OpenShift from a security perspective

Alfred Bach
Technical Partner Enablement Manager
abach@redhat.com

Red Hat

About me ..

Alfred Bach
Technical Partner Enablement Manager

Living in Austria near Vienna

4 Years with Red Hat
Coming from CA and SUSE/NOVELL

abach@redhat.com

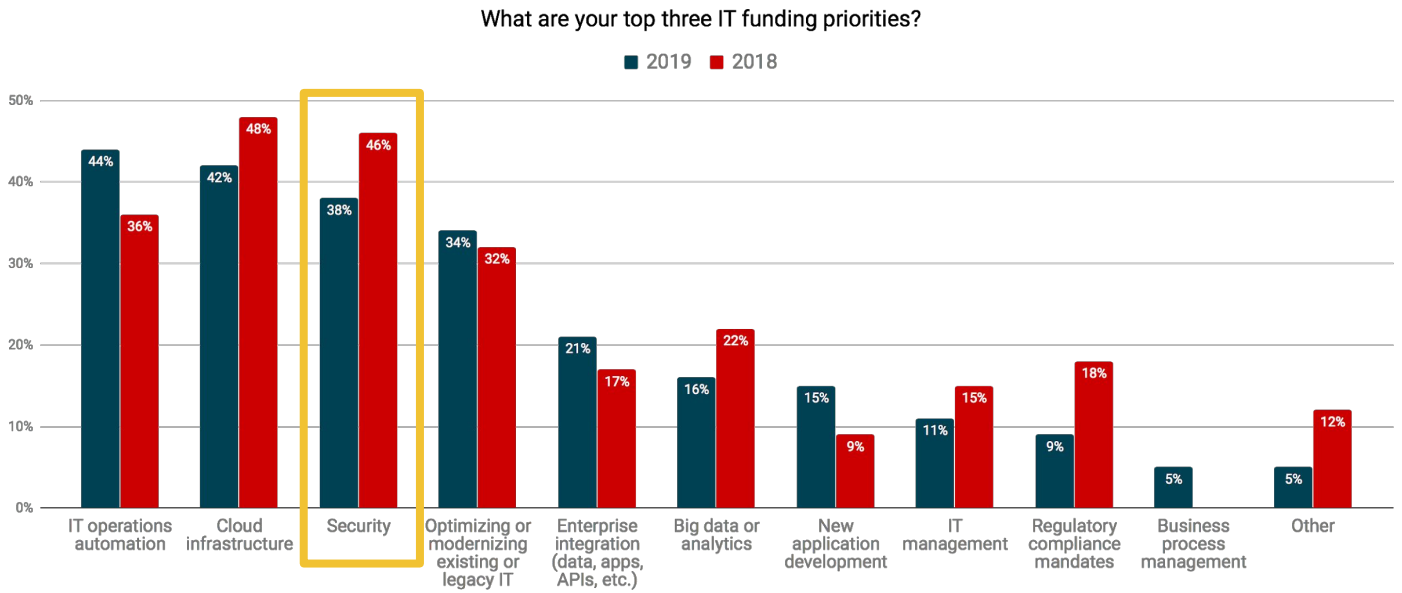
Red Hat

About you ?

https://www.menti.com/7ykxr3eaid

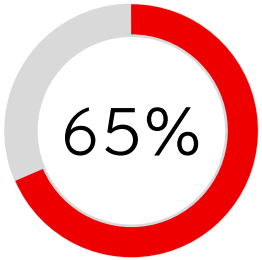
Red Hat

Top Funding Priorities for 2019: Automation, Cloud, & Security

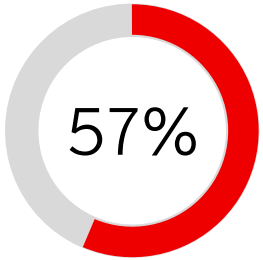


Source: Global IT Trends & Priorities Research, November 2018, Qualtrics and Red Hat

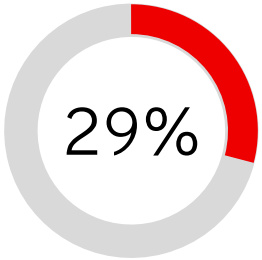
The Cyber Security Challenge is not Getting Easier



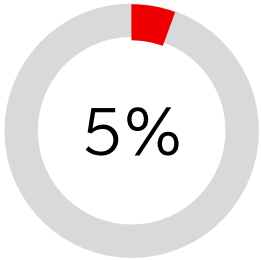
Reported increased Severity of attacks



Said the time to resolve an incident has grown



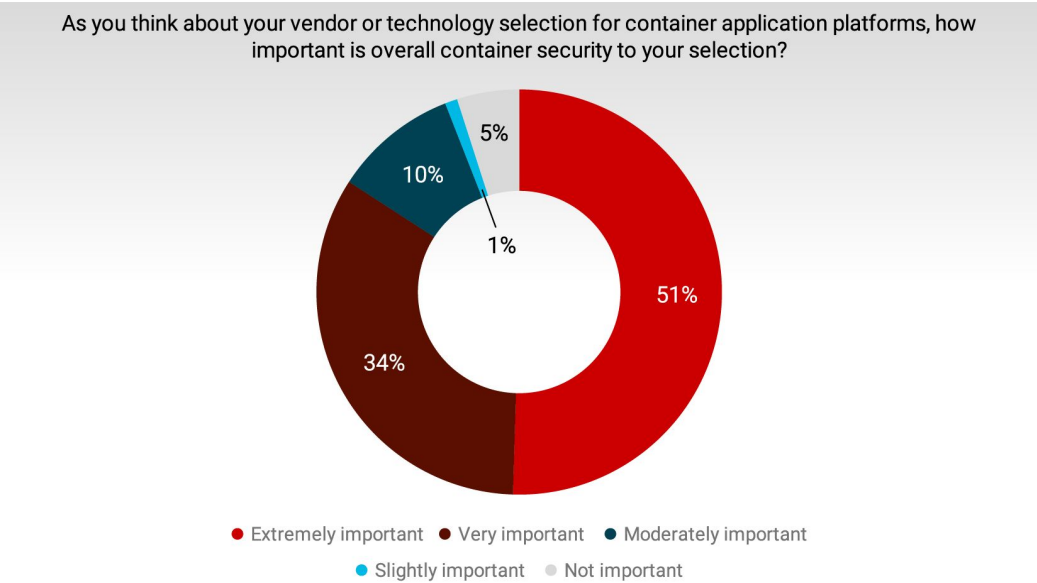
Have their ideal security-skilled staffing level, making it the #2 barrier to Cyber resilience



Portio of alerts coming in that the average security team examines every day

[1] The Third Annual Study on the Cyber Resilient Organization - Ponemon Institute, 2018 (Sponsored by IBM)
[2] https://venturebeat.com/2017/12/16/the-lesson-behind-2017s-biggest-enterprise-security-story/

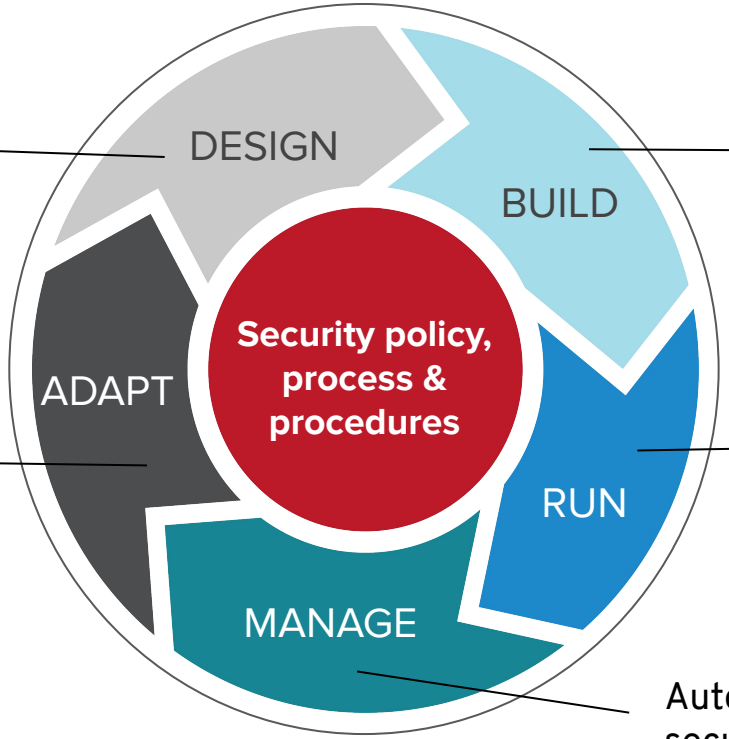
Choosing a Container Platform Vendor? Security is Critical



Source: Global IT Trends & Priorities Research, November 2018, Qualtrics and Red Hat

SECURITY MUST BE CONTINUOUS

And integrated throughout the IT lifecycle



Identify security requirements & governance models

Built-in from the start; not bolted-on

Deploy to trusted platforms with enhanced security capabilities

Automate systems for security & compliance

Revise, update, remediate as the landscape changes

Red Hat

COMPREHENSIVE CONTAINER SECURITY

CONTROL
Application Security

- Container Content
- CI/CD Pipeline
- Container Registry
- Deployment Policies

DEFEND
Infrastructure

- Container Platform
- Container Host Multi-tenancy
- Network Isolation
- Storage
- Audit & Logging
- API Management

EXTEND

- Security Ecosystem

Red Hat

Hardening tools & applicability guides

OpenShift 3

- NIST National Checklist for Red Hat OpenShift Container Platform 3
- FISMA Moderate
- ISO 27001
- PCI-DSS Reference Architecture

OpenShift Hardening Guide for 3.11

- Inspired by CIS Kubernetes benchmark v1.2

Targeted for 1H CY 2020

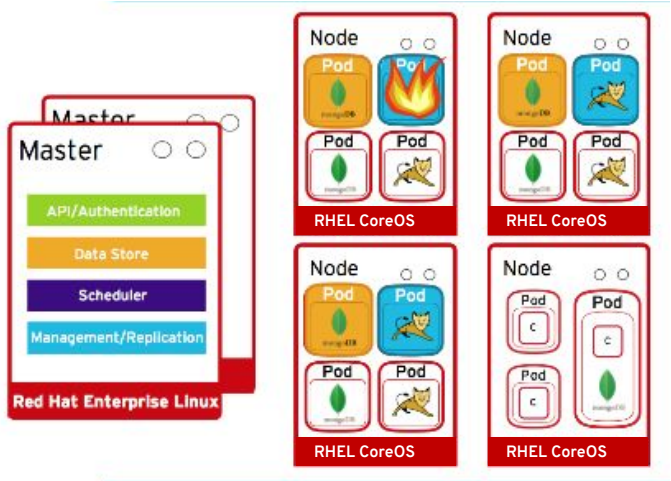
OpenShift 4

- Target Q1 CY 2020
  - HIPAA
  - HITRUST
  - OCP Hardening Guide for 4.3
- Target Q2 CY 2020
  - FISMA
  - ISO 27001
  - PCI-DSS

What Does It Take To Secure the Infrastructure?

OpenShift security features include

- Host & Runtime security
- Identity and Access Management
- Role-based Access Controls
- Project namespaces
- Network isolation
- Integrated & extensible secrets management
- Logging, Monitoring, Metrics, Audit



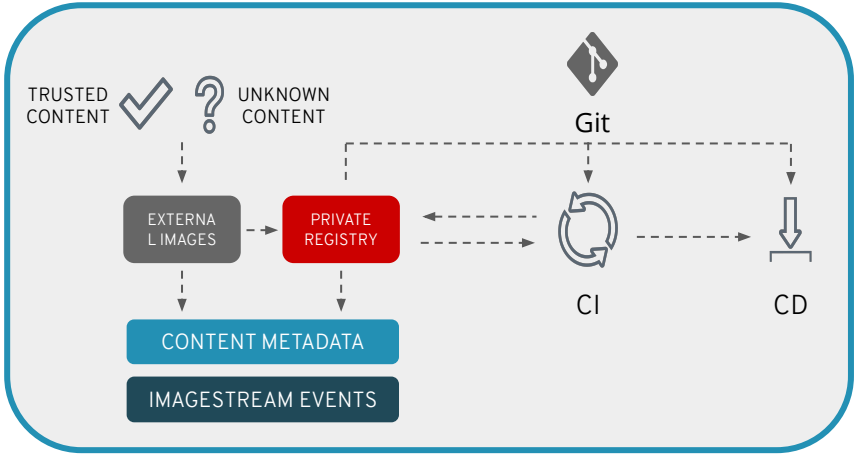
Red Hat

Securing Containerized Applications

An opportunity to shift security left

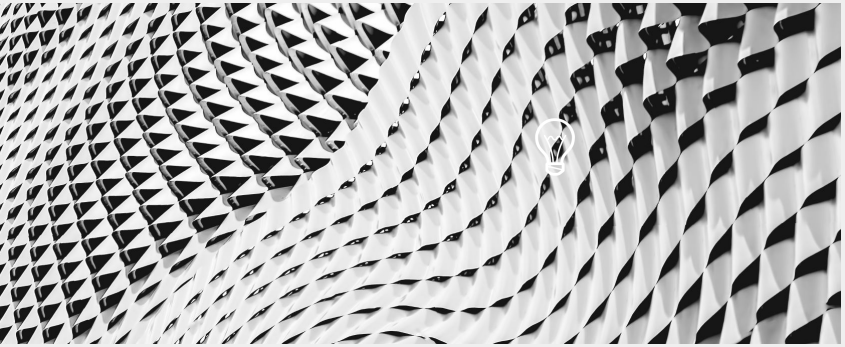
Best practices

- Use trusted sources for external content
- Use a private registry to manage images
- CI/CD must have security gates
- Application secrets management
- Apply runtime security policies
- Rebuild and redeploy - never patch a running container
- Ensure application logging, monitoring



Red Hat

DEFEND INFRASTRUCTURE

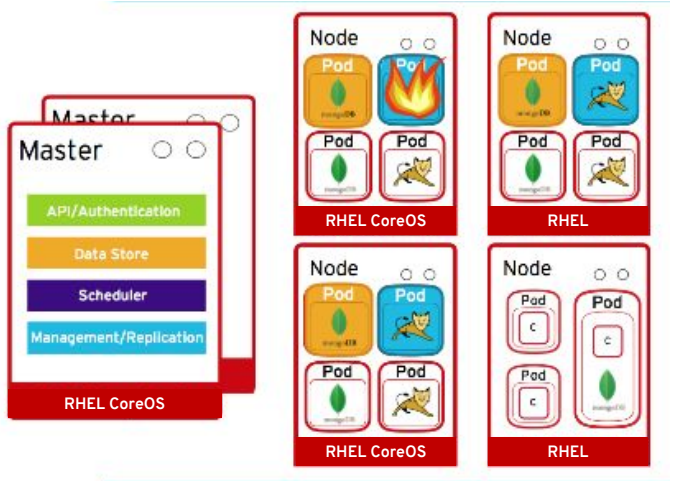


Red Hat

SECURING THE CONTAINER PLATFORM

Security Features Include

- Host & Runtime security
- Identity and Access Management
- Role-based Access Controls
- Project namespaces
- Integrated SDN - Network Policies is default
- Integrated & extensible secrets management
- Logging, Monitoring, Metrics

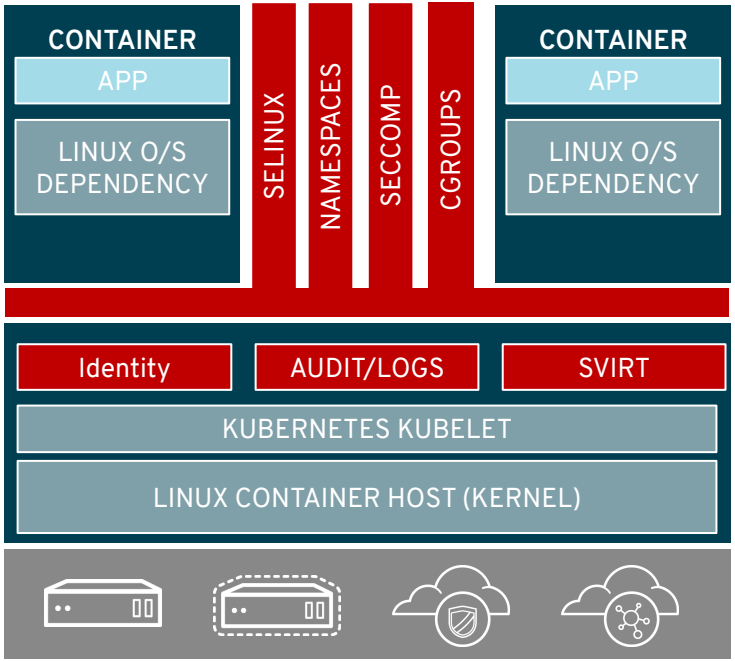


Red Hat

HOST OS CONTAINER MULTI-TENANCY

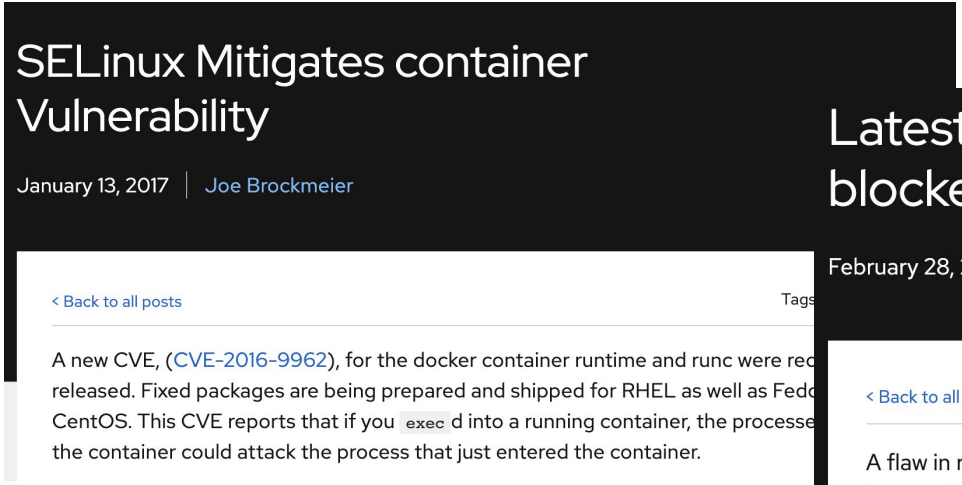
Container Security starts with Linux Security

- Security in the RHEL host applies to the container
- SELINUX and Kernel Namespaces are the one-two punch no one can beat
- Protects not only the host, but containers from each other
- RHEL CoreOS provides minimized attack surface



Red Hat

SELINUX MITIGATES CONTAINER RUNTIME VULNERABILITIES



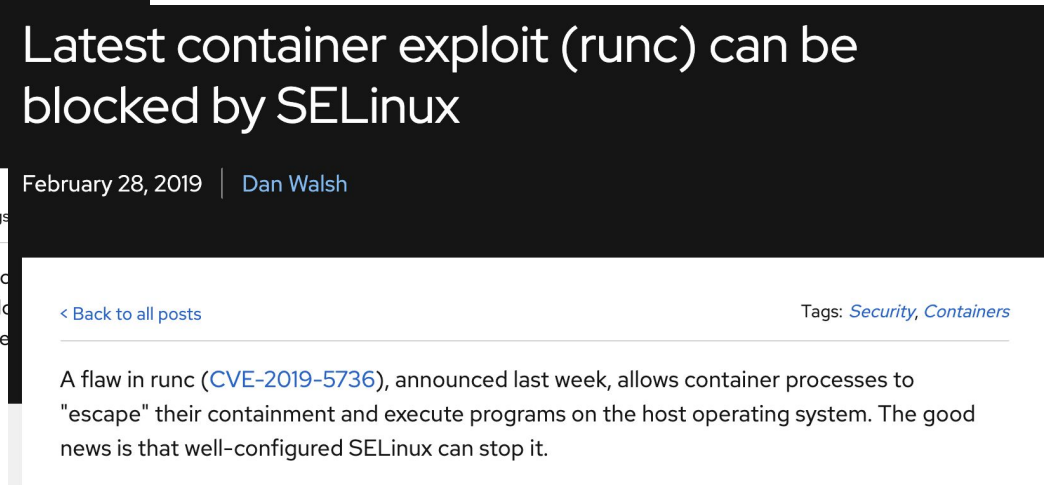
SELinux Mitigates container Vulnerability

January 13, 2017 | Joe Brockmeier

< Back to all posts Tags

A new CVE, (CVE-2016-9962), for the docker container runtime and runc were rec
released. Fixed packages are being prepared and shipped for RHEL as well as Fedo
CentOS. This CVE reports that if you `exec` d into a running container, the processe
the container could attack the process that just entered the container.

https://www.redhat.com/en/blog/selinux-mitigates-container-vulnerability



Latest container exploit (runc) can be blocked by SELinux

February 28, 2019 | Dan Walsh

< Back to all posts Tags: *Security*, *Containers*

A flaw in runc (CVE-2019-5736), announced last week, allows container processes to "escape" their containment and execute programs on the host operating system. The good news is that well-configured SELinux can stop it.

https://www.redhat.com/en/blog/latest-container-exploit-runc-can-be-blocked-selinux

Container Host Vision

An Ideal Container Host would be	RHEL CoreOS
Minimal	Only what's needed to run containers
Secure	Read-only & locked down
Immutable	Immutable image-based deployments & updates
Always up-to-date	OS updates are automated and transparent
Updates never break my apps	Isolates all applications as containers
Updates never break my cluster	OS components are compatible with the cluster
Supported on my infra of choice	Inherits majority of the RHEL ecosystem
Simple to configure	Installer generated configuration
Effortless to manage	Managed by Kubernetes Operators

IMMUTABLE OPERATING SYSTEM

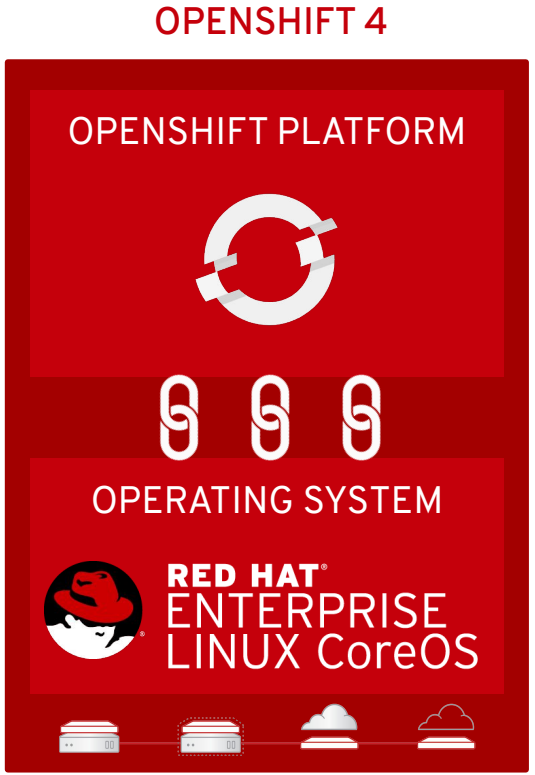
Red Hat Enterprise Linux CoreOS is versioned with OpenShift

CoreOS is tested and shipped in conjunction with the platform. Red Hat runs thousands of tests against these configurations.

Red Hat Enterprise Linux CoreOS is managed by the cluster

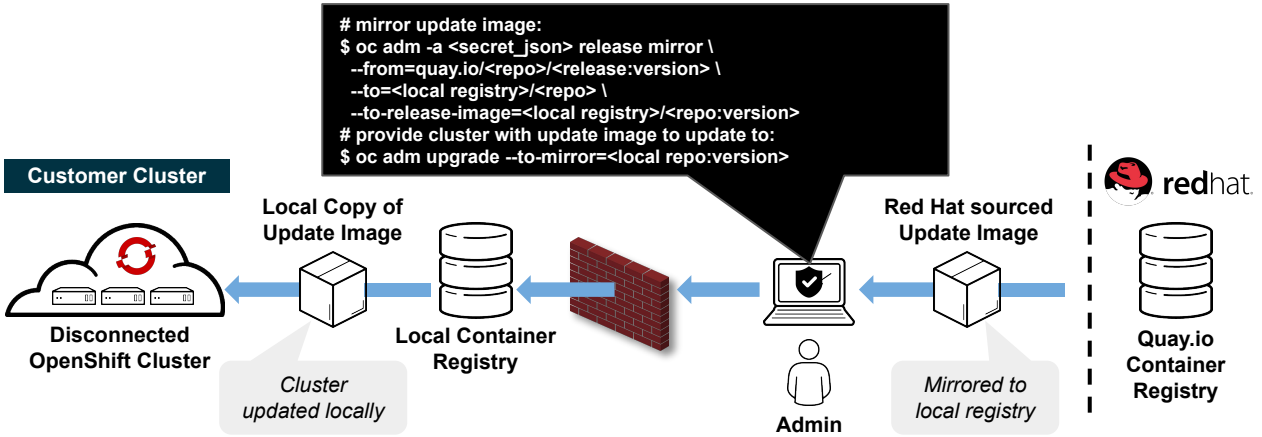
The Operating system is operated as part of the cluster, with the config for components managed by Machine Config Operator:

- CRI-O config
- Kubelet config
- Authorized registries
- SSH config



Red Hat

DISCONNECTED “AIR-GAPPED” INSTALL & UPGRADE



Overview

- 4.2 introduces support for installing and updating OpenShift clusters in disconnected environments
- Requires local Docker 2.2 spec compliant container registry to host OpenShift content
- Designed to work with the user provisioned infrastructure deployment method
  - *Note: Will not work with Installer provisioned infrastructure deployments*

Installation Procedure

- Mirror OpenShift content to local container registry in the disconnected environment
- Generate install-config.yaml: $ ./openshift-install create install-config --dir <dir>
  - Edit and add pull secret (PullSecret), CA certificate (AdditionalTrustBundle), and image content sources (ImageContentSources) to install-config.yaml
- Set the OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE environment variable during the creation of the ignition configs
- Generate the ignition configuration: $ ./openshift-install create ignition-configs --dir <dir>
- Use the resulting ignition files to bootstrap the cluster deployment

Generally Available

Red Hat

OpenShift 4 and Fips 140-2

FIPS ready Services

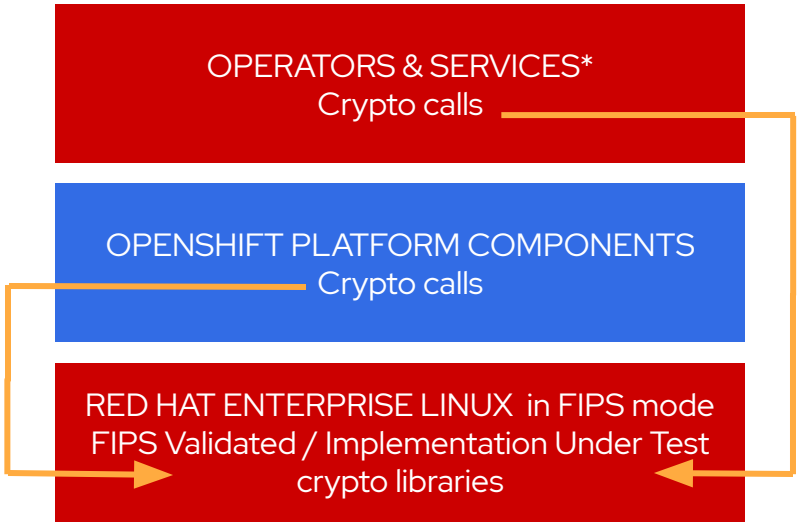
- When built with RHEL 7 base image

OpenShift calls FIPS validated crypto

- When running on RHEL 7.6 in FIPS mode, OpenShift components bypass go cryptographic routines and call into a RHEL FIPS 140-2 validated cryptographic library
- This feature is specific to binaries built with the RHEL go compiler and running on RHEL

RHEL CoreOS FIPS mode

- Configure at install to enforce use of FIPS Implementation Under Test* modules



*When built with RHEL base images

More about RHEL go and FIPS 140-2

RUNTIME SECURITY POLICIES

SCC (Security Context Constraints)

Allow administrators to control permissions for pods

Restricted SCC is granted to all users

By default, no containers can run as root

Admin can grant access to privileged SCC

Custom SCCs can be created

```
$ oc describe scc restricted
Name:                                   restricted
Priority:                               <none>
Access:
  Users:                                <none>
  Groups:                               system:authenticated
Settings:
  Allow Privileged:                     false
  Default Add Capabilities:             <none>
  Required Drop Capabilities:           KILL,MKNOD,SYS_CHROOT,SETUID,SETGID
  Allowed Capabilities:                 <none>
  Allowed Seccomp Profiles:             <none>
  Allowed Volume Types:                 configMap,downwardAPI,emptyDir,persistentVolumeClaim,projected,
  Allow Host Network:                   false
  Allow Host Ports:                     false
  Allow Host PID:                       false
  Allow Host IPC:                       false
  Read Only Root Filesystem:            false
  Run As User Strategy: MustRunAsRange
```

IDENTITY AND ACCESS MANAGEMENT

OpenShift includes an OAuth server, which does three things:

- Identifies the person requesting a token, using a configured identity provider
- Determines a mapping from that identity to an OpenShift user
- Issues an OAuth access token which authenticates that user to the API

Managing Users and Groups in OpenShift
Configuring Identity Providers

Supported Identity Providers include

- Keystone
- LDAP
- GitHub
- GitLab
- GitHub Enterprise (new with 3.11)
- Google
- OpenID Connect
- Security Support Provider Interface (SSPI) to support SSO flows on Windows (Kerberos)

RESTRICT ACCESS BY NEED TO KNOW

Role based authorization

- Project scope & cluster scope available
- Matches request attributes (verb,object,etc)
- If no roles match, request is denied (deny by default)
- Operator- and user-level roles are defined by default
- Custom roles are supported

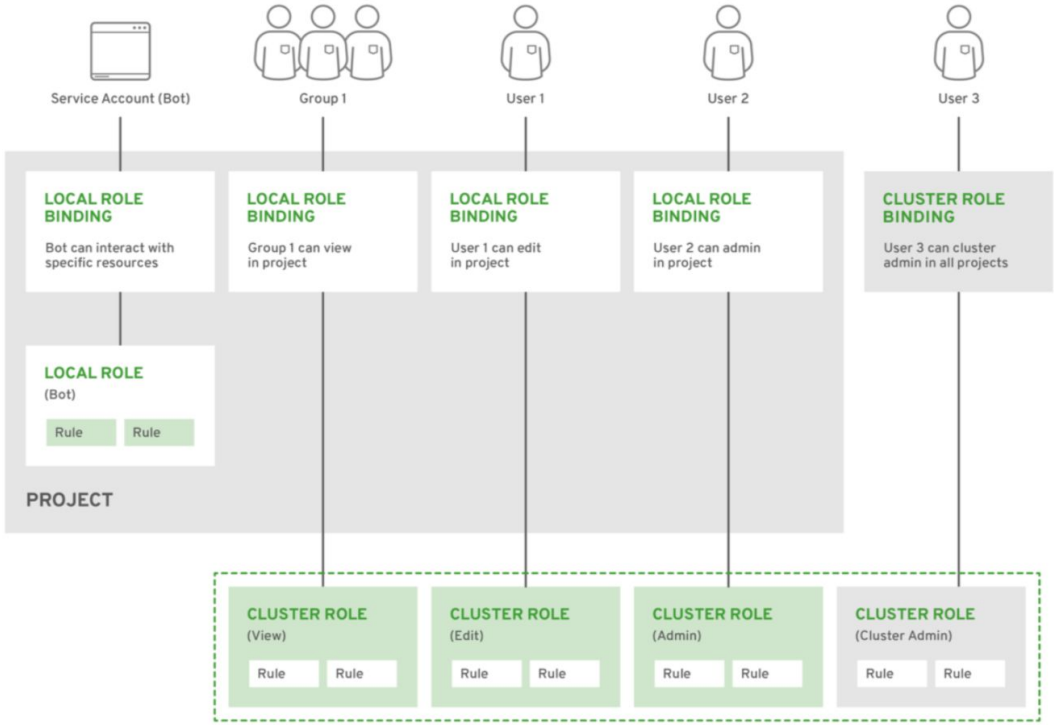


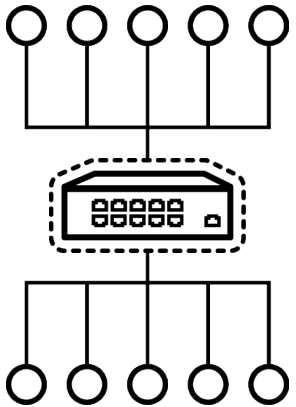
Figure 12 - Authorization Relationships

For more information see: [Managing RBAC in OpenShift]

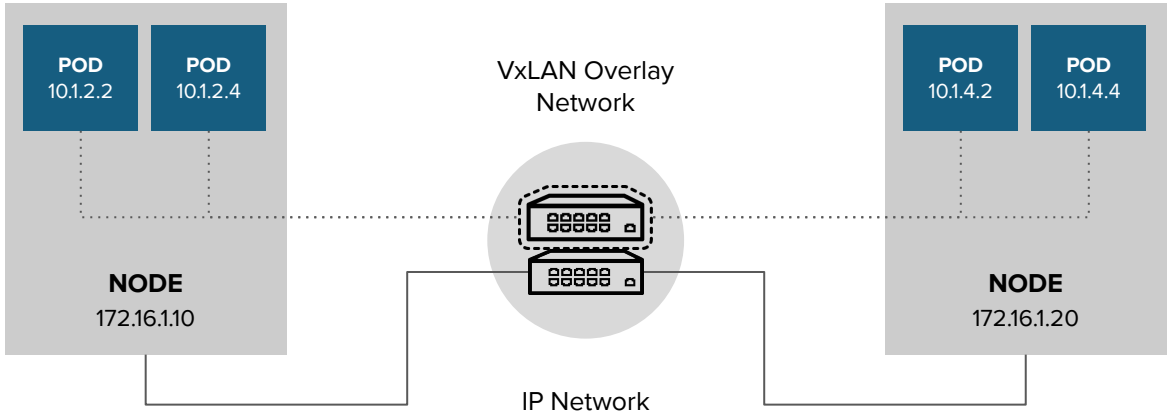
Network security

OPENSHIFT NETWORKING

- Built-in internal DNS to reach services by name
- Software Defined Networking (SDN) for a unified cluster network to enable pod-to-pod communication
- OpenShift follows the Kubernetes Container Networking Interface (CNI) plug-in model
- Isolate applications from other applications within a cluster
- Isolate environments (Dev / Test / Prod) from other environments within a cluster

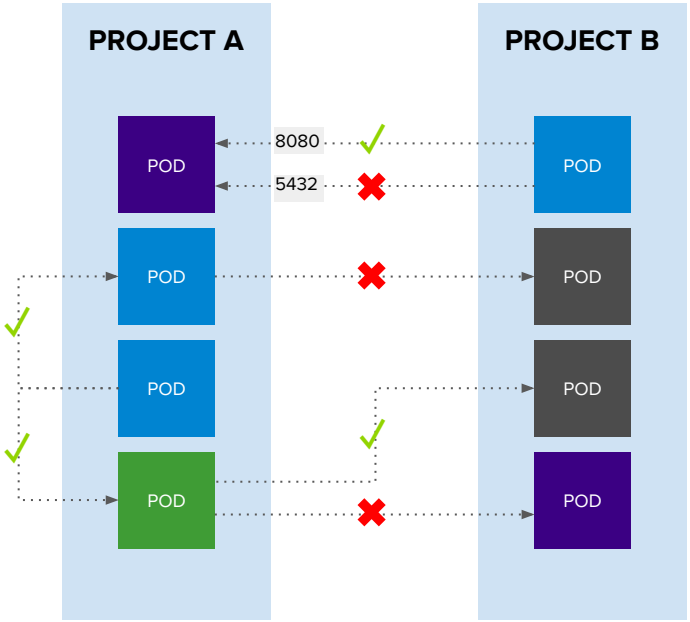


OPENSHIFT POD to POD NETWORKING



OPENSHIFT SDN

Network Policy enabled by default in OpenShift 4



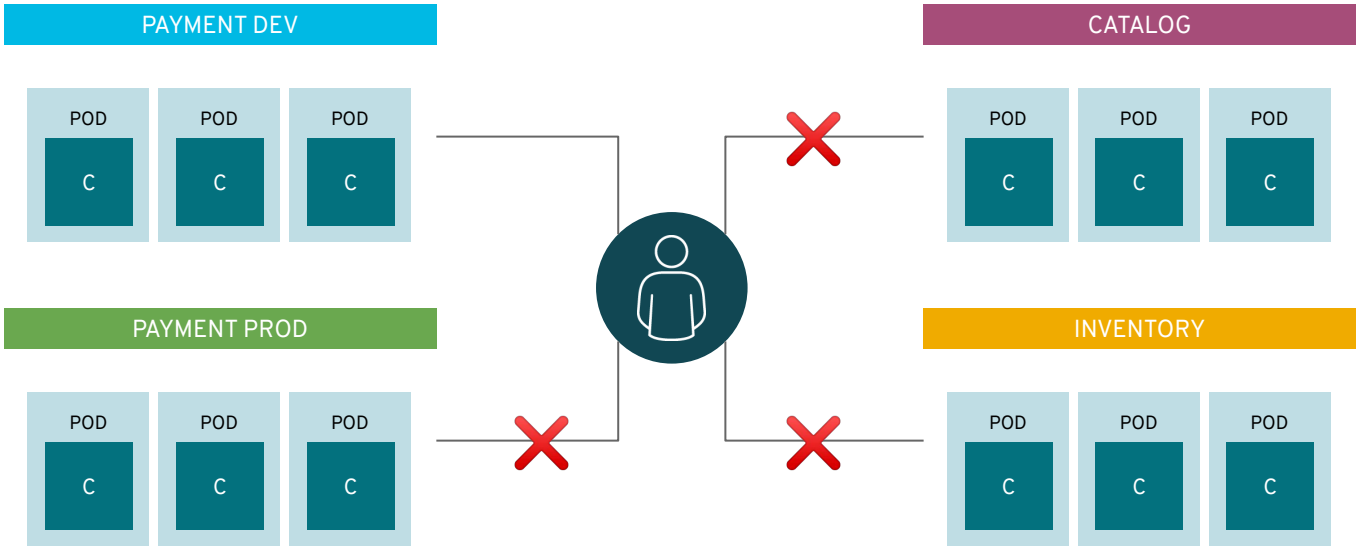
Example Policies

- Allow all traffic inside the project
- Allow traffic from green to gray
- Allow traffic to purple on 8080

```
apiVersion: extensions/v1beta1
kind: NetworkPolicy
metadata:
  name: allow-to-purple-on-8080
spec:
  podSelector:
    matchLabels:
      color: purple
  ingress:
  - ports:
    - protocol: tcp
      port: 8080
```

PROJECTS ISOLATE APPLICATIONS

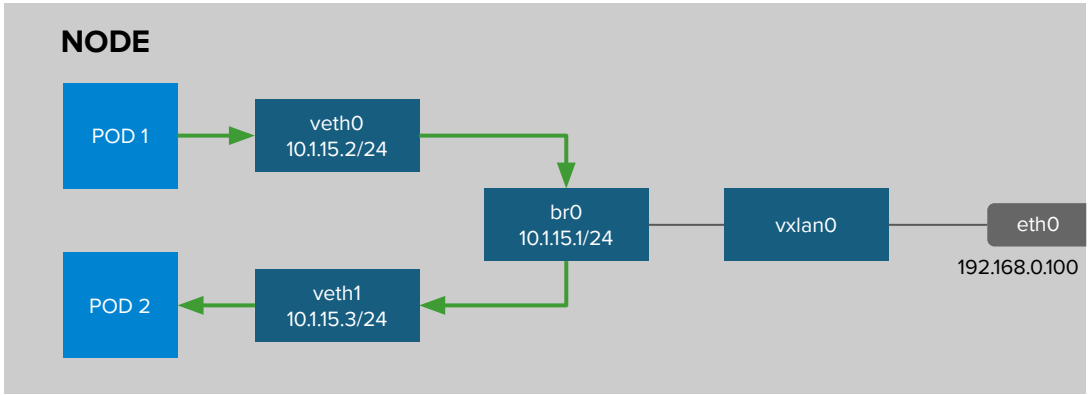
across teams, groups and departments



Red Hat

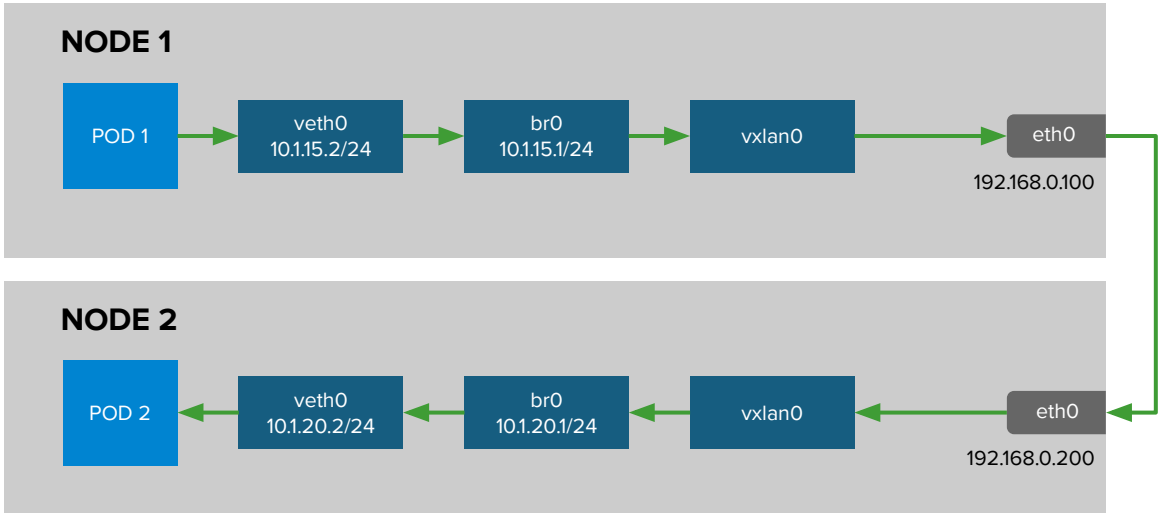
OPENSHIFT SDN - OVS PACKET FLOW

Container to Container on the Same Host



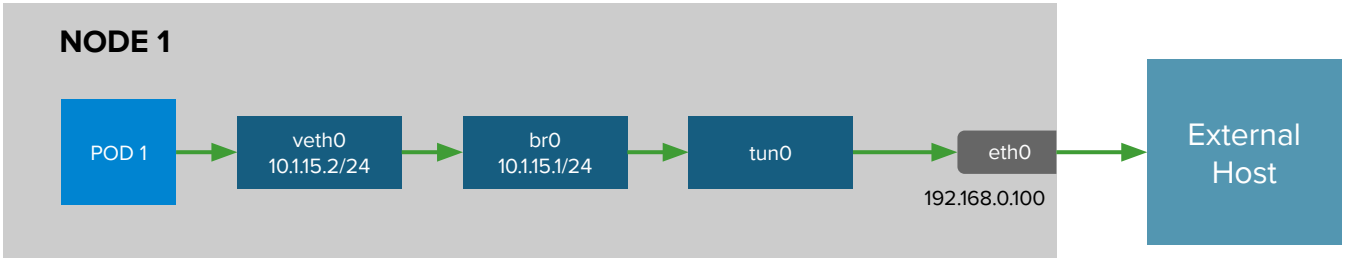
OPENSHIFT SDN - OVS PACKET FLOW

Container to Container on the Different Hosts



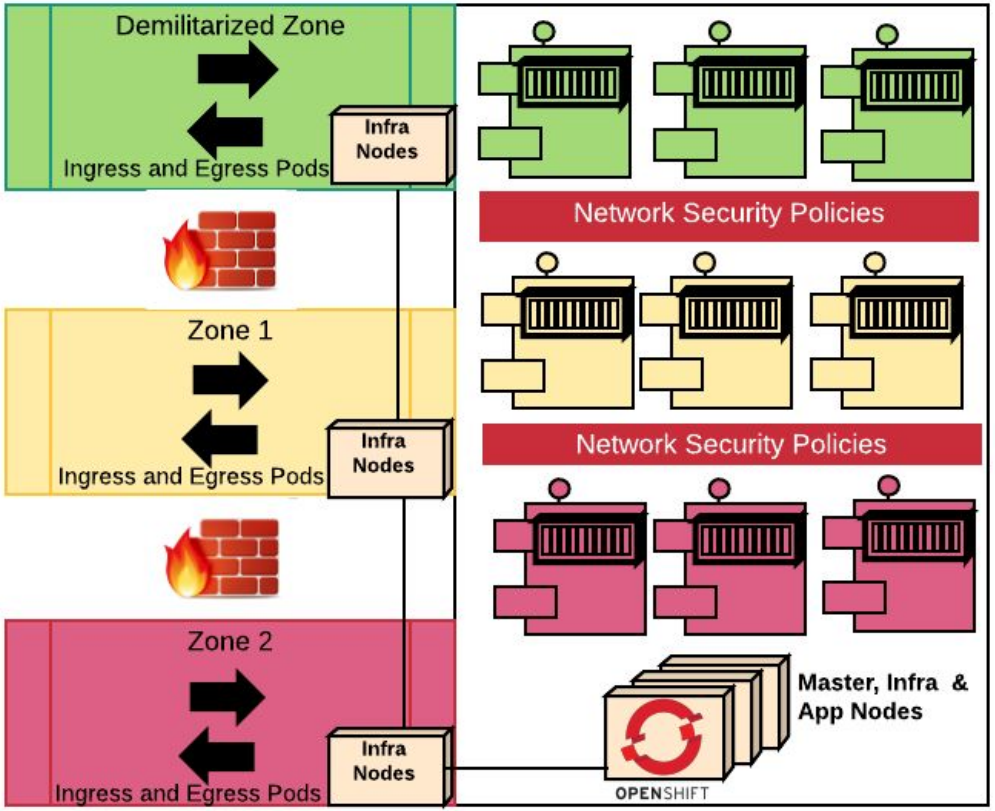
OPENSHIFT SDN - OVS PACKET FLOW

Container Connects to External Host



OpenShift cluster with multiple zones

Using multiple ingress controllers, network policies, multiple egress pods



Application pods run on one OpenShift Cluster.

Microsegmented with Network Security policies.

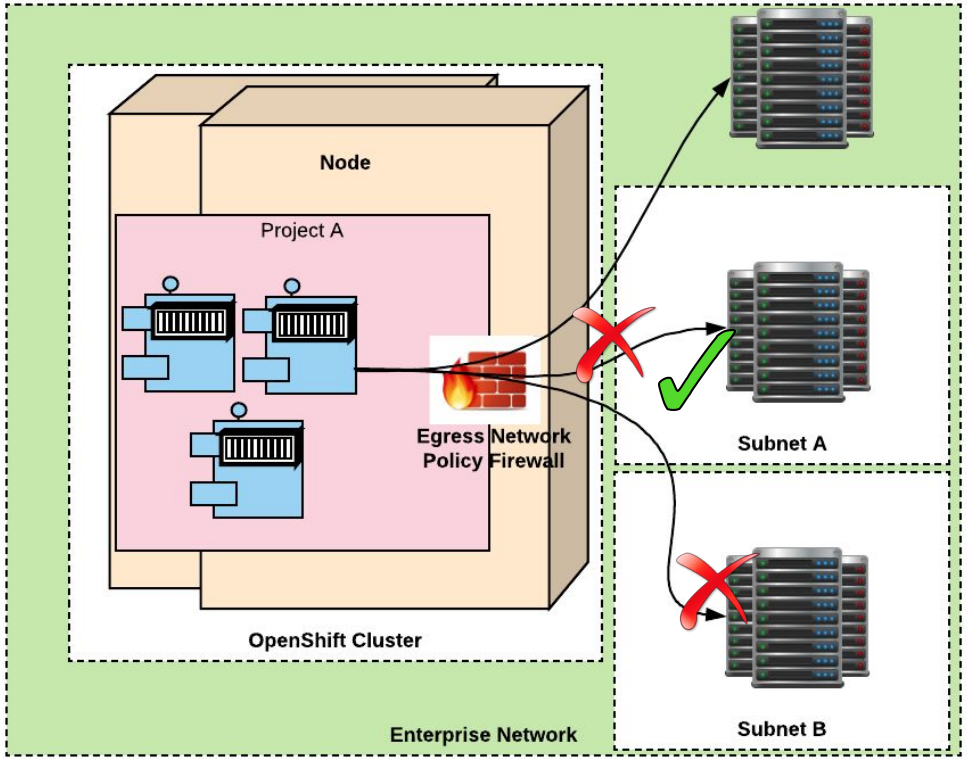
Infra Nodes in each zone run Ingress and Egress pods for specific zones.

If required, physical isolation of pods to specific nodes is possible with node-selectors. But that can reduce worker node density.

Red Hat

EGRESS FIREWALL TO LIMIT ACCESS

Cluster admin can limit the external addresses accessed by some or all pods



Examples:

A pod can talk to hosts (outside OpenShift cluster) but cannot connect to public internet

A pod can talk to public internet, but cannot connect to hosts (outside OpenShift cluster)

Public Internet

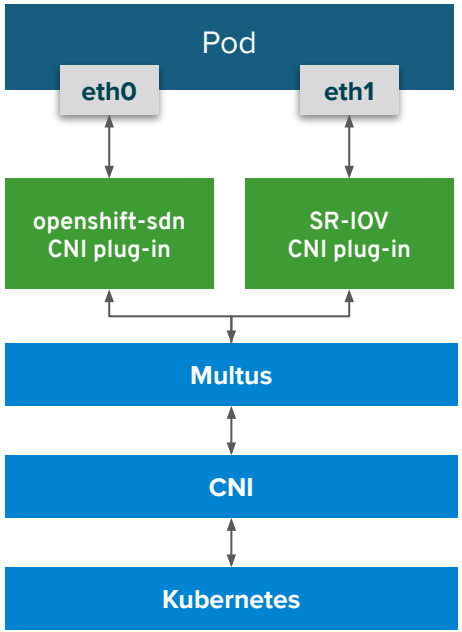
A pod cannot reach specific subnets/hosts

Red Hat

Multus

- Problem: Kubernetes only supports one network interface, “eth0”, but we need:
  - Functional separation of control/data planes
  - Link aggregation for network redundancy
  - Different network protocol stacks, capabilities, SLAs
  - Traffic isolation / Network segregation and security
  - QoS
- Solution: Multus “meta plug-in” for Kubernetes CNI
- Enables multiple network interfaces per pod, each assigned a different CNI plug-in defined in pod spec
  - Each with its configuration defined in CRD objects
- SR-IOV enablement

Pod with Multus

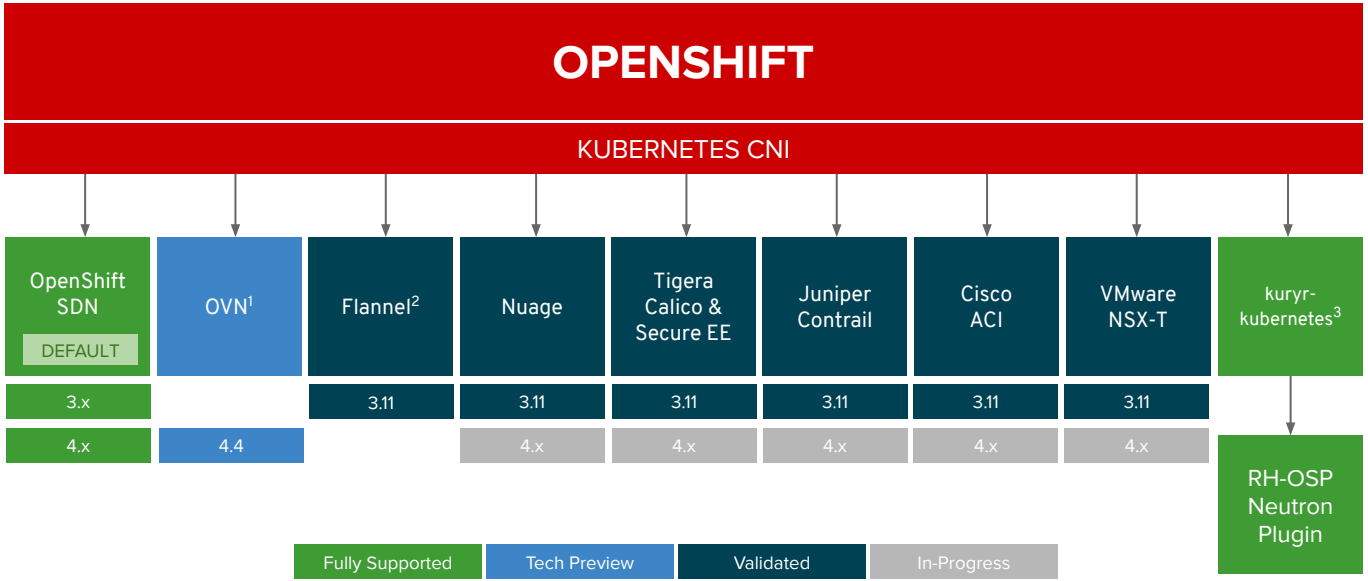


BUILDING UPON MULTUS

Development is now unblocked for a number of customer use cases

- Functional separation of control/data planes
  - Default SDN on primary interface for management traffic
  - Data-plane traffic on secondary (or tertiary…) interface
- Link aggregation for network redundancy
- Different network protocol stacks, capabilities, SLAs
- Network traffic isolation, segregation and security
- User-space networking (e.g. VPP, OVS-DPDK)
- VLAN, MACVLAN, host device networking
- Video Streaming (Multicast)
- High Performance Networking - e.g. SR-IOV
  - Openshift-SDN/OVN-Kubernetes as primary interface for management traffic
  - SR-IOV Device Plugin for VF hardware resource scheduling
  - SR-IOV CNI plugs VFs assigned by device plugin into pods as data plane interfaces
- QoS
- CNV (kubevirt) Networking
- SDN Migration (e.g. our next-gen SDN)

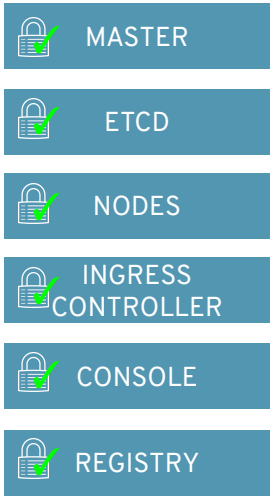
OPENSHIFT NETWORK PLUGINS



[1]Targeting GA at OCP 4.3 (not default SDN)
[2]Flannel is minimally verified and is supported only and exactly as deployed in the OpenShift on OpenStack reference architecture
[3]Available as an install-time option at 3.11.119 and 4.2.z (targeting 4.2.2)

CERTIFICATE MANAGEMENT

- Certificates are used to provide secure connections to
  - master and nodes
  - Ingress controller and registry
  - etcd
- Certificate rotation is automated
- Optionally configure external endpoints to use custom certificates
- For example:
  [Requesting and Installing Let's Encrypt Certificates for OpenShift 4]

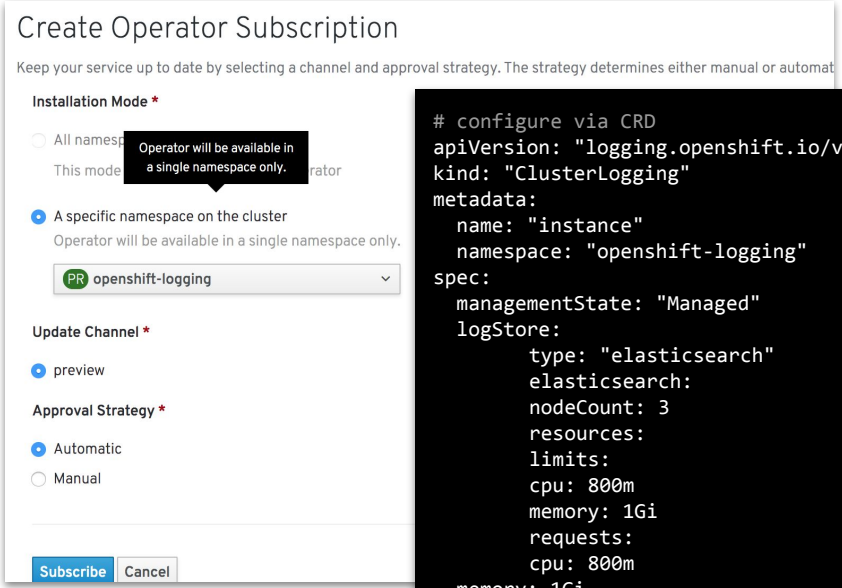


Red Hat

CLUSTER LOG MANAGEMENT

Install the Elasticsearch and Cluster Logging Operators from OperatorHub

- EFK stack aggregates logs for hosts and applications
  - Elasticsearch: a search and analytics engine to store logs
  - Fluentd: gathers logs and sends to Elasticsearch.
  - Kibana: A web UI for Elasticsearch.
- Access control
  - Cluster administrators can view all logs
  - Users can only view logs for their projects
  - Central Audit policy configuration
- Ability to send logs elsewhere
  - External elasticsearch, Splunk, etc

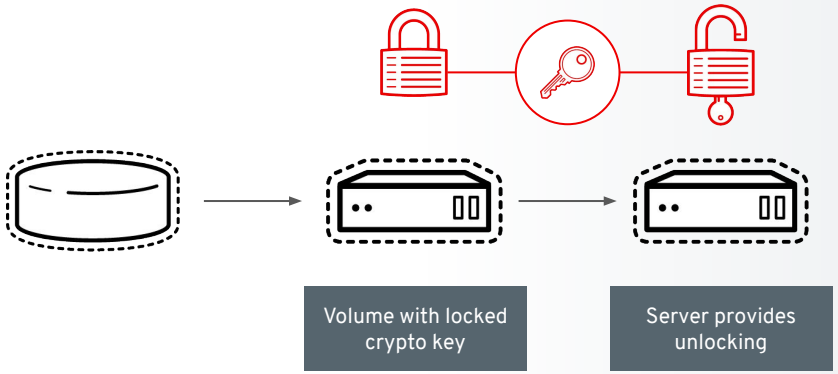



```
# configure via CRD
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  logStore:
        type: "elasticsearch"
        elasticsearch:
        nodeCount: 3
        resources:
        limits:
        cpu: 800m
        memory: 1Gi
        requests:
        cpu: 800m
  memory: 1Gi
  storage:
        storageClassName: gp2
        size: 100G
        redundancyPolicy: "SingleRedunda
  visualization:
        type: "kibana"
        kibana:
        replicas: 1
  curation:
```

Attached storage

Secure storage by using

- SELinux access controls
- Secure mounts
- Supplemental group IDs for shared storage
- Network bound disk encryption

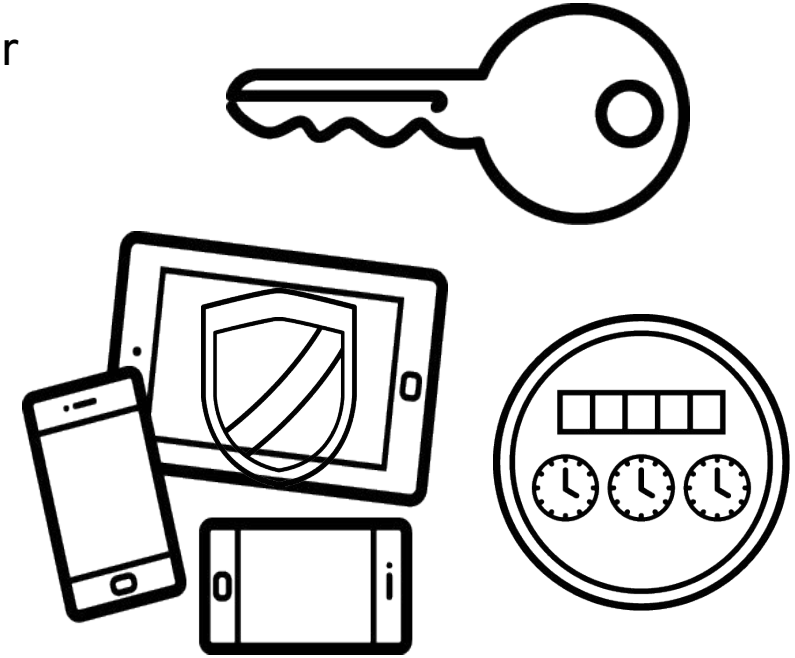


Red Hat

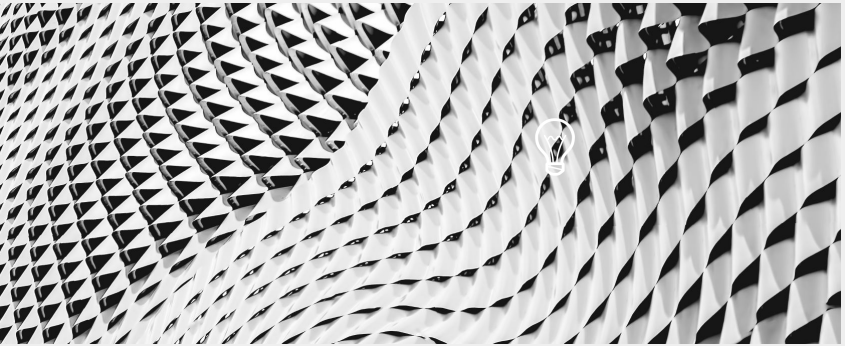
APPLICATION API MANAGEMENT

Consider configuring an API gateway for container platform & application APIs

- Authentication and authorization
- LDAP integration
- End-point access controls
- Rate limiting



CONTROL APPLICATION SECURITY



Red Hat

DEVSECOPS

THROUGH THE ADOPTION OF CONTAINERS

We created Dev and Ops and Security user stories and tackled them together.

DEVELOPER

I can break builds if security and compliance rules aren't followed…

SECURITY

We're empowering the developers and ideally empowering them straight to production.

OPERATIONS

[DevNation Federal - The Journey to DevSecOps]

Next generation container tools

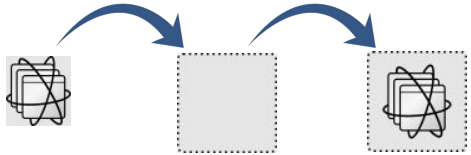
NEXT-GEN CONTAINER TOOLS

Providing stability, flexibility and performance with containers and images

Container-tools - OCI tooling to create, run, and manage, Linux Containers with an enterprise life cycle.

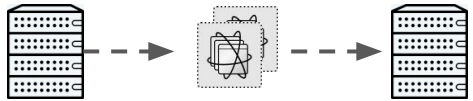
- Conform to the OCI image and runtime specifications
- Daemon-less, OS-native container tooling
- Separation of concerns

buildah



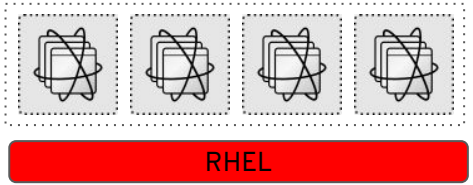
Build OCI/docker Images

skopeo



Inspect, copy, & sign Images

podman



run, manage, debug containers

cri-o

A lightweight, OCI-compliant container runtime

Optimized for Kubernetes

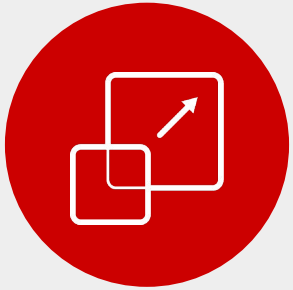
Any OCI-compliant container from any OCI registry (including docker)

Improve Security and Performance at scale

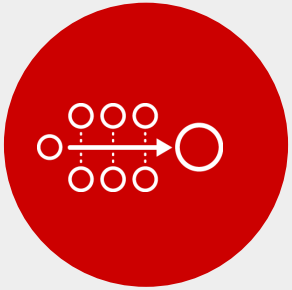
[CRI - the Container Runtime Interface]
[OpenShift 4 defaults to CRI-O]
[Red Hat contributes CRI-O to the Cloud Native Computing Foundation]

OPENSHIFT LOVES CI/CD

JENKINS-AS-A SERVICE
ON OPENSHIFT



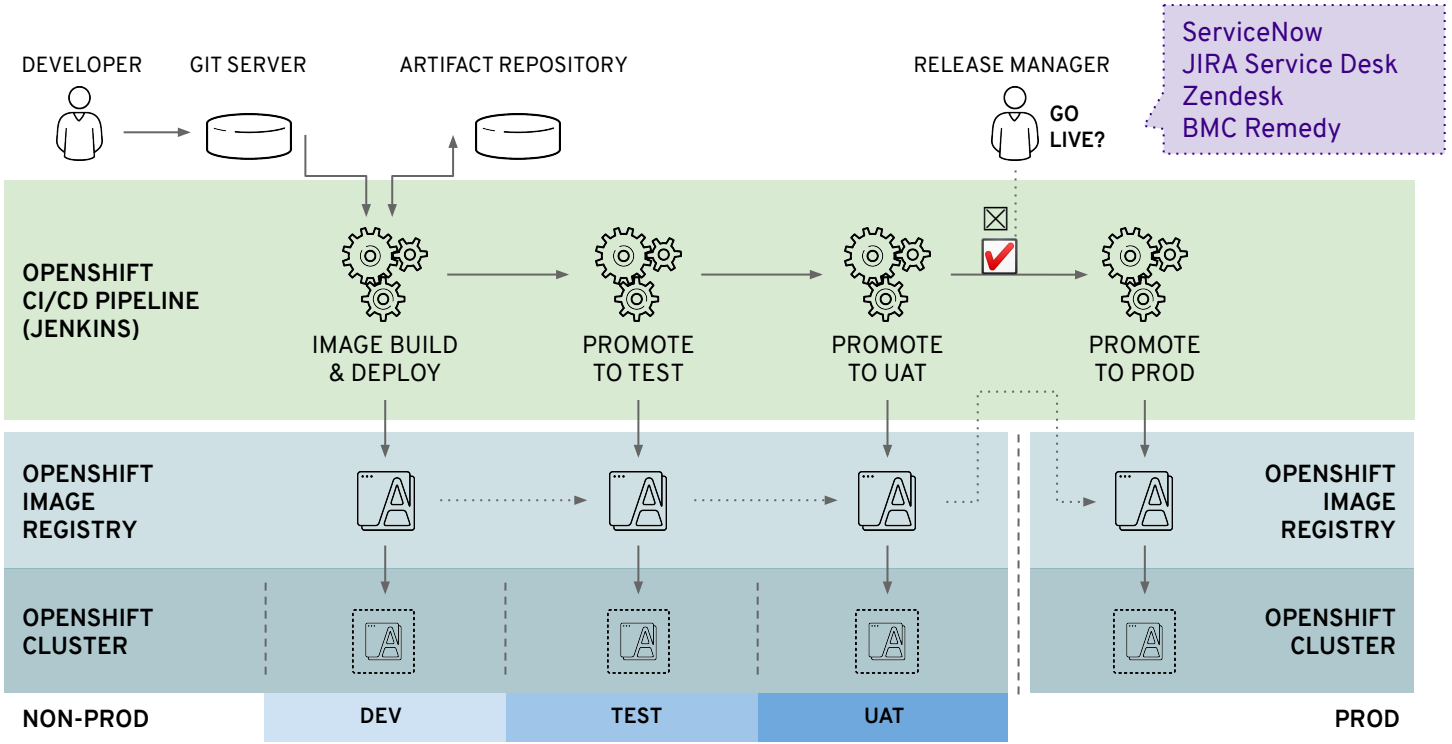
HYBRID JENKINS INFRA
WITH OPENSHIFT



EXISTING CI/CD
DEPLOY TO OPENSHIFT

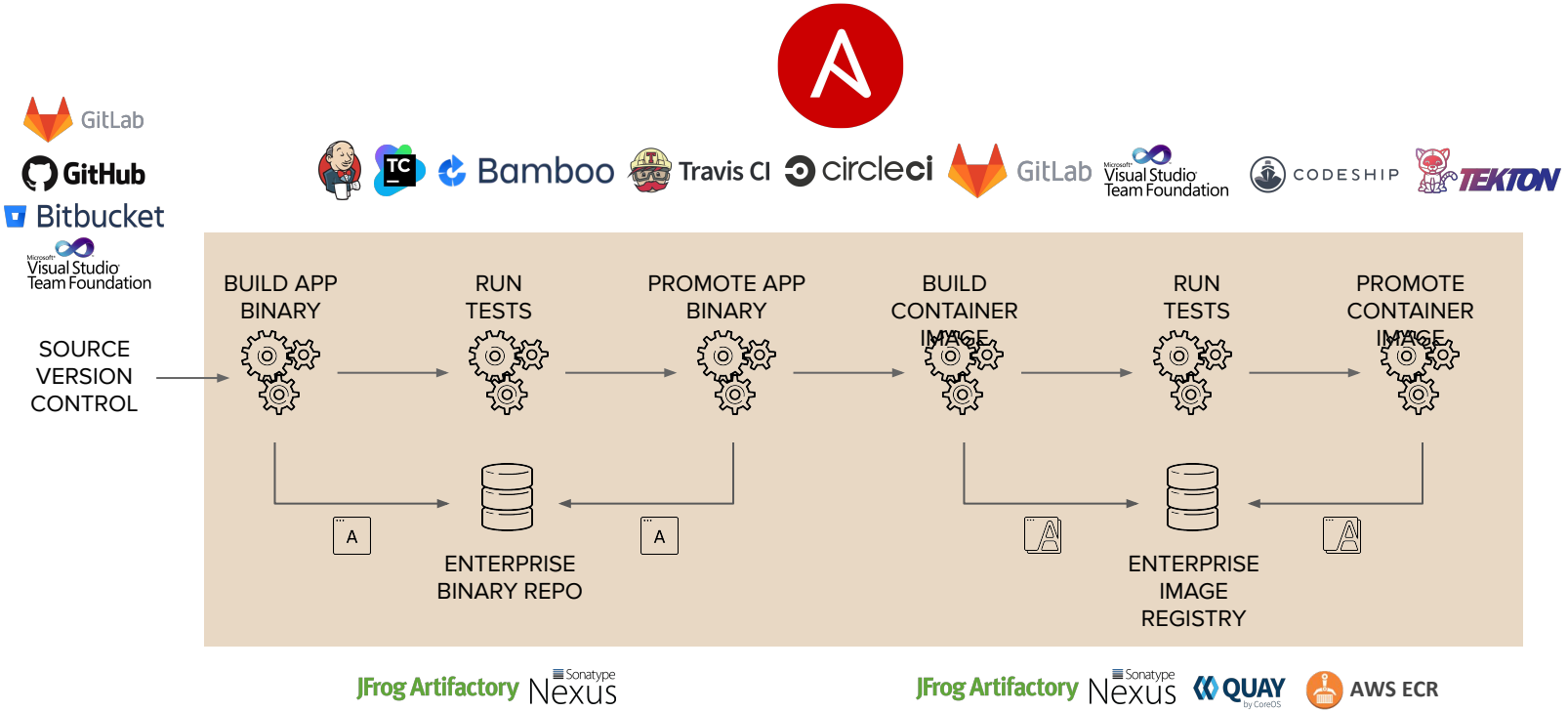
Red Hat

USE THE OPENSHIFT PIPELINE



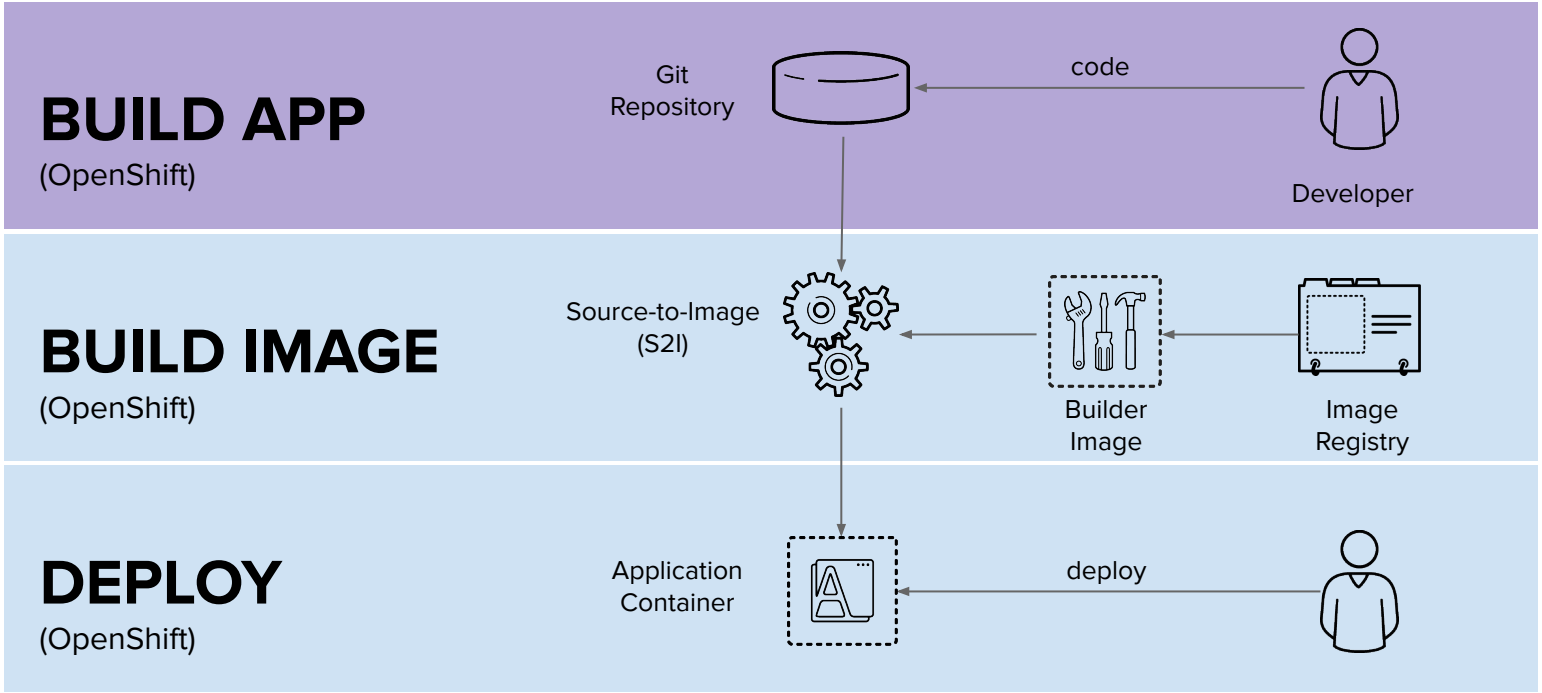
Red Hat

OR USE EXISTING DELIVERY PROCESSES



Red Hat

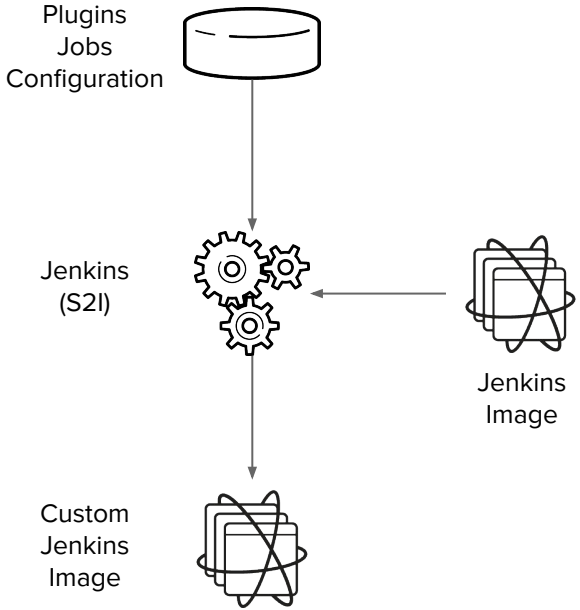
DEPLOY IMAGES, APPLICATION BINARIES OR SOURCE CODE WITH OPENSHIFT



Red Hat

JENKINS-AS-A-SERVICE ON OPENSHIFT

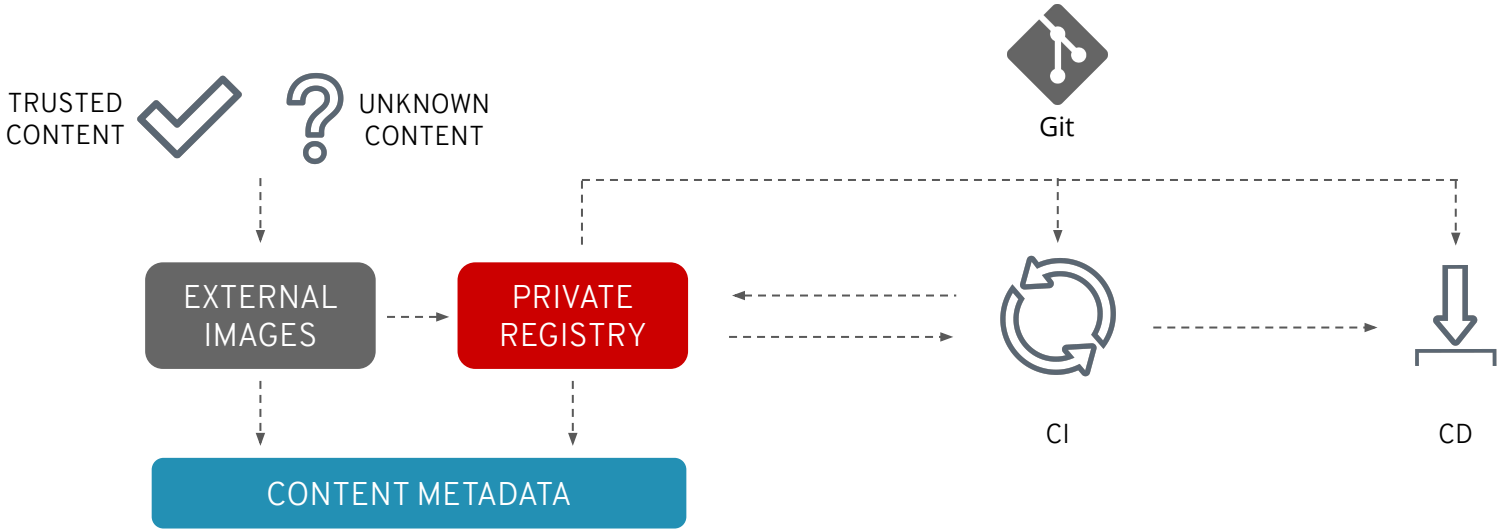
- Certified Jenkins images with pre-configured plugins
  - Provided out-of-the-box
  - Follows Jenkins 1.x and 2.x LTS versions
- Jenkins S2I Builder for customizing the image
  - Install Plugins
  - Configure Jenkins
  - Configure Build Jobs
- OpenShift plugins to integrate authentication with OpenShift and also CI/CD pipelines
- Dynamically deploys Jenkins slave containers



Red Hat

SECURE & AUTOMATE THE CONTENT LIFECYCLE

Elements of the Openshift container pipeline

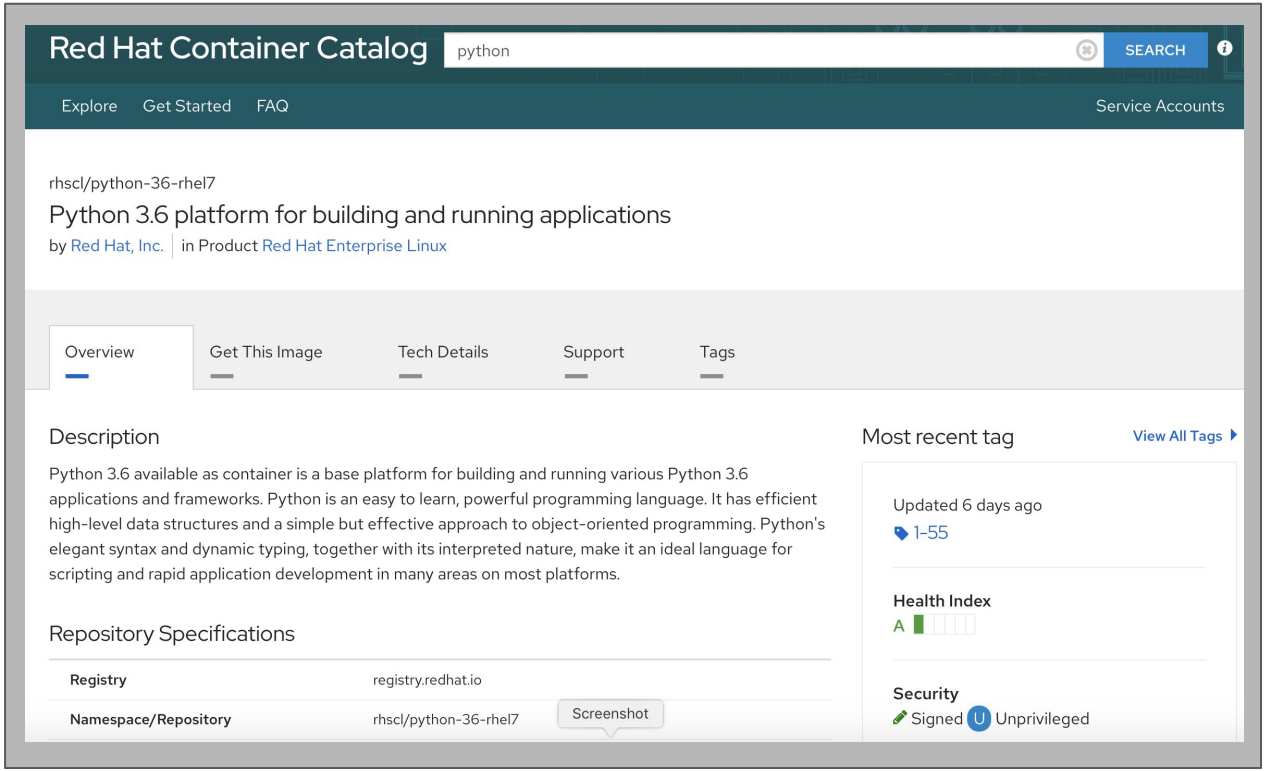


Red Hat

EXTERNAL CONTENT: USE TRUSTED SOURCES

Red Hat Container Images

- Signed Images
- Health Index (A to F grade)*
- Security advisories & errata (patches)

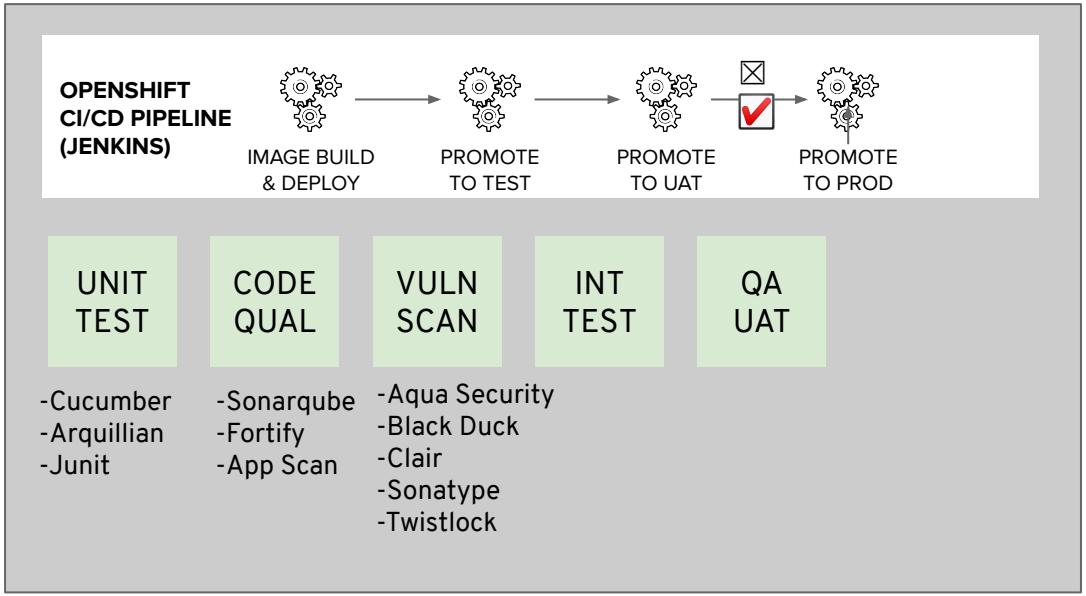


*More about the Health Index

Red Hat

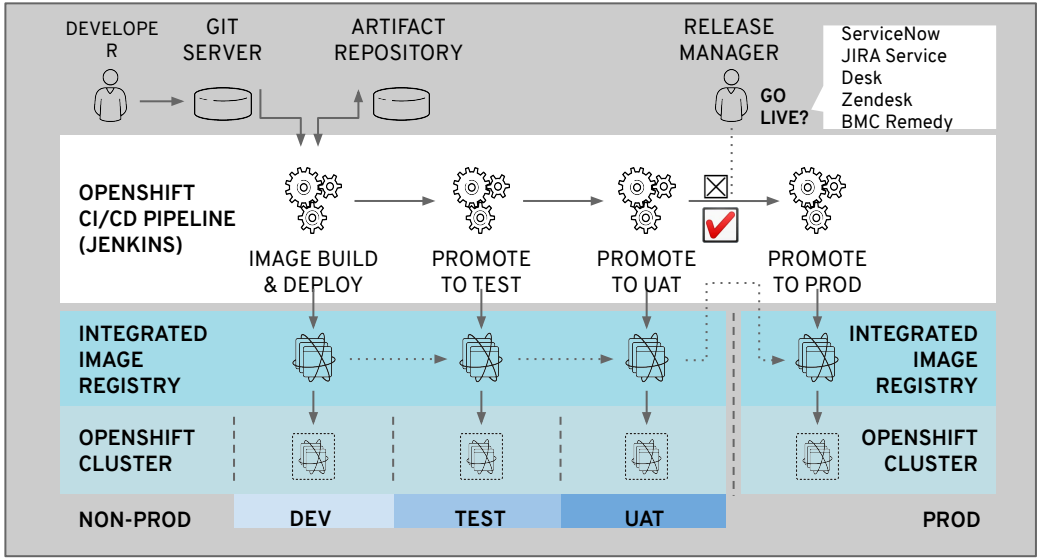
CI/CD MUST INCLUDE SECURITY GATES

- Integrate security testing into your build / CI process
- Use automated policies to flag builds with issues
- Sign your custom container images



MANAGING CONTAINER DEPLOYMENT

- Deployments: Containerized App Configuration as Code
- Whitelist / Blacklist external repos
- Apply runtime security policies
- Validate image signatures
- Monitor for new vulnerabilities
- Trust is temporal:
  rebuild & redeploy as needed



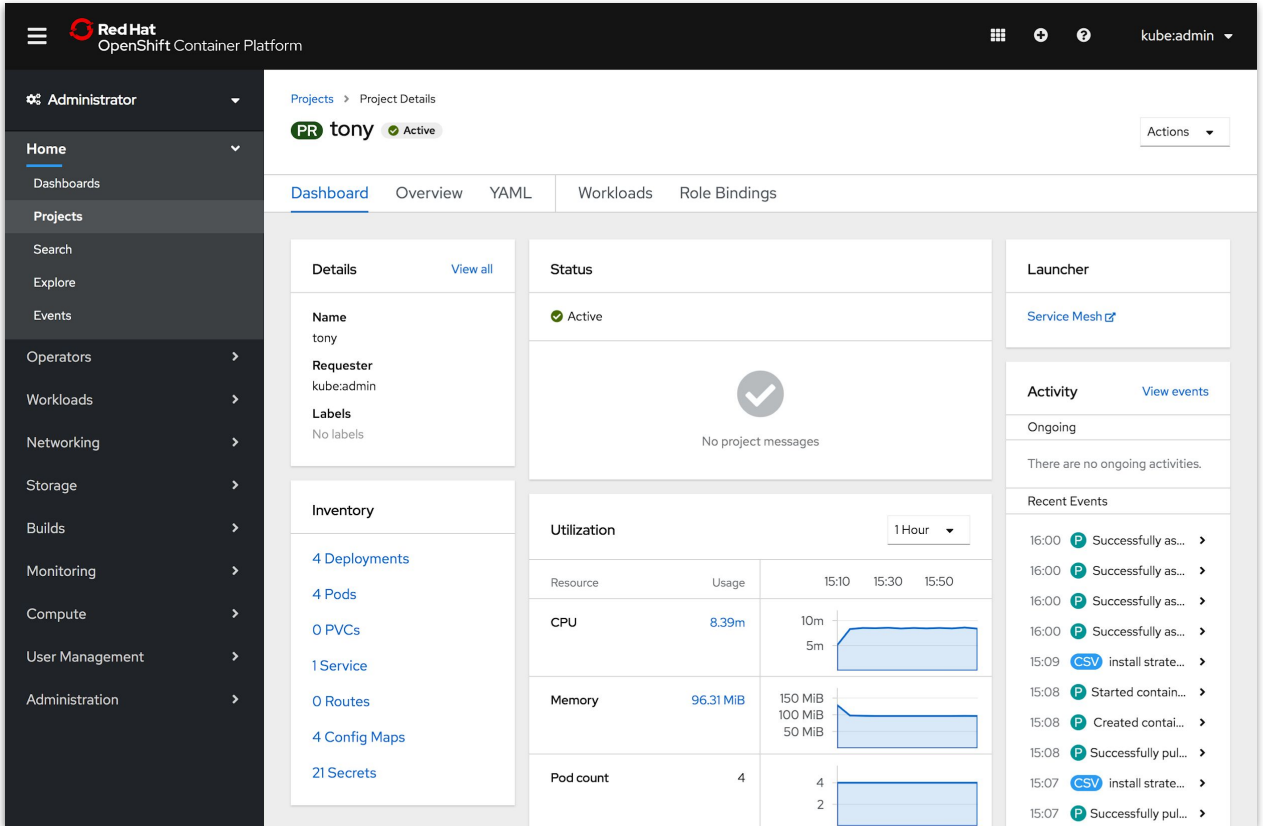
Red Hat

Enhanced Visibility with the New Project Dashboard

Project-scope Dashboard gives Developer Clear Insights

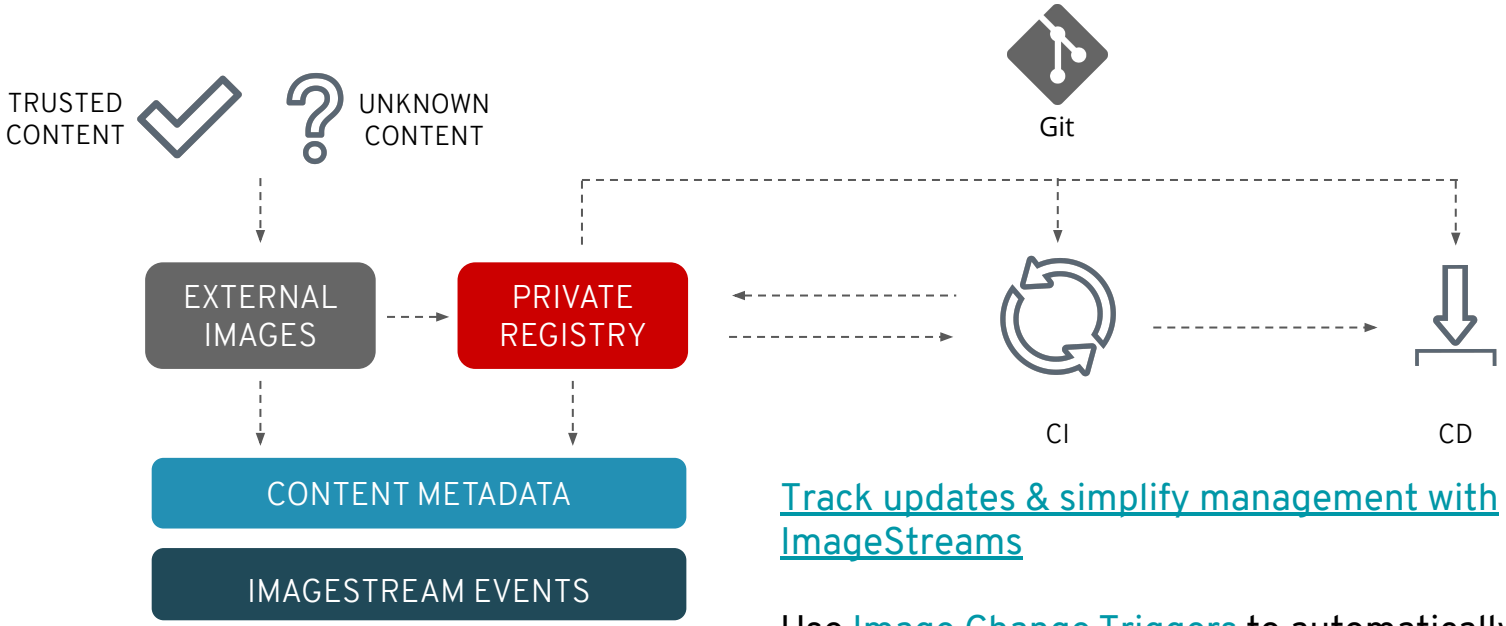
Drill down in context from the new project dashboard widgets:

- Project Details
- Project Status/Health
- Project External Links (Launcher)
- Project Inventory
- Project Utilization
- Project Resource Quota
- Project Activity (Top consumers)



SECURE & AUTOMATE THE CONTENT LIFECYCLE

Trust is temporal; rebuild and redeploy as needed



Track updates & simplify management with ImageStreams

Use Image Change Triggers to automatically rebuild custom images with updated (patched) external images

Red Hat

Red Hat Quay

RED HAT QUAY V3

Quay v3 Key Features

- Full support for Docker Registry API v2_s2
- Multi-arch (manifest) and Windows images
- Rebranded and rebased (RHEL)
- New config UI to save & upload Quay config
- Parallel upgrade (2x 5min downtime only)

Red Hat Quay and OpenShift

Red Hat Quay and OpenShift work well together. This includes both running Quay **on** and using Quay **with** OpenShift.

Introducing Red Hat Quay 3

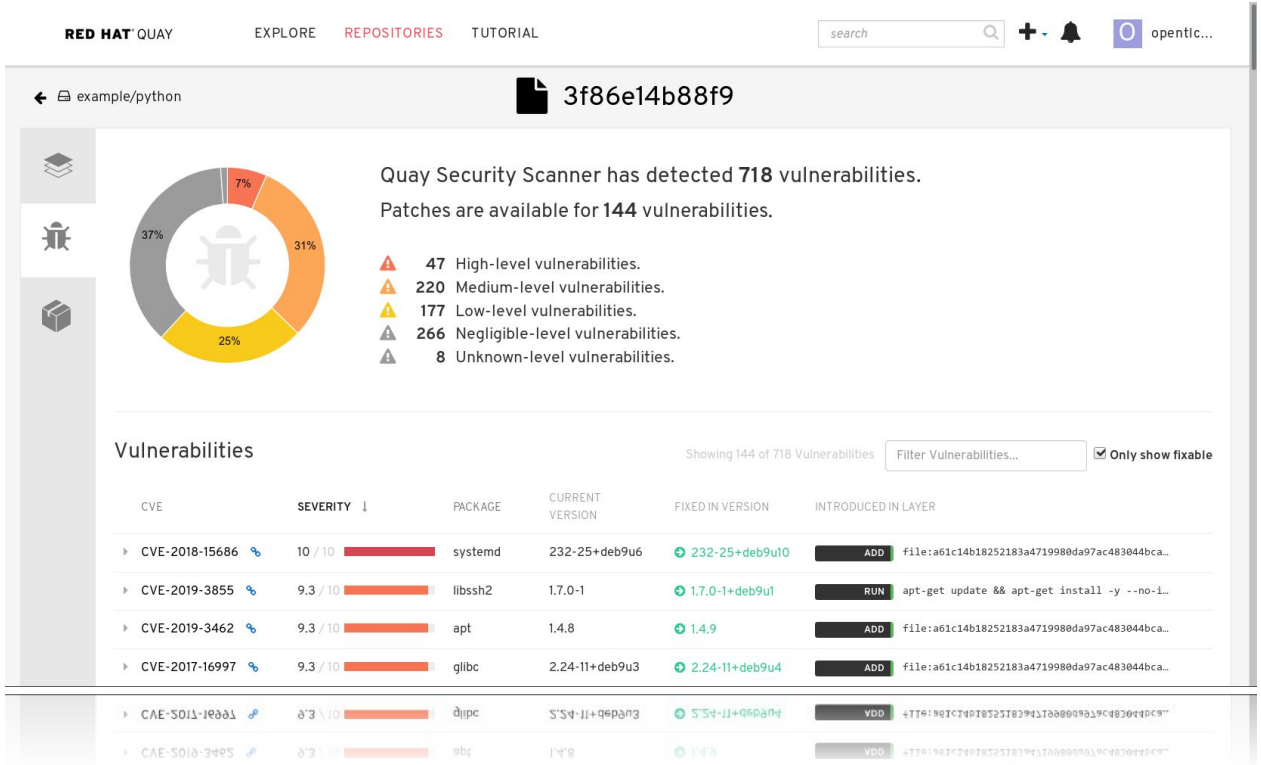
REGISTRY FEATURES

- **Vulnerability Scanning (powered by Clair)**
  Continually scan your containers for vulnerabilities, giving you complete visibility into known issues and how to fix them
- **Geographic Replication**
  Reliably store, build and deploy a single set of container images across multiple geographies
- **Automated software deployments**
  Streamline your continuous integration/continuous delivery (CI/CD) pipeline with build triggers, git hooks, and robot accounts
- **Advanced Access Control Management**
  Fine-grain access control of the registry with multiple identity and authentication providers as well as support for teams and organization mapping

VULNERABILITY SCANNING - CLAIR

Quay integrates with Clair to continually scan your containers for vulnerabilities.

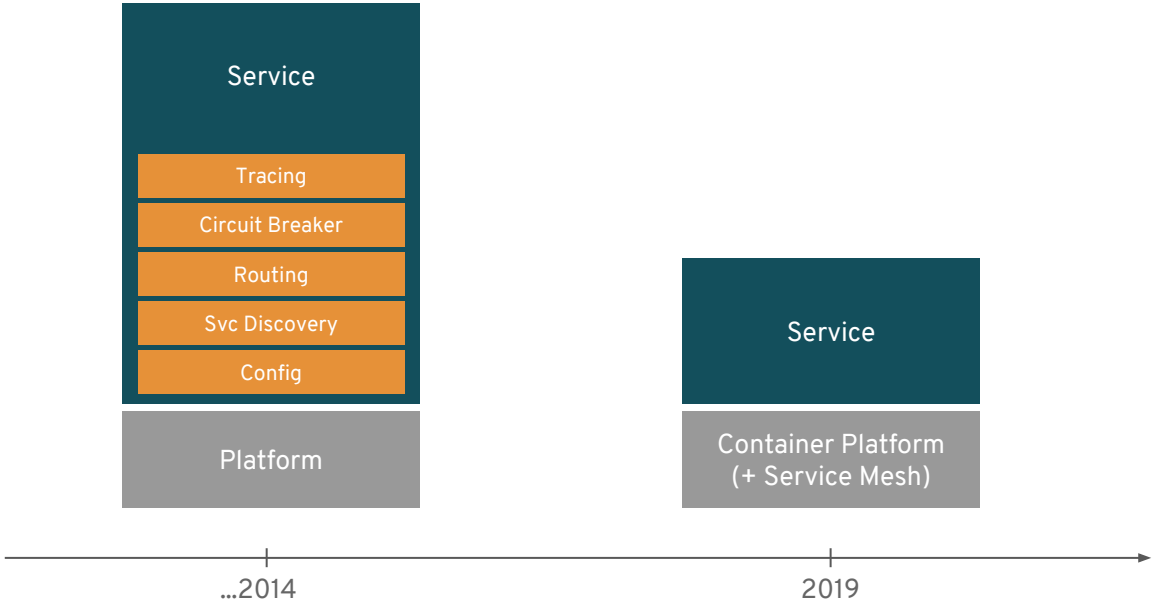
You get complete visibility into known issues and how to fix them.



Red Hat

Red Hat Service Mesh

MICROSERVICES EVOLUTION



Red Hat

RED HAT SERVICE MESH

Applies security, route rules, policies and reports traffic telemetry at the pod level

Jaeger

Secure

Istio

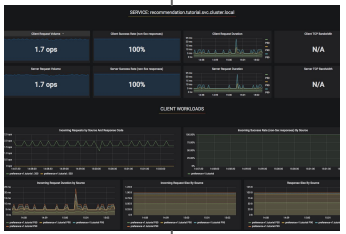
Observe

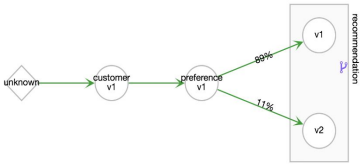
Connect

Control

Prometheus

Observe





kiali

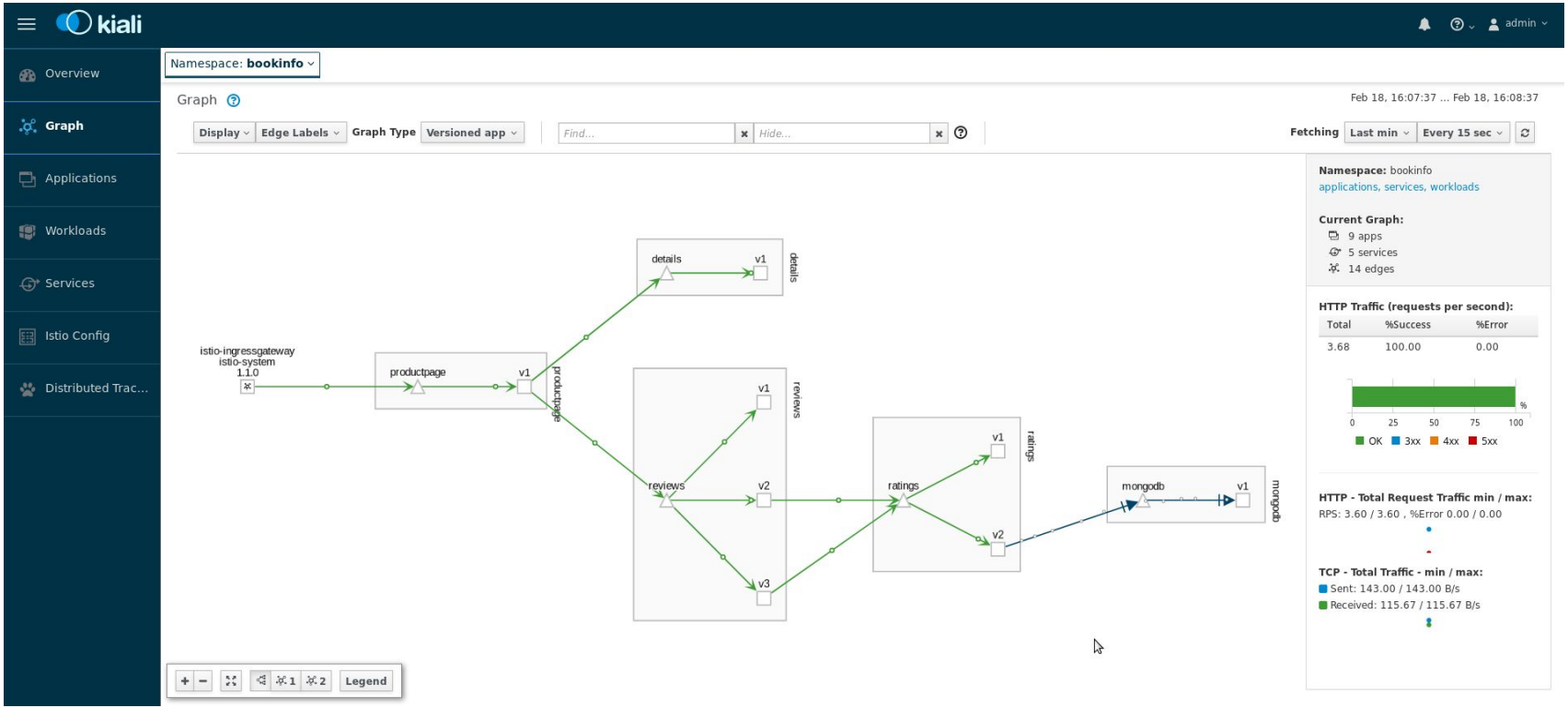
Kiali

Grafana

Generally Available in July

Red Hat

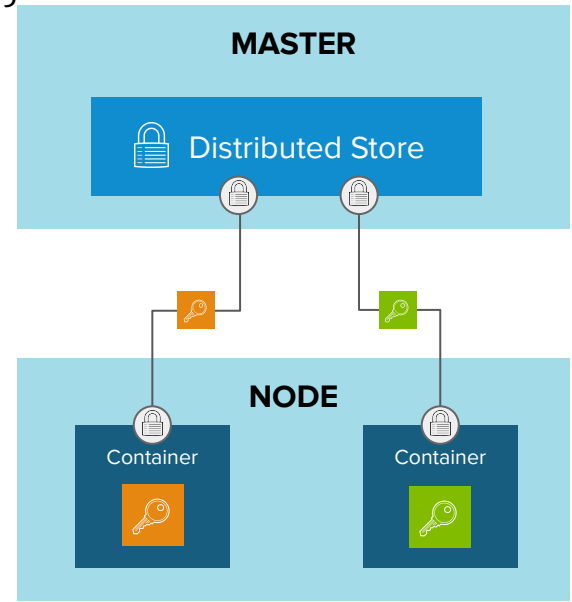
OBSERVABILITY WITH KIALI



Red Hat

SECRETS MANAGEMENT

- Secure mechanism for holding sensitive data e.g.
  - Passwords and credentials
  - SSH Keys
  - Certificates
- Secrets are made available as
  - Environment variables
  - Volume mounts
  - Interaction with external systems (e.g. vaults)
- Encrypted in transit and at rest*
- Never rest on the nodes

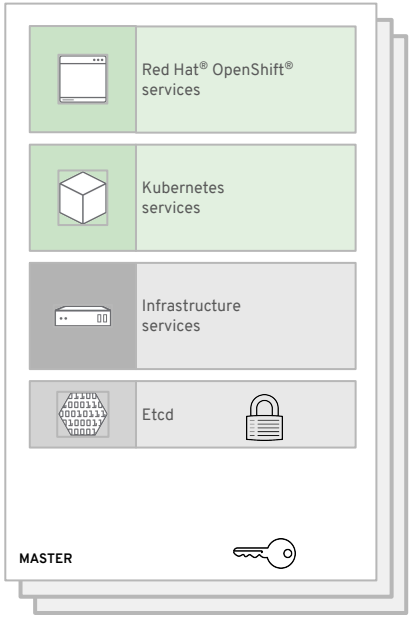


* Full disk encryption is supported for AWS, VSphere. Encrypting the etcd datastore is not available in OCP 4.1. Targeted for OCP 4.3.

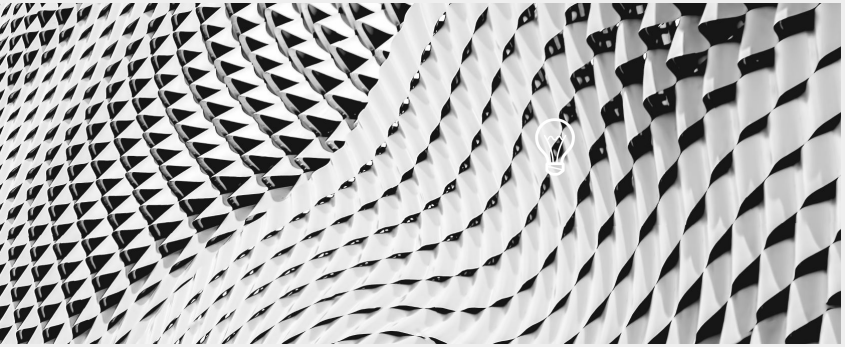
OpenShift 4 etcd Encryption

Encrypt secrets, config maps...

- Encryption of the etcd datastore is optional. Once enabled, encryption cannot be disabled.
- The aes-cbc cipher is used.
- Keys are created and automatically rotated by an operator and stored on the master node's file system.
- Keys are available as a secret via the kube API to a cluster admin.
- Assuming a healthy cluster: after enabling encryption, within a day, all relevant items in etcd are encrypted
- Backup: The etcd data store should be backed up separately from the file system with the key.
- Disaster recovery: a backup of both the encrypted etcd data and encryption keys must be available.



EXTEND SECURITY



Red Hat

THE SECURITY ECOSYSTEM

For enhanced security, or to meet existing policies, you may choose to integrate with enterprise security tools, such as

- Identity and Access management / Privileged Access Management
- External Certificate Authorities
- External Vaults / Key Management solutions
- Filesystem encryption tools
- Container content scanners & vulnerability management tools
- Container runtime analysis tools
- Security Information and Event Monitoring (SIEM)

THE BROADER SECURITY ECOSYSTEM

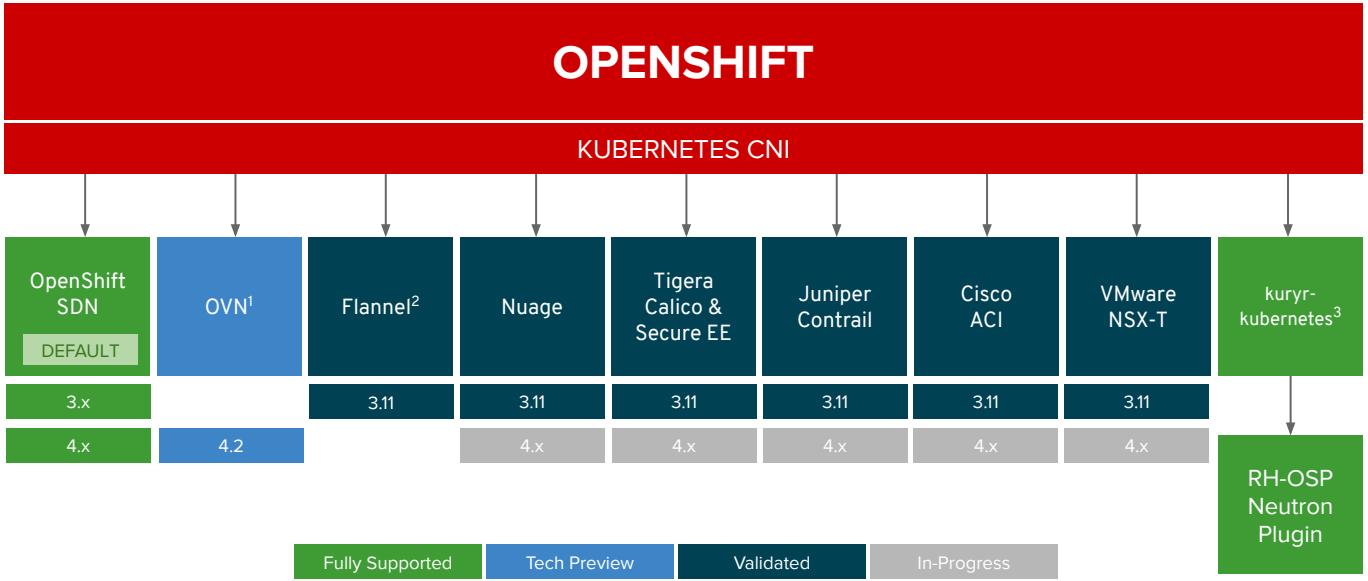






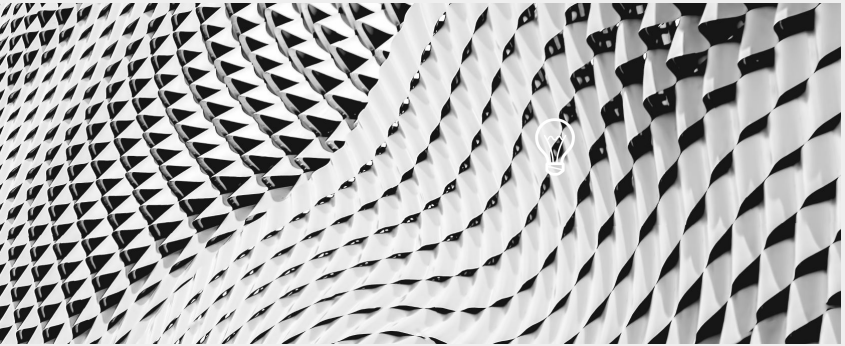

Red Hat

OpenShift network plugins



[1]Targeting GA at OCP 4.3 (not default SDN)
[2]Flannel is minimally verified and is supported only and exactly as deployed in the OpenShift on OpenStack reference architecture
[3]Available as an install-time option at 3.11.119 and 4.2.z (targeting 4.2.2)

ROADMAP



Red Hat

OpenShift Security Roadmap

Near Term (4.3)

- VPC / VPN support
- Encrypt etcd datastore
- FIPS compliance
- RHCOS disk encryption
- Reference architecture for GitOps based cluster config management with Argo CD
- External DNS (DNS Forwarding)
- Ingress Cipher and TLS Policy API

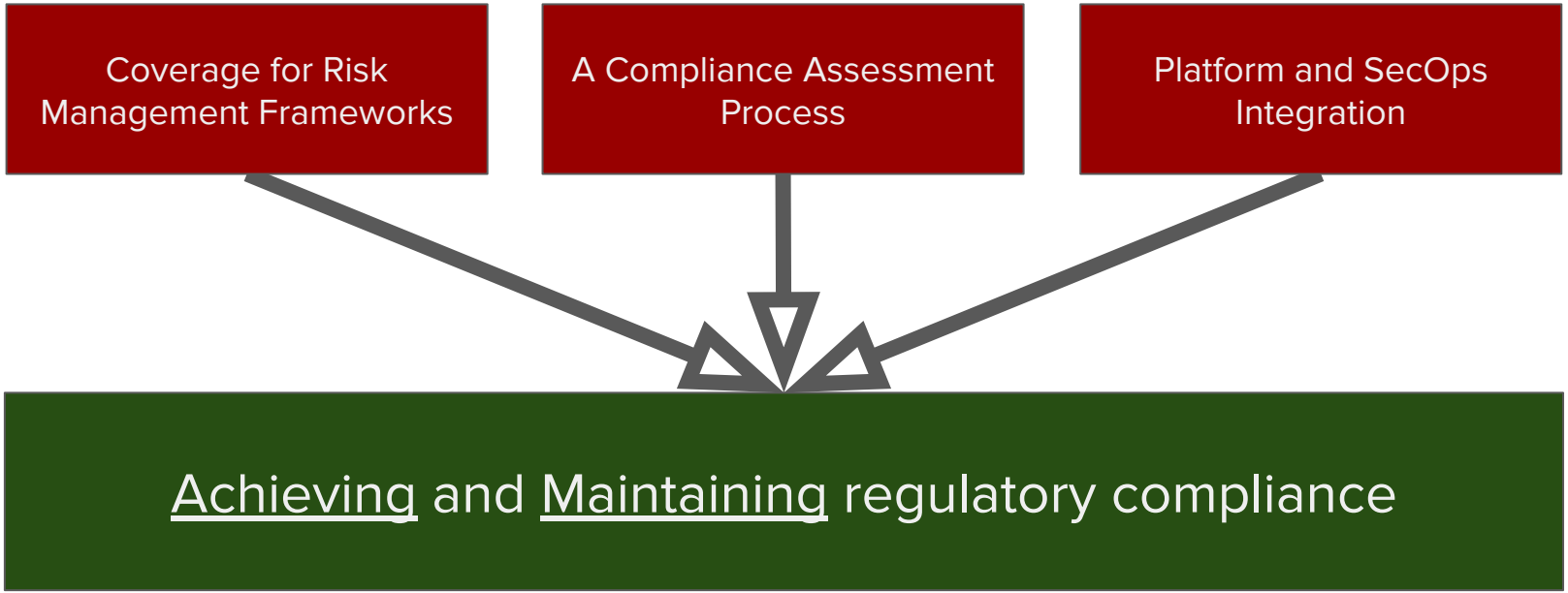
Longer Term (4.4+)

- Automate rotation of Service CA
- Global Options to Enable HTTP Strict Transport Security (HSTS)
- Full Support for IPv6, HTTP/2
- External Keycloak integration
- Service for application certificate lifecycle management
- Configure cipher suite for API server
- Integration workload identity with Cloud Provider IAM solutions
- Compliance operator
- Consume group membership from external Identity Provider

* Roadmap items are subject to change

RISK MANAGEMENT AND COMPLIANCE GOALS

Automate compliance audit and remediation



Linux Host Security
- RHCOS Immutable user space & automated updates
- SELinux+
- LUKS volume encryption
- FIPS mode

Authentication & Authorization
- Embedded OAuth Server
- Supports 9 Identity Providers including AD/LDAP
- Multi-Level Access Control (Users and Groups)
- Secrets and certificate management

Data Protection
- Encrypt secrets at rest (etcd datastore)
- All API server traffic is encrypted
- Configure cipher suites*
- Encrypt east / west traffic (Service Mesh)

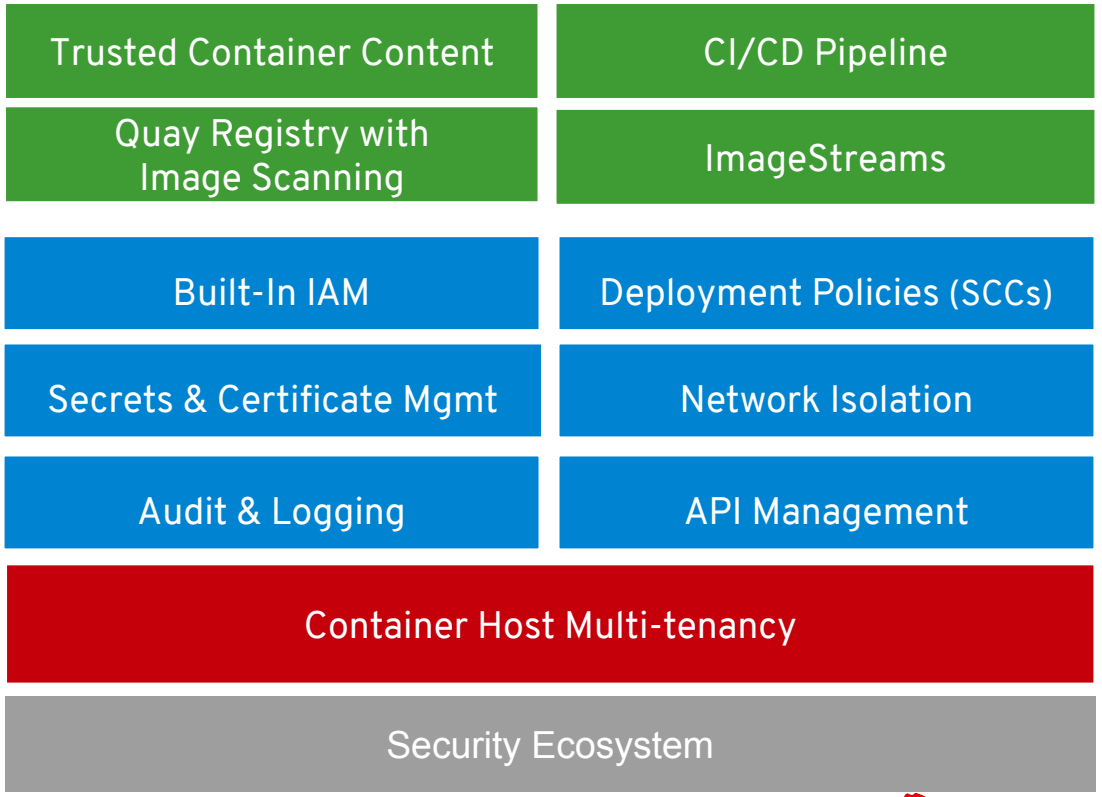
Image Security
- ImageStreams
- Scanning (Quay with Clair)
- Deployment policies (admission controllers)

Integrated Audit, Logging, Monitoring

Security Policies
- SCC (Security Context Controls)
- Non-Root Containers
- Controlled Access to Resources

Networking Isolation
- Ingress / Egress control
- Network microsegmentation

Defense in Depth with comprehensive security features



Ten Layers of Container Security

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