

NixOS

NixOS allows declarative configuration management of the whole system and deployed services.

To start using NixOS choose **[vpsAdminOS] NixOS** template.

If you have set your public SSH keys in vpsAdmin the key will be automatically deployed and you can login as root right away. To be able to login with a password you need to set the password manually after your container is created.

Initial setup & informations

After installing NixOS (mainly talking about stable release) you **have to know**, that there is activated firewall by default blocking everything except ssh (port no. 22) so if you want to run some webserver (for example nginx) you have to tell firewall to not block your ports with this in your `/etc/nixos/configuration.nix:nixos`

```
networking.firewall.allowedTCPPorts = [ 80 443 ];
```

Then dont forget to do

```
nixos-rebuild switch
```

so your changes will take effect (without rebooting)

Configuration

Configuration is available at

```
/etc/nixos/configuration.nix
```

This can be altered to change container configuration on the fly or for next reboot. Try changing container hostname and running

```
nixos-rebuild switch
```

to build new configuration and directly switch to it. To switch to new configuration after reboot (set as a default boot profile) use

```
nixos-rebuild boot
```

If you are using channels you can update your system by running

```
nixos-rebuild switch --upgrade
```

Package installation

Packages can be installed system wide via `configuration.nix` or for local profile via

```
nix-env
```

To install vim use

```
nix-env -iA nixos.vim
```

To search for a package use

```
nix-env -qaP postgre
```

Manual

Consult NixOS manual for more <https://nixos.org/nixos/manual/>

Legacy OpenVZ templates

Due to compatibility issues with OpenVZ it is no longer recommended to use the old infrastructure for new deployments.

Common issues

SSH public key deployment fails

This can occur on first boot of the template when there's no `/root` yet. Disable automatic key deployment and deploy key manually **after** the containers boot.

error: while setting up the build environment: unable to load seccomp BPF program: Invalid argument

Nix \geq 1.11.10 requires seccomp, which is not available on vpsFree. To fix this, pin Nix to 1.11.9 by adding the following option to `configuration.nix` and rebuilding the system:

```
nix.package = (import (pkgs.fetchFromGitHub {
  owner = "NixOS";
  repo = "nixpkgs";
  rev = "300fa462b31ad2106d37fcdb4b504ec60dfd62aa";
  sha256 = "1cbjmi34ll5xa2nafz0jlscliivj62mq78qr3zl4skgdk6scl328s";
}) {}).nix;
```

In the future, it will be possible to [disable seccomp usage](#).

systemd 233+ compatibility issues

Causes boot failures, workaround is pinning systemd to version 232:

```
nixpkgs.config.packageOverrides = super:
  let systemdGperfCompat = super.systemd.override { gperf =
super.gperf_3_0; };
  in {
    systemd = systemdGperfCompat.overrideAttrs ( oldAttrs: rec {
      version = "232";
      name = "systemd-${version}";
      src = pkgs.fetchFromGitHub {
        owner = "nixos";
        repo = "systemd";
        rev = "66e778e851440fde7f20cff0c24d23538144be8d";
        sha256 =
"1valz8v2q4cj0ipz2b6mh5p0rjxpy3m88gg9xa2rcc4gcmscndzk";
      };
    });
  };
```

Details

Log contents `Failed to canonicalize path /etc/systemd/system/local-fs.target.d: Too many levels of symbolic links`

```
$ journalctl -o verbose
```

```
_EXE=/nix/store/n5ksbh1hx275zfbqbalhghzpxplw73lf-
systemd-234/lib/systemd/systemd
_CMDLINE=/run/current-system/systemd/lib/systemd/systemd --system --
deserialize 19
CODE_FILE=src/shared/dropin.c
CODE_LINE=133
CODE_FUNC=unit_file_find_dir
ERRNO=40
MESSAGE=Failed to canonicalize path /etc/systemd/system/local-fs.target.d:
Too many levels of symbolic links
```

From:
<https://kb.vpsfree.cz/> - Znalostní Báze

Permanent link:
<https://kb.vpsfree.cz/navody/distribuce/nixos>

Last update: **2019/03/11 12:29**

