

Immutable Linux

What if you cannot write to /usr

At 15.12.2021 by Daniel Schier

Agenda

- Introduction
- Immutability
- Examples
 - CoreOS (Red Hat/Fedora)
 - microOS (Suse Linux)
 - k3os (Rancher/Suse Linux)
 - NixOS
- Conclusion

Speaker

Daniel Schier

Founder of while-true-do.io
and DDOSUG, with a huge heart
for Open Source.

Making Open Source more
accessible is my passion.

And... I like penguins, cats
and cookies.



Immutability

- /usr Read-Only?
- /etc often not
- Very FS dependent
- More secure?
- More stable?
- More solid?
- Build > Configure

Examples

- CoreOS
- microOS
- k3os
- NixOS

CoreOS

- Fedora
 - IoT, CoreOS, Silverblue, Kinoite
- RHEL
 - CoreOS
- OSTree based
- Deploy, Pin, Rollback
- Ignition for unattended installations
- Mutable and Immutable parts
- Podman and Toolbox (and Docker)

<https://getfedora.org/coreos?stream=stable>



OFFICIAL
EDITION

Fedora IoT provides a trusted open source platform as a strong foundation for IoT ecosystems.

[Learn more.](#)

[Download Now](#)

Emerging Fedora Editions

Preview the future of Fedora.



fedora
COREOS

Fedora CoreOS is an automatically updating, minimal, container-focused operating system.

[Learn more.](#)



fedora
SILVERBLUE

Fedora Silverblue is an immutable desktop operating system aimed at good support for container-focused workflows.

[Learn more](#)

microOS

- Suse only
- Btrfs based
- Deploy, snapshot, rollback
- Pretty default Suse Linux
- Combustion, ignition and cloud-init for initial deployment
- Built with Kubic in mind (Kubernetes distribution)

<https://microos.opensuse.org/>

<https://kubic.opensuse.org/>



openSUSE Kubic

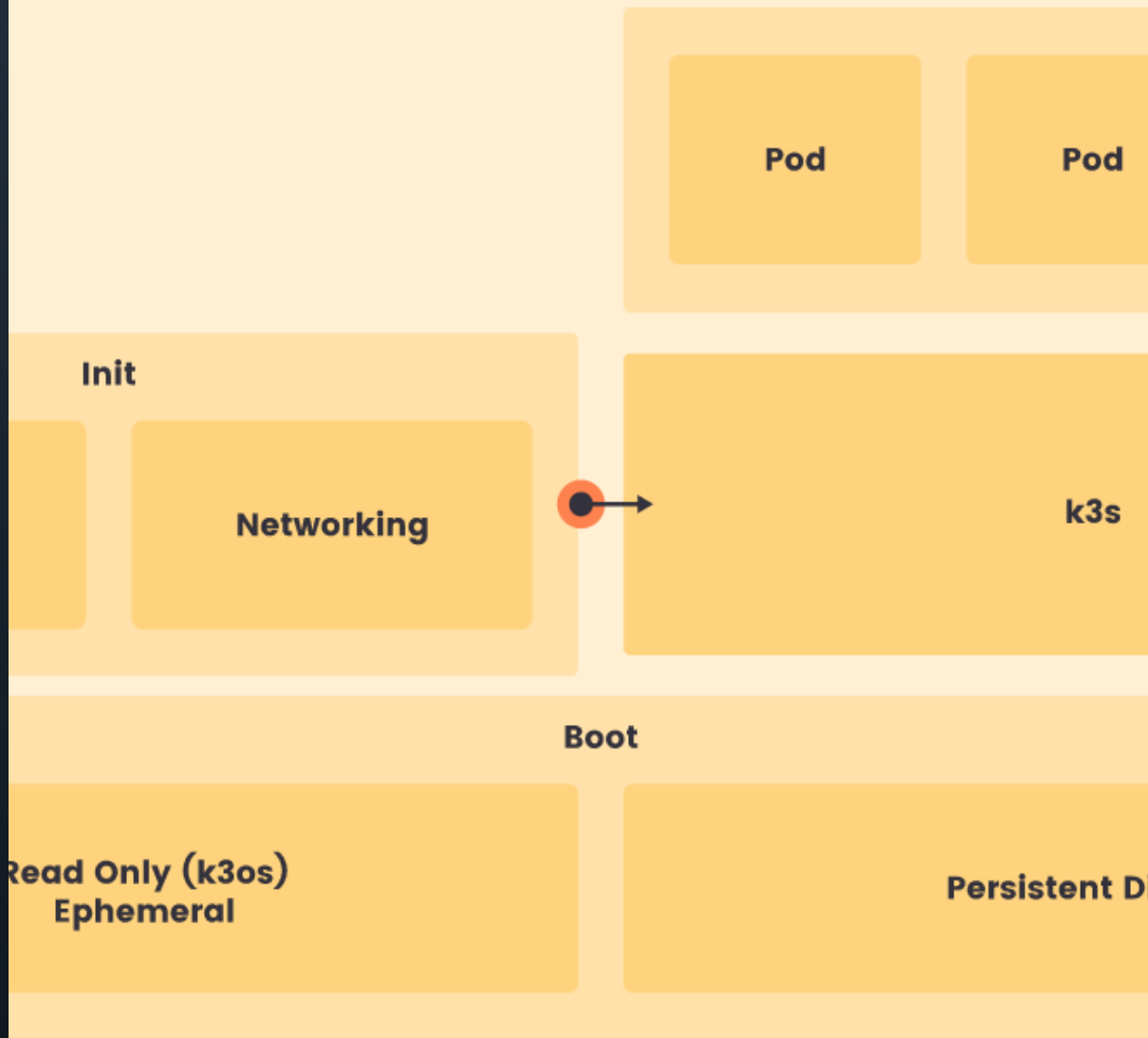
Certified Kubernetes distribution &
container-related technologies built by the
openSUSE community

k3os

- Made by Rancher/Suse
- Built for k3s (Kubernetes distribution)
- k3s is PID 1
- Based on Alpine + Ubuntu Kernel
- No traditional package manager
- Kubernetes is the frontend
- Configuration and updates done via CRD

<https://k3os.io/>

<https://k3s.io/>



NixOS

- Independent
- Declarative via JSON config
- Nix+NixShell → NixOS
- Designed for container/cloud/ISO abstraction
- Every (manual) installation is a container, but somewhat hidden
- You can start on your own machine

<https://nixos.org/>

Reproducible builds and deployments.

Nix is a tool that takes a unique approach to package management and system configuration. Learn how to make reproducible, declarative and reliable systems.

Download

Get started

Conclusion

- Tested all of them
- Worked with 3 of them (4 tbd)
- Solving some issues
- Applications not ready
- Some hoops and loops here and there
- CoreOS seems very mature, followed by k3s
- None of them are ready:
 - “critical production workload”
 - being used by general Linux operators
- All of them
 - Very cloud and container focussed
 - Different ideas with the same goal

The future of Linux is immutable, but my
answers and opinions not. Please don't
hesitate :)