using *c1default (model_name)* and *additional_storage: true* in *sase-deployment-models.yml* during SASE deployment. *c1default* is a test model for illustration purposes.
- The storage disks corresponding to *c1default* model are created and attached to the decoder node (by the installation scripts). These disks are not yet configured.

```
c1default:
  Decoder:
    machine_type: n2-standard-4
    storage_class: STANDARD
    # retention size in TB
    warm_retention: 1
    disks:
      # allocate to decoder root, index, sessiondb, metadb
      decodersmall:
        disk_name: decodersmall
        disk_type: pd-standard
        disk_size: 641
      # allocate to packetdb
      decoder:
        disk_name: decoder
        disk_type: pd-standard
        disk_size: 69
```

Follow these steps to configure SASE Decoder storage:

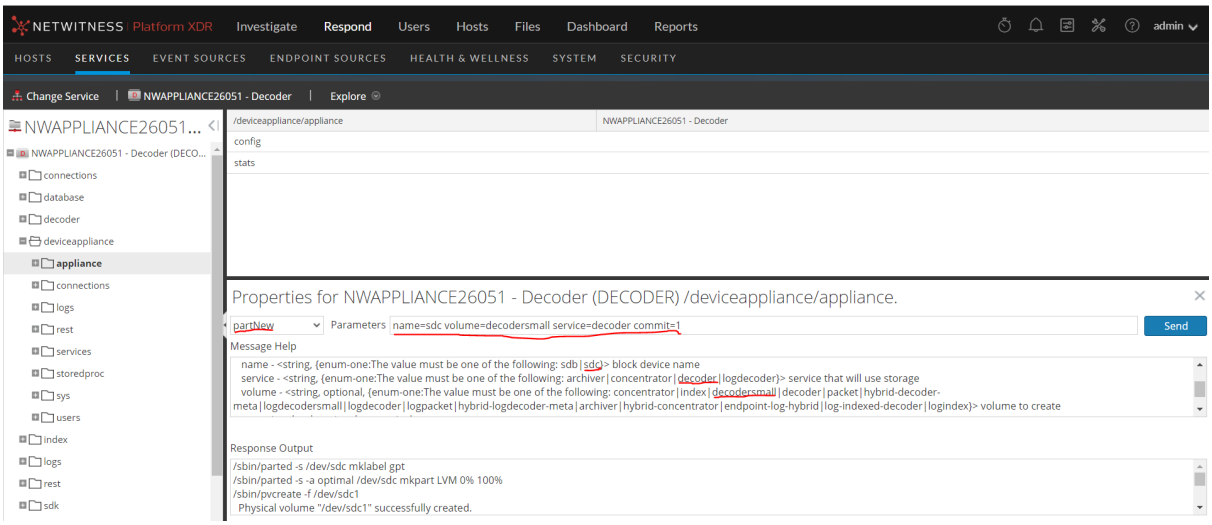
1. Login to SA UI. To list all the devices, Navigate to **Hosts -> Select the Decoder node -> Services -> Actions -> View -> Explore -> deviceappliance -> appliance-> Right Click -> Properties** drop-down **->** Select **devlist -> Send**.

   All the block devices (both configured and unconfigured) are returned. The configured devices has the *'used=1'* and unconfigured has '*used=0*'. In this case, the block devices are sdb and sdc. sda is the boot disk and no changes are allowed.

   Note: decodersmall must be partitioned before decoder volume.

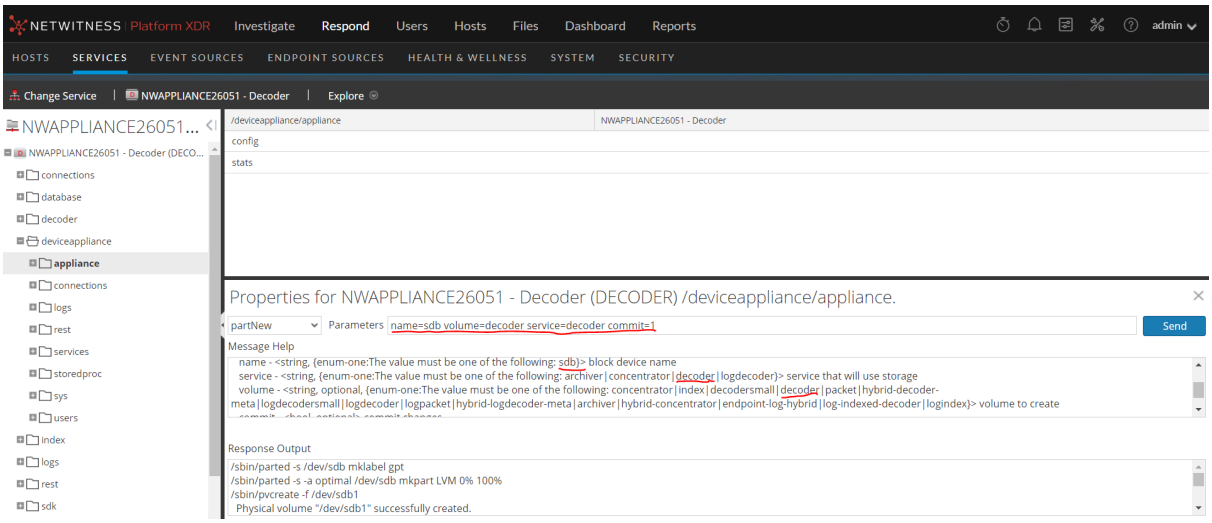
2. Use **partNew** command to partition unused block devices. For decoder service, the larger devices (decoder) are always allocated partitioned for packetdb and smaller devices (decodersmall) are allocated for *root/session/meta* and *index* databases. To partition sdc, select **partNew** from the **Properties** drop-down with the following parameters and click **Send**..

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



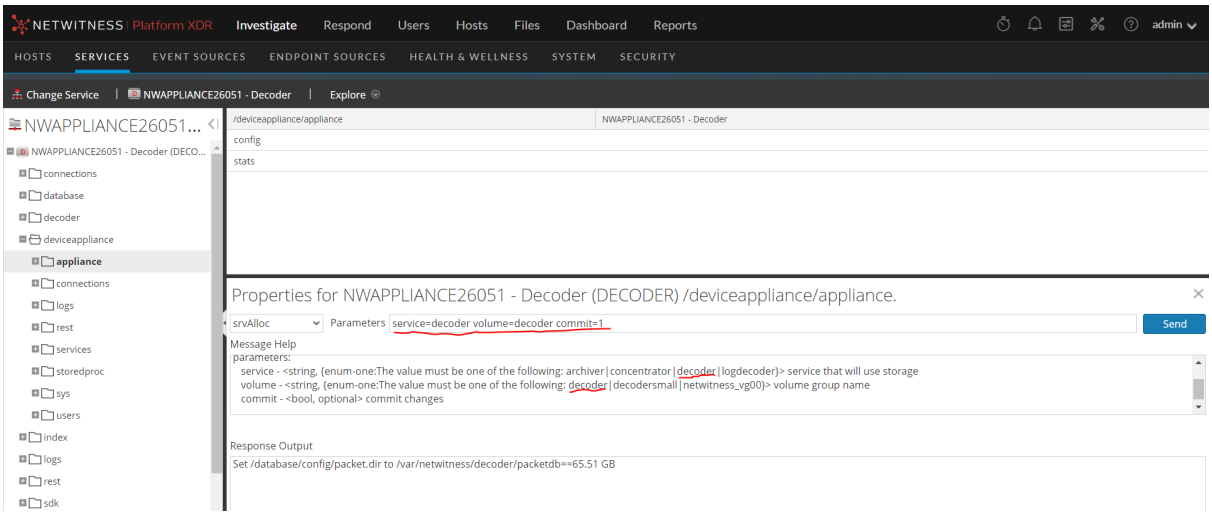
3. To partition *sdb* device, select 'partNew' from the **Properties** drop-down with the following parameters and click **Send**.

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



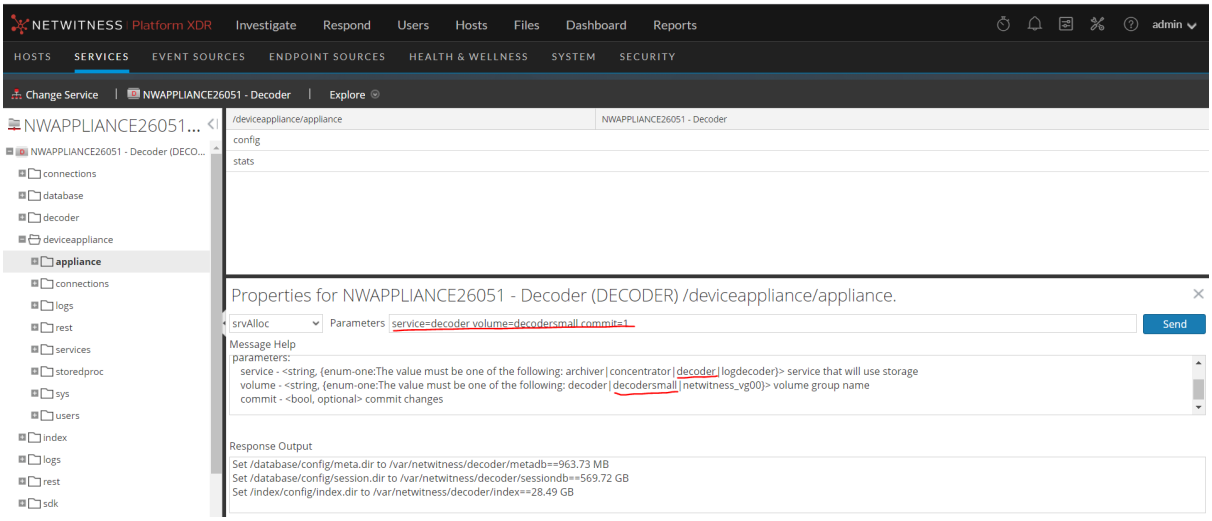
4. Allocate the above configured partitions to decoder service by selecting **srvAlloc** in the **Properties** drop-down and click **Send**.

```
service=decoder volume=decoder commit=1
```



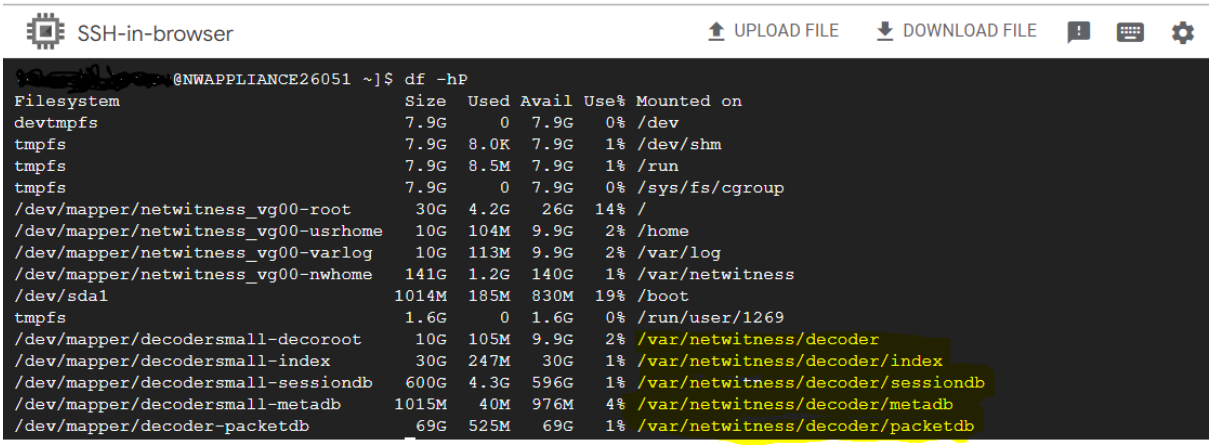
5. Allocate decodersmall to decoder service using *srvAlloc* property.

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



6. View the storage allocation by issuing the below command in SSH-in-browser. The configured storage is highlighted in yellow.

```
df -hP
```



Appendix C - Sample Scenario for Configuring SASE Concentrator Storage

The sample storage configuration in this section uses REST API utility to complete the configuration of the storage disks created during installation. *Refer to Configure Storage Using the REST API section for details on REST API usage.*

Assumptions:

- The SASE Concentrator node is deployed using *c2default (model_name)*and *additional_storage: true* values in *sase-deployment-models.yml* during SASE deployment. *c2default* is a test or custom model for illustration purposes.
- The storage disks corresponding to *c2default* model are successfully created and attached to the

concentrator node. These disks are not yet configured.

```
Concentrator:
 machine_type: n2-standard-4
 storage_class: STANDARD
 # retention size in TB
 warm_retention: 1
 disks:
  # allocate to concentrator root, metadb, sessiondb
  concentrator:
   disk_name: concentrator
   disk_type: pd-standard
   disk_size: 38
  # allocate to index
  index:
   disk_name: index
   disk_type: pd-ssd
   disk_size: 11
  # allocate to concentrator root, metadb, sessiondb
  concentrator0:
   disk_name: concentrator0
   disk_type: pd-standard
   disk_size: 38
  # allocate to index
  index0:
   disk_name: index0
   disk_type: pd-ssd
   disk_size: 11
```

Follow these steps to configure Concentrator storage:

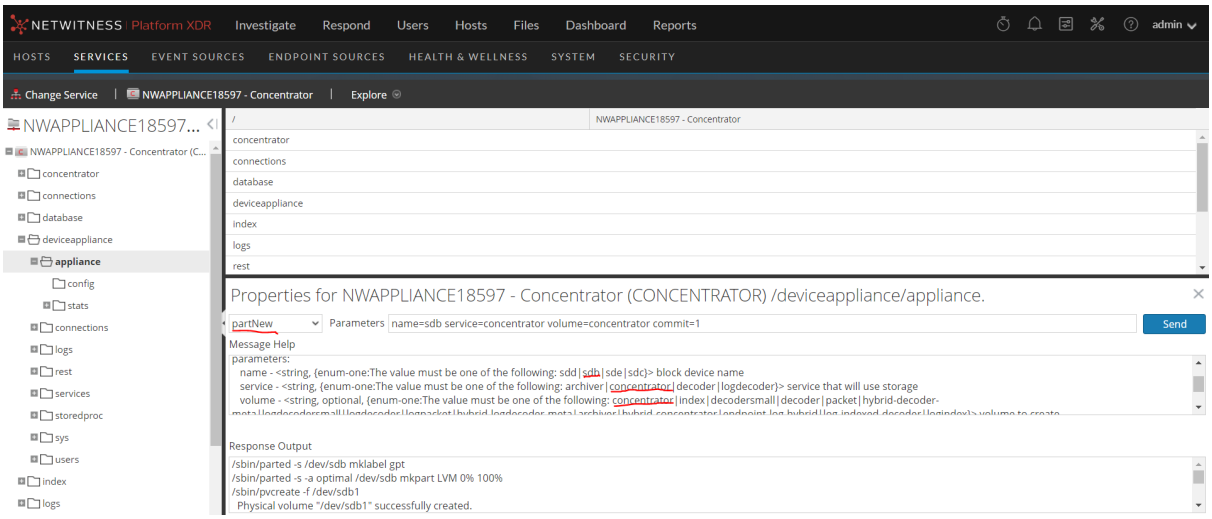
1. Log into SA UI. To list all the devices, Navigate to **Hosts- > Select the Concentrator node -> Services -> Actions -> View -> Explore -> deviceappliance -> appliance -> Right Click -> Properties drop-down -> Select devlist -> Send**.

   Response Output:

   In this case, the block devices are *sdb*, *sdc*, *sdd* and *sde*. The '**used=0**' indicates that this device is not configured yet. When configured, the 'used' attribute value is set to '**1**'. As noted earlier, sda is the boot disk and no changes are allowed.

2. Use **partNew** command to partition unused block devices. For concentrator service, the smaller devices (disk_type: pd-ssd) are allocated to index database and larger devices are allocated to root,session and metadb. To partition sdb , select 'partNew' from the **Properties** drop-down with the following parameters and click **Send**.

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

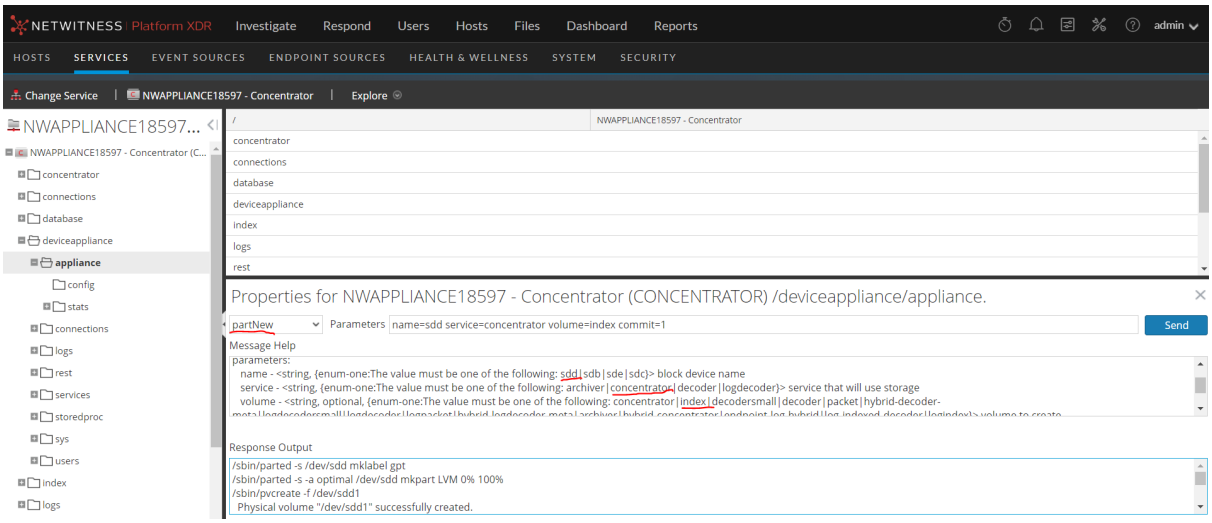


3. Repeat the above step for **concentrator0** volume. Use **partNew** command to partition unused block devices. To partition *sdc*, select 'partNew' from the **Properties** drop-down with the following parameters and click **Send**.

   ```
   name=sdc volume=concentrator0 service=concentrator commit=1
   ```

4. To partition sdd device, select 'partNew' from the **Properties** drop-down with the following parameters and click **Send**.

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

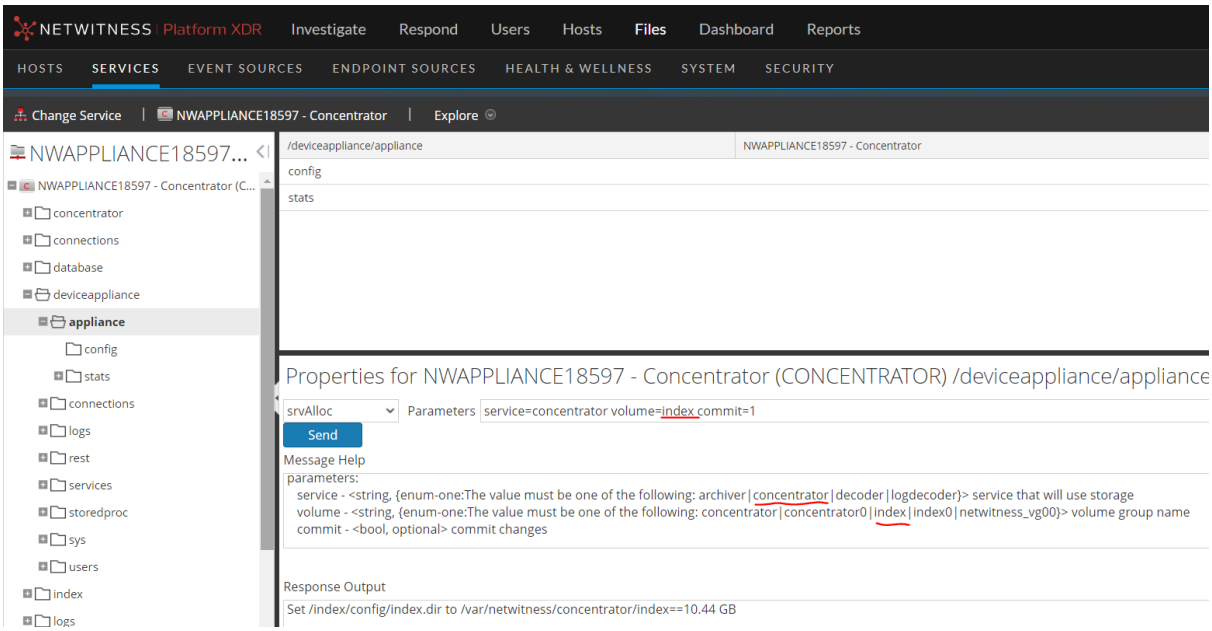


   Repeat the above for sde.

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

5. Allocate the index volume to concentrator service using **srvAlloc** from the **Properties** drop-down with the following parameters and clicking **Send**.

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



6. Complete the service allocation of the three remaining volumes (index, index0, concentrator0) using srvAlloc property using the above step.

```
service=concentrator volume=index0 commit=1
service=concentrator volume=concentrator commit=1
service=concentrator volume=concentrator0 commit=1
```

7. The configured storage can be viewed by issuing the below command in SSH-in-browser. The configured storage is highlighted in yellow.

```
df -hP
```

```
[root@NWAPPLIANCE18597 ~]# df -hP
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                            7.9G     0  7.9G   0% /dev
tmpfs                               7.9G  8.0K  7.9G   1% /dev/shm
tmpfs                               7.9G  8.6M  7.9G   1% /run
tmpfs                               7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-root     30G  4.1G   26G  14% /
/dev/mapper/netwitness_vg00-usrhome  10G  104M  9.9G   2% /home
/dev/mapper/netwitness_vg00-varlog   10G  115M  9.9G   2% /var/log
/dev/mapper/netwitness_vg00-nwhome  141G  1.2G  140G   1% /var/netwitness
/dev/sda1                          1014M  185M  830M  19% /boot
tmpfs                               1.6G     0  1.6G   0% /run/user/1269
/dev/mapper/concentrator-root        30G  248M   30G   1% /var/netwitness/concentrator
/dev/mapper/concentrator-sessiondb  811M   38M  773M   5% /var/netwitness/concentrator/sessiondb
/dev/mapper/concentrator-metadb     7.2G   84M  7.2G   2% /var/netwitness/concentrator/metadb
/dev/mapper/index-index              11G  111M   11G   1% /var/netwitness/concentrator/index
/dev/mapper/index0-index             11G  111M   11G   1% /var/netwitness/concentrator/index0
/dev/mapper/concentrator0-sessiondb 3.8G   60M  3.8G   2% /var/netwitness/concentrator/sessiondb0
/dev/mapper/concentrator0-metadb     35G  277M   34G   1% /var/netwitness/concentrator/metadb0
```

Appendix D - Sample Scenario for Extending for SASE Decoder Storage

1. SSH into node-z (Admin Server). Identify the new model and update *model_name*, *additional_storage* attribute values in */root/.sase/sase-deployment-models.yml* and save the changes.

   The below screen shot shows the updated storage configuration in */root/.sase/sase-deployment-models.yml: model_name: c1r12m60, additional_storage: true*. Additional storage disks are created for the updated *model_name* only when *additional_storage* is set to true.

```
# Define model name to configure host and storage
model_name: c1r12m60
additional_storage: true
```

2. Run the *nw-create-cloud-hybrid* script on node-z.

   `nw-create-cloud-hybrid --enable-cloud-sase`

```
[root@124Admin026 12.4.0.0]# nw-create-cloud-hybrid --help
Usage:
 nw-create-cloud-hybrid command [options]

Commands:
 --enable-cloud-sase            Deploys NetWitness SASE focused assets to the cloud
 --disable-cloud-sase           Undeploys NetWitness SASE focused assets from the cloud
 --upgrade-overlay-network      Upgrades overlay network resources
 --reissue-all-certs            reissues all overlay network certificates
 --reissue-node-certs           reissues overlay network certificates for a specific node
     required parameter:
       --uuid                   UUID of NetWitness Node (see nw-manage -l)
 --backup-cloud-nodes, -b       Backup configuration of all cloud nodes
 --restore-cloud-node, -r       Restores configuration of specified Cloud node
     required parameter:
       --uuid                   UUID of NetWitness Cloud Node (see nw-manage -l)
 --check-overlay-status, -c     Checks inter-connectivity of all nw-ppn overlay network hosts
 --check-cert-status, -s        Checks overlay network certificate expiration status

Command Options:
     --cloud-provider           Required: Destination cloud provider (gcp|aws)
     --deployment-model         Optional Name of deployment model in template
                                  defaults to pre-defined '(gcp|aws) default'
     --cloud-key-path           Optional Cloud based Service Account key data path
                                  defaults to .(gcp|aws) specific file
```

3. Login to Admin Server or UI. To list the all the block devices, Navigate to **Hosts -> Select the Decoder node -> Services -> Actions -> View -> Explore -> deviceappliance -> appliance -> Right Click -> Properties** drop-down **-> Select devlist ->** Click **Send**.

   - In this case, the model *c1r12m60* defines four disks for decoder service: *decoder*, *decodersmall*, *decoder0* and *decodersmall*0.
   - The response output for devlist returns both configured and unconfigured block devices. '*used=1*' indicate that the devices are configured. '*used=0*' indicates that the device(s) are unconfigured. No changes are required for configured blockdevice(s).
   - All existing configuration is preserved. When extending storage, existing disks as defined in host-models.yml (in this case *decoder* and *decodersmall*) are not recreated. Only new disks (*decoder0* and *decodersmall0*) are created.

4. Repeat steps 2 through 6 from Appendix B - Sample Scenario for Configuring SASE Decoder Storage section to complete configuration of the unused (used=0) block devices.

Appendix E - Sample Scenario for Extending SASE Concentrator Storage

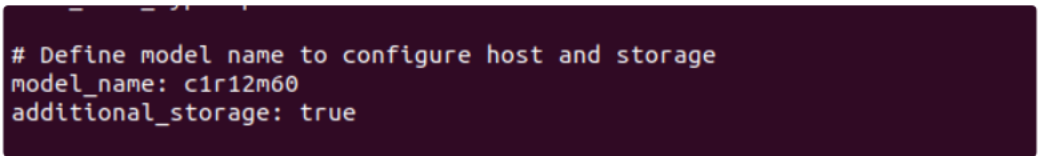
This sample scenario describes the Storage extension for concentrator configured with ***c1r12m60*** model. The supported extension models are *c1r6m30*, *c1r12m60* and *c1r23m120*.

Follow these steps to extend storage for Concentrator:

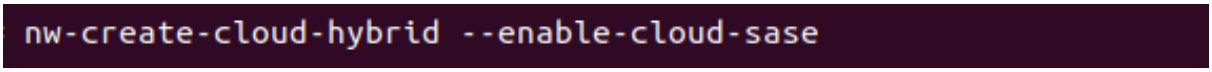
1. SSH into node-z (Admin Server). Identify the new model and update *model_name* and *additional_storage* values in ***/root/.sase/sase-deployment-models.yml*** and save the changes.

   The below screen shot shows an example configuration updated with extended storage *model_name: c1r12m60, additional_storage: true* . If the *additional_storage* is not set to true, additional storage is not created even though the model_name is updated.

   The below screen shot shows the updated storage configuration in */root/.sase/sase-deployment-models.yml: model_name: c1r12m60, additional_storage: true*. Additional storage disks are created for the updated *model_name* only when *additional_storage* is set to true.



2. Run the *nw-create-cloud-hybrid* script on node-z.



3. Login to SA UI. To list the all the block devices, Navigate to **Hosts -> Select the current Host (concentrator node) -> Services -> Actions -> View -> Explore -> deviceappliance -> appliance -> Right Click -> Properties** drop-down **-> Select devlist ->** Click **Send**.

   - The new disks corresponding to '**used=0**' indicate that these disks are not configured.
   - During storage extension, all existing configuration is preserved.

4. Repeat steps 2 through 7 from Appendix C - Sample Scenario for Configuring SASE Concentrator Storage section to complete configuration of the unused (used=0) block devices.

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 NetWitness 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

NetWitness 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.

Enable Security on SED Capable Drive groups on Host with a mix of SED and NON SED Drives

The encryptSedVd.py may fail to identify the SED Capable Virtual Drives when there is mix of both SED and NON-SED drives on the appliance. The below steps are applicable when both SED and NON-SED capable virtual drives exist on the host.

1. SSH to the appliance and enable security on the PERC H740 (mini) Adaptor. The controller number for this adaptor is **0**. The PERC H840 Adaptor is shown as **1**.
   To list all the controllers on the appliance:
   `/opt/MegaRAID/perccli/perccli64 show | egrep -A3 'Model'`
   The first column (**Ctl**) lists out the controller index on the appliance. In this case, the controller '**0**' corresponds to '**PERC H740 Mini**' and controller '**1**' corresponds to '**PERC H840 Adaptor**' . The columns '**DGs**' and '**VDs**' displays the virtual drives and drive groups on the controller.

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 show | egrep -A3 'Model'
Ctl Model          Ports PDs DGs DNOpt VDs VNOpt BBU sPR DS EHS ASOs Hlth
------------------------------------------------------------------------
  0 PERCH740PMini       8  10   3     0   3     0 Opt On  -  N      0 Opt
  1 PERCH840Adapter     8  12   1     0   1     0 Opt On  -  N      0 Opt
[root@116Decoder perccli]#
```

2. To enable the security on the 'PERC H740 (mini) Adaptor', for example, Controller '**0**', execute the following command:
   `/opt/MegaRAID/perccli/perccli64 /c0 set securitykey='<SOME_STRING_VALUE>'!' keyid='< SOME_STRING_VALUE >'`
   Example:
   `/opt/MegaRAID/perccli/perccli64 /c0 set securitykey='Netwitness1!' keyid=1`
   `'Netwitness1' is the securityKey and '1' is ID. Preserve both the Key and keyID securely.`

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 /c0 set securitykey='Netwitness1!' keyid='netwitness'
Controller = 0
Status = Success
Description = None


Controller Properties :
=====================

--------------------
Ctrl Method  Result
--------------------
   0 set Key Success
--------------------
```

3. Identify the correct Drive group (DG) / Virtual Drive (VD) corresponding to the SED Capable drives that you are trying to enable security.
   `/opt/MegaRAID/perccli/perccli64 /c0 /vall show | egrep -A5 'DG/VD'`
   Refer to first two and last column to identify the correct Drive Group (DG) / Virtual Drive (VD) correspond to the 6 SED enabled drives. On Series 6 appliances, there is only one DG/VD with **RAID6**. '**NAME**' column can be used to identify the VD/DG. In this case, the DG/VD is '**2**'. Using a combination of '**Type**', '**Name**' and '**Size**' columns (these are defined by the user when the VDs are created above).

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 /c0 /vall show | egrep -A5 'DG/VD'
DG/VD TYPE  State Access Consist Cache Cac sCC     Size Name
------------------------------------------------------------------------
0/0   RAID1 Optl  RW     Yes     RWBD  -   OFF 931.0 GB
1/1   RAID1 Optl  RW     Yes     RWBD  -   OFF 1.818 TB
2/2   RAID6 Optl  RW     Yes     RWBD  -   OFF 8.730 TB Virtual Disk 2
------------------------------------------------------------------------
[root@116Decoder perccli]#
```

4. To turn on Security on the disk group (created out of the 6 SED Capable drives), execute the below command:

```
/opt/MegaRAID/perccli/perccli64 /c0 /d2 set security=on
```

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 /c0 /d2 set security=on
Controller = 0
Status = Success
Description = Success
```

5. Get the Enclosure ID (**EID**) using on the controller '**0**'. In this case, it is '64'

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

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 /c0 /eall show
Controller = 0
Status = Success
Description = None


Properties :
==========

------------------------------------------------------------------------------
EID State Slots PD PS Fans TSs Alms SIM Port#      ProdID    VendorSpecific
------------------------------------------------------------------------------
 64 OK       10 10  0    0   0    0   1 00 & 00 x8 BP14G+EXP  +
------------------------------------------------------------------------------

EID-Enclosure Device ID |PD-Physical drive count |PS-Power Supply count|
TSs-Temperature sensor count |Alms-Alarm count |SIM-SIM Count


[root@116Decoder perccli]#
```

6. To confirm that the drives / Drive Groups (DG) are **SED Enabled** and **Secured**, run the below command and verify the **SED Capable**, **Secured**, **SED Enabled** flags are set as '**Yes**' for drives in slots 4 (**s4**) through 9 (**s9**).

```
/opt/MegaRAID/perccli/perccli64 /c0 /e64/sall show all | egrep -i '
(Policies/Settings |SED Capable|Secured|SED Enabled)'
```

```
Drive /c0/e64/s0 Policies/Settings :
SED Capable = No
SED Enabled = No
Secured = No
Drive /c0/e64/s1 Policies/Settings :
SED Capable = No
SED Enabled = No
Secured = No
Drive /c0/e64/s2 Policies/Settings :
SED Capable = No
SED Enabled = No
Secured = No
Drive /c0/e64/s3 Policies/Settings :
SED Capable = No
SED Enabled = No
```

```
Secured = No
Drive /c0/e64/s4 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s5 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s6 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s7 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s8 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s9 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
```

Appendix B. Encrypt a Series 6E or Series 7 Core or Hybrid Host (`encryptSedVd.py`)

NetWitness 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 12.5 and later. The following directory is for 12.5.

`rsa-sa-tools-12.5.0.0-xxxx.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:

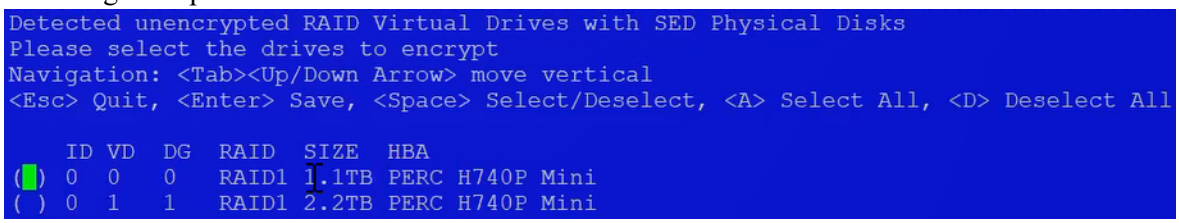
   ```
   OWB_ALLOW_NON_FIPS=1 ./encryptSedVd.py
   ```

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

   - If the drives are encrypted, the script displays the following message.

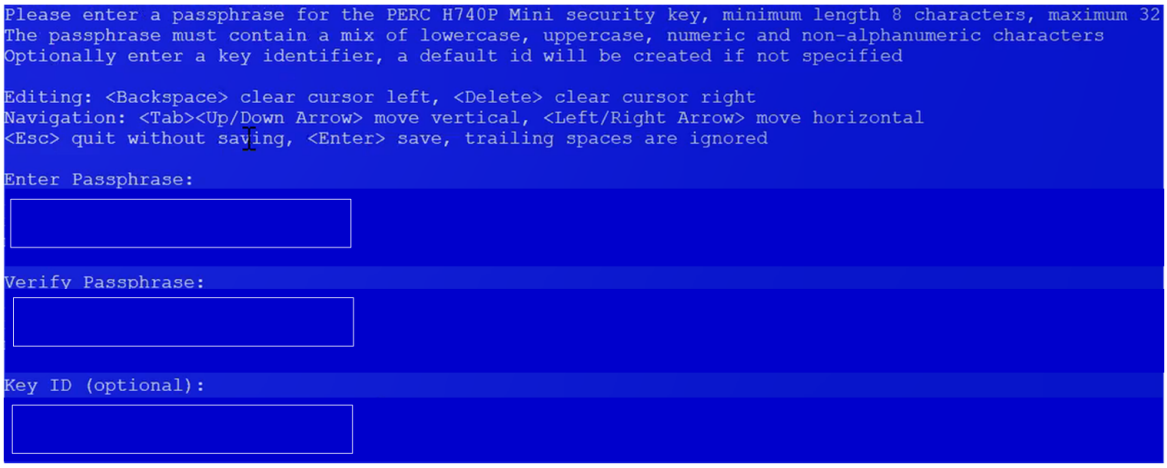
     ```
     No unencrypted RAID virtual drives with SED physical drives found.
     ```

   - If the drives are not encrypted, the script identifies the unencrypted drives as shown in the following example.



4. If the drives are not encrypted and you want to encrypt them:

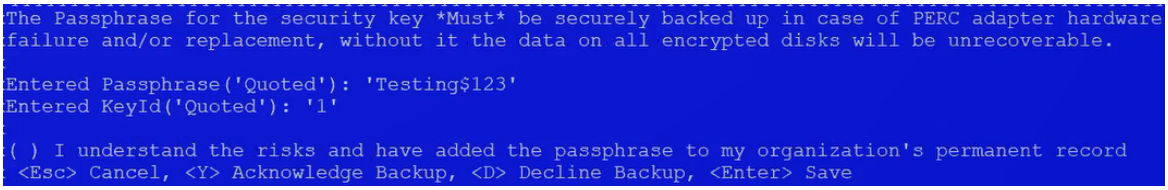
   a. Select the drives you want to encrypt with the space bar and press **Enter**.
   The following prompt is displayed.



   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.



   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
-------------------------------------------------------------------------
```

Note: The SED Enabled and Secured label values are set to Yes, if the drives are SED enabled and secured.
To check the drives on controller 0 and enclosure 247 use the below command:
`/opt/MegaRAID/perccli/perccli64 /c1 /e247/sall show all | egrep -i '(Policies/Settings|SED Capable|Secured|SED Enabled)'`

You will find detailed information on `perccli` commands in the Dell PowerEdge RAID Controller CLI Reference Guide (http://l4u-00.jinr.ru/pub/misc/h-w/LSI/dell-sas-hba-12gbps_reference-guide_en-us.pdf).

Enable SED on configured Drive Groups

Virtual Drives configured are SED Capable but are NOT SED Enabled.

To enable virtual drives or drive groups using PERC H840 Adaptors (External storage):

1. SSH to the appliance and run the below script to encrypt the virtual drive (on external storage).

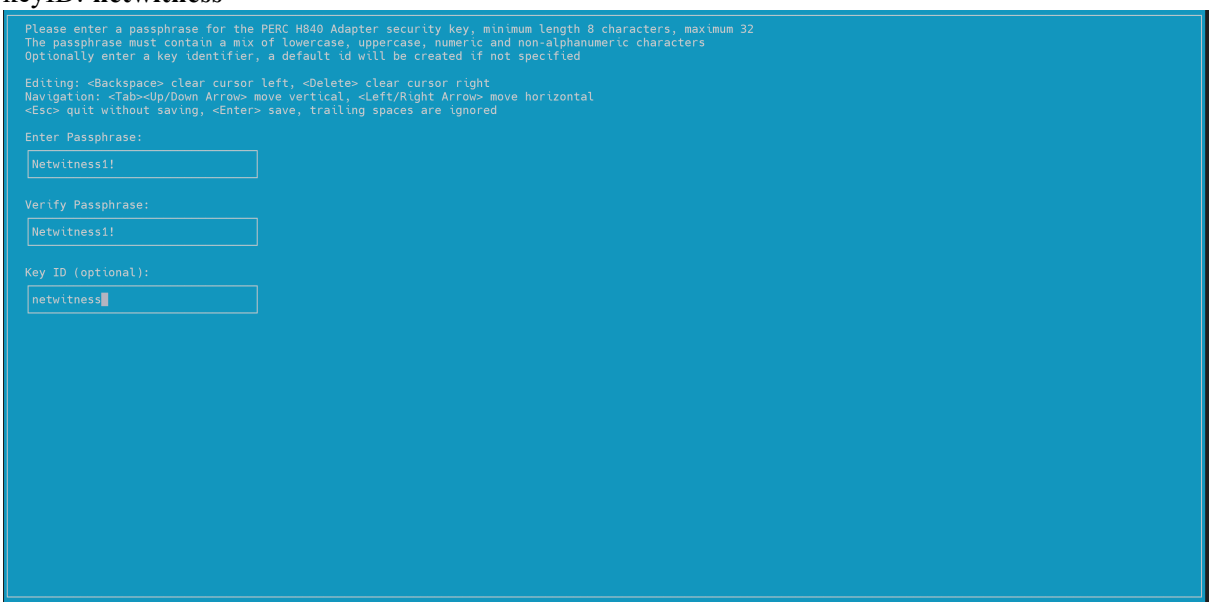
Note: The encryptSedVd.py script turn on the SED feature only on Virtual Drives or Drive Groups on the PERC H840 Adaptors (external storage) and NOT on PERC H740 mini. Refer to Enable Virtual Drives / Drive Groups - PERC H740 (Mini) Adaptors (Internal storage) to enable SED on PERC H740 Mini .

```
OWB_ALLOW_NON_FIPS=true /opt/rsa/saTools/supportScript/encryptSedVd.py
```

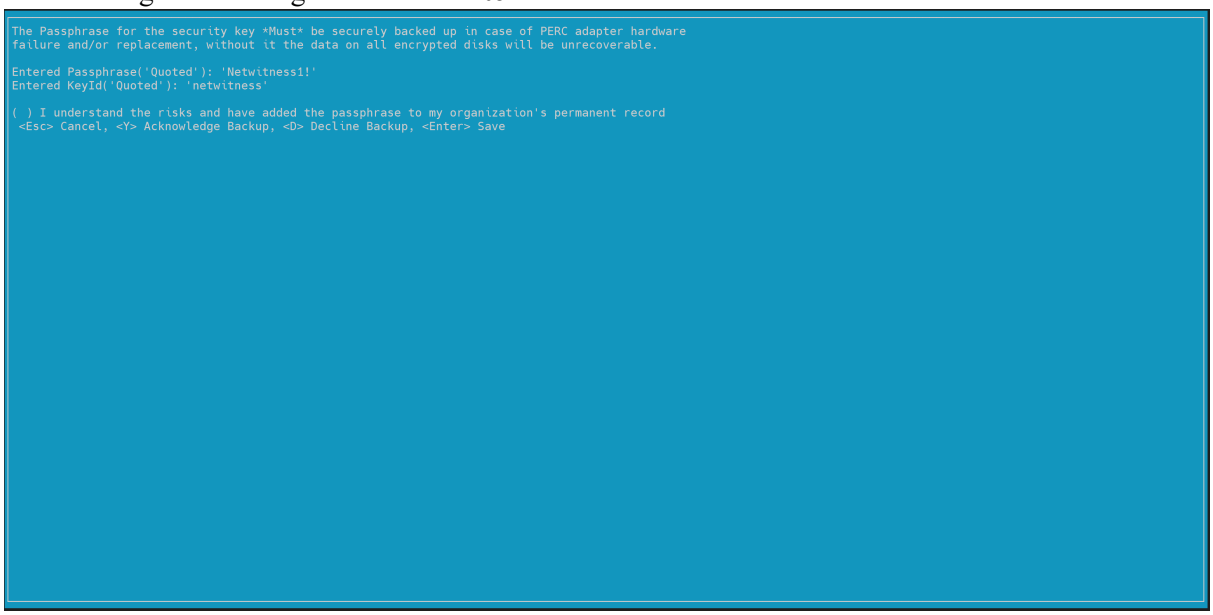
2. Select the Virtual Drive and press **Enter**.
   Passphrase screen is displayed.
3. Enter the Passphrase and press **Enter**.
   For Example,

   Passphrase : **Netwitness1!**

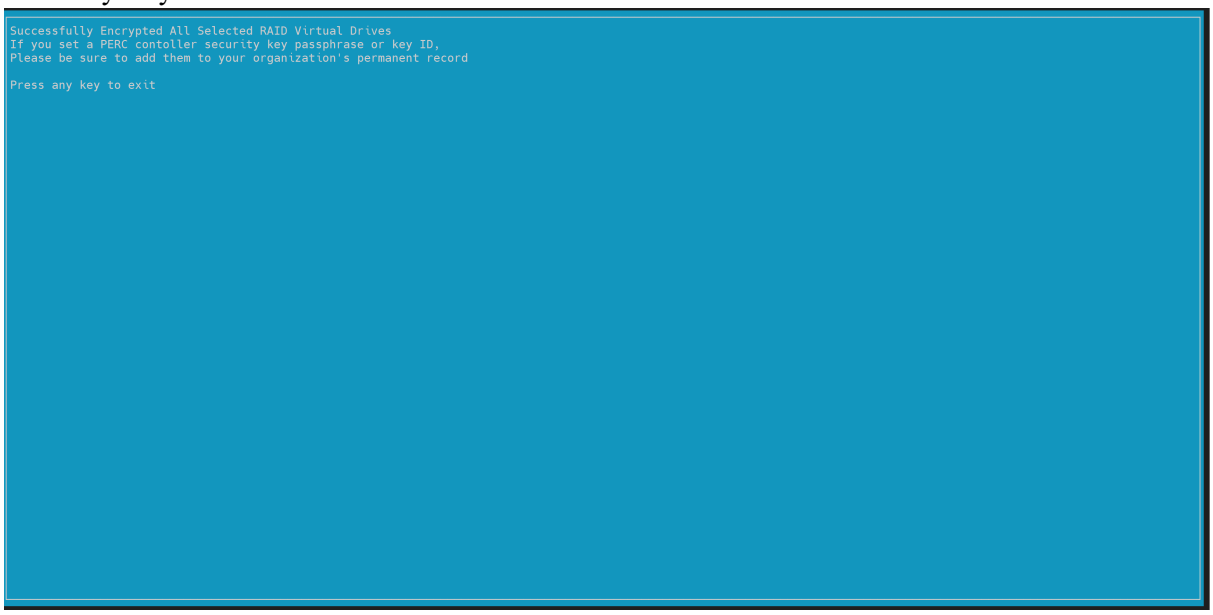
   keyID: **netwitness**



4. Acknowledge the message and Press **Enter** to Save.



5. Press any Key to Exit.



6. To confirm that the drives are SED Enabled and secured, run the following command and verify the SED Enabled and Secured returns Yes.

```
/opt/MegaRAID/perccli/perccli64 /c1 /e247/sall show all | egrep -i '
(Policies/Settings|SED Capable|Secured|SED Enabled)'
Drive /c1/e247/s0 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s1 Policies/Settings :
SED Capable = Yes
```

```
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s2 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s3 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s4 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s5 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s6 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s7 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s8 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s9 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s10 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s11 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
```

```
Secured = Yes
```

Enable Virtual Drives / Drive Groups - PERC H740 (Mini) Adaptors (Internal storage)

You can enable the SED capability on the Virtual Drive or Drive Groups created out of on-board SED capable drives (in slots 4 through 9 – total of 6 drives) using the percli64 utility. You cannot use `/opt/rsa/saTools/supportScript/encryptSedVd.py` to turn on Security on the Virtual drives on the PERC H740 (mini) Adaptor.

1. SSH to the appliance and enable security on the PERC H740 (mini) Adaptor. The controller number for this adaptor is **0**. The PERC H840 Adaptor is shown as **1**.

To list all the controllers on the appliance run the following command:

```
/opt/MegaRAID/perccli/perccli64 show | egrep -A3 'Model'
```

The first column (**Ctl**) lists out the controller index on the appliance. In this case, the controller **0** corresponds to **PERC H740 Mini** and controller **1** corresponds to **PERC H840 Adaptor**. The columns **DGs** and **VDs** dislpays the virtual drives and drive groups on the controller.

2. To enable the security on the **PERC H740 (mini) Adaptor**, for example, Controller **0**, run the following command:
   ```
   /opt/MegaRAID/perccli/perccli64 /c0 set securitykey='<String>'!'
   keyid='<String>'
   ```

   Example:

   ```
   /opt/MegaRAID/perccli/perccli64 /c0 set securitykey='Netwitness1!'
   keyid='netwitness'
   ```

   ```
   'Netwitness1' is the securityKey and 'netwintess' is ID.
   ```

   Make a note of both the Key and keyID securely.

3. Identify the correct Drive group (DG) or Virtual Drive (VD) corresponding to the SED Capable drives that you want to enable security.
   ```
   /opt/MegaRAID/perccli/perccli64 /c0 /vall show | egrep -A5 'DG/VD'
   ```

   Check the first two and last column to identify the correct Drive Group / Virtual Drive correspond to the 6 SED enabled drives that are SED Capable. On Series 6 appliances, there is only one DG or VD with **RAID6** type. Name column can be used to identify the VD or DG. In this case, the DG or VD is **2**. Using a combination of **Type**, **Name** and **Size** columns (these are defined when you created VDs above).

4. To turn on Security on the disk group (created out of the 6 SED Capable drives) for **decodersmall** volume group, run the below command:
   ```
   /opt/MegaRAID/perccli/perccli64 /c0 /d2 set security=on
   ```
5. Get the Enclosure ID (**EID**) using on the controller **0**. In this case, it is **64**
   ```
   /opt/MegaRAID/perccli/perccli64 /c0 /eall show
   ```

6. To confirm that the drives or drive groups are **SED Enabled** and **Secured**, run the below command and verify the **SED Capable**, **Secured**, **SED Enabled** flags are set as **Yes** for drives in slots 4 (s4) through 9 (s9).

```
/opt/MegaRAID/perccli/perccli64 /c0 /e64/sall show all | egrep -i '
(Policies/Settings |SED Capable|Secured|SED Enabled)'
```

Drive /c0/e64/s0 Policies/Settings :

```
SED Capable = No
SED Enabled = No
Secured = No
Drive /c0/e64/s1 Policies/Settings :
SED Capable = No
SED Enabled = No
Secured = No
Drive /c0/e64/s2 Policies/Settings :
SED Capable = No
SED Enabled = No
Secured = No
Drive /c0/e64/s3 Policies/Settings :
SED Capable = No
SED Enabled = No
Secured = No
Drive /c0/e64/s4 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s5 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s6 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s7 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s8 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s9 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
```

```
Secured = Yes
```

Enable SED on configured Virtual Drives/ Drive Groups on Power Vault (PERC 840)

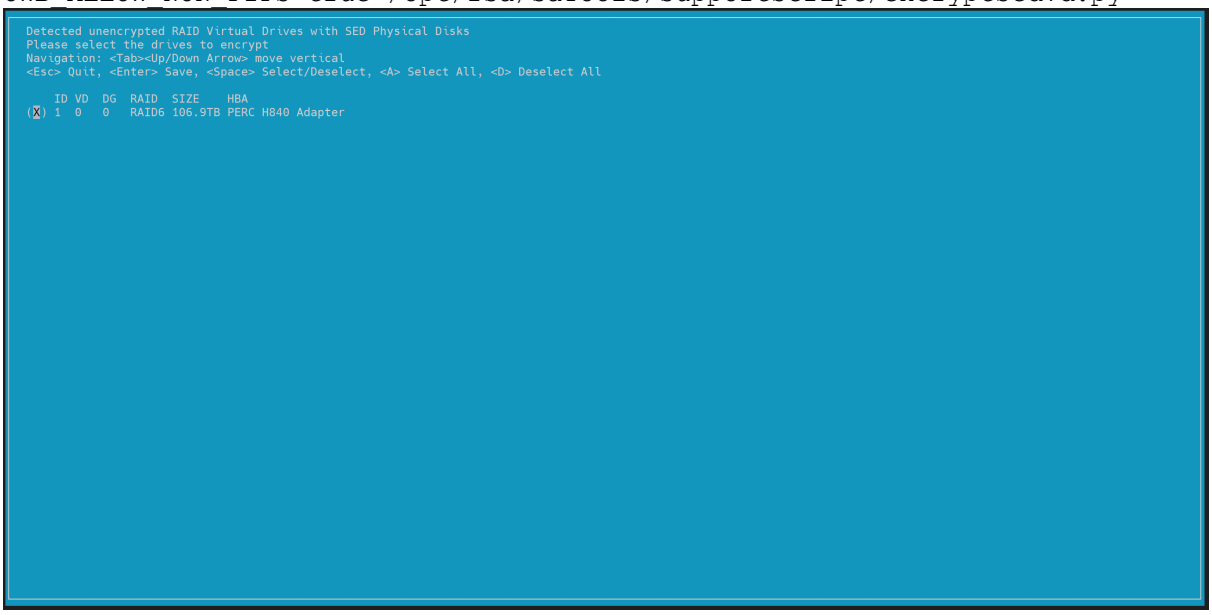
Enable Virtual Drives / Drive Groups - PERC H840 Adaptors

Note: The virtual disk created in *Configure Block Devices for PowerVaults* section in Configure Drive Pack(s) is SED capable but NOT SED Enabled.

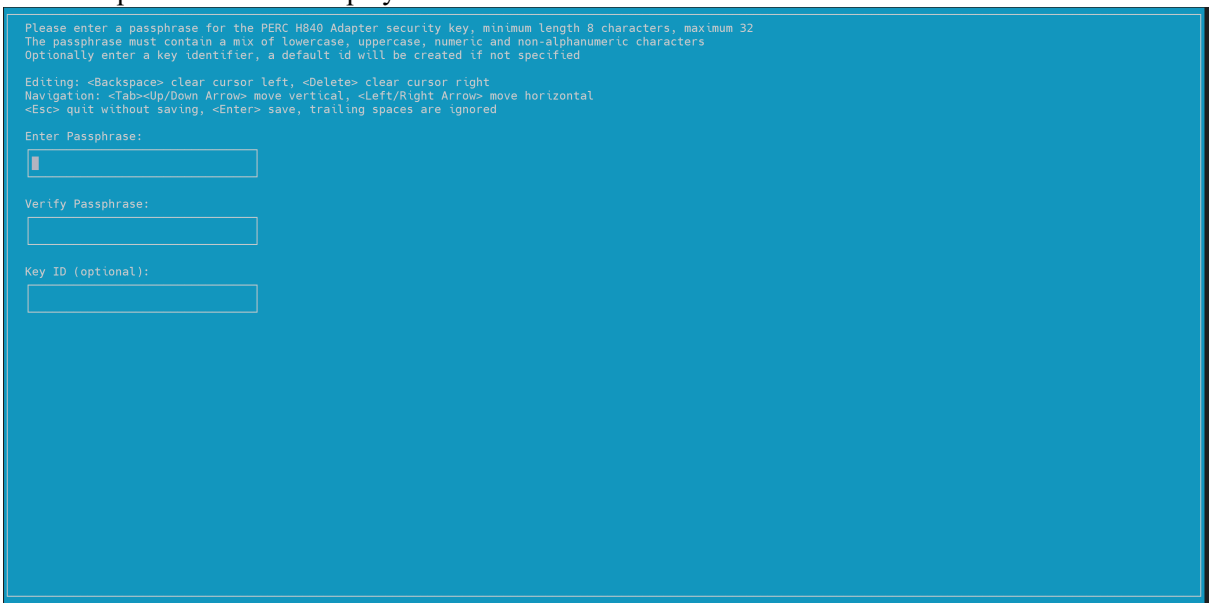
1. To enable, SSH into the appliance and run the below script to encrypt the VD (on external storage).
   `OWB_ALLOW_NON_FIPS=true /opt/rsa/saTools/supportScript/encryptSedVd.py`

   Note: The `encryptSedVd.py` script turn on the SED feature only on Virtual Drives or Drive Groups on the PERC H840 Adaptors (external storage) and NOT on PERC H740 mini. Refer to **Enable Virtual Drives / Drive Groups - PERC H740 (Mini) Adaptors (Internal storage)** to enable SED on PERC H740 Mini

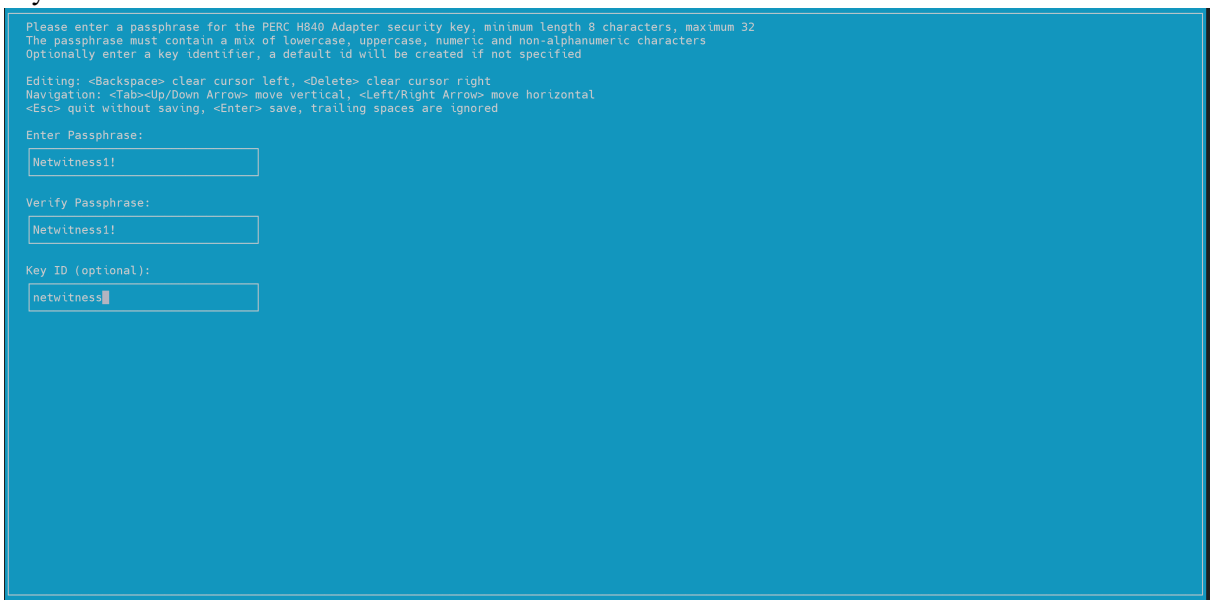
   `OWB_ALLOW_NON_FIPS=true /opt/rsa/saTools/supportScript/encryptSedVd.py`



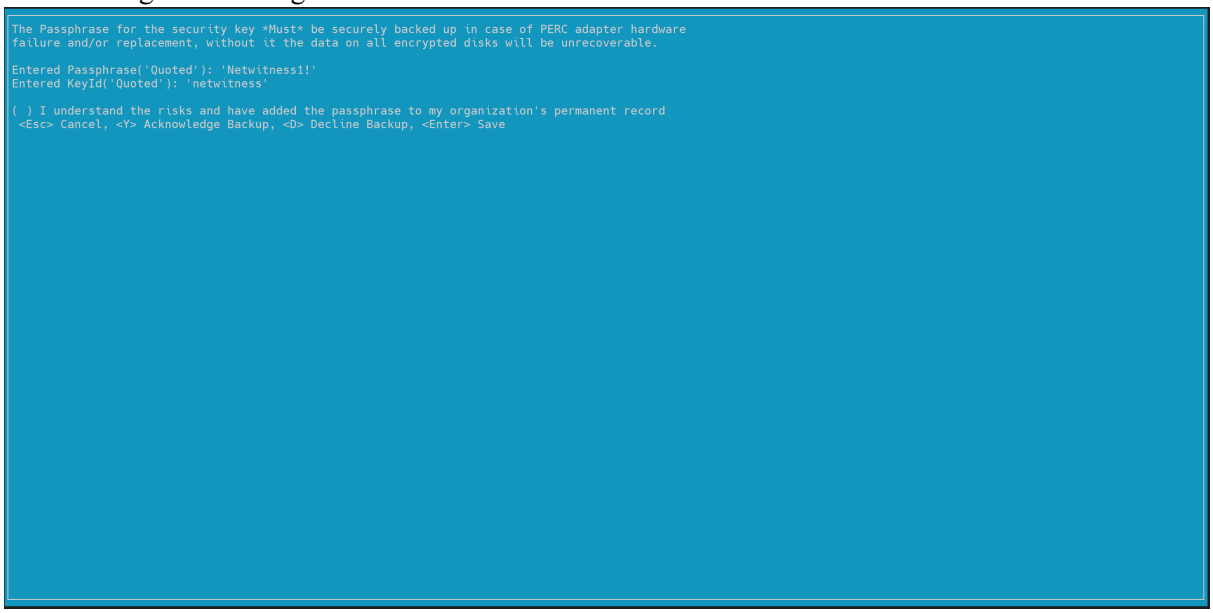
2. Select both the **Virtula Disks** and press **Enter**.
   The Passphrase screen is displayed.



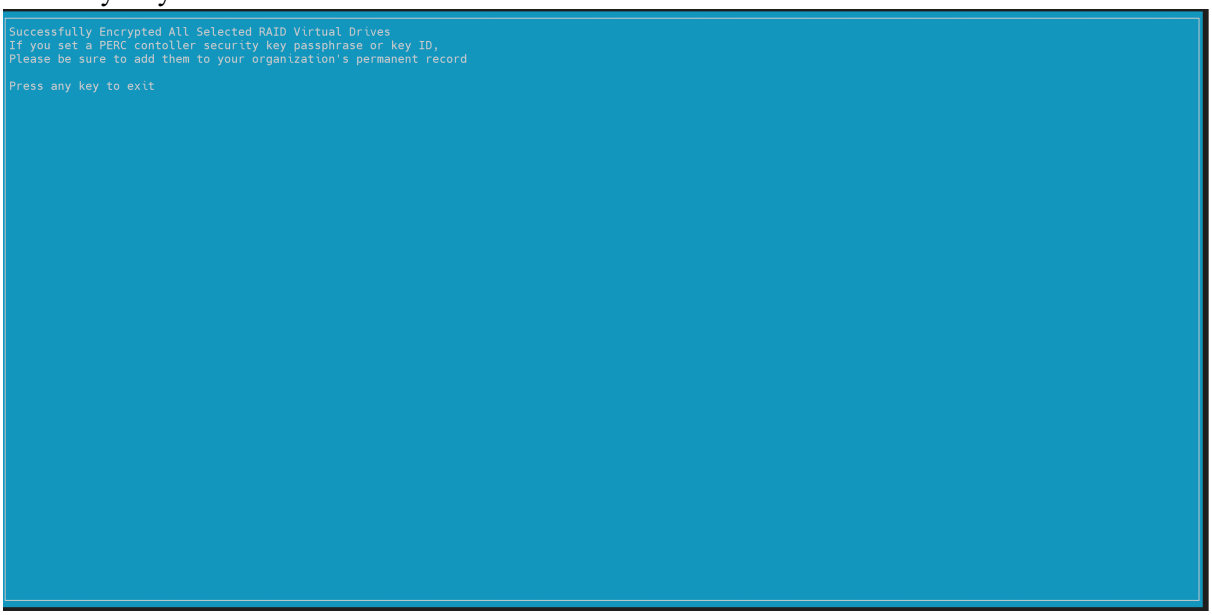
3. Enter the Passphrase and press **Enter**.
   For example,
   Passphrase : **Netwitness1!**
   keyID: **netwitness**



4. Acknowledge the message and Press **Enter** to Save.



5. Press any Key to Exit.



6. To confirm that the drives are SED Enabled and secured, run the below command and verify the SED Enabled and Secured returns Yes.
   `/opt/MegaRAID/perccli/perccli64 /c1 /e247/sall show all | egrep -i '(Policies/Settings|SED Capable|Secured|SED Enabled)'`
   Drive /c1/e247/s0 Policies/Settings :

```
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s1 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s2 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s3 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s4 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
```

```
Secured = Yes
Drive /c1/e247/s5 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s6 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s7 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s8 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s9 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s10 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c1/e247/s11 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
```

Enable Security on SED Capable Drive groups on Host with a mix of SED and NON SED Drives

The encryptSedVd.py may fail to identify the SED Capable Virtual Drives when there is mix of both SED and NON-SED drives on the appliance. The below steps are applicable when both SED and NON-SED capable virtual drives exist on the host.

1. SSH to the appliance and enable security on the PERC H740 (mini) Adaptor. The controller number for this adaptor is **0**. The PERC H840 Adaptor is shown as **1**.
   To list all the controllers on the appliance:
   `/opt/MegaRAID/perccli/perccli64 show | egrep -A3 'Model'`
   The first column **(Ctl)** lists out the controller index on the appliance. In this case, the controller '**0**' corresponds to '**PERC H740 Mini**' and controller '**1**' corresponds to '**PERC H840 Adaptor**' . The columns '**DGs**' and '**VDs**' displays the drive groups and virtual drives on the controller.

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 show | egrep -A3 'Model'
Ctl Model          Ports PDs DGs DNOpt VDs VNOpt BBU sPR DS EHS ASOs Hlth
------------------------------------------------------------------------
  0 PERCH740PMini      8  10   3     0   3     0 Opt On  -  N      0 Opt
  1 PERCH840Adapter    8  12   1     0   1     0 Opt On  -  N      0 Opt
[root@116Decoder perccli]#
```

2. To enable the security on the 'PERC H740 (mini) Adaptor' i.e Controller '**0**', execute the following command:
   `/opt/MegaRAID/perccli/perccli64 /c0 set securitykey='<SOME_STRING_VALUE>'!' keyid='< SOME_STRING_VALUE >'`

   Example:
   `/opt/MegaRAID/perccli/perccli64 /c0 set securitykey='Netwitness1!' keyid=1`

   'Netwitness1' is the securityKey and '1' is ID. Preserve both the Key and keyID securely.

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 /c0 set securitykey='Netwitness1!' keyid='netwitness'
Controller = 0
Status = Success
Description = None


Controller Properties :
=====================

---------------------
Ctrl Method  Result
---------------------
   0 set Key Success
---------------------
```

3. Identify the correct Drive group (DG) / Virtual Drive (VD) corresponding to the SED Capable drives that we are trying to enable security.
   `/opt/MegaRAID/perccli/perccli64 /c0 /vall show | egrep -A5 'DG/VD'`

   Refer to first two and last column to identify the correct Drive Group (DG) / Virtual Drive (VD) correspond to the 6 SED enabled drives. On Series 6 appliances, there is only one DG/VD with **RAID6**. '**Name**' column can be used to identify the VD/DG. In this case, the DG/VD is '**2**'. Using a combination of '**Type**', '**Name**' and '**Size**' columns (these were defined by the user when the VDs are created above).

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 /c0 /vall show | egrep -A5 'DG/VD'
DG/VD TYPE  State Access Consist Cache Cac sCC     Size Name
-------------------------------------------------------------------------
0/0   RAID1 Optl  RW     Yes     RWBD  -   OFF 931.0 GB
1/1   RAID1 Optl  RW     Yes     RWBD  -   OFF 1.818 TB
2/2   RAID6 Optl  RW     Yes     RWBD  -   OFF 8.730 TB Virtual Disk 2
-------------------------------------------------------------------------
[root@116Decoder perccli]#
```

4. To turn on Security on the disk group (created out of the 6 SED Capable drives), execute the below command:

```
/opt/MegaRAID/perccli/perccli64 /c0 /d2 set security=on
```

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 /c0 /d2 set security=on
Controller = 0
Status = Success
Description = Success
```

5. Get the Enclosure ID (**EID**) using on the controller '**0**'. In this case, it is '64'

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

```
[root@116Decoder perccli]# /opt/MegaRAID/perccli/perccli64 /c0 /eall show
Controller = 0
Status = Success
Description = None


Properties :
==========

------------------------------------------------------------------------------
EID State Slots PD PS Fans TSs Alms SIM Port#      ProdID    VendorSpecific
------------------------------------------------------------------------------
 64 OK       10 10  0    0   0    0   1 00 & 00 x8 BP14G+EXP  +
------------------------------------------------------------------------------

EID-Enclosure Device ID |PD-Physical drive count |PS-Power Supply count|
TSs-Temperature sensor count |Alms-Alarm count |SIM-SIM Count


[root@116Decoder perccli]#
```

6. To confirm that the drives / Drive Groups (DG) are **SED Enabled** and **Secured**, run the below command and verify the **SED Capable**, **Secured**, **SED Enabled** flags are set as '**Yes**' for drives in slots 4 (**s4**) through 9 (**s9**).

```
/opt/MegaRAID/perccli/perccli64 /c0 /e64/sall show all | egrep -i '
(Policies/Settings |SED Capable|Secured|SED Enabled)'

Drive /c0/e64/s0 Policies/Settings :

SED Capable = No

SED Enabled = No

Secured = No

Drive /c0/e64/s1 Policies/Settings :

SED Capable = No

SED Enabled = No

Secured = No

Drive /c0/e64/s2 Policies/Settings :

SED Capable = No

SED Enabled = No

Secured = No
```

```
Drive /c0/e64/s3 Policies/Settings :
SED Capable = No
SED Enabled = No
Secured = No
Drive /c0/e64/s4 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s5 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s6 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s7 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s8 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
Drive /c0/e64/s9 Policies/Settings :
SED Capable = Yes
SED Enabled = Yes
Secured = Yes
```

Note: The steps for enabling encryption on Series 7 hardware (with or without external storage configuration i.e MD2412) are like Series 6 steps described in this section. On Series 7 hardware:
1. The internal HBA is PERC H965i and external HBA is PERC H965e.
2. Perccli64 is not supported on Series 7 hardware. Use */opt/MegaRAID/perccli/perccli2*. Substitute perccli64 with perccli2 when executing the perccli commands on Series 7 hardware.

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

Th 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 for 15-Drive DACs

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

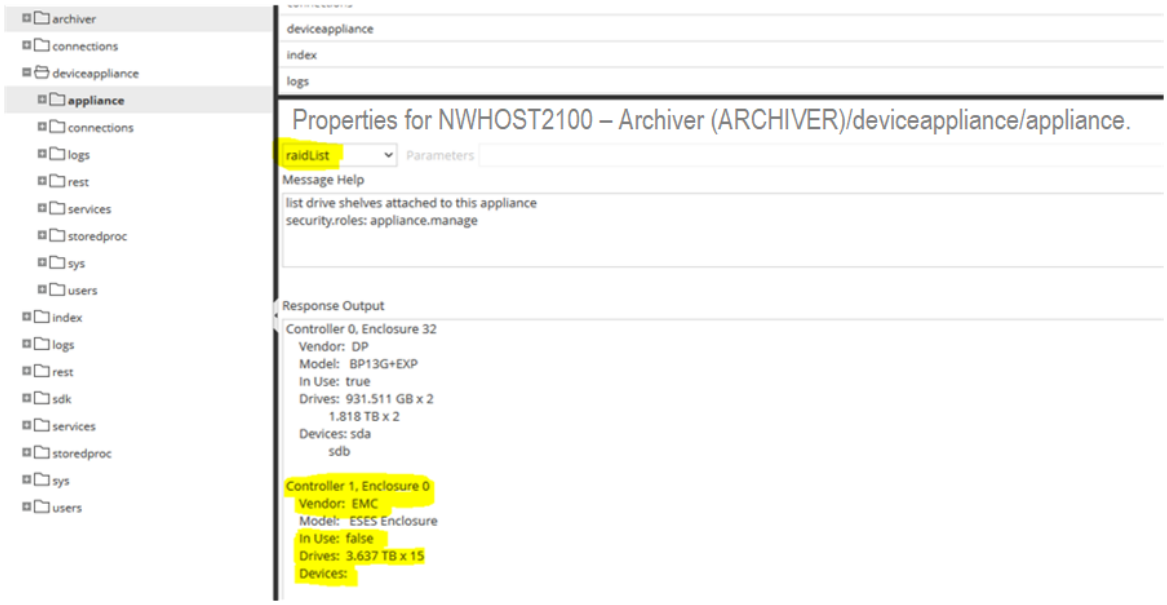
Configure Storage for Archiver

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>
   ```

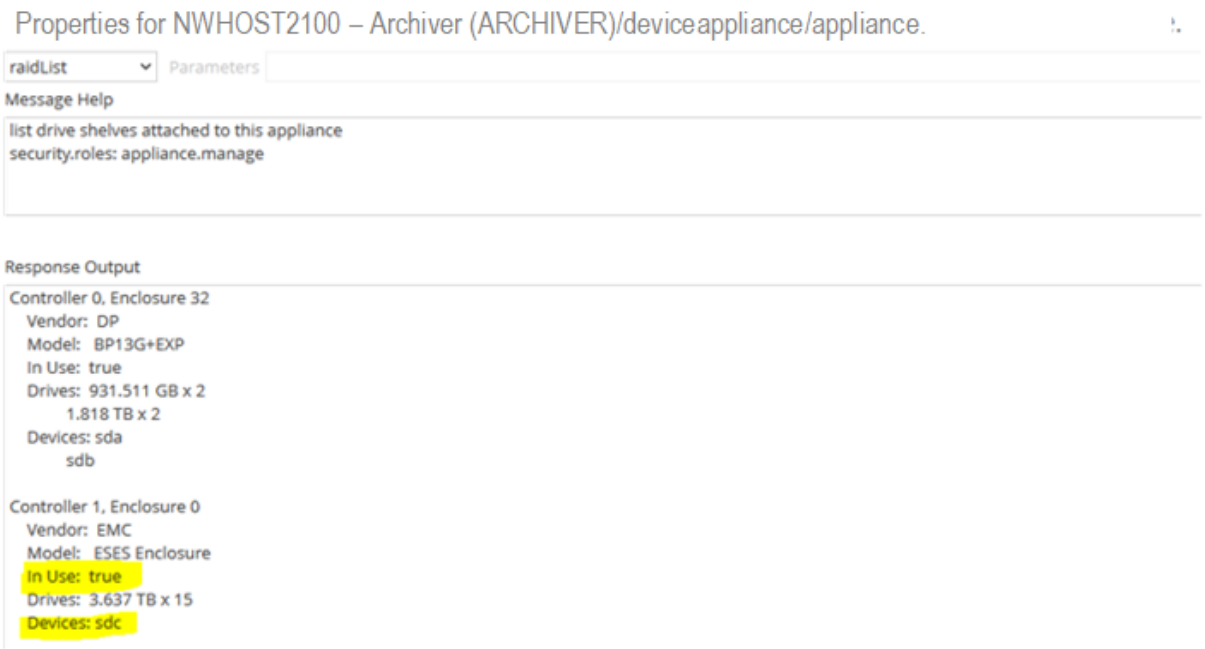
   b. Verify the Drive Count, Size, and Vendor.
   The following example illustrates what you should see before you create a RAID array.



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
   ```

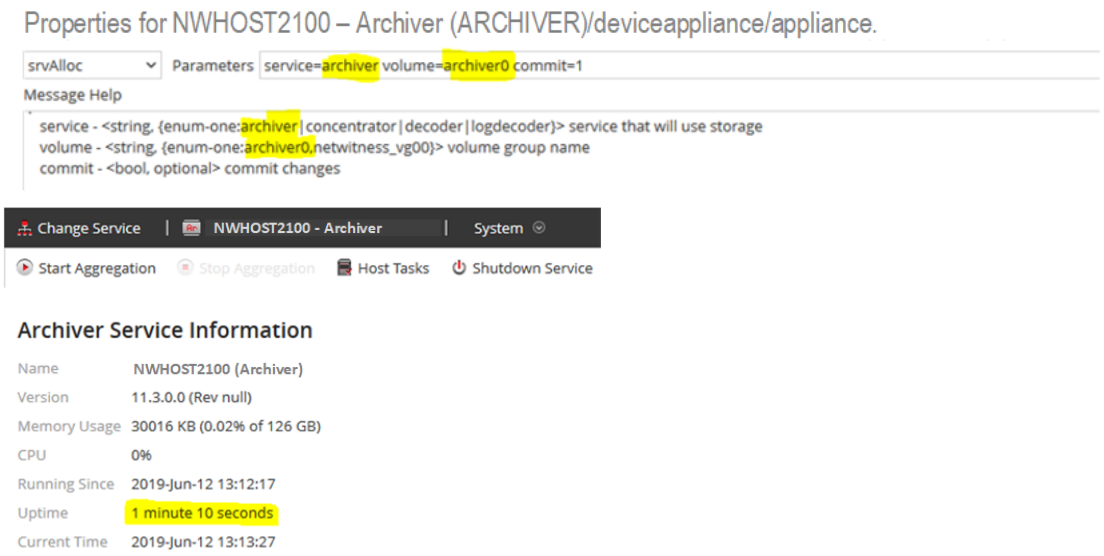
The following example illustrates what you should see after you create a RAID array.

3. 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`)

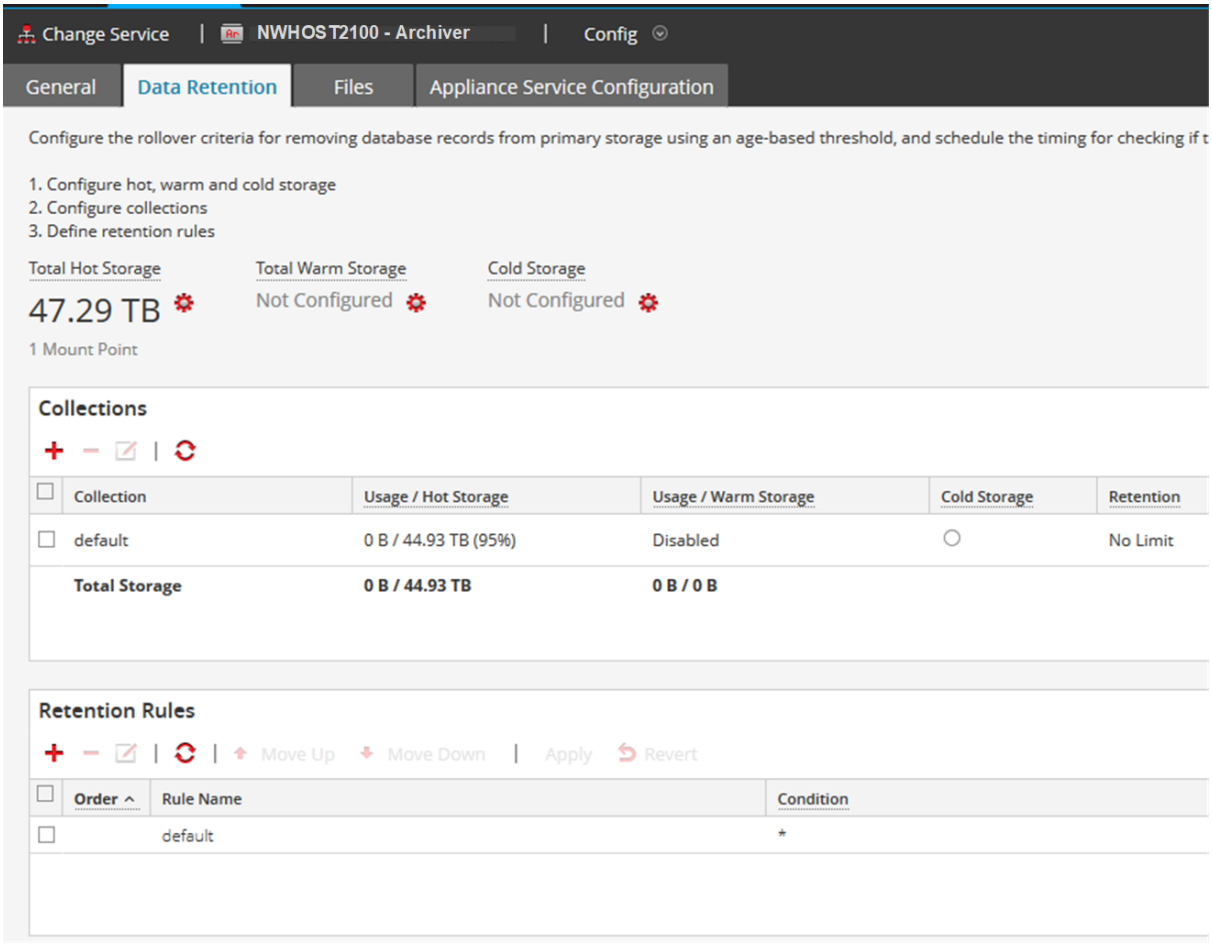


4. Execute the `partNew` command with the following parameters to create partitions and mount points in the `etc/fstab` file.
   `name=<device> (for example, sdc) service=archiver volume=archiver commit=1`

5. 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`



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



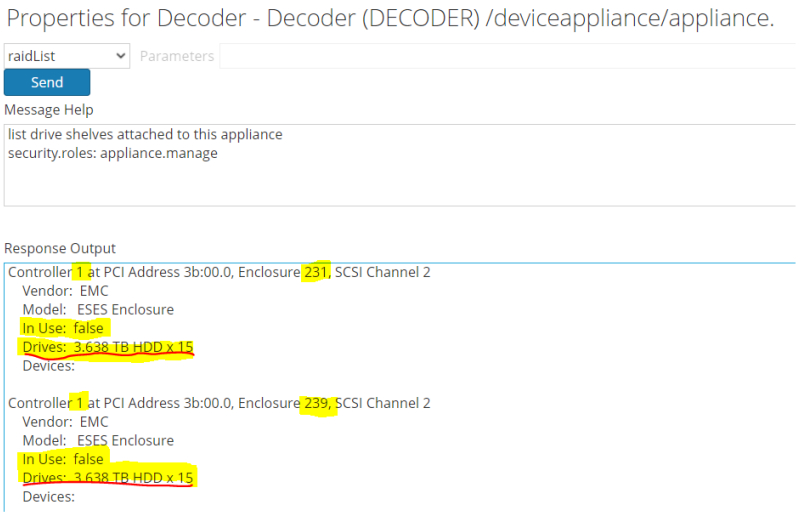
Configure Storage for Network (Packet) Decoder

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.

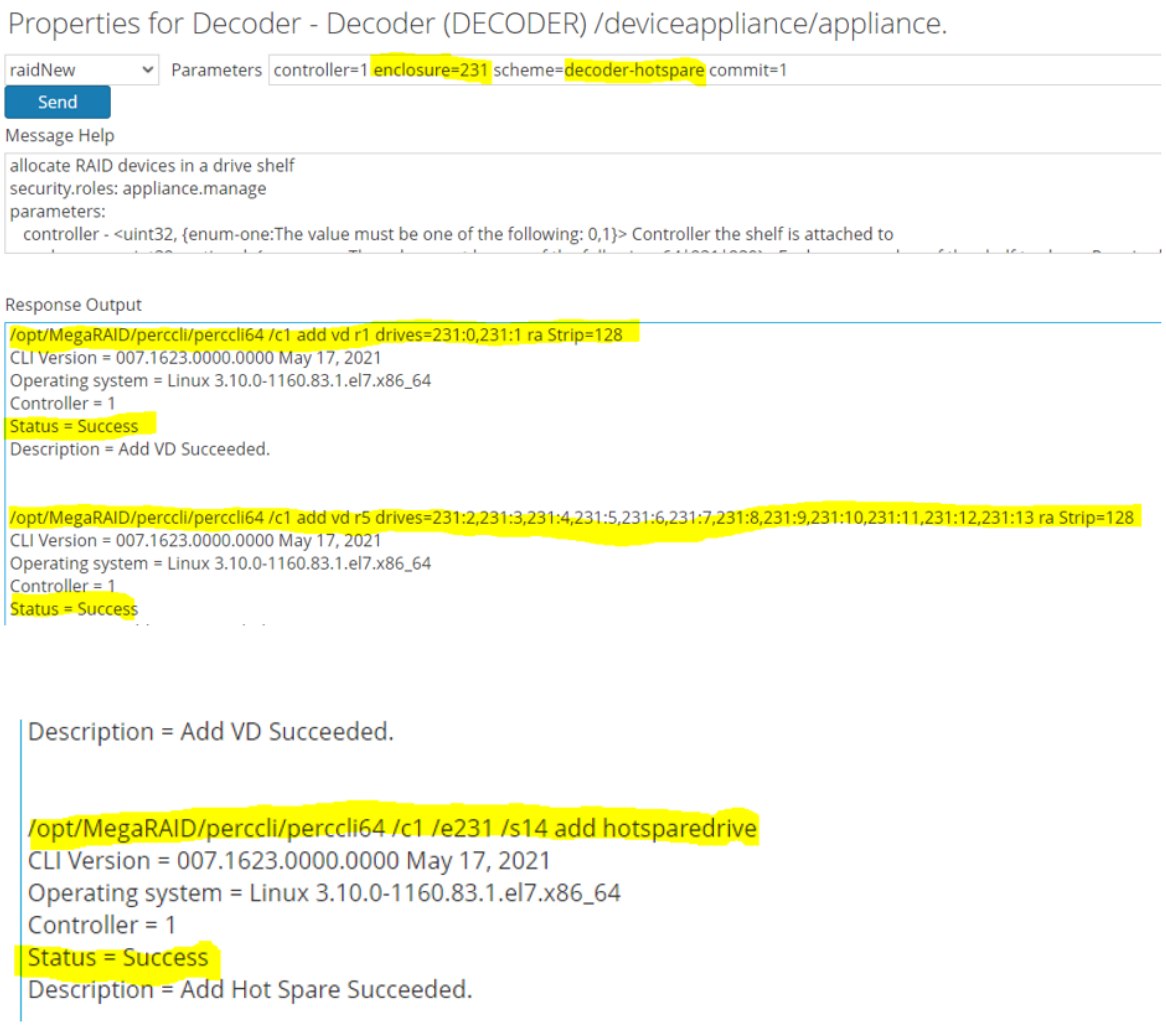
   a. Record the Controller Number, Enclosure Number, In Use, Drives, and Devices.
   You should see the following information.
   ```
   In Use: FALSE
   Devices: <empty>
   ```

   b. Verify the Drive Count, Size, and Vendor.
   The following example illustrates what you should see before you create a RAID array.



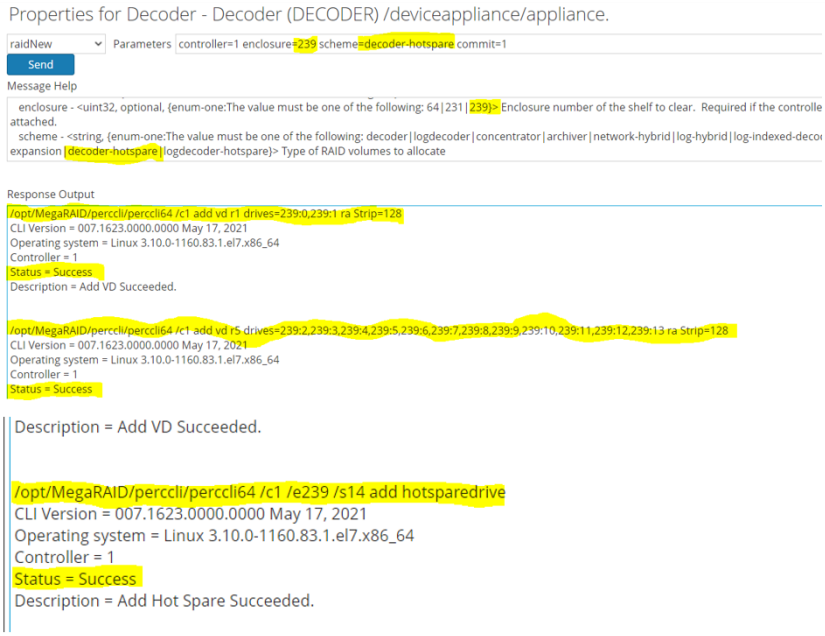
2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.

   - Parameters for the first enclosure:
     ```
     controller=1 enclosure=231 scheme=decoder-hotspare commit=1
     ```

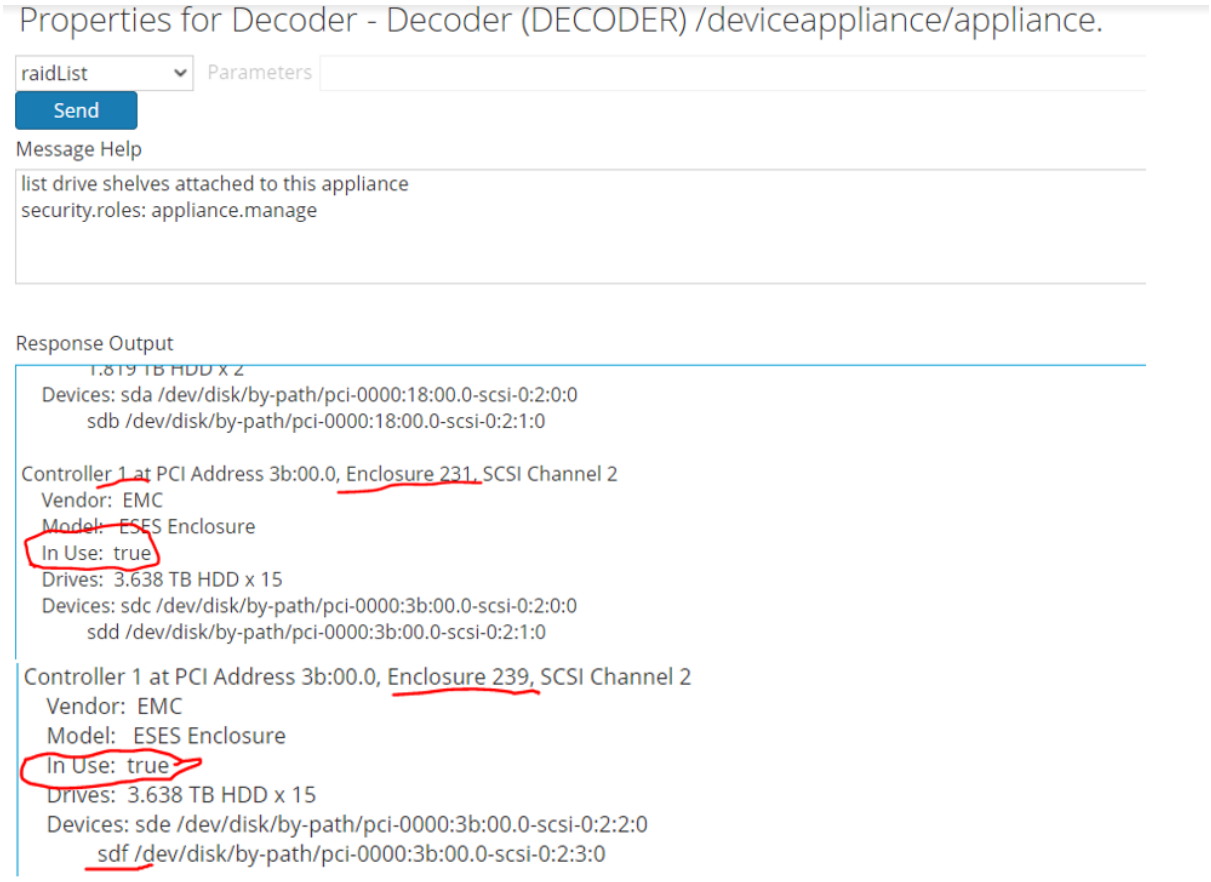

Properties for Decoder - Decoder (DECODER) /deviceappliance/appliance.
raidNew
Parameters
controller=1 enclosure=231 scheme=decoder-hotspare commit=1
Send
Message Help
allocate RAID devices in a drive shelf
security.roles: appliance.manage
parameters:
controller - <uint32, {enum-one:The value must be one of the following: 0,1}> Controller the shelf is attached to
Response Output
/opt/MegaRAID/perccli/perccli64 /c1 add vd r1 drives=231:0,231:1 ra Strip=128
CLI Version = 007.1623.0000.0000 May 17, 2021
Operating system = Linux 3.10.0-1160.83.1.el7.x86_64
Controller = 1
Status = Success
Description = Add VD Succeeded.
/opt/MegaRAID/perccli/perccli64 /c1 add vd r5 drives=231:2,231:3,231:4,231:5,231:6,231:7,231:8,231:9,231:10,231:11,231:12,231:13 ra Strip=128
CLI Version = 007.1623.0000.0000 May 17, 2021
Operating system = Linux 3.10.0-1160.83.1.el7.x86_64
Controller = 1
Status = Success

Description = Add VD Succeeded.
/opt/MegaRAID/perccli/perccli64 /c1 /e231 /s14 add hotsparedrive
CLI Version = 007.1623.0000.0000 May 17, 2021
Operating system = Linux 3.10.0-1160.83.1.el7.x86_64
Controller = 1
Status = Success
Description = Add Hot Spare Succeeded.

- Parameters for the second enclosure:
  `controller=1 enclosure=239 scheme=decoder-hotspare commit=1`



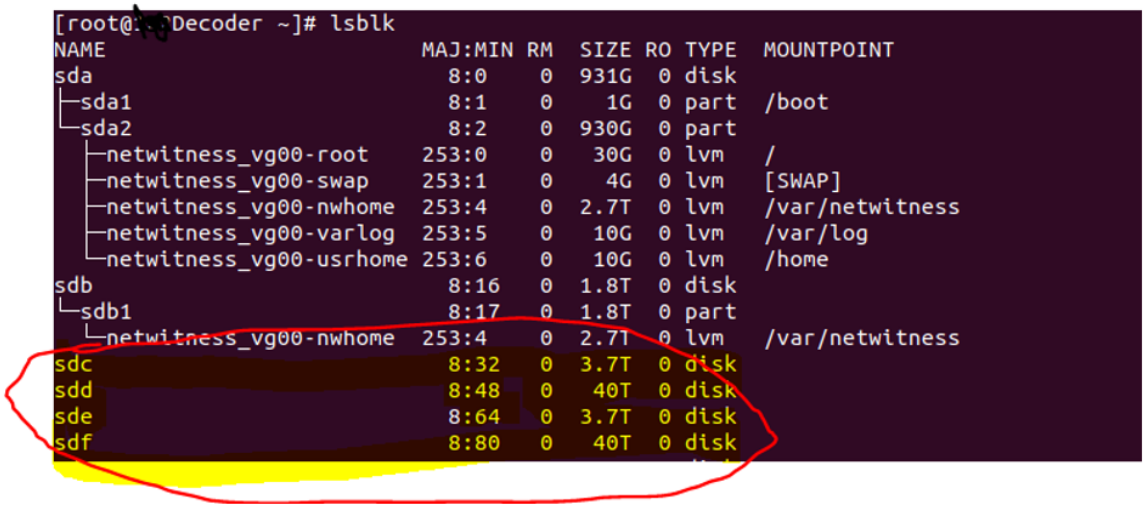
3. Use the `raidList` command to display block devices for enclosures so you can verify `In Use: TRUE`.



4. Use devlist to view the new block devices and their sizes. The new devices are highlighted in red.

5. SSH to the Network Decoder and use the **`lsblk`** command to confirm the block device sizes. The smaller block devices are always allocated to decodersmall and larger devices are allocated to

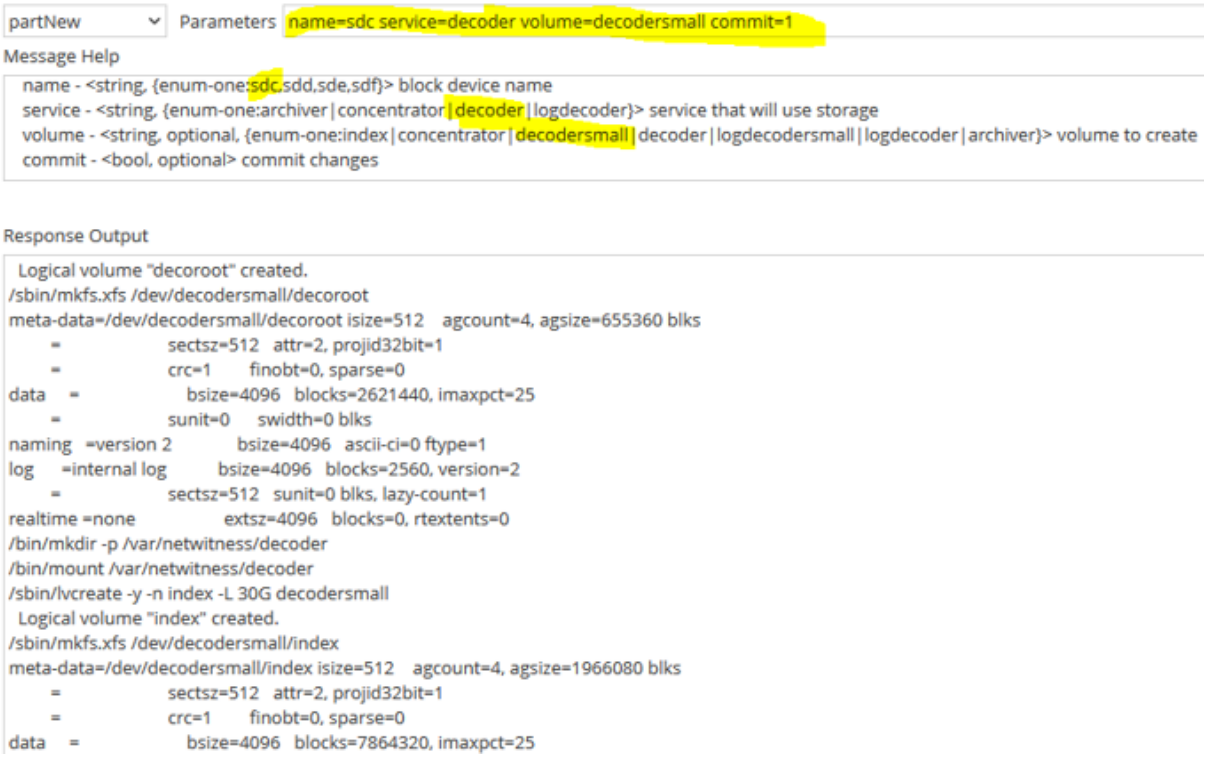
decoder volume.



Note: When configuring for 10g capture, use decoder-hotspare for both the enclosures for performance reasons. For non 10g captures, use decoder-hotspare for the first enclosure and packet-expansion for the second enclosure.

6. Execute the `partNew` command to create the **decodersmall** partition first (decoder dir, index, metadb, sessiondb) (First Enclosure, SDC, SDD) with the following parameters.

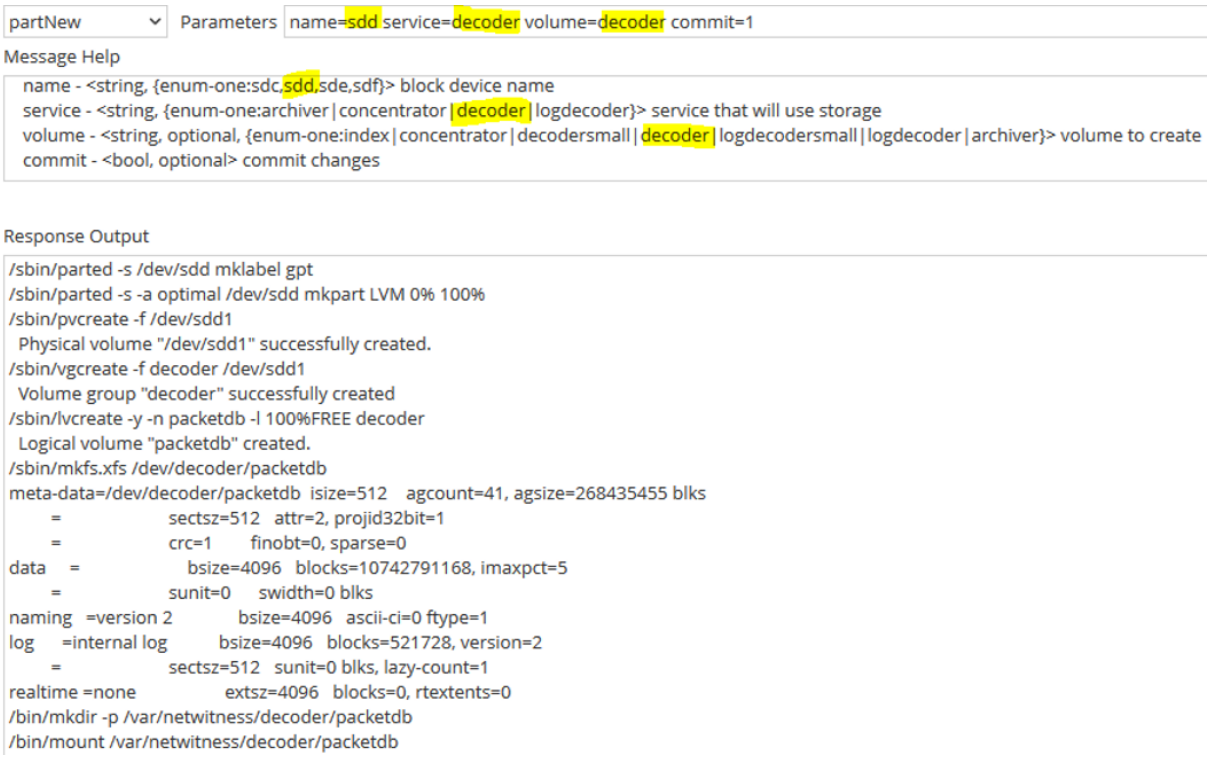
   `name=sdc service=decoder volume=decodersmall commit=1`



```
[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/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
[root@NWHOST2000 ~]#
```

7. 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`



```
[root@[redacted]Decoder ~]# df -hP
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                             63G     0   63G   0% /dev
tmpfs                                63G   60K   63G   1% /dev/shm
tmpfs                                63G   11M   63G   1% /run
tmpfs                                63G     0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-root     30G  3.9G   27G  13% /
/dev/sda1                          1014M   89M  926M   9% /boot
/dev/mapper/netwitness_vg00-varlog   10G   48M   10G   1% /var/log
/dev/mapper/netwitness_vg00-nwhome  2.7T  609M  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-usrhome  10G   33M   10G   1% /home
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   34M  600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decodersmall-metadb     3.1T   34M  3.1T   1% /var/netwitness/decoder/metadb
/dev/mapper/decoder-packetdb         41T   34M   41T   1% /var/netwitness/decoder/packetdb
```

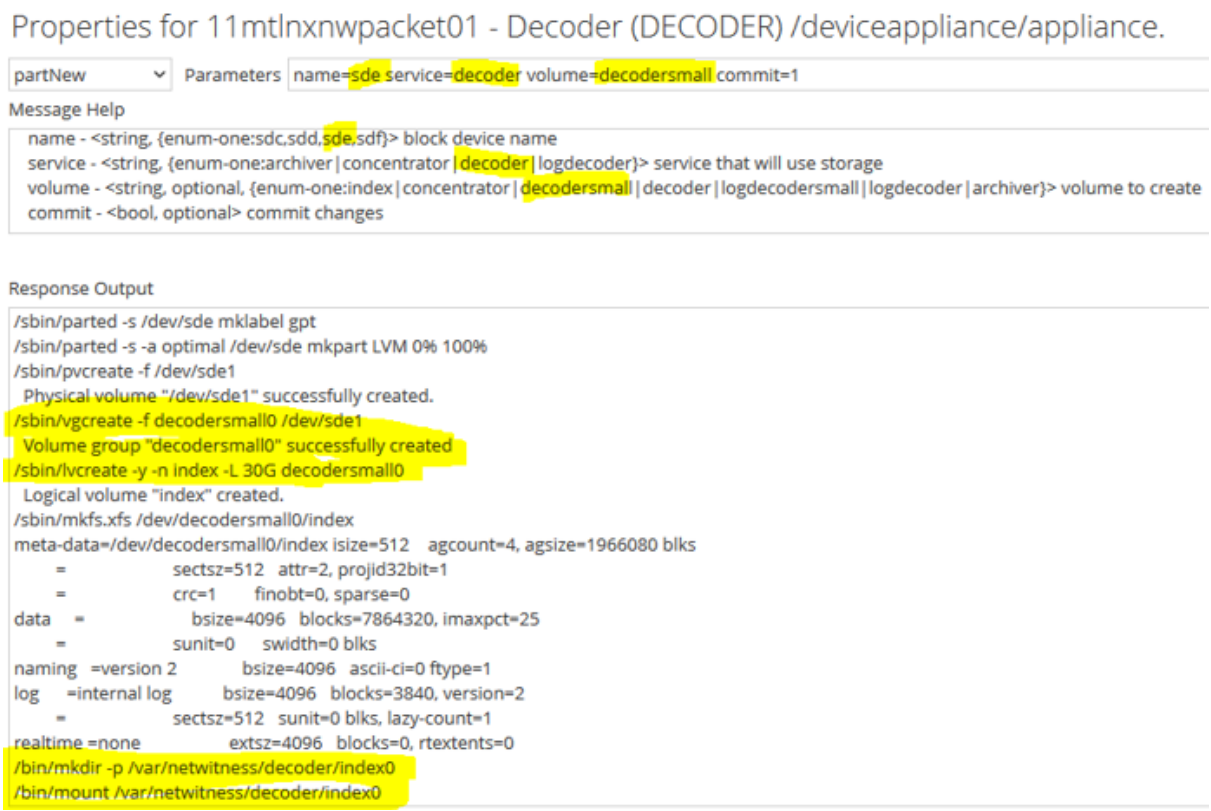
In this example, the below partitions (highlighted in Yellow) are created on sdc and sdd (Enclosure 231).

```
[root@[redacted]Decoder ~]# 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  253:4    0  2.7T  0 lvm   /var/netwitness
  ├─netwitness_vg00-varlog  253:5    0   10G  0 lvm   /var/log
  └─netwitness_vg00-usrhome 253:6    0   10G  0 lvm   /home
sdb                           8:16   0  1.8T  0 disk
└─sdb1                        8:17   0  1.8T  0 part
  └─netwitness_vg00-nwhome  253:4    0  2.7T  0 lvm   /var/netwitness
sdc                           8:32   0  3.7T  0 disk
└─sdc1                        8:33   0  3.7T  0 part
  ├─decodersmall-decoroot   253:7    0   10G  0 lvm   /var/netwitness/decoder
  ├─decodersmall-index      253:8    0   30G  0 lvm   /var/netwitness/decoder/index
  ├─decodersmall-sessiondb  253:9    0  600G  0 lvm   /var/netwitness/decoder/sessiondb
  └─decodersmall-metadb     253:10   0    3T  0 lvm   /var/netwitness/decoder/metadb
sdd                           8:48   0   40T  0 disk
└─sdd1                        8:49   0   40T  0 part
  └─decoder-packetdb        253:11   0   40T  0 lvm   /var/netwitness/decoder/packetdb
```

At this point, you add the second DAC enclosure.

8. 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`



9. 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

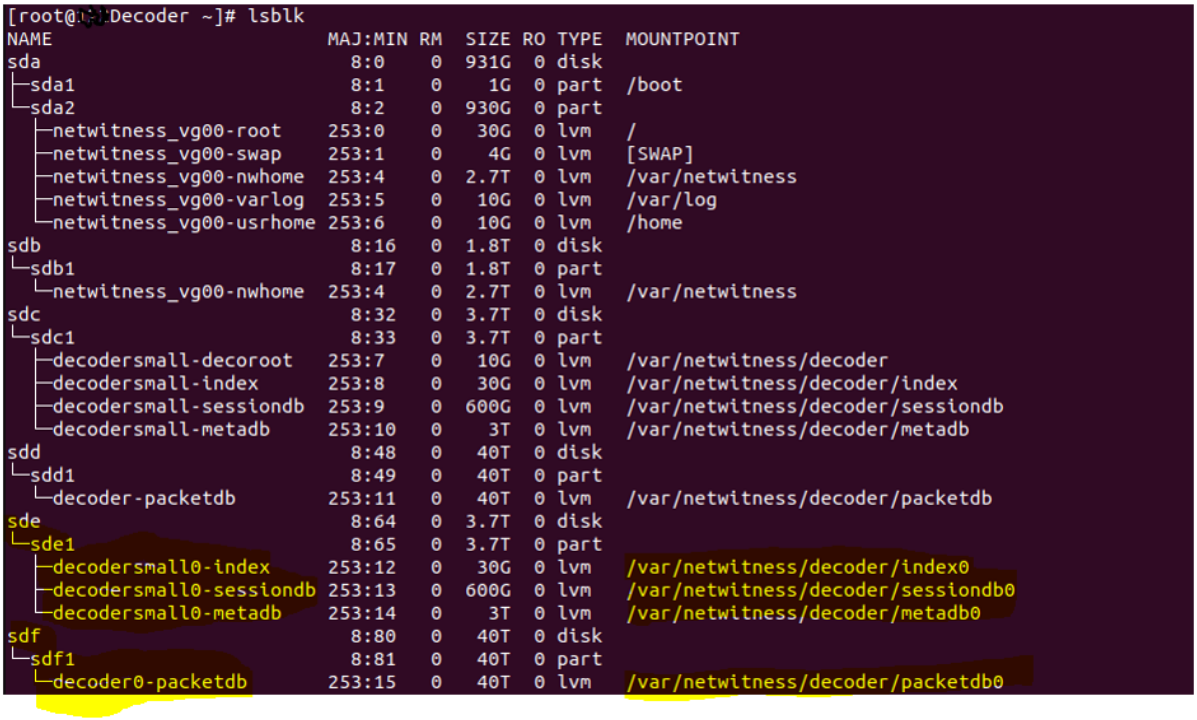
Message Help

```
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

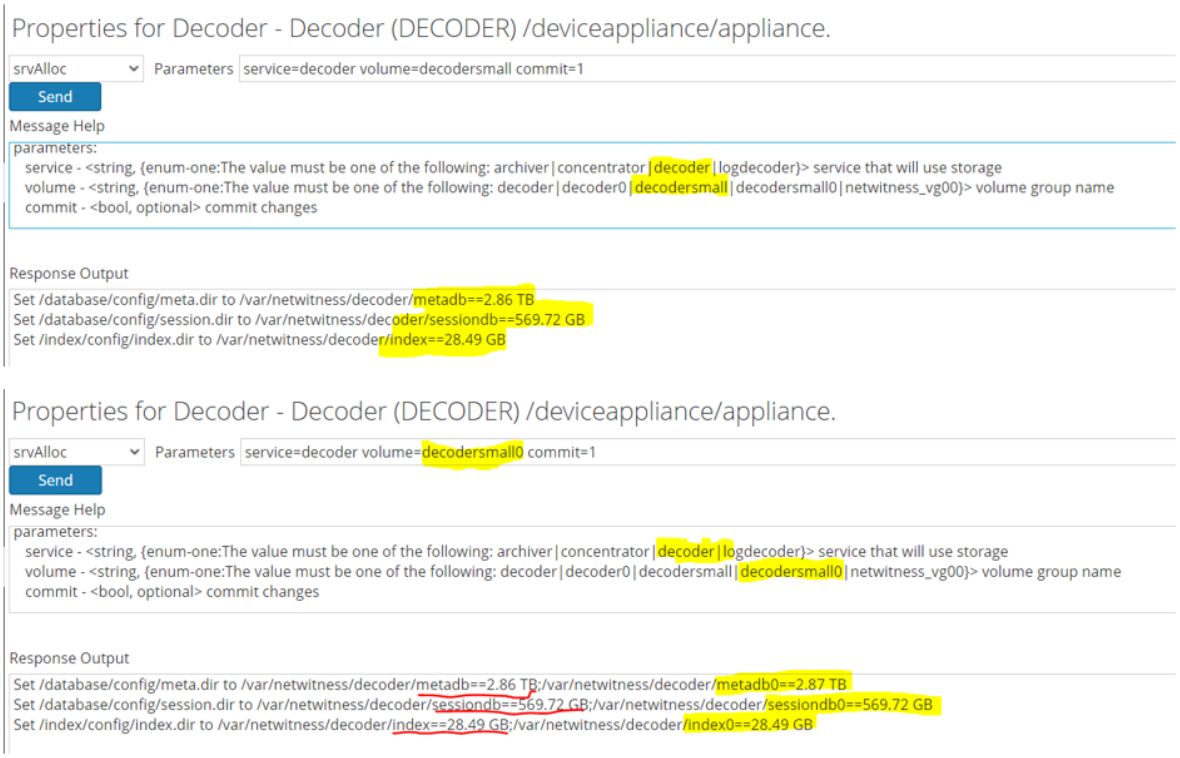
```
/sbin/parted -s /dev/sdf mklabel gpt
/sbin/parted -s -a optimal /dev/sdf mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdf1
  Physical volume "/dev/sdf1" successfully created.
/sbin/vgcreate -f decoder0 /dev/sdf1
  Volume group "decoder0" successfully created
/sbin/lvcreate -y -n packetdb -l 100%FREE 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@[redacted]Decoder ~]# df -hP
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                             63G     0   63G   0% /dev
tmpfs                                63G   60K   63G   1% /dev/shm
tmpfs                                63G   11M   63G   1% /run
tmpfs                                63G     0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-root     30G  3.9G   27G  13% /
/dev/sda1                          1014M   89M  926M   9% /boot
/dev/mapper/netwitness_vg00-varlog   10G   48M   10G   1% /var/log
/dev/mapper/netwitness_vg00-nwhome  2.7T  609M  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-usrhome  10G   33M   10G   1% /home
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   34M  600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decodersmall-metadb     3.1T   34M  3.1T   1% /var/netwitness/decoder/metadb
/dev/mapper/decoder-packetdb         41T   34M   41T   1% /var/netwitness/decoder/packetdb
/dev/mapper/decodersmall0-index      30G   33M   30G   1% /var/netwitness/decoder/index0
/dev/mapper/decodersmall0-sessiondb 600G   34M  600G   1% /var/netwitness/decoder/sessiondb0
/dev/mapper/decodersmall0-metadb    3.1T   34M  3.1T   1% /var/netwitness/decoder/metadb0
/dev/mapper/decoder0-packetdb        41T   34M   41T   1% /var/netwitness/decoder/packetdb0
```

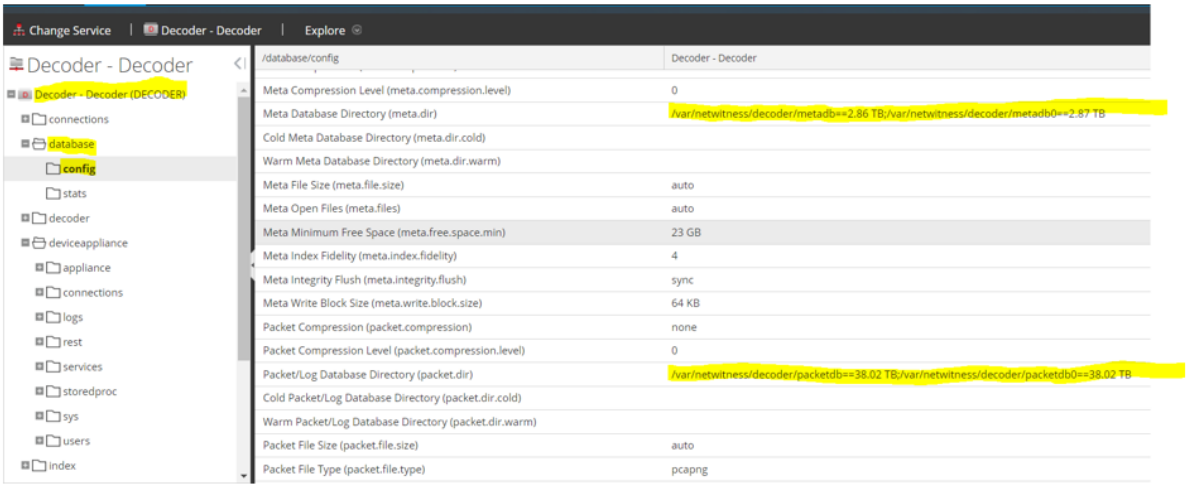


```
[root@[redacted]Decoder ~]# lsblk
NAME                          MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                             8:0    0  931G  0 disk
├─sda1                          8:1    0    1G  0 part  /boot
└─sda2                          8:2    0  930G  0 part
  ├─netwitness_vg00-root      253:0    0   30G  0 lvm   /
  ├─netwitness_vg00-swap      253:1    0    4G  0 lvm   [SWAP]
  ├─netwitness_vg00-nwhome    253:4    0  2.7T  0 lvm   /var/netwitness
  ├─netwitness_vg00-varlog    253:5    0   10G  0 lvm   /var/log
  └─netwitness_vg00-usrhome   253:6    0   10G  0 lvm   /home
sdb                             8:16   0  1.8T  0 disk
└─sdb1                          8:17   0  1.8T  0 part
  └─netwitness_vg00-nwhome    253:4    0  2.7T  0 lvm   /var/netwitness
sdc                             8:32   0  3.7T  0 disk
└─sdc1                          8:33   0  3.7T  0 part
  ├─decodersmall-decoroot     253:7    0   10G  0 lvm   /var/netwitness/decoder
  ├─decodersmall-index        253:8    0   30G  0 lvm   /var/netwitness/decoder/index
  ├─decodersmall-sessiondb    253:9    0  600G  0 lvm   /var/netwitness/decoder/sessiondb
  └─decodersmall-metadb       253:10   0    3T  0 lvm   /var/netwitness/decoder/metadb
sdd                             8:48   0   40T  0 disk
└─sdd1                          8:49   0   40T  0 part
  └─decoder-packetdb          253:11   0   40T  0 lvm   /var/netwitness/decoder/packetdb
sde                             8:64   0  3.7T  0 disk
└─sde1                          8:65   0  3.7T  0 part
  ├─decodersmall0-index       253:12   0   30G  0 lvm   /var/netwitness/decoder/index0
  ├─decodersmall0-sessiondb   253:13   0  600G  0 lvm   /var/netwitness/decoder/sessiondb0
  └─decodersmall0-metadb      253:14   0    3T  0 lvm   /var/netwitness/decoder/metadb0
sdf                             8:80   0   40T  0 disk
└─sdf1                          8:81   0   40T  0 part
  └─decoder0-packetdb         253:15   0   40T  0 lvm   /var/netwitness/decoder/packetdb0
```

10. 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=decodersmall0 commit=1`
    - `service=decoder volume=decoder commit=1`
    - `service=decoder volume=decoder0 commit=1`
11. Allocate decodersmall and decodersmall0 volumes to decoder service as shown below. Similarly, decoder and decoder0 volumes are allocated to decoder service.



12. Use **Explore->database->config->Meta Database Directory / Session Database Directory / Packet/Log Database Directory** parameter to confirm the values set using srvAlloc.



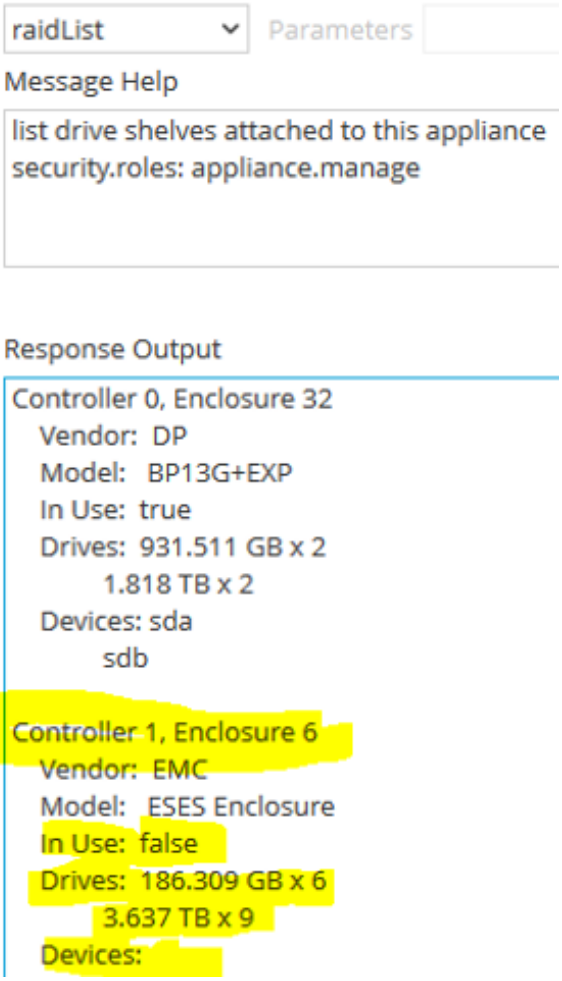
Packet Open Files (packet.files)	auto
Packet Minimum Free Space (packet.free.space.min)	23 GB
Packet Index Fidelity (packet.index.fidelity)	1
Packet Integrity Flush (packet.integrity.flush)	sync
Packet Write Block Size (packet.write.block.size)	64 KB
Session Database Directory (session.dir)	/var/netwitness/decoder/sessiondb==569.72 GB;/var/netwitness/decoder/sessiondb0==569.72 GB
Cold Session Database Directory (session.dir.cold)	
Warm Session Database Directory (session.dir.warm)	
Session File Size (session.file.size)	**auto**
Session Open Files (session.files)	auto
Session Minimum Free Space (session.free.space.min)	23 GB
Session Integrity Flush (session.integrity.flush)	sync
Session Write Block Size (session.write.block.size)	32 KB

13. 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.

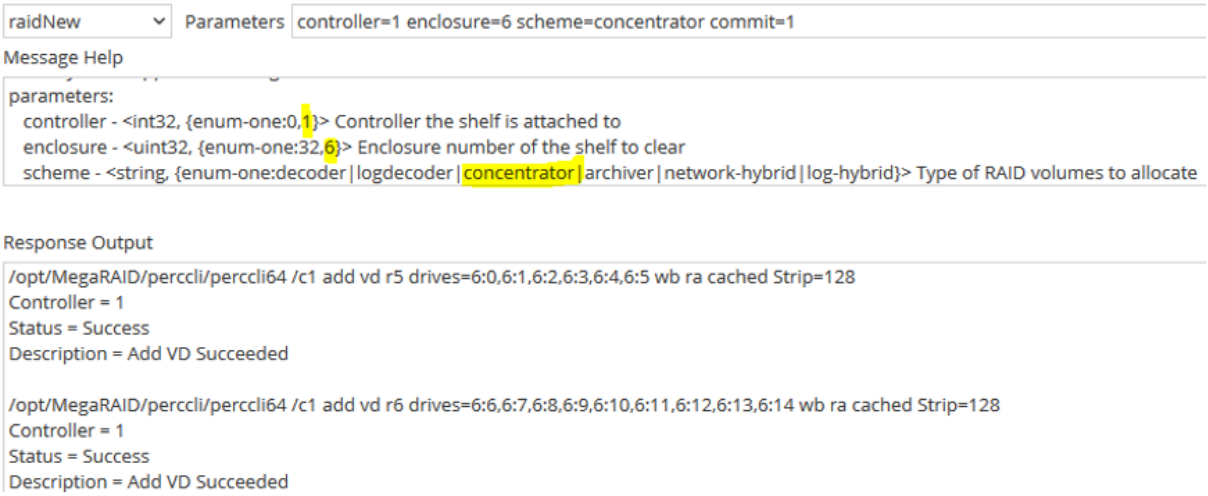
Configure Storage for Network Concentrator

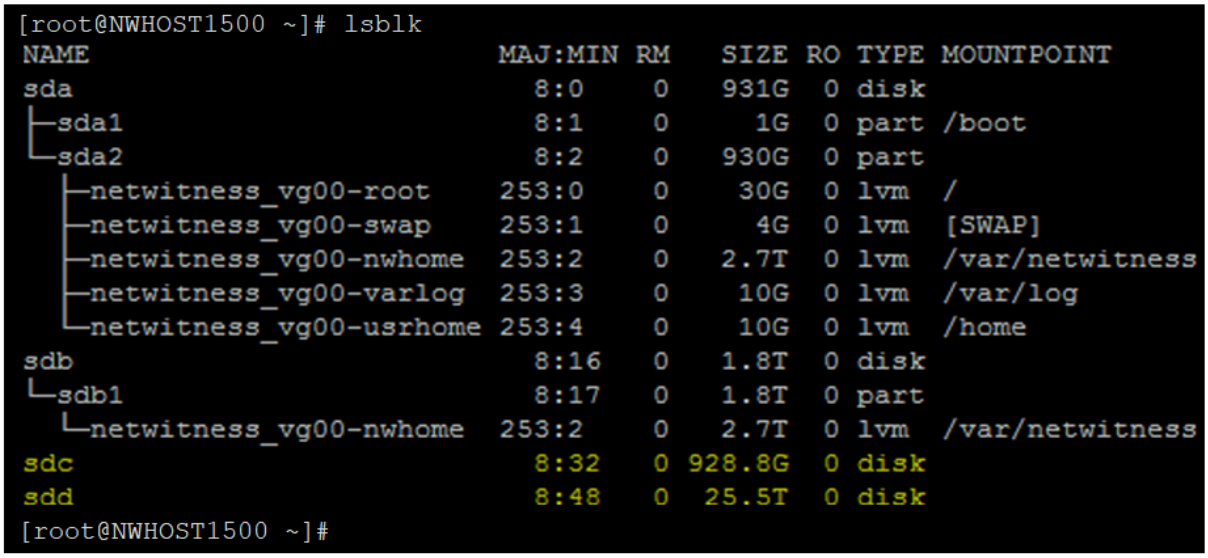
The following scenario configures storage on one, non-encrypted, 15-Drive DAC for a Network Concentrator physical host.

1. Execute the `raidList` command.



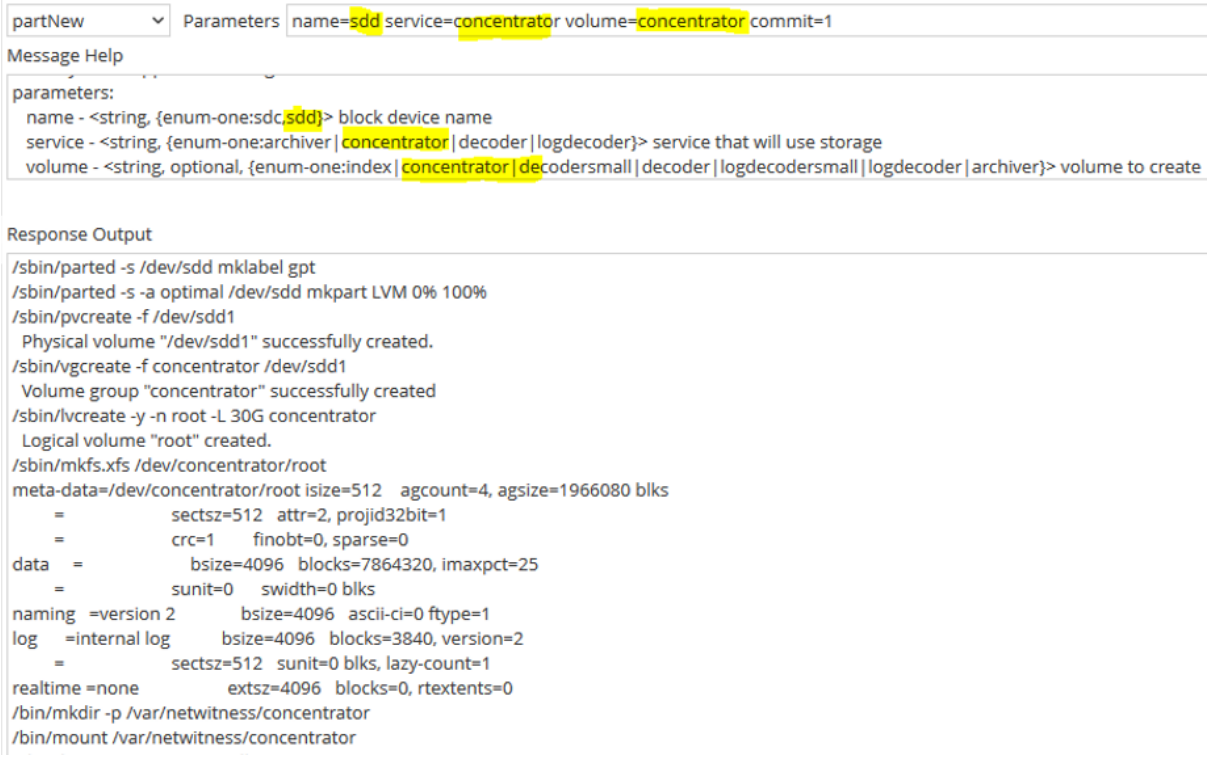
2. Execute the `raidNew` command with the following parameters.
   `controller=1 enclosure=6 scheme=concentrator`





3. 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
```




```
[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
```

4. Execute the partNew command with the following parameters 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

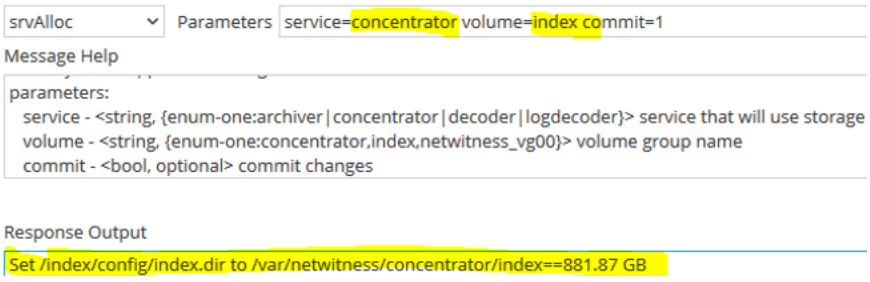
Response Output

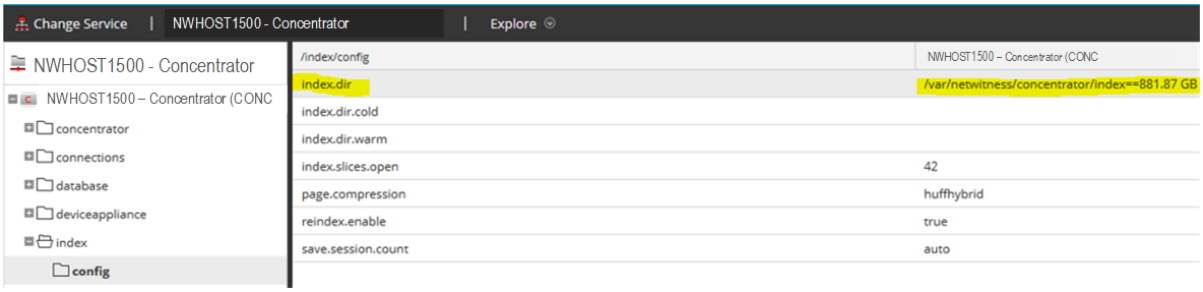
```
/sbin/parted -s /dev/sdc mklabel gpt
/sbin/parted -s -a optimal /dev/sdc mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdc1
  Physical volume "/dev/sdc1" successfully created.
/sbin/vgcreate -f index /dev/sdc1
  Volume group "index" successfully created
/sbin/lvcreate -y -n index -l 100%FREE index
  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
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
└─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
```

5. Execute the `srvAlloc` command with the following parameters.
   `service=concentrator volume=index commit=1`





6. Execute the `srvAlloc` command with the following parameters.
   `service=concentrator volume=concentrator commit=1`

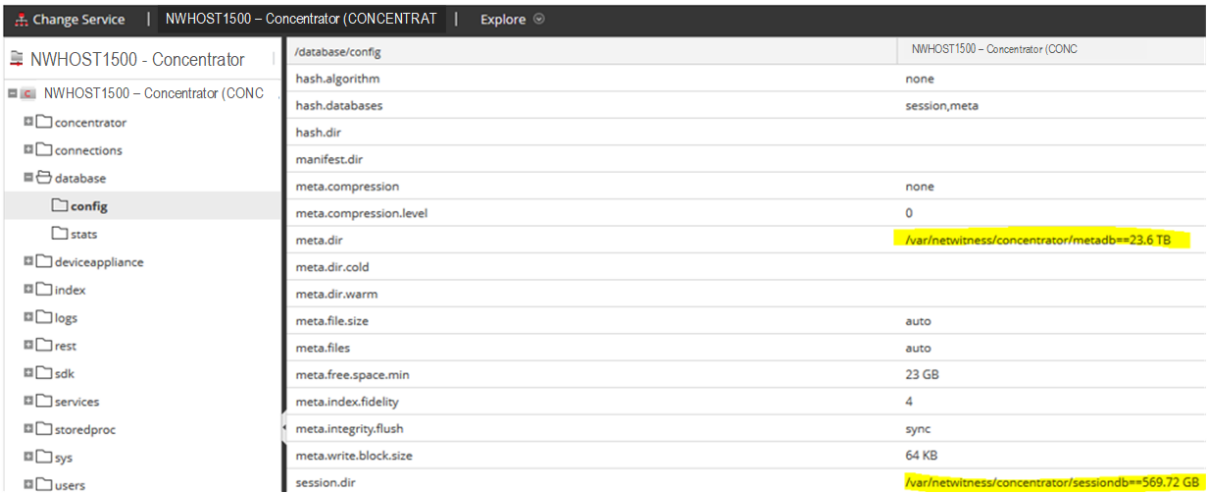
srvAlloc Parameters service=concentrator volume=concentrator commit=1

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

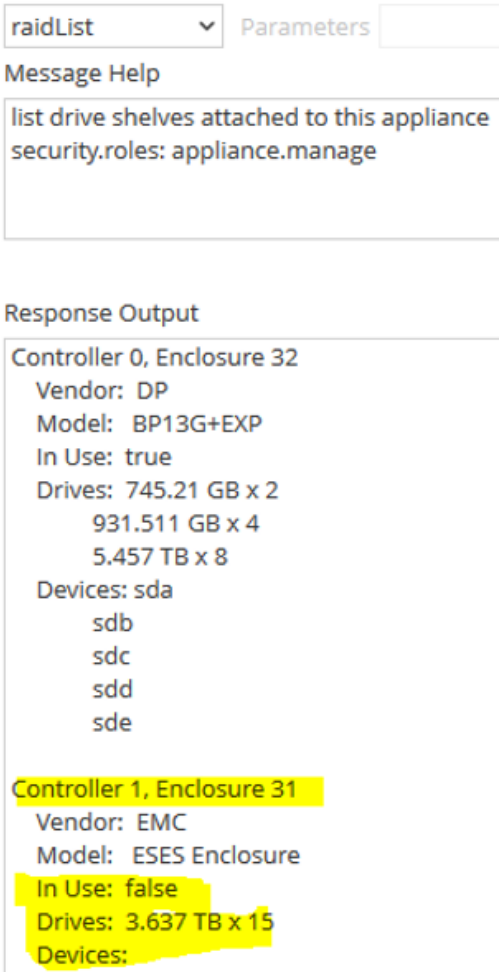
Set /database/config/meta.dir to /var/netwitness/concentrator/metadb==23.6 TB
Set /database/config/session.dir to /var/netwitness/concentrator/sessiondb==569.72 GB



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.



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
```

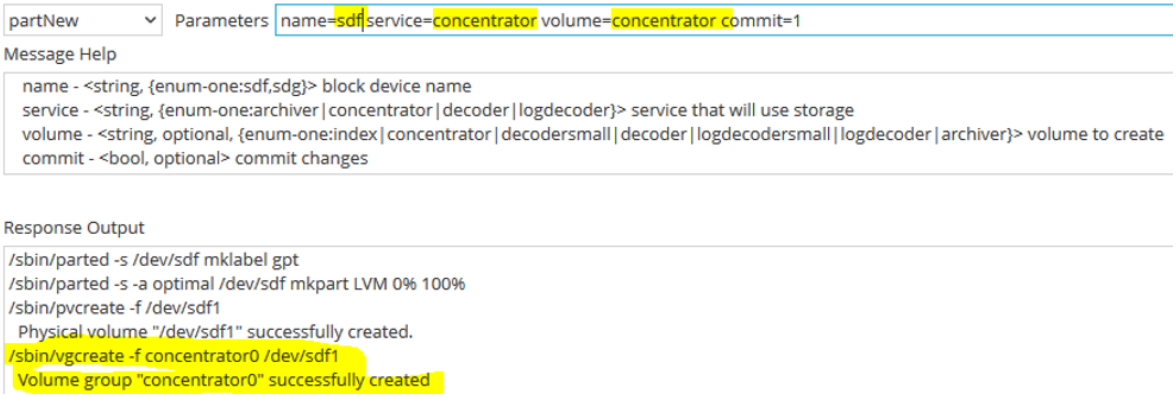
Response Output

```
/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`




```
[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 Parameters name=sdg service=logdecoder volume=logdecoder 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
```

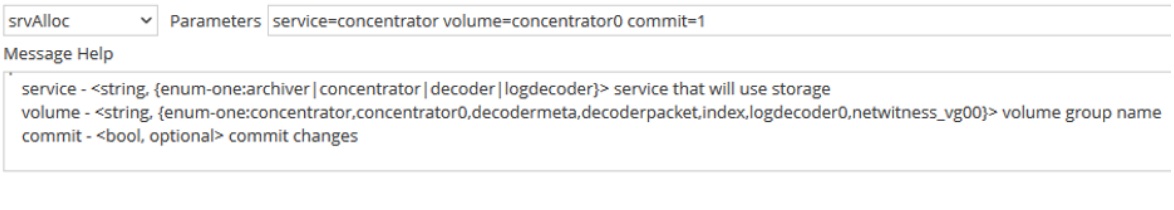
Response Output

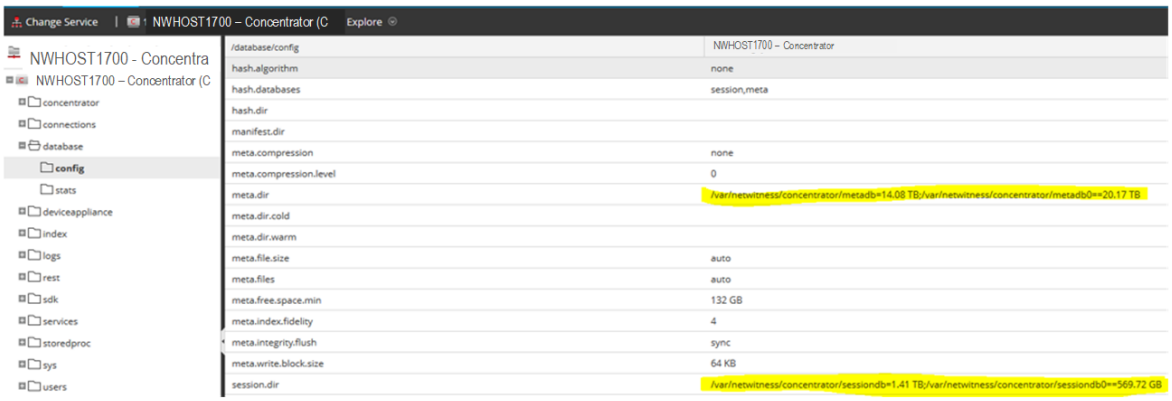
```
/sbin/parted -s /dev/sdg mklabel gpt
/sbin/parted -s -a optimal /dev/sdg mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdg1
  Physical volume "/dev/sdg1" successfully created.
/sbin/vgcreate -f logdecoder0 /dev/sdg1
  Volume group "logdecoder0" successfully created
```

```
[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`





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

Appendix E. Sample Storage Configuration Scenarios for 8 or 12-Drive PowerVault

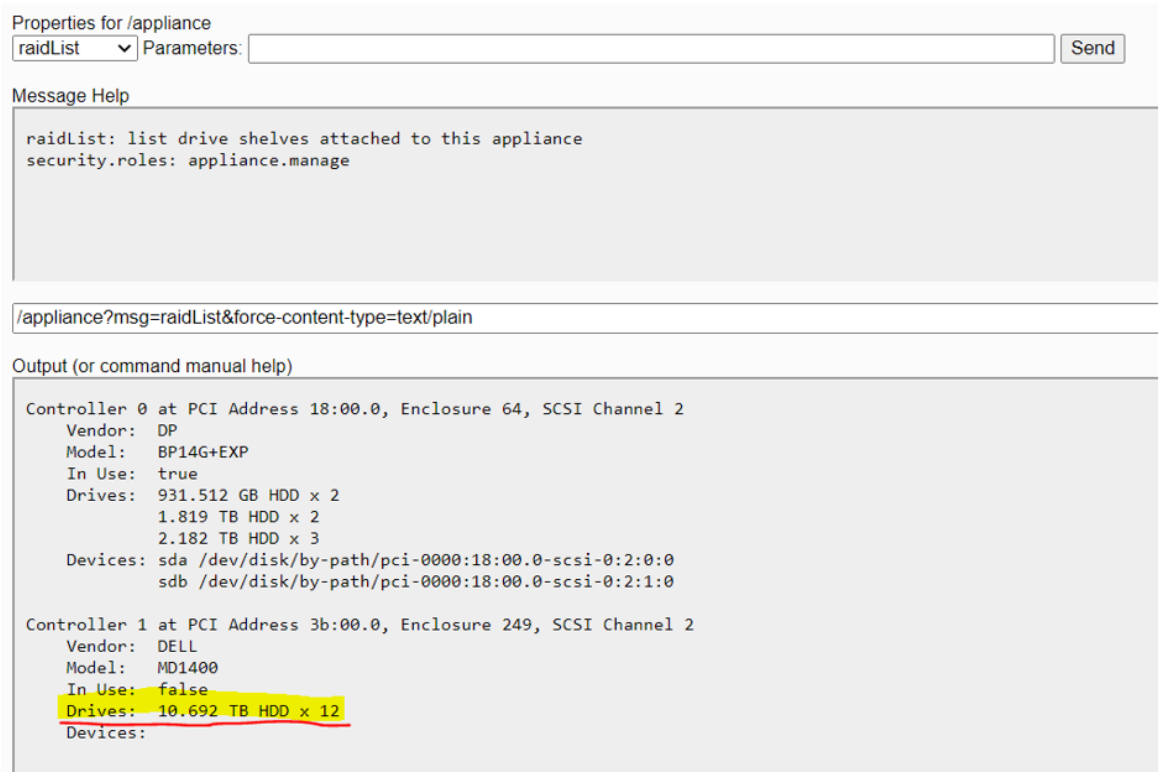
This appendix illustrates the following example of how to configure storage on one non-encrypted 8 or 12-drive PowerVault external storage devices.

Configure Storage for Archiver using NW-PV-A/NW-PV-A-N

The following scenario configures storage on one, non-encrypted, 12-Drive PowerVault 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>
   ```

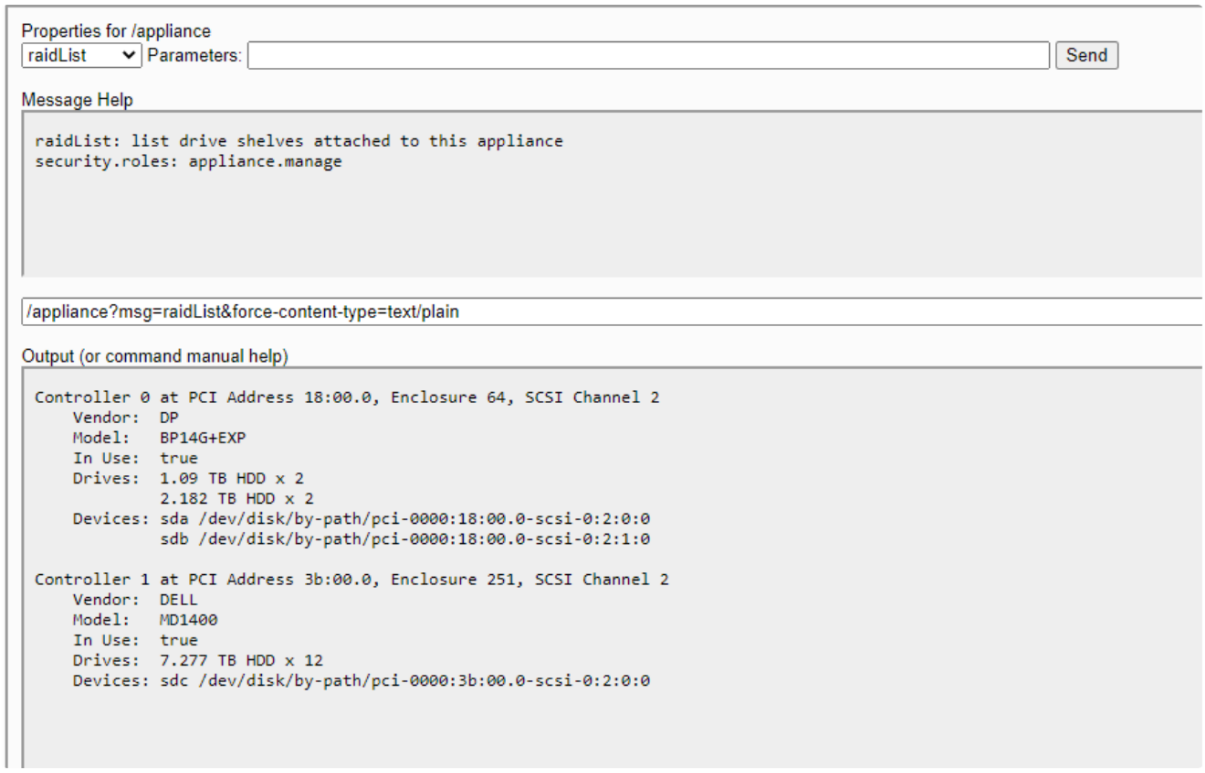
b. Verify the Drive Count, Size, and Vendor.
The following example illustrates what you should see before you create a RAID array.



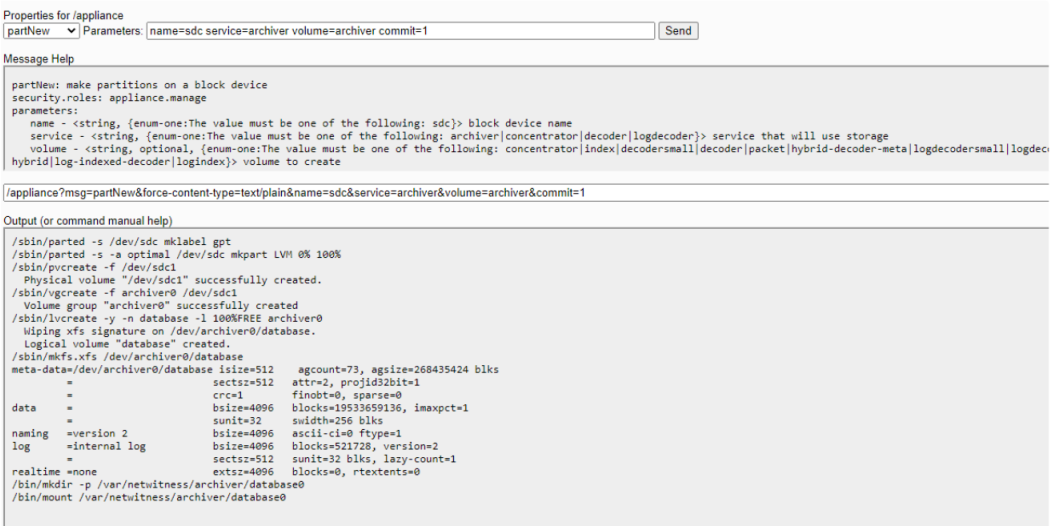
2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
`controller=1 enclosure=251 scheme=archiver commit=1`
The following example illustrates what you should see after you create a RAID array.

3. 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`)



4. Execute the `partNew` command with the following parameters to create partitions and mount points in the `etc/fstab` file.
   `name=sdc service=archiver volume=archiver commit=1`



5. 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`

Configure Storage for Decoder using NW-PV-B/NW-PV-B-N

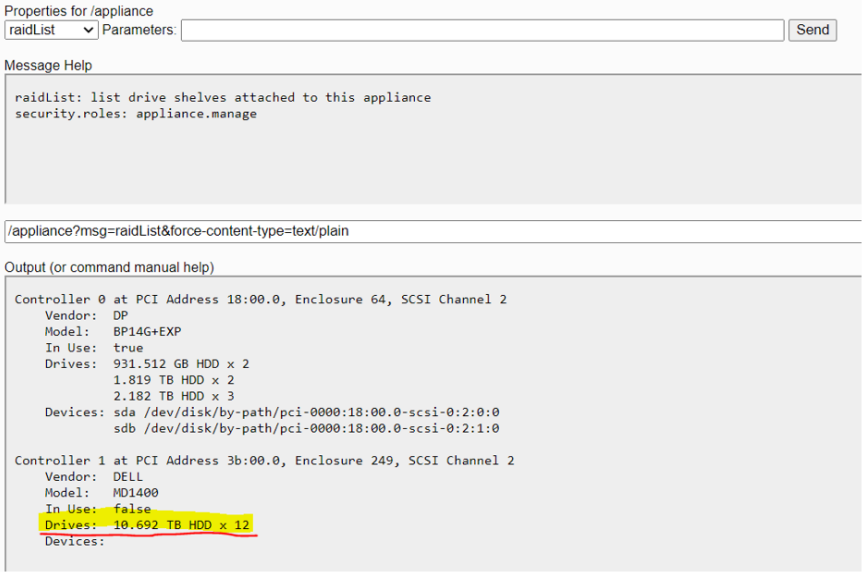
The following scenario configures storage on one, non-encrypted, 12-Drive PowerVault for a Network Decoder physical host.

Note: The block device size depends on the PV type (drive count) and the drive size (8 TB or 12 TB or 16 TB).

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.



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-hotspare commit=1`

Scheme	Enclosure Type	Drives Required	Allocation
decoder	External	8 or 12 or 15 HDDs	3x drives in RAID 5 for decodersmall, all remaining drives in RAID 5 for decoder
log decoder	External	8 or 12 or 15 HDDs	Same as decoder configuration
decoder hot spare	External	8 or 12 or 15 HDDs	2x drives in RAID 1 for decodersmall, 1 drive as hot spare, all remaining drives in RAID 5 for decoder
log decoder hot spare	External	8 or 12 or 15 HDDs	Same as decoder hot spare configuration
archiver	External	8 or 12 or 15 HDDs	All drives in RAID 6 for archiver database volume
packet expansion	External	8 or 12 or 15 HDDs	All drives in RAID 6 for decoder volume. No drives allocated for decodersmall.
network hybrid	External	8 or 12 or 15 HDDs	3x drives in RAID 5 for meta expansion, all remaining drives in RAID 5 for packet expansion
network hybrid	Internal	S5 / S6 hybrid drive set	2x small HDD RAID 1 for decoder meta, 5x large HDD decoder, 3x large HDD concentrator, 2x SSD index
log hybrid	External	8 or 12 or 15 HDDs	Half of the drives in RAID 5 for meta expansion, half the drives in RAID 5 for packet expansion
log hybrid	Internal	S5 / S6 hybrid drive set	2x small HDD RAID 1 for decoder meta, 4x large HDD decoder, 4x large HDD concentrator, 2x SSD index
endpoint hybrid	Internal	S5 / S6 hybrid drive set	2x small HDD RAID 1 for decoder meta, 4x large HDD RAID 10 for log decoder and endpoint, 4x large HDD RAID 5 for concentrator, 2x SSD index
log indexed decoder	Internal	S6E hybrid drive set	10x HDD RAID 6 for log decoder meta and packet, 2x SSD index
concentrator	External	2 or more SSDs, 4 or more HDDs	All SSDs in RAID 1 or RAID 5 for index, all HDDs in RAID 6 for meta

raidNew Parameters controller=1 enclosure=249 scheme=decoder-hotspare commit=1

Send

Message Help

parameters:
controller - <uint32, {enum-one:The value must be one of the following: 0,1}> Controller the shelf is attached to
enclosure - <uint32, optional, {enum-one:The value must be one of the following: 64|249}> Enclosure number of the shelf to clear. Required if the controller has more than one enclosure attached.
scheme - <string, {enum-one:The value must be one of the following: decoder|logdecoder|concentrator|archiver|network-hybrid|log-hybrid|log-indexed-decoder|endpoint-log-hybrid|packet-

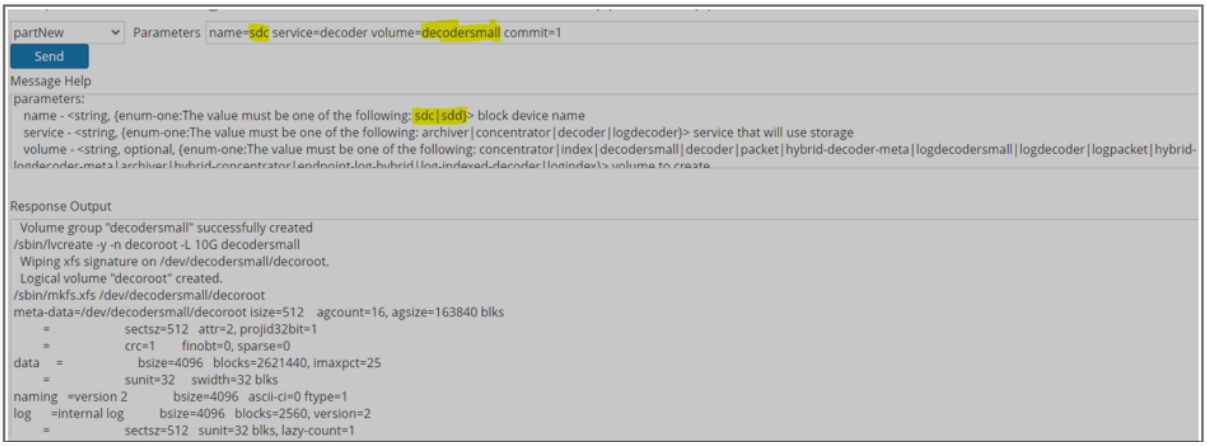
Response Output

```
/opt/MegaRAID/perccli/perccli64 /c1 add vd r1 drives=249:0,249:1 ra Strip=128
CLI Version = 007.1623.0000.0000 May 17, 2021
Operating system = Linux 3.10.0-1160.83.1.el7.x86_64
Controller = 1
Status = Success
Description = Add VD Succeeded.


/opt/MegaRAID/perccli/perccli64 /c1 add vd r5 drives=249:2,249:3,249:4,249:5,249:6,249:7,249:8,249:9,249:10 ra Strip=128
CLI Version = 007.1623.0000.0000 May 17, 2021
Operating system = Linux 3.10.0-1160.83.1.el7.x86_64
Controller = 1
Status = Success
Description = Add VD Succeeded.
```

```
Description = Add VD Succeeded.


/opt/MegaRAID/perccli/perccli64 /c1 /e249 /s11 add hotsparedrive
CLI Version = 007.1623.0000.0000 May 17, 2021
Operating system = Linux 3.10.0-1160.83.1.el7.x86_64
Controller = 1
Status = Success
Description = Add Hot Spare Succeeded.
```

```
[root@[redacted]]# 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 10.7T  0 disk
sdd                         8:48   0 85.5T  0 disk
```

3. 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`




```
[root@s6coreappliance ~]# df -kh
Filesystem                              Size  Used Avail Use% Mounted on
devtmpfs                                 63G     0   63G   0% /dev
tmpfs                                    63G   40K   63G   1% /dev/shm
tmpfs                                    63G   19M   63G   1% /run
tmpfs                                    63G     0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-root         30G  6.3G   24G  21% /
/dev/sda1                              1014M   91M  924M   9% /boot
/dev/mapper/netwitness_vg00-nwhome      2.7T  1.2G  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-usrhome      10G   33M   10G   1% /home
/dev/mapper/netwitness_vg00-varlog       10G  915M  9.1G   9% /var/log
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   34M  600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decodersmall-metadb          11T   34M   11T   1% /var/netwitness/decoder/metadb
/dev/mapper/decoder-packetdb             86T   37M   86T   1% /var/netwitness/decoder/packetdb
```

4. 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`

```
[root@s6coreappliance ~]# 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 10.7T  0 disk
└─sdc1                          8:33   0 10.7T  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 10.1T  0 lvm  /var/netwitness/decoder/metadb
sdd                             8:48   0 85.5T  0 disk
└─sdd1                          8:49   0 85.5T  0 part
  └─decoder-packetdb          253:9    0 85.5T  0 lvm  /var/netwitness/decoder/packetdb
```

Use lsblk and df -kh to confirm the block device sizes and disk allocation.

```
[root@s6coreappliance ~]# df -kh
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                             63G     0   63G   0% /dev
tmpfs                                63G   40K   63G   1% /dev/shm
tmpfs                                63G   19M   63G   1% /run
tmpfs                                63G     0   63G   0% /sys/fs/cgroup
/dev/mapper/netwitness_vg00-root     30G  6.3G   24G  21% /
/dev/sda1                          1014M   91M  924M   9% /boot
/dev/mapper/netwitness_vg00-nwhome   2.7T  1.2G  2.7T   1% /var/netwitness
/dev/mapper/netwitness_vg00-usrhome   10G   33M   10G   1% /home
/dev/mapper/netwitness_vg00-varlog    10G  915M  9.1G   9% /var/log
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   34M  600G   1% /var/netwitness/decoder/sessiondb
/dev/mapper/decodersmall-metadb       11T   34M   11T   1% /var/netwitness/decoder/metadb
/dev/mapper/decoder-packetdb          86T   37M   86T   1% /var/netwitness/decoder/packetdb
```

5. 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=decoder commit=1`

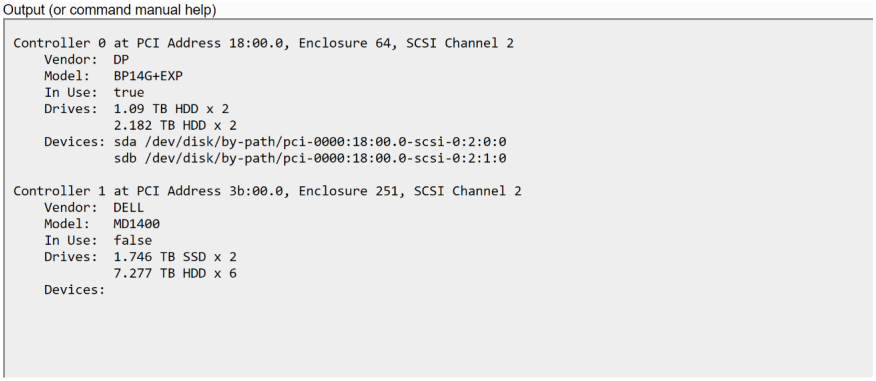
Configure Storage for Concentrator using NW-PV-C/NW-PV-C-N

The following scenario configures storage on one, non-encrypted, 8-Drive PowerVault for a Network Concentrator physical host.

1. Execute the `raidList` command.

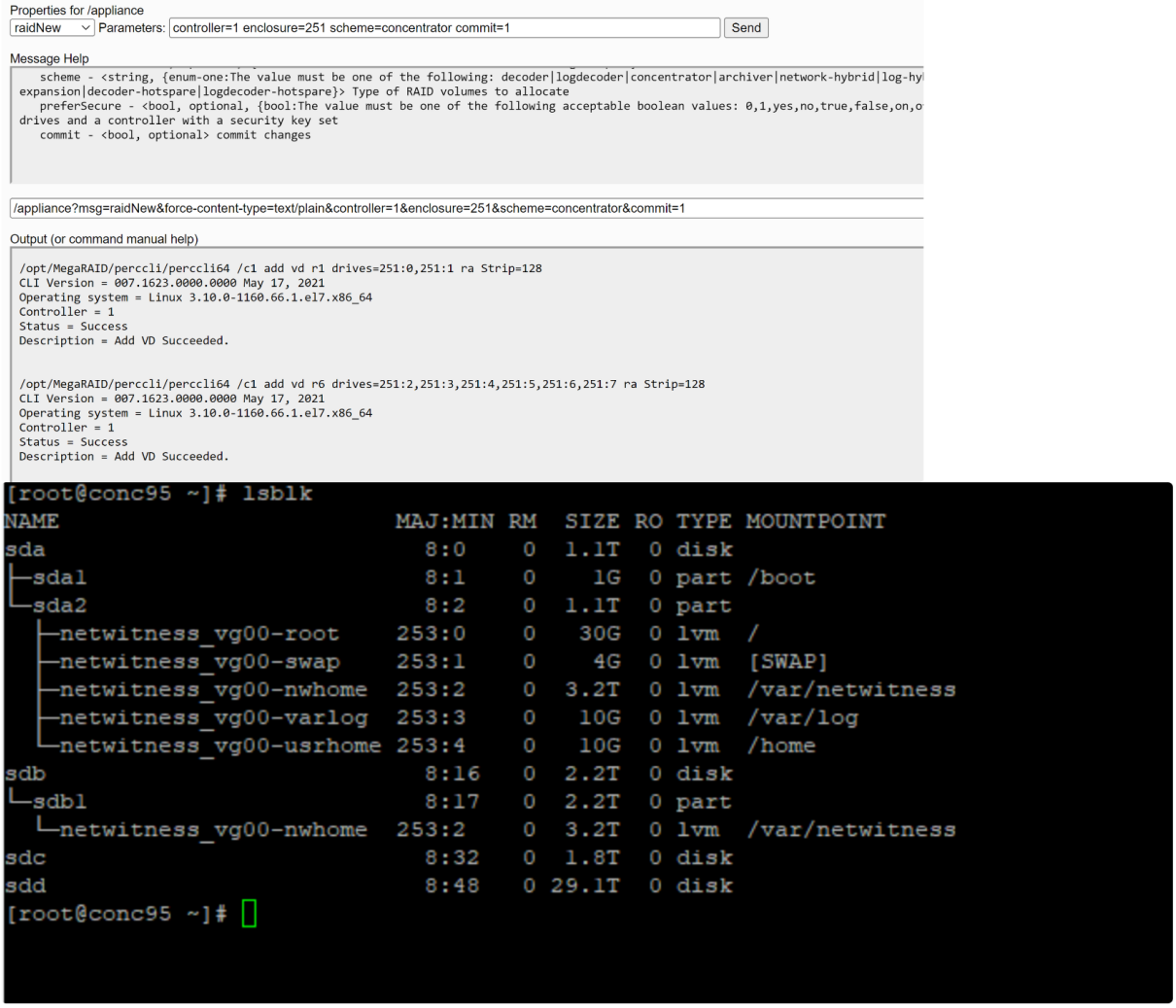
   a. Record the Controller Number, Enclosure Number, In Use, Drives, and Devices.
   You should see the following information.
   ```
   In Use: FALSE
   Devices: <empty>
   ```

   b. Verify the Drive Count, Size, and Vendor.
   The following example illustrates what you should see before you create a RAID array.



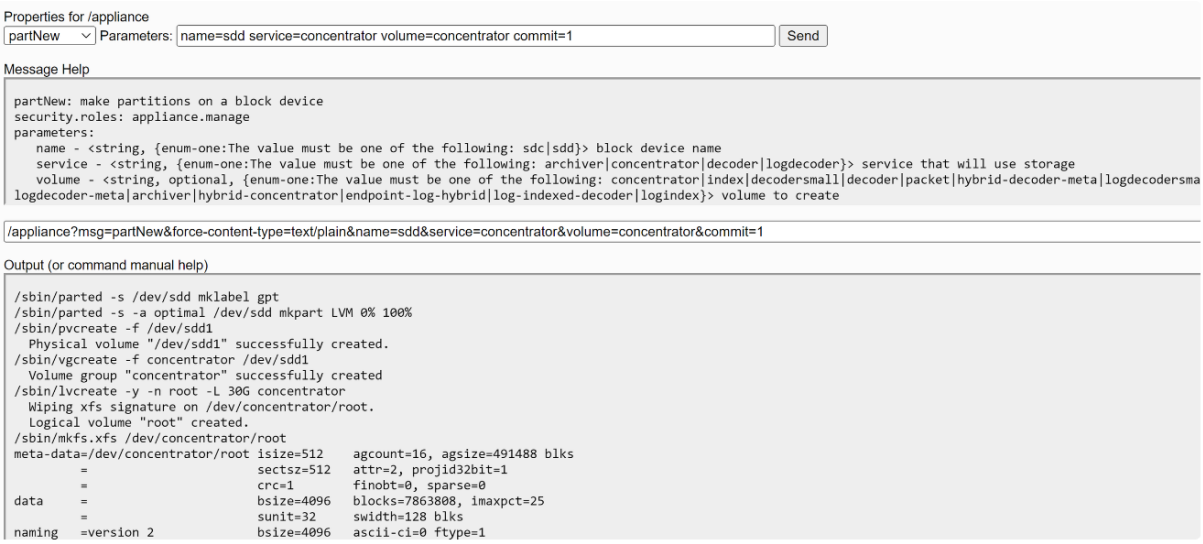
2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
   ```
   controller=1 enclosure=251 scheme=concentrator commit=1
   ```

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



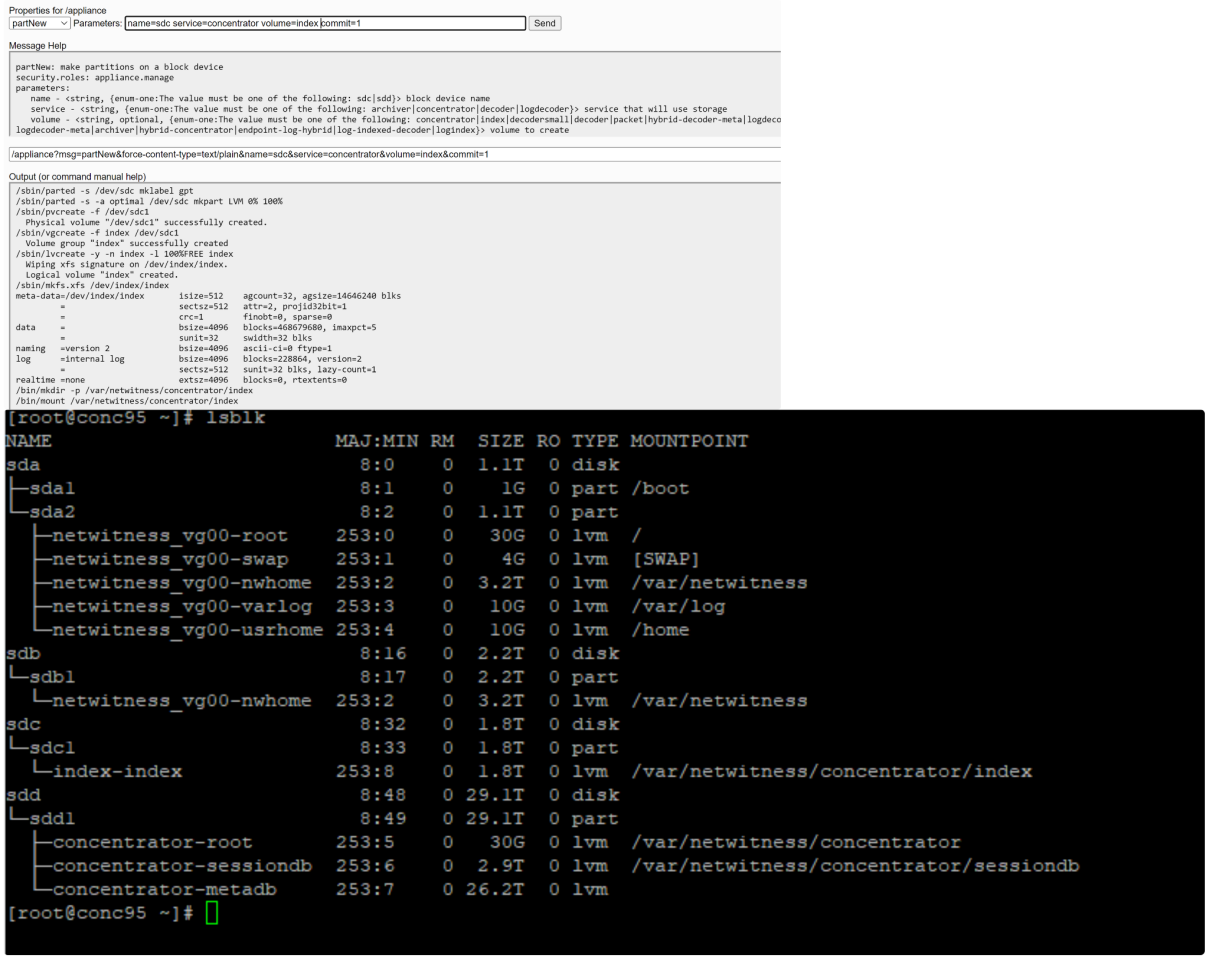
3. Execute the `partNew` command to create the **concentrator** partition first with the following parameters. You must create the **concentrator** volume before **index** volume or it will fail.

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



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

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



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

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

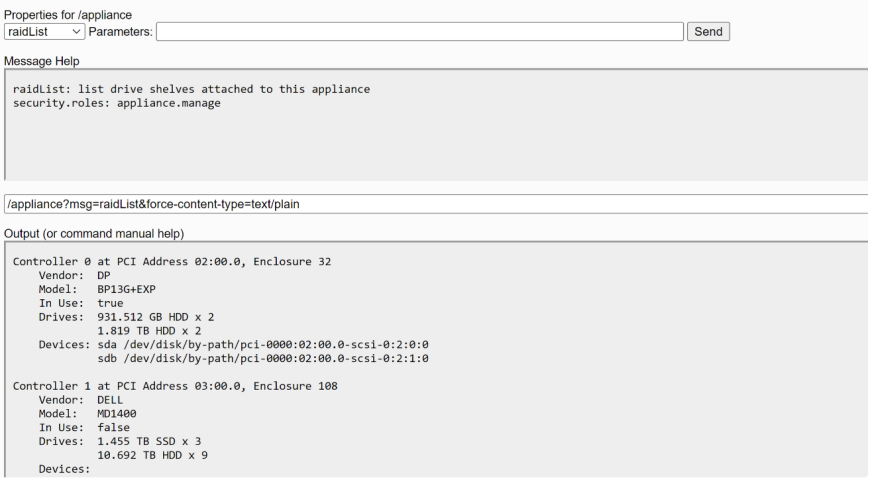
Configure Storage for Concentrator using NW-PV-D/NW-PV-D-N

The following scenario configures storage on one, non-encrypted, 12-Drive PowerVault for a Log Decoder Hybrid physical host.

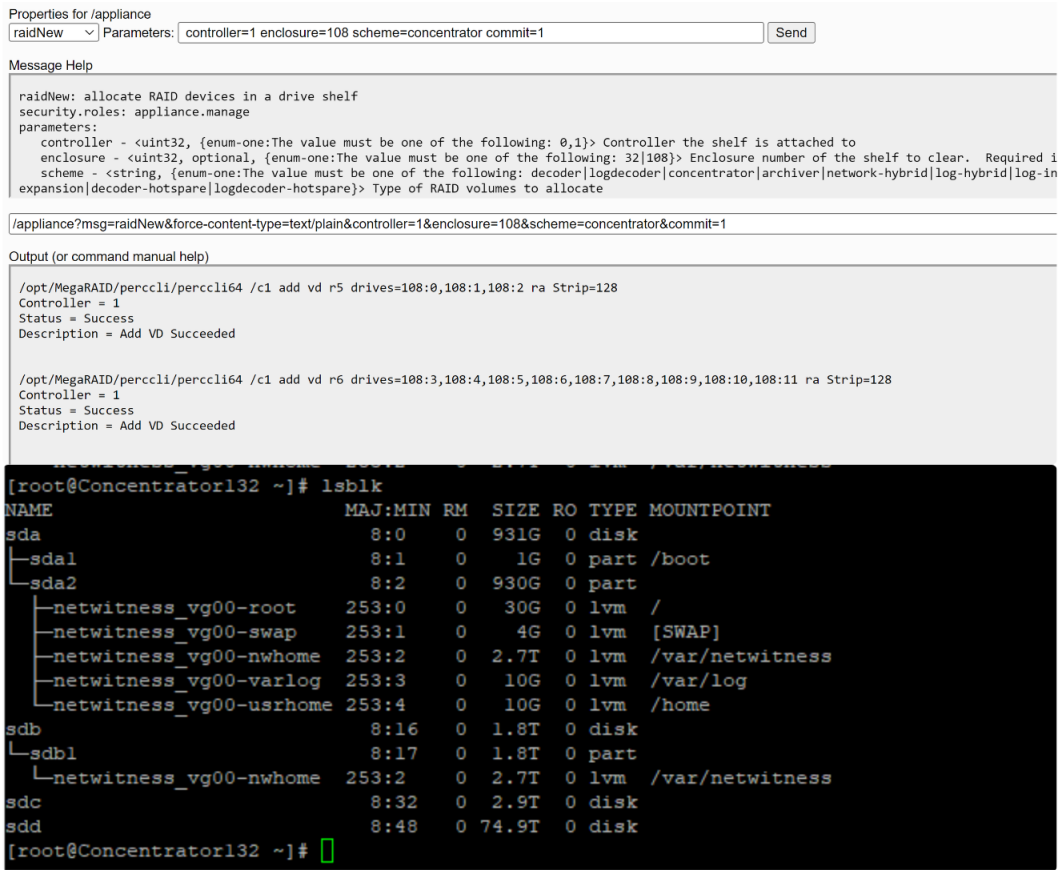
1. Execute the `raidList` command.

   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.

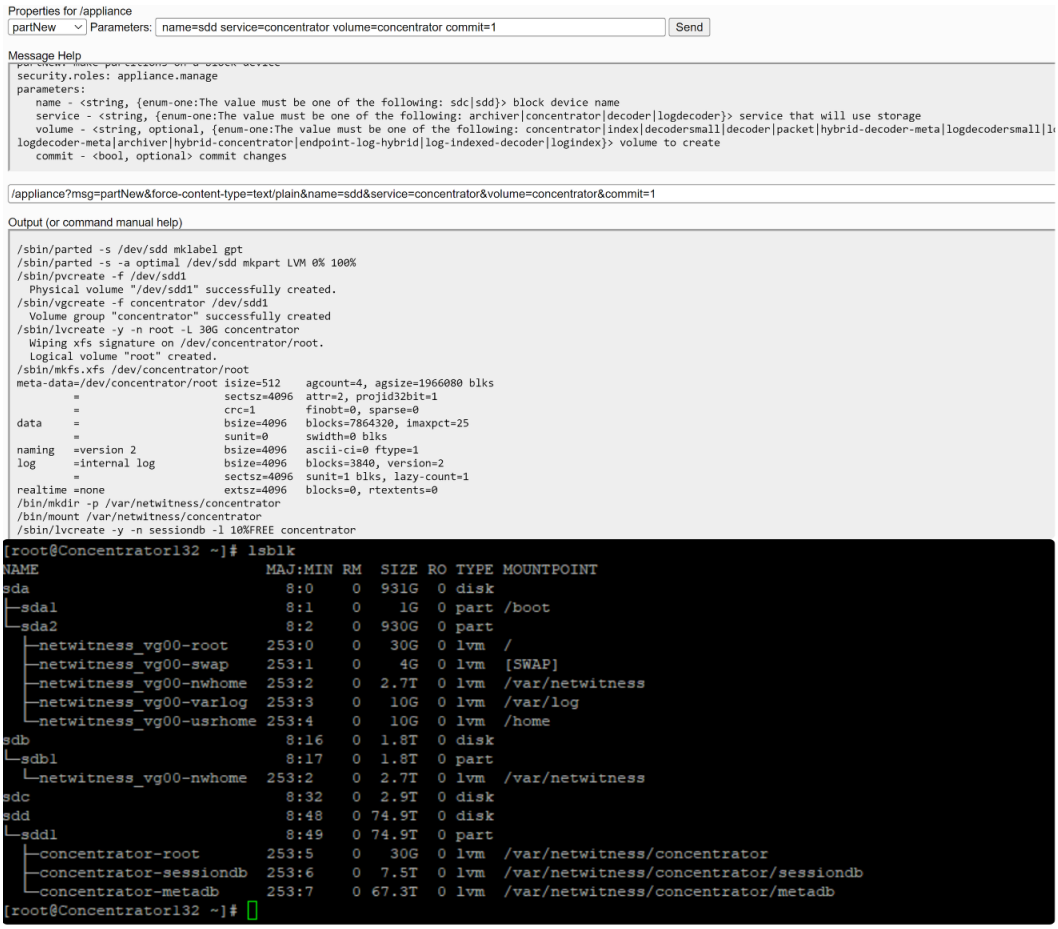


2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
   `controller=1 enclosure=108 scheme=concentrator commit=1`
   The following example illustrates what you should see after you create a RAID array.



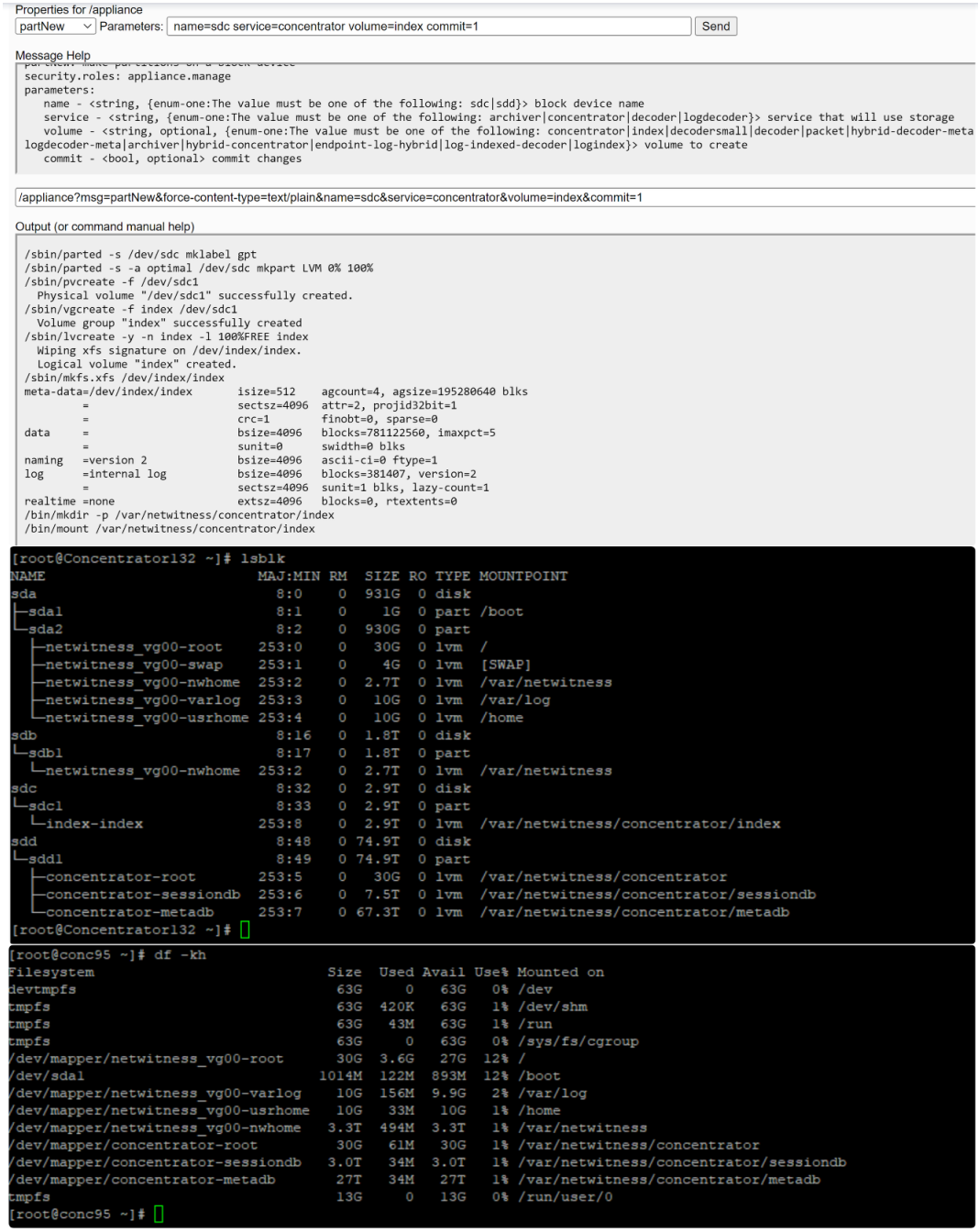
3. Execute the `partNew` command to create the **concentrator** partition first with the following parameters. You must create the concentrator volume before index volume or it will fail.

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



4. Execute the `partNew` command with the following parameters to create an **index** on SSDs.

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



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

   `service=concentrator volume=concentrator commit=1`

Configure Storage for Log Hybrid using NW-PV-A/NW-PV-A-N

The following scenario configures storage on one, non-encrypted, 12-Drive PowerVault for a Log Hybrid physical host.

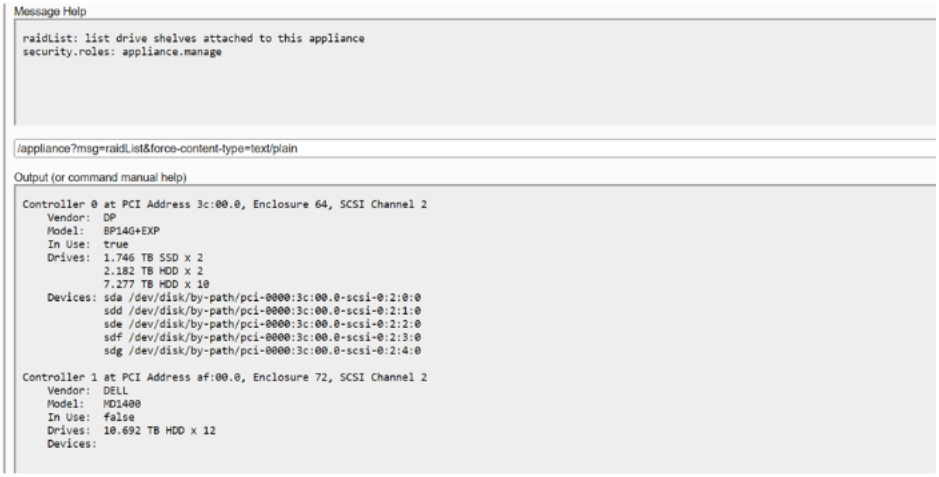
1. Execute the `raidList` command

   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.

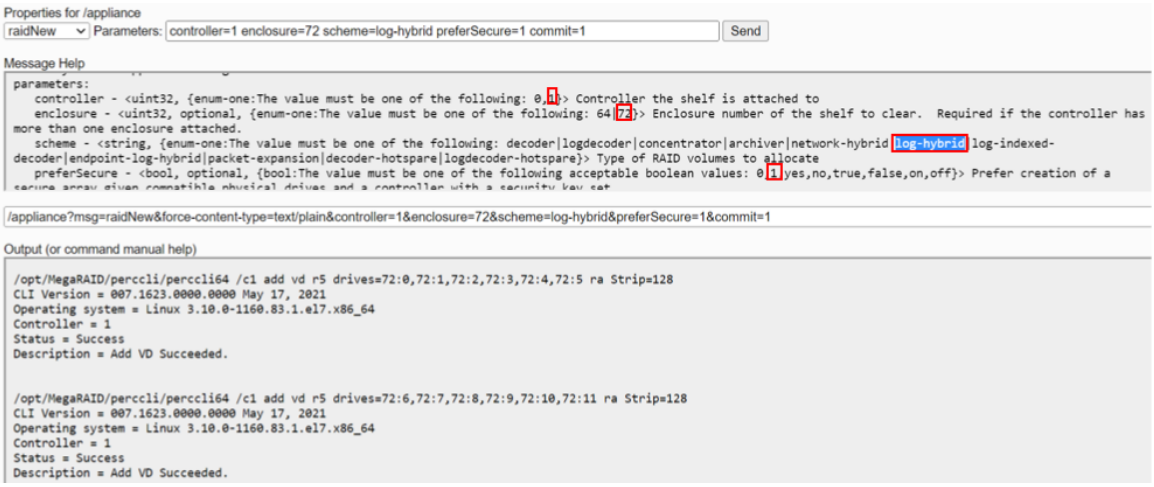
   The following example illustrates what you should see before you create a RAID array.



2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.

   ```
   controller=1 enclosure=72 scheme=log-hybrid preferSecure=1 commit=1
   ```

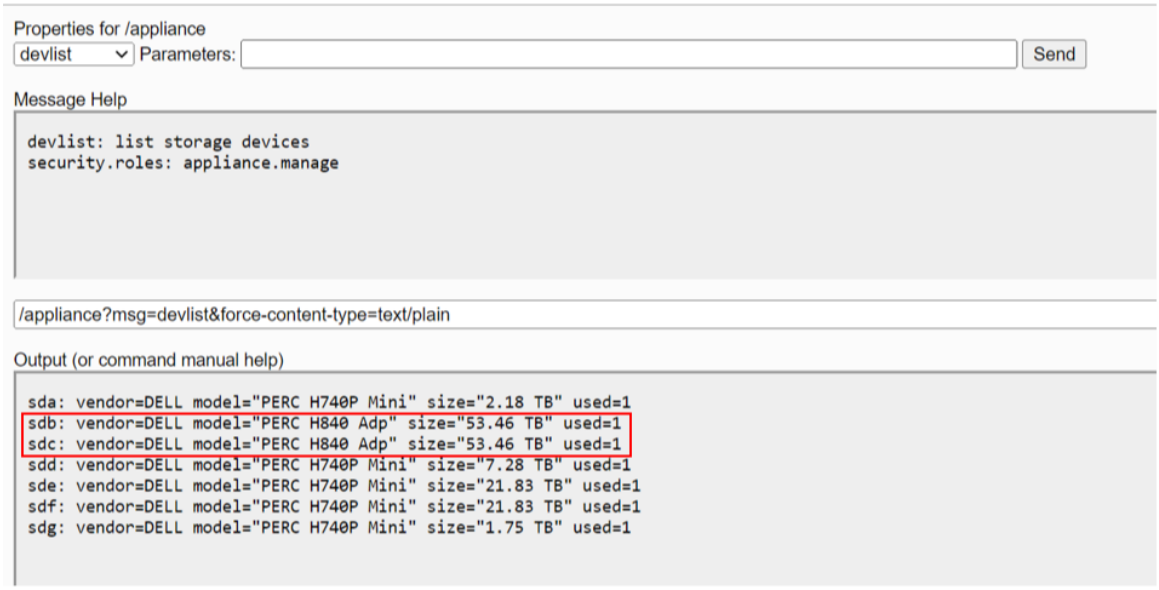
   a. The following example illustrates what you should see after you create a RAID array.



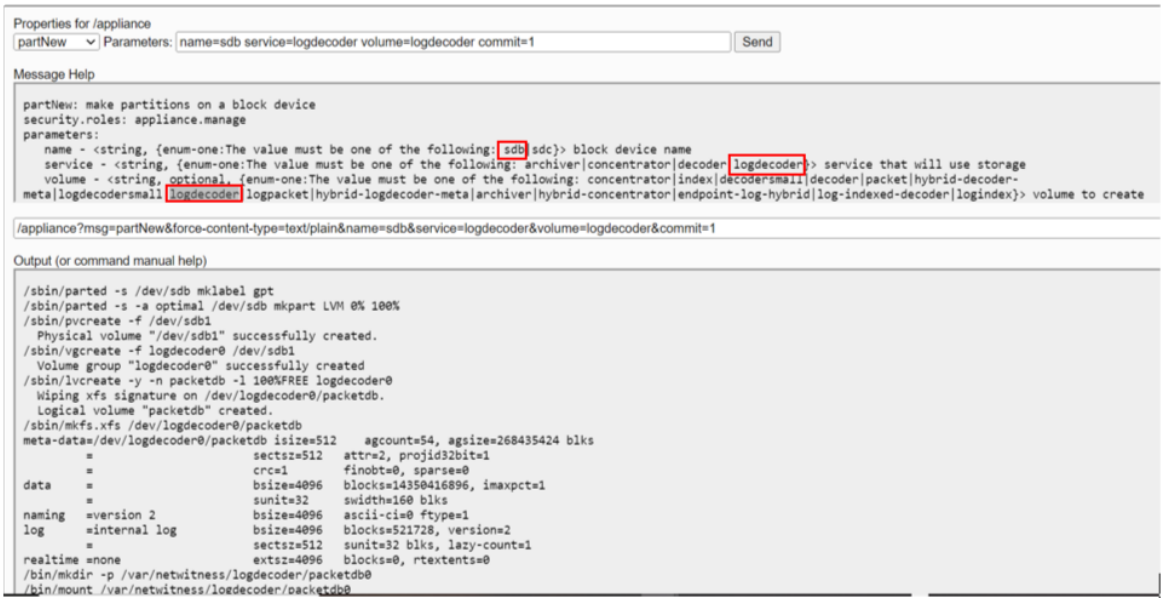
   b. Execute the `devlist` command to find the exact memory distribution between the newly created RAIDS,

   in log-hybrid the distribution is exactly equal so any service can be installed in any of the

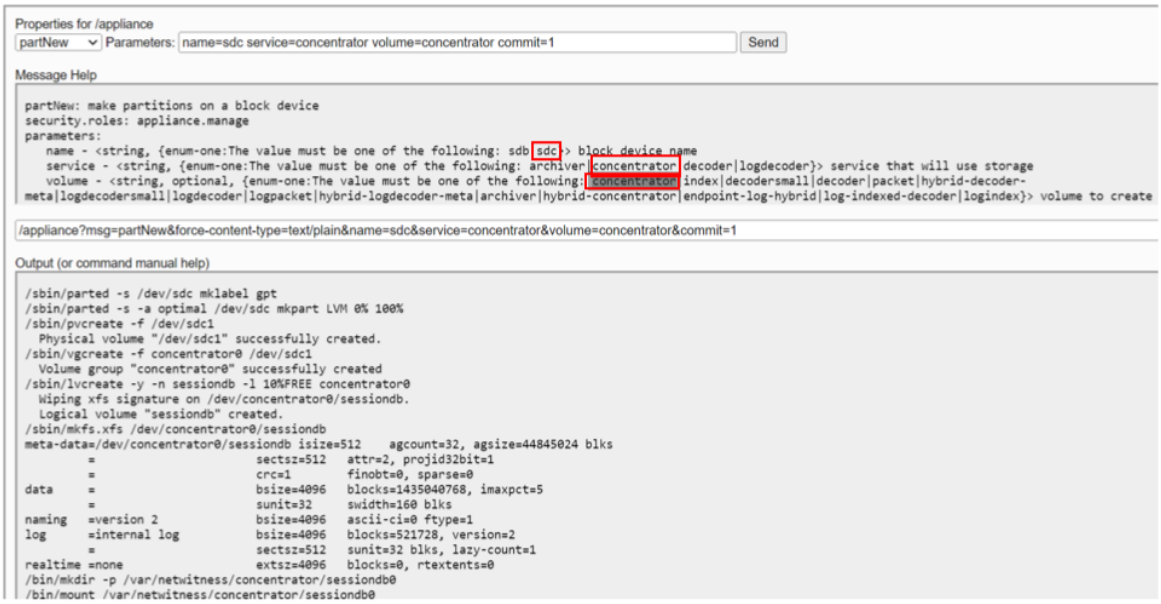
RAIDS.



3. Execute the `partNew` command with the following parameters to create partitions and mount points in the `/etc/fstab` file.
   a. `name=sdb service=logdecoder volume=logdecoder commit=1`



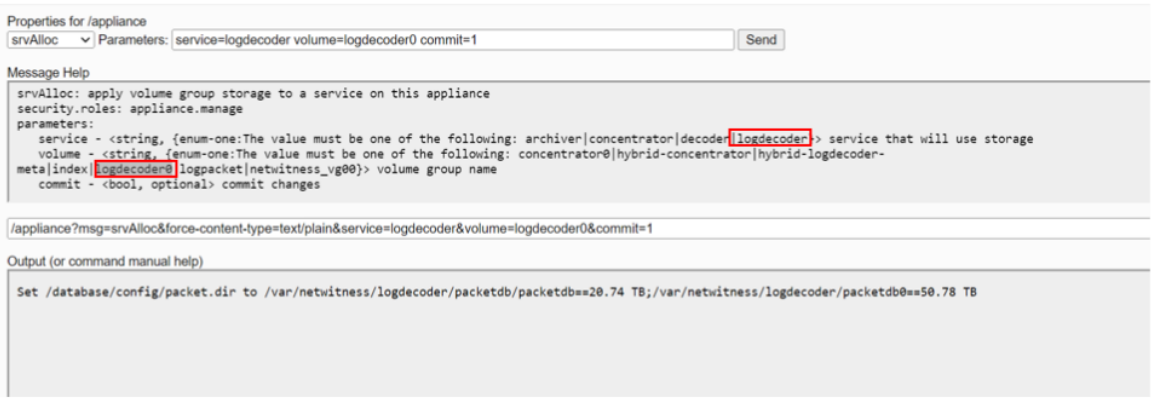
b. `name=sdc service=concentrator volume=concentrator commit=1`



4. Execute the `srvAlloc` command with the following parameters to allocate the space to logdecoder and concentrator services.

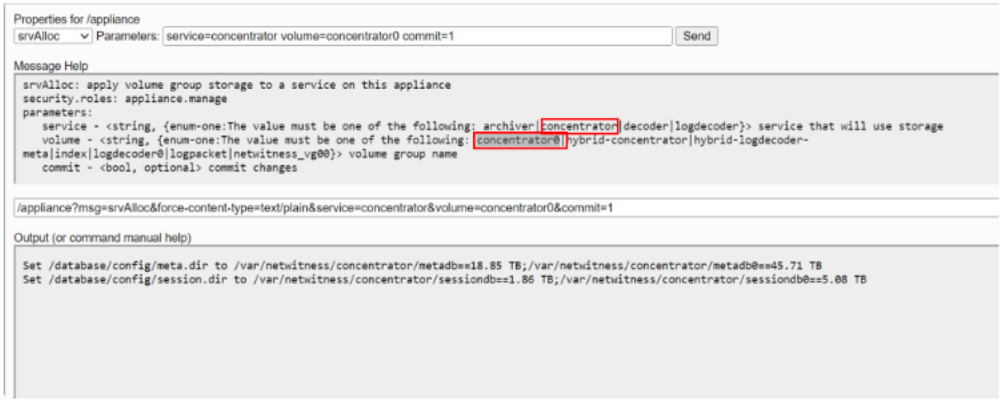
   a. This adds storage to the logdecoder service configuration and restarts the service every time it is executed.

   `service=logdecoder volume=logdecoder0 commit=1`

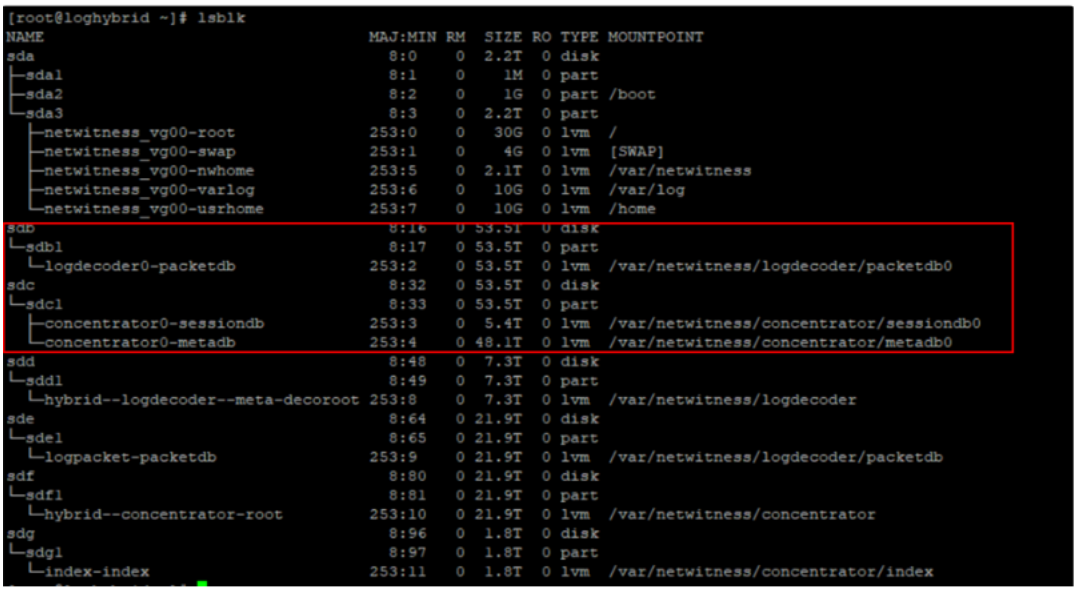


   b. This adds storage to concentrator service configuration and restarts the service every time it is executed.

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



5. Execute the `lsblk` command in backed to see all the raids and partitions inside the service.



Configure Storage for Network Hybrid using NW-PV-A/NW-PV-A-N

The following scenario configures storage on one, non-encrypted, 12-Drive PowerVault for a Network Hybrid physical host.

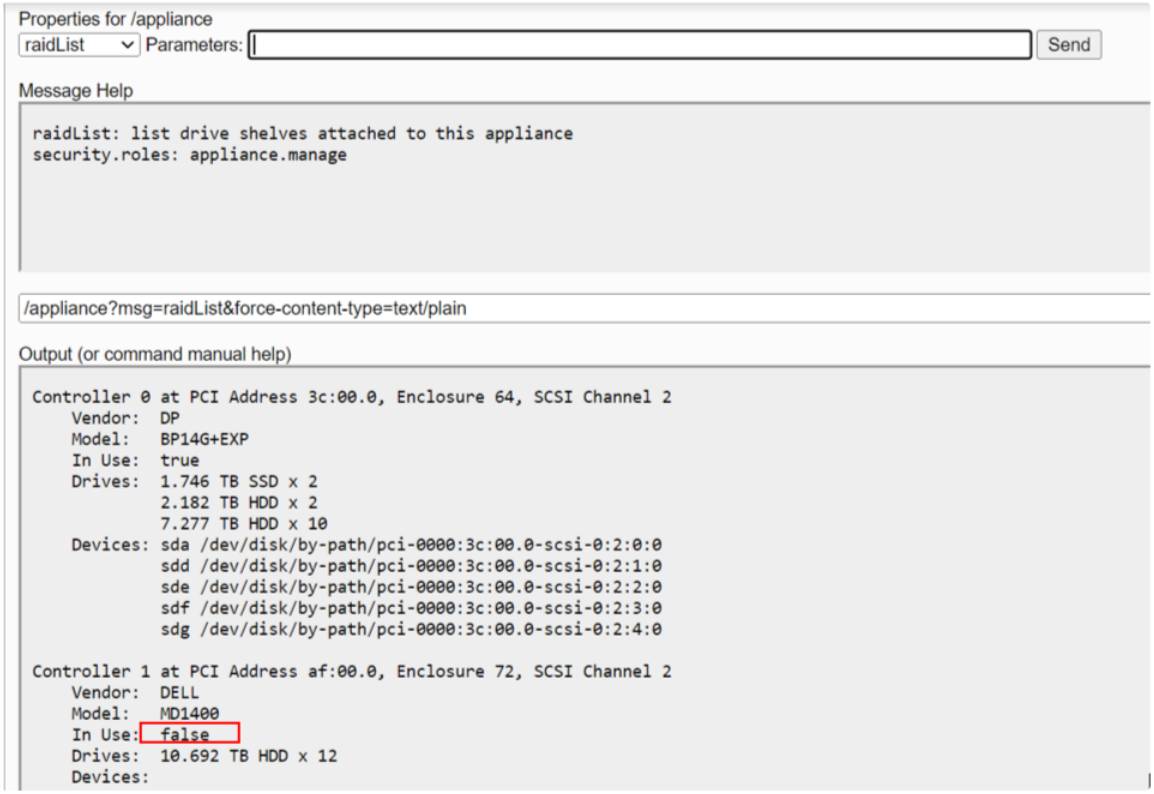
1. Execute the raidList command
   a. Record the Controller Number, Enclosure Number, In Use, Drives, and Devices. You should see the following information.

      ```
      In Use: FALSE
      ```

      ```
      Devices: <empty>
      ```

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

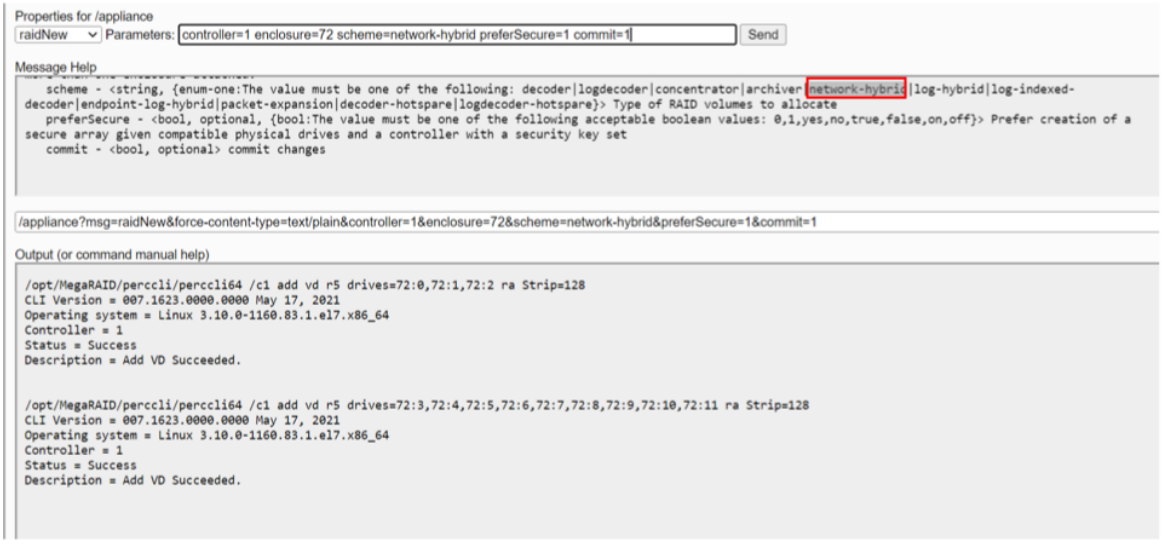
The following example illustrates what you should see before you create a RAID array.



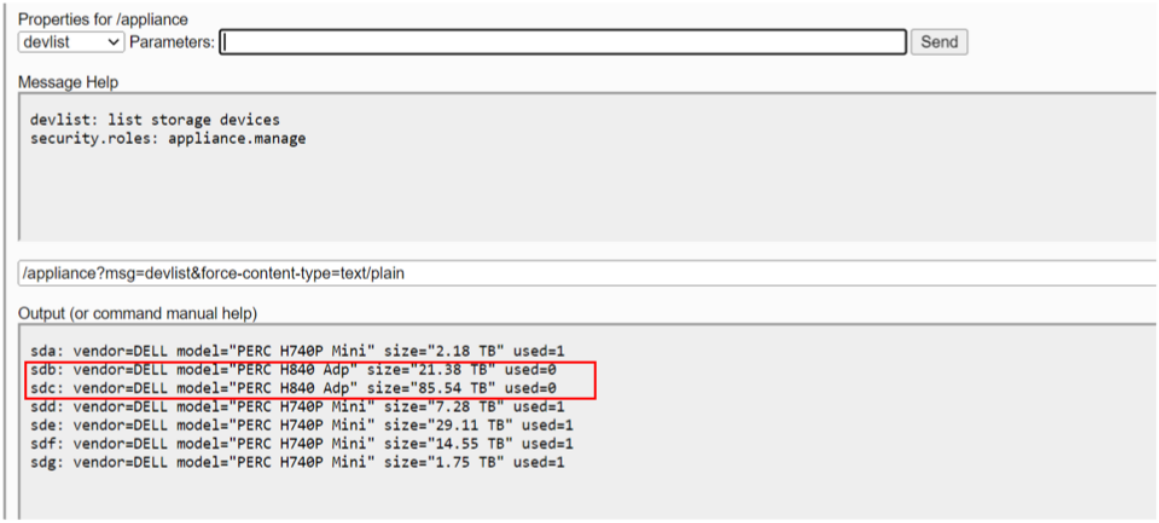
2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.

   ```
   controller=1 enclosure=72 scheme=network-hybrid preferSecure=1 commit=1
   ```

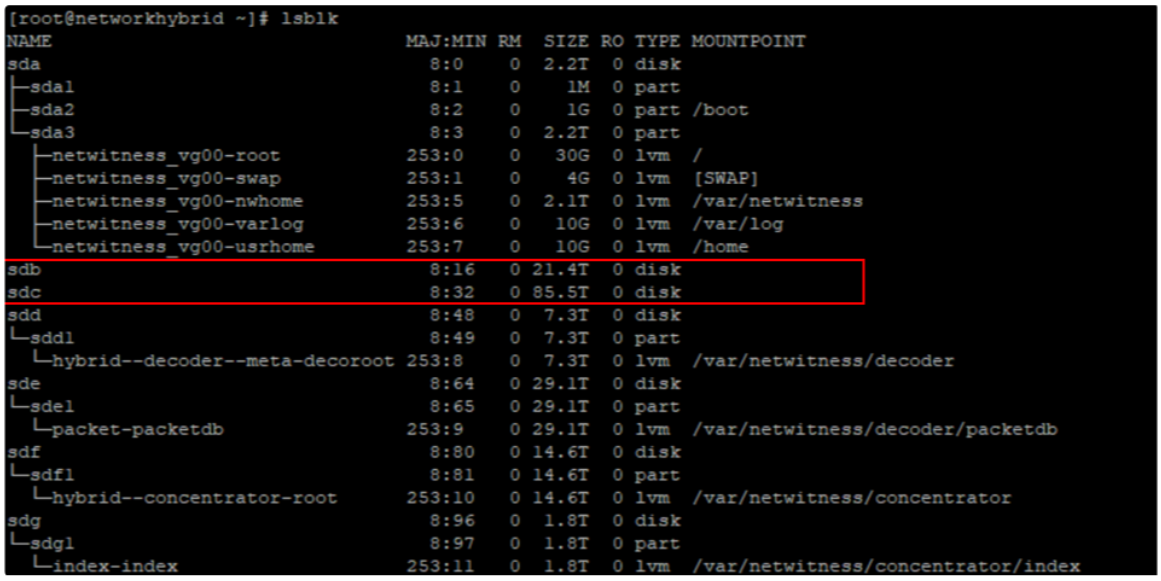
   a. The following example illustrates what you should see after you create a RAID array.



b. Execute `devlist` command to find the exact memory distribution between the newly created raids , in network hybrid the memory for ‘sdc’ raid is more than ‘sdb' raid ,therefore packetdb will be installed in the raid with higher memory allocated that is 'sdc’ .

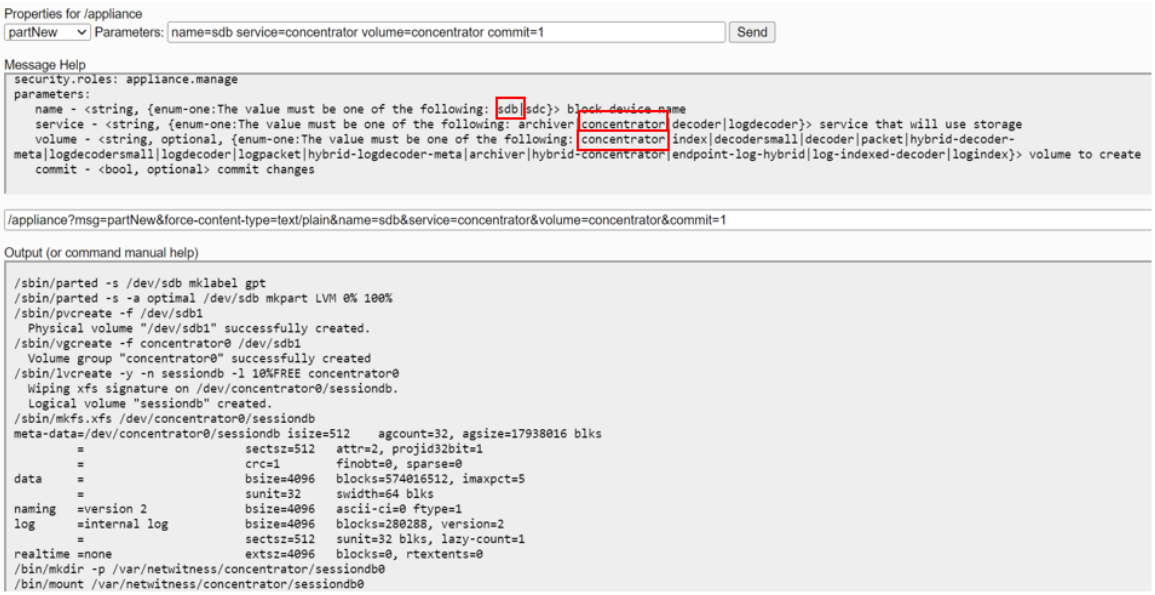


c. Execute the `lsblk` command to list all the raids that are newly created.

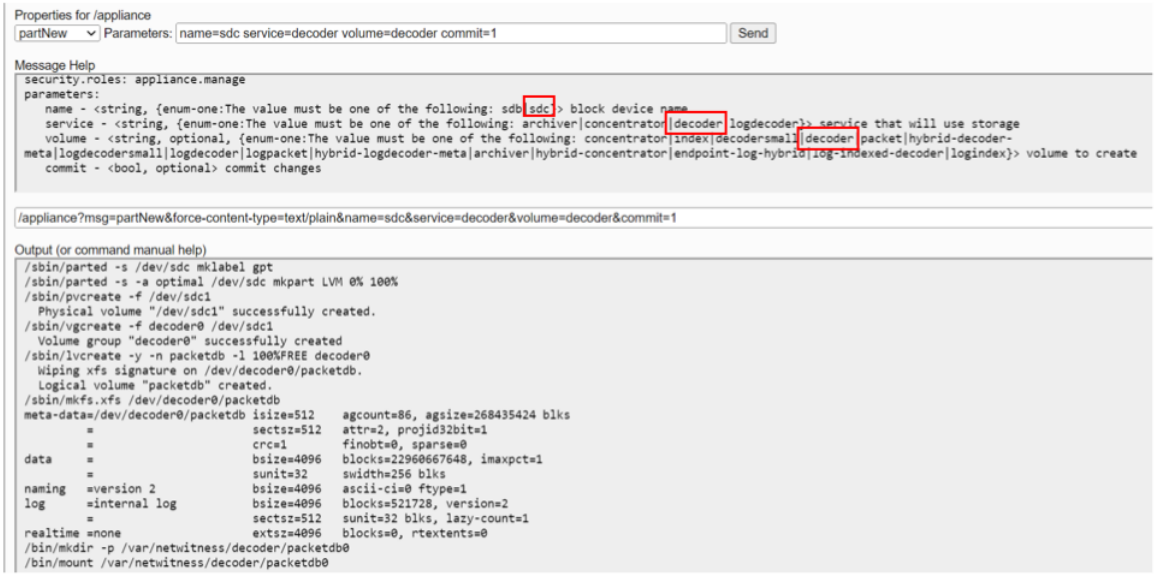


3. Execute the `partNew` command with the following parameters to create partitions and mount points in the `/etc/fstab` file.

a. `name=sdb service=concentrator volume=concentrator commit=1`



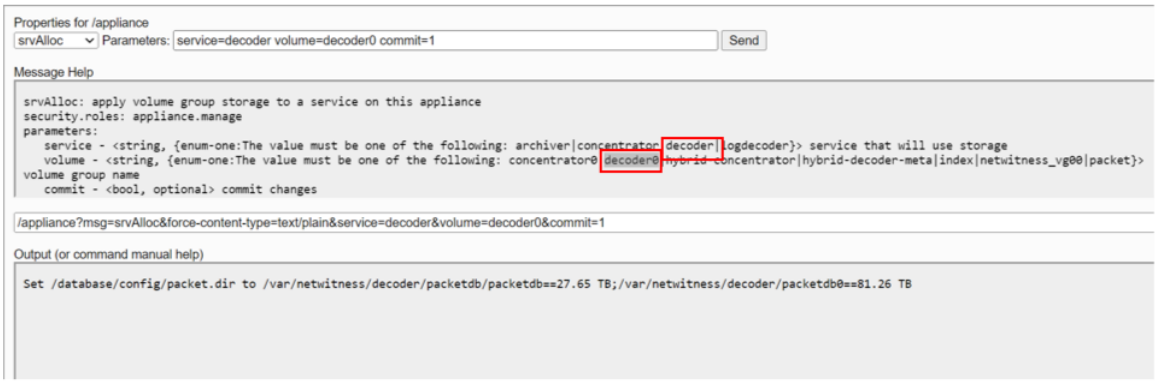
b. `name=sdc service=decoder volume=decoder commit=1`



4. Execute the `srvAlloc` command with the following parameters to allocate the space to decoder and concentrator services.

   a. This adds storage to the decoder service configuration and restarts the service every time it is executed.

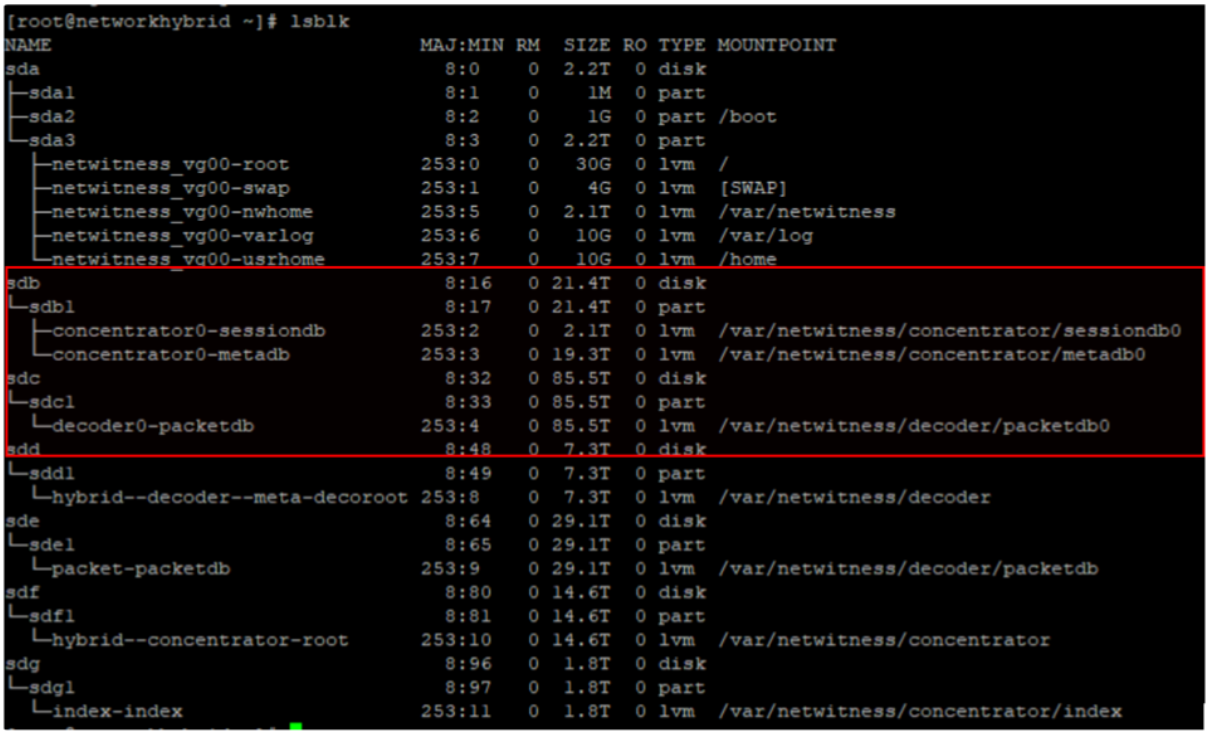
      `service=decoder volume=decoder0 commit=1`



b. This adds storage to the concentrator service configuration and restarts the service every time it is executed.

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

5. Execute the `lsblk` command to list all the raids and partition in the service.



```
[root@networkhybrid ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                   8:0    0  2.2T  0 disk
├─sda1                                8:1    0    1M  0 part
├─sda2                                8:2    0    1G  0 part /boot
└─sda3                                8:3    0  2.2T  0 part
  ├─netwitness_vg00-root            253:0    0   30G  0 lvm  /
  ├─netwitness_vg00-swap            253:1    0    4G  0 lvm  [SWAP]
  ├─netwitness_vg00-nwhome          253:5    0  2.1T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog          253:6    0   10G  0 lvm  /var/log
  └─netwitness_vg00-usrhome         253:7    0   10G  0 lvm  /home
sdb                                   8:16   0 21.4T  0 disk
└─sdb1                                8:17   0 21.4T  0 part
  ├─concentrator0-sessiondb         253:2    0  2.1T  0 lvm  /var/netwitness/concentrator/sessiondb0
  └─concentrator0-metadb            253:3    0 19.3T  0 lvm  /var/netwitness/concentrator/metadb0
sdc                                   8:32   0 85.5T  0 disk
└─sdc1                                8:33   0 85.5T  0 part
  └─decoder0-packetdb               253:4    0 85.5T  0 lvm  /var/netwitness/decoder/packetdb0
sdd                                   8:48   0  7.3T  0 disk
└─sdd1                                8:49   0  7.3T  0 part
  └─hybrid--decoder--meta-decoroot  253:8    0  7.3T  0 lvm  /var/netwitness/decoder
sde                                   8:64   0 29.1T  0 disk
└─sde1                                8:65   0 29.1T  0 part
  └─packet-packetdb                 253:9    0 29.1T  0 lvm  /var/netwitness/decoder/packetdb
sdf                                   8:80   0 14.6T  0 disk
└─sdf1                                8:81   0 14.6T  0 part
  └─hybrid--concentrator-root       253:10   0 14.6T  0 lvm  /var/netwitness/concentrator
sdg                                   8:96   0  1.8T  0 disk
└─sdg1                                8:97   0  1.8T  0 part
  └─index-index                     253:11   0  1.8T  0 lvm  /var/netwitness/concentrator/index
```

Configure Storage for Endpoint Log Hybrid using NW-PV-A/NW-PV-A-N

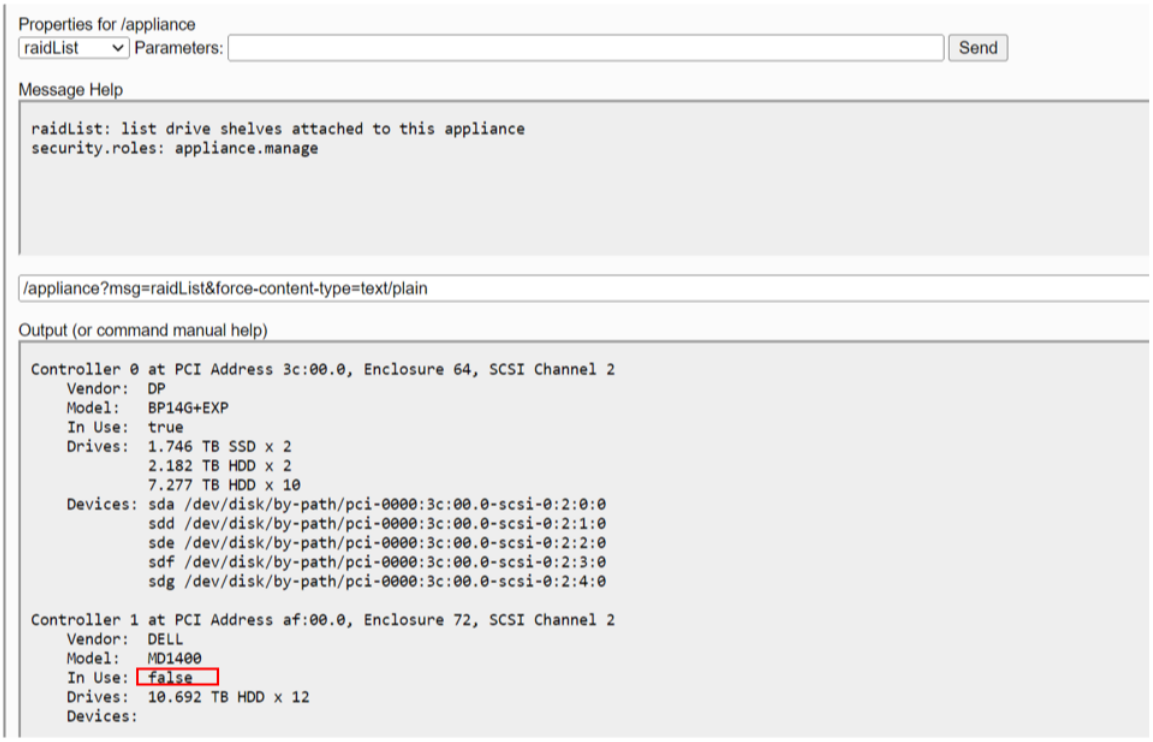
The following scenario configures storage on one, non-encrypted, 12-Drive PowerVault for a Endpoint Log Hybrid physical host.

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.

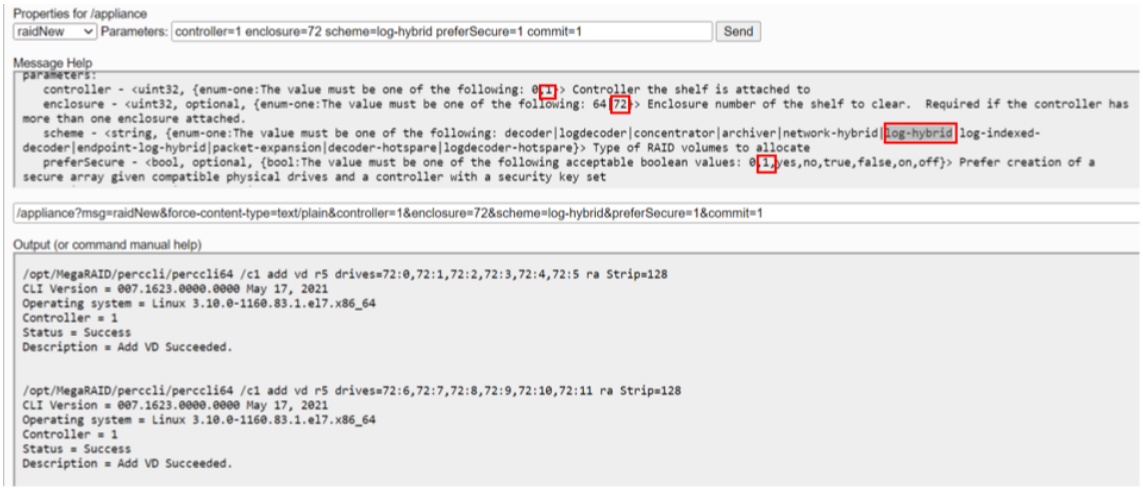
The following example illustrates what you should see before you create a RAID array.



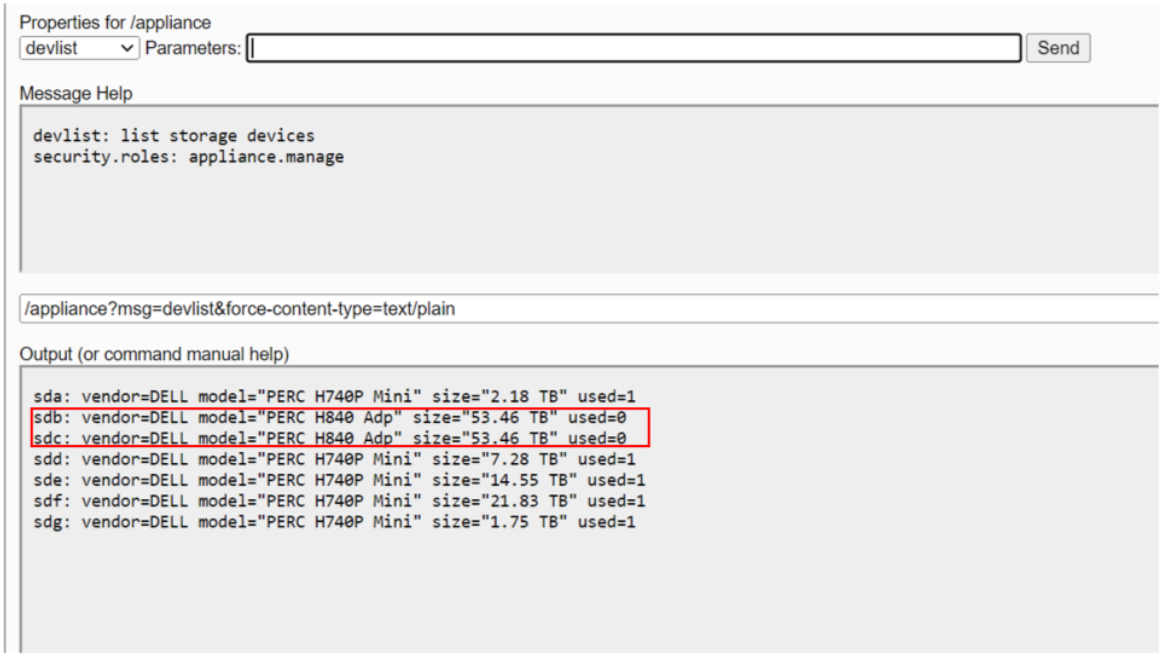
2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.

   `controller=1 enclosure=72 scheme=log-hybrid preferSecure=1 commit=1`

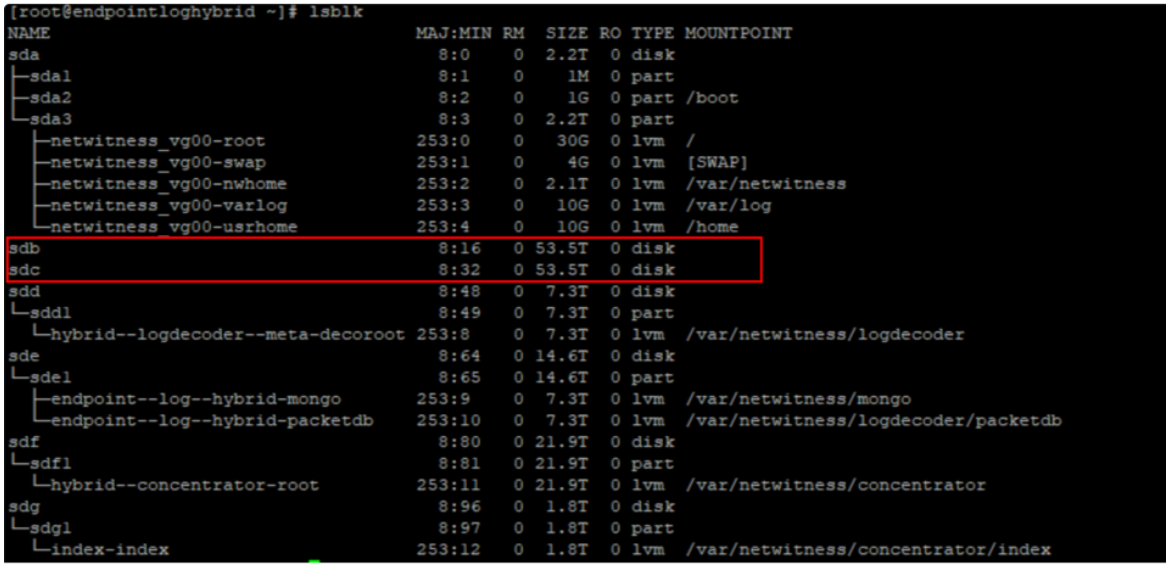
   a. The following example illustrates what you should see after you create a RAID array.



   b. Execute the `devlist` command to find the exact memory distribution for the newly created raids, in endpoint-log-hybrid both are exactly equal , therefore any service can be installed in any of the raid.

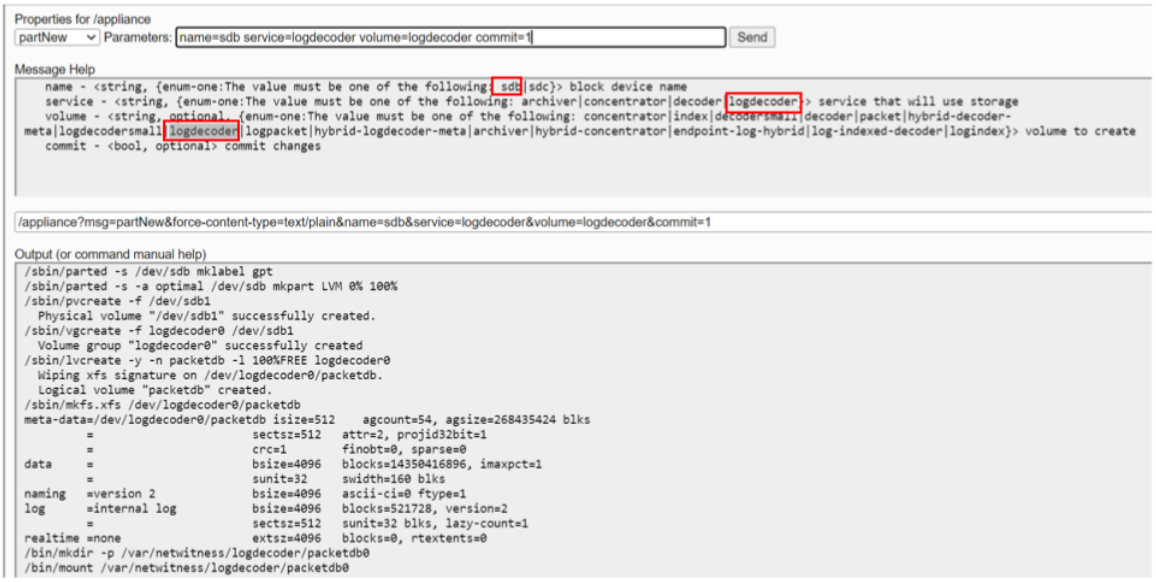


c. Execute the `lsblk` command at backend to list the newly created raids.

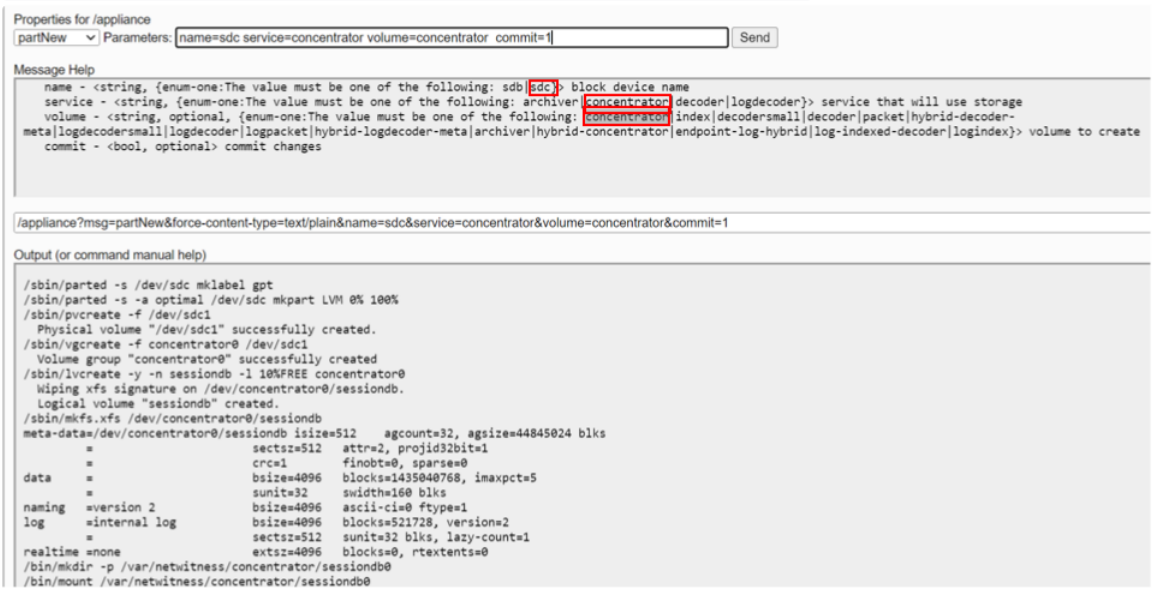


3. Execute the `partNew` command with the following parameters to create partitions and mount points in the `/etc/fstab` file.

a. `name=sdb service=logdecoder volume=logdecoder commit=1`



b. `name=sdc service=concentrator volume=concentrator commit=1`



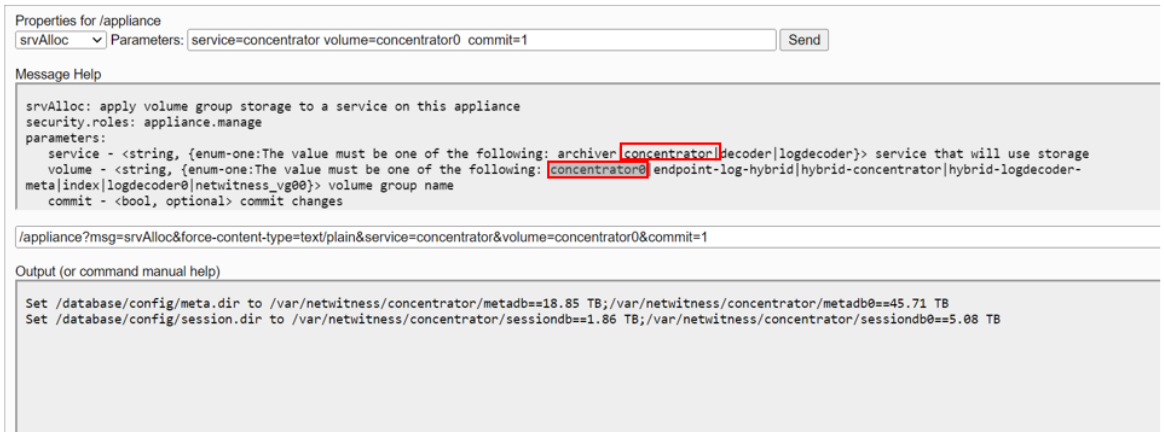
4. Execute the `srvAlloc` command with the following parameters to allocate the space to logdecoder and concentrator services.

   a. This adds storage to the logdecoder service configuration and restarts the service every time it is executed.

      `service=logdecoder volume=logdecoder0 commit=1`

b. This adds storage to the concentrator service configuration and restarts the service every time it is executed.

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



5. Execute `lsblk` command to list all the raids and partitions in the service.

```
[root@endpointloghybrid ~]# lsblk
NAME                                 MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                    8:0    0  2.2T  0 disk
├─sda1                                 8:1    0    1M  0 part
├─sda2                                 8:2    0    1G  0 part /boot
└─sda3                                 8:3    0  2.2T  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  2.1T  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 53.5T  0 disk
└─sdb1                                 8:17   0 53.5T  0 part
  └─logdecoder0-packetdb             253:5    0 53.5T  0 lvm  /var/netwitness/logdecoder/packetdb0
sdc                                    8:32   0 53.5T  0 disk
└─sdc1                                 8:33   0 53.5T  0 part
  ├─concentrator0-sessiondb          253:6    0  5.4T  0 lvm  /var/netwitness/concentrator/sessiondb0
  └─concentrator0-metadb             253:7    0 48.1T  0 lvm  /var/netwitness/concentrator/metadb0
sdd                                    8:48   0  7.3T  0 disk
└─sdd1                                 8:49   0  7.3T  0 part
  └─hybrid--logdecoder--meta-decoroot 253:8    0  7.3T  0 lvm  /var/netwitness/logdecoder
sde                                    8:64   0 14.6T  0 disk
└─sde1                                 8:65   0 14.6T  0 part
  ├─endpoint--log--hybrid-mongo      253:9    0  7.3T  0 lvm  /var/netwitness/mongo
  └─endpoint--log--hybrid-packetdb   253:10   0  7.3T  0 lvm  /var/netwitness/logdecoder/packetdb
sdf                                    8:80   0 21.9T  0 disk
└─sdf1                                 8:81   0 21.9T  0 part
  └─hybrid--concentrator-root        253:11   0 21.9T  0 lvm  /var/netwitness/concentrator
sdg                                    8:96   0  1.8T  0 disk
└─sdg1                                 8:97   0  1.8T  0 part
  └─index-index                      253:12   0  1.8T  0 lvm  /var/netwitness/concentrator/index
```

Appendix F. Sample Storage Configuration Scenarios for S7 Physical Hosts with 12-Drive PowerVault MD2412

This appendix illustrates the following examples of how to configure storage on unencrypted MD2412 PowerVault (12 * 16 TB Self Encrypting Drives (SEDs)) external storage devices attached to S7 physical hosts.

- Configure Storage for Archiver using MD2412
- Configure Storage for Decoder using MD2412
- Configure Storage for Concentrator using MD2412
- Configure Storage for Log Hybrid using MD2412
- Configure Storage for Network Hybrid using MD2412
- Configure Storage for Endpoint Log Hybrid using MD2412

Note:
The Enable Encryption on Series 6E or Series 7 Hosts topic describes the process to turn on encryption on NW servers and configured storage devices (MD2412).

If the user chooses to enable or turn on encryption before configuring storage (MD2412), the security on the controllers (internal and external) must be enabled and controller security key must be set. To enable the security on the controllers (internal and external) and set the controller security key, refer to *Chapter 8 ->Security Key and RAID Management ->Create a security key* in the *Dell PowerEdge RAID Controller 12 User's Guide*.

Once the security key is set, to enable encryption on the external storage, use `preferSecure=1` when executing raidNew command to create RAID.

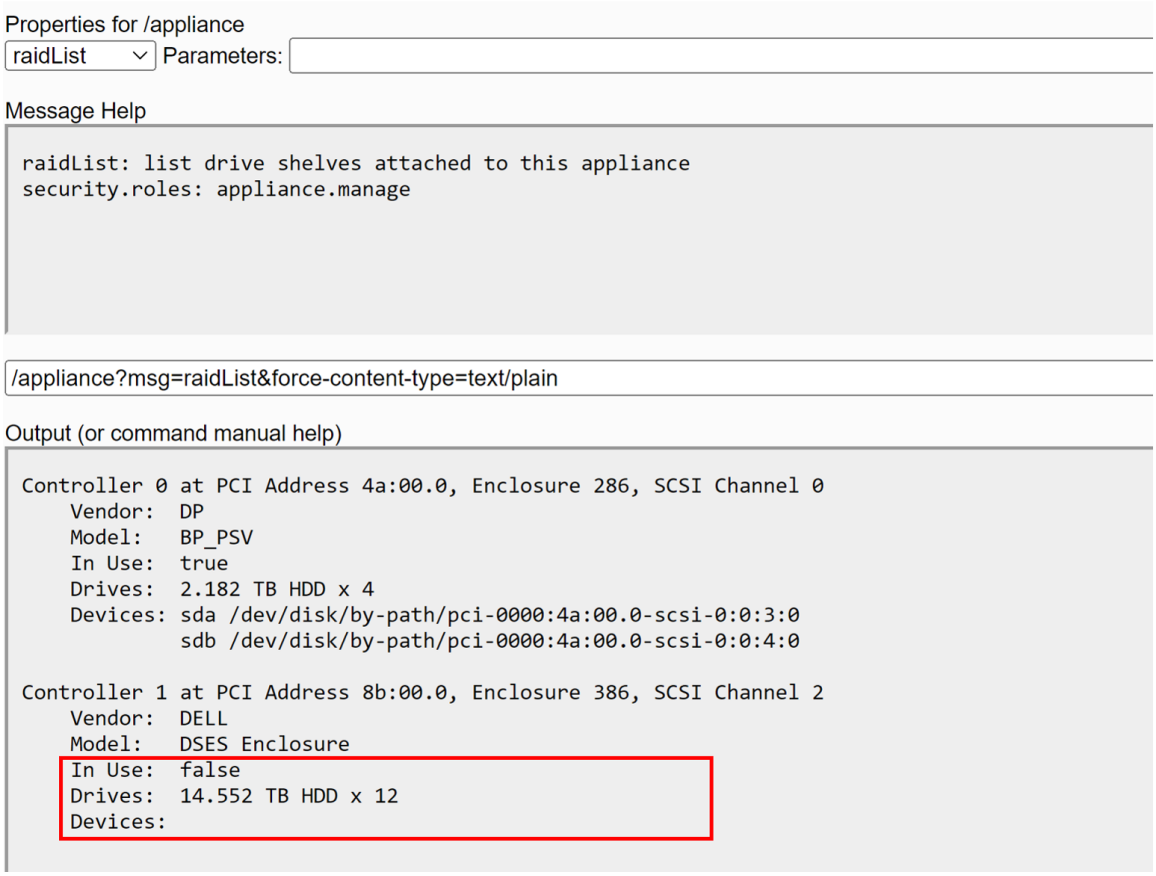
Alternatively, configure the external storage with `preferSecure=0` when creating RAID and then follow the steps in Enable Encryption on Series 6E or Series 7 Hosts to turn on encryption.

Configure Storage for Archiver using MD2412

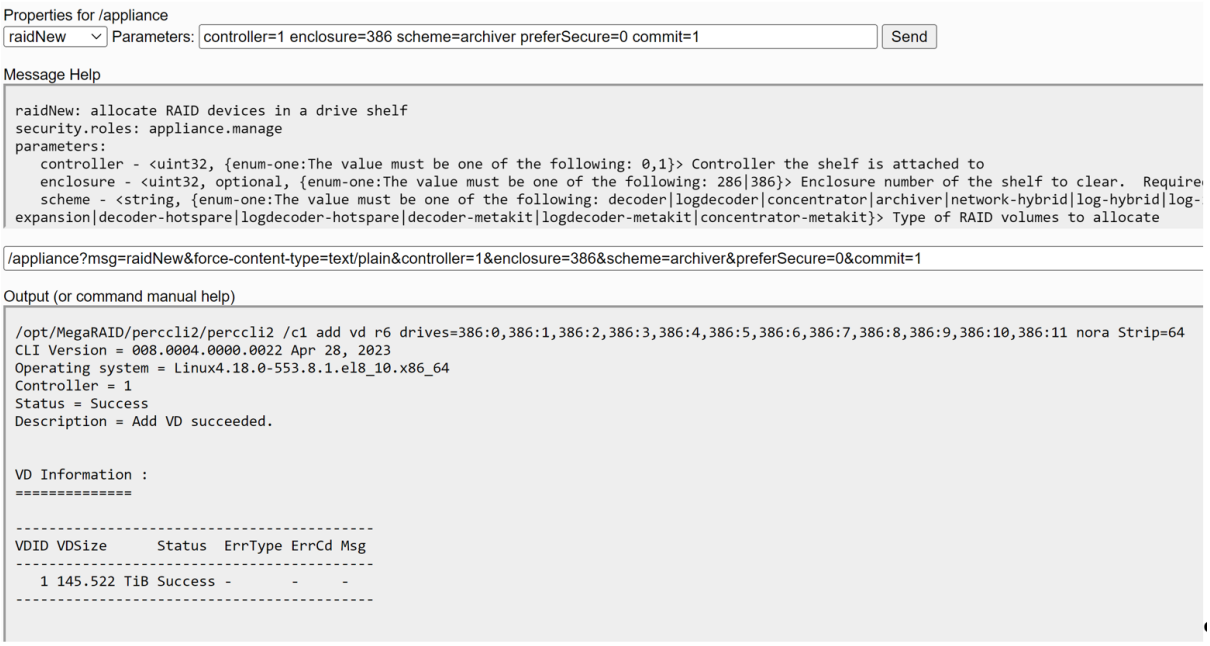
The following scenario configures storage on one, unencrypted, 12-Drive MD2412 PowerVault for an Archiver physical host.

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>
      ```

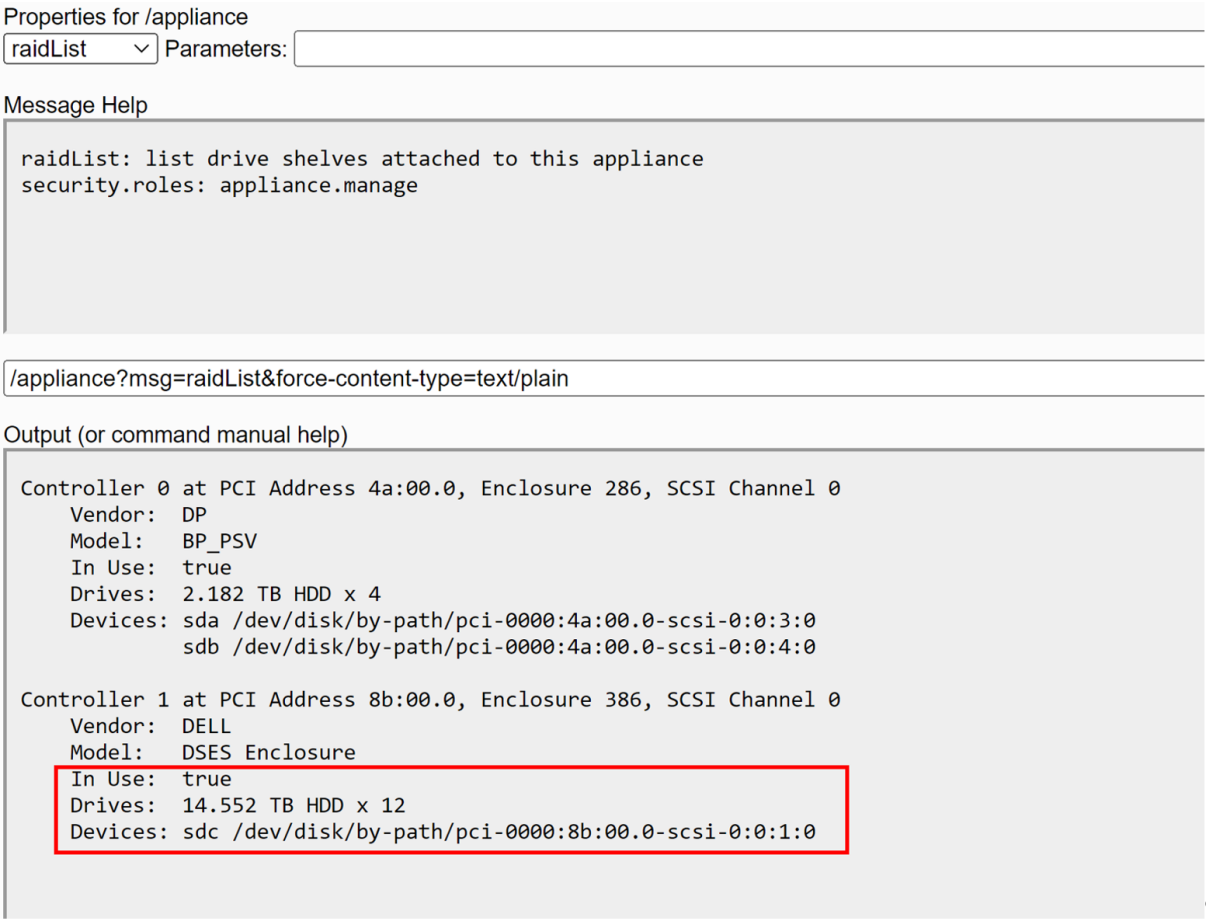
b. Verify the Drive Count, Size, and Vendor.
The following example illustrates what you should see before you create a RAID array.



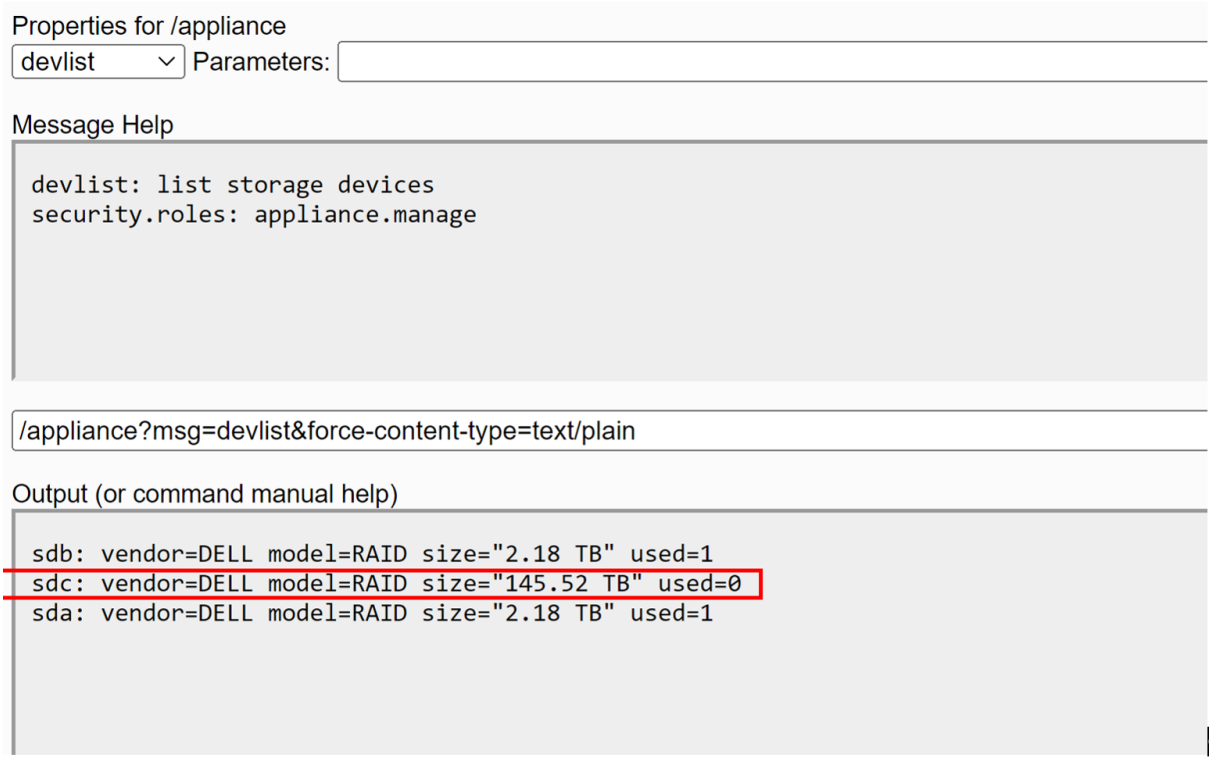
2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
`controller=1 enclosure=386 scheme=archiver preferSecure=0 commit=1`
The following example illustrates what you should see after you create a RAID array.



3. Execute the `raidList` command to verify the new RAID array.
   `In Use: TRUE`
   `Devices: <device>(for example, sdc)`



4. Execute the `devlist` command to find the exact memory distribution for the newly created raids.



5. Execute the `lsblk` command at backend to list the newly created raids.

```
[root@S7Archiver ~]# lsblk
NAME                          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                             8:0    0   2.2T  0 disk
├─sda1                          8:1    0   256M  0 part /boot/efi
├─sda2                          8:2    0     1G  0 part /boot
└─sda3                          8:3    0   2.2T  0 part
  ├─netwitness_vg00-nwhome    253:0    0   4.3T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog    253:1    0    10G  0 lvm  /var/log
  ├─netwitness_vg00-usrhome   253:2    0    10G  0 lvm  /home
  ├─netwitness_vg00-root      253:3    0    30G  0 lvm  /
  └─netwitness_vg00-swap      253:4    0     4G  0 lvm  [SWAP]
sdb                             8:16   0   2.2T  0 disk
└─sdb1                          8:17   0   2.2T  0 part
  └─netwitness_vg00-nwhome    253:0    0   4.3T  0 lvm  /var/netwitness
sdc                             8:32   0 145.5T  0 disk
└─sdc1                          8:33   0  72.8T  0 part
```

6. Execute the `partNew` command with the following parameters to create partitions and mount points as required.

   `name=sdc service=archiver volume=archiver commit=1`

Properties for /appliance

partNew Parameters: name=sdc service=archiver volume=archiver commit=1 Send

Message Help

```
partNew: make partitions on a block device
security.roles: appliance.manage
parameters:
   name - <string, {enum-one:The value must be one of the following: sdc}> block device name
   service - <string, {enum-one:The value must be one of the following: archiver|concentrator|decoder|logdecoder}> service that w
   volume - <string, optional, {enum-one:The value must be one of the following: concentrator|index|decodersmall|decoder|packet|h
logdecoder-meta|archiver|hybrid-concentrator|endpoint-log-hybrid|log-indexed-decoder|logindex}> volume to create
```

/appliance?msg=partNew&force-content-type=text/plain&name=sdc&service=archiver&volume=archiver&commit=1

Output (or command manual help)

```
/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 archiver0 /dev/sdc1
  Volume group "archiver0" successfully created
/sbin/lvcreate -y -n database -l 100%FREE archiver0
  Logical volume "database" created.
/sbin/mkfs.xfs /dev/archiver0/database
meta-data=/dev/archiver0/database isize=512    agcount=146, agsize=268435440 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0 inobtcount=0
data     =                       bsize=4096   blocks=39063387136, imaxpct=1
         =                       sunit=16     swidth=160 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=521728, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/archiver/database0
/bin/mount /var/netwitness/archiver/database0
```

7. Execute `devlist` command to verify partitions created.

Properties for /appliance

devlist Parameters:

Message Help

```
devlist: list storage devices
security.roles: appliance.manage
```

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

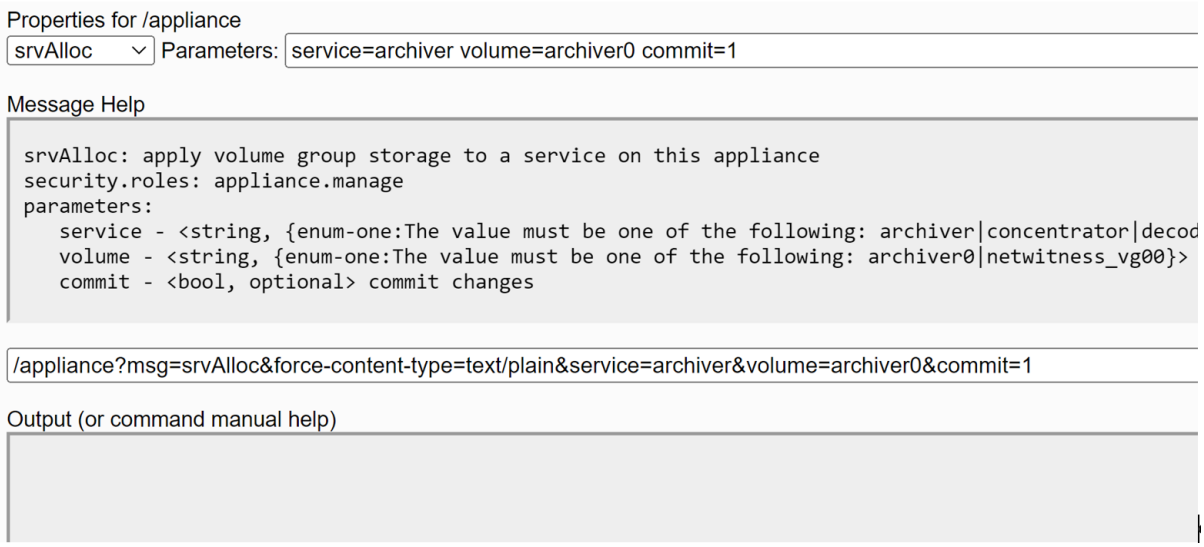
Output (or command manual help)

```
sdb: vendor=DELL model=RAID size="2.18 TB" used=1
sdc: vendor=DELL model=RAID size="145.52 TB" used=1
sda: vendor=DELL model=RAID size="2.18 TB" used=1
```

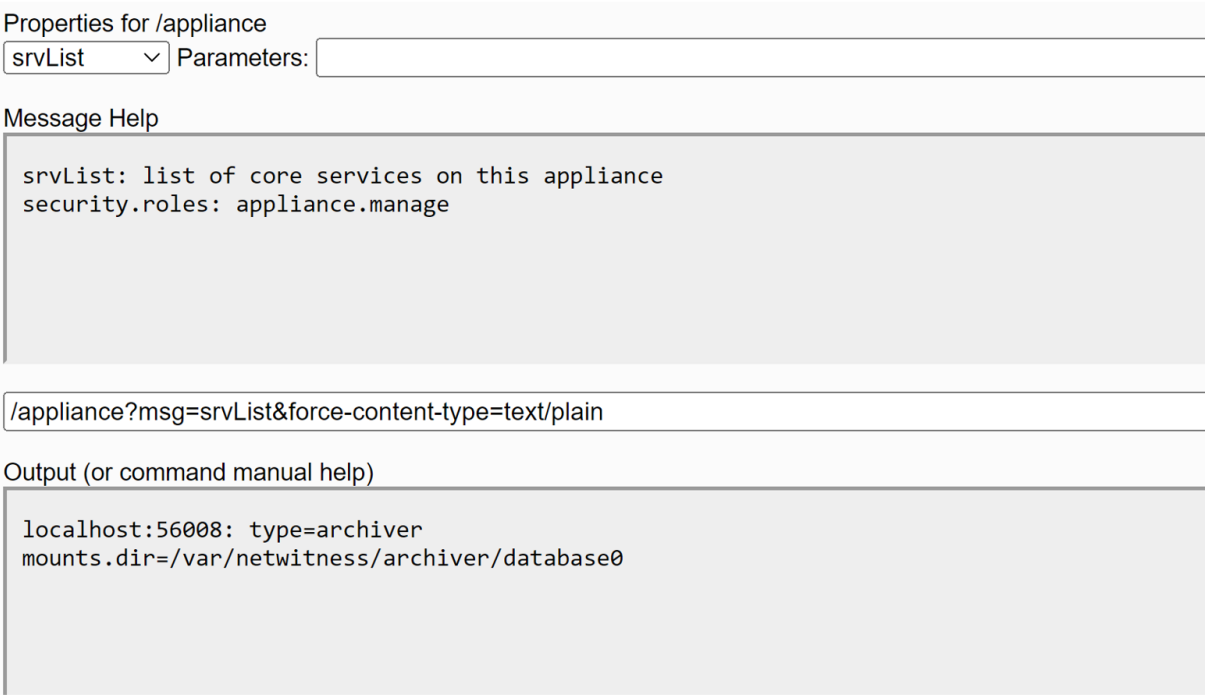
8. Execute the `srvAlloc` command with the following parameters to allocate the space to archiver service.

   This adds storage to the archiver service configuration and restarts the service every time it is executed.

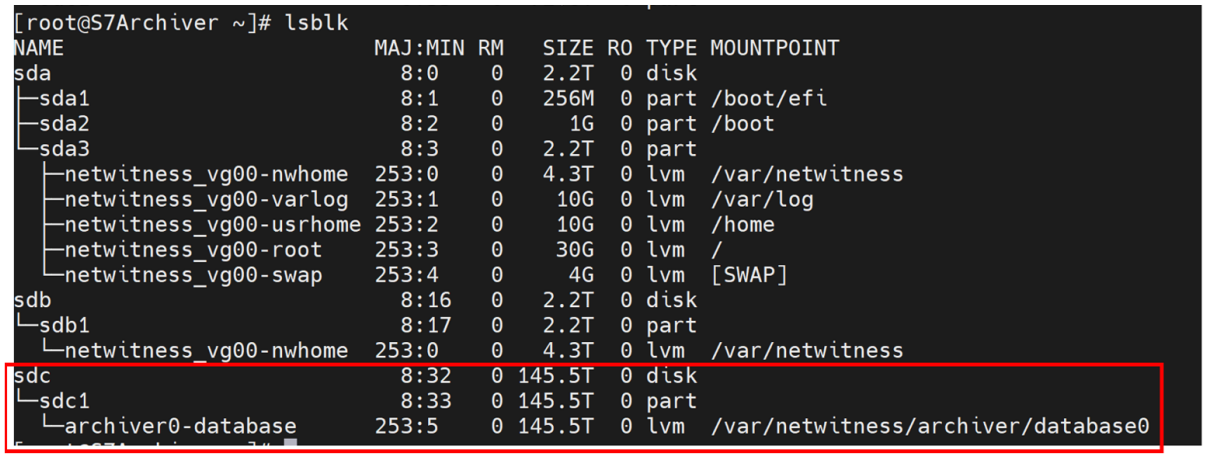
   ```
   service=archiver volume=archiver0 commit=1
   ```



9. Execute the `srvList` command to verify the services allocated on the appliance.



10. Execute `lsblk` command to list all the raids and partitions in the service.



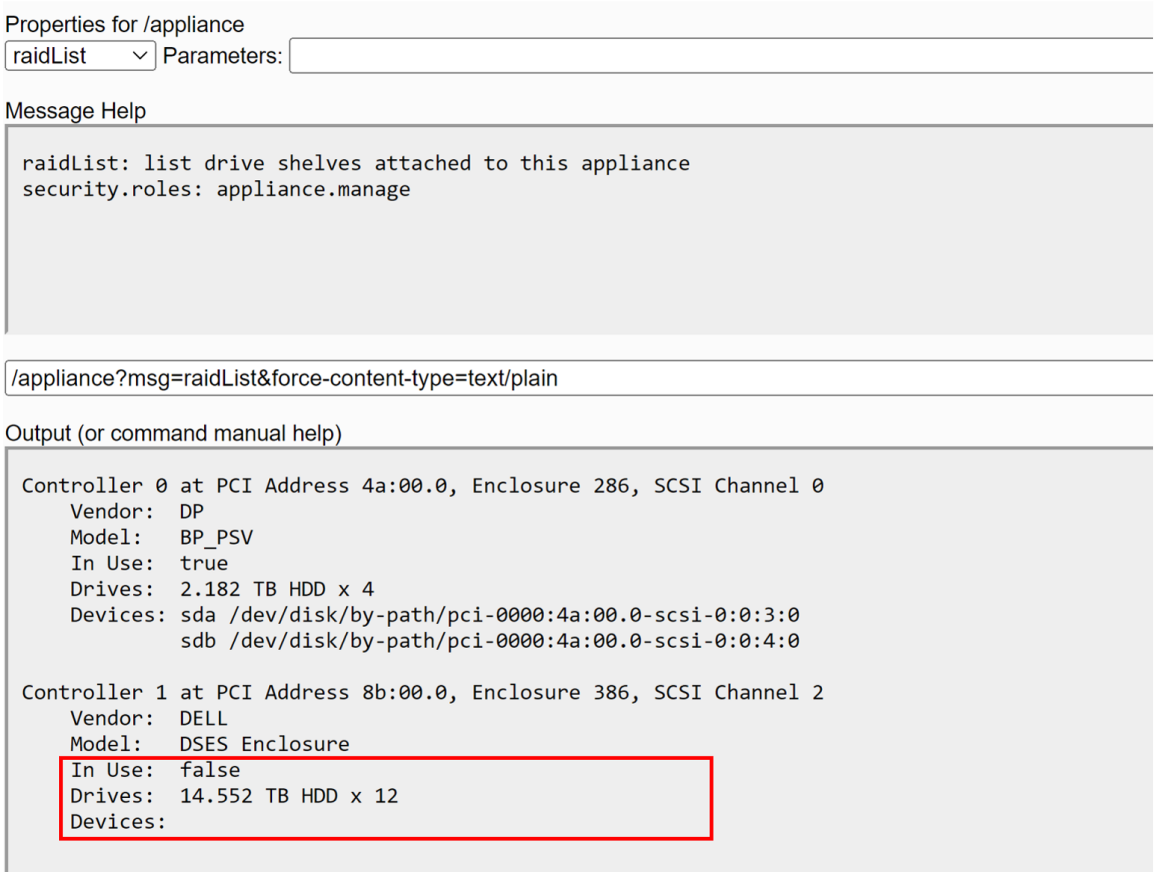
Configure Storage for Decoder using MD2412

The following scenario configures storage on one, unencrypted, 12-Drive MD2412 PowerVault for a Network Decoder physical host..

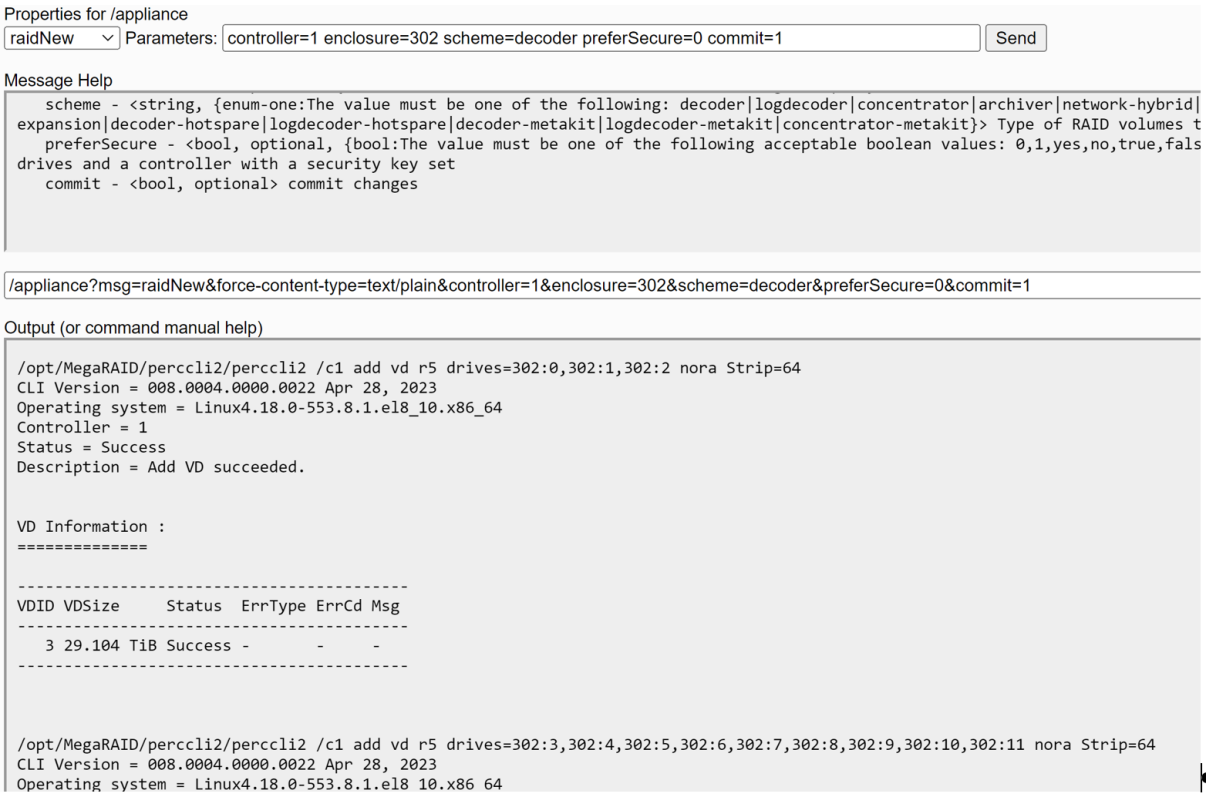
1. Execute the `raidList` command.

   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.



2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
`controller=1 enclosure=302 scheme=decoder preferSecure=0 commit=1`

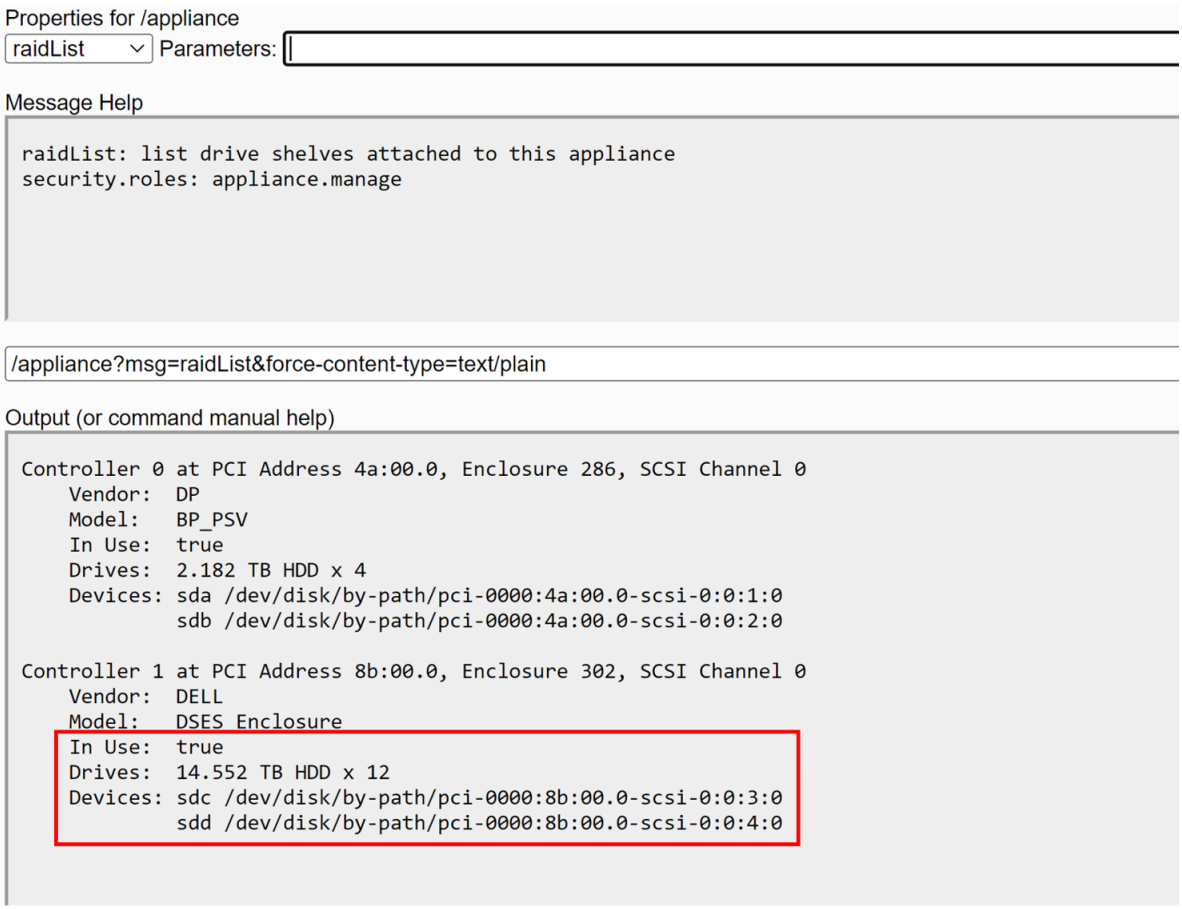


3. Execute the `raidList` command to verify the new RAID array.

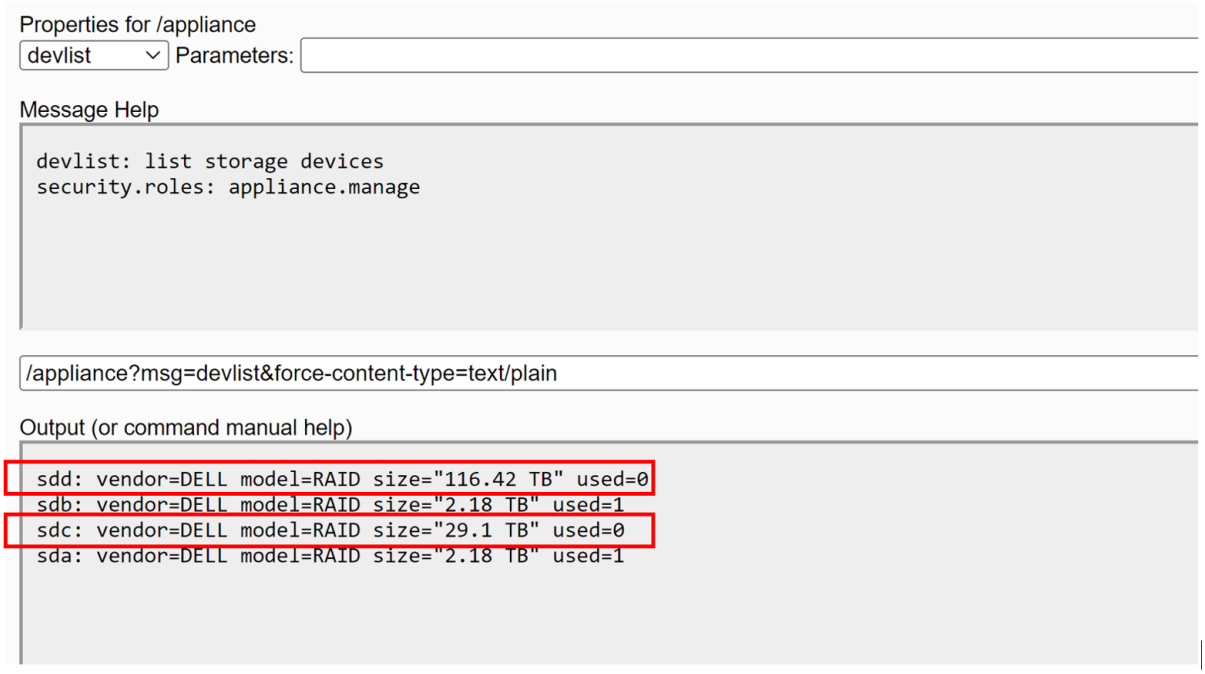
   You should now see the following information.

   `In Use: TRUE`

   `Devices: <devices>(for example, sdc, sdd)`



4. Execute the `devlist` command to find the exact memory distribution for the newly created raids..



5. Execute `lsblk` command at the backend to verify block devices created on the host.

```
[root@S7Decoder ~]# lsblk
NAME                          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                             8:0    0   2.2T  0 disk
├─sda1                          8:1    0   256M  0 part /boot/efi
├─sda2                          8:2    0     1G  0 part /boot
└─sda3                          8:3    0   2.2T  0 part
  ├─netwitness_vg00-nwhome    253:0    0   4.3T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog    253:1    0    10G  0 lvm  /var/log
  ├─netwitness_vg00-usrhome   253:2    0    10G  0 lvm  /home
  ├─netwitness_vg00-root      253:3    0    30G  0 lvm  /
  └─netwitness_vg00-swap      253:4    0     4G  0 lvm  [SWAP]
sdb                             8:16   0   2.2T  0 disk
└─sdb1                          8:17   0   2.2T  0 part
  └─netwitness_vg00-nwhome    253:0    0   4.3T  0 lvm  /var/netwitness
sdc                             8:32   0  29.1T  0 disk
sdd                             8:48   0 116.4T  0 disk
```

6. Execute the `partNew` command to create the decodersmall partition first (decoder dir, index, metadb, sessiondb) on the smaller block device (sdc) with the following parameters.

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

Properties for /appliance

partNew Parameters: name=sdc service=decoder volume=decodersmall commit=1

Message Help

```
partNew: make partitions on a block device
security.roles: appliance.manage
parameters:
   name - <string, {enum-one:The value must be one of the following: sdd|sdc}> block device name
   service - <string, {enum-one:The value must be one of the following: archiver|concentrator|deco
   volume - <string, optional, {enum-one:The value must be one of the following: concentrator|inde
logdecoder-meta|archiver|hybrid-concentrator|endpoint-log-hybrid|log-indexed-decoder|logindex}> vo
```

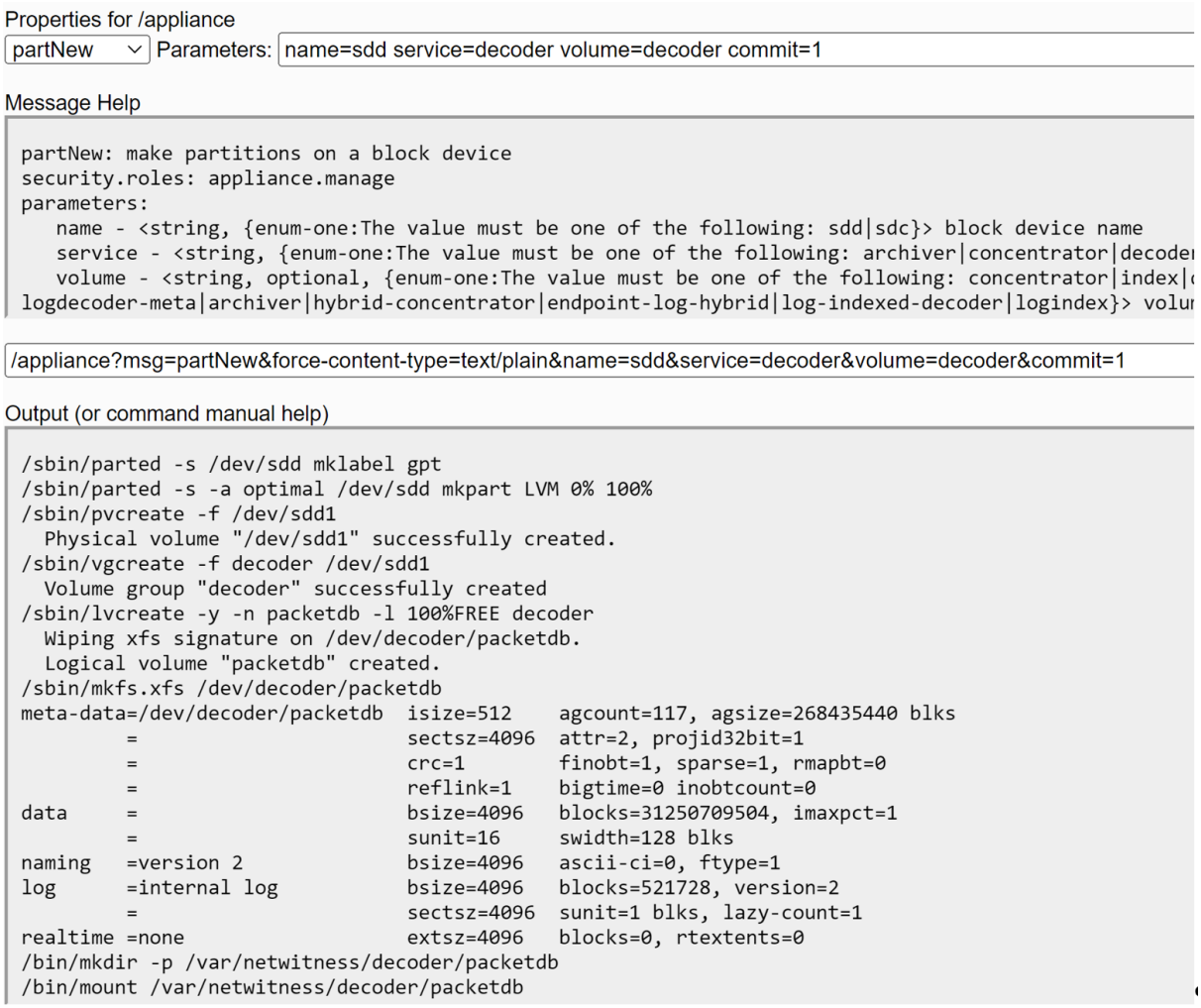
/appliance?msg=partNew&force-content-type=text/plain&name=sdc&service=decoder&volume=decodersmall&commit=

Output (or command manual help)

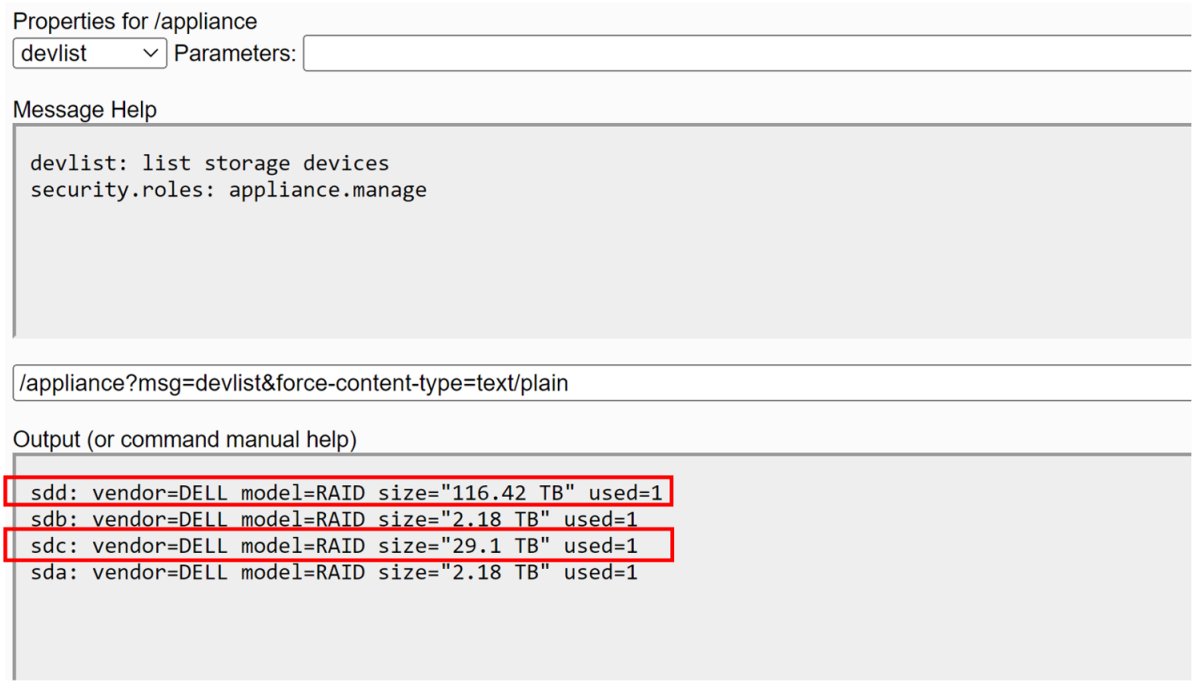
```
/sbin/parted -s /dev/sdc mklabel gpt
/sbin/parted -s -a optimal /dev/sdc mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdc1
  Physical volume "/dev/sdc1" successfully created.
/sbin/vgcreate -f decodersmall /dev/sdc1
  Volume group "decodersmall" successfully created
/sbin/lvcreate -y -n decoroot -L 20G decodersmall
  Wiping xfs signature on /dev/decodersmall/decoroot.
  Logical volume "decoroot" created.
/sbin/mkfs.xfs /dev/decodersmall/decoroot
meta-data=/dev/decodersmall/decoroot isize=512    agcount=16, agsize=327664 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0 inobtcount=0
data     =                       bsize=4096   blocks=5242624, imaxpct=25
         =                       sunit=16     swidth=32 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/decoder
/bin/mount /var/netwitness/decoder
```

7. Execute the `partNew` command to create the decoder volume (packetdb) on the bigger block device of the two created (sdd) with the following parameters.

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

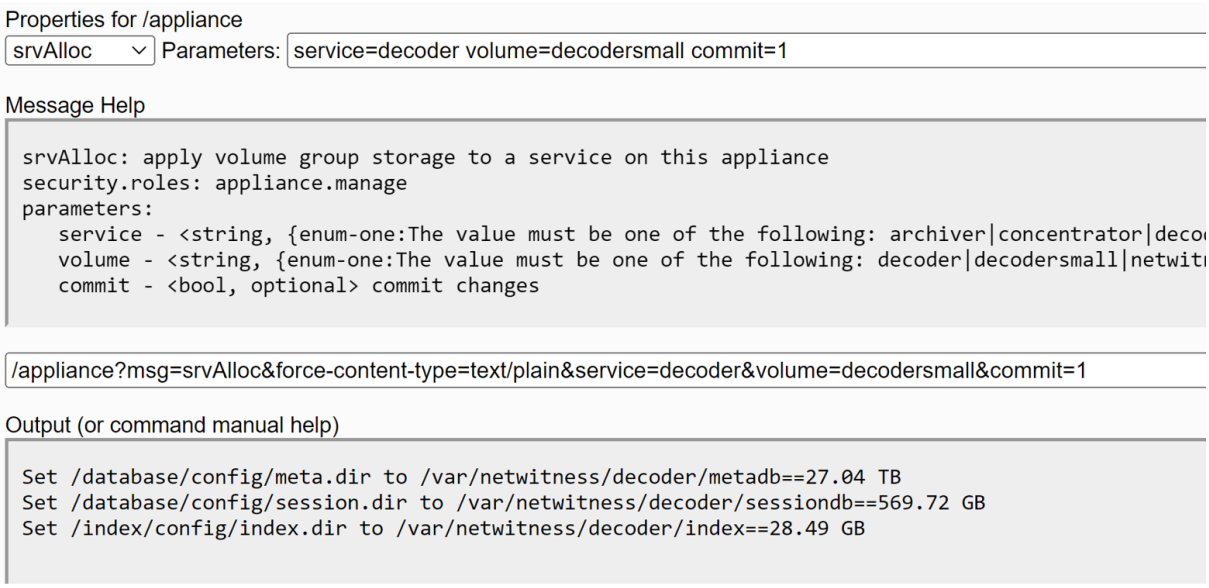


8. Execute `devlist` command to verify partitions created.

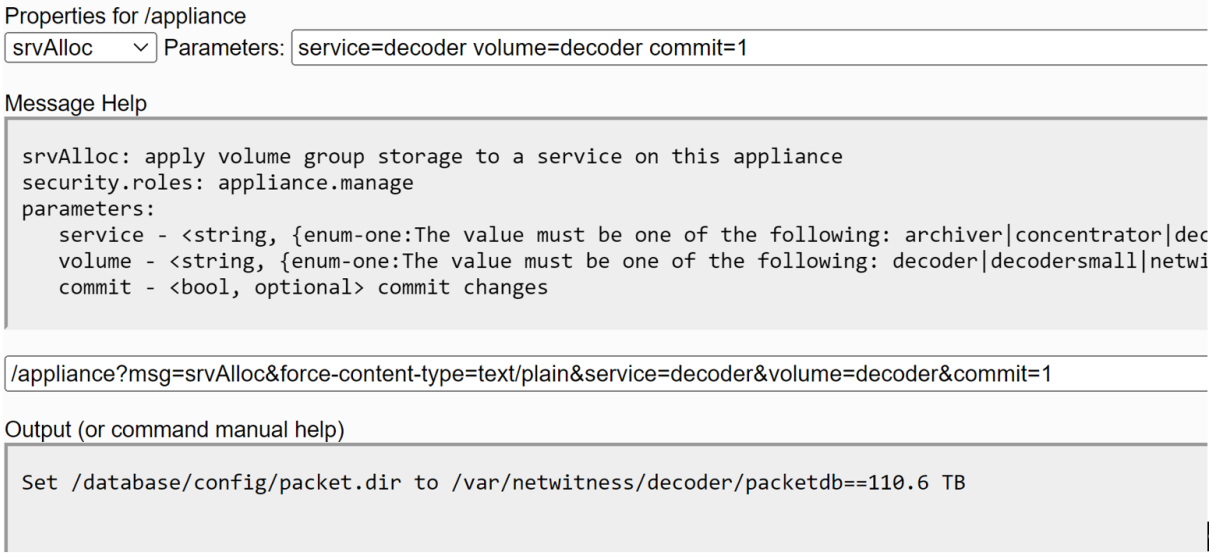


9. Execute the `srvAlloc` command with the following parameters to add the storage information into the Service Configuration settings. Service is restarted each time the command is executed.

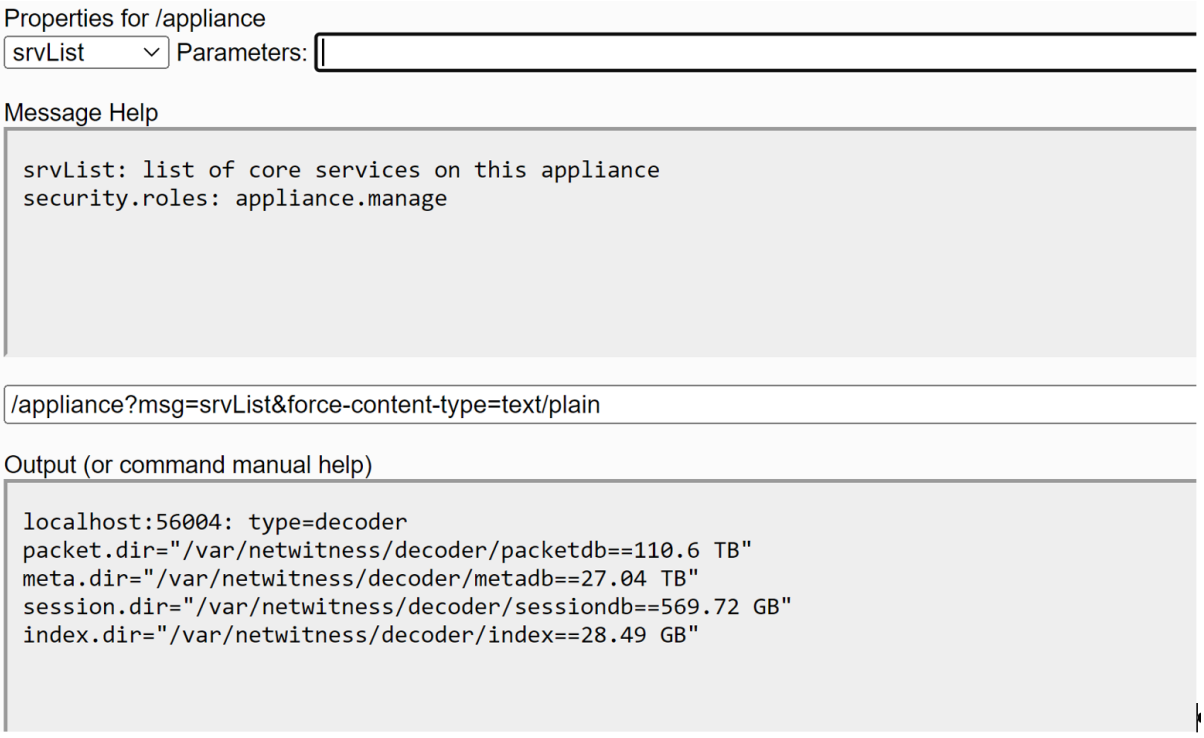
```
service=decoder volume=decodersmall commit=1
```




```
service=decoder volume=decoder commit=1
```



10. Execute the `srvList` command to verify the services allocated on the appliance.



11. Execute `lsblk` command to list all the raids and partitions in the service.

```
[root@S7Decoder ~]# lsblk
NAME                          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                             8:0    0   2.2T  0 disk
├─sda1                          8:1    0   256M  0 part /boot/efi
├─sda2                          8:2    0     1G  0 part /boot
└─sda3                          8:3    0   2.2T  0 part
  ├─netwitness_vg00-nwhome    253:5    0   4.3T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog    253:6    0    10G  0 lvm  /var/log
  ├─netwitness_vg00-usrhome   253:7    0    10G  0 lvm  /home
  ├─netwitness_vg00-root      253:8    0    30G  0 lvm  /
  └─netwitness_vg00-swap      253:9    0     4G  0 lvm  [SWAP]
sdb                             8:16   0   2.2T  0 disk
└─sdb1                          8:17   0   2.2T  0 part
  └─netwitness_vg00-nwhome    253:5    0   4.3T  0 lvm  /var/netwitness
sdc                             8:32   0  29.1T  0 disk
└─sdc1                          8:33   0  29.1T  0 part
  ├─decodersmall-decoroot     253:1    0    20G  0 lvm  /var/netwitness/decoder
  ├─decodersmall-index        253:2    0    30G  0 lvm  /var/netwitness/decoder/index
  ├─decodersmall-sessiondb    253:3    0   600G  0 lvm  /var/netwitness/decoder/sessiondb
  └─decodersmall-metadb       253:4    0  28.5T  0 lvm  /var/netwitness/decoder/metadb
sdd                             8:48   0 116.4T  0 disk
└─sdd1                          8:49   0 116.4T  0 part
  └─decoder-packetdb          253:0    0 116.4T  0 lvm  /var/netwitness/decoder/packetdb
```

Configure Storage for Concentrator using MD2412

The following scenarios configure storage on one or more unencrypted, 12-Drive MD2412 PowerVault for a Concentrator physical host.

Note: All 12 drives on PowerVault MD2412 are HDDs. This is also used as storage for concentrator service whereas older PowerVault MD1400 for concentrator storage has mixed drive configuration with HDDs and SSDs. Since MD2412 has no support for SSDs and the concentrator service requires atleast 2 SSDs for index database, SSDs are to be installed on the concentrator physical host in slots 4 through 9 with a minimum of 3 drives.

Supported SSD configurations on the S7 concentrator host -
- 3 drives - slots 4,5 and 6 - RAID 5
- 6 drives - slots 4,5,6,7,8 and 9 - RAID 6
- 3 Drives + 3 Drives - RAID 5, initially 3 drives are installed followed by extension with another 3 drives.

The SSD drive packs are available in two sizes - 3 * 3.2 TB or 3 * 7.68 TB. In the following examples, the drive pack of 3 * 3.2 TB is used. When a larger drive pack (3 * 7.68 TB) is used, the virtual drives are larger in size.

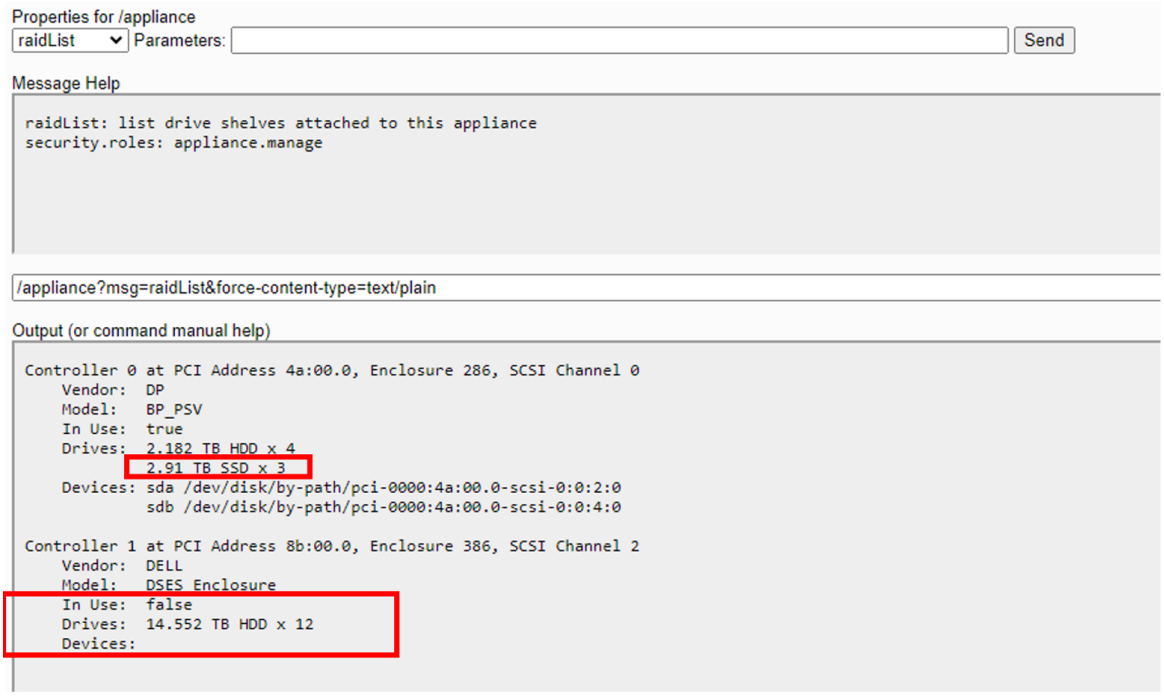
Scenario 1:

3 SSD drives are added to the Concentrator physical host. Host is connected to one unencrypted 12-Drive MD2412 PowerVault.

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.

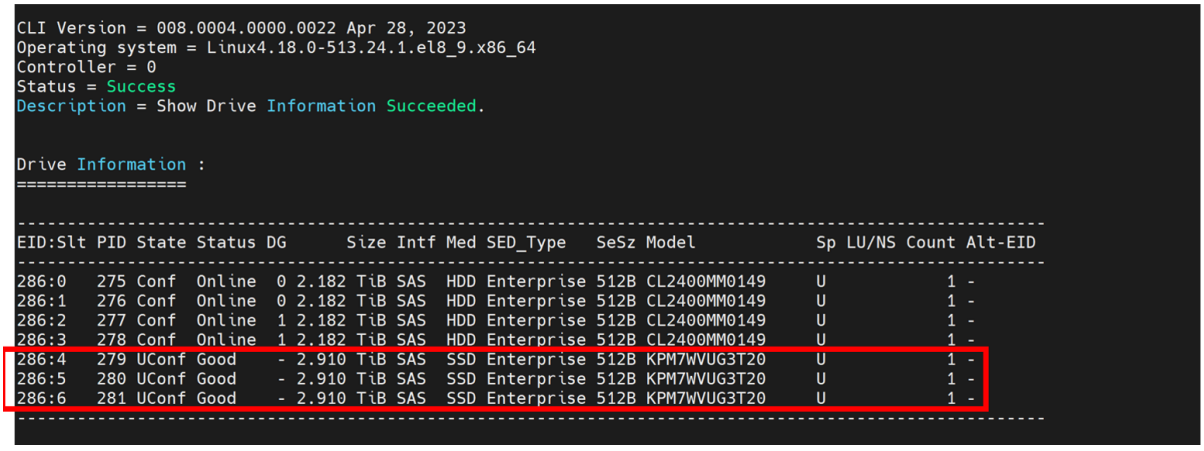


SSD configuration on this host is as seen below. 3 SSDs are added to slots 4,5 and 6 on the concentrator host for configuring index.

To view details about the SSDs installed on controller 0, run the following command on the concentrator:

```
/opt/MegaRAID/perccli2/perccli2 /c0/eall/sall show
```

Partial output of the above command is as shown below.

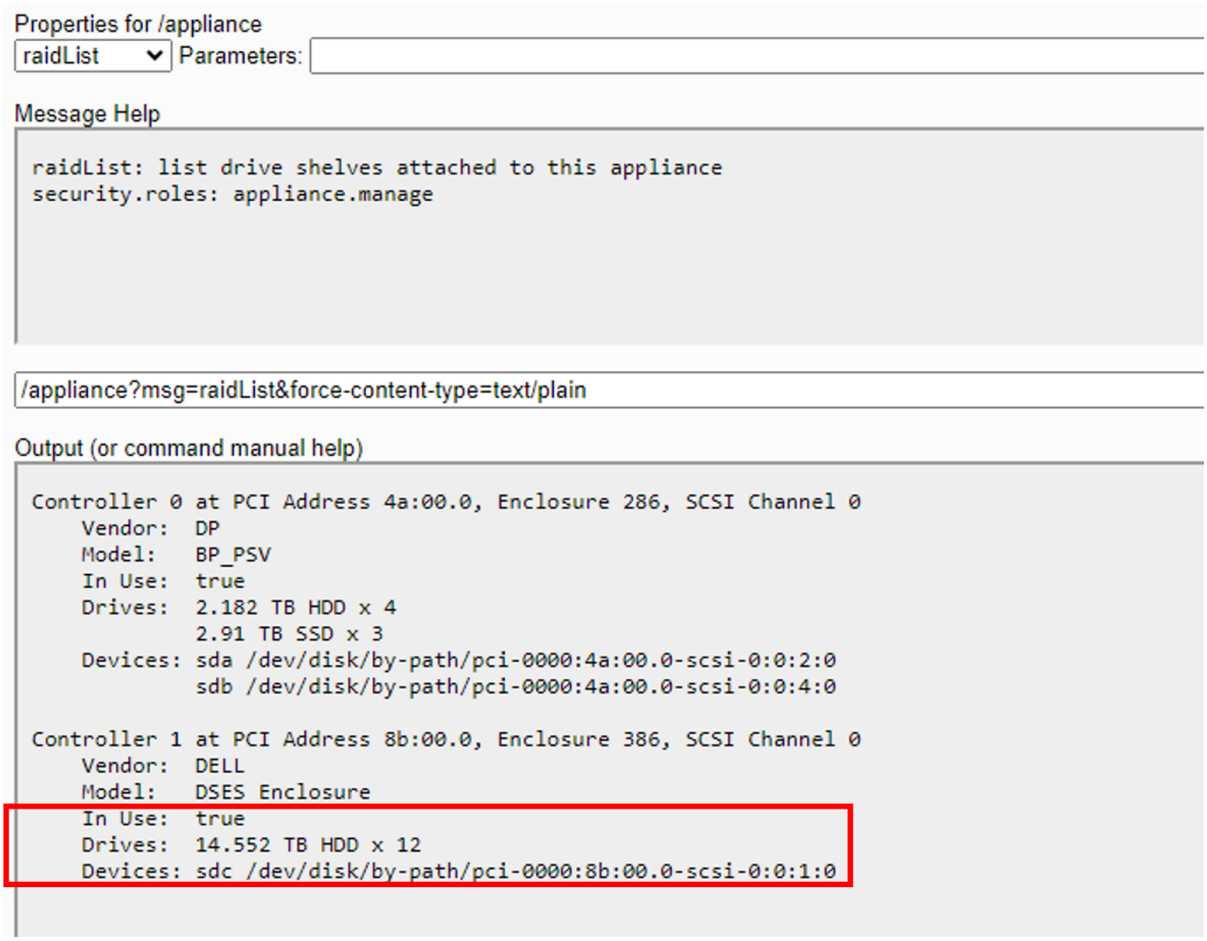


2. Execute the raidNew command with `scheme=concentrator for controller (1)` with HDDs and `scheme=concentrator-metakit` for controller (0) with SSDs.

   To configure RAID for concentrator scheme, execute `raidNew` with the following parameters using the controller number and the enclosure number you just recorded from `raidList`.

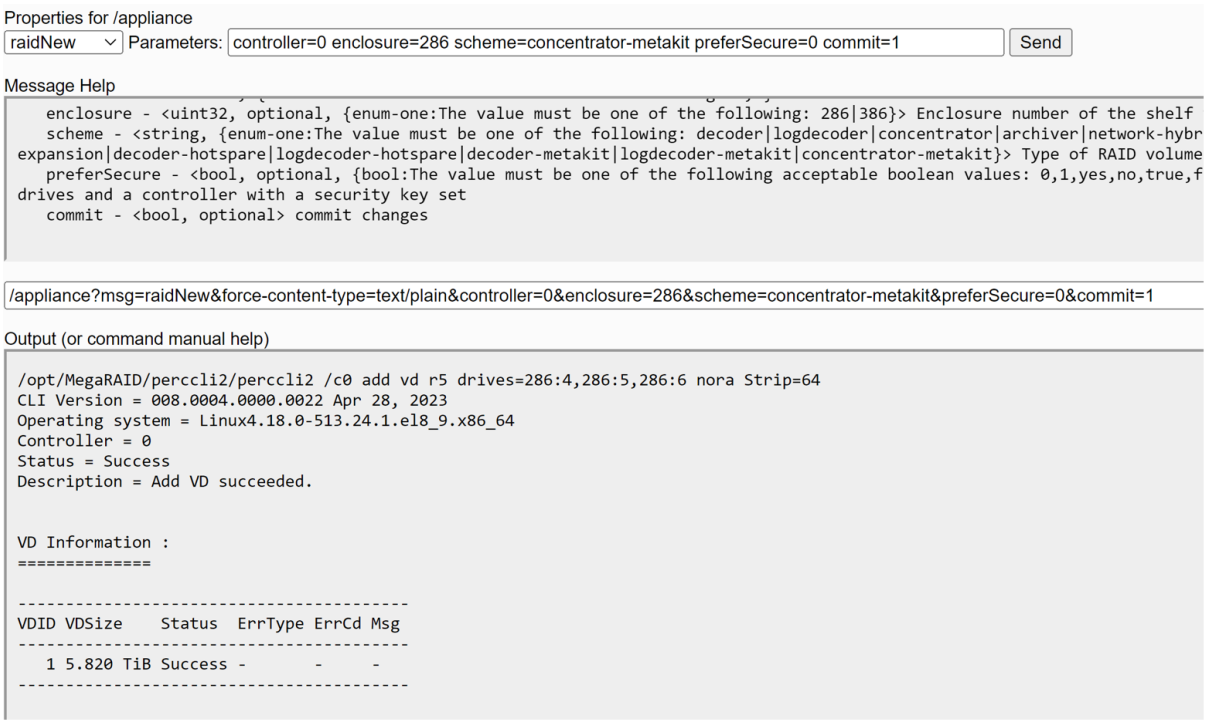
```
controller=1 enclosure=386 scheme=concentrator preferSecure=0 commit=1
```

3. Execute the `raidList` command to verify the concentrator RAID created.

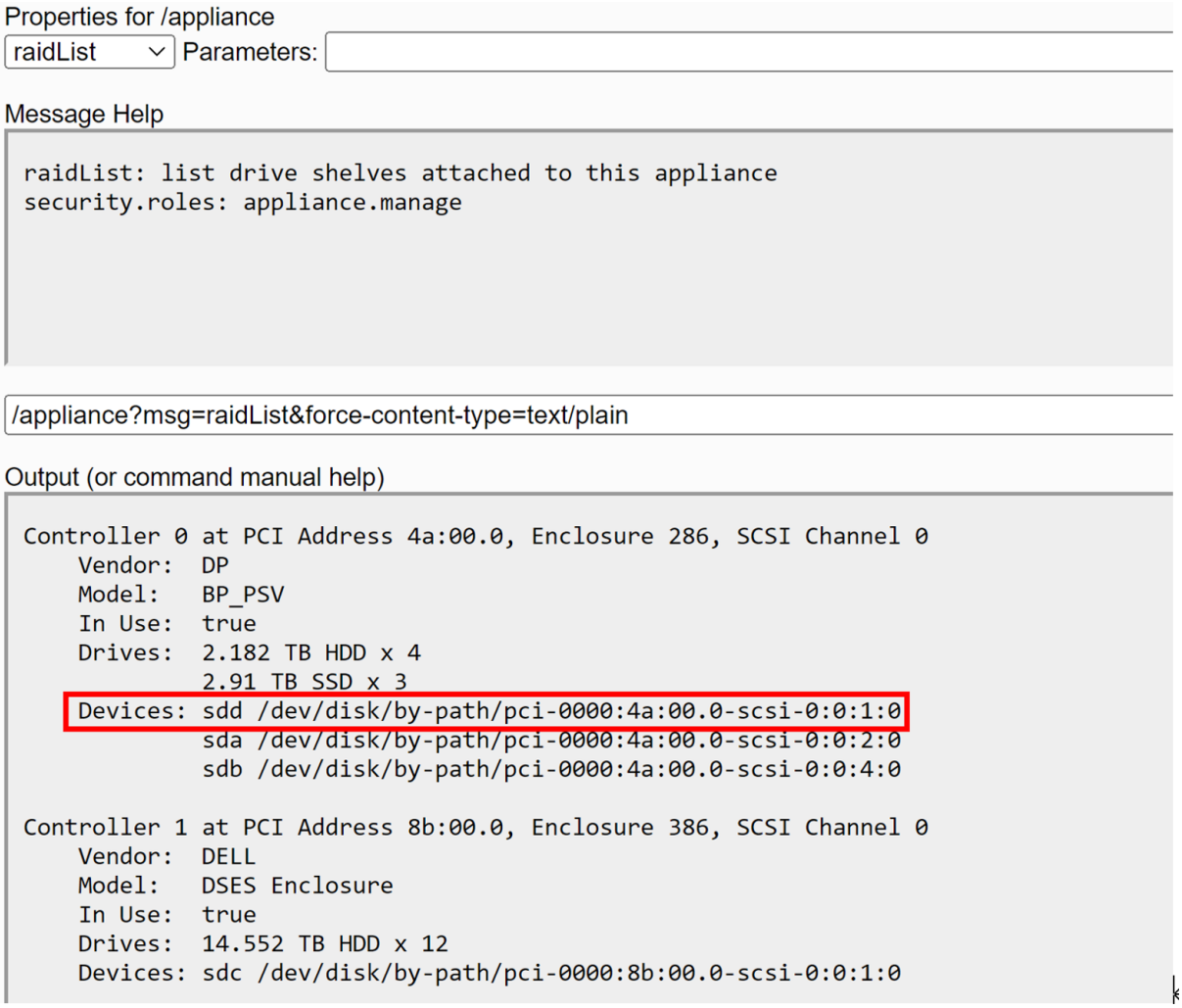

Properties for /appliance
raidList
Parameters:
Message Help
raidList: list drive shelves attached to this appliance
security.roles: appliance.manage
/appliance?msg=raidList&force-content-type=text/plain
Output (or command manual help)
Controller 0 at PCI Address 4a:00.0, Enclosure 286, SCSI Channel 0
Vendor: DP
Model: BP_PSV
In Use: true
Drives: 2.182 TB HDD x 4
2.91 TB SSD x 3
Devices: sda /dev/disk/by-path/pci-0000:4a:00.0-scsi-0:0:2:0
sdb /dev/disk/by-path/pci-0000:4a:00.0-scsi-0:0:4:0
Controller 1 at PCI Address 8b:00.0, Enclosure 386, SCSI Channel 0
Vendor: DELL
Model: DSES Enclosure
In Use: true
Drives: 14.552 TB HDD x 12
Devices: sdc /dev/disk/by-path/pci-0000:8b:00.0-scsi-0:0:1:0

4. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number associated with SSDs added to the concentrator host. This creates the RAID for index.

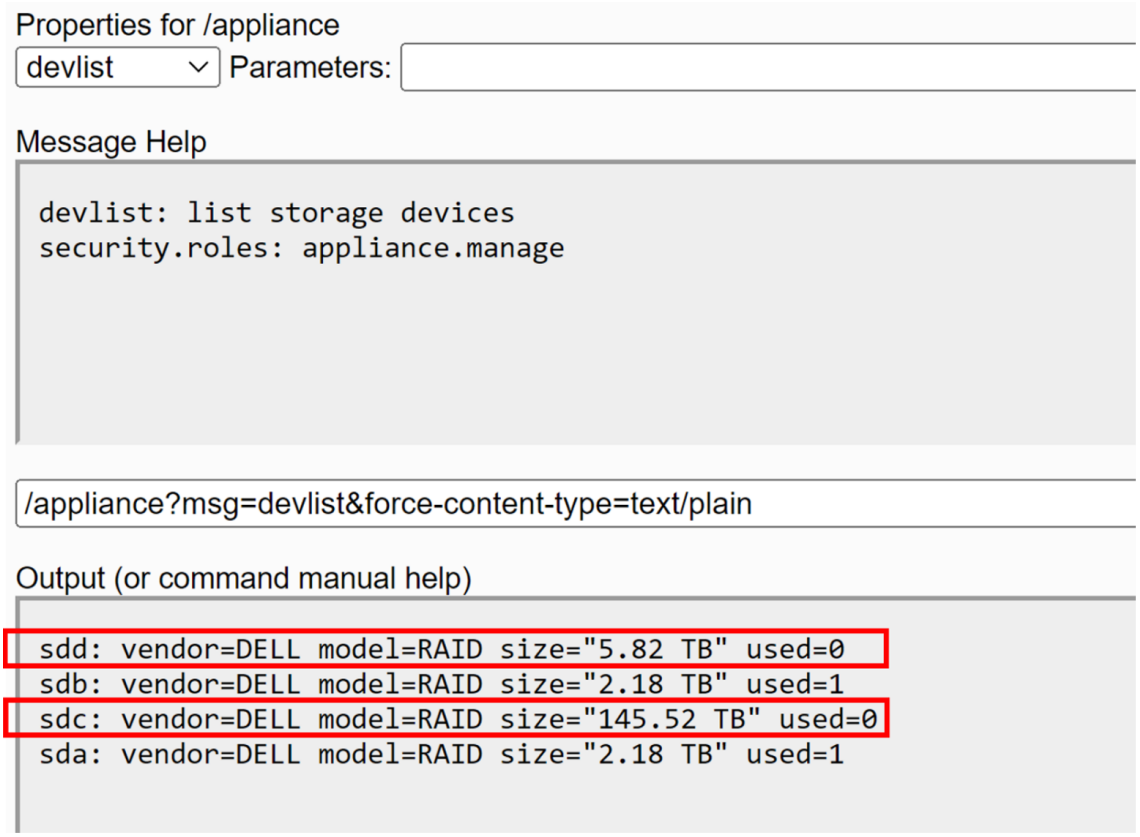
   ```
   controller=0 enclosure=286 scheme=concentrator-metakit preferSecure=0
   commit=1
   ```



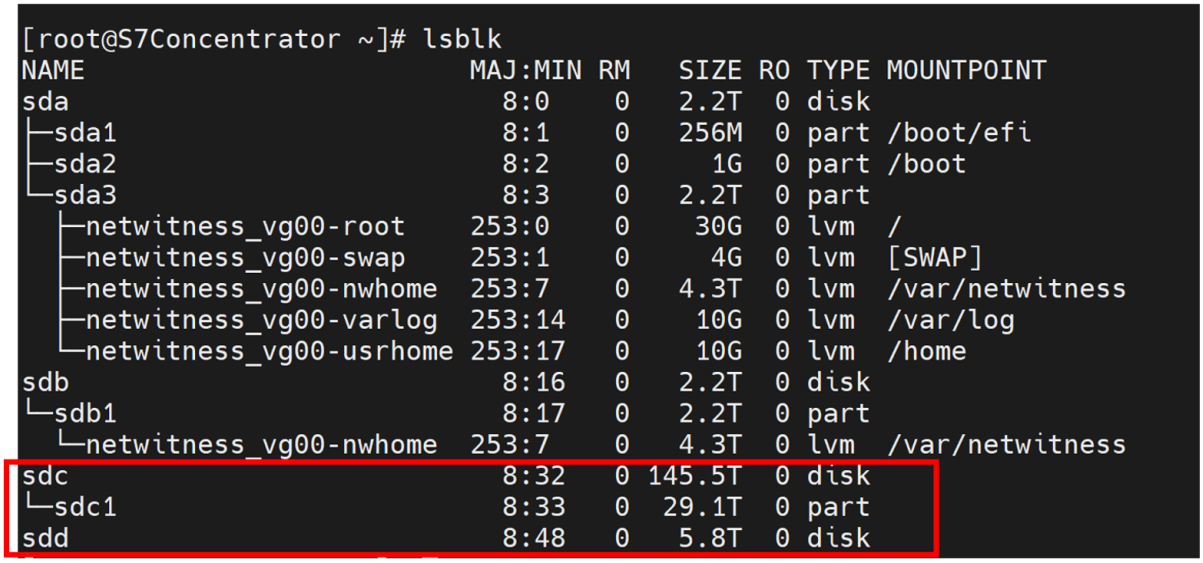
5. Execute the `raidList` command to verify the index RAID created.



6. Execute `devlist` to verify block devices created.

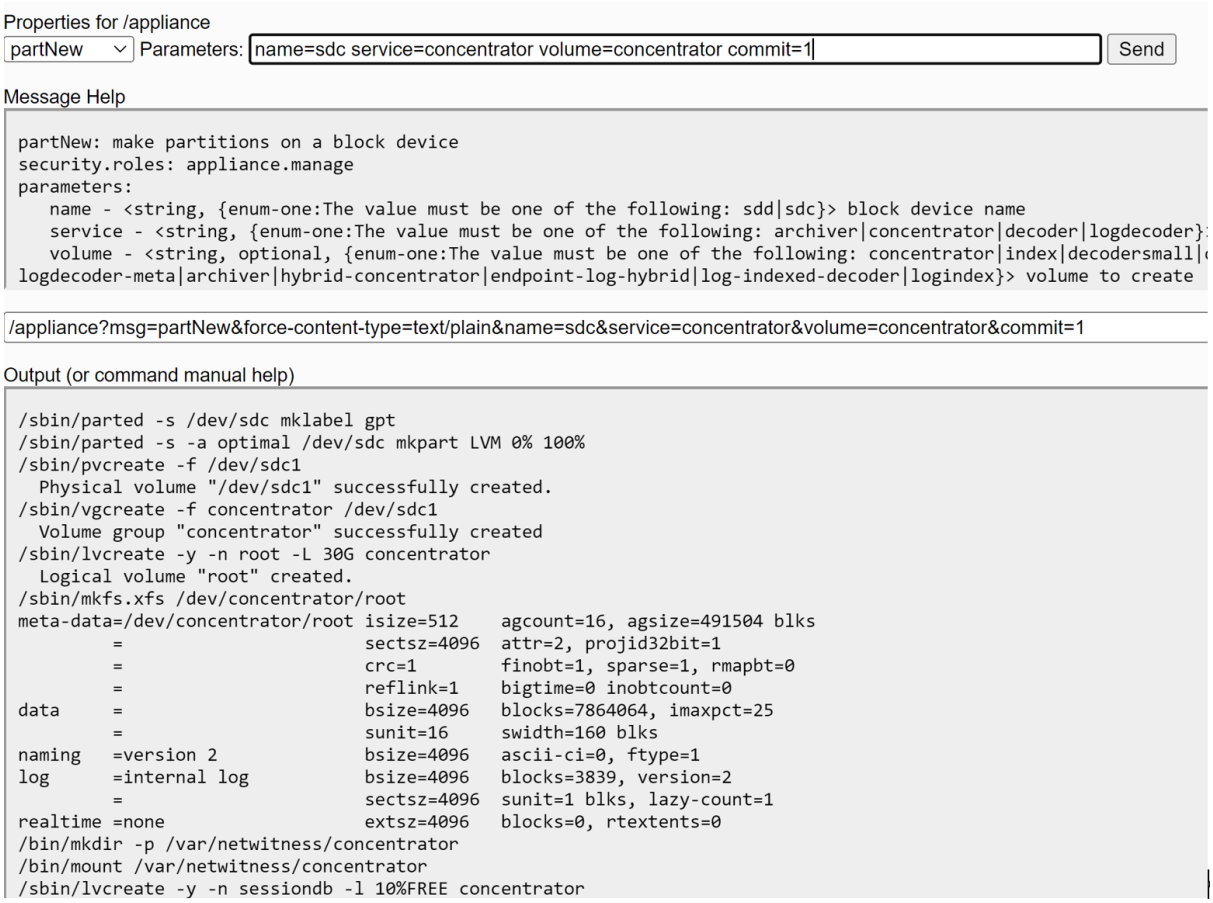


The same can be verified using `lsblk`.



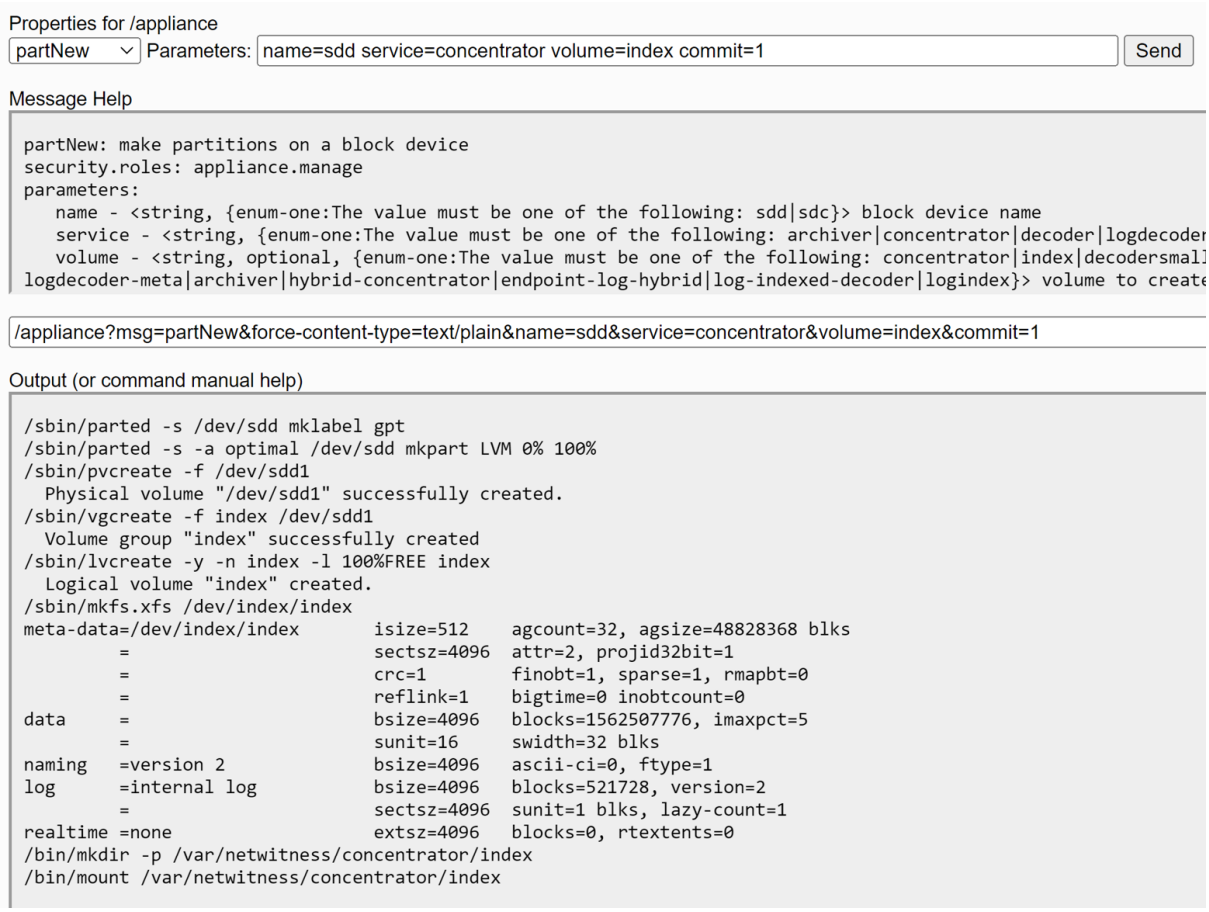
7. Execute the `partNew` command to create the concentrator partition first with the following parameters. You must create the concentrator volume before index volume or it will fail. sdc is assoviated with HDDs on PowerVault 2412 and is to be used for concentrator volume.

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

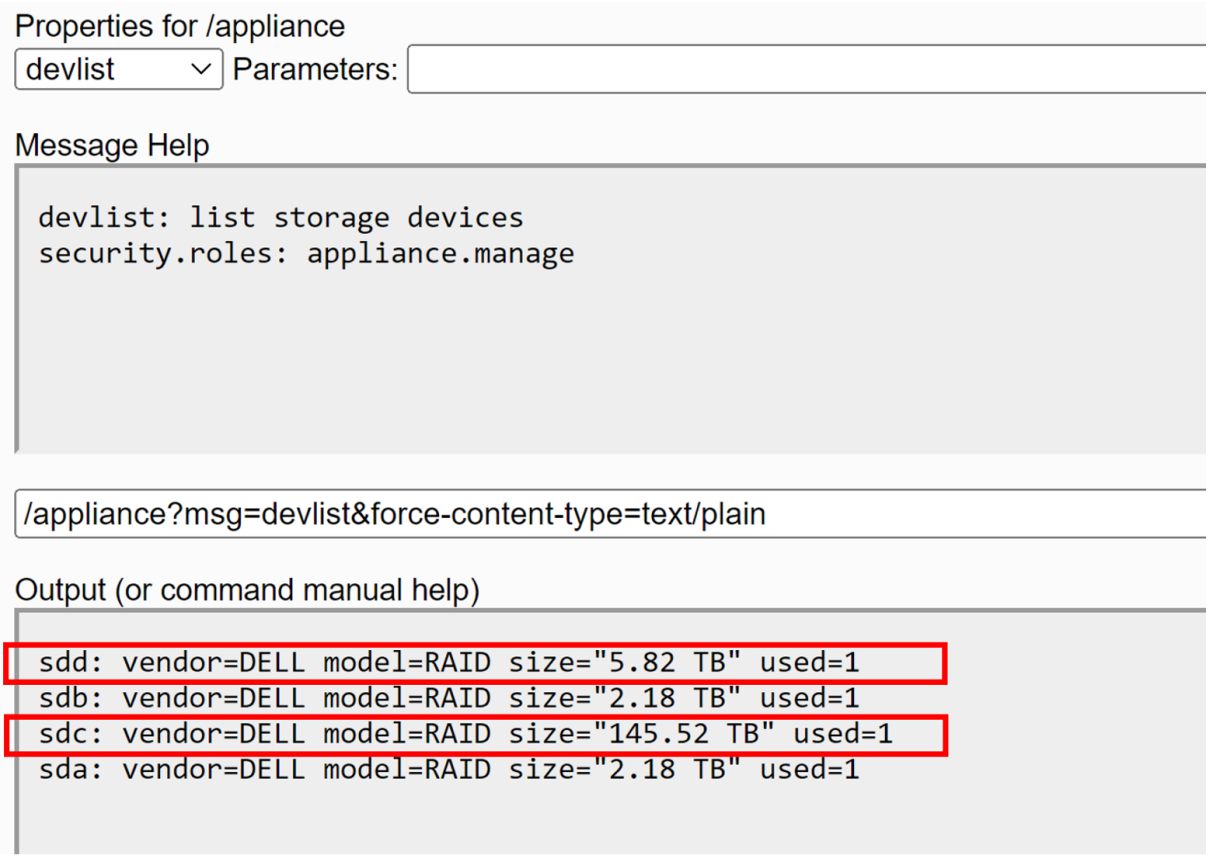


8. Execute the `partNew` command with the following parameters to create an index on SSDs. sdd is associated with the SSDs added and is to be used for index volume.

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

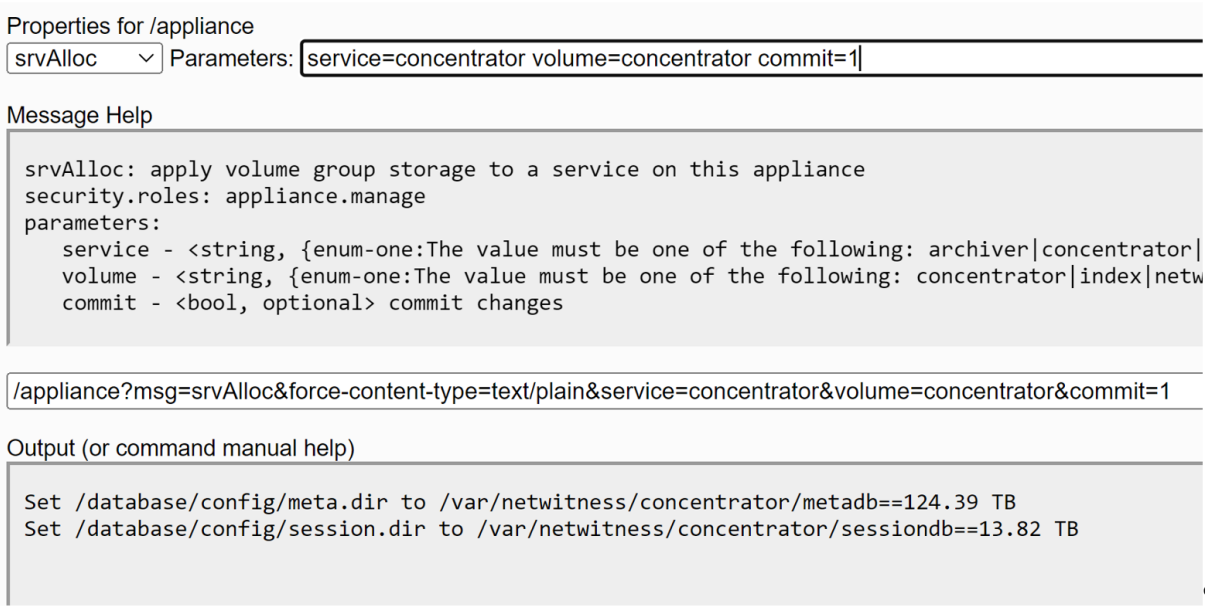


9. Run `devlist` to verify block devices have been configured.

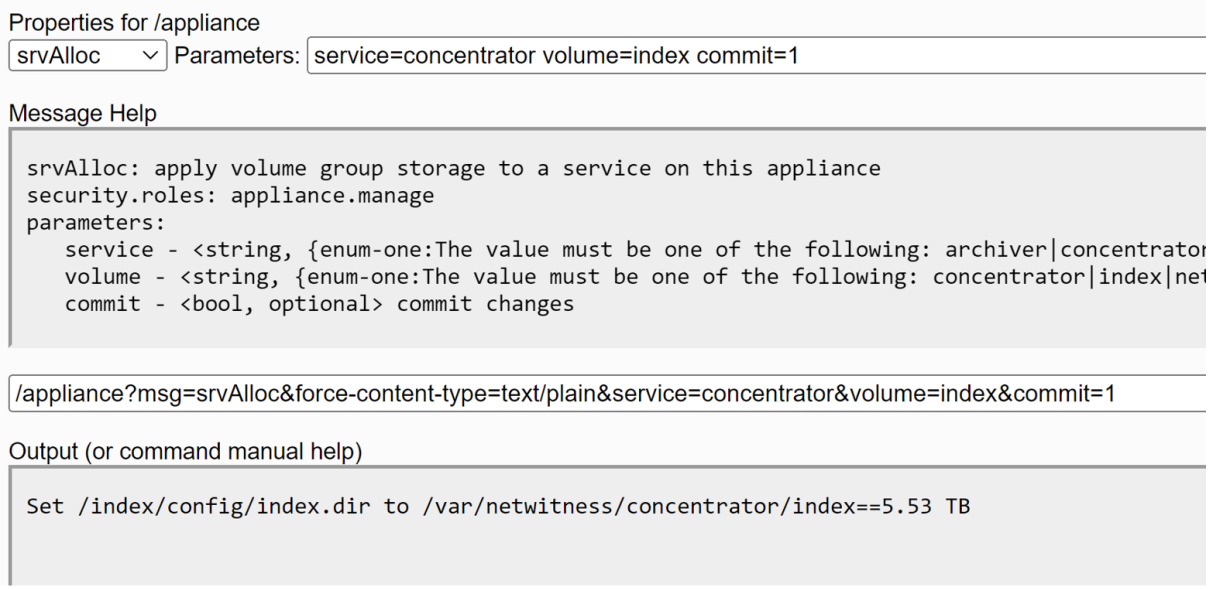


10. Execute the `srvAlloc` command with the following parameters.

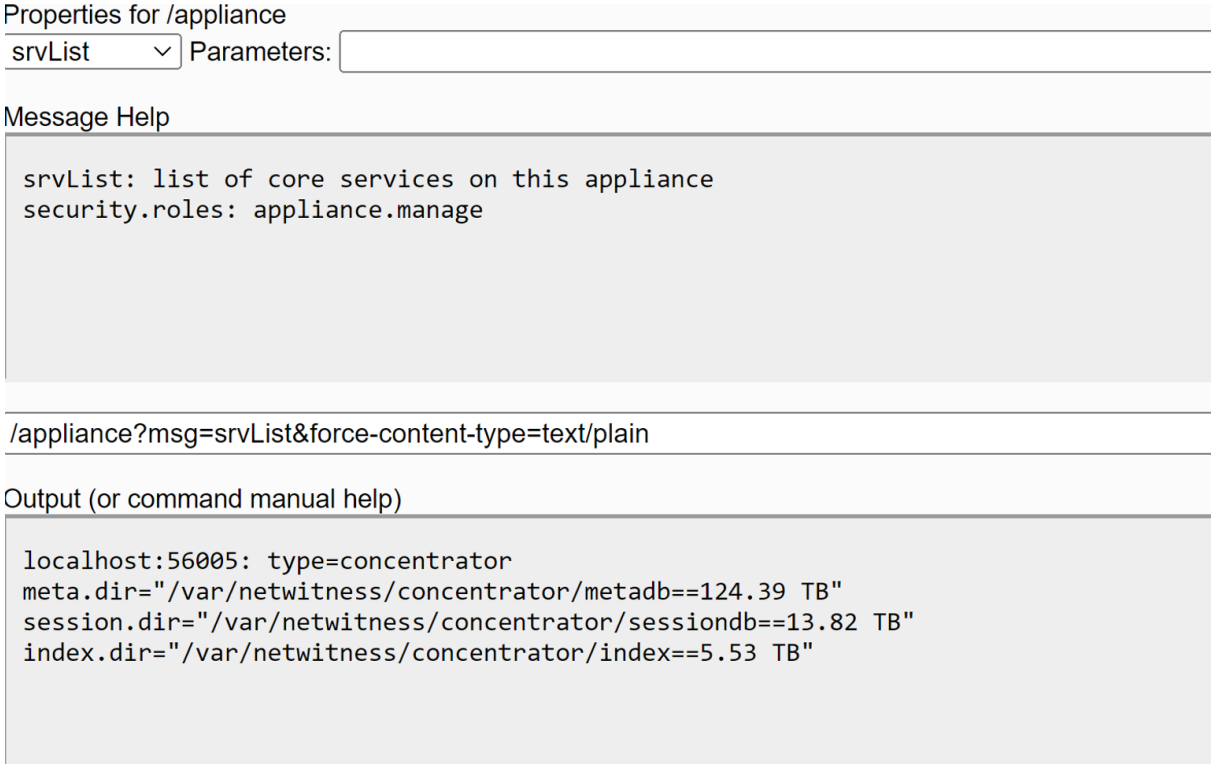
```
service=concentrator volume=concentrator commit=1
```




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



11. Execute `srvList` to verify storage has been assigned as required.



12. Verify partitions and mounts on `lsblk`.

```
[root@S7Concentrator ~]# lsblk
NAME                          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                             8:0    0   2.2T  0 disk
├─sda1                          8:1    0   256M  0 part /boot/efi
├─sda2                          8:2    0     1G  0 part /boot
└─sda3                          8:3    0   2.2T  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   4.3T  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   2.2T  0 disk
└─sdb1                          8:17   0   2.2T  0 part
  └─netwitness_vg00-nwhome    253:2    0   4.3T  0 lvm  /var/netwitness
sdc                             8:32   0 145.5T  0 disk
└─sdc1                          8:33   0 145.5T  0 part
  ├─concentrator-root         253:5    0    30G  0 lvm  /var/netwitness/concentrator
  ├─concentrator-sessiondb    253:6    0  14.6T  0 lvm  /var/netwitness/concentrator/sessiondb
  └─concentrator-metadb       253:7    0   131T  0 lvm  /var/netwitness/concentrator/metadb
sdd                             8:48   0   5.8T  0 disk
└─sdd1                          8:49   0   5.8T  0 part
  └─index-index               253:8    0   5.8T  0 lvm  /var/netwitness/concentrator/index
```

Scenario 2:

3 SSD Drives + 3 SSD Drives are installed on the Concentrator physical host. This means that initially 3 SSD drives are installed, followed by extension with another 3 drives.

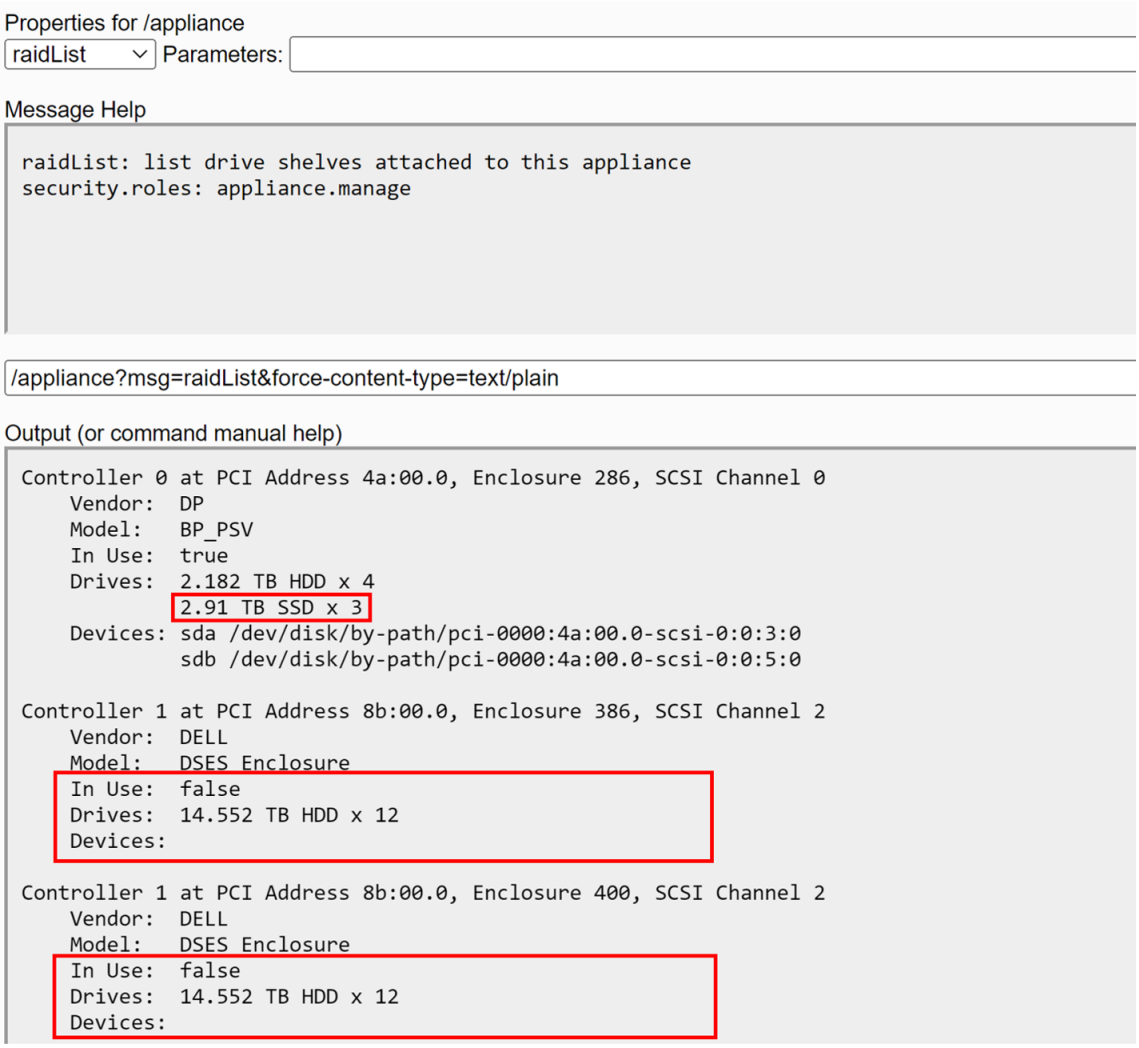
Host is connected to two unencrypted 12-Drive MD2412 PowerVaults.

The following scenario configures storage as per the scenario described above.

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.



SSD configuration on this host is as seen below. 3 SSDs from the first drive pack are added to slots 4,5 and 6 on the concentrator host for configuring index.

To view details about the SSDs installed on controller 0, run the following command on the concentrator:

```
/opt/MegaRAID/perccli2/perccli2 /c0/eall/sall show
```

Partial output of the above command is as shown below.

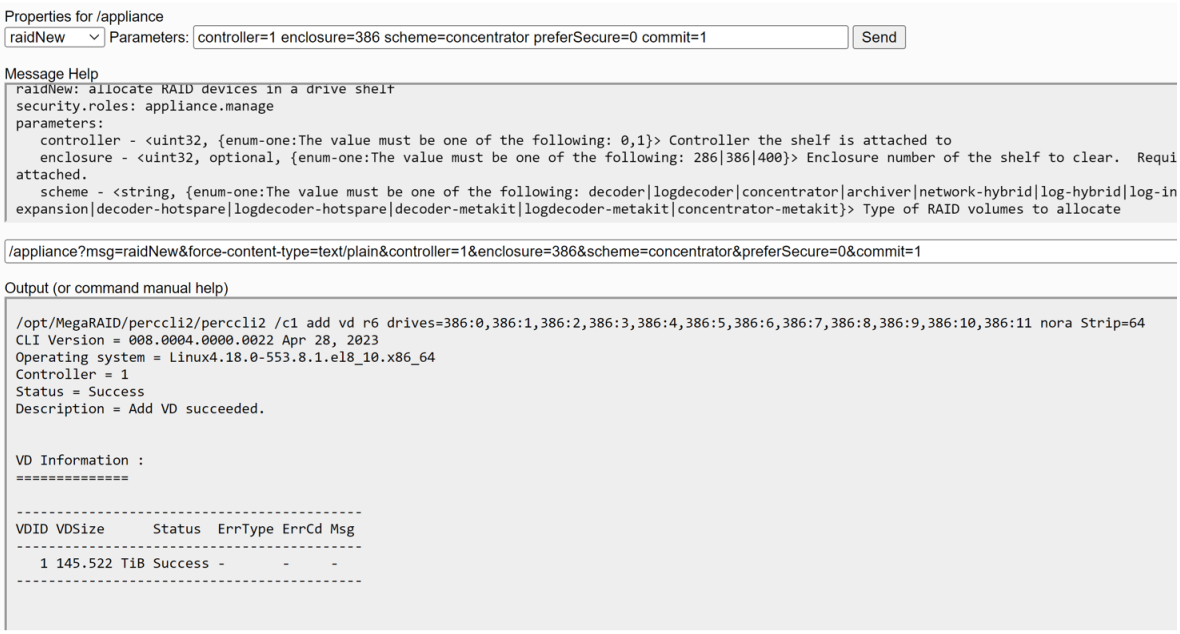
```
CLI Version = 008.0004.0000.0022 Apr 28, 2023
Operating system = Linux4.18.0-553.8.1.el8_10.x86_64
Controller = 0
Status = Success
Description = Show Drive Information Succeeded.


Drive Information :
==================

-------------------------------------------------------------------------------------------------------------
EID:Slt PID State Status DG      Size Intf Med SED_Type   SeSz Model            Sp LU/NS Count Alt-EID
-------------------------------------------------------------------------------------------------------------
286:0   275 Conf  Online  0 2.182 TiB SAS  HDD Enterprise 512B CL2400MM0149     U            1 -
286:1   276 Conf  Online  0 2.182 TiB SAS  HDD Enterprise 512B CL2400MM0149     U            1 -
286:2   277 Conf  Online  1 2.182 TiB SAS  HDD Enterprise 512B CL2400MM0149     U            1 -
286:3   278 Conf  Online  1 2.182 TiB SAS  HDD Enterprise 512B CL2400MM0149     U            1 -
286:4   279 UConf Good    - 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
286:5   280 UConf Good    - 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
286:6   281 UConf Good    - 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
-------------------------------------------------------------------------------------------------------------
```

2. Execute the raidNew command with scheme=concentrator for controller (1) with HDDs and scheme=concentrator-metakit for controller (0) with SSDs.

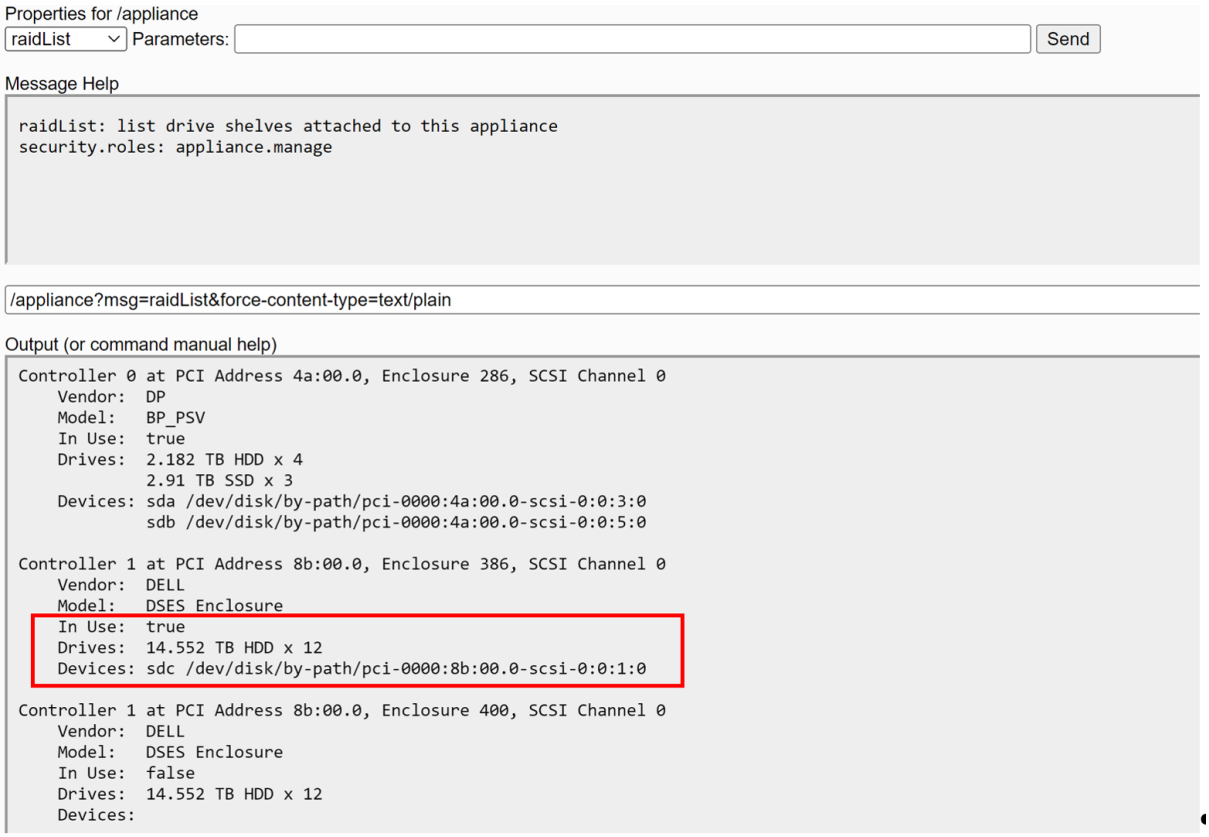
   To configure RAID for concentrator scheme, execute `raidNew` with the following parameters using the controller number and the enclosure number you just recorded from raidList.

   ```
   controller=1 enclosure=386 scheme=concentrator preferSecure=0 commit=1
   ```

   This creates concentrator RAID using the HDDs in PowerVault MD2412.

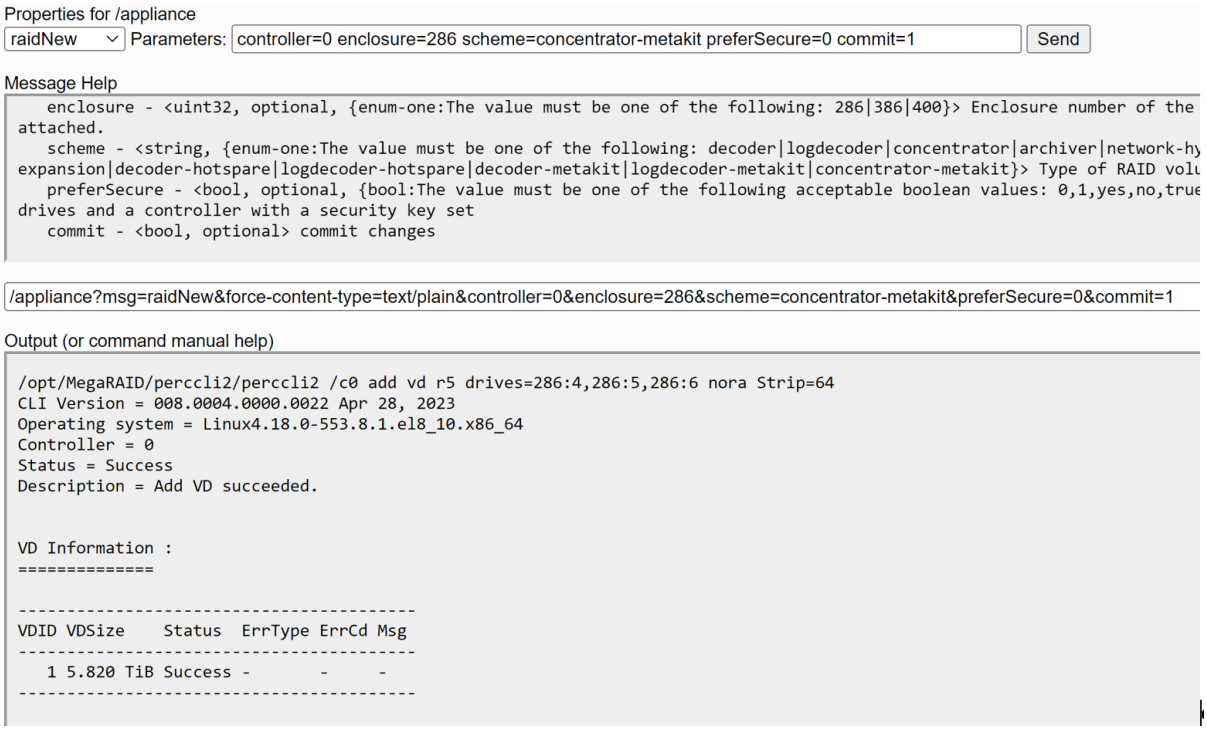


3. Execute the `raidList` command to verify the concentrator RAID created.

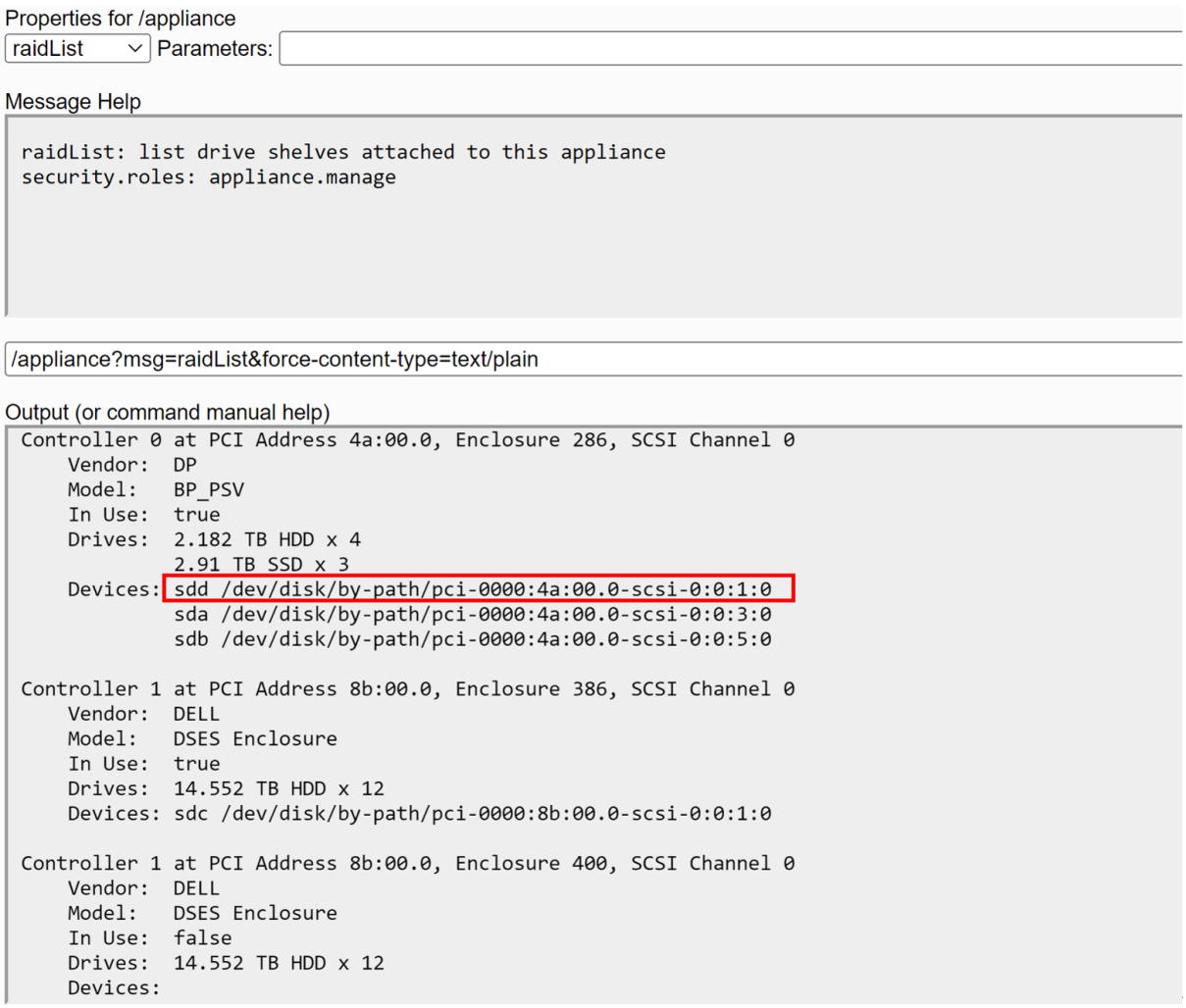


4. To configure RAID for index scheme, execute `raidNew` with the following parameters using the controller number and the enclosure number associated with SSDs added to the concentrator host. This creates the RAID for index.

   ```
   controller=0 enclosure=286 scheme=concentrator-metakit preferSecure=0
   commit=1
   ```



5. Execute `raidList` command to verify the index RAID created.



6. Add the second drive pack with another 3 SSDs.

   SSD configuration on this host is as seen below. 3 SSDs from the second drive pack are added to slots 7,8 and 9 on the concentrator host for configuring index.

   To view details about the SSDs installed on controller 0, run the following command on the concentrator:

   ```
   /opt/MegaRAID/perccli2/perccli2 /c0/eall/sall show
   ```

   Partial output of the above command is as shown below.

```
CLI Version = 008.0004.0000.0022 Apr 28, 2023
Operating system = Linux4.18.0-553.8.1.el8_10.x86_64
Controller = 0
Status = Success
Description = Show Drive Information Succeeded.


Drive Information :
==================

-----------------------------------------------------------------------------------------------------------
EID:Slt PID State Status DG      Size Intf Med SED_Type   SeSz Model            Sp LU/NS Count Alt-EID
-----------------------------------------------------------------------------------------------------------
286:0   275 Conf  Online  1 2.182 TiB SAS  HDD Enterprise 512B CL2400MM0149     U            1 -
286:1   276 Conf  Online  1 2.182 TiB SAS  HDD Enterprise 512B CL2400MM0149     U            1 -
286:2   277 Conf  Online  2 2.182 TiB SAS  HDD Enterprise 512B CL2400MM0149     U            1 -
286:3   278 Conf  Online  2 2.182 TiB SAS  HDD Enterprise 512B CL2400MM0149     U            1 -
286:4   279 Conf  Online  0 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
286:5   280 Conf  Online  0 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
286:6   281 Conf  Online  0 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
286:7   282 UConf Good    - 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
286:8   283 UConf Good    - 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
286:9   284 UConf Good    - 2.910 TiB SAS  SSD Enterprise 512B KPM7WVUG3T20     U            1 -
-----------------------------------------------------------------------------------------------------------
```

7. Configure RAID on the second PowerVault. Execute the `raidList` command.

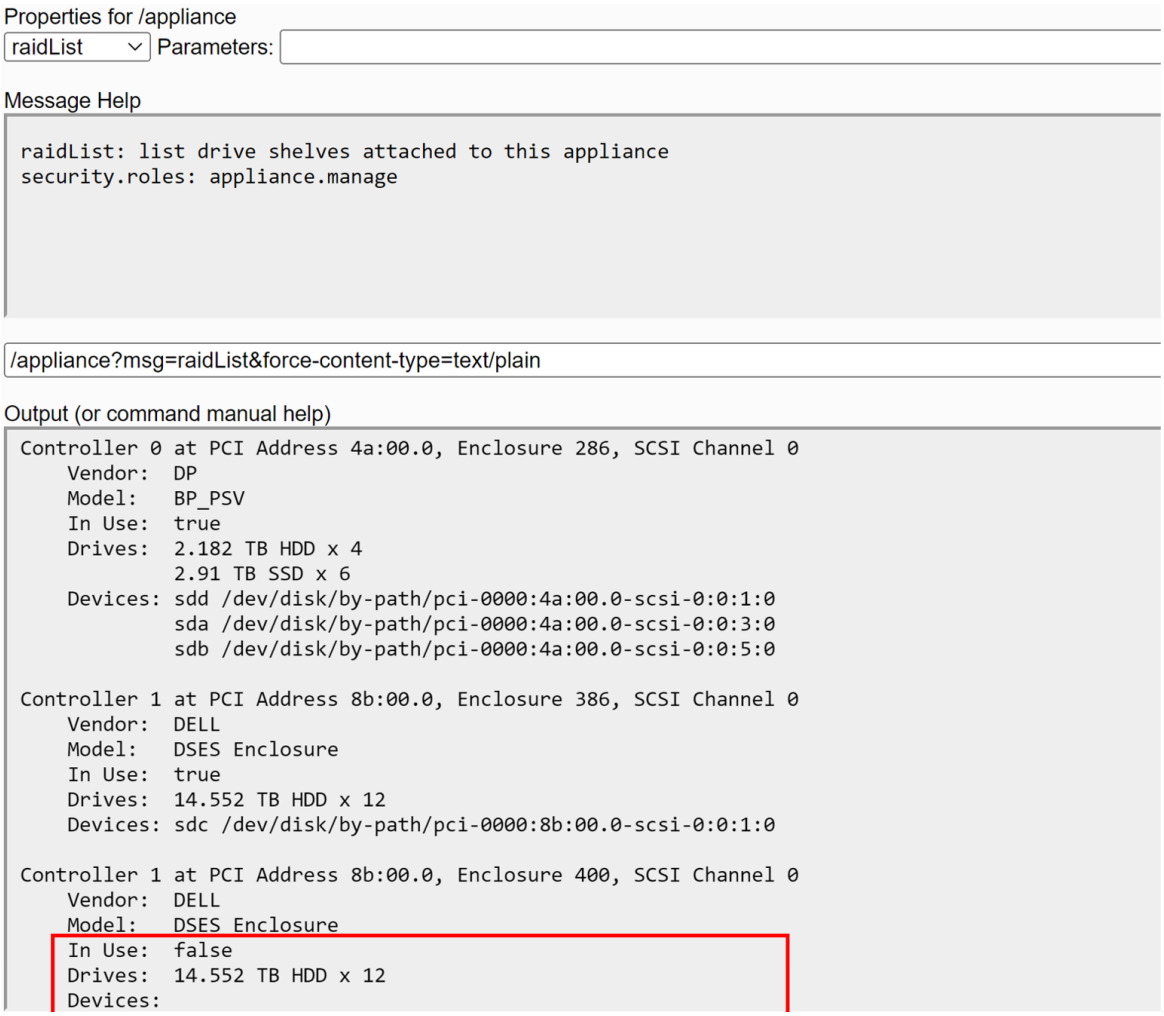
   a. Record the Controller Number, Enclosure Number, In Use, Drives, and Devices. You should see the following information.

      `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.

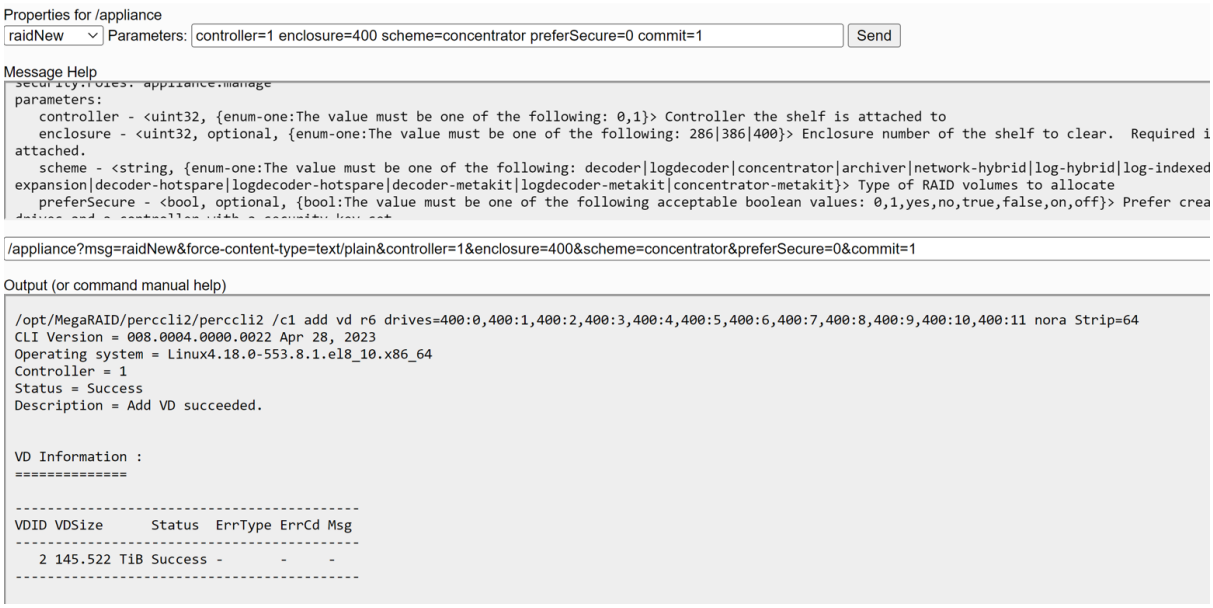


8. Execute the `raidNew` command with `scheme=concentrator` for controller (1) with HDDs and `scheme=concentrator-metakit` for controller (0) with SSDs.

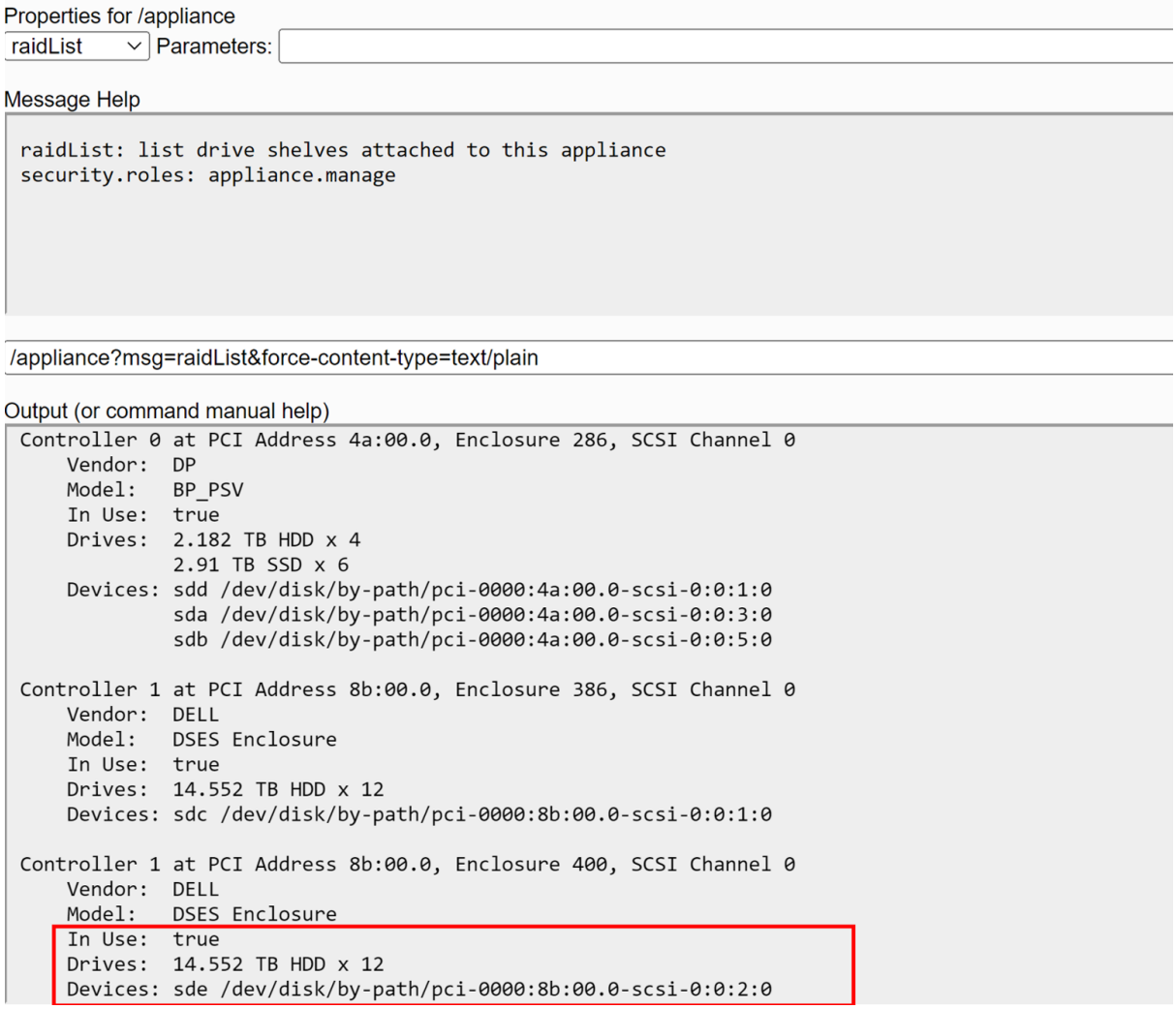
   To configure RAID for concentrator scheme, execute raidNew with the following parameters using the controller number and the enclosure number you just recorded from raidList.

   This creates concentrator RAID using the HDDs in PowerVault MD2412.

   ```
   controller=1 enclosure=400 scheme=concentrator preferSecure=0 commit=1
   ```

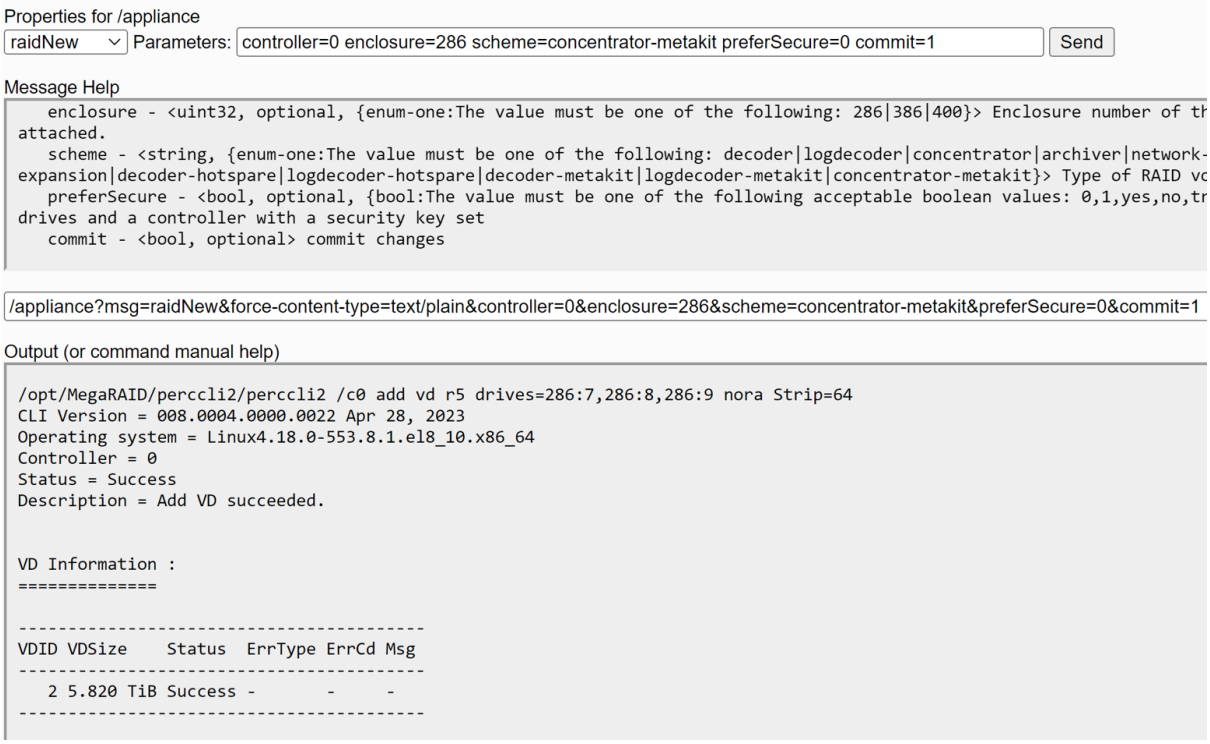


9. Execute the `raidList` command to verify the concentrator RAID created.

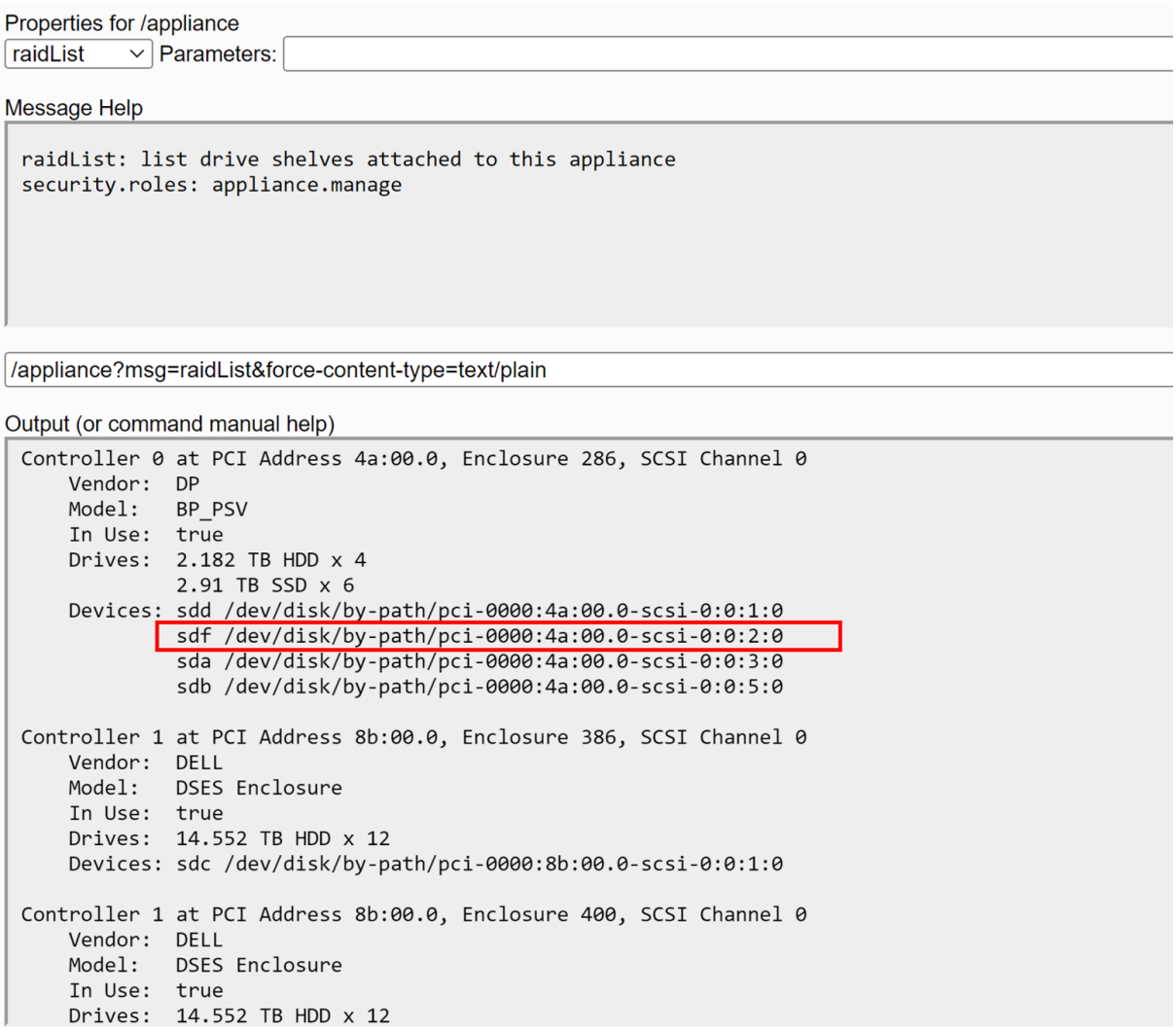


10. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number associated with SSDs added to the concentrator host. This creates the RAID for index.

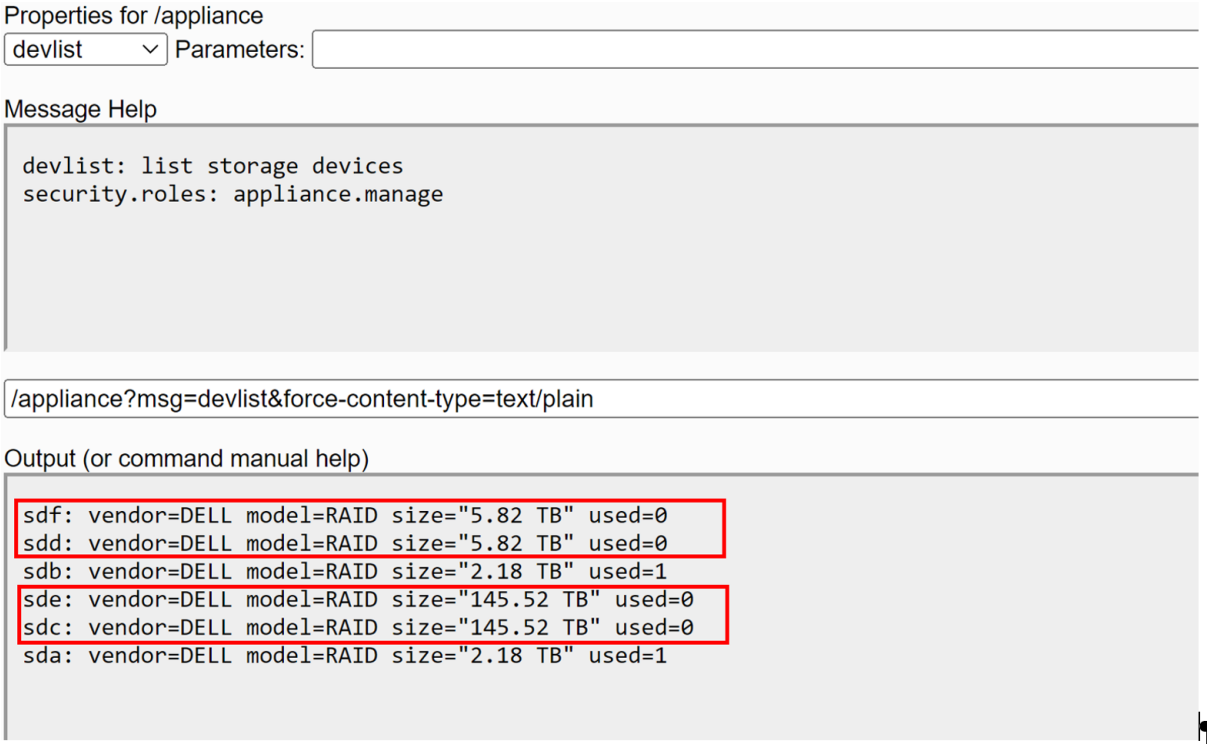
```
controller=0 enclosure=286 scheme=concentrator-metakit preferSecure=0
commit=1
```



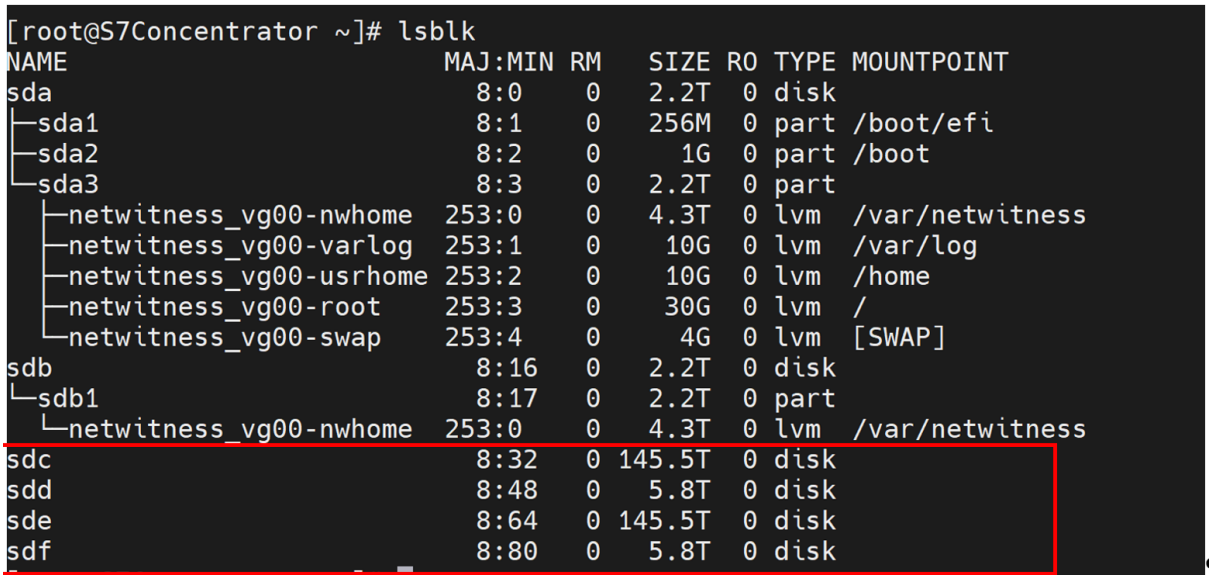
11. Execute the `raidList` command to verify the index RAID created.



12. Execute `devlist` to verify block devices created.



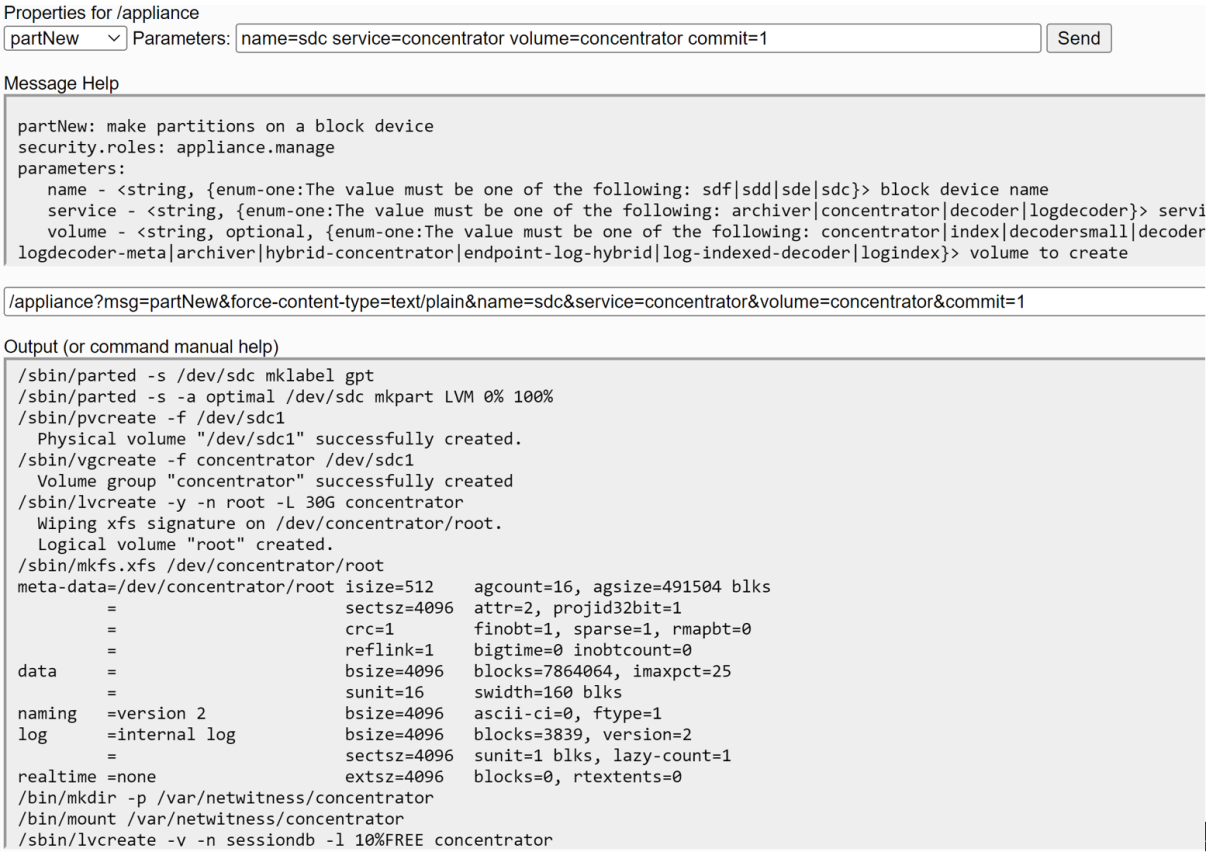
13. The same can be verified using `lsblk`.



14. 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.

    sdc is associated with HDDs on PowerVault MD2412 and is to be used for concentrator volume.

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



15. Execute the `partNew` command with the following parameters to create an index on SSDs. sdd is associated with the SSDs added and is to be used for index volume.

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

Properties for /appliance

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

Message Help

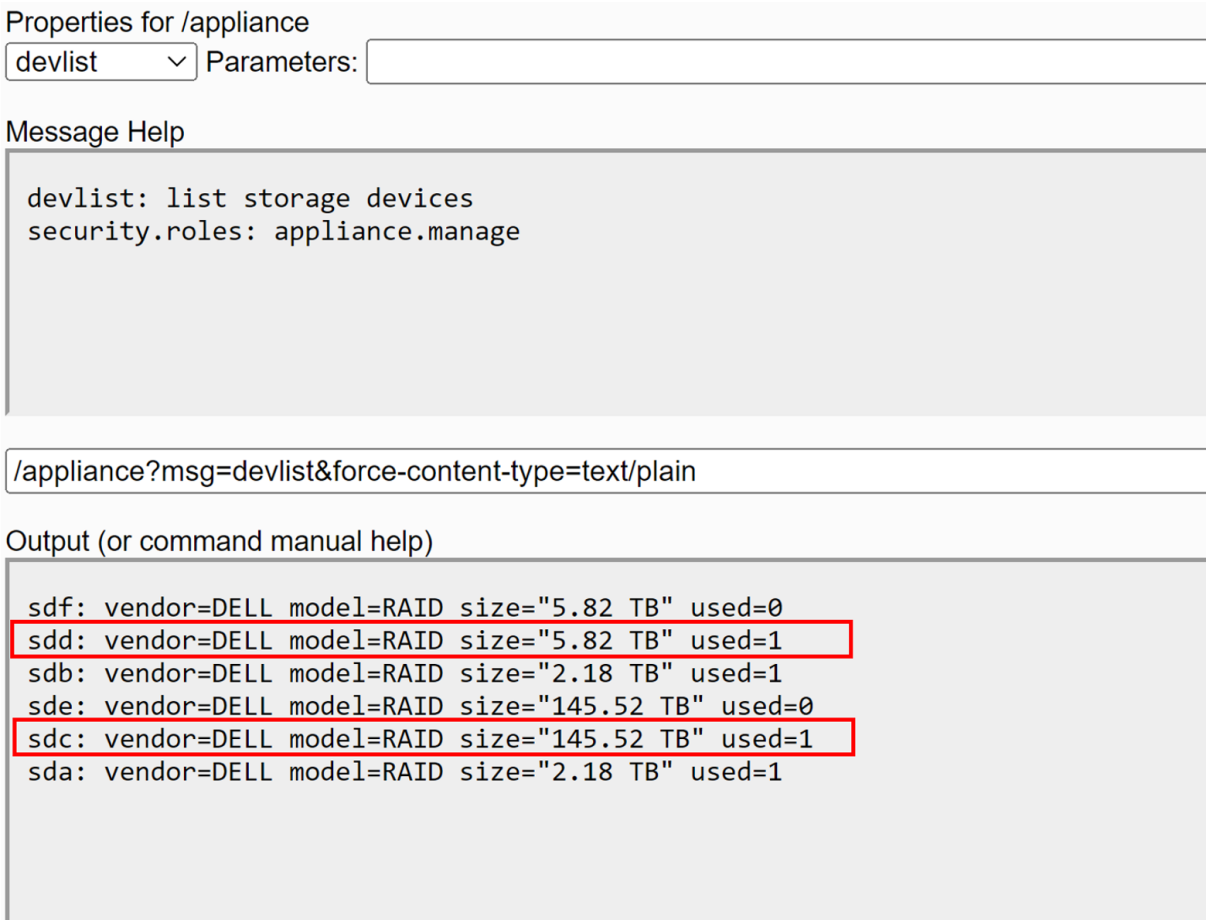
```
partNew: make partitions on a block device
security.roles: appliance.manage
parameters:
   name - <string, {enum-one:The value must be one of the following: sdf|sdd|sde}> block device name
   service - <string, {enum-one:The value must be one of the following: archiver|concentrator|decode
   volume - <string, optional, {enum-one:The value must be one of the following: concentrator|index|
logdecoder-meta|archiver|hybrid-concentrator|endpoint-log-hybrid|log-indexed-decoder|logindex}> volu
```

/appliance?msg=partNew&force-content-type=text/plain&name=sdd&service=concentrator&volume=index&commit=1

Output (or command manual help)

```
/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 index /dev/sdd1
  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=32, agsize=48828368 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0 inobtcount=0
data     =                       bsize=4096   blocks=1562507776, imaxpct=5
         =                       sunit=16     swidth=32 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=521728, 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
/bin/mount /var/netwitness/concentrator/index
```

16. Run `devlist` to verify block devices have been configured.



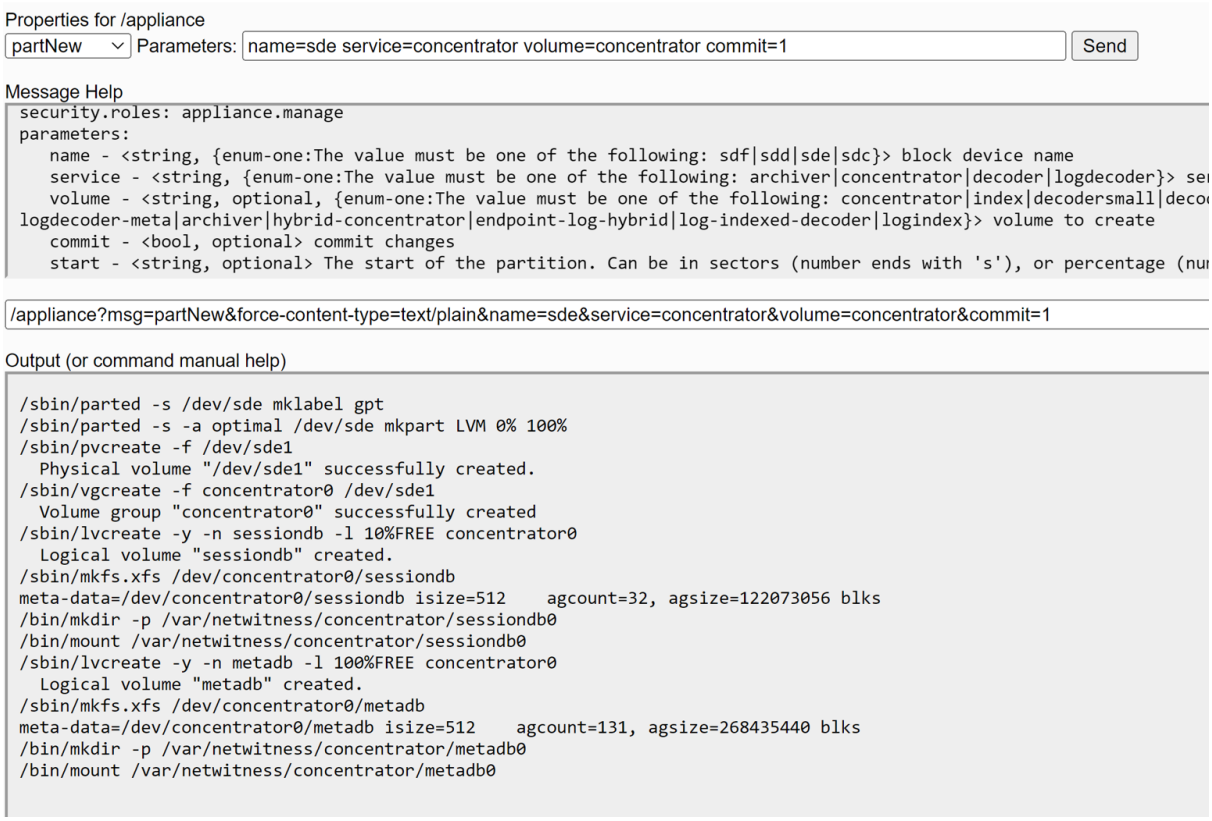
17. Verify partitions and mounts on `lsblk`.

```
[root@S7Concentrator ~]# lsblk
NAME                        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                           8:0    0   2.2T  0 disk
├─sda1                        8:1    0   256M  0 part /boot/efi
├─sda2                        8:2    0     1G  0 part /boot
└─sda3                        8:3    0   2.2T  0 part
  ├─netwitness_vg00-nwhome  253:0    0   4.3T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog  253:1    0    10G  0 lvm  /var/log
  ├─netwitness_vg00-usrhome 253:2    0    10G  0 lvm  /home
  ├─netwitness_vg00-root    253:3    0    30G  0 lvm  /
  └─netwitness_vg00-swap    253:4    0     4G  0 lvm  [SWAP]
sdb                           8:16   0   2.2T  0 disk
└─sdb1                        8:17   0   2.2T  0 part
  └─netwitness_vg00-nwhome  253:0    0   4.3T  0 lvm  /var/netwitness
sdc                           8:32   0 145.5T  0 disk
└─sdc1                        8:33   0 145.5T  0 part
  ├─concentrator-root       253:5    0    30G  0 lvm  /var/netwitness/concentrator
  ├─concentrator-sessiondb  253:6    0  14.6T  0 lvm  /var/netwitness/concentrator/sessiondb
  └─concentrator-metadb     253:7    0   131T  0 lvm  /var/netwitness/concentrator/metadb
sdd                           8:48   0   5.8T  0 disk
└─sdd1                        8:49   0   5.8T  0 part
  └─index-index             253:8    0   5.8T  0 lvm  /var/netwitness/concentrator/index
sde                           8:64   0 145.5T  0 disk
sdf                           8:80   0   5.8T  0 disk
```

18. Execute the `partNew` command to create the concentrator partition on the second PowerVault with the following parameters. You must create the concentrator volume before index volume or it will fail.

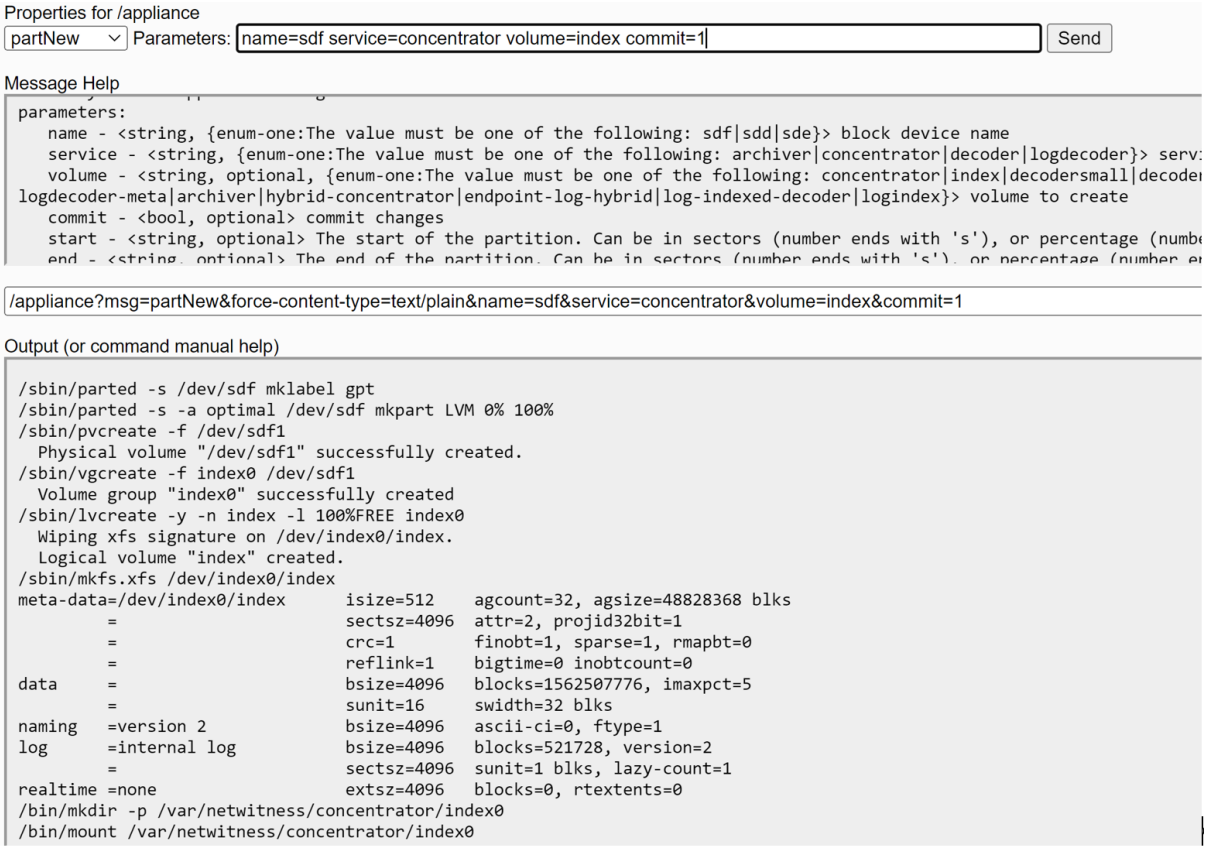
    sde is associated with HDDs on PowerVault MD2412 and is to be used for concentrator volume.

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

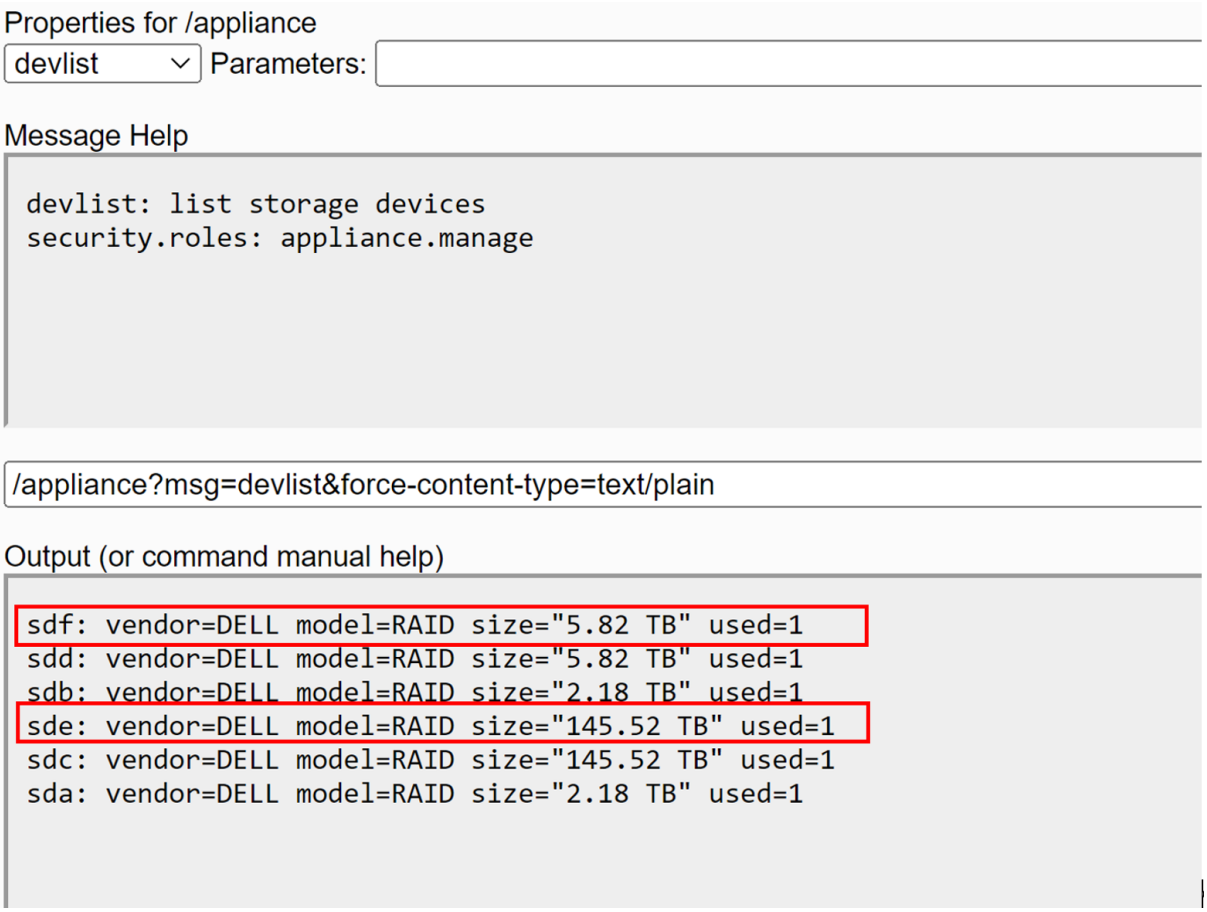
    

19. Execute the `partNew` command to create the index partition on SSDs associated with the second SSD drive-pack. sdf is associated with these SSDs and is to be used for index volume.

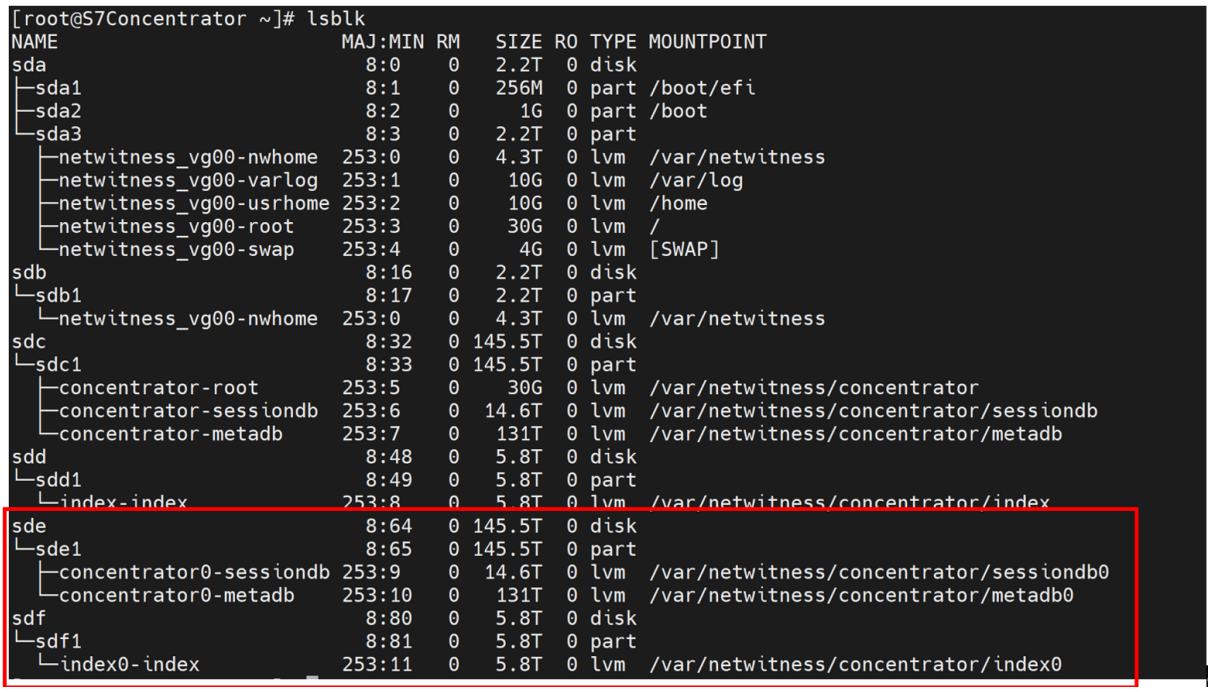
    ```
    name=sdf service=concentrator volume=index commit=1
    ```



20. Run `devlist` to verify block devices have been configured.

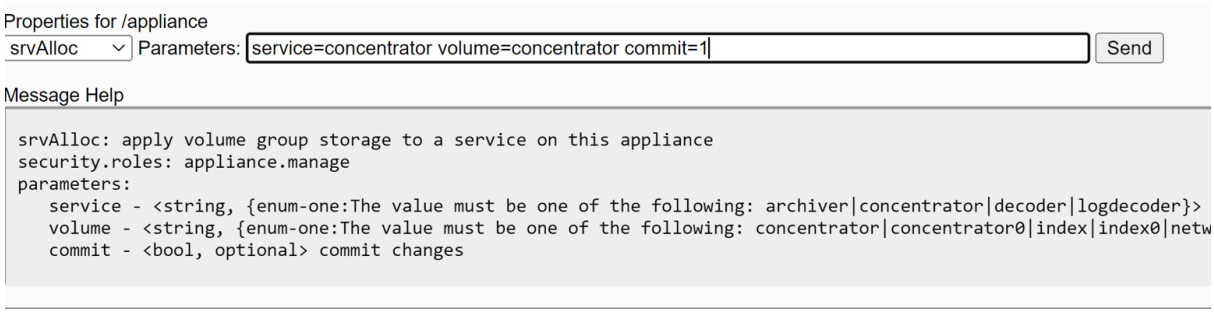


21. Verify partitions and mounts on `lsblk`.

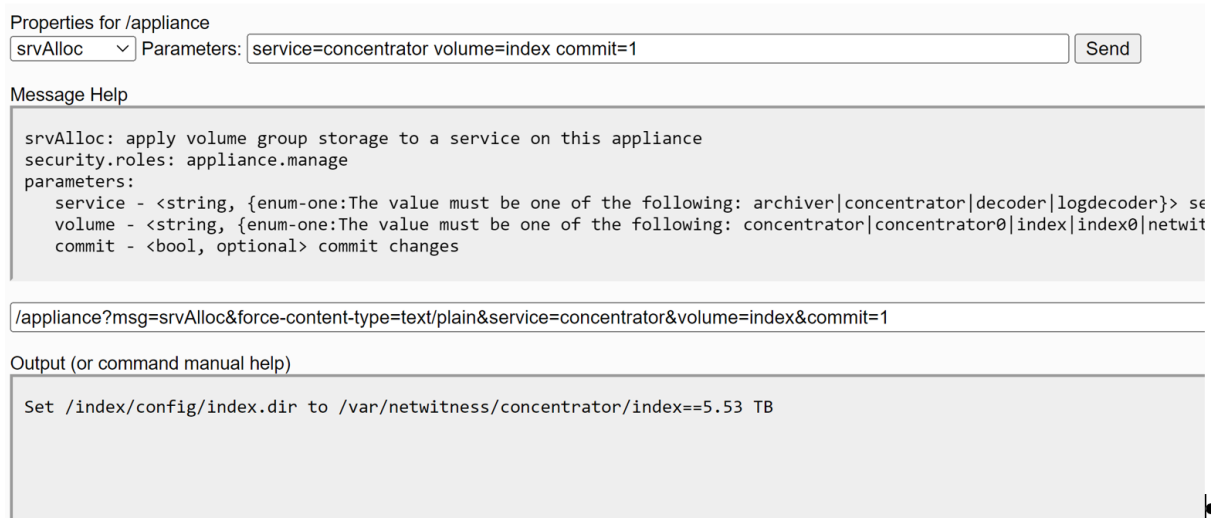


22. Execute the `srvAlloc` commands with the following parameters.

`service=concentrator volume=concentrator commit=1`

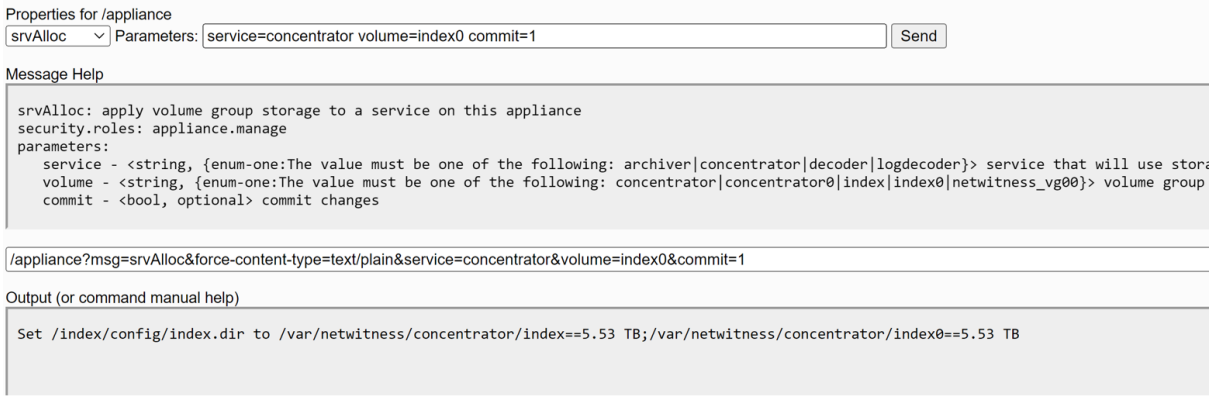


`service=concentrator volume=index commit=1`



`service=concentrator volume=concentrator0 commit=1`

```
service=concentrator volume=index0 commit=1
```



23. Execute `srvList` to verify storage has been assigned as required.

24. Verify partitions and mounts on `lsblk`.

```
[root@S7Concentrator ~]# lsblk
NAME                        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                           8:0    0   2.2T  0 disk
├─sda1                        8:1    0   256M  0 part /boot/efi
├─sda2                        8:2    0     1G  0 part /boot
└─sda3                        8:3    0   2.2T  0 part
  ├─netwitness_vg00-nwhome  253:0    0   4.3T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog  253:1    0    10G  0 lvm  /var/log
  ├─netwitness_vg00-usrhome 253:2    0    10G  0 lvm  /home
  ├─netwitness_vg00-root    253:3    0    30G  0 lvm  /
  └─netwitness_vg00-swap    253:4    0     4G  0 lvm  [SWAP]
sdb                           8:16   0   2.2T  0 disk
└─sdb1                        8:17   0   2.2T  0 part
  └─netwitness_vg00-nwhome  253:0    0   4.3T  0 lvm  /var/netwitness
sdc                           8:32   0 145.5T  0 disk
└─sdc1                        8:33   0 145.5T  0 part
  ├─concentrator-root       253:5    0    30G  0 lvm  /var/netwitness/concentrator
  ├─concentrator-sessiondb  253:6    0  14.6T  0 lvm  /var/netwitness/concentrator/sessiondb
  └─concentrator-metadb     253:7    0   131T  0 lvm  /var/netwitness/concentrator/metadb
sdd                           8:48   0   5.8T  0 disk
└─sdd1                        8:49   0   5.8T  0 part
  └─index-index             253:8    0   5.8T  0 lvm  /var/netwitness/concentrator/index
sde                           8:64   0 145.5T  0 disk
└─sde1                        8:65   0 145.5T  0 part
  ├─concentrator0-sessiondb 253:9    0  14.6T  0 lvm  /var/netwitness/concentrator/sessiondb0
  └─concentrator0-metadb    253:10   0   131T  0 lvm  /var/netwitness/concentrator/metadb0
sdf                           8:80   0   5.8T  0 disk
└─sdf1                        8:81   0   5.8T  0 part
  └─index0-index            253:11   0   5.8T  0 lvm  /var/netwitness/concentrator/index0
```

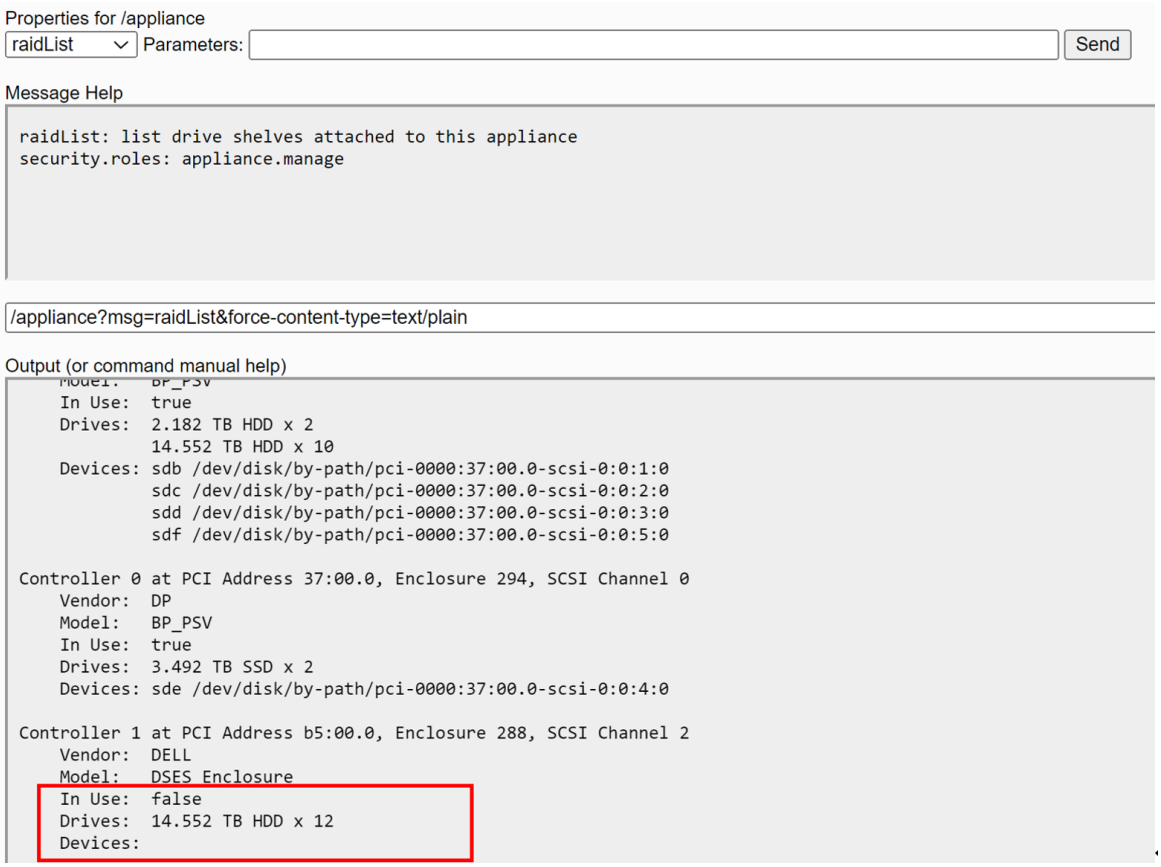
Configure Storage for Log Hybrid using MD2412

The following scenario configures storage on one unencrypted, 12-Drive MD2412 PowerVault for a Log Hybrid physical host.

1. Execute the `raidList` command.

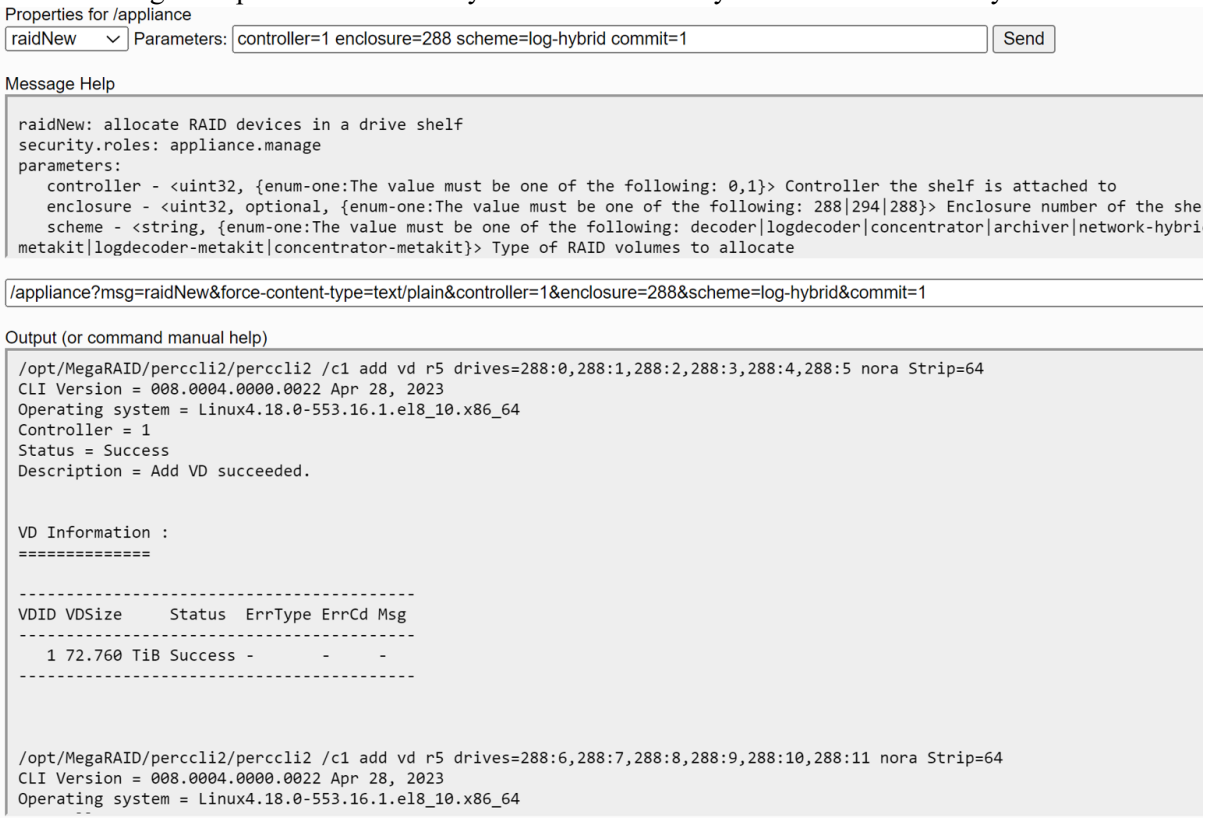
   a. Record the Controller Number, Enclosure Number, In Use, Drives, and Devices.
   You should see the following information.
   ```
   In Use: FALSE
   Devices: <empty>
   ```

   b. Verify the Drive Count, Size, and Vendor.
   The following example illustrates what you should see before you create a RAID array.

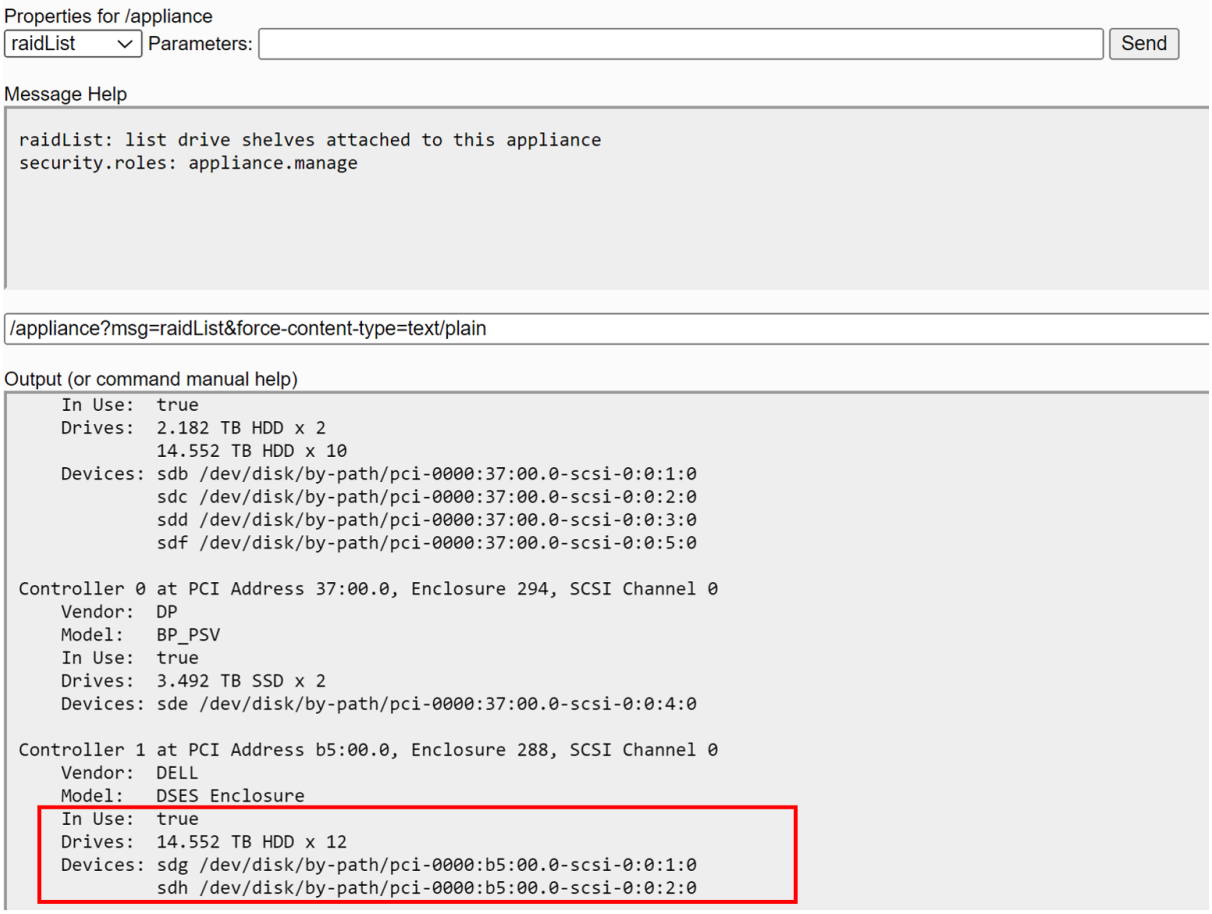


2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
   ```
   controller=1 enclosure=288 scheme=log-hybrid commit=1
   ```

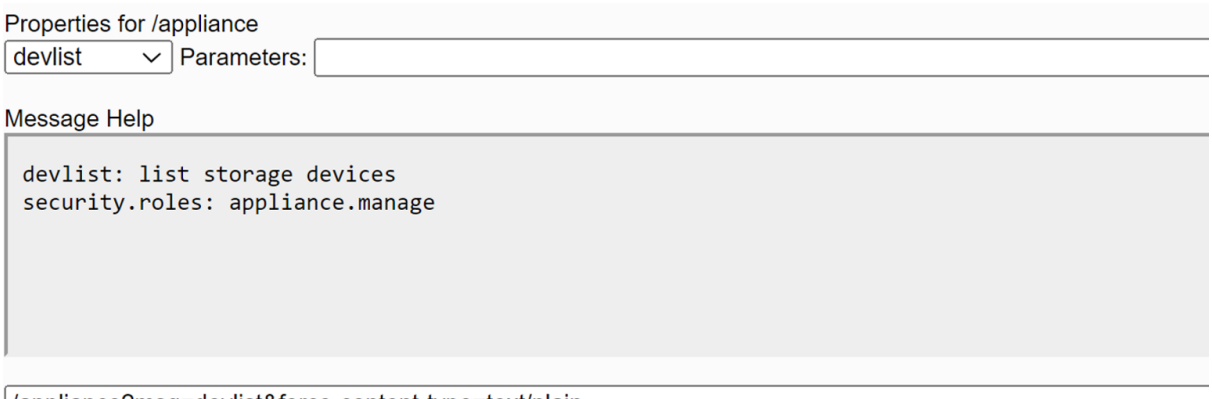
The following example illustrates what you should see after you create a RAID array.



Execute the `raidList` command to verify the RAID created.



Execute the `devlist` command to find the exact memory distribution between the newly created RAIDS. In Log Hybrid, the distribution is exactly equal so any service can be installed in any of the RAIDs.



Execute `lsblk` to verify the block devices created.

```
[root@S7LogHybrid ~]# lsblk
NAME                                  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                     8:0    1 14.3G  0 disk
└─sda1                                  8:1    1 14.3G  0 part
sdb                                     8:16   0 14.6T  0 disk
└─sdb1                                  8:17   0 14.6T  0 part
  └─hybrid--logdecoder--meta-decoroot 253:8    0 14.6T  0 lvm  /var/netwitness/logdecoder
sdc                                     8:32   0 43.7T  0 disk
└─sdc1                                  8:33   0 43.7T  0 part
  └─logpacket-packetdb                253:7    0 43.7T  0 lvm  /var/netwitness/logdecoder/packetdb
sdd                                     8:48   0 43.7T  0 disk
└─sdd1                                  8:49   0 43.7T  0 part
  └─hybrid--concentrator-root         253:6    0 43.7T  0 lvm  /var/netwitness/concentrator
sde                                     8:64   0  3.5T  0 disk
└─sde1                                  8:65   0  3.5T  0 part
  └─index-index                       253:5    0  3.5T  0 lvm  /var/netwitness/concentrator/index
sdf                                     8:80   0  2.2T  0 disk
├─sdf1                                  8:81   0  256M  0 part /boot/efi
├─sdf2                                  8:82   0    1G  0 part /boot
└─sdf3                                  8:83   0  2.2T  0 part
  ├─netwitness_vg00-nwhome            253:0    0  2.1T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog            253:1    0   10G  0 lvm  /var/log
  ├─netwitness_vg00-usrhome           253:2    0   10G  0 lvm  /home
  ├─netwitness_vg00-root              253:3    0   30G  0 lvm  /
  └─netwitness_vg00-swap              253:4    0    4G  0 lvm  [SWAP]
sdg                                     8:96   0 72.8T  0 disk
sdh                                     8:112  0 72.8T  0 disk
```

3. Execute the `partNew` commandwith the following parameters to create partitions and mount points in the */etc/fstab* file.

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

Properties for /appliance

partNew Parameters: name=sdg service=logdecoder volume=logdecoder commit=1 Send

Message Help

```
partNew: make partitions on a block device
security.roles: appliance.manage
parameters:
   name - <string, {enum-one:The value must be one of the following: sdg|sdh}> block device name
   service - <string, {enum-one:The value must be one of the following: archiver|concentrator|decoder|logdecoder}> servi
   volume - <string, optional, {enum-one:The value must be one of the following: concentrator|index|decodersmall|decoder
hybrid|log-indexed-decoder|logindex}> volume to create
```

/appliance?msg=partNew&force-content-type=text/plain&name=sdg&service=logdecoder&volume=logdecoder&commit=1

Output (or command manual help)

```
/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
/sbin/lvcreate -y -n packetdb -l 100%FREE logdecoder0
  Wiping xfs signature on /dev/logdecoder0/packetdb.
  Logical volume "packetdb" created.
/sbin/mkfs.xfs /dev/logdecoder0/packetdb
meta-data=/dev/logdecoder0/packetdb isize=512    agcount=73, agsize=268435424 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0 inobtcount=0
data     =                       bsize=4096   blocks=19531693056, imaxpct=1
         =                       sunit=16     swidth=80 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=521728, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/logdecoder/packetdb0
/bin/mount /var/netwitness/logdecoder/packetdb0
```

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

Properties for /appliance

partNew Parameters: name=sdh service=concentrator volume=concentrator commit=1 Send

Message Help

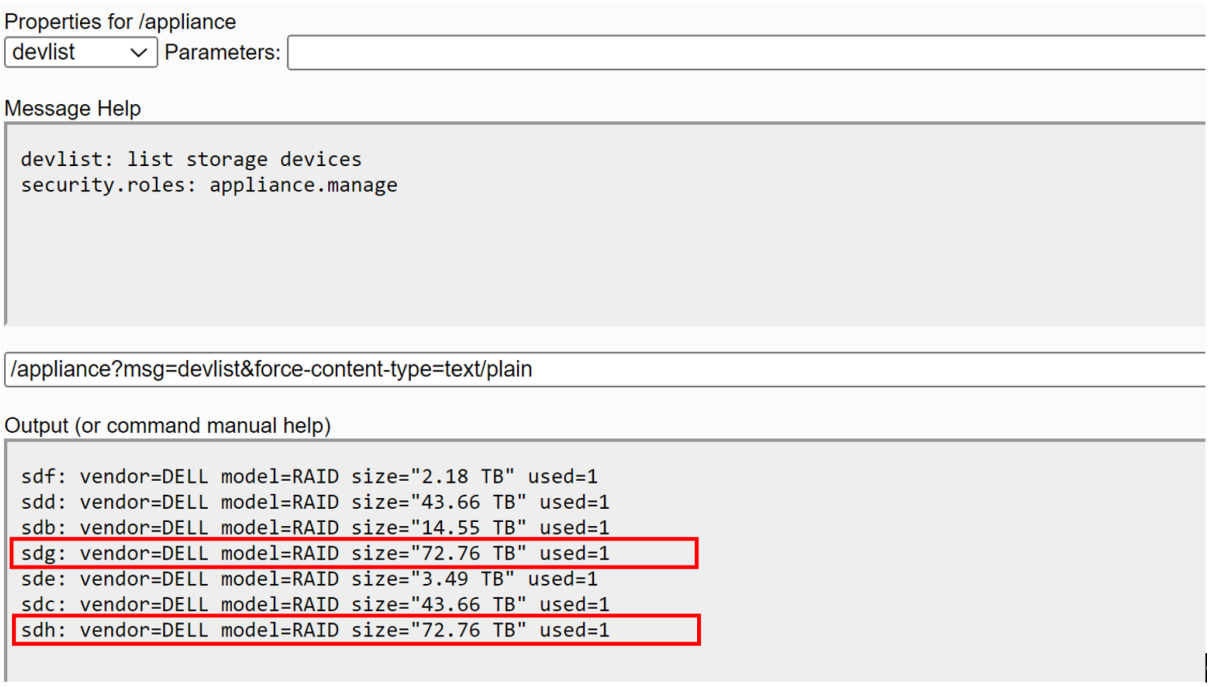
```
partNew: make partitions on a block device
security.roles: appliance.manage
parameters:
   name - <string, {enum-one:The value must be one of the following: sdg|sdh}> block device name
   service - <string, {enum-one:The value must be one of the following: archiver|concentrator|decoder|logdecoder}> service that will use sto
   volume - <string, optional, {enum-one:The value must be one of the following: concentrator|index|decodersmall|decoder|packet|hybrid-decoc
hybrid|log-indexed-decoder|logindex}> volume to create
```

/appliance?msg=partNew&force-content-type=text/plain&name=sdh&service=concentrator&volume=concentrator&commit=1

Output (or command manual help)

```
/sbin/parted -s /dev/sdh mklabel gpt
/sbin/parted -s -a optimal /dev/sdh mkpart LVM 0% 100%
/sbin/pvcreate -f /dev/sdh1
  Physical volume "/dev/sdh1" successfully created.
/sbin/vgcreate -f concentrator0 /dev/sdh1
  Volume group "concentrator0" successfully created
/sbin/lvcreate -y -n sessiondb -l 10%FREE concentrator0
  Wiping xfs signature on /dev/concentrator0/sessiondb.
  Logical volume "sessiondb" created.
/sbin/mkfs.xfs /dev/concentrator0/sessiondb
meta-data=/dev/concentrator0/sessiondb isize=512    agcount=32, agsize=61036512 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0 inobtcount=0
data     =                       bsize=4096   blocks=1953168384, imaxpct=5
         =                       sunit=16     swidth=80 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=521728, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
/bin/mkdir -p /var/netwitness/concentrator/sessiondb0
/bin/mount /var/netwitness/concentrator/sessiondb0
/sbin/lvcreate -y -n metadb -l 100%FREE concentrator0
```

Execute `devlist` to verify partitions created.



4. Execute the `srvAlloc` command with the following parameters to allocate the space to logdecoder and concentrator services.

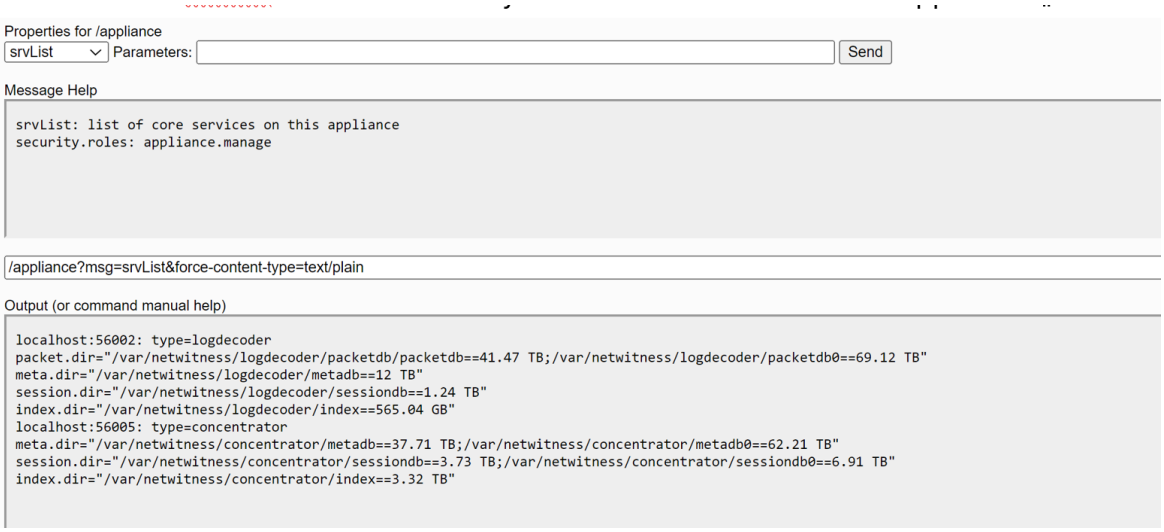
   a. This adds storage to the logdecoder service configuration and restarts the service every time it is executed.

   `service=logdecoder volume=logdecoder0 commit=1`

   b. This adds storage to concentrator service configuration and restarts the service every time it is executed.

   `service=concentrator volume=concentrator0 commit=1`

c. Execute the `srvList` command to verify the services allocated on the appliance.



5. Execute the `lsblk` command in backend to see all the raids and partitions inside the service.

```
[root@S7LogHybrid ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                   8:0    1 14.3G  0 disk
└─sda1                                8:1    1 14.3G  0 part
sdb                                   8:16   0 14.6T  0 disk
└─sdb1                                8:17   0 14.6T  0 part
  └─hybrid--logdecoder--meta-decoroot 253:8    0 14.6T  0 lvm  /var/netwitness/logdecoder
sdc                                   8:32   0 43.7T  0 disk
└─sdc1                                8:33   0 43.7T  0 part
  └─logpacket-packetdb                253:7    0 43.7T  0 lvm  /var/netwitness/logdecoder/packetdb
sdd                                   8:48   0 43.7T  0 disk
└─sdd1                                8:49   0 43.7T  0 part
  └─hybrid--concentrator-root         253:6    0 43.7T  0 lvm  /var/netwitness/concentrator
sde                                   8:64   0  3.5T  0 disk
└─sde1                                8:65   0  3.5T  0 part
  └─index-index                       253:5    0  3.5T  0 lvm  /var/netwitness/concentrator/index
sdf                                   8:80   0  2.2T  0 disk
├─sdf1                                8:81   0  256M  0 part /boot/efi
├─sdf2                                8:82   0    1G  0 part /boot
└─sdf3                                8:83   0  2.2T  0 part
  ├─netwitness_vg00-nwhome            253:0    0  2.1T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog            253:1    0   10G  0 lvm  /var/log
  ├─netwitness_vg00-usrhome           253:2    0   10G  0 lvm  /home
  ├─netwitness_vg00-root              253:3    0   30G  0 lvm  /
  └─netwitness_vg00-swap              253:4    0    4G  0 lvm  [SWAP]
sdg                                   8:96   0 72.8T  0 disk
└─sdg1                                8:97   0 72.8T  0 part
  └─logdecoder0-packetdb              253:9    0 72.8T  0 lvm  /var/netwitness/logdecoder/packetdb0
sdh                                   8:112  0 72.8T  0 disk
└─sdh1                                8:113  0 72.8T  0 part
  ├─concentrator0-sessiondb           253:10   0  7.3T  0 lvm  /var/netwitness/concentrator/sessiondb0
  └─concentrator0-metadb              253:11   0 65.5T  0 lvm  /var/netwitness/concentrator/metadb0
```

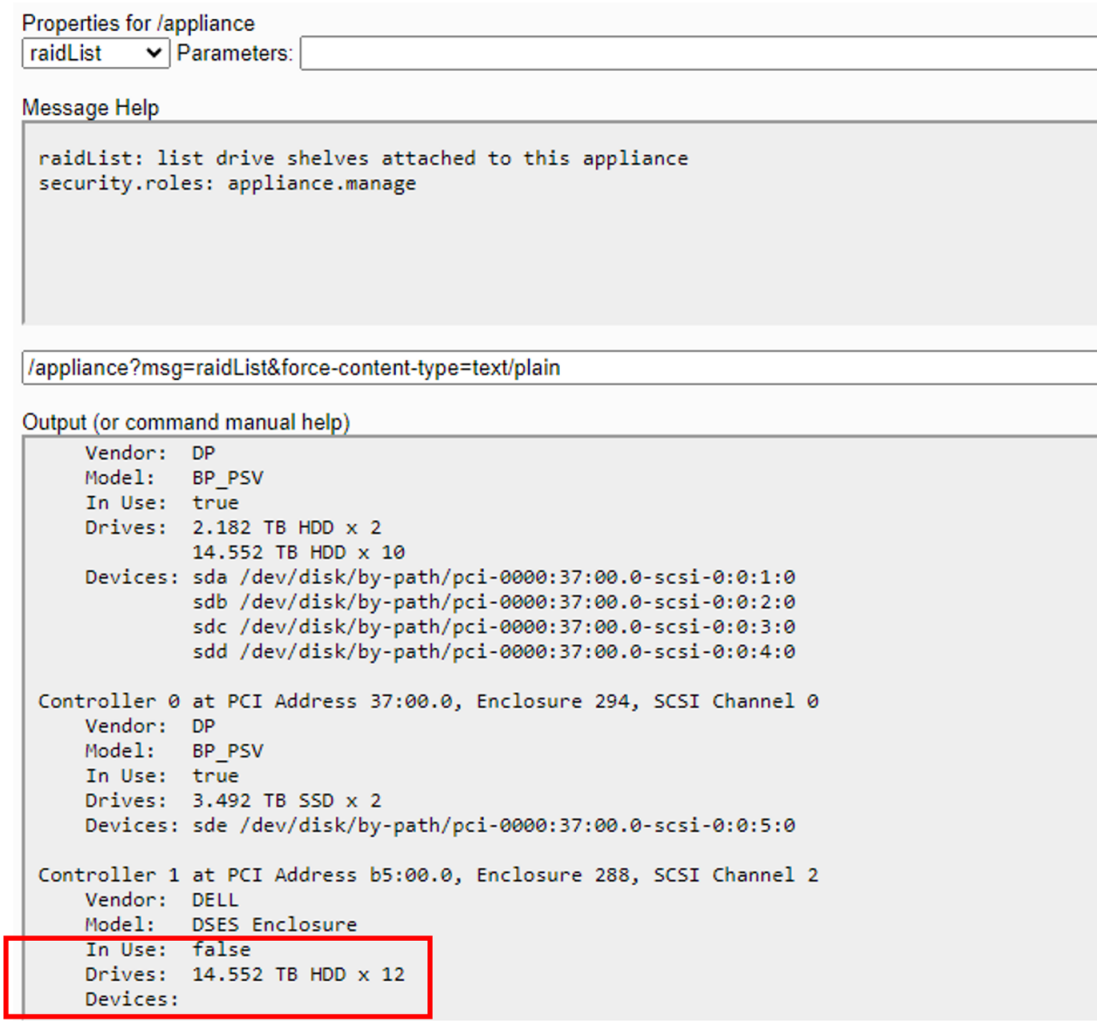
Configure Storage for Network Hybrid using MD2412

The following scenario configures storage on an unencrypted, 12-Drive MD2412 PowerVault for a Network Hybrid physical host.

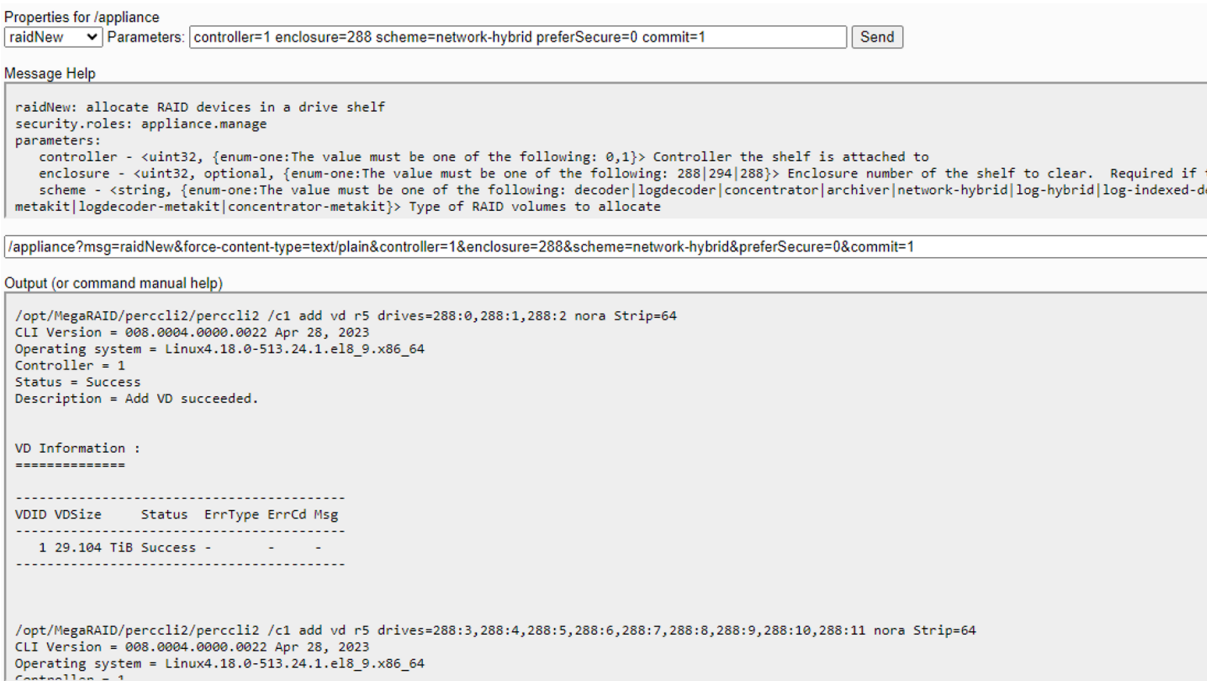
1. Execute the `raidList` command.

   a. Record the Controller Number, Enclosure Number, In Use, Drives, and Devices.
   You should see the following information.
   ```
   In Use: FALSE
   Devices: <empty>
   ```

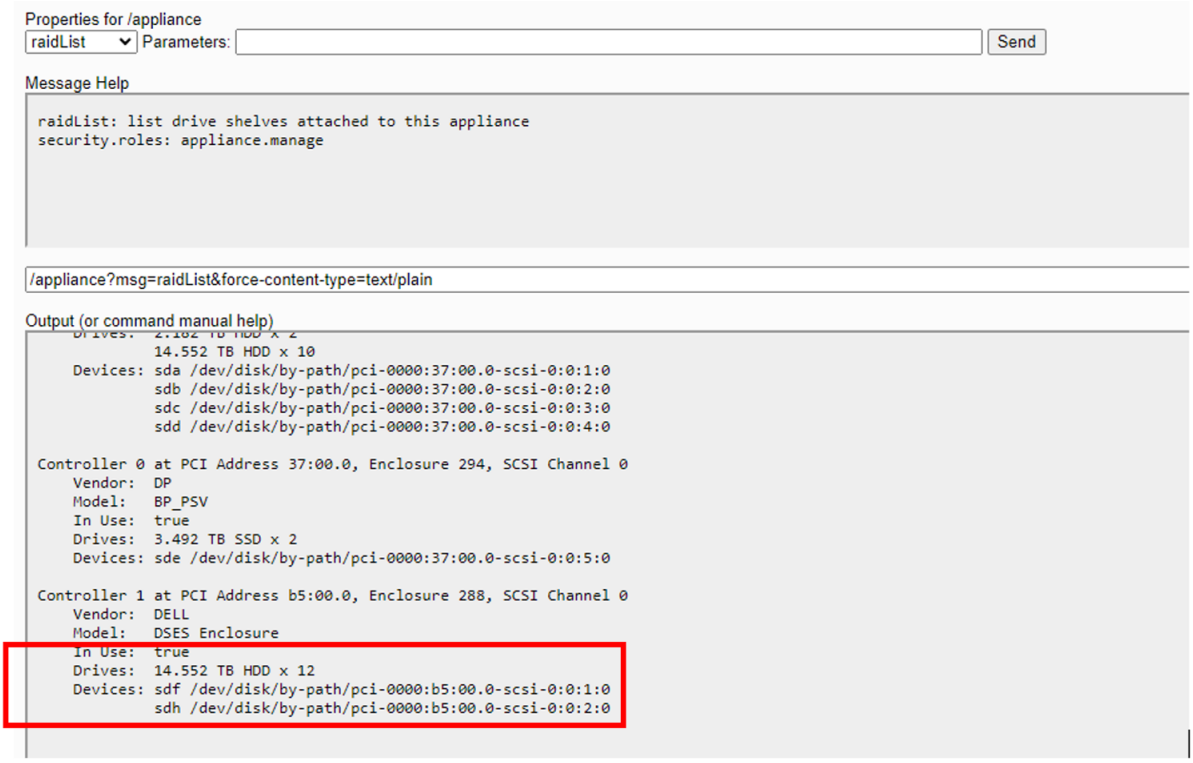
   b. Verify the Drive Count, Size, and Vendor.
   The following example illustrates what you should see before you create a RAID array.



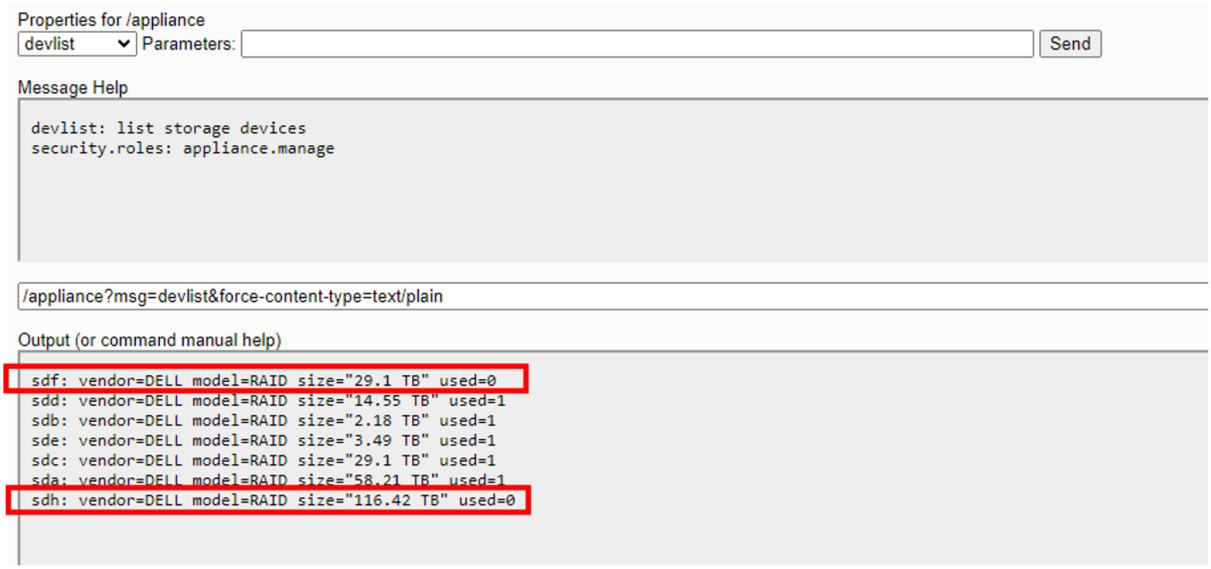
2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.
   ```
   controller=1 enclosure=288 scheme=network-hybrid preferSecure=0 commit=1
   ```
   The following example illustrates what you should see after you create a RAID array.



Following is the response from raidList after creating RAID array.

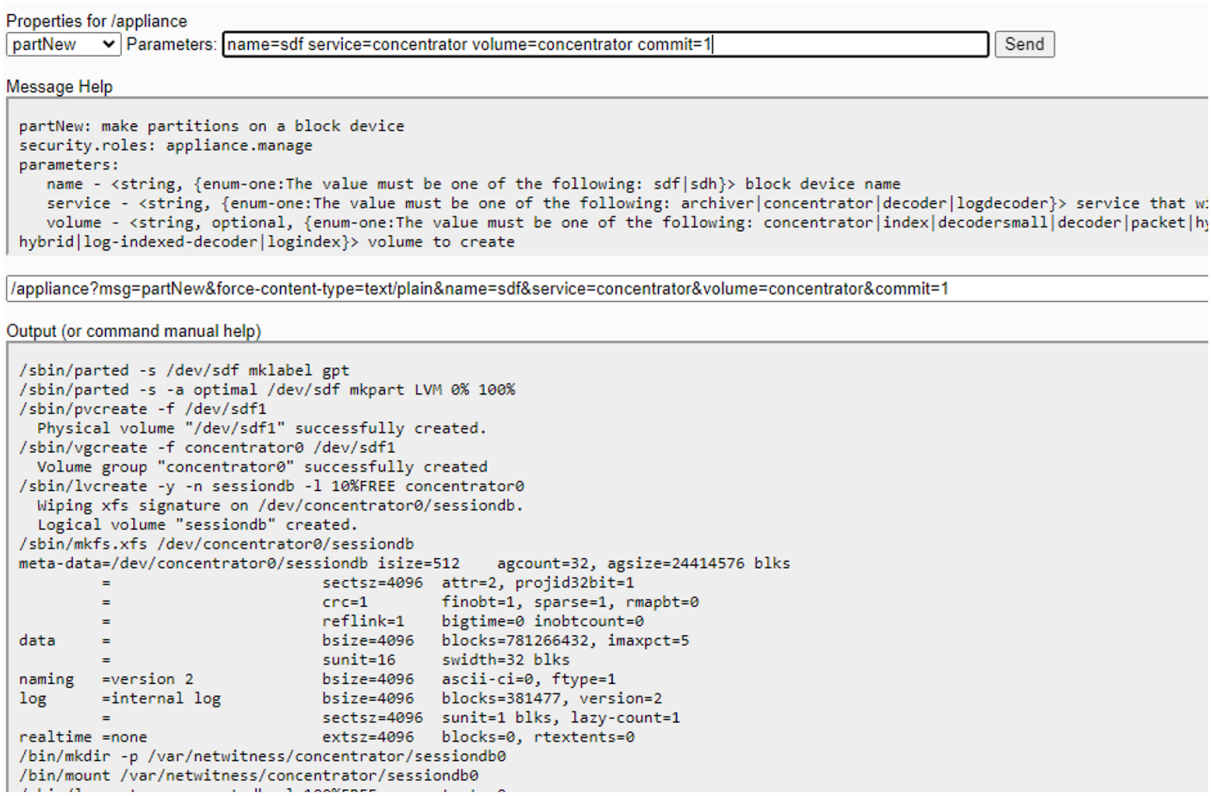


Execute `devlist` command to find the exact memory distribution between the newly created RAIDs. In Network Hybrid, the memory for ‘sdh’ block device is more than ‘sdf’ block device, therefore packetdb will be installed in the device with higher memory allocated which is ‘sdh’ .

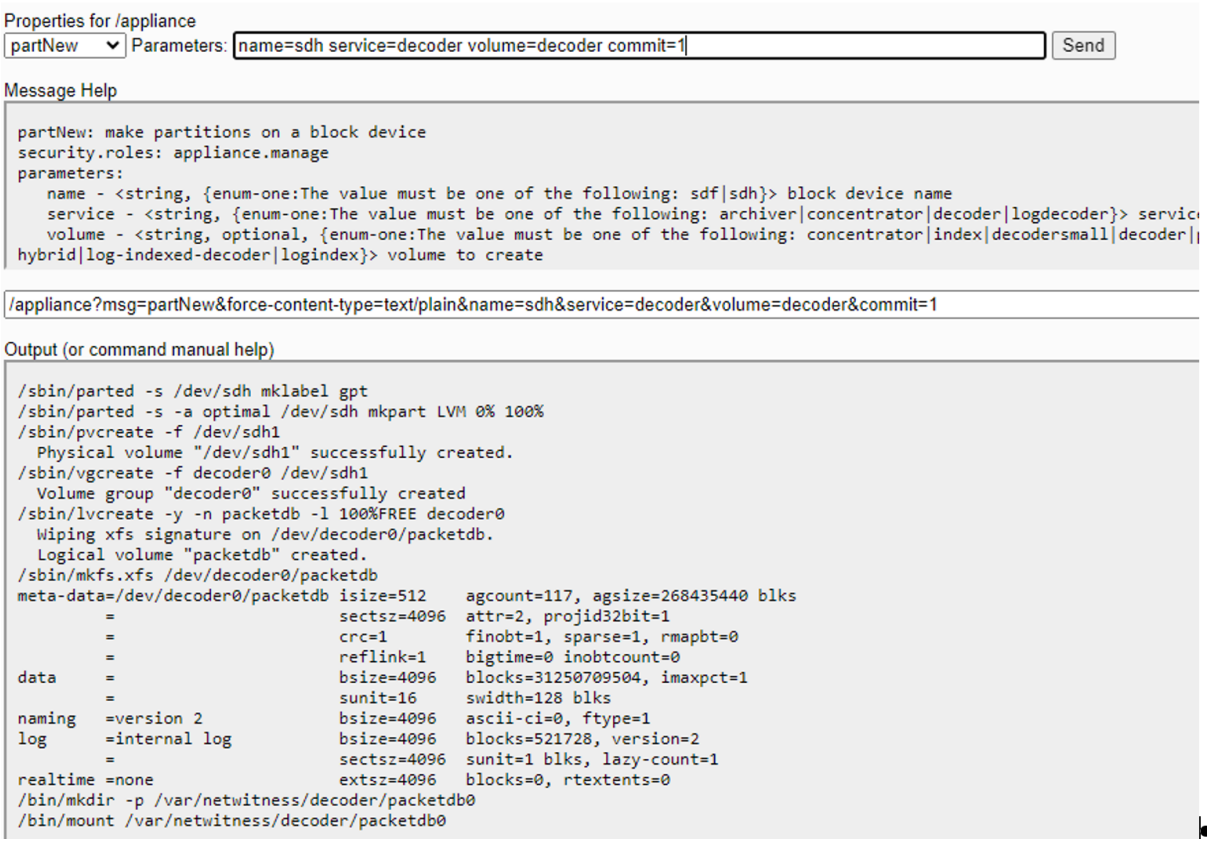


3. Execute the `partNew` command with the following parameters to create partitions and mount points in the */etc/fstab* file.

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



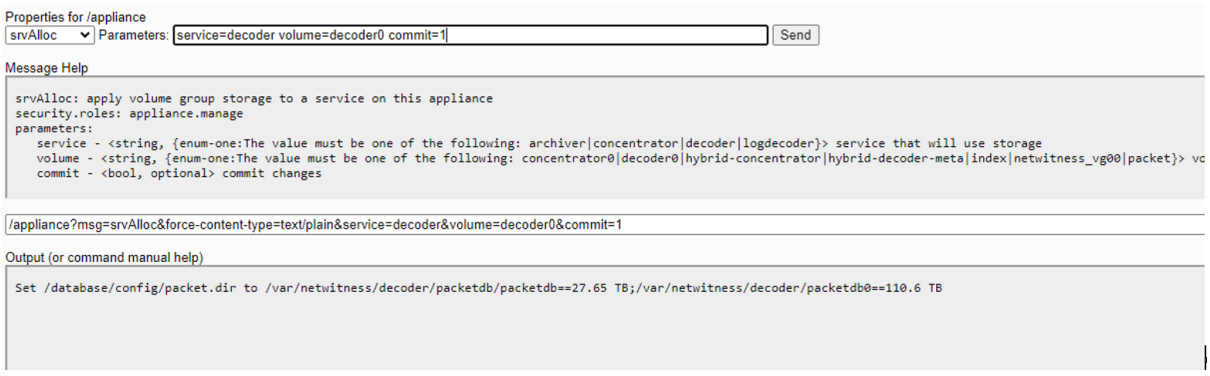

```
name=sdh service=decoder volume=decoder commit=1
```



4. Execute the `srvAlloc` command with the following parameters to allocate the space to decoder and concentrator services.

   This adds storage to the decoder service configuration and restarts the service every time it is executed.

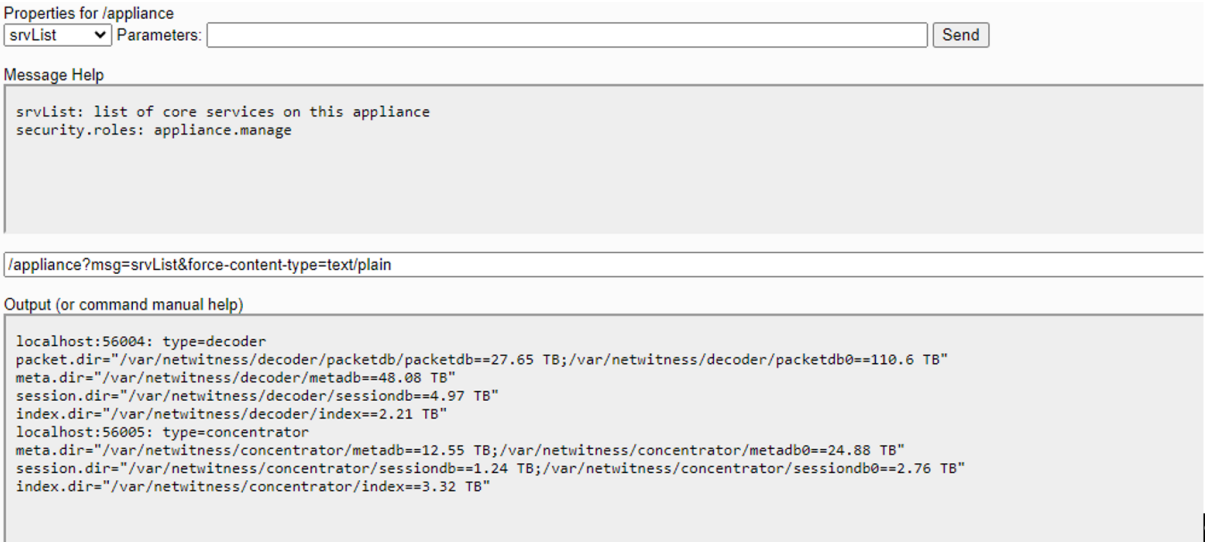
   ```
   service=decoder volume=decoder0 commit=1
   ```



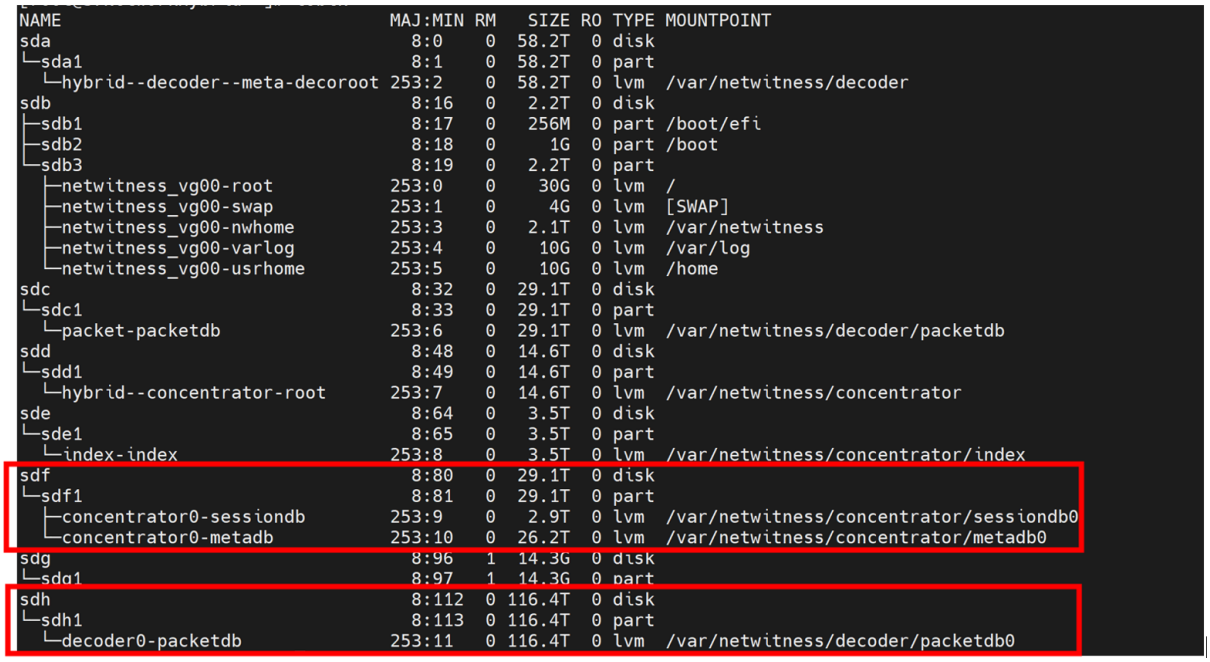
   This adds storage to the concentrator service configuration and restarts the service every time it is executed.

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

Execute the `srvList` command to verify the services allocated on the appliance.



5. Execute `lsblk` command to list all the raids and partitions in the service.



```
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                                   8:0    0  58.2T  0 disk
└─sda1                                8:1    0  58.2T  0 part
  └─hybrid--decoder--meta-decoroot  253:2    0  58.2T  0 lvm  /var/netwitness/decoder
sdb                                   8:16   0   2.2T  0 disk
├─sdb1                                8:17   0   256M  0 part /boot/efi
├─sdb2                                8:18   0     1G  0 part /boot
└─sdb3                                8:19   0   2.2T  0 part
  ├─netwitness_vg00-root            253:0    0    30G  0 lvm  /
  ├─netwitness_vg00-swap            253:1    0     4G  0 lvm  [SWAP]
  ├─netwitness_vg00-nwhome          253:3    0   2.1T  0 lvm  /var/netwitness
  ├─netwitness_vg00-varlog          253:4    0    10G  0 lvm  /var/log
  └─netwitness_vg00-usrhome         253:5    0    10G  0 lvm  /home
sdc                                   8:32   0  29.1T  0 disk
└─sdc1                                8:33   0  29.1T  0 part
  └─packet-packetdb                 253:6    0  29.1T  0 lvm  /var/netwitness/decoder/packetdb
sdd                                   8:48   0  14.6T  0 disk
└─sdd1                                8:49   0  14.6T  0 part
  └─hybrid--concentrator-root       253:7    0  14.6T  0 lvm  /var/netwitness/concentrator
sde                                   8:64   0   3.5T  0 disk
└─sde1                                8:65   0   3.5T  0 part
  └─index-index                     253:8    0   3.5T  0 lvm  /var/netwitness/concentrator/index
sdf                                   8:80   0  29.1T  0 disk
└─sdf1                                8:81   0  29.1T  0 part
  ├─concentrator0-sessiondb         253:9    0   2.9T  0 lvm  /var/netwitness/concentrator/sessiondb0
  └─concentrator0-metadb            253:10   0  26.2T  0 lvm  /var/netwitness/concentrator/metadb0
sdg                                   8:96   1  14.3G  0 disk
└─sdg1                                8:97   1  14.3G  0 part
sdh                                   8:112  0 116.4T  0 disk
└─sdh1                                8:113  0 116.4T  0 part
  └─decoder0-packetdb               253:11   0 116.4T  0 lvm  /var/netwitness/decoder/packetdb0
```

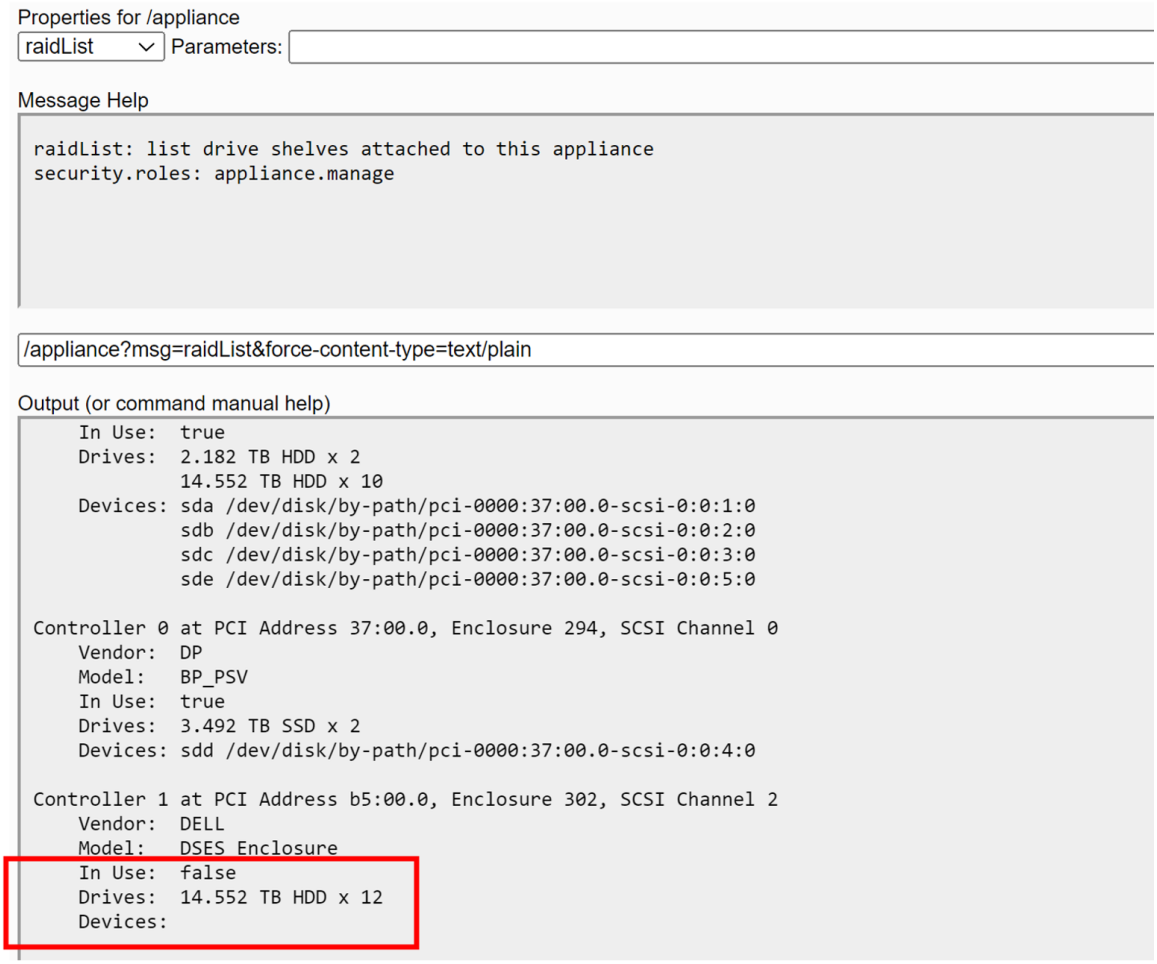
Configure Storage for Endpoint Log Hybrid using MD2412

The following scenario configures storage on one unencrypted, 12-Drive MD2412 PowerVault for an Endpoint Log Hybrid 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.

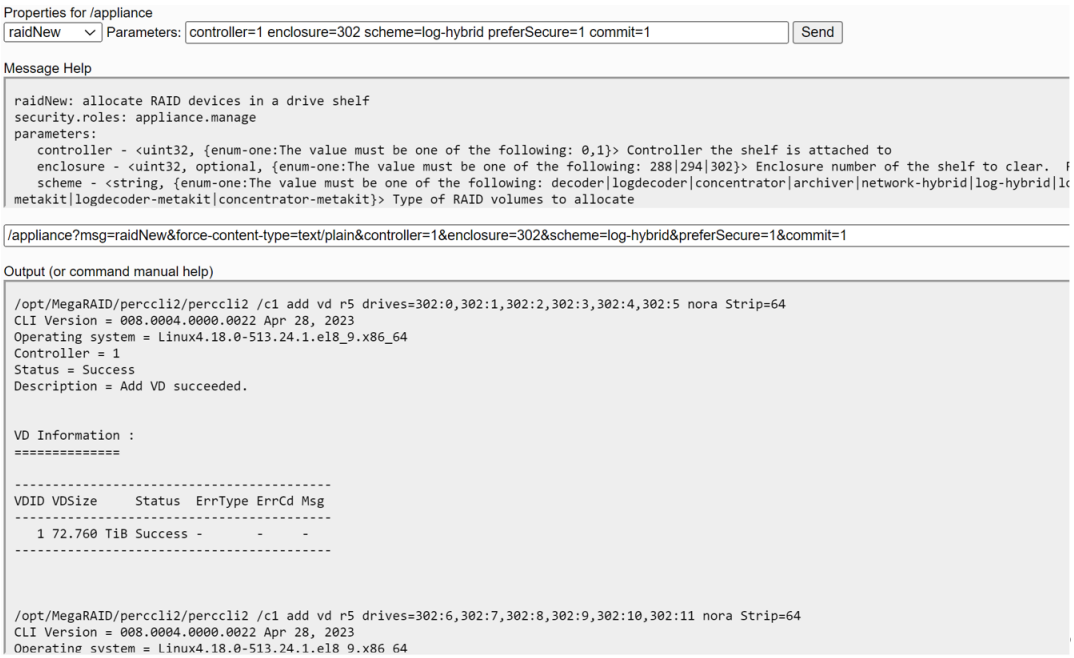


2. Execute the `raidNew` command with the following parameters using the controller number and the enclosure number you just recorded.

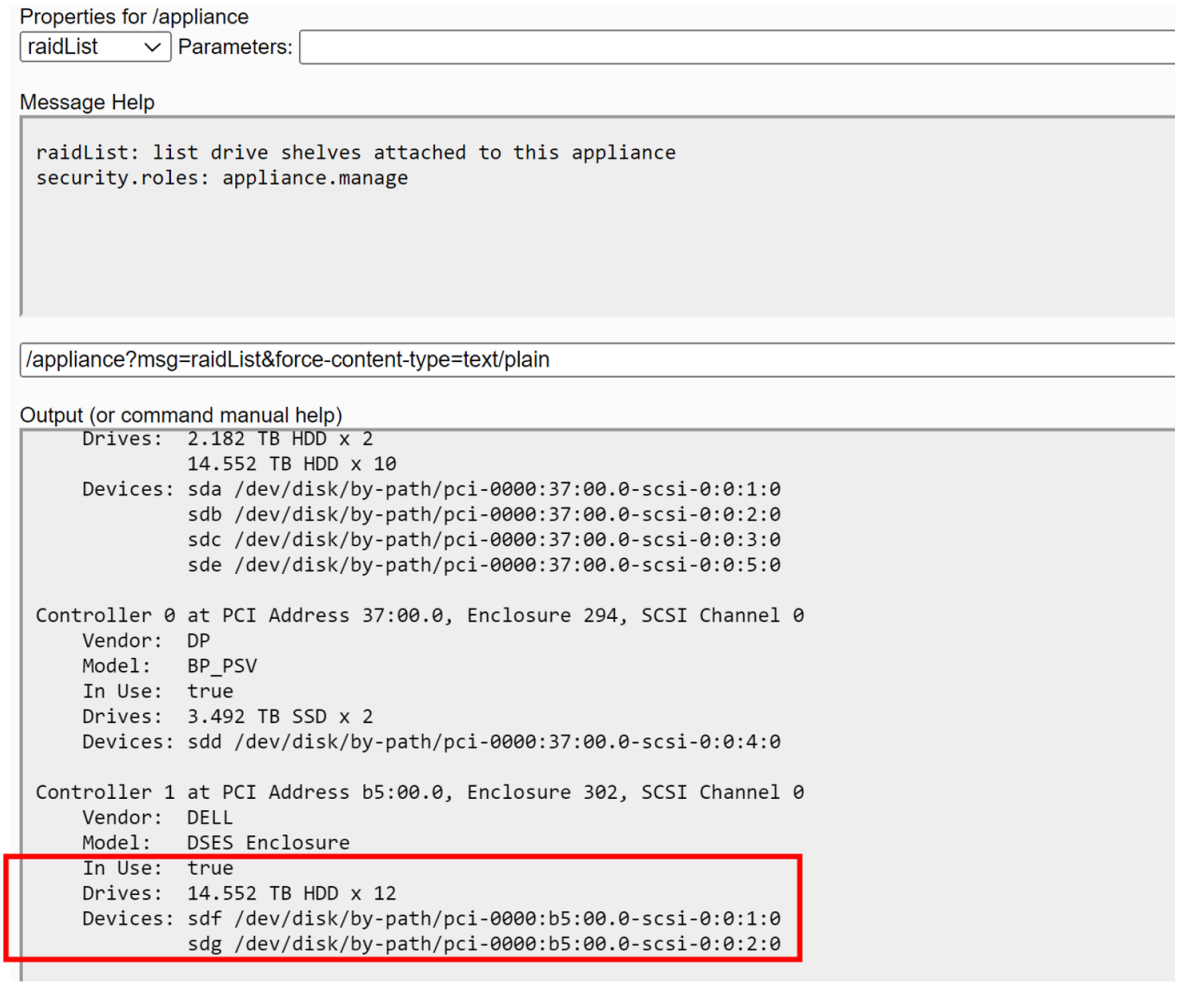
   In this case, the security key has been set on the controller PERCH965e. Hence, `preferSecure=1` parameter is used to enable encryption. If the security key is not set on the controller, parameter `preferSecure=0` can be used to complete configuring RAID on the controller.

   ```
   controller=1 enclosure=302 scheme=log-hybrid preferSecure=1 commit=1
   ```

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



Following is the response from raidList after creating RAID array.



Execute the `devlist` command to find the exact memory distribution for the newly created raids. In Endpoint Log Hybrid, both are exactly equal. Hence, any service can be installed in any of the raid.

Properties for /appliance

devlist Parameters:

Message Help

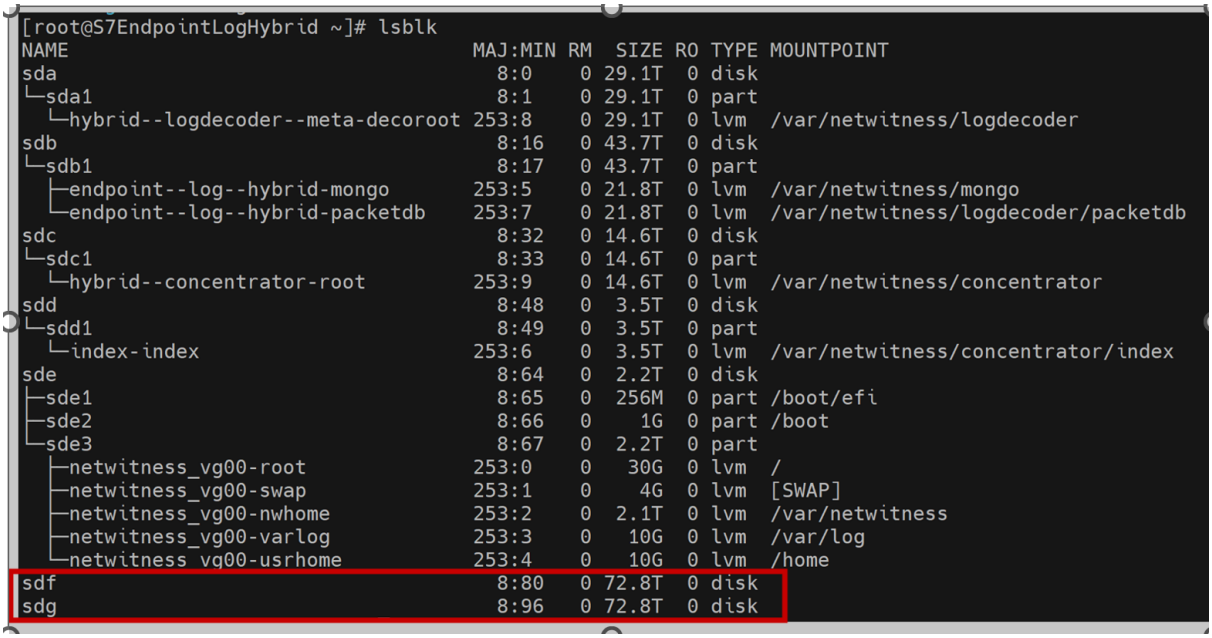
```
devlist: list storage devices
security.roles: appliance.manage
```

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

Output (or command manual help)

```
sdf: vendor=DELL model=RAID size="72.76 TB" used=0
sdd: vendor=DELL model=RAID size="3.49 TB" used=1
sdb: vendor=DELL model=RAID size="43.66 TB" used=1
sdg: vendor=DELL model=RAID size="72.76 TB" used=0
sde: vendor=DELL model=RAID size="2.18 TB" used=1
sdc: vendor=DELL model=RAID size="14.55 TB" used=1
sda: vendor=DELL model=RAID size="29.1 TB" used=1
```

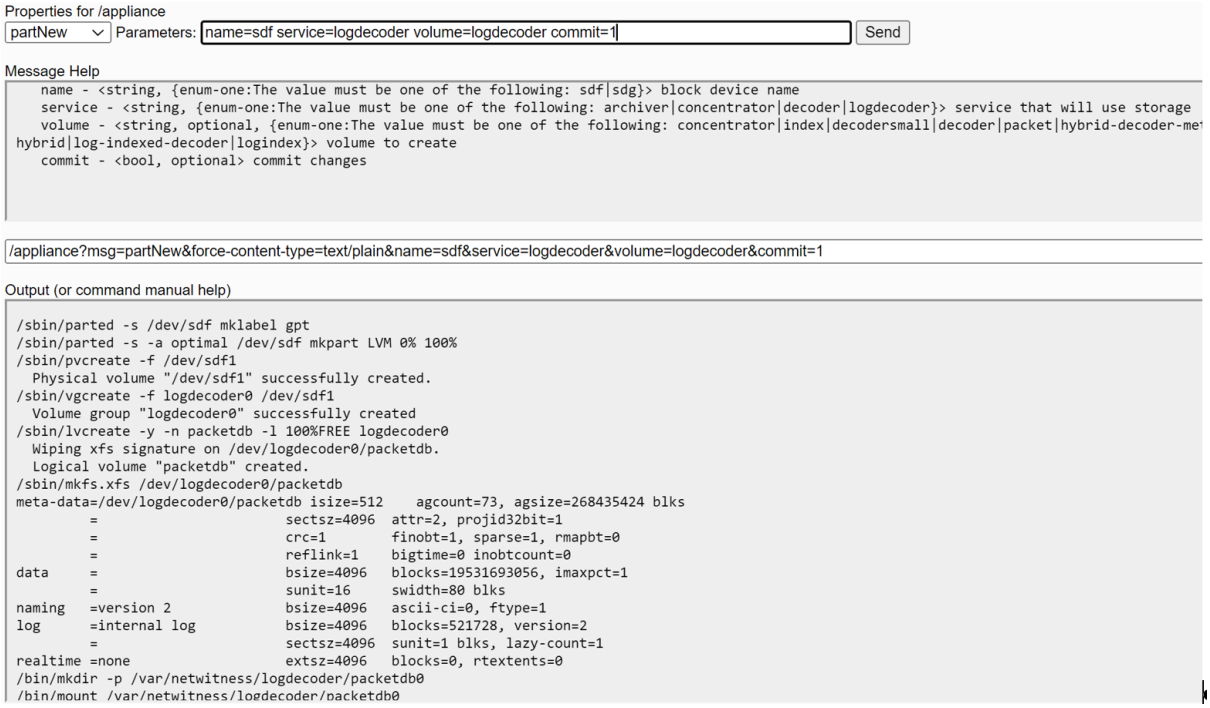
Execute the `lsblk` command at backend to list the newly created raids on the Endpoint Log Hybrid appliance.



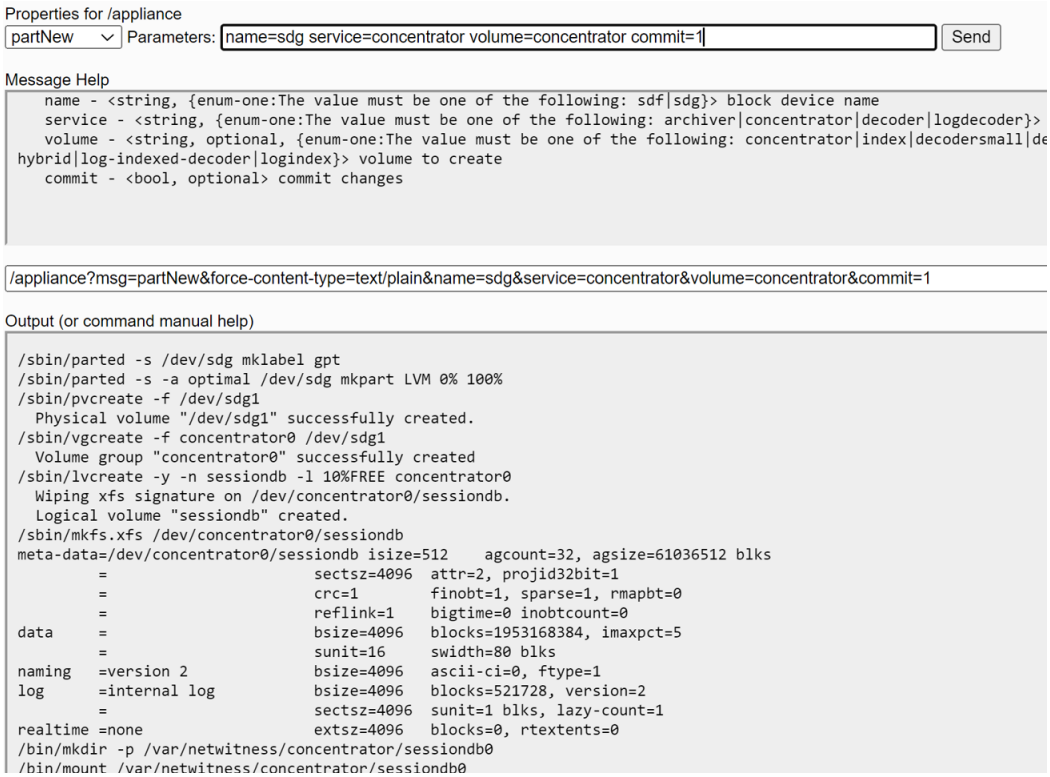
```
[root@S7EndpointLogHybrid ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                   8:0    0 29.1T  0 disk
└─sda1                                8:1    0 29.1T  0 part
  └─hybrid--logdecoder--meta-decoroot 253:8    0 29.1T  0 lvm  /var/netwitness/logdecoder
sdb                                   8:16   0 43.7T  0 disk
└─sdb1                                8:17   0 43.7T  0 part
  ├─endpoint--log--hybrid-mongo       253:5    0 21.8T  0 lvm  /var/netwitness/mongo
  └─endpoint--log--hybrid-packetdb    253:7    0 21.8T  0 lvm  /var/netwitness/logdecoder/packetdb
sdc                                   8:32   0 14.6T  0 disk
└─sdc1                                8:33   0 14.6T  0 part
  └─hybrid--concentrator-root         253:9    0 14.6T  0 lvm  /var/netwitness/concentrator
sdd                                   8:48   0  3.5T  0 disk
└─sdd1                                8:49   0  3.5T  0 part
  └─index-index                       253:6    0  3.5T  0 lvm  /var/netwitness/concentrator/index
sde                                   8:64   0  2.2T  0 disk
├─sde1                                8:65   0  256M  0 part /boot/efi
├─sde2                                8:66   0    1G  0 part /boot
└─sde3                                8:67   0  2.2T  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  2.1T  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
sdf                                   8:80   0 72.8T  0 disk
sdg                                   8:96   0 72.8T  0 disk
```

3. Execute the `partNew` command with the following parameters to create partitions and mount points as required.

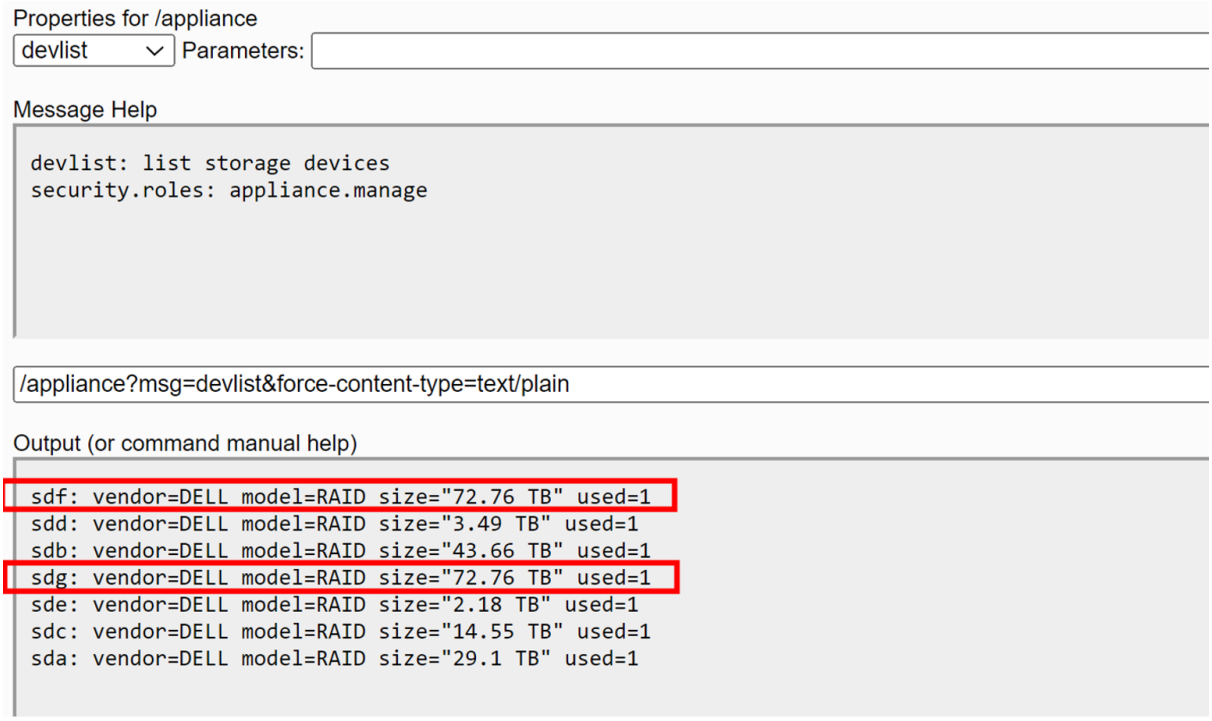
   ```
   name=sdf service=logdecoder volume=logdecoder commit=1
   ```



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



Execute the `devlist` command to verify partitions created.



4. Execute the `srvAlloc` command with the following parameters to allocate the space to logdecoder and concentrator services.

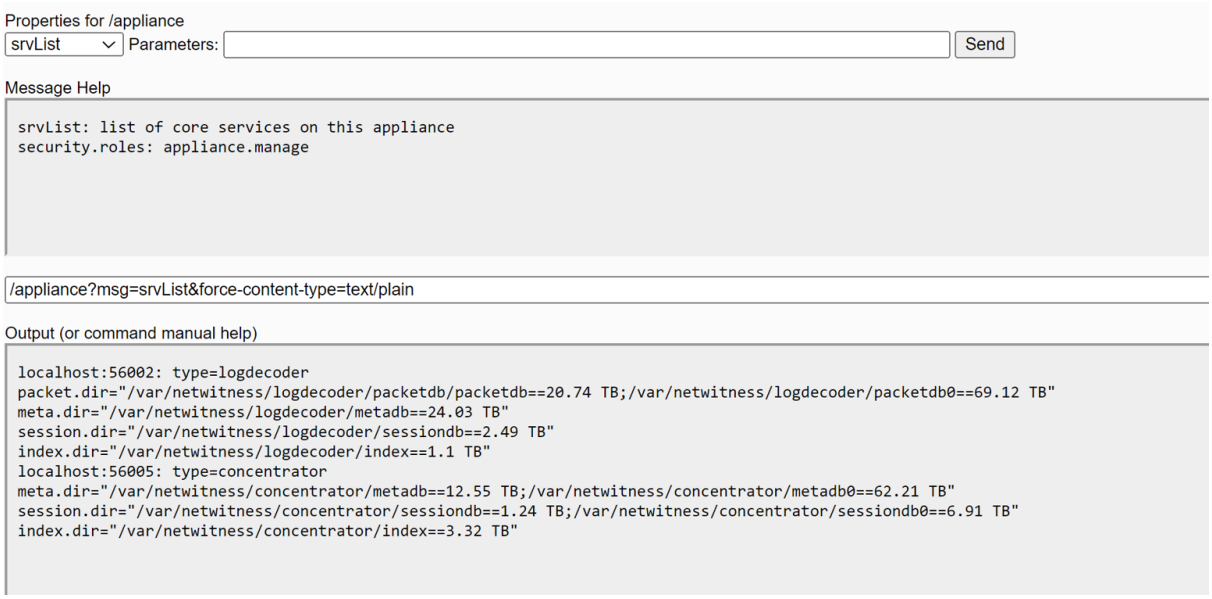
   This adds storage to the logdecoder service configuration and restarts the service every time it is executed.

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

   This adds storage to the concentrator service configuration and restarts the service every time it is executed.

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

Execute the `srvList` command to verify the services allocated on the appliance.



5. Execute `lsblk` command to list all the raids and partitions in the service.

```
[root@S7EndpointLogHybrid ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                   8:0    0 29.1T  0 disk
└─sda1                                8:1    0 29.1T  0 part
  └─hybrid--logdecoder--meta-decoroot 253:8    0 29.1T  0 lvm  /var/netwitness/logdecoder
sdb                                   8:16   0 43.7T  0 disk
└─sdb1                                8:17   0 43.7T  0 part
  ├─endpoint--log--hybrid-mongo       253:5    0 21.8T  0 lvm  /var/netwitness/mongo
  └─endpoint--log--hybrid-packetdb    253:7    0 21.8T  0 lvm  /var/netwitness/logdecoder/packetdb
sdc                                   8:32   0 14.6T  0 disk
└─sdc1                                8:33   0 14.6T  0 part
  └─hybrid--concentrator-root         253:9    0 14.6T  0 lvm  /var/netwitness/concentrator
sdd                                   8:48   0  3.5T  0 disk
└─sdd1                                8:49   0  3.5T  0 part
  └─index-index                       253:6    0  3.5T  0 lvm  /var/netwitness/concentrator/index
sde                                   8:64   0  2.2T  0 disk
├─sde1                                8:65   0  256M  0 part /boot/efi
├─sde2                                8:66   0    1G  0 part /boot
└─sde3                                8:67   0  2.2T  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  2.1T  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
sdf                                   8:80   0 72.8T  0 disk
└─sdf1                                8:81   0 72.8T  0 part
  └─logdecoder0-packetdb              253:10   0 72.8T  0 lvm  /var/netwitness/logdecoder/packetdb0
sdg                                   8:96   0 72.8T  0 disk
└─sdg1                                8:97   0 72.8T  0 part
  ├─concentrator0-sessiondb           253:11   0  7.3T  0 lvm  /var/netwitness/concentrator/sessiondb0
  └─concentrator0-metadb              253:12   0 65.5T  0 lvm  /var/netwitness/concentrator/metadb0
```