

Deploying Containers without Kubernetes

Clément Verna
Engineering Manager - Red Hat

 cverna@fedoraproject.org

 [@clemsverna](https://twitter.com/clemsverna)



Sometimes k8s is not the solution



- **Scalability**
- Clever updates
- On-prem or cloud

- Reduce sysadmin cost, infrastructure cost, time of people
- Reduce the time needed to make projects available (MVP)
- Version control (GitOps)

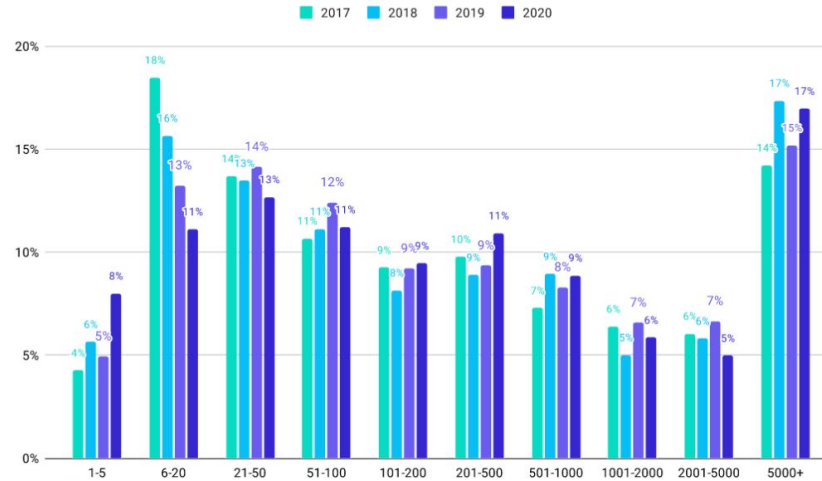


Containers in small infrastructures



From the 2020 CNCF Survey about 20% of respondent have a fleet of 20 or less.

On average, how many machines are in your fleet (including VM, bare metal, etc.)?



Clouds and IaaS



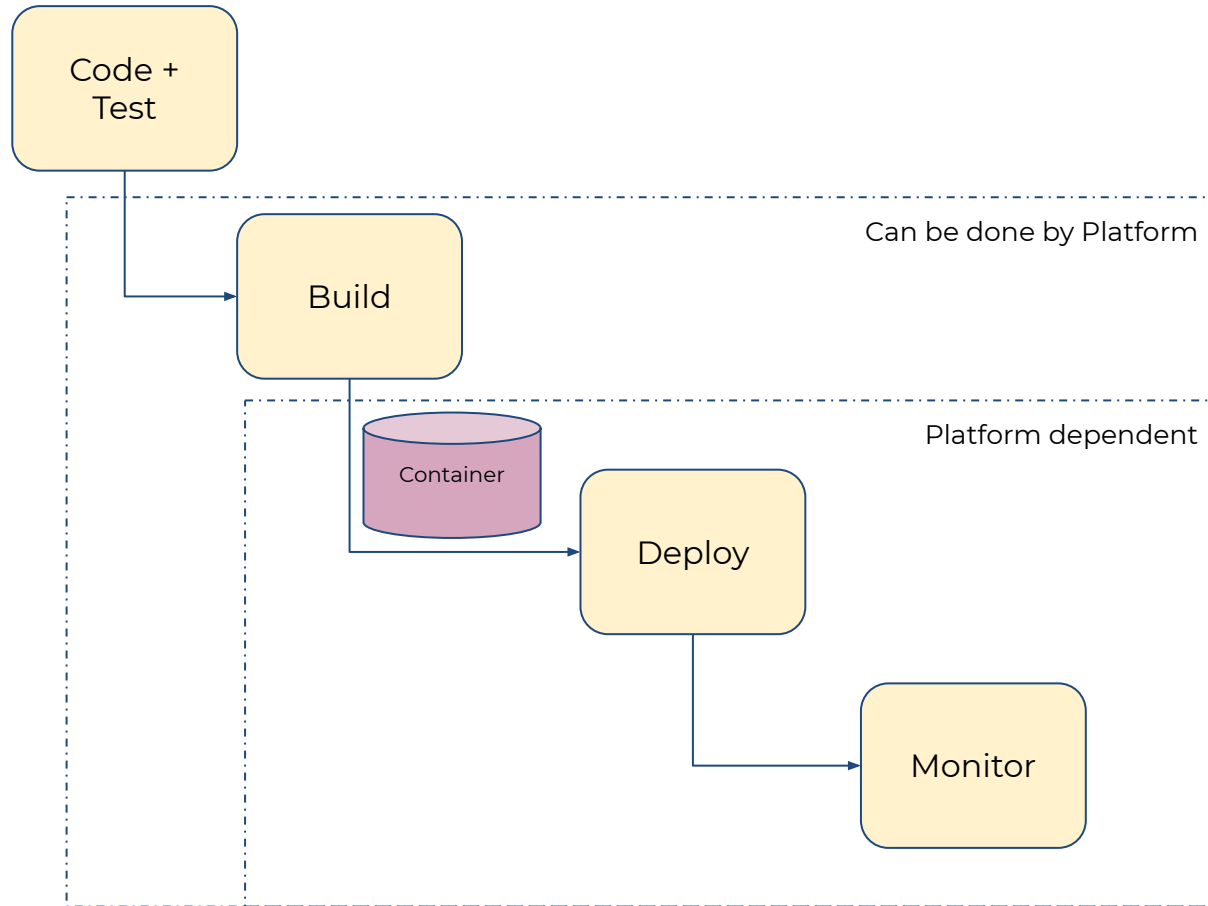
Services available to run containers

- Azure Container Instance
- AWS Fargate
- Google Cloud - Cloud Run

 No need to care about the Infrastructure and Operating systems

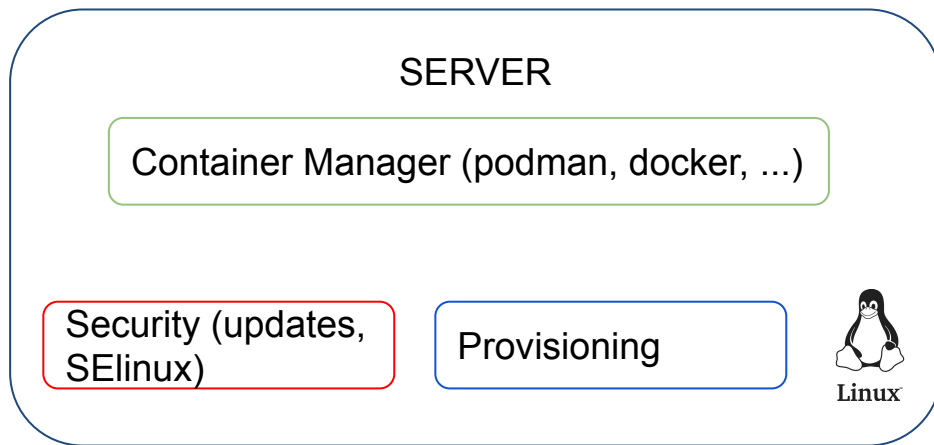
 Delivery pipeline strongly coupled to service provider

Clouds and IaaS - Workflow

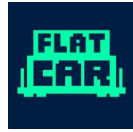


Alternatives ? What about Linux ?

Dedicated Linux solution focused on running containerized applications



Tiny Linux Ecosystem





- **Reduce sysadmin cost, infrastructure cost, time of people**
 - Automatic Updates (deploy it and forget about it)
- **Reduce the time needed to make projects available (MVP)**
 - Easy to deploy and provision (ready in less than a minute)
- **Version control (GitOps)**
 - Configuration as YAML/JSON
- **Clever updates**
 - 3 versions streams (next, testing and stable), update don't break
- **On-prem or cloud**
 - Currently available on 12 platforms (AWS, GCP, Azure, Digital Ocean, OpenStack, libvirt, VMware ..)



- Fedora CoreOS features Automatic Updates by default
 - Automatic updates → Reliable updates
 - Extensive tests in automated CI pipelines
 - Several update streams to preview what's coming
 - Users run various streams to help find issues
 - Managed upgrade rollouts over several days
 - Halt the rollout if issues are found
 - For when things go wrong
 - rpm-ostree rollback can be used to go back
 - future: automated rollback
 - based on user specified health checks



- Offered update streams with automatic updates
 - **next** - experimental features, Fedora major rebases
 - **testing** - preview of what's coming to stable
 - point in time snapshot of Fedora stable rpm content
 - **stable** - most reliable stream offered
 - promotion of testing stream after some bake time
- Goals
 - Publish new releases into update streams every two weeks
 - Find issues in next/testing streams before they hit stable



Automated Provisioning

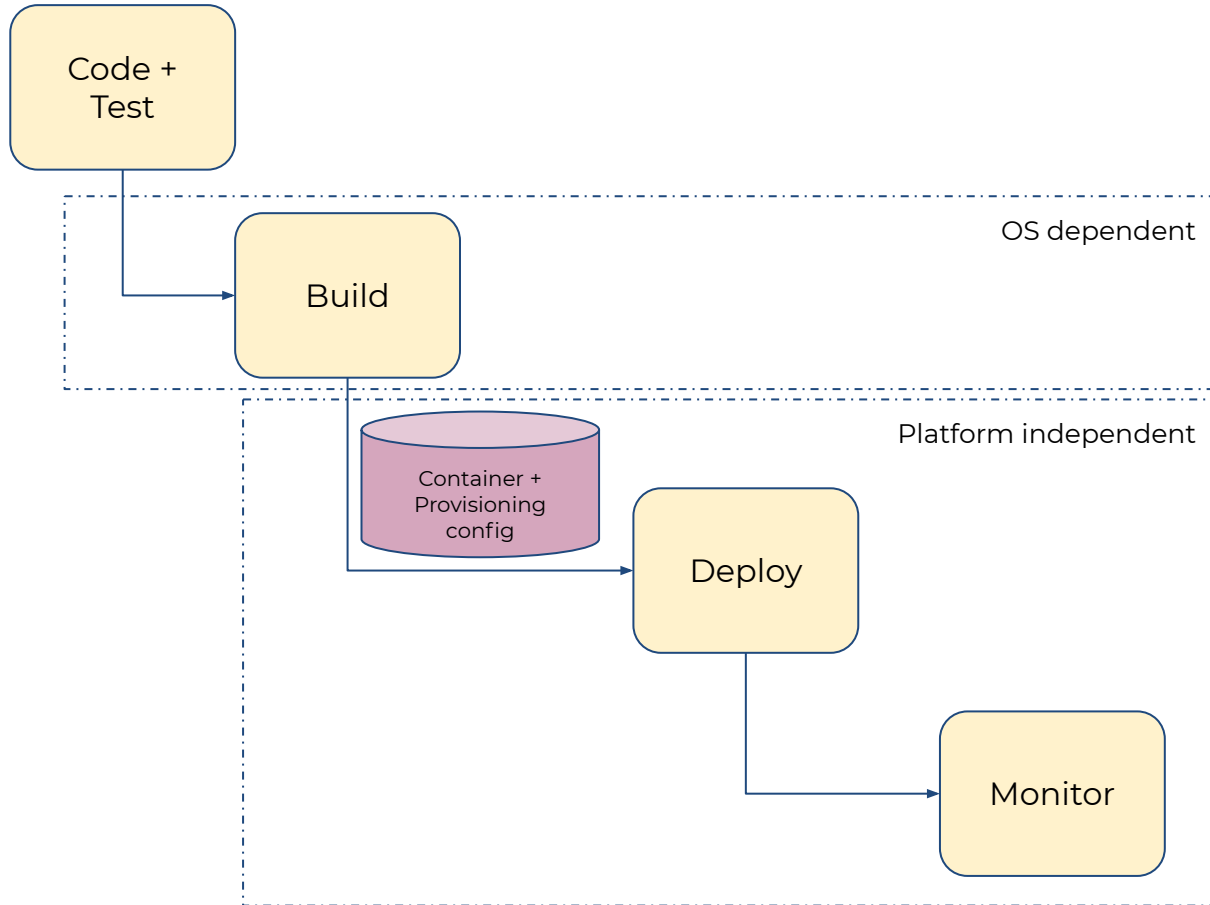
- Fedora CoreOS uses [Ignition](#) to automate provisioning
 - Any logic for machine lifetime is encoded in the config
 - Very easy to automatically re-provision nodes
 - Same starting point whether on bare metal or cloud
 - Use Ignition everywhere as opposed to kickstart for bare metal and cloud-init for cloud



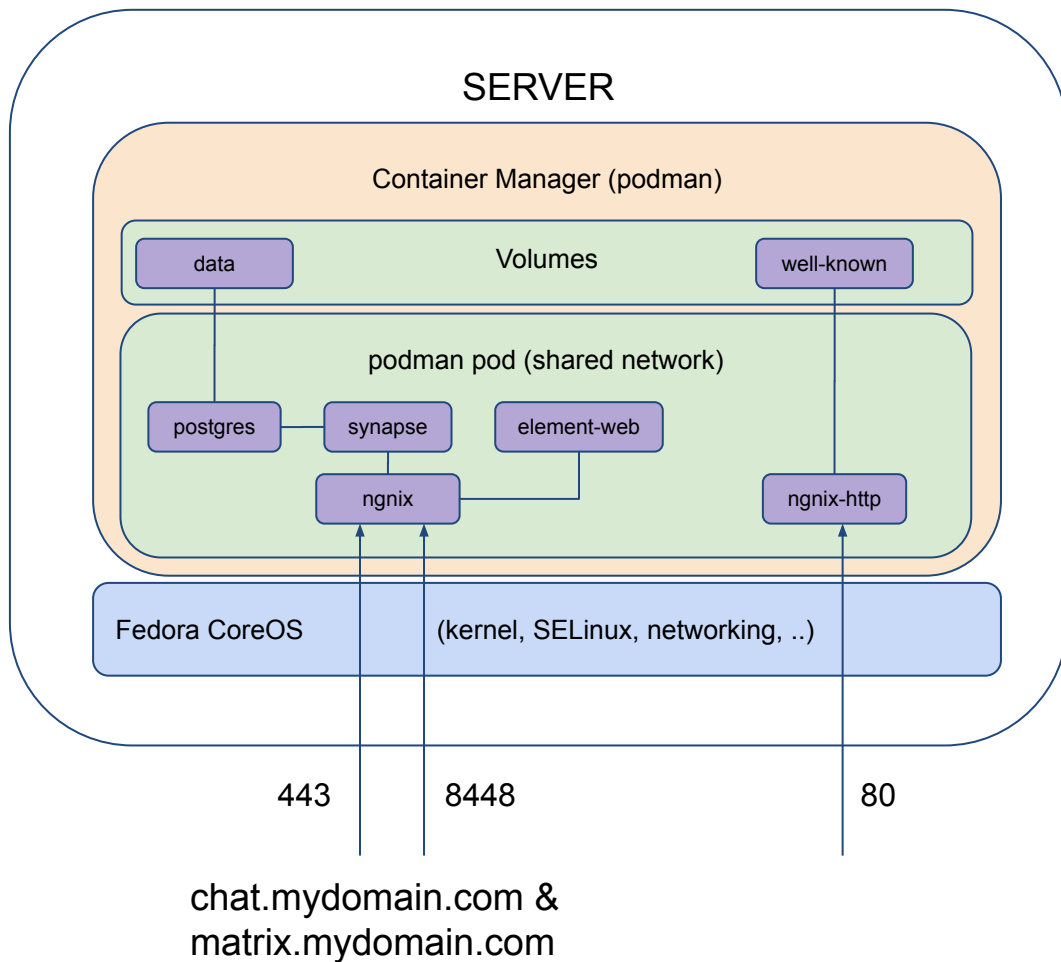
OS Versioning & Security

- Fedora CoreOS uses rpm-ostree technology
 - “Like git for your Operating System”
 - 32.20200615.2.0 - 86c0246
 - A single identifier tells you all software in that release
 - Uses read-only filesystem mounts
 - Prevents accidental OS corruption (rm -rf)
 - Prevents novice attacks from modifying system
- SELinux enforcing by default
 - Prevents compromised apps from gaining further access

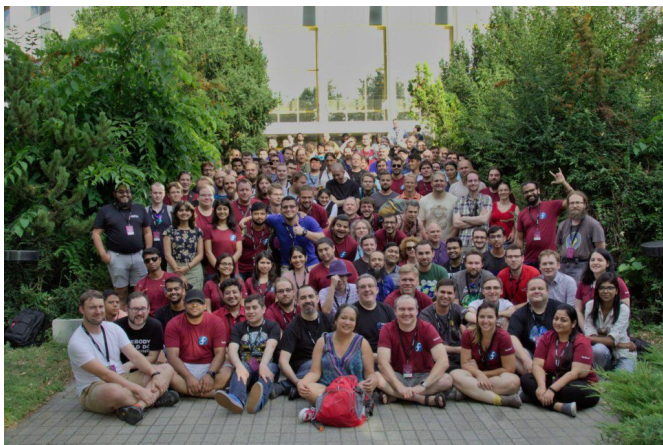
Container OS - Workflow



Example - Matrix Server



- Community





- Getting started
 - <https://docs.fedoraproject.org/en-US/fedora-coreos/>
- How to get in touch
 - Discourse :
<https://discussion.fedoraproject.org/c/server/coreos/5>
 - IRC: #fedora-coreos on freenode
 - GitHub : <https://github.com/coreos>
- [Getting Started with Fedora CoreOS - A Hands-on lab](#)

(Tomorrow)