

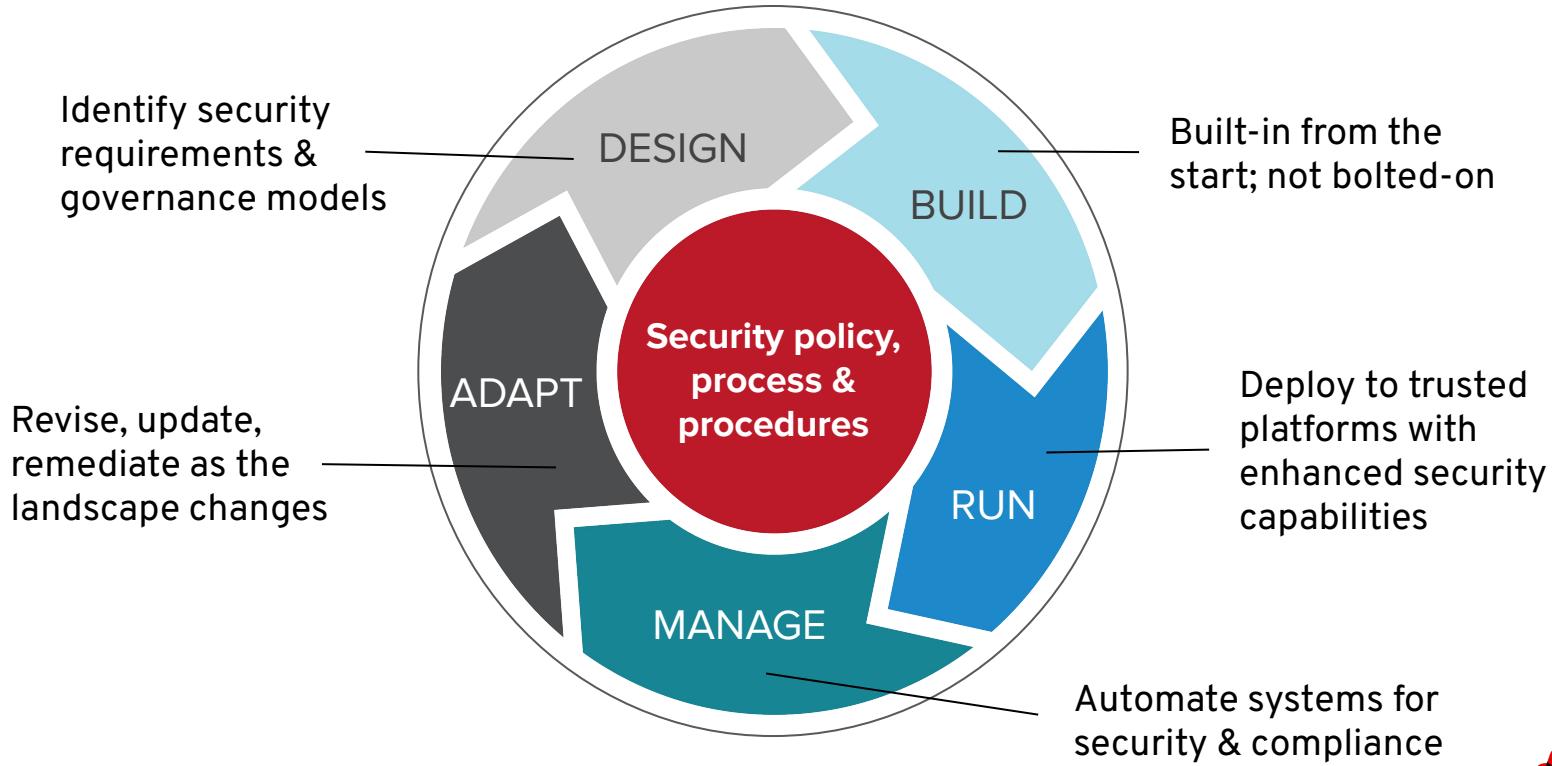
# SECURING CONTAINERS WITH OPENSHIFT

Alfred Bach  
Principal Solution Architect Cloud /Infra  
EMEA Partner Enablement



# Security must be continuous

## And integrated throughout the IT lifecycle



# 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

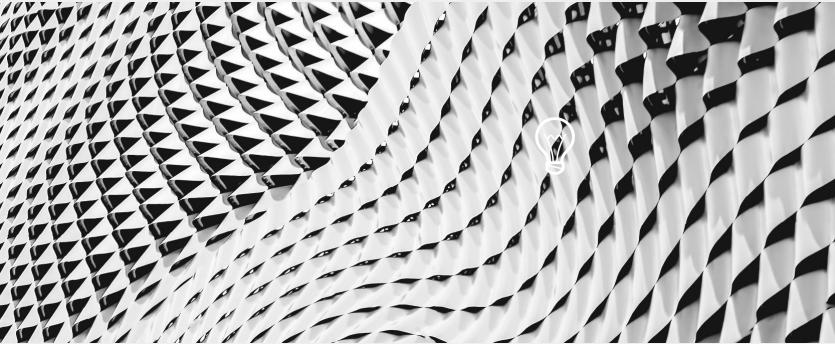
API Management



## EXTEND

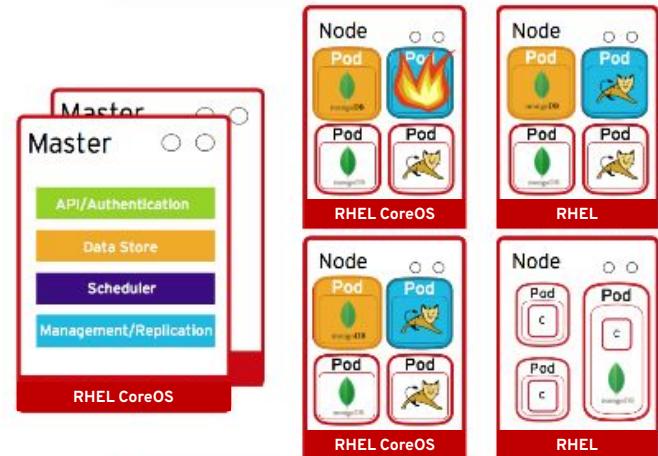
Security Ecosystem

# DEFEND INFRASTRUCTURE



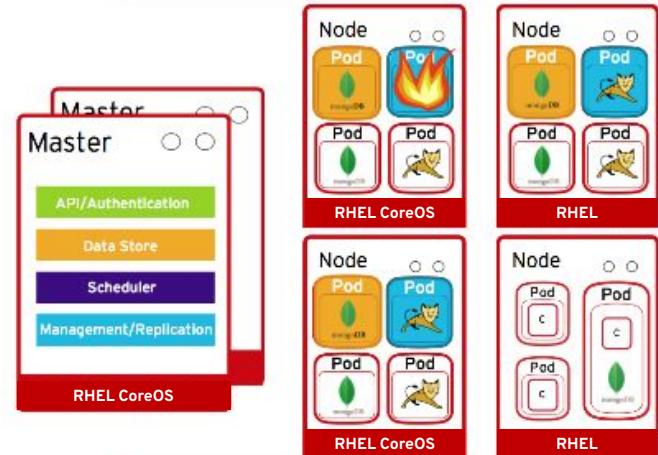
# Securing the container platform

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



# 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



# Automated Configuration and Lifecycle Management

## Dramatically simplified for the Hybrid Cloud



### Machines

Machines are complex for ops



Make machines easy  
(like containers)

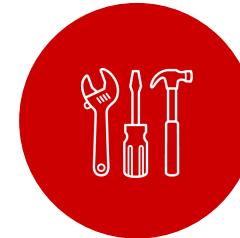


### Configuration

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

INSTALL

DEPLOY

HARDEN

OPERATE

## AUTOMATED OPERATIONS

Infra provisioning

Full-stack deployment

Secure defaults

Multicloud aware

Embedded OS

On-premises and cloud

Network isolation

Monitoring and alerts

Unified experience

Audit and logs

Full-stack patch & upgrade

Signing and policies

Zero-downtime upgrades

Vulnerability scanning

# The Value Of Kubernetes Operators

No need for operator

Requires custom Operator built with SDK



## 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.

# Kubernetes Machine API Operator

## Using Kubernetes To Provision And Scale Clusters

The screenshot shows the Red Hat OpenShift web console interface. The left sidebar is a navigation menu with the following items: Workloads, Networking, Storage, Builds, Monitoring, Administration (with sub-options: Cluster Settings, Namespaces, Nodes, Machine Deployments, Machine Sets, and Machines), Service Accounts, Roles, Role Bindings, Resource Quotas, Limit Ranges, and CRDs. The 'Machines' item under 'Administration' is currently selected. The main content area displays a table titled 'Machines' with columns: NAME, NAMESPACE, REGION, and AVAILABILITY ZONE. The table lists several machine objects, each with a three-dot menu icon. A search bar at the top right of the table says 'Filter Machines by name...'. The URL in the browser is 'console-openshift-console.apps.robszumski-0100.cloud.robszumski.com'.

The screenshot shows the Red Hat OpenShift web console interface. The left sidebar is a navigation menu with the following items: Workloads, Networking, Storage, Builds, Monitoring, Administration (with sub-options: Cluster Settings, Namespaces, Nodes, Machine Deployments, Machine Sets, and Machines), Service Accounts, Roles, Role Bindings, Resource Quotas, Limit Ranges, and CRDs. The 'Machine Sets' item under 'Administration' is currently selected. The main content area displays a 'Machine Set Details' page for 'robszumski-0100-worker-us-east-2a'. It includes tabs for Overview, YAML, and Machines. The Overview tab shows a detailed configuration for a machine set named 'robszumski-0100-worker-us-east-2a'. The configuration includes fields like 'spec': { 'metadata': { 'creationTimestamp': null, 'providerSpec': { 'value': 'useroluteSecret: robszumski-0100-worker-user-data placement: availabilityZone: us-east-2a region: us-east-2 keyName: null creationTimestamp: null instanceType: m4.large metadata: creationTimestamp: null publicIp: null securityGroups: - arn: null filters: - name: 'tag:Name' values: - robszumski-0100\_worker\_sg id: null kind: AKSMachineProviderConfig loadBalancers: null tags: - name: openshiftClusterID }, 'spec': { 'replicas': 2, 'selector': { 'matchLabels': { 'app': 'nginx', 'tier': 'ingress' } } } } }. The URL in the browser is 'console-openshift-console.apps.robszumski-0100.cloud.robszumski.com'.

# 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

The screenshot shows the Red Hat OpenShift Container Platform web interface. The top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and a user dropdown for 'kube:admin'. The left sidebar has a dark theme with white text. It lists various cluster management sections: Home, Catalog, Workloads, Networking, Storage, Builds, Monitoring, Compute (with sub-options for Nodes, Machines, Machine Sets, Machine Configs, Machine Config Pools), Administration (with sub-options for Cluster Status, Cluster Settings, Namespaces, Service Accounts, Roles, Role Bindings, Resource Quotas, Limit Ranges, Custom Resource Definitions), Infrastructure, Ingress, Network, OAuth, Project, and Scheduler. The 'Cluster Settings' section is currently selected. The main content area is titled 'Cluster Settings' and contains a table-like list of configuration resources. Each resource row has a name on the left and an 'Edit YAML' button on the right. The resources listed are: APIServer, Authentication, Build, ClusterVersion, Console, DNS, FeatureGate, Image, Infrastructure, Ingress, Network, OAuth, Project, and Scheduler. A note at the top of the main area says, 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.'

# 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

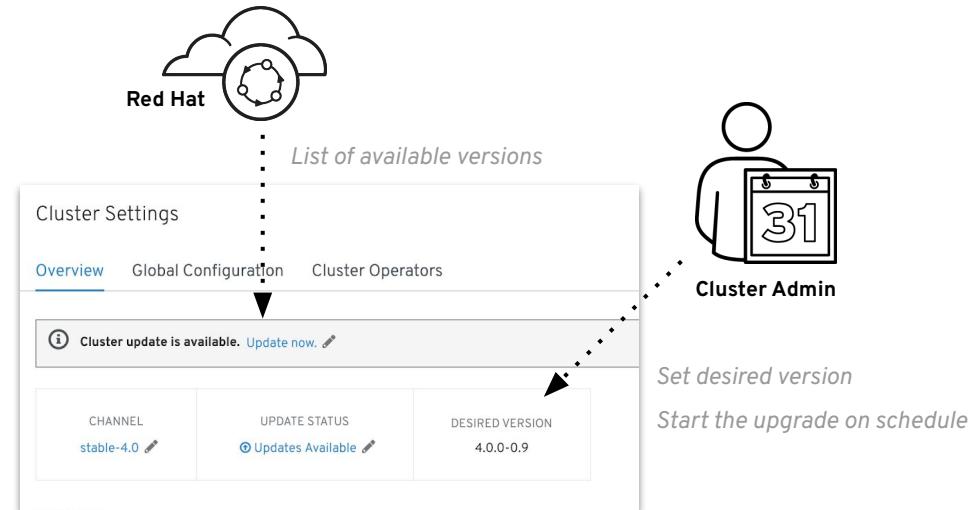
No need for separate windows for each component

## Upgrade runs completely on the cluster

No more long running processes on a workstation

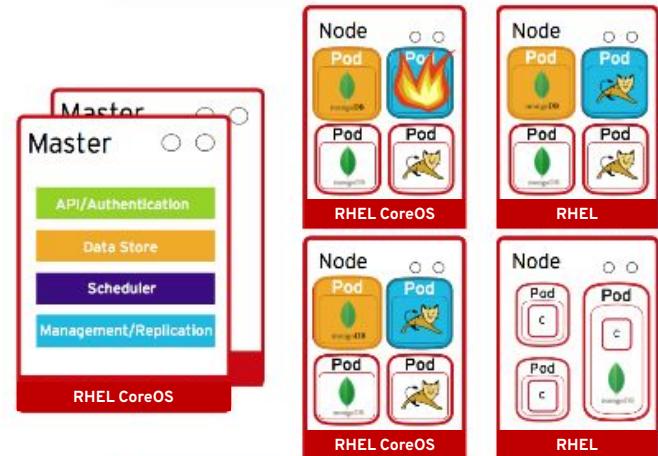
## Constant health checking from each Operator

Operators are constantly looking for incompatibilities and issues that might arise



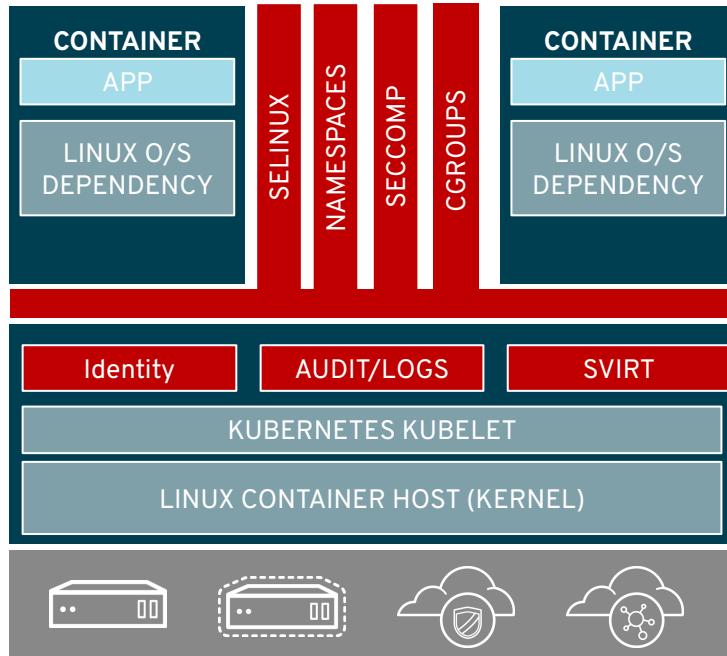
# Securing the container platform

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



# 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

OPENSHIFT 4

OPENSHIFT PLATFORM



OPERATING SYSTEM



RED HAT<sup>®</sup>  
ENTERPRISE  
LINUX CoreOS



## 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



# 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](#)



# Security Profile Operator

Helps admins use **SELinux** and **seccomp effectively**



**Easy seccomp and SELinux profile creation** by recording what your application needs and creates a profile from it



**Manages profiles across nodes and namespaces**

It also validates if node supports seccomp and doesn't synchronize it if

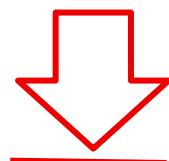
not



**Validate your profile**



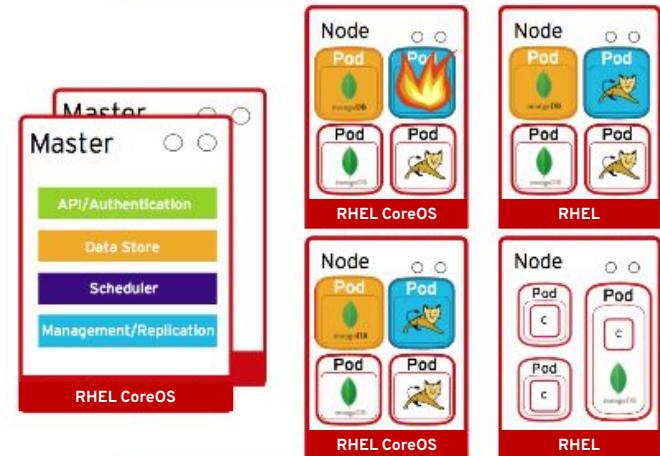
**Reuse profiles** across namespaces



**Available in OperatorHub**

# 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



# 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

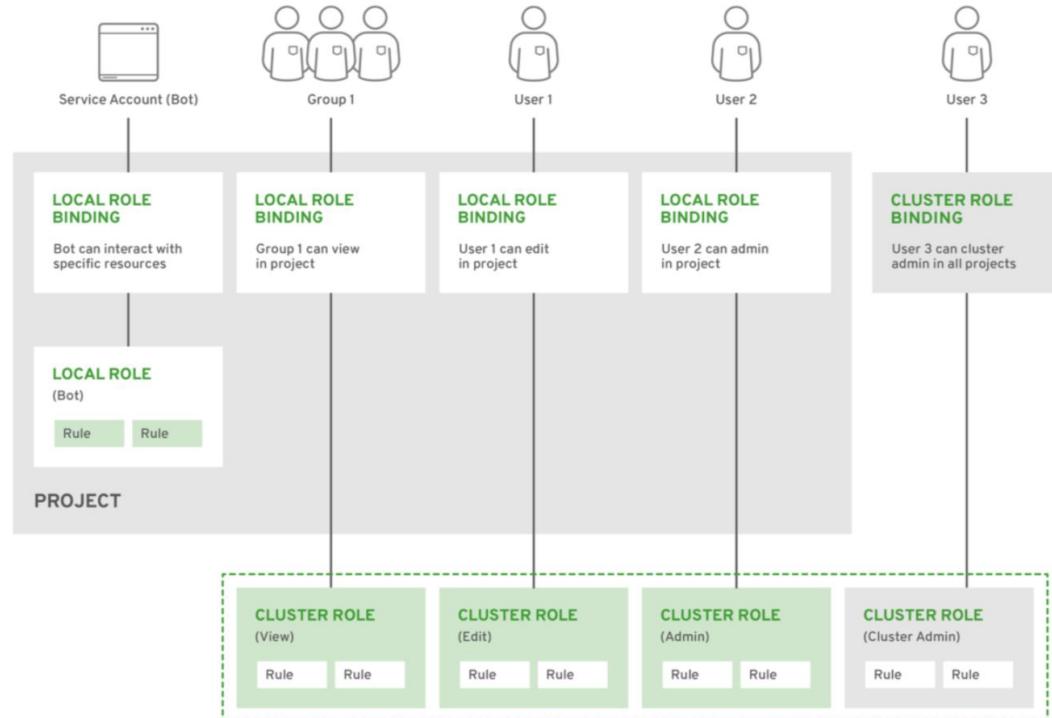
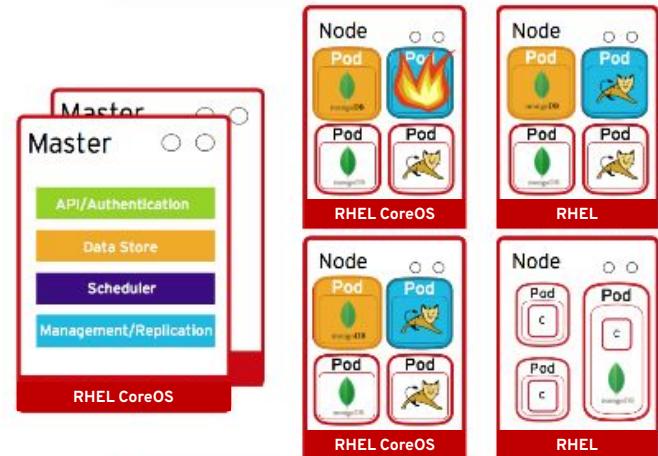


Figure 12 - Authorization Relationships

# 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



# 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](#)



MASTER



ETCD



NODES



INGRESS  
CONTROLLER



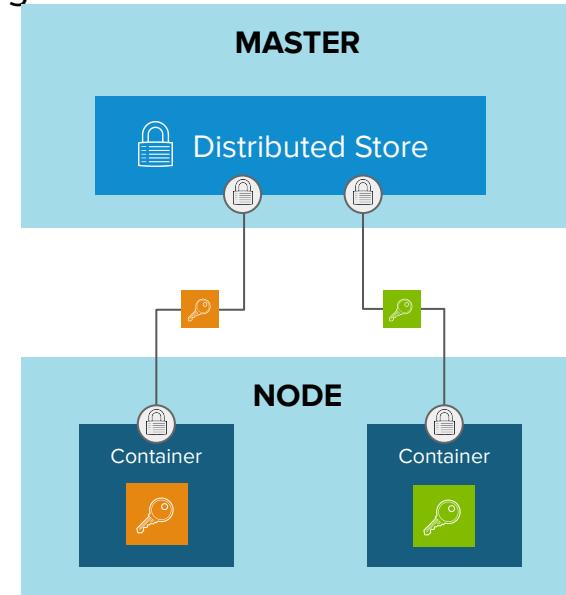
CONSOLE



REGISTRY

# 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

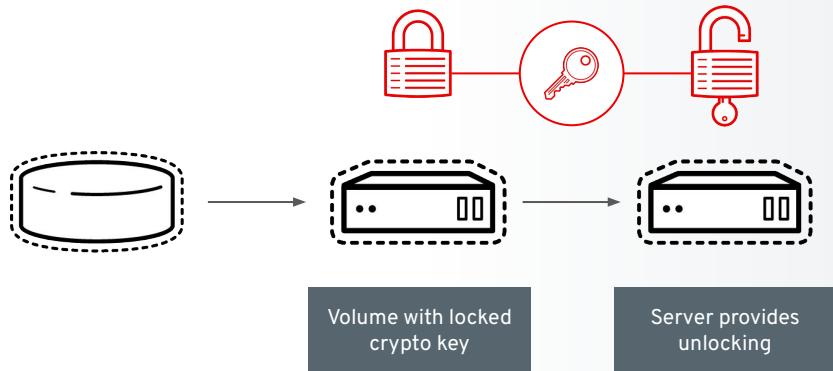
- Provides encryption for local storage
- Addresses disk/image theft
- Platform/cloud agnostic implementation
- TPM/vTPM (v2) and Tang endpoints for automatic decryption



# Attached storage

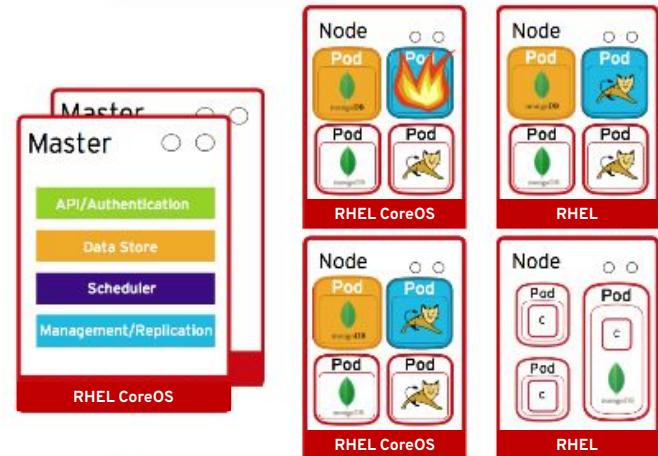
Secure storage by using

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



# 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



# 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

### Create Operator Subscription

Keep your service up to date by selecting a channel and approval strategy. The strategy determines either manual or automatic updates.

#### Installation Mode \*

All namespaces  
This mode  
Operator will be available in a single namespace only.

A specific namespace on the cluster  
Operator will be available in a single namespace only.

openshift-logging

#### Update Channel \*

preview

#### Approval Strategy \*

Automatic

Manual

**Subscribe** **Cancel**

```
# 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: "SingleRedundancy"
    visualization:
      type: "kibana"
      kibana:
        replicas: 1
    curation:
      type: "curator"
```

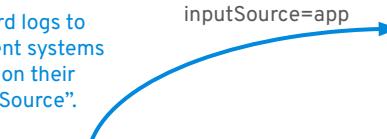
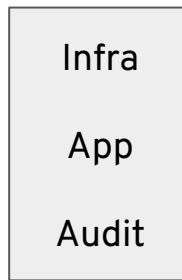
# Introduce new log forwarding API

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

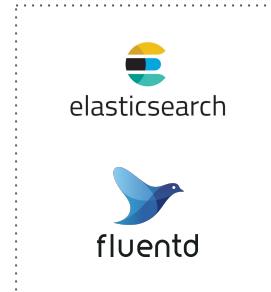
- Introduce a new, cluster-wide *ClusterLogForwarder* CRD (API) that replaces needs to configure log forwarding via Fluentd ConfigMap.
- The API helps to reduce probability to misconfigure Fluentd and helps bringing in more stability into the Logging stack.
- Route logs based on their source type (infra, 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.



Forward logs to different systems based on their "inputSource".



```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogForwarder"
spec:
  outputs:
    - name: MyLogs
      type: Syslog
      syslog:
        Facility: Local0
        url: localstore.example.com:9200
  pipelines:
    - inputs: [Infrastructure, Application, Audit]
      outputs: [MyLogs]
```



# Cluster monitoring

## Cluster monitoring is installed by default

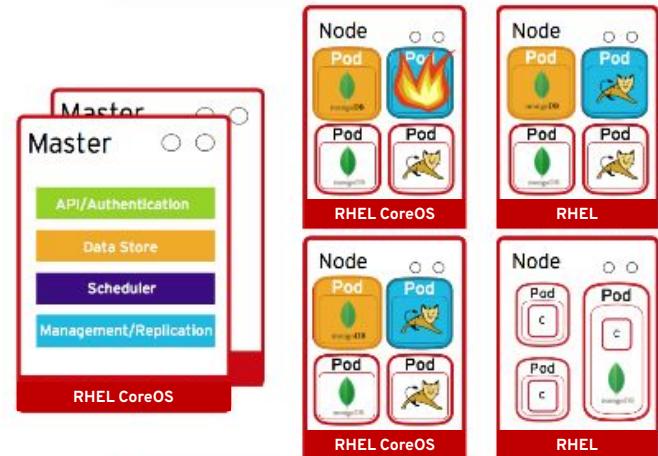
- Exposes resource metrics for Horizontal Pod Autoscaling (HPA) by default
  - HPA based on custom metric is tech preview
- No manual etcd monitoring configuration anymore
- New screens for managing Alerts & Silences
- More metrics available for troubleshooting purposes (e.g. HAProxy)
- Configuration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

The screenshot shows the Red Hat OpenShift Container Platform web interface. The left sidebar has a dark theme with the Red Hat logo at the top. It includes sections for OperatorHub, Operator Management, Workloads, Networking, Storage, Builds, Monitoring (which is currently selected), Alerts, Silences, Metrics, Dashboards, Compute, and Nodes. The Monitoring section has sub-options for Alerts, Silences, Metrics, and Dashboards. The main content area is titled "Alerts" and includes a sub-link to "Alertmanager UI". Below this, a message states: "Alerts help notify you when certain conditions in your environment are met. Learn more about how alerts are triggered." There are four buttons at the top of the alert list: "12 Firing", "0 Silenced", "0 Pending", and "77 Not Firing", followed by a "Select All Filters" button. The alert list itself has columns for NAME, STATE, and LAST FIRED. The first three alerts are of type "CPUThrottlingHigh" and the last one is "KubeDeploymentReplicasMismatch".

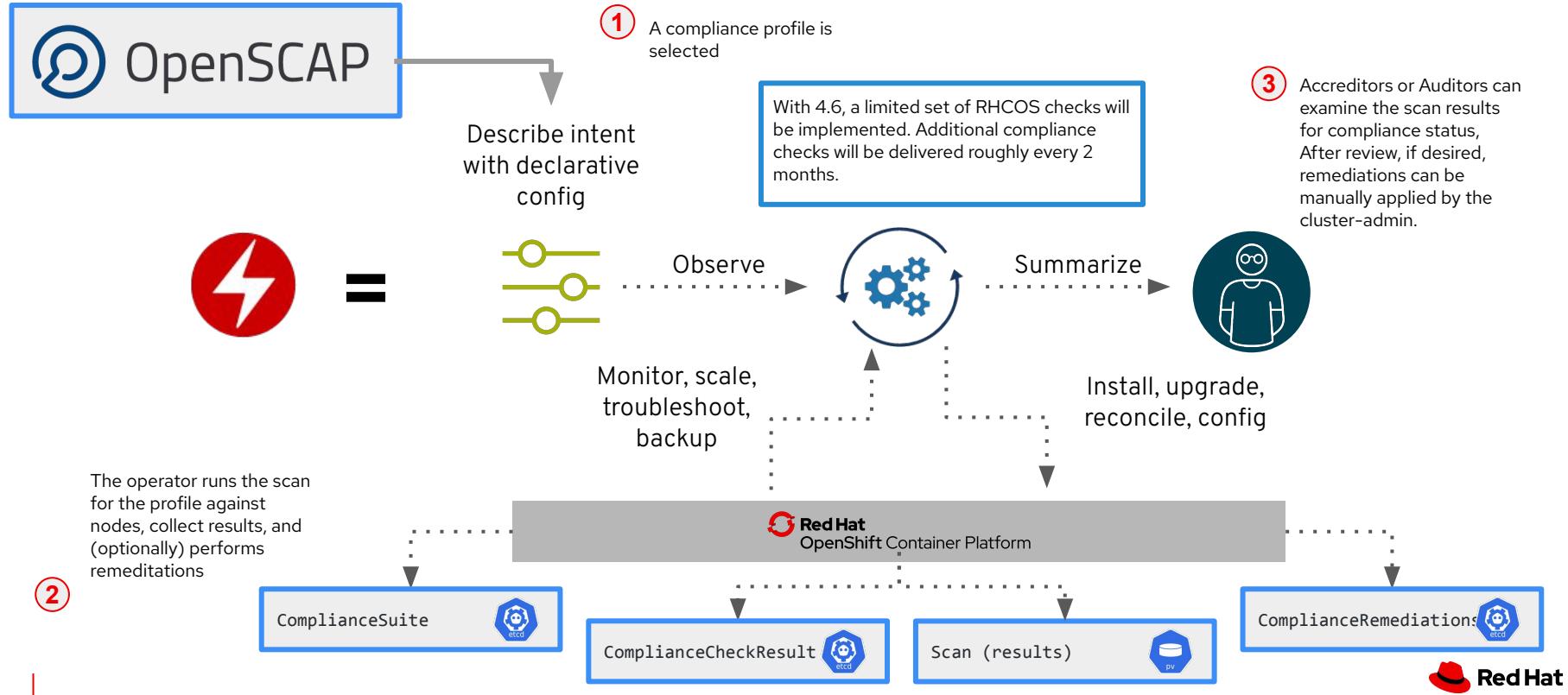
NAME	STATE	LAST FIRED
AL CPUThrottlingHigh	Firing	Since Apr 29, 11:52 am
AL CPUThrottlingHigh	Firing	Since May 2, 6:47 am
AL CPUThrottlingHigh	Firing	Since May 2, 6:47 am
AL KubeDeploymentReplicasMismatch	Firing	Since May 2, 1:34 pm
AL KubePodCrashLooping	Firing	Since Apr 29, 2:52 pm

# Securing the container platform

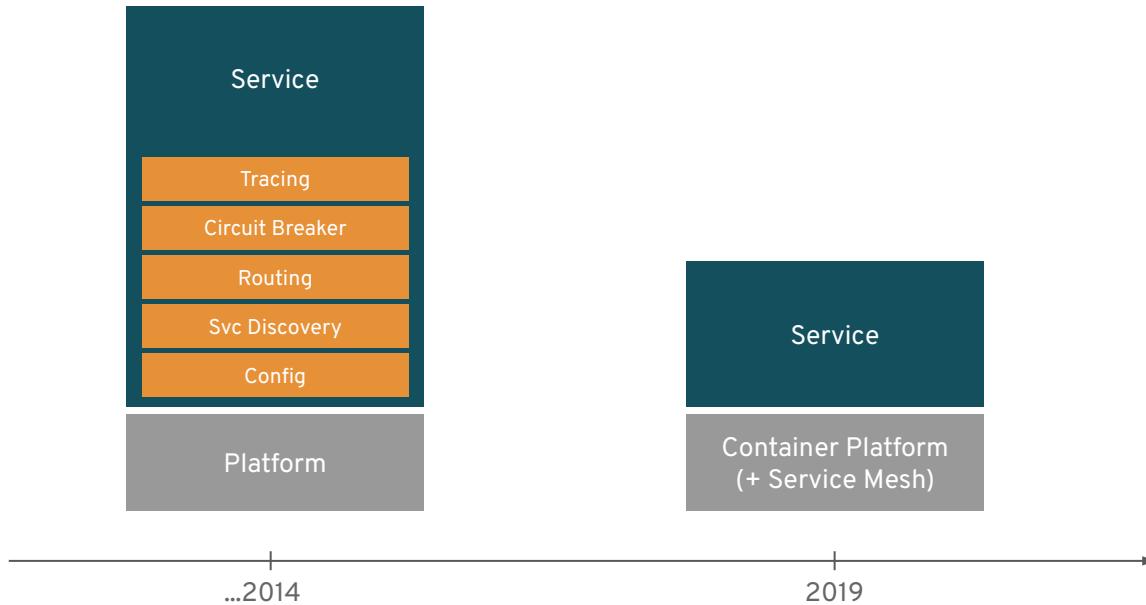
- Configuration and lifecycle management
- Host & runtime security
- Identity and Access Management
- Data at rest, data in transit
- Logging, Monitoring, Metrics
- **Audit and Compliance**



# Openshift Compliance Operator: Declarative Security Compliance



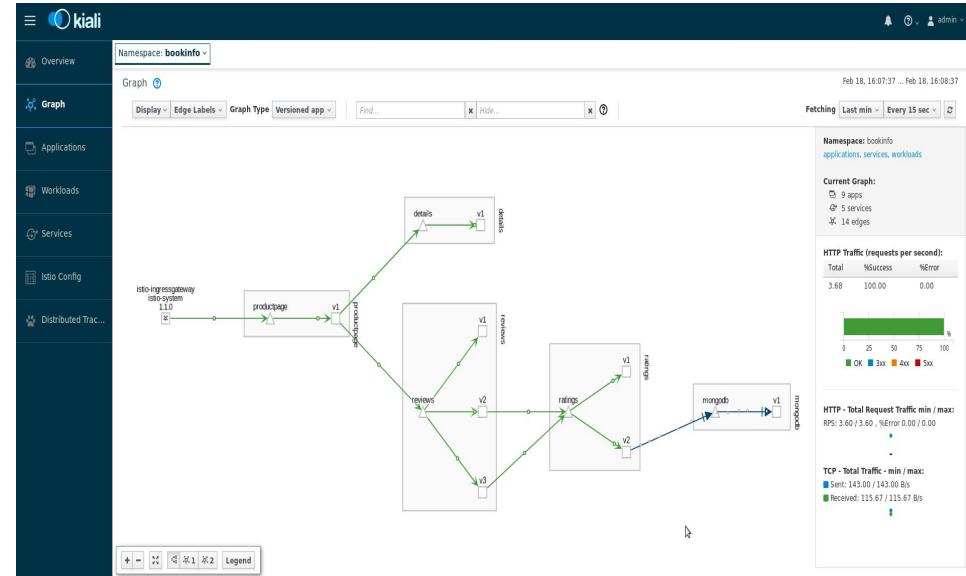
# Microservices evolution



# 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



# Observability with Kiali

Kiali

Namespace: bookinfo

Graph

Display Edge Labels Graph Type Versioned app Find... Hide... ?

Fetching Last min Every 15 sec

Namespace: bookinfo applications, services, workloads

Current Graph:

- 9 apps
- 5 services
- 14 edges

HTTP Traffic (requests per second):

Total	%Success	%Error
3.68	100.00	0.00

OK 3xx 4xx 5xx

HTTP - Total Request Traffic min / max:  
RPS: 3.60 / 3.60, %Error 0.00 / 0.00

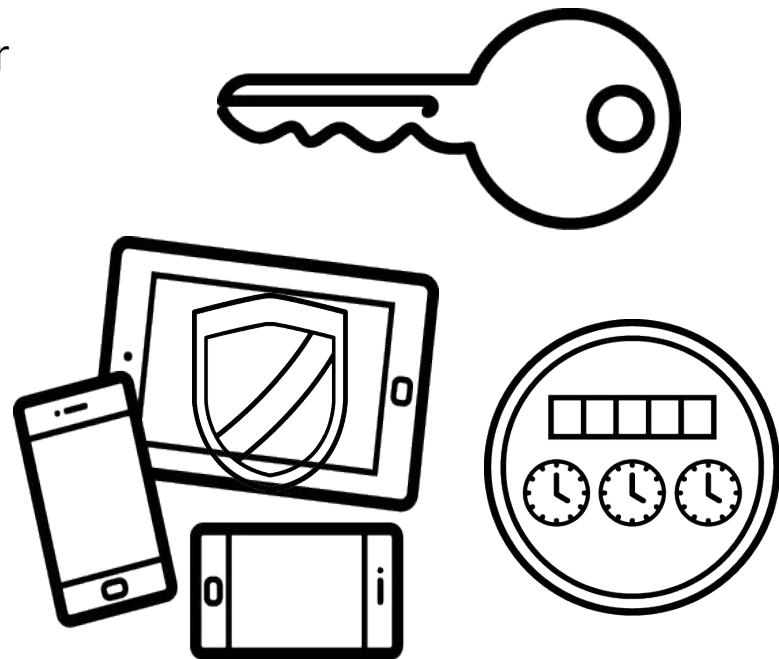
TCP - Total Traffic - min / max:  
Sent: 143.00 / 143.00 B/s  
Received: 115.67 / 115.67 B/s

The Kiali interface shows a service mesh graph for the 'bookinfo' namespace. The graph consists of several boxes representing services: 'istio-ingressgateway' (version 1.1.0), 'productpage', 'reviews', 'ratings', and 'mongodb'. Edges represent traffic between these services, with labels indicating the direction and type of traffic. For example, there are edges from 'istio-ingressgateway' to 'productpage' and from 'productpage' to 'details'. The 'reviews' service has an edge to 'v2' and another to 'v3'. The 'ratings' service has edges to 'v1' and 'v2'. The 'mongodb' service has an edge to 'v1'. On the right side of the interface, there are three cards providing real-time observability data: 'HTTP Traffic (requests per second)', 'HTTP - Total Request Traffic min / max', and 'TCP - Total Traffic - min / max'. The 'HTTP Traffic' card shows a single bar at 100.00% success rate. The 'HTTP - Total Request Traffic' card shows RPS values of 3.60. The 'TCP - Total Traffic' card shows B/s values of 143.00 for both sent and received traffic.

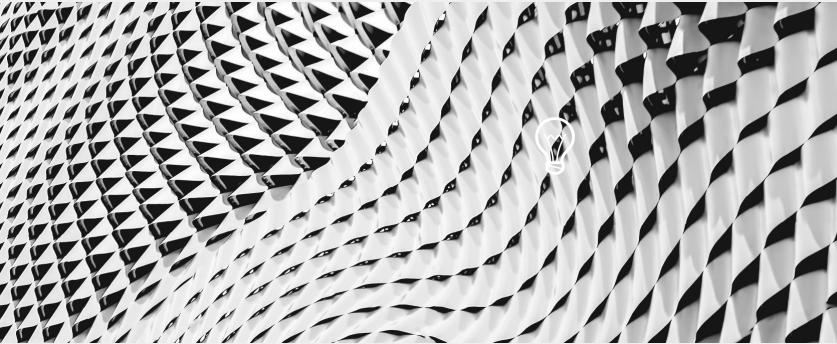
# Application API management

Consider configuring an API gateway for container platform & application APIs

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



# EXTEND SECURITY

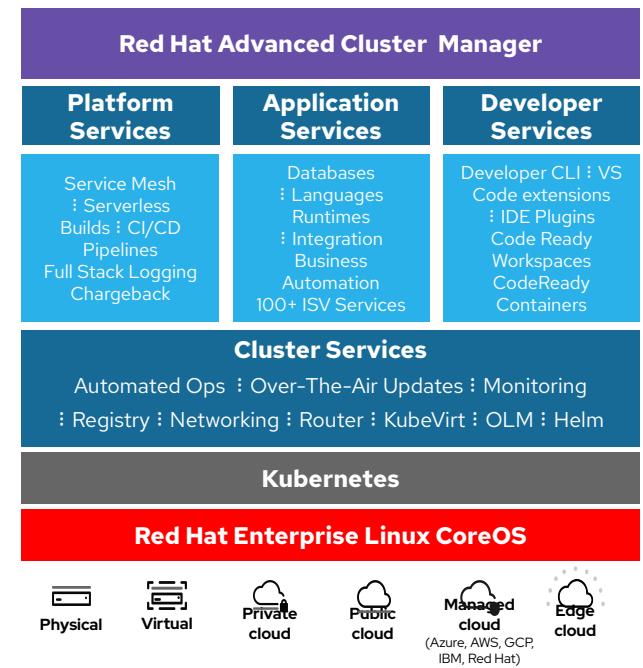


# The Security Ecosystem

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



# 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](https://twitter.com/RedHat)