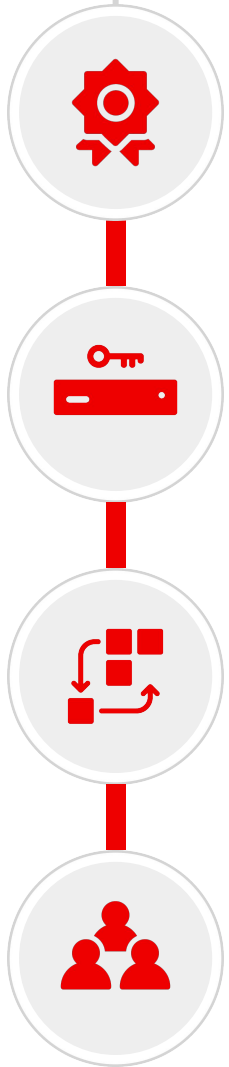
Red Hat Quay

RED HAT QUAY TECHNICAL DECK

Detailed Feature Descriptions and Operational Guidance

Red Hat

Red Hat Quay



Industry-leading, **trusted, and open source registry platform** operating at scale since 2014

Built to **efficiently manage content** under governance and security **controls** globally

Runs **everywhere**, easy to **integrate** and **automate** but works best with **OpenShift**

Developed in **collaboration** with a broad open source, customer, and ecosystem **community**

Red Hat Quay Key Features

SCALABILITY

Massive Scale Testing Quay.io
Real Time Garbage Collection
Automated Squashing

BUILD AUTOMATION

Seamless Git Integration
Build Workers
Webhooks

INTEGRATION

Extensible API
Webhooks, OAuth
Robot Accounts

REGISTRY

High Availability
Full Standards / Spec Support
Long-Term Protocol Support
Application Registry
Enterprise Grade Support
Regular Updates

SECURITY

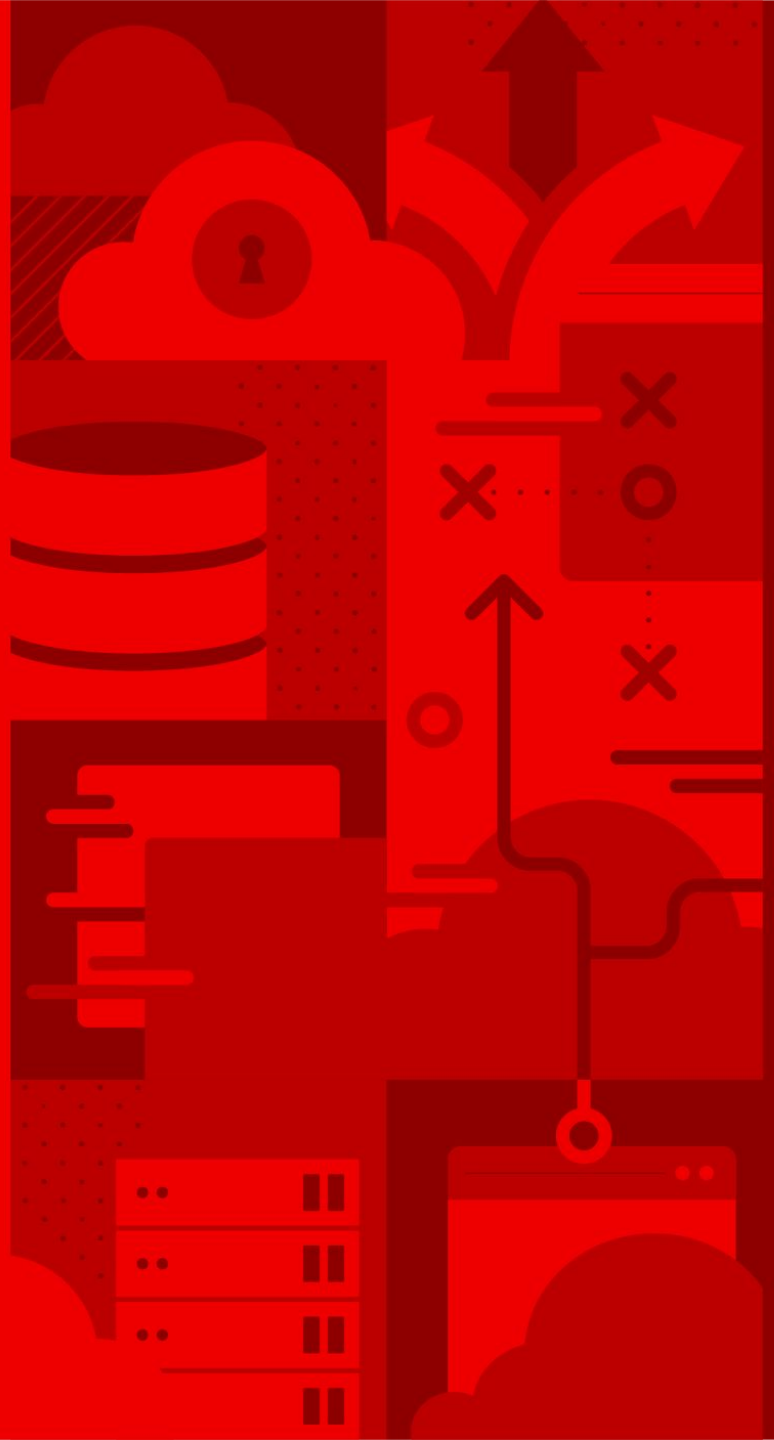
Vulnerability Scanning
Logging & Auditing
Notifications & Alerting

CONTENT DISTRIBUTION

Geo-Replication
Repository Mirroring
Air-Gapped Environments

ACCESS CONTROL

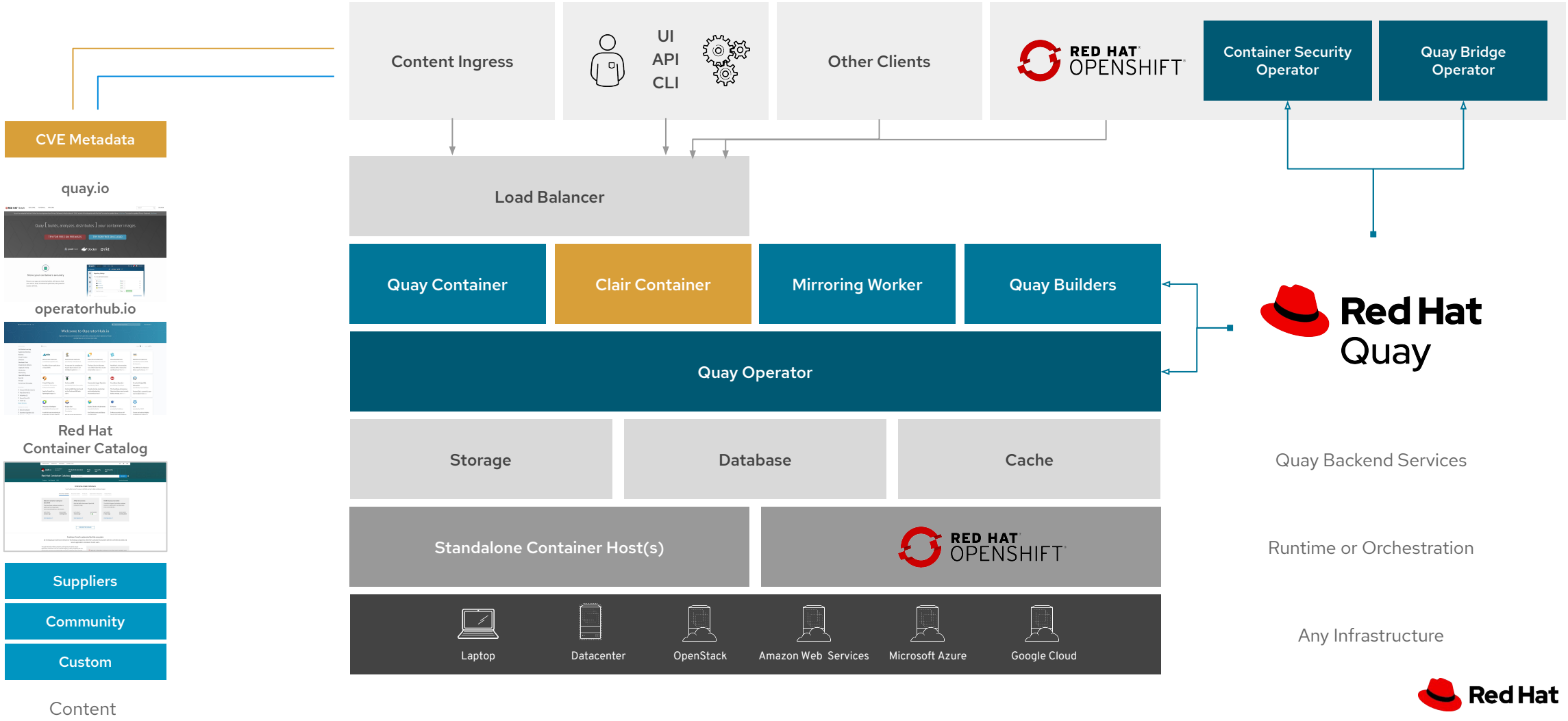
Authentication Providers
Fine-Grained RBAC
Organizations & Teams



Quay Architecture

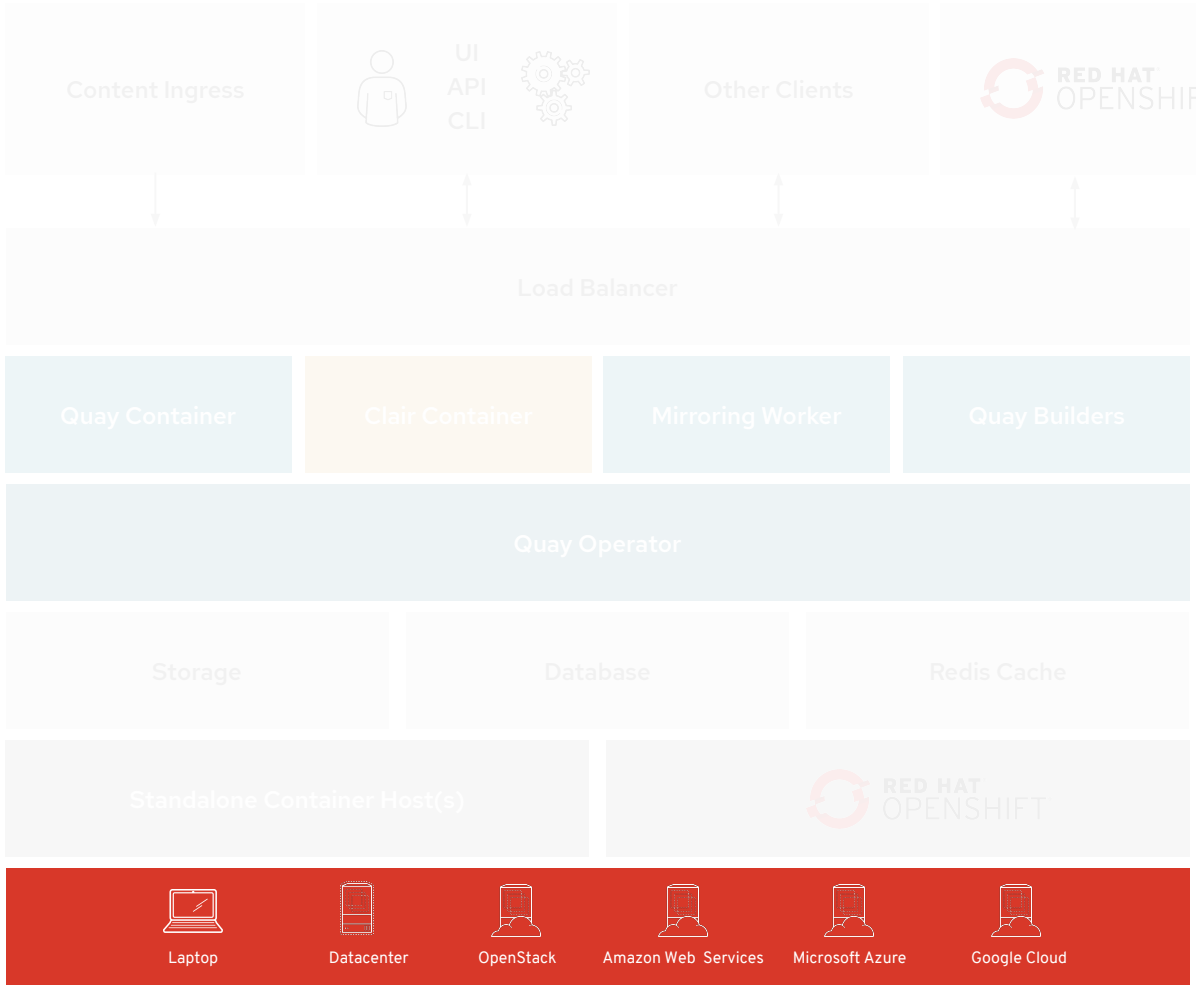
Red Hat

Red Hat Quay Architecture



Red Hat

Prerequisite: Infrastructure



Quay runs on any* physical or virtual infrastructure both on-premise or public cloud**

Quay scales from an all-in-one setup on a developer laptop over running highly available on OpenShift up to geographically dispersed setup across multiple availability zones and regions

Recommendation: if Quay runs on public cloud -> use the public cloud services for Quay backend services to ensure proper HA and scalability

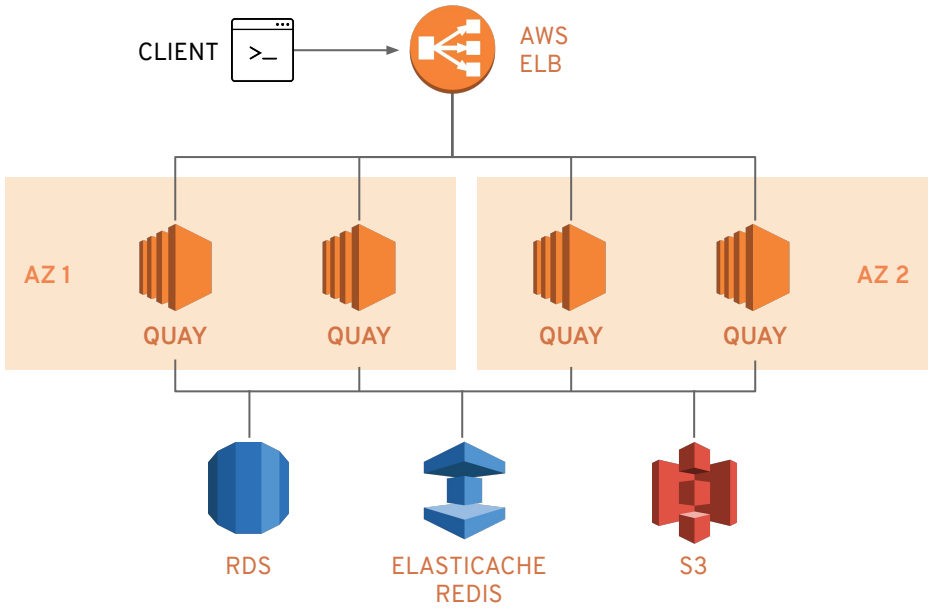
* Further details can be found in the Quay 3.x tested configuration matrix: https://access.redhat.com/articles/4067991

** Further details can be found in the Quay Support Policy: https://access.redhat.com/support/policy/updates/rhquay/policies

Red Hat

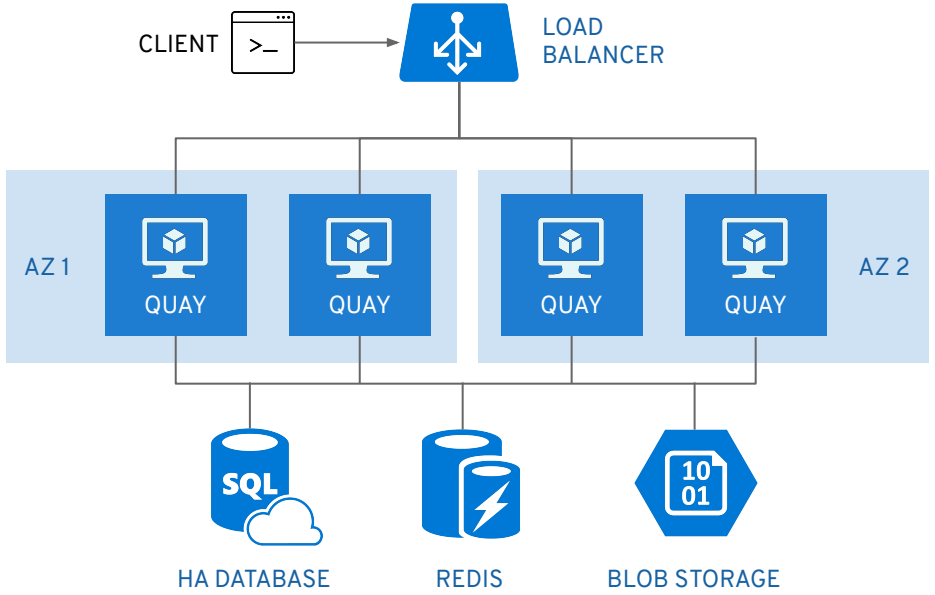
Running Red Hat Quay on Public Cloud

Full list of tested and supported configurations can be found inside the Red Hat Quay Tested Integrations Matrix: https://access.redhat.com/articles/4067991



If Quay runs on AWS you can use:

- AWS Elastic Load Balancer
- AWS S3 (hot) blob storage
- AWS RDS database
- AWS ElastiCache Redis
- EC2 VMs recommendation: M3.Large or M4.XLarge

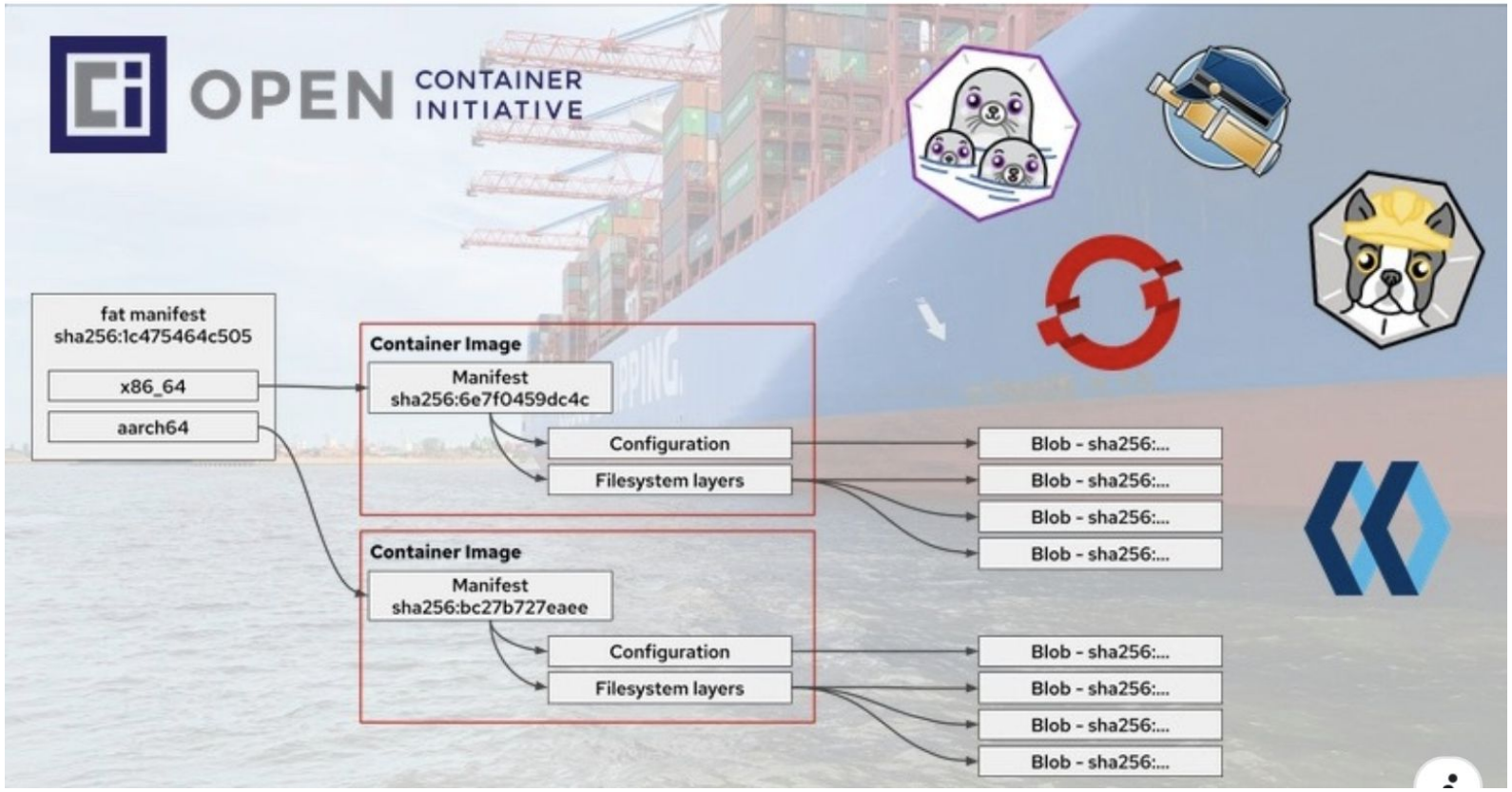


If Quay runs on MS Azure you can use:

- Azure managed services such as HA PostgreSQL
- Azure Blob Storage must be hot storage (not Azure Cool Blob Storage)
- Azure Cache for Redis

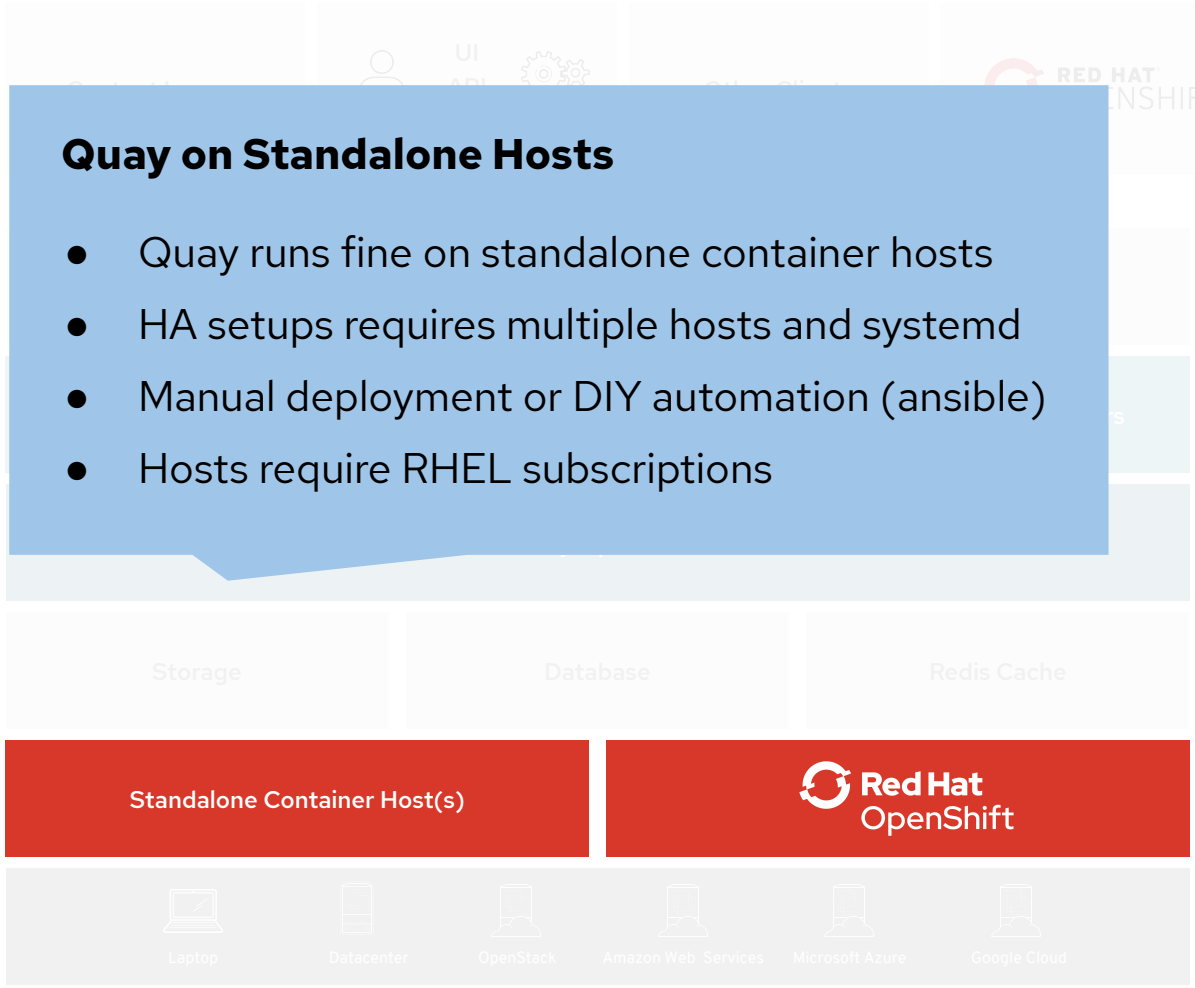
Red Hat

Multi Architecture Containers



Red Hat

Prerequisite: Container Runtime or Orchestration



Quay on Standalone Hosts

- Quay runs fine on standalone container hosts
- HA setups requires multiple hosts and systemd
- Manual deployment or DIY automation (ansible)
- Hosts require RHEL subscriptions

Benefits of running Quay on OpenShift:

- Run Quay where you run all other container workloads and services
- Seamless deployment and day2 management of Red Hat Quay via the Quay operator and OLM
- Leverage all orchestration and management capabilities of Red Hat OpenShift incl. Operator Lifecycle Mgt (OLM), monitoring, dashboards, ...
- Quay can run on OpenShift infra nodes (no further subscriptions required)

Full list of tested and supported configurations can be found inside the Red Hat Quay Tested Integrations Matrix: https://access.redhat.com/articles/4067991

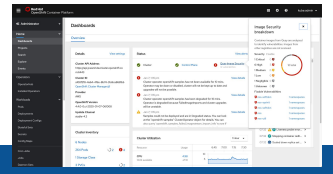
Red Hat

Red Hat Quay works best with OpenShift

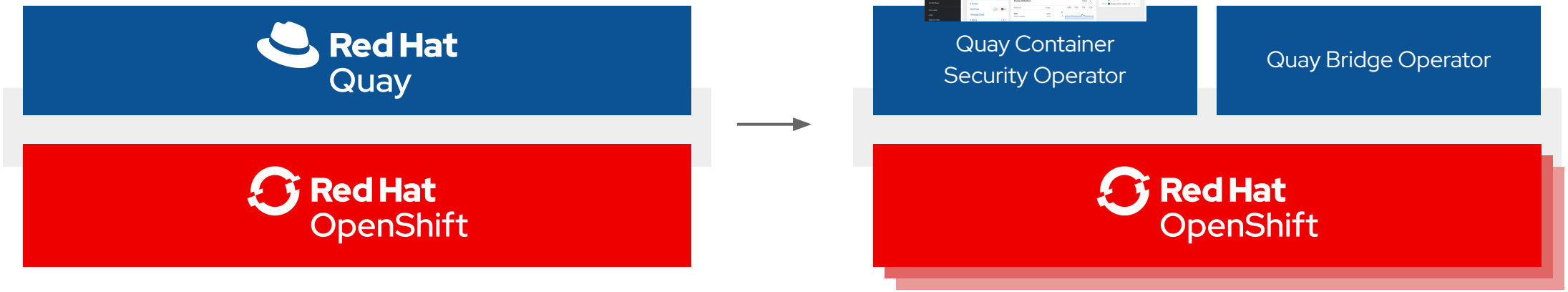
Red Hat Quay runs on any infrastructure but **runs best on OpenShift**

The **Quay Operator** ensures seamless deployment and management of Quay running on OpenShift

CSO brings Quay / Clair vulnerability data into the OpenShift Console



The **Quay Bridge Operator** ensures seamless integration and user experience for using Quay **with** OpenShift



Quay serves content to **one or many OpenShift clusters**, wherever they're running.

With or without using the OpenShift internal registry but leveraging all OpenShift capabilities.

Red Hat

Benefits of running Quay on OpenShift

- **Zero to Hero** - Simplified deployment of Quay and associated components means that you can starting using the product immediately
- **Scalability -** Leverage cluster compute capacity to manage expected demand
- **Simplified Networking** - Diverse ingress options using well established patterns for any application deployed on the platform
- **Centralized configuration management** - Configurations stored in etcd provide a centralized source of truth
- **Repeatability** - Consistency regardless of the number of replicas of Quay / Clair
- **Expanded Options** - Additional solutions that are specifically designed to take advantage of an OpenShift deployment

Red Hat

Quay Sizing Recommendations

- Scalability of Quay is one of its key strengths since the same code base runs on a developer laptop with a PoC sizing, as a typical mid-size deployment with ~2,000 users serving content to dozens of kubernetes clusters up to thousands of clusters world-wide (Quay.io)
- As for any other product there are no “typical sizing recommendations” since sizing heavily depends on a multitude of factors (no of users / images / concurrent pulls and pushes, etc.)
- **Stateless** components can be **scaled-out** (will cause more load on backend services though)
  - Auto-scaling on k8s deployments currently tech-preview, future via Quay operator
  - Note: Scaling out stateless components will add load to stateful components
- **Minimum** requirements as documented in the Quay Product Docs:
  - Quay: min 4GB, recommended 6GB, 2 or more vCPUs
  - Clair: recommended 2GB RAM, 2 or more vCPUs
  - Clair database requirements for security metadata: min 200MB
  - Storage depends on no of images, recommended min 30GB

Red Hat

Quay Sample Sizings

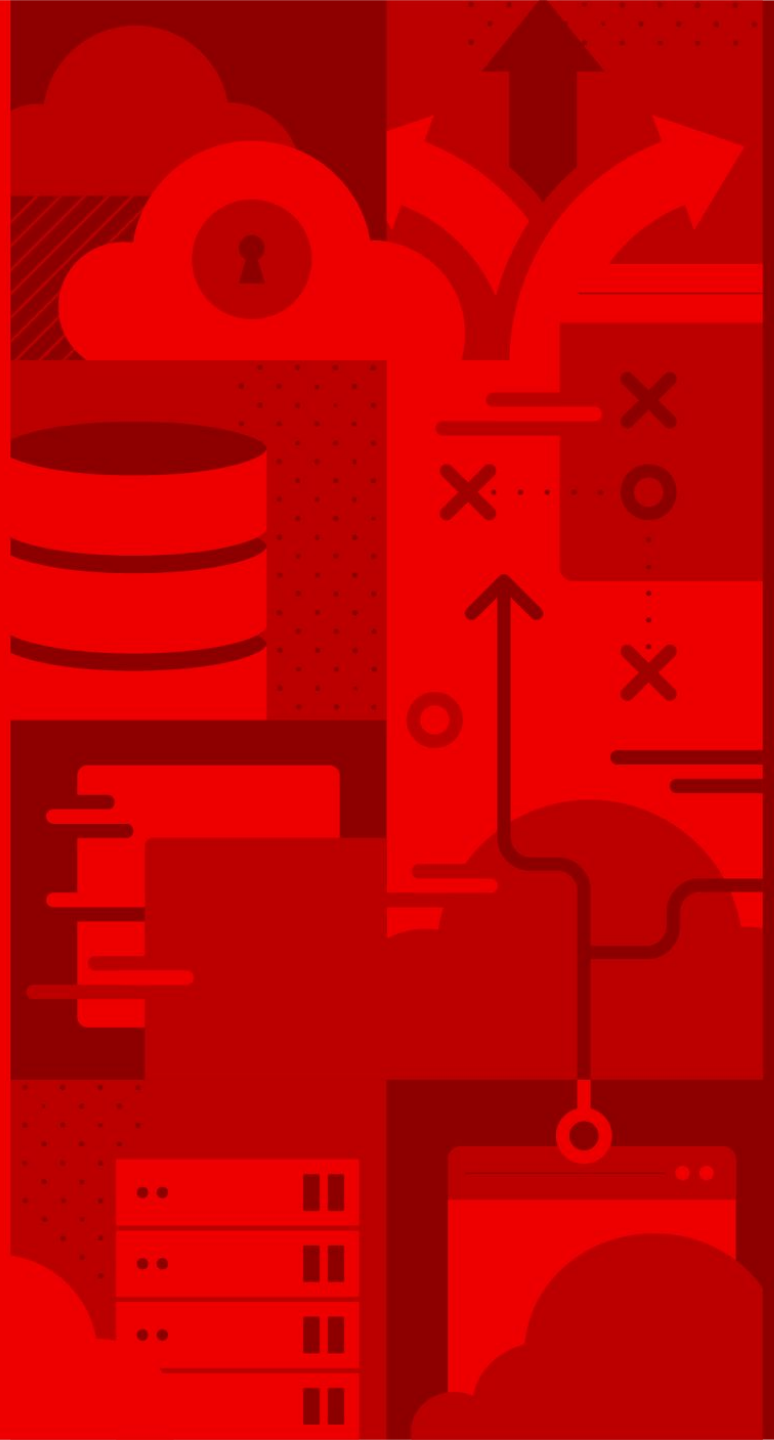
Note: Those are sample sizings of existing Quay deployments. Whether a specific deployment runs fine with the same metrics depends on too many other factors as well not shown here.

Metric	Minimum Setup	Mid/Large Setup	XXXXL (Quay.io)
No of Quay containers by default	1	4	15
No of Quay containers max at scale-out	N/A	8	30
No of Clair containers by default	1	3	10
No of Clair containers max at scale-out	N/A	6	15
No of mirroring pods[1] (to mirror 100 repos)	1	5-10	N/A
Database sizing		4-8 Cores / 6-32 GB RAM	32 cores 244GB, 1+ TB disk
Storage Backend Sizing	10-20 GB	1 - 20 TB	50+ TB up to PB
Redis Cache Sizing[2]		2 Cores / 2-4 GB RAM	4 cores / 28 GB RAM
Underlying node sizing (phys or virtual)	2-4 Cores / 6 GB RAM	4-6 Cores, 12-16 GB RAM	Quay: 13 cores 56GB RAM Clair: 2 cores 4 GB RAM

[1] see repository mirroring section for further details on sizing & related recommendations

[2] since Redis cache is only used for Quay builders the sizing can be very tiny if builders aren't used

Red Hat



Quay and OpenShift

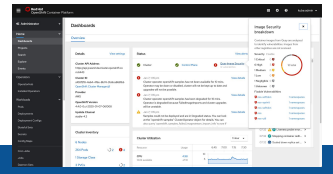
Red Hat

Red Hat Quay works best with OpenShift

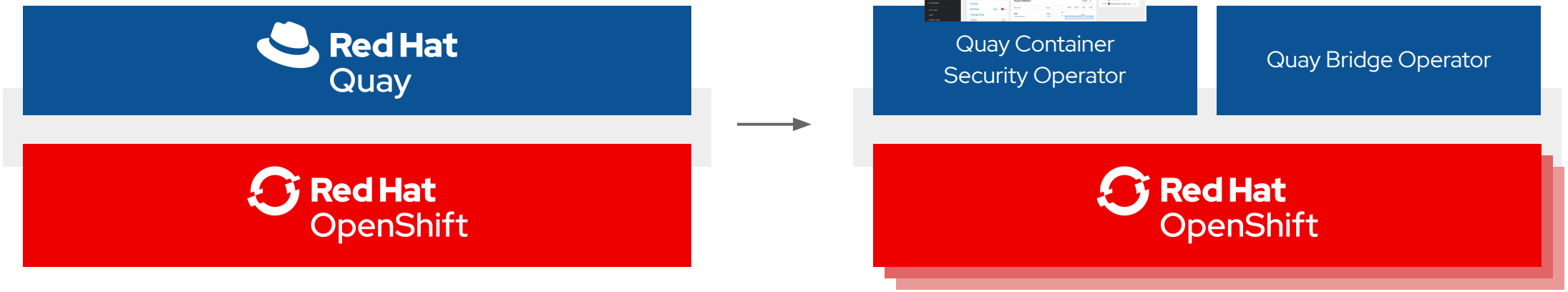
Red Hat Quay runs on any infrastructure but **runs best on OpenShift**

The **Quay Operator** ensures seamless deployment and management of Quay running on OpenShift

CSO brings Quay / Clair vulnerability data into the OpenShift Console



The **Quay Bridge Operator** ensures seamless integration and user experience for using Quay **with** OpenShift



Quay serves content to **one or many OpenShift clusters**, wherever they're running.

With or without using the OpenShift internal registry but leveraging all OpenShift capabilities.

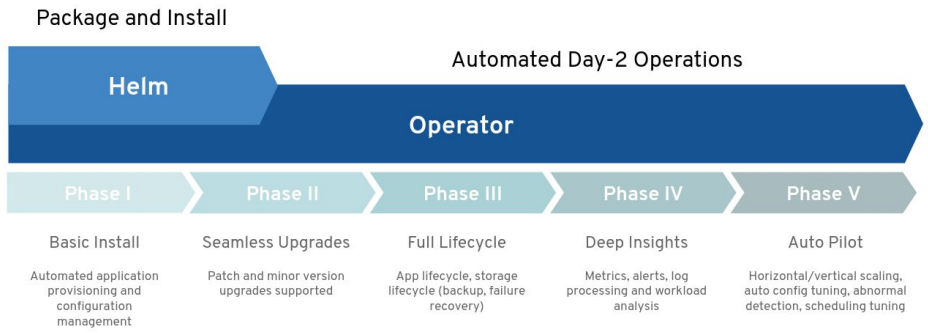
Red Hat

Benefits of running Quay on OpenShift

- **Zero to Hero** - Simplified deployment of Quay and associated components means that you can starting using the product immediately
- **Scalability -** Leverage cluster compute capacity to manage expected demand
- **Simplified Networking** - Diverse ingress options using well established patterns for any application deployed on the platform
- **Centralized configuration management** - Configurations stored in etcd provide a centralized source of truth
- **Repeatability** - Consistency regardless of the number of replicas of Quay / Clair
- **Expanded Options** - Additional solutions that are specifically designed to take advantage of an OpenShift deployment

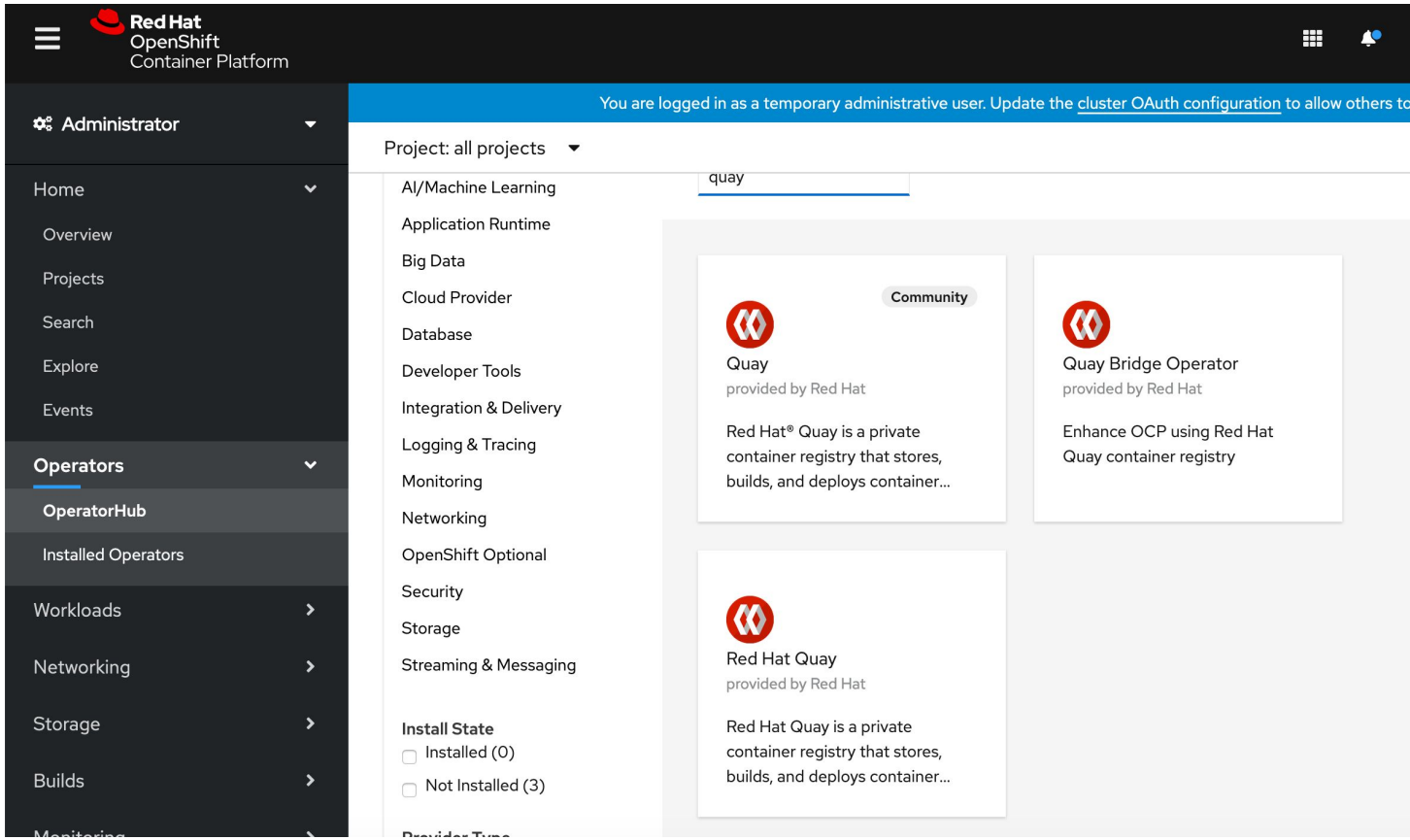
Red Hat

Quay - Focus on Operators



Focus and direction for the Quay product are kubernetes operators and running Quay on OpenShift / kubernetes given the advantages of operators compared with its alternatives

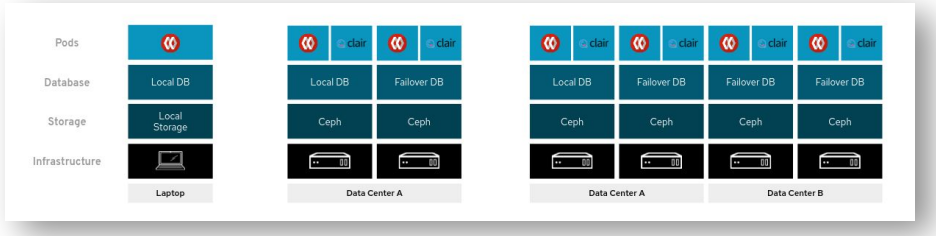
Maintaining another deployment and management tooling for non-k8s deployments is not feasible and not aligned to our prioritization and roadmap (Quay v4 will run on k8s by default)

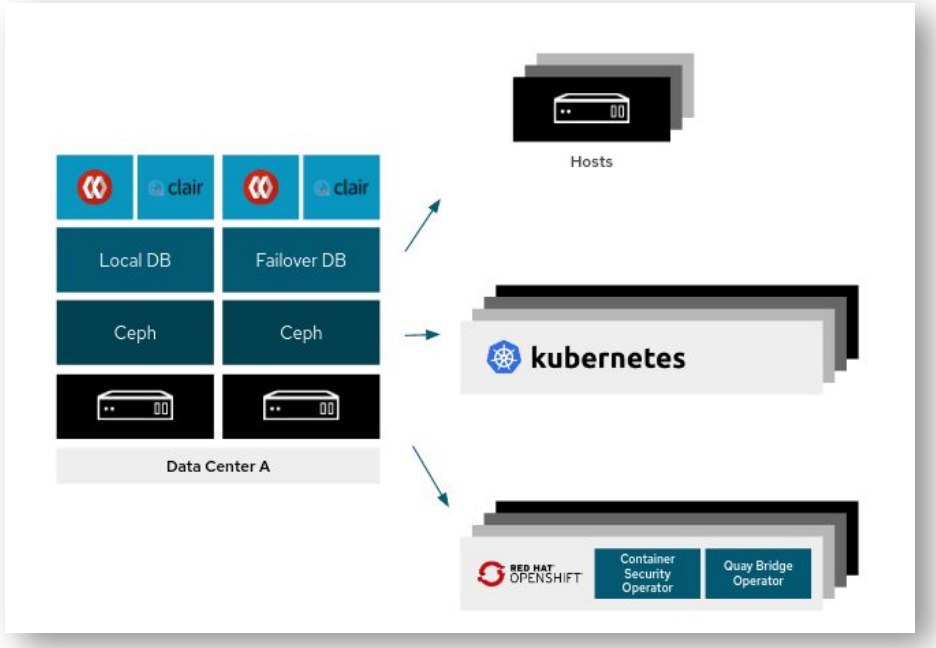


Red Hat

Quay Deployment Examples

- Quay can run on standalone container hosts or OpenShift (recommended)
- A Quay deployment can be distributed across multiple DCs or even OCP clusters (geo-repl)
- Typically Quay is used for **more than one / many OpenShift clusters**
- Components which can run **on-cluster**: Quay, Clair, mirroring workers
- Components which should / must run **off cluster** (today): Quay builders, databases (if not an operator), storage





Red Hat

Quay Builders on OpenShift

- Quay builders require a docker runtime and do not work with buildah yet
- As of today (Quay 3.3) the Quay builders can’t run on OpenShift 3 + 4 and therefore should run off-cluster (also for security reasons)
- Preferably on bare metal due to performance reasons
- Technically Quay builders can run on OCP 4 bare metal, documentation and a small enhancement of the Quay config app targeted for Quay 3.4

Red Hat

OpenShift and External Registries

- OpenShift can utilize an external container registry as a source for operations on the platform
  - Build source and output
  - Runtime content
- From an OpenShift point of view, Quay as with any external registry is not as deeply integrated as the OpenShift internal registry
  - No automatic RBAC isolation based on OpenShift cluster permissions
  - No real-time automatic ImageStream notifications and updates

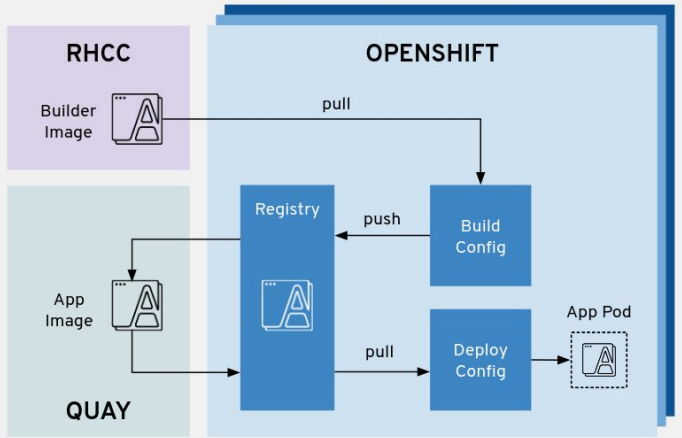
Red Hat

Using Quay With or Without Internal Registry

- Quay can be used as an external registry in front of an entire OpenShift cluster with its registry
- Quay also can be used directly without using the internal registry which requires a couple of changes (secrets, build and deployment configs) which are **partially** done automatically by QBO

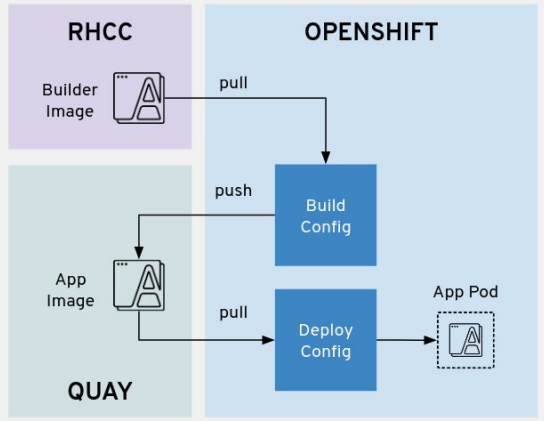
Quay as Upstream Registry with OpenShift

- Images pulled from Quay into the integrated OpenShift registry
- Images are pushed to the integrated OpenShift registry, and synced externally with Quay



Quay as OpenShift Registry

- Images are pushed directly by builds to Quay
- Images are pulled directly from Quay



Red Hat

Using Quay for OpenShift Builds

- Quay (and any) external registry can be integrated into the OpenShift build process as both a location to store images built in the platform and as an image source for builds.
- Images that are produced by an image build within OpenShift can be stored in a remote registry instead of the OpenShift integrated registry.
- The input and output sections of the BuildConfig specification defines the image source
  - An ImageStream or direct DockerImage reference can be used
  - Secret must be specified when accessing protected registries

```
"output": {
    "to": {
        "kind": "DockerImage",
        "name":
"external-registry.example.com:5000/publicrepo/sample-app:latest"
    },
    "pushSecret": {
        "name": "external-registry"
    },
```


Red Hat

Quay as the Registry for OpenShift

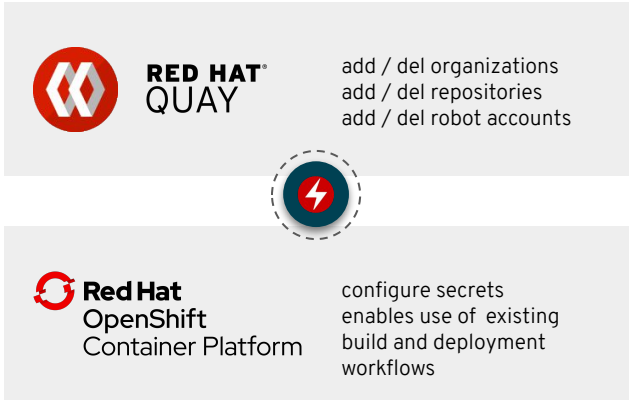
- Configurations in OpenShift may need be specified in order to use Quay as an image registry
  - Image configuration resources define the list of trusted registries and their associated details
  - Partially driven through the MCO
- Image Push and Pull Secrets must be specified when accessing protected image repository
- Quay Bridge Operator begins the journey toward Quay acting as external registry with integration to OpenShift
  - Future versions of OpenShift and Quay will increase the integration between these components

Red Hat

Quay and OpenShift Entity Mapping

Quay Entity	OpenShift Entity
Organization	Project/Namespace
Repositories	ImageStream
Robot Accounts	ServiceAccount
Quay Team	Group
Build (docker build trigger by git actions)	Build (s2i, docker build, jenkins, tekton)

Red Hat



Quay Bridge Operator

Operator which runs on OpenShift and integrates Quay into OpenShift workflows similar to the existing internal registry experience.

Built in strong collaboration with the Red Hat internal and customer communities

Supports multi-cluster setups, features OCP build integration

Red Hat

Quay and OpenShift Mapping

- Each OpenShift **namespace** results in an **organization** within Quay (automatically created)
- Each **ImageStream** within an namespace results in a new **repository** within Quay
- The three primary service accounts (Builder, Deployer and Default) result in **Robot accounts** in each organization with the following permissions
  - Builder - Write | Default - Read | Deployer - Read
- Supports **multi-cluster setups** (no name collisions)
- **Secrets** from each Robot account within an org are created in each OpenShift namespace
- **Service accounts** are configured to leverage these secrets as mountable secrets in order to push images as well as pull secrets in order to pull images
- **Builds** resulting in new images getting pushed to Quay update the **ImageStream**
- If enabled globally the OCP internal registry will no longer be used

Red Hat

Quay and OpenShift - Bridge Operator

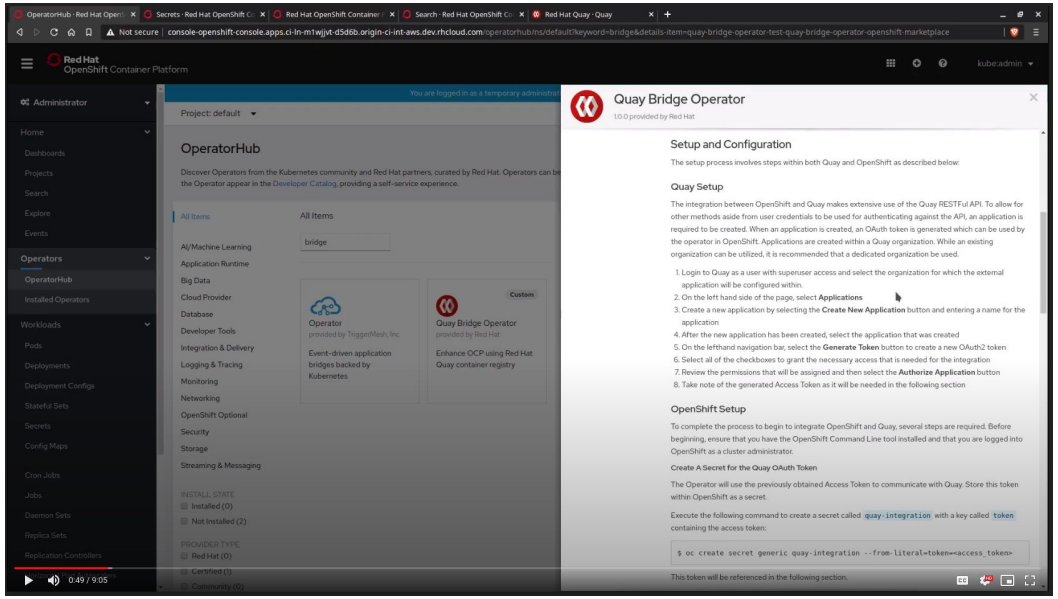
- Operator which runs on OpenShift and automates some integration pieces similar to our existing internal registry user experience
- Sample use cases:
  - New project in OpenShift -> new organization in Quay + robot accounts + configure pull and push secrets in OpenShift
  - New app in OpenShift -> build results pushed to Quay using the robot accounts and push secrets created earlier
  - New deployment in OpenShift -> pull image from Quay using the pull secret
  - Delete a project in OpenShift -> delete repositories, robot accounts and org in Quay
  - OpenShift cluster name = CRD config option -> supports multiple OCP clusters in Quay

Red Hat

Quay and OpenShift - Bridge Operator

Deploy the Quay Bridge Operator using the Operatorhub in OpenShift 4.

There are some prerequisite steps before you can deploy the QBO to an OpenShift cluster (as of Quay 3.3 and OCP 4.4)



Check out the Quay Bridge Operator Demo:
https://drive.google.com/open?id=1EcDGK9gRwDTFH0etdHyq_zI-nixgyZFX

Red Hat

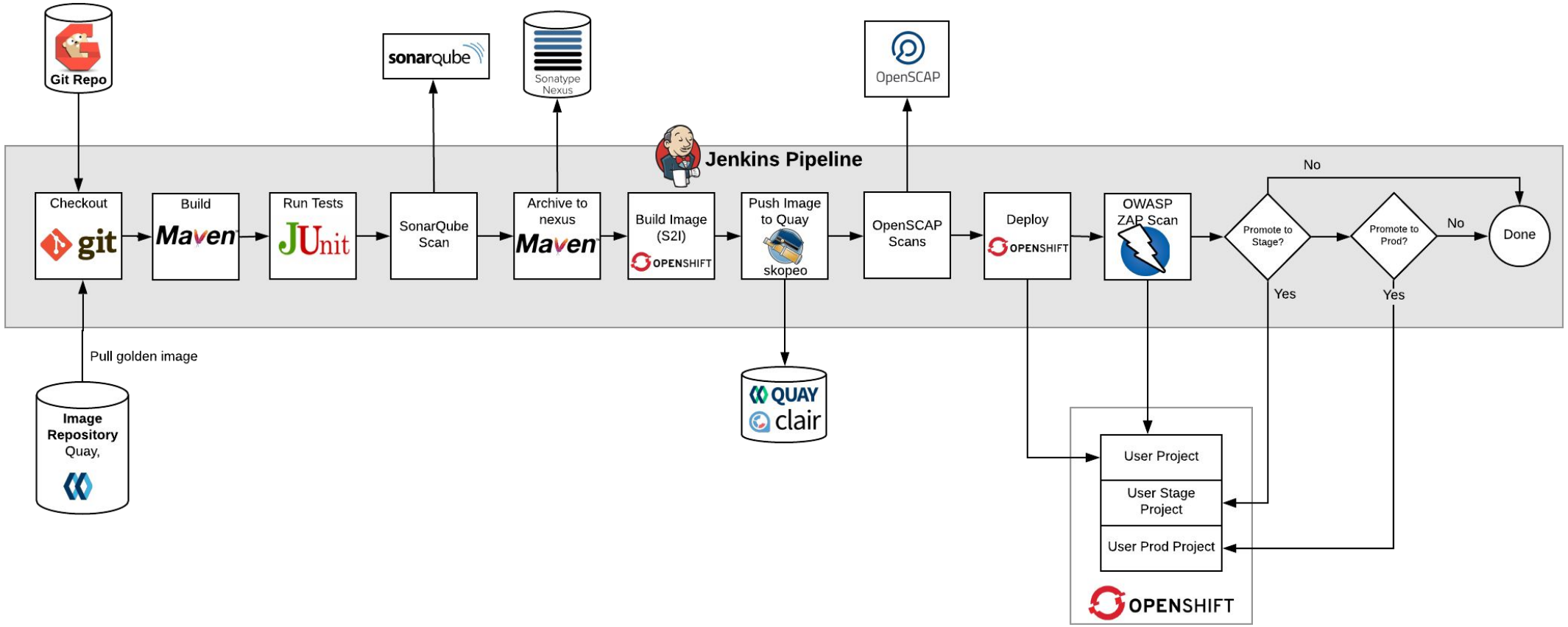
Quay and OpenShift - ImageStreams

- Abstraction of container images within OpenShift
- Images referenced in ImageStreams may reside within the internal OpenShift registry or an alternate registry, such as Quay
- Alternate registries lose many of the benefits of the internal registry
  - Native RBAC integration
  - Automatic notifications when new images and tags are available
    - ImageStreams can be *Scheduled* to poll for changes to existing images

Red Hat

Quay and OpenShift - Pipeline Integration

Quay and Clair can be integrated into existing CI/CD pipelines



Quay triggering Redeployments on OpenShift

- Quay Bridge Operator automatically updates ImageStreams upon the completion of a build and push to Quay
  - ImageStream triggers allow for automatically trigger new deployments within OpenShift
- OpenShift ImageStreams can refer to images in Quay that are not managed by the Quay Bridge Operator
  - Updates to images will not be reflected by default in OpenShift
  - ImageStreams can be “scheduled” to automatically query registry for image updates
- Additional integrations can also be created
  - Quay’s built in notifications using Webhook POST can participate in a CI/CD flow

Red Hat

Quay triggers Rebuilds on OpenShift

- New images arriving in Quay can be used as input sources for builds in OpenShift
- Images referenced by ImageStreams can automatically trigger builds when a change is detected
  - New images in Quay may not be immediately reflected in an ImageStream compared to the internal registry
  - Setting an ImageStream to be “Scheduled” will poll the remote source on an interval for changes
  - Updated image will be used as the base for the OpenShift build

Red Hat

Red Hat Quay Container Images

Quay

Core Registry Image

Also contains the repo mirroring worker component

Clair

Vulnerability Scanning engine deeply integrated into Quay

Stable: v2 / Beta: v4

Image Name Change:
Clair v2: clair-jwt / Clair v4: clair

Builder

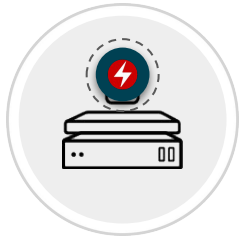
Triggered by git actions and builds container images

Requires docker runtime

Requires privileges

Red Hat

Red Hat Quay Operators



Quay Operator

Automates the initial deployment of Quay, Clair and backends on OpenShift

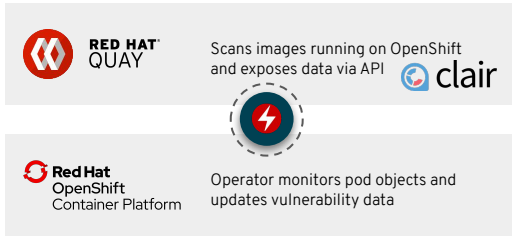
Simplifies Quay installation & Day 2 operations

Configures all relevant OpenShift objects (routes, secrets, certificates, etc.)

Runs (only) on the OpenShift Cluster Quay is running on

Not needed with Quay.io

Added in Quay 3.1



Container Security Operator

Operator which runs on OpenShift and fetches vulnerability data from Quay / Clair if Kubernetes pod objects change and stores it in CRs

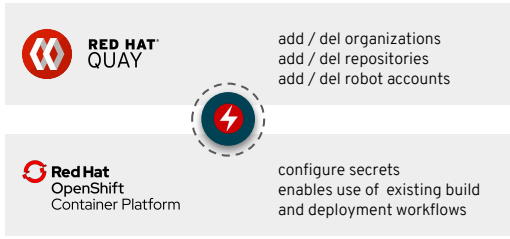
Synchronous Updates of vulnerability information

Vulnerability data shown in OpenShift Console

Runs on every OpenShift cluster Quay is serving content to

Works with Quay.io

Added in Quay 3.2



Quay Bridge Operator

Operator which runs on OpenShift and integrates Quay into OpenShift workflows similar to the existing internal registry experience.

Built in strong collaboration with Red Hat internal and customer / open source communities

Runs on every OpenShift cluster Quay is serving content to

Does **not work** with Quay.io

added in Quay 3.3

Note: All Quay Operators are fully support on OpenShift 4 only due to OLM in Tech Preview on OCP 3

Red Hat

How to Pull Red Hat Quay Software

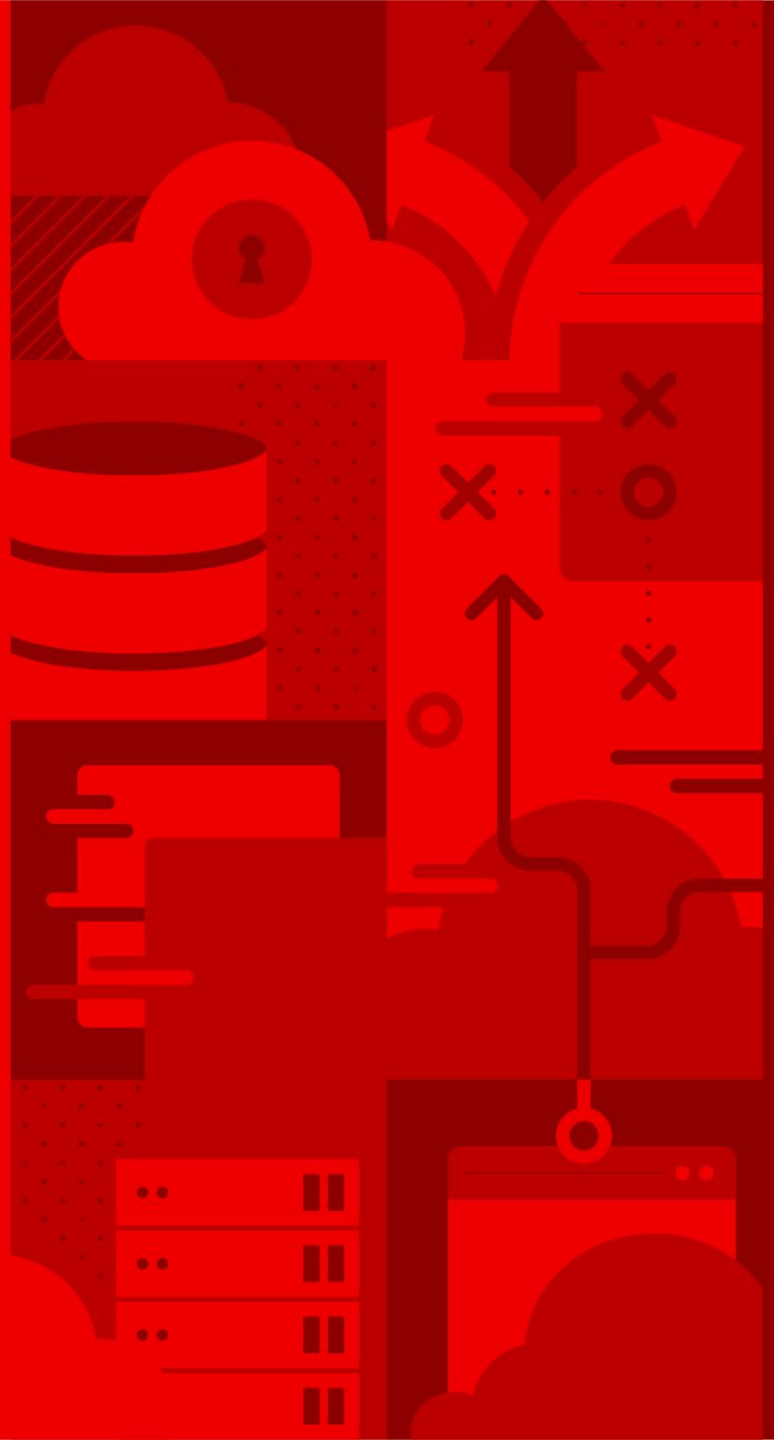
Red Hat Quay Container Images

- Recommend way to deploy Quay is using the Quay Setup Operator
- Simply deploy via the embedded OperatorHub in OCP Console
- Air-gapped or non-OpenShift infra: you need to pull the images / operators
- Currently all three images are distributed via Quay.io only
- Requires a pull secret documented here: https://access.redhat.com/solutions/3533201

Red Hat Quay Operators

- All 3 Quay Operators are shown inside embedded OperatorHub in OCP Console
- Instructions for non-OpenShift k8s deployments shown in Operatorhub.io
- Deployment, configuration and lifecycle management via OLM on OpenShift
- Air-gapped environments: follow the instructions for air-gapped CatalogSources
- Enhanced support for air-gapped deployments targeted for Quay 3.4+

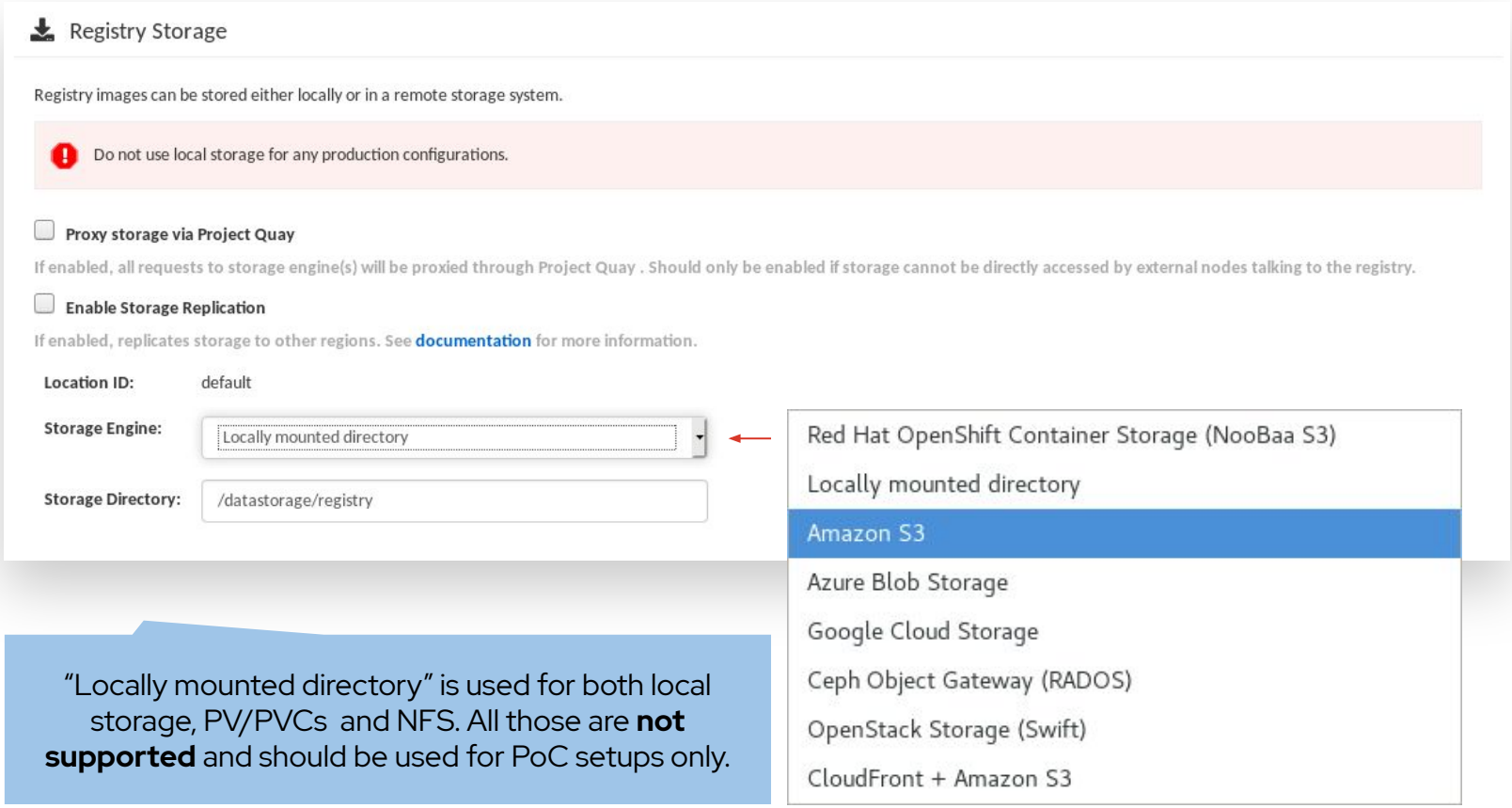
Red Hat



Quay Registry Storage

Red Hat

Red Hat Quay - Registry Storage Configuration



"Locally mounted directory" is used for both local storage, PV/PVCs and NFS. All those are **not supported** and should be used for PoC setups only.

- Storage configuration happens during Quay deployment (Config UI or Quay Operator)
- Storage must preexist
- 3 major options:
  - Proxy Storage
  - Geo-Replication
  - Storage Engine

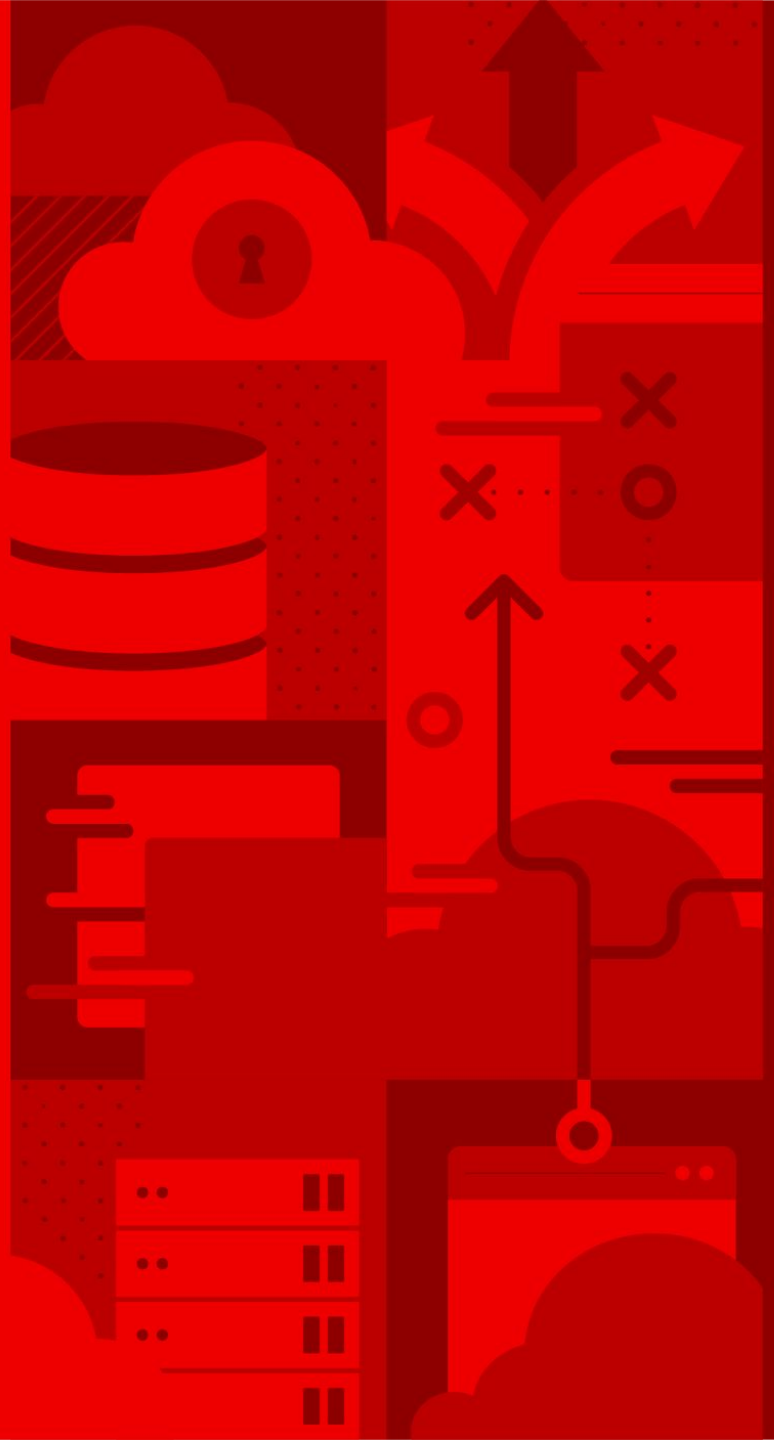
Red Hat

Red Hat OpenShift Container Storage 4

Red Hat OpenShift Container Storage 4 is now an **on-prem storage backend** of Quay.

We're using the embedded **Multi-Cloud Object Gateway object service, based on NooBaa project**, as a storage engine with Quay.

Note: this is a GA version of the Multi-Cloud Object Gateway.

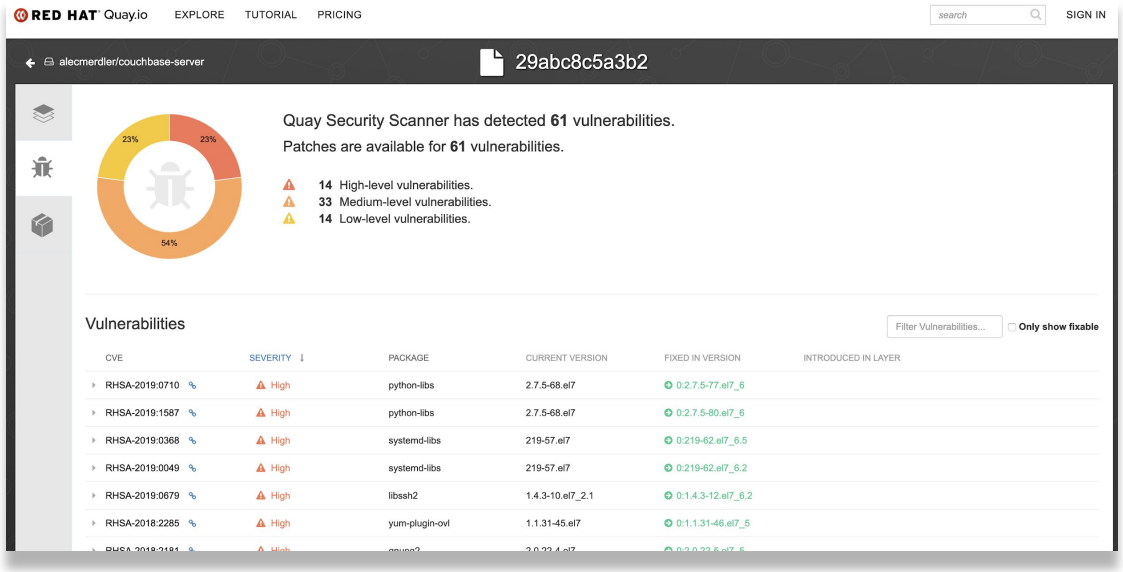


Built-In Vulnerability Scanning via Clair

Red Hat

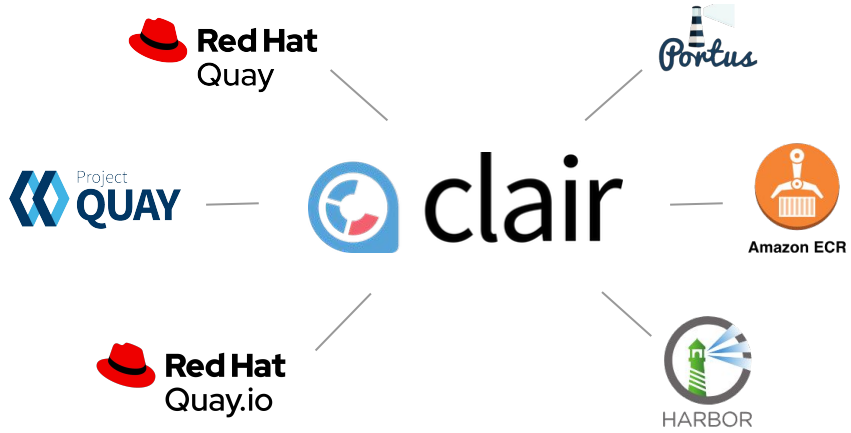
Clair Overview

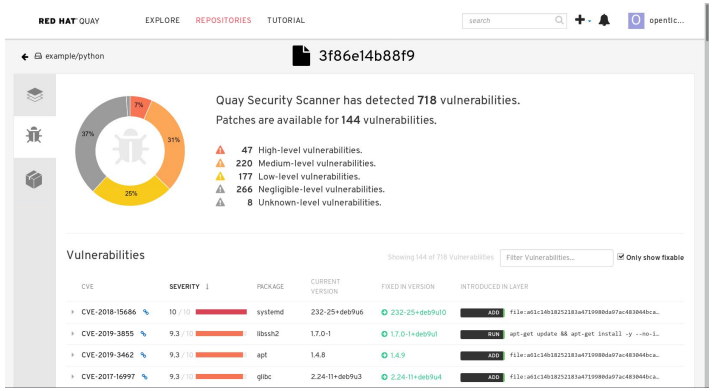
- Clair is an open source tool for static analysis of vulnerabilities in application containers
- Developed by CoreOS for Quay and it's massive scale usage at Quay.io
- Used by various other projects and third party products
- Upstream Repositories: https://github.com/quay/clair



Red Hat

clair v4





Clair v4 (Tech Preview with Quay 3.3)

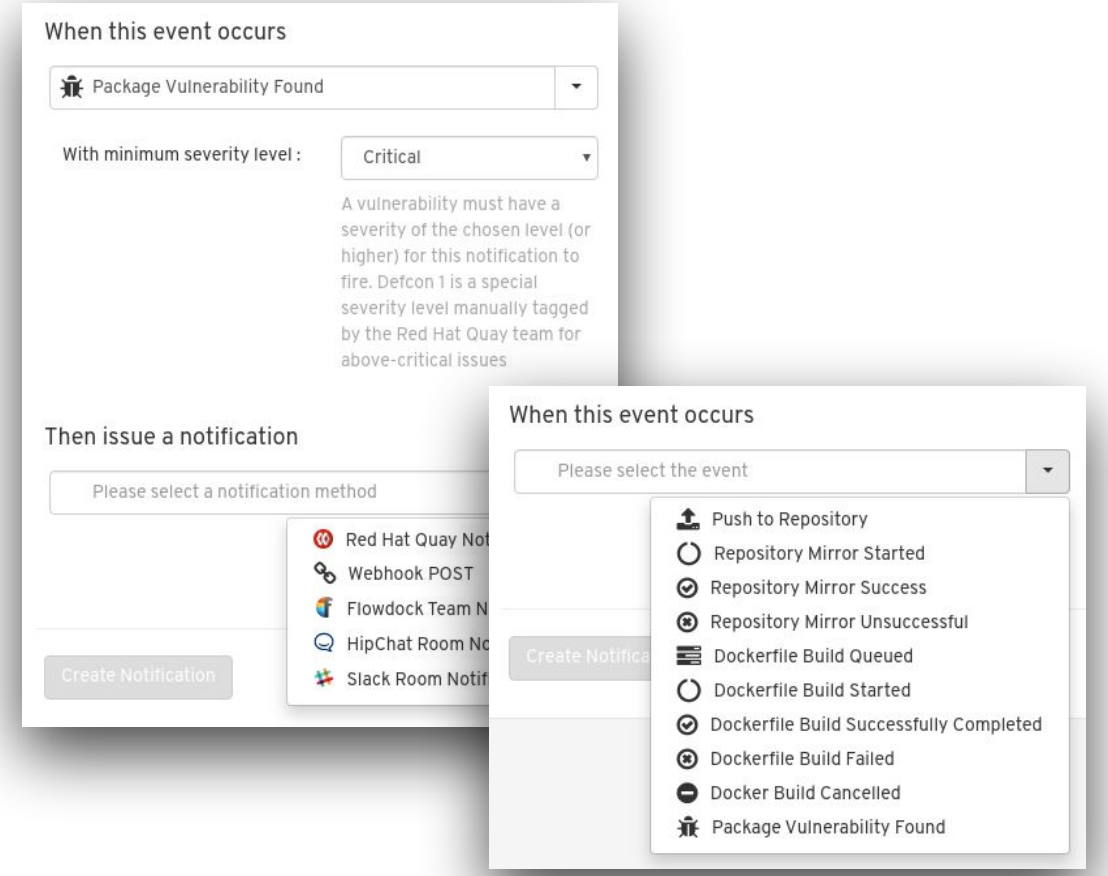
Clair v4 is the newest version of Clair after a massive refactoring in order to make several big enhancements possible. This includes:

- Support for programming language package managers (3.3: python)
- immutable data model & new manifest-oriented API
- Refocus on latest container specifications (OCI) (Content addressability)

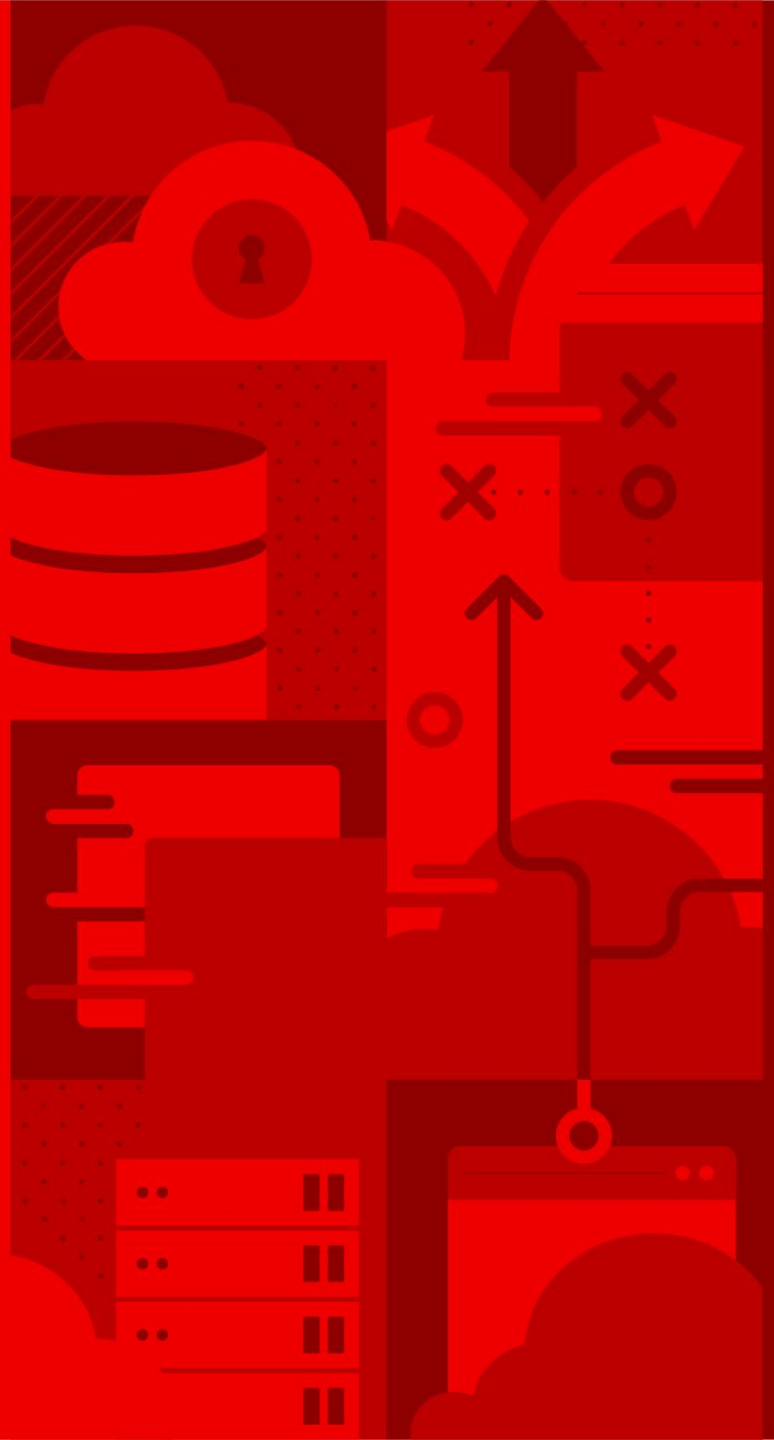
Red Hat

Notifications for Vulnerabilities found by Clair

- **Quay triggers different notifications for various repository events** (depends on enabled features)
- This includes the event type **“Package Vulnerability Found”**
- Additional Filter can be applied for **Severity Level**
- **Various Notification Methods**
- Custom Notification Title (optional)

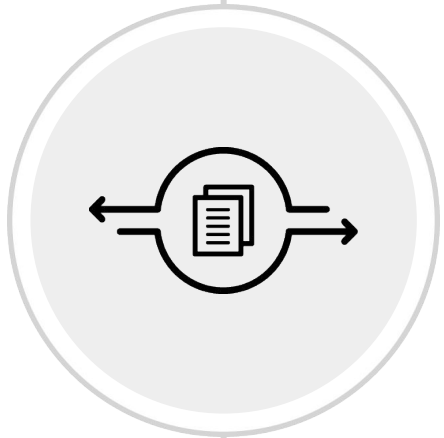


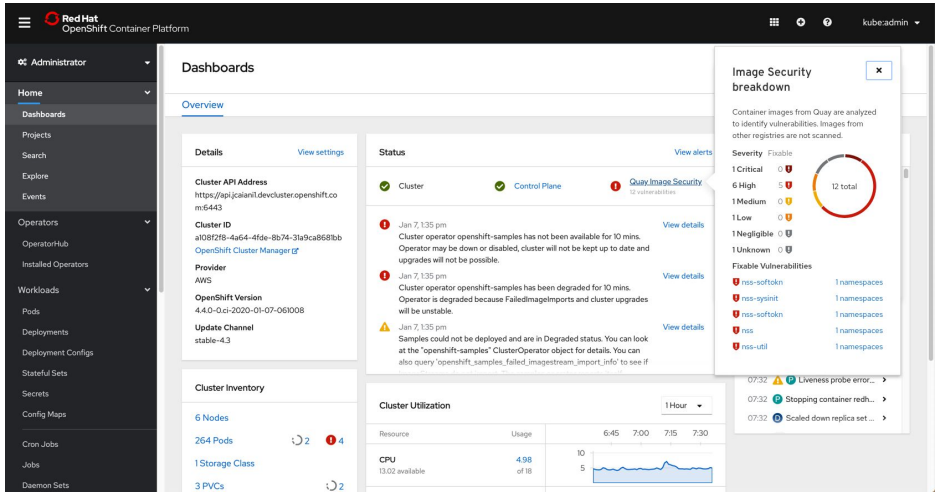
Red Hat



Container Security Operator and OpenShift Console Integration

Red Hat





RED HAT® QUAY

Scans images running on OpenShift and exposes data via API

clair

Red Hat OpenShift Container Platform

Operator monitors pod objects and updates vulnerability data

Container Security Operator - Vulnerability Data in OpenShift

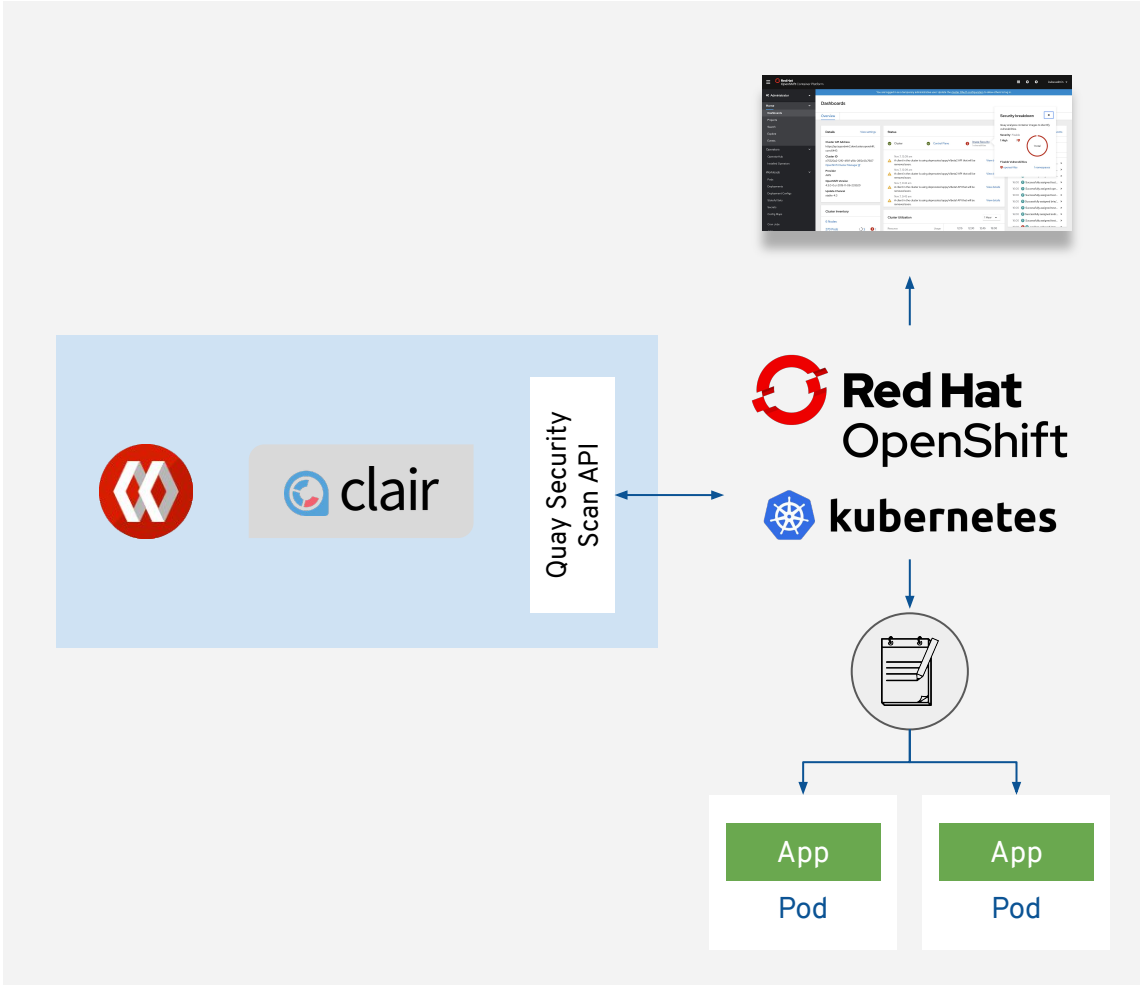
Operator which runs on OpenShift and fetches vulnerability from Quay / Clair if Kubernetes pod objects change

Synchronous Updates of vulnerability information

Prerequisite to leverage / show vulnerability data in OpenShift Console

Red Hat

Container Security Operator (CSO)

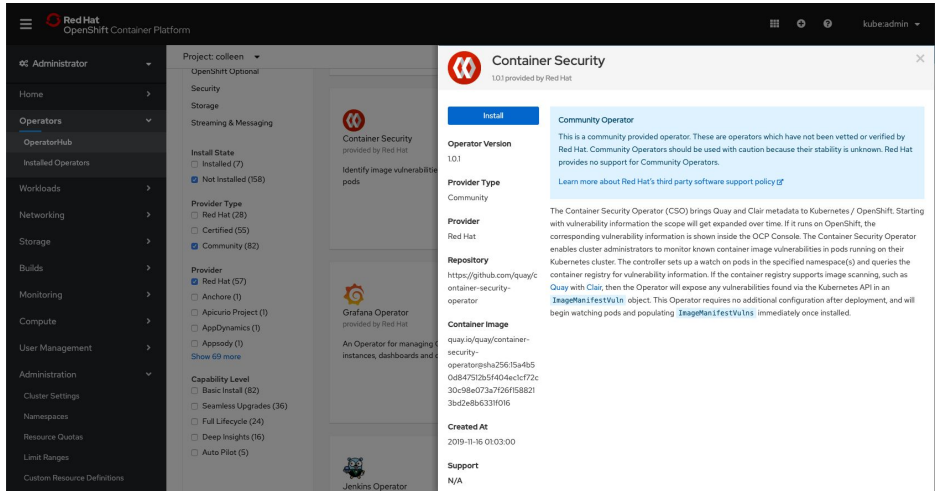


- Container Security Operator (CSO) runs on OpenShift and watches pod objects
- Pod object changes triggering a data fetch from Quay/Clair and stores vulnerability information in CRs (by image manifest ID)
- CRs gets deleted if pod gets deleted
- Configurable interval to update vulnerability data from Quay / Clair (default: 5min)
- Data available via k8s CLI / APIs
- Supposed to be used by partner security products as well (consistent data ingress)

Red Hat

How to deploy the Container Security Operator (CSO)

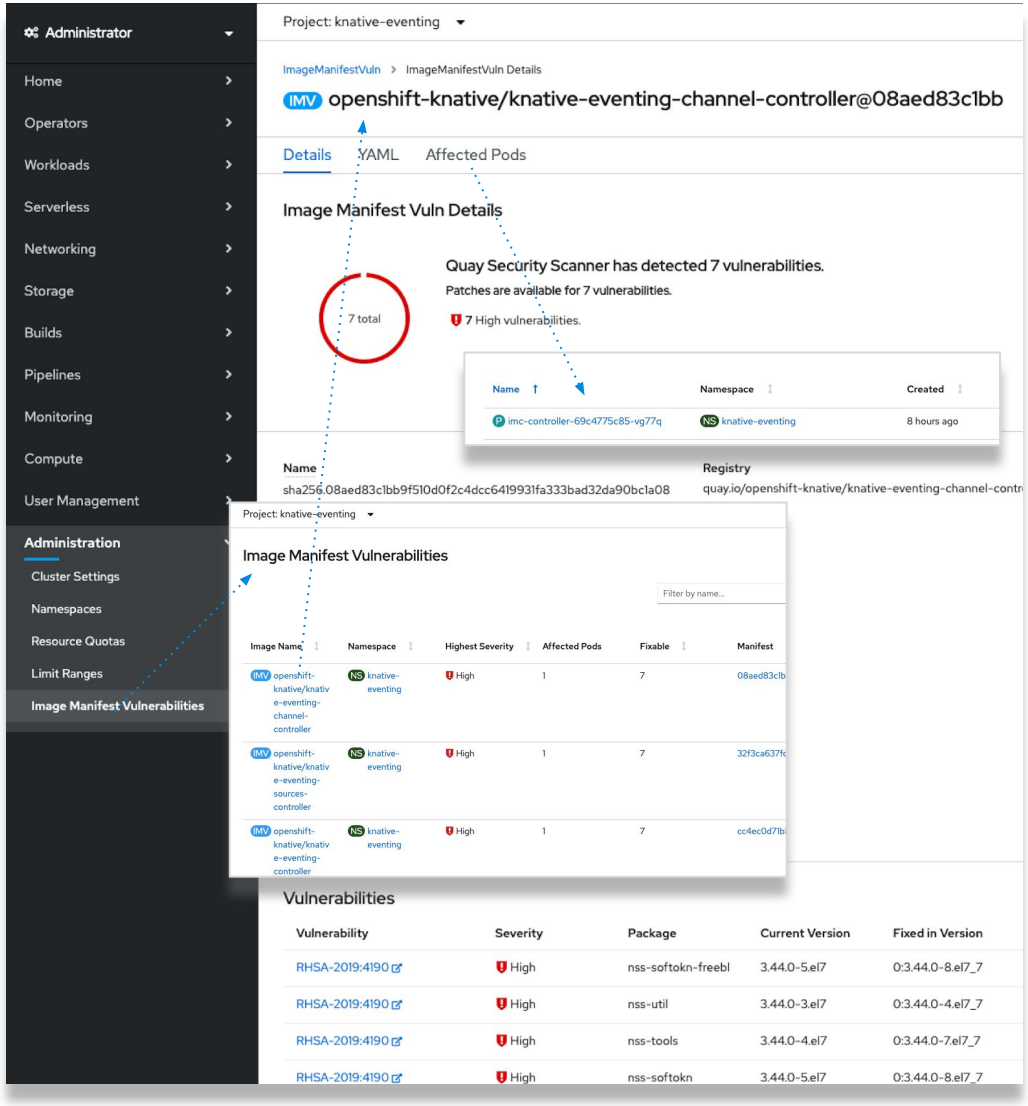
- CSO supposed to run on all OCP clusters Quay is serving images to (not limited to the cluster Quay is running **on**)
- CSO available in OpenShift embedded operatorhub (upstream version in operatorhub.io)
- Deployment via Operator Lifecycle Manager to ensure that OLM takes care of RBAC permissions, dependency resolution and automatic upgrades
- Works with both Red Hat Quay and Quay.io




```
securitylabeller:
  host: # Leave empty to use in-cluster config
  prometheusAddr: "0.0.0.0:8081"
  interval: 1m
  workers: 1
  labelPrefix: secscan # Security labels' "namespace"
  namespaces: # List of namespaces to label in the cluster
    - default
    - dev
  securityScanner:
    host: "https://quay.mycompany.com"
    apiVersion: 1
    type: "Quay"
```


Red Hat

OpenShift Console Integration via CSO



List view - easily view vulnerabilities of the images

- Highest severity
- Number of affected pods
- Number fixable
- Manifest SHA external link for viewing the vulnerability in Quay

Details view - see a list of vulnerabilities of an image

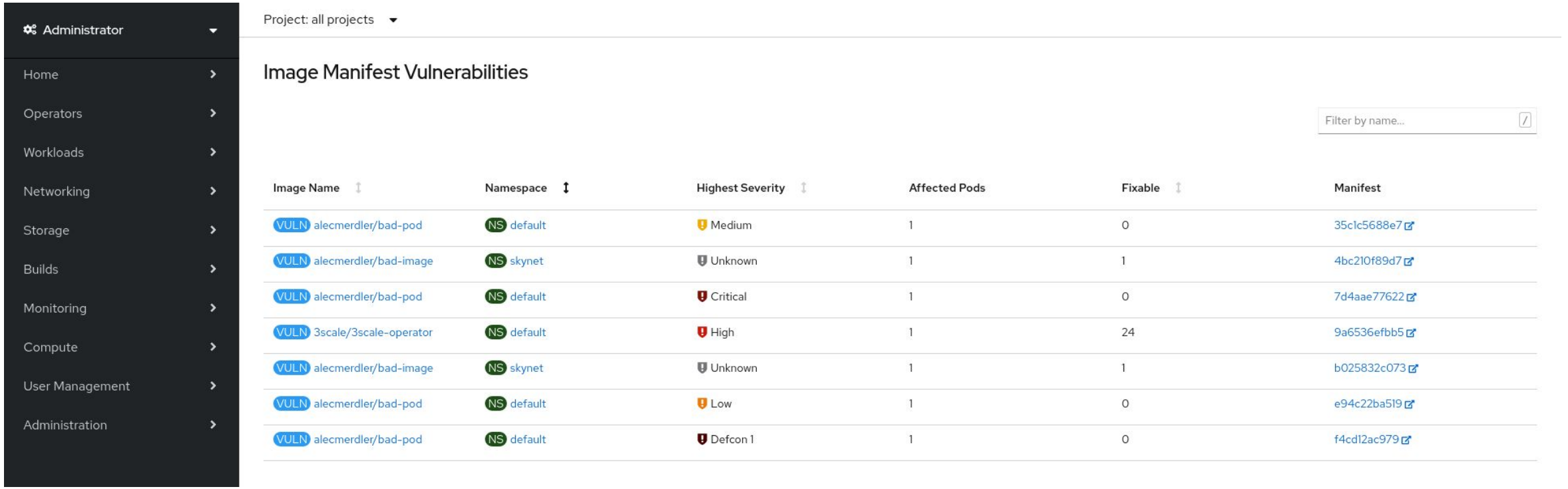
- Vulnerability info, severity, package, current package, and the fixed version.

Affected Pods tab - easy access to the affected pods to quickly update with the fixes.

Learn more about the new views we added to the OpenShift Console with OCP 4.4:
https://blog.openshift.com/openshift-4-4-not-on-my-watch-image-vulnerabilities-list/

Red Hat

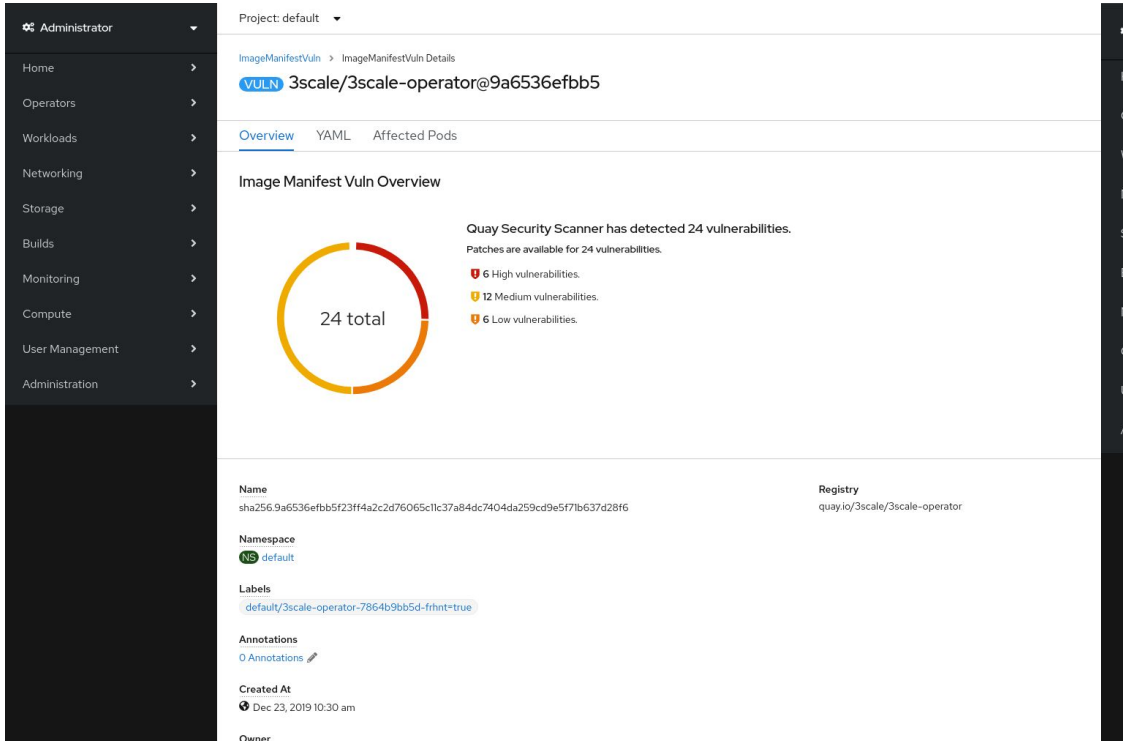
Vulnerability Data inside the OpenShift Console

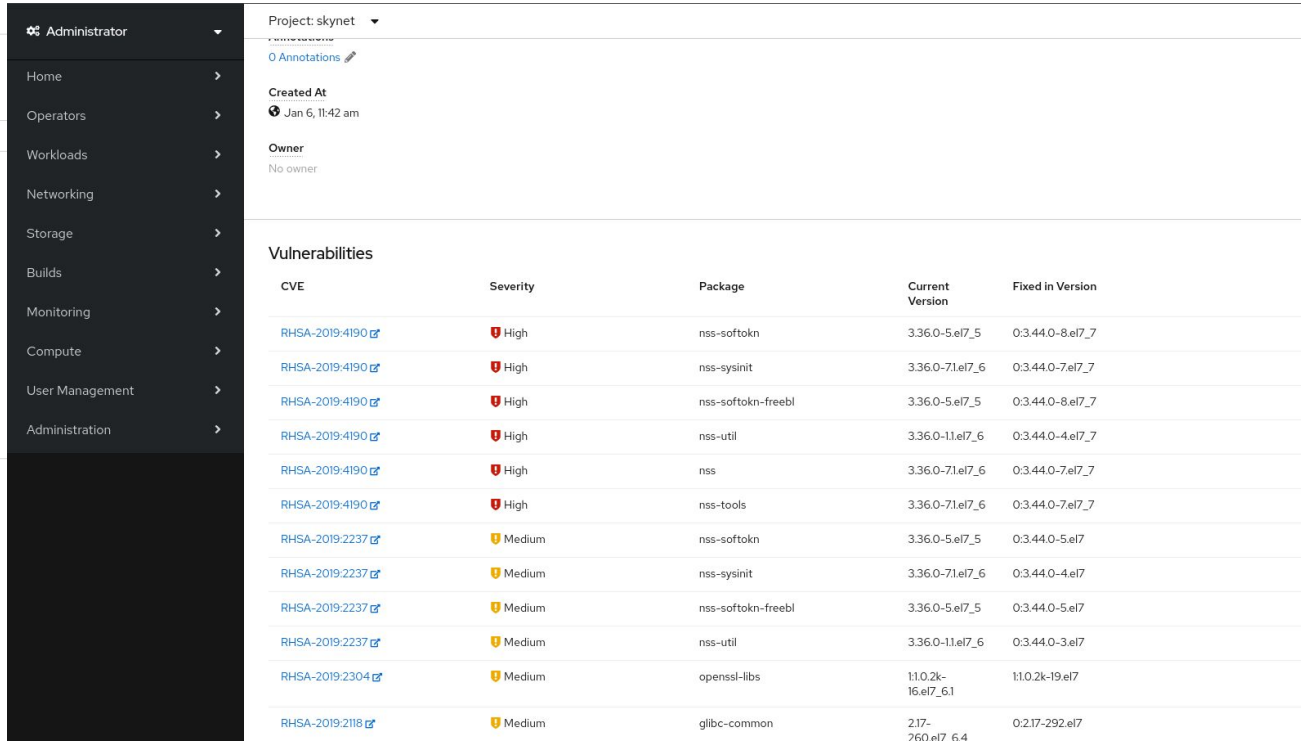


`ImageManifestVuln` list view

Red Hat

Vulnerability Data inside the OpenShift Console

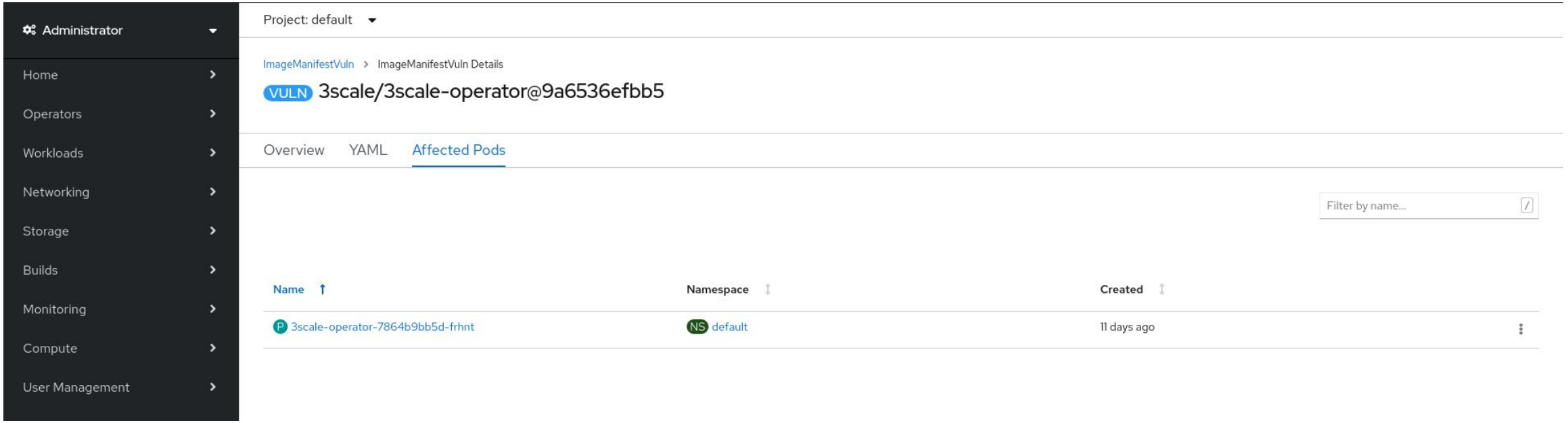




ImageManifestVuln detail view

Red Hat

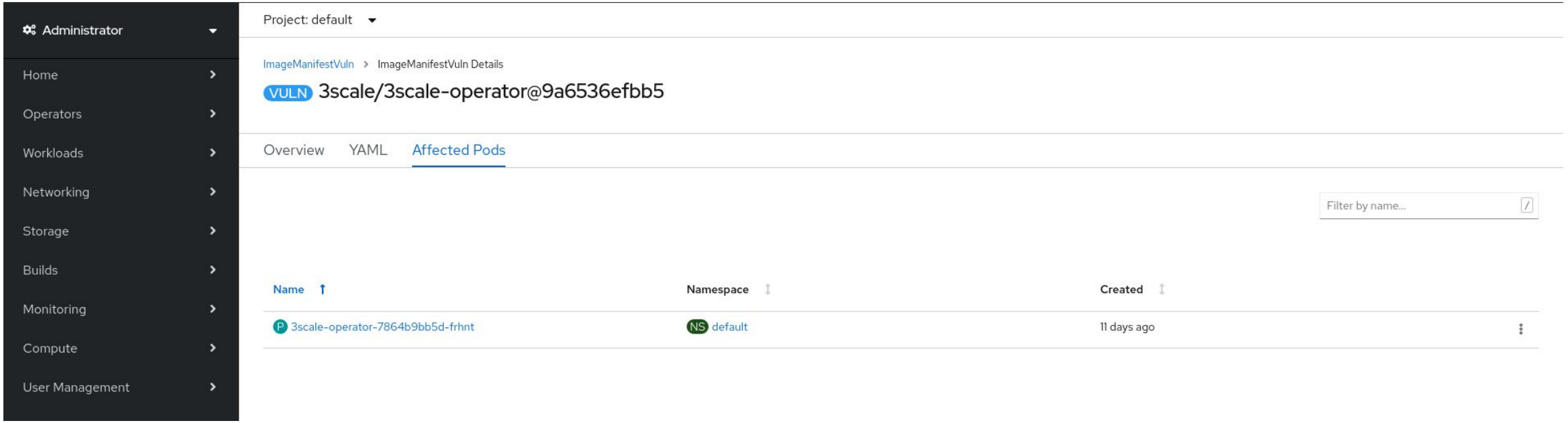
Vulnerability Data inside the OpenShift Console



ImageManifestVuln detail view (affected pods)

Red Hat

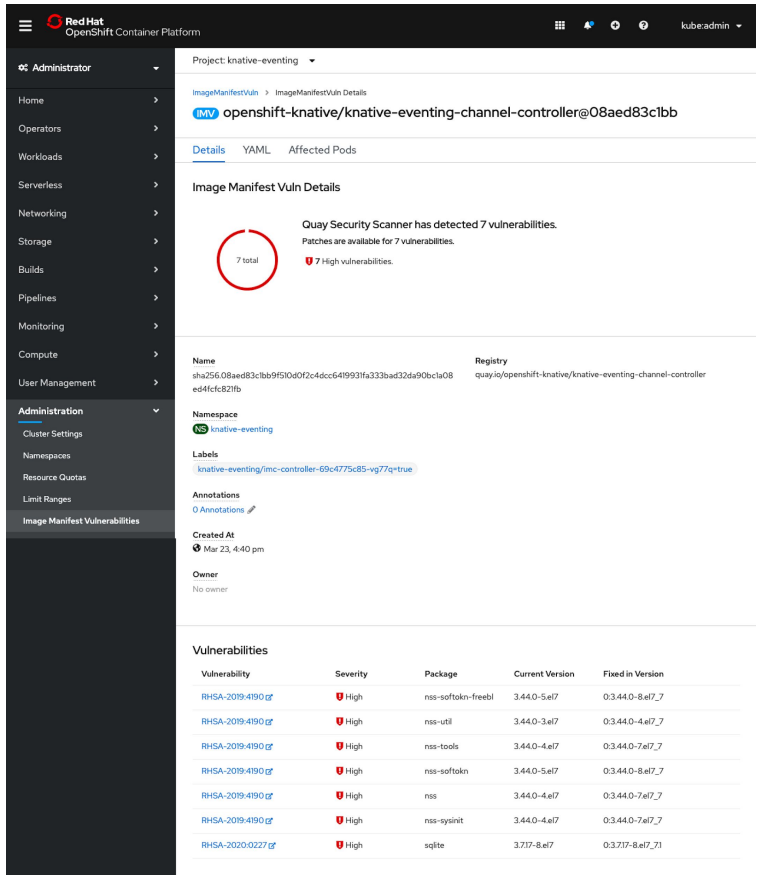
Vulnerability Data inside the OpenShift Console

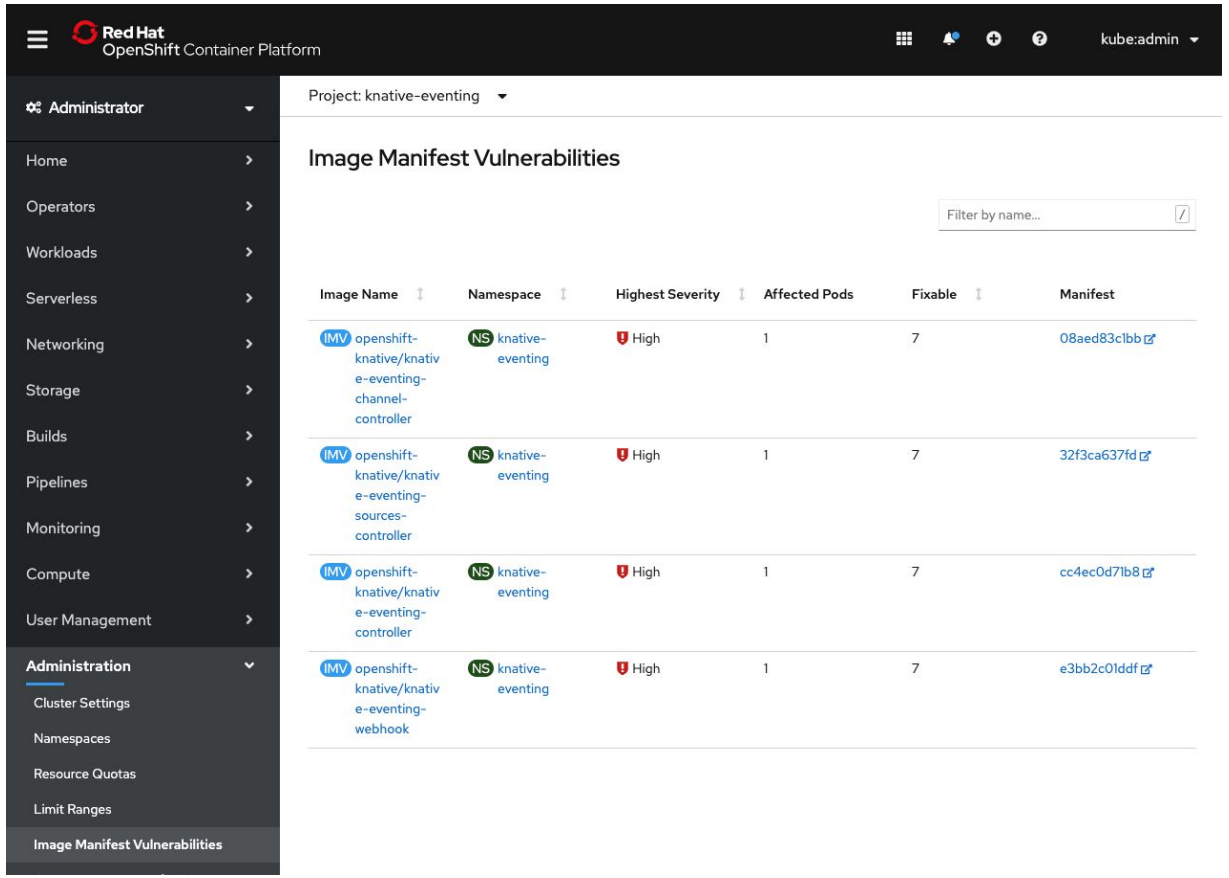


Kebab action on Pods list view

Red Hat

OpenShift Console Vulnerability Information Enhancements





Learn more about the new views which have been added to the OpenShift Console as part of OCP 4.4:
https://www.openshift.com/blog/4.4-openshift-not-on-my-watch-image-vulnerabilities-list

Red Hat

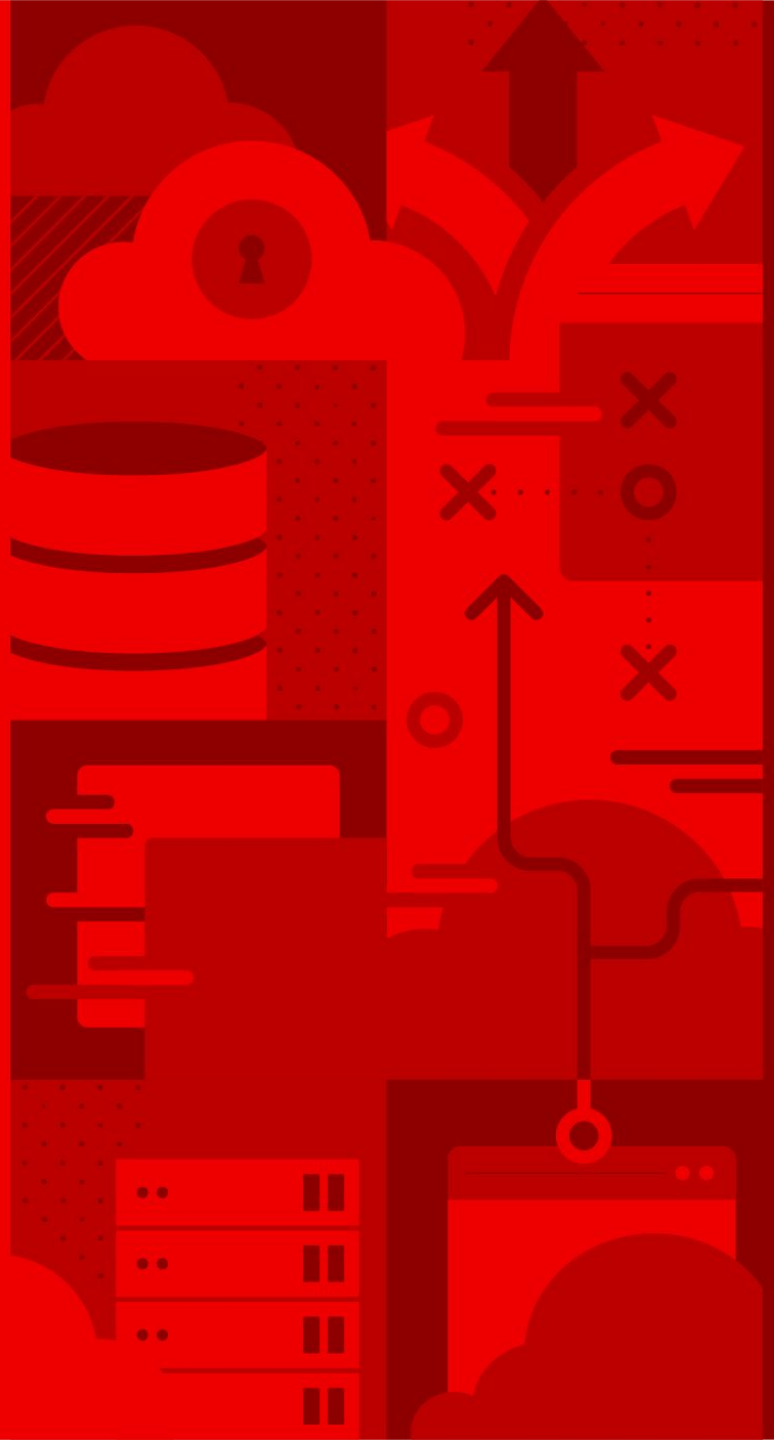
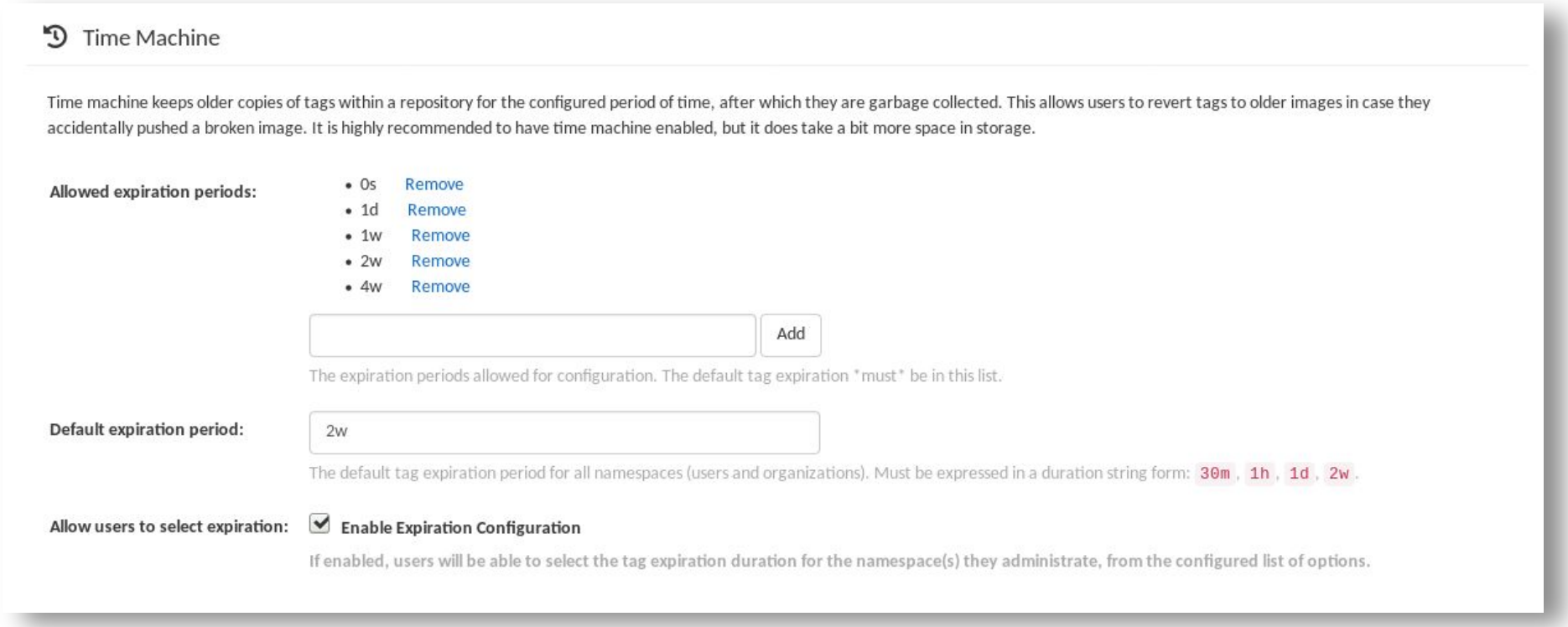


Image Rollback / Time Machine

Red Hat

Quay Time Machine

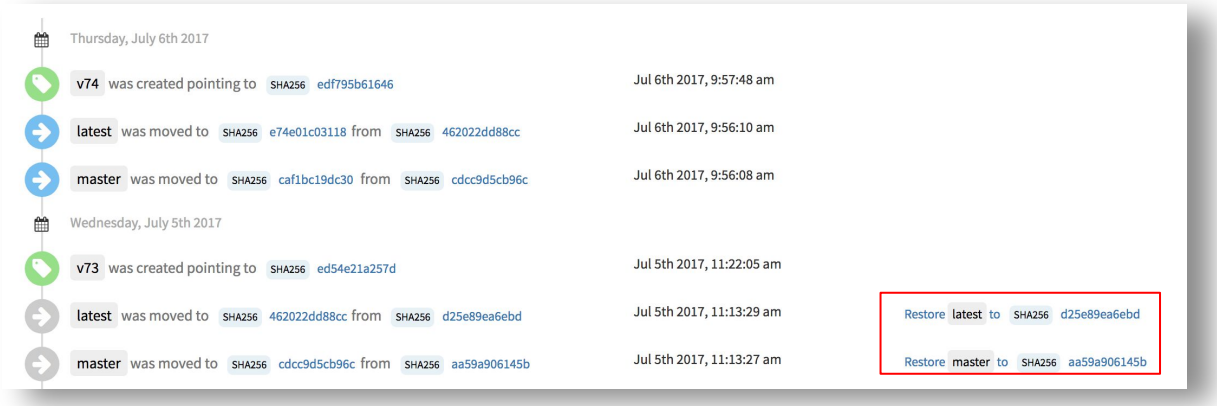


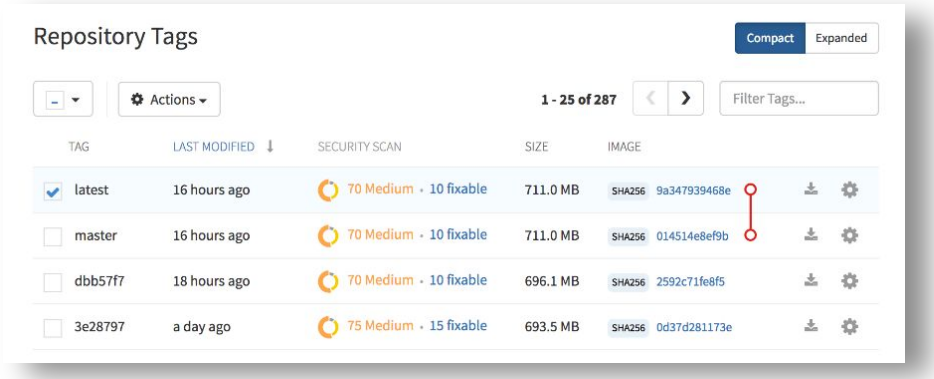
Time Machine registry wide settings are defined via Quay config app

They can be overridden by an **organization default value**.

Red Hat

Image Rollback / Time Machine





Time Machine provides image rollback

Description: view history of tags for up to two weeks and have the ability to revert tags to a previous state

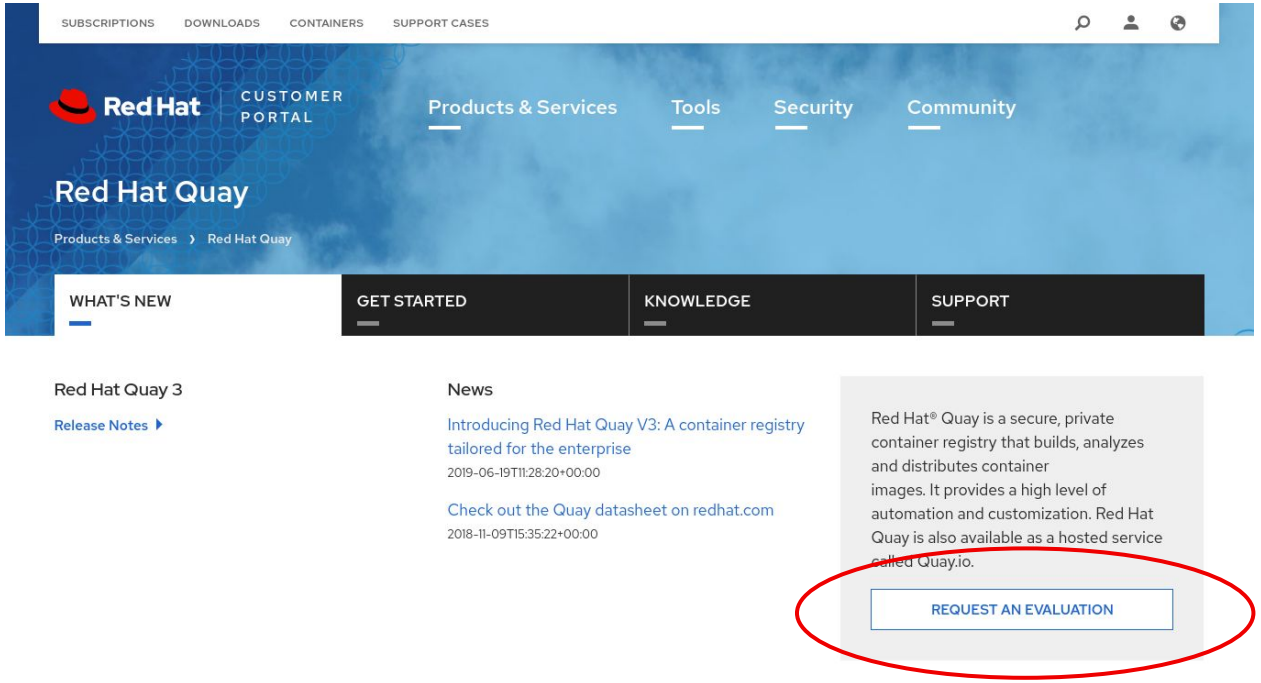
How it Works:

- Select Repository → Click on Tags tab
- Add a new tag / move a tag / delete tag
- View tag history
- Click on Restore to roll back

Red Hat

Try it out!

https://access.redhat.com/products/red-hat-quay



On all Quay product pages you can find an evaluation form which grants you access to the software for a 90 day trial period.

Alternatively you can signup **for free** on Quay.io

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat