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

RED HAT QUAY

First hosted registry in the market with private repos

2nd biggest hosted registry overall

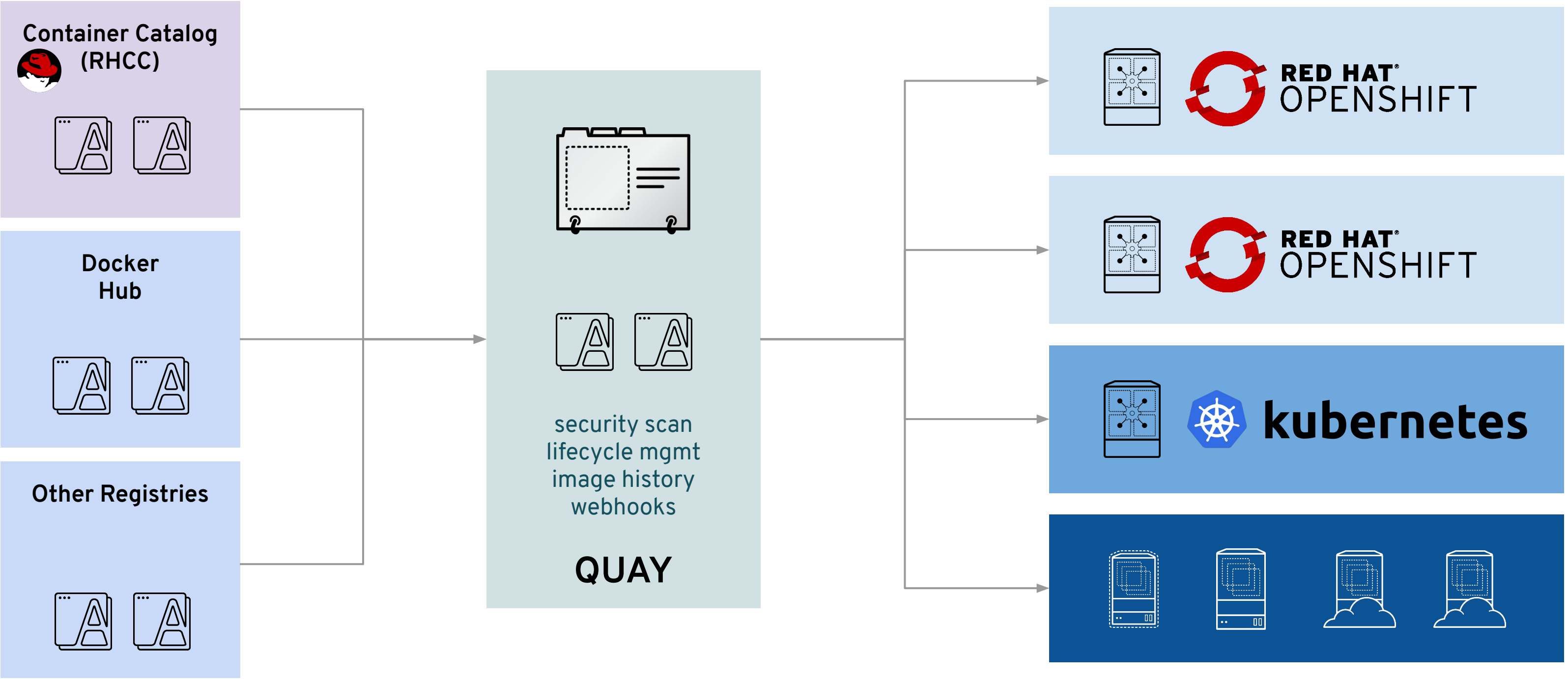
Red Hat Quay Feature Highlights

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

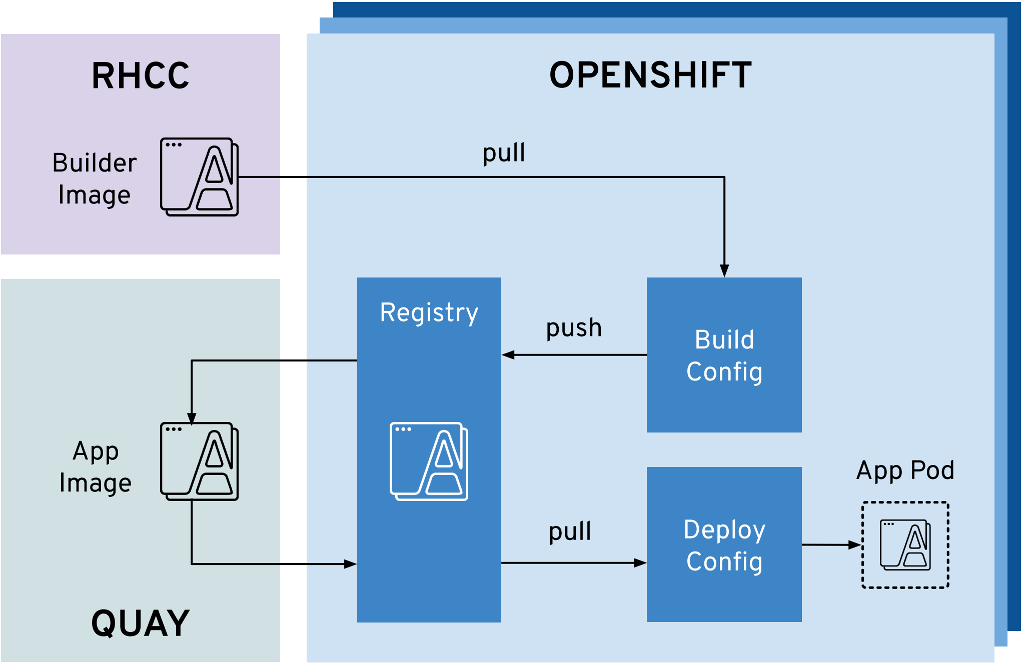
Content Ingress with Quay



Quay as Upstream Registry with OpenShift

- Images pulled from Quay into the integrated OpenShift registry

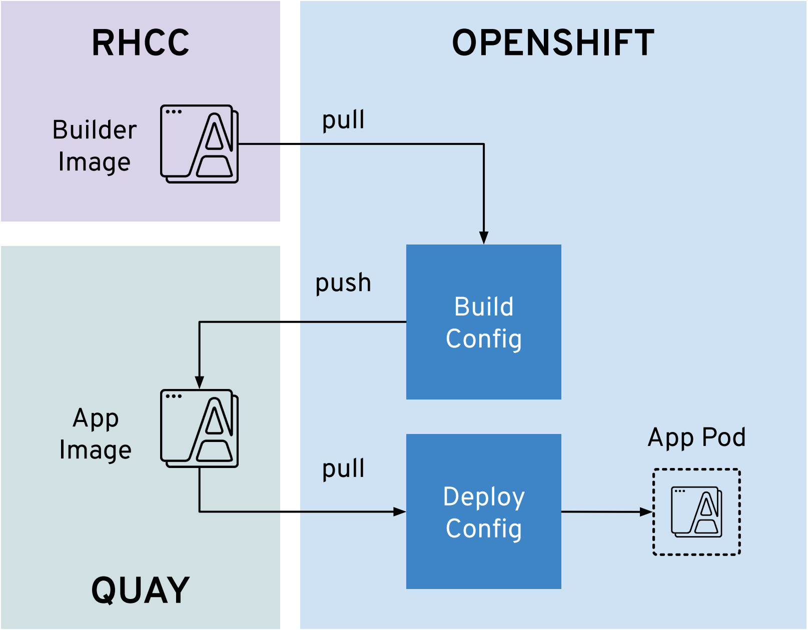
- Images are pushed to the integrated OpenShift registry, and synced externally with Quay



Quay as OpenShift Registry

- Images are pushed directly by builds to Quay

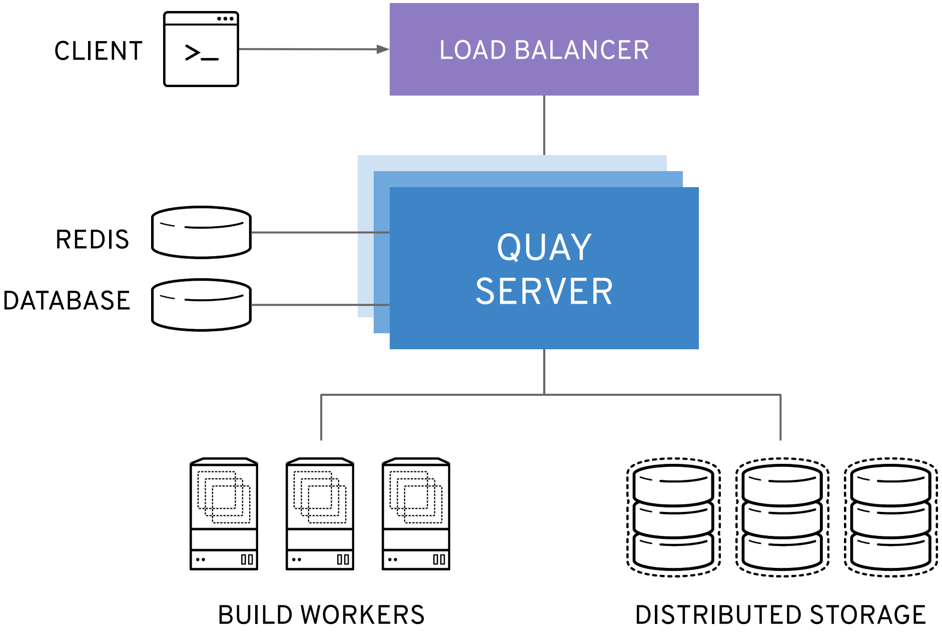
- Images are pulled directly from Quay



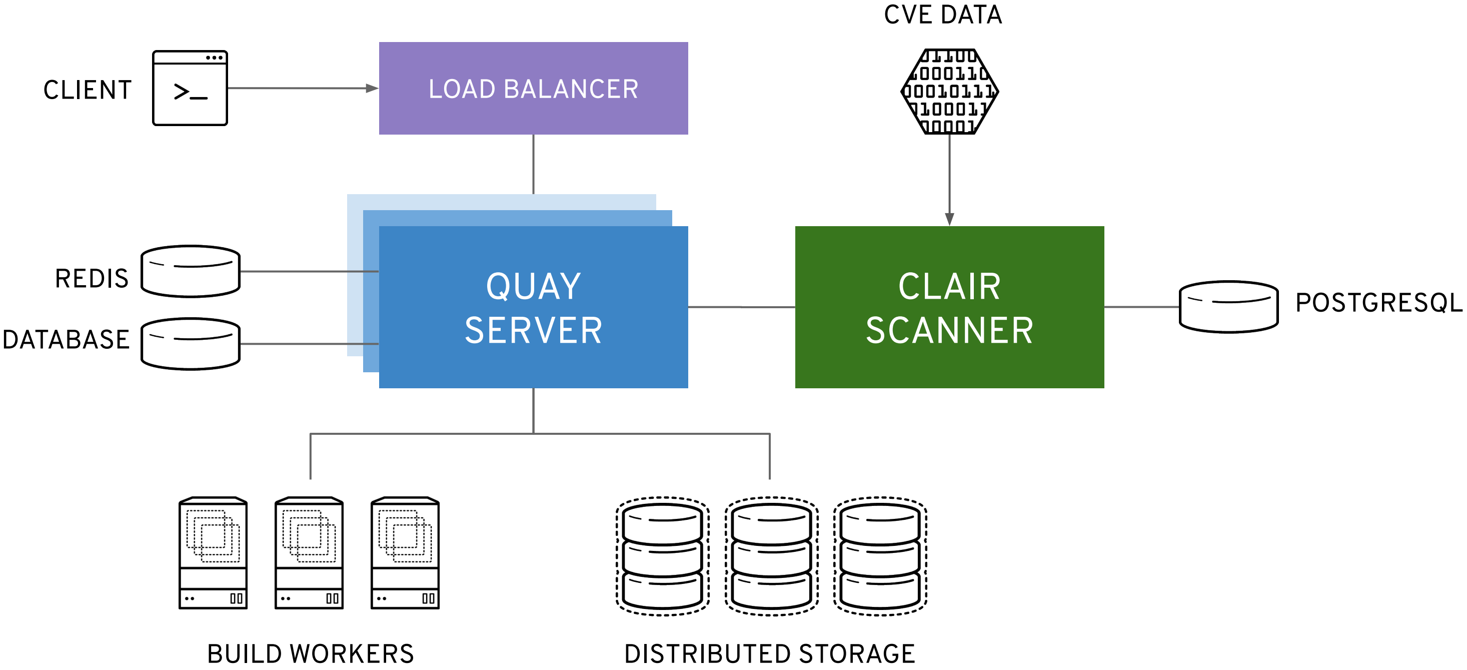
Red Hat

Quay Architecture

Quay Architecture



Quay Architecture with Image Scanning



Prerequisite 1: Supported Database

Available via Red Hat Software Collections but 3rd party works as well

Always favor PostgreSQL

Clair requires PostgreSQL due to use of recursive queries.

Great for demo/testing

Only MySQL 5.7+

RECOMMENDATION: Customers should have DBA group manage the database, or to use a managed database solution such as RDS.

Prerequisite 2: Storage Engines

- AWS S3
- Google Cloud Storage
- Ceph Rados
- OpenStack Swift
- Azure Blob Storage
- Local Disk Mount (NAS)

Red Hat Gluster Storage Support planned for future releases of Quay.

NOTE: Local Storage and NFS not recommended (see next slide)

Prerequisite 2: Storage Engines

- **Local Storage only for PoC / non-prod environments!**
  - Geo-replication is **not supported** with local storage!
  - No way to switch to another storage engine

- **NFS not recommended for large-scale and production environments!**
  - Many customers will attempt to use the local storage engine with NFS. **Always** steer customers toward another storage engine unless there is **literally no other option**.

Prerequisite 3: Redis Cache

- Provided via Red Hat Software Collections but any other redis works, too
- Mostly used by builds, workers and tutorial
- Data stored is ephemeral in nature, Redis does not **need** to be HA.
- If Redis goes down you will lose access to:
    - Live build logs
    - Tutorial

Quay Setup Sizing Recommendations

- As for any other product there are no “typical sizing recommendations” since sizing heavily depends on a multitude of factors
- However, the **scalability of Quay** is one of its strengths (Quay.io)
- Stateless components can be scaled-out
  - Auto-scaling on kubernetes deployments currently tech-preview
  - Note: Scaling out **stateless components** will add load to stateful components
- Minimum requirements as documented in the Quay Product Docs:
  - Quay: min 2GB, recommended 4GB, 2 or more vCPUs
  - Clair: recommended 1GB RAM, 2 or more vCPUs
  - Clair database requirements for security metadata: min 200MB

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

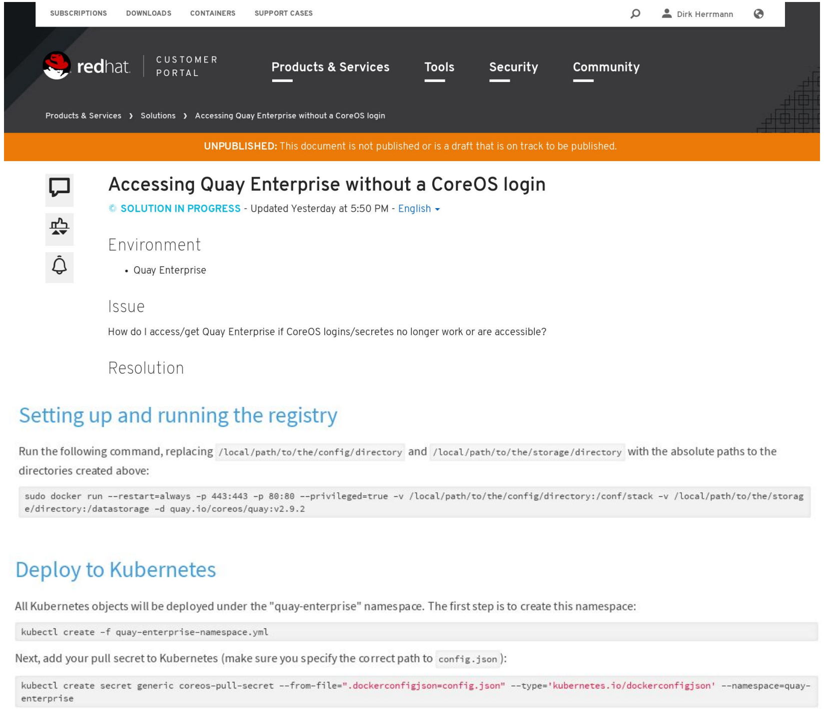
Underlying Infrastructure

- 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/



UNPUBLISHED: This document is not published or is a draft that is on track to be published.

Accessing Quay Enterprise without a CoreOS login

SOLUTION IN PROGRESS - Updated Yesterday at 5:50 PM - English

Environment

- Quay Enterprise

Issue

How do I access/get Quay Enterprise if CoreOS logins/secretes no longer work or are accessible?

Resolution

Setting up and running the registry

Run the following command, replacing `/local/path/to/the/config/directory` and `/local/path/to/the/storage/directory` with the absolute paths to the directories created above:

```
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/storag
e/directory:/datastorage -d quay.io/coreos/quay:v2.9.2
```

Deploy to Kubernetes

All Kubernetes objects will be deployed under the "quay-enterprise" namespace. The first step is to create this namespace:

```
kubectl create -f quay-enterprise-namespace.yml
```

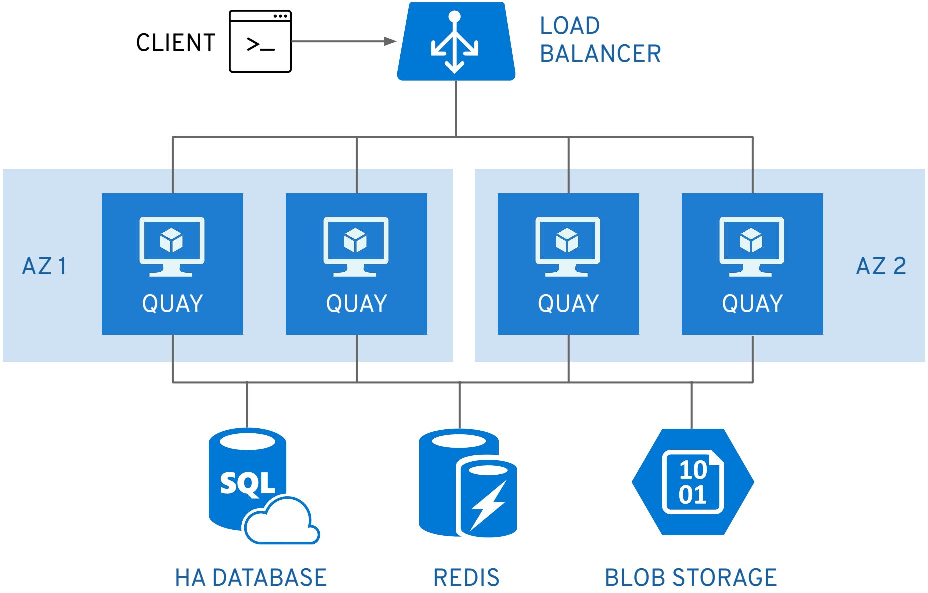
Next, add your pull secret to Kubernetes (make sure you specify the correct path to `config.json`):

```
kubectl create secret generic coreos-pull-secret --from-file=".dockerconfigjson=config.json" --type='kubernetes.io/dockerconfigjson' --namespace=quay-
enterprise
```

How to run Quay Microsoft Azure

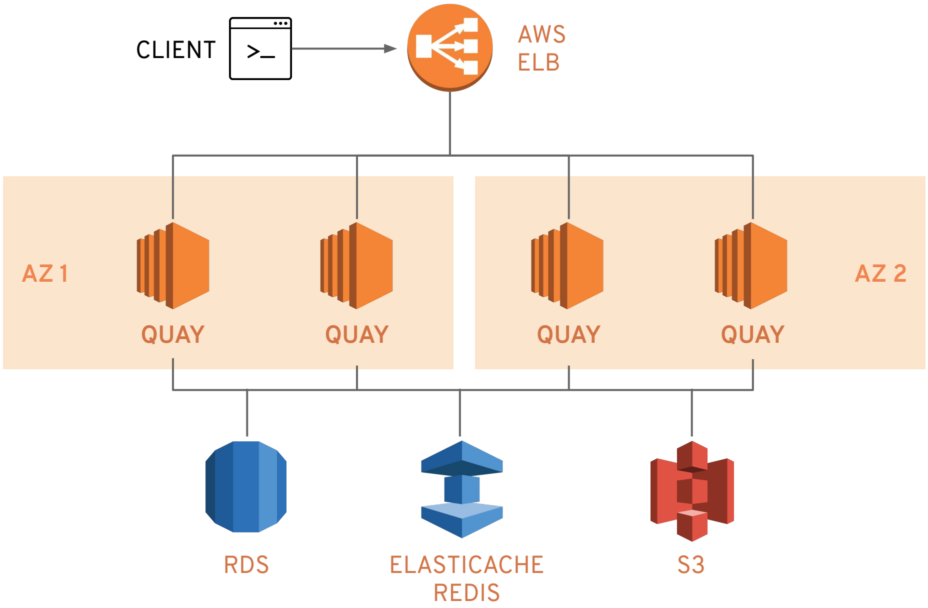
- Utilize Azure managed services such as HA PostgreSQL

- Azure Blob Storage must be **hot storage** (not Azure Cool Blob Storage)



How to run Quay on AWS

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



Running Quay on OpenShift

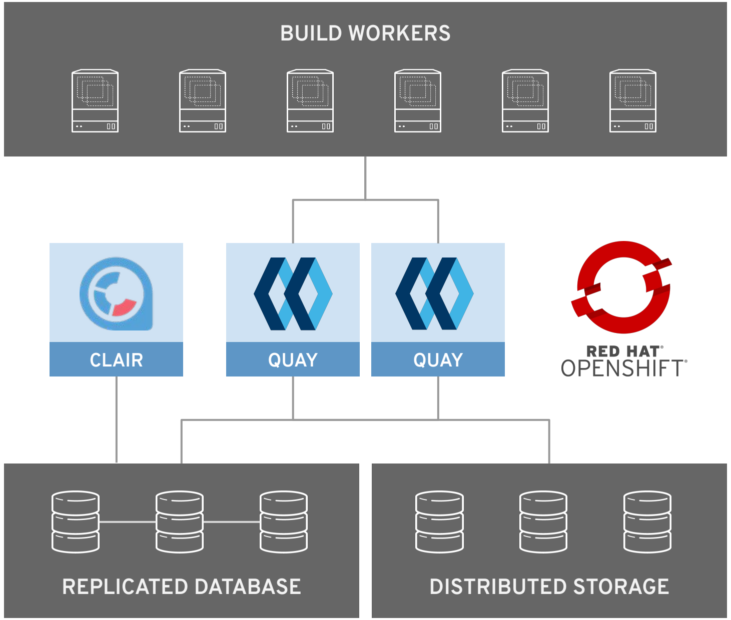
Quay on OpenShift: Recommended Setup

On OpenShift Cluster:

- Quay Enterprise
- Clair

Outside OpenShift cluster:

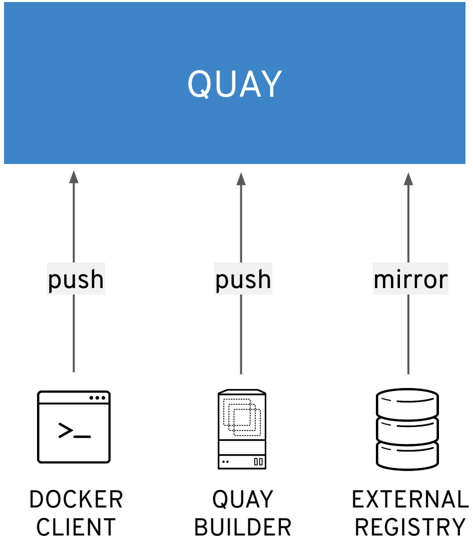
- Database
- Storage
- Builders



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)



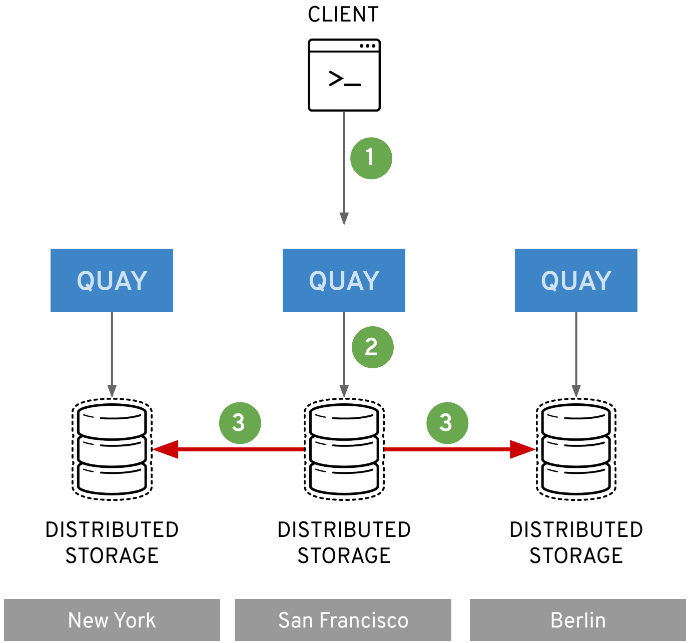
REPLICATION and HA

Quay Geo-Replication

Description: Geo-replication allows for a single globally-distributed Quay Enterprise to serve container images from localized storage

How it Works:

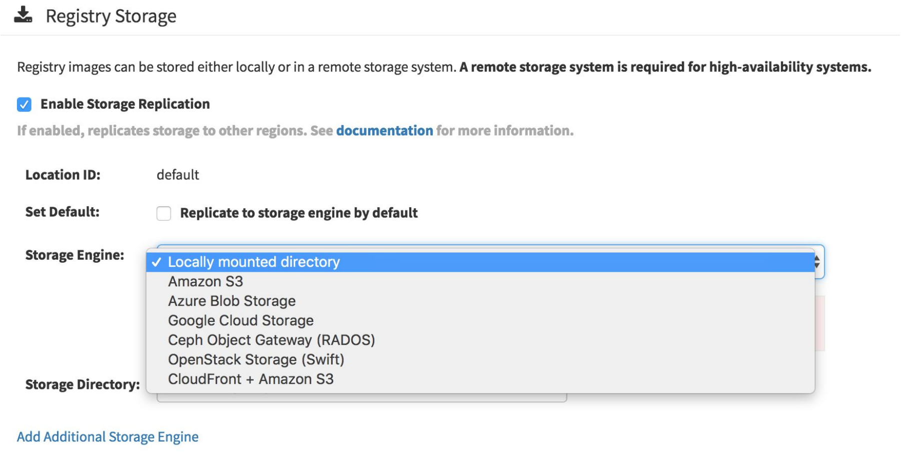
- Image data will be asynchronously replicated in the background to other storage engines
- By default all images are replicated to all storage engines configured



Quay Geo-Replication

Geo-Replication Requirements

- Requires object storage engine in each geographic region
- Local disk storage **<u>not supported</u>**
- Each region must be able to access **every** storage engine
- Contact support if geo-replication on a namespace level needed
- All instances need to be connected to the same database



Registry Storage

Registry images can be stored either locally or in a remote storage system. **A remote storage system is required for high-availability systems.**

☑ **Enable Storage Replication**

If enabled, replicates storage to other regions. See **documentation** for more information.

Location ID: default

Set Default: □ **Replicate to storage engine by default**

Storage Engine:
✓ Locally mounted directory
Amazon S3
Azure Blob Storage
Google Cloud Storage
Ceph Object Gateway (RADOS)
OpenStack Storage (Swift)
CloudFront + Amazon S3

Storage Directory:

Add Additional Storage Engine

Note: Geo-replication occurs in the background. Images are **NOT** immediately localized in all storage engines and regions but **are** immediately **pullable** in all regions

Quay High-Availability Setup

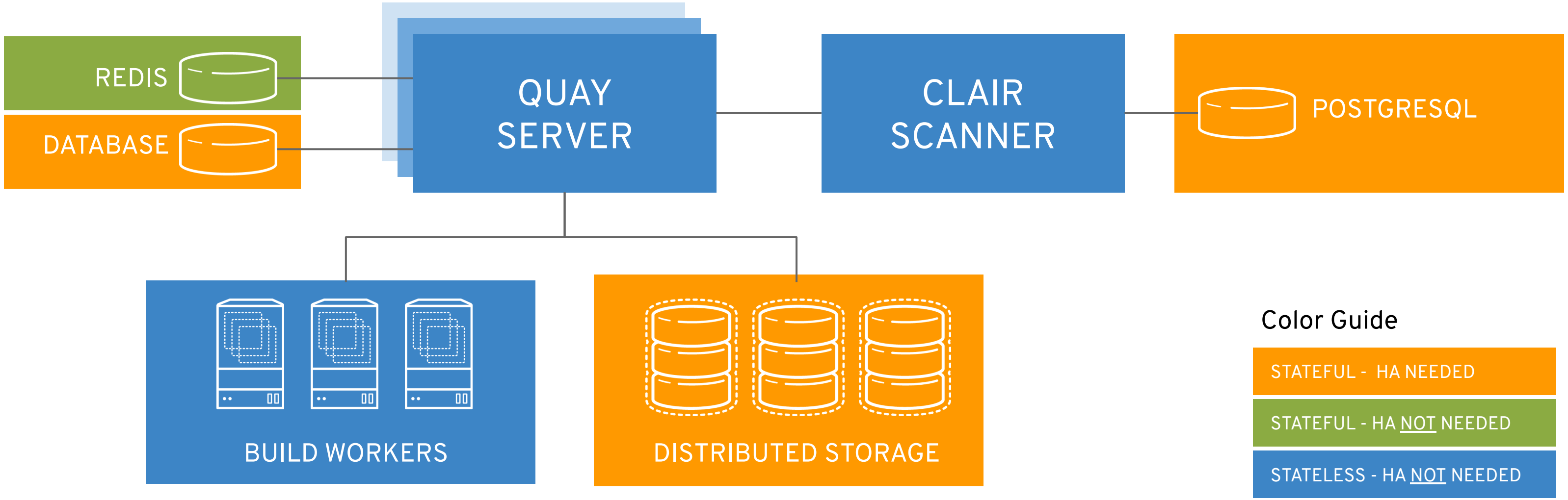
Description: high-availability reference architecture prevents critical single PoF by running multiple instances of Quay

How it Works:

- Stateful components in HA mode
- Stateless components can be horizontally scaled arbitrarily

Note: Scaling out stateless components will add load to the stateful components, which must be accounted for in capacity planning.

Quay Components



REDIS

DATABASE

QUAY SERVER

CLAIR SCANNER

POSTGRESQL

BUILD WORKERS

DISTRIBUTED STORAGE

Color Guide

- 🟧 STATEFUL - HA NEEDED
- 🟩 STATEFUL - HA NOT NEEDED
- 🟦 STATELESS - HA NOT NEEDED

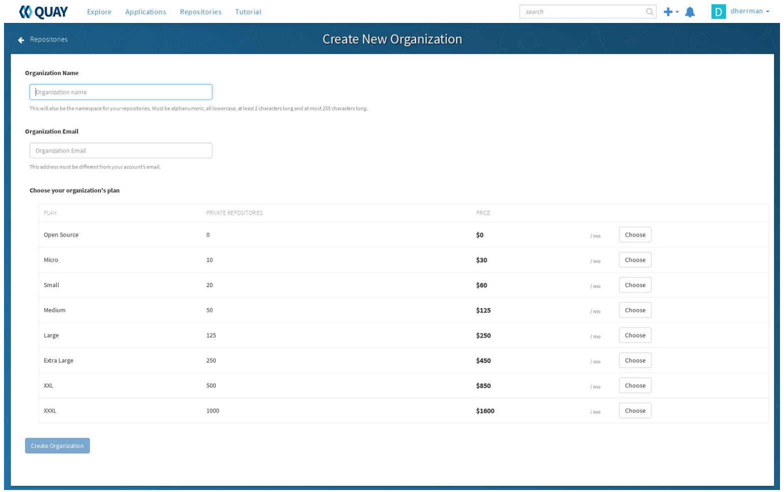
Quay High-Availability Setup

- Required Dependencies:
  - a decent sized **database** with **automatic backup and failover** (Postgres HA, RDS)
  - a **high available distributed storage** engine such as S3, Ceph Rados or SWIFT
  - A redis server running on a medium sized machine (HA not required)
  - A **load balancer** capable of TCP passthrough.
  - At least **three** medium-sized machines for the cluster
- Health checking instances: *https://{instanceip}/health/instance* (OK: 200)
- Health checking cluster: *https://{loadbalancer}/health/endtoend* (OK: 200)
- Autoscaling via monitoring metrics / thresholds

Authentication and Authorisation

Organizations, Teams, Users, Robot Accounts

- **Organizations**
  - sharing repositories under a common namespace that belongs to many users
  - are organized into a set of teams which provide access to repositories under that namespace
- **Teams**
  - Provide a way for an organization to delegate permissions (both global and on specific repositories) to sets or groups of users
  - Permissions: Member, Creator, Admin



Create New Organization

Organization Name

Organization name

This will also be the namespace for your repositories. Must be alphanumeric, all lowercase, at least 2 characters long and at most 255 characters long.

Organization Email

Organization Email

This address must be different from your account's email.

Choose your organization's plan

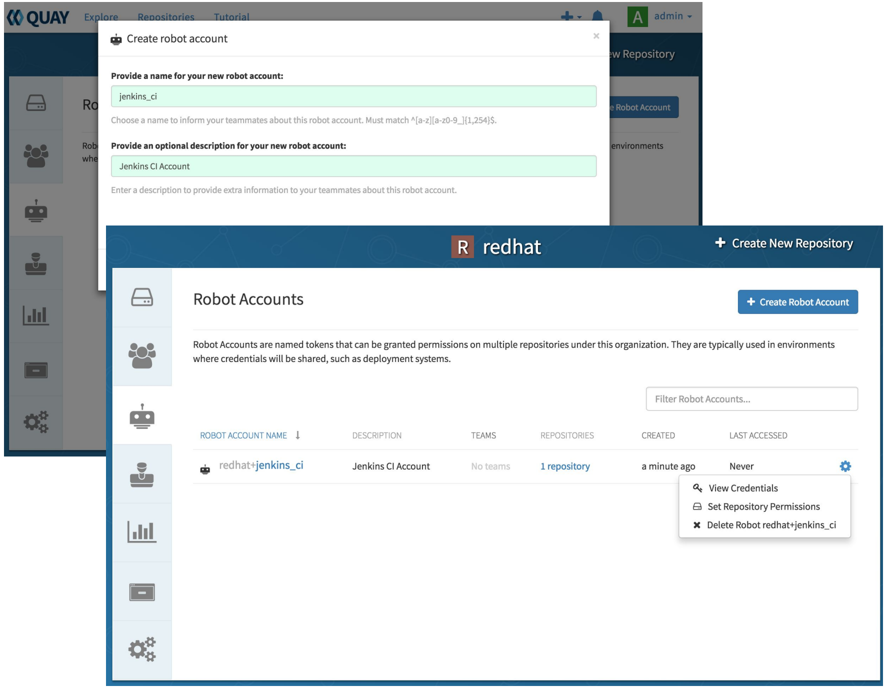
PLAN	PRIVATE REPOSITORIES	PRICE		
Open Source	0	**$0**	/ mo	Choose
Micro	10	**$30**	/ mo	Choose
Small	20	**$60**	/ mo	Choose
Medium	50	**$125**	/ mo	Choose
Large	125	**$250**	/ mo	Choose
Extra Large	250	**$450**	/ mo	Choose
XXL	500	**$850**	/ mo	Choose
XXXL	1000	**$1600**	/ mo	Choose

Create Organization

Organizations, Teams, Users, Robot Accounts

- **Users**
  - Key element of setting repository permissions / RBAC

- **Robot accounts**
  - Allow for automatic software deployments
  - Can be shared by multiple repositories owned by a user or organization
  - Managed inside the organization view -> *Robot Accounts* tab



Create robot account

Provide a name for your new robot account:

jenkins_ci

Choose a name to inform your teammates about this robot account. Must match ^[a-z][a-z0-9_]{1,254}$.

Provide an optional description for your new robot account:

Jenkins CI Account

Enter a description to provide extra information to your teammates about this robot account.

Robot Accounts

+ Create Robot Account

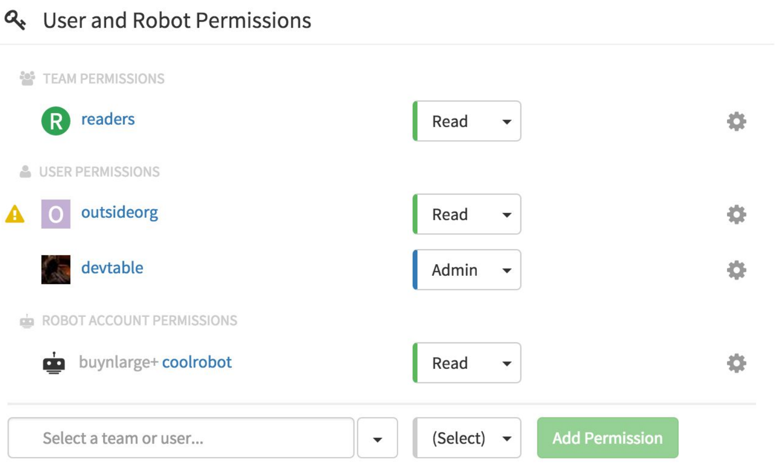
Robot Accounts are named tokens that can be granted permissions on multiple repositories under this organization. They are typically used in environments where credentials will be shared, such as deployment systems.

Filter Robot Accounts...

ROBOT ACCOUNT NAME ↓	DESCRIPTION	TEAMS	REPOSITORIES	CREATED	LAST ACCESSED
redhat+jenkins_ci	Jenkins CI Account	No teams	1 repository	a minute ago	Never

Repository Permissions

- Define which users, robot accounts and teams have can
  - pull (read)
  - push (write)
  - Administer (admin)
- Repository admins can
  - Add new permissions
  - Change existing permissions
  - Revoke permissions



User and Robot Permissions

TEAM PERMISSIONS

readers	Read

USER PERMISSIONS

outsideorg	Read
devtable	Admin

ROBOT ACCOUNT PERMISSIONS

buynlarge+coolrobot	Read

Note: viewing and changing permissions requires **repository admin permission.**

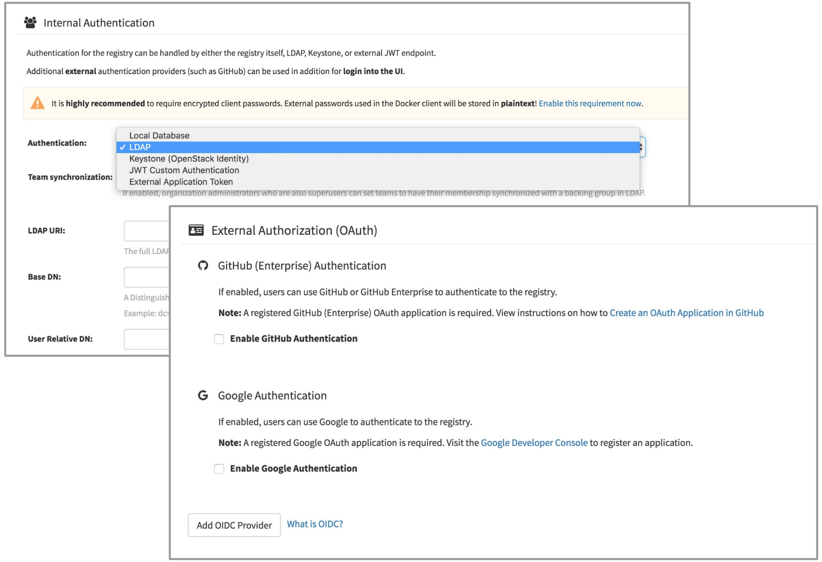
Enterprise Authorization and Authentication

Leverage existing identify mgt

Description: Red Hat Quay allows you to integrate your existing identity infrastructure and use a fine-grained permissions system to map to your organizational structure and grant access to whole teams to manage specific repositories.

Support auth providers:

- Built-in Database Auth
- LDAP auth and sync
- External OIDC provider
- OpenStack Keystone



Internal Authentication

Authentication for the registry can be handled by either the registry itself, LDAP, Keystone, or external JWT endpoint.

Additional **external** authentication providers (such as GitHub) can be used in addition for **login into the UI**.

It is **highly recommended** to require encrypted client passwords. External passwords used in the Docker client will be stored in **plaintext**! Enable this requirement now.

Authentication: Local Database / ✓ LDAP / Keystone (OpenStack Identity) / JWT Custom Authentication / External Application Token

Team synchronization:

If enabled, organization administrators who are also superusers can set teams to have their membership synchronized with a backing group in LDAP.

LDAP URI:

The full LDAP

Base DN:

A Distinguish

Example: dc=

User Relative DN:

External Authorization (OAuth)

GitHub (Enterprise) Authentication

If enabled, users can use GitHub or GitHub Enterprise to authenticate to the registry.

Note: A registered GitHub (Enterprise) OAuth application is required. View instructions on how to Create an OAuth Application in GitHub

☐ **Enable GitHub Authentication**

Google Authentication

If enabled, users can use Google to authenticate to the registry.

Note: A registered Google OAuth application is required. Visit the Google Developer Console to register an application.

☐ **Enable Google Authentication**

Note: Auth integration for OCP coming soon.

Clair

Clair Vulnerability Scanning

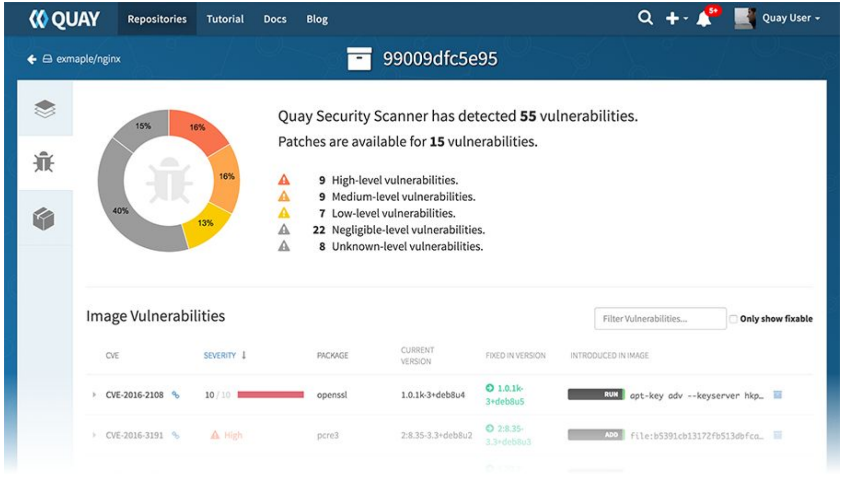
clair

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



Quay Security Scanner has detected **55** vulnerabilities.

Patches are available for **15** vulnerabilities.

- **9** High-level vulnerabilities.
- **9** Medium-level vulnerabilities.
- **7** Low-level vulnerabilities.
- **22** Negligible-level vulnerabilities.
- **8** Unknown-level vulnerabilities.

Image Vulnerabilities

Filter Vulnerabilities... □ Only show fixable

CVE	SEVERITY ↓	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN IMAGE
CVE-2016-2108	10 / 10	openssl	1.0.1k-3+deb8u4	1.0.1k-3+deb8u5	RUN apt-key adv --keyserver hkp…
CVE-2016-3191	High	pcre3	2:8.35-3.3+deb8u2	2:8.35-3.3+deb8u3	ADD file:b5391cb13172fb513dbfca…

How to setup Clair - Step 3

- Red Hat OVAL streams are configured by default
- Clair v2 limited to one namespace (RHEL/Alpine **or** pip but **not both**)
- Clair v3 will add support for other language level package managers (pip, npm, etc.) and additional namespaces (OS **and** languages)

Note: Users can't add additional data sources without programming!

Note: Due to OVAL usage same limitations apply to Clair as to openSCAP (RHEL Base Chan only)

Thank you !