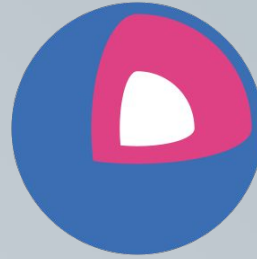# CNCF Member Webinar

CNCF has 99+ K8s distros, & this is how (and why) we built one more!

OKD4 on FCOS
September 2, 2020

Dusty Mabe
Charro Gruver
Christian Glombek
Vadim Rutkovsky

Agenda

What is OKD?

Operators Overview

The Machine Config Operator

What is Fedora CoreOS?

DEMO

Questions!

Red Hat

Agenda

What is OKD?

Operators Overview

The Machine Config Operator

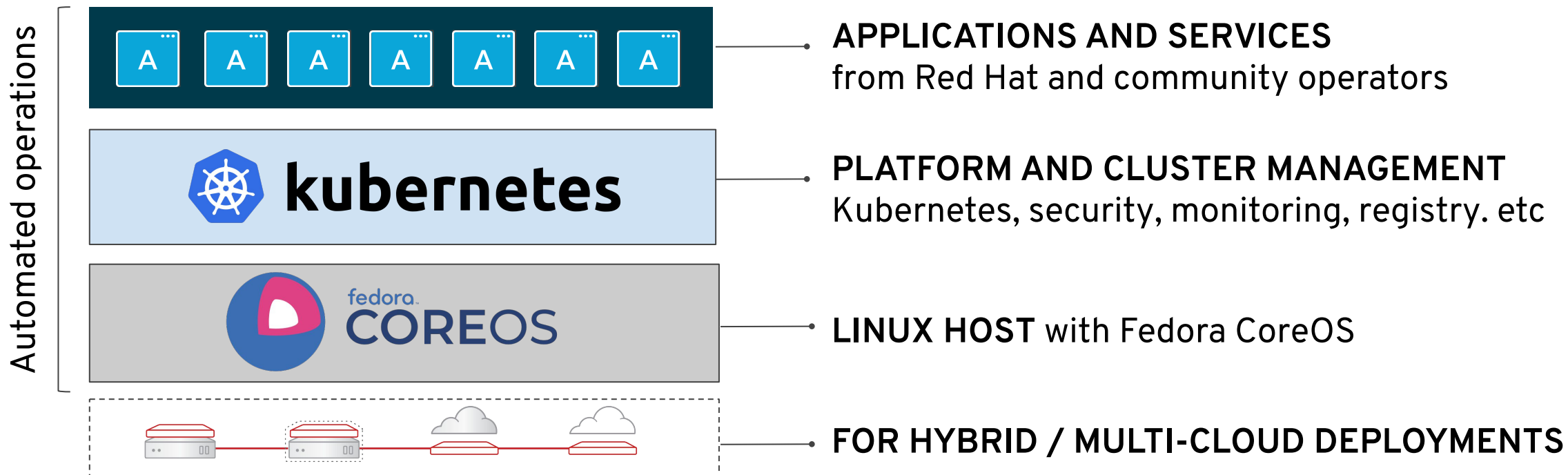What is Fedora CoreOS?

DEMO

Questions!

Red Hat

What is OKD?

A Community Distribution of Kubernetes

OpenShift codebase + Fedora CoreOS

okd.io

Red Hat

OKD 4: A Community Distribution of Kubernetes++

Automated installation, patching, and updates from the OS up

Automated operations

APPLICATIONS AND SERVICES
from Red Hat and community operators

PLATFORM AND CLUSTER MANAGEMENT
Kubernetes, security, monitoring, registry. etc

LINUX HOST with Fedora CoreOS

FOR HYBRID / MULTI-CLOUD DEPLOYMENTS

OKD 4: A Community Distribution of Kubernetes++

Automated installation, patching, and updates from the OS up

Automated operations

APPLICATIONS AND SERVICES
from Red Hat and community operators

PLATFORM AND CLUSTER MANAGEMENT
Kubernetes, security, monitoring, registry. etc

LINUX HOST with Fedora CoreOS

FOR HYBRID / MULTI-CLOUD DEPLOYMENTS

Agenda

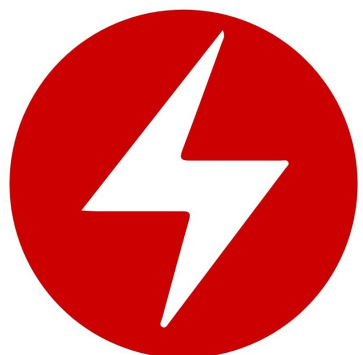What is OKD?

Operators Overview

The Machine Config Operator

What is Fedora CoreOS?

DEMO

Questions!

Red Hat
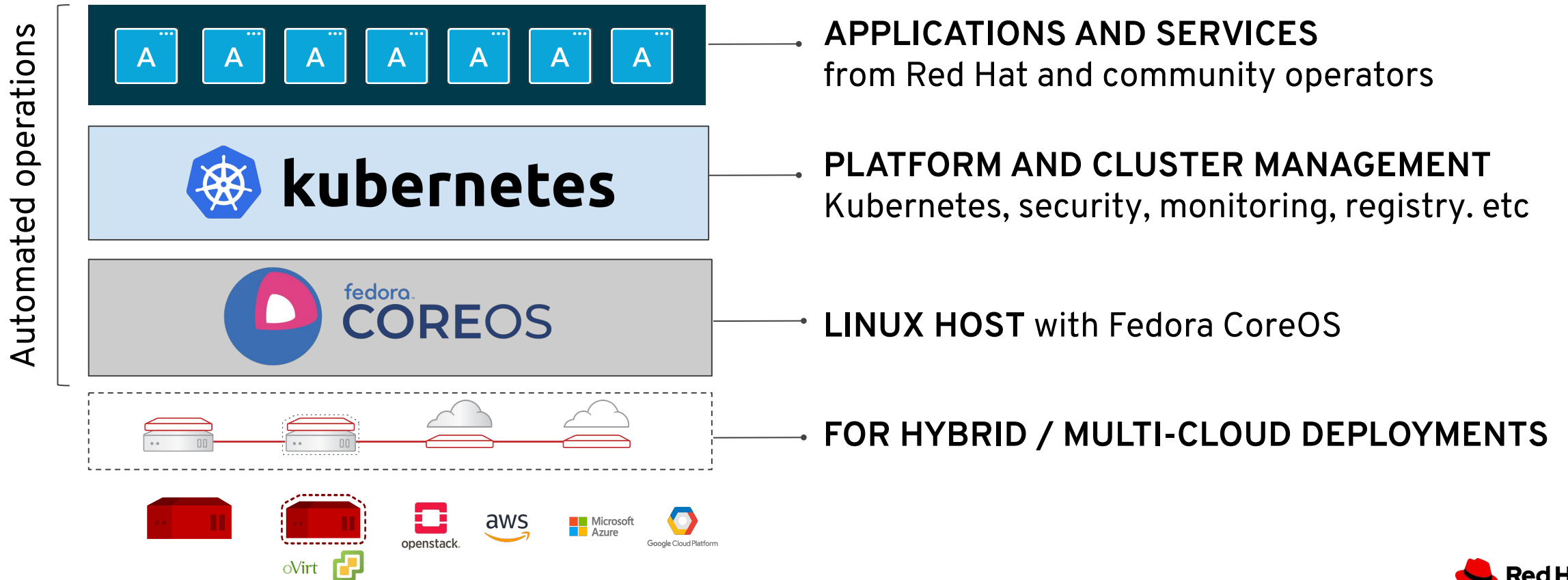
Operator Pattern

OPERATOR SDK

Operators are a method of packaging, deploying, and

managing a Kubernetes application.

Operators are controlled via Custom Resources (CR).

Red Hat

Operators all the way down

Cluster Version Operator
    Ensures all top-level operators are present

Kube-apiserver, kube-controller-manager, kube-scheduler, etcd
    These operators ensure core Kubernetes components are configured
Network
    Ensures CNI plugins are installed and SDN is configured

Image Registry - ensures internal registry is set up
Monitoring - ensures all component metrics are collected and displayed
Ingress - ensures router is setup
Storage - ensures CSI plugins are installed and storageclasses exist

OperatorHub

OperatorHub is a community-sourced index of optional operators, i.e:

Grafana, KEDA, Strimzi, Argo CD, Kubefed, OpenEBS, KubeVirt etc.

Operator Lifecycle Manager (OLM) takes care of operator scope (cluster-wide or namespace only), ensures it can be updated manually and manages permissions to use and install operators.

OperatorHub is integrated in OpenShift console, so developers can install operators via self-service interface.

Red Hat

Agenda

What is OKD?

Operators Overview

The Machine Config Operator

What is Fedora CoreOS?

DEMO

Questions!

Red Hat

The Machine Config Operator

Core OpenShift Operator

Manages machine configuration customization

Consumes and renders desired state as Ignition config

Reconciles all Machines in Cluster with desired state

Applies OS Updates with rpm-ostree

The MachineConfig CRD

```go
// MachineConfig defines the configuration for a machine
type MachineConfig struct {
    metav1.TypeMeta   `json:",inline"`
    metav1.ObjectMeta `json:"metadata,omitempty"`

    Spec MachineConfigSpec `json:"spec"`
}

// MachineConfigSpec is the spec for MachineConfig
type MachineConfigSpec struct {
    // OSImageURL specifies the remote location that will be used to
    // fetch the OS.
    OSImageURL string `json:"osImageURL"`
    // Config is a Ignition Config object.
    Config runtime.RawExtension `json:"config"`

    // +nullable
    KernelArguments []string `json:"kernelArguments"`
    Extensions      []string `json:"extensions"`

    FIPS       bool   `json:"fips"`
    KernelType string `json:"kernelType"`
}
```

The Machine Config Daemon

- Supports a subset of Ignition configuration for customization

- Applies Ignition config from rendered MachineConfig to machines and reconciles changes

- Updates the underlying OS (Fedora CoreOS)
  - Pulls OSTree commit from machine-os-content container in payload
  - Writes OSTree to disk and triggers reboot

Red Hat

Agenda

What is OKD?

Operators Overview

The Machine Config Operator

What is Fedora CoreOS?

DEMO

Questions!

Red Hat

Fedora CoreOS in one Sentence

"Fedora CoreOS is… An automatically updating, minimal, monolithic, container-focused operating system, designed for clusters but also operable standalone, optimized for Kubernetes but also great without it."

Fedora CoreOS in two **shorter** Sentences

- Fedora CoreOS is an auto updating container OS.
- You can run it with kubernetes or without it.

Fedora CoreOS Lineage

- Came from the merging of two communities:
  - CoreOS Inc's Container Linux
  - Project Atomic's Atomic Host
- Incorporates Container Linux
  - Philosophy
  - Provisioning Stack
  - Cloud Native Expertise
- Incorporates Atomic Host
  - Fedora Foundation
  - Update Stack
 - SELinux Enhanced Security

Red Hat
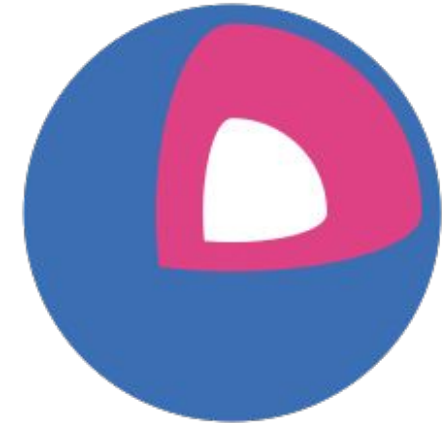
Fedora CoreOS Features

Features: Automatic Updates

- 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
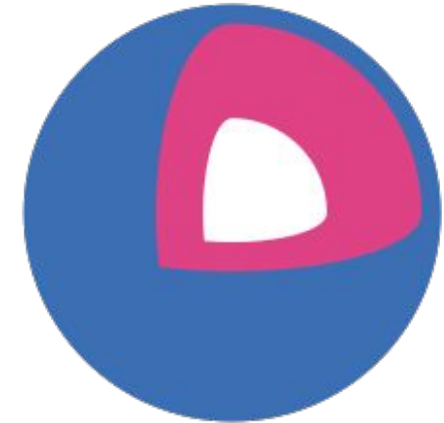
Multiple Update Streams

- 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

Features: 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

Red Hat

Features: Cloud Native & Container Focused

- Software runs in containers
  - podman or moby engine container runtimes
- Ready for clustered deployments
  - Spin up 100 nodes and have them join a cluster
    - Ignition configs used to automate cluster join
  - Spin down nodes when no longer needed
  - Spin up nodes again when load increases
- Offered on (or for) a plethora of cloud/virt platforms
  - Alibaba, AWS, Azure, DigitalOcean, Exoscale, GCP, Openstack, Vultr, VMWare, QEMU/KVM

Red Hat

Features: 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

Red Hat
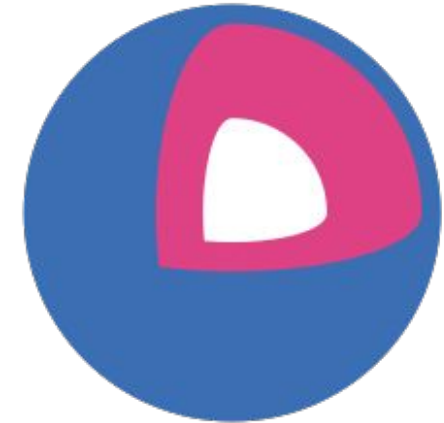
Ignition: Details

Ignition configs
- Declarative JSON documents provided via user data
- Runs exactly once, during the initramfs stage on first boot
- Can write files and systemd units, create users and groups, partition disks, create RAID arrays, format filesystems
- If provisioning fails, the boot fails (no half provisioned systems)
- Ignition configs are machine-friendly (JSON), currently spec v3

Writing Configs
- Fedora CoreOS Config Transpiler to translate to Ignition spec
  - Configs are Human friendly (YAML)
  - Ignition semantics, plus sugar for common operations
  - Transpiler catches common errors at build time

```
{
  "ignition": {
    "config": {},
    "timeouts": {},
    "version": "3.0.0"
  },
  "passwd": {
    "users": [
      {
        "name": "core",
        "passwordHash":
"$6$43y3tkl...",
        "sshAuthorizedKeys": [
          "key1"
        ]
      }
    ]
  },
  "storage": {},
  "systemd": {}
}
```

These features in use in OpenShift OKD

- Automated Provisioning:
    - `openshift-install` generates Ignition configs
    - When each node is started:
        - Ignition applies the configuration
        - Subsequent processes join the node to the cluster
    - A single bootstrap node Ignition config is ~300KiB
        - A lot of data is conveyed in the Ignition configuration in order to bootstrap a cluster from scratch.
- OS Versioning & Security
    - OKD builds on top of Fedora CoreOS
        - Know exactly what was delivered and running. Makes bug reporting is easy.
- Cloud Native & Container Focused
    - New nodes are easily added to the cluster (via machine API + Ignition)
    - Software runs in containers
- Automatic Updates
    - New releases of OKD can be applied automatically or with the click of a button.

Recap: What is Fedora CoreOS?

An automatically updating Linux OS

Aimed at Containerized Workloads

Based on RPM-OSTree and Ignition

Composed of Fedora RPM Packages

Great for running Kubernetes Clusters on top

Agenda

What is OKD?
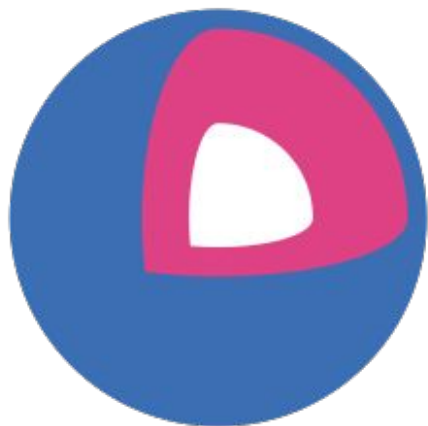
Operators Overview

The Machine Config Operator

What is Fedora CoreOS?

DEMO

Questions!

Red Hat

DEMO

Red Hat

Fedora CoreOS Working Group

IRC
#fedora-coreos on Freenode

Issue Tracker
github.com/coreos/fedora-coreos-tracker

Discussion Forum
discussion.fedoraproject.org/c/server/coreos

Mailing List
coreos@lists.fedoraproject.org

Weekly Meetings
apps.fedoraproject.org/calendar/CoreOS/

OKD Working Group

Slack
#openshift-dev on **kubernetes.slack.com**
#general on **openshiftcommons.slack.com**

Google Group
groups.google.com/forum/#!forum/okd-wg

Bi-weekly Video Conference Meetings
apps.fedoraproject.org/calendar/okd

Repositories
github.com/openshift/community
github.com/openshift/okd

Resources

okd.io

docs.okd.io

github.com/openshift/okd

github.com/openshift/community

Agenda

What is OKD?

Operators Overview

The Machine Config Operator

What is Fedora CoreOS?

DEMO

Questions!

Red Hat