

# vpsAdminOS

Since [OpenVZ](#) is dead and new distributions aren't supporting it, we had to find a way to upgrade our kernel, which meant choosing a different virtualization technology. Linux kernel now has some support for containers, so we've decided to stick with that. Next, we needed some distribution that we could use on nodes to serve as hypervisors, as a replacement of Scientific Linux 6 with OpenVZ kernel. We've chosen [NixOS](#), which allows you to declare the system and its configuration and then reproducibly build it. And since we have a bit specific requirements, we've created our own distribution on top of NixOS.

[vpsAdminOS](#) is based on [NixOS](#) and [not-os](#). It's a *live* distribution serving as a hypervisor for container virtualisation. It's as capable as OpenVZ Legacy was in its time. We have our own userspace tools to manage containers called `osctl`, which internally uses LXC. [vpsAdminOS](#) naturally integrates with [vpsAdmin](#), our administration interface with web interface, which you're all using to manage your VPS. However, [vpsAdminOS](#) is meant to be fully usable even on its own, as a replacement to OpenVZ Legacy deployments. If you have some OpenVZ servers and would like a newer system, you can consider [vpsAdminOS](#). We also have [scripts](#) to help with migration of OpenVZ containers onto [vpsAdminOS](#).

## Migration from OpenVZ to vpsAdminOS

The upgrade of our infrastructure with all VPSes to [vpsAdminOS](#) is divided into several phases:

1. Developing [vpsAdminOS](#) into something usable
2. Integration with [vpsAdmin](#)
3. Opening of a staging environment with [vpsAdminOS](#)
  1. Testing, fixing bugs, implementing missing features, preparing for production
4. New production nodes are using [vpsAdminOS](#) (🚧 we're here 🚧)
  1. [vpsAdminOS](#) is available in Prague
5. Gradual migration of all VPS from OpenVZ nodes to [vpsAdminOS](#), one node after another
6. End of story

## What does it mean for members

We're trying to make the migration to [vpsAdminOS](#) as seamless as possible, so that one day your VPS will stop on the OpenVZ node and will start on [vpsAdminOS](#) node a while later, without you having to do anything. However, it depends on what programs you're running and what configuration changes you have made. That's why we recommend for everyone to try VPS on [vpsAdminOS](#) in the [staging environment](#), so that we can find and solve problems before we start migration production VPS.

## Changes in VPS behaviour

## User namespaces

VPS in vpsAdminOS are using so called *user namespaces*. User namespace means that your system user and group IDs are mapped to different values on the host. For example, the root user in your VPS has UID 0, but from the host's point of view, its UID is e.g. 666000. Every member has been assigned a unique user namespace, which ensures that your data is isolated from other users. In case an attacker manages to leave the container, he will not be able to access data from VPS belonging to other members.

Every member is assigned a user namespace of 524288 user/group IDs. It means that you can use UID/GID from 0 to 524287. All VPS from one member are in the same user namespace. In the future, it will be possible to define custom UID/GID maps for VPS and NAS datasets, which will let each member to isolate his own VPS and yet share some chosen range of user/group IDs.

## General

Changes regarding VPS, but independent on the distribution used:

- `/proc/stat` reports only user (including system) and idle
- Swap is not shown in `/proc/meminfo`

## Debian/Ubuntu/Alpine

- Network is configured using `ip` from `iproute2`, you no longer need `ifconfig` from `net-tools`
- `/etc/network/interfaces.{head,tail}` aren't inserted into `/etc/network/interfaces`, but rather included using `source`, i.e. they do not affect contents of `/etc/network/interfaces` directly, like it was with `vzctl`.
- If there is a directory called `/etc/network/interfaces.d`, it is sourced before `/etc/network/interfaces.tail`.

## Behaviour changes in vpsAdmin

- NAS and snapshots are not accessed using vpsAdmin [mounts](#), but [NFS exports](#)
- IP address management is split into routed and interface addresses
- Reinstalling VPS on vpsAdminOS no longer deletes subdatasets and does not reset its configuration to the initial state, e.g. VPS features remain as they were.
- VPS features: bridge, iptables and NFS aren't configurable, they're always on.
- It is possible to change the network interface name within the VPS in VPS details page.

## Staging environment

In order for all members to test VPS on vpsAdminOS, we've created so called staging environment. It's similar to playground, where everyone can create a VPS. When creating a VPS, just select location **Staging** and deselect **Keep platform**. The VPS will be created on a vpsAdminOS node.

It's terms of use are similar to [playground VPS](#), only it can be a bit rougher, like unplanned outages and reboots when we need to fix something. Everyone can use up to 8 CPUs, 4 GB RAM, 120 GB disk space, 4 public IPv4 addresses and 32 IPv6 /64 addresses. You can split these resources among 4 VPS.

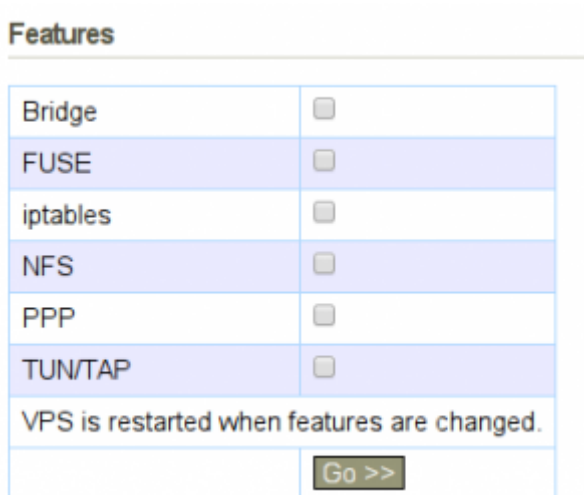
You can either create a new VPS or clone an existing production VPS. All mounts are removed when cloning, because NAS isn't accessible as of yet, see [user namespaces](#).

## Supported distributions

- Alpine 3.8, 3.9
- Arch
- CentOS 7, 8
- Debian 9, 10
- Fedora 29, 30
- Gentoo
- NixOS
- openSUSE Leap 15.1, Tumbleweed
- Slackware 14.2
- Ubuntu 16.04, 18.04
- Void Linux

## Features

Features can be turned on/off individually. When any change is made, the VPS restarts.



- Docker (experimental) - Enables support for Docker.
- FUSE - "Filesystem in Userspace" Enables the kernel module to allow non-privileged users create their own file systems.
- KVM - "Kernel-based Virtual Machine" Enables KVM for hardware support of virtualization.
- LXC nesting - "Linux Containers" Enables nested LXC containers.
- PPP - "Point-to-Point Protocol" Enables communications protocol used to establish a direct connection between point-to-point links.
- TUN/TAP - "TUN routing/TAP bridging" Enables the creation of virtual interfaces that are then

bridged.

We recommend only setting the features that your really need.

## More about vpsAdminOS

- <https://vpsadminos.org>
- <https://github.com/vpsfreecz/vpsadminos>
- IRC chat.freenode.net #vpsadminos

## Reporting bugs and ideas

Choose at your own discretion:

- IRC: #vpsfree and #vpsadminos on chat.freenode.net
- [podpora@vpsfree.cz](mailto:podpora@vpsfree.cz)
- vpsAdminOS issues: <https://github.com/vpsfreecz/vpsadminos/issues>
- vpsAdmin issues: <https://github.com/vpsfreecz/vpsadmin/issues>

From:

<https://kb.vpsfree.org/> - **Knowledge Base**

Permanent link:

<https://kb.vpsfree.org/manuals/vps/vpsadminos>

Last update: **2020/04/11 00:44**