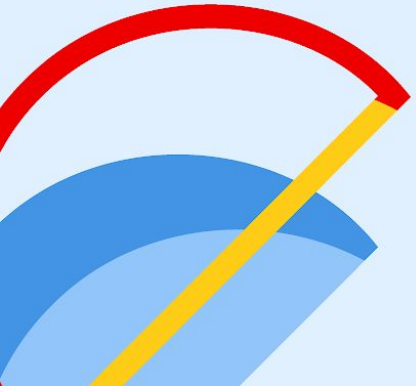
OpenShift in abgeschotteter (air-gapped) Umgebung

Robert Bohne

Principal Specialist Solution Architect,
OpenShift - Cross Segment

Red Hat

Foundation

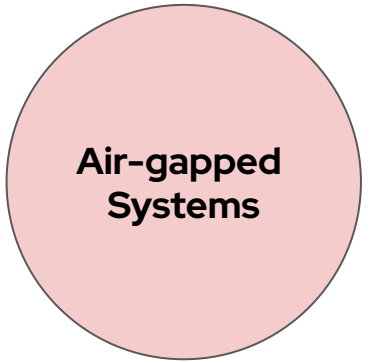


Red Hat

What is Air-gapped?

Environment with no physical connection to other networks.

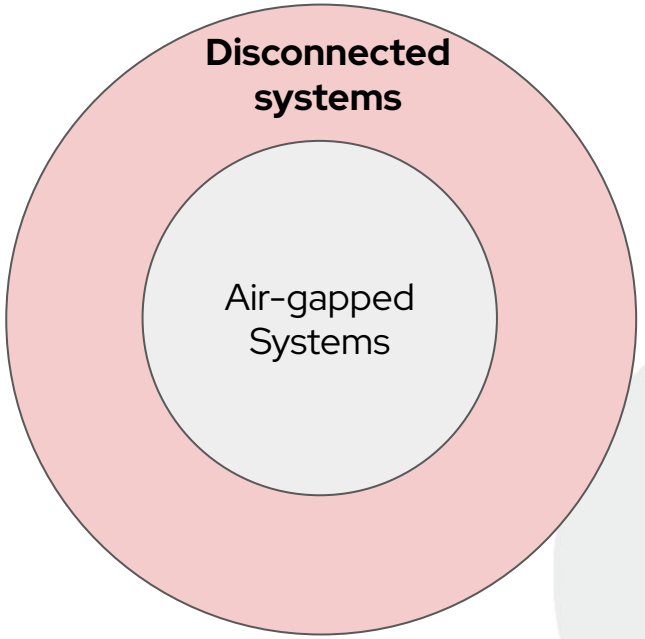
Data transfer only via a physical media.



Red Hat

Disconnected?

Environment with isolated network and might have a **bastion host with access** to wider organizational networks (and potentially the internet)

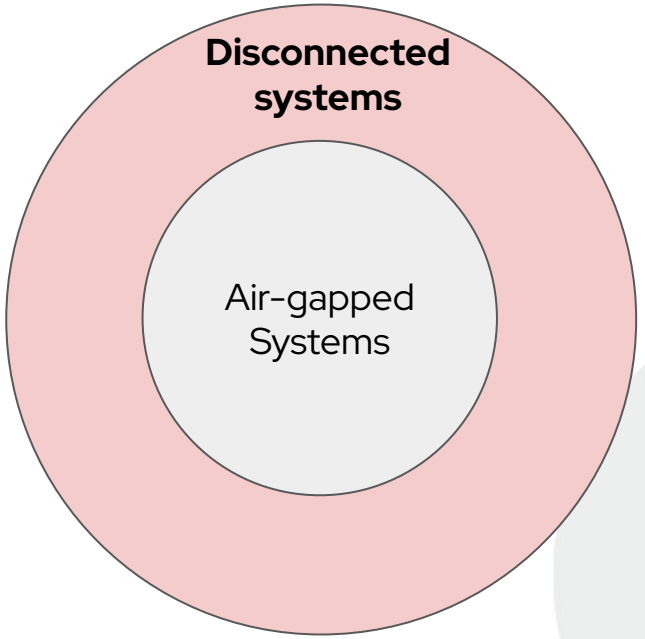


Red Hat

Difference?

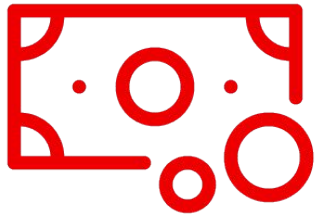
How to transfer data?

No Difference for OpenShift!

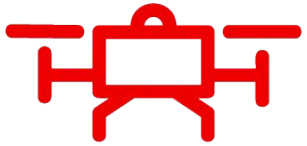


Red Hat

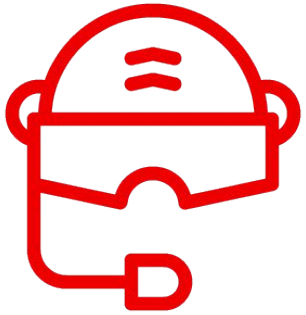
Where do you find air-gapped environments?



FSI



OT / Operational environments



Military / intelligence / law enforcement

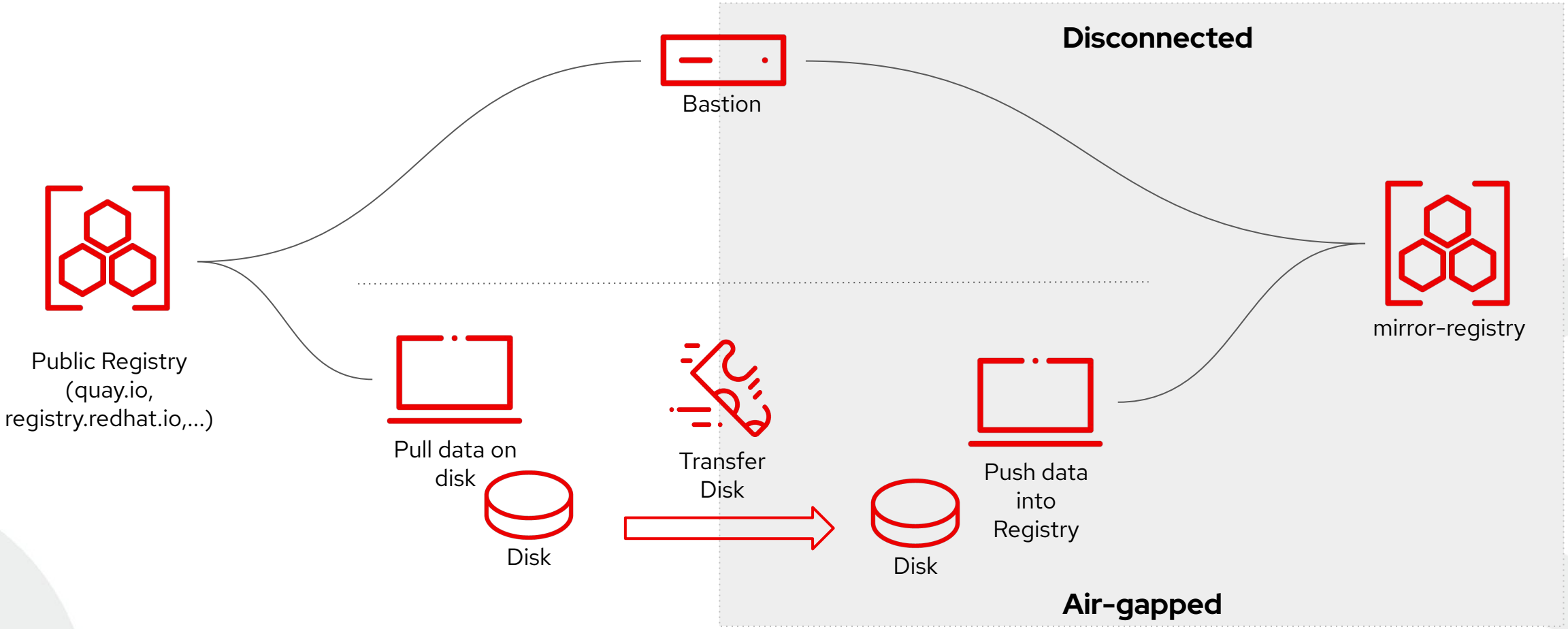
Telco

What's the base of OpenShift?

Container Image

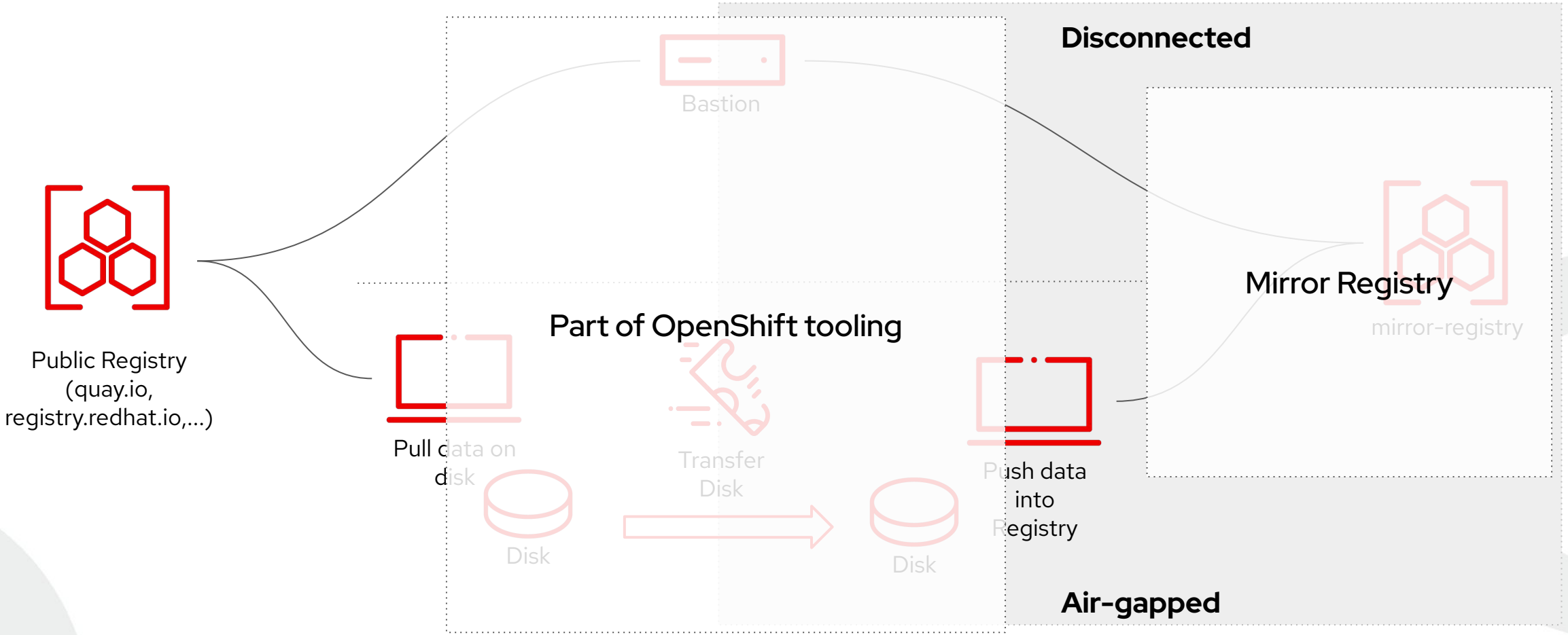
- Core OpenShift
- Operators - via OperatorHub
- Red Hat Enterprise Linux CoreOS
- ...all container images!
- Core OpenShift Binaries (oc, kubectl, installer,..)
- What is it not: Red Hat Satellite (RPM's)

Image transfer



Red Hat

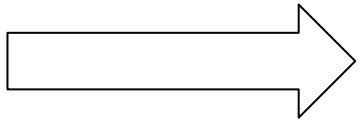
Image transfer

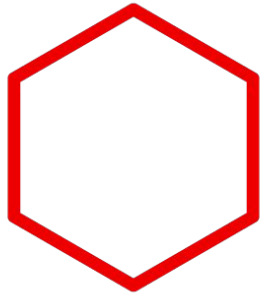


Red Hat

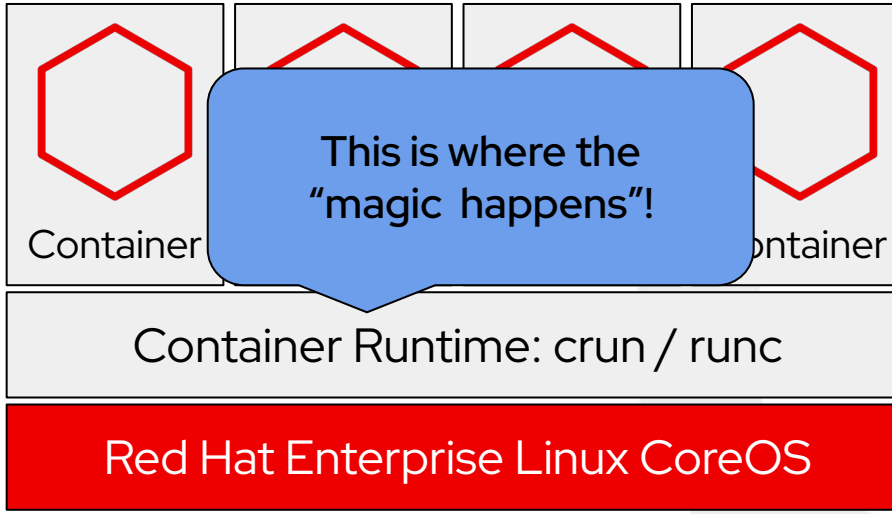
And now?

Container Image





Running Container



Red Hat

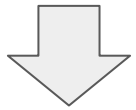
How do we address images?

- **qualified image:**
  - `host[:port]`
  - `host[:port]/namespace[/namespace…]`
  - `host[:port]/namespace[/namespace…]/repo`
  - `host[:port]/namespace[/namespace…]/repo(:_tag|@digest = @sha256:..)`
    - `registry.access.redhat.com/ubi9/ubi-micro:9.3-9`
    - `registry.access.redhat.com/ubi9/ubi-micro@sha256:f9426baf6a67db59561e58d7ae0...`
- **unqualified image:**
  - `namespace[/namespace…]/repo(:_tag|@digest)`
  - For RHEL, set runtime search in the CNI registry configuration. In OpenShift, use containerRuntimeSearchRegistries in image.config.openshift.io/cluster (the cluster image configuration)
- **Recommended to use qualified image names!**

And the magic? 🪄

- **Remapping and mirroring registries**

```
registry.access.redhat.com/ubi9/ubi-micro@sha256:f9426baf6a67db59561e58d7ae0...
```



```
local-registry.example.com/foo/bar/ubi9/ubi-micro@sha256:f9426baf6a67db59561e58d7ae0...
```

Container Runtime: crun / runc

Red Hat Enterprise Linux CoreOS

```
# cat /etc/containers/registries.conf
...
[[registry]]
  prefix = ""
  location = "registry.access.redhat.com/ubi9/ubi-micro"

  [[registry.mirror]]
     location = "local-registry.example.com/foo/bar/ubi9/ubi-micro"
     pull-from-mirror = "digest-only"
...
```


Red Hat

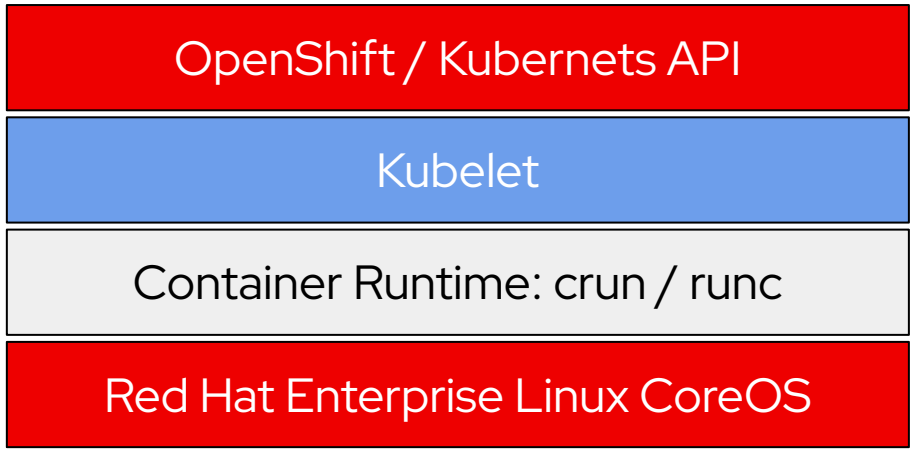
Demo!

- **Demo of remapping and mirroring registries**

https://asciinema.org/a/630862

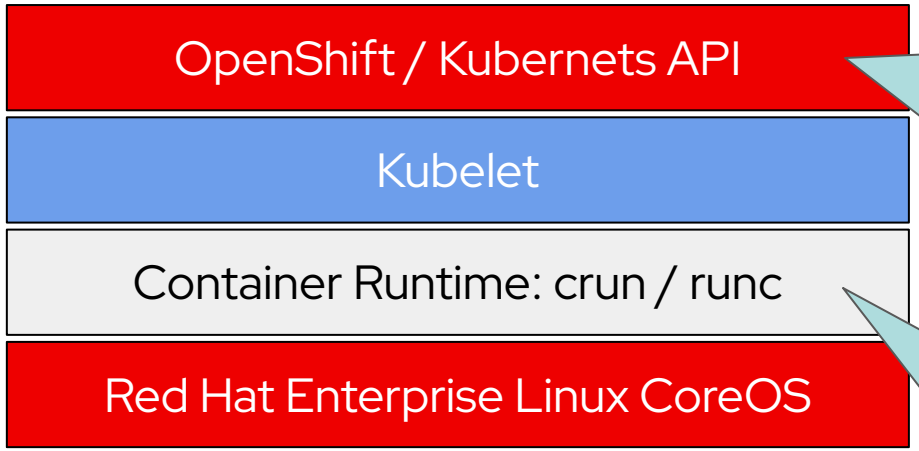
Red Hat

How do you configure this in OpenShift?



- ImageDigestMirrorSet (IDMS)
- ImageTagMirrorSet (ITMS)
- ~~ImageContentSourcePolicy (ICSP)~~

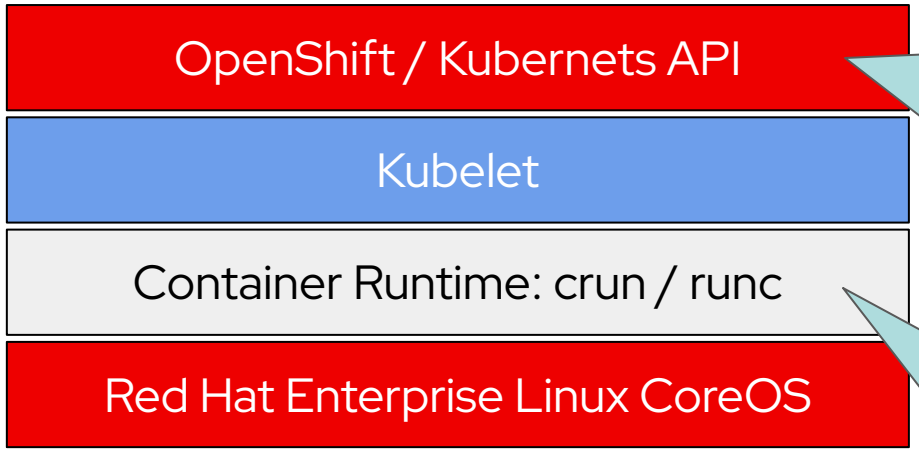
How do you configure this in OpenShift?




```
$ oc get itms  ubi -o yaml
apiVersion: config.openshift.io/v1
kind: ImageTagMirrorSet
metadata:
  name: ubi
spec:
  imageTagMirrors:
  - mirrors:
    - local-registry.example.com/foo/bar/ubi9/ubi
    source: registry.access.redhat.com/ubi9/ubi
```

```
# cat /etc/containers/registries.conf
...
[[registry]]
  prefix = ""
  location = "registry.access.redhat.com/ubi9/ubi"

  [[registry.mirror]]
    location = "local-registry.example.com/foo/bar/ubi9/ubi"
    pull-from-mirror = "tag-only"
...
```

How do you configure this in OpenShift?




```
$ oc get itms  ubi -o yaml
apiVersion: config.openshift.io/v1
kind: ImageDigestMirrorSet
metadata:
  name: ubi-micro
spec:
  imageDigestMirrors:
  - mirrors:
    - local-registry.example.com/foo/bar/ubi9/ubi-micro
    source: registry.access.redhat.com/ubi9/ubi-micro
```

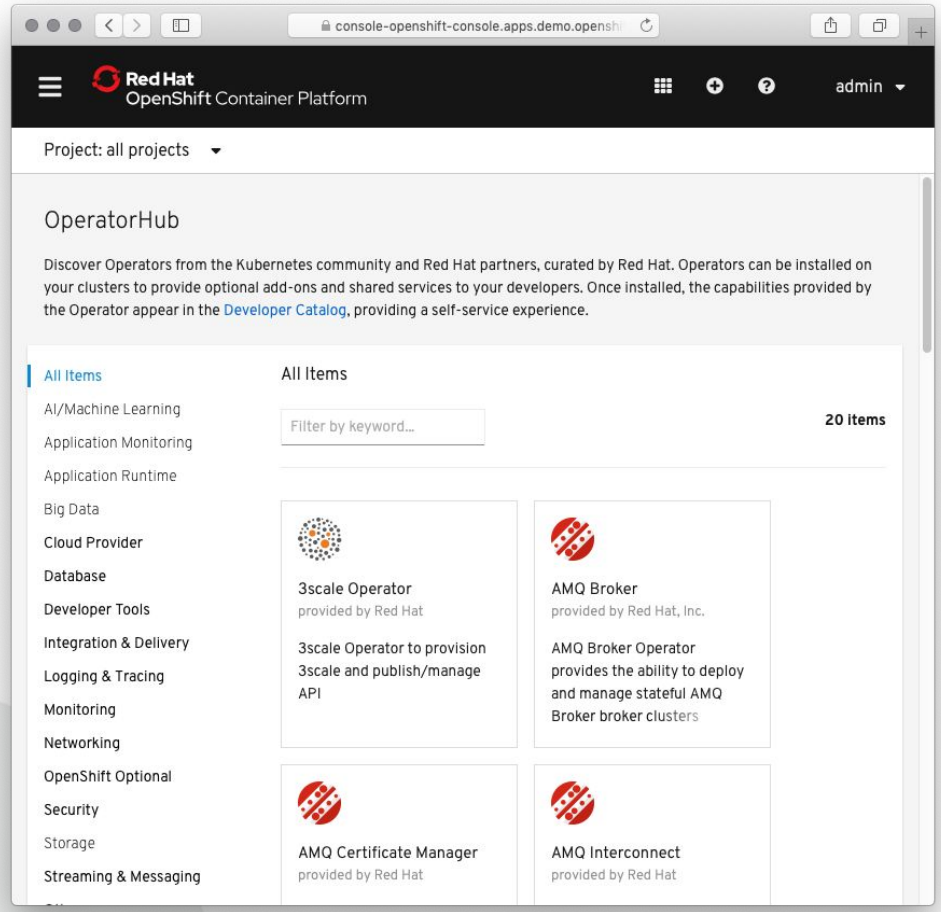
```
# cat /etc/containers/registries.conf
...
[[registry]]
  prefix = ""
  location = "registry.access.redhat.com/ubi9/ubi-micro"

  [[registry.mirror]]
    location = "local-registry.example.com/foo/bar/ubi9/ubi-micro"
    pull-from-mirror = "digest-only"
...
```


Red Hat

That’s the basics and enough for OpenShift Core!

What about Operators? Works, but how?

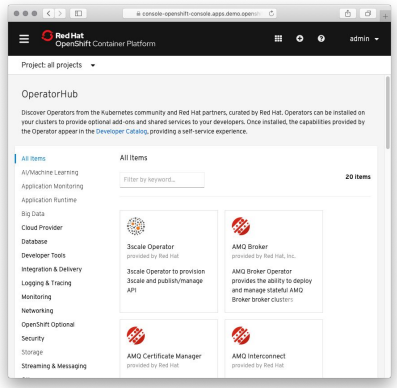
Operator Hub

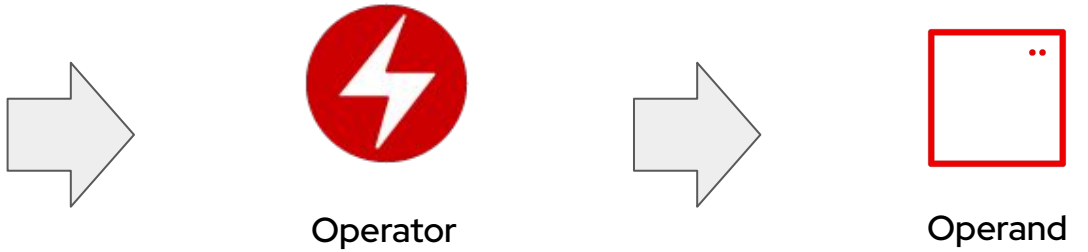


- **Backend is a Container Image**
  - Called Index Image
  - Referenced in CatalogSource
- **Everything is in the Index Image**
  - CustomResourceDefinitions
  - Operator Deployment artifacts
    - K8s/OCP YAML's
    - ServiceAccount, Deployment...
  - ClusterRoleBindings

Red Hat

Operator Hub

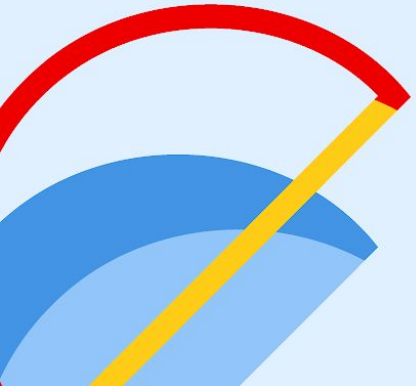




- **During the sync you have to know all Container Images**

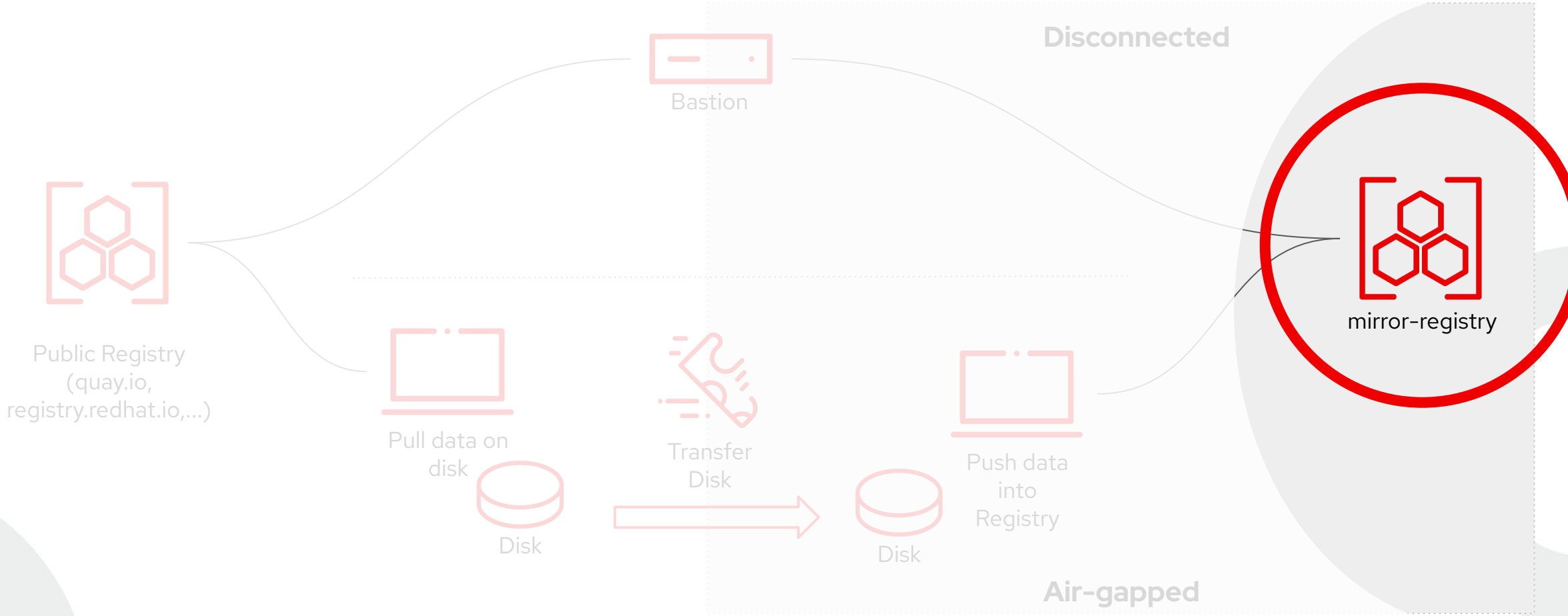
Red Hat

Content download, Image Mirror Part



Red Hat

First, let's talk about the **mirror-registry**



Red Hat

Mirror registry

- Every registry who supports 1.0.1 OCI distribution specification and
  - OCI image manifest
  - Docker image manifest Version 2 Schema 2
  - It's basically all registries on the market.
    - **Do you have other experience – please let us know!**

- Red Hat Mirror registry for OpenShift
  - A minimal Quay deployment serving as a registry to bootstrap your first disconnected cluster. Included in every OpenShift subscription.

Fundamentals

- Obtained as a offline capable, self-sufficient tarball container installer binaries and runtime images
- Requires podman >= 3.3+ and RHEL 8+
- Installer deploys mirror registry on local machine or given target host
- Configures registry for auto-start via systemd

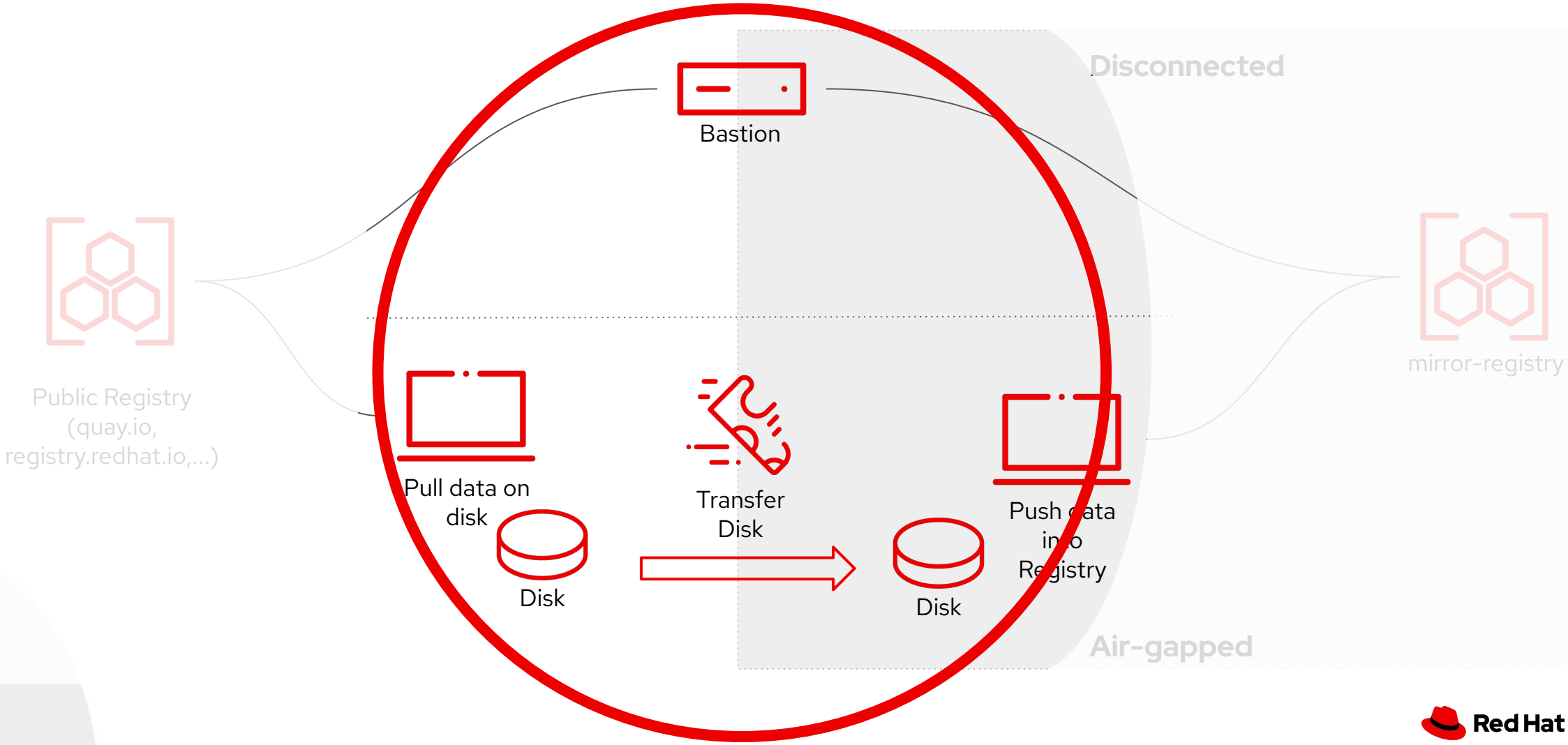
```
   __   __
  /  \ /  \     ______   _    _      __   __   __
 / /\ / /\ \   /   __  \ | |  | |    /  \  \ \ / /
/ /  / /  \ \  | |  | | | |  | |   / /\ \  \   /
\ \  \ \  / /  | |__| | | |__| |  / ____ \  | |
 \ \/ \ \/ /   \_   ___/  \____/ /_/    \_\ |_|
  \__/ \__/      \ \__
                  \___\ by Red Hat
 Build, Store, and Distribute your Containers

Usage:
  mirror-registry [command]

Available Commands:
  help        Help about any command
  install     Install Quay and its required dependencies.
  uninstall   uninstall will remove all Quay dependencies.

Flags:
  -h, --help      help for mirror-registry
  -v, --verbose   Display verbose logs

Use "mirror-registry [command] --help" for more information about a
command.
```

Mirror registry SLA

The mirror registry is a minimal, streamlined deployment of Quay on a single node with local disk storage. Bear in mind:

- Uptime is tied to the uptime of the single host running it, clustered deployments are not supported (if needed, use Red Hat Quay)
- Storage is local filesystem storage with no built-in replication, only store images that can easily be downloaded again if you don't have a SAN
- When the mirror registry is down, connected clusters will not be able to:
  - perform cold start (kubelet dependency)
  - add additional nodes or reboot nodes
  - update to a newer version / update OLM operators
  - perform node maintenance / failover workloads (unless images in cache)

Now let's talk about the **transfer**



Red Hat

Different options

- Copy container images via (disconnected & air-gapped)
  - **oc mirror plugin (recommended)**
    - **oc mirror plugin v2 - Tech Preview**
  - oc adm release mirror / oc adm catalog mirror
- Just configure a proxy registry (disconnected only)

Fundamentals

- Plugin for `oc` client, simply put in `$PATH`, obtained as a single executable
- Requires RHEL 8+, shipped as part of OCP but without specific release dependency
- Supports mirror all content (OCP releases, OCP operator catalogs, helm charts, custom images, CLI tools) of all OCP release
- Supports OCI compatible registries, including Quay >= 3.6

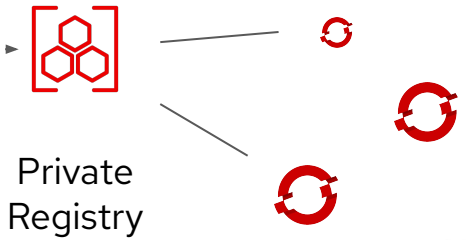
```
Create and publish user-configured mirrors with a declarative
configuration input.

Usage:
  oc-mirror [flags]
  oc-mirror [command]

Examples:
  # Mirror to a directory
  oc-mirror --config mirror-config.yaml file://mirror

  # Mirror to mirror publish
  oc-mirror --config mirror-config.yaml docker://localhost:5000

  # Publish a previously created mirror archive
  oc-mirror --from mirror_seq1_000000.tar docker://localhost:5000

  # Publish to a registry and add a top-level namespace
  oc-mirror --from mirror_seq1_000000.tar
docker://localhost:5000/namespace

Available Commands:
  completion  Generate the autocompletion script for the specified
shell
  ...
```

Operating model 1/3

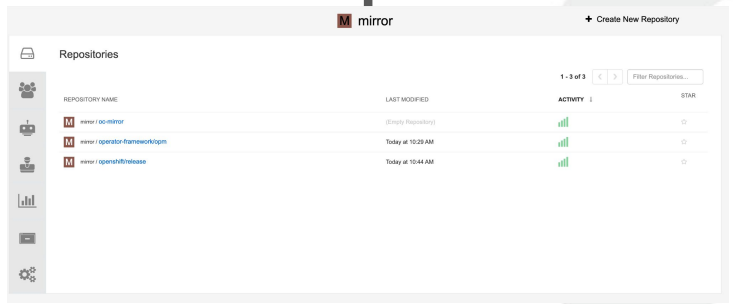
ImageSetConfiguration

oc mirror

ImageSet on disk / removable media



Private Registry




```
---
apiVersion: mirror.openshift.io/v1alpha1
kind: ImageSetConfiguration
mirror:
  ocp:
    channels:
      - name: stable-4.17
        versions:
          - 4.17.0
          - 4.17.3
  operators:
    - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.17
      headsOnly: false
      packages:
        - name: cincinnati-operator
          channels:
            - name: v1
              minVersion: '5.0.2'
              maxVersion: '5.0.2'
```

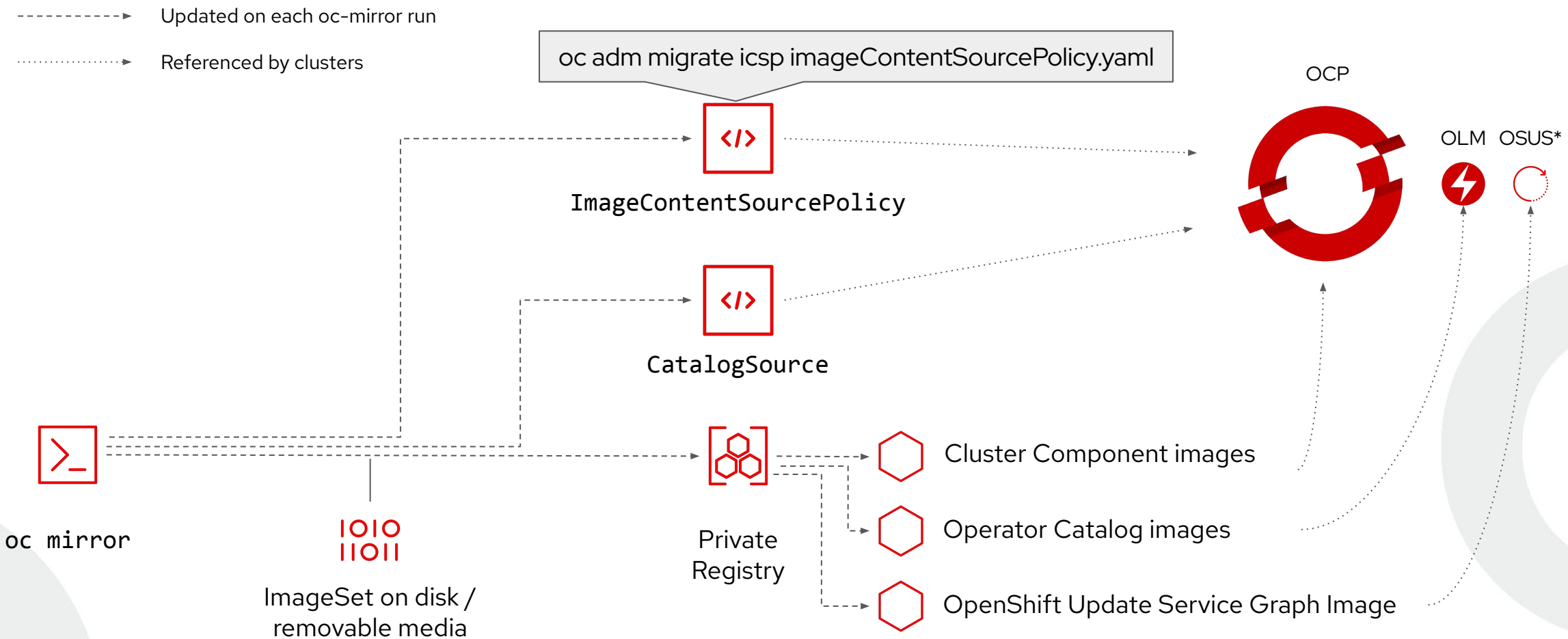
```
ls -hl archives
total 20G
-rw-rw-r--. 1 centos centos 20G Mar  4 08:32 mirror_seq1_000000.tar
drwxr-xr-x. 2 centos centos   6 Mar  4 10:11 oc-mirror-workspace
```

Operating model 2/3

Keeping mirrors up-to-date:

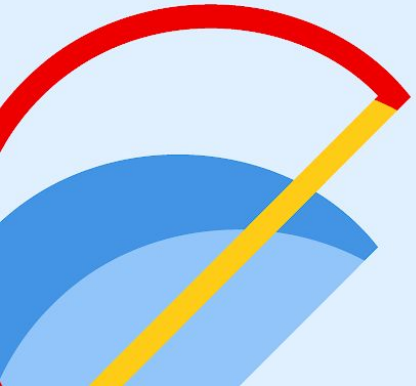
- Run oc-mirror again, with the same or updated config file
  - **declarative way: Remove a version, images would delete!** (**v1**!)
- Differential mirror:
  - will only download newer OCP releases (and required intermediates)
  - will only download newer Operator versions (and required intermediates)
- Produces new catalog images in place for seamless operator updates
- Produces newer cincinnati graph image for OSUS

Operating model 3/3



Red Hat

Run OpenShift Core Installation



Red Hat

OpenShift 4.17 Supported Providers

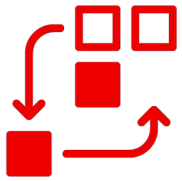
Installation Experiences

aws
Outposts
Wavelength
Local Zones

Azure
Azure Stack Hub
Alibaba Cloud (Tech Preview)

IBM Cloud
IBM Power Systems
IBM Z and
IBM LinuxONE

Bare Metal
Google Cloud
vmware vSphere
ORACLE Cloud
NUTANIX
RED HAT OPENSTACK PLATFORM



Automated

Installer Provisioned Infrastructure

- Auto-provisions infrastructure
- *KS like
- Enables self-service

Full Control

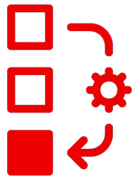
User Provisioned Infrastructure

- Bring your own hosts
- You choose infrastructure automation
- Full flexibility
- Integrate ISV solutions

Interactive – Connected

Assisted Installer

- Hosted web-based guided experience
- Agnostic, bare metal, vSphere and Nutanix
- ISO driven



Local – Disconnected

Agent-based Installer

- Restricted network (disconnected / air -gapped)
- Automatable installations via CLI
- Bare metal, vSphere, SNO
- ISO driven

Red Hat

OpenShift 4.17 Supported Providers

Installation Experiences

aws Outposts Wavelength Local Zones

Azure

Azure Stack Hub

Alibaba Cloud (Tech Preview)

✅ Works

Cloud

IBM Z and IBM LinuxONE

Bare Metal

Google Cloud

vmware vSphere

ORACLE Cloud

NUTANIX

RED HAT OPENSTACK PLATFORM

Automated

Installer Provisioned Infrastructure

- Au… infrastructure
- *K…
- Enables self-service

✅ plain openshift-install command

Full Control

User Provisioned Infrastructure

- Bring your own hosts
- You choose infrastructure automation
- Full flexibility
- Integrate ISV solutions

Interactive – Connected

Assisted Installer

- Host… web based guided … experience
- Agn… bare metal, vSphere and Nutanix
- ISO driven

❌

Local – Disconnected

Agent-based Installer

- Restrict… (disconn… …-gapped)
- Automa… …ations via CLI
- Bare metal, vSphere, SNO
- ISO driven

✅

Red Hat

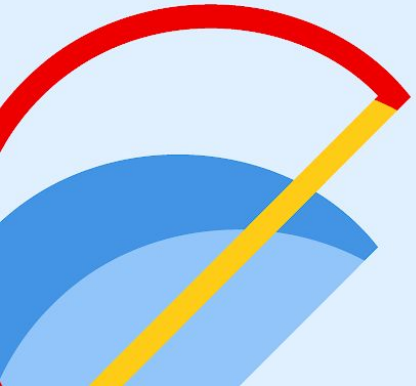
Air-gapped openshift-install

- vSphere IPI example
- `openshift-install create cluster` just works!
- `openshift-install explain` is your friend!
- Disable samples operator via composable OpenShift.

```
apiVersion: v1
baseDomain: example.com
...
platform:
  vsphere:
    apiVIP: 172.16.0.10
...
    clusterOSImage:
http://quay.example.com:8080/rhcos-4.7.0-x86_64-vmware.x86_64.ova?sha2
56=13d92692b8eed717ff8d0d113a24add339a65ef1f12eceeb99dabcd922cc86d1
...
imageDigestSources:
- mirrors:
  - quay.example.com/infra/openshift4
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - quay.example.com/infra/openshift4
  source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  MIIEQzCCAyugAwIBAgIUVwbzbrQNDW3tU2xdZDVsF5VzTlEwDQYJKoZIhvcNAQEL
...
```

Post-Configuration

Just a few...



Red Hat

Configure Root CA to mirror-registry

- additionalTrustedCA: A reference to a config map containing additional CAs that should be trusted during image stream import, pod image pull, openshift-image-registry pullthrough, and builds.

```
apiVersion: config.openshift.io/v1
kind: Image
metadata:
  name: cluster
spec:
…
  additionalTrustedCA:
    name: my-registry-ca
…
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-registry-ca
data:
  registry.example.com: |
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
  registry-with-port.example.com..5000: |
    -----BEGIN CERTIFICATE-----
```

Configure search path to mirror-registry

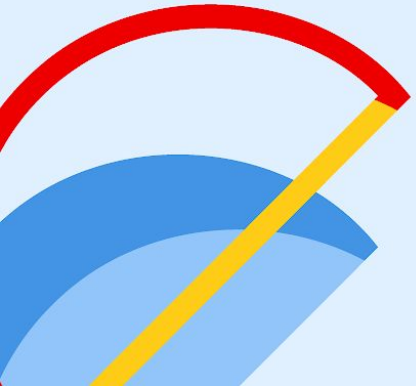
- containerRuntimeSearchRegistries

```
apiVersion: config.openshift.io/v1
kind: Image
metadata:
  name: cluster
spec:
…
  registrySources:
    containerRuntimeSearchRegistries:
     - mirror-registry.example.com
…
```

OperatorHub

- Disabling the default OperatorHub catalog sources
- Apply catalog source from our oc mirror workspace
  - Don’t forget to convert to the ImageContentSourcePolicy (ICSP)
    And apply ImageDigestMirrorSet (IDMS) / ImageTagMirrorSet (ITMS)

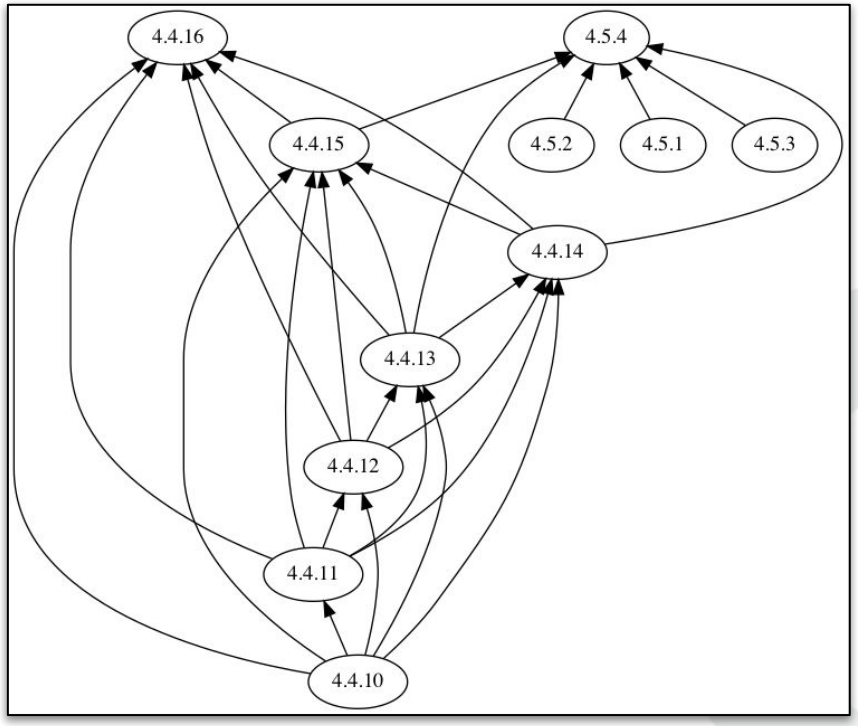
OpenShift Updates



Red Hat

OpenShift update graph

- OpenShift updates have to follow a strict graph/path
- Connected cluster get the graph from
  https://api.openshift.com/api/upgrades_info/v1/graph
- Web tool to get Update Path information:
  Customer Portal → Tools → Customer Portal Labs
  https://access.redhat.com/labs/ocpupgradegraph/update_path/



Red Hat

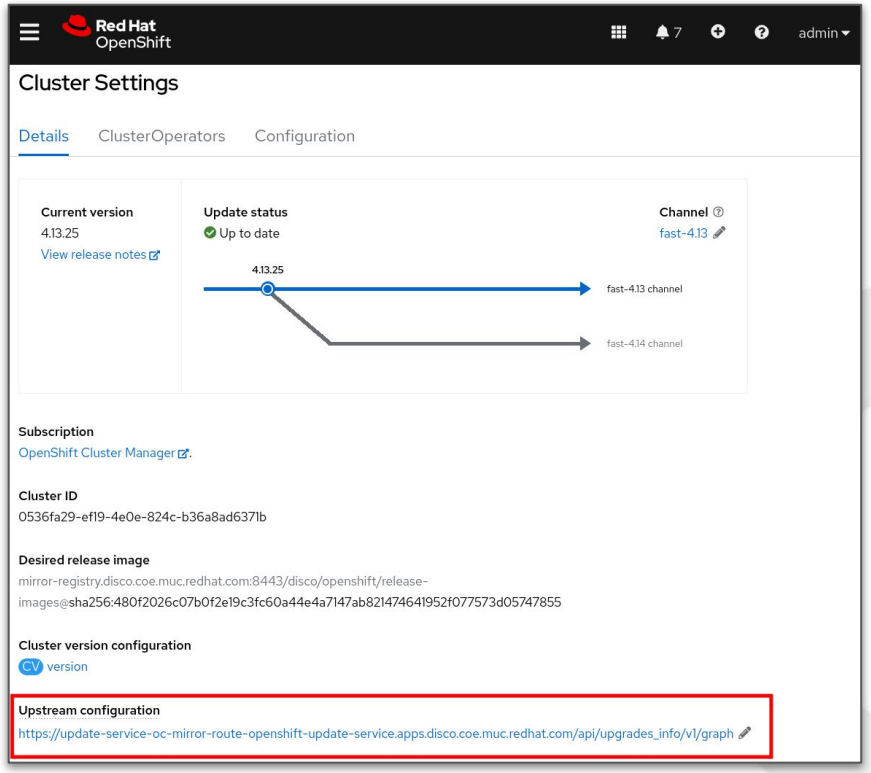
Perform OpenShift updates “manually”

- Follow the update path / graph
- Updating a cluster in a disconnected environment without the OpenShift Update Service

```
oc adm upgrade --allow-explicit-upgrade --to-image <defined_registry>/<defined_repository>@<digest>
```

Let’s do it better: OpenShift Update Service

- Provide the graph to your OpenShift Clusters
- Updating a cluster in a disconnected environment using the OpenShift Update Service

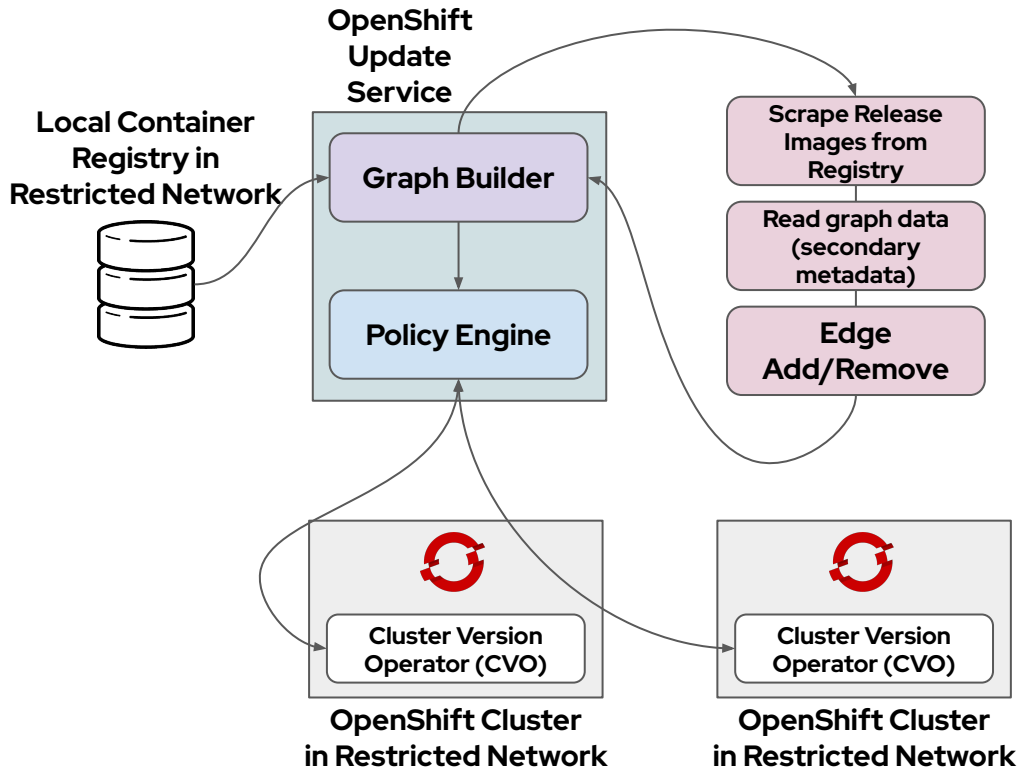


Red Hat

OpenShift Update Service

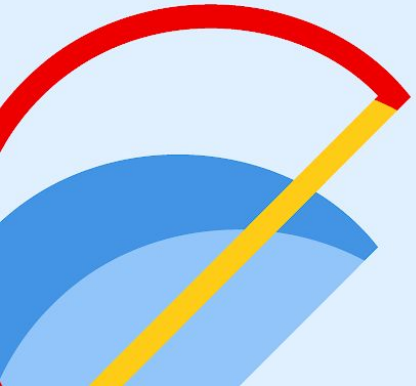
Update manager for your clusters in restricted networks

- OpenShift Update Service (OSUS) is the on-premise release of Red Hat's hosted update service in OCP
- 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



Red Hat

Closeout & Questions



Red Hat

Red Hat Summit

Connect

Thank you

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat