



Proxmoxバックアップドキュメント

リリース4.0.11-2

Proxmoxサポートチーム

2025年8月06日(水曜日)

目次

1 はじめに	3
1.1 Proxmox Backup Serverとは	3
1.2 Architecture	3
1.3 主な機能	3
1.4 データ・バックアップの理由	4
1.5 ソフトウェア・スタック	5
1.6 ヘルプの入手	5
1.6.1 エンタープライズ・サポート	5
1.6.2 コミュニティ・サポート・フォーラム	5
1.6.3 メーリングリスト	5
1.6.4 バグトラッカー	5
1.7 ライセンス	6
1.8 歴史	6
2 インストール	7
2.1 システム要件	7
2.1.1 評価用サーバーの最小要件	7
2.1.2 推奨サーバー・システム要件	7
2.1.3 ウェブ・インターフェイスにアクセスするためにサポートされているウェブ・ブラウザ	8
2.2 Installation Medium	8
2.2.1 インストール媒体の準備	8
2.2.2 Prepare a USB Flash Drive as Installation Medium	8
2.2.3 GNU/Linux 用インストラクション	8
2.2.4 macOS 用インストラクション	9
2.2.5 Windows	10
2.2.6 インストールメディアを使用する	10
2.3 サーバーのインストール	10
2.3.1 インストーラを使用して Proxmox Backup Server をインストールします。	10
2.3.2 Proxmox Backup Serverを無人インストールします。	18
2.3.3 Install Proxmox Backup Server on Debian	19
2.3.4 Proxmox VE に Proxmox Backup Server をインストールします。	19
2.4 クライアントのインストール	20
2.4.1 ProxmoxバックアップクライアントのDebianへのインストール	20
2.4.2 静的にリンクされた Proxmox Backup Client のインストール	20
2.5 Debian Package Repositories	20
2.5.1 Repository Formats	20
2.5.2 Debian ベースリポジトリ	21
2.5.3 Proxmox Backup Enterprise リポジトリ	21
2.5.4 Proxmox Backup サブスクリプションなしリポジトリ	22
2.5.5 Proxmox バックアップテストリポジトリ	22
2.5.6 Proxmox Backup クライアント専用リポジトリ	22
2.5.7 SecureApt	23
2.5.8 HTTP プロキシを介したリポジトリアクセス	24

3 用語解説	25
3.1 バックアップ・コンテンツ	25
3.1.1 Image Archives: <name>.img	25
3.1.2 ファイルアーカイブ: <name>.pxar	25
3.1.3 バイナリ・データ (BLOB)	25
3.1.4 カタログファイル: catalog.pcat1	25
3.1.5 マニフェスト: index.json	25
3.2 バックアップ・ネームスペース	26
3.3 バックアップタイプ	26
3.4 バックアップID	26
3.5 バックアップ時間	26
3.6 バックアップ・グループ	26
3.7 バックアップスナップショット	26
4 グラフィカル ユーザー インターフェイス	27
4.1 特徴	27
4.2 ログイン	27
4.2.1 Consent Banner	28
4.3 GUI Overview	28
4.4 サイドバー	28
4.4.1 ダッシュボード	29
4.4.2 コンフィグレーション	29
4.4.3 Administration	29
4.4.4 テープ・バックアップ	31
4.4.5 データストア	32
5 バックアップストレージ	33
5.1 ディスク管理	33
5.2 データストア	35
5.2.1 データストアの構成	36
5.2.2 バックアップ名前空間	41
5.2.3 オプション	42
5.3 ランサムウェアの保護と復旧	44
5.3.1 Built-in Protection	44
5.3.2 The 3-2-1 Rule with Proxmox Backup Server	44
5.3.3 制限付きユーザーとアクセス管理	44
5.3.4 ランサムウェア検出	45
5.3.5 一般的な予防方法とベストプラクティス	45
6 ユーザー管理	47
6.1 User Configuration	47
6.2 APIトークン	49
6.3 Access Control	50
6.3.1 Privileges	50
6.3.2 アクセス・ロール	51
6.3.3 オブジェクトとパス	52
6.3.4 構成と管理	53
6.3.5 APIトークン権限	54
6.3.6 有効なパーミッション	54
6.4 二要素認証	54
6.4.1 Introduction	54
6.4.2 Available Second Factors	55
6.4.3 Setup	55
6.4.4 TFA and Automated Access	56
6.4.5 二要素認証の制限とロックアウト	56
6.5 認証領域	56
6.5.1 Linux PAM	56
6.5.2 Proxmox Backup 認証サーバ	57
6.5.3 LDAP	57

6.5.4 アクティブディレクトリ	57
7 バックアップクライアントの使用	59
7.1 バックアップ・リポジトリの場所.....	59
7.2 静的にリンクされたバックアップクライアント	60
7.3 環境変数.....	60
7.4 システムとサービスの資格情報.....	61
7.5 出力フォーマット	61
7.6 バックアップの作成	62
7.6.1 バックアップからのファイル/ディレクトリの除外	63
7.6.2 変更検出モード	64
7.7 暗号化.....	64
7.7.1 マスター・キーの使用による暗号化キーの保存と復元	65
7.8 データの復元	66
7.8.1 インタラクティブ・リストア	67
7.8.2 FUSE によるアーカイブのマウント	68
7.9 ログインとログアウト	68
7.10 バックアップグループの所有者の変更	68
7.11 バックアップの刈り込みと削除	68
7.12 ガベージコレクション	70
7.13 ベンチマーク	71
8 Proxmox VEの統合	73
8.1 Proxmox VE ウェブインターフェースの使用	73
8.2 Proxmox VE コマンドラインの使用	73
9 pxarコマンドラインツール	75
9.1 アーカイブの作成	75
9.2 アーカイブの抽出	76
9.3 アーカイブの内容の一覧表示	76
9.4 アーカイブのマウント	76
10 テープバックアップ	77
10.1 テープ技術入門	78
10.2 対応ハードウェア	78
10.2.1 ドライブ性能	78
10.2.2 LTO-9+ に関する考慮事項	79
10.3 用語解説	79
10.4 テープクイックスタート	80
10.5 構成	80
10.5.1 テープチェンジャー	80
10.5.2 テープ・ドライブ	83
10.5.3 メディア・プール	85
10.5.4 テープ・バックアップ・ジョブ	88
10.6 管理	90
10.6.1 ラベルテープ	90
10.6.2 テープバックアップの実行	91
10.6.3 テープからのリストア	91

10.6.4 インベントリの更新	92
10.6.5 カタログのリストア	92
10.6.6 暗号化キーの管理	93
10.6.7 テープ・クリーニング	94
10.7 WORM テープ	94
10.8 セットアップ例	95
10.8.1 単一継続メディアセット	95
10.8.2 平日スキーム	95
10.8.3 異なるポリシーの複数プール	95
10.9 ディザスタリカバリー	96
10.9.1 データストアの設定	96
10.9.2 テープ・ドライブのセットアップ	96
10.9.3 テープからのデータのリストア	97
11 リモートと同期の管理	99
11.1 リモート	99
11.2 同期ジョブ	100
11.2.1 名前空間サポート	101
11.2.2 帯域幅の制限	102
11.2.3 同期方向プッシュ	103
12 メンテナスタスク	105
12.1 剪定	105
12.1.1 剪定シミュレーター	106
12.1.2 剪定ジョブ	106
12.1.3 手動剪定	107
12.1.4 保持設定の例	108
12.2 ガベージコレクション	108
12.2.1 GCの背景	108
12.2.2 GCの手動起動	109
12.2.3 スケジュールされた GC	109
12.3 検証	110
12.4 通知	110
12.5 メンテナスマード	110
13 ホストシステム管理	111
13.1 Linux上のZFS	111
13.1.1 ハードウェア	112
13.1.2 ZFS の管理	112
13.2 ホストブートローダー	117
13.2.1 インストーラが使用するパーティショニングスキーム	117
13.2.2 proxmox-boot-toolによるESPの内容の同期	117
13.2.3 使用されるブートローダの決定	120
13.2.4 グラブ	121
13.2.5 システムブート	122
13.2.6 カーネルコマンドラインの編集	122
13.2.7 次のブートのためのカーネルバージョンの上書き	122
13.2.8 セキュアブート	123

13.3	証明書の管理	126
13.3.1	API および SMTP 用証明書	126
13.3.2	カスタム証明書のアップロード	126
13.3.3	Let's Encrypt (ACME) を介した信頼できる証明書	126
13.3.4	ACME HTTP チャレンジプラグイン	128
13.3.5	ACME DNS API チャレンジプラグイン	128
13.3.6	ACME 証明書の自動更新	129
13.3.7	コマンドラインによる証明書の手動変更	129
13.4	外部メトリックサーバ	130
13.4.1	InfluxDB (HTTP) プラグインの設定	130
13.4.2	InfluxDB (UDP) プラグインの設定	131
13.5	サービスデーモン	131
13.5.1	プロックスモックスバックアッププロキシ	131
13.5.2	プロックスモックスバックアップ	131
13.6	コマンドラインツール	131
13.6.1	プロックスモックスバックアップクライアント	131
13.6.2	プロックスモックスバックアップマネージャ	131
13.6.3	プロックスステープ	131
13.6.4	pmt	131
13.6.5	pmtx	131
13.6.6	pxar	132
13.6.7	プロックスモックスファイルリストア	133
13.6.8	proxmox-backup-debug	133
14	ネットワーク管理	135
14.1	交通管制	137
14.2	ネットワークデバイス名の上書き	138
14.2.1	diff -y /etc/network/interfaces /etc/network/interfaces.new	139
14.2.2	proxmox-network-interface-pinning generate	139
14.2.3	proxmox-network-interface-pinning generate --prefix myprefix	139
14.2.4	proxmox-network-interface-pinning generate --interface enp1s0	139
14.2.5	proxmox-network-interface-pinning generate --interface enp1s0 --target-name if42	139
15	通知	141
15.1	概要	141
15.2	通知対象	141
15.2.1	センドメール	141
15.2.2	SMTP	142
15.2.3	ゴティファイ	142
15.2.4	ウェブフック	142
15.3	通知マッチャー	144
15.3.1	カレンダーマッチングルール	144
15.3.2	フィールド照合ルール	144
15.3.3	重大度照合ルール	145
15.4	通知イベント	145
15.5	システムメール転送	146
15.6	アクセス許可	146
15.7	通知モード	146
15.7.1	レガシーセンドメール通知モードの設定	146
15.8	通知テンプレートのオーバーライド	146
16	技術的概要	149

16.1 データストア	149
16.2 スナップショット	149
16.3 チャンク	149
16.3.1 固定サイズのチャンク	150
16.3.2 動的なサイズのチャンク	150
16.3.3 暗号化されたチャンク	150
16.3.4 ファイル・ベース・バックアップの変更検出モード	151
16.4 警告と制限	152
16.4.1 ハッシュ衝突に関する注意事項	152
16.4.2 ファイルベースのバックアップ	153
16.4.3 暗号化されたチャンクの検証	153
16.5 トラブルシューティング	153
16.5.1 Proxmox Backup Serverが稼働していない状態でのリストア	154
17 よくある質問	155
17.1 Proxmox Backup Server (PBS) はどのディストリビューションに基づいていますか?	155
17.2 バックアップソース（クライアント）としてサポートされているプラットフォームは？	155
17.3 Proxmox Backup Serverは32ビットプロセッサで動作しますか？	155
17.4 Proxmox Backup Server のバージョンのサポート期間は？	155
17.5 Proxmox Backup Server を次のポイントリリースにアップグレードする方法を教えてください。	155
17.6 Proxmox Backup Serverを次のメジャーリリースにアップグレードする方法は？	156
17.7 データストアを別の場所にコピーまたは同期できますか？	156
17.8 Proxmox Backup Serverはバックアップアーカイブのデータ整合性を検証できますか？	156
17.9 リモートサーバーにバックアップする場合、リモートサーバーを信頼する必要がありますか？	156
17.10 バックアップは増分／重複／フルですか？	157
A コマンド構文	159
A.1 プロックスモックスバックアップクライアント	159
A.1.1 カタログシェルコマンド	171
A.2 プロックスモックスバックアップマネージャ	173
A.3 プロックスステープ	222
A.4 pmt	237
A.5 pmxt	240
A.6 pxar	241
A.7 プロックスモックスファイルリストア	243
A.8 プロックスモックスバックアップデバッグ	245
B 設定ファイル	249
B.1 acl.cfg	249
B.1.1 ファイル形式	249
B.1.2 役割	249
B.2 datastore.cfg	250
B.2.1 ファイル形式	250
B.2.2 オプション	250
B.3 ドメイン.cfg	252
B.3.1 ファイル形式	252
B.3.2 オプション	252
B.4 メディアプール	256
B.4.1 ファイル形式	256
B.4.2 オプション	256
B.5 node.cfg	256
B.5.1 オプション	256

B.6	通知.cfg	257
B.6.1	ファイル形式	257
B.6.2	オプション	257
B.7	通知非公開.cfg	260
B.7.1	ファイル形式	260
B.7.2	オプション	260
B.8	prune.cfg	260
B.8.1	ファイル形式	260
B.8.2	オプション	261
B.9	tape.cfg	261
B.9.1	ファイル・フォーマット	261
B.9.2	オプション	262
B.10	tape-job.cfg	262
B.10.1	ファイル形式	262
B.10.2	オプション	263
B.11	user.cfg	263
B.11.1	ファイル形式	263
B.11.2	オプション	264
B.12	リモート.cfg	264
B.12.1	ファイル形式	264
B.12.2	オプション	265
B.13	sync.cfg	265
B.13.1	ファイル形式	265
B.13.2	オプション	265
B.14	検証.cfg	266
B.14.1	ファイル形式	266
B.14.2	オプション	266
C	ファイル形式	269
C.1	Proxmox ファイルアーカイブ形式 (.pxar)	269
C.2	Proxmox ファイルアーカイブ形式 - メタ (.mpxar)	269
C.3	Proxmox ファイルアーカイブ形式 - ペイロード (.ppxar)	270
C.4	データblob形式 (.blob)	270
C.5	固定インデックス形式 (.fidx)	271

C.6	動的インデックス形式 (.didx)	.271
D	バックアッププロトコル	273
D.1.1	プロブのアップロード	.273
D.1.2	チャンクのアップロード	.273
D.1.3	固定インデックスのアップロード	.273
D.1.4	動的インデックスのアップロード	.274
D.1.5	Finish Backup	.274
D.2	リストア/リーダープロトコルAPI	.274
D.2.1	プロブのダウンロード	.274
D.2.2	ダウンロード・チャンク	.274
D.2.3	インデックスファイルのダウンロード	.274
E	カレンダーイベント	275
E.1	Introduction and Format	.275
E.2	systemdとの相違点	.276
E.3	スケジューリングに関する注意事項	.276
F	マークダウン入門	277
F.1	Markdown Basics	.277
F.1.1	見出し	.277
F.1.2	Emphasis	.277
F.1.3	リンク	.277
F.1.4	リスト	.277
F.1.5	テーブル	.278
F.1.6	ブロックの引用	.278
F.1.7	コードとスニペット	.278
G	用語集	281
H	GNU フリー・ドキュメンテーション・ライセンス	283
索引		289

著作権 (C) 2019-2025, Proxmox Server Solutions GmbH バージョン 4.0.11 -- 2025

年8月6日水曜日

GNUフリー・ドキュメンテーション・ライセンス、バージョン1.3、またはフリーソフトウェアファウンデーションによって発行されたそれ以降のバージョンの条項の下で、この文書を複製、頒布、改変することを許可します。このライセンスのコピーは「GNU Free Documentation License」と題されたセクションに含まれています。

はじめに

1.1 Proxmox Backup Server とは?

Proxmox Backup Serverは、[仮想マシン](#)、[コンテナ](#)、物理ホストをバックアップできるエンタープライズクラスのクライアントサーババックアップソリューションです。Proxmox Backup ServerはProxmox Virtual Environment プラットフォームに最適化されており、リモートサイト間でもデータを安全にバックアップでき、Webベースのユーザインターフェースで簡単に管理できます。

重複排除、圧縮、認証暗号化（AE）をサポートしています。実装言語として[Rust](#)を使用することで、高性能、低リソース使用、安全で高品質なコードベースを保証します。

Proxmox Backupは、クライアント-サーバ間の通信とバックアップコンテンツの[暗号化](#)の両方に最先端の暗号技術を使用しています。すべてのクライアント-サーバ通信は[TLS](#)を使用し、バックアップデータは送信前にクライアント側で暗号化することができます。

1.2 アーキテクチャ

Proxmox Backup Serverはクライアント-サーバモデルを採用しています。サーバはバックアップデータを保存し、データストアを作成および管理するためのAPIを提供します。APIを使用すると、ディスクやその他のサーバ側のリソースを管理することも可能です。

`proxmox-backup-client`コマンドラインツールを使用して、ファイルのバックアップを作成およびリストアできます。[Proxmox Virtual Environment](#)内の[QEMU](#)と[LXC](#)用に、統合されたクライアントを提供します。

1つのバックアップに複数のアーカイブを含めることができます。例えば、[仮想マシン](#)をバックアップする場合、各ディスクはバックアップ内に個別のアーカイブとして保存されます。仮想マシンのコンフィギュレーション自体は追加ファイルとして保存されます。このように、バックアップ全体をスキャンする必要なく、バックアップの重要な部分のみにアクセスし、リストアすることができます。

1.3 主な機能

Proxmox VEのサポート

[Proxmox Virtual Environment](#)は完全にサポートされており、[仮想マシン](#)や[コンテナ](#)を簡単にバックアップできます。[仮想マシン](#)と[コンテナ](#)を簡単にバックアップできます。

パフォーマンス

ソフトウェアスタック全体は、高速性とメモリ効率を提供するために[Rust](#)で書かれています。

重複排除

定期的なバックアップは大量の重複データを生成します。重複排除レイヤーは冗長性を回避し、使用するス

トレージ容量を最小限に抑えます。

増分バックアップ

バックアップ間の変更は通常少ないものです。差分のみを読み込んで送信することで、バックアップによるストレージやネットワークへの影響を軽減します。

データの完全性

内蔵のSHA-256チェックサムアルゴリズムが、バックアップの正確性と一貫性を保証します。

リモート同期

リモートサイトへの効率的なデータ同期が可能です。新しいデータを含むデルタのみが転送されます。

圧縮

超高速のZstandard圧縮は、1秒間に数ギガバイトのデータを圧縮できます。

暗号化

バックアップは、AES-256 GCMを使用してクライアント側で暗号化することができます。この認証暗号化（AE）モードは、最新のハードウェアで非常に高いパフォーマンスを発揮します。クライアント側の暗号化に加え、すべてのデータは安全なTLS接続を介して転送されます。

テープバックアップ

Proxmox Backup Serverは、データの長期的なアーカイブのために、テープへのバックアップとテープライブラリの管理を幅広くサポートしています。

ランサムウェア保護

Proxmox Backup Serverのきめ細かなアクセス制御、データ整合性の検証、再モート同期とテープバックアップによるオフサイトバックアップにより、[ランサムウェア攻撃から重要なデータを保護](#)します。

ウェブインターフェース

統合されたWebベースのユーザーインターフェイスでProxmox Backup Serverを管理します。

オープンソース

秘密はありません。Proxmox Backup Serverはフリーでオープンソースのソフトウェアです。ソースコードはAGPL, v3でライセンスされています。

制限なし

Proxmox Backup Serverには、バックアップストレージやバックアップクライアントに人工的な制限はありません。

エンタープライズサポート

Proxmox Server Solutions GmbHは、Proxmox Backup Serverサブスクリプションプランの形でエンタープライズサポートを提供しています。すべてのサブスクリプションレベルのユーザはProxmox Backup Enterpriseリポジトリにアクセスできます。さらに、ベーシック、スタンダードまたはプレミアムサブスクリプションでは、[Proxmoxカスタマーポータル](#)にアクセスできます。

1.4 データバックアップの理由

バックアップの主な目的は、データ損失から保護することです。データ損失は、ハードウェアの欠陥と人為的ミスの両方によって引き起こされる可能性があります。

よくあるミスは、まだ必要なファイルやフォルダを誤って削除してしまうことです。仮想化では、仮想マシン全体をボタン1つで簡単に削除できるため、この問題がさらに大きくなる可能性があります。

管理者にとって、バックアップはデータを一時的に保存するための便利なツールキットです。例えば、主要なソフトウェア・アップデートをインストールする前に、フル・バックアップを実行するのが一般的です。何か問題が発生した場合、簡単に以前の状態に戻すことができます。

バックアップのもう一つの理由は、法的要件です。一部のデータ、特にビジネス記録は、必要に応じてアクセスできる

ように、法律で数年間安全な場所に保管しなければなりません。

一般的に、データの損失はビジネスに深刻なダメージを与えるため、非常にコストがかかります。したがって、定期的にバックアップを実行し、リストアテストを実行するようにしてください。

1.5 ソフトウェアスタック

Proxmox Backup Serverは複数のコンポーネントで構成されています：

- RESTful API、超高速非同期タスク、軽量な使用統計収集、スケジューリングイベント、特権環境と非特権実行環境の厳格な分離などを提供するサーバーデーモン。
- JavaScript管理ウェブインターフェース
- サーバ用管理CLIツール(*proxmox-backup-manager*)
- 任意のLinux amd64環境からサーバに簡単にアクセスするためのクライアントCLIツール(*proxmox-backup-client*)

ウェブインターフェースを除き、Proxmox Backup Serverのほとんどの部分はRustプログラミング言語で書かれています。

"Rust "プログラミング言語は、より速く、より信頼性の高いソフトウェアを書くのに役立ちます。プログラミング言語の設計において、高レベルの人間工学と低レベルの制御はしばしば対立します。Rustは、強力な技術的能力と優れた開発者エクスペリエンスのバランスを取ることで、従来のような面倒な制御を行うことなく、低レベルの詳細（メモリ使用量など）を制御するオプションを提供します。"

-Rustプログラミング言語

1.6 ヘルプ

1.6.1 エンタープライズサポート

Proxmox Backup Serverベーシック、スタンダード、またはプレミアムサブスクリプションプランのユーザは、[Proxmoxカスタマーポータル](#)にアクセスできます。カスタマーポータルでは、Proxmox開発者による応答時間保証付きのサポートを提供しています。詳細またはボリュームディスカウントについては、sales@proxmox.comまでお問い合わせください。

1.6.2 コミュニティサポートフォーラム

Proxmoxでは、[Proxmoxコミュニティフォーラム](#)を利用して、ユーザーの皆様が議論し、知識を共有することを常に奨励しています。フォーラムはProxmoxサポートチームによって管理されています。大規模なユーザーベースは世界中に広がっています。このような大規模なフォーラムが情報を得るのに最適な場所であることは言うまでもありません。

1.6.3 メーリングリスト

Proxmox Backup Serverは完全にオープンソースであり、貢献を歓迎します！開発者のための主なコミュニケーションチャネルはこちらです：

開発者向けメーリングリスト

[Proxmox Backup Server開発者リスト](#)

1.6.4 バグトラッカー

Proxmoxは<https://bugzilla.proxmox.com>で公開バグトラッカーを運営しています。問題が発生したら、そこに報告してください。問題はバグだけでなく、新機能や機能強化の要望でも構いません。バグトラッカーは問題を追跡するのに役立ち、問題が解決されると通知を送信します。

1.7 ライセンス

著作権 (C) 2019-2025, Proxmox Server Solutions GmbH

このソフトウェアはProxmox Server Solutions GmbHによって書かれました。<support@proxmox.com>

Proxmox Backup Serverはフリーでオープンソースのソフトウェアです。フリーソフトウェアファウンデーションによって発行されたGNU Affero General Public Licenseのバージョン3、またはそれ以降のバージョンのいずれかに従って、使用、再配布、および/または変更を行うことができます。

このプログラムは有用であることを期待して配布されていますが、商品性や特定の目的への適合性についての暗黙の保証もなく、いかなる保証もありません。詳細はGNU Affero General Public Licenseをご覧ください。

あなたは、このプログラムとともにGNU Affero一般公衆利用許諾書のコピーを受け取っているはずです。そうでない場合、[AGPL3](#)をご覧ください。

1.8 歴史

バックアップはIT管理の中心的な側面です。データ損失からの復旧の必要性は基本的なものであり、仮想化によってその必要性は増すばかりです。

このため、Proxmox VEには当初からバックアップツールを同梱しています。このツールはvzdumpと呼ばれ、実行中のLXCコンテナやKVM仮想マシンの一貫したスナップショットを作成することができます。

しかし、vzdumpはフルバックアップしかできません。少量のバックアップであれば問題ありませんが、大規模なVMを使用しているユーザーにとっては負担となります。この場合、バックアップ時間とストレージ使用量の両方が高すぎます。

特に、同じVMのバックアップを何度も取りたいユーザーにとっては。これらの問題を解決するために、私たちは重複排除と増分バックアップを提供する必要がありました。

2018年10月に開発を開始しました。私たちはいくつかの技術やフレームワークを調査し、最終的に高速性とメモリ効率を提供するために、実装言語として[Rust](#)を使用することに決めました。2018年版のRustは、私たちの要件にとって有望に思えました。

2020年7月にProxmox Backup Serverの最初のベータ版をリリースし、2020年11月に最初の安定版をリリースしました。Proxmox Backupは、暗号化と増分バックアップ、完全重複排除をサポートすることで、ネットワーク負荷を大幅に軽減し、貴重なストレージスペースを節約するセキュアな環境を提供します。

インストール

Proxmox Backupはサーバ部とクライアント部に分かれています。サーバ部分はグラフィカルインストーラを使うか、提供されているパッケージリポジトリからDebian上にインストールすることができます。

2.1 システム要件

Proxmox Backupを本番環境で実行する場合は、高品質のサーバハードウェアを使用することをお勧めします。障害が発生したホストの影響を軽減するために、他のProxmox Backup Serverインスタンスから定期的、効率的、インクリメンタルなデータストアの同期を設定することができます。

2.1.1 評価用の最小サーバー要件

これらの最小要件は評価用であり、本番環境では使用しないでください。

- CPU64ビット (x86-64またはAMD64) 、2コア以上
- メモリ (RAM): 2 GB RAM
- ハードドライブ：8GB以上の空き容量
- ネットワークカード (NIC)

2.1.2 推奨サーバーシステム要件

- CPU最新のAMDまたはIntel 64ビットベースCPU、4コア以上
- メモリ: OS、ファイルシステムキャッシュ、Proxmox Backup Serverデーモン用に最低4GB。TiBストレージスペースごとに少なくとももう1GiB追加。
- OSストレージ：
 - 32GiB以上のストレージ空き容量
 - バッテリ保護書き込みキャッシュ(BBU)付きハードウェアRAIDまたは冗長ZFSセットアップを使用します(ZFSはハードウェアRAIDコントローラと互換性がありません)。
- バックアップストレージ
 - ランダムIOワークロードには、高いIOPSを提供する高速ストレージを優先し、最良の結果を得るにはエンタープライズSSDのみを使用します。
 - HDDを使用する場合：メタデータキャッシュを使用することを強く推奨します。例えば、ZFS特殊デバイスマラーを追加します。

- 冗長マルチビット/秒ネットワーク・インターフェイス・カード(NIC)

2.1.3 WebインターフェースにアクセスするためにサポートされているWebブラウザ

サーバーのWebベースのユーザーインターフェイスにアクセスするには、以下のブラウザのいずれかを使用することをお勧めします：

- Firefox、現行リリース、または最新の拡張サポートリリース
- Chrome、今年リリースされたもの
- マイクロソフトが現在サポートしているバージョンのEdge
- Safari（現行リリース）

2.2 インストール方法

Proxmox Backup Serverはさまざまな方法でインストールできます。推奨される方法は、インタラクティブインストーラを起動するだけのインストールメディアの使用です。

2.2.1 インストールメディアの準備

<https://www.proxmox.com/downloads> からインストーラ ISO イメージをダウンロードします。

Proxmox Backup Server インストールメディアはハイブリッド ISO イメージです。2つの方法で動作します：

- DVDに書き込む準備ができたISOイメージファイル。
- USBフラッシュドライブ(USBスティック)にコピーする準備ができている生セクタ(IMG)イメージファイル。

Proxmox Backup Serverをインストールするには、USBフラッシュドライブを使用することをお勧めします。

2.2.2 インストールメディアとしてUSBフラッシュドライブを準備します。

フラッシュドライブには少なくとも2GBのストレージ容量が必要です。

注意

UNetbootin は使用しないでください。Proxmox Backup Serverのインストールイメージでは動作しません。

重要

USBフラッシュドライブ上の既存のデータは上書きされます。そのため、USBフラッシュドライブにまだ必要なデータが入っていないことを確認し、その後アンマウントしてから先に進んでください。

2.2.3 GNU/Linuxの手順

Unix 系 OS では、`dd` コマンドを使って ISO イメージを USB フラッシュドライブにコピーします。まずUSBフラッシュドライブの正しいデバイス名を見つけてください（下記参照）。次に`dd`コマンドを実行してください。お使いの環境によっては、`dd`を実行するためにroot権限が必要です。

```
# dd bs=1M conv= fdatasync if=./proxmox-backup-server_*.iso of=/dev/XYZ=
```

注

`dev/XYZ`を正しいデバイス名に置き換え、入力ファイル名(`if`)のパスに合わせるようにしてください。

⚠️ 注意

間違ったディスクを上書きしないよう、十分注意してください！

正しいUSBデバイス名の検索

USBフラッシュ・ドライブの名前を調べるには、2つの方法があります。1つ目は、フラッシュ・ドライブを接続する前と後の `dmesg` コマンド出力の最後の行を比較する方法です。もう1つは、`lsblk` コマンドの出力を比較する方法です。ターミナルを開いて実行してください：

```
# lsblk
```

次にUSBフラッシュ・ドライブを接続し、もう一度コマンドを実行します：

```
# lsblk
```

新しいデバイスが表示されます。これが使用するデバイスです。念のため、報告されたサイズがお使いのUSBフラッシュ・ドライブと一致しているかどうかを確認してください。

2.2.4 macOSの手順

ターミナルを開きます。

`hdiutil` の `convert` オプションを使って `.iso` ファイルを `.dmg` フォーマットに変換します：

```
# hdiutil convert proxmox-backup-server_*.iso -format UDRW -o proxmox-backup-server_*.dmg
```

ℹ️ 注

macOSは出力ファイル名に `.dmg` を自動的に追加する傾向があります。

現在のデバイスリストを取得するには、コマンドを実行してください：

```
# diskutil list
```

USBフラッシュドライブを挿入し、このコマンドをもう一度実行して、どのデバイスノードが割り当てられているかを確認します。(例: `/dev/diskX`)。

```
# diskutil list
# diskutil unmountDisk /dev/diskX
```

ℹ️ 注

`X` を最後のコマンドのディスク番号に置き換えます。

```
# sudo dd if=proxmox-backup-server_*.dmg bs=1M of=/dev/rdiskX=
```

ℹ️ 注

最後のコマンドでは `diskX` の代わりに `rdiskX` を指定しています。これにより書き込み速度が向上します。

2.2.5 Windows用の説明

エッチャーの使い方

Etcherは箱から出してすぐに使えます。<https://etcher.io> から Etcher をダウンロードしてください。ISO と USB フラッシュドライブの選択プロセスを案内します。

Rufus を使う

Rufusはより軽量な代替手段ですが、動作させるには**DD**モードを使用する必要があります。<https://rufus.ie/> から Rufus をダウンロードしてください。インストールするか、ポータブル版を使用してください。インストール先ドライブとダウンロードしたProxmox ISOファイルを選択してください。

■ 重要

Start をクリックしたら、異なるバージョンのGrubをダウンロードするよう求めるダイアログで "No" をクリックします。次のダイアログで**DD**モードを選択します。

2.2.6 インストールメディアの使用

作成したUSBフラッシュドライブ(またはDVD)をサーバーに挿入します。インストーラの章を読み進めてください。

2.3 サーバのインストール

バックアップサーバは、実際にバックアップされたデータを保存し、ディスク管理などのさまざまな管理タスクのためのWebベースのGUIを提供します。

■ 注意

バックアップサーバは必ず必要です。Proxmox Backupをサーバなしで使用することはできません。

便利なインストーラ、完全なDebianシステム、Proxmox Backup Serverに必要なすべてのパッケージが含まれています。

`installation_medium` を作成すると、起動したインストーラがセットアッププロセスを案内します。ディスクのパーティション分割、言語、タイムゾーン、ネットワーク設定などの基本的な設定の適用を支援し、最後に必要なパッケージを数分以内にインストールします。

対話型インストーラの代わりに、上級ユーザはProxmox Backup Serverを無人インストールすることもできます。

十分なDebianの知識があれば、Proxmox Backup ServerをDebianの上にインストールすることもできます。にインストールすることもできます。

推奨はしませんが、Proxmox Backup ServerをProxmox VEにインストールすることもできます。

2.3.1 インストーラを使用して Proxmox Backup Server をインストールします。

<https://www.proxmox.com/downloads> から ISO をダウンロードします。これには以下が含まれます：

- ローカルディスクをext4、xfs、またはZFSでパーティション分割し、オペレーティングシステムをインストールするProxmox Backup Serverインストーラ。

- 完全なオペレーティングシステム（Debian Linux、64ビット）

- ZFSをサポートするProxmox Linuxカーネル
- バックアップと必要なリソースを管理するための完全なツールセット
- ウェブベースの管理インターフェイス

i 注意事項

選択したドライブ上の既存のデータは、インストール処理中に上書きされます。インストーラは、他のオペレーティングシステムのブートメニューインストラクションを追加しません。

インストールメディア(USB フラッシュドライブや DVD など)を挿入し、そこから起動してください。

i 注意

サーバーのファームウェア設定で、インストールメディア(USBなど)からの起動を有効にし、希望の起動順序を設定する必要があるかもしれません。[Proxmox Backup Server](#)バージョン3.1より前のインストーラを起動する場合は、セキュアブートを無効にする必要があります。



正しいエントリ(たとえば、*Boot from USB*)を選択すると、Proxmox Backup Serverメニューが表示され、次のオプションのいずれかを選択できます：

Proxmox Backup Server のインストール (グラフィカル)

通常のインストールを開始します。

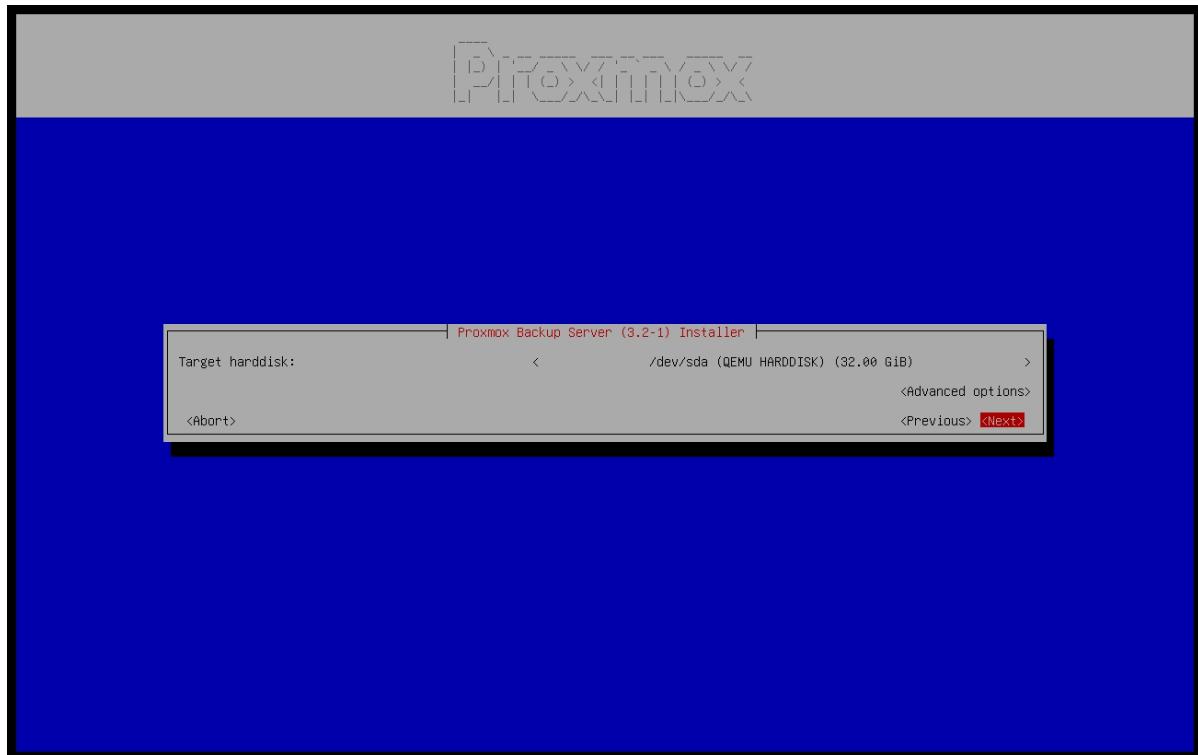
ヒント: キーボードのみでインストールウィザードを使用することができます。ボタンは、ALTキーと各ボタンの下線文字の組み合わせでクリックできます。たとえば、[次へ] ボタンを押すには、ALT+Nを押します。

Proxmox バックアップサーバのインストール（コンソール）

ターミナルモードのインストールウィザードを開始します。グラフィカルインストーラと同じ全体的なインストールエクスペリエンスを提供しますが、非常に古いハードウェアや非常に新しいハードウェアとの互換性が一般的に優れています。

Proxmox Backup Server のインストール（ターミナル UI、シリアルコンソール）

ターミナルモードのインストールウィザードを開始し、さらにLinuxカーネルがマシンの(最初の)シリアルポートを入出力に使用するように設定します。これは、マシンが完全にヘッダレスで、シリアルコンソールしか利用できない場合に使用できます。



どちらのモードでも、実際のインストールプロセスには同じコードベースを使用し、10年以上にわたるバグ修正の恩恵を受け、機能の同等性を確保しています。

ヒント: コンソールまたはターミナル UI オプションは、ドライバの問題などでグラフィカルインストーラが正しく動作しない場合に使用できます。nomodeset_kernel_param も参照してください。

詳細オプション：Proxmox バックアップサーバのインストール（デバッグモード）

デバッグモードでインストールを開始します。いくつかのインストールステップでコンソールが開きます。これは、何か問題が発生した場合のデバッグに役立ちます。デバッグコンソールを終了するには、CTRL-D を押してください。このオプションを使用すると、基本的なツールがすべて利用可能なライブシステムを起動できます。例えば、次のように使用できます、

を使用して、デグレードした ZFS rpool を修復したり、既存の Proxmox Backup Server セットアップのホストブートローダを修正したりできます。

詳細オプション：Proxmox Backup Server のインストール（ターミナルUI、デバッグモード）

グラフィカルデバッグモードと同じですが、代わりにターミナルベースのインストーラを実行するようにシステムを準備します。

詳細オプション：Proxmox バックアップサーバのインストール（シリアルコンソールデバッグモード）

ターミナルベースのデバッグモードと同じですが、入出力にマシンの(最初の)シリアルポートを使用するようにLinuxカーネル

ネルを追加設定します。

詳細オプション：レスキュー ブート

このオプションを使用すると、既存のインストールを起動できます。接続されている全てのハードディスクを検索します。既存のインストールが見つかれば、ISO の Linux カーネルを使ってそのディスクに直接ブートします。これは

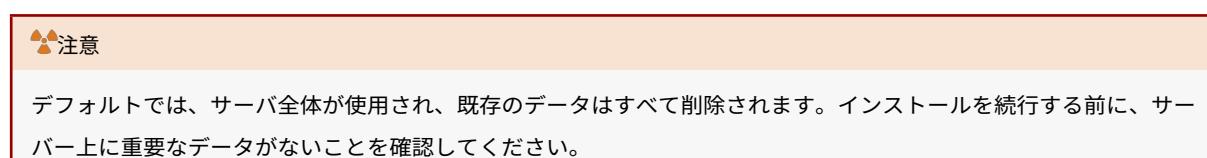
ブートローダ (GRUB/systemd-boot) に問題があったり、BIOS/UEFI がディスクからブートブロックを読み取れない場合に便利です。

高度なオプションメモリのテスト (**memtest86+**)

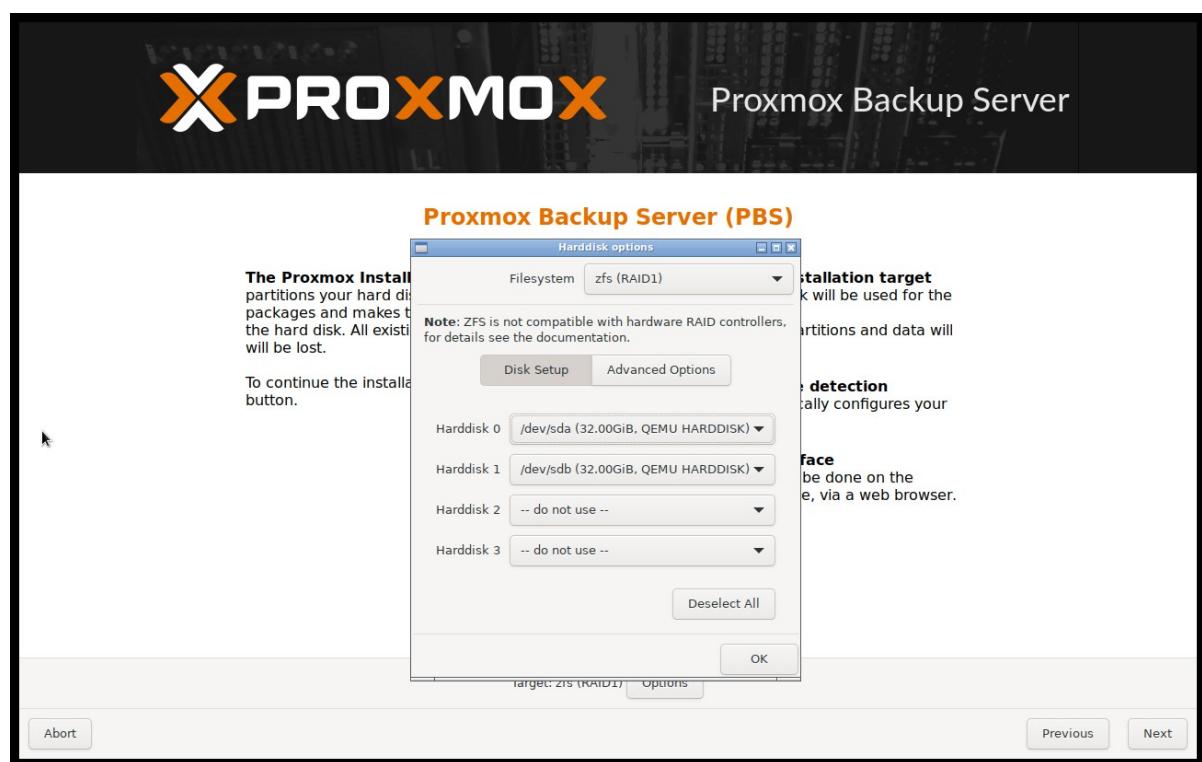
memtest86+ を実行します。メモリが機能しているか、エラーがないかをチェックするのに便利です。このオプションを実行するには、UEFI フームウェアセットアップユーティリティで Secure Boot をオフにする必要があります。

通常、*Install Proxmox Backup Server (Graphical)* を選択してインストールを開始します。

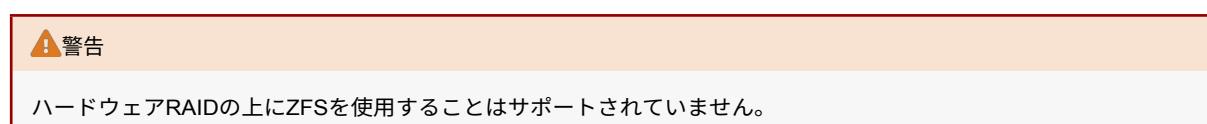
最初のステップは、EULA (End User License Agreement)を読むことです。続いて、インストール先のハードディスクを選択します。

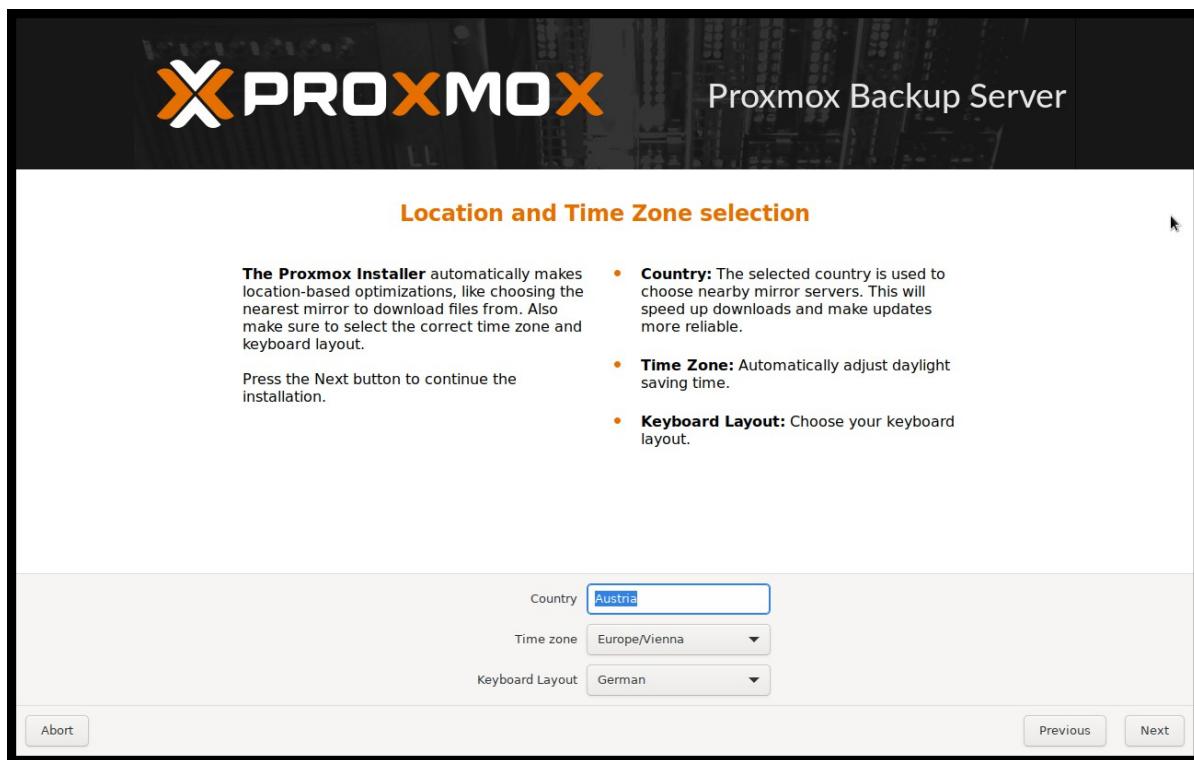


Options ボタンをクリックすると、ターゲットファイルシステムを選択できます。ファイルシステムとして ext4 または xfs を選択すると、インストーラは LVM を使用します。

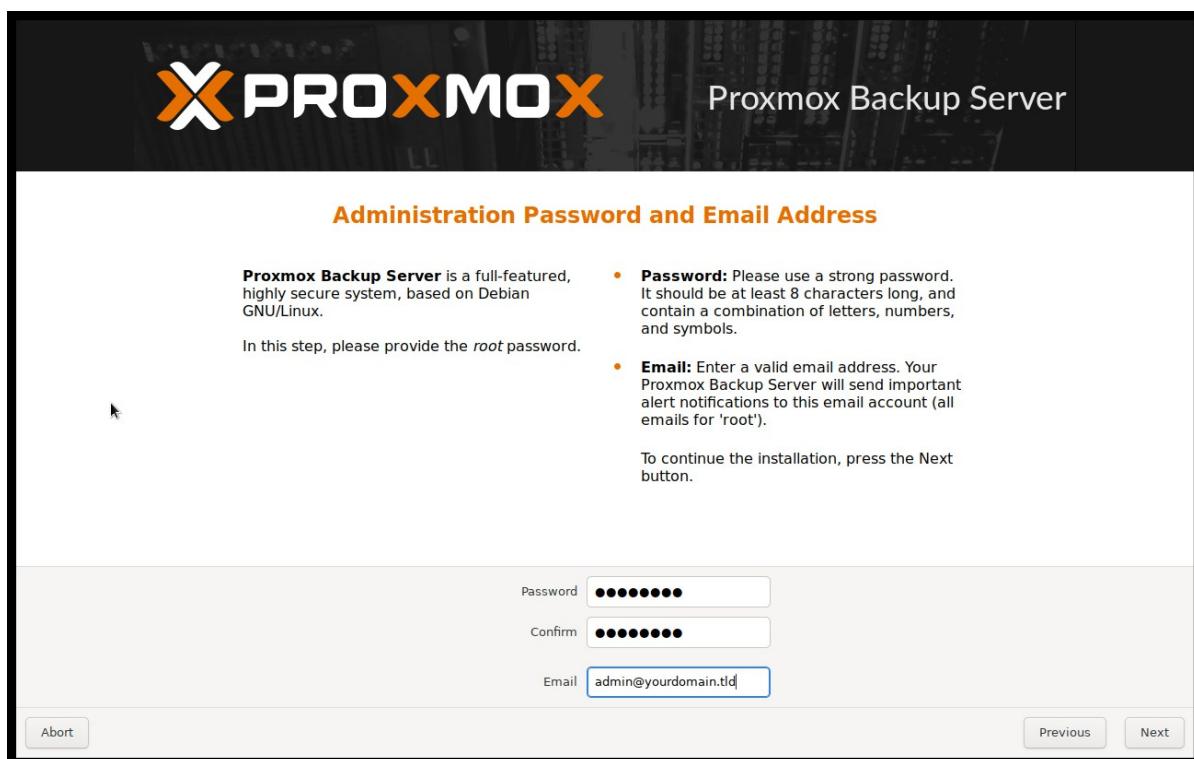


Proxmox Backup ServerはZFSにインストールすることもできます。ZFSにはいくつかのソフトウェアRAIDレベルが用意されているため、ハードウェアRAIDコントローラを搭載していないシステム向けのオプションです。オプションダイアログでターゲットディスクを選択する必要があります。ZFS固有の設定は、詳細オプションで変更できます。





次のページでは、お住まいの場所、タイムゾーン、キーボードレイアウトなどの基本的な設定オプションを尋ねられます。位置情報は、アップデートの速度を上げるために、近くのダウンロードサーバーを選択するために使用されます。インストーラは通常、これらの設定を自動検出できますので、自動検出に失敗した場合や、あなたの国で一般的に使用されていないキーボードレイアウトを使用したい場合にのみ、稀に設定を変更する必要があります。



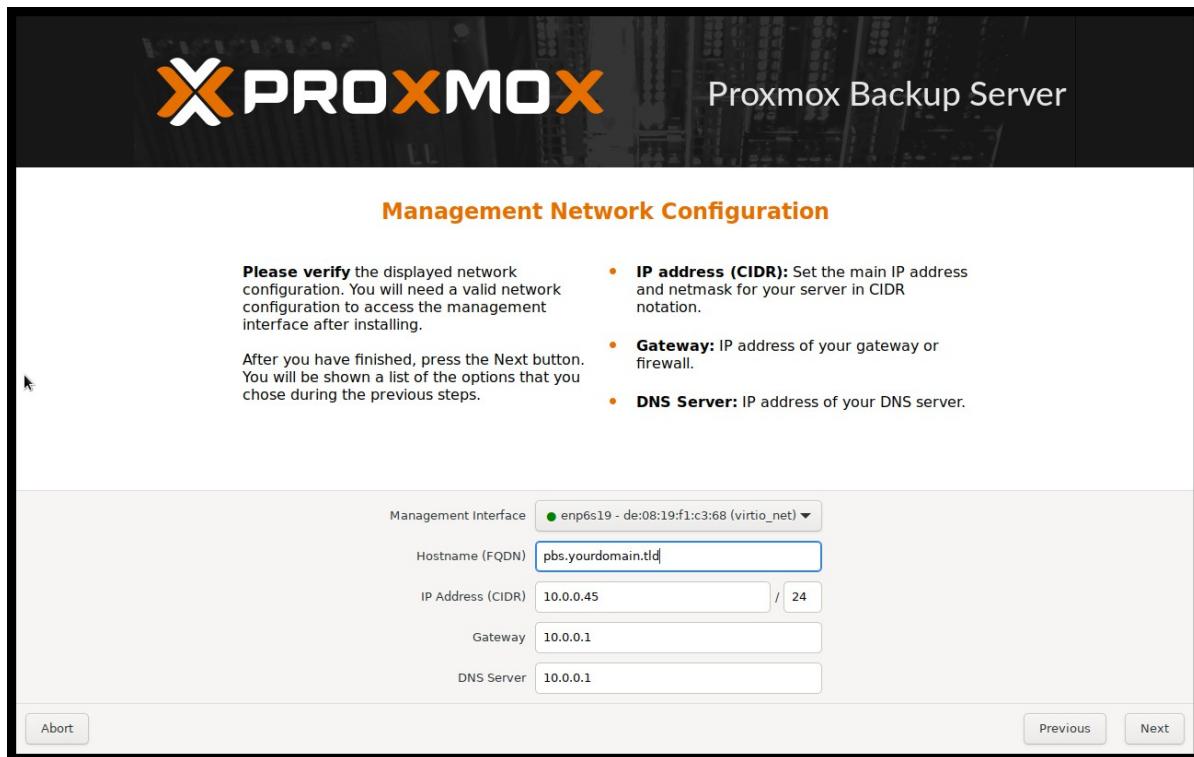
次に、スーパーユーザー（root）のパスワードとメールアドレスを指定する必要があります。パスワードは少なくとも8文字以上でなければなりません。より強力なパスワードを使用することを強くお勧めします。

いくつかのガイドラインがあります：

- パスワードの長さは12文字以上にしてください。
- 小文字と大文字のアルファベット、数字、記号を使用します。
- 文字の繰り返し、キーボードのパターン、一般的な辞書の単語、文字や数字の並び、ユーザー名、親戚やペットの名前、恋愛関係（現在または過去）、伝記的な情報（ID番号、先祖の名前や日付など）は避けてください。

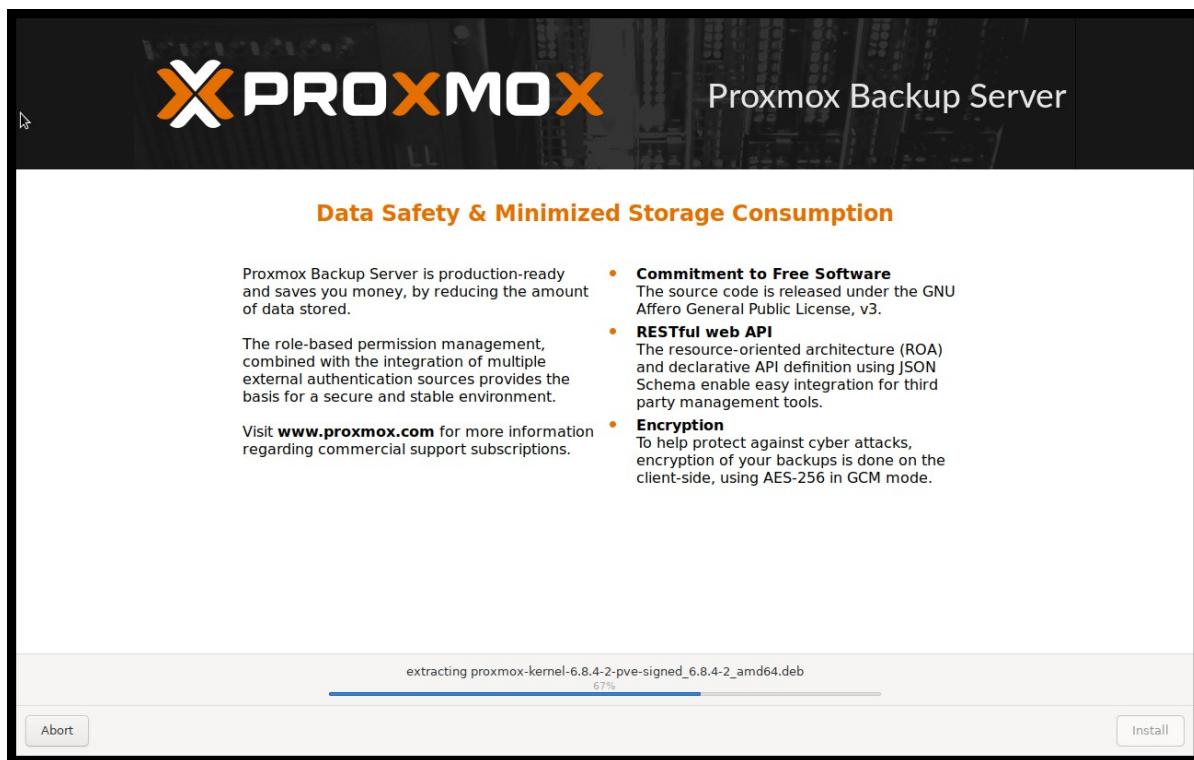
電子メールアドレスは、システム管理者に通知を送信するために使用されます。例えば

- 利用可能なパッケージのアップデートに関する情報。
- 定期的な *cron* ジョブからのエラーメッセージ。



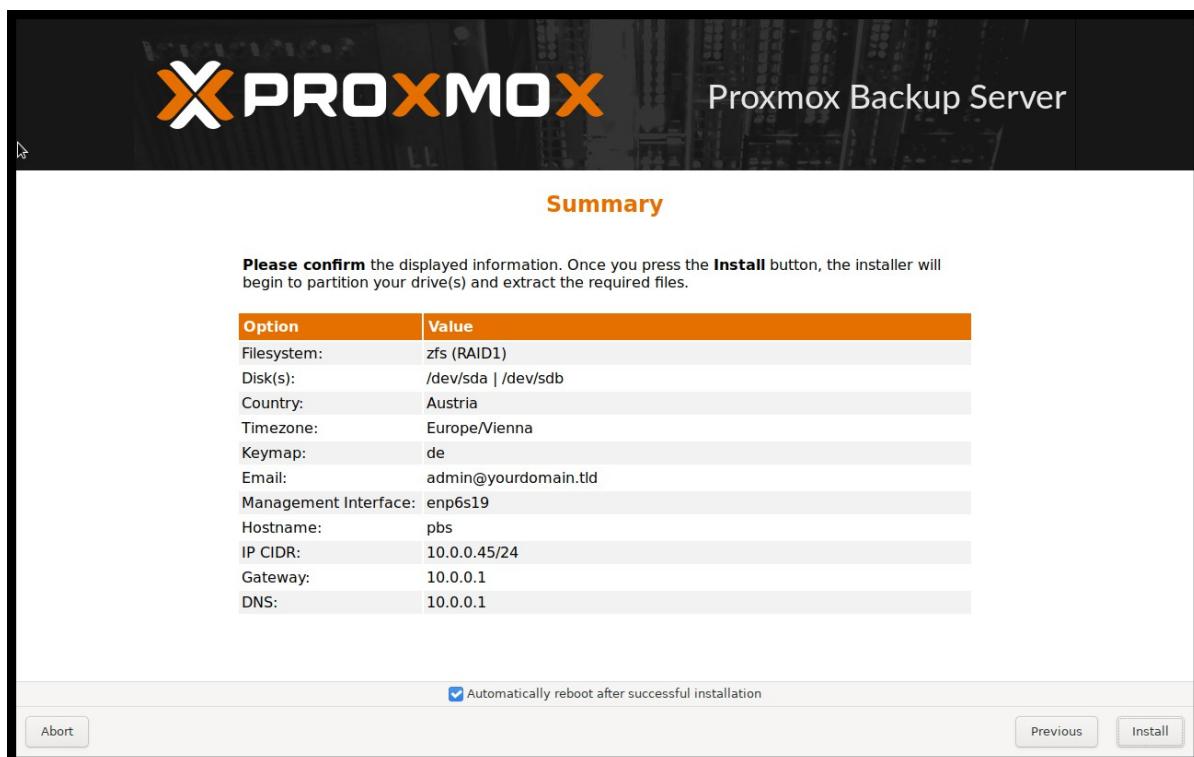
これら全ての通知メールは指定されたメールアドレスに送信されます。

最後のステップはネットワーク設定です。*UP* されたネットワークインターフェイスはドロップダウンメニューの名前に塗りつぶされた円が表示されます。インストール中はIPv4アドレスかIPv6アドレスのどちらかを指定できますが、両方を指定することはできません。デュアルスタックノードを設定するには、インストール後にIPアドレスを追加します。



次のステップでは、以前に選択したオプションの概要が表示されます。すべての設定を再確認し、設定を変更する必要がある場合は、[Previous]ボタンを使用してください。

インストールをクリックすると、インストーラがディスクのフォーマットとターゲットディスクへのパッケージのコピーを開始します。このステップが終了するまで待ち、インストールメディアを取り外してシステムを再起動してください。



パッケージのコピーには通常数分かかりますが、そのほとんどはインストールメディアの速度とターゲットディスクの性能に依存します。

パッケージのコピーとセットアップが終了したら、サーバを再起動します。これはデフォルトでは数秒後に自動的に行われます。

インストールの失敗

インストールに失敗した場合は、2台目のTTY(CTRL+ALT+F2)で特定のエラーを確認し、システムが最小要件を満たしていることを確認してください。

それでもインストールがうまくいかない場合は、[ヘルプの入手方法の章](#)を参照してください。

インストール後の管理インターフェイスへのアクセス

インストールとシステムの再起動が正常に完了したら、Proxmox Backup Server の Web インタフェースを使用してさらに設定を行うことができます。

- ブラウザをインストール時に指定されたIPアドレスとポート8007（例：<https://pbs.yourdomain.tld:8007>）に合わせます。
- root (realm Linux PAM 標準認証)** ユーザー名とインストール時に選択したパスワードを使用してログインします。
- Enterprise リポジトリにアクセスするには、サブスクリプションキーをアップロードしてください。そうでない場合は、セキュリティ修正、バグ修正、新機能のアップデートを取得するために、あまりテストされていないパブリックパッケージリポジトリのいずれかをセットアップする必要があります。
- IP 設定とホスト名を確認します。
- タイムゾーンを確認します。

高度な LVM 設定オプション

インストーラは、**pbs** という名前のボリュームグループ(VG)を作成します。

ファイルシステムとしてext4またはxfsを使用している場合は、ルートとスワップ。これらのボリュームのサイズを制御するには

- hdszie**

使用するハードディスクの総サイズを定義します。こうすることで、さらなるパーティション分割のためにハードディスクの空き領域を確保することができます。

- スワップサイズ**

スワップボリュームのサイズを定義します。デフォルトはインストールされているメモリのサイズで、最小4 GB、最大8 GBです。結果の値は **hdszie/8** より大きくすることはできません。

0に設定すると、スワップボリュームは作成されません。

- minfree**

LVMボリュームグループ**pbs**に残すべき空き容量を定義します。128GB以上のストレージが使用可能な場合、デフォルトは16GBで、そうでない場合は**hdszie/8**が使用されます。

高度なZFS設定オプション

ZFSを使用する場合、インストーラはZFS プール `rpool` を作成します。スワップ領域は作成されませんが、スワップ用にインストールディスク上のパーティション化されていない領域を確保することができます。インストール後にスワップ `zvol`

を作成することもできますが、問題が発生する可能性があります ([ZFS swap notes を参照](#))。

- `ashift`

作成されるプールの `ashift` 値を定義します。`ashift` は、少なくとも基盤となるディスクのセクタサイズ (`ashift` の2乗がセクタサイズ) に設定する必要があります。

- `compress`

`rpool` で圧縮を有効にするかどうかを定義します。

- チェックサム

`rpool` にどのチェックサムアルゴリズムを使用するかを定義します。

- コピー

`rpool` の `copies` パラメータを定義します。se-mantics、および、なぜこれがディスクレベルの冗長性を置き換えないかについては、[zfs\(8\) man](#) ページを確認してください。

- `hdszie`

使用するハードディスクの合計サイズを定義します。`hdszie` は、ブート可能なディスク、つまり RAID0、RAID1、RAID10 の最初のディスクまたはミラー、RAID-Z[123] のすべてのディスクに対してのみ有効です。

ZFSパフォーマンスのヒント

ZFSは大容量のメモリで最適に動作します。ZFSを使用する場合は、十分なRAMを用意してください。TBの未加工ディスク容量に対して、4GBと1GBのRAMを使用するのが良い計算です。

ZFSは、ZFS Intent Log (ZIL) と呼ばれる書き込みキャッシュとして専用ドライブを使用できます。これには高速ドライブ (SSD) を使用してください。インストール後に以下のコマンドで追加できます：

```
zpool add <pool-name> log </dev/path_to_fast_ssd> # zpool add <pool-name> log </dev/path_to_fast_ssd
```

nomodesetカーネルパラメータの追加

非常に古いハードウェアや非常に新しいハードウェアでは、グラフィックドライバが原因で問題が発生することがあります。ブート中にインストールがハングする場合は、`nomodeset` パラメータを追加してみてください。これは Linux カーネルがグラフィックドライバをロードしないようにし、BIOS/UEFI が提供するフレーム

バッファを使い続けさせます。

Proxmox Backup Server ブートローダメニューで、[Install Proxmox Backup Server (Con-sole)]に移動し、eキーを押してエントリを編集します。矢印キーを使用してlinuxで始まる行に移動し、カーソルをその行の最後に移動して、パラメータ `nomodeset` をスペースで区切って追加します。

次に Ctrl-X または F10 を押して設定をブートします。

2.3.2 Proxmox Backup Server の無人インストール

Proxmox Backup Serverを自動でインストールすることができます。これにより、ベアメタル上でのセットアッププロセスを完全に自動化できます。インストールが完了し、ホストが起動したら、Ansibleのような自動化ツールを使用してインストールをさらに構成できます。

インストーラに必要なオプションは、アンサーファイルで提供する必要があります。このファイルでは、どのディスク

やネットワークカードを使用するかを決定するフィルタールールを使用できます。

自動インストールを使用するには、まずインストール ISO を準備する必要があります。無人インストールの詳細と情報については、[wiki](#)を参照してください。

2.3.3 Proxmox バックアップサーバのDebianへのインストール

Proxmoxは標準的なDebianインストールにインストールできるDebianパッケージのセットとして出荷されます。[Debian パッケージリポジトリ](#)を設定した後、以下を実行する必要があります：

```
# apt update
# apt install proxmox-backup-server
```

上記のコマンドは、現在の (Debian) カーネルを維持し、必要なパッケージの最小 セットをインストールします。

インストーラと同じパッケージセットをインストールしたい場合は、以下を使用してください：

```
# apt update
# apt install proxmox-backup
```

これで、必要なパッケージ、ZFS サポート付き Proxmox カーネル、一般的で便利なパッケージー式がインストールされます。

⚠️ 注意

既存のDebianインストール上にProxmox Backupをインストールするのは簡単そうに見えますが、ベースシステムとローカルストレージが正しく設定されていることが前提です。

ベースシステムとローカルストレージが正しくセットアップされていることが前提です。一般的に、これは些細なことではありません。

特にLVMやZFSを使用する場合です。ネットワークの設定も完全にあなた次第です。

ⓘ 注記

Proxmox Backup ServerのWebインターフェースには、ポート8007でHTTPSを使用してWebブラウザからアクセスできます。例えば、<https://<ip-or-dns-name>:8007>。

2.3.4 Proxmox VE に Proxmox Backup Server をインストールします。

[Debian パッケージリポジトリ](#)を設定した後、以下を実行してください：

```
# apt update
# apt install proxmox-backup-server
```

⚠️ 注意

バックアップ・サーバーをハイパーバイザーに直接インストールすることは推奨されません。別の物理サーバーを使用してバックアップを保存する方が安全です。ハイパーバイザーサーバーに障害が発生しても、バックアップにアクセスできます。

ⓘ 注

ポート 8007 で HTTPS を使用して、Web ブラウザから Proxmox Backup Server の Web インターフェイスにアクセスできます。たとえば、<https://<ip-or-dns-name>:8007>。

2.4 クライアントのインストール

2.4.1 ProxmoxバックアップクライアントのDebianへのインストール

Proxmox は標準的な Debian インストールの上にインストールする Debian パッケージのセットとして出荷されます。

APT ベースのProxmox Backup Client リポジトリを設定した後、以下を実行する必要があります：

```
# apt update
# apt install proxmox-backup-client
```

2.4.2 静的にリンクされたProxmoxバックアップクライアントのインストール

Proxmox は、最新の x86-64 Linux システムで動作する Proxmox バックアップクライアントの静的にリンクされたビルドを提供します。

現在、Debianパッケージとして提供されています。*APTベースのProxmoxバックアップクライアントリポジトリを設定した後、以下を実行する必要があります：*

```
# apt update
# apt install proxmox-backup-client-static
```

このパッケージは`proxmox-backup-client`パッケージと競合します。どちらもクライアントを実行ファイルとして`/usr/bin/proxmox-backup-client`パスに提供するからです。

この実行ファイルを他の、例えば Debian 以外の Linux システムにコピーすることができます。

Proxmoxバックアップクライアントの使用方法の詳細については、[バックアップクライアントの使用方法](#)を参照してください。

2.5 Debianパッケージリポジトリ

Debian ベースのシステムでは、パッケージ管理ツールとして `APT` を使用しています。リポジトリのリストは `/etc/apt/sources.list` と `/etc/apt/sources.d/` ディレクトリにある `.list` または `.sources` ファイルで定義されます。アップデートは、`apt` コマンドラインツールを使って直接インストールすることも、GUI を使ってインストールすることもできます。

GUI からもインストールできます。

2.5.1 リポジトリ形式

`APT` リポジトリは、古いシングルラインフォーマットと新しい deb822 フォーマットの 2 つの異なるフォーマットで設定できます。どの形式を選んでも、`apt update` は設定されたすべてのソースから情報を取得します。

單一行

單一行のリポジトリは `.list` ファイルで定義され、1 行に 1 つのパッケージリポジトリを、最も優先されるソースを最初にリストします。空行は無視され、行のどこかに `#` 文字があると、その行の残りはコメントとしてマークされます。

deb822 スタイル

新しい deb822 複数行フォーマットが `.sources` ファイルで使われています。各リポジトリは複数のキーと値のペアを持つスタンザで構成されます。スタンザは単一のグループです。空白行で区切ることで、1 つのファイルに複数のスタンザを含めることができます。行をコメントアウトするには `#` を使用します。

注意

Debian Trixie では、リポジトリの近代化を推奨します。`apt modernize-sources` コマンドを実行すると、既存のリポジトリを自動的に近代化できます。

2.5.2 Debian ベースリポジトリ

Debian 自体が提供するすべてのパッケージのアップデートを取得するには、最低限 Debian ベースリポジトリが必要です：

リスト 1： ファイル /etc/apt/sources.list.d/debian.sources

```
タイプ: deb
URL: http://deb.debian.org/debian/ サイト: trixie
trixie-updates
コンポーネント: main contrib non-free-firmware
署名バイ: /usr/share/keyrings/debian-archive-keyring.gpg Type: deb
URI: http://security.debian.org/debian-security/ サイト: trixie-security
コンポーネント: main contrib non-free-firmware
署名バイ: /usr/share/keyrings/debian-archive-keyring.gpg
```

さらに、Proxmox Backupのアップデートを取得するには、Proxmoxのパッケージリポジトリが必要です。

Enabled	Types	URIs	Suites	Components	Options	Origin	Comment
<input checked="" type="checkbox"/>	deb	http://ftp.at.debian.org/debian	bullseye	main contrib		Debian	
<input checked="" type="checkbox"/>	deb	http://ftp.at.debian.org/debian	bullseye-updates	main contrib		Debian	
<input checked="" type="checkbox"/>	deb	http://deb.debian.org/debian	bullseye-backports	main contrib		Debian Ba...	
<input checked="" type="checkbox"/>	deb	http://security.debian.org	bullseye-security	main contrib		Debian	security updates
<input type="checkbox"/>	deb	http://download.proxmox.com/debian/pbs	bullseye	pbs-test		Proxmox	
<hr/>							
<input checked="" type="checkbox"/>	deb	https://enterprise.proxmox.com/debian/pbs	bullseye	pbs-enterprise		Proxmox	
<hr/>							
<input checked="" type="checkbox"/>	deb	http://repo.proxmox.com/staging/pbs	bullseye	pbs-2		Proxmox	

2.5.3 Proxmox Backup Enterprise リポジトリ

これは安定した推奨リポジトリです。すべての Proxmox Backup サブスクリプションユーザーが利用できます。最も安定したパッケージが含まれており、実稼働環境での使用に適しています。
このリポジトリは

pbs-enterprise リポジトリはデフォルトで有効になっています：

リスト 2： ファイル /etc/apt/sources.list.d/pbs-enterprise.sources

```
タイプ: deb
URL: https://enterprise.proxmox.com/debian/pbs
サイト: trixie
コンポーネント: pbs-enterprise
署名バイ: /usr/share/keyrings/proxmox-archive-keyring.gpg
```

重要なセキュリティ修正を見逃さないために、新しいパッケージが利用可能になるとすぐに、スーパーユーザ (root@pam ユーザ) に電子メールで通知されます。各パッケージの変更ログと詳細は GUIで見ることができます(利用可能な場合)。

このリポジトリにアクセスするには有効なサブスクリプションキーが必要です。サブスクリプションレベルと価格に関する詳細は <https://www.proxmox.com/en/proxmox-backup-server/pricing> を参照してください。

注意

このリポジトリを無効にするには、スタンザに `Enabled: false` という行を追加します。

2.5.4 Proxmox Backup サブスクリプションなしリポジトリ

名前が示すように、このリポジトリにアクセスするためにサブスクリプションキーは必要ありません。テストや本番環境以外での使用に使用できます。これらのパッケージは常に厳重にテストされ検証されているわけではないので、本番サーバーで使用することはお勧めしません。

この リポジトリ を `/etc/apt/sources.list.d/proxmox.sources` に設定することをお勧めします。

リスト3： ファイル： ファイル：
 /etc/apt/sources.list.d/proxmox.sources

```
タイプ: deb
URI: http://download.proxmox.com/debian/pbs
ソースネーム: trixie pbs-no-subscription
署名済み: /usr/share/keyrings/proxmox-archive-keyring.gpg
```

2.5.5 Proxmox バックアップテストリポジトリ

このリポジトリには最新のパッケージが含まれており、開発者が新機能をテストするために多用されています。

このリポジトリにアクセスするには、`/etc/apt/sources.list.d/proxmox.sources` に以下のスタンザを追加します：

リスト4: pbstest の sources.list エントリ

```
タイプ: deb
URI: http://download.proxmox.com/debian/pbs
ソースネーム: trixie pbs-test
署名バイ: /usr/share/keyrings/proxmox-archive-keyring.gpg
```

2.5.6 Proxmox Backup クライアント専用リポジトリ

*Proxmox*プロジェクトをベースにしていないLinuxディストリビューションを使用しているシステムで *Proxmox Backup Client* を使用したい場合は、クライアント専用のリポジトリを使用できます。

現在、APTベースのシステム用のクライアントリポジトリしかありません。

APTベースの Proxmox Backup Client リポジトリ

DebianやUbuntuの派生版のように、`apt`をパッケージマネージャーとして使っている最新のLinuxディストリビューションでは、APTベースのリポジトリを使うことができます。

このリポジトリを設定するには、まず *Proxmox* リリースキーをセットアップする必要があります。その後、リポジトリの URL を APT のソースリストに追加してください。

Debian 13 (Trixie) ベースのリリース用リポジトリ

このリポジトリは

- Debian Trixie

ファイル `/etc/apt/sources.list.d/pbs-client.sources` を編集し、以下のスニペットを追加してください リスト5:

ファイル: `/etc/apt/sources.list.d/pbs`

```
タイプ: deb
URI: http://download.proxmox.com/debian/pbs-client
スイート: trixie
ソースネジット: main
署名バイ: /usr/share/keyrings/proxmox-archive-keyring.gpg
DEBIAN 12 (bookworm) ベースのリリース用リポジトリ
```

このリポジトリは

- Debian Bookworm

`etc/apt/sources.list.d/pbs-client.list` ファイルを編集し、以下のスニペットを追加してください。

リスト 6: ファイル: `/etc/apt/sources.list`

```
deb http://download.proxmox.com/debian/pbs-client bookworm main
```

Debian 11 (Bullseye) ベースのリリース用リポジトリ

このリポジトリは

- Debian Bullseye

`etc/apt/sources.list.d/pbs-client.list` ファイルを編集し、以下のスニペットを追加してください。

リスト 7: ファイル: `/etc/apt/sources.list`

```
deb http://download.proxmox.com/debian/pbs-client bullseye main
```

Debian 10 (Buster) ベースのリリース用リポジトリ

このリポジトリは

- Debian バスター
- Ubuntu 20.04 LTS

これより古いバージョンでも動くかもしれませんし、最近のリリースバージョンでも動くはずです。

`etc/apt/sources.list.d/pbs-client.list` ファイルを編集し、以下のスニペットを追加してください。

リスト 8: ファイル: `/etc/apt/sources.list`

```
deb http://download.proxmox.com/debian/pbs-client buster main
```

2.5.7 SecureApt

リポジトリ内の `Release` ファイルは GnuPG で署名されています。APTはこれらの署名を使用して、すべてのパッケージが信頼できるソースからのものであることを検証しています。

Proxmox Backup Server を公式 ISO イメージからインストールする場合、検証キーはすでにインストールされています。

Proxmox Backup ServerをDebian上にインストールする場合は、以下のコマンドでキーをダウンロードしてインストールしてください：

```
# wget https://enterprise.proxmox.com/debian/proxmox-archive-keyring-trixie.gpg -O /usr/share/→keyrings/proxmox
→keyrings/proxmox-archive-keyring.gpg
```

注

上記の `wget` コマンドは、Debian Trixie ベースの Proxmox リリース用のキーリングを追加します。`proxmox-archive-keyring` パッケージがインストールされると、このファイルが管理されます。その時点で、ハッシュ

がこのファイルのハッシュと一致しなくなる可能性があります。これは意図的なもので、aptは信頼できる鍵のみが使用されていることを保証します。**proxmox-archive-keyring**がインストールされたら、このファイルを変更することはお勧めしません。

SHA256チェックサムを確認します：

```
# sha256sum /usr/share/keyrings/proxmox-archive-keyring.gpg
/usr/share/keyrings/proxmox-
'--archive-keyring.gpg
```

と md5sum を実行します：

```
# md5sum /usr/share/keyrings/proxmox-archive-keyring.gpg
/usr/share/keyrings/proxmox-archive-keyring.gpg
```

i 注意

鍵をダウンロードするパスが、あなたのファイルの
で指定したパスと一致していることを確認してください。

2.5.8 HTTP プロキシ経由のリポジトリアクセス

セットアップによっては、インターネットへのアクセスが制限されており、中央プロキシ経由でしかアクセスできない場合があります。Proxmox Backup Server の Web インターフェイスの [設定] -> [認証] タブで HTTP プロキシを設定できます。

このプロキシを設定すると、適切なネットワーク要求と Proxmox Backup Server サポートサブスクリプションの確認に使用されます。

標準的なHTTPプロキシ設定が使用できます。[http://]<host>[:port]ここで、<host>部分には認証が含まれる場合があります。例: `http://user:pass@proxy.example.org:12345`

用語解説

3.1 バックアップ内容

重複排除を行う場合、パフォーマンスや重複排除率で最適な結果を得るために様々な戦略があります。データの種類に応じて、固定サイズまたは可変サイズのチャンクに分割することができます。

固定サイズのチャンキングは最小限のCPUパワーを必要とし、仮想マシンイメージのバックアップに使用されます。

可変サイズのチャンキングはより多くのCPUパワーを必要としますが、ファイルアーカイブの重複排除率を向上させるために不可欠です。

Proxmox Backup Serverは両方の戦略をサポートしています。

3.1.1 イメージアーカイブ:<名前>.img

これは仮想マシンイメージやその他の大きなバイナリデータに使用されます。コンテンツは固定サイズのチャンクに分割されます。

3.1.2 ファイルアーカイブ:<name>.pxar

ファイルアーカイブは完全なディレクトリツリーを保存します。コンテンツは*Proxmox File Archive Format (.pxar)*を使用して保存され、可変サイズのチャンクに分割されます。このフォーマットは、重複排除率が高くなるように最適化されています。

3.1.3 バイナリデータ (BLOB)

このタイプは、設定ファイルなどの小さい（16MB未満）バイナリデータの保存に使用されます。より大きなファイルはイメージアーカイブとして保存する必要があります。

⚠️ 注意

すべてのファイルをBLOBとして保存しないでください。代わりに、ディレクトリツリー全体を格納するためにファイルアーカイブを使用してください。

3.1.4 カタログファイル: catalog.pcatt

カタログ ファイルはファイルアーカイブのインデックスです。このファイルには、含まれるファイルのリストが含まれており、検索操作を高速化するために使用されます。

3.1.5 マニフェスト: `index.json`

マニフェストには、バックアップされたすべてのファイルのリストと、そのサイズとチェックサムが含まれます。バックアップの一貫性を検証するために使用されます。

3.2 バックアップ名前空間

チャプタ

名前空間は、名前の衝突を回避し、よりきめ細かいアクセス制御を可能にしながら、複数のソースに対して单一のチャンクストア重複排除ドメインを再利用することを可能にします。

基本的には単純なディレクトリ構造として実装され、個別の設定は不要です。

3.3 バックアップの種類

バックアップサーバはバックアップをタイプ別にグループ化します：

vm

このタイプは仮想マシンに使用されます。通常、仮想マシンの設定ファイルと各ディスクのイメージアーカイブで構成されます。

ct

このタイプはコンテナに使用されます。コンテナの設定ファイルと、ファイルシステムのコンテンツ用の单一のファイルアーカイブから構成されます。

ホスト

このタイプは、マシン内から作成されるファイル/ディレクトリバックアップに使用されます。通常は物理ホストですが、仮想マシンやコンテナでも可能です。このようなバックアップにはファイルやイメージのアーカイブを含めることができます。

3.4 バックアップID

特定のバックアップ・タイプとバックアップ・ネームスペースに固有のID。通常、仮想マシンまたはコンテナのID。ホストタイプのバックアップでは通常、ホスト名が使用されます。

3.5 バックアップ時間

バックアップが作成された時刻。

3.6 バックアップ・グループ

<type>/<id>のタブルをバックアップ・グループと呼びます。このグループには1つ以上のバックアップ・スナップショットが含まれます。

3.7 バックアップ・スナップショット

<type>/<id>/<time>のトリプレットをバックアップ・スナップショットと呼びます。これは、ネームスペース内の特定のバックアップを一意に識別します。

リスト1: バックアップ・スナップショットの例

```
vm/104/2019-10-09T08:01:06Z  
host/elsa/2019-11-08T09:48:14Z
```

ご覧のとおり、時刻フォーマットはRFC3339の協定世界時（UTC、末尾のZで識別）です。

グラフィカルユーザインターフェース

Proxmox Backup Serverは、サーバを管理するための統合されたWebベースのインターフェイスを提供します。これは、Webブラウザからすべての管理タスクを実行でき、追加の管理ツールをインストールする必要がないことを意味します。ウェブインターフェースはビルトインコンソールも提供しますので、コマンドラインがお好みの場合や、より高度な制御が必要な場合は、このオプションをご利用ください。

ウェブインターフェースは <https://youripaddress:8007> からアクセスできます。デフォルトのログインは `root` で、パスワードはインストール時に指定したものか、Debian 上にインストールした場合の `root` ユーザのパスワードのいずれかです。

4.1 機能

- Proxmox Backup Serverのシンプルな管理インターフェース
- タスク、ログ、リソース使用量の監視
- ユーザー、権限、データストアなどの管理
- セキュアなHTML5コンソール
- 複数の認証ソースのサポート
- 複数言語のサポート
- ExtJS 6.x JavaScriptフレームワークがベース

4.2 ログイン

The screenshot shows the 'Proxmox Backup Server Login' window. It contains the following fields:

- User name: `root`
- Password: (Redacted)
- Realm: Linux PAM standard authentication
- Language: English
- Save User name:
- Login button

ウェブインターフェースに接続すると、最初にログインウィンドウが表示されます。Proxmox Backup Serverは様々な言語と認証バックエンド(*Realms*)をサポートしています。

注意

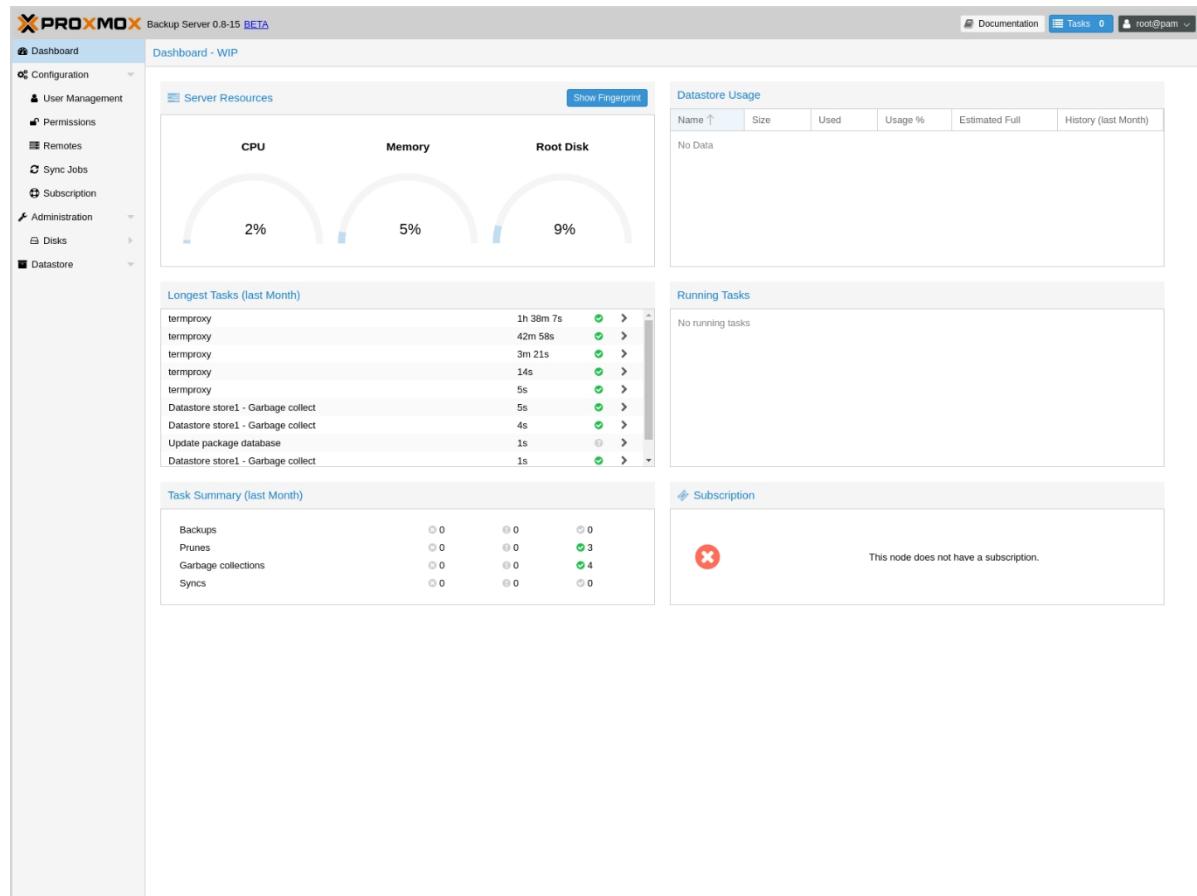
便宜上、ウィンドウ下部の "Save User name" チェックボックスを選択することで、クライアント側でユーザ名を保存することができます。

4.2.1 同意バナー

ログイン前に承諾する必要があるカスタム同意バナーは、設定

→ その他 → 一般 → 同意テキスト.コンテンツがない場合、同意バナーは表示されません。テキストは/etc/proxmox-backup/node.cfg設定ファイルにbase64文字列として保存されます。

4.3 GUIの概要



Proxmox Backup ServerのWebインターフェースは、3つの主要セクションで構成されています：

- ヘッダー：上部。ここにはバージョン情報が表示され、ドキュメンテーションの表示、実行中のタスクの監視、言語設定、各種表示設定、ログアウトのボタンがあります。
- サイドバー：左側。サーバーの管理オプションが含まれます。
- 設定パネル：中央。サイドバーの管理オプションの各コントロールインターフェイスが含まれます。

4.4 サイドバー

ページの左側にあるサイドバーには、特定の管理活動に関連するさまざまな項目が表示されます。

4.4.1 ダッシュボード

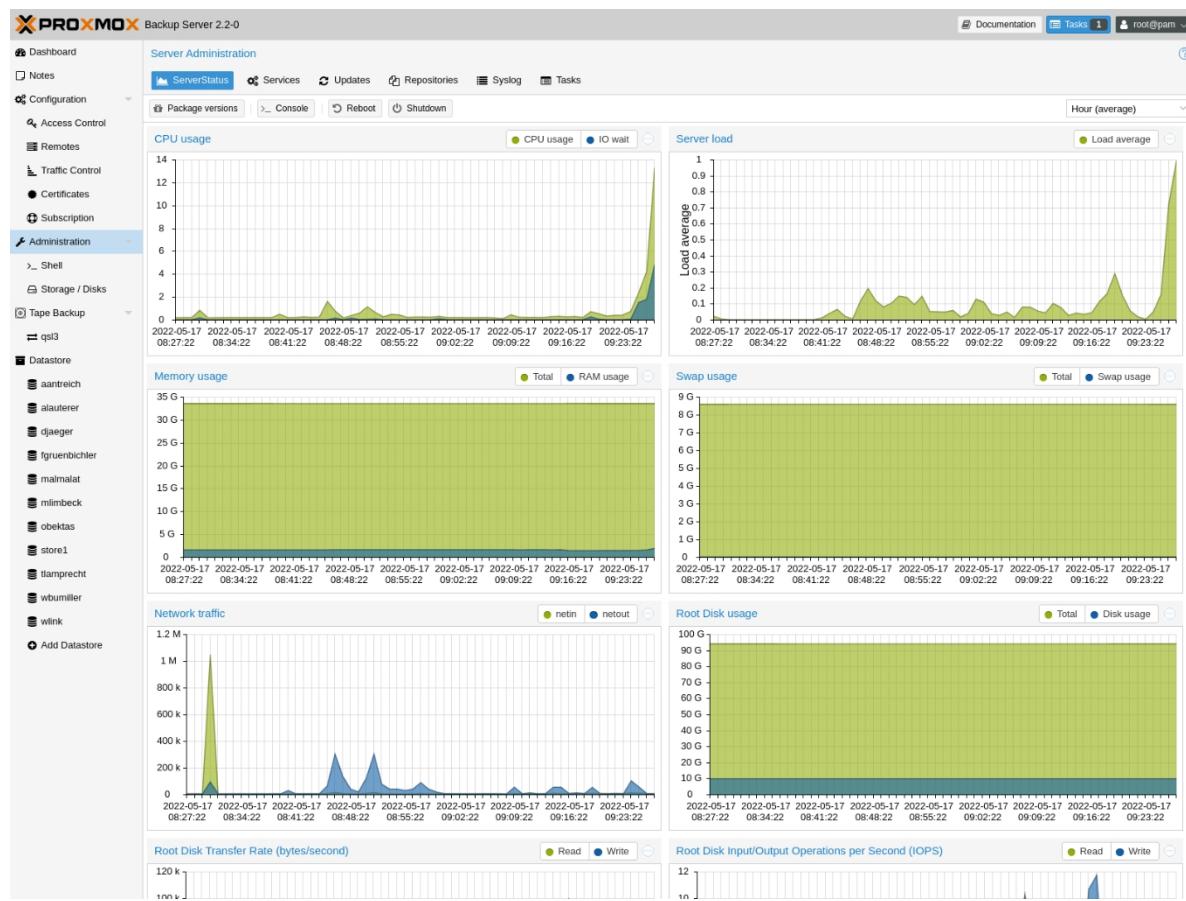
ダッシュボードには、サーバーのアクティビティとリソースの使用状況の概要が表示されます。具体的には、ハードウェアの使用状況、以前および現在実行中のタスクのサマリー、サブスクリプション情報が表示されます。

4.4.2 設定

設定セクションには、時間、ネットワーク、WebAuthn、HTTP プロキシ設定などのシステム オプションが含まれます。また、以下のサブセクションもあります：

- アクセス制御：ユーザー、API トークン、およびこれらのアイテムに関連付けられたパーミッションの追加と管理
- リモート：リモートの追加、編集、削除（[リモート](#)参照）
- 証明書ACMEのアカウントを管理し、SSL証明書を作成します。
- サブスクリプション：サブスクリプションキーのアップロード、サブスクリプションステータスの表示、テキストベースのシステムレポートへのアクセス。

4.4.3 管理



管理セクションにはトップパネルがあり、さらなる管理タスクと情報があります。以下はその例です：

- サーバーステータス：コンソール、電源オプション、およびさまざまなリソース使用統計へのアクセスを提供します。
- サービス：システムサービスの管理と監視
- **Updates**: アップデート：パッケージをアップグレードするためのインターフェイス

- リポジトリAPTリポジトリを設定するためのインターフェース
- システムログ：サーバーからのログメッセージの表示
- タスク複数のフィルターオプションによるタスク履歴

Device	Type	Usage	Size	GPT	Model	Serial	S.M.A.R.T.	M...	Wearout
/dev/sda	Hard Disk	mounted	16.00 TB	No	ST1600NM001G-2KK103	ZL21WKQX	passed	No	N/A
/dev/sdb	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22H7VG	passed	No	N/A
/dev/sdc	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL223443	passed	No	N/A
/dev/sdd	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22NCFQ	passed	No	N/A
/dev/sde	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22LHSF	passed	No	N/A
/dev/sdf	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22B4HK	passed	No	N/A
/dev/sdg	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22DBA5	passed	No	N/A
/dev/sdh	SSD	mounted	120.03 GB	Yes	INTEL_SSDSC2BB120G4	PHWL5352024K120LGN	passed	No	2%
/dev/sdh3	partition	LVM	119.50 GB	Yes				No	N/A
/dev/sdh1	partition	BIOS boot	1.03 MB	Yes				No	N/A
/dev/sdh2	partition	EFI	596.87 MB	Yes				Yes	N/A

管理メニューには、ディスク管理のサブセクションもあります：

- ディスク：利用可能なディスクに関する情報の表示
 - ディレクトリ：ext4 および xfs ディスクの作成と情報の表示
 - ZFS**：ZFSディスクの作成と情報の表示

4.4.4 テープバックアップ

The screenshot shows the Proxmox Tape Backup interface. On the left, a sidebar lists various management sections like Dashboard, Configuration, Access Control, Remotes, Subscriptions, Administration, Shell, Storage/Disks, Tape Backup (which is selected), Datastore, pve-backup, remote-sync, and Add Datastore. The main area has three tabs: Slots, Drives, and Import-Export Slots.

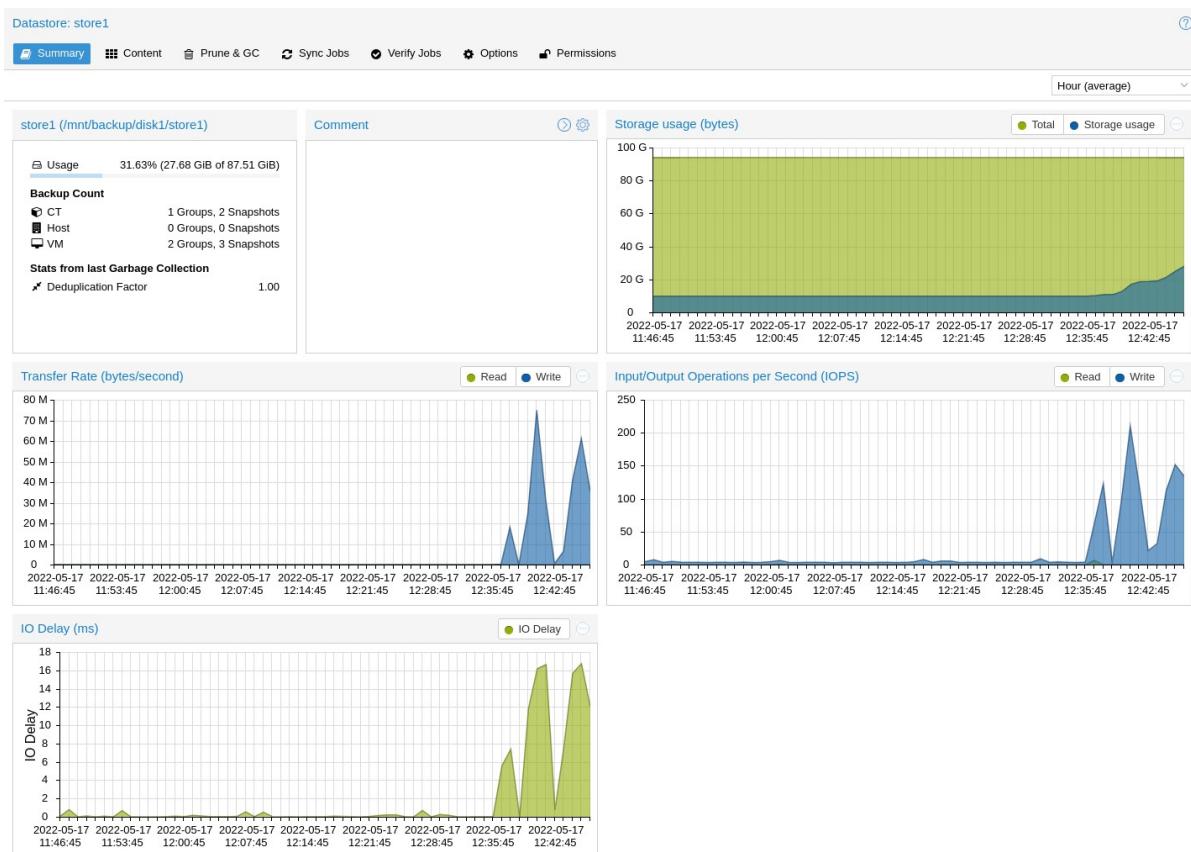
- Slots:** A table with columns ID, Content, Inventory, and Actions. It lists slots from 1 to 20. Most slots (1-19) have "TAPESL4" as content and "writable (pve-backup)" as inventory. Slot 20 is empty. Actions for each slot include a trash bin icon.
- Drives:** A table with columns Content, Inventory, Name, and State. It shows two drives: qdrive0 (idle) and qdrive1 (idle). Both are labeled "writable (pve-backup)".
- Import-Export Slots:** A table with columns ID, Content, Inventory, and Actions. It lists slots 21, 22, 23, and 24, all of which are empty.

テープ・バックアップ・セクションには、テープ・メディア・セット、インベントリ、ドライブ、チェンジャー、暗号化キー、およびテープ・バックアップ・ジョブ自体を管理するためのオプションを備えたトップ・パネルがあります。タブは以下のとおりです：

- コンテンツ：テープバックアップの内容に関する情報
- インベントリ：システムに接続されているテープの管理
- チェンジャー：テープ・ローディング・デバイスの管理
- ドライブ：テープの読み書きに使用するドライブを管理します。
- メディア・プール：テープの論理プールを管理
- 暗号化キー：テープバックアップ暗号化キーの管理
- バックアップジョブ：テープ・バックアップ・ジョブの管理

このセクションには、スタンドアロン・ドライブとチェンジャごとのサブセクションもあり、これらのデバイスのステータスと管理ビューが表示されます。

4.4.5 データストア



データストアセクションには、データストアを作成および管理するためのインターフェースが含まれます。このセクションには、サーバ上に新しいデータストアを作成するためのボタンと、システム上の各データストアのサブセクションがあり、トップパネルを使用して表示することができます：

- **Summary** (サマリー)：データストアの使用状況に関するさまざまな統計情報にアクセスできます。
- 内容：コンテンツ：データストアのバックアップグループとそれぞれのコンテンツに関する情報
- **Prune & GC**：プルーニングとガベージコレクション操作をスケジュールし、手動でガベージコレクションを実行します。
- 同期ジョブ：リモートサーバからの[同期ジョブ](#)の作成、管理、実行
- 検証ジョブ：データストアの[検証ジョブ](#)の作成、管理、実行
- オプション通知と検証の設定
- 権限：データストアの権限管理

第5章

バックアップ・ストレージ

5.1 ディスク管理

Device	Type	Usage	Size	GPT	Model	Serial	S.M.A.R.T.	M...	Wearout
/dev/sdb	Hard Disk	mounted	16.00 TB	No	ST1600NM001G-2KK103	ZL21WKQX	passed	No	N/A
/dev/sdb	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22H7VG	passed	No	N/A
/dev/sdc	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL223443	passed	No	N/A
/dev/sdd	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22NCFQ	passed	No	N/A
/dev/sde	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22LHSF	passed	No	N/A
/dev/sdf	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22B4HK	passed	No	N/A
/dev/sdg	Hard Disk	filesystem	16.00 TB	No	ST1600NM001G-2KK103	ZL22DBA5	passed	No	N/A
/dev/sdh	SSD	mounted	120.03 GB	Yes	INTEL_SSDSC2BB120G4	PHWL5352024K120LGN	passed	No	2%
/dev/sdh3	partition	LVM	119.50 GB	Yes				No	N/A
/dev/sdh1	partition	BIOS boot	1.03 MB	Yes				No	N/A
/dev/sdh2	partition	EFI	536.87 MB	Yes				Yes	N/A

Proxmox Backup Serverにはディスクユーティリティのセットが付属しており、ディスクサブコマンドまたはWebインターフェイスを使用してアクセスします。このサブコマンドを使用すると、ディスクの初期化、さまざまなファイルシステムの作成

を作成したり、ディスクに関する情報を取得したりできます。

システムに接続されているディスクを表示するには、ウェブインターフェイスの**Administration -> Storage/Disks**に移動するか、**disk**のlistサブコマンドを使用します：

```
# proxmox-backup-manager disk list
```

名前	使用	gpt	ディスクタイプ	サイズ	モデル	wearout	ステータス
sda	lvm	1	hdd	34359738368	QEMU_HARDDISK -	パス済み	
sdb	未使用 1		hdd	68719476736	QEMU_HARDDISK -	パス済み	

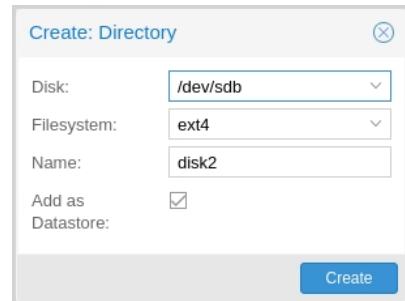
(次のページに続きます)

SDC 未使用 1	HDD	68719476736 q e m u _ h a r d d i s k	- パス
-------------	-----	---	------

(前ページから続く)

新しいGPTでディスクを初期化するには、**initialize**サブコマンドを使います：

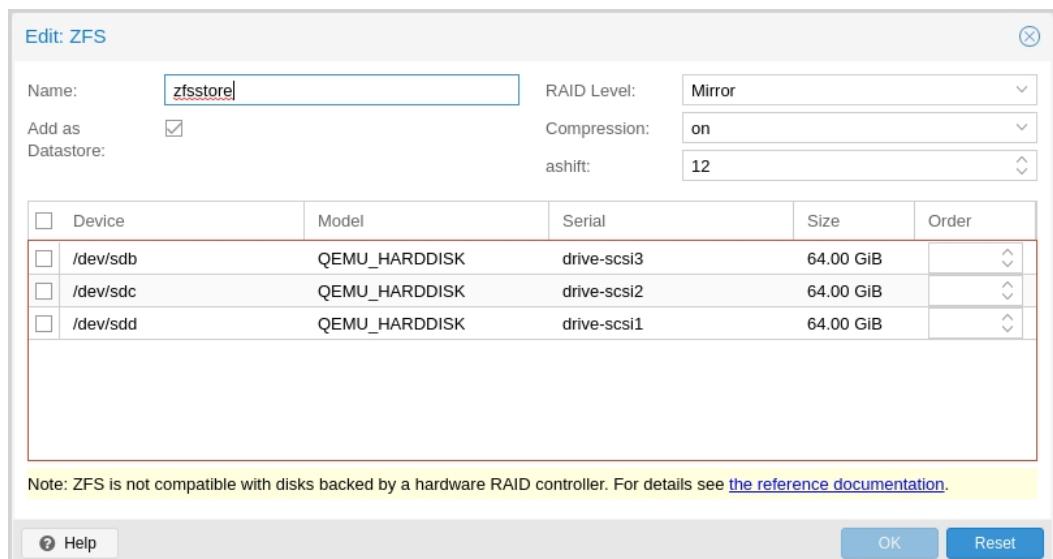
```
# proxmox-backup-manager disk initialize sdX
```



ディスク上に ext4 または xfs ファイルシステムを作成するには **fs create** を使用するか、ウェブインターフェイスの **Administration -> Storage/Disks -> Directory** に移動して作成します。以下のコマンドは ext4 ファイルシステムを作成し、--add-datastore パラメータを渡してディスク上にデータストアを自動的に作成します。これにより、以下の場所にデータストアが作成されます。

/mnt/datastore/store1 :

```
# proxmox-backup-manager disk fs create store1 --disk sdX --filesystem ext4 --add-datastore  
→true
```



Note: ZFS is not compatible with disks backed by a hardware RAID controller. For details see [the reference documentation](#).

ウェブインターフェイスの**Administration -> Storage/Disks -> ZFS**から、またはzpool createを使用して、様々な RAIDレベルのzpoolを作成することもできます。以下のコマンドは2つのディスクを使ってミラー化されたzpoolを作成し、/mnt/datastore/zpool1の下にマウントします：

```
# proxmox-backup-manager disk zpool create zpool1 --devices sdX,sdY --raidlevel mirror
```



ここで--add-datastoreパラメータを渡すと、ディスクからデータストアが自動的に作成されます。

disk fs listとdisk zpool listを使用して、それぞれファイルシステムとzpoolを追跡できます。

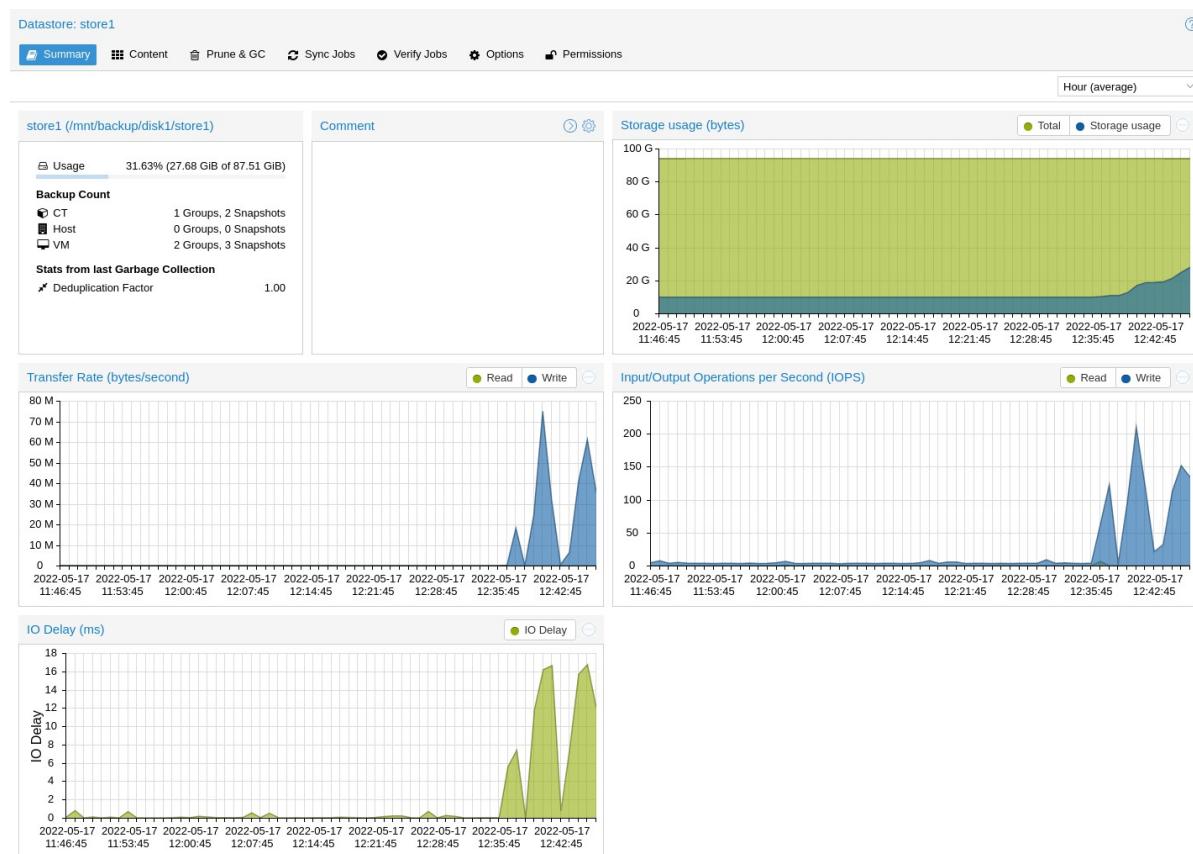
Proxmox Backup Serverはsmartmontoolsパッケージを使用しています。これは、ローカルハードディスクのS.M.A.R.T.システムを監視および制御するためのツールセットです。ディスクがS.M.A.R.T.機能をサポートしており、これを有効にしている場合、Webインターフェイスから、またはコマンドを使用してS.M.A.R.T.属性を表示できます：

以下のコマンドを使用します。 # proxmox-backup-manager disk smart-attributes sdX

注記

この機能は、smartmontools パッケージに含まれる smartctl コマンドを使用して直接アクセスすることもできます（詳細は man smartctl を参照）。

5.2 データストア



データストアは、バックアップが保存される場所を指します。現在の実装では、標準的なUnixファイルシステム（ext4、xfs、またはzfs）内のディレクトリを使用してバックアップデータを格納します。

データストアは単純なIDで識別されます。これは、データストアを設定するときに構成できます。データストアの設定情報は/etc/proxmox-backup/datasstore.cfgファイルに保存されます。

注

ファイルレイアウトでは、ファイルシステムがディレクトリごとに少なくとも65538個のサブディレクトリをサポートする必要があります。

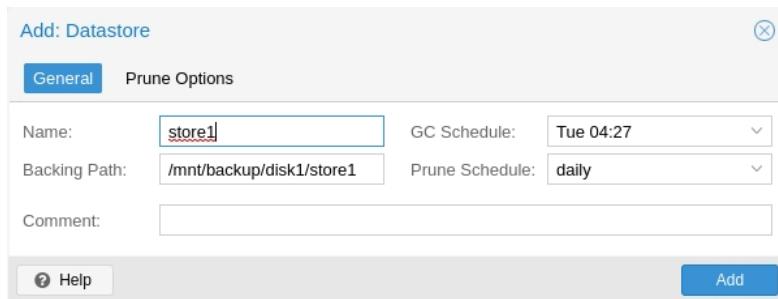
この数は、 2^{16} 個のチャンク・ネームスペース・ディレクトリと、.および...のデフォルト・ディレクトリ・エントリによって事前に作成されます。この要件により、特定のファイルシステムやファイルシステム構成がデータストアでサポートされなくなります。たとえば、ext3全体やext4でdir_nlink機能が手動で無効になっている場合などです。

5.2.1 データストアの構成

Backup Group ↑	Comment	Actions ↑	Backup Time ↑	Size	Count	Owner	Encrypted	Verify State
Root Namespace			2022-05-17 12:41:54		2	root@pam	No	All OK
vm/115					1			None
vm/118								
dc-eu-central								
dc-eu-west								
dc-oceania								
dc-singapore								
dc-usa								

複数のデータストアを構成できます。最低1つのデータストアを構成する必要があります。データストアは単純な名前で識別され、ファイルシステム上のディレクトリを指します。各データストアには、時間毎、日毎、週毎、月毎、年毎のバックアップ・スナップショット数と、そのストアに保存するバックアップの時間非依存数が設定されます。ストアに保持するバックアップ数です。ブルーニングと削除バックアップとカベージコレクションを実行するように設定することもできます。データストアごとに設定されたスケジュール（「カレンダー・イベント」参照）に基づいて、定期的に実行されます。

データストアの作成



Webインターフェイスから新しいデータストアを作成するには、サイド・メニューの [データストア] セクションで [データストアの追加] をクリックします。設定ウィンドウで

- 「名前」はデータストアの名前です。
- 「Backing Path」は、データストアを作成するディレクトリへのパスです。
- 「GC スケジュール」は、ガベージ・コレクションを実行する時間と間隔です。
- 「プルーンスケジュール」とは、プルーニングを実行する頻度のことです。
- 「Prune Options (プルーニング・オプション)」は、保持するバックアップの量を設定します（[バックアップのプルーニングと再移動を参照](#)）。
- 「Comment (コメント)」を使用して、データストアにコンテキスト情報を追加できます。

または、コマンドラインから新しいデータストアを作成することもできます。以下のコマンドは /backup/disk1/store1 に store1 という新しいデータストアを作成します。

```
# proxmox-backup-manager datastore create store1 /backup/disk1/store1
```

リムーバブルデータストア

リムーバブルデータストアはバックティングデバイスが関連付けられており、マウントとアンマウントが可能です。それ以外は通常のデータストアと同じように動作します。

通常のデータストアと同様に、ext4 または xfs で正しくフォーマットされたパーティションに作成できます。のいずれかでなければなりませんが、Proxmox Linuxカーネルがサポートするほとんどの最新のファイルシステムがサポートするほとんどの最新ファイルシステムで動作するはずです。

注意

FATベースのファイル・システムはPOSIXファイル所有権の概念をサポートしておらず、ディレクトリあたりのファイル数の制限も比較的緩やかです。したがって、データストアの作成はFATファイルシステムではサポートされません。外付けドライブの中には、このようなFATベースのファイルシステムで事前にフォーマットされているものがあるため、リムーバブル・データストアのバックティング・デバイスとして使用する前に、ドライブを再フォーマットする必要があります。

管理] > [ディスク / ストレージ] > [ディレクトリ] を使用して、完全に未使用的ディスクに作成することもできます。この方法を使用すると、ディスクはデータストア用に自動的にパーティション設定およびフォーマットされます。

データストアが1つしかないデバイスは自動的にマウントされます。アンマウントは、サマリーページで "Unmount" をクリックするか、CLI を使用して UI から行う必要があります。アンマウントに失敗すると、その理由がアンマウントタスクログに記録され、データストアはメンテナンスマードになります。

になり、IO 操作ができなくなります。このような場合、メンテナンスマードを次のように手動でリセットする必要があります。を使って手動でリセットする必要があります：

```
# proxmox-backup-manager datastore update --maintenance-mode offline
```

を使ってIOを防ぐか、クリアしてください：

```
# proxmox-backup-manager datastore update --delete maintenance-mode
```

1つのデバイスに複数のデータストアを格納することができます。

リムーバブル・データストアは、作成時に指定された相対パスのデバイス上に作成されます。

のデバイス上に作成されます。複数のPBSインスタンスでデータストアを使用するには、1つのインスタンスでデータストアを作成し、他のインスタンスではReuse existing datastoreにチェックを入れて追加する必要があります。作成時に設定したパスは、1つのデバイス上の複数のデータストアを識別する方法です。そのため、新しいPBSインスタンスに追加するときは

作成時に設定したパスと一致する必要があります。

```
# proxmox-backup-manager datastore unmount store1
```

は実行中のタスクの終了を待ってデバイスをアンマウントします。

すべてのリムーバブルデータストアは /mnt/datastore/<name> の下にマウントされ、指定されたパスはデバイス上のパスを参照します。

デバイス上に存在する全てのデータストアは proxmox-backup-debug を使って一覧できます。

```
# proxmox-backup-debug inspect device /dev/...
```

リムーバブルデータストアがマウントされていない場合、ベリファイ、プルーン、およびガベージコレクションのジョブはスケジュール時にスキップされます。同期ジョブは開始されますが、データストアがマウントされていないというエラーで失敗します。スケジュールどおりに同期が行われないことは、少なくとも目につくはずだからです。

S3バックエンドを持つデータストア

Proxmox Backup Serverは、データストアのストレージバックエンドとしてS3互換オブジェクトストアをサポートしています。このためには、"Configuration">"Remotes">"S3 Endpoints"でS3エンドポイントを設定する必要があります。

■ 重要

S3データストア・バックエンドは現在テクノロジー・プレビューです。

■ 重要

S3 バックアップオブジェクトストアとして運用すると、追加コストが発生する可能性があることに留意してください。プロバイダは、ストレージスペースとバケットへのAPIリクエスト、イグレスと帯域幅の料金を請求するかもしれません。したがって、これらの値と最終的なコストを監視することを強くお勧めします。

エンドポイント設定では、オブジェクトストアの REST API エンドポイントを指定します。

ポイントはプロバイダに依存し、バケットとリージョンのテンプレート化が可能です。たとえば、エンドポイントを {{bucket}}.s3.{{region}}.amazonaws.com と設定すると、リージョン eu-central-1 にある my-pbs-bucket という名前のバケットを設定した場合、my-pbs-bucket.s3.eu-central-1.amazonaws.com に拡張されます。

同じエンドポイントが複数のバケットで再利用される可能性があるためです。そのため、同じバケットを使って複数のデータストアを作成することも可能です。

■ 注意

Proxmox Backup Serverはバケットの作成とアクセス制御を行いません。データストアのオブジェクトを格納するバケットとアクセスキーは、S3プロバイダのインターフェイスで事前に設定する必要があります。Proxmox Backup Serverはクライアントとして動作し、バケット内のオブジェクトを取得、リストアップ、および削除するためのパミッションが必要です。

ほとんどのプロバイダは、バケツ名をエンドポイントアドレスの一部とする vhost 形式のアドレス指定か、バケツ名をリクエストのパスコンポーネントのプレフィックスとするパス形式のアドレス指定でバケツにアクセスすることができます。Proxmox Backup Server は両方のスタイルをサポートしていますが、vhost スタイルの URL をパススタイルよりも優先しています。パス形式のアドレスを使用するには、対応する設定フラグを設定します。

Proxmox Backup ServerはS3 APIとのプレーンテキスト通信をサポートしていません。したがって、自己署名証明書を使用するセルフホストS3オブジェクトストアでは、一致するフィンガープリントをエンドポイント設定に提供する必要があります。そうしないと、クライアントはS3オブジェクトストアへの接続を拒否します。

次の例は新しい s3 エンドポイント設定のセットアップを示しています：

```
# proxmox-backup-manager s3 endpoint create my-s3-ep --access-key 'my-access-key' --secret-key 'my-secret-key' --endpoint '{{bucket}}.s3.{{region}}.amazonaws.com' --region eu-central-1
```

s3エンドポイントの設定を一覧表示するには、以下を実行してください：

```
# proxmox-backup-manager s3 endpoint list
```

S3バックエンドを持つ新しいデータストアは、設定されたS3エンドポイントのいずれかを使用して作成できます。すべてのコンテンツをS3オブジェクトストアに保存しますが、データストアは、パフォーマンスを向上させ、バックエンドへのリクエスト数を減らすために使用される、ローカルのサイサントごとのキャッシュを必要とします。このために、通常のデータストアのセットアップと同様に、データストア作成時にローカルファイルシステムパスを提供する必要があります。ただし、通常のデータストアとは異なり、ローカルキャッシュのサイズには制限があります。キャッシュされたデータストアのコンテンツにはデータチャunkも含まれるため、64GiBから128GiBを推奨します。最適なのは、専用のディスク、パーティション、またはクオータ付きのZFSデータセットをローカルキャッシュとして使用することです。ただし、既存の通常のデータストアを使用することはできません。さらに、キャッシュに揮発性メモリのみを使用することは、現在のところできません。

`pbs-s3-bucket`というバケットに`my-s3-store`という新しいデータストアをセットアップするには、以下を実行します：

```
# proxmox-backup-manager datastore create my-s3-store /mnt/datastore/my-s3-store-cache ->backend type=s3,client= my-s3-ep,bucket= pbs-s3-bucket
```

データストアは複数のProxmox Backup Serverインスタンス間で共有することはできません。ただし、使用できなくなつたインスタンスで使用されていたデータストアのコンテンツは、新しいインストールで再利用できます。データストアを再作成するには

フラグを渡す必要があります。データストア名は
をプレフィックスとして使用する場合は、同じデータストア名を使用する必要があります。

```
# proxmox-backup-manager datastore create my-s3-store /mnt/datastore/my-new-s3-store-cache ->backend type=s3,client= my-s3-ep,bucket pbs-s3-bucket= --reuse-datastore true --overwrite-in-use true
```

注

S3オブジェクトストアに書き込み操作を行っている間に容量不足になった場合、スナップショットディレクトリ内のコンテンツのクリーンアップなどのクリーンアップ操作も失敗する可能性が高いです。推奨される手順は、S3オブジェクトストア上のこのスナップショットに対応する野良オブジェクトを手動でクリーンアップすることです。

を手動でクリーンアップし、CLIまたはUI経由でS3リフレッシュを介してコンテンツをリフレッシュすることです。

UIを使用します。

5.2.データストア

S3データストア・バックエンドの構成例

以下は、`/etc/proxmox-backup/s3.cfg`と`/etc/proxmox-backup/datasstore.cfg`からの抜粋（S3設定に関連するデータのみ）として、いくつかの典型的なS3オブジェクトストア・プロバイダの設定例を示しています：

プレーンIPアドレス、カスタムポート、自己署名証明書、パススタイルのバケットを使用した、Ceph Rados Gatewayを使用した自己ホスト型S3オブジェクトストア：

```
# cat /etc/proxmox-backup/s3.cfg s3-  
endpoint: ceph-s3-rados-gw  
access-key XXXXXXXXXXXXXXXXXXXXXXX endpoint  
172.16.0.200 fingerprint _____  
  
→XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:  
パススタイルtrue ポート  
7480  
シケレットキー XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
# cat /etc/proxmox-backup/datasstore.cfg データストア: ceph-s3-rgw-  
store  
バックエンドバケット:pb=ceph-bucket,client=ceph-s3-rados-gw,type=s3パス  
/mnt/datasstore/ceph-s3-rgw-store-local-cache
```

AWS S3 で vhost スタイルのバケットアドレス指定で、バケット名とリージョンをエンドポイント URL のテンプレートとして使用します：

```
# cat /etc/proxmox-backup/s3.cfg s3-  
endpoint: aws-s3  
    アクセスキー XXXXXXXXXXXXXXXXXXXXXXXX  
    エンドポイント {{バケット}}.s3.{{リージョン}}.amazonaws.com リージョン  
eu-central-1  
    シークレットキー XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
  
# cat /etc/proxmox-backup/datastore.cfg  
    backend bucket=pbs-s3-bucket,client=aws-s3,type=s3 path  
データ /mnt/datastore/s3-store-s3-store-local-cache
```

Cloudflare R2 のバケットアドレスはパス形式になっています：

```
# cat /etc/proxmox-backup/s3.cfg s3-  
endpoint: cloudflare-r2  
    アクセスキー XXXXXXXXXXXXXXXXXXXXXXXXX  
    エンドポイント XXXXXXXXXXXXXXXXXXXXXXXXX.r2.cloudflarestorage.com パススタイル true  
    リージョン auto  
    secret-key XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
  
# cat /etc/proxmox-backup/datastore.cfg  
    backend bucket=pbs-r2-bucket,client=cloudflare-r2,type=s3 path /mnt/datastore/r2-  
    s3-store-local-cache  
データストア: r2-s3-store
```

データストアの管理

コマンドラインから既存のデータストアを一覧表示するには、以下を実行します：

# proxmox-backup-manager データストア リスト		
名前	パス	コメント
store1	/backup/disk1/store1	これは私のデフォルトストレージです。

データストアのガベージ・コレクションおよびプルーンの設定は、GUIからデータストアを編集するか、`update`サブコマンドを使用して変更できます。例えば、以下のコマンドは`update`サブコマンドでガベージコレクションのスケジュールを変更し、`show`サブコマンドでデータストアのプロパティを表示します：

```
# proxmox-backup-manager datastore update store1 --gc-schedule 'Tue 04:27'  
# proxmox-backup-manager datastore show store1
```

名前	値
名前	store1

(次のページに続きます)

パス	//バックアップ/ディスク1/ストア1
コメント	これは私のデフォルトのストレージです。
gcスケジュール	火 04:27
キープラスト	7
剪定スケジュール	毎日

(前ページから続く)

最後に、データストアの設定を削除します：

```
# proxmox-backup-manager datastore remove store1
```

注意

上記のコマンドはデータストアの設定のみを削除します。基礎となるディレクトリのデータは削除されません。

ファイルのレイアウト

データストアを作成すると、以下のデフォルトのレイアウトが表示されます：

```
# ls -alrh /backup/disk1/store1 276493 -rw-
r--r-- 1 backup バックアップ
0 Jul 8 12:35 .lock
276490 drwxr-x--- 1 backup backup 1064960 Jul 8 12:35 .chunks
```

.lock はプロセスのロックに使用される空のファイルです。

.chunksディレクトリには、0000から始まりfffffまで16進数で増加するフォルダが含まれます。これらのディレクトリには、バックアップ操作の実行後に、チェックサムごとに分類されたチャンクデータが格納されます。

```
# ls -alrh /backup/disk1/store1/.chunks 545824 drwxr-x--- 2
backup backup 4.0K Jul 415621 drwxr-x--- 2 backup backup 4.0K Jul 415620
drwxr-x--- 2 backup backup 4.0K Jul 353187 drwxr-x--- 2
backup backup 4.0K Jul 344995 drwxr-x--- 2 backup backup 4.0K Jul 144078
drwxr-x--- 2 backup backup 4.0K Jul
(1).4. 4077 drwxr-x--- 2 backup backup 4.0K Jul
403180 drwxr-x--- 2 backup backup 4.0K Jul 403179 drwxr-x---
- 2 backup backup 4.0K Jul 403177 drwxr-x--- 2 backup
backup 4.0K Jul 402530 drwxr-x--- 2 backup backup 4.0K Jul
402513 drwxr-x--- 2 backup backup 4.0K Jul 402509 drwxr-x---
- 2 backup backup 4.0K Jul 276509 drwxr-x--- 2 backup
backup 4.0K Jul 276508 drwxr-x--- 2 backup backup 4.0K Jul
276507 drwxr-x--- 2 backup backup 4.0K Jul 276501 drwxr-x---
- 2 backup backup 4.0K Jul 276499 drwxr-x--- 2 backup
backup 4.0K Jul 276498 drwxr-x--- 2 backup backup 4.0K Jul
276494 drwxr-x--- 2 backup backup 4.0K Jul 276489 drwxr-x---
x 3 backup backup 4.0K Jul 276490 drwxr-x--- 1 backup
backup 1.1M Jul
8 12:35 ffff
8 12:35 fffe
8 12:35 fffd
8 12:35 ffcc
8 12:35 ffbb
8 12:35 ffda
8 12:35 ff99
8 12:35 ff88
8 12:35 fff7
8 12:35 000c
8 12:35 000b
8 12:35 000a
8 12:35 0009
8 12:35 0008
8 12:35 0007
8 12:35 0006
8 12:35 0005
8 12:35 0004
8 12:35 0003
8 12:35 0002
8 12:35 0001
8 12:35 0000
8 12:35 ..
8 12:35 .
```

バックアップのアップロードまたはネームスペースの作成が完了すると、バックアップ・タイプ (ct, vm, host) とネームスペース階層の開始 (ns) が表示されます。

5.2.2 バックアップ・ネームスペース

データストアは、基礎となるストレージが十分に大きく、ユーザのユースケースに必要なパフォーマンスを提供する限り、多くのバックアップをホストすることができます。しかし、階層化や分離を行わないと、特に同じデータストアを使

用する場合、名前の衝突が発生しやすくなります。

同じデータストアを複数のProxmox VEインスタンスや複数のユーザに使用する場合は特にそうです。

バックアップ名前空間階層を使用すると、一般的に異なるユーザまたはバックアップソースを明確に分けることができ、名前の競合を回避し、整理されたバックアップコンテンツビューを提供します。

各ネームスペース・レベルは、任意のバックアップ・タイプ、CT、VM、ホストをホストすることができます。

ネームスペースの権限

特定のネームスペースにのみパーミッションを設定することで、データストアのパーミッション構成をより細かくすることができます。

データストアを表示するには、少なくともAUDIT、MODIFY、READ、またはBACKUP権限を持つ権限が必要です。権限が必要です。

ネームスペースを作成または削除するには、親ネームスペースの変更権限が必要です。したがって、最初にネームスペースを作成するには、データストア自体の MODIFY 特権を含むアクセス・ロールを持つ権限が必要です。

バックアップ・グループについては、既存の権限ルールが引き続き適用されます。十分な権限を持つか、バックアップ・グループの所有者になる必要があります。

5.2.3 オプション

Datastore: store1	
	⑦
Summary	Content
Prune & GC	Sync Jobs
Verify Jobs	Options
Permissions	
Edit	
Notify	Verify=Always, Sync=Always, GC=Always, Prune=Always
Notify User	root@pam
Verify New Snapshots	No
Maintenance mode	None
Tuning Options	Chunk Order: Default (None), Sync Level: Default (Filesystem)
	Remove Datastore

データストアごとにいくつかのオプションがあります：

- 通知モードとレガシー通知設定

- メンテナンスマード
- 受信バックアップの検証

チューニング

データストアには、より高度なチューニング関連のオプションがいくつかあります：

- チャンク・オーダー：ベリファイとテープバックアップのチャンク順序：

Proxmox Backup Serverがベリファイやテープへのバックアップを行う際にチャンクを反復処理する順序を指定できます。2つのオプションがあります：

 - inode*（デフォルト）：inode（デフォルト）：ファイルシステムのinode番号順にチャンクを並べ替えてから繰り返し処理します。ほとんどのストレージ、特に回転ディスクではこれで問題ないでしょう。
 - none* インデックスファイル (.fidx/.didx) に表示されている順番にチャンクを反復します。多くの低速ストレージでは反復処理が遅くなるかもしれません、非常に高速なストレージ（例えば、NVME）では、収集とソートには、ソートされた反復処理で得られる時間よりも多くの時間がかかる可能性があります。このオプションは

```
# proxmox-backup-manager datastore update <storename> --tuning 'chunk-order=none' で設定できます。
```

- sync-level：データストアの同期レベル：

データストアのチャンクの同期レベルを設定できます。これは電源喪失やハードシャットオフ時のバックアップのクラッシュ耐性に影響します。現在3つのレベルがあります：

- none*：チャンク書き込み時に同期を行いません。カーネルは最終的にディスクへの書き込みをフラッシュするので、これは高速で通常は問題ありません。カーネル sysctls の *dirty_expire_centisecs* と *dirty_writeback_centisecs* を使って動作を調整しますが、デフォルトでは ~30 秒後に古いデータをフラッシュします。
- ファイルシステム（デフォルト）：バックアップの後、タスクが *OK* を返す前に *syncfs(2)* をトリガします。こうすることで、書き込まれたバックアップがディスク上にあることが保証されます。これは速度と一貫性の良いバランスです。基礎となるストレージデバイスは内部エフェメラルキャッシュをパーマネントストレージレイヤーにフラッシュするための電力損失から自身を保護する必要があることに注意してください。
- ファイルこのモードでは、チャンクの挿入ごとに *fsync* がトリガーされ、各チャンクができるだけ早くディスクに到達するようにします。このモードは最高レベルの一貫性を実現しますが、多くのストレージ（特に低速のストレージ）にとっては速度と引き換えになります。多くのユーザーにとってはファイルシステムモードの方が適していますが、非常に高速なストレージではこのモードでも問題ありません。

これは

```
# proxmox-backup-manager datastore update <storename> --tuning 'sync-level=filesystem'
```

- gc-atime-safety-check*：データストアのGC atime更新安全性チェック：データストアの作成とガベージ・コレクションで実行される *atime update safety check* を明示的に有効または無効にできます。これは、ガベージ・コレクションによって *atime* 更新が期待どおりに処理されるかどうかをチェックします。
ファイルシステムの予期しない動作によるデータ損失のリスクを回避します。これはデフォルト値でもあります。
- gc-atime-cutoff*：チャンク・クリーンアップのためのデータストアGCの *atime* カットオフ：これは、ガベージコレクションのフェーズ2の間、チャンクがまだ使用中とみなされるカットオフを設定することができます（古い書き込みがない場合）。チャンクの *atime* が範囲外の場合、チャンクは削除されます。
- gc-cache-capacity*：データストアGCの最近使用されたキャッシュ容量：ガベージ・コレクションの第1フェーズでアクセス・タイムがすでに更新されたチャンクを追跡するために使用されるキャッシュ容量を制御できます。

ガベージ・コレクションのフェーズ 1 で更新されたチャンクを追跡するために使用されるキャッシュ容量を制御できます。これにより、複数の更新が回避され、GC の実行時間のパフォーマンスが向上します。値を高くすると、メモリ使用量の増加と引き換えに GC の実行時間を短縮できます。指定された値は、利用可能なキャッシュ・スロットの数を設定します。

1048576 (= 1024 * 1024) がデフォルトで、8388608 (= 8192 * 1024) が最大値です。

複数のチューニングオプションを同時に設定したい場合は、次のようにカンマで区切れます：

```
# proxmox-backup-manager データストア更新<ストア名> --tuning 'sync-level=filesystem,chunk-<br/>'<br/>'order=none'
```

5.3 ランサムウェアの保護と復旧

ランサムウェアは、身代金を支払うまでファイルを暗号化するマルウェアの一種です。Proxmox Backup Serverは、オフサーバおよびオフサイト同期とバックアップからの簡単な復元を提供することで、ランサムウェア攻撃を軽減し、回復するのに役立つ機能を備えています。

5.3.1 ビルトインの保護

Proxmox Backup Serverは既存のブロックのデータを書き換えません。これは、Proxmox VEホストまたはクライアントを使用してデータをバックアップする他の侵害されたシステムが、既存のバックアップを破損または変更できないことを意味します。

5.3.2 Proxmox Backup Serverの3-2-1ルール

3-2-1ルールはシンプルですが、火災や自然災害、敵によるインフラへの攻撃など、あらゆる脅威から重要なデータを守るために効果的です。つまり、少なくとも2つの異なるタイプのストレージメディアに3つのバックアップを作成し、そのうち1つのコピーをオフサイトに保管するというルールです。

Proxmox Backup Serverは、バックアップの追加コピーをリモートロケーションや様々な種類のメディアに保存するためのツールを提供します。

リモートのProxmox Backup Serverを設定することで、[リモート同期ジョブ](#)機能を利用し、バックアップのオフサイトコピーを簡単に作成することができます。オフサイトのインスタンスはローカルネットワーク内のランサムウェアに感染する可能性が低いため、この方法をお勧めします。リモートソース上でスナップショットが消えても削除しないように同期ジョブを設定することで、ソースを乗っ取った攻撃者がターゲットホスト上でバックアップを削除することを防ぐことができます。ソースホストがランサムウェア攻撃の被害に遭った場合、同期ジョブが失敗し、エラー通知がトリガーされる可能性が高くなります。

第二のストレージメディアとして[テープバックアップ](#)を作成することも可能です。この方法では、長期保存用に設計された別の記憶媒体にデータの追加コピーを取得できます。さらに、オフサイトに移動したり、オンラインの耐火保管庫に移動して素早くアクセスすることも可能です。

5.3.3 制限付きユーザーとアクセス管理

Proxmox Backup Serverは、包括的で細かいユーザおよびアクセス管理システムを提供します。たとえば、*Datastore.Backup*権限では、既存のバックアップの作成のみが許可され、削除や変更は許可されません。

このアクセス制御システムを活用する最善の方法は、次のとおりです：

- Proxmox Backup Server にデータをバックアップできるホストまたは Proxmox VE Cluster ごとに個別の API トークンを使用します。
- このような API トークンには最小限の権限のみを設定します。たとえば、*/datastore/tank/pve-abc-cluster* のように、特定のデータストア上の特定のネームスペースに制限された非常に狭い ACL パスで DataStore アクセスロール

を付与する 1 つのパーミッションのみを持つようにします。

ヒント

ランサムウェアから保護するためのベストプラクティスの1つは、削除権限を付与することではなく、Proxmox Backup Serverで剪定ジョブを使用してバックアップの剪定を直接実行することです。

同じことが同期ジョブにも適用されることに注意してください。シンクユーザーまたはアクセストークンの権限を、バックアップの削除ではなく書き込みのみに制限することで、侵害されたクライアントは既存のバックアップを削除できなくなります。

5.3.4 ランサムウェアの検出

Proxmox Backup Serverは、安全でないネットワーク内、サーバーへの物理的なアクセス、脆弱な認証情報、または保護が不十分な認証情報によって危険にさらされる可能性があります。そのような場合、オンラインバックアップがランサムウェアによって暗号化されると、バックアップのSHA-256チェックサムは以前に記録されたものと一致しなくなるため、バックアップの復元は失敗します。

侵害されたゲスト内のランサムウェアを検出するには、バックアップの復元と起動を頻繁にテストすることをお勧めします。必ず新しいゲストにリストアし、現在のゲストを上書きしないようにしてください。バックアップされたゲストが多数ある場合は、このリストアテストを自動化することをお勧めします。これが不可能な場合は、バックアップからランダムなサンプルを定期的(たとえば、週または月に一度)にリストアすることをお勧めします。

ランサムウェア攻撃の際に迅速に対応できるよう、バックアップからの復元を定期的にテストすることをお勧めします。必ず新しいゲストにリストアし、現在のゲストを上書きしないようにしてください。一度に多くのゲストをリストアするのは面倒な場合があります。そのため、このタスクを自動化し、自動化されたプロセスが機能することを検証することをお勧めします。これが実行不可能な場合は、バックアップからランダムなサンプルをリストアすることをお勧めします。バックアップを作成することは重要ですが、バックアップが機能するかどうかを検証することも同様に重要です。これにより、緊急時に迅速に対応することができ、サービスの中止を最小限に抑えることができます。

検証ジョブは、Proxmox Backup Server上のランサムウェアの存在を検出するのにも役立ちます。検証ジョブは、すべてのバックアップが記録されているチェックサムと一致しているかどうかを定期的にチェックするため、ランサムウェアが既存のバックアップを暗号化し始めると失敗し始めます。十分に高度なランサムウェアはこのメカニズムを回避する可能性があることにご注意ください。したがって、検証ジョブはあくまで追加的なものであり、十分な保護手段ではないとお考えください。

5.3.5 一般的な予防方法とベストプラクティス

Proxmox Backup Serverが提供するものとは別に、追加のセキュリティ対策を講じることをお勧めします。これらの推奨事項には以下が含まれますが、これらに限定されません：

- ファームウェアとソフトウェアを常に最新の状態に保ち、エクスプロイトや脆弱性 (SpectreやMeltdownなど) にパッチを適用すること。
- 安全でセキュアなネットワークプラクティスに従うこと。たとえば、ロギングおよび監視ツールを使用すること、VLANを設定するなどして、インフラストラクチャトラフィックとユーザーまたはパブリックトラフィックが分離されるようにネットワークを分割すること。
- 長期保存の設定。ランサムウェアの中には、データを暗号化し始める前に数日から数週間眠っているものがあるため、古い既存のバックアップが危険にさらされる可能性があります。したがって、少なくとも数個のバックアップを長期間保存しておくことが重要です。

ランサムウェア攻撃を回避する方法や、ランサムウェアに感染した場合の対処方法についての詳細は、[CISA \(米国\)](#) の [5.3.ランサムウェアの保護と復旧](#)

ユーザー管理

6.1 ユーザ設定

User name ↑	Realm ↑	Enabled	Expire	Name	Comment
alauterer	pbs	Yes	never		
atwin	pbs	No	never		
cperez	pbs	Yes	never		
csparks	pbs	Yes	never		
djaeger	pbs	No	never		
febner	pbs	Yes	never		
fgruenbichler	pbs	Yes	never		
jbrown	pbs	Yes	never	John Smith	
krichards	pbs	Yes	never		
krodriguez	pbs	Yes	never		
malmat	pbs	Yes	never		
mlimbeck	pbs	Yes	never		
obektas	pbs	Yes	never		
rbarrett	pbs	Yes	never		
rdixon	pbs	Yes	never		
rmartin	pbs	Yes	never		
root	pam	Yes	never	Superuser	
srutz	pbs	Yes	never		
tdavis	pbs	Yes	never		
tlamprecht	pbs	Yes	never		
tmarr	pbs	No	never		
wbumiller	pbs	Yes	never		
wlink	pbs	No	never		

Proxmox Backup Serverはいくつかの認証レルムをサポートしており、新規ユーザを追加する際にレルムを選択する必要があります。選択可能なレルムは以下のとおりです：

pam

Linux PAM標準認証。Linuxシステムユーザとして認証する場合に使用します。ユーザはホストシステム上に存在する必要があります。

pbs

Proxmox Backup Server レルム。このタイプはハッシュ化されたパスワードを/etc/proxmox-backup/shadow.jsonに保存します。

openid

OpenID Connect サーバー。ユーザは外部のOpenID Connectサーバに対して認証を行うことができます。

ldap

ad LDAP サーバー。外部のLDAPサーバーに対して認証を行うことができます。

Active Directoryサーバ。外部のActive Directoryサーバーに対して認証を行うことができます。

インストール後は、Unixのスーパーユーザに相当するroot@pamという単一のユーザが存在します。ユーザ設定情報は /etc/proxmox-backup/user.cfg ファイルに保存されます。proxmox-backup-managerコマンドラインツールを使ってユーザの一覧表示や操作ができます：

The screenshot shows two parts: a terminal window and a graphical user interface (GUI) window.

In the terminal window, the command # proxmox-backup-manager user list is run, showing the following table:

ユーザーID	enable	expire	firstname	lastname	email	コメント
root@pam	1					スーパーユーザ

In the GUI window, the "Add: User" dialog is open. It contains fields for User name (j.smith), First Name (John), Last Name (Smith), E-Mail (j.smith@example.com), Password, Confirm password, Expire (never), Enabled (checked), and Comment. There are "Help" and "Add" buttons at the bottom.

スーパーユーザはすべての管理者権限を持ちますので、他のユーザはより低い権限で追加することをお勧めします。新しいユーザを追加するには、ユーザ作成サブコマンドを使うか、ウェブインターフェースから設定→アクセス制御のユーザ管理タブで行います。createサブコマンドでは、--emailや--passwordのような多くのオプションを指定できます。ユーザプロパティは後でuser updateサブコマンドを使って更新・変更できます（GUIで編集）：

```
# proxmox-backup-manager user create john@pbs --email john@example.com
# proxmox-backup-manager user update john@pbs --firstname John --lastname Smith
# proxmox-backup-manager user update john@pbs --comment "ユーザの例です。"
```

結果のユーザーリストは以下のようになります：

The screenshot shows the terminal window again with the command # proxmox-backup-manager ユーザーリスト run, displaying the following table:

ユーザーID	enable	expire	firstname	lastname	email	コメント
john@pbs 1			John	スミス	john@example.com	ユーザーの例。
root@pam 1						Superuser

新規に作成されたユーザには何の権限もありません。アクセス権限の設定方法については、[アクセス制御のセクション](#)をお読みください。

ユーザー アカウントを無効にするには、--enable を 0 に設定します：

```
# proxmox-backup-manager user update john@pbs --enable 0
```

もししくは

```
# proxmox-backup-manager user remove john@pbs
```

6.2 API トークン

User	Token name	Enabled	Expire	Comment
john@pbs	client1	Yes	never	
john@pbs	johnStore1	Yes	never	

認証されたユーザはAPIトークンを生成することができ、ユーザ名とパスワードを直接入力する代わりにAPIトークンを使って様々なクライアントを設定することができます。

APIトークンには2つの目的があります：

1. クライアントが危険にさらされた場合の簡単な失効
2. 各クライアント/トークンの権限をユーザー権限内に制限

APIトークンは、ユーザー名、レルム、トークン名(user@realm!tokenname)からなる識別子と、シークレット値の2つの部分から構成されます。これらはそれぞれ、ユーザID (user@realm) とユーザパスワードの代わりにクライアントに提供される必要があります。

Token Secret

Token ID:	john@pbs!client1
Secret:	58a77e1c-77ea-4e7d-bf2c-e265b43d93c0

Please record the API token secret - it will only be displayed now

Copy Secret Value

APIトークンは、メソッドPBSAPITokenでAuthorization HTTPヘッダをTOKENID:TOKENSECRETの値に設定することで、クライアントからサーバに渡されます。

トークンはGUIまたはproxmox-backup-managerを使って生成できます：

```
# proxmox-backup-manager user generate-token john@pbs client1 結果:{  
  "tokenId": "john@pbs!client1",  
  "值": "d63e505a-e3ec-449a-9bc7-1da610d4ccde"  
}
```

1 注意

表示されたシークレット値は、APIトークン生成後に再度表示することができないため、保存しておく必要があります。

`user list-tokens` サブコマンドを使うと、トークンとそのメタデータを表示できます：

```
# proxmox-backup-manager user list-tokens john@pbs
+-----+-----+-----+-----+
| tokenid | enable | expire | comment |
+-----+-----+-----+-----+
| john@pbs!client1 | 1 | | |
```

同様に、`user delete-token` サブコマンドを使ってトークンを再度削除することができます。

新しく生成されたAPIトークンには権限がありません。アクセス許可の設定方法については、次のセクションをお読みください。

6.3 アクセス制御

デフォルトでは、新規ユーザーとAPIトークンにはパーミッションがありません。代わりに、許可するものと許可しないものを指定する必要があります。

Proxmox Backup Serverは、ロールおよびパスベースの権限管理システムを使用します。パーミッションテーブルのエントリは、「オブジェクト」または「パス」にアクセスするときに、ユーザ、グループ、またはトークンが特定の役割を担うことを許可します。つまり、アクセスルールは「(パス、ユーザ、ロール)」、「(パス、グループ、ロール)」、「(パス、トークン、ロール)」のトリプルとして表現され、ロールは許可されたアクションのセットを含み、パスはこれらのアクションのターゲットを表します。

6.3.1 特権

特権はアクセスロールの構成要素です。これらは内部的に、APIにおける実際の権限チェックを実施するために使用されます。

現在、以下の権限をサポートしています：

Sys.Audit

Sys.Auditにより、ユーザはシステムとその状態を知ることができます。

Sys.Modify

Sys.Modifyにより、ユーザはシステムレベルの構成を変更し、更新を適用できます。

Sys.PowerManagement

Sys.Modifyにより、ユーザはシステムの電源を切断し、再起動することができます。

データストア監査

Datastore.Audit を使用すると、構成エントリの読み取りや内容の一覧表示など、データストアに関する情報を取得できます。

データストア割り当て

Datastore.Allocate を使用すると、ユーザーはデータストアを作成または削除できます。

データストア.Modify

Datastore.Modify を使用すると、ユーザはデータストアとその内容を変更したり、データストア内のネームスペースを作成または削除したりできます。

データストア.Read

Datastore.Read を使用すると、バックアップグループの所有者に関係なく、任意のバックアップコンテンツを読

み取ることができます。

Datastore.Verify

データストア内のバックアップ・スナップショットの検証を許可します。

データストア・バックアップ

`Datastore.Backup` ユーザが新しいバックアップ スナップショットを作成でき、`Datastore.Read` および `Datastore.Verify` の権限も提供しますが、バックアップグループの所有者がそのユーザまたはそのトークンの 1 つである場合に限ります。

Datastore.Prune

`Datastore.Prune` を使用すると、スナップショットを削除できますが、バックアップの所有権が必要です。

権限.Modify

`Permissions.Modify`を使用すると、ユーザはACLを変更できます。



注

ユーザは、ユーザ権限によって制限されるため、常に自分のAPIトークンの権限を設定できます。

リモート監査

`Remote.Audit`を使用すると、ユーザはリモートと同期の設定エントリを読むことができます。

リモート変更

`Remote.Modify`を使用すると、ユーザはリモート設定を変更できます。

リモート読み込み

`Remote.Read`を使用すると、設定済みのリモートからデータを読み取ることができます。

シスコンソール

`Sys.Console`を使用すると、ユーザーはシステムのコンソールにアクセスできます。`root@pam`以外では、有効なシステム・ログインが必要です。

テープ監査

`Tape.Audit`を使用すると、ユーザーはテープドライブ、チェンジャー、バックアップの構成とステータスを読み取ることができます。

テープ修正

`Tape.Modify`により、ユーザーはテープドライブ、チェンジャー、バックアップの構成を変更できます。

テープ書き込み

`Tape.Write`により、ユーザーはテープ・メディアに書き込むことができます。

テープ・リード

`Tape.Read`を使用すると、ユーザーはテープ・メディアからテープ・バックアップ構成と内容を読み取ることができます。

レルム・アロケート

`Realm.Allocate`を使用すると、ユーザーの認証レルムを表示、作成、変更、および削除できます。

6.3.2 アクセス・ロール

アクセス・ロールは、1つまたは複数の権限を組み合わせて、オブジェクト・パス上のユーザまたはAPIトークンに割り当てる能够性を示すものです。

現在のところ、組み込みのロールしかありません。つまり、独自のロールを作成することはできません。以下の

ロールがあります：

NoAccess

アクセスを無効にします。

管理者

割り当てられたオブジェクトパス上で、何でもできます。

監査

データストア管理

既存のデータストアに対して何でもできます。

データストア監査

データストアのメトリック、設定、およびリストの内容を表示できます。ただし、実際のデータを読むことはできません。

データストアリーダー

データストアまたはネームスペースのコンテンツを検査し、リストアを実行できます。

データストア・バックアップ

所有するバックアップのバックアップとリストアが可能。

データストアPowerUser

所有するバックアップのバックアップ、リストア、およびブルーニングを実行できます。

リモート管理

リモート管理

リモート監査

リモートの設定を表示できます。

リモートシンク

リモートからのデータ読み取りが許可されています。

テープ管理

テープ・バックアップに関するあらゆる操作が可能。

テープ監査

テープ関連のメトリクス、設定、ステータスを表示できます。

TapeOperator

テープ・バックアップとリストアを実行できますが、設定を変更することはできません。

TapeReader

テープ構成とメディア・コンテンツを読み取り、検査できます。

6.3.3 オブジェクトとパス

アクセス許可は、データストア、ネームスペース、一部のシステム・リソースなどのオブジェクトに割り当てられます。

これらのオブジェクトにアクセスするために、ファイルシステムのようなパスを使用します。これらのパスは自然なツリーを形成し、上位レベル（より短いパス）のパーミッションは、オプションでこの階層内に伝搬させることができます。

パスはテンプレート化することができます。これはすなわち、設定エントリの実際のIDを参照できることを意味します。API呼び出しがテンプレート化されたパスに対する権限を要求するとき、そのパスはAPI呼び出しのパラメータへの参照を含むことができます。これらの参照は中かっこで指定します。

いくつかの例を示します：

/データストア	Proxmox Backupサーバ上のすべてのデータストアにアクセスします。
/datastore/{store}	Proxmox Backupサーバ上の特定のデータストアにアクセスします。
/datastore/{store}/{ns}	特定のデータストア上の特定のネームスペースへのアクセス
/リモート	すべてのリモートエントリーにアクセス
/システム/ネットワーク	ホストネットワークの設定へのアクセス
/テープ	テープ・デバイス、プール、ジョブへのアクセス
/アクセス/ユーザー	ユーザー管理
/access/openid/{id}	特定のOpenID Connectレルムへの管理アクセス

継承

前述したように、オブジェクトパスはファイルシステムのようなツリーを形成し、パーミッションはデフォルトで設定されている propagate フラグによって、ツリーの下のオブジェクトに継承されます。以下の継承ルールを使用します：

- API トークンのパーミッションは、常にユーザのものに制限されます。
- より深い、より具体的なレベルのパーミッションは、上位レベルから継承されたものに置き換わります。

6.3.4 設定と管理

The screenshot shows a modal dialog titled "Add: User Permission". It contains four input fields: "Path" set to "/datastore/store1", "User" set to "j.smith@pbs", "Role" set to "DatastoreAdmin", and a "Propagate" checkbox which is checked. At the bottom left is a "Help" button, and at the bottom right is a blue "Add" button.

アクセス許可情報は /etc/proxmox-backup/acl.cfg に保存されます。このファイルには 5 つのフィールドがあり、コロン (:) で区切られています。典型的なエントリは次のような形式です：

```
acl:1:/datastore:john@pbs:DatastoreBackup
```

各フィールドのデータは以下のとおりです：

1. ACL 識別子
2. 伝搬が有効か無効かを表す 1 または 0。
3. 権限が設定されているオブジェクト。これは、特定のオブジェクト（単一のデータストア、リモートなど）またはトップ・レベルのオブジェクトで、プロパゲーションを有効にすると、そのオブジェクトのすべての子も表します。
4. 権限が設定されるユーザ/トークン。
5. 設定されるロール

ウェブ・インターフェイスの Configuration -> Access Control -> Permissions でパーミッションを管理できます。同様に、acl サブコマンドを使用して、コマンドラインからユーザ権限を管理および監視することもできます。例えば、以下のコマンドは /backup/disk1/store1 にあるデータストア store1 の **Data-storeAdmin** としてユーザ john@pbs を追加します：

```
# proxmox-backup-manager acl update /datastore/store1 DatastoreAdmin --auth-id john@pbs
```

各ユーザー/トークンの ACL は以下のコマンドで一覧できます：

# proxmox-backup-manager acl list			
ugid	パス	propagate	roleid
john@pbs	/datastore/store1		
Datastore Admin			

単一のユーザ/トークンに、異なるデータストアに対して複数の権限セットを割り当てることができます。

注

ここでは命名規則が重要です。ホスト上のデータストアでは、次の規約を使用する必要があります。
/ を使用する必要があります。たとえば、/mnt/backup/disk4/store2 にマウントされたデータストアにパーミッションを設定するには、パスに /datastore/store2 を使用します。リモートストアの場合は、/remote/{remote}/{ストア名} という規則を使用します。

ここで{remote}はリモートの名前（以下の「リモート」を参照）、{storename}はリモート上のデータストアの名前です。

6.3.5 APIトークンのパーミッション

APIトークンのパーミッションは、対応するユーザーのものとは別に、そのIDを含むACLに基づいて計算されます。その結果、与えられたパス上のパーミッション・セットは、対応するユーザーのそれと交差します。

実際には、これは次のことを意味します：

1. APIトークンは独自のACLエントリーを必要とします。
2. APIトークンは対応するユーザー以上のこととはできません。

6.3.6 有効なパーミッション

ユーザーあるいはAPIトークンの実効パーミッション・セットを計算して表示するには

`proxmox-backup-manager`のユーザパーミッションコマンドです：

```
# proxmox-backup-manager user permissions john@pbs --path /datastore/store1 (*) の付いた特権には伝播フラグが設定されています。
パス /datastore/store1
- データストア.監査 (*)
- データストア.バックアップ (*)
- データストア.変更 (*)
- データストア.ブルーン (*)
- データストア.読み込み (*)
- データストア.検証 (*)

# proxmox-backup-manager acl update /datastore/store1 DatastoreBackup --auth-id 'john@pbs'

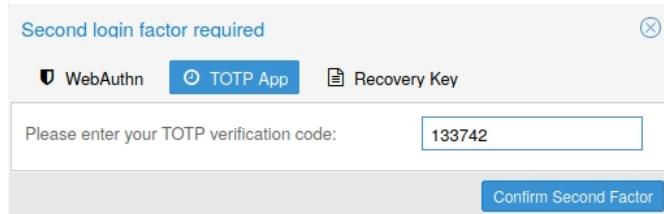
# proxmox-backup-manager user permissions 'john@pbs!client1' --path /datastore/store1 (*) の付いた権限はpropagateフラグが設定されています。
パス /データストア/ストア1
- データストア.バックアップ (*)
```

6.4 二要素認証

6.4.1 はじめに

単純認証では、例えばProxmox Backup Serverの特定のインスタンスで`root@pam`としてログインできるように、パスワード（單一要素）のみでIDの主張（認証）に成功する必要があります。この場合、パスワードが漏れたり盗まれたりすると、ログインが許可されていない場合でも、誰でもそのパスワードを使ってログインすることができます。

二要素認証(TFA)では、ユーザは自分の認証性を確認するために追加の要素を要求されます。この追加要素は、ユーザーだけが知っているもの（パスワード）に頼るのでなく、ユーザーだけが持っているもの、例えばハードウェアの一部（セキュリティキー）やユーザーのスマートフォンに保存された秘密などを要求します。これにより、たとえパスワードを持っていたとしても、物理的なオブジェクト（セカンドファクター）にはアクセスできないため、遠隔地のユーザーがアカウントに不正アクセスすることを防ぎます。



6.4.2 利用可能な第二要素

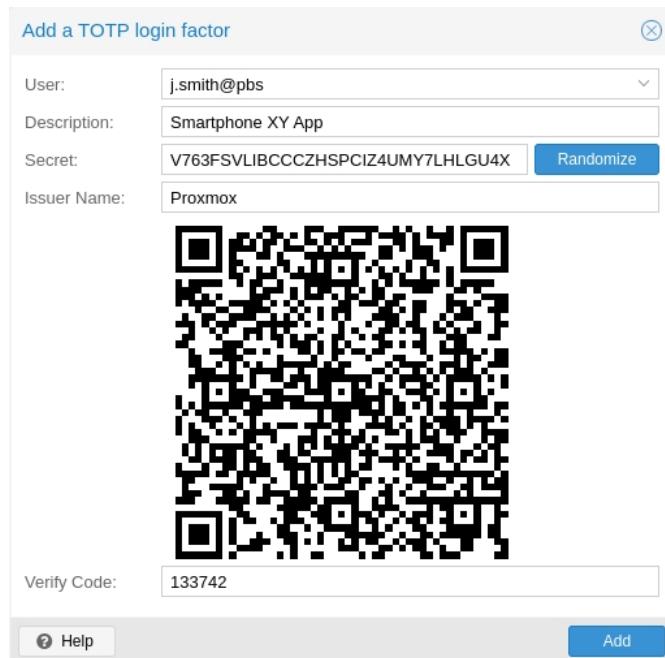
スマートフォンやセキュリティキーを紛失してアカウントから永久にロックされる事態を避けるため、複数の二要素認証を設定することができます。

Proxmox Backup Serverは3つの異なる2要素認証方法をサポートしています：

- TOTP (Time-based One-Time Password)。共有秘密と現在時刻から派生する短いコードで、30秒ごとに変更されます。
- WebAuthn (ウェブ認証)。一般的な認証規格。コンピュータやスマートフォンのハードウェアキーやTPM (Trusted Platform Module) など、さまざまなセキュリティデバイスによって実装されます。
- シングル・ユース・リカバリ・キー (Single use Recovery Keys)。印刷して安全な場所に保管するか、電子保管庫にデジタル保存する必要がある鍵のリスト。各キーは一度しか使用できません。他のすべてのセカンドファクターが紛失または破損した場合でも、ロックアウトされないようにするために最適です。

6.4.3 セットアップ

TOTP



サーバーの設定は不要です。スマートフォンにTOTPアプリ（FreeOTPなど）をインストールし、Proxmox Backup Serverのウェブインターフェースを使用してTOTPファクターを追加するだけです。

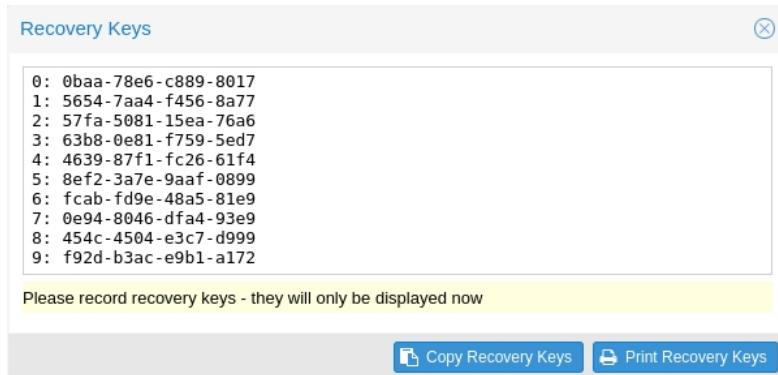
WebAuthn

WebAuthnを動作させるには、2つのものが必要です：

- 信頼できる HTTPS 証明書 (Let's Encrypt を使用するなど)。信頼されていない証明書でもおそらく動作しますが、ブラウザによっては信頼されていないと WebAuthn の操作を警告したり拒否したりすることがあります。
- WebAuthn 設定をセットアップします (Proxmox Backup Server Web インターフェイスの【設定】->【その他】を参照)。これはほとんどのセットアップで自動入力されます。

これらの両方の要件を満たしたら、WebAuthn設定を
アクセス制御パネルの「二要素認証」タブ。

リカバリキー



リカバリキーコードを準備する必要はありません。アクセスコントロールパネルの二要素認証タブでリカバリキーのセットを作成するだけです。

注

一度に使用できるリカバリキーは、ユーザごとに1セットのみです。

6.4.4 TFAと自動アクセス

二要素認証はウェブインターフェースにのみ実装されています。他のすべてのユースケース、特に非インタラクティブなユースケース（例えば、Proxmox Backup ServerをストレージとしてProxmox VEに追加する場合）にはAPI Tokensを使用する必要があります。

6.4.5 二要素認証の制限とロックアウト

二要素認証は、パスワードが何らかの形で漏れたり推測されたりした場合にユーザーを保護するためのものです。しかし、ブルートフォースによって破られる可能性があります。このため、第2要素によるログインに何度も失敗すると、ユーザーはロックアウトされます。

TOTPの場合、8回失敗すると、ユーザーのTOTPファクターは無効になります。回復キーでログインすると解除されます。TOTPが唯一の利用可能な要素であった場合、管理者の介入が必要となり、ユーザに直ちにパスワードの変更を要求することが強く推奨されます。

FIDO2/Webauthnとリカバリキーはブルートフォースアタックの影響を受けにくいため、リミットは高くなります（100回）、それを超えるとすべてのセカンドファクターが1時間ブロックされます。

管理者はウェブUIのユーザーリストビュー、またはコマンドラインを使っていつでもユーザーの二要素認証を解除できます：

```
proxmox-backup-manager user tfa unlock joe@pbs
```

6.5 認証レルム

6.5.1 Linux PAM

Linux PAM は、システム全体のユーザー認証のためのフレームワークです。これらのユーザは、adduserなどのコマンドを使用してホストシステム上に作成されます。

PAM ユーザーがホストシステムに存在する場合、対応するエントリを Proxmox Backup Server に追加して、これらのユ

ユーザーがシステムのユーザー名とパスワードでログインできるようにすることができます。

6.5.2 Proxmox Backup 認証サーバ

これはUnixライクなパスワードストアで、ハッシュ化されたパスワードを/etc/proxmox-backup/shadow.jsonに保存します。パスワードはSHA-256ハッシュアルゴリズムを使ってハッシュ化されます。

これは、ユーザがProxmox Backup Server以外へのアクセスを必要としない小規模（または中規模）のインストールに最も便利な領域です。この場合、ユーザはProxmox Backup Serverによって完全に管理され、GUIを介して自分のパスワードを変更することができます。

6.5.3 LDAP

Proxmox Backup Serverはユーザー認証に外部LDAPサーバーを利用できます。これを実現するには、ldapタイプのレルムを設定する必要があります。

LDAPでは、ユーザーはドメイン(dn)によって一意に識別されます。例えば、以下のLDIFデータセットでは、ユーザーuser1は一意なドメインuid=user1,ou=People,dc=ldap-test,dc=comを持っています：

```
#ldap-test.comのPeopleのuser1
dn: uid=user1,ou=People,dc=ldap-test,dc=com objectClass: top
objectClass: person
objectClass: organizationalPerson objectClass:
inetOrgPerson
```

```
uid: user1
cn : テストユーザー1
sn: テスター
の説明を参照してください：これは最初のテストユーザーです。
```

同様に、Proxmox Backup Serverはユーザーを一意に識別するためにユーザー識別子(userid)を使用します。したがって、Proxmox Backup ServerのuseridとLDAP dnの間のマッピングを確立する必要があります。このマッピングはuser-attr設定パラメータによって確立されます。

有効なProxmox Backup Serverユーザ識別子を含むLDAP属性名。

上記の例では、user-attrをuidに設定すると、ユーザーuser1@<realm-name>がLDAPエンティティuid=user1,ou=People,dc=ldap-test,dc=comにマッピングされます。ユーザーのログイン時に、Proxmox Backup Serverは設定されたベースドメイン(base-dn)の下でサブツリー検索を実行し、ユーザーのdnを照会します。dnがわかる

LDAPバインド操作が実行され、LDAPサーバーに対してユーザーを認証します。

すべてのLDAPサーバーが匿名検索操作を許可しているわけではないため、バインドドメイン(bind-dn)とバインドパスワード(password)を設定することができます。設定すると、Proxmox Backup Serverは検索操作を実行する前に、これらの認証情報を使用してLDAPサーバーにバインドします。

すべての設定パラメータの完全なリストは[domains.cfg](#)にあります。

注意

特定のユーザがLDAPサーバーを使用して認証できるようにするには、そのユーザをProxmox Backup Serverでそのレルムのユーザとして追加する必要があります。これは同期によって自動的に実行されます。

6.5.4 アクティブディレクトリ

Proxmox Backup Serverはユーザ認証に外部のMicrosoft Active Directoryサーバーを利用することもできます。これを実現するには、広告タイプのレルムを設定する必要があります。

Active Directoryレルムでは、認証ドメイン名とサーバーアドレスを指定する必要があります。[LDAP](#)のほとんどのオプションがActive Directoryにも適用されます。

クレデンシャルbind-dnとpassword。これは通常、Microsoft Active Directoryではデフォルトで必要です。bind-dnはAD固有のuser@company.net構文か、あるいは一般的なLDAP-DN構文で指定します。

一般的なLDAP-DN構文で指定できます。

認証ドメイン名を指定しなければならぬのは、匿名バインドを要求する場合だけです。bind 認証情報を指定した場合は、ドメイン名は Active Directory サーバが報告する bind ユーザのベースドメインから自動的に推測されます。

すべての設定パラメータの一覧は [domains.cfg](#) にあります。

 注意

特定のユーザーにActive Directoryサーバを使用した認証を許可するには、Proxmox Backup Serverでそのレルムのユーザーとして追加する必要があります。これは同期によって自動的に実行されます。

 注意

現在、大文字と小文字を区別しないユーザ名はサポートされていません。

LDAP/AD領域でのユーザー同期

Proxmox Backup Server にユーザーを手動で追加する代わりに、LDAP および AD ベースのレルムでユーザーを自動的に同期することができます。同期オプションは、GUIのLDAP レルム設定ダイアログウィンドウで設定できます。

`proxmox-backup-manager ad` コマンドで設定できます。

ユーザーの同期は GUI の **Configuration > Access Control > Realms** でレルムを選択して *Sync* ボタンを押すことで開始できます。同期ダイアログでは、レルム設定で設定されたデフォルトオプションのいくつかを上書きすることができます。あるいは、ユーザーの同期を `proxmox-backup-manager ldap sync` と `proxmox-backup-manager ad sync` コマンドを使うこともできます。

バックアップクライアントの使用法

Proxmox Backup Serverのコマンドラインクライアントは`proxmox-backup-client`と呼ばれます。

7.1 バックアップリポジトリの場所

クライアントは、バックアップ・サーバ上のデータストア・リポジトリを指定するのに以下の形式を使用します（ユーザー名は`user@realm`の形式で指定します）：

[ユーザー名@]サーバ[:ポート]:]データストア

`username`のデフォルト値は`root@pam`です。サーバを指定しない場合、デフォルトはローカル・ホスト (`localhost`) です。

バックアップ・サーバがデフォルト以外のポートでしか到達できない場合は、ポートを指定できます（NATやポート転送の設定など）。

サーバーがIPv6アドレスを使用している場合は、角括弧で囲んで記述する必要があることに注意してください（例：`[fe80::01]`、`[fe80::01]`）。

リポジトリは--repositoryコマンドラインオプションか、環境変数

`PBS_REPOSITORY` 環境変数を設定します。

Web インターフェースでは、データストアのサマリーで、接続情報の表示ボタンを使ってコピー可能なリポジトリテキストを提供します。

以下は、有効なリポジトリとそれに対応する実際の値の例です：

例	ユーザー	ホスト:ポート	データストア
データストア	<code>root@pam</code>	ローカルホスト:8007	<code>mydatas-store</code>
<code>myhostname:mydatastore</code>	<code>root@pam</code>	マイホスト名:8007	マイデータを
<code>user@pbs@myhostname:mydatastore</code>	ユーザー	ホスト名:8007	<code>mydatas-</code> を削除します。
<code>ユーザー@pbs!token@host:store</code>	<code>user@pbs !</code> トークン	ホスト:8007	ストア
<code>192.168.55.55:1234:mydatastore</code>	<code>root@pam</code>	<code>192.168.55.55:1234</code>	<code>mydatas-</code> ストア
<code>[ff80::51]:mydatastore</code>	<code>root@pam</code>	<code>[ff80::51]:8007</code>	<code>mydatas-</code> を削除します。
<code>[ff80::51]:1234:mydatastore</code>	<code>root@pam</code>	<code>[ff80::51]:1234</code>	<code>mydatas-</code> を削除します。

7.2 静的にリンクされたバックアップ・クライアント

Proxmox Backupクライアントの静的にリンクされたバージョンは、通常のクライアントが使用できないLinuxベースのシステムで使用できます。静的にリンクされたクライアントは完全な代替ではないため、可能な限り通常のクライアントを使用することをお勧めします。例えば、名前解決はlibclによって提供されるメカニズムではなく、純粋にRustプログラミング言語で書かれたリゾルバを使用します。そのため、Name Service Switchが提供する機能やモジュールは使用できません。

静的にリンクされたクライアントは、`pbs-client` リポジトリ経由で入手できます。
セクションで説明します。

7.3 環境変数

`PBS_REPOSITORY`

デフォルトのバックアップリポジトリ。

`PBS_PASSWORD`

設定すると、この値がバックアップサーバーのパスワードとして使用されます。APIトークンのシークレットに設定することもできます。

`pbs_password_fd, pbs_password_file, pbs_password_cmd`

`PBS_PASSWORD`と似ていますが、オープンファイルディスクリプター、ファイル名、または標準出力からデータを読み取ります。
それぞれコマンドの上記の順序から最初に定義された環境変数が優先されます。
が優先されます。

`pbs_encryption_password`

設定されると、この値は秘密の暗号化キーにアクセスするために使われます（パスワードで保護されている場合）。

`pbs_encryption_password_fd, pbs_encryption_password_file, pbs_encryption_password_cmd`

`PBS_ENCRYPTION_PASSWORD`と似ていますが、オープンファイルディスクリプター、ファイル名、またはコマンドの標準出力からデータを読み込みます。
をそれぞれコマンドの標準出力から読み取ります。上記の順序から最初に定義された環境変数が優先されます。
の順に優先されます。

`pbs_fingerprint`

設定すると、この値がサーバ証明書の検証に使用されます（システムCA証明書が証明書を検証できない場合にのみ使用されます）。

`ALL_PROXY`

設定すると、クライアントはバックアップ・サーバへのすべての接続に指定のHTTPプロキシを使用します。現在、HTTP プロキシのみがサポートされています。有効なプロキシ構成は以下の形式です：
`[http://][user:password@]<host>[:port]`。特に指定がない場合、デフォルトのポートは1080です。

注意

ホストのシールドには、HTTP プロキシを使用するのではなく、wireguardなどのトンネルを使用することをお勧めします。

注意

パスワードは有効なUTF-8でなければならず、改行を含めることはできません。便宜上、Proxmox Backup Serverは最初の行のみをパスワードとして使用するので、最初の改行の後に任意のコメントを追加できます。

7.4 システムとサービスの資格情報

上記の環境変数のいくつかは、システムやサービスの認証情報を使って設定することができます。

環境変数	クレデンシャル名
PBS_REPOSITORY	proxmox-backup-client.リポジトリ
PBS_PASSWORD	proxmox-backup-client.password。
PBS_ENCRYPTION_PASSWORD	プロックスモックスバックアップクライアント.暗号化パスワード
pbs_fingerprint	proxmox-backup-client.fingerprint

例えば、リポジトリのパスワードは以下のように暗号化されたファイルに保存することができます：

```
# systemctl-ask-password -n|systemd-creds encrypt --name= proxmox-backup-client.password - my-
'>api-token.cred
```

このクレデンシャルは、以下のようにユニットファイル内や一時的なスコープユニット内で再利用することができます：

```
# systemd-run --pipe --wait \ --property LoadCredentialEncrypted .
--property=LoadCredentialEncrypted= proxmox-backup-client.password:/full/path/to/my-api'認証情報を再利用します。
'>token.cred
--property=SetCredential= proxmox-backup-client.repository:'my_default_repository' \
proxmox-backup-client ...
```

さらに、システムクレデンシャル(例えばハイパーバイザーから SMBIOS タイプ 11 経由で仮想マシンに渡される)は、マニュアルページ

`systemd.exec(5)` で説明されています。

7.5 出力フォーマット

出力を生成するほとんどのコマンドは`--output-format`パラメータをサポートしています。これは以下の値を受け付けます：

テキスト

テキスト・フォーマット(デフォルト)。構造化されたデータは表としてレンダリングされます。

json

JSON(1行)。

json-pretty

JSON(複数行、きれいにフォーマット)。

また、以下の環境変数で出力の挙動を変更できます：

`proxmox_output_format`
デフォルトの出力フォーマットを定義します。

`proxmox_output_no_border`
設定されている場合(任意の値)、テーブルのボーダーをレンダリングしません。

`proxmox_output_no_header`
(任意の値に)設定すると、テーブルのヘッダをレンダリングしません。

注意

テキスト形式は人間が読めるように設計されており、自動化ツールによる解析は想定していません。出力を処理する必要がある場合は、`json`形式を使用してください。

7.6 バックアップの作成

このセクションでは、マシン内からバックアップを作成する方法を説明します。これは、物理ホスト、仮想マシン、またはコンテナである可能性があります。このようなバックアップには、ファイルアーカイブやイメージアーカイブが含まれることがあります。この場合、特に制限はありません。



注

Proxmox VE上の仮想マシンまたはコンテナをバックアップする場合は、[Proxmox VE Integration](#)を参照してください。

以下の例では、バックアップサーバーをセットアップし、認証情報を使用し、リポジトリ名を知っている必要があります。以下の例では `backup-server:store1` を使用しています。

```
# proxmox-backup-client backup root.pxar:/ --repository backup-server:store1 Starting backup:
host/elsa/2019-12-03T09:35:01Z
クライアント名:elsa
スキップマウントポイント: "/boot/efi" キップ
マウントポイント: "/dev"
スキップマウントポイント: "/run" キップ
マウントポイント: "/sys"
87秒で12129チャングをアップロード (564 MB/s)。終了時刻: 2019-12-03T10:36:29+01:00
```

パスワードの入力を求めるプロンプトが表示され、/ディレクトリ内のすべてのファイルを含む`root.pxar`という名前のファイルアーカイブがアップロードされます。



`proxmox-backup-client`は自動的にマウントポイントを含まないことに注意してください。その代わりに、それぞれのマウントポイントに対して短いスキップメッセージが表示されます。このアイデアはマウントされたディスクごとに個別のファイルアーカイブを作成することです。を使用して明示的に含めることができます。

`--include-dev`オプション（例：`--include-dev /boot/efi`）。このオプションは複数回使用できます。

`repository`オプションは非常に長くなり、すべてのコマンドで使用されます。環境変数 `PBS_REPOSITORY` を設定することで、この値を入力する手間を省くことができます。複数のセッションでこの値を設定し続けたい場合は、代わりに `.bashrc` ファイルに追加してください。

```
# export PBS_REPOSITORY = backup-server:store1
```

これで、`--repository`オプションを指定しなくても、すべてのコマンドを実行できるようになります。

1つのバックアップに複数のアーカイブを含めることができます。例えば、`/mnt/disk1`と`/mnt/disk2`にマウントされた2つのディスクをバックアップしたい場合：

```
# proxmox-backup-client backup disk1.pxar:/mnt/disk1 disk2.pxar:/mnt/disk2
```

これで両方のディスクのバックアップが作成されます。

バックアップターゲットに名前空間を使用したい場合は、`--ns`パラメータを追加します：

```
# proxmox-backup-client backup disk1.pxar:/mnt/disk1 disk2.pxar:/mnt/disk2 --ns a/b/c
```

`backup` コマンドは、サーバー上のアーカイブ名、アーカイブのタイプ、クライアントでのアーカイブソースを含むバックアップ指定のリストを受け取ります。書式は

<アーカイブ名>.<タイプ>:<ソースパス>

アーカイブ名には、英数字、ハイフン、アンダースコアのみを含める必要があります。一般的なタイプは、ファイルアーカイブの場合は`.pxar`、ブロックデバイスイメージの場合は`.img`です。ブロックデバイスのバックアップを作成するには、以下のコマンドを実行してください：

```
# proxmox-backup-client backup mydata.img:/dev/mylvm/mydata
```

7.6.1 バックアップからのファイル/ディレクトリの除外

バックアップアーカイブから特定のファイルやディレクトリを除外したい場合があります。Proxmox Backup クライアントにファイルとディレクトリを無視するタイミングと方法を指示するには、ファイルシステム階層に `.pxareexclude` という名前のテキストファイルを配置します。バックアップクライアントがこのようなファイルをディレクトリでこのようなファイルが見つかると、各行をバックアップから除外するファイルとディレクトリのグローブ一致パターンとして解釈します。

として解釈します。

ファイルには、1行に1つのグローブ・パターンを含める必要があります。空行と # (コメントを示す) で始まる行は無視されます。行頭の ! は、グローブ・マッチ・パターンを除外から明示的包含に反転させます。

を付けると、グローブ・マッチ・パターンが除外から明示的な包含に反転します。これにより、一部のファイル/サブディレクトリを除き、ディレクトリ内のすべてのエントリを除外することができます。で終わる行は、ディレクトリにのみマッチします。`.pxareexclude` ファイルを含むディレクトリは、指定されたパターンのルートとみなされます。これはこのディレクトリとそのサブディレクトリにあるファイルをマッチさせることができます。

1 注意

先頭の / がないパターンはサブディレクトリでもマッチしますが、 / があるパターンはカレントディレクトリでのみマッチします。

があるパターンはカレントディレクトリにのみマッチします。

\ は、特殊なグローブ文字をエスケープするために使用されます。は任意の1文字にマッチします。* は、空文字列を含む任意の文字にマッチします。** は、カレント・ディレクトリとサブディレクトリにマッチします。例えば、パターン `*/*.tmp` を使用すると、ディレクトリとそのサブディレクトリ内の `.tmp` で終わるすべてのファイルを除外します。 [...] は、大括弧の中で指定された文字の中から1文字にマッチします。 [...] は補完を行い、括弧内に含まれない任意の1文字にマッチします。で区切って範囲を指定することもできます。例えば、`[a-z]` はアルファベットの小文字に、`[0-9]` は数字1文字にマッチします。

つまり、後のエントリが前のエントリを上書きします。これは、ディレクトリツリーの奥深くで遭遇したマッチパターンにも当てはまり、前の除外を上書きすることができます。

1 注意

除外されたディレクトリはバックアップクライアントによって読み取られません。したがって、除外されたサブディレクトリの `.pxareexclude` ファイルは何の効果もありません。`.pxareexclude` ファイルは通常のファイルとして扱われ、バックアップアーカイブに含まれます。

例えば、次のようなディレクトリ構造を考えてみましょう：

```
# ls -aR folder
folder:
pxareexclude
folder/subfolder0:
... file0 file1      サブフォルダ0 サブフォルダ1
folder/subfolder1:
... file2 file3 .pxareexclude
... file0 file1      file2 file3
```

異なる `.pxareexclude` ファイルには、以下の内容が含まれます：

```
# cat folder/.pxareexclude
/サブフォルダ-0/ファイル1
/サブフォルダ-1
!/subfolder1/file2
```

```
# cat folder/subfolder0/.pxareexclude file3
```

これにより、サブフォルダ0のfile1とfile3、およびfile2を除くサブフォルダ1のすべてが除外されます。

このバックアップをリストアすると

```
# ls -aR restored
restored/:
..pxareexclude subfolder0 subfolder1 restored/subfolder0
.:
... file0 file2 .pxareexclude
restored/subfolder1:
... file2
```

同じ構文を `--exclude` パラメータを使用して cli で直接使用することもできます。例えば

```
# proxmox-backup-client backup archive-name.pxar ./linux --exclude /usr
```

このように複数のパスを除外することができます：

```
# proxmox-backup-client backup archive-name.pxar ./linux --exclude= /usr --exclude= /rust=
```

7.6.2 変更検出モード

Proxmox バックアップクライアントのデフォルトの動作は、すべてのデータを読み込んで pxar アーカイブにエンコードするためです。エンコードされたストリームは可変サイズのチャンクに分割されます。各チャンクについてダイジェストが計算され、そのチャンクをアップロードする必要があるか、それともアップロードせずにインデックス化できるかを決定するために使用されます。バックアップされたファイルがほとんど変更されていない場合、該当するチャンクを再読み込みし、結局アップロードする必要がないことを検出するのは時間がかかり、望ましくありません。

バックアップクライアントの変更検出モードをデフォルトからメタデータベースの検出に切り替えることで、上記のような制限を減らし、ファイルの再読み込みを避けるようにクライアントに指示することができます。

バックアップスナップショットは、可能な限りメタデータを変更せずに保存されます。このモードを使用する場合、通常の pxar アーカイブの代わりに、バックアップスナップショットは 2 つの別々のファイルに保存されます：アーカイブのメタデータを含む `mpxar` と、ファイルの内容を連結した `ppxar` です。この分割により

効率的なメタデータ検索が可能になります。バックアップアーカイブを作成する際、現在のファイルのメタデータは以前の `mpxar` アーカイブで検索されたものと比較されます。操作の詳細については、[テクニカルドキュメント](#)を参照してください。

`change-detection-mode` を `data` に設定すると、メタデータモードを使用したときと同じ分割アーカイブを作成することができます。このモードの詳細については、[技術文書](#)を参照してください。

モード	説明
レガシー (現在のデフォルト)	すべてのファイルを自己完結型の pxar アーカイブにエンコードします。
データ	すべてのファイルを分割されたデータとメタデータの pxar アーカイブにエンコードします。
メタデータ	変更されたファイルをエンコードし、前回のスナップショットから変更せずに再利用し、分割されたアーカイブを作成します。

以下はメタデータモードでのクライアント呼び出しの例です：

```
# proxmox-backup-client backup archive-name.pxar ./linux --change-detection-mode= metadata
```

7.7 暗号化

Proxmox Backup は GCM モードの AES-256 によるクライアントサイド暗号化をサポートしています。これを設定するには、まず暗号化キーを作成する必要があります：

```
# proxmox-backup-client key create my-backup.key 暗号化キーパスワード:
```

鍵はデフォルトでパスワードで保護されています。この特別な保護が不要な場合は、パスワードなしで作成することができます：

```
# proxmox-backup-client key create /path/to/my-backup.key --kdf none
```

このキーを作成した後、--keyfileを渡すことで暗号化バックアップを作成することができます。パラメータに鍵ファイルのパスを指定します。

```
# proxmox-backup-client backup etc.pxar:/etc --keyfile /path/to/my-backup.key パスワードを入力してください
: *****
(.) E(.) n.ryption Key Password : *****
```

i 注意

バックアップ・キーの名前を指定しない場合、キーはデフォルトの場所に作成されます。
~/proxmox-backup-clientは--keyfileパラメータが指定されていない場合、デフォルトでこの場所を検索します。

環境変数 PBS_PASSWORD と

pbs_encryption_password を設定することでパスワードの入力を避けることができます。

7.7.1 マスターキーによる暗号化キーの保存と復元

また、proxmox-backup-client keyを使ってRSA公開鍵/秘密鍵ペアを作成することもできます。

の暗号化されたバージョンを各バ
ックアップと一緒に保存し、後で
それを復元するために使用できま
す。

マスター・キーを設定するには

1. バックアップ用の暗号化キーを作成します：

```
# proxmox-backup-client key create
 にデフォルトキーを作成します："~/.config/proxmox-backup/encryption-key.json" に作成します。
(.) E(.) n.ryption Key Password : *****
```

作成されたファイルは ~/.config/proxmox-backup/encryption-key.json に保存されます。

2. RSA公開鍵と秘密鍵のペアを作成します：

```
# proxmox-backup-client key create-master-key
(.) M(.) a.ster Key Password : *****
```

カレント ディレクトリに2つのファイルを作成します、 master-public.pem と
master-private.pemです。

3. インポート をインポートします。 新しく 新しく作成した をインポートします。 パブリック 証明書を作成します、 ですから を作成しました。

proxmox-backup-clientはバックアップ時にこの証明書を見つけて使うことができます。

```
# proxmox-backup-client key import-master-pubkey /path/to/master-public.pem パブリックマスターkeyを
"~/.config/proxmox-backup/master-public.pem" にインポートします。
```

4. これらのファイルを配置した状態でバックアップジョブを実行します：

```
# proxmox-backup-client backup etc.pxar:/etc
```

鍵はrsa-encrypted.keyという名前でバックアップに保存されます。

i 注意

暗号化キーがデフォルトパスにある場合、--keyfileパラメータは除外できます。作成時に別のパスを指定した場合は、--keyfileパラメータを渡す必要があります。

5. すべてがうまくいったかテストするには、バックアップから鍵をリストアします：

```
# proxmox-backup-client restore /path/to/backup/ rsa-encrypted.key /path/to/target
```

i 注意

このファイルを取り出すのに暗号化キーは必要ありません。ただし、暗号化解除された場所（`~/.config/proxmox-backup/encryption-key.json`）に鍵が存在する場合、プログラムは暗号化鍵のパスワードの入力を要求します。`encryption-key.json`をこのディレクトリの外をこのディレクトリから移動するだけで、この問題は解決します。

6. 次に、以前に生成したマスターキーでファイルを復号化します：

```
# proxmox-backup-client key import-with-master-key /path/to/target --master-keyfile '/→path/to/master-keyfile /'→path/to/master-private.pem --encrypted-keyfile /path/to/rsa-encrypted.key マスターキーのパスワード：*****
```

新しいパスワード*****
検証パスワード*****

7. ターゲット・ファイルには、暗号化キー情報がプレーン・テキストで含まれるようになります。バックアップからファイルを復号化する際に、`--keyfile`パラメータを指定して結果のjsonファイルを渡すことで、この成功が確認できます。

⚠️ 警告

キーがないと、バックアップしたファイルにアクセスできなくなります。したがって、キーは順序よく、バックアップするコンテンツとは別の場所に保管する必要があります。たとえば、システム全体をバックアップし、そのシステムのキーを使用することができます。何らかの理由でシステムにアクセスできなくなり、復元する必要が生じた場合、暗号化キーは壊れたシステムとともに失われてしまうため、復元は不可能です。

マスターキーは安全な場所に保管することをお勧めしますが、迅速な災害復旧のため、簡単にアクセスできるようにすることをお勧めします。そのため、マスターキーの保管場所としては、すぐに復旧可能なパスワード・マネージャー内が最適です。このバックアップとして、キーをUSBフラッシュドライブに保存し、安全な場所に保管してください。こうすることで、どのシステムからも切り離された状態になりますが、それでも緊急時には簡単に復旧できます。最後に、最悪の場合に備えて、マスターキーの紙媒体も安全な場所に保管しておくことをお勧めします。サブコマンド `paperkey`サブコマンドを使用すると、マスター・キーのQRエンコード・バージョンを作成できます。以下のコマンドは `paperkey`コマンドの出力をテキストファイルに送信し、簡単に印刷できるようにします。

```
proxmox-backup-client key paperkey --output-format text> qrkey.txt
```

7.8 データのリストア

バックアップの定期的な作成はデータ損失を避けるために必要なステップです。しかし、より重要なのは復元です。定期的にリカバリーテストを行い、災害時にデータにアクセスできることを確認するのがよい方法です。

まず、復元したいスナップショットを見つける必要があります。`snapshot list`コマンドはサーバ上の全てのスナップショットのリストを提供します：

```
# proxmox-backup-client snapshot list
```

スナップショット	サイズ	ファイル
host/elsa/2019-12-03T09:30:15Z	51788646825	root.pxar catalog.pcat1 index.json
host/elsa/2019-12-03T09:35:01Z	51790622048	root.pxar catalog.pcat1 index.json
...		

ヒント

リストは、既定ではルート・ネームスペース自体のバックアップ・スナップショットのみを出力します。別のネームスペースからのバックアップを一覧表示するには、`--ns <ns>` オプションを使用します。

カタログを検査して特定のファイルを見つけることができます。

```
(.) # proxmox-backup-client catalog dump host/elsa/2019-12-03T09:35:01Z
d  "./root.pxar.didx/etc/cifs-utils"
l  "./root.pxar.didx/etc/cifs-utils/idmap-plugin"
(.) d(.) "(.) ./root.pxar.didx/etc/console-setup"
```

リストアコマンドはバックアップから単一のアーカイブをリストアします。

```
# proxmox-backup-client restore host/elsa/2019-12-03T09:35:01Z root.pxar /target/path/
```

アーカイブの内容を取得するには、リポジトリのindex.jsonファイルをターゲットパス'-'にリストアします。これにより、内容が標準出力にダンプされます。

```
# proxmox-backup-client restore host/elsa/2019-12-03T09:35:01Z index.json -.
```

7.8.1 インタラクティブリストア

いくつかのファイルを個別にリストアするだけであれば、対話型リカバリシェルを使う方が簡単な場合が多いです。

```
# proxmox-backup-client catalog shell host/elsa/2019-12-03T09:35:01Z root.pxar 対話型シェルの開始
pxar:/> ls
b(.) i(.) n. ブート dev その他 ホーム lib lib32
```

対話型復旧シェルは、カタログに保存されているメタデータを利用して、ファイルアーカイブ内のファイルをすばやくリスト、ナビゲート、検索する最小限のコマンドラインインターフェイスです。ファイルを復元するには、ファイルを個別に選択するか、グローブパターンで一致させます。

ナビゲーションのためにカタログを使用すると、カタログのみをダウンロードし、オプションで復号化する必要があるため、オーバーヘッドが大幅に削減されます。実際のチャンクにアクセスするのは、カタログのメタデータが不十分な場合か、実際のリストアの場合のみです。

一般的な UNIX シェルと同様に、`cd` と `ls` は作業ディレクトリを変更したり、アーカイブ内のディレクトリの内容を一覧表示するために使用されるコマンドです。

アーカイブの内容を素早く検索する機能は、一般的に必要とされる機能です。そこで、カタログが最も役に立ちます。

例えば

```
pxar:/> find etc/**/*txt --select "/etc/X11/rgb.txt"
pxar:/> リスト選択 etc/**/*txt
pxar:/> restore-selected /target/path
```

`list-selected`は、これらのパターンを表示し、`restore-selected`は、最終的にローカルホスト上の`/target/`パスにパターンに一致するアーカイブ内のすべてのファイルをリストアします。これはアーカイブ全体をスキヤンします。

`restore`コマンドを使用すると、バックアップアーカイブ内のすべてのファイルをリストアできます。これは、`-pattern <glob>`オプションと組み合わせると最も便利です。

特定のパターンに一致するすべてのファイルをリストアできるからです。たとえば、`/etc`にある設定ファイルをリストアしたい場合は、次のようにします：

```
(.) pxar:/> restore target/ --pattern etc/**/*conf
```

上記は、`/etc`以下のすべてのディレクトリをスキヤンし、`.conf`で終わるすべてのファイルをリストアします。

7.8.2 FUSEによるアーカイブのマウント

pxarアーカイブの *FUSE* 実装では、ファイルアーカイブを読み込み専用のファイルシステムとしてホスト上のマウントポイントにマウントすることができます。

```
# proxmox-backup-client mount host/backup-client/2020-01-29T11:29:22Z root.pxar /mnt/
→マウントポイント
# ls /mnt/mountpoint
bin      dev home lib32 libx32  lost+found mnt
lib      lib64          proc    run
                  srv
                  sys  usr  boot   etc
                  tmp  var
```

これにより、アーカイブの全コンテンツにシームレスにアクセスできるようになります。



注意

FUSE接続はバックアップサーバのデータストアからチャンクを取得して復号化する必要があるため、マウントされたファイルシステムで実行する操作によっては、ホストに追加のネットワーク負荷とCPU負荷がかかる可能性があります。

ファイルシステムをアンマウントするには、マウントポイントでumountコマンドを使用します：

```
# umount /mnt/mountpoint
```

7.9 ログインとログアウト

クライアントツールでは、バックアップサーバーにアクセスするとすぐにログインパスワードの入力を求められます。サーバは認証情報をチェックし、2時間有効なチケットを返信します。クライアントツールはそのチケットを自動的に保存し、このサーバへのさらなるリクエストに使用します。

loginおよびlogoutコマンドを使用して、手動でこのログイン/ログアウトをトリガすることもできます：

```
# proxmox-backup-client login パスワードを入力してください
```

チケットを削除するには、ログアウトを発行します：

```
# proxmox-backup-client logout
```

7.10 バックアップグループの所有者の変更

デフォルトでは、バックアップ・グループの所有者は、そのバックアップ・グループを最初に作成したユーザ（同期ジョブの場合はroot@pam）です。つまり、ユーザmike@pbsがバックアップを作成した場合、別のユーザjohn@pbsを使用して同じバックアップ・グループのバックアップを作成することはできません。万が一バックアップの所有者を変更したい場合は、データストアのDatastore.Modify権限を持つユーザを使用して、以下のコマンドを実行します。

```
# proxmox-backup-client change-owner vm/103 john@pbs
```

これはWebインターフェイスから行うこともできます。

に移動し、[アクション]列の下にあるユーザアイコンを選択します。よくあるケースとしては、同期ジョブの所有者をroot@pamから変更したり、バックアップ・グループを再利用したりする場合です。

7.11 バックアップの刈り込みと削除

forgetコマンドを使用してバックアップスナップショットを手動で削除することができます：

```
# proxmox-backup-clientスナップショット forget <スナップショット>
```

 注意

このコマンドはバックアップスナップショット内のすべてのアーカイブを削除します。それらはアクセスできなくなり、回復できなくなります。

ルート名前空間に含まれるスナップショットを忘れない場合は、`namespace --ns` パラメータを追加することを忘れないでください：

```
# proxmox-backup-client snapshot forget <snapshot> --ns <ns>
```

手動での削除が必要な場合もありますが、`prune` コマンドは通常、古いバックアップをシステム的に削除するために使用します。`prune` では、保持するバックアップ・スナップショットを指定できます。以下の保持オプションがあります。以下の保持オプションが使用できます：

-keep-last <N> (最後の<N>個のバックアップスナップショットを保持)

最後の<N>個のバックアップ・スナップショットを保持します。

--keep-time <N>

直近<N>時間のバックアップを保持します。1時間に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない時間はカウントされません。

--keep-daily <N>

過去<N>日間のバックアップを保持します。1日に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない日はカウントされません。

--keep-weekly (毎週) <N>

過去<N>週間分のバックアップを保持します。1週間に複数のバックアップがある場合は、最新のものだけが保持されます。バックアップのない週はカウントされません。

 注

週は月曜日に始まり、日曜日に終わります。このソフトウェアはISOの週日付システムを使用し、年末の週を正しく処理します。

--keep-monthly <N>

過去<N>ヶ月分のバックアップを保持します。1つの月に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない月はカウントされません。

--keep-yearly <N>

過去<N>年間のバックアップを保持します。1年間に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない年はカウントされません。

保持オプションは上記の順序で処理されます。各オプションは、その期間内のバックアップのみを対象とします。次のオプションでは、すでに対象となったバックアップは処理されません。古いバックアップのみを考慮します。

未完了のバックアップや不完全なバックアップは、最後に成功したバックアップよりも新しいものでない限り、`prune` コマンドによって削除されます。この場合、最後に失敗したバックアップが保持されます。

```
# proxmox-backup-client prune <group> --keep-daily 7 --keep-weekly 4 --keep-monthly 3
```

`dry-run` オプションを使用すると、設定をテストできます。これは既存のスナップショットのリストと`prune` が取るアクションを表示するだけです。

```
# proxmox-backup-client prune host/elsa --dry-run --keep-daily 1 --keep-weekly 3
```

スナップショット	キープ
host/elsa/2019-12-04T13:20:37Z	1

(次のページに続きます)

host/elsa/2019-12-03T09:35:01Z	0
host/elsa/2019-11-22T11:54:47Z	1
host/elsa/2019-11-21T12:36:25Z	0
host/elsa/2019-11-10T10:42:20Z	1

i 注

pruneコマンドもforgetコマンドも、chunk-storeのスペースを解放しません。chunk-storeにはまだデータブロックが残っています。スペースを解放するには、[ガベージ・コレクション](#)を実行する必要があります。

(前ページからの続き)

単一のスナップショットをプルーニングや削除から保護することも可能です：

```
# proxmox-backup-client snapshot protected update <snapshot> true
```

これにより、スナップショットに保護フラグが設定され、このフラグが再び削除されるまで、このスナップショットの刈り込みや手動での削除ができなくなります：

```
# proxmox-backup-client snapshot protected update <snapshot> false
```

保護されたスナップショットを持つグループが削除されると、保護されていないものだけが削除され、残りは残ります。

i 注意

このフラグはプルジョブや同期ジョブを使用しているときには同期されません。同期されたスナップショットを保護したい場合は、ターゲットバックアップサーバで再度手動で行う必要があります。

7.12 ガベージ・コレクション

pruneコマンドは、データストアからデータではなく、バックアップ・インデックス・ファイルのみを削除します。データストアから未使用的バックアップ・データを削除するには、ガベージ・コレクションを使用します。したがってガベージ・コレクション・タスクを定期的にスケジュールすることをお勧めします。ガベージコレクションの動作原理については、関連する[背景のセクション](#)で詳しく説明しています。

クライアント側からガベージコレクションを開始するには、以下のコマンドを実行してください：

```
# proxmox-backup-client garbage-collect
```

i 注

このコマンドは既存のすべてのバックアップインデックスファイルを読み込み、完全なchunk-storeに触れる必要があります。チャンクの数やディスクの速度によっては長い時間がかかることがあります。

ガベージコレクションの進行状況は以下の例のように表示されます：

```
# proxmox-backup-client garbage-collect starting garbage
collection on store store2 GC フェーズ1を開始します。
GCフェーズ2を開始 (未使用チャンクを掃引) % done : 1
、チャングカウント219
完了率：2、チャンク数：453453
```

(次のページに続きます)

(前ページから続く)

```
パーセント完了 : 99、チャンク数21188
削除されたバイト数411368505
削除されたチャンク数:203
削除されたデータバイト数327160886391 ディスクバイト
: 52767414743 (16 %)
平均スループット: 21221
平均チャンクサイズ: 2486565 TASK OK
```

ガベージコレクションはproxmox-backup-managerまたはProxmox Backup ServerのWebインターフェイスからスケジュールすることもできます。

7.13 ベンチマーク

バックアップクライアントにはベンチマークツールも付属しています。このツールは圧縮と暗号化の速度に関する様々なメトリクスを測定します。Proxmox Backupリポジトリ（リモートまたはローカル）が指定されている場合、TLSアップロード速度も測定されます。

ベンチマークはproxmox-backup-clientのbenchmarkサブコマンドで実行できます：

注意

TLSスピードテストはバックアップサーバーリポジトリが指定されている場合のみ含まれます。

```
# proxmox-backup-client benchmark 5秒間に 1517チャンクをアップロードしました。リクエストあたりの時間:
3309マイクロ秒。TLS速度: 1267.41 MB/s
SHA256速度: 2066.73 MB/s 圧縮速度: 775.11
MB/s 展開速度: 1233.35 MB/s AES256/GCM速度
: 3688.27 MB/s
```

ペリファイ速度：783.43 MB/s

名称	値
TLS (バックアップの最大アップロード速度)	1267.41 MB/s (103%)
SHA256 チェックサム計算速度	2066.73 MB/s (102%)
ZStd レベル1圧縮速度	775.11 MB/s (103%)
ZStd レベル1解凍速度	1233.35 MB/s (103%)
チャンク検証速度	783.43 MB/s (103%)
AES256 GCM 暗号化速度	3688.27 MB/s (101%)

注

出力表に示されたパーセンテージは、Ryzen 7 2700Xとの比較に対応しています。

output-formatパラメータを渡すと、デフォルトの表形式ではなくjson形式で統計情報を出力することもできます。

Proxmox veの統合

Proxmox Backup Serverは、Proxmox VEのストレージとして追加することで、Proxmox VEのスタンドアロンまたはクラスタセットアップに統合できます。

Proxmox VE 固有のドキュメントについては、『Proxmox VE 管理ガイド』の「Proxmox VE Storage - Proxmox Backup Server」のセクションも参照してください。

8.1 Proxmox VE ウェブインターフェースの使用

Proxmox VE 6.3 では、Proxmox Backup Server のネイティブ API と Web インターフェイスが統合されています。

Proxmox Backup Server は Datacenter → Storage に追加できます。

8.2 Proxmox VE コマンドラインの使用

Proxmox VE ノードで 'pbs' タイプの新しいストレージを定義する必要があります。以下の例では、ストレージ名として store2 を使用し、サーバアドレスが localhost で、user1@pbs として接続することを想定しています。

```
# pvesm add pbs store2 --server localhost --datastore store2
pvesm set store2 --username user1@pbs --password <secret> # pvesm add pbs store2 -erver localhost -atastore store2
```

i 注意

パスワードをプレーンテキストで入力したくない場合は、引数なしで --password pa-rameter を渡すことができます。これにより、コマンド入力時にパスワードの入力を求めるプロンプトが表示されます。

を入力するようになります。

バックアップサーバーが自己署名証明書を使用する場合、証明書フィンガープリントを設定に追加する必要があります。バックアップサーバ上で以下のコマンドを実行することでフィンガープリントを取得できます：

```
# proxmox-backup-manager cert info | grep Fingerprint フィンガープリント (sha256) :
```

信頼関係を確立するために、そのフィンガープリントを設定に追加してください：

```
# pvesm set store2 --fingerprint 64:d3:ff:3a:50:38:53:5a:9b:f7:50:...:ab:fe
```

これで、ストレージの状態を見ることができます：

# pvesm status --storage store2	名前	タイプ	合計	中古	使用可能	%
データストア	pbs	アクティ	3905109820	1336687816	2568422004	34.23%

Proxmox Backup ServerデータストアをProxmox VEに追加すると、環境内の他のストレージデバイスと同じ方法でVMとコンテナをバックアップできます(Proxmox VE Admin Guideを参照)：「バックアップとリストア」を参照してください。

pxarコマンドラインツール

pxar は *Proxmox ファイルアーカイブ形式(.pxar)* のアーカイブを作成および操作するためのコマンドラインユーティリティです。同様のユースケースに対応する casync ファイルアーカイブフォーマットから着想を得ています。この .pxar フォーマットは、*Proxmox Backup Server* の特定のニーズを満たすように調整されています、ハードリンクの効率的な保存このフォーマットは、高度な重複排除を実現することで削減するように設計されています。

9.1 アーカイブの作成

次のコマンドを実行して、source という名前のフォルダのアーカイブを作成します：

```
# pxar create archive.pxar /path/to/source
```

これにより、archive.pxar という新しいアーカイブが作成され、source フォルダの内容が格納されます。

i 注意

pxar は既存のアーカイブを上書きしません。ターゲット フォルダに同じ名前のアーカイブがすでに存在する場合、作成は失敗します。

デフォルトでは、pxar は特定のマウントポイントをスキップし、デバイスの境界に従いません。この設計上の決定は、バックアップ用のアーカイブ作成という主なユースケースに基づいています。を無視することは理にかなっています。

を無視することは理にかなっています。この動作を変更してデバイスの境界に従うようにするには、--all-file-systems フラグを使用します。

特定のファイルやフォルダをアーカイブから除外するには、--exclude パラメータに gitignore 形式のマッチパターンを指定します。

たとえば、.txt で終わるすべてのファイルをアーカイブから除外するには、次のようにします：

```
# pxar create archive.pxar /path/to/source --exclude '**/*.txt'
```

pxar を起動する前に、シェル自体がグロブ・パターンを展開しようとすることに注意してください。これを避けるには、すべてのグロブを正しく引用符で囲む必要があります。

複数のパターンにマッチさせるために、--exclude パラメータを複数回渡すことができます。これにより、より複雑なファイルの包含/除外動作を使用できます。しかし、このような場合は代わりに .pxarexclude ファイルを使用することをお勧めします。

例えば、特定のファイル以外のすべての .txt ファイルをアーカイブから除外したい場合です。これは、先頭に ! を付けた否定マッチパターンによって実現できます。グロブパターンはすべてソースディレクトリからの相対パスです。

```
# pxar create archive.pxar /path/to/source --exclude '**/*.txt' --exclude '!/folder/file.txt' # pxar create archive.pxar /path/to/source --exclude '**/*.txt' --exclude '!'
```

1 注

グロブ・マッチ・パターンの順序は重要です。同じパターンを並べ替えると、異なる結果になります。

`pxar` は、コマンド ラインからパラメータとして渡されたグロブ マッチ パターンのリストを、アーカイブのルートにある `.pxarexclude-cli` というファイルに格納します。アーカイブ作成時に、この名前のファイルがソース フォルダに既に存在する場合、このファイルはアーカイブに含まれず、新しいパターンを含むファイルがアーカイブに追加されます。元のファイルは変更されません。

アーカイブからファイルを除外する、より便利で永続的な方法は、グロブ マッチ パターンを `.pxarexclude` ファイルに配置することです。このファイルは、ファイルシステム・ツリーの任意のディレクトリに作成して配置することができます。これらのファイルには、1 行に 1 つのパターンを含める必要があります。

を上書きします。パターンは、指定されたディレクトリ内またはツリー内のさらに下にあるファイルの排他を制御します。動作は [バックアップの作成](#) で説明したものと同じです。

9.2 アーカイブの抽出

既存のアーカイブ `archive.pxar` は、次のコマンドで目的のディレクトリに展開されます：

```
# pxar extract archive.pxar /path/to/target
```

ターゲットが指定されない場合、アーカイブの内容は現在の作業ディレクトリに展開されます。

アーカイブの一部、単一のファイル、フォルダのみを復元するには、対応するグロブ マッチ パターンを追加パラメータとして渡すか、ファイルに格納されているパターンを使用します：

```
# pxar extract etc.pxar /restore/target/etc --pattern '**/*.conf'
```

上記の例では、アーカイブ `etc.pxar` 内のサブ フォルダで見つかったすべての `.conf` ファイルをターゲット `/restore/target/etc` にリストアします。マッチ パターンを含むファイルへのパスは、`-files-from` パラメータを使用して指定できます。

9.3 アーカイブの内容の一覧表示

アーカイブ `archive.pxar` に含まれるファイルとディレクトリを表示するには、次のコマンドを実行します：

```
# pxar list archive.pxar
```

これにより、アーカイブのルートを基準とした各ファイルやディレクトリのフルパスが表示されます。

9.4 アーカイブのマウント

`pxar` を使用すると、FUSE 経由でアーカイブをマウントし、その内容を検査することができます。`archive.pxar` という名前のアーカイブをマウントポイント `/mnt` にマウントするには、次のコマンドを実行します：

```
# pxar mount archive.pxar /mnt
```

アーカイブがマウントされると、指定したマウントポイント下でそのコンテンツにアクセスできます。

```
# cd /mnt
# ls
bin      dev   home   lib32  lib64  lost+found  mnt          メディア opt    root sbin sys usr boot  etc
lib      proc   run    srv    tmp    var
```

テープバックアップ

The screenshot shows the Proxmox Backup Server 1.0-14 interface. The left sidebar has a tree view with nodes like Dashboard, Configuration, Access Control, Remotes, Subscription, Administration, Shell, Storage / Disks, Tape Backup (which is expanded), ibm3584, gpx720, Datastore, pve-backup, remote-sync, and Add Datastore. The main content area has tabs for Changer: qpx720, Reload, Barcode Label, and Inventory. Under Changer: qpx720, there are three sections: Slots, Drives, and Import-Export Slots. The Slots section lists slots 1 through 20, each with an ID, Content (e.g., TAPES2L4, TAPES3L4, etc.), Inventory (writable (pve-backup)), and Actions (refresh, info). The Drives section shows two drives: qdrive0 (idle) and qdrive1 (idle), both labeled writable (pve-backup). The Import-Export Slots section lists slots 21 through 24.

Proxmoxテープバックアップは、データストアのコンテンツを磁気テープに保存する簡単な方法を提供します。これによりデータの安全性が向上します：

- データの追加コピー
- 異なるメディアタイプ（テープ）のデータの追加コピー、
- 追加ロケーションへのコピー（テープをオフサイトに移動可能）

ほとんどのリストアジョブでは、最後のバックアップジョブのデータのみがリストアされます。リストア要求は、データが古くなるほどさらに減少します。この点を考慮すると、テープ・バックアップはディスク使用量の削減にも役立ちます。これは特に数年間データを保持する必要がある場合に当てはまります。

テープバックアップは、保存されたデータへのランダムアクセスを提供しません。その代わりに、再度アクセスする前にデータをディスクにリストアする必要があります。また、テープを（何らかのテープ保管サービスを使用して）オフサイトに保管している場合、リストアを行う前に、テープをオンサイトに戻す必要があります。そのため、テープからのリストアはディ

スクからのリストアよりもはるかに時間がかかる可能性があることを考慮してください。

10.1 テープ技術入門

2021年現在、広く普及しているテープ・テクノロジー規格はリニア・テープ・オープン (LTO) のみです。さまざまなベンダーがLTO Ultriumテープドライブ、オートローダー、LTOテープカートリッジを提供しています。

性能と容量で若干の利点を持つ独自のドライブを提供するベンダーもいくつかあります。しかし、これらには大きなデメリットがあります：

- プロプライエタリ（単一ベンダー）
- 購入コストがはるかに高い

そのため、現在のところ、このようなドライブのテストは行っていません。

一般的に、LTOテープには次のような利点があります：

- 耐久性（30年の寿命）
- 大容量（12 TB）
- TBあたりのコストが比較的低い
- コールドメディア
- 移動可能（保管庫内で保管可能）
- 複数のベンダー（メディアとドライブの両方）
- AES-GCM暗号化エンジン内蔵

*Proxmox Backup Server*はすでに圧縮データを保存しているため、テープ圧縮機能を使用するメリットはありません。

10.2 サポートするハードウェア

*Proxmox Backup Server*は、リニアテープオープンジェネレーション5(LTO-5)以降をサポートし、ジェネレーション4(LTO-4)をベストエフォートでサポートしています。多くのLTO-4システムは動作することが知られていますが、ファームウェアのアップデートが必要なものや、*Proxmox Backup Server*で動作するために必要な機能が実装されていないものもあります。

テープ交換はSCSI Medium Changerプロトコルを使用して行われるため、最新のテープライブラリーはすべて動作するはずです。

注

私たちは Rust で書かれたカスタムのユーザースペース・テープドライバを使用しています。このドライバは、SCSI 汎用インターフェースを使用してテープドライブと直接通信します。古いLinuxカーネルテープドライバと併用すると副作用があるため、*Proxmox*テープバックアップではこのドライバを使用しないでください。

10.2.1 ドライブ性能

現在のLTO-8テープのリード/ライト速度は最大360 MB/sです。つまり、1本のテープを完全に書き込んだり、読み込んだりするのに最低9時間かかります（最高速度でも）。

このデータ速度を速める唯一の方法は、複数のドライブを使用することです。そうすれば、複数のバックアップ・ジョ

ブを並行して実行したり、他のダイブをバックアップに使用している間にリストア・ジョブを実行したりすることができます。

また、最初にデータストア（ディスク）からデータを読み込む必要があることも考慮してください。しかし、単一の回転ディスクでは、この速度でデータを配信することはできません。実際に計測したところ、最大60MB/sから100MB/s程度でしたので、LTO-8テープ1本分の12TBを読み込むのに33時間かかります。テープへの書き込みをフルスピードで行いたい場合は、ソース データストアがそのパフォーマンスを提供できることを確認してください（たとえば、SSDを使用するなど）。

10.2.2 LTO-9+に関する考慮事項

LTO-9以降、新しいメディアをドライブで初期化する必要があります。これを*Media Optimization*と呼びます。通常、メディア1枚あたり40~120分かかります。ドライブやチェンジャーのハードウェアベンダーが提供するツールを使用して、この方法でメディアを初期化することをお勧めします。一部のテープ・チェンジャーには、メディアを「一括」初期化する方法があります。

このため、テープのフォーマットはProxmox Backup Serverでは異なる方法で処理され、各フォーマット/ラベリングでの再最適化を避けることができます。Proxmox Backup Serverで初めて使用する場合、または他のプログラムで使用した後にメディアをフォーマットする場合は、ドライブ/チェンジャーの機能を使用するか、またはcliで'slow'フォーマットを使用します：

```
# proxmox-tape format --drive your-drive --fast 0
```

これにより、既存のデータが完全に削除され、*Media Optimization*が実行されます。

パーティション分割されたLTO-9メディアを'fast'メソッド(デフォルトまたは-fast 1)でフォーマットした場合、最初のパーティションのみがフォーマットされます。

10.3 用語解説

テープラベル：

テープを一意に識別するために使用します。通常、粘着性のある紙のラベルをカートリッジの前面に貼ります。さらに、ラベル・テキストをテープに磁気的に保存します（テープ上の最初のファイル）。

バーコード：

は、電子的に読み取り可能な特殊なテープラベルです。ほとんどのLTOテープ・ロボットは、LTO Ultriumカートリッジ・ラベル仕様で定義されている [Code 39](#)としてエンコードされた8文字の文字列を使用します。

このようなバーコードラベルは、カートリッジベンダーから購入するか、自分で印刷することができます。ご自身で印刷される場合は、[LTO Barcode Generator](#)アプリをご利用ください。

注意事項

物理ラベルおよび関連する粘着剤は、貼付するカートリッジの環境仕様と同等以上の環境性能を持つ必要があります。

メディア・プール：

メディア・プールとは、テープの論理コンテナです。バックアップ・ジョブは1つのメディア・プールを対象とするため、ジョブはそのプールのテープのみを使用します。プールはさらに、バックアップ・ジョブがテープにデータを追加できる期間（割り当てポリシー）とデータを保持する期間（保持ポリシー）を定義します。

メディア・セット：

連続的に書き込まれるテープのグループ（すべて同じメディア・プールから）。

テープ・ドライブ：

テープにデータを読み書きするための装置。スタンドアロン・ドライブもありますが、ドライブは通常テープ・ライブラリ内に搭載されています。

テープ・チェンジャー：テープを交換する装置：

テープ・ドライブ内のテープを交換する装置（テープ・ロボット）。通常はテープ・ライブラリーの一部。

テープ・ライブラリー：

1台または複数台のテープ・ドライブ、テープ・カートリッジを収納する多数のスロット、テープ・カートリッジ

を識別するためのバーコード・リーダー、テープを装填する自動化された方法（ロボット）を含むストレージ・デバイス。

これは一般に「オートローダー」、「テープ・ロボット」、「テープ・ジャュークボックス」とも呼ばれています。

インベントリ

インベントリには、既知のテープのリスト（追加のステータス情報付き）が保存されます。

カタログ：

メディア・カタログには、メディア・コンテンツに関する情報が格納されます。

10.4 テープ・クイック・スタート

1. テープハードウェア（ドライブおよびチェンジャー）の構成
2. 1つまたは複数のメディアプールの構成
3. テープカートリッジのラベル付け
4. 最初のテープバックアップジョブの開始

10.5 構成

グラフィカル・ユーザー・インターフェースまたはコマンドライン・インターフェースを使用して何でも設定できることに注意してください。どちらの方法でも同じ設定になります。

10.5.1 テープチェンジャー

Name	Path	Vendor	Model	Serial	Import/Export Slots
ibm3584	/dev/tape/by-id/scsi-0000078328220400	IBM	03584L32	0000078328220400	
qpx720	/dev/tape/by-id/scsi-1814C50000	QUANTUM	PX720	1814C50000	

テープ・チェンジャー（ロボット）はテープ・ライブラリーの一部です。テープ・チェンジャーには、テープ・カートリッジを保持するための多数のスロット、テープ・カートリッジを識別するためのバーコード・リーダー、テープをロードするための機械式装置などがあります。

ードするための自動化された方法が含まれています。

スタンドアロン・ドライブを使用している場合は、この手順を省略できます。

Linuxはこれらのデバイスを自動検出できます：

パス	ベンダー	モデル	シリアル
/dev/tape/by-id/scsi-CC2C52	Quantum	Superloader3	CC2C52

Proxmox Backup Serverでデバイスを使用するには、設定エントリを作成する必要があります：

```
# proxmox-tape changer create sl3 --path /dev/tape/by-id/scsi-CC2C52
```

sl3は任意の名前です。



注意
dev/tape/by-id/内の永続デバイスパス名を使用してください。dev/sg0のような名前は、再起動後に別のデバイスを指す可能性があります。



この操作は GUI からも実行でき、**Tape Backup** の **Changers** タブに移動して **Add** をクリックします。

最終的な設定は

名前	値
名前	sl3
パス	/dev/tape/by-id/scsi-CC2C52

あるいは、設定されている全てのチェンジャーデバイス(GUIの **Changers** タブで見ることができます)をリストアップすることもできます：

名前	パス	ベンダー	モデル	シリアル
sl3	/dev/tape/by-id/scsi-CC2C52	Quantum	Superloader3	CC2C52

ベンダー、モデル、シリアル番号は自動検出されますが、デバイスがオンラインの場合にのみ表示されます。セットア

ップをテストするには、次のようにしてチェンジャーデバイスのステータスを問い合わせてください：

entry-kind	entry-id	changer-id	load-ed-slot
ドライブ	0	vtape1	1
スロット	1		
スロット	2	vtape2	
...	...		

(次のページに続く)

スロット	1 6	
------	-----	--

(前ページから続く)

テープ・ライブラリは通常、いくつかの特別なインポート/エクスポート・スロット（「メール・スロット」とも呼ばれます）を備えています。これらのスロットの内側にあるテープには外部からアクセスでき、ライブラリへのテープの追加や取り出しが簡単に行えます。これらのテープは「オフライン」と見なされるため、バックアップ・ジョブはこれらを使用しません。これらの特別なスロットは

は自動検出され、ステータス・コマンドでインポート/エクスポート・スロットとしてマークされます。

小規模なテープ・ライブラリには、このようなスロットがないものがあることは注目に値します。メール・スロット」と呼ばれるものもありますが、そのスロットは単にグリッパーからテープをつかむためのものです。ロボットが他のことをしている間、メディアを保持することはできません。また、SCSIインターフェイス上で "Mail Slot" を公開しないので、ステータス出力には表示されません。

回避策として、通常のスロットのいくつかをエクスポートスロットとしてマークすることができます。ソフトウェアはそれらのスロットを実際のインポート/エクスポート・スロットのように扱い、それらのスロット内のメディアは「オフライン」（バックアップに利用できない）と見なされます：

```
# proxmox-tape changer update sl3 --export-slots 15,16
```

この後、ステータス出力で人工的なインポート/エクスポートスロットを見ることができます：

エントリの種類	entry-id	changer-id	load ed slot
ドライブ	0	vtape1	1
インポート・エクスポート	1 5		
インポート・エクスポート	1 6		
スロット	1		
スロット	2	vtape2	
...	...		
スロット	1 4		

高度なオプション

すべてのテープチェンジャーが同じ動作をするわけではないので、時には高度なオプションを設定する必要があります。

現在、以下のようなものがあります：

- *eject-before-unload*：これはドライブからアンロードする前にテープをイジェクトする必要があるチェンジャーのために必要です。

これらのオプションは *proxmox-tape* で以下のように設定できます：

```
# proxmox-tape changer update sl3 --eject-before-unload true
```

10.5.2 テープドライブ

The screenshot shows the Proxmox Tape Backup interface. On the left, there's a sidebar with navigation links like Dashboard, Configuration, Access Control, Remotes, Subscription, Administration, Shell, Storage / Disks, and Tape Backup. The Tape Backup section is currently selected. At the top, there are tabs for Content, Inventory, Changers, Drives (which is selected), Media Pools, Encryption Keys, and Backup Jobs. Below the tabs is a table with columns: Name, Path, Vendor, Model, Serial, and Drive Num... A toolbar below the table has Add, Edit, and Remove buttons. The main table displays two entries under 'Changer ibm3584 (4 items)': drive0 and drive1, both connected to IBM ULT3580-TD4 tape drives. Another entry under 'Changer qpx720 (2 items)': qdrive0 and qdrive1, both connected to IBM ULT3580-TD1 tape drives.

Linuxはテープドライブを自動検出することができます：

The terminal window shows the command `# proxmox-tape drive scan` and its output:

```
# proxmox-tape drive scan
| パス | ベンダー | モデル | シリアル |
| /dev/tape/by-id/scsi-12345-sg | IBM | ult3580-td4 | 1 2 3 4 5 |
```

Below the terminal is an 'Add: Drive' dialog box:

Name:	<input type="text" value="qdrive0"/>
Changer:	<input type="text" value="qpx720"/>
Drive Number:	<input type="text" value="0"/>
Path:	<input type="text" value="/dev/tape/by-id/scsi-01E"/>

A blue 'Add' button is at the bottom right of the dialog.

Proxmoxでそのドライブを使用するには、設定エントリを作成する必要があります。これはGUIの**Tape Backup -> Drives**から行うか、以下のコマンドを使用します：

```
# proxmox-tape drive create mydrive --path /dev/tape/by-id/scsi-12345-sg
```

メモ

`dev/tape/by-id/`内の永続デバイスパス名を使用してください。`dev/sg0`のような名前は、リブート後に別のデバイスを指す可能性があります。

テープライブラリーがある場合は、関連するチェンジャー・デバイスも設定する必要があります：

```
# proxmox-tape drive update mydrive --changer sl3 --changer-drivenum 0
```

changer-drivenumはテープライブラリーが複数のドライブを含む場合にのみ必要です（**changer status**コマンドはすべてのドライブ番号を一覧表示します）。

最終的な設定は

# proxmox-tape ドライブ設定 mydrive	
名前	値
名前	mydrive
パス	/dev/tape/by-id/scsi-12345-sg
チェンジャー	sl3



changer-drivenum 値 0 はデフォルトのため、コンフィギュレーションには保存されません。

設定されている全てのドライブを一覧表示するには

# proxmox-tape drive list				
名前	パス	changer	vendor	model
mydrive	/dev/tape/by-id/scsi-12345-sg	sl3	IBM	ult3580-td4 1 2 3 4 5

ベンダー、モデル、シリアル番号は自動検出され、デバイスがオンラインの場合のみ表示されます。テストするには、

次のようにしてドライブのステータスを照会します：

# proxmox-tape status --drive mydrive	
名前	値
ブロックサイズ	0
密度	LTO4
圧縮率	1
バッファモード	1
アラートフラグ	(空)
ファイル番号	0
ブロック番号	0
製造	2019年12月13日(金)01:00:00
書き込みバイト数	501.80 GiB
バイトリード	4.00 MiB
ミディアムパス	20
ミディアムウェアアウト	0.12
ボリュームマウント	2

1 注

ブロックサイズは常に0（可変ブロックサイズ・モード）であるべきです。いずれにせよ、これがデフォルトです。

10.5.3 メディア・プール

Name ↑	Allocation ...	Retention ...	Encryption	Comment
pve-backup	continue	keep	Yes	
remote-backup	continue	keep	No	

メディア・プールはテープの論理コンテナです。バックアップ・ジョブは1つのメディア・プールを対象とするため、ジョブはそのプールのテープのみを使用します。

メディア・セット

メディア・セットは、継続的に書き込まれるテープのグループで、大きなプールを復元可能な小さな単位に分割するため使用されます。つまでは複数のバックアップ・ジョブがメディア・セットに書き込み、順序付けられたテープ・グループを作成します。メディア・セットは一意のIDで識別されます。このIDとシーケンス番号は、そのセットの各テープ（テープ・ラベル）に保存されます。

メディア・セットはリストアタスクの基本単位です。つまり、メディア・セットの内容をリストアするには、セット内のすべてのテープが必要です。メディア・セット内のデータは完全に重複排除されます。

メディア・セット割り当てポリシー

プールはさらに、バックアップ・ジョブがメディア・セットにデータを追加できる期間を定義します。次の設定が可能です：

- 現在のメディア・セットを使用しようとしています（続行）。

この設定は1つの大きなメディア・セットを生成します。これは非常にスペース効率が良いのですが（重複排除、

重複排除、未使用領域なし)、リストアジョブはセット内のすべてのテープを読み取る必要があるため、リストア時間が長くなる可能性があります。

注

データはメディアセット内で完全に重複排除されます。これは、データがセット内のテープにランダムに分散されることも意味します。したがって、単一のVMをリストアする場合でも、メディアセット内のすべてのテープからデータを読み取る必要がある場合があります。

また、メディアセットが大きいと、破損したテープが1本でもあるとリストアが失敗するため、エラーが発生しやすくなります。

使用シナリオ：主にテープ・ライブラリで使用します。exportオプションを指定してバックアップ・ジョブを実行し、手動で新しいセットを作成します。

注記

保持期間は、新しいメディア・セットが存在するときに開始します。

- 常に新しいメディア・セットを作成する (always)。

この設定では、バックアップ・ジョブごとに新しいメディア・セットが作成されます。最後のセットのメディアが完全に書き込まれず、残りの領域が未使用のままになる可能性があるためです。

利点は、最小サイズのメディア・セットを作成できることです。小さなセットは取り扱いが簡単で、オフサイトのデータ保管庫への移動がより便利になり、リストアもはるかに高速になります。

注

保存期間は、メディアセットの作成時間から開始します。

- 指定したカレンダーイベントがトリガーされた時に新しいセットを作成します。

これにより、systemdのカレンダーイベント指定 (systemd.time manpage を参照)と同じように、時点を指定することができます。

例えば、weekly(またはMon *-*-* 00:00:00)と指定すると、毎週新しいセットを作成します。これはスペース効率とメディア数のバランスをとるためです。

注意

保持期間は、次のメディアセットの作成時刻、またはそれが存在しない場合は、現在のメディアセットの開始時刻の次にカレンダーイベントがトリガーされたときに開始されます。

さらに、次のイベントは新しいメディアセットを割り当てる可能性があります：

- 必要なテープがオフライン (テープライブラリを使用)。
- 現在のセットに破損または引退したテープが含まれている場合。
- メディアプールの暗号化が変更されました

- データベースの整合性エラー（インベントリに必要なメディア情報が含まれていない、または矛盾する情報（古いデータ）が含まれている場合など）。

保持ポリシー

データの保存期間を定義します。

- 常にメディアを上書きします。
 - 指定された期間、データを保護します。
- `systemd like time spans` を使って期間を指定します ([systemd.time manpage](#) を参照)。
- データを上書きしません。

ハードウェア暗号化

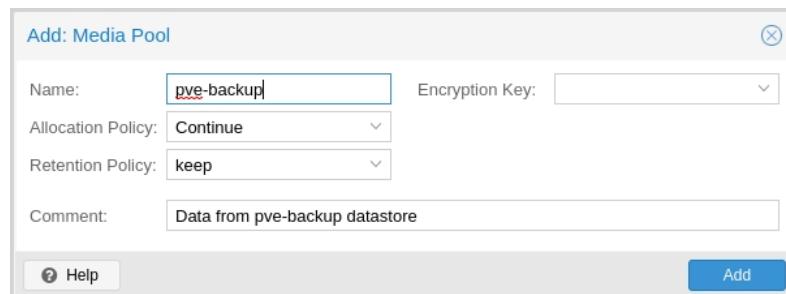
LTO-4 (またはそれ以降) テープドライブはハードウェア暗号化をサポートしています。暗号化を使用するようにメディアプールを設定すると、テープに書き込まれるすべてのデータは設定されたキーを使用して暗号化されます。

これにより、権限のないユーザがメディアからデータを読み取れなくなります。たとえば、オフサイトに輸送中にテープを紛失した場合などです。

注

バックアップクライアントもデータを暗号化する場合、テープ上のデータは二重に暗号化されます。

パスワードで保護されたキーは各メディアに保存されるため、パスワードを使ってキーを復元することができます。キーの復元が必要になった場合に備えて、パスワードは必ず覚えておいてください。



新しいメディアプールを作成するには、GUIの**Tape Backup -> Media Pools**から追加するか、以下のコマンドを入力してください：

```
// proxmox-tape pool create <name> --drive <string> [OPTIONS].
# proxmox-tape pool create daily --drive mydrive
```

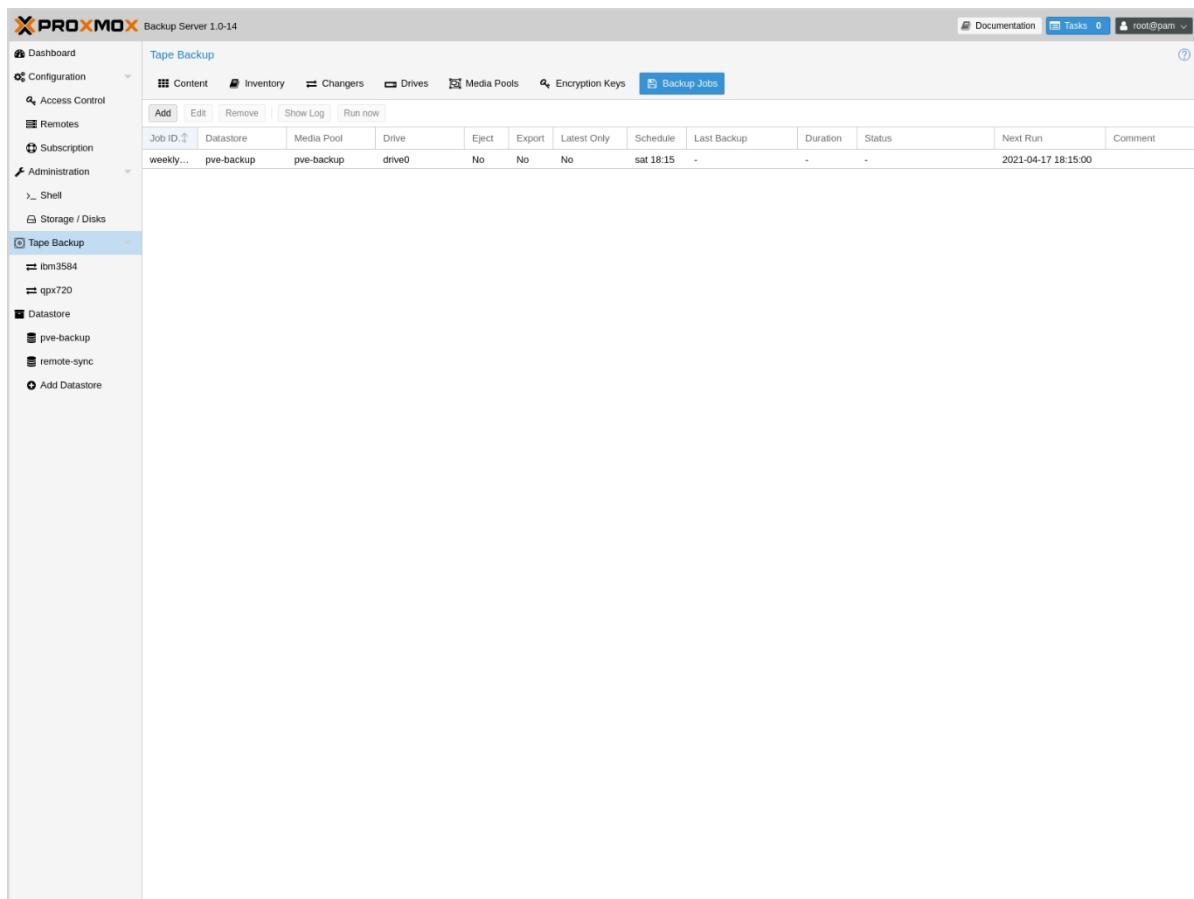
後で update コマンドを使用して追加のオプションを設定できます：

```
# proxmox-tape pool update daily --allocation daily --retention 7days
```

設定されているすべてのプールを一覧表示するには

```
# proxmox-tape pool list
|名前|ドライブ|アロケーション|リテンション|テンプレート|
|daily|mydrive|daily|7days||
```

10.5.4 テープバックアップジョブ



テープバックアップを自動化するには、特定のタイムスケジュールに基づいてデータストアのコンテンツをメディアアプールに書き込むテープバックアップジョブを構成します。必要な設定は次のとおりです：

- ストアバックアップするデータストア
- プール：メディアアプール - そのプールのテープカートリッジのみが使用されます。
- ドライブ：テープ・ドライブ。
- スケジュール：ジョブ・スケジュール（「カレンダー・イベント」を参照）

例えば、データストア `vmstore1` のテープバックアップジョブを設定するには、次のようにします：

```
# proxmox-tape backup-job create job2 --store vmstore1 --pool yourpool --drive youdrive --schedule: ジョブのスケジュール。
- プール yourpool - ドライブ youdrive - スケジュール daily
```

バックアップには、デフォルトでバックアップグループのすべてのスナップショットが含まれます。最新のスナップショットのみを含めるには、

latest-only

フラグを設定できます：

バックアップに最新のスナップショットのみを含めるには、以下のように `# proxmox-tape backup-job update job2 --latest-only` フラグを設定します。

バックアップジョブはテープ要求の通知やエラーの報告に電子メールを使用できます。通知ユーザは

```
# proxmox-tape backup-job update job2 --notify-user root@pam
```

注

電子メールアドレスはユーザーのプロパティです（[ユーザー管理を参照](#)）。

バックアップ後にドライブからテープをイジェクトすると便利な場合があります。スタンダードアロン・ドライブの場合、`eject-media` オプションを使用するとテープがイジェクトされ、次のバックアップでテープが使用できないようになります（誰かが手動でテープを再度ロードしない限り）。テープ・ライブラリの場合、このオプションはテープを空きスロットにアンロードします。

を空きスロットにアンロードします：

```
# proxmox-tape backup-job update job2 --eject-media
```

i 注

失敗したジョブの場合、テープはドライブ内に残ります。

テープ・ライブラリの場合、`export-media-set` オプションはメディア・セットからすべてのテープをエクスポート・スロットに移動し、次のバックアップでテープが使用できないようにします。オペレータは、これらのテープを保管庫に移動します。

```
# proxmox-tape backup-job update job2 --export-media-set
```

i 注意

`export-media-set` オプションを使用すると、現在のセットのテープがオンラインでなくなるため、新しいメディアセットを強制的に開始できます。

バックアップジョブを手動で実行することも可能です：

```
# proxmox-tape backup-job run job2
```

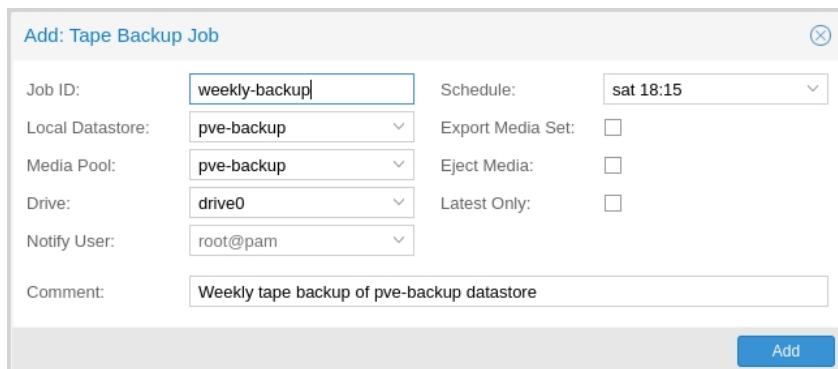
ジョブを削除するには

```
# proxmox-tape backup-job remove job2
```

デフォルトでは、データストアのすべての（再帰的な）名前空間がテープバックアップに含まれます。`ns` で単一の名前空間を指定し、`max-depth` で深さを指定できます。例えば

```
# proxmox-tape backup-job update job2 --ns mynamespace --max-depth 3
```

`max-depth` が指定されない場合、すべての再帰名前空間が含まれます。



これと同じ機能がGUIにもあり、**Tape Backup** の **Backup Jobs** タブで、*Local Datastore* はバックアップするデータストアに関連し、*Media Pool* はバックアップ先のプールです。

10.6 管理

`proxmox-tape`コマンドラインツールの多くのサブコマンドは`-drive`というパラメータを取り、作業するテープドライブを指定します。便宜上、これを環境変数

環境変数で設定できます：

```
# export PROXMOX_TAPE_DRIVE=mydrive
```

この場合、コマンドから`--drive`パラメータを省略することができます。ドライブに関連するチェンジャー・デバイスがある場合は、チェンジャー・デバイスを必要とするコマンドから`changer`パラメータを省略することもできます。

デバイスを必要とするコマンドでは`changer`パラメータを省略することもできます：

```
# proxmox-tape changer status
```

は`mydrive`ドライブに関連付けられているチェンジャーデバイスのチェンジャーのステータスを表示します。

10.6.1 テープのラベル

デフォルトでは、テープカートリッジは全て同じように見えます。まず、人間が読めるテキストが書かれた付箋紙のラベルをカートリッジに貼ります。

テープライブラリを使用する場合は、LTO Ultrium Cartridge Label Specification で定義されている [Code 39](#) としてエンコードされた 8 文字の文字列を使用する必要があります。このようなバーコードラベルは、カートリッジベンダーから購入するか、自分で印刷することができます。当社の [LTO Barcode Generator](#) アプリを使用して印刷できます。

次に、ソフトウェアがテープを一意に識別できるように、同じラベル・テキストをテープに書き込む必要があります。

スタンダードアロンドライブの場合は、新しいテープカートリッジを手動でドライブに挿入して実行します：

```
# proxmox-tape label --changer-id <label-text> [--pool <pool-name>]
```

どのプールでもテープを使用できるようにするには`--pool`引数を省略できます。

注意

安全上の理由から、テープにデータが含まれている場合、このコマンドは失敗します。上書きする場合は、まずテープを消去してください。

ラベルを読み返して成功を確認できます：

```
# proxmox-tape read-label
名前          値
| チェンジャーID | vtape1
| uuid          | 7f42c4dd-9626-4d89-9f2b-c7bc6da7d533
| ctime         | 水 Jan  6 09:07:51 2021
| プール         | デイリー
| media-set-uuid | 00000000-0000-0000-0000-000000000000
| media-set-ctime | Wed Jan  6 09:07:51 2021
```

注

すべてのゼロを使用する`media-set-uuid`は、空のテープ(どのメディア・セットでも使用されていない)を示します。

テープ・ライブラリーがある場合は、まず粘着性のあるバーコード・ラベルをテープ・カートリッジに貼ります。次に、空のテープをライブラリにロードします。そうすれば、ラベルの付いていないテープに1つのコマンドでラベルを付けることができます：

```
# proxmox-tape barcode-label [--pool <pool-name>]
```

10.6.2 テープバックアップの実行

手動でバックアップジョブを実行するには、GUIで*Run Now*をクリックするか、コマンドを使用します：

```
# proxmox-tape backup <store> <pool> [OPTIONS] を使用します。
```

以下のオプションが利用可能です：

--eject-media	ジョブ完了時にメディアを排出します。 通常、使用後にテープをイジェクトするのは良い習慣です。これにより、ドライブからテープがアンマウントされ、テープが埃まみれになるのを防ぎます。
--export-media-set	ジョブ完了時にメディア・セットをエクスポートします。 バックアップ・ジョブが正常に完了すると、使用済みのメディア・セットからすべてのテープがインポート/エクスポート・スロットに移動します。その後、オペレータはこれらのテープをピックアップしてメディア保管庫に移動できます。
-ns	バックアップするネームスペース。 特定のネームスペースだけをバックアップする場合に使用します。省略すると、ルート・ネームスペースが想定されます。
--max-depth	ネームスペースを再帰する深さ。 0は、再帰をまったく行わないことを意味します(指定されたネームスペースのみ)。省略すると、すべてのネームスペースが再帰されます(指定されたネームスペースより下)。

10.6.3 テープからのリストア

リストアはメディアセット単位で行われるため、まずリストアしたいデータがどのメディアセットに含まれているかを調べる必要があります。この情報はメディアカタログに保存されています。メディアカタログがない場合は、まずそれをリストアする必要があります。データを見つけるにはカタログが必要ですが、完全なメディアセットのリストアにはメディアカタログは必要ないことに注意してください。

以下のコマンドはメディアの内容をリストアップします(カタログから)：

# proxmox-tape media content			
label-text	pool	media-set-name	seq-nr スナップショット
TEST01L8	p2	Wed Jan 13 13:55:55 2021 0	vm/2021/2021-01-11T10:43:48Z
→9da37a55-aac7-4deb-91c6-482b3b675f30			
...

リストアジョブはメディアセットからデータを読み込み、データディスク(データストア)に戻します：

```
// proxmox-tape restore <メディアセット-uuid> <データストア>
# proxmox-tape restore 9da37a55-aac7-4deb-91c6-482b3b675f30 mystore
```

シングルスナップショットリストア

メディアセット全体をリストアするのではなく、テープから特定のスナップショットだけをリストアする必要がある場合があります。これはsnapshotsパラメータで実現できます：

```
// proxmox-tape restore <media-set-uuid> <datastore> [<snapshot>].
# proxmox-tape restore 9da37a55-aac7-4deb-91c6-482b3b675f30 mystore sourcestore:host/hostname/
→2022-01-01T00:01:00Z
```

これはまずスナップショットを一時的な場所にリストアし、次に関連するチャンクアーカイブをリストアし、最後にスナップショットデータをターゲットデータストアにリストアします。

1回のリストア操作で複数のスナップショットをリストアするために、`snapshot`パラメータを複数回渡すことができます。

i 注意

シングル・スナップショット・リストアを使用する場合、テープを複数回トラバースする必要があり、一度に多数のスナップショットをリストアする場合、データストア全体をリストアするよりも時間がかかることがあります。

名前空間

リストア中にメディア・セットから特定のネームスペースを選択してマッピングすることもできます。これは、`namespaces`パラメータを使用することで可能です。パラメータの形式は以下のとおりです：

```
store=<source-datastore>[,source=<source-ns>][,target=<target-ns>][,max-depth=<depth>]。
```

`source`または`target`が指定されていない場合は、ルート・ネームスペースが想定されます。`max-depth`が指定されていない場合は、ソース・ネームスペースが完全に再帰されます。

リストアコマンドの例です：

```
# proxmox-tape restore 9da37a55-aac7-4deb-91c6-482b3b675f30 mystore --namespaces_
→store=sourcedatastore,ソース=ns1,ターゲット=ns2,最大深度2=
```

このパラメータは複数回指定できます。また、`snapshots`パラメータと組み合わせて、これらのスナップショットのみをリストアし、それらを異なるネームスペースにマッピングすることもできます。

10.6.4 インベントリの更新

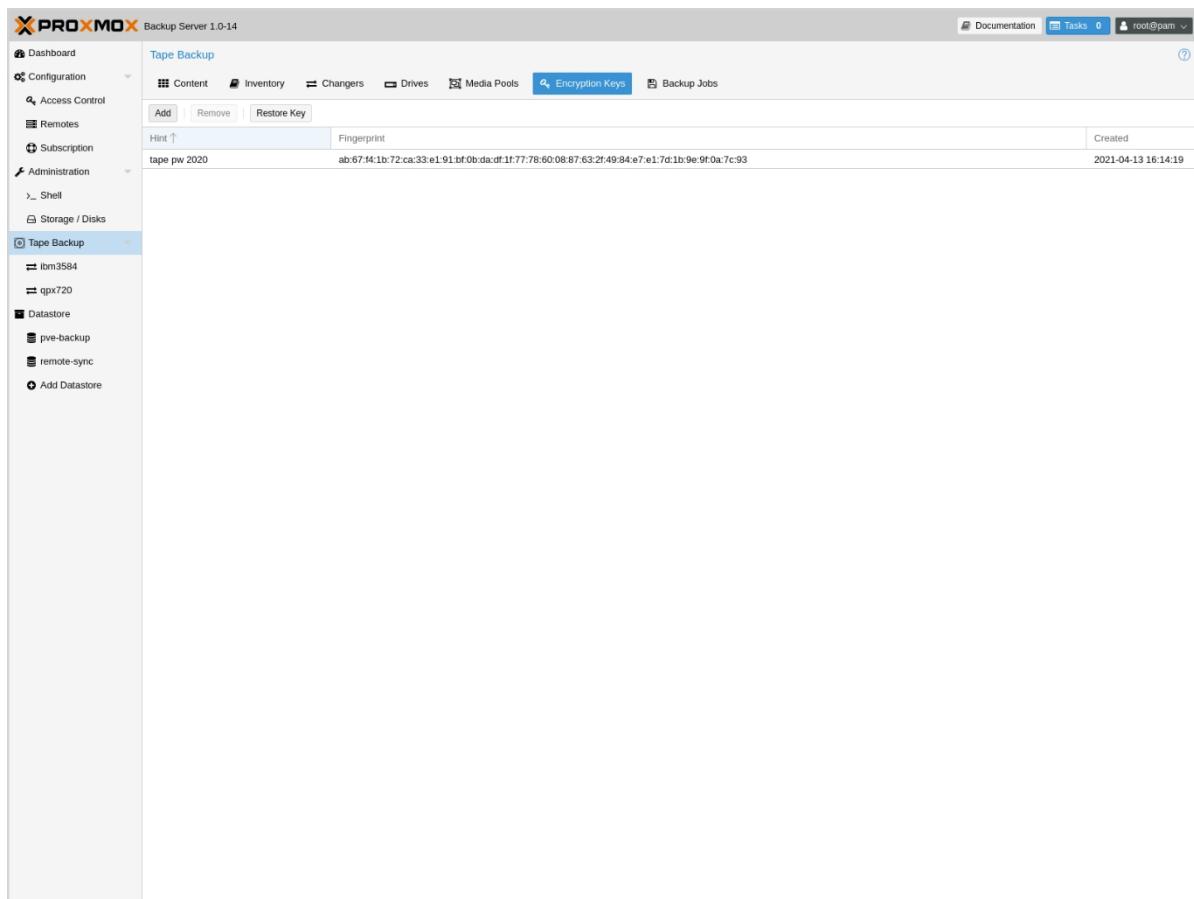
10.6.5 カタログのリストア

既存のテープからカタログをリストアするには、テープをドライブに挿入して実行します：

```
# ブロックスマックステープカタログ
```

既存のカタログがなくてもテープからリストアすることができます。この場合、カタログは自動的に作成されます。

10.6.6 暗号化キーの管理



Proxmox Backup Serverには、バックアップサーバ上で暗号化キーを処理するためのインターフェイスも用意されています。

暗号化キーは、**GUI**の【テープバックアップ】->【暗号化キー】セクションで管理できます。

または**proxmox-tape key**コマンドラインツールから管理できます。コマンドラインから新しい暗号化キーを作成するにはコマンドラインから作成します：

```
# proxmox-tape key create --hint "tape pw 2020" テープ暗号化キーの
パスワード: *****
パスワードの確認*****  

→"14:f8:79:b9:f5:13:e5:dc:bf:b6:f9:88:48:51:81:dc:79:bf:a0:22:68:47:d1:73:35:2d:b6:20:e1:7f:f5:0f
→"
```

既存の暗号化キーをリストアップします：

```
# proxmox-tape key list
| フィンガープリント | ヒント |
| 14:f8:79:b9:f5:13:e5:dc:... :b6:20:e1:7f:f5:0f | tape pw 2020 |
```

暗号化キーの詳細を表示するには

```
# proxmox-tape key show 14:f8:79:b9:f5:13:e5:dc:...:b6:20:e1:7f:f5:0f
| 名前 | 値 |
| kdf | スクリプト |
| 作成 | 1月23日（土）14:47:21 2021 |
| 変更 | 1月23日（土）14:47:21 2021 |
| フィンガープリント | 14:f8:79:b9:f5:13:e5:dc:...:b6:20:e1:7f:f5:0f |
```

(次のページに続きます)

ヒント	テープpw 2020
-----	------------

(前ページから続く)

paperkeyサブコマンドは、テープ暗号化キーのQRエンコード・バージョンを作成するために使用できます。以下のコマンドは**paperkey**コマンドの出力をテキストファイルに送り、簡単に印刷できるようにします：

```
proxmox-tape key paperkey <fingerprint> --output-format text> qrkey.txt
```

暗号化キーの復元

鍵の生成に使用したパスワードを使用して、テープから暗号化鍵を復元することができます。まず、復元したいテープをドライブにロードします。それから実行してください：

```
# proxmox-tape key restore
テープ暗号化キーのパスワード : *****
```

パスワードが正しければ、キーはデータベースにインポートされます。それ以降のリストア・ジョブは、利用可能なキーを自動的に使用します。

10.6.7 テープのクリーニング

LTOテープドライブは定期的なクリーニングが必要です。これは、クリーニング・カートリッジをドライブにロードすることによって行われます。

テープ・ライブラリの場合、クリーニング・カートリッジは "CLN" で始まる特別なラベルで識別されます。例えば、私たちのテープライブラリでは、スロット3の中にクリーニングカートリッジがあります：

```
# proxmox-tape changer status sl3
+-----+-----+-----+-----+
| エントリ-種類 | entry-id | changer-id | l o a d e d - s l o t |
+-----+-----+-----+-----+
| ドライブ      | 0        | vtape1     | 1           |
+-----+-----+-----+-----+
| スロット      | 1        |             |             |
+-----+-----+-----+-----+
| スロット      | 2        | vtape2     |             |
+-----+-----+-----+-----+
| スロット      | 3        | CLN001CU   |             |
+-----+-----+-----+-----+
| ...          | ...      |             |             |
+-----+-----+-----+-----+
```

クリーニングを開始するには

```
# ブロックスマックスステープクリーン
```

このコマンドは以下のことを行います：

- クリーニングテープの検索（スロット3）
- ドライブから現在のメディアをアンロードします（スロット1に戻ります）。
- クリーニングテープをドライブにロード
- ドライブのクリーニング操作を実行します。
- クリーニング・テープを（スロット3に）アンロードします。

10.7 WORMテープ

WORM (write once, read many) テープは、削除や上書きができない特別なカートリッジです。これは、法的または保護目的で有用です。

WORMテープを使用する場合は、*保持ポリシーを持つメディア・プール*を使用する必要があります。そうしないと、バックアップジョブがテープを消去または上書きしようとして失敗する可能性があります。

Proxmox Backup Serverは、通常のテープとWORMテープを区別しません。混乱を避けるため、WORMバックアップには別の命名スキームを使用し、専用のメディアプールを使用してください。同じメディアプールにWORMテープと非WORMテープを混在させないでください。

10.8 セットアップ例

メディアプールとスケジュールを管理するための設定例をいくつか紹介します。これは網羅的なリストではなく、便利な設定の組み合わせはもっとたくさんあります。

10.8.1 単一の継続メディアセット

最もシンプルな設定: メディアセットを常に継続し、期限切れにならないようにします。

割り当てポリシー

継続

保持ポリシー:

保持

このセットアップには、管理が簡単で、重複排除のメリットを可能な限り再利用できるという利点があります。しかし、冗長性もないため、1つのテープに障害が発生すると、そのテープのチャンクを参照するすべてのバックアップが使用できなくなります。

新しいメディアセットを手動で開始したい場合は、セットの現在書き込み可能なメディアを「フル」に設定するか、場所をオフサイトの保管庫に設定します。

10.8.2 平日スキーム

もう少し複雑なスキームでは、たとえば月曜日から金曜日まで、平日ごとに独立したテープやメディアセットを持つことが目標になります。これは、「月曜日」、「火曜日」など、曜日ごとに独立したメディア・プールを持つことで解決できます。

割り当てポリシー:

月曜日'プールには'mon'、火曜日'プールには'tue'などとします。

保持ポリシー:

上書き

各プールには、対応する平日に1つ以上のテープバックアップジョブがあるはずです。このスキームは、平日ごとに1つのメディア・セットで非常に管理しやすく、オフサイトに簡単に移動できます。

10.8.3 異なるポリシーを持つ複数のプール

複数のメディアプールを異なる割り当ておよび保持ポリシーで構成する複雑なセットアップも可能です。

たとえば、2つのメディアプールがあります。最初のプールは、毎週割り当てられ、数週間保持されます:

割り当てポリシー:

月

保持ポリシー:

3週間

2番目のプールは、有効期限なしの年間割り当てで構成されています：

割り当てポリシー：

毎年

保持ポリシー 保持

適切なブルーニング設定とテープ・バックアップ・スケジュールを組み合わせることで、一部のバックアップの長期保存を実現する一方、最近のバックアップは、およそ4週間ごと（つまり、3週間に現在の週を加えた期間）に期限切れになる小さなメディア・セットに保存します。

10.9 災害復旧

大規模な災害が発生した場合、重要なデータやサーバ全体が破壊されたり、少なくともバックアップサーバを含むすべてをバックアップから復元しなければならないほど損傷することがあります。このような場合、Proxmox Backup Serverをセットアップし、テープバックアップからすべてをリストアするために、以下のステップバイステップのガイドが役立ちます。

以下のガイドでは、Web GUIとコマンドラインツールの両方を使用して必要な手順を説明します。コマンドラインツールの概要については、「[コマンドラインツール](#)」を参照してください。

10.9.1 データストアのセットアップ

インストールの章で説明したように、新しいProxmox Backup Serverをセットアップしたら、まずデータストアをセットアップしてテープをリストアできるようにします：

1. **Administration -> Storage / Disks**に移動し、データストアとして使用するディスクが表示されていることを確認します。
2. ディレクトリ】または[ZFS]タブで、それぞれディレクトリの作成またはZFS zpoolの作成を選択できます。ここで、新しく作成したディレクトリまたはZFS zpoolをデータストアとして直接追加することもできます。

あるいは、[proxmox-backup-manager](#)を使用して同じタスクを実行することもできます。詳細については、[データストアのドキュメント](#)を確認してください。

10.9.2 テープドライブのセットアップ

1. リストアしたいメディアと一致する正しく動作するテープドライブやチェンジャーがあることを確認してください。
2. テープ・チェンジャーとテープ・ドライブをバックアップ・サーバに接続します。これらはLinuxによって自動的に検出されるはずです。利用可能なドライブのリストは

# proxmox-tape drive scan			
パス	ベンダー	モデル	シリアル
/dev/tape/by-id/scsi-12345-sg	IBM	ult3580-td4	12345

利用可能なチェンジャーのリストは以下のようにして取得できます：

# proxmox-tape changer scan			
パス	ベンダー	モデル	シリアル
/dev/tape/by-id/scsi-CC2C52	Quantum	Superloader3	CC2C52

詳細については、[テープチェンジャーとテープドライブの章](#)をお読みください。

-
3. テープチェンジャーがある場合は、Proxmox Backup Server の Web インタフェースで、**Tape Backup -> Changers** に移動して追加します。コマンドラインを使用する例については、[テープチェンジャーの章](#)をお読みください。Linuxでチェンジャーが正しく検出されると、リストにチェンジャーが表示されます。

- ウェブ・インターフェイスで、[テープ・バックアップ] → [ドライブ] に進み、テープの読み取りに使用するテープ・ドライブを追加します。コマンドラインを使用する例については、[テープ・ドライブの章](#)をお読みください。テープ・ドライブがLinuxで正しく検出されると、ドライブがリストに表示されます。ドライブにテープ・チェンジャーもある場合は、チェンジャーも選択し、正しいドライブ番号を割り当ててください。

10.9.3 テープからのデータのリストア

以下のガイドでは、テープからデータをリストアするために必要な手順を説明します。コマンドラインの詳細については、[proxmox-tapeツールのドキュメント](#)をお読みください。

テープからデータをリストアするには、次のようにします：

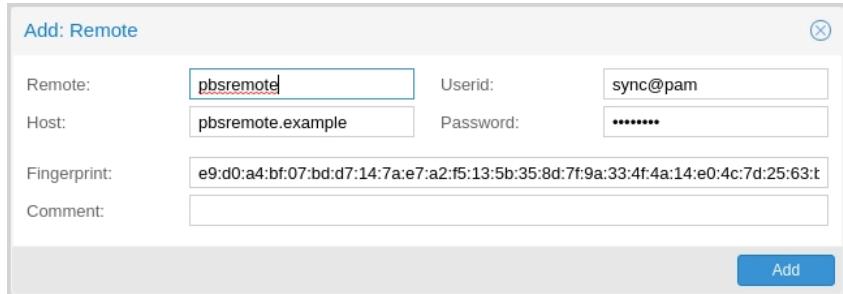
- 最初のテープ（ラベルに表示されている）をテープドライブに挿入するか、テープチェンジャーがある場合はテープチェンジャーを使用してテープを正しいドライブに挿入します。Web GUI を使用して、チェンジャーを選択してテープ・ドライブ間でテープをロードまたは転送することもできます。
- バックアップが暗号化されている場合は、暗号化キーも復元する必要があります。暗号化キー] タブで [キーの復元] を押します。詳細またはコマンドラインを使用する例については、[暗号化キー管理の章](#)をお読みください。
- データを復元する手順は、スタンドアロン・テープ・ドライブを使用しているか、チェンジャーを使用しているかによって若干異なります：
 - チェンジャーの場合、手順は簡単です：
 - リストアするメディアセットからすべてのテープを挿入します。
 - ウェブGUIでチェンジャーをクリックし、[インベントリ] をクリックして、[カタログの復元] が選択されていることを確認し、[OK] を押します。
 - スタンドアロン・ドライブの場合、手順は次のようになります：
 - メディア・セットの最初のテープを挿入します。
 - [カタログ] をクリックします。
 - テープを取り出し、メディア・セットの残りのテープについて手順を繰り返します。
- テープバックアップに戻ります。**Content**] タブで [Restore] を押し、目的のメディア・セットを選択します。リストアするスナップショットを選択し、[Next] を押して、ドライブとターゲットのデータストアを選択し、[Restore] を押します。
- データがリストアされたデータストアに移動すると、[コンテンツ] タブの下にリストアされたスナップショットが表示されます。別のマシンからバックアップにアクセスするには、バックアップサーバへのアクセスを設定する必要があります。設定
→ **Access Control**に進み、新しいユーザを作成するか、新しいAPIトークンを作成します（APIトークンは、トークンが漏洩した場合に簡単に失効させることができます）。**Permissions**（アクセス許可）の下に、必要なアクセス許可を追加します。
例：**DatastoreBackup**。
- 仮想マシン、コンテナ、またはファイルのリストアを実行できるようになりました。以下のオプションがあります：
 - Proxmox 製品をベースにしていない Linux ディストリビューションでファイルをリストアしたい場合、またはコマンドラインツールを使用したい場合は、[データのリストアの章](#)で説明したように、[proxmox-backup-client](#) を使用できます。新しく作成した API トークンを使ってデータにアクセスできます。その後、個々のファイルをリストアしたり、アーカイブをシステムにマウントしたりできます。
 - Proxmox VEサーバ上の仮想マシンまたはコンテナをリストアする場合は、バックアップサーバのデータス

トアをストレージとして追加し、[Backups]に進みます。ここで、VMとコンテナを設定を含めてリストアできます。Proxmox VE でのバックアップのリストアに関する詳細は、Proxmox VE ドキュメントの Restore の章を参照してください。

リモートと同期の管理

11.1 リモート

リモートとは、Proxmox Backup Serverを個別にインストールし、そのインストール上のユーザーを指し、同期ジョブでデータストアをローカルのデータストアに同期することができます。リモートの設定は、Web インターフェイスの【設定】->【リモート】で行います。あるいはサブコマンドを使用することもできます。リモートの設定情報は/etc/proxmox-backup/remote.cfgに保存されます。



リモートを追加するには、リモートのホスト名またはIPアドレス、リモートのユーザIDとパスワード、証明書のフィンガープリントが必要です。フィンガープリントを取得するには、リモートで proxmox-backup-manager cert info コマンドを使用するか、リモートのウェブインターフェイスで **Dashboard** に移動し、**Show Fingerprint** を選択します。

```
# proxmox-backup-manager cert info |grep Fingerprint
フィンガープリント (sha256): 64:d3:ff:3a:50:38:53:5a:9b:f7:50:...:ab:fe
```

上記の情報を使って、**Remotes**設定パネルから、またはコマンドを使ってリモートを追加できます：

```
または、以下のコマンドを使用します。 # proxmox-backup-manager remote create pbs2 --host pbs2.mydomain.example --userid sync@pam --
'--password' '--password' '--pam' '--pam'
--password 'SECRET' --fingerprint 64:d3:ff:3a:50:38:53:5a:9b:f7:50:...:ab:fe
```

リモートを管理するにはproxmox-backup-manager remoteのlist, show, update, removeサブコマンドを使ってください：

```
# proxmox-backup-manager remote update pbs2 --host pbs2.example
# proxmox-backup-manager remote list
```

名前	ホスト	ユーザーID	指紋	コメント
pbs2	pbs2.example	sync@pam	64:d3:ff:3a:50:38:53:5a:9b:f7:50:...:ab:fe	

```
# proxmox-backup-manager remote remove pbs2
```

11.2 ジョブの同期

Add: Sync Job

Options Group Filter

Local Datastore: store1 Source Remote: pbsremote
 Local Namespace: Root Source Datastore: remotestore
 Local Owner: root@pam Source Namespace: Root
 Sync Schedule: Wed 02:30 Max. Depth: Full
 Rate Limit: 50 MiB/s Remove vanished:
 Comment: offsite

Help Advanced Add

同期ジョブは、リモート上のデータストアの内容をローカルのデータストアにプルするように設定されます。同期ジョブはウェブインターフェース、データストアパネルの同期ジョブタブ、またはデータストア自体から管理できます。また、以下のコマンドで管理することもできます。

コマンドで管理することもできます。同期ジョブの設定情報は `/etc/proxmox-backup/sync.cfg` に保存されています。新しい同期ジョブを作成するには、GUIで追加ボタンをクリックするか、`create`サブコマンドを使用します。同期ジョブを作成したら、手動で開始することもできます。

をGUIから選択するか、定期的に実行するスケジュール（「カレンダー・イベント」を参照）を指定します。リモート・データストアで使用できなくなったバックアップ・スナップショット、グループ、およびネームスペースは、同期ジョブの`remove-vanished`オプションを設定することで、ローカル・データストアからも削除できます。同期ジョブにを設定することで、同期ジョブをバックアップ・スナップショットに限定できます。

それぞれ検証または暗号化されています。これは特に

バックアップを信頼性の低いリモートバックアップサーバに送信する場合、これは特に重要です。

```
# proxmox-backup-manager sync-job create pbs2-local --remote pbs2 --remote-store local -->
--store local --schedule 'Wed 02:30'
proxmox-backup-manager sync-job update pbs2-local --comment 'offsite' # proxmox-backup-manager
# proxmox-backup-manager sync-job list --remote-store local --
| id | store | remote | remote-store | schedule | comment |
|----|----|----|----|----|----|
| pbs2-local | local | pbs2 | ローカル | Wed 02:30 | オフサイト |
# proxmox-backup-manager sync-job remove pbs2-local
```

同期ジョブを設定するには、設定ユーザーに以下の権限が必要です：

1. `remote/{remote}/{remote-store}`パス上の`Remote.Read`
2. ローカル・ターゲット・データストア (`/datastore/{store}`) 上の少なくとも`Datastore.Backup`

注意

同期ジョブは、構成されたリモートのユーザ/APIトークンが読み取り可能なバックアップグループのみを同期できます。リモートが`Datastore.Backup`権限のみを持つユーザ/APIトークンで構成されている場合、そのユーザ/APIトークンが所有するアクセス可能なスナップショットの限られたセットのみを同期できます。

`remove-vanished`オプションが設定されている場合は、ローカルのデータストアでも`Datastore.Prune`が必要です。`owner`オプションが設定されていない場合（デフォルトは`root@pam`）、または構成ユーザー以外に設定されている場合は、`Datastore.Modify`も必要です。

`group-filter`オプションが設定されている場合、指定された基準の少なくとも1つに一致するバックアップグループのみが同期されます。使用可能な基準は以下のとおりです：

- 例えば、`ct(コンテナ)`タイプのグループのみを同期する場合：

```
# proxmox-backup-manager sync-job update ID --group-filter type:ct
```

- 特定のバックアップグループを同期するための完全なグループ識別子：

完全なグループ識別子、特定のバックアップグループを同期する場合 # proxmox-backup-manager sync-job update ID --group-filter group:vm/100

- 完全なグループ識別子に対してマッチする正規表現

```
# proxmox-backup-manager sync-job update ID --group-filter regex:'^vm/1d{2,3}$'
```

同じフィルターがローカルグループにも適用され、remove-vanishedオプションの処理に使用されます。グループフィルターは exclude: を前につけることで反転させることができます。

- マッチを除外する正規表現の例

```
# proxmox-backup-manager sync-job update ID --group-filter exclude:regex:'^vm/1d{2,3}'  
^--3/S'
```

includeフィルタとexcludeフィルタを混在させる場合、以下のルールが適用されます：

- no filters: すべてのバックアップグループ
- include: includeフィルタに一致するもののみ
- exclude: excludeフィルタに一致するもの以外すべて
- 両方: インクルードフィルタにマッチするもの。

i 注意

リモートバックアップスナップショットの保護フラグは同期されません。

詳細オプションの「resync-corrupt」を有効にすると、前回の検証時に検証に失敗したすべてのスナップショットが再同期されます。したがって、「resync-corrupt」を使用した同期ジョブを実行する前に、検証ジョブを実行する必要があります。「resync-corrupt」ジョブはデータストア内のすべてのスナップショットのマニフェストをチェックする必要があるため、通常の同期ジョブよりもはるかに時間がかかる場合があることに注意してください。

run-on-mountフラグが設定されている場合、関連するリムーバブルデータストアがマウントされるたびに同期ジョブが自動的に開始されます。リムーバブルデータストアをマウントすると複数の同期ジョブが開始される場合ジョブを開始する場合、これらのジョブは ID に基づいてアルファベット順に実行されます。

11.2.1 名前空間のサポート

同期ジョブは、データストアを同期するだけでなく、ネームスペースまたはネームスペース・サブツリーの形式でデータストアのサブセットを同期するように構成できます。以下のパラメータは、同期ジョブの実行時にネームスペースがどのように扱われるかに影響します：

- remote-ns: リモート・ネームスペース・アンカー（デフォルト：ルート・ネームスペース）
- ns : ローカルのネームスペース・アンカー（既定値：ルート・ネームスペース）。
- max-depth: リモート・ネームスペース・アンカーのサブネームスペースを再帰的に反復するかどうか（既定値：なし）。

max-depth を 0 に設定すると、再帰処理なしで、remote-ns から ns にグループが同期されます。None(空のまま)に設定すると、再帰の深さは remote-ns の値とリモート側のネームスペース・サポートの有無に依存します：

- remote-ns にルート・ネームスペース以外を設定した場合：リモートはネームスペースに対応している必要があります、remote-ns から完全な再帰が開始します。
- remote-ns をルート名前空間に設定し、リモートが名前空間をサポートする場合：ルート名前空間から開始する完全再帰。

- **remote-ns** がルート名前空間に設定され、リモートが名前空間をサポートしていない場合：後方互換モード、ルート名前空間のみが **ns** に同期され、再帰は行われません。

例えば、**remote-ns** を *location_a/department_b* に設定し、**max-depth** を 1 に設定すると、*location_a/department_b* と、さらに最大 1 レベルのサブネームスペースが同期されます。

remote-ns で始まるネームスペース・ツリーは、**max-depth** の深さまでの **ns** にマッピングされます。例えば、リモート側に以下のネームスペースがあります：

- *location_a*
 - *location_a/department_x*
 - * *location_a/department_x/team_one*
 - * *location_a/department_x/team_two*
 - *location_a/department_y*
 - * *location_a/department_y/team_1*
 - * *location_a/department_y/team_2*
- *location_b*

に設定され、**remote-ns** が *location_a/department_x* に設定され、**ns** が *location_a_dep_x* に設定されると、同期ターゲット上のネームスペース・ツリーは以下のようになります：

- *location_a_dep_x* (リモートの*location_a/department_x*を含む)
 - *location_a_dep_x/team_one* (リモートの*location_a/department_x/team_one*を含む)
 - *location_a_dep_x/team_two* (リモートの*location_a/department_x/team_two*を含む) と、(この同期ジョブによつて) 同期されない残りのリモート・ネームスペースおよびグループ。

リモート名前空間が同期ジョブのスコープに含まれているがローカルに存在しない場合、その名前空間が作成されます (同期ジョブの所有者に十分な権限がある場合)。

remove-vanished オプションが設定されている場合、同期ジョブのスコープに含まれるがローカルにしか存在しないネームスペースは消滅したものとして扱われ、削除されます (ただし、同期ジョブの所有者に十分な権限がある場合)。

注意

同期範囲に関する他のすべての制限事項 (リモート・ユーザ/API トークン権限、グループ・フィルタなど) は、1 つまたは複数のネームスペースを含む同期ジョブにも適用されます。

11.2.2 帯域幅の制限

データストアをアーカイブに同期すると、多くのトラフィックが発生し、ネットワークの他のユーザーに影響を与える可能性があります。ネットワークやストレージの輻輳を回避するために、プル方向の同期ジョブの帯域幅を制限できます。

プル方向の同期ジョブの帯域幅を制限することができます。

proxmox-backup-manager コマンドラインツールを使用します：

```
# proxmox-backup-manager sync-job update ID --rate-in 20MiB
```

プッシュ方向の同期ジョブには、代わりに**rate-out**オプションを使用してください。トラフィックのバーストを許容するために、**burst-in**または**burst-out**パラメータでトラフィック制限に使われるトークンバケットフィルタのサイズを設定できます。

11.2.3 同期方向 プッシュ

同期ジョブはプルまたはプッシュ方向に設定できます。プッシュ方向の同期ジョブは、リモートサーバAPIを経由したターゲットデータストアへのアクセスが制限されるため、動作は同一ではありません。最も注目すべき点は、プッシュされたコンテンツは常にリモート設定で設定されたユーザーによって所有され、同期ジョブで設定されたローカルユーザーとは独立していることです。後者は、プッシュする側の権限チェックとスコープチェックにのみ使用されます。

注意

プッシュ方向の各同期ジョブに専用のリモート設定を作成し、リモート上で専用のユーザーを使用することを強くお勧めします。そうしないと、同じターゲットにプッシュする同期ジョブは、remove vanishedフラグが設定されている場合、互いのスナップショットやグループを削除したり、バックアップ時間が増分でない場合、スナップショットをスキップしたりする可能性があります。これは、ターゲット上のバックアップグループがリモート設定で指定されたユーザーによって所有されているためです。

プッシュ方向の同期ジョブには、以下のパーミッションが必要です：

1. `remote/{remote}`の`Remote.Audit`と`/remote/`の`Remote.DatastoreBackup`。
2. 同期ジョブの所有者の場合は、少なくともローカル・ソース・データストア・ネームスペース（`/datastore/{store}/{ns}`）上の`Datastore.Read`と`Datastore.Audit`、または`Datastore.Backup`。
3. `Remote/{remote}/{remote-store}/{remote-ns}`パスで`Remote.DatastorePrune`を実行し、消失したスナップショットとグループを削除します。プッシュ方向の同期ジョブには、必ず専用のリモートを使用してください。
4. リモート・データストア変更 オン `/remote/{remote}/{remote-store}/{remote-ns}` のように指定します。
パスを使用して消えたネームスペースを削除します。リモート・バックアップ・サーバ・インスタンスでは、アクセス制限付きのリモート・ユーザを使用する必要があります。
を使用する必要があります。後述の意味を考慮してください。

注釈

`Remote.DatastoreModify` を使用すると、所有権に関係なく、リモート・ターゲット・データ・ストア上のネームスペース全体を削除できます。`remote.cfg`で構成されたユーザがリモート側で限定された権限があることを確認してください。

注意

プッシュ方向の同期ジョブを実行するには、リモートのProxmox Backup Serverインスタンスで名前空間をサポートしている必要があります(最小バージョンは2.2)。

メンテナンスタスク

12.1 プルーニング

プルーニングでは、保持するバックアップ・スナップショットを指定し、他のスナップショットを削除できます。スナップショットをプルーニングする場合、スナップショットのメタデータ（マニフェスト、インデックス、プロブ、ログ、およびノート）のみが削除されます。実際のバックアップ・データを含み、プルーニングされたスナップショットによって以前に参照されたチャンクは、ガベージ・コレクションの実行によって削除される必要があります。

⚠️ 注意

所定のデータチャンクに保存されている機密情報は、少なくとも1つのバックアップスナップショットによって参照されている限り、刈り込まれたスナップショットよりも長生きし、データストアに存在し続けることを考慮してください。さらに、スナップショットが指定されたチャンクを参照していない場合でも、ガベージコレクションによって削除されるまで存在し続けます。

さらに、変更検出モードメタデータを使用して作成されたファイルレベルバックアップは、前回のバックアップ以降に消失したファイルを含むバックアップチャンクを参照できます。これらのファイル

チャンクの生データ（クライアント側またはサーバ側）を読み取ることで、これらのファイルにアクセスできる可能性があります。

機密データを含むチャンクを削除するには、そのデータがバックアップ入力の一部であった間に作成されたスナップショットを削除し、ガベージバックアップを実行します。

ガベージコレクションを実行します。さらに、変更検出モードのメタデータでファイルベースのバックアップを使用する場合、機密データがバックアップ入力の一部でなくなったため、すべてのスナップショットを削除し、ガベージコレクションを実行します。

参照されなくなったチャンクは、次回のガベージコレクション実行時に削除マークが付けられ、猶予期間後に実行される後続の実行で削除されます。

プルーニングには以下の保持オプションがあります：

keep-last <N>

最後の<N>個のバックアップスナップショットを保持します。

キープアワー<N>

直近<N>時間のバックアップを保持します。1時間に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない時間はカウントされません。

毎日<N>

過去<N>日間のバックアップを保持します。1日に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない日はカウントされません。

毎週<N>

過去<N>週間分のバックアップを保持します。1週間に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない週はカウントされません。

 注意

週は月曜日に始まり、日曜日に終わります。このソフトウェアはISOの週日付けシステムを使用しており、年末の週を正しく処理します。

毎月<N>

過去<N>ヶ月のバックアップを保持します。1つの月に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない月はカウントされません。

年毎<N>

過去<N>年間のバックアップを保持します。単一の年に複数のバックアップがある場合、最新のものだけが保持されます。バックアップのない年はカウントされません。

保持オプションは上記の順序で処理されます。各オプションは、その期間内のバックアップのみを対象とします。次のオプションでは、すでにカバーされているバックアップは処理されません。古いバックアップのみを考慮します。

古い未完成または不完全なバックアップは、最後に成功したバックアップより新しくない限り、pruneコマンドによって削除されます。この場合、最後に失敗したバックアップが保持されます。

12.1.1 プルーンのシミュレータ

内蔵のプルーンのシミュレータを使用して、さまざまなバックアップ・スケジュールでさまざまな保持オプションの効果を調べることができます。

12.1.2 プルーンのジョブ

The screenshot shows the Proxmox Backup & Restore web interface. The top navigation bar includes tabs for Summary, Content, Prune & GC (which is selected), Sync Jobs, Verify Jobs, Options, and Permissions. A help icon is also present.

Garbage Collection:

- Buttons: Edit, Start Garbage Collection.
- Text: Garbage Collection Schedule 21:00.

Prune Jobs:

- Buttons: Add, Edit, Remove, Show Log, Run now.
- Table headers: Job ID, Namespace, Max. Depth, Schedule, Keep, Last Prune, Duration, Status, Next Run, Comment.
- Table rows:
 - s-7d... dc-usa daily 3 13 8 11 1 - - - 2022-11-29 00:00:00
 - s-a1... dc-eu-central hourly 3 13 8 11 9 2022-11-28 14:00:00 <0.1s ✓ OK 2022-11-28 15:00:00

プルーンのジョブは、データストアまたはそのサブセットを定期的にプルーニングするように構成されます。プルーンの管理

ジョブはウェブインターフェイスのデータストアパネルの**Prune & GC**タブ、またはデータストア自体のタブから管理できます。あるいは、`proxmox-backup-manager prune-job`コマンドで管理することもできます。pruneジョブの設定情報は`/etc/proxmox-backup/prune.cfg`に保存されています。新しいpruneジョブを作成するには、GUIで追加ボタンをクリックするか、マネージャーCLIの`create`サブコマンドを使用します。pruneジョブを作成したら、GUIから手動で開始するか、スケジュールを指定します(カレンダーを参照)。

定期的に実行するスケジュール ([「カレンダー・イベント」](#) 参照) を指定できます。

各剪定ジョブには、保持、範囲の制限、および頻度の設定があります。

ストア <データストア

このプルーンのジョブを実行するデータストア。

ns <ネームスペース

プルーン・ジョブを特定のネームスペースに制限します。

max-depth <N

構成されたネームスペース以下からプルーンするネームスペースの深さを構成します。例えば、構成されたネームスペース自体で直接使用可能なバックアップ・グループのみをプルーンするには`0`を指定します。以下の完全な深さまでスキャンするには、パラメータを省略します。

Add: Prune Job

Datastore:	store1	Prune Schedule:	Mon..Fri 21:00
Namespace:	dc-eu-west	Enabled:	<input checked="" type="checkbox"/>
Max. Depth:	Full		
Keep Last:	3	Keep Hourly:	
Keep Daily:	13	Keep Weekly:	8
Keep Monthly:	8	Keep Yearly:	
Comment:	Work-day pruning schedule for EU West namespace		
Help		Advanced	Add

スケジュール

このジョブを自動的にトリガする[カレンダイベント間隔](#)を構成します。ジョブを手動でのみ起動する場合は省略できます。

keep-X

上記の様々な保持オプションの説明を参照してください。

disable

ジョブの設定を保持したまま、一時的にジョブを無効にする設定です。

コメント

ジョブの意図などに関する短いコメントを追加できます。

12.1.3 手動プルーニング

Prune Datastore 'test'

keep-last:	1	Backup Time ↓	keep
keep-hourly:	1	2020-11-07 15:44:54	true
keep-daily:	2	2020-11-07 15:44:50	false
keep-weekly:		2020-11-07 15:39:25	false
keep-monthly:		2020-11-07 11:00:32	true
keep-yearly:		2020-11-07 10:43:42	false
		2020-08-19 18:25:50	true
		2020-07-23 13:32:43	true
Prune			

特定のバックアップグループを手動でプルーニングするには、[バックアップのプルーニングと削除で説明するproxmox-backup-client](#)のプルーニングサブコマンドを使用するか、データストアのコンテンツタブに移動し、関連するバックアップグループのアクション列にあるハサミアイコンをクリックします。

12.1.4 保持設定の例

バックアップの頻度や古いバックアップの保持は、データの変更頻度や、特定のワークロードにおける古い状態の重要性によって異なります。バックアップが企業の文書アーカイブとして機能する場合、バックアップスナップショットの保存期間に関する法的要件がある場合もあります。

この例では、毎日バックアップを行い、保存期間を10年とし、保存されるバックアップの間隔が徐々に長くなっていくと仮定します。

- **keep-last:** 3 - 毎日バックアップを行う場合でも、管理者は大きなアップグレードの直前または直後に追加のバックアップを作成したいと思うかもしれません。keep-lastを設定することで、これが確実になります。
 - **keep-hourly:** 未設定 - 毎日のバックアップには関係ありません。手動バックアップはkeep-lastですでにカバーされています。
 - **keep-daily:** 13 - keep-last（最低1日）と合わせて、最低2週間分のバックアップを確保します。
 - **keep-weekly:** 8 - 毎週少なくとも2ヶ月分のバックアップを確保します。
 - **keep-monthly:** 11 - これまでのkeep設定と合わせて、少なくとも1年分の月別バックアップを確保します。
 - **keep-yearly:** 9 - これは長期アーカイブ用です。前のオプションで現在の年をカバーしたので、残りの年についてはこれを9に設定します。
- 少なくとも10年間をカバーすることができます。

保存期間が不必要に長いと感じたら、いつでも減らすことができますが、過去のバックアップスナップショットを再作成することはできません。

12.2 ガベージコレクション

ガベージコレクション (GC) は、チャンクストレージから未使用のバックアップチャンクをすべて削除することで、データストアのスペースを解放するプロセスです。GCはバックアップスナップショットの刈り込みを完了し、基礎となるバックアップデータではなくメタデータのみを削除します。

未使用領域が定期的にクリーンアップされるようにスケジュールを設定することをお勧めします。ほとんどのセットアップでは、毎週スケジュールを開始するのに適した間隔を提供します。

12.2.1 GCの背景

Proxmox Backup Serverでは、バックアップデータは直接保存されず、各バックアップスナップショットのインデックスによって再度参照されるチャンクとして保存されます。このアプローチにより、重複排除によるチャンクの再利用が可能になります。

バックアップスナップショットを削除する場合、Proxmox Backup Serverはそれに関連するチャンクを直接削除できません。過剰な負荷とパフォーマンスの低下を避けるため、スナップショット削除のたびに同じチャンクへの参照について他のすべてのインデックスをスキヤンするためにデータストア全体をロックすることはできません。さらに、データストア全体をロックすることは、削除処理が完了するまで新しいバックアップがロックされるため、実行不可能です。

そのため、Proxmox Backup Serverはガベージコレクション (GC) プロセスを使用して、データストア内のどのスナップショットでも不要になった未使用のバックアップチャンクを特定し、削除します。GCプロセスは、データストアのパフォーマンスや他のバックアップに影響を与えることなく、これらのチャンクが占有する領域を効率的に再生するよう

に設計されています。

ガベージコレクション (GC) プロセスはデータストアごとに実行され、2つのフェーズに分けられます：

- フェーズ1 (マーク) :

すべてのインデックスファイルが読み込まれ、参照されているチャンクファイルのアクセスタイム (atime) が更新されます。

- フェーズ2 (Sweep) :

タスクはすべてのチャンクを反復し、ファイルのアクセス時間をカットオフ時間と照合します。カットオフ時刻は、最も古いバックアップライターインスタンス (存在する場合) か、ガベージコレクション開始の24時間5分前のいずれかで与えられます。

ガベージコレクションは、カットオフ時間より古いアクセス時間を持つチャンクファイルは、バックアップスナップショットのインデックスによって参照されておらず、現在実行中のバックアップジョブの一部でもないとみなします。したがって、これらのチャンクは安全に削除できます。

猶予期間内のチャンクは削除されず、ガベージコレクションタスクの最後に削除保留として記録されます。

i 注意

バックアップ・チャンク削除の猶予期間は恣意的なものではなく、ファイルシステムは通常デフォルトで`relatime`オプションを付けてマウントされるという事実に由来しています。この結果、`atime`プロパティを更新するのは、最終アクセス以降にファイルが変更された場合か
最後のアクセスが少なくとも24時間前である場合にのみ更新されます。

12.2.2 GCの手動起動

`proxmox-backup-manager`の`garbage-collection`サブコマンドを使用して、Proxmox Backup Serverのガベージコレクションを監視および実行できます。`start`サブコマンドを使用して、データストア全体のガベージコレクションを手動で開始できます。

マンドを使用してガベージコレクションに関連する属性を確認できます。

この機能は、各データストアの [Prune & GC] タブにある *[Start Garbage Collection]* ボタンを使用して、Web UIでもアクセスできます。

12.2.3 スケジュールされた GC 2011

通常、データストアの管理者はGCを手動でトリガする手間をかけたくないかもしれません。そのため、Proxmox Backup ServerがGCを処理するようにスケジュールを設定することができます。

または データストアの GC スケジュールを編集 を使用してすることができます。
`proxmox-backup-manager` データストア更新<データストア>`--gc-schedule`
`<schedule>` CLIコマンド、または各データストアの**Prune & GC**タブのWeb UIの編集ウィンドウを使用します。

GCスケジューリングは、カレンダーイベント形式を使用します。

i ヒント

Web UIでフィールドのコンテンツをクリアするか、`proxmox-backup-manager`のデータストアの更新
`<datastore> --delete gc-schedule` CLIコマンド。これは、例えば、メンテナンス中や、データストアを永久にアーカイブする場合に便利です。
 これはメンテナンス中やデータストアを永久にアーカイブする場合などに便利です。

12.3 検証

Add: VerifyJob

Datastore: store1 Skip verified snapshots:

Schedule: daily Re-Verify After (days): 30

Comment:

Help Add

Proxmox Backup Serverには、バックアップデータが無傷であることを確認するためのさまざまな検証オプションがあります。検証は通常、検証ジョブの作成によって実行されます。これは、指定された間隔で検証を実行するスケジュールされたタスクです（カレンダーイベントを参照）。このジョブでは、すでに検証されたスナップショットを無視するかどうかを設定したり、スナップショットを再チェックする期間を設定したりすることもできます。検証ジョブを作成するためのインターフェースは、データストアの【検証ジョブ】タブにあります。

注意

前回のベリファイが成功した場合でも、すべてのバックアップを少なくとも毎月ベリファイすることをお勧めします。これは、物理ドライブは時間の経過とともに損傷を受けやすく、[ビット腐敗/データ劣化](#)と呼ばれるプロセスで、古いバックアップが破損する可能性があるためです。定期的に（毎時/毎日）検証ジョブを実行し、新しいバックアップと期限切れのバックアップをチェックし、さらに毎週/毎月ジョブを実行し、すべてを元に戻すのがよい方法です。こうすることで、データの復元時に驚くことがなくなります。

検証ジョブを使用する以外に、データストア全体、バックアップグループ、またはスナップショットに対して手動で検証を実行することもできます。これを実行するには、データストアの【Content】タブに移動し、[Verify All] をクリックするか、テーブルの【Actions】列から [V.] アイコンを選択します。

12.4 通知

Proxmox Backup Serverは、自動的にスケジュールされた検証、ガバージ収集、および同期のタスクの結果に関する通知を送信できます。

詳細については、[通知の章](#)を参照してください。

12.5 メンテナンスモード

Proxmox Backup Serverは、データストアに読み取り専用とオフラインのメンテナンスモードを設定できます。

有効化されると、モードに応じて、データストアへの新しい読み取りまたは書き込みがブロックされ、管理者は基礎となるストレージなどでメンテナントタスクを安全に実行できるようになります。

Proxmox Backup Serverは、各データストアアクセスが書き込み操作か読み取り操作かを内部的に追跡するため、メンテナンスモードを有効にする前に開始した競合する操作を終了させることで、それぞれのモードに優雅に移行することができます。

ホストシステム管理

Proxmox Backupは、有名なDebian Linuxディストリビューションをベースにしています。これは、Debianパッケージの全範囲にアクセスでき、基本システムが十分に文書化されていることを意味します。Debian Administrator's Handbookはオンラインで入手可能で、Debian オペレーティングシステムの包括的な紹介をしています。

Proxmox Backupの標準的なインストールでは、Debianのデフォルトリポジトリを使用するため、バグフィックスやセキュリティアップデートはDebianのリポジトリから入手できます。さらに、Proxmoxに関連するすべてのパッケージを展開するために、独自のパッケージリポジトリを提供しています。これには必要に応じて Debian パッケージのアップデートも含まれます。

また、Ubuntuカーネルをベースに特別に最適化されたLinuxカーネルも提供しています。このカーネルにはZFS用のドライバが含まれています。

以下のセクションでは、バックアップ関連のトピックを中心に説明します。Proxmox Backupと異なる点、またはProxmox Backupでよく使用されるタスクについて説明します。その他のトピックについては、標準のDebianドキュメントを参照してください。

13.1 Linux上のZFS

ZFSはSun Microsystemsによって設計された、ファイルシステムと論理ボリュームマネージャを組み合わせたものです。ZFSモジュールを手動でコンパイルする必要はありません。

ZFSを使用することで、低予算のハードウェアで最大限のエンタープライズ機能を実現することができ、SSDキャッシュやSSDのみのセットアップを活用することで、高性能システムも実現できます。ZFSを使用することで、高価なハードウェアのレイドカードの代わりに、CPUとメモリの負荷を抑え、簡単に管理することができます。

ZFSの一般的な利点

- GUIとCLIによる簡単な設定と管理。
- 信頼性
- データ破損からの保護
- ファイルシステムレベルでのデータ圧縮
- スナップショット
- コピーオンライトクローン
- 様々なレイドレベルRAID0、RAID1、RAID10、RAIDZ-1、RAIDZ-2、RAIDZ-3
- キャッシュにSSDを使用可能

- セルフヒーリング
- 繙続的な整合性チェック
- 大容量ストレージに対応
- ネットワーク経由の非同期レプリケーション

- オープンソース
- 暗号化

13.1.1 ハードウェア

ZFSはメモリに大きく依存するため、最低8GBのメモリを用意することをお勧めします。実際には、ハードウェアや予算に見合うだけのメモリを使用してください。データの破損を防ぐため、高品質のECC RAMの使用をお勧めします。

専用のキャッシングおよび/またはログディスクを使用する場合は、エンタープライズクラスのSSD（例えば、Intel SSD DC S3700シリーズ）を使用する必要があります。これにより、全体的なパフォーマンスが大幅に向上します。

重要: 独自のキャッシング管理を持つハードウェアコントローラの上でZFSを使用しないでください。ZFSはディスクと直接通信する必要があります。HBAアダプタまたはLSIコントローラのようなものを推奨します。のようなものを推奨します。

13.1.2 ZFS の管理

このセクションでは、一般的なタスクの使用例を紹介します。ZFS 自体は本当に強力で、多くのオプションを提供します。ZFSを管理する主なコマンドは`zfs`と`zpool`です。両コマンドには豊富なマニュアルページがあり、次のようにして読むことができます：

```
# man zpool
# man zfs
```

新しいzpoolの作成

新しいプールを作成するには、少なくとも1つのディスクが必要です。`ashift`は、基礎となるディスクと同じセクタサイズ(`ashift`の2乗)以上である必要があります。

```
# zpool create -f -o ashift= 12 <プール> <デバイス
```

RAID-0で新しいプールを作成します。

最小1ディスク

```
# zpool create -f -o ashift= 12 <プール> <デバイス1> <デバイス2
```

RAID-1で新しいプールを作成

最小2ディスク

```
# zpool create -f -o ashift= 12 <プール> ミラー <デバイス1> <デバイス2
```

RAID-10で新しいプールを作成します。

最小4ディスク

```
# zpool create -f -o ashift= 12 <pool> mirror <device1> <device2> mirror <device3> <device4
```

RAIDZ-1で新しいプールを作成

最小3ディスク

```
# zpool create -f -o ashift= 12 <pool> raidz1 <device1> <device2> <device3
```


RAIDZ-2で新しいプールを作成します。

最小4ディスク

```
# zpool create -f -o ashift=12 <pool> raidz2 <device1> <device2> <device3> <device4>
```

キャッシュ (L2ARC) 付きプールを新規作成

専用のキャッシュドライブ・パーティションを使用して読み取り性能を向上させることができます (SSDを使用)。

<device>については、「RAID* で新しいプールを作成する」に示すように、複数のデバイスを使用できます。

```
# zpool create -f -o ashift=12 <pool> <device> cache <cache_device>.
```

ログ付きプールを新規作成 (ZIL)

書き込み性能を上げるために専用のキャッシュドライブ・パーティションを使用することも可能です (SSDを使用)。

<device>については、"RAID* で新しいプールを作成する" に示されているように、複数のデバイスを使用することができます。

```
# zpool create -f -o ashift=12 <pool> <device> log <log_device>.
```

既存のプールにキャッシュとログを追加

プールの作成後に、キャッシュ・デバイスとログ・デバイスをプールに追加できます。この例では、単一のドライブをキャッシュとログの両方に使用します。まず、*parted*または*gdisk*を使用してSSD上に2つのパーティションを作成する必要があります。

重要

常にGPTパーティションテーブルを使用してください。

ログデバイスの最大サイズは物理メモリの半分程度にすべきなので、通常はかなり小さくなります。SSDの残りはキャッシュとして使用できます。

```
# zpool add -f <pool> log <device-part1> cache <device-part2>.
```

失敗したデバイスの変更

```
zpool replace -f <pool> <旧デバイス> <新デバイス> # zpool replace -f <pool> <旧デバイス> <新デバイス>
```

失敗したブータブルデバイスの変更

Proxmox Backupのインストール方法によって、ブートローダとして*grub*または*systemd-boot*を使用しています。

どちらの場合でも、パーティションテーブルのコピー、GUIDの再発行、ZFSパーティションの交換という最初のステップは同じです。新しいディスクからシステムを起動可能にするには、使用するブートローダによって異なる手順が必要です。

```
# sgdisk <健康なブート可能デバイス> -R <新しいデバイス>
# sgdisk -G <新しいデバイス>
# zpool replace -f <pool> <古い zfs パーティション> <新しい zfs パーティション>
```

注

zpool status -v コマンドを使用して、新しいディスクの resilvering プロセスがどの程度進んでいるかを監視します。

systemd-boot を使用します：

```
# proxmox-boot-tool format <新しいESP> # proxmox-boot-tool init <新しいESP>
# proxmox-boot-tool init <新しいESP>
```

注意

ESPはEFIシステムパーティションの略で、Proxmox Backupインストーラによってセットアップされたブータブルディスクのパーティション#2としてセットアップされます。詳細については、[同期ESPとして使用する新しいパーティションの設定](#)を参照してください。

*grub*の場合：

通常 *grub.cfg* は */boot/grub/grub.cfg* にあります。

```
# grub-install <新しいディスク>
# grub-mkconfig -o /path/to/grub.cfg
```

電子メール通知の有効化

ZFSにはZEDというイベントデーモンが付属しており、ZFSカーネルモジュールによって生成されたイベントを監視します。このデーモンは、プールエラーなどのZFSイベント発生時に電子メールを送信することもできます。新しいZFSパッケージでは、デーモンは別のパッケージ*zfs-zed*で出荷されており、Proxmox Backupにはデフォルトでインストールされているはずです。

お好みのエディタを使用して、*/etc/zfs/zed.d/zed.rc* ファイルでデーモンを設定できます。電子メール通知に必要な設定は *ZED_EMAIL_ADDR* で、デフォルトでは *root* に設定されています。

```
ZED_EMAIL_ADDR="ルート"
```

Proxmox Backupは *root*宛のメールを *root* ユーザに設定されたメールアドレスに転送することに注意してください。

ZFSメモリ使用量の制限

ホストのパフォーマンス低下を防ぐため、ZFS ARCにはシステムメモリの最大50%（デフォルト）を使用するのが良いでしょう。お好みのエディタを使用して、*/etc/modprobe.d/zfs.conf* の設定を変更して挿入します：

```
オプション zfs zfs_arc_max=8589934592
```

上記の例では、使用量を8GiB ('8 * 2^30') に制限しています。

重要

希望する *zfs_arc_max* 値が *zfs_arc_min*（デフォルトはシステム・メモリの 1/32）以下の場合、*zfs_arc_max* は無視されます。したがって、このケースで動作させるには、*zfs_arc_min* を最大でも *zfs_arc_max - 1* に設定する必要があります。このためには、*/etc/modprobe.d/zfs.conf* で設定を更新する必要があります：

```
options zfs zfs_arc_min=8589934591 options zfs
zfs_arc_max=8589934592
```

この設定例では、単純に *zfs_arc_max* を設定するだけでは機能しない、総メモリ量が 256 GiB を超えるシステムでの使用量を 8 GiB ('8 * 2^30') に制限しています。

重要

ルート・ファイル・システムがZFSの場合、この値が変更されるたびに *initramfs* を更新する必要があります。

```
# update-initramfs -u
```

ZFSでのスワップ

zvol上に作成されたスワップスペースは、サーバーをブロックしたり、高いIO負荷を発生させるなどの問題を引き起こす可能性があります。

メモリ不足に陥らないよう、十分なメモリを使用することを強くお勧めします。スワップを追加したい場合は、物理ディスク上にパーティションを作成し、スワップデバイスとして使用することをお勧めします。インストーラの詳細オプションで、この目的のために空き領域を残すことができます。さらに、*swappiness* の値を下げるこどもできます。サーバに適した値は 10 です：

```
# sysctl -w vm.swappiness=10
```

*swappiness*を永続的にするには、*/etc/sysctl.conf*をお好みのエディタで開き、以下の行を追加します：

```
vm.swappiness=10
```

表1：Linuxカーネルの*swappiness*パラメータ値

値	戦略
vm.swappiness=0	カーネルは「メモリ不足」状態を回避するためにのみスワップを行います。
vm.swappiness=1	スワップを完全に無効にしない最小限のスワップ量
vm.swappiness=10	十分な量のスワップが必要な場合、パフォーマンスを向上させるために推奨されることがあります。
	メモリはシステム内に存在します。
vm.swappiness=60	デフォルト値。
vm.swappiness=100	カーネルが積極的にスワップします。

ZFS圧縮

圧縮を有効にするには

```
zpool set compression=lz4 <pool> # zpool set compression lz4 <pool>
```

lz4 アルゴリズムを使用することをお勧めします。*lzb, zstd, gzip-N*(Nは圧縮率を表す1-9の整数で、1が最も速く、9が最も良い圧縮率です)のような他のアルゴリズムも利用可能です。アルゴリズムとデータの圧縮率によっては、圧縮を有効にすることでI/Oパフォーマンスが向上することもあります。

圧縮は次のようにしていつでも無効にできます：

```
# zfs set compression=off <データセット
```

この変更によって影響を受けるのは新しいブロックだけです。

ZFSの特殊デバイス

バージョン0.8.0以降、ZFSは特殊デバイスをサポートしています。プール内の特別なデバイスは、メタデータ、重複排除テーブル、およびオプションで小さなファイルブロックを格納するために使用されます。

特殊デバイスは、メタデータの変更が多い低速回転ハードディスクで構成されるプールの速度を向上させることができます。たとえば、大量のファイルを作成、更新、または削除するようなワークロードでは、特殊デバイスの存在が役立ちます。ZFSデータセットは、特別なデバイスに小さなファイルを保存するように構成することもでき、パフォーマンスをさらに向上させることができます。特殊デバイスには高速SSDを使用してください。

■ 重要

特殊デバイスはプール全体の障害点となるため、特殊デバイスの冗長性はプールの冗長性と一致させる必要があります。

■ 警告

プールに特殊デバイスを追加すると、元に戻すことはできません！

特殊デバイスとRAID-1を使用してプールを作成するには、以下の手順に従います：

```
# zpool create -f -o ashift=12 <プール> ミラー <デバイス1> <デバイス2> 特別ミラー <デバイス3>
→<デバイス4>
```

RAID-1で既存のプールに特別なデバイスを追加します：

```
<device1> <device2> # zpool add <pool> special mirror <device1> <device2>
```

ZFSデータセットは、*special_small_blocks=<size>*プロパティを公開します。sizeは、特別なデバイスに小さなファイルブロックを保存しないようにする0、または512Bから128Kの範囲の2のべき乗にすることができます。このプロパティを設定すると、sizeより小さい新しいファイル・ブロックが特殊デバイス上に割り当てられます。

■ 重要

*special_small_blocks*の値がデータセットのレコードサイズ（デフォルト128K）以上である場合、すべてのデータが特別なデバイスに書き込まれるため、注意が必要です！

プールで*special_small_blocks*プロパティを設定すると、すべての子ZFSデータセットのプロパティのデフォルト値が変更されます（たとえば、プール内のすべてのコンテナは、小さなファイルブロックをオプトインします）。

プール全体で4Kブロックより小さいファイルをすべてオプトインします：

```
# zfs set special_small_blocks=4K <プール>。
```

単一のデータセットに対して小さなファイルブロックを選択します：

```
# zfs set special_small_blocks=4K <プール>/<ファイルシステム>
```

単一のデータセットに対して小さなファイルブロックからオプトアウトします：

```
# zfs set special_small_blocks=0 <pool>/<filesystem> 以下のようにします。
```

トラブルシューティング

壊れたキャッシュファイル

ル

*zfs-import-cache.service*はZFSキャッシュファイルを使用してZFSプールをインポートします。このファイルが破損すると、サービスはそこから読み取ることができないプールをインポートできなくなります。

その結果、ZFSキャッシュ・ファイルが破損している場合、一部のボリュームはブート中にマウントされず、後で手動でマウントする必要があります。

各プールについて、以下を実行します：

```
# zpool set cachefile=/etc/zfs/zpool.cache POOLNAME
```

次に、*initramfs*を更新します：


```
# update-initramfs -u -k all
```

を実行してinitramfsを更新し、最後にノードを再起動します。

この問題のもう1つの回避策は、デバイススキャン経由でプールを検索してインポートする*zfs-import-scan.service*を有効にすることです（通常は遅い）。

13.2 ホストブートローダ

Proxmox Backupは現在、インストーラで選択されたディスクセットアップに応じて、2つのブートローダのいずれかを使用します。

ルートファイルシステムとしてZFSを使用してインストールされたEFIシステムでは、セキュアブートが有効になっていない限り、*systemd-boot*が使用されます。それ以外の展開では、標準のgrubブートローダを使用します（これは通常、Debian上にインストールするシステムにも当てはまります）。

13.2.1 インストーラが使用するパーティションスキーム

Proxmox Backupインストーラは、インストール用に選択されたすべてのディスクに3つのパーティションを作成します。

- 作成されるパーティションは以下の通りです：
- 1 MB BIOSブートパーティション(gdiskタイプEF02)
 - 512 MBのEFIシステムパーティション(ESP、gdiskタイプEF00)
 - 設定されたhdsizeパラメータ、または選択されたストレージタイプで利用可能な残りの領域にまたがる第3パーティション

ルートファイルシステムとしてZFSを使用するシステムは、512 MBのEFIシステムパーティションに保存されたカーネルとinitrdイメージで起動します。レガシ BIOSシステム、およびセキュアブートが有効なEFIシステムの場合、セキュアブートなしのEFIシステムでは*systemd-boot*が使用されます。どちらもインストールされESPを指すように設定します。

BIOSモード(--target i386-pc)のgrubは、grubで起動した全てのシステム(つまり、ext4やxfsでrootを使った全てのインストール、EFI以外のシステムでZFSでrootを使ったインストール)の選択したディスクのBIOSブートパーティションにインストールされます。

13.2.2 proxmox-boot-toolによるESPの内容の同期

*proxmox-boot-tool*は、EFIシステムパーティションの内容を適切に設定し、同期させるために使用されるユーティリティです。特定のカーネルバージョンを全てのESPにコピーし、*vfat*でフォーマットされたESPからブートするようブートローダを設定します。ルートファイルシステムとしてのZFSの文脈では、これは、grubのZFS実装にも存在するサブセットや、小さな別のブートプールを作成する代わりに、ルートプールですべてのオプション機能を使用できることを意味します(参照: grubによるルートでのZFSの起動)。

冗長性のあるセットアップでは、全てのディスクはインストーラによってESPでパーティショニングされます。これにより、最初のブートデバイスが故障したり、BIOSが特定のディスクからしか起動できない場合でも、システムが確実に起動します。

通常の操作では、ESPはマウントされたままにはなりません。これにより、システムクラッシュ時に*vfat*フォーマットされたESPのファイルシステムが破壊されるのを防ぐことができます。

/を手動で変更する必要がなくなります。

*proxmox-boot-tool*は以下のタスクを処理します：

- 新しいパーティションのフォーマットと設定
- 新しいカーネルイメージとinitrdイメージをリストされた全てのESPにコピーし設定します。

- カーネル・アップグレードやその他の保守作業時の設定の同期
- 同期されるカーネルバージョンのリストの管理

- 特定のカーネルバージョンをブートするようにブートローダを設定する (pinning) 実行す

ることで、現在設定されている ESP とその状態を見ることができます：

```
# proxmox-boot-tool status
```

同期 ESP として使う新しいパーティションの設定

パーティションを同期 ESP としてフォーマットして初期化するには、例えば rpool 内の故障した vdev を置き換えた後に proxmox-kernel-helper の proxmox-boot-tool を使います。



format コマンドは <partition> をフォーマットします。 正しい device/partition を渡してください！

例えば、空のパーティション /dev/sda2 を ESP としてフォーマットするには、以下を実行します：

```
# proxmox-boot-tool format /dev/sda2
```

Proxmox Backup のカーネル更新同期メカニズムに含めるために、/dev/sda2 にある既存のマウントされていない ESP をセットアップするには、以下を使用します：

```
# proxmox-boot-tool init /dev/sda2
```

または

```
# proxmox-boot-tool init /dev/sda2 grub
```

で、systemd-boot の代わりに Grub を使って強制的に初期化します。

この後、/etc/kernel/proxmox-boot-uuids' に新しく追加されたパーティションのUUIDの行が追加されるはずです。initコマンドは、設定されているすべてのESPの更新も自動的にトリガします。

すべてのESPの設定の更新

全てのブータブルカーネルをコピーして設定し、/etc/kernel/proxmox-boot-uuids にリストされている全ての ESP を同期させておくには、以下のコマンドを実行するだけです：

```
# proxmox-boot-tool refresh
```

(rootでext4やxfsのシステムでupdate-grubを実行するのと同じです)。

これは、カーネルコマンドラインに変更を加えた後や、すべてのカーネルとinitrdsを同期させたい場合に必要です。



注意

update-initramfs と apt (必要な場合) の両方が自動的にリフレッシュを起動します。

proxmox-boot-tool が考慮するカーネルバージョン

以下のカーネルバージョンがデフォルトで設定されています：

- 現在動作中のカーネル
- パッケージアップデートで新しくインストールされるバージョン
- 既にインストールされている最新のカーネル
- 該当する場合は、最後から2番目のカーネルシリーズの最新バージョン (5.0, 5.3など)

- 手動で選択したカーネル

手動でカーネルを起動可能に維持

特定のカーネルと initrd イメージを起動可能なカーネルのリストに追加したい場合は、次のようにします

proxmox-boot-tool kernel add を使ってください。

例えば、ABIバージョン5.0.15-1-pveのカーネルを全てのESPにインストールして同期しておくカーネルリストに追加するには、以下を実行してください：

```
# proxmox-boot-tool kernel add 5.0.15-1-pve
```

proxmox-boot-tool kernel list は現在起動用に選択されている全てのカーネルバージョンを一覧表示します：

```
# proxmox-boot-tool kernel list 手動で選択  
したカーネル : 5.0.15-1-pve  
自動的に選択されたカーネル 5.0.12-1-pve  
4.15.18-18-pve
```

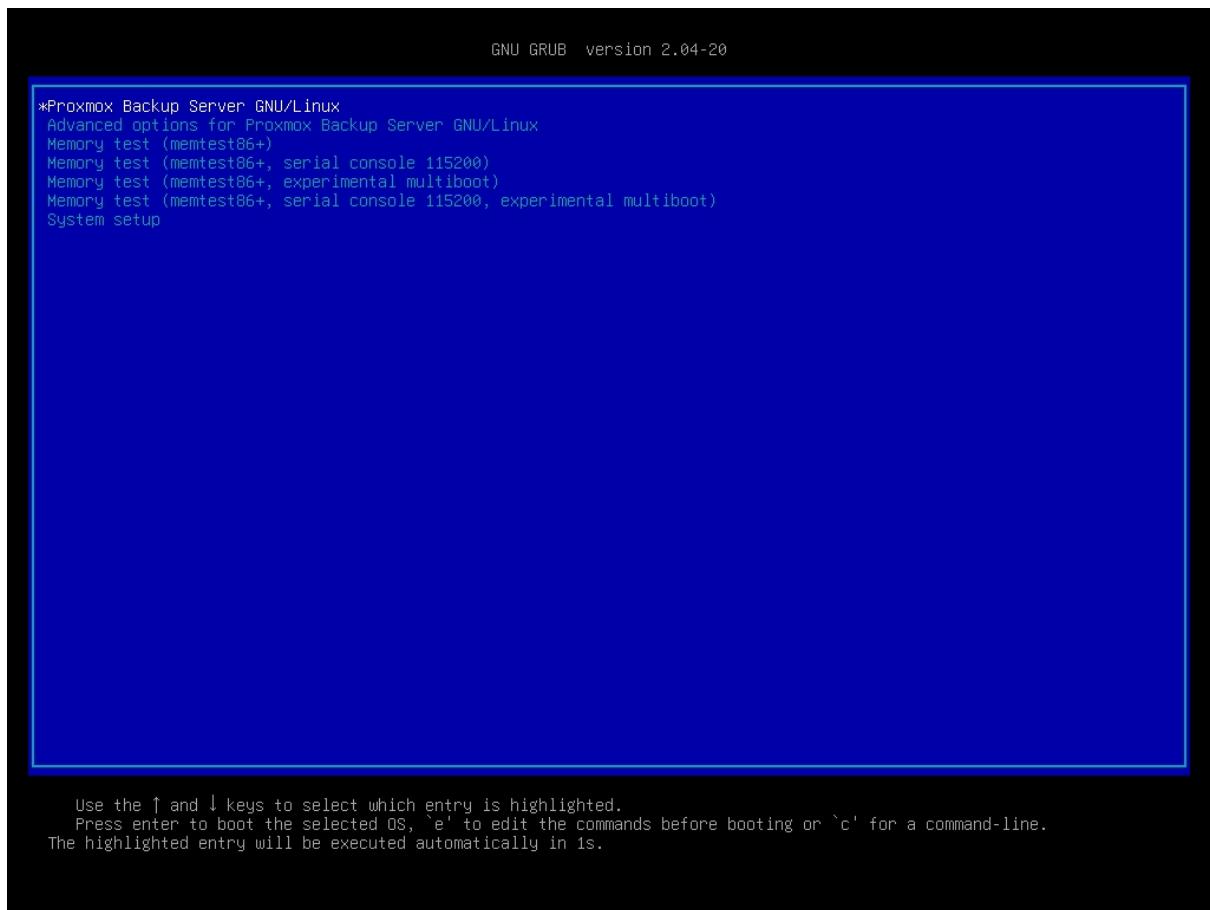
proxmox-boot-tool kernel remove を実行して手動で選択したカーネルのリストからカーネルを削除してください：

```
例: # proxmox-boot-tool kernel remove 5.0.15-1-pve
```

注意

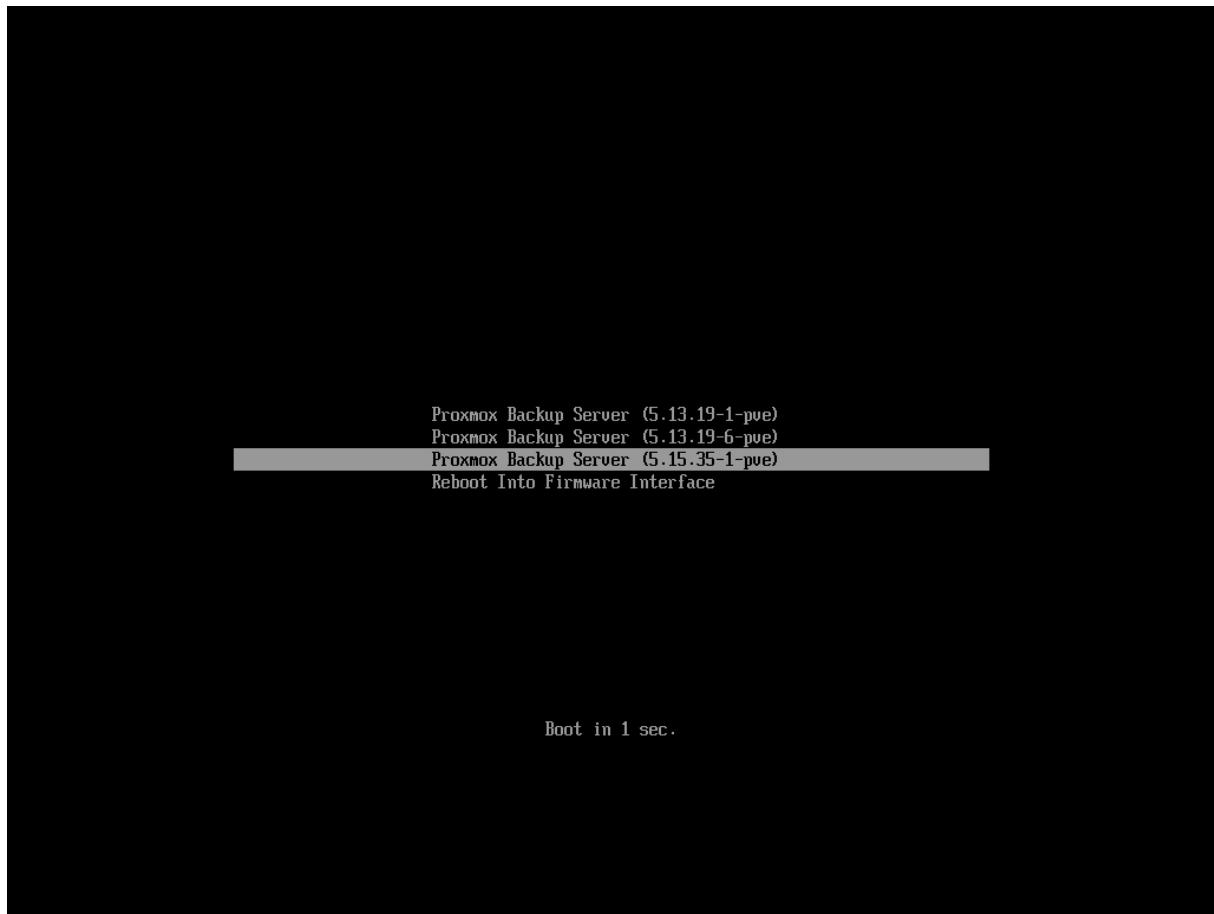
手動でカーネルを追加・削除した後に全ての EFI システムパーティション (ESP) を更新するには **proxmox-boot-tool refresh** を実行する必要があります。

13.2.3 どのブートローダが使われているかの判断



どのブートローダが使用されているかを判断する最も簡単で信頼できる方法は、Proxmox Backupノードのブートプロセスを見ることです。

grubの青いボックスか、白地に黒のシンプルなsystemd-bootが表示されます。



起動中のシステムからブートローダを判断するのは100%正確ではないかもしれません。最も確実な方法は、以下のコマンドを実行することです：

```
# efibootmgr -v
```

`efibootmgr -v` # `efibootmgr -v` EFI変数がサポートされていないというメッセージが返ってきたら、`grub`はBIOS/Legacyモードで使用されています。出力に以下のような行があれば、`grub`はUEFIモードで使用されています。

```
Boot0005* proxmox [...] File(\EFI\proxmox\grubx64.efi)
```

出力に以下のような行が含まれる場合、`systemd-boot`が使用されます。

```
Boot0006* Linux Boot Manager [...] File(\EFI\systemd\bootx64.efi)
```

以下のコマンドを実行すると、`proxmox-boot-tool`が設定されているかどうか調べることができます：

```
# proxmox-boot-tool status
```

13.2.4 grub

`grub`は長年Linuxシステムを起動するためのデファクトスタンダードであり、非常によく文書化されています（[Grubマニュアルを参照](#)）。

設定

`grub`の設定変更は、`defaults` ファイル `/etc/default/grub` または `/etc/default/grub.d` の設定スニペットで行います：

```
# update-grub
```

注意

システム	使用している	プロックスモックスブートツール	はを呼
び出します。	を呼び出します。	を呼び出します。	
を呼び出します。			

13.2.5 systemd-boot

systemd-boot は軽量な EFI ブートローダです。インストールされている EFI サービスパーティション (ESP) からカーネルと initrd イメージを直接読み込みます。ESP からカーネルを直接読み込む主な利点はカーネルをESPから直接読み込む主な利点は、ストレージにアクセスするためのドライバを再実装する必要がないことです。Proxmox Backupでは、[proxmox-boot-tool](#)を使用してESP上の設定を同期させます。

設定

systemd-boot は、EFI System Partition (ESP) のルートディレクトリにある `loader/loader.conf` ファイルで設定します。詳細は `loader.conf(5)` man ページを参照してください。

各ブートローダエントリは、`loader/entries/` ディレクトリ内のそれ自身のファイルに置かれます。

`entry.conf` の例は次のようにになります (/ は ESP のルートを指します)：

```
タイトル      タイトル
メニュー      5.0.15-1-pve
オプション    root=ZFS=rpool/ROOT/pve-1 boot=zfs
Linux        /EFI/proxmox/5.0.15-1-pve/vmlinuz-5.0.15-1-pve initrd
              /EFI/proxmox/5.0.15-1-pve/initrd.img-5.0.15-1-pve
```

13.2.6 カーネルコマンドラインの編集

カーネルコマンドラインは、使用するブートローダに応じて以下の場所で変更できます：

Grub

カーネルコマンドラインは `/etc/default/grub` ファイル内の変数 `GRUB_CMDLINE_LINUX_DEFAULT` に置く必要があります。`update-grub`を実行すると、その内容が`/boot/grub/grub`内のすべての`linux`エントリに追加されます。`/`に追加されます。

systemd-boot

カーネルコマンドラインは `/etc/kernel/cmdline` に1行で配置する必要があります。変更を適用するには、`proxmox-boot-tool refresh`を実行して、`loader/entries/proxmox*.conf` の全ての設定ファイルのオプション行として設定します。

13.2.7 次のブートのためのカーネルバージョンの上書き

現在のデフォルトカーネルではないカーネルを選択するには、以下の方法があります：

- ブートプロセスの最初に表示されるブートローダーメニューを使います。
- `proxmox-boot-tool` を使ってシステムをカーネルバージョンに固定します。

これにより、新しいカーネルバージョンとハードウェアの非互換性を回避することができます。

i 注意

このような pin はできるだけ早く削除して、最新のカーネルからのセキュリティパッチもすべてシステムに適用されるようにしてください。

例えば、起動用にバージョン 5.15.30-1-pve を恒久的に選択するには、次のように実行します：

```
# proxmox-boot-tool kernel pin 5.15.30-1-pve
```

💡 ヒント

ピン機能は `Proxmox-boot-tool` を使って ESP の内容を同期しているシステムだけでなく、すべての Proxmox Backup システムで動作します。
の呼び出しをスキップすることもできます。

また、カーネルバージョンを次のシステム起動時のみ起動するように設定することもできます。これは、例えば、更新されたカーネルが問題を解決したかどうかをテストするのに便利です：

```
# proxmox-boot-tool kernel pin 5.15.30-1-pve --next-boot
```

pin されたバージョン設定を削除するには、`unpin` サブコマンドを使います：

```
# proxmox-boot-tool kernel unpin
```

`unpin` には `--next-boot` オプションもありますが、これは
`--next-boot`。これは起動時に自動的に行われるため、手動で実行してもほとんど意味がありません。

固定バージョンを設定またはクリアした後は、`refresh`サブコマンドを実行してESPのコンテンツと設定を同期させる必要があります。

💡 ヒント

`proxmox-boot-tool`を管理するシステムに対して、対話的にツールを呼び出すと、自動的に実行するようにプロンプトが表示されます。

```
# proxmox-boot-tool refresh
```

13.2.8 セキュアブート

Proxmox Backup 3.1以降、セキュアブートは署名されたパッケージと`proxmox-boot-tool`の統合によってすぐにサポートされます。

セキュアブートを有効にするには以下のパッケージをインストールする必要があります：

- `shim-signed` (Microsoft によって署名された shim ブートローダ)
- `shim-helpers-amd64-signed` (Proxmox によって署名されたフォールバックブートローダと MOKManager)
- `grub-efi-amd64-signed` (Proxmox により署名された Grub EFI ブートローダ)
- `proxmox-kernel-6.X.Y-Z-pve-signed` (Proxmox により署名されたカーネルイメージ)

他の署名済みブートローダパッケージがないため、ブートローダとして Grub だけがサポートされています。Proxmox Backupを新規インストールすると、上記のパッケージがすべて自動的に組み込まれます。

セキュアブートの仕組みとセットアップのカスタマイズ方法の詳細については、[wikiを参照してください。](#)

既存のインストールをセキュアブートに切り替える

⚠️ 警告

これは、正しく行われないと、場合によっては起動不能なインストールにつながる可能性があります。ホストを再インストールすると、セキュアブートが利用可能な場合は、余分な操作をすることなく、自動的にセキュアブートがセットアップされます。**Proxmox Backup**ホストの動作し、十分にテストされたバックアップがあることを確認してください！

Proxmox Backupを最初から再インストールしなくても、必要に応じて既存のUEFIインストールをセキュアブートに切り替えることができます。

まず、すべてのシステムが最新であることを確認します。次に、上記のように必要な署名済みパッケージをすべてインストールします。Grubは、デフォルトのshim経由で起動するために必要なEFIブートエントリを作成します。

systemd-boot

systemd-bootをブートローダとして使う場合([どのブートローダを使うかを決定する](#)を参照)、いくつかの追加設定が必要です。これは、Proxmox BackupがZFS-on-rootでインストールされている場合のみです。

後者を確認するには、以下を実行します：

```
# findmnt /
```

ホストが本当にルート・ファイルシステムとしてZFSを使用している場合、FSTYPE列にzfsが含まれるはずです：

TARGET	SOURCE	FSTYPEオプション
/	rpool/ROOT/pbs-1 zfs	rw,relatime,xattr,noacl.

次に、適切なESP（EFIシステムパーティション）を見つけなければなりません。これは

lsblkコマンドを使います：

```
# lsblk -o +FSTYPE
```

出力は次のようにになります：

名前	マージ:ミン rm	サイズ	ro	タイプ	マウントポイント	fstype
sda	8:0	0	32G	0	ディスク	
└─sda1	8:1	0	1007K	0	部分	
└─sda2	8:2	0	512M	0	部分	vfat zfs_member
sdb	8:3	0	31.5G	0	パート	
└─sdb1	8:16	0	32G	0	ディス	
└─sdb2	8:17	0	1007K	ク		
└─sdb3	8:18	0	512M	0	部分	vfat zfs_member
	8:19	0	31.5G	0	部分	
				0	パート	

この場合、パーティションsda2とsdb2がターゲットです。パーティションのサイズは512Mで、FSTYPEはvfatです。

これらのパーティションは**proxmox-boot-tool**を使ってGrubで起動するように適切に設定する必要があります。このコマンド(例としてsda2を使用)は、個々のESPに対して個別に実行する必要があります：

```
# proxmox-boot-tool init /dev/sda2 grub
```

その後、以下のコマンドを実行して、セットアップの正常性を確認できます：

```
# efibootmgr -v
```

このリストには、以下のようないエントリーが含まれているはずです：

[--]	Boot0009* proxmox [...]	HD(2,GPT,...,0x800,0x100000)/File(\EFI\proxmox\shimx64.efi)
[...]		

1 注意

古いsystemd-bootブートローダは維持されますが、Grubが優先されます。こうすることで、セキュアブートモードでGrubを使って起動しても何らかの理由でうまくいかない場合でも、セキュアブートをオフにしたsystemd-bootを使ってシステムを起動することができます。

これでホストを再起動し、UEFI フームウェアセットアップユーティリティで Secure Boot を有効にすることができます。

再起動すると UEFI フームウェアのブートメニューに proxmox という新しいエントリが選択できるようになっているはずで、署名済みの EFI shim を使って起動します。

何らかの理由で UEFI ブートメニューに proxmox エントリが見つからない場合、(フームウェアがサポートしていれば) カスタムブートエントリとして | proxmox | shimx64.efi を追加して手動で追加してみることができます。

i Note

いくつかの UEFI フームウェアは再起動時に proxmox ブートオプションを落とすことが知られています。これは proxmox ブートエントリがディスク上の Grub インストールを指していて、ディスク自体がブートオプションにならない場合に起こります。可能であれば、UEFI フームウェアセットアップユーティリティでディスクをブートオプションとして追加し

でディスクをブートオプションとして追加し、**proxmox-boot-tool** をもう一度実行してみてください。

💡 ヒント

カスタムキーを登録するには、付属の [Secure Boot wiki ページ](#)を参照してください。

セキュアブートでの DKMS/サードパーティモジュールの使用

セキュアブートが有効になっているシステムでは、カーネルは信頼できる鍵で署名されていないモジュールのロードを拒否します。カーネルパッケージに同梱されているデフォルトのモジュールセットは、カーネルイメージに埋め込まれたエフェメラルキーで署名されており、その特定のバージョンのカーネルイメージでは信頼されています。

DKMS を使ってビルドされたモジュールや手動でビルドされたモジュールなど、他のモジュールをロードするには、以下のように署名する必要があります。

をセキュアブートスタックに信頼された鍵で作成する必要があります。これを実現する最も簡単な方法は、**mokutil** を使用してマシンオーナーキー (MOK) として登録することです。

dkms ツールが自動的にキーペアと証明書を **/var/lib/dkms/mok.key**

および**/var/lib/dkms/mok.pub**にキーペアと証明書を自動的に生成し、ビルドおよびインストールするカーネルモジュールの署名に使用します。

証明書の内容は次のようにして表示できます。

```
# openssl x509 -in /var/lib/dkms/mok.pub -noout -text
```

を実行し、以下のコマンドを使用してシステムに登録します：

```
# mokutil --import /var/lib/dkms/mok.pub input password :  
もう一度パスワードを入力してください :
```

mokutilコマンドは(一時的な)パスワードを2回要求します！システムを再起動すると、自動的にブートすると MOKManager EFI バイナリが起動し、**mokutil** を使ってキー/証明書を検証し、登録開始時に選択したパスワードを使用して登録を確認できます。その後、カーネルは（登録されたMOKで署名された）DKMSでビルドされたモジュールのロードを許可するはずです。必要であれば、MOKはカスタムEFIバイナリやカーネルイメージの署名にも使用できます。

DKMSで管理されていないカスタム/サードパーティモジュールにも同じ手順を使用できますが、その場合は鍵/証明書の生成と署名の手順を手動で行う必要があります。

13.3 証明書の管理

APIおよびWebベースの管理インターフェイスへのアクセスは常にhttpsで暗号化されます。各Proxmox Backupホストはデフォルトで独自の(自己署名)証明書を作成します。この証明書はホストのproxmox-backup-proxyサービスとの暗号化通信に使用されます。

ユーザーまたはバックアップクライアントとウェブインターフェイス間のAPI呼び出しに使用されます。

Proxmox Backup Serverへバックアップを送信する際の証明書の検証は、ストレージ/リモート設定に証明書のフィンガープリントをピン留めするか、信頼できる認証局によって署名された証明書を使用して行われます。

13.3.1 APIおよびSMTP用の証明書

Proxmox Backupは証明書と鍵を次の場所に保存します：

- /etc/proxmox-backup/proxy.pem
- /etc/proxmox-backup/proxy.key

証明書には以下のオプションがあります：

1. etc/proxmox-backup/proxy.pem にあるデフォルトの自己署名証明書を使用し続けます。
2. 外部から提供された証明書(例えば、商用のCA(Certificate Authority)によって署名されたもの)を使用します。
3. Let's EncryptのようなACMEプロバイダを使用して、自動更新付きの信頼できる証明書を取得します。これは Proxmox Backup APIとWebインターフェースにも統合されています。

証明書は Proxmox Backup ウェブインターフェース/API または proxmox-backup-managerを使用します。

13.3.2 カスタム証明書のアップロード

Proxmox Backup ホストで使用したい証明書がすでにある場合、Web インターフェースから証明書をアップロードできます。

The screenshot shows a modal dialog titled "Upload Custom Certificate". It contains two input fields: "Private Key (Optional)" with the value "No change" and a "From File" button. Below that is a "Certificate Chain:" field containing a large amount of certificate text starting with "-----BEGIN CERTIFICATE-----". There is another "From File" button next to it and a large "Upload" button at the bottom right.

証明書キーファイルはパスワードで保護されてはならないことに注意してください。

13.3.3 Let's Encrypt (ACME)経由の信頼できる証明書

Proxmox Backupには自動証明書管理環境(**ACME**)プロトコルの実装が含まれており、Proxmox Backupの管理者はLet's EncryptのようなACMEプロバイダを使用してTLS証明書を簡単にセットアップすることができます。

現在、実装されているACMEエンドポイントは、Let's Encrypt (LE) の本番環境とステージング環境の2つです。私たちのACMEクライアントは、組み込みのウェブサーバを使用したhttp-01チャレンジの検証と、すべてのDNS APIエンドポイントをサポートするDNSプラグインを使用したdns-01チャレンジの検証をサポートしています。[acme.sh](#)が行います。

ACME アカウント

Register Account

Account Name: default

E-Mail: admin@example.com

ACME Directory: Let's Encrypt V2
<https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf>

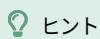
Accept TOS

Register

クラスタごとに、使用するエンドポイントで ACME アカウントを登録する必要があります。そのアカウントに使用される電子メールアドレスは、ACME エンドポイントからの更新期限または同様の通知の連絡先として機能します。

ACMEアカウントの登録や停止は、WebインターフェースのCertificates -> ACME Accounts、またはproxmox-backup-managerコマンドラインツールで行うことができます。

```
proxmox-backup-manager acme account register<アカウント名><mail@example.com>
```



レート制限のため、実験や初めてACMEを使用する場合は、LEステージングを使用する必要があります。

ACME プラグイン

ACMEプラグインの役割は、あなた、ひいてはあなたの運用下にあるProxmox Backup Serverがドメインの本当の所有者であることを自動的に確認することです。これは自動証明書管理の基本的な構成要素です。

例えばhttp-01では、ウェブサーバーがドメインを管理していることを証明するために、特定のトークンを含むファイルを提供します。時々、これは

技術的な制限や、レコードのアドレスが公共のインターネットから到達できない場合などです。dns-01チャレンジはこのような場合に使用できます。このチャレンジは、ドメインのゾーンに特定のDNSレコードを作成することで実行されます。

Add: ACME DNS Plugin

Plugin ID: pdns1

Validation Delay: 30

DNS API: PowerDNS server

PDNS_ServerId: pdns1

PDNS_Token: sl1GDX8fbDPKX/UT6aDpHEGukUwWqVv1y3zFgODYBXM=

PDNS_Ttl: 10

PDNS_Url: 192.168.0.2

Add

Proxmox Backupはこれらのチャレンジタイプの両方をサポートしており、Webインターフェイスの[証明書]->[ACMEチャレンジ]または

`proxmox-backup-manager acme plugin add` コマンドを使用します。

ACMEプラグインの設定は`/etc/proxmox-backup/acme/plugins.cfg`に保存されます。

ドメイン

新しいドメインエントリを追加したり、既存のドメインエントリを管理するには、証明書の下にある `proxmox-backup-manager`コマンドを使用します。

Create: Domain	
Challenge Type:	DNS
Plugin:	pdns1
Domain:	pbs-demo.example.com
Create	

ノードの希望するドメインを設定し、希望する ACME アカウントが選択されていることを確認した後、Web インターフェースで新しい証明書を注文できます。成功すると、インターフェースはおよそ10秒後にリロードされます。

更新は[自動的に行われます](#)。

13.3.4 ACME HTTP チャレンジプラグイン

ポート 80 で生成される内蔵 Web サーバー経由で `http-01` チャレンジを検証するための、暗黙的に構成されたスタンドアロンプラグインが常にあります。

i 注意

スタンドアロンという名前は、サードパーティのサービスなしで、それ自身で検証を提供できることを意味します。

これをLet's Encrypt ACMEで証明書管理に使用するには、いくつかの前提条件があります。

- アカウントを登録するには、Let's EncryptのToSに同意する必要があります。
- ノードのポート**80**がインターネットから到達可能である必要があります。
- ポート 80 に他のリスナーが存在しないこと。
- 要求された(サブ)ドメインはProxmox BackupホストのパブリックIPに解決する必要があります。

13.3.5 ACME DNS APIチャレンジプラグイン

`http-01`メソッドによる検証のための外部アクセスが不可能または望ましいシステム上では、`dns-01`検証メソッドを使用することができます。この検証方法には、API経由でTXTレコードをプロビジョニングできるDNSサーバーが必要です。

検証のためのACME DNS APIの設定

Proxmox Backupは`acme.sh`¹プロジェクト用に開発されたDNSプラグインを再利用します。特定のAPIの設定の詳細については、そのドキュメントを参照してください。

DNS APIを使用して新しいプラグインを設定する最も簡単な方法は、Webインターフェース（Certificates → ACME Accounts/Challenges）を使用することです。

ここでは、APIプロバイダーを選択し、そのAPIを介してアカウントにアクセスするためのクレデンシャルデータを入力することで、新しいチャレンジプラグインを追加できます。

¹ `acme.sh` <https://github.com/acmesh-official/acme.sh>

💡 ヒント

プロバイダのAPIクレデンシャル取得に関する詳細情報については、[acme.sh How to use DNS API wiki](#)を参照してください。プラグインによってはエラーになるものもあります。

多くのDNSプロバイダとAPIエンドポイントがあるため、Proxmox Backupは自動的に認証情報のフォームを生成します。

フォームを自動的に生成しますが、すべてのプロバイダがまだアノテーションされているわけではありません。そのようなプロバイダには大きなテキストエリアが表示され、そこにクレデンシャルのKEY= VALUEペアをすべてコピーする必要があります。

CNAMEエイリアスによるDNS検証

プライマリ/リアルDNSがAPI経由のプロビジョニングをサポートしていない場合、特別なエイリアスマードを使用して、別のドメイン/DNSサーバで検証を処理することができます。手動で`_acme-challenge.domain1.example`を指す`_acme-challenge.domain2.example`のパーマネントCNAMEレコードを設定します。

`domain2.example`を指す`_acme-challenge.domain1.example`のパーマネントCNAMEレコードを手動で設定し、Proxmox Backupノード設定ファイルでエイリアスのプロパティを設定します。

`domain2.example`のDNSサーバーが`domain1.example`のすべてのチャレンジを検証できるように、`/etc/proxmox-backup/node.cfg`のエイリアス・プロパティを`domain2.example`に設定します。

ワイルドカード証明書

ワイルドカードDNS名は*.プレフィックスで始まり、検証されたドメインのすべての(1レベル)サブドメイン名に対して有効とみなされます。したがって、`*.domain.example`の証明書は、`foo.domain.example`と`bar.domain.example`には有効ですが、`baz.foo.domain.example`には有効ではありません。

現在、DNSチャレンジタイプではワイルドカード証明書しか作成できません。

プラグインの組み合わせ

異なる要件/DNSプロビジョニング機能を持つ複数のドメインを経由してノードに到達可能な場合、`http-01`と`dns-01`の検証を組み合わせることができます。複数の

プロバイダまたはインスタンスのDNS APIを混在させることもできます。

💡 ヒント

複数のドメインで同じサービスにアクセスすると複雑さが増すため、可能であれば避けるべきです。

13.3.6 ACME証明書の自動更新

ノードが（`proxmox-backup-manager`経由またはWeb-interface/API経由で）ACME提供の証明書で正しく設定された場合、証明書は`proxmox-backup-daily-update.service`によって自動的に更新されます。現在のところ、証明書の更新は証明書がすでに期限切れであるか、今後30日以内に期限切れになる場合に更新されます。

13.3.7 コマンドラインによる証明書の手動変更

証明書検証の警告を消したい場合は、サーバに有効な証明書を生成する必要があります。

Proxmoxバックアップにsshでログインするか、コンソールを使用します：

```
openssl req -newkey rsa:2048 -nodes -keyout key.pem -out req.pem
```

画面に表示される指示に従ってください：

国名（2文字のコード）[AU] : AT
 都道府県名（フルネーム）[州]：ウェーン
 都道府県名（市區町村など）[:]：ウェーン
 組織名（例、会社）[Internet Widgets Pty Ltd] : Proxmox GmbH
 組織名（例：セクション）[:] : Proxmox Backup
 共通名（例：お客様の名前）[:] : yourproxmox.yourdomain.com メールアドレス
 [:support@yourdomain.com]
 証明書要求とともに送信される以下の「追加」属性を入力してください。チャレンジパスワード[:]：必要ありません。
 任意の会社名[:]：必要ありません

証明書要求が完了したら、`req.pem`ファイルを認証局(CA)に送信する必要があります。CAは要求に基づいて証明書(BASE64エンコード)を発行します。このファイルを`cert.pem`としてProxmox Backupに保存します。

新しい証明書をアクティブにするには、Proxmoxバックアップで次の操作を行います。

```
cp key.pem /etc/proxmox-backup/proxy.key cp cert.pem  
/etc/proxmox-backup/proxy.pem
```

その後、APIサーバを再起動します：

```
proxmox-backup-proxy を再起動します
```

ブラウザを使って新しい証明書をテストします。

注意

Proxmox Backupとの間でファイルを転送するには、セキュアコピーを使用できます：デスクトップがLinuxの場合、`scp`コマンドラインツールを使用できます。デスクトップPCがWindowsの場合は、WinSCPのような`scp`クライアントを使用してください (<https://winscp.net/> を参照)。

13.4 外部メトリックサーバ

Proxmox Backup Serverは、設定された外部メトリックサーバに、ホストのメモリ、ネットワーク、ディスクのアクティビティに関するさまざまなメトリックを定期的に送信します。

現在サポートされているのは

- InfluxDB (HTTP) (<https://docs.influxdata.com/influxdb/v2/> を参照)
- InfluxDB (UDP) (<https://docs.influxdata.com/influxdb/v1/> を参照)

外部メトリックサーバーの定義は`/etc/proxmox-backup/metricserver.cfg`に保存され、Webインターフェースから編集できます。

注意

InfluxDB v2ではUDPのサポートが廃止されたため、HTTPの使用を推奨します。

13.4.1 InfluxDB (HTTP) プラグインの設定

InfluxDB 1.8.xには、このv2 APIと互換性のあるAPIエンドポイントが含まれています。

InfluxDBのv2 APIは認証がないと利用できないため、正しいバケットに書き込めるトークンを生成して設定する必要があります。

1.8.xのv2互換APIでは、「user:password」をトークンとして使用することができます(必要な場合)。

また、「max-body-size」設定で最大バッチサイズ(デフォルトは25000000バイト)を設定できます(これは同名のInfluxDB

設定に対応します)。

13.4.2 InfluxDB (UDP) プラグインの設定

Proxmox Backup ServerはUDP経由でデータを送信することもできます。これにはInfluxDBサーバーが正しく設定されている必要があります。必要に応じてMTUもここで設定できます。

以下はInfluxDBの設定例です（InfluxDBサーバ上）：

```
[udp
  enabled=true
  バインドアドレス="0.0.0.0:8089"
  データベース="proxmox"
  バッчサイズ=1000
  バッчタイムアウト="1s"]
```

この設定により、InfluxDBサーバーはポート8089ですべてのIPアドレスをリッスンし、*proxmox*データベースにデータを書き込みます。

13.5 サービスデーモン

13.5.1 proxmox-backup-proxy

このデーモンはTCPポート8007でHTTPSを使用してProxmox Backup Server API全体を公開します。ユーザーバックアップとして実行され、権限は非常に制限されています。より多くのパーミッションを必要とする操作はローカルのproxmox-backupサービスに転送されます。

13.5.2 proxmox-backup

このデーモンは127.0.0.1:82でProxmox Backup Server管理APIを公開します。このデーモンはrootとして実行され、すべての特権操作を実行する権限を持っています。

注意：デーモンはローカルアドレスのみをリッスンするので、外部からアクセスすることはできません。そのため、外部からアクセスすることはできません。

デーモンはAPIを外部に公開します。

13.6 コマンドラインツール

13.6.1 proxmox-backup-client

このツールはバックアップサーバークライアントを実装しており、バックアップサーバに接続して管理コマンドを発行したり、バックアップを作成したりリストアしたりできます。

13.6.2 proxmox-バックアップマネージャ

このツールはコマンドラインでバックアップサーバー管理API全体を公開します。

13.6.3 proxmox-tape

このツールはテープバックアップの設定と管理ができます。

13.6.4 pmt

pmtコマンドはLinuxテープデバイスを制御します。

13.6.5 pmtnx

pmtnxコマンドはSCSIメディアエンジヤーデバイス(テープオートローダー)を制御します。

13.6.6 pxar

`pxar` は *Proxmox* ファイルアーカイブ形式 (.pxar) のアーカイブを作成・操作するためのコマンドラインユーティリティです。同様のユースケースに対応する `casync` ファイルアーカイブフォーマットから着想を得ています。この .pxar フォーマットは、*Proxmox Backup Server* の特定のニーズを満たすように調整されています、ハードリンクの効率的な保存このフォーマットは、高度な重複排除を実現することを削減するように設計されています。

アーカイブの作成

以下のコマンドを実行して、`source` という名前のフォルダのアーカイブを作成します：

```
# pxar create archive.pxar /path/to/source
```

これにより、`archive.pxar` という新しいアーカイブが作成され、`source` フォルダの内容が格納されます。

i 注意

`pxar` は既存のアーカイブを上書きしません。ターゲット フォルダに同じ名前のアーカイブがすでに存在する場合、作成は失敗します。

デフォルトでは、`pxar` は特定のマウントポイントをスキップし、デバイスの境界に従いません。この設計上の決定は、バックアップ用のアーカイブ作成という主なユースケースに基づいています。を無視することは理にかなっています。

を無視することは理にかなっています。この動作を変更してデバイスの境界に従うようにするには、`--all-file-systems` フラグを使用します。

特定のファイルやフォルダをアーカイブから除外するには、`--exclude`

パラメータに `gitignore` 形式のマッチパターンを指定します。

たとえば、`.txt` で終わるすべてのファイルをアーカイブから除外するには、次のようにします：

```
# pxar create archive.pxar /path/to/source --exclude '**/*.txt'
```

`pxar` を起動する前に、シェル自体がグロブ・パターンを展開しようとすることに注意してください。これを避けるには、すべてのグロブを正しく引用符で囲む必要があります。

複数のパターンにマッチさせるために、`--exclude` パラメータを複数回渡すことができます。これにより、より複雑なファイルの包含/除外動作を使用できます。しかし、このような場合は代わりに `.pxarexclude` ファイルを使用することをお勧めします。

例えば、特定のファイル以外のすべての `.txt` ファイルをアーカイブから除外したい場合です。これは、先頭に `!` を付いた否定マッチパターンで実現できます。グロブパターンはすべてソースディレクトリからの相対パスです。

```
# pxar create archive.pxar /path/to/source --exclude '**/*.txt' --exclude '!/folder/file.txt' # pxar create archive.pxar
```

```
/path/to/source --exclude '**/*.txt' --exclude '!'
```

i 注

グロブ・マッチ・パターンの順序は重要です。同じパターンを並べ替えると、異なる結果になります。

`pxar` は、コマンド ラインからパラメータとして渡されたグロブ マッチ パターンのリストを、アーカイブのルートにある `.pxarexclude-cli` というファイルに格納します。アーカイブ作成時に、この名前のファイルがソース フォルダに既に存在する場合、このファイルはアーカイブに含まれず、新しいパターンを含むファイル

がアーカイブに追加されます。元のファイルは変更されません。

アーカイブからファイルを除外する、より便利で永続的な方法は、グロブ マッチ パターンを `.pxarexclude` ファイルに配置することです。このファイルは、ファイルシステム・ツリーの任意のディレクトリに作成して配置することができます。これらのファイルには、1行に1つのパターンを含める必要があります。

を上書きします。パターンは、指定されたディレクトリ内またはツリー内のさらに下にあるファイルの排他を制御します。動作はバックアップの作成で説明したものと同じです。

アーカイブの抽出

既存のアーカイブarchive.pxarは、次のコマンドで目的のディレクトリに展開されます：

```
# pxar extract archive.pxar /path/to/target
```

ターゲットが指定されない場合、アーカイブの内容は現在の作業ディレクトリに展開されます。

アーカイブの一部、単一ファイル、フォルダのみを復元するには、対応するグローブ マッチ パターンを追加パラメータとして渡すか、ファイルに格納されているパターンを使用します：

```
# pxar extract etc.pxr /restore/target/etc --pattern '**/*.*conf'
```

上記の例では、アーカイブ etc.pjar内のサブフォルダで見つかったすべての.confファイルをターゲット /restore/target/etcにリストアします。マッチパターンを含むファイルへのパスは、-files-from パラメータを使用して指定できます。

アーカイブの内容の一覧表示

アーカイブ archive.pxr に含まれるファイルとディレクトリを表示するには、次のコマンドを実行します。

```
# pxar list archive.pxr
```

これにより、アーカイブのルートを基準とした各ファイルやディレクトリのフルパスが表示されます。

アーカイブのマウント

`pxar`を使用すると、FUSE経由でアーカイブをマウントし、その内容を検査することができます。`archive.pxar`という名前のアーカイブをマウントポイント/`mpt`にマウントするには、次のコマンドを実行します：

```
# pxar mount archive.pxar /mnt
```

アーカイブがマウントされると、指定したマウントポイント下でそのコンテンツにアクセスできます。

```
# cd /mnt  
# ls
```

13.6.7 プロックスモックスファイルリストア

Proxmox Backup アーカイブからファイルやディレクトリをリストアするコマンドラインツールです。proxmox-backup-client とは対照的に、コンテナ/ホストと VM の両方のバックアップをサポートします。

13.6.8 proxmox-backup-debug

Proxmox Backupデータストアのファイルを検査し、モダンクの整合性を検証するバックアップ機能を実装します。

`diff` サブコマンドにより、2つの任意のスナップショットの .pxar アーカイブを比較できます。追加/変更/削除されたファイルの一覧が表示されます。

また、'api'サブコマンドを含んでおり、任意のapiパスを呼び出し(get/create/set/delete)、パラメータ(usage)や子リンク(ls)を表示することができます。

デフォルトでは、https経由でlocalhost上のproxmox-backup-proxyに接続しますが、環境変数`PROXMOX_DEBUG_API_CODE`を1に設定することで、ツールは対応するコードを直接呼び出します。



`PROXMOX_DEBUG_API_CODE` を使用することは危険であり、デバッグのみを目的としています。本番システムでの使用は意図されていません。

ネットワーク管理

The screenshot shows the Proxmox Backup Server 2.2-0 web interface. The left sidebar contains navigation links for Dashboard, Notes, Configuration (with sub-links for Access Control, Remotes, Traffic Control, Certificates, Subscription), Administration (Shell, Storage / Disks), Tape Backup, and Datastore (with sub-links for qpib, aantreich, alauterer, djaeger, fgruenbichler, malmalat, milimbeck, obektas, store1, tlamprecht, wbumiller, wlink, and Add Datastore). The main content area is titled 'Configuration: System' and shows the 'Network/Time' tab selected. It displays the 'Time' section with Time zone set to Europe/Vienna and Server time as 2022-05-17 12:01:12, and the 'DNS' section with Search domain set to proxmox.com and DNS server 1 set to 192.168.2.15. Below these is the 'Network Interfaces' section, which lists network interfaces: eno1 (Network Device, Yes, No, No), eno2 (Network Device, No, No, No), ens5f0 (Network Device, Yes, Yes, No, 192.168.16.120/20, 192.168.16.1), and ens5f1 (Network Device, No, No, No).

Proxmox Backup Serverには、ネットワーク設定のためのWebインターフェースとコマンドラインツールの両方が用意されています。設定オプションは、Webインターフェイスのネットワークインターフェイスセクションにあります。コマンドラインツールは、ネットワークサブコマンドでアクセスします。これらのインターフェイスを使用すると、基本的なネットワーク管理タスクを実行できます、これらのインターフェイスを使用すると、ネットワークインターフェイスの追加、設定、削除など、基本的なネットワーク管理タスクを実行できます。

注意

ネットワーク設定に加えた変更は、【設定の適用】をクリックするか、`network reload` コマンドを入力するまで適用されません。これにより、一度に多くの変更を行うことができます。また、以下のように、適用する前に変更が正しいことを確認できます。

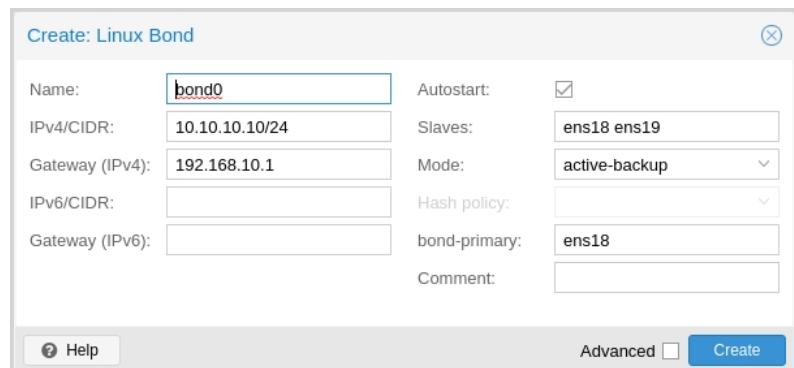
ここを間違えると、ネットワーク経由でサーバーにアクセスできなくなります。

利用可能なインターフェースの一覧を取得するには、以下のコマンドを使用します：

# proxmox-backup-manager network list						
名前	タイプ	オートスタート	方法	アドレス	ゲートウェイ	ポート/スレーブ
bond0	ボンド	1	静的	x.x.x.x/x	x.x.x.x	ens18 ens19
ens18	eth	1	マニュアル			
ens19	eth	1	マニュアル			

新しいネットワークインターフェースを追加するには、関連するパラメータを指定して `create` サブコマンドを使用します。例えば、ネットワークの冗長性のためにボンドを設定することができます。次のコマンドは、上のリストに示したボンドを作成するためのテンプレートを示しています：

```
# proxmox-backup-manager network create bond0 --type bond --bond_mode active-backup --slaves
  ↪ens18,ens19 --autostart true --cidr x.x.x/x --gateway x.x.x.x
```



`update`サブコマンドでネットワークインターフェースの設定を変更できます：

```
# proxmox-backup-manager network update bond0 --cidr y.y.y/y
```

ネットワークインターフェースを削除することもできます：

```
# proxmox-backup-manager network remove bond0
```

ネットワーク設定ファイルの保留中の変更がウェブインターフェイスの下部に表示されます。コマンドを使ってこれらの変更を見ることもできます：

```
# proxmox-backup-manager network changes
```

この時点で全ての変更をキャンセルしたい場合は、**Revert**ボタンをクリックするか、以下のコマンドを使用します：

```
# proxmox-backup-manager network revert
```

変更内容に問題がなく、設定ファイルに書き込む場合は、次のように選択します。
を選択します。対応するコマンドは

```
# proxmox-backup-manager network reload
```

注

このコマンドと対応するGUIボタンは、`ifupdown2`パッケージの`ifreload`コマンドに依存します。このパッケージは Proxmox Backup Serverインストールに含まれていますが、Proxmox Backup Serverを Debian または Proxmox VE バージョン 7 より前のもの。

DNS設定は、コンフィギュレーションの**DNS**セクションから、または`dns`サブコマンドを使って設定することもできます。

14.1 トラフィックの制御

Edit: Traffic Control Rule

Name:	weekday-in-office-limit	Comment:	don't trash the office net						
Rate In:	10 MiB/s	Burst In:	Same as Rate MiB/s						
Rate Out:	50 MiB/s	Burst Out:	Same as Rate MiB/s						
Network(s):	0.0.0.0/0, ::/0 (Apply on all Networks)								
Timeframes:									
Time Start	Time End	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
07:00	19:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	+ Add				

+ Add

? Help OK Reset

バックアップの作成と復元には多くのトラフィックが発生し、共有ストレージやネットワーク上の他のユーザに影響を与える可能性があります。

Proxmox Backup Serverでは、トークンバケットフィルタ(TBF)を使用して、指定したネットワーク内のクライアントのネットワークトラフィックを制限することができます。

これにより、ネットワークの輻輳を回避し、特定のホストからのトラフィックを優先することができます。

トラフィック制御は、Webインターフェイスまたはproxmox-backup-managerコマンドラインツールのtraffic-controlコマンドを使用して管理できます。トラフィックはレート(rate-inとrate-out)で制限され、トークンバケットサイズ(burst-inとburst-out)を設定することで短いバーストを可能にします。

注意

サーバー上の同期ジョブは、設定されたレート制限の影響を受けません。プル型同期ジョブの受信トラフィックまたはプッシュ型同期ジョブの送信トラフィックを制限したい場合は、ジョブ固有のレートイン制限を設定する必要があります。[同期ジョブ](#)を参照してください。

以下のコマンドは、すべてのIPv4クライアント(ネットワーク0.0.0.0/0)を100MB/sに制限するトラフィック制御ルールを追加します：

```
# proxmox-backup-manager traffic-control create rule0 --network 0.0.0.0/0 --rate-in 100MB --rate-out 100MB --comment "全クライアントのデフォルトレートリミット(100MB/s)"
```

1 注

IPv4とIPv6の両方のネットワーク空間を制限するには、2つのネットワークパラメータ ::/0 と 0.0.0.0/0 を渡す必要があります。

会社の営業時間など、特定の時間帯にルールを制限することも可能です：

💡 ヒント

SI単位（10進数：KB、MB、...）またはIEC単位（2進数：KiB、MiB、...）を使用できます。

```
# proxmox-backup-manager traffic-control update rule0 \
--timeframe "mon..fri 8-12" \
--timeframe "月,金14:30-18"
```

複数のルールがある場合、サーバーはより小さいネットワークのものを選択します。例えば、プライベートネットワーク（とサーバー自体）の設定を上書きします：

```
# proxmox-backup-manager traffic-control create rule1 \
--network 192.168.2.0/24 \
--network 127.0.0.0/8 \
--rate-in 20GB --rate-out 20GB \
--comment "ローカルネットワークに20GB/sを使用"
```

1 注意

同じネットワークに対して複数のルールがある場合の動作は未定義です。

特定のネットワークにマッチするルールが複数ある場合、それらはすべて適用され、バケツが最も早く満杯になるため、最も小さいものが勝つことになります。

現在のルールを一覧表示するには

# proxmox-backup-manager トライックコントロールリスト		
name	rate-in	rate-out network
rule0	100 MB	100 MB ["0.0.0.0/0"] ["mon..fri 8-12", "月,金14:30-18"]
rule1	20 GB	20 GB ["192.168.2.0/24", ...]

ルールは削除することもできます：

```
# proxmox-backup-manager traffic-control remove rule1
```

設定されている全てのルールの状態(現在のデータレート)を表示するには以下のコマンドを使います：

# proxmox-backup-manager traffic-control traffic		
名前	cur-rate-in	cur-rate-out
ルール0	0 B	0 B
rule1	1.161 GiB	19.146 KiB

14.2 ネットワークデバイス名の上書き

カーネルのアップグレード、PCIeデバイスの追加、BIOSのアップデートを行うと、自動的に生成されるネットワークリ

インターフェイス名が変更されることがあります。この問題を軽減するために、Proxmox Backup Serverは、ネットワークデバイスの名前を上書きするための.linkファイルを自動的に生成するツールを提供します。また、*/etc/network/interfaces*内の古いインターフェイス名を自動的に置き換えます。

生成されたリンクファイルは `/usr/local/lib/systemd/network` に保存されます。interfaces ファイルについては、同じ場所に接尾辞 `.new` を付けて新しいファイルが生成されます。このようにして、`diff` (またはお好みの `diff` ビューア) を使って、設定に加えられた変更を調べることができます：

14.2.1 `diff -y /etc/network/interfaces /etc/network/interfaces.new`

問題のある変更が見つかった場合、または `pin-ning` ツールによる変更をリブートする前に元に戻したい場合は、単にすべての `.new` ファイルとそれぞれのリンクファイルを `/usr/local/lib/sysfaces` から削除してください。
/からすべての `.new` ファイルとそれぞれのリンクファイルを削除します。

次のコマンドを実行すると、`.link` ファイルがまだないすべての物理ネットワーク・インターフェースの `.link` ファイルが生成され、選択した Proxmox VE 設定ファイル（上記参照）が更新されます。生成された名前はデフォルトのプレフィックス `nic` を使用するため、結果のインターフェース名は `nic1`、`nic2`、... となります。

14.2.2 `proxmox-network-interface-pinning` が生成されます。

デフォルトのプレフィックスは `--prefix` フラグで上書きできます：

14.2.3 `proxmox-network-interface-pinning generate --prefix myprefix`

また、特定のインターフェースだけをピン留めすることもできます：

14.2.4 `proxmox-network-interface-pinning generate --interface enp1s0`

特定のインターフェースをピン留めする場合、そのインターフェースをピン留めする正確な名前を指定できます：

14.2.5 `proxmox-network-interface-pinning generate --interface enp1s0 --target-name if42`

`proxmox-network-interface-pinning` によって行われた変更をネットワーク設定に適用するには、ホストを再起動する必要があります。

通知

15.1 概要

- Proxmox Backup Serverは、システムで注目すべきイベントが発生すると**通知イベント**を発行します。これらのイベントはグローバル通知設定に基づいて処理されます。各通知イベントには、タイムスタンプ、重大度レベル、タイプ、イベント固有の追加フィールドなどのメタデータが含まれます。
- 通知マッチャ**は、通知イベントを1つ以上の通知ターゲットにルーティングします。マッチャは、通知イベントのメタデータに基づいて選択的にルーティングするマッチルールを持つことができます。
- 通知ターゲット**は、マッチャーによって通知イベントがルーティングされる宛先です。ターゲットには複数のタイプがあり、メールベース（SendmailとSMTP）とGotifyがあります。

グローバル通知設定は、GUIの[Configuration Notifications]で構成できます。設定は、*notifications.cfg*と*notifications-priv.cfg*に保存されます。後者には、通知ターゲットのパスワードや認証トークンのような機密性の高い設定オプションが含まれています。

`root`に読まれます。

15.2 通知ターゲット

Proxmox Backup Serverは複数の通知ターゲットを提供しています。

15.2.1 センドメール

sendmailバイナリは、電子メールメッセージの送信を処理するUnixライクなオペレーティングシステムで一般的に見られるプログラムです。コマンドラインユーティリティで、ユーザやアプリケーションはコマンドラインから、またはスクリプト内から直接電子メールを送信できます。

sendmail通知ターゲットは sendmail バイナリを使用して、設定されたユーザーまたはメールアドレスのリストに電子メールを送信します。ユーザが受信者として選択された場合、ユーザの設定で構成された電子メールアドレスが使用されます。`root@pam`ユーザーの場合、これはインストール時に入力された電子メールアドレスです。

の場合、これはインストール時に入力された電子メール・アドレスです。ユーザーの電子メール・アドレスは、設定→アクセス制御→ユーザー

管理ユーザーに関するメールアドレスがない場合、メールは送信されません。



注意

Proxmox Backup Serverの標準インストールでは、sendmailバイナリはPostfixによって提供されます。Postfixがメールを正しく配信できるように設定する必要がある場合があります。

外部メールリレー(スマートホスト)を設定します。配信に失敗した場合は、Postfixデーモンによって記録されたメッセージをシステムログで確認してください。

すべての設定オプションについては、*notifications.cfg* を参照してください。

15.2.2 SMTP

SMTP通知ターゲットはSMTPメールリレーに直接メールを送信できます。このターゲットは、電子メールの配信にシステムのMTAを使用しません。sendmail ターゲットと同様に、ユーザが受信者として選択されている場合、ユーザの構成済み電子メール・アドレスが使用されます。

i 注意

sendmail ターゲットとは異なり、SMTP ターゲットにはメール配送に失敗した場合のキューイング/リトライ機構はありません。

すべての設定オプションについては、[notifications.cfg](#) を参照してください。

15.2.3 Gotify

Gotify は、様々なデバイスやアプリケーションにプッシュ通知を送ることができるオープンソースのセルフホスト型通知サーバーです。シンプルなAPIとウェブインターフェースを提供しており、様々なプラットフォームやサービスと簡単に統合することができます。

i 注意

Gotify のターゲットは、その他の HTTP プロキシ設定からの HTTP プロキシ設定を尊重します。

すべての設定オプションについては、[notifications.cfg](#) を参照してください。

15.2.4 ウェブフック

Webhook 通知ターゲットは、設定可能な URL への HTTP リクエストを実行します。以下の設

定オプションが利用可能です：

- **url** : HTTP リクエストを実行する URL。メッセージ
コンテンツ、メタデータ、およびシークレットを注入するためのテンプレート化をサポートします。
- メソッド : 使用する HTTP メソッド (POST/PUT/GET)
- ヘッダの配列 : リクエストに設定されるべき HTTP ヘッダの配列。メッセージの内容、メタデータ、秘密を
注入するためのテンプレート化をサポートします。
- ボディ : 送信されるべき HTTP ボディ。メッセージの内容、メタデータ、および秘密を注入するためのテンプレ
ート化をサポートします。
- **secret**: 秘密のキーと値のペアの配列。これらはrootのみが読める保護された設定ファイルに保存されます。シ
ークレットは、**secrets** 名前空間を介して body/header/URL テンプレートでアクセスできます。
- **comment** : このターゲットに対するコメント。

テンプレート化をサポートする設定オプションの場合、Handlebars構文を使用して以下のプロパティにアクセスできま
す：

- **{title}** : レンダリングされた通知タイトル
- {メッセージレンダリングされた通知本文
- **{{ severity }}** : 通知の重大度 (**info**, **notice**, **warning**, **error**, **unknown**)
- **{{ timestamp }}** : UNIXエポックとしての通知のタイムスタンプ (秒単位)。
- **{{ fields.<name> }}** : 通知のメタデータフィールドのサブネームスペース。in-stanceの場合、**fields.type**には
通知タイプが含まれます。利用可能なすべてのフィールドについては、[通知イベント](#)を参照してください。

- {{ secrets.<name>} } : secretsのサブネームスペース。例えば、token
という名前の秘密はsecrets.tokenでアクセスできます。

便宜上、以下のヘルパーが利用可能です：

- {{ url-encode <value/property>} } : プロパティ/リテラルをURLエンコードします。
- {{ escape <value/property>} } : JSON文字列として安全に表現できない制御文字をエスケープします。
- {{ json <value/property>} } : 値をJSONとしてレンダリングします。これは、JSONペイロードの一部としてサブ名前空間全体 (fieldsなど) を渡すのに便利です (例：{{ json fields }})。

注意

Webhook ターゲットは、Configuration Other HTTP proxy の HTTP プロキシ設定を尊重します。

例 - ntfy.sh

- メソッドPOST
- URL: <https://ntfy.sh/{{ secrets.channel }}>
- ヘッダ
 - マークダウン：はい
- ボディ

```
```  
{メッセージ
```
```

- 秘密
 - チャネル: <あなたの ntfy.sh チャネル>。

例 - Discord

- メソッドPOST
- URL: <https://discord.com/api/webhooks/{{ secrets.token }}>
- ヘッダ
 - コンテンツタイプ: application/json
- ボディ

```
{  
  "content": "```\n{{ エスケープメッセージ }}\n```"  
}
```

- 秘密
 - トークン: <トークン

例 - Slack

- メソッドPOST
- URL: <https://hooks.slack.com/services/{{ secrets.token }}>
- ヘッダー
 - コンテンツタイプ: application/json
- ボディ

```
{
  "text": "``{{escape message}}`",
  "type": "mrkdwn"
}
```

- 秘密
 - トークン: <トークン

15.3 通知マッチャー

通知マッチャーは、そのマッチングルールに基づいて通知を通知ターゲットにルーティングします。

ルールに基づきます。これらのルールは、タイムスタンプ(match-calendar)、通知の重大度(match-severity)、メタデータフィールド(match-field)など、通知の特定のプロパティにマッチすることができます。通知がマッチャーによってマッチされると、マッチャー用に設定されたすべてのターゲットが通知を受け取ります。

任意の数のマッチャーを作成することができ、それが独自のマッチングルールと通知するターゲットを持ちます。ターゲットが複数の matcher で使用されている場合でも、全てのターゲットは全ての通知に対して最大一度だけ通知されます。

設定されたターゲットは常に通知されます。すべての設定オプションについては、[notifications.cfg](#) を参照してください。

。

15.3.1 カレンダーマッチングルール

カレンダーマッチャは、通知のタイムスタンプにマッチします。例

- match-calendar 8-12
- match-calendar 8:00-15:30
- match-calendar 月-金 9:00-17:00
- match-calendar 日、火、水、金 9-17

15.3.2 フィールドマッチ規則

通知には、マッチング可能なメタデータ・フィールドがあります。マッチング・モードとしてexactを使用する場合、**、**を区切り文字として使用できます。マッチング・ルールは、メタデータ・フィールドが指定された値のいずれかを持つ場合にマッチングします。

例

- match-field exact:type=gc ガベージコレクションジョブの通知のみにマッチします。
- match-field exact:type=prune,verify プルーン ジョブと検証ジョブの通知にマッチします。
- match-field regex: datastore=^backup-.*\$ backup で始まるデータストアにマッチします。

バックアップ。

notification がマッチしたフィールドを持たない場合、ルールはマッチしません。例えば、match-field regex: datastore=.*ディレクティブは、データストアのメタデータフィールドを持つすべての通知にマッチしますが、フィールドが存在しない場合はマッチしません。

15.3.3 重大度のマッチングルール

notificationはマッチすることができる関連する重大度を持っています。例

- **match-severity error** : エラーにのみマッチします。
- **match-severity warning,error** : 警告とエラーにマッチ

次の重大度が使用されます: info、notice、warning、error、unknown。

15.4 通知イベント

次の表は、Proxmox Backup サーバーのすべての通知イベントのリスト、そのタイプ、重大度、および追加のメタデータフィールドを含んでいます。タイプだけでなく、その他のメタデータフィールドもマッチフィールドのマッチルールで使用できます。

イベント	タイプ	重大度	メタデータフィールド（タイプに加えて）
ACME 証明書の更新に失敗	アクメ	エラー	ホスト名
ガベージコレクションの失敗	gc	エラー	データストア、ホスト名
ガベージコレクション成功	gc	情報	データストア、ホスト名
成功			
パッケージ 更新 avail- 可能	パッケージ更新	情報	ホスト名
プルーンの失敗	プルーン	エラー	データストア、ホスト名、ジョブID
プルーンジョブの成功	プルーン	情報	データストア、ホスト名、ジョブID
リモート同期失敗	同期	エラー	データストア、ホスト名、ジョブID
リモート同期成功	同期	情報	データストア、ホスト名、ジョブID
テープバックアップジョブの失敗	テープバックアップ	エラー	データストア、ホスト名、メディアプール ジョブID
テープバックアップジョブの成功	テープバックアップ	情報	データストア、ホスト名、メディアプール ジョブID
テープロード要求	テープロード	通知	ホスト名
検証ジョブの失敗	検証	エラー	データストア、ホスト名、ジョブID
検証ジョブの成功	検証	情報	データストア、ホスト名、ジョブID

次の表は、すべての使用メタデータ・フィールドの説明です。これらはすべて

マッチフィールドのマッチルールで使用することができます。

メタデータ・フィールド		説明
データストア		データストアの名前
ホスト名		バックアップサーバのホスト名
ジョブID		ジョブID
メディアプール		テープ・メディア・プールの名前
タイプ		通知イベントの種類

注

利用可能なシステム・アップデートをチェックするデイリー・タスクは、ノードにアクティブなサブスクリプションがある場合にのみ通知を送信します。

15.5 システムメールの転送

`smartd`などの特定のローカル・システム・デーモンは、ローカル・ルート・ユーザーに通知メールを送信します。これらのメールは、タイプが `system-mail`、重大度が `unknown` の通知イベントに変換されます。

メールが `sendmail` ターゲットに転送される場合、メールの内容とヘッダーはそのまま転送されます。それ以外のターゲットに対しては、システムはメールのコンテンツから件名と本文の両方を抽出しようとします。メールがHTMLコンテンツのみで構成されている場合、このプロセスでプレーンテキスト形式に変換されます。

15.6 アクセス許可

通知ターゲットの構成を変更/表示するには、`/sys.Modify/Sys.Audit`

パーミッションが必要です。

15.7 通知モード

データストアおよびテープバックアップ/リストア・ジョブ構成には、`notification-mode`オプションがあります：

- グローバル通知設定に基づいて通知を送信します（`notification-system`）。
- システムの `sendmail` コマンド経由で通知メールを送信します（`legacy-sendmail`）。グローバルな通知設定からのtar-getsやmatcherは無視されます。このモードは Proxmox Backup Serverバージョン3.2以前の通知動作と同等です。Proxmox Backup Serverの後のリリースでは削除される可能性があります。

15.7.1 legacy-sendmail通知モードの設定

`notification-mode`が`legacy-sendmail`に設定されている場合、Proxmox Backup Serverはシステムの`sendmail`コマンドを介して、`notify-user`オプションで設定されたユーザに設定されたメールアドレスに通知メールを送信します（設定されていない場合は`root@pam`にフォールバックします）。

データストアの場合、`notify`オプションでタスクタイプごとに受け取る通知のレベルを変更することもできます。オプションで変更できます。

- `Always`: スケジュールされたタスクに対して、結果に関係なく通知を送信します。
- `Errors`: エラーが発生したスケジュールされたタスクに対して通知を送信します。
- `Never`: まったく通知を送信しません。

グローバル通知設定（`notification-mode`が`notification-system`に設定されている）を使用する場合、`notify-user`と`notify`オプションは無視されます。

15.8 通知テンプレートの上書き

Proxmox Backup Serverは通知をレンダリングするためにHandlebarsテンプレートを使用します。Proxmox Backup Serverが提供するオリジナルのテンプレートは`/usr/share/proxmox-backup/templates/default/`に保存されています。

通知テンプレートは、`/etc/proxmox-backup/notification-templates/default/`のoverride directoryにカスタムテンプレートファイルを指定することでオーバーライドできます。指定されたタイプの通知をレンダリングするとき、Proxmox Backup Serverは最初に

ディレクトリからテンプレートをロードしようとします。このテンプレートが存在しないか、レンダリングに失敗すると、元のテンプレートが使用されます。

テンプレートファイルは`<type>-<body|subject>.txt.hbs`という命名規則に従っています。例えば、`gc-error-body.txt.hbs`はガベージコレクションエラーの通知をレンダリングするためのテンプレートを含んでいます。

`package-updates-subject.txt.hbs` は利用可能なパッケージ更新の通知のサブジェクト行をレンダリングするために使われます。

技術的概要

16.1 データストア

データストアは、[バックアップ・スナップショット](#)とそのチャンクが格納される論理的な場所です。スナップショットはマニフェスト、プロブ、動的インデックスおよび固定インデックスで構成され（「[用語](#)」を参照）、以下のディレクトリ構造に格納されます：

<datastore-root>/<type>/<id>/<time>/に格納されます。

データストアの重複排除は、バックアップスナップショット内のインデックスが参照するチャンクの再利用に基づいています。つまり、複数のインデックスが同じチャンクを参照できるため、データを格納するために必要なスペースの量を削減できます（バックアップスナップショット間でも）。

16.2 スナップショット

スナップショットは、バックアップを表すマニフェスト、プロブ、およびインデックスのコレクションです。クライアントがスナップショットを作成するとき、プロブ（チャンクされていない単一ファイル、たとえばクライアント・ログ）、または1つ以上のインデックス（固定または動的）をアップロードできます。

インデックスをアップロードする場合、クライアントはまずソースデータを読み込んでチャンク化し、識別チェックサムとともにチャンクとしてデータをサーバに送信する必要があります。変更検出モードを使用する場合、変更されていないファイルのペイロード・チャンクは以前のスナップショットから再利用されるため、ソース・データを再度読み込む必要はありません。

バックアップグループに以前のスナップショットがある場合、クライアントはまず以前のスナップショットのチャンクリストをダウンロードします。サーバ上に既に存在するチャンクを検出した場合、データとチェックサムの代わりにチェックサムのみを送信できます。このように、スナップショットの実際のアップロードは増分ですが、各スナップショットはすべてのチャンクを参照するため、フルバックアップとなります。

すべてのデータをアップロードした後、クライアントはバックアップが終了したことをサーバに通知する必要があります。接続が閉じる前にそれが行われない場合、サーバは未完成のスナップショットを削除します。

16.3 チャンク

チャンクとは、末尾にCRC-32チェックサム、先頭にタイプマーカーを持つ（暗号化されている可能性のある）データのことです。チャンクはそのコンテンツのSHA-256チェックサムによって識別されます。

このようなチャンクを生成するために、バックアップデータは固定サイズまたは動的サイズのチャンクに分割されます

。同じコンテンツは同じチェックサムにハッシュされます。

データストアのチャンクは

<データストアのルート>/chunks/にあります。

このチャンクディレクトリは、チェックサム2バイトのプレフィックス（16進数4桁で指定）でチャンクをグループ化したディレクトリにさらに細分化されます。

a342e8151cbf439ce65f3df696b54c67a114982cc0aa751f2852c2f7acc19a8b

있습니다。

<datastore-root>/.chunks/a342/ にあります。

これは、ディレクトリあたりのファイル数を減らすために行われます。

これらのチャンク・ディレクトリ ('0000'～'ffff') は、データストアの作成時に事前に割り当てられます。

16.3.1 固定サイズのチャンク

ブロックベースのバックアップ（VMなど）では、固定サイズのチャンクが使用されます。コンテンツ（ディスクイメージ）は同じ長さのチャンク（通常4MiB）に分割されます。

ゲスト上のファイルシステムは多くの場合、連続した断片でファイルを割り当てるようとするため、新しいファイルは新しいブロックを取得し、既存のファイルを変更するとそのブロックだけが変更されます。

最適化として、[Proxmox VE](#)のVMはイメージの変更されたブロックを追跡できる「ダーティ・ビットマップ」を利用できます。これらのビットマップはチャンクに分割されたイメージの表現でもあるため、イメージのダーティブロックとアップロードが必要なチャンクの間には直接的な関係があります。したがって、ディスクの変更されたチャンクだけをバックアップにアップロードする必要があります。

イメージは常に同じサイズのチャンクに分割されるため、変更されていないブロックはそのチャンクのチェックサムが同じになります。そのようなチャンクは再度バックアップする必要はありません。このように、変更されたブロックを見つけるためにストレージのスナップショットは必要ありません。

一貫性を保つために、[Proxmox VE](#)はストレージスナップショットに依存しないQEMU内部スナップショットメカニズムを使用しています。

16.3.2 動的なサイズのチャンク

ブロックベースのシステムではなく、ファイルベースのシステムで作業する場合、固定サイズのチャンクを使用することは良いアイデアではありません。ファイルのサイズが変更されるたびに、残りのデータが移動され、多くのチャンクが変更され、重複排除の量が減少します。

これを改善するために、[Proxmox Backup Server](#)は代わりに動的なサイズのチャンクを使用します。イメージを固定サイズに分割する代わりに、まず一貫性のあるファイルアーカイブ（*pxar*）を生成し、このオンザフライで生成されたアーカイブに対してローリングハッシュを使用してチャンクの境界を計算します。

私たちは、巡回多項式アルゴリズムであるBuzhashの変種を使用しています。このアルゴリズムは、データを繰り返しながらチェックサムを連続的に計算し、特定の条件下でハッシュ境界をトリガーします。

バックアップされるシステム上のほとんどのファイルが変更されていないと仮定すると、最終的にアルゴリズムは以前のバックアップと同じデータで境界をトリガーし、再利用可能なチャンクを生成します。

16.3.3 暗号化チャンク

暗号化チャンクは特殊なケースです。固定サイズのチャンクも動的サイズのチャンクも暗号化することができ、通常のチャンクとは少し異なる方法で扱われます。

暗号化されたチャンクのハッシュは、実際の（暗号化された）チャンクのコンテンツではなく、プレーンテキストのコンテンツに暗号化キーを連結して計算されます。こうすることで、同じデータで異なる鍵で暗号化された2つのチャンクは2つの異なるチェックサムを生成し、複数の暗号化鍵の衝突は発生しません。

これは、実際にアップロードされるチャンクだけを暗号化する必要があるため、バックアップのクライアント部分を高

速化するために行われます。前回のバックアップで既に存在していたチャンクは暗号化されてアップロードされる必要はありません。

16.3.4 ファイルベース バックアップの変更検出モード

変更検出モードは、後続のバックアップ実行の間に変更されなかったファイルを検出して対処する方法と、ディレクトリエンコードするために使用されるアーカイブファイル形式を制御します。

現在のデフォルトのレガシーモードと、データとメタデータの3つのモードがあります。レガシーモードがすべてのコンテンツを単一のpxarアーカイブにエンコードするのに対し、後者の2つのモードはデータとメタデータをppxarアーカイブとmpxarアーカイブに分割します。これは、メタデータモードが再利用可能なファイルを検出するために使用する、前回のスナップショットとのメタデータの高速比較を可能にするために行われます。データモードは、ファイルを無条件に再チャンクすることで、変更されていないファイルの再利用を控えます。このモードはそのため、メタデータが変更されていなくても、ファイルの変更が見逃されることはありません。

i 注

データストアが両方のタイプのアーカイブ スナップショットを保存している場合、pxar と mpxar/ppxar ファイルフォーマットは異なるため、効率的に重複排除できません。

変更検出モードはクライアント側の変更であるため、Proxmox Backup Serverの旧バージョンとの後方互換性があります。ただし、ウェブインターフェースで新しいアーカイブ形式のバックアップコンテンツを検索するには、バージョン3.2.5以上のProxmox Backup Serverが必要です。機能の完全な互換性を確保するには、最新バージョンへのアップグレードをお勧めします。

レガシーモード

ファイルシステムのバックアップスナップショットは、ディレクトリエンコードを再帰的にスキャンすることで作成されます。スナップショットに含まれるすべてのエントリが読み込まれ、pxar アーカイブ形式を使用してエンコードすることによってシリアライズされます。得られたストリームは、動的にサイズ調整されたチャンクにチャンクされ、Proxmoxバックアップサーバーにアップロードされます。

mox Backup Serverへアップロードされ、コンテンツダイジェストに基づいてチャンクが重複排除され、スペース効率よく保存されます。ファイルのコンテンツは無条件で読み込まれ、チャンク化されます。変更されていないファイルを検出するためのチェックは行われません。

データモード

レガシーモードと同様に、ファイルコンテンツは無条件で読み込まれ、チャンク化されます。

しかし、エントリのメタデータとデータを1つの自己完結型のpxarアーカイブに保存するレガシー モードとは対照的に、データモードではメタデータとファイルのコンテンツが2つの別々のストリームにエンコードされます。その結果、バックアップスナップショットには、エントリのメタデータを含むmpxar_{for-mat}アーカイブと、実際のファイルの内容を含むppxar フォーマットのアーカイブという、分割されたアーカイブが含まれます。アーカイブが含まれます。メタデータのアーカイブには

を対応するペイロードアーカイブエントリにオフセットします。これらのアーカイブはどちらも Proxmox バックアップ クライアントによってチャンク化され、アップロードされます。

mpxarアーカイブは、同じチャンク内に格納されたペイロードデータのオーバーヘッドなしに、アーカイブエントリに関連するメタデータを効率的にフェッチするために使用できます。これは例えば以下のように使用されます。

アーカイブの内容を一覧表示したり、FUSE 実装を介してマウントされたファイルシステムをナビゲートするためのエントリ検索に使用します。したがって、このモードを使用してエンコードされたアーカイブには、専用のカタログは作成されません。

メタデータを以前のバックアップスナップショットと比較しないため、メタデータモードとは対照的に、このモードでは再利用可能なファイルはありません。メタデータモードでは、ファイルの内容が変更された後にリストアされるため、ファイルサイズとmtimeは変更されませんが、変更されたファイルを再利用することができます。

メタデータモード

メタデータ・モードは、後続のバックアップ実行の間にファイル・メタデータが変更されなかったファイルを検出します。メタデータの比較には、ファイル サイズ、ファイル タイプ、所有者、およびパーミッションが含まれます。

詳細については [pxar メタデータアーカイブフォーマット](#) を参照してください。ファイルの ctime と inode 番号は保存されません。

ファイル ctime と inode 番号は保存されず、比較に使用されません。

に同期させる場合があるからです。このような場合

ctime と inode 番号は常に変更されます。

このモードでは、前回のバックアップスナップショットから未変更のファイルコンテンツのチャンクを再利用することで、可能な限りファイルコンテンツの読み込みと再チャンクを回避します。

メタデータを比較するために、以前のスナップショット **mpxar** メタデータアーカイブがバックアップ実行の開始時にダウンロードされ、参照として使用されます。さらに、ペイロードアーカイブ **ppxar** のインデックスが取得され、ファイルコンテンツチャンクのダイジェストを検索するために使用されます。

のインデックスを再作成するために使用されます。

バックアップ中、メタデータアーカイブとペイロードアーカイブはデータモードと同じ方法でエンコードされますが、メタデータモードでは、各エントリがメタデータの参照アーカイブで検索され、最初に比較されます。リファレンスと比較してファイルに変更がなければ

ファイルは変更されていないとみなされ、Proxmoxバックアップクライアントは先読みキャッシュモードに入ります。このモードでは、クライアントはファイルシステム内の次のエントリが再利用可能である限り、読み取りと比較を続けます。さらに、これらのファイルコンテンツが格納されているペイロードアーカイブのオフセット範囲を追跡します。ファイル境界はチャンク境界に揃える必要がないため、無条件に再利用されるチャンクには無駄なチャンクコンテンツ（パディングとも呼ばれる）が含まれる可能性があるため、追加のルックアヘッドキャッシングが必要です。

ルックアヘッド・キャッシュは、キャッシュ・サイズの上限に達するか、メタデータが変更されたファイル・エントリに遭遇するか、再利用が考慮されるペイロード・チャンクの範囲が連続的でなくなる時点まで、すべての変更されていないエントリを貪欲にキャッシュします。後者の例としては、後続のバックアップ実行の間にファイルが消失し、範囲に穴が空いたことが挙げられます。この時点でキャッシュモードは無効になり、クライアントはこの時点までにキャッシュされたすべての未変更ファイルのペイロードチャンクを再利用することで発生する無駄なパディングサイズを計算します。パディングが許容範囲内（実際に再利用されるチャンク・コンテンツの10%というプリセットの制限値以下）であれば、ファイルは、更新されたオフセット参照を使用してメタデータ・アーカイブにエンコードされて再利用されます。

ントへの更新されたオフセット参照を使用してメタデータアーカイブでエンコードし、新しい **ppxar** アーカイブで既存のチャンクのインデックスを再作成することで再利用されます。しかし、パディングがを超えた場合、すべてのキャッシュされたエントリは再エンコードされます。

既存のデータ。キャッシュされたメタデータは、キャッシュされたファイルのコンテンツが再利用されようが再エンコードされようが、メタデータアーカイブにエンコードされます。

ルックアヘッド・キャッシングと、内容が変更されていないファイルに対する既存のペイロード・アーカイブ・チャンクの再利用を組み合わせることで、可能な限りファイル内容の再読み込みや再チャンクを回避し、バックアップ処理を高速化します。

パディングを減らし、チャンクの再利用性を高めるため、データモードとメタデータモードでアーカイブを作成する際、**pxar** エンコーダはスライディングウィンドウ・チャンカーにチャンクの境界候補としてファイルの境界を通知します。チャンカーは内部状態に基づいて、次のように決定します。

に基づいて決定します。

16.4 警告と制限

16.4.1 ハッシュの衝突に関する注意

すべてのハッシュアルゴリズムには衝突が発生する可能性があります。つまり、2つ以上の入力が同じチェックサムを生成するということです。SHA-256の場合、この可能性はごくわずかです。このような衝突の可能性を計算するには、確率論の「誕生日問題」の考え方を使うことができます。大きな数の場合、これは実際には通常のコンピュータでは計算不可能ですが、良い近似があります：

$$p(n, d) = 1 - e^{-n^{\frac{1}{2d}}}$$

ここで、nは試行回数、dは可能性の数です。具体的な例として、1PiBの大きなデータストアと4MiBの平均チャンクサイズを仮定します。つまり、 $n = 268435456$ 回のトライ、 $d = 2^{256}$ 回の可能性ということになります。これらの値を先ほどの式に挿入すると、このシナリオでの衝突の確率がわかります：

$$3.1115 \times 10^{-61}$$

例えば、45個の数字の中から6個の数字を当てる宝くじゲームでは、6個の数字すべてを正しく当てる確率はわずか 1.2277×10^{-7} です。つまり、衝突の可能性は、このような宝くじゲームに8回連続で当選するよりも低いということです： $(1.2277 \times 10^{-7})^8 = 5.1623 \times 10^{-56}$ 。

結論として、このような衝突が通常のデータストアで偶然発生する可能性は極めて低い。

さらに、SHA-256は長さ拡張攻撃を受けやすいですが、チャンクの大きさには上限があるため、潜在的な攻撃者がその上限を超えて任意にデータにコンテンツを追加することはできないため、これは問題ではありません。

16.4.2 ファイルベースのバックアップ

動的なサイズのチャンク（ファイルベースのバックアップの場合）は、ファイルに直接ではなく、カスタムアーカイブ形式（pxar）で作成されるため、ファイルとチャンクの間に関係はありません。このため、Proxmox Backup Clientはバックアップのたびにすべてのファイルを再度読み込む必要があります。そうしないと元のチャンクを再利用できる一貫性のある独立したpxarアーカイブを生成できません。ただし、新しいチャンクや変更されたチャンクのみがアップロードされます。

このような制限を回避するために、変更検出モードメタデータが導入されました。

16.4.3 暗号化されたチャンクの検証

暗号化されたチャンクの場合、元の（平文の）データのチェックサムしか利用できないため、（暗号化キーを持たない）サーバーがデータの内容を照合することは不可能です。代わりに CRC-32 チェックサムだけがチェックされます。

16.5 トラブルシューティング

インデックスファイル(.fidx, .didx)には、ファイルの再構築方法に関する情報が含まれています。より正確には、元のファイルが分割されたチャンクへの参照の順序付けられたリストが含まれています。スナップショットに何か問題がある場合、そのスナップショットで参照されているチャンクを見つけ、それらが存在し、無傷であるかどうかをチェックすることが有用な場合があります。

してください。`proxmox-backup-debug`コマンドラインツールを使うことができます。

使うことができます。例えば、参照されているチャンクのリストを得るには

`.fidx`インデックスの

```
# proxmox-backup-debug inspect file drive-scsi0.img.fidx
```

同じコマンドを `.blob` ファイルの検査にも使用できます。`-decode` パラメータを指定しないと、サイズと暗号化タイプがあればそれだけが表示されます。`-decode` が設定されている場合、`.blob` ファイルは指定されたファイルにデコードされます（'-'は標準出力に直接デコードします）。

次の例では、`qemu-server.conf.blob` のデコードされた内容が表示されます。検査しようとしているファイルが暗号化されている場合、`--keyfile` を使用して鍵ファイルへのパスを提供する必要があります。

```
# proxmox-backup-debug inspect file qemu-server.conf.blob --decode -.
```

また、特定のチャンクファイルがどのインデックスファイルを参照しているかも確認できます：

```
# proxmox-backup-debug inspect chunk_
→b531d3ff9bd7c65748a61198c060678326a431db7edeb874c327b7986e595e0 --reference-filter /path/
→in/a/datastore/directory
```

ここで **--reference-filter** はインデックスファイルの検索先を指定します。これは任意のパスにすることができます。何らかの理由でチャンクのファイル名が変更された場合、**--digest** を使って明示的にダイジェストを指定することができます。デフォルトでは、チャンクのファイル名が検索するダイジェストとして使われます。もし **--reference-filter** が指定されていない場合は、チャンクのCRCと暗号化の状態のみが表示されます。

また、**--decode** フラグを指定することで、チャンクをデコードすることもできます。チャンクが暗号化されている場合は、**--keyfile** を提供する必要があります。

16.5.1 Proxmox Backup Serverが起動していない状態でのリストア

実行中のProxmox Backup Serverインスタンスがなくても、**recover**サブコマンドを使用してスナップショットから特定のファイルをリストアできます。バックアップが暗号化されている場合は、対応するキーファイルも必要です。

```
# proxmox-backup-debug recover index drive-scsi0.img.fidx /path/to/.chunks
```

上記の例では、*/path/to/.chunks*引数はチャンクを含むディレクトリへのパスで、*drive-scsi0.img.fidx*はリストアしたいファイルのインデックスファイルです。どちらのパスも

は絶対または相対です。**-skip-crc**を指定すると、チャンクのCRCチェックを無効にすることができます。これにより、処理が若干高速化され、（部分的に）破損したチャンクの復元を試みることができます。これはスキップ-CRCオプションなしで試すことを推奨します。

よくある質問

17.1 Proxmox Backup Server (PBS)はどのディストリビューションをベースにしていますか？

Proxmox Backup ServerはDebian GNU/Linuxをベースにしています。

17.2 バックアップソース(クライアント)としてサポートされているプラットフォームは？

クライアントツールはほとんどの最新のLinuxシステムで動作します。つまり、Debianベースのバックアップに限定されません。

17.3 Proxmox Backup Serverは32ビットプロセッサで動作しますか？

Proxmox Backup Serverは64ビットCPU（AMDまたはIntel）のみをサポートしています。将来的に32ビットプロセッサをサポートする予定はありません。

17.4 Proxmox Backup Serverバージョンのサポート期間は？

Proxmox Backup バージョン	Debian バージョン	最初の 再リリース	Debian EOL	Proxmox バックアップ終了
Proxmox バックアップ 4	Debian 13 (Trixie)	2025-08	未定	未定
Proxmox バックアップ 3	Debian 12 (Bookworm)	2023-06	2026-08	2026-08
Proxmox バックアップ 2	Debian 11 (Bullseye)	2021-07	2024-07	2024-07
Proxmox バックアップ 1	Debian 10 (Buster)	2020-11	2022-08	2022-07

17.5 Proxmox Backup Serverを次のポイントリリースにアップグレードするにはどうすればいいですか？

マイナーバージョンアップ、例えばProxmox Backup Serverのバージョン3.1から3.2や3.3へのアップグレードは、通常のアップデートと同様に行うことができます。ただし、[リリースノート](#)で関連する重要な変更、または破壊的な変更を確認してください。

アップデート自体には、Web UIのノード-> アップデートパネルまたはCLIを使用します：

```
apt update  
apt full-upgrade
```

i 注意

常にパッケージリポジトリを正しくセットアップし、`apt update`でエラーが出なかった場合のみ、実際のアップグレードを続行してください。

17.6 Proxmox Backup Serverを次のメジャーリリースバージョンにアップグレードする方法を教えてください。

Proxmox Backup Server 2.4から3.1へのメジャーバージョンアップもスーパーサポートされています。アップグレードは慎重に計画、テストする必要があります、重要なバックアップのオフサイトコピー(リモート同期やテープなど)を準備せずに開始しないでください。

アップグレードの具体的な手順はそれぞれのセットアップによって異なりますが、ここでは一般的な手順とアップグレードの方法について説明します：

- Proxmox Backup Server 3から4へのアップグレード
- Proxmox Backup Server 2から3へのアップグレード
- Proxmox Backup Server 1から2へのアップグレード

17.7 データストアを別の場所にコピーまたは同期できますか？

Proxmox Backup Serverでは、リモートと同期ジョブを使用して、データストアを他の場所にコピーまたは同期することができます。リモートとは、ローカルストアと同期可能なデータストアを持つ別のサーバを指します。同期ジョブは、データストアの内容をリモートからローカルのデータストアにプルするために使用されるプロセスです。

17.8 Proxmox Backup Serverはバックアップアーカイブのデータ整合性を検証できますか？

Proxmox Backup Serverは、データの整合性を保証するために、組み込みのSHA-256チェックサムアルゴリズムを使用しています。各バックアップには、マニフェストファイル(index.json)が作成され、すべてのバックアップファイルのリストとサイズ、チェックサムが含まれます。このマニフェストファイルは、各バックアップの整合性を検証するために使用されます。

17.9 リモートサーバにバックアップする場合、リモートサーバを信頼する必要がありますか？

Proxmox Backup ServerはトランSPORTレイヤセキュリティ(TLS)を介してデータを転送し、さらにクライアント側の暗号化をサポートしています。これは、データが安全に転送され、サーバに到達する前に暗号化できることを意味します。したがって、攻撃者がサーバやネットワークの任意のポイントにアクセスした場合でも、データを読み取ることはできません。

i 注意

暗号化はデフォルトでは有効になっていません。暗号化を設定するには、[バックアップクライアントの暗号化のセクション](#)を参照してください。

17.10 バックアップは増分/重複/フルですか？

Proxmox Backup Serverでは、バックアップはサーバに増分的に送信され、データはサーバ上で重複排除されます。これにより、消費されるストレージとネットワークへの影響を最小限に抑えます。各バックアップはすべてのデータを参照し、フルバックアップとなります。詳細については、[技術概要](#)を参照してください。

コマンド構文

注記

コマンドラインツールのログの冗長性は、PBS_LOG (pxarの場合はPXAR_LOG) 環境変数で制御できます。設定可能な値は off、error、warn、info、debug、trace で、info がデフォルトです。

A.1 proxmox-backup-client

`proxmox-backup-client backup {<backupspec>} [OPTIONS]` バックアップを作成します。

(ホストの)バックアップを作成します。

<backupspec>

[<string>] バックアップ元指定のリストです：

[<archive-name>.<type>:<source-path>] ...

archive-name」には英数字、ハイフン、アンダースコアのみを、「type」には「pxar」、「img」、「conf」、「log」のいずれかを指定します。複数回指定することもできます。

オプションのパラメータ：

`--all-file-systems <boolean> (default=false)`

マウントされているすべてのサブディレクトリを含めます。

`-バッックアップID <文字列>`

バックアップID。

`--backup-time <integer> (1 - N)`

バックアップ時間 (Unixエポック)。

`--バッックアップタイプ vm|ct|host`

バックアップタイプ。

`-burst <文字列>`

トークン・バケットのサイズ (トークン・バケット・フィルタの場合)。バイト単位で、単位は任意です (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, Gib, ...)。

`--変更検出モード legacy|data|metadata (default=legacy)`

前回のバックアップ実行時からのファイルの変更を検出するモード

`--chunk-size <integer> (64 - 4096) (default=4096)`

チャンクサイズを KB 単位で指定。2 のべき乗でなければなりません。

`--暗号化モード none|encrypt|sign-only (default=encrypt)`

データを暗号化 (AEAD暗号を使用) するか、署名のみを行うか、あるいはどちらも行わないかを定義します。

`--dry-run <boolean> (default=false)`

バックアップの実行内容を表示するだけで、何もアップロードしません。

```
--entries-max <integer> (default=1048576)
メモリに保持するエントリの最大数。

--exclude <文字列>
除外するファイルのパスまたはパターンのリスト。複数指定可能。

--include-dev <文字列>
指定したファイルと同じ st_dev 番号 (man fstat を参照) のマウントポイントを含めます。複数回指定できます。

--keyfd <integer> (0 - N)
既にオープンされているファイル記述子を介して暗号化キーを渡します。

--keyfile <文字列>
暗号化キーへのパス。すべてのデータはこの鍵を使って暗号化されます。

--master-pubkey-fd <integer> (0 - N)
既にオープンされているファイル記述子を介してマスター公開鍵を渡します。

--master-pubkey-file <文字列>
マスター公開鍵へのパス。バックアップに使用される暗号化キーは、このキーを使用して暗号化され、バックアップに追加されます。

--no-cache <boolean> (default=false)
ネットワークストレージのローカルデータストアキャッシュをバイパスします。

--ns <文字列>
名前空間を指定します。

--rate <文字列>
レート制限 (トーケンバケットフィルタの場合) をバイト/秒単位で指定します。単位は任意です (B、KB (base 10)、MB
、GB、...、KiB (base 2)、MiB、GiB、...)。

--リポジトリ <文字列>
リポジトリの URL。
```

proxmox-backup-client benchmark [OPTIONS] ベンチマークを実行します。

ベンチマークテストを実行します

:

```
--keyfile <文字列>
暗号化キーへのパス。すべてのデータはこのキーを使って暗号化されます。

--no-cache <boolean> (default=false)
ネットワーク・ストレージのローカル・データストア・キャッシュをバイパスします。

--出力形式 text|json|json-pretty
出力フォーマット

--リポジトリ <文字列>
リポジトリのURL。
```

proxmox-backup-client catalog dump <snapshot> [オプション]。

カタログをダンプします。

<snapshot>

[<string>] スナップショットパス。

オプションのパラメータ：

--keyfd <integer> (0 - N)

すでに開いているファイル記述子を介して暗号化キーを渡します。

--keyfile <文字列>

暗号化キーへのパス。

--ns <文字列>

名前空間。

--リポジトリ <文字列>

リポジトリのURL。

proxmox-backup-client catalog shell <スナップショット><アーカイブ名> [オプション].

スナップショットの検査と復元を対話的に行うシェルです。

<スナップショット>

[<string>] グループ/スナップショットパス。

<archive-name>

[<string>] バックアップアーカイブ名。

オプションのパラメータです：

--keyfd <integer> (0 - N)

既にオープンされているファイル記述子を介して暗号化キーを渡します。

--keyfile <文字列>

暗号化キーへのパス。

--ns <文字列>

名前空間を指定します。

--リポジトリ <文字列>

リポジトリの URL。

proxmox-backup-client change-owner <group> <new-owner> [OPTIONS] バックアップグループの所有者を変更します。

バックアップグループの所有者を変更します

<group>

[<string>] バックアップグループ。

<new-owner>

[<string>] 認証ID

オプションのパラメータです：

--ns <文字列>

名前空間。

--リポジトリ <文字列>

リポジトリのURL。

proxmox-backup-client garbage-collect [OPTIONS] ガベージコレクションを開始します。

特定のリポジトリに対してガベージコレクションを開始し

ます。オプションのパラメータです：

--出力形式 text|json|json-pretty

出力形式を指定します。

--リポジトリ <文字列

リポジトリのURL。

proxmox-backup-client group forget <group> [オプション]。

バックアップスナップショットを忘却(削除)します。

<グループ>

[<string>] バックアップグループ

オプションのパラメータ

--ns <文字列>

名前空間を指定します。

--リポジトリ <文字列>

リポジトリのURL。

proxmox-backup-client help [{<コマンド>}] [オプション]。

指定したコマンド（またはサブコマンド）のヘルプを表示します。

<コマンド>

[<string>] コマンド。ネストされたサブコマンドを指定するためにリストにすることができます。複数指定可能。

オプションのパラメータ：

--verbose <ブール値>

詳細なヘルプを表示します。

proxmox-backup-client key change-passphrase [<path>] [OPTIONS] 暗号化キーのパスワードを変更します。

暗号化キーのパスワードを変更します。

<path>

[<string>] 鍵ファイル。これを指定しないと、デフォルトのキーのパスワードが変更されます。

オプションのパラメータ：

--ヒント <文字列>

パスワードのヒント。

--kdf none|scrypt|pbkdf2 (default=scrypt)

パスワードで保護された暗号鍵用の鍵生成関数。

proxmox-backup-client key create [<path>] [OPTIONS] 新しい暗号鍵を作成します。

新しい暗号化キーを作成します。

<path>

[<string>] 出力ファイル。これを指定しないと、キーは新しいデフォルトの暗号化キーになります。

オプションのパラメータ：

--ヒント <文字列>

パスワードのヒント。

--kdf none|scrypt|pbkdf2 (default=scrypt)

パスワードで保護された暗号鍵のための鍵導出関数です。

proxmox-backup-client key create-master-key

対称バックアップ暗号化キーの暗号化バージョンを各バックアップと一緒にバックアップサーバに置くために使用されるRSA公開/秘密鍵ペアを作成します。

proxmox-backup-client key import-master-pubkey <パス>

対称バックアップ暗号化キーの暗号化バージョンを各バックアップと一緒にバックアップサーバに置くために使用されるRSA公開キーをインポートします。

インポートされた鍵は、同じローカルユーザによる今後の起動のためのデフォルトのマスターキーとして使用されます。

<path>

[<string>] PEM形式のRSA公開鍵へのパスです。

proxmox-backup-client key import-with-master-key [<path>] --encrypted-keyfile

--master-keyfile <string> [OPTIONS] 暗号化されたバックアップをインポートします。

◦

(プライベート)マスターキーを使用して暗号化された暗号化キーのバックアップをインポートします。

<path>

[<string>] 出力ファイル。これを指定しないと、暗号鍵は新しいデフォルトの暗号鍵になります。

--暗号化キーファイル <文字列>

インポートするRSA暗号化キーファイル。

-マスター鍵ファイル <文字列>

(秘密)マスターキーを使用します。

オプションのパラメータ：

--hint <文字列>

パスワードのヒント。

--kdf none|scrypt|pbkdf2 (default=scrypt)

パスワードで保護された暗号鍵のための鍵生成関数。

proxmox-backup-client key paperkey [<path>] [OPTIONS]

暗号化キーを含む印刷可能で人間が読めるテキストファイルを生成します。これには高速な鍵の復元用にスキャン可能なQRコードも含まれます。

<path>

[<string>] 鍵ファイル。これを指定しないと、デフォルトのキーが使用されます。

オプションのパラメータ：

--出力形式 text|html

Paperkeyの出力形式

--件名 <文字列>

指定した件名をタイトルテキストとして含めます。

proxmox-backup-client key show [<path>] [OPTIONS]

暗号化キーのメタデータを表示します。

<path>

[<string>] 鍵ファイル。これを指定しないと、デフォルトのキーのメタデータが表示されます。

オプションのパラメータ：

--出力形式 text|json|json-pretty
出力形式。

proxmox-backup-client key show-master-pubkey [<path>] [OPTIONS] マスターキーの情報を一覧表示します。

マスターキーに関する情報を一覧表示します。

<path>
[<string>] PEM フォーマットの RSA 公開鍵へのパス。指定しない場合はデフォルトの場所が使用されます。

オプションのパラメータ：

--出力形式 text|json|json-pretty
出力形式。

proxmox-backup-client list [OPTIONS] バックアップグループをリストします。

バックアップグループをリ

ストします。

オプションのパラメータ

です：

--ns <文字列
名前空間を指定します。

--output-format text|json|json-pretty
出力形式を指定します。

--リポジトリ <文字列
リポジトリのURL。

proxmox-backup-clientログイン [OPTIONS].

ログインを試みます。成功したらチケットを保

存します。オプションのパラメータ

--リポジトリ <文字列
リポジトリ URL。

proxmox-backup-client logout [OPTIONS] ログアウトします。

ログアウト (保存されたチケットを削除)

します。オプションのパラメータです

:

--リポジトリ <文字列
リポジトリのURLです。

proxmox-backup-client map <snapshot> <archive-name> [OPTIONS] マップします。

VM バックアップからローカルループバックデバイスにライブイメージをマップします。元に戻すには 'unmap' を使用します。警告: 信頼されたバックアップでのみ実行してください！

<スナップショット

A.1. proxmox-backup-client

[<string>] グループ/スナップショットパス。

<アーカイブ名> [<文字列>

[<string>] バックアップアーカイブ名です。

オプションのパラメータです：

--keyfile <文字列>

暗号化キーへのパス。

--ns <文字列>

名前空間。

--リポジトリ <文字列>

リポジトリのURL。

--verbose <boolean> (default=false)

冗長出力でフォアグラウンドに留まります。

proxmox-backup-client mount <スナップショット> <アーカイブ名> <ターゲット> [OPTIONS].

pxar アーカイブをマウントします。

<スナップショット>

[<string>] グループ/スナップショットのパス。

<アーカイブ名> [<文字列>]

[<string>] バックアップアーカイブ名。

<ターゲット>

[<string>] ターゲットディレクトリのパス。

オプションのパラメータです：

--keyfile <文字列>

暗号化キーへのパス。

--ns <文字列>

名前空間。

--リポジトリ <文字列>

リポジトリのURL。

--verbose <boolean> (default=false)

冗長出力をを行い、フォアグラウンドに留まります。

proxmox-backup-client namespace create [<ns>] [OPTIONS] 新しい名前空間を作成します。

新しい名前空間を作成します。

<ns>

[<string>] 名前空間。

オプションのパラメータです：

--リポジトリ <文字列>

リポジトリのURL。

proxmox-backup-client namespace delete [<ns>] [OPTIONS] オプションを指定します。

既存のネームスペースを削除します。

<ns>

[<string>] 名前空間。

オプションのパラメータです：

--delete-groups <論理値>

階層内のすべてのグループを削除します。

--リポジトリ <文字列
リポジトリのURL。

proxmox-backup-client namespace list [<ns>] [OPTIONS] リポジトリ内の名前空間を一覧表示します。

リポジトリ内の名前空間を一覧表示します。

<ns>

[<string>] 名前空間。

オプションのパラメータです：

--max-depth <integer> (0 - N)
最大再帰深度

--出力形式 text|json|json-pretty
出力形式。

--リポジトリ <文字列
リポジトリの URL。

proxmox-backup-client prune <group> [OPTIONS] バックアップリポジトリを削除します。

バックアップリポジトリをプルーンします。

<group>

[<string>] バックアップグループ

オプションのパラメータです：

--dry-run <ブール値
pruneが何をするのかを示すだけで、何も削除しません。

--keep-daily <整数> (1 - N)
毎日保持するバックアップの数。

--keep-hourly <整数> (1 - N)
1時間ごとに保持するバックアップの数。

--keep-last <integer> (1 - N)
保持するバックアップの数。

--keep-monthly <整数> (1 - N)
毎月保持するバックアップの数。

--keep-weekly <整数> (1 - N)
毎週保持するバックアップの数。

--keep-yearly <整数> (1 - N)
保持する年間バックアップ数。

--max-depth <整数> (0 - 7)
何レベルの名前空間を操作するか (0== 再帰なし、空== 自動的な完全再帰、名前空間の深さは最大許容値を減らす)

--ns <文字列
名前空間。

--output-format text|json|json-pretty
出力形式。

--quiet <ブール値> (default=false)
最小限の出力 - リムーバルのみを表示します。

--リポジトリ <文字列
リポジトリのURL。

proxmox-backup-client restore <snapshot> <archive-name> <target> [OPTIONS] バックアップリポジトリをリストアします。

バックアップリポジトリをリストアします。

<スナップショット

[<string>] グループ/スナップショットパス。

<アーカイブ名> [<文字列>

[<string>] バックアップアーカイブ名。

<ターゲット>

[<string>] 対象ディレクトリのパス。標準出力に書き出す場合は'!'を使用します。

標準出力に書き出す際には、「.pxar」アーカイブは抽出しません。

オプションのパラメータ

--allow-existing-dirs <boolean> (default=false)

ディレクトリが既に存在する場合に失敗しないようにします。

-burst <文字列>

トークン・バケット（トークン・バケット・フィルタ用）のサイズをバイト単位で指定します。

--暗号化モード none|encrypt|sign-only (default=encrypt)

データを暗号化（AEAD暗号を使用）するか、署名のみを行うか、あるいはどちらも行わないかを定義します。

--ignore-acls <boolean> (default=false)

acl 設定を無視します。

--ignore-extract-device-errors <boolean> (default=false) acl 設定を無視します。

デバイスノード抽出時に発生するエラーを無視します。

--ignore-ownership <boolean> (default=false)

所有者の設定を無視します (chown なし)

--ignore-permissions <boolean> (default=false)

パーミッションの設定を無視 (chmod なし)

--ignore-xattrs <boolean> (default=false)

xattr の設定を無視

--keyfd <整数> (0 - N)

既にオープンされているファイル記述子を介して暗号鍵を渡します。

--keyfile <文字列>

暗号化キーへのパス。すべてのデータはこの鍵を使って暗号化されます。

--ns <文字列>

名前空間。

--overwrite <boolean> (default=false)

既存のファイルを上書きします。

--overwrite-files <boolean> (default=false)

既存のファイルを上書き

--overwrite-hardlinks <boolean> (default=false)

既に存在するエントリをアーカイブのハードリンクで上書きします。

--overwrite-symlinks <boolean> (default=false)

既存のエントリをアーカイブのシンボリックリンクで上書き

--pattern <文字列>

リストアするファイルを制限するためのパスまたはマッチパターン。複数回指定できます。

--prelude-target <文字列>

prelude をリストアするパス (pxar v2 アーカイブのみ)。

--rate <文字列>

レート制限(トークンバケットフィルタの場合)。バイト/秒、単位は任意(B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)。

--リポジトリ <文字列>

リポジトリの URL。

proxmox-backup-client snapshot files <snapshot> [OPTIONS] スナップショットファイルを一覧表示します。

スナップショットファイルを一覧表示します。

<snapshot>

[<string>] スナップショットのパス。

オプションのパラメータです：

--ns <文字列>

名前空間。

--output-format text|json|json-pretty

出力形式を指定します。

--リポジトリ <文字列>

リポジトリの URL。

proxmox-backup-client snapshot forget <snapshot> [OPTIONS] バックアップスナップショットを削除します。

バックアップスナップショットを忘却(削除)します。

<スナップショット>

[<string>] スナップショットのパス。

オプションのパラメータです：

--ns <文字列>

ネームスペース。

--リポジトリ <文字列>

リポジトリの URL。

proxmox-backup-client snapshot list [<group>] [OPTIONS] バックアップスナップショットを一覧表示します。

バックアップスナップショットを一覧表示します。

<group> [<string>] バックアップスナップショットを一覧表示します。

[<string>] バックアップグループを表示します。

オプションのパラメータです：

--ns <文字列>

名前空間を指定します。

--output-format text|json|json-pretty

出力形式を指定します。

--リポジトリ <文字列>

リポジトリの URL。

proxmox-backup-client snapshot notes show <snapshot> [オプション].

ノートを表示します。

<スナップショット

[<string>] スナップショットパス。

オプションのパラメータです：

--ns <文字列

名前空間。

--output-format text|json|json-pretty

出力形式を指定します。

--リポジトリ <文字列

リポジトリのURL。

proxmox-backup-client snapshot notes update <snapshot> <notes> [OPTIONS] アップデートノート

アップデートノート

<スナップショット

[<string>] スナップショットのパスです。

<notes>

[<string>] ノート。

オプションのパラメータです：

--ns <文字列

ネームスペース。

--リポジトリ <文字列

リポジトリURL。

proxmox-backup-client snapshot protected show <snapshot> [OPTIONS] 指定したスナップショットの保護状態を表示します

。

指定したスナップショットの保護状態を表示します。

<スナップショット

[<string>] スナップショットパス。

オプションのパラメータです：

--ns <文字列

名前空間を指定します。

--output-format text|json|json-pretty

出力形式を指定します。

--リポジトリ <文字列

リポジトリのURL。

proxmox-backup-client snapshot protected update <snapshot> <protected> [OPTIONS] スナップショットの保護状態を更新します。

スナップショットの保護ステータスを更新します。

<スナップショット

[<string>] スナップショットのパス。

<protected>

[<boolean>] 保護ステータス。

オプションのパラメータ：

--ns <文字列

--リポジトリ <文字列
リポジトリのURL。

proxmox-backup-client snapshot upload-log <snapshot> <logfile> [OPTIONS] バックアップログファイルをアップロードします。

バックアップログファイルをアップロードします。

<スナップショット

[<string>] グループ/スナップショットのパス。

<logfile>

[<string>] アップロードするログファイルのパスです。

オプションのパラメータです：

--crypt-mode none|encrypt|sign-only (**default=encrypt**)

データを暗号化 (AEAD暗号を使用) するか、署名のみを行うか、あるいはどちらも行わないかを定義します。

--keyfd <整数> (0 - N)

既にオープンされているファイル記述子を介して暗号化キーを渡します。

--keyfile <文字列

暗号化キーへのパス。すべてのデータはこの鍵を使って暗号化されます。

--ns <文字列

名前空間を指定します。

--リポジトリ <文字列

リポジトリの URL。

proxmox-backup-client status [OPTIONS] リポジトリの状態を取得します。

リポジトリの状態を取得

します。

オプションのパラメータで

す：

--出力形式 text|json|json-pretty

出力形式を指定します。

--リポジトリ <文字列

リポジトリのURL。

proxmox-backup-client task list [OPTIONS] このリポジトリユーザーの実行中のサーバータスクを一覧表示します。

このリポジトリユーザの実行中のサーバタスクを一覧表示

します。オプションのパラメータです：

--all <ブール値

停止中のタスクもリストアップします。

--limit <integer> (1 - 1000) (**デフォルト=50**)

一覧表示するタスクの最大数を指定します。

--出力形式 text|json|json-pretty

出力形式を指定します。

--リポジトリ <文字列

リポジトリのURL。

タスクログを表示します。

<upid>

[<string>] ユニークなプロセス/タスク識別子

オプションのパラメータ

--リポジトリ <文字列>

リポジトリの URL。

proxmox-backup-client task stop <upid> [OPTIONS] 特定のタスクを停止します。

特定のタスクを停止します。

<upid>

[<string>] 一意なプロセス/タスク識別子

オプションのパラメータ：

--リポジトリ <文字列>

リポジトリの URL。

proxmox-backup-client unmap [<name>] ループデバイスのマップを解除します。

map' でマップされたループデバイスのマップを解除し、すべてのリソースを解放します。

<name>

[<string>] アーカイブ名、loopdev へのパス (/dev/loopX)、またはループデバイス番号。省略すると現在のマッピングを全てリストアップし、残ったインスタンスを強制的にクリーンアップします。

proxmox-backup-client version [OPTIONS] クライアントとオプションのサーバーのバージョンを表示します。

クライアントとオプションのサーバのバージョ

ンを表示 オプションのパラメータです：

--output-format text|json|json-pretty
出力形式を指定します。**--リポジトリ <string>**
リポジトリのURL

A.1.1 カタログ シェル コマンド

以下のコマンドは対話式リストシェルで利用できます：

proxmox-backup-client shell <スナップショット> <名前.pxar**cd [<パス>]**

現在の作業ディレクトリを新しいディレクトリに変更します

<path>

[<string>] ターゲットパス。

clear-selected

リストア対象として選択されているファイルのリストを消去します。

選択解除 <パス>

リストアするエントリの選択を解除します。

リストア用に選択されたエントリのリストにエントリが見つからなかった場合、これはエラーを返します。

<path>

[<string>] リストから削除するエントリのパス。

終了

シェルを終了します。

find <pattern> [オプション] 検索します。

与えられたマッチパターンにマッチするエントリをカタログから検索します。

<pattern>

[<string>] ファイル名にマッチするパスまたはマッチパターン。

オプションのパラメータ：

--select <boolean> (default=false)

一致するファイル名をリストアに追加します。

help [{<コマンド>}] [オプション]。

指定したコマンド（またはサブコマンド）に関するヘルプを表示します。

<コマンド

[<string>] コマンド。ネストしたサブコマンドを指定するために、リストにすることができます。複数回指定することもできます。

オプションのパラメータ：

--verbose <論理値

冗長なヘルプ。

list-selected [オプション]。

リストア用に現在選択されているエントリをリストします

。オプションのパラメータ：

--patterns <boolean> (default=false)

マッチするファイルの代わりにマッチパターンをリストします。

ls [<path>]

作業ディレクトリまたは指定されたパスの内容を一覧表示します。

<path>

[<string>] 対象のパス。

pwd

現在の作業ディレクトリをリストアップします。

restore <target> [OPTIONS] サブアーカイブをリストアします。

現在の作業ディレクトリで指定されたサブアーカイブをターゲットにリストアします。

パターンを指定することで、サブアーカイブのサブセットに限定して復元することができます。pattern が存在しないか空の場合、完全なアーカイブがターゲットにリストアされます。

<ターゲット

[<string>] ローカルファイルシステム上のリストア対象パス。

オプションのパラメータ：

--パターン <文字列

ファイル名にマッチするパスまたはマッチパターン。

restore-selected <ターゲット

選択されたエントリを指定されたターゲットパスにリストアします。タ

ーゲットはクライアントのファイルシステム上に存在してはいけません

。

<target>

[<string>] ローカルファイルシステム上のリストアターゲットパス。

選択 <パス

リストアするエントリを選択します。

エントリが既にリストに存在する場合や、無効なパスが指定された場合はエラーが返されます。

<path>

[<string>] 対象のパスを指定します。

stat <path>

指定したディレクトリエントリのメタデータを読み込みます。

データを pxar アーカイブから読み込む必要があるため、ネットワーク経由で読み込むことになり、コストがかかります

。

<path>

[<string>] ターゲットパス。

A.2 proxmox-backup-manager

proxmox-backup-manager acl list [OPTIONS] アクセスコントロールリストです。

アクセス制御リスト。

オプションのパラメータ

です：

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager acl update <path> <role> [OPTIONS] アクセスコントロールリストを出力します。

アクセス制御リスト（ACL）を更新します。

<path>

[<string>] アクセス制御のパス。

<role>

[<role>] [PRIVILEGES]の組み合わせでロールを表す列挙型。

特権はビットフラグとして実装されているため、各特権の一意な組み合わせは、この列挙定義で使用される單一の一意なu64値にマッピングされます。

オプションのパラメータ

--auth-id <文字列>

認証ID。

--delete <論理値>

権限を（追加する代わりに）削除します。

--digest <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは、同時修正を防ぐために使用できます。

--グループ <文字列>

グループIDを指定します。

--propagate <boolean> (default=true)

パーティションの伝播（継承）を許可します。

proxmox-backup-manager acme account deactivate <name> [OPTIONS] ACME アカウントを無効にします。

ACME アカウントを停止します。

<name>

[<string>] ACME アカウント名。

オプションのパラメータ：

--force <boolean> (default=false)

サーバがアカウントの停止を拒否した場合でも、アカウントデータを削除します。

proxmox-backup-manager acme account info <name> [OPTIONS] acmeアカウント情報を表示します。

acme アカウント情報を表示します。

<name>

[<string>] ACME アカウント名。

オプションのパラメータです：

--output-format text|json|json-pretty

出力形式。

proxmox-backup-manager acme account list [OPTIONS] acmeアカウントを一覧表示します。

acmeアカウントをリスト

します。

オプションのパラメータ

です：

--出力フォーマット text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager acme account register <name> <contact> [OPTIONS] アクメアカウントを登録します。

ACMEアカウントを登録します。

<アカウント名>

[<string>] ACMEアカウント名。

<contact>

[<string>] 電子メールアドレスのリスト。

オプションのパラメータ：

--ディレクトリ <文字列

ACMEディレクトリ。

proxmox-backup-manager acme account update <name> [OPTIONS] ACMEアカウントを更新します。

ACMEアカウントを更新します。

<name>

[<string>] ACMEアカウント名。

オプションのパラメータ：

--contact <文字列

メールアドレスのリスト。

proxmox-backup-manager acme cert order [OPTIONS] 新しいACME証明書を注文します。

新しいACME証明書を注文します。オプ

ションのパラメータです：

--force <boolean> (default=false)

証明書の有効期限が間近でなくても更新を強制します。

proxmox-backup-manager acme cert revoke [OPTIONS] 新しいACME証明書を注文します。

新しいACME証明書を注文します。

proxmox-backup-manager acme plugin add <type> <id> --api <string> --data

<string> [OPTIONS] アクメアカウント情報を表示します。

acme アカウント情報を表示します。

<type>

[<string>] ACMEチャレンジプラグインのタイプです。

<id>

[<string>] ACMEチャレンジプラグインID。

--api <文字列

DNS API プラグイン ID。

--data <文字列

プラグインデータを含むファイル。

オプションのパラメータです：

--disable <boolean> (default=false)

設定を無効にするフラグ。

--validation-delay <integer> (0 - 172800) (default=30)

検証を要求するまでの待ち時間を秒単位で指定します。DNSレコードのTTLが

長い場合に対応できます。

proxmox-backup-manager acme plugin config <id> [OPTIONS] アクメのアカウント情報を表示します。

acme アカウント情報を表示します。

<id>

[<string>] プラグインID

オプションのパラメータ：

--output-format text|json|json-pretty
出力フォーマット。

proxmox-backup-manager acme plugin list [OPTIONS] acmeプラグインの一覧を表示します。

acmeプラグインをリストア
ップします。

オプションのパラメータ
:

--出力フォーマット text|json|json-pretty
出力フォーマット。

proxmox-backup-manager acme plugin remove <id> アクメプラグインを削除します。

ACMEプラグインの設定を削除します。

<id>
[<string>] ACMEチャレンジプラグインID。

proxmox-backup-manager acme plugin set <id> [OPTIONS] ACMEプラグイン構成を更新します。

ACME プラグイン設定を更新します。

<id>
[<string>] ACME チャレンジプラグイン ID。

オプションのパラメータ :

--api <文字列>
DNS APIプラグインID

--data <文字列>
DNS プラグインのデータ (base64 エンコード、パディング付き)。

--delete disable|validation-delay
削除するプロパティのリスト。複数指定可能。

--digest <文字列>
同時更新から保護するためのダイジェスト。

--disable <boolean> (default=false)
コンフィグを無効にするフラグ。

--validation-delay <integer> (0 - 172800) (default=30)
検証を要求するまでの待ち時間を秒単位で指定します。DNSレコードのTTLが
長い場合に対応できます。

proxmox-backup-manager ad create <realm> --server1 <string> [OPTIONS] 新しいADレルムを作成します。

新しいADレルムを作成します。

<realm>
[<string>] レルム名です。

--サーバ1 <文字列>
ADサーバのアドレス

オプションのパラメータ

--base-dn <文字列
LDAP ドメイン

--bind-dn <文字列>
LDAP ドメイン

--capath <文字列>
サーバに使用する CA 証明書。パスはファイルまたはディレクトリを指定します。ファイルを指す場合、そのパスに格納されている PEM 形式の X.509 証明書が信頼できる証明書として追加されます。パスがディレクトリを指す場合、そのディレクトリはシステムのデフォルトの証明書ストア /etc/ssl/certs を置き換えます。

--コメント <文字列>
コメント。

--default <boolean> (default=false)
ログイン時に選択される レルムをデフォルトにしたい場合は true を指定します。

--filter <文字列>
ユーザ同期用のカスタム LDAP 検索フィルタ

--mode ldap|ldap+starttls|ldaps (default=ldap)
LDAP 接続タイプ

--パスワード <文字列>
AD バインドパスワード

--ポート <整数> (0 - 65535)
AD サーバーのポート番号

--サーバ2 <文字列>
フォールバック AD サーバのアドレス

--sync-attributes [<email=<string>>] [<firstname=<string>>] [<lastname=<string>>]
どの LDAP 属性をどの PBS ユーザーフィールドにマッピングするかを指定するための key=value ペアのカンマ区切りリスト。例えば、LDAP 属性の mail を PBS の email にマッピングするには、email=mail と記述します。

--sync-defaults-options [<enable-new=<1|0>>] [<remove-vanished=<string>>] と記述します。
同期デフォルトオプション

--user-classes [<string>, ...]
(default=inetorgperson,posixaccount, person, user)
ユーザ同期で許可される objectClass 値のカンマ区切りリスト。例えば、user-classes を person, user に設定すると、ユーザ同期は、objectClass: person または objectClass: user を持つすべての LDAP エンティティを考慮します。

--verify <boolean> (default=false)
サーバ証明書を検証

proxmox-backup-manager ad delete <realm> [OPTIONS] (オプション)

LDAP realm の設定を削除します

<realm>

[<string>] レルム名。

オプションのパラメータです：

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは同時修正を防ぐために使用できます。

proxmox-backup-manager ad list [OPTIONS] 設定されている AD レルムを一覧表示します。

設定されている AD レルム一覧を

表示します。 オプションパラメ

ータです：

```
--output-format text|json|json-pretty
    出力フォーマット。
```

proxmox-backup-manager ad show <realm> [オプション].

ADレルムの設定を表示

<realm>

[<string>] レルム名。

オプションのパラメータです：

```
--output-format text|json|json-pretty
    出力形式を指定します。
```

proxmox-backup-manager ad sync <realm> [OPTIONS] 指定されたLDAPレルムを同期します。

指定されたLDAPレルムを同期します。

<realm>

[<string>] 認証ドメインID

オプションのパラメータ

--dry-run <boolean> (default=false)
 設定された場合、何も作成/削除しません。

--enable-new <ブール値>
 新しく同期されたユーザを直ちに有効にします。

--remove-vanished [<acl>|エントリ|プロパティ, ...]。
 セミコロンで区切られた、ユーザ同期中にユーザまたはユーザが消滅したときに削除するもののリスト。aclはユーザが同期から戻されなかったときにACLを削除します。

proxmox-backup-manager ad update <realm> [OPTIONS] (プロックスモックスバックアップマネージャ広告更新 <realm> [オプション])

ADレルムの設定を更新します。

<realm>

[<string>] レルム名。

オプションのパラメータ：

--base-dn <文字列>
 LDAPドメイン

--bind-dn <文字列>
 LDAPドメイン

--capath <文字列>
 サーバに使用するCA証明書。パスはファイルまたはディレクトリを指定します。ファイルを指す場合、そのパスに格納されているPEM形式のX.509証明書が信頼できる証明書として追加されます。パスがディレクトリを指す場合、そのディレクトリはシステムのデフォルトの証明書ストア /etc/ssl/certs を置き換えます。

--コメント <文字列>
 コメント。

--default <boolean> (default=false)
 ログイン時に選択されるデフォルトのレルムにしたい場合はtrueを指定します。

```
--delete
server2|port|comment|default|verify|mode|bind-dn|password|filter|sync-defaults-optionsを削除します。
削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列>
現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため
に使用できます。

-フィルター <文字列>
ユーザ同期用のカスタムLDAP検索フィルタ

--mode ldap|ldap+starttls|ldaps (default=ldap)
LDAP接続タイプ

--パスワード <文字列>
ADバインドパスワード

--ポート <整数> (0 - 65535)
ADサーバーのポート番号

--サーバ1 <文字列>
ADサーバーのアドレス

-サーバー2 <文字列>
フォールバックADサーバーアドレス

-同期属性 [[email=<string>] [,firstname=<string>] [,lastname=<string>] ]。
どの LDAP 属性をどの PBS ユーザーフィールドにマッピングするかを指定するための key=value ペアのカンマ
区切りリスト。例えば、LDAP 属性の mail を PBS の email にマップするには、email=mail と記述します。

--sync-defaults-options [[enable-new=<1|0>] [,remove-vanished=<string>]] と記述します。
同期デフォルトオプション

--user-classes [<string>, ...]
(default=inetorgperson,posixaccount,person,user)
ユーザ同期に許可されるobjectClass値のカンマ区切りリスト。例えば、user-classesをperson,userに設定すると
、ユーザ同期は、objectClass: person または objectClass: user のすべてのLDAPエンティティを考慮します
。

--verify <boolean> (default=false)
サーバ証明書を検証します
```

```
proxmox-backup-manager cert info
```

ノード証明書情報を表示します。

```
proxmox-backup-manager cert update [OPTIONS] ノード証明書情報を表示します。
```

ノード証明書を更新し、必要なファイル/ディレクトリをすべて生成します。オプショ

ンのパラメータ

```
--force <ブール値>
新しいSSL証明書を強制的に生成します。
```

```
proxmox-backup-manager datastore create <name> <path> [OPTIONS] データストアを作成します。
```

新しいデータストア設定を作成します。

```
<name>
[<string>] データストア名です。
```

<path>
 [<string>] データストア ディレクトリへの絶対パス、またはリムーバブル データストア用のデバイス上の相対パスのいずれかを指定します。

オプションのパラメータです：

--backend <バックエンド設定
 データストアのバックエンド構成

-バッキングデバイス <文字列
 リムーバブル・データストアのファイルシステム・パーティションの UUID。

--コメント <文字列
 コメント。

--gc-schedule <カレンダーイベント
 指定したスケジュールでガベージコレクションジョブを実行します。

--keep-daily <整数> (1 - N)
 毎日保持するバックアップの数。

--keep-hourly <整数> (1 - N)
 保持する毎時バックアップ数。

--keep-last <integer> (1 - N)
 保持するバックアップの数。

--keep-monthly <整数> (1 - N)
 每月保持するバックアップの数。

--keep-weekly <整数> (1 - N)
 毎週保持するバックアップの数。

--keep-yearly <整数> (1 - N)
 保持する年間バックアップ数。

--maintenance-mode [type=<enum> [,message=<string>]] を指定します。
 メンテナンスマード、タイプは 'offline' または 'read-only' のいずれか、メッセージは "で囲んでください。

**--notification-mode legacy-sendmail|notification-system
(default=notification-system)**
 このデータストアの通知をどのように送信するかを設定します。
 通知をシステムの *sendmail* 実行ファイルを介して *notify-user* で設定されたユーザに送信します。
notification-system 通知システムに一致する通知イベントを発行します。

--notify [[gc=<enum>] [,prune=<enum>] [,sync=<enum>] [,verify=<enum>]] と指定します。
 enumには'always'、'never'、'error'のいずれかを指定します。

--notify-user <文字列
 ユーザID

--output-format text|json|json-pretty
 出力形式。

--overwrite-in-use <boolean> (default=false)
 使用中のマーカーを上書きします (S3でバックアップされたデータストアのみ)。

--prune-schedule <カレンダーイベント
 指定したスケジュールで剪定ジョブを実行します。

--reuse-datastore <boolean> (default=false)
 既存のデータストア・ディレクトリを再利用します。

--tuning [[chunk-order=<enum>] [,gc-atime-cutoff=<integer>] [,gc-atime-safety-check=<1|0>] [,gc-cache-capacity=<integer>] [,sync-level=<enum>]]。
 データストア・チューニング・オプション

--verify-new <ブール値>

有効にすると、すべての新しいバックアップは完了直後に検証されます。

proxmox-backup-manager datastore list [OPTIONS] データストアのリストです。

データストアのリストです

◦

オプションのパラメータ

です：

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager datastore mount <store> [OPTIONS] リムーバブルデータストアをマウントします。

リムーバブルデータストアをマウントします。

<store>

[<string>] データストア名。

オプションのパラメータ：

-digest <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合に変更を防止します。これは、同時変更を防ぐために使用できます。

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager datastore remove <name> [OPTIONS] データストア設定を削除します。

データストア設定を削除します。

<name>

[<string>] データストア名です。

オプションのパラメータ：

--destroy-data <boolean> (default=false)

データストアの基本コンテンツを削除します。

--digest <文字列>

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するためには使用できます。

--keep-job-configs <boolean> (default=false)

有効にすると、このデータストアに関連するジョブ設定を保持します。

proxmox-backup-manager データストア s3-refresh <ストア>

データストアの内容をS3からローカルキャッシュストアにリフレッシュします。

<store>

[<string>] データストア名。

proxmox-backup-manager datastore show <name> [OPTIONS] データストアの設定を表示します。

データストアの設定を表示します

<name>

[<string>] データストア名。

オプションのパラメータです：

--output-format text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager datastore unmount <store> [OPTIONS] リムーバブルデータストアをアンマウントします。

リムーバブルデータストアをアンマウントします。

<store>

[<string>] データストア名。

オプションのパラメータ：

-digest <文字列

現在の設定ファイルのSHA256ダイジェストが異なる場合に変更を防止します。これは同時修正を防ぐために使用できます。

proxmox-backup-manager datastore update <name> [OPTIONS] データストアを更新します。

データストアの構成を更新します。

<name>

[<string>] データストア名。

オプションのパラメータです：

--comment <文字列

コメント。

--delete

コメント|gc-schedule|prune-schedule|keep-last|keep-hourly|keep-daily|keep-weekly|keep-monthly
削除するプロパティのリスト。複数指定可能。

-digest <文字列

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時修正を防ぐために使用できます。

--gc-schedule <カレンダーイベント>

指定したスケジュールでガベージコレクションを実行します。

--keep-daily <integer> (1 - N)
毎日保持するバックアップの数。

--keep-hourly <整数> (1 - N)
保持する毎時バックアップ数。

--keep-last <整数> (1 - N)
保持するバックアップの数。

--keep-monthly <整数> (1 - N)
毎月保持するバックアップの数。

--keep-weekly <整数> (1 - N)
毎週保持するバックアップの数。

--keep-yearly <整数> (1 - N)
保持する年間バックアップ数。

--maintenance-mode [<type=>[, message=<string>]] を指定します。

メンテナンスマード、typeは'offline'または'read-only'、messageは""で囲んでください。

--notification-mode legacy-sendmail|notification-system

(default=notification-system)

このデータストアの通知をどのように送信するかを設定します。

システムの sendmail 実行ファイルを介して notify-user で設定されたユーザーへの通知。

notification-system は通知システムにマッチする通知イベントを発行します。

--notify [[gc=<enum>] [,prune=<enum>] [,sync=<enum>] [,verify=<enum>]] のいずれか。
enumには'always'、'never'、'error'のいずれかを指定します。

--notify-user <文字列
ユーザID

--output-format text|json|json-pretty
出力形式。

--prune-schedule <カレンダーイベント
指定したスケジュールで剪定ジョブを実行します。

--tuning [[chunk-order=<enum>] [,gc-atime-cutoff=<integer>] [,gc-atime-safety-check=<1|0>] [,gc-cache-capacity=<integer>] [,sync-level=<enum>]]。
データストア・チューニング・オプション

--verify-new <ブール値
有効にすると、すべての新規バックアップが完了直後に検証されます。

proxmox-backup-manager datastore uuid-mount <uuid> [OPTIONS] [オプション]。

UUID を指定してリムーバブルデータストアのマウントを試みます。

<uuid>
[<string>] マウントするデバイスの UUID
オプションのパラメータ

--出力形式 text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager disk fs create <name> --disk <string> [OPTIONS] ファイルシステムを未使用のデバイスに作成します。

未使用のディスクにファイルシステムを作成します。mnt/datastore/<name>の下にマウントされます。

<name>
[<string>] データストア名。
--disk <文字列
ブロックデバイス名 (/sys/block/<name>).
オプションのパラメータ：
--add-datastore <boolean> データストアを構成します。
ディレクトリを使用してデータストアを構成します。
--ファイルシステム ext4|xfs
ツールでサポートされているファイルシステムタイプです。

proxmox-backup-manager disk fs delete <name> ファイルシステムを削除します。

mnt/datastore/<name>以下にマウントされたファイルシステムを削除します。

<name>
[<string>] データストア名です。

proxmox-backup-manager disk fs list [OPTIONS] システム管理データストアのマウントを一覧表示します。

systemd データストアのマウントユニットをリ

ストします。オプションのパラメータです：

--output-format text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager disk initialize <disk> [OPTIONS] 空のディスクを GPT で初期化します。

空のディスクを GPT で初期化します。

<disk>

[<string>] ブロックデバイス名 (/sys/block/<name>).

オプションのパラメータ：

--uuid <文字列

GPTテーブルのUUIDです。

proxmox-backup-manager disk list [OPTIONS] ディスクリスト。

ローカルディスクのリスト

です。

オプションのパラメータ

です：

--出力フォーマット text|json|json-pretty

出力フォーマット。

proxmox-backup-manager disk smart-attributes <disk> [オプション] SMART属性を表示します。

SMART属性を表示します。

<disk>

[<string>] ブロックデバイス名 (/sys/block/<name>).

オプションのパラメータ：

--出力形式 text|json|json-pretty

出力形式。

proxmox-backup-manager disk wipe <disk>

ディスクをワイプ

<ディスク

[<string>] (パーティション)ブロックデバイス名(/sys/class/block/<name>).

proxmox-backup-manager disk zpool create <name> --devices [<string>, ...] --raidlevel。

--raidlevel single|mirror|raid10|raidz|raidz2|raidz3 [OPTIONS] です。

zfs プールを作成します。

<name>

[<string>] データストア名。

--devices [<string>, ...] ディスク名のリスト。

カンマ区切りのディスク名のリスト。

--raidlevel single|mirror|raid10|raidz|raidz2|raidz3

使用する ZFS RAID レベル。

オプションのパラメータです：

--add-データストア <ブール値>

zpoolを使用してデータストアを構成します。

```
--ashift <integer> (9 - 16) (default=12)
プール・セクタ・サイズの指数。

--compression gzip|lz4|lzjb|zle|zstd|on|off (default=On)
使用するZFS圧縮アルゴリズム。
```

proxmox-backup-managerディスクzpoolリスト[OPTIONS]。

ローカルzfsプール。

オプションのパラメータ
です：

```
--output-format text|json|json-pretty
出力フォーマットを指定します。
```

proxmox-backup-manager dns get [OPTIONS] DNS設定を読み込みます。

DNS設定を読み込みます：

```
--出力形式 text|json|json-pretty
出力形式を指定します。
```

proxmox-backup-manager dns set [OPTIONS] DNS設定を更新します。

DNS設定を更新します。

オプションパラメータで
す：

```
--delete dns1|dns2|dns3
削除するプロパティのリスト。複数回指定できます。
```

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合に変更を防止します。これは、同時変更を防止するためには使用できます。

```
--dns1 <文字列>
最初のネームサーバーのIPアドレス。
```

```
--dns2 <文字列>
2番目のネームサーバーIPアドレス。
```

```
--dns3 <文字列>
3番目のネームサーバーIPアドレス
```

```
--search <文字列>
ホスト名を検索するドメインを指定します。
```

proxmox-backup-manager garbage-collection list [OPTIONS] ガベージコレクションの一覧を表示します。

gcジョブのないデータストアを含むすべてのデータストアのガベージコレクションジョブステータスをリストし

ます。オプションのパラメータです：

```
--output-format text|json|json-pretty
出力形式。
```

proxmox-backup-manager garbage-collection start <store> [OPTIONS] ガベージコレクションを開始します。

<store>

[<string>] データストア名。

オプションのパラメータ：

--出力形式 text|json|json-pretty
出力形式。

proxmox-backup-manager garbage-collection status <store> [OPTIONS] 特定のデータストアのガベージコレクションステータスを表示します。

特定のデータストアのガベージコレクションステータスを表示します。

<store>

[<string>] データストア名。

オプションのパラメータです：

--出力形式 text|json|json-pretty
出力フォーマット。

proxmox-backup-manager help [<command>] [OPTIONS] 指定したコマンド(またはサブコマンド)に関するヘルプを取得します。

指定したコマンド(またはサブコマンド)に関するヘルプを取得します。

<コマンド>

[<string>] コマンド。ネストしたサブコマンドを指定するためにリストにすることができます。複数指定可能。

オプションのパラメータ：

--verbose <ブール値>
詳細なヘルプを表示します。

proxmox-backup-manager ldap create <realm> --base-dn <string> --server1

<string> --user-attr <string> [OPTIONS] オプションを指定します。

新しいLDAPレルムを作成します。

<realm>

[<string>] レルム名。

--base-dn <文字列>
LDAPドメイン

--server1 <文字列>
LDAPサーバのアドレス

--user-attr <文字列>

ユーザー名属性。useridをLDAP dnにマッピングするために使用します。

オプションのパラメータ：

--bind-dn <文字列>
LDAPドメイン

--capath <文字列>

サーバーに使用するCA証明書。パスはファイルまたはディレクトリを指します。ファイルを指す場合、そのパスに格納されているPEM形式のX.509証明書が信頼できる証明書として追加されます。パスがディレクトリを指す場合、そのディレクトリはシステムの既定の証明書ストア /etc/ssl/certs を置き換えます。

--コメント <文字列>
コメント。

```
--default <boolean> (default=false)
ログイン時に選択されるレルムをデフォルトにしたい場合はtrueを指定します。

--filter <文字列>
ユーザー同期用のカスタムLDAP検索フィルター

--mode ldap|ldap+starttls|ldaps (default=ldap)
LDAP接続タイプ

--password <文字列>
LDAPバインドパスワード

--port <整数> (0 - 65535)
ポート番号

--server2 <文字列>
フォールバックLDAPサーバーのアドレス

-同期属性 [[email=<string>] [,firstname=<string>] [,lastname=<string>] ]。
どの LDAP 属性をどの PBS ユーザーフィールドにマップするかを指定するための key=value ペアのカンマ区切りリスト。例えば、LDAP 属性の mail を PBS の email にマップするには、email=mail と記述します。

--sync-defaults-options [[enable-new=<1|0>] [,remove-vanished=<string>]] と記述します。
同期デフォルトオプション

--user-classes [<string>, ...]
(default=inetorgperson posixaccount person user)
ユーザ同期に許可されるobjectClass値のカンマ区切りリスト。例えば、user-classesをperson,userに設定すると、ユーザ同期は、objectClass: person または objectClass: user のすべてのLDAPエンティティを考慮します。

--verify <boolean> (default=false)
サーバ証明書を検証します。
```

proxmox-backup-manager ldap delete <realm> [OPTIONS] LDAPレルム設定を削除します。

LDAPレルム設定を削除します

<realm>

[<string>] レルム名。

オプションのパラメータ

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは同時修正を防ぐために使用できます。

proxmox-backup-manager ldap list [OPTIONS] 設定されているLDAPレルム一覧を表示します。

設定されているLDAPレルム一覧を表示

します。オプションパラメータです：

```
--出力フォーマット text|json|json-pretty
出力フォーマットを指定します。
```

proxmox-backup-manager ldap show <realm> [OPTIONS] LDAPレルムの設定を表示します。

LDAPレルムの設定を表示します

<realm>

[<string>] レルム名。

オプションのパラメータです：

--output-format text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager ldap sync <realm> [OPTIONS] 指定されたLDAPレルムを同期します。

指定されたLDAPレルムを同期します。

<realm>

[<string>] 認証ドメインID

オプションのパラメータ

--dry-run <boolean> (default=false)
設定されている場合、何も作成/削除しません。

--enable-new <ブール値
新しく同期されたユーザを直ちに有効にします。

--remove-vanished [<acl|entry|properties, ...> [<acl|entry|プロパティ, ...>]]
セミコロンで区切られた、ユーザ同期中にユーザまたはユーザが消滅したときに削除するもののリスト。aclはユーザが同期から戻されなかったときにACLを削除します。

proxmox-backup-manager ldap update <realm> [OPTIONS] LDAP レルム設定を更新します。

LDAP レルムの設定を更新します。

<realm>

[<string>] レルム名。

オプションのパラメータです：

--base-dn <文字列
LDAP ドメイン

--bind-dn <文字列
LDAP ドメイン

--capath <文字列
サーバに使用する CA 証明書。パスはファイルまたはディレクトリを指すことができます。ファイルを指す場合、そのパスに格納されている PEM 形式の X.509 証明書が信頼できる証明書として追加されます。パスがディレクトリを指す場合、そのディレクトリはシステムのデフォルトの証明書ストア /etc/ssl/certs を置き換えます。

--コメント <文字列
コメント。

--default <boolean> (default=false)
ログイン時に選択されるデフォルトのレルムにしたい場合はtrueを指定します。

--delete
server2|port|comment|default|verify|mode|bind-dn|password|filter|sync-defaults-options|を削除します。
削除するプロパティのリスト。複数回指定できます。

-ダイジェスト <文字列
現在の設定ファイルの SHA256 ダイジェストが異なる場合に変更を防止します。これは、同時変更を防止するために使用できます。

--フィルタ <文字列
ユーザー同期用のカスタムLDAP検索フィルター

```
--mode ldap|ldap+starttls|ldaps (default=ldap)
LDAP接続タイプ

--password <文字列>
LDAP/ BIND パスワード

--port <整数> (0 - 65535)
ポート番号

--server1 <文字列>
LDAPサーバーのアドレス

--server2 <文字列>
フォールバックLDAPサーバーのアドレス

-同期属性 [[email=<string>] [,firstname=<string>] [,lastname=<string>] ]。
どの LDAP 属性をどの PBS ユーザーフィールドにマップするかを指定するための key=value ペアのカンマ区切りリスト。例えば、LDAP 属性の mail を PBS の email にマップするには、email=mail と記述します。

--sync-defaults-options [[enable-new=<1|0>] [,remove-vanished=<string>]] と記述します。
同期デフォルトオプション

--user-attr <文字列>
ユーザー名属性。useridをLDAPのdnにマップするために使用します。

--user-classes [<string>, ...]
(default=inetorgperson posixaccount person user)
ユーザ同期に許可されるobjectClass値のカンマ区切りリスト。例えば、user-classesをperson,userに設定すると、ユーザ同期は、objectClass: person または objectClass: user のすべてのLDAPエンティティを考慮します。

--verify <boolean> (default=false)
サーバ証明書を検証
```

proxmox-backup-manager ネットワークの変更

保留中の設定変更を表示 (diff)

proxmox-backup-manager network create <iface> [OPTIONS] (日本語)

ネットワークインターフェースの設定を作成します。

<iface>
[<string>] ネットワークインターフェース名。

オプションのパラメータです：

--autostart <ブール値>
インターフェースを自動起動します。

-bond-primary<文字列>
ネットワークインターフェース名。

--bond_mode balance-rr|active-backup|balance-xor|broadcast|802.3ad|balance-tlb|balance-alb
Linux ボンドモード

--bond_xmit_hash_policy レイヤー2|レイヤー2+3|レイヤー3+4
LACP (802.3ad) 用ボンド送信ハッシュポリシー

--bridge_ports [<string>, ...] ネットワークデバイスのリスト。
カンマ区切りのネットワークデバイスのリスト。

```
--bridge_vlanan_aware <論理値>
  ブリッジ vlan サポートを有効にします。

-cidr <文字列>
  ネットマスク付きIPv4アドレス（CIDR表記）。

--cidr6 <文字列>
  ネットマスク付きIPv6アドレス（CIDR表記）。

--コメント <文字列>
  コメント（inet、複数行にまたがる可能性あり）

--コメント6 <文字列>
  コメント（inet5、複数行にまたがる可能性あり）

--gateway <文字列>
  IPv4アドレス。

--gateway6 <文字列>
  IPv6アドレス。

--method manual|static|dhcp|loopback
  インタフェース設定方法

--method6 manual|static|dhcp|loopback
  インタフェース設定方法

--mtu <整数> (46 - 65535) (default=1500)
  最大伝送単位。

--slaves [<文字列>, ...].
  カンマ区切りのネットワークデバイスのリスト。

--type loopback|eth|bridge|bond|vlan|alias|unknown
  ネットワークインターフェイスのタイプ

--vlan-id <整数>
  VLAN ID。
```

proxmox-backup-manager network list [OPTIONS] ネットワークリスト。

ネットワークデバイスの
リスト。

オプションのパラメータ

:

--出力形式 text|json|json-pretty
出力フォーマット。

ネットワークリロード

ネットワークの変更をリロードします。

proxmox-backup-manager network remove <iface> [OPTIONS] ネットワークインターフェースの設定を削除します。

ネットワークインターフェースの設定を削除します。

<iface>

[<string>] ネットワークインターフェース名。

オプションのパラメータです：

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは同時修正を防ぐために使用できます。

proxmox-backup-manager network revert

ネットワーク設定を元に戻します(`rm /etc/network/interfaces.new`)。

proxmox-backup-manager network update <iface> [OPTIONS] ネットワークインターフェースの設定を更新します。

ネットワークインターフェースの設定を更新します。

<iface>

[<string>] ネットワークインターフェース名。

オプションのパラメータです：

--autostart <布尔值>

インターフェースを自動起動します。

-bond ボンドプライマリ <文字列>

ネットワークインターフェイス名。

--bond_mode balance-rr|active-backup|balance-xor|broadcast|802.3ad|balance-tlb|balance-alb

Linux ボンドモード

--bond_xmit_hash_policy レイヤー2|レイヤー2+3|レイヤー3+4

LACP (802.3ad) 用ボンド送信ハッシュポリシー

--bridge_ports [<string>, ...] ネットワークデバイスのリスト。

カンマ区切りのネットワークデバイスのリスト。

--bridge_vlanan_aware <論理値>

ブリッジ vlan サポートを有効にします。

-cidr <文字列>

ネットマスク付きIPv4アドレス (CIDR表記)。

--cidr6 <文字列>

ネットマスク付きIPv6アドレス (CIDR表記)。

--コメント <文字列>

コメント (inet、複数行にまたがる可能性あり)

--コメント6 <文字列>

コメント (inet5、複数行にまたがる可能性あり)

--delete cidr|cidr6|gateway|gateway6|method|method6|comments|comments6|mtu|autostart|bridge_port

削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列>

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため使用できます。

--gateway <文字列>

IPv4アドレス。

--gateway6 <文字列>

IPv6アドレス。

--method manual|static|dhcp|loopback

インターフェース設定方法

```
--method6 manual|static|dhcp|loopback
  インタフェース設定方法

--mtu <整数> (46 - 65535) (default=1500)
  最大伝送単位。

--slaves [<文字列>, ...].
  カンマ区切りのネットワークデバイスのリスト。

--type loopback|eth|bridge|bond|vlan|alias|unknown
  ネットワークインターフェイスのタイプ

--vlan-id <整数>
  VLAN ID.

--vlan-raw-device <string> VLAN ID。
  ネットワークインターフェース名
```

proxmox-backup-manager node show [OPTIONS] ノードの設定を表示します。

ノードの設定を表示します：

```
--output-format text|json|json-pretty
  出力フォーマットを指定します。
```

proxmox-backup-manager node update [OPTIONS] ノードの設定を更新します。

ノードの設定を更新します：

```
--acme [account=<string>] このノードで使用する acme アカウントです。
  このノードで使用する acme アカウント。

--acmedomain0 [domain=<string> [,alias=<string>] [,plugin=<string>]] ACME ドメイン構成文字列。
  ACME ドメイン設定文字列

--acmedomain1 [ドメイン=<文字列> [,エイリアス=<文字列>] [,プラグイン=<文字列>]] ACME ドメイン設定文字列。
  ACME ドメイン設定文字列

--acmedomain2 [ドメイン=<文字列> [,エイリアス=<文字列>] [,プラグイン=<文字列>]] ACME ドメイン設定文字列
  ACME ドメイン設定文字列

--acmedomain3 [ドメイン=<文字列> [,エイリアス=<文字列>] [,プラグイン=<文字列>]] ACME ドメイン設定文字列
  ACME ドメイン設定文字列

--acmedomain4 [ドメイン=<文字列> [,エイリアス=<文字列>] [,プラグイン=<文字列>]] ACME ドメイン設定文字列
  ACME ドメイン設定文字列
```

```
--暗号化-tls-1.2 <文字列>
  プロキシが TLS<= 1.2 で使用する OpenSSL 暗号のリスト。
```

```
-暗号-tls-1.3 <文字列>
  プロキシが TLS 1.3 用に使う OpenSSL 暗号スイートリスト
```

```
--consent-text <文字列>
  同意のバナーテキスト
```

```
--default-lang ar|ca|da|de|en|es|eu|fa|fr|gl|he|hu|it|ja|kr|nb|nl|nn|pl|pt_BR|ru|sl|sv|tr|zh_CN|zh_TW
  Proxmoxで利用可能なすべての言語。pt_BR、zh_CN、および
  zh_TW は翻訳ファイルで同じ大文字と小文字を使います。
```

```
--delete
acme|acmedomain0|acmedomain1|acmedomain2|acmedomain3|acmedomain4|http-proxy|email-from|3|ciphers-tls-
1.2|default-lang|description|task-log-max-days|consent-text。
削除するプロパティのリスト。複数回指定できます。

--説明 <文字列>
コメント（複数行）。
```

-ダイジェスト <文字列>
同時更新を防ぐためのダイジェスト

```
--email-from <文字列>
メールアドレス。
```

```
--http-proxy [http://]<host>[:port]
HTTPプロキシの設定 [http://]<host>[:port]
```

```
-task-log-max-days<整数> (0 - N)
タスクログを保存する最大日数
```

proxmox-backup-manager 通知エンドポイント `gotify create <name> --server <string> --token <string> [オプション]`.

新しい gotify エンドポイントを追加します。

<name>
[<string>] ターゲットとマッチャーのスキーマ名を指定します。

--server <string>
Gotify サーバーの URL。

--トークン <文字列>
認証トークン オプション

のパラメータです：

--コメント <文字列>
コメント。

--disable <論理値>
このターゲットを無効にします。

--filter <文字列>
非推奨。

--origin user-created|builtin|modified-builtin
通知構成エントリのオリジン。

proxmox-backup-manager 通知エンドポイント `gotify delete <name>`

gotify エンドポイントを削除します。

<name>
[<string>] ターゲットとマッチャーの名前スキーマ

proxmox-backup-manager 通知エンドポイント `gotify list [OPTIONS]` すべてのエンドポイントをリストアップします。

全てのエンドポイントを
リストアップします。

オプションのパラメータ
です：

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager notification endpoint gotify show <name> [OPTIONS] 一つのエンドポイントを表示します。

単一のエンドポイントを表示します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

オプションのパラメータ

--出力形式 text|json|json-pretty

出力フォーマットを指定します。

proxmox-backup-manager 通知エンドポイント gotify update <name> [OPTIONS] オプションです。

gotify エンドポイントを更新します。

<name>

[<string>] ターゲットとマッチャーのスキーマ名

オプションのパラメータです：

--コメント <文字列

コメント

--delete comment|disable

削除するプロパティのリスト。複数回指定可能。

--digest <文字列

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため
に使用できます。

--disable <論理値

このターゲットを無効にします。

--server <文字列

GotifyサーバーのURLです。

--トークン <文字列

認証トークン

proxmox-backup-manager 通知エンドポイント sendmail create <name> [OPTIONS] [OPTIONS] を指定します
。

新しい sendmail エンドポイントを追加します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

オプションのパラメータ

--author <文字列

メールの作成者。デフォルトは 'Proxmox Backup Server (\$hostname)' です。

--コメント <文字列

コメント。

--disable <論理値

このターゲットを無効にします。

--filter <文字列

非推奨。

--送信元アドレス <文字列>

送信元アドレス。このパラメータが設定されていない場合、プラグインはnode.cfg (PBS)のemail-from設定にフォールバックします。このパラメータも設定されていない場合、プラグインのデフォルトは root@\$hostname (\$hostname はノードのホスト名) になります。

-mailto <文字列>

メールの送信先。複数指定可能。

-mailto-user <文字列>

メールを送信するユーザ。ユーザのメールアドレスはusers.cfgで検索されます。複数回指定できます。

--origin user-created|builtin|modified-builtin

通知設定エントリの発信元です。

proxmox-backup-manager notification endpoint sendmail delete <name>

sendmail エンドポイントを削除します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

proxmox-backup-manager notification endpoint sendmail list [OPTIONS] すべてのエンドポイントをリストします。

全てのエンドポイントを

リストアップします。

オプションのパラメータ

です：

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager notification endpoint sendmail show <name> [OPTIONS] 一つのエンドポイントを表示します。

単一のエンドポイントを表示します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

オプションのパラメータ

--output-format text|json|json-pretty

出力フォーマットを指定します。

proxmox-backup-manager notification endpoint sendmail update <name> [OPTIONS] sendmailのエンドポイントを更新します。

sendmail エンドポイントを更新します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

オプションのパラメータ

--author <文字列>

メールの作成者。デフォルトは 'Proxmox Backup Server (\$hostname)' です。

--コメント <文字列>

コメント。

--delete author|comment|disable|from-address|mailto|mailto-user

--digest <文字列>

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため使用できます。

--disable <論理値>

このターゲットを無効にします。

--送信元アドレス <文字列>

送信メールの差出人アドレス。このパラメータが設定されていない場合、プラグインはnode.cfg (PBS)のemail-from設定にフォールバックします。このパラメータも設定されていない場合、プラグインのデフォルトはroot@\$hostname (\$hostname はノードのホスト名)になります。

-mailto <文字列>

メールの送信先。複数指定可能。

--mailto-user <文字列>

メールを送信するユーザ。ユーザのメールアドレスはusers.cfgで検索されます。複数回指定できます。

proxmox-backup-manager notification endpoint smtp create <name> -from-address<string>。

--from-address <string> --server <string> [OPTIONS] を指定します。

新しいsmtpエンドポイントを追加します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

--from-address <文字列>

メールの差出人アドレス。SMTPリレーでは、なりすましを避けるためにこのアドレスがユーザのものであることを要求する場合があります。メールのFromヘッダは\$author <\$from-address>に設定されます。

--サーバ <文字列>

SMTPリレーのホスト名またはIP。

オプションのパラメータ

--author <文字列>

メールの作成者。デフォルトは 'Proxmox Backup Server (\$hostname)' です。

--コメント <文字列>

コメント。

--disable <論理値>

このターゲットを無効にします。

-mailto <文字列>

メールの送信先。複数指定可能。

--mailto-user <文字列>

メールを送信するユーザ。ユーザのメールアドレスはusers.cfgで検索されます。複数回指定できます。

--mode insecure|starttls|tls (default=tls)

接続セキュリティ

--origin user-created|builtin|modified-builtin

通知構成エントリのオリジン。

--パスワード <文字列>

SMTP認証パスワード

--ポート <整数> (0 - 65535)

接続するポート。設定されていない場合、使用されるポートのデフォルトは25 (secure)、465 (tls)、または587 (starttls)です。

--ユーザ名 <文字列>

認証時に使用するユーザ名。ユーザ名が設定されていない場合、認証は行われません。PLAIN および LOGIN 認証方法がサポートされています。

proxmox-backup-manager 通知エンドポイント smtp 削除 <名前>

smtp エンドポイントを削除します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

proxmox-backup-manager 通知エンドポイント smtp list [OPTIONS] すべてのエンドポイントを一覧表示します。

全てのエンドポイントを
リストアップします。

オプションのパラメータ

です：

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager notification endpoint smtp show <name> [OPTIONS] 一つのエンドポイントを表示します。

単一のエンドポイントを表示します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマです。

オプションのパラメータ

--出力形式 text|json|json-pretty

出力フォーマットを指定します。

proxmox-backup-manager 通知エンドポイント smtp update <name> [OPTIONS] オプションです。

smtp エンドポイントを更新します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

オプションのパラメータです：

--author <文字列>

メールの作成者。デフォルトは 'Proxmox Backup Server (\$hostname)' です。

--コメント <文字列>

コメントを削除します。

--delete author|comment|disable|mailto|mailto-user|password|port|username

削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列>

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため
に使用できます。

--disable <論理値>

このターゲットを無効にします。

--フロムアドレス <文字列>

メールの差出人アドレス。SMTPリレーでは、なりすましを避けるためにこのアドレスがユーザのものであること

を要求する場合があります。メールの*From*ヘッダは\$author<\$from-address>に設定されます。

-mailto <文字列>

メールの送信先。複数指定可能。

-mailto-user <文字列>

メールを送信するユーザ。ユーザのメールアドレスはusers.cfgで検索されます。複数回指定できます。

--モード insecure|starttls|tls (default=tls)
接続セキュリティ

--パスワード <文字列>
SMTP認証パスワード

--ポート <整数> (0 - 65535)
接続先ポート。設定されていない場合、使用されるポートのデフォルトは25(安全でない)、465(tls)、または587(starttls)です。

--サーバ <文字列>
SMTPリレーのホスト名またはIP。

--ユーザー名 <文字列>
認証時に使用するユーザー名。ユーザー名が設定されていない場合、認証は行われません。PLAIN および LOGIN 認証方法がサポートされています。

proxmox-backup-manager notification endpoint webhook create <name> --method post|put|get --url <string> [OPTIONS] 新しいウェブフックエンドポイントを追加します。

新しいウェブフックエンドポイントを追加します。

<name>

[<string>] ターゲットとマッチャーのスキーマ名を指定します。

--method post|put|get (default=post)
使用する HTTP メソッド。

--url <文字列>
HTTP(s)URLとオプションのポート。

オプションのパラメータ：

--ボディ <文字列>
送信する HTTP ボディ。テンプレート化をサポートします。

--コメント <文字列>
コメント。

--disable <論理値>
このターゲットを無効にします。

--header [name=<string> [,value=<string>]] HTTPヘッダの配列。
HTTPヘッダの配列。各エントリは、名前と値を持つプロパティ文字列です。value プロパティは、base64 エンコーディングのヘッダを含みます。テンプレート化をサポートしています。複数回指定可能。

--origin user-created|builtin|modified-builtin
通知構成エントリのオリジン。

--secret [name=<string> [,value=<string>]] シークレットの配列です。
秘密の配列。各エントリは、名前とオプションの値を持つプロパティ文字列です。value プロパティには、base64 エンコーディングの秘密が含まれます。エンドポイント設定を返す API エンドポイントでは、シークレット名のみが返され、値は返されません。設定を更新する際には、保持したいすべてのシークレットも送信します。テンプレートからアクセスできます。複数回指定できます。

proxmox-backup-manager 通知エンドポイント webhook delete <名前>

ウェブフックエンドポイントを削除します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

proxmox-backup-manager 通知エンドポイントウェブフックリスト [OPTIONS] すべてのエンドポイントをリストアップします。

全てのエンドポイントを

リストアップします。

オプションのパラメータ

です：

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager notification endpoint webhook show <name> [OPTIONS] 一つのエンドポイントを表示します。

単一のエンドポイントを表示します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマです。

オプションのパラメータ

--出力形式 text|json|json-pretty

出力フォーマットを指定します。

proxmox-backup-manager notification endpoint webhook update <name> [OPTIONS] ウェブフックエンドポイントを更新します。

ウェブフックエンドポイントを更新します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ。

オプションのパラメータ

--ボディ <文字列>

送信する HTTP ボディ。テンプレート化をサポートします。

--コメント <文字列>

コメントを削除します。

--delete comment|disable|header|body|secret,

削除するプロパティのリスト。複数指定可能。

--digest <文字列>

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため
に使用できます。

--disable <論理値>

このターゲットを無効にします。

--header [name=<string> [,value=<string>]] HTTPヘッダの配列。

HTTPヘッダの配列。各エントリは、名前と値を持つプロパティ文字列です。value プロパティは、base64 エン
コーディングのヘッダを含みます。テンプレート化をサポートしています。複数回指定できます。

-メソッド post|put|get (default=post)

使用するHTTPメソッド。

--secret [name=<string> [,value=<string>]] を指定します。

秘密の配列。各エントリは、名前とオプションの値を持つプロパティ文字列です。value プロパティには、
base64 エンコーディングの秘密が含まれます。エンドポイント設定を返す API エンドポイントでは、シークレッ

ト名のみが返され、値は返されません。設定を更新する際には、保持したいすべてのシークレットも送信します。
。テンプレートからアクセスできます。複数回指定可能。

--url <文字列>

HTTP(s)のURLとオプションのポート。

proxmox-backup-manager notification matcher create <name> [OPTIONS] 新しい通知マッチャを追加します。

新しいnotification matcherを追加します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマです。

オプションのパラメータです：

--コメント <文字列>

コメント。

--disable <論理値>

このマッチャーを無効にします。

--invert-match <論理値>

フィルタ全体のマッチを反転します。

--match-calendar <文字列>

マッチする重大度レベルのリスト。複数指定可能。

-マッチフィールド <文字列>

一致するメタデータ・フィールドのリスト。複数回指定可能。

-マッチ度 <文字列>

一致する重大度レベルのリスト。複数回指定できます。

--mode all|any (default=all)

マッチの結果を結合するモード。

--origin user-created|builtin|modified-builtin,

通知構成エントリのオリジンです。

--ターゲット <文字列>

通知するターゲット。複数指定できます。

proxmox-backup-manager 通知マッチャ削除 <名前>

通知マッチャを削除します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

proxmox-backup-manager notification matcher list [OPTIONS] 通知マッチャの一覧を表示します。

通知マッチャの一覧を表示し

ます。

オプションのパラメータです：

--output-format text|json|json-pretty

出力フォーマット

proxmox-backup-manager notification matcher show <name> [OPTIONS]。

単一の matcher を表示します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマです。

オプションのパラメータ

--出力形式 text|json|json-pretty
出力フォーマットを指定します。

proxmox-backup-manager notification matcher update <name> [OPTIONS] 通知マッチャを更新します。

通知マッチャを更新します。

<name>

[<string>] ターゲットとマッチャーの名前スキーマです。

オプションのパラメータです：

--コメント <文字列>

コメント。

--delete

comment|disable|invert-match|match-calendar|match-field|match-severity|mode|target。

削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合に変更を防止します。これは、同時変更を防止するために使用できます。

--disable <論理値>

このマッチャを無効にします。

-invert-match <ブール値>

フィルタ全体のマッチを反転します。

--match-calendar <文字列>

マッチする重大度レベルのリスト。複数指定可能。

-マッチフィールド <文字列>

一致するメタデータ・フィールドのリスト。複数回指定できます。

-マッチ重大度 <文字列>

一致する重大度レベルのリスト。複数回指定可能。

-mode all|any (default=all)

マッチの結果を結合するモード。

--ターゲット <文字列>

通知するターゲット。複数指定できます。

proxmox-backup-manager notification target list [OPTIONS] 通知対象をリストします。

ターゲットをリストします。

オプションのパラメータです：

--output-format text|json|json-pretty

出力フォーマットを指定します。

proxmox-backup-manager notification target test <name>

指定された通知ターゲットをテストします。

<name>

[<string>] ターゲットとマッチャーの名前スキーマ

proxmox-backup-manager openid create <realm> --client-id <string> --issuer-url <string> [OPTIONS]

新しい OpenId レルムを作成します

```

<realm>
  [<string>] レルム名。

--client-id <string>
  OpenID クライアント ID

--issuer-url <文字列>
  OpenID Issuer Url オプショ

ンパラメータ：

--acr-values [<string>, ...] OpenID ACR リスト。
  OpenID ACR リスト

--autocreate <boolean> (default=false)
  ユーザが存在しない場合に自動的に作成します。

--client-key <文字列>
  OpenIDクライアントキー

--comment <文字列>
  コメント。

--default <boolean> (default=false)
  ログイン時に選択されるレルムをデフォルトにしたい場合は true。

--prompt <文字列>
  OpenIDプロンプト

--scopes [<string>, ...] (default=email profile)
  OpenIDスコープリスト

--username-claim <文字列>
  この属性/クレームの値を一意なユーザ名として使用します。一意性を保証するかどうかは ID プロバイダ次第
  です。OpenIDの仕様では、Subject('sub')が一意であることだけが保証されています。また、ユーザが自分でその属
  性を変更できないようにしてください！

```

proxmox-backup-manager openid delete <realm> [OPTIONS] OpenIDレルム設定を削除します。

OpenIDレルム設定を削除します

```
<realm>
  [<string>] レルム名。

オプションのパラメータです：
```

```
-digest <文字列>
  現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは同時修正を防ぐために使
  用できます。
```

proxmox-backup-manager openid list [OPTIONS] プロックスモックスバックアップマネージャのオープンIDリストです。

設定されている OpenId レルム一覧 オ

プションパラメータ：

```
--出力形式 text|json|json-pretty
  出力フォーマット
```

proxmox-backup-manager openid show <realm> [OPTIONS] OpenIDレルムの設定を表示します。

OpenIDレルムの設定を表示

[<string>] レルム名。

オプションのパラメータです：

```
--output-format text|json|json-pretty
出力形式。
```

proxmox-backup-manager openid update <realm> [OPTIONS] OpenID レルム設定を更新します。

OpenID レルム設定を更新します。

<realm>

[<string>] レルム名。

オプションのパラメータです：

```
--acr-values [<string>, ...] OpenID ACRリスト。
```

OpenID ACR リスト

```
--autocreate <boolean> (default=false)
```

ユーザが存在しない場合に自動的に作成します。

--client-id <文字列

OpenID クライアント ID

--client-key <文字列

OpenID クライアントキー

--コメント <文字列

コメント。

```
--default <boolean> (default=false)
```

ログイン時に選択されるデフォルトのrealmにしたい場合はtrueを指定します。

```
--delete client-key|comment|default|autocreate|scopes|prompt|acr-values
```

削除するプロパティのリスト。複数回指定できます。

-ダイジェスト <文字列

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため
に使用できます。

--issuer-url <文字列

OpenID 発行者 URL

--プロンプト <文字列

OpenID プロンプト

```
--scopes [<string>, ...] (default=email profile)
```

OpenID スコープリスト

proxmox-backup-manager prune-job create <id> --schedule <calendar-event>

--store <string> [OPTIONS] (オプション)

新しいpruneジョブを作成します。

<id>

[<string>] ジョブ ID。

--スケジュール <calendar-event>

指定したスケジュールで剪定ジョブを実行します。

--ストア <文字列>

データストア名。

オプションのパラメーター

:

--コメント <文字列

コメント。

```
--disable <boolean> (default=false)
このジョブを無効にします。

--keep-daily <integer> (1 - N)
毎日保持するバックアップの数。

--keep-hourly<整数> (1 - N)
1時間ごとに保持するバックアップの数。

--keep-last <integer> (1 - N)
保持するバックアップの数。

--keep-monthly <整数> (1 - N)
毎月保持するバックアップの数。

--keep-weekly <整数> (1 - N)
毎週保持するバックアップの数。

--keep-yearly <整数> (1 - N)
年間のバックアップ保持数。

--max-depth <integer> (0 - 7)
何レベルのネームスペースを操作するか (0== 再帰なし、空== 自動的な完全再帰、ネームスペースの深さは最大許容値を減らします)。

--ns <文字列>
名前空間を指定します。
```

proxmox-backup-manager prune-job list [OPTIONS] すべてのプルーンジョブをリストアップします。

すべての剪定ジョブをリ

ストアップします：

```
--output-format text|json|json-pretty
出力形式を指定します。
```

proxmox-backup-manager prune-job remove <id> [OPTIONS] プルーンジョブの設定を削除します。

プルーンのジョブ設定を削除します

<id>

[<string>] ジョブ ID。

オプションのパラメータです：

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合に変更を防止します。これは同時変更を防ぐために使用できます。

proxmox-backup-manager prune-job run <id> [OPTIONS] プルーンジョブを実行します。

指定されたpruneジョブを実行します。

<id>

[<string>] ジョブID。

オプションのパラメータです：

```
--output-format text|json|json-pretty
出力形式を指定します。
```

proxmox-backup-manager prune-job show <id> [OPTIONS] プルーンジョブの設定を表示します。

剪定ジョブの設定を表示します

<**id**>

[<string>] ジョブ ID。

オプションのパラメータ：

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager prune-job update <id> [OPTIONS] プルーンジョブの設定を更新します。

pruneジョブ設定を更新します。

<**id**>

[<string>] ジョブ ID。

オプションのパラメータ：

--comment <文字列

コメント。

--delete

comment|disable|ns|max-depth|keep-last|keep-hourly|keep-daily|keep-weekly|keep-monthly|。
削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため
に使用できます。

--disable <boolean> (default=false)

このジョブを無効にします。

--keep-daily <integer> (1 - N)

毎日保持するバックアップの数。

--keep-hourly<整数> (1 - N)

1時間ごとに保持するバックアップの数。

--keep-last <integer> (1 - N)

保持するバックアップの数。

--keep-monthly <整数> (1 - N)

毎月保持するバックアップの数。

--keep-weekly <整数> (1 - N)

毎週保持するバックアップの数。

--keep-yearly<整数> (1 - N)

保持する年間バックアップ数。

--max-depth <integer> (0 - 7)

何レベルのネームスペースを操作するか(0==再帰なし、空==自動的な完全再帰、ネームスペースの深さは最大
許容値を減らします)。

--ns <文字列

名前空間を指定します。

--スケジュール <calendar-event

指定したスケジュールで剪定ジョブを実行します。

--store <文字列

データストア名。

```
proxmox-backup-manager pull <remote> <remote-store> <store> [オプション].
```

別のリポジトリからデータを取得してデータストアを同期します。

<リモート

[<string>] リモートID。

<リモートストア

[<string>] データストア名。

<store> [<文字列

[<string>] データストア名。

オプションのパラメータ：

--バーストイン <文字列

バイトサイズとオプションの単位 (B、KB (base 10)、MB、GB、..., KiB (base 2)、MiB、Gib、...)。

--バーストアウト <文字列

バイトサイズ、単位は任意 (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, Gib, ...)。

--暗号化のみ <ブール値

暗号化されたバックアップスナップショットのみを同期し、他は除外します。

--group-filter [<exclude:>|<include:>]<type:><vm|ct|host>|group:GROUP|正規表現

:RE>] を指定します。

グループフィルターのリスト。複数指定可能。

--max-depth <integer> (0 - 7) (default=7)

何レベルの名前空間を操作するか (0== 再帰なし)。

--ns <文字列

名前空間を指定します。

--output-format text|json|json-pretty

出力形式。

--rate-in <文字列

バイトサイズ、単位は任意 (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, Gib, ...)。

--rate-out <文字列

バイトサイズ、単位は任意 (B、KB (ベース 10)、MB、GB、..., KiB (ベース 2)、MiB、Gib、...)。

--remote-ns <文字列

ネームスペース。

--remove-vanished <boolean> (default=false)

消えたバックアップを削除します。リモートバックアップが削除された場合、ローカルコピーを削除します。

--resync-corrupt <boolean> (デフォルト=false)

ローカルスナップショットの検証に失敗した場合、再同期を試みます。

--transfer-last <integer> (1 - N)

グループごとに 最後のN個のスナップショットに転送を制限します。

--verified-only <ブール値

検証済みのバックアップスナップショットのみを同期し、他は除外します。

```
proxmox-backup-manager push <store> <remote> <remote-store> [OPTIONS] (オプション)
```

別のリポジトリにプッシュしてデータストアを同期します。

<store>

[<string>] データストア名。

<remote>
[<string>] リモートID。

<リモートストア
[<string>] データストア名。

オプションのパラメータ：

--バーストイン <文字列
バイトサイズとオプションの単位 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)。

--バーストアウト <文字列
バイトサイズ、単位は任意 (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, GiB, ...)。

--暗号化のみ <論理値
暗号化されたバックアップスナップショットのみ同期し、他は除外します。

--group-filter [<exclude:>|<include:>]<type:><vm|ct|host>|group:GROUP|正規表現
:RE>。
グループ フィルタのリスト。複数回指定できます。

--max-depth <integer> (0 - 7) (default=7)
何レベルの名前空間を操作するか (0== 再帰なし)。

--ns <文字列
名前空間を指定します。

--output-format text|json|json-pretty
出力形式。

--rate-in <文字列
バイトサイズ、単位は任意 (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, GiB, ...)。

--rate-out <文字列
バイトサイズ、単位は任意 (B、KB (ベース 10)、MB、GB、...、KiB (ベース 2)、MiB、GiB、...)。

--remote-ns <文字列
ネームスペース。

--remove-vanished <boolean> (default=false)
消えたバックアップを削除します。リモート・バックアップが削除された場合、ローカル・コピーを削除します。

--transfer-last <integer> (1 - N)
グループごとに) 最後のN個のスナップショットに転送を制限します。

--verified-only <論理値
検証済みのバックアップスナップショットのみを同期し、他は除外します。

proxmox-backup-manager remote create <name> --auth-id <string> --host
<string> --password <string> [オプション]を指定します。

新しいリモートを作成します。

<name> を指定します。

[<string>] リモートID。

--auth-id <文字列
認証ID

--ホスト <文字列
DNS名またはIPアドレス。

--パスワード <文字列
リモートホストのパスワードまたは認証トークン。

オプションのパラメータ

--コメント <文字列

コメント。

-フィンガープリント <文字列

X509証明書のフィンガープリント(sha256)。

--ポート <整数

(オプションの)ポート

proxmox-backup-manager remote list [OPTIONS] 設定されているリモートの一覧を表示します。

設定されているリモートの一

覧を表示します。

オプションのパラメータです

:

--output-format text|json|json-pretty

出力フォーマットを指定します。

proxmox-backup-manager remote remove <name> [OPTIONS] 設定ファイルからリモートを削除します。

設定ファイルからリモートを削除します。

<name>

[<string>] リモートID。

オプションのパラメータ：

-ダイジェスト <文字列

現在の設定ファイルの SHA256 ダイジェストが異なる場合に変更を防止します。これは同時変更を防ぐために使用できます。

proxmox-backup-manager remote show <name> [OPTIONS] リモート設定を表示します。

リモート設定を表示します。

<name>

[<string>] リモートID。

オプションのパラメータです：

--output-format text|json|json-pretty

出力形式を指定します。

proxmox-backup-manager remote update <name> [OPTIONS] リモート設定を更新します。

リモート設定を更新します。

<name>

[<string>] リモート ID。

オプションのパラメータです：

--auth-id <文字列

認証ID

--コメント <文字列

コメント。

--delete comment|fingerprint|port

削除するプロパティのリスト。複数指定可能。

現在の設定ファイルの SHA256 ダイジェストが異なる場合に変更を防止します。これは、同時変更を防止するため使用できます。

-フィンガープリント <文字列>
X509証明書のフィンガープリント(sha256)。

--host <文字列>
DNS名またはIPアドレス。

--パスワード <文字列>
リモートホストのパスワードまたは認証トークン。

--ポート <整数>
(オプションの)ポート

proxmox-backup-manager レポート
システムレポート

proxmox-backup-manager s3 check <s3-endpoint-id> <bucket> [OPTIONS] (オプション)

指定されたS3クライアント設定の基本的なサニティチェックを実行します。

<s3-endpoint-id>
[<string>] S3クライアント設定を一意に識別するためのIDです。
<バケット>
[<string>] S3オブジェクトストアのバケット名。

オプションのパラメータです：

--store-prefix <文字列>
S3オブジェクトキーのバケット内のストア接頭辞(一般的にはデータストア名)

proxmox-backup-manager s3 endpoint create <id> --access-key <string> --endpoint <string> --secret-key <string> [OPTIONS] を指定します。

新しいs3クライアント設定を作成します。

<id>
[<string>] s3クライアント設定を一意に識別するID。
--access-key <string>
S3オブジェクトストアのアクセキー。
--endpoint <文字列>
S3オブジェクトストアにアクセスするためのエンドポイント。
--secret-key <string>
S3シークレットキー

— オプションパラメータ

:

-フィンガープリント <文字列>
X509 証明書のフィンガープリント (sha256)。

--path-style <boolean> (default=false)
vhost スタイルではなく path スタイルのバケットアドレスを使用します。

--ポート <整数>
S3オブジェクトストアにアクセスするポート。

--provider-quirks skip-if-none-match-header
プロバイダ固有の機能実装の癖のリスト。複数指定可能。

--put-rate-limit <整数>

put リクエストのレート制限を #request/s で指定します。

--region <文字列>

S3 オブジェクトストアにアクセスするリージョン。

proxmox-backup-manager s3 endpoint list [OPTIONS] 設定されているS3クライアントを一覧表示します。

設定済みのs3クライアントを

リストアップします。

オプションのパラメータです：

--出力フォーマット text|json|json-pretty

出力形式。

proxmox-backup-manager s3 エンドポイントリストバケット <s3-endpoint-id>

指定したS3クライアント設定によってアクセス可能なバケットをリストアップします。

<s3-endpoint-id>

[<string>] S3クライアント設定を一意に識別するためのIDです。

proxmox-backup-manager s3 endpoint remove <id> [OPTIONS] S3クライアント設定を削除します。

s3クライアント設定を削除します。

<id>

[<string>] ジョブID。

オプションのパラメータ：

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防ぎます。これは同時修正を防ぐために使用できます。

proxmox-backup-manager s3 endpoint update <id> [OPTIONS] s3クライアントの設定を更新します。

s3クライアントの設定を更新します。

<id>

[<string>] ジョブID。

オプションのパラメータです：

--access-key <文字列>

S3オブジェクトストアのアクセスキーです。

--delete port|region|fingerprint|path-style|provider-quirks

削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列>

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時修正を防止するために使用できます。

--エンドポイント <文字列>

S3オブジェクトストアにアクセスするエンドポイント。

-フィンガープリント <文字列>

X509証明書のフィンガープリント(sha256)。

--path-style <boolean> (default=false)

vhost形式ではなくpath形式のバケットアドレスを使用します。

--port <整数>

```
--provider-quirks skip-if-none-match-header
    プロバイダー固有の機能実装の癖のリスト。複数回指定可能。
```

```
--put-rate-limit <整数>
    put リクエストのレート制限を #request/s で指定。
```

```
--region <文字列>
    S3オブジェクトストアにアクセスするリージョン。
```

```
--secret-key <文字列>
    S3クライアントの秘密鍵。
```

proxmox-backup-manager subscription get [OPTIONS] サブスクリプション情報を読み取ります。

サブスクリプション情報を

読み込みます。

オプションのパラメータです：

```
--output-format text|json|json-pretty
    出力フォーマットを指定します。
```

proxmox-backup-manager subscription remove サブスクリプション情報を削除します。

サブスクリプション情報を削除します。

proxmox-backup-manager subscription set <key>

サブスクリプションキーを設定し、確認します。

<key>

[<string>] Proxmox Backup Serverのサブスクリプションキーです。

proxmox-backup-manager サブスクリプション **set-offline-key <データ>**

(内部使用のみ!) オフラインキーとして署名されたサブスクリプション情報blobを設定します

<data>

[<string>] base64エンコードされた署名付きサブスクリプション情報

proxmox-backup-manager サブスクリプション **更新 [OPTIONS]** サブスクリプションの状態を確認し、更新します。

サブスクリプションのステータスをチェックし

更新します。オプションのパラメータです：

```
--force <boolean> (default=false)
    キャッシュ内の情報が最新であっても常にサーバに接続します。
```

proxmox-backup-manager sync-job create <id> --remote-store <string> --store <string> [OPTIONS] 同期ジョブを作成します。

新しい同期ジョブを作成します。

<id>

[<string>] ジョブIDです。

--remote-store <string>

A.2. proxmox-backup-manager

--ストア <文字列>

データストア名。

オプションのパラメータ：

--バーストイン <文字列>

バイトサイズを任意の単位で指定します (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, Gib, ...)。

--バーストアウト <文字列>

バイトサイズ。単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、Gib、...)。

--コメント <文字列>

コメント。

--暗号化のみ <論理値>

暗号化されたバックアップスナップショットのみを同期し、他は除外します。

--group-filter [<exclude:>|<include:>]<type:><vm|ct|host>|group:GROUP|正規表現:>RE>]。

グループ フィルタのリスト。複数回指定できます。

--最大深度 <整数> (0 - 7)

何レベルの名前空間を操作するか (0== 再帰なし、空== 自動的な完全再帰、名前空間の深さは最大許容値を減らす)

--ns <文字列>

名前空間。

--owner <文字列>

認証ID

--rate-in <文字列>

バイトサイズ、単位は任意 (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, Gib, ...)。

--rate-out <文字列>

バイトサイズ。単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、Gib、...)。

--リモート <文字列>

リモートID。

--remote-ns <文字列>

名前空間。

--remove-vanished <boolean> (default=false)

消えたバックアップを削除します。リモート・バックアップが削除された場合、ローカル・コピーを削除します。

--resync-corrupt <ブール値>

ローカル・スナップショットの検証に失敗した場合、再試行します。

--run-on-mount<ブール値>

関連するデータストアがマウントされたときにこのジョブを実行します。

--スケジュール <calendar-event>

指定したスケジュールで同期ジョブを実行します。

--同期方向 pull|push (default=pull)

同期ジョブの方向 (プッシュまたはプル) を指定します。

--transfer-last <integer> (1 - N)

グループごとに 最後のN個のスナップショットに転送を制限します。

--verified-only <論理値>

検証済みのバックアップスナップショットのみ同期し、他は除外します。

proxmox-backup-manager sync-job list [OPTIONS] 同期ジョブのリスト。

同期ジョブリスト。

オプションのパラメータです：

```
--output-format text|json|json-pretty  
出力形式を指定します。  
  
-同期方向 all|push|pull (default=pull)  
リストされた同期ジョブの方向: push, pull, all.
```

proxmox-backup-manager sync-job remove <id> [OPTIONS] 同期ジョブの設定を削除します。

同期ジョブの設定を削除します。

<id>

[<string>] ジョブID。

オプションのパラメータです：

```
--digest <文字列>  
現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは同時修正を防ぐために使  
用できます。
```

proxmox-backup-manager sync-job run <id> [OPTIONS] 指定した同期ジョブを実行します。

指定した同期ジョブを実行します。

<id>

[<string>] ジョブID。

オプションのパラメータ：

```
--出力形式 text|json|json-pretty  
出力形式を指定します。
```

proxmox-backup-manager sync-job show <id> [OPTIONS] 同期ジョブの設定を表示します。

同期ジョブの設定を表示します

<id>

[<string>] ジョブID。

オプションのパラメータです：

```
--output-format text|json|json-pretty  
出力形式を指定します。
```

proxmox-backup-manager sync-job update <id> [OPTIONS] 同期ジョブの設定を更新します。

同期ジョブの設定を更新します。

<id>

[<string>] ジョブID。

オプションパラメータ：

```
--バーストイン <文字列>  
バイトサイズを任意の単位で指定します (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)  
  
--バーストアウト <文字列>  
バイトサイズ。単位は任意 (B、KB (10進数)、MB、GB、...、KiB (2進数)、MiB、GiB、...)  
  
--コメント <文字列>  
コメント。
```

```
--delete
remote|owner|comment|schedule|remove-vanished|group-filter|rate-in|burst-in|rate-out|bu
削除するプロパティのリスト。複数回指定できます。
```

-ダイジェスト <文字列>
現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するために使用できます。

--暗号化のみ <論理値>
暗号化されたバックアップスナップショットのみを同期し、他は除外します。

--group-filter [<exclude:>|<include:>]<type:><vm|ct|host>|group:GROUP|正規表現:RE>]
グループ フィルタのリスト。複数回指定できます。

--最大深度 <整数> (0 ~ 7)
何レベルの名前空間を操作するか (0== 再帰なし、空== 自動的な完全再帰、名前空間の深さは最大許容値を減らす)

--ns <文字列>
名前空間。

--owner <文字列>
認証ID

--rate-in <文字列>
バイトサイズ、単位は任意 (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, Gib, ...)。

--rate-out <文字列>
バイトサイズ。単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、Gib、...)。

--リモート <文字列>
リモートID。

--remote-ns <文字列>
名前空間。

--remote-store <文字列>
データストア名。

--remove-vanished <boolean> (default=false)
消えたバックアップを削除します。リモート・バックアップが削除された場合、ローカル・コピーを削除します。

--resync-corrupt <ブール値>
ローカル・スナップショットの検証に失敗した場合、再試行します。

--run-on-mount <ブール値>
関連するデータストアがマウントされたときにこのジョブを実行します。

--スケジュール <calendar-event>
指定したスケジュールで同期ジョブを実行します。

--ストア <文字列>
データストア名。

--同期方向 pull|push (default=pull)
同期ジョブの方向、プッシュまたはプル。

--transfer-last <integer> (1 ~ N)
(グループごとに)最後のN個のスナップショットに転送を制限します。

--verified-only <ブール値>
検証済みのバックアップスナップショットのみ同期し、他は除外します。

proxmox-backup-managerタスクリスト [OPTIONS] (オプション)

実行中のサーバタスクを一覧表示します。

オプションのパラメータです：

--all <ブール値>

停止中のタスクも一覧表示します。

--limit <integer> (1 - 1000) (default=50)

リストアップするタスクの最大数。

--出力形式 text|json|json-pretty

出力形式。

proxmox-backup-manager タスクログ <upid>

タスクログを表示します。

<upid>

[<string>] 一意なプロセス/タスク識別子

proxmox-backup-manager タスク 停止 <upid>

特定のタスクを停止しようとします。

<upid>

[<string>] ユニークなプロセス/タスク識別子

proxmox-backup-manager traffic-control create <name> --network <string> [OPTIONS] 新しいトラフィック制御ルールを作成します。

新しいトラフィック制御ルールを作成します。

<name>

[<string>] ルールID。

--ネットワーク <文字列>

ルールはこのネットワーク内のソースIPに適用されます。

オプションのパラメータ：

--バーストイン <文字列>

バイトサイズ、単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)。

--バーストアウト <文字列>

バイトサイズ。単位は任意 (B、KB (10進数)、MB、GB、...、KiB (2進数)、MiB、GiB、...)。

--コメント <文字列>

コメント。

--rate-in <文字列>

バイトサイズ、単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)。

--rate-out <文字列>

バイトサイズ。単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)。

--timeframe <文字列>

特定の時間にルールを有効にします。

proxmox-backup-manager トラフィックコントロールリスト [OPTIONS] 設定されたトラフィックコントロールルールを一覧表示します。

設定されたトラフィックコントロールルールを

一覧表示します。オプションのパラメータです

:

--output-format text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager traffic-control remove <name> [OPTIONS] 設定からトラフィック制御ルールを削除します。
設定ファイルからトラフィック制御ルールを削除します。

<name>

[<string>] ルールID。

オプションのパラメータです：

-ダイジェスト <文字列

現在の設定ファイルのSHA256ダイジェストが異なる場合に変更を防止します。これは同時修正を防ぐために使用できます。

proxmox-backup-manager traffic-control show <name> [OPTIONS] トラフィックコントロールの設定を表示します。
トラフィックコントロールの設定を表示します。

<name>

[<string>] ルールID。

オプションのパラメータです：

--output-format text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager traffic-control traffic [OPTIONS] トラフィックを制御します。

すべてのルールの現在のトラフィックを表示し

ます。オプションのパラメータ：

--出力形式 text|json|json-pretty
出力形式。

proxmox-backup-manager traffic-control update <name> [OPTIONS] トラフィック制御の設定を更新します。
トラフィック制御の設定を更新します。

<name>

[<string>] ルールID。

オプションパラメータ：

--バーストイン <文字列

バイトサイズを任意の単位で指定します (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)。

--バーストアウト <文字列

バイトサイズ。単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)。

--コメント <文字列

コメント。

--delete rate-in|burst-in|rate-out|burst-out|comment|timeframe
削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため使用できます。

--ネットワーク <文字列>

ルールはこのネットワーク内のソースIPに適用されます。

--rate-in <文字列>

バイトサイズ、単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、GiB、...)。

--rate-out <文字列>

バイトサイズ、単位は任意 (B、KB (ベース 10)、MB、GB、...、KiB (ベース 2)、MiB、GiB、...)。

--タイムフレーム <文字列>

特定の時間にルールを有効にします。

proxmox-backup-manager user create <userid> [OPTIONS] 新規ユーザを作成します。

新規ユーザを作成します。

<userid>

[<string>] ユーザID

オプションのパラメータ

--コメント <文字列>

コメント。

--email <文字列>

メールアドレス。

--enable <boolean> (`default=true`)

アカウントを有効にします (デフォルト)。0'に設定するとアカウントを無効にします。

--expire <integer> (0 - N) (`default=0`)

アカウントの有効期限 (エポックからの秒数)。0」は有効期限なしを意味します。

--名 <文字列>

名前。

--姓 <文字列>

姓。

--パスワード <文字列>

ユーザのパスワード。

proxmox-backup-manager user delete-token <userid> <token-name> [OPTIONS] ユーザのAPIトークンを削除します。

ユーザのAPIトークンを削除します。

<userid>

[<string>] ユーザID

<token-name> [<文字列>]

[<string>] APIトークン認証IDのトークンID部分。

これだけでは API トークンを一意に識別することはできません。

オプションのパラメータ

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防ぎます。これは同時修正を防ぐために使用できます。

proxmox-backup-manager user generate-token <userid> <token-name> [OPTIONS] (オプション)

指定されたメタデータを持つ新しいAPIトークンを生成します。

```
<userid>
  [<string>] ユーザーID
<token-name>
  [<string>] API トークン認証IDのトークンID部分。
  これだけでは API トークンを一意に識別することはできません。

オプションのパラメータ

--コメント <文字列>
  コメント。

--ダイジェスト <文字列>
  現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため
  に使用できます。

--enable <boolean> (<default=true>)
  アカウントを有効にします（デフォルト）。アカウントを無効にするには、これを「0」に設定します。

--expire <integer> (0 - N) (<default=0>)
  アカウントの有効期限（エポックからの秒数）。0」は有効期限なしを意味します。
```

proxmox-backup-manager user list [OPTIONS] 設定済みのユーザを一覧表示します。

設定されているユーザー
をリストアップします。

オプションのパラメータ
です：

```
--output-format text|json|json-pretty
  出力形式を指定します。
```

proxmox-backup-manager user list-tokens <userid> [OPTIONS] ユーザに関連付けられたトークンを一覧表示します。

ユーザに関連付けられたトークンを一覧表示します。

```
<userid>
  [<string>] ユーザID
```

オプションのパラメータです：

```
--output-format text|json|json-pretty
  出力形式。
```

proxmox-backup-manager user permissions <auth-id> [OPTIONS] ユーザ/トークンのアクセス許可を一覧表示します。

ユーザ/トークンのパーミッションを一覧表示します。

```
<auth-id>
  [<string>] 認証ID
```

オプションのパラメータ：

```
--output-format text|json|json-pretty
  出力形式。
```

```
--path <文字列>
  アクセス制御のパス。
```

proxmox-backup-manager user remove <userid> [OPTIONS] 設定ファイルからユーザを削除します。

設定ファイルからユーザを削除します。

<userid>

[<string>] ユーザID

オプションのパラメータです：

-**digest <文字列>**

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは同時変更を防ぐために使用できます。

proxmox-backup-manager user tfa delete <userid> <id> [OPTIONS] (オプション)

单一のTFAエントリを削除します。

<userid>

[<string>] ユーザID

<id>

[<string>] TFA エントリ ID

オプションのパラメータ

--**password <文字列>**

パスワード。

proxmox-backup-manager user tfa list <userid> [OPTIONS] ユーザのtfaメソッドを一覧表示します。

ユーザの全てのtfaメソッドをリストアップします。

<userid>

[<string>] ユーザID

オプションのパラメータです：

--**output-format text|json|json-pretty**

出力フォーマットを指定します。

proxmox-backup-managerユーザtfaロック解除<ユーザID>

ユーザーのTFA認証を解除します。

<ユーザーID>

[<string>] ユーザID

proxmox-backup-manager user update <userid> [OPTIONS] ユーザ設定を更新します。

ユーザ設定を更新します。ユーザのパスワードを変更するには 'PUT /access/password' エンドポイントを使います。

<userid>

[<string>] ユーザーID

オプションのパラメータ：

--**comment <文字列>**

コメント。

--**delete comment|ファーストネーム|ラストネーム|メール**

削除するプロパティのリスト。複数指定可能。

--**digest <文字列>**

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは同時修正を防ぐために使用できます。

--**email <文字列>**

メールアドレス。

```
--enable <boolean> (default=true)
アカウントを有効にします（デフォルト）。アカウントを無効にするには、これを「0」に設定します。

--expire <integer> (0 - N) (デフォルト=0)
アカウントの有効期限（エポックからの秒数）。0」は有効期限なしを意味します。

-名前 <文字列>
名前。

--姓 <文字列>
姓。

--パスワード <文字列>
ユーザのパスワードを変更するには 'PUT /access/password' を使用してください。
```

proxmox-backup-manager verify <store> [OPTIONS] バックアップを検証します。

バックアップの検証

```
<store>
[<string>] データストア名。
```

オプションのパラメータです：

```
--ignore-verified <boolean> (default=true)
検証が古くない場合、検証済みのバックアップを検証しません。

--outdated-after <整数> (0 - N)
検証が古くなる日数。(0は非推奨)

--output-format text|json|json-pretty
出力形式を指定します。
```

proxmox-backup-manager verify-job create <id> --store <string> [OPTIONS] 新しい検証ジョブを作成します。

新しい検証ジョブを作成します。

```
<id>
[<string>] ジョブIDです。
```

```
--store <文字列>
データストア名です
```

。

オプションのパラメータです：

```
--コメント <文字列>
コメント。
```

```
--ignore-verified <boolean> (default=true)
検証が古くない場合、検証済みのバックアップを検証しません。
```

```
--max-depth <integer> (0 - 7) (default=7)
何レベルのネームスペースを操作するか(0==再帰なし)。
```

```
--ns <文字列>
名前空間を指定します。
```

```
--outdated-after <整数> (0 - N)
検証が古くなる日数。(0は非推奨)
```

```
-スケジュール <カレンダー・イベント>
指定したスケジュールで検証ジョブを実行します。
```

proxmox-backup-manager verify-job list [OPTIONS] すべての検証ジョブを一覧表示します。

すべての検証ジョブを一覧表示し

ます：

--出力形式 text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager verify-job remove <id> [OPTIONS] 検証ジョブの設定を削除します。

検証ジョブの設定を削除します

<id>

[<string>] ジョブIDです。

オプションのパラメータです：

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは同時修正を防ぐために使用できます。

proxmox-backup-manager verify-job run <id> [OPTIONS] オプションを指定します。

指定された検証ジョブを実行します

<id>

[<string>] ジョブID。

オプションのパラメータです：

--出力形式 text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager verify-job show <id> [OPTIONS] 検証ジョブの設定を表示します。

検証ジョブの設定を表示します

<id>

[<string>] ジョブIDです。

オプションのパラメータです：

--output-format text|json|json-pretty
出力形式を指定します。

proxmox-backup-manager verify-job update <id> [OPTIONS] 検証ジョブの設定を更新します。

検証ジョブの設定を更新します。

<id>

[<string>] ジョブIDです。

オプションのパラメータです：

--コメント <文字列>

コメント。

--delete ignore-verified|comment|schedule|outdated-after|ns|max-depth.
削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列>

現在の設定ファイルの SHA256 ダイジェストが異なる場合に変更を防止します。これは、同時変更を防止するため使用できます。

--ignore-verified <boolean> (default=true)

すでに検証済みのバックアップの検証が古くない場合、そのバックアップを検証しません。

--max-depth <integer> (0 - 7) (default=7)

何レベルのネームスペースを操作するか (0== 再帰なし)。

--ns <文字列>

名前空間を指定します。

--outdated-after <integer> (0 - N)

検証が古くなる日数。(0は非推奨)

--スケジュール <カレンダー・イベント>

指定したスケジュールで検証ジョブを実行します。

--ストア <文字列>

データストア名。

proxmox-backup-manager versions [OPTIONS] プロックスモックスバックアップマネージャのバージョン。

Proxmox Backup Server の重要なパッケージのバージョンを一覧表示します。オプションのパ

ラメータ

--出力形式 text|json|json-pretty

出力形式を指定します。

--verbose <boolean> (default=false)

冗長なパッケージ情報を出力します。output-format が指定されている場合は無視されます。

A.3 proxmox-tape

proxmox-tape backup <store> <pool> [OPTIONS] データストアをテープにバックアップします。

データストアをテープメディアプールにバックアップします。

<store>

[<string>] データストア名。

<pool>

[<string>] メディアプール名。

オプションのパラメータです：

--ドライブ <文字列>

ドライブ識別子。

--eject-media <論理値>

ジョブ完了時にメディアを排出します。

--export-media-set <ブール値>

ジョブ完了時にメディアセットをエクスポートします。

--force-media-set <boolean> (default=false)

割り当てポリシーを無視して、新しいメディアセットを開始します。

--groups [<exclude:>|<include:>]<type:<vm|ct|host>|group:GROUP|正規表現:RE>]。

グループ フィルタのリスト。複数指定可能。

-latest-only<ブール値>

最新のスナップショットのみをバックアップします。

```
--max-depth <integer> (0 - 7) (default=7)
何レベルのネームスペースを操作するか (0== 再帰なし)。

--notify-user <文字列>
ユーザーID

--ns <文字列>
名前空間。

--output-format text|json|json-pretty
出力形式を指定します。
```

proxmox-tape backup-job create <id> --drive <string> --pool <string> --store <string> [OPTIONS]
--store <string> [OPTIONS] 新規テープバックアップジョブを作成します。

新しいテープバックアップジョブを作成します。

<id>
[<string>] ジョブID。

--ドライブ <文字列>
ドライブ識別子。

--プール <文字列>
メディアプール名。

--ストア <文字列>
データストア名。

オプションのパラメータ：

--コメント <文字列>
コメント。

--eject-media <論理値>
ジョブ完了時にメディアを排出します。

--export-media-set <boolean> ジョブの完了時にメディアセットをエクスポートします。
ジョブ完了時にメディアセットをエクスポートします。

--group-filter [<exclude:>|<include:>]<type:><vm|ct|host>|group:GROUP|正規表現
:RE>]。

グループ フィルタのリスト。複数指定可能。

-latest-only<布尔值>
最新のスナップショットのみをバックアップします。

--max-depth <integer> (0 - 7) (default=7)
何レベルのネームスペースを操作するか (0== 再帰なし)。

--notification-mode legacy-sendmail|notification-system
(default=notification-system)
このデータストアの通知をどのように送信するかを設定します。
通知をシステムの *sendmail* 実行ファイルを介して *notify-user* で設定されたユーザに送信します。
notification-system 通知システムに一致する通知イベントを発行します。

--notify-user <文字列>
ユーザーID

--ns <文字列>
名前空間

--スケジュール <calendar-event>
指定したスケジュールで同期ジョブを実行します。

--worker-threads <integer> (1 - 32) (default=1)
ジョブに使用するワーカースレッド数を設定します

proxmox-tape backup-job list [OPTIONS] テープバックアップジョブの一覧を表示します。

テープバックアップジョブ
のリスト。

オプションのパラメータ
:

--出力形式 text|json|json-pretty
出力フォーマットを指定します。

proxmox-tape backup-job remove <id> [OPTIONS] テープバックアップジョブの設定を削除します。

テープバックアップジョブの設定を削除します

<id>

[<string>] ジョブID。

オプションのパラメータ :

--digest <文字列

現在の構成ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは同時修正を防ぐために使用できます。

proxmox-tapeバックアップジョブ実行 <id>

THapeバックアップジョブを実行します

<id>

[<string>] ジョブ ID。

proxmox-tape backup-job show <id> [OPTIONS] テープバックアップジョブの設定を表示します。

テープバックアップジョブの設定を表示

<id>

[<string>] ジョブ ID。

オプションのパラメータ :

--output-format text|json|json-pretty
出力形式。

proxmox-tape backup-job update <id> [OPTIONS] (プロックスモックスステープバックアップジョブアップデート <id> [オプション])

テープバックアップジョブを更新します

<id>

[<string>] ジョブID。

オプションのパラメータです :

--コメント <文字列
コメント。

--delete
comment|schedule|eject-media|export-media-set|latest-only|notify-user|notification-mode
削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列

現在の設定ファイルのSHA256ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するため
に使用できます。

-ドライブ <文字列>
ドライブ識別子。

--eject-media <論理値>
ジョブ完了時にメディアを排出します。

--export-media-set <boolean> ジョブの完了時にメディアセットをエクスポートします。
ジョブ完了時にメディアセットをエクスポートします。

--group-filter [<exclude:>|<include:>]<type:><vm|ct|host>|group:GROUP|正規表現:>RE>]を指定します。
グループフィルターのリスト。複数指定可能。

-latest-only<ブール値>
最新のスナップショットのみをバックアップ。

--max-depth <integer> (0 - 7) (default=7)
何レベルのネームスペースを操作するか (0== 再帰なし)。

--notification-mode legacy-sendmail|notification-system (default=notification-system)
このデータストアの通知をどのように送信するかを設定します。
通知をシステムの *sendmail* 実行ファイルを介して *notify-user* で設定されたユーザに送信します。
notification-system 通知システムに一致する通知イベントを発行します。

--notify-user <文字列>
ユーザID

--ns <文字列>
名前空間を指定します。

--pool <文字列>
メディアプール名。

--スケジュール <カレンダーイベント>
指定したスケジュールで同期ジョブを実行します。

--ストア <文字列>
データストア名。

--worker-threads <integer> (1 - 32) (default=1)
ジョブに使用するワーカースレッドの数を設定します。

proxmox-tape barcode-label [OPTIONS] チェンジヤデバ

イスからのバーコードでメディアにラベルを付けます：

-drive <文字列>
ドライブ識別子。

--output-format text|json|json-pretty
出力フォーマット。

--pool <文字列>
メディアプール名。

proxmox-tape cartridge-memory [OPTIONS] オプションです。

Read Cartridge Memory (Medium auxiliary memory attributes) オプションパラ

メータ：

--ドライブ <文字列>
ドライブ識別子。

```
--output-format text|json|json-pretty
    出力フォーマット。
```

proxmox-tape catalog [OPTIONS] メディアを
スキャンして内容を記録 オプションのパラメー
タです：

--drive <文字列
 ドライブ識別子。

--force <論理値
 既存のインデックスを上書きします。

--output-format text|json|json-pretty
 出力形式。

-scan <論理値
 保存されたバージョンをリストアする代わりに、テープ全体を再読み込みしてカタログを再構築します。

--verbose <ブール値
 Verbose モード - 見つかった全てのチャンクをログに記録します。

proxmox-tape changer config <name> [OPTIONS] テープチェンジャーの設定を取得します。

テープチェンジャーの設定を取得します。

<name>
[<string>] テープチェンジャー識別子。

オプションのパラメータ：

--output-format text|json|json-pretty
 出力フォーマット。

proxmox-tape changer create <name> --path <string> [OPTIONS] 新しいチェンジャーデバイスを作成します。

新しいチェンジャーデバイスを作成します

<name>
[<string>] テープチェンジャー識別子。

--path <文字列
 Linux 汎用 SCSI デバイスへのパス (例: '/dev/sg4') オプションのパ

ラメータ：

--eject-before-unload <boolean> (default=false)
 true に設定すると、アンロード前にテープを手動で排出します。

--export-slots [<integer>, ...] (整数)
 カンマ区切りのスロット番号のリスト。これらのスロットはインポート/エクスポート用に予約されています。

proxmox-tape changer list [OPTIONS] チェンジャーのリストです。

チェンジャーのリストオ

プションパラメータです

：

--output-format text|json|json-pretty

A.3. ブロックスモックス

テープ

proxmox-tape changer remove <名前> を削除します。

テープチェンジャーの設定を削除します

<name>

[<string>] テープチェンジャー識別子。

proxmox-tape changer scan [OPTIONS] テープチェンジャーをスキャンします。

SCSI テープチェンジャーをスキャンしま

す：

--出力形式 text|json|json-pretty

出力フォーマット

proxmox-tape changer status [<name>] [OPTIONS] テープチェンジャーの状態を取得します。

テープチェンジャーのステータスを取得

<name>

[<string>] テープチェンジャー識別子。

オプションのパラメータ：

--cache <boolean> (default=true)

キャッシュされた値を使用。

--出力形式 text|json|json-pretty

出力形式。

proxmox-tape changer transfer [<name>] --from <integer> (1 - N) --to

<integer> (1 - N)

あるスロットから別のスロットへメディアを転送

<name>

[<string>] テープチェンジャー識別子。

--から <integer> (1 - N)

ソーススロット番号

-to <integer> (1 - N)

デスティネーションスロット番号

proxmox-tape changer update <name> [OPTIONS] (プロックスモックス・テープ・チェンジャー・アップデート <名前> [オプション])

テープチェンジャーの設定を更新します

<name>

[<string>] テープチェンジャー識別子。

オプションのパラメータ

--delete export-slots|eject-before-unload

削除するプロパティのリスト。複数指定可能。

-ダイジェスト <文字列>

現在の設定ファイルのSHA256ダイジェストが異なる場合に変更を防止します。これは、同時変更を防止するため
に使用できます。

```
--eject-before-unload <boolean> (default=false)
    true に設定すると、アンロード前にテープを手動で排出します。

--export-slots [<integer>, ...] (エクスポート・スロット [<整数>, ...])
    カンマ区切りのスロット番号のリスト。これらのスロットはインポート/エクスポート用に予約されています。つまり、これらのスロットにあるメディアは「オフライン」とみなされます。

--パス <文字列>
    Linux 汎用 SCSI デバイスへのパス (例 '/dev/sg4')
```

proxmox-tape clean [OPTIONS] (プロックスモックスステープクリーン)

ドライブをクリーン

オプションのパラメータ：

```
--ドライブ <文字列>
    ドライブ識別子。

--output-format text|json|json-pretty
    出力フォーマット。
```

proxmox-tape drive config <name> [OPTIONS] プールの設定を取得します。

プールの設定を取得

```
<name>
    [<string>] ドライブ識別子。
```

オプションのパラメータ：

```
--output-format text|json|json-pretty
    出力形式。
```

proxmox-tape drive create <name> --path <string> [OPTIONS] 新しいドライブを作成します。

新しいドライブを作成

```
<name>
    [<string>] ドライブ識別子。

--パス <文字列>
    LTO SCSIジェネリック・テープ・デバイスへのパス (例 '/dev/sg0') オプション・パラメータ：
```

```
--changer <文字列>
    テープチェンジャー識別子。

--changer-drivenum <integer> (0 - 255) (デフォルト=0)
    関連するチェンジャードライブ番号 (changer オプションが必要)
```

proxmox-tape drive list [OPTIONS] (プロックスモックスステープドライブリスト)

ドライブをリストアップ

オプションのパラメータ

```
--output-format text|json|json-pretty
    出力フォーマット。
```

proxmox-tape drive remove <名前> ドライブの設定を削除します。

ドライブ構成を削除

<name>

[<string>] ドライブ識別子。

proxmox-tape drive scan [OPTIONS] ドライブをスキャンします。

ドライブのスキャン オプ

ションパラメータ：

--output-format text|json|json-pretty

出力フォーマット。

proxmox-tape drive update <name> [OPTIONS] ドライブの設定を更新します。

ドライブの設定を更新

<name>

[<string>] ドライブ識別子。

オプションのパラメータ：

--changer <文字列

テープチェンジャー識別子。

--changer-drivenum <integer> (0 - 255) (デフォルト=0)

関連するチェンジャー・ドライブ番号 (changer オプションが必要)

--delete changer|changer-drivenum

削除するプロパティのリスト。複数指定可。

-ダイジェスト <文字列

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは、同時変更を防止するために使用できます。

--パス <文字列

LTO SCSIジェネリックテープデバイスへのパス (例 '/dev/sg0')

proxmox-tape eject [OPTIONS] ドライブメディアをイジェクト/アンロードします。

ドライブメディアのイジェクト/

アンロード オプションのパラメ

ータです：

-drive <文字列

ドライブ識別子。

--output-format text|json|json-pretty

出力フォーマット。

proxmox-tape eod [OPTIONS] (オプション)

メディアの最後に移動 (MTEOM、デバッグに使用) オプショ

ンパラメータ：

A.3. プロックスモックス

テ

--ドライブ <文字列

ドライブ識別子。

proxmox-tape export-media <label-text> [OPTIONS] 指定されたラベルのメディアをエクスポートします。

指定されたラベルでメディアをエクスポートします

<label-text>

[<string>] メディアラベル/バーコード。

オプションのパラメータ：

--ドライブ <文字列

ドライブ識別子。

proxmox-tape format [OPTIONS] プロックスモックスステープフォーマット。

メディアのフォーマット

オプションパラメータ：

--ドライブ <文字列

ドライブ識別子。

--fast <boolean> (default=true)

高速消去を使用。

--output-format text|json|json-pretty

出力フォーマット。

proxmox-tape help [{<コマンド>}]] [OPTIONS] (オプション)

指定したコマンド（またはサブコマンド）のヘルプを表示します。

<コマンド

[<string>] コマンド。ネストしたサブコマンドを指定するため、リストにすることができます。複数指定可能。

オプションのパラメータ：

--verbose <ブール値

冗長なヘルプを表示します。

proxmox-tape inventory [OPTIONS] (プロックスモックス・テープ・インベントリ)

メディアラベルをリストアップ（および更新）します（Changer

Inventory） オプションパラメータ：

-catalog <boolean> (default=false)

テープからカタログの復元を試みます。

--ドライブ <文字列

ドライブ識別子。

--output-format text|json|json-pretty

出力形式。

--read-all-labels <boolean> (default=false)

すべてのテープを読み込み、ラベルの読み取りを試みます（すでにインベントリ化されている場合でも）。

--read-labels <boolean> (default=false)

未知のテープを読み込み、ラベルの読み込みを試行

proxmox-tape key change-passphrase <fingerprint> --hint <string> [OPTIONS] 暗号化キーを変更します。

暗号化キーのパスワードを変更します。

<fingerprint> --hint <string> [OPTIONS] 暗号化キーのパスワードを変更します。

[<string>] テープ暗号化キーの指紋 (sha256)。

--hint <文字列

パスワードのヒント

。

オプションのパラメータ

:

--force <boolean> (default=false)

テープキーのパスフレーズをリセット。

--kdf none|scrypt|pbkdf2 (default=scrypt)

パスワードで保護された暗号鍵のための鍵生成関数です。

proxmox-tape key create --hint <string> [OPTIONS] キーを作成します。

鍵を作成 (標準入力からパスワードを読み込みます)

--hint <文字列

パスワード復元のヒント。

オプションのパラメータ :

--kdf none|scrypt|pbkdf2 (default=scrypt)

パスワードで保護された暗号鍵のための鍵導出関数。

proxmox-tape キー リスト [OPTIONS]

鍵をリストアップします。

オプションのパラメータ :

--出力形式 text|json|json-pretty

出力形式。

proxmox-tape key paperkey <fingerprint> [OPTIONS] (プロックスモックス・テープ・キー・ペーパーキー<フィンガープリント>)

暗号化キーを含む、印刷可能で人間が読めるテキストファイルを生成します。このファイルに

は、キーを素早く復元するためのスキャン可能なQRコードも含まれています。

<fingerprint> [<string>] テープ暗号化キー

[<string>] テープ暗号化キーの指紋 (sha256)。

オプションのパラメータ :

--出力形式 text|html

Paperkeyの出力形式

--件名 <文字列

指定した件名をタイトルテキストとして含めます。

proxmox-tape key remove <fingerprint> [OPTIONS] 暗号化キーをデータベースから削除します。

データベースから暗号化キーを削除します。

このキーを使ったテープへのアクセスはできなくなりますので注意してください。

<フィンガープリント

[<string>] テープ暗号化キーの指紋 (sha256)。

オプションのパラメータ：

-**digest** <文字列

現在の設定ファイルの SHA256 ダイジェストが異なる場合、変更を防止します。これは、同時修正を防ぐために使用できます。

proxmox-tape key restore [OPTIONS] 暗号化キーをテープからリストアします。

暗号化キーをテープまたはバックアップファイル/文字列からリストアします：

--**drive** <文字列

ドライブ識別子。

--**key** <文字列

json文字列またはエクスポートされたpaperkey-formatからキーをインポートします。

--**key-file** <文字列

json またはエクスポートされた paperkey 形式のファイルからキーをインポートします。

proxmox-tape key show <fingerprint> [OPTIONS] 暗号化キーを表示します。

暗号化キーのメタデータを表示します。

<**fingerprint**>

[<string>] テープ暗号化キーの指紋 (sha256)。

オプションのパラメータ：

--**output-format** text|json|json-pretty

出力形式。

proxmox-tape label --label-text <string> [オプション]。

ラベルメディア

--**ラベルテキスト** <文字列

メディアラベル/バーコード。

オプションのパラメータ：

--**drive** <文字列

ドライブ識別子。

--**output-format** text|json|json-pretty

出力形式。

--**pool** <文字列

メディアプール名

proxmox-tape load-media <label-text> [オプション]

指定されたラベルでメディアをロード

<**label-text**>

[<string>] メディアラベル/バーコード。

オプションのパラメータ：

--ドライブ <文字列>
 ドライブ識別子。

--output-format text|json|json-pretty
 出力フォーマット。

proxmox-tape load-media-from-slot <source-slot> [OPTIONS] 指定されたスロットからメディアをロードします。

指定されたスロットからメディアをロード

<ソーススロット>

[<integer> (1 - N)] ソーススロット番号。

オプションのパラメータ：

--ドライブ <文字列>
 ドライブ識別子。

proxmox-tape media content [OPTIONS] メディアの内容をリストします。

メディアの内容を一覧表

示 オプションパラメータ

:

--バックアップID <文字列>
 バックアップID。

--バックアップタイプ vm|ct|host
 バックアップの種類。

--label-text <文字列>
 メディア・ラベル/バーコード。

--メディア <文字列>
 メディアUUID。

--メディアセット <文字列>
 MediaSet Uuid (特定のプールの空のメディアを再検索するために、すべてゼロのUuidを使用します)。

--output-format text|json|json-pretty
 出力形式を指定します。

--pool <文字列>
 メディアプール名。

proxmox-tape media destroy [<label-text>] [OPTIONS] メディアを破棄します。

メディアを破棄します (データベースから完全に削除します)

<label-text>

[<string>] メディアラベル/バーコード。

オプションのパラメータ：

--force <論理値>
 メディアがメディアセットで使用されている場合でも) 強制的に削除します。

--uuid <文字列>
 メディアの Uuid。

proxmox-tape media list [OPTIONS] プールのメディアをリストアップします。

リストプールメディア オ

ーションパラメータ：

```
--output-format text|json|json-pretty
    出力フォーマット。
```

```
--pool <文字列>
    メディアプール名。
```

```
--update-status <boolean> (default=true)
    テープ・ライブラリのステータスを更新しようとします（どのテープがオンラインであるかをチェックします）。
```

```
--update-status-changer <string> テープ・チェンジャー識別子。
    テープチェンジャー識別子。
```

proxmox-tape pool config <name> [OPTIONS] メディアプールの設定を取得します。

メディアプールの設定を取得

<name>
 [<string>] メディアプール名。

オプションのパラメータ：

```
--output-format text|json|json-pretty
    出力形式。
```

proxmox-tape pool create <name> [OPTIONS] 新しいメディアプールを作成します。

新しいメディアプールを作成します

<name>
 [<string>] メディアプール名。

オプションのパラメータ

--allocation <文字列>
 メディアセットの割り当てポリシー ('continue'、'always'、またはカレンダーイベント)。

--コメント <文字列>
 コメント。

--encrypt <文字列>
 テープ暗号化キーのフィンガープリント (sha256)。

--retention <文字列>
 メディア保持ポリシー ('overwrite'、'keep'、またはタイムスパン)。

--テンプレート <文字列>
 メディアセットの命名テンプレート (strftime() の時間フォーマット指定を含む可能性があります)。

proxmox-tape pool list [OPTIONS] メディアプールをリストします。

メディアプールをリスト

アップ オプションパラメ

ータ：

```
--出力フォーマット text|json|json-pretty
    出力フォーマットを指定します。
```

proxmox-tape pool remove <名前> を削除します。

メディアプール構成の削除

<名前>

A.3. ブロックスマックステ

ープ

[<string>] メディアプール名。

proxmox-tape pool update <name> [OPTIONS] メディアプールの設定を更新します。

メディアプールの設定を更新

<name> [<文字列>] メディアプール名

[<string>] メディアプール名。

オプションのパラメータ：

--allocation <文字列>

メディアセットの割り当てポリシー ('continue'、'always'、またはカレンダーイベント)。

--コメント <文字列>

コメント。

--delete allocation|retention|template|encrypt|comment

削除するプロパティのリスト。複数指定可能。

--encrypt <文字列>

テープ暗号化キーのフィンガープリント (sha256)。

--retention <文字列>

メディア保持ポリシー ('overwrite'、'keep'、またはタイムスパン)。

--テンプレート <文字列>

メディアセットの命名テンプレート(strftime())の時間フォーマット指定を含むかもしれません)。

proxmox-tape read-label [OPTIONS] プロックスモックスステープのラベルを読み取ります。

メディアラベルを読む オ

ーションパラメータ

--ドライブ <文字列>

ドライブ識別子。

--inventorize <論理値>

メディアのインベントライズ

--output-format text|json|json-pretty

出力形式。

proxmox-tape restore <メディアセット> <ストア> [<スナップショット>] [オプション]。

メディアセットからデータをリストア

<メディアセット>

[<string>] メディアセットのUUID。

<store>

[[(<ソース>)?<ターゲット>, ...]]。カンマ区切りのデータストア・マッピング（または単一のデータストア）のリスト。例えば、'a=b,e' はソース・データストア 'a' をターゲット 'b and all other' にマッピングします。

ソースを既定の 'e' にマッピングします。既定値が指定されない場合、指定されたソースのみがマッピングされます。

<スナップショット>

[store:[ns/namespace/...]type/id/time] スナップショットのリスト。複数回指定できます。

オプションのパラメータ：

--ドライブ <文字列>

ドライブ識別子。

--namespaces [store=<string> [,max-depth=<integer>] [,source=<string>] [,target=<string>]] リスト。
リストアするネームスペースのリスト。複数回指定できます。

--notify-user <文字列
ユーザID

--出力形式 text|json|json-pretty
出力形式。

--owner <文字列
認証ID

proxmox-tape rewind [OPTIONS] テープを巻き戻します。

テープを巻き戻す オプション

ヨンパラメータ

--ドライブ <文字列
ドライブ識別子

--output-format text|json|json-pretty
出力フォーマット。

proxmox-tape scan [OPTIONS] オプションを指定します。

巻き戻し、メディアの内容を読み取り、デバッグ情報を表示

注意: これは、ドライバがIOエラーを返さない限り読み込みますので、このメソッドはEOTに達したときに失敗すると思われます。

オプションのパラメータ

--ドライブ <文字列
ドライブ識別子。

proxmox-tape status [OPTIONS] ドライブ/メディアのステータスを取得します。

ドライブ/メディアのステータス

を取得 オプションパラメータ:

--ドライブ <文字列
ドライブ識別子。

--output-format text|json|json-pretty
出力フォーマット

proxmox-tape unload [OPTIONS] チェンジャー経由でメディアをアンロードします。

チェンジャー経由でメディアをア

ンロード オプションパラメータ

:

--ドライブ <文字列
ドライブ識別子。

--output-format text|json|json-pretty
出力フォーマット。

--ターゲットスロット <integer> (1 - N)

ターゲット・スロット番号。省略された場合、デフォルトはドライブがロードされたスロットです。

proxmox-tape volume-statistics [OPTIONS] ボリューム統計 (SCSI ログ・ページ 17h)

ボリューム統計を読み込む (SCSI ログページ 17h) オプション

のパラメータです：

--ドライブ <文字列

ドライブ識別子。

--output-format text|json|json-pretty

出力フォーマット。

A.4 pmt

すべてのコマンドは、テープ・デバイスを指定するための以下のパラメータをサポートしています：

--device </パス Linux テープ・デバイスへのパス。**--ドライブ <名前** Proxmox Backup Server構成のドライブを使用します。

出力を生成するコマンドは--output-format/パラメータをサポートしています。以下の値を受け付けます：

テキスト

テキスト形式（デフォルト）。人間が読めます。

json

JSON (單一行)。

json-pretty

JSON (複数行、きれいにフォーマットされたもの)。

pmt asf<カウント

テープをカウントファイルの先頭（ファイルマークカウントの後）に配置

<count>

[<integer> (0 - 2147483647)] ファイルマークの位置 (0 は BOT)。

pmt bsf<count

後方スペースカウントファイル（ファイルマークの前の位置）。

テープは前のファイルの最後のブロックに配置されます。

<count>

[<integer> (1 - 2147483647)] ファイルマーク数。

pmt bsfm <カウントバックワードスペースでファイル数をカウントし、次にフォワードスペースで1レコード（ファイルマークの後の位置）をカウントします。
。

これにより、テープは現在のファイルより count - 1 ファイル前のファイルの最初のブロックに位置します。

<count>

[<integer> (1 - 2147483647)] ファイルマーク数。

pmt bsr <count>

後方スペースレコード

<count>

[<integer> (1 - 2147483647)] レコード数。

pmt カートリッジメモリ

カートリッジメモリの読み込み

pmt イジェクト

ドライブメディアの取り出し

pmt eod

メディアの最後に移動

pmt erase [OPTIONS]

メディアの消去（現在の位置から）オプション

のパラメータ：

--fast <boolean> (default=true)

高速消去を使います。

pmt format [OPTIONS] メディアをフ

オーマットします：

--fast <boolean> (default=true)

高速消去を行います。

pmt fsf <カウント>

前方スペース・カウント・ファイル（ファイル・マーク後の位置）

。テープは次のファイルの最初のブロックに配置されます。

<count>

[<integer> (1 - 2147483647)] ファイルマーク数。

pmt fsfm <カウント>

フォワードスペースでファイル数をカウントし、その後バックスペースで1レコード（ファイルマークの前の位置）スペース。

これにより、テープは現在のファイルから count - 1 ファイル先のファイルの最後のブロックに位置することになります。

<count>

[<integer> (1 - 2147483647)] ファイルマーク数。

pmt fsr <カウント

フォワードスペースレコード。

<count>

[<integer> (1 - 2147483647)] レコード数。

pmt help [{<コマンド>}] [OPTIONS] 指定されたコマンド (またはサブコマンド) に関するヘルプを表示します。

指定したコマンド (またはサブコマンド) に関するヘルプを表示。

<コマンド>

[<string>] コマンド。ネストされたサブコマンドを指定するためにリストにすることができます。複数指定可能。

オプションのパラメータ：

--verbose <ブール値>

詳細なヘルプ。

pmt load

メディアをロード

pmt lock

テープドライブのドアをロック

pmt options [OPTIONS] 様々な

ドライブ・オプションを設定 オ

プション・パラメーター

--blocksize <integer> (0 - 8388608)

テープ・ドライブのブロック長を設定します (0 は可変長)。

-buffer_mode <論理値>

ドライブ・バッファを使用します。

--compression <論理値>

圧縮の有効/無効を設定。

--defaults <ブール値>

デフォルトのオプションを設定

pmt rewind

テープを巻き戻す

pmt scan

既存のテープチェンジャーデバイスをスキャン

pmtステータス

ライブステータス

pmt tape-alert-flags

テープ警告フラグの読み込み

pmt アンロック

テープドライブドアのロック解除

pmt ボリューム統計

ボリューム統計

pmt weof [<count>] ボリューム統計

現在の位置に EOF マークを書き込みます。

<count>

[<integer> (1 - 2147483647)] ファイルマーク数。

A.5 pmtx

pmtx help [{<command>}] [OPTIONS] 指定されたコマンド（またはサブコマンド）に関するヘルプを表示します。

指定したコマンド（またはサブコマンド）に関するヘルプを表示。

<コマンド>

[<string>] コマンド。ネストされたサブコマンドを指定するためにリストにすることができます。複数指定可能。

オプションのパラメータ：

--verbose <ブール値>

詳細なヘルプ。

pmtx 照会

問い合わせ

pmtx インベントリ

インベントリ

pmtx load <slot> [OPTIONS] (日本語)

ロード

<slot>

[<integer>] ストレージスロット番号 (ソース)。

オプションのパラメータ：

--drivenum <integer>

ターゲットドライブ番号(デフォルトはドライブ0)

pmtx スキャン

既存のテープチェンジャーデバイスのスキャン

PMTX ステータス

チェンジャーステータス

pmtx 転送 <from> <to>

転送

<from>

[<integer>] 転送元ストレージスロット番号。

<to>

[<integer>] ターゲットのストレージスロット番号。

pmtx unload [OPTIONS] アンロード

アンロード

オプションのパラメータ：

--drivenum <整数>

ターゲットドライブ番号 (デフォルトはドライブ 0)

--スロット <integer>

ストレージ・スロット番号 (ターゲット)。省略時のデフォルトは、ドライブがロードされたスロット。

A.6 pxar

pxar create <archive> <source> [OPTIONS] 新しい .pxar アーカイブを作成します。

新しい .pxar アーカイブを作成します。

<アーカイブ>

[<string>] アーカイブ名。

<ソース>

[<string>] ソースディレクトリ。

オプションのパラメータ：

--all-file-systems <boolean> (デフォルト=false)

マウントされた sudir を含めます。

--entries-max <integer> (0 - 9223372036854775807) (default=1048576)

一度にメモリに読み込まれるエントリの最大数

--exclude <文字列>

除外するパスまたはパターンマッチングファイルのリスト。複数指定可能。

--no-acls <boolean> (default=false)

アクセス制御リストのエントリを無視します。

--no-device-nodes <boolean> (default=false) デバイスノードを無視します。

デバイスノードを無視します。

--no-fcaps <boolean> (default=false)

ファイル機能を無視します。

```
--no-fifos <boolean> (default=false)
    fifos を無視します。

--no-sockets <boolean> (default=false)
    ソケットを無視します。

--no-xattrs <boolean> (default=false)
    拡張ファイル属性を無視します。

--ペイロード出力 <文字列>
    'ppxar' 分割アーカイブを作成するためのペイロード出力データファイル。
```

pxar extract <アーカイブ> [<ターゲット>] [オプション].

アーカイブを展開します。

<archive>

[<string>] アーカイブ名。

<target>

[<string>] 対象ディレクトリ

オプションのパラメータ：

--allow-existing-dirs <boolean> (デフォルト=false)
 リストア時に既に存在するディレクトリを許可します。

--files-from <文字列>
 リストアするファイルのマッチパターンを含むファイル。

--no-acls <boolean> (default=false)
 アクセス制御リストのエントリを無視します。

--no-device-nodes <boolean> (default=false)
 デバイスノードを無視します。

--no-fcaps <boolean> (default=false)
 ファイル機能を無視します。

--no-fifos <boolean> (default=false)
 fifos を無視。

--no-sockets <boolean> (default=false)
 ソケットを無視。

--no-xattrs <boolean> (default=false)
 拡張ファイル属性を無視。

--overwrite <boolean> (default=false)
 既存のファイル、シンボリックリンク、ハードリンクを上書きします。

--overwrite-files <boolean> (default=false)
 既存のファイルを上書き

--overwrite-hardlinks <boolean> (default=false)
 既存のエントリをアーカイブのハードリンクで上書き

--overwrite-symlinks <論理値> (default=false)
 既存のエントリをアーカイブのシンボリックリンクで上書き

--pattern <文字列>
 リストアするファイルを制限するためのパスまたはマッチパターン。複数指定可能。

-payload-input (ペイロード入力) <文字列>
 'ppxar'ペイロードの入力データファイルで、分割されたアーカイブをリストアします。

--prelude-target <文字列>
 pxar アーカイブのプレリュードをリストアするパス。

--strict <論理値> (default=false)

エラー時に停止します。そうでなければ、ほとんどのエラーは単に警告するだけです。

pxar help [<コマンド>] [OPTIONS] (オプション)

指定したコマンド（またはサブコマンド）のヘルプを表示します。

<コマンド>

[<string>] コマンド。ネストしたサブコマンドを指定するためにリストにすることができます。複数指定可能。

オプションのパラメータ：

--verbose <ブール値>

詳細なヘルプ。

pxar list <archive> [OPTIONS] アーカイブの内容を一覧表示します。

アーカイブの内容をリストアップ。

<archive>

[<string>] アーカイブ名。

オプションのパラメータ：

-ペイロード入力 <文字列>

'ppxar' 分割アーカイブのペイロード入力データファイルです。

pxar mount <アーカイブ> <マウントポイント> [OPTIONS].

指定されたマウントポイントに FUSE 経由でアーカイブをマウントします。

<アーカイブ名>

[<string>] アーカイブ名です。

<マウントポイント>

[<string>] ファイルシステムのマウントポイント。

オプションのパラメータです：

--payload-input <文字列>

'ppxar' 分割アーカイブをリストアするためのペイロード入力データファイル。

--verbose <boolean> (default=false)

デバッグ用にフォアグラウンドで動作します。

A.7 proxmox-file-restore

proxmox-file-restore extract <snapshot> <path> [<target>] [OPTIONS] バックアップスナップショットからファイルをリストアします。

バックアップスナップショットからファイルをリストアします。

<スナップショット>

[<string>] グループ/スナップショットパス。

<path>

[<string>] リストアするパス。標準出力に展開された場合、ディレクトリはアーカイブファイルとしてリストアされます。

<ターゲット>

[<string>] 対象ディレクトリのパス。標準出力に書き出すには'-'を使用します。

オプションのパラメータ：

```
--base64 <boolean> (default=false)
設定された場合、'path'はbase64エンコードされたものとして解釈されます。

-暗号化モード none|encrypt|sign-only (default=encrypt)
データを暗号化（AEAD暗号を使用）するか、署名のみにするか、あるいはそのいずれでもないかを定義します。

--ドライバ Qemu
使用するブロックドライバタイプ。

--format plain|pxar|zip|tar
結果のフォーマット

--keyfd <整数> (0 - N)
既にオープンされているファイル記述子を介して暗号鍵を渡します。

--keyfile <文字列>
暗号化キーへのパス。すべてのデータはこの鍵を使って暗号化されます。

--ns <文字列>
名前空間。

--リポジトリ <文字列>
リポジトリのURL。

--verbose <boolean> (default=false)
冗長な情報を表示

--zstd <boolean> (default=false)
true の場合、出力は zstd 圧縮されます。
```

proxmox-file-restore help [<コマンド>] [OPTIONS] (オプション)

指定したコマンド（またはサブコマンド）のヘルプを表示します。

<コマンド>

[<string>] コマンド。ネストされたサブコマンドを指定するためにリストすることができます。複数指定可能。

オプションのパラメータ：

```
--verbose <ブール値>
冗長なヘルプを表示します。
```

proxmox-file-restore list <snapshot> <path> [OPTIONS] バックアップスナップショットからディレクトリをリストアップします。

バックアップスナップショットからディレクトリをリストアップします。

<snapshot>

[<string>] グループ/スナップショットのパス。

<path>

[<string>] リストする(サブ)パス。

オプションのパラメータ：

```
--base64 <boolean> (default=false)
設定された場合、'path'はbase64エンコードされたものとして解釈されます。
```

```
-暗号化モード none|encrypt|sign-only (default=encrypt)
データを暗号化（AEAD暗号を使用）するか、署名のみにするか、あるいはそのいずれでもないかを定義します。
```

--ドライバ Qemu

使用するブロックドライバの種類。

```
--keyfd <integer> (0 - N)
既にオープンされているファイル記述子を介して暗号化キーを渡します。
```

--keyfile <文字列>
暗号化キーへのパス。すべてのデータはこの鍵を使って暗号化されます。

--ns <文字列>
名前空間。

--output-format text|json|json-pretty
出力形式。

--リポジトリ <文字列>
リポジトリのURL。

--timeout <integer> (1 - N)
呼び出しにかかる最大時間を指定します。

proxmox-file-restore status [OPTIONS] (オプション)

現在実行中/マップされたリストイメージのステータス情報を取得します。 オプションパラメ

ータです：

--ドライバ qemu
使用するブロックドライバタイプ。

--output-format text|json|json-pretty
出力フォーマット。

proxmox-file-restore stop <name>

指定されたイメージを直ちに停止/アンマップします。 VMはタイマーの後に停止するので、通常は必要ありません。

<name>
[<string>] 停止する VM の名前です。

A.8 proxmox-backup-debug

proxmox-backup-debug api create <api-path> [OPTIONS] プロックスモックスバックアップデバッグAPIを呼び出します。

<api-path> で API を呼び出します。

<api-path>
[<string>] API パス。

オプションのパラメータ：

--output-format text|json|json-pretty
出力形式。

proxmox-backup-debug api delete <api-path> [OPTIONS] APIを削除します。

<api-path>/ APIを呼び出します。

<api-path>
[<string>] API パス。

オプションのパラメータ：

--出力形式 text|json|json-pretty
出力形式。

proxmox-backup-debug api get <api-path> [OPTIONS] オプションを指定します。

<api-path> でAPIを呼び出します。

<api-path>

[<string>] API パス。

オプションのパラメータ：

--出力形式 text|json|json-pretty

出力形式。

proxmox-backup-debug api ls [<path>] [OPTIONS].

<path> のAPI 利用情報を取得します。

<path>

[<string>] API パス。

オプションのパラメータ：

--出力形式 text|json|json-pretty

出力形式を指定します。

proxmox-backup-debug api set <api-path> [OPTIONS] オプションを指定します。

<api-path>/ でAPIを呼び出します。

<api-path>

[<string>] API パス。

オプションのパラメータ：

--出力形式 text|json|json-pretty

出力形式。

proxmox-backup-debug api usage <path> [OPTIONS] オプションを指定します。

<path> のAPI利用情報を取得します。

<path>

[<string>] API パス。

オプションのパラメータ：

--verbose <boolean> (default=false)

冗長出力フォーマット

proxmox-backup-debug diff archive <prev-snapshot> <snapshot> <archive-name> [OPTIONS] (オプション)

二つのスナップショットでアーカイブを差分します。このコマンドは、追加、変更、削除されたファイルのリストを出力します。変更されたファイルについては、ファイルのメタデータ（モード、uid、gid、サイズなど）が考慮されます。ファイルの内容の変更の検出には、デフォルトでは mtime だけが使用されます。compare-content フラグが指定された場合、mtimeは無視され、ファイルの内容が比較されます。

<prev-スナップショット>

[<string>] 最初のスナップショットのパス。

<スナップショット>

[<string>] 2番目のスナップショットのパス。

<アーカイブ名> [<文字列>

[<string>] バックアップアーカイブ名。

オプションのパラメータ：

-color always|auto|never (default=auto)
カラー出力オプション

--compare-content <boolean> (default=false)
変更されたファイルの検出を mtime だけに頼るのではなく、ファイルの内容を比較します。

--keyfd <integer> (0 - N)
既にオープンされているファイル記述子を介して暗号化キーを渡します。

--keyfile <文字列>
暗号化キーへのパス。

--ns <文字列>
名前空間。

--リポジトリ <文字列>
リポジトリの URL。

proxmox-backup-debug help [{<コマンド>}]] [OPTIONS].

指定したコマンド（またはサブコマンド）に関するヘルプを表示します。

<コマンド>

[<string>] コマンド。入れ子になったサブコマンドを指定するため、リストにすることができます。複数指定可能。
。

オプションのパラメータ：

--verbose <ブール値>
詳細なヘルプを表示します。

proxmox-backup-debug inspect chunk <chunk> [OPTIONS] チャンクを検査します。

チャンクを検査します。

<chunk>

[<string> チャンクファイル。]

オプションのパラメータです：

--デコード <文字列>
チャンクをデコードするファイルへのパス、'-' は標準出力にデコードします。

-ダイジェスト <string>
リファレンスを検索する際に必要です。もし設定されていれば、デコード時の検証に使用されます。

--キーファイル <文字列>
チャンクが暗号化されたキーファイルへのパス。

--output-format text|json|json-pretty
出力形式。

--reference-filter <文字列>
参照を検索するディレクトリへのパス。

--use-filename-as-digest <boolean> (default=true)
ダイジェストが指定されていない場合、ファイル名をリファレンス検索とデコード検証のダイジェストとして使用します。

```
proxmox-backup-debug inspect device <device> [OPTIONS] (オプション)
```

デバイス上のデータストアを検査します。

<device>

[<string>] デバイスパス、通常は /dev/...

オプションのパラメータ

--出力フォーマット text|json|json-pretty

出力形式。

```
proxmox-backup-debug inspect file <file> [OPTIONS] ファイルを検査します。
```

デコードなしのblobファイルの場合、サイズと暗号化モードのみが表示されます。

<ファイル

[<string>] ファイルへのパス。

オプションのパラメータ

--デコード <文字列

デコード先のファイルへのパス。'-'は標準出力にデコードします。

--キーファイル <文字列

ファイルが暗号化されたキーファイルへのパス。

--output-format text|json|json-pretty

出力形式。

```
proxmox-backup-debug recover index <file> <chunks> [OPTIONS] インデックスファイルからデータを復元します。
```

チャンクが保存されているディレクトリ、インデックスファイル、復号に必要な場合はキーファイルを指定して、インデックスファイルからデータを復元します。

<file>

[<string>] インデックスファイル (.fidxまたは.didx)へのパス。

<chunks>

[<string>] チャンクを含むディレクトリへのパス。

オプションのパラメータ

--ignore-corrupt-chunks <論理値> (default=false)

チャンクが破損している場合、警告を表示し、代わりに0バイトを書き込み、部分的な復旧を試みます。

--ignore-missing-chunks <boolean> (default=false) チャンクが見つからない場合に警告。

チャンクが欠損している場合、警告を発し、代わりに0バイトを書き込み、部分的な復旧を試みます。

--keyfile <文字列

キーファイルへのパス。データが暗号化されている場合、復号化にはキーファイルが必要です。

--出力パス <文字列

出力ファイルのパス。デフォルトは拡張子なしのファイル。

--skip-crc <boolean> (default=false)

crc検証をスキップし、リストア速度を大幅に向上させます。

設定ファイル

Proxmox Backup Serverのすべての設定ファイルは/etc/proxmox-backup/ディレクトリにあります。

B.1 acl.cfg

B.1.1 ファイル形式

このファイルには Proxmox Backup Server API のアクセス制御リストが含まれます。各行はacl:で始まり、コロンで区切られた4つの追加値が続きます。

プロパゲート
パーミッションを階層下に伝播します。

パス
オブジェクトパス

ユーザー/トークン
ユーザーとトークンのリスト

ロール
割り当てられたロールのリスト

以下はリストの例です：

```
acl:1::root@pam!test:Admin acl:1:/datastore/store1:user1@pbs:DatastoreAdmin
```

このファイルを操作するにはproxmox-backup-manager aclコマンドを使用します。

B.1.2 ロール

以下のロールが存在します：

管理者
管理者

監査
監査人

アクセス禁止
アクセス禁止

データストア管理者
データストア管理者

データストア リーダー
データストア・リーダー (データストアの内容を検査し、リストアを実行します。)

データストアバックアップ
データストアのバックアップ (所有するバックアップのバックアップとリストア)

データストアPowerUser

Datastore PowerUser (所有バックアップのバックアップ、リストア、プルーニング)

データストア監査

データストア監査

リモート監査

リモート監査人

リモート管理者

リモート管理者

リモート同期オペレータ

同期オペレーター

リモート同期プッシュオペレータ

同期オペレーター (プッシュ方向)

リモートデータストアパワーユーザー

リモート・データストア・プルーン

リモートデータストア管理者

リモート・データストア管理者

テープ監査

テープ監査人

テープ管理者

テープ管理者

テープ操作者

テープ・オペレーター

テープ・リーダー

テープ・リーダー

B.2 データストア.cfg

B.2.1 ファイル形式

このファイルには、データストア設定セクションのリストが含まれています。各セクションはヘッダ

datastore: <name> で始まり、その後にデータストア設定オプションが続きます。

```
datastore: <name1>
  パス<path1>
  <option1><value1>
datastore: <name2>
  パス<path2>
```

このファイルを操作するにはproxmox-backup-manager datastoreコマンドを使用します。

B.2.2 オプション

必須プロパティ**パス**

[<string>] データストア・ディレクトリの絶対パス、またはリムーバブル・データストアのデバイス上の相対パスを指定します。

オプションのプロパティです：

バックエンド

[<backend-config>] データストアのバックエンド構成

キャッシングデバイス

[<string>] リムーバブル・データストアのファイルシステム・パーティションのUUID。

コメント

[<string>] コメント。

gc-schedule

[<calendar-event>] 指定したスケジュールでガベージコレクションジョブを実行します。

メンテナンスマード

[[<type=<enum> [,message=<string>]]] メンテナンスマード。] メンテナンスマード、タイプは'offline'または'read-only'、メッセージは""で囲んでください。

type= read-only|offline|delete|unmount|s3-refresh,
メンテナンスの種類。

メッセージ= <文字列>

メンテナンスの理由を説明するメッセージ。

通知モード

[/legacy-sendmail/notification-system (default=notification-system)] このデータストアの通知をどのように送信するかを設定します。

システムの *sendmail* 実行ファイルを介して *notify-user* で構成されたユーザに電子メール通知を送信します。*notification-system* 一致する通知イベントを通知システムに発行します。

notify

[[[gc=<enum> [,prune=<enum>] [,sync=<enum>] [,verify=<enum>]]]] データストア通知設定、enum は'always'、'never'、'error' のいずれかです。

gc= never|always|error
いつ通知を送るか

prune= never|always|error
いつ通知を送るか

sync= never|always|error
通知を送信するタイミング

verify= never|always|error
いつ通知を送るか

通知ユーザー

[<string>] ユーザーID

剪定スケジュール

[<calendar-event>] 指定したスケジュールでプルーンのジョブを実行します。

チューニング

[[[chunk-order=<enum>] [,gc-atime-cutoff=<integer>] [,gc-atime-safety-check=<1|0>] を指定します。[,gc-cache-capacity=<integer>] [,sync-level=<enum>]]] を参照してください。データストアのチューニング・オプション

chunk-order= none|inode (default=inode)
チャンクのソート順

gc-atime-cutoff= <integer> (1 - 2880) (default=1445)
ガベージコレクションフェーズ2でのチャンククリーンアップのatimeチェックのカットオフ（分単位）
(default=24h 5m)

gc-atime-safety-check= <boolean> (default=true)
ストア作成時およびガベージコレクション時にファイルシステムのatime更新をチェックします。

gc-cache-capacity= <整数> (0 - 8388608) (デフォルト=1048576)
ガベージコレクションチャンクダイジェストキャッシュ容量

sync-level= none|file|filesystem (default=filesystem)
データストアに書き込む際の同期レベル。

verify-new

[<boolean>] 有効にすると、すべての新規バックアップが完了直後に検証されます。

keep-daily

[<integer> (1 - N)] 毎日保持するバックアップの数。

キープアワーリー

[<integer> (1 - N)] 每時バックアップを保持する数。

キープラスト

[<integer> (1 - N)] 保持するバックアップの数。

keep-monthly

[<integer> (1 - N)] 毎月保持するバックアップの数。

keep-weekly

[<integer> (1 - N)] 毎週保持するバックアップの数。

年間保持

[<integer> (1 - N)] 年間バックアップの数。

B.3 ドメイン.cfg

B.3.1 ファイル形式

このファイルには認証領域のリストが含まれます。

各ユーザ設定セクションは、<realm-type>: <name> というヘッダで始まり、その後にレルムの設定オプションが続きます。

LDAP レルムの場合、LDAP バインドパスワードは ldap_password.json に格納されます。

```
openid: master
  クライアント ID pbs
  コメント
  issuer-url http://192.168.0.10:8080/realms/master
  username-claim ユーザー名
ldap: ldap-server
  base-dn OU=People,DC=ldap-server,DC=example,DC=com mode ldaps
    サーバー-1 192.168.0.10
    sync-attributes 電子メール=メール
    sync-defaults-options enable-new=0,remove-vanished=acl;entry
    ユーザー属性 uid
    ユーザークラス inetorgperson posixaccount person user
```

proxmox-backup-manager の openid、proxmox-backup-manager の ldap および

proxmox-backup-manager ad コマンドを使ってこのファイルを操作できます。

B.3.2 オプション

セクションタイプ 'ad' : AD レルム設定プロパティ。

必須プロパティです :

server1

[<string>] AD サーバアドレス

オプションのプロパティ :

base-dn

[<string>] LDAP ドメイン

バインド-dn

[<string>] LDAP ドメイン

capath

[<string>] サーバに使用する CA 証明書。パスはファイルかディレクトリを指します。ファイルを指す場合、そのパスに格納されている PEM 形式の X.509 証明書が信頼される証明書として追加されます。

を信頼できる証明書として追加します。パスがディレクトリを指す場合、そのディレクトリは /etc/ssl/certs にあるシステムのデフォルトの証明書ストアを置き換えます。

コメント

[<string>] コメント。

デフォルト

[<boolean> (default=false)] ログイン時に選択されるデフォルトのレルムにしたい場合は真。

filter

[<string>] ユーザ同期用のカスタムLDAP検索フィルタ

モード

[ldap|ldap+starttls|ldaps (default=ldap)] LDAP接続タイプ [<integer> (0 - 65535)] ADサーバ

ポート ポート**サーバー2**

[<string>] フォールバックADサーバーアドレス

同期属性

[[[email=<string>] [,firstname=<string>] [,lastname=<string>]]] PBS のどのユーザーフィールドにどの LDAP 属性をマップするかを指定するための key=value ペアのカンマ区切りリストです。例えば、LDAP 属性の mail を PBS の email にマッピングするには、email=mail と記述します。

email= <文字列>

ユーザーのメールアドレスを含む LDAP 属性の名前

firstname= <文字列>

ユーザーのファーストネームを含む LDAP 属性名

lastname= <文字列>

ユーザーの姓を含む LDAP 属性名

sync-defaults-options

[[[enable-new=<1|0>] [,remove-vanished=<string>]]] 同期デフォルトオプション

enable-new= <論理値>

同期後の新規ユーザーを有効にします

remove-vanished= [acl|entry|properties, ...]

セミコロンで区切られた、ユーザー同期中にユーザーまたはユーザーが消滅したときに削除するもののリストです。
。aclは、ユーザーが同期から返されなかったときにACLを削除します。
を削除します。

ユーザクラス

[<string>, ...] (default=inetorgperson, posixaccount, person, user)]. ユーザ同期で許可される objectClass 値のカンマ区切りリスト。例えば

に設定されている場合、ユーザー同期は、objectClass: person または objectClass: user のすべての LDAP エンティティを考慮します。

検証

[<boolean> (default=false)] サーバ証明書を検証します。

セクション・タイプ 'ldap' : LDAP 設定プロパティ。

必須プロパティ :

base-dn

[<string>] LDAP ドメイン

server1

[<string>] LDAP サーバーアドレス

ユーザ属性

[<string>] ユーザー名属性。LDAPへのuserIdをLDAP dnにマッピングするために使用します。

オプションのプロパティ：

bind-dn

[<string>] LDAP ドメイン

capath

[<string>] サーバに使用するCA証明書。パスはファイルかディレクトリを指すことができます。ファイルを指す場合は、そのパスに格納されている PEM 形式の X.509 証明書が追加されます。

を信頼される証明書としてロードします。パスがディレクトリを指している場合、そのディレクトリは /etc/ssl/certs にあるシステムのデフォルトの証明書ストアを置き換えます。

コメント

[<string>] コメント。

デフォルト

[<boolean> (default=false)] ログイン時に選択されるデフォルトのレルムにしたい場合は真。

フィルタ

[<string>] ユーザ同期用のカスタムLDAP検索フィルタ

モード

[ldap|ldap+starttls|ldaps (default=ldap)] LDAP接続タイプ [<integer> (0 - 65535)] ポート

ポート

サーバー2

[<string>] フォールバック LDAP サーバーアドレス

同期属性

[[[email=<string>] [,firstname=<string>] [,lastname=<string>]]] どのLDAP属性をどのPBSユーザーフィールドにマッピングするかを指定するためのキー=値のペアのカンマ区切りリストです。例えば、LDAP 属性の mail を PBS の email に対応させるには、email=mail と記述します。

電子メール= <文字列>

ユーザーのメールアドレスを含むLDAP属性名

firstname= <文字列>

ユーザーのファーストネームを含むLDAP属性名

lastname= <文字列>

ユーザーの姓を含むLDAP属性名

sync-defaults-options

[[[enable-new=<1|0>] [,remove-vanished=<string>]]] 同期デフォルトオプション

enable-new= <論理値>

同期後の新規ユーザーを有効にします

remove-vanished= [acl|エントリ|プロパティ, ...]

aclは、ユーザが同期から返されないときにACLを削除します。aclは、ユーザが同期から返されないときにACLを削除します。
を削除します。

ユーザクラス

[<string>, ...] (default=inetorgperson, posixaccount, person, user)]. ユーザ同期で許可される objectClass 値のカンマ区切りリスト。スタン

スでは、user-classes が person, user に設定されている場合、ユーザ同期では、objectClass が person, user であるすべての

objectClass: person または objectClass: user のすべての LDAP エンティティを考慮します。

検証

[<boolean> (default=false)] サーバ証明書を検証します。

セクションタイプ 'openid' : OpenID設定プロパティ。

必須プロパティ :

client-id

[<string>] OpenIDクライアントID

issuer-url

[<string>] OpenID発行者URL

オプションのプロパティです :

acr値

[[<string>, ...]] を指定します。OpenID ACRリスト

自動作成

[<boolean> (default=false)] ユーザが存在しない場合に自動的に作成します。

client-key

[<string>] OpenIDクライアントキー

コメント

[<string>] コメント。

デフォルト

[<boolean> (default=false)] ログイン時に選択されるデフォルトのレルムにしたい場合は真。

プロンプト

[<string>] OpenIDプロンプト。

スコープ

[[<string>, ...] (default=email profile)] OpenIDスコープリスト。OpenID スコープリスト

ユーザ名-クレーム

[<string>] この属性/claimの値を一意なユーザ名として使用します。一意性を保証するかどうかは ID プロバイダ次第です。OpenID の仕様では、Subject ('sub') が一意であることだけが保証されています。

ject('sub')が一意であることだけを保証しています。また、ユーザが自分でその属性を変更できないようにしてください！

セクションタイプ 'pam' : 組み込みの PAM レルム設定プロパティ。

必須のプロパティです :

タイプ

[pam|pbs|openid|ldap|ad] レルムのタイプ。

オプションのプロパティ :

コメント

[<string>] コメント。

default

[<boolean> (default=false)] ログイン時に選択されるデフォルトのレルムにしたい場合は真。

セクションタイプ 'pbs' : 組み込みのProxmox Backup Serverレルム設定プロパティです。

必須プロパティです :

タイプ

[pam|pbs|openid|ldap|ad] レルムのタイプ

オプションのプロパティ :

コメント

[<string>] コメント。

デフォルト

[<boolean> (default=false)] ログイン時に選択されるデフォルトのレルムにしたい場合は真。

B.4 メディアプール.cfg**B.4.1 ファイル形式**

各エントリは pool: <name> というヘッダで始まり、メディアプールの設定オプションが続きます。

```
pool: company1
    割り当て 常に保持 上書き
pool: ...
```

proxmox-tape pool コマンドを使ってこのファイルを操作することができます。

B.4.2 オプション

オプションのプロパティです：

アロケーション

[<string>] メディアの割り当てポリシー('continue'、'always'、またはカレンダーイベント)を設定します。

コメント

[<string>] コメント。

暗号化

[<string>] テープ暗号化キー指紋 (sha256)。

保持

[<string>] メディアの保持ポリシー ('overwrite'、'keep'、または期間)。

テンプレート

[<string>] メディアセットの命名テンプレート (strftime()時間フォーマット指定を含むことができます)。

B.5 node.cfg**B.5.1 オプション**

このファイルには以下のオプションが含まれます：

acme

このノードで使用する ACME アカウント。

acmedomain0

ACME ドメイン。

acmedomain1

ACME ドメイン。

acmedomain2

ACME ドメイン。

acmedomain3

ACME ドメイン。

acmedomain4

ACME ドメイン。

http-proxy

aptとサブスクリプションチェック用のプロキシを設定します。

email-from

通知を送信する予備の電子メール。

ciphers-tls-1.3

プロキシが使用する TLS 1.3 用の TLS 暗号のリスト。コロンで区切られ、優先度の降順 (<https://docs.openssl.org/master/man1/openssl-ciphers/>)。(変更を有効にするにはプロキシを再起動する必要があります)。

ciphers-tls-1.2

プロキシが使う TLS<= 1.2 の TLS 暗号のリスト。コロンで区切られ、優先度の降順 (<https://docs.openssl.org/master/man1/openssl-ciphers/>)。(変更を有効にするにはプロキシを再起動する必要があります)。

デフォルト言語

GUI で使われるデフォルトの言語。

description

ノードの説明。

task-log-max-days

タスクログを保存する最大日数。

例えば

```
acme: ローカル
acmedomain0: first.domain.com
acmedomain1: second.domain.com
acmedomain2: third.domain.com
acmedomain3: fourth.domain.com
acmedomain4: fifth.domain.com
http-proxy: internal.proxy.com email-
from: proxmox@mail.com
ciphers-tls-1.3: TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_8_SHA256:TLS_CHACHA20_POLY1305_SHA256
暗号-TLS-1.2 : RSA_WITH_AES_128_CCM:DHE_RSA_WITH_AES_128_CCM デフォルトの言語: ja
説明プライマリ PBS インスタンス task-log-max-
days : 30
```

`proxmox-backup-manager` ノードコマンドを使ってこのファイルを操作できます。

B.6 通知.cfg

B.6.1 ファイルフォーマット

このファイルには通知ターゲットと通知マッチャーの設定が含まれています。

B.6.2 オプション

セクションタイプ 'gotify' : Gotify 通知エンドポイントの設定

必須プロパティ

サーバ

[<string>] Gotify サーバー URL。

オプションのプロパティです：

コメント

[<string>] コメント。

無効

[<boolean>] このターゲットを無効にします。

フィルタ

[<string>] 非推奨。

オリジン

セクションタイプは 'matcher' です : Sendmail 通知エンドポイントの設定

オプションのプロパティ :

コメント

[<string>] コメント。

disable

[<boolean>] このマッチャを無効にします。

反転マッチ

[<boolean>] フィルタ全体のマッチを反転します。

マッチカレンダー

[<string>] マッチする重大度レベルのリスト。複数指定可能。

マッチフィールド

[<string>] 一致したメタデータ・フィールドのリスト。複数回指定することができます。

一致度

[<string>] 一致する重大度レベルのリスト。複数回指定可能。

モード

[all|any (default=all)] マッチの結果を結合するモード。

オリジン

[user-created|builtin|modified-builtin] 通知構成エントリのオリジンです。

ターゲット

[<string>] 通知するターゲット。複数指定できます。

セクションタイプ 'sendmail' : sendmail通知エンドポイントの設定

オプションのプロパティ :

作者

[<string>] メールの作成者です。デフォルトは 'Proxmox Backup Server (\$hostname)' です。

コメント

[<string>] コメント。

disable

[<boolean>] このターゲットを無効にします。

フィルタ

[<string>] 非推奨。

差出人アドレス

[<string>] 送信メールの差出人アドレス。このパラメータが設定されていない場合、プラグインはnode.cfg (PBS) のemail-from設定にフォールバックします。このパラメータも設定されていない場合、プラグインはデフォルトで root@\$hostname, \$hostnameはノードのホスト名です。

mailto

[<string>] メール送信先のメールアドレス。複数指定できます。

mailto-user

[<string>] メールを送信するユーザー。ユーザのメールアドレスはusers.cfgで検索されます。複数指定できます。

オリジン

[user-created|builtin|modified-builtin] 通知設定エントリのオリジン。

セクションタイプ 'smtp' : sendmail通知エンドポイントの設定

必須プロパティ

from-address
[<string>] メールの送信元アドレス。SMTPリレーでは、なりすましを避けるためにこのアドレスがユーザのものであることを要求する場合があります。メールのFromヘッダは\$author
<\$from-address>/になります。

サーバ
[<string>] SMTPリレーのホスト名またはIP。

オプションのプロパティ :

author
[<string>] メールの作成者です。デフォルトは 'Proxmox Backup Server (\$hostname)' です。

コメント
[<string>] コメント。

disable
[<boolean>] このターゲットを無効にします。

mailto
[<string>] 送信先のメールアドレス。複数指定可能。

mailto-user
[<string>] メールを送信するユーザ。ユーザのメールアドレスはusers.cfgで検索されます。複数指定可能。

モード
[insecure|starttls|tls (default=tls)] 接続セキュリティ

オリジン
[user-created|builtin|modified-builtin] 通知設定エントリのオリジン。

ポート
[<integer> (0 - 65535)] 接続先ポート。設定されていない場合、デフォルトは 25 (安全でない)、465 (tls)、または 587 (starttls) です。

ユーザ名
[<string>] 認証時に使用するユーザ名。ユーザ名が設定されていない場合、認証は行われません。PLAIN および LOGIN 認証方式をサポートしています。

セクションタイプ 'webhook' : Webhook通知エンドポイントの設定

必須プロパティ

メソッド
[post|put|get (default=post)] 使用するHTTPメソッド。

url
[<string>] HTTP(s)URLとオプションのポート。

オプションのプロパティ :

ボディ
[<string>] 送信するHTTPボディ。テンプレート化をサポートしています。

コメント
[<string>] コメント。

disable
[<boolean>] このターゲットを無効にします。

ヘッダ
[[name=<string> [,value=<string>]]] HTTPヘッダの配列。] HTTP ヘッダの配列。各エントリは、名前と値を持つプロパティ文字列です。value プロパティは、base64 エンコーディングのヘッダを含みます。
テンプレート化をサポートしています。複数回指定できます。

起源

[user-created|builtin|modified-builtin] 通知設定エントリのオリジン。

secret

`[[name=<string> [,value=<string>]]` 秘密の配列です。] 秘密の配列。各エントリは、名前とオプションの値を持つプロパティ文字列です。valueプロパティは、base64エンコードされたシークレットを含みます。

を返します。エンドポイント設定を返す API エンドポイントでは、シークレット名のみが返され、値は返されません。設定を更新する際には、保持したいすべてのシークレットも送信してください。テンプレートからアクセスできます。複数回指定できます。

B.7 通知-priv.cfg

B.7.1 ファイルフォーマット

このファイルは通知ターゲットのための保護された認証情報を含んでいます。

B.7.2 オプション

セクションタイプ 'gotify' : Gotify 通知エンドポイント用のプライベート設定です。この設定はより厳しいパーミッションで別の設定ファイルに保存されます(root:root 0600)

必須プロパティ

トークン

[<string>] 認証トークン

セクションタイプ 'smtp' : SMTP通知エンドポイントのプライベート設定。この設定はより厳しいパーミッションで別の設定ファイルに保存されます(root:root 0600)。

オプションのプロパティ

パスワード

[<string>] 認証時に使用するパスワード。

セクションタイプ 'webhook' : Webhook 通知エンドポイント用のプライベート設定。このコンフィグは、より厳しいパーミッション(root:root 0600)で別のコンフィグファイルに保存されます。

オプションのプロパティ

シークレット

`[[name=<string> [,value=<string>]]` シークレットの配列です。] 秘密の配列。各エントリは、名前と値のプロパティを持つプロパティ文字列です。valueプロパティには、base64エンコーディングの秘密が含まれます。

テンプレートからアクセスできます。複数回指定できます。

B.8 prune.cfg

B.8.1 ファイル形式

各エントリは `prune: <name>` というヘッダで始まり、ジョブの設定オプションが続きます。

```
prune: prune-store2
    スケジュール 月・金 10:30
    store my-datastore
prune: ...
```

proxmox-backup-managerのprune-jobコマンドを使ってこのファイルを操作できます。

B.8.2 オプション

必須プロパティです：

スケジュール

[<calendar-event>] 指定したスケジュールでpruneジョブを実行します。

ストア

[<string>] データストア名。

オプションのプロパティ：

コメント

[<string>] コメント。

disable

[<boolean> (default=false)] このジョブを無効にします。

最大深度

[<integer> (0 - 7)] 何レベルのネームスペースを操作するか (0 == 再帰なし、空== 自動完全再帰、ネームスペースの深さは最大許容値を減らします)

値)

ns

[<string>] 名前空間。

keep-daily

[<integer> (1 - N)] 毎日保持するバックアップの数。

keep-hourly

[<integer> (1 - N)] 每時バックアップを保持する数。

キープラスト

[<integer> (1 - N)] 保持するバックアップの数。

keep-monthly

[<integer> (1 - N)] 毎月保持するバックアップの数。

keep-weekly

[<integer> (1 - N)] 毎週保持するバックアップの数。

年単位

[<integer> (1 - N)] 年間バックアップを保持する数。

B.9 tape.cfg

B.9.1 ファイルフォーマット

各LTOドライブコンフィギュレーションセクションは、ヘッダlto: <name>で始まり、ドライブコンフィギュレーションオプションが続けます。

テープチェンジャーのコンフィギュレーションは、ヘッダ changer: <name> から始まり、その後にチェンジャーのコンフィギュレーションオプションが続けます。

```
lto: hh8
  チェンジャー sl3
  /バス /dev/tape/by-id/scsi-10WT065325-nst changer: sl3
  輸出スロット 14,15,16
  パス  /dev/tape/by-id/scsi-CJ0JBE0059
```

proxmox-tape drive と proxmox-tape changer コマンドを使ってこのファイルを操作することができます。

i 注意

仮想: ドライブタイプは実験的なものであり、デバッグにのみ使用してください。

B.9.2 オプション

セクション・タイプ 'changer' : SCSI テープ・チェンジャー

必須プロパティ

パス

[<string>] Linux汎用SCSIデバイスへのパス (例: '/dev/sg4')

オプションのプロパティ

アンロード前の取り出し

[<boolean> (default=false)] true に設定すると、アンロード前にテープを手動で排出します。

エクスポート・スロット

[[<integer>, ...]] スロット番号のリスト。カンマ区切りのスロット番号のリスト。これらのスロットはインポート/エクスポート用に予約されています。

セクションタイプ 'lto' : Lto SCSI テープドライバ

必須プロパティ

パス

[<string>] LTO SCSI 汎用テープデバイスへのパス (例: '/dev/sg0')

オプションのプロパティです:

changer

[<string>] テープチェンジャー識別子。

changer-drivernum

[<integer> (0 - 255) (default=0)] チェンジヤードライブ番号 (changer オプションが必要)

セクションタイプ 'virtual' : テープドライブをシミュレートします (テストとデバッグのみ)

必須プロパティ

パス

[<string>] ディレクトリへのパス

オプションのプロパティ

最大サイズ

[<integer> (0 - N)] 仮想テープサイズ

B.10 tape-job.cfg

B.10.1 ファイル形式

各エントリは backup:<name> というヘッダで始まり、ジョブの設定オプションが続きます。

```
バックアップ: job1
  ドライブ hh8
  ブールp4
  百六アダプタstore3
  スケジュール: ...
```

proxmox-tape backup-jobコマンドを使用してこのファイルを操作できます。

B.10.2 オプション

必須プロパティ

ドライブ

[<string>] ドライブ識別子。

プール

[<string>] メディアプール名。

ストア

[<string>] データストア名。

オプションのプロパティ：

コメント

[<string>] コメント。

スケジュール

[<calendar-event>] 指定したスケジュールで同期ジョブを実行します。

メディア取り出し

[<boolean>] ジョブの完了時にメディアを取り出します。

エクスポートメディアセット

[<boolean>] ジョブ完了時にメディアセットをエクスポートします。

グループフィルタ

[<exclude:>|<include:>]<type:<vm|ct|host>|group:GROUP|regex:RE>]を指定します。

グループ

フィルタのリスト

グループ

フィルタのリスト。複数回指定できます。

最新のみ

[<boolean>] 最新のスナップショットのみをバックアップします。

最大深度

[<integer> (0 - 7) (default=7)] 何レベルのネームスペースを操作するか (0== 再帰なし)。

通知モード

/legacy-sendmail/notification-system (default=notification-system)] このデータストアの通知をどのように送信するかを設定します。

システムの *sendmail* 実行ファイルを介して *notify-user* で構成されたユーザに電子メール通知を送信します。*notification-system* 通知システムに一致する通知イベントを発行します。

notify-user

[<string>] ユーザーID

ns

[<string>] 名前空間。

ワーカースレッド

[<integer> (1 - 32) (default=1)] ジョブに使用するワーカースレッド数を設定します。

B.11 user.cfg

B.11.1 ファイル形式

このファイルにはAPIユーザーとAPIトークンのリストが含まれています。

各ユーザ設定セクションは、ヘッダ `user: <name>` で始まり、ユーザ設定オプションが続きます。

APIトークンのコンフィギュレーションは、ヘッダ `token: <userid!token_name>` で始まり、トークンのコンフィギュレーションが続きます。トークンの認証に使用されるデータは、別のファイル (`token.shadow`) に保存されます。

```

user: root@pam
  コメント Superuser
  ()e()m@ltest@example.local
トークン: root@pam!token1
  コメント APIテスト・トークン
  有効
  有効期限0
ユーザー: ...

```

このファイルを操作するにはproxmox-backup-manager userコマンドを使用します。

B.11.2 オプション

セクションタイプ 'token' : ApiToken プロパティ。

オプションのプロパティ :

コメント

[<string>] コメント。

有効

[<boolean> (default=true)] アカウントを有効にします (デフォルト)。0'に設定するとアカウントが無効になります。

有効期限

[<integer> (0 - N) (default=0)] アカウントの有効期限 (エポックからの秒数)。0」は有効期限なしを意味します。

セクションタイプ 'user' : ユーザのプロパティ。

オプションのプロパティ :

コメント

[<string>] コメント。

電子メール

[<string>] メールアドレス。

有効

[<boolean> (default=true)] アカウントを有効にします (デフォルト)。0'に設定するとアカウントが無効になります。

期限切れ

[<integer> (0 - N) (default=0)] アカウントの有効期限 (エポックからの秒数)。0」は有効期限がないことを意味します。

名

[<string>] 名。

ラストネーム

[<string>] 姓。

B.12 リモート .cfg

B.12.1 ファイル形式

このファイルにはリモートサーバにアクセスするための情報が含まれています。

各エントリは remote: <name> というヘッダで始まり、リモート設定オプションが続きます。

```

remote: server1
  hostserver1.local
  ()u(.)t h-d sync@pbs
remote: ...

```

このファイルを操作するにはproxmox-backup-managerリモートコマンドを使用します。

B.12.2 オプション

必須プロパティ

パスワード

[<string>] リモートホストのパスワードまたは認証トークン（base64文字列として保存されます）。

auth-id

[<string>] 認証ID

ホスト [<string>] DNS名またはIPアドレス。

オプションのプロパティ：

コメント

[<string>] コメント。

フィンガープリント

[<string>] X509証明書のフィンガープリント(sha256)。

ポート [<integer>] (オプション) ポート。

B.13 sync.cfg

B.13.1 ファイル形式

各エントリは sync: <name> というヘッダで始まり、その後にジョブ設定オプションが続きます。

```
sync: job1
  ストア store1
  リモートストア store1
  (sync)r(e)m.otelina
```

このファイルを操作するには proxmox-backup-manager sync-job コマンドを使用します。

B.13.2 オプション

必須プロパティ

リモートストア

[<string>] データストア名。

ストア

[<string>] データストア名。

オプションのプロパティ：

コメント

[<string>] コメント。

暗号化のみ

[<boolean>] 暗号化されたバックアップスナップショットのみ同期し、他は除外します。

グループフィルタ

[<exclude:>|<include:>]<type:<vm|ct|host>|group:GROUP|regex:RE>]を指定します。

グループ

フィルタのリスト

グループ

フィルタのリスト。複数回指定できます。

最大深度

[<integer> (0 - 7)] 何レベルのネームスペースを操作するか (0 == 再帰なし、空== 自動完全再帰、ネームスペースの深さは最大許容値を減らす)

値)

ns

[<string>] 名前空間。

所有者

[<string>] 認証ID

リモート

[<string>] リモートID。

remote-ns

[<string>] 名前空間。

remove-vanished

[<boolean> (default=false)] 消去されたバックアップを削除します。リモートバックアップが削除された場合、ローカルコピーは削除されます。

再同期-破損

[<boolean>] ローカルのスナップショットで検証に失敗した場合、再試行します。

ランオンマウント

[<boolean>] 関連するデータストアがマウントされたときにこのジョブを実行します。

スケジュール

[<calendar-event>] 指定したスケジュールで同期ジョブを実行します。

同期方向

[pull|push (default=pull)] 同期ジョブの方向、プッシュまたはプル。

トランスファラスト

[<integer> (1 - N)] 転送を直近のN個のスナップショット（グループ毎）に制限し、他はスキップします。

検証のみ

[<boolean>] 検証済みのバックアップ・スナップショットのみを同期し、他は除外します。

バーストイン

[<string>] バイトサイズを任意の単位で指定します (B, KB (base 10), MB, GB, ..., KiB (base 2), MiB, Gib, ...)。

バーストアウト

[<string>] バイトサイズ。単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、Gib、...)。

レートイン

[<string>] バイトサイズ。単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、Gib、...)。

レートアウト

[<string>] バイトサイズ、単位は任意 (B、KB (base 10)、MB、GB、...、KiB (base 2)、MiB、Gib、...)。

B.14 検証.cfg

B.14.1 ファイル・フォーマット

各エントリは、`verification:<name>` というヘッダで始まり、ジョブの設定オプションが続きます。

```
verification: verify-store2 ignore-
  verified true
    outdated-after 7
      毎日スケジュール
      ストア store2
  verification: ...
```

このファイルを操作するには `proxmox-backup-manager verify-job` コマンドを使用します。

B.14.2 オプション

必須プロパティです：

ストア

[<string>] データストア名。

オプションのプロパティ

コメント

[<string>] コメント。

無視検証

[<boolean> (default=true)] 検証が古くない場合、検証済みのバックアップを検証しません。

最大深度

[<integer> (0 - 7) (default=7)] 何レベルのネームスペースを操作するか (0== 再帰なし)

ns

[<string>] 名前空間。

outdated-after

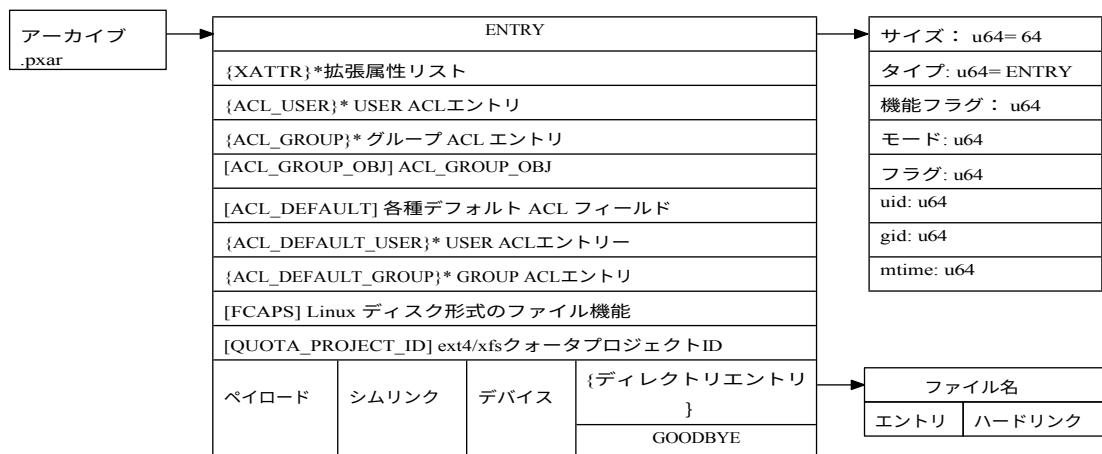
[<integer> (0 - N)] 検証が古くなる日数。(0は非推奨)

スケジュール

[<calendar-event>] 指定したスケジュールで検証ジョブを実行します。

ファイル形式

C.1 Proxmox ファイル・アーカイブ形式(.pxar)

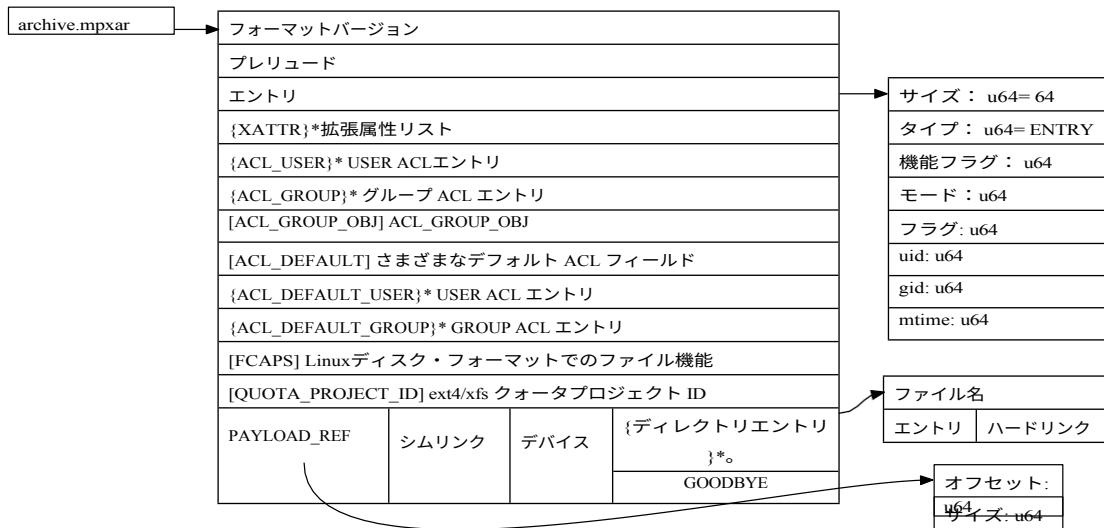


C.2 Proxmox ファイルアーカイブフォーマット - メタ (.mpxar)

Pxar メタデータアーカイブ。通常の pxar アーカイブと同じ構造ですが、regu-lar ファイルのペイロードはアーカイブ自体には含まれず、対応する pxar ペイロード (.ppxar) ファイルへのペイロードとして保存されます。

対応するpxarペイロード(.ppxar)ファイルへの参照。

ファイルペイロードによるサイズのオーバーヘッドなしに、すべてのアーカイブエントリとメタデータを検索するために使用することができます。



C.3 Proxmox ファイルアーカイブフォーマット -ペイロード (.ppxar)

対応する pxar メタデータ (.mpxar) アーカイブから参照およびアクセスされる、通常ファイルのペイロードを格納する Pxar ペイロードファイル。通常のファイルペイロードを連結したもので、それぞれの先頭に PAYLOAD ヘッダが付きます。さらに、実際に参照されるペイロードエントリは、パディングで区切られているかもしれません。

(これは、チャンクの境界がペイロードエントリーのオフセットと一致していない場合に、以前のバックアップ実行のチャンクを再利用するときに導入されます。

すべてのヘッダはリトルエンディアンで格納されます。

ペイロード開始	タイプハッシュとサイズからなる [u8; 16] のヘッダ。
PAYOUT	ヘッダは [u8; 16] で、ハッシュ型とサイズから構成されます。 メタデータアーカイブ
ペイロード	生の通常ファイルのペイロード
パディング	部分的/全体的に参照されないペイロード。 アリ
...	ペイロードヘッダ、ペイロード、パディングのさらなる連結
ペイロードテイルマーク	型ハッシュとサイズからなる [u8; 16] のヘッダ; マーク end

C.4 データ・プロブ・フォーマット (.blob)

データプロブフォーマットは小さなバイナリデータを格納するために使用されます。マジックナンバーが正確なフォーマットを決定します：

[66, 171, 56, 7, 190, 131, 112, 161]	非暗号化	非圧縮
[49, 185, 88, 66, 111, 182, 163, 127]	非暗号化	圧縮
[123, 103, 133, 190, 34, 45, 76, 240]	暗号化	非圧縮
[230, 89, 27, 191, 11, 191, 216, 11]	暗号化	圧縮

圧縮アルゴリズムはzstdを使用。暗号化暗号はAES_256_GCM。暗号化されていないblobは以下の形式を使用します：

MAGIC: [u8; 8]
CRC32: [u8; 4]
データ: (最大16MiB)

暗号化されたblobにはさらに、16バイトの初期化ベクトル（IV）、16バイトのauthenticated encryption（AE）タグ、暗号化データが続きます：

MAGIC: [u8; 8]
CRC32: [u8; 4]
IV: [u8; 16]
TAG: [u8; 16]
データ： (最大16MiB)

C.5 固定インデックスフォーマット (.fidx)

すべての数値はリトルエンディアンで格納されます。

マジック : [u8; 8]	[47, 127, 65, 237, 145, 253, 15, 205]
uuid: [u8; 16]、	ユニークID
ctime: i64、	作成時間 (エポック)
index_csum: [u8 ; 32],	SHA-256 オーバー その インデックス (を使用します ヘッダなし) 。 SHA256(ダイジェスト1 ダイジェスト2 ...)
size: u64、	画像サイズ
chunk_size: u64、	チャンクサイズ
reserved : [u8; 4016],	全体のヘッダーサイズは1ページ(4096バイト)
ダイジェスト1: [u8; 32]	最初のチャンクのダイジェスト
ダイジェスト2: [u8; 32]	2つ目のチャンクダイジェスト
...	次のチャンクダイジェスト ...

C.6 動的インデックスフォーマット (.didx)

すべての数値はリトルエンディアンで格納されます。

MAGIC: [u8; 8]	[28, 145, 78, 165, 25, 186, 179, 205]
uuid: [u8; 16]、	ユニークID
ctime: i64、	作成時間 (エポック)
index_csum :	SHA-256 オーバー その インデックス (を使用します。 ヘッダ) [u8; 32]、 SHA256(オフセット1 ダイジェスト1 オフセット2 ダイジェ スト2 ...)
は予約されています : [u8; 4032],	全体のヘッダーサイズは1ページ(4096バイト)
offset1: u64	最初のチャンクの終わり
digest1: [u8 ; 32]	最初のチャンクのダイジェスト
offset2: u64	2つ目のチャンクの終わり

digest2: [u8 ; セカンド・チャック・ダイジェスト
32]
... 次のチャックのオフセット/ダイジェスト

付録

バックアッププロトコル

Proxmox Backup ServerはRESTベースのAPIを使用しています。管理インターフェイスは通常のHTTPを使用しますが、実際のバックアップとリストアのインターフェイスはパフォーマンス向上のためにHTTP/2を使用します。HTTPとHTTP/2はどちらもよく知られた標準であるため、以下のセクションではこれらの使用方法を熟知していることを前提としています。

D.1 バックアッププロトコルAPI

新しいバックアップを開始するには、APIコールのGET /api2/json/backupをプロトコル名としてproxmox-backup-protocol-v1を使用するHTTP/2接続にアップグレードする必要があります：

```
GET /api2/json/backup HTTP/1.1 UPGRADE: proxmox-backup-protocol-v1
```

サーバはHTTP 101 Switching Protocolステータスコードで応答し、更新されたHTTP/2接続でRESTコマンドを発行できます。

バックアッププロトコルでは3種類のファイルをアップロードできます：

- ・ チャンクとプロブ（バイナリデータ）
- ・ 固定インデックス（固定サイズのチャンクのリスト）
- ・ 動的インデックス（サイズが可変のチャンクのリスト）

以下のセクションでは、このようなファイルをアップロードする方法について簡単に紹介します。利用可能なRESTコマンドの詳細については、API Viewerを使用してください。

D.1.1 プロブのアップロード

プロブはPOST /blobを使用してアップロードします。HTTPボディには、*Data Blob*としてエンコードされたデータが含まれます。ファイル名は**.blob**で終わる必要があり、POST /finishの呼び出しに続いてバックアップマニフェストに自動的に追加されます。

D.1.2 チャンクのアップロード

チャンクはインデックスに属するので、まずインデックスを開く必要があります（下記参照）。その後、POST /fixed_chunkおよびPOST /dynamic_chunkを使用してチャンクをアップロードできます。HTTPボディには、*Data Blob*としてエンコードされたチャンクデータが含まれます）。

D.1.3 固定インデックスのアップロード

固定インデックスはVMイメージデータの保存に使用されます。VMイメージは等しいサイズのチャンクに分割され、個別にアップロードされます。インデックスファイルには、チャンクダイジェストのリストが含まれます。

POST /fixed_indexで固定インデックスを作成します。そしてPOST /fixed_chunkでチャンクをアップロードし、PUT /fixed_indexでインデックスに追加します。終了したら、POST /fixed_close でインデックスを閉じる必要があります。

ファイル名は `.fidx` で終わる必要があり、`POST /finish` の呼び出しに続いてバックアップマニフェストに自動的に追加されます。

付録

D.1.4 ダイナミック インデックスのアップロード

ダイナミック インデックスは、ファイル アーカイブ データを格納するために使用されます。アーカイブ データは動的にサイズ調整されたチャンクに分割され、個別にアップロードされます。インデックスファイルには、チャンクのダイジェストとオフセットのリストが含まれます。

`POST /dynamic_index` で動的なサイズのインデックスを作成できます。次に、`POST /dynamic_chunk` でチャンクをアップロードし、`PUT /dynamic_index` でインデックスに追加します。終了したら、`POST /dynamic_close` でインデックスを閉じる必要があります。

ファイル名は `.didx` で終わる必要があり、`POST /finish` の呼び出しに続いてバックアップ・マニフェストに自動的に追加されます。

D.1.5 バックアップの終了

すべてのデータをアップロードしたら、`POST /finish` を呼び出す必要があります。これにより、すべてのデータがコミットされ、バックアッププロトコルが終了します。

D.2 リストア/リーダー プロトコル API

新しいリーダーを起動するには、APIコールの`GET /api2/json/reader`をプロトコル名として`proxmox-backup-reader-protocol-v1`を使用したHTTP/2接続にアップグレードする必要があります：

```
GET /api2/json/reader HTTP/1.1
アップグレード: proxmox-backup-reader-protocol-v1
```

サーバーはHTTP 101 Switching Protocolステータスコードで返信し、更新されたHTTP/2接続でRESTコマンドを発行できます。

リーダープロトコルでは、3種類のファイルをダウンロードできます：

- チャンクとプロブ（バイナリデータ）
- 固定インデックス（固定サイズのチャンクのリスト）
- 動的インデックス（サイズが可変のチャンクのリスト）

以下のセクションでは、このようなファイルをダウンロードする方法について簡単に紹介します。利用可能なRESTコマンドの詳細については、API Viewerを使用してください。

D.2.1 プロブのダウンロード

Blobは、`GET /download`を使用してダウンロードします。HTTPボディには、`Data Blob`としてエンコードされたデータが含まれます。

D.2.2 チャンクのダウンロード

チャンクは、`GET /chunk`を使用してダウンロードされます。HTTPボディには、`Data Blob`としてエンコードされたデータが含まれます。

D.2.3 インデックスファイルのダウンロード

インデックスファイルは、`GET /download`を使ってダウンロードします。HTTP本文にはとしてエンコードされたデータが含まれます。

カレンダーイベント

E.1 導入とフォーマット

特定のタスク、たとえばプルーニングやガベージコレクションは定期的に実行する必要があります。Proxmox Backup Serverは、[systemd Time and Date Specification](#) ([systemd.time manpage](#)を参照)にヒントを得たカレンダーイベントと呼ばれるフォーマットをスケジュールに使用します。

カレンダーイベントは1つ以上の時点を指定する式です。systemdのカレンダーイベントとほぼ互換性があります。

一般的なフォーマットは次のとおりです：

リスト1: カレンダーイベント

[平日] [[年-]月-日] [時間:分[:秒]]

少なくとも曜日、日付、時刻のいずれかの部分がなければならぬことに注意してください。曜日または日付の部分が省略された場合、すべての(週)日が含まれます。時刻の部分が省略された場合は、00:00:00が含まれます。(例：'2020-01-01'は'2020-01-01 00:00:00'を指します。)

平日は英語の省略形で指定します：**月**、**火**、**水**、**木**、**金**、**土**、**日**。各フィールドには以下の形式で複数の値を指定できます：

- カンマ区切り：例：01,02,03
- 範囲指定：例：01..10
- 繰り返し：例：05/10（10ごとに5から始まるという意味）
- 上記の組み合わせ：例：01,05...10,12/02
- または、すべての可能な値に対して*を付けることもできます：例：*:00

特別な意味を持つ値もあります：

値	構文
詳細	*-*-* *.*:00
毎時	*-*-* *:00:00
毎日	*-*-* 00:00:00
週間	月-*-* 00:00:00
マンスリー	*-*-* 01 00:00:00
毎年または毎年	*-01-01 00:00:00
四半期ごと	*-01,04,07,10-01 00:00:00
半年毎または半期毎	*-01,07-01 00:00:00

付録

便利な例を表にまとめました：

例	選択肢	説明
月、火、水、木	月・金	毎営業日00:00
土、日	土・日	週末のみ00:00
月・水・金	-	月、水、金 00:00
12:05		毎日午後12時5分
*:00/5	0/1:0/5	5分ごと
月・水	月、火、水	月・火・水 30分後、40分後、50分後
*:30/10	*:30/10	毎正時
月・金	-	毎営業日、午前8時から午後6時までは15分ごと
8.17,22:0/15		午後10時から午後11時まで
12日 (金) 13:5/20	金曜日12,13:5/20	金曜日12:05、12:25、12:45、13:05、13:25、13:45
12,14,16,18,20,	12/2:5	毎日12:05から22:05まで2時間毎
.	0/1:0/1	毎分（最小間隔）
*-05		毎月5日
土	*-1..7	毎月第1土曜日15:00
15:00		
2015-10-21	-	2015年10月21日 00:00

E.2 systemdとの違い

systemd カレンダーイベントのすべての機能が実装されているわけではありません：

- Unix タイムスタンプ (@12345 など) を使わない：代わりに日付と時刻を使って特定の時刻を指定します。
- タイムゾーンなし：すべてのスケジュールはサーバーのタイムゾーンを使います。
- 秒以下の解像度なし
- 逆日付構文なし (例: 2020-03~01)
- 範囲の繰り返しなし (例: 1..10/2)

E.3 スケジューリングに関する注意事項

Proxmox Backupでは、ほとんどのタスクのスケジューリングは`proxmox-backup-proxy`で行われます。このデーモンはすべてのジョブスケジュールを1分ごとにチェックし、期限切れのジョブがあるかどうかを確認します。つまり、カレンダイベントに秒数が含まれていても、チェックされるのは1分間に1回だけです。

また、すべてのスケジュールはProxmox Backup Serverで設定されたタイムゾーンと照合されます。

マークダウン入門

"MarkdownはウェブライターのためのテキストからHTMLへの変換ツールです。Markdownを使用することで、読みやすく、書きやすいプレーンテキスト形式を使って書くことができ、その後、構造的に有効な XHTML（またはHTML）に変換することができます。"

-John Gruber, <https://daringfireball.net/projects/markdown/>

Proxmox Backup Serverウェブインターフェースの "Notes" パネルは、Markdownテキストのレンダリングをサポートしています。

Proxmox Backup Serverは、テーブルやタスクリストのようなGFM（GitHub Flavoured Markdown）のほとんどの拡張機能でCommonMarkをサポートしています。

F.1 Markdownの基本

ここでは基本的なことしか説明していません。例えば、<https://www.markdownguide.org/>。

F.1.1 見出し

```
# これは見出します h1 ## これは見出します h2 ##### これは見出します h5
```

F.1.2 強調

強調には *text* または _text_ を使います。

太字のテキストには **text** または __text__ を使います。

組み合わせも可能です：

```
組み合わせも可能です。
```

F.1.3 リンク

リンクの自動検出を使うことができます。例えば、<https://forum.proxmox.com/>、クリック可能なリンクに変換します。

また、リンクテキストをコントロールすることもできます：

```
今、[カッコ内の部分がリンクテキストになります]([https://forum.proxmox.com/]).
```

F.1.4 リスト

順序なしリスト

順序なしリストには * または - を使用します：

- * 項目 1
- * 項目 2
- * 項目 2a
- * 項目 2b

インデントを追加することで、入れ子になったリストを作成できます。

順序付きリスト

1. 項目 1
1. 項目 2
1. 項目 3
 1. 項目 3a
 1. 項目 3b

注: 順番に並んだリストの整数は正しくなくともかまいません。

タスクリスト

タスクリストでは、未完了のタスクには空白のボックス [] を、完了したタスクには x のボックスを使用します。例えば

- [x] 最初のタスクは完了
- [x] 2つ目も
- [まだやることがあります]
- [これも]

F.15 テーブル

テーブルでは、パイプ記号|で列を区切り、-でテーブルのヘッダーと本文を区切れます。その区切りで、テキストの配置を設定し、1つの列を左揃え、中央揃え、または右揃えにすることもできます。

右寄せにすることもできます。

Left columns	Right columns	Some	More	Cols.	中央揃えも使えます。
左フー	右フー	最初の	行	ここ >center<	
左バー	右バー	二番目 行	ここ	12345	
左バズ	右バズ	番目 行	ここ	テスト	
左ザブ	右ザブ	第四 行	ここ	●●●	
左ラブ	右ラブ	そして	最後	ここ 終わり	

列を空白できれいに揃える必要はありませんが、その方が表の編集が簡単になることに注意してください。

F.16 ブロック引用符

プレーンテキストの電子メールと同様に、行の先頭に>、ブロック引用符を入力することができます。

```
>Markdownはプレーンテキスト形式の構文を持つ軽量のマークアップ言語です、
>2004年にJohn GruberがAaron Swartzとともに作成しました。
>>Markdownはしばしばreadmeファイルのフォーマット、オンライン・ディスカッション
>フーラム、
>>そして、プレーンテキストエディタを使用してリッチテキストを作成するために使用されます。
```

F.17 コードとスニペット

バックステイックを使って、単語や段落のグループを処理しないようにすることができます。これはコードや設定の塊が誤ってマークダウンとして解釈されるのを防ぐのに便利です。

インラインコード

行の一部をバックステイックで囲むと、例えばインラインでコードを書くことができます：

このホストのIPアドレスは `10.0.0.1` です。

コードブロック全体

数行に渡るコードブロックの場合、トリプルバックスラッシュでブロックの開始と終了を指定できます：

```
...
# これはここで覚えておきたいネットワーク設定です auto vmbr2

iface vmbr2 inet static
    address 10.0.0.1/24
    bridge-ports ens20
    bridge-stp off bridge-fd 0
    bridge-vlan-aware yes bridge-
    vids 2-4094

...
```


用語集

仮想マシン

仮想マシンは、エミュレートされたハードウェア環境内でオペレーティング・システム全体を実行できるプログラムです。

コンテナ

コンテナは、隔離されたユーザー空間です。プログラムはホストのカーネル上で直接実行されますが、ホストのリソースへのアクセスは制限されています。

データストア

バックアップを保存する場所。バックアップデータを格納するディレクトリ。現在の実装はファイルシステムベースです。

Rust

Rustは、高速でメモリ効率の良い新しいシステムプログラミング言語です。ランタイムもガベージコレクタもありません。Rustの豊富な型システムと所有権モデルは、メモリ安全性とスレッド安全性を保証します。これにより、多くのクラスのバグをコンパイル時に取り除くことができます。

Sphinx

インテリジェントで美しいフォーマットのドキュメントを簡単に作成できるツールです。もともとはPythonプログラミング言語のドキュメントのために作られました。様々な言語のソフトウェアプロジェクトのドキュメントを作成するための、優れた機能を備えています。

reStructuredText

読みやすく、見たままが得られる、プレーンテキストのマークアップ構文とパーサシステムです。

FUSE

Filesystem in Userspace ([FUSE](#)) は、ファイルシステムをカーネルで実装するのではなく、ユーザ空間で実装することを可能にするインターフェースを定義しています。FUSE カーネル・ドライバはファイルシステムのリクエストを処理し、ユーザ空間アプリケーションに送信します。

リモート

リモートのProxmox Backup Serverインストールとそのユーザの認証情報。冗長バックアップを行うために、リモートからローカルのデータストアにデータストアをプルすることができます。

GNUフリー・ドキュメント・ライセンス

バージョン 1.3, 2008 年 11 月 3 日

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.< <https://fsf.org/>>

すべての人に、このライセンス文書の逐語的な複製と頒布を許可します。

ただし、変更することは許されません。

0. 前文

本許諾書の目的は、マニュアルや教科書、あるいはその他の機能的で有用な 文書を、自由という意味で「自由」にすることです。つまり、商業的であれ非商業的であろうと、改変の有無に関わらず、誰もがそれを複製し再配布できる実効的な自由を保証することです。第二に、本許諾書は、作者や出版者に、他者による改変に責任を負わされることなく、自分たちの作品に対する謝意を得る方法を提供します。

このライセンスは一種の「コピーレフト」であり、この文書の派生物も同じ意味で自由でなければなりません。これは GNU一般公衆ライセンス(GNU General Public License)を補完するもので、自由ソフトウェアのために設計されたコピーレフトのライセンスです。

なぜなら、自由なソフトウェアには自由な文書が必要だからです。しかし、本許諾書はソフトウェアのマニュアルに限定されるものではありません。題材や印刷された書籍として出版されるかどうかに関わらず、あらゆるテキスト作品に利用することができます。私たちは、主に指導や参照を目的とした作品に本許諾書を推奨します。

1. 適用範囲と定義

本許諾書は、著作権者によって本許諾書の条項の下で頒布することができるという告知が含まれた、媒体を問わずあらゆるマニュアルやその他の作品に適用されます。このような告知は、本許諾書に記載された条件下でその著作物を使用するための、世界中、期間無制限、ロイヤリティフリーのライセンスを付与するものです。以下の「文書」とは、そのようなマニュアルや作品を指します。一般利用者はライセンサーであり、「あなた」として扱われます。あなたは、著作権法上の許可を必要とする方法で著作物を複製、変更、頒布する場合、本使用許諾に同意するものとします。

文書の「改変版」とは、文書またはその一部をそのまま、あるいは改変を加えて、および/または他の言語に翻訳して複製した著作物を意味します。

二次的セクション(Secondary Section)とは、その文書の出版者または著者と、その文書全体の主題との関係(または関連事項)のみを扱い、その文書全体の主題に直接含まれる可能性のあるものを含まない、名称の付録または文書の前文セクションのことです。(したがって、『文書』の一部が数学の教科書である場合、二次セクションでは数学について一切説明しないことができる)。その関係とは、その主題や関連事項との歴史的な関係、あるいはそれらに関する法律的、商業的、哲学的、倫理的、政治的な立場の問題です。

「変更不可能な節」とは、『文書』が本許諾書の下でリリースされたことを示す通知の中で、変更不可能な節であるとしてタイトルが指定されている、ある種の二次的な節のことです。もしもあるセクションが上記の「二次的セクション」の定義に当てはまらない場合、そのセクションを「変更不可セクション」として指定することは許されません。文書には『変更不可セクション』がゼロであってもかまいません。もし『文書』がいかなるInvariant Sectionも特定しない場合、そ

れはInvariant Sectionが存在しないことを意味します。

表紙テキスト」とは、『文書』が本許諾書の下でリリースされることを示す告知の中で、表紙テキストまたは裏表紙テキストとして列挙される特定の短い文章のことです。表表紙テキストは最大5語、裏表紙テキストは最大25語です。

透明な(Transparent)』『文書』のコピーとは、その仕様が一般公衆に利用可能な形式で表現され、一般的なテキストエディタや(ピクセルで構成される画像については)一般的なペイントプログラム、あるいは(図面については)広く利用可能な描画エディタを用いて文書を直接修正するのに適しており、テキストフォーマッタへの入力や、テキストフォーマッタへの入力に適した様々な形式への自動翻訳に適している、機械可読なコピーを意味します。そうでなければTransparentなファイルフォーマットで作成されたコピーで、そのマークアップ、あるいはマークアップの不在が、読者によるその後の改変を妨げたり、阻止したりするように配置されているものは、Transparentではありません。画像フォーマットは、かなりの量のテキストに使われる場合、「透過的」ではありません。透明』でないコピーは「不透明」と呼ばれます。

トランスペアレントコピーに適したフォーマットの例としては、マークアップのないプレーンなASCII、Texinfo入力フォーマット、LaTeX入力フォーマット、一般に公開されているDTDを使ったSGMLやXML、人間が修正できるように設計された標準準拠のシンプルなHTML、PostScript、PDFなどがあります。親を越える画像フォーマットの例としては、PNG、XCF、JPGがあります。不透明なフォーマットには、プロプライエタリなワードプロセッサーによってのみ読み取りおよび編集が可能なプロプライエタリなフォーマット、DTDおよび/または処理ツールが一般に利用可能でないSGMLまたはXML、および一部のワードプロセッサーが出力のみを目的として生成する機械生成のHTML、PostScriptまたはPDFが含まれます。

タイトルページ」とは、印刷された書籍の場合、タイトルページそれ自体と、本許諾書がタイトルページに掲載することを要求する素材を読みやすく保持するために必要な、それに続くページを意味します。そのようなタイトルページを持たない形式の著作物については、「タイトルページ」とは、本文の冒頭に先立つ、著作物のタイトルが最も目立つ位置にあるテキストを意味します。

発行者」とは、『文書』の複製物を公衆に頒布する個人または団体を意味します。

XYZと題された』セクションとは、そのタイトルが正確にXYZであるか、またはXYZを他の言語で翻訳したテキストの後に括弧で囲んでXYZを含む、『文書』の名前付き小部分を意味します。(ここでXYZは、「謝辞」、「献辞」、「裏書」、「歴史」など、後述の特定のセクション名を表します)。あなたが文書を変更したときに、そのようなセクションの"タイトルを保持する"ということは、そのセクションがこの定義に従って"XYZと題された"セクションのままであることを意味します。

文書』には、本許諾書が『文書』に適用されることを示す告示の隣に、保証の否認が含まれている場合があります。これらの保証の否認は本許諾書に参考として含まれるものとみなされますが、それは保証の否認に関するみです：これらの保証の否認が持つ他の暗示は無効であり、本許諾書の意味には何の影響も及ぼしません。

2. 逐語的複製

あなたは、本許諾書、著作権表示、および本許諾書が『文書』に適用されることを示す使用許諾表示がすべての複製物に複製され、かつ本許諾書の条件にいかなる他の条件も追加されないことを条件として、営利・非営利を問わず、『文書』をいかなる媒体にも複製および頒布することができます。あなたは、あなたが作成または頒布した複製物の閲覧やさらなる複製を妨害または制御する技術的手段を用いてはならない。しかし、あなたはコピーの対価として報酬を受け取ることができます。あなたが十分な数の複製物を頒布する場合、あなたは第3節の条件にも従わなければなりません。

あなたはまた、上記と同じ条件の下で、複製物を貸与することもできますし、複製物を公に展示することもできます。

3. 大量複製

あなたが『文書』の印刷された複製物(あるいは一般的に印刷された表紙を持つ媒体の複製物)を100部以上発行し、『文書』のライセンス告知がCover Texts(表紙文)を要求する場合、あなたはその複製物を、以下のCover Texts(表紙文)全て

を明瞭かつ読みやすく記載した表紙に封入しなければなりません：表表紙には「Front-Cover Texts」、裏表紙には「Back-Cover Texts」。また、両表紙とも、あなたがこれらのコピーの発行者であることを明確かつ判読しやすいように表示してください。表表紙は、タイトルの全単語を等しく目立たせ、見えるようにしなければなりません。また、表紙に他の素材を加えてもかまいません。表紙に限定して変更を加えた複製は、『文書』の題名を保持し、これらの条件を満たす限り、その他の点では逐語的複製として扱うことができます。

いづれかの表紙に必要な文章が多すぎて読みやすく収まらない場合は、最初に挙げたもの（合理的に収まる数）を実際の表紙に載せ、残りは隣のページに続けて載せてください。

あなたが『文書』の『不透明な複製物』を100部以上発行または頒布する場合、あなたは各『不透明な複製物』に機械可読の『透過的な複製物』を同梱するか、あるいは各『不透明な複製物』に、一般のネットワーク利用者が公衆標準ネットワークプロトコルを用いて『文書』の完全な『透過的な複製物』(追加資料のないもの)をダウンロードできるコンピュータネットワーク上の場所を明記しなければなりません。後者の選択肢を使用する場合、あなたは『不透明なコピー』の配布を大量に開始する際に、この『透過的なコピー』が、あなたがその版の『不透明なコピー』を(直接、あるいはあなたの代理人や小売業者を通じて)公衆に配布した最後の時点から少なくとも1年後まで、指定された場所でこのようにアクセス可能であり続けることを保証するために、合理的に慎重な手段を講じなければなりません。

大量のコピーを再配布する前に、『文書』の著者によく連絡し、あなたに『文書』の最新版を提供する機会を与えることが要求されますが、必須ではありません。

4. 変更

あなたは、上記第2項および第3項の条件の下で、『文書』の改変された版を複製し頒布することができます。ただし、改変された版をまさに本許諾書の下でリリースし、改変された版が『文書』の役割を果たし、その結果、改変された版の頒布と改変がそのコピーを所持する誰に対しても許諾されることを条件とします。さらに、あなたは改変されたバージョンにおいて以下のことを行わなければなりません：

- A. タイトルページ(および表紙(もしあれば))には、『文書』のタイトルとも、以前のバージョンのタイトル(もしあれば、『文書』の「履歴」セクションに記載されているはずです)とも異なるタイトルを使用してください。旧版の発行元が許可している場合は、旧版と同じタイトルを使用してもかまいません。
- B. タイトルページには、『文書』の主要な著者のうち少なくとも5名(主要な著者の数が5名未満の場合はその全員)とともに、改変された『バージョン』における改変の著者に責任を負う1名以上の個人または団体を、著者として記載してください(ただし、彼らがこの要件からあなたを解放する場合を除きます)。
- C. タイトルページには、修正版の発行者として発行者の名前を明記してください。
- D. 文書のすべての著作権表示を保持します。
- E. 他の著作権表示に隣接して、あなたの改変に対する適切な著作権表示を追加してください。
- F. 著作権表示の直後に、以下の補遺に示された形式で、本許諾書の条項の下で改変されたバージョンの利用を公衆に許可する利用許諾表示を含めてください。
- G. そのライセンス告知には、『文書』のライセンス告知に示されている、変更不可セクションの完全なリストと要求されるカバーテキストを保持してください。
- H. 本許諾書の変更されていない複製物を同梱してください。
- I. 「History」と題されたセクションを保全し、そのタイトルを保全し、少なくともタイトルページに記載されている改変版のタイトル、年、新著者、および発行者を記載した項目をそのセクションに追加してください。もし『文書』に「沿革」と題された項がない場合、そのタイトルページに記載されているように、『文書』のタイトル、年、著者、発行者を記述した項を作成し、前文で述べられているように、改変されたバージョンについて記述した項を追加してください。
- J. 文書の透過的なコピーを公開するために、その文書で指定されているネットワークの場所(もしあれば)を保存し、同様に、その文書の基になった旧版の文書で指定されているネットワークの場所も保存してください。これらは「履歴」セクションに置くことができます。あなたは、『文書』自身より少なくとも4年前に出版された著作物、あるいはその著作物が参照する版の原出版者が許可した場合、その著作物のネットワーク上の位置を省略することができます。

- K. 謝辞」または「献辞」と題されたセクションについては、そのセクションのタイトルを保持し、そのセクションで述べられている各寄稿者の謝辞および/または献辞の内容および論調をすべて保持してください。

- L. 文書のすべての不变セクションを、そのテキストもタイトルも変更せずに保存します。セクション番号またはそれに相当するものは、セクションタイトルの一部とは見なされません。
- M. 「裏書」というタイトルのセクションを削除してください。このようなセクションは、修正版に含めることはできません。
- N. 既存のセクションのタイトルを「エンドースメント」に変更したり、変更不可のセクションとタイトルが矛盾するように変更したりしないでください。
- O. 保証の免責条項はすべて保存してください。

変更後のバージョン』に、『二次的著作物』(Secondary Section)として適格であり、『文書』からコピーされた素材を含まない、新しい前文セクションや付録が含まれる場合、あなたは任意でこれらのセクションの一部または全部を変更不可のセクションとして指定することができます。これを行うには、改変された版のライセンス告知にある「変更不可のセクション」のリストに、それらのセクションのタイトルを追加してください。これらのタイトルは、他のいかなるセクションのタイトルとも異なるものでなければなりません。

Endorsements』と題されたセクションを追加することもできますが、そのセクションには、様々な当事者によるあなたの改変版の推奨--例えば、査読の記述や、テキストが標準の権威ある定義としてある組織によって承認されたこと--しか含まれません。

表紙テキストとして5語までの文章を、裏表紙テキストとして25語までの文章を、修正版の表紙テキストリストの最後に追加することができます。表表紙テキストと裏表紙テキストは、それぞれ1つの主体によって（または主体による取り決めによって）1つの箇所のみ追加することができます。もし『文書』に同じ表紙のカバーテキストが既に含まれており、それが以前にあなたによって追加されたものである場合、あるいはあなたが代行する同じ団体によって手配されたものである場合、あなたは別のカバーテキストを追加することはできません。

文書の作者および発行者は、本許諾書によって、その名前を宣伝のために使用すること、あるいは改変された版の支持を主張したり示唆したりすることを許可しないものとします。

5. 文書の結合

あなたは、本許諾書の下でリリースされた他の文書と『文書』を、上記第4項で改変版について定義された条件の下で結合することができる。ただし、その結合において、元の文書のすべての変更不可部分を変更せずに含み、それらをすべてあなたの結合著作物の変更不可部分としてそのライセンス告知に記載し、かつそれらのすべての保証の否認を保持することを条件とする。

結合された著作物には本許諾書の複製物が一部含まれていればよく、複数の同一の『変更不可部分』は一つの複製物で置き換えることができます。同じ名前で異なる内容の複数の変更不可部分が存在する場合、そのような部分のタイトルの末尾に、その部分の原著作者または発行者の名前(既知であれば)、あるいは一意の番号を括弧で囲んで追加することで、そのような部分を一意にしてください。結合著作物のライセンス告知にある、変更不可の節の一覧の節のタイトルにも、同じ調整を加えてください。

同様に、「謝辞」(Acknowledgements)と題された節と「献辞」(Dedications)と題された節を結合してください。Endorsements "と題されたセクションはすべて削除してください。

6. 文書のコレクション

あなたは、『文書』や本許諾書の下でリリースされたその他の文書から成るコレクションを作成し、様々な文書に含まれる本許諾書の個々のコピーを、コレクションに含まれる単一のコピーに置き換えることができます。

あなたは、そのような文書集から一つの文書を抜き出し、本許諾書の下で個々に頒布することができます。ただし、抜き出した文書に本許諾書のコピーを挿入し、その文書の逐語的コピーに関する他のすべての点において本許諾書に従うことを条件とします。

7. 独立した著作物との集成

文書』またはその派生物と、他の独立した文書または著作物とを、一巻の記憶媒体または頒布媒体に編集したものを「編集物」と呼びます。

編集物から生じる著作権が、個々の著作物が許容する範囲を超えて編集物の利用者の法的権利を制限するために利用されない場合。文書』が総集編に含まれる場合、本許諾書は総集編に含まれる他の著作物であって、それ自体が『文書』の二次的著作物ではないものには適用されない。

第3項のカバーテキストの要件が当該文書のコピーに適用される場合、当該文書が総体全体の2分の1未満であれば、当該文書のカバーテキストは、総体内で当該文書を囲むカバー、または当該文書が電子形式であればカバーに相当する電子表紙に掲載することができます。それ以外の場合は、総体全体を囲む印刷された表紙に掲載しなければなりません。

8. 翻訳

翻訳は一種の改変とみなされますので、あなたは第4節の条件のもとで、この文書の翻訳を頒布することができます。変更不可部分.Invariant Section)を翻訳に置き換えるにはその著作権者から特別な許可を得る必要がありますが、あなたはこれらの変更不可部分のオリジナル版に加えて、一部または全部の変更不可部分の翻訳を含めることができます。あなたは、本許諾書、『文書』中のすべてのライセンス表示、および保証の否認の翻訳を含めることができます。ただし、本許諾書の英語原文、およびそれらの表示と否認の原文も含めることができます。翻訳と本許諾書の原文、あるいは通知や免責事項の原文との間に不一致がある場合、原文が優先されます。

文書』中のあるセクションのタイトルが「謝辞」、「献辞」、または「歴史」である場合、そのタイトルを保持するための再要件(第4項)は(第1項)、通常、実際のタイトルを変更する必要があります。

9. 終了

あなたは、本許諾書の下で明示的に規定されている場合を除き、『文書』を複製、変更、サブライセンス、または頒布することはできません。それ以外の方法で複製、変更、サブライセンス、または頒布しようとする行為は無効であり、本使用許諾に基づくあなたの権利は自動的に消滅します。

ただし、あなたが本許諾書に対する違反をすべて止めた場合、特定の著作権者からのあなたのライセンスは、(a)著作権者があなたのライセンスを明示的かつ最終的に抹消しない限り、暫定的に、(b)著作権者が違反の停止後60日以前に何らかの合理的な手段によってあなたに違反を通知しなかった場合には、永久的に復活するものとします。

さらに、複製権者が何らかの合理的な手段であなたに違反を通知し、あなたがその著作権者から本許諾書に対する違反の通知を(いかなる作品についても)初めて受け取り、あなたがその通知を受け取ってから30日以前に違反を是正した場合、特定の著作権者からのあなたのライセンスは永続的に復活します。

本節に基づくあなたの権利の終了は、本許諾書に基づいてあなたから複製物や権利を受領した当事者のライセンスを終了させるものではありません。あなたの権利が終了し、恒久的に復活しない場合、同じ資料の一部または全部の複製物を受け取ったとしても、それを使用する権利はあなたに与えられません。

10. 本許諾書の将来の改訂

フリーソフトウェアファンデーションは、GNU自由文書利用許諾書の新しい改訂版を隨時公表することができるものとします。そのような新バージョンは、現在のバージョンと精神的には類似していますが、新たな問題や懸念に対処するために詳細が異なるかもしれません。<https://www.gnu.org/licenses/>をご覧ください。

本許諾書の各バージョンには、バージョン番号が付けられています。文書』において、本許諾書の特定のバージョン番号「またはそれ以降のバージョン」が適用されると指定されている場合、あなたはその指定されたバージョンか、フリーソフトウェア財団によって公表された(草案としてではない)それ以降のバージョンのいずれかの条項と条件に従うという選択肢を持つことになります。文書』に本許諾書のバージョン番号が明記されていない場合、あなたはフリーソフトウェア財団によって(草案としてではなく)公表されたどのバージョンでも選ぶことができます。文書』において、代理人が本許諾書の将来のどの版を使用できるかを決定できることを指定している場合、その代理人がある版を受諾すると宣言することで、あなたは『文書』においてその版を選択することが永久に許可されることになります。

11. 再許諾

「大規模マルチ・オーサー・コラボレーション・サイト」(Massive Multiauthor Collaboration Site)(または「MMCサイト」)とは、著作権で保護される著作物を公開し、かつ誰でもそれらの著作物を編集することができる著名な設備を提供するワールド・ワイド・ウェブ・サーバを意味します。

を意味します。誰でも編集できる公開ウィキは、そのようなサーバの一例です。サイトに含まれる「Massive Multiauthor Collaboration」(または「MMC」)とは、このようにしてMMCサイトで公表された著作権保護可能な作品群を意味します。

「CC-BY-SA」とは、カリフォルニア州サンフランシスコに主たる事業所を置く非営利法人クリエイティブ・コモンズ・コーポレーション (Creative Commons Corporation) により発行されたクリエイティブ・コモンズ表示-継承3.0ライセンス、及び同団体により発行された同ライセンスの将来のコピーレフト版を意味します。

「取り込む」とは、文書の全部または一部を他の文書の一部として発行または再出版することを意味します。

MMCが「再許諾の資格がある」のは、それが本許諾書の下でライセンスされており、本許諾書の下で本MMC以外のどこかで最初に公表され、その後にMMCに全部あるいは一部が組み込まれたすべての作品が、(1)カバーテキストや不变部分がなく、(2)こうして2008年11月1日より前に組み込まれた場合です。

MMCサイトの運営者は、2009年8月1日以前であればいつでも、CC-BY-SAに基づき、同サイトに含まれるMMCを再公開することができます。

索引

C

コンテナ、[281](#)

D

データストア、[281](#)

F

FUSE, [281](#)

R

リモート、[281](#)

reStructuredText, [281](#)

Rust, [281](#)

S

Sphinx, [281](#)

V

仮想 マシン、[281](#)