



# Proxmoxバックアップドキュメント

リリース 3.2.7-1

Proxmoxサポートチーム

2024年7月3日（水）



# 目次

<b>1</b>	<b>はじめに</b>	<b>3</b>
1.	1Proxmoxバックアップサーバとは何ですか?	3
1.	2アーキテクチャ	3
1.	3主な特徴	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.	1Enterprise Support	5
1.	2Community Support Forum	5
1.	1.6.3 メーリングリスト	6
1.	1.6.4バグトラッカー	6
1.	7ライセンス	6
1.	8History	6
<b>2</b>	<b>インストール</b>	<b>7</b>
2.1	システム要件	7
2.1.1	評価のための	7
	最小サーバー要件	7
2.1.	2Recommended Server System Requirements	7
2.1.	3Supported Web Browsers for Accessing the Web Interface	8
2.	2Debian パッケージリポジトリ	8
	2.2.1 セキュアアパート	9
	2.2.2 Proxmox Backup Enterprise リポジトリ	9
2.2.	3Proxmox バックアップ 無サブスクリプション リポジトリ	9
	2.2.3 Proxmox バックアップテストリポジトリ	10
	2.2.4 Proxmox バックアップ クライアント専用リポジトリ	10
2.2.	6Repository Access Behind HTTP Proxy	11
2.	3Server Installation	11
	2.3.1インストーラを使用したProxmox Backup Serverのインストール	11
	2.3.2Install Proxmox Backup Server Unattended	12
	2.3.3Install Proxmox Backup Server on Debian	12
	2.3.4ProxmoxVEにProxmox Backup Serverをインストールします。	12
2.4	2.4 クライアントのインストール	13
	2.4.1Install Proxmox Backup Client on Debian	13
<b>3</b>	<b>用語解説</b>	<b>15</b>
3.	1Backup Content	15
	3.1.1 画像アーカイブ: <名前>.img	15
	3.1.2 ファイルアーカイブ: <名前>.pxar	15
	3.1.3 バイナリデータ (BLOB)	15
	3.1.4 カタログファイル: catalog.pcat1	16
	3.1.5マニフェスト: index.json	16
3.2	3.2 バックアップネームスペース	16
3.	3Backup Type	16
3.	4Backup ID	16

3.5 バックアップ時間.....	16
3.6 バックアップ・グループ.....	17
3.7 バックアップ・スナップショット.....	17
<b>4 グラフィカル・ユーザー・インターフェース <b>19</b></b>	
4.1 特徴.....	19
4.2 ログイン.....	19
4.3 GUIの概要.....	20
4.4 サイドバー.....	21
4.4.1 ダッシュボード.....	21
4.4.2 構成.....	21
4.4.3 管理部門.....	22
4.4.4 テープバックアップ.....	24
4.4.5 データストア.....	25
<b>5 バックアップ・ストレージ <b>27</b></b>	
5.1 ディスク管理.....	27
5.2 データストア.....	29
5.2.1 データストアの構成.....	30
5.2.2 バックアップ名前空間.....	32
5.2.3 オプション.....	33
5.3 ランサムウェアの保護と復旧.....	34
5.3.1 内蔵プロテクション.....	34
5.3.2 Proxmox Backup Serverの3-2-1ルール.....	35
5.3.3 制限的なユーザーとアクセス管理.....	35
5.3.4 ランサムウェアの検出.....	35
5.3.5 一般的な予防法とベストプラクティス.....	36
<b>6 ユーザー管理 <b>37</b></b>	
6.1 ユーザー設定.....	37
6.2 APIトークン.....	39
6.3 アクセス制御.....	40
6.3.1 特典.....	40
6.3.2 アクセスの役割.....	42
6.3.3 オブジェクトとパス.....	43
6.3.4 設定と管理.....	43
6.3.5 APIトークンのパーミッション.....	44
6.3.6 効果的なパーミッション.....	44
6.4 二要素認証.....	45

6.4.1	はじめに	45
6.4.2	利用可能なセカンドファクター	45
6.4.3	設定	45
6.4.4	TFAと自動アクセス	47
6.4.5	二要素認証の制限とロックアウト	47
6.5	認証領域	47
6.5.1	LDAP	47
6.5.2	アクティブディレクトリ	48
<b>7</b>	<b>バックアップクライアントの使用状況</b>	<b>49</b>
7.1	バックアップ・リポジトリの場所	49
7.2	環境変数	50
7.3	出力フォーマット	50
7.4	バックアップの作成	51
7.4.1	バックアップからのファイル/ディレクトリの除外	52
7.4.2	変更検出モード	53
7.5	暗号化	54
7.5.1	マスター・キーの使用による暗号化キーの保存と復元	54
7.6	データの復元	56
7.6.1	インタラクティブ・リストア	56
7.6.2	FUSEによるアーカイブのマウント	57
7.7	ログインとログアウト	57
7.8	バックアップ・グループの所有者の変更	58
7.9	バックアップの剪定と削除	58
7.10	ゴミ収集	59
7.11	ベンチマーкиング	60
<b>8</b>	<b>Proxmox VEの統合</b>	<b>63</b>
8.1	Proxmox VEウェブインターフェースの使用	63
8.2	Proxmox VEコマンドラインの使用	63
<b>9</b>	<b>pxarコマンドラインツール</b>	<b>65</b>
9.1	アーカイブの作成	65
9.2	アーカイブの抽出	66
9.3	アーカイブの内容をリストアップ	66
9.4	アーカイブのマウント	66
<b>10</b>	<b>テープバックアップ</b>	<b>67</b>
10.1	テープ技術入門	68
10.2	対応ハードウェア	68
10.2.1	駆動性能	68

10.2.2 LTO-9+の考察	69
10.3 用語解説	69
10.4 テープクイックスタート	70
10.5 構成	70
10.5.1 テープチェンジャー	70
10.5.2 テープドライブ	73
10.5.3 メディア・プール	75
10.5.4 テープバックアップの仕事	78
10.6 管理部門	80
10.6.1 ラベルテープ	80
10.6.2 テープバックアップの実行	81
10.6.3 テープからのリストア	81
10.6.4 在庫の更新	82
10.6.5 カタログの復元	82
10.6.6 暗号鍵の管理	83
10.6.7 テープクリーニング	84
10.7 WORMテープ	85
10.8 セットアップ例	85
10.8.1 シングル継続メディア・セット	85
10.8.2 平日スキーム	85
10.8.3 異なるポリシーを持つ複数のプール	86
<b>11 リモコンと同期の管理</b>	<b>87</b>
11.1 リモート	87
11.2 シンク・ジョブズ	88
11.2.1 名前空間のサポート	89
11.2.2 帯域制限	90
<b>12 メンテナンスタスク</b>	<b>91</b>
12.1 剪定	91
12.1.1 ブルーンシミュレータ	92
12.1.2 ブルーンの求人	92
12.1.3 手動剪定	93
12.1.4 保持設定例	93
12.2 ゴミ収集	94
12.2.1 GCの背景	94
12.2.2 GCの手動起動	95
12.2.3 GC予定	95

12.3 検証	95
12.4 お知らせ	96
12.5 メンテナンスマード	96
<b>13 ホストシステム管理</b>	<b>97</b>
13.1 Linux上のZFS	97
13.1.1 ハードウェア	98
13.1.2 ZFS管理	98
13.2 ホストブートローダー	103
13.2.1 インストーラーが使用するパーティション方式	103
13.2.2 <code>proxmox-boot-tool</code> を使ったESPの内容の同期	104
13.2.3 使用されるブートローダーの決定	106
13.2.4 グラブ	108
13.2.5 システムブート	108
13.2.6 カーネル・コマンドラインの編集	108
13.2.7 次回ブート時のカーネルバージョンの上書き	109
13.2.8 セキュアブート	110
13.3 証明書管理	112
13.3.1 APIとSMTPの証明書	112
13.3.2 カスタム証明書のアップロード	113
13.3.3 Let's Encrypt (ACME)による信頼できる証明書	113
13.3.4 ACME HTTPチャレンジプラグイン	115
13.3.5 ACME DNS APIチャレンジプラグイン	115
13.3.6 ACME証明書の自動更新	116
13.3.7 コマンドラインでの証明書の手動変更	116
13.4 サービスデーモン	117
13.4.1 プロックスモックスバックアッププロキシ	117
13.4.2 プロックスモックスバックアップ	117
13.5 コマンドラインツール	117
13.5.1 <code>proxmox-backup-client</code>	117
13.5.2 <code>proxmox-backup-manager</code>	117
13.5.3 プロックスモックスステープ	117
13.5.4 <code>pmt</code>	118
13.5.5 <code>pmtx</code>	118
13.5.6 <code>pxar</code>	118
13.5.7 プロックスファイルリストア	120
13.5.8 <code>proxmox-backup-debug</code>	120
<b>14 ネットワーク管理</b>	<b>121</b>
14.1 交通管制	123
<b>15 お知らせ</b>	<b>125</b>

15.1	概要	125
15.2	通知目標	125
15.2.1	センドメール	125
15.2.2	SMTP	126
15.2.3	ゴティファイ	126
15.3	通知マッチャー	126
15.3.1	カレンダーのマッチングルール	126
15.3.2	フィールド・マッチング・ルール	127
15.3.3	深刻度マッチングルール	127
15.4	通知イベント	127
15.5	システムメール転送	128
15.6	アクセス許可	128
15.7	通知モード	128
15.7.1	レガシー送信メール通知モードの設定	129
<b>16</b>	<b>技術概要</b>	<b>131</b>
16.1	データストア	131
16.2	スナップ写真	131
16.3	チャンクス	131
16.3.1	固定サイズのチャンク	132
16.3.2	動的なサイズのチャンク	132
16.3.3	暗号化されたチャンク	133
16.4	注意点と限界	133
16.4.1	ハッシュ衝突に関するメモ	133
16.4.2	ファイルベースのバックアップ	133
16.4.3	暗号化されたチャンクの検証	134
16.5	トラブルシューティング	134
16.5.1	Proxmox Backup Serverが稼働していない状態でのリストア	134
<b>17</b>	<b>よくあるご質問</b>	<b>135</b>
17.1	Proxmox Backup Server (PBS)はどのディストリビューションに基づいていますか?	135
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?	135
17.3	Proxmox Backup Serverは32ビットプロセッサで動作しますか?	135
17.4	Proxmox Backup Serverバージョンのサポート期間は?	135
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。	135
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。	136
17.7	データストアを別の場所にコピーまたは同期できますか?	136
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?	136
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?	136

17.10 バックアップは増分/重複/フルですか?	137
<b>A コマンド構文</b>	<b>139</b>
A.1 proxmox-backup-client	139
A.1.1 カタログシェルコマンド	139
A.2 proxmox-backup-manager	139
A.3 プロックスモックスステープ	139
A.4 pmt	139
A.5 pmtx	140
A.6 pxar	140
A.7 proxmox- ファイルリストア	140
A.8 proxmox-backup-debug	140
<b>B 設定ファイル</b>	<b>141</b>
B.1 acl.cfg	141
B.1.1 ファイル形式	141
B.1.2 役割	141
B.2 datastore.cfg	141
B.2.1 ファイル形式	141
B.2.2 オプション	142
B.3 ドメイン.cfg	142
B.3.1 ファイル形式	142
B.3.2 オプション	142
B.4 メディアプール.cfg	142
B.4.1 ファイル形式	142
B.4.2 オプション	143
B.5 通知.cfg	143
B.5.1 ファイル形式	143
B.5.2 オプション	143
B.6 通知非公開.cfg	143
B.6.1 ファイル形式	143
B.6.2 オプション	143
B.7 tape.cfg	143
B.7.1 ファイル形式	143
B.7.2 オプション	143
B.8 tape-job.cfg	143
B.8.1 ファイル形式	143

B.8.2 オプション .....	.144
B.9. 1File Format .....	.144
B.9.2 オプション .....	.144
B.10 remote.cfg .....	.144
B.10.1 ファイルフォーマット .....	.144
B.10.2 オプション .....	.145
B.11 sync.cfg .....	.145
B.11.1 ファイルフォーマット .....	.145
B.11.2 オプション .....	.145
B.12 verification.cfg .....	.145
B.12.1 ファイルフォーマット .....	.145
B.12.2 オプション .....	.145
<b>Cファイルフォーマット</b>	<b>147</b>
C. 1Proxmox ファイルアーカイブ形式 (.pxar) .....	.147
C. 2Proxmox ファイルアーカイブ形式 - メタ (.mpxar) .....	.147
C. 3Proxmox ファイルアーカイブ形式 - ペイロード (.ppxar) .....	.148
C.4 データ・ブロブ・フォーマット (.blob) .....	.148
C.5 固定インデックス形式 (.fidx) .....	.149
C.6 ダイナミック・インデックス・フォーマット (.didx) .....	.149
<b>Dバックアップ・プロトコル</b>	<b>151</b>
D.1 バックアッププロトコル API .....	.151
D.1.1 アップ ロード Blob .....	.151
D.1.2 チャンクの アップロード .....	.151
D.1.3 固定インデックスの アップロード .....	.152
D.1.4 動的インデックスの アップロード .....	.152
D.1.5 バックアップを 終了します。 .....	.152
D.2 リストア /リーダー・プロトコル API .....	.152
D.2.1 ブロブの ダウンロード .....	.153
D.2. 2ダウンロードチャンク .....	.153
D.2. 3D インデックスファイルの ダウンロード .....	.153
<b>Eカレンダー イベント</b>	<b>155</b>
E. 1Introduction and Format .....	.155
E. 2systemdとの 相違点 .....	.156
E. 3Notes on Scheduling .....	.156
<b>Fマークダウン入門</b>	<b>157</b>
F. 1Markdown Basics .....	.157
F.1. 2Emphasis .....	.157
F.1.1 見出し .....	.157
F.1.3 リンク .....	.158
F.1.4 リスト .....	.158
F.1.5 表 .....	.158
F.1.6 ブロック引用 .....	.159
F.1.7 コードとスニペット .....	.159
<b>G用語集</b>	<b>161</b>
<b>H GNU Free Documentation License</b>	<b>163</b>
<b>索引</b>	<b>169</b>

Copyright (C) 2019-2023, Proxmox Server Solutions GmbH

Version 3.2.7 -- 2024年07月03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あって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るため、完全に信頼されていないターゲットへのデータのバックアップをより安全に行うことができます。

### 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。

バックアップクライアントはこのAPIを使用してバックアップされたデータにアクセスします。`proxmox-backup-client`コマンドラインツールを使用してファイル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 に対して、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必要がなく、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簡単になります。

## 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データを生成します。重複排除レイヤーは冗長性を回避し、使用するストレージ容量を最小限に抑えます。

## 増分バックアップ

バックアップ間の変更は通常、少ないものです。差分のみを読み込んで送信することで、バックアップ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発揮します。クライアント側の暗号化に加え、すべてのデータは安全なTLS接続を介して転送されます。

## テープバックアップ

Proxmox Backup Serverは、データの長期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、必要に応じて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徹底してください。

## 1.5 ソフトウェアスタック

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

- RESTful API、超高速非同期タスク、軽量な使用統計収集、スケジューリングイベント、特権環境と非特権実行環境の厳格な分離などを提供するサーバーデーモン。
- JavaScript管理ウェブインターフェース
- サーバーの管理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*)

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

"Rust"プログラミング言語は、より高速で信頼性の高いソフトウェアを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します。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なし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3年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クアントについては、[sales@proxmox.com](mailto: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世界中に広がっています。このような大規模なフォーラムが情報を得るのに最適な場所であることは言うまでもありません。

### 1.6.3 メーリングリスト

Proxmox Backup Serverは完全にオープンソースであり、貢献を歓迎します！開発者との主な連絡手段はこれらです：

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](https://lists.proxmox.com/listinfo/proxmox-backup-dev)

### 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) 2019-2023, Proxmox Server Solutions GmbH

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

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

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。このツールはvzdumpと呼ばれ、実行中のLXCコンテナやKVM仮想マシンの一貫したスナップショットを作成することができます。

しかし、vzdump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

## **Proxmox Backup ドキュメント、リリース 3.2.7-1**

---

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、高速性とメモリ効率を実現するために、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を本番環境で実行する場合は、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す）。
- バックアップストレージ:
    - ランダムIOワークロードに高い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を使用する場合: メタデータキャッシュを使用することを強く推奨します。例えば、ZFS [特殊デバイスマラー](#)を追加します。
  - 冗長マルチビット/秒ネットワーク・インターフェイス・カード(NIC)

## 2.1.3 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 Debian パッケージリポジトリ

すべての Debian ベースのシステムでは、パッケージ管理ツールとして APT を使用しています。リポジトリのリストは /etc/apt/sources.list と /etc/apt/sources.d/ ディレクトリにある .list ファイルで定義されます。アップデートは apt コマンドラインツール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。

## Proxmox Backup ドキュメント、リリース 3.2.7-1

The screenshot shows the Proxmox Backup Server 2.2-0 interface. The left sidebar has a tree view with nodes like Dashboard, Notes, Configuration, Remotes, Traffic Control, Certificates, Subscription, Administration (selected), Shell, Storage / Disks, Tape Backup, and Datastore. The main area is titled "Server Administration" with tabs for ServerStatus, Services, Updates, Repositories (selected), Syslog, and Tasks. Under "Status", it says "All OK, you have production-ready repositories configured!" with a green checkmark icon. The "APT Repositories" section has tabs for Reload, Add, and Enable. It lists repositories under "/etc/apt/sources.list" (5 repositories) and "/etc/apt/sources.list.d/pbs-enterprise.list" (1 repository). The table columns are Enabled, Types, URIs, Suites, Components, Options, Origin, and Comment. The "pbs-test" repository in the first row has a yellow warning icon in the Options column.

Enabled	Types	URIs	Suites	Components	Options	Origin	Comment
<input checked="" type="checkbox"/>	deb	http://ftp.at.debian.org/debian	bullseye	main contrib	<span style="color: red;">?</span>	Debian	
<input checked="" type="checkbox"/>	deb	http://ftp.at.debian.org/debian	bullseye-updates	main contrib	<span style="color: red;">?</span>	Debian	
<input checked="" type="checkbox"/>	deb	http://deb.debian.org/debian	bullseye-backports	main contrib	<span style="color: red;">?</span>	Debian Ba...	
<input checked="" type="checkbox"/>	deb	http://security.debian.org	bullseye-security	main contrib	<span style="color: red;">?</span>	Debian	security updates
<input type="checkbox"/>	deb	http://download.proxmox.com/debian/pbs	bullseye	pbs-test <span style="color: yellow;">!</span>		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	

## 2.2.1 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
7da6fe34168adc6e479327ba517796d4702fa2f8b4f0a9833f5ea6e6b48f6507a6da403a274fe201595edc86a84463d5
→ /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
```

0383d07f64bdd

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
```

## 2.2.2 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、本番環境での使用に適しています。

`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履歴と詳細は 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。

### 2.2.3 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に使用できます。これらのパッケージは常に厳重にテストされ検証されているわけではないので、本番サーバーで使用することは推奨されません。

このリポジトリは `/etc/apt/sources.list` で設定することをお勧めします。

### 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ていません
deb http://download.proxmox.com/debian/pbs bookworm pbs-no-subscription
# セキュリティ・アップデート
deb http://security.debian.org/debian-security bookworm-security main contrib
```

### 2.2.4 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
```

### 2.2.5 Proxmox Backup クライアント専用リポジトリ

*Proxmox* 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 *Proxmox Backup Client* を使用する場合は、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の派生版のように、*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ベースのリポジトリを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。その後、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
```

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

デブ <http://download.proxmox.com/debian/pbs-client> ブルズアイ・メイン

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しれませんし、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

### 2.2.6 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、中央プロキシ経由のみで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 ネットワーク要求と Proxmox Backup Server サポートサブスクリプションの確認に使用されます。

標準的なHTTPプロキシ設定が受け入れられます。`[http://]<host>[:port]` ここで、`<host>` の部分には認証が含まれることがあります。

## 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。

---

Proxmox が提供するディスクイメージ (ISO ファイル) には、完全な Debian システムと Proxmox Backup Server に必要なすべてのパッケージが含まれています。

インストーラがセットアッププロセスを案内し、ローカルディスクのパーティション設定、基本的なシステム設定 (タイムゾーン、言語、ネットワークなど) の適用、必要なパッケージのインストールを行います。提供されている ISO を使えば、ほんの数分で使い始めることができます。新規ユーザーにも既存ユーザーにもお勧めの方法です。

あるいは、Proxmox Backup Server を既存の Debian システムの上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: インストールプロセスでは、デフォルトで完全なサーバーが使用され、既存のデータはすべて削除されます。

---

### 2.3.2 Proxmox Backup Serverの無人インストール

pve}を無人で自動インストールすることが可能です。これにより、ベアメタル上でのセットアッププロセスを完全に自動化できます。インストールが完了し、ホストが起動したら、Ansible のような自動化ツールを使用してインストールをさらに設定できます。

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一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次第です。

**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ーに障害が発生しても、バックアップにアクセスできます。

注意: 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
```

注意: クライアント専用のリポジトリは、最近の Debian と Ubuntu の派生版で使えるはずです。



## 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](#)は両方の戦略を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データに使用されます。コンテンツは固定サイズのチャンクに分割されます。

#### 3.1.2 ファイルアーカイブ:<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

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、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。

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、ファイルホスト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。

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RFC3339の協定世界時（UTC、末尾のZで識別）です。



## グラフィカルユーザーインターフェース

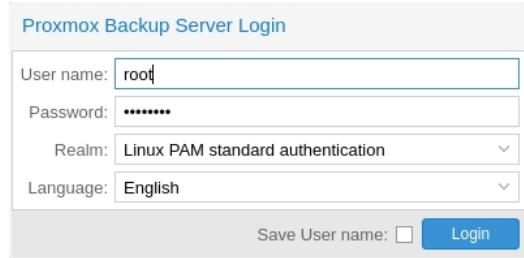
Proxmox Backup Serverは、サーバを管理するための統合されたWebベースのインターフェイスを提供します。つまり、ウェブブラウザですべての管理タスクを実行でき、追加の管理ツールをインストールする必要がありません。ウェブインターフェースはビルトインコンソールも提供するため、コマンドラインがお好みの場合や、より高度な制御が必要な場合は、このオプションをご利用ください。

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



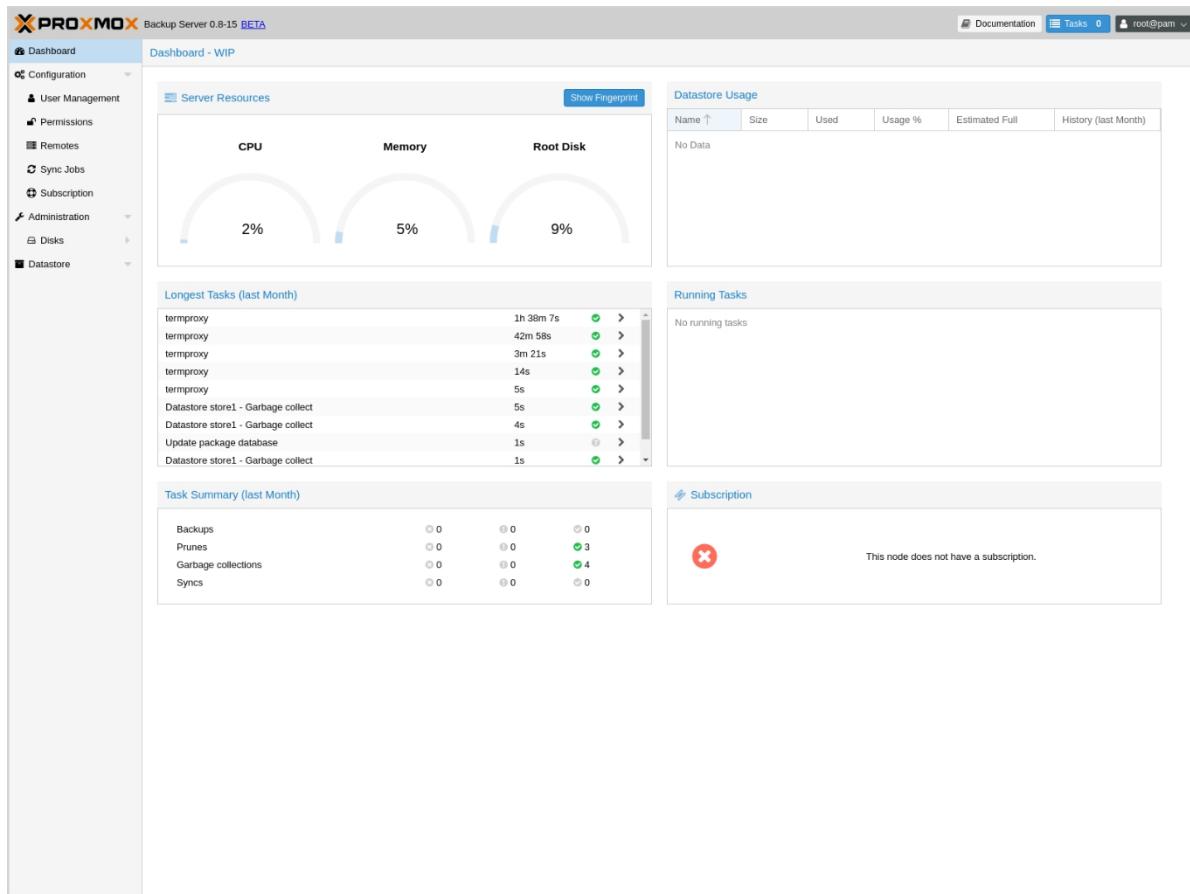
ウェブインターフェースに接続すると、最初にログインウィンドウが表示されます。Proxmox Backup Serverは様々な言語と認証バックエンド(*Realms*)をサポートしています。

---

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ー名を保存できます。

---

## 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、各種表示設定、ログアウトなどのボタンがあります。
- サイドバー：左側。サーバーの管理オプションが含まれています。
- 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

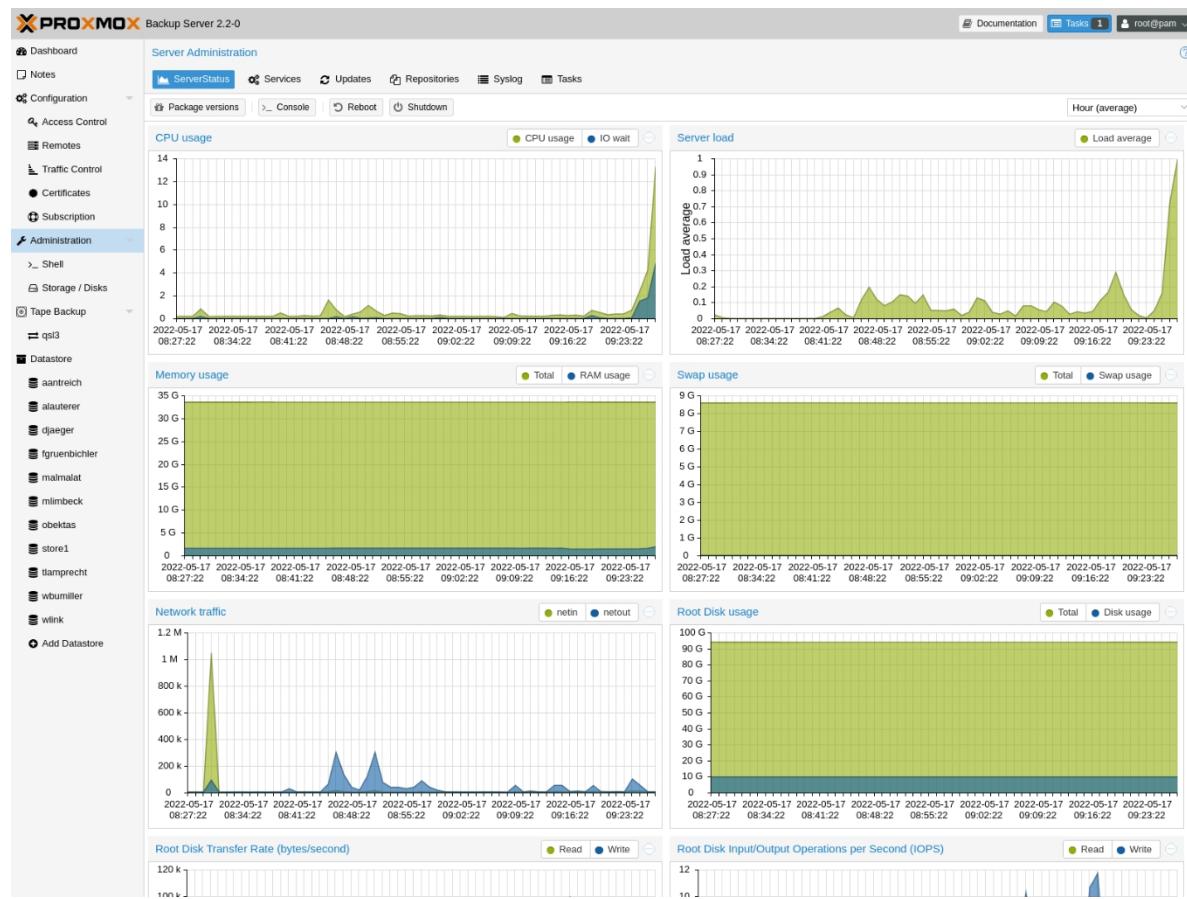
ダッシュボードには、サーバーのアクティビティとリソースの使用状況の概要が表示されます。具体的には、ハードウェアの使用状況、以前および現在実行中のタスクの概要、サブスクリプション情報が表示されます。

### 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
- シスログ：サーバーからのログメッセージの表示
- タスク複数のフィルターオプション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	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

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, the navigation menu includes options like Dashboard, Configuration, Access Control, Remotes, Subscription, Administration, Shell, Storage / Disks, Tape Backup, and Datastore. The Tape Backup section is currently selected. The main area has three tabs: Slots, Drives, and Import-Export Slots.

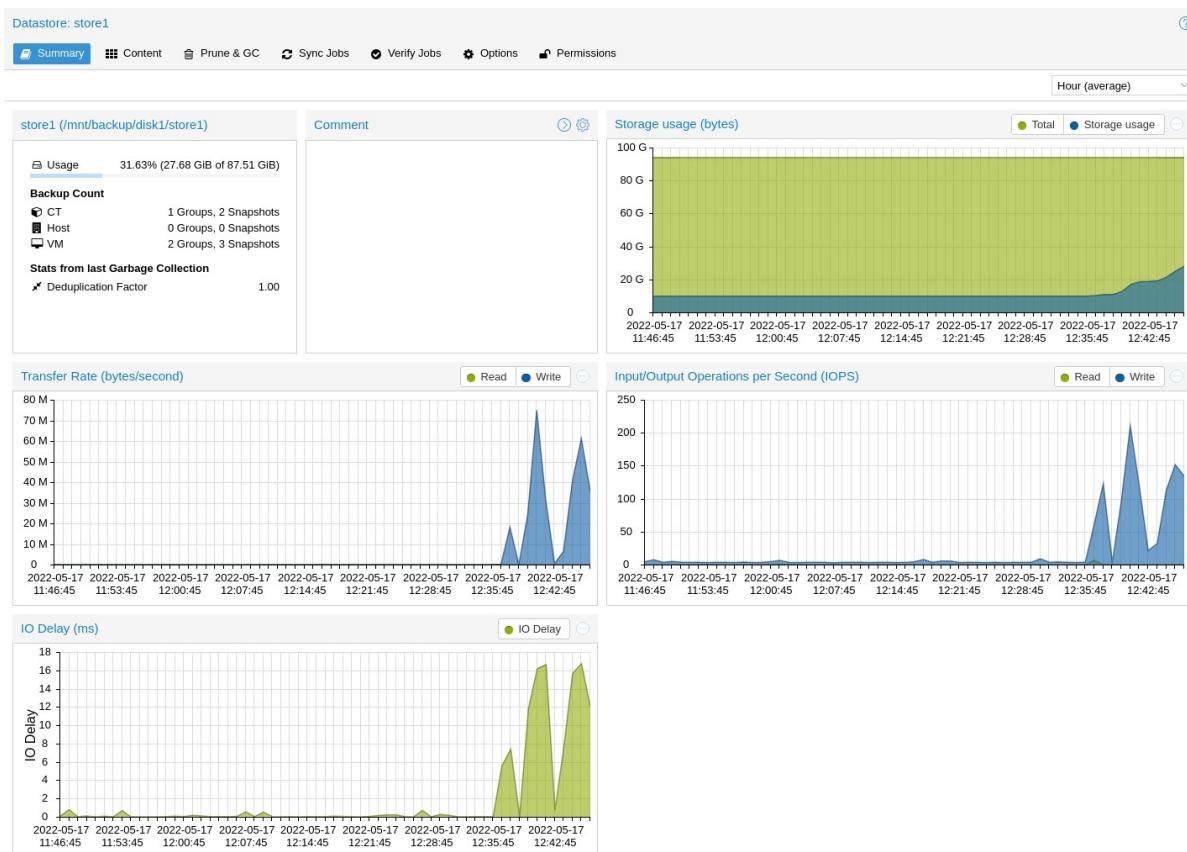
- Slots:** A table showing 20 slots. Rows 3 and 4 are labeled TAPES2L4 and TAPES3L4 respectively, with content "writable (pve-backup)". Rows 5 through 20 are empty.
- Drives:** A table showing two drives: qdrive0 (TAPES0L4) and qdrive1 (TAPES1L4), both in "idle" state.
- Import-Export Slots:** A table showing four slots (21, 22, 23, 24) which are empty.

テープバックアップセクションには、テープメディアセット、インベントリ、ドライブ、チェンジャー、暗号化キー、およびテープバックアップジョブ自体を管理するためのオプションを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
- プルーニングと**GC**: プルーニングとガベージコレクションの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: データストア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。



---

## チャプター ファイブ

---

# 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	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	ZL22J443	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	ZL22DBAS	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									
名前	使用	gpt	ディスクタイプ	サイズ	モデル	消耗	ステータス	使用	gpt
イプ	サイズ	モデル	消耗	ステータス					ディスクタ
SDA	LVM	1	hdd	34359738368	QEMU_HARDDISK	-	通過		
QEMU_HARDDISK									
sdb	未使用	1	hdd	68719476736	QEMU_HARDDISK	-	パス済み		。

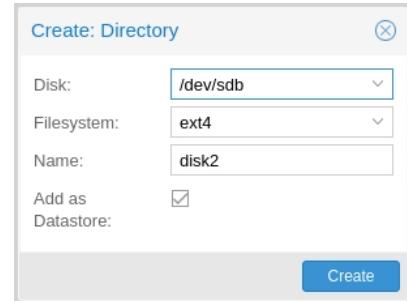
(次ページへ続く)

(前ページから続く)

sdc   未使用	1	hdd   68719476736   QEMU_HARDDISK	-   パスされた     。
-----------	---	-----------------------------------	-----------------

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：

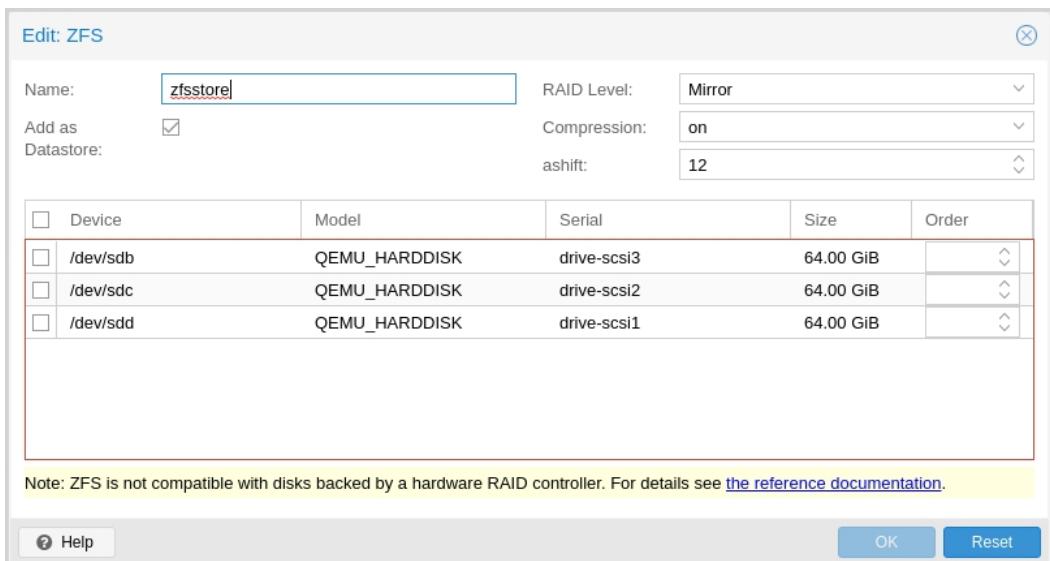
```
# proxmox-backup-manager disk initialize sdX
```



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



ウェブインターフェイスの **Administration -> Storage/Disks -> ZFS** から、または `zpool create` を使用して、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。

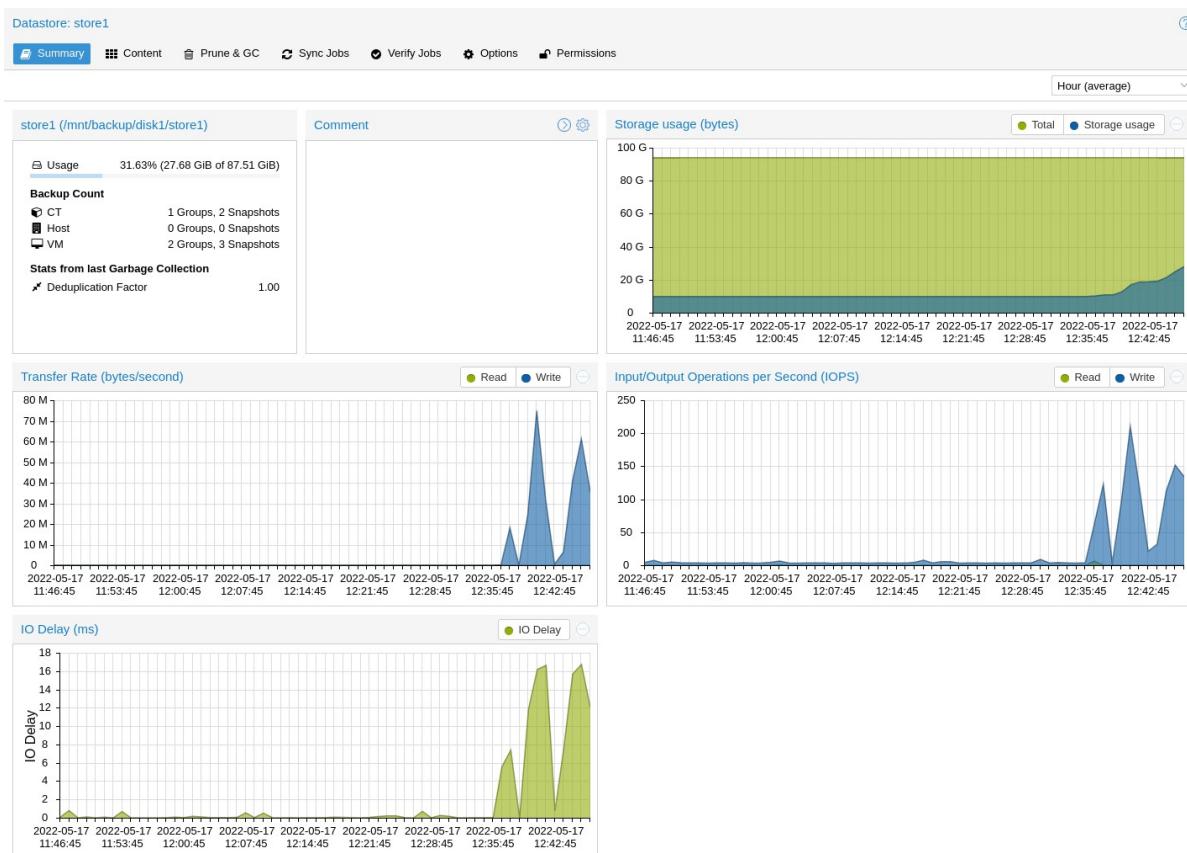
Proxmox Backup Serverはsmartmontoolsパッケージを使用しています。これは、ローカルハードディスクのS.M.A.R.T.システムを監視および制御するためのツールセットです。ディスクがS.M.A.R.T.機能をサポートしており、かつ

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（詳細は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、1つのディレクトリーにつき少なくとも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つの<sup>16</sup> 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			V.	2022-05-17 12:41:54	2	root@pam	No	All OK
vm/115			V.		1			None
vm/118			V.					
dc-eu-central			V.					
dc-eu-west			V.					
dc-oceania			V.					
dc-singapore			V.					
dc-usa			V.					

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、時間ごと、日ごと、週ごと、月ごと、年ごとの間隔ごとにバックアップ・スナップショットを何個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時間非依存数があります。[ブルーニングと削除](#) 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

General		Prune Options	
Name:	store1	GC Schedule:	Tue 04:27
Backing Path:	/mnt/backup/disk1/store1	Prune Schedule:	daily
Comment:			
<a href="#">? Help</a>		Add	

## **Proxmox Backup ドキュメント、リリース 3.2.7-1**

---

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。
- プルーンスケジュールとは、プルーニングを行う頻度のことです。
- プ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 ([バックアップのプルーニングと再移動を参照](#))。
- 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
```

## データストアの管理

コマンドラインから既存のデータストアを一覧表示するには、以下を実行します：

# proxmox-backup-manager データストア一覧		
名前	パス	コメント
store1	/backup/disk1/store1	これは私のデフォルトのストレージです。

データストアのガベージ・コレクションおよびプルーンの設定は、GUIからデータストアを編集するか、updateサブコマンドを使用して変更できます。たとえば、次のコマンドはupdateサブコマンドを使用してガベージ・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	
名前	値
名前	店舗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-schedule	Tue 04:27
キープラスト	7
剪定スケジュール	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 -arlh /backup/disk1/store1
276493 -rw-r--r-- 1 backup           backup0 Jul 8 12:35 .lock
276490 drwxr-x--- 1 backup backup 1064960 Jul 8 12:35 .chunks
```

.lockはプロセスのロックに使用される空のファイルです。

.chunksディレクトリには、*0000*から始まり*fffff*まで16進数で増加するフォルダが含まれます。これらのディレクトリには、バックアップ操作が実行された後、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 -arlh /backup/disk1/store1/.chunks
545824 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 ffff
545823 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 fffe
415621 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 fffd
415620 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 fffc
353187 drwxr-x--- 2 backup バックアップ 4.0K Jul 8 12:35 fffb
344995 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 fffa
144079 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 fff9
144078 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 fff8
 1 4 4077 drwxr-x--- 2 backup バックアップ 4.0K Jul 8 12:35 fff7
403180 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 000c
403179 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 000b
403177 drwxr-x--- 2 バックアップ バックアップ 4.0K Jul 8 12:35 000a
402530 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 0009
402513 drwxr-x--- 2 バックアップ バックアップ 4.0K Jul 8 12:35 0008
402509 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 0007
276509 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 0006
276508 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 0005
276507 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 0004
276501 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 0003
276499 drwxr-x--- 2 backup backup 4.0K Jul 8 12:35 0002
276498 drwxr-x--- 2 backup バックアップ 4.0K Jul 8 12:35 0001
276494 drwxr-x--- 2 バックアップ バックアップ 4.0K Jul 8 12:35 0000
276489 drwxr-xr-x 3 backup backup 4.0K Jul 8 12:35 .
276490 drwxr-x--- 1 backup backup 1.1M Jul 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	
	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
- メンテナンスマード
- 受信バックアップの検証

### チューニング

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

- チャンクオーダーベリファイとテープバックアップのチャンク順:

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処理します。ほとんどのストレージ、特に回転ディスクではこれで問題ないでしょう。
- *none* インデックスファイル(.fidx/.didx)に現れる順番でチャンクを反復します。これは多くの遅い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)の収集と並べ替えには、並べ替えの繰り返しで得られる時間よりも多くの時間がかかることがあります。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'
```

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、身代金を支払うまでファイルを暗号化するマルウェアの一種です。Proxmox Backup Serverは、オフサーバ、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つは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、オンサイト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、作成だけはできます。

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[フルーンジョブ](#)を使用して直接バックアップのフ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、削除は行わないようすることで、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や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。



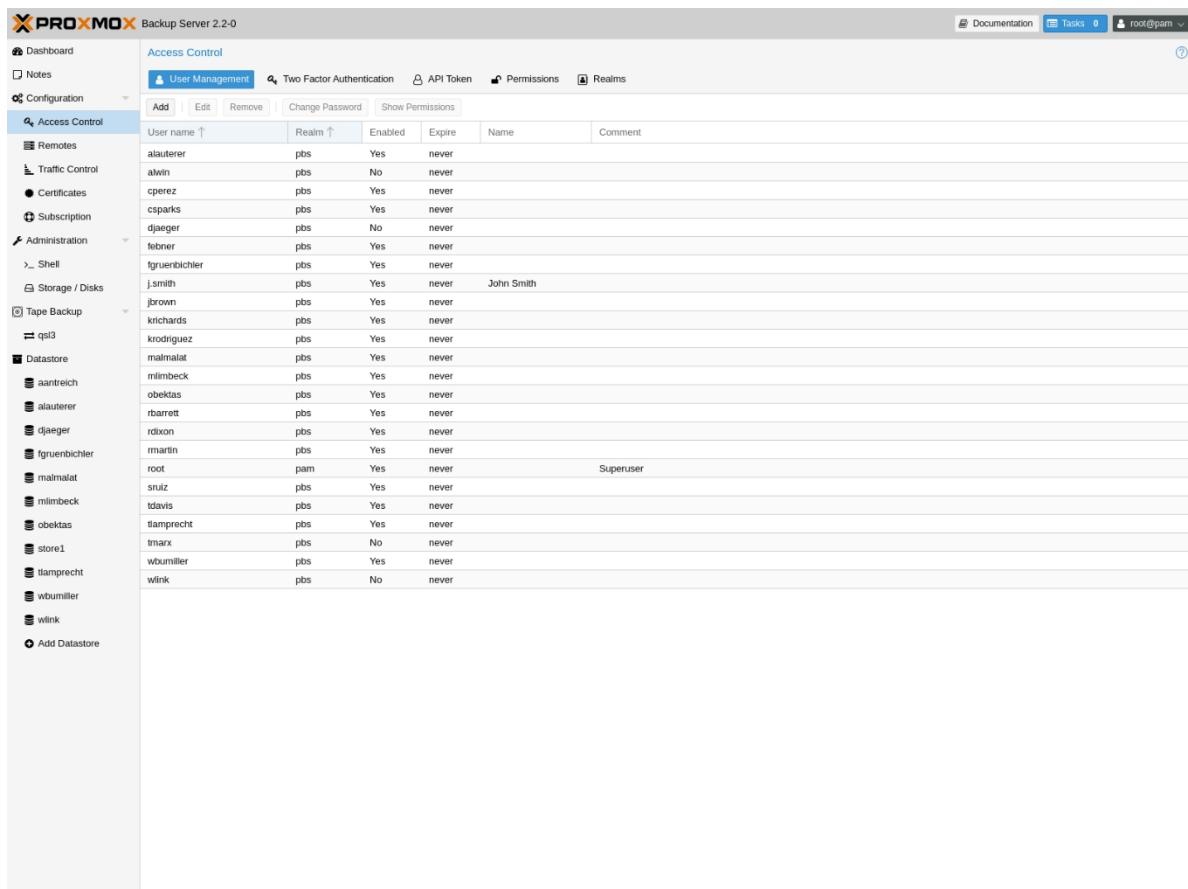
---

## チャプター シックス

---

### ユーザー管理

#### 6.1 ユーザー設定



The screenshot shows the Proxmox Backup Server 2.2-0 interface. On the left is a sidebar with various management sections like Dashboard, Notes, Configuration, and Tape Backup. The main area is titled 'Access Control' and specifically 'User Management'. It displays a table of users with columns: User name, Realm, Enabled, Expire, Name, and Comment. The table lists several users such as alauterer, alwin, cperez, csparks, djaeger, fechner, fgruenbichler, jsmith, jbrown, krichards, krodriguez, malmat, milimbeck, obektas, rbarrett, rdixon, rmarin, root, sruiz, tdavis, tamprecht, tmarr, wbumiller, and wilink. The 'Enabled' column shows values like 'Yes' or 'No', and the 'Expire' column shows 'never' or specific dates.

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 (ユーザはシステム上に存在する必要があります)。

##### pbs

Proxmox Backup Server realm. 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。

## エル

LDAPサーバー。ユーザは外部のLDAPサーバに対して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							
	userid	enable	expire	firstname	lastname	email	comment
ルート@パム	1			スーパーユーザー			

Add: User

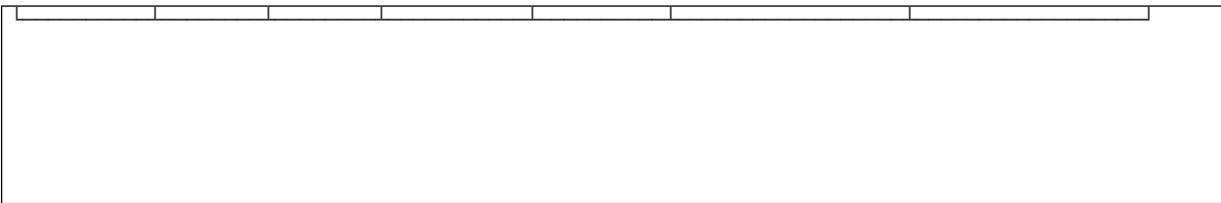
User name:	j.smith	First Name:	John
Password:	*****	Last Name:	Smith
Confirm password:	*****	E-Mail:	j.smith@example.com
Expire:	never		
Enabled:	<input checked="" type="checkbox"/>		
Comment:			
<a href="#">Help</a>	<a href="#">Add</a>		

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 --email john@example.com
# proxmox-backup-manager user update john@pbs --firstname John --lastname Smith
# proxmox-backup-manager user update john@pbs --comment "ユーザの一例です。"
```

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							
	userid	enable	expire	firstname	lastname	email	comment
john@pbs	1		ジョン		スミス	john@example.com	ユーザの一例です
.							
root@pam	1				スーパーユーザー		



新しく作成されたユーザー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",
```

(次ページへ続く)

(前ページから続く)

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

**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	有効	有効期限	コメント	トークンID
john@pbs!クライアント1		1 .			

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システムを使用しています。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状態を知ることができます。

#### シス.モディファイ

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

#### 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カンの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 トーカンは、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 トーカ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・メディアに書き込むことができます。

**テープ・リード**

`Tape.Read`を使用すると、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能够です。

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の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、プ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内容を読み取り、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タリーエへのアクセス
/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、常にユーザーの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、上位レベルから継承されたものに置き換わります。

### 6.3.4 設定と管理

The screenshot shows a modal dialog titled "Add: User Permission". It contains the following fields:

- Path: /datastore/store1
- User: j.smith@pbs
- Role: DatastoreAdmin
- Propagate:

At the bottom of the dialog are two buttons: "Help" and "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(:)で区切られています。典型的なエントリは次のような形式です：

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

各フィールドのデータは以下の通り：

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(单一のデータストア、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ー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：

ugid	パス	プロパゲート	roleid	
john@pbs	/datastore/store1	1	DatastoreAdmin	データストア管理者

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、パスに/datastore/store2を使用します。リモートストアの場合は、/remote/{remote}/{ストア名}という規約を使用します。{remote}はリモートの名前（後述の「リモート」を参照）、{ストア名}はリモート上のデータストアの名前です。

### 6.3.5 API トークンのパーミッション

API トークンのパーミッションは、対応するユーザーのものとは別に、そのIDを含むACLに基づいて計算されます。その結果、与えられたパスのパーミッション・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 ドキュメント、リリース 3.2.7-1

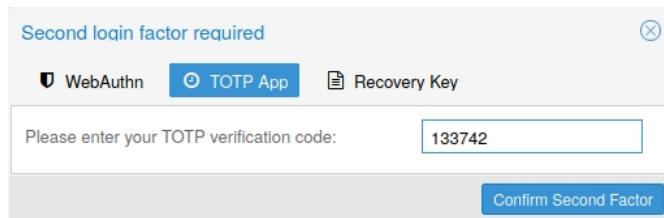
```
# proxmox-backup-manager user permissions john@pbs --path /datastore/store1 (*)が付  
いた特権にはpropagateフラグが設定されています。  
パス/データストア/ストア1  
- データストア.監査 (*)  
- データストア.バックアップ (*)  
- データストアの変更 (*)  
- データストア.ブルーン (*)  
- データストア.読み込み (*)  
- データストア.ベリファイ (*)  
# proxmox-backup-manager acl update /datastore/store1 DatastoreBackup --auth-id 'john@pbs!  
'→client'  
# 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）に成功するためには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、誰でもログインできるようになります。

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を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。各キーは一度しか使用できません。これらは、他のすべてのセカンドファクターが紛失または破損した場合でも、ロックアウトされないようにするために最適です。

### 6.4.3 セットアップ

## 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。

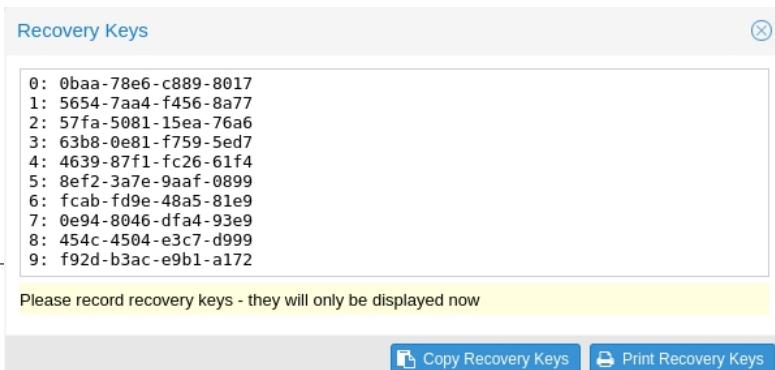
## ウェブオート

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。

これらの両方の要件を満たしたら、WebAuthn 設定をアクセスコントロールパネルの二要素認証タブ。

## リカバリーキー



### 6.4.二要素認証

Please record recovery keys - they will only be displayed now

Copy Recovery Keys Print Recovery Keys

リカバリキ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: 1ユーザーにつき1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によって破られる可能性があります。このため、第2要素によるログインに何度も失敗すると、ユーザーはロックアウトされます。

TOTPの場合、8回失敗すると、ユーザーのTOTPファクターは無効になります。回復キーでログインすると解除されます。TOTPが唯一の利用可能な要素であった場合、管理者の介入が必要であり、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が（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 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。

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

```
# ldap-test.comのPeopleのuser1
dn: uid=user1,ou=People,dc=ldap-
test,dc=com objectClass: top
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。dnがわかると、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。

---

## 6.5.認証領域

47



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サーバにバインドします。

全設定パラメータの一覧は [domains.cfg](#) にあります。

---

注意: 特定のユーザに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。

---

### 6.5.2 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 認証情報を指定した場合は、ドメイン名は Active Directory サーバが報告する bind ユーザのベースドメインから自動的に推測されます。

全設定パラメータの一覧は [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。

ユーザ同期は、GUI の **Configuration > Access Control > Realms** でレルムを選択して *Sync* ボタンを押すことで開始できます。同期ダイアログでは、レルム設定で設定されたデフォルトオプションの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。

---

## チャプター セブン

---

# 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リ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

user@pbs@マイホスト名:mydatastore	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	user@pbs! 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
[ff80::51]:マイデータストア	ルート@パム	[ff80::51]:8007	マイデータ 引き裂く
[ff80::51]:1234:mydatastore	ルート@パム	[ff80::51]:1234	マイデータ 引き裂く

## 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。上記の順番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 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。**

(任意の値に) 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**

(任意の値に) 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。

---

## 7.4 バックアップの作成

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

---

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

---

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

```
# 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" マウントポイントをスキップ:  
"/sys" マウントポイントをスキップ:  
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.pxb:/:/mnt/disk1 disk2.pxb:/:/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

一般的なタイプは、ファイル・アーカイブの .pxar とブロック・デバイス・イメージの .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
```

#### 7.4.1 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 .pxarexclude 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、グロブ一致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、一部のファイル/サブディレクトリを除き、ディレクトリ内のすべてのエントリを除外することができます。で終わる行は、ディレクトリにのみマッチします。.pxarexclude ファイルを含むディレクトリは、指定されたパターンのルートとみなされます。こ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小文字に、[0-9] は数字1文字にマッチします。

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

---

注意: 除外されたディレクトリはバックアップクライアントによって読み取られません。したがって

---

、除外されたサブディレクトリの .pxareexclude ファイルは何の効果もありません。  
.pxareexclude ファイルは通常のファイルとして扱われ、バックアップアーカイブに含まれます。

例えば、次のようなディレクトリ構造を考えてみましょう：

```
# ls -aR folder
folder/:
.:..,.pxareexclude サブフォルダ0 サブフォルダ1
:.../.pxareexclude/サブフォルダ0:
.... file0 file1 file2 file3 .pxareexclude folder/subfolder1:
.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
/サブフォルダー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 サブフォルダ0 サブフォルダ1
restored/subfolder0:
... file0 file2 .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.pxr:./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.pxr:./linux --exclude=/usr --exclude=/rust
```

#### 7.4.2 変更検出モード

Proxmoxバックアップクライアントのデフォルトの動作は、すべてのデータを読み込んでpxarアーカイブにエンコードするためです。エンコードされたストリームは可変サイズのチャンクに分割されます。各チャンクについてダイジェストが計算され、そのチャンクをアップロードする必要があるか、それともアップロードせずにインデックス化できるかを決定するために使用されます。バックアップされたファイルがほとんど変更されていない場合、再読み込みを行い、対応するチャンク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変更検出モードをデフォルトからメタデータベースの検出に切り替えることで、上記のような制限を減らし、可能な限り未変更のメタデータを持つ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します。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。

`change-detection-mode` を `data` に設定することで、メタデータモードを使用したときと同じ分割アーカイブを作成することができます。バックアップアーカイブを作成する際、現在のファイルのメタデータは以前の mpxar アーカイブで検索されたものと比較されます。メタデータの比較には、ファイルサイズ、ファイルタイプ、所有者とパーミッション情報、acls と属性、そして最も重要なファイルの mtime が含まれます。詳細は [pxar メタデータアーカイブフォーマット](#) を

---

参照してください。

変更されていない場合、エントリは、以前のバックアップスナップショットからのコンテンツを含む既に存在するチャンクをインデックス化することで、再読み込みせずにコンテンツのチャンクを再利用できるようにキャッシュされます。ファイルはチャンクを部分的にしか再利用しない可能性があるため（それによってパディングという無駄なスペースが発生します）、現在キャッシュされているエントリを再利用するか再エンコードするかの決定は、パディングの可能性を閾値と比較し、十分な情報が利用可能になるまで延期されます。

モード	説明
レガシー (現在のデフォルト) :	すべてのファイルを自己の 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.pxar:./linux --change-detection-mode=metadata
```

## 7.5 暗号化

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

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

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.. encryption 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 `~/.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。

### 7.5.1 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できます。

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。  
.E n.. 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  
M a.. key Password: *****
```

これにより、カレントディレクトリに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 をインポートします。 をインポートします。  
します。 証明書をインポートします、 ですから その  
`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。
```



- 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せん。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。

---

- 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。

---

- 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
/.
```

新規パスワードの確認\*\*\*\*\*  
パスワードの確認\*\*\*\*\*

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

警告キーがないと、バックアップしたファイルにアクセスできなくなります。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、最悪のシナリオに備えて、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

## **Proxmox Backup ドキュメント、リリース 3.2.7-1**

---

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
```

## 7.6 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。しかし、より重要なのは復元です。定期的にリカバリーテストを実施し、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.pcat1 index.json |
root.pxar catalog.pcat1 index.json
| host/elsa/2019-12-03T09:35:01Z | 51790622048 | root.pxar catalog.pcat1 index.json | です。
.+. -.
```

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カタログダンプ host/elsa/2019-12-03T09:35:01Z
d "./root.pxar.didx/etc/cifs-utils"
l "./root.pxar.didx/etc/cifs-utils/idmap-plugin"
d "../root.pxar.didx/etc/console-setup"
```

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。
```

### 7.6.1 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
bin... boot デブ その他 ホーム lib lib32
```

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

ナビゲーション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:/ > リスト選択された等/**/*txt
.p x.. ar:/ > 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：

```
.p x.. ar:/ > 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。

## 7.6.2 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
bindev home lib32libx32 mediaopt rootsbin sys usr
boot etc liblib64 lost+found mntproc run srvtmp
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、マウントポイントで `umount` コマンドを使用します：

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

## 7.7 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
```

## 7.8 バックアップ・グループの所有者の変更

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

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

これはWebインターフェイスから行うこともできます。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ー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。

## 7.9 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時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時間はカウントされません。

**--キープ・デイリー <N>**

過去<N>日間のバックアップを保持します。1日に複数のバックアップがある場合は、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 | キープ
host/elsa/2019-12-04T13:20:37Z | 1 | .
| host/elsa/2019-12-03T09:35:01Z | 0 | です。
| host/elsa/2019-11-22T11:54:47Z | 1 | .
| host/elsa/2019-11-21T12:36:25Z | 0 | .
host/elsa/2019-11-10T10:42:20Z | 1 | .
```

---

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。

---

**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

◦

---

## 7.10 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。このタスクはガベージ・コレクション・コマンドに任されています。ガベージ・コレクションは定期的に実行することをお勧めします。

ガベージコレクションは2つのフェーズで動作します。最初のフェーズでは、まだ使用されているすべてのデータブロックがマークされます。2番目のフェーズでは、未使用的データブロックが削除されます。

---

注意 このコマンドは、既存のすべてのバックアップ・インデックス・ファイルを読み込み、完全な

チャンクストア。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。

---

注：ガベージコレクションは、少なくとも1日（ちょうど24時間5分）使用されていないチャンクのみを削除します。使用中のチャンクは、`atime`（アクセス時間）プロパティを更新するチャンクに触れることでマークされるため、この猶予期間は必要です。ファイルシステムはデフォルトで`relatime`オプションでマウントされます。この結果、最後のアクセスが少なくとも24時間前である場合にのみ`atime`プロパティを更新することで、パフォーマンスが向上します。欠点は、この24時間以内にチャンクをタッチしても、常に`atime`プロパティが更新されるとは限らないことです。

猶予期間内のチャンクは、ガベージコレクションタスクの終了時に、削除保留として記録されます。

```
# proxmox-backup-client garbage-collect
store2の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
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。

## 7.11 ベンチマー킹

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）が指定されている場合、TLSアップロード速度も測定されます。

`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 ドキュメント、リリース 3.2.7-1

```
# 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 伸長速度: 1233.35
MB/s AES256/GCM 速度: 3688.27 MB/s
ペリファイ速度: 783.43MB/秒
```

名前	価値
TLS (バックアップの最大アップロード速度)	1267.41 MB/s (103%)
SHA256チェックサムの計算速度	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%)

---

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
```

その後、ストレージのステータスを表示できるようになります：

```
# pvesm status --storage store2
名称          種類      ステータス        合計          使用          利用可能
ストア2          pbs    アクティブ    3905109820    1336687816
2568422004          34.23%
```

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

## ナイン

---

## 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。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.pjar /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 `.pxareexclude-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 `.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「[バックアップの作成](#)」で説明したものと同じです。

## 9.2 アーカイブの抽出

既存のアーカイブ、`archive.pxar` は、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` ファイルをターゲット `/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、アーカイブのルートを基準として、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：

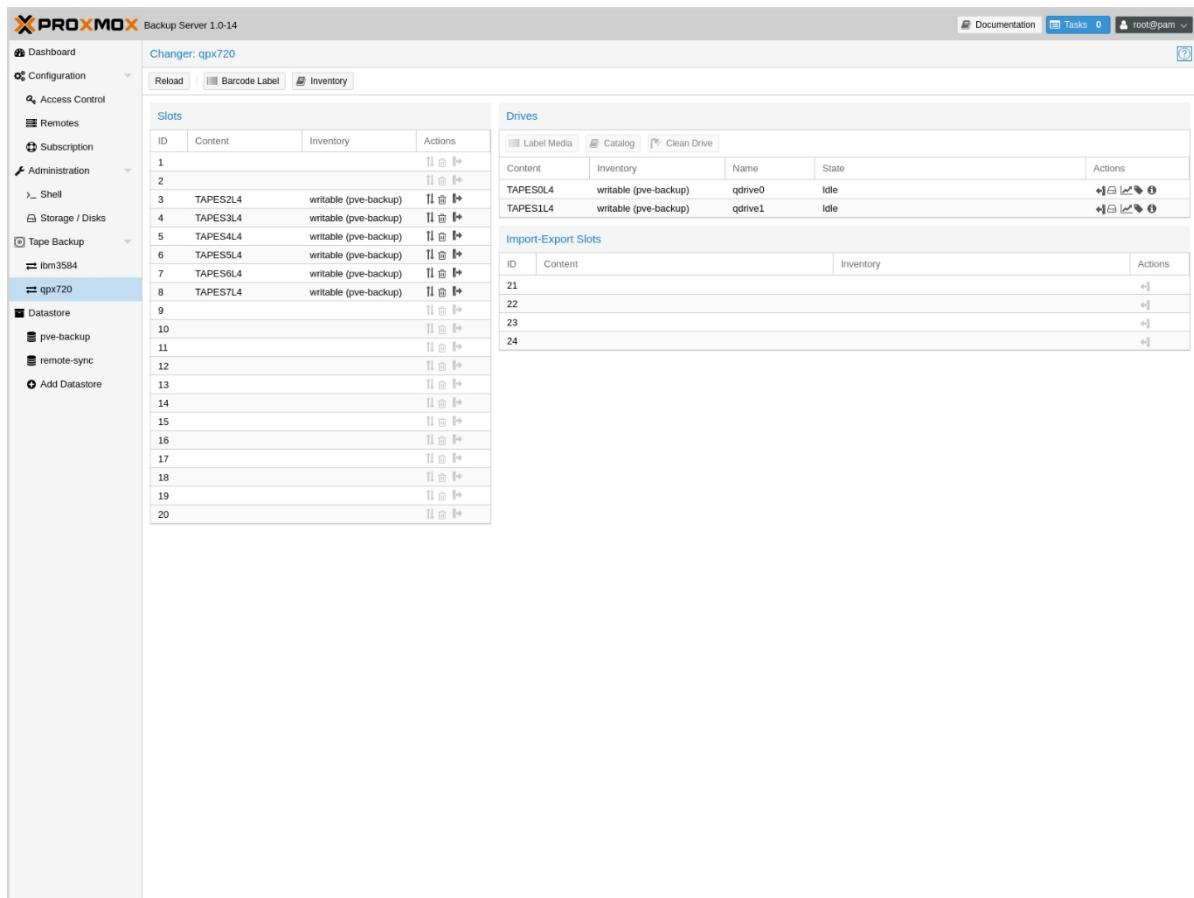
```
# pxar mount archive.pxar /mnt
```

アーカイブがマウントされると、指定されたマウントポイントでそのコンテンツにアクセスすることができます。

```
# cd /mnt
# ls
bindev home lib32libx32 mediaopt rootsbin sys usr
boot etc liblib64 lost+found mntproc run srvtmp
var
```

テン

## テープバックアップ



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。

---

## チャプ

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

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

---

注：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、他のドラ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、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'低速'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**パス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](#)で定義されているCode 39としてエンコードされた8文字の文字列を使用しています。

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。独立したドライブもありますが、ドライブは通常テープ・ライブラリ内に搭載されています。

**テープチェンジャー**

テープ・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（追加のステータス情報付き）が保存されます。

#### カタログ

メディアカタログは、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す
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・スキャン
[パス|ベンダー| モデル|シリアル|]
[dev/tape/by-id/scsi-CC2C52 | Quantum | Superloader3 | CC2C52 | ]
```

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

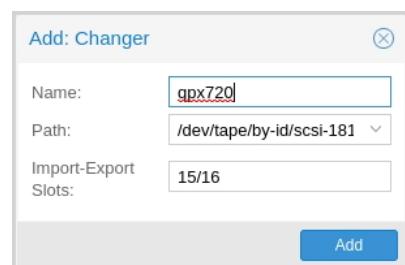
```
# proxmox-tape changer create sl3 --path /dev/tape/by-id/scsi-CC2C52
```

sl3は任意の名前です。

---

**注意:** /dev/tape/by-id/ 内の永続デバイスパス名を使用してください。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]
[名前 | sl3]
[パス | /dev/tape/by-id/scsi-CC2C52 | | | /dev/tape/by-id/scsi-CC2C52]
```

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

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

表示されます。セットアップをテストするには、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-tape changer status sl3				
エンtryーの	種類	エンtryーのID	チェンジャーのID	ロードされたスロット
ドライブ	0	vtape1	1	
スロット	1			
スロット	2	vtape2		

(次ページへ続く)

(前ページから続く)

...	..	.		
スロット		16		

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外部から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」と呼ばれるものもありますが、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もありません。

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
```

その後、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				
エンタリーの種類	エンタリーのID	チェンジャーのID	ロードされたスロット	ロードされたスロット
ドライブ	0	vtape1	1	
輸出入	15			
輸出入	16			
スロット	1			
スロット	2	vtape2		
...	..	.		
スロット	14			

## 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 is a sidebar with navigation links like Dashboard, Configuration, Access Control, Remotes, Subscription, Administration, Shell, Storage / Disks, Tape Backup (which is selected), Datastore, pve-backup, remote-sync, and Add Datastore. The main area has tabs for Content, Inventory, Changers, Drives (which is selected), Media Pools, Encryption Keys, and Backup Jobs. Under Drives, there are buttons for Add, Edit, and Remove. A table lists tape drives and changers:

Name ↑	Path	Vendor	Model	Serial	Drive Num...
Changer ibm3584 (4 items)					
drive0	/dev/tape/by-id/scsi-0187999213-sg	IBM	ULT3580-TD4	0187999213	0
drive1	/dev/tape/by-id/scsi-0649938692-sg	IBM	ULT3580-TD4	0649938692	1
drive2	/dev/tape/by-id/scsi-1293966585-sg	IBM	ULT3580-TD4	1293966585	2
drive3	/dev/tape/by-id/scsi-0301768295-sg	IBM	ULT3580-TD1	0301768295	3
Changer qpx720 (2 items)					
qdrive0	/dev/tape/by-id/scsi-1775099095-sg	IBM	ULT3580-TD1	1775099095	0
qdrive1	/dev/tape/by-id/scsi-0036472022-sg	IBM	ULT3580-TD4	0036472022	1

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|シリアル|。

/dev/tape/by-id/scsi-12345-sg	IBM	ULT3580-TD4	12345	/dev/tape/by-id/scsi-12345-sg	IBM	ULT3580-TD4	12345
-------------------------------	-----	-------------	-------	-------------------------------	-----	-------------	-------

Add: Drive

Name:	<input type="text" value="qdrive0"/>
Changer:	<input type="text" value="qpx720"/>
Drive Number:	<input type="text" value="0"/>
Path:	<input type="text" value="/dev/tape/by-id/scsi-0187999213-sg"/>

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
名前 | 値
名前 | マイドライブ
パス | /dev/tape/by-id/scsi-12345-sg。
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
名前 | パス | チェンジャー | ベンダー | モデル | シリアル
mydrive | /dev/tape/by-id/scsi-12345-sg | sl3 | IBM | ULT3580-TD4 | 12345 | です。
```

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
名前 | 値
プロック サイズ | 0
密度 | LT04
圧縮 | 1
バッファモード | 1
警告フラグ | (空)
ファイル 番号 | 0
プロック 番号 | 0
製造 | Fri Dec 13 01:00:00 2019 | (金) 12月13日01時00分00秒
書き込みバイト数 | 501.80 GiB
バイト- リード | 4.00 MiB
```

## Proxmox Backup ドキュメント、リリース 3.2.7-1

メディアム・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 like Dashboard, Configuration, Access Control, Remotes, Subscription, Administration, Shell, Storage / Disks, and Tape Backup (which is selected). The main area has tabs for Content, Inventory, Changers, Drives, Media Pools (which is selected), Encryption Keys, and Backup Jobs. Under Media Pools, there's a table with columns Name, Allocation, Retention, Encryption, and Comment. It contains two entries: 'pve-backup' with 'Allocation ...', 'Retention ...', 'Encryption Yes', and 'Comment'; and 'remote-backup' with 'Allocation ...', 'Retention ...', 'Encryption No', and 'Comment'. There are 'Add', 'Edit', and 'Remove' buttons at the top of the table.

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に使用されます。1つまたは複数のバックアップ・ジョブがメディア・セットに書き込み、順序付けられたテープ・グループを作成します。メディア・セットは一意のIDで識別されます。このIDとシーケンス番号は、そのセットの各テープ（テー

プ・ラベル）に保存されます。  
メディアセット割り当てボタン

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。

- 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



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。

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後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できます：

- 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。

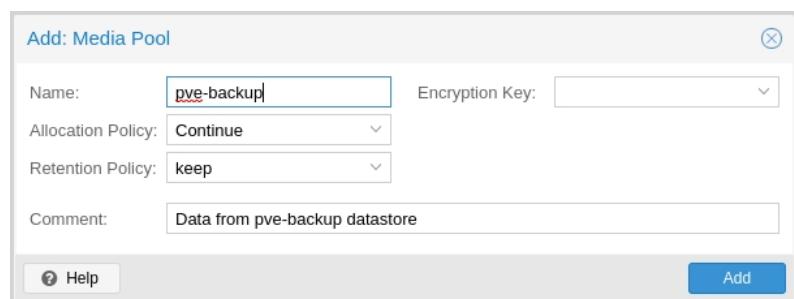
## ハードウェア暗号化

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、オフサイトへの輸送中にテープを紛失した場合などです。

**注:** バックアップ・クライアントもデータを暗号化する場合、テープ上のデータは二重に暗号化されます。

パスワードで保護されたキーは各メディアに保存され、パスワードを使って [キーを復元](#)することができます。万が一、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	テンプレート
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, Shell, Storage / Disks, and Tape Backup (which is selected). The main area has tabs for Content, Inventory, Changers, Drives, Media Pools, Encryption Keys, and Backup Jobs. The Backup Jobs tab is active, showing a table with one row. The table columns are: Job ID, Datastore, Media Pool, Drive, Eject, Export, Latest Only, Schedule, Last Backup, Duration, Status, Next Run, and Comment. The single row contains: weekly..., pve-backup, pve-backup, drive0, No, No, No, sat 18:15, -, -, -, 2021-04-17 18:15:00.

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
```

---

注意: `export-media-set`オプションを使用すると、現在のセットのテープがオンラインでなくなるため、新しいメディア・セットを強制的に開始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
```

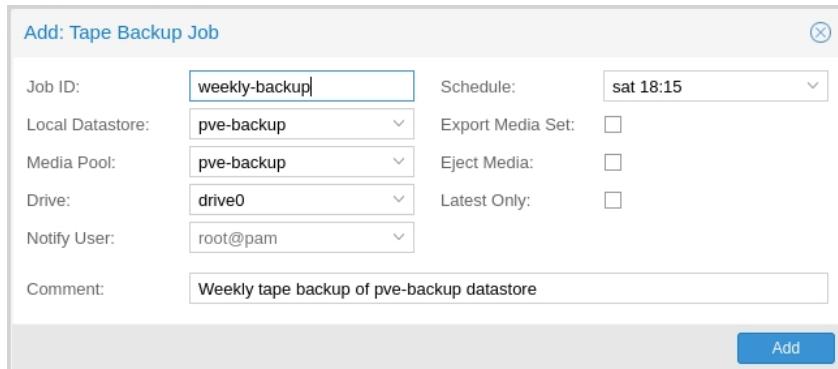
ジョブを削除するには

```
# 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：安全上の理由から、テープにデータがある場合、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 Backup ドキュメント、リリース 3.2.7-1

# proxmox-tape read-label
名前   価値
チェンジャーID   vtape1
uuid   7f42c4dd-9626-4d89-9f2b-c7bc6da7d533   (英語)
ctime   1月6日(水)09時07分51秒 2021
プール   毎日
media-set-uuid   00000000-0000-0000-000000000000   メディア・セット・ユアイディー
media-set-ctime   1月6日(水)09時07分51秒 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あります：

**--eject-media** ジョブ完了時にメディアを排出します。

通常、使用後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。

--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
label-text | ←メディア設定
| vm/201/2021- ↗9da37a55-a
| ...
| ..
| .
```

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場所にリストアし、次に関連するチャンクアーカイブをリストアし、最後にスナップショットデータをターゲットデータストアにリストアします。

1回のリストア操作で複数のスナップショットをリストアするために、`snapshot`パラメータを複数回渡すことができます。

---

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,source=ns1,target=ns2,max-depth=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 1.0-14

Documentation Tasks 0 root@pam ~

**Tape Backup**

**Content** **Inventory** **Changers** **Drives** **Media Pools** **Encryption Keys** **Backup Jobs**

**Add** **Remove** **Restore Key**

Hint ↑	Fingerprint	Created
tape pw 2020	ab:67:f4:1b:72:ca:33:e1:91:bf:0b:da:df:1f:77:78:60:08:87:63:2f:49:84:e7:e1:7d:1b:9e:9f:0a:7c:93	2021-04-13 16:14:19

**Administration**

Shell

Storage / Disks

**Tape Backup**

- ibm3584
- gpx720

**Datastore**

- pve-backup
- remote-sync

**Add Datastore**

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	
	指紋   ヒント
14:f8:79:b9:f5:13:e5:dc:... :b6:20:e1:7f:f5:0f	テープ pw 2020     テープ pw 2020

暗号化キーの詳細を表示します:

## Proxmox Backup ドキュメント、リリース 3.2.7-1

```
# proxmox-tape key show 14:f8:79:b9:f5:13:e5:dc:...:b6:20:e1:7f:f5:0f
```

名前	価値
kdf	スクリプト
作成日時	1月23日(土)14時47分21秒2021
修正	1月23日(土)14時47分21秒

(次ページへ続く)

(前ページから続く)

フインガープリント | 14:f8:79:b9:f5:13:e5:dc:...:b6:20:e1:7f:f5:0f |.  
|-----|-----|-----|-----|-----|-----|-----|  
ヒント | テープ pw 2020 |

`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。それ以降のリストアジョブでは、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になります。

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。例えば、当社のテープ・ライブラリーのクリーニング・カートリッジはスロット3の中にあります:

# proxmox-tape changer status sl3					
エントリーの ドライブ	種類	エントリーのID	チェンジャーのID	ロードされたスロット	ロードされたスロット
	0	vtape1	1		
スロット	1				
スロット	2	vtape2			
スロット	3	CLN001CU			
...	...	.			

クリーニング作業を開始するには、単に実行します：

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、法的または保護目的で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ことがあります。

Proxmox Backup Serverは、通常のテープとWORMテープを区別しません。混乱を避けるため、WORMバックアップには別の命名スキームを使用し、専用のメディア・プールを使用します。同じメディア・プールにWORMテープと非WORMテープを混在させないでください。

## 10.8 セットアップ例

メディア・プールとスケジュールを管理するための設定例をいくつか紹介します。これは網羅的なリストではなく、便利な設定の組み合わせはもっとたくさんあります。

### 10.8.1 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、そのテープのチヤ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。  
10.7.WORMテープ

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

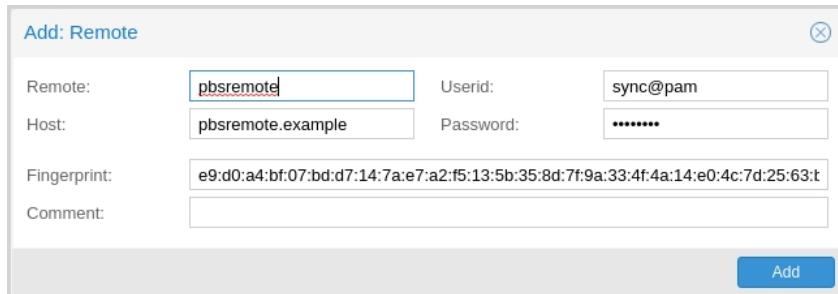
フルーンの設定やテープバックアップのスケジュールと組み合わせることで、一部のバックアップの長期保存を実現する一方、最近のバックアップは、およそ4週間ごと（つまり、3週間に現在の週を加えた期間）に期限切れになる小さなメディアセットに保存します。



## 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用し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|          フィンガープリント|
コメント |。
| pbs2 | pbs2.example | sync@pam | 64:d3:ff:3a:50:38:53:5a:9b:f7:50:...:ab:fe | .
# proxmox-backup-manager remote remove pbs2
```

## 11.2 シンク・ジョブズ

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。

```
# proxmox-backup-manager sync-job create pbs2-local --remote pbs2 --remote-store local -- です
。→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 | comment | | | id| store | remote |
remote-store | schedule | comment | | | | |
PBS2-LOCAL | ローカル | PBS2 | ローカル | Wed 02:30 | オフサイト | off site
# 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。`owner` オプションが設定されていない場合（デフォルトは `root@pam`）または設定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` オプションを設定すると、指定し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}$'
```

`remove-vanished`オプションを扱うために、同じフィルターがローカルグループにも適用されます。グループフィルタの前に `exclude:` を付けることで、グループファイル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・フィルターにマッチするもの。

---

**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。

---

### 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形でデータストアのサブセットを同期するように構成できます。以下のパラメータは、同期ジョブの実行時にネームスペースがどのように扱われるかに影響します:

- `remote-ns:` リモート名前空間アンカー (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。

## Proxmox Ratelimitsがなく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-nsが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 同期範囲に関するその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オプションを設定することで、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
```



## 保守作業

### 12.1 剪定

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：

#### 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>時間のバックアップを保持します。1時間に複数のバックアップがある場合は、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>日間のバックアップを保持します。1日に複数のバックアップがある場合は、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 プルーンの求人

The screenshot shows the Proxmox Backup interface with the following details:

- Datastore:** store1
- Prune & GC Tab:** Selected tab.
- Garbage Collection:**
  - Start Garbage Collection button.
  - Garbage Collection Schedule: 21:00.
- Prune Jobs:**

Job ID	Namespace	Max. Depth	Schedule	Keep					Last Prune	Duration	Status	Next Run	Comment
				Last	Hourly	Daily	Weekly	Monthly					
s-7d...	dc-usa	daily	3	13	8	11	1	-	-	-	2022-11-29 00:00:00		
s-a1...	dc-eu-central	hourly	3	-	13	8	11	9	2022-11-28 14:00:00	<0.1s	OK	2022-11-28 15:00:00	

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データストアパネルの**[Prune & GC]**タブまたはデータストア自体からプルーンのジョブを管理できます。また、`proxmox-backup-manager prune-job`コマンドで管理することもできます。`prune`ジョブの設定情報は`/etc/proxmox-backup/prune.cfg`に保存されています。新しい`prune`ジョブを作成するには、GUIで追加ボタンをクリックするか、マネージャーCLIの`create`サブコマンドを使用します。`prune`ジョブを作成したら、GUIから手動で開始するか、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するデータストア。

**ns <名前空間**

プルーンのジョブを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。

Add: Prune Job

Datastore: store1

Prune Schedule: Mon..Fri 21:00

Namespace: dc-eu-west

Enabled:

Max. Depth: Full

Keep Last: 3

Keep Daily: 13

Keep Monthly: 8

Keep Hourly: 0

Keep Weekly: 0

Keep Yearly: 0

Comment: Work-day pruning schedule for EU West namespace

Help Advanced Add

## スケジュール

このジョブを自動的に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できます。

## 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'

	Backup Time ↓	keep
keep-last:	1	true
keep-hourly:	1	false
keep-daily:	2	false
keep-weekly:	0	true
keep-monthly:	0	false
keep-yearly:	0	true
	2020-11-07 15:44:54	true
	2020-11-07 15:44:50	false
	2020-11-07 15:39:25	false
	2020-11-07 11:00:32	true
	2020-11-07 10:43:42	false
	2020-08-19 18:25:50	true
	2020-07-23 13:32:43	true

Prune

特定のバックアップグループを手動でプルーニングするには、[バックアップのプルーニングと削除](#)で説明した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グループの**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と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10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 - 毎日のバックアップには関係ありません。手動バックアップは keep-last 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。

保存期間が不必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ッ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ッ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ッ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）がそのチャックを参照している可能性があるため、そのチャックと関連付けられたチャッ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ッ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つた未使用のバックアップチャンクを特定し、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、これらのチャンクが占有する領域を効率的に再生するように設計されています。

ガベージコレクション (GC) プロセスはデータストアごとに実行され、2つのフェーズに分けられます：

- 第一段階マーク すべてのインデックスファイルが読み込まれ、参照されたチャンクファイルのアクセスタイムが更新されます。
- フェーズ2：タスクはすべてのチャンクを反復し、そのファイルアクセス時間をチェックし、それがカットオフ時間（すなわち、GCが開始された時間と、安全性とLinuxファイルシステムの動作のための若干の余裕を加えた時間）より古い場合、タスクはそのチャンクがどのバックアップインデックスでも参照されておらず、スキャンするインデックスを持たない現在実行中のバックアップの一部でもないことを認識します。そのため、チャンクは安全に削除できます。

### 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サブコマンドを使用してデータストア全体のガベージコレクションを手動で開始し、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ないかもしれません。そのため、Proxmox Backup ServerがGCを処理するようにスケジュールを設定することができます。

データストアのGCスケジュールの設定や編集は

`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ーカイブする場合などに便利です。

### 12.3 檢証

Add: VerifyJob

Datastore:	store1	Skip verified snapshots:	<input checked="" type="checkbox"/>
Schedule:	<input type="button" value="daily"/> <input style="width: 20px; height: 20px;" type="button" value="▼"/>	Re-Verify After (days):	<input type="text" value="30"/> <input style="width: 20px; height: 20px;" type="button" value="▼"/>
Comment:	<input type="text"/>		

[Help](#) [Add](#)

Proxmox Backup Serverには、バックアップデータが無傷であることを確認するためのさまざまな検証オプションがあります。検証は通常、検証ジョブの作成によって実行されます。これは、指定された間隔で検証を実行するスケジュールされたタスクです（[カレンダーイベントを参照](#)）。このジョブでは、すでに検証されたスナップショットを無視するかどうかを設定したり、スナップショットを検証する順序を変更したりすることができます。

ツ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、ガベージコレクション、同期タスクの結果に関する通知を送信することができます。

詳細については、「[通知](#)」の章を参照してください。

## 12.5 メンテナンスマード

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マ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、メンテナンスマ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 Backupは有名なDebian Linuxディストリビューションをベースにしています。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のリポジトリから入手できます。さらに、Proxmoxに関連するすべてのパッケージを展開するために、独自のパッケージリポジトリを提供しています。これには、必要に応じて Debian パッケージのアップデートも含まれます。

また、Ubuntuカーネルをベースに特別に最適化されたLinuxカーネルも提供しています。このカーネルにはZFS用のドライバが含まれています。

以下のセクションでは、バックアップ関連のトピック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を使用することで、低予算のハードウェアで最大限のエンタープライズ機能を実現することができます。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
- 繙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メモリを使用してください。データの破損を防ぐため、高品質のECC RAMの使用をお勧めします。

専用のキャッシュおよび/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モードでフラッシュされた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 <プール> <デバイス1> <デバイス2>
```

## RAID-1で新しいプールを作成

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 <プール> ミラー <デバイス1> <デバイス2> ミラー <デバイス3> <デバイス4>
```

## RAIDZ-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 <プール> raidz1 <デバイス1> <デバイス2> <デバイス3>
```

## RAIDZ-2で新しいプールを作成

最低4ディスク

```
# zpool create -f -o ashift=12 <プール> 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\*による新規プールの作成」で示すように、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、单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、ブートローダーとして*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。

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ピングスペースは、サーバーをブロックしたり、高いIO負荷を発生させるなどの問題を引き起こす可能性があります。

メモリ不足に陥らないよう、十分なメモリを使用することを強くお勧めします。スワップを追加したい場合は、物理ディスク上にパーティションを作成し、スワップデバイスとして使用することをお勧めします。インストーラの詳細オプションで、この目的のために空き領域を残すことができます。さらに、*swappiness* の値を下げるることもできます。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、以下の行を追加します：

```
vm.swappiness = 10
```

表1: Linuxカーネルのスワッピネス・パラメータ値

価値	戦略
vm.swappiness = 0	カーネルは「メモリ不足」状態を回避するためだけにスワップを行います。
vm.swappiness = 1	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。
vm.swappiness = 10	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。
vm.swappiness = 60	デフォルト値。
vm.swappiness = 100	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。*lzb*、*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。アルゴリズムや圧縮可能なデータによっては、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 <データセット
```

この変更の影響を受けるのは新しいブロックだけです。

## 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。

特別なデバイスは、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。特別なデバイスには高速SSDを使用してください。

---

**重要：** 特殊デバイスの冗長性は、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 <プール> ミラー <デバイス1> <デバイス2> 特別ミラー <デバイス3>  
'--><デバイス4
```

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

### ユーファイル

`zfs-import-cache.service` は、ZFSキャッシュファイルを使用してZFSプールをインポートします。このファイルが破損すると、サービスはそこから読み取ることができないプールをインポートできなくなります。

その結果、ZFSキャッシュファイルが破損した場合、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 POOLNAME
```

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起動できない場合でも、システムが確実に起動します。

通常の操作では、ESP はマウントされたままにはなりません。これにより、システムクラッシュ時に `vfat` フォーマットの ESP で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` は以下のタスクを処理します：

- 新しいパーティションのフォーマットと設定
- 新しいカーネル・イメージと `initrd` イメージをリストされたすべての 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
- 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

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

る (pinning) 現在設定されているESPとその状態は、以下を実行すること

で見ることができます：

#### 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、以下を実行します：

```
# 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）は、どちらも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

一ネルを削除してください:

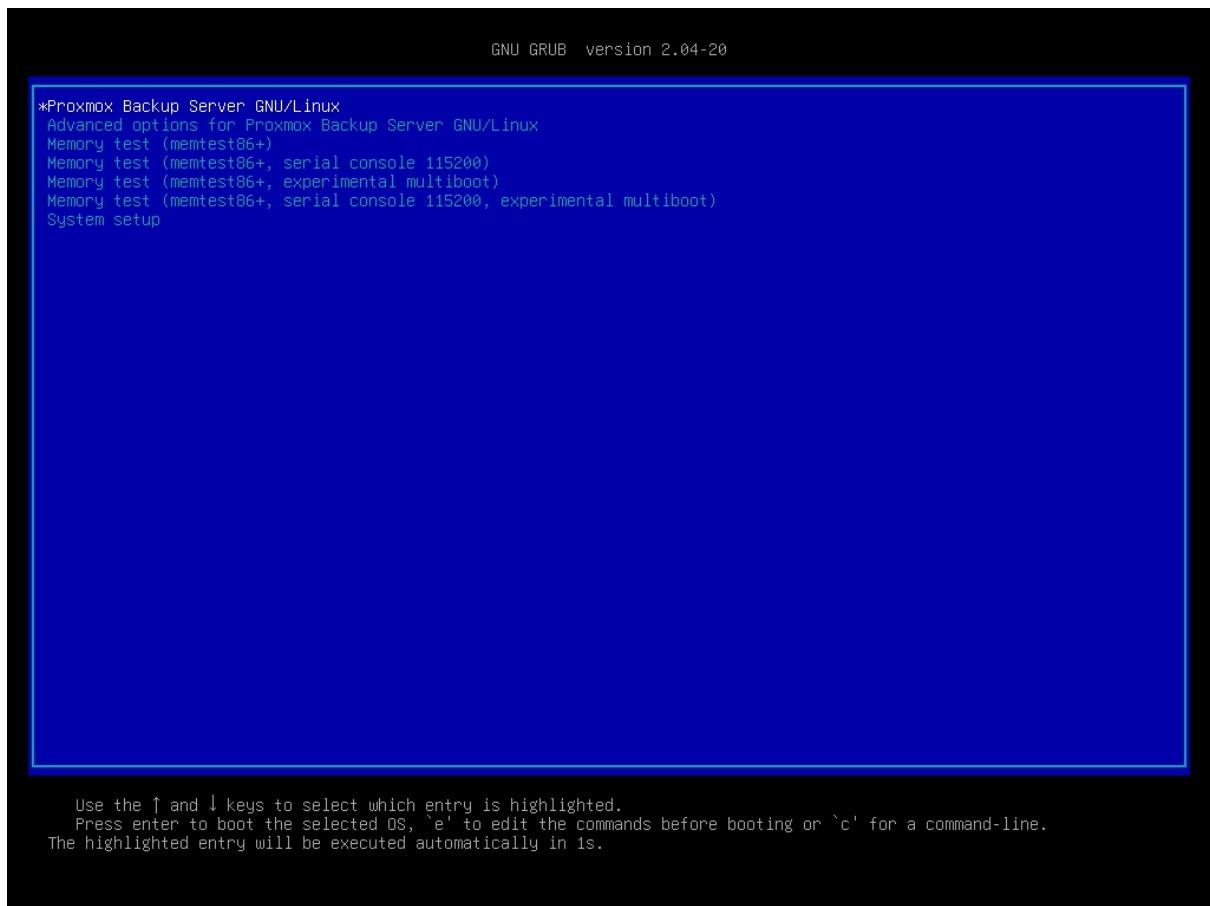
```
# proxmox-boot-tool kernel remove 5.0.15-1-pve
```

---

**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 変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(| 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）：

```
名前: 5.0.15-1-pve
オプション: root=ZFS=rpool/ROOT/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カーネルを選択するには、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だけ、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 上のコンテンツと設定を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しなく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、上記のように必要な署名済みパッケージをすべてインストールします。Grubは、デフォルトの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 /
```

## Proxmox Backup ドキュメント、リリース 3.2.7-1

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が含まれるはずです：

```
ターゲット・ゾーン rpool/R00T/pbs-1 FSTYPEオプション  
└─s  zfs rw,relatime,xattr,noacl  
。
```

次に、適切なESP（EFIシステムパーティション）を見つけなければなりません。これはlsblkコマンドは以下のようになります：

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

出力は次のようになるはずです：

name	maj:min	rm	size	re	type	マウントポイント	fstype
スダ	8:0		032G	0	部	ディスク	
└─sda1	8:1		10 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	8:16		0 32G	0部	部	ディスク	
ース	8:17		0 1007K	0部			
└─sdb1	8:18		0 1007K	0部			

(次ページへ続く)

```
|---sdb2 8: 180 512M 0部 vfat
|---sdb3 8: 190 31.5G 0部 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 を使用)は個々の ESP に対して個別に実行する必要があります:

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

その後、以下のコマンドを実行して、セットアップの正常性をチェックできます:

```
# 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ていれば) カスタムブートエントリとしてファイル `/boot/loader/entries/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`ツールは自動的にキーペアと証明書を`/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でビルト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は証明書と鍵を保存します：

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

証明書には以下のオプションがあります：

1. `etc/proxmox-backup/proxy.pem` にあるデフォルトの自己署名証明書を使用し続けます。
2. 外部から提供された証明書（例えば、商用の認証局（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 Mail Gatewayホストで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。

Upload Custom Certificate

Private Key (Optional):  
No change

From File

Certificate Chain:  
-----BEGIN CERTIFICATE-----  
MIIFBzCCAu+gAwIBAgIUVesRZvSVZKb9pB4O+1ILk1Ft4o0wDQYJKoZIhvcNAQEL  
BQAwezERMA8GA1UEAwvflcG1nLWRlbW8wHhcNMjAwNDIyMTMzOTA1WhcNMzAwNDIw  
MTMzOTA1WiATMREwDwYDVOODAhwbWctZGVtbzCCAiwlDOYJKoZIhvcNAOEBOAD  
From File

Upload

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

Register Account

Account Name: default

E-Mail: admin@example.com

ACME Directory: Let's Encrypt V2

<https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf>

Accept TOS

Register

クラスタごとに、使用するエンドポイントに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。

## **Proxmox Backup ドキュメント、リリース 3.2.7-1**

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

**Add**

Proxmox Backupはこれらのチャレンジタイプの両方をサポートしており、Webインターフェイスの [Certificates] -> [ACME Challenges]、または `proxmox-backup-manager acme plugin add`コマンドを使用してプラグインを設定できます。

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

**Create**

ノードの希望するドメインを設定し、希望する ACME アカウントが選択されていることを確認した後、Web インターフェースで新しい証明書を注文できます。成功すると、インターフェースはおよそ10秒後にリロードされます。

更新は[自動的](#)に行われます。

### 13.3.4 ACME HTTPチャレンジプラグイン

ポート80で生成され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は、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 Encrypts 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.sh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。<sup>1</sup> プロジェクトで開発された DNS プラグインを使用します。特定のAPIの設定の詳細については、そのドキュメントを参照してください。

DNS APIを使用して新しいプラグインを設定する最も簡単な方法は、Webインターフェース（Certificates -> ACME Accounts/Challenges）を使用することです。

ここで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は認証情報用のフォームを自動的に生成しますが、すべてのプロバイダにまだ注釈が付けら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-challenge.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 サーバーが `domain1.example` のすべてのチャレンジを検証できるようにします。

---

<sup>1</sup> 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（2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 × 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 <b>任意の証明書管理必要ありません</b>	119
---	-----

## **Proxmox Backup ドキュメント、リリース 3.2.7-1**

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 サービスデーモン

### 13.4.1 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。より多くのパーミッションを必要とする操作はローカルのproxmox-backupサービスに転送されます。

### 13.4.2 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、すべての特権操作を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、外部から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を外部に公開します。

## 13.5 コマンドラインツール

### 13.5.1 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。

### 13.5.2 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全体をコマンドラインで公開します。

### 13.5.3 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。

### 13.5.4 pmt

pmt コマンドは Linux テープ・デバイスを制御します。

### 13.5.5 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。

### 13.5.6 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ピライされています。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。グロー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.pjar /path/to/source --exclude '**/*.txt' --exclude '!/folder/file.txt'
```

---

注：グローブ・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 `.pxareexclude-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 `.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「[バックアップの作成](#)」で説明したものと同じです。

## アーカイブの抽出

既存のアーカイブ、`archive.pxar` は、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` ファイルをターゲット `/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、アーカイブのルートを基準として、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ポイントでそのコンテンツにアクセスすることができます

。

```
# cd /mnt
# ls
bindev home lib32 libx32      メディア opt   rootsbin sys usr
boot etc      liblib64 lost+found    mntproc run   srvtmp
var
```

### 13.5.7 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。

### 13.5.8 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、ツールは対応するコードを直接呼び出します。

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。



## ネットワーク管理

Name	Type	Active	Autostart	VLAN	Ports/Staves	Bond Mode	CIDR	Gateway	Comment
eno1	Network Device	Yes	No	No					
eno2	Network Device	No	No	No					
ens5f0	Network Device	Yes	Yes	No			192.168.16.120/20	192.168.16.1	
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 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、一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
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   ポート/スレーブ

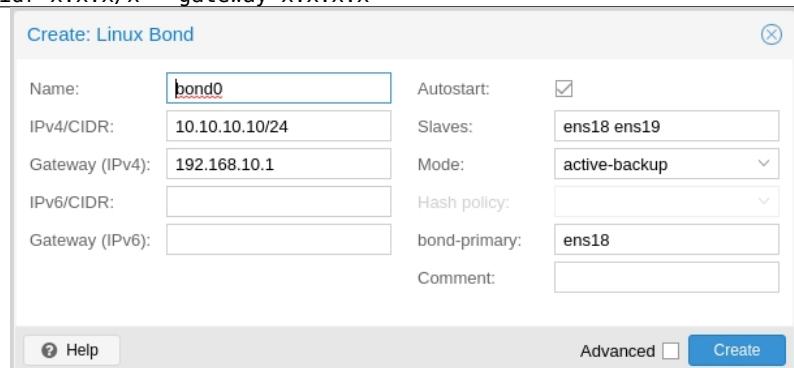
(次ページへ続く)

bond0	ボンド	1   静的	x.x.x.	x/x   x.x.x.	x   ens18	ens19	1   静的
	x.x.x/x   x.x.x.x	ens18 ens19					
ens18	eth	1   マニュアル					
ens19	eth	1   マニュアル					

(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
```

この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、設定ファイルに書き込む場合は、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 checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																	
<input type="button" value="Add"/>		<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。プロ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 \
--rate-in 100MB --rate-out 100MB
```

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。  
および 0.0.0.0/0。

---

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）の設定を次のように上書きします:

```
# 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 | 時間枠 ...
rule0 | 100 MB | 100 MB | ["0.0.0.0/0"] | ["mon.fri ..."]
ルール1 | 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
名前 | cur-rate-in (カーレート・イン) |
ルール0 | rate-out (カーレート・アウト) | 0 B
ルール1 | 1.161 GiB | 19.146 KiB |
```



## 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。
- [通知ターゲット](#)は、マッチャーによって通知イベントがルーティングされる宛先です。ターゲットには複数のタイプがあり、メールベース（SendmailとSMTP）とGotifyがあります。

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、後者には通知ターゲットのパスワードや認証トークンのような機密性の高い設定オプションが含まれており、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 ゴティファイ

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。

すべての設定オプションについては、[notifications.cfg](#) を参照してください。

## 15.3 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-.*$` で始まるデータストアにマッチします。

#### バックアップ

`notification`がマッチしたフィールドを持たない場合、ルールはマッチしません。例えば、`match-field regex:datastore=.*`ディレクティブは、データストアのメタデータフィールドを持つすべての通知にマッチしますが、フィールドが存在しない場合はマッチしません。

### 15.3.3 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、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 窮み しました	エラー		ホスト名
ガベージコレクションの失 敗	ジーシー	エラー	データストア、ホスト名
ガベージコレクション	ジーシー	インフ オメー	データストア、ホスト名

## Proxmox Backup ドキュメント、リリース 3.2.7-1

セス			ション
パッケージの更新 有能	パッケージアップ データ	インフ オメー ション	ホスト名
ブルーンの失敗 ブルーンの仕事の成功	スマモ スマモ	エラー インフ オメー 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
リモート同期の失敗 リモート同期成功	同期 同期	エラー インフ オメー 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
テープバックアップジョブ の失敗	テープバックアッ プ	エラー	データストア、ホスト名、メディアプー ル、 ジョブID
テープバックアップジョブ の成功	テープバックアッ プ	インフ オメー ション	データストア、ホスト名、メディアプー ル、 ジョブID
テープ装填要求	テープロード	お知ら せ	ホスト名
検証ジョブの失敗	ベリフィケーショ ン	エラー	データストア、ホスト名、ジョブID
検証ジョブの成功	ベリフィケーショ ン	インフ オメー ション	データストア、ホスト名、ジョブID

次の表は、すべての使用メタデータ・フィールドの説明です。これらはすべて `match-field` マッチルール。

メタデータフィールド	説明
データストア	データストアの名前
ホスト名	バックアップサーバーのホスト名
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イリータスクは、ノードにアクティブなサブスクリプションがある場合にのみ通知を送信します。

---

## 15.5 システムメール転送

`smartd`などの特定のローカル・システム・デーモンは、ローカル・ルート・ユーザーに通知メールを送信します。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

通知対象の構成を変更/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システムにオプトインします。APIまたは`proxmox-backup-manager` CLI経由でデータストアを作成した場合、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 Serverの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-mode`が`legacy-sendmail`に設定されている場合、Proxmox Backup Serverはシステムの`sendmail`コマンドを使用して、`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。



## 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、blob、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によって参照され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、blob、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

<データストアのルート>/.chunks/.

このチャンクのディレクトリは、チャンクのチェックサムの最初の4バイトで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、ディレクトリあたりのファイル数を減らすために行われます。ディレクトリあたりのファイル数が多いと、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、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は良いアイデアではありません。なぜなら、ファイルのサイズが変更されるたびに、残りのデータが移動され、その結果、多くのチャンクが変更され、重複排除の量が減少するからです。

これを改善するために、[Proxmox Backup Server](#)は代わりに動的なサイズのチャンクを使用します。イメージを固定サイズに分割する代わりに、まず一貫性のあるファイルアーカイブ ([pxar](#)) を生成し、このオンザフライで生成されたアーカイブ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。

## 16.4 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、確率論の「誕生日問題」の考え方を使うことができます。大きな数の場合、これは実際には通常のコンピュータでは計算不可能ですが、良い近似があります：

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

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、 $d = 2^{256}$ 回の可能性があります。これらの値を先ほどの式に插入すると、このシナリオにおける衝突の確率は次のようになります：

$$3.1115 * 10^{-61}$$

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 $1.2277 * 10^{-7}$ 。つまり、衝突する確率は、このような宝くじゲームに13回連続で当選するのとほぼ同じです。

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、チャンクの大きさには上限があるため、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ができないため、

### 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。

### 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）サーバーがデータの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(.*idx*, .*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、.*idx*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.idx
```

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）。

以下の例では、*qemu-server.conf.blob*のデコードされた内容が表示されます。検査しようとしているファイルが暗号化されている場合は、--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ファイルから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_u.  
→b531d3ffc9bd7c65748a61198c060678326a431db7edeb874c327b7986e595e0 --reference-filter /path/  
↳in\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。これはアービトリ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* は復元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ベースにしていますか?

Proxmox Backup ServerはDebian GNU/Linuxをベースにしています。

### 17.2 バックアップソース（クライアント）としてサポートされているプラットフォームは?

クライアントツールはほとんどの最新の Linux システムで動作します。つまり、Debian ベースのバックアップに限定され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のアップグレード、たとえばProxmox Backup Server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ー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場所にコピーまたは同期できますか?

Proxmox Backup Serverでは、リモートと同期ジョブを使用して、データストアを他の場所にコピーまたは同期することができます。リモートとは、ローカルストアと同期可能なデータストアを持つ別のサーバを指します。同期ジョブは、データストアの内容をリモートからローカルのデータストアに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クション](#)を参照してください。

---

## 17.10 バックアップは増分/重複/フルですか?

Proxmox Backup Serverでは、バックアップはサーバに増分的に送信され、データはサーバ上で重複排除されます。これにより、消費されるストレージとネットワークへの影響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の場合はPXAR\_LOG) 環境変数で制御できます。設定可能な値は*off*、*error*、*warn*、*info*、*debug*、*trace* で、*info*がデフォルトです。

---

### A.1 proxmox-backup-client

#### 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
```

### A.2 プロックスバックアップマネージャ

### A.3 プロックスステープ

### 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）。

---

## A.5 pmtx

---

付録

## A.6 ピクサー

## A.7 プロックスファイルリストア

## A.8 proxmox-backup-debug

## 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

以下の役割があります：

### 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; <名前1  
パス <path1>  
オプション  
データストア; <名前2  
パス ...  
<path2>
```

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

## 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ヘッダ <realm-type>: <name> 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/realm/master  
ldap: username_claimユーザー名  
  ldapサーバー  
    base-dn OU=People、DC=ldap-server、DC=example、  
    DC-company mode ldap.10  
    同期属性 email=email  
    sync-defaults-options enable-new=0,remove-vanished=acl;エント  
    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はこのファイルを操作するコマンドを追加します。

### B.3.2 オプション

## B.4 メディアプール

### 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  
プール: 上書き  
...
```

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

# B.5 通知.cfg

## B.5.1 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.5.2 オプション

# B.6 通知非公開.cfg

## B.6.1 ファイル形式

このファイルには、通知ターゲットの保護された認証情報が含まれています。

## B.6.2 オプション

# B.7 tape.cfg

## B.7.1 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。

テープチェンジヤーコンフィギュレーションは、ヘッダーの changer: <name> で始まり、その後にチェンジヤーコンフィギュレーションオプションが続きます。

```
lto: hh8
  チェンジヤー
  パス /dev/tape/by-id/scsi-10WT065325-nst
  チェンジヤー: sl3
  輸出スロット 14, 15, 16
  パス /dev/tape/by-id/scsi-CJ0JBEB0059
```

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。

---

注意：仮想：ドライブ・タイプは実験的なものであり、デバッグにのみ使用してください。

---

## B.7.2 オプション

### B.8 tape-job.cfg

#### B.8.1 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  
  バルp4  
  フォルダ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.8.2 オプション

## B.9 ユーザー設定

### B.9.1 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、ヘッダートークン: <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  
  e-mail  
トークン: test@example.local  
root@pam 对応APIテストトークン  
  有効にする  
  0点  
user: ...
```

このファイルを操作するにはproxmox-backup-managerユーザーコマンドを使用します。

### B.9.2 オプション

## B.10 リモート.cfg

### B.10.1 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 h-id sync@pbs  
リモート server1.local  
...
```

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.10.2 オプション

### B.11 sync.cfg

#### B.11.1 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
sync: rem...
tina
```

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.11.2 オプション

### B.12 verify.cfg

#### B.12.1 ファイル形式

各エントリは、ヘッダverify: <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-
  リモートストア true
  毎日のスケジュール
  リモートストア2
検証: ...
```

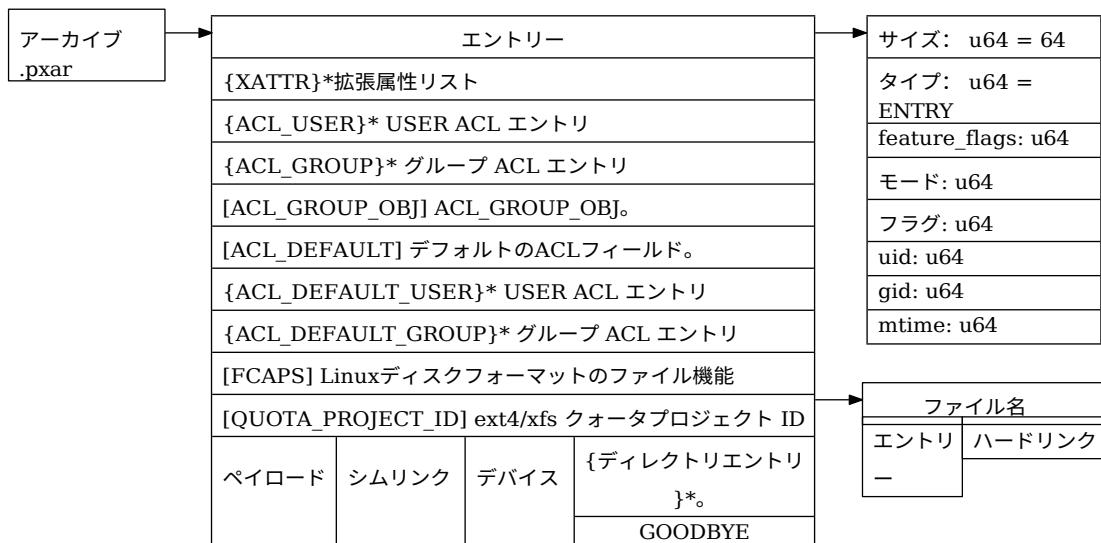
このファイルを操作するにはproxmox-backup-manager verify-jobコマンドを使用します。

## B.12.2 オプション



## ファイル形式

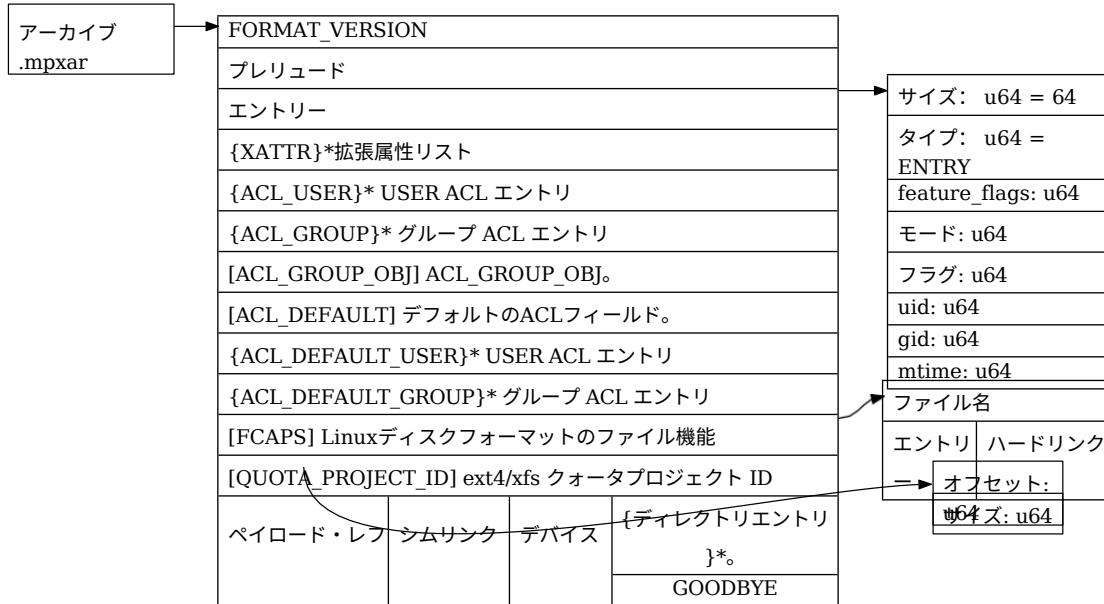
### C.1 Proxmox ファイルアーカイブ形式 (.pxar)



### 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unkを再利用するときに、チャunkの境界がペイロードエントリのオフセットに揃わなかつ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。

KFD	
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] のヘッダは、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。
	アリ
...	ペイロードヘッダ、ペイロード、パディングのさらなる連結
ペイロード・テール・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 データプロブ形式 (.blob)

データプロブフォーマットは、小さなバイナリデータを格納するために使用されます。マジックナンバーが正確なフォーマットを決定します:

[66, 171, 56, 7, 190, 131, 112, 161]	非暗号化	非圧縮
[49, 185, 88, 66, 111, 182, 163, 127]	非暗号化	圧縮
[123, 103, 133, 190, 34, 45, 76, 240]	暗号化	非圧縮
[230, 89, 27, 191, 11, 191, 216, 11]	暗号化	圧縮

圧縮アルゴリズムはzstdを使用。暗号化暗号はAES\_256\_GCM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）、暗号化されたデータが続きます：

---

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; なし 32],	SHA-256 以上 その インデックス 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unkのダイジェスト
ダイジェスト2: [u8; 32]	2つ目のチャunkのダイジェスト
...	次のチャunk・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 ヘッダなし) SHA256(offset1  digest1  offset2  digest2  ...)

---

```
を予約しました: [u8;           全体のヘッダーサイズは1ページ(4096バイト)
4032],  
オフセット1: u64   最初のチャunkの終わり  
digest1: [u8;     最初のチャunkのダイジェスト  
32]  
オフセット2: u64   2つ目のチャunkの終わり  
digest2: [u8;     2つ目のチャunkのダイジェスト  
32]  
...                 次のチャunk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unk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unk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unkをアップロードし、PUT /fixed\_indexでインデックスに追加します。終了したら、POST /fixed\_close でインデックスを閉じる必要があります。

ファイル名は .fidx で終わる必要があり、POST /finish の呼び出しに続いてバックアップマニフェストに自動的に追加されます。

### 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unk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unk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unkをアップロードし、PUT /dynamic\_indexでインデックスに追加します。終了したら、POST /dynamic\_closeでインデックスを閉じる必要があります。

ファイル名は .didx で終わる必要があり、POST /finish の呼び出しに続いてバックアップマニフェストに自動的に追加されます。

### D.1.5 バックアップの終了

すべてのデータをアップロードしたら、POST /finishを呼び出す必要があります。これ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

チャンクはGET /chunkを使用してダウンロードされます。HTTPボディには*Data Blob*としてエンコードされたデータが含まれます。

### D.2.3 インデックスファイルのダウンロード

インデックスファイルはGET /download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、時刻のいずれかの部分がなければならぬことに注意してください。曜日または日付の部分が省略された場合、すべての(週)日が含まれます。時刻の部分が省略された場合、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。)

平日は英語の省略形で指定します： 月、火、水、木、金、土、日。各フィールドには以下

の形式で複数の値を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 8..17,22:0/15	-	毎営業日、午前8時から午後6時までの間、15分ごと 午後10時から11時の間
12日 (金) 13:5/20 12,14,16,18,20, *:*	12,13日(金) ):5/20 12/2:5 0/1:0/1	金曜日12:05、12:25、12:45、13:05、13:25、13:45 毎日12:05から22:05まで2時間毎 毎分（最小間隔）
*-05 土 15:00	-	毎月5日 毎月第1土曜日15:00
2015-10-21	-	2015年10月21日 00:00

## E.2 systemdとの違い

systemd カレンダーア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、カレンダーア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

```
"あなたはそれを組み合わせることができます
使用します。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。

例えば、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 - を使用します:

- \* 項目 1
- \* 項目 2
- \* 項目 2a
- \* 項目 2b

インデントを追加することで、入れ子になったリストを作成できます。

### 注文リスト

1. 項目 1
1. 項目 2
1. 項目 3
1. 項目 3a

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、終了したタスクには x のボックスを使用します。例えば

- [X] 最初のタスクはすでに完了しました!
- [X] 2つ目も
- [ ] これはまだto-do
- [ ] これも

のボックスを使用します。例えば

## F.1.5 テーブル

テーブルは、パイプ記号 | で列を区切り、-で表のヘッダーと本文を区切れます。その区切りで、テキストの配置を設定し、1つの列を左揃え、中央揃え、または右揃えにすることもできます。

左列	右列	一部	その他	中央揃えも有効
左フー	右			foo 最初の 行 ここ >center<

左バー	右バズ	バー   2行目   ここ   12345	
左ザブ	右	3列目   ここ   テスト	
左狂犬病	右	ザーブ   4列目   ここ   <del>aaaa</del>   右ザーブ ラブ   そして   最後   ここで   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。

## 付録

## 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・スワードと共に作成。  
>> 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われます。  
'-フォーラム  
>> 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。

### インラインコード

行の一部をバックステイックで囲むと、例えばインラインでコードを書くことができます：

```
このホストのIPアドレスは`10.0.0.1`です。
```

### コードブロック全体

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・バックス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  
    address 10.0.0.1/24  
    bridge-ports ens20  
    bridge-stp off  
    bridge-fd 0  
    bridge-vlan-aware  
    yes bridge-vids 2-  
4094
```





## 用語集

### 仮想マシン

仮想 マシンは、エミュレートされたハードウェア環境内でオペレーティング・システム全体を実行できるプログラムです。

### コンテナ

コンテナは、隔離されたユーザー空間です。プログラムはホストのカーネル上で直接実行されますが、ホストのリソースへのアクセスは制限されます。

### 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モデルは、メモリ安全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されました。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インストールとそのユーザの認証情報。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

著作権 (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できません。

### 0. 前文

本許諾書の目的は、マニュアルや教科書、あるいはその他の機能的で有用な文書を、自由という意味で「自由」にすることです。つまり、商業的であれ非商業的であろうと、改变の有無に関わらず、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、他者による改变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身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、ソフトウェアと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身となる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「セカンダリ」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特定の短い文章のことです。表表紙テキストは最大5語、裏表紙テキストは最大25語です。

透明な(Transparent)』『文書』のコピーとは、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(図面については)広く利用可能な何らかの図面エディタを用いて文書を簡単に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。そうでなければ透過的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場合、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過的ではありません。画像フォーマットは、かなりの量のテキストに使われている場合、「透過的」ではありません。透明」でないコピーは「不透明」と呼ばれます。

トランスペアレントコピーに適したフォーマットの例としては、マークアップのないプレーンなASCII、Texinfo入力フォーマット、LaTeX入力フォーマット、一般に公開されているDTDを使ったSGMLやXML、人間が修正できるように設計された標準準拠のシンプルなHTML、PostScript、PDFなどがあります。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、プロプライエタリなワードプロセッサーによってのみ読み取りや編集が可能なプロプライエタリなフォーマット、DTDや処理ツールが一般的に利用できないSGMLやXML、一部のワードプロセッサーが出力のみを目的として生成する機械生成のHTML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、著作物のタイトルの最も顕著な出現部分に近い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を他の言語で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、「謝辞」、「献辞」、「裏書」、「歴史」など、後述の特定のセクション名を表します)。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。これらの保証の否認は本許諾書に参考として含まれるものとみなされますが、それは保証の否認に関してのみです。

## 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追加しないことを条件として、『文書』を商業的であるか非商業的であるかを問わず、いかなる媒体にも複製および頒布することができます。あなたは、あなたが作成または頒布した複製物の閲覧や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』のライセンス告知がCover Texts(表紙文)を要求している場合、あなたはその複製物を、以下のCover Texts(表紙文)の全てをはっきりと読みやすく記載した表紙に封入しなければなりません：表紙には「Front-Cover Texts」、裏表紙には「Back-Cover Texts」。また、両表紙とも、あなたがこれらのコピーの発行者であることを明確かつ判読しやすいように表示してください。表表紙は、タイトルの全単語を等しく目立たせ、見えるようにしなければなりません。また、表紙に他の素材を加えてもかまいません。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』のタイトルとも、旧版のタイトル（旧版がある場合は、『文書』の「履歴」セクションに記載されているはずです）とも異なるタイトルを使用してください。旧版の発行元が許可している場合は、旧版と同じタイトルを使用してもかまいません。
- B. タイトル・ページには、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物または団体を、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

---

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』それ自体よりも少なくとも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を与えた場合には、その著作物のネットワーク上の位置を省略することができます。
- K. 謝辞」または「献辞」と題されたセクションについては、そのセクションのタイトルを保持し、そ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. 文書のすべての不变セクションを、本文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は、それぞれ1つの主体によって（または主体による取り決めによって）1つの箇所のみ追加することができます。もし『文書』に同じ表紙のカバーテキストが既に含まれており、それが以前にあなたによって追加されたものである場合、あるいはあなたが代行する同じ団体によって手配されたものである場合、あなたは別のカバーテキストを追加することはできません。

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、あなたがその結合国家に、改変されていないすべての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(もし分かっていれば)、あるいは一意な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、「謝辞」と題されたセクションと「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』および本許諾書の下でリリースされたその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、それ自体が『文書』の二次的著作物ではないものには適用されない。

第3項のカバーテキストの要件が当該文書のコピーに適用される場合、当該文書が総体全体の2分の1未満であれば、当該文書のカバーテキストは、総体内で当該文書を囲むカバー、または当該文書が電子形式であればカバーに相当する電子表紙に掲載することができます。それ以外の場合は、総体全体を囲む印刷された表紙に掲載しなければなりません。

#### 8. 翻訳

翻訳は一種の改変とみなされますので、あなたは第4節の条件のもとで、この文書の翻訳を頒布することができます。変更不可部分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、および保証免責条項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、およびこれらの表示と免責条項の原文も含めるものとします。翻訳と本許諾書の原版、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停止しない限り、暫定的に、(b)著作権者が停止後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、あなたがその著作権者から本許諾書に対する違反の通知(いかなる作品についても)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日以内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ものではありません。

## 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、バージョン番号が付けられています。文書』において、本許諾書の特定のバージョン番号「またはそれ以降のバージョン」が適用されると指定されている場合、あなたはその指定されたバージョンか、フリーソフトウェア財団によって(草案としてではなく)発行されたそれ以降のバージョンのいずれかの条項と条件に従うという選択肢を持つことになります。文書』に本許諾書のバージョン番号が明記されていない場合、あなたはフリーソフトウェア財団によって(草案としてではなく)公表されたどのバージョンでも選ぶことができます。文書』において、代理人が本許諾書の将来のどの版を使用できるかを決定できる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 Multiauthor 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MMCを再公開することができます。

## インデックス

### C

コンテナ, **161**

### D

データストア, **161**

### F

ヒューズ, **161**

### R

リモート, **161**

reStructuredText, **161**

さび, **161**

### S

スフィンクス, **161**

### V

仮想マシン、**161**