Frequently Asked Questions

Proxmox Server Solutions GmbH
<support@proxmox.com>
version 8.2.2, Thu Apr 25 09:24:16 CEST 2024

New FAQs are appended to the bottom of this section.

1. *What distribution is Proxmox VE based on?*

   Proxmox VE is based on Debian GNU/Linux

2. *What license does the Proxmox VE project use?*

   Proxmox VE code is licensed under the GNU Affero General Public License, version 3.

3. *Will Proxmox VE run on a 32bit processor?*

   Proxmox VE works only on 64-bit CPUs (AMD or Intel). There is no plan for 32-bit for the platform.

   

   VMs and Containers can be both 32-bit and 64-bit.

4. *Does my CPU support virtualization?*

   To check if your CPU is virtualization compatible, check for the `vmx` or `svm` tag in this command output:

   ```
   egrep '(vmx|svm)' /proc/cpuinfo
   ```

5. *Supported Intel CPUs*

   64-bit processors with Intel Virtualization Technology (Intel VT-x) support. (List of processors with Intel VT and 64-bit)

6. *Supported AMD CPUs*

   64-bit processors with AMD Virtualization Technology (AMD-V) support.

7. *What is a container/virtual environment (VE)/virtual private server (VPS)?*

   In the context of containers, these terms all refer to the concept of operating-system-level virtualization. Operating-system-level virtualization is a method of virtualization, in which the kernel of an operating system allows for multiple isolated instances, that all share the kernel. When referring to LXC, we call such

instances containers. Because containers use the host's kernel rather than emulating a full operating system, they require less overhead, but are limited to Linux guests.

8. *What is a QEMU/KVM guest (or VM)?*

   A QEMU/KVM guest (or VM) is a guest system running virtualized under Proxmox VE using QEMU and the Linux KVM kernel module.

9. *What is QEMU?*

   QEMU is a generic and open source machine emulator and virtualizer. QEMU uses the Linux KVM kernel module to achieve near native performance by executing the guest code directly on the host CPU. It is not limited to Linux guests but allows arbitrary operating systems to run.

10. *How long will my Proxmox VE version be supported?*

    Proxmox VE versions are supported at least as long as the corresponding Debian Version is oldstable. Proxmox VE uses a rolling release model and using the latest stable version is always recommended.

Proxmox VE Version	Debian Version	First Release	Debian EOL	Proxmox EOL
Proxmox VE 8	Debian 12 (Bookworm)	2023-06	tba	tba
Proxmox VE 7	Debian 11 (Bullseye)	2021-07	2024-07	2024-07
Proxmox VE 6	Debian 10 (Buster)	2019-07	2022-09	2022-09
Proxmox VE 5	Debian 9 (Stretch)	2017-07	2020-07	2020-07
Proxmox VE 4	Debian 8 (Jessie)	2015-10	2018-06	2018-06
Proxmox VE 3	Debian 7 (Wheezy)	2013-05	2016-04	2017-02
Proxmox VE 2	Debian 6 (Squeeze)	2012-04	2014-05	2014-05
Proxmox VE 1	Debian 5 (Lenny)	2008-10	2012-03	2013-01

11. *How can I upgrade Proxmox VE to the next point release?*

    Minor version upgrades, for example upgrading from Proxmox VE in version 7.1 to 7.2 or 7.3, can be done just like any normal update. But you

should still check the release notes for any relevant noteable, or breaking change.

For the update itself use either the Web UI *Node → Updates* panel or through the CLI with:

```
apt update
apt full-upgrade
```

> Always ensure you correctly setup the package repositories and only continue with the actual upgrade if `apt update` did not hit any error.

12. *How can I upgrade Proxmox VE to the next major release?*

    Major version upgrades, for example going from Proxmox VE 4.4 to 5.0, are also supported. They must be carefully planned and tested and should **never** be started without having a current backup ready.

    Although the specific upgrade steps depend on your respective setup, we provide general instructions and advice of how a upgrade should be performed:

    - Upgrade from Proxmox VE 7 to 8
    - Upgrade from Proxmox VE 6 to 7
    - Upgrade from Proxmox VE 5 to 6
    - Upgrade from Proxmox VE 4 to 5
    - Upgrade from Proxmox VE 3 to 4

13. *LXC vs LXD vs Proxmox Containers vs Docker*

    LXC is a userspace interface for the Linux kernel containment features. Through a powerful API and simple tools, it lets Linux users easily create and manage system containers. LXC, as well as the former OpenVZ, aims at **system virtualization**. Thus, it allows you to run a complete OS inside a container, where you log in using ssh, add users, run apache, etc...

    LXD is built on top of LXC to provide a new, better user experience. Under the hood, LXD uses LXC through `liblxc` and its Go binding to create and manage the containers. It's basically an alternative to LXC's tools and distribution template system with the added features that come from being controllable over the network.

    Proxmox Containers are how we refer to containers that are created and managed using the Proxmox Container Toolkit (`pct`). They also target **system virtualization** and use LXC as the basis of the container offering. The Proxmox Container Toolkit (`pct`) is tightly coupled with Proxmox VE. This means that it is aware of cluster setups, and it can use the same network

and storage resources as QEMU virtual machines (VMs). You can even use the Proxmox VE firewall, create and restore backups, or manage containers using the HA framework. Everything can be controlled over the network using the Proxmox VE API.

Docker aims at running a **single** application in an isolated, self-contained environment. These are generally referred to as “Application Containers”, rather than “System Containers”. You manage a Docker instance from the host, using the Docker Engine command-line interface. It is not recommended to run docker directly on your Proxmox VE host.

> If you want to run application containers, for example, *Docker* images, it is best to run them inside a Proxmox QEMU VM.

Version 8.2.2
Last updated Thu Apr 25 09:24:16 CEST 2024