Yundera Cybersecurity and Data Privacy Technical Report

Executive Summary

This report outlines the minimum viable security and data protection stance suitable for initial public deployment of Yundera personal cloud service. First, we provide clear mental models to achieve “strategic minimization” by reducing attack surfaces, operational overhead, and recovery complexity while maximizing protection value. Second, we describe the newly implemented security measures to address the most critical gaps in Yundera’s security stance. Finally, we make further technical and organizational recommendations for long-term platform growth. This report is intended to be a ‘living document’, ideally hosted in a version control system and updated as the system itself changes.

Current Strengths

- Infrastructure Foundation: Proxmox VE provides solid virtualization security
- Key-based SSH access for all servers
- Strong vertical tech ownership due to open source based stack.

Critical Gaps Addressed

A summary of the newly implemented measures configured and tested as part of this report.

Area	Prior Stance	Newly Implemented
Network isolation	None; servers accessible via public IP.	Tailscale overlay VPN implemented. See ‘Tailscale Setup’ document.
Access control	None; single root user has access to all VM user data	Planned but not implemented - Definition of fine-grained role based privileges configured for different user types. See ‘Role Based Privileges’ section within this document.
Backup	None	Proxmox Backup Server in place with automatic VM backups and protection against Ransomware

Next Steps Recommendations

Immediately Actionable

- New `proxmox-middlware` features
  - Randomize user password upon VM instantiation (and return it on /create endpoint)
    - Today, VMs are cloned from the same image without changes to the ID or password.
  - Monitor VMs and hosts for abnormal resource usage, Push alerts to admins
  - Create script to automate moving all VMs out of a given host, so that the host can be upgraded safely. Make use of the seamless cross-host VM migration capability demonstrated in the `limitless-pc` project.
- Improve SSH key access organization (see relevant section)

- Automate Proxmox Backup Server recovery testing

Further Planning Necessary / Out of Scope

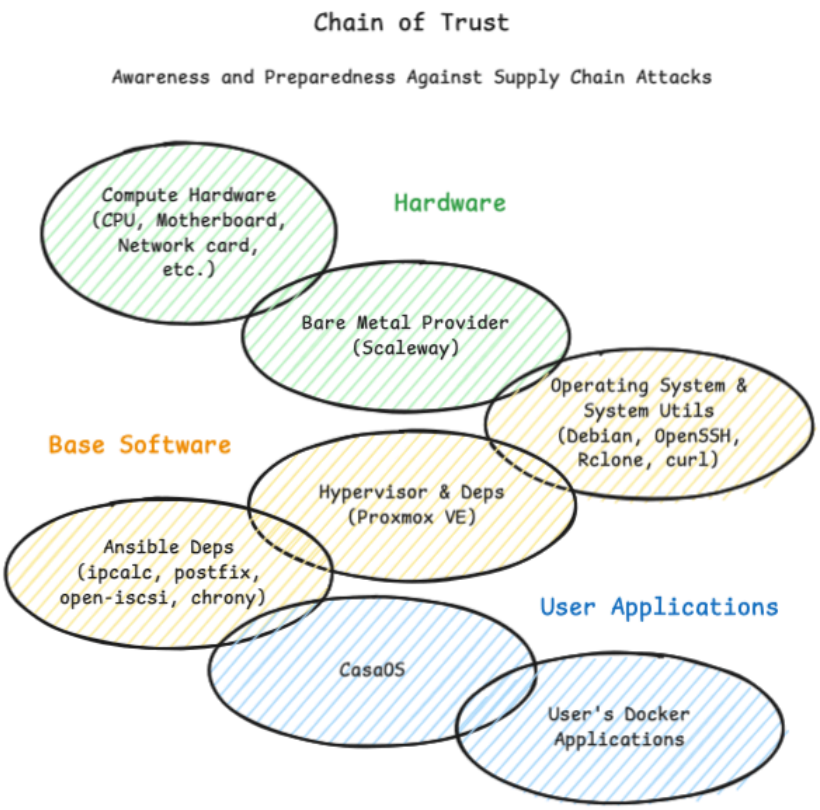
The present report assumes no dedicated cybersecurity team, and therefore advanced topics such as penetration testing and security audits are out of scope.

- Network-wide IDS (Intrusion Detection System) installation.
- Penetration testing.
- Audit logs
- Legal and compliance
- Education and company culture against social engineering attacks

Prevention

Minimizing the attack surface

Managing Supply Chain Risk: Criticality-Weighted Trust



The mental model that we bring to effective management of supply chain risk is called “Criticality-Weighted Trust”. This principle dictates that the level of scrutiny and trustworthiness for software must be directly proportional to their proximity to the core. Components operating at foundational levels, due to their broad privileges and high potential for widespread impact upon compromise, demand the highest degree of assurance

and vetting. In our example, we are careful to limit our dependencies at the base software levels to industry-standard, widely used, heavily tested software that has a proven track record of active security patches that are backed by strong governance and predictable financial motives. On the other hand, user-level software such as those installed by users from the CasaOS library of docker containers can be assessed with a relatively higher risk tolerance. The next section demonstrates an example rubric for evaluating dependencies.

Dependency Trustworthiness Rubrics

Minimizing risk of each dependency

- **Adoption Scale**: Widespread use can be a signal of maturity and trustworthiness.
- **Community Health**: Active development and support community
- **Security Response**: Track record of timely patches in response to security bugs (CVEs). All software has bugs, therefore lack of security patches is not a good signal high security software at all. At times, it is the community that finds security bugs (see XZ utils backdoor near-miss incident) so this metric is tied to the one above.
- **Organizational Trust**: The highest scores are given to organizations who have a stable governance and clearly self-sufficient financial structure. On the other hand, 'hobby' projects that rely solely on voluntary work or projects that funded through irregular sources of funds get lower scores.

The recommendation is to update this table as modifications are made to the infrastructure, continually re-assessing the necessity and suitability of each component that constitute the Yundera system.

	Criticality Weight	Adoption scale	Community Health	Security Response	Organizational Trust	Final Suitability
Debian	HIGH	⭐⭐⭐	⭐⭐⭐	⭐⭐⭐	⭐⭐⭐	✅
OpenSSH	HIGH	⭐⭐⭐	⭐⭐⭐	⭐⭐⭐	⭐⭐⭐	✅
Rclone	HIGH	⭐⭐	⭐⭐	⭐⭐⭐	⭐⭐⭐	✅
curl	HIGH	⭐⭐⭐	⭐⭐	⭐⭐⭐	⭐⭐⭐	✅
Proxmox VE	HIGH	⭐⭐	⭐⭐	⭐⭐	⭐⭐	✅
ipcalc	MEDIUM	⭐⭐	⭐	⭐	⭐⭐	✅
postfix	MEDIUM	⭐⭐	⭐	⭐⭐	⭐⭐	✅
open-iscsi	MEDIUM	⭐⭐	⭐	⭐⭐	⭐⭐	✅
chrony	MEDIUM	⭐⭐	⭐	⭐⭐	⭐	✅
Jellyfin	LOW	⭐⭐	⭐⭐⭐	⭐⭐	⭐	✅

Upgrade-First Software

Minimizing security lag

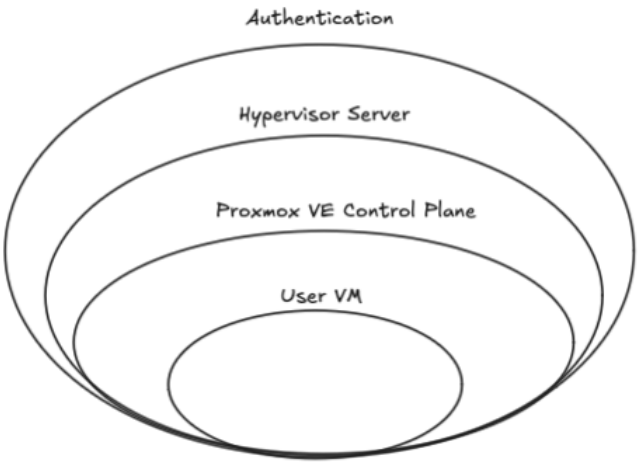
In the previous section, we examined the trustworthiness of each dependency with respect to their vendors and community support. Next, we optimize for minimizing the time lag between the discovery of a vulnerability in each dependency and the corresponding update to our production system.

'Install now, figure out how to upgrade later' is a recipe for unmaintainable outdated systems. In order to actively prevent software obsolescence within our systems, we must be selective in introducing components into our system such that each new component has a clear documented path to future upgrades.

Component	Upgrade Path	Schedule Considerations	Responsible person/team
Debian & Proxmox VE or PBS	`apt-get upgrade` and `apt-get dist-upgrade` handles both underlying Debian and Proxmox VE and Proxmox Backup Server.	`unattended-upgrades` is not recommended for production. Move VMs to other nodes in the cluster before running upgrades.	(To be determined organizationally)
In-house software (Proxmox middlware, PCS Orchestrator, frontend)	Use Github dependabot to get notifications on dependency updates. Run tests with new dependency versions. Then, release it.	Can be done regularly. Consider setting up automated PRs for dependabot.	(To be determined organizationally)
Template VMs	'Clone' a VM template, run upgrades, clean the VM, then save as template again.	Can be done regularly but is best done manually. Consider setting up a weekly or monthly personal reminder.	(To be determined organizationally)

Protection Against Malicious External Actors: Defense in Depth

Defense in depth relies on multiple layers of authentication and authorization controls, along with impermeable barriers between each layer. In this section, we will enumerate the logical layers that exist within our system and discuss how the boundaries between each layer could be strengthened.



Authentication: SSH key management strategy

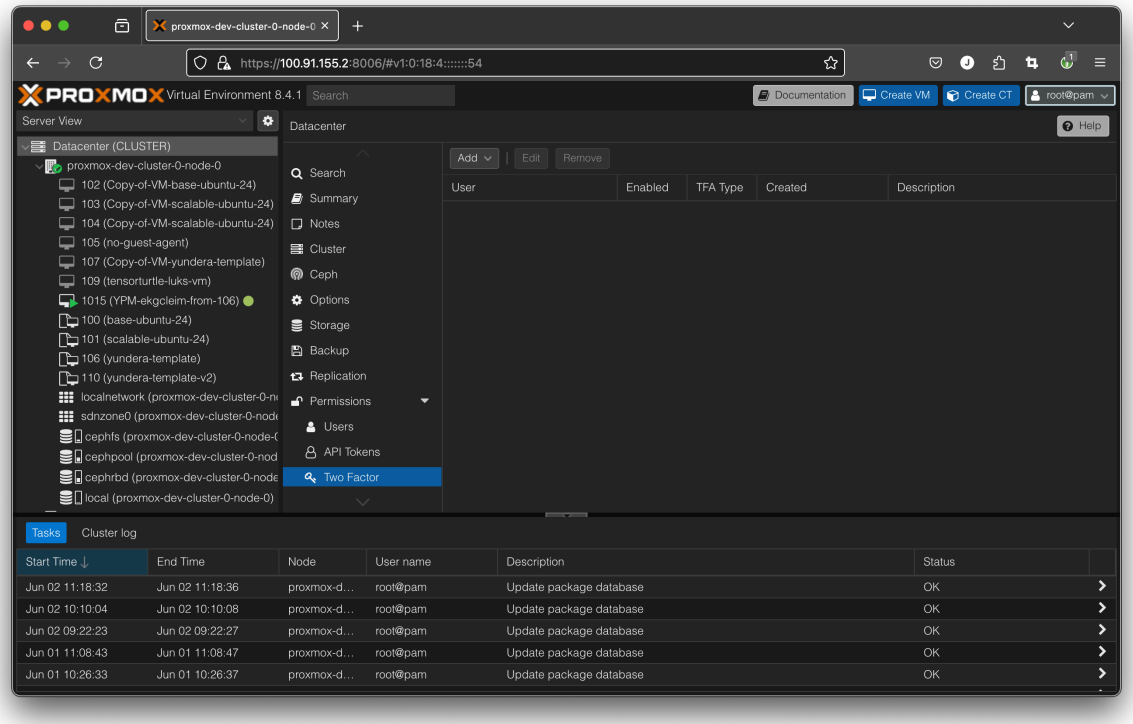
SSH keys can be shared within the following groups, but must not be shared across them:

- Proxmox VE clustered hosts
  - There is no point to using independent keys for each, because all hosts within a cluster have root access to each other through the Proxmox cluster.
- Proxmox Backup Server - has decent protection against Ransomware attacks.
- Guest VMs: As part of the initial VM provisioning process, an SSH public key is copied to the customer's VM, and then various configuration files and scripts are copied over through SSH. This SSH key pair can be newly

generated at any time, but the `proxmox-middleware` service then needs to be restarted with the new public key.

- It would also be possible to modify `proxmox-middleware` to generate new SSH key pairs for each new instance (keeping the private key only in the `proxmox-middleware`'s process memory), but doing so may introduce unnecessary complexity given our threat model.

Two factor authentication (TFA) should be enabled for all administrators. The configuration page can be found in the top level cluster's menu:



Beside these measures, SSH keys should be rotated regularly and stored securely (Password Manager).

Authorization: Role Based Access Controls in Proxmox VE

Proxmox provides quite fine-grained permissions (called 'privileges') that can be combined into 'roles', which is assigned to each user account.

https://pve.proxmox.com/wiki/User_Management#pveum_permission_management

Virtual machine related privileges

- `SDN.Use`: access SDN vnets and local network bridges
- `VM.Allocate`: create/remove VM on a server
- `VM.Audit`: view VM config
- `VM.Backup`: backup/restore VMs
- `VM.Clone`: clone/copy a VM
- `VM.Config.CDROM`: eject/change CD-ROM
- `VM.Config.CPU`: modify CPU settings
- `VM.Config.Cloudinit`: modify Cloud-init parameters
- `VM.Config.Disk`: add/modify/remove disks
- `VM.Config.HWType`: modify emulated hardware types
- `VM.Config.Memory`: modify memory settings
- `VM.Config.Network`: add/modify/remove network devices
- `VM.Config.Options`: modify any other VM configuration
- `VM.Console`: console access to VM
- `VM.Migrate`: migrate VM to alternate server on cluster
- `VM.Monitor`: access to VM monitor (kvm)
- `VM.PowerMgmt`: power management (start, stop, reset, shutdown, …)
- `VM.Snapshot.Rollback`: rollback VM to one of its snapshots
- `VM.Snapshot`: create/delete VM snapshots

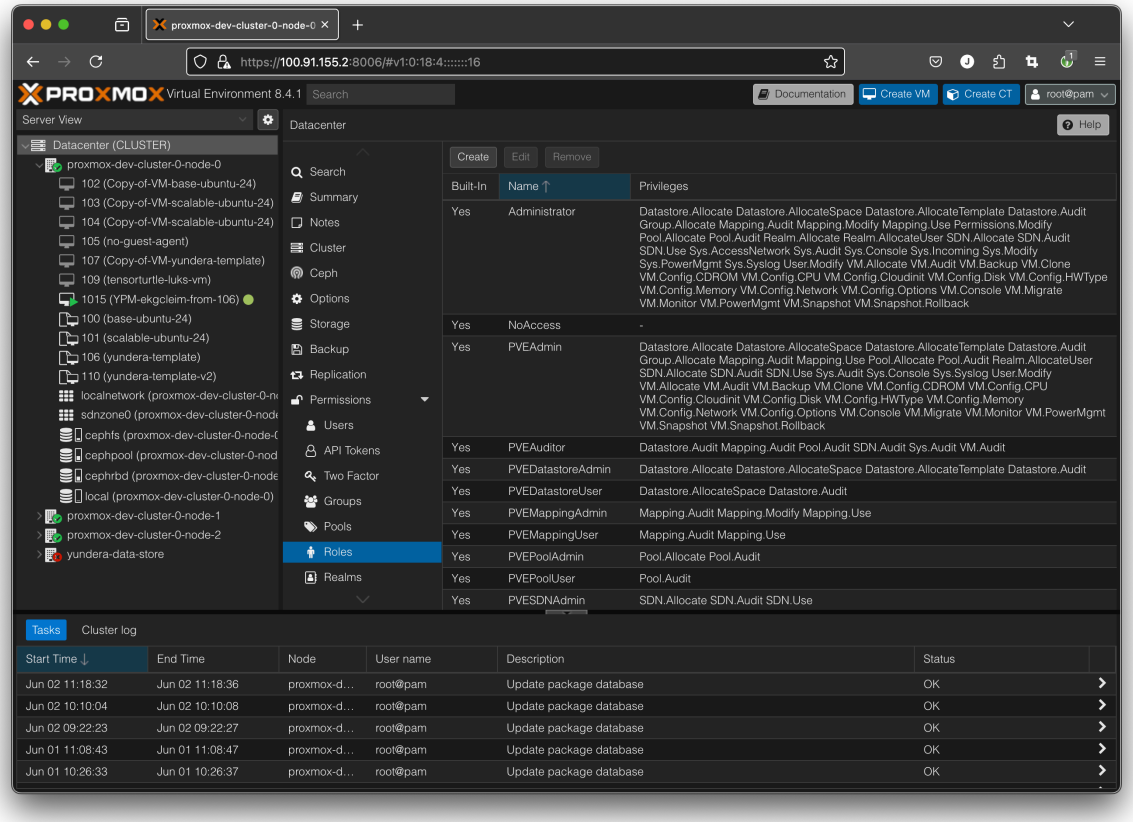
Proxmox provides a selection of pre-configured roles:

Roles

A role is simply a list of privileges. Proxmox VE comes with a number of predefined roles, which satisfy most requirements.

- `Administrator`: has full privileges
- `NoAccess`: has no privileges (used to forbid access)
- `PVEAdmin`: can do most tasks, but has no rights to modify system settings (`Sys.PowerMgmt`, `Sys.Modify`, `Realm.Allocate`) or permissions (`Permissions.Modify`)
- `PVEAuditor`: has read only access
- `PVEDatastoreAdmin`: create and allocate backup space and templates
- `PVEDatastoreUser`: allocate backup space and view storage
- `PVEMappingAdmin`: manage resource mappings
- `PVEMappingUser`: view and use resource mappings
- `PVEPoolAdmin`: allocate pools
- `PVEPoolUser`: view pools
- `PVESDNAdmin`: manage SDN configuration
- `PVESDNUser`: access to bridges/vnets
- `PVESysAdmin`: audit, system console and system logs
- `PVETemplateUser`: view and clone templates
- `PVEUserAdmin`: manage users
- `PVEVMAdmin`: fully administer VMs
- `PVEVMUser`: view, backup, configure CD-ROM, VM console, VM power management

However, we recommend configuring an more customized set of roles due to the specific operational requirements of Yundera. Note that these roles are not configured in the development cluster currently and require further testing.



Custom Role Name	Description	Granted Privileges
`root`	All-powerful. Only use in break-glass scenarios. Consider four-eyes requirements for this usage, enforced by multiple factor authentication.	Same as `Administrator` Role
`RootNoVMConsole`	All-powerful, but does not have Console access to VMs, meaning that customer data is not visible to this user through the web UI. HOWEVER, `Sys.Console` means that they can still exfiltrate the VM image through the command line. This is only a basic security measure.	Same as `Administrator` but excluding `VM.Console`
`ClusterInstaller`	Install and manage Proxmox and Ceph storage clustering, SDN configuration. No access to customer VMs.	`Group.Allocate` `Pool.Allocate` `SDN.Allocate` `Sys.Console` `Sys.Modify` `Sys.PowerMgmt` `Datastore.Allocate` `Datastore.AllocateSpace` `Datastore.AllocateTemplate` `Datastore.Audit`
`TemplateMaker`	Needs access to SSH and VNC into VMs that they create, and to convert them into templates and vice versa.	All ‘Virtual machine related privileges’.

Network Separation

The following types of traffic flow through the network:

- Corosync: Low throughput but strict latency requirements. The realtime backend for Proxmox VE clustering.
- Ceph cluster: Very high throughput. Our cluster is hyper-converged so any and all storage goes through this network. Typically this network becomes the bottleneck in the ceph storage system.
- Internet gateway: VMs access the open internet through the `vnet` configured in the cluster’s SDN. High throughput but less than ceph cluster - limited by outgoing ethernet connection to the open internet.

In the development cluster, the entry-level servers only have one network interface so network segregation is not implemented. Further work is required using the production cluster to clarify this aspect.

VPN for Control Plane

Please see the associated document called “Tailscale Setup”. It details step-by-step instructions for configuring a Proxmox Cluster with Tailscale. It is possible to migrate a naive Proxmox Cluster to use Tailscale, albeit with some manual steps.

Tailscale is a zero-config, mesh, high-performance VPN overlay that makes it easy to lock down the Proxmox web UI and API paths to be accessible only from other systems authorized onto the VPN.

Inter-VM traffic.

Conveniently, since we created a ‘Simple Zone’ when configuring SDN, inter-VM traffic is disabled. It is recommended to keep this configuration going forward.

https://pve.proxmox.com/pve-docs/chapter-pvesdn.html#pvesdn_zone_plugin_simple

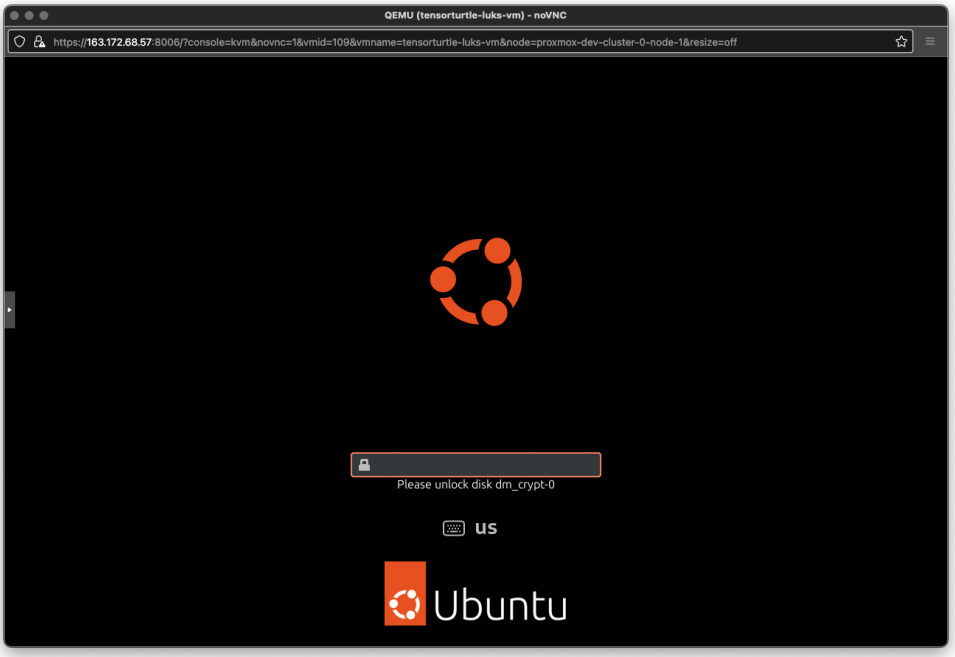
You can confirm this by running

```
nmap -sn 10.0.0.0/24
```

and observe that only the gateway (10.0.0.1) and the machine itself are the two devices that show up on the network.

VM-level Encryption: Explored but not recommended.

It is possible to encrypt the VM images using Ubuntu’s built-in LUKS with no modification to any hypervisor configuration - we simply select the option during initial Ubuntu installation. Enabling LUKS can potentially improve data privacy because it means that when the VM is shut down, the user data at rest is not accessible even to the Proxmox root user. However, this somewhat extreme measure is not recommended because reboots require manual unlocking through the VNC, or some usage of `cryptsetup` which is deemed too fragile and complex for our use case. Role-based access permissions (previous section), along with encrypted backups, provide a more reasonable approach to protecting user data.



Detection

Direct detection of intrusion is not trivial.

- User IP or fingerprint based anomaly detection
- Resource monitoring on VM and hypervisor hosts.
    - Either `proxmox-middleware` service can be modified to include an endpoint for fetching and summarizing resource usage across VMs and nodes, or a separate service can be created to do this task.

Uptime alerts: Get notified of system outages, regardless of cause (hardware/software failure, DDoS or ransomware attack). Recommended tools:

- Uptime robot (paid, freemium), https://github.com/louislam/uptime-kuma (self hosted, free)

Recovery

Infrastructure Restoration

Minimizing knowledge fragmentation

Assuming that the whole infrastructure needs to be wiped and reconfigured, do we have the documentation and know-how?

For scalable organizational growth, building up the entire system from the ground up can become a challenge. It is therefore crucial to enforce at the organizational level strong documentation, process-based vs. human-based workflows, and infrastructure as code wherever possible.

Data Resilience: Proxmox Backup Server

Minimizing possibility of data loss

At the core, Yundera provides compute and storage. Generally, users are much more sensitive to permanent data loss than temporary compute outage. Especially given Yundera's product positioning that caters to heavy data users, data is arguably the most valuable asset to protect against all forms of threats discussed above. This section shows how Proxmox Backup Server is integrated into the development cluster.

Recommended reading:

https://pve.proxmox.com/wiki/Backup_and_Restore

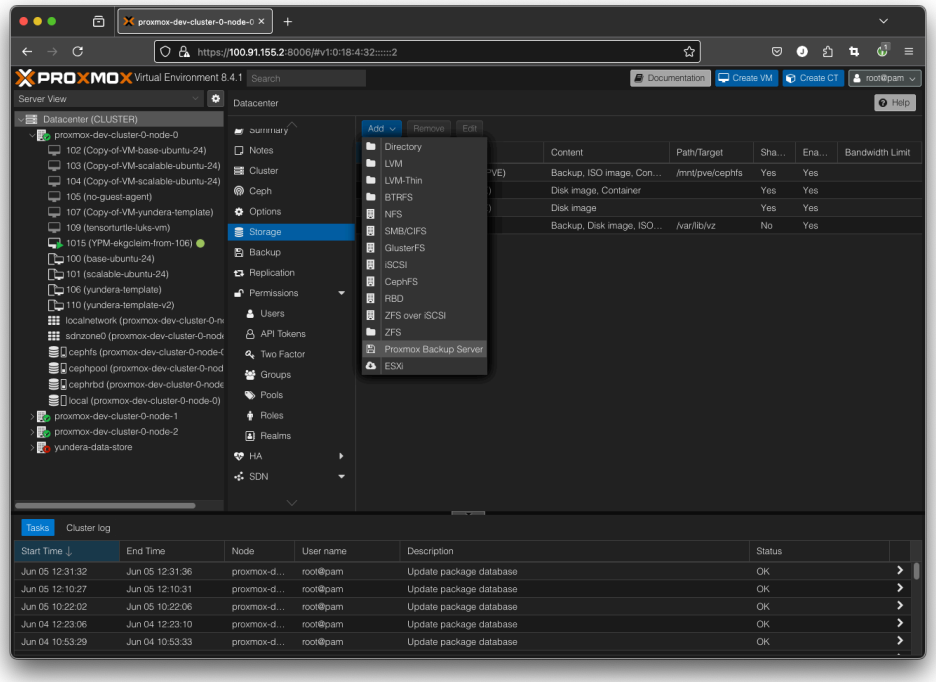
Set up Proxmox Backup Server (PBS)

Please see the associated document "PBS Setup" and "Tailscale: Securing the Control Plane" documents first.

The remainder of this document assumes that a basic PBS server has been set up and is reachable through the same tailnet as the Proxmox VE servers.

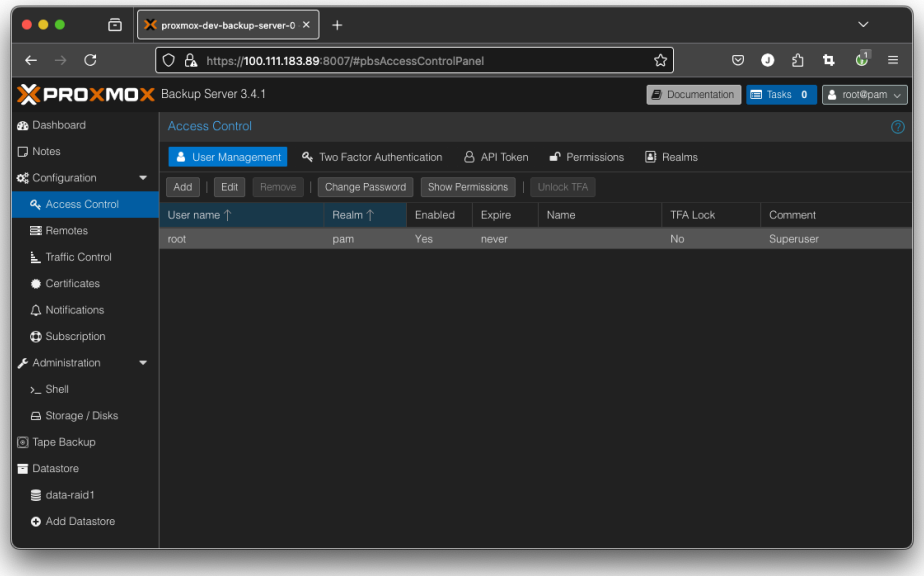
Setting up Proxmox VE backups to Proxmox Backup Server

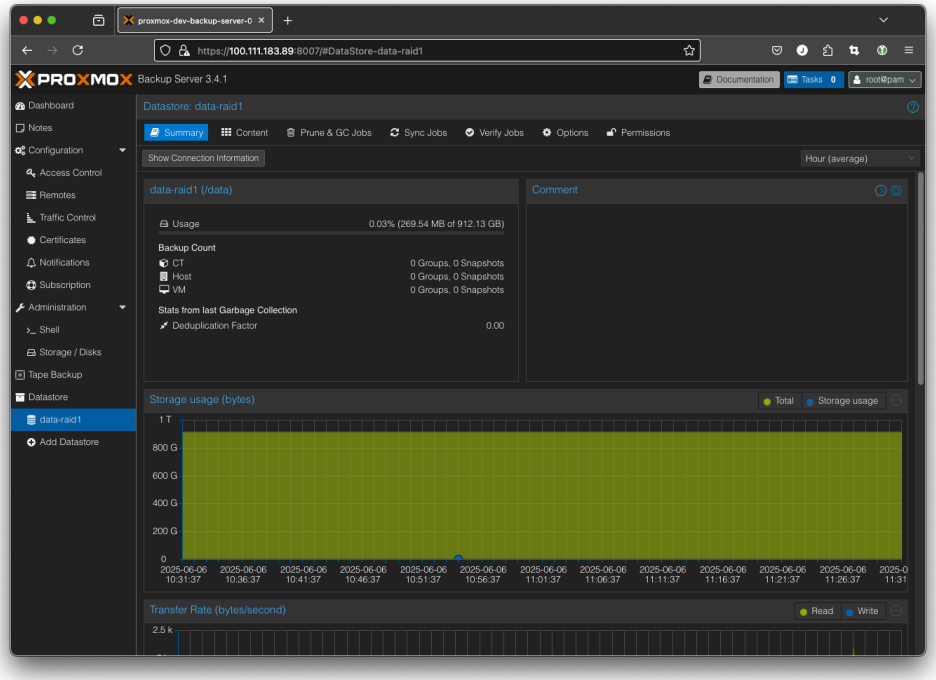
First, we log into the Proxmox VE dashboard. Go to the top-level 'Datacenter' settings.



https://pve.proxmox.com/pve-docs/chapter-pvesm.html#storage_pbs

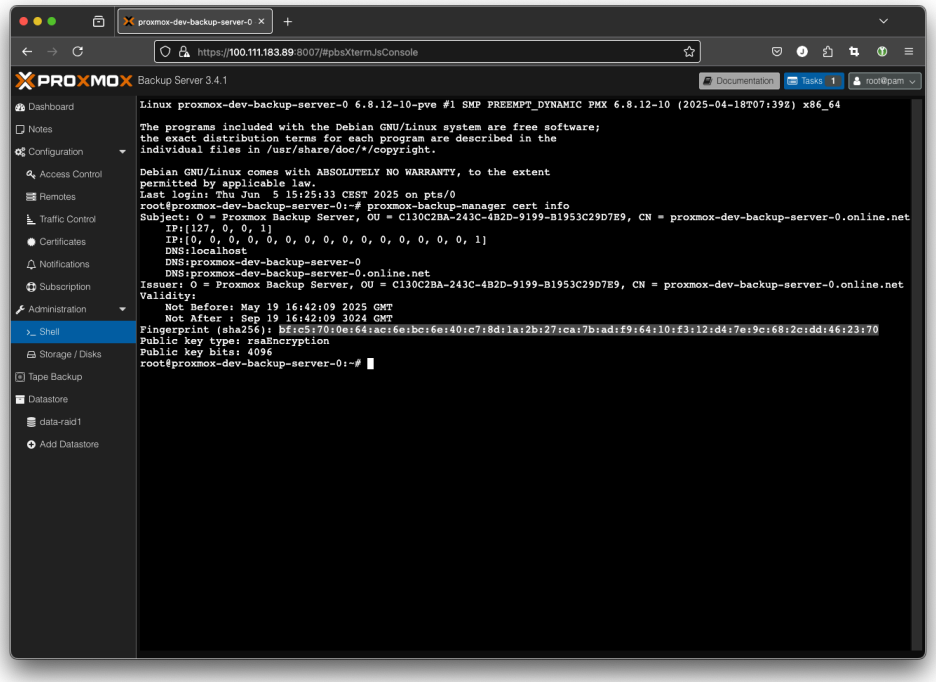
In PBS, the default user is 'root@pam'. However, this user is overly powerful. In the 'PBS Setup' document, we created a 'backup-admin@pbs' user. Enter this along with the password in the Proxmox VE storage configuration page.

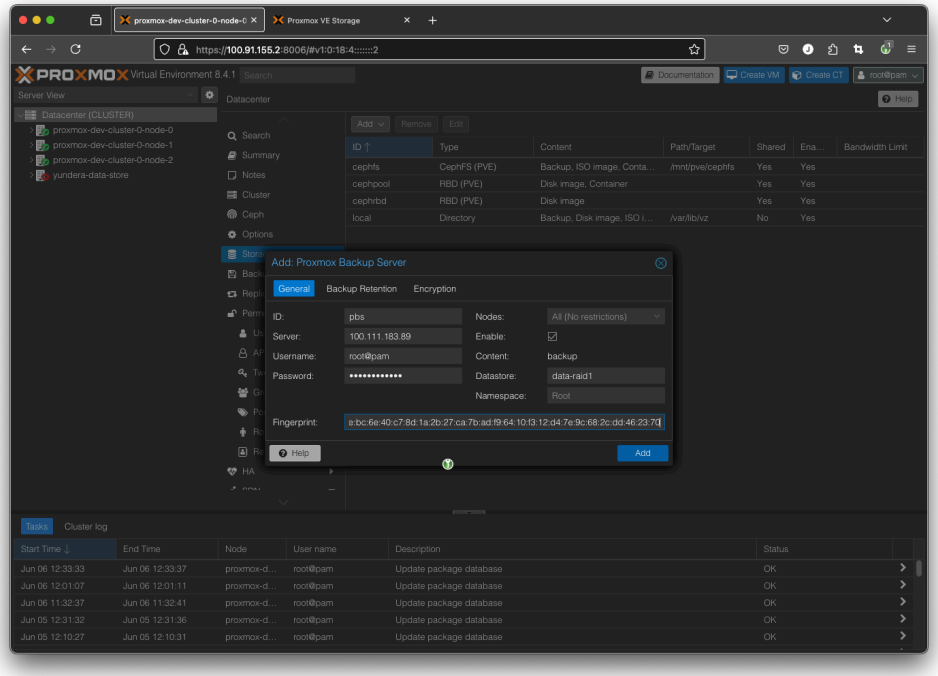


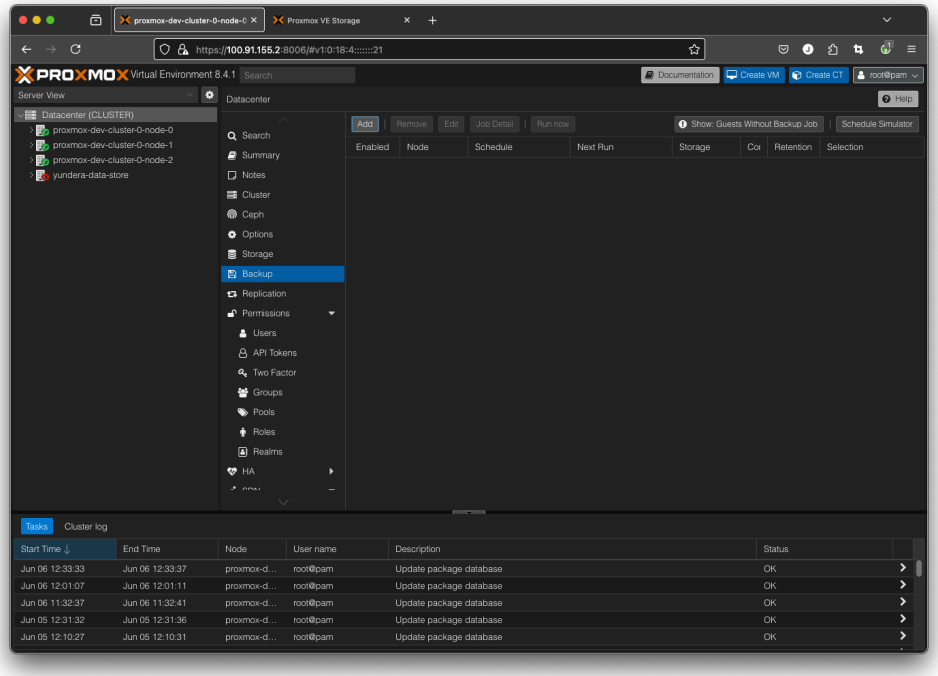


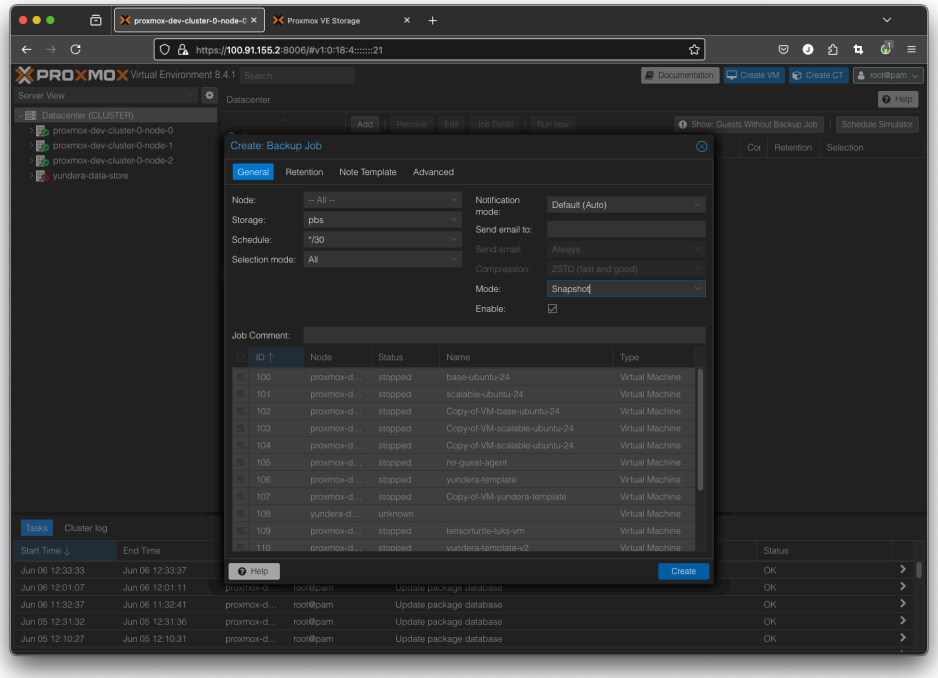
The ‘datastore’ refers to the name of the Datastore that we created in PBS earlier. In this example, it’s ‘data-raid1’.

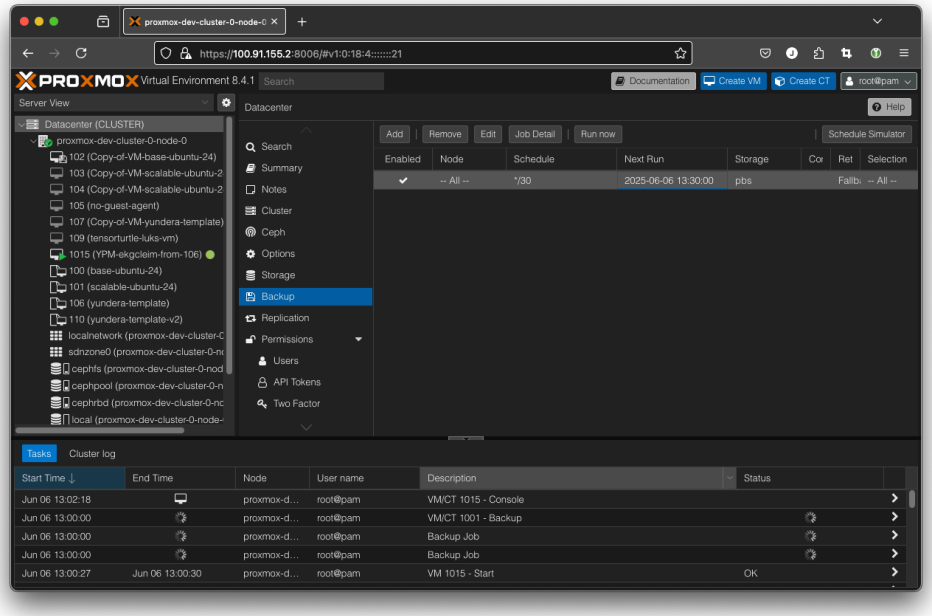
`proxmox-backup-manager cert info`

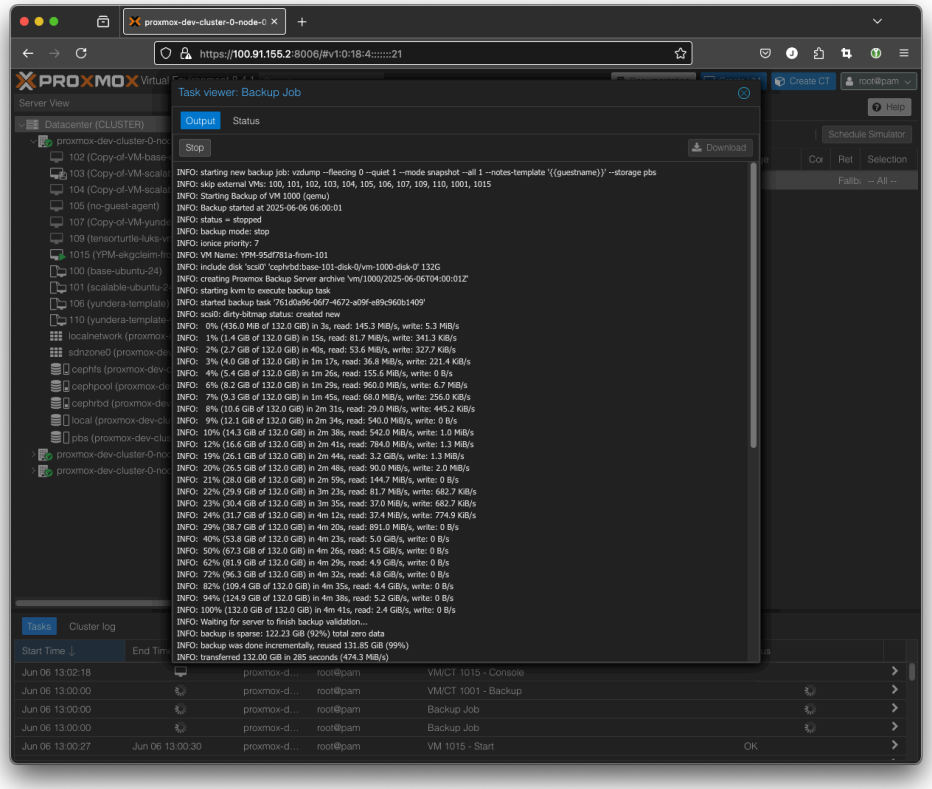

proxmox-dev-backup-server-0
https://100.111.183.89:8007/#pbsXtermJsConsole
PROXMOX Backup Server 3.4.1
Documentation
Tasks 1
root@pam
Dashboard
Notes
Configuration
Access Control
Remotes
Traffic Control
Certificates
Notifications
Subscription
Administration
Shell
Storage / Disks
Tape Backup
Datastore
data-raid1
Add Datastore
Linux proxmox-dev-backup-server-0 6.8.12-10-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-10 (2025-04-18T07:39Z) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jun 5 15:25:33 CEST 2025 on pts/0
root@proxmox-dev-backup-server-0:~# proxmox-backup-manager cert info
Subject: O = Proxmox Backup Server, OU = C130C2BA-243C-4B2D-9199-B1953C29D7E9, CN = proxmox-dev-backup-server-0.online.net
IP:[127, 0, 0, 1]
IP:[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
DNS:localhost
DNS:proxmox-dev-backup-server-0
DNS:proxmox-dev-backup-server-0.online.net
Issuer: O = Proxmox Backup Server, OU = C130C2BA-243C-4B2D-9199-B1953C29D7E9, CN = proxmox-dev-backup-server-0.online.net
Validity:
Not Before: May 19 16:42:09 2025 GMT
Not After : Sep 19 16:42:09 2024 GMT
Fingerprint (sha256): bf:c5:70:0e:64:ac:6e:bc:6e:40:c7:8d:1a:2b:27:ca:7b:ad:f9:64:10:f3:12:d4:7e:9c:68:2c:dd:46:23:70
Public key type: rsaEncryption
Public key bits: 4096
root@proxmox-dev-backup-server-0:~#


proxmox-dev-cluster-0-node-0
Proxmox VE Storage
https://100.91.155.2:8006/#v1:0:18:4:::::2
PROXMOX Virtual Environment 8.4.1
Documentation
Create VM
Create CT
root@pam
Server View
Datacenter
Datacenter (CLUSTER)
proxmox-dev-cluster-0-node-0
proxmox-dev-cluster-0-node-1
proxmox-dev-cluster-0-node-2
yundera-data-store
Search
Summary
Notes
Cluster
Ceph
Options
Add
Remove
Edit
ID
Type
Content
Path/Target
Shared
Bandwidth Limit
cephfs
CephFS (PVE)
Backup, ISO image, Conta...
/mnt/pve/cephfs
Yes
cephpool
RBD (PVE)
Disk image, Container
cephrbd
RBD (PVE)
Disk image
local
Directory
Backup, Disk image, ISO i...
/var/lib/vz
No
Add: Proxmox Backup Server
General
Backup Retention
Encryption
ID:
pbs
Server:
100.111.183.89
Username:
root@pam
Password:
Nodes:
All (No restrictions)
Enable:
Content:
backup
Datastore:
data-raid1
Namespace:
Root
Fingerprint:
e:bc:6e:40:c7:8d:1a:2b:27:ca:7b:ad:f9:64:10:f3:12:d4:7e:9c:68:2c:dd:46:23:70
Help
Add
Tasks
Cluster log
Start Time
End Time
Node
User name
Description
Status
Update package database
OK


proxmox-dev-cluster-0-node-0
Proxmox VE Storage
https://100.91.155.2:8006/#v1:0:18:4::::::21
PROXMOX Virtual Environment 8.4.1
Search
Documentation
Create VM
Create CT
root@pam
Server View
Datacenter
Help
Datacenter (CLUSTER)
proxmox-dev-cluster-0-node-0
proxmox-dev-cluster-0-node-1
proxmox-dev-cluster-0-node-2
yundera-data-store
Search
Summary
Notes
Cluster
Ceph
Options
Storage
Backup
Replication
Permissions
Users
API Tokens
Two Factor
Groups
Pools
Roles
Realms
HA
Add
Remove
Edit
Job Detail
Run now
Show: Guests Without Backup Job
Schedule Simulator
Enabled
Node
Schedule
Next Run
Storage
Retention
Selection
Tasks
Cluster log
Start Time
End Time
Node
User name
Description
Status
Jun 06 12:33:33
Jun 06 12:33:37
Update package database
OK
Jun 06 12:01:07
Jun 06 12:01:11
Jun 06 11:32:37
Jun 06 11:32:41
Jun 05 12:31:32
Jun 05 12:31:36
Jun 05 12:10:27
Jun 05 12:10:31


Create: Backup Job
General
Retention
Note Template
Advanced
Node:
-- All --
Storage:
pbs
Schedule:
*/30
Selection mode:
All
Notification mode:
Default (Auto)
Send email to:
Send email:
Always
Compression:
ZSTD (fast and good)
Mode:
Snapshot
Enable:
Job Comment:
ID
Name
Type
100 base-ubuntu-24 Virtual Machine
101 scalable-ubuntu-24 Virtual Machine
102 Copy-of-VM-base-ubuntu-24 Virtual Machine
103 Copy-of-VM-scalable-ubuntu-24 Virtual Machine
104 Copy-of-VM-scalable-ubuntu-24 Virtual Machine
105 no-guest-agent Virtual Machine
106 yundera-template Virtual Machine
107 Copy-of-VM-yundera-template Virtual Machine
108 yundera-d... unknown Virtual Machine
109 tensorturtle-luks-vm Virtual Machine
110 yundera-template-v2 Virtual Machine
Help
Create





Note: While it may initially be concerning that the backup process is trying to back up all 132GB (in our example, 32GB base image + 100GB data additional), Proxmox is intelligent enough to quickly skip over any large chunks of empty (zero) data within the image. Therefore, the first tens of gigabytes (which contain actual OS or user data information) are slow to transfer over, but it speeds up massively once it reaches the end of actual files.

Furthermore, Proxmox Backup Server implements quite good deduplication. The final compression ratio can be seen in PBS (after garbage collection, which happens daily by default):

Scheduled random sampling of recovery from backups is crucial. It is said that: “If you don’t test your backups, it’s just wishful thinking.” More work is needed to implement this in an automated manner.