

Deploy Ubuntu 26.04 LTS Server on Proxmox VE

Create, install, and validate an Ubuntu 26.04 LTS Server virtual machine on Proxmox VE: a q35 and OVMF build whose Secure Boot state is verified on the EFI disk and again inside the guest, VirtIO storage and networking, guest-agent integration, a firewall and update baseline, and a backup that ends in a restore test.

PROXMOX VE 9.2 · UBUNTU 26.04 LTS · OVMF SECURE BOOT · VIRTIO · HOW-TO GUIDE

General educational guidance, not affiliated with or endorsed by Proxmox Server Solutions GmbH or Canonical Ltd. Wizard values and guest behavior were validated on Proxmox VE 9.2.10 with the Ubuntu 26.04 GA server image; sizing, addresses, and policy are deployment choices. Published 2026-09-08.

9 steps

the host record first, then a verified install, then autostart and backup

3 checks before the installer runs

the firmware package version, the ISO checksum and Canonical's signature, then the Secure Boot certificates on the EFI disk

This guide reproduces a validated Ubuntu 26.04 LTS Server build, the VM named `callisto` on node `pve01`: a q35 machine with OVMF firmware and pre-enrolled Secure Boot keys, VirtIO SCSI storage on an NVMe LVM-thin pool, VirtIO networking, and the QEMU Guest Agent. It lists every wizard value with its reason, marks the settings you have to decide for your own site, and ends with guest verification, a security baseline, and a restore-tested backup.

CONTEXT

What is fixed and what is a choice

The example sizes are a baseline

Two vCPUs, 4 GiB of fixed RAM, and a 64 GiB system disk suit a general-purpose server. Canonical's requirements page suggests at least 3 GB of RAM and 25 GB of storage, and states its figures are for Ubuntu 24.04 amd64. Treat the sizes here as workload decisions.

Check which Secure Boot certificates the EFI disk received

The wizard's **Pre-Enroll keys** checkbox requests the standard key set, but the pre-creation summary cannot show which Microsoft certificates the EFI disk received. Step 04 reads the marker back from the created VM before the installer runs.

Fixed memory trades away guest-usage reporting

With the ballooning device disabled, Proxmox loses the guest-memory information that device supplies, so the VM summary can show nearly all 4 GiB in use while Ubuntu is mostly idle. Judge real pressure inside Ubuntu with `free -h`, especially the **available** column.

MAP The validated example

ITEM	VALIDATED VALUE
Node / Proxmox VE	pve01 , Proxmox VE 9.2.10, management at 192.168.1.70/24
Storage roles	local for ISOs; vmfast (NVMe LVM-thin) for VM disks; backup-local for local backups
VM network	Bridge vmbr0 , no VLAN tag

PROFILE	VALIDATED VALUE
VM	ID 100 , named callisto ; guest type Linux, ostype: l26
Machine / firmware	q35; OVMF (UEFI); EFI disk on vmfast , 4 MiB type, pre-enrolled keys; no TPM
CPU / memory	1 socket, 2 cores, type host ; 4096 MiB fixed, ballooning device disabled
System disk	SCSI 0, 64 GiB on vmfast ; discard, IO thread, SSD emulation, and backup enabled; no cache; io_uring default
Network / agent	VirtIO on vmbr0 , firewall flag enabled; QEMU Guest Agent channel enabled

In Proxmox VE 9.2.10 the wizard's Linux version selector reads **7.x - 2.6 Kernel** and stores **ostype: l26** ; older guides quoting "6.x - 2.6 Kernel" describe an outdated label. The management address belongs to the Proxmox host: the Ubuntu guest gets its own DHCP lease or approved static address.

WHY Design decisions

Q35 and OVMF, with Secure Boot on from the start

MODERN PCIE TOPOLOGY • PRE-ENROLLED KEYS

q35 with OVMF gives the VM a PCI Express topology and a UEFI boot environment, and keeps later PCIe passthrough open. **Pre-Enroll keys** loads distribution and Microsoft Standard keys onto the EFI disk and enables Secure Boot by default. Ubuntu also runs fine on the SeaBIOS and i440fx defaults where legacy compatibility is a requirement; switching an installed VM between firmware types or machine families changes the hardware the guest sees, so treat that as a migration with a backup and a tested plan.

VirtIO SCSI single with an I/O thread

PROXMOX'S PERFORMANCE RECOMMENDATION • UBUNTU INCLUDES THE DRIVERS

Proxmox recommends the VirtIO SCSI single controller with the IO Thread option where disk performance is a requirement, and it is the default for newly created Linux VMs since Proxmox VE 7.3: each disk gets its own controller and QEMU handles its I/O in a dedicated thread. Ubuntu includes the VirtIO drivers, so no driver ISO is involved.

LVM-thin with discard and SSD emulation

TRIM REACHES THE THIN POOL • MONITOR THE POOL CAPACITY

Discard lets the guest's TRIM requests reach the thin pool so freed blocks are reclaimed, and SSD emulation presents the disk as non-rotational to match the NVMe backing. The no-cache default avoids an extra QEMU write-back layer. Thin provisioning still needs monitoring: when the host pool fills, guest writes can fail even while the guest filesystem reports free space.

Host CPU and fixed memory, and what each one costs

STANDALONE NODE • BALLOON: 0 HIDES GUEST USAGE

The wizard defaults to `x86-64-v2-AES`; the `host` CPU type chosen here exposes the node's full feature set and costs portability to differing hosts, an acceptable trade on the standalone `pve01`. Disabling the ballooning device (`balloon: 0`) makes the 4096 MiB allocation unambiguous but drops the guest-usage reporting the device supplies; Proxmox's documentation says to generally leave it enabled, and setting **Minimum memory** equal to **Memory** keeps the allocation just as fixed with reporting intact. The example's disabled device is a deliberate choice.

01 Approvals, capacity, and the host record

- Confirm the deployment is approved, the node is healthy with pending updates evaluated under change policy, and current backups of anything the change could touch exist.
- Confirm free capacity: do not allocate the example's resources where they would create contention. `vmfast` must be enabled and healthy with room for the 64 GiB disk, `local` must permit **ISO image** content, and `vmbr0` must be up on the intended network.
- Confirm the address plan (DHCP reservation or approved static address, DNS records, VLAN decision), that VM ID `100` is free (or let Proxmox assign one), the backup target and retention, and have the administrator credential and an SSH public key ready.

```
pveversion --verbose
lscpu | grep -E '^ (Architecture|CPU(s)|Model name):'
free -h
pvesm status
ip -br address show vmbr0
ip route
```

Resolve unexpected storage, memory, bridge, address, or route results before creating the VM. Then confirm the OVMF firmware package can write the current Microsoft certificate set:

```
ovmf_version="$(dpkg-query -W -f='${Version}' pve-edk2-firmware-ovmf 2>/dev/null)"
printf 'pve-edk2-firmware-ovmf: %s\n' "${ovmf_version:-not installed}"
if [ -n "$ovmf_version" ] && dpkg --compare-versions "$ovmf_version" ge '4.2025.05-1'
then
    echo '2023-certificate-capable package confirmed'
else
    echo 'Update required before relying on a newly created EFI disk'
fi
```

`pve-edk2-firmware-ovmf` is the package that provides `OVMF_VARS_4M.ms.fd`, the variables template **Pre-Enroll keys** copies onto a new EFI disk. Proxmox's documentation names `pve-edk2-firmware`, which on Proxmox VE 9 is a metapackage kept for the upgrade from the older single package and carries no firmware; nothing on a fresh 9.x install pulls it in, so that name reports no version on a node whose firmware is current.

On Proxmox VE 9.1 and later this check passes on its own, because `qemu-server` itself depends on a firmware package above this floor. It earns its place on a Proxmox VE 9.0 node still carrying `qemu-server` 9.0.24 or older, and on a host part-way through an upgrade from Proxmox VE 8.

Do not proceed until the confirmation line prints. If an update is needed, update the node through the normal full-node update and reboot process and re-run the check; upgrading one package out of the supported set is a different procedure with its own risks.

02 Download, verify, and upload the ISO

The example uses the Ubuntu 26.04 LTS GA (general availability) server image, downloaded only from an official Ubuntu release location or an approved mirror. Canonical publishes this SHA-256 for the GA filename:

```
ubuntu-26.04-live-server-amd64.iso
dec49008a71f6098d0bcfc822021f4d042d5f2db279e4d75bdd981304f1ca5d9
```

- The checksum above belongs to this exact GA filename. Canonical's download page now leads with the 26.04.1 point release and keeps both images in the same directory, so check whichever file you took against the current `SHA256SUMS`. Take 26.04.1 for a new deployment; the GA image is named here because it is what this build was validated on. Put your own filename into the commands below.
- Verify Canonical's signature on that checksum list before trusting it. A hash comparison shows the image arrived intact; the signature is what shows Canonical published it. Run the lines below in the directory holding the downloaded ISO: `sha256sum -c` reads files from the current directory and reports `no file was verified` when the image is elsewhere.

```
curl -O https://releases.ubuntu.com/26.04/SHA256SUMS
curl -O https://releases.ubuntu.com/26.04/SHA256SUMS.gpg
gpg --keyid-format long --keyserver hks://keyserver.ubuntu.com \
    --recv-keys 0x46181433FBB75451 0xD94AA3F0EFE21092
gpg --keyid-format long --verify SHA256SUMS.gpg SHA256SUMS
sha256sum -c SHA256SUMS --ignore-missing
```

`gpg --verify` prints `WARNING: This key is not certified with a trusted signature!` on a good signature too; it says only that you have not countersigned the key. The line to read is `Good signature from`, and the fingerprint printed under it has to match one of the two Canonical publishes in its verification tutorial.

- Upload the image at `pve01 > local > ISO Images > Upload`, wait for the task to finish, and confirm the file is listed.
- Re-hash the uploaded copy from the node's `Shell`, naming the file you uploaded; the result must match exactly, or delete the copy and obtain the image again:

```
sha256sum /var/lib/vz/template/iso/ubuntu-26.04-live-server-amd64.iso
```

03 Create the VM to the validated profile

Select **Create VM** and enable the wizard's **Advanced** view, which reveals startup behavior, SSD emulation, ballooning, and several other fields used below. Verify each value rather than assuming the defaults.

TAB	SETTING	VALUE
General	Name / Start at boot	<code>callisto</code> ; start at boot disabled until validation is complete; pool, HA, and architecture at their defaults
OS	ISO / Type / Version	<code>ubuntu-26.04-live-server-amd64.iso</code> , or the point release you took; Linux, 7.x - 2.6 Kernel

TAB	SETTING	VALUE
System	Graphic card / Machine / BIOS	Default display; q35; OVMF (UEFI); no TPM
System	EFI disk / Controller / Agent	EFI disk on <code>vmfast</code> with Pre-Enroll keys enabled; VirtIO SCSI single ; Qemu Agent enabled
Disks	SCSI 0 on <code>vmfast</code>	64 GiB baseline; Discard, IO thread, SSD emulation, and Backup enabled; cache and Async IO at their defaults
CPU	Sockets / Cores / Type	1 / 2 baseline / <code>host</code> ; NUMA, limits, affinity, and extra flags at their defaults
Memory	Fixed allocation	4096 MiB baseline; Ballooning Device disabled (see the design card); Allow KSM left enabled, which is its default
Network	Bridge / Model	<code>vmbrio</code> ; VirtIO (paravirtualized) ; no VLAN tag; Firewall flag enabled in this example, set it per site policy; MAC, MTU, and queues at their defaults

Do not put the disk on `local-lvm` just because the wizard offers it first: the design places VM disks on `vmfast`. **Skip replication** changes nothing here, since Proxmox storage replication supports only ZFS (`zfspool`) storage and `vmfast` is LVM-thin. Enter a VLAN tag only where the bridge, uplink, and switch are configured for it, and enable `nested-virt` only for an assessed nested-virtualization workload.

The per-NIC **Firewall** flag only makes the interface eligible for Proxmox firewall processing; switching that firewall on at the datacenter and VM level drops inbound traffic to the guest until rules allow it, so add the SSH rule there before you enable it. A guest that takes its address by DHCP also needs the **DHCP** option under its own **Firewall > Options**, which is off by default, or it loses the lease at the next renewal.

Review the Confirm tab, enable **Start after created**, and select **Finish**.

04 Verify the EFI disk before installing

The wizard summary only shows what you asked for. As soon as the VM exists, read the EFI disk back on the node:

```
qm config 100 --current 1 | grep '^efidisk0:'
```

- The line must contain `efitype=4m`, `pre-enrolled-keys=1`, and `ms-cert=2023k`. The 4 MiB type is the GUI default for new VMs and is required for Secure Boot.
- `--current 1` is what makes this a check on the running VM. Plain `qm config` prints pending changes as though they were already applied; the remediation path below depends on that difference.
- `ms-cert=2023k` means the Microsoft UEFI CA 2023, Windows UEFI CA 2023, and Microsoft Corporation KEK 2K CA 2023 certificates are enrolled. Proxmox documents that the original 2011 certificate set expires in June 2026 and that an EFI disk carrying only it rejects bootloaders signed with the 2023 certificates.
- The markers `ms-cert=2023` and `ms-cert=2023w` are deprecated and may indicate partial enrollment (the VM start task log warns about them); treat them like a missing marker.
- The marker records what Proxmox enrolled onto the disk. Step 07 reads the certificates themselves from inside the guest.

- **If the marker is missing or partial**

Do not install yet. Shut the VM down and run `qm enroll-efi-keys 100`, which writes the certificates while the disk is idle. The **Hardware > EFI Disk > Disk Action > Enroll Updated Certificates** route works on a running VM, and Proxmox applies it the next time the VM starts, so stop and start the VM before reading the configuration again. Continue only once `qm config 100 --current 1` shows `ms-cert=2023k`.

Where the VM was created without **Pre-Enroll keys**, neither route enrolls anything: the command prints `skipping - no pre-enrolled keys or already got ms-cert=2023k marker` and exits without error, and the menu item stays greyed out. The marker is missing here, so that line is telling you the disk has no pre-enrolled keys. Nothing on the disk matters yet, so with the VM shut down, delete the EFI disk under **Hardware** and add it back with **Pre-Enroll keys** enabled.

On a running VM both edits become pending changes, and `qm config 100 --current 1` goes on showing the old disk until it restarts.

This build applies the fix before any operating system is on the disk; for an existing production VM, take a verified backup and follow change control before altering EFI certificates.

05 Install Ubuntu Server

Open the VM **Console**, start the VM if needed, and choose the normal installation option at the boot menu (commonly **Try or Install Ubuntu Server**). Installer wording shifts slightly between versions, so work from the decisions rather than the exact screen text:

- Pick the language and keyboard, apply an offered installer update when outbound access and change policy permit, and choose the standard Ubuntu Server installation unless a minimized build is an explicit requirement.
- Confirm the VirtIO interface shows the expected DHCP address, gateway, and DNS, or enter the approved static values. Leave the proxy blank and the mirror at its default unless the site requires otherwise.
- Use the whole 64 GiB virtual disk. The installer's guided LVM layout is fine and does not conflict with the host's LVM-thin pool: one runs inside the guest, the other on the Proxmox host.
- Enable guest disk encryption only with an approved key-management and unattended-restart plan. This profile has no TPM, so there is no hardware-backed unlock and the passphrase is typed at the console at every boot. You can add a TPM state device before installing, but Proxmox's documentation says an emulated TPM gives no real security benefit, because its state sits on an ordinary volume that anyone with host access can edit; on this design that volume is in the same `vmfast` pool as the encrypted disk.
- Guided LVM's default sizing policy deliberately leaves volume-group space free for snapshots and growth, allocating about half of it to the root logical volume at this disk size, so the root filesystem comes out well under 64 GiB. Grow the root volume later with LVM's `lvextend` when the workload needs the space.
- Name the server `callisto` (to reproduce the example), create the approved administrator account with a unique credential, and skip Ubuntu Pro unless the deployment requires it.
- Install the OpenSSH server where remote administration is planned and import an approved public key. Skip featured server snaps that are outside the solution design; they can be added later.
- The confirmation step erases the VM's 64 GiB virtual disk and writes only there. It does not touch the Proxmox host's system disk, the other volumes in the `vmfast` pool, or the NVMe's partition table. Confirm and let the installation finish.

Reboot and detach the ISO. Select **Reboot Now**; if the installer asks to remove the medium, edit the CD/DVD drive under **Hardware** to **Do not use any media** (leave the SCSI disk alone) and continue. Confirm Ubuntu boots from the virtual disk rather than the installer; the ISO file stays in `local` for future builds.

06 Integrate the guest

Sign in on the console before relying on SSH. Update, then install the guest agent and the baseline services:

```
sudo apt update
sudo apt upgrade
sudo apt install qemu-guest-agent
sudo systemctl restart qemu-guest-agent
systemctl is-active qemu-guest-agent
sudo apt install openssh-server ufw
systemctl is-active ssh.socket
sudo ss -tlnp | grep -E '.*:22\s'
if test -f /run/reboot-required; then cat /run/reboot-required; fi
```

- Review the proposed upgrade before approving it. `qemu-guest-agent` and `ssh.socket` should both return `active`, and if a reboot is requested, take it in the approved window with `sudo reboot`.
- Ubuntu starts sshd through systemd socket activation. `ssh.socket` opens port 22 at boot and hands the listening socket to `ssh.service` when the first connection arrives, so `systemctl is-active ssh` reads `inactive` until someone logs in and `active` from then on. Check `ssh.socket`, which is up from boot either way; the `ss` line confirms something is listening on the port.
- `openssh-server` is already there if you selected it during the install. The command covers a build that skipped it.
- `qemu-guest-agent` comes from Ubuntu's official `universe` component, where Canonical publishes the security updates the community produces and gives no commitment of its own. If APT cannot find the package, fix the configured sources; do not substitute an unverified third-party package.

In the Proxmox summary, "Guest Agent not running" is expected until the package and the channel are both up; refresh after the restart and confirm the guest IP appears.

07 Verify the guest

```
hostnamectl
lsblk -o NAME,SIZE,TYPE,FSTYPE,MOUNTPOINTS
df -hT
ip -br address
ip route
resolvectl status
getent hosts archive.ubuntu.com
timedatectl status
free -h
nproc
test -d /sys/firmware/efi && echo "UEFI boot confirmed"
systemctl is-enabled fstrim.timer
systemctl is-active qemu-guest-agent
```

- Check the hostname first, then the partitions, mount points, and free space. Under guided LVM the root filesystem is deliberately smaller than the 64 GiB disk, and the rest of the volume group stays free for expansion.
- The address and default route must be the guest's own, never the host's `192.168.1.70`.
- DNS must resolve, time synchronization must be active, `nproc` must report two processors, and the `available` memory figure must be reasonable for the workload.
- The UEFI check must print its confirmation line, and `qemu-guest-agent` must report `active`.
- The `getent` line tests resolution of Ubuntu's public archive name; where public DNS is blocked or an internal mirror is used, substitute an approved internal name and record the result.

- If `fstrim.timer` is not enabled, review storage policy and then enable it; periodic TRIM is normally desirable on this discard-enabled thin storage: `sudo systemctl enable --now fstrim.timer`

Then verify Secure Boot inside the guest:

```
sudo apt install mokutil
mokutil --sb-state
mokutil --db | grep -i 'UEFI CA 2023'
mokutil --kek | grep -i 'KEK 2K CA 2023'
```

`mokutil` is already on a standard Ubuntu Server build; the install line covers a minimized one. The expected result for this pre-enrolled-key OVMF build is `SecureBoot enabled`, with the two certificate lines each returning a match.

`--sb-state` reports that the firmware is enforcing and says nothing about which certificates are present, so the `--db` and `--kek` lines are what confirm the 2023 set reached the variables. Investigate any other answer before counting Secure Boot as a verified control.

08 Apply the network and security baseline

- Give the guest a predictable address: a DHCP reservation on its generated MAC address, or an approved static configuration. Create the forward and reverse DNS records once the address is set.
- Test SSH from an authorized management system first, then allow it before you enable Ubuntu's firewall, so turning the firewall on cannot lock you out of the session you are working in. `ufw enable` asks `Command may disrupt existing ssh connections. Proceed with operation (y|n)?` before it starts:

```
sudo ufw app list
sudo ufw allow OpenSSH
sudo ufw enable
sudo ufw status verbose
```

- If no `OpenSSH` application profile is listed, allow the approved port explicitly instead, for example `sudo ufw allow 22/tcp`. That profile is fixed at 22/tcp, so moving SSH to another port needs a rule of its own. Add only the ports the server's role requires. UFW and the Proxmox firewall are separate enforcement layers; keep the rules consistent at both.
- Harden SSH deliberately: give each administrator their own key instead of a shared password, restrict the source addresses to the management networks where the network allows it, disable direct root login, and keep a tested console path open before you change anything about authentication. On this socket-activated setup the listening port and address come from `sshd_config` through a systemd generator, so a port change needs `systemctl daemon-reload` and a restart of `ssh.socket` before it takes effect.
- Verify automatic security updates rather than assuming them. Ubuntu Server installs `unattended-upgrades` by default and enables automatic security updates through the `apt-daily.timer` and `apt-daily-upgrade.timer` units, but images and policy automation can change that default:

```
dpkg-query -W -f='${db:Status-Abbrev} ${binary:Package} ${Version}\n' unattended-upgrades
systemctl list-timers 'apt-daily*'
sudo unattended-upgrade --dry-run --debug
```

Install the package from the official repositories and re-run the checks if it is missing. Schedule application-aware maintenance and reboot handling to the service's availability requirements. This step sets a baseline; the *anoBytes guides Secure and harden SSH access on Linux, Configure and harden a host firewall on Linux, and Establish a secure server baseline on Linux* take each part further.

09 Finalize in Proxmox: readback, autostart, backup

Read the effective configuration back and compare it against the profile table above; the actual LVM volume names and generated MAC address differ from the wizard's shorthand, values left at their defaults such as the cache mode and Async IO do not appear at all, and after the ISO detach the CD/DVD entry may be absent or show no media. Confirm `efidisk0` still carries all three step 04 values:

```
qm config 100 --current 1
```

Enable **Start at boot** under **Options** only after the owner confirms the service should start with the node, and set a startup order, delay, or shutdown timeout only where a real dependency or workload requires one. Then pick the backup mode deliberately:

MODE	PROXMOX BEHAVIOR	USE WHEN
Snapshot	Lowest downtime, small inconsistency risk; with the guest agent enabled and running, Proxmox freezes and thaws the guest filesystems during the live backup.	The routine mode for this VM once the agent is verified.
Stop	Orderly guest shutdown, then a background live backup; highest consistency at the cost of a short interruption.	The service tolerates downtime and needs the highest VM-level consistency.
Suspend	Suspends the VM before a snapshot-mode backup; longer downtime without necessarily better consistency. Proxmox recommends snapshot mode instead.	Compatibility only.

- A filesystem freeze improves filesystem consistency; databases and transactional workloads keep their own supported backup or quiescing procedure for application-consistent recovery.
- Run **Backup now** to `backup-local` in the chosen mode and confirm the task ends with **TASK OK**, then configure the scheduled job and retention under **Datacenter > Backup**.
- `backup-local` protects against failure of the separate `vmfast` disk but still sits inside the same server, so replicate critical backups to separate hardware or another approved location, and prove the chain with a controlled restore test: restore the backup into a new VM ID with a fresh MAC address, confirm it boots, and keep the copy isolated or delete it, since it still carries the original's hostname and address.
- Ask for that fresh MAC explicitly, with `qmrestore ... --unique` or the **Unique** box in the web interface. Both routes reuse the original's MAC by default, and a second VM answering on it costs the running VM its network on the same bridge and takes over its DHCP reservation.

LIMITS Operational limits to hand the owner

TOPIC	GUIDANCE
Thin-pool capacity	Monitor <code>vmfast</code> data and metadata usage on the host and filesystem capacity in the guest separately; a full pool fails guest writes even while the guest shows free space.
Memory reporting	With <code>balloon: 0</code> the host summary overstates usage; judge pressure inside Ubuntu, or enable the ballooning device with minimum memory equal to memory to restore reporting with the same fixed allocation.
Mobility and replication	The <code>host</code> CPU type limits migration to matching hosts, and Proxmox storage replication is unavailable on LVM-thin (ZFS storage only).

TOPIC	GUIDANCE
Backups	<code>backup-local</code> shares the chassis with the node; keep an approved off-host copy for anything you cannot afford to lose, and restore-test on a schedule, restoring to a new VM ID with <code>--unique</code> so the copy gets its own MAC address. LVM-thin supports snapshots, which make a quick checkpoint before a risky guest change, but a snapshot lives in the same <code>vmfast</code> pool as the disk it snapshots, so losing that pool loses both.
KSM	Kernel samepage merging stays at the host default (enabled). Proxmox documents research showing one VM can infer information about another on the same host through KSM, so a host serving multiple tenants should weigh disabling it, and adds that some countries require it off. Turn it off for one guest with <code>qm set 100 --allow-ksm 0</code> , or for the node.
Updates and certificates	Ubuntu 26.04 LTS carries five years of standard security maintenance from its April 2026 release, extendable with Ubuntu Pro. That commitment covers the <code>main</code> component. <code>qemu-guest-agent</code> comes from <code>universe</code> , which Canonical leaves to community updates on a standard LTS and covers for ten years through Ubuntu Pro's esm-apps stream, so track that package yourself or attach Ubuntu Pro. Keep host and guest on documented patch schedules, and re-check the EFI certificate marker and guest Secure Boot state after material firmware or Secure Boot changes.

HELP Troubleshooting

- **The VM returns to the installer after a reboot**

Detach the ISO from the CD/DVD drive and check **Options > Boot Order** so the SCSI system disk comes before removable media. Do not switch from OVMF to SeaBIOS as a troubleshooting step; that changes the boot environment the guest was installed for.

- **The VM does not boot in OVMF**

Confirm the EFI disk exists, the machine is q35 with OVMF, and `qm config 100 --current 1` shows the three values step 04 checks; remediate per step 04 while the VM is shut down. Confirm the installation completed onto the SCSI disk, and read the task log and console before changing anything about Secure Boot.

- **The guest has no network address**

Inside Ubuntu run `ip -br address` and `ip route`; in Proxmox confirm the NIC is VirtIO on `vmbr0` and not marked **Disconnect**. Confirm DHCP is reachable or the static values are right, the VLAN decision matches the VM, bridge, uplink, and switch, and review the Proxmox firewall and UFW independently.

- **Proxmox says the guest agent is not running**

Confirm **Options > QEMU Guest Agent** is enabled, the `qemu-guest-agent` package is installed, and `systemctl status qemu-guest-agent` is active. Reboot the guest if the package went in after the VM had already started and the channel is still not detected.

- **Proxmox shows nearly 100% memory usage**

Expected with `balloon: 0`; it does not alone prove Ubuntu is out of memory. Check `free -h` and `vmstat 1 5` inside the guest and investigate low available memory, sustained swap activity, or out-of-memory events. If host-side guest-memory reporting is required, shut the VM down and enable the ballooning device with minimum memory equal to memory.

- **Discard does not reclaim host capacity**

Confirm the disk has **Discard** enabled, `fstrim.timer` is enabled in Ubuntu, and the filesystem supports TRIM, then allow time for the usage displays to refresh. Do not run storage-repair or thin-pool commands without a backup and an approved recovery plan.

DONE What you achieved

- An Ubuntu 26.04 LTS Server VM on q35 and OVMF with three Secure Boot facts on record: `ms-cert=2023k` on the EFI disk before the install, then `SecureBoot enabled` and the 2023 certificates themselves read out of the guest's own variables with `mokutil` after it.
- VirtIO SCSI single with an I/O thread, discard reaching the NVMe thin pool, VirtIO networking, and the guest agent reporting the guest IP into Proxmox.
- A guest with its own verified address, DNS, and time sync, a UFW baseline brought up behind a tested SSH path, and automatic security updates checked on the running server.
- A backup that ended **TASK OK** with a scheduled job, an off-host copy planned, a restore test on the calendar, and the build's limits recorded for the owner.

REF References

Proxmox VE documentation: Qemu/KVM Virtual Machines (OVMF and Secure Boot certificates, CPU types, memory and ballooning, disks and IO threads), Backup and Restore, Storage Replication, and the qm manual · pve.proxmox.com/pve-docs

Proxmox VE wiki: Qemu-guest-agent · pve.proxmox.com/wiki

Canonical: Ubuntu 26.04 LTS release files and SHA256SUMS, download verification, Server installation and the installer's screen-by-screen reference, system requirements, OpenSSH, firewalls, automatic updates, and the release cycle · releases.ubuntu.com · ubuntu.com/server/docs · ubuntu.com/tutorials/how-to-verify-ubuntu · canonical-subiquity.readthedocs-hosted.com

Canonical: SSHd socket-based activation in Ubuntu 22.10 and later, and the maintenance the universe component carries · discourse.ubuntu.com · ubuntu.com/security/esm

Ubuntu package archive: qemu-guest-agent in 26.04 (universe) · packages.ubuntu.com · pve-manager source: the guest OS selector labels · github.com/proxmox/pve-manager