



Proxmoxバックアップドキュメント

リリース 3.4.0-1

Proxmoxサポートチーム

2025年 4月 09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	6
1.6.4 バグトラッカー	6
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 Supported Web Browsers for Accessing the Web Interface	8
2.2 Installation Medium	8
2.2.1 インストール・メディアの準備	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 Instructions for macOS	9
2.2.5 Windows	10
2.2.6 インストール・メディアの使用	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。	11
2.3.2 Proxmox Backup Serverを無人インストールします。	19
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をDebianにインストールします。	20
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	20
2.5 Debian Package Repositories	20
2.5.1 SecureApt	21
2.5.2 Proxmox Backup Enterprise リポジトリ	22
2.5.3 Proxmox Backup サブスクリプションなしリポジトリ	22
2.5.4 Proxmox バックアップテストリポジトリ	22
2.5.5 Proxmox バックアップ クライアント専用リポジトリ	23
2.5.6 Repository Access Behind HTTP Proxy	24
3 用語解説	25
3.1 バックアップ・コンテンツ	25

3.1.1	イメージアーカイブ: <名前>.img .	25
3.1.2	ファイルアーカイブ: <名前>.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 .	26
3.1.5	マニフェスト: index.json .	26
3.2	バックアップ 名前空間 .	26
3.3	バックア タイプ	26
3.4	バックア ID	26
3.5	バックア 時間	26
3.6	バックア グループ	27
3.7	バックア スナップショット .	27
4	グラフィカル・ユーザー・インターフェース	29
4.1	特徴	29
4.2	ログイン .	29
	4.2.1 Consent Banner	30
4.3	GUI Overview	30
4.4	サイドバー .	31
	4.4.1 ダッシュボード .	31
	4.4.2 コンフィグレーション	31
	4.4.3 Administration	32
	4.4.4 テープバックアップ	34
	4.4.5 データストア .	35
5	バックアップ・ストレージ	37
5.1	ディスク管理	37
5.2	データストア .	39
	5.2.1 データストアの構成	40
	5.2.2 バックアップ名前空間	43
	5.2.3 オプション	44
5.3	ランサムウェアの保護と復旧	45
	5.3.1 Built-in Protection	46
	5.3.2 The 3-2-1 Rule with Proxmox Backup Server	46
	5.3.3 制限的なユーザーとアクセス管理	46
	5.3.4 ランサムウェア検出	47
	5.3.5 一般的な予防方法とベストプラクティス	47
6	ユーザー管理	49
6.1	User Configuration	49
6.2	APIトークン .	51
6.3	Access Control	52
	6.3.1 Privileges	52
	6.3.2 アクセス・ロール	54
	6.3.3 Objects and Paths	55
	6.3.4 構成と管理	55
	6.3.5 APIトークン権限	56
	6.3.6 有効なパーミッション	56
6.4	二要素認証	57
	6.4.1 Introduction	57
	6.4.2 Available Second Factors	57
	6.4.3 Setup	57
	6.4.4 TFA and Automated Access	59
	6.4.5 二要素認証の制限とロックアウト	59
6.5	認証領域	59
	6.5.1 Linux PAM .	59

6.5.2	Proxmox Backup authentication server	59
6.5.3	LDAP	60
6.5.4	Active Directory	60

7	バックアップクライアントの使用状況	63
7.1	バックアップ・リポジトリの場所.....	63
7.2	環境変数.....	64
7.3	システムおよびサービスのクレデンシャル.....	64
7.4	出力フォーマット.....	65
7.5	バックアップの作成.....	65
7.5.1	バックアップからのファイル/ディレクトリの除外.....	66
7.5.2	変更検出モード.....	68
7.6	暗号化.....	68
7.6.1	マスター・キーの使用による暗号化キーの保存と復元.....	69
7.7	データの復元.....	70
7.7.1	インタラクティブ・リストア.....	71
7.7.2	FUSEによるアーカイブのマウント.....	71
7.8	ログインとログアウト.....	72
7.9	バックアップグループの所有者の変更.....	72
7.10	バックアップの剪定と削除.....	72
7.11	ゴミ収集.....	74
7.12	ベンチマー킹.....	74
8	Proxmox VEの統合	77
8.1	Proxmox VEウェブインターフェースの使用.....	77
8.2	Proxmox VEコマンドラインの使用法.....	77
9	pxarコマンドラインツール	79
9.1	アーカイブの作成.....	79
9.2	アーカイブの抽出.....	80
9.3	アーカイブの内容一覧.....	80
9.4	アーカイブのマウント.....	80
10	テープバックアップ	81
10.1	テープ技術入門.....	82
10.2	対応ハードウェア.....	82
10.2.1	ドライブ性能.....	82
10.2.2	LTO-9+の考察.....	83
10.3	用語解説.....	83
10.4	テープクイックスタート.....	84
10.5	構成.....	84
10.5.1	テープチェンジャー.....	84
10.5.2	テープドライブ.....	87
10.5.3	メディア・プール.....	89
10.5.4	テープバックアップの仕事.....	92
10.6	管理部門.....	94
10.6.1	ラベルテープ.....	94
10.6.2	テープバックアップの実行.....	95
10.6.3	テープからのリストア.....	95
10.6.4	在庫の更新.....	96
10.6.5	カタログの復元.....	96

10.6.6 暗号鍵の管理	97
10.6.7 テープクリーニング	98
10.7 WORMテープ	99
10.8 セットアップ例	99
10.8.1 シングル継続メディアセット	99
10.8.2 平日スキーム	99
10.8.3 異なるポリシーを持つ複数のプール	100
10.9 災害復旧	100
10.9.1 データストアの設定	100
10.9.2 テープ・ドライブのセットアップ	101
10.9.3 テープからのデータの復元	101
11 リモコンと同期の管理	103
11.1 リモート	103
11.2 シンク・ジョブズ	104
11.2.1 名前空間サポート	105
11.2.2 帯域制限	106
11.2.3 同期方向プッシュ	106
12 メンテナスタスク	109
12.1 剪定	109
12.1.1 ブルーンシミュレータ	110
12.1.2 ブルーンの求人	110
12.1.3 マニュアル剪定	111
12.1.4 保持設定例	112
12.2 ゴミ収集	112
12.2.1 GCの背景	112
12.2.2 GCの手動起動	113
12.2.3 GC予定	113
12.3 検証	114
12.4 お知らせ	114
12.5 メンテナスマード	114
13 ホストシステム管理	115
13.1 Linux上のZFS	115
13.1.1 ハードウェア	116
13.1.2 ZFS管理	116
13.2 ホストブートローダー	121
13.2.1 インストーラーが使用するパーティション方式	121
13.2.2 proxmox-boot-toolを使ったESPの内容の同期	122
13.2.3 使用されるブートローダの決定	124
13.2.4 グラブ	126
13.2.5 システムブート	126
13.2.6 カーネル・コマンドラインの編集	126
13.2.7 次のブート用にカーネルバージョンを上書き	127
13.2.8 セキュアブート	128
13.3 証明書管理	130

13.3.1 APIとSMTPの証明書.....	130
13.3.2 カスタム証明書のアップロード	131
13.3.3 Let's Encrypt (ACME) による信頼できる証明書.....	131
13.3.4 ACME HTTPチャレンジプラグイン.....	133
13.3.5 ACME DNS APIチャレンジプラグイン	133
13.3.6 ACME証明書の自動更新	134
13.3.7 コマンドラインでの証明書の手動変更	134
13.4 外部メトリックサーバー.....	135
13.4.1 InfluxDB (HTTP) プラグインの設定.....	135
13.4.2 InfluxDB (UDP) プラグインの設定	135
13.5 サービスデーモン	136
13.5.1 プロックスマックスバックアッププロキシ	136
13.5.2 プロックスマックスバックアップ	136
13.6 コマンドラインツール	136
13.6.1 プロックスマックスバックアップクライアント	136
13.6.2 プロックスマックスバックアップマネージャ	136
13.6.3 プロックスマックスステープ	136
13.6.4 pmt	136
13.6.5 pmtn	136
13.6.6 pxar	137
13.6.7 プロックスマックスファイルリストア	138
13.6.8 proxmox-backup-debug.....	138
14 ネットワーク管理	141
14.1 交通管制	143
15 お知らせ	145
15.1 概要	145
15.2 通知対象	145
15.2.1 センドメール	145
15.2.2 SMTP	146
15.2.3 ゴティファイ	146
15.2.4 ウェブフック	146
15.3 通知マッチャー	148
15.3.1 カレンダーの一致ルール	148
15.3.2 フィールドマッチング規則	148
15.3.3 深刻度マッチングルール	149
15.4 通知イベント	149
15.5 システムメール転送	150
15.6 アクセス許可	150
15.7 通知モード	150
15.7.1 レガシー送信メール通知モードの設定	151
15.8 通知テンプレートのオーバーライド	151
16 技術概要	153
16.1 データストア	153
16.2 スナップ写真	153
16.3 チャンクス	153

16.3.1 固定サイズのチャンク	154
16.3.2 動的なサイズのチャンク	154
16.3.3 暗号化されたチャンク	155
16.3.4 ファイルベース・バックアップの変更検出モード	155
16.4 警告と限界	157
16.4.1 ハッシュ衝突に関するメモ	157
16.4.2 ファイルベースのバックアップ	157
16.4.3 暗号化されたチャンクの検証	157
16.5 ブラブルシューティング	157
16.5.1 Proxmox Backup Serverが稼働していない状態でのリストア	158
17 よくあるご質問	159
17.1 Proxmox Backup Server (PBS) はどのいますか?	159
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?	159
17.3 Proxmox Backup Serverは32ビットプロセッサで動作しますか?	159
17.4 Proxmox Backup Serverバージョンのサポート期間は?	159
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。	159
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を教えてください。	160
17.7 データストアを別のコピーまたは同期できますか?	160
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?	160
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?	160
17.10 バックアップは増分/重複/フルですか?	161
A コマンド構文	163
A.1 ブロックスマックスバックアップクライアント	163
A.1.1 カタログシェルコマンド	175
A.2 proxmox-backup-manager	177
A.3 ブロックスマックスステープ	224
A.4 pmt	239
A.5 pmtx	242
A.6 pxar	243
A.7 ブロックスマックスファイルリストア	245
A.8 proxmox-backup-debug	247
B 設定ファイル	251
B.1 acl.cfg	251
B.1.1 ファイル形式	251
B.1.2 役割	251
B.2 datastore.cfg	252
B.2.1 ファイル形式	252
B.2.2 オプション	253
B.3 ドメイン.cfg	254
B.3.1 ファイル形式	254
B.3.2 オプション	255
B.4 メディアプール	258
B.4.1 ファイル形式	258
B.4.2 オプション	258
B.5 node.cfg	259
B.5.1 オプション	259

B.6	通知.cfg.....	260
B.6.1	ファイル形式.....	260
B.6.2	オプション	260
B.7	通知非公開.cfg	262
B.7.1	ファイル形式.....	262
B.7.2	オプション	263
B.8	ブルーン	263
B.8.1	ファイル形式.....	263
B.8.2	オプション	263
B.9	tape.cfg	264
B.9.1	ファイル形式.....	264
B.9.2	オプション	264
B.10	tape-job.cfg	265
B.10.1	ファイル形式.....	265
B.10.2	オプション	265
B.11	user.cfg	266
B.11.1	ファイル形式.....	266
B.11.2	オプション	267
B.12	リモート.cfg.....	267
B.12.1	ファイル形式.....	267
B.12.2	オプション	268
B.13	同期.cfg.....	268
B.13.1	ファイル形式.....	268
B.13.2	オプション	268
B.14	検証.cfg.....	269
B.14.1	ファイル形式.....	269
B.14.2	オプション	270
C	ファイル形式	271
C.1	Proxmox ファイルアーカイブ形式 (.pxar).....	271
C.2	Proxmox ファイルアーカイブ形式 - メタ (.mpxar).....	271
C.3	Proxmox ファイルアーカイブ形式 - ペイロード (.ppxar).....	272
C.4	データblob形式 (.blob)	272
C.5	固定インデックス形式 (.fidx).....	273
C.6	動的インデックス形式 (.didx).....	273
D	バックアップ・プロトコル	275
D.1	バックアッププロトコルAPI	275
D.1.1	ブロブのアップロード	275
D.1.2	チャunkのアップロード	275
D.1.3	固定インデックスのアップロード	276
D.1.4	動的インデックスのアップロード	276
D.1.5	バックアップ完了	276
D.2	リストア/リーダープロトコルAPI	276
D.2.1	ダウンロード Blobs	277
D.2.2	ダウンロード Chunks	277

D.2.3 インデックスファイルのダウンロード	277
E カレンダー・イベント	279
E.1 序論と形式	279
E.2 systemd との違い	280
E.3 スケジューリングに関する注意事項	280
F マークダウン入門	281
F.1 マークダウンの基本	281
F.1.1 見出し	281
F.1.2 強調	281
F.1.3 リンク	282
F.1.4 リスト	282
F.1.5 テーブル	282
F.1.6 ブロック名言集	283
F.1.7 コードとスニペット	283
G 用語集	285
H GNU自由文書ライセンス	287
索引	293

Copyright (C) 2019-2025, Proxmox Server Solutions GmbH Version 3.4.0 -- 2025年

04月09日(水)

GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. ライセンスのコピーは「GNU Free Documentation License」の項に含まれています。

はじめに

1.1 Proxmox Backup Serverとは何ですか?

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ーバックアップソリューションです。[Proxmox Virtual Environment](#)プラットフォーム用に特別に最適化されており、リモートサイト間でも安全にデータをバックアップすることができます。

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が保証されます。

Proxmox Backupは、クライアント-サーバ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 建築

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](#) のために、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。たとえば、[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格納されます。仮想マシンの設定自体は追加ファイルとして保存されます。こうすることで、バックアップ全体をスキャンすることなく、バックアップの重要な部分のみにアクセスしてリストアすることが容易になります。

1.3 主な特長

Proxmox VEのサポート

Proxmox仮想環境は完全にサポートされています。

仮想マシンとコンテナ

パフォーマンス

ソフトウェアスタック全体は[Rust](#)で書かれており、高速性とメモリ効率を実現しています。

重複排除

定期的なバックアップは大量の重複生成します。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、バックアップのストレージと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は、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

遠隔地とのデータ同期を効率的に行うことができます。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することができます。

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提供します。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。

ランサムウェア対策

Proxmox Backup Serverのきめ細かいアクセス制御、データ完全性の検証、リ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。

企業サポート

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 Repositoryにアクセスできます。さらに、ベーシック、スタンダード、またはプレミアムの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は、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、アクセスできるように、法律により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。そのため、定期的なバックアップとリストアテストの実施をください。

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
- サーバーの管理CLIツール(*proxmox-backup-manager*)
- 任意のLinux *amd64*環境からサーバーに簡単にアクセスできるクライアントCLIツール(*proxmox-backup-client*)

Web インターフェイスを除き、Proxmox Backup Server のほとんどの部分は Rust プログラミング言語で記述されています。

"Rust" プログラミング言語は、より高速で信頼性の高い書くのに役立ちます。高レベルの人間工学と低レベルの制御は、プログラミング言語ではしばしば対立。Rustは、強力な技術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、低レベルの詳細（メモリ使用量など）を、従来のような煩わしさを伴わずに制御するオプションを提供します。"

-ラスト・プログラミング言語

1.6 ヘルプ

1.6.1 企業サポート

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ーは、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。詳細情報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](#)を使用して、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！開発者との主な連絡手段はこちらです：

開発者向けメーリングリスト
[Proxmoxバックアップサーバ開発リスト](#)

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。Free Software Foundation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 General Public License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 沿革

バックアップは、これまで、そしてこれからも、IT管理の中心的な侧面です。データ損失からの復旧の必要性は基本的なものであり、仮想化によってさらに高まります。

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で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。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、バックアップ期間もストレージ使用量もこのケースでは高すぎます。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、高速性とメモリ実現するために、実装言語として[Rust](#)を使用することに決めました。Rustの2018年版は、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、セキュアな環境を提供し、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を実行する場合は、高品質のサーバー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さらに軽減するために、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できます。

2.1.1 評価用サーバーの最小要件

これらの最小要件は評価のみを目的としたものであり、製造に使用するものではありません。

- 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4ギガバイト。TiBストレージスペースごとに少なくとももう1GiB追加します。
- OSストレージ:
 - 32GB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) 付きのハードウェアRAIDを使用するか、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選択します。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をする場合: メタデータキャッシュを使用することを強くお勧めします。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 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

サーバーのウェブ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、当年のリリース、または最新の拡張サポートリリース
- クローム、今年からのリリース
- マイクロソフトが現在サポートしているEdgeのバージョン
- サファリ、今年からのリリース

2.2 設置媒体

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、対話型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 インストール媒体の準備

<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てください。

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

- DVDに書き込める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)にコピー可能な生セクタ(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

フラッシュドライブには少なくとも2GBの記憶容量が必要です。

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、まだ必要なデータが含まれていないことを確認し、次に進む前に再度アンマウントしてください。

2.2.3 GNU/Linux用説明書

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`コマンド出力の最終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ます：

```
# lsblk
```

その後、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てください：

```
# 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の説明

ターミナルを開きます。

`hdiutil`の`convert`、`.iso`ファイルを`.dmg`形式に変換します：

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

注意：macOSは出力ファイル自動的に`.dmg`を付ける傾向があります。

現在のデバイスのリストを取得するには、次のコマンドを実行します：

```
# 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
```

注：Xは前回のコマンドのディスク番号に置き換えてください。

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

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

2.2.5 Windows用説明書

エッチャーの使用

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。

ルーファスの使用

Rufusはより軽量な代替品ですが、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選択します。

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 "をクリックする必要があります。次のダイアログで**DD**モードを選択します。

2.2.6 インストール媒体を使用

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 サーバー設置

バックアップサーバー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様々な管理タスクのためのウェブベースの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。サーバー部分なしでProxmox Backupを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。

インストールメディアを作成すると、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設定、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を行い、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に代わる方法として、上級ユーザは Proxmox Backup Server を無人でインストールすることができます。

十分な Debian の知識があれば、Proxmox Backup Server を [Debian 上に](#)インストールすることもできます。あなたの自身です。

推奨しませんが、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 をダウンロードしてください。以下のものが含まれています：

- Proxmox Backup Serverインストーラは、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します。
- 完全なオペレーティングシステム (Debian Linux, 64-bit)
- ZFSをサポートした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

注意: 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。インストーラーは、他のオペレーティングシステムのブートメニューントリーを追加しません。

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、起動してください。

注: 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場合があります。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。



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

通常のインストールを開始します。

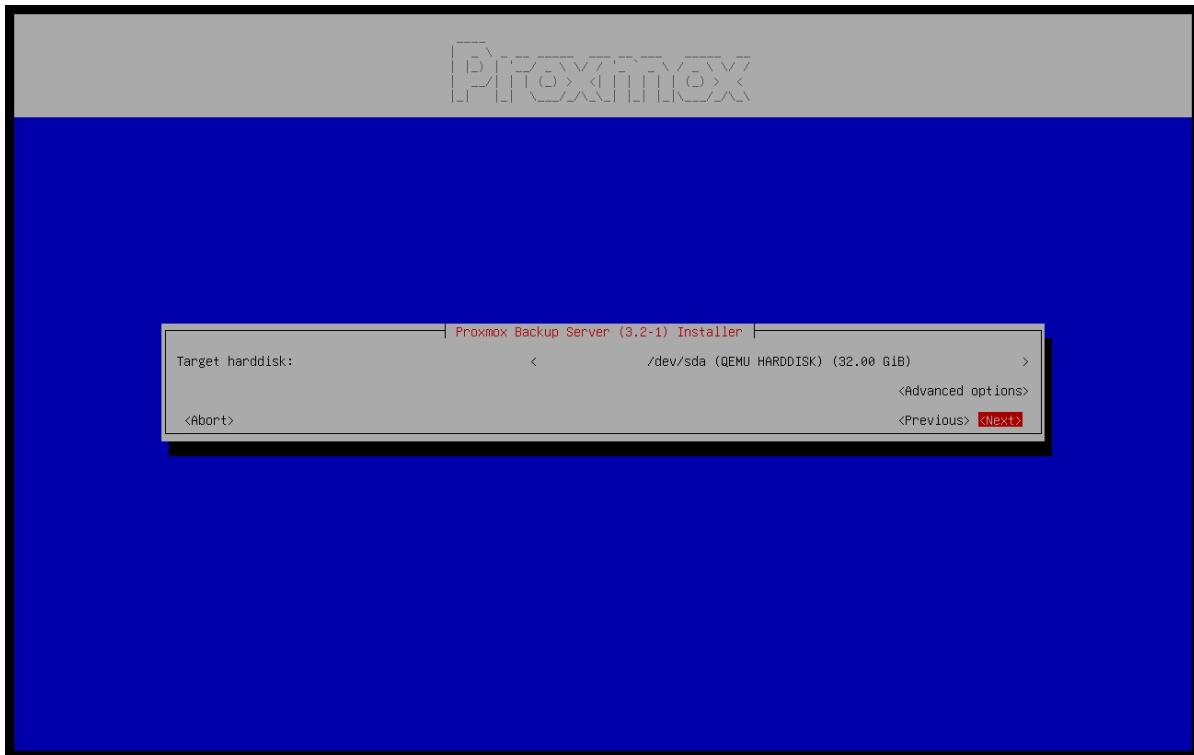
ヒント: インストールウィザードをキーボードだけで使用することができます。ボタンは、ALT キーと各ボタンの下線文字を組み合わせて押すことでクリックできます。例えば、「次へ」ボタンを押すにはALT + Nです。

Proxmox Backup Server (コンソール)のインストール

ターミナルモードのインストールウィザードを起動します。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起動し、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使用可能なライブシステムを起動できます。たとえば、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する場合に使用できます。

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。

ターミナル・ベースのデバッグ・モードと同じですが、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うように設定します。

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。接続されているすべ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ると、ISO から 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+**)

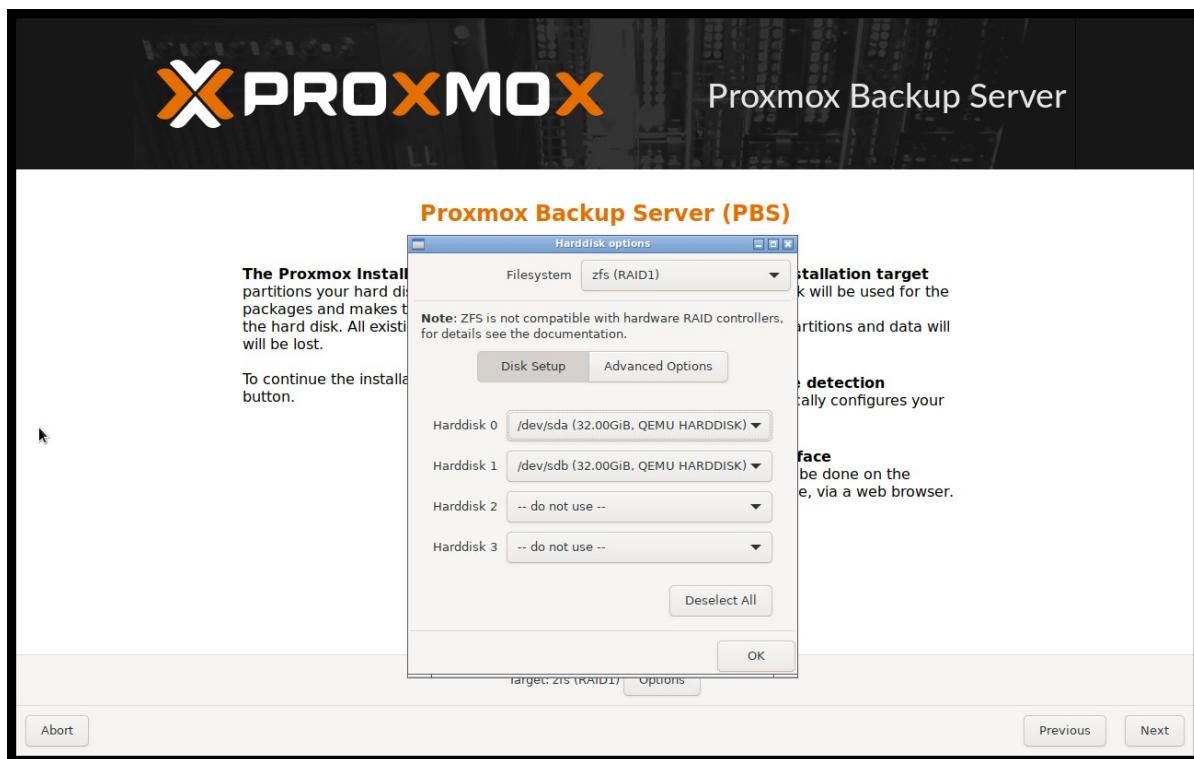
memtest86+ を実行します。メモリが機能しているか、エラーがないかをチェックするのに便利です。このオプションを実行するには、UEFI フームウェアセットアップユーティリティで Secure Boot をオフにする必要があります。

通常、*Proxmox Backup Server (Graphical)* のインストール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。

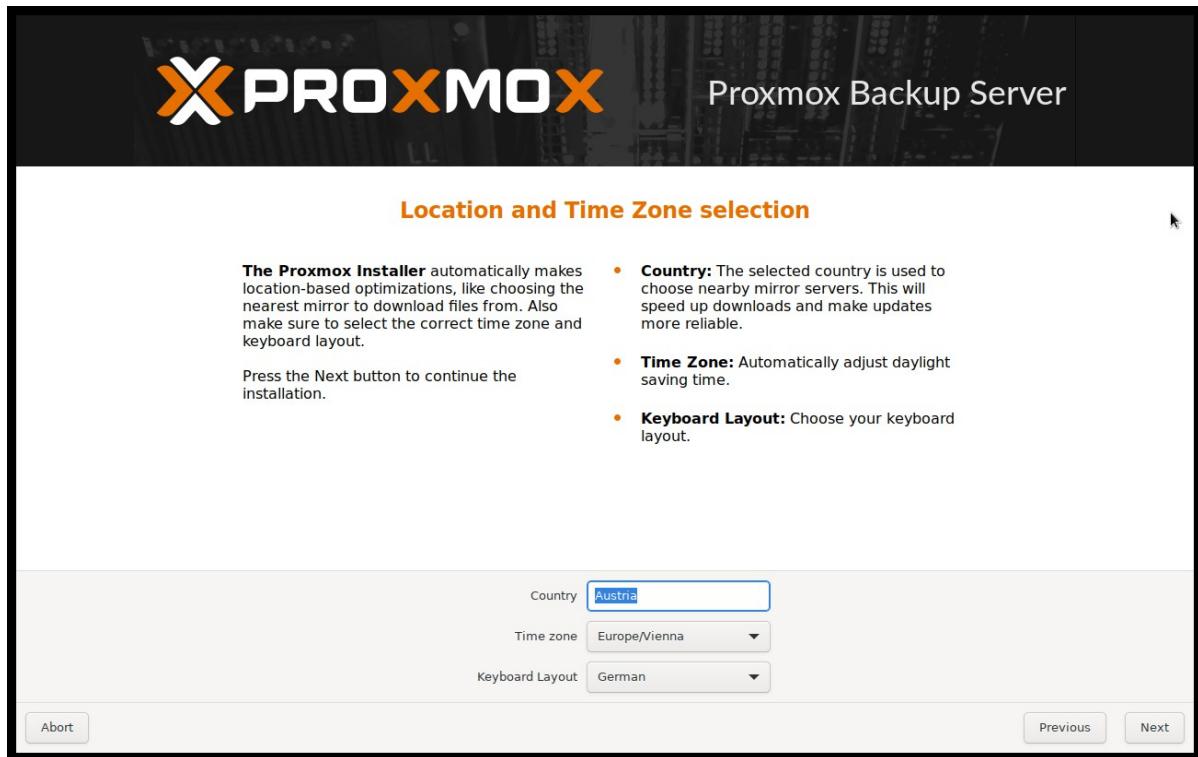
注意デフォルトでは、サーバー全体が使用され、既存のデータはすべて削除されます。インストールを続行する前に、サーバー上に重要なデータがないことを確認してください。

オプション]ボタンで、ターゲットファイルシステムを選択できます。デフォルトは ext4 です。ファイルシステムとして ext4 または xfs を選択した、インストーラは LVM を使用し、LVM 領域を制限する追加オプションを提供します (下記参照)。

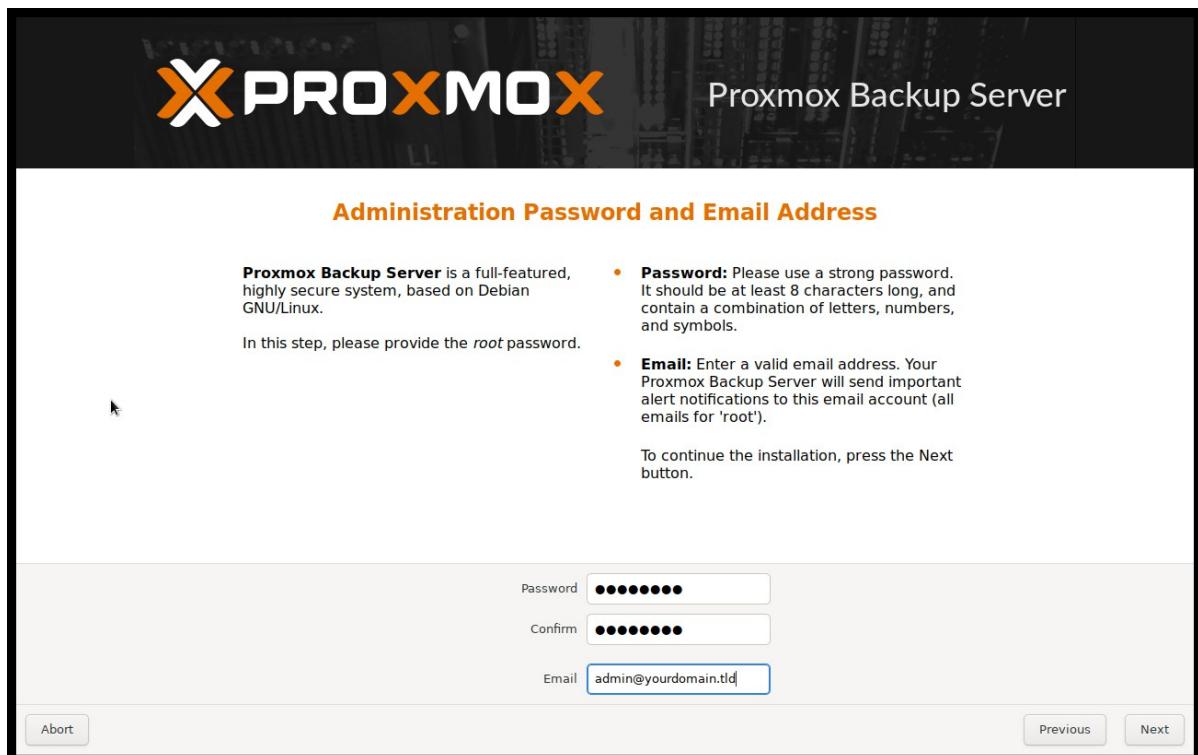


Proxmox Backup ServerはZFSにインストールすることもできます。ZFSにはいくつかのソフトウェアRAIDレベルが用意されているため、ハードウェアRAIDコントローラを搭載していないシステム向けのオプションです。オプション]ダイアログでターゲットディスクを選択する必要があります。ZFS固有の設定は、詳細オプションで変更できます。

警告ハードウェアRAID上のZFSはサポートされていません。



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るので、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。

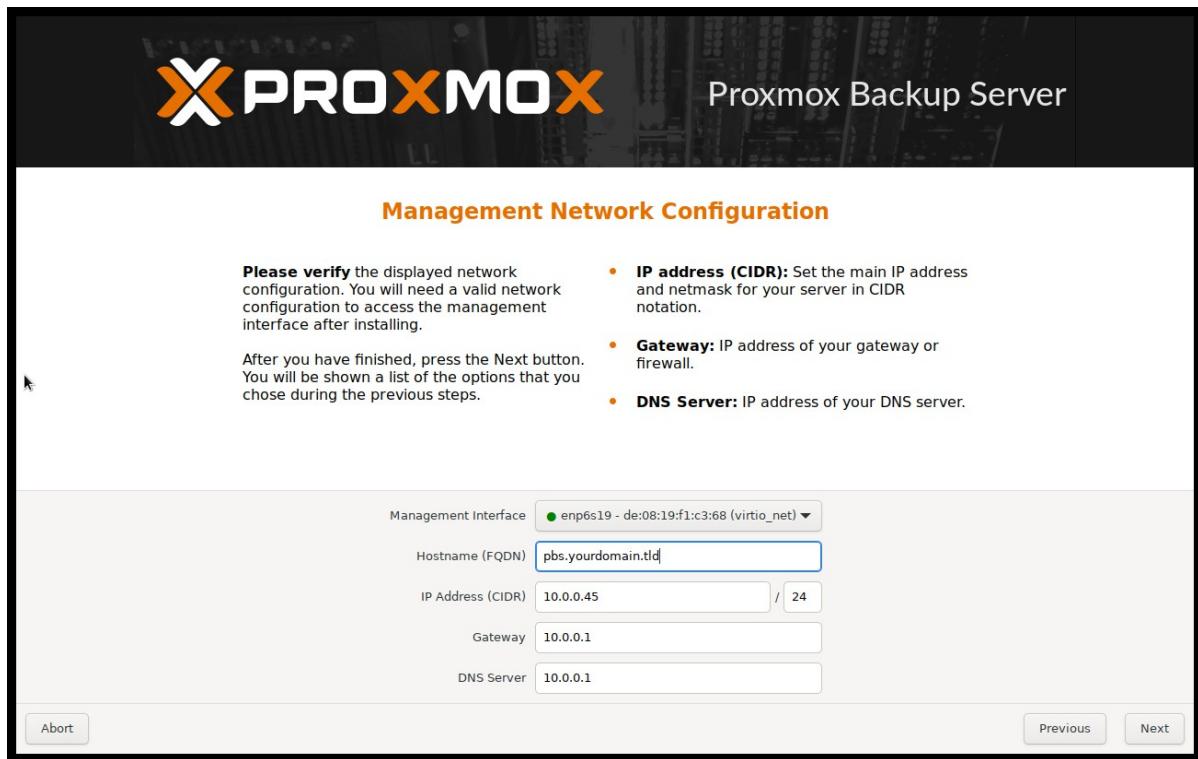


次に、スーパーユーザー (root) のパスワードとメールアドレスを指定する必要があります。パスワードは少なくとも8文字で構成されていなければなりません。より強力なパスワードを使用することを強くお勧めします。いくつかのガイドラインがあります：

- パスワードの長さは最低12以上にしてください。
- 小文字、大文字のアルファベット、数字、記号。
- 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。

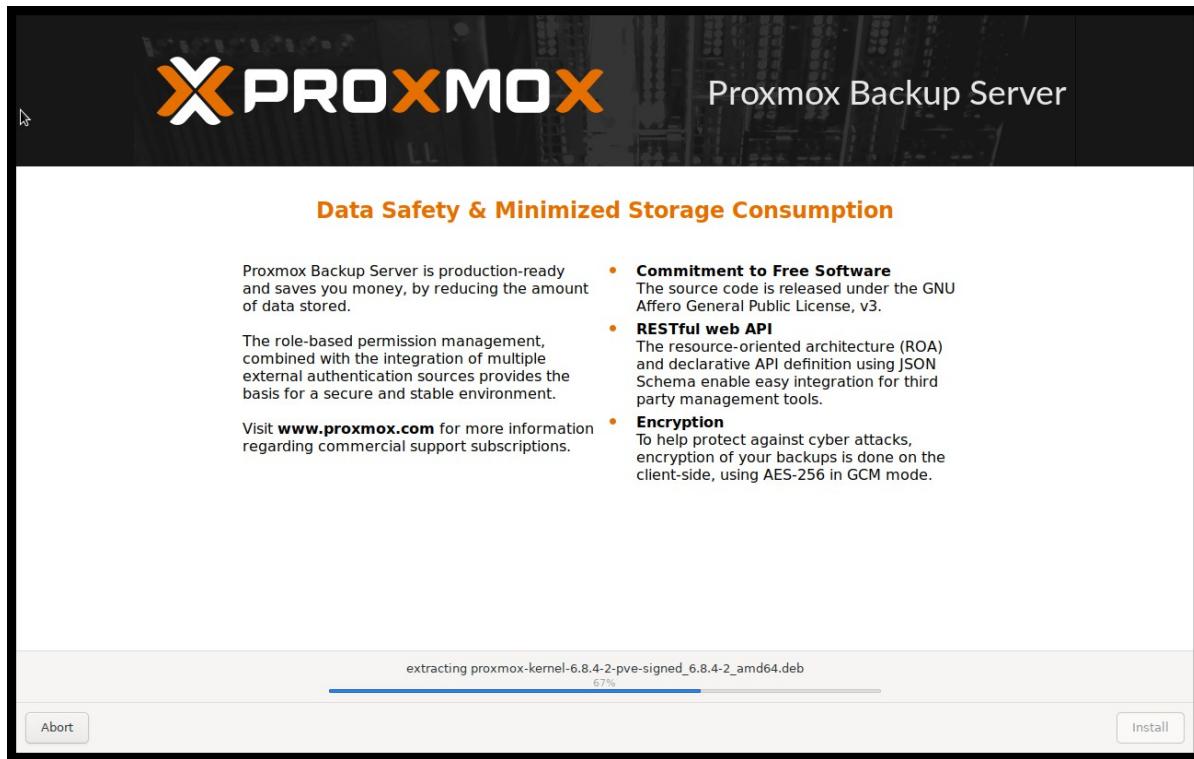
電子メールアドレスは、システム管理者に通知を送信するために使用されます。例えば

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



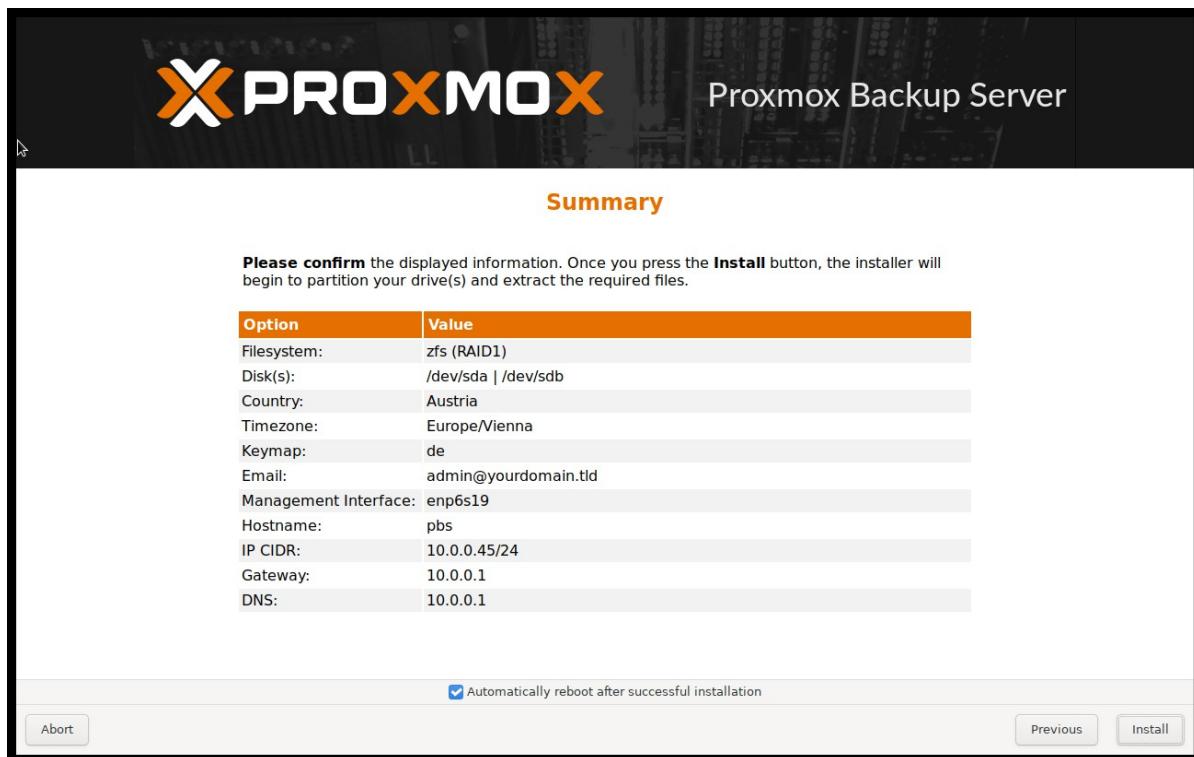
これらの通知メールはすべて指定されたメールアドレスに送信されます。

最後のステップはネットワーク設定です。UPされている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の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、[前へ]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出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。

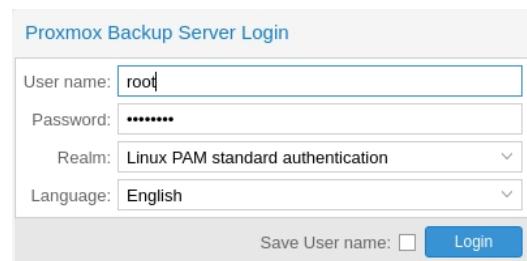
パッケージのコピーとセットアップが終了したら、サーバーを再起動します。デフォルトでは数秒後に自動的に行われます。

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)で特定のエラーをチェックし、システムが最小要件を満たしていることを確認します。

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](#)」の章をご覧ください。

インストール後の管理インターフェイスへのアクセス



システムのインストールと再起動が完了したら、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*) のユーザー名と、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コンフィグレーションとホスト名を確認してください。
- タイムゾーンを確認してください。

高度なLVM構成オプション

インストーラは、pbs という名前のボリュームグループ (VG) と、次のような追加の論理ボリューム (LV) 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

- ハードサイズ

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。

- ミンフリー

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利用可能な場合、デフォルトは16GBで、そうでない場合はhdsize/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、インストールディスク上にスワップ用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 スワップに関する注記を参照](#))。

- **アッシュシフト**

作成されるプールの `ashift` 値を定義します。`ashift` は、少なくとも基礎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) に設定される必要があります。

- **湿布**

`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。 `zfs(8)` man ページをチェックしてください。

- **ハードサイズ**

使用するハードディスクの合計サイズを定義します。`hdsizes` 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もよく動作します。ZFSを使用する、十分なRAMを用意してください。未加工のディスク容量1TBに対して、4GBと1GBのRAMで計算するのが良いでしょう。

ZFSは、ZFS Intent Log (ZIL) と呼ばれる専用ドライブを書き込みキャッシュとして使用できます。これには高速ドライブ (SSD) を使用します。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_ssds>
```

nomodesetカーネル・パラメーターの追加

グラフィックドライバが原因で、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るようにします。

Proxmox Backup Server ブートローダのメニューで、[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、既存の最後のパラメータからスペースで区切ってパラメータ`nomodeset`を追加します。

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は、無人で自動的に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設定できます。

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、フィルタールールを使用して、使用するディスクとネットワークカードを決定できます。

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 インストールの上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、必要なパックエイジの最小セットをインストールします。

インストーラーがインストールするパッケージと同じ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
```

これにより、必要なパッケージ、ZFS サポート付き Proxmox カーネル、および一般的で便利なパッケージ式がインストールされます。

注意既存のDebianインストールに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セットアップされていることが前提です。一般的に、特にLVMやZFSを使用する場合これは些細なことではありません。ネットワークの設定も完全にあなたです。

注意: Proxmox Backup Server の Web インターフェイスには、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*を設定したら、次のコマンドを実行してください：

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

注意ハイパーバイザーにバックアップ・サーバーを直接インストールすることは推奨されません。バックアップの保存には別の物理サーバを使用する方が安全です。ハイパーバイザーサーバに障害が発生しても、バックアップにアクセスできます。

注意: Proxmox Backup Server の Web インターフェイスには、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>。

2.4 クライアントインストール

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/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
```

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。両者とも */usr/bin/proxmox-backup-client* パスの実行ファイルとしてクライアントを提供するからです。

この実行ファイルを他の、たとえば Debian 以外の Linux コピーできます。Proxmoxバックアップクライアントの使用法の詳細については、[バックアップクライアントの使用法](#)を参照してください。

2.5 Debian パッケージリポジトリ

Debian ベースのシステムはすべて、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* 定義されます。アップデートは apt コマンドラインツールで直接インストールすることも、GUI 経由でインストールすることもできます。

APT *sources.list* ファイルは、1行に1つのパッケージリポジトリをリストし、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。設定されたソースから利用可能な情報は、*apt update* によって取得されます。

リスト1: ファイル: /etc/apt/sources.list

```
deb http://deb.debian.org/debian bookworm main contrib
deb http://deb.debian.org/debian bookworm-updates main contrib
# セキュリティ・アップデート
deb http://security.debian.org/debian-security bookworm-security main contrib
```

また、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://fp.debian.org/debian	bullseye	main contrib		Debian	
<input checked="" type="checkbox"/>	deb	http://fp.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	pbstest		Proxmox	
File: /etc/apt/sources.list.d/pbs-enterprise.list (1 repository)							
<input checked="" type="checkbox"/>	deb	https://enterprise.proxmox.com/debian/pbs	bullseye	pbs-enterprise		Proxmox	
File: /etc/apt/sources.list.d/pbs-staging.list (1 repository)							
<input checked="" type="checkbox"/>	deb	http://repo.proxmox.com/staging/pbs	bullseye	pbs-2		Proxmox	

2.5.1 セキュアアパート

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確認しています。

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-release-bookworm.gpg -O /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
```

その後、SHA512チェックサムを以下の期待される出力で検証してください:

```
# sha512sum /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
7da6fe34168adc6e479327ba517796d4702fa2f8b4f0a9833f5ea6e6b48f6507a6da403a274fe201595edc86a84463d50383d07f64bdd
/etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
```

と md5sum を入力します:

```
# md5sum /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg 41558dc019ef90bd0f6067644a51cf5b
/etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
```

2.5.2 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、本番環境でのいます。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 ファイル:
 /etc/apt/sources.list.d/ pbs-enterprise.list

```
deb https://enterprise.proxmox.com/debian/pbs 本の虫 pbs-enterprise
```

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り次第、スーパーユーザ（root@pamユーザ）に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: 上記の行を#（行頭）でコメントアウトすることで、このリポジトリを無効にすることができます。これにより、サブスクリプションキーがない場合のエラーメッセージを防ぐことができます。その場合は、pbs-no-subscription リポジトリを設定してください。

2.5.3 Proxmox バックアップ ノーサブスクリプションリポジトリ

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推奨されません。

このリポジトリは /etc/apt/sources.list で設定するお勧めします。

リスト3: ファイル: /etc/apt/sources.list

```
deb http://deb.debian.org/debian bookworm main contrib
deb http://deb.debian.org/debian bookworm-updates main contrib
# Proxmox Backup Server pbs-no-subscription リポジトリは proxmox.com によって提供されています # 本番環境での利用は推奨
されません
deb http://download.proxmox.com/debian/pbs bookworm pbs-no-subscription
# セキュリティ更新
deb http://security.debian.org/debian-security bookworm-security main contrib
```

2.5.4 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。このリポジトリには、/etc/apt/sources.list に以下の行を追加することでアクセスできます：

リスト4: pbstest の sources.list エントリ

```
deb http://download.proxmox.com/debian/pbs bookworm pbstest
```

2.5.5 Proxmox Backup クライアント専用リポジトリ

*Proxmox Proxmox Backup Client を使用する*プロジェクトに基づかない 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ベースのシステムにはクライアント・レポジトリしかありません。

APTベースのProxmoxバックアップクライアントリポジトリ

DebianやUbuntuの派生ように、*apt*をパッケージマネージャ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ベースの使うことができるかもしれません。

このリポジトリを設定するには、まず *Proxmox リリースキーを設定する*必要があります。、リポジトリの URL を APT ソースリストに追加します。

Debian 12 (Bookworm) ベースのリリース用リポジトリ

このリポジトリは以下の環境でテストされています：

- Debianの本の虫

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ます。

リスト5: 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

このリポジトリは以下の環境でテストされています：

- 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ます。

リスト6: ファイル: /etc/apt/sources.list

```
deb http://download.proxmox.com/debian/pbs-client ブルズアイ・メイン
```

Debian 10 (Buster) ベースのリリース用リポジトリ

このリポジトリは以下の環境でテストされています：

- Debian バスター
- Ubuntu 20.04 LTS

古いバージョンでも動くかも、最近リリースされた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ます。

リスト7: ファイル: /etc/apt/sources.list

```
デブ http://download.proxmox.com/debian/pbs-client バスター・メイン
```

2.5.6 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、中央プロキシを通じてのみアクセスできる場合があります。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、apt ネットワークリクエストと 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プロキシ設定が受け入れられます。[http://]<host>[:port]ここで、<host>の部分には認証が含まれる場合があります。

用語解説

3.1 バックアップ・コンテンツ

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。データの種類によって、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パワーしか必要とせず、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は不可欠です。

Proxmox Backup Serverは両方のサポートしています。

3.1.1 画像アーカイブ: <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: <名前>.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を高めるため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、コンフィギュレーション・ファイルのような小さい（16MB未満）バイナリ・データを保存するのに使われます。それ以上のファイルはイメージアーカイブとして保存してください。

ご注意 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.pcat1

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。これは含まれているファイルのリストを含んでおり、検索操作を高速化するためには使用されます。

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、サイズ、およびチェックサムが含まれます。これは、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

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単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、これらは単純なディレクトリ構造として実装されており、個別の設定は必要ありません。

3.3 バックアップタイプ

バックアップサーバーはバックアップをタイプ別にグループ化します:

ブイエム

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。コンテナのコンフィギュレーションと、ファイルシステムのコンテンツを格納する単一のファイルアーカイブで構成されます。

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。通常は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

2番目の解像度で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>のタプルをバックアップ・グループと呼びます。このようなグループには1つ以上のバックアップ・スナップショットを含めることができます。

3.7 バックアップスナップショット

<type>/<ID>/<time>のトリプレットはバックアップ・スナップショットと呼ばれます。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

ご覧のように、時刻のフォーマットはRFC協定世界時 (UTC、末尾のZで識別) です。

グラフィカルユーザーインターフェース

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。つまり、ウェブブラウザですべての管理タスクを実行でき追加の管理インストールする必要がありません。ビルトインコンソールも提供するため、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には <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です。

4.1 特徴

- Proxmox Backup Serverのシンプルな管理インターフェイス
- タスク、ログ、リソース使用状況の監視
- 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
- 多言語対応
- ExtJS 6.x JavaScriptフレームワークをベースにしています。

4.2 ログイン

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

- User name:
- Password:
- Realm:
- Language:

At the bottom left is a checkbox labeled 'Save User name: '. At the bottom right is a blue '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: 利便性のため、ウィンドウの下部にある「ユーザー名を保存」チェックボックスを選択すると、クライアント側でユーザ

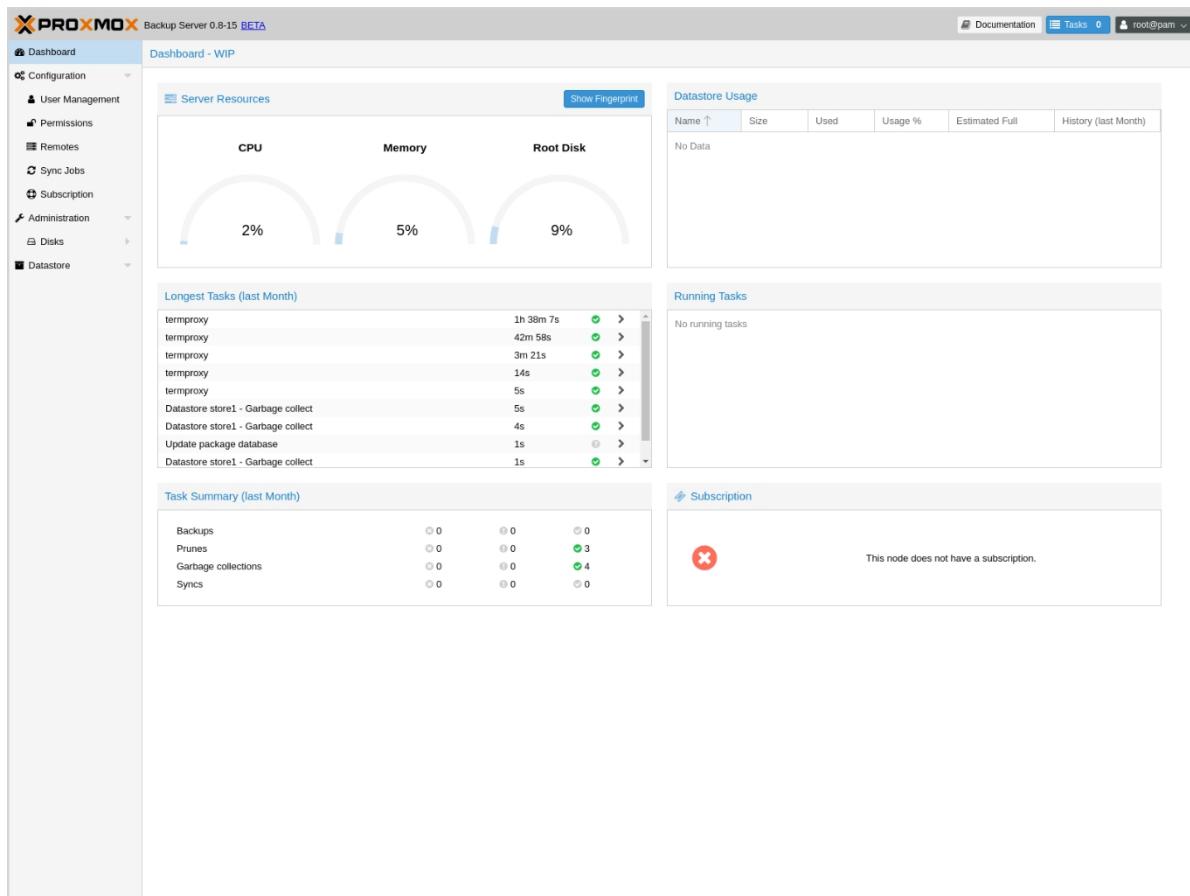
一名を保存できます。

4.2.1 同意バナー

ログイン前に承諾が必要なカスタム同意バナーは、設定で設定できます。

→ その他 → 全般 → 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ウェブ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、ドキュメントの閲覧、実行中のタスクの監視、言語設定、各種表示、ログアウトなどのボタンがあります。
- サイドバー：左側。ここにはサーバーの管理オプションがあります。
- 設定パネル：中央にあります。これには、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

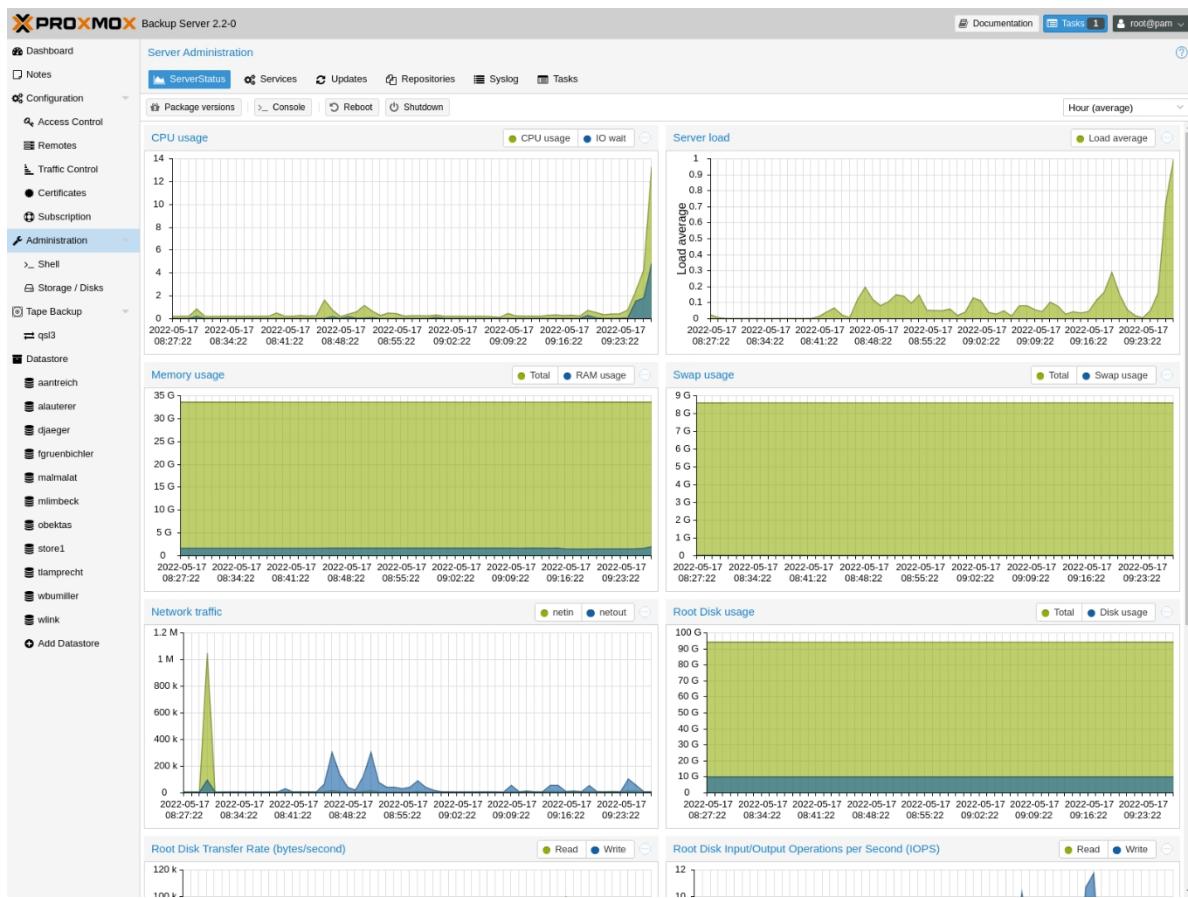
ダッシュボードには、サーバーのアクティビティとリソースの使用状況の概要が表示されます。具体的には、ハードウェアの使用状況、以前および現在実行中のタスクの概要、サブスクリプション情報が表示されます。

4.4.2 構成

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：

- アクセスコントロール：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権限の追加と管理
- 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。
- 購読：購読キーのアップロード、購読ステータスの表示、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 管理部門



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。これらは以下の通りです:

- **ServerStatus:** コンソール、電源オプション、各種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
- アップデートパッケージのアップグレード用インターフェース
- 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
- **Syslog:** 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を備えたタスク履歴

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	536.87 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:

- ディスク使用可能なディスクの情報を表示
 - 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ディスクの情報の作成と表示

4.4.4 テープバックアップ

The screenshot shows the Proxmox Tape Backup interface. On the left is a sidebar with navigation links: Dashboard, Configuration, Access Control, Remotes, Subscription, Administration, Shelf, Storage / Disks, Tape Backup, ibm3584, qpx720 (which is selected), Datastore, pve-backup, remote-sync, and Add Datastore.

The main area has three tabs: Changer: qpx720, Reload, Barcode Label, and Inventory. The Changer tab is active.

Slots: A table with columns ID, Content, Inventory, and Actions. It lists slots 1 through 20, each containing a tape labeled TAPESL4 or TAPESL64, with the inventory status as "writable (pve-backup)".

Drives: A table with columns Content, Inventory, Name, State, and Actions. It shows two drives: qdrive0 (idle) and qdrive1 (idle), both connected to TAPESL4.

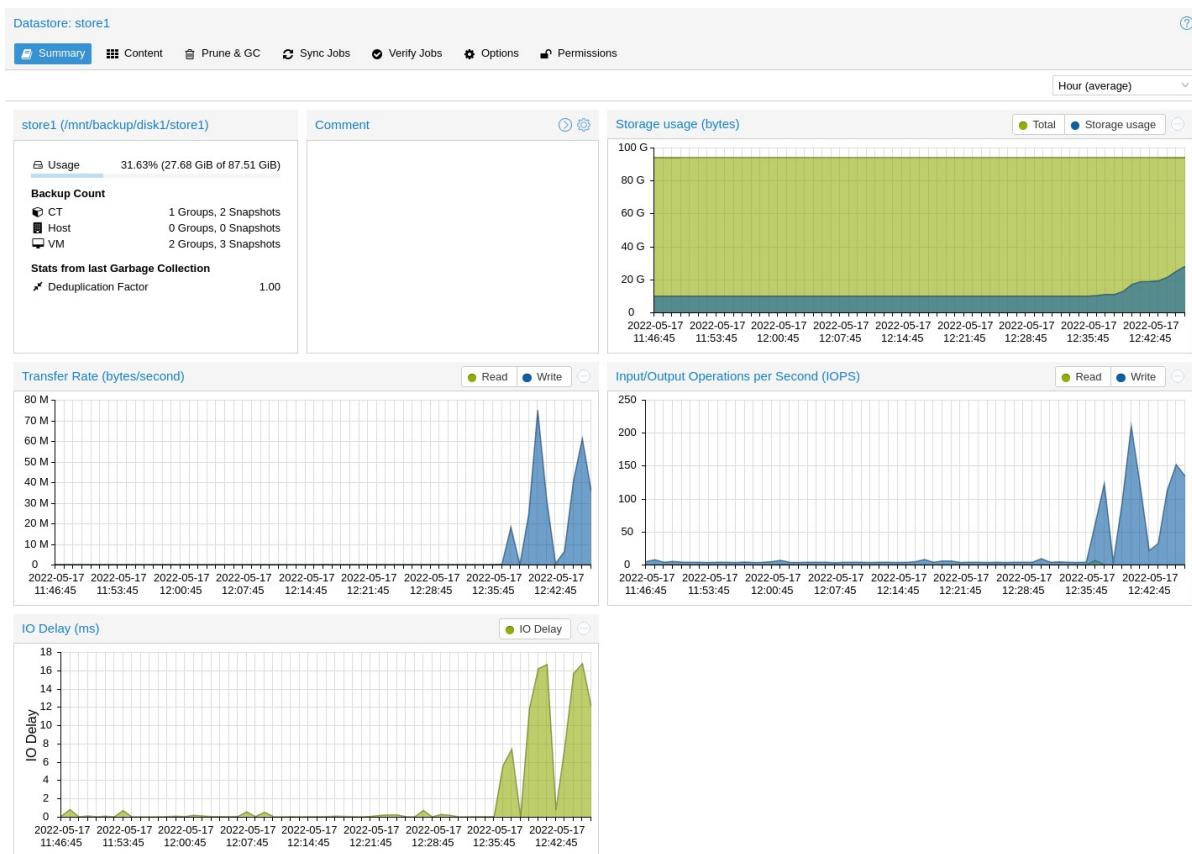
Import-Export Slots: A table with columns ID, Content, Inventory, and Actions. It lists slots 21 through 24, which are currently empty.

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含むトップパネルがあります。タブは以下のとおりです：

- 内容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の管理
- 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ーごとのサブセクションもあり、そ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。また、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できます：

- 要約: データストアの使用統計情報へのアクセス
- 内容: データストアのバックアップ・グループとそれぞれの内容に関する情報
- ブルーニングと**GC**: ブルーニングとスケジュールを設定し、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ー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
- パーミッションデータストアの権限を管理します。

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/sda	Hard Disk	mounted	16.00 TB	No	ST16000NM001G-2KK103	ZL21WKQX	passed	No	N/A
/dev/sdb	Hard Disk	filesystem	16.00 TB	No	ST16000NM001G-2KK103	ZL22H7VG	passed	No	N/A
/dev/sdc	Hard Disk	filesystem	16.00 TB	No	ST16000NM001G-2KK103	ZL223443	passed	No	N/A
/dev/sdd	Hard Disk	filesystem	16.00 TB	No	ST16000NM001G-2KK103	ZL22NCFQ	passed	No	N/A
/dev/sde	Hard Disk	filesystem	16.00 TB	No	ST16000NM001G-2KK103	ZL22LHSF	passed	No	N/A
/dev/sdf	Hard Disk	filesystem	16.00 TB	No	ST16000NM001G-2KK103	ZL22B4HK	passed	No	N/A
/dev/sdg	Hard Disk	filesystem	16.00 TB	No	ST16000NM001G-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、ディスクに関する情報の取得ができます。

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ースの [Administration] → [Storage/Disks] に移動するか、ディスクの [list] サブコマンドを使用します：

# proxmox-backup-manager ディスクリスト						ウェアアウト
名称	使用	gpt	ディスクタイプ	サイズ	モデル	
SDA	LVM		1 hdd	34359738368	QEMU_HARDDISK -	
sdb	未使用 1		hdd	68719476736	QEMU_HARDDISK -	

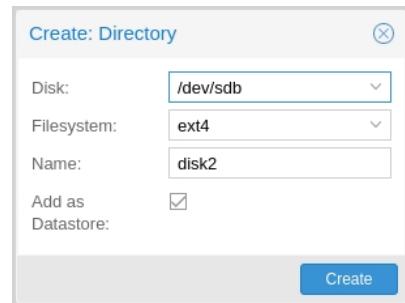
(次ページへ続く)

SDC 未使用 1	HDD	68719476736	
-------------	-----	-------------	--

(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`サブコマンドを使用します：

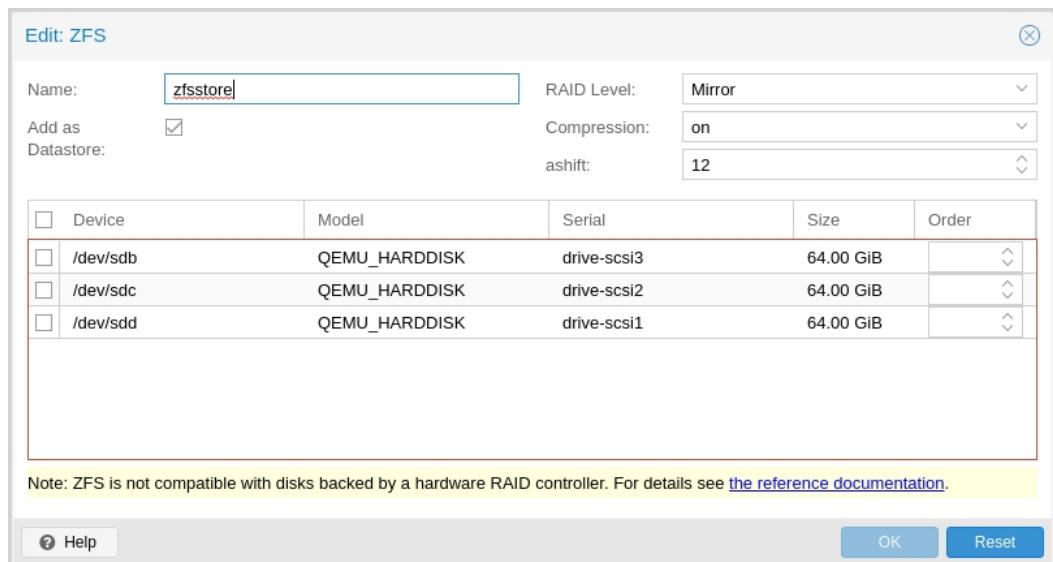
```
# proxmox-backup-manager disk initialize sdX
```



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ースの **Admin-istration -> Storage/Disks -> Directory** に移動して、そこから作成します。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
```



ウェブインターフェースの**Administration -> Storage/Disks -> ZFS**から、または`zpool create`を使用して、さまざまなレイドレベルのzpoolを作成することもできます。以下のコマンドは、2つのディスクを使用してミラー化された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
```

注意：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を自動的に作成することもできます。

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

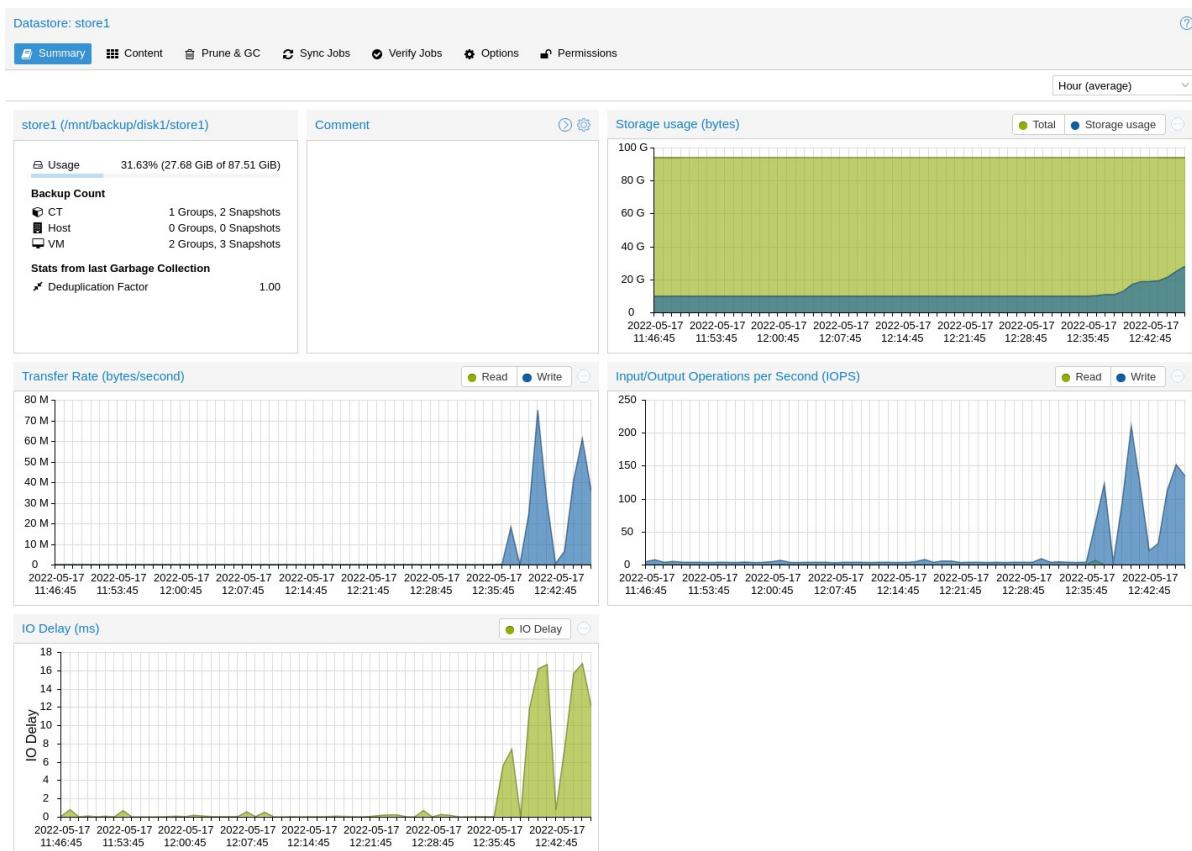
Proxmox Backup Serverはsmartmontoolsパッケージを使用します。これは、ローカルハードディスクのS.M.A.R.T.システムを監視および制御するためのツールセットです。ディスクがS.M.A.R.T.機能をサポートしており、かつ

を有効にすると、ウェブ・インターフェイス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：

```
# proxmox-backup-manager disk smart-attributes sdX
```

注：この機能は、smartmontoolsパッケージの一部として提供されている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（man smartctlを参照してください）。

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設定できます。データストアの設定情報は/etc/proxmox-backup/datastore.cfgファイルに保存されます。

注意： [ファイルレイアウトでは](#)、ファイルシステムが1つのディレクトリにつき少なくとも 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、

... デフォルトのディレクトリ・エントリ。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。例えば、ext3全体や、dir_nlink機能を手動で無効にしたext4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 1	root@pam	No 1	All OK None
vm/115								
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、年毎などの間隔ごとにバックアップ・スナップショットをいくつ保持するかという保持設定と、そのストアに保持するバックアップの数に時間依存しない設定もあります。プルーニングと削除バックアップとガベージ・コレクションは、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」を参照）に基づいて、定期的に実行するように設定することもできます。

データストアの作成

Name:	store1	GC Schedule:	Tue 04:27
Backing Path:	/mnt/backup/disk1/store1	Prune Schedule:	daily
Comment:			
<input type="button" value="Help"/>		<input type="button" value="Add"/>	

Web インタフェー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。セットアップ・ウィンドウで

- Name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スケジュールとは、ガベージコレクションが実行される時間と間隔のことです。
- ブルーンスケジュールとは、ブルーニングを行う頻度のことです。
- ブルーンのオプションは、保持するバックアップの量を設定します（[バックアップの再移動をブルーニングと参照](#)）。
- 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することができます。

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。次のコマンドは、/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 データストア作成 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инг・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じようにext4かxfsであるべきですが、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が比較的低いです。したがって、データストアの作成はFATファイルサポートされていません。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イ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る場合があります。

また、"Administration" (管理) > "Disks / Storageディスク/ストレージ> "Directory"(ディレクトリ)を使用して、完全に使用されていない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分割され、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、データストアはアンマウントされたメンテナスマードのままとなり、IO操作ができなくなります。このような、メンテナスマードを手動でリセットする必要があります：

```
# proxmox-backup-manager データストア更新 -メンテナスマード オフライン
```

IOを防ぐため、または使用をクリアするために：

```
# proxmox-backup-manager データストアの更新 --delete maintenance-mode
```

一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が、唯一の制限は、データストアを入れ子にすることができないということです。

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。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場合は、作成時に設定したパスと一致させる必要があります。

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

どちらも実行中のタスクが終了するのを待って、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、およびガーベージコレクションの各ジョブがスケジュール時にスキップされました。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です。

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、次のように実行します：

```
# proxmox-backup-manager データストアリスト
+-----+-----+-----+
| 名称 | 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 |
+-----+-----+-----+
```

データストアのガーベージ・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。たとえば、次のコマンドはupdateサブコマンドを使用してガーベージ・コレクション・スケジュールを変更し、showサブコマンドを使用してデータストアのプロパティを表示します：

```
# proxmox-backup-manager データストア更新 store1 --gc-schedule 'Tue 04:27'
# proxmox-backup-manager datastore show store1
+-----+-----+
| 名称      | 値           |
+-----+-----+
| 店名      | ストア1       |
+-----+-----+
| パス      | //バックアップ/ディスク1/ストア1 |
+-----+-----+
| コメント   | これが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            |
+-----+-----+
| prune-schedule | 毎日         |
+-----+-----+
```

最後に、データストア構成を削除することができます：

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

注： 上記のコマンドは、データストア構成のみを削除します。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・レイアウトが表示されます：

```
# ls -alrh /backup/disk1/store1
r--r-- 1 backup バックアップ          0 7/8 12:35 .ロック
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、バックアップ操作が実行された後、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 545823 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 144079 drwxr-x--- 2 backup backup
4.0K Jul 144078 drwxr-x--- 2 backup backup 4.0K Jul
144077 drwxr-x--- 2 backup バックアップ 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
バックアップ バックアップ 4.0K Jul 276489 drwxr-xr-x 3 バッ
クアップ バックアップ 4.0K Jul 276490 drwxr-x--- 1 バックア
ップ バックアップ 1.1M Jul
8 12:35 ffff
8 12:35 fffe
8 12:35 fffd
8 12:35 fffc
8 12:35 fffb
8 12:35 fffa
8 12:35 fff9
8 12:35 fff8
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 バックアップ名前空間

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できます。しかし、階層構造や分離がないと、特に複数のProxmox VEインスタンスや複数のユーザ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。

バックアップ名前空間の階層構造により、異なるユーザまたはバックアップソース全般を明確に分けることができ、名前の衝突を回避し、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、またはホストだけでなく、他のネームスペースもホストできます。

名前空間パーミッション

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、BACKUPのいずれかのパーミッションが必要です。特権は、そのネームスペースに含まれるすべてのネームスペースに適用されます。

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 MODIFY 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 権限を含むアクセス・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そのまま適用されます。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
	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)

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
- メンテナンスモード
- 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:

Proxmox Backup Serverがベリファイまたはテープへのバックアップを実行するとき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) : ファイルシステムの inode 番号でチャンクをソートしてから繰り返します。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)に現れる順番にチャンクを反復します。これは多くの低速ストレージでは反復処理を遅くするかもしれません、非常に高速なストレージ（例えば、NVME）では、収集と並べ替えに、並べ替えによる反復処理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 データストア更新 <ストア名> --tuning 'chunk-order=none'
```


- 同期レベル：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することができ、停電やハード 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書き込みをディスクに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、内部の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にフラッシュするために、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。

で設定できます：

```
# proxmox-backup-manager データストア更新 <ストア名> --tuning 'sync-level=filesystem'
```

- *gc-atime-safety-check*：データストアの GC 2011 時刻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するため、予期しないファイルシステムの動作によるデータ損失のリスクを回避できます。これはデフォルト値もあります。
- *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・コレクションのフェーズ1中にアクセス・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るようになります。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、メモリ使用量が増加する代償として GC 2011 の実行時間を短縮できます。

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ってください：

```
# proxmox-backup-manager データストア更新 <ストア名> -チューニング 'sync-level=filesystem,chunk-'
'>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、オフサイト同期、バックアップからの容易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 内蔵プロテクション

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ホストや、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。要するに、少なくとも2種類の記憶媒体に3つのバックアップを作成し、そのうち1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犠牲になった場合、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。

第二の記憶媒体として[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できます。

5.3.3 制限的な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とアクセス管理システムを提供します。例えば、*Datastore.Backup*権限では、既存のバックアップの削除や変更はできませんが、作成だけは可能です。

このアクセス・コントロール・システムを活用する最善の方法は

- Proxmox Backup Server にデータをバックアップするホストまたは 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。例えば、`/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パス上でDataStoreアクセス・ロールを付与する単一のパーミッションのみを持つべきです。

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、削除権限を付与せず、Proxmox Backup Serverで[プルーンジョブ](#)を使用して直接バックアップのプルーニングを実行することです。

これはシンクジョブについても同様です。シンクユーザまたはアクセストークンの権利をバックアップの書き込みのみに限定し、行わないようにすることで、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。このような事態が発生し、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)のパッチを適用します。
- 安全でセキュアなネットワーク・プラクティスに従います。たとえば、ロギングやモニタリング・ツールを使用したり、VLANを設定するなどして、インフラストラクチャーのトラフィックとユーザー・公共のトラフィックが分離されるようにネットワークを分割したりします。
- 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。そのため、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は、CISA(米国)のガイドのような政府の公式勧告や、ENSIAのThreat Landscape for Ransomware Attacksやnomoreransom.orgのようなEUのリソースを参照してください。

ユーザー管理

6.1 ユーザー設定

User name	Realm	Enabled	Expire	Name	Comment
alauterer	pbs	Yes	never		
alwin	pbs	No	never		
cerez	pbs	Yes	never		
csparks	pbs	Yes	never		
djaeger	pbs	No	never		
fechner	pbs	Yes	never		
fgruenbichler	pbs	Yes	never		
j.smith	pbs	Yes	never	John Smith	
jbrown	pbs	Yes	never		
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
sruiz	pbs	Yes	never		
tdavis	pbs	Yes	never		
tamprecht	pbs	Yes	never		
tm Marx	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、新規ユーザー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次のとおりです：

パム

Linux PAM標準認証。Linux システム・ユーザとして認証したい場合に使用します。ユーザはホストシステム上に既に存在している必要があります。

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。

オープニド

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。

ます。

エル

LDAPサーバー。ユーザは外部のLDAP認証を行うことができます。

ダッ

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

プア

ド

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を一覧表示したり、操作することができます：

# proxmox-backup-manager ユーザーリスト					
ユーザーID	enable	expire	firstname	lastname	email
root@pam		1			

|コメント
| スーパーユーザー |

Add: User

User name:	j.smith	First Name:	John
Password:	*****	Last Name:	Smith
Confirm password:	*****	E-Mail:	j.smith@example.com
Expire:	never		
Enabled:	<input checked="" type="checkbox"/>		
Comment:			
Help	Add		

スーパーユーザはすべての管理権限を持ちますので、より低い権限の他のユーザを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、後でユーザ更新サブコマンドを使用して更新または変更できます（GUIで編集）：

```
# proxmox-backup-manager user create john@pbs 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。"
```

出来上がったユーザーリストは次のようにになります：

# proxmox-backup-manager ユーザーリスト					
ユーザーID	enable	expire	firstname	lastname	email
john@pbs	1		ジョン	スミス	john@example.com
root@pam	1				

|コメント
| ユーザの一例 |
| スーパーユーザー |

新しく作成されたユーザにはアクセス権がありません。アクセス権を設定する方法については、[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
```

でユーザーを完全に削除することもできます：

```
# 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	

認証されたユーザーは誰でもAPIトークンを生成することができ、ユーザー名とパスワードを直接提供する代わりに、APIトークンを使用し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. ユーザー権限内で各クライアント/トークンの権限を制限します。

APIトークンは、ユーザ名、realm、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。どちらも、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する必要があります。

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のAuthorization HTTPヘッダーに値TOKENID:TOKENSECRETを設定することで、クライアントからサーバーに渡されます。

GUIまたはproxmox-backup-managerを使ってトークンを生成できます：

```
# proxmox-backup-manager user generate-token john@pbs client1 結果: {
  "tokenid": "john@pbs!client1",
  "secret": "58a77e1c-77ea-4e7d-bf2c-e265b43d93c0"
}
```



```
{
    "値": "d63e505a-e3ec-449a-9bc7-1da610d4ccde"
}
```

(前ページから続く)

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` サブコマンドを使用すると、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
+-----+-----+-----+
| トークンID | enable | expire |
+-----+-----+-----+
| john@pbs!client1 | | |
```

同様に、`user delete-token`サブコマンドは、トークンを再度削除するために使用できます。

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トークンには権限がありません。その代わりに、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を使用しています。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、このようなアクセスルールは「(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すことができ、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 特典

権限はアクセス・ロールの構成要素です。内部的には、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、以下の特権をサポートしています：

システム監査

Sys.Auditにより、ユーザーはシステムとその知ることができます。

システム変更

Sys.Modify は、ユーザーがシステムレベルの構成を変更し、アップデートを適用できるようにします。

電源管理

Sys.Modifyを使用すると、ユーザーはシステムの電源を切って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、ユーザがデータストアについて知ることができます。

データストアアロケート

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

データストア修正

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・読み込み

`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、ユーザが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。

データストア・バリファイ

データストア内のバックアップスナップショットを検証できるようにします。

データストア・バックアップ

`Datastore.Backup` を使用すると、ユーザは新しいバックアップ・スナップショットを作成でき、`Datastore.Read` と `Datastore.Verify` の権限も提供されますが、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つである場合に限られます。

データストア・プルーン

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

パーミッションの変更

`Permissions.Modify` により、ユーザーは ACL を変更できます。

注意: API トークンは、ユーザ権限によって制限されるため、ユーザは常に自分の 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、ユーザー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・モディファイ

`Remote.Modify` を使用すると、ユーザーはリモート構成を変更できます。

リモート・リード

`Remote.Read` を使用すると、ユーザーは設定された `Remote` からデータを読み取ることができます。

シスコン

`Sys.Console` は、ユーザーがシステムのコンソールにアクセスすることを可能にします。`root@pam` 以外では、有効なシステムログインが必要であることに注意してください。

テープ監査

`Tape.Audit` では、テープドライブ、チェンジャー、バックアップの設定や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、バックアップの設定を変更できます。

テープ書き込み

`Tape.Write` により、ユーザーはテープ・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、テープメディアからテープバックアップの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、ユーザの認証レルムの表示、作成、変更、削除を行うことができます。

6.3.2 アクセス ロール

アクセス・ロールは、1つ以上の権限を組み合わせて、オブジェクト・パス上のユーザーまたはAPIトークンに割り当てることができるものです。

現在のところ、組み込みのロールしかありません。つまり、独自のカスタム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：

アクセス禁止

アクセスを無効にする - 何も許可されません。

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

物事の状態や設定を見ることはできますが、設定を変更することはできません。

データストア管理

既存のデータストアで何でもできます。

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とリストアが可能。

データストアパワーユーザー

所有するバックアップのバックアップ、リストア、ブルーニングが可能です。

リモート管理

リモコンで何でもできます。

リモート監査

リモートの設定を見ることができます。

リモートシンクオペレーター

リモートからデータを読み込むこと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ことなら何でもできます。

テープ監査

テープ関連のメトリクス、構成、ステータスを表示できます。

テープオペレータ

テープバックアップとリストアはできますが、コンフィギュレーションを変更することはできません。

テープリーダー

テープの構成やメディアの読み取り、検査することができます。

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、または一部のシステム再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を扱うのに、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、より高いレベル（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の下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。

いくつか例を挙げましょう：

/データストア	Proxmox Backup サーバー上のすべてのデータストアへのアクセス
/datastore/{store}	Proxmox Backup サーバーの特定のデータストアへのアクセス
/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へのアクセス
/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

相続

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、その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 トークンの権限は、常に権限に制限されます。
- より深く、より具体的なレベルのパーミッションは、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

Add: User Permission

Path:	/datastore/store1
User:	j.smith@pbs
Role:	DatastoreAdmin
Propagate:	<input checked="" type="checkbox"/>
<input type="button" value="Help"/> <input type="button" value="Add"/>	

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通り：

1. ACL 識別子

2. 伝搬が有効か無効かを表す1または0。
3. パーミッションが設定されるオブジェクト。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オブジェクトも表します。

4. 権限が設定されたユーザー/トークン

5. 設定されている役割

パーミッションは、ウェブインターフェイスの**Configuration -> Access Control -> Permissions**で管理できます。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できます。たとえば、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、以下のコマンドを使用して一覧表示できます：

```
# proxmox-backup-manager acl list
+-----+-----+-----+-----+
| プギッド | パス          | 伝播 | roleid |
+-----+-----+-----+-----+
| john@pbs | /datastore/store1 1 |    ||
```

1つのユーザー/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には、/datastore/{ストア名}という命名規則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、/datastore/store2を使用します。リモートストアの場合、/remote/{remote}/{storename}という規約を使用します。{remote}はリモートの名前（後述の「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、指定されたパスのパーミッション・セットは、対応するユーザーのパーミッション・セットと交差します。

実際には、これが意味するところは

1. API トークンは独自のACLエントリを必要とします。
2. API トークンは、対応するユーザー

6.3.6 効果的なパーミッション

ユーザーまたは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ユーザ権限コマンド：

```
# proxmox-backup-manager user permissions john@pbs --path /datastore/store1 (*)が付いた特権には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.監査 (*)
- データストア.バックアップ (*)
- データストアの変更 (*)
- 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.ベリファイ (*)

# proxmox-backup-manager acl update /datastore/store1 DatastoreBackup --auth-id 'john@pbs!
client1'

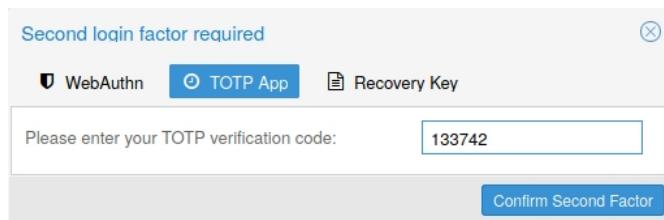
# proxmox-backup-manager user permissions 'john@pbs!client1' --path /datastore/store1 (*)の付いた特権にはpropagateフラグが設定されています。
パス/データストア/ストア1
- データストア.バックアップ (*)
```


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な ID の主張（認証）を成功させるために必要なのはパスワード（単一要素）のみです。この場合、パスワードが漏れたり盗まれたりすると、誰でもログインすることができます。

二要素認証(TFA)では、ユーザが認証されていることを確認するために、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）に頼るのではなく、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（第二の要素）にはアクセスできないため、遠隔地にいるユーザーが不正にアクセスすることを防ぐことができます。



6.4.2 利用可能なセカンドファクター

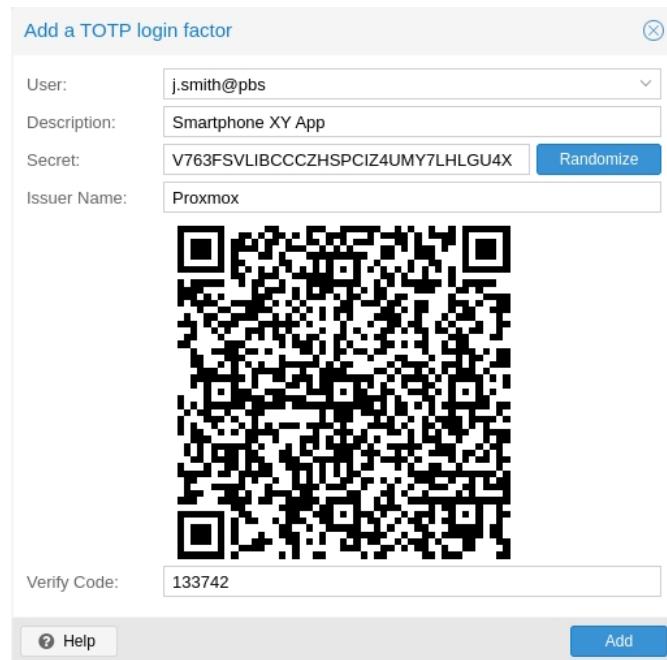
スマートフォンやセキュリティキーを紛失してアカウントから永久にロックされてしまう事態を避けるため、複数のセカンドファクターを設定することができます。

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)。共有された秘密情報と現在時刻から導き出される短いコードで、30秒ごとに変更されます。
- WebAuthn (Web Authentication)。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。
- シングルユースのリカバリーキー。プリントアウトして安全な場所に施錠するか、電子保管庫にデジタル保存する必要があるキーのリスト。各キーは1回のみ使用できます。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のに最適なキーです。

6.4.3 セットアップ

トットピー



サーバーの設定は不要です。スマートフォンにTOTPアプリ（例えばFreeOTP）をインストールし、Proxmoxバックアップサーバー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。

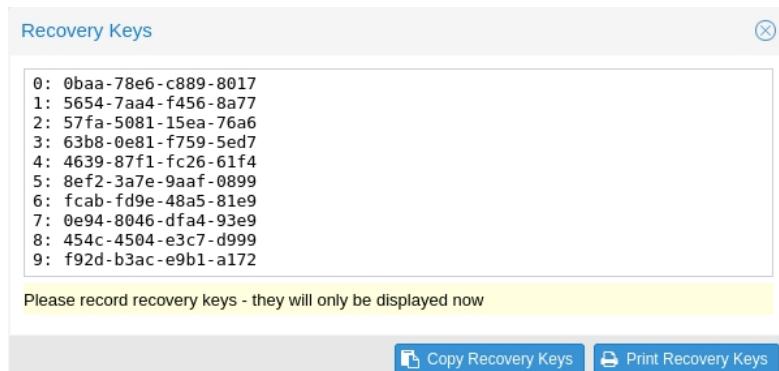
ウェブオート

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、信頼されていない場合、ブラウザによっては 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 設定を設定します (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。

これらの両方の満たしたら、WebAuthn 設定を
アクセスコントロール二要素認証タブ。

リカバリーキー



リカバリーコードの準備は必要ありません。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: 11セットしか使用できません。

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

第二の要因は、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。しかしながら、いくつかの要素は総当たりで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であり、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、Web UIのユーザーリストビュー、またはコマンドラインを使用して、いつでもユーザーの二要素認証を解除することができます：

```
proxmox-backup-manager ユーザ tfa ロック解除 joe@pbs
```

6.5 認証領域

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、ホストシステム上でadduser などのコマンドを使用して作成します。

ホストシステムに PAM ユーザーが存在する場合、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バックアップ認証サーバー

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 ライトウェイトディレクトリアクセスプロトコル

Proxmox Backup Serverは、ユーザ認証に外部のLDAPサーバを利用できます。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。

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
オブジェクトクラス: 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。そのため、Proxmox Backup ServerのユーザID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、有効なProxmox Backup Serverのユーザー識別子を含むLDAP属性の名前が含まれます。

上記の例では、user-attrをuidに設定すると、user1@<realm-name>がuid=user1,ou=People,dc=ldap-test,dc=comというLDAPエンティティ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。わかると、LDAPサーバーに対してユーザーを認証するためにLDAPバインド操作が実行されます。

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。

全設定パラメータの[domains.cfg](#) にあります。

注意: 特定のユーザにLDAPサーバを使用した認証を許可するには、Proxmox Backup Serverでも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アドレスを指定しなければなりません。[LDAP](#) のほとんどのオプションが Active Directory にも適用されます。特に重要なのは bind-dn と password です。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、一般的な LDAP-DN 構文で指定します。

認証ドメイン名を指定するのは、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。

全設定パラメータの[domains.cfg](#) にあります。

注意Active Directoryサーバを使用して特定のユーザの認証を許可するには、Proxmox Backup Serverでも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。

注意：現在、大文字と小文字を区別しないユーザー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のではなく、LDAP および AD ベースのレルムに対してユーザーを自動的に同期することが可能です。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と`proxmox-backup-manager ldap`および`proxmox-backup-manager ad`コマンドでそれぞれ設定できます。

ユーザの同期はGUIで設定> アクセス制御> レルム の下でレルムを選択し、*同期*ボタンを押すことで開始できます。同期ダイアログでは、realm設定で設定されたデフォルト設定の一部を上書きすることができます。あるいは、`proxmox-backup-manager ldap sync`と`proxmox-backup-manager ad sync`コマンドでそれぞれユーザ同期を開始することもできます。

バックアップクライアントの使用

Proxmox Backup Serverのコマンドラインクライアントは`proxmox-backup-client`です。

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（`username`は`user@realm`の形式で指定します）：

[ユーザー名@]サーバー[: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（NATやポート転送の設定など）、ポートを指定できます。

サーバーがIPv6アドレスを使用している場合は、角括弧で括らなければならないことに注意してください。
`[fe80::01]`。

リポジトリを渡すには、`--repository` コマンドラインオプションを使用します。

`PBS_REPOSITORY` 環境変数。

Web インターフェイスでは、データストアのサマリにコピー可能なリポジトリテキストが表示されます。

以下は、有効なリポジトリとそれに対応する実数の例です：

例	ユーザー	ホスト:ポート	データストア
マイデータストア	ルート@パム	ローカルホスト:8007	マイデータ
マイホスト名:マイデータストア	ルート@パム	ホスト名:8007	マイデータ 引き裂く
<code>user@pbs@マイホスト名:mydatastore</code>	ユーザー	ホスト名:8007	マイデータ 引き裂く
ユーザー@pbs! トークン@ホスト:ストア	<code>user@pbs!</code> トークン	ホスト:8007	店舗
192.168.55.55:1234:mydatastore	ルート@パム	192.168.55.55:1234	マイデータ 引き裂く
<code>[ff80::51]:マイデータストア</code>	ルート@パム	<code>[ff80::51]:8007</code>	マイデータ 引き裂く
<code>[ff80::51]:1234:mydatastore</code>	ルート@パム	<code>[ff80::51]:1234</code>	マイデータ 引き裂く

注意: proxmoxバックアップの静的にリンクされたバイナリを使用している場合、クライアントの名前解決はlibcが提供するメカニズムではなく、純粋に

Rust プログラミング言語。そのため、ネームサービス提供する機能やモジュールは使用できません。

7.2 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と同様ですが、それぞれオープンファイルディスクリプター、ファイル名、コマンドの標準出力からデータを読み込みます。上記の順番から最初に定義された環境変数が優先されます。

pbs_encryption_password

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

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、ファイル名またはをそれぞれコマンドの標準出力から返します。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。

pbs_fingerprint

設定すると、この値がサーバー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された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：ホストをシールドするための推奨されるソリューションは、HTTPプロキシを使用する代わりに、wireguardなどのトンネルを使用することです。

注：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ため、最初の改行の後に任意のコメントを追加することができます。

7.3 システムおよびサービスの認証情報

上記の環境変数のいくつかは、代わりに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	proxmox-backup-client.encryption-password
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、次のようにユニット・ファイルの中やトランジエント・スコープ・ユニットの中で再利用することができます：

```
# systemd-run --pipe --wait \
--property= LoadCredentialEncrypted= proxmox-backup-client.password:/full/path/to/my-api-.
--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、SMBIOS タイプ 11 を介してハイパーバイザーから仮想マシンに渡される）は、`systemd.exec(5)` のマニュアルページで説明されているように、`LoadCredential=` を介してサービスにロードすることができます。

7.4 出力形式

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

テキスト形式（デフォルト）。構造化データは表としてレンダリングされます。

json

JSON（1行）。

json-プリティ

JSON（複数行、きれいにフォーマットされています）。

また、以下の環境変数は出力の動作を変更することができます：

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

(任意の) 設定されている場合、テーブルのボーダーをレンダリングしません。

ヘッダなし

(任意の) 設定されている場合、テーブルヘッダをレンダリングしません。

注：テキスト・フォーマットは人間が読めるように設計されており、オートメーション・ツールで解析することを意図していません。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。

7.5 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、リポジトリいる必要があります。以下の例では、`backup-server:store1` を使用します。

```
# proxmox-backup-client backup root.pxar:/ --repository backup-server:store1 バックアップの開始:
host/elsa/2019-12-03T09:35:01Z
クライアント名: elsa
マウントポイントをスキップ: "/boot/efi" マウント
ポイントをスキップ: "/dev"
マウントポイントをスキップ: "/run" マウント
トポイントをスキップ: "/sys"
87秒で12129チャンクをアップロード (564 MB/秒)。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。

注意してください: 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。その代わりマウントポイントをスキップする短いメッセージがそれぞれ表示されます。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。マウントポイントは--include-devオプション（例: --include-dev /boot/efi）。このオプションは、インクルードするマウントポイントごとに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設定を維持したい場合は、代わりに.bashrc ファイルに以下の行を追加してください。

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

この後、--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。たとえば、/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する場合です:

```
# 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・ターゲットにネームスペースを使用する場合は、--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、次のコマンドを実行します:

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

7.5.1 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。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 Proxmox Backup クライアントに指示するには、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・クライアントがディレクトリでこのようなファイルを見つけると、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ブ一致パターンとして解釈します。

ファイルには、1行に1つのグロブ・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、グロブ・マッチ・パターンを除外から明示的な包含に反転させます。これにより、一部のフ

айл/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。ディレクトリ

.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見なされます。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にのみマッチします。

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。
/ はカレント・ディレクトリ内でのみマッチします。

\ 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いずれかにマッチします。 [...] 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、[0-9] は数字 1 文字にマッチします。

グローブ・マッチ・パターンの順序は、ファイルが含まれるか除外されるかを定義します。これは、ディレクトリツリーの奥深くでマッチパターンに遭遇した場合にも当てはまり、前の除外を上書きすることができます。

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みましょう:

```
# ls -aR フォルダー/
フォルダー/
.pxareexclude フォルダ/サブサブフォルダ0 サブフォルダ1 ファイル2 フ
オルダ0:
.ファイル0 ファイル1      アイル3 .pxareexclude
フォルダ/サブフォルダ1:
.ファイル0 ファイル1      ファイル2 ファイル3
```

それぞれの .pxareexclude ファイルには以下の内容が含まれています:

```
# cat folder/.pxareexclude
/サブフォルダ0/ファイル1
/サブフォルダ1
サブフォルダ1/ファイル2
```

```
# 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:
.ファイル0 ファイル2 .pxareexclude
restored/subfolder1:
.ファイル2
```

同じ構文を --exclude パラメータを使って直接 cli で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=
```

7.5.2 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。エンコードされた可変サイズのチャunkに分割されます。各チャunkについてダイジェストが計算され、そのチャunk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、再読み込みし、対応するチャunkが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change-detection-modeをデフォルトからメタデータベースの検知に切り替えることで、上記のような制限を減らし、可能な限りメタデータが変更されていない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。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、アーカイブのメタデータを含むmpxarとファイルコンテンツの連結を含むppxarの2つの別々のファイルに格納されます。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比較されます。操作の詳細については、[技術文書](#)で詳しく説明されています。

change-detection-modeをdataに設定すると、メタデータモードを使用した場合と同じ分割アーカイブを作成できますが、以前の参照を使用しないため、すべてのファイルのペイロードを再エンコードする必要がありません。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
遺産	(現在のデフォルト) : すべてのファイルを自己の 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。
アーカイブ	

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= メタデータ
```

7.6 暗号化

Proxmox BackupはGCMAES-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:  
*****
```

キー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もできます:

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

このキーを作成すると、暗号化バックアップを作成できるようになります。

パラメータにキー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: *****  
.E.n.cryption Key Password: *****
```

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 location `~/.config/proxmox-`

backup/encryption-key.json に作成されます。--keyfile パラメータが指定されない場合、proxmox-backup-client は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。

7.6.1 マスター・キーの使用による暗号化キーの保存と復元

また、proxmox-backup-client keyを使用してRSA公開鍵/秘密鍵ペアを作成し、対称バックアップ暗号化鍵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、後で復元するために使用することも。

マスターキーを設定するには

1. バックアップ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" にデフォルトキーを作成します。
Encryption Key Password: *****
```

結果のファイルは~/.config/proxmox-backup/encryption-key.jsonに保存されます。

2. RSA公開鍵/秘密鍵ペアを作成します:

```
# proxmox-backup-client key create-master-key
Master Key Password: *****
```

現在のディレクトリに2つのファイルが作成されます、master-public.pemとmaster-private.pem。

3. 輸入 インポート 新しく作成された証明書をインポートします、ですから proxmox-backup-clientはバックアップ時にそれを見つけて使用することができます。

```
# proxmox-backup-client key import-master-pubkey /path/to/master-public.pem 公開マスターキーを
"~/.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すべて配置し、バックアップジョブを実行します:

```
# proxmox-backup-client バックアップ etc.pxar:/etc
```

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

注意 暗号化キーがデフォルトのパスにある場合、--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ことをテストするには、バックアップからキーを復元します:

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

注意: このファイルを展開するために暗号化キーは必要ありません。ただし、暗号化キーがデフォルトの場所 (~/.config/proxmox-backup/encryption-key.json) に存在する場合、プログラムは暗号化キー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から移動するだけで、この問題は解決します。

6. 次に、先に生成したマスター・キーを使って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-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*****
パスワードの確認*****
```

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。そのため、キーは順序を付けて、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しておくことをお勧めしますが、迅速な災害復旧のために、簡単にアクセスできるようにしておく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・キーの紙のコピーを安全な場所に保管しておくことも検討してください。`paperkey`サブコマンドを使用すると、マスターキーのQRエンコード版を作成することができます。次のコマンドは、`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
```

7.7 データの復元

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。より重要なのは復元です。定期的にリカバリーテストを実施し、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.pcatl index.json |
| host/elsa/2019-12-03T09:35:01Z | 51790622048 | root.pxar catalog.pcatl 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リストするには、`--ns <ns>`オプションを使用します。

カタログを調べ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/コンソール設定"
```

復元コマンドを使用すると、バックアップから単一のアーカイブを復元できます。

```
# 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` ファイルを対象のパス '`'` 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 - 以下のようにになります。
```


7.7.1 インタラクティブ・リストア

個々のファイルをいくつか復元したいだけなら、対話型復元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 対話シェルの開始
pxar:/> ls
b.i.n.      ブート      デブ      その他      ホーム      lib      lib32
```

対話型復元シェルは、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、`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したりするのに使われるコマンドです。

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。カタログの価値はそこにあります。例えば

```
pxar:/> find etc/**/*txt --select
"/etc/X11/rgb.txt"
pxar:/> 選ばれたリストなど/**/*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ャンします。

`restore` コマンドを使用すると、バックアップ・アーカイブに含まれる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、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ます。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ャンし、`.conf`で終わるすべてのファイルをリストアします。

7.7.2 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取り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  lib64  lost+found  mnt
          mediaplayer  proc  run  srv
          root  sbin  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することができます。

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フェッチ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、ホストにネットワーク負荷とCPU負荷が追加され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`umount` コマンドを使用します：

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

7.8 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。

また、loginコマンドや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ーすることもできます：

```
# proxmox-backup-client login Password: *****
```

チケットを削除するには、ログアウトを発行してください：

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

7.9 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バックアップを作成することはできません。バックアップの所有者を変更する場合は、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、バックアップグループを含むデータストアのコンテンツセクションに移動し、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。

7.10 バックアップの剪定と削除

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できます：

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

注意このコマンドは、この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。これらはアクセス不能になり、回復できなくなります。

ルート・ネームスペースに保持されてい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：

--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>

過去<N>時間のバックアップを保持します。1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時間はカウントされません。

--キープ・デイリー <N>

過去<N>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は、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。

--キープウィークリー<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。本ソフトウェアはISO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。

--キープマンスリー<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は、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。

--キープ・イヤー・リー<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は、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。

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がどのようなアクションを取るか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
host/elsa/2019-12-04T13:20:37Z]
host/elsa/2019-12-03T09:35:01Z]
host/elsa/2019-11-22T11:54:47Z]
host/elsa/2019-11-21T12:36:25Z]
host/elsa/2019-11-10T10:42:20Z]
```

注意：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。

また、単一のスナップショットを刈り込み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

。

7.10.バックアップの剪定と削除

73

注: プルまたは同期ジョブを使用する場合、このフラグは同期されません。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。

7.11 ゴミ収集

`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。データストアからの未使用のバックアップデータの削除は、ガベージコレクションで実行されます。したがって、ガベージ・コレクション・タスクを定期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ます：

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

注意： このコマンドは既存のバックアップ・インデックス・ファイルをすべて読み取り、完全なチャンク・ストア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 store
ガベージコレクションを開始 GCフェーズ1を開始 (使用済みのチャンクをマーク)
GCフェーズ2の開始 (未使用チャンクの掃引) 完了率: 1、チャンク数219

,percentage done: 2, chunk count: 453
完了率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: 203
元のデータバイト数327160886391 ディスク・バイト:
52767414743 (16 %)
ディスクチャンク: 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`を使用して、またはProxmox Backup ServerのWebインターフェースからスケジュールすることもできます。

7.12 ベンチマークリング

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さまざまな指標を測定します。Proxmox Backupリポジトリ（リモートまたはローカル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。

`proxmox-backup-client`の`benchmark`サブコマンドを使ってベンチマークを実行できます：

注： 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](#)にのみ含まれます。

```
# 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
```

ペリフアイ速度: 783.43 MB/秒

TLS (バックアップの最大アップロード速度)	1267.41 MB/s (103%)
SHA256チェックサムの計算速度	2066.73 MB/s (102%)
ZStdレベル1圧縮速度	775.11 MB/s (103%)
ZStdレベル1の解凍速度	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%)

(前ページから続く)

注：出力表のパーセンテージはRyzen 7 2700Xとの比較です。

また、`--output-format` パラメータを渡すと、統計情報をデフォルトの表形式ではなく json 形式で出力することができます。

プロックスモックスVE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ストレージ - Proxmoxバックアップサーバ」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は、Proxmox VE 6.3からProxmox Backup ServerのネイティブAPIとWebインターフェースを統合しました。

Proxmox Backup Serverは、Datacenter -> Storageで追加できます。

8.2 Proxmox VEコマンドラインの使用

Proxmox VE ノードに'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、サーバアドレスを localhost とし、user1@pbs として接続するものとします。

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

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 パラメータ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。

バックアップ・サーバが自己署名証明書を使用している場合、証明書の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ーで次のコマンドを実行すると、フィンガープリントを取得できます：

```
# proxmox-backup-manager cert info | grep Fingerprint Fingerprint (sha256):
64:d3:ff:3a:50:38:53:5a:9b:f7:50:...:ab:fe
```

信頼関係を確立するため、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
```

、ストレージのステータスを表示できるなります：

名前	タイプ	合計	中古	利用可能	%
store2	pbs	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をバックアップとリストア参照):」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・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にインスパイアされています。このユーティリティは

.pxarフォーマットは、*Proxmox Backup Server*特有のニーズ、例えばハードリンクの効率的な保存に適合しています。このフォーマットは、高レベルの重複排除を実現することで、サーバー上の必要なストレージを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ーのアーカイブを作成します：

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

これにより、ソースフォルダの内容を含む `archive.pxar` という新しいアーカイブが作成されます。

注: `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名のアーカイブがすでに存在する場合は、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、デバイスの境界には従いません。この設計上の決定は、バックアップ用のアーカイブを作成するという主なユースケースに基づいています。特定の一時ファイルやシステム固有ファイルの内容をバックアップで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境界に従うようにするには、`--all-file-systems` フラグを使用します。

を渡すことで、特定のファイルやフォルダをアーカイブから除外することができます。

パラメータに `gitignore` 形式のマッチパターンを指定します。

例えば、`.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、次のように実行します：

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

シェルは `pxar` を起動する前にグロブ・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も可能ですが。これにより、より複雑なファイルの包含/除外動作を使用することができます。ただし、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とします。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。すべてのグロブパターンは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'
```

注: グロブ・マッチ・パターンの順序は重要で、後のものが前のものを上書きします。同じパターンを許可しても結果は異なります。

`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、代わりに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・ツリーのどのディレクトリにも作成、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、後のパターンが前のパターンを上書きします。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「[ノックアップの作成](#)」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、以下のコマンドでターゲット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ていない場合、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、フォルダのみをリストアするには、対応するグロブ一致パターンを追加パラメータとして渡すか、ファイルに保存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` ターゲット `/restore/target/etc` にリストアします。一致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 アーカイブの内容一覧

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
```

これは、アーカイブの基準として、各ファイルやディレクトリのフルパスを表示します。

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、指定されたマウントそのコンテンツにアクセスすることができます。

```
# cd /mnt
# ls
bin      dev  home  lib32  libx32          メディアオプト    root  sbin  sys  usr  boot  etc
lib      lib64  lost+found  mnt            proc  runsrv   tmp  var
```

テープバックアップ

ID	Content	Inventory	Actions
1			刷新 信息
2			刷新 信息
3	TAPESL4	writable (pve-backup)	刷新 信息
4	TAPESL4	writable (pve-backup)	刷新 信息
5	TAPESL4	writable (pve-backup)	刷新 信息
6	TAPESL4	writable (pve-backup)	刷新 信息
7	TAPESL4	writable (pve-backup)	刷新 信息
8	TAPESL4	writable (pve-backup)	刷新 信息
9			刷新 信息
10			刷新 信息
11			刷新 信息
12			刷新 信息
13			刷新 信息
14			刷新 信息
15			刷新 信息
16			刷新 信息
17			刷新 信息
18			刷新 信息
19			刷新 信息
20			刷新 信息

Content	Inventory	Name	State	Actions
TAPESL4	writable (pve-backup)	qdrive0	idle	写入 读取 清洁
TAPESL4	writable (pve-backup)	qdrive1	idle	写入 读取 清洁

ID	Content	Inventory	Actions
21			写入 读取
22			写入 读取
23			写入 读取
24			写入 读取

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、
- 別のメディア・タイプ（テープ）で、
- を追加の場所に移動します（テープをオフサイトに移動できます）。

ほとんどのリストア・ジョブでは、最後のバックアップ・ジョブから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、数年間データを保持する必要がある場合に特に当てはまります。

テープ・バックアップでは、保存されたデータにランダムにアクセスすることはできません。その代わりに、データをディスクにリストアしてから、再度アクセスする必要があります。また、テープをオフサイトに保存している場合（テープ保管サービスを使用して場合）、リストアする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。

性能と容量で若干有利な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、これらには大きな欠点があります:

- プロプライエタリ
- はるかに高い購入コスト

ですから、現在のところ、そ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)
- 結核1本あたりのコストは比較的低い
- コールドメディア
- 可動式 (金庫内に収納可能)
- 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 対応ハードウェア

*Proxmox Backup Server*はリニアテープオーブ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: Rustで書かれたカスタム・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。このドライバを古い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・バックアップでは使用しないでください。

10.2.1 駆動性能

現在のLTO-8テープの読み書き速度は最大360MB/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には、(最高速度でも) 最低でも9時間かかります。

データ転送速度を上げる唯一の方法は、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、單一回転ディスクでは、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+の考察

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回避します。Proxmox Backup Serverで初めて使用する、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、cliの'高速'フォーマットを使用してください：

```
# 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、**メディア最適化**がトリガーされます。

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ので、必ず'slow'メソッドを使用してください。

10.3 用語解説

テープラベル:

はテープを一意に識別するためのものです。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 Cartridge Label Specification](#)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ています。

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：物理的ラベルと関連する粘着剤は、それが貼られ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から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たテープのグループ（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。スタンドアロンのドライブもありますが、ドライブは通常テープ・ライブラリに内蔵されています。

テープチェンジャー

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:

1台以上のテープ・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）を含む記憶装置。

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。

在庫

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

カタログ

メディアカタログは、メディアコンテンツに関する情報を格納します。

10.4 テープクイックスタート

1. テープ・ハードウェア（ドライブとチェンジャー）の設定
2. 1つまたは複数のメディアプールの設定
3. テープカートリッジにラベルを貼りましょう
4. 最初のテープバックアップジョブを開始 ...

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（ロボット）はテープ・ライブラリーの一部で、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バーコード・リーダー、テープをロードする自動化された方法を備えています。

スタンダードアロン・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することができます：

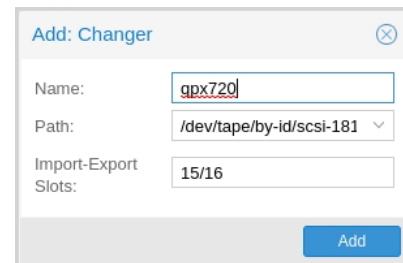
# ブロックスモックス・テープ・チェンジャー		
バス	ベンダー	モデル
/dev/tape/by-id/scsi-CC2C52	Quantum	Superloader3

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。

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。のような名前を使用してください。
/dev/sg0は再起動後に別のデバイスを指すかもしれません。



この操作は、テープ・バックアップの [Changers] タブに移動して [Add] をクリックすることで、GUIからも実行できます。

で最終的なコンフィギュレーションを表示できます：

# proxmox-tape changer config sl3	
名前	値
name	sl3
パス	

または、設定されているすべてのチェンジャー・デバイスをリストアップすることもできます(GUI の Changers タブで見ることができます)：

# ブロックスモックス・テープ・チェンジャー・				
リスト	バス	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か、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、以下の方法でチェンジャー・デバイスのステータスを照会してください：

Proxmox Backup ドキュメント、リリース 3.4.0-1

```
# proxmox-tape changer status sl3
```

エントリー種別	entry-id	changer-id
ドライブ	0	vtape1
スロット		
スロット	2	vtape2

(次ページへ続く)



(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。これらのスロットにあるテープにはアクセスできるため、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・ジョブでは使用されません。これらの特別なスロットは自動検出され、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。

注目すべきは、小型のテープ・ライブラリーにはそのようなスロットがないことです。メール・スロットと呼ばれるもののはありますが、そのスロットは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、「Mail Slot」をSCSIインターフェイス上に公開しないので、ステータス出力に表示されることはありません。

回避策として、通常のスロットの一部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。ソフトウェアはこ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、こ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
```

、ステータス出力に人工的なインポート／エクスポート・スロットが表示されます：

```
# proxmox-tape changer status sl3
| エントリー種別 | entry-id | changer-id |
|-----|-----|-----|
| ドライブ | 0 | vtape1 |
|-----|-----|-----|
| 輸出入 | | |
|-----|-----|-----|
| 輸出入 | | |
|-----|-----|-----|
| スロット | | |
|-----|-----|-----|
| スロット | 2 | vtape2 |
|-----|-----|-----|
| ... | ... | ... |
|-----|-----|-----|
| スロット | | |
```

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。

現在は以下の通りです：

- *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排出する必要がある一部のチェンジャーに必要です。

これらのオプションは`proxmox-tape`で次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, Storage / Disks, Tape Backup (which is selected), and Datastore. The main area is titled "Tape Backup" and contains tabs for Content, Inventory, Changers, Drives (which is selected), Media Pools, Encryption Keys, and Backup Jobs. Under the Drives tab, there's a table with columns: Name, Path, Vendor, Model, Serial, and Drive Num... The table lists several entries, including drive0, drive1, drive2, drive3, qdrive0, and qdrive1, each with its respective details.

Linuxはテープドライブを自動検出することができます:

```
# proxmox テープドライブ スキャン
[パス] | ベンダー | モデル |
[ /dev/tape/by-id/scsi-12345-sg ] [ IBM ] [ ult3580-td4 ]
```

The dialog box is titled "Add: Drive". It has fields for Name (qdrive0), Changer (qpx720), Drive Number (0), and Path (/dev/tape/by-id/scsi-018). There is also a dropdown menu for Changer and a "Add" button at the bottom.

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
```

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。のような名前を使用してください。
/dev/sg0は再起動後に別のデバイスを指すかもしれません。

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が必要なのは、テープ・ライブラリーに複数のドライブが含まれている場合だけです（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）。

で最終的なコンフィギュレーションを表示できます：

```
# proxmox-tape drive config mydrive
| 名称 | 値 | | |
|---|---|---|---|
| 名称 | マイドライブ |
|----|----|
| パス | || |
|----|----|
| チェンジャー | SL3 |
```

注意：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。

設定されているすべてのドライブを一覧表示するには

```
# proxmox-tape ドライブリスト
| 名称 | パス | チェンジャー | ベンダー | モデル |
|----|----|----|----|----|
| mydrive | /dev/tape/by-id/scsi-12345-sg | sl3 | IBM | ult3580-td4 |
```

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にのみ表示されます。テストでは、次
の方法でドライブのステータスを照会できます：

```
# proxmox-tape status --drive mydrive
| 名称 | 値 |
|----|----|
| ブロックサイズ | 0 |
|----|----|
| 密度 | LT04 |
|----|----|
| 圧縮 | 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 | 12月13日(金) 01:00:00 |
| bytes-written | 501.80 GiB |
|----|----|
| バイトリード | 4.00 MiB |
|----|----|
| ミディアムパス | 20 |
|----|----|
| 中程度の磨耗 | 0.12 |
|----|----|
| ボリュームマウント | 2 |
```

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。これがデフォルトです。

10.5.3 メディア・プール

The screenshot shows the Proxmox Tape Backup interface. On the left is a sidebar with navigation links: Dashboard, Configuration, Access Control, Remotes, Subscription, Administration, Shelf, Storage / Disks, and Tape Backup (which is selected). The main area is titled "Tape Backup" and contains tabs for Content, Inventory, Changers, Drives, Media Pools (selected), Encryption Keys, and Backup Jobs. Under "Media Pools", there are buttons for Add, Edit, and Remove. A table lists two entries:

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。1つまたは複数のバックアップ・ジョブがメディア・書き込み、順序付けられたテープ・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時間を定義します。以下の設定が可能です：

- 現在のメディアセットを使用してみてください（続行）。

この設定により、1つの大きなメディアセットが作成されます。これは非常にスペース効率に優れていますが（重複排除、未使用スペースなし）、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、データがランダムに

第10章. テープバックアップ

セット内のテープに分散されます。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ります。

また、メディア・セットが大きいと、1本のテープが損傷しただけで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：保存期間は、より新しいメディア・セットが存在する時点から始まります。

- 常に新しいメディアセットを作成します（常に）。

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、残りのスペースが未使用のままになる可能性があるためです。

この利点は、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、オフサイトの保管庫への移動がより便利になり、より速くリストアできます。

注：保存期間はメディアセットの作成時間から始まります。

- 指定されたカレンダーイベントがトリガーされると、新しいセットを作成します。

[systemd.time man](#) ページを参照してください。

例えば、`weekly`（または`Mon *-*-* 00:00:00`）という値を指定すると、毎週新しいセットが作成されます。

これは、スペース効率とメディア数のバランスです。

注：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、以下のイベントが新しいメディア・セットを割り当てることがあります：

- 必要なテープがオフラインになっている（テープライブラリを使用している）場合。
- 現在のセットには、破損または引退したテープが含まれています。
- 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。たとえば、インベントリに必要なメディア情報が含まれていない場合や、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指定します。

- メディアは常に上書きしてください。
- 指定された期間、データを保護します。

[systemd](#) をタイムスパンのように使って、2週間などの期間を指定します（[systemd.time manpage](#) を参照）。

- 決してデータを上書きしないでください。

ハードウェア暗号化

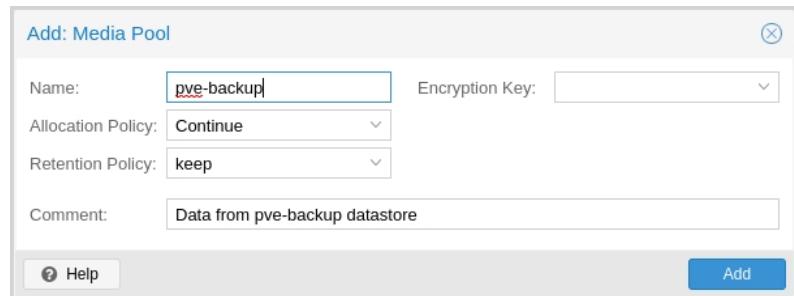
LTO-4（またはそれ以降）テープ ドライブはハードウェア暗号化をサポートしています。暗号化を使用するようにメディア プールを構成すると、テープに書き込まれるすべてのデータは構成されたキーを使用して暗号化されます。

こうすることで、権限のないユーザーがメディアからデータを読み取ることができなくなります。たとえば、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ツイストに保存され、パスワードを使ってキーを復元することができます。下が

一、キーを復元する必要が生じた場合に備えて、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、次のコマンドを入力します：

```
// proxmox-tape pool create <name> --drive <string> [OPTIONS].
# proxmox-tape pool create daily --drive mydrive
```

追加のオプションは、updateコマンドを使って後で設定できます：

```
# 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 ||]
[毎日| マイドライブ | 毎日 | 7日間 ]
```

10.5.4 テープバックアップの仕事

The screenshot shows the Proxmox Tape Backup interface. On the left is a sidebar with navigation links: Dashboard, Configuration, Access Control, Remotes, Subscription, Administration, Shelf, Storage / Disks, and Tape Backup (which is selected). The main area is titled "Tape Backup" and contains a table of backup jobs. The table has columns for Job ID, Datastore, Media Pool, Drive, Eject, Export, Latest Only, Schedule, Last Backup, Duration, Status, Next Run, and Comment. One job is listed: "weekly... pve-backup pve-backup drive0 No No No sat 18:15 - - 2021-04-17 18:15:00". Below the table are buttons for Add, Edit, Remove, Show Log, and Run now.

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・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・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。
- スケジュール仕事のスケジュール（[カレンダーイベントを参照](#)）

たとえば、データストアvmstore1のテープバックアップジョブを構成するには、次のようにします：

```
# proxmox-tape backup-job create job2 --store vmstore1 \
-プール yourpool -ドライブ yourdrive -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フラグを使用して、最新のスナップショットのみを含めることができます：

```
# proxmox-tape backup-job update job2 --latest-only
```

バックアップジョブは、電子メールを使用してテープ要求通知を送信したり、エラーを報告したりできます。通知ユーザーは

```
# proxmox-tape backup-job update job2 --notify-user root@pam
```

注：電子メールアドレスはユーザのプロパティです（「[ユーザ管理](#)」を参照）。

バックアップ後にドライブからテープを取り出すと便利な場合があります。スタンドアロン・ドライブでは `eject-media` オプションはテープをイジェクトし、次のバックアップがテープを使用できないようにします。

(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:

```
# proxmox-tape backup-job update job2 --eject-media
```

注: 失敗したジョブの場合、テープはドライブに残ります。

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ピックアップしてデータ保管庫に移動できます。

```
# proxmox-tape backup-job update job2 --export-media-set
```

注記: `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することができます。

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
```

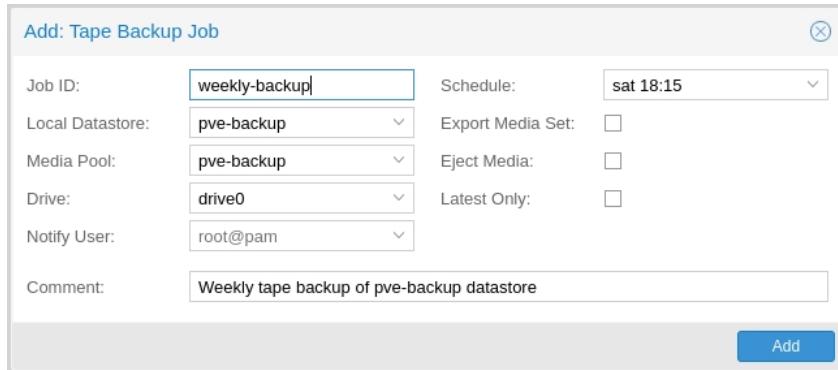
ジョブを削除するには

```
# 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）ネームスペースがテープ・バックアップに含まれます。`ns` で単一のネームスペースを指定し、`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`を指定しないと、すべての再帰名前空間を含めます。



これと同じ機能がGUIの**Tape Backup**の**Backup Jobs**タブにもあり、*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 管理部門

`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ます。便宜上、これを環境変数に設定することができます:

```
# export PROXMOX_TAPE_DRIVE= mydrive
```

その場合、コマンドから `--drive` パラメータを省略できます。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などからchangerパラメータを省略することもできます:

```
# ブロックスマックス・テープ・チェンジャーのステータス
```

は、`mydrive` ドライブに関連付けられているチェンジャデバイスのチェンジャステータスを表示する必要があります。

10.6.1 ラベルテープ

デフォルトでは、テープ・カートリッジはすべて同じように見えるので、固有の識別のためにラベルを貼る必要があります。まず、人間が読めるような文字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することができます。

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>].
```

-pool引数を省略すると、どのプールでもテープを使用できるようになります。

注意: 安全上の、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。

ラベルを読み返せば、成功を確認できます:

```
# proxmox-tape read-label
|名稱          |値|
|-----+-----+-----|
|チェンジャーID|vtape1|
|-----+-----+-----|
|ユーザーID     |||
|-----+-----+-----|
|時間          |1月6日（水）09:07:51 2021|
|-----+-----+-----|
|プール          |毎日|
|-----+-----+-----|
|media-set-uuid|||
|-----+-----+-----|
|media-set-ctime|Jan 6 Wed 09:07:51 2021|
```

注: すべてゼロの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。その後、ラベルの付いていないすべてのテープに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 [今すぐ実行] をクリックするか、コマンドを使用します：

```
# proxmox-tape backup <store> <pool> [OPTIONS].
```

以下のオプションが：

--eject-media	ジョブ完了時にメディアを排出します。
	通常、使用後はテープをイジェクトするのがよい方法です。こうすることで 、テープがドライブから取り外され、ホコリが付きにくくなります。
--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・ジョブが成功すると、使用済みメディア・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	バックアップする名前空間。
	特定のネームスペースのみ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	名前空間を再帰する深さ。
	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。

次のコマンドは、(カタログから) メディアコンテンツを一覧表示します：

```
# proxmox-tape メディアコンテンツ
+-----+-----+-----+-----+-----+
|label-text| pool | media-set-name | seq-nr | スナップショット |
+-----+-----+-----+-----+-----+
| メディアセット-uuid | | | | |
+-----+-----+-----+-----+-----+
| TEST01L8 | p2 | Wed Jan 13 13:55:55 2021 0 | | vm/201/2021-01-11T10:43:48Z | | | | | |
| | | | | |
| | ... | ... | | ... | | ... | | ... |
| | | | | | | | | | |
+-----+-----+-----+-----+-----+
```

リストアジョブはメディアセットからデータを読み取り、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復元

メディア・セット全体をリストアする必要はなく、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もあります。このような場合は、`snapshots`パラメータを使用します：

```
// 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。

注： 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1回以上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可能です。これには `namespaces` パラメータを使用します。パラメータの形式は次のとおりです：

```
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：

```
# 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。`snapshots` パラメータと組み合わせることで、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プすることもできます。

10.6.4 在庫の更新

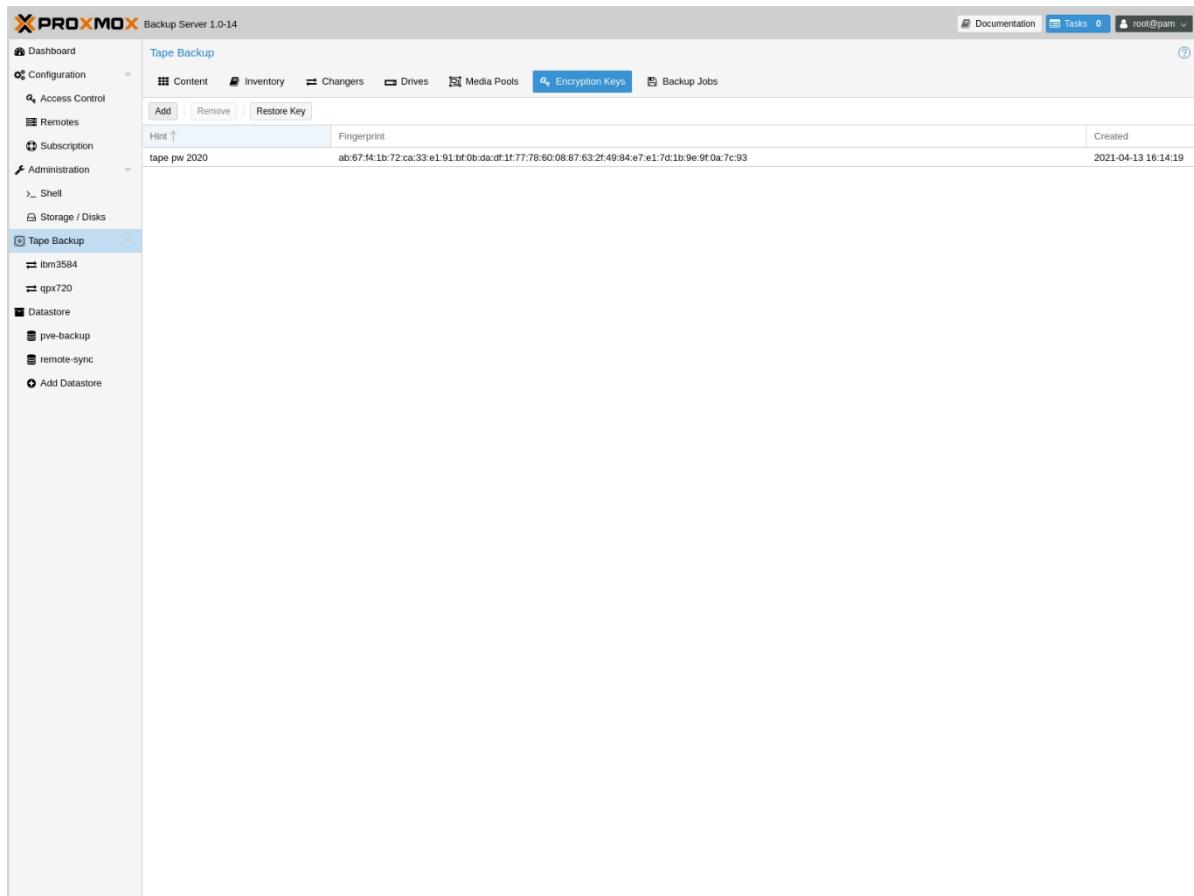
10.6.5 カタログの復元

既存のテープからカタログを復元するには、テープをドライブに挿入して実行するだけです：

```
# 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 暗号鍵の管理



Proxmox Backup Serverには、バックアップサーバー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**Tape Backup -> Encryption Keys**セクションまたは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、以下の手順に従います：

```
# proxmox-tape key create --hint "tape pw 2020" テープ暗号化キー
のパスワードです: *****
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一覧表示します：

```
# proxmox-tape キーリスト
| 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 |
```

暗号化キーの詳細を表示します：

```
# proxmox-tape key show 14:f8:79:b9:f5:13:e5:dc:...:b6:20:e1:7f:f5:0f
| 名称 | 値 |
| Kdf | スクリプト |
| 作成 | 1月23日（土）14:47:21 2021 |
| 修正済み | 1月23日（土）14:47:21 2021 |
```

(次ページへ続く)

指紋 14:f8:79:b9:f5:13:e5:dc:...:b6:20:e1:7f:f5:	
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。次のコマンドは、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-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。次に実行します：

```
# 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。それ以降のリストジョブでは、利用可能なキーが自動的に使用されます。

10.6.7 テープクリーニング

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装填することによって行われますが、スタンドアロン・ドライブの場合は手作業となります。

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。たとえば、当社のテープ・ライブラリのスロット3にはクリーニング・カートリッジがあります：

# proxmox-tape changer status sl3			
エントリー種別 entry-id changer-id			
ドライブ 0 vtape1			
スロット			
スロット 2 vtape2			
スロット 3 CLN001CU			
... ..			

クリーニング作業を開始するには、単に実行します：

```
# ブロックスマックス・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を探す（スロット3）
- ドライブから現在のメディアをアンロード（スロット1に戻る）
- クリーニング・テープをドライブにセット
- ドライブクリーニングの実行
- クリーニングテープを外す（スロット3へ）

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、法的または保護有用な場合があります。

これを使用する場合は、保持ポリシーが*keep*のメディア・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、専用のメディアアプールを使用します。同じメディアアプールに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 シングル継続メディアセット

最もシンプルな設定：メディアセットは常に継続され、期限切れになることはありません。

配分方針：
続ける

保持ポリシー：
キープ

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、冗長性がないため、1本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する場合は、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、曜日ごとに別々のメディア・プールを持つことで解決できます。

配分方針：
は月曜日'のプールには'mon'、'火曜日'のプールには'tue'といった具合です。

保持ポリシー：
オーバーライト

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、簡単にオフサイトに移動できます。

10.8.3 異なるポリシーを持つ複数のプール

また、複数のメディアプールを異なる割り当てポリシーや保持ポリシーで構成するなど、複雑なセットアップも可能です。

例えば、2つのメディアプールがあるとします。1つ目は、毎週割り当てられ、数週間保持されるように構成されています：

配分方針：

モン

保持ポリシー：

3週間

2つ目のプールは、有効期限のない年間割り当てで構成されています：

配分方針：

毎年

保持ポリシー：

キープ

プルーンの設定やテープバックアップのスケジュールと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、以下のステップバイステップガイドでは、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方法を説明します。

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 データストアの設定

[インストール](#)の説明したように、新しいProxmox Backup Serverをセットアップしたら、まずテープがリストアできるようにデータストアをセットします：

1. **管理**→【ストレージ/ディスク】に移動し、データストアとして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**タブまたは**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](#)を使って同じタスクを実行することもできます。詳細はデータストアのドキュメントを参照してください。

10.9.2 テープ・ドライブのセットアップ

- リストアするメディアと同じテープ・ドライブやチェンジャーが正しく動作していることを確認してください。
- テープ・チェンジャーとテープ・ドライブをバックアップ・サーバーに接続します。これらはLinuxによって自動的に検出されるはずです。使用可能なドライブのリストは

```
# proxmox テープドライブ スキャン
[バス | ベンダー | モデル]
/dev/tape/by-id/scsi-12345-sg | IBM | ult3580-td4 ||
```

で使用可能なチェンジャーのリストを取得できます：

```
# ブロックスマックス・テープ・チェンジャー・
[スキャン | ベンダー | モデル]
/dev/tape/by-id/scsi-CC2C52 | Quantum | Superloader3 ||
```

詳しくは、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。

- テープチェンジャーがある場合は、Proxmox Backup Server の Web インターフェイスで、[Tape Backup] -> [Changers] に進み、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、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復元

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復元するには、次のようにします：

- ラベルに表示されている) 最初のテープをテープ・ドライブに挿入するか、テープ・チェンジャーが使用可能な場合は、テープ・チェンジャーを使用して正しいドライブにテープを挿入します。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。
- データの復元手順は、スタンドアロン・テープ・ドライブとチェンジャーのどちらを使用しているかによって若干異なります：
 - チェンジャーにとって、手続きはです：
 - 復元メディアセットのテープをすべて挿入します。
 - ウェブGUIでチェンジャーをクリックし、[Inventory]をクリックして、[Restore Catalog]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：

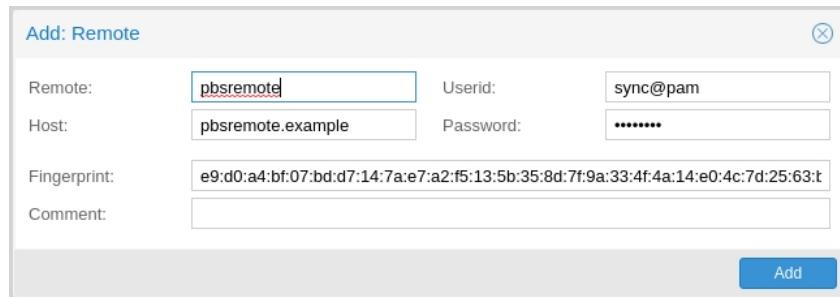
1. メディア・セットの最初のテープを挿入します。

2. カタログをクリックしてください。
3. テープをイジェクトし、メディア・残りのテープについて手順を繰り返します。
4. 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・データストアを選択し、[Restore] を押します。
5. データがリストアされたデータストアにアクセスすると、[Content] 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で、新しいユーザーを作成するか、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 の下に、必要なパーミッションを追加します、例: **DatastoreBackup**。
6. 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や、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。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することができます。
 - 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とコンテナをその構成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の詳細については、Proxmox VE ドキュメントの [Restore](#) の章を参照してください。

リモコンと同期の管理

11.1 リモート

リモートとは、Proxmox Backup Server の別個のインストールと、そのインストール上のユーザーを指します。リモートから同期ジョブを使ってデータストアをローカルのデータストアに同期できます。リモートは、Web インターフェイスの【設定】->【リモート】で設定できます。または、リモート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ファイルに保存されます。



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 Fingerprint (sha256):
64:d3:ff:3a:50:38:53:5a:9b:f7:50:...:ab:fe
```

上記の情報を使用して、**Remotes** 設定パネルから、またはコマンドを使用してリモートを追加できます：

```
# proxmox-backup-manager remote create pbs2 --host pbs2.mydomain.example --userid sync@pam -->パスワード 'SECRET' -指紋 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サブコマンドを使用します：

```
# proxmox-backup-manager remote update pbs2 --host pbs2.example
# proxmox-backup-manager リモートリスト
|名前| ホスト          |ユーザーID      |指紋
|pbs2| pbs2.example| sync@pam |64:d3:ff:3a:50:38:53:5a:9b:f7:50:...:ab:|
# proxmox-backup-manager remote remove pbs2
```

11.2 シンク・ジョブズ

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:	<input type="checkbox"/>
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取り込むように設定されます。同期ジョブは Web インターフェイスの **Datastore** パネルの **Sync Jobs** タブやデータストア自体で管理できます。また、`proxmox-backup-manager sync-job` 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、GUI から手動で開始するか、スケジュール（カレンダーイベントを参照）を指定して定期的に実行することができます。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。`verified-only` フラグまたは `encrypted-only` フラグを設定すると、同期ジョブをそれぞれ検証済みまたは暗号化済みのバックアップスナップショットに制限できます。これは特に、信頼性の低いリモートバックアップサーバにバックアップを送信する場合に役立ちます。

```
# 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 sync-job list
+-----+-----+-----+-----+
| id   | store| remote| remote-store| schedule |
+-----+-----+-----+-----+
| pbs2-local | local | pbs2      | ローカル      | Wed 02:30 |
+-----+-----+-----+-----+
# 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.Read` を `/remote/{remote}/{remote-store}` パスで実行します。
2. 少なくともローカルのターゲット・データストア (`/datastore/{store}`) 上の `Datastore.Backup`

注：同期ジョブが同期できるのは、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です。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。所有者オプションが設定されていないか（デフォルトが `root@pam` に設定されている）、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ようにバックアップタイプを設定します：

```
# 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：

```
# proxmox-backup-manager sync-job update ID --group-filter group:vm/100
```

- 完全なグループ識別子と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。

グループフィルタの前に exclude: を付けることで、グループフィルタを反転させることができます。

- マッチを除いた正規表現の例:

```
# proxmox-backup-manager sync-job update ID --group-filter exclude:regex:'^vm/1d{2,3}$'
```

インクルードフィルタと除外フィルタの混合には、以下のルールが適用されます:

- フィルタなし: すべてのバックアップグループ
- include: インクルードフィルタにマッチするもののみ
- exclude: 除外フィルタにマッチするものを除くすべて
- 両方: インクルードフィルタにマッチするもの。

注: 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。

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のサブツリーという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: リモート名前空間アンカー (デフォルト: ルート名前空間)
- ns: ローカル名前空間アンカー (デフォルト: ルート名前空間)
- max-depth: リモート名前空間アンカーのサブ名前空間を再帰的に反復処理するかどうか (デフォルト: なし)

max-depthを0に設定すると、再帰なしで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名前空間サポートの有無に依存します:

- remote-nsをルート・ネームスペース以外に設定する場合、リモートはnamespacesをサポートし、remote-nsから始まる完全再帰をサポートする必要があります。
- 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: 後方互換モードでは、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。

たとえば、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、最大でもう1つの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 ns にマップされます。例えば、リモート

側に以下の名前空間があります：

- *location_a*
 - 場所_a/部署_x
 - * 場所_a/部署_x/チーム_ワン
 - * 場所_a/部署_x/チーム_2
 - 場所_a/部署_y
 - * 場所_a/部署_y/チーム_ワン
 - * 場所_a/部署_y/チーム_2
- ロケーション

に設定され、remote-*location_a/department_x*に設定され、nsが*location_a_dep_x*に設定された結果、同期ターゲットのネームスペース・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って) 同期されない残りのリモートの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ているがローカルにしか存在しない名前空間は消滅したものとして扱われ、削除されます（同期ジョブの所有者に十分な権限がある場合）。

注釈 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避けるために、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でrate-in設定することで、ブ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 sync-job update ID --rate-in 20MiB
```

プッシュ方向のシンクジョブには、代わりにレートアウトオプションを使用してください。

11.2.3 同期方向プッシュ

同期ジョブはブ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、プッシュ側で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対して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、remove vanished フラグが設定されている場合、同じターゲットにプッシュして

いる同期ジョブがお互いのスナップショットやグループを削除してしまう可能性があります。

またはバックアップ時間がインクリメンタルでない場合はスナップショットをスキップします。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構成で指定されたユーザによって所有されているためです。

プッシュ同期ジョブには以下の権限が必要です：

1. Remote.Auditは/remote/{remote}に、Remote.DatastoreBackupは/remote/にあります。
{remote}/{remote-store}/{remote-ns}のパスまたはサブネーム空間。
2. 同期ジョブの所有者の場合は、ローカル・ソース・データストア・ネームスペース (/datastore/{store}/{ns}) の少なくとも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削除します。上記のように、プッシュ方向の同期ジョブごとに専用のリモートを使用するようにしてください。
4. Remote.DatastoreModify on /remote/{remote}/{remote-store}/{remote-ns} path to remove vanished namespaces.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。以下に示す意味を考慮してください。

注： 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・データストアのネームスペース全体を削除できます。remote.cfgで設定されたユーザがリモート側で制限された権限を持っていることを確認してください。

注： プッシュ方向の同期ジョブには、リモートのProxmox Backup Serverインスタンスでネームスペースのサポートが必要です（最小バージョン2.2）。

保守作業

12.1 剪定

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なければなりません。

注意所定のデータチャンクに格納された機密情報は、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そのデータチャンクを参照していないなくても、ガベージコレクションによって削除されるまで、そのデータチャンクは存在し続けます。

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することができます。これらのファイルは、チャンクの生データを（クライアント側またはサーバ側で）読み取ることで、まだ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すべて削除し、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を含むファイルベースのバックアップを使用する場合は、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、次のガベージ・コレクションの実行時に削除マークが付けられ、猶予期間後に次の実行で削除されます。

剪定には以下の保持オプションがあります：

キープラスト <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。

時間毎 <N>

過去<N>バックアップを保持します。1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時間はカウントされません。

キープデイリー <N>

過去<N>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は、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。

キープウィークリー<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。本ソフトウェアはISO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。

キープマンスリー<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は、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。

年間 <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は、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も新しい場合を除き、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 プルーンシミュレーター

内蔵のプルーン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 プルーンの仕事

The screenshot shows the Proxmox Backup Web Interface for Datastore 'store1'. The top navigation bar includes 'Summary', 'Content', 'Prune & Go' (which is active), 'Sync Jobs', 'Verify Jobs', 'Options', and 'Permissions'. Below the navigation is a 'Garbage Collection' section with a 'Start Garbage Collection' button and a schedule of '21:00'. The main area is titled 'Prune Jobs' with tabs for 'Add', 'Edit', 'Remove', 'Show Log', and 'Run now'. A table lists two prune jobs:

Job ID	Namespace	Max. Depth	Schedule	Keep						Last Prune	Duration	Status	Next Run	Comment
				Last	Hourly	Daily	Weekly	Monthly	Yearly					
s-7d...	dc-usa		daily	3	13	8	11	1	-	-	-	2022-11-29 00:00:00		
s-81...	dc-eu-central		hourly	3		8	11	9	2022-11-28 14:00:00	<0.1s	✓ OK	2022-11-28 15:00:00		

剪定ジョブは、データストアまたはそのサブセットを定期的に剪定するように構成されます。Webインターフェイスの

Datastoreパネルの**Prune & GC**タブまたはDatastore自体のタブでPruneジョブを管理できます。また、`proxmox-backup-manager prune-job`で管理することもできます。

コマンドを使用します。pruneジョブの設定情報は/etc/proxmox-backup/prune.cfgに保存されます。新しい剪定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、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。

各剪定ジョブには、保持、範囲の制限、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データストア。

ns <名前空間

プルーンの特定のネームスペースに制限します。

最大深度 <N

構成されたネームスペースの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ニングするには0を指定します。このパラメータを省略すると、以下の完全な深さまでスキャンされます。

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:	(dropdown)
Keep Daily:	13	Keep Weekly:	8
Keep Monthly:	8	Keep Yearly:	(dropdown)
Comment:	Work-day pruning schedule for EU West namespace		

スケジュール

このジョブを自動的に起動するカレンダイベント間隔を設定します。ジョブを手動でのみトリガしたい場合は省略。

キープエックス

上記の様々な保持オプションの説明をご覧ください。

無効にする

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場合に設定します。

コメント

仕事について、例えばその意図について短いコメントを追加することができます。

12.1.3 手動剪定

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

12.1.剪定

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](#)で説明したproxmox-backup-clientのpruneサブコマンドを使用するか、データストアの**Content**タブに移動し、関連するバックアップ**Actions**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、バックアップを毎日行い、保存期間を年とし、保存されるバックアップの間隔は徐々に長くなっていくと仮定します。

- **keep-last:** 3 - 毎日のバックアップだけであっても、管理者は大きなアップグレードの直前や直後に追加のバックアップを作成したくなるかもしれません。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 - 每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 - 少なくとも1日をカバーするkeep-lastと合わせて、少なくとも2週間分のバックアップを確保します。
- **keep-weekly:** 8 - 少なくとも2ヶ月分の週次バックアップを確保します。
- **keep-monthly:** 11 - 前回の保管設定と合わせて、少なくとも1年間の月次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、合計で少なくとも10年間をカバーすることになります。

保存期間が不要に長い場合は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 ゴミ収集

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。

未使用領域を定期的にクリーンアップするために、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、1週間ごとのスケジュールを設定するとよいでしょう。

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るのではなく、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、他のバックアップ（たとえ実行中のバックアップであっても）がそのチャンクを参照している可能性があるため、そのチャンクと関連付けられたチャンクを直接削除することはできません。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ャ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、削除処理が完了するまで新しいバックアップがブロックされるため、データストア全体をロックすることは実行不可能です。

そのため、Proxmox Backup Serverは、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的バックアップチャックを特定し、削除するためにガベージコレクション(GC)プロセスを使用します。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、これらのチャッ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：

- 第1段階（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ックファイルのアクセスタイム (atime) が更新されます。

- 第2段階（スイープ）：

タスクはすべてのチャックを繰り返し、そのファイル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では、カットオフタイムより古いアクセスタイムを持つチャックファイルは、どのバックアップスナップショットのインデックスからも参照されておらず、現在実行中のバックアップジョブの一部でもないと見なされます。したがって、これらのチャックは安全に削除できます。

猶予期間内のチャックは削除されず、ガベージコレクションタスクの最後に *Pending removals* として記録されます。

注：バックアップチャック削除の猶予期間は任意ではなく、ファイルシステムが通常デフォルトで `relatime` オプションでマウントされていることに起因しています。これにより、最終アクセス以降にファイルが変更された場合、または最終アクセスが24時間以上前の場合にのみ `atime` プロパティが更新されるため、パフォーマンスが向上します。

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で [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し、`status` サブコマンドを使用して [ガベージコレクション](#) に関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からもアクセスできます。

12.2.3 GC予定

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ありません。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 編集 データストアの GC スケジュール は、
を使用して することができます。
`proxmox-backup-manager datastore update <datastore> --gc-schedule <schedule>` CLI コマンドまたは Web UI の各データストアの **[Prune & GC]** タブの編集ウィンドウ。

GC 2011 のスケジューリングは、[Calendar Events](#) フォーマットを使用します。

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` データストアの更新を使用してスケジュールをクリアすることで、自動 GC 2011 の実行を無効にできます。

`<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

一カイブする場合などに便利です。

12.3 検証

The screenshot shows a configuration dialog for a verification job. The 'Datastore' is set to 'store1'. The 'Schedule' is set to 'daily'. The 'Skip verified snapshots' checkbox is checked. The 'Re-Verify After (days)' field is set to '30'. There is a 'Comment' field which is empty. At the bottom, there is a 'Help' button and a blue 'Add' button.

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: 前回の検証が成功した場合でも、少なくとも毎月すべてのバックアップを再検証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、すべてのバックアップを復元するのがよい方法です。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.

12.4 お知らせ

Proxmox Backup Serverは、自動的にスケジュールされた検証、ゴミ収集、および同期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や書き込みがブロックされるため、管理者は基礎となるストレージなどでメンテナンス作業を安全に実行できます。

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し、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入ることができます。

ホストシステム管理

Proxmox 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。つまり、Debian パッケージの全範囲にアクセスでき、基本システムは十分に文書化されています。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の入手できます。さらに、Proxmox関連の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について説明します。Proxmox Backupと異なる点や、Proxmox Backupでよく使わ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可能です。また、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、高性能なシステムを実現することもできます。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を、CPUやメモリの負荷を抑えながら、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。
- 信頼できる
- 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
- スナップ写真
- 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
- 繼続的な完全性チェック
- 大容量収納に対応した設計

- 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ので、最初は少なくとも8GBを用意することをお勧めします。実際には、ハードウェア/予算に見合うだけの量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。

専用のキャッシュおよび/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（たとえば、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: 独自のキャッシュ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アダプタか、ITフラッシュされたLSIコントローラのようなものをお勧めします。

13.1.2 ZFS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ています。ZFSを管理する主なコマンドは`zfs`と`zpool`です。両コマンドには広範なマニュアルページが付属しています：

```
# 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) 以上でなければなりません。

```
# zpool create -f -o ashift= 12 <pool> <device>.
```

RAID-0で新しいプールを作成

ディスク1枚以上

```
# zpool create -f -o ashift= 12 <pool> <device1> <device2>.
```

RAID-1で新しいプールを作成

ディスク2枚以上

```
# zpool create -f -o ashift= 12 <pool> mirror <device1> <device2>.
```


RAID-10で新しいプールを作成

最低4ディスク

```
# zpool create -f -o ashift= 12 <pool> mirror <device1> <device2> mirror <device3> <device4>.
```

RAID-1で新しいプールを作成

ディスク3枚以上

```
# zpool create -f -o ashift= 12 <pool> raidz1 <デバイス1> <デバイス2> <デバイス3>
```

RAID-2で新しいプールを作成

最低4ディスク

```
# zpool create -f -o ashift= 12 <pool> raidz2 <デバイス1> <デバイス2> <デバイス3> <デバイス4>
```

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も可能です(SSDを使用)。<デバイス>については、「RAID*を使用した新しいプールの作成」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> cache <cache_device>.
```

ログ(ZIL)付きプールを新規作成

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)。<デバイス>については、「RAID*を使用した新しいプールの作成」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約半分であるべきなので、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>.
```

故障したデバイスの変更

```
# zpool replace -f <プール> <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> <old zfs partition> <new zfs partition>
```

メモ: *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 で:

```
# proxmox-boot-tool format <新しい ESP>
# proxmox-boot-tool init <新しい ESP>
```

注: *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。

食事をしながら:

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。

デーモンの設定は、`/etc/zfs/zed.d/zed.rc` ファイルで、お好みのエディタを使って行います。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="root"
```

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、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: `zfs_arc_max` の値が `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。この場合、`/etc/modprobe.d/zfs.conf` の設定を更新する必要があります：

```
オプション zfs zfs_arc_min=8589934591 オプション 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、スワッピング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` の値を下げるこもできます。サーバに適した値は 10 です：

```
# sysctl -w vm.swappiness= 10
```

スワップを永続的にするには、`/etc/sysctl.conf`をお好みのエディターで開き、以下の行を追加：

```
vm.swappiness= 10
```

表1: Linuxカーネルのスワッピング・パラメータ値

値	戦略
<code>vm.swappiness= 0</code>	カーネルは「メモリ不足」状態を回避するためだけにスワップを行います。
<code>vm.swappiness= 1</code>	スワッピングを完全に無効にすることなく、最小限のスワッピング。
<code>vm.swappiness= 10</code>	十分な量の場合、パフォーマンスを向上させるために推奨されることもあります。 メモリはシステム内に存在します。
<code>vm.swappiness= 60</code>	デフォルト値。
<code>vm.swappiness= 100</code>	カーネルは積極的にスワップを行います。

ZFS圧縮

圧縮を有効にするには

```
# zpool set compression= lz4 <pool>
```

lz4 アルゴリズムの使用を推奨します。*lzbz*, *zstd*, *gzip-N* (*N*は圧縮率を表す1～9の整数で、1が最速、9が最良の圧縮率)などの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性能が向上することもあります。

でいつでも圧縮を無効にできます:

```
# zfs set compression= off <dataset>
```

この変更の影響を受けるのは新しいブロックのみです。

ZFS専用デバイス

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の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: 特別な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。
特別なデバイスはプール全体の障害点となります。

警告プールに特別なデバイスを追加すると、元に戻すことはできません!

特別なデバイスとRAID-1でプールを作成するには:

```
# zpool create -f -o ashift= 12 <pool> mirror <device1> <device2> special mirror <device3>.  
'-<デバイス4'
```

既存のRAID-1プールに特別なデバイスを追加します:

```
# zpool add <pool> special mirror <device1> <device2>.
```

ZFSデータセットでは、*special_small_blocks=<size>*プロパティを公開しています。*size*には、小さなファイル・ブロックを特別なデバイスに格納しないようにする0、または512Bから128Kの範囲の2のべき乗を指定します。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・ブロックが特別なデバイス上に割り当てられます。

重要: *special_small_blocks*の値が*recordsize*以上（デフォルトではデータセットの128K）に書き込まれると、すべてのデータが特別なデバイスに書き込まれますので、注意してください!

プールで*special_small_blocks*プロパティを設定すると、すべての子ZFSデータセットでそ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選択します）。

プール全体で4Kブロックより小さいすべてのファイルをオプトインします:

```
# zfs set special_small_blocks= 4K <pool>
```

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 <pool>/<filesystem>
```

単一のデータセットに対する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>
```

トラブルシューティング

壊れたキャッシュファイ

ル

`zfs-import-cache.service` は、ZFSキャッシュを使用して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た場合、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、実行します:

```
# zpool set cachefile= /etc/zfs/zpool.cache P00LNAME
```

その後、`initramfs` を更新します:

```
# update-initramfs -u -k all
```

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・ブートローダー

Proxmox Backup は現在、インストーラで選択したディスクの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。

ZFS をルートファイルシステムとしてインストールした 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 上にインストールされたシステムにも当てはまります)。

13.2.1 インストーラーが使用するパーティション方式

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次のとおりです:

- 1MBのBIOSブートパーティション (gdiskタイプEF02)
- 512 MB 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スペースにまたがる第3 のパーティション。

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、512MB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。レガシー BIOS システム、およびセキュアブートが有効な EFI システムでは grub が使用され、セキュアブートのない 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を指すように設定されます。

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、非 EFI システムで ZFS で root を使ったインストール)で、選択した全てのディスクの 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を使ったESPのコンテンツの同期

`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、同期するために使用されるユーティリティです。これは、特定のカーネルバージョンを全ての ESP にコピーし、`vfat` フォーマットの ESP からブートするように、それぞれの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、小さな別個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 (See: [Booting ZFS on root with grub](#))。

冗長性のあるセットアップでは、すべてのディスクはインストーラによって ESP でパーティション設定されます。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ブートできない場合でも、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、システムがクラッシュした場合に `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。

`/etc/fstab` は、プライマリブートデバイスが故障した場合に備えています。

`proxmox-boot-tool` は以下の処理します：

- 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すべての 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メンテナンス・タスクにおけるコンフィギュレーションの同期化
- 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) 現在

```
# proxmox-boot-tool status
```

設定されているESPとその状態は、以下を実行することで確認できます：

同期ESPとして使用するための新しいパーティションの設定

パーティションを同期された 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、`proxmox-kernel-helper` の `proxmox-boot-tool` を使うことができます。

警告: `format` コマンドは<パーティション>をフォーマットします。正しいデバイス/パーティションを渡してください！

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、実行します：

```
# proxmox-boot-tool format /dev/sda2
```

`dev/sda2` にある既存のマウントされていない ESP を Proxmox Backup のカーネル更新同期メカニズムに含めるように設定するには、以下を使用します：

```
# proxmox-boot-tool init /dev/sda2
```

または

```
# proxmox-boot-tool init /dev/sda2 grub
```

セキュアブートをサポートするためなど、`systemd-boot` ではなく `Grub` で初期化するように強制します。

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`

マンドはまた、設定されているすべてのESPのリフレッシュを自動的にトリガします。

すべてのESPでの設定の更新

すべ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すべてのESPを同期させておくには、以下を実行するだけです：

```
# proxmox-boot-tool refresh
```

(ext4またはxfsのシステムでroot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：

- 現在実行中のカーネル
- パッケージ更新時に新しくインストールされるバージョン
- インストール済みの最新カーネル
- 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

手動でカーネルをブート可能にする方法

特定のカーネルとinitrdイメージをブート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 カーネル追加。

たとえば、以下を実行して、ABIバージョン5.0.15-1-pveのカーネルを、すべてのESPにインストールして同期しておくカーネルのリストに追加します：

```
# 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：

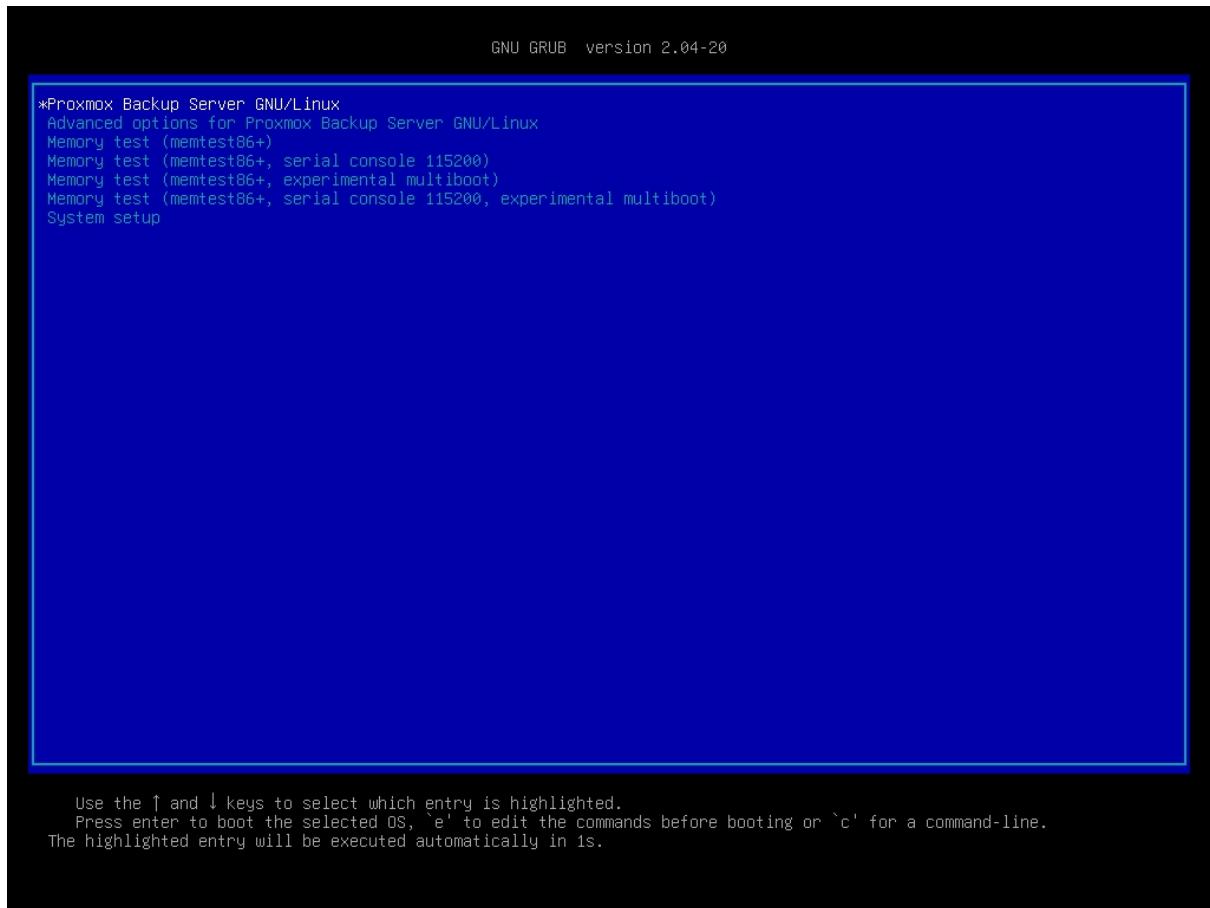
```
# proxmox-boot-tool kernel list 手動で選択
したカーネル: 5.0.15-1-pve
自動的に選択されたカーネル5.0.12-1-pve
4.15.18-18-pve
```

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：

```
# proxmox-boot-tool kernel remove 5.0.15-1-pve
```

Note: 手動でカーネルを追加・削除した後、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 使用するブートローダーの決定



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確実な方法は、Proxmox Backupノードのブートプロセスを観察することです。

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。



実行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
```

EFI 变数がサポートされていないというメッセージを返した場合、grub は BIOS/Legacy モードで使われます。出力に以下のような行が含まれる場合、grub は UEFI モードで使われます。

```
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(\|\|bootx64.efi)
```

以下のコマンドを実行することで、proxmox-boot-toolが設定されているかどうかを調べることができます：

```
# proxmox-boot-tool status
```

13.2.4 食事

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](#)）。

構成

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 proxmox-boot-tool を使っているシステムは
更新グラブ

13.2.5 システムブート

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から直接カーネルを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ています。

構成

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/ ディレクトリにあるそれ自身のファイルに置かれます。

`entry.conf`の例は以下のようになります（/はESPのルートを指します）：

タイトル	ブロックス
バージョン	5.0.15-1-pve
オプション	root=ZFS=rpool/R00T/pve-1 boot=zfs
リナックス	/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箇所で変更できます：

食事

カーネルコマンドラインは/etc/default/grubファイルの変数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/etc/default/grubのすべてのlinuxエントリに追加されます。`/boot/grub/grub.cfg`。

システムドブート

カーネルコマンドラインは `/etc/kernel/cmdline` に1行として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`のすべ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 次のブートのためにカーネルバージョンを上書きします。

現在デフォルトでない選択するには、以下のいずれかの方法があります：

- 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用します。
- `proxmox-boot-tool`を使用して、システムをカーネルバージョンに一度または永久に（ピンがリセットされるまで）固定します。

これは、新しいカーネルバージョンとハードウェア間の非互換性を回避するのに役立つはずです。

注意：このようなピンはできるだけ早く取り外して、最新のカーネルからのすべてのセキュリティパッチがシステムにも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：ピン留め機能は `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。

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、最初にバージョンを固定する原因となった問題が、更新されたカーネルによって解決され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
```

固定されているバージョン設定を削除するには、`unpin` サブコマンドを使用します：

```
# 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`。これはブート時にすでに自動的に行われる所以、手動で起動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 上のコンテンツと同期させる必要があります。

Tip: `proxmox-boot-tool` が管理するシステムで対話的にツールを呼び出す場合、自動的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、署名付きパッケージとproxmox-boot-toolの統合により、セキュアブートがすぐにサポートされるようになりました。

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 フォールバックブートローダとMOKManager、Proxmoxにより署名済み)
- grub-efi-amd64-signed (Grub EFI ブートローダ、Proxmoxにより署名済み)
- 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)

他の署名済みの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のみ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、自動的に上記のパッケージがすべて含まれるようになります。

セキュアブートの仕組みやセットアップのカスタマイズ方法の詳細については、[wikiをご覧ください](#)。

既存のインストールをセキュアブートに切り替える

警告正しく行わないと、場合によってはインストールが起動不能になることがあります。ホストを再停止すると、セキュアブートが利用可能な場合は、余分な操作なしで自動的にセットアップが行われます。Proxmox Backupホストが動作し、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!

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、システムがすべて最新であることを確認してください。次に、上記の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、デフォルトのシム経由で起動するために必要なEFIブートエントリーを自動的に作成します。

システムドブート

systemd-bootをブートローダとして使用する場合([使用するブートローダを決定するを参照](#))、追加の設定が必要です。これはProxmox BackupがZFS-on-rootでインストールされた場合のみです。

後者を確認するには

```
# 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列にはzfsが含まれているはずです：

ターゲット・ソース	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) をなりません。これはlsblkコマンドは以下のようになります：

```
# lsblk -o +FSTYPE
```

出力はようになるはずです：

名前	マウントポイント	fstype	
スダ	8:0	32G 0 ディスク	
-sda1	8:1	0 1007K 0 部	
-sda2	8:2	0 512M 0部	バット
-sda3	8:3	0 31.5G 0部	zfs_member
データベース	8:16	0 32G 0ディスク	

| └─sdb1 8:17 0 1007K 0部

(次ページへ続く)

—sdb2	8:18	0 512M 0 パート	vfat
—sdb3	8:19	0 31.5G 0 パート	zfs_member

(前ページから続く)

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 から起動するように適切にセットアップする必要があります。このコマンド例として sda2 (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チェックできます:

```
# efibootmgr -v
```

このリストには、次のような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)
[...].
```

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 ファームウェアセットアップユーティリティでセキュアブートを有効にすることができます。

再起動すると、proxmox という新しいエントリが UEFI ファームウェアブートメニューで選択できるようになっていて、署名済みの EFI shim を使って起動するようになっているはずです。

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、(ファームウェアでサポートされれば) カスタムブートエントリとしてファイル | proxmox | shimx64.efi を追加することで、手動で追加してみることができます。

Note: いくつかの UEFI ファ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 の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っていない場合に起こります。可能であれば、UEFI ファ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に追加して 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とともに出荷され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、特定のバージョンのカーネルイメージで信頼されています。

DKMS または手動でビルドしたモジュールなど、他のモジュールをロードするには、セキュアブートスタックに信頼された鍵で署名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 を使用してそれらをマシンオーナーキー (MOK) として登録することです。

dkmsツールは自動的に鍵ペアと証明書を/var/lib/dkms/mok.keyに生成します。

および/var/lib/dkms/mok.pubをビルドして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

```
# openssl x509 -in /var/lib/dkms/mok.pub -noout -text
```

を作成し、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回尋ねますが、このパスワードはプロセスの次のステップでもう1回入力する必要があります！システムを再起動すると、自動的にMOKManager EFIバイナリが起動し、mokutilを使用して登録を開始するときに選択したパスワードを使用して、キー/証明書を検証し、登録を確認することができます。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に署名するためにも使用できます。

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、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ース間のAPI呼び出しに使用されます。

Proxmox Backup Server に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/リモート構成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

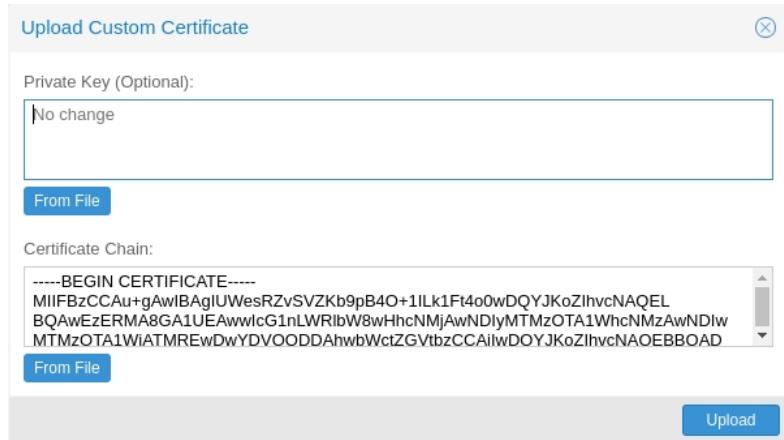
証明書には以下のオプションがあります：

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、商用の認証局（CA）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インターフェイスにも統合されています。

証明書はProxmox BackupのWebインターフェイス/APIまたは proxmox-backup-manager CLI ツール。

13.3.2 カスタム証明書のアップロード

Proxmox Backupホストに使用する証明書をすでにお持ちの場合は、Webインターフェイスから証明書をアップロードするだけです。



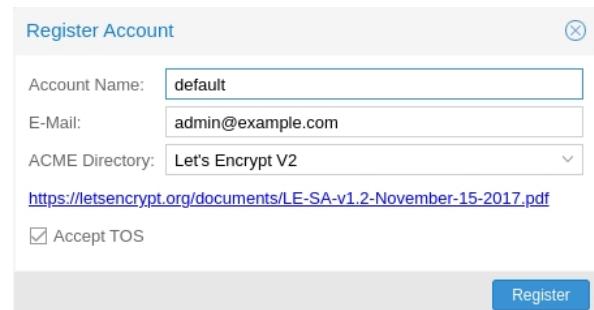
証明書キー・ファイルはパスワードで保護されてはならないことに注意してください。

13.3.3 Let's Encrypt (ACME)による信頼できる証明書

Proxmox Backupには**ACME** (Automatic Certificate Management Environment) プロトコルの実装が含まれているため、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Webサーバーを使用したhttp-01チャレンジの検証と、`acme.sh`が行うすべての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アカウント



クラスタごとに、使用するエンドポイントに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無効化は、Webインターフェース Certificates -> ACME Accounts または proxmox-backup-manager コマンドラインツールを使用して行うことができます。

```
proxmox-backup-manager acme アカウント登録< アカウント名 >< mail@example.com>
```

ヒント: レート制限のため、実験やACMEを初めて使用する場合は、LEステージングを使用し、そこですべてが動作するまで使用し、その後で本番用ディレクトリに切り替えてください。

ACMEプラグイン

ACMEプラグインの役割は、お客様、ひいてはお客様の運用下にあるProxmox Backup Serverがドメインの真の所有者であることを自動的に検証することです。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。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、これが不可能なこともあります。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

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[Certificates] -> [ACME Challenges]、または`proxmox-backup-manager acme plugin add`コマンドを使用してプルーフィングを設定できます。

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ーを追加したり、既存のドメインエントリーを管理したりするには、「証明書」の下にある`proxmox-backup-manager`コマンドを実行します。

Create: Domain	
Challenge Type:	DNS
Plugin:	pdns1
Domain:	pbs-demo.example.com

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で生成された組み込みのウェブサーバを経由してhttp-01チャレンジを検証するために、常に暗黙的に設定されたスタンドアロンプラグインがあります。

注: 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。

Let's Encrypt ACMEでの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**は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には他のリスナーがいてはいけません。
- 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可能です。この検証方法では、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.sh¹プロジェクトで開発されたDNSプラグインを再利用しています。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APIプロバイダーを選択し、API経由で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。

ヒント: プロバイダのAPI認証情報の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は自動的にクレデンシャル用のフォームを生成しますが、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。これらの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 経由で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できます。手動で _acme-challenge.domain2.example を指す _acme-domain1.example のペーマネント CNAME レコードを設定し、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。

/etc/proxmox-backup/node.cfg を domain2.example に変更して、domain2.example の DNS domain1.example のす

べてのチャレンジを検証できるようにします。

¹ acme.sh <https://github.com/acmesh-official/acme.sh>

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。つまり、*.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経由で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。ドメインごとに異なるプラグインインスタンスを指定することで、複数のプロバイダやインスタンスのDNS APIを混在させることも可能です。

ヒント 複数のドメインで同じサービスにアクセスすることは複雑さを増すので、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ている場合、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、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がトリガーされます。

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なくしたい場合は、サーバーに有効な証明書を生成する必要があります。

ssh経由でProxmoxバックアップにログインするか、コンソールを使用します：

```
openssl req -newkey rsa:2048 -nodes -keyout key.pem -out req.pem
```

画面の指示に従ってください：

```
国名 (レターコード) [AU]: AT
都道府県名 (フルネーム) [都道府県名]: ウィーン 地方名 (都市名など) []: ウィーン
組織名 (例、会社) [Internet Widgits Pty Ltd]: Proxmox GmbH 組織単位名 (例: セクション) []: Proxmox Backup
コモンネーム (例: あなたの名前) []: 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 []: 不要
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バックアップに保存します。

新しい証明書をアクティベートするには、Proxmoxバックアップで以下を実行します。

```
cp key.pem /etc/proxmox-backup/proxy.key cp cert.pem
/etc/proxmox-backup/proxy.pem
```

その後、APIサーバーを再起動します：

```
systemctl restart proxmox-backup-proxy
```


ブラウザを使用して、新しい証明書をテストします。

注: Proxmox/バックアップ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ー

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定義は'etc/proxmox-backup/metricserver.cfg'に保存され、ウェブインターフェースで編集できます。

注: 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

プラグインは、InfluxDB 2.x の HTTP(s) API を使用するように設定できます。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は認証がないと使えないで、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、トークンとして'username:password'を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 25000000バイト）を設定することもできます（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。この場合、InfluxDB サーバーが正しく設定されている必要があります。MTU も必要に応じてここで設定できます。

以下はInfluxDBの設定例です（InfluxDBサーバー上）：

```
[udp].
  enabled= true
  バインドアドレス= "0.0.0.0:8089"
  データベース= "proxmox"
  バッチサイズ= 1000
  バッチタイムアウト= "1s"
```

この設定では、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。

13.5 サービスデーモン

13.5.1 プロックスバックアッププロキシ

このデーモンは、HTTPSを使用してTCPポート8007で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。それ以上のパーミッションを必要とする操作はローカルのproxmox-backup転送されます。

13.5.2 プロックスバックアップ

このデーモンは、127.0.0.1:82のProxmox Backup Server管理APIを公開します。このデーモンはrootであり、すべての特権行う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、アクセスすることはできません。デーモンはproxmox-backup-proxyデーモンはAPIを公開します。

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、バックアップを作成または復元することができます。

13.6.2 プロックスバックアップマネージャ

このツールは、バックアップサーバー管理API全体をコマンド公開します。

13.6.3 プロックスステープ

このツールは、テープバックアップの設定と管理を行うことができます。

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 pmtx

pmx コマンドは SCSI メディア・チェンジャー・デバイス（テープ・オートローダー）を制御します。

13.6.6 ピクサー

`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にインスパイアされています。このユーティリティは

.pxar フォーマットは、*Proxmox Backup Server* 特有のニーズ、例えばハードリンクの効率的な保存に適合しています。このフォーマットは、高レベルの重複排除を実現することで、サーバー上の必要なストレージを削減するように設計されています。

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ーのアーカイブを作成します：

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

これにより、ソースフォルダの内容を含む `archive.pxar` という新しいアーカイブが作成されます。

注: `pxar` は既存のアーカイブを上書きしません。ターゲット フォルダに同名のアーカイブがすでに存在する場合は、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、デバイスの境界には従いません。この設計上の決定は、バックアップ用のアーカイブを作成するという主なユースケースに基づいています。特定の一時ファイルやシステム固有ファイルの内容をバックアップで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境界に従うようにするには、`--all-file-systems` フラグを使用します。

を渡すことで、特定のファイルやフォルダをアーカイブから除外することができます。

パラメータに `gitignore` 形式のマッチパターンを指定します。

例えば、`.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、次のように実行します：

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

シェルは `pxar` を起動する前にグロブ・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も可能です。これにより、より複雑なファイルの包含/除外動作を使用することができます。ただし、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とします。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'
```

注: グロブ・マッチ・パターンの順序は重要で、後のものが前のものを上書きします。同じパターンを許可しても結果は異なります。

`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 フォルダに既にする場合、このファイルはアーカイブに含まれず、代わりに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、グローブ マッチ パターンを `.pxareexclude` 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・ツリーのどのディレクトリにも作成、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、後のパターンが前のパターンを上書きします。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「[ノックアップの作成](#)」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、以下のコマンドでターゲット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ていない場合、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、フォルダのみをリストアするには、対応するグロブ一致パターンを追加パラメータとして渡すか、ファイルに保存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ターゲット /restore/target/etcにリストアします。一致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。

アーカイブの内容一覧

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
```

これは、アーカイブの基準として、各ファイルやディレクトリのフルパスを表示します。

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/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、指定されたマウントそのコンテンツにアクセスすることができます。

```
# cd /mnt
# ls
/bin      dev home lib32 libx32          メディアオプト      root sbin sys usr boot etc
lib      lib64 lost+found mnt           proc runsrv     tmp var
```

13.6.7 プロ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・ファイルを検査し、チャンクの整合性を検証するデバッグ機能を実装します。

diff」サブコマンドを使用すると、任意の2つ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」サブコマンドも含まれており、任意の api パスを呼び出したり (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、ツールは対応する直接呼び出します。

警告 `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。

フォーティーン

ネットワーク管理

Name	Type	Active	Autostart	VLAN	Ports/Slaves	Bond Mode	CIDR	Gateway	Comment
en01	Network Device	Yes	No	No					
en02	Network Device	No	No	No					
ens5f0	Network Device	Yes	Yes	No		192.168.16.120/20	192.168.16.1		
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。Webインターフェイスの設定オプションは、【設定】メニューのツリー項目の【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・ツールには、`network` 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、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: ネットワーク・コンフィギュレーションに加えられた変更は、コンフィギュレーションの適用をクリックするか、ネットワーク・リロード・コマンドを入力するまで適用されません。これにより、度に多くの変更を行うことができます。また、ここでミスをす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る可能性があるため、変更を適用する前に変更が正しいことを確認できます。

利用可能なインターフェースのリストを取得するには、次のコマンドを使用します:

```
# proxmox-backup-manager ネットワークリスト
```

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
----	-----	---------	----	------	--------	---------

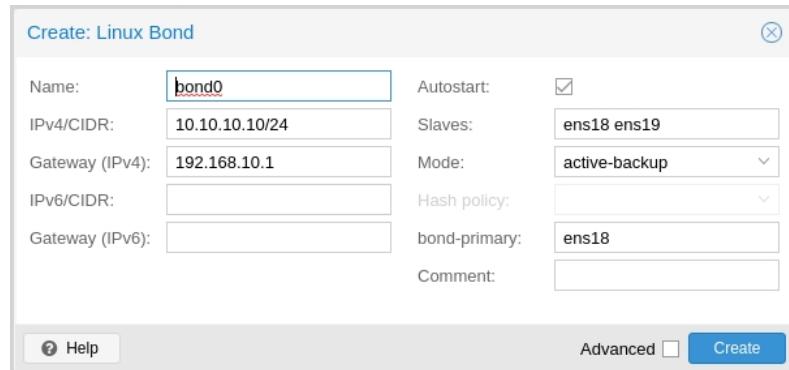
(次ページへ続く)

bond0	ポンド	1	静的	x.x.x.x/x	x.x.x.x	ens18
ens18	eth	1				
ens19	eth	1				

(前ページから続く)

新しいネットワーク・インターフェースを追加するには、`create` サブコマンドに関連するパラメータを指定します。例えば、ネットワークの冗長化のためにボンドを設定したい場合があります。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` サブコマンドを使用すると、ネットワーク・インターフェイ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・コンフィギュレーション・ファイルの保留中の変更は、ウェブ・インターフェースの下部に表示されます。コマンドを使用して、これらの変更を表示することもできます：

```
# proxmox-backup-manager ネットワークの変更
```

時点ですべての変更をキャンセルしたい場合は、【元に戻す】ボタンをクリックするか、次のコマンドを使用します：

```
# proxmox-backup-manager network revert
```

変更に満足し、設定書き込む場合は、次のように選択します。

設定を適用します。対応するコマンドは

```
# proxmox-backup-manager network reload
```

注意: 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またはバージョン7以前のProxmox VEにインストールしている場合は、自分でインストールする必要があります。

DNSの設定は、**Configuration**の**DNS**セクションから、またはdns `proxmox-backup-manager`のサブコマンドです。

14.1 交通規制

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:	<table border="1"> <tr> <th>Time Start</th> <th>Time End</th> <th>Mon</th> <th>Tue</th> <th>Wed</th> <th>Thu</th> <th>Fri</th> <th>Sat</th> <th>Sun</th> </tr> <tr> <td>07:00</td> <td>19:00</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </table>					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"/>				
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"/>																			
<input type="button" value="Add"/>																							
<input type="button" value="Help"/> <input type="button" value="OK"/> <input type="button" value="Reset"/>																							

バックアップの作成と復元は多くのトラフィックを発生させ、共有ストレージやネットワーク上の他のユーザーに影響を与える可能性があります。

Proxmox Backup Serverでは、トーカ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・コントロールは、ウェブ・インターフェイスまたはトラフィック・コントロールのコマンドを実行します。

注: サーバー上の同期ジョブは、設定されたレートイン制限の影響を受けません。プルベースの同期ジョブが生成する受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。

次のコマンドは、すべてのIPv4クライアント（ネットワーク0.0.0.0/0）を100 MB/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 \
--レートイン 100MB --レートアウト 100MB
-コメント "全クライアントのデフォルトレート制限 (100MB/s)"
```

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 を渡す必要があります。

例えば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 \
--時間枠 "月...金 8-12" \
--時間枠 "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（とサーバ自身）の設定を次の上書きします:

```
# 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。
-コメント "ローカルネットワークに20GB/sを使用"
```

注: 同じネットワークに複数のルールがある場合、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 トライック制御リスト
+-----+-----+-----+-----+
|名前| レートイン| レートアウト| ネットワーク |
+-----+-----+-----+-----+
|rule0| 100 MB| 100 MB| ["0.0.0.0/0"] |
+-----+-----+-----+-----+
|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
```

設定されているすべてのルールの状態（現在のデータレート）を表示するには、以下を使用します:

```
# proxmox-backup-manager traffic-control トライック制御
+-----+-----+
|name| cur-rate-in|
+-----+-----+
|ルール0| 0 B|
+-----+-----+
|rule1| 1.161 GiB| 19.146 |
```


お知らせ

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通知システムによって処理されます。通知イベントには、タイムスタンプ、重大度レベル、タイプ、その他のメタデータフィールドなどのメタデータがあります。
- 通知マッチャーは、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。マッチャーは、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ためのマッチルールを持つことができます。
- 通知、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。

データストアとテープバックアップジョブには、設定可能な通知モードがあります。このモードでは、通知システムと、通知メールを送信するレガシーモードを選択できます。レガシーモードは、Proxmox Backup Server 3.2以前の通知方法と同じです。

通知システムは、GUI の「Configuration Notifications」で設定できます。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などの機密性の高い設定オプションが含まれており、`root` によってのみ読み取ることができます。

15.2 通知対象

Proxmox Backup Serverには複数の通知先があります。

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 系オペレーティングシステムでよく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あり、ユーザやアプリケーションがコマンドラインやスクリプト内から直接電子メールを送信することができます。

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 → ユーザー管理で設定できます。ユーザのメールアドレスが設定されていない場合、メールは送信されません。

注意: Proxmoxバックアップサーバ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。外部メールリレー(スマートホスト)を設定するなどして、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かもしれません。配信に失敗した場合は、Postfix демон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電子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。

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、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。

注意: Gotify 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、構成可能な URL への HTTP リクエストを実行します。以下の設

定オプションがあります：

- **url:** HTTP リクエストを実行する URL。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。
- メソッドを使用します： 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、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だけが読める保護された設定ファイルに保存されます。秘密はsecrets名前空間を通してbody/header/URLテンプレートでアクセスできます。
- コメント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構成オプションでは、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：

- {タイトル }}: レンダリングされた通知のタイトル

- {{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(情報、通知、警告、エラー、不明)
- {{timestamp}}: 通知のタイムスタンプをUNIXエポック(秒単位)で指定します。

- {{ fields.<name> }}: 通知のメタデータフィールドのサブ名前空間。in-stanceの場合、fields.typeは通知タイプを含み、すべての利用可能な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 はsecrets.tokenでアクセスできます。

便宜上、以下のヘルパーをいます:

- {{ 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（例えばfields）を渡すのに便利です（例えば {{ json fields }}）。

注: 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 }}>
- ヘッダー
 - 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 }}>
- ヘッダー
 - コンテンツタイプ: application/json
- ボディ

```
{
  "content": "`` `{{ エスケープメッセージ }}` `` `"
}
```

- 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":
  "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、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。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一致させることができます。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マッチャで使用されている場合でも、各ターゲットは通知の度に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。全ての設定オプションは [notifications.cfg](#) を参照してください。

15.3.1 カレンダーのマッチングルール

カレンダーマッチャーは、通知のタイムスタンプにマッチします。例

- マッチカレンダー 8-12
- 試合カレンダー 8:00-15:30
- 試合カレンダー 月～金 9:00～17:00
- 試合カレンダー 日・火・水・金 9-17

15.3.2 フィールド・マッチング・ルール

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-.*$` 任意の データストアで始まるものにマッチします。
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 深刻度マッチングルール

通知にはマッチする重大度があります。例

- `match-severity` エラー: マッチエラーのみ
- `match-severity` 警告, エラー: マッチの警告とエラー

次の深刻度が使用されています: `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、そのタイプ、重大度、および追加のメタデータフィールドの一覧です。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	重症度	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しました た ガベージコレクションの失敗 ガベージコレクション セス パッケージ アップデート avail- 有能 ブルーンの失敗 ブルーンの仕事の成功 リモート同期の失敗 リモート同期成功 テープバックアップジョブの失敗 テープバックアップジョブの成功 テープ装填要求 検証ジョブの失敗 検証ジョブの成功	窮み ジーシー ジーシー パッケージアップデー タ スモモ スモモ 同期 同期 テープバックアップ テープバックアップ テープロード ベリフィケーション ベリフィケーション	エラー エラー インフォメ ーション インフォメ ーション エラー エラー エラー エラー エラー エラー エラー お知らせ	ホスト名 データストア、ホスト名 データストア、ホスト名 ーション ホスト名 ーション データストア、ホスト名、ジョブID データストア、ホスト名、ジョブID ーション データストア、ホスト名、ジョブID データストア、ホスト名、ジョブID ーション データストア、ホスト名、メディアプール、 ジョブID データストア、ホスト名、メディアプール、 ジョブID ホスト名 データストア、ホスト名、ジョブID データストア、ホスト名、ジョブID ーション

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 `match-field` マッチルール。

メタデータフィー	説明
ルド	
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ーのホスト名
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

注意: 利用可能なシステムアップデートをチェックするディリータスクは、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・通知メールを送信します。Proxmox Backup Serverは、これらのメールを通知システムにタイプsystem-mail、深刻度unknownの通知として送信します。

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、2つの値のいずれかを指定できます：

- `legacy-sendmail`: システムの `sendmail` コマンドで通知メールを送信します。通知システムはバイパスされ、設定されたターゲット/マッチャーは無視されます。このモードはProxmox Backup Server 3.2より前のバージョンの通知動作と同じです。
- 通知システム: 新しいフレキシブルな通知システムをご利用ください。

notification-modeオプションが設定されていない場合、Proxmox Backup Serverのデフォルトは次のようにになります。`legacy-sendmail`.

Proxmox Backup Server 3.2以降、UIで作成されたデータストアは自動的に新しい通知オプトインします。データストアがAPIまたは`proxmox-backup-manager` CLI経由で作成された場合、通知システムを使用する場合はnotification-modeオプションをnotification-systemに明示的に設定する必要があります。

レガシー送信モードはProxmox Backup 後のリリースで削除される可能性があります。

15.7.1 レガシーメール通知モードの設定

`notification-legacy-sendmail`に設定されている場合、Proxmox Backup Serverはシステムの`sendmail`コマンドを使用して、`notify-user`オプションで設定されたユーザーの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）。

データストアでは、タスクの種類ごとに、`notify` オプションがあります。

- 常に: 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。
- エラー: エラーが発生したスケジュールタスクに通知を送信します。
- Never: 通知を送信しません。

`notify-mode` が次のように設定されている場合、`notify-user` と `notify` オプションは無視されます。通知システム。

15.8 通知テンプレートのオーバーライド

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/`のオーバーライドディレクトリにカスタムテンプレートファイルを提供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はまず`override`ディレクトリからテンプレートのロードを試みます。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-err-body.txt.hbs` ファイル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み、`package-updates-subject.txt.hbs` 利用可能なパッケージアップデート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、および動的インデックスと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：

<データ・ストア・ルート>/<タイプ>/<id>/<タイム>/。

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 スナップ写真

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、blob（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、識別用のチェックサムを付けた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前回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できます。サーバに既に存在するチャンクを検出すると、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かった場合、サーバは未完成のスナップショットを削除します。

16.3 チャンクス

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。チャンクはその内容の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

<データストアルート>/chunks/.

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 lives in  
<データストアルート>/chunks/a342/.
```

これは、ディレクトリごとのファイル数を減らすために行われます。ディレクトリごとに多くのファイルがあると、ファイルシステムのパフォーマンスが低下する可能性があるからです。

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（通常は4MiB）に分割されます。

ゲスト上のファイルシステムは、ほとんどの場合、連続した断片でファイルを割り当てようとするため、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。そのため、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QEMU内部のスナップショット機構を使用しており、ストレージのスナップショットにも依存していません。

16.3.2 ダイナミックにサイズ調整されたチャンク

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のではなく、動的（ダイナミック）な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の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。

これを改善するために、[Proxmox Backup Server](#)は代わりに動的（ダイナミック）な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の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、データを反復処理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 暗号化されたチャンク

暗号化された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）チャンクの内容ではなくプレーンテキストの内容に暗号化キーを連結したもので計算されます。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。

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です。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、ディレクトリエントリを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。そのため、このモードでは、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。

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 旧バージョンとの後方互換性があります。ただし、Webインターフェイ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、ディレクトリエントリ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ライが読み込まれ、pxarアーカイブ形式を使用してエンコードしてシリアル化されます。結果のストリームは動的にチャンク化され、Proxmox Backup 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に読み込まれ、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、ファイルの内容は無条件に読み込まれ、チャンクされます。

しかし、エントリのメタデータとデータを1つのpxarアーカイブに格納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 フォーマットのアーカイブと、一貫性チェックのためのペイロードが含まれます。

ードヘッダによって分離された、実際のファイルコンテンツを含む 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。メタデータアーカイブには、対応するペイロードアーカイブエントリへの参照オフセットが格納されるため、ファイルコンテンツにアクセスすることができます。どちらも

これらのアーカイブはチャンク化され、Proxmoxバックアップクライアントによって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するメタデータを効率的に取得するために使用できます。これは例えば、アーカイブの内容をリストアップするためのエントリの検索や、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に使用されます。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ことで、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とみなされる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されるため、ファイルサイズと`mtime`は変更されませんでしたが、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情報、`acls`、属性、そして最も重要なファイルの`mtime`が含まれます。一部のツール（`vzdump`など）では、Proxmoxバックアップクライアントで実際にバックアップを実行する前に、ファイルシステムの内容を一時的な場所に同期するため、ファイルの`ctime`とinode番号は保存されず、比較に使用されません。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変更されていないファイルのファイル・コンテンツ・チャンクを再利用することで、可能な限りファイル・コンテンツの読み取り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、バックアップ実行の開始時に以前のスナップショット`mpxar`メタデータアーカイブがダウンロードされ、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。

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い場合、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ルックアヘッド・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。ファイルの境界はチャンクの境界と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消えてしまい、範囲に穴が空いたファイルがあります。この時点でキャッシュ無効になり、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上限以下）の場合、コンテンツ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にエンコードし、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、ファイルが再利用されます。しかし、パディングが制限を超えて許容できない場合は、すべてのキャッシュされたエントリが再エンコードされ、既存のデータは再利用されません。されたファイルの内容が再利用されるか再エンコードされるかにかかわらず、キャッシュされたメタデータは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、コンテンツが変更されていないファイルに対して、可能な限りファイル・コンテンツの再読み込みと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に、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に、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境界の提案として、遭遇したファイル境界を通知します。チャンカーは内部状態に基づいて、提案された境界を受け入れるか無視するかを決定します。

16.4 注意点と限界

16.4.1 ハッシュ衝突に関する注意事項

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（またはそれ以上）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「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：

$$p(n, d) = 1 - e^{-n^2/(2d)}$$

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 = 2²⁵⁶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、このシナリオにおける衝突のは次のようになります：

$$3.1115 \times 10^{-61}$$

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、衝突の確率は、このような宝くじゲームを8回連続で当て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、通常のデータストアで偶然にこのような衝突が起こ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、チャンクの大きさ上限があるため、潜在的な攻撃者は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ので、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、アップロードされるのは新規または変更されたチャンクのみです。

このような制限を避けるために、Change Detection Mode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、オリジナル（平文）データのチェックサムのみが利用可能であるため、（暗号鍵を持たない）サーバーが内容を照合することは不可能です。その代わりに 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、どのチャンクが参照されているかを調べ、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は有用でしょう。
`proxmox-backup-debug`コマンドラインツールはそのようなファイルを検査し、内容を回復するために使用できます。
 例えば、.fidxインデックスの参照されているチャンクのリストを取得するには、以下のようにします：

```
# proxmox-backup-debug inspect file drive-scsi0.img.fidx
```

同じコマンドを使用して、.blobファイルを検査できます。-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）。

次の例は、qemu-server.conf.blobのデコードされた内容を表示します。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は、--keyfileを使用して鍵ファイルへのパスを指定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参照されているかを確認することもできます：

```
# proxmox-backup-debug inspect chunk
  ↗b531d3ffc9bd7c65748a61198c060678326a431db7edeb874c327b7986e595e0 --reference-filter /path/
  ↗in a データストアディレクトリ
```

ここで--reference-filterは、インデックス・ファイルを検索する場所を指定します。これはアービトラリパスでもかまいません。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用されます。もし

--reference-filterを指定すると、チャンクのCRCと暗号化状態のみを表示します。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、デコードするには--keyfileを指定する必要があります。

16.5.1 Proxmox Backup Serverが稼働していない状態でのリストア

Proxmox Backup Serverインスタンスを実行せずに、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することができます。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。どちらのパスも絶対パスでも相対パスでもかまいません。-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。最初にskip-CRCオプションなしで試すことをお勧めします。

よくあるご質問

17.1 Proxmox Backup Server (PBS)はどのディストリビューションをいますか?

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ることはありません。

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/バックアップバージョン	Debian バージョン	第1回 再リース	Debian EOL	プロックスモックス バックアップEOL
Proxmox/バックアップ3	Debian 12 (本の虫)	2023-06	未定	未定
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を次の再リースポイントにアップグレードする方法を教えてください。

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または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、リリースノートをチェックして、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 *Node -> Updates* パネルを使用するか、CLI を使用します：

```
アプトアップデート  
apt full-upgrade
```

注意: 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サーバにアップグレードする方法は?

Proxmox Backup Server 2.4から3.1へのメジャーバージョンアップも可能です。このようなアップグレードは慎重に計画し、テストする必要があります。また、リモート同期やテープなど、重要なバックアップのオフサイトコピーを準備せずにアップグレードを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、ここではアップグレードの一般的な手順とアドバイスを提供します:

- 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取り込む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?

Proxmox Backup Serverは、トランSPORTレイヤーセキュリティ(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。

注: 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](#)

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の両方が最小限に抑えられます。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。

コマンドの構文

注意: コマンドラインツールのログの冗長性は、PBS_LOG (pxarの場合: 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].
```

ホスト) バックアップを作成します。

</バックアップ仕様

[<string>] バックアップ・ソース仕様のリスト:

[<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。

--バックアップID <文字列>

バックアップID

--バックアップ時間 <整数> (1 - N)

バックアップ時間 (Unixエポック)

--バックアップタイプ vm|ct|host

バックアップタイプ。

--バースト <文字列>

トークン・バケツ (トークン・バケツ・フィルタ用) のサイズをバイト単位で表し、単位は任意 (B、KB (基数10)、MB、GB、...、KiB (基数2)、MiB、GiB、...)。

--change-detection-mode legacy|data|metadata (default=legacy)

前回のバックアップ実行以降のファイル変更を検出するモード

--chunk-size <integer> (64 - 4096) (default=4096)

KB単位のチャンクサイズ。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、またはどちらも行わないかを定義します。

--ドライラン <ブール値> (デフォルト=false)

バックアップの示すだけで、何もアップロードしないでください。

--entries-max <整数> (デフォルト=1048576)

メモリに保持するエントリの最大数。

--除外 <文字列>

除外するファイルにマッチするパスまたはパターンのリスト。複数回指定できます。

--include-dev <文字列>

指定されたファイルと同じ 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 <整数> (0 - N)

すで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。

--キーファイル <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。

-マスター・パブキー--fd <整数> (0 - N)

既に開い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。

--マスター・パブキーファイル <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。

--ns <文字列>

名前空間。

--レート <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 (ベース 10)、MB、GB、...、KiB (ベース 2)、MiB、GiB、...)。

--リポジトリ <文字列>

リポジトリのURL。

--skip-e2big-xattr <boolean> (default=false)

xattrs を取得する際に E2BIG エラーを無視します。これはファイルを含みますが、メタデータは破棄します。

--skip-lost-and-found <boolean> (default=false)

lost+found ディレクトリをスキップします。

proxmox-backup-client benchmark [OPTIONS] (プロックスモックスバックアップクライアントベンチマーク)

ベンチマークテストの実

行 オプションのパラメー

タ:

--キーファイル <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。

--output-format 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> [OPTIONS].

ダンプのカタログ

<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:

--keyfd <整数> (0 - N)

すで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。

--キーファイル <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。

<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

[<string>] バックアップアーカイブ名。

オプションのパラメータ:

--keyfd <整数> (0 - N)

すで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。

--キーファイル <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 change-owner <group> <new-owner> [OPTIONS].
```

バックアップグループの所有者の変更

<グループ

[<string>] バックアップグループです。

<新オーナー

[<string>] 認証ID オプション・パ

ラメーター:

--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] (プロックスモックスバックアップクライアントガベージコレクト)
```

特定のリポジトリの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:

--output-format 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 group forget <group> [オプション].
```

バックアップスナップショットを忘れる（削除する）。

<グループ

[<string>] バックアップグループ オ

ーションのパラメーターです：

--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 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 <ブール値

冗長なヘルプ。

```
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。

</パス

[<string>] キーファイル。これを指定しないと、デフォルト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：

--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導出機能。

```
proxmox-backup-client key create [<path>] [OPTIONS].
```

新しい暗号化キーを作成します。

</パス

[<string>] 出力ファイル。これを指定しないと、キーが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：

--hint <文字列

パスワードのヒント

パスワードで保護された暗号鍵のための鍵導出機能。

```
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ー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・ユーザーによる今後の呼び出しのデフォルト・マスター鍵として使用されます。

</パス

[<string>] PEM形式のRSA公開鍵へのパス。

```
proxmox-backup-client key import-with-master-key [<path>].  
--encrypted-keyfile <string> --master-keyfile <string> [OPTIONS].
```

(プライベート)マスターキーを使用して暗号化キーの暗号化バックアップをインポートします。

</パス

[<string>] 出力ファイル。これを指定しないと、キーが新しいデフォルトの暗号化キーになります。

--暗号化キーファイル <文字列

インポートする RSA 暗号化鍵ファイル。

--マスターキーファイル <文字列

(プライベート) 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導出機能。

```
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。このファイルには高速にキーを復元するための
スキャナ可能なQRコードも含まれています。

</パス

[<string>] キーファイル。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:

--output-format text|html

Paperkey 出力フォーマット

--件名 <文字列

指定され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。

</パス

[<string>] キーファイル。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:

--output-format text|json|json-pretty
出力フォーマット

`proxmox-backup-client key show-master-pubkey [<path>] [OPTIONS].`

マスターキーに関する情報一覧

</パス

[<string>] PEMフォーマットされた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:

--output-format text|json|json-pretty
出力フォーマット

`proxmox-backup-client list [OPTIONS]` (プロックスモックスバックアップクライアントリスト)

バックアップグループを一

覧表示します。

オプションのパラメータ

:

--ns <文字列

名前空間。

--output-format 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 login [OPTIONS]` (プロックスモックスバックアップクライアントログイン)

ログインを試みます。成功した場合、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

リポジトリのURL。

`proxmox-backup-client logout [OPTIONS] [オプション].`

ログアウト(保存されたチケットの削除)。

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 map <スナップショット> <アーカイブ名> \[オプション\].](#)

A.1. proxmox-backup-client

VMバックアップからドライブイメージをローカルのループバックデバイスにマッピングします。元に戻すには'unmap'を使用します。警告：これは信頼できるバックアップでのみ行ってください！

<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

[<string>] バックアップアーカイブ名。

オプションのパラメータ:

--キー <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/スナップショットパス。

<アーカイブ名>

[<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。

オプションのパラメータ:

--キー <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 名前空間作成 [<ns>] [OPTIONS].

新しいネームスペースを作成します。

<ns>

[<string>] 名前空間。

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 名前空間の削除 [<ns>] [OPTIONS].

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:

--グループ削除 <ブール値

階層内のすべてのグループを破棄します。

--リポジトリ <文字列

リポジトリのURL。

```
proxmox-backup-client 名前空間リスト [<ns>] [OPTIONS].
```

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:

--最大深度 <整数> (0 - N)

最大再帰深度

--output-format 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 prune <group> [OPTIONS].
```

バックアップリポジトリを削除します。

<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ーです:

-ドライラン <ブール値>

pruneがするかを示すだけで、何も削除しないでください。

--keep-daily <整数> (1 - N)

毎日保持するバックアップの数。

--1時間ごと <整数> (1 - N)

1時間ごとに保存するバックアップの数。

--keep-last <整数> (1 - N)

保持するバックアップの数。

--keep-monthly <整数> (1 - N)

毎月保存するバックアップの数。

--キープウィークリー <整数> (1 - N)

週ごとのバックアップ。

--キープ・イヤー・リー <整数> (1 - N)

年間バックアップ数

--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します)

--ns <文字列>

名前空間。

--output-format text|json|json-pretty

出力フォーマット

--quiet <boolean> (default=false)

最小限の出力 - 削除だけを表示します。

--リポジトリ <文字列>

リポジトリのURL。

```
proxmox-backup-client restore <スナップショット> <アーカイブ名> <ターゲット> [OPTIONS].
```

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

[<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には'.'を使用します。標準

出力に書き出す場合、「.pxar」アーカイブは展開しません。

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する場合は失敗しません。

--バースト <文字列

トークン・バケツ（トークン・バケツ・フィルタ用）のサイズをバイト単位で表し、単位は任意（B、KB（基数10）、MB、GB、...、KiB（基数2）、MiB、GiB、...）。

--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、またはどちらも行わないかを定義します。

--ignore-acls <ブール値> (default=false)

acl 設定を無視

--ignore-extract-device-errors <boolean> (default=false)

デバイス・ノード抽出中に発生したエラーを無視

--ignore-ownership <boolean> (default=false)

所有者設定を無視（chownなし）

--ignore-permissions <boolean> (default=false)

パーミッション設定を無視（chmodなし）

--ignore-xattrs <boolean> (default=false)

xattrの設定を無視

--keyfd <整数> (0 - N)

すで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。

--キーファイル <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。

--ns <文字列

名前空間。

--overwrite <boolean> (default=false)

既存のファイルを上書き

--overwrite-files <boolean> (default=false)

既存のファイルを上書き

--overwrite-hardlinks <boolean> (default=false)

アーカイブズハードリンクによる既存のエントリーの上書き

--overwrite-symlinks <boolean> (default=false)

既に存在するエントリをシンボリックリンクで上書きします。

--パターン <文字列>

リストアするファイルを制限するパスまたはマッチパターン。複数回指定できます。

-プレリュード・ターゲット <文字列>

prelude の復元先パス (pxar v2 アーカイブのみ)。

--レート <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 (ベース 10)、MB、GB、...、KiB (ベース 2)、MiB、GiB、...)。

--リポジトリ <文字列

リポジトリのURL。

```
proxmox-backup-client snapshot <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

[<string>] スナップショットのパス。

オプションのパラメータ:

--ns <文字列

名前空間。

--output-format 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 snapshot forget <snapshot> [OPTIONS].
```

バックアップスナップショットを忘れる (削除する)。

<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:

--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 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。

<グループ

[<string>] バックアップグループです。

オプションのパラメータ:

--ns <文字列

名前空間。

--output-format 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 snapshot notes show <snapshot> [オプション].
```

ショーノート

<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:

--ns <文字列

名前空間。

--output-format 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 snapshot notes update <snapshot> <notes> [オプション].
```

更新ノート

<スナップショット

[<string>] スナップショットのパス。

<注釈

[<string>] メモ。

オプションのパラメータ:

--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 snapshot protected show <snapshot> [オプション].
```

指定したスナップショットの保護ステータスを表示

<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:

--ns <文字列

名前空間。

--output-format 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 snapshot protected update <snapshot> <protected> [OPTIONS].
```

スナップショットの保護ステータスの更新

<スナップショット

[<string>] スナップショットのパス。

<プロジェクト

[<boolean>] 保護ステータス。

オプションのパラメータ:

--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 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。

<スナップショット>

[<string>] グループ/スナップショットパス。

<ログファイル>

[<string>] アップロードするログファイルのパス。

オプションのパラメータ:

--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、またはどちらも行わないかを定義します。

--keyfd <整数> (0 - N)

すで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。

--キーファイル <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。

--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 status [OPTIONS] (プロックスモックスバックアップクライアントステータス)
```

リポジトリのステータスを

取得します。

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

--リポジトリ <文字列>

リポジトリのURL。

```
proxmox-backup-clientタスクリスト [OPTIONS].
```

このリポジトリの実行中のサーバータスクを一覧表示しま

す。オプションのパラメータ:

--all <ブール値>

また、停止中のタスクもリストアップします。

--リミット <整数> (1 - 1000) (デフォルト=50)

リストアップするタスクの最大数。

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 タスクログ <upid> [OPTIONS].

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> [オプション].`

特定のタスクを停止してみてくだ
さい。

<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 unmap [<名前>].`

map' でマッピングされたループ・デバイスのマッピングを解除し、すべてのリソースを解放します。

<名前

[<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
ッピングをすべてリストアップし、残っているインスタンスを強制的にクリーンアップします。

`proxmox-backup-client` のバージョン [OPTIONS].

クライアントとオプションのサーバーバージョ

ンを表示 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

--リポジトリ <文字列

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型リストアシェルで使用できます:

```
proxmox-backup-client shell <スナップショット> <名前.pxar
```

cd [<path>]

現在の作業ディレクトリを新しいディレクトリに変更

<パス

[<string>] ターゲットパス。

選択済みクリア

復元用に選択したファイルのリストを消去します。

選択解除 <パス

復元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。

<パス

[<string>] リストから削除するエントリへのパス。

出口

シェルの終了

find <pattern> [OPTIONS] (パターン検索)

与えられたマッチパターンにマッチするカタログ内のエントリを検索します。

<パターン

[<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>}] [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。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 <ブール値>

冗長なヘルプ。

list-selected[OPTIONS]を選択します。

リストア用に現在選択されているエントリをリストアしま

す。オプションのパラメータ:

--パターン <ブール値> (デフォルト=偽)

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。

<パス

[<string>] ターゲットパス。

プード

現在の作業ディレクトリを一覧表示します。

`restore <ターゲット> [オプション].`

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。パターンが存在しない場合、または空の場合は、アーカイブ全体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。

選択された<ターゲット>を復元

選択されたエントリを指定されたターゲットリストアします。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。

<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。

セレクト <パス>

復元するエントリを選択します。

エントリがすでに存在する場合や、無効なパスが指定された場合はエラーを返します。

<パス

[<string>] ターゲットパス。

stat <パス>

指定されたディレクトリエントリのメタデータを読み込みます。

pxarアーカイブからデータを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。

<パス

[<string>] ターゲットパス。

A.2 プロックスバックアップマネージャ

`proxmox-backup-manager acl list [OPTIONS]` (プロックスモックスバックアップマネージャクリルリスト)

アクセス制御リスト。

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 acl update <パス> <ロール> [オプション].`

アクセス制御リスト (ACL) の更新。

</パス

[<string>] アクセス制御パス。

<役割

[<role>] [PRIVILEGES]の組み合わせによってロールを表す列挙。

特権はビットフラグとして実装されているため、特権のユニークな組み合わせは、この列挙定義で使用される単一のユニークな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

--削除 <ブール値>

権限を削除します（追加代わりに）。

-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使用できます。

--グループ <文字列>

グループID

-propagate <boolean> (default=true)

パーミッションの伝搬（継承）を許可します。

`proxmox-backup-manager acme account deactivate <name> [OPTIONS] [オプション].`

ACMEアカウントを停止します。

<名前>

[<string>] ACMEアカウント名。

オプションのパラメータ:

--force <boolean> (デフォルト=false)

サーバー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 アカウント情報 <名前> [オプション].`

アクメのアカウント情報を表示します。

<名前>

[<string>] ACMEアカウント名。

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のacmeアカウントリスト [OPTIONS].`

アクメの口座一覧

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 account register <name> <contact> [OPTIONS] [オプション].
```

ACMEアカウントを登録します。

<名前

[<string>] ACMEアカウント名。

<連絡先

[<string>] 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。

<名前>

[<string>] ACMEアカウント名。

オプションのパラメータ:

--連絡先 <文字列>

メールアドレスのリスト

```
proxmox-backup-manager acme cert order [OPTIONS] (プロックスモックスバックアップマネージャクメ証明書注文)
```

新しいACME証明書を注文します。オプシ

ョンのパラメータ

--force <boolean> (デフォルト=false)

証明書の有効期限が迫っていない場合でも、強制的に更新します。

```
proxmox-backup-manager acme cert revoke
```

新しいACME証明書をご注文ください。

```
proxmox-backup-manager acme plugin add <type> <id> --api <string> --data
<文字列> [OPTIONS]
```

アクメのアカウント情報を表示します。

<タイプ>

[<string>] ACMEチャレンジプラグインタイプ。

<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。

--データ <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:

-無効 <ブール値> (デフォルト=偽)

設定を無効にするフラグ。

--validation-delay <integer> (0 - 172800) (default=30)

検証を要求する前に待つ、秒単位の追加遅延。DNS レコードの TTL が長い場合

に対応できるようにします。

```
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。

<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 プラグイン一覧 [OPTIONS] [オプション]

acmeプラグインを一覧表示
します。

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 プラグイン remove <id>

ACME プラグイン設定を削除します。

<id>
[<string>] ACMEチャレンジプラグインID。

proxmox-backup-manager acme plugin set <id> [OPTIONS].

ACMEプラグインの設定を更新します。

<id>
[<string>] ACMEチャレンジプラグインID。オプショ

ンのパラメータです:

--api <文字列

DNS API プラグイン ID。

--データ <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。指定可能。

-ダイジェスト <文字列

同時更新を防ぐダイジェスト

-無効 <ブール値> (デフォルト=偽)

設定を無効にするフラグ。

--validation-delay <integer> (0 - 172800) (default=30)

検証を要求する前に待つ、秒単位の追加遅延。DNS レコードの TTL が長い場合

に対応できるようにします。

[proxmox-backup-manager ad create <realm> --server1 <string> \[OPTIONS\].](#)

新しいADレルムの作成

<領域

[<string>] レルム名。

--サーバ1 <文字列

ADサーバーのアドレ

ス オプションのパラメータ

一:

--ベース-dn <文字列>
LDAPドメイン

--bind-dn <文字列>
LDAPドメイン

--キヤパス <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> (デフォルト=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。

--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

--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サーバーポート

--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 属性がどの属性にマッピングされるかを指定する key=value ペアのカンマ区切りリスト。
PBS のユーザーフィールドです。 例えば、LDAP 属性の mail を PBS の email にマップするには、次のように書きます。
email=mail.

-sync-defaults-options[[enable-new=<1|0>][,remove-vanished=<string>]]。
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 が person,user に設定されている場合、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)
サーバー証明書の検証

```
proxmox-backup-manager ad delete <realm> [オプション].
```

LDAPレルム設定の削除

<領域>

[<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に変更を防止します。これは同時変更を防ぐ使用できます。

proxmox-backup-managerの廣告リスト [OPTIONS]。

設定された AD レルムのリスト 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 ad show <realm> [オプション].

ADレルムの設定を表示

<領域

[<string>] レルム名。

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 ad sync <realm> [オプション].

指定されたLDAPレルムの同期

<領域

[<string>] 認証ドメインID オプションのパラメー

ター:

--ドライラン <ブール値> (デフォルト=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ーをすぐに有効化

--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えたときに削除するもののリストです。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。

proxmox-backup-manager ad update <realm> [オプション].

ADレルム設定の更新

<領域

[<string>] レルム名。

オプションのパラメータ:

--ベース-dn <文字列>

LDAPドメイン

--bind-dn <文字列>

LDAP ドメイン

--キヤパス <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> (デフォルト=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。

--削除

server2|port|comment|default|verify|mode|bind-dn|password|filter|sync-defaults-options|。

削除するプロパティのリスト。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ぐ使用できます。

--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

--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サーバーポート

--サーバ1 <文字列>

ADサーバーアドレス

--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 属性がどの属性にマッピングされるかを指定する key=value ペアのカンマ区切りリスト。

PBS のユーザーフィールドです。 例えば、LDAP 属性の mail を PBS の email にマップするには、次のように書きます。
email=mail.

-sync-defaults-options[[enable-new=<1|0>][,remove-vanished=<string>]]。

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 が person,user に設定されている場合、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)

サーバー証明書の検証

proxmox-backup-managerの証明書情報

ノード証明書情報を表示します。

proxmox-backup-managerの証明書の更新[OPTIONS]。

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

A.2. proxmox-backup-manager

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` データストア作成 <名前> </パス> [オプション].

新しいデータストア設定を作成します。

<名前

[<string>] データストア名。

</パス

[<string>] データストア・ディレクトリへの絶対パスか、リムーバブル・データストアの場合はデバイス上の相対パスです。

オプションのパラメータ:

--バックキングデバイス <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>.

指定され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。

--1時間ごと <整数> (1 - N)

1時間ごとに保存するバックアップの数。

--keep-last <整数> (1 - N)

保持するバックアップの数。

--keep-monthly <整数> (1 - N)

毎月保存するバックアップの数。

--キープウィークリー <整数> (1 - N)

週ごとのバックアップ。

--キープ・イヤー・リー <整数> (1 - N)

年間バックアップ数

--maintenance-mode [<type=> <message=>] [<タイプ=> [<メッセージ=>]]

メンテナンスモード、タイプは「オフライン」または「読み取り専用」、メッセージは "で囲んでください。

--notification-mode legacy-sendmail|notification-system
(default=legacy-sendmail)

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` 実行ファイルを経由して `notify-user` で設定されたユーザーに通知を送ります。

通知システムは、一致する通知イベントを通知発行します。

--notify [[<gc=>] [<prune=>] [<sync=>] [<verify=>]]

データストア通知設定。enum には 'always'、'never'、'error' のいずれかを指定します。

--通知ユーザー <文字列>

ユーザーID

--output-format text|json|json-pretty

出力フォーマット

--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、すべての新しいバックアップが完了直後に検証されます。

`proxmox-backup-manager` データストア一覧 [OPTIONS] [OPTIONS] を参照してください。

データストアのリスト。

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` datastore mount <store> [OPTIONS].

リムーバブルデータストアをマウントします。

<ストア>

[<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使用できます。

`proxmox-backup-manager` データストア削除 <名前> [オプション].

データストア構成を削除します。

<名前>

[<string>] データストア名。

オプションのパラメータ:

--destroy-data <boolean> (default=false)

データストアの基礎となるコンテンツの削除

-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構成が保持されます。

`proxmox-backup-manager` データストア表示 <名前> [オプション].

データストアの構成を表示

<名前>

[<string>] データストア名。

オプションのパラメータ:

--output-format text|json|json-pretty

```
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。

<ストア>

[<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使用できます。

`proxmox-backup-manager データストア更新 <名前> [オプション].`

データストア構成を更新します。

<名前>

[<string>] データストア名。

オプションのパラメータ:

--comment <文字列>

コメント

--削除

コメント|gcスケジュール|剪定スケジュール|keep-last|keep-hourly|keep-daily|keep-weekly|keep-monthly
削除するプロパティのリスト。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に変更を防止します。これは同時変更を防ぐ使用できます。

--gc-schedule <カレンダーイベント>.

指定され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。

--1時間ごと <整数> (1 - N)

1時間ごとに保存するバックアップの数。

--keep-last <整数> (1 - N)

保持するバックアップの数。

--keep-monthly <整数> (1 - N)

毎月保存するバックアップの数。

--キープウィークリー <整数> (1 - N)

週ごとのバックアップ。

--キープ・イヤー・リー <整数> (1 - N)

年間バックアップ数

--maintenance-mode [<type=> [<message=>]] [タイプ=[メッセージ=>]].

メンテナンスマード、タイプは「オフライン」または「読み取り専用」、メッセージは "で囲んでください。

**--notification-mode legacy-sendmail|notification-system
(default=legacy-sendmail)**

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。

`notify-user` で設定されたユーザーに、システムの`sendmail`実行ファイルを介して通知を送ります。

通知システムは、一致する通知イベントを通知発行します。

--notify [[gc=<enum>]] [,prune=<enum>] [,sync=<enum>] [,verify=<enum>]

データストア通知設定。enum には 'always'、'never'、'error' のいずれかを指定します。

--通知ユーザー <文字列>

ユーザーID

--output-format text|json|json-pretty

出力フォーマット

--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、すべての新しいバックアップが完了直後に検証されます。

proxmox-backup-manager データストア uuid-mount <uuid> [OPTIONS].

UUIDを指定してリムーバブル・データストアをマウントしてみてください。

<uuid>

[<string>] マウントするデバイスの UUID オプション・パラメーター:

--output-format text|json|json-pretty

出力フォーマット

proxmox-backup-manager disk fs create <name> --disk <string> [OPTIONS].

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/<名前>*の下にマウントされます。

<名前>

[<string>] データストア名。

--ディスク <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 <ブール値>

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。

proxmox-backup-manager disk fs delete <name>

*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。

<名前>

[<string>] データストア名。

proxmox-backup-manager disk fs list [OPTIONS]を指定します。

systemd データストアのマウントユニットを一

覧表示します。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 disk initialize <disk> [OPTIONS].

<ディスク

[<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。

`proxmox-backup-manager`ディスクリスト [オプション].

ローカルディスクのリスト

。

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 disk smart-attributes <disk> [OPTIONS].`

SMART属性を表示します。

<ディスク

[<string>] ブロックデバイス名 (/sys/block/<name>).

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` ディスクワイプ <ディスク

ワイプディスク

<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 single|mirror|raid10|raidz|raidz2|raidz3 [OPTIONS].`

zfsプールの作成

<名前

[<string>] データストア名。

--devices [<string>, ...] (デバイス [<文字列>, ...])

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:

--add-datastore <ブール値>

zpoolを使用してデータストアを構成します。

--ashift <整数> (9 - 16) (デフォルト=12)

プール・セクター・サイズの指数。

--圧縮 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 disk zpool list [OPTIONS].
```

ローカル zfs プール。

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 dns get [OPTIONS].
```

DNS設定の読み込み 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 dns set [OPTIONS] [オプション].
```

DNS設定を更新します。

オプションのパラメータ

:

--delete dns1|dns2|dns3

削除するプロパティのリスト。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ぐ使用できます。

--dns1 <文字列

最初のネームサーバーのIPアドレス。

--dns2 <文字列

セカンドネームサーバーのIPアドレス。

--dns3 <文字列

3番目のネームサーバーのIPアドレス。

--検索 <文字列

ホスト名のためにドメインを検索します。

```
proxmox-backup-manager garbage-collection list [OPTIONS] (プロックスモックスバックアップマネージャガベージコレクションリスト)
```

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

トします。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 garbage-collection start <store> [OPTIONS].`

特定のデータストアのガベージ・コレクションを開始します。

<ストア

[<string>] データストア名。

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 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。

<ストア

[<string>] データストア名。

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 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。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 <ブール値
冗長なヘルプ。

proxmox-backup-manager ldap create <realm> --base-dn <string> --server1 <string> --user-attr <string> [OPTIONS].

新しいLDAPレルムの作成

<領域

[<string>] レルム名。

--ベース-dn <文字列

LDAPドメイン

--サーバ1 <文字列

LDAPサーバーアドレス

--user-attr <文字列

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。

オプションのパラメータ:

--bind-dn <文字列
LDAPドメイン

--キヤパス <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> (デフォルト=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。

--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

```
--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

--server2 <文字列>
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 属性がどの属性にマッピングされるかを指定する key=value ペアのカンマ区切りリスト。
  PBS のユーザーフィールドです。 例えば、LDAP 属性の mail を PBS の email にマップするには、次のように書きます。
  email=mail.

-sync-defaults-options[[enable-new=<1|0>][,remove-vanished=<string>]]。
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 が person,user に設定されている場合、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)
サーバー証明書の検証
```

`proxmox-backup-manager ldap delete <realm> [OPTIONS].`

LDAPレルム設定の削除

<領域>

[<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に変更を防止します。これは同時変更を防ぐ使用できます。

`proxmox-backup-manager ldap list [OPTIONS]`を実行します。

設定されている LDAP レルム一覧 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 ldap show <realm> [OPTIONS].`

LDAPレルム設定の表示

<領域>

[<string>] レルム名。

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 ldap sync <realm> [OPTIONS].
```

指定された LDAP レルムの同期

<領域

[<string>] 認証ドメインID オプションのパラメー

ター:

--ドライラン <ブール値> (デフォルト=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ーをすぐに有効化

--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えたときに削除するもののリストです。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。

```
proxmox-backup-manager ldap update <realm> [OPTIONS].
```

LDAP レルム設定の更新

<領域

[<string>] レルム名。

オプションのパラメータ:

--ベース-dn <文字列>

LDAP ドメイン

--bind-dn <文字列>

LDAP ドメイン

--キヤパス <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既定の証明書ストア /etc/ssl/certs に置き換わります。

--コメント <文字列>

コメント

--default <boolean> (デフォルト=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。

--削除

server2|port|comment|default|verify|mode|bind-dn|password|filter|sync-defaults-options|。

削除するプロパティのリスト。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ぐ使用できます。

--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

Proxmox Backup ドキュメント、リリース 3.4.0-1
--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

```
--ポート <整数> (0 - 65535)
  ポート

--サーバ1 <文字列>
  LDAPサーバーアドレス

--サーバー2 <文字列>
  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 属性がどの属性にマッピングされるかを指定する key=value ペアのカンマ区切りリスト。
  PBS のユーザーフィールドです。 例えば、LDAP 属性の mail を PBS の email にマップするには、次のように書きます。
  email=mail.

-sync-defaults-options[[enable-new=<1|0>][,remove-vanished=<string>]]。
  同期デフォルトオプション

--user-attr <文字列>
  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。

--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 が person,user に設定されている場合、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)
  サーバー証明書の検証
```

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] (プロックスモックスバックアップマネージャネットワーククリエイト<iface> [オプション])

ネットワークインターフェイスの設定を作成します。

<顔

[<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:

--autostart <布尔値>

自動起動インターフェース。

-ボンドプライマリ <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) のボンド送信ハッシュポリシー

--bridge_ports [<string>, ...]

カンマ区切りのネットワークデバイスのリスト。

-bridge_vlan_aware<ブール値

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

インターフェース設定方法

--method6 manual|static|dhcp|loopback

インターフェース設定方法

--mtu <整数> (46 - 65535) (デフォルト=1500)

最大伝送単位。

-スレーブ [<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ースの種類

--vlan-id <整数>

VLAN ID。

--vlan-raw-device <文字列>

ネットワークインターフェース名。

proxmox-backup-manager ネットワークリスト [オプション]

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

:

--output-format text|json|json-pretty

出力フォーマット

proxmox-backup-manager ネットワーククリード

ネットワーク変更の再読み込み

proxmox-backup-manager network remove <iface> [オプション].

ネットワークインターフェイスの設定を削除します。

<顔>

[<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:

-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使用できます。

`proxmox-backup-manager` ネットワークの復帰

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]` (プロックスモックスバックアップマネージャネットワークアップデート `<iface>` [オプション])

ネットワークインターフェイスの設定を更新します。

<顔

[<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:

--autostart <布尔値

自動起動インターフェース。

-bondプライマリ <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) のボンド送信ハッシュポリシー

--bridge_ports [<string>, ...]

カンマ区切りのネットワークデバイスのリスト。

-bridge_vlan_aware<布尔値

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。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ぐ使用できます。

--gateway <文字列

IPv4アドレス。

--gateway6 <文字列

IPv6アドレス。

-メソッド manual|static|dhcp|loopback

インターフェース設定方法

--method6 manual|static|dhcp|loopback

インターフェース設定方法

--mtu <整数> (46 - 65535) (デフォルト=1500)

最大伝送単位。

-スレーブ [<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ースの種類

--vlan-id <整数>
VLAN ID。

--vlan-raw-device <文字列>
ネットワークインターフェース名。

proxmox-backup-manager node show [OPTIONS] (プロックスモックスバックアップマネージャノードショー)
Show node configuration 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ノードの更新[OPTIONS]を実行します。
ノードのコンフィギュレーションを更新し

ます:

--acme [<アカウント=<文字列>].
このノードで使用する acme アカウント。

--acmedomain0 [<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

--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

--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

--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

--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

--暗号-tls-1.2 <文字列>
プロキシが TLS<= 1.2 で使用する OpenSSL 暗号リスト。

--暗号-tls-1.3 <文字列>
プロキシが TLS 1.3 で使用する OpenSSL 暗号スイートリスト

--consent-text <文字列>
同意バナーテキスト

--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は翻訳ファイルでも同じケースを使用します。

--削除

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。指定可能。

--説明 <文字列>

コメント（複数行）。

-digest <文字列>

同時更新を防ぐダイジェスト

--email-from <文字列>

Eメールアドレス

--http-proxy [http://]<host>[:port]

HTTPプロキシの設定 [http://]<host>[:port]

--タスクログ最大日数 <整数> (0 - N)

タスクログの最大保存日数

```
proxmox-backup-manager 通知エンドポイント gotify create <名前> --server
<string> --token <string> [OPTIONS].
```

新しい gotify エンドポイントを追加します。

<名前>

[<string>] ターゲットとマッチャーの名前スキーマ

--サーバ <文字列>

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:

--コメント <文字列>

コメント

--無効 <ブール値>

このターゲットを無効にします。

--フィルタ <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 削除 <名前>
```

gotify エンドポイントを削除します。

<名前>

[<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 リスト [オプション].
```

すべてのエンドポイント

をリストします。

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 通知エンドポイント gotify show <名前> [オプション].`

单一のエンドポイントを表示します。

<名前

[<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

proxmox-backup-manager 通知エンドポイント gotify update <name> [OPTIONS].

gotify エンドポイントを更新しました。

<名前

[<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:

--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。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ぐ使用できます。

--無効 <ブール値

このターゲットを無効にします。

--サーバ <文字列

GotifyサーバーのURL。

--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].

新しい sendmail エンドポイントを追加します。

<名前

[<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

--無効 <ブール値

このターゲットを無効にします。

--フィルタ <文字列

非推奨。

-発信元アドレス <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、プラグインはデフォルトでroot@\$hostname (\$hostnameはノードのホスト名)になります。

--宛先 <文字列

メールの送信。指定可能です。

--メール・ユーザー <文字列>

メールを送信ユーザー。ユーザのメールアドレスはusers.cfgで検索されます。複数指定することもできます。

--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 sendmail 削除 <名前>

sendmail エンドポイントを削除します。

<名前>

[<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 sendmail リスト [OPTIONS].

すべてのエンドポイント

をリストします。

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 通知エンドポイント sendmail show <name> [OPTIONS].

単一のエンドポイントを表示します。

<名前>

[<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

proxmox-backup-manager 通知エンドポイント sendmail update <name> [OPTIONS].

sendmail エンドポイントを更新しました。

<名前>

[<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

削除するプロパティのリスト。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に変更を防止します。これは同時変更を防ぐ使用できます。

--**無効 <ブール値**

このターゲットを無効にします。

-発信元アドレス <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、プラグインはデフォルトでroot@\$hostname (\$hostnameはノードのホスト名) になります。

--宛先 <文字列>

メールの送信。指定可能です。

--メール・ユーザー <文字列>

メールを送信ユーザー。ユーザのメールアドレスはusers.cfgで検索されます。複数指定することもできます。

```
proxmox-backup-manager 通知エンドポイント smtp 作成 <名前>
--from-address <string> --server <string> [OPTIONS].
```

新しいsmtpエンドポイントを追加します。

<名前>

[<string>] ターゲットとマッチャーの名前スキーマ

-発信元アドレス <文字列>

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

--無効 <ブール値>

このターゲットを無効にします。

--宛先 <文字列>

メールの送信。指定可能です。

--メール・ユーザー <文字列>

メールを送信ユーザー。ユーザのメールアドレスはusers.cfgで検索されます。複数指定することもできます。

--モード 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 (安全ではありません)、465 (tls)、または 587 (starttls) です。

--ユーザー名 <文字列

認証時に使用するユーザー名。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 認証方式に対応しています。

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。

<名前

[<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リスト [オプション].

すべてのエンドポイント
をリストします。

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` notification endpoint smtp show <name> [オプション].

単一の終点を表示します。

<名前

[<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

`proxmox-backup-manager` notification endpoint smtp update <name> [オプション].

smtpエンドポイントを更新しました。

<名前

[<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|mailto|mailto-user|password|port|username

削除するプロパティのリスト。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ぐ使用できます。

--無効 <ブール値>

このターゲットを無効にします。

-発信元アドレス <文字列>

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。

--宛先 <文字列>

メールの送信。指定可能です。

--メール・ユーザー <文字列>

メールを送信ユーザー。ユーザのメールアドレスはusers.cfgで検索されます。複数指定することもできます。

--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。

```
proxmox-backup-manager 通知エンドポイント webhook create <name> --method post|put|get --url <string> [OPTIONS].
```

新しいWebhookエンドポイントを追加します。

<名前>

[<string>] ターゲットとマッチャーの名前スキーマ

-メソッド post|put|get (default=post)

使用する HTTP メソッド。

--url <文字列>

HTTP(s)の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

--無効 <ブール値>

このターゲットを無効にします。

--header [名前=<文字列> [, 値=<文字列>]]

HTTP ヘッダーの配列。各エントリは、名前と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。

--origin user-created|builtin|modified-builtin

通知構成エントリのオリジン。

--secret [name=<string> [, value=<string>]] (名前=<文字列> [, 値=<文字列>])

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 エンコード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。

```
proxmox-backup-manager 通知エンドポイントウェブフック削除 <名前>
```

ウェブフックエンドポイントを削除します。

<名前>

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

をリストします。

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 notification endpoint webhook show <name> [オプション].

単一の終点を表示します。

<名前

[<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

proxmox-backup-manager 通知エンドポイント webhook 更新 <名前> [オプション].

ウェブフックエンドポイントを更新します。

<名前

[<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

--delete comment|disable|header|body|secret

削除するプロパティのリスト。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ぐ使用できます。

--無効 <ブール値

このターゲットを無効にします。

--header [名前=<文字列> [, 値=<文字列>]]

HTTP ヘッダーの配列。各エントリは、名前と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>]]

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。

--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> [オプション].`

新しい通知マッチャーを追加。

<名前

[<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:

--コメント <文字列

コメント

--無効 <ブール値

このマッチャーを無効にします。

--反転マッチ <ブール値

フィルター全体の反転マッチ。

-マッチカレンダー <文字列

一致した重大度レベルのリスト。複数回指定できます。

--マッチフィールド <文字列

一致したメタデータ・フィールドのリスト。指定可能。

--マッチ度 <文字列

一致した重大度レベルのリスト。複数回指定できます。

--モード all|any (default=all)

試合結果を組み合わせるモード。

--origin user-created|builtin|modified-builtin

通知構成エントリのオリジン。

--ターゲット <文字列

通知対象。指定可能。

proxmox-backup-manager 通知マッチャ削除 <名前

通知マッチャーを削除します。

<名前

[<string>] ターゲットとマッチャーの名前スキーマ

proxmox-backup-manager 通知マッチャーリスト [OPTIONS].

通知マッチャーの一覧。

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 notification matcher show <name> [オプション].

単一のマッチャーを表示します。

<名前

[<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

```
proxmox-backup-manager notification matcher update <name> [オプション].
```

通知マッチャーを更新しました。

<名前

[<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:

--コメント <文字列

コメント

--削除

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。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ぐ使用できます。

--無効 <ブール値

このマッチャーを無効にします。

--反転マッチ <ブール値

フィルター全体の反転マッチ。

-マッチカレンダー <文字列

一致した重大度レベルのリスト。複数回指定できます。

--マッチフィールド <文字列

マッチしたメタデータ・フィールドのリスト。指定可能。

--マッチ度 <文字列

一致した重大度レベルのリスト。複数回指定できます。

--モード all|any (default=all)

試合結果を組み合わせるモード。

--ターゲット <文字列

通知対象。指定可能。

proxmox-backup-manager 通知対象リスト [OPTIONS] を指定します。

リスト対象。

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 通知対象テスト <名前

指定された通知テストします。

<名前

[<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。

<領域

[<string>] レルム名。

--クライアントID <文字列

OpenIDクライアントID

--issuer-url <文字列>

OpenID Issuer Url オプ

ションのパラメータ:

--acr値 [<文字列>, ...]。

OpenID ACRリスト

--autocreate <boolean> (**default=false**)

ユーザーが存在しない場合、自動的にユーザーを作成します。

--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> (**デフォルト=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。

--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

--username-claim <string>

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> [オプション].

OpenIDレルム設定の削除

<領域>

[<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に変更を防止します。これは同時変更を防ぐ使用できます。

proxmox-backup-manager のopenidリスト [OPTIONS]。

設定されている OpenId レルム一覧 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 openid show <realm> [オプション].

OpenIDレルムの設定を表示

<領域

[<string>] レルム名。

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 openid update <realm> [オプション].
```

OpenIDレルム設定の更新

<領域

[<string>] レルム名。

オプションのパラメータ:

--acr-values [<string>, ...] (--acr-values [<string>, ...])

OpenID ACRリスト

--autocreate <boolean> (default=false)

ユーザーが存在しない場合、自動的にユーザーを作成します。

--クライアント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> (デフォルト=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 client-key|comment|default|autocreate|scopes|prompt|acr-values

削除するプロパティのリスト。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ぐ使用できます。

--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].
```

新しい剪定ジョブを作成します。

<id>

[<string>] ジョブID。

--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:

A.2. proxmox-backup-manager

--コメント <文字列

コメント

-無効 <ブール値> (デフォルト=偽)

このジョブを無効にします。

--keep-daily <整数> (1 - N)

毎日保持するバックアップの数。

--1時間ごと <整数> (1 - N)

1時間ごとに保存するバックアップの数。

--keep-last <整数> (1 - N)

保持するバックアップの数。

--keep-monthly <整数> (1 - N)

毎月保存するバックアップの数。

--キープウィークリー <整数> (1 - N)

週ごとのバックアップ。

--キープ・イヤー・リー <整数> (1 - N)

年間バックアップ数

--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します)

--ns <文字列>

名前空間。

proxmox-backup-managerのprune-jobリスト [OPTIONS]。

すべてのプルーンのジョブ

ブをリストします:

--output-format text|json|json-pretty

出力フォーマット

proxmox-backup-manager prune-job remove <id> [オプション]。

プルーンのジョブ構成を削除します。

<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、変更を防止します。これは同時変更を防ぐ使用できます。

proxmox-backup-manager prune-job run <id> [OPTIONS]。

指定された剪定ジョブの実行

<id>

[<string>] ジョブID。オプショ

ンのパラメータ

--output-format text|json|json-pretty

出力フォーマット

```
proxmox-backup-manager prune-job show <id> [オプション].
```

剪定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

出力フォーマット

proxmox-backup-manager prune-job update <id> [OPTIONS].

プルーンのジョブ設定を更新しました。

<id>

[<string>] ジョブID。オプショ

ンのパラメータ

--コメント <文字列>

コメント

--削除**comment|disable|ns|max-depth|keep-last|keep-our hourly|keep-daily|keep-weekly|keep-monthly|。**

削除するプロパティのリスト。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ぐ使用できます。

-無効 <ブール値> (デフォルト=偽)

このジョブを無効にします。

--keep-daily <整数> (1 - N)

毎日保持するバックアップの数。

--1時間ごと <整数> (1 - N)

1時間ごとに保存するバックアップの数。

--keep-last <整数> (1 - N)

保持するバックアップの数。

--keep-monthly <整数> (1 - N)

毎月保存するバックアップの数。

--キープウィークリー <整数> (1 - N)

週ごとのバックアップ。

--キープ・イヤー・リー <整数> (1 - N)

年間バックアップ数

--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します)

--ns <文字列>

名前空間。

--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剪定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 pull <remote> <remote-store> <store> [OPTIONS].
```

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

<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。

<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

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 Gib...)。

--バーストアウト <文字列

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 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>]。

グループフィルターのリスト。指定可能。

--最大深度 <整数> (0 - 7) (デフォルト=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。

--output-format text|json|json-pretty

出力フォーマット

--rate-in <文字列

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 Gib...)。

--rate-out <文字列

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 Gib...)。

--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、ローカルコピーが削除されます。

--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は、再度ブルしてください。

--トランクファー・ラスト <整数> (1 - N)

グループごとに直近のN個のスナップショットに転送を制限し、他はスキップします。

--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].
```

別のリポジトリにプッシュしてデータストアを同期

<ストア

[<string>] データストア名。

<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。

オプションのパラメータ:

--バーストイン <文字列>

任意の単位のバイトサイズ (B、KB (ベース 10)、MB、GB...、KiB (ベース 2)、MiB、GiB...)。

--バーストアウト <文字列>

任意の単位のバイトサイズ (B、KB (ベース 10)、MB、GB...、KiB (ベース 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>]。

グループフィルターのリスト。指定可能。

--最大深度 <整数> (0 - 7) (デフォルト=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。

--output-format text|json|json-pretty

出力フォーマット

--rate-in <文字列>

任意の単位のバイトサイズ (B、KB (ベース 10)、MB、GB...、KiB (ベース 2)、MiB、GiB...)。

--rate-out <文字列>

任意の単位のバイトサイズ (B、KB (ベース 10)、MB、GB...、KiB (ベース 2)、MiB、GiB...)。

--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、ローカルコピーが削除されます。

--トランスマネージャー・ラスト <整数> (1 - N)

グループごとに直近のN個のスナップショットに転送を制限し、他はスキップします。

--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 remote create <name> --auth-id <string> --host <string> --password <string> [オプション]。
```

新しいリモートを作成します。

<名前>

[<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

-指紋 <文字列

X509 証明書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リモートリスト [OPTIONS].
```

設定されているリモートを一
覧表示します。

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 remote remove <name> [オプション].
```

設定ファイルからリモートを削除します。

<名前

[<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使用でき
ます。

```
proxmox-backup-manager remote show <名前> [オプション].
```

リモート設定の表示

<名前

[<string>] リモート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 remote update <name> [オプション].
```

リモート設定を更新します。

<名前

[<string>] リモートID。オプション

のパラメータ

--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。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に変更を防止します。これは同時変更を防ぐ使用できます。

-**指纹 <文字列**

X509 証明書フィンガープリント (sha256)。

--**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。

--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サブスクリプションの取得[オプション]

購読情報を読む

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 サブスクリプションの削除

購読情報を削除します。

proxmox-backup-manager サブスクリプションセット <キー>

サブスクリプション・キーを設定し、ください。

<キー>

[<string>] Proxmox Backup Serverサブスクリプションキー。

proxmox-backup-manager サブスクリプション set-offline-key <data>

(内部使用のみ!) 署名された購読情報 blob をオフラインキーとして設定します。

<データ>

[<string>] base64 エンコードされた署名付き購読情報

proxmox-backup-managerサブスクリプションの更新[OPTIONS]。

購読ステータスの確認と更新オプションのパラ

メータ

--force <boolean> (デフォルト=false)

キャッシュ内の情報が最新であっても、常にサーバーに接続します。

proxmox-backup-manager sync-job create <id> --remote-store <string> --store <文字列> [OPTIONS]

新しい同期ジョブを作成。

<id>

[<string>] ジョブID。

--リモートストア <文字列>

データストア名。

--ストア <文字列>

データストア名。

オプションのパラメータ:

--バーストイン <文字列>

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 Gib...)。

--バーストアウト <文字列>

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 Gib...)。

--コメント <文字列>

コメント

--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>]。

グループフィルターのリスト。指定可能。

--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します)

--ns <文字列>

名前空間。

--所有者 <文字列>

認証ID

--rate-in <文字列>

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 Gib...)。

--rate-out <文字列>

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 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、ローカルコピーが削除されます。

--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は、再度プルしてください。

--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。

--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

--ransformer・ラスト <整数> (1 - N)

グループごとに直近のN個のスナップショットに転送を制限し、他はスキップします。

--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 sync-job list [OPTIONS]` (プロックスモックスバックアップマネージャ同期ジョブリスト)

ジョブリストを同期します

◦

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

--同期方向 all|push|pull (default=pull)

リストされた同期ジョブの方向: プッシュ、プル、またはすべて。

proxmox-backup-manager sync-job remove <id> [オプション].

同期ジョブ設定の削除

<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に変更を防止します。これは同時変更を防ぐ使用できます。

proxmox-backup-manager sync-job run <id> [OPTIONS].

指定された同期ジョブの実行

<id>

[<string>] ジョブID。オプショ

ンのパラメータ

--output-format text|json|json-pretty

出力フォーマット

proxmox-backup-manager sync-job show <id> [オプション].

同期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

出力フォーマット

proxmox-backup-manager sync-job update <id> [OPTIONS].

同期ジョブの設定を更新します。

<id>

[<string>] ジョブID。オプショ

ンのパラメータ

--バーストイン <文字列>

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 Gib...).

--バーストアウト <文字列>

任意の単位のバイトサイズ (B、KB (ベース 10)、MB、GB...、KiB (ベース 2)、MiB、GiB...)。

--コメント <文字列

コメント

--削除

リモート|所有者|コメント|スケジュール|削除-バニッシュ|グループフィルタ|レートイン|バーストイン|レートアウト|ぶ
削除するプロパティのリスト。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に変更を防止します。これは同時変更を防ぐ使用できます。

--encrypt-only <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>]。

グループフィルターのリスト。指定可能。

--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します)

--ns <文字列>

名前空間。

--owner <文字列>

認証ID

--rate-in <文字列>

任意の単位のバイトサイズ (B、KB (ベース 10)、MB、GB...、KiB (ベース 2)、MiB、GiB...)。

--rate-out <文字列>

任意の単位のバイトサイズ (B、KB (ベース 10)、MB、GB...、KiB (ベース 2)、MiB、GiB...)。

--remote <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、ローカルコピーが削除されます。

--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は、再度プルしてください。

--schedule <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。

--store <文字列>

データストア名。

--sync-direction pull|push (default=pull)

同期ジョブの方向 (プッシュまたはプル

--transfers-per-last <整数> (1 - N)

グループごとに直近のN個のスナップショットに転送を制限し、他はスキップします。

--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タスクリスト [オプション].

実行中のサーバー一覧表示します。

オプションのパラメータ:

--all <ブール値

また、停止中のタスクもリストアップします。

--リミット <整数> (1 - 1000) (デフォルト=50)

リストアップするタスクの最大数。

--output-format text|json|json-pretty

出力フォーマット

proxmox-backup-manager のタスクログ <upid> です。

タスクログを表示します。

<upid>

[<string>] 一意のプロセス/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てみてください。

<upid>

[<string>] 一意のプロセス/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。

<名前>

[<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>

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 GiB...).

--バーストアウト <文字列>

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 GiB...).

--コメント <文字列>

コメント

--rate-in <文字列>

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 GiB...).

--rate-out <文字列>

任意の単位のバイトサイズ (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する 複数指定可

proxmox-backup-manager トラフィック制御リスト [OPTIONS] を指定します。

設定されたトラフィック制御ルールをリスト

トします。オプションのパラメータ:

--output-format text|json|json-pretty

```
proxmox-backup-manager traffic-control remove <name> [オプション].
```

設定トラフィック制御ルールを削除します。

<名前

[<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。これは同時変更を防ぐ使用できます。

`proxmox-backup-manager traffic-control show <name> [オプション].`

トラフィック制御設定の表示

<名前

[<string>] ルールID。オプショ

ンのパラメータです:

--output-format text|json|json-pretty

出力フォーマット

`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:

--output-format text|json|json-pretty

出力フォーマット

`proxmox-backup-manager traffic-control update <name> [OPTIONS].`

交通管制設定の更新

<名前

[<string>] ルールID。オプショ

ンのパラメータです:

--バーストイン <文字列

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 2)、 MiB、 Gib...).

--バーストアウト <文字列

任意の単位のバイトサイズ (B、 KB (ベース 10)、 MB、 GB...、 KiB (ベース 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。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ぐ使用できます。

--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

任意の単位のバイトサイズ (B、KB (ベース 10)、MB、GB...、KiB (ベース 2)、MiB、GiB...)。

--rate-out <文字列

任意の単位のバイトサイズ (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する 複数指定可

```
proxmox-backup-manager user create <userid> [OPTIONS].
```

新しいユーザーを作成

します。

<ユーザーID>

[<string>] ユーザーID オプショ

ンのパラメーター:

--コメント <文字列>

コメント

--email <文字列>

Eメールアドレス

--有効にする <ブール値> (デフォルト=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 <整数> (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' は有効ないことを意味します。

--名 <文字列>

ファーストネーム

--ラストネーム <文字列>

苗字

--パスワード <文字列>

ユーザーパスワード

```
proxmox-backup-manager user delete-token <userid> <token-name> [OPTIONS].
```

ユーザーAPIトークンの削除

<ユーザーID>

[<string>] ユーザーID

<トークン名>

[<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。このような場合は完全なAuthidを使用してください。

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使用できます。

```
proxmox-backup-manager user generate-token <userid> <token-name> [OPTIONS].
```

指定されたメタデータで新しいAPIトークンを生成します。

<ユーザーID>

[<string>] ユーザーID

<トークン名>

[<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。このような場合は完全なAuthidを使用してください。

オプションのパラメータ:

--コメント <文字列>

コメント

-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使用できます。

--有效 <ブール値> (デフォルト=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 <整数> (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' は有効ないことを意味します。

proxmox-backup-manager ユーザーリスト [オプション].

設定されたユーザーを一

覧表示します。

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 user list-tokens <userid> [OPTIONS].

ユーザーに関連付けられたトークンをリストします。

<ユーザーID>

[<string>] ユーザーID オプショ

ンのパラメーター:

--output-format text|json|json-pretty

出力フォーマット

proxmox-backup-manager user permissions <auth-id> [OPTIONS].

ユーザー/トークンの権限を一覧表示します。

<auth-id>

[<string>] 認証ID オプション・パ

ラメーター:

--output-format text|json|json-pretty

出力フォーマット

--パス <文字列>

アクセス制御パス。

proxmox-backup-manager user remove <userid> [オプション].

設定ファイルからユーザーを削除します。

<ユーザーID>

[<string>] ユーザーID オプショ

ンのパラメーター:

-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使用できます。

```
proxmox-backup-manager user tfa delete <userid> <id> [オプション].
```

TFAエントリーを1つ削除します。

<ユーザーID

[<string>] ユーザーID

<id>

[<string>] tfa エントリ ID。

オプションのパラメータ:

--パスワード <文字列

パスワード

```
proxmox-backup-manager user tfa list <userid> [オプション].
```

ユーザーのすべての tfa メソッドをリストします。

<ユーザーID

[<string>] ユーザーID オプショ

ンのパラメーター:

--output-format text|json|json-pretty

出力フォーマット

```
proxmox-backup-manager user tfa unlock <userid
```

ユーザーのTFA認証を解除します。

<ユーザーID

[<string>] ユーザーID

```
proxmox-backup-manager user update <userid> [オプション].
```

ユーザー設定の更新ユーザー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用します。

<ユーザーID

[<string>] ユーザーID オプショ

ンのパラメーター:

--コメント <文字列

コメント

--delete comment|firstname|lastname|email

削除するプロパティのリスト。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ぐ使用できます。

--email <文字列

Eメールアドレス

--有効にする <ブール値> (デフォルト=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 <整数> (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' は有効ないことを意味します。

--名 <文字列>

ファーストネーム

--ラストネーム <文字列>

苗字

--パスワード <文字列>

ユーザー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> [オプション].
```

バックアップの確認

<ストア>

[<string>] データストア名。

オプションのパラメータ:

--ignore-verified <boolean> (default=true)

検証が古くない場合は、検証済みの検証しないでください。

--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

出力フォーマット

```
proxmox-backup-manager verify-job create <id> --store <string> [OPTIONS].
```

新しい検証ジョブを作成します。

<id>

[<string>] ジョブID。

--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

--ignore-verified <boolean> (default=true)

検証が古くない場合は、検証済みの検証しないでください。

--最大深度 <整数> (0 - 7) (デフォルト=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。

--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ベリファイジョブを実行します。

`proxmox-backup-manager`の`verify-job list [OPTIONS]`を指定します。

すべての検証ジョブのリスト

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 verify-job remove <id> [オプション].
```

検証ジョブ構成の削除

<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に変更を防止します。これは同時変更を防ぐ使用できます。

```
proxmox-backup-manager verify-job run <id> [オプション].
```

指定された検証ジョブの実行

<id>

[<string>] ジョブID。オプショ

ンのパラメータ

--output-format text|json|json-pretty

出力フォーマット

```
proxmox-backup-manager verify-job show <id> [オプション].
```

検証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

出力フォーマット

```
proxmox-backup-manager verify-job update <id> [OPTIONS].
```

検証ジョブ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

--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。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ぐ使用でき

ます。

--ignore-verified <boolean> (default=true)

検証が古くない場合は、検証済みの検証しないでください。

--最大深度 <整数> (0 - 7) (デフォルト=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。

--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ベリファイ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のバージョン [オプション] .

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です:

--output-format text|json|json-pretty

出力フォーマット

--verbose <boolean> (default=false)

冗長なパッケージ出力します。output-format が指定されている場合は無視されます。

A.3 プロックステープ

proxmox-tape backup <store> <pool> [OPTIONS].

テープ・メディア・プールへのデータストアのバックアップ

<ストア

[<string>] データストア名。

<プール

[<string>] メディアプール名。

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。

--eject-media <プール値>

ジョブ時にメディアを排出します。

--エクスポートメディアセット <プール値>

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>]。

グループフィルターのリスト。指定可能。

-latest-only<プール値>

最新のスナップショットのみをバックアップします。

--最大深度 <整数> (0 - 7) (デフォルト=7)

何レベルの名前空間を操作するか (0== 再帰なし)

--通知ユーザー <文字列>

ユーザーID

--ns <文字列>

名前空間。

--output-format text|json|json-pretty

出力フォーマット

```
proxmox-tape backup-job create <id> --drive <string> --pool <string>
--store <string> [OPTIONS].
```

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 <ブール値>

ジョブ時にメディアを排出します。

--エクスポートメディアセット <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。

```
--group-filter [<exclude:>|<include:>]<type:<vm|ct|host>|group:GROUP|正規表現
:RE>]。
```

グループフィルターのリスト。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。

--最大深度 <整数> (0 - 7) (デフォルト=7)

何レベルの名前空間を操作するか (0== 再帰なし)

--notification-mode legacy-sendmail|notification-system
(default=legacy-sendmail)

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。

notify-user で設定されたユーザーに、システムのsendmail実行ファイルを介して通知を送ります。

通知システムは、一致する通知イベントを通知発行します。

--通知ユーザー <文字列>

ユーザーID

--ns <文字列>

名前空間。

--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。

--worker-threads <integer> (1 - 32) (default=1)

ジョブに使用するワーカースレッド数の設定

proxmox-tapeバックアップジョブリスト [OPTIONS].

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

:

--output-format text|json|json-pretty

出力フォーマット

proxmox-tape backup-job remove <id> [オプション].

テープバックアップジョブ構成の削除

<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に変更を防止します。これは同時変更を防ぐ使用できます。

proxmox-tape バックアップジョブの実行 <id>

THapeバックアップジョブの実行

<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

<id>

[<string>] ジョブID。オプショ

ンのパラメータ

--output-format text|json|json-pretty

出力フォーマット

proxmox-tape backup-job update <id> [OPTIONS].

テープバックアップジョブの更新

<id>

[<string>] ジョブID。オプショ

ンのパラメータ

--コメント <文字列

コメント

--削除コメント|スケジュール|イジェクトメディア|エクスポートメディアセット|最新のみ|通知ユーザー|通知モード
削除するプロパティのリスト。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ぐ使用できます。

--ドライブ <文字列>

ドライブ識別子。

--eject-media <ブール値>

ジョブ時にメディアを排出します。

--エクスポートメディアセット <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。

--group-filter [<exclude:>|<include:>]<type:<vm|ct|host>|group:GROUP|正規表現
:RE>]。
グループフィルターのリスト。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。

--最大深度 <整数> (0 - 7) (デフォルト=7)
何レベルの名前空間を操作するか (0== 再帰なし)

--notification-mode legacy-sendmail|notification-system
(default=legacy-sendmail)
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* 実行ファイルを経由して *notify-user* で設定されたユーザーに通知を送ります。
通知システムは、一致する通知イベントを通知発行します。

--通知ユーザー <文字列
ユーザーID

--ns <文字列
名前空間。

--プール <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数の設定

proxmox-tape barcode-label [OPTIONS] チェンジャ

ーデバイスからのバーコードでメディアにラベルを貼りま
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
--プール <文字列
メディアプール名。

proxmox-tape cartridge-memory [OPTIONS] (プロックスモックスステープカートリッジメモリ)

Read Cartridge Memory (Medium 補助メモリー属性) オプション・パラメー

ター：

--ドライブ <文字列
ドライブ識別子。

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：

--ドライブ <文字列>

ドライブ識別子。

--force <ブール値>

既存のインデックスを強制的に上書きします。

--output-format text|json|json-pretty

出力フォーマット

--スキャン <ブール値>

保存されたバージョンを復元する代わりに、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>

詳細モード - 見つかったチャックをすべて記録します。

`proxmox-tape changer config <name> [OPTIONS].`

テープチェンジャーの設定を取得

<名前>

[<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・デバイスの作成

<名前>

[<string>] テープチェンジャー識別子。

--パス <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、テープはアンロード前に手動で排出されます。

--export-slots [<integer>, ...] (--export-slots [<integer>, ...])

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].`

リストチェンジャー 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 remove <名前>.`

テープチェンジャー構成の削除

<名前

[<string>] テープチェンジャー識別子。

プロックスモックス・テープ・チェンジャー・スキャン[OPTIONS].

SCSIテープ・チェンジャーのスキャン

オプション・パラメーター:

--output-format 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

<名前

[<string>] テープチェンジャー識別子。

オプションのパラメータ:

--キヤッショ <ブール値> (デフォルト=true)

キヤッショされた使用します。

--output-format text|json|json-pretty

出力フォーマット

proxmox-tape changer transfer [<name>] --from <integer> (1 - N --to)

<整数> (1 - N)

あるスロットから別のスロットへメディアを転送

<名前

[<string>] テープチェンジャー識別子。

--から <整数> (1 - N)

ソーススロット番号

--から <整数> (1 - N)

送信先スロット番号

proxmox-tape changer update <name> [OPTIONS].

テープチェンジャー構成の更新

<名前

[<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。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ぐ使用できます。

--eject-before-unload <boolean> (default=false)

trueに設定すると、テープはアンロード前に手動で排出されます。

--export-slots [<integer>, ...] (--export-slots [<integer>, ...])

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。

--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)
```

クリーンドライブ

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] (プロックスモックス・テープ・ドライブ・コンフィグ <名前> [オプション])
```

プール設定の取得

<名前

[<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

出力フォーマット

```
proxmox-tape drive create <name> --path <string> [OPTIONS].
```

新しいドライブの作成

<名前

[<string>] ドライブ識別子。

--パス <文字列>

LTO SCSI汎用テープ・デバイスへのパス (例: '/dev/sg0') オプションの

パラメータ:

--チェンジャー <文字列>

テープチェンジャー識別子。

--changer-drivenum <整数> (0 - 255) (デフォルト=0)

関連チェンジャーのドライブ番号 (オプションチェンジャーが必要です)

```
proxmox-tape ドライブリスト [OPTIONS].
```

リストドライブ

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>.
```

ドライブ構成の削除

<名前

[<string>] ドライブ識別子。

proxmox-tape ドライブスキャン [OPTIONS] を選択します。

ドライブのスキャン 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 update <name> [OPTIONS] [オプション].

ドライブ構成の更新

<名前

[<string>] ドライブ識別子。

オプションのパラメータ:

--チェンジャー <文字列

テープチェンジャー識別子。

--changer-drivenum <整数> (0 - 255) (デフォルト=0)

関連チェンジャーのドライブ番号 (オプションチェンジャーが必要です)

--delete changer|changer-drivenum

削除するプロパティのリスト。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ぐ使用できます。

--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] (ブロックスマックス・テープ・イJECT)

ドライブメディアの取り出し/アンロー

ド 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

proxmox-tape eod [OPTIONS] (ブロックスマックス・テープ・イーオーディー)

メディア終端への移動 (MTEOM、デバッグに使用) 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 export-media <label-text> [OPTIONS].
```

ラベルを指定してメディアをエクスポート

<ラベルテキスト

[<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・テープ・フォーマット)

フォーマット・メディア

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。

--fast <boolean> (default=true)

高速消去を使用します。

--output-format text|json|json-pretty

出力フォーマット

`proxmox-tape 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。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 <ブール値>

冗長なヘルプ。

`proxmox-tape inventory [OPTIONS]` (プロックスモックス・テープ・インベントリー)

メディア・ラベルの一覧表示（および更新）（Changer Inventory）

オプション・パラメータ:

--カタログ <boolean> (デフォルト=false)

テープからカタログを復元してみてください。

--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

--read-all-labels <boolean> (default=false)

すべてのテープをセットし、ラベルを読んでみてください。

--read-labels <boolean> (default=false)

未知のテープをセットし、ラベルを読み取ります。

```
proxmox-tape key change-passphrase <fingerprint> --hint <string> [OPTIONS].
```

暗号化キーのパスワードを変更します。

<指紋

[<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> (デフォルト=false)

テープキーのパスフレーズをリセットします。

--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機能。

proxmox-tape key create --hint <string> [OPTIONS].

キーを作成（標準入力からパスワードを読み込み）

--hint <文字列>

パスワード復元ヒント

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機能。

proxmox-tapeキーリスト [OPTIONS].

キー一覧

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 key paperkey <fingerprint> [オプション].

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。これにはスキャン

可能な QR コードも含まれているため、キーをすばやく復元できます。

<指紋>

[<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|html

Paperkey 出力フォーマット

--件名 <文字列>

指定され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] [オプション].

データベースから暗号化キーを削除

このキーを使用してテープにアクセスすることはできませんのでご注意ください。

<指紋>

[<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に変更を防止します。これは同時変更を防ぐ使用できます。

```
proxmox-tape key restore [OPTIONS] (プロックステープ・キー・リストア)
```

テープまたはバックアップファイル/文字列から暗号化キーを復元（標準入力からパスワードを読み込む）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。

--キー <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。

--キーファイル <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> [オプション].
```

暗号化キーのメタデータを表示します。

<指紋>

[<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

出力フォーマット

```
proxmox-tape label --label-text <string> [OPTIONS].
```

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：

--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

--ブール <文字列>

メディアブール名。

```
proxmox-tape load-media <label-text> [OPTIONS].
```

指定したラベルでメディアをセット

<ラベルテキスト>

[<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。

出力フォーマット

`proxmox-tape load-media-from-slot <source-slot>[OPTIONS]` (プロックスモックス・テープ・ロード・メディア・フロム・スロット<ソース・スロット> [オプション])。

指定し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` のメディアコンテンツ [OPTIONS]。

メディアの内容を一覧表

示 オプションのパラメー

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

--ラベルテキスト <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・ワイド

--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

出力フォーマット

--プール <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 (データベースから完全に削除)

<ラベルテキスト

[<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 <ブール値>

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・ワイド

`proxmox-tape` メディアリスト [OPTIONS].

リスト・プール・メディ

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てください（オンラインになっているテープを確認してください）。

```
--update-status-changer <string>.
```

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

<名前

[<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
```

出力フォーマット

```
proxmox-tape pool create <name> [OPTIONS] [オプション].
```

新しいメディアプールの作成

<名前

[<string>] メディアプール名。

オプションのパラメータ:

```
--allocation <文字列
```

メディアの割り当てポリシー（「継続」、常に」、またはカレンダーイベント）。

```
--コメント <文字列
```

コメント

```
--暗号化 <文字列
```

テープ暗号化キーの指紋（sha256）。

```
--保持 <文字列
```

メディア保持ポリシー（「上書き」、「保持」、または期間）。

```
--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ことがあります）。

```
proxmox-tape プールリスト [OPTIONS].
```

リスト・メディア・プー

ル オプション・パラメー

ター:

```
--output-format text|json|json-pretty
```

出力フォーマット

```
proxmox-tape pool remove <名前>.
```

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].

メディアプール設定の更新

<名前

[<string>] メディアプール名。

オプションのパラメータ:

--allocation <文字列>

メディアの割り当てポリシー（「継続」、常に」、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。指定可能。

--暗号化 <文字列>

テープ暗号化キーの指紋（sha256）。

--保持 <文字列>

メディア保持ポリシー（「上書き」、「保持」、または期間）。

--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ことがあります）。

proxmox-tape read-label [OPTIONS].

メディア・ラベルの読み

取り 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。

--inventorize <ブール値>

メディアの発明

--output-format text|json|json-pretty

出力フォーマット

proxmox-tape restore <メディアセット> <ストア> [{<スナップショット>}] [OPTIONS].

メディアセットからデータを復元

<メディアセット>

[<string>] メディアのUUIDを設定します。

<ストア>

[[(<source>=<target>, ...)] を参照してください。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' にマッピングし、他のすべてのソースを既定の 'e' にマッピングします。既定値が指定されていない場合は、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>]]。
リストアするネームスペースのリスト。複数回指定できます。

--通知ユーザー <文字列

ユーザーID

--output-format text|json|json-pretty

出力フォーマット

--所有者 <文字列>

認証ID

proxmox-tape巻き戻し [OPTIONS]。

テープ巻き戻し オプショ

ン・パラメーター:

--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] (プロックスステープスキャン)

巻き戻し、メディア内容の読み込み、デバッグ情報の表示

注: ドライバがIO Errorを返さない限り、このメソッドは読み込みますのでEOTに達すると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] (プロックスモックスステープステータス)

Get drive/media status 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 unload [OPTIONS] (プロックスモックス・テープ・アンロード)

チェンジャー経由でメディアをアンロ

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

--ターゲットスロット <整数> (1 - N)

ターゲット・スロット番号。省略した場合は、ドライブがロードされたスロットがデフォルトになります。

proxmox-tape volume-statistics [OPTIONS] (プロックスモックスステープボリューム統計)

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

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の設定にある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。これは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）。可読形式。

json

JSON（1行）。

json-プリティ

JSON（複数行、きれいにフォーマットされています）。

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・カウント後）に配置

<カウント

[<integer> (0 - 2147483647)] ファイルマークの位置（0はBOT）。

pmt bsf <カウント

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）。

テープは前の最後のブロックに配置されます。

<カウント

[<integer> (1 - 2147483647)] ファイルマーク数。

pmt bsfm <カウント

後方スペースでファイルをカウントし、1レコード（ファイルマークの後の位置）を前方スペースにします。

これにより、テープは現在のファイルよりカウント-1ファイル前のファイルの最初のブロックに位置することになります。

<カウント

[<integer> (1 - 2147483647)] ファイルマーク数。

pmt bsr <カウント

後方空間の記録。

<カウント

[<integer> (1 - 2147483647)] レコード数。

pmt カートリッジメモリー

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)

高速消去を使用してください。

`pmt format [OPTIONS] メディアの`

フォーマット、シングルパーティショ

ンオプションのパラメータ：

--fast <boolean> (default=true)

高速消去を使用してください。

`pmt fsf <カウント`

フォワード・スペース・カウント・ファイル（ファイル・マーク後

の位置）。テープは次の最初のブロックに配置されます。

<カウント

[<integer> (1 - 2147483647)] ファイルマーク数。

`pmt fsm <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、次にバックワードスペースで1レコード（ファイルマークの前の位置）をカウントします。

これにより、テープは現在のファイルからカウント-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。

<カウント

[<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の記録

<カウント

[<integer> (1 - 2147483647)] レコード数。

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 <ブール値

冗長なヘルプ。

pmtロード

ロードメディア

PMTロック

テープドライブドアのロック

pmt options [OPTIONS] 各種ド

ライブオプションの設定 オプション

パラメータです：

--block-size <整数> (0 - 8388608)

テープ・ドライブの block_length を設定します（0 は可変長）。

--batch-mode <ブール値

ドライブバッファを使用します。

--compress <ブール値

圧縮の有効/無効。

--defaults <ブール値

デフォルトオプションの設定

PMTリワインド

テープの巻き戻し

pmt スキャン

既存のテープチェンジャー機器のスキャン

PMTステータス

ドライブステータス

pmt テープアラートフラグ

テープ警告フラグの読み取り

PMTロック解除

テープドライブドアのロック解除

PMT出来高統計

pmt weof [<カウント>].

書き込み回数（デフォルト 1） 現在のEOF マーク。

<カウント

[<integer> (1 - 2147483647)] ファイルマーク数。

A.5 pmtx

`pmtx 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 <ブール値

冗長なヘルプ。

PMTX問い合わせ

お問い合わせ

PMTXインベントリー

インベントリー

`pmtx load <slot> [OPTIONS].`

負荷

<スロット

[<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:

--ドリヴェナム <整数

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機器のスキャン

PMTXステータス

チェンジャー・ステータス

`pmtx 転送 <from> <to>`

譲渡

<から

[<integer>] ソースストレージのスロット番号。

<へ

[<integer>] ターゲットストレージのスロット番号。

pmtx unload [OPTIONS] (アンロード)

アンロード

オプションのパラメータ:

--ドリヴェナム <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 <整数>

ストレージ・スロット番号（ターゲット）。省略した場合、デフォルトはドライブがロードさスロットです。

A.6 ピクサー

`pxar create <アーカイブ> <ソース> [オプション].`

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です。

オプションのパラメータ:

--all-file-systems <boolean> (default=false)

マウントされたスディールを含みます。

--entries-max <integer> (0 - 9223372036854775807) (default=1048576)

メモリに一度にロードされるエントリの最大数

--除外 <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。複数回指定できます。

--no-acls <ブール値> (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)

フィフオは無視。

--no-sockets <boolean> (default=false)

ソケットは無視。

--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>] [OPTIONS].`

アーカイブを展開します。

<アーカイブ>

[<string>] アーカイブ名。

<ターゲット

[<string>] 対象ディレクトリ オプ

ションのパラメータ:

--allow-existing-dirs <boolean> (default=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。

--ファイル <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 <ブール値> (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)

フィフオは無視。

--no-sockets <boolean> (default=false)

ソケットは無視。

--no-xattrs <boolean> (default=false)

拡張ファイル属性を無視します。

--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)

アーカイブズハードリンクによる既存のエントリーの上書き

--overwrite-symlinks <boolean> (default=false)

既に存在するエントリをシンボリックリンクで上書きします。

--パターン <文字列>

リストアするファイルを制限するパスまたはマッチパターン。複数回指定できます。

-ペイロード入力 <文字列>

'pxar' は分割アーカイブを復元するためのペイロード入力データファイルです。

-プレリュード・ターゲット <文字列>

pxar アーカイブのプレリュードをリストアパス。

--strict <boolean> (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].`

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 <ブール値>

冗長なヘルプ。

`pxar list <アーカイブ> [オプション].`

アーカイブの内容を一覧表示します。

<アーカイブ

[<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。

```
pxar mount <アーカイブ> <マウントポイント> [オプション].
```

提供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

[<string>] アーカイブ名。

<マウントポイント

[<string>] ファイルマウントポイント。

オプションのパラメータ:

-ペイロード入力 <文字列

'ppxar' は分割アーカイブを復元するためのペイロード入力データファイルです。

--verbose <boolean> (default=false)

フォアグラウンドで実行される詳細出力デバッグ用)。

A.7 プロックスファイルリストア

```
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。

<スナップショット

[<string>] グループ/スナップショットパス。

<パス

[<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)

設定されている場合、'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、署名のみを行うか、またはどちらも行わ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)

すで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。

--キーファイル <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。

--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 [{<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 <ブール値

冗長なヘルプ。

proxmox-file-restore list <スナップショット> </パス> [オプション].

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。

</パス

[<string>] (サブ)リストへのパス。

オプションのパラメータ:

--base64 <boolean> (default=false)

設定されている場合、'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、署名のみを行うか、またはどちらも行わ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。

--keyfd <整数> (0 - N)

すで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。

--キーファイル <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。

--ns <文字列

名前空間。

--output-format 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。

--タイムアウト <整数> (1 - N)

通話にかかる最大時間を指定します。

```
proxmox-file-restore status [OPTIONS] ブロックスマックスファイルリストアステータス [OPTIONS] ブロックスマックスファイルリストアステータス
```

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:

-- 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 <名前>.
```

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。

<名前>

[<string>] 停止するVMの名前。

A.8 proxmox-backup-debug

```
proxmox-backup-debug api create <api-path> [OPTIONS] [オプション].
```

<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
```

出力フォーマット

```
proxmox-backup-debug api delete <api-path> [OPTIONS] (オプション)
```

<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
```

出力フォーマット

```
proxmox-backup-debug api get <api-path> [OPTIONS].
```

<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
```

出力フォーマット

```
proxmox-backup-debug api ls [<path>] [OPTIONS].
```

</パス> のAPI使用情報を取得

</パス>

[<string>] API パス。

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-debug api set <api-path> [OPTIONS] [オプション].

<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

出力フォーマット

proxmox-backup-debug api usage <path> [OPTIONS].

</パス> の API 使用情報を取得

</パス

[<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].

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 フラグを指定すると、mtime は無視され、ファイルの内容が比較されます。

<プリバースナップショット

[<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

[<string>] バックアップアーカイブ名。

オプションのパラメータ:

-色 always|auto|never (default=auto)

カラー出力オプション

-comparare-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 <整数> (0 - N)

すで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。

--キーファイル <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。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 <ブール値

冗長なヘルプ。

```
proxmox-backup-debug inspect chunk <chunk> [OPTIONS].
```

チャunkの検査

<チャunk

[<string>] チャunkファイル。

オプションのパラメータ:

--デコード <文字列

チャunkをデコードするファイルへのパス。

-ダイジェスト <文字列

リファレンスを検索する際に必要で、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unkが暗号化されたキーファイルへのパス。

--output-format text|json|json-pretty

出力フォーマット

--参照フィルタ <文字列

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、ファイル名は参照検索とデコード検証のためのダイジェストとして使用されるべきです。

```
proxmox-backup-debug inspect device <device> [OPTIONS].
```

デバイスにデータストアがあるかどうかを検査します。

<デバイス

[<string>] デバイスパス、通常は /dev/...

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-debug inspect file <file> [オプション].
```

デコードされていない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。'-' は標準出力へのデコード。

--キーファイル <文字列

ファイルを暗号化したキーファイルへのパス。

--output-format text|json|json-pretty

出力フォーマット

```
proxmox-backup-debug recover index <file> <chunks> [OPTIONS].
```

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、および復号化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。

<ファイル

[<string>] インデックスファイル (.fidxまたは.didx)へのパス。

<チャンク

[<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 <boolean> (default=false)

チャンクが破損している場合は、警告を表示し、代わりに0バイトを書き込み、部分的な回復を試みます。

--ignore-missing-chunks <boolean> (default=false)

チャンクが見つからない場合は、警告を表示し、代わりに0バイトを書き込み、部分的な回復を試みます。

--キーファイル <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検証をスキップすることで、リストア速度が大幅に向上します。

設定ファイル

すべてのProxmox Backup Server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。

広める
パーミッションの階層への伝搬
パス
オブジェクトのパス
ユーザー/トークン
ユーザーとトークンのリスト
役割
割り当てられた役割のリスト

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 役割

以下の役割が:

管理者
管理者
監査
監査役
アクセス禁止
アクセス禁止
データストア管理
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・バックアップ (所有バックアップのバックアップとリストア)

データストアパワーユーザー

データストアPowerUser (バックアップ、リストア、所有バックアップの削除)

データストア監査

データストア監査人

リモート監査

リモート監査人

リモート管理

リモート管理者

リモートシンクオペレーター

同期オペレーター

RemoteSyncPushOperator

シンクロオペレーター (プッシュ方向)

リモートデータストアPowerUser

リモート・データストア・プルーン

リモートデータストアアドミン

リモートデータストア管理

テープ監査

テープ監査人

テープ管理

テープ管理者

テープオペレーター

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 datastore.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>` の後にデータストア設定オプションを続けます。

```
データストア:< name1>
パス< /path1>
<option1>< value1>
データストア:< name2>
path< /path2>
...<option2>< value2>
```

このファイルを操作するには`proxmox-backup-manager`データストアコマンドを使用します。

B.2.2 オプション

必要なプロパティ:

パス

[<string>] データストア・ディレクトリへの絶対パスか、リムーバブル・データストアの場合はデバイス上の相対パスです。

オプションのプロパティ:

バックイングデバイス

[<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です。

コメント

[<string>] コメント

gcスケジュール

[<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>]]] とします。] メンテナンスマード、タイプは'オフライン' または'読み込み専用'、メッセージは " type= 読み込み専用|オフライン|削除|アンマウント" で囲む必要があります。

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=legacy-sendmail)] このデータストアの通知をどのようにするかを設定します。 legacy-sendmail は、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に電子メール通知を送信します。

知らせる

= [[[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。

通知を送信するタイミング

prune= never|always|error
通知を送信するタイミング

sync= never|always|error
通知を送信するタイミング

verify= never|always|error
通知を送信するタイミング

通知ユーザー

[<string>] ユーザーID

プルンスケジュール

[<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>]]] を参照してください。 [,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unkのソート順

gc-atime-cutoff= <整数> (1 - 2880) (デフォルト=1445)

ガベージコレクション・フェーズ2でのチャunk・クリーンアップのatimeチェックのカットオフ（分単位）
(デフォルト24時間5分)

gc-atime-safety-check= <boolean> (default=true)
ストアの作成とガベージコレクションの間、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に行われる同期のレベル。

ベリニュー

[<boolean>] 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。

キープデイリー

[<integer> (1 - N)] 毎日保持するバックアップの数。

毎時

[<integer> (1 - N)] 1時間ごとに保持するバックアップの数。

キープラスト

[<integer> (1 - N)] 保存するバックアップの数。

毎月

[<integer> (1 - N)] 毎月の数。

キープウィークリー

[<integer> (1 - N)] 毎週保持するバックアップの数。

ねんかん

[<integer> (1 - N)] 年間数。

B.3 domains.cfg

B.3.1 ファイル形式

このファイルには認証レルムのリストが含まれています。

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 レルムでは、LDAP バインドパスワードは `ldap_passwords.json` に格納されます。

```
オープンID: マスター
  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サーバー
  base-dn OU= People,DC= ldap-server,DC= example,DC= com mode ldaps

  server1 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` はこの操作するコマンドを追加します。

B.3.2 オプション

セクション・タイプ「ad」：ADレルム構成プロパティ。

必要なプロパティ：

サーバ1

[<string>] ADサーバーアドレス

オプションのプロパティ：

ベースデン

[<string>] LDAP ドメイン

バインドエン

[<string>] LDAP ドメイン

カパス

[<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、/etc/ssl/certs にあるシステムの既定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。

フィルター

[<string>] ユーザー同期用のカスタムLDAP検索フィルター

モード

[ldap|ldap+starttls|ldaps (default=ldap)] LDAP接続タイプ [<integer> (0 - 65535)] ADサー

ポートバーポート

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>]]]。カンマ区切りの key=value ペアのリストで、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します。例えば、LDAP 属性の mail PBS の email にマッピングするには、email=mail と記述します。email= <string>

ユーザーのメールアドレスを含むLDAP属性名

名= <文字列

ユーザーのファーストネームを含むLDAP属性名

ラストネーム= <文字列

ユーザーの姓を含むLDAP属性名

同期デフォルトオプション

[enable-new=<1|0>] [,remove-vanished=<文字列>]]]。

sync defaults オプション enable-new= <boolean>

同期後の新規ユーザーの有効化

remove-vanished= [acl|entry|properties, ...].

セミコロンで区切られた、ユーザ同期中にユーザが消えたときに削除するもののリスト。`entry`は同期から返されなかったユーザーを削除します。

`acl` は、ユーザが同期から戻らなかった場合に ACL を削除します。

ユーザークラス

[<string>, ...] (default=inetorgperson, posixaccount, person, user)] を参照してください。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 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。

ベリファイ

[<boolean> (default=false)] サーバ証明書の検証

セクション・タイプ 'ldap': LDAP 設定プロパティ。

必要なプロパティ：

ベースディレクトリ

[<string>] LDAP ドメイン

サーバ1

[<string>] LDAP サーバーアドレス

ユーザー名

[<string>] ユーザー名属性。LDAPへのuseridをLDAP dnにマップするために使用します。

オプションのプロパティ：

バインディング

[<string>] LDAP ドメイン

カーパス

[<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、/etc/ssl/certs にあるシステムの既定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。

フィルター

[<string>] ユーザ同期用のカスタムLDAP検索フィルター

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>]]]。カンマ区切りの key=value ペアのリストで、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します。例えば、LDAP 属性の mail PBS の email にマッピングするには、email=mail と記述します。email= <string>

ユーザーのEメールアドレスを含むLDAP属性名

名= <文字列>

ユーザーのファーストネームを含むLDAP属性名

ラストネーム= <文字列>

ユーザーの姓を含むLDAP属性名

同期デフォルトオプション

[enable-new=<1|0>] [, remove-vanished=<文字列>]]。

sync defaults オプション enable-new= <boolean>

同期後の新規ユーザーの有効化

remove-vanished= [acl|entry|properties, ...].

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えたときに削除するもののリストです。

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。

ユーザクラス

[[<string>, ...] (default=inetorgperson,posixaccount,person,user)] を参照してください。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 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。

ペリファイ

[<boolean> (default=false)] サーバ証明書の検証

セクションタイプ 'openid': OpenID 構成プロパティ。

必要なプロパティ：

クライアントID

[<string>] OpenIDクライアントID

発行者URL

[<string>] OpenID発行元URL

オプションのプロパティ：

アーク値

[[<文字列>, ...]].OpenID ACRリスト

オートクリエイト

[<boolean> (default=false)] ユーザが存在しない場合、自動的に作成します。

クライアントキー

[<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。

迅速

[<string>] OpenIDプロンプト

規模

[[<文字列>, ...] (デフォルト=電子メールプロファイル)]。OpenIDスコープリスト

ユーザー名-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。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。

必要なプロパティ:

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。

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。

必要なプロパティ:

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、ヘッダプール: <name> で始まり、メディアプールコンフィギュレーションオプションが続きます。

```
プール: company1
    アロケーション常に保持上書き
プール: ...
```

このファイルを操作するには proxmox-tape pool コマンドを使用します。

B.4.2 オプション

オプションのプロパティ:

割当

[<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

エンクリプト

[<string>] テープ暗号化キーの指紋 (sha256)。

保持

[<string>] 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

[<string>] メディア・セット命名テンプレート (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ています：

窮み

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。

アカメドメイン2

ACMEドメイン。

acmedomain3

ACMEドメイン。

acmedomain4

ACMEドメイン。

エイチティーティーピーブロキシ

aptとサブスクリプションチェックのプロキシを設定します。

メールフロム

通知をするフォールバックメール。

暗号-TLS-1.3

プロキシが使う 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)。

暗号-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で使用されるデフォルト言語。

記述

ノードの説明。

タスクログ最大日数

タスクログの最大保存日数。

例えば

```
アクメ: 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@gmail.com
ciphers-tls-1.3: TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_8_SHA256:TLS_CHACHA20_POLY1305_SHA256
ciphers-tls-1.2: RSA_WITH_AES_128_CCM:DHE_RSA_WITH_AES_128_CCM default-lang: ja
説明プライマリ PBS インスタンス task-log-max-
days: 30
```

このファイルを操作するにはproxmox-backup-managerノードコマンドを使用します。

B.6 通知.cfg

B.6.1 ファイル形式

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「フィルター」： 旧フィルタ設定 - いずれ削除できます。

セクションタイプ 'gotify': Gotify 通知エンドポイントの設定

必要なプロパティ：

サーバー

[<string>] Gotify サーバー URL。

オプションのプロパティ：

コメント

[<string>] コメント

無効にする

[<boolean>] このターゲットを無効にします。

フィルター

[<string>] 非推奨。

起原

[user-created|builtin|modified-builtin] 通知構成エントリの生成元。

セクションタイプ「グループ」： 旧ターゲットグループの設定 - いずれ削除可能です。

セクションタイプ 'matcher' 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

無効にする

[<boolean>] このマッチャーを無効にします。

インバートマッチ

[<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>] 一致した重大度レベルのリスト。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。指定可能。

一致度

[<string>] 一致した重大度レベルのリスト。指定可能。

モード

[all|any (default=all)] マッチの結果を組み合わせるモード。

起原

[user-created|builtin|modified-builtin]通知構成エントリの生成元。

ターゲット

[<string>]通知対象。指定可能。

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:

著者

[<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

無効にする

[<boolean>] このターゲットを無効にします。

フィルター

[<string>] 非推奨。

フォームアドレス

[<string>] 送信。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。

メール

[<string>] 送信メールアドレス。指定可能です。

メールトゥユーザー

[<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。

起原

[user-created|builtin|modified-builtin] 通知構成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

必要なプロパティ:

フォームアドレス

[<string>] 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ことがあります。メールの *From* ヘッダは \$author に設定されます。
<\$from-address>.

サーバー

[<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:

著者

[<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

無効にする

[<boolean>] このターゲットを無効にします。

メール

[<string>] 送信メールアドレス。指定可能です。

メールトゥユーザー

[<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。

モード

[insecure|starttls|tls (default=tls)] 接続セキュリティ

起原

[user-created|builtin|modified-builtin] 通知構成エントリの生成元。

ポート

[<integer> (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。

ユーザー名

[<string>] 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認証方式に対応しています。

セクションタイプ 'webhook': Webhook 通知エンドポイントの設定

必要なプロパティ:

方法

[post|put|get (default=post)] HTTP メソッドを使用します。

url

[<string>] HTTP(s)の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ます。

コメント

[<string>] コメント

無効にする

[<boolean>] このターゲットを無効にします。

ヘッダー

[[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。

起原

[user-created|builtin|modified-builtin] 通知構成エントリの生成元。

シークレット

[[名前=<文字列> [,値=<文字列>]]]。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 エンコードされた秘密が含まれます。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。コンフィグを更新する際には、保持した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 通知非公開.cfg

B.7.1 ファイル形式

このファイルには、通知ターゲットの保護された認証情報が含まれています。

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)

必要なプロパティ:

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 (root:root 0600)で別の設定ファイルに保存されます。

オプションのプロパティ:

秘密

[[名前=<文字列> [, 値=<文字列>]]]。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 プルーン

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、その後にジョブ設定オプションが続きます。

```
プルーン: プルーンストア2
    スケジュール 月・金 10:30
    store my-datastore
prune: ...
```

このファイルを操作するには`proxmox-backup-manager prune-job`コマンドを使用します。

B.8.2 オプション

必要なプロパティ:

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。

店舗

[<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

無効にする

[<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します)。

ナノ秒

[<string>] 名前空間。

キープデイリー

[<integer> (1 - N)] 毎日保持するバックアップの数。

毎時

[<integer> (1 - N)] 1時間ごとに保持するバックアップの数。

キープラスト

[<integer> (1 - N)] 保存するバックアップの数。

毎月

[<integer> (1 - N)] 月間バックアップの保存数。

キープウィークリー

[<integer> (1 - N)] 毎週保持するバックアップの数。

ねんかん

[<integer> (1 - N)] 年間のバックアップ数。

B.9 tape.cfg

B.9.1 ファイル形式

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きます。

テープチェンジャーコンフィギュレーションは、ヘッダーの `changer: <name>` で始まり、その後にチェンジャーコンフィギュレーションオプションが続きます。

```
lto: hh8
  チェンジャーSL3
  パス /dev/tape/by-id/scsi-10WT065325-nst チェンジャー:
sl3
  輸出スロット14,15,16
  パス /dev/tape/by-id/scsi-CJ0JBE0059
```

このファイルを操作するには `proxmox-tape drive` と `proxmox-tape changer` コマンドを使用します。

注: 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「チェンジャーSCSI テープチェンジャー」

必要なプロパティ:

バス

[<string>] Linux汎用SCSIデバイスへのバス (例: '/dev/sg4')

オプションのプロパティ:

イジェクトビフォアアンロード

[<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。

輸出スロット

[[<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

必要なプロパティ:

パス

[<string>] LTO SCSI ジェネリック テープデバイスへのパス (例: '/dev/sg0')

オプションのプロパティ:

チェンジャー

[<string>] テープチェンジャー識別子。

チェンジャー・ドライブ番号

[<integer> (0 - 255 (default=0))] チェンジャー・ドライブ番号 (オプション・チェンジャーが必要)

セクションタイプ「仮想」: テープドライブをシミュレートします。

必要なプロパティ:

パス

[<string>] ディレクトリへのパス

オプションのプロパティ:

マックスサイズ

[<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、その後にジョブ設定オプションが続きます。

```
バックアップ: job1
  ドライブ hh8
  プール p4
  ストアストア3
  毎日のバックアップのス
ケジュール: ...
```

このファイルを操作するには proxmox-tape backup-job コマンドを使用します。

B.10.2 オプション

必要なプロパティ:

ドライブ

[<string>] ドライブ識別子。

プール

[<string>] メディアプール名。

店舗

[<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。

イジェクトメディア

[<boolean>] ジョブ完了時にメディアを排出します。

エクスポートメディアセット

[<boolean>] ジョブ完了時に設定されたメディア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]] を参照してください。 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)] 何階層の名前空間を操作するか (0 == 再帰なし)

通知モード

/legacy-sendmail|notification-system (default=legacy-sendmail)] このデータストアの通知をどのようにするかを設定します。 legacy-sendmail は、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に電子メール通知を送信します。

通知ユーザー

[<string>] ユーザーID

ナノ秒

[<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 ユーザー設定

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 user: <name> というヘッダで始まり、その後にユーザ設定オプションが続きます。

APIトークン設定は、ヘッダトークン: <userid!token_name>で始まり、トークン設定が続きます。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。

```
ユーザー: root@pam
コメント スーパーユーザー
emtest@example.local
令牌: root@pam!token1
コメント API テストトークン
有効にする
期限 0
ユーザー: ...
```

このファイルを操作するには proxmox-backup-manager ユーザ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

イネーブル

[<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。

絶え入る

[<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。

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

メール

[<string>] メールアドレス。

イネーブル

[<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。

絶え入る

[<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>] ファーストネーム。

ラストネーム

[<string>] 名字。

B.12 リモート.cfg

B.12.1 ファイル形式

このファイルには、リモートサーバー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、その後にリモート設定オプションが続きます。

```
リモート: server1
ホスト server1.local
リモート: ...
```

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

必要なプロパティ:

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 文字列として格納)。

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。

```
同期: job1
    ストアストア1
    リモートストア store1
同期: REMOTEリナ
同期: ...
```

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

必要なプロパティ:

リモートストア

[<string>] データストア名。

店舗

[<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を同期し、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>]]を参照してください。 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します)。

ナノ秒

[<string>] 名前空間。

所有者

[<string>] 認証ID

リモート

[<string>] リモートID。

リモートNS

[<string>] 名前空間。

リムーバブル

[<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。

再同期破損

[<boolean>] ローカルスナップショットの検証に失敗した場合、再度プルしてください。

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

[プル | プッシュ (デフォルト=プル)] 同期ジョブの方向、プッシュまたはプル

転送ラスト

[<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、他はスキップ

確認済みのみ

[<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、単位は任意 (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 ファイル形式**

各エントリは、ヘッダ検証: <name> で始まり、ジョブ設定オプションが続きます。

```
検証: verify-store2 ignore-verified
      true
      アフター7
      毎日のスケジュール
      ストアストア2
検証: ...
```

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

必要なプロパティ：

店舗

[<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)] すでに検証済みのバックアップの検証が古くない場合は検証しません。

最大深度

[<integer> (0 - 7) (default=7)] 何階層の名前空間を操作するか (0 == 再帰なし)

ナノ秒

[<string>] 名前空間。

時代遅れ

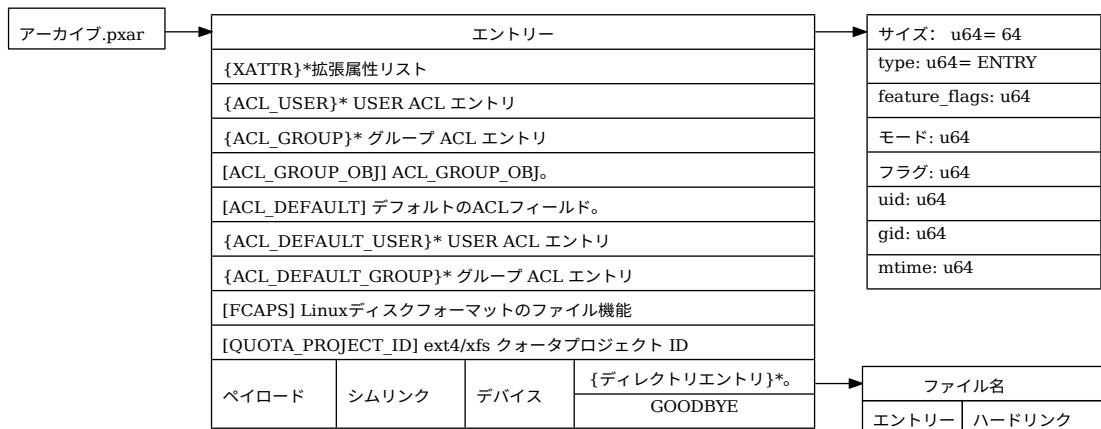
[<integer> (0 - N)] 検証が古くなる日数。 (0は非推奨)

スケジュール

[<calendar-event>] 指定したスケジュールでベリファイジョブを実行します。

ファイル形式

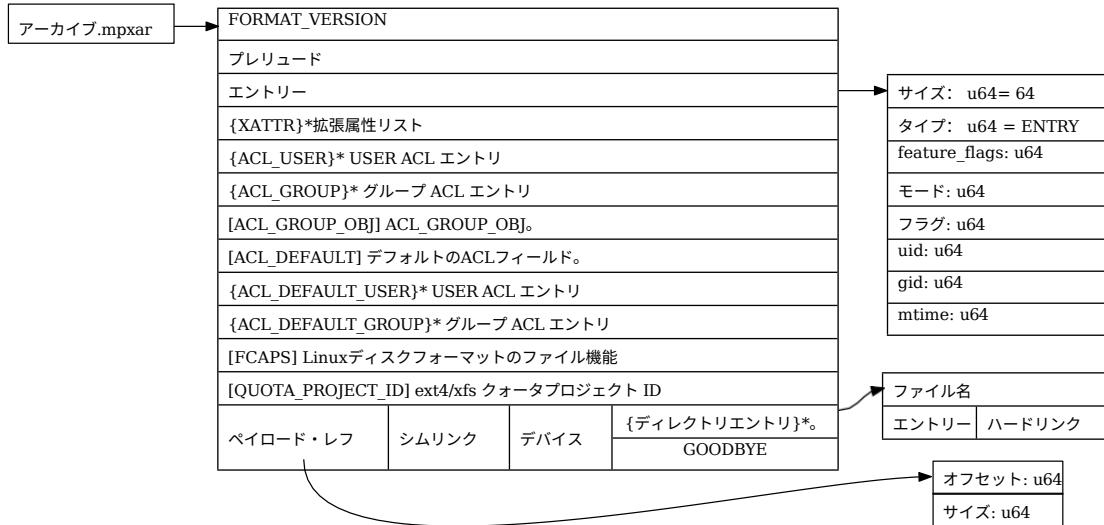
C.1 Proxmox ファイルアーカイブ形式 (.pxar)



C.2 Proxmox ファイルアーカイブ形式 - メタ (.mpxar)

通常の pxar アーカイブと同じ構造を持つ Pxar メタデータアーカイブ。ただし、レギュラ ファイル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参照として保存されます。

ファイルペイロードによってもたらされ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形式 - 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 (参照されない完全/部分的なペイロード) によって区切られている可能性があります。これは、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、チャンクの境界がペイロードエントリのオフセットに揃わなかったために発生します。

すべてのヘッダはリトルエンディアンで格納されます。

ペイロード開始マーク	u8;16] のヘッダはハッシュ型とサイズから構成されます。
ペイロード	u8; 16] の、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/完全な参照なしペイロード。
アーリィ	
...	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・テール・マーク	u8;16] のヘッダはハッシュ型とサイズからなります。

C.4 データブロブ形式 (.blob)

データブ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です。暗号化されていないblobは以下の形式を使

用します：

マジック: [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ター（IV）、16バイトの暗号化タグ（AE）、暗号化データが続きます：

マジック: [u8; 8]
CRC32: [u8; 4].
IV: [u8; 16].
TAG: [u8; 16]
データ: (最大16MiB)

C.5 固定インデックス形式(.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 以上 その インデックス (ヘッダなし ヘッダなし) SHA256(digest1 digest2 ...)
size: u64、	画像サイズ
chunk_size: u64、	チャンクサイズ
予約済みです: [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つ目のチャンクのダイジェスト
...	次のチャンク・ダイジェスト

C.6 ダイナミック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。

マジック: [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: [u8; 32],	SHA-256 オーバー の インデックス の ヘッダなし SHA256(offset1 digest1 offset2 digest2 ...)
予約済みです: [u8; 4032],	全体のヘッダーサイズは1ページ(4096バイト)
オフセット1: 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
オフセット2: u64	2つ目のチャンクの終わり
digest2: [u8; 32]	2つ目のチャンクのダイジェスト

...

次のチャンクのオフセット/ダイジェスト

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HTTP/2を使用してパフォーマンスを向上させます。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理解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
```

サーバーは 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）
- ・ 動的インデックス可変サイズ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。

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。次に、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ます。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.1.4 動的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の保存に使用されます。アーカイブデータは動的な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ます。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。これで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。

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することができます:

- 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（可変サイズ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](#)をご利用ください。

D.2.1 ダウンロード Blobs

プロブはGET /download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.2.2 ダウンロード Chunks

チャunk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でダウンロードされます。HTTPボディには
固定インデックスまたはダイナミック・インデックス。

カレンダーイベント

E.1 イントロダクションとフォーマット

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。

一般的な形式は以下の通りです：

リスト 1: カレンダー・イベント

平日] [[年-]月-日] [時:分[:秒]] [年-]月-日

少なくとも曜日、日付、時刻のいずれかの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、時刻は00:00:00を意味します。(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入れることができます：

- カンマ区切り：例：01,02,03
- を範囲として指定します
- 繰り返しとして：例：05/10（10回ごとに5から始まるという意味）
- および上記の組み合わせ：例：01,05...10,12/02
- または、可能な値すべてに * を付けます

特定の意味を持つ特別な値がいくつかあります：

価値	構文
ことこまかに	*-*-* *.*:00
毎時	*-*-* *:00:00
デイリー	*-*-* 00:00:00
ウィークリー	月-*-* 00:00:00
毎月	*-*-* 01 00:00:00
毎年または毎年	*-01-01 00:00:00
クオータリー	*-01,04,07,10-01 00:00:00
半年ごとまたは半期ごと	*-01,07-01 00:00:00

便利な例を表にまとめました:

例	オルタナティブ	説明
月、火、水、木	月・金	毎営業日00:00
土・日	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	月、火、水	月・火・水 30分後、40分後、50分後
*:30/10	*:30/10	毎時間
月・金	-	毎営業日、午前8時から午後6時までの間、15分ごと
8.17,22:0/15		午後10時から11時の間
12日 (金) 13:5/20	12,13日(金):5/20	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のタイムスタンプ (@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点を指定します。
- no timezone: すべてのスケジュールはサーバーのタイムゾーンを使用します。
- 秒以下の分解能なし
- 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 スケジュールに関する注意事項

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、期限が迫ってい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。

値下げ入門

「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文章を書き、それを構造的に妥当なXHTML（またはHTML）に変換することができます。」

ジョン・グルーバー <https://daringfireball.net/projects/markdown/>

Proxmox Backup Serverウェブインターフェースの "ノート" 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 マークダウンの基本

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

```
# This is 見出し h1 ## This is 見  
出し h2  
##### これは見出します。
```

F.1.2 強調

強調したい場合は、*text* または _text_ を使用してください。

太字のテキストには **text** または __text__ を使用します

"あなたはそれを組み合わせることができます

。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/> はクリック可能なリンクに変換します。

、リンクテキストをコントロールすることもできます：

では、[カッコ内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：

- * 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。

注文リスト

1. 項目 1
1. 項目 2
1. 項目 3
 1. 項目 3a
 1. 項目 3b

注：順番に並べられたリストの整数は正確である必要はありません。

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

- [X] 最初のタスクはすでに完了しました!
- [X] 2つ目も
- [] これはまだto-do
- [これも

。例えば

F.1.5 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、右揃えにすることもできます。

左カラム 右カラム 一部 その他 Cols. センタリングも有効	----- -----: ----- ----- :-----				
左フー	右フー	最初の 行 ここ >center<			
左バー	右バー	番目 行 ここ 12345			
レフトバズ	ライトバズ	番目 行 こちら テスト			
レフトザブ	右ザブ	第4 行 ここ ^--			
左狂犬病	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.1.6 ブロック名言集

ブロック引用符は、プレーンテキストの電子メールと同様に、行の先頭に> を付けることで入力できます。

>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、ジョン・グルーバーがアーロン・スワーツとともに作成。
>
>>Markdownは、Readmeファイルのフォーマットや、オンラインディスカッションのメッセージによく使用されます。■
'→フォーマム
>>また、プレーンテキストエディタを使ってリッチテキストを作成することもできます。

F.1.7 コードとスニペット

バックステイックを使用すると、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。

オンラインコード

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

複数の行にまたがるコード・ブロックの場合、例えば、トリプル・バックティックを使ってそのようなブロックを開始したり終了したりすることができます:

```
```  
ここで覚えておきたいネットワーク設定 auto vmbr2

iface vmbr2 inet static
 アドレス 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。

さび

Rustは、高速でメモリ効率の高い新しいシステム・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。

スフィンクス

インテリジェントできれいな書式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テーション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作成のための優れた機能を備えています。

再構造化テキスト

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。

ヒューズ

Filesystem in Userspace (FUSE) は、カーネルでファイルシステムを実装するのとは対照的に、ユーザ空間でファイルシステムを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インストールとユーザの認証情報。バックアップを冗長化するために、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できます。

gnu自由文書ライセンス

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/>>

すべての人は、本書の逐語的なコピーをコピーし、配布することが許可されています。

ライセンス文書を変更することは。

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社に、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保持します。

このライセンスは一種の「コピーレフト」であり、この文書の派生物自体も同じ意味で自由でなければならないことを意味します。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: 自由なプログラムには、同じ自由を提供するマニュアルが付属しているべきです。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否かに関わらず、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、この利用許諾契約書を、主に指導や参照を目的とした作品に推奨します。

1. 適用と定義

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ている、媒体を問わずマニュアルやその他の著作物に適用されます。このような告知は、本許諾書に記載された条件の下で、その作品を使用するための、世界的な、期間無制限の、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。あなたが著作権法上の許可を必要とする方法で作品を複製、変更、頒布する場合、あなたはこのライセンスに同意するものとします。

本件文書の「改変版」とは、本件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的、商業的、哲学的、倫理的、政治的な立場の問題である可能性があります。

変更不可セクション」とは、『文書』が本許諾書の下でリリースされたことを示す告知において、そのタイトルが変更不可セクションであるとして指定されている特定のセカンダリ・セクションのことです。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認められません。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。

表紙文」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、表紙文または裏表紙文として列挙される、ある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、機械可読なコピーであって、その仕様が一般公衆に利用可能なフォーマットで表現されており、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や(画面については)広く利用可能な何らかの画面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フォーマットへの自動翻訳に適しているものを意味します。そうでなければ透明なファイルフォーマットで作成されたコピーで、マークアップがある、あるいはマークアップがないものが、読者によるその後の変更を妨げたり、阻止したりするように配置されているものは、透明ではありません。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。透明ではないコピーは「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、一般に利用可能なDTDを使用したSGMLまたは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。トランスペアレント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、独自のワープロでのみ読み取りや編集が可能な独自のフォーマット、DTDや処理ツールが一般に公開されていないSGMLやXML、一部のワープロが出力のみを目的として生成する機械生成のHTMLやPostScript、PDFなどがあります。

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資料を読みやすく掲載するために必要な次のページを加えたものを意味します。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、タイトルが正確にXYZであるか、またはXYZを他の翻訳したテキストの後に括弧で括られたXYZを含む、『文書』の名前付きサブユニットを意味します。(ここでXYZは、後述の特定のセクション名、例えば"Acknowledgements"、"Dedication"、"Endorsements"、"History"を意味します)。こ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、ドキュメントを変更したときに、そのセクションがこの定義に従って "XYZというタイトル"の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してのみです。

2. まるうつし

あなたは、本許諾書、著作権表示、および本許諾書が『文書』に適用される旨の使用許諾表示をすべての複製物に複製し、本許諾書の条件に他のいかなる条件も加えないことを条件に、『文書』を商業的であるか非商業的であるかを問わず、あらゆる媒体に複製し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更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項の条件にも従わなければなりません。

また、上記と同じ条件でコピーを貸与し、コピーを公に展示することもできます。

3. 量り売り

あなたが『文書』の印刷された複製物(また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』のライセンス告知でカバー・テキストが要求されている場合、あなたはその複製物を、以下の全てのカバー・テキスト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：「Front-Cover Texts」、紙には「Back-Cover Texts」。また、両表紙とも、あなたがこれら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、他の素材を加えてもかまい

ません。文書の題名を保持し、これらの条件を満たす限り、表紙に限定して変更を加えた複製は、その他の点では逐語的複製として扱うことができます。

どちらの表紙にも必要な文章が多すぎて読みにくい場合は、最初に挙げたもの（無理のない範囲で）を実際の表紙に載せ、残りは隣のページに続けて載せます。

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、一般的な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。後者の選択肢を用いる場合、あなたは『不透明な複製物』の配布を大量に開始する際に、その『透明な複製物』が、あなたがその版の『不透明な複製物』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年経過するまで、指定された場所でアクセス可能であり続けるよう、合理的かつ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』の作成者によく連絡し、『文書』の更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. 変更点

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。加えて、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（および表紙がある場合は表紙）には、『文書』のタイトルおよび旧版のタイトル（旧版が場合は、『文書』の「歴史」の項に記載されているはずです）とは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、改変された版における改変の著者に責任を負う一人以上の人または団体を、その文書における主要な著者のうち少なくとも5人(主要な著者の数が5人未満である場合は全員)とともに、著者として記載してください。
- C. タイトルページには、発行者として修正版の発行者名を明記してください。
- D. 文書のすべての著作権表示を保存してください。
- 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、以下の補遺に示す形式で、本許諾書の条項の下で改変された版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. そのライセンス通知には、『文書』のライセンス通知で指定されている、変更不可のセクションと必要なカバーテキストの完全なリストを記載してください。
- H. 本使用許諾の変更されていないコピーを同封してください。
- I. "History"と題されたセクションを保存し、そのタイトルを保存し、少なくともタイトルページに記載されている修正版のタイトル、年、新しい著者、出版社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「歴史」と題されたセクションがない場合、『文書』のタイトルページに記載されている『文書』のタイトル、年、著者、出版社を記載したセクションを作成し、前文にあるように、修正版について記述した項目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』自身よりも少なくとも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版発行者が許可した場合、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、そこに記載された各寄稿者の謝辞および/または献辞の内容およびトーンをすべて保持します。

- L. 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. エンドースメント」と題された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、修正版には含まれません。
- 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、不变のセクションのタイトルと矛盾させたりしないでください。
- O. 保証の免責事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区別されなければなりません。

エンドースメント」というセクションを追加することができますが、そのセクションには、様々な当事者によるあなたの修正版に対するエンドースメント（例えば、査読の記述や、ある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等）だけを含めることができます。

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のそれぞれ1つの文章のみを、1つの団体によって（または団体による取り決めによって）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。

本許諾書によって、『文書』の著者および発行者は、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『文書』の支持を表明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、改変されたバージョンについて上記第4項で定義された条件の下で、『文書』を結合することができます。ただし、その結合国家に、改変されていない元の文書のすべての変更不可部分を含め、そのライセンス告知にあなたの結合著作物の変更不可部分としてそれらすべてを記載し、それらのすべての保証の否認を保持することを条件とします。

結合された著作物には本許諾書が一部含まれていればよく、複数の同一の変更不可部分が一つのコピーで置き換えられてもよい。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、そのような各「変更不可部分」のタイトルの末尾に、その「変更不可部分」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書きで追加することによって、その「変更不可部分」を一意なものとしてください。結合著作物の利用許諾の告知にある、変更されないセクションの一覧のセクションタイトルにも、同じ調整を加えてください。

同様に、"Acknowledgements"（謝辞）と題されたセクションと "Dedications"（献辞）と題されたセクションもすべて組み合わせてください。裏書き」と題されたセクションはすべて削除してください。

6. 文献集

あなたは、『文書』と本許諾書に基づいてリリースされた他の文書から成る文書集を作成し、様々な文書に含まれる本許諾書の個々のコピーを、その文書集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、本許諾書の下で個別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、その文書の逐語的な複製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. 独立作品とのアグリゲーション

以下の場合、『文書』またはその派生物と、他の個別かつ独立した文書または著作物とを、記憶媒体または頒布媒体の一巻にまとめたものを「集合体」と呼びます。

編集物から生じる『文書』の著作権は、編集物の利用者の法的権利を、個々の著作物が許容する範囲を超えて制限するために使用されるものではない。『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、それ自身が『文書』の二次的著作物で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。変更不可部分を翻訳に置き換えるにはその著作権者の特別な許可が必要ですが、あなたは変更不可部分の一部または全部の翻訳を、変更不可部分の原版に加えて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。その場合、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ものとします。翻訳版と本許諾書の原版、あるいは通知や免責事項の原版との間に不一致がある場合、原版が優先されます。

文書の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（セクション1）を保持するための再要件（セクション4）では、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試みは無効であり、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すべての違反を停止した場合、特定の著作権者からのあなたのライセンスは、(a)著作権者が明示的かつ最終的にあなたのライセンスを終了させない限り、暫定的に、(b)著作権者が違反停止後60日以前に何らかの合理的な手段であなたに違反を通知しなかった場合、永続的に復活します。

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、あなたがその著作者から本許諾書に対する違反の通知（著作物を問わず）を初めて受け取り、あなたがその通知を受け取ってから30日よりも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久に復活します。

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コピーの受領は、あなたにそれを使用する権利を与えるではありません。

10. 本ライセンスの将来の改訂

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発表することができます。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似ていますが、新たな問題や懸念に対処するために細部において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、識別可能なバージョン番号が付されています。『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（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と指定されている場合、その代理人がある版を受諾すると公言することで、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. リライセンシング

「Massive Multiauthor Collaboration Site」（または「MMCサイト」）とは、著作権のある著作物を公開し、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・サーバーを意味します。

の作品です。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。このサイトに含まれる「Massive Multi-author Collaboration」(あるいは「MMC」)とは、このようにしてMMCのサイトで公開された、著作権で保護されるべき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であるクリエイティブ・コモンズ・コーポレーション (Creative Commons Corporation) が発行するクリエイティブ・コモンズ表示-継承3.0ライセンス (Creative Commons Attribution-Share Alike 3.0) 、および同団体が発行する同ライセンスの将来のコピーレフト版を意味します。

「組込」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は、それが本許諾書の下でライセンスされ、かつ、本許諾書の下で本MMC以外のどこかで最初に公表され、その後にそのMMCに全部また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、「再ライセンスの対象」となります。

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のもと、同サイトに含まれる再公開することができます。

インデックス

C

コンテナ、**285**

D

データストア、**285**

F

ヒューズ、**285**

R

リモート、**285**

reStructuredText、**285**

サビ、**285**

S

スフィンクス、**285**

V

仮想マシン、**285**