



RSA[®] NetWitness Platform

Version 11.6

ストレージガイド



連絡先情報

RSA Link(<https://community.rsa.com>)では、よくある質問への回答や、既知の問題の解決方法を含むナレッジベースを公開しています。また、製品ドキュメント、コミュニティディスカッション、ケース管理なども公開されています。

商標

RSA Conferenceのロゴ、RSA、その他の商標は、RSA Security LLCまたはその関連会社(「RSA」)の商標です。RSAの商標のリストについては、<https://www.rsa.com/ja-jp/company/rsa-trademarks>を参照してください。その他の商標は、各社の商標または登録商標です。

使用許諾契約

本ソフトウェアと関連ドキュメントは、RSA Security LLCまたはその関連会社が著作権を保有しており、使用許諾契約に従って提供されます。本ソフトウェアと関連ドキュメントの使用と複製は、使用許諾契約の条項に従い、上記の著作権を侵害しない場合のみ許諾されます。本ソフトウェアと関連ドキュメント、およびその複製物を他人に提供することは一切認められません。

本使用許諾契約によって、本ソフトウェアと関連ドキュメントの所有権およびその他の知的財産権が譲渡されることはありません。本ソフトウェアと関連ドキュメントを不正に使用または複製した場合、民事および刑事上の責任が課せられる可能性があります。

本ソフトウェアは予告なく変更されることがありますので、あらかじめご承知おきください。

サードパーティライセンス

本製品にはRSA以外のサードパーティによって開発されたソフトウェアが含まれます。本製品内のサードパーティ製ソフトウェアに適用される使用許諾契約の内容については、RSA Linkの製品ドキュメントページで確認できます。本製品を使用することにより、本製品のユーザは、これらの使用許諾契約の条項に同意したものとみなされます。

暗号技術に関する注意

本製品には、暗号技術が組み込まれています。これらの暗号技術の使用、輸入、輸出は、各国の法律で禁止または制限されています。本製品を使用、輸入、輸出する場合は、各国における使用または輸出入に関する法律に従わなければなりません。

配布

本文書に記載される、RSA Security LLCまたはその関連会社(「RSA」)のいかなるソフトウェアの使用、複製、配布にも、適切なソフトウェアライセンスが必要です。

RSAは、本文書に記載される情報が、発行日時点で正確であるとみなしています。この情報は予告なく変更されることがあります。

本文書に記載される情報は、「現状有姿」の条件で提供されています。RSAでは、本文書に記載される情報に関するいかなる内容についても表明保証条項を設けず、特に、商品性や特定の目的に対する適応性に対する黙示的保証は提供しません。

© 2020 RSA Security LLC or its affiliates. All Rights Reserved.

7月 2021

目次

ストレージの概要	5
ストレージ要件	6
ドライブの仕様	6
必要なNetWitness Platformストレージ ボリューム	6
パフォーマンスに関する推奨事項	8
1秒あたりのI/O処理数	8
NetWitness Platformホストによるデータの保存方法の一般的な説明	8
物理ストレージの準備	9
ドライブ パックのブロック デバイスの構成	9
シリーズ6/6Eドライブ パックのメリット	9
Decoderメタのユース ケース	9
Concentratorインデックスのユース ケース	10
Decoder/Log Decoderのブロック デバイスの構成	10
Concentratorのブロック デバイスの構成	14
PowerVaultのブロック デバイスの構成	14
Decoder/Log Decoderのブロック デバイスの構成	14
Concentratorのブロック デバイスの構成	17
ストレージの構成	17
Decoder/LogDecoderのストレージの構成	17
Concentratorのストレージの構成	18
SED対応可能ドライブのセキュリティの有効化	18
仮想ストレージまたはクラウド ストレージの準備	19
Decoder、Log Decoder、Concentrator、Archiver	19
NW Server、ESA Primary、ESA Secondary、Malware Analysis	19
Log Collector	20
Endpoint Log Hybrid	20
追加のEndpoint Log Hybridパーティション	24
UEBA	26
REST APIを使用したストレージの構成	27
REST APIストレージ構成コマンド	27
ストレージ構成タスク	28
タスク1: ストレージをホストに接続し、REST APIストレージ構成コマンドにアクセスする	28
タスク2:(条件付き)PowerVaultおよびDAC用のRAID構成	30
タスク3: パーティション、ボリュームグループ、論理ボリュームにブロック デバイスを割り当てる	32
タスク4: ボリューム グループをNetWitnessサービスに割り当てる: srvAlloc	34

タスク5:(オプション)ストレージ構成を10G収集用に再構成する	34
Unityストレージの準備	37
タスク1 - Unisphereユーザー インターフェイス(UI)にアクセスする	38
タスク2 - プールを作成する	39
タスク3 - LUNを作成する	42
タスク4 - ホストを登録する	44
タスク5 - LUNをホストに割り当てる	46
タスク6 - PowerPathをインストールする	49
別のストレージ タイプへのデータの移行	51
Warm階層およびHot階層オプションを使用したデータ移行	51
サービスを停止する	51
PowerVaultを設定する	51
マウント ポイントを構成する	52
Warm階層とHot階層を設定する	53
DACを廃止する	55
DACからPowerVaultへのデータの移動	56
DACから移動した後のPowerVault上のデータ	59
付録A: NetWitness Platformホストによるデータの保存方法	60
Decoderホスト	60
Concentratorホスト	60
Archiverホスト	61
Hybridホスト	61
SAN構成のオプション	61
パフォーマンスに関する推奨事項	61
SEDドライブと非SEDドライブが混在するホスト上のSED対応ドライブ グループでのセキュリティの有効化	62
付録B: シリーズ6EコアまたはHybridホストの暗号化 (encryptSedVd.py)	67
構成済みのドライブ グループでのSEDの有効化	69
仮想ドライブ/ドライブ グループの有効化: PERC H740(Mini) アダプタ(内部ストレージ)	74
PowerVault(PERC 840) で構成された仮想ドライブ/ドライブ グループでのSEDの有効化	77
仮想ドライブ/ドライブ グループの有効化: PERC H840アダプタ	77
付録C: トラブルシューティング	87
REST APIを使用した、デコーダーに接続された事前構成済みDACの再構成	87
付録D: ストレージ構成のシナリオの例	88
Archiverのストレージの構成	88
Network(Packet) Decoderのストレージの構成	91
Network Concentratorのストレージの構成	101
Log Decoder Hybridのストレージの構成	107
改訂履歴	112

ストレージの概要

このガイドでは、ストレージ要件と、RSA NetWitness Platformの物理ストレージ デバイス(DAC、PowerVault、Unity) および仮想ストレージ デバイスにストレージを割り当てる方法を説明します。また、次のトピックも含まれています。

- 既存のPowerVaultでの暗号化の検出
- 別のデバイスへのデータ移行

これらのデバイスを、RSA NetWitness PlatformコアおよびHybrid物理ホストに接続する方法については、次の各ハードウェア セットアップ ガイドを参照してください。

- PowerVault (MD 1400) Setup Guide(「Hardware Description」の「Enclosure Options」セクションを参照) : RSA Link <https://community.rsa.com/docs/DOC-94091>
- 60-Drive DAC Setup Guide : RSA Link <https://community.rsa.com/docs/DOC-44956>
- 15-Drive DAC Setup Guide : RSA Link <https://community.rsa.com/docs/DOC-44957>

ストレージ要件

このセクションには、NetWitness Platform導入ホストシステムにストレージを正常に接続するために必要なすべてのストレージ要件が含まれています。これには、必要なドライブタイプ、適切なボリューム、必要なパフォーマンスIOPSが含まれます。

ドライブの仕様

コアNetWitness Platformホストの一般的な仕様は、次のとおりです。

- IOサイズ(490/Dec)
- レスポンスのレイテンシ20ms未満
- Decoder 10/90読み取り/書き込み(低ランダムI/O)
- Concentrator 50/50読み取り/書き込み(高ランダムI/O)

RAIDグループ	適切なボリューム
NL-SASまたは10K SAS	すべてのPacket Decoderボリューム すべてのLog Decoderボリューム すべてのArchiverボリューム Concentratorメタ ボリューム
SSD	Concentratorインデックス ボリューム

必要なNetWitness Platformストレージ ボリューム

サービス ボリューム名

サービス	ボリューム名	作成されるファイルシステム
Network Decoder	Decoder	packetdb
Network Decoder	Decodersmall	decoder root、index、sessiondb、metadb
Log Decoder	logdecoder	packetdb
Log Decoder	logdecodersmall	logdecoder root、index、sessiondb、metadb
Concentrator	Concentrator	concentrator root、metadb、sessiondb
Concentrator	index	index
Archiver	archiver	database

ボリュームのサイズ設定

以下のボリュームサイズは、「[REST APIを使用したストレージの構成](#)」で説明されているように、NetWitness Platformストレージ ツールを使用すると自動的に作成されます。

Volume	Filesystem	マウント ポイント	サイズ
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	decodersmallボリューム上の空き容量の100%
Decoder	packetdb	/var/netwitness/decoder/packetdb	decoderボリューム上の空き容量の100%
logdecodersmall	Decoroot	/var/netwitness/logdecoder	10 GB
logdecodersmall	Index	/var/netwitness/logdecoder/index	30 GB
logdecodersmall	Sessiondb	/var/netwitness/logdecoder/sessiondb	600 GB
logdecodersmall	metadb	/var/netwitness/logdecoder/metadb	logdecodersmallボリューム上の空き容量の100%
logdecoder	packetdb	/var/netwitness/logdecoder/packetdb	logdecoderボリューム上の空き容量の100%
Concentrator	Root	/var/netwitness/concentrator	30 GB
Concentrator	Sessiondb	/var/netwitness/concentrator/sessiondb	concentratorボリューム上の空き容量の10%
Concentrator	metadb	/var/netwitness/concentrator/metadb	concentratorボリューム上の空き容量の100%
Index	Index	/var/netwitness/concentrator/index	indexボリューム上の空き容量の100%
archiver	使用して	/var/netwitness/archiver/database	archiverボリュームの空き容量の100%

パフォーマンスに関する推奨事項

RSAは、Packet DecoderとLog Decoderに2つのLUNまたはブロック デバイスを割り当てることを推奨しています。1つはパケット データ用で、もう1つは他のすべてのデータベース用です。これにより、高帯域幅のパケット データベースを他のデータベースから分離して、I/O帯域幅が他のアクティビティと競合しないようにすることができます。

Concentratorには、優れたパフォーマンスを得るために、個別のSSDベースのインデックス ボリュームが必要です。このインデックス ボリュームは、NL-SASに保存できるConcentratorメタ データベース ボリュームとは別のRAIDグループに格納する必要があります。Archiverは、アプライアンスごとに1つの大容量NL-SASストレージ ボリュームを使用できます。

1秒あたりのI/O処理数

次の表に、DecoderホストとConcentratorホストのIOPS要件を示します。

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

パケット	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

NetWitness Platformホストによるデータの保存方法の一般的な説明

NetWitness Platformホストがデータを保存する方法については、「[付録A: NetWitness Platformホストによるデータの保存方法](#)」を参照してください。

物理ストレージの準備

重要: RSAでは、RSA NetWitnessストレージのブロック デバイスを作成することをお勧めします。

このセクションでは、ブロック デバイスを構成するための2つのオプションについて説明します。

- [ドライブ パックのブロック デバイスの構成](#)
- [PowerVaultのブロック デバイスの構成](#)

注: ブロック デバイスは、仮想ドライブまたはドライブ グループとも呼ばれます。

ドライブ パックのブロック デバイスの構成

シリーズ6または6Eアプライアンスにドライブを追加して、さまざまなユース ケースに対応することができます。これらのドライブは、decoderのメタ ボリュームまたはconcentratorのインデックス ボリュームをアプライアンスに常駐させる機能を提供します。2台以上かつ6台以下のドライブが可能です。ドライブの数は、必要なメタ キャッシュまたはインデックスの量によって異なります。



シリーズ6/6Eドライブ パックのメリット

- **PowerVaultストレージ容量を最大化:** 従来は、PowerVaultストレージにより、Decoderメタデータのボリュームが割り当てられていました。これにより、PowerVaultで使用可能なストレージが減少します。ドライブ パックでは、さらに20TBの使用可能なPowerVaultストレージが提供されるため、この問題が軽減されます。
- **メタのみのユース ケースのコストを削減:** メタデータのみの導入の場合、ドライブ パックは、RSAからのハードウェアの購入を望むお客様に適しています。これで、より費用効果の高いソリューションが提供されます。ドライブ パックはPowerVaultの代わりに使用できるためです。
- **既存の導入を有効にして圧縮オプションを利用。**
- **メタ キーおよび関連するインデックス作成を拡張する機能を提供。**

Decoderメタのユース ケース

- メタのみ
- PowerVaultストレージの最大化

2台以上の2.4TB 10K SAS SEDドライブを、decodersmallまたはlogdecodersmallボリュームのDecoderに追加できます。これらのボリュームを使用してDecoderでメタ キャッシュが格納されます。

Log DecoderとNetwork Decoderはどちらも、収集したRAWトラフィックからメタ データを解析します。次に、メタ データが、インデックス作成のためにConcentratorに集約されます。

ホストには、Concentrator集約のデータ収集中に抽出されたメタのキャッシュを格納するストレージが必要です。Decoder上のメタキャッシュは、一般にサイズが固定されていますが、Decoderとそれに対応するConcentratorの間の接続が失われないようにするための追加のキャッシュをサポートするように拡張できます。

通常、decodersmallまたはlogdecodersmallボリュームは、最初と2番目(10G構成のみ)のPowerVaultエンクロージャの最初の3台のドライブに格納されます。ドライブパックオプションを使用することで、代わりに、これら3つのドライブをpacketdbに使用できます(Power Vaultストレージが最大になります)。



メタのみのシナリオでは、decodersmallボリュームがドライブパックに保存されるため、PowerVaultが不要になります。

Concentratorインデックスのユースケース

- 追加のメタキー インデックス作成のサポート
- 既存の導入の圧縮を有効にする機能

2台以上の3.84TB SSD SEDドライブをConcentratorに追加してインデックスボリュームを増やすことができます。インデックスストレージのニーズは、NetWitness Platformの導入の保存要件に基づいて調整されます。追加のメタキーが有効になっており、そのインデックスが作成されている場合は、インデックスの保存に影響する可能性があります。

既存の導入では、圧縮を有効にする必要がある場合は、SSDインデックスドライブパックが必要です。packetdbとmetadbを圧縮する場合は、これらのデータベースの圧縮をサポートするために、さらにインデックスが必要です。

Decoder/Log Decoderのブロックデバイスの構成

ドライブパックブロックデバイスは、RAID 5、RAID 6、RAID 1のいずれかで構成することをお勧めします。

ドライブパックSEDドライブはスロット4~9に追加されます。仮想ドライブ構成では、コントローラーIDとエンクロージャID(EID)を特定する必要があります。たとえば、シリーズ6 R640アプライアンスでは、コントローラーIDとエンクロージャIDが0と64です。

値を特定するには、次の操作を実行します。

- PERC H740P MiniのコントローラーID(Ctl) を特定します。次の図では、コントローラーIDは0です。ドライブ数はPDの下に表示されます。

```

/opt/MegaRAID/perccli/perccli64 show
[root@116S6Core1 ~]# /opt/MegaRAID/perccli/perccli64 show
Status Code = 0
Status = Success
Description = None

Number of Controllers = 2
Host Name = 116S6Core1
Operating System = Linux 3.10.0-1160.21.1.el7.x86_64
StoreLib IT Version = 07.0400.0200.0400
StoreLib IR3 Version = 15.50-0

System Overview :
=====

-----
Ctl Model                Ports PDs DGs DNOpt VDs VNOpt BBU sPR DS EHS ASOs Hlth
-----
  0 PERCH740PMini         8  10   2   0   2     0 Opt On -  N    0 Opt
  1 PERCH840Adapter       8  12   0   0   0     0 Opt On -  N    0 Opt
-----

Ctl=Controller Index|DGs=Drive groups|VDs=Virtual drives|Fld=Failed
PDs=Physical drives|DNOpt=DG NotOptimal|VNOpt=VD NotOptimal|Opt=Optimal
Msg=Missing|Dgd=Degraded|NdAtn=Need Attention|Unkwn=Unknown
sPR=Scheduled Patrol Read|DS=DimmerSwitch|EHS=Emergency Hot Spare
Y=Yes|N=No|ASOs=Advanced Software Options|BBU=Battery backup unit
Hlth=Health|Safe=Safe-mode boot

```

- コントローラ"0"のエンクロージャID(EID)を特定します。この場合、EIDは64です。

```

/opt/MegaRAID/perccli/perccli64 /c0 /eall show
[root@116S6Core1 ~]# /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

```

3. コントローラPERC H740P Mini上のSED対応ディスクのロット番号(ロット4~9)を特定します。これらのドライブは、どのドライブグループ(DG)にも属していません。これらのドライブの[DG]列には、"- "状態が"UGood"、SED値が"Y"と表示されます。

```
[root@116S6Core1 ~]# /opt/MegaRAID/perccli/perccli64 /c0 /eall /sall show
Controller = 0
Status = Success
Description = Show Drive Information Succeeded.

Drive Information :
=====
```

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
64:4	4	UGood	-	2.182 TB	SAS	HDD	Y	N	512B	ST2400MM0149	U
64:5	5	UGood	-	2.182 TB	SAS	HDD	Y	N	512B	ST2400MM0149	U
64:6	6	UGood	-	2.182 TB	SAS	HDD	Y	N	512B	ST2400MM0149	U
64:7	7	UGood	-	2.182 TB	SAS	HDD	Y	N	512B	ST2400MM0149	U
64:8	8	UGood	-	2.182 TB	SAS	HDD	Y	N	512B	ST2400MM0149	U
64:9	9	UGood	-	2.182 TB	SAS	HDD	Y	N	512B	ST2400MM0149	U

```
-----
EID-Enclosure Device ID|Slt-Slot No.|DID-Device ID|DG-DriveGroup
DHS-Dedicated Hot Spare|UGood-Unconfigured Good|GHS-Global Hotspare
UBad-Unconfigured Bad|Onln-Online|Offln-Offline|Intf-Interface
Med-Media Type|SED-Self Encryptive Drive|PI-Protection Info
SeSz-Sector Size|Sp-Spun|U-Up|D-Down/PowerSave|T-Transition|F-Foreign
UGUnsp-Unsupported|UGShld-UnConfigured shielded|HSPShld-Hotspare shielded
CFShld-Configured shielded|Cpybck-CopyBack|CBShld-Copyback Shielded
```

4. ホスト上の既存のブロックデバイスを特定します。ブロックデバイス名は、[NAME]列で特定されます。下に表示されているブロックデバイス名は、sdaとsdbです。"lsblk"を使用してブロックデバイスを一覧表示します。

```
lsblk
[root@116S6Core1 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0  1.1T  0 disk
├─sda1                               8:1    0    1G  0 part /boot
├─sda2                               8:2    0  1.1T  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  3.2T  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  3.2T  0 lvm /var/netwitness
```

5. 以下のコマンドでロット4~9のディスクを使用して、PERC H740P上に仮想ドライブまたはドライブグループ(DG)を作成します。

注: [DG]列の下に表示されるコントローラには、2台の既存の仮想ドライブ(0と1)があります。これらのドライブはNetWitnessソフトウェアをホストしており、アプライアンスのイメージング中に作成されます。これらの仮想ドライブを削除したり、上書きしたりしないでください。perccli64の使用法の詳細については、Dell.comにある『Dell EMC PowerEdge RAID Controller CLI Reference Guide』を参照してください。

```
/opt/MegaRAID/perccli/perccli64 /c0 add vd type=raid6 drives=64:4-9 strip=128
```

```
[root@116S6Core1 ~]# /opt/MegaRAID/perccli/perccli64 /c0 add vd type=raid6 drives=64:4-9 strip=128
Controller = 0
Status = Success
Description = Add VD Succeeded
```

6. 新しい仮想ドライブは、[DG]/[VG]列の下に"2/2"として表示されます。

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

```
[root@116S6Core1 ~]# /opt/MegaRAID/perccli/perccli64 /c0 /vall show
Controller = 0
Status = Success
Description = None

Virtual Drives :
=====

-----
DG/VD TYPE   State Access Consist Cache Cac sCC      Size Name
-----
0/0  RAID1 Optl RW    Yes  RWBD -  OFF 1.090 TB
1/1  RAID1 Optl RW    Yes  RWBD -  OFF 2.182 TB
2/2  RAID6 Optl RW    No   RWBD -  OFF 8.730 TB
-----

Cac=CacheCade|Rec=Recovery|OfLn=OffLine|Pdgd=Partially Degraded|Dgrd=Degraded
Optl=Optimal|R0=Read Only|RW=Read Write|HD=Hidden|TRANS=TransportReady|B=Blocked|
Consist=Consistent|R=Read Ahead Always|NR=No Read Ahead|WB=WriteBack|
FWB=Force WriteBack|WT=WriteThrough|C=Cached IO|D=Direct IO|sCC=Scheduled
Check Consistency
```

7. ホスト上の新しいブロック デバイスを特定します。ブロック デバイス名は、[NAME]列で特定されます。新しいブロック デバイスは"sdc"です。このブロック デバイス名は、ストレージを構成するときに必要です。"lsblk"を使用してブロック デバイスを一覧表示します。

```
lsblk
```

```
[root@116S6Core1 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0  1.1T  0 disk
├─sda1                               8:1    0    1G  0 part /boot
├─sda2                               8:2    0  1.1T  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   3.2T  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   3.2T  0 lvm  /var/netwitness
sdc                                  8:32   0  8.7T  0 disk
[root@116S6Core1 ~]#
```

- この後は、「[ストレージの構成](#)」セクションで説明されている"Decoder/LogDecoderのストレージの構成"を行って構成を完了する必要があります。

Concentratorのブロック デバイスの構成

追加のメタキー インデックス作成をサポートし、既存の導入の圧縮を有効にするには、ドライブ パックの SSD SEDドライブ上のブロック デバイスを構成する必要があります。インデックス ボリュームのドライブ パックブロック デバイスは、「[Decoder/Log Decoderのブロック デバイスの構成](#)」と同様のステップを使用して作成できます。percli64ユーティリティを使用してブロック デバイスを構成します。ブロック デバイスを構成した後は、「[Concentratorのストレージの構成](#)」に従ってストレージ構成を完了します。

PowerVaultのブロック デバイスの構成

Decoder、Log Decoder、Concentrator、Archiverの物理、仮想、またはクラウドのNetWitnessホストには、ブロック ストレージを接続する必要があります。割り当てられたストレージが、すべてのストレージ要件を満たしていることを確認してください。具体的には、必要なストレージ ボリュームが作成されていることを確認します(詳細については、「[ストレージ要件](#)」の「必要なNetWitness Platformストレージ ボリューム」を参照してください)。さらに、以下の条件も満たす必要があります。

- 少なくとも2つのブロック デバイスがDecoder用に作成されていること(メタ ボリューム、セッション ボリューム、パケット ボリューム)。

注: 大きい方のブロック デバイスがパケット ボリュームを保持し、小さい方のブロック デバイスがメタ ボリュームとセッション ボリュームを保持していること。

- 少なくとも2つのブロック デバイスがConcentrator用に作成されていること(インデックス ボリュームとメタ ボリューム)。
- 想定取得レートの最小IOPSをブロック デバイスが必ず満たしていること。

Decoder/Log Decoderのブロック デバイスの構成

ブロック デバイスRAID構成の作成時のベストプラクティスは、大きい方のNL-SASドライブ用にRAID 6を、10k SASまたはSSDタイプのドライブ用にRAID 5または1を構成することです。

- "PERC H840Pアダプター"のコントローラーID(Ctl)を特定します。

```
/opt/MegaRAID/perccli/perccli64 show
```

次の図では、コントローラーIDは"1"で、"PERCH840PAdaptor"に対応しています。

```
[root@116S6Core1 ~]# /opt/MegaRAID/perccli/perccli64 show
Status Code = 0
Status = Success
Description = None

Number of Controllers = 2
Host Name = 116S6Core1
Operating System = Linux 3.10.0-1160.21.1.el7.x86_64
StoreLib IT Version = 07.0400.0200.0400
StoreLib IR3 Version = 15.50-0

System Overview :
=====

-----
Ctl Model          Ports PDs DGs DNOpt VDs VNOpt BBU sPR DS EHS ASOs Hlth
-----
 0 PERCH740PMini    8  10   2   0   2     0 Opt On -  N    0 Opt
 1 PERCH840Adapter  8  12   0   0   0     0 Opt On -  N    0 Opt
-----

Ctl=Controller Index|DGs=Drive groups|VDs=Virtual drives|Fld=Failed
PDs=Physical drives|DNOpt=DG NotOptimal|VNOpt=VD NotOptimal|Opt=Optimal
Msg=Missing|Dgd=Degraded|NdAtn=Need Attention|Unkwn=Unknown
sPR=Scheduled Patrol Read|DS=DimmerSwitch|EHS=Emergency Hot Spare
Y=Yes|N=No|ASOs=Advanced Software Options|BBU=Battery backup unit
Hlth=Health|Safe=Safe-mode boot
```

2. コントローラ"1"のエンクロージャID(EID)を特定します。この場合、EIDは"247"です。

```
/opt/MegaRAID/perccli/perccli64 /c1 /eall show
[root@116S6Core1 ~]# /opt/MegaRAID/perccli/perccli64 /c1 /eall show
Controller = 1
Status = Success
Description = None

Properties :
=====

-----
EID State Slots PD PS Fans TSs Alms SIM Port# ProdID VendorSpecific
-----
247 OK          12 12  2   4   7   0   2 01 x4 MD1400
-----

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

3. ホスト上の既存のブロックデバイスを特定します。ブロックデバイス名は、[NAME]列で特定されます。以下に表示されているブロックデバイス名は、sda、sdb、sdcです。"lsblk"を使用してブロックデ

バースを一覧表示します。

```
[root@116S6Core1 ~]# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   1.1T  0 disk
├─sda1                               8:1    0     1G  0 part /boot
└─sda2                               8:2    0   1.1T  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   3.2T  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   3.2T  0 lvm  /var/netwitness
sdc                                  8:32    0   8.7T  0 disk
[root@116S6Core1 ~]#
```

- 以下のコマンドで、スロット0~9のディスク(すべてのドライブなど)を使用して、仮想ドライブまたはドライブグループ(DG)をPERCH840PA adaptor上に作成します。

警告: すべてのデコーダには、メタ用のlogdecodersmallまたはdecodersmallボリュームが必要です。この例では、別のPowerVaultまたはドライブパックにメタボリュームがすでに存在していることを前提としています。このエンクロージャをメタボリュームが使用する場合は、最初の2台または3台のドライブをメタボリュームのブロックデバイスに割り当てる必要があります。packetdbボリュームの残りのドライブを使用して、もう一つのブロックデバイスを作成する必要があります。

```
/opt/MegaRAID/percccli/percccli64 /c1 add vd type=raid6 drives=247:0-11
strip=128 force
```

```
[root@116S6Core1 ~]# /opt/MegaRAID/percccli/percccli64 /c1 add vd type=raid6 drives=247:0-11 strip=128 force
Controller = 1
Status = Success
Description = Add VD Succeeded

[root@116S6Core1 ~]#
```

- 上記のステップで作成した仮想ドライブを表示するには、次の操作を実行します。

```
/opt/MegaRAID/percccli/percccli64 /c1 /vall show
[root@116S6Core1 ~]# /opt/MegaRAID/percccli/percccli64 /c1 /vall show
Controller = 1
Status = Success
Description = None

Virtual Drives :
=====

-----
DG/VD TYPE  State Access Consist Cache Cac sCC      Size Name
-----
0/0  RAID6 Optl RW      No      RWBD  -   OFF 106.918 TB
-----

Cac=CacheCade|Rec=Recovery|OfLn=OffLine|Pdgd=Partially Degraded|Dgrd=Degraded
Optl=Optimal|R0=Read Only|RW=Read Write|HD=Hidden|TRANS=TransportReady|B=Blocked|
Consist=Consistent|R=Read Ahead Always|NR=No Read Ahead|WB=WriteBack|
FWB=Force WriteBack|WT=WriteThrough|C=Cached IO|D=Direct IO|sCC=Scheduled
Check Consistency
```

6. ホスト上の新しいブロック デバイスを特定します。ブロック デバイス名は、[NAME]列で特定されます。上記の仮想ドライブに対応する新しいブロック デバイスはsddです。このブロック デバイス名は、ストレージを構成するときに必要です。"lsblk"を使用してブロック デバイスを一覧表示します。

lsblk

```
[root@116Decoder perccli]# 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   8.7T  0 disk
sdd                                  8:48   0  106.9T  0 disk
[root@116Decoder perccli]#
```

7. 「[ストレージの構成](#)」に従って、Decoder/ Log DecoderとConcentratorのストレージを構成して構成を完了する必要があります。

Concentratorのブロック デバイスの構成

PowerVaultでブロック デバイスを構成した後に、ConcentratorのストレージとしてPowerVaultを構成する必要があります。ブロック デバイスは、perccli64ユーティリティーを使用して、「[Decoder/Log Decoderのブロック デバイスの構成](#)」と同様のステップで構成することができます。SSDドライブをインデックスに、残りのドライブをメタDBまたはセッションDBに使用します。

ストレージの構成

Decoder/LogDecoderのストレージの構成

REST APIツールを使用して、上記のブロック デバイスまたは仮想ドライブを、Decoder/Log DecoderまたはConcentratorのストレージとして構成します。詳細については、「[REST APIを使用したストレージの構成](#)」トピックの「ストレージ構成タスク」のタスク3および4(DecoderまたはLog Decoderの場合)、およびタスク1~5(Concentratorの場合)を参照してください。

サービス	コントローラ	Volume	ブロック デバイス
Decoder/Log Decoder	PERC H740 Miniアダプター	Decodersmall	「 Decoder/Log Decoderのブロック デバイスの構成 」(Decoder/Log Decoder)のステップ7を参照してください。この例では、ブロック デバイスは"sdc"です。
Decoder/Log Decoder	PERC H840アダプター	Decoder	「 Decoder/Log Decoderのブロック デバイスの構成 」のステップ6を参照してください。この例では、ブロック デバイスは"sdd"です。

Concentratorのストレージの構成

REST APIツールを使用して、ドライブ パックやPowerVault上に作成されたブロック デバイスを構成します。SSD上に作成されたブロック デバイスはインデックス データベースに割り当てられ、HDD上に作成されたブロック デバイスはメタ/セッション データベースに割り当てられます。「[REST APIを使用したストレージの構成](#)」トピックにある、Concentrator用のストレージ構成タスク(タスク3および4)を参照してください。

SED対応可能ドライブのセキュリティの有効化

PERC H740 MiniおよびPERC H840アダプタのSED対応可能ドライブ グループでセキュリティを有効にするには、「[付録B: シリーズ6EコアまたはHybridホストの暗号化 \(encryptSedVd.py\)](#)」を参照してください。

仮想ストレージまたはクラウド ストレージの準備

このセクションでは、次のタイプのコンポーネント ホスト用に、仮想ストレージまたはクラウド ストレージをセットアップする方法を説明します。

- [Decoder、Log Decoder、Concentrator、Archiver](#)
- [NW Server、ESA Primary、ESA Secondary、Malware Analysis](#)
- [Log Collector](#)
- [Endpoint Log Hybrid](#)
- [追加のEndpoint Log Hybridパーティション](#)
- [UEBA](#)

Decoder、Log Decoder、Concentrator、Archiver

Decoder、Log Decoder、Concentrator、Archiverの仮想NetWitnessホストまたはクラウドNetWitnessホストには、ブロックストレージを接続する必要があります。割り当てられたストレージが、すべてのストレージ要件を満たしていることを確認してください。具体的には、必要なストレージ ボリュームが作成されている(「[ストレージ要件](#)」の「必要なNetWitness Platformストレージ ボリューム」を参照) ことと、以下を確認してください。

- 少なくとも2つのブロック デバイスがDecoder用に作成されていること(メタ ボリューム、セッション ボリューム、パケット ボリューム)。
- 少なくとも2つのブロック デバイスがConcentrator用に作成されていること(インデックス ボリュームとメタ ボリューム)。
- 予想される取得レートの最小IOPSをブロック デバイスが確実に満たすことができること。

ホスティング プラットフォームのネイティブ手順に従って、割り当てられたストレージをNetWitnessホストに接続します。

- VMware: vSphereコンソール(VMにディスクを追加)
- Hyper-V: マネージャー コンソール(VMにディスクを追加)
- Azure: 管理対象ディスクを仮想インスタンスに追加
- AWS: 仮想インスタンスにEBSストレージを追加
- Google Cloud Platform(GCP): 仮想 インスタンスにストレージを追加

ストレージが仮想ホストに接続されたら、「[REST APIを使用したストレージの構成](#)」の「タスク3: パーティション、ボリュームグループ、論理ボリュームにブロック デバイスを割り当てる」に進みます。

NW Server、ESA Primary、ESA Secondary、Malware Analysis

/var/netwitness/パーティションを拡張するために外部ボリュームを接続します。

lsblkを実行して物理ボリューム名を取得します。

2 TBのディスクを接続する場合は、次のコマンドを実行します。

1. `pvcreate <pv_name>(pv_nameは、/dev/sdcなど)`
2. `vgextend netwitness_vg00 /dev/sdc`
3. `lvextend -L 1.9T /dev/netwitness_vg00/nwhome`
4. `xfs_growfs /dev/netwitness_vg00/nwhome`

RSAでは、次のパーティション定義を推奨しています。ただし、これらの値は、保存日数に基づいて変更できます。

LVM	フォルダ	ブロックストレージ
/dev/netwitness_vg00/nwhome	/var/netwitness/	クラウド プロバイダのブロックストレージのセットアップ(ストレージ)の表を参照してください。

Log Collector

/var/netwitness/パーティションを拡張するために外部ボリュームを接続します。

`lsblk`を実行して物理ボリューム名を取得します。

500 GBボリュームを1つ接続する場合は、次のコマンドを実行します。

1. `pvcreate <pv_name>(pv_nameは、dev/sdcなど)`
2. `vgextend netwitness_vg00 /dev/sdc`
3. `lvextend -L 600G /dev/netwitness_vg00/nwhome`
4. `xfs_growfs /dev/netwitness_vg00/nwhome`

RSAでは、次のパーティション定義を推奨しています。ただし、これらの値は、保存日数に基づいて変更できます。

LVM	フォルダ	ブロックストレージ
/dev/netwitness_vg00/nwhome	/var/netwitness/	クラウド プロバイダのブロックストレージのセットアップ(ストレージ)の表を参照してください。

Endpoint Log Hybrid

必要な合計ディスクサイズは、データ保持期間によって異なります。以下の1日あたりのディスク使用量の指標値を使用して、導入したストレージに必要なディスクサイズを計算できます。たとえば、30日分のデータを保持するには、以下の1日あたりのディスク使用量の値を30で乗算します。

次の表は、1回のフルスキャンのディスク使用量を示しています。フルスキャンのディスク使用量の値は、以下のイベント数に基づきます。

- ファイル数 : 1,100
- プロセス数 : 100
- DLL数 : 500
- ドライバー数 : 150
- サービス数 : 500
- タスク数 : 100

Endpoint Log Hybrid(50K詳細エージェント : フルスキャンあたりのディスク使用量)

	MetaDB	PacketDB	SessionDB	Index	合計
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(最初のフルスキャン) 30 GB(後続のスキャンごとの増加分)

次の表は、データをトラッキングするための1日あたりのディスク使用量を示しています。エージェントごとの1日あたりの合計トラッキングイベント数は29,000です。

Endpoint Log Hybrid(50K詳細エージェント : 拡張ネットワーク可視化なしのトラッキングデータ)

	MetaDB	PacketDB	SessionDB	Index	合計
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(最初のフルスキャン) 1.5 GB(追跡データの1日あたりの増加分)

次の表は、データをトラッキングするための1日あたりのディスク使用量を示しています。エージェントごとの1日あたりの合計トラッキングイベント数は33,000です。

Endpoint Log Hybrid(50K詳細エージェント : 拡張ネットワーク可視化ありのトラッキングデータ)

	MetaDB	PacketDB	SessionDB	Index	合計
Log Decoder	1800 GB	152 GB	55 GB	NA	2007 GB
Concentrator	1900 GB	NA	55 GB	36 GB	1991 GB

Endpoint Log Hybrid(50K詳細エージェント : 拡張ネットワーク可視化ありのトラッキングデータ)

MongoDB	NA	NA	NA	NA	35 GB(最初のフル スキャン) 1.5 GB(追跡データの1日あたりの増加分)
---------	----	----	----	----	--

次の表は、Insightsエージェントの1日あたりのディスク使用量を示しています。エージェントごとの1日あたりの合計トラッキング データ数は、10,800に、毎日1回のフル スキャン分を加えた数です。

Endpoint Log Hybrid(拡張ネットワーク可視化ありの50K Insightsエージェント)

	MetaDB	PacketDB	SessionDB	Index	合計
Log Decoder	500 GB	52 GB	18 GB	NA	570 GB
Concentrator	600 GB	NA	18 GB	13 GB	631 GB
MongoDB	NA	NA	NA	NA	35 GB(最初のフル スキャン) 30 GB(後続のスキャンごとの増加分)

次の表に、機能に基づいたEndpointエージェントのサイズ設定を示します。

機能	説明	エージェントまたはEndpoint Server
Endpointのみ	データのスキャンとトラッキングのみ	最大50KのEndpointエージェントのみ
Windowsログのみ	エージェントからのWindowsログのみ。1秒あたり20KのイベントがHybridでサポートされると想定しています。	最大20Kのエージェント： • 1秒あたり20Kのログ イベントを生成します。
ファイル収集のみ	エージェントからのファイル収集のみ。1秒あたり20KのイベントがHybridでサポートされると想定しています。	最大20Kのエージェント： • 1秒あたり20Kのログ イベントを生成します。

機能	説明	エージェントまたはEndpoint Server
EndpointとWindowsログ	<p>エージェントごとの1秒あたりのイベント</p> <ul style="list-style-type: none"> • (Windowsログの場合) 1つのエージェントから毎秒1イベントが送信されます • (イベントのトラッキングの場合) 1つのエージェントから毎秒0.4イベントが送信されます • 1秒あたり20KのイベントがHybridでサポートされます <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p>注: 総エージェント数は、次のように計算されます。 1秒あたりのHybridイベント/ (1エージェントのWindowsログ Endpoint Server+ 1エージェントのトラッキング イベント Endpoint Server) 例: 20,000 / (1.0 + 0.4)</p> </div>	<p>最大(約) 15Kのエージェント:</p> <ul style="list-style-type: none"> • (約) 15KのWindowsログ イベントを生成します <p>および</p> <ul style="list-style-type: none"> • (約) 15KのエージェントEDRデータを生成します
Endpoint、Windowsログ、ファイル収集	<p>エージェントごとの1秒あたりのイベント:</p> <ul style="list-style-type: none"> • (Windowsログの場合) 1つのエージェントから毎秒1イベントが送信されます • (イベントのトラッキングの場合) 1つのエージェントから毎秒0.4イベントが送信されます • (ファイル収集の場合) 1つのエージェントから毎秒1つのイベントが送信されます • 1秒あたり20,000のイベントがHybridでサポートされます <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p>注: 総エージェント数は、次のように計算されます。 1秒あたりのHybridイベント/ (1エージェントのWindowsログ Endpoint Server+ 1エージェントのトラッキング イベント Endpoint Server + ファイル収集) 例: 20,000 / (1.0 + 1.0 + 0.4)</p> </div>	<p>最大(約) 10Kのエージェント:</p> <ul style="list-style-type: none"> • (約) 10KのWindowsログ イベントを生成します <p>および</p> <ul style="list-style-type: none"> • (約) 10KのEndpointエージェント データを生成します <p>および</p> <ul style="list-style-type: none"> • (約) 10Kのエージェント ファイル収集データを生成します

ファイルシステムの拡張

Endpoint Serverの場合は、`/var/netwitness/`パーティションを拡張するために外部ディスクを接続します。`nwhome`というサフィックスの付いた外部ディスクを作成します。

次のステップを実行します。

1. 新しいディスクを追加したことを確認してください。詳細については、『RSA NetWitness Platform仮想ホスト インストールガイド』の「タスク1:新しいディスクの追加」を参照してください。RSA NetWitness Platform 11.xのすべてのドキュメントの一覧は、「[マスター目次](#)」で確認できます。
2. たとえば、6 TBのディスクを1つ接続する場合は、`lsblk`を実行し、物理ボリューム名を取得します。
3. `pvcreate <pv_name>`(`pv_name`は`/dev/sdc`など)
4. `vgextend netwitness_vg00 /dev/sdc`
5. `lvextend -L 5.9T /dev/netwitness_vg00/nwhome`
6. `xfs_growfs /dev/mapper/netwitness_vg00-nwhome`

RSAでは、Endpoint Server用のパーティションを推奨しています(保存日数に基づいて変更できます)。

LVM	Folder	Size	Disk Type
<code>/dev/netwitness_vg00/nwhome</code>	<code>/var/netwitness/</code>	6TB	HDD

Mongo DBの場合は、`/var/netwitness/mongo`パーティションを拡張するために外部ディスクを接続します。`nwhome`というサフィックスの付いた外部ディスクを作成します。

次のステップを実行します。

1. 新しいディスクを追加したことを確認してください。詳細については、『RSA NetWitness Platform仮想ホスト インストールガイド』の「タスク1:新しいディスクの追加」を参照してください。RSA NetWitness Platform 11.xのすべてのドキュメントの一覧は、「[マスター目次](#)」で確認できます。
2. たとえば、6 TBのディスクを1つ接続する場合は、`lsblk`を実行し、物理ボリューム名を取得します。
3. `pvcreate <pv_name>`(`pv_name`は`/dev/sdc1`など)
4. `vgextend hybrid /dev/sdc1`
5. `lvextend -L 5.9T /dev/hybrid-vmng`
6. `xfs_growfs /dev/mapper/hybrid-vmng`

RSAでは、Mongo DB用のパーティションを推奨しています(保存日数に基づいて変更できます)。`var/netwitness`に推奨される最小サイズは500 GBです。

LVM	Folder	Size	Disk Type
<code>/dev/hybrid-vmng</code>	<code>/var/netwitness/mongo</code>	6TB	HDD

追加のEndpoint Log Hybridパーティション

次のパーティションは、ボリューム グループ エンドポイント上にあり、単一のRAID 0アレイ内にある必要があります。

フォルダ	LVM	ボリュームグループ
/var/netwitness/mongo	hybrid-mongo	Endpoint
/var/netwitness/concentrator	concentrator-concroot	Endpoint
/var/netwitness/concentrator/index	hybrid-concindex	Endpoint
/var/netwitness/logdecoder	hybrid-ldecroot	Endpoint

lsblkを実行して物理ボリューム名を取得し、次のコマンドを実行します。

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. 上記のすべてのLVMについて、上記のステップを繰り返します。

RSAでは、次のパーティションを推奨しています。ただし、これらの値は、保存日数に基づいて変更できません。

LVM	フォルダ	ブロックストレージ
/dev/netwitness_vg00/nwhome	/var/netwitness/	クラウド プロバイダのブロックストレージのセットアップ(ストレージ)の表を参照してください。
/dev/endpoint/hybridmongo	/var/netwitness/mongo	クラウド プロバイダのブロックストレージのセットアップ(ストレージ)の表を参照してください。
/dev/endpoint/concentratorconcroot	/var/netwitness/concentrator	クラウド プロバイダのブロックストレージのセットアップ(ストレージ)の表を参照してください。
/dev/endpoint/hybridconcindex	/var/netwitness/concentrator/index	クラウド プロバイダのブロックストレージのセットアップ(ストレージ)の表を参照してください。
/dev/endpoint/hybridldecroot	/var/netwitness/logdecoder	クラウド プロバイダのブロックストレージのセットアップ(ストレージ)の表を参照してください。

UEBA

次の手順では、外部ディスクを接続し、`/var/netwitness/`パーティションを拡張します。永続的なディスク サフィックスとして`nwhome`を使用する必要があります。この手順は、2 TBディスクを追加する方法を示しています。

注: `/var/netwitness`は、このボリュームに常駐できる唯一のパーティションです。

1. 物理ボリューム名を一覧表示します。

```
lsblk( dev/mapper/sdcなど)
```

2. `/var/netwitness/`パーティションを拡張します。

```
pvcreate <pv_name>( pv_nameはdev/mapper/sdc)
vgextend netwitness_vg00 /dev/mapper/sdc
lvextend -L 1.9T /dev/mapper/netwitness_vg00/nwhome
xfs_growfs /dev/mapper/netwitness_vg00-nwhome
```

このパーティションは、UEBAのRSA推奨パーティションです。これは、保存日数に基づいて変更できません。

REST APIを使用したストレージの構成

NetWitness Platform 11.3以降のリリースでは、すべてのストレージ構成操作にREST APIを使用します。REST APIの使用方法については、「RESTful APIユーザガイド」を参照してください。RSA NetWitness Platform 11.xのすべてのドキュメントの一覧は、「[マスター目次](#)」で確認できます。

REST APIストレージ構成コマンド

以下のリストにある各コマンドには、その機能と使用方法を説明するヘルプが組み込まれています。RESTインターフェイスを使用している場合は、ドロップダウンメニューでコマンドを選択すると、ヘルプテキストが表示されます。REST APIストレージ構成コマンドの例については、「[付録D: ストレージ構成のシナリオの例](#)」を参照してください。

直接接続されたRAIDボリューム用のコマンド

- `raidList`: このホストに存在するRAIDコントローラーと直接接続エンクロージャを一覧表示します。
- `raidNew`: 直接接続されたエンクロージャをブロック デバイスに割り当てます。

ブロック デバイスをストレージとして割り当てるためのコマンド

- `devlist`: ホスト上の使用可能なブロック デバイスを一覧表示します。
- `partNew`: ブロック デバイス上でパーティションを割り当て、ボリューム グループを作成します。
- `vgs`: ボリューム グループとしてのブロック デバイスの編成方法の要約を表示します。

ストレージをサービスに割り当てるためのコマンド

- `srvList`: ホスト上のサービスと、それらに割り当てられたストレージ パスを一覧表示します。
- `srvAlloc`: ボリューム グループをサービスに割り当てます。
- `srvFree`: ボリューム グループをサービスから削除します。

すべての新しいストレージを検出して使用するためにサービスを再構成するコマンド

- `reconfig`: 新しいストレージの構成後に、関連づけられたサービスおよびデータベースで、新しいストレージを検出し、使用します。

ストレージ構成タスク

タスク1: ストレージをホストに接続し、REST APIストレージ構成コマンドにアクセスする。

タスク2: (条件付き) 必要に応じてRAIDを構成する。

タスク3: パーティション、ボリュームグループ、論理ボリュームにブロック デバイスを割り当てる。

タスク4: ボリューム グループをNetWitnessサービスに割り当てる。

タスク5: 新しいストレージを検出し、適切に使用するようにサービスとデータベースを再構成する。

タスク1: ストレージをホストに接続し、REST APIストレージ構成コマンドにアクセスする

重要: タスク1は、NetWitness Platformバージョン11.5.0.0および11.5.0.1には適用されません。

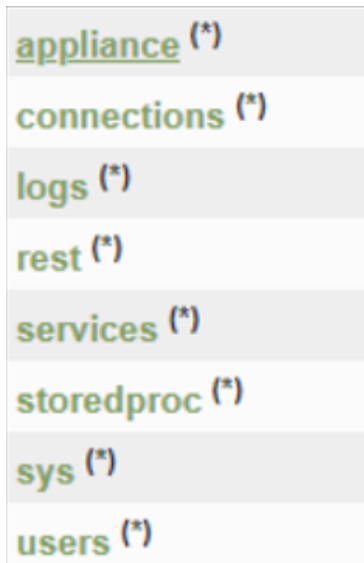
次のステップを実行して、外部ストレージ デバイスをホストに接続し、REST APIを介して使用可能なストレージ構成コマンドにアクセスします。

1. ストレージを接続し、このホストで使用できるようにします。
 - PVストレージを接続するには、「[PowerVault\(Dell MD 1400\) セットアップ ガイド](#)」を参照してください。
 - サードパーティーのストレージの場合は、「[ストレージ要件](#)」内のリストにあるボリュームと一致するようにRAIDグループを作成します。
2. REST APIストレージ コマンドにアクセスする方法は2つあります。ブラウザ、またはユーザ インターフェイスの[サービス] > [エクスプローラー]ビューで。

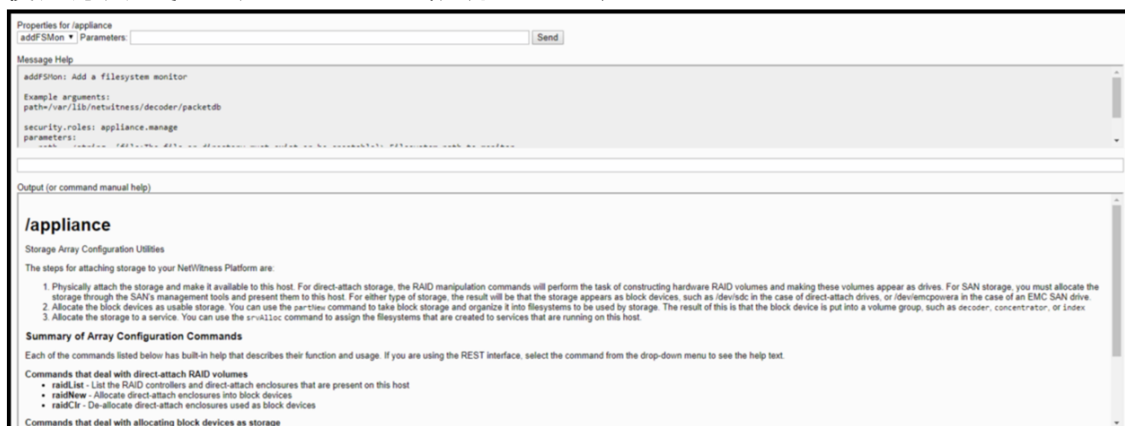
注: REST APIにアクセスした後は、アクセスに使用した方法に関係なく、実行するステップは同じです。



- ブラウザでアクセスする方法は、次のとおりです。
 - a. ブラウザを開き、ポート50106を持つホストのIPアドレスを指定します。
 次の例はDecoderですが、REST APIを使用してストレージを構成するあらゆるホストハードウェアでは、ポート50106を使用する必要があります。
 https://<decoder-ip-address>:50106

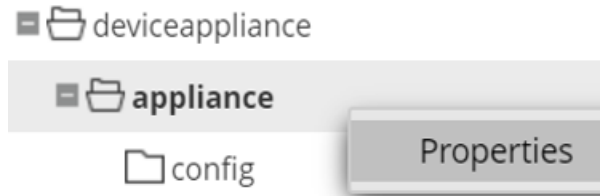
- b. adminアカウントの認証情報を使用してログインします。
 以下のREST APIメニューが表示されます。



- c. アプライアンスの隣にある(*)をクリックしてRESTコマンドセットにアクセスします。
 初期RESTメニューの下に[/applianceのプロパティ]ダイアログが表示されます。[出力(またはコマンド マニュアル ヘルプ)]セクションでは、REST APIがデバイスに送信できるコマンド、その使用方法、そのパラメータについて説明しています。



- ユーザ インターフェイスでアクセスする方法は、次のとおりです。
 - a. NetWitness Platformメニューで  (管理) > [サービス]に移動します。
 - b. サービス(Concentratorなど)を選択します。
 - c.  (アクション)で[表示]>[エクスプローラ]を選択します。
 - d. deviceappliance/applianceに移動し、[プロパティ]を右クリックしてからクリックします。



注: NetWitness Platformバージョン11.5.0.0または11.5.0.1を使用している場合は、[システム]>[ホスト タスク]>[タスク]に移動する必要があります。

これで、[プロパティ]ダイアログから各ストレージ コマンドにアクセスできます。

3. 以下に進みます。
 - **タスク2:** PowerVaultまたはDAC用にRAIDを構成する必要がある場合。
 - **タスク3:** RAIDを構成する必要がなく、すでにブロック デバイスを使用できる場合。

タスク2:(条件付き) PowerVaultおよびDAC用のRAID構成

重要: NetWitness Platformバージョン11.5.0.0または11.5.0.1を使用している場合、タスク2は必須です。

NetWitness Platformハードウェアは、直接接続されたSASドライブをストレージに使用します。これらのドライブはSASエンクロージャに収容されています。SASエンクロージャは、SASホスト バス アダプタに接続されたケーブルによってNetWitnessノードに提供されたドライブのシェルフです。

SASエンクロージャは、"DAC"(直接接続容量)、“JBOD”(Jumbo Box of Disks)、“Dell PowerVault”などの別名でも知られています。

NetWitness Platformは、Dell PERC SASホスト バス アダプタを使用します。NetWitness Platformデバイスには、通常、2つのSASホスト バス アダプタが含まれています。1つは、NetWitnessノードの内部にあるコントローラードライブとして使用され、もう1つは、SASエンクロージャに接続されたドライブを制御するために使用されます。内部コントローラおよびドライブは、ノードの構築時に構成されますが、外部SASエンクロージャは構成されません。raidListおよびraidNewコマンドを実行して、外部SASエンクロージャを特定し、構成します。

これらのコマンドは、次のSASエンクロージャタイプで機能します。

- EMC ESAS 15ドライブ エンクロージャ
- EMC ESAS 60ドライブ エンクロージャ
- Dell PowerVault 12ドライブ エンクロージャ

注: EMC 60ドライブ エンクロージャは、4つの個別の15ドライブ サブエンクロージャとして論理的に編成されています。これらは、15ドライブ エンクロージャが4つ存在するように動作し、それぞれを個別に構成できます。

raidListおよびraidNewコマンドはエンクロージャ全体で動作します。raidListを実行してエンクロージャを特定します。raidNewを実行して、NetWitness Platformノード内で事前に決定された役割の1つを実行するようにエンクロージャを構成します。

ストレージをホストに接続し、REST APIストレージ コマンドにアクセスした後に、必要に応じて、次のステップを実行してRAIDを作成します。

1. raidListコマンドを実行して、システムに接続されているコントローラーとエンクロージャを特定します。
次の例では、コントローラー1にブロック デバイスが表示されません。これは、アレイが構成されていないことを示しています。

```

Properties for /appliance
raidList Parameters: Send

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

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

Output (or command manual help)
Controller 0, Enclosure 32
Vendor: DP
Model: BP13G+EXP
In Use: true
Drives: 931.511 GB x 2
        1.818 TB x 2
Devices: sda
        sdb

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

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

2. エンクロージャのRAIDレイアウト スキームを選択します。
次の表に、サポートされている割り当てスキームを示します。

注: RAID構成の場合、Decoderが10G収集用に構成されているときは、最初の2つのエンクロージャにはdecoderスキームを、それに続くエンクロージャにはarchiverスキームを使用します。10G収集用に構成していないときは、最初のエンクロージャにdecoderスキームを、それに続くエンクロージャにはarchiverスキームを使用します。これらの構成により、ストレージ容量とパフォーマンスが最大になります。

スキーム	必要なドライブ	アロケーション
decoderまたはlogdecoder	12台または15台のHDD	RAID 5の3台のドライブはdecodersmallまたはlogdecodersmall、残りは全てRAID 5
Archiver	12台または15台のHDD	RAID 6のすべてのドライブがarchiverまたはdecoderデータベースボリューム
Networkhybrid	12台または15台のHDD	RAID 5の3台のドライブはメタ拡張、RAID 5の残りのすべてのドライブはパケット拡張
Loghybrid	12台または15台のHDD	RAID 5のドライブの半数はメタ拡張、RAID 5のドライブの半数はパケット拡張
Concentrator	3台以上のSSD、3台以上のHDD	RAID 5のすべてのSSDをインデックス、RAID 6のすべてのHDDをメタ

3. コントローラ、エンクロージャ、スキームを特定したら、raidNewコマンドを実行してRAIDボリュームを作成します。以下に例を示します。

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

commit=1パラメータを追加して、この操作を実際に行います。raidListコマンドを実行して、作成したブロック デバイスを一覧表示します。

4. (オプション) SED(自動暗号化ドライブ)を構成します。raidNewコマンドが自動暗号化ドライブを検出し、セキュリティ キーがコントローラに設定されていた場合、raidNewコマンドは、セキュアなアレイを作成しようとします。コントローラにセキュリティ キーを設定するには、raidKeyコマンドを実行します。以下に例を示します。

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

- セキュリティ キーが設定されたコントローラに接続された物理デバイス上に、セキュアな(つまり、暗号化された)アレイを作成するには、raidNewの使用時にpreferSecure=trueを指定します。
 - セキュリティ キーが設定されたコントローラに接続された物理デバイス上に、セキュアでない(つまり、暗号化されていない)アレイを作成するには、raidNewの使用時にpreferSecure=falseを指定します。
5. RAIDボリュームを作成したら、「[タスク3: パーティション、ボリュームグループ、論理ボリュームにブロック デバイスを割り当てる](#)」に移動します。

タスク3: パーティション、ボリュームグループ、論理ボリュームにブロック デバイスを割り当てる

partNewコマンドで、NetWitness Platformで使用するストレージ デバイスを準備します。次のタスクが実行されます。

- ブロック デバイス上にパーティション テーブルを作成する。
- Linuxボリューム マネージャーの物理デバイス パーティションを作成する。

- 物理デバイスを含むボリュームグループを作成する。
- ボリュームグループに論理ボリュームを作成する。
- 各論理ボリュームにXFSファイルシステムを作成する。
- 各論理ボリュームの/etc/fstabエントリを作成する。
- 各論理ボリュームをマウントします。

パーティション、ボリュームグループ、論理ボリュームにブロック デバイスを割り当てるには、次のステップを実行します。

1. `devlist` コマンドを実行して、未使用のブロック デバイスを見つけます。次の例は、`devlist` コマンドの出力を示します。

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
```

ストレージで使用されるサービス名を指定する必要があります。たとえば、Network Decoder サービスの場合は `decoder`、Concentrator サービスの場合は `concentrator` を指定します。ボリュームタイプを指定することもできます。デフォルトのボリュームタイプ名は、サービス名と同じです。

2. `partNew` コマンドを実行して、パーティション、ボリュームグループ、論理ボリュームにブロック デバイスを割り当てます。

デフォルトでは、`partNew` コマンドによっては変更されません。コマンド文字列をコミットした場合に実行されるアクションが表示されます。実際にシステムに変更を加えるには、`commit=true` パラメーターをコマンドに追加します。

たとえば、`sdd` デバイスと `sde` デバイスを Decoder に割り当てるには、次のコマンドを実行します。

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

注意: `decoder` および `concentrator` サービスの場合は、特定の順序でストレージ ボリュームを作成する必要があります。

- `decoder` には、`decodersmall` ボリュームと `decoder` ボリュームがあります。`decodersmall` ボリュームを作成した後に `decoder` ボリュームを作成します。これは、`decodersmall` には、`/var/netwitness/decoder` にマウントされた小規模のファイルシステムが含まれるためです。

- `concentrator` には、`concentrator` ボリュームと `index` ボリュームがあります。`concentrator` ボリュームを作成した後に `index` ボリュームを作成します。そうしないと、エラーが発生し、次のメッセージが表示されます。

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

3. `vgs` コマンドを実行して、正しい論理ボリュームが `partNew` コマンドによって作成されたことを検証します。

このコマンドの出力は、次のようになります。

- このホスト上のすべてのボリュームグループが列挙されます。
 - ボリュームグループを構成する物理ボリュームと、ボリュームグループ内の論理ボリュームが表示されます。
4. 「[タスク4: ボリュームグループをNetWitnessサービスに割り当てる: srvAlloc](#)」に進みます。

タスク4: ボリュームグループをNetWitnessサービスに割り当てる: srvAlloc

srvAllocコマンドで、ボリュームグループ内のストレージを使用するように、ホスト上のサービスを構成します。構成するサービスの名前と、サービスに割り当てるボリュームグループを指定する必要があります (提供するサービスはホストにインストールする必要があります)。NetWitness Platformサービスボリュームの詳細については、「[ストレージ要件](#)」の「NetWitness Platformサービスボリューム リファレンス」を参照してください。

次の順序でサービスを割り当てます。

- Decoderの場合は、最初にdecodersmall、次にdecoder
- Concentratorの場合は、最初にconcentrator、次にindex



注: デフォルトでは、srvAllocコマンドによっては変更されません。commit=true/パラメータをコマンド文字列に追加してシステムに実際に変更を加え、変更後に、指定したサービスを再起動する必要があります。

1. srvLstコマンドを実行して、このホストにインストールされているサービスのリストを表示します。srvLstコマンドは、SSLポートを介してサービスと通信します。ホストにカテゴリをインストールします。カテゴリは、単一のサービス、または同じホスト上の複数の関連サービスにすることができます。
2. srvAllocコマンドを実行して、ボリュームグループ内のストレージを使用するように、ホスト上のサービスを構成します。以下に例を示します。

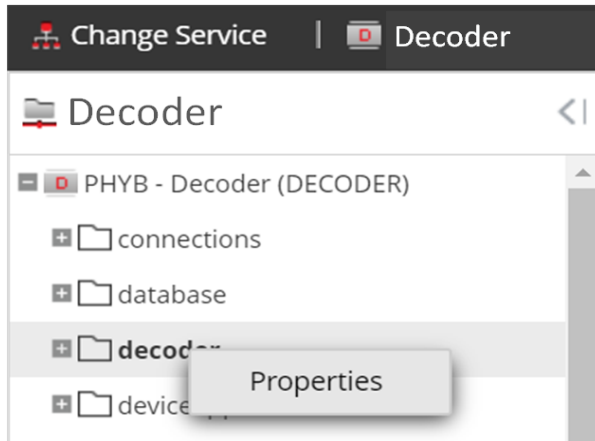

```
service=concentrator volume=concentrator commit=1
service=concentrator volume=index commit=1
```
3. 「[タスク5: 新しいストレージを検出し、適切に使用するようにサービスとデータベースを再構成する](#)」に進みます。

タスク5: (オプション) ストレージ構成を10G収集用に再構成する

Decoderサービスおよびデータベースを10G収集用に再構成する必要があります。次の手順を実行して、Network Decoderサービスとそのデータベースが、新しい空き領域を検出して使用するようになります。

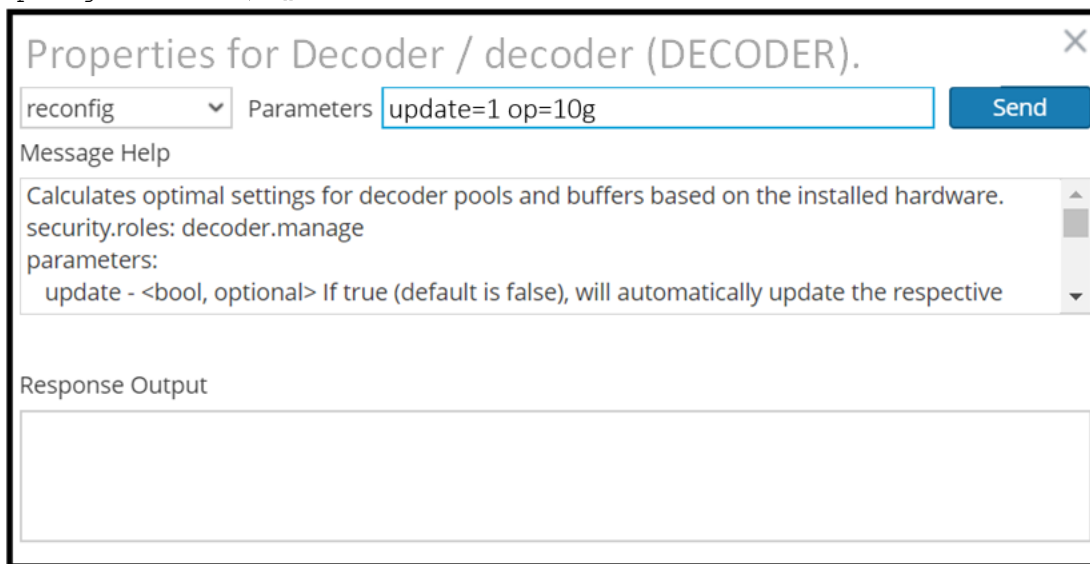
1. NetWitness Platformメニューで  (管理) > [サービス] に移動します。
[サービス]ビューが表示されます。
2. decoderを選択します。
3.  (アクション) の下の[表示] > [エクスプローラー]を選択します。
サービスの[エクスプローラー]ツリーが表示されます。

4. decoderサービス上のスペースを再構成します。
 - a. decoderに移動し、[プロパティ]を右クリックしてからクリックします。

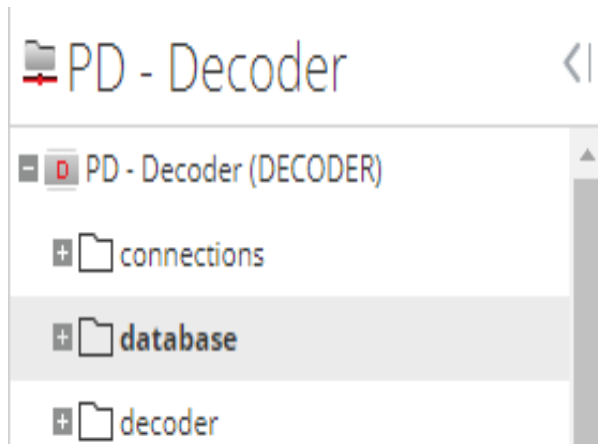


[プロパティ]ダイアログが表示されます。

- b. ドロップダウン リストからを選択してreconfigコマンドを実行します。[パラメータ]でupdate=1 op=10gを指定し、[送信]をクリックします。

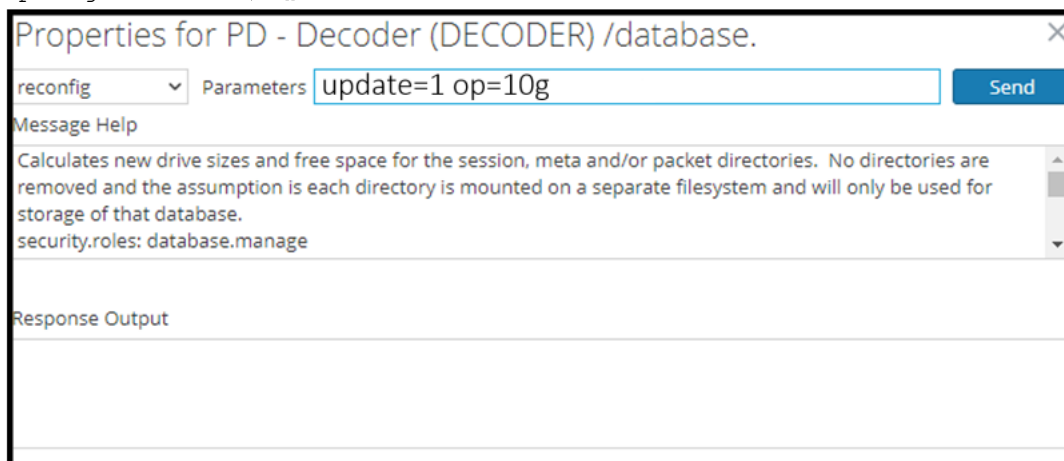


5. データベース上のスペースを再構成します。
 - a. サービスの[エクスプローラ]ツリーのdatabaseに移動し、[プロパティ]を右クリックしてからクリックします。



[プロパティ]ダイアログが表示されます。

- b. ドロップダウン リストからを選択してreconfigコマンドを実行します。[パラメータ]でupdate=1 op=10gを指定し、[送信]をクリックします。



Unityストレージの準備

Dell EMCストレージ エンジニアと協力して、RSA NetWitness PlatformのストレージをUnity環境内で割り当て、割り当てられたストレージが必ず、RSA NetWitness Platformストレージ要件をすべて満たすようにする必要があります。具体的には、以下のことを確認します。

- 少なくとも2つのLUNがDecoder用に作成されている(メタ/セッションおよびパケット ボリューム)。
- 少なくとも2つのLUNがConcentrator用に作成されている(インデックス ボリュームとメタ ボリューム)。
- 予想される取得レートの最小IOPSをブロック デバイスが満たすことができることを確認する。

Unityストレージを使用するすべてのRSA NetWitnessホストを、Unityインターフェイス内のホストとして追加する必要があります。ホストとLUNを作成したら、LUNをホストに割り当てる必要があります。LUNをホストに割り当てると、ストレージがホストに表示されるようになるため、ホストは、ホストベースのDell EMC PowerPathソフトウェアを介してストレージを見つけることができます。

注: Dell EMCエンジニアは、次のUnityアレイを構成します。

Unityストレージを準備するには、次のタスクを実行する必要があります。

タスク1 - Unisphereユーザー インターフェイス(UI)にアクセスする

タスク2 - プールを作成する

タスク3 - LUNを作成する

タスク4 - ホストを登録する

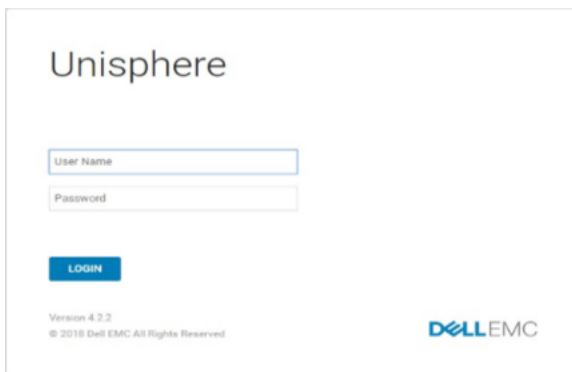
タスク5 - LUNをホストに割り当てる

タスク6 - PowerPathをインストールする

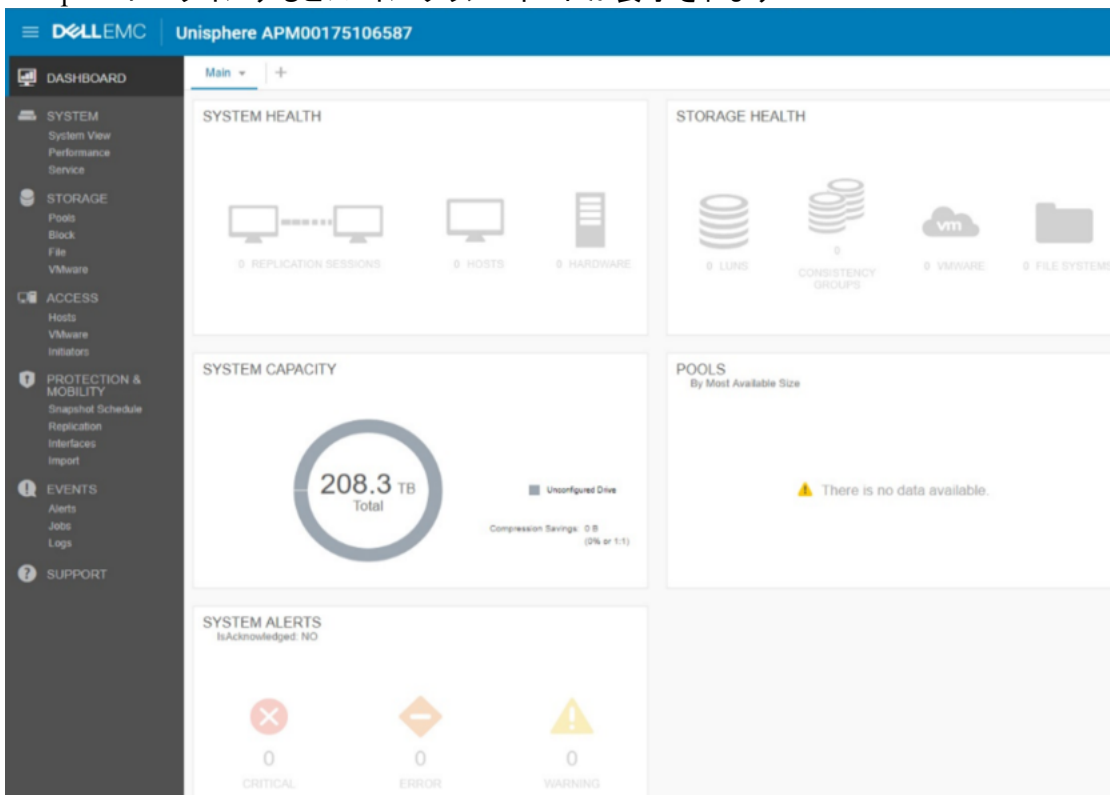
タスク1 - Unisphereユーザー インターフェイス(UI) にアクセスする

1. UNITYと同じサブネットにワークステーションを接続します。
2. ブラウザーを開き、<http://<unisphereIP>>にアクセスしてUnisphere UIに接続します。
3. Dell EMC CEから提供された認証情報でログインします。デフォルトの認証情報は **admin/Password123!#**です。

注： Unisphereにより、最初のログイン時にパスワードの変更が求められます。また、アレイを構成する前にライセンスをインストールすることも求められます(これは、Dell EMC CEが行う場合があります)。その場合は、新しい管理者パスワードを取得する必要があります)。



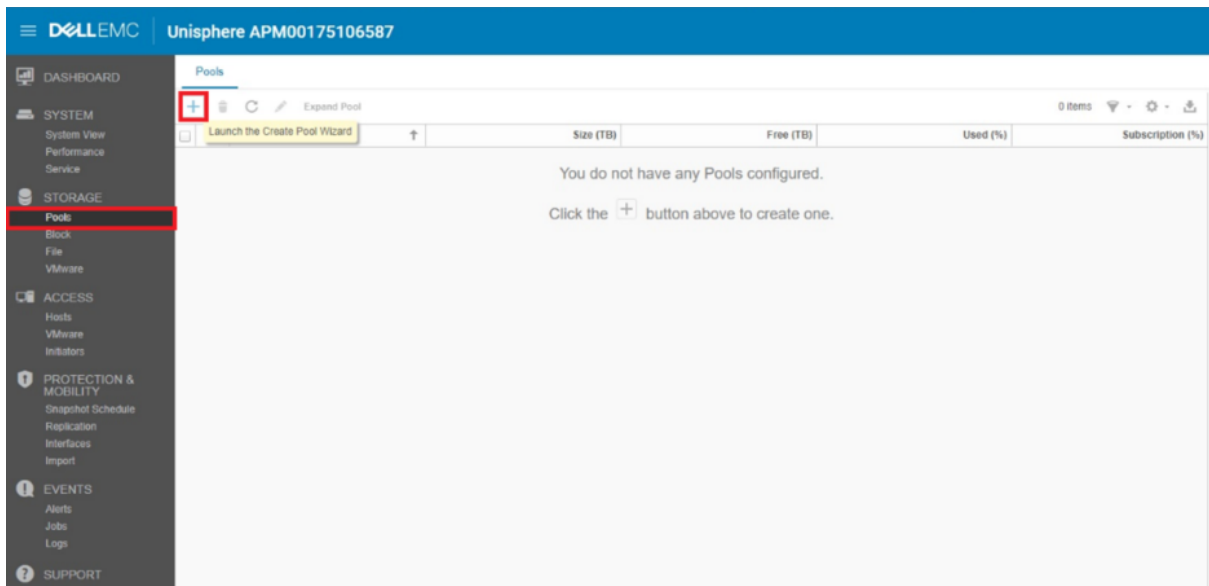
Unisphereにログインすると、メイン ダッシュボードが表示されます。



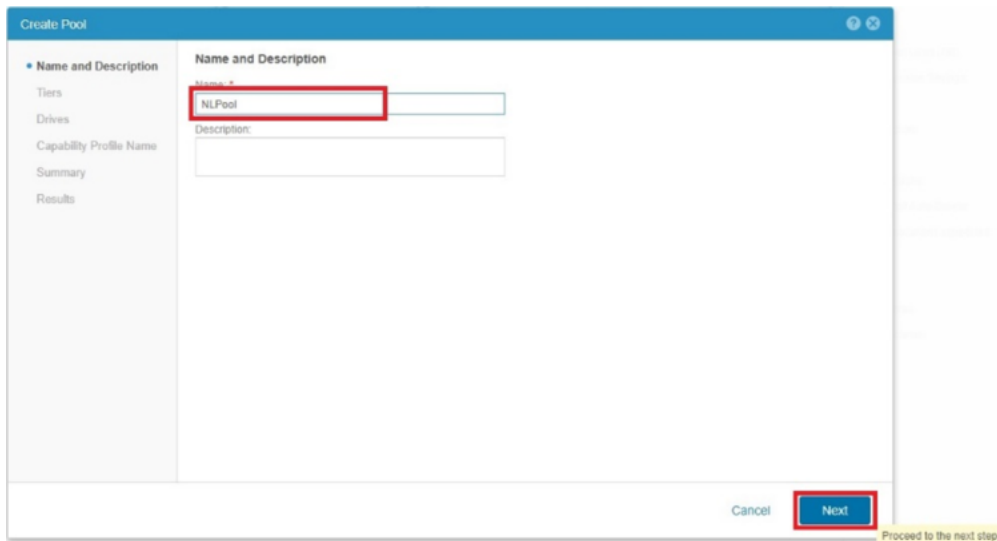
タスク2 - プールを作成する

NetWitness構成は、2つの異なるプールから成ります。1つのプールはNL-SASドライブ専用で、もう1つはSSD専用です。

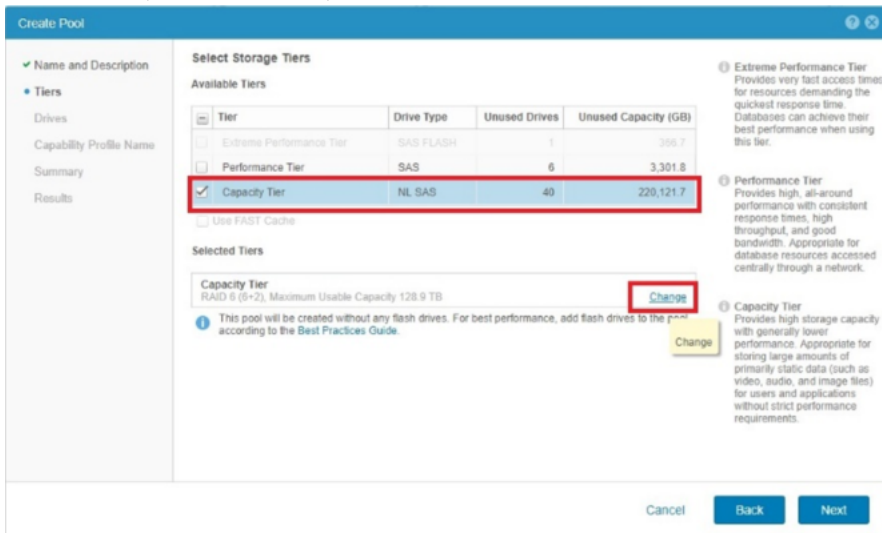
1. [ストレージ]セクションで[プール] > **+** (追加)をクリックして[プールの作成]ウィザードを起動します。



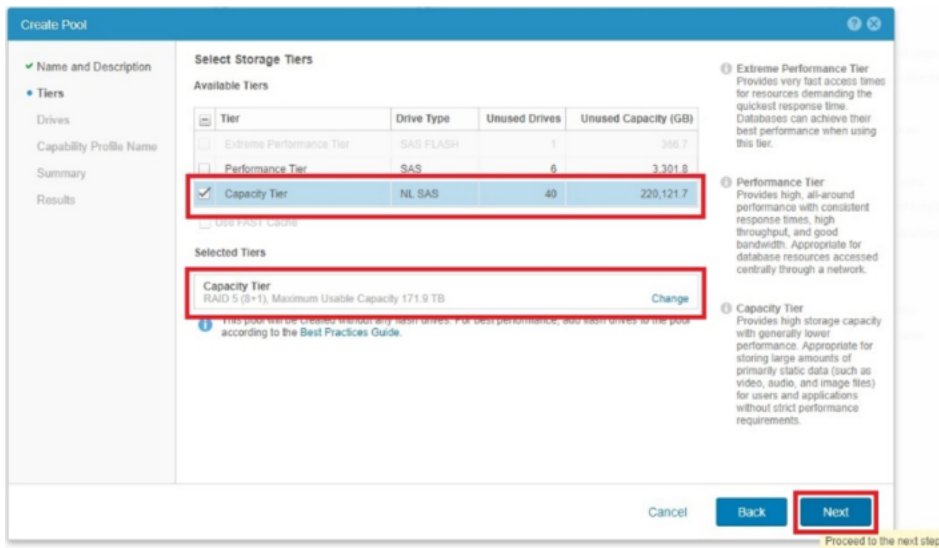
2. プールの名前(NLPoolなど)を入力し、[次へ]をクリックします。必要に応じて、プールの説明も入力できます。



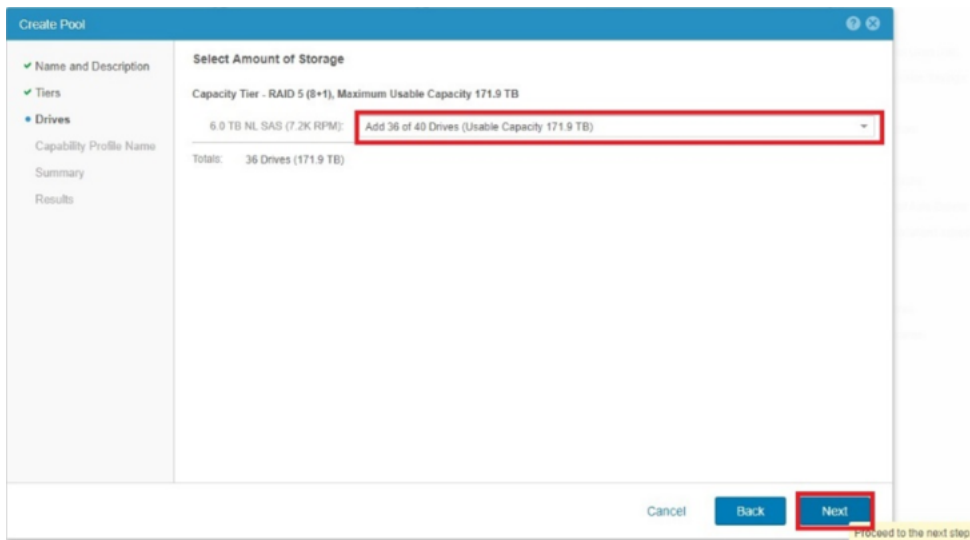
3. 階層タイプ(ドライブタイプ)の[階層]の下にある[容量階層]を選択し、[変更]をクリックします。



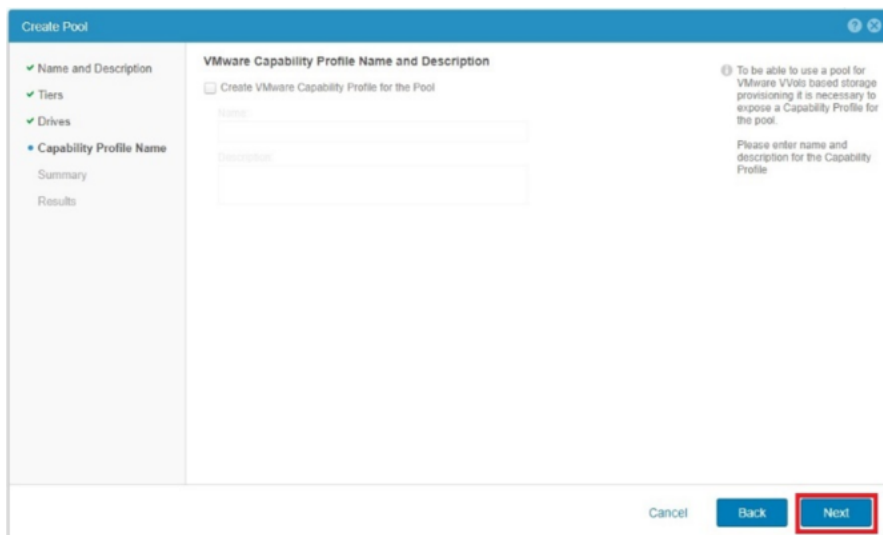
4. RAIDタイプを選択し、ドロップダウンでRAIDサイズを選択します。
RAIDのタイプとサイズは、お客様の要望に従って選択します。唯一の要件は、ログまたはパケットの収集とクエリーに対応するために十分なIOPSをプール内に確保することです。次の例では、RAID 5(8+1)という構成が選択されていますが、お客様によっては、RAID 6(10+2または12+2)が選択される場合もあります。
5. 正しいRAIDタイプおよびサイズを選択していることを確認してください。



6. プールに追加するドライブの数を選択し、[次へ]をクリックします。

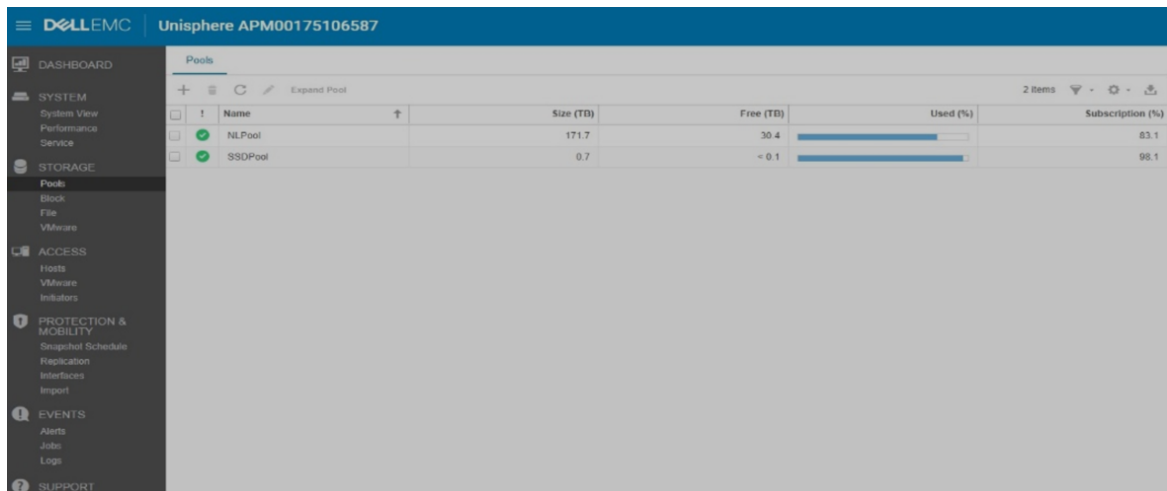


7. [VMware機能] セクションをスキップし、[次へ]をクリックします。



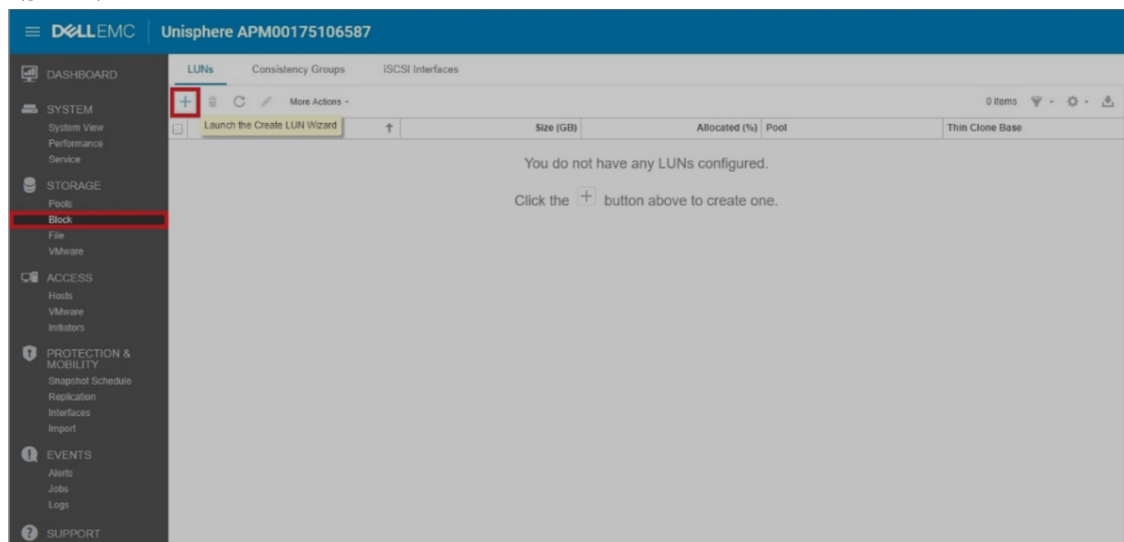
8. すべてが正しいことを[サマリー]タブで確認し、[終了]をクリックします。
9. ステップ2～8を使用して、SSD用に別のプールを作成します。
- 別のプールの名前(SDDPoolなど)を入力し、[次へ]をクリックします。必要に応じて、プールの説明も入力できます。
 - 階層タイプ(ドライブタイプ)の[階層]の下にある[最大パフォーマンス階層]を選択し、[変更]をクリックします。
 - RAIDタイプを選択し、ドロップダウンでRAIDサイズを選択し、[OK]をクリックします。

注: RAID 5(4+1) RAID構成は、容量階層とは異なります。



タスク3 - LUNを作成する

1. [ストレージ] セクションで[ブロック] > **+** (追加) をクリックして[LUNの作成ウィザード]を起動します。

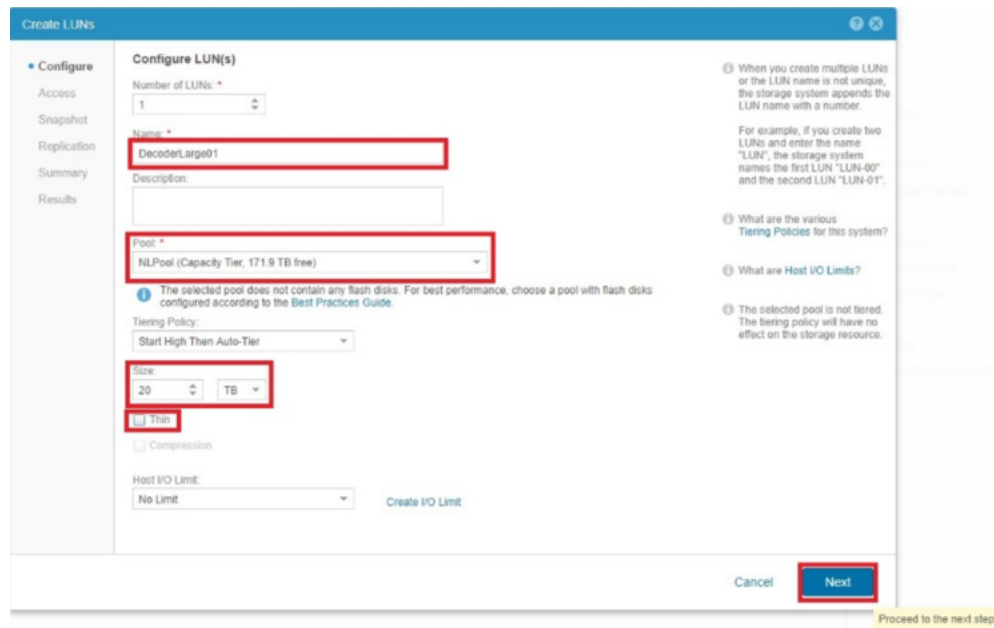


次の表に、作成が必要になる可能性のあるすべてのLUNを示します。SSDプールに割り当てる必要があるLUNはConIndexのみです。LUNサイズは、以下にリストされている値を超えないようにしてください。

DecoderLarge01	75TB以下	NLプール	No
DecoderSmall01	20TB以下	NLプール	No
Concentrator01	15TB以下	NLプール	No
Archiver01	75TB以下	NLプール	No

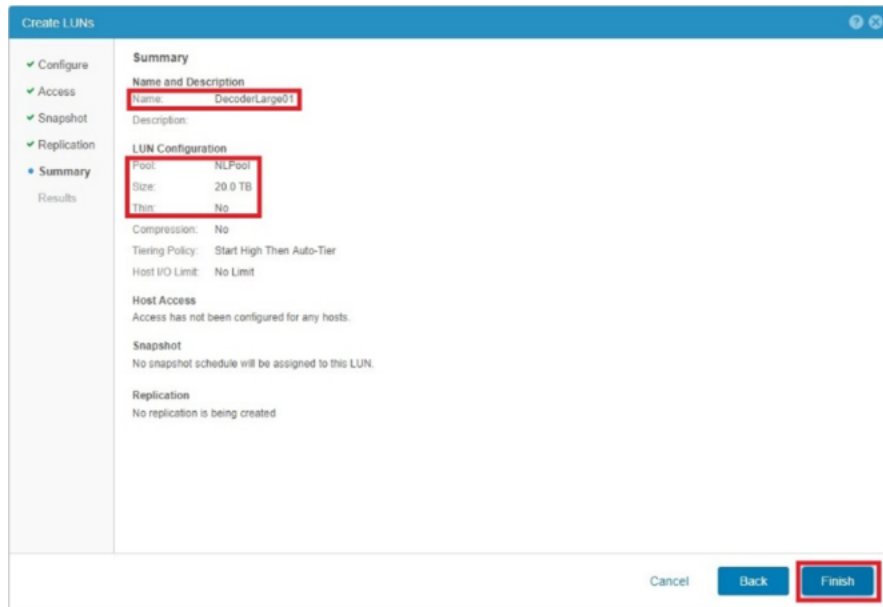
ConIndex01 3TB以下 SSDプール No

2. リストにあるLUN名を入力します。必要に応じて、LUNの説明を入力できます。
3. ドロップダウンメニューのリストから、適切なプールを選択します。
4. [シン] チェックボックスをオフにします(これらは、完全にプロビジョニングされたLUNになります)。
5. [次へ]を選択して次のメニューに進みます。



6. サマリー セクションが表示されるまで[次へ]をクリックします。
7. [名前]、[プール]、[サイズ]、[シン]をすべて正しく選択していることを確認します。

8. [終了]をクリックして、LUNの作成を完了します。



9. ステップ2～8を繰り返して、残りのLUNを作成します。

タスク4 - ホストを登録する

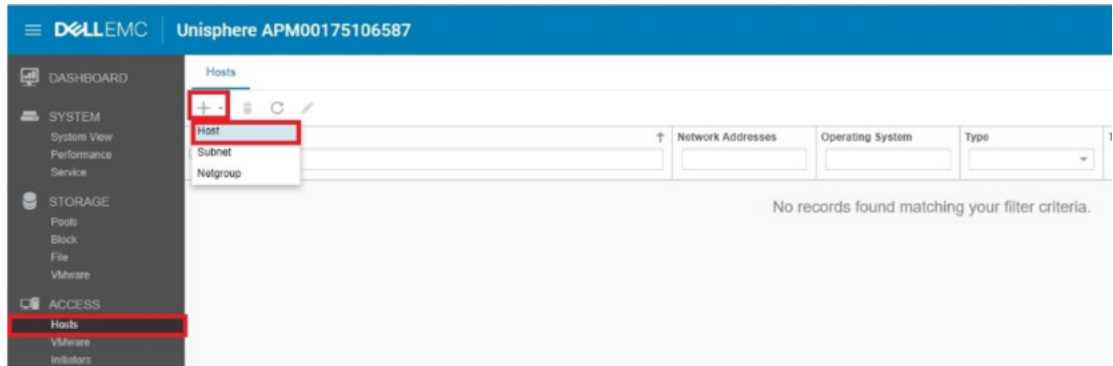
先に進む前に、ヘッド ユニットのホスト名とIPアドレスを記録し、ヘッド ユニットのHBAがUNITYに正しくケーブル接続されていることを確認します。

1. [アクセス]セクションで[イニシエーター]をクリックします。
2. [イニシエーターパス]タブで、ヘッド ユニットの登録に使用する正しいHBAが選択されていることを確認します。

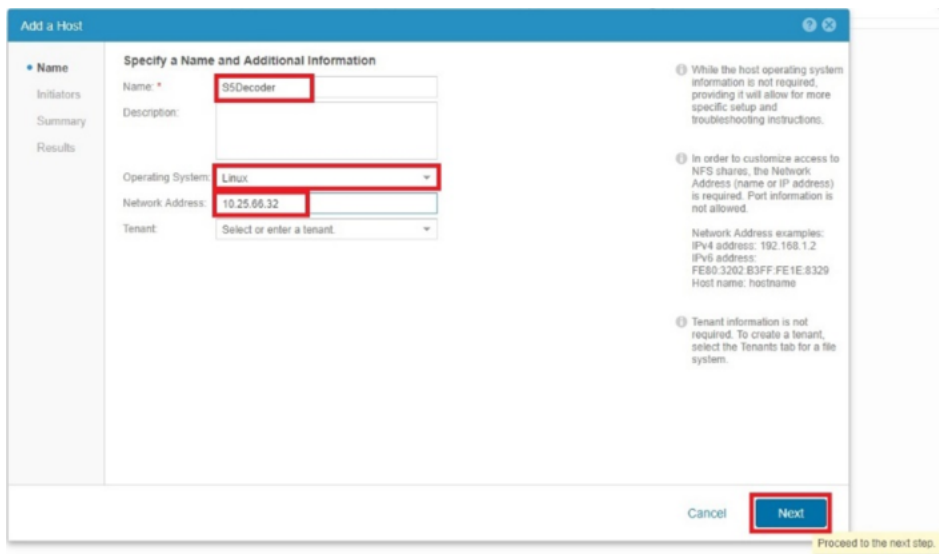
ヘッド ユニットごとに2つのイニシエーターが表示される必要があります。これは、ポート1からSPA、およびポート1からSPBへのファイバー接続を表します。複数のヘッド ユニットがある場合は、それぞれの電源を切り、再び電源を入れ、1つずつ登録することが最も簡単な方法です。

Initiator	Host	Host Type	Target Port	Logged In	Protocol
Initiator IGNWVN					
20:00:90:FA:A7:F3:5A:10:00:00:90:FA:A7:F3:5A	--	--	SP B I/O Module 1 FC Port 3	Yes	FC
20:00:90:FA:A7:F3:5B:10:00:00:90:FA:A7:F3:5B	--	--	SP A I/O Module 1 FC Port 3	Yes	FC
20:00:90:FA:A7:F5:E5:10:00:00:90:FA:A7:F5:E5	--	--	SP A I/O Module 1 FC Port 1	Yes	FC
20:00:90:FA:A7:F5:E7:10:00:00:90:FA:A7:F5:E7	--	--	SP B I/O Module 1 FC Port 1	Yes	FC
20:00:90:FA:A7:FA:BE:10:00:00:90:FA:A7:FA:BE	--	--	SP A I/O Module 1 FC Port 2	Yes	FC
20:00:90:FA:A7:FA:BF:10:00:00:90:FA:A7:FA:BF	--	--	SP B I/O Module 1 FC Port 2	Yes	FC

3. [アクセス] セクションで[ホスト] > **+** (追加) > [ホスト] をクリックしてホスト構成を追加します。



4. ヘッド ユニットのホスト名を入力します。
5. [オペレーティングシステム] の下のドロップダウンメニューで[Linux]を選択します。
6. ヘッド ユニットのIPアドレスを入力します。
7. [次へ] をクリックして次のセクションに進みます。



8. [イニシエーター]セクションで、ヘッドユニットに関連付けられている正しいポートに対応する2つのイニシエーターを選択し、[次へ]をクリックして続行します。

The screenshot shows the 'Add a Host' wizard in the 'Select Discovered Initiators or Manually Add Initiators' step. The 'Automatically Discovered Initiators' table has the following data:

Initiator IQN/WWN	Connected To
<input checked="" type="checkbox"/> 20.00.00.FA.A7.F5.E6:10.00.00.90.FA.A7.F5.E6	SP A iO Module 1 FC Port 1
<input type="checkbox"/> 20.00.00.FA.A7.FA.BF:10.00.00.90.FA.A7.FA.BF	SP B iO Module 1 FC Port 2
<input type="checkbox"/> 20.00.00.90.FA.A7.F3.6A:10.00.00.90.FA.A7.F3.6A	SP B iO Module 1 FC Port 3
<input checked="" type="checkbox"/> 20.00.00.90.FA.A7.F5.E7:10.00.00.90.FA.A7.F5.E7	SP B iO Module 1 FC Port 4

The 'Manually Added Initiators' section is empty with a message: 'No initiators have been manually added yet. Click the + button to manually add an initiator.'

Buttons: Cancel, Back, Next. A yellow callout box says 'Proceed to the next step.'

9. 名前、OS、IP、WWNが正しいことを確認し、[終了]をクリックします。

The screenshot shows the 'Add a Host' wizard in the 'Review the host configuration' step. The configuration is as follows:

- Name: SSDecoder
- Description: (empty)
- Operating System: Linux
- Network Addresses: 10.25.66.32
- Tenant: (empty)

The 'Initiators to be registered with this host' table has the following data:

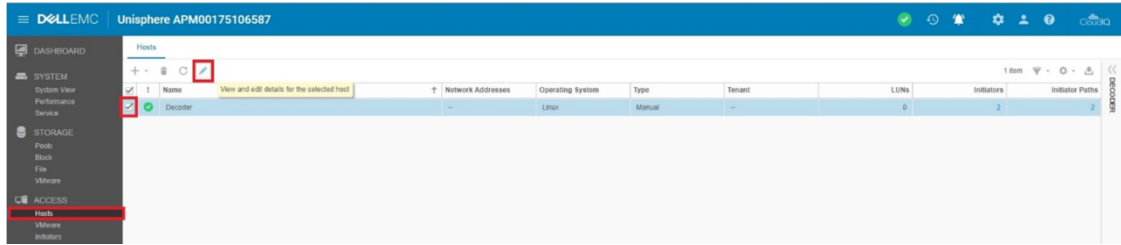
Protocol	Initiator IQN/WWN
FC	20.00.00.90.FA.A7.F5.E6:10.00.00.90.FA.A7.F5.E6
FC	20.00.00.90.FA.A7.F5.E7:10.00.00.90.FA.A7.F5.E7

Buttons: Cancel, Back, Finish.

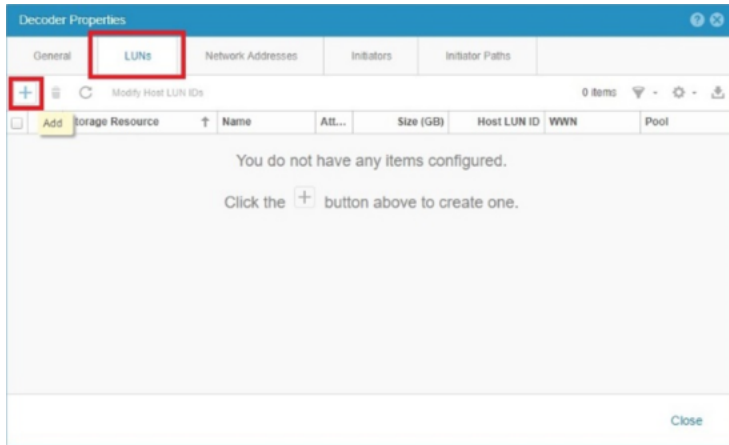
10. すべてのヘッドユニットについてステップ2～9を繰り返します。
11. [イニシエーター]セクションで、ヘッドユニットに関連付けられている正しいポートに対応する2つのイニシエーターを選択します。[次へ]をクリックして続行します。

タスク5 - LUNをホストに割り当てる

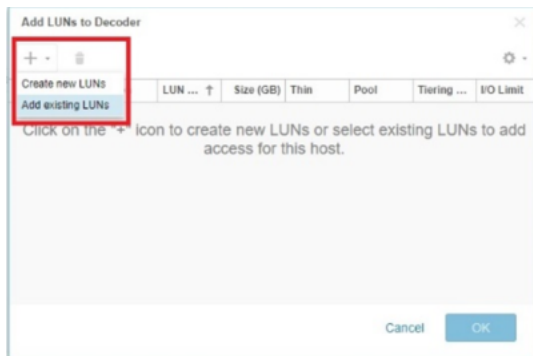
1. [アクセス]セクションで[ホスト]をクリックし、ヘッドユニット([Decoder]など)を選択し、(編集)をクリックして、選択したホストの詳細を表示し、編集します。



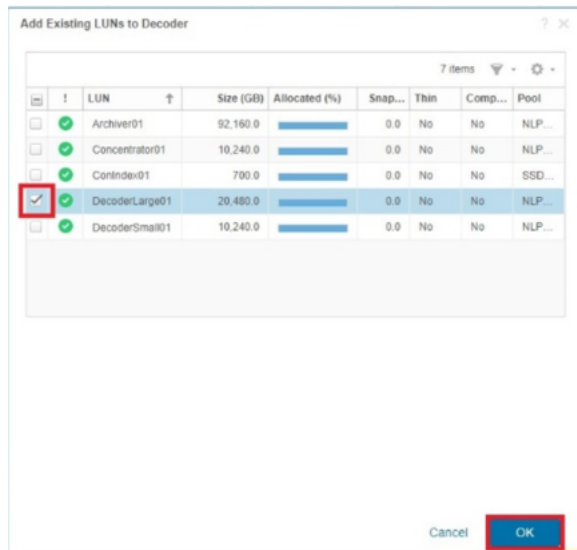
2. [プロパティ]セクションで[LUN]タブをクリックし、**+** (追加アイコン)をクリックします。



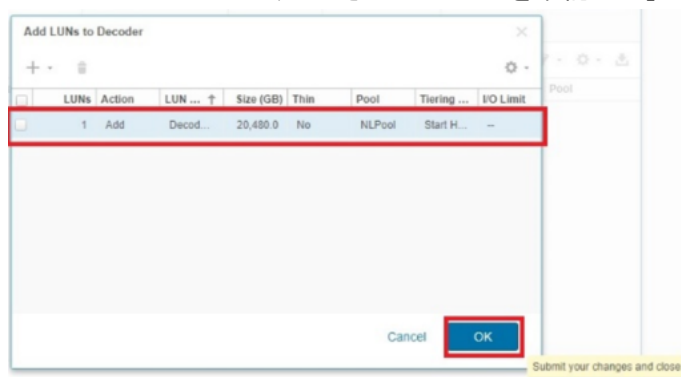
3. [**<Host>にLUNを追加**]ポップアップで**+** > [**既存のLUNの追加**]をクリックします。



4. ヘッド ユニットに追加するLUNを選択し、[OK]をクリックします。

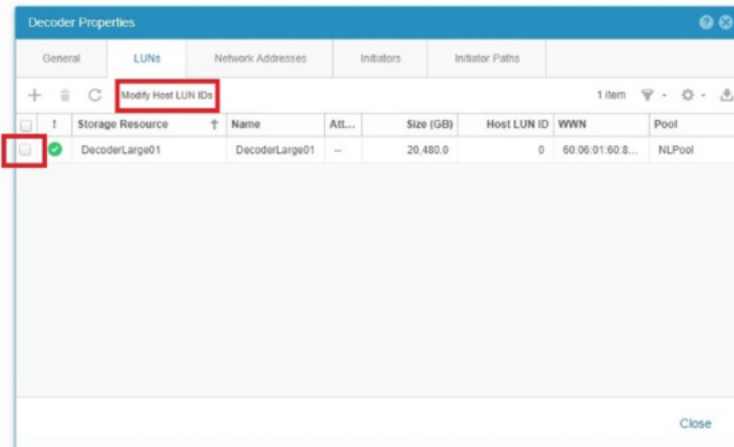



5. 正しいLUNがホストに追加されていることを確認し、[OK]をクリックします。



6. (オプション) HLU(ホストLUNの一意のID)を変更する必要がある場合は、次の手順を実行します。

- a. 変更するLUNを選択します。
- b. [ホストLUN IDの変更]をクリックします。



7.  (編集)をクリックし、必要な数字にHLUを変更し、[OK]をクリックします。

タスク6 - PowerPathをインストールする

1. DecoderホストのEmulexポートがUnityに接続されていることを確認します。
2. Unityに接続されたDecoder上のrootにadmin資格情報でログインします。
3. PowerPathをインストールし、Unityハードウェア用のDell EMC PowerPathライセンスを登録します。

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

注: RSAが提供するUnityを購入すると、PowerPathライセンスが届きます。PowerPathは support.dell.comからダウンロードできます。

注: Dell EMCからダウンロードしたRPMは、RSAデバイスで使用できる証明書で署名されていない可能性があります。このためにpackage not signedエラーでインストールが失敗する場合があります。ソフトウェアのインストールを有効にする--ngpgcheckオプションを使用してyumインストールを実行します。

4. すべてのPowerPath接続が正しいことを確認します。
- ```
powermt display dev=all
```

次の出力は、有効なPowerPath接続の例です。

```

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

Pseudo name=emcpowerb
Unity ID=APM00174407815 [Host_62]
Logical device ID=600601609D9046006996745A46B60AB6 [DecoderSmall01]
state=alive; policy=CLAROpt; queued-I/Os=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-I/Os Errors
=====
 15 lpfc sdd SP A6 active alive 0 0
 18 lpfc sdf SP B6 active alive 0 0

```

5. emcpregコマンドを使用してPowerPathライセンスがインストールされていることを確認します。  

```
[root@NWAPPLIANCE24932 ~]# emcpreg -list
```

```
Key BQPO-DB4M-VFC2-Q24R-ML9Z-EQTU
```

```
Product: PowerPath
```

```
Capabilities: A1
```
6. 次の文字列を/etc/lvm/lvm.confファイルに追加して、LVM(論理ボリューム マネージャー)をフィルタリングすることで、重複したボリュームが無視されるようにします。  

```
filter = ["a|^/dev/sda2$|", "a|^/dev/sdb1$|",
"a|^/dev/emcpower.*|", "r|.*/|"]
```
7. 次のコマンドを、次の順序で実行します。
  - a. `systemctl enable PowerPath.service`
  - b. `systemctl start PowerPath.service`
8. Decoderを再起動します。
9. [「REST APIを使用したストレージの構成」](#)にあるステップを完了してストレージ構成を完了します。

## 別のストレージタイプへのデータの移行

このセクションでは、DACからPowerVaultにデータを移動するための2つの選択肢について説明します。

### [Warm階層およびHot階層オプションを使用したデータ移行](#)

#### [DACからPowerVaultへのデータの移動](#)

RSA NetWitness Platformホストおよびストレージハードウェアをセットアップする詳細な手順については、RSAリンク(<https://community.rsa.com/community/products/netwitness/hardware-setup-guides>)にあるハードウェアセットアップガイドを参照してください。



## Warm階層およびHot階層オプションを使用したデータ移行

この手順では、DACのWarm階層を構成して、新しいデータをDACが書き込まないようにします。Warm階層は、分析操作には引き続き使用できます。PowerVaultをHot階層として構成します。ここに、新しいデータが書き込まれ、それをアナリストが使用できます。必要なデータをHot階層で保持できる場合は、Warm階層を廃止できます。

Warm階層とHot階層を設定するには、次のタスクを実行します。

- [サービスを停止する](#)
- [PowerVaultを設定する](#)
- [マウントポイントを構成する](#)
- [Warm階層とHot階層を設定する](#)
- [DACを廃止する](#)

### サービスを停止する

1. NetWitness Platformユーザーインターフェイスにログインします。
2.  (管理) > [サービス]に移動し、サービス(Log Decoderなど)を選択します。
3.  > [表示] > [構成]をクリックします。次に、[Log Decoderの構成]で[収集の自動開始]チェックボックスをオフにし、[適用]をクリックします。
4. メニューバーで、[構成]の隣にある下矢印をクリックし、[システム]を選択します。次に、パネルの上部で[収集の停止]をクリックします。
5. NwConsoleのコマンドラインインターフェイスで、次のコマンドを実行してサービスを停止します。  
`systemctl stop nwlogdecoder`

### PowerVaultを設定する

1. サービス(この例ではLog Decoder)のIPアドレスを入力して、サービスのREST APIに移動します。たとえば、172.16.0.1:50106と入力します。
2. サービスの隣にあるアスタリスク(\*)をクリックします(例: decoder (\*))。

3. [/decoderのプロパティ]の下の下矢印をクリックし、[RaidNew]を選択します。次に、以下のパラメーターを入力し、schemeのサービスの名前を入力します。この例では、「logdecoder.  
controller=1 enclosure=75 scheme=logdecoder commit=1」を使用します。
4. [送信]をクリックします。
5. パーティションを構成するには、下矢印をもう一度クリックし、[PartNew]を選択してから、以下のパラメーターを入力します。  
name=sde service=logdecoder volume=logdecodersmall commit=1
6. [送信]をクリックします。
7. [PartNew]を選択したままで、次のパラメーターを入力します。  
name=sdf service=logdecoder volume=logdecoder commit=1

**注:** パーティション定義をコミットする前に検証するには、これらのパラメーターをcommit=1なしで入力し、[送信]をクリックします。パラメーターを検証した後に、#commit=1を追加し、[送信]をクリックしてパラメーターの設定をコミットします。



## マウント ポイントを構成する

1. NwConsole上で、サービス(Log Decoderなど)のrootレベルでdf -hを実行します。  
マウントされたパーティションのリストが表示されます。
2. DACの古いストレージ ポイントをすべてアンマウントし、すべてのデータをLog Decoderにコピーします。rootレベルで、各パーティションのパス名を指定してumountコマンドを実行します。パス名は、次の例のように連結できます。  
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. ファイルにアクセスするために、/mntディレクトリのdecorootフォルダ内のパーティションを一時的にマウントします。以下に例を示します。  
mount /dev/mapper/logdecodersmall-decoroot /mnt/decoroot/
4. decorootの内容を、/mntから/var/netwitness/logdecoderにコピーし、プロンプトに「Y」(はい)と答えます。  
cp -R statdb /var/netwitness/logdecoder/
5. /mnt/decorootをアンマウントします。  
umount /mnt/decoroot
6. decorootを/etc/fstabからコメントアウトします。これは、DACにあり、DACは廃止されるためです。  
#/dev/logdecodersmall/decoroot  
/var/netwitness/logdecoder/xfs/noatime,nosuid 1 2
7. 残りのすべてのファイルシステムをマウントします。  
mount -a
8. nwlogdecoderサービスを開始します(収集を無効にしたままで)。  
systemctl start nwlogdecoder



## Warm階層とHot階層を設定する

**注意:** Warm階層とHot階層を設定する前に、各コレクションの適切なWarm階層とHot階層のエントリがわかっており、正確に設定できることを確認してください。

1.  (管理) > [サービス]に移動し、サービス(Log Decoderなど)を選択します。
2. Log Decoderサービスの場合は、 > [表示] > [エクスプローラ]をクリックし、[database] > [config]に移動します。
  - a. 次の例に示すように、meta.dirの内容をコピーし、meta.dir.warmに貼り付けます。

| logdecoder - Log Dec... < |                        | logdecoder - Log Decoder                  |
|---------------------------|------------------------|-------------------------------------------|
| /database/config          | hash.algorithm         | none                                      |
|                           | hash.databases         | session,meta,packet                       |
|                           | hash.dir               |                                           |
|                           | manifest.dir           |                                           |
|                           | meta.compression       | none                                      |
|                           | meta.compression.level | 0                                         |
|                           | meta.dir               | /var/netwitness/logdecoder/metadb=4.58 TB |
|                           | meta.dir.cold          |                                           |
|                           | meta.dir.warm          |                                           |
|                           | meta.file.size         | auto                                      |
|                           | meta.files             | auto                                      |

| logdecoder - Log Dec... < |                        | logdecoder - Log Decoder                  |
|---------------------------|------------------------|-------------------------------------------|
| /database/config          | hash.algorithm         | none                                      |
|                           | hash.databases         | session,meta,packet                       |
|                           | hash.dir               |                                           |
|                           | manifest.dir           |                                           |
|                           | meta.compression       | none                                      |
|                           | meta.compression.level | 0                                         |
|                           | meta.dir               | /var/netwitness/logdecoder/metadb=4.58 TB |
|                           | meta.dir.cold          |                                           |
|                           | meta.dir.warm          | /var/netwitness/logdecoder/metadb=4.58 TB |

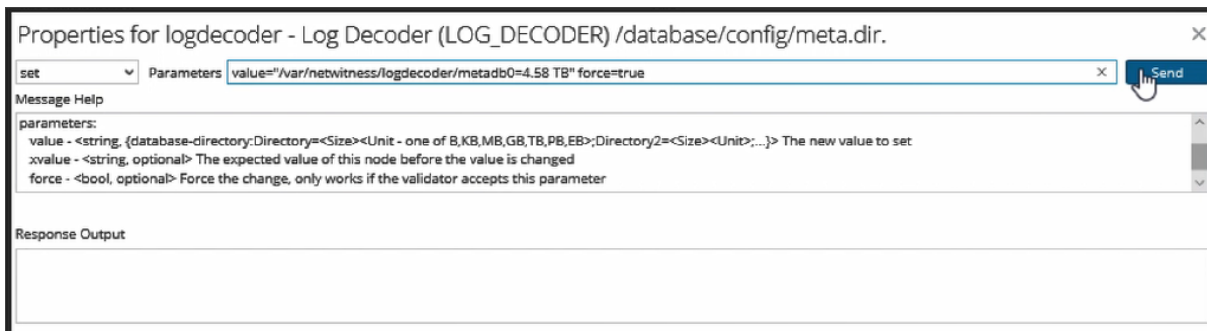
- b. 同様に、packet.dir内のパケット データベースをpacket.dir.warmにコピーします。
  - c. session.dir内のセッション データベースをsession.dir.warmにコピーします。
3. [index] > [config]に移動し、index.dirをindex.dir.warmにコピーします。

新しいボリュームの名前は0で終わるため、PowerVaultは、0で終わる名前のディレクトリに書き込みます。以下に例を示します。

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

パスに0を追加して、PowerVaultマウントへのパスでDecoder構成を更新します。

1. /database/config列で[meta.dir]を右クリックし、[プロパティ]をクリックします。
2. [logdecoderのプロパティ]で[set]を選択し、この例に示すように[パラメータ]に value="/var/netwitness/logdecoder/metadb0=4.58 TB" and add force=trueと入力します。次に、[送信]をクリックします。





3. session.dir、packet.dir、index.dirについて、ステップ2を繰り返します。「=xx GB」のサイズがDACと同じかどうかを気にする必要はありません。これは、次のステップで更新されます。

**注：**ここでは、PowerVaultパスを\*.dir値に入れるだけです。


4. ライブPowerVaultボリュームのサイズを更新します。
  - a. Log Decoderの[エクスプローラ]ビューの左側のパネルで[database]右クリックし、[プロパティ]をクリックします。
  - b. [reconfig]を選択し、[パラメータ]でupdate=1と入力します。次に[送信]をクリックします。
  - c. indexについて、ステップaおよびbを繰り返します。

5. `systemctl restart nwlogdecoder`サービスを再起動します。




6.  (管理) > [サービス]に移動し、Log Decoderサービスを選択し、 > [表示] > [システム]をクリックします。
7. [収集の開始]をクリックします。
8. [構成]ビューに移動し、[収集の自動開始]を選択します。次に[適用]をクリックします。
9. ホストをリポートします。


## DACを廃止する

DACデータが古くなったら、[エクスプローラ]ビューに戻り、セッション、メタ、パケット、インデックスの

\*.dir.warm構成をすべて削除する必要があります。Log Decoderで > [表示]で[エクスプローラ]ビューに移動すると、DACデータが古くなった時期を特定できます。Hot階層とWarm階層があるため、構成統計が2セットあることに注意する必要があります。たとえば、パケットDecoderの場合は、packet.oldest.file.time内のパケットの最も古い時間と、packet.oldest.file.time.hotの値を確認します。そして、30日前までDACにストレージがあった場合は、DACをオフラインにし、廃止することができます。

これは、DACを廃止するための基本的なステップです。DACを廃止する場合は、カスタマーサポート担当者の協力を得ることをお勧めします。

1.  (管理) > [サービス]に移動し、サービス(Log Decoderなど)を選択します。
2.  > [表示] > [構成]をクリックします。次に、[Log Decoderの構成]で[収集の自動開始]チェックボックスをオフにし、[適用]をクリックします。
3. メニューバーで、[構成]の隣にある下矢印をクリックし、[システム]を選択します。次に、パネルの上部で[収集の停止]をクリックします。
4. NwConsoleのコマンドラインインターフェイスで、次のコマンドを実行してサービスを停止します。  
systemctl stop nwlogdecoder
5. Warm階層をアンマウントします。rootレベルで、各パーティションのパス名を指定してumountコマンドを実行します。パス名は、次の例のように連結できます。  
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. /etc/fstabから、古いDACデータベースをすべてコメントアウトして、PowerVaultデータベースのみが残るようにします。
7. サービスを開始します。  
systemctl start nwlogdecoder
8. ユーザーインターフェイスで (管理) > [サービス]に移動し、LogDecoderサービスを選択します。

9.  > [表示] > [エクスプローラー] をクリックし、Warm階層構成を削除します。
  - a. [database] > [config] で、meta.dir.warm、packet.dir.warm、session.dir.warmの内容を削除します。
  - b. [index] > [config] で、index.dir.warmの内容を削除します。
  - c. [構成]ビューに移動し、[収集の自動開始]を選択します。次に[適用]をクリックします。
  - d. [システム]ビューに移動し、[収集の開始]をクリックします。

10.

`systemctl restart nwlogdecoder` サービスを再起動します。

これで、DACがアンマウントされ、DecoderでWarmストレージ用の構成から削除され、完全に消去できる状態になります。

1. 論理ボリュームを削除します。lvscanを実行して、論理ボリュームのリストを取得します。
2. 古い論理ボリュームでlvremoveを実行します。以下に例を示します。  

```
/dev/logdecodersmall/decroot /dev/lvremove /dev/logdecodersmall/index
/dev/logdecodersmall/sessiondb /dev/logdecodersmall/metadb
/dev/logdecodersmall/packetdb
```
3. ボリューム グループを削除します。vgscanを実行して、ボリューム グループのリストを取得します。
4. 古いボリューム グループでvgremoveを実行します(名前が0で終わるボリューム グループはPowerVaultであるため、削除しないように注意してください)。
5. pvscanを実行して、解放されたブロック デバイスを表示します。
6. DACが正常に削除されたら、ホストを再起動します。

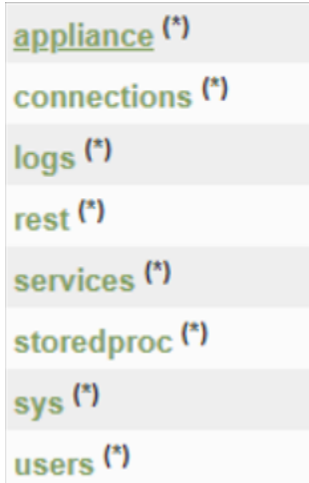
## DACからPowerVaultへのデータの移動

次の手順では、DACからPowerVaultにデータを移動する方法を説明します。2つのDACから2つのPowerVaultにデータを移動する前に、2つのDACが接続され、Decoderに対して構成された状態でpvvs(物理ボリューム サイズ) コマンドをDecoder Linuxコンソールから実行(またはDecoderにSSHで接続)した場合は、次のような表が表示されます。列見出しは、物理ボリューム(PV)、ボリューム グループ(VG)、Linuxフォーマット(Fmt)、Linux属性(Attr)、物理ボリューム サイズ(PSize)、物理ボリューム空き領域(PFree)です。

| 現在価値      | 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     |

次のステップを実行して、DACからPowerVaultにデータを移動します。

1. 2つのPowerVaultを、Decoder上の別のPERCコントローラーに接続します。
2. デバイスを作成します。
  - a. ブラウザーを開き、Network Decoderとポート50106のIPアドレスを指定してRESTツールにアクセスします。
  - b. adminアカウントの認証情報でログインします。



- c. [appliance]の隣の(\*)をクリックしてRESTコマンドセットにアクセスします。
- d. raidListを実行して、コントローラー/エンクロージャの組み合わせおよび新しいPowerVaultエンクロージャを表示します。  
次の例の出力では、コントローラー2、エンクロージャ246に/dev/sdgと/dev/sdhが表示されています。

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

- e. [/applianceのプロパティ]でraidNewを選択し、controller=<PowerVault-controller-id> enclosure=<PowerVault-enclosure-id> scheme=decoder preferSecure=falseを指定します。次に[送信]をクリックします。

**注：**PowerVaultドライブがSEDドライブでない場合は、「preferSecure=false」を指定します。PowerVaultドライブがSEDドライブであり、それらを暗号化しない場合は、「preferSecure=false」を指定します。PowerVaultドライブがSEDドライブであり、それらを暗号化する場合は、「preferSecure=true」を指定する必要があります。

3. Decoder Linuxコンソールに移動するか、DecoderにSSH接続し、次のコマンドを実行します。
 

```
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
```

ボリュームが正常に作成された場合は、次のメッセージが表示されます。

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

**注:** 各ブロックデバイスに対して、このステップを繰り返します。ブロックデバイス名は、percカードスロットあたりのエンクロージャ数に応じて異なる場合があります。

4. 次のコマンド文字列を実行して、DACボリュームグループ( `decoder`、`decodersmall`) をPowervault物理ボリュームに拡張します。

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

5. 次のコマンド文字列を実行してDACからPowerVaultにデータを移動します。この次のコマンド文字列では、`/dev/sdc`がDACで、`/dev/sdg`がPowerVaultです。

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

**注:** 1) `pvmove`コマンドは、ボリューム間でデータを同期して、移行の実行中にNetWitnessがデータの取得または集約を継続できるようにします。失敗する場合は、`pvmove`コマンドを複数回、実行できます。2) ドライブ上のデータ量によっては、データの移動に時間がかかる場合があります。たとえば、テストでは、1 TBのデータの移動に4時間かかりました。

6. 移動が完了した後に、次のコマンドを実行してDACドライブを削減し、取り外します。

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

7. DACからホストへの物理接続を切断します。
8. 物理ボリュームがDACからPowerVaultに移動されていることを確認します。

- a. ホストをリブートします。

```
reboot
```

- b. `/etc/fstab`ファイルが正しいことを確認します。

- c. `pvs`コマンドを実行し、PowerVault上のPSizeとPFreeの値が正しいことを確認します。

```
root@nitirercdecoder ~# pvs
 PU UG 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
```

## DACから移動した後のPowerVault上のデータ

2つのDACから2つのPowerVaultにデータを移動した後に、2つのPowerVaultが接続され、Decoderに対して構成された状態で `pvs(物理ボリュームサイズ)` コマンドをDecoder Linuxコンソールから実行(またはDecoderにSSHで接続)した場合は、次のような表が表示されます。列見出しは、物理ボリューム(PV)、ボリュームグループ(VG)、Linuxフォーマット(Fmt)、Linux属性(Attr)、物理ボリュームサイズ(PSize)、物理ボリューム空き領域(PFree)です。

| 現在価値      | 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  |



## 付録A: NetWitness Platformホストによるデータの保存方法

ほとんどの場合、NetWitness PlatformのDecoder、Log Decoder、Concentrator、Archiver、Hybridホストに、データを保存するための外部ストレージが必要です。外部ストレージの使用方法や、外部ストレージで期待されるスループットとパフォーマンスは、ホストごとに異なります。一部のホストでは、高い頻度でシーケンシャル書き込みが発生しますが、他のホストでは、ランダム読み取り/書き込みが発生する頻度の方が高くなります。

### Decoderホスト

Log DecoderとNetwork Decoderは、データを収集し、メタを解析します。この2つのホストの違いは、収集するデータのタイプです。

- Log Decoderはログを収集します。
- Network Decoderはパケットを収集します。

Log DecoderとNetwork Decoderはどちらも、収集したRAWトラフィックからメタデータを解析します。次に、メタデータが、インデックス作成のためにConcentratorに集約されます。ホストには、RAWペイロードデータ(RAWパケットまたはRAWログ)と、Concentratorの集計のためのデータ取得中に抽出されたメタデータ用のキャッシュを格納するストレージが必要です。

保存要件は、RAWパケットまたはRAWログに必要なストレージの量を決定する際の重要な要素です。ほとんどの導入において、時間の経過とともに、保存要件の増加や収集レートの増加に基づいてストレージが追加されます。RAWデータ用ストレージは、ランダム読み取りを伴う大量のシーケンシャル書き込みをサポートする必要があります。特に、高速なNetwork Decoder環境の場合は、少なくとも2つのパーティションをホストに公開して、読み取りおよび書き込み用のパーティションの間のスロットリングをサポートすることを推奨します。

Decoder上のメタキャッシュは、一般にサイズが固定されていますが、Decoderとそれに対応するConcentratorの間の接続が失われた場合に備えて、追加のキャッシュがサポートされるように拡張できます。メタキャッシュは、Decoderが抽出したメタによる書き込みと、Concentratorに集計されるメタへの読み取りに対応した、ランダムIOPSレートをサポートする必要があります。

### Concentratorホスト

Concentratorは、Decoderからのメタデータを集約し、そのインデックスを作成します。メタとインデックスの両方のストレージのニーズは、NetWitness Platform導入の保存に関する要件に基づいて調整されます。Decoderに格納されるRAWデータと同様に、メタおよびインデックスの両方のストレージを、時間の経過とともに増やして、保存に関する要件を満たすことが必要になる場合があります。

メタストレージには、Network DecoderまたはLog Decoderから抽出されたすべてのメタデータが格納されます。抽出するメタの量の比率は変化しますが、メタストレージのパフォーマンスに対する要求は、パケット収集環境とログ収集環境の両方に対する要求と同じです。メタストレージについては、メタデータのランダム読み取りを伴うシーケンシャル書き込みの一定量の持続をサポートする必要があります。

インデックスストレージには、Concentratorに集計されたメタデータから生成された作成中のインデックスが格納されます。インデックスのサイズは、メタストレージエリアのサイズに直接関係します。持続される書き込みのIOPSのサポートに加え、インデックスでは、アナリストとの対話、レポート、アラートによって発生したクエリに基づいて、メタで認識されるよりも高い読み取りレートのIOPSをサポートすることも必要です。



### Archiverホスト

Archiverホストには、メタとRAWログのストレージの両方に1つのパーティションが必要です。ストレージプールでは、Log Decoder、Network Decoderから書き込まれる長期データのシーケンシャル書き込みと、レポートと解析のためのランダム読み取りを主に扱います。

### Hybridホスト

Hybridでは、1つのホストで2つ以上のサービスをホストします。以下に例を示します。

- Network Hybridでは、パケットを排他的に処理するDecoderサービスとConcentratorサービスの両方をホストします。パケット データを収集し、このデータのインデックスをConcentratorサービスに対して作成します。ストレージ パフォーマンスに対する要求は、専用Network Decoderホストと専用Concentratorホストに関するものと一致します。
- Log Hybridは、ログを排他的に処理するLogDecoderサービスとConcentratorサービスの両方をホストします。ログ データを収集し、そのインデックスをConcentratorサービスに対して作成します。パフォーマンスの要件に対する予想は、専用Log Decoderと専用Concentratorの概要と一致します。
- Endpoint Log Hybridは、Endpoint Server、Log Decoder、Concentrator、Log Collector、Endpoint Brokerの各サービスをホストします。Windows、Mac、Linuxの各ホストからのエンドポイントデータの収集と管理、およびWindowsホストからのログファイルとWindowsログの収集を行い、メタデータを生成して、ログやパケットなどの他のイベント ソースからのセッションとエンドポイント データを関連させます。

### SAN構成のオプション

ストレージ エリア ネットワーク (SAN) を使用する場合は、それ以外のRSAストレージ デバイスに使用するものと同じ基本的なドライブ グループとパーティション構成を使用します。SAN構成とオーバーヘッドによっては、PowerVaultまたはDACと同じパフォーマンスで動作するために、より多くのエンクロージャとドライブがSAN構成に必要な場合があります。SAN、PowerVault、DACのどれを使用するか決定する際は、SANの追加のオーバーヘッドが、必要な最小要件を決定するにおいて、重要になります。

### パフォーマンスに関する推奨事項

RSAは、Packet DecoderとLog Decoderに2つのLUNまたはブロック デバイスを割り当てることを推奨しています。1つはパケット データ用で、もう1つは他のすべてのデータベース用です。これにより、高帯域幅のパケット データベースを他のデータベースから分離して、I/O帯域幅が他のアクティビティと競合しないようにすることができます。

Concentratorには、優れたパフォーマンスを得るために、個別のSSDベースのインデックス ボリュームが必要です。このインデックス ボリュームは、NL-SASに保存できるConcentratorメタ データベース ボリュームとは別のRAIDグループに格納する必要があります。Archiverは、アプライアンスごとに1つの大容量NL-SASストレージ ボリュームを使用できます。

## SEDドライブと非SEDドライブが混在するホスト上のSED対応ドライブグループでのセキュリティの有効化

SEDドライブと非SEDドライブの両方がアプライアンスに混在している場合は、encryptSedVd.pyがSED対応仮想ドライブの識別に失敗する可能性があります。SED対応と非SED対応の両方の仮想ドライブがホストに存在する場合は、以下の手順が適用されます。

1. アプライアンスにSSHで接続し、PERC H740(ミニ)アダプターでセキュリティを有効にします。このアダプターのコントローラー番号は0です。PERC H840アダプターは1として表示されます。アプライアンス上のすべてのコントローラーを一覧表示するには、次のコマンドを実行します。

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

最初の列(Ctl)には、アプライアンス上のコントローラーインデックスが一覧表示されます。この場合、コントローラー"0"は"PERC H740 Mini"に、コントローラー"1"は"PERC H840アダプター"に対応します。列"DG"および"VD"には、コントローラー上の仮想ドライブおよびドライブグループが表示されます。

```
[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 PERC740PMini 8 10 3 0 3 0 Opt On - N 0 Opt
1 PERC840Adapter 8 12 1 0 1 0 Opt On - N 0 Opt
[root@116Decoder perccli]#
```

2. "PERC H740 (mini) Adaptor"(コントローラー"0"など)でセキュリティを有効にするには、次のコマンドを実行します。

```
/opt/MegaRAID/perccli/perccli64 /c0 set securitykey='<SOME_STRING_VALUE>'!'
keyid='< SOME_STRING_VALUE >'
```

例:

```
/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. セキュリティを有効にしようとしているSED対応ドライブに対応する正しいドライブグループ(DG)/仮想ドライブ(VD)を特定します。

```
/opt/MegaRAID/perccli/perccli64 /c0 /vall show | egrep -A5 'DG/VD'
```

最初の2列と最後の列を参照して、6つのSED対応ドライブに対応する正しいドライブグループ(DG)/仮想ドライブ(VD)を特定します。シリーズ6アプライアンスでは、RAID6を使用するDG/VDは1つだけです。[NAME]列を使用して、VDまたはDGを特定することができます。この場合、DG/VDは"2"です。"Type"列、"Name"列、"Size"列(これらは、上記でVDが作成されるときにユー

ザーによって定義されます)の組み合わせを使用しています。

```
[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. ディスクグループ(6台のSED対応ドライブから作成)のセキュリティを有効にするには、次のコマンドを実行します。

```
/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. コントローラーで"0"で使用されるエンクロージャID(EID)を取得します。この場合は"64"です。

```
/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. ドライブ/ドライブグループ(DG)がSED対応かつ安全であることを確認するために、以下のコマンドを実行し、SED対応、安全、SED対応の各フラグが、スロット4(s4)～9(s9)のドライブについて"Yes"に設定されていることを確かめます。

```
/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 :

## ストレージガイド

---

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

## 付録B: シリーズ6EコアまたはHybridホストの暗号化 (encryptSedVd.py)

RSAシリーズ6EコアおよびHybridホストには、自動暗号化ドライブ(SED)があります。encryptSedVd.pyスクリプトは、次の処理を行います。

- シリーズ6Eホストの暗号化の設定が正しいことを検証します。
- 暗号化されていないドライブを暗号化します。

**注:** PowerVaultなどの外部ストレージデバイスについては、「REST APIを使用したストレージの構成」の「[REST APIを使用したストレージの構成](#)」で、SEDドライブを暗号化する方法を参照してください。

次のシナリオは、encryptSedVd.pyを使用する理由の例です。

- 物理ホストで暗号化が行われるかどうかを確認する。この場合は、デバイスで暗号化が行われないとスクリプトが判断すると、暗号化する機会が与えられます。
- 暗号化なしでデバイスをセットアップしており、それを暗号化する必要がある。

このスクリプトは、リリース11.4.0.0以降のrsa-sa-toolsディレクトリにあります。次のディレクトリは11.4.0.0用です。

```
rsa-sa-tools-11.4.0.0-xxxx.noarch.rpm
```

次の手順は、スクリプトの使用方法を示しています。

1. rootとしてログインします。
2. rsa-sa-tools RPMベース ディレクトリ、  
cd /opt/rsa/saTools/supportScript/  
に移動します。

3. 次のコマンドを実行します。

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

スクリプトは、ディスクが暗号化されているかどうかを示しています。

- ドライブが暗号化されている場合は、次のメッセージがスクリプトに表示されています。  
No unencrypted RAID virtual drives with SED physical drives found.
- ドライブが暗号化されていない場合は、次の例に示すように、暗号化されていないドライブがスクリプトに示されます。

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

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

4. ドライブが暗号化されておらず、それを暗号化する場合は、次の操作を実行します。

- a. 暗号化するドライブをスペースバーで選択し、Enterキーを押します。

次のプロンプトが表示されます。

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

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

Enter Passphrase:
[]

Verify Passphrase:
[]

Key ID (optional):
[]
```

- b. [パスワードを入力]テキスト ボックスに<passphrase>(nFreDaW\$792など)を入力し、Tabキーを押します。
- c. [Verify Passphrase]テキスト ボックスで、検証のためにパスワードを再入力します。
- d. [Key ID (optional)]テキスト ボックスに、256文字未満のセキュリティキーのID文字列を入力します。セキュリティキーを使用しない場合は、空白のまま、Enterキーを押します。次のプロンプトが表示されます。

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

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

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

- e. [<Y>]を選択し、Enterキーを押して、パスワードの追加を確定します。
- f. 次のコマンド文字列を送信して、SEDドライブが暗号化されていることを確認します。  
/opt/MegaRAID/percccli/percccli64 /c0 show more  
以下の情報が表示されます。4台のSEDドライブのすべてが暗号化されていることがわかります  
(つまり、[SED]列で各ドライブにYが表示されます)。

```
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

```



**注:** ドライブがSED対応で、安全である場合は、[SED Enabled] および[Secured] ラベルの値が [Yes] に設定されます。  
コントローラ0とエンクロージャ247のドライブを確認するには、次のコマンドを使用します。  
`/opt/MegaRAID/percccli/percccli64 /c1 /e247/sall show all | egrep -i '(Policies/Settings|SED Capable|Secured|SED Enabled)'`

各percccliコマンドの詳細情報は、「Dell PowerEdge RAID Controller CLI Reference Guide」([http://14u-00.jinr.ru/pub/misc/h-w/LSI/dell-sas-hba-12gbps\\_reference-guide\\_en-us.pdf](http://14u-00.jinr.ru/pub/misc/h-w/LSI/dell-sas-hba-12gbps_reference-guide_en-us.pdf))に記載されています。

## 構成済みのドライブグループでのSEDの有効化

構成された仮想ドライブは、SEDへの対応が可能ですが、SED対応にはなっていません。

PERC H840アダプター(外部ストレージ)を使用して、仮想ドライブまたはドライブグループを有効にするには、以下の操作を実行します。

1. アプライアンスにSSHで接続し、以下のスクリプトを実行して仮想ドライブ(外部ストレージ上の)を暗号化します。

**注:** encryptSedVd.pyスクリプトでSED機能が有効になるのは、PERC H840アダプター(外部ストレージ)の仮想ドライブまたはドライブグループのみであり、PERC H740 Miniでは有効になりません。PERC H740 MiniでSEDを有効にするには、「[仮想ドライブ/ドライブグループの有効化: PERC H740 \(Mini\) アダプタ\(内部ストレージ\)](#)」を参照してください。

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

2. 仮想ドライブを選択し、Enterキーを押します。  
パスフレーズ画面が表示されます。
3. パスフレーズを入力して、Enterキーを押します。  
以下に例を示します。

Passphrase: **Netwitness1!**

**keyID: netwitness**

```
Please enter a passphrase for the PERC H840 Adapter security key, minimum length 8 characters, maximum 32
The passphrase must contain a mix of lowercase, uppercase, numeric and non-alphanumeric characters
Optionally enter a key identifier, a default id will be created if not specified

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

Enter Passphrase:
Netwitness!

Verify Passphrase:
Netwitness!

Key ID (optional):
netwitness
```

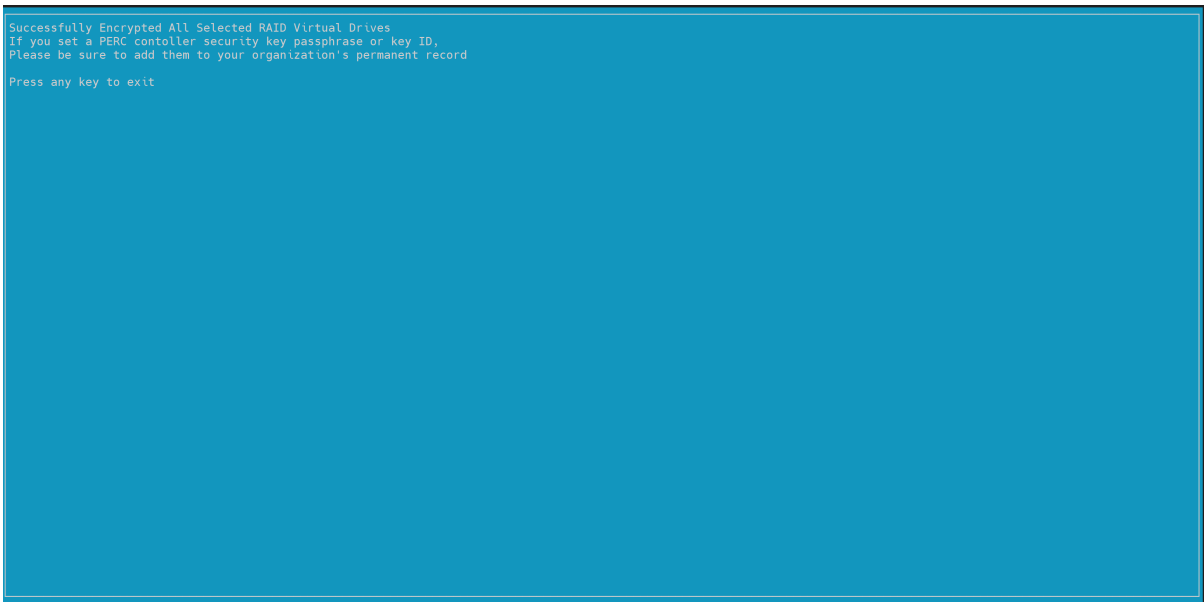
**4. メッセージを確認し、Enterキーを押して保存します。**

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

Entered Passphrase('Quoted'): 'Netwitness!'
Entered KeyId('Quoted'): 'netwitness'

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

5. 任意のキーを押して終了します。



6. ドライブがSED対応で、安全であることを確認するには、次のコマンドを実行し、[SED Enabled]と[Secured]に[Yes]が返されることを確認します。

```
/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

## ストレージガイド

---

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

## 仮想ドライブドライブグループの有効化: PERC H740( Mini) アダプタ( 内部ストレージ)

percli64ユーティリティを使用して、オンボードSED対応ドライブ( スロット4~9、合計6台のドライブ) から作成された仮想ドライブまたはドライブグループでSED機能を有効にすることができます。  
/opt/rsa/saTools/supportScript/encryptSedVd.pyを使用して、PERC H740( Mini) アダプタ上の仮想ドライブでセキュリティを有効にすることはできません。

1. アプライアンスにSSHで接続し、PERC H740( ミニ) アダプターでセキュリティを有効にします。このアダプターのコントローラ番号は0です。PERC H840アダプタは、1として表示されます。

アプライアンス上のすべてのコントローラを一覧表示するには、次のコマンドを実行します。

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

最初の列( Ctl) には、アプライアンス上のコントローラインデックスが一覧表示されます。この場合、コントローラ0はPERC H740 Miniに、コントローラ1はPERC H840アダプタに対応します。列[ DG] および[ VD] には、コントローラ上の仮想ドライブおよびドライブグループが表示されます。

**PERC H740( Mini) アダプタ( コントローラ0など) でセキュリティを有効にするには、次のコマンドを実行します。**

```
/opt/MegaRAID/perccli/perccli64 /c0 set securitykey='<String>'!
keyid='<String>'
```

例:

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

- 2.

'Netwitness1' is the securityKey and 'netwitness' is ID.

KeyとkeyIDの両方を確実にメモしておきます。

3. セキュリティを有効にするSED対応可能ドライブに対応する、正しいドライブグループ( DG) または仮想ドライブ( VD) を特定します。

```
/opt/MegaRAID/perccli/perccli64 /c0 /vall show | egrep -A5 'DG/VD'
```

最初の2列と最後の列をチェックして、SED対応可能である6台のSED対応ドライブに対応する、正しいドライブグループ/仮想ドライブを特定します。シリーズ6アプライアンスでは、RAID6タイプを使用するDGまたはVDは1つだけです。[Name]列を使用して、VDまたはDGを特定することができます。この場合、DGまたはVDは2です。[タイプ]、[名前]、[サイズ]列(これらは、上記でVDを作成したときに定義されています)の組み合わせを使用しています。

4. **decodersmall**ボリュームグループで、6台のSED対応可能ドライブから作成されたディスクグループのセキュリティを有効にするには、次のコマンドを実行します。

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

5. コントローラ0で使用されるエンクロージャID( EID) を取得します。この場合は64です。

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

6. ドライブまたはドライブグループがSED対応であり、安全であることを確認するには、以下のコマンドを実行し、[SED Capable]、[Secured]、[SED Enabled]の各フラグがスロット4(s4)～9(s9)でドライブについて[Yes]に設定されていることを確認します。

```
/opt/MegaRAID/perccli/perccli64 /c0 /e64/sall show all | egrep -i '
```

```
(Policies/Settings |SED Capable|Secured|SED Enabled)'
```

ドライブの/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

## PowerVault( PERC 840) で構成された仮想ドライブドライブグループでのSEDの有効化

### 仮想ドライブドライブグループの有効化 : PERC H840アダプタ

注 : 「[物理ストレージの準備](#)」の「PowerVaultのブロック デバイスの構成」セクションで作成した仮想ディスクはSED対応可能ですが、SED対応になっていません。

1. 有効にするには、アプライアンスにSSHで接続し、以下のスクリプトを実行してVD(外部ストレージ上の)を暗号化します。

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

**注:** encryptSedVd.pyスクリプトでSED機能が有効になるのは、PERC H840アダプタ(外部ストレージ)の仮想ドライブまたはドライブグループのみであり、PERC H740 Miniでは有効になりません。PERC H740 MiniでSEDを有効にするには、「仮想ドライブドライブグループの有効化: PERC H740(Mini) アダプタ(内部ストレージ)」を参照してください。

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

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

 ID VD DG RAID SIZE HBA
 (X) 1 0 0 RAID6 106.9TB PERC H840 Adapter
```

2. 両方の仮想ディスクを選択し、Enterキーを押します。  
パスフレーズ画面が表示されます。

```
Please enter a passphrase for the PERC H840 Adapter security key, minimum length 8 characters, maximum 32
The passphrase must contain a mix of lowercase, uppercase, numeric and non-alphanumeric characters
Optionally enter a key identifier, a default id will be created if not specified

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

Enter Passphrase:

Verify Passphrase:

Key ID (optional):

```

3. パスフレーズを入力し、**Enter**キーを押します。  
以下に例を示します。

Passphrase: **Netwitness!**

keyID: **netwitness**

```
Please enter a passphrase for the PERC H840 Adapter security key, minimum length 8 characters, maximum 32
The passphrase must contain a mix of lowercase, uppercase, numeric and non-alphanumeric characters
Optionally enter a key identifier, a default id will be created if not specified

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

Enter Passphrase:
Netwitness!

Verify Passphrase:
Netwitness!

Key ID (optional):
netwitness
```

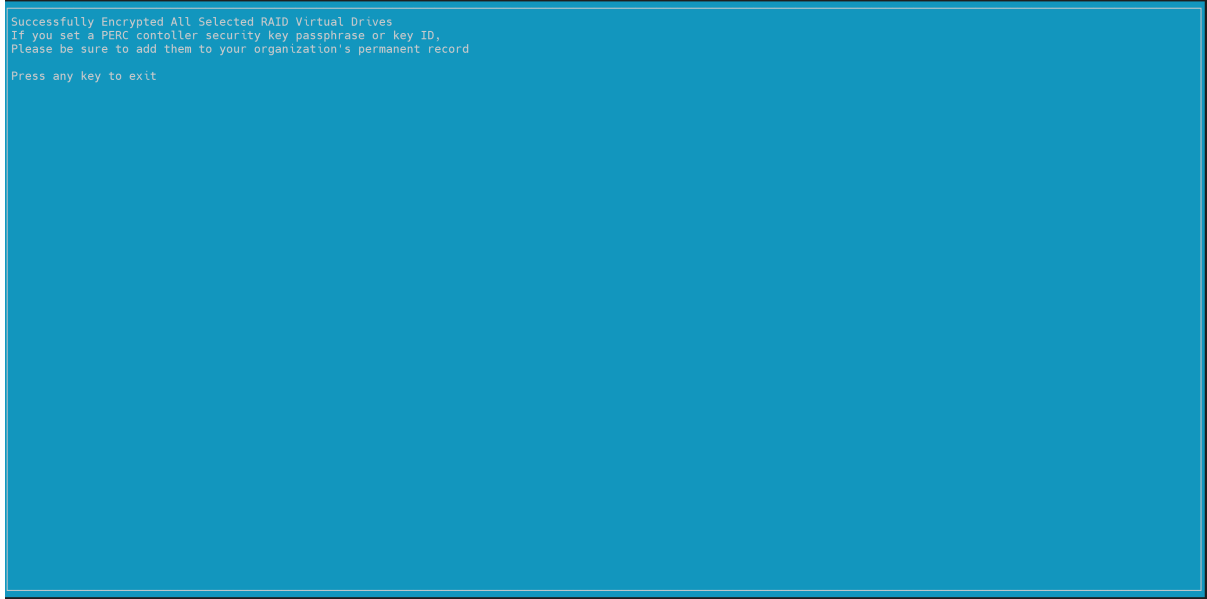
4. メッセージを確認し、**Enter**キーを押して保存します。

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

Entered Passphrase('Quoted'): 'Netwitness!'
Entered KeyId('Quoted'): 'netwitness'

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

## 5. 任意のキーを押して終了します。



ドライブがSED対応であり、安全であることを確認するには、以下のコマンドを実行し、SED EnabledとSecuredにYesが返されることを確かめます。

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

ドライブ /c1/e247/s0 Policies/Settings:

```
SED Capable = Yes
```

## 6.

```
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

## SEDドライブと非SEDドライブが混在するホスト上のSED対応ドライブグループでのセキュリティの有効化

SEDドライブと非SEDドライブの両方がアプライアンスに混在している場合は、`encryptSedVd.py`がSED対応仮想ドライブの識別に失敗する可能性があります。SED対応と非SED対応の両方の仮想ドライブがホストに存在する場合は、以下の手順が適用されます。

1. アプライアンスにSSHで接続し、PERC H740(ミニ)アダプターでセキュリティを有効にします。このアダプターのコントローラー番号は0です。PERC H840アダプターは1として表示されます。アプライアンス上のすべてのコントローラを一覧表示するには、次のコマンドを実行します。

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

最初の列(Ctl)には、アプライアンス上のコントローラインデックスが一覧表示されます。この場合、コントローラ"0"は"PERC H740 Mini"に、コントローラ"1"は"PERC H840アダプター"に対応します。列[DG]および[VD]には、コントローラ上のドライブグループおよび仮想ドライブが表示されます。

```
[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. "PERC H740(ミニ)アダプター"、つまり、コントローラ"0"のセキュリティを有効にするには、次のコマンドを実行します。

```
/opt/MegaRAID/perccli/perccli64 /c0 set securitykey='<SOME_STRING_VALUE>'!'
keyid='< SOME_STRING_VALUE >'
```

例:

```
/opt/MegaRAID/perccli/perccli64 /c0 set securitykey='Netwitness!' keyid=1
```

"Netwitness!"はsecurityKeyで、"1"はIDです。KeyとkeyIDの両方を安全に保管します。

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

Controller Properties :
=====

Ctrl Method Result

0 set Key Success
```

3. セキュリティを有効にしようとしているSED対応可能ドライブに対応する、正しいドライブグループ(DG)または仮想ドライブ(VD)を特定します。

```
/opt/MegaRAID/perccli/perccli64 /c0 /vall show | egrep -A5 'DG/VD'
```

最初の2列と最後の列を参照して、6つのSED対応ドライブに対応する正しいドライブグループ(DG)/仮想ドライブ(VD)を特定します。シリーズ6アプライアンスでは、RAID6を使用するDG/VDは1つだけです。[Name]列を使用して、VDまたはDGを特定することができます。この場合、DG/VDは"2"です。[タイプ]、[名前]、[サイズ]列(これらは、上記でVDを作成したときにユーザーによって定義されています)の組み合わせを使用しています。

```
[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. ディスクグループ(6台のSED対応ドライブから作成)のセキュリティを有効にするには、次のコマンドを実行します。

```
/opt/MegaRAID/percccli/percccli64 /c0 /d2 set security=on
```

```
[root@116Decoder perccli]# /opt/MegaRAID/percccli/percccli64 /c0 /d2 set security=on
Controller = 0
Status = Success
Description = Success
```

5. コントローラーで"0"で使われるエンクロージャID (EID) を取得します。この場合は"64"です。

```
/opt/MegaRAID/percccli/percccli64 /c0 /eall show
```

```
[root@116Decoder perccli]# /opt/MegaRAID/percccli/percccli64 /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]#
```

ドライブ/ドライブグループ(DG)がSED対応かつ安全であることを確認するために、以下のコマンドを実行し、[SED Capable]、[Secured]、[SED Enabled]の各フラグが、スロット4(s4)～9(s9)のドライブについて[Yes]に設定されていることを確かめます。

```
/opt/MegaRAID/percccli/percccli64 /c0 /e64/sall show all | egrep -i '
(Policies/Settings |SED Capable|Secured|SED Enabled)'
```

```
Drive /c0/e64/s0 Policies/Settings :
```

- 6.

```
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

## 付録C:トラブルシューティング

このセクションでは、REST APIを使用して、さまざまなストレージ タスクを解決する手順を説明しています。

### REST APIを使用した、デコーダーに接続された事前構成済みDACの再構成

このシナリオでは、別のツールを使用して構成されたDACを、REST APIを使用して再構成し、(不要になった場合、または別のストレージ デバイスにバックアップされている場合に) 既存のデータをすべて消去する方法を説明します。

次の情報は、REST APIを使用してストレージ デバイスを再構成する前の、ホストとストレージ ハードウェアの状態を説明しています。

追加されたときにDACには古いデータがありました。また、DACは(REST APIを使用せずに) 構成されていました。このために、REST APIが`raidNew`コマンドを実行できず、"Physical disk does not have appropriate attributes" というエラー メッセージが返されました。

次のステップでは、シナリオとその解決策について説明します。

1. Decoder Linuxコンソールから(またはDecoderにSSH接続して)、次のコマンド文字列を送信しました。

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

`perccli`コマンドの詳細情報は、『**Dell PowerEdge RAID Controller CLI Reference Guide**』([https://topics-cdn.dell.com/pdf/dell-sas-hba-12gbps\\_reference-guide\\_en-us.pdf](https://topics-cdn.dell.com/pdf/dell-sas-hba-12gbps_reference-guide_en-us.pdf))に記載されていません。

これにより、すべての外部構成がコントローラー2から削除され、すべてのデータがDACから消去されました。

2. DACにパーティションを作成しようとしたのですが、その情報はすでにDACで定義されているため、`partNew`コマンドは失敗しました。`partNew`には、使用可能なデバイスを1つ使用する必要があると表示されましたが、`devList`には、それが使用中であると表示されました。
3. パーティションが定義されていると仮定して、ストレージ デバイスを割り当てようとしたのですが、DACがマウントされていないため、割り当てることができませんでした。
4. コマンド ラインからDACをマウントしようとしたのですが、"mount failed: structure needs to be cleaned"というエラー メッセージを受信しました。
5. DACに保持する必要のあるデータがなかったため、構造をクリーン アップするために次のコマンド文字列を送信しました。

```
mkfs.xfs -f /dev/decoder0/packetdb
mkfs.xfs -f /dev/decoder1/packetdb
```
6. `/var/netwitness/decoder`内の適切な場所にデバイスをマウントしました。
7. 「[REST APIを使用したストレージの構成](#)」の説明に従って、該当する残りの手順を完了して、DACを再構成しました。

## 付録D: ストレージ構成のシナリオの例

この付録では、以下に示す、2つの非暗号化15ドライブDAC外部ストレージデバイスでのストレージの構成方法の例について説明します。

- [Archiverのストレージの構成](#)
- [Network\(Packet\) Decoderのストレージの構成](#)
- [Network Concentratorのストレージの構成](#)
- [Log Decoder Hybridのストレージの構成](#)

### Archiverのストレージの構成

次のシナリオでは、Archiverの物理ホストのために、1つの非暗号化15ドライブDAC上にストレージを構成します。

1. `raidList`コマンドを実行します。
  - a. コントローラ番号、エンクロージャ番号、使用中、ドライブ、デバイスの情報を記録します。  
次の情報が表示されます。  
In Use: FALSE  
Devices: <empty>
  - b. ドライブ数、サイズ、ベンダーを確認します。  
次の例は、RAIDアレイを作成する前に表示される内容を示しています。

2. 記録したコントローラ番号とエンクロージャ番号を使用して、以下のパラメータを指定して`raidNew`コマンドを実行します。

```
controller=1 enclosure=0 scheme=archiver commit=1
```

次の例は、RAIDアレイを作成した後に表示される内容を示しています。

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

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

Message Help

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

Response Output

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

- raidListコマンドを実行して、新しいRAIDアレイを検証します。  
次の情報が表示されます。

In Use: TRUE

Devices: <device>(sdcなど)

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

raidList Parameters

Message Help

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

Response Output

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

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

- 次のパラメーターを指定してpartNewコマンドを実行し、etc/fstabファイル内にパーティションとマウントポイントを作成します。  
name=<device>(for example, sdc) service=archiver volume=archiver commit=1
- 次のパラメーターを指定してsrvAllocコマンドを実行して、archiverサービスにスペースを割り当てます。これにより、archiverサービス構成にストレージが追加され、実行されるたびにこのサービスが再起動されます。  
service=archiver volume=archiver0 commit=1

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

```

srvAlloc Parameters service=archiver volume=archiver0 commit=1
Message Help
service - <string, {enum-one:archiver|concentrator|decoder|logdecoder}> service that will use storage
volume - <string, {enum-one:archiver0|netwitness_vg00}> volume group name
commit - <bool, optional> commit changes

```

Change Service | NWHOST2100 - Archiver | System

Start Aggregation Stop Aggregation Host Tasks Shutdown Service

Archiver Service Information

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

6. [データ保存]で[Hotストレージ]を確認してください。

Change Service | NWHOST2100 - Archiver | Config

General Data Retention Files Appliance Service Configuration

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

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

Total Hot Storage 47.29 TB

Total Warm Storage Not Configured

Cold Storage Not Configured

1 Mount Point

**Collections**

+ - [edit] [refresh]

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

**Retention Rules**

+ - [edit] [refresh] | Move Up Move Down | Apply Revert

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

7. 次のArchiverサービスを再構成して、「[タスク5:\(オプション\)ストレージ構成を10G収集用に再構成する](#)」の説明に従って、すべての空き領域を検出し、利用します。

## Network( Packet) Decoderのストレージの構成

次のシナリオでは、10G収集の物理ホストのNetwork Decoderのために、2つの非暗号化15ドライブDAC上にストレージを構成します。

1. raidListコマンドを実行します。
  - a. コントローラ番号、エンクロージャ番号、使用中、ドライブ、デバイスの情報を記録します。  
次の情報が表示されます。

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

- b. ドライブ数、サイズ、ベンダーを確認します。  
次の例は、RAIDアレイを作成する前に表示される内容を示しています。

Properties for NWHOST2100 -

raidList Parameters

Message Help

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

---

Response Output

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

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

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

2. 記録したコントローラ番号とエンクロージャ番号を使用して、次のパラメータでraidNewコマンドを実行します。

- 最初のエンクロージャのパラメータ:  
controller=1 enclosure=0 scheme=decoder commit=1

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

Message Help

parameters:  
 controller - <int32, {enum-one:0,1}> Controller the shelf is attached to  
 enclosure - <uint32, {enum-one:32,0,2}> Enclosure number of the shelf to clear  
 scheme - <string, {enum-one:decoder|logdecoder|concentrator|archiver|network-hybrid|log-hybrid}> Type of RAID volumes to allocate

Response Output

```

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

/opt/MegaRAID/perccli/perccli64 /c1 add vd r5 drives=0:3,0:4,0:5,0:6,0:7,0:8,0:9,0:10,0:11,0:12,0:13,0:14 wb ra cached Strip=128
Controller = 1
Status = Success
Description = Add VD Succeeded

```



- 2番目のエンクロージャのパラメータ:

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

raidList ▼ Parameters

Message Help

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

Response Output

```
Devices: sda
 sdb

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

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

3. raidListコマンドを使用して、In Use: TRUEを確認できるように、エンクロージャのブロック デバイスを表示します。

4. Network DecoderにSSH接続し、lsblkコマンドを使用して、**decodersmall**のサイズを確認します。

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

**注:** RAID構成で、10G収集のデコーダを使用する場合は、エンクロージャとパフォーマンスの両方の理由により、**decoder**を使用します。10G収集に**decoder**を使用しない場合は、**decoder**と**archiver**をエンクロージャに使用してストレージを最大化します。これは、**archiver**構成では、2番目のエンクロージャが1つのRAIDであるためです。

5. **partNew**コマンドを実行して、次のパラメーターで**decodersmall**パーティションを最初に作成します ( **decoder dir**、**index**、**metadb**、**sessiondb** ) ( 最初のエンクロージャ、SDC、SDD )。
- ```
name=sdC service=decoder volume=decodersmall commit=1
```

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

Message Help

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

Response Output

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

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

- partNewコマンドを実行して、次のパラメータでdecoderボリューム(packetdb) (最初のエンクロージャ、SDC、SDD)を作成します。

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

```
partNew Parameters name=sdd service=decoder volume=decoder commit=1
Message Help
name - <string, {enum-one:sd, sdd, sde, sdf}> block device name
service - <string, {enum-one:archiver | concentrator | decoder | logdecoder}> service that will use storage
volume - <string, optional, {enum-one:index | concentrator | decodersmall | decoder | logdecodersmall | logdecoder | archiver}> volume to create
commit - <bool, optional> commit changes
```

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

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

以下の例では、以下のパーティションがSDC、SDD(エンクロージャ0)用に作成されています。

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

この時点で、2番目のDACエンクロージャを追加します。

- partNewコマンドを実行して、次のパラメータでdecodersmallパーティション(2番目のエンクロージャ、SDE、SDF)を最初に作成します。

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

Properties for 11mtlnxnwpacket01 - Decoder (DECODER) /deviceappliance/appliance.

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

Message Help

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

Response Output

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

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

- partNewコマンドを実行して、以下のパラメータでpacketdb decoderボリューム(2番目のエンクロージャ、SDE、SDF)を作成します。

```
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|decodersmall|decoder|logdecodersmall|logdecoder|archiver}> volume to create
commit - <bool, optional> commit changes
```

Response Output

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



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

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

9. 次のパラメータでsrvAllocコマンドを実行して、サービス構成設定にストレージ情報を追加します。

- service=decoder volume=decodersmall commit=1
- service=decoder volume=decodersmall0 commit=1
- service=decoder volume=decoder commit=1
- service=decoder volume=decoder0 commit=1

srvAlloc Parameters service=decoder commit=1 volume=decoder0

Message Help

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

Response Output

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

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

- 次のNetwork Decoderサービスとそのデータベースを再構成して、「[タスク5:\(オプション\)ストレージ構成を10G収集用に再構成する](#)」の説明に従って、すべての空き領域を検出し、利用します。

Network Concentratorのストレージの構成

次のシナリオでは、Network Concentratorの物理ホストのために、1つの非暗号化15ドライブDAC上にストレージを構成します。

1. `raidList`コマンドを実行します。

Message Help

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

Response Output

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

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

2. 次のパラメータを指定してraidNewコマンドを実行します。

```
controller=1 enclosure=6 scheme=concentrator
```

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

Message Help

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

Response Output

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

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

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

3. partNewコマンドを実行して、次のパラメータでconcentratorパーティションを最初に作成します。concentratorボリュームは、indexボリュームの前に作成する必要があります。そうしない場合は、エラーが発生します。

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

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

Message Help

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

Response Output

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

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

4. 次のパラメータでpartNewコマンドを実行して、SSDにインデックスを作成します。

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

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

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

5. 次のパラメータを指定して `srvAlloc` コマンドを実行します。

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

Property	Value
index.dir	/var/netwitness/concentrator/index==881.87 GB
index.dir.cold	
index.dir.warm	
index.slices.open	42
page.compression	huffybrid
reindex.enable	true
save.session.count	auto

6. 次のパラメータを指定して `srvAlloc` コマンドを実行します。

`service=concentrator volume=concentrator commit=1`

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

Message Help

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

Response Output

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

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

7. 次のNetwork Concentratorサービスとそのデータベースを再構成して、「[タスク5:\(オプション\)ストレージ構成を10G収集用に再構成する](#)」の説明に従って、すべての空き領域を検出し、利用します。

Log Decoder Hybridのストレージの構成

次のシナリオでは、Log Decoder Hybridの物理ホストのために、1つの非暗号化15ドライブDAC上にストレージを構成します。

1. `raidList`コマンドを実行します。

raidList

Message Help

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

Response Output

Controller 0, Enclosure 32

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

Devices: sda

sdb
 sdc
 sdd
 sde

Controller 1, Enclosure 31

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

2. 次のパラメータを指定して`raidNew`コマンドを実行します。

```
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. 次のパラメータを指定してpartNewコマンドを実行します。

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

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

Message Help

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

Response Output

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

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

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

partNew 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. 次のパラメータを指定して `srvAlloc` コマンドを実行します。

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

Parameters

Message Help

```

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

Response Output

```

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

Property	Value
/database/config	NWHOST1700 - Concentrator
hash.algorithm	none
hash.databases	session,meta
hash.dir	
manifest.dir	
meta.compression	none
meta.compression.level	0
meta.dir	/var/netwitness/concentrator/metadb=14.08 TB;/var/netwitness/concentrator/metadb0==20.17 TB
meta.dir.cold	
meta.dir.warm	
meta.file.size	auto
meta.files	auto
meta.free.space.min	132 GB
meta.index.fidelity	4
meta.integrity.flush	sync
meta.write.block.size	64 KB
session.dir	/var/netwitness/concentrator/sessiondb=1.41 TB;/var/netwitness/concentrator/sessiondb0==569.72 GB

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

`packet.dir` `/var/netwitness/logdecoder/packetdb=15.37 TB;/var/netwitness/logdecoder/packetdb0==24.19 TB`

5. 次のLog Decoderサービスとそのデータベースを再構成して、「[タスク5:\(オプション\)ストレージ構成を10G収集用に再構成する](#)」の説明に従って、すべての空き領域を検出し、利用します。

改訂履歴

改訂	日付	説明
1.0	2021年5月	11.6リリース アップデート