

Sao lưu với Proxmox Backup Server

- [Tổng quan về Proxmox Backup Server](#)
- [Cài đặt PBS](#)
- [Sao lưu với PBS \(Simple\)](#)
- [Sao lưu với PBS \(Advance\)](#)

Tổng quan về Proxmox Backup Server

Proxmox Backup Server (PBS) là gì?

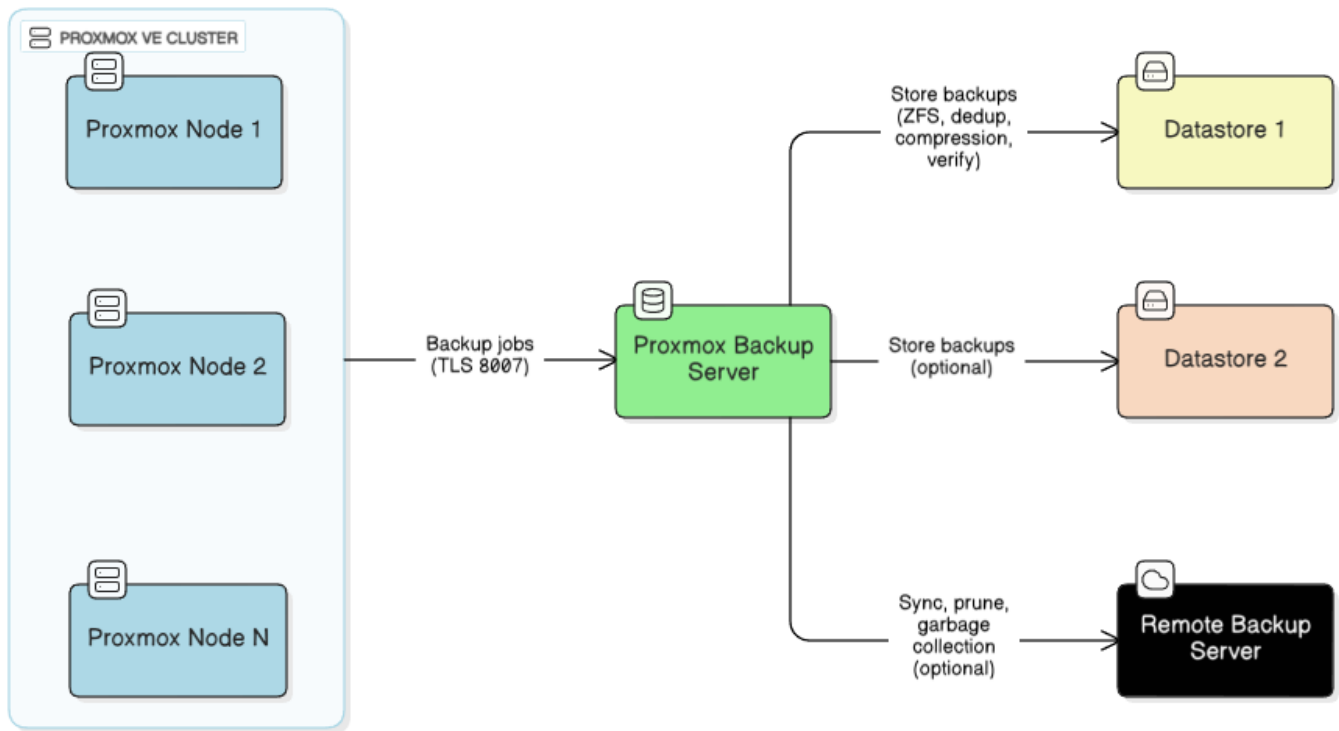
PBS là phần mềm sao lưu cấp doanh nghiệp do đội ngũ **Proxmox Server Solutions GmbH** phát triển. Nó được thiết kế để sao lưu và khôi phục VM, CT, Host và cả các thư mục, file riêng lẻ.

PBS là một hệ thống chuyên dụng đảm nhiệm vai trò máy chủ lưu trữ cho phép nhiều máy khách (PVE) kết nối và gửi bản sao lưu đến thông qua giao thức HTTPS

PBS hoàn toàn miễn phí với giấy phép mã nguồn mở (AGPLv3). Cũng có phiên bản trả phí với hỗ trợ đến từ đội ngũ.

Cách PBS hoạt động.

PBS không tự theo dõi hay sao lưu các PVE mà hoạt động theo mô hình Client - Server:



 eraser

- Các PVE tạo và gửi backup cho PBS
- PBS lưu, kiểm tra và đồng bộ dữ liệu
- PBS lưu dữ liệu vào các datastore

Mức độ tiêu thụ tài nguyên.

Môi trường sử dụng	CPU	RAM	Ổ lưu trữ	Ghi chú
Homelab (3-5 VM)	2 cores	4 GB	SSD hoặc NFS	Rất mượt
Trung bình (10-15 VM)	4 cores	8-12 GB	SSD/NAS 1 TB	Ổ định, CPU dùng ~30-50 % khi backup
Nhiều VM (>20)	6-8 cores	16 GB+	NVMe / ZFS RAID	Cần tối ưu song song & dedup metadata

Lưu ý

PBS không cần chạy liên tục 24/7 chỉ cần bật khi có job backup hoặc sync

Có thể dùng cron hoặc schedule để:

- Bật VM PBS trước khi backup

- Chạy backup job
- Tự động shutdown sau khi PBS chạy xong

Cài đặt PBS

Trong trường hợp sử dụng này sẽ thực hiện backup với những yếu tố sau

- Chạy PBS như một **máy ảo** trên PVE
- PBS sẽ lưu dữ liệu vào ổ **NFS trên Synology DSM**

Setup trên Synology DSM

- Tại shared folder trên DSM
- Bật NFS
- Cấp quyền NFS cho Shared Folder

Cài đặt VM PBS

- Tải ISO của PBS
- Tạo VM PBS
 - System:
 - Machine: q35
 - BIOS: OVMF (UEFI)
 - CPU:
 - Cores: 2
 - Type: host
 - Memory: 4096 MiB
 - Network:
 - Bridge: vmbr0
 - Model: VirtIO
- Chạy VM PBS và cài đặt trên giao diện Graphic bình thường.

Cài đặt NFS cho PBS

- Sử dụng giao diện console trên VM hoặc kết nối tới VM bằng ssh
- Cập nhật bằng `apt update` và `apt upgrade`
- Chạy `apt install nfs-common` để đảm bảo nfs đã được cài đặt
- Tạo thư mục để mount `mkdir /mnt/dsm`
- Thực hiện mount và kiểm tra
 - `mount -t nfs <dsm_local_ip>:<nfs_folder> <pbs_mount_folder>`
 - EX: `mount -t nfs 192.168.1.100:/volume1/pbs_storage /mnt/xpenology`

- Chạy `df -h` và kiểm tra xem có thư mục vừa mount chưa.
- Mount tự động
 - Sửa file `fstab` bằng lệnh sau `nano /etc/fstab`
 - Thêm dòng sau vào cuối `<dsm_local_ip>:<nfs_folder> <pbs_mount_folder> nfs defaults,_netdev 0 0`
- Tạo Datastore
 - Truy cập giao diện web của PBS `localIP:8007`
 - Chọn Datastore > Add datastore
 - Name: <Tên bất kỳ>
 - Backing path: thư mục được mount ở trên `/mnt/dsm`
 - **Add**

Kết nối PVE với PBS

- Lấy Fingerprint:
 - Trên PBE > Dashboard > Show Fingerprint
- Tạo Storage trên PVE: Datacenter > Storage > Add > Proxmox Backup Server
 - ID: Tên bất kỳ
 - Server: IP của PBS
 - User: root@pam
 - Datastore: Tên datastore vừa tạo ở bước trên
 - Fingerprint: fingerprint vừa tạo ở trên
- **Add**

Kiểm tra

- Chọn một VM/CT bất kỳ trên PVE (nên chọn VM/CT có kích thước nhỏ)
- Chọn Backup > Backup now
 - Storage: chọn storage vừa tạo ở bước trên
 - Mode: Snapshot
- **Backup**

Nếu thành setup thành công bạn sẽ thấy console log chạy và có task backup chạy trên PBS

Fix 1

Khi chạy apt install nếu gặp lỗi như hình dưới có thể thử cách sau

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@pbs:~# apt update
Get:1 http://security.debian.org/debian-security trixie-security InRelease [43.4 kB]
Get:2 http://deb.debian.org/debian trixie InRelease [140 kB]
Get:3 http://security.debian.org/debian-security trixie-security/main amd64 Packages [69.3 kB]
Get:4 http://deb.debian.org/debian trixie-updates InRelease [47.3 kB]
Get:5 http://security.debian.org/debian-security trixie-security/main Translation-en [45.2 kB]
Get:6 http://deb.debian.org/debian trixie/main amd64 Packages [9,669 kB]
Get:7 http://security.debian.org/debian-security trixie-security/non-free-firmware amd64 Packages [544 B]
Get:8 http://security.debian.org/debian-security trixie-security/non-free-firmware Translation-en [352 B]
Get:9 http://deb.debian.org/debian trixie/main Translation-en [6,484 kB]
Get:10 http://deb.debian.org/debian trixie/contrib amd64 Packages [53.8 kB]
Get:11 http://deb.debian.org/debian trixie/contrib Translation-en [49.6 kB]
Get:12 http://deb.debian.org/debian trixie/non-free-firmware amd64 Packages [6,868 B]
Get:13 http://deb.debian.org/debian trixie/non-free-firmware Translation-en [4,704 B]
Get:14 http://deb.debian.org/debian trixie-updates/main amd64 Packages [5,412 B]
Get:15 http://deb.debian.org/debian trixie-updates/main Translation-en [4,096 B]
Err:16 https://enterprise.proxmox.com/debian/pbs trixie InRelease
401 Unauthorized [IP: 2402:1f00:8001:f7a::65 443]
Error: Failed to fetch https://enterprise.proxmox.com/debian/pbs/dists/trixie/InRelease 401 Unauthorized [IP: 2402:1f00:8001:f7a::65 443]
Error: The repository 'https://enterprise.proxmox.com/debian/pbs trixie InRelease' is not signed.
Notice: Updating from such a repository can't be done securely, and is therefore disabled by default.
Notice: See apt-secure(8) manpage for repository creation and user configuration details.
```

- Chạy nano /etc/apt/sources.list.d/pbs-enterprise.sources để mở file source
 - Mở file sẽ thấy nội dung dạng như sau
- ```
Types: deb
URIs: https://enterprise.proxmox.com/debian/pbs
Suites: trixie
Components: pbs-enterprise
```
- Thêm dòng Enabled: no vào cuối rồi save lại
  - Chạy nano /etc/apt/sources.list.d/pbs-no-subscription.sources để thêm source **no-subscription**
  - Điền nội dung sau rồi lưu lại

```
Types: deb
URIs: http://download.proxmox.com/debian/pbs
Suites: trixie
Components: pbs-no-subscription
```

# Fix 2



# Sao lưu với PBS (Simple)

Sao lưu PVE sẽ chia làm 2 phần.

- Sao lưu VM/CTs: Được PBS hỗ trợ nên có thể thực hiện đơn giản trên GUI
- Sao lưu cấu hình PVE: Phần này phải sử dụng script

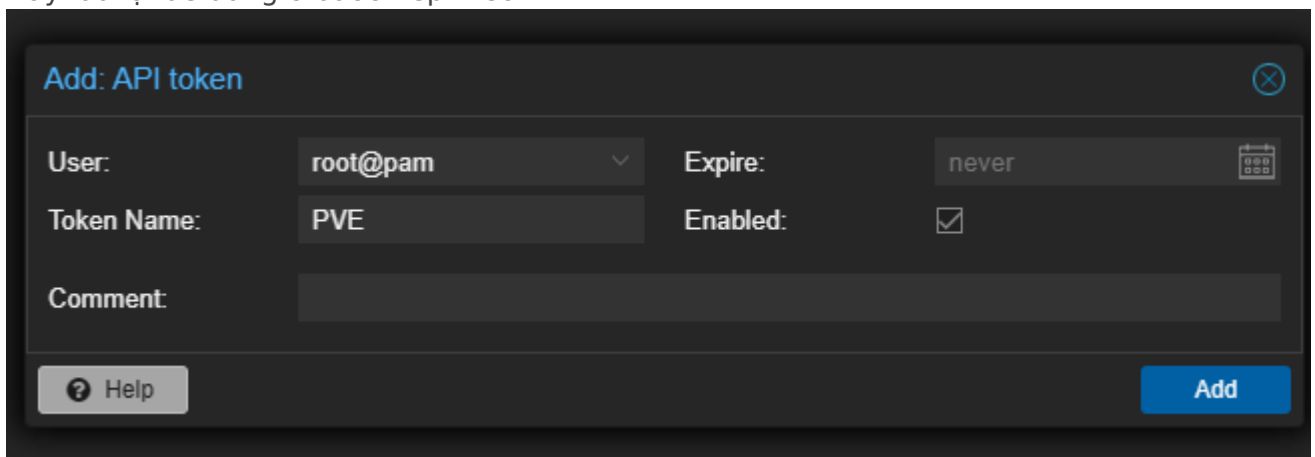
## Sao lưu cấu hình PVE

Vì PVE không có chức năng backup cấu hình nên chúng ta sẽ cần sử dụng script và dùng **systemd timer** để lên lịch backup tự động.

### Backup script

Trước khi bắt đầu với script chúng ta cần chuẩn bị 2 thứ là PBS API Token và Fingerprint

- Fingerprint
  - PBS Web > Dashboard > Show fingerprint
- API Token
  - PBS Web > Configuration > Access Control > API Token
  - Nếu chưa có hãy thêm mới. Sau khi nhấn **Add** bạn sẽ nhận được Token ID và Secret. Hãy lưu lại để dùng ở bước tiếp theo.



The screenshot shows a web form titled "Add: API token". It contains the following fields and controls:

- User:** A dropdown menu with "root@pam" selected.
- Token Name:** A text input field containing "PVE".
- Expire:** A dropdown menu with "never" selected, accompanied by a calendar icon.
- Enabled:** A checkbox that is checked.
- Comment:** A large text area for additional notes.
- Buttons:** A "Help" button with a question mark icon and an "Add" button.

Tạo script backup bằng lệnh `nano /usr/local/bin/pve-host-backup.sh` với nội dung sau. Điền thông tin đã chuẩn bị vào phần Cấu hình.

```
#!/usr/bin/env bash
set -euo pipefail

#####
C?U H?NH PBS
#####
IP/FQDN PBS và Datastore
```

```

PBS_HOST='PBS IP'
PBS_DATASTORE='Datastore'

PBS_USER_TOKEN='Token ID'
PBS_TOKEN_SECRET='API Token Secret'
PBS_FINGERPRINT='Fingerprint'

#####
TU? CH?N CHUNG
#####
BACKUP_ID="${BACKUP_ID:-"$(hostname)-hostcfg"}"
DRY_RUN="${DRY_RUN:-0}"

export PBS_REPOSITORY="${PBS_USER_TOKEN}@${PBS_HOST}:${PBS_DATASTORE}"
export PBS_PASSWORD="${PBS_TOKEN_SECRET}"
export PBS_FINGERPRINT="${PBS_FINGERPRINT}"

#####
NGU?N BACKUP (CH? TH? M?C, KHÔNG FILE L?)
#####
SOURCES=(
 "pve-etc.pxar:/etc" # Bao tr?n: pve, network, hosts, resolv.conf, hostname,
apt, ssh, default, kernel, modules, modprobe.d, sysc>
 "pve-cron.pxar:/etc/cron.d" # Cron.d
 "pve-crontabs.pxar:/var/spool/cron/crontabs" # T?t c? crontab user (bao g?m root)
 "pve-root-home.pxar:/root" # Script & key qu?n tr? (cân nh?c exclude b?t)
 "pve-usrlocal.pxar:/usr/local" # Script/service t? thêm (n?u có)
)

Lo?i tr? cho nh? (tu? b?n ch?nh)
EXCLUDES=(
 "--exclude=/root/*.iso"
 "--exclude=/root/*.img"
 "--exclude=/root/.cache"
)

#####
TI?N KI?M TRA
#####
echo "[*] Checking PBS login..."
proxmox-backup-client login --repository "${PBS_REPOSITORY}"

#####
TH?C HI?N BACKUP
#####
echo "[*] Starting host-config backup (backup-id=${BACKUP_ID})"
CMD=(proxmox-backup-client backup)
CMD+=("${SOURCES[@]}")
CMD+=("${EXCLUDES[@]}")
CMD+=("--backup-id "${BACKUP_ID}")
CMD+=("--backup-type host)

if [["${DRY_RUN}" == "1"]]; then
 CMD+=("--dry-run)
 echo "[*] DRY RUN enabled"
fi

In câu l?nh (?n secret) r?i ch?y
echo "[*] Running: ${CMD[*]}/${PBS_TOKEN_SECRET}/**SECRET**}"
"${CMD[@]}"

echo "[?] Backup finished"

D?n bi?n nh?y c?m
unset PBS_PASSWORD

```

Cấp quyền thực thi cho script `chmod +x /usr/local/bin/pve-host-backup.sh`

Dùng lệnh `DRY_RUN=1 /usr/local/bin/pve-host-backup.sh` để kiểm tra. Nếu kết thúc bằng `[?] Backup finished` tức là thành công.

Dùng lệnh `/usr/local/bin/pve-host-backup.sh` để chạy script

## L?p l?ch t? ??ng

Thực hiện trên console hoặc ssh của PVE

- Tạo file `.service`. Đây là hành vi sẽ được thực hiện bởi `.timer`

- `nano /etc/systemd/system/pve-host-backup.service`

- Sử dụng nội dung sau

```
[Unit]
Description=Backup PVE host configuration to PBS
After=network-online.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/pve-host-backup.sh
```

- Tạo file `.timer`. Nó sẽ thực thi `.service` theo thời gian được chỉ định.

- `nano /etc/systemd/system/pve-host-backup.timer`

- Sử dụng nội dung sau

```
[Unit]
Description=Schedule PVE host config backup (Tue, Thu, Sat at 2:00)

[Timer]
OnCalendar= Tue, Thu, Sat 2:00
Persistent=true
AccuracySec=1min

[Install]
WantedBy=timers.target
```

- Kích hoạt timer

- Reload systemd `systemctl daemon-reload`

- Kích hoạt timer `systemctl enable --now pve-host-backup.timer`

- Kiểm tra

- `systemctl list-timers | grep backup`

- `systemctl status pve-host-backup.timer`

- Kết quả tương tự như hình sau

```
root@pve:~# systemctl list-timers | grep backup
Tue 2025-11-11 05:00:00 +07 3h 54min Sun 2025-11-09 03:43:53 +07 - backup.timer backup.service
Wed 2025-11-12 00:00:00 +07 22h Tue 2025-11-11 00:00:43 +07 1h 4min ago dpkg-db-backup.timer dpkg-db-backup.service
root@pve:~# systemctl status backup.timer
● backup.timer - Schedule PVE host config backup (Tue, Thu, Sat at 5:00)
 Loaded: loaded (/etc/systemd/system/backup.timer; enabled; preset: enabled)
 Active: active (waiting) since Tue 2025-11-11 00:55:26 +07; 10min ago
 Invocation: 3a68e6b2d359453b954eb86930c11eac
 Trigger: Tue 2025-11-11 05:00:00 +07; 3h 54min left
 Triggers: ● backup.service

Nov 11 00:55:26 pve systemd[1]: Started backup.timer - Schedule PVE host config backup (Tue, Thu, Sat at 5:00).
```

## Kết quả kiểm tra

```
Mon 2025-11-10 00:00:00 +07 20h Sun 2025-11-09 00:00:43 +07 3h 47min ago dpkg-db-backup.timer dpkg-db-backup.service
```

```
Tue 2025-11-11 02:00:00 +07 1 day 22h - - backup.timer backup.service
```

# T? ??ng Verify sau khi backup

Chuẩn bị ssh key trên PVE

- Tạo **cặp khóa SSH (public/private key)** cho tài khoản `root` trên máy PVE  
`ssh-keygen -t ed25519 -N "" -f /root/.ssh/id_ed25519`
- Sao chép public key (file .pub) từ máy PVE sang máy PBS, để PBS cho phép `root@PVE` đăng nhập mà không cần mật khẩu.  
`ssh-copy-id -i /root/.ssh/id_ed25519.pub root@192.168.1.104`  
Lần đầu chạy sẽ yêu cầu nhập đúng mật khẩu.
- Để kiểm tra hãy thử truy cập `ssh root@192.168.1.104` nếu không yêu cầu mật khẩu là thành công.

Thêm code `post-verify` vào cuối script backup trên.

```
==== POST: VERIFY DATASTORE TRÊN PBS SAU KHI BACKUP THÀNH CÔNG ====
if [["${DRY_RUN}" != "1"]]; then
 echo "[*] Trigger verify on PBS for datastore '${PBS_DATASTORE}' ..."
 # Ch?y verify toàn datastore. M?c ??nh --ignore-verified=true nên ch? verify snapshot m?i/ch?a verified
 # ho?c snapshot ?ã quá h?n theo --outdated-after
 ssh -o BatchMode=yes root@"${PBS_HOST}" \
 "proxmox-backup-manager verify '${PBS_DATASTORE}' --ignore-verified true --outdated-after 7" \
 && echo '[?] Verify job started on PBS' \
 || echo '[!] Failed to start verify job on PBS'
fi
```

## Sao lưu VM/CTs

Mục tiêu là lên lịch để PVE tự động chạy backup mỗi thứ 3, 5, 7 vào lúc 4h và thực hiện verify sau đó vào lúc 5h

### Lên l?ch backup

- PVE web > Datacenter > Backup > Add
  - Storage: Chọn PBS đã thêm ở bước cài đặt
  - Schedule: Điền *tue,thu,sat 04:00* để chọn thứ 3, 5, 7 vào lúc 4h. Thay đổi tùy ý
  - Selection mode: All (hoặc các lựa chọn khác nếu muốn)
  - Mode: Snapshot
  - Enable: true
  - **Add**

Edit: Backup Job

General
Notifications
Retention
Note Template
Advanced

Node:
-- All --
Compression:
ZSTD (fast and good)
Storage:
VM\_PBS
Mode:
Snapshot
Schedule:
tue,thu,sat 04:00
Enable:
☒
Selection mode:
All
Job Comment:

| <input type="checkbox"/> | ID ↑ | Node | Status  | Name   | Type            |
|--------------------------|------|------|---------|--------|-----------------|
| <input type="checkbox"/> | 100  | pve  | stopped | Win11  | Virtual Machine |
| <input type="checkbox"/> | 101  | pve  | running | DSM    | Virtual Machine |
| <input type="checkbox"/> | 102  | pve  | running | NPM    | LXC Container   |
| <input type="checkbox"/> | 103  | pve  | running | MSEdge | LXC Container   |
| <input type="checkbox"/> | 104  | pve  | running | PBS    | Virtual Machine |

? Help
OK

## Lên lịch verify

- PBS web > Datastore > Verify Jobs > Add
  - Local Datastore: chọn datastore đã tạo khi cài đặt
  - Schedule: *tue,thu,sat 05:00* để lên lịch verify lúc 5h, sau khi backup
  - Skip Verified: chọn để bỏ qua những file đã verified. tiết kiệm tài nguyên
  - Re-Verify After: 7 ngày. Verify lại những file đã verify quá 7 ngày
  - Add**

Edit: Verification Job

Local Datastore: DSM\_NFS

Schedule: tue,thu,sat 05:00

Namespace: Root

Skip Verified: ☒

Max. Depth: Full

Re-Verify After: 7 days

Comment:

Job ID: v-b357bb25-e98d

Help

Advanced ☒

OK

## Lưu ý

Với task backup cho DSM. PVE sẽ thực hiện backup tất cả các ổ đĩa được passthrough cho DSM nên dung lượng cần backup là **rất lớn**, và điều này cũng là **không cần thiết** vì DSM đã sử dụng Hyper Backup.

Để tắt backup cho 2 ổ cứng này PVE Web > pve > VM DSM > Hardware > Disk. Tick chọn Advanced > Bỏ tích Backup

Edit: Hard Disk (sata2)

Disk

Bandwidth

Disk image: /dev/disk/by-id/ata-ST2000[

Cache: Default (No cache)

Discard: ☐

IO thread: ☐

SSD emulation: ☐

Backup: ☐

Read-only: ☐

Skip replication: ☐

Async IO: Default (io\_uring)

Help

Advanced ☒

OK

# Sao lưu với PBS (Advance)

Trong phần trước mình đã tìm hiểu về cách tạo backup với PBS. Phần này mình muốn nâng cấp thêm một chút.

Vì thời gian cần backup không nhiều. Việc để PBS chạy lên tục có thể gây lãng phí tài nguyên nên trong phần này mình hướng tới tự động hóa quy trình **bật PBS > Backup > Verify > tắt PBS**

Để thực hiện việc này thì sẽ sử dụng script và systemd timer.

Trước khi đến với script chính. Thì thêm một điều cần chú ý. Script này sẽ vẫn sử dụng script backup pve host đã viết ở bài [Sao lưu với PBS \(Simple\)](#)

Ok, Lets go!

## Quy trình backup

1. Bật PBS
2. Kiểm tra kết nối đảm bảo PBS sẵn sàng
3. Backup PVE host
4. Verify PVE host backup
5. Backup các VM được khai báo
6. Backup các CT được khai báo
7. Verify datastore chứa cả VM và CT backup
8. Prune
9. GC
10. Tắt PBS

Các bước trên được thực hiện tuần tự. Xong bước trên mới thực hiện bước dưới. Dừng ngay khi có lỗi.

## Script backup

PVE Web > Shell

- Tạo file thực thi backup `nano /usr/local/bin/backup.sh`
- Cấp quyền thực thi `chmod +x /usr/local/bin/backup.sh`

- Chạy thử `/usr/local/bin/backup.sh`
- Log kết thúc bằng dòng `===== BACKUP PIPELINE DONE =====` là thành công.

```
#!/usr/bin/env bash
set -euo pipefail

#####
C?U HÌNH CHUNG
#####
PBS_VMID=104

PBS_HOST='192.168.1.104' # ??a ch? IP c?a PBS
PBS_DATASTORE='DSM_NFS' # Datastore ?ã t?o trên PBS
PBS_USER_TOKEN='root@pam!PVE' # API Token ID
PBS_TOKEN_SECRET='0218e32d-fefe-4bce-9042-c2dd89389856' # API Token Secret
PBS_FINGERPRINT='35:72:be:a5:08:5f:7a:d1:b1:50:0e:69:fa:6c:bf:78:c9:94:d1:9e:45:34:e6:a0:21:88:3d:ab:59:a6:d8:2d' # Fingerprinter

Prune and GC job
PRUNE_JOB_ID=${PRUNE_JOB_ID:-default-prune} # Tên Prune job
PRUNE_SCHEDULE=${PRUNE_SCHEDULE:-'Sun 00:00'} # L?ch prune # Không quan tr?ng vì job này ch?y b?ng script
ch? k ch?y t? ??ng
KEEP_BACKUPS=${KEEP_BACKUPS:-7} # s? b?n backup t?i ?ã s? gi?

export PBS_REPOSITORY="${PBS_USER_TOKEN}@${PBS_HOST}:${PBS_DATASTORE}"
export PBS_PASSWORD="${PBS_TOKEN_SECRET}"
export PBS_FINGERPRINT="${PBS_FINGERPRINT}"

Script backup host pve
?ã vi?t ? ph?n tr??c
BACKUP_HOST_SCRIPT="/usr/local/bin/pve-self-backup.sh"

Danh sách VM/CT c?n backup
Không c?n d?u ph?y gi?a các s?
VM_LIST=(100 101 104) # VM IDs
CT_LIST=(102 103) # CT IDs

Tu? ch?n vmdump dùng cho VM/CT
VZDUMP_STORAGE="VM_PBS"
VZDUMP_OPTS=(--mode snapshot --compress zstd --ionice 7 --bwlimit 0 --storage "${VZDUMP_STORAGE}" --
notes-template '{{cluster}} - {{guestname}} - {{node}} - {{vmid}}')

Th?i gian ??i PBS online t?i ?a (giây)
WAIT_PBS_READY_SEC=$((10*60)) # 10 phút
Th?i gian ??i PBS shutdown t?i ?a (giây)
WAIT_PBS_SHUT_SEC=$((5*60)) # 5 phút

#####
HÀM TI?N ÍCH
#####
L?y ngày gi?
Th??ng dùng ?? in ra trong log
ts() { date '+%F %T'; }

Ki?m tra xem l?nh có chính xác không
require_cmd() {
 command -v "$1" >/dev/null 2>&1 || { echo "[${ts}] Missing command: $1"; exit 1; }
}

Ki?m tra VM có ch?y không
qm_is_running() {
 local vmid=$1
 qm status "$vmid" 2>/dev/null | grep -q 'status: running'
}
```

```

Kiểm tra PBS sẵn sàng chưa bằng cách
Thử TCP connect
Kiểm tra PBS
Kiểm tra dịch vụ proxmox-backup-proxy
wait_for_pbs_ready() {
 local deadline=$(($(date +%s) + WAIT_PBS_READY_SEC))
 local ok_streak=0
 echo "[${ts}] Waiting for PBS services (up to ${WAIT_PBS_READY_SEC}s)..."

 while ;; do
 # 1) Port 8007 mở?
 if ! timeout 2 bash -c "echo > /dev/tcp/${PBS_HOST}/8007" 2>/dev/null; then
 ok_streak=0; goto_sleep; continue
 fi
 # 2) Login repo thành công?
 if ! proxmox-backup-client login --repository "${PBS_REPOSITORY}" >/dev/null 2>&1; then
 ok_streak=0; goto_sleep; continue
 fi
 # 3) Dịch vụ proxy active bên trong PBS?
 if ! ssh -o BatchMode=yes -o ConnectTimeout=3 root@"${PBS_HOST}" \
 'systemctl is-active --quiet proxmox-backup-proxy'; then
 ok_streak=0; goto_sleep; continue
 fi

 ok_streak=$((ok_streak+1))
 if ((ok_streak >= 3)); then
 echo "[${ts}] PBS ready (proxy up, port 8007 OK, login OK)."
- sleep 5 # grace thêm vài giây
- return 0

```

```

verify_datastore() {
 echo "[$(ts)] Trigger verify on PBS datastore '${PBS_DATASTORE}'..."
 ssh -o BatchMode=yes root@"${PBS_HOST}" \
 "proxmox-backup-manager verify '${PBS_DATASTORE}' --ignore-verified true" \
 && echo "[$(ts)] Verify finished." \
 || { echo "[$(ts)] ERROR: Verify failed"; return 1; }
}

Sao l?u PVE host s? d?ng pve-self-backup.sh
backup_host() {
 echo "[$(ts)] === BACKUP HOST PVE ==="
 local tries=0
 local max=5
 local rc
 while ((tries<max)); do
 if DRY_RUN=0 "${BACKUP_HOST_SCRIPT}"; then
 echo "[$(ts)] Host backup DONE."; return 0
 fi
 rc=$?
 # N?u l?i k?t n?i thì retry, còn l?i khác thì fail ngay
 if journalctl -u run-backup-pipeline.service -n 50 --no-pager 2>/dev/null | grep -qE 'No route to
host|client error \(\Connect\)'; then
 tries=$((tries+1))
 echo "[$(ts)] Connect error, retry ${tries}/${max}..."
 sleep $((5*tries))
 else
 return $rc
 fi
 done
 echo "[$(ts)] ERROR: Host backup failed after retries."; return 1
}

Backup VMs
backup_vms() {
 echo "[$(ts)] === BACKUP VMs ==="
 for id in "${VM_LIST[@]}"; do
 echo "[$(ts)] vzdump VM ${id} ..."
 vzdump "${id}" "${VZDUMP_OPTS[@]}"
 echo "[$(ts)] VM ${id} backup DONE."
 done
}

Backup CTs
backup_cts() {
 echo "[$(ts)] === BACKUP CTs ==="
 for id in "${CT_LIST[@]}"; do
 echo "[$(ts)] vzdump CT ${id} ..."
 vzdump "${id}" "${VZDUMP_OPTS[@]}"
 echo "[$(ts)] CT ${id} backup DONE."
 done
}

Prune
prune_datastore() {
 local job="$PRUNE_JOB_ID"
 echo "[$(ts)] Ensure PRUNE job '${job}' exists (store='${PBS_DATASTORE}', keep-
last=${KEEP_BACKUPS})..."

 # N?u job ?ã t?n t?i -> update (gi? nguyên l?ch ho?c ??t l?i), disable ?? không t? ch?y
 if ssh -o BatchMode=yes root@"${PBS_HOST}" "proxmox-backup-manager prune-job show '${job}' >/dev/null
2>&1"; then
 ssh -o BatchMode=yes root@"${PBS_HOST}" \
 "proxmox-backup-manager prune-job update '${job}' \
 --store '${PBS_DATASTORE}' \
 --keep-last ${KEEP_BACKUPS} \

```

```

 --schedule '${PRUNE_SCHEDULE}' \
 --disable true"
else
 # T?o job m?i: c?n l?ch h?p l?, nh?ng disable ?? không t? ??ng ch?y
 ssh -o BatchMode=yes root@"${PBS_HOST}" \
 "proxmox-backup-manager prune-job create '${job}' \
 --store '${PBS_DATASTORE}' \
 --schedule '${PRUNE_SCHEDULE}' \
 --keep-last ${KEEP_BACKUPS} \
 --disable true"
fi || { echo "[$(ts)] ERROR: create/update prune job failed"; return 1; }

echo "[$(ts)] Running PRUNE job '${job}'..."
ssh -o BatchMode=yes root@"${PBS_HOST}" \
 "proxmox-backup-manager prune-job run '${job}'" \
 && echo "[$(ts)] Prune finished." \
 || { echo "[$(ts)] ERROR: Prune failed"; return 1; }
}

GC
gc_datastore() {
 echo "[$(ts)] Running GARBAGE COLLECTION on datastore '${PBS_DATASTORE}'..."
 ssh -o BatchMode=yes root@"${PBS_HOST}" \
 "proxmox-backup-manager garbage-collection start '${PBS_DATASTORE}'" \
 && echo "[$(ts)] GC finished." \
 || { echo "[$(ts)] ERROR: GC failed"; return 1; }
}

B?t PBS
pbs_start() {
 echo "[$(ts)] Starting PBS VM ${PBS_VMID}..."
 if qm_is_running "${PBS_VMID}"; then
 echo "[$(ts)] PBS already running."
 else
 qm start "${PBS_VMID}"
 fi
 wait_for_pbs_ready
}

T?t PBS
pbs_shutdown() {
 echo "[$(ts)] Shutting down PBS VM ${PBS_VMID}..."

 # 0) N?u ?ã t?t thì thoát s?m
 if ! qm_is_running "${PBS_VMID}"; then
 echo "[$(ts)] PBS already off."
 return 0
 fi

 # 1) ?u tiên t?t t? bên trong guest (an toàn nh?t)
 # Yêu c?u b?n ?ã c?u hình SSH key PVE -> PBS
 # Dùng systemd ?? poweroff không ch? (fork ra n?n)
 ssh -o BatchMode=yes -o ConnectTimeout=5 root@"${PBS_HOST}" \
 'nohup systemctl poweroff >/dev/null 2>&1 &' || true
 echo "[$(ts)] Sent 'systemctl poweroff' to PBS."

 # 2) Ch? PBS t?t h?n (polling qm status)
 SHUT_WAIT=${WAIT_PBS_SHUT_SEC:-300} # 5 phút m?c ??nh
 deadline=$(($(date +%s) + SHUT_WAIT))
 while qm_is_running "${PBS_VMID}"; do
 if (($(date +%s) > deadline)); then
 echo "[$(ts)] Guest did not poweroff within ${SHUT_WAIT}s. Trying 'qm shutdown'..."
 break
 fi
 sleep 5
 done
}

```

```

done

3) N?u v?n c?n ch?y -> th? 'qm shutdown' v?i timeout d?i h?n
if qm_is_running "${PBS_VMID}"; then
 qm shutdown "${PBS_VMID}" --timeout "${SHUT_WAIT}" || true
fi

4) Ch? th?m ch?t n?a
end2=$(($(date +%s) + 60))
while qm_is_running "${PBS_VMID}"; do
 if (($(date +%s) > end2)); then
 echo "[$(ts)] Still running after graceful attempts. Forcing 'qm stop'..."
 qm stop "${PBS_VMID}" || true
 break
 fi
 sleep 5
done

5) K?t lu?n: coi nh? th?nh c?ng n?u th?c s? ?? t?t
if ! qm_is_running "${PBS_VMID}"; then
 echo "[$(ts)] PBS is off."
 return 0
else
 echo "[$(ts)] WARNING: PBS looks stuck running. Please check 'qm monitor ${PBS_VMID}'."
 return 1
fi
}

#####
MAIN
#####
require_cmd qm
require_cmd proxmox-backup-client
require_cmd ssh
require_cmd vzdump

echo "[$(ts)] ===== BACKUP PIPELINE START ====="

1) Start PBS & wait
pbs_start

2) Backup host PVE
backup_host

3) Verify sau host backup
verify_datastore

4) Backup c?c VM/CT
backup_vms
backup_cts

5) Verify sau VM/CT backup
verify_datastore

6) PRUNE r?i GC
prune_datastore
gc_datastore

7) Shutdown PBS
pbs_shutdown

echo "[$(ts)] ===== BACKUP PIPELINE DONE ====="

```

# Tự động hóa

Thực hiện trên console hoặc ssh của PVE

- Tạo file `.service`. Đây là hành vi sẽ được thực hiện bởi `.timer`

- `nano /etc/systemd/system/backup.service`

- Sử dụng nội dung sau

```
[Unit]
Description=Backup PVE host configuration to PBS
After=network-online.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/backup.sh
```

- Tạo file `.timer`. Nó sẽ thực thi `.service` theo thời gian được chỉ định.

- `nano /etc/systemd/system/backup.timer`

- Sử dụng nội dung sau

```
[Unit]
Description=Schedule PVE host config backup (Tue, Thu, Sat at 2:00)

[Timer]
OnCalendar=Tue,Thu,Sat 2:00
Persistent=true
AccuracySec=1min

[Install]
WantedBy=timers.target
```

- Kích hoạt timer

- Reload systemd `systemctl daemon-reload`

- Kích hoạt timer `systemctl enable --now backup.timer`

- Kiểm tra

- `systemctl list-timers | grep backup`

- `systemctl status pve-host-backup.timer`

- Kết quả tương tự như hình sau là thành công

```
root@pve:~# systemctl list-timers | grep backup
Tue 2025-11-11 05:00:00 +07 3h 54min Sun 2025-11-09 03:43:53 +07 - backup.timer backup.service
Wed 2025-11-12 00:00:00 +07 22h Tue 2025-11-11 00:00:43 +07 1h 4min ago dpkg-db-backup.timer dpkg-db-backup.service
root@pve:~# systemctl status backup.timer
● backup.timer - Schedule PVE host config backup (Tue, Thu, Sat at 5:00)
 Loaded: loaded (/etc/systemd/system/backup.timer; enabled; preset: enabled)
 Active: active (waiting) since Tue 2025-11-11 00:55:26 +07; 10min ago
 Invocation: 3a68e6b2d359453b954eb86930c11eac
 Trigger: Tue 2025-11-11 05:00:00 +07; 3h 54min left
 Triggers: ● backup.service
Nov 11 00:55:26 pve systemd[1]: Started backup.timer - Schedule PVE host config backup (Tue, Thu, Sat at 5:00).
```