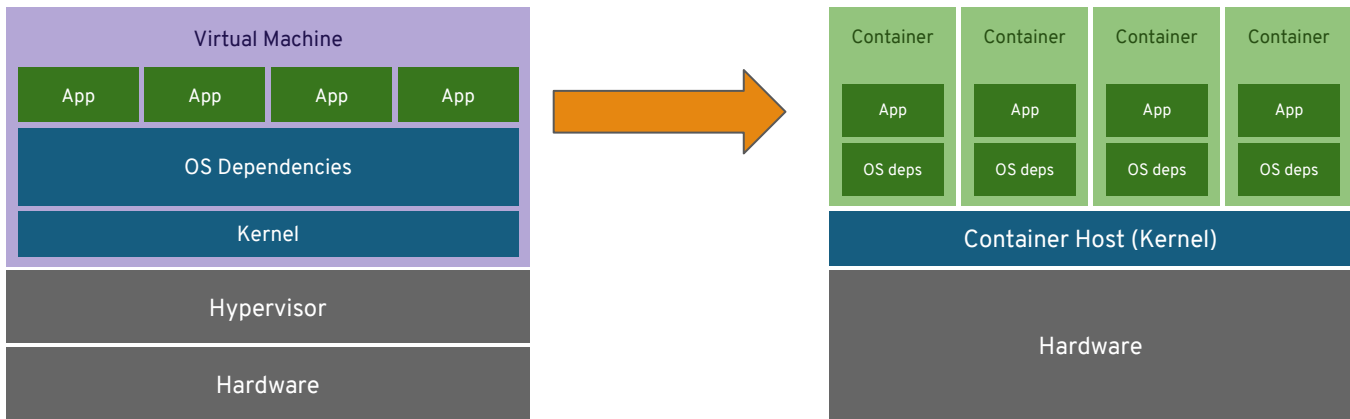


Securing A Multitenant Kubernetes Cluster

Kirsten Newcomer
Senior Principal Product Manager, OpenShift

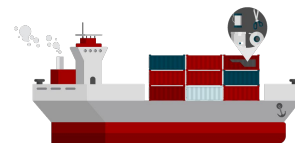
CONTAINERS ARE THE NEW WAY TO DELIVER APPLICATIONS



VMs virtualize the hardware

Containers virtualize the process

CONTAINER DEPLOYMENTS ARE INCREASING



KUBERNETES IS THE NEW WAY OF AUTOMATING APPLICATION RESILIENCY

- Auto scale
- Health checks
- Networking (CNI) & Routing
- Platform HA
- Application HA



KUBERNETES ENABLES AGILITY, SPEED TO MARKET



“Red Hat OpenShift allows us to go to market faster. We can move microservices and applications on OpenShift in a few seconds. That’s the impact this has on our business.” -- Luis Uguina, Chief Digital Officer, Macquarie Bank

- Digital-first bank, reshaping the Australian banking market
- Rethinking their mobile customer experience.
- Using RHEL, OpenShift and JBoss Fuse
- **More than 60 business critical applications on OpenShift**

This new model is helping us hire and retain top talent.

View the [Macquarie Bank keynote](#)

KUBERNETES IMPROVES UTILIZATION



Increased
application
development
throughput by a
factor of

10

from 20 to 200
changes a day

“Red Hat solutions have enabled us to deliver value to our customers much faster, with improved performance and stability.”

-- KERRY PEIRSE, GENERAL MANAGER OF I.T. INFRASTRUCTURE AND OPERATIONS, CATHAY PACIFIC AIRWAYS LIMITED

- A leading, international airline
- Goal: transform legacy infrastructure into a modern [hybrid cloud architecture](#).
- Forming the bridge to the public cloud is Red Hat OpenShift Container Platform, which supports more than 50 consumer-facing applications.
- Throughput for application deployment has been increased by a factor of 10.
- Reduced infrastructure footprint in terms of hardware, maintenance, and operations cost.
- Lowered the total cost of ownership (TCO) of production environments

[Cathay Pacific Takes Customer Experiences to New Heights with Red Hat’s Hybrid Cloud Technologies](#), May 2018.

AT ENTERPRISE SCALE, THE CLUSTER IS NOT THE SECURITY BOUNDARY

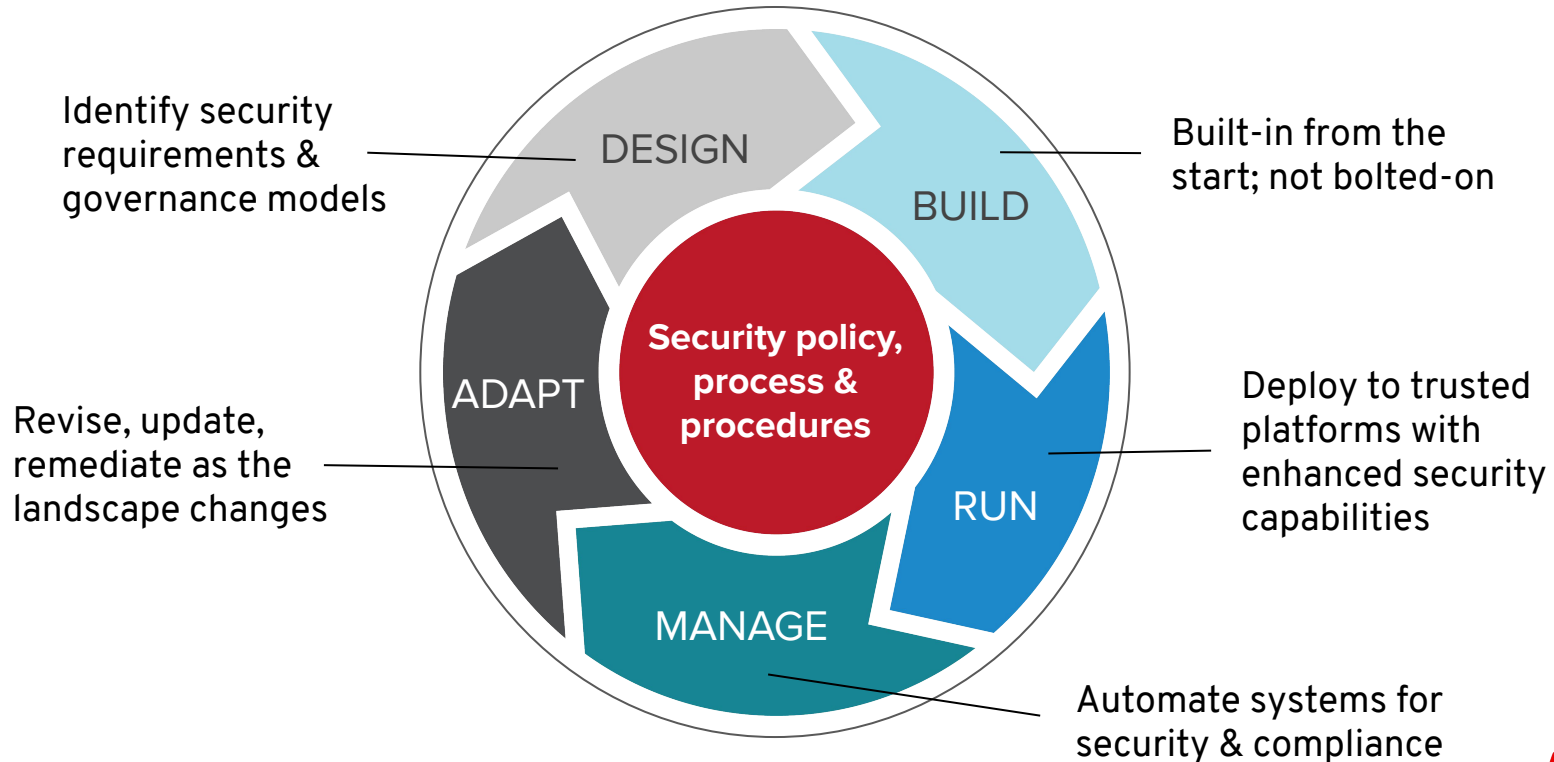
How do we ensure application security and process isolation on a Kubernetes cluster running complex, diverse workloads from multiple teams?

When should you consider single tenant clusters vs. multi-tenant?

How do you help your auditors understand this new world?

SECURING A MULTI-TENANT CLUSTER

Requires security throughout the stack and the IT lifecycle



ENTERPRISE KUBERNETES MULTI-TENANCY

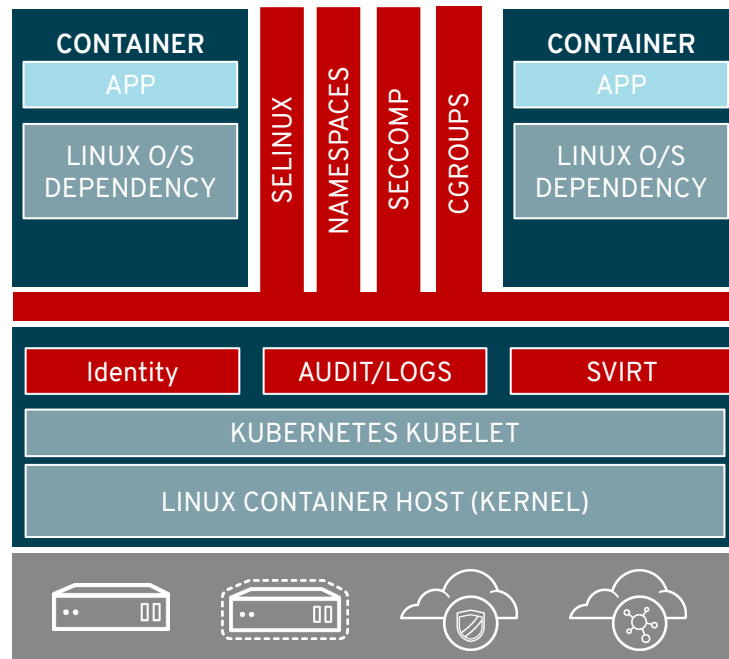
Layers and Lifecycle

1. Host OS
2. Container platform
3. Network
4. Containerized applications

1. HOST OS CONTAINER MULTI-TENANCY

Container Security starts with Linux Security

- Security in the host OS 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
- A container optimized OS provides a minimized attack surface
- Look for certifications such as Common Criteria cert with the container framework



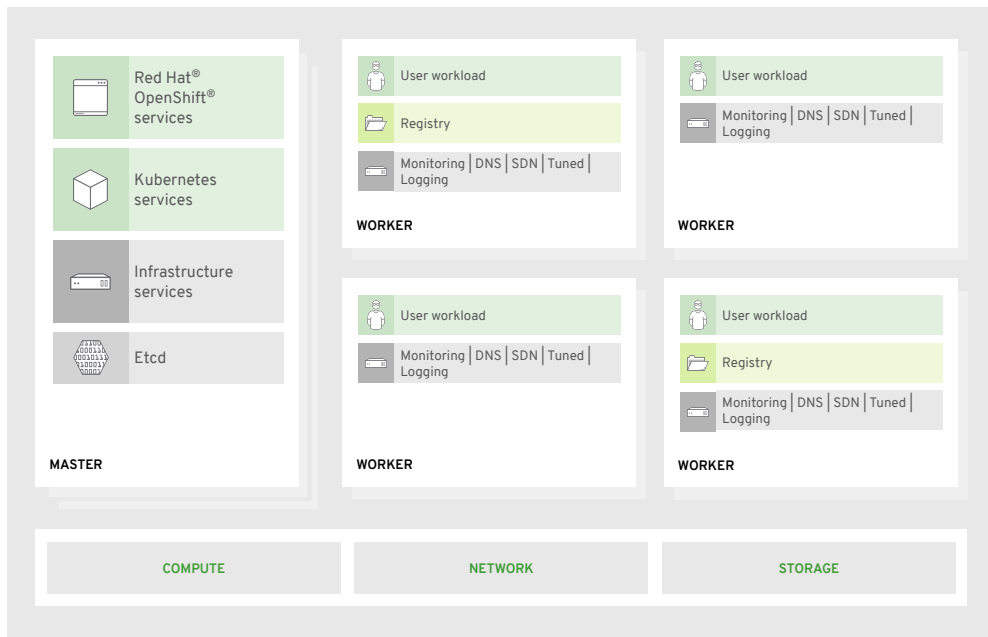
CONTAINER HOST VISION

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	Has a large ecosystem of supported solutions
Simple to configure	Installer generated configuration
Effortless to manage	Managed by Kubernetes Operators

2. THE CONTAINER PLATFORM

Necessary Multi-tenancy Features

- Host & Runtime security
- Identity and Access Management
- Project namespaces
- Integrated & extensible secrets management
- Log management & audit



USE HARDENING GUIDES & TOOLS

[CIS Kubernetes Benchmarks](#)

OpenShift Hardening Guide

Open source tools

- [docker-bench](#) supports versions 1.13 and 17.06
- [kube-bench](#)

RUNTIME SECURITY POLICIES

([Pod Security Policies](#) / [Security Context Constraints](#))

Allow administrators to control permissions for pods

Grant a restricted PSP / SCC to all users

By default, ensure no containers can run as root

Admin can grant access to privileged PSP / 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

Users and Service Accounts

Users are managed with outside identity providers such as

- OpenID Connect
- LDAP
- GitHub, GitHub Enterprise
- GitLab
- Google

API authentication strategies include

- X.509 client certificates
- Bearer tokens
- Authenticating proxy
- HTTP basic auth

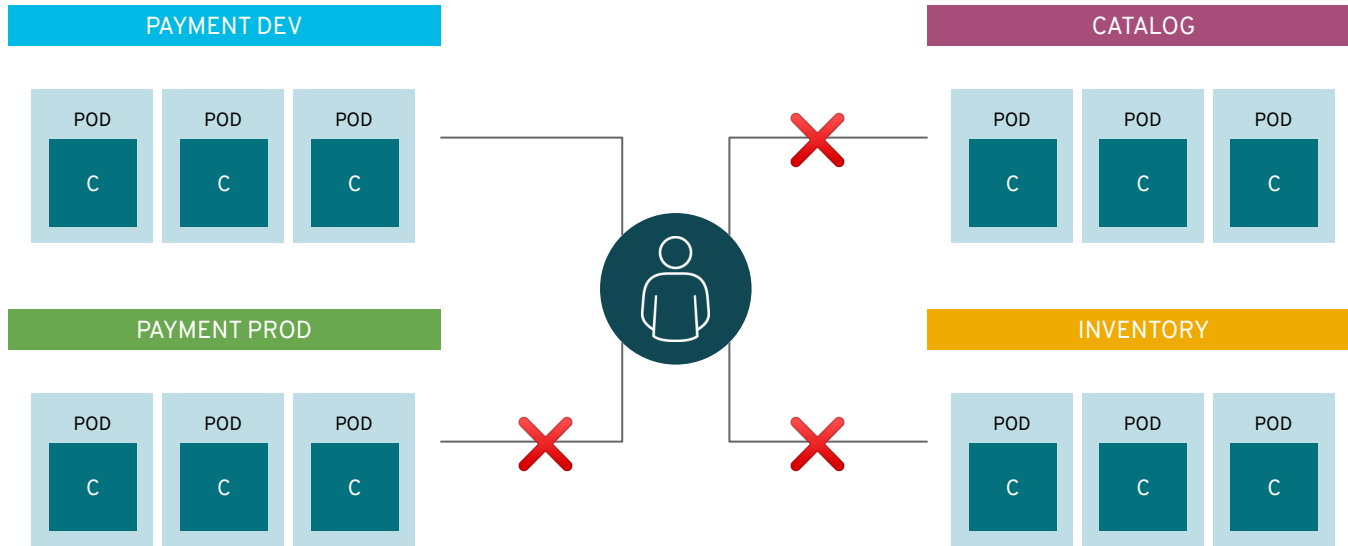
For example, OpenShift includes an OAuth server, which does three things:

- 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](#)

KUBERNETES NAMESPACES ISOLATE APPLICATIONS

across teams, groups and departments



RESTRICT ACCESS BY NEED TO KNOW

Role based authorization (RBAC) in OpenShift

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

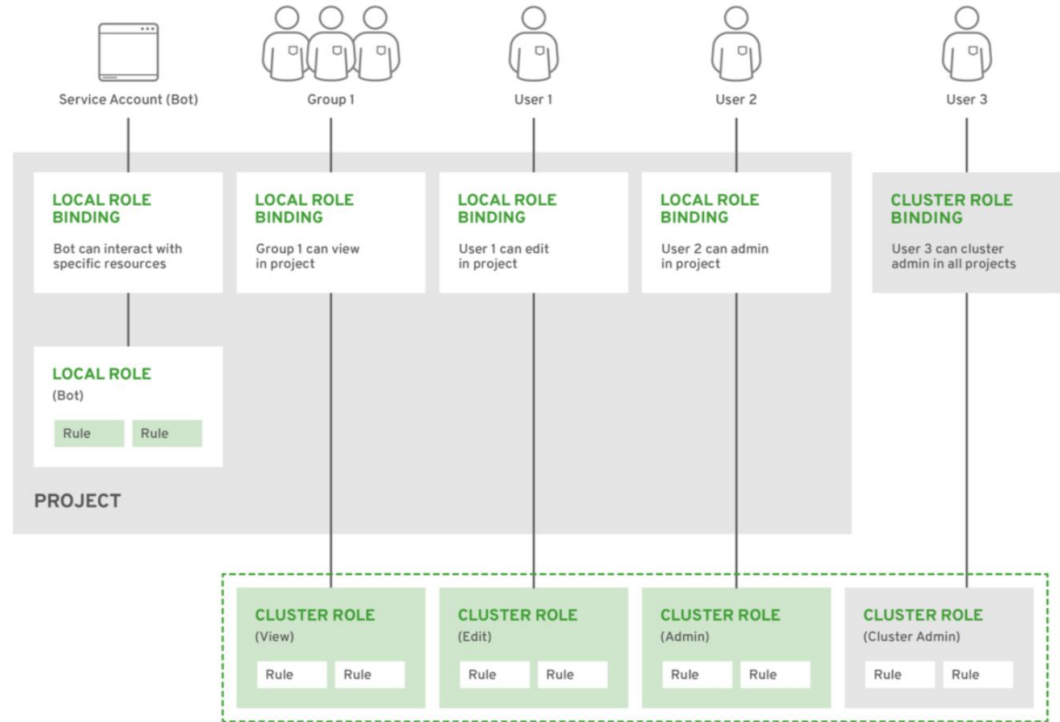
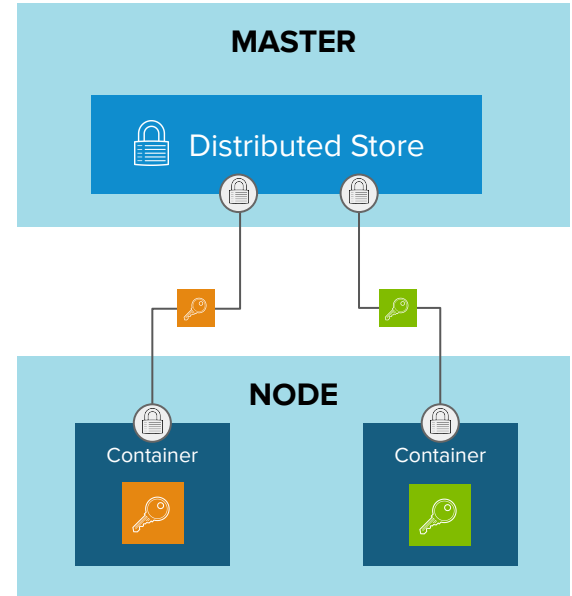


Figure 12 - Authorization Relationships

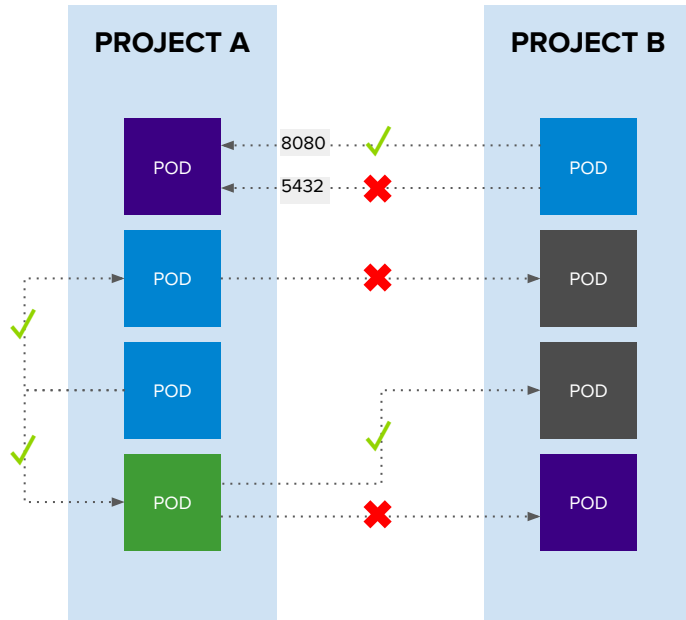
SECRETS MANAGEMENT

- Store platform secrets in etcd
 - Passwords and credentials
 - SSH Keys
 - Certificates
- Store application secrets in etcd or external vault
 - NOT in a pod definition or container image
- Make secrets available as
 - Environment variables
 - Volume mounts
 - Or through Interaction with external vaults
- Encrypt the etcd datastore



3. NETWORK MULTI-TENANCY

Fine Grained Control with Network Policy



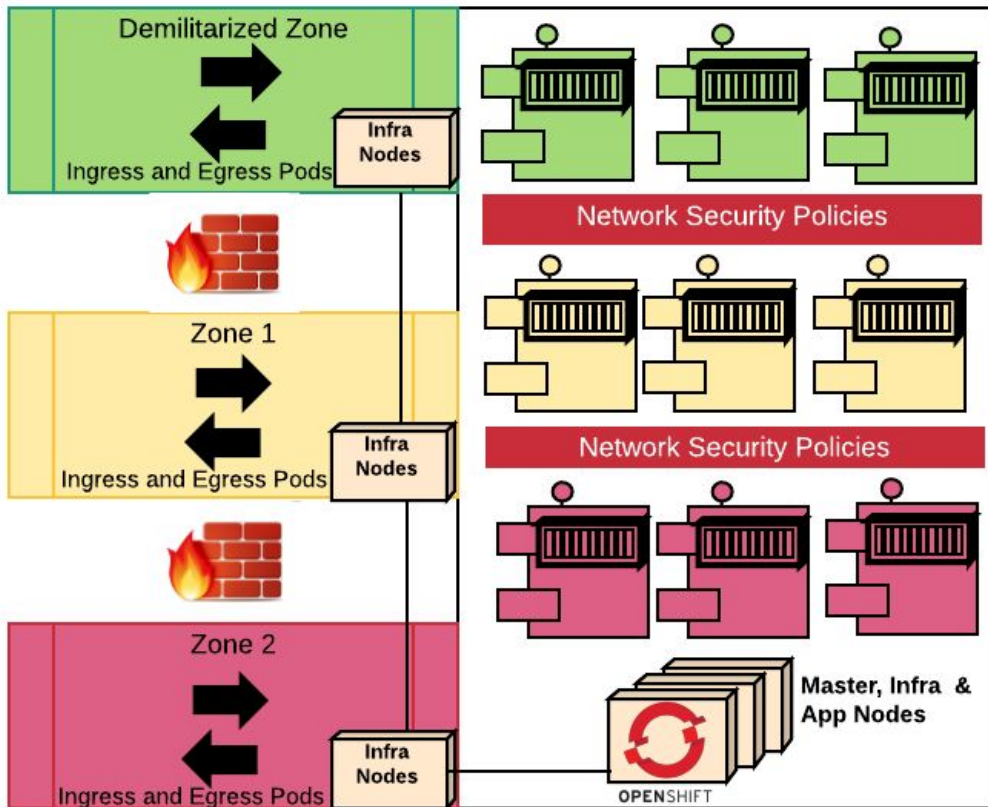
Example Policies

- 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
```

Enabled by default in OpenShift 4

MULTI-TENANT INGRESS & EGRESS CONTROL



Application pods run on a single cluster. Microsegmented with Network security policies.

Infra Nodes in each zone run Ingress and Egress pods for specific zones. Egress firewall used to limit external addresses accessed.

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

There may be cases where a single tenant cluster is preferred.

OPTIONALLY SEPARATE CONTROL PLANE AND DATA PLANE

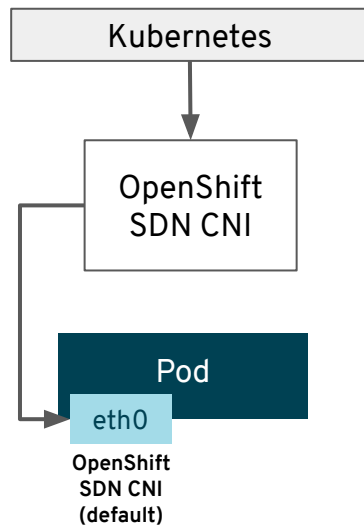
OpenShift Multus Enables Multiple Networks & New Functionality to Existing Networking

The Multus CNI “meta plugin” for Kubernetes enables one to create multiple network interfaces per pod, and assign a CNI plugin to each interface created.

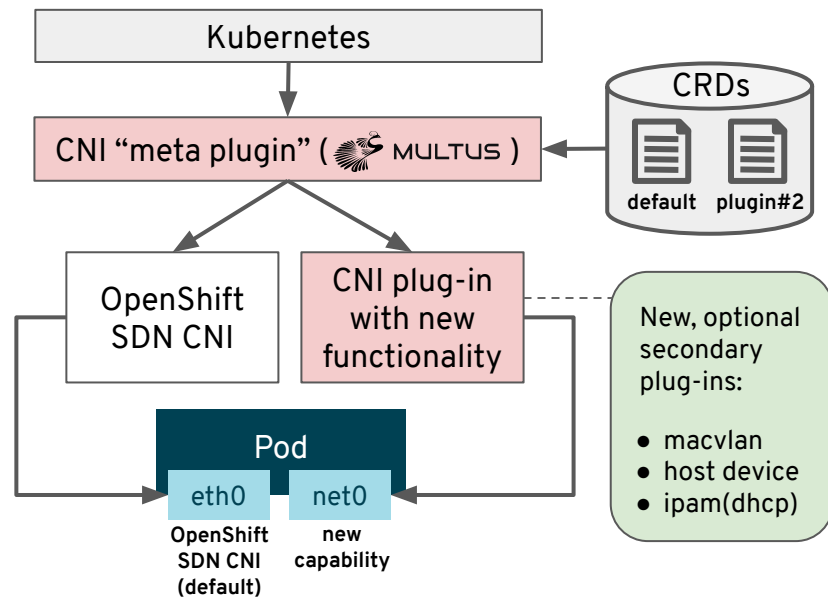
1. Create pod annotation(s) to call out a list of intended network attachments...
2. ...each pointing to CNI network configurations packed inside CRD objects

For more information, see [Managing Multiple Networks](#)

3.x Capability...

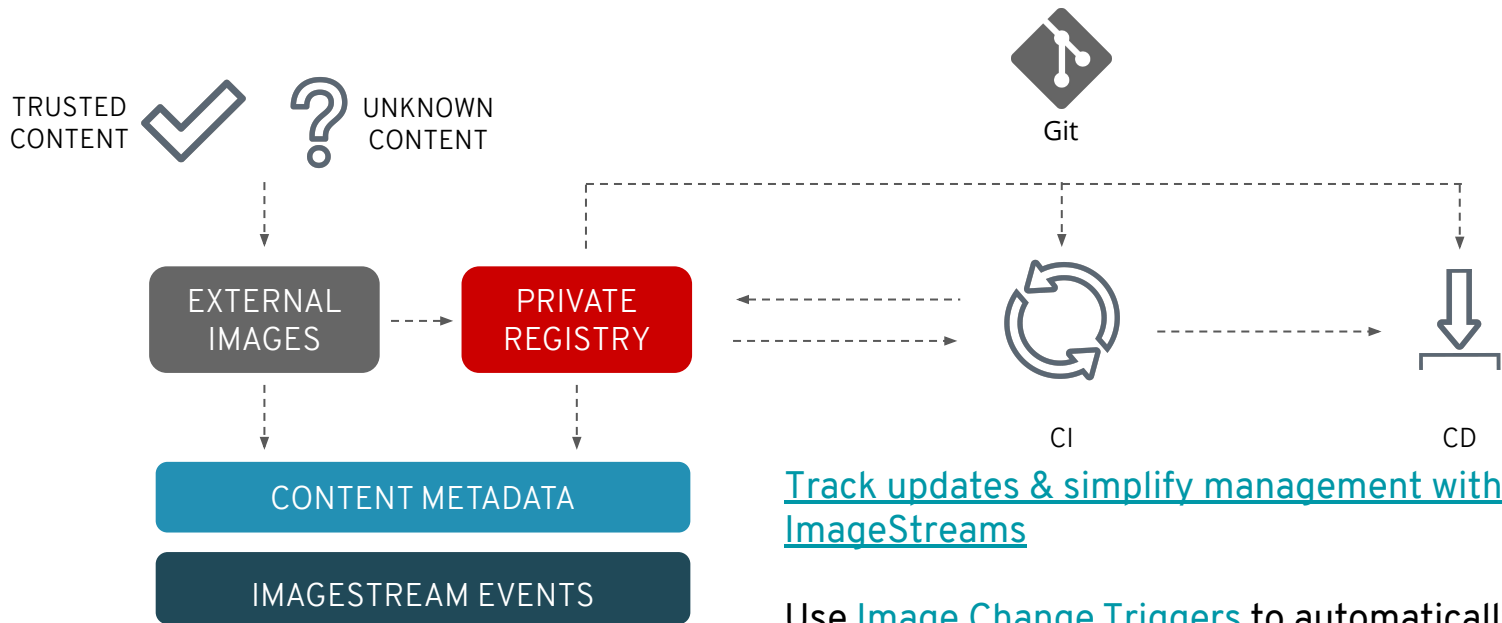


4.x Capability...



4. SECURING CONTAINERIZED APPLICATIONS

Secure and Automate the Content Lifecycle
Trust is temporal; rebuild and redeploy as needed

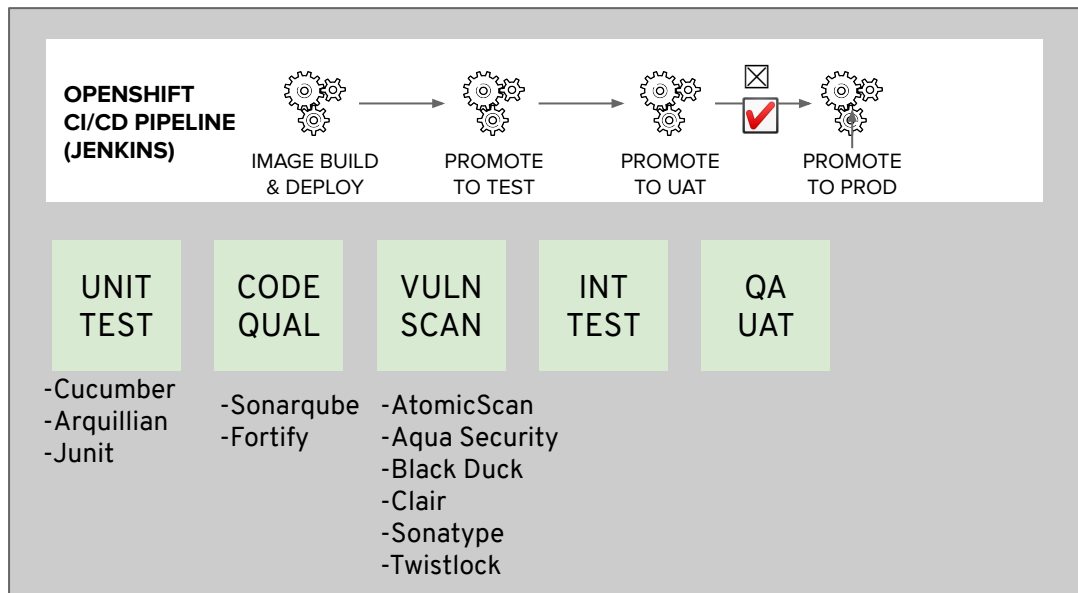


Track updates & simplify management with ImageStreams

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

CI/CD MUST INCLUDE SECURITY GATES

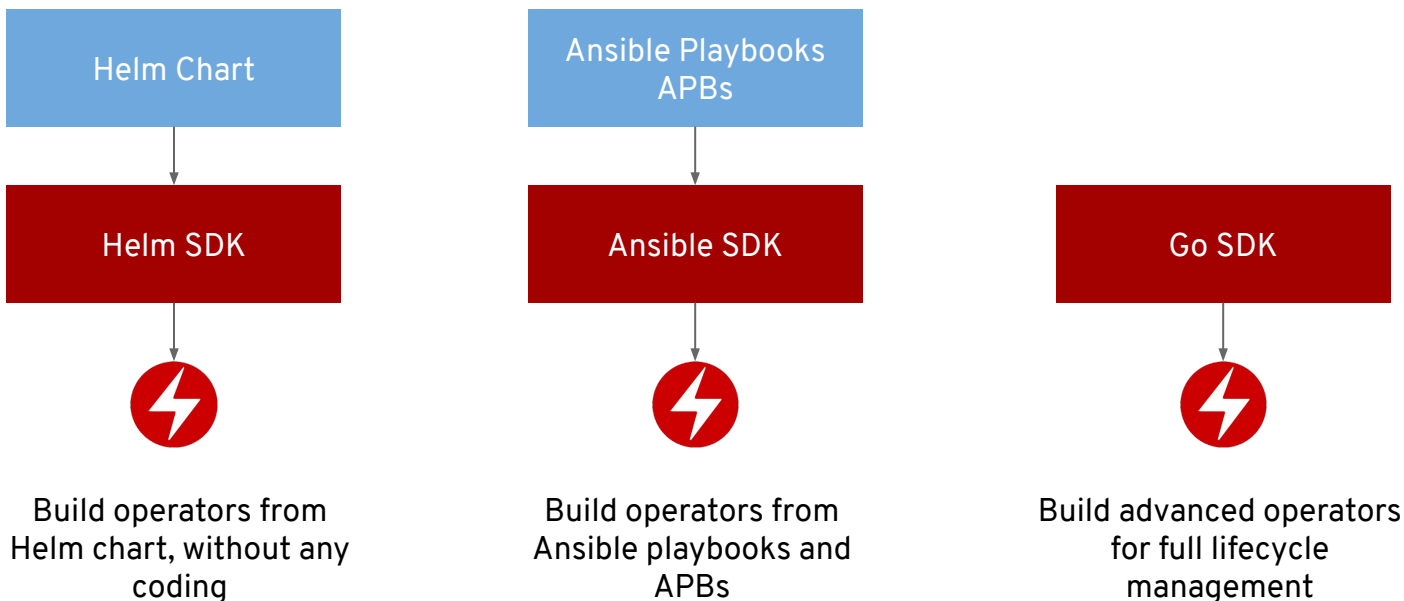
- Integrate security testing into your build / CI process
- Use an enterprise registry with integrated vulnerability scanning
- Use automated policies to flag builds with issues
- Sign your custom container images



BUILD OPERATORS FOR YOUR APPS



Use OLM to Manage your Application Lifecycle





docker



PostgreSQL

- Containerized



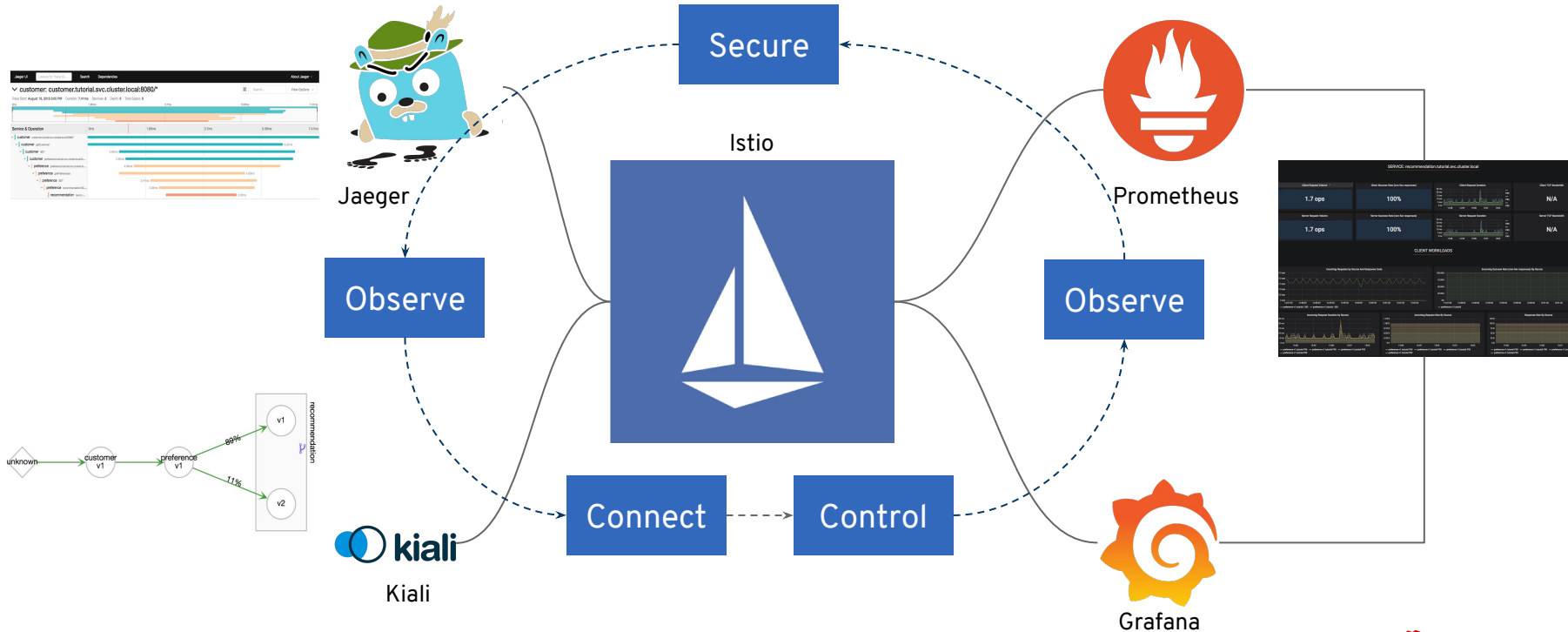
AWS RDS

- Containerized
- Cloud storage ready
- Replicated
- Backup
- Automated updates



- Containerized
- Container storage ready
- Replicated
- Backup
- Automated updates
- Enhanced observability
- Customization
- Local development
- Fully Open Source
- Any Kubernetes
- Certified on OpenShift

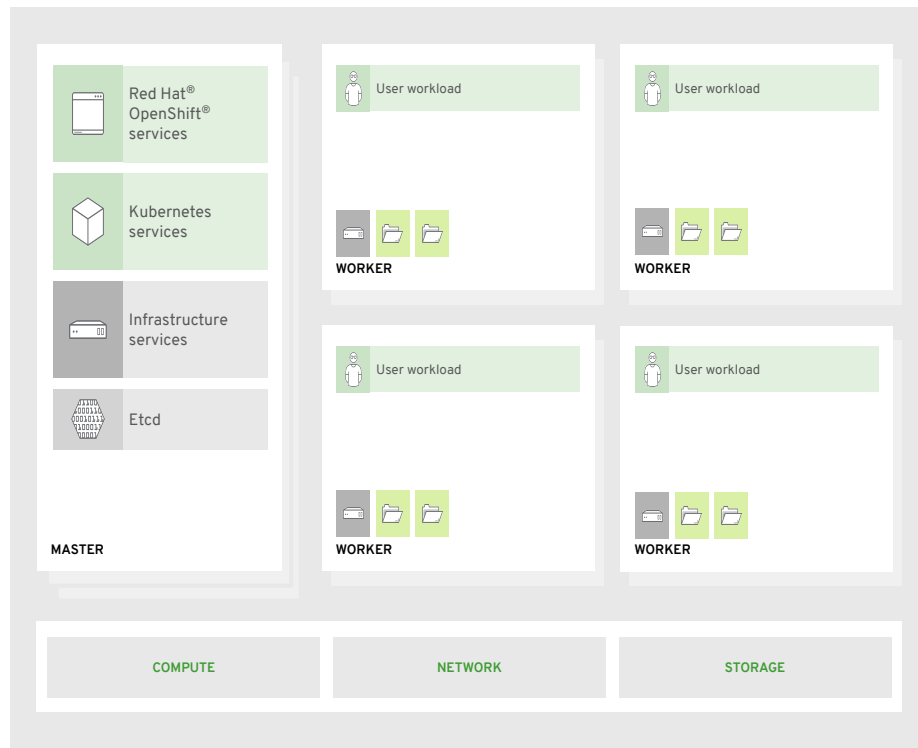
SECURE MICROSERVICES WITH SERVICE MESH



ATTACHED STORAGE

Secure storage by using

- SELinux access controls
- Secure mounts
- Supplemental group IDs for shared storage



LEVERAGE THE BROAD SECURITY ECOSYSTEM




Signal Sciences




THALES




Questions?

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

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

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

 twitter.com/RedHat



Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



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



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



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



twitter.com/RedHat