Building Fedora CoreOS at Nest with Fedora 2022

Renata Ravanelli - Red Hat
Senior Software Engineer

✉ renata.ravanelli@gmail.com
# ravanelli on libera.chat

Saqib Ali - Red Hat
Software Engineer Intern

✉ saqibsali00@gmail.com
# saqali on libera.chat

Today's Talk

- History
- What is Fedora CoreOS?
- Red Hat CoreOS Vs Fedora CoreOS
- Multiple Update Streams
- Why learn how FCOS is built?
- Build Process
- Components of the “config”
- CoreOS-Assembler (cosa)
- Overrides and New Packages
- What about testing?
- How do we deliver FCOS?
- Demos
- Challenge!
- Get involved!
- Your Questions!

renata.ravanelli@gmail.com saqibsali00@gmail.com

History

fedora COREOS = container linux + PROJECT ATOMIC fedora

Came from

Fedora, Container Linux and Atomic host

Container Linux

Philosophy: automatic updates

Provisioning Stack,

Immutable infra

Cloud Native Expertise

Atomic Host

Fedora Foundation, base OS and its structure such as packages and Kernel

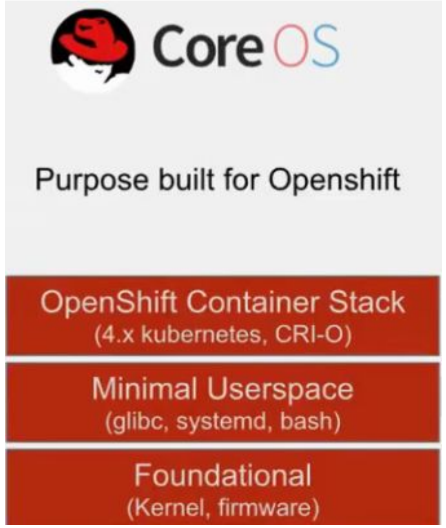
Update Stack

Selinux Enhanced Security

What is Fedora CoreOS?

Fedora CoreOS is an automatically-updating, minimal operating system for running containerized workloads securely and at scale. It is currently available on multiple platforms, with more coming soon.

Red Hat CoreOS Vs Fedora CoreOS



CoreOS

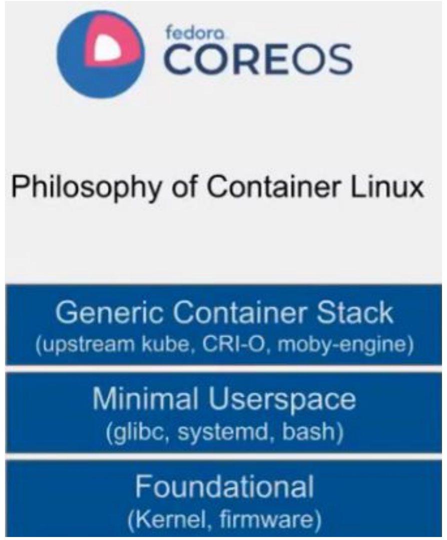
Purpose built for Openshift

OpenShift Container Stack
(4.x kubernetes, CRI-O)

Minimal Userspace
(glibc, systemd, bash)

Foundational
(Kernel, firmware)

Based on RHEL;
Is only meant to be used with OpenShift;
Red Hat CoreOS is not a standalone OS, it is a component of Openshift;
Automated provisioning via ignition;
SELinux enforcing;
Updates and configuration controlled by cluster operator;
RPM-OStree technology



fedora COREOS

Philosophy of Container Linux

Generic Container Stack
(upstream kube, CRI-O, moby-engine)

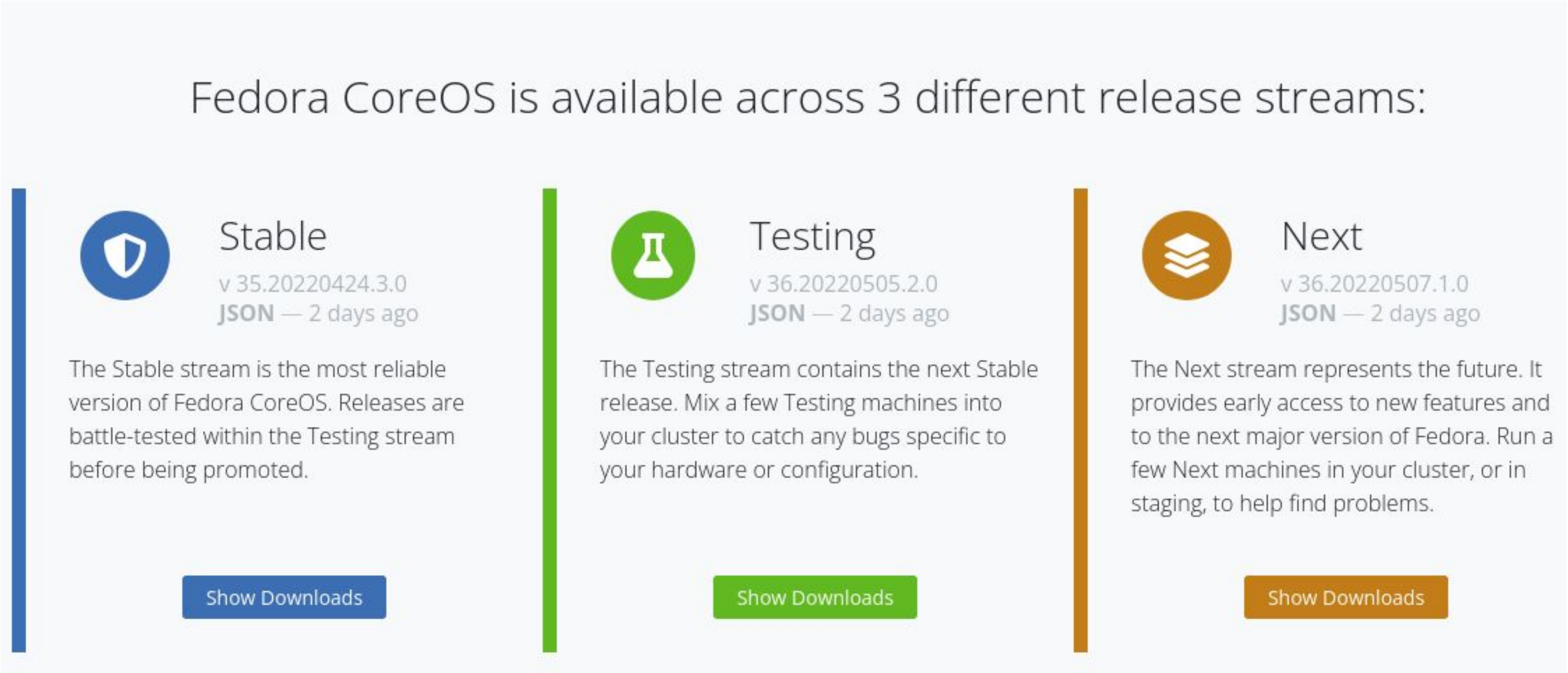
Minimal Userspace
(glibc, systemd, bash)

Foundational
(Kernel, firmware)

Based on Fedora;
Standalone OS with automatic updates (Reliable updates);
Automated provisioning via ignition;
SELinux enforcing;
Podman or moby engine container runtimes by default;
Can work as part of a cluster with OKD;
Share components and tooling with RHEL CoreOS;
RPM-OStree technology

renata.ravanelli@gmail.com saqibsali00@gmail.com

Multiple Update Streams



Fedora CoreOS is available across 3 different release streams:

Stable

v 35.20220424.3.0

JSON — 2 days ago

The Stable stream is the most reliable version of Fedora CoreOS. Releases are battle-tested within the Testing stream before being promoted.

Testing

v 36.20220505.2.0

JSON — 2 days ago

The Testing stream contains the next Stable release. Mix a few Testing machines into your cluster to catch any bugs specific to your hardware or configuration.

Next

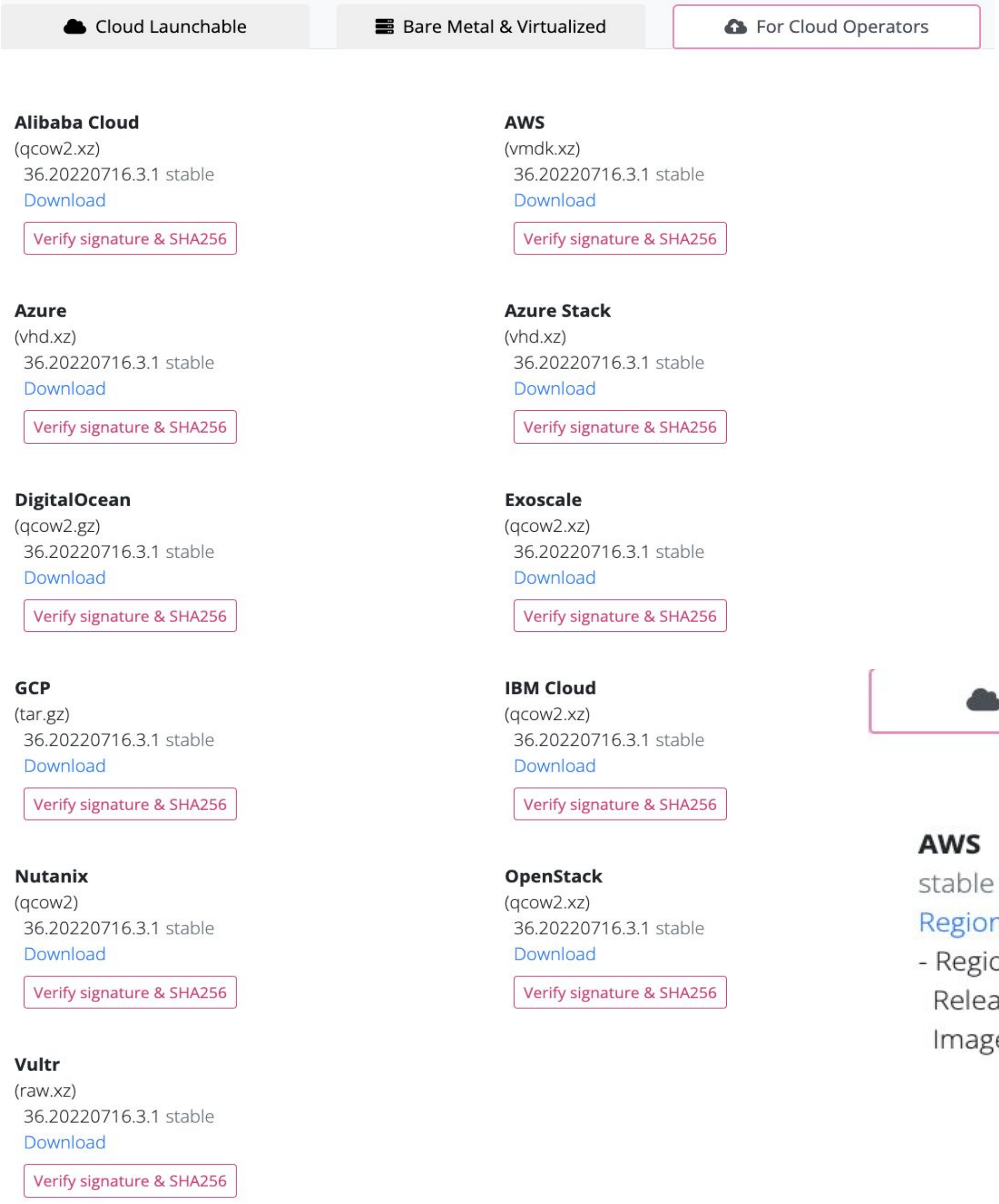
v 36.20220507.1.0

JSON — 2 days ago

The Next stream represents the future. It provides early access to new features and to the next major version of Fedora. Run a few Next machines in your cluster, or in staging, to help find problems.

✉ renata.ravanelli@gmail.com ✉ saqibsali00@gmail.com

Multi-arch



Alibaba Cloud
(qcow2.xz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

AWS
(vmdk.xz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

Azure
(vhd.xz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

Azure Stack
(vhd.xz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

DigitalOcean
(qcow2.gz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

Exoscale
(qcow2.xz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

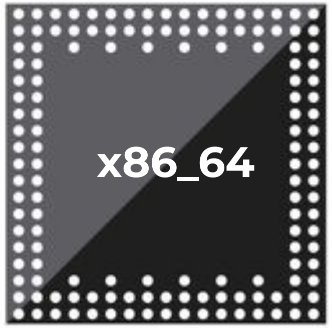
GCP
(tar.gz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

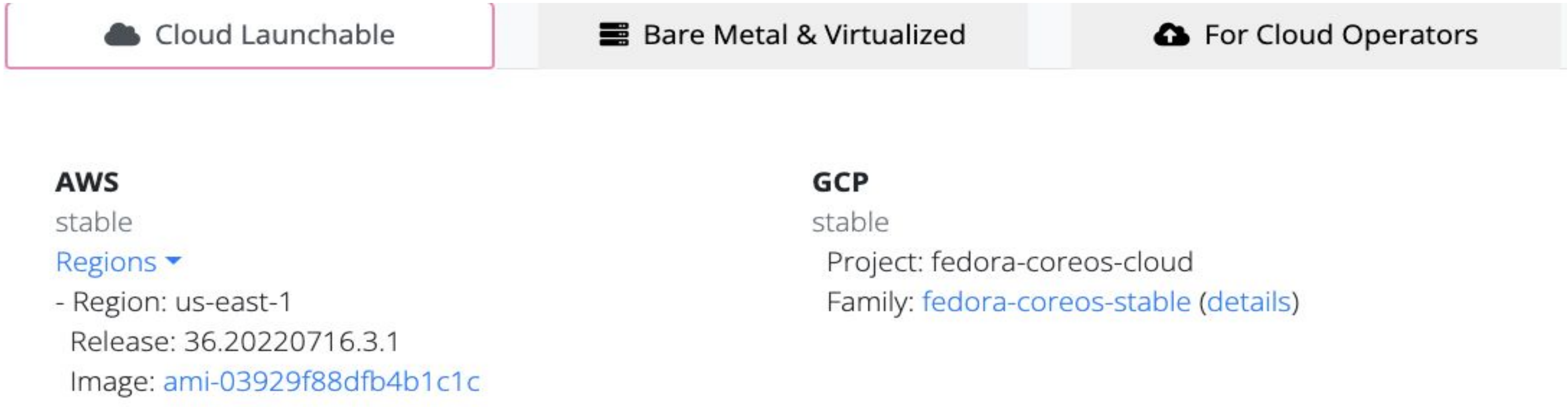
IBM Cloud
(qcow2.xz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

Nutanix
(qcow2)
36.20220716.3.1 stable
Download
Verify signature & SHA256

OpenStack
(qcow2.xz)
36.20220716.3.1 stable
Download
Verify signature & SHA256

Vultr
(raw.xz)
36.20220716.3.1 stable
Download
Verify signature & SHA256





AWS
stable
Regions
- Region: us-east-1
Release: 36.20220716.3.1
Image: ami-03929f88dfb4b1c1c

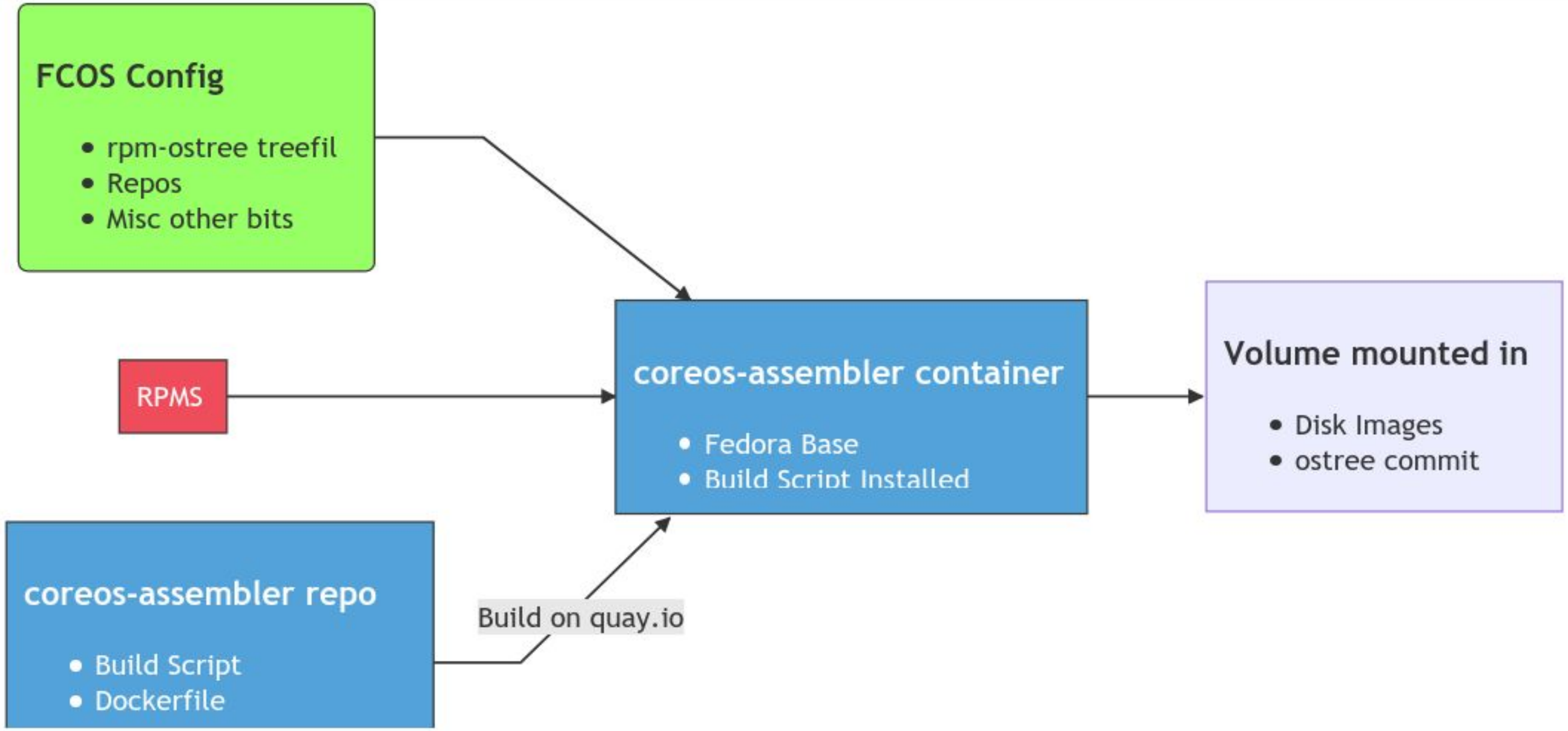
GCP
stable
Project: fedora-coreos-cloud
Family: fedora-coreos-stable (details)

renata.ravanelli@gmail.com saqibsali00@gmail.com

Why learn how FCOS is built?

- Build FCOS yourself!
- Build a custom FCOS-like OS
- Learn about the components that make up FCOS

Build Process



FCOS Config

- rpm-ostree treefil
- Repos
- Misc other bits

RPMS

coreos-assembler container

- Fedora Base
- Build Script Installed

Volume mounted in

- Disk Images
- ostree commit

coreos-assembler repo

- Build Script
- Dockerfile

Build on quay.io

CoreOS-Assembler (cosa)

tooling to build the OS

- containerized collection of tools used to build FCOS-style systems
- serves both local development and production level build systems
- Built images found at *quay.io/coreos-assembler/coreos-assembler*

Components of the “config”

the build schema

manifest.yaml

tells RPM-OSTree how to generate OSTree commits with list of RPMS

overlay.d/

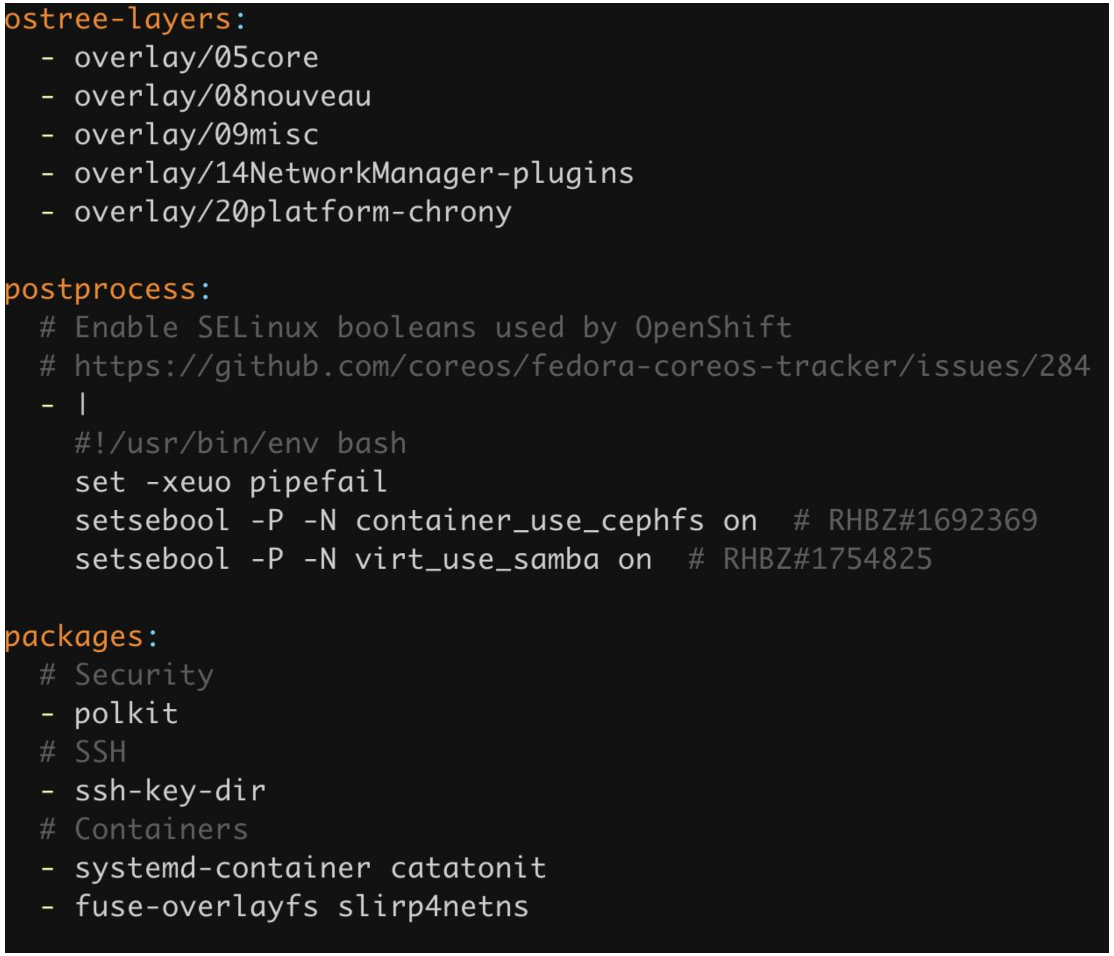
additional information that is layered in the OSTree commit

image.yaml

configuration of the final disk images

manifest.yaml (treefile)

Components of the “config”



```
ostree-layers:
  - overlay/05core
  - overlay/08nouveau
  - overlay/09misc
  - overlay/14NetworkManager-plugins
  - overlay/20platform-chrony

postprocess:
  # Enable SELinux booleans used by OpenShift
  # https://github.com/coreos/fedora-coreos-tracker/issues/284
  - |
    #!/usr/bin/env bash
    set -xeuo pipefail
    setsebool -P -N container_use_cephfs on  # RHBZ#1692369
    setsebool -P -N virt_use_samba on  # RHBZ#1754825

packages:
  # Security
  - polkit
  # SSH
  - ssh-key-dir
  # Containers
  - systemd-container catatonit
  - fuse-overlayfs slirp4netns
```

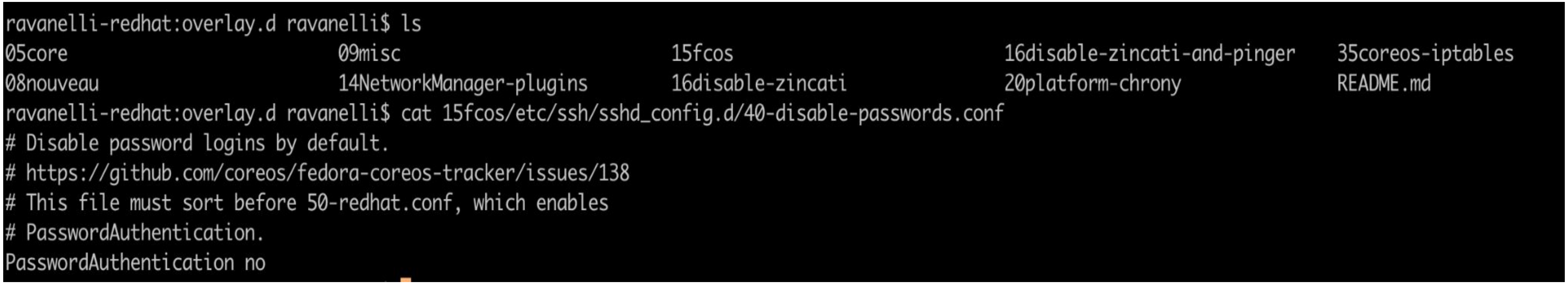
For generating OSTree commits, cosa uses manifest.yaml

It is a list of RPMs and a set of rpm-md repositories they come from. It also supports postprocess to make arbitrary changes

overlay.d/

Components of the “config”

- subdirectories are added to OSTree commit
- used to modify default configuration (ie. disable SSH passwords)

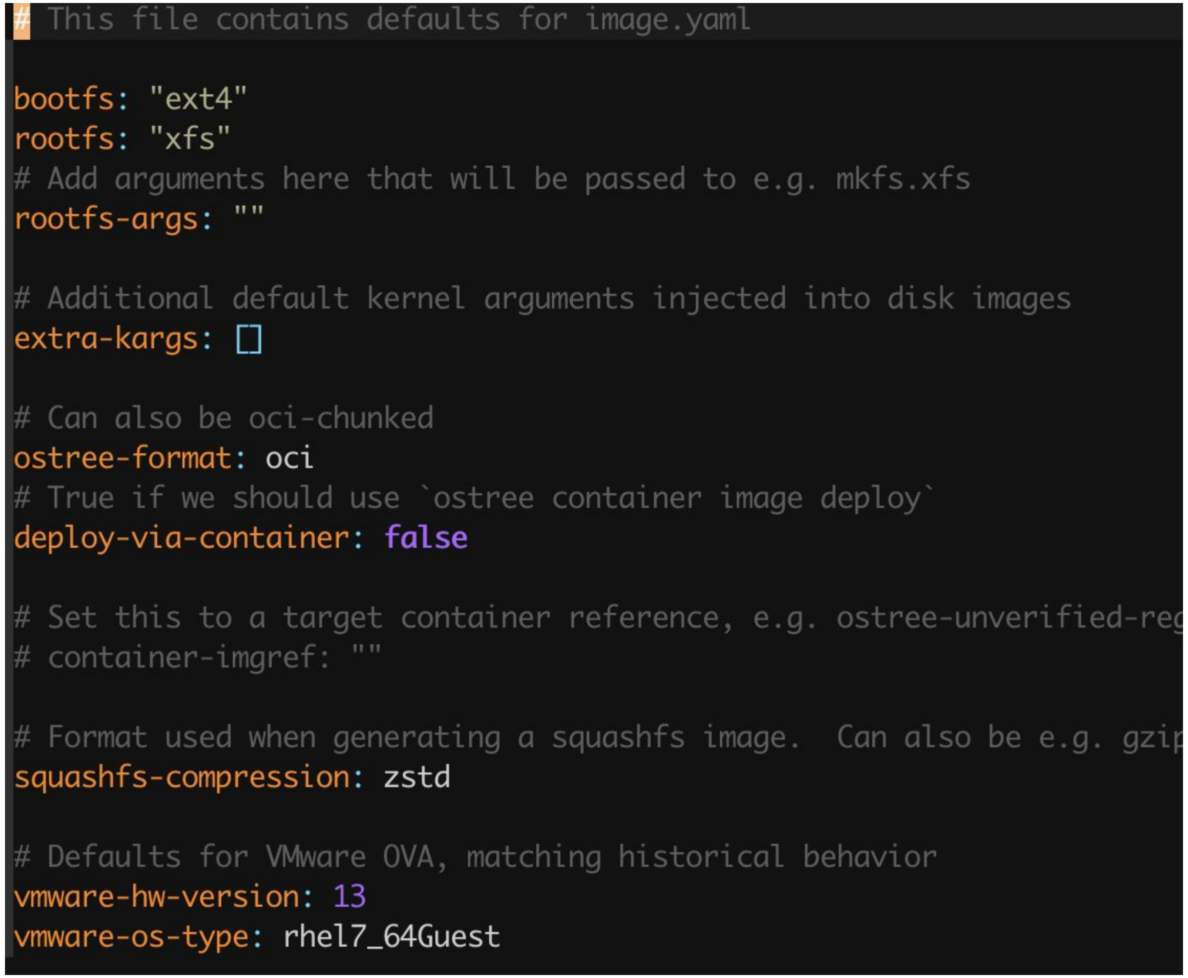


```
ravanelli-redhat:overlay.d ravanelli$ ls
05core                          09misc                          15fcos                          16disable-zincati-and-pinger    35coreos-iptables
08nouveau                       14NetworkManager-plugins        16disable-zincati               20platform-chrony               README.md
ravanelli-redhat:overlay.d ravanelli$ cat 15fcos/etc/ssh/sshd_config.d/40-disable-passwords.conf
# Disable password logins by default.
# https://github.com/coreos/fedora-coreos-tracker/issues/138
# This file must sort before 50-redhat.conf, which enables
# PasswordAuthentication.
PasswordAuthentication no
```

renata.ravanelli@gmail.com saqibsali00@gmail.com

image.yaml

Components of the “config”



```yaml
# This file contains defaults for image.yaml

bootfs: "ext4"
rootfs: "xfs"
# Add arguments here that will be passed to e.g. mkfs.xfs
rootfs-args: ""

# Additional default kernel arguments injected into disk images
extra-kargs: []

# Can also be oci-chunked
ostree-format: oci
# True if we should use `ostree container image deploy`
deploy-via-container: false

# Set this to a target container reference, e.g. ostree-unverified-reg
# container-imgref: ""

# Format used when generating a squashfs image.  Can also be e.g. gzip
squashfs-compression: zstd

# Defaults for VMware OVA, matching historical behavior
vmware-hw-version: 13
vmware-os-type: rhel7_64Guest
```

- Supports customization of disk images
- Provides default “opinionated” settings

✉ renata.ravanelli@gmail.com ✉ saqibsali00@gmail.com

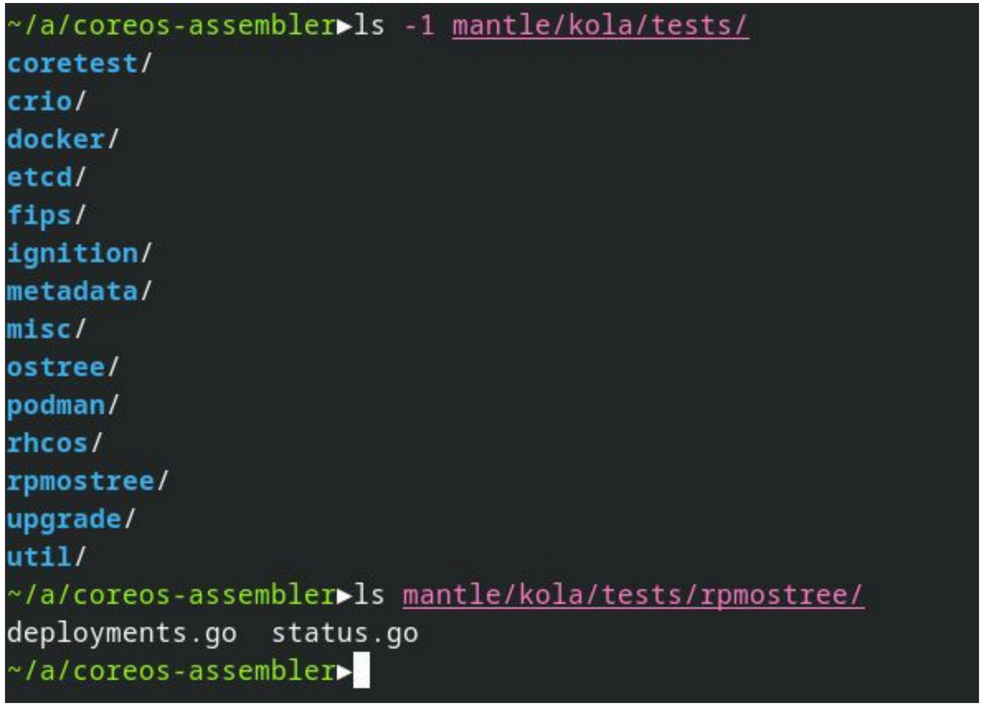
Overrides and New Packages

- Development Overrides:
  - In cosa you can do it via overrides. There are two subdirectories of overrides: **overrides/rootfs** and **overrides/rpm**
  - When you are hacking/testing a build that is the easier way to change and test packages and other configurations, especially because you won't need to care with repositories for example.
  - In fedora-coreos-config there is the **manifest-lock.overrides.yaml**. You can also override a package using the manifest file.

- New packages:
  Should be added in **manifest-lock.x86_64.json** for the specific architecture and **fedora-coreos-base.yaml**

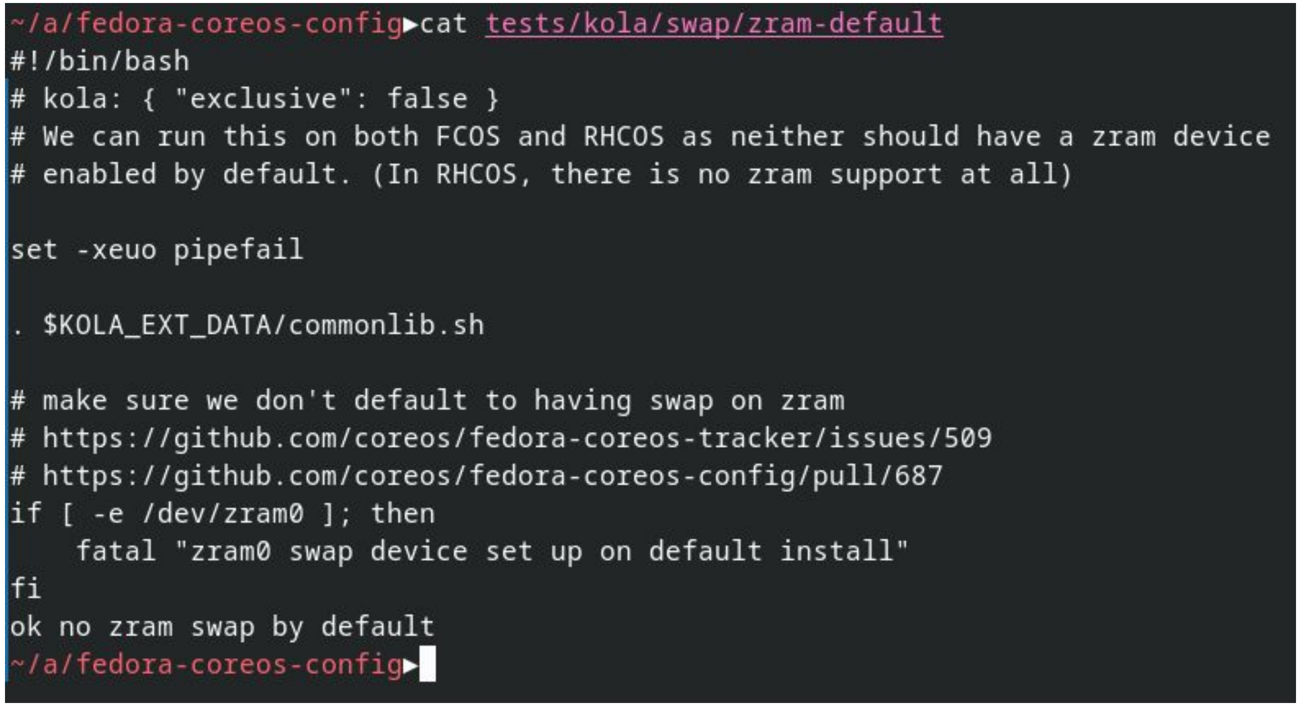
What about testing?

Compiled in Kola More complex tests written in Golang



```
~/a/coreos-assembler▶ls -1 mantle/kola/tests/
coretest/
crio/
docker/
etcd/
fips/
ignition/
metadata/
misc/
ostree/
podman/
rhcos/
rpmostree/
upgrade/
util/
~/a/coreos-assembler▶ls mantle/kola/tests/rpmostree/
deployments.go  status.go
~/a/coreos-assembler▶
```

External Tests Bash scripts that live alongside the config



```
~/a/fedora-coreos-config▶cat tests/kola/swap/zram-default
#!/bin/bash
# kola: { "exclusive": false }
# We can run this on both FCOS and RHCOS as neither should have a zram device
# enabled by default. (In RHCOS, there is no zram support at all)

set -xeuo pipefail

. $KOLA_EXT_DATA/commonlib.sh

# make sure we don't default to having swap on zram
# https://github.com/coreos/fedora-coreos-tracker/issues/509
# https://github.com/coreos/fedora-coreos-config/pull/687
if [ -e /dev/zram0 ]; then
    fatal "zram0 swap device set up on default install"
fi
ok no zram swap by default
~/a/fedora-coreos-config▶
```

renata.ravanelli@gmail.com saqibsali00@gmail.com

cosa kola

the testing framework

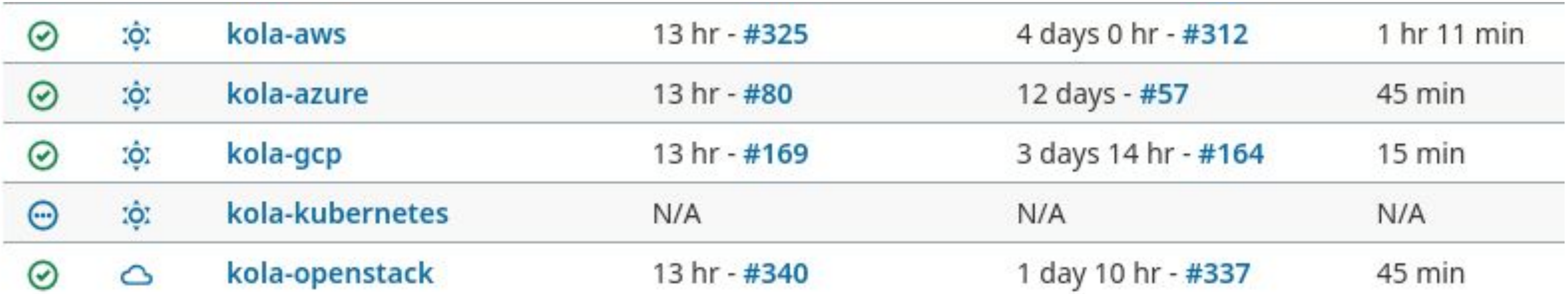
- Local testing with QEMU
- Supports testing with multiple cloud providers (AWS, Azure, GCP, OpenStack)
- Nifty features
  - ignition configs for each test
  - reboots
  - reruns and timeouts
  - put multiple tests in one VM
- As simple as “cosa kola run”

How do we deliver FCOS?

How are we able to ship 3 streams and 3 architectures every 2 weeks?

Investment in CI/CD

None of this happens without... Passing Tests!



✓	☀	kola-aws	13 hr - #325	4 days 0 hr - #312	1 hr 11 min
✓	☀	kola-azure	13 hr - #80	12 days - #57	45 min
✓	☀	kola-gcp	13 hr - #169	3 days 14 hr - #164	15 min
⋯	☀	kola-kubernetes	N/A	N/A	N/A
✓	☁	kola-openstack	13 hr - #340	1 day 10 hr - #337	45 min

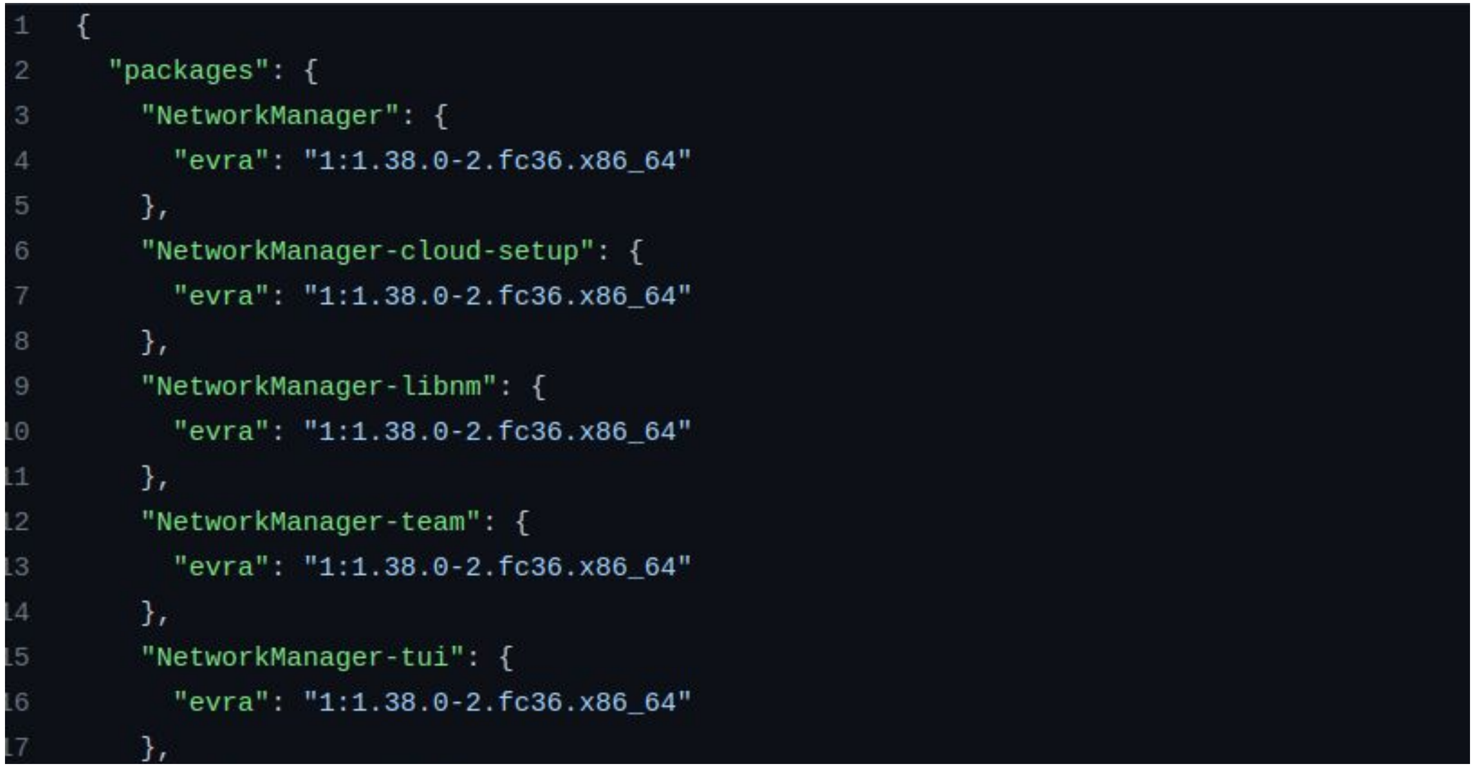
Versatile Tooling: cosa

- cosa container has all the tooling to build and test
  - building
  - testing
  - compressing builds
  - uploading build
  - editing build metadata
  - and so on....
- easy to launch tests on major cloud providers
- same cosa for local tests or production builds

Lockfiles

controlling package versions

- “lockfiles” allow us to control the package versions for each stream
- jenkins job “bumps” lockfile package versions after test run



```
{
  "packages": {
    "NetworkManager": {
      "evra": "1:1.38.0-2.fc36.x86_64"
    },
    "NetworkManager-cloud-setup": {
      "evra": "1:1.38.0-2.fc36.x86_64"
    },
    "NetworkManager-libnm": {
      "evra": "1:1.38.0-2.fc36.x86_64"
    },
    "NetworkManager-team": {
      "evra": "1:1.38.0-2.fc36.x86_64"
    },
    "NetworkManager-tui": {
      "evra": "1:1.38.0-2.fc36.x86_64"
    },
```

Overrides: Lockfiles

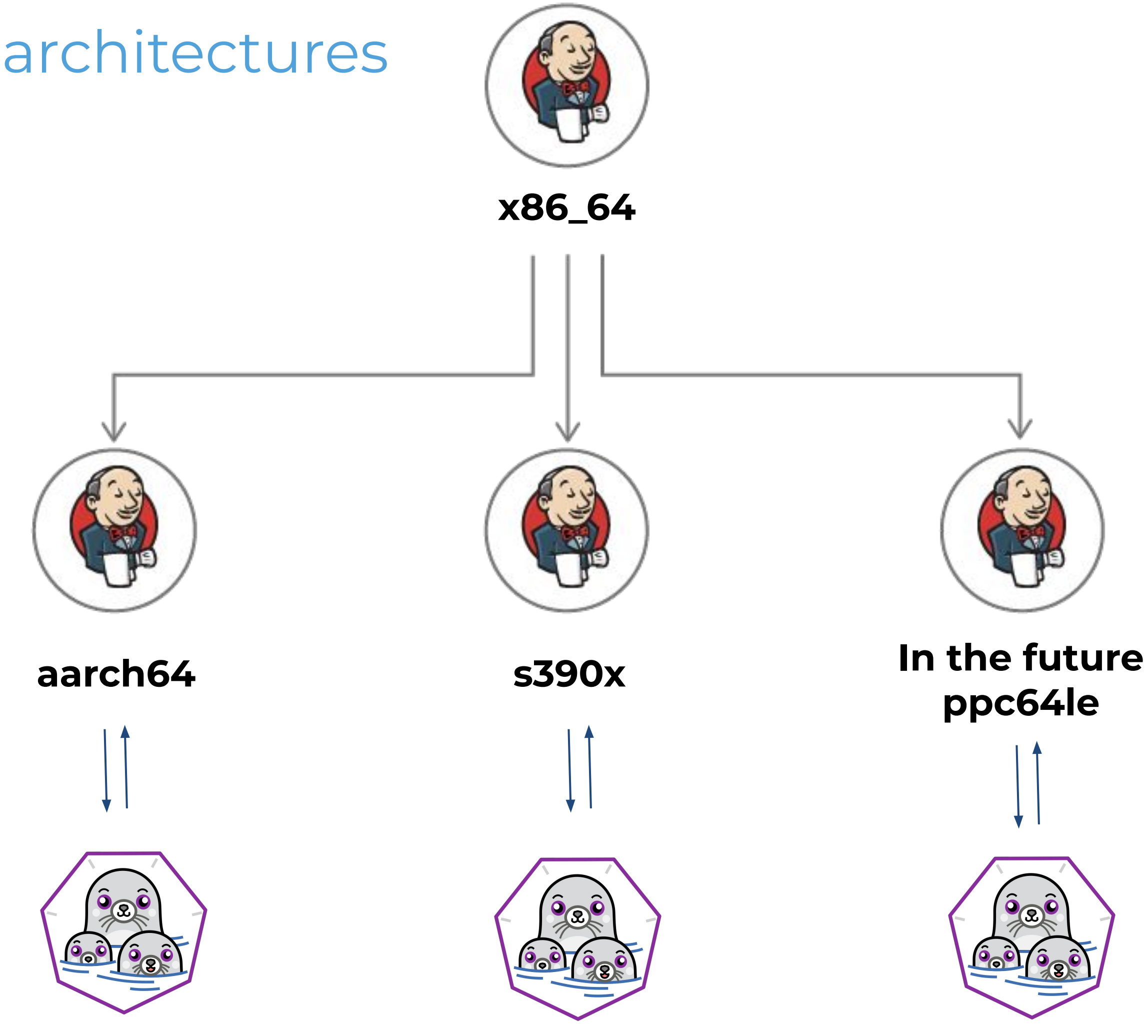
controlling package versions

- Lockfiles are flexible with overrides
- Override the latest available package version
  - Pinning: use an older version
  - Fast-track: use a new package not yet available through fedora stable channels

Multi-arch Builders

building multiple architectures

- After the build job for x86_64 passes it triggers the multi-arch builds
- One build job is created for each architecture
- Multi-arch builds are farmed out to individual nodes of that architecture running Fedora CoreOS
- We use podman remote to access other architectures. It means only one Jenkins instance is responsible for all builds
- Same process is done for all



renata.ravanelli@gmail.com   saqibsali00@gmail.com

Demos

Override Kernel packages via cosa and manifest.yaml

Demos

Adding a new package

Challenge! Try out Fedora CoreOS and join our Community!

Get involved!

- Web: https://getfedora.org/coreos
- Issues: https://github.com/coreos/fedora-coreos-tracker/issues
- Forum: https://discussion.fedoraproject.org/tag/coreos
- Docs: https://docs.fedoraproject.org/en-US/fedora-coreos/
- Mailing list: coreos@lists.fedoraproject.org
- IRC: libera.chat #fedora-coreos
- Matrix #coreos:fedoraproject.org

Go checkout the tutorials:

- https://docs.fedoraproject.org/en-US/fedora-coreos/tutorial-setup/

Questions

✉ renata.ravanelli@gmail.com ✉ saqibsali00@gmail.com

Thank you!

✉ renata.ravanelli@gmail.com ✉ saqibsali00@gmail.com