

# Virtualization: Proxmox VE

This page outlines how I installed, configured, and currently use **Proxmox VE** as the foundation of my home lab virtualization platform. It includes steps for installing Proxmox, setting up storage, creating virtual machines and LXC containers, and using templates for rapid deployment.

## ? 1. Proxmox VE Installation

### ? Requirements:

- CPU with virtualization support (Intel VT-x / AMD-V)
- 8GB+ RAM (16GB recommended)
- SSD/HDD storage
- Bootable USB with Proxmox ISO

### ? Steps:

1. Download the latest ISO: <https://www.proxmox.com/en/downloads>
2. Create a bootable USB drive with Rufus or the `dd` command.
3. Boot from USB and follow the installation wizard:
  - Accept EULA
  - Select target disk (use ZFS if using multiple disks or for snapshots)
  - Set country, time zone, and keyboard
  - Set root password and email
  - Assign a static IP, hostname (e.g., `proxmox.local`), and gateway
4. Reboot and access the web GUI:  
`https://<proxmox-ip>:8006`

---

## ?? 2. Initial Configuration

### ? Post-Install Checklist:

- Update system:

```
apt update && apt full-upgrade -y
```

- (Optional) Remove subscription nag:

```
sed -i.bak "s|data.status != 'Active'|false|" /usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js
```

```
systemctl restart pveproxy
```

- Add no-subscription repo in `/etc/apt/sources.list.d/pve-enterprise.list`:

```
deb http://download.proxmox.com/debian/pve bookworm pve-no-subscription
```

- Add SSH key for secure login
- Set up storage (local-lvm, additional disks via ZFS or ext4)

## ? 3. Creating a Virtual Machine (VM)

### Steps:

1. Upload ISO to `local` or `local-lvm` storage
2. Go to **Create VM**
3. General:
  - Name: `ubuntu-server`
4. OS:
  - Select ISO image
5. System:
  - Use default (UEFI or BIOS)
6. Hard Disk:
  - Choose `local-lvm`, disk size
7. CPU & Memory:
  - 2 cores, 4GB RAM (example)
8. Network:
  - Select `vmbri0` (bridge to LAN)
9. Confirm and finish

### Post-VM:

- Start VM
- Install OS (e.g., Ubuntu, Debian, Windows)
- Set static IP, install guest agent:

```
apt install qemu-guest-agent  
systemctl enable --now qemu-guest-agent
```

## ? 4. Creating LXC Containers

### LXC Benefits:

- Lightweight, fast
- Ideal for services like DNS, Pi-hole, NGINX, Git, etc.

## Steps:

1. Go to **Create CT**
2. General:
  - Set hostname and password
3. Template:
  - Download from Proxmox template library
4. Root Disk:
  - Choose size and storage
5. CPU & Memory:
  - Allocate as needed
6. Network:
  - Assign static IP or DHCP
7. Confirm and finish

## Example Services via LXC:

- Pi-hole
  - NGINX reverse proxy
  - Node Exporter
  - Lightweight Docker host (with nesting enabled)
- 

## ? 5. Using and Creating Templates

### VM Templates:

1. Create and configure a VM (e.g., Ubuntu minimal)
2. Shutdown VM
3. Convert to template:

```
qm template <vmid>
```

### LXC Templates:

1. Download from **CT Templates** in Proxmox GUI
2. Or upload custom `.tar.gz` template to:

```
/var/lib/vz/template/cache/
```

## ?? 6. Best Practices & Tips

- **Backups:** Schedule daily/weekly backups to external storage (e.g., NFS or USB)
- **Snapshots:** Take before major updates or reconfigurations
- **Networking:** Use VLANs and firewall rules via pfSense

- **High Availability (Optional):** Use Proxmox cluster for HA with shared storage
  - **Resource Monitoring:** Integrate with **Zabbix** or built-in usage graphs
- 

Revision #1

Created 13 July 2025 06:51:13 by Admin

Updated 13 July 2025 07:11:15 by Admin