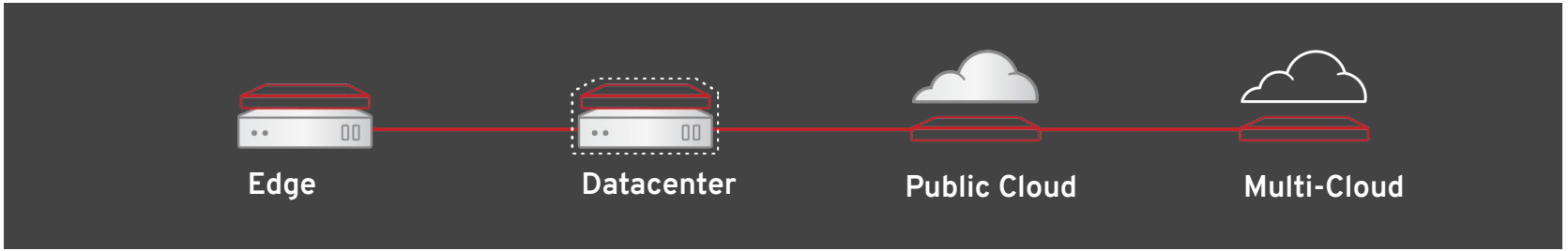
SECURING CONTAINERS WITH OPENSHIFT

Alfred Bach
Principal Solution Architect Cloud /Infra
EMEA Partner Enablement

Red Hat

What makes an effective hybrid cloud platform?

BROAD ECOSYSTEM	BROADEST APPLICATION SUPPORT	DEVELOPER EXPERIENCE & ON-DEMAND
AUTOMATED OPERATIONS	STANDARDS, PORTABILITY & INTEROPERABILITY	SECURITY & COMPLIANCE



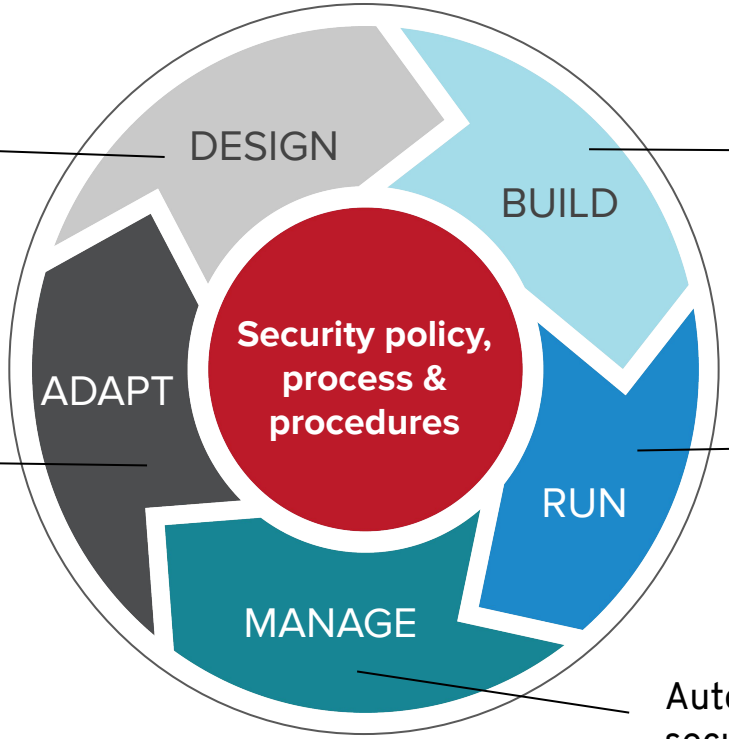
Red Hat

Security must be continuous

And integrated throughout the IT lifecycle

Identify security requirements & governance models

Built-in from the start; not bolted-on



Revise, update, remediate as the landscape changes

Deploy to trusted platforms with enhanced security capabilities

Automate systems for security & compliance

Red Hat

A Comprehensive Approach to Securing Containers

CONTROL
Application Security

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

DEFEND
Infrastructure

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

EXTEND

- Security Ecosystem

Hardening, applicability guides, certifications

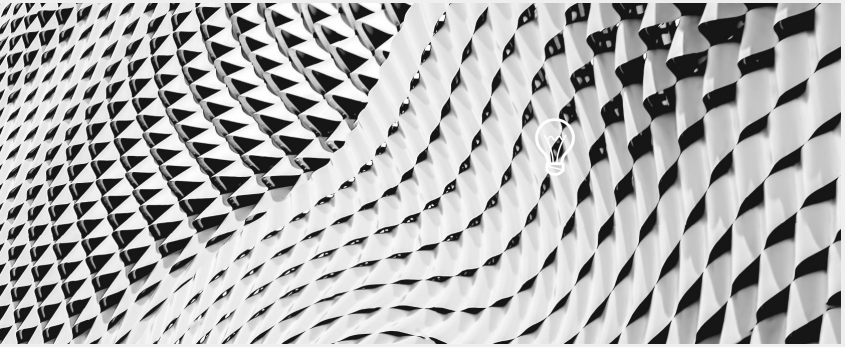
OpenShift 4

- Available now
  - HIPAA
  - ISO 27001 (ask RH for a copy)
  - FISMA (ask RH for a copy)
  - The OpenShift Security Guide
  - OpenShift 4 Hardening Guide (ask RH for a copy)
- Target Q1 CY 2021
  - CIS OpenShift Benchmark
  - HITRUST
  - PCI-DSS

Managed Services certifications

- SOC2-type 1, SOC2-type 2
  - OpenShift Dedicated (OSD) on AWS
  - ARO, IBM ROKS, ROSA
  - In process for OSD on GCP
- ISO-27001
  - OSD on AWS, ARO
- PCI-DSS
  - ARO, IBM ROKS
  - In process for OSD on AWS and GCP, ROSA
- FedRAMP
  - ARO, IBM ROKS
  - In process for OSD on AWS, ROSA
- HIPAA and/or HITRUST
  - ARO, IBM ROKS
  - In discussion for OSD and ROSA

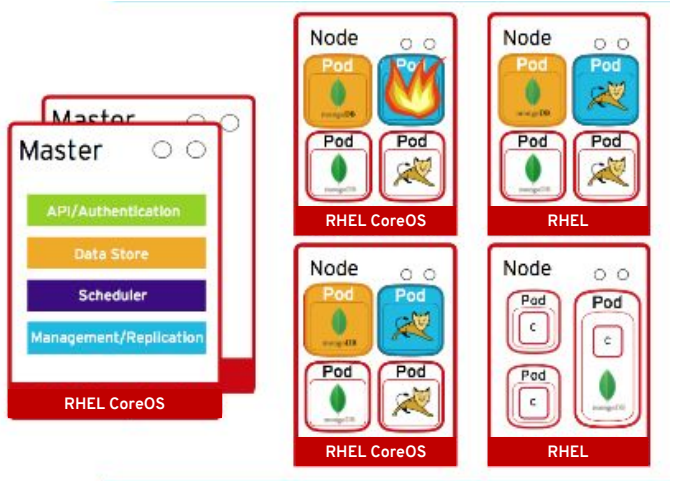
DEFEND INFRASTRUCTURE



Red Hat

Securing the container platform

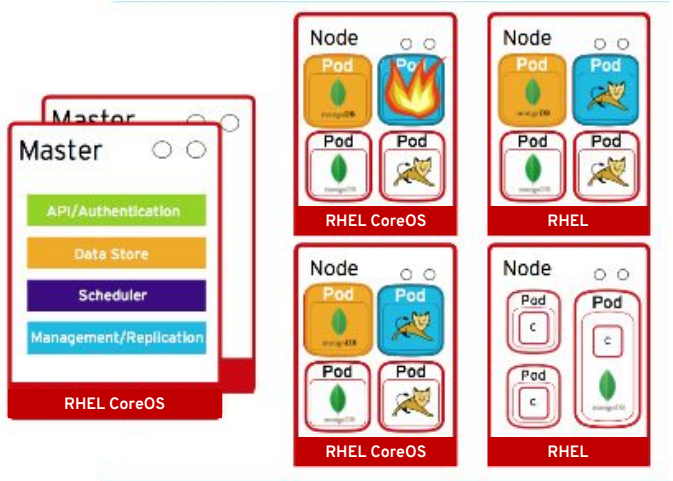
- Configuration and lifecycle management
- Host & runtime security
- Identity and Access Management
- Data at rest, data in transit
- Logging, Monitoring, Metrics
- Audit and Compliance



Red Hat

Securing the container platform

- **Configuration and lifecycle management**
- Host & runtime security
- Identity and Access Management
- Data at rest, data in transit
- Logging, Monitoring, Metrics
- Audit and Compliance



Red Hat

Automated Configuration and Lifecycle Management

Dramatically simplified for the Hybrid Cloud

Machines

Machines are complex for ops

Make machines easy
(like containers)

Configuration

Config change is risky

Make config management
and config change
easy and safe

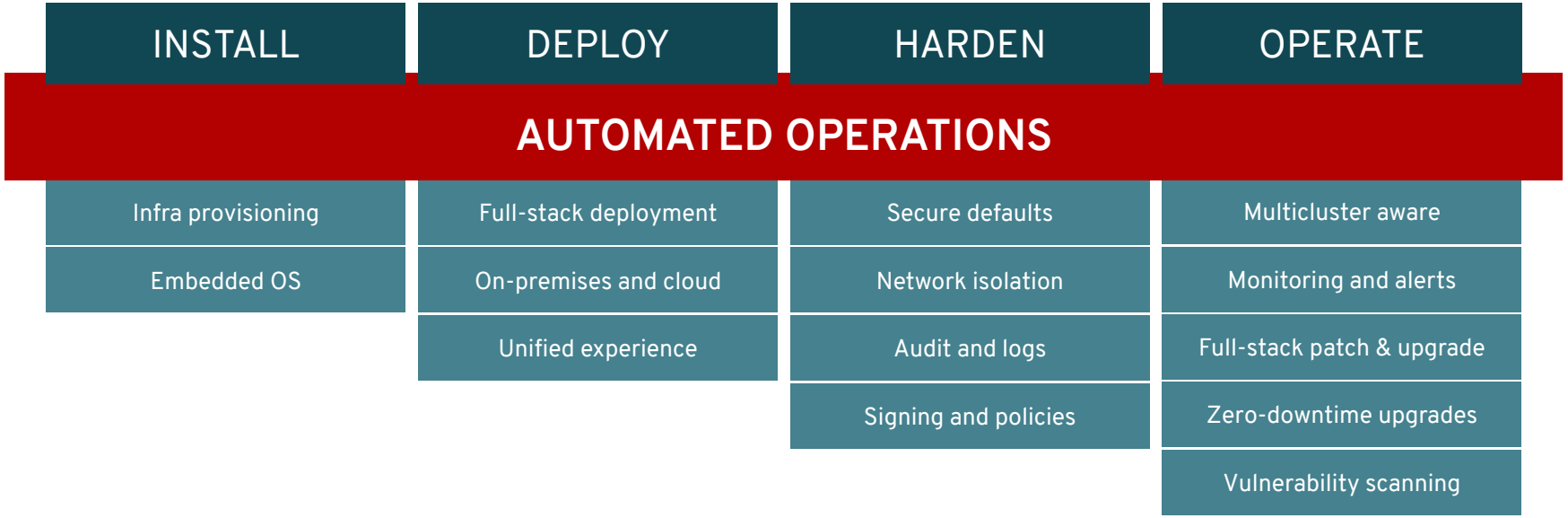
Lifecycle

Software lifecycle is hard

Automate software
lifecycle on Kube

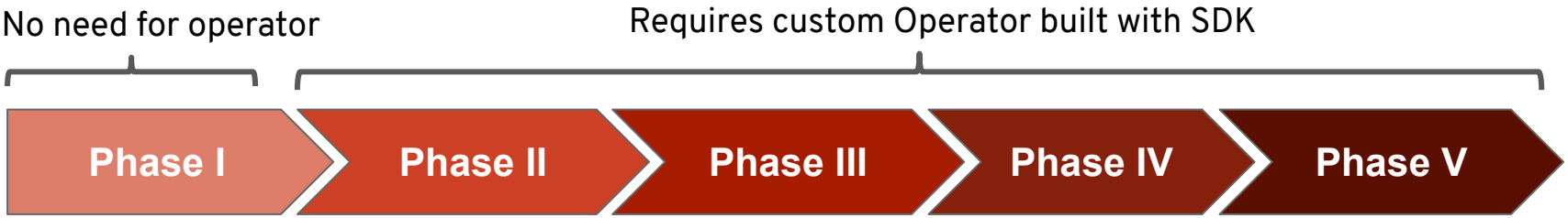
Automated Container Operations

FULLY AUTOMATED DAY-1 AND DAY-2 OPERATIONS



Red Hat

The Value Of Kubernetes Operators



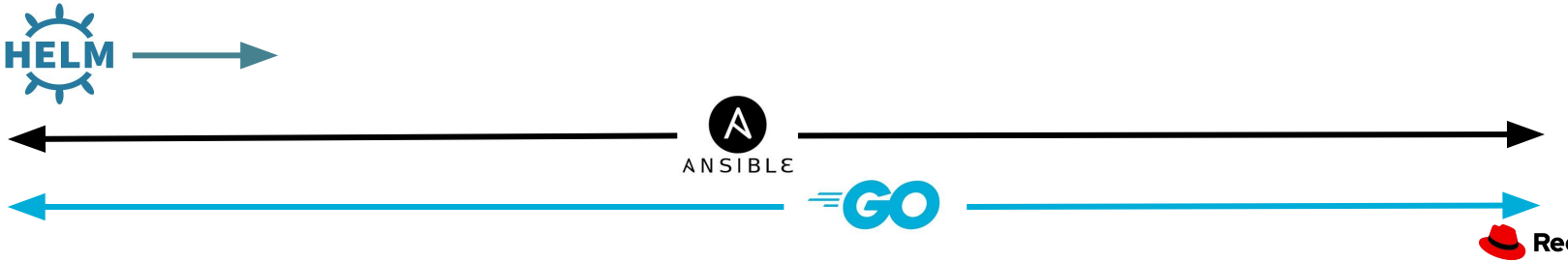
Installation
Automated application provisioning and configuration management

Upgrades
Patch and minor version upgrades supported

Lifecycle
App lifecycle, storage lifecycle (backup, failure recovery)

Deep Insights
Metrics, alerts, log processing and workload analysis

Auto-pilot
Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning...



Day 1 Installation

OPENSHIFT CONTAINER PLATFORM

Full Stack Automation

Simplified opinionated "Best Practices" for cluster provisioning

Fully automated installation and updates including host container OS.

Red Hat Enterprise Linux CoreOS

Pre-existing Infrastructure

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries

Red Hat Enterprise Linux CoreOS

Red Hat Enterprise Linux

HOSTED OPENSHIFT

Azure Red Hat OpenShift

Deploy directly from the Azure console. Jointly managed by Red Hat and Microsoft Azure engineers.

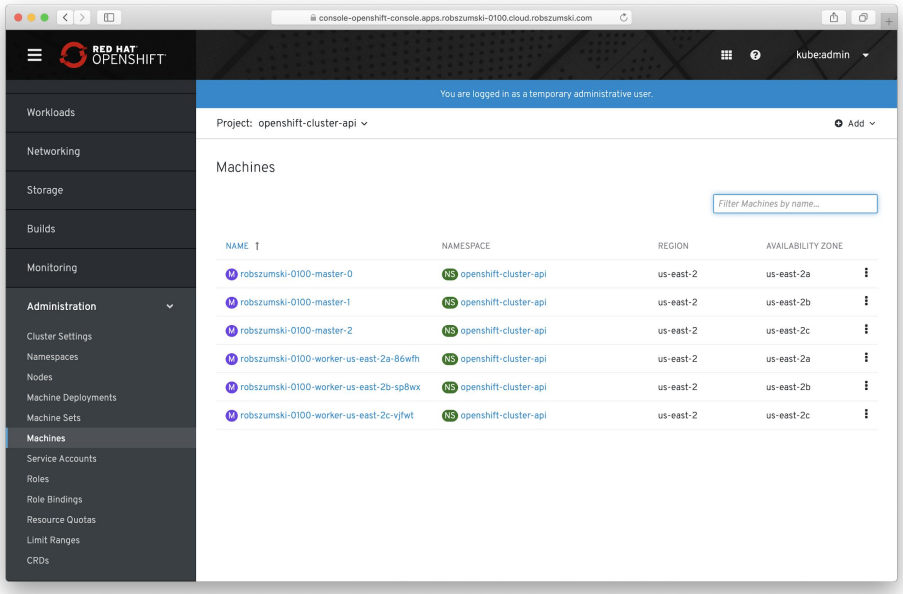
OpenShift Dedicated

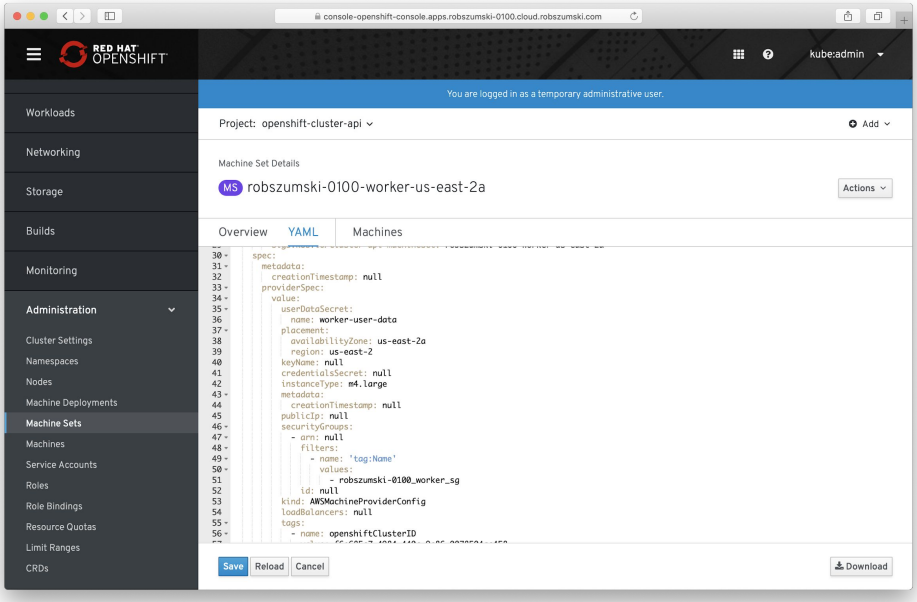
Get a powerful cluster, fully Managed by Red Hat engineers and support.

Generally Available

Kubernetes Machine Api Operator

Using Kubernetes To Provision And Scale Clusters





Day 2 Configuration

Global Configuration
You complete most of the cluster configuration and customization after you deploy your OpenShift Container Platform cluster.

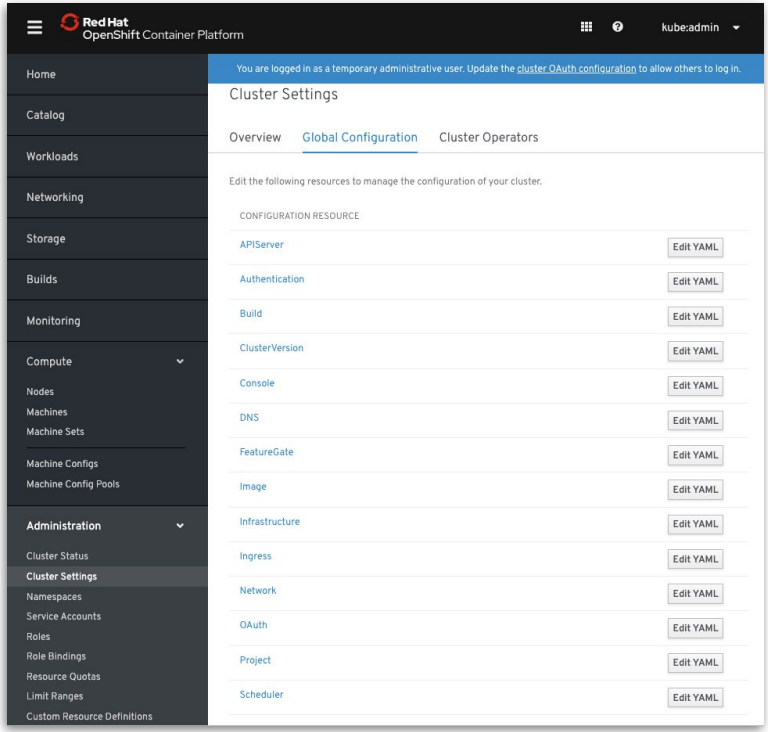
Change via Cluster Settings screen

Once you have discovered your desired settings (prev. slide), changes can be made via Console or CLI.

Operators apply these updates

One or more Operators are responsible for propagating these settings through the infrastructure

- Identity Provider
- Ingress Controller
- Logging, Metrics



Smarter Software Updates

No downtime for well behaving apps

Applications with multiple replicas, using liveness probes, health checks and taints/tolerations

Node Pools with more than one worker and slack resources

Maintenance window for entire cluster

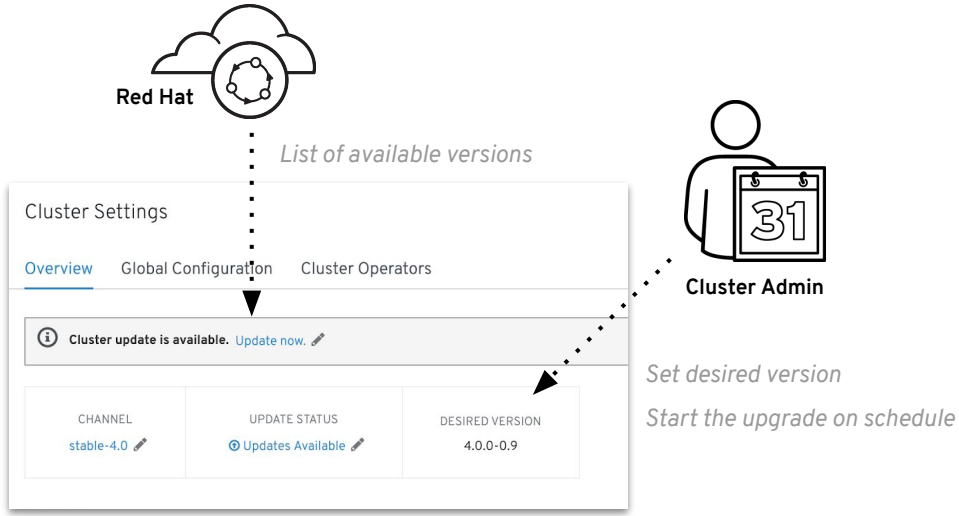
No need for separate windows for each component

Upgrade runs completely on the cluster

No more long running processes on a workstation

Constant health checking from each Operator

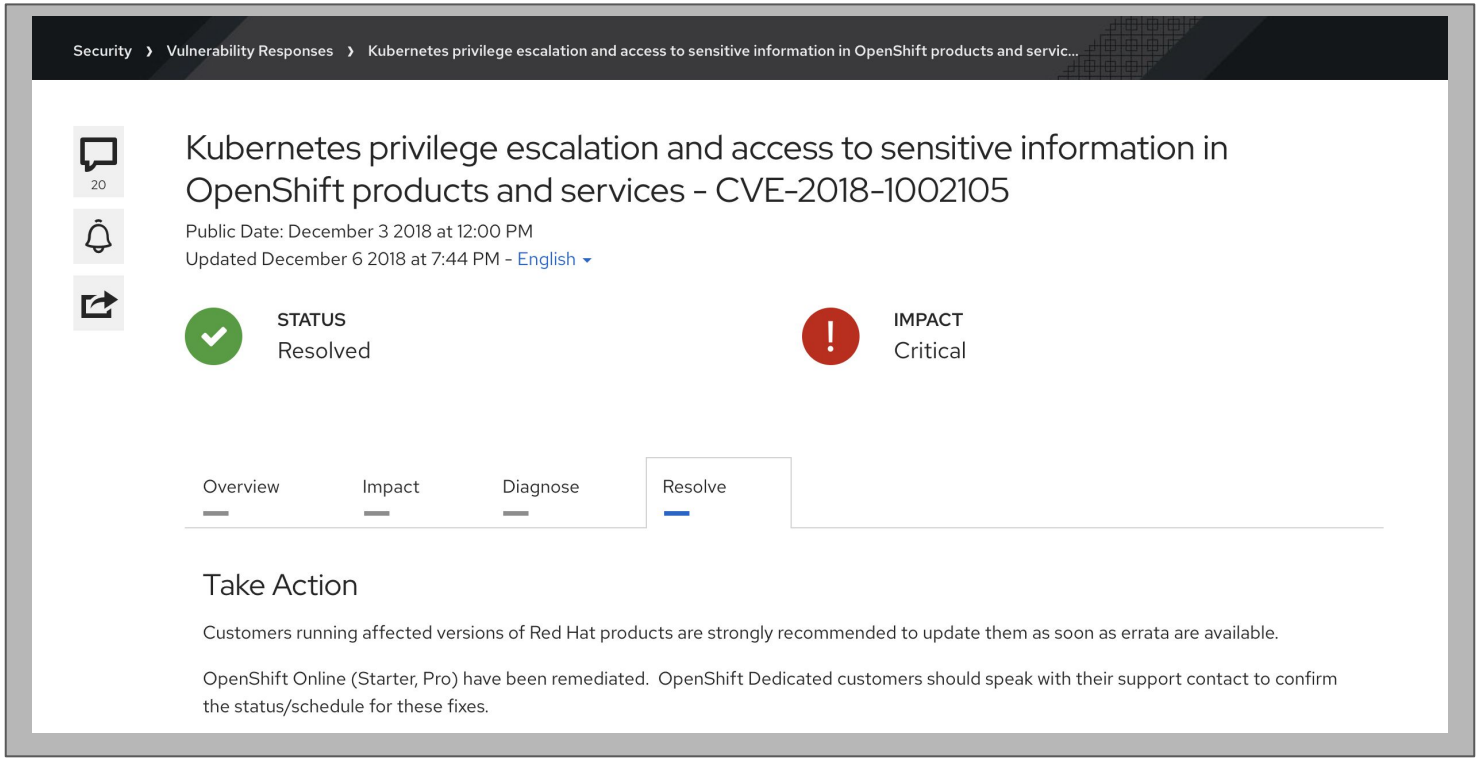
Operators are constantly looking for incompatibilities and issues that might arise



[NEW: OpenShift Update Service for restricted networks]

Red Hat

Fixes to Kubernetes vulnerabilities



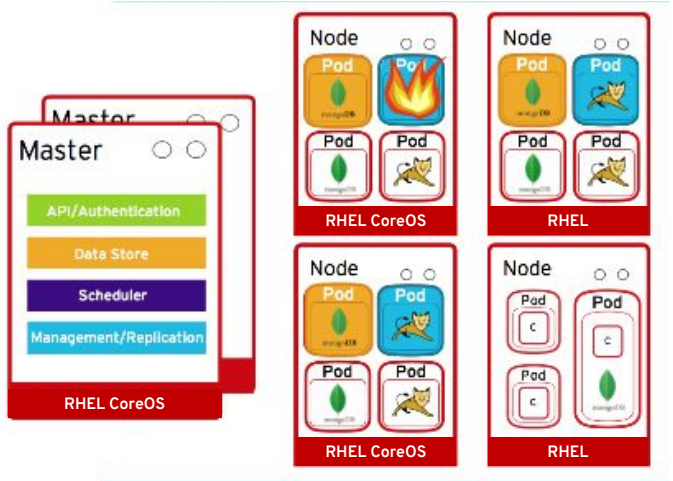
https://access.redhat.com/security/vulnerabilities/3716411

Fix backported all the way back to OpenShift 3.2 (Kubernetes 1.2)

Red Hat

Securing the container platform

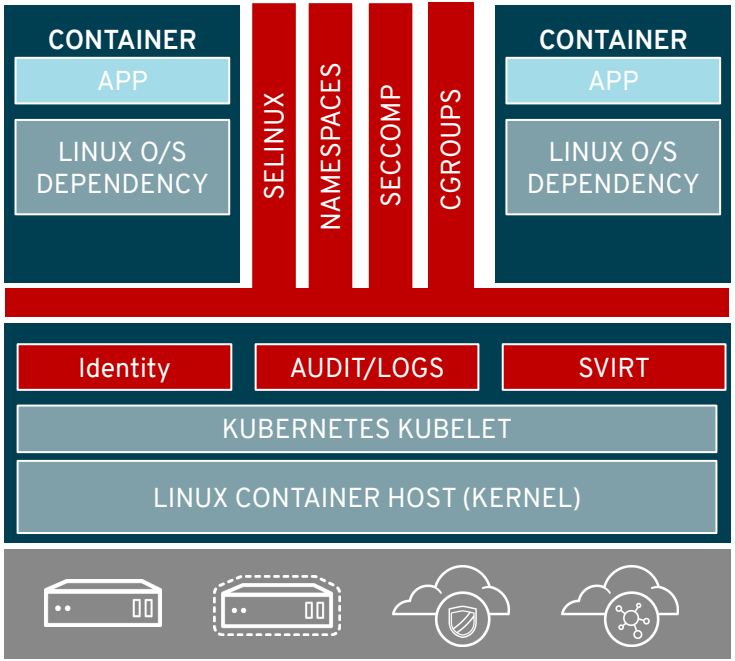
- Configuration and lifecycle management
- **Host & runtime security**
- Identity and Access Management
- Data at rest, data in transit
- Logging, Monitoring, Metrics
- Audit and Compliance



Red Hat

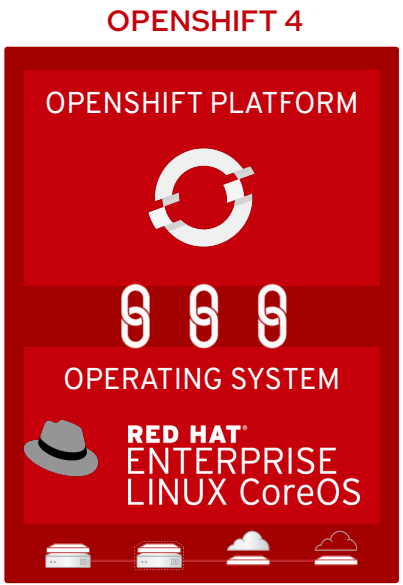
Container security starts with Linux security

- Security in the RHEL host applies to the container
- RHEL enables container multitenancy
- 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 Enterprise Linux CoreOS

The Immutable Container Optimized Operating System



Role in OpenShift Ecosystem

- Versioned and validated for specific OpenShift version
- Required for masters. RHEL option for workers
- User space read-only

Managed by the OpenShift Cluster

- Considered a member of an OpenShift Deployment
- Configuration managed by the Machine Config Operator
  - Container runtime
  - Kubelet configuration
  - Authorized container registries
  - SSH Configuration

Red Hat

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]

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
    UID:                                <none>
    UID Range Min:                      <none>
    UID Range Max:                      <none>
  SELinux Context Strategy: MustRunAs
    User:                               <none>
    Role:                               <none>
    Type:                               <none>
    Level:                              <none>
  FSGroup Strategy: MustRunAs
    Ranges:                             <none>
  Supplemental Groups Strategy: RunAsAny
    Ranges:                             <none>
```

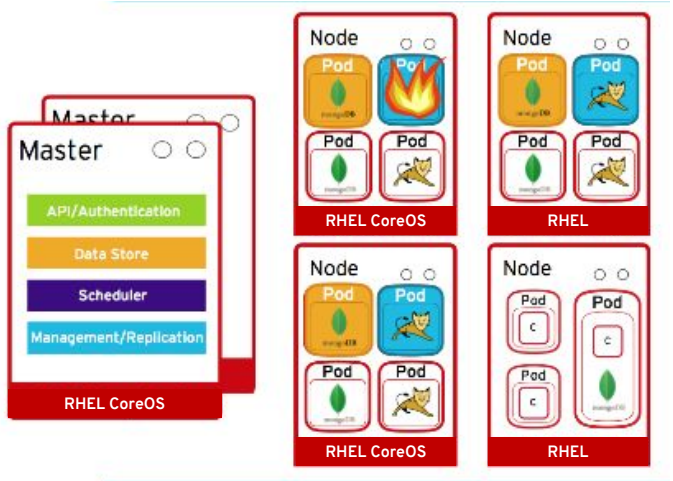
1. Lists which users and service accounts the SCC is applied to.
2. Lists which groups the SCC is applied to.

(Security Context Constraints)

Red Hat

Securing the container platform

- Configuration and lifecycle management
- Host & runtime security
- **Identity and Access Management**
- Data at rest, data in transit
- Logging, Monitoring, Metrics
- Audit and Compliance



Red Hat

Identity and access management

OpenShift includes an OAuth server, which does three things:

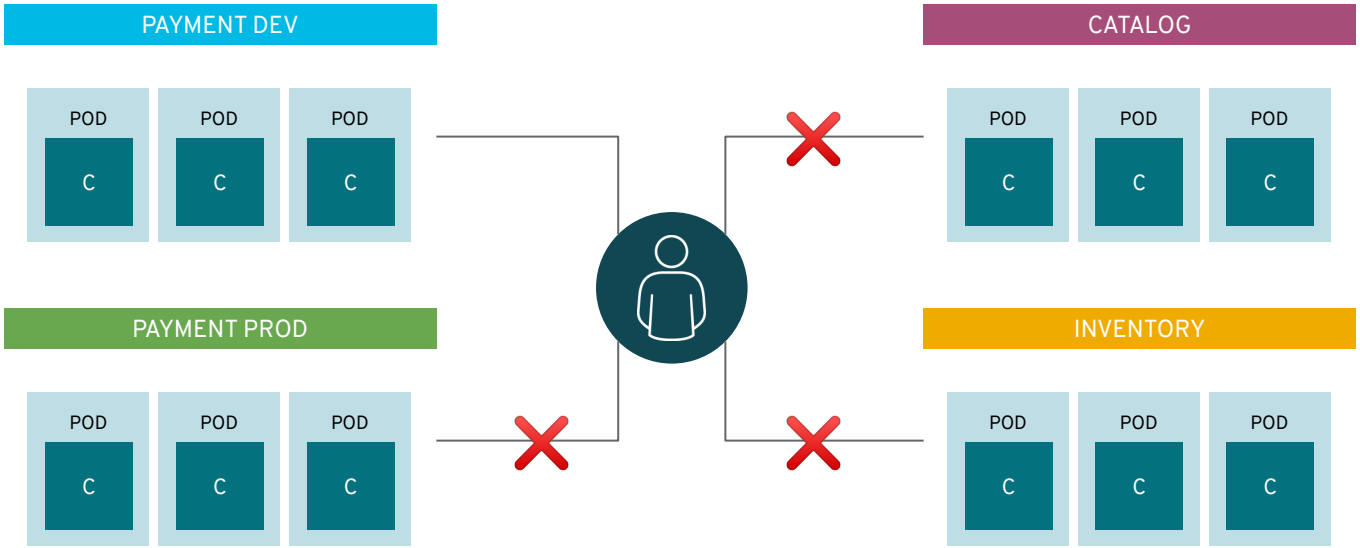
- Identifies the person requesting a token, using a configured identity provider
- Determines a mapping from that identity to an OpenShift user
- Issues an OAuth access token which authenticates that user to the API

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)

Projects isolate applications across teams, groups and departments



Red Hat

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

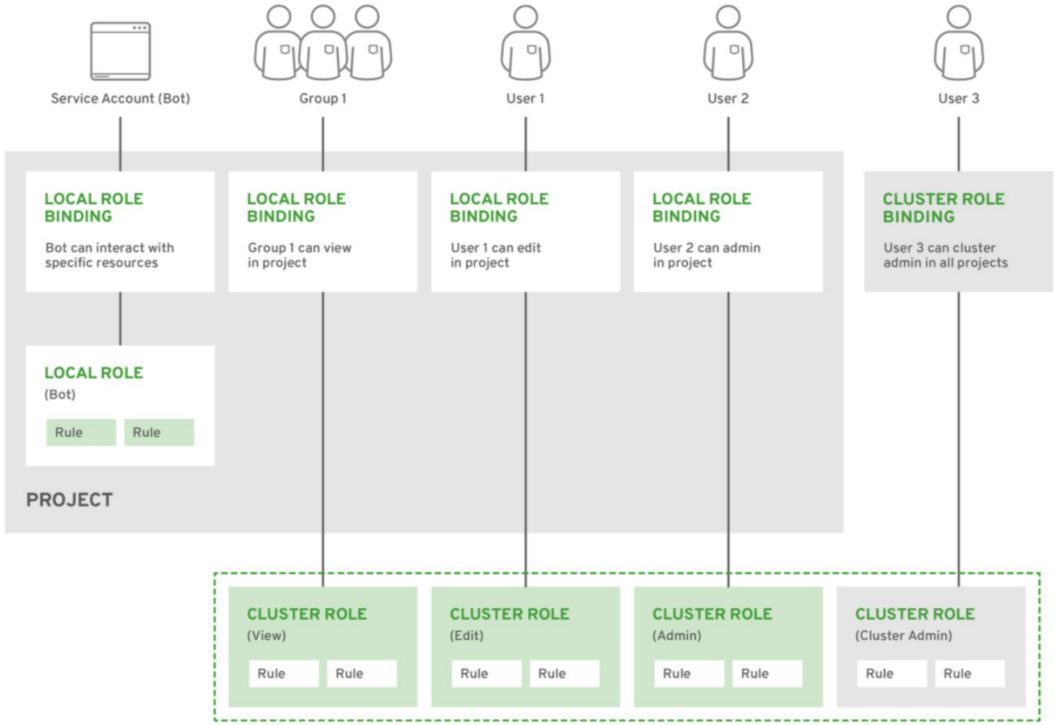
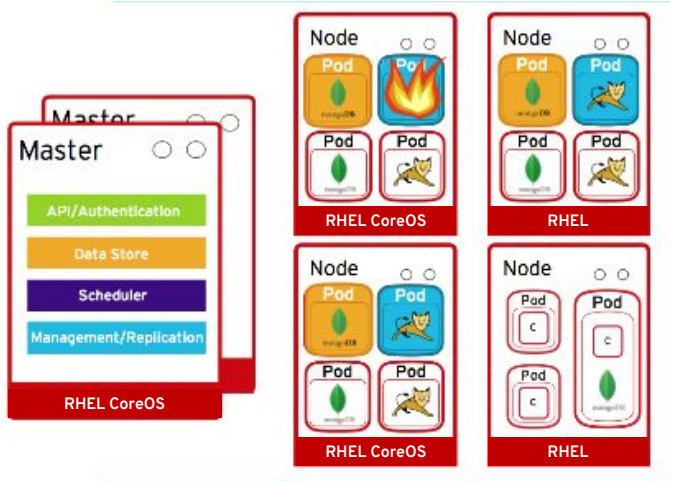


Figure 12 - Authorization Relationships

For more information see: [Managing RBAC in OpenShift]

Securing the container platform

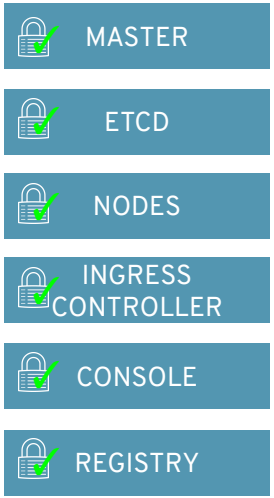
- Configuration and lifecycle management
- Host & runtime security
- Identity and Access Management
- **Data at rest, data in transit**
- Logging, Monitoring, Metrics
- Audit and Compliance



Red Hat

Certificate management

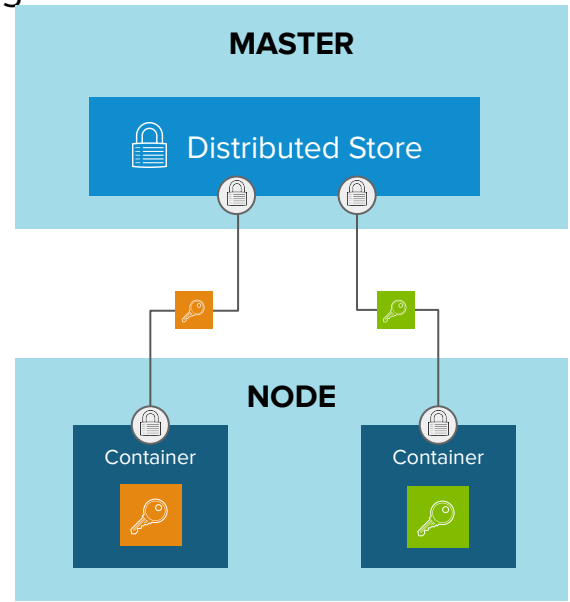
- Certificates are used to provide secure connections to
  - master and nodes
  - Ingress controller and registry
  - etcd
- Certificate rotation is automated
- Optionally configure external endpoints to use custom certificates
- For example:
  [Requesting and Installing Let's Encrypt Certificates for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



Red Hat

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
  - Encrypt the etcd datastore
  - Encrypt RHCOS volumes
- Never rest on the nodes



Volume Encryption

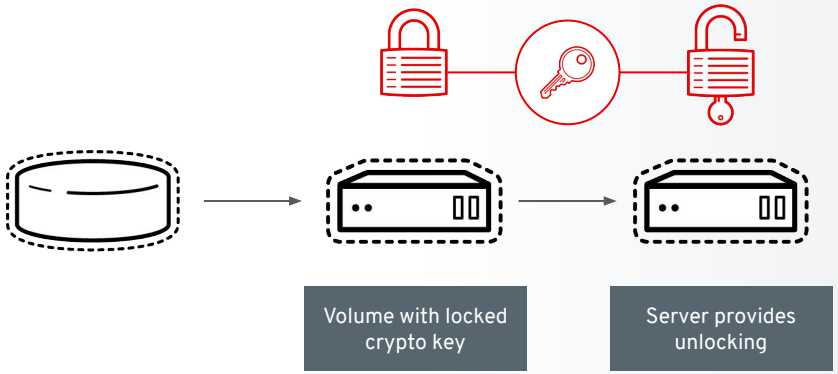
Network Bound Disk Encryption

- Provides encryption for local storage
- Addresses disk/image theft
- Platform/cloud agnostic implementation
- TPM/vTPM (v2) and Tang endpoints for automatic decryption

Attached storage

Secure storage by using

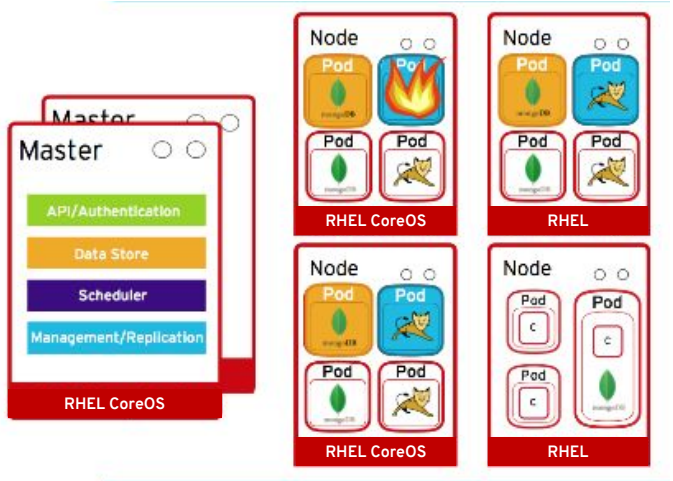
- SELinux access controls
- Secure mounts
- Supplemental group IDs for shared storage
- Network bound disk encryption



Red Hat

Securing the container platform

- Configuration and lifecycle management
- Host & runtime security
- Identity and Access Management
- Data at rest, data in transit
- **Logging, Monitoring, Metrics**
- Audit and Compliance

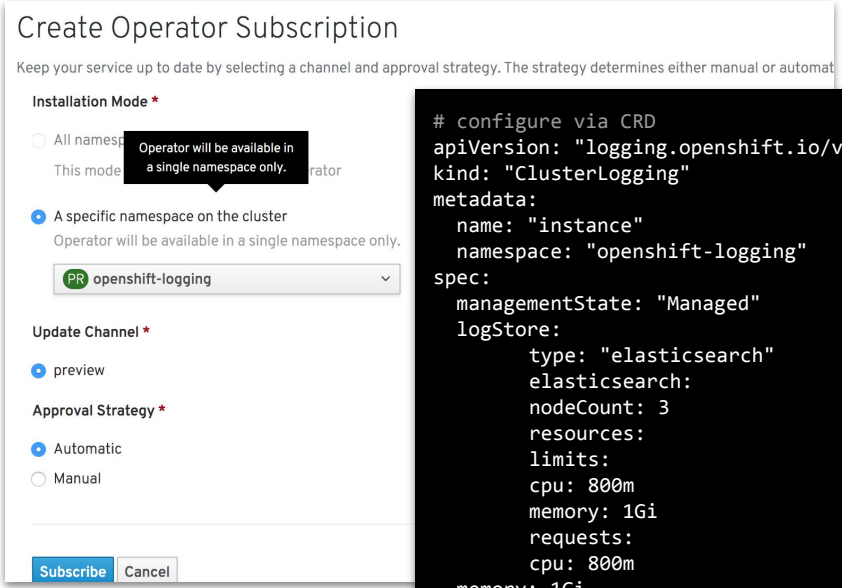


Red Hat

Cluster log and audit management

Install the Elasticsearch and Cluster Logging Operators

- 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
- API server events are automatically audited
- Logging pipelines collect API server and host audit logs as well as cluster and application logs for forwarding to the SIEM of your choice

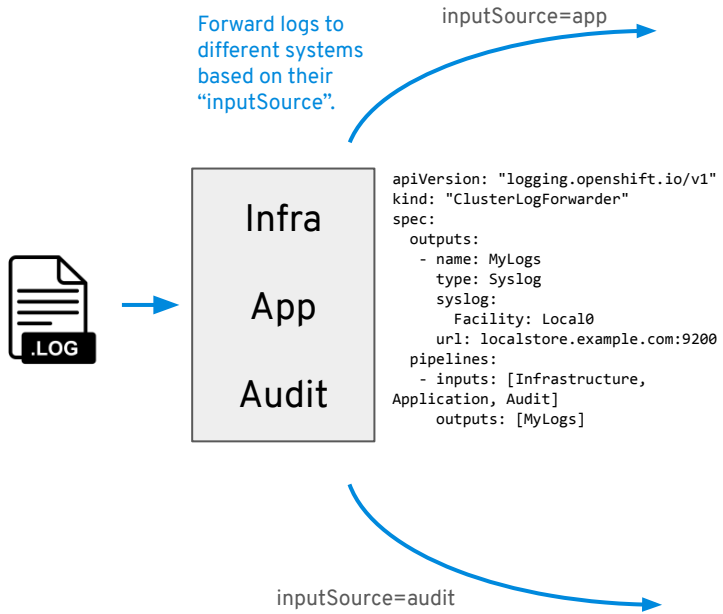



```
# configure via CRD
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  logStore:
        type: "elasticsearch"
        elasticsearch:
        nodeCount: 3
        resources:
        limits: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

Introduce new log forwarding API

Abstract Fluentd configuration by introduce new log forwarding API to improve support and experience for customers.

- Introduce a new, cluster-wide *ClusterLogForwarder* CRD (API) that replaces needs to configure log forwarding via Fluentd ConfigMap.
- The API helps to reduce probability to misconfigure Fluentd and helps bringing in more stability into the Logging stack.
- Route logs based on their source type (infra, app or audit logs) and filter them further by namespaces.
- Collecting and forwarding audit logs
- With the API, we also introduce the following endpoint improvements to bring more value on to the table:
    - Improved syslog support adding TLS for secure communication + support for the newest standard (RFC5424).
    - Kafka support.

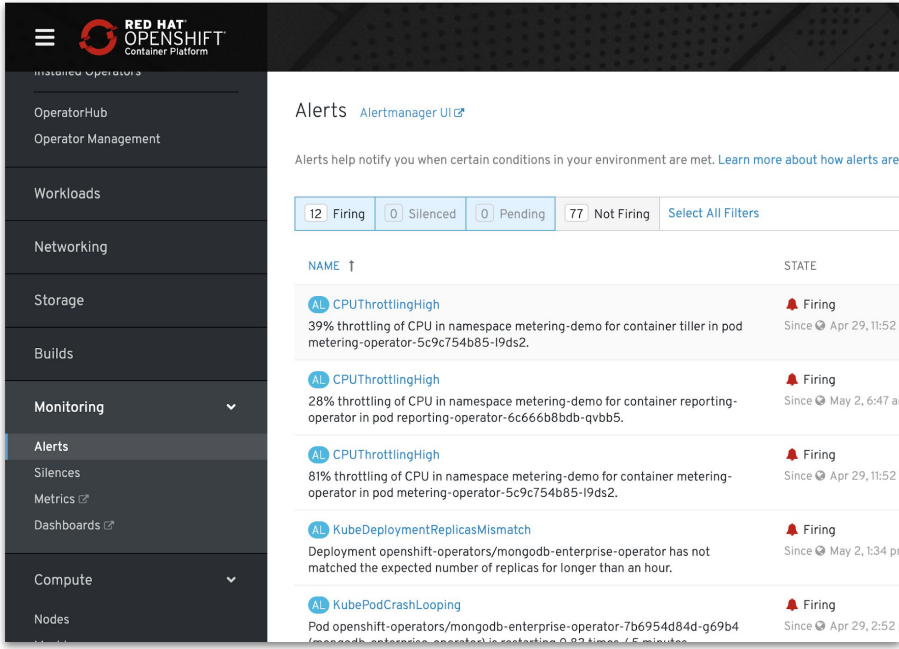


Red Hat

Cluster monitoring

Cluster monitoring is installed by default

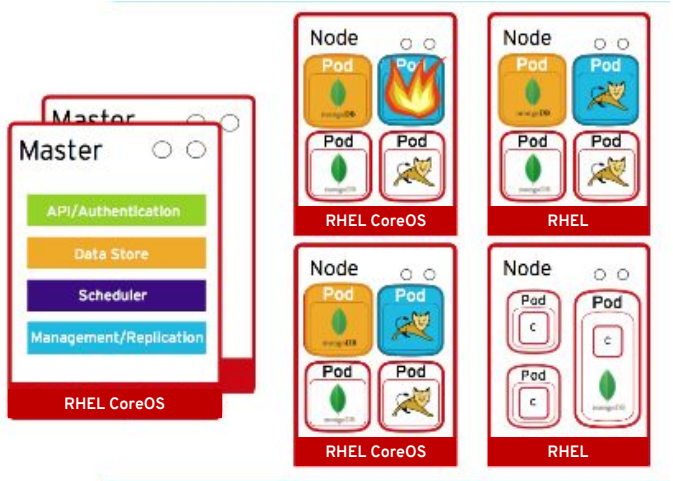
- Exposes resource metrics for Horizontal Pod Autoscaling (HPA) by default
  - HPA based on custom metric is tech preview
- No manual etcd monitoring configuration anymore
- New screens for managing Alerts & Silences
- More metrics available for troubleshooting purposes (e.g. HAproxy)
- Configuration via ConfigMaps and Secrets



Red Hat

Securing the container platform

- Configuration and lifecycle management
- Host & runtime security
- Identity and Access Management
- Data at rest, data in transit
- Logging, Monitoring, Metrics
- **Audit and Compliance**

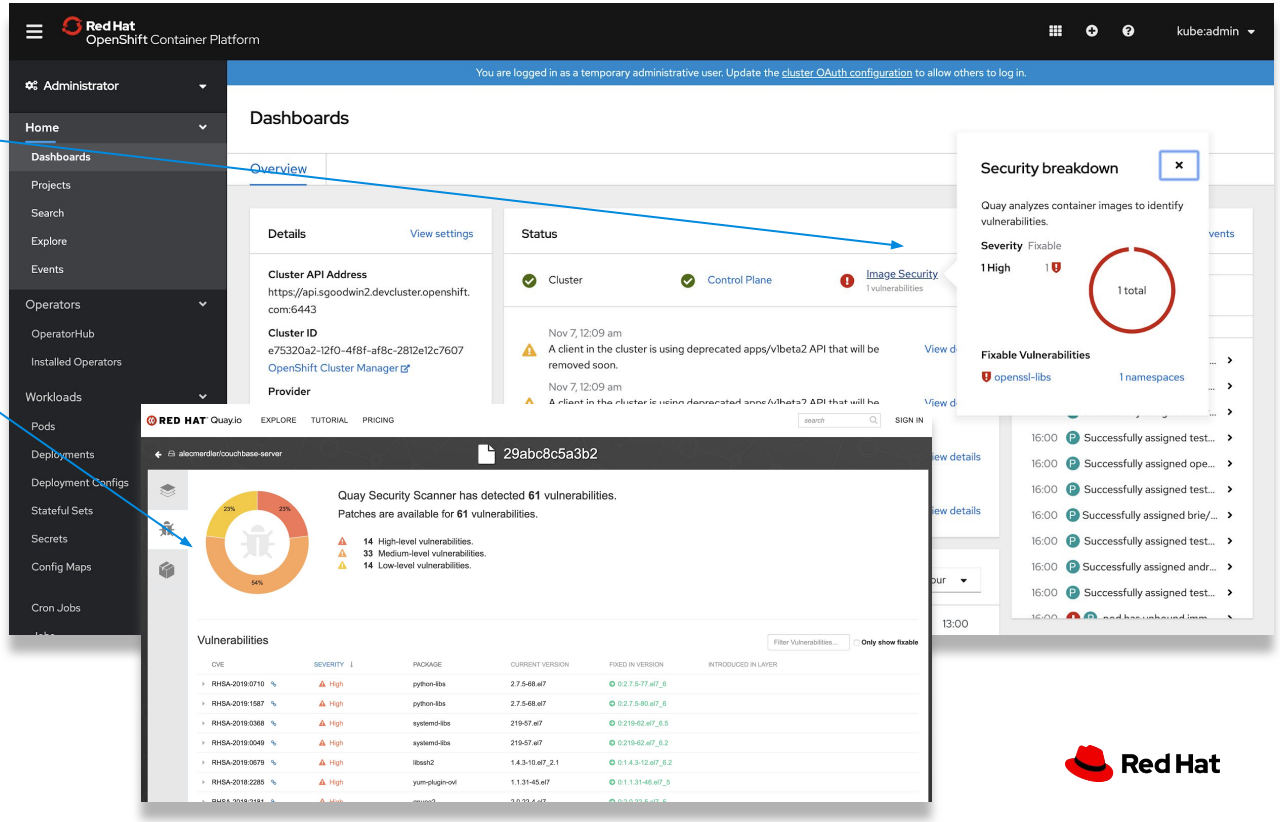


Red Hat

View Security Vulnerabilities with the Quay Operator

See all your Container Vulnerabilities right from the Console Dashboard

- Link out to **Red Hat Quay** for more in depth information
- The Quay Operator supports both **On-premise and External** Quay Registries
- Currently uses **Clair for Security Scan**; Planning to expand to other Vendors(TwistLock, Aqua, e.g.)
- *Only works for images managed by Quay*



Red Hat

Openshift Compliance Operator: Declarative Security Compliance

OpenSCAP

1 A compliance profile is selected

Describe intent with declarative config

With 4.6, a limited set of RHCOS checks will be implemented. Additional compliance checks will be delivered roughly every 2 months.

3 Accreditors or Auditors can examine the scan results for compliance status, After review, if desired, remediations can be manually applied by the cluster-admin.

2 The operator runs the scan for the profile against nodes, collect results, and (optionally) performs remeditations

Red Hat OpenShift Container Platform

ComplianceSuite

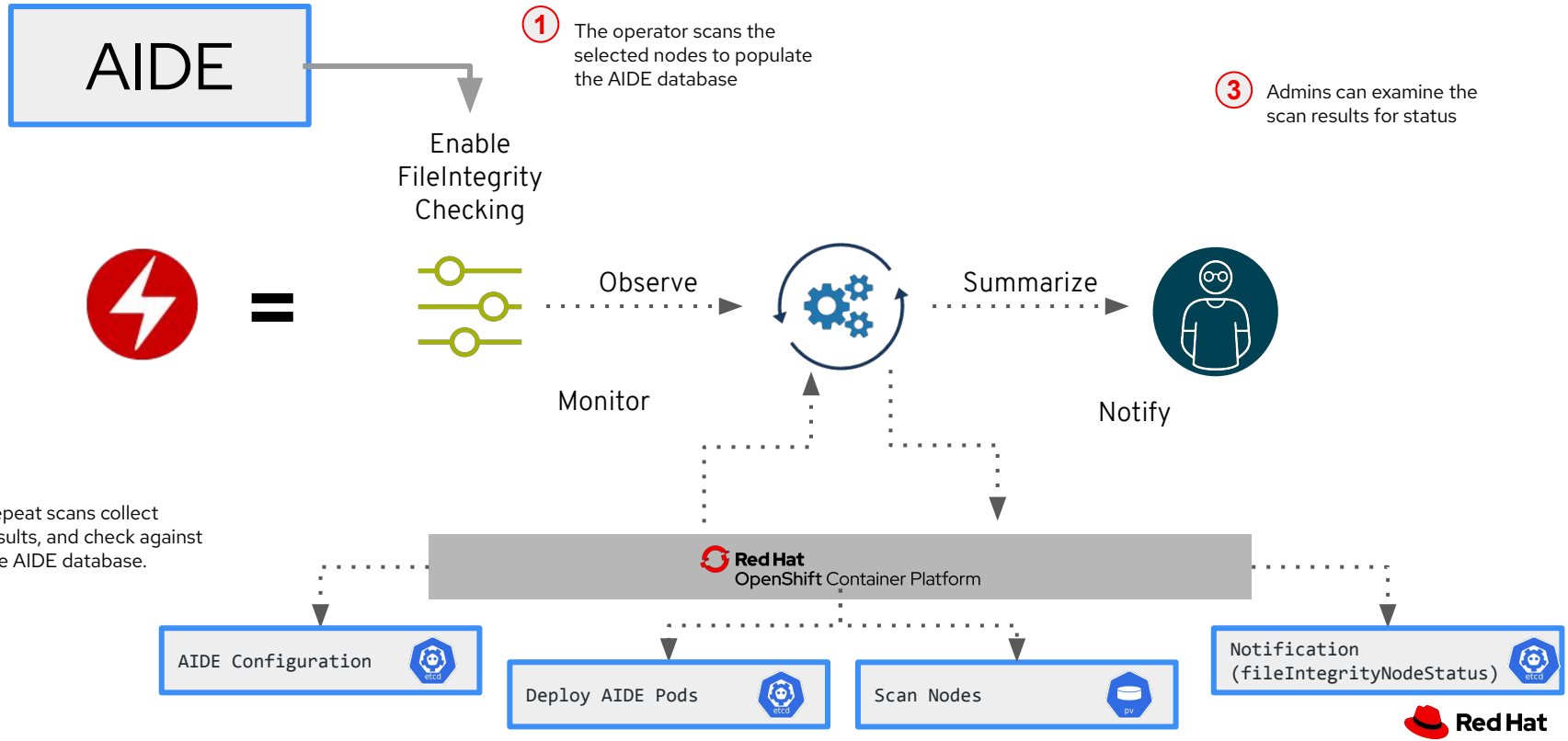
ComplianceCheckResult

Scan (results)

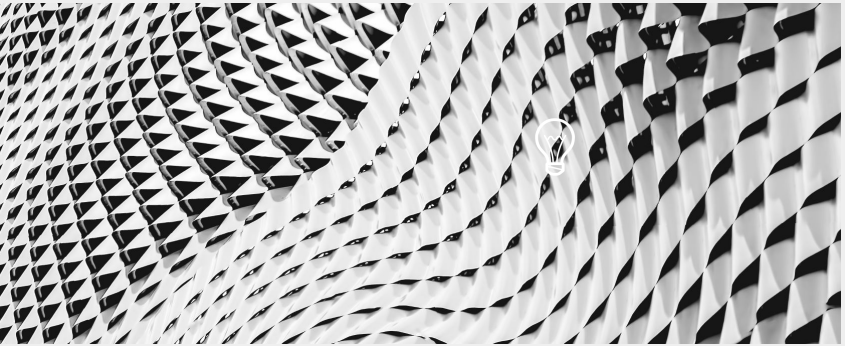
ComplianceRemediations

Red Hat

Openshift File Integrity Operator



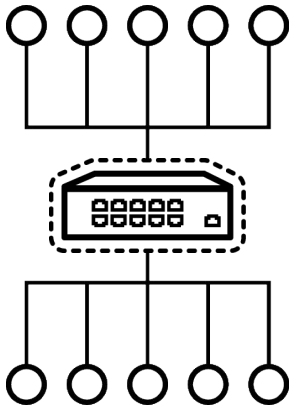
DEFEND THE NETWORK



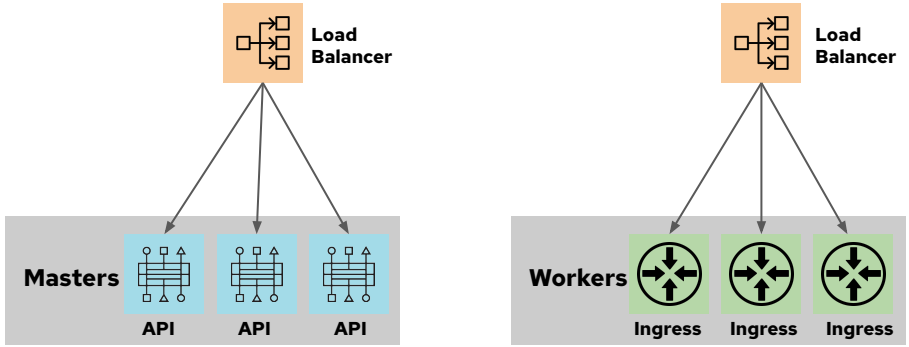
Red Hat

OpenShift networking

- Built-in internal DNS to reach services by name
- OpenShift follows the Kubernetes Container Networking Interface (CNI) plug-in model
- Software Defined Networking (SDN) for a unified cluster network to enable pod-to-pod communication
- Network Policies to isolate applications from other applications within a cluster
- Service Mesh provides advanced capabilities for managing service to service communication
- Multiple options for ingress and egress control



External Access to Cluster Resources



Ingress Traffic

- Two primary entry points into OpenShift
  - API
  - Ingress/Router
- Proper DNS entries must be configured
- Additional ingress types available
  - NodePort (requires additional port resources)
  - LoadBalancer

Ingress NodePortService

Some admins don't want OpenShift to manage a cloud load balancer and DNS for their IngressControllers.

They may want IngressControllers to be exposed through node ports to enable custom integration with a front-end load balancing solution.

The NodePortService endpoint publishing strategy publishes the Ingress Controller using a Kubernetes NodePort service, which enables:

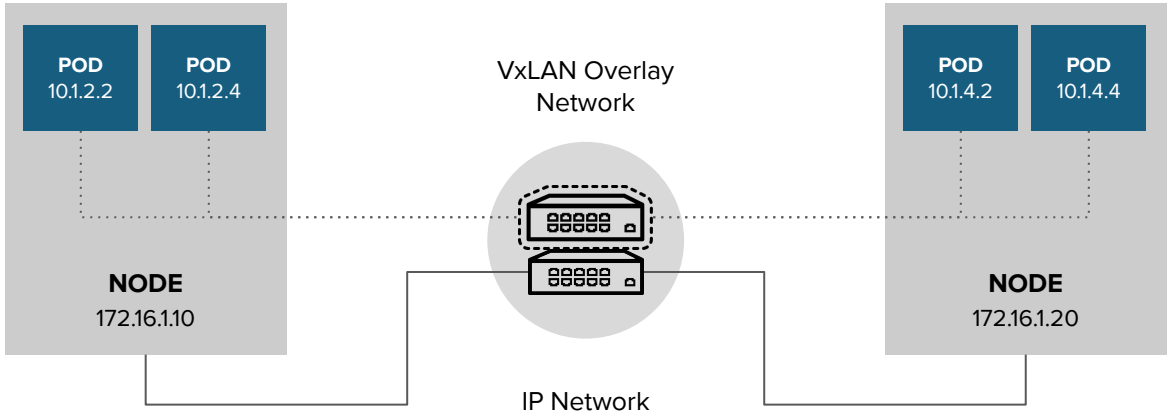
- alternate forms of ingress for better support of bare metal deployments
- use cases requiring more than one instance of the HAProxy router per node

```yaml
apiVersion: v1
kind: Service
metadata:
  name: router-default
  namespace: openshift-ingress
  annotations:
    operator.openshift.io/node-port-service-for: default
spec:
  type: NodePort
  externalTrafficPolicy: Local
  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: http
  - name: https
    port: 443
    protocol: TCP
    targetPort: https
  selector:
    ingresscontroller.operator.openshift.io/deployment-ingresscontroller: default
```

Routes vs Ingress

Feature	Ingress	Route
Standard Kubernetes object	X	
External access to services	X	X
Persistent (sticky) sessions	X	X
Load-balancing strategies (e.g. round robin)	X	X
Rate-limit and throttling	X	X
IP whitelisting	X	X
TLS edge termination	X	X
TLS re-encryption	X	X
TLS passthrough	X	X
Multiple weighted backends (split traffic)		X
Generated pattern-based hostnames		X
Wildcard domains		X

OpenShift pod to pod networking

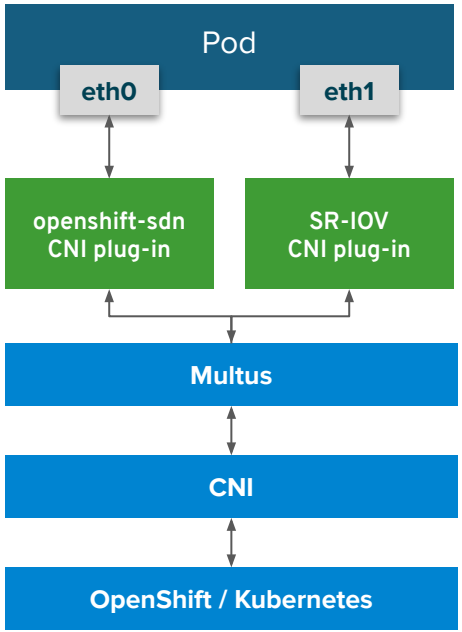


Red Hat

Multus

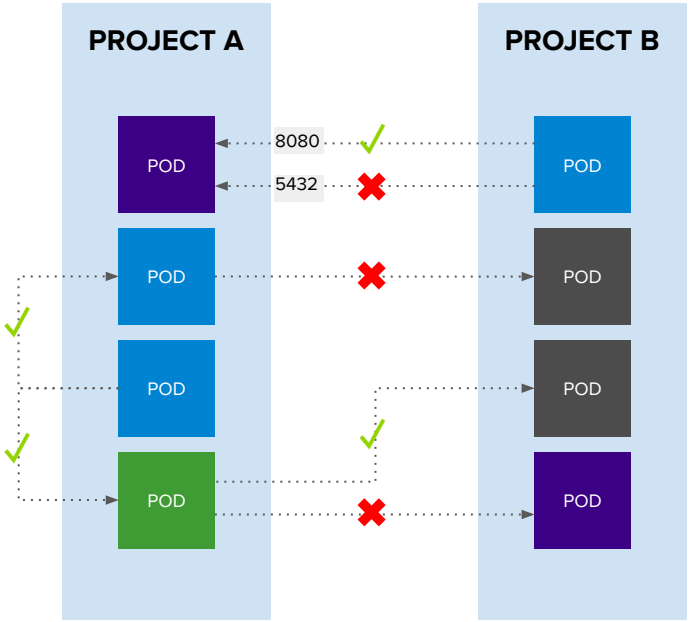
- Problem: Kubernetes only supports one network interface, “eth0”, but we need:
  - Functional separation of control/data planes
  - Link aggregation for network redundancy
  - Different network protocol stacks, capabilities, SLAs
  - Traffic isolation / Network segregation and security
  - QoS
- Solution: 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



OpenShift SDN

Network policy enabled by default



Example Policies

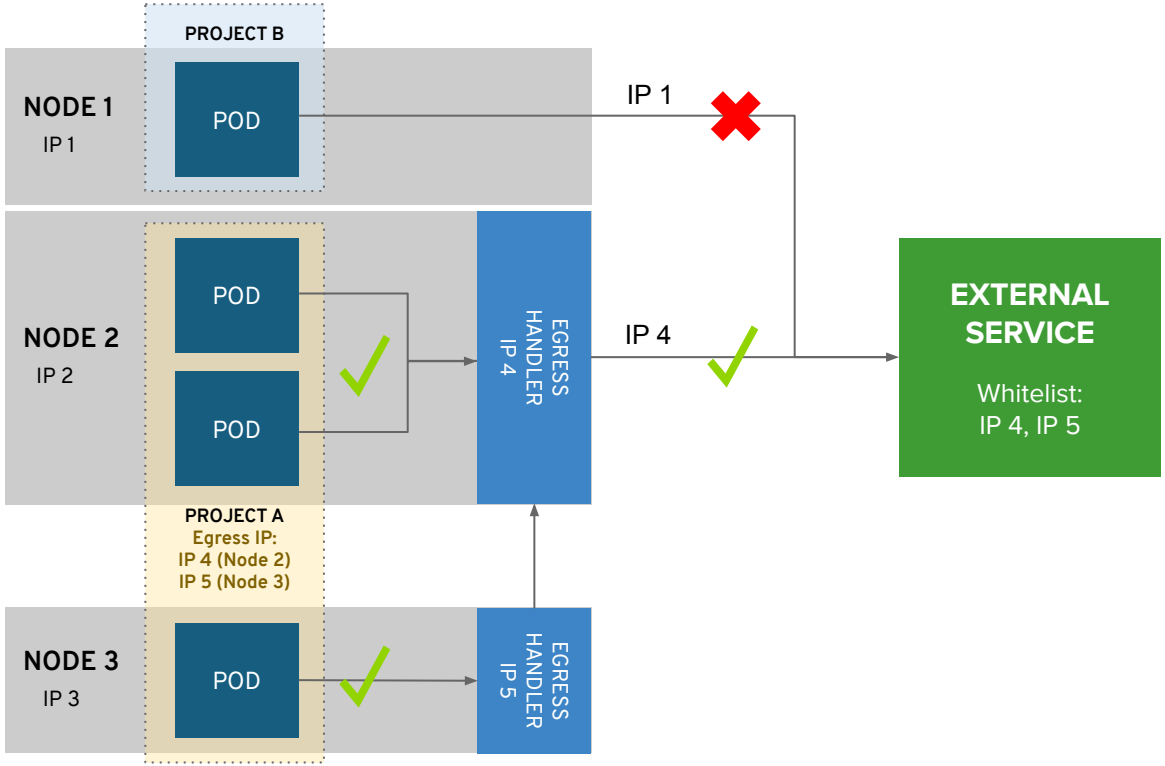
- Allow all traffic inside the project
- Allow traffic from green to gray
- Allow traffic to purple on 8080

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


Red Hat

Controlling Egress Traffic

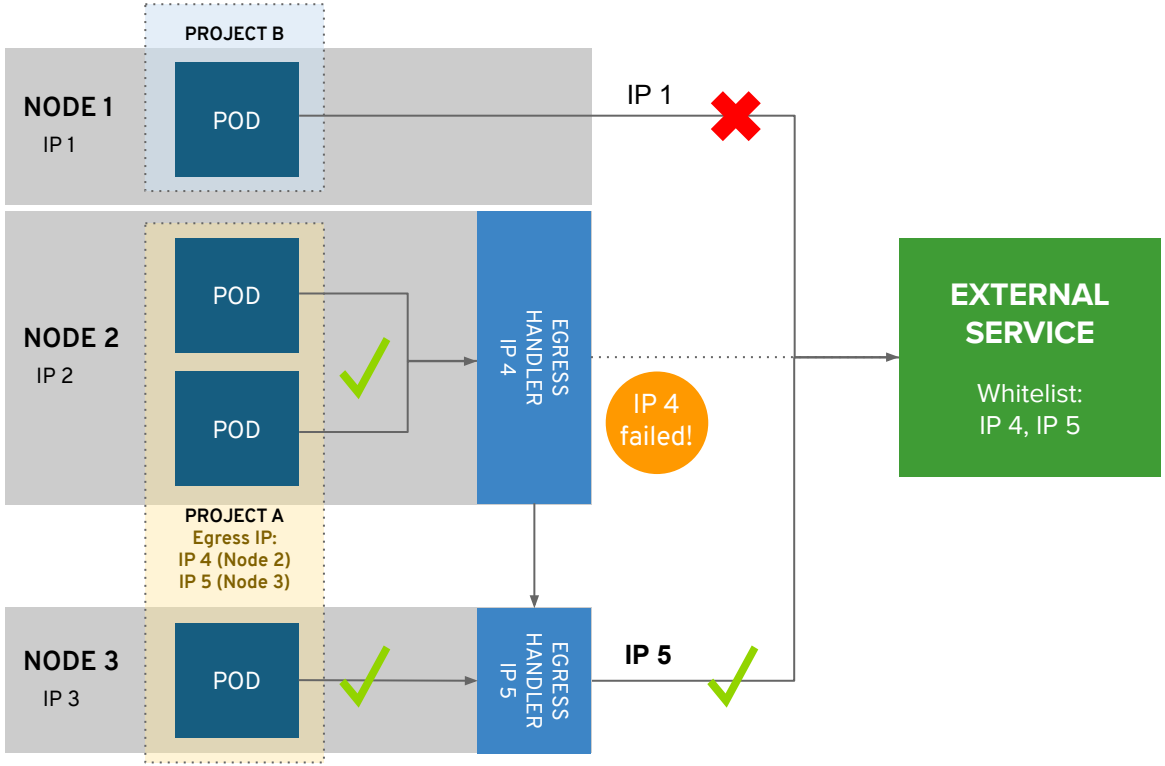
Egress IP high availability (multiple IPs)



Red Hat

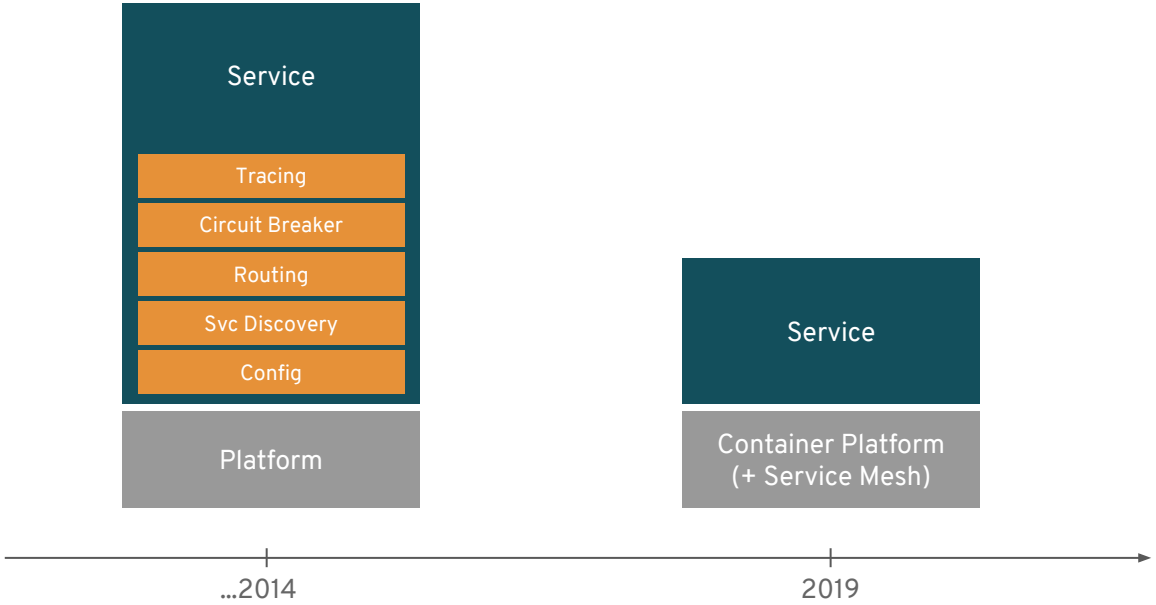
Controlling Egress Traffic

Egress IP high availability (multiple IPs)



Red Hat

Microservices evolution

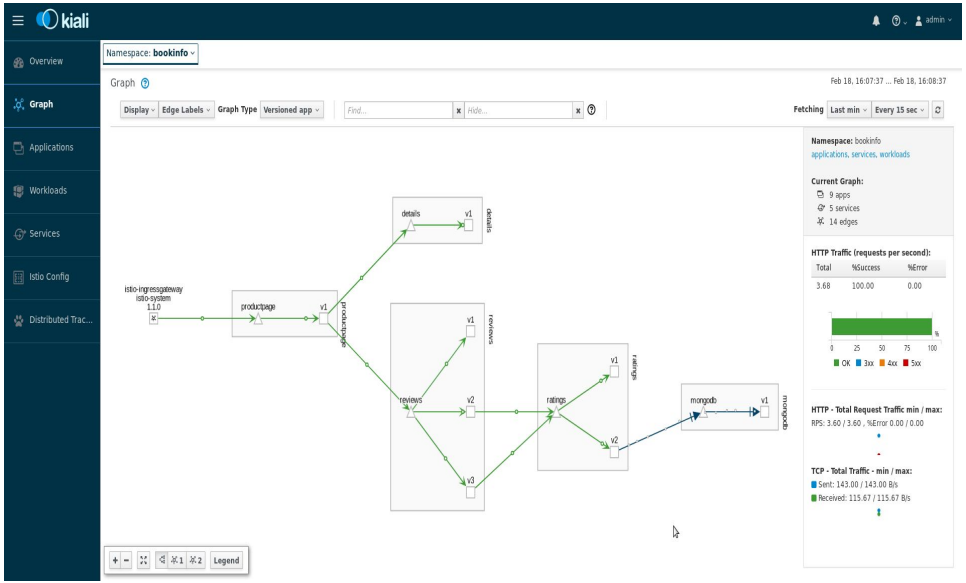


Red Hat

Secure microservices with Service Mesh

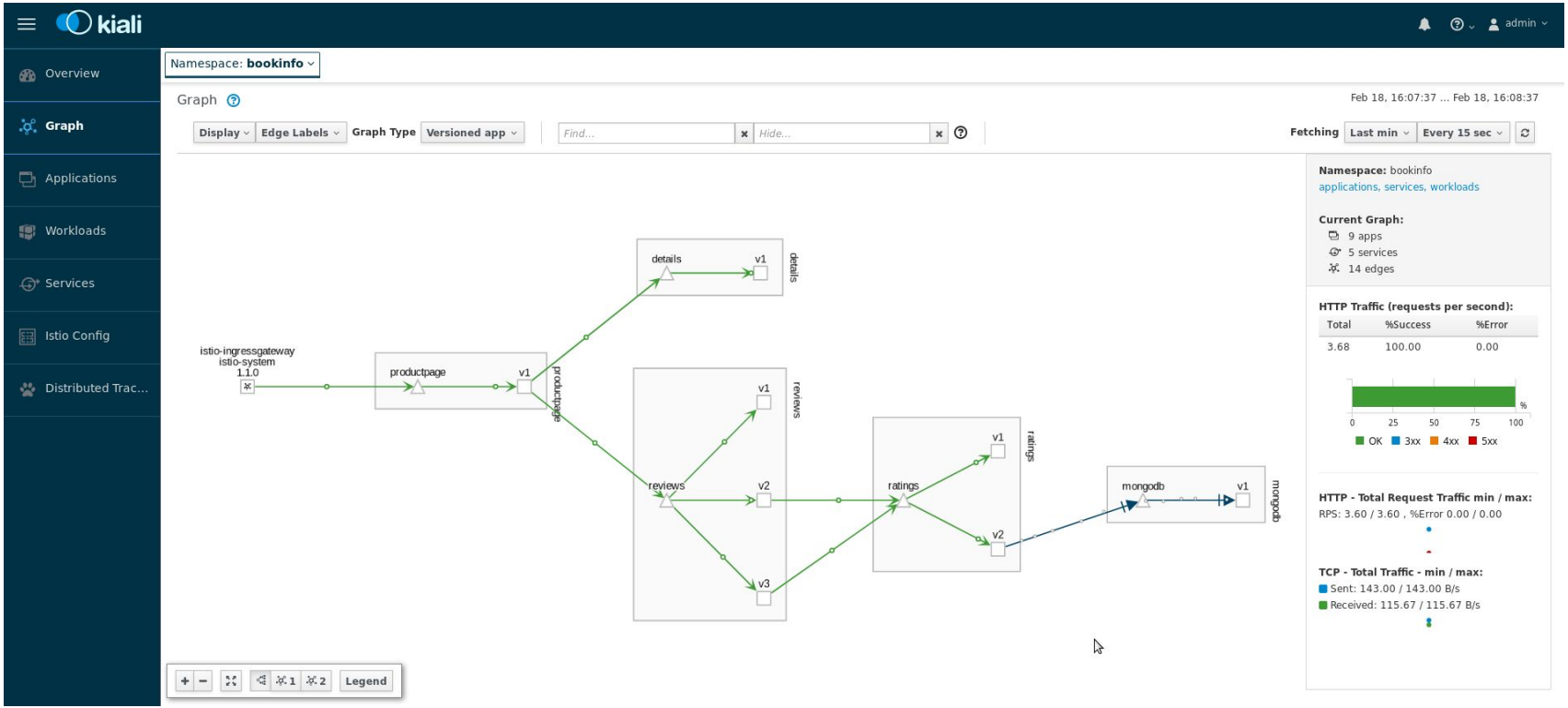
Key Features

- A dedicated network for service to service communications
- Observability and distributed tracing
- Policy-driven security
- Routing rules & chaos engineering
- Powerful visualization & monitoring
- Will be available via OperatorHub



Red Hat

Observability with Kiali

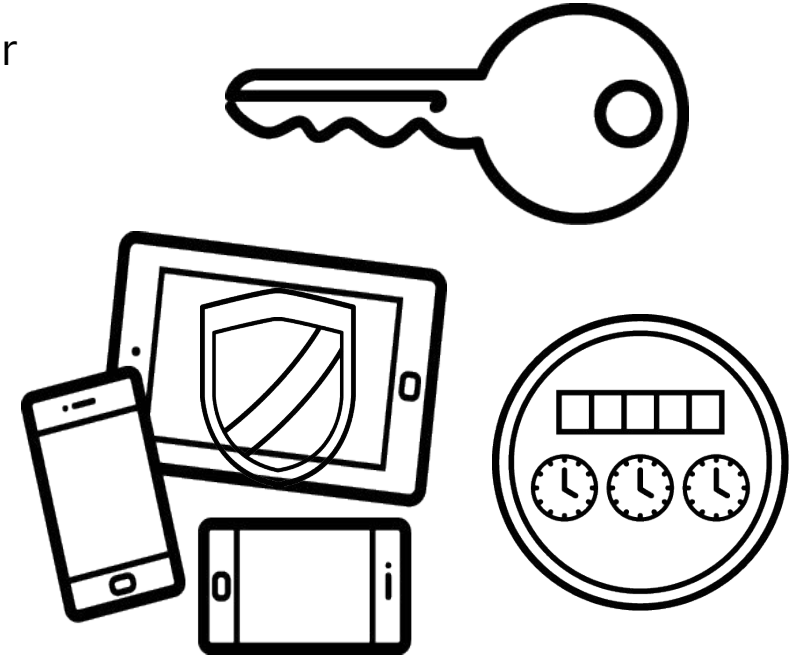


Red Hat

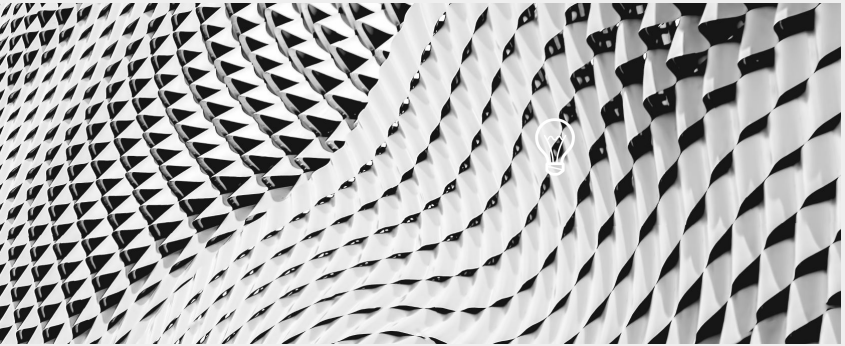
Application API management

Consider configuring an API gateway for container platform & application APIs

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



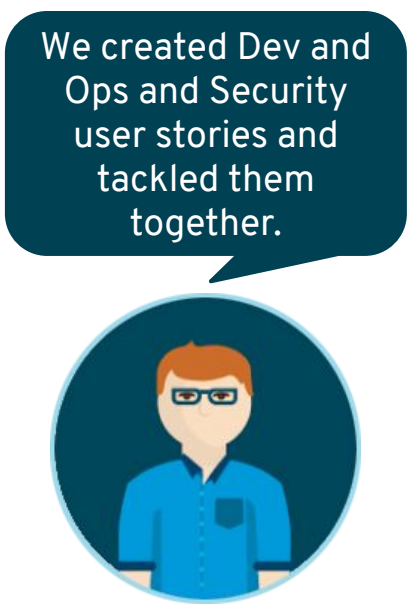
CONTROL APPLICATION SECURITY



Red Hat

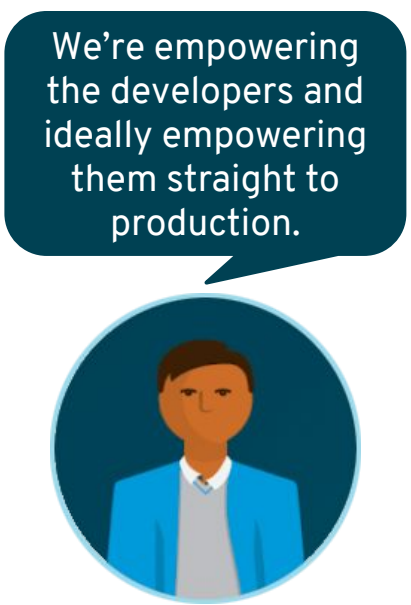
DEVSECOPS

THROUGH THE ADOPTION OF CONTAINERS



DEVELOPER

SECURITY



OPERATIONS

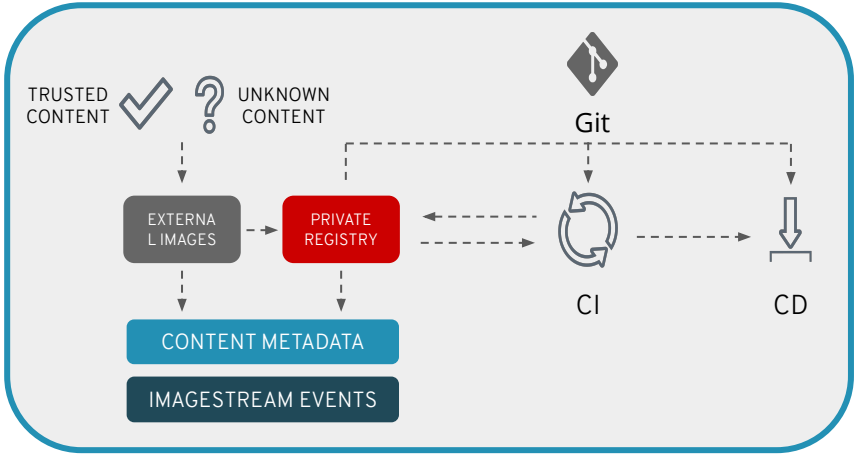
[DevNation Federal - The Journey to DevSecOps]

Securing Containerized Applications

An opportunity to shift security left

Best practices

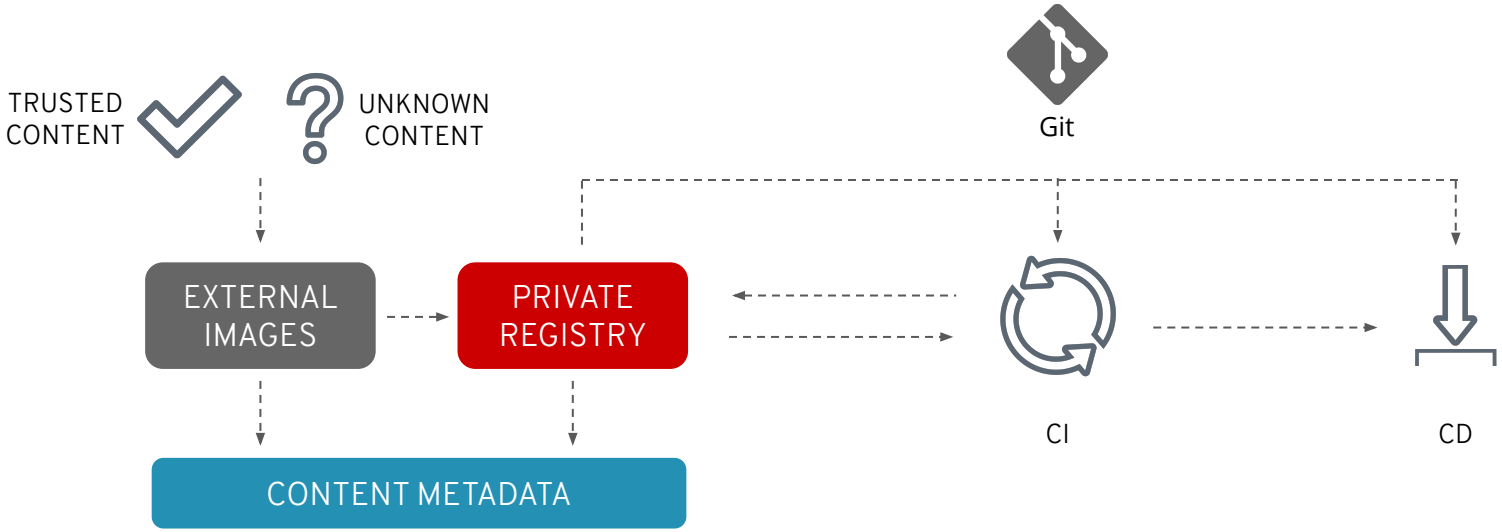
- Use trusted sources for external content
- Use a private registry to manage images
- CI/CD must have security gates
- Application secrets management
- Apply runtime security policies
- Rebuild and redeploy - never patch a running container
- Ensure application logging, monitoring



Red Hat

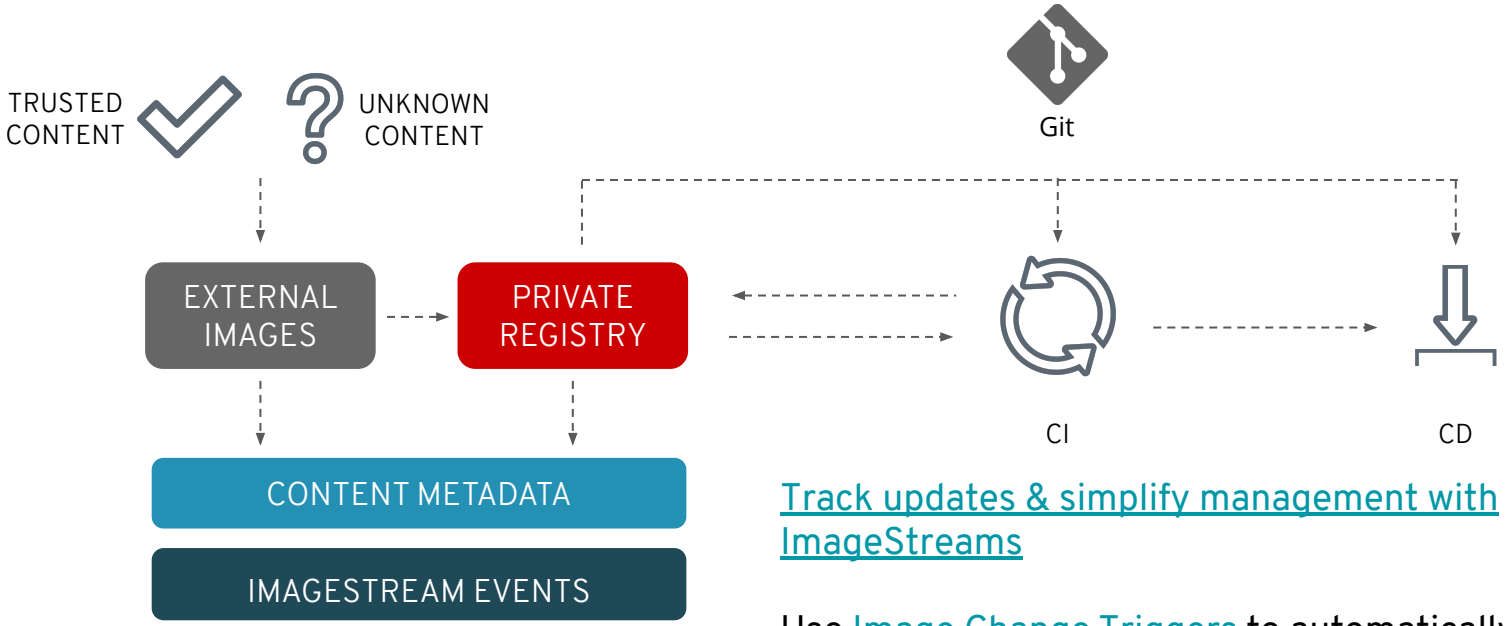
Secure & Automate The Content Lifecycle

Elements of the Openshift container pipeline



Red Hat

Trust is temporal: rebuild and redeploy as needed



Track updates & simplify management with ImageStreams

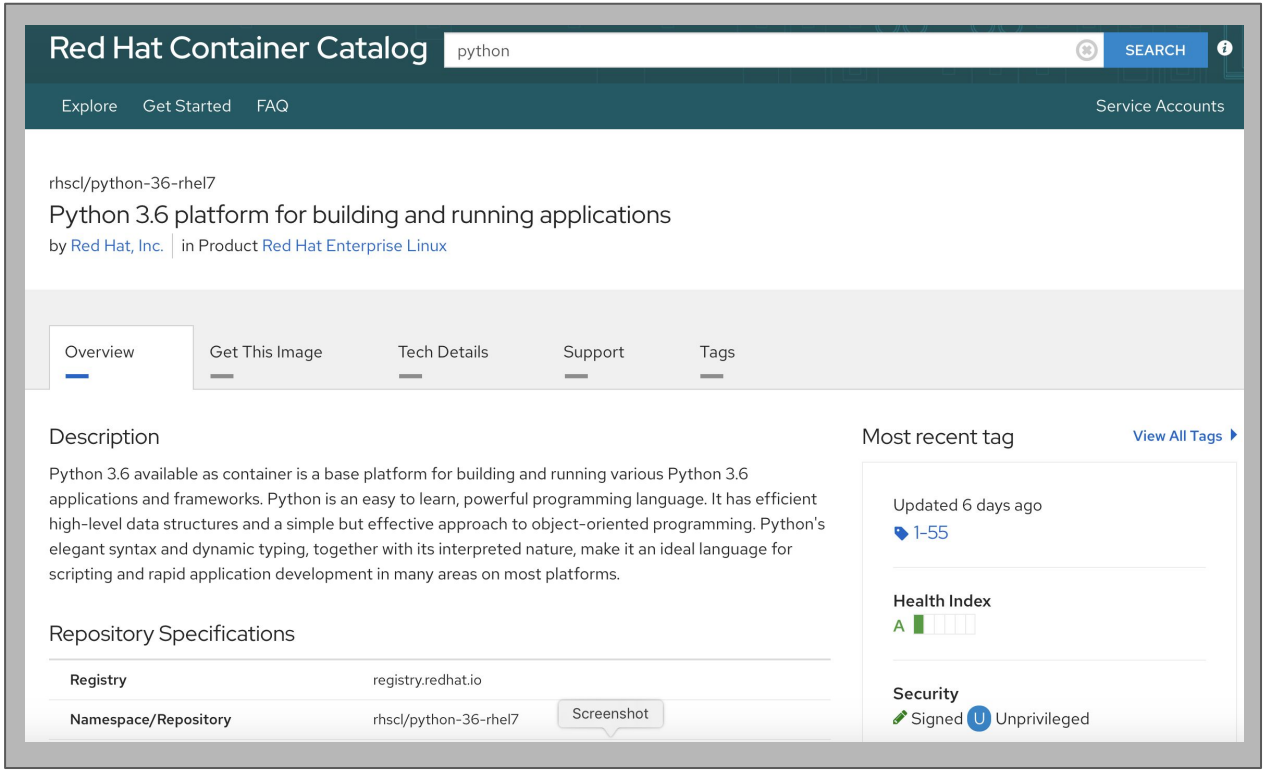
Use Image Change Triggers to automatically rebuild custom images with updated (patched) external images

Red Hat

External Content: Use Trusted Sources

Red Hat Container Images

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

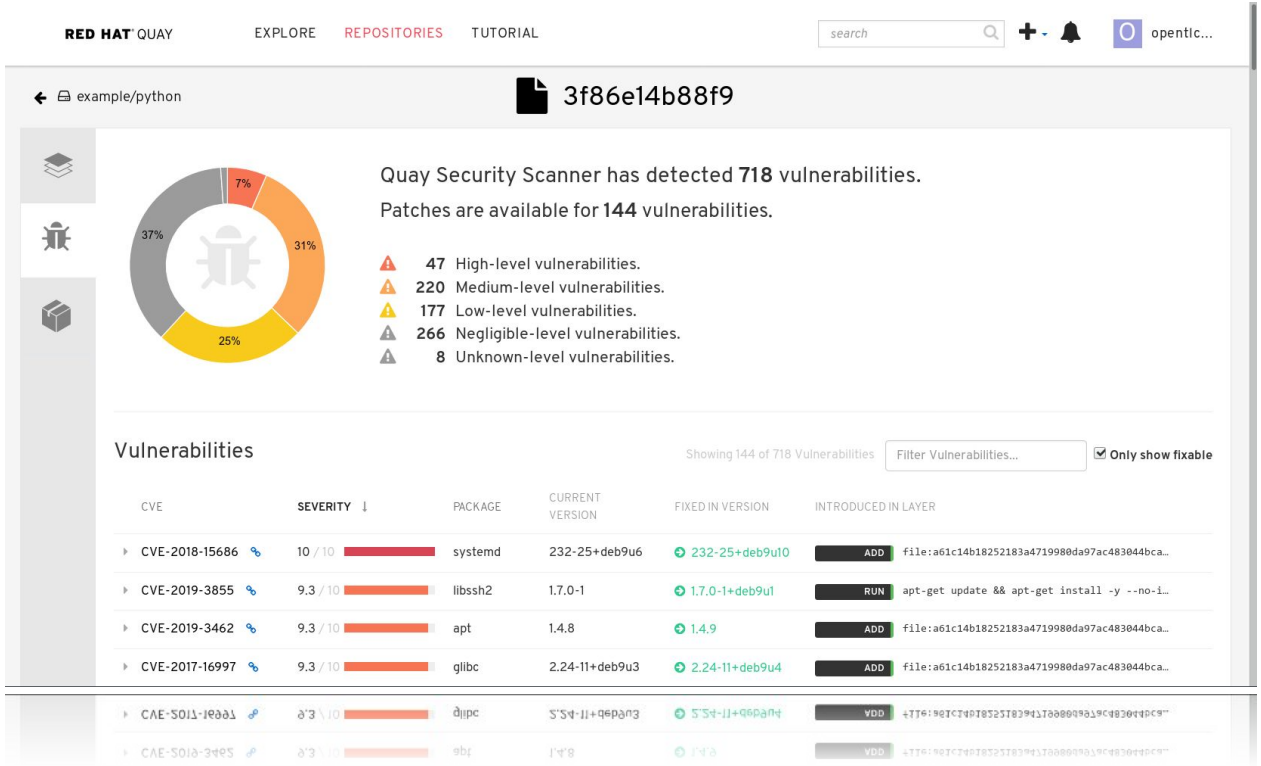


*More about the Health Index

Red Hat

Red Hat Quay
Enterprise Container Registry

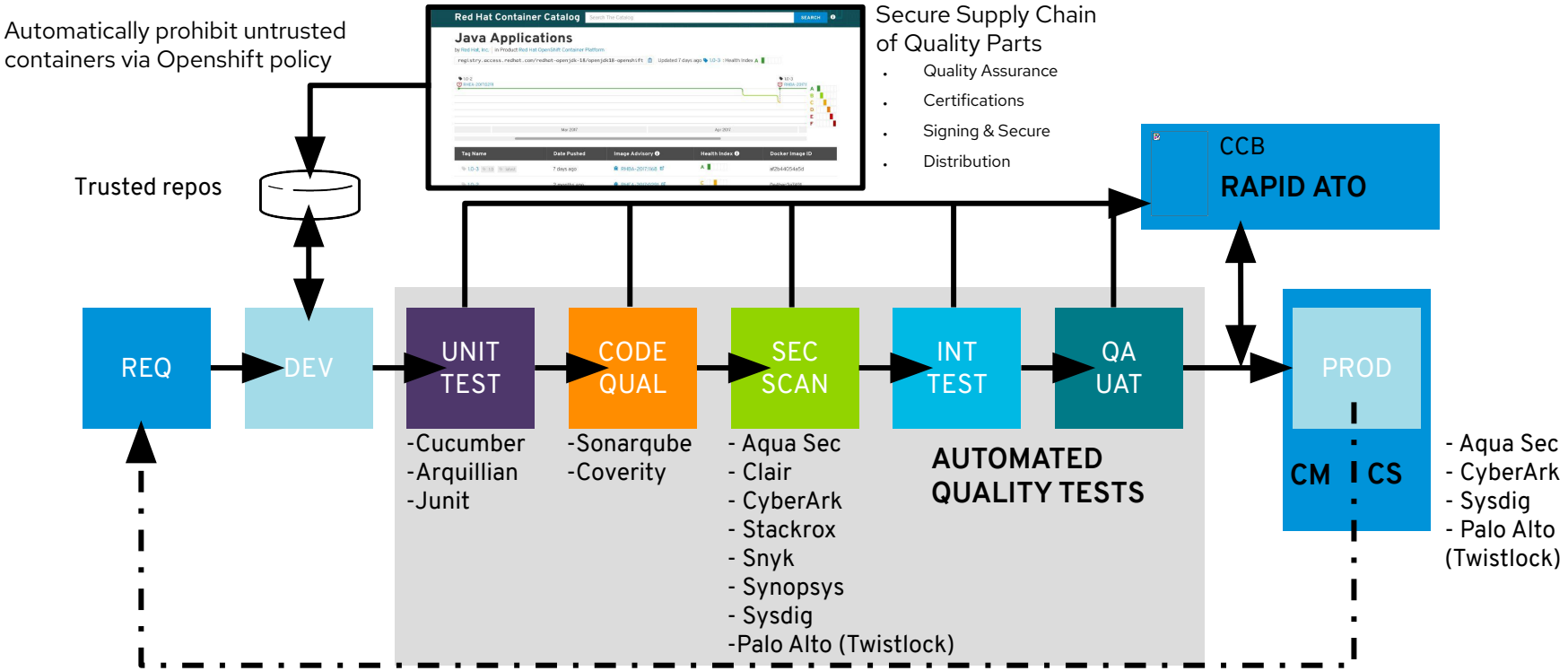
- Offered as self-managed and as-a-service
- Vulnerability Scanning (Clair)
- Geographic Replication
- Build Image Triggers
- Image Rollback with Time Machine



Red Hat

Integrate Security in your CI/CD Pipeline

Automated quality and security: because you can't inspect quality into a product



[Red Hat Container Catalog: Java Applications]

[More about the Container Health Index]

Red Hat

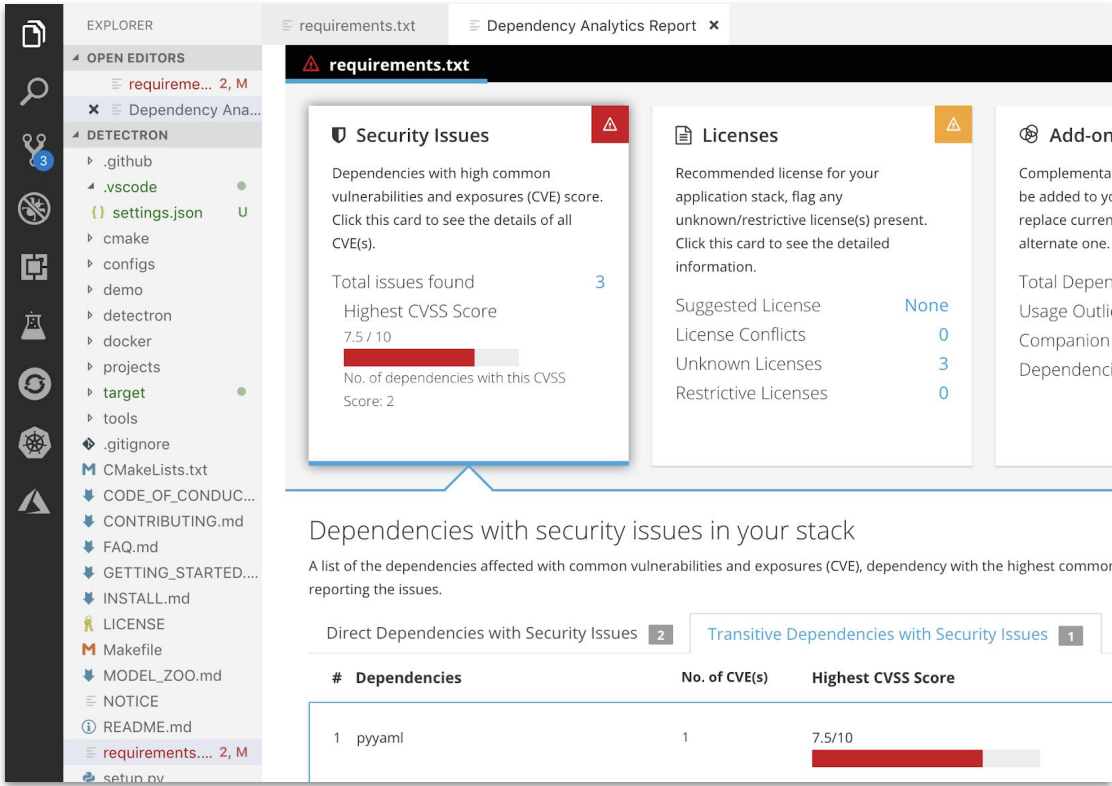
Plugins available for VSCode and Eclipse IDEs

Enhancing Secure Application Development and DevSecOps

"Shift Left" - find CVEs and license issues during development

Red Hat Dependency Analytics IDE plugins provide security and license warnings for any project dependency:

- Be notified of CVEs in any package or sub-package
- Remediation advice (upgrade / downgrade)
- Uses open source and Snyk CVE databases
- Supported for Java, Node, Python

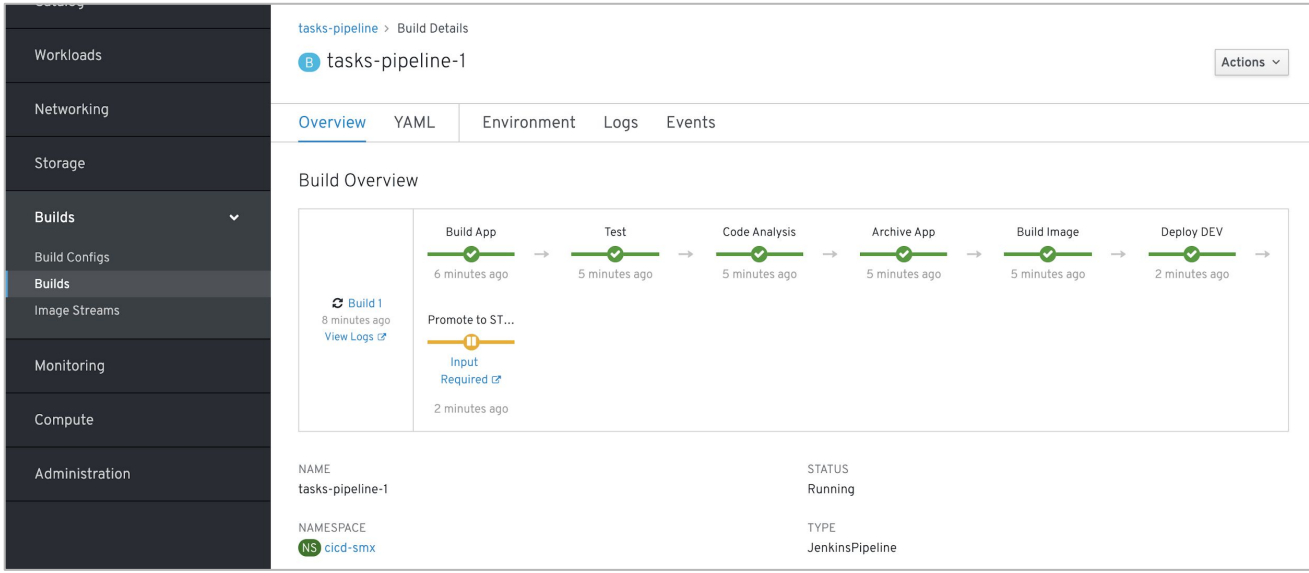


Jenkins CI/CD, run in OpenShift and deploy to OpenShift

Jenkins is still the most used CI/CD platform in enterprises and can be used from inside OpenShift.

An intuitive pipeline visualization makes it simple for users to see how builds are progressing.

The full Jenkins UI is also available.



Why? Build in, or for, OpenShift from your enterprise CI/CD system.

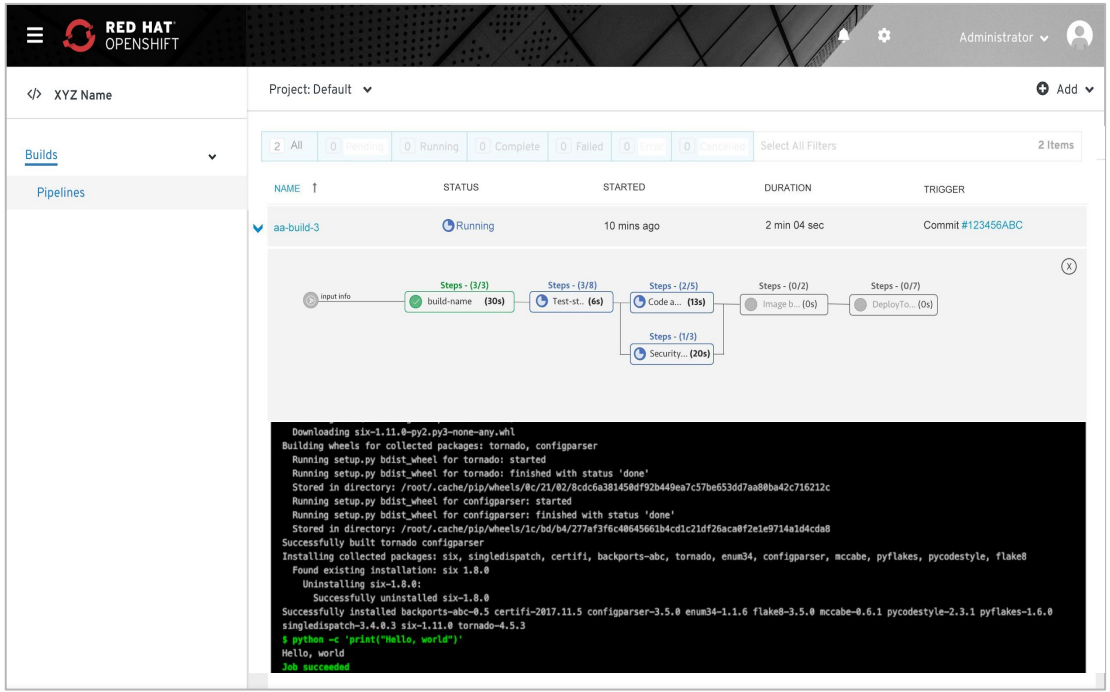
Red Hat

OpenShift Pipelines: A Kubernetes-native CI/CD platform

Provides a next-gen Kubernetes CI/CD pipeline that works for containers (including serverless).

Based on the Tekton project (which was spun out of the Knative Pipelines project) started by Google, Red Hat and others.

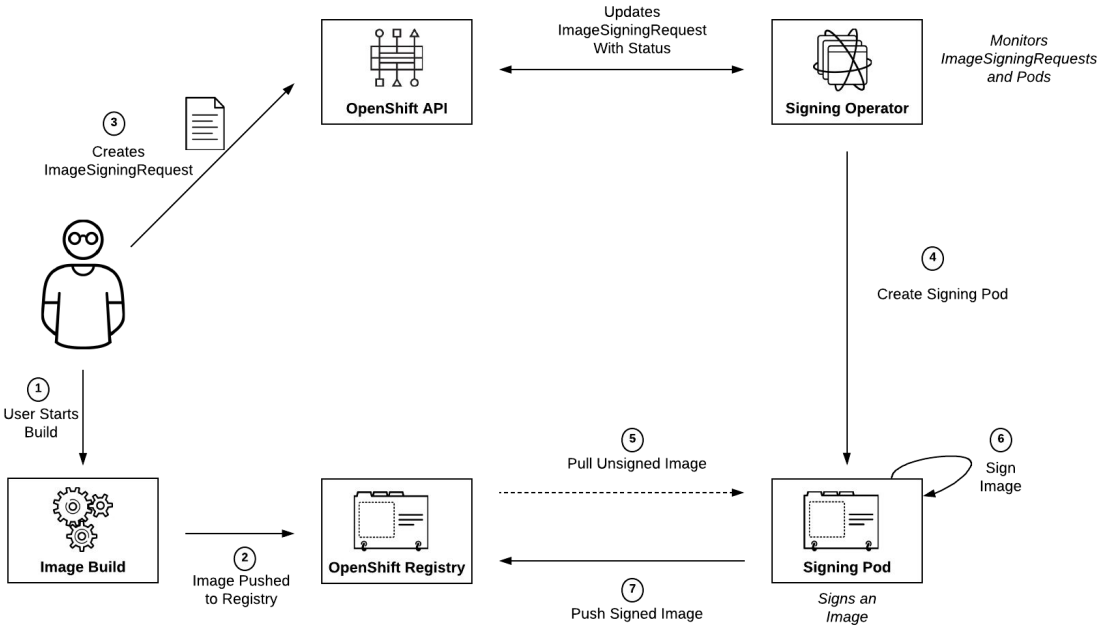
Target general availability in OpenShift 4.7.



Why? A faster, less resource-intensive CI/CD platform that's Kubernetes-native.

Container Signing

A simplified and automated approach to signing container images

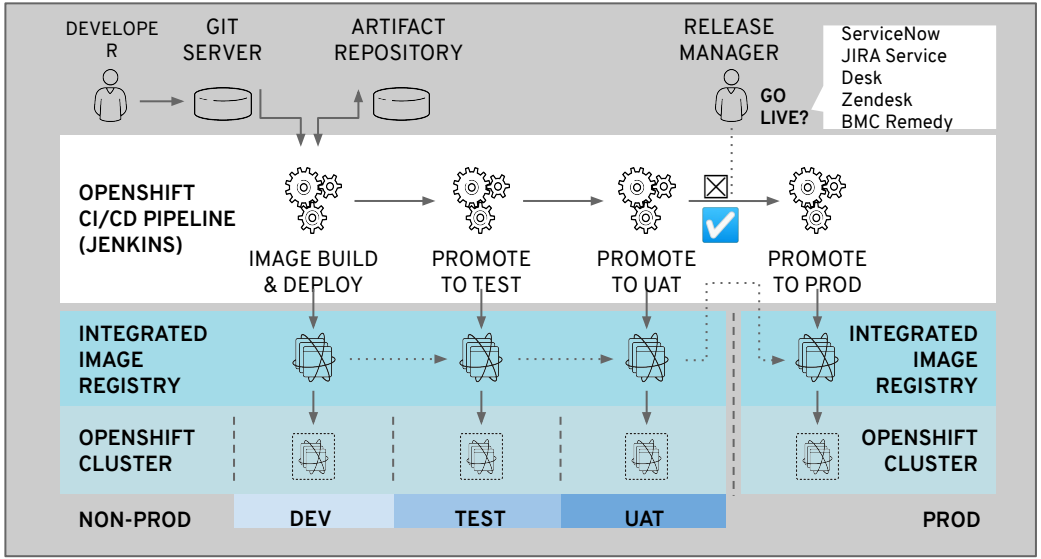


Red Hat

Summary

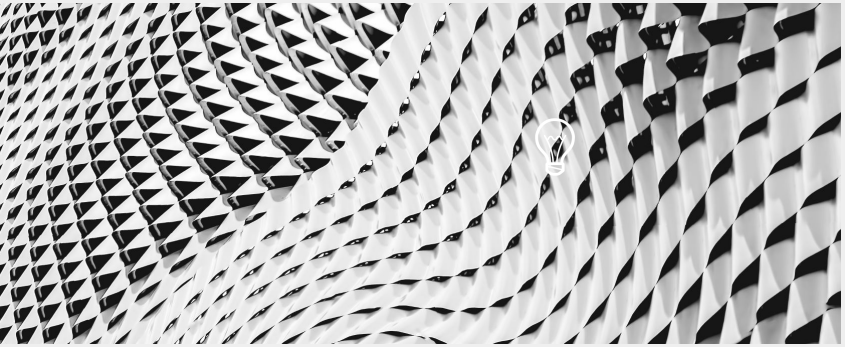
Managing container deployment

- 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

EXTEND SECURITY



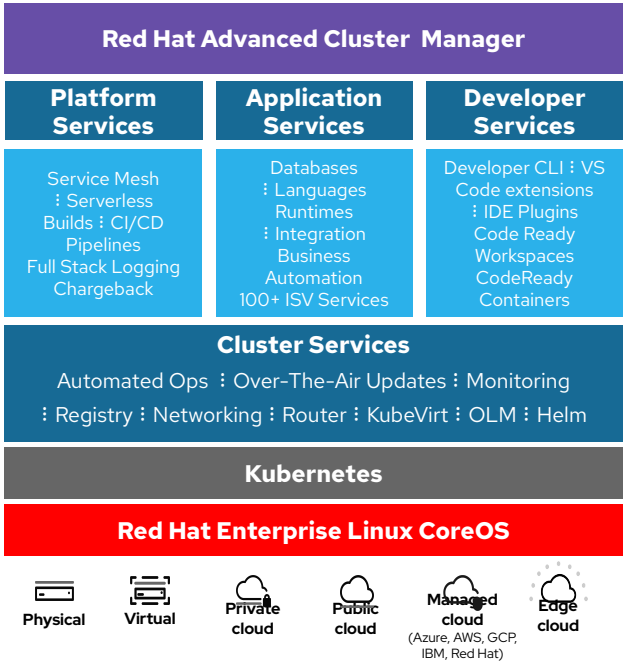
Red Hat

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)

Red Hat OpenShift certified operators - Security



Red Hat

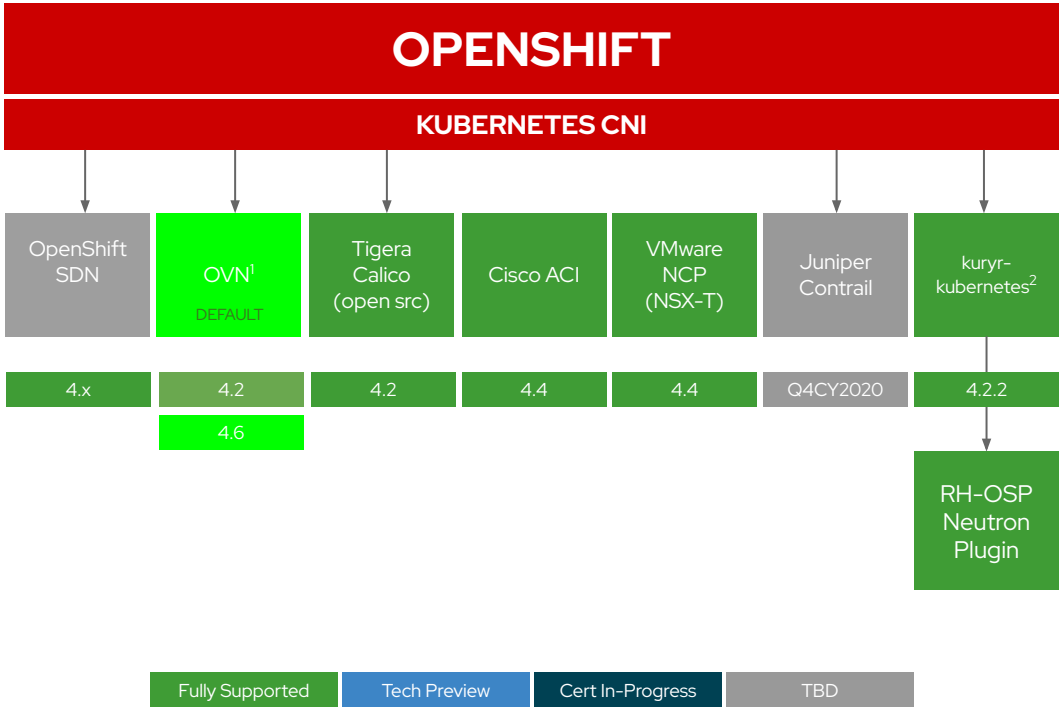
3rd-Party CNI Plug-in Certification Update

The following 3rd-party Kubernetes CNI plug-ins have begun the OpenShift certification process and are at varying stages of progress:

- cisco ACI
- vmware NSX-T

The certification process primarily consists of:

1. Formalizing the partnership
2. Certifying the container(s)
3. Certifying the Operator
4. Successfully passing the same Kubernetes networking conformance tests that OpenShift uses to validate its own SDN



Generally Available

Red Hat

Red Hat
Advanced Cluster Security
for Kubernetes

StackRox | Red Hat ACS

Kubernetes is the standard for application innovation…

- Microservices architecture
- Declarative definition
- Immutable infrastructure

…and Kubernetes-native security is increasingly critical

- Secure supply chain
- Secure infrastructure
- Secure workloads

DevOps → DevSecOps ← Security

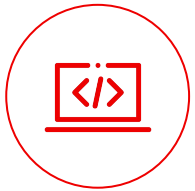
Benefits of a Kubernetes-native approach to security

Lower operational cost

DevOps and Security teams can use a common language and source of truth

Reduce operational risk

Ensure alignment between security and infrastructure to reduce application downtime



Increase developer productivity

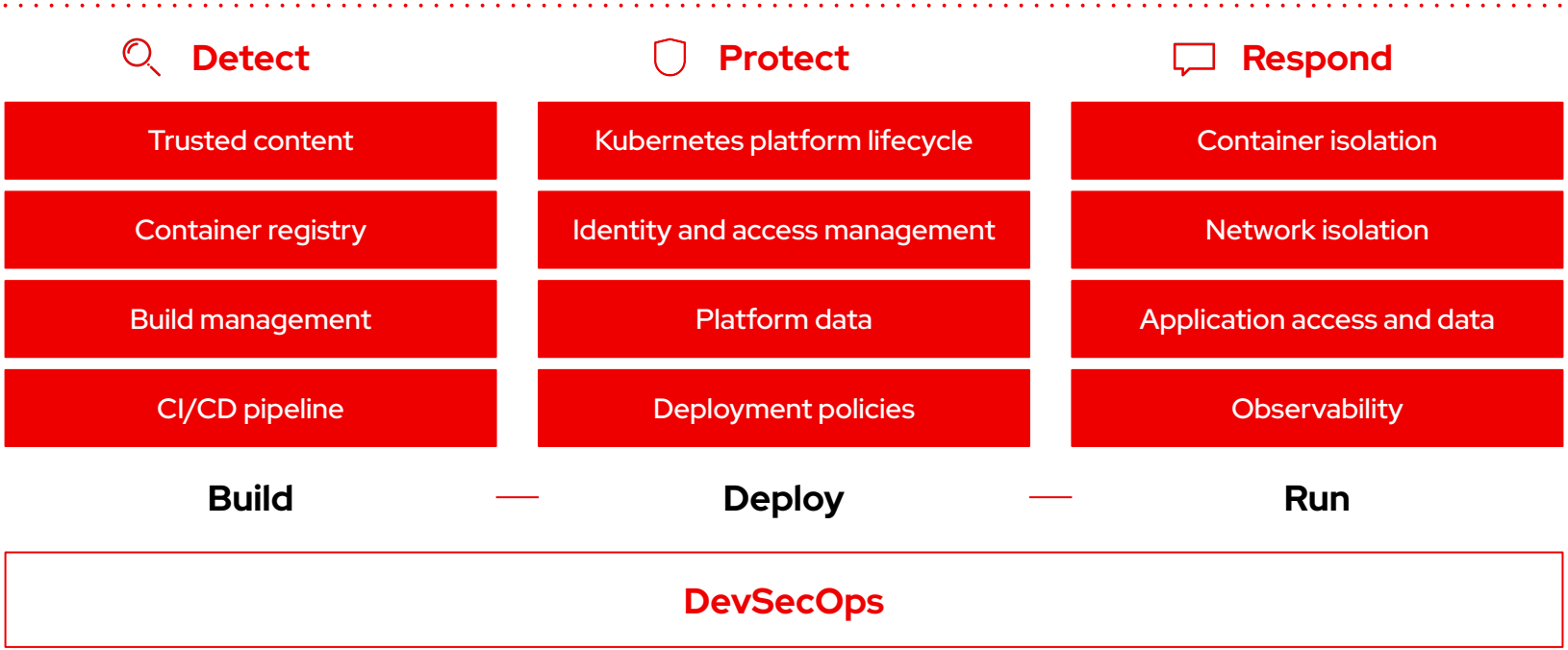
Leverage Kubernetes to seamlessly provide guardrails supporting developer velocity

Red Hat Advanced Cluster Security for Kubernetes

A cloud workload protection platform and cloud security posture management to enable you to “shift left”

Shift left	Cloud security posture management (CSPM)	Cloud workload protection (CWPP)
Secure supply chain	Secure infrastructure	Secure workloads
Extend scanning and compliance into development (DevSecOps)	Leverage built-in Kubernetes CSPM to identify and remediate risky configurations	Maintain and enforce a “zero-trust execution” approach to workload protection

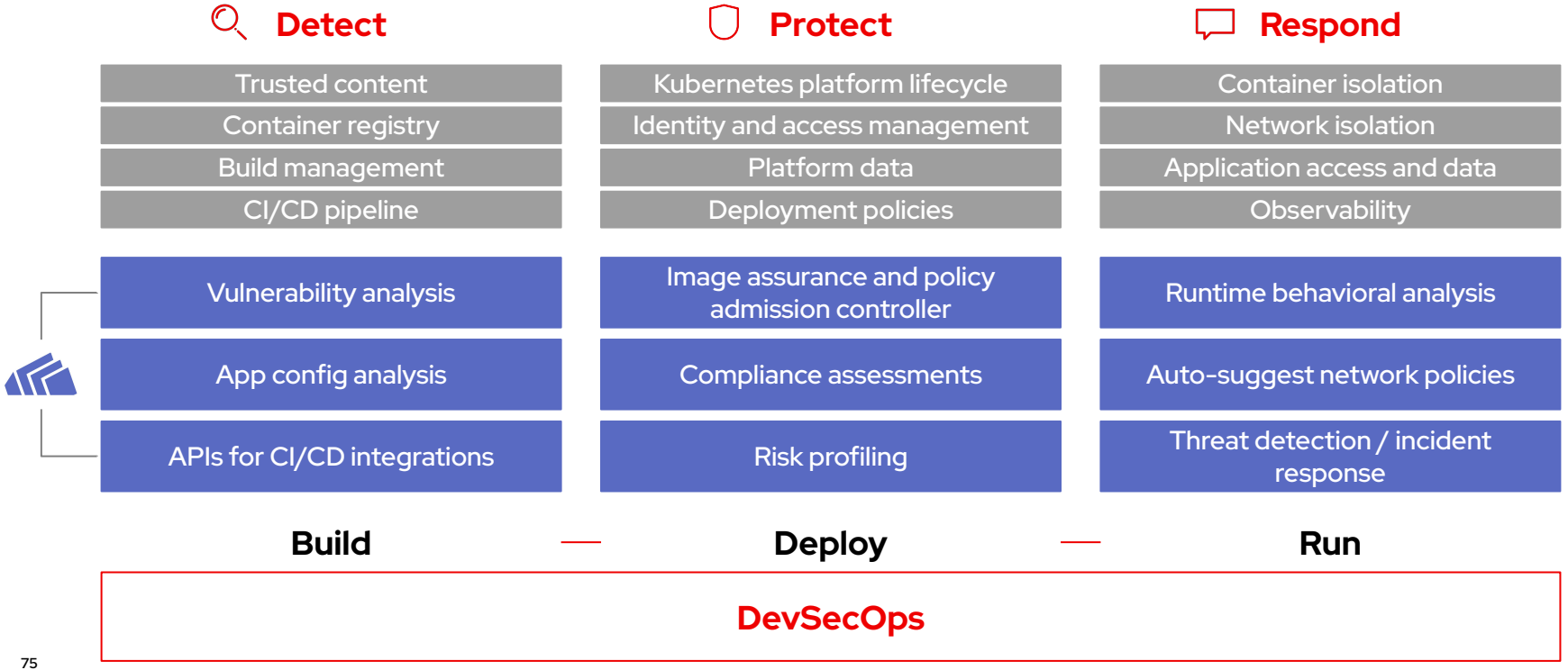
Red Hat OpenShift provides a secure foundation



A layered approach to container and Kubernetes security

Red Hat

RHACS delivers security depth to entire application lifecycle



RHACS

Securing Kubernetes Deployments

It's all about the Application in the container

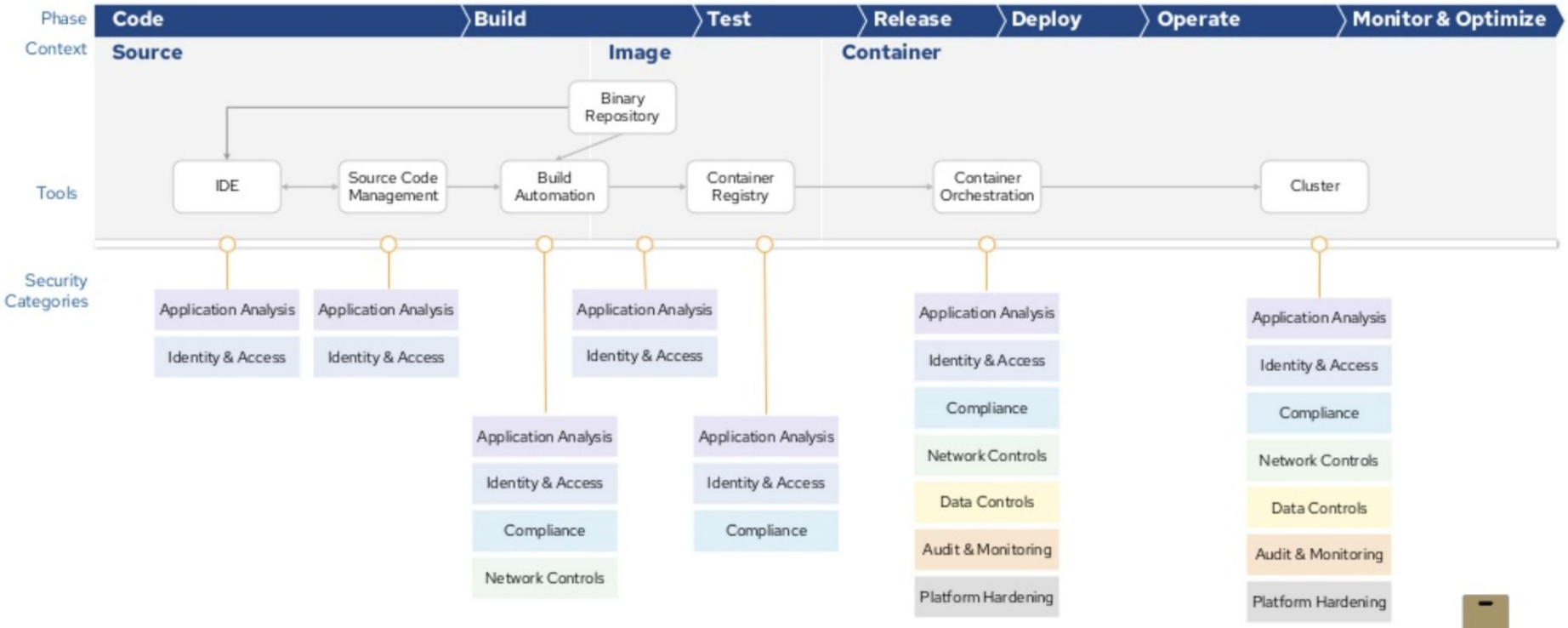
- plus a Registry Scanner.

It's not:

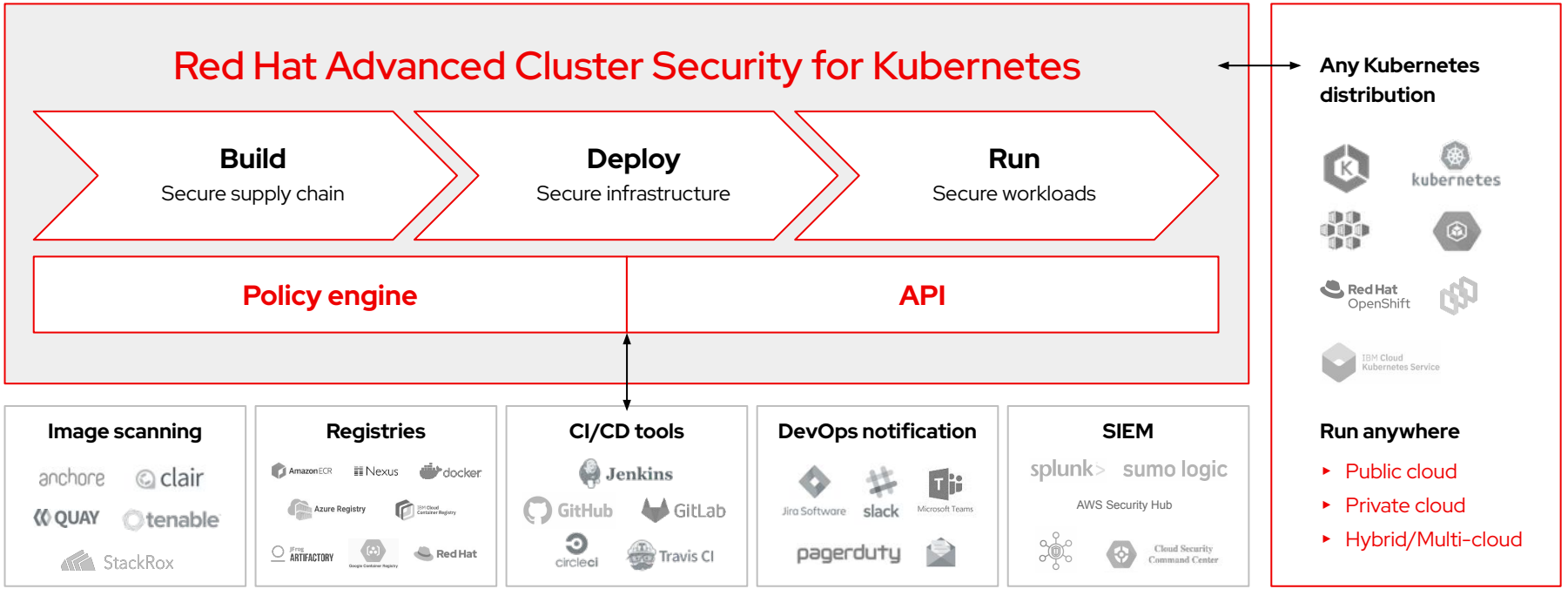
- End 2 End Monitoring -> Dynatrace
- Infrastructure Monitoring - RHACM
- Infrastructure Compliance Monitoring - RHACM
- Access Control / Audit to and in OpenShift IBM QRadar or CyberARC
- SIEM Solution -> Splunk
- Certificate Management - Cert Manager
- API Management - 3scale
- Application Performance Management
- Registry - QUAY
- Service MESH

RHACS

adding security to dev ops for your kubernetes native applications

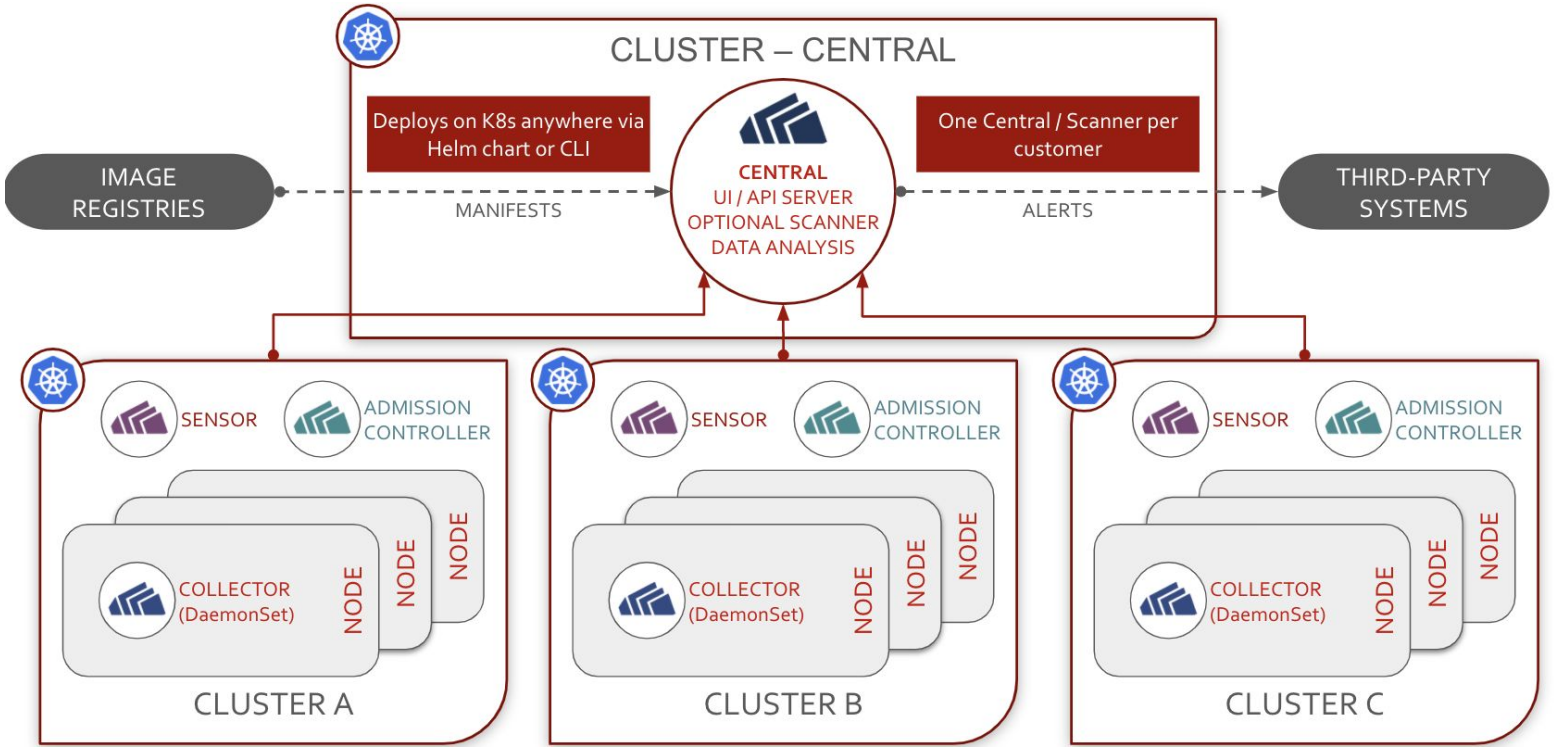


The first Kubernetes-native security platform



Red Hat

Architecture



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