Proxmox VE to Proxmox Backup Server (PBS) Mechanism Documentation

September 2, 2025

Contents

1 Overview of Proxmox VE to PBS Backup Mechanism

This document provides a detailed explanation of how Proxmox Virtual Environment (VE) backs up Containers (CTs) and Virtual Machines (VMs) to Proxmox Backup Server (PBS). It covers the mechanisms, deduplication, incremental backups, block-level operations, and the role of SHA256 hashing, based on a session log from September 2, 2025.

2 Backup Mechanisms for VMs and Containers

Proxmox VE employs distinct mechanisms for backing up VMs and CTs to PBS, tailored to their underlying technologies.

2.1 Virtual Machines (VMs) QEMU/KVM

- **Mechanism**: Utilizes the QEMU backup API to stream block-level data from virtual disks.
- **Process**: The API accesses the VM's disk at the block level, capturing data in chunks for efficient transfer.

2.2 Containers (CTs) LXC

- **Mechanism**: Uses the `vzdump` tool to create snapshots, suspend, or stop containers.
- **Process**: Streams filesystem data (e.g., container root filesystem) to PBS.

3 Backup Process Details

The backup process to PBS involves several key steps to ensure efficiency, security, and integrity.

3.1 Chunking

- Disk data (for VMs or CTs) is split into fixed-size chunks, typically 4 MB.
- Each chunk is processed independently for further operations.

3.2 Hashing with SHA256

- Each chunk is hashed using the SHA256 algorithm to generate a unique identifier.
- **Purposes**:
  - **Deduplication**: Identical chunks (same hash) are stored only once.
  - **Integrity**: Hashes are verified during restore to detect corruption.
  - **Incremental Efficiency**: Chunks with existing hashes are skipped during backups.
  - **Security**: Supports client-side encryption by ensuring data integrity.

3.3 Deduplication

- **Across Backups**: Identical chunks within multiple backups of the same VM/CT are stored once.

- **Across VMs/CTs**: Common data (e.g., identical `/bin/bash` in two VMs) is stored only once.
- **Example**: Two VMs with the same 4 MB chunk of `/bin/bash` produce identical SHA256 hashes, resulting in a single copy stored on PBS.

3.4 Incremental Backups

- Only changed chunks (new or modified data) are uploaded after the initial full backup.
- PBS maintains an index of chunks to track which are needed for each backup.

3.5 Compression and Transfer

- Chunks are compressed using the Zstandard algorithm to reduce storage and network usage.
- Data is transferred over a TLS-encrypted connection to PBS.
- Optional client-side encryption can be applied for additional security.

4 Example: Backup and Deduplication Scenario

Consider a VM with a 20 GB disk. The following table illustrates the backup process over three days, assuming deduplication and incremental backups.

Backup Run	VM Disk Size	Changed Data	Uploaded to PBS	Stored on PBS
Day 1	20 GB	20 GB	20 GB	20 GB
Day 2	20 GB	2 GB	2 GB	22 GB
Day 3	20 GB	500 MB	500 MB	22.5 GB

Table 1: Example of incremental backups with deduplication.

4.1 Explanation of the Example

- **Day 1**: Full backup uploads all 20 GB as no prior data exists.
- **Day 2**: Only 2 GB of changed data is uploaded, leveraging incremental backup.
- **Day 3**: Only 500 MB of changed data is uploaded, further reducing transfer and storage needs.
- **Storage Efficiency**: Deduplication ensures identical chunks (e.g., unchanged OS files) are not stored redundantly.

5 Restore Process

To restore a VM or CT:

- PBS uses the chunk index to identify and retrieve necessary chunks.
- Chunks are decompressed and reassembled into the original disk or filesystem.
- Hashes are verified to ensure data integrity during restoration.

6 Benefits of the Proxmox VE to PBS Backup Mechanism

- **Storage Efficiency**: Deduplication minimizes storage requirements.
- **Network Bandwidth Savings**: Incremental backups reduce data transfer.
- **Data Integrity**: SHA256 hashing ensures reliable verification.
- **Security**: TLS and optional client-side encryption protect data.

7 Conclusion

The Proxmox VE to PBS backup mechanism leverages advanced techniques like block-level chunking, SHA256 hashing, deduplication, incremental backups, and compression to provide an efficient, secure, and reliable backup solution. This approach optimizes storage and network resources while ensuring data integrity and security.