

OpenShift 4.x Architecture Workshop

Enterprise Registry QUAY



July 2019

What Is Quay?

- Market leading enterprise container registry
- Available on-premise, on public cloud and as a hosted service (SaaS)
- Key strengths:
 - Security
 - Robustness & speed
 - Automation
- Quay works with any container environment or orchestration platform



First hosted registry in the market with private repos

2nd biggest hosted registry overall

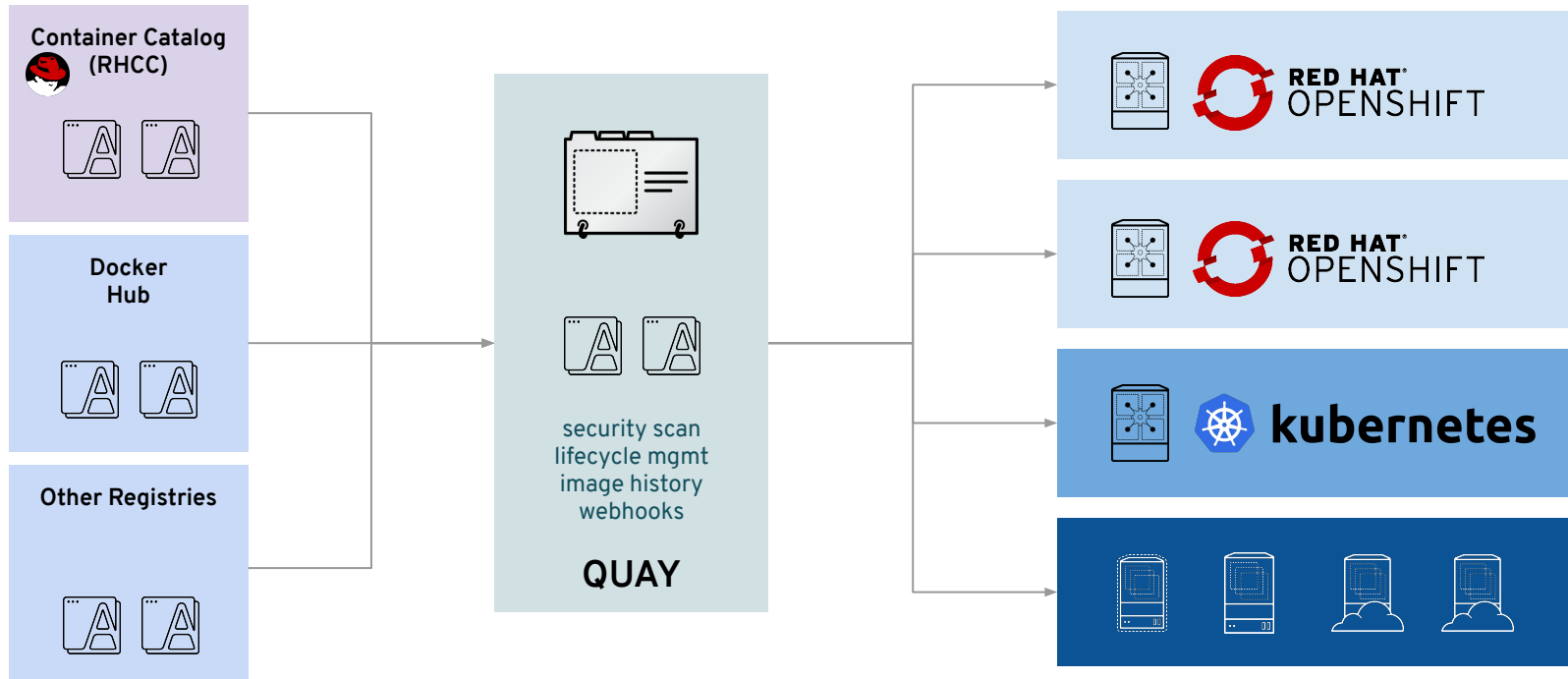
Red Hat Quay Feature Highlights

Security	Robustness and Speed	Automation
Support multiple authentication systems and identity providers	High availability & scalability	Build triggers
Vulnerability scanning	Geo-synchronous replication	Git hook compatible
Encrypted CLI passwords	Continuous, zero-downtime garbage collection	Robot accounts
Detailed logging for auditing	Torrent Distribution	Webhooks
Orgs & team support	Integration with multiple storage backends	Extensible API

Quay Use Cases

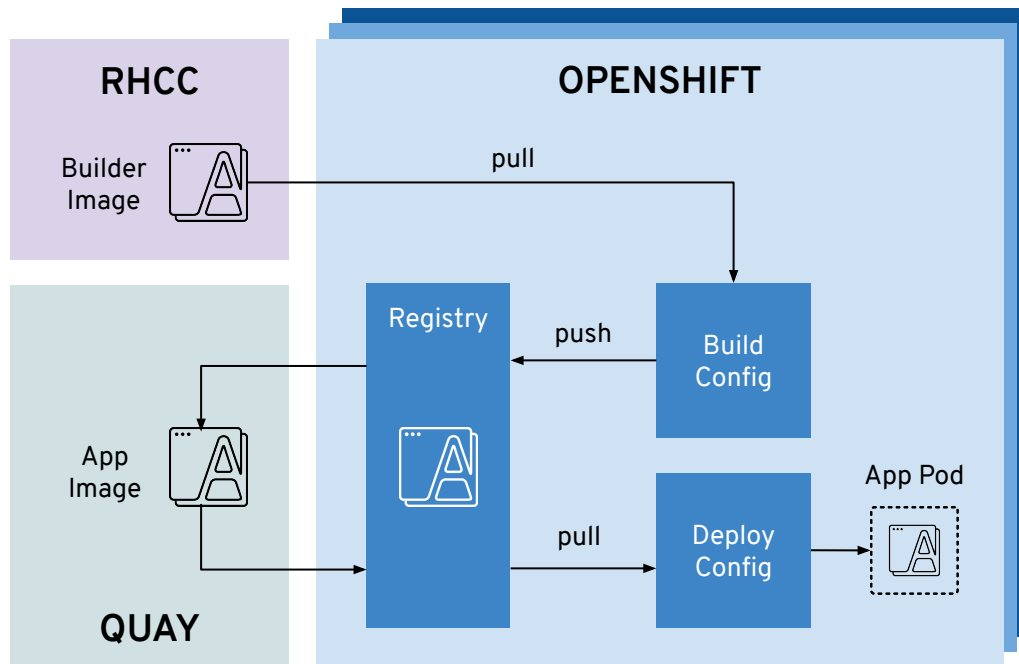
- Large-scale and distributed environments (thousands of users and images)
- Customer has multiple OpenShift/Kubernetes clusters (content ingress)
- Customer needs OpenShift/Kubernetes in multiple geographical regions
- Customer needs governance for container images (scanning)
- Customer has high image maintenance and automation requirements
- Large number of build and high requirements on image delivery throughput

Content Ingress with Quay



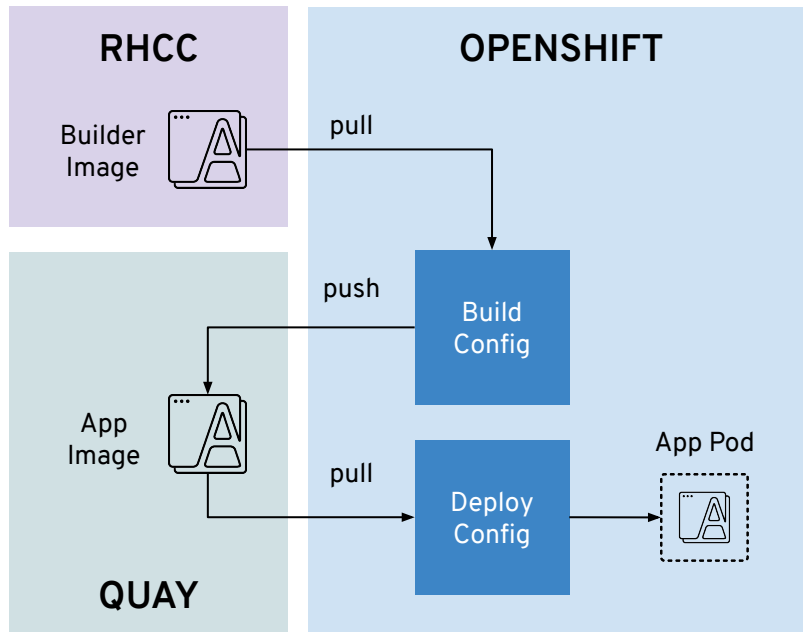
Quay as Upstream Registry with OpenShift

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



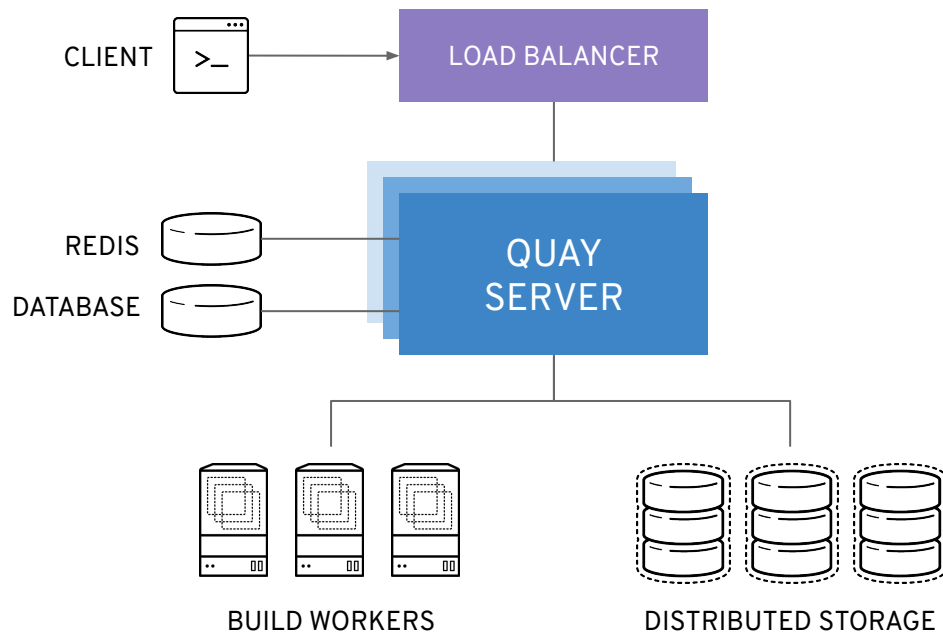
Quay as OpenShift Registry

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

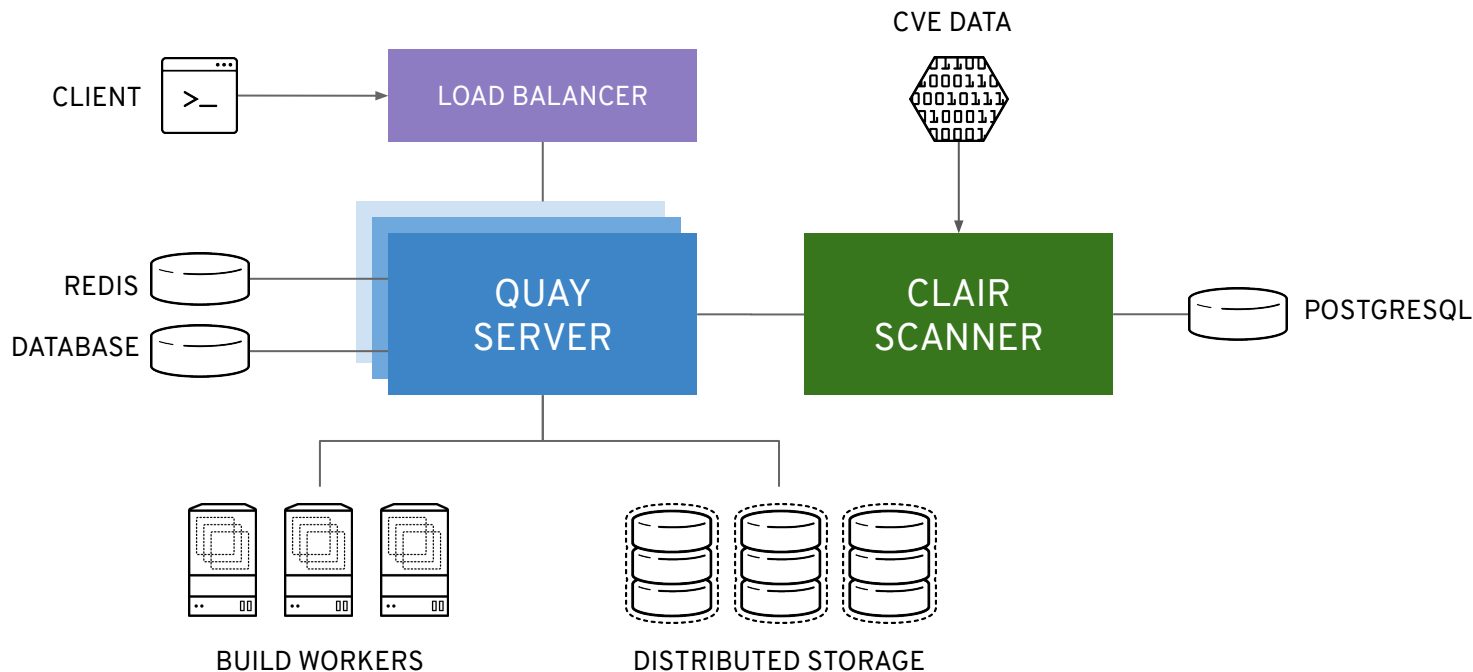


Quay Architecture

Quay Architecture

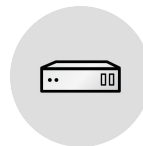


Quay Architecture with Image Scanning



Underlying Infrastructures Quay can run

- Quay can run on
 - standalone container host
 - (Tectonic) / Kubernetes / OpenShift
- Quay runs on any public cloud infrastructure as well
 - Quay.io runs on AWS
- Reference Architectures in planning



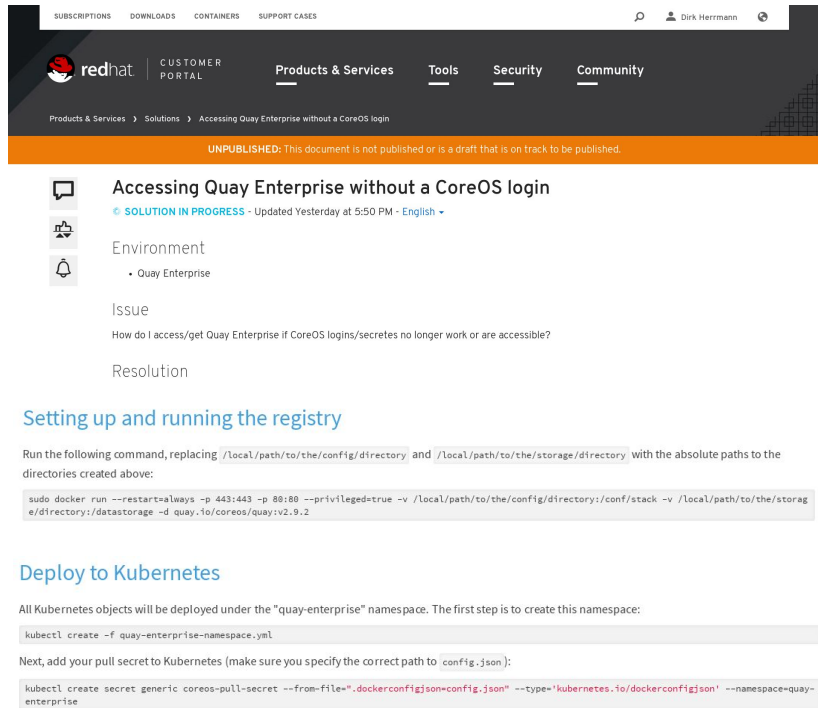
Underlying Infrastructure

- Quay is shipped as container images
 - Images are distributed via Quay.io (will move to RHCC later)
 - Required secret to pull them in customer portal (requires login)

<https://access.redhat.com/solutions/3533201>

- Install procedure documentation at

https://access.redhat.com/documentation/en-us/red_hat_quay/2.9/



The screenshot shows the Red Hat Customer Portal interface. At the top, there are navigation links for SUBSCRIPTIONS, DOWNLOADS, CONTAINERS, and SUPPORT CASES. Below this is a header with the Red Hat logo and 'CUSTOMER PORTAL'. The main navigation bar includes 'Products & Services', 'Tools', 'Security', and 'Community'. A sub-navigation bar shows 'Products & Services' > 'Solutions' > 'Accessing Quay Enterprise without a CoreOS login'. A status bar indicates 'UNPUBLISHED: This document is not published or is a draft that is on track to be published.' The article title is 'Accessing Quay Enterprise without a CoreOS login', with a status of 'SOLUTION IN PROGRESS' and a last update of 'Yesterday at 5:50 PM'. The article is categorized under 'Environment' > 'Quay Enterprise'. The 'Issue' section asks 'How do I access/get Quay Enterprise if CoreOS logins/secrets no longer work or are accessible?'. The 'Resolution' section is titled 'Setting up and running the registry' and provides instructions to run a Docker command, replacing placeholder paths with absolute paths. The command is: `sudo docker run --restart=always -p 443:443 -p 80:80 --privileged=true -v /local/path/to/the/config/directory:/conf/stack -v /local/path/to/the/storage/directory:/datastorage -d quay.io/coreos/quay:v2.9.2`. The next section is 'Deploy to Kubernetes', which states that all Kubernetes objects will be deployed under the 'quay-enterprise' namespace. The first step is to create this namespace, with the command: `kubectl create -f quay-enterprise-namespace.yml`. The next step is to add a pull secret to Kubernetes, specifying the correct path to 'config.json': `kubectl create secret generic coreos-pull-secret --from-file=.dockerconfigjson=config.json --type=kubernetes.io/dockerconfigjson --namespace=quay-enterprise`.

Running Quay on OpenShift

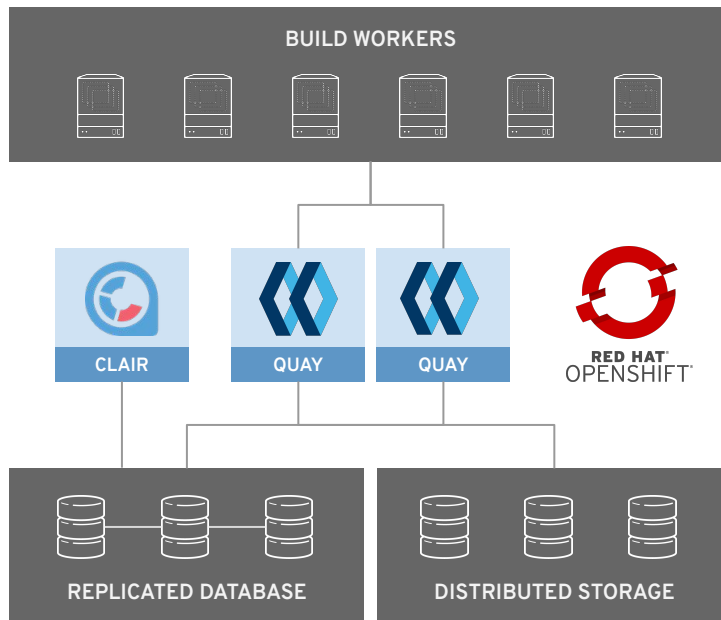
Quay on OpenShift: Recommended Setup

On OpenShift Cluster:

- Quay Enterprise
- Clair

Outside OpenShift cluster:

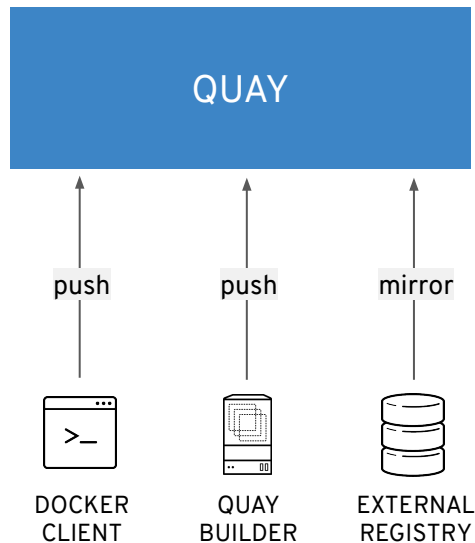
- Database
- Storage
- Builders



Getting Images into QUAY

Getting Images into Quay Registry

- Multiple ways to get images into Quay
 - Push images to Quay
 - Quay builders
 - Repository mirroring (coming soon)
- Any compliant Docker client can push images into Quay
 - OpenShift build config
 - Docker CLI
 - Skopeo (recommended)



Clair



Clair Vulnerability Scanning



Complete Visibility into known vulnerabilities
and how to fix them

Description: Quay integrates with Clair to
continually scans your containers for vuln's.

How it Works:

- Static analysis of vulnerabilities
- Multiple drivers and data sources
- Synchronous update of vuln metadata
- New vuln's trigger notifications
- Rich Clair API
- Can run single-instance or HA



Thank you !