

VISORMON

# User Guide

Monitor and manage your Proxmox VE servers from your iPhone or Android phone.

Updated November 2026

For VisorMon on iPhone and Android

Jimmy Fat Masterson

jimmyfatmasterson.com

# Contents


1 **Welcome to VisorMon**

1.1 What VisorMon does
1.2 What you need
1.3 The app at a glance
1.4 How this guide is organized
1.5 Conventions in this guide

2 **Getting started**

2.1 Before you start
2.2 First launch
2.3 Creating an API token
2.4 Signing in
2.5 Trying the demo
2.6 More than one server
2.7 Signing out

3 **The dashboard**

3.1 What the dashboard shows
3.2 Accounts that see part of the cluster
3.3 Keeping the figures up to date
3.4 The top bar and the tabs
3.5 When a node is down
3.6 Hiding and reordering panes

4 **Nodes**

4.1 The Nodes tab
4.2 A node's page
4.3 Actions and sections
4.4 Rebooting or shutting down a node
4.5 Starting, stopping, suspending or migrating every guest
4.6 The node shell
4.7 Services
4.8 System log
4.9 Updates
4.10 Network
4.11 Disks
4.12 Firewall
4.13 Tasks
4.14 System

## 5 Guests

5.1 Three places to work with guests
5.2 Reading a guest card
5.3 The guest overview on the dashboard
5.4 The guests dashboard
5.5 The Guests tab
5.6 Questions about guests

## 6 Working with a guest

6.1 A guest's page
6.2 Power actions
6.3 The console
6.4 Snapshots
6.5 Backup and restore
6.6 Clone
6.7 Migrate
6.8 Hardware
6.9 Options
6.10 Tags
6.11 Pending changes
6.12 Cloud-Init
6.13 Firewall
6.14 Convert to template
6.15 Notes and tasks
6.16 Creating a VM or container
6.17 Permissions, read-only mode and VisorMon Pro

## 7 Storage

7.1 The Storage tab
7.2 A storage's contents
7.3 What you can do from the Storage tab

## 8 Search, tasks and alerts

8.1 Searching the cluster
8.2 Cluster tasks
8.3 Alerts and notifications

## 9 Settings

9.1 The Settings tab
9.2 Connection and saved hosts
9.3 Read only and Data refresh
9.4 Appearance and date format
9.5 Terminal settings
9.6 The console lock and Face ID lock
9.7 Notifications
9.8 Usage statistics and crash reports
9.9 Audit log
9.10 Arrange panes
9.11 VisorMon Pro
9.12 Signing out
9.13 Help, version and licenses
9.14 All settings and their defaults

## 10 Security and privacy

10.1 At a glance
10.2 How VisorMon connects
10.3 Your sign-in details
10.4 Certificates
10.5 Read-only mode
10.6 The audit log
10.7 Face ID lock and the console lock
10.8 Usage statistics and crash reports
10.9 What stays on your phone
10.10 Privacy policy

## 11 VisorMon Pro

11.1 What needs VisorMon Pro
11.2 The free trial and the plans
11.3 Subscribing
11.4 Restoring a purchase
11.5 Checking, changing or canceling
11.6 When the trial or subscription ends
11.7 The demo and VisorMon Pro
11.8 Questions

## 12 Troubleshooting

12.1 Start here
12.2 Messages when connecting or signing in
12.3 Login expired, or VisorMon opens at the sign-in screen
12.4 "Could not refresh" and saved data
12.5 Grayed-out actions and empty lists
12.6 Shell and console problems
12.7 Notifications
12.8 Locks
12.9 Other questions
12.10 Getting help

## 13 Quick reference

13.1 Where to find things
13.2 Colors and states
13.3 Glossary
13.4 Support

CHAPTER 1

# Welcome to VisorMon

What VisorMon does, what you need, and a tour of the app.

## 1.1 What VisorMon does

VisorMon puts your Proxmox VE servers in your pocket. It shows how your cluster, nodes, virtual machines (VMs), containers and storage are doing, and lets you act on them from your iPhone or Android phone:

- **See what needs attention.** The dashboard sums up the cluster's health, its nodes and the guests under the most load.
- **Control your guests.** Start, shut down, reboot and stop VMs and containers, open their consoles, take snapshots and backups, migrate and clone them, and change their hardware and options.
- **Look after your nodes.** Open a node's shell, check its services, updates, network, disks and firewall, and reboot or shut it down.
- **Find anything.** Search the whole cluster, follow tasks as they run, and read their logs.

VisorMon talks straight to your Proxmox server from your phone, through the same API the Proxmox web interface uses. What you can see and do depends on the permissions of the account you sign in with, just as it does on the web.

## 1.2 What you need

- A Proxmox VE server or cluster that your phone can reach, on your network or over a VPN.
- A Proxmox account, or better, an API token, with the permissions for what you want to do.
- VisorMon on an iPhone or an Android phone.

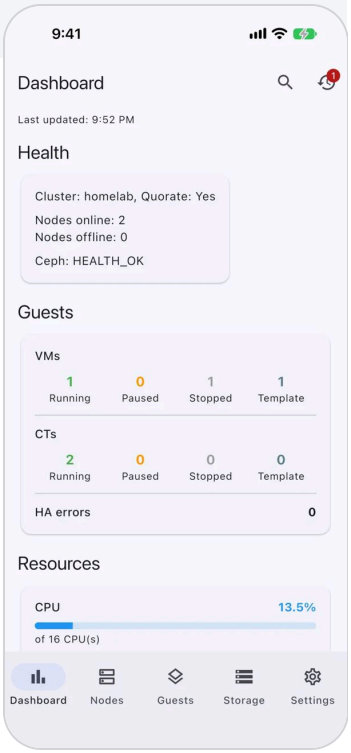
To look around before connecting a server, try the demo, which runs on sample data. Getting started walks you through signing in and the demo.

## 1.3 The app at a glance

Five tabs along the bottom of the screen take you to each part of your cluster:

| Tab | What it is for |
|---|---|
| **Dashboard** | The whole cluster at a glance. See The dashboard. |
| **Nodes** | Each server in the cluster: readings, power, shell and administration. See Nodes. |

| Tab | What it is for |
|---|---|
| **Guests** | Your VMs and containers: find one and act on it. See Guests. |
| **Storage** | Your storage and what is on it. See Storage. |
| **Settings** | How VisorMon behaves, your servers and your subscription. See Settings. |



The Dashboard tab of the demo cluster.

At the top of the main tabs are up to three buttons:

**Server** Once you have signed in to a server, the server button appears. Tap it to see which server you are on, switch to another saved server, or **Add a server**. It is not shown in the demo.

**Search** **Search cluster** finds nodes, guests and storage by name, tag and more. See Search, tasks and alerts.

**Tasks** **Cluster tasks** lists recent tasks and their logs. A number on the button shows how many tasks are running.

The dashboard and the guest lists refresh on their own, as often as **Settings › Data refresh** says. Pull down on a list to refresh it straight away.

## 1.4 How this guide is organized

The chapters follow the app, from your first sign-in to troubleshooting:

| | Chapter | What it covers |
|---|---|---|
| 1 | **Welcome to VisorMon** | What VisorMon does, what you need, and a tour of the app. |
| 2 | **Getting started** | First launch, signing in to your Proxmox server, demo mode and managing servers. |
| 3 | **The dashboard** | Your cluster at a glance: health, guests, resources, nodes and subscriptions. |
| 4 | **Nodes** | A node's readings and history, power actions, the shell, services, updates, network, disks and more. |
| 5 | **Guests** | The guest overview, the guests dashboard and the Guests tab: find guests, see how they are doing, act on them. |
| 6 | **Working with a guest** | A guest's page: power, console, snapshots, backups, hardware, migrate, clone and creating guests. |

| | Chapter | What it covers |
|---|---|---|
| 7 | **Storage** | Your storage, what is on it, and what you can do with it. |
| 8 | **Search, tasks and alerts** | Searching the cluster, following tasks and their logs, notifications and alerts. |
| 9 | **Settings** | Every setting, what it does and its default. |
| 10 | **Security and privacy** | How VisorMon protects your servers and your data: credentials, read-only mode, the audit log and locks. |
| 11 | **VisorMon Pro** | What is free, what Pro adds, and managing your subscription. |
| 12 | **Troubleshooting** | What error messages mean and how to fix common problems. |
| 13 | **Quick reference** | Where to find things, the colors and states the app uses, and a glossary. |

## 1.5 Conventions in this guide

- Words you see on screen are in bold, such as **Pin guests…**. A path such as **Settings › Arrange panes** means: open Settings, then tap Arrange panes.
- Proxmox privileges and other exact values are in a fixed-width font, such as `VM.PowerMgmt`.
- The screenshots show the iPhone app connected to the demo cluster. On Android the screens are the same; only the phone's own status bar and controls differ.

> **Need help?** Email support@jimmyfatmasterson.com. Troubleshooting explains the most common error messages and how to fix them.

CHAPTER 2

# Getting started

First launch, signing in to your Proxmox server, demo mode and managing servers.

## 2.1 Before you start

VisorMon connects straight from your phone to your Proxmox VE server. There is no VisorMon server in between, so check these things first:

- **A phone.** An iPhone with iOS 15 or later, or an Android phone with Android 8.0 or later.
- **A network path to the server.** On the same Wi-Fi or LAN as the server, VisorMon just works. Away from home, connect the phone to your network with Tailscale or a similar VPN. Keep the Proxmox web interface (port 8006) off the internet.
- **The server's address.** The address you use for the Proxmox web interface, with the port, for example `https://192.168.1.30:8006`. If you reach Proxmox through a reverse proxy, use the proxy's address instead.
- **A way to sign in.** An API token (recommended; see Creating an API token) or a Proxmox username and password.
- **VisorMon Pro or its free trial**, to use your own server. The demo is free and needs nothing. See VisorMon Pro.

> **Quick check.** Open the server's address in your phone's web browser. If the Proxmox login page loads (perhaps after a certificate warning), VisorMon can reach the server too.

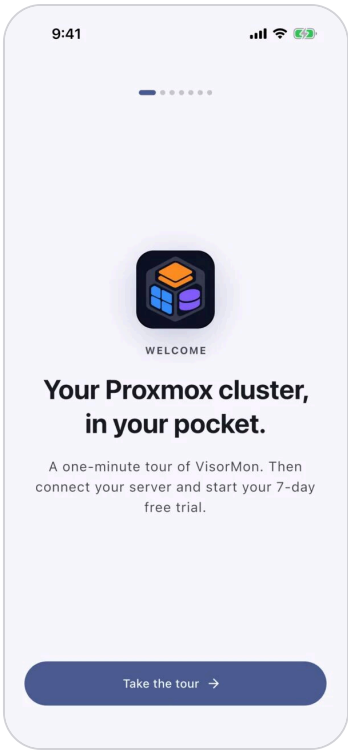
## 2.2 First launch

The first time you open VisorMon, it shows a welcome screen and a short tour of the app.

In the European Union, the rest of the European Economic Area, the United Kingdom and Switzerland, VisorMon first asks **Help improve VisorMon?**: whether it may send anonymous usage statistics and crash reports. Tap **Allow** or **Don’t allow**. Elsewhere VisorMon does not ask. Either way you can change your answer later; see Security and privacy.

### The tour

Tap **Take the tour** on the welcome screen. Move through the pages with the button at the bottom, or swipe. To go back a page, tap the back arrow at the top left, or use Back on Android. The dots at the top show where you are.




The welcome screen.

| Page | What it covers | Button |
|---|---|---|
| **See your whole cluster at a glance.** | CPU, memory and storage across every node | **Got it** |
| **Every VM and container, one tap away.** | Power actions, snapshots, backups and consoles | **Makes sense** |
| **Built so a phone can't break things.** | Confirmations, read-only mode, the app lock and alerts | **Continue** |
| **Your server has to be reachable.** | Reaching the server at home and away | **Good to know** |
| **Connect your server.** | Signing in, or skipping it for now | **Skip for now**, or **Continue** once you are signed in |
| The last page | VisorMon Pro and the free trial | See below |

## Signing in during the tour

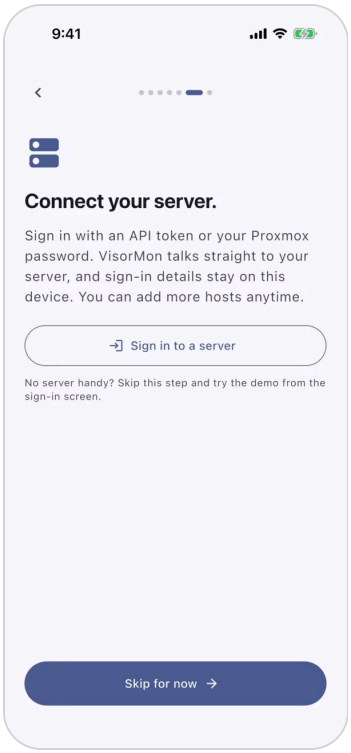
On **Connect your server.**, tap **Sign in to a server** to open the sign-in form (see Signing in). It closes by itself once you are signed in, and the page then shows which server and user you are signed in as. The form also has **Try demo**; if you choose it, the page says **Exploring the demo with sample data** and offers **Sign in to a real server**.

You do not have to sign in during the tour. Tap **Skip for now** and sign in when the tour is over.

### The end of the tour

The last page offers VisorMon Pro and its 7-day free trial. You can start the trial there, or finish the tour without it, for example to try the demo first. VisorMon Pro explains this page and the plans.

When the tour is over you go to the sign-in screen. If you signed in during the tour, you go to your server instead, or to the paywall if you have no trial or subscription yet. If you chose the demo, you go to the demo. To see the tour again, tap **Settings › Replay onboarding**.



Sign in now, or skip for now.

## 2.3 Creating an API token

An API token is the recommended way to sign in. It is separate from your password, you decide what it may do, and you can delete it at any time without changing your password. You create it in the Proxmox web interface.

1. In the Proxmox web interface, select **Datacenter**, then **Permissions › API Tokens**, and click **Add**.
2. Choose the **User** the token belongs to, for example `root@pam`, and type a **Token ID**, for example `visormon`.
3. Leave **Privilege Separation** ticked. The token then has only the permissions you give the token itself. If you untick it, the token has all of its user's permissions.
4. If you want the token to stop working on a set date, fill in **Expire**.
5. Click **Add**. Proxmox shows the token's ID and its secret. Copy the secret now: Proxmox shows it only once.
6. If privilege separation is on, give the token its permissions: go to **Datacenter › Permissions**, click **Add › API Token Permission**, set **Path** to `/`, choose your token and a **Role** from the table below, leave **Propagate** ticked, and click **Add**. Repeat for a second role if you need one.

| To let VisorMon | Give the token |
|---|---|
| Show everything without changing anything | `PVEAuditor` |
| Also start, stop and manage VMs and containers | `PVEAuditor` and `PVEVMAdmin`, as two permissions |
| Do everything it can, including on nodes and storage | `Administrator` |

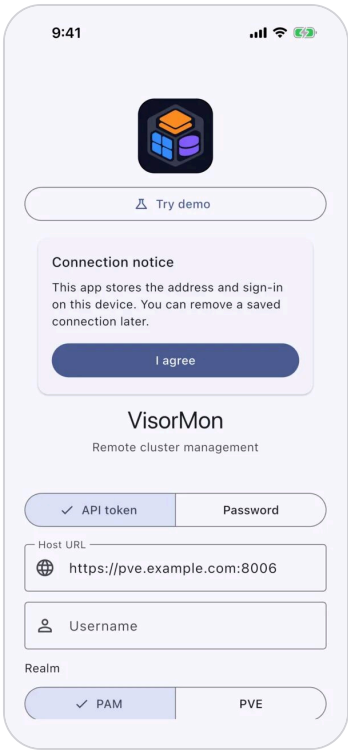
A token with privilege separation can never do more than its user, so the user needs these permissions too (`root@pam` has them all). Some jobs also need privileges on a storage, for example creating a guest or making a backup. The Proxmox documentation on user management lists every role and privilege.

VisorMon reads the permissions when you sign in and grays out what the token cannot do, instead of letting an action fail.

## 2.4 Signing in

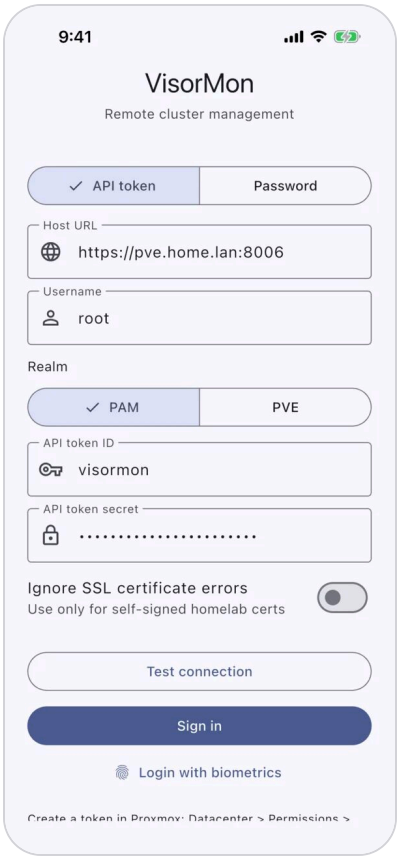
The sign-in screen appears after the tour, and whenever you are not signed in. It has two ways in, **API token** and **Password**, chosen at the top of the form.

The first time, the screen shows a **Connection notice**: **This app stores the address and sign-in on this device. You can remove a saved connection later.** Tap **I agree** to put it away.

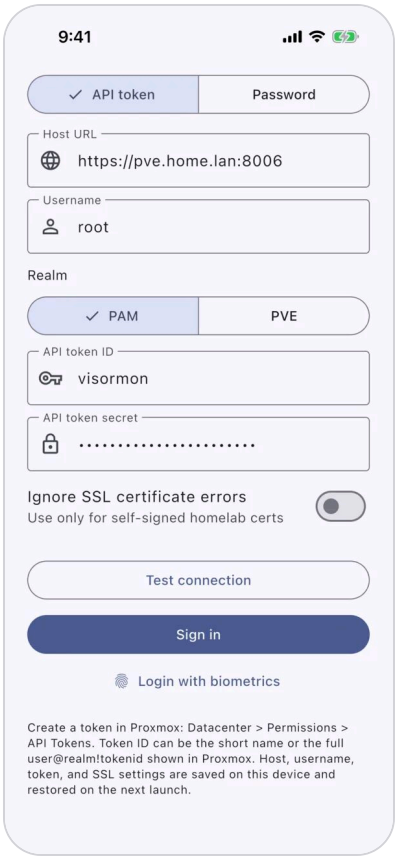



The sign-in screen the first time, with the connection notice.

### With an API token




The form filled in for an API token.




Test the connection, then sign in.

1. Leave **API token** selected at the top of the form.
2. **Host URL**: your server's address with the port, for example `https://pve.home.lan:8006`. The field starts with an example address; replace it. If you leave out `https://`, VisorMon adds it. It does not add the port, so type `:8006` yourself.
3. **Username**: the user the token belongs to, without the realm, for example `root`.

4. **Realm**: **PAM** for Linux accounts on the server, such as `root`, or **PVE** for users created in Proxmox.
5. **API token ID**: the token's short name (`visormon`) or the full ID that Proxmox shows (`root@pam!visormon`). Both work. With the full ID, the realm in it is the one used.
6. **API token secret**: the secret you copied from Proxmox.
7. If your server has a self-signed certificate, turn on **Ignore SSL certificate errors** (see Self-signed certificates).
8. Tap **Test connection** to check the details, then tap **Sign in**.

The buttons stay gray until the address, username, token ID and secret are all filled in. Once you are signed in, the dashboard opens (see The dashboard), unless your cluster shows a consent notice first, or you have no trial or subscription yet.

## Test connection

**Test connection** checks that VisorMon can reach the server and that Proxmox accepts your details, without signing you in or saving anything. If it works, it shows the server's version, for example **Proxmox VE 8.2.4 (8.2)**. If not, it says what went wrong in red, for example **Authentication failed. Check the username, realm, password, or API token.** Troubleshooting explains each message.

## With a password

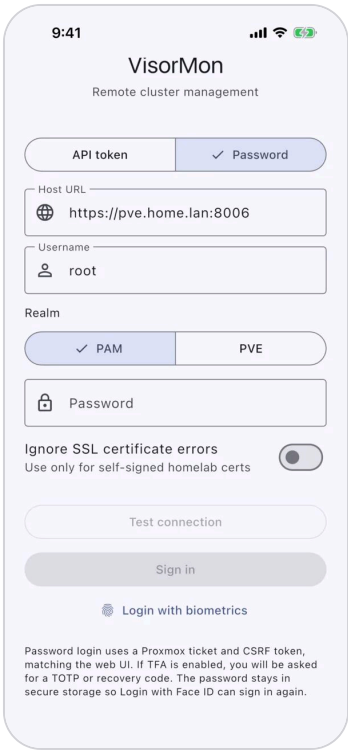
Password sign-in works like the Proxmox web interface: VisorMon asks Proxmox for a login ticket with your username and password.

1. Tap **Password** at the top of the form.
2. Fill in **Host URL**, **Username** and **Realm** as for a token, and type your **Password**.
3. Turn on **Ignore SSL certificate errors** if you need to, tap **Test connection**, then tap **Sign in**.

VisorMon keeps the password in the phone's secure storage, so that **Login with Face ID** can sign you in again (see Signing in with Face ID).

**Two-factor authentication.** If your account uses two-factor authentication, Proxmox checks your password first and then asks for a second factor. VisorMon says **Two-factor authentication required. Enter a TOTP or recovery code.**, adds a **TOTP or recovery code** field, and the **Sign in** button becomes **Verify**.

1. Type the current code from your authenticator app (a time-based one-time password, or TOTP), or one of your Proxmox recovery keys, which look like `1a2b-3c4d-5e6f-7a8b`.
2. Tap **Verify**. VisorMon tells a code and a recovery key apart by their form, and signs you in.



Password mode.

If the code is wrong or has expired, VisorMon says **Authentication failed. Check the username, realm, password, or API token.** and keeps the field: type the next code and tap **Verify** again. **Login with Face ID** asks for the code in the same way, every time. A recovery key works only once, as in the Proxmox web interface.

VisorMon cannot use a security key (WebAuthn or U2F) or a YubiKey one-time password. If your account has a security key and also TOTP or unused recovery keys, VisorMon asks for one of those. If it has nothing VisorMon can use, VisorMon says so instead of asking for a code, for example **This account requires WebAuthn, which is not supported yet. Use TOTP or an API token.** (see Troubleshooting). An API token avoids the second step altogether, because Proxmox does not ask a token for a second factor.

## API token or password?

| | API token | Password |
|---|---|---|
| What you can do | What you allow the token | Everything your user may do |
| How long you stay signed in | Until you sign out, or the token is deleted or expires | As long as Proxmox's login ticket lasts (Proxmox ends it after two hours); then you sign in again |
| Two-factor authentication | Not asked | Asked for, if your account uses it |
| Container consoles and node shells | Not available: Proxmox does not let API tokens open them | Available (a node's login shell needs `root@pam`) |
| VM consoles | Available | Available |

If you need a container's console or a node's shell, sign in with a password. Consoles and shells are covered in Working with a guest and Nodes.

## Self-signed certificates

A new Proxmox server uses a self-signed certificate, which your phone does not trust. Sign-in then fails with **The secure connection to the server failed. If it uses a self-signed certificate, turn on “Ignore SSL certificate errors” and try again.** You have two choices:

- **Give Proxmox a trusted certificate**, for example from Let's Encrypt with Proxmox's built-in ACME support (on the node, under **System › Certificates**). Then no switch is needed.
- **Turn on Ignore SSL certificate errors** on the sign-in form. VisorMon then accepts the server's certificate without checking it. As the switch says, use it only for self-signed homelab certificates, on a network you trust.

The switch is saved with each server, so you set it once per server.

### Cluster consent notice

If your cluster has consent text set (in the Proxmox web interface, under **Datacenter › Options**), VisorMon shows it after you sign in, on a screen called **Cluster consent**: **This cluster requires you to accept the following notice before continuing.** Tap **I agree** to go on, or **Sign out**. VisorMon remembers your answer for each server and asks again only if the text changes.

### Signing in with Face ID

When your phone can use it, the sign-in screen has a button below **Sign in**: **Login with Face ID** on an iPhone with Face ID, and **Login with biometrics** on other phones. It confirms it is you, then signs in to the selected server with its saved token or password, so you don't have to type anything. It works once you have signed in to that server at least once.

This is separate from the Face ID lock, which protects the whole app (see below).

### Opening VisorMon again

Next time you open VisorMon, it signs back in to the server you used last and opens the dashboard. Otherwise:

- **The sign-in has ended**, for example because a password ticket ran out or the token was deleted. The sign-in screen says **Login expired** and **Your session ended. Sign in again to view this data.** Sign in again. The same happens if a sign-in ends while you are using the app.
- **The server cannot be reached**, for example because the phone is away from home without its VPN. The sign-in screen opens with the server selected. Connect, then sign in; a token's details are filled in for you, and a password can be typed again or replaced by **Login with Face ID**.
- **You turned on the Face ID lock** (**Settings › Face ID lock**, or **Biometric lock** on Android). VisorMon first shows **Unlock VisorMon** and asks for Face ID. If that fails, or you cancel, tap **Unlock** to try again, or **Continue without Face ID** (**Continue without biometrics** on Android) to open the app anyway. The lock is off until you turn it on; see Security and privacy.

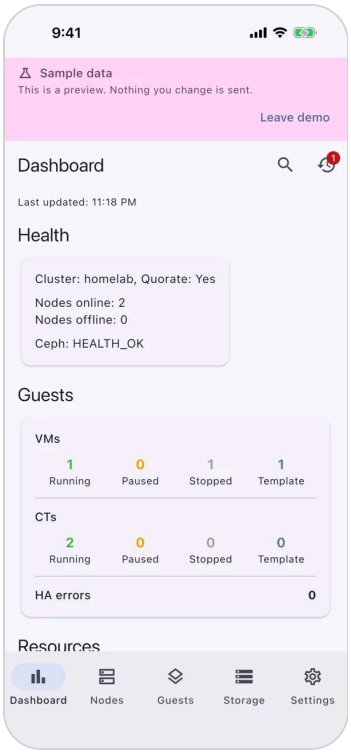
## 2.5 Trying the demo

The demo lets you explore VisorMon without a server. Tap **Try demo** at the top of the sign-in screen. VisorMon opens on a sample cluster called `homelab`, with two nodes, a few VMs and containers, a template, storage and tasks.

- Every tab and screen works, on sample data. VisorMon does not contact any Proxmox server.
- A **Sample data** banner stays at the top of the screen: **This is a preview. Nothing you change is sent.**
- Actions that would change something are not carried out. VisorMon says **This is sample data. Nothing was sent.**

- Consoles and shells do not open: **Sample data has no live console. Sign in to a Proxmox server to open one.**
- The demo is free and needs no trial or subscription (see VisorMon Pro).

To leave the demo, tap **Leave demo** in the banner, or **Settings › Sign out**. You go back to the sign-in screen, with any server you had saved still there.



The demo, with the Sample data banner.

## 2.6 More than one server

VisorMon can keep any number of servers, and you can switch between them without typing your details again.

### Saved hosts

VisorMon saves a server, or *host*, the first time you sign in to it. A sign-in that fails does not save anything. Each saved host keeps its address, username, realm, sign-in method, token ID and certificate setting; its token secret, password and current sign-in are kept in the phone's secure storage (see Security and privacy). A host is saved once for each address, username and realm, so the same server with two different users is saved twice.

Hosts are named *user@address*, for example **root@pve.home.lan**.

### Switching servers

Tap the server button at the top of the **Dashboard**, **Nodes**, **Guests**, **Storage** and **Settings** tabs, to the left of the search button. It appears once you have a saved host. It lists every saved host, last used first, with **(current)** after the one you are using. Tap another host to switch to it.

VisorMon signs out of the current server and signs in to the one you chose:

- A host that uses an **API token** signs in straight away with its saved token.
- A host that uses a **password** signs in only while its last Proxmox login is still valid. Otherwise the sign-in screen opens with that host selected: type the password, or use **Login with Face ID**, and tap **Sign in**.
- If the host cannot be reached, the sign-in screen opens with that host selected.

You can also switch in **Settings › Saved hosts**: tap any host that is not marked **current**.

### Adding a server

1. Tap the server button and choose **Add a server**, or tap **Settings › Add a server**.

2. The sign-in screen opens under the heading **Add a server**, with a new form. Fill it in as in Signing in, and tap **Sign in**.

In the demo, **Add a server** leaves the demo and opens the ordinary sign-in screen.

The new server is saved and becomes the current one. To go back to the server you had open without adding one, tap **Back to** and that server's name at the top of the screen, or use Back on Android. If VisorMon cannot sign back in, it says **Could not sign back in to** followed by the server's name, and the form shows that server so you can sign in.

On the sign-in screen itself, when you are signed out, tap **Add host** below the saved hosts to clear the form for a new server.

### Changing or removing a server

When you are signed out, the sign-in screen lists your **Saved hosts**, last used first. Tap one to fill in the form with its details. To change a host, select it, edit the fields and sign in; the saved host is updated.

To remove a host and delete its saved sign-in details:

- **On the sign-in screen**, tap the bin button (**Remove host**) beside it.
- **In Settings › Saved hosts**, tap the bin button beside it.

Either way VisorMon asks **Remove host?** first; tap **Remove**, or **Cancel** to keep it. If it is the server you are signed in to, the dialog says that removing it signs you out, and the button reads **Remove and sign out**.

Removing a host only removes it from VisorMon. The server and the API token are not changed; to stop the token working, delete it in Proxmox under **Datacenter › Permissions › API Tokens**.

## 2.7 Signing out

1. Go to **Settings** and tap **Sign out**, below the VisorMon Pro section.
2. VisorMon asks **Sign out?**: **Disconnects this app from Proxmox. Saved credentials stay on this device.** Tap **Sign out**.

The sign-in screen opens with the server still selected. Its saved details stay on the phone, so you can sign in again quickly, but VisorMon will not sign in by itself the next time you open it. To delete a server's saved details as well, remove it (see Changing or removing a server).

**Sign out** is also on the **Cluster consent** screen and on the paywall you see after signing in without a trial or subscription. In the demo, **Leave demo** does the same.

CHAPTER 3

# The dashboard

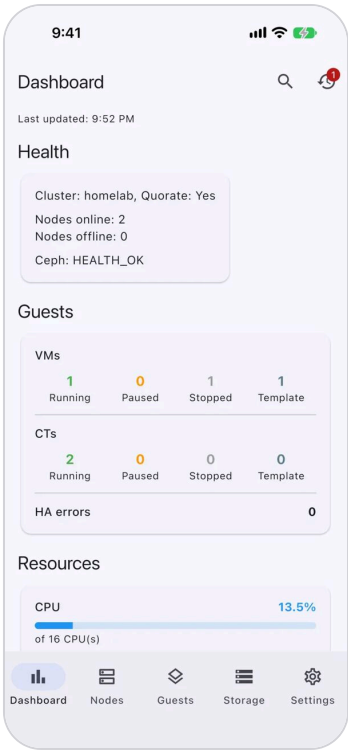
Your cluster at a glance: health, guests, resources, nodes and subscriptions.

## 3.1 What the dashboard shows

The **Dashboard** tab is the first screen you see after signing in. It answers the questions you would ask the Proxmox web interface's datacenter summary: is the cluster healthy, what is running, how much capacity is left, and is every node up.

The dashboard is made of panes, one below the other. In their default order they are:

| Pane | Shows |
|---|---|
| **Health** | The cluster's name and quorum, how many nodes are online and offline, and Ceph health |
| **Guests** | How many VMs and containers are running, paused, stopped or templates, and how many HA errors there are |
| **Resources** | CPU, memory and storage in use across the cluster |
| **Nodes** | A card for each node: online or offline, CPU, memory and uptime |
| **Guest overview** | Five guests as cards with their readings |
| **Subscriptions** | The cluster's Proxmox support subscription level |




The top of the dashboard.

You can hide panes you do not use and put the rest in the order you like (see Hiding and reordering panes).

### Health

The first line describes the cluster:

- **Cluster: homelab, Quorate: Yes** for a cluster, with its name and whether it has quorum. **Quorate: No** means too few nodes can see each other for the cluster to make changes.
- **Standalone node — no cluster defined** for a single server that is not part of a cluster.
- **Cluster status is not permitted for this account.** when your account may not read the cluster's status (see Accounts that see part of the cluster).

Below it, **Nodes online** and **Nodes offline** count the nodes. If the cluster runs Ceph, a last line shows Ceph's own health word, for example **Ceph: HEALTH_OK** or **Ceph: HEALTH_WARN**. Without Ceph the line is left out. If your account may not read Ceph's status, it reads **Ceph: not permitted**.

### Guests

Two rows, **VMs** and **CTs** (containers), each count their guests as **Running** (green), **Paused** (orange), **Stopped** (gray) and **Template**. A guest whose state Proxmox cannot report, such as one on a node that is down, counts as stopped here.

**HA errors** is the number of guests whose high availability (HA) state is error. It turns red when it is above 0. Check those guests in Proxmox.

### Resources

Three bars, each with a percentage:

| | |
|---|---|
| **CPU** | How much of the cluster's CPU is in use, with the number of CPUs below, for example **of 16 CPU(s)**. |
| **Memory** | Memory in use across the nodes, for example **11.0 GB of 32.0 GB**. |
| **Storage** | Space used across the cluster's storage. A shared storage counts once, not once per node, as in the Proxmox web interface. Storage that is not available is left out. |

### Nodes

Each node has a card, in name order, with:

- its name and **online**, or **offline** in red,
- its IP address, when Proxmox reports it,
- a **CPU** bar, and a **Memory** bar with the memory used and the total,
- how long it has been up, for example **Uptime 10d 0h**.

Tap a node's card to open its page, with its full readings, history charts and actions. See Nodes.

### Guest overview

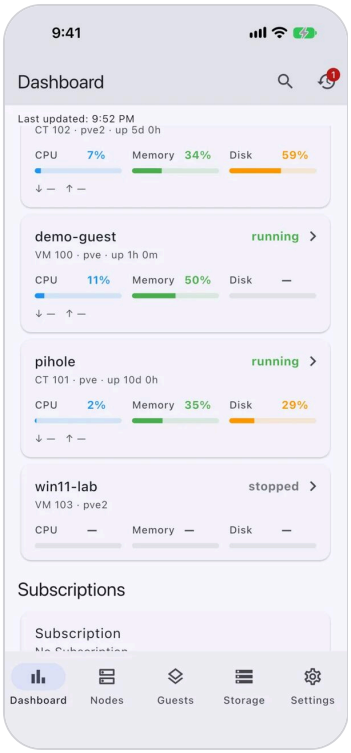
The guest overview, headed **Guests** on the dashboard, shows five guests as cards with their CPU, memory, disk and network readings, chosen in the order you pick and with any guests you have pinned first. It has its own chapter, which covers choosing the order, pinning, the critical count and **View all**: see Guests.

The overview is left out when the cluster has no guests other than templates. Note that the dashboard then has two panes headed **Guests**: the counts near the top and the overview further down. In **Settings › Arrange panes** they are called **Guests** and **Guest overview**.

### Subscriptions

The **Subscription** line shows the Proxmox support subscription on the cluster's nodes: **Community**, **Basic**, **Standard**, **Premium** or **No Subscription**. When the nodes have different levels, it reads **Mixed**. Each node's subscription details are on its page, under **System › Subscription** (see Nodes).

This is your subscription with Proxmox, not your VisorMon subscription. For that, see VisorMon Pro.




Further down: the guest overview and Subscriptions.

## 3.2 Accounts that see part of the cluster

VisorMon reads your account's permissions from Proxmox when you sign in. If your account, or your API token, does not have `Sys.Audit` on the whole datacenter (the path `/`), the dashboard is a **scoped view**: it shows only what Proxmox lets your account see.

- A line under the **Health** heading says **Showing only resources this account can access.**
- The **Guests** pane's rows read **VMs you can access** and **CTs you can access**, and count only those guests.
- **Resources** and **Nodes** cover only the nodes and storage your account can see.
- Reading the cluster's status needs `Sys.Audit` on `/`, so the first Health line usually reads **Cluster status is not permitted for this account.** and the Ceph line reads **Ceph: not permitted**.

This is normal for an account set up to manage only some guests or one node. To see the whole cluster, give the account or token `Sys.Audit` on `/` in Proxmox (the PVEAuditor role includes it). For more on permissions, see Security and privacy.

> **API tokens.** A token with privilege separation has only the permissions given to the token itself, not all of its user's. If the dashboard is scoped when you expected the whole cluster, check the token's permissions in Proxmox under Datacenter › Permissions.

## 3.3 Keeping the figures up to date

**Last updated**, at the top of the dashboard, shows the time the figures were last fetched from your server.

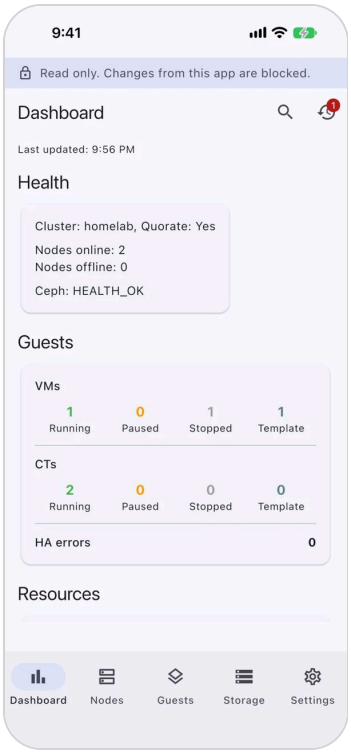
- **Automatically:** the dashboard refreshes on the schedule set in **Settings › Data refresh**: about every 5 seconds (**Real-Time**), every 30 seconds (**Balanced**, the default) or every 5 minutes (**Battery Saver**). See Settings.

- **Now:** pull down on the dashboard to refresh it straight away.

### Banners above the dashboard

A banner across the top of the screen, above every main tab, tells you when something is different from normal:

| | |
|---|---|
| **Read only. Changes from this app are blocked.** | Read-only mode is on. You can look at everything, but VisorMon sends no changes. See Security and privacy. |
| **Sample data** | You are in the demo. Tap **Leave demo** to sign in to a real server. See Getting started. |
| **Showing saved data from a date and time** | The last refresh failed, for example because your phone has left your home network. The dashboard keeps showing what it last loaded, and **Last updated** stays at the time of that load. |
| **Needs attention: Could not refresh** | Appears with the saved-data banner. Tap it to see the alert. If **Enable notifications** is on in Settings, your phone also shows one **Could not refresh** notification. |




The banner shown while read-only mode is on.

The saved-data banner and the **Needs attention** card go away as soon as a refresh succeeds. If nothing has loaded since you opened VisorMon, the dashboard shows the error itself, with a **Retry** button. For help with connection errors, see Troubleshooting.

## 3.4 The top bar and the tabs

### Buttons in the top bar

The **Dashboard**, **Nodes**, **Guests**, **Storage** and **Settings** tabs share the same buttons at the top right:

| Button | What it does |
|---|---|
| Server (a server icon) | Switches between the Proxmox servers saved on this device. The menu lists each one, marks the one in use **(current)**, and ends with **Add a server**. It appears once you have a saved server. See Getting started. |
| **Search cluster** (magnifier) | Searches the whole cluster. See Search, tasks and alerts. |
| **Cluster tasks** (clock) | Opens the cluster's recent tasks. A red badge shows how many are running. See Search, tasks and alerts. |

Touch and hold a button to see its name.

### The tabs at the bottom

| Tab | For |
|---|---|
| **Dashboard** | The cluster at a glance (this chapter) |
| **Nodes** | Each node, its readings, its shell and its settings. See Nodes. |
| **Guests** | Finding a VM or container and acting on it. See Guests. |
| **Storage** | Each storage, how full it is and what is on it. See Storage. |
| **Settings** | Your servers, safety, refresh, appearance and the terminal. See Settings. |

Each tab keeps its place when you switch away and back. The **Nodes**, **Guests** and **Storage** tabs do not refresh on a timer the way the dashboard does: pull down on one of them to bring it up to date.

## 3.5 When a node is down

In a cluster, the dashboard keeps working when one node is powered off or not answering. VisorMon does not wait for the node that is down; it loads the rest of the cluster as usual.

- **Nodes offline** in the Health pane counts the node.
- The node's card in the **Nodes** pane shows **offline** in red, with dashes or empty bars instead of its figures.
- Guests on that node count as stopped in the **Guests** pane, and show as **unknown** in the guest lists, until the node is back.
- Everything on the other nodes works as normal.

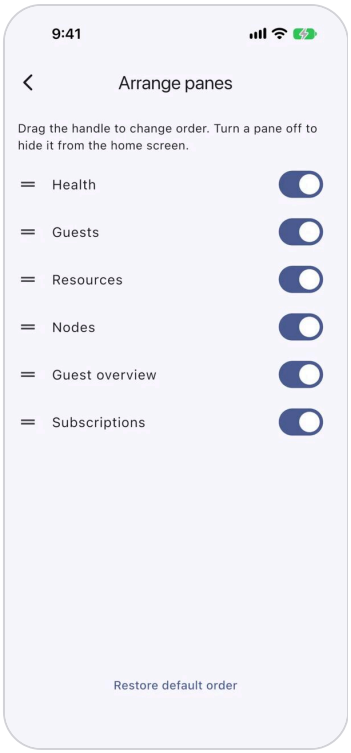
Tap the offline node's card to open its page. If the node has a Wake-on-LAN address saved, you can wake it from there. See Nodes.

> **The node in your server address.** VisorMon reaches the whole cluster through the node whose address you signed in with. If that node is the one that is down, nothing can load, and the dashboard shows the saved-data banner described above. To keep a way in, you can save another node of the same cluster as a second server and switch to it with the server button.

## 3.6 Hiding and reordering panes

1. Go to **Settings › Arrange panes**.
2. Turn a pane's switch off to hide it from the dashboard, or on to show it again.
3. Drag a pane by its handle to move it.

The dashboard changes as you go. **Restore default order** puts every pane back in the default order and turns them all on. Your arrangement is kept on this device. See Settings for the rest of the settings.




Settings › Arrange panes.

CHAPTER 4

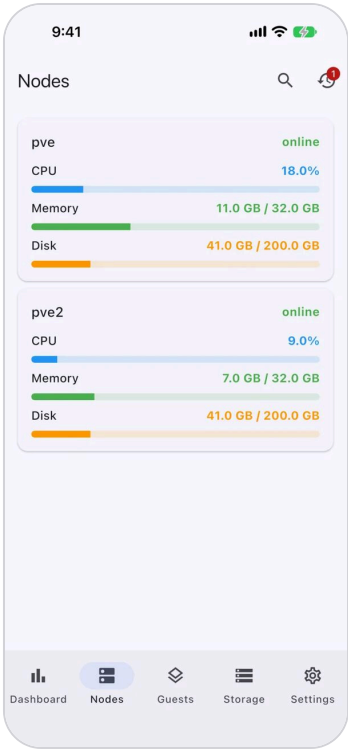
# Nodes

A node's readings and history, power actions, the shell, services, updates, network, disks and more.

## 4.1 The Nodes tab

The **Nodes** tab lists every node your account can see, in name order. Each card shows:

- the node's name and its state: **online**, or **offline** or **unknown** in red,
- **CPU**, as a percentage,
- **Memory** and **Disk** (the node's root disk), each as used and total, for example **11.0 GB / 32.0 GB**.



The Nodes tab.

Tap a card to open the node's page. The list does not refresh by itself: pull down on it to bring the figures up to date.

If the list is empty, the screen says **No permission** or **No nodes in your scope**: Proxmox shows this account no nodes. Give the account `Sys.Audit` on the nodes it should see.

You can also open a node's page from the **Nodes** pane on the dashboard (see The dashboard).

## 4.2 A node's page

A node's page matches the node summary in the Proxmox web interface, with the node's actions and tools in the middle. From top to bottom it has the node's readings, its **Power** and **Manage** buttons, its notes, and its history charts. Pull down to refresh the page; it does not refresh by itself.

Opening a node's page needs `Sys.Audit` on that node. Without it, the page shows **No permission** and a **Retry** button.

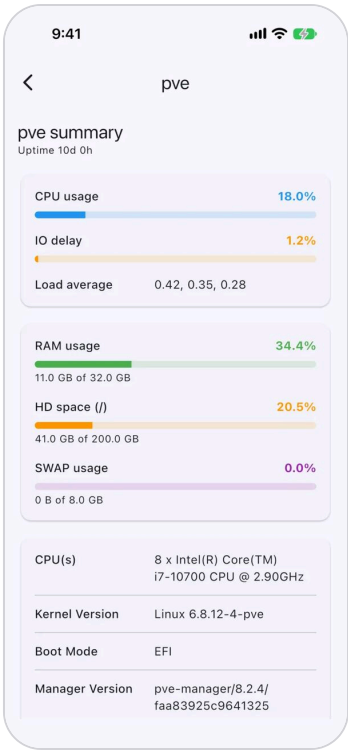
### Readings

Under the heading, for example **pve summary**, and the node's uptime:

**CPU usage** How busy the node's CPUs are.

**IO delay**

| | |
|---|---|
| | The share of time the CPUs spent waiting for disks. A high figure means storage is slowing the node down. |
| **Load average** | The usual Linux load figures for the last 1, 5 and 15 minutes. |
| **RAM usage** | Memory used, as a percentage and as used of total. **KSM sharing** appears below it when the node is sharing identical memory pages between guests. |
| **HD space (/)** | Space used on the node's root file system. |
| **SWAP usage** | Swap in use, or **None** when the node has no swap. |
| **CPU(s)** | How many CPUs and their model, for example **8 x Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz**. |
| **Kernel Version, Boot Mode, Manager Version** | The running kernel, **EFI** or **Legacy BIOS** (with **Secure Boot** when it is on), and the Proxmox VE version. |



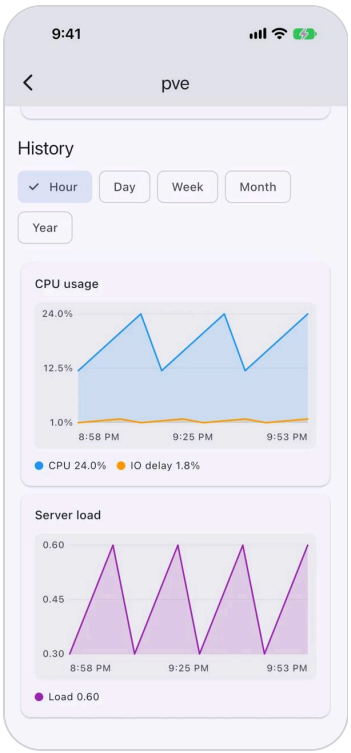

The top of a node's page.

## History charts

Under **History**, choose **Hour**, **Day**, **Week**, **Month** or **Year**. The page opens on **Hour**. The charts are the ones the Proxmox web interface draws:

- **CPU usage**: CPU and IO delay.
- **Server load**: the load average.
- **Memory usage**: used, available, ZFS ARC and total. A line Proxmox does not record, such as ZFS ARC on a node without ZFS, is left out.
- **Network traffic**: in and out, per second.
- **IO pressure**: the share of time work on the node was held up waiting for CPU, disk IO or memory (**CPU some**, **IO some**, **Memory some**). This chart appears only when Proxmox records pressure figures.

Each chart's legend shows the latest value of each line. History needs `Sys.Audit` on the node; without it the section reads **History is not permitted for this account.** If Proxmox has nothing for the period you chose, it reads **No history data available for this timeframe.**




History, over the last hour.

### Notes

If your account has `Sys.Audit` on the whole datacenter (`/`), the page shows the node's notes, the same notes as on the node's Summary in the Proxmox web interface, or **No notes yet.** Tap **Edit** to change them, which needs `Sys.Modify` on `/`. Without that privilege the button reads **Open** and shows the notes read-only. See Notes and options.

### When the node is offline

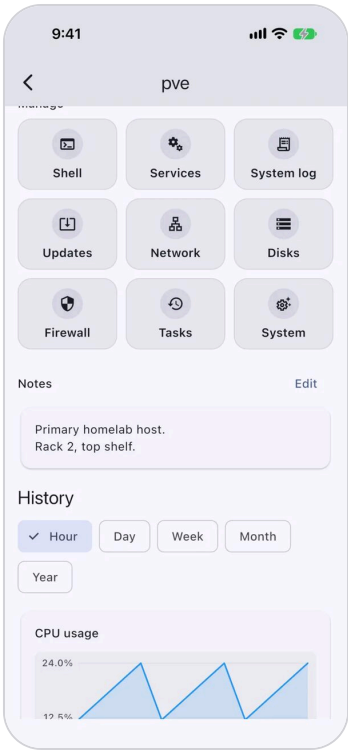
The page of a node that is off or not answering shows ***name* is offline** instead of its readings, with an error, usually **The node did not answer (HTTP 595). It may be powered off, or the server this app is connected to cannot reach it.**

If the node has a Wake-on-LAN MAC address saved in its options, tap **Wake on LAN**. The node VisorMon is connected to sends the wake-up packet, and VisorMon confirms with a message such as **Wake-on-LAN packet sent to AA:BB:CC:DD:EE:FF. pve2 needs a minute or two to start.** Pull down a little later to see whether it is back.

- Waking a node needs `Sys.PowerMgmt` on it.
- Save the MAC address while the node is running: the page of an offline node shows only this card, so you cannot reach its options then. See Notes and options.
- The node's network card and BIOS must have Wake-on-LAN turned on.

## 4.3 Actions and sections

Below the readings, a node's page has two groups of buttons. **Power** acts on the node and its guests. **Manage** opens the node's tools.




A node's Power and Manage buttons.

| Button | What it does | Proxmox privilege |
|---|---|---|
| **Reboot** | Restarts the node | `Sys.PowerMgmt` on the node |
| **Shutdown** | Powers the node off | `Sys.PowerMgmt` on the node |
| **Start all** | Starts the node's guests that are set to start at boot | `VM.PowerMgmt` on `/` |
| **Stop all** | Shuts down every guest on the node | `VM.PowerMgmt` on `/` |
| **Suspend all** | Suspends every VM on the node to disk | `VM.PowerMgmt` and `VM.Config.Disk` on `/` |
| **Migrate all** | Moves every guest to another node | `VM.Migrate` on `/` |
| **Shell** | A terminal on the node | `Sys.Console` on the node |
| **Services** | The node's system services | `Sys.Audit` on the node to view, `Sys.Modify` to control |
| **System log** | The node's journal | `Sys.Syslog` on the node |
| **Updates** | Package updates, and the upgrade | `Sys.Modify` on the node |
| **Network** | The node's network interfaces | `Sys.Audit` on the node to view, `Sys.Modify` to change |

| Button | What it does | Proxmox privilege |
|---|---|---|
| **Disks** | Disks, SMART, and storage made from disks | `Sys.Audit` on the node to view, `Sys.Modify` on `/` to change |
| **Firewall** | The node's firewall options and rules | `Sys.Audit` on the node to view, `Sys.Modify` to change |
| **Tasks** | Recent tasks on this node | Always available |
| **System** | DNS, hosts, time, subscription, certificates, repositories and more | Always available; each entry has its own (see System) |

VisorMon reads your account's permissions from Proxmox and grays out a button your account cannot use, with the reason under it, such as **No power permission on this node** or **No guest power permission**. While a power action is being sent, a progress bar appears under the **Power** buttons.

> **Read-only mode.** With **Read only** on in Settings, VisorMon sends no changes. Every action in this chapter that changes something, from **Reboot** to saving a DNS server, stops with **Read only is on. Turn it off in Settings to make this change.** and is recorded in the audit log. Opening the node shell counts as a change, so it is blocked too. Viewing still works. See Security and privacy.

## 4.4 Rebooting or shutting down a node

Both ask first, and say what will happen to the node's guests:

| Button | VisorMon asks | Then |
|---|---|---|
| **Reboot** | **Reboot node**: **Reboot *name*? Its guests are shut down first, and the node is unreachable until it is back up.** | **Reboot requested for *name*.** |
| **Shutdown** | **Shut down node**: **Shut down *name*? Its guests are stopped, and the node stays off until someone powers it on at the machine.** | **Shutdown requested for *name*.** |

Proxmox accepts the request at once and takes the node down afterwards, so the node shows as offline a little later. A node that is shut down can be started again with **Wake on LAN** if it is set up for it (see When the node is offline).

> **The node you are connected to.** VisorMon reaches the cluster through the node in your server address. If you reboot or shut down that node, VisorMon loses its connection until the node is back.

## 4.5 Starting, stopping, suspending or migrating every guest

These four buttons act on all the guests on the node at once. Each asks first:

| Button | VisorMon asks |
|---|---|
| **Start all** | **Start all guests: Start every guest on *name* that is set to start at boot?** |
| **Stop all** | **Stop all guests: Shut down every guest on *name*? A guest that has not stopped after 3 minutes is powered off.** |
| **Suspend all** | **Suspend all VMs: Suspend every VM on *name* to disk? Each VM's memory is saved to storage and the VM stops running; start it again to resume. Containers are not affected.** |

**Migrate all** opens **Migrate all guests** instead: **Move every guest on *name* to another node. Running guests are moved live when their storage allows it.** Choose the node under **Move to**; only online nodes are offered. Turn on **Include local disks** (**Live storage migration. Slower.**) to move guests whose disks are on the node's own storage. Then tap **Migrate all**. If no other node is online, VisorMon says **There is no other online node to move guests to.**

When the request is sent, VisorMon shows a message such as **Starting guests on pve.** with a **Tasks** button that opens the node's tasks, where you can follow each guest.

- **Start all**, **Stop all** and **Suspend all** leave out templates and guests that high availability (HA) manages; HA looks after those. **Migrate all** moves those too.
- To choose which guests **Start all** starts, turn on **Start at boot** in each guest's options (see Working with a guest).

> **Permissions.** Proxmox carries out these four actions only for an account that has the privilege on the whole datacenter ( `/` ): `VM.PowerMgmt` for Start all and Stop all, `VM.PowerMgmt` and `VM.Config.Disk` for Suspend all, and `VM.Migrate` for Migrate all. VisorMon shows the buttons to an account that has the privilege on any guest, so an account with rights on only some guests can tap them and get a permission error from Proxmox.

## 4.6 The node shell

**Shell** opens a terminal on the node itself, like **Shell** in the Proxmox web interface. Use it for anything VisorMon has no button for. The terminal is built for a phone: a row of extra keys, copy and paste, find, saved commands, and an option to keep your session alive in tmux.

### What the shell needs

- **A password sign-in.** Proxmox does not let API tokens open shells. Signed in with a token, the screen reads **The node shell needs a username and password login. Proxmox does not allow API tokens to open shells.** See Getting started.
- `Sys.Console` **on the node.** Without it the button is grayed out: **No console permission on this node**.

- **The root@pam account.** Proxmox logs you straight in as root only when you are signed in as root@pam. Any other account gets the node's own `login:` prompt. If Proxmox refuses to open the shell, VisorMon says **The node shell is only available when signed in as root@pam.**
- **Read only off.** See the note in Actions and sections.

The demo has no live shell: it shows **Sample data has no live console. Sign in to a Proxmox server to open one.** If you have turned on **Face ID for consoles** (on Android, **Biometric lock for consoles**), VisorMon asks your phone to confirm it is you before the shell opens. See Security and privacy.

## The terminal screen

The screen is titled with the node's name, for example **pve shell**. Your phone's screen stays on while the shell is connected.

| | |
|---|---|
| **Top bar** | **Find in output** (magnifier), **Reload console** (circular arrow), and **More** (⋮) with **Text size** and **Share log**. |
| **History strip** | Under the top bar, for example **History: 1,000 lines**, with **Change in Settings**. It ends in **(this terminal)** when you chose a longer history for this shell only. Inside tmux it reads **tmux keeps this session running if the connection drops. Scroll back with Ctrl-b, then [.** |
| **Terminal** | Swipe up and down to scroll back through the output. When you have scrolled back, a round ↓ button (**Jump to the latest output**) takes you to the newest line. |
| **Key bar** | Along the bottom, the keys a phone keyboard lacks (below). |

## The key bar

The key bar scrolls sideways to show more keys. A key acts when you lift your finger, so you can scroll the bar without pressing anything.

| Key | What it does |
|---|---|
| ⌨ | Shows or hides the phone's keyboard |
| **Esc**, **Tab** | Sends Escape or Tab. Tab completes commands and file names in the shell. |
| **Ctrl** | Holds Ctrl for the next key you type, and lights up blue until then. Tap **Ctrl**, then `d`, for Ctrl-D. |
| **^C** | Sends Ctrl-C straight away, to stop the running command |
| **Paste** | Types the text on your clipboard into the shell |
| **Copy** | Copies the selected text, or the whole screen when nothing is selected. It lights up while text is selected. |
| **Select** | Turns select mode on or off (see below) |
| **Cmds** | Opens your saved commands (see Saved commands) |

| Key | What it does |
|---|---|
| ↑ ↓ ← → | Arrow keys. Hold one to repeat it. Up and down step through the shell's command history. |
| **\| ~ / - ** | Types the character. These are awkward to reach on a phone keyboard. |
| **Home**, **End**, **PgUp**, **PgDn** | The usual keys, for programs such as `less` and `nano` |

## Copying and pasting

1. Tap **Select**. VisorMon says **Drag over the text to select it, then tap Copy.** While select mode is on, a finger selects instead of scrolling.
2. Press on the first word you want and drag to the last. Drag to the top or bottom edge to scroll and keep selecting.
3. Tap **Copy**. VisorMon confirms with, for example, **Copied the selection (3 lines)**, and turns select mode off.

Tap **Copy** without selecting anything to copy everything on the screen: **Copied the screen (24 lines)**.

**Paste** types your clipboard into the shell. If the clipboard holds several lines and the program in the shell would run each line as it arrives, VisorMon asks first: **Paste 3 lines? This shell may run each line as soon as it receives it, without waiting for you.** A shell that holds pasted text until you press Enter is not asked about.

## Finding text and sharing the log

Tap **Find in output** to search everything the terminal remembers. A bar opens at the top: type, and the phone's keyboard types into the search instead of the shell. Matches are not case sensitive. VisorMon starts at the newest match and shows where you are, for example **2 of 7**.

- ▲ goes to an older match, ▼ to a newer one. Enter also goes to the next older match.
- ✕, Esc or **^C** closes the search and returns your typing to the shell.

**More › Share log** saves everything the terminal remembers as a text file, named for the shell and the time, such as `pve-shell-2026-11-17-2130.log`, and opens your phone's share sheet, so you can save it to Files, mail it or send it to another app. If a full-screen program such as tmux or `vim` is open, what it shows is added at the end under `--- screen when saved ---`. The log holds whatever was on the screen, so check it before you share it.

## Saved commands

Saved commands are commands you type often, a tap away. Tap **Cmds** on the key bar to open **Saved commands**, then tap one. A command marked with a return arrow (**Runs right away**) is typed and run. One marked with a pencil (**Typed for you to finish**) is typed and waits, for a command that still needs

something added, such as a file name.

VisorMon starts you off with `apt update`, `journalctl -f`, `df -h`, `free -h`, `qm list` and `pct list`.

**To change the list,** tap **Edit list** in the sheet, or go to **Settings › Saved commands**:

- Tap **+** (**Add a command**) to add one. Enter the **Command** (one line), an optional **Name** for the list, and choose whether to **Run it right away**. Tap **Save**.
- Tap a command to change it, or to **Delete** it.
- Drag a command by its handle to reorder the list.
- If you remove them all, **Add some suggestions** brings the starting set back.

Saved commands are kept on this device. Container consoles use the same list.

## Resuming in tmux

Normally, when the connection drops, for example when your phone changes networks or locks, the shell on the node ends, and so does anything still running in it. With **Settings › Resume node shell with tmux** on, VisorMon keeps the shell in a tmux session on the node instead, so you can pick up where you left off.

- Each time the shell connects, VisorMon types a command that joins the tmux session named `visormon` on the node, or starts it. The command is cleared from the screen.
- After a dropped connection, tap **Reconnect**, or open the shell again, to return to the same session with your programs still running.
- Opening the shell takes the session over from anywhere else it is open.
- To end the session for good, type `exit` in it.
- tmux keeps its own history. To scroll back, tap **Ctrl**, then type `b`, then `[`; press `q` to leave.

The node needs tmux installed (`apt install tmux`). Without it the shell opens as usual and says **tmux is not installed on this node, so this shell ends when the connection drops.** The setting applies only to the node shell, not to container consoles or the shell that runs an upgrade.

## Scrollback and text size

| | |
|---|---|
| **Settings › Terminal scrollback** | How many lines of output the node shell and container consoles remember once they scroll off the screen: **1,000 lines (default)**, **5,000 lines**, **10,000 lines** or **50,000 lines**. A longer history uses more memory. The change applies at once, even to a shell that is open. **Change in Settings** on the history strip takes you to this setting. |
| **Settings › Terminal text size** | Also in the terminal under **More › Text size**. Tap **Smaller text** or **Larger text**, from 9 to 24 (13 is the default), and watch the preview. A bigger size fits fewer columns on the screen. It applies straight away to open terminals. **Reset** goes back to 13. |

These settings are kept on this device. For the rest of the settings, see Settings.

### Upgrade commands

An upgrade prints a lot of output, more than the default history keeps. When your history is shorter than 10,000 lines and you type an upgrade command, such as `apt upgrade`, `apt-get dist-upgrade`, `apt full-upgrade` or `pveupgrade`, VisorMon holds back the Enter key and asks **Run the upgrade?** first. The check happens on your phone.

- **10,000 lines, just for this** (the default choice) keeps more history in this shell only.
- **10,000 lines, and make it my setting** also changes **Terminal scrollback**, so VisorMon stops asking.
- **Keep 1,000 lines** (or whatever your setting is) leaves the history as it is.

Tap **Run it** to go ahead, or **Cancel** to leave the command typed at the prompt without running it.

### Leaving, reloading and reconnecting

- **Leaving.** Going back from a connected shell asks **Close the terminal? Anything still running in it stops.** Inside tmux it says **Your tmux session keeps running on the node. Open the shell again to return to it.** On iPhone, the swipe back from the edge of the screen is turned off while the shell is connected; use the back button.
- **Reloading. Reload console** starts a new shell. When one is connected, VisorMon asks **Reload the terminal?** first.
- **A dropped connection.** A red strip above the terminal says why the shell stopped, and what it printed stays readable. Tap **Reconnect** to connect again on the same screen; a `--- reconnected ---` line marks where the new connection starts. Without tmux, this is a new shell.

## 4.7 Services

**Services** lists the node's system services, the same list as **System** in the Proxmox web interface. Each shows a green dot when it is running and red when it is not, its description, and its state, for example **running · enabled**.

With `Sys.Modify` on the node, each service has a menu (⋮): **Start** for a service that is stopped, or **Restart**, **Reload** and **Stop** for one that is running. The list refreshes a moment after the change.

- **Start** and **Reload** go straight away.
- **Stop** and **Restart** ask first, for example **Restart chrony? Restart chrony on pve?**
- For `pveproxy` and `pvedaemon`, which serve the Proxmox API that VisorMon talks to, the question warns you. Restarting one drops VisorMon's connection to the node for a moment. Stopping one drops it until the service is started again from a shell on the node.

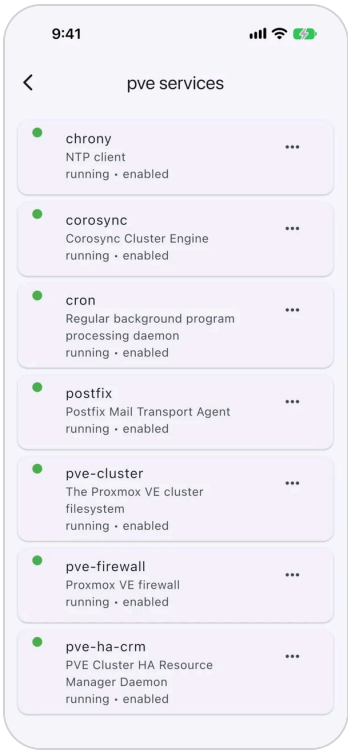
Without `Sys.Modify` the screen says **View only. This account cannot change services.**

## 4.8 System log

**System log** shows the node's journal, newest at the bottom, and opens scrolled to the end. It needs `Sys.Syslog` on the node.

- **Show**: **Latest 500** lines, or everything from the **Last hour**, **Last day** or **Last week**.
- **Service**: type a service name, such as `pveproxy`, and press the keyboard's search key to see only its lines. Clear the field and press search again to see everything.
- **Kernel messages only**: only the kernel's own messages.

Touch and hold the text to select and copy it. Pull down to load the latest lines. If nothing matches, the screen says **No log lines match.**



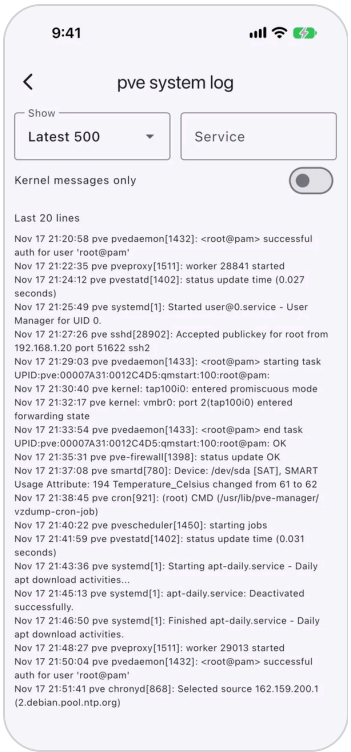

Node › Services.

## 4.9 Updates

**Updates** shows the packages the node can upgrade, each with its old and new version, for example **8.2.4 → 8.2.7**. The card at the top says **Up to date** or how many updates are available. Proxmox requires `Sys.Modify` on the node even to list them.

**Refresh** — The list is what the node already knows about. Refresh asks the node to check its repositories for newer versions, like `apt update`, and reloads the list when it finishes. If the check takes longer than about a minute and a half, VisorMon says **The refresh is still running. Pull down to reload the list later.**

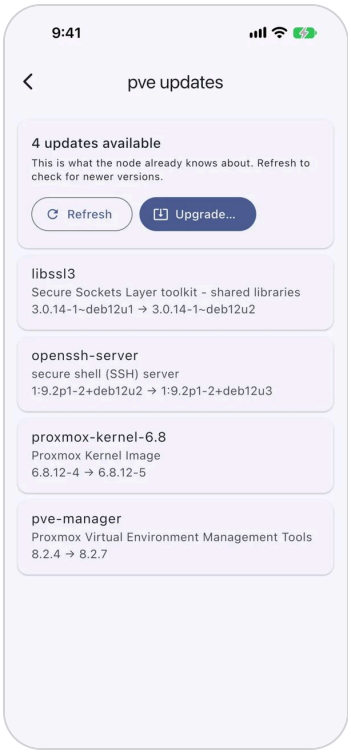
**Upgrade…** — Appears when updates are available and your account has `Sys.Console` on the node. It asks **Upgrade packages? This opens a shell on *name* and starts the upgrade there, the same way the web interface does. You confirm it inside the shell. The node may need a reboot afterwards.** Tap **Open shell**, then answer the upgrade's own question in the shell.




Node › System log.

If your terminal history is shorter than 10,000 lines, the question also offers to keep more of the upgrade's output, with the same three choices as typing an upgrade command (see Upgrade commands). Proxmox runs the upgrade shell only for root@pam signed in with a password.

Where the updates come from is set under **System › Repositories** (see System).




Node › Updates.

## 4.10 Network

**Network** lists the node's network interfaces. Each shows a green dot when it is active (gray when not), its type (**Bridge**, **Bond**, **Network device**, **VLAN** and so on), its address and gateway, its ports, its comment, and **autostart** when it starts at boot.

Changing the network needs `Sys.Modify` on the node. Without it the screen says **View only. This account cannot change the network.** As in the Proxmox web interface, a change happens in two steps: you **save** it, which stores it on the node, then you **apply** it, which reloads the node's network.

### Adding and changing interfaces

- **Add:** tap **Add interface** and choose **Linux bridge**, **Linux bond** or **Linux VLAN**. VisorMon suggests a free name; bridges are named `vmbr0`, `vmbr1` and so on, and bonds `bond0`, `bond1`.
- **Change:** tap a bridge, bond, network device or VLAN. Open vSwitch and SDN interfaces are listed but changed in the Proxmox web interface.

The form has an **IPv4 address** with its prefix (for example `192.168.1.10/24`), a **Gateway**, optional IPv6 address, IPv6 gateway and **MTU**, a **Comment** and **Start at boot**. A bridge adds **Bridge ports** and **VLAN aware**; a bond adds **Slaves**, **Mode** and, depending on the mode, **Primary** or **Hash policy**; a VLAN adds **Runs on** and **VLAN ID**. Tap the interface names under a field to fill it in. Tap **Save**: **Saved. Apply the changes to make them take effect.**

To delete a bridge, bond or VLAN, open it and tap the bin icon. You type its name to confirm; guests using it lose their network once the change is applied. Network devices cannot be deleted.




Node › Network.

### Applying or discarding changes

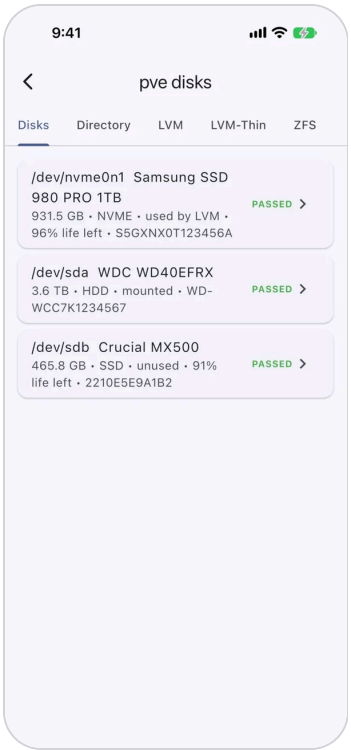
When changes are saved but not applied, an orange card says **Changes are saved but not applied**. Tap **Show changes** to see exactly what will change, then:

- **Apply** asks **Apply network changes? Applying reloads the network on the node now. VisorMon cannot undo it: if the change cuts the connection, it has to be fixed at the node itself or from another network.** VisorMon then waits and reports **Network changes applied.**, the reason it failed, or **Still applying, or the connection was interrupted.**
- **Revert** asks **Discard network changes?** and throws away what has not been applied.

> **Protecting your own connection.** If your server address in VisorMon is the node's IPv4 address, VisorMon knows which interfaces carry your connection, including the ports, bonds and VLAN parents beneath them. Saving a change to one of those makes you type `change`, and applying any network changes makes you type `apply`, with a warning that names the interface. With a host name as your server address, VisorMon cannot tell which interface you use, so check carefully before you apply.

## 4.11 Disks

**Disks** has five tabs: **Disks**, **Directory**, **LVM**, **LVM-Thin** and **ZFS**.



Node › Disks.

### The disks and their health

The **Disks** tab lists each physical disk: its device path and model, size, type, whether it is **unused**, **mounted** or **used by** something (such as LVM or ZFS), how much life an SSD has left, and its serial number. On the right is its SMART health: **PASSED** or **OK**, **FAILED** in red, or anything else in orange.

Tap a disk to see its SMART attributes, with failing ones in red, or the disk's own SMART report. Proxmox checks SMART data against `Sys.Audit` on `/`.

### Erasing a disk

An unused disk's page has an **Erase** section, if your account has `Sys.Modify` on `/`:

- **Initialise GPT** writes a new, empty GPT partition table to the disk.
- **Wipe disk** erases everything on the disk.

Both ask you to type the disk's device path, such as `/dev/sdb`, to confirm, and say **This cannot be undone.** or **Anything on it is lost.** A disk that is in use shows what uses it instead: remove that first.

### Creating and removing storage

The other four tabs list the node's directory, LVM, LVM-thin and ZFS storage. With `Sys.Modify` on `/`, each tab has a button such as **Create ZFS**:

1. Enter a **Name**, the storage ID, for example `tank`.
2. Choose an unused **Disk**. For a directory, choose its **Filesystem** (`ext4` or `xfs`). For ZFS, tick the disks and choose the **RAID level**, **Compression** and **ashift** (12 suits most disks).
3. Leave **Add as storage** on to make it available for guests and backups.
4. Tap the create button, then type `erase` to confirm that everything on the chosen disks is erased.

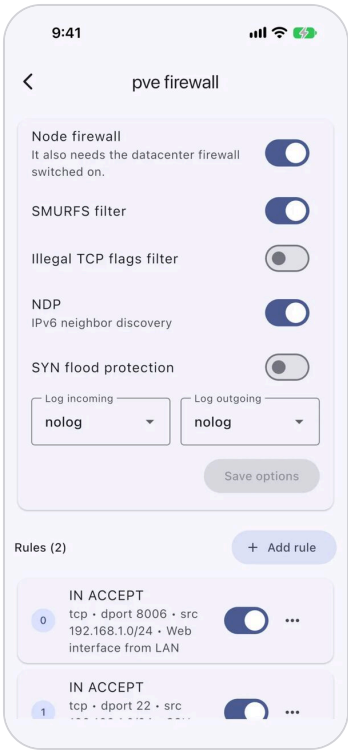
A ZFS pool needs enough disks for its RAID level: exactly one for `single`, at least two for `mirror`, three for `raidz`, four for `raidz2`, five for `raidz3`, and an even number of at least four for `raid10`. Only unused disks are offered; if there are none, the form says **No unused disks**.

To remove a storage, tap its bin icon. Choose whether to **Remove its storage entry** (on by default) and **Wipe the disks** (off by default), type the storage's name, and tap **Remove**. Anything stored on it is lost.

Adding or removing the storage entry also needs `Datastore.Allocate` on `/storage` in Proxmox. To see what is stored on a storage, use the **Storage** tab (see Storage).

## 4.12 Firewall

**Firewall** shows the node's own firewall, as under the node's Firewall in the Proxmox web interface. It filters traffic to the node itself; each guest's firewall is on the guest's page (see Working with a guest). Changing it needs `Sys.Modify` on the node.



Node › Firewall.

### Options

Turn on **Node firewall** (it also needs the datacenter firewall switched on), and choose the **SMURFS filter**, **Illegal TCP flags filter**, **NDP** (IPv6 neighbor discovery) and **SYN flood protection**, and the **Log incoming** and **Log outgoing** levels. Nothing changes until you tap **Save options**.

Turning the firewall on asks first: **Turn the node firewall on? Traffic to *name* is filtered by its rules from now on. Make sure a rule allows the Proxmox web interface (port 8006) from where you connect, or you can lock yourself out.**

### Rules

Rules are listed in order, with their position number; Proxmox checks them from the top.

- **Add:** tap **Add rule**. Choose the **Direction** (`in`, `out` or `forward`), **Action** (`ACCEPT`, `DROP` or `REJECT`) and **Protocol**, and fill in what you need of **Source**, **Destination**, ports, **Interface**, **Macro** (such as `SSH`) and **Comment**. Ports need a protocol. The rule is added at the end of the list.
- An enabled incoming rule that drops or rejects traffic asks **Block incoming traffic?** first, because it could cut off VisorMon's own connection.
- **Turn a rule on or off** with its switch. This takes effect straight away.
- **Move or delete** a rule from its menu (⋮): **Move up**, **Move down** or **Delete**. Delete asks **Delete rule?** first.

## 4.13 Tasks

**Tasks** shows the cluster's recent tasks that ran on this node, newest first: what each one was, who started it, how long ago, and its result, **running** in orange, **OK** in green, or the error in red. Tap **Refresh** or pull down to update it.

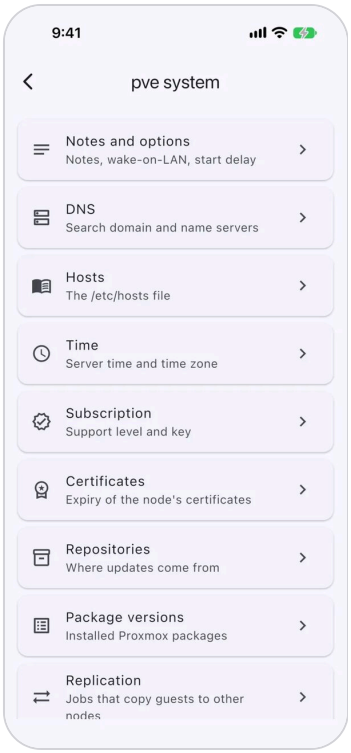
Tap a task to read its log. A running task can be stopped there with **Abort**. The log and the cluster-wide task list are described in Search, tasks and alerts.



Node › Tasks.

## 4.14 System

**System** holds the rest of the node's settings and information. An entry your account may not open is grayed out with **No audit permission on this node**. Inside, a screen your account can read but not change says so with a **View only** line.




Node › System.

| Entry | What you can do | Needs |
|---|---|---|
| **Notes and options** | Edit the node's notes, its Wake-on-LAN address and the start delay | `Sys.Audit` on `/` to open, `Sys.Modify` on `/` to save |
| **DNS** | Set the search domain and up to three DNS servers | `Sys.Audit` on the node, `Sys.Modify` to save |
| **Hosts** | Edit the node's `/etc/hosts` file | `Sys.Audit` on `/` to open, `Sys.Modify` on the node to save |
| **Time** | See the node's clock and set its time zone | `Sys.Audit` on the node, `Sys.Modify` to save |
| **Subscription** | See the node's Proxmox subscription, check it, or enter a key | Anyone can open it; `Sys.Modify` on the node to check or set a key |
| **Certificates** | See the node's certificates and when they expire | Anyone can open it |
| **Repositories** | Turn package repositories on or off, or add a standard one | `Sys.Audit` on the node, `Sys.Modify` to change |
| **Package versions** | See the installed Proxmox packages and the running kernel | `Sys.Audit` on the node |
| **Replication** | See replication jobs and run one now | Anyone can open it; `VM.Replicate` on the guest to run its job |

| Entry | What you can do | Needs |
|---|---|---|
| **Hardware** | See the node's PCI and USB devices | `Sys.Audit` or `Sys.Modify` on `/`; USB devices need `Sys.Modify` on `/` |
| **System report** | Read and copy the report Proxmox support asks for | `Sys.Audit` on the node |

### Notes and options

- **Notes:** free text about the node, shown on its page.
- **Wake-on-LAN:** the node's **MAC address** (like `AA:BB:CC:DD:EE:FF`) and, if your network needs it, a **Broadcast address**. VisorMon uses these to wake the node when it is off; the node you are connected to sends the packet.
- **Start at boot: Delay before starting guests (seconds)**, from 0 to 300. Guests set to start at boot wait this long after the node starts.

Tap **Save**: **Options saved.**

### DNS, hosts and time

- **DNS:** a **Search domain** is required; DNS servers must be IP addresses. Saving does not ask first.
- **Hosts:** edit the file as text. Saving asks **Save hosts file? This rewrites /etc/hosts on *name*. A wrong entry for the node itself can break name lookups and cluster traffic.**
- **Time:** shows **Time on the node** and its zone. Pick a **Time zone** from the list or type one, such as `Europe/Berlin`, and tap **Save**. The clock itself is not set here.

### Subscription and certificates

**Subscription** shows the node's Proxmox subscription status and details: product, level, key, server ID, sockets, and the registration, due and last-checked dates. **Check now** asks Proxmox to check the subscription again. **Set key** enters a new subscription key; then tap **Check now** to validate it. This is your Proxmox subscription, not VisorMon Pro (see VisorMon Pro).

**Certificates** lists each certificate file with its subject, issuer, validity dates, key, names and fingerprint. The label on the right says how many days are left: green, orange when fewer than 30 days remain, and red once it has expired. Certificates are uploaded and renewed in the Proxmox web interface.

### Repositories and package versions

**Repositories** lists the node's package repositories, each with a switch, and shows any warnings Proxmox gives about them. Turning a repository on or off asks first, for example **Disable repository?** Under **Add a standard repository**, tap **Add** next to one the node does not have yet; it is added turned on.

**Package versions** lists the installed Proxmox packages and their versions. Type in **Filter packages** to find one.

### Replication, hardware and the system report

- **Replication** lists the node's replication jobs: guest and target node, **ok**, **failing** or **disabled**, the schedule, the last and next run, and any errors. **Run now** starts a job straight away. Jobs are created and changed in the Proxmox web interface.
- **Hardware** lists PCI devices, with their IOMMU groups, and USB devices. An account without `Sys.Modify` on `/` sees **This account cannot list USB devices.**
- **System report** shows the node's full report. Tap **Copy all** to copy it, for example to send to Proxmox support.

CHAPTER 5

# Guests

The guest overview, the guests dashboard and the Guests tab: find guests, see how they are doing, act on them.

## 5.1 Three places to work with guests

Proxmox calls virtual machines (VMs) and containers *guests*. VisorMon shows them in three places, each for a different job:

| Where | Best for | How to get there |
|---|---|---|
| **Guest overview** | A glance at the few guests that matter most to you | The **Dashboard** tab, in the **Guests** section below **Nodes** |
| **Guests dashboard** | How every guest is doing, all at once | **View all** in the guest overview, or the gauge button on the Guests tab |
| **Guests tab** | Finding a guest and doing something with it | The **Guests** tab at the bottom of the screen |

Tap a guest in any of the three to open its own page, with its full details, console, snapshots, backups and settings. That page has its own chapter: see Working with a guest.

## 5.2 Reading a guest card

The guest overview and the guests dashboard show each guest as a card. Every card is laid out the same way, so once you can read one you can read them all.

**Name** — Shown in gray when the guest is not running.

**critical** — A red flag that appears when a reading reaches 90% (see Critical guests below).

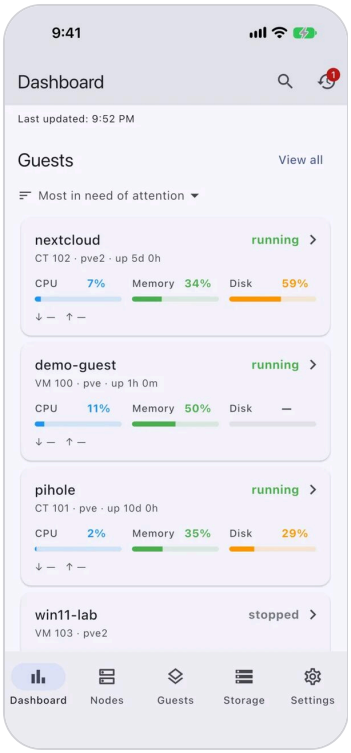
**Pin** — Appears when you have pinned the guest to the dashboard (see Pinning guests).

**State** — **running**, **stopped**, **paused** or **unknown**, on the right. Unknown usually means the guest's node is down. The arrow beside the state means the card opens the guest's page.

**Details** — VM or CT (container) and its ID, the node it runs on, and how long it has been up, for example **VM 100 · pve · up 4d 3h**.

**CPU, Memory, Disk** — A figure and a bar for each reading.

**Network** — How fast data is moving: ↓ received and ↑ sent, per second. Only running guests show it.




The guest overview on the dashboard.

| Reading | What it measures | Bar |
|---|---|---|
| **CPU** | How much of the guest's assigned CPU cores it is using | Blue |
| **Memory** | Memory in use, as a share of the memory assigned to the guest | Green |
| **Disk** | For a container, space used on its root disk as a share of the disk's size. For a VM, a dash: Proxmox does not report a VM's disk use in its cluster overview. | Orange |

The figures refresh on the schedule set in **Settings › Data refresh** (see Settings). Pull down on a list to refresh it straight away.

## Critical guests

A running guest is **critical** when its CPU, memory or disk reaches 90% of its limit. That reading's figure and bar turn red, the card gets a red **critical** flag, and the guest counts toward the critical totals on the dashboard and the guests dashboard. The 90% level is fixed.

Red is kept for trouble: a critical reading, the **Stop** action, and a high-availability error. Everything else uses the app's usual colors.

## When a figure shows a dash

- **The guest is not running.** A stopped guest has no readings.

- **A VM's disk.** Open the VM to see its disk. If the QEMU guest agent is running in the VM, its page shows how much of the disk is used.
- **Network, just after you open a screen.** Proxmox only reports how many bytes a guest has sent and received since it started. VisorMon works out a speed from two readings at least 4 seconds apart, so the speed appears after the next refresh.

### With a screen reader

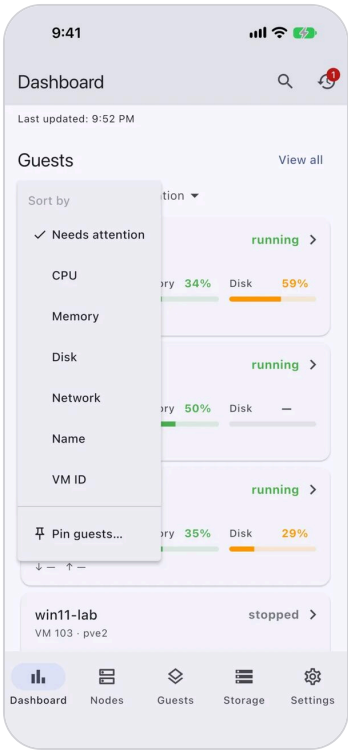
VoiceOver and TalkBack read each card as one item: the guest's name, kind, ID, node, state, readings, and whether it is critical or pinned. The cards also work at the largest text sizes: figures shrink and long names are shortened rather than running off the card.

## 5.3 The guest overview on the dashboard

The **Dashboard** tab has a **Guests** section below **Nodes**. It shows five guests, chosen the way you decide, so the ones you care about are always in view. Templates are never included.

### Choosing which guests come first

1. On the Dashboard, tap the line under the **Guests** heading. Until you change it, it reads **Most in need of attention**.
2. Under **Sort by**, tap an order. The dashboard changes straight away, and VisorMon remembers your choice.




The menu under the Guests heading.

| Order | The line reads | Shown first |
|---|---|---|
| **Needs attention** (default) | Most in need of attention | Critical guests, then guests with a reading from 80% to 89%, then other running guests by their highest reading. Guests that are off come last. |
| **CPU** | Highest CPU | The guests using the most CPU |
| **Memory** | Highest memory | The guests using the most of their memory |
| **Disk** | Fullest disk | The fullest container disks. VMs have no disk figure, so they come after the containers. |
| **Network** | Busiest network | The most data moving in and out together |
| **Name** | By name | A to Z |
| **VM ID** | By VM ID | The lowest ID |

Guests without a figure for the chosen reading, such as guests that are off, come after the rest. Ties are broken by name.

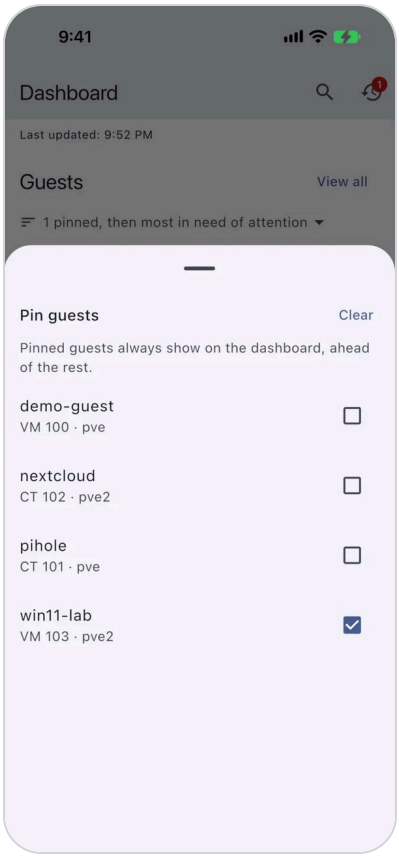
## Pinning guests

Pin the guests you always want to see. Pinned guests come first, each marked with a pin, and the rest of the five places are filled in the order you chose. The line under the heading then says so, for example **1 pinned, then most in need of attention**.

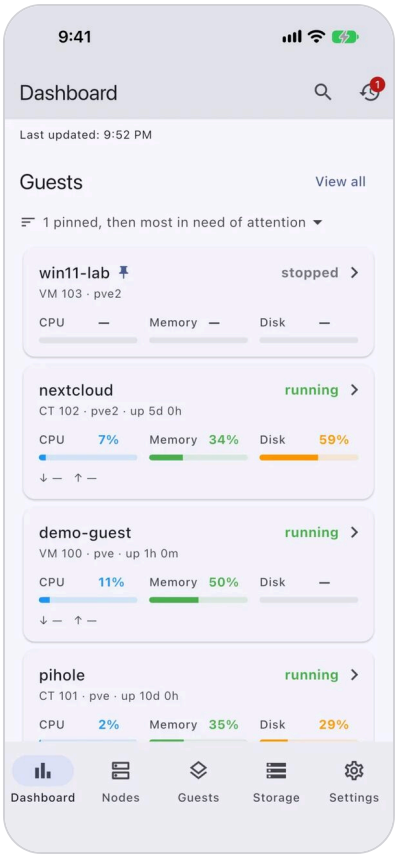
**To pin from the dashboard:**

1. Tap the line under the **Guests** heading, then choose **Pin guests…** from the menu.
2. Tick the guests you want. Each tick is saved as you make it. With seven or more guests there is a search box at the top.
3. Swipe the sheet down, or tap outside it, when you are done.

**To pin from the Guests tab:** tap a guest's ⋮ menu and choose **Pin to dashboard**. VisorMon confirms with a message such as **Pinned win11-lab to the dashboard**.




Choosing guests to pin. Each tick is saved as you make it.




The pinned guest comes first, even though it is stopped.

**Good to know:**

- Pinned guests are shown in your chosen order among themselves.
- If you pin more than five, the dashboard shows every pinned guest and no others.
- The pin also shows on the guest's card on the guests dashboard and the Guests tab.
- Pins are kept on this device for each server, by its address, because a VM ID only identifies a guest within one cluster. If you change a server's address in VisorMon, pin its guests again.
- A pinned guest that has since been deleted takes up no place.
- Templates cannot be pinned.
- To unpin, untick the guest in **Pin guests…**, tap **Clear** there to unpin every guest on this server, or choose **Unpin from dashboard** on the Guests tab.
- Pinning only changes what this device shows. Nothing is sent to your server.

## The critical count and View all

- When any guest is critical, a red count such as **2 critical** appears beside the heading. Tap it to open the guests dashboard showing only the critical guests. The count includes every guest, not just the five shown, so a critical guest is one tap away even when your pins or your order leave it out.
- **View all** opens the guests dashboard with every guest.

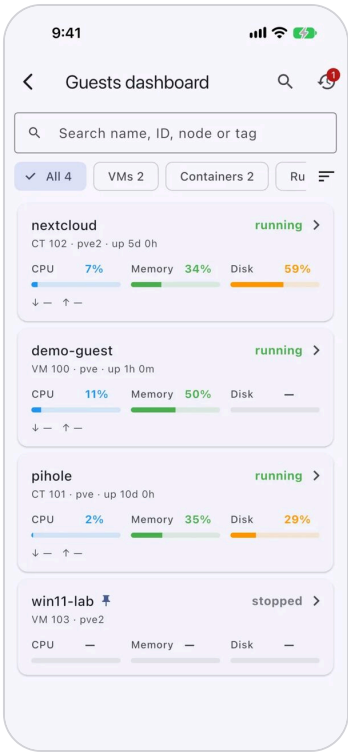
To hide the overview or move it, use **Settings › Arrange panes**, where it is listed as **Guest overview** (see Settings).

## 5.4 The guests dashboard

The guests dashboard shows every VM and container as a card, so you can see how they are all doing at once.

To open it:

- tap **View all** in the dashboard's guest overview,
- tap the gauge button (**Guests dashboard**) at the top of the Guests tab,
- tap the red critical count on the dashboard, which opens it showing only the critical guests, or
- tap a red **critical** flag on the Guests tab, which does the same.



The guests dashboard.

### Searching and filtering

The search box looks at each guest's name, VM ID, node and tags. Below it, a row of chips narrows the list, each with a count:

| Chip | Shows |
|---|---|
| **All** | Every VM and container |
| **VMs** / **Containers** | One kind only |
| **Running** | Guests that are on |
| **Critical** | Critical guests only. The chip turns red when there are any. |

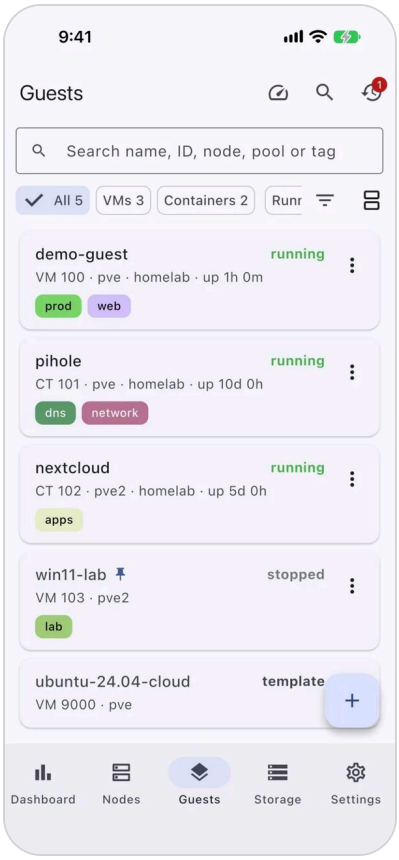
### Sorting and layout

The sort button at the end of the chip row offers the same seven orders as the dashboard (see Choosing which guests come first). This screen starts at **Needs attention** each time you open it; the order you choose for the dashboard's overview is remembered separately.

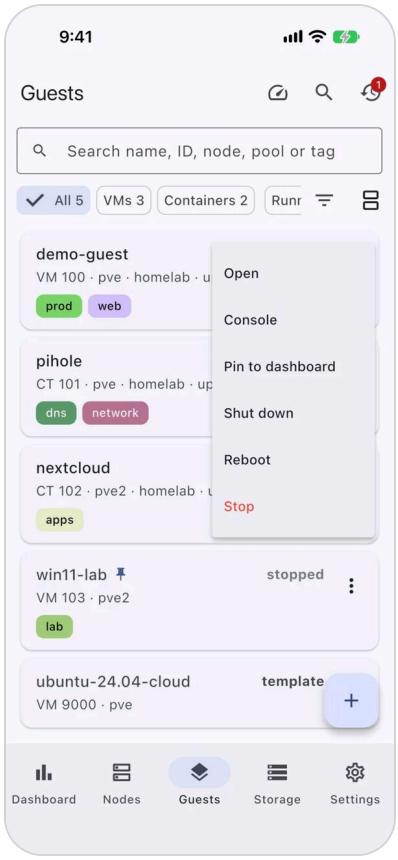
A phone shows one card per row; a tablet or a phone on its side fits two or three. On a short screen, such as a phone on its side, the search box and chips scroll away with the cards to leave room for them. Templates are not listed here, because they do not run; find them on the Guests tab.

# 5.5 The Guests tab

The Guests tab is for finding a guest and doing something with it. Its cards show what each guest is and where, rather than its readings, which are on the guests dashboard.




The Guests tab, with tags and a pinned guest.

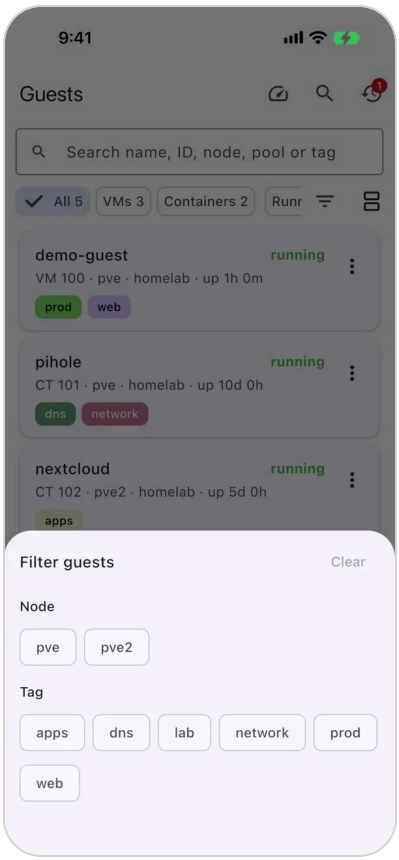



A running guest's menu.

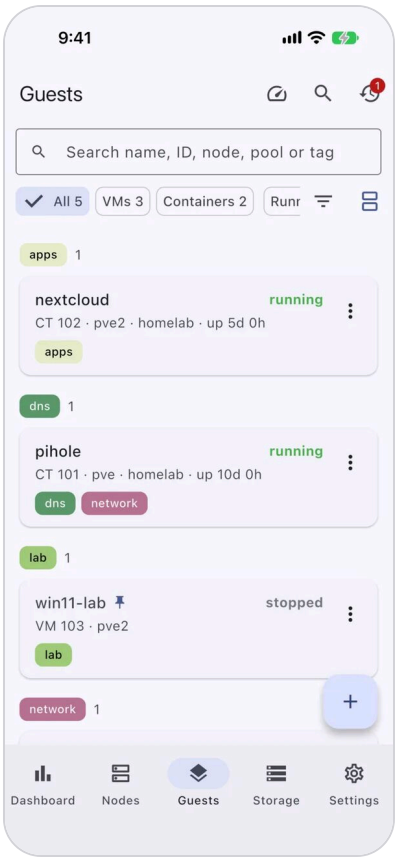
## Finding a guest

- **Search** looks at the name, VM ID, node, pool and tags. Tap × to clear it.
- **All, VMs, Containers** show every guest or one kind. Only one is on at a time.
- **Running, Stopped, Templates** can be combined, for example Running and Stopped together. Tap one again to turn it off. A template does not count as stopped.
- **Filter by node or tag** (the filter button) opens a sheet of the cluster's nodes and tags. A guest shows if it is on any node you choose and has any tag you choose. The button shows how many you have chosen, and **Clear** removes them all.

## Grouping




Filtering by node and tag.




Grouped by tag. Each heading is drawn in its tag's color.

The **Group by** button arranges the list under headings, each with a count: **No grouping**, **By node**, **By tag** or **By pool**. VisorMon remembers your choice.

- **By tag:** a guest with two tags appears under both. Untagged guests are listed last, under **No tag**.
- **By pool:** guests without a pool are listed last, under **No pool**.

## What a card shows

- The **name**, the red **critical** flag when it applies (tap it to see every critical guest), the **pin**, and the **state**. Templates show **template**.
- The **details line** includes the guest's pool, for example **VM 100 · pve · homelab · up 4d 3h**.
- Its **tags**, in the same colors the Proxmox web interface gives them.
- A **high-availability (HA) chip** when the cluster's HA manager looks after the guest:

| Chip | Meaning |
|---|---|
| HA | HA keeps the guest in the state it was asked for (started or stopped). |
| HA off | HA is turned off for this guest. |

| Chip | Meaning |
|---|---|
| HA · error | In red: HA reports a problem (error, fence or recovery). Check the guest in Proxmox. |
| HA · migrate | In orange: HA is in the middle of something, such as migrating or relocating the guest. |

## Acting on a guest from the list

Tap a guest's ⋮ button, or touch and hold its card, to open its menu. The menu offers what makes sense for the guest's state:

| When the guest is | The menu offers |
|---|---|
| Running | Open, Console, Pin to dashboard, Shut down, Reboot, Stop |
| Stopped | Open, Pin to dashboard, Start |
| Paused | Open, Pin to dashboard, Resume, Stop |
| A template | Open |

Anything that takes a guest down asks first, in the same words as the guest's own page:

| Action | VisorMon asks |
|---|---|
| **Shut down** | Shut down *name*? The guest is asked to power off. |
| **Reboot** | Reboot *name*? |
| **Stop** | Stop *name*? It powers off immediately. |

**Start** and **Resume** go straight away. While an action is being sent, the card shows a spinner and VisorMon shows a message such as **Starting win11-lab**; the list refreshes a moment later with the new state. If your account may not do something, it is grayed out and the menu says **Greyed out: your account can't do this.**

| To | Proxmox privilege |
|---|---|
| See guests and their readings | `VM.Audit` on the guests (included in the PVEAuditor role) |
| Start, shut down, reboot, stop or resume a guest | `VM.PowerMgmt` on that guest |
| Open a guest's console | `VM.Console` on that guest |
| Create a VM or container | `VM.Allocate` |
| Change the dashboard's order or pins | None. These are kept on your device. |

### Creating a guest

If your account may create guests, the Guests tab has a **+** button. It asks whether to **Create VM** or **Create container**; with the VMs or Containers chip on, it goes straight to that form. The forms are described in Working with a guest.

## 5.6 Questions about guests

**Why does a VM's disk show a dash?**

Proxmox reports disk use for containers, but not for VMs, in its cluster overview. Open the VM to see its disk. If the QEMU guest agent is installed and running in the VM, its page shows how much of the disk is used.

**Why is the network speed a dash?**

VisorMon needs two readings to work out a speed. It appears after the next refresh. A guest that is not running has no speed.

**A guest is critical, but it is not among the five on the dashboard.**

Your pins or your order put other guests first. The red critical count beside the heading still includes it; tap the count to see every critical guest. Choose **Needs attention** to always show critical guests first.

**Where are my templates?**

Templates do not run, so they are left out of the guest overview and the guests dashboard. They are on the Guests tab; tap the **Templates** chip to show only templates.

**My pins have gone.**

Pins are kept for each server by its address. If you changed the server's address in VisorMon, or reinstalled the app, pin the guests again.

**How many guests does the dashboard show?**

Five, or every pinned guest if you have pinned more than five. **View all** shows the rest.

CHAPTER 6

# Working with a guest

A guest's page: power, console, snapshots, backups, hardware, migrate, clone and creating guests.

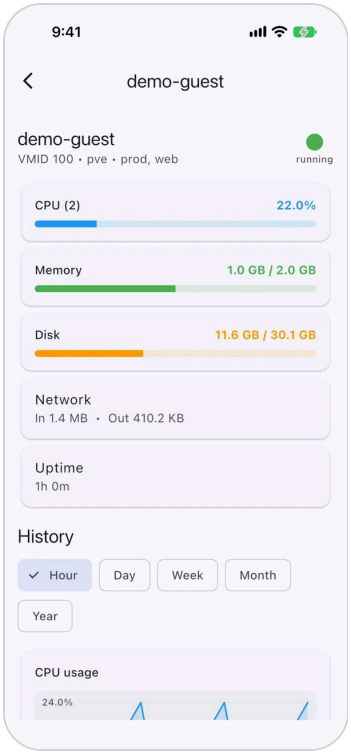
## 6.1 A guest's page

Every virtual machine (VM) and container has its own page. It shows how the guest is doing right now and over time, and it is where you start, stop and change it. Open a guest by tapping it on the dashboard, the guests dashboard, the Guests tab or in cluster search; see Guests and Search, tasks and alerts.

From top to bottom, the page has:

- **The heading.** The guest's name, then its VMID (the guest's ID number), the node it runs on, **template** if it is one, and its tags.
- **Its state**, on the right: **running**, **paused**, **stopped** with a red dot, or **template**. While VisorMon follows a reboot it shows **rebooting** and then **starting**.
- **Readings**: CPU, memory, disk, network and uptime.
- **History**: charts over the last hour, day, week, month or year.
- **Power** and **Manage**: the buttons for everything you can do with the guest.
- **Notes** and **Tasks**: the guest's notes from Proxmox and its recent tasks.

The page loads when you open it and loads again after a power action or when you come back from one of its sections. It does not refresh on its own; to see fresh readings, go back and open the guest again. If something could not be loaded, a box explains why and offers **Retry**.




The top of a guest's page.

### Readings

| | |
|---|---|
| **CPU** | How busy the guest's CPU is, as a share of the cores it has. The number of cores is in brackets, for example **CPU (2)**. |
| **Memory** | Memory in use and the memory assigned, for example **1.0 GB / 2.0 GB**. |
| **Disk** | How full the guest's disk is, when that can be known. See Disk usage below. |
| **Network** | Data received (**In**) and sent (**Out**) since the guest last started. |
| **Uptime** | How long the guest has been running. A dash means it is not running. |

### Disk usage

Proxmox measures how full a container's root disk is, but not a VM's disk. For a VM, VisorMon asks the QEMU guest agent, a small program that runs inside the VM, how much space its file systems use. It adds up the file systems the agent reports, so the total is the space inside the VM, which can differ from the size of the virtual disk.

When usage cannot be read, the **Disk** row shows the size that is allocated and one of these lines:

| The Disk row says | Why |
|---|---|
| **Usage is shown while the VM is running with the guest agent.** | The VM is not running. |
| **Turn on the QEMU guest agent to see usage.** | The agent is switched off in the VM's options. Turn it on in Options, and install the agent inside the VM if it is not there yet. |
| **The guest agent is not answering, so usage cannot be read.** | The agent is switched on, but it did not answer. It may not be installed or running in the VM, or your account may not be allowed to ask it (see Permissions). |
| **Usage needs the QEMU guest agent running in the VM.** | The agent did not answer, and Proxmox did not say it is switched on. Usually it is switched off: turn it on in Options and install it inside the VM. |
| **Usage is shown while the container is running.** | The container is not running. |
| **Proxmox did not report disk usage.** | The container is running, but Proxmox sent no figure. |

### History

Choose **Hour**, **Day**, **Week**, **Month** or **Year** above the charts. Each chart's legend shows its latest value.

- **CPU usage**.
- **Memory usage**: memory used, and the total.
- **Disk usage**: space used, and the disk's size. Containers only: Proxmox does not record disk usage history for VMs, and the page says so under the charts.
- **Network traffic**: data received and sent per second.
- **Disk I/O**: data read from and written to disk per second.

If your account may not read a guest's history, the charts are replaced by **History is not permitted for this account.**

### The Manage buttons

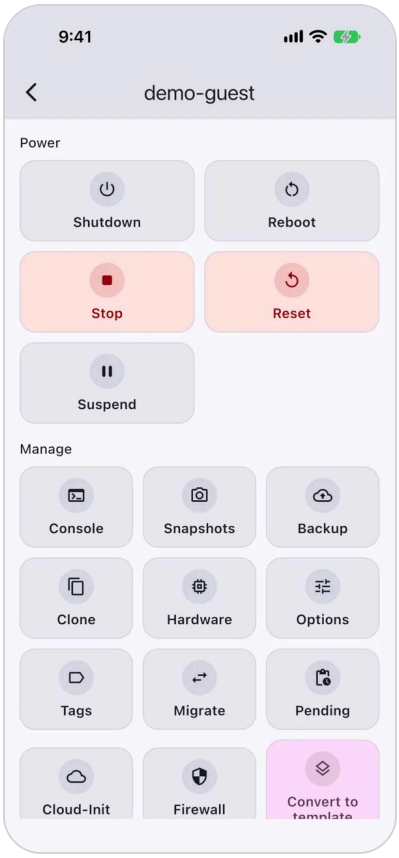
Below the history, **Power** holds the power actions (see Power actions) and **Manage** opens everything else:

| Button | Opens | Shown |
|---|---|---|
| **Console** | The console | Always |
| **Snapshots** | Snapshots | Always |
| **Backup** | Backup and restore | Always |
| **Clone** | Clone | When your account may clone at least one guest |
| **Hardware** | Hardware | Always |
| **Options** | Options | Always |
| **Tags** | Tags | Always |
| **Migrate** | Migrate | When your account may migrate at least one guest |
| **Pending** | Pending changes | Always |
| **Cloud-Init** | Cloud-Init | VMs only |
| **Firewall** | Firewall | Always |
| **Convert to template** | Convert to template | Not on templates, and only when your account may create guests |

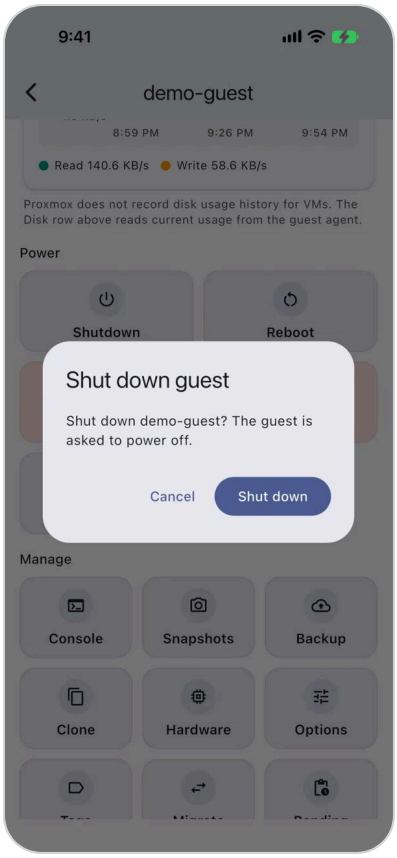
A button your account may not use on this guest is grayed out, with the reason under its label, for example **No console permission on this VM**.

## 6.2 Power actions

The **Power** buttons change with the guest's state. A stopped guest has **Start**. A running guest has **Shutdown**, **Reboot**, **Stop**, **Suspend** and, for a VM, **Reset**. A paused guest has **Resume** and **Stop**. Templates cannot run, so they have no **Power** section.




Power and Manage on a running VM. Stop and Reset are red.




Every action except Start and Resume asks first.

| Button | What it does | Asks first |
|---|---|---|
| **Start** | Starts a stopped guest. | No |
| **Shutdown** | Asks the guest to shut down cleanly. For a VM this is like pressing a computer's power button: the operating system has to respond. | **Shut down guest**: **Shut down *name*? The guest is asked to power off.** |
| **Reboot** | Shuts the guest down and starts it again. Proxmox applies any pending changes on the way. VisorMon follows the reboot until the guest is back (see below). | **Reboot guest**: **Reboot *name*?** |
| **Stop** | Powers the guest off at once, like pulling the plug. Use it when a shutdown does not work. | **Stop guest**: **Stop *name*? It powers off immediately.** |
| **Reset** | VMs only. Restarts the VM at once, like pressing a computer's reset button. | **Reset guest**: **Hard-reset *name*? Unsaved work inside the guest is lost.** |
| **Suspend** | Pauses the guest. It keeps its memory and carries on where it stopped when you tap **Resume**. Proxmox marks suspending a container as experimental. | **Suspend guest**: **Suspend *name*? It stays in RAM until you resume it.** |
| **Resume** | Carries on a paused guest. | No |

Each dialog has **Cancel** and a button named after the action. After the action, VisorMon waits a moment and reloads the page. If Proxmox refuses, the reason appears at the bottom of the screen and in a box under the buttons. While an action is running, the power buttons are grayed out.

All power actions need the `VM.PowerMgmt` privilege on the guest. Without it the buttons are grayed out with **No power permission on this VM** (or **container**).

### Following a reboot

After **Reboot**, an orange box shows **Rebooting VM *name*…** (or **container**), then **Starting *name* back up…** once the guest has gone down. When it is running again, the box turns green and says ***name* has rebooted and is back up.**

VisorMon waits up to 5 minutes for a VM and 90 seconds for a container. If the reboot takes longer, it stops waiting and shows a message that starts **Reboot did not finish in time.** and says whether the guest is still running or still down. The reboot itself carries on in Proxmox.

## 6.3 The console

Tap **Console** to open the guest's console full screen, titled with the guest's name, for example **web01 console**. A VM opens its screen, the same one the Proxmox web interface shows. A container opens a text terminal. The phone's screen stays on while a console is connected.

The console needs the `VM.Console` privilege on the guest.

### A VM's console

The VM's screen is scaled to fit the phone. A row of keys runs along the bottom; swipe it sideways to see them all.

| | |
|---|---|
| ⌨ | Shows or hides the phone's keyboard. What you type is sent to the VM as key presses. |
| **Esc, Tab, Del, Home, End, PgUp, PgDn, F1 to F12** | The keys a phone keyboard lacks. |
| **Ctrl, Alt, Win** | Held for the next key you press, then released. The key is highlighted while it is held. |
| **Ctrl-Alt-Del** | Sends Ctrl+Alt+Del, for example to reach a Windows sign-in screen. |
| ↑ ↓ ← → | Arrow keys. Hold one to repeat it. |

Touch the screen to use the VM's mouse: a tap is a left click, a two-finger tap a right click, and dragging with one finger drags with the left button held. A keyboard connected to the phone also works, including shortcuts such as Ctrl+C.

The refresh button at the top (**Reload console**) connects again. If the connection drops, a red strip at the top says why, with **Reconnect**.

### A container's console

A container's console is a terminal attached to the container. It shows the container's own login prompt, so sign in with an account inside the container, for example `root` and the password set when the container was created.

Proxmox only opens container consoles for accounts signed in with a username and password. If you signed in with an application programming interface (API) token, VisorMon shows **Container consoles need a username and password login. Proxmox does not allow API tokens to open LXC shells.** instead.

The terminal is the same one VisorMon uses for a node's shell, and its tools work the same way; see Nodes for the details. In short:

- The row of keys along the bottom has ⌨, **Esc**, **Tab**, **Ctrl** (for the next key), **^C**, **Paste**, **Copy**, **Select**, **Cmds**, the arrow keys, **|**, **~**, **/**, **-**, ****, **Home**, **End**, **PgUp** and **PgDn**.
- **Copy** copies the selected text, or the whole screen if nothing is selected. Turn on **Select** and drag to select. **Paste** types the phone's clipboard into the terminal; pasting several lines into a shell that would run each one as it arrives asks **Paste *n* lines?** first.
- **Cmds** opens your saved commands.
- At the top, the magnifier (**Find in output**) searches what the terminal has printed, and the **More** menu has **Text size** and **Share log**.
- A strip above the terminal shows how many lines of history it keeps, with **Change in Settings**.
- Going back, or reloading, while the terminal is connected asks first: **Close the terminal?** or **Reload the terminal?**, because anything still running in it stops.
- If the connection drops, a red strip says why, with **Reconnect**. What the terminal printed stays readable above the new session.

### The console lock

You can make VisorMon ask for Face ID, Touch ID or your passcode (on Android, your fingerprint, face or screen lock) before any console or node shell opens. Turn on **Face ID for consoles** in Settings; see Settings. On Android the switch is called **Biometric lock for consoles**, and it appears only when the phone has a fingerprint or face set up. Turning it on or off asks too, so someone holding your unlocked phone cannot switch it off.

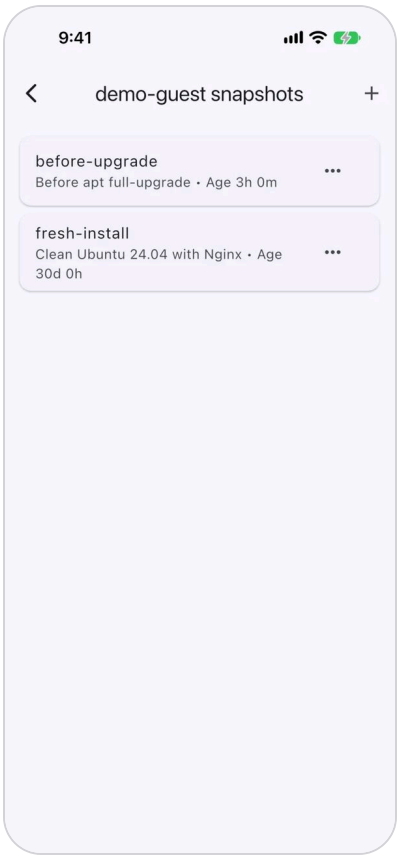
With the lock on, tapping **Console** shows **Unlock to open *name* console** and asks the phone to confirm it is you. Nothing connects to the server until it does. If you cancel, the console stays closed with **Unlock** to try again. On a phone with no passcode and no biometrics, VisorMon cannot ask, so it offers **Open without unlocking** instead.

### When the console does not open

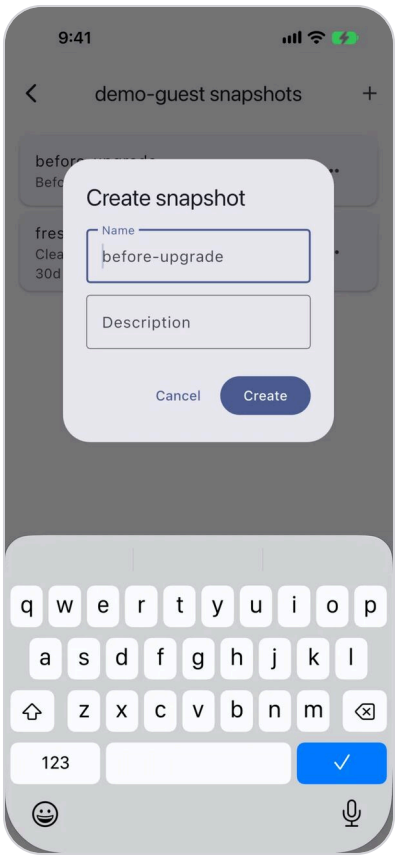
- **Read-only mode is on.** Proxmox opens a console through a request that VisorMon counts as a change, so read-only mode blocks it with **Read only is on. Turn it off in Settings to make this change.**
- **You are in the demo.** The sample data has no live console: **Sample data has no live console. Sign in to a Proxmox server to open one.**
- **A container, with an API token.** See A container's console.
- **No permission.** The **Console** button is grayed out without `VM.Console`.

## 6.4 Snapshots

A snapshot saves the state of a guest's disks so you can go back to it later, for example before an upgrade. **Snapshots** lists the guest's snapshots, newest first, with each one's description and age. Pull down to refresh the list.




A guest's snapshots.




Creating a snapshot.

**To create a snapshot:**

1. Tap **+** (**Create snapshot**) at the top.

2. Enter a **Name** and, if you like, a **Description**. Proxmox accepts names of 2 to 40 characters that start with a letter and use only letters, numbers, `-` and `_`.
3. Tap **Create**. VisorMon confirms with **Snapshot *name* created.**

Snapshots made in VisorMon save the disks only, not a running VM's memory (RAM).

**To roll back:** tap the menu button beside a snapshot and choose **Rollback**. VisorMon asks **Roll *name* back to *snapshot*? Current disk state is replaced.**; tap **Rollback**. Everything written to the guest's disks since the snapshot is lost. Proxmox stops the guest if it is running, and leaves it stopped afterwards unless the snapshot includes the VM's memory; start it again with **Start**.

**To delete a snapshot:** tap the menu button and choose **Delete**, then confirm **Delete snapshot *name*?** with **Delete**. The guest itself is not changed.

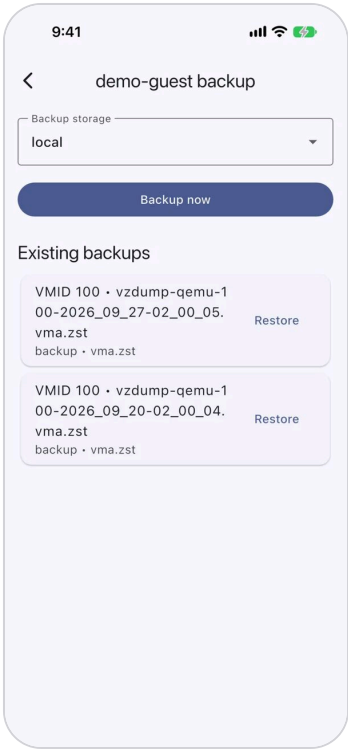
Creating, rolling back and deleting all need the `VM.Snapshot` privilege. Without it the **+** button is grayed out and the menu buttons are hidden.

## 6.5 Backup and restore

**Backup** backs the guest up to a storage on its node, and lists the backups of this guest that are already there.

**To back up a guest:**

1. Choose a **Backup storage**. Only storages on the guest's node that accept backups are listed.
2. Tap **Backup now**. VisorMon does not ask first. It starts a backup in snapshot mode, compressed with Zstandard (`zstd`), and says **Backup started. Follow it in Tasks.**



Back up now, or restore an existing backup.

The backup runs on the server as a task; follow it under Tasks on the guest's page or in cluster tasks (see Search, tasks and alerts).

**Existing backups** lists this guest's backups on the chosen storage. Choose another storage to see the backups there.

Backing up needs `VM.Backup` on the guest. Without it, a lock box says **No backup permission on this VM** and **Backup now** is grayed out. Proxmox also checks `Datastore.AllocateSpace` on the backup storage.

### Restoring a backup

VisorMon restores a backup as a *new* guest, with the next free VMID. The original guest is not touched, and the new one gets new media access control (MAC) addresses for its network cards, so the two can run side by side.

1. Tap **Restore** beside a backup.
2. VisorMon asks **Restore *backup* as a new VM with VMID *n*? The original guest is left unchanged.** (**CT** for a container).
3. Check the **Disk storage**, where the new guest's disks go (see below).
4. Tap **Restore**. VisorMon says **Restore started as VMID *n*.** The restore runs as a task.

**Disk storage** is not the **Backup storage**: one is where the backup is kept, the other is where the restored guest's disks are written. The menu lists only storages on the guest's node that can hold them, those whose content in Proxmox includes Disk image (for a VM) or Container (for a container). A Proxmox Backup Server storage is never listed, and neither is a directory storage such as `local` that holds only ISO images, templates and backups.

- **A VM** starts on **Original storage**: **Each disk goes back to the storage it was on when the backup was made.** Choose a storage instead to put every disk of the new VM there.
- **A container** always needs a storage, because Proxmox puts all of a restored container's volumes on one storage. VisorMon starts on the storage that holds the container's root disk now, or, if that one cannot hold containers, the first one that can. If no storage on the node can, the dialog says **No storage on *node* can hold container volumes.** and **Restore** is grayed out.

Restoring needs `VM.Allocate`. Proxmox also checks `Datastore.AllocateSpace` on the storage the disks go to.

> **Where the restored disks go. Original storage** uses the storages recorded in the backup. If one of them is not on the guest's node, for example a backup made on another node, Proxmox refuses the restore. Choose a storage under **Disk storage** instead.

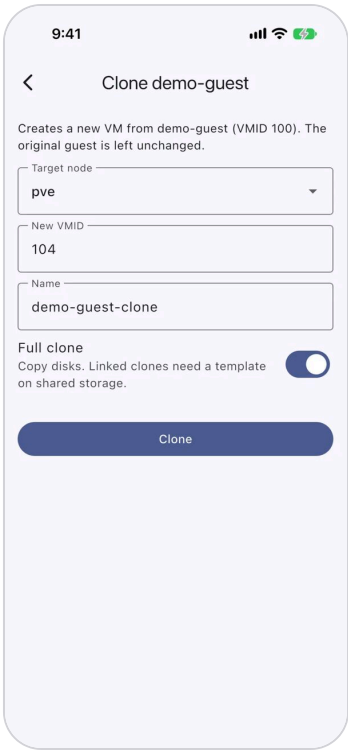
## 6.6 Clone

**Clone** makes a new guest from this one. The original is not changed.

1. Check the **Target node**. It starts as the guest's own node. Proxmox only allows another node when the guest's disks are on shared storage.
2. Check the **New VMID**. The next free one is filled in.
3. Check the **Name** (**Hostname** for a container). It starts as the guest's name with `-clone` added. Names start with a letter or number and use only letters, numbers and hyphens, up to 63 characters. If you clear the field, Proxmox picks a name.
4. Leave **Full clone** on to copy the disks. Turning it off asks for a linked clone, which shares the source's disks; Proxmox only makes linked clones of templates.
5. Tap **Clone**. VisorMon does not ask first.

The button shows **Cloning…** until the new guest appears in the cluster, for up to 3 minutes. VisorMon then says **Cloning *name* to VMID *n*.** and returns to the guest's page. Copying the disks can take longer; follow the clone task in Tasks.

The **Clone** button appears only if your account may clone some guest, and works on this guest with `VM.Clone`. Proxmox also checks `VM.Allocate` for the new VMID and `Datastore.AllocateSpace` on the storage the copy goes to.




Cloning a VM.

## 6.7 Migrate

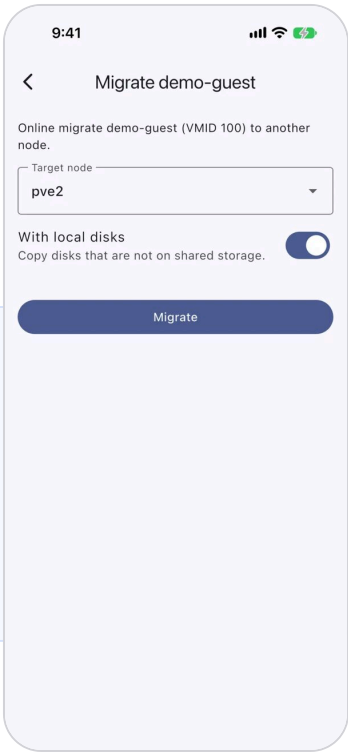
**Migrate** moves the guest to another node in the cluster. The screen says how: an online migration for a running VM, a restart migration for a running container, or an offline one for a stopped guest.

1. Choose the **Target node**. Only other nodes that are online are listed. On a single node, VisorMon says **Migration needs another online node in the cluster.**
2. For a VM, leave **With local disks** on to copy disks that are not on shared storage. A container has no such switch: Proxmox copies its local volumes on its own.
3. Tap **Migrate**, and confirm **Migrate *name* to *node*? The guest moves to that node.** with **Migrate**.

VisorMon says **Migrating *name* to *node*.** and returns to the guest's page. The migration runs as a task.

> **Containers.** Proxmox cannot move a running container live, so VisorMon uses a restart migration: the screen says **Restart migrate *name* (VMID *n*) to another node.**, and the confirmation says **Proxmox shuts the container down, moves it, and starts it again on *node*. It is offline until then.** If the move fails partway, Proxmox starts the container again on its old node. A stopped container moves offline and stays stopped.

The **Migrate** button appears only if your account may migrate some guest, and works on this guest with `VM.Migrate`.




Migrating to another node.

## 6.8 Hardware

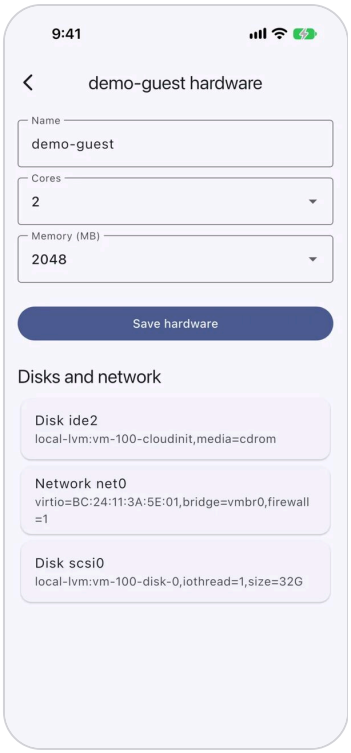
**Hardware** changes the guest's name, cores and memory, and lists its disks and network devices.

- **Name** (**Hostname** for a container): starts with a letter or number and uses only letters, numbers and hyphens, up to 63 characters.
- **Cores**: 1, 2, 4, 8 or 16, or the guest's current number.
- **Memory (MB)**: 512, 1024, 2048, 4096 or 8192, or the guest's current amount.

Tap **Save hardware**. For a running guest VisorMon says **Hardware saved. Some changes apply after reboot.**; changes Proxmox cannot make while the guest runs wait in Pending changes.

**Disks and network** lists each disk, mount point, network card and passthrough device with its settings, for reference. Change those in the Proxmox web interface.

VisorMon lets you edit these fields if your account has any of `VM.Config.Disk`, `VM.Config.CPU`, `VM.Config.Memory`, `VM.Config.Network`, `VM.Config.HWType` or `VM.Config.CDROM` on the guest; otherwise a lock box says **Read-only on this VM**. Saving sends all three values, and Proxmox checks each: cores need `VM.Config.CPU`, memory `VM.Config.Memory`, a VM's name `VM.Config.Options` and a container's hostname `VM.Config.Network`.




Name, cores, memory, disks and network.

## 6.9 Options

| | |
|---|---|
| **Start at boot** | Starts the guest when its node starts. |
| **Protection** | While it is on, Proxmox refuses to remove the guest or its disks. |
| **QEMU guest agent** | VMs only. Tells Proxmox the VM runs the guest agent, which VisorMon needs to show a VM's disk usage. The agent also has to be installed in the VM. Proxmox applies the change the next time the VM starts, or after **Reboot**. Turning it on or off keeps the agent's other settings made in the Proxmox web interface, such as its type. |
| **Startup (order,up,down)** | The guest's place in the boot order and its delays, for example `order=1,up=30`. Leave it empty to remove the setting. |

Tap **Save options**. VisorMon sends only the options you changed and says **Options saved.** If nothing changed, it says **No changes to save.**

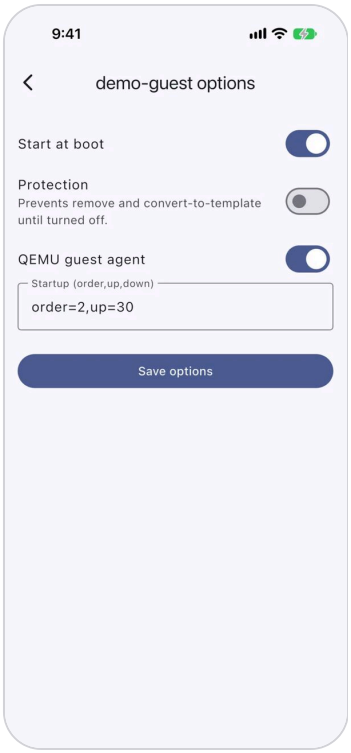
Options need `VM.Config.Options` on the guest. For a VM, Proxmox also checks `Sys.Modify` on `/` when the startup setting changes, so changing or emptying a VM's **Startup (order,up,down)** needs both.

## 6.10 Tags

Tags are short labels, such as `prod` or `web`, that help you group guests. The Guests tab and cluster search can filter and group by them; see Guests and Search, tasks and alerts.

1. Type a tag in **Add tag** and tap **Add tag**. Tags start with a letter or number and use only letters, numbers, `_` and `-`, up to 64 characters.
2. Remove a tag with the **×** on its chip.
3. Tap **Save tags**. Nothing changes on the server until you do. VisorMon says **Tags saved.**

Tags need `VM.Config.Options`. Depending on your datacenter's tag settings, Proxmox may also require `Sys.Modify` on `/` for some tags.



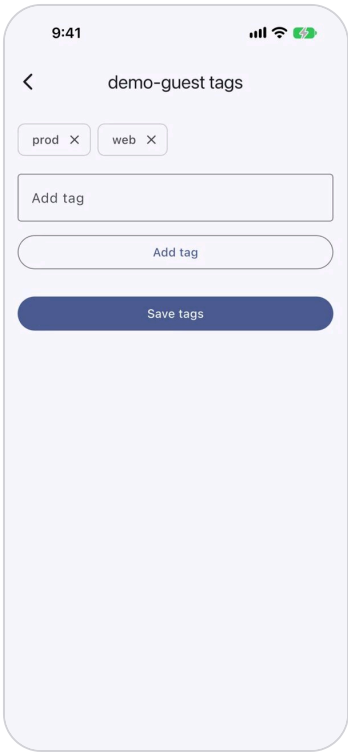

Guest options.

## 6.11 Pending changes

Some changes cannot be made while a guest is running, such as more memory for a VM without memory hotplug. Proxmox keeps them as pending changes and applies them when the guest is next started, or rebooted with **Reboot**.

**Pending** lists each waiting change by its Proxmox setting name, with its current value and what it will become, for example **Now 2048 → 4096**, or **Will delete** for a setting that will be removed. With no changes waiting it says **No pending changes.**

To cancel a change, tap **Revert** and confirm **Revert *setting* on *name*? The pending value is discarded.** with **Revert**. VisorMon enables **Revert** when your account has `VM.Config.Options` on the guest.




Editing tags.

## 6.12 Cloud-Init

Cloud-Init sets up a VM's user, network and keys from settings Proxmox writes to a small Cloud-Init drive. VMs only; the VM needs a Cloud-Init drive and an image that runs Cloud-Init.

**User** The user Cloud-Init sets up, instead of the image's default user.

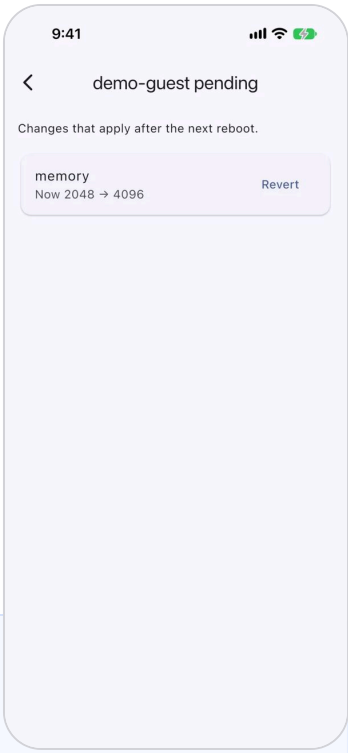
**Password** That user's password. Leave it blank to keep the current one.

**DNS servers**

| | |
|---|---|
| | Domain Name System (DNS) servers, separated by spaces, for example `1.1.1.1 8.8.8.8`. |
| **Search domain** | The domain to search for short host names. |
| **IP config (net0)** | The address of the VM's first network card, for example `ip=dhcp` or `ip=192.168.1.50/24,gw=192.168.1.1`. |
| **SSH public keys** | Secure Shell (SSH) keys to allow, one per line. |

Emptying a field (other than **Password**) removes that setting.

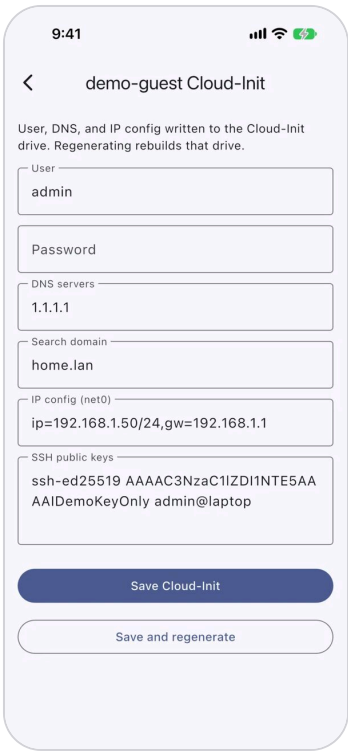
- **Save Cloud-Init** saves the settings: **Cloud-Init saved.**
- **Save and regenerate** also rebuilds the Cloud-Init drive, after asking **Save Cloud-Init and regenerate the drive for *name*?** Tap **Regenerate**.




A memory change waiting for a reboot.

**SSH keys.** Keys already set on the VM fill the **SSH public keys** field when the screen opens, so saving keeps them unless you change or remove them. Proxmox checks each key when you save, and refuses the save if one is not a valid public key.

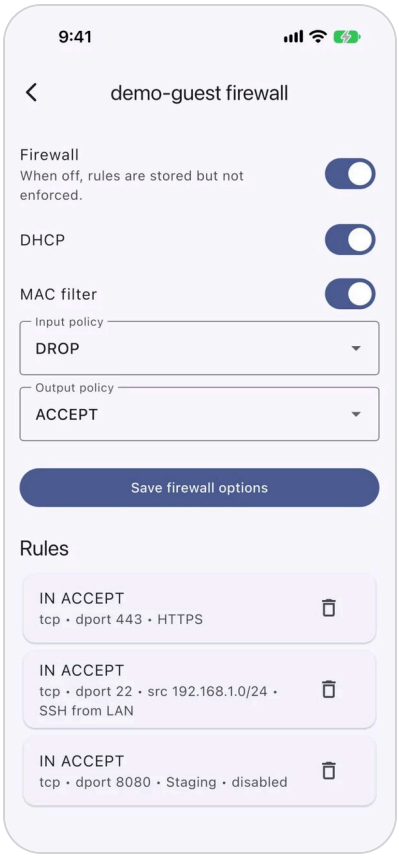
Cloud-Init needs `VM.Config.Cloudinit` on the VM; otherwise a lock box says **Read-only on this VM** and the buttons are grayed out.



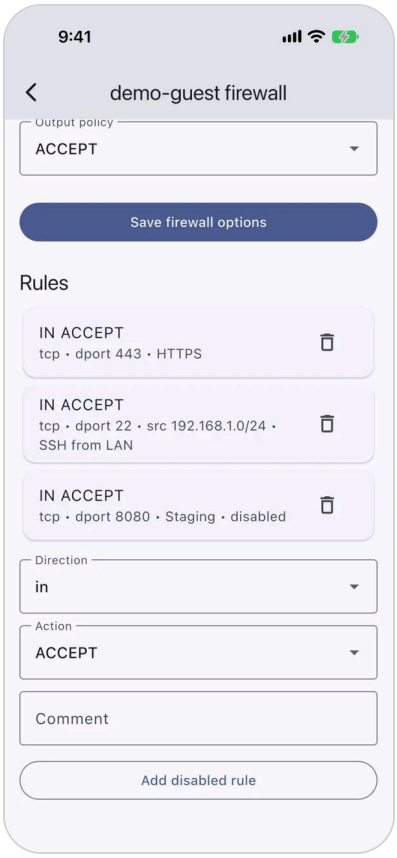

Cloud-Init settings.

## 6.13 Firewall

**Firewall** controls the guest's own Proxmox firewall. It only filters the guest's traffic when the firewall is also on for the datacenter and on the guest's network card.




Firewall options.




Firewall rules.

| | |
|---|---|
| **Firewall** | Turns the guest's firewall on or off. When it is off, the rules are kept but not enforced. |
| **DHCP** | Lets Dynamic Host Configuration Protocol (DHCP) traffic through, so the guest can get an address. |
| **MAC filter** | Drops traffic the guest sends from MAC addresses other than its own. |
| **Input policy, Output policy** | What happens to traffic no rule matches: `ACCEPT`, `DROP` or `REJECT`. |

Tap **Save firewall options**. VisorMon says **Firewall options saved.**

## Rules

**Rules** lists the guest's rules in order. Each shows its direction and action, for example **IN ACCEPT**, and its details: macro, protocol, interface, ports, source and destination, comment, and **disabled** if it is switched off.

- **To delete a rule**, tap its trash button and confirm **This guest firewall rule is removed. The guest keeps running.** with **Delete**.
- **To add a rule**, choose a **Direction** (`in` or `out`) and an **Action**, type a **Comment**, and tap **Add disabled rule**. If you leave the comment empty, VisorMon uses `pgui`.

New rules are added switched off, so a mistake cannot cut off access. They have no protocol or port, so they would match all traffic in their direction. Edit and enable them in the Proxmox web interface.

Changing the firewall needs `VM.Config.Network` on the guest; otherwise a lock box says **Read-only on this VM**.

## 6.14 Convert to template

A template is a guest you clone new guests from. It cannot be started. **Convert to template** turns this guest into one.

1. Stop the guest. If it is running, VisorMon says **Stop the guest before converting it to a template.**
2. Proxmox refuses to convert a guest that has snapshots, so delete them first in Snapshots.
3. Tap **Convert to template**. VisorMon asks **Convert *name* (VMID *n*) to a template? This cannot be undone. Clone first if you still need a runnable guest.** Tap **Convert**.

VisorMon says ***name* is now a template.** The page then shows **template** and has no **Power** section.

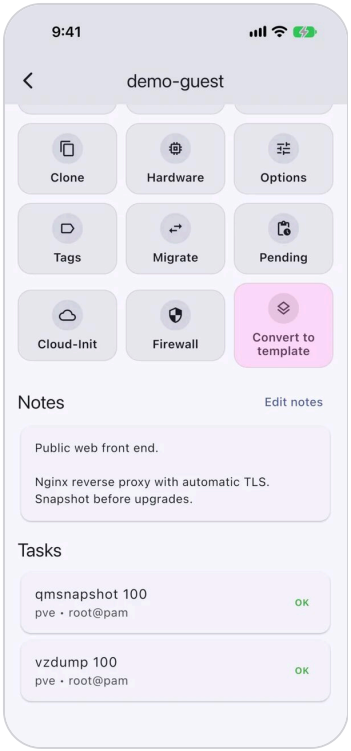
The button appears only if your account may create guests, and works on this guest with `VM.Allocate`.

## 6.15 Notes and tasks

**Notes** shows the guest's notes from Proxmox, or **No notes**. VisorMon shows them as plain text, so Markdown formatting appears as typed; notes written in HTML are shown formatted.

To change them, tap **Edit notes**, edit the text and tap **Save**. Editing notes needs `VM.Config.Options`; without it **Edit notes** is grayed out.

**Tasks** lists up to 12 of the guest's recent tasks from the cluster's task list, newest first, such as **vzdump 100** for a backup. Each shows its node and user and its result: **running**, **OK**, or the error in red. Tap a task to read its log; see Search, tasks and alerts. With nothing recent, it says **No recent tasks for this guest.**



Notes and recent tasks.

## 6.16 Creating a VM or container

VisorMon creates a simple VM or container with one disk and one network card. For anything more, such as more disks, a CPU type or passthrough, use the Proxmox web interface, or change the guest afterwards.

On the Guests tab, tap **+** and choose **Create VM** or **Create container**. With the VMs or Containers chip on, **+** goes straight to that form. The **+** button only appears if your account may create guests; see Guests.

The form is titled **Create VM** or **Create CT**. Its fields, with what they start as:




The Create VM form.

| Field | Starts as | Notes |
|---|---|---|
| **Node** | The first node | Changing it reloads the storages and images for that node. |
| **VMID** | The next free VMID | |
| **Name** (VM) or **Hostname** (container) | `vm` or `ct` and the VMID, for example `vm105` | Starts with a letter or number; letters, numbers and hyphens, up to 63 characters. |
| **Cores** | 1 | 1, 2, 4 or 8. |
| **Memory (MB)** | 512 | 512, 1024, 2048 or 4096. |
| **Disk storage** | The first storage on the node that holds disks | For a VM, storages that accept disk images; for a container, also those that accept container disks. |
| **Disk (GB)** | 8 GB | 4, 8, 16 or 32 GB. |
| **ISO (optional)** | The first ISO image found on the node | VMs only. A CD or DVD image to start from. Choose **None** for no image. Not shown if the node has no ISO images. |
| **Template** | The first container template found on the node | Containers only, and required. Without any, the form says **No container templates found on this node.** |

| Field | Starts as | Notes |
|---|---|---|
| **Root password** | Empty | Containers only. At least 5 characters. |
| **Bridge** | `vmbr0` | The network bridge for the network card. |

Tap **Create**. VisorMon does not ask first. If something is missing it says **Enter a valid name and choose node and storage.** or, for a container, **Containers need a template and a password of at least 5 characters.**

### What VisorMon creates

- **A VM** set up for Linux, with a VirtIO SCSI disk of the chosen size on the chosen storage, a VirtIO network card on the bridge, and the ISO image, if any, in a CD drive.
- **A container** that is unprivileged, from the chosen template, with a root disk of the chosen size, the root password you entered, and a network card `eth0` on the bridge that gets its address by DHCP.

VisorMon says **Creating *name* (VMID *n*).** and returns to the Guests tab. Proxmox creates the guest as a task, and the new guest is not started. Open it and tap **Start** when it appears.

Creating a guest needs `VM.Allocate`. Proxmox also checks `Datastore.AllocateSpace` on the disk storage.

## 6.17 Permissions, read-only mode and VisorMon Pro

VisorMon reads your Proxmox account's permissions and hides or grays out what you may not do. Until it has read them, or if it cannot, it offers everything and leaves the decision to Proxmox. A lock box saying **Read-only on this VM** on one of the guest's screens means your account may not change that screen's settings. It is not the same as VisorMon's read-only mode.

### Privileges by action

A privilege counts when it is granted on the guest (for example on `/vms/100`), on its pool, or higher up, such as on `/`, unless the table names another path. For what privileges and roles are, see Security and privacy.

| Action | VisorMon looks for | Proxmox also checks |
|---|---|---|
| See the guest's page | | `VM.Audit` |
| See the history charts | `VM.Audit` | |
| See a VM's disk usage | | `VM.GuestAgent.Audit` or `VM.GuestAgent.Unrestricted` on current Proxmox VE |

| Action | VisorMon looks for | Proxmox also checks |
|---|---|---|
| Start, Shutdown, Reboot, Stop, Reset, Suspend, Resume | `VM.PowerMgmt` | |
| Console | `VM.Console` | A password sign-in, for a container |
| Create, roll back or delete a snapshot | `VM.Snapshot` | Proxmox also accepts `VM.Snapshot.Rollback` for a rollback, but VisorMon shows the menu only with `VM.Snapshot` |
| Backup now | `VM.Backup` | `Datastore.AllocateSpace` on the backup storage |
| Restore a backup | `VM.Allocate` | `Datastore.AllocateSpace` on the target storage |
| Clone | `VM.Clone` | `VM.Allocate` for the new VMID, `Datastore.AllocateSpace` |
| Migrate | `VM.Migrate` | |
| Save hardware | Any of `VM.Config.Disk`, `VM.Config.CPU`, `VM.Config.Memory`, `VM.Config.Network`, `VM.Config.HWType`, `VM.Config.CDROM` | `VM.Config.CPU` and `VM.Config.Memory`, plus `VM.Config.Options` (VM name) or `VM.Config.Network` (container hostname) |
| Save options | `VM.Config.Options` | `Sys.Modify` on `/`, when a VM's startup setting changes |
| Save tags | `VM.Config.Options` | Possibly `Sys.Modify` on `/`, depending on your datacenter's tag settings |
| Revert a pending change | `VM.Config.Options` | |
| Save or regenerate Cloud-Init | `VM.Config.Cloudinit` | |
| Firewall options and rules | `VM.Config.Network` | |
| Convert to template | `VM.Allocate` | |
| Edit notes | `VM.Config.Options` | |
| Create a VM or container | `VM.Allocate` anywhere | `VM.Allocate` for the new VMID, `Datastore.AllocateSpace` on the disk storage |

The **Clone**, **Migrate** and **Convert to template** buttons, and the Guests tab's **+**, are hidden when your account has that privilege on no guest at all. The `root@pam` user passes every check.

## Read-only mode

With read-only mode on (**Read only** in Settings), VisorMon sends no changes to your server. Everything on a guest's page that changes something is blocked: power actions, notes, every **Save** button, snapshots, backups and restores, clones, migrations, converting to a template, firewall rules, reverting pending changes and creating guests. Opening a console is blocked too. You can still tap the buttons, and confirm where VisorMon asks first, but instead of making the change VisorMon shows **Read only is on. Turn it off in Settings to make this change.** and records the attempt in the audit log. See Security and privacy.

In the demo, changes are not sent either; VisorMon says **This is sample data. Nothing was sent.**

## VisorMon Pro

Nothing on a guest's page needs a separate purchase. Using VisorMon with a real Proxmox server needs the free trial or VisorMon Pro, and that covers everything in this chapter; see VisorMon Pro.

CHAPTER 7

# Storage

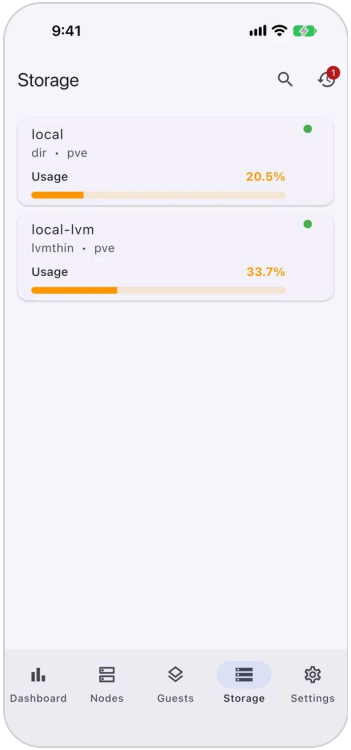
Your storage, what is on it, and what you can do with it.

## 7.1 The Storage tab

The **Storage** tab lists each storage on your Proxmox VE servers and how full it is. Tap a storage to see what is on it.

Each storage has a card:

| | |
|---|---|
| **Name** | The storage's ID in Proxmox, such as `local` or `local-lvm`. |
| **Type and node** | The storage type as Proxmox names it, then the node the card is for, for example **dir · pve** or **lvmthin · pve**. |
| **Usage** | How much of the storage is used, as a percentage and an orange bar. |
| **Dot** | Green when Proxmox has current figures for the storage. Red when it has none, for example because the node is offline or the storage can't be reached. A storage with a red dot usually shows **0.0%**, because there are no figures to show. |




The Storage tab.

Cards are in alphabetical order by name.

Proxmox reports storage per node, so VisorMon lists a storage once for each node that can use it. A storage shared by several nodes, such as a network share or a Ceph pool, appears once per node, and so does `local`, which is separate on each node. VisorMon does not mark which storages are shared. For the space used across the whole cluster, with each shared storage counted once, see the **Resources** pane in The dashboard.

### Keeping the list up to date

The list is loaded when VisorMon connects to your server, and it does not refresh on a timer. Pull down on the list to update it.

The buttons at the top right switch servers, search the cluster and open the cluster's tasks. See Search, tasks and alerts.

### When the list is empty

| | |
|---|---|
| **No permission** | **This storage list is blank because you don't have permission to view it.** Your account has no storage privileges, or Proxmox turned down part of the request. Tap **Retry** after the permissions are fixed. |
| **No storage in your scope** | **No datastores are visible to this account.** The request worked, but Proxmox returned no storage for this account. |
| **An error message** | VisorMon could not reach the server or the server returned an error. Tap **Retry**. For help, see Troubleshooting. |

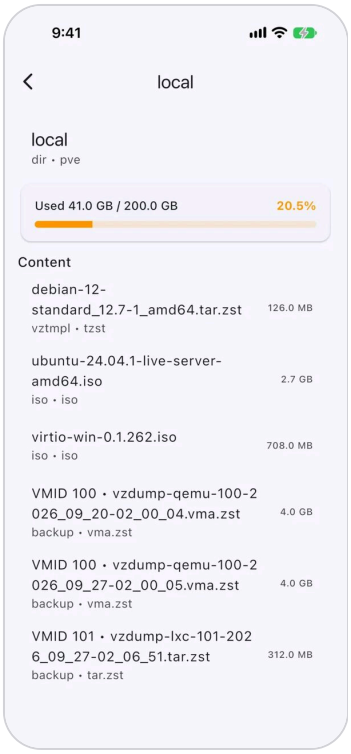
To see a storage, your account needs `Datastore.Audit` on it: on `/storage/<storage ID>`, or on a path above it such as `/storage` or `/`. The built-in `PVEAuditor` role includes it. You set roles under **Datacenter › Permissions** in the Proxmox web interface.

## 7.2 A storage's contents

Tap a card on the Storage tab to open that storage. You can also open one from cluster search (see Search, tasks and alerts).

The screen shows:

- The storage's name, with its type and node below it.
- A card with the space used and the total, for example **Used 41.0 GB / 200.0 GB**, and the percentage used.
- Under **Content**, everything Proxmox reports on the storage, in alphabetical order.



The contents of a storage.

Each item shows:

| | |
|---|---|
| **Name** | The file or volume name. A backup's name starts with the ID of the guest it belongs to, for example **VMID 100 • vzdump-qemu-100-2026_09_27-02_00_05.vma.zst**, so backups sort together under V. |
| **Details** | The content type, then, for a disk, the ID of the guest it belongs to (VMID), then the format, for example **iso • iso**, **backup • vma.zst** or **images • VMID 100 • raw**. |
| **Size** | On the right. A dash means Proxmox did not report a size for the item. |

VisorMon counts sizes in units of 1,024, so 1 GB here is what Proxmox calls 1 GiB. If the storage holds nothing you can see, the list says **No content**.

Each card on the Storage tab is for one node, and the contents are read from that node. Opening `local` on two nodes shows two different lists.

This screen does not refresh. To see changes, go back and tap the storage again. If its node is offline, the contents can't be read and VisorMon shows the error with a **Retry** button.

## Content types

The details line uses Proxmox's own names for the kinds of content:

| Shown as | What it is |
|---|---|
| `iso` | An ISO image, for installing an operating system in a virtual machine (VM) |
| `vztmpl` | A container template |
| `backup` | A backup of a VM or container |
| `images` | A VM's disk, or the disk of a VM template |
| `rootdir` | A container's disk |

Other kinds of content that Proxmox reports are listed the same way, under their Proxmox names.

## What your account can see

To open a storage's contents, your account needs `Datastore.Audit` or `Datastore.AllocateSpace` on the storage. Without either, VisorMon shows **No permission** where the list would be.

Proxmox also leaves out items your account is not allowed to use, so two accounts can see different lists for the same storage:

| Items | Privileges Proxmox asks for |
|---|---|
| Everything on the storage | `Datastore.Allocate` on the storage |
| ISO images and container templates | `Datastore.Audit` or `Datastore.AllocateSpace` on the storage |
| A guest's backups | `Datastore.AllocateSpace` on the storage, and `VM.Backup` on that guest |
| A guest's disks | `Datastore.Audit` on the storage, and `VM.Config.Disk` on that guest |

For example, an account with only the `PVEAuditor` role sees ISO images and templates, but no backups or disks.

## 7.3 What you can do from the Storage tab

The Storage tab is for looking. Its screens send no changes to your server: there is no way to upload, download from a URL, delete or restore anything from them.

Related jobs live elsewhere:

| To | Go to |
|---|---|
| Back up a guest, or restore one of its backups as a new guest | The guest's page, **Backup**. See Working with a guest. |
| Create a VM from an ISO image, or a container from a template | The **+** button on the Guests tab. See Guests. |
| See a node's physical disks | The node's page, **Disks**. See Nodes. |
| Upload an ISO image or template, download one from a URL, or delete a file or backup | The Proxmox web interface |

Because nothing here changes your server, read-only mode makes no difference to the Storage tab (see Security and privacy). On a real server, the Storage tab needs VisorMon Pro or the free trial, like the rest of the app; in demo mode it shows sample storage (see VisorMon Pro).

CHAPTER 8

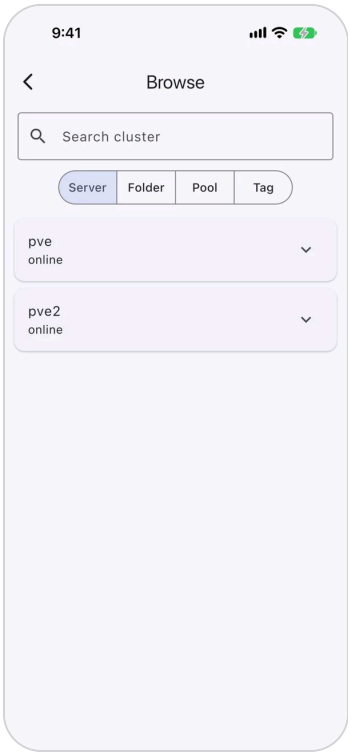
# Search, tasks and alerts

Searching the cluster, following tasks and their logs, notifications and alerts.

## 8.1 Searching the cluster

Cluster search finds anything in your cluster that your account can see: nodes, virtual machines (VMs), containers, storage and resource pools. Open it from the magnifier button, **Search cluster**, at the top right of the **Dashboard**, **Nodes**, **Guests**, **Storage** and **Settings** tabs, or of the guests dashboard.

1. Tap the magnifier. The **Browse** screen opens, with everything grouped by node.
2. Type in the **Search cluster** box. Its hint, **Name, VMID, node, pool, tag**, lists the main things you can search for. The list narrows as you type.
3. To group the results differently, tap **Server**, **Folder**, **Pool** or **Tag**.
4. Tap a group to open it, and tap it again to close it.
5. Tap an item to open it.



Browse, in the Server view.

Each item shows its name and a line of details, for example **VM • VMID 100 • pve • running • pool homelab** for a guest, or **pve • dir • available** for a storage. If nothing matches, the screen says **No matching resources**.

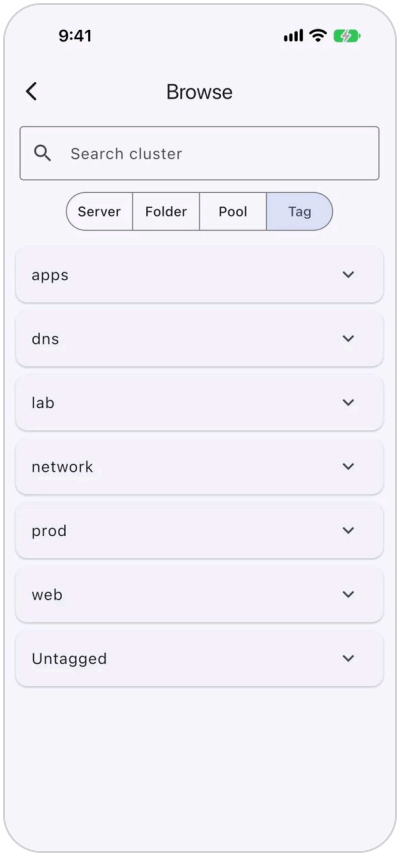
Browse opens in the Server view with an empty search box each time.

### Views

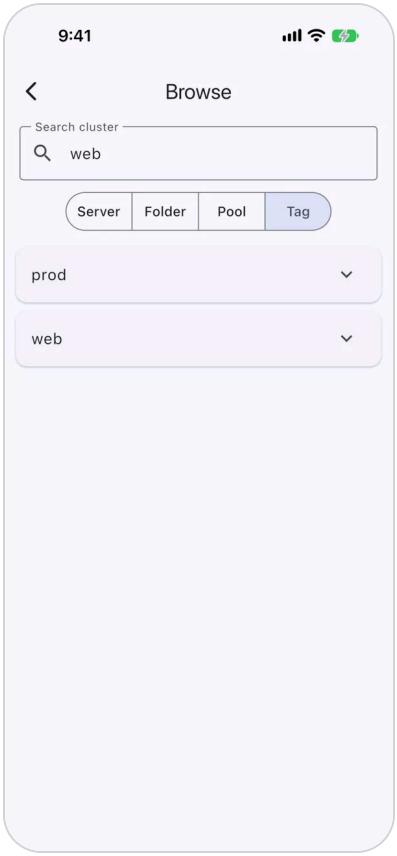
| View | Groups |
|---|---|
| **Server** | A group for each node, with its status, such as **online**, under its name. Inside are the guests and storage on that node. If your cluster has resource pools, a **Pools** group at the end lists them. A node with nothing on it says **No guests or storage**. |
| **Folder** | A group for each kind of item: **Nodes**, **Virtual machines**, **Containers**, **Storage** and **Pools**. Anything else is under **Other**. |
| **Pool** | A group for each resource pool, with the guests in it. Everything that is not in a pool, including nodes and storage, is under **No pool**. |

| View | Groups |
| --- | --- |
| **Tag** | A group for each tag. A guest with several tags appears in the group tag, including nodes and storage, is under **Untagged** at the end. |

Inside a group, nodes come first, then VMs, then containers, then storage, then pools. Guests are in order of VMID, and everything else by name.




The Tag view.




Searching for web. The guest tagged both prod and web is in both groups.

## What the search matches

The search ignores capital letters and matches any part of:

- a name, a VMID, a node's name, a pool's name or a tag;
- the details line, such as a status ( `running` , `stopped` , `online` ), a storage type ( `dir` , `zfspool` ) or the word `template` .

So typing `template` finds your templates, and typing `stopped` finds the guests that are off.

In the Server view, if what you type matches a node, that node's group shows everything on the node, not only the items that match.

### Opening a result

| Tap | To open |
|---|---|
| A node | The node's page. See Nodes. |
| A VM or container | The guest's page. See Working with a guest. |
| A storage | The storage's contents. See Storage. |
| A pool | A list of the guests in the pool. The guests in that list can't be opened from it: go back and tap the guest in the search results instead. |

A few kinds of item can't be opened, such as software-defined networking (SDN) zones. Tapping one shows a message such as **sdn is not opened yet**.

### What Browse shows

Browse loads the cluster's resources when you open it and does not refresh on its own. Pull down on the list to load it again.

It lists only what Proxmox lets your account see. Nodes are always listed. A guest needs `VM.Audit` on it, a storage `Datastore.Audit` and a pool `Pool.Audit`. A guest's pool shows in its details, and in the Pool view, only if you have `Pool.Audit` on that pool.

Browse only finds and opens things. It sends no changes to your server.

## 8.2 Cluster tasks

Proxmox runs longer jobs, such as backups, migrations and starting a guest, as tasks, and keeps a log of each one. The task list shows the cluster's recent tasks, from every node.

To open the list, tap the clock button, **Cluster tasks**, at the top right of any main tab. When tasks are running, a number on the button shows how many.

The newest tasks are at the top. Each task shows:

**Name** The task type, followed by what it is for, usually a guest's VMID. For example, **vzdump 102** is a backup of guest 102. See Reading task names.

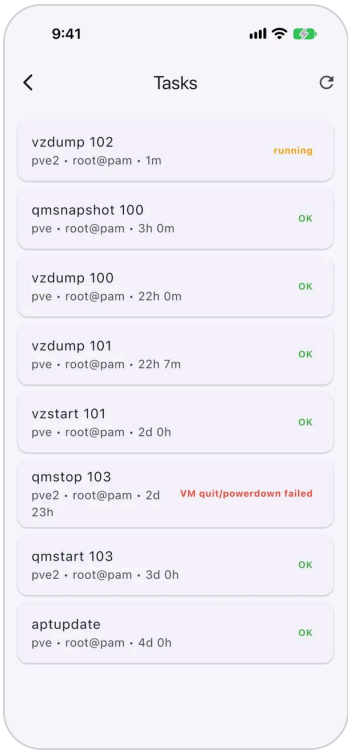
**Details** The node it ran on, the user who started it, and how long ago it started, for example **pve2 • root@pam • 1m**.

**Status** **running** while it runs, **OK** when it finished without problems, or the message Proxmox recorded, in red, such as **VM quit/powerdown failed**. Warnings are shown in red too.

The list refreshes on its own while VisorMon is open, as often as **Settings › Data refresh** says: about every 5 seconds, 30 seconds or 5 minutes (see Settings). It also refreshes right after you start some tasks from VisorMon, such as a backup, a restore, a migration or a clone. To refresh it now, tap **Refresh** at the top right, or pull down on the list. If there are no tasks, it says **No cluster tasks**.

The list has no filter or search. To see the tasks of one node, open the node's **Tasks** (see Nodes). A guest's page lists that guest's recent tasks (see Working with a guest).

With `Sys.Audit` on `/`, you see every task in the cluster. Without it, Proxmox shows only the tasks started by your own account.



The cluster's recent tasks.

## Reading task names

Proxmox names tasks with short codes. Names that start with `qm` are for VMs and names that start with `vz` are for containers, except `vzdump`, which backs up both. Some common ones:

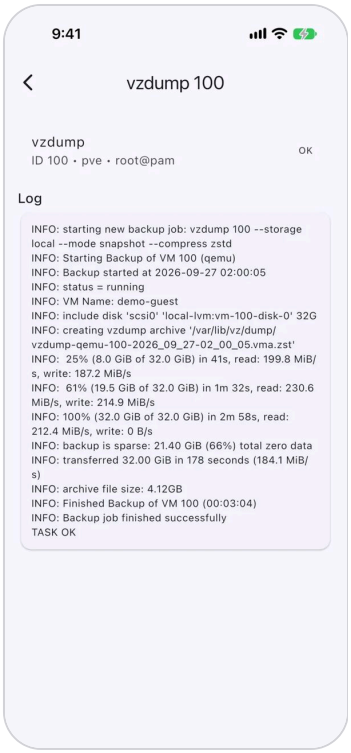
| Task | What it does |
|---|---|
| `vzdump` | Backs up a VM or container |
| `qmstart`, `qmshutdown`, `qmstop`, `qmreboot` | Starts, shuts down, stops or reboots a VM |
| `vzstart`, `vzshutdown`, `vzstop`, `vzreboot` | The same for a container |
| `qmsnapshot`, `vzsnapshot` | Takes a snapshot |
| `qmigrate`, `vzmigrate` | Migrates a guest to another node |
| `qmclone`, `vzclone` | Clones a guest |
| `qmrestore`, `vzrestore` | Restores a backup |
| `aptupdate` | Refreshes a node's list of available updates |

## A task's log

Tap a task to open it. At the top are the task type, its details (for example **ID 100 • pve • root@pam**) and its status. Below, under **Log**, are the lines Proxmox recorded, in a fixed-width font. Touch and hold the log to select and copy text. A task with an empty log says **No log lines**.

**Good to know:**

- VisorMon shows the first 50 lines of the log. For a longer log, such as a large backup's, read the rest in the Proxmox web interface.
- The log is read once, when you open the task. For a task that is still running, go back and open it again to see newer lines.
- The status at the top is the one the task had when you opened it.
- You can always read the logs of your own tasks. For other people's tasks, Proxmox asks for `Sys.Audit` on the node (`/nodes/<node>`). Without it, VisorMon shows **No permission** where the log would be.



A backup task's log.

### Stopping a running task

A task that was running when you opened it has an **Abort** button at the top right.

1. Open the running task from the list.
2. Tap **Abort**.
3. VisorMon asks **Abort task?**, naming the task, and reminds you that **Work already finished stays as it is.** Tap **Abort** to stop it, or **Cancel**.

VisorMon asks Proxmox to stop the task, then reloads the log and the task list. The open task page keeps its old status and its **Abort** button, so go back to the list to see how the task ended.

- **Privileges.** You can stop your own tasks. To stop someone else's, Proxmox asks for `Sys.Modify` on the node (`/nodes/<node>`). If Proxmox turns the request down, VisorMon shows **No permission** where the log would be.
- **Read-only mode.** Stopping a task is a change, so read-only mode blocks it. VisorMon shows **Read only is on. Turn it off in Settings to make this change.** where the log would be. See Security and privacy.
- **Audit log.** VisorMon records the request in its audit log, or records that read-only mode blocked it (see Security and privacy).
- **Demo mode.** Nothing is sent. VisorMon shows **This is sample data. Nothing was sent.**

## 8.3 Alerts and notifications

VisorMon has one alert: it tells you when it can't refresh the dashboard from your server. It shows the alert in the app and, if you allow it, as a notification on your phone.

### When VisorMon raises the alert

The dashboard refreshes on its own as often as **Settings › Data refresh** says, even while you are on another tab, and when you pull down on it. If a refresh fails, for example because the server is off, your phone has left the network the server is on, or the server does not answer in time, VisorMon does the following, as long as it has loaded the dashboard successfully on this device at least once before:

- A banner at the top of the main tabs reads **Showing saved data from**, followed by the date and time of the last successful refresh.
- A **Needs attention** bar appears below it, reading **Could not refresh**. Tap it to open the **Alerts** screen, which lists **Could not refresh**: **Showing the last saved dashboard.**
- The dashboard keeps showing the figures from its last successful refresh, and **Last updated** stays at that time. If nothing has loaded since you opened VisorMon, the dashboard shows the error instead (see The dashboard).
- If **Enable notifications** is on, your phone shows a notification titled **Could not refresh**, with the text **Showing the last saved dashboard.**

You get one notification each time the connection is lost, not one per failed refresh. When a refresh succeeds again, the banner and the **Needs attention** bar go away, and the next failure raises the alert again.

The banner and the bar show on the five main tabs, not on the screens you open from them. For help with the connection itself, see Troubleshooting.

> **Not the same thing.** The dashboard's guest overview has a sort order also called **Needs attention**, which puts the guests closest to their limits first. It has nothing to do with this alert. See Choosing which guests come first.

### Turning notifications on or off

Phone notifications are controlled by **Enable notifications** in Settings, which is on until you turn it off (see Settings). Turning it off stops only the phone notification. The banner and the **Needs attention** bar still appear in the app.

Your phone must also allow VisorMon to send notifications:

- **iPhone.** VisorMon asks for permission the first time it starts. If you said no, allow notifications for VisorMon in the iPhone's Settings app. The switch in VisorMon can't override the phone's setting.
- **Android.** VisorMon asks for permission when you turn **Enable notifications** on. The switch starts out on, so if you have never been asked, turn it off and on again. The notifications use a channel called **VisorMon Alerts**, which you can also manage in Android's settings for the app.

### What the alert can't do

- **It only covers the connection.** VisorMon does not notify you when a guest stops, a node goes offline, a task fails, storage fills up or a reading is high. In a cluster, if one node goes down but the server VisorMon connects to still answers, the refresh succeeds and there is no alert. You see these things when you open the dashboard, the task list or the Storage tab (see The dashboard).
- **It only works while VisorMon is running.** VisorMon does not use push notifications. The app itself does the checking, on your phone, so once your phone suspends or closes VisorMon, nothing is checked and no alert arrives.
- **It needs a first successful load.** Until VisorMon has loaded the dashboard from a server on this device, a failed refresh raises no alert.

To be told about problems around the clock, set up notifications in Proxmox VE itself, under **Datacenter › Notifications** in the web interface.

CHAPTER 9

# Settings

Every setting, what it does and its default.

## 9.1 The Settings tab

Settings is the last tab at the bottom of the screen. It holds your server connections, the safety switches, how the app looks and behaves, your VisorMon Pro subscription, and links to help. A change takes effect as soon as you make it; there is no Save button. Settings belong to this phone and apply to every server you use on it.

The top bar has the same buttons as the other tabs: the server button for switching servers (shown once you have a saved server), **Search cluster** and **Cluster tasks**. See The dashboard for what they do.

| From top to bottom | What it holds |
|---|---|
| **Connection** | Who you are signed in as, how, and to which server |
| **Saved hosts** | The servers kept on this phone, and **Add a server** |
| **Read only** and **Data refresh** | Blocking changes from this app, and how often figures update on their own |
| **Appearance** | Light or dark, then the rows below it: date and time format, the terminal settings, saved commands, the two locks, notifications, usage statistics, the audit log and **Arrange panes** |
| **VisorMon Pro** | Your subscription, restoring a purchase, and replaying the tour |
| **Sign out**, **Help** and **About** | Signing out, Proxmox's own help, the version, open-source licenses |

The rows after the three theme choices have no heading of their own; they run on under **Appearance** until **VisorMon Pro**. This chapter takes them one group at a time, in the order the app shows them. All settings and their defaults at the end of the chapter lists every one in a table.

## 9.2 Connection and saved hosts

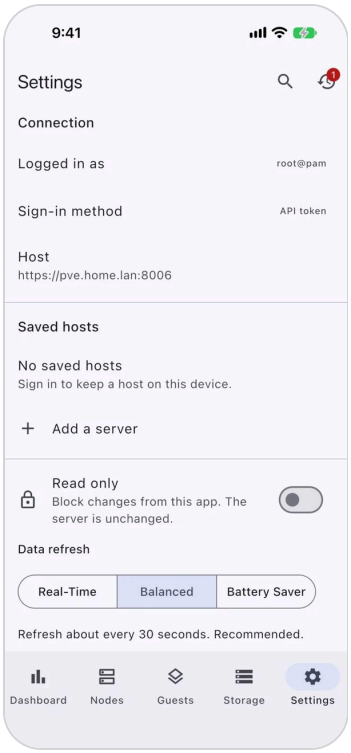
The **Connection** rows show the server you are using now. They are for reading only.

**Logged in as** The Proxmox user name you signed in with, without the realm, for example `root`.

**Sign-in method** **Password**, or **API token** for an application programming interface (API) token.

**Host** The address of the server, for example `https://pve.example.com:8006`.

With the sample data from **Try demo**, these rows show `demo@pam` and `https://demo.local:8006`.



The top of Settings.

### Saved hosts

Every server you have signed in to on this phone is listed here, most recently used first. Each shows the user and server name, for example **root@pve.example.com**, with its address underneath. The server you are using is marked **current**.

- **To switch servers,** tap another one. VisorMon signs out of the current server and signs in to the one you tapped. A server you use with an API token signs straight in with its saved secret. Switching away from a server you signed in to with a password ends that session, so going back to it normally means signing in again: type the password, or use **Login with Face ID**.
- **To remove a server,** tap its trash button (**Remove host**). VisorMon asks **Remove host?**; tap **Remove**, or **Cancel** to keep it. The server goes, along with the token secret, password and session saved with it. Removing the server you are using signs you out, so for that one the dialog says so and the button reads **Remove and sign out**. See Security and privacy for what else removing a server deletes.
- **To add a server,** tap **Add a server**. The sign-in screen opens for the new server, and your current server stays saved: **Back to** and its name returns you to it.

When no server is saved, for example if you went straight to the sample data, the list reads **No saved hosts**. For signing in, adding a server and changing a saved one, see Getting started.

## 9.3 Read only and Data refresh

### Read only

**Read only** (**Block changes from this app. The server is unchanged.**) stops VisorMon from sending any change to Proxmox. Use it when you only want to look. It is off until you turn it on, it stays on until you turn it off, and it covers every server on this phone.

While it is on, the main tabs show **Read only. Changes from this app are blocked.** at the top, and anything that would change a server stops with **Read only is on. Turn it off in Settings to make this change.** Opening a console or the node shell is blocked too. For exactly what it covers, see Security and privacy.

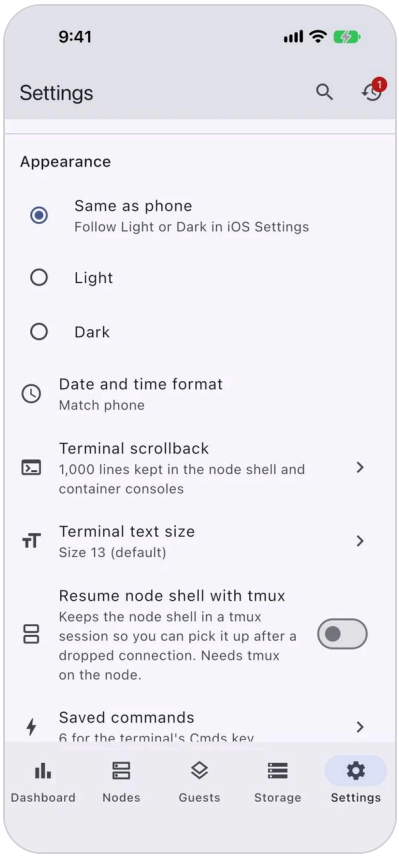
### Data refresh

Choose how often VisorMon fetches new figures on its own. The setting drives the dashboard, the guests dashboard and the running-task count on the **Cluster tasks** button. Other screens load when you open them.

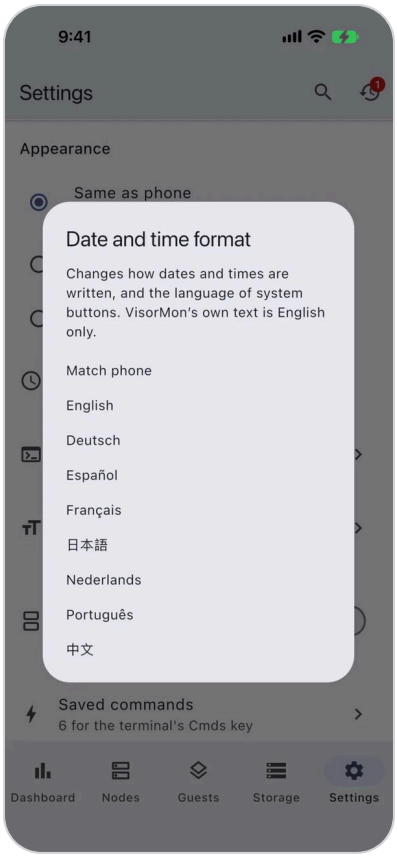
| Choice | The app says |
|---|---|
| **Real-Time** | Refresh about every 5 seconds. |
| **Balanced** (default) | Refresh about every 30 seconds. Recommended. |
| **Battery Saver** | Refresh about every 5 minutes. |

Whatever you choose, you can pull the dashboard down to refresh it at once. A faster choice sends more requests to your server.

## 9.4 Appearance and date format




Appearance and the rows below it.




Choosing a date and time format.

Under **Appearance**, choose how VisorMon looks:

- **Same as phone** (default) follows Light or Dark in your phone's settings, and changes when the phone does.

- **Light** or **Dark** keeps that look whatever the phone is set to.

### Date and time format

**Date and time format** sets how dates and times are written, and the language of the system's own buttons and menus. VisorMon's own text is in English whatever you choose. Tap the row and pick **Match phone** (default), English, Deutsch, Español, Français, 日本語, Nederlands, Português or 中文.

## 9.5 Terminal settings

Four rows set up the text terminals: the node shell (see Nodes) and container consoles (see Working with a guest). The console of a virtual machine (VM) is a picture of its screen, so of these settings only the console lock applies to it. While any console is connected, VisorMon keeps the phone's screen on; this has no setting.

### Terminal scrollback

How many lines of output a terminal keeps once they have scrolled off the screen. The row shows the current value, for example **1,000 lines kept in the node shell and container consoles**.

1. Tap **Terminal scrollback**.
2. Choose **1,000 lines (default)**, **5,000 lines**, **10,000 lines** or **50,000 lines**. A tick marks the current one.

A longer history uses more memory. The change applies at once, including to a terminal that is open. Each terminal shows its history length in a strip at the top, for example **History: 1,000 lines**; its **Change in Settings** button opens Settings at this row.

When you type a package upgrade command in a terminal that keeps fewer than 10,000 lines, VisorMon offers a longer history first. Choosing the option that ends **and make it my setting** changes this setting to 10,000 lines.

### Terminal text size

The size of the text in terminals, from 9 to 24. The default is 13, shown as **Size 13 (default)**.

1. Tap **Terminal text size**.
2. Use the **Smaller text** and **Larger text** buttons. The sample line below them changes as you go.
3. Tap **Done**. **Reset** goes back to 13.

A bigger size fits fewer columns on the screen. The size applies at once to open terminals. You can also change it from a terminal's **More** menu, under **Text size**.

### Resume node shell with tmux

Off by default. When it is on, VisorMon starts the node shell inside a tmux session named `visormon`, so a dropped connection does not end what is running: open the shell again and you are back in the same session. It needs tmux installed on the node. If tmux is missing, the shell says **tmux is not installed on this node, so this shell ends when the connection drops.** and carries on as a normal shell.

The setting applies to the node shell only, not to container consoles. Inside tmux, the history strip reads **tmux keeps this session running if the connection drops. Scroll back with Ctrl-b, then [.** For more on the shell, see Nodes; for problems, see Troubleshooting.

### Saved commands

Commands you can type with one tap from a terminal's **Cmds** key. The row shows how many you have, for example **6 for the terminal's Cmds key**. Until you change the list, it holds six suggestions: `apt update`, `journalctl -f`, `df -h`, `free -h`, `qm list` and `pct list`.

**To add a command:**

1. Tap **Saved commands**, then the **Add a command** button (+) at the top right.
2. Type the **Command**. It must be one line, up to 500 characters.
3. Optionally fill in **Name (optional)**; the list then shows the name above the command.
4. Leave **Run it right away** on to run the command when you tap it. Turn it off to have the command typed and left at the prompt for you to finish, for example `tail -f` followed by a file name.
5. Tap **Save**.

- **To change or delete a command,** tap it. The dialog has **Delete** at the bottom.
- **To reorder,** drag a command by the handle on its right.
- **If the list is empty, Add some suggestions** puts the six suggestions back.

In a terminal, the **Cmds** key opens the list. An arrow marks a command that runs when tapped; a pencil marks one that is typed and waits. **Edit list** opens this screen.

## 9.6 The console lock and Face ID lock

Two switches ask the phone to confirm it is you before something opens. The phone does the check with Face ID, Touch ID, a fingerprint or face, or its passcode; VisorMon only learns whether it passed.

| | |
|---|---|
| **Face ID for consoles** | Off by default. The row says **Ask before the node shell or a guest console opens**. Every console then asks first: the node shell, container consoles and VM consoles. Turning this switch on or off also asks, so someone holding your unlocked phone cannot simply switch it off. If the phone does not confirm, VisorMon says so, for example **The console lock stays on: this phone did not confirm it.** |

| | |
|---|---|
| **Face ID lock** | Off by default. The row says **Require biometrics to open the app**. When VisorMon starts, it shows **Unlock VisorMon** and asks before anything else appears. The lock takes effect as soon as you turn it on. Turning it off does not ask. |

On Android the switches are called **Biometric lock for consoles** and **Biometric lock**, and they appear only when a fingerprint or face is set up on the phone. On an iPhone they always appear and keep their Face ID names, also on a phone with Touch ID.

> **Know the limits.** The Face ID lock asks when VisorMon starts, not when you come back to it while it is still running, and its lock screen offers **Continue without Face ID** when the check fails or is canceled. Read how both locks behave in Security and privacy before you rely on them.

## 9.7 Notifications

**Enable notifications** is on by default. VisorMon sends one kind of notification: **Could not refresh**, **Showing the last saved dashboard.** It comes when the dashboard's refresh fails, for example because the server went offline or your phone left the network, and only once until a refresh works again.

The notification comes from the app on your phone; there is no push service. The check is the dashboard's own refresh, so VisorMon has to be running for it. VisorMon does not notify you when a task finishes or a guest stops. For alerts inside the app, see Search, tasks and alerts.

On iPhone, VisorMon asks for permission to send notifications the first time it starts. On Android, it asks when you turn the switch on; because the switch starts on, turn it off and on again if notifications never appear. If you declined, allow notifications for VisorMon in the phone's own settings. For what the dashboard shows when a refresh fails, see Troubleshooting.

## 9.8 Usage statistics and crash reports

**Share usage statistics and crash reports** (**Anonymous. Never includes your servers, addresses, or credentials.**) lets VisorMon send anonymous figures about which screens and features are used, whether connections work, and reports when the app crashes.

- In most regions it is on until you turn it off.
- If your phone's region is in the European Union, the wider European Economic Area, the United Kingdom or Switzerland, VisorMon asks once, on its first screen: **Help improve VisorMon?** Nothing is sent unless you tap **Allow**.

Turn the switch off at any time to stop sending. For exactly what is and is not sent, see Security and privacy.

## 9.9 Audit log

**Audit log** lists the changes VisorMon has sent to your servers from this phone, the changes read-only mode blocked, and each time read-only mode was turned on or off. The row shows how many entries there are, or **No writes from this app yet**.

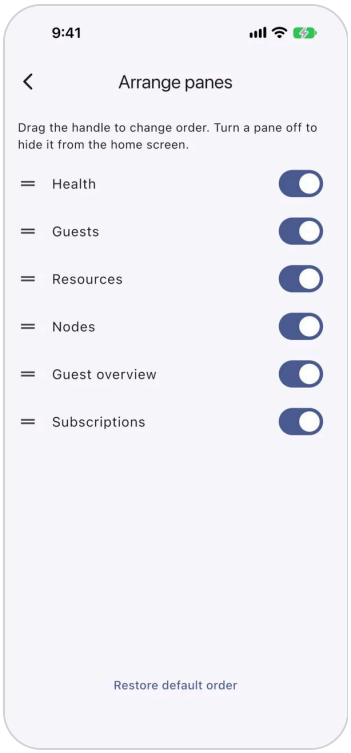
1. Tap **Audit log**. The newest entry is at the top, with its date and time.
2. To empty the log, tap **Clear** at the top right. It clears at once, without asking, and cannot be undone.

The log keeps the latest 200 entries and stays on this phone. For what each entry means and what the log does not record, see Security and privacy.

## 9.10 Arrange panes

**Arrange panes** (**Hide or reorder the home screen cards**) sets which panes the dashboard shows and in what order.

1. Tap **Arrange panes**.
2. To move a pane, hold the handle on its left and drag it up or down.
3. To hide a pane, turn its switch off. Turn it on to show the pane again.
4. To go back to the built-in layout, tap **Restore default order**. This also shows every hidden pane again.



Hiding and reordering dashboard panes.

Changes save as you make them. The panes, in their default order:

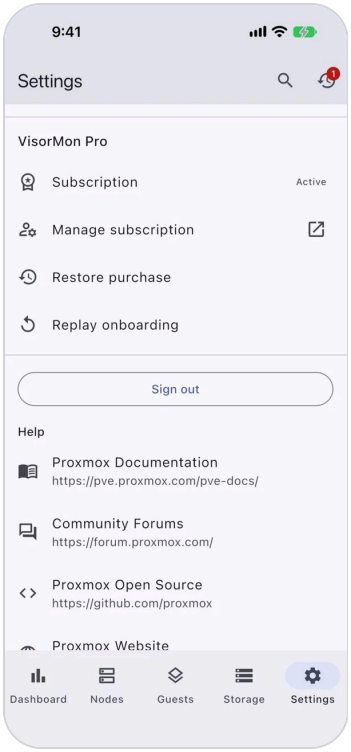
| Pane | On the dashboard |
|---|---|
| **Health** | The cluster, its quorum, how many nodes are online and offline, and Ceph's health |
| **Guests** | How many VMs and containers are running, paused, stopped or templates |
| **Resources** | CPU, memory and storage use across the cluster |
| **Nodes** | Each node and how it is doing |
| **Guest overview** | Cards for the guests you choose, under a second **Guests** heading |
| **Subscriptions** | Your Proxmox subscription status |

The order is the same for every server. Which guests the guest overview shows, and its pins, are set on the dashboard; see Guests. For the panes themselves, see The dashboard.

## 9.11 VisorMon Pro

| | |
|---|---|
| **Subscription** | Your status: **Free trial**, **Active** or **Not subscribed**. |
| **Manage subscription** | Shown while you have a trial or Pro. Opens your subscriptions in the App Store or Google Play, where you can change or cancel. |
| **Restore purchase** | Looks for a subscription bought with the Apple Account or Google account on this phone. If there is none, the row says so, for example **No subscription found for this Apple Account.** |
| **Replay onboarding** | Shows the welcome tour again. |

For plans, the trial and purchases, see VisorMon Pro.



VisorMon Pro and Sign out.

## 9.12 Signing out

1. Tap **Sign out**, below VisorMon Pro.
2. The dialog says **Disconnects this app from Proxmox. Saved credentials stay on this device.** Tap **Sign out**.

VisorMon disconnects, deletes the session from the phone, and opens the sign-in screen. The server stays under **Saved hosts** with its address, user name, and token secret or password, so signing in again is quick. To delete a server's saved details instead, remove it from Saved hosts.

## 9.13 Help, version and licenses

Under **Help**, four links open Proxmox's own resources in your browser:

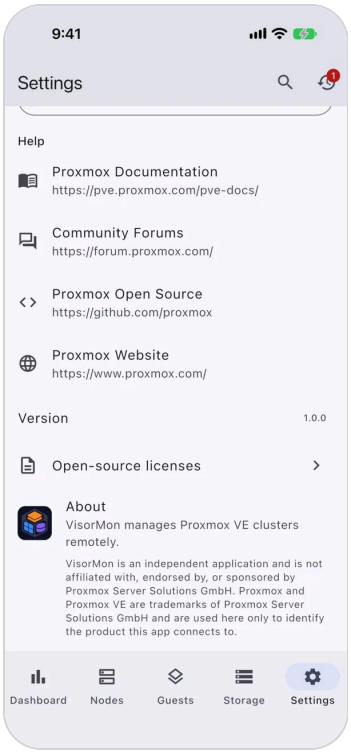
- **Proxmox Documentation**: https://pve.proxmox.com/pve-docs/
- **Community Forums**: https://forum.proxmox.com/
- **Proxmox Open Source**: https://github.com/proxmox
- **Proxmox Website**: https://www.proxmox.com/

These are for Proxmox itself. For help with VisorMon, see Troubleshooting, which also has the support address.

| | |
|---|---|
| **Version** | VisorMon's version number. Give it when you contact support. |
| **Open-source licenses** | The open-source software built into VisorMon, including xterm.js for terminals and noVNC for VM screens, with each one's license. |
| **About** | **VisorMon manages Proxmox VE clusters remotely.** Below it, a note that VisorMon is independent of Proxmox Server Solutions GmbH, which owns the |

Proxmox trademarks.

## 9.14 All settings and their defaults




The bottom of Settings.

| Setting | Choices | Default |
|---|---|---|
| **Read only** | On or off | Off |
| **Data refresh** | Real-Time, Balanced, Battery Saver | Balanced |
| **Appearance** | Same as phone, Light, Dark | Same as phone |
| **Date and time format** | Match phone, English, Deutsch, Español, Français, 日本語, Nederlands, Português, 中文 | Match phone |
| **Terminal scrollback** | 1,000, 5,000, 10,000 or 50,000 lines | 1,000 lines |
| **Terminal text size** | 9 to 24 | 13 |
| **Resume node shell with tmux** | On or off | Off |
| **Saved commands** | Your own list | Six suggestions |
| **Face ID for consoles** | On or off | Off |
| **Face ID lock** | On or off | Off |
| **Enable notifications** | On or off | On |

| Setting | Choices | Default |
| --- | --- | --- |
| **Share usage statistics and crash reports** | On or off | On; in the European Union, the European Economic Area, the United Kingdom and Switzerland, off until you answer the prompt |
| **Arrange panes** | Order, and show or hide each pane | Health, Guests, Resources, Nodes, Guest overview, Subscriptions, all shown |

CHAPTER 10

# Security and privacy

How VisorMon protects your servers and your data: credentials, read-only mode, the audit log and locks.

## 10.1 At a glance

VisorMon can power off a node or delete a virtual machine (VM), so it is built to keep both your servers and your sign-in details safe. In short:

- **Direct connection.** VisorMon connects straight from your phone to your Proxmox server. There is no VisorMon account and no VisorMon server in between.
- **Secrets in secure storage.** Token secrets, passwords and sessions are kept in the phone's secure storage and sent only to your server.
- **Read only.** One switch stops the app from changing anything, and the audit log lists what the app sent.
- **Locks.** The phone can be made to confirm it is you before a console opens, and when VisorMon starts.
- **Anonymous statistics.** Usage statistics and crash reports never include your servers, names or credentials, and you can turn them off.

This chapter explains each one, and what it does not do.

## 10.2 How VisorMon connects

VisorMon uses the Proxmox VE application programming interface (API) at the address you sign in with: the same API the Proxmox web interface uses. Requests go from your phone to that address, over your Wi-Fi or mobile data, and through your virtual private network (VPN) if you use one. Nothing passes through a server of ours. In a cluster, requests for other nodes go through the node at that address, as they do in the Proxmox web interface.

Consoles connect to the same address. The terminal and the VM screen run from code built into VisorMon and load nothing from the internet.

Apart from your server, VisorMon connects only to:

- **Firebase** (Google), for usage statistics and crash reports, and only while that setting is on. See Usage statistics and crash reports.
- **Superwall**, on iPhone, which shows the VisorMon Pro offer, and only while the same setting is on.

- **The App Store or Google Play**, to buy or restore VisorMon Pro. VisorMon never sees your payment details.
- **Web pages you open** from the app, such as the Help links in Settings, which open in your browser.

## 10.3 Your sign-in details

VisorMon keeps what it needs to sign you in again, split by how sensitive it is. Secrets go in the phone's secure storage: the Keychain on iPhone, and encrypted storage protected by the Android Keystore on Android. The rest is kept with the app's other settings.

| What | Where it is kept | How long |
|---|---|---|
| API token secret | Secure storage | Until you remove the server |
| Password, for password sign-ins | Secure storage, so **Login with Face ID** can sign you in again | Until you remove the server |
| The live session: the API token as sent to Proxmox, or the Proxmox login ticket and its cross-site request forgery (CSRF) prevention token | Secure storage | Until you sign out, switch servers, or Proxmox ends the session |
| Server address, user name, realm, token ID, sign-in method, and the **Ignore SSL certificate errors** choice | App settings on the phone | Until you remove the server |

VisorMon sends your token or session only to the server address you entered, and never to the services listed above. The password is sent only when you sign in.

On Android the sign-in button reads **Login with biometrics**, as it does on an iPhone with Touch ID.

### Signing out and removing a server

- **Sign out** disconnects VisorMon and deletes the session from the phone. The server stays saved with its token secret or password, so you can sign in again quickly.
- **Removing a server** takes it off the list and deletes the token secret, password and session saved with it, and its details on the sign-in form. When no other saved server has the same address, VisorMon also forgets the consent notice you accepted there and the guests you pinned for it. VisorMon asks first, in **Settings › Saved hosts** and on the sign-in screen: **Remove host?** If it is the server you are signed in to, the button reads **Remove and sign out**.
- **When Proxmox ends a session**, for example when a password sign-in's ticket expires, VisorMon deletes the session and shows **Login expired**.

For the steps, see Settings and Getting started. If you lose your phone, or want a token never to work again, delete the token in the Proxmox web interface under **Datacenter › Permissions › API Tokens**; for a password sign-in, change the password.

### Choosing an account for VisorMon

VisorMon can do whatever the account you sign in with may do in Proxmox, and no more. It reads that account's permissions when you sign in, and grays out or hides what they do not allow. For the least risk, sign in with an API token that has only the privileges you need. A look-only token, together with read-only mode, means nothing can be changed by accident. Note that the node shell and container consoles need a password sign-in; see Troubleshooting.

## 10.4 Certificates

Proxmox uses HTTPS (secure web connections) on port 8006. By default, VisorMon checks your server's certificate before it sends a request to the API, the way your phone checks any secure website. A new Proxmox install uses a self-signed certificate, which that check refuses, with the message **The secure connection to the server failed. If it uses a self-signed certificate, turn on “Ignore SSL certificate errors” and try again.**

You have two choices:

- **Give the server a certificate your phone trusts.** This is the better choice. Proxmox can get one from Let's Encrypt with its built-in Automatic Certificate Management Environment (ACME) support, set up in the Proxmox web interface under the node's **System › Certificates**.
- **Turn on Ignore SSL certificate errors** on the sign-in screen (**Use only for self-signed homelab certs**). VisorMon then accepts any certificate at that address. The connection is still encrypted, but VisorMon can no longer tell your server from an impostor, so anyone able to intercept your traffic could pretend to be the server and capture your token or password. Use it only on networks you trust, such as your home network or your own VPN.

The choice is saved with each server. To change it for a saved server, sign out, tap the server under **Saved hosts** on the sign-in screen, change the switch, and sign in.

VisorMon adds `https://` to an address that has none. If you type an address that starts with `http://`, VisorMon uses it as typed, and nothing sent to it is encrypted.

## 10.5 Read-only mode

Read-only mode stops VisorMon from changing anything on your servers. Turn it on in **Settings › Read only** when you only want to look, or when you hand your phone to someone. It is one switch for every server on this phone, and it stays on until you turn it off.

VisorMon checks each request before it leaves the phone. Any request that would change something is stopped and never sent, and you see **Read only is on. Turn it off in Settings to make this change.** That covers everything VisorMon can change, for example:

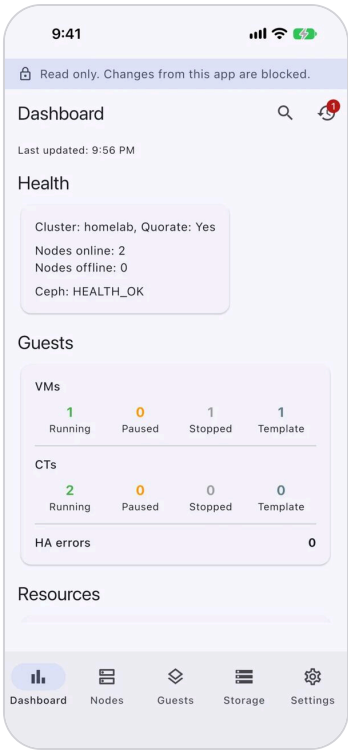
- Power actions on guests and nodes, including starting or stopping every guest on a node.

- Snapshots, backups and restores, clones, migrations, and creating or converting guests.
- Saving hardware, options, notes, tags, firewall rules and node settings.
- Starting and stopping services, checking a node for updates, and applying network changes.
- Opening a console or the node shell, which needs a request of the same kind. With read-only mode on, consoles do not open.

While it is on, the main tabs show **Read only. Changes from this app are blocked.** at the top. Each stopped request is listed in the audit log as **Blocked write**.

**What read-only mode does not do:**

- It changes nothing on the server. The Proxmox web interface, other apps and other phones work as before.
- It does not reach into a console that is already open. What you type there goes straight to the server, so close open consoles too if nothing should change.
- It does not stop VisorMon reading: lists, figures, logs and task output still load, and you can still sign in.

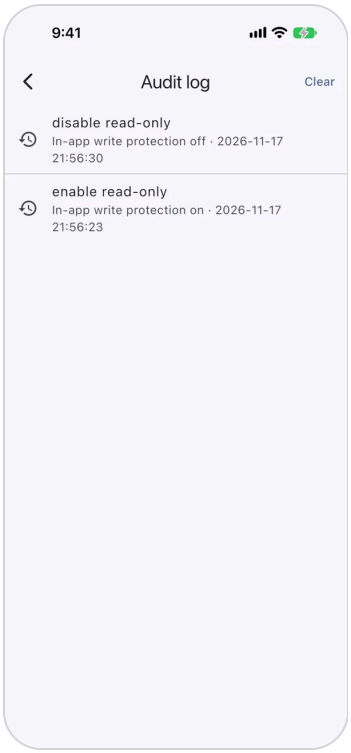



The dashboard with read-only mode on.

> **Two kinds of read-only.** A lock with **Read-only on this VM** on a guest's settings screen is not this switch: it means your Proxmox account may not change that guest. See Troubleshooting.

## 10.6 The audit log

The audit log is VisorMon's own record of what it tried to change from this phone. Open it from **Settings › Audit log**. The newest entry is at the top, with the date and time in your phone's time zone.




The audit log.

| Entry | Detail | Means |
|---|---|---|
| **Write** | The request, for example `POST /api2/json/nodes/pve/qemu/100/status/shutdown` | VisorMon sent this change to the server |
| **Blocked write** | The request, in the same form | Read-only mode stopped it, and nothing was sent |
| **enable read-only** | **In-app write protection on** | Read-only mode was turned on |
| **disable read-only** | **In-app write protection off** | Read-only mode was turned off |

A request reads as a method and a path. `POST`, `PUT` and `DELETE` are the kinds of request that change things; requests that only read are not listed. The path names the node and the guest's VM ID, as in `nodes/pve/qemu/100` for VM 100 on node pve.

- **It shows what was sent, not whether it worked.** An entry is made as the request leaves the phone, before Proxmox answers. Check **Cluster tasks**, or the screen you used, for the result.
- **It covers all your servers together.** Entries do not name the server.
- **It does not record** what you type in a console or the node shell, signing in, or changes made in the Proxmox web interface or other tools. For a record kept by the server, use Proxmox's own task history and system log.

The log keeps the latest 200 entries. It is stored only on this phone and is never sent anywhere. It stays when you sign out or remove a server. To empty it, tap **Clear** at the top right: it clears at once, without asking, and cannot be undone.

## 10.7 Face ID lock and the console lock

Two switches in Settings ask the phone to confirm it is you. The phone does the check, with Face ID, Touch ID, a fingerprint or face, or the phone's passcode. VisorMon never sees your face or fingerprint; it only learns whether the check passed.

| | **Face ID for consoles** | **Face ID lock** |
|---|---|---|
| Asks | Each time the node shell or a guest console opens, before anything connects | When VisorMon starts |
| Covers | The node shell, container consoles and VM consoles | The whole app |
| Turning it off | Asks first | Does not ask |
| If the check fails or you cancel | The console stays closed: **The console stays closed until this phone confirms it is you.** Tap **Unlock** to try again. | A message saying why, for example **Face ID was canceled. Tap Unlock to try again.** (see Troubleshooting), and **Continue without Face ID**, which opens the app |
| Phone with no passcode and no biometrics | The lock cannot ask, so it offers **Open without unlocking** | As above |

On Android the switches are **Biometric lock for consoles** and **Biometric lock**, and the button is **Continue without biometrics**. They appear only when a fingerprint or face is set up on the phone.

**What the locks do not cover:**

- The Face ID lock asks when VisorMon starts. If you switch to another app and come back while VisorMon is still running, it does not ask again.
- Because of **Continue without Face ID**, the Face ID lock keeps VisorMon private at a glance, but does not stop someone holding your unlocked phone from opening it.
- The console lock protects consoles only. Power actions and other changes are not locked; use read-only mode for those.

Your phone's own passcode is what keeps someone else from using it. The console lock adds a second check where it matters most: consoles, where what you type runs straight on your servers.

# 10.8 Usage statistics and crash reports

With **Settings › Share usage statistics and crash reports** on, VisorMon sends anonymous figures to Firebase Analytics and Firebase Crashlytics (Google) so problems get fixed and the app improves. Each statistic is a count, a fixed label or a yes or no, never something your cluster returned or you typed.

## What is sent

- Which screens you open, by a fixed name such as the dashboard or a node's services.
- Features that worked, by a fixed label, for example a snapshot created or a console connected.
- Guest power actions: which action, whether on a VM or a container, and whether it worked, failed, was blocked by read-only mode or was tried on sample data.
- Sign-ins (API token or password), starting the demo, and failed connections with a fixed reason such as unreachable, certificate, timeout, sign-in or permission.
- Steps of the welcome tour, when the subscription offer is shown, and whether a purchase or restore worked.
- Three facts about your setup: whether you use a real server, the sample data or neither; whether you have a trial, Pro or neither; and roughly how many guests you have (none, 1 to 5, 6 to 20, 21 to 50, or more than 50).
- Crash reports: the kind of error and where in VisorMon's code it happened. Not the error's message, because a message can quote a server address.

Firebase adds the basic details it records for any app, such as the phone model, the system version and VisorMon's version.

## What is never sent

- Server addresses or host names.
- User names, token IDs, token secrets or passwords.
- Names of nodes, guests or storage, or anything else Proxmox returns.
- Anything you type, including commands in a terminal.
- An advertising identifier. VisorMon does not use one, so it never asks to track you.

## Your choice

In most regions the setting starts on. If the region set on your phone is in the European Union, the wider European Economic Area, the United Kingdom or Switzerland, nothing is sent until you answer **Help improve VisorMon?** on the first screen with **Allow** or **Don’t allow**.

Turn the switch off at any time. Usage statistics stop straight away, and crash reports not yet sent are deleted rather than kept for later. On iPhone, Superwall stops showing the subscription offer at once and stops entirely the next time VisorMon starts; VisorMon shows its own built-in offer instead, and you can

still subscribe.

## 10.9 What stays on your phone

VisorMon has no account and no server of its own, so what it remembers is kept on your phone:

- Your saved servers and their sign-in details (see Your sign-in details).
- Your settings, including read-only mode, the locks and the terminal settings.
- The dashboard's pane order and hidden panes, the guest overview's order, and pinned guests. Pinned guests are kept for each server address, and forgotten when you remove the last saved server with that address.
- How the Guests tab is grouped.
- Your saved commands.
- The audit log.
- That you accepted a cluster's consent notice, for each server address, until you remove the last saved server with that address.
- The time of the dashboard's last successful refresh, shown in **Showing saved data from**, with the name of one of your nodes.
- When your trial or subscription runs to, so the app can open without waiting for the store.

Everything else VisorMon shows, such as figures, lists, logs and task output, is fetched from your server while you use the app and is not saved on the phone. The one exception is **Share log** in a terminal, which writes the output to a temporary file for the share sheet; VisorMon deletes those files once they are an hour old, the next time you share a log.

VisorMon does not copy any of this anywhere. Your phone's own backups may include it, as they do other apps' data.

## 10.10 Privacy policy

The VisorMon privacy policy sets out what is collected, by whom, and your rights: https://jimmyfatmasterson.com/privacy.html#visormon

Questions about privacy can go to support@jimmyfatmasterson.com.

CHAPTER 11

# VisorMon Pro

What is free, what Pro adds, and managing your subscription.

## 11.1 What needs VisorMon Pro

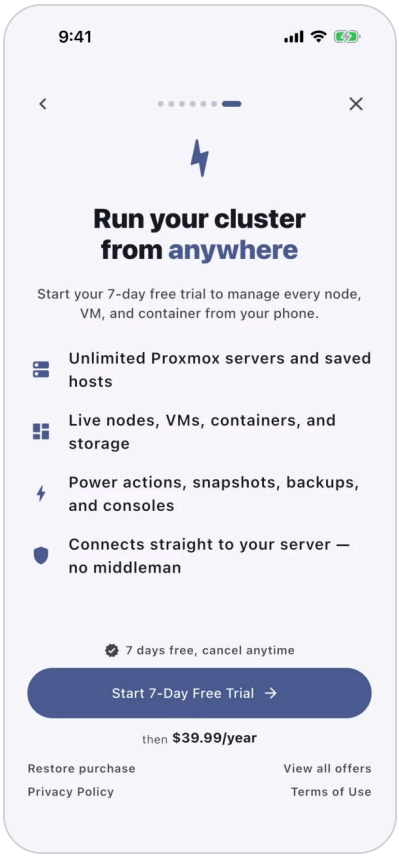
VisorMon Pro is VisorMon's subscription. You need Pro, or its 7-day free trial, to use VisorMon with your own Proxmox servers. You can try the app before you pay: the tour, the sign-in screen and the whole demo are free.

| | Without a trial or Pro | With a trial or Pro |
|---|---|---|
| The tour, the sign-in screen and **Test connection** | Yes | Yes |
| The demo, with every screen on sample data | Yes | Yes |
| Signing in to your own server | Yes, but the paywall opens instead of your server | Yes |
| Using your own servers: the dashboard, nodes, guests, storage, consoles and every other screen | No | Yes |
| How many servers | None (only the demo) | As many as you like |

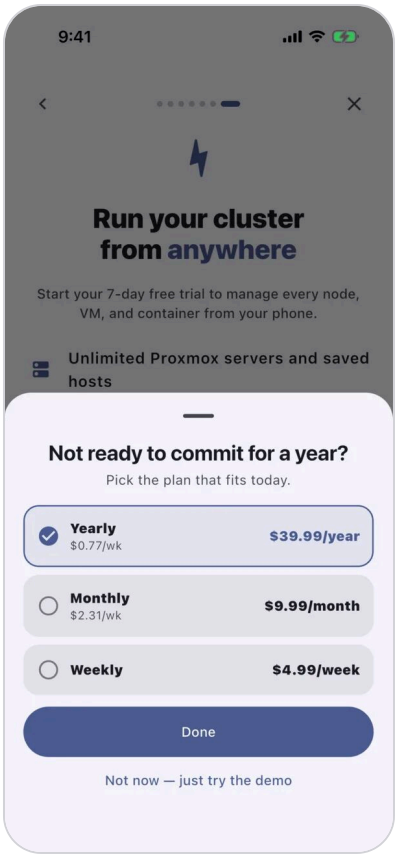
Nothing else is held back. With a trial or Pro, every screen and action in VisorMon is available. What you can do on a server still depends on your Proxmox permissions, and on read-only mode if you turn it on (see Security and privacy).

> **Not a Proxmox subscription.** The **Subscriptions** pane on the dashboard shows the Proxmox VE subscription of your nodes, which you buy from Proxmox. It has nothing to do with VisorMon Pro (see The dashboard).

# 11.2 The free trial and the plans




The paywall at the end of the tour.




Choosing a plan.

## The free trial

New subscribers can try VisorMon Pro free for 7 days. The App Store or Google Play decides whether you can have the trial, and the paywall changes to match:

| | When you can have the trial | When you can't (for example, you have had it) |
|---|---|---|
| The paywall says | **7 days free, cancel anytime** | **No commitment, cancel anytime** |
| The button | **Start 7-Day Free Trial** | **Subscribe Now** |
| Below the button | **then** and the plan's price | The plan's price |
| Billing | Starts when the 7 days are up, unless you cancel first | Starts straight away |

When the trial ends, the store charges you for the plan you chose and renews it from then on, until you cancel (see Checking, changing or canceling).

### The plans

There are three plans: **Yearly**, **Monthly** and **Weekly**. They all include the same thing, everything in VisorMon for every server, and differ only in how often you pay. The paywall starts with **Yearly** selected. The App Store and Google Play set the prices in your currency; the paywall shows the amount you are billed, such as the price per year, and the store shows the final price again before you confirm. In **View all offers**, **Yearly** and **Monthly** also show what they work out to per week, in small type.

## 11.3 Subscribing

VisorMon offers Pro in two places:

- **The last page of the tour**, the first time you open the app (or after **Settings** › **Replay onboarding**).
- **After you sign in to a server**, when you have no trial or subscription. The paywall opens instead of your server.

1. To choose a plan other than **Yearly**, tap **View all offers**. In **Not ready to commit for a year?**, tap a plan, then tap **Done**.
2. Tap **Start 7-Day Free Trial**, or **Subscribe Now**.
3. Confirm the purchase with the App Store or Google Play.

As soon as the store confirms it, the tour ends, or your server opens. The subscription belongs to your Apple Account or Google account, and it covers every server you add in VisorMon.

If the store's plans have not loaded, for example because the phone is offline, VisorMon says **Subscriptions are unavailable right now. Check your network and try again.**

> **On iPhone** the paywall can look different from the one pictured here. Whatever it looks like, the purchase is made through the App Store.

### Finishing the tour without subscribing

On the last page of the tour, tap ✕ at the top right, or **View all offers**, then **Not now — just try the demo**. If VisorMon shows a different paywall on your iPhone, close it. The tour ends and you can try the demo, or sign in and subscribe later.

## 11.4 Restoring a purchase

Your subscription belongs to your Apple Account or Google account, not to the phone or to a server. On a new phone, or after reinstalling VisorMon, restore it:

1. Make sure the phone is signed in to the App Store or Google Play with the account you subscribed with.

2. Tap **Restore purchase**. It is on the paywall, and in **Settings › VisorMon Pro**.

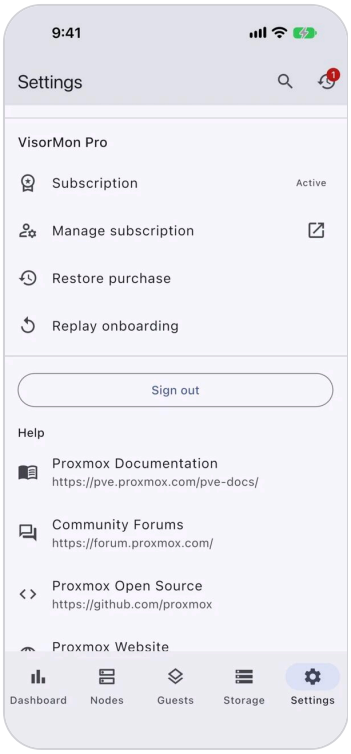
If the store finds your subscription, Pro is back straight away. If not, VisorMon says **No subscription found for this Apple Account.** or **No subscription found for this Google account.** Check that you are using the right account.

A subscription bought on iPhone does not unlock VisorMon on Android, or the other way round, because each is sold by its own store.

## 11.5 Checking, changing or canceling

Go to **Settings › VisorMon Pro**:

| | |
|---|---|
| **Subscription** | Where you stand: **Free trial**, **Active** or **Not subscribed**. On Android, a trial shows as **Active**. |
| **Manage subscription** | Opens your subscriptions in the App Store or Google Play, where you can change plan or cancel. Shown only while you have a trial or Pro. |
| **Restore purchase** | Asks the store for your subscription again (see Restoring a purchase). If it finds none, the reason appears below it. |
| **Replay onboarding** | Shows the tour again, ending with the paywall if you have no trial or Pro. |



Settings › VisorMon Pro.

You can also manage the subscription from the phone itself: on iPhone, in **Settings** › your name › **Subscriptions**; on Android, in the Google Play Store app, under your profile picture › **Payments & subscriptions** › **Subscriptions**.

### Canceling

Cancel in the App Store or Google Play, as above. Deleting VisorMon does not cancel the subscription.

- To avoid being charged when the free trial ends, cancel before it ends. Apple asks you to cancel at least 24 hours before a trial or a paid period ends.
- After you cancel, you keep Pro until the end of the trial or the period you have paid for.

## 11.6 When the trial or subscription ends

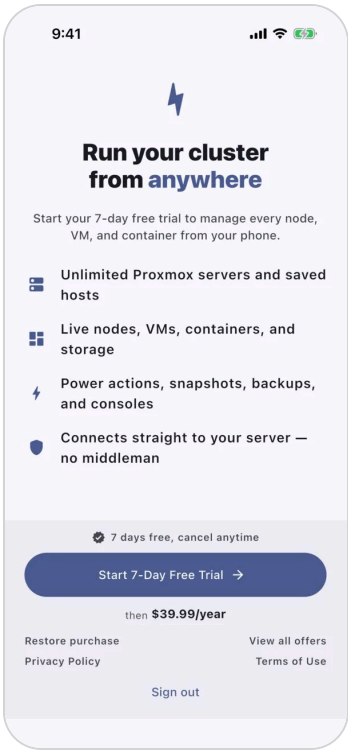
The next time you open VisorMon with one of your servers, it may show **Checking your subscription…** for a moment while it asks the store. Then it shows the paywall instead of your server. If you have had the trial, the button reads **Subscribe Now**, and VisorMon's own paywall adds **Thanks for trying it out — we'd love to have you back!**

The paywall has **Restore purchase**, **View all offers** and, at the bottom, **Sign out**. If the plans open over another page instead and you close them, that page stays, with the heading **Subscribe to keep going** and the line **VisorMon needs a subscription to connect to your server. The demo stays free.** Either way you can:

- subscribe again (on that page, tap **See plans**),
- tap **Restore purchase** if you have a subscription on this store account, perhaps bought on another phone, or
- tap **Sign out**, then **Try demo** on the sign-in screen to use the demo.

Nothing on your servers changes, and VisorMon keeps your saved servers and settings. Subscribe again and everything is as you left it.

**Offline.** When the phone cannot reach the store, VisorMon trusts the last answer it had for a while: on iPhone until the date your trial or subscription was due to renew, and on Android for three days after Google Play last confirmed it.




The paywall after signing in, for someone who has not had the trial.

## 11.7 The demo and VisorMon Pro

The demo is always free. It never shows the paywall, and every screen works on its sample cluster, so you can see exactly how VisorMon works before you subscribe. See Getting started for how to start and leave it.

- Using the demo does not start or use up your free trial. The trial starts only when you tap **Start 7-Day Free Trial** and confirm with the store.
- To subscribe from the demo, tap **Settings › Replay onboarding** and go to the last page, or leave the demo and sign in to your server.
- When a trial or subscription ends, the demo keeps working.

## 11.8 Questions

**Do I need a subscription for each server?**

No. One subscription covers every server you add in VisorMon.

**I subscribed, but VisorMon still shows the paywall.**

Tap **Restore purchase**, and check that the phone is signed in to the App Store or Google Play with the account you subscribed with. If the phone was offline, connect it and try again.

**Can I use VisorMon on iPhone and Android with one subscription?**

No. A subscription works on the platform where you bought it. On another phone of the same kind, signed in to the same store account, tap **Restore purchase**.

**What happens to my servers if I cancel?**

Nothing. VisorMon never changes anything on a server because of your subscription. When Pro ends, VisorMon shows the paywall instead of your servers, and keeps your saved servers on the phone.

CHAPTER 12

# Troubleshooting

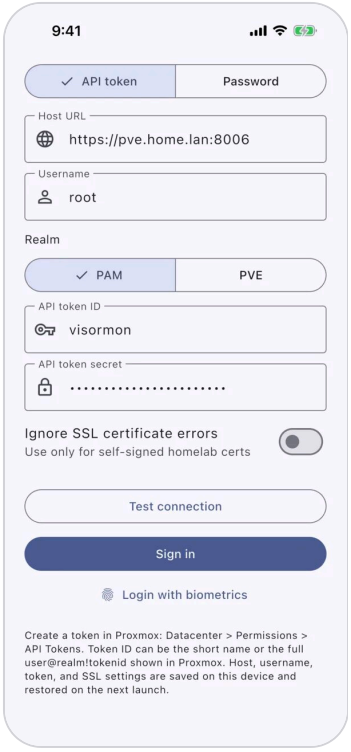
What error messages mean and how to fix common problems.

## 12.1 Start here

Most problems come down to the address, the network, the certificate or permissions. When VisorMon cannot connect, work through these steps before anything else.

1. **Check the address.** It needs the port, for example `https://192.168.1.30:8006`. VisorMon adds `https://` when you leave it out, but it does not add a port.
2. **Check the phone can reach the server.** Open the same address in the phone's browser. If the Proxmox login page does not appear there either, the problem is the network path, not VisorMon: join the same Wi-Fi as the server, or connect your virtual private network (VPN).
3. **Use Test connection.** On the sign-in screen, fill in the form and tap **Test connection**. It tries the address and your details without signing in. When it works, it shows the server's Proxmox VE version; when it does not, it shows one of the messages below.
4. **Look the message up** in Messages when connecting or signing in.

Questions about the guest lists, such as a virtual machine (VM) disk or network speed that shows a dash, pins, or templates missing from the dashboards, are answered in Guests.




Test connection is on the sign-in screen.

## 12.2 Messages when connecting or signing in

Some messages include a Hypertext Transfer Protocol (HTTP) status code: the number a web server answers with, such as 401 or 403.

| Message | What it means | What to do |
|---|---|---|
| **That address is not valid. Use a host name or IP address, with the port, for example https://pve.example.com:8006.** | The address cannot be read as a web address. | Retype it in the form shown, with no spaces. |

| Message | What it means | What to do |
|---|---|---|
| **No server was found at that address. Check the host name for typos.** | The name did not look up to any server. | Check the spelling. A name that only works at home, such as one ending in `.lan` or `.local`, may not work on mobile data or over a VPN; try the server's IP address. |
| **The server refused the connection. Check the port (Proxmox uses 8006) and that the server is running.** | Something is at that address, but nothing accepted the connection on that port. | Add or correct the port. Check that the server is on and that no firewall rejects the phone. |
| **There is no network route to the server. Check that this device is online and on the same network, or connected through a VPN.** | The phone has no way to reach that address from where it is. | Join the server's network or connect your VPN. See Timeouts and no route. |
| **The server did not answer in time. Check the address and port, and that this device can reach it (same Wi-Fi, or a VPN).** | No answer came back before VisorMon gave up waiting. | As above. A firewall that silently drops traffic also causes this. |
| **The server closed the connection. Check the address and port (Proxmox uses 8006).** | The server hung up before answering. | Check the port. A proxy in front of Proxmox can cause this. |
| **This address answered, but not like a Proxmox server. Check the host and port (Proxmox uses 8006), and any reverse proxy in front of it.** | A different service, such as a router page, a web site or a proxy, answered. | Use the Proxmox server's own address and port 8006, or fix the reverse proxy. The same text after **HTTP** and a number means a web page answered with that status. |
| **This port did not answer with HTTPS. Proxmox uses HTTPS on port 8006; check the address and port.** | The address starts with `https://` but the port does not use secure connections. | Use port 8006. |
| **The secure connection to the server failed. If it uses a self-signed certificate, turn on "Ignore SSL certificate errors" and try again.** | VisorMon did not accept the server's certificate. | See Certificate errors. |
| **Could not reach the server. Check the address and port, and that this device can reach it.** | The connection failed for another reason. | Work through Start here. |
| **Authentication failed. Check the username, realm, password, or API token.** | Proxmox rejected the sign-in (HTTP 401). | See Sign-in fails. |
| **Permission denied for this resource. Check the user or API token ACLs.** | You are signed in, but the account may not do this (HTTP 403). | See Grayed-out actions and empty lists. |

| Message | What it means | What to do |
|---|---|---|
| **The node did not answer (HTTP 595). It may be powered off, or the server this app is connected to cannot reach it.** The number can be 595 to 599. | Your server is fine, but another node in the cluster did not answer it. | See A node does not answer. |
| **Two-factor authentication required. Enter a TOTP or recovery code.** | The account uses two-factor authentication, and Proxmox accepted the password. | Type the current time-based one-time password (TOTP) from your authenticator app, or one of your recovery keys (such as `1a2b-3c4d-5e6f-7a8b`), then tap **Verify**. |
| **This account requires WebAuthn, which is not supported yet. Use TOTP or an API token.** | The account's only second factor is a security key or passkey, which VisorMon cannot use. It has no TOTP and no unused recovery keys. | Sign in with an application programming interface (API) token instead, or add TOTP to the account in Proxmox. |
| **This account's second factor cannot be used in VisorMon. Use TOTP or an API token.** | The account's only second factor is a YubiKey one-time password, or it has no second factor left, for example because every recovery key has been used. | Sign in with an API token, or add TOTP to the account in Proxmox. |
| **Proxmox did not return a session ticket.** | Proxmox accepted the request but sent no session back. | Try again. If it persists, check the server in the Proxmox web interface. |
| **Could not sign back in to** a server's name, then **Check the details and sign in.** | After you started adding a server, VisorMon could not sign back in to the one you were using, for example because its password session has ended or it cannot be reached. | Check the details on the form, enter the password if it is a password sign-in, and sign in. |
| **Sign-in did not work. Check the address and your details, then try again.** | An unexpected error. | Check the form and try again. If it keeps happening, contact support. |
| **HTTP** and a number, then text | Proxmox answered with an error; the text is Proxmox's own message. | Read Proxmox's message. It usually names what is wrong. |

## Wrong address or port

Proxmox answers on port 8006, so the address needs `:8006` at the end: `https://pve.example.com:8006` or `https://192.168.1.30:8006`. Without a port, VisorMon connects to port 443, where Proxmox does not answer unless you have a reverse proxy there. If you use a reverse

proxy, enter its address as you would in a browser, and make sure it passes websocket connections through, or consoles will not open.

To correct a saved server's address, sign out, tap the server under **Saved hosts** on the sign-in screen, change **Host URL**, and sign in.

## Certificate errors

A new Proxmox install uses a self-signed certificate, which VisorMon does not accept until you say so. Either give Proxmox a trusted certificate, for example from Let's Encrypt through Proxmox's built-in support, or turn on **Ignore SSL certificate errors** on the sign-in screen for a server on a network you trust. What that switch risks is explained in Security and privacy.

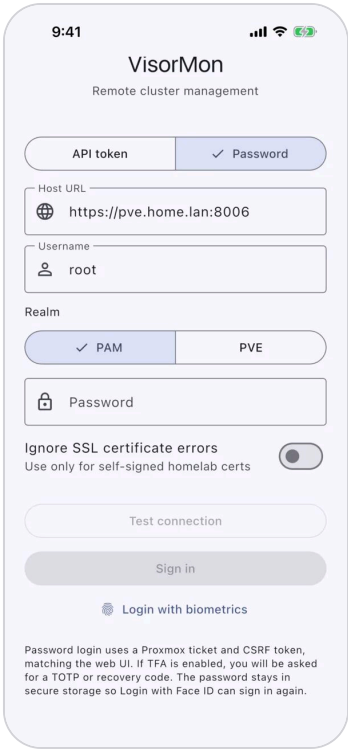
## Sign-in fails

**With an API token:**

- The **Username** and **Realm** must be those of the user who owns the token, for example `root` and **PAM**.
- The **API token ID** can be the short name, such as `visormon`, or the full ID, such as `root@pam!visormon`.
- Proxmox shows a token's secret only once, when you create it. If you no longer have it, create a new token.
- A token stops working if it is deleted, or passes the expiry date set in Proxmox.

**With a password:**

- Choose the realm the user belongs to: **PAM** for system users such as `root`, **PVE** for users created in Proxmox.
- If the account uses two-factor authentication, VisorMon asks for a **TOTP or recovery code** after the password. If **Verify** then says **Authentication failed**, the code was wrong or had expired, or the recovery key had already been used: type the next code and tap **Verify** again. If it keeps failing, check that the phone's clock is set automatically, because TOTP codes depend on the time.




The sign-in form in Password mode.

For creating a token and the sign-in form, see Getting started.

## Timeouts and no route

VisorMon connects straight from the phone to your server, so the phone must be able to reach it. At home, that usually means the same Wi-Fi as the server. Away from home, connect the VPN you use to reach your network first.

On iPhone, VisorMon needs Local Network access to reach a server on your home network. If you tapped **Don't Allow** when iOS asked, turn VisorMon on under **Settings › Privacy & Security › Local Network** in the iPhone's settings.

### A node does not answer (HTTP 595 to 599)

VisorMon talks to one node: the one whose address you signed in with. That node passes requests for the other nodes of the cluster on to them. When one of them is off, or cannot be reached from your node, Proxmox answers with a code from 595 to 599, and VisorMon says **The node did not answer**. Your connection is fine.

- Check that the node is powered on and connected to the cluster's network, for example in the Proxmox web interface.
- The rest of the cluster keeps working in VisorMon. The node stays in the lists, without figures, until it answers again. See The dashboard.
- If the node you signed in through is the one that is down, VisorMon cannot reach the cluster at all. Add another node's address as a second server with **Settings › Add a server**, and use that while the first is down.

## 12.3 Login expired, or VisorMon opens at the sign-in screen

**Login expired: Your session ended. Sign in again to view this data.** Proxmox stopped accepting your session. With a password sign-in, this happens when the Proxmox login ticket expires, two hours after you sign in on a standard Proxmox VE server; VisorMon does not renew it. Sign in again, or tap **Login with Face ID**. With an API token, Proxmox no longer accepts the token, for example because it was deleted or has expired; create a new one in Proxmox.

**VisorMon opens at the sign-in screen with your details filled in, but no message, although you did not sign out.** It could not reach your server when it started. Check your network or VPN, then tap **Sign in**: if the server still cannot be reached, you see the reason. With a password sign-in, type the password again or use **Login with Face ID**.

## 12.4 "Could not refresh" and saved data

When the dashboard's refresh fails, for example because the server went offline or the phone left the network, VisorMon keeps showing the figures it last loaded, so you still have something to look at. It tells you so in three ways:

- A strip at the top of the main tabs: **Showing saved data from** and the date and time of the last refresh that worked.
- A **Needs attention** card below it, reading **Could not refresh**. Tap it to open **Alerts**, which says **Showing the last saved dashboard.**

- With **Enable notifications** on, one notification with the same words. You do not get another until a refresh has worked again.

All of these clear on the next refresh that works. Pull the dashboard down to try now, or wait for the next automatic refresh; see **Data refresh** in Settings.

The figures are the ones loaded while VisorMon was open. VisorMon does not keep a copy of the dashboard after it closes, so if nothing has loaded since you opened it, the dashboard shows the error with **Retry** instead. Other screens do not use saved data: when one cannot load, it shows the error and **Retry**.

## 12.5 Grayed-out actions and empty lists

VisorMon reads your account's permissions from Proxmox when you sign in, and grays out or hides what the account may not do. A grayed-out action says why underneath, for example **No power permission on this VM**.

| The reason mentions | Proxmox privilege needed |
|---|---|
| **power**, on a VM or container | `VM.PowerMgmt` on that guest |
| **power**, on a node | `Sys.PowerMgmt` on the node |
| **guest power**, on a node's **Start all**, **Stop all** and **Suspend all** | `VM.PowerMgmt` on at least one guest |
| **console**, on a VM or container | `VM.Console` |
| **console**, on a node (the shell) | `Sys.Console` |
| **clone** | `VM.Clone` |
| **migrate**, or **guest migrate** on a node's **Migrate all** | `VM.Migrate` |
| **allocate** (convert to template) | `VM.Allocate` |
| **backup** | `VM.Backup` |
| **audit**, on a node | `Sys.Audit` |
| **log**, on a node | `Sys.Syslog` |
| **update**, on a node | `Sys.Modify` |

Other signs that the account lacks permission:

- **Read-only on this VM** or **Read-only on this container** at the top of a guest's settings screen: the account may not change those settings. This is about Proxmox permissions, not VisorMon's read-only mode. See Working with a guest for what each screen needs.
- **No nodes in your scope**, **No guests in your scope** or **No storage in your scope**: the account can see none.

- **No permission**, with a line such as **This area is blank because you don't have permission to view it.**

### API tokens with privilege separation

A common cause is an API token created with **Privilege Separation** ticked. Such a token has only the permissions given to the token itself, never more than its user's, and starts with none. To give it some, in the Proxmox web interface:

1. Go to **Datacenter › Permissions** and choose **Add › API Token Permission**.
2. Choose the path: `/` for the whole cluster, or a narrower one such as `/vms/100` or `/nodes/pve`.
3. Choose the token and a role, such as `PVEAuditor` to look only or `PVEVMAdmin` to manage guests, and add it.

VisorMon reads permissions when you sign in and when it starts, not while it runs. After changing them, close VisorMon and open it again, or sign out and back in.

If VisorMon cannot read the permissions at all, it shows every action, and Proxmox refuses the ones the account may not do with **Permission denied for this resource. Check the user or API token ACLs.**

### Changes refused by VisorMon itself

- **Read only is on. Turn it off in Settings to make this change.** VisorMon's read-only mode is on. Turn off **Settings › Read only**. See Security and privacy.
- **This is sample data. Nothing was sent.** You are in the demo, which never changes anything. Tap **Leave demo** and sign in to your server.

## 12.6 Shell and console problems

**The node shell needs a username and password login. Proxmox does not allow API tokens to open shells.**

Sign out, then sign in with **Password**.

**Container consoles need a username and password login. Proxmox does not allow API tokens to open LXC shells.**

The same: container consoles need a password sign-in.

**The node shell is only available when signed in as root@pam.**

VisorMon adds this line when the node shell does not open. Sign in with a password as `root` in the **PAM** realm. If the line above it says **Read only is on**, the cause is read-only mode instead: turn it off and try again.

**Read only is on. Turn it off in Settings to make this change. when opening a console**

Read-only mode also stops consoles from opening. Turn off **Settings › Read only**.

**The Console or Shell button is grayed out**

The account lacks `VM.Console` on the guest, or `Sys.Console` on the node. See Grayed-out actions and empty lists.

**Console connection timed out, or Console disconnected in a red strip**

The console's connection failed or dropped. Tap **Reconnect** in the strip, or **Retry** if the console never opened; a terminal keeps what it showed above the new session. A console uses a websocket connection to the same address and port as the rest of the app, so if everything else works, check that any reverse proxy in front of Proxmox passes websockets through. To keep a node shell's work alive across drops, use tmux (below).

**Unlock to open, then the console's name**

**Face ID for consoles** is on. Tap **Unlock** and pass the phone's check. If you see **This phone has no passcode or biometrics to ask for**, the phone cannot ask; tap **Open without unlocking**, or turn the console lock off in Settings.

**Sample data has no live console. Sign in to a Proxmox server to open one.**

The demo has no consoles. Sign in to your own server.

**The screen does not turn off**

While a console is connected, VisorMon keeps the phone's screen on so a long job stays in view. The screen can sleep again once you close the console or it disconnects.

### Resume node shell with tmux

- **tmux is not installed on this node, so this shell ends when the connection drops.** Install tmux on the node, for example with `apt install tmux` in the node shell, or turn off **Settings › Resume node shell with tmux**.
- **Getting back to your session.** Open the node shell again; it rejoins the session named `visormon`. Opening it from another phone takes the session over there.
- **Closing.** When you close the terminal, VisorMon says **Your tmux session keeps running on the node. Open the shell again to return to it.** To end the session for good, type `exit` in the shell first.
- **Scrolling back.** Inside tmux, VisorMon's history does not apply. Press Ctrl-b, then [, to scroll in tmux, and q to stop.

## 12.7 Notifications

**I never get notifications.**

VisorMon sends only one: **Could not refresh**, when the dashboard cannot refresh. It comes from the app itself, so VisorMon must be running. Check that **Settings › Enable notifications** is on and that the phone allows notifications for VisorMon in its own settings. On Android, turn the switch off and on again to be asked for permission.

**I got a notification saying "Could not refresh".**

The dashboard could not reach your server. See "Could not refresh" and saved data.

**Can VisorMon tell me when a task finishes or a guest goes down?**

No. Follow tasks in the app; see Search, tasks and alerts.

## 12.8 Locks

**The Face ID lock did not open the app.**

The lock screen says why:

- **Face ID was canceled. Tap Unlock to try again.** You, or the phone, closed the Face ID prompt.
- **Face ID did not recognize you. Tap Unlock to try again.**
- **Face ID is not set up. You can continue, or turn the lock off in Settings.** The phone has no Face ID or passcode to check with. Set one up in the phone's settings, or turn the lock off.
- **Face ID is locked after too many tries. Unlock the device with its passcode, then tap Unlock.**
- **Face ID could not check just now. Tap Unlock to try again.** Something else stopped the check.

Whatever the message, **Continue without Face ID** also opens the app. On Android the messages say **Biometrics** or name the fingerprint or face, a cancel reads **The check was canceled. Tap Unlock to try again.**, and the button is **Continue without biometrics**.

**The console lock stays on: this phone did not confirm it.**

Turning the console lock off needs the phone's check, and it did not pass. Try again; the phone's passcode works too.

How the locks work is explained in Security and privacy.

## 12.9 Other questions

**I don't see the server button in the top bar.**

It appears once at least one server is saved on this phone.

**A VM's disk or a guest's network speed shows a dash, my pins have gone, or templates are missing from the dashboards.**

See Guests, which answers these.

**Restore purchase finds nothing, or a subscription problem.**

See VisorMon Pro.

# 12.10 Getting help

If this chapter has not solved it, email support@jimmyfatmasterson.com. Please include:

- Your phone model, and its iOS or Android version.
- VisorMon's version, from **Settings › Version**.
- Your Proxmox VE version. The Proxmox web interface shows it, and so does **Test connection** on the sign-in screen while you are signed out.
- Whether you sign in with an API token or a password (**Settings › Sign-in method**).
- The exact message you see, ideally as a screenshot, and what you were doing.

> **Keep your secrets.** Never send your token secret or password.

## Sharing a terminal's output

If the problem shows in the node shell or a container console, you can send what the terminal printed:

1. In the terminal, tap the **More** button at the top right and choose **Share log**.
2. VisorMon saves everything the terminal still holds as a text file named after the terminal and the time, such as `pve-shell-2026-10-28-1530.log`, and opens the phone's share sheet.
3. Attach it to your email, or save it to Files first.

The file holds whatever was on the screen, so read it through for passwords or other secrets before you send it. VisorMon deletes its copy once it is an hour old, the next time you share a log. VM consoles are pictures of a screen and have no log to share; take a screenshot instead. The audit log cannot be exported; take a screenshot of it if it helps.

CHAPTER 13

# Quick reference

Where to find things, the colors and states the app uses, and a glossary.

## 13.1 Where to find things

| To | Go to |
|---|---|
| Switch to another server, or add one | The server button at the top of the main tabs, or **Settings › Saved hosts** |
| See whether the cluster is healthy | **Dashboard › Health** |
| Hide or reorder the dashboard's panes | **Settings › Arrange panes** |
| Choose which guests the dashboard shows, or pin one | **Dashboard** › the line under **Guests**, or **Guests** tab › ⋮ › **Pin to dashboard** |
| See every guest's readings | **Dashboard › View all**, or the gauge button on the Guests tab |
| Find a guest by name, node, tag or pool | **Guests** tab: search, **Filter by node or tag**, **Group by** |
| Start, shut down, reboot or stop a guest | **Guests** tab › ⋮, or the guest's page |
| Open a guest's console | The guest's page › **Console**, or **Guests** tab › ⋮ › **Console** |
| Snapshot, back up or restore a guest | The guest's page › **Snapshots** or **Backup** |
| Change a guest's hardware or options | The guest's page › **Hardware** or **Options** |
| Create a VM or container | **Guests** tab › **+** |
| Open a node's shell | **Nodes** › the node › **Shell** |
| Check a node for updates | **Nodes** › the node › **Updates** |
| Reboot or shut down a node | **Nodes** › the node › **Reboot** or **Shutdown** |
| See what is on a storage | **Storage** › the storage |
| Search the whole cluster | **Search cluster**, the magnifier at the top of the main tabs |
| Follow a task, read its log or stop it | **Cluster tasks**, the clock at the top of the main tabs |
| Stop VisorMon changing anything | **Settings › Read only** |
| See what VisorMon has changed | **Settings › Audit log** |
| Ask for Face ID before consoles open | **Settings › Face ID for consoles** |
| Change how often figures update | **Settings › Data refresh** |

| To | Go to |
|---|---|
| Manage, restore or cancel VisorMon Pro | **Settings › VisorMon Pro** |
| See the tour again | **Settings › Replay onboarding** |
| Sign out | **Settings › Sign out** |

## 13.2 Colors and states

| You see | Where | It means |
|---|---|---|
| **online** / **offline** | Nodes | The node is up, or is down or not answering. |
| **running** | Guests | The guest is on. |
| **stopped** | Guests | The guest is off. |
| **paused** | Guests | The guest is paused. Resume it from its menu or page. |
| **template** | Guests tab | A template, used to create new guests. |
| **unknown** | Guests | VisorMon cannot get the guest's state, usually because its node is down. |
| Blue, green, orange bars | Guest cards | CPU, memory and disk, below 90%. |
| A red bar or figure, **critical** | Guest cards | That reading is at 90% or more: the guest is critical. |
| **running** / **OK** / a message in red | Cluster tasks | The task is running, finished without problems, or failed or warned. |
| A green or red dot | Storage | Proxmox has current figures for the storage, or has none (for example, its node is offline). |
| **Read only. Changes from this app are blocked.** | Top of the main tabs | Read-only mode is on. |
| **Showing saved data from …** | Top of the main tabs | The last refresh failed; the dashboard shows what it loaded before. |
| **Sample data** | Top of the screen | You are in the demo. Nothing you do is sent anywhere. |

## 13.3 Glossary

**API token** A key for signing in to Proxmox without your password. It belongs to a user and can be limited to only some permissions. See Getting started.

| | |
|---|---|
| **Audit log** | VisorMon's own list, kept on your phone, of the changes it sent to your servers. |
| **Backup** | A copy of a guest saved to a storage by Proxmox's backup tool, vzdump. A backup can be restored as a new guest. |
| **Ceph** | Storage spread across the nodes of a cluster. The dashboard shows its health when the cluster runs it. |
| **Cloud-Init** | A way to set a VM's user, password, SSH keys and network the first time it starts, without logging in to it. |
| **Cluster** | A group of Proxmox nodes managed together. A single server is a standalone node. |
| **Console** | A guest's screen and keyboard (for a VM) or terminal (for a container), opened from VisorMon. |
| **Container (CT)** | A lightweight Linux system that shares its node's kernel. Proxmox uses LXC containers. |
| **Demo** | VisorMon running on built-in sample data, with no server. It is free. |
| **Guest** | Proxmox's word for a VM or a container. |
| **High availability (HA)** | A Proxmox feature that keeps chosen guests running, starting them on another node if theirs fails. |
| **Migrate** | Move a guest from one node to another. |
| **Node** | One physical server running Proxmox VE. |
| **Pool** | A resource pool: a named group of guests and storage, often used to give someone permissions on just that group. |
| **Privilege** | One permission in Proxmox, such as `VM.PowerMgmt` (power a guest on and off). Privileges are grouped into roles, such as `PVEAuditor`. |
| **Privilege separation** | An API token setting. With it, the token has only the permissions given to the token itself; without it, the token has all of its user's permissions. |
| **QEMU guest agent** | A small program inside a VM that tells Proxmox more about it, such as its disk use and IP addresses. |
| **Quorum** | Enough nodes of a cluster can see each other to make changes safely. The dashboard shows **Quorate: Yes** or **No**. |
| **Read-only mode** | A VisorMon setting that stops the app sending any change to your servers. |
| **Realm** | Where a Proxmox user is defined: **PAM** for Linux accounts on the server, such as `root`, or **PVE** for users created in Proxmox. |
| **Snapshot** | The state of a guest saved at a moment, which you can roll back to. |
| **Storage** | Where Proxmox keeps disks, ISO images, templates and backups, such as a local directory, LVM, ZFS, a network share or Ceph. |
| **Tag** | A label you give guests in Proxmox, shown in color in VisorMon and the Proxmox web interface. |

| | |
|---|---|
| **Task** | A job Proxmox runs in the background, such as a backup or a migration, with a log you can read. |
| **Template** | A guest kept as a pattern for new guests. It does not run. |
| **tmux** | A program on the node that keeps a shell running when the connection drops, so you can carry on where you left off. |
| **VM** | A virtual machine: a complete computer, with its own operating system, run by Proxmox with QEMU/KVM. |
| **VMID** | The number Proxmox gives each guest, such as 100. It is unique within a cluster. |

## 13.4 Support

Questions or problems? Email support@jimmyfatmasterson.com. It helps to say which phone you use, which version of Proxmox VE you run, and what you see on screen. Troubleshooting covers the most common problems.

See also the VisorMon privacy policy.

Proxmox is a registered trademark of Proxmox Server Solutions GmbH. VisorMon is an independent app and is not affiliated with, endorsed by, or sponsored by Proxmox Server Solutions GmbH. iPhone, Face ID and App Store are trademarks of Apple Inc. Android and Google Play are trademarks of Google LLC.