Overview

This documenation describes a real-world virtualization lab by running Proxmox VE inside a VirtualBox VM, enabling a fully local environment for learning DevOps, infrastructure engineering, virtualization, and cloud concepts.

Proxmox & clustering — why it matters

Proxmox VE is a Debian-based, open-source hypervisor that runs KVM virtual machines and LXC containers, all managed through a web UI.

Benefits of clustering

- Single control plane: manage multiple Proxmox nodes from one dashboard.
- Live migration: move VMs between nodes with minimal downtime.
- High availability (HA): automatically restart VMs on healthy nodes if a node fails.
- Replication: scheduled syncing of VM data across nodes for fast recovery.
- Scalability: add nodes to increase capacity without reorganizing your setup.
- Better resource utilization: distribute CPU, memory, and storage load across the cluster.

When to use it

- Learning DevOps, infrastructure, or cloud concepts.
- Testing HA, migration, and replication workflows.
- Running multi-node labs that mirror production operations.

Key trade-offs

- Network and storage design become more important (latency, bandwidth, shared storage).
- Cluster management adds operational complexity and requires monitoring.
- Some features (e.g., HA, efficient replication) need reliable networking and proper fencing/qpinger setup.

Quick checklist to get started

1. Ensure time sync (NTP) and reliable networking between nodes.
2. Use separate networks for management, replication (migration), and cluster communication.
3. Configure fencing/qdevice or quorum helpers for safety in failure scenarios.
4. Test live migration and replication in your lab before trusting production workloads.

Key components

- Three computers or VMs with at least 6 GB RAM and 100 GB storage space per machine
    - Storage can be lower as we will be using dynamically allocated virtual disks
- VirtualBox — host hypervisor that runs the Proxmox VM
- Proxmox VE ISO — installed as the nested hypervisor inside VirtualBox
- Ubuntu Server ISO — guest OS installed in a VM managed by Proxmox
- Debian Server ISO — guest OS install in a VM managed by Proxmox
- VBoxManage — CLI for creating and configuring the outer VirtualBox VM
- Proxmox Web UI — dashboard used to manage the inner guest VMs

2-Nods vs 3-Node Cluster

A 2-node cluster is simpler to set up and fine for learning, but it lacks a proper quorum (the voting system that keeps a cluster running when a node fails). That means HA isn’t reliable without extra workarounds to prevent the cluster from stalling.

A 3-node cluster includes quorum by default: two of three nodes can keep the cluster running if one goes down. It’s more stable, supports real HA testing, and scales better—though it requires one more machine and a bit more setup.

We’ll use a 3-node cluster because it’s the best balance of stability and realistic, hands-on experience.

Lab architecture (my current setup)

- AMD Ryzen 7 7800X3D

- 32 GB RAM
- 1 TB NVMe storage

Overview of what will be built

- Server Template inside VirtualBox to clone
- Installation Proxmox VE inside a VirtualBox VM clones
- Configured networking so the Proxmox Web UI is reachable from the host browser.
- Uploaded the Ubuntu Server ISO into Proxmox and created a guest VM.
- Launched and installed Ubuntu inside Proxmox.
- Resolved nested-virtualization KVM errors by disabling KVM for the guest.

Network configuration

- Adapter 1: Bridge — outbound internet access and management network.
- Adapter 2: Host-Only — host ? Proxmox cluster communication (coresync) network.
- Adapter 3: Host-Only — host ? Proxmox replication (migration) network.

I have choosen the following IP-Addresses for management, coresync and replication:

- Node 1:
  - nic0 `192.168.0.201` (hostname `prox01.local`)
  - nic1 `172.20.1.201`
  - nic2 `172.20.2.201`
- Node 2:
  - nic0 `192.168.0.202` (hostname `prox02.local`)
  - nic1 `172.20.1.201`
  - nic2 `172.20.2.201`
- Node 3:
  - nic0 `192.168.0.203` (hostname `prox03.local`)

- nic1 `172.20.1.201`
- nic2 `172.20.2.201`

You should adapt the IP address of nic0 (the main interface of the Proxmox node) so that it is in the same subnet as your PC (for example, both in 192.168.0.0/24). This allows direct communication without additional configuration. If they are in different subnets, a router with the correct routing rules is required — otherwise, your PC will not be able to reach the Proxmox node.

Once you have downloaded all the ISO images, continue on the next page with the installation and setup of the Proxmox nodes and networking.

Revision #4

Created 2026-03-30 07:59:26 UTC by Carsten

Updated 2026-03-31 16:09:31 UTC by Carsten