

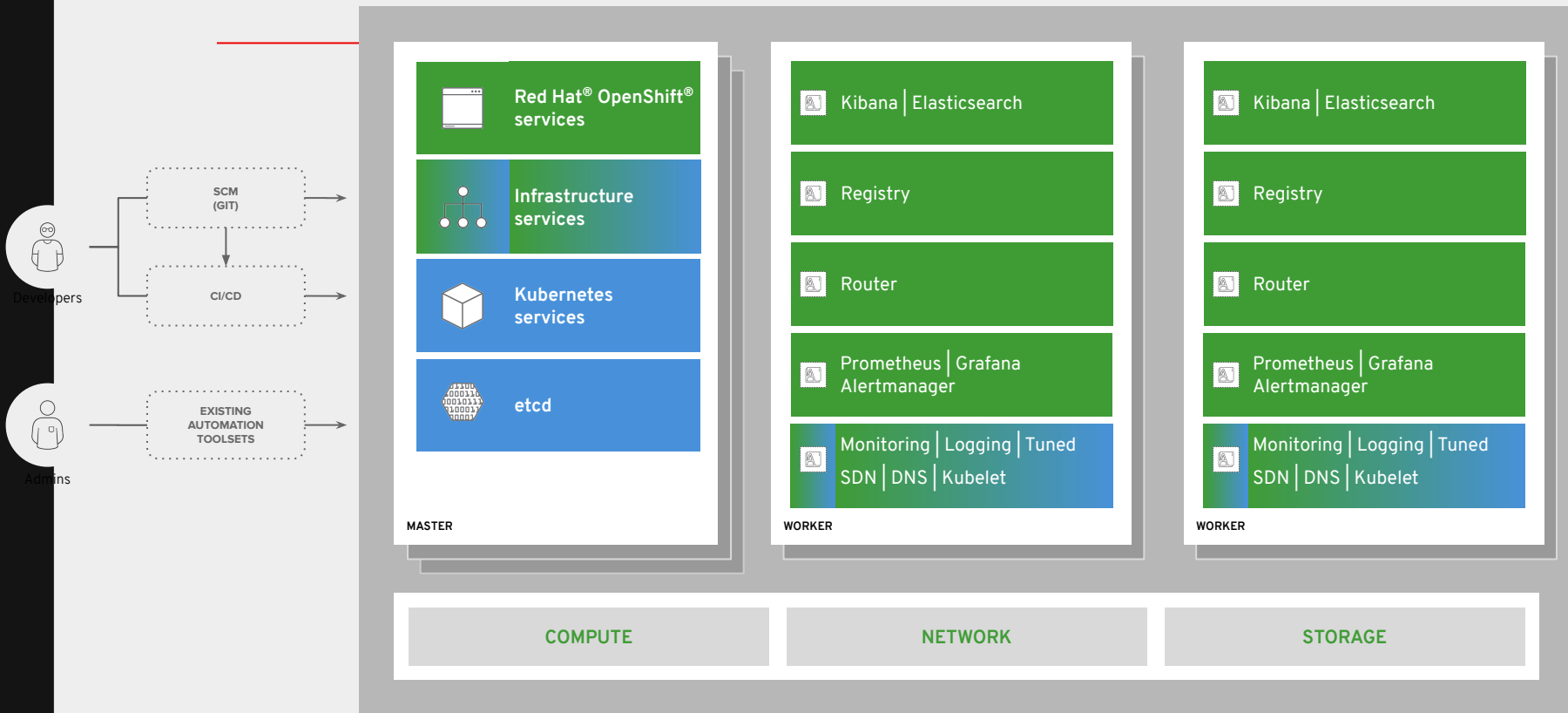


Red Hat OpenShift

Installation & Operating

OpenShift Architecture

2



Installation Paradigms

OPENSIFT CONTAINER PLATFORM

Full Stack Automated (IPI)

Simplified opinionated “Best Practices” for cluster provisioning

Fully automated installation and updates including host container OS.



Pre-existing Infrastructure (UPI)

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries



HOSTED OPENSIFT

Red Hat OpenShift on IBM Cloud *

Deploy directly from the IBM Cloud console. An IBM service, master nodes are managed by IBM Cloud engineers.

Azure Red Hat OpenShift **

Deploy directly from the Azure console. A MSFT service, jointly managed by Red Hat and Microsoft

OpenShift Dedicated **

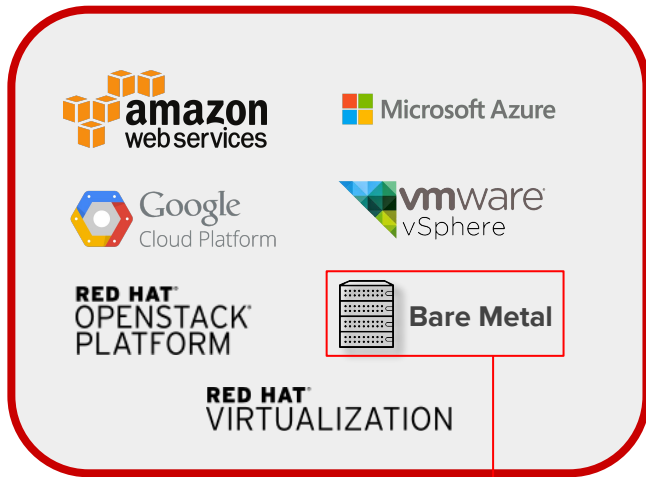
Get a powerful cluster, fully managed by Red Hat engineers and support; a Red Hat service.

* Based on OCP v4.3 GA slated for March; public beta available now

** Entitlements of OCP obtained through a Cloud Pak purchase are not transferable to these environments

4.6 Supported Providers

Full Stack Automation (IPI)



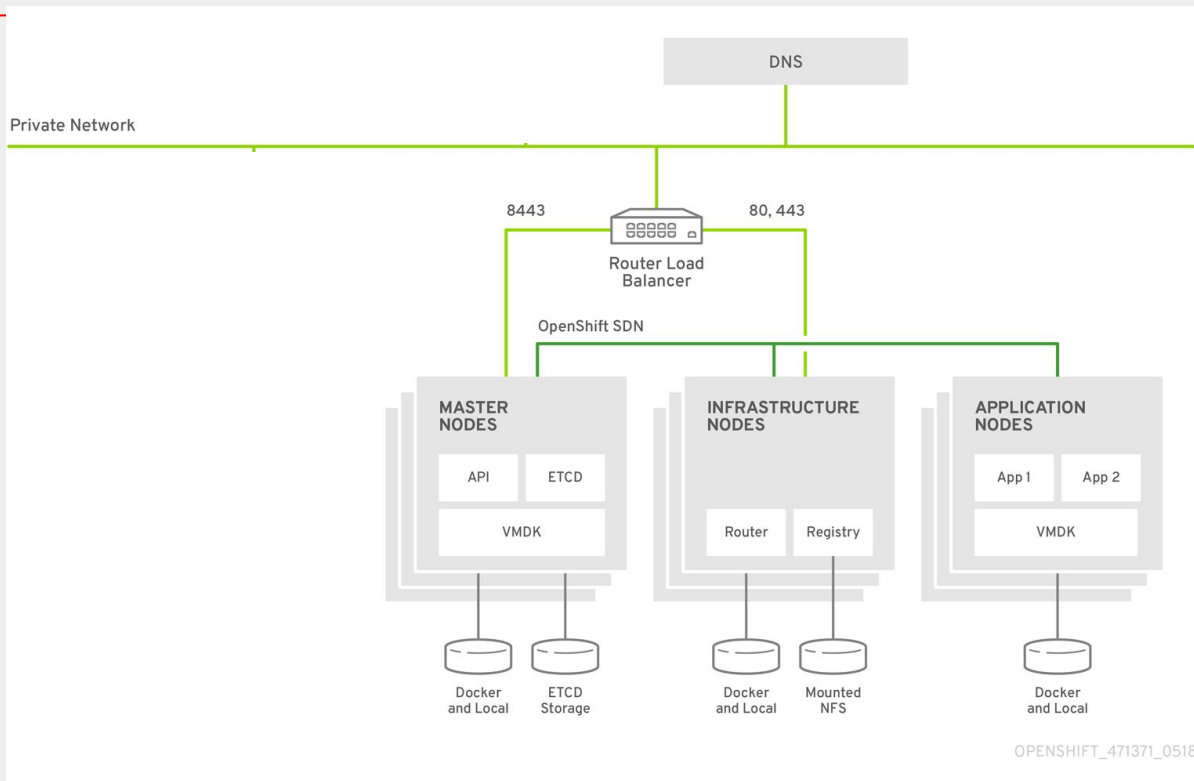
New addition in OCP 4.6

Pre-existing Infrastructure (UPI)



Now supports deploying to VMware vSphere 7.0

OCP Layout



[← View All](#) / Create a new OpenShift cluster

Create a new OpenShift cluster

Location

Availability ⓘ

Single zone

Multizone

Geography

Europe ▾

Worker zone ⓘ

- ☐ Amsterdam 03
- ☐ Frankfurt 02
- ☐ Frankfurt 04
- ☐ Frankfurt 05
- ☐ London 02

Order summary

b3c.4x16 - 4 vCPUs 16GB RAM

3 worker nodes

0,81 \$ / hr

[IP allocation](#)

OCP license fee ⓘ

1.200,00 \$ / month

Total***1.781,04 \$ / month**

estimated


*Actual monthly total will vary with [tiered pricing](#) for the hourly worker nodes and the 30-day fixed [OCP license fee](#).


Additional charges for bandwidth might apply. [Learn more.](#)

Create cluster

Add to estimate

Need help? [Contact IBM Cloud Sales](#)

**Red Hat**

Red Hat OpenShift Cluster Manager

Clusters

[Documentation](#)

[OperatorHub.io](#)

[Cluster Manager Feedback](#)

[Report an OpenShift Bug](#)

Infrastructure Provider

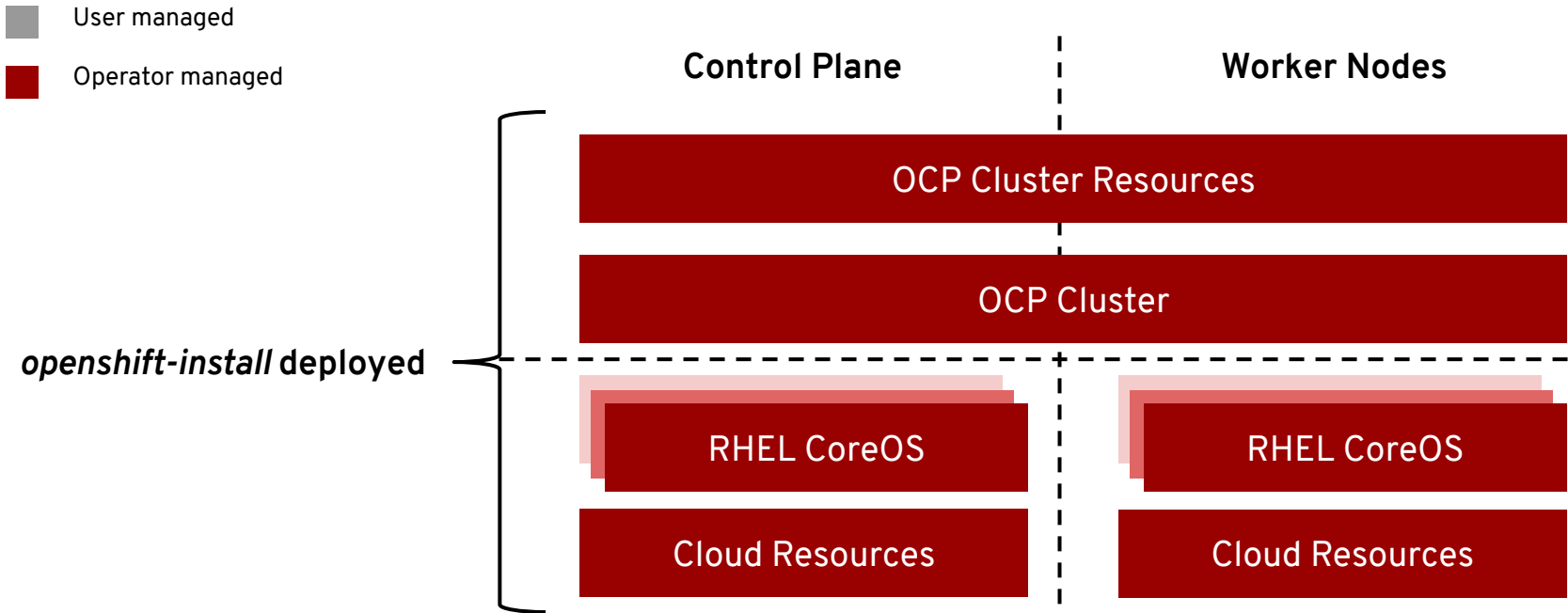



Bare Metal


<> Developer Preview



Full-stack Automated Installation (aka IPI)



Full Stack Automated Deployments

Simplified Cluster Creation

Designed to easily provision a “best practices” OpenShift cluster

- New CLI-based installer with interactive guided workflow that allows for customization at each step
- Installer takes care of provisioning the underlying Infrastructure significantly reducing deployment complexity
- Leverages RHEL CoreOS for all node types enabling full stack automation of installation and updates of both platform and host OS content

Faster Install

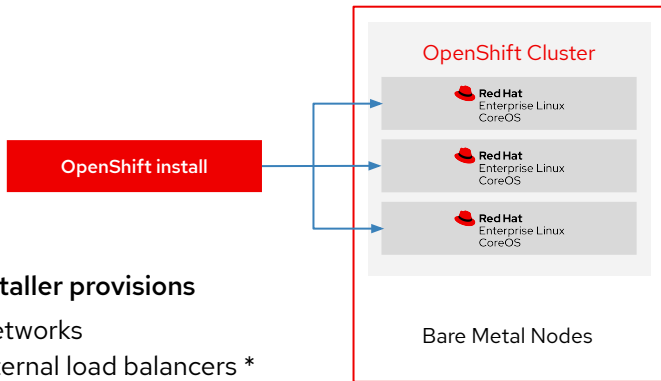
The installer typically finishes within 30 minutes

- Only minimal user input needed with all non-essential install config options now handled by component operator CRD's
- [See the OpenShift documentation for more details](#)

```
$ ./openshift-install --dir ./demo create cluster
? SSH Public Key /Users/demo/.ssh/id_rsa.pub
? Platform aws
? Region us-west-2
? Base Domain example.com
? Cluster Name demo
? Pull Secret [? for help]
*****
INFO Creating cluster...
INFO Waiting up to 30m0s for the Kubernetes API...
INFO API v1.11.0+c69f926354 up
INFO Waiting up to 30m0s for the bootstrap-complete event...
INFO Destroying the bootstrap resources...
INFO Waiting up to 10m0s for the openshift-console route to be created...
INFO Install complete!
INFO Run 'export KUBECONFIG=<your working directory>/auth/kubeconfig' to
manage the cluster with 'oc', the OpenShift CLI.
INFO The cluster is ready when 'oc login -u kubeadmin -p <provided>'
succeeds (wait a few minutes).
INFO Access the OpenShift web-console here:
https://console-openshift-console.apps.demo.example.com
INFO Login to the console with user: kubeadmin, password: <provided>
```

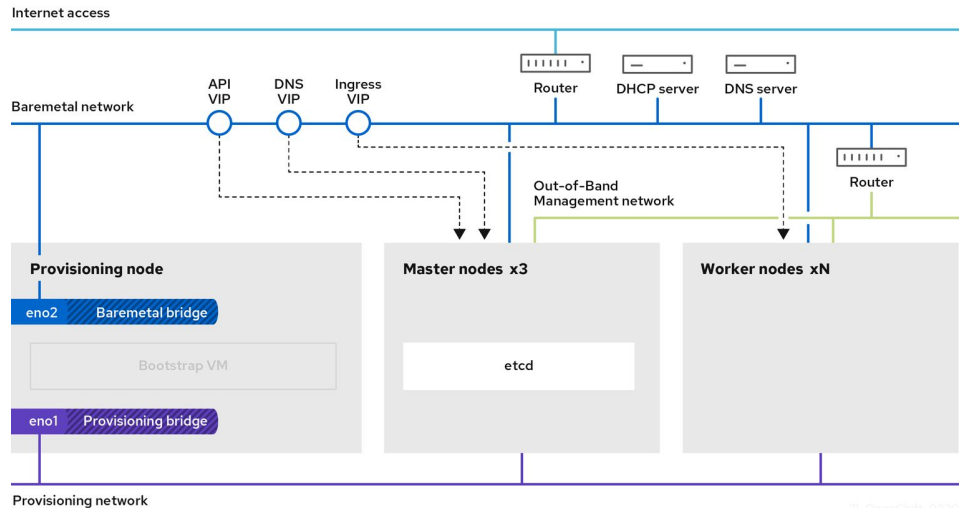
Full stack automation (IPI) installation on Bare Metal

Deploying Red Hat OpenShift on Bare Metal on Installer-Provisioned Infrastructure (IPI)



► Installer provisions

- Networks
- Internal load balancers *
- Internal DNS *
- Red Hat CoreOS installation
- CoreOS ignition configs
- OpenShift nodes
- OpenShift cluster resources


© 2021 Red Hat, Inc. OpenShift 4.6

Full stack automation (IPI) installation on Bare Metal

Deploying Red Hat OpenShift on Bare Metal on Installer-Provisioned Infrastructure (IPI)

Bare Metal Management

Powered by Metal³ and OpenStack Ironic under the hood

Host Power Management

Redfish, IPMI, iDrac, iLo.

Provisioning over the network

Installation over DHCP/PXE or Virtual Media

Disconnected Installations

RHCOS image cache and disconnected registry



Metal³



OpenStack Ironic

```
apiVersion: v1
basedomain: <domain>
metadata:
  name: <cluster-name>
networking:
  machineCIDR: <public-cidr>
  networkType: OVNKubernetes
compute:
- name: worker
  replicas: 2
controlPlane:
  name: master
  replicas: 3
platform:
  baremetal: {}
```

```
platform:
  baremetal:
    apiVIP: <api-ip>
    ingressVIP: <wildcard-ip>
    provisioningNetworkInterface: <NIC1>
    provisioningNetworkCIDR: <CIDR>
    hosts:
      - name: openshift-master-0
        role: master
        bmc:
          address: ipmi://<out-of-band-ip>
          username: <user>
          password: <password>
          bootMACAddress: <NIC1-mac-address>
          hardwareProfile: default
      - name: openshift-master-1
        role: master
        bmc:
          address: ipmi://<out-of-band-ip>
          username: <user>
          password: <password>
          bootMACAddress: <NIC1-mac-address>
          hardwareProfile: default
```


How everything deployed comes under management

Masters (Special)

- Terraform provisions initial masters*
- Machine API adopts existing masters post-provision
- Each master is a standalone Machine object
- Termination protection (avoid self-destruction)

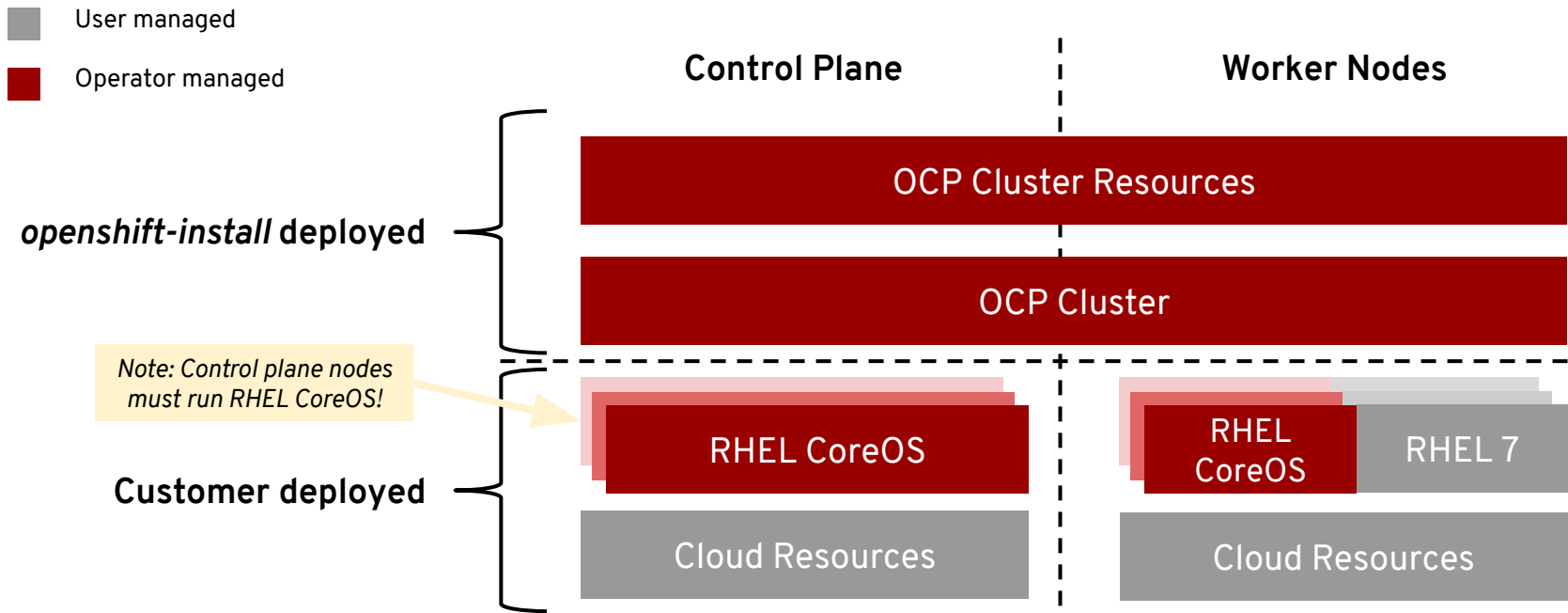
Workers

- Each Machine Pool corresponds to MachineSet
- Optionally autoscale (min,max) and health check (replace if not ready > X minutes)

Multi-AZ

- MachineSets scoped to single AZ
- Installer stripes N machine sets across AZs by default
- Post-install best effort balance via cluster autoscaler

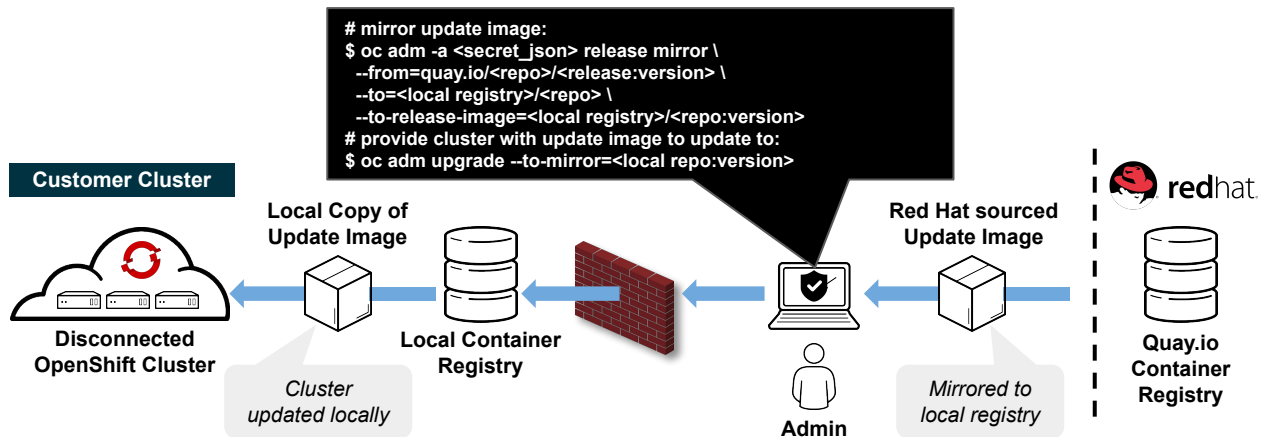
Pre-existing Infrastructure Installation (aka UPI)



Comparison of Paradigms

	Full Stack Automation	Pre-existing Infrastructure
Build Network	Installer	User
Setup Load Balancers	Installer	User
Configure DNS	Installer	User
Hardware/VM Provisioning	Installer	User
OS Installation	Installer	User
Generate Ignition Configs	Installer	Installer
OS Support	Installer: RHEL CoreOS	User: RHEL CoreOS + RHEL 7
Node Provisioning / Autoscaling	Yes	Only for providers with OpenShift Machine API support

Disconnected “Air-gapped” Installation & Upgrading



Overview

- 4.2 introduces support for installing and updating OpenShift clusters in disconnected environments
- Requires local Docker 2.2 spec compliant container registry to host OpenShift content
- Designed to work with the user provisioned infrastructure deployment method
 - *Note: Will not work with Installer provisioned infrastructure deployments*

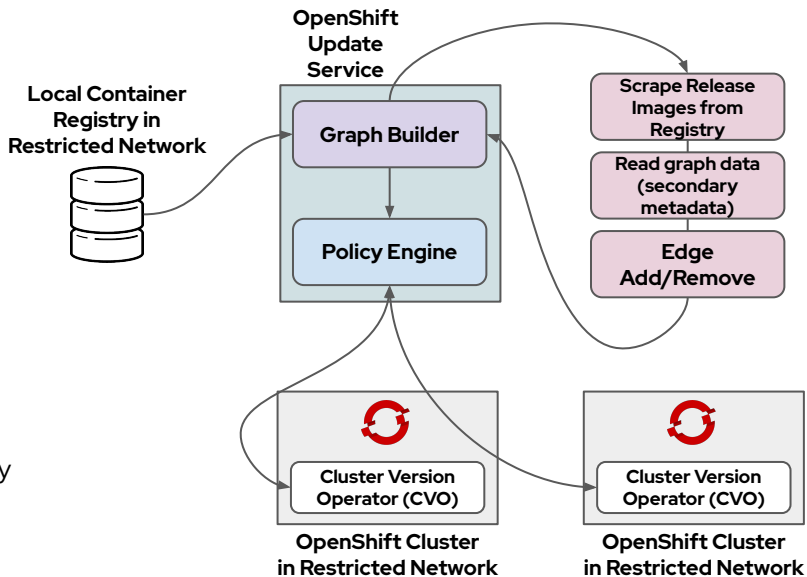
Installation Procedure

- Mirror OpenShift content to local container registry in the disconnected environment
- Generate install-config.yaml: `./openshift-install create install-config --dir <dir>`
 - Edit and add pull secret (PullSecret), CA certificate (AdditionalTrustBundle), and image content sources (ImageContentSources) to install-config.yaml
- Set the `OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE` environment variable during the creation of the ignition configs
- Generate the ignition configuration: `./openshift-install create ignition-configs --dir <dir>`
- Use the resulting ignition files to bootstrap the cluster deployment

OpenShift Update Service

Update manager for your clusters in restricted or disconnected networks

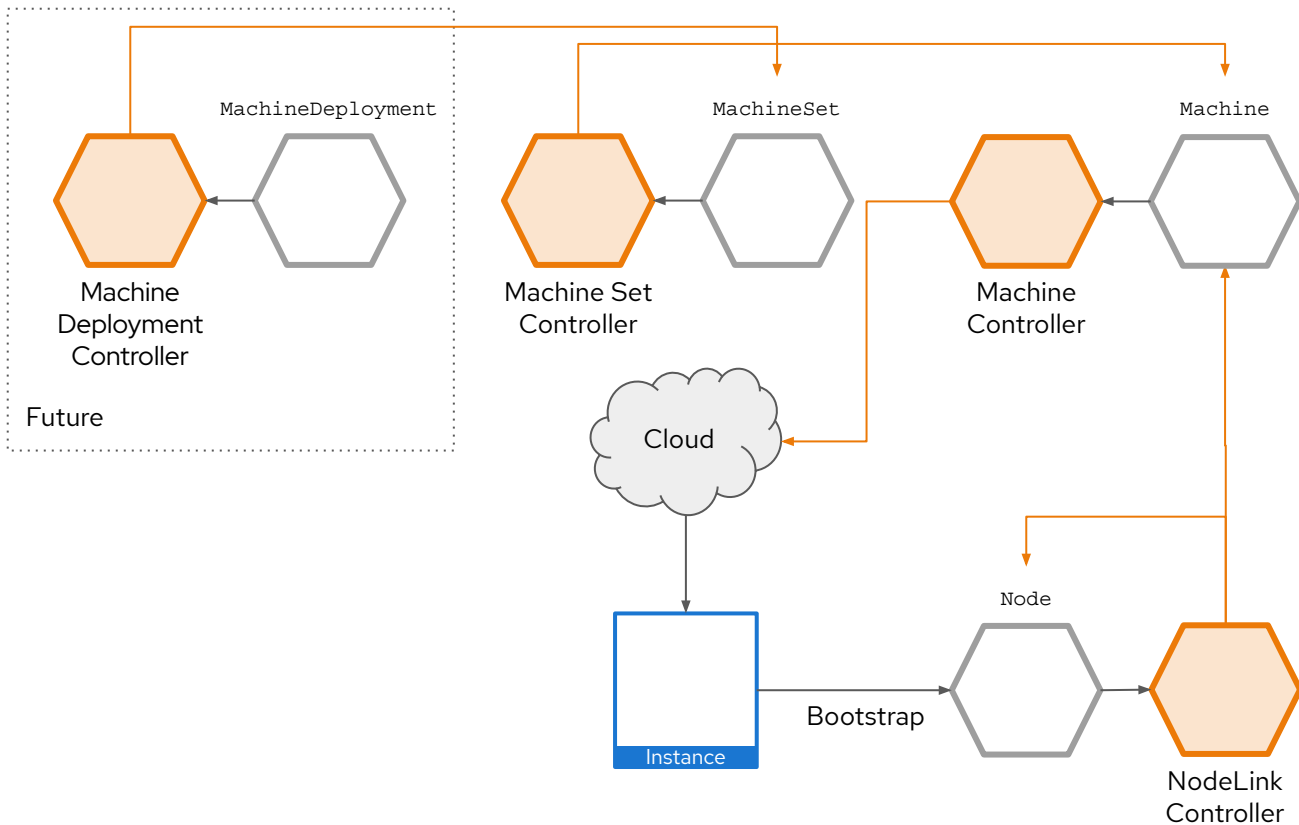
- OpenShift Update Service (OSUS) is the on-premise release of Red Hat's hosted update service
- Supports the publishing of upgrade graph information to clusters in restricted networks
- Provides clusters with a list of next recommended update versions based on the current version installed on the cluster
- Comprised of two services:
 - **Graph Builder:** Fetches OpenShift release payload information (primary metadata) from any container registry (compatible with [Docker registry V2 API](#)) and builds a [directed acyclic graph](#) (DAG) representing valid upgrade edges
 - **Policy Engine:** Responsible for selectively serving updates to every cluster by altering a client's view of the graph with a set of filters
- GA release planned for post-4.6 and will be distributed on Operator Hub as an optional add-on operator
- [Blog post announcing OpenShift Update Service](#)



OpenShift 4 Cluster Management

Powered by Operators,
OpenShift 4 automates
many cluster
management activities

Cloud API



Machine Config Operator

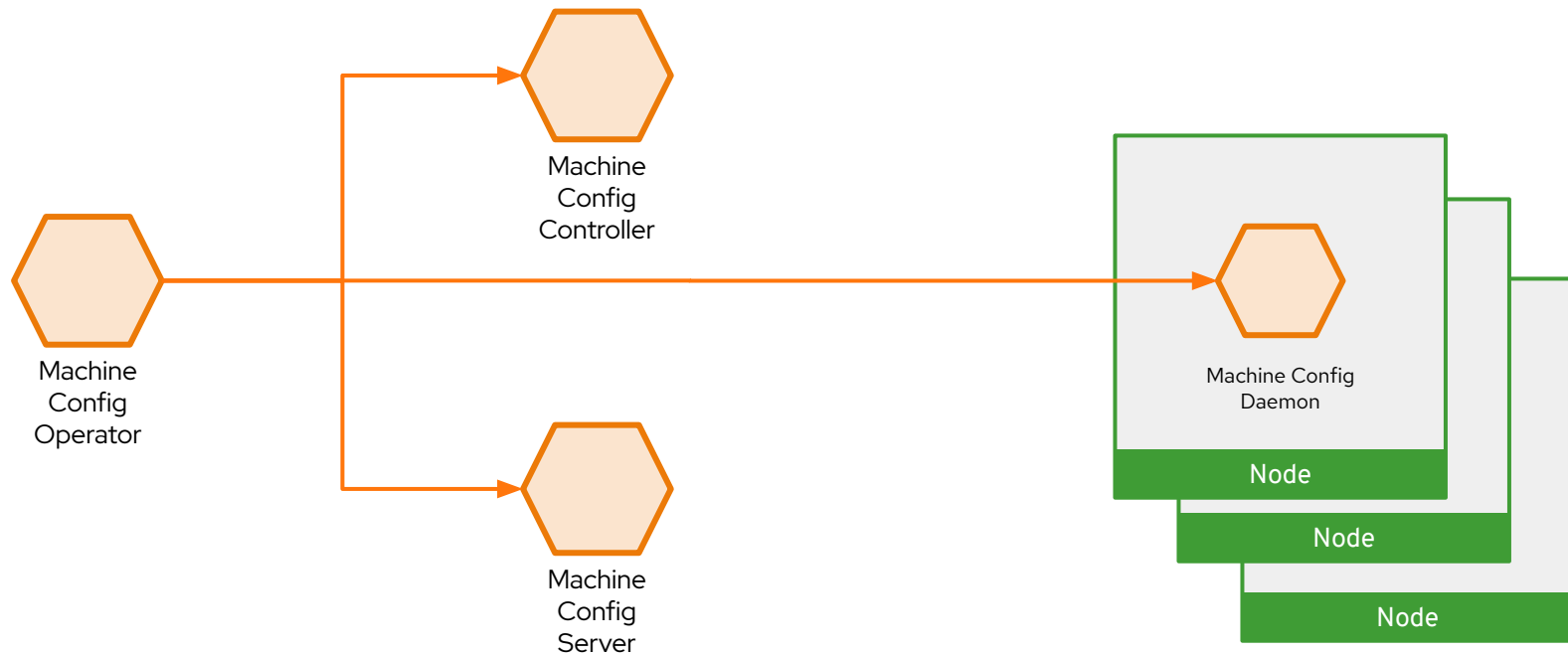
A Kube-native way to configure hosts

OS configuration is stored and applied across the cluster via the Machine Config Operator.

- Subset of ignition modules applicable post provisioning
 - SSH keys
 - Files
 - systemd units
 - kernel arguments
- Standard k8s YAML/JSON manifests
- Desired state of nodes is checked/fixed regularly
- Can be paused to suspend operations

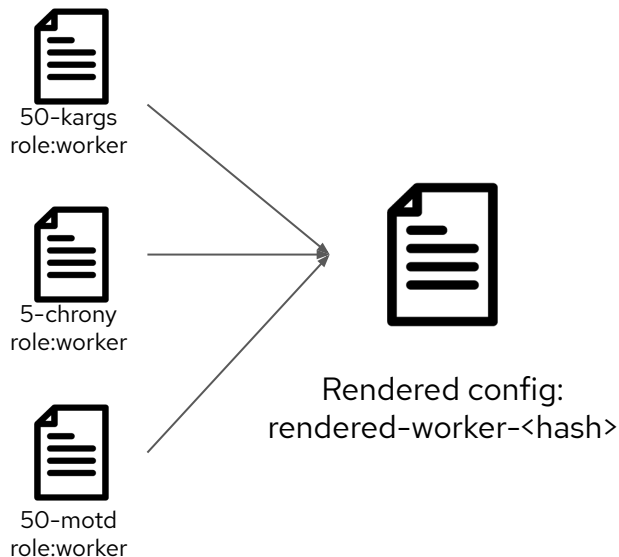
```
# test.yaml
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: test-file
spec:
  config:
    storage:
      files:
        - contents:
            source: data:,hello%20world%0A
            verification: {}
          filesystem: root
          mode: 420
          path: /etc/test
```

Operator/Operand Relationships



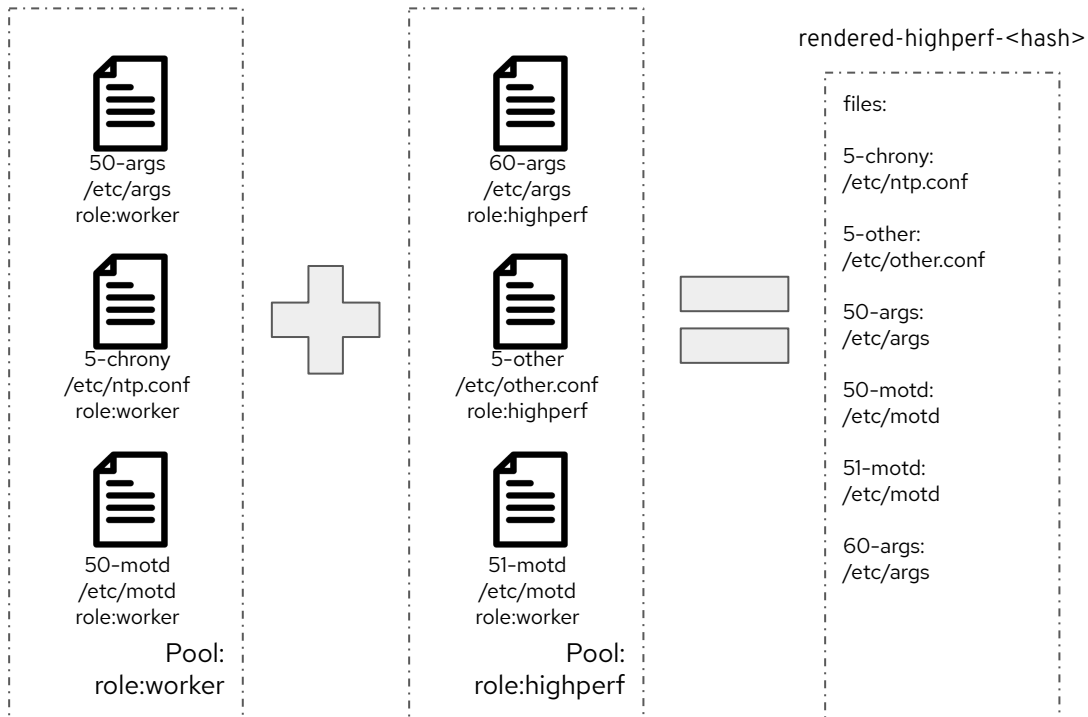
Machine Config and Machine Config Pool

Inheritance-based mapping of configuration to nodes



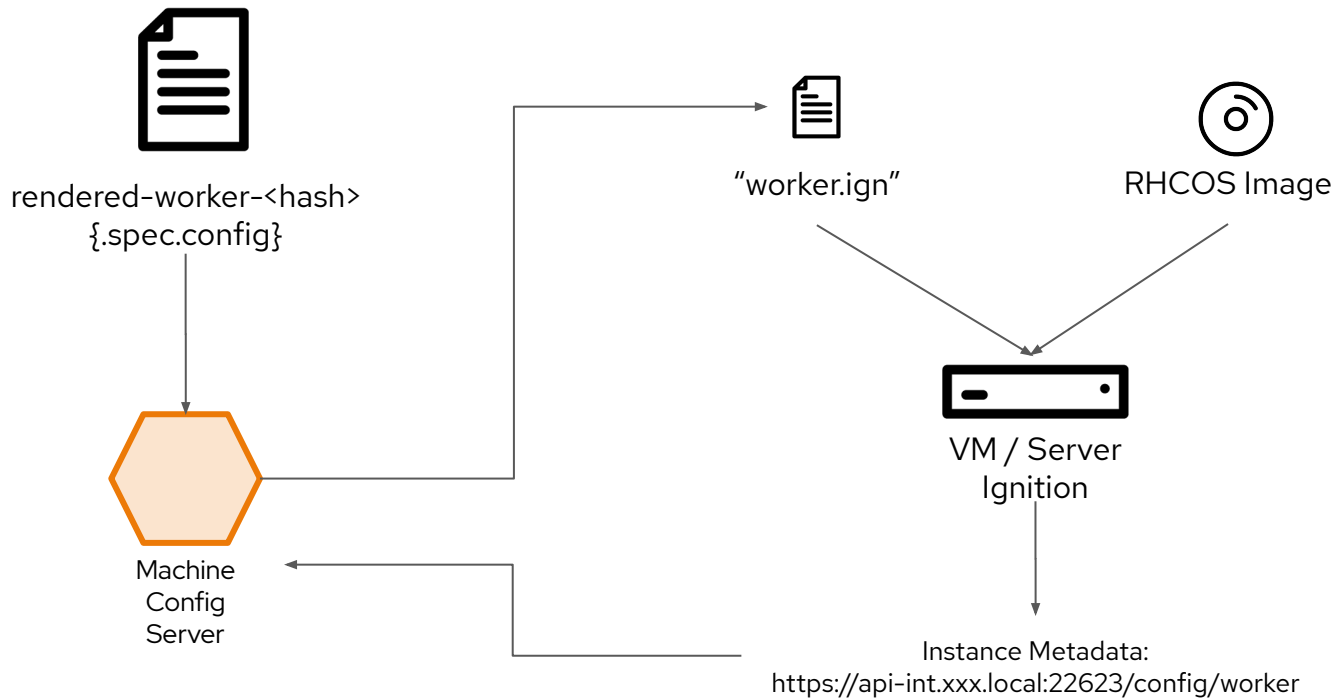
Custom Machine Config Pools

Hierarchical/layered configuration rendering



Machine Config Server

Providing Ignition configuration for provisioning



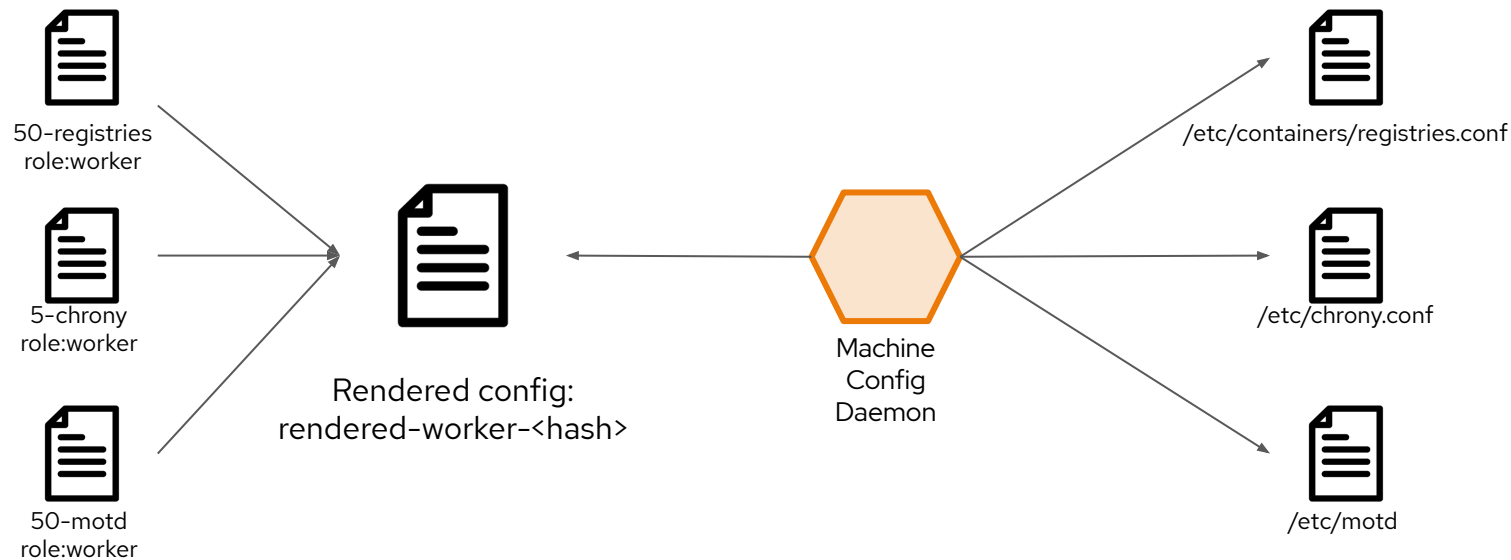
Machine Config Server

Identical nodes at massive scale



Machine Config Daemon

Preventing drift



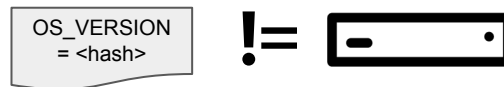
Machine Config Daemon

Acting on drift

The MCO coordinates with the MCD to perform the following actions, in a rolling manner, when OS updates and/or configuration changes are applied:

- Cordon / uncordon nodes
- Drain pods
- Stage node changes
 - OS upgrade
 - config changes
 - systemd units
- Reboot

1. Validates node state matches desired state



2. Validate cluster state & policy to apply change



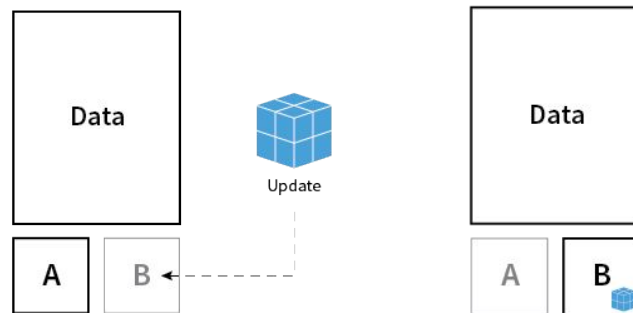
3. Change is rolled across cluster



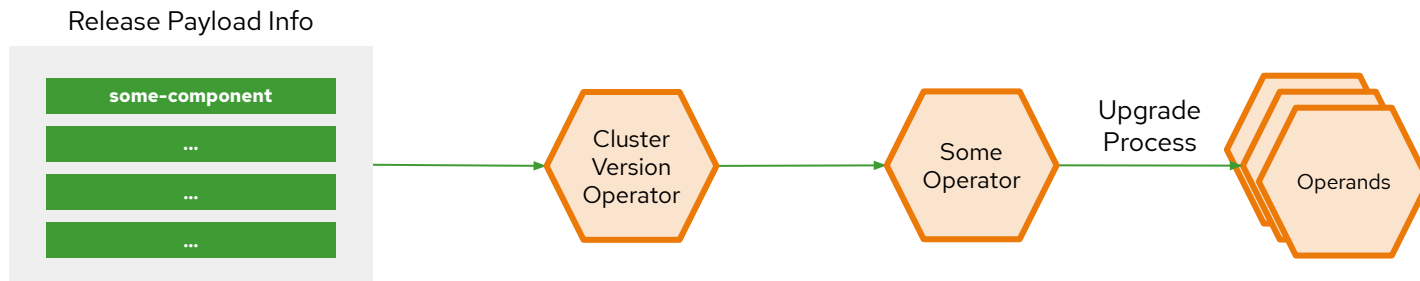
Transactional updates with rpm-ostree

Transactional updates ensure that RHEL CoreOS is never altered during runtime. Rather it is booted directly into an always “known good” version.

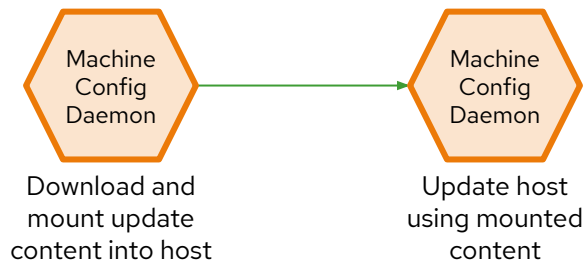
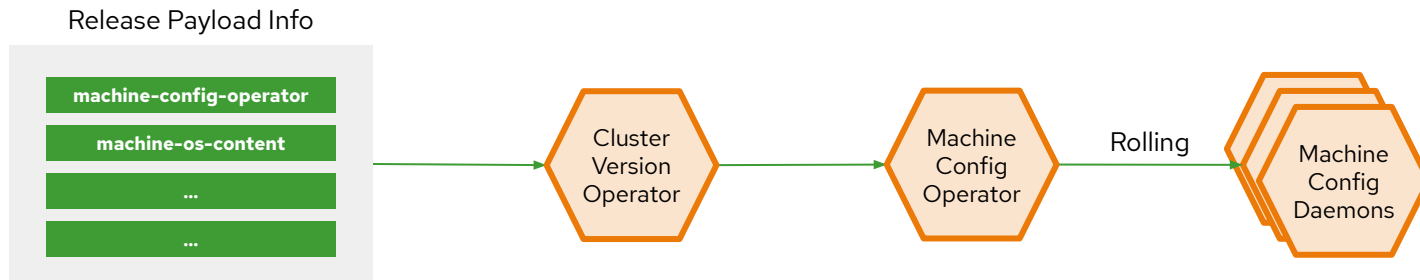
- Each OS update is versioned and tested as a complete image.
- OS binaries (/usr) are read-only
- OS updates encapsulated in container images
- file system and package layering available for hotfixes and debugging



Over-the-air updates: Cluster Components



Over-the-air updates: Nodes





linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHat