**SnapSend & DelSnap - Automating ZFS Snapshots & Cleanup in servers eg. Proxmox cluster**

**Introduction**

Efficient data management and backup solutions are critical in virtualized environments. **SnapSend & DelSnap** is a powerful script-based solution designed for **automating ZFS snapshots and cleanup**. It integrates seamlessly with **Proxmox**, ensuring reliable backups, snapshot synchronization, and lifecycle management.

This article explores how SnapSend (snapsend.sh) and DelSnap (delsnaps.sh) streamline **ZFS snapshot creation, transfer, and cleanup** to maintain optimal storage efficiency and redundancy.

**Key Features**

* **Automated ZFS snapshot creation** – With configurable schedules using cron.
* **Efficient incremental snapshot synchronization** – Sending only changed data between nodes.
* **Snapshot pruning policies** – Automatically remove outdated snapshots based on retention rules.
* **Multi-mode operation** – Supports local and remote backups.
* **Proxmox Cluster support** – Ensures VM dataset synchronization between nodes.
* **Verbose logging and dry-run mode** – Helps troubleshoot potential issues before execution.

**How SnapSend Works**

**Modes of Operation**

SnapSend (snapsend.sh) operates in **four different modes**, providing flexibility for different backup strategies:

1. **only.snapshot** – Creates a local snapshot without transferring it.
2. **local.backup** – Stores a snapshot in a secondary local dataset.
3. **remote.synchro** – Syncs snapshots to a remote server.
4. **remote.backup** – Transfers snapshots to a remote dataset on another node.

**Example Use Cases**

**Scenario 1: Proxmox Cluster Backup**

* **pve1** hosts virtual machines (VMs) and their disk images.
* **pve2** is a backup node with additional HDD storage.
* snapsend.sh ensures **hourly snapshot synchronization** between pve1 and pve2.
* delsnaps.sh automatically deletes expired snapshots based on retention policies.

**Setup Guide**

**Installing the Scripts**

To integrate SnapSend & DelSnap with your Proxmox environment:

1. Clone the repository:

git clone https://github.com/AdalbertKing/ZFS-CLONE-THEM-ALL-2.0.git

1. Move the scripts to an appropriate directory:
2. mv ZFS-CLONE-THEM-ALL-2.0/snapsend.sh /usr/local/bin/
3. mv ZFS-CLONE-THEM-ALL-2.0/delsnaps.sh /usr/local/bin/

chmod +x /usr/local/bin/snapsend.sh /usr/local/bin/delsnaps.sh

**Configuring SnapSend**

To automate snapshot creation, add a cron job in /etc/crontab:

37 \* \* \* \* /root/scripts/zfs-snapshot-all/snapsend.sh -m automated\_hourly\_ -z -v 3 rpool/data/vm-106-disk-0,hdd/vm-disks/subvol-101-disk-0 2>>/root/scripts/cron.log

This runs SnapSend **every hour**, creating snapshots with the prefix automated\_hourly\_.

**Configuring DelSnap**

To ensure old snapshots are deleted, add another cron job:

51 \* \* \* \* /root/scripts/zfs-snapshot-all/delsnaps.sh -R hdd/vm-disks,rpool/data automated\_hourly -h24 2>>/root/scripts/cron.log

This command removes **hourly snapshots older than 24 hours**.

**Advanced Usage**

**Incremental Snapshot Synchronization**

To enable incremental snapshot synchronization between nodes, run:

snapsend.sh -m automated\_ -e -I -F -z -v 3 rpool/data hdd/backups

* -I ensures incremental transfers.
* -F forces full synchronization if needed.
* -z enables compression for faster transfers.
* -e processes the latest snapshot without creating a new one.

**Debugging with Dry-Run Mode**

Before executing changes, simulate the operation using -n:

snapsend.sh -n -e -r rpool/test 192.168.11.11:

This outputs the expected actions without modifying datasets.

**Conclusion**

SnapSend & DelSnap provide a **comprehensive solution** for **ZFS snapshot automation** in **Proxmox environments**. By leveraging **cron-based automation**, incremental synchronization, and snapshot lifecycle management, this tool ensures **efficient storage utilization and data redundancy**.

By implementing **SnapSend & DelSnap**, system administrators can **reduce backup complexity**, **improve disaster recovery strategies**, and **enhance Proxmox data resilience**.

🚀 **Get started now**: [GitHub Repository](https://github.com/AdalbertKing/ZFS-CLONE-THEM-ALL-2.0)