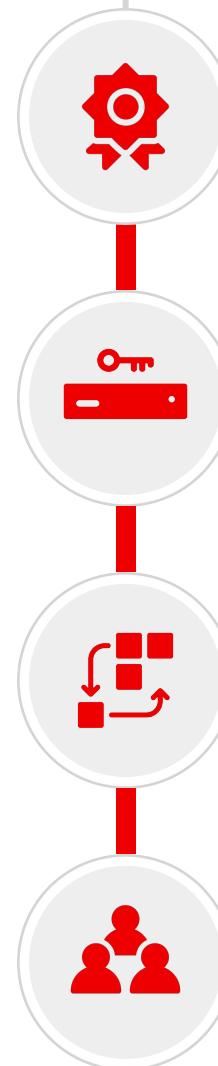




RED HAT QUAY TECHNICAL DECK

Detailed Feature Descriptions and Operational Guidance



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

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

SCALABILITY

Seamless Git Integration
Build Workers
Webhooks

BUILD AUTOMATION

Extensible API
Webhooks, OAuth
Robot Accounts

INTEGRATION

Vulnerability Scanning
Logging & Auditing
Notifications & Alerting

SECURITY

REGISTRY

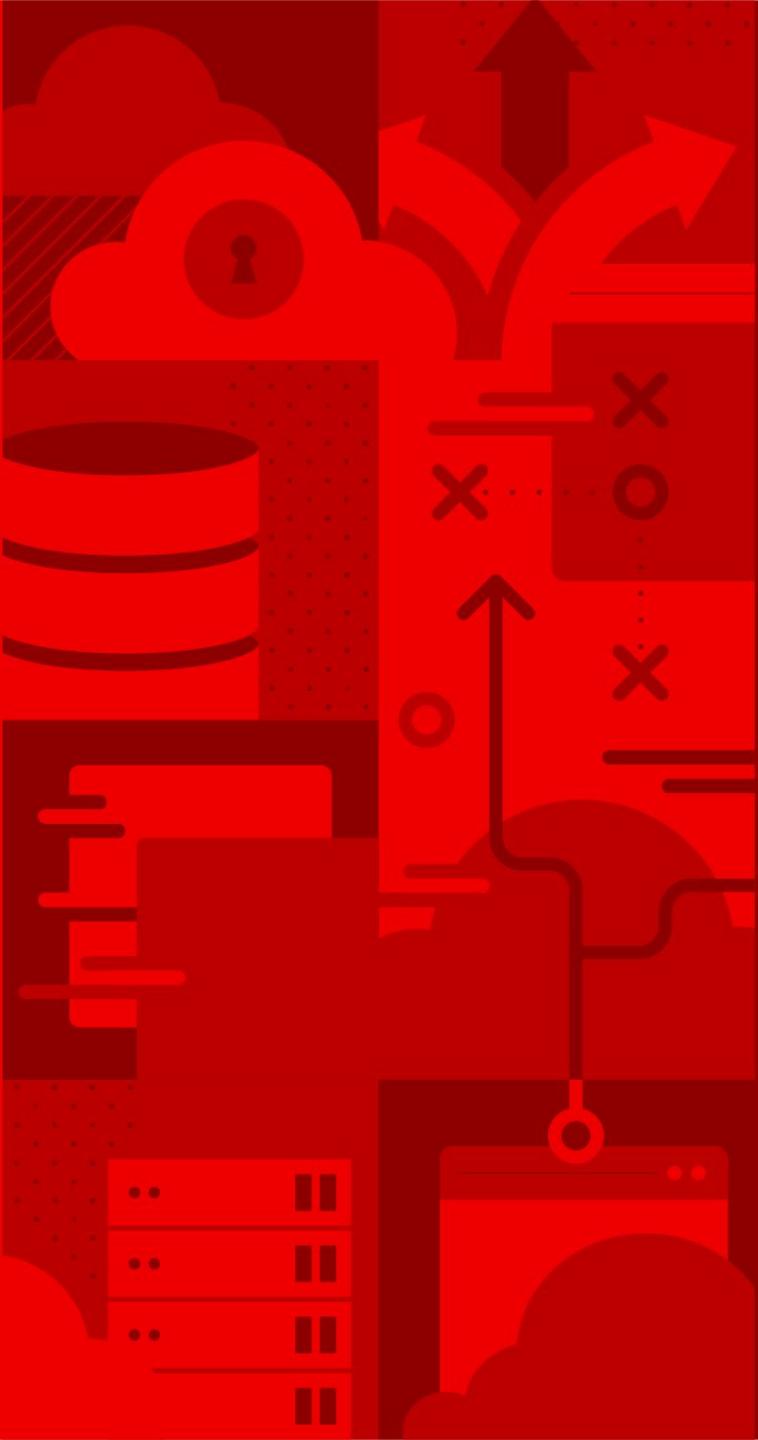
High Availability
Full Standards / Spec Support
Long-Term Protocol Support
Application Registry
Enterprise Grade Support
Regular Updates

CONTENT DISTRIBUTION

Geo-Replication
Repository Mirroring
Air-Gapped Environments

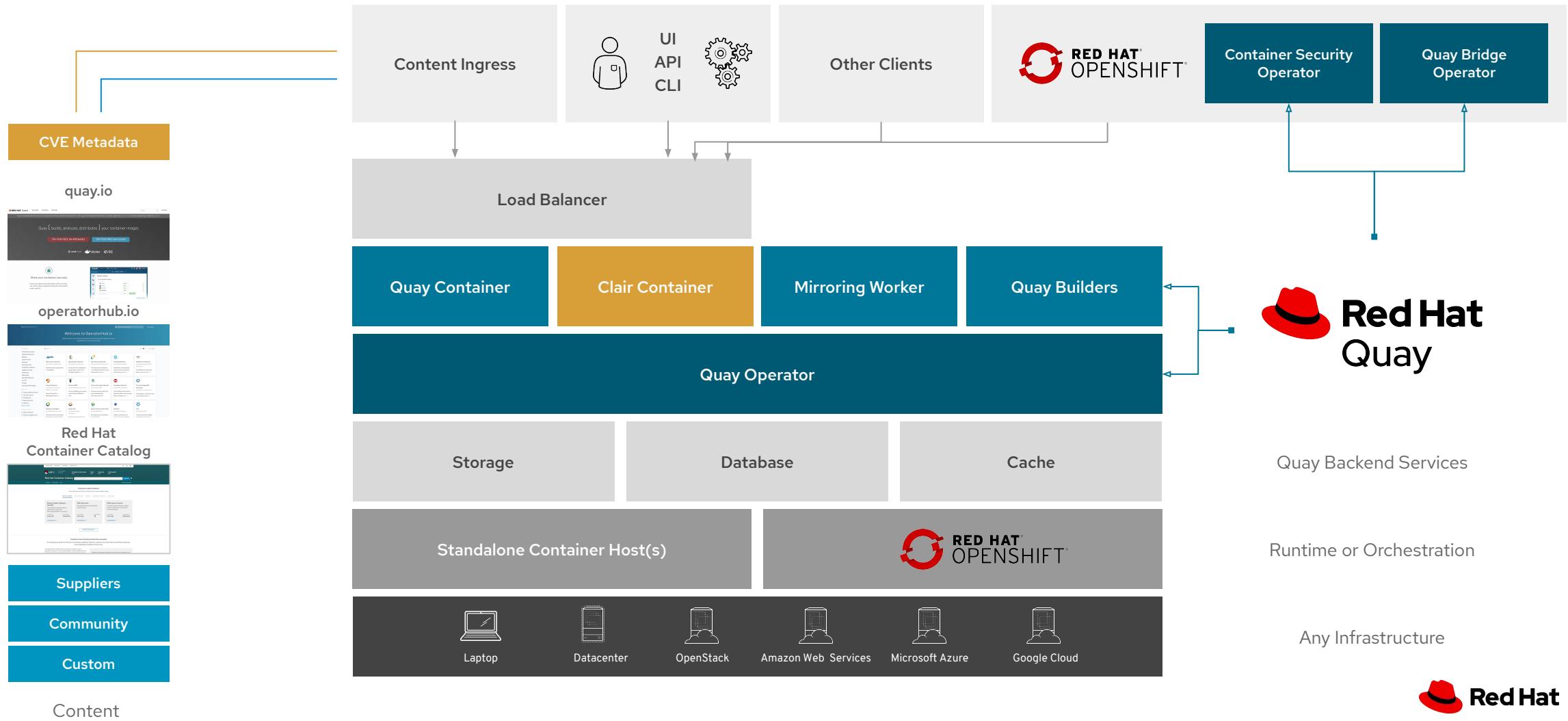
ACCESS CONTROL

Authentication Providers
Fine-Grained RBAC
Organizations & Teams

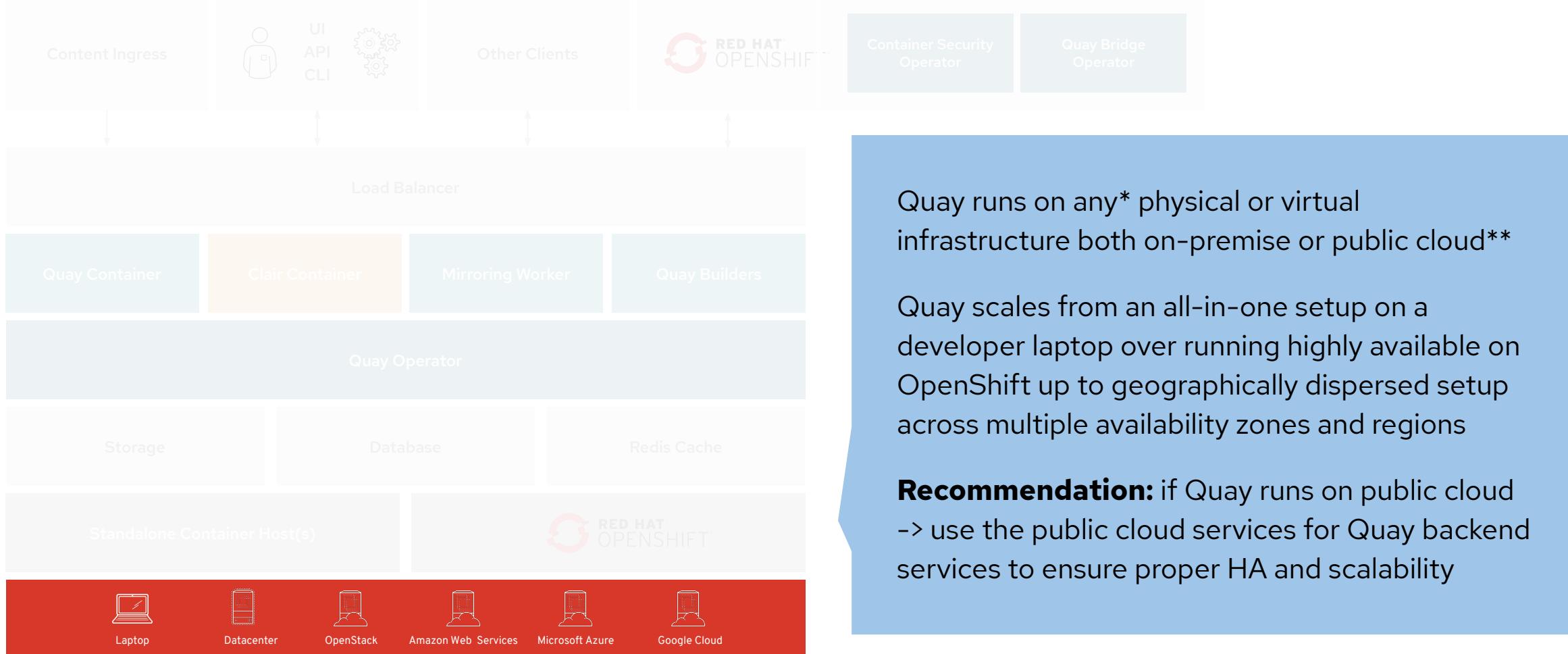


Quay Architecture

Red Hat Quay Architecture



Prerequisite: Infrastructure

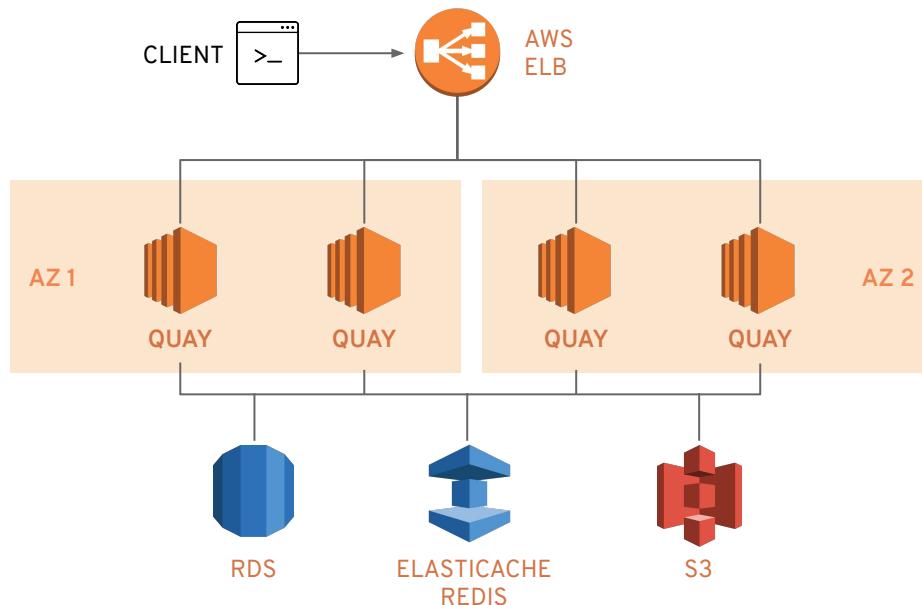


* Further details can be found in the Quay 3.x tested configuration matrix: <https://access.redhat.com/articles/4067991>

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

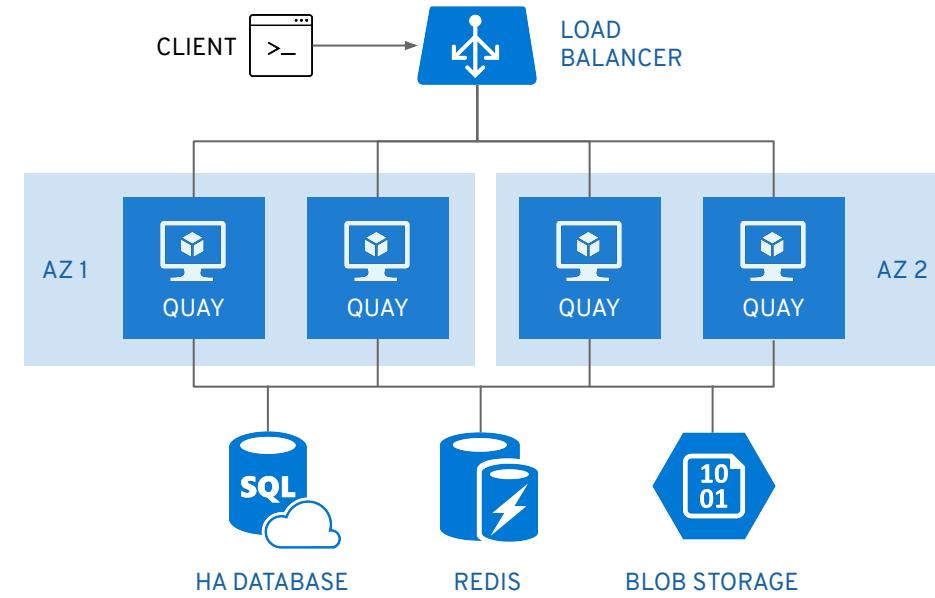
Running Red Hat Quay on Public Cloud

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:

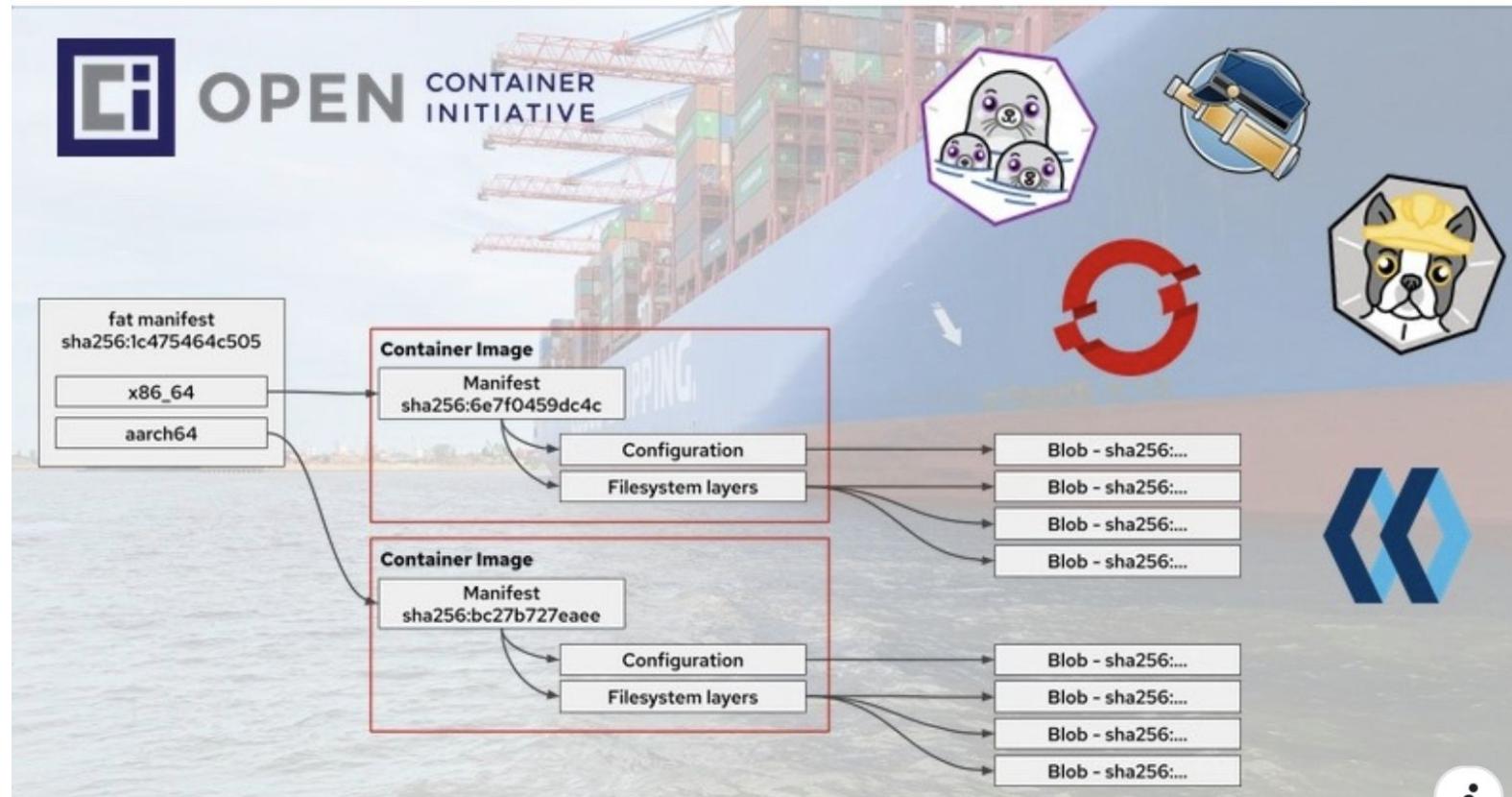
- AWS Elastic Load Balancer
- AWS S3 (hot) blob storage
- AWS RDS database
- AWS ElastiCache Redis
- EC2 VMs recommendation: M3.Large or M4.XLarge



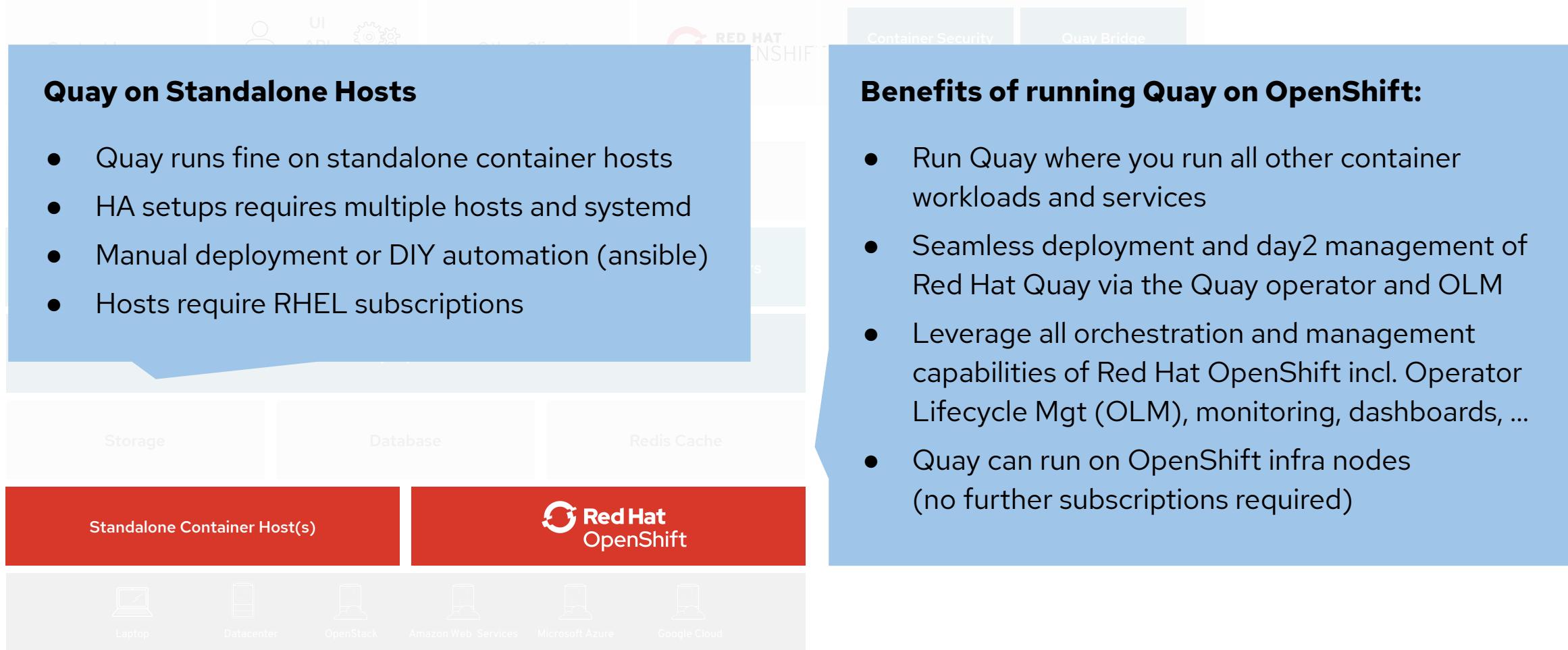
If Quay runs on MS Azure you can use:

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

Multi Architecture Containers



Prerequisite: Container Runtime or Orchestration

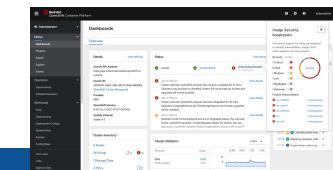


Red Hat Quay works best with OpenShift

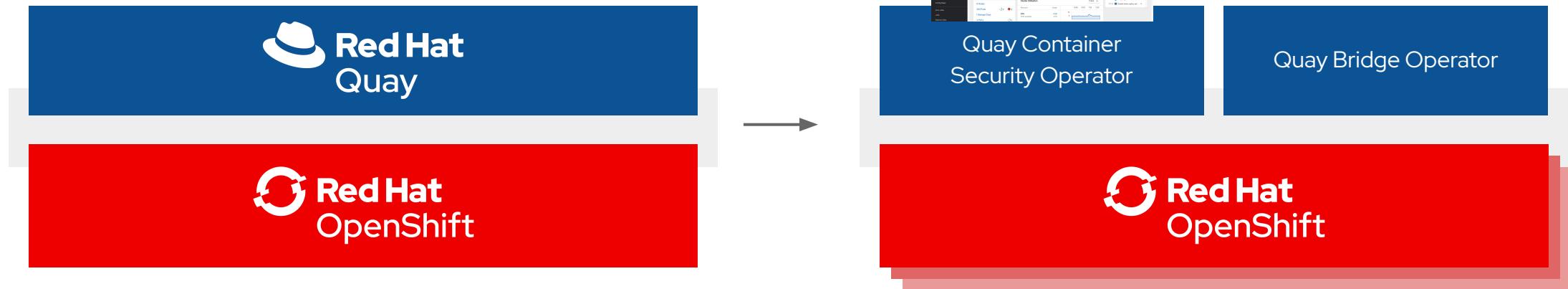
Red Hat Quay runs on any infrastructure
but **runs best on OpenShift**

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.

Benefits of running Quay on OpenShift



- **Zero to Hero** - Simplified deployment of Quay and associated components means that you can start 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

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

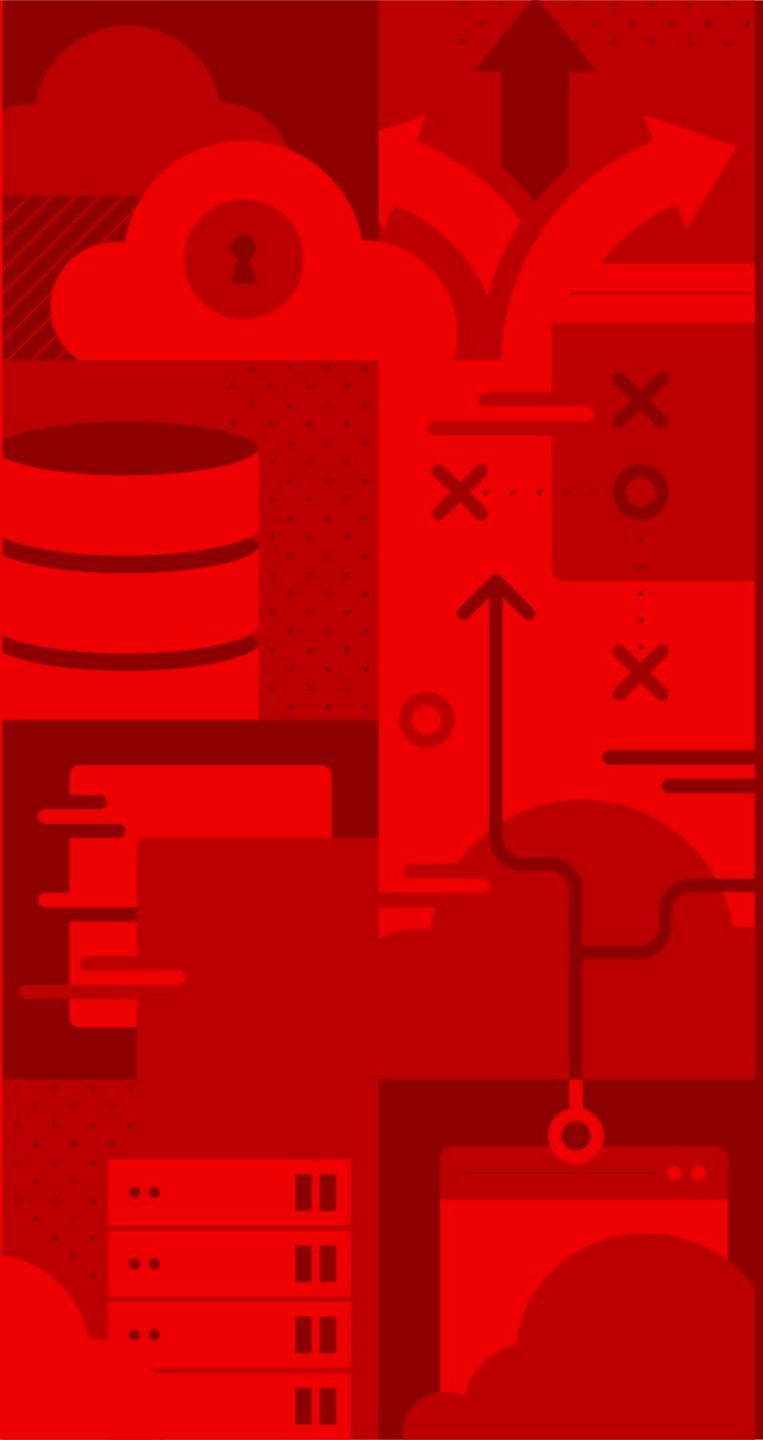
Quay Sample Sizings

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



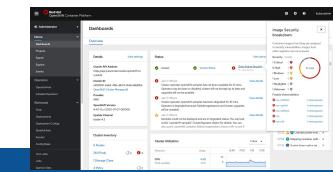
Quay and OpenShift

Red Hat Quay works best with OpenShift

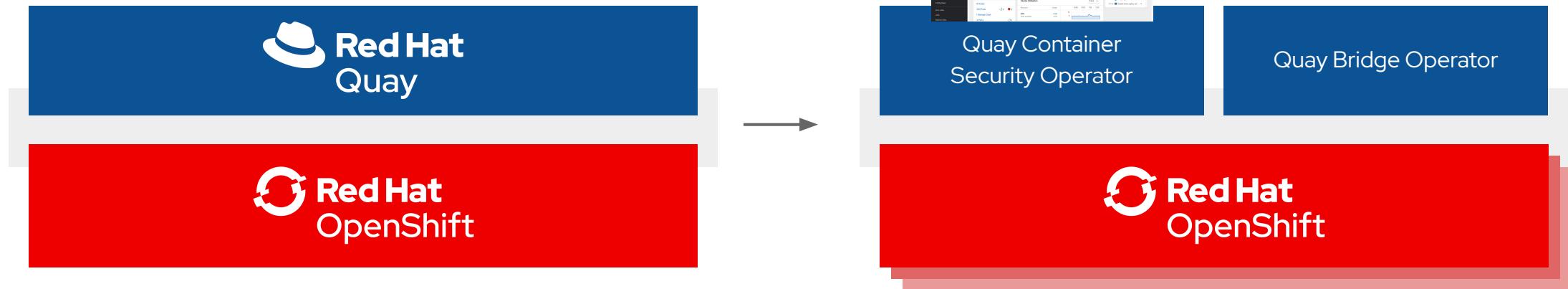
Red Hat Quay runs on any infrastructure
but **runs best on OpenShift**

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.

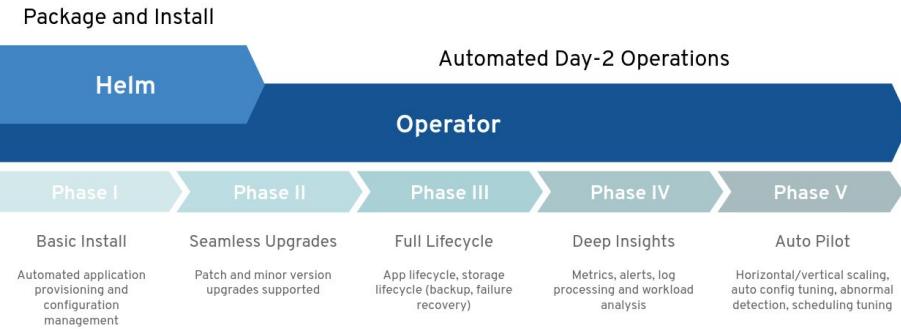
With or without using the OpenShift internal registry but leveraging all OpenShift capabilities.

Benefits of running Quay on OpenShift



- **Zero to Hero** - Simplified deployment of Quay and associated components means that you can start 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

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

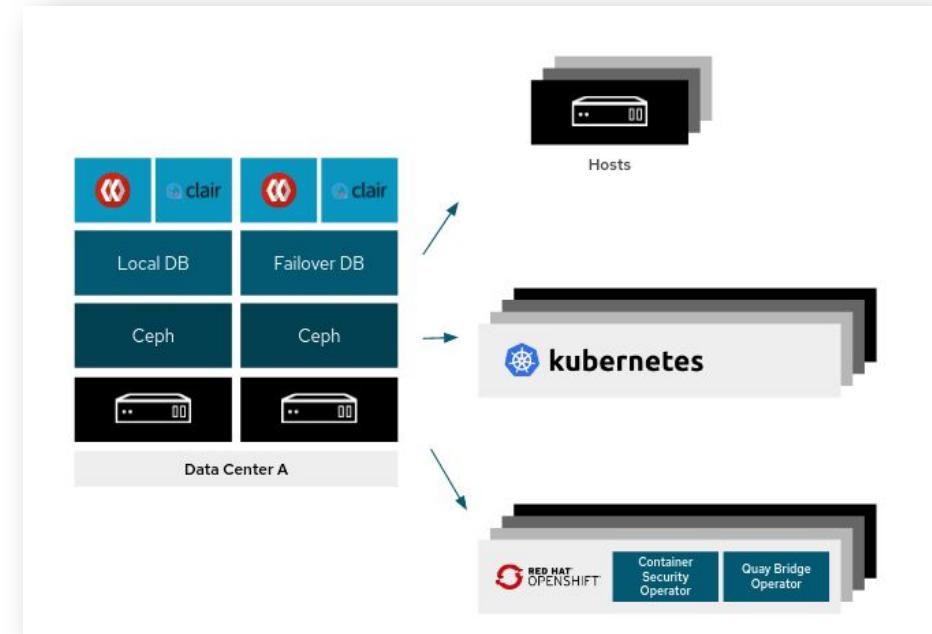
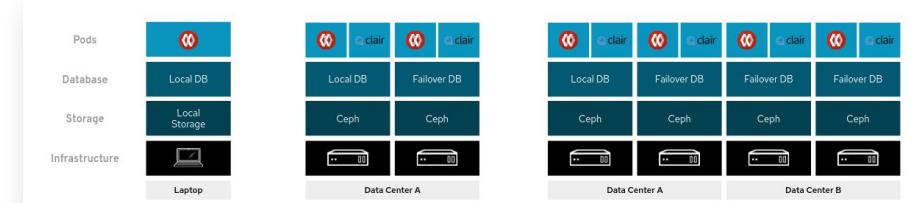
Maintaining another deployment and management tooling for non-k8s deployments is not feasible and not aligned to our prioritization and roadmap (Quay v4 will run on k8s by default)

The screenshot shows the Red Hat OpenShift Container Platform interface. The left sidebar is titled 'Administrator' and includes links for Home, Overview, Projects, Search, Explore, Events, Operators (which is currently selected), OperatorHub, Installed Operators, Workloads, Networking, Storage, Streaming & Messaging, Install State, and Builds. The main content area is titled 'Project: all projects' and shows a search bar with 'quay'. Below the search bar, there are three cards in the 'Community' section:

- Quay** provided by Red Hat: Red Hat Quay is a private container registry that stores, builds, and deploys container...
- Quay Bridge Operator** provided by Red Hat: Enhance OCP using Red Hat Quay container registry
- Red Hat Quay** provided by Red Hat: Red Hat Quay is a private container registry that stores, builds, and deploys container...

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



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

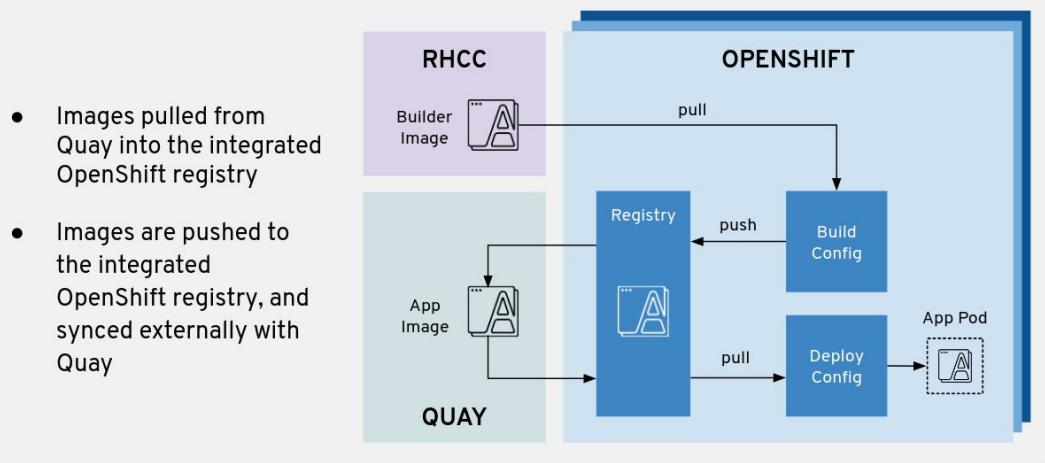
OpenShift and External Registries

- 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

Using Quay With or Without Internal Registry

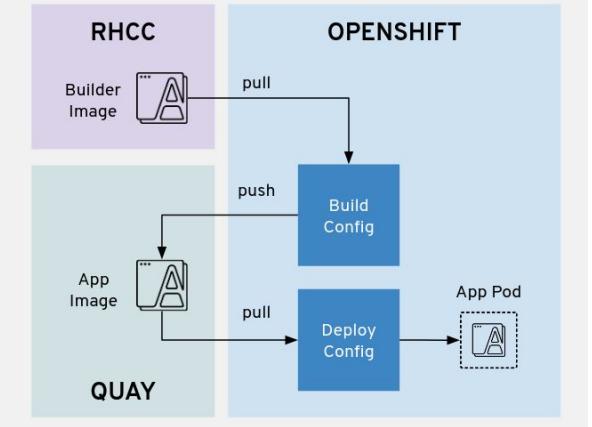
- Quay can be used as an external registry in front of an entire OpenShift cluster with its registry
- Quay also can be used directly without using the internal registry which requires a couple of changes (secrets, build and deployment configs) which are **partially** done automatically by QBO

Quay as Upstream Registry with OpenShift



Quay as OpenShift Registry

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



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

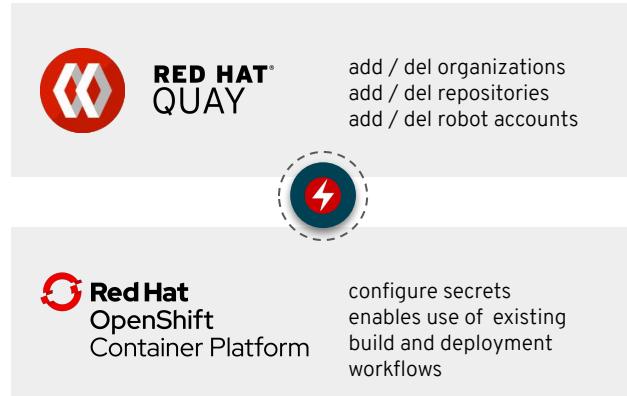
```
"output": {  
    "to": {  
        "kind": "DockerImage",  
        "name":  
        "external-registry.example.com:5000/publicrepo/sample-app:latest"  
    },  
    "pushSecret": {  
        "name": "external-registry"  
    },  
}
```

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

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)



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

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

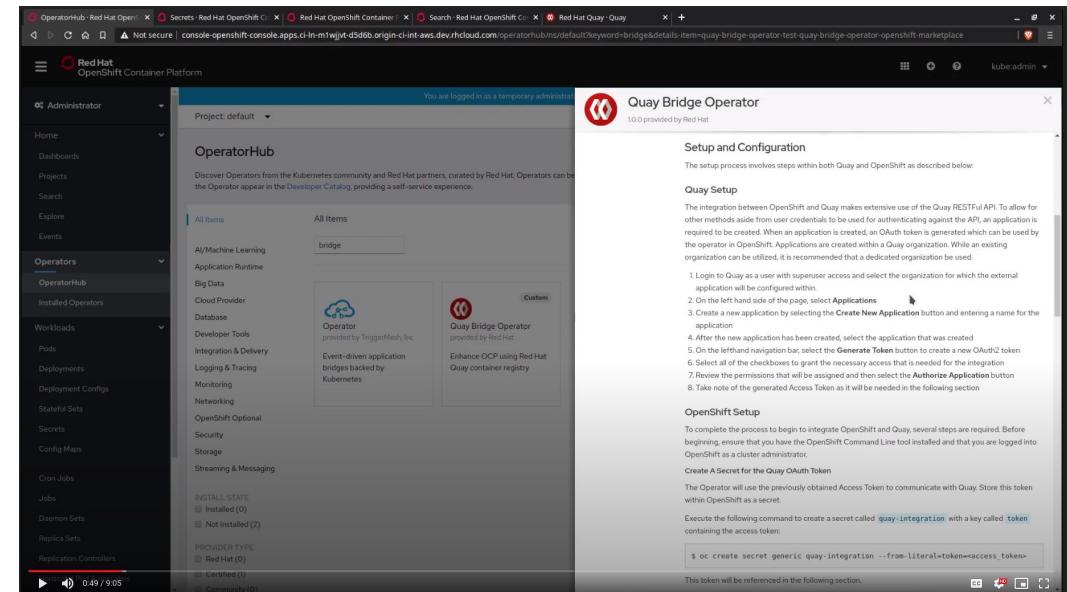
Quay and OpenShift - Bridge Operator

- 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

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)



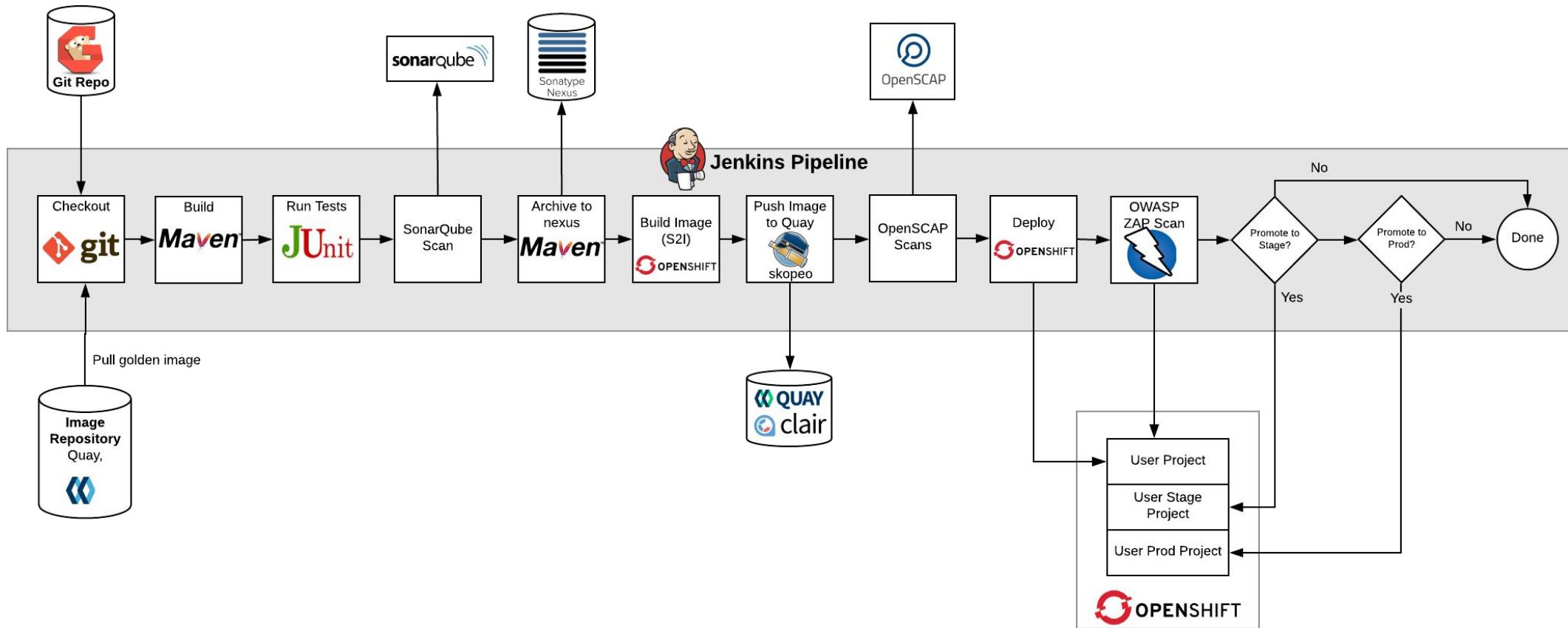
Check out the Quay Bridge Operator Demo:
https://drive.google.com/open?id=1EcDGK9gRwDTFH0etdHyq_zl-nixgyZFX

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

Quay and OpenShift - Pipeline Integration

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



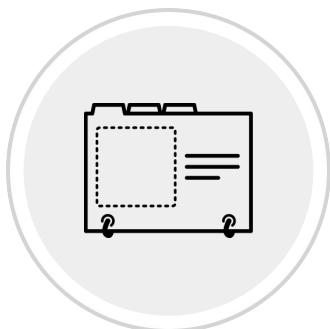
Quay triggering Redeployments on OpenShift

- 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

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 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

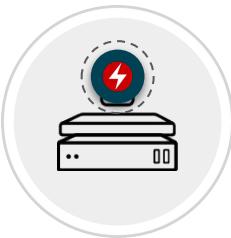


Builder

Triggered by git actions and
builds container images
Requires docker runtime
Requires privileges

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

Red Hat Quay Operators



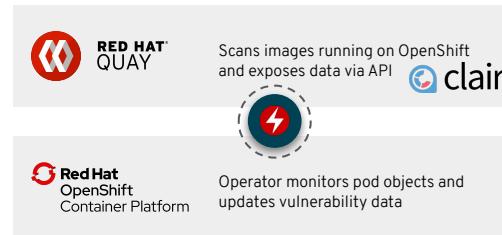
Quay Operator

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



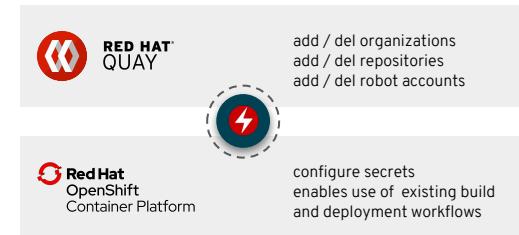
Container Security Operator

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

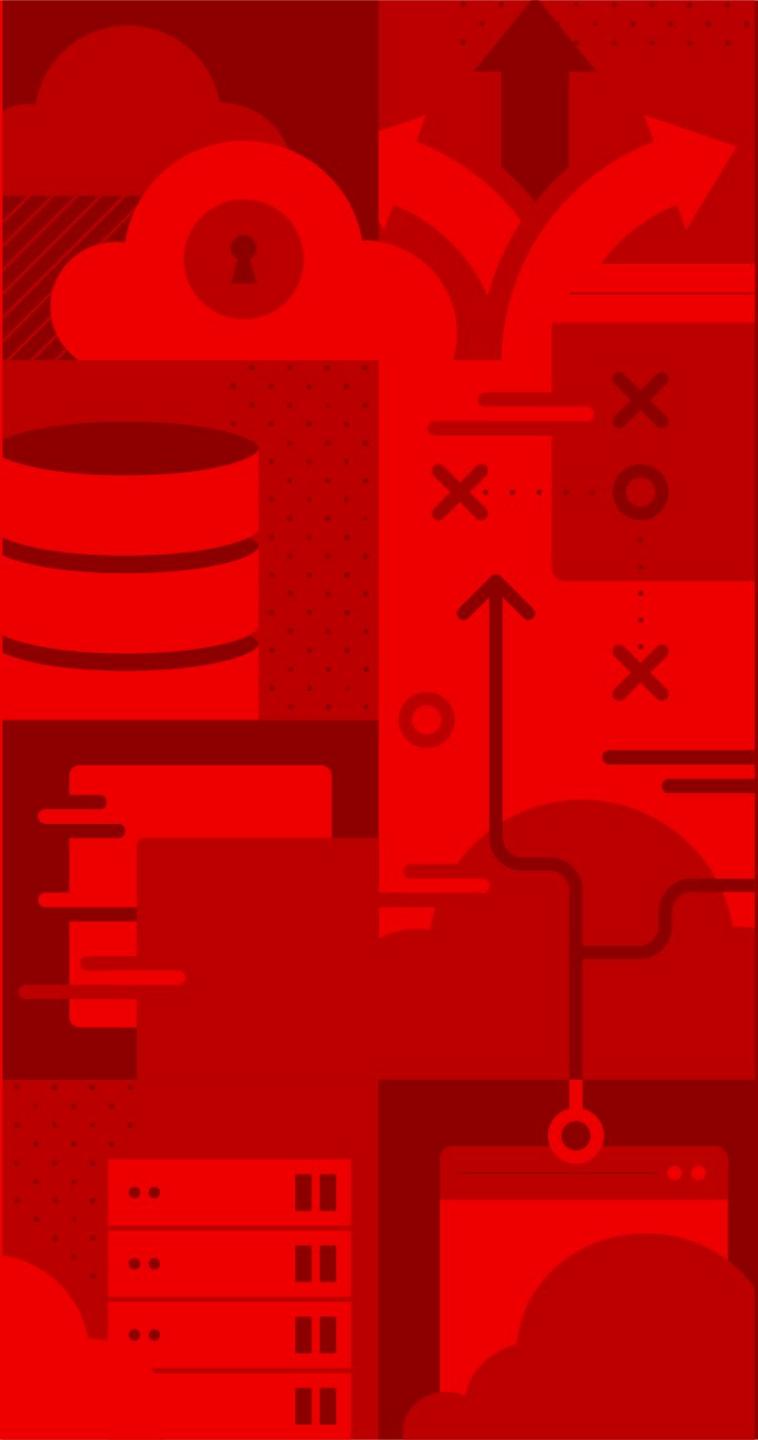
How to Pull Red Hat Quay Software

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+



Quay Registry Storage

Red Hat Quay - Registry Storage Configuration

Registry Storage

Registry images can be stored either locally or in a remote storage system.

Do not use local storage for any production configurations.

Proxy storage via Project Quay
If enabled, all requests to storage engine(s) will be proxied through Project Quay . Should only be enabled if storage cannot be directly accessed by external nodes talking to the registry.

Enable Storage Replication
If enabled, replicates storage to other regions. See [documentation](#) for more information.

Location ID: default

Storage Engine: Locally mounted directory

Storage Directory: /datastorage/registry

“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.

Red Hat OpenShift Container Storage (NooBaa S3)

Locally mounted directory

Amazon S3

Azure Blob Storage

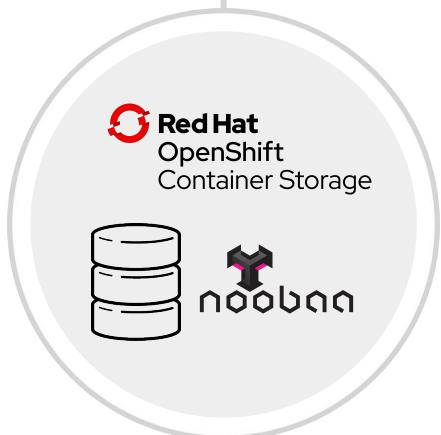
Google Cloud Storage

Ceph Object Gateway (RADOS)

OpenStack Storage (Swift)

CloudFront + Amazon S3

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

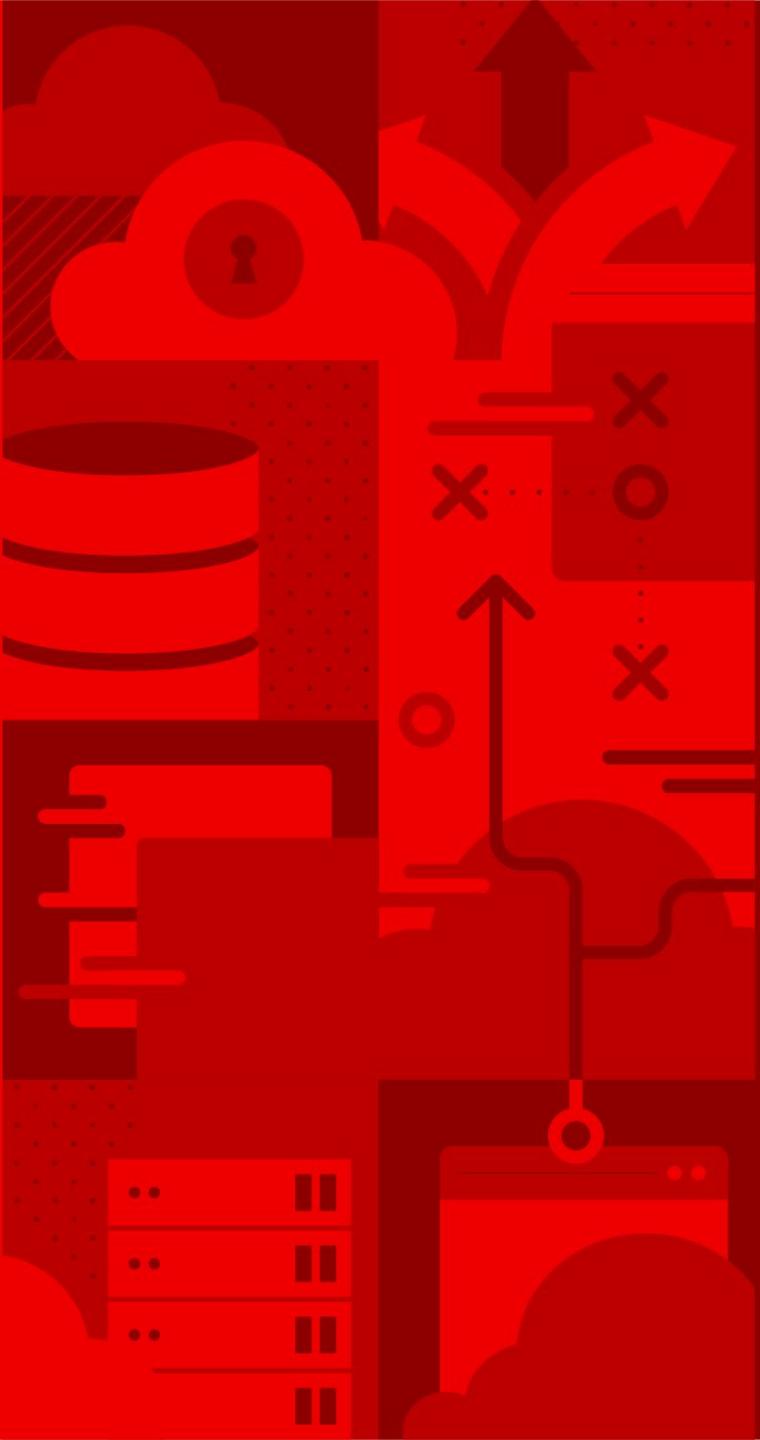


Red Hat OpenShift Container Storage 4

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.

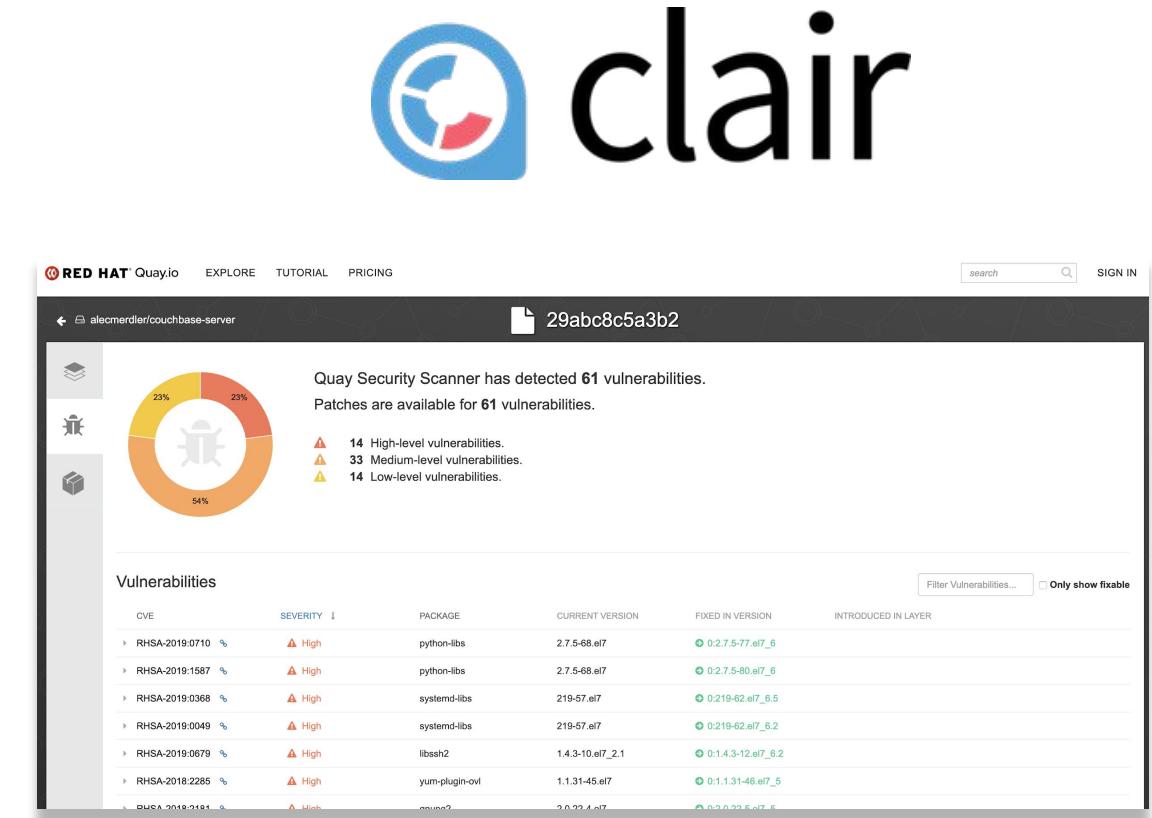
Note: this is a GA version of the Multi-Cloud Object Gateway.



Built-In Vulnerability Scanning via Clair

Clair Overview

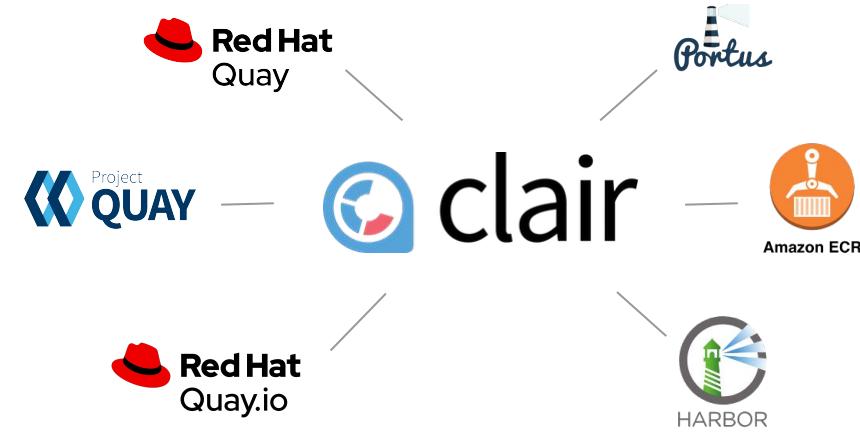
- Clair is an open source tool for static analysis of vulnerabilities in application containers
- Developed by CoreOS for Quay and it's massive scale usage at Quay.io
- Used by various other projects and third party products
- Upstream Repositories:
<https://github.com/quay/clair>





The screenshot shows the Red Hat Quay interface for a Python image. At the top, it says 'Quay Security Scanner has detected 718 vulnerabilities. Patches are available for 144 vulnerabilities.' Below this is a pie chart showing the distribution of vulnerabilities by severity: 7% High-level, 37% Medium-level, 31% Low-level, 23% Negligible-level, and 8% Unknown-level. A table below lists 144 vulnerabilities, including:

CVE	SEVERITY	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN LAYER
CVE-2018-15686	10 / 10	systemd	232-25+deb9u6	232-25+deb9u7	ADD file: a1c14b182521b3a7f1998bd07ac48304bc...
CVE-2019-3855	9.3 / 10	libssh2	1.7.0-1	1.7.0-1+deb9u1	RUN apt-get update & apt-get install -y --re...
CVE-2019-3462	9.3 / 10	apt	1.4.8	1.4.9	ADD file: a1c14b182521b3a7f1998bd07ac48304bc...
CVE-2017-16997	9.3 / 10	glibc	2.24-11+deb9u3	2.24-11+deb9u4	ADD file: a1c14b182521b3a7f1998bd07ac48304bc...



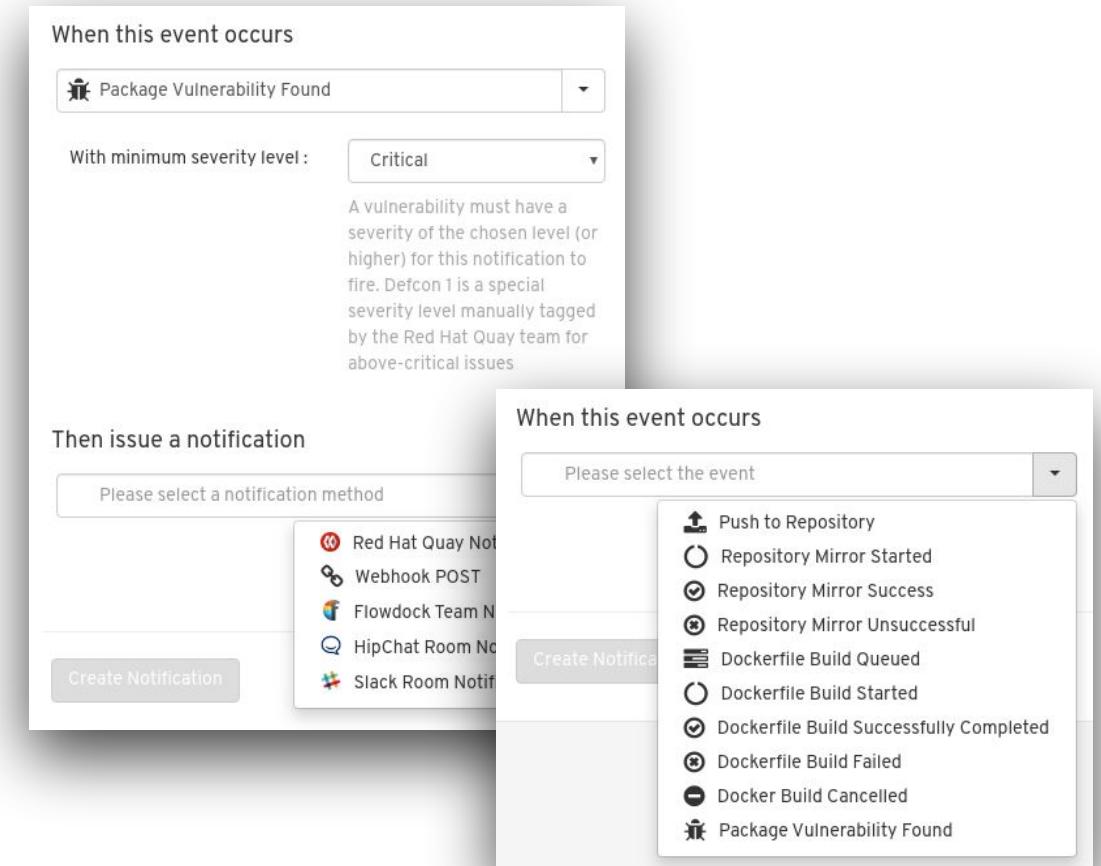
Clair v4 (Tech Preview with Quay 3.3)

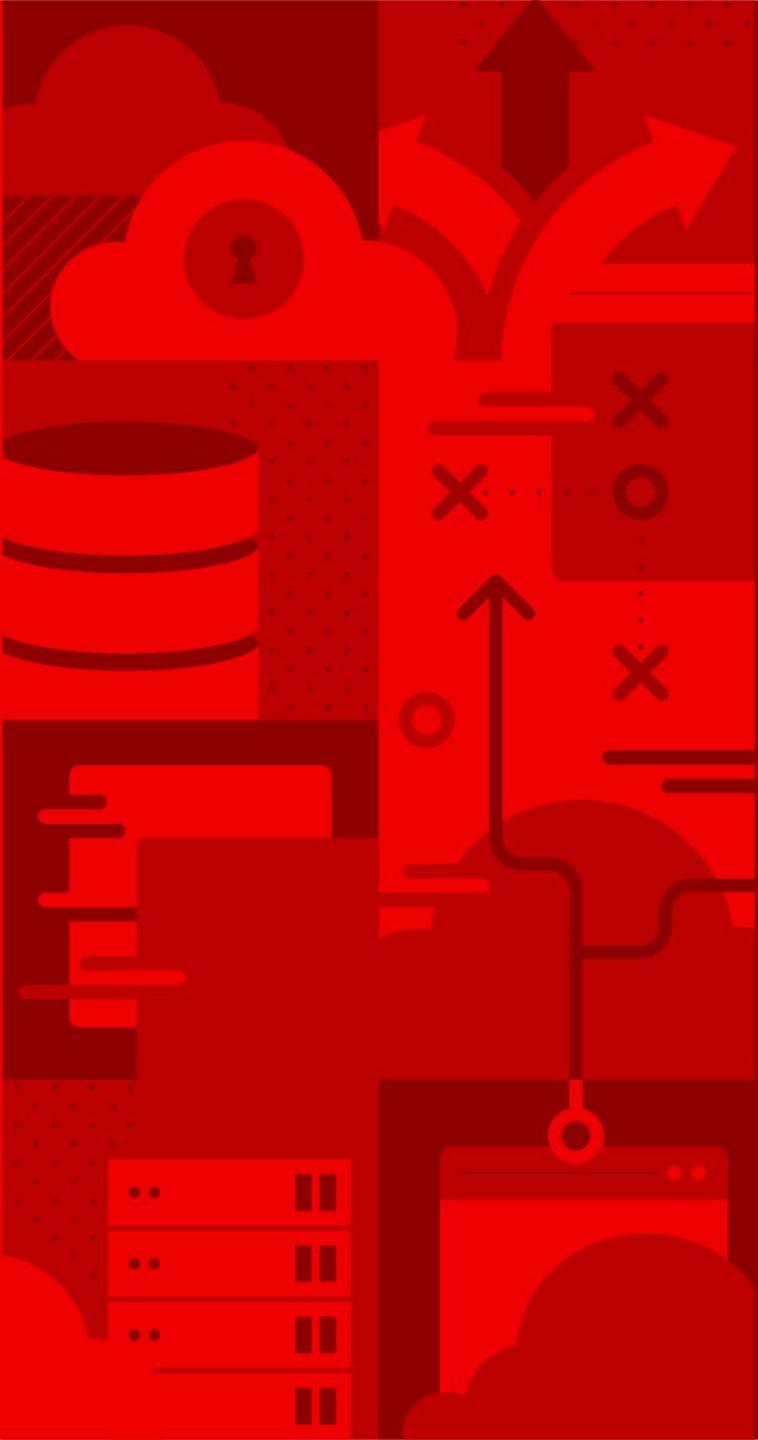
Clair v4 is the newest version of Clair after a massive refactoring in order to make several big enhancements possible. This includes:

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

Notifications for Vulnerabilities found by Clair

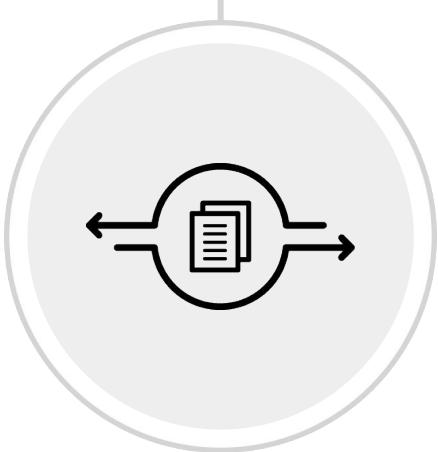
- **Quay triggers different notifications for various repository events** (depends on enabled features)
- This includes the event type “**Package Vulnerability Found**”
- Additional Filter can be applied for **Severity Level**
- **Various Notification Methods**
- Custom Notification Title (optional)





Container Security Operator and OpenShift Console Integration

Quay Container Security Operator (CSO)



The screenshot shows the Red Hat OpenShift Container Platform dashboard under the 'Administrator' account. The 'Dashboards' section includes:

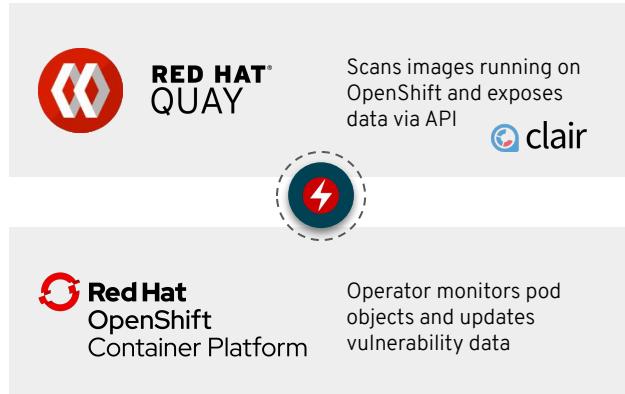
- Cluster API Address:** https://api-jcain1.devcluster.openshift.com:6443
- Cluster ID:** a108f2f6-4a64-4fde-8b74-31a9ca868fb
- Provider:** AWS
- OpenShift Version:** 4.4.0-01-2020-01-07-061008
- Update Channel:** stable-4.3

Status: Cluster (green), Control Plane (green), Quay Image Security (red, 12 vulnerabilities).

Image Security Breakdown: 1 Critical, 6 High, 1 Medium, 1 Low, 1 Negligible, 1 Unknown. A pie chart shows 12 total vulnerabilities across namespaces.

Alerts: 3 critical and 1 warning alerts related to cluster operator availability and degraded status.

Cluster Utilization: CPU usage over the last hour.



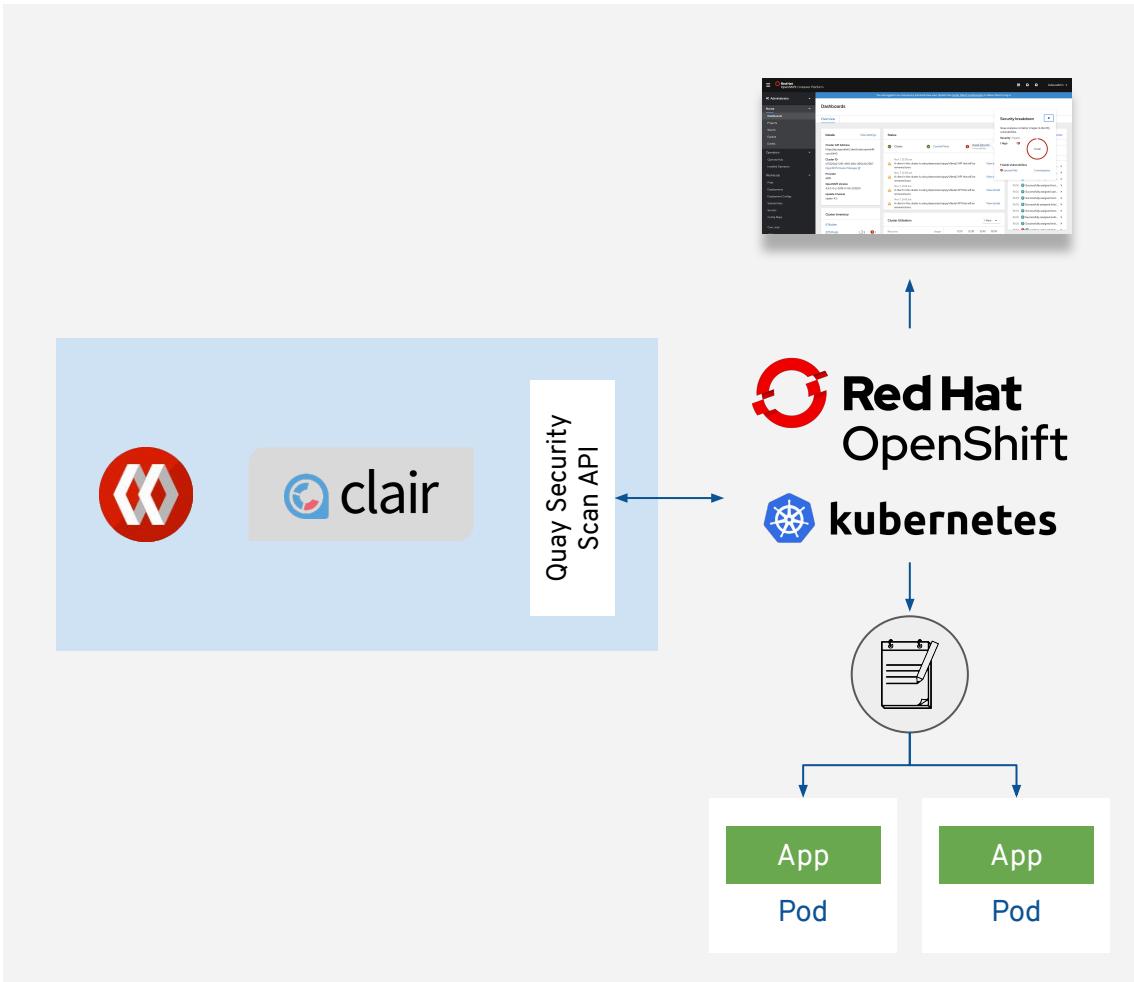
Container Security Operator - Vulnerability Data in OpenShift

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

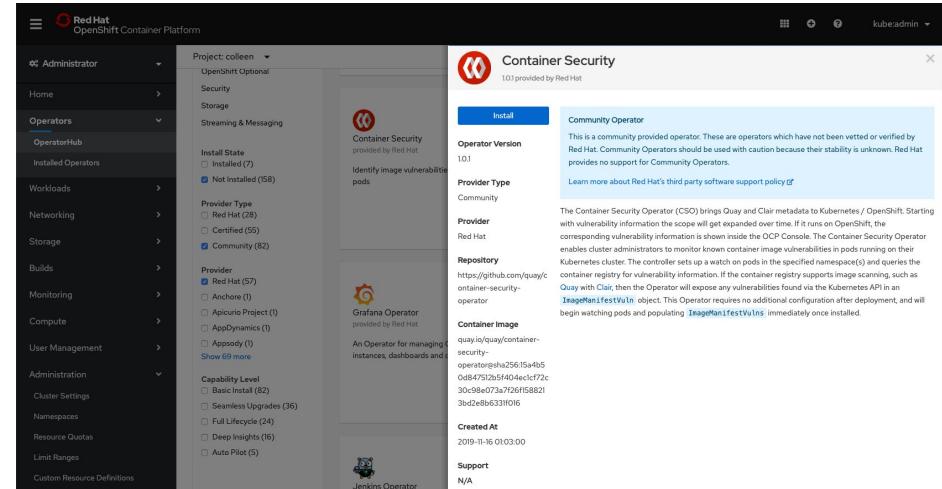
Container Security Operator (CSO)



- 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)

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



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

OpenShift Console Integration via CSO

Vulnerability	Severity	Package	Current Version	Fixed in Version
RHSA-2019:4190	High	nss-softokn-freebl	3.44.0-5.el7	0:3.44.0-8.el7_7
RHSA-2019:4190	High	nss-util	3.44.0-3.el7	0:3.44.0-4.el7_7
RHSA-2019:4190	High	nss-tools	3.44.0-4.el7	0:3.44.0-7.el7_7
RHSA-2019:4190	High	nss-softokn	3.44.0-5.el7	0:3.44.0-8.el7_7

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/>

Vulnerability Data inside the OpenShift Console

The screenshot shows the OpenShift Console interface with the sidebar menu open. The 'Image Manifest Vulnerabilities' section is selected. A table lists vulnerabilities across various namespaces and images, including their severity, affected pods, fixable status, and manifest hash.

Image Name	Namespace	Highest Severity	Affected Pods	Fixable	Manifest
VULN alecmerdler/bad-pod	NS default	Medium	1	0	35c1c5688e7 ↗
VULN alecmerdler/bad-image	NS skynet	Unknown	1	1	4bc210f89d7 ↗
VULN alecmerdler/bad-pod	NS default	Critical	1	0	7d4aae77622 ↗
VULN 3scale/3scale-operator	NS default	High	1	24	9a6536efbb5 ↗
VULN alecmerdler/bad-image	NS skynet	Unknown	1	1	b025832c073 ↗
VULN alecmerdler/bad-pod	NS default	Low	1	0	e94c22ba519 ↗
VULN alecmerdler/bad-pod	NS default	Defcon 1	1	0	f4cd12ac979 ↗

ImageManifestVuln list view

Vulnerability Data inside the OpenShift Console

The screenshot shows the Quay Container Security Operator (CSO) interface. The top navigation bar shows 'Project: default'. Below it, the path 'ImageManifestVuln > ImageManifestVuln Details' is visible. The main title is 'VULN 3scale/3scale-operator@9a6536efbb5'. The interface includes a 'Overview' tab (which is selected), 'YAML' and 'Affected Pods' tabs. The 'Image Manifest Vuln Overview' section displays a donut chart with the total count of vulnerabilities. Below the chart, text indicates 'Quay Security Scanner has detected 24 vulnerabilities.' and 'Patches are available for 24 vulnerabilities.' with breakdowns for High, Medium, and Low severity levels.

Name	Registry
sha256.9a6536efbb5f23ff4a2c2d76065c1c37a84dc7404da259cd9e5f7lb637d28f6	quay.io/3scale/3scale-operator

Details for the image manifest include:

- Namespace:** NS default
- Labels:** default/3scale-operator-7864b9bb5d-frht=true
- Annotations:** 0 Annotations
- Created At:** Dec 23, 2019 10:30 am
- Owner:** None

The screenshot shows the OpenShift console interface. The top navigation bar shows 'Project: skynet'. Below it, the path '0 Annotations' is visible. The main title is 'Created At' (Jan 6, 11:42 am). The interface includes a 'Vulnerabilities' section with a table listing various vulnerabilities.

CVE	Severity	Package	Current Version	Fixed in Version
RHSA-2019-4190	High	nss-softokn	3.36.0-5.el7_5	0:3.44.0-8.el7_7
RHSA-2019-4190	High	nss-sysinit	3.36.0-7.el7_6	0:3.44.0-7.el7_7
RHSA-2019-4190	High	nss-softokn-freebl	3.36.0-5.el7_5	0:3.44.0-8.el7_7
RHSA-2019-4190	High	nss-util	3.36.0-11.el7_6	0:3.44.0-4.el7_7
RHSA-2019-4190	High	nss	3.36.0-7.el7_6	0:3.44.0-7.el7_7
RHSA-2019-4190	High	nss-tools	3.36.0-7.el7_6	0:3.44.0-7.el7_7
RHSA-2019-2237	Medium	nss-softokn	3.36.0-5.el7_5	0:3.44.0-5.el7
RHSA-2019-2237	Medium	nss-sysinit	3.36.0-7.el7_6	0:3.44.0-4.el7
RHSA-2019-2237	Medium	nss-softokn-freebl	3.36.0-5.el7_5	0:3.44.0-5.el7
RHSA-2019-2237	Medium	nss-util	3.36.0-11.el7_6	0:3.44.0-3.el7
RHSA-2019-2304	Medium	openssl-libs	1:1.0.2k-16.el7_6.1	1:1.0.2k-19.el7
RHSA-2019-2118	Medium	glibc-common	2.17-260.el7_6.4	0:2.17-292.el7

ImageManifestVuln detail view

Vulnerability Data inside the OpenShift Console

The screenshot shows the OpenShift console interface. On the left is a dark sidebar menu with various navigation items: Home, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, and User Management. The 'Workloads' item is currently selected and expanded, showing sub-options like Overview, YAML, and Affected Pods. The 'Affected Pods' tab is active, indicated by an underline. The main content area displays a table of affected pods. The table has columns for Name, Namespace, and Created. One pod is listed: '3scale-operator-7864b9bb5d-frhnt' in the 'default' namespace, created '11 days ago'. A 'Filter by name...' input field is located at the top right of the table area. The URL in the browser's address bar is 'ImageManifestVuln > ImageManifestVuln Details'.

Name	Namespace	Created
3scale-operator-7864b9bb5d-frhnt	default	11 days ago

ImageManifestVuln detail view (affected pods)

Vulnerability Data inside the OpenShift Console

The screenshot shows the OpenShift Console interface. On the left is a dark sidebar with a navigation menu:

- Administrator
- Home
- Operators
- Workloads
- Networking
- Storage
- Builds
- Monitoring
- Compute
- User Management

At the top right, there is a dropdown for "Project: default". Below it, the breadcrumb navigation shows "ImageManifestVuln > ImageManifestVuln Details". The main content area displays a vulnerability detail for the image "3scale/3scale-operator@9a6536efbb5" with a "VULN" status indicator.

Below this, there are three tabs: "Overview", "YAML", and "Affected Pods", with "Affected Pods" being the active tab. A search bar labeled "Filter by name..." is located on the right side of the table.

A table lists the affected pods:

Name	Namespace	Created	Actions
3scale-operator-7864b9bb5d-frhnt	default	11 days ago	⋮

Kebab action on Pods list view

OpenShift Console Vulnerability Information Enhancements

The image displays two screenshots of the OpenShift Container Platform console, illustrating enhanced vulnerability information features.

Left Screenshot (Detailed View):

- Header:** Red Hat OpenShift Container Platform, Project: knative-eventing, kubeadmin.
- Breadcrumbs:** ImageManifestVuln > ImageManifestVuln Details.
- Section:** IMV openshift-knative/knative-e-evening-channel-controller@08aed83clbb
- Sub-section:** Details, YAML, Affected Pods.
- Content:**
 - Image Manifest Vuln Details:** Quay Security Scanner has detected 7 vulnerabilities. Patches are available for 7 vulnerabilities. 7 High vulnerabilities.
 - Image Details:** Name: sha256:08aed83clbb9f510d0f2c4dc64993fa333bad32d90bc08e4fcfc82ff6, Registry: quay.io/openshift/knative/knative-e-evening-channel-controller.
 - Annotations:** 0 Annotations.
 - Created At:** Mar 23, 4:40 pm.
 - Owner:** No owner.
- Vulnerabilities:** A table listing vulnerabilities with columns: Vulnerability, Severity, Package, Current Version, Fixed in Version.

Right Screenshot (List View):

- Header:** Red Hat OpenShift Container Platform, Project: knative-eventing, kube:admin.
- Breadcrumbs:** ImageManifestVuln > Image Manifest Vulnerabilities.
- Section:** Image Manifest Vulnerabilities.
- Filter:** Filter by name... (with a search icon).
- Table:** A list of Image Name, Namespace, Highest Severity, Affected Pods, Fixable, and Manifest for four different images, all labeled as High severity and affecting 1 pod each.

Image Name	Namespace	Highest Severity	Affected Pods	Fixable	Manifest
IMV openshift-knative/knative-e-evening-channel-controller	NS knative-eventing	High	1	7	08aed83clbb
IMV openshift-knative/knative-e-evening-sources-controller	NS knative-eventing	High	1	7	32f3ca637fd
IMV openshift-knative/knative-e-evening-controller	NS knative-eventing	High	1	7	cc4ec0d71b8
IMV openshift-knative/knative-e-evening-webhook	NS knative-eventing	High	1	7	e3bb2c01ddf

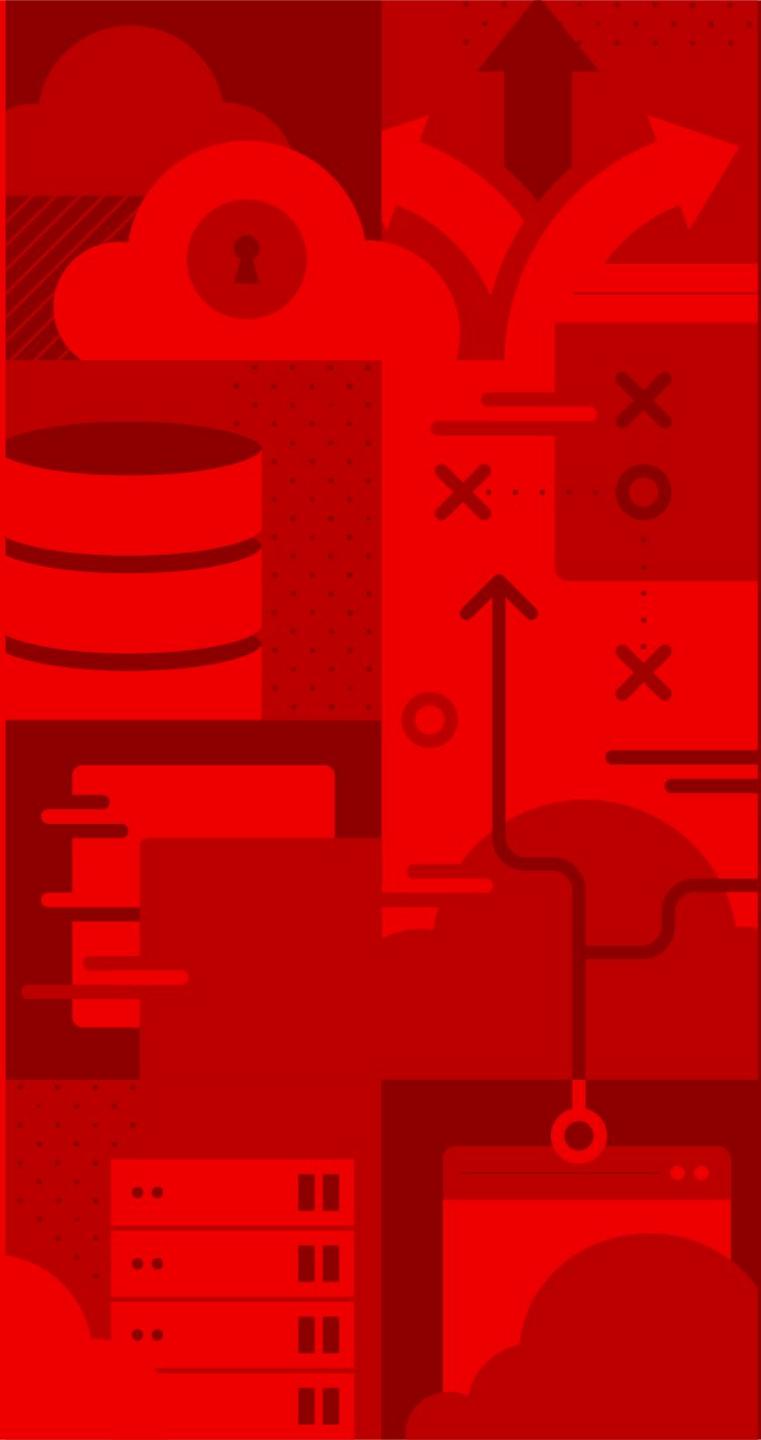


Image Rollback / Time Machine

Quay Time Machine

 Time Machine

Time machine keeps older copies of tags within a repository for the configured period of time, after which they are garbage collected. This allows users to revert tags to older images in case they accidentally pushed a broken image. It is highly recommended to have time machine enabled, but it does take a bit more space in storage.

Allowed expiration periods:

- 0s [Remove](#)
- 1d [Remove](#)
- 1w [Remove](#)
- 2w [Remove](#)
- 4w [Remove](#)

The expiration periods allowed for configuration. The default tag expiration *must* be in this list.

Default expiration period:

The default tag expiration period for all namespaces (users and organizations). Must be expressed in a duration string form: `30m`, `1h`, `1d`, `2w`.

Allow users to select expiration: [Enable Expiration Configuration](#)

If enabled, users will be able to select the tag expiration duration for the namespace(s) they administrate, from the configured list of options.

Time Machine registry wide settings are defined via Quay config app

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

Image Rollback / Time Machine

Thursday, July 6th 2017		
v74 was created pointing to SHA256 edf795b61646	Jul 6th 2017, 9:57:48 am	
latest was moved to SHA256 e74e01c03118 from SHA256 462022dd88cc	Jul 6th 2017, 9:56:10 am	
master was moved to SHA256 caf1bc19dc30 from SHA256 cdcc9d5cb96c	Jul 6th 2017, 9:56:08 am	
Wednesday, July 5th 2017		
v73 was created pointing to SHA256 ed54e21a257d	Jul 5th 2017, 11:22:05 am	
latest was moved to SHA256 462022dd88cc from SHA256 d25e89ea6ebd	Jul 5th 2017, 11:13:29 am	Restore latest to SHA256 d25e89ea6ebd
master was moved to SHA256 cdcc9d5cb96c from SHA256 aa59a906145b	Jul 5th 2017, 11:13:27 am	Restore master to SHA256 aa59a906145b

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

Repository Tags					
TAG	LAST MODIFIED	SECURITY SCAN	SIZE	IMAGE	
<input checked="" type="checkbox"/> latest	16 hours ago	70 Medium · 10 fixable	711.0 MB	SHA256 9a347939468e	
<input type="checkbox"/> master	16 hours ago	70 Medium · 10 fixable	711.0 MB	SHA256 014514e8ef9b	
<input type="checkbox"/> dbb57f7	18 hours ago	70 Medium · 10 fixable	696.1 MB	SHA256 2592c71fe8f5	
<input type="checkbox"/> 3e28797	a day ago	75 Medium · 15 fixable	693.5 MB	SHA256 0d37d281173e	

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

Try it out!

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

SUBSCRIPTIONS DOWNLOADS CONTAINERS SUPPORT CASES

Red Hat CUSTOMER PORTAL

Products & Services Tools Security Community

Red Hat Quay

Products & Services > Red Hat Quay

WHAT'S NEW GET STARTED KNOWLEDGE SUPPORT

Red Hat Quay 3

Release Notes ▶

News

Introducing Red Hat Quay V3: A container registry tailored for the enterprise
2019-06-19T11:28:20+00:00

Check out the Quay datasheet on redhat.com
2018-11-09T15:35:22+00:00

Red Hat® Quay is a secure, private container registry that builds, analyzes and distributes container images. It provides a high level of automation and customization. Red Hat Quay is also available as a hosted service called Quay.io.

REQUEST AN EVALUATION



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

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](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat