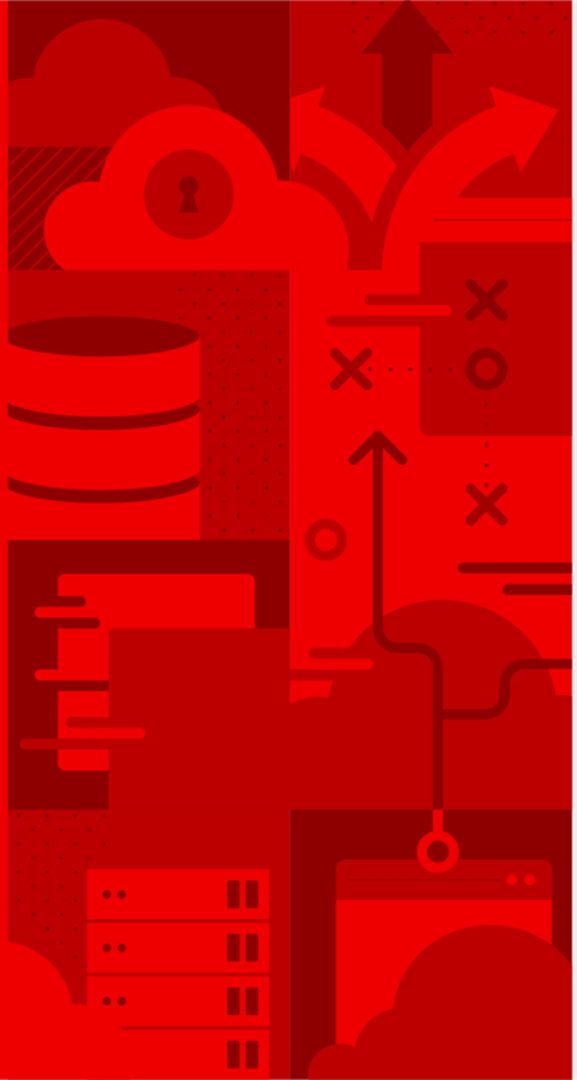


OPENSHIFT 4 CONTAINER PLATFORM

TECHNICAL OVERVIEW

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat

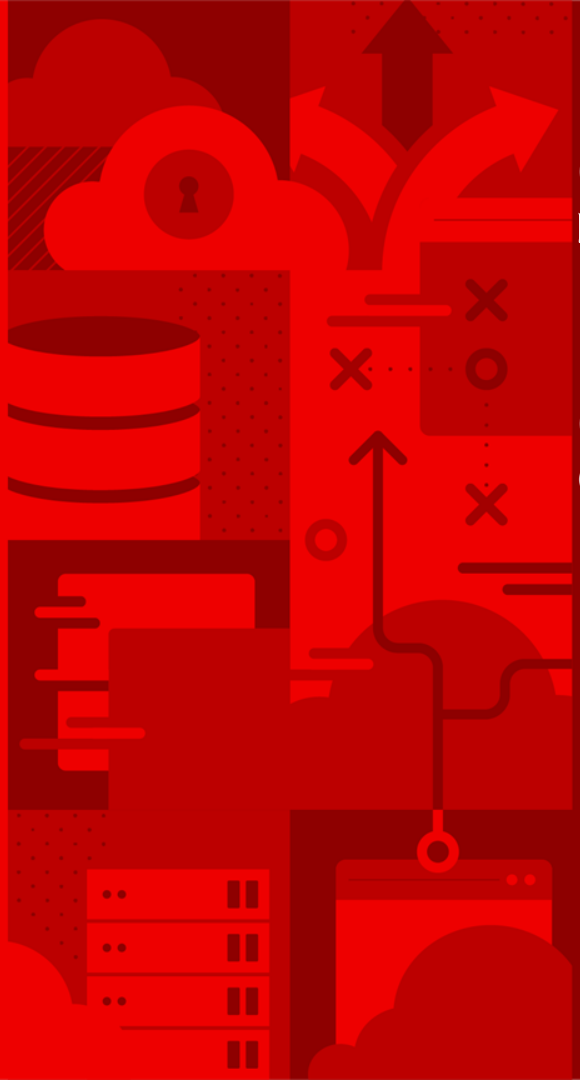
Red Hat
OpenShift
Container Platform

ETMall OpenShift – Design Workshop

06 Sep 2023

Wu Yi Chung
Consultant

Red Hat



Red Hat Scope and Expectation.

Objective:

瞭解客戶實際環境及應用部署需求，討論並設計平台部署計畫.

Outcome:

OCP 平台部署文件

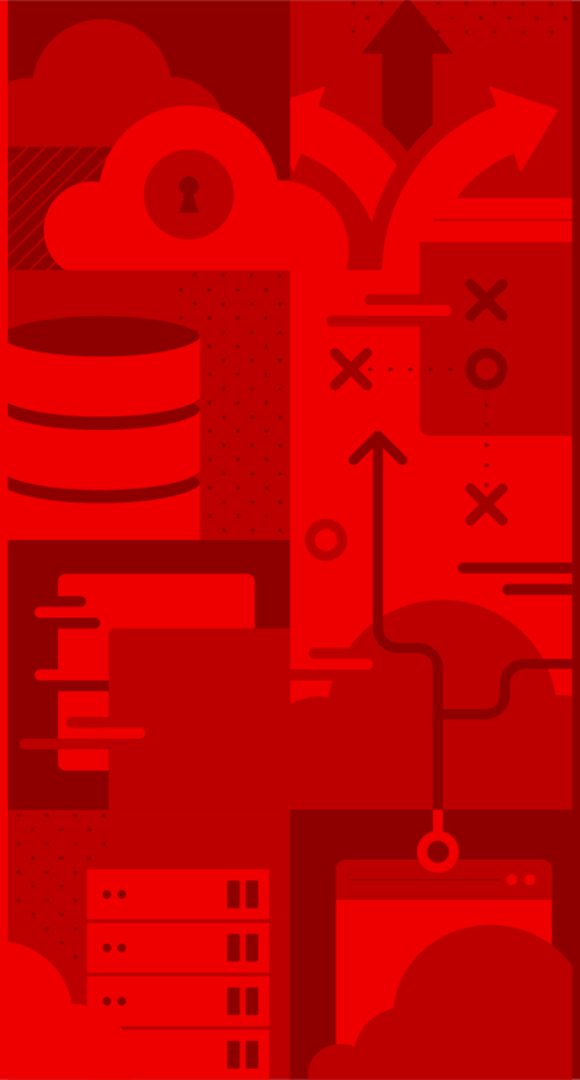
Red Hat

Agenda

Day 1 (09/06)	10:00 – 11:00	Introduction to OpenShift
	11:15 – 12:30	Discussion: OpenShift Installation
		Launch Break
	14:00 – 15:00	Discussion: Networking
	15:15 – 16:00	Discussion: Node Sizing (Computing Resource)
	16:15 – 16:45	Discussion: SSL Certificate & Security
	16:45 – 17:00	Q&A

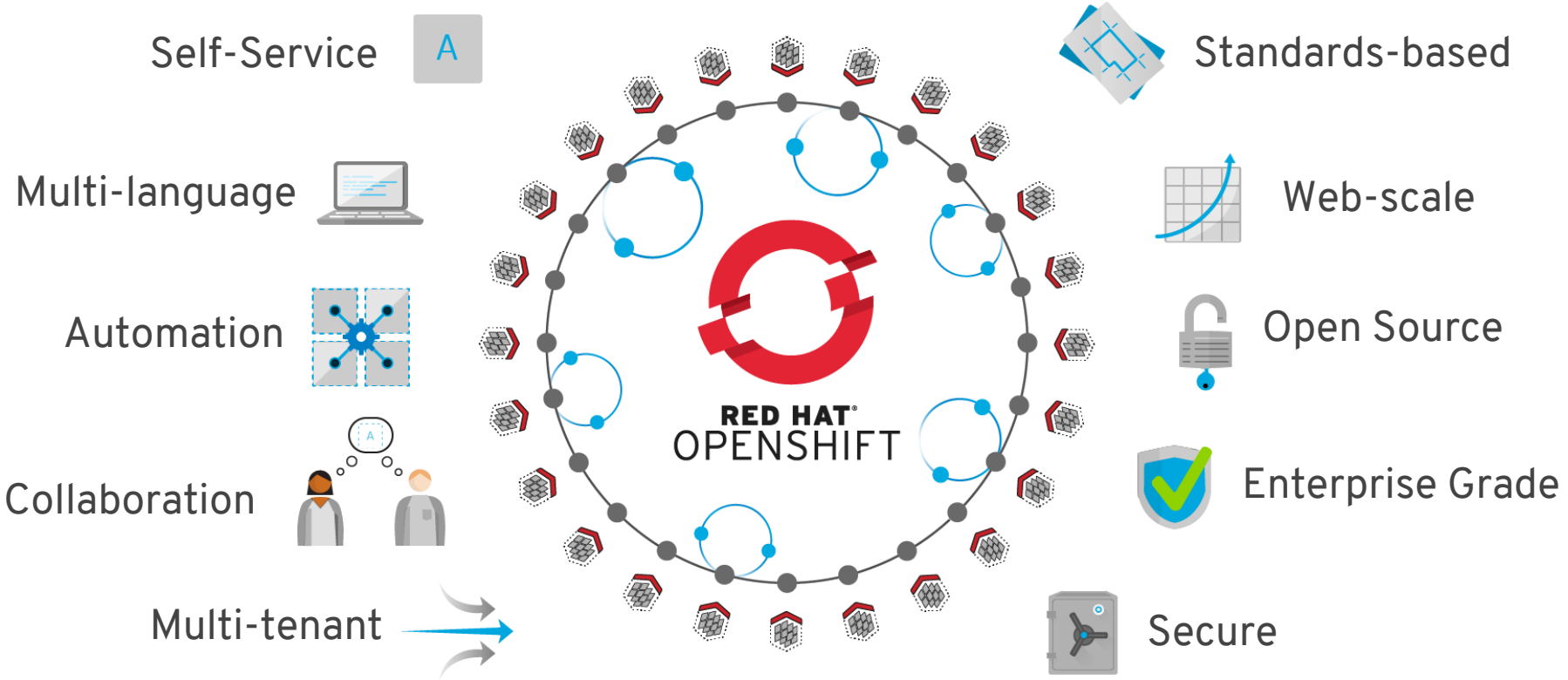
Agenda

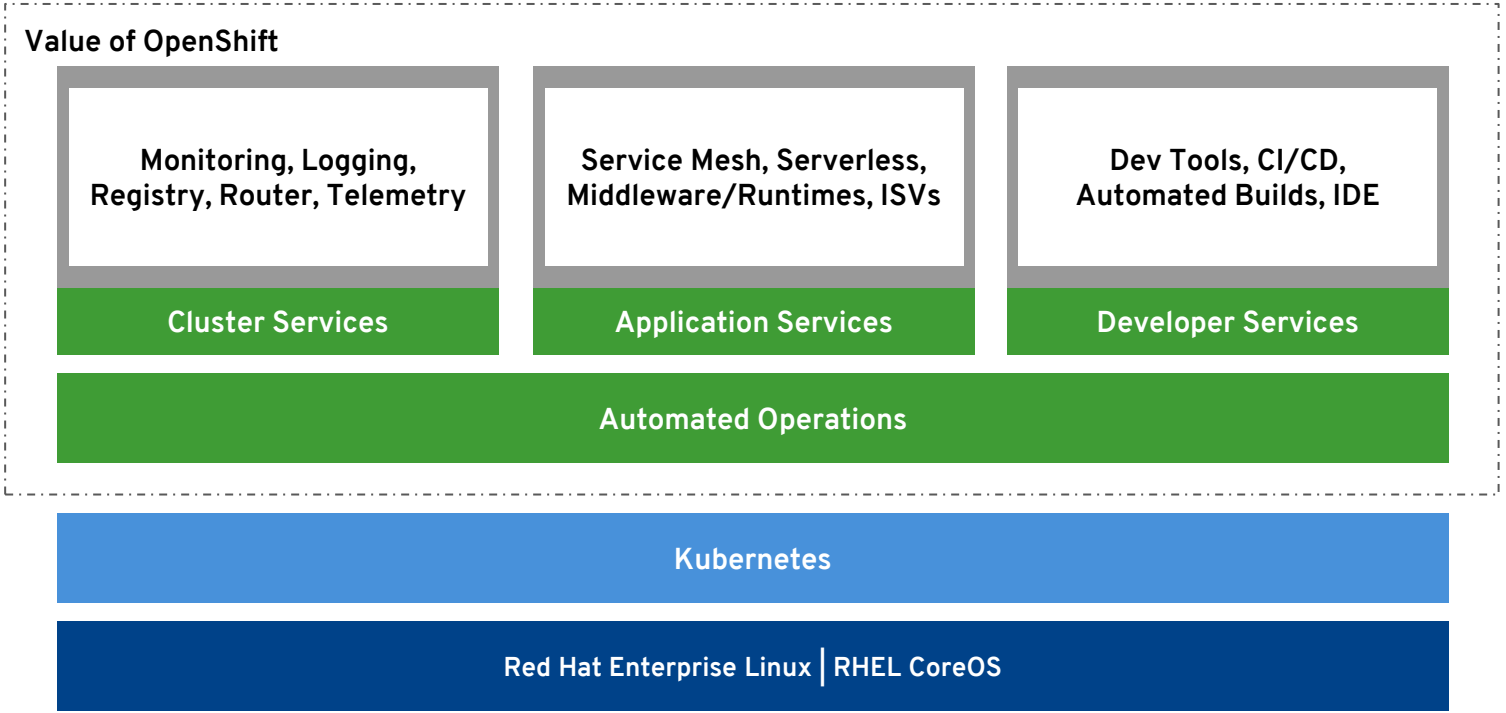
Day 2 (09/07)	10:00 – 11:00	Discussion: Storage (with NetApp)
	11:15 – 12:30	Discussion: Node Sizing (Storage)
		Launch Break
	14:00 – 15:00	Discussion: Quay & Image Registry
	15:15 – 16:00	Discussion: User Role & Permission
	16:15 – 16:45	Discussion: Others Need Attentions
	16:45 – 17:00	Q&A



Functional overview

Red Hat


Self-Service
Multi-language
Automation
Collaboration
Multi-tenant
RED HAT®
OPENSHIFT
Standards-based
Web-scale
Open Source
Enterprise Grade
Secure

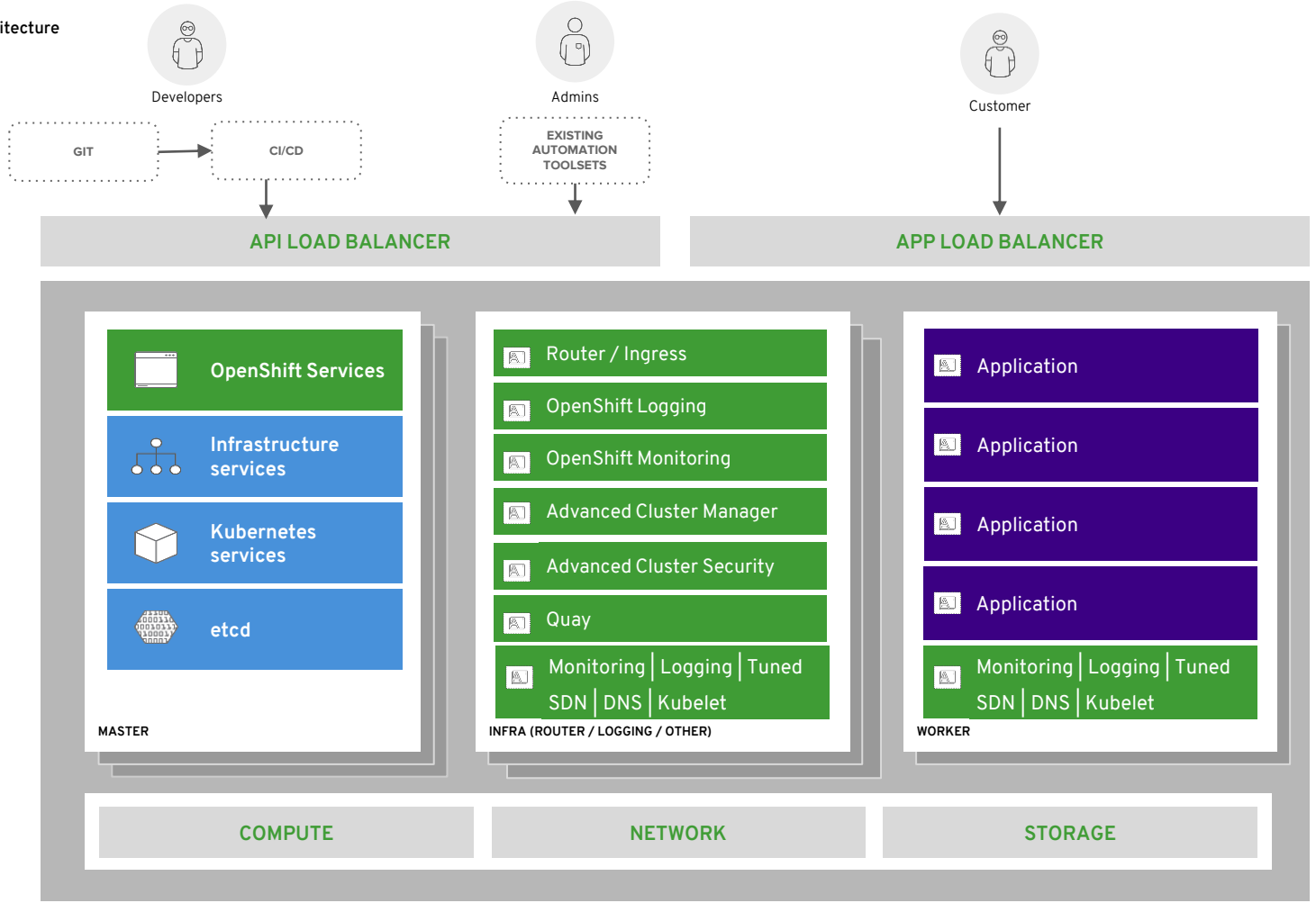


Best IT Ops Experience

CaaS ⟷ PaaS ⟷ FaaS

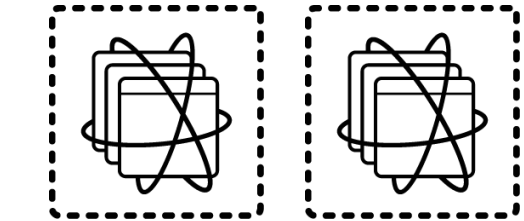
Best Developer Experience

Architecture



Red Hat

WHY DO CONTAINERS NEED KUBERNETES?



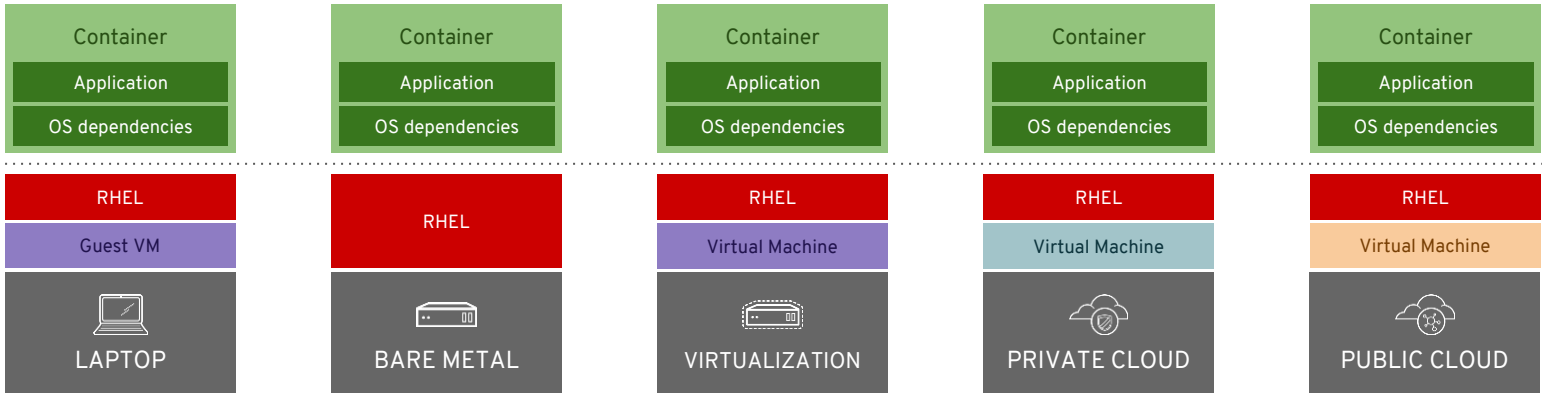
CONTAINERIZED APPLICATIONS

kubernetes

- Rich Ecosystem Diversity
- Infrastructure Management
- Easy to Scale Containers
- High Availability

APPLICATION PORTABILITY WITH CONTAINERS

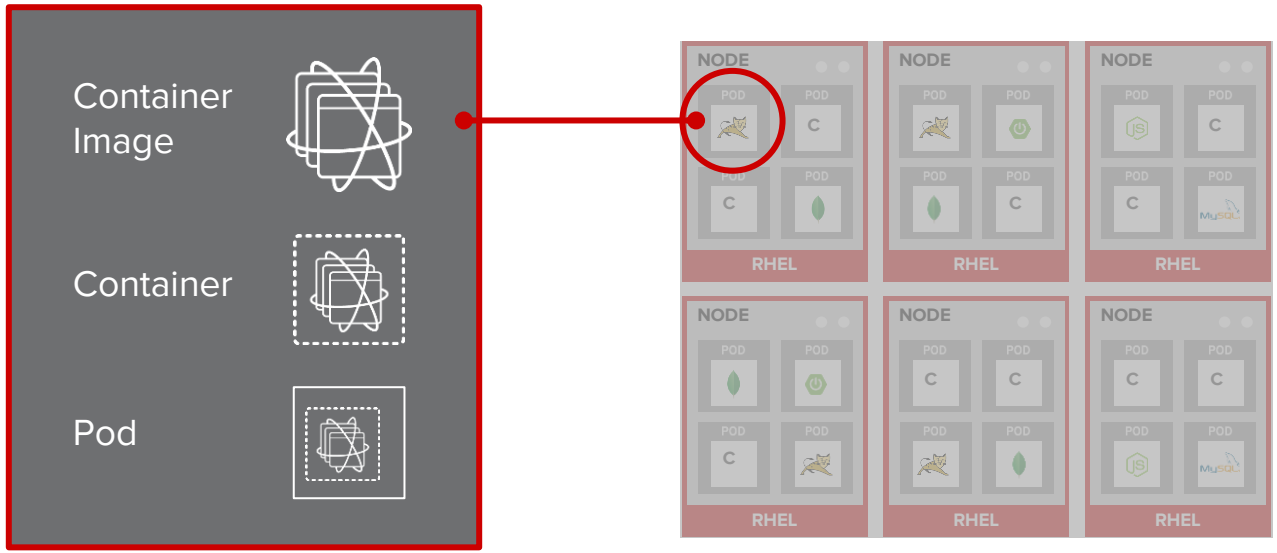
RHEL Containers + RHEL Host = Guaranteed Portability
Across Any Infrastructure



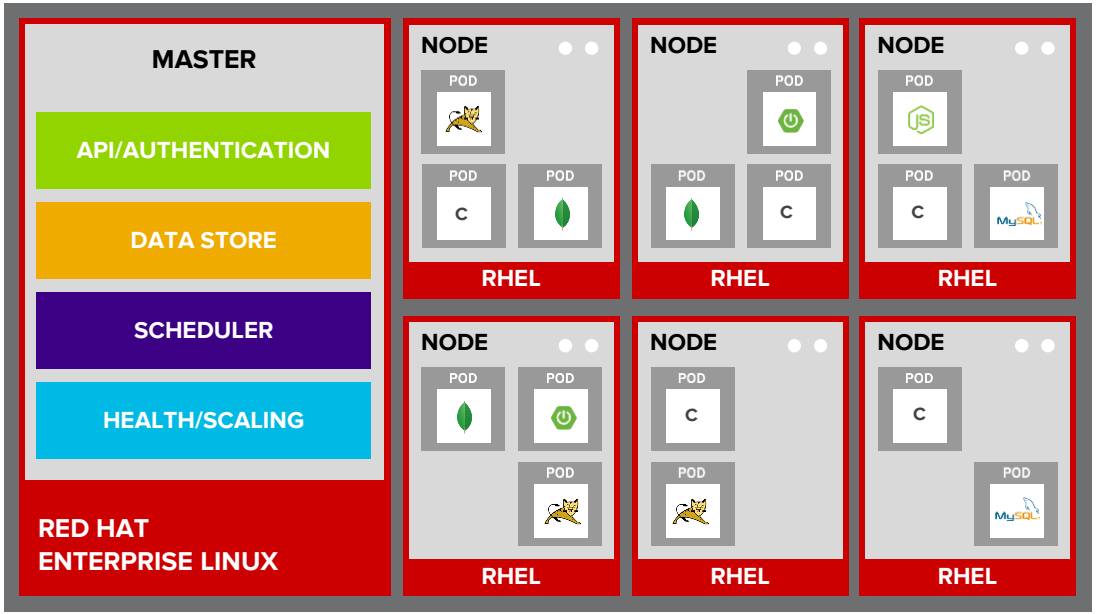
Auto Self-Healing

Red Hat

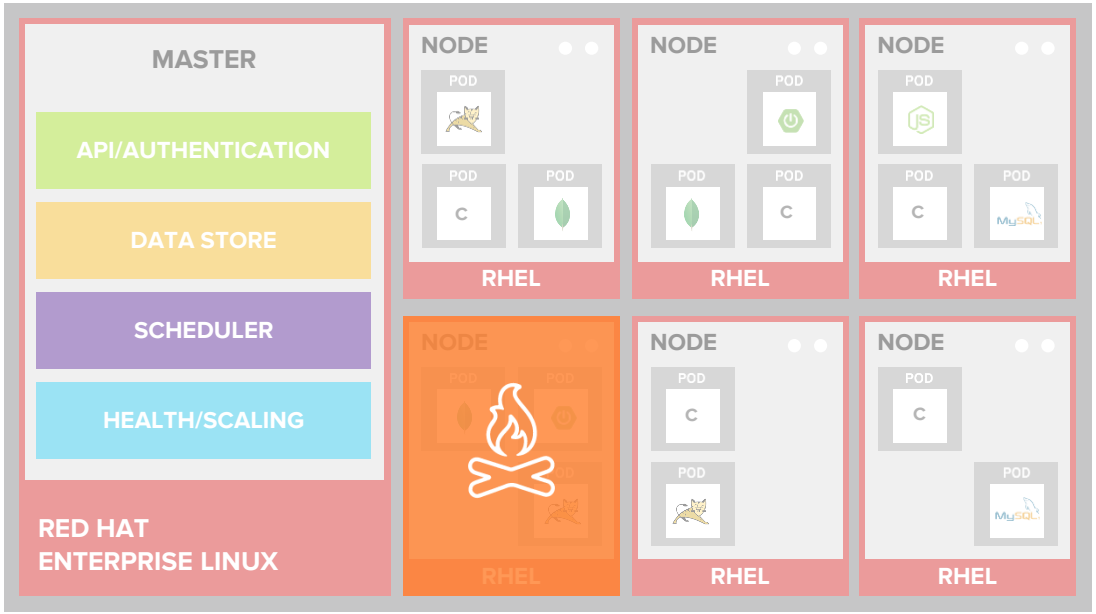
APPS RUN IN CONTAINERS



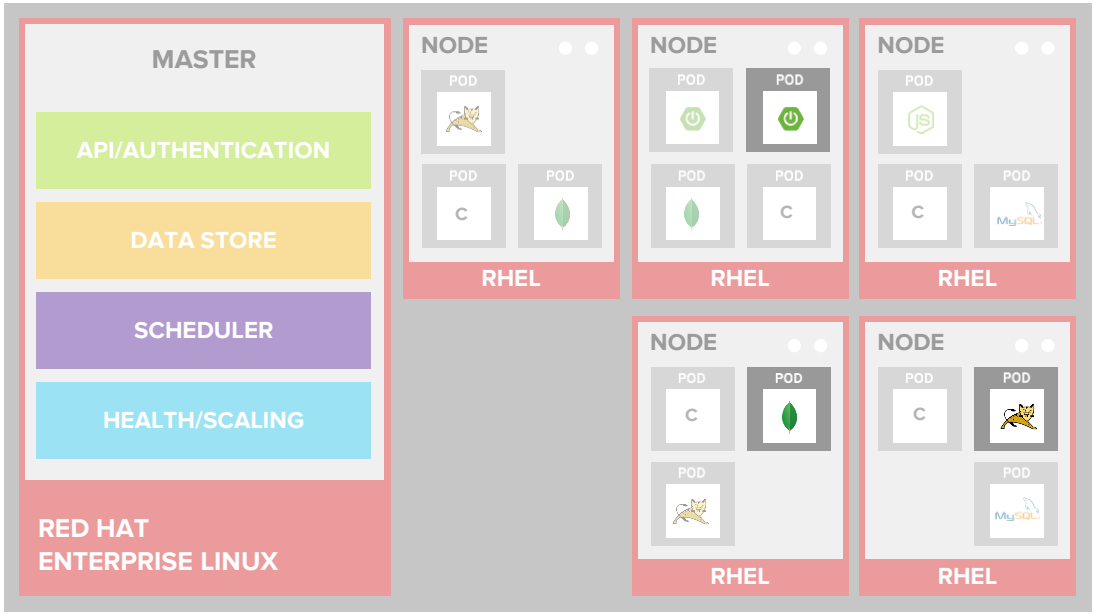
Container Orchestration



AUTO-HEALING FAILED CONTAINERS



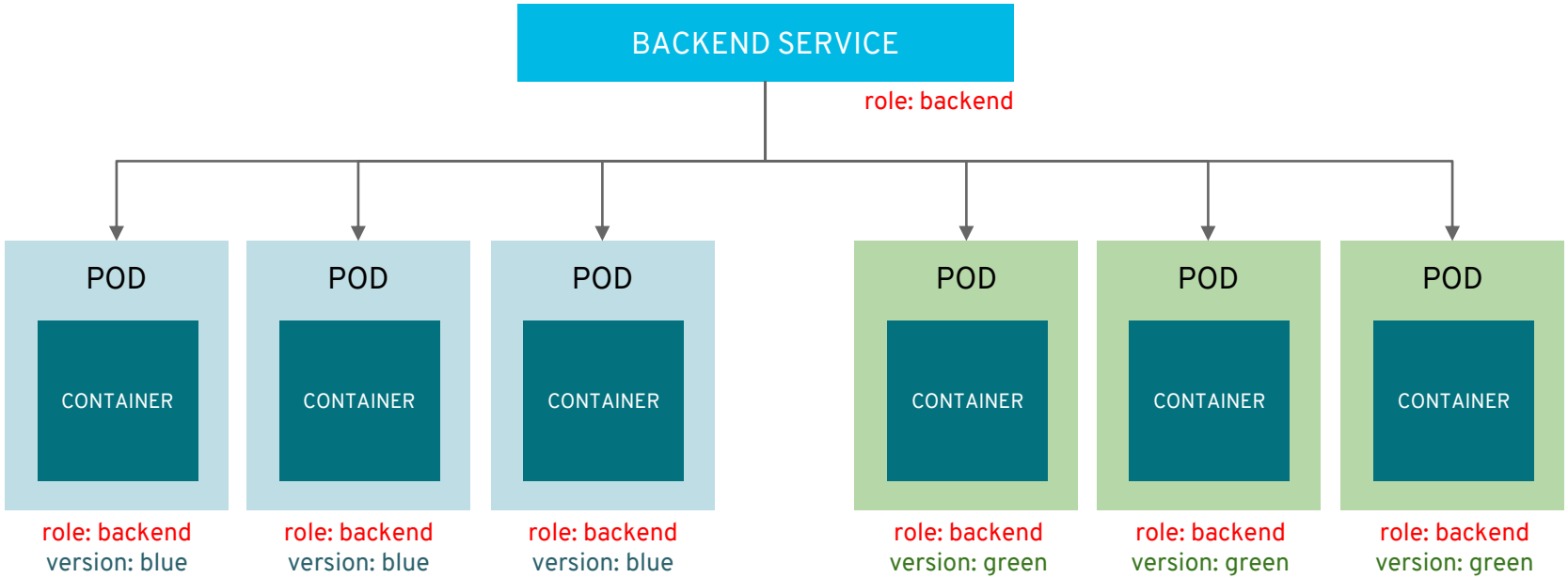
AUTO-HEALING FAILED CONTAINERS



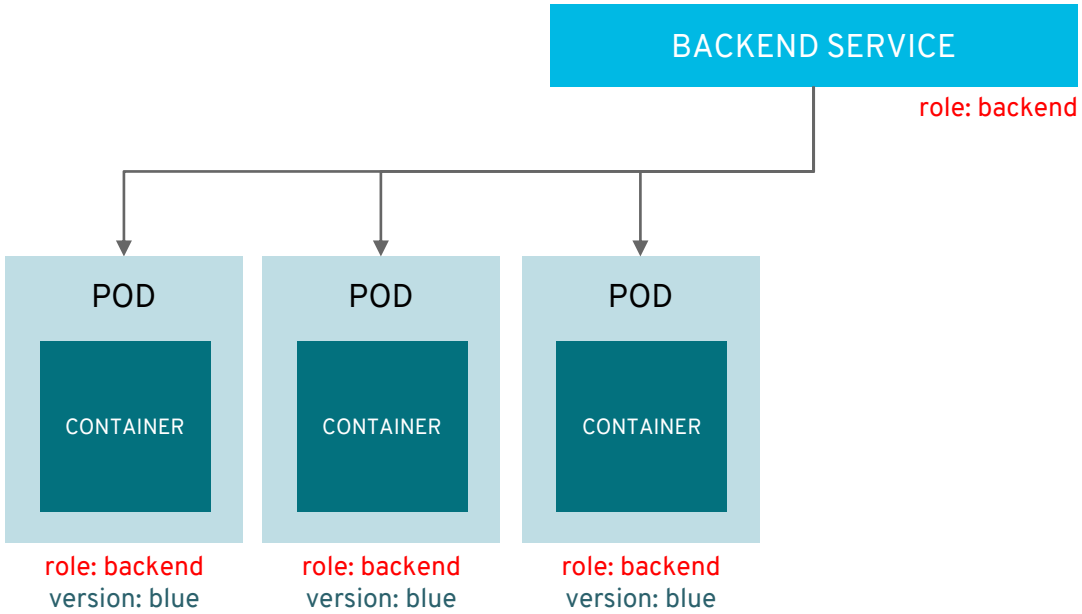
Rolling Upgrade

Red Hat

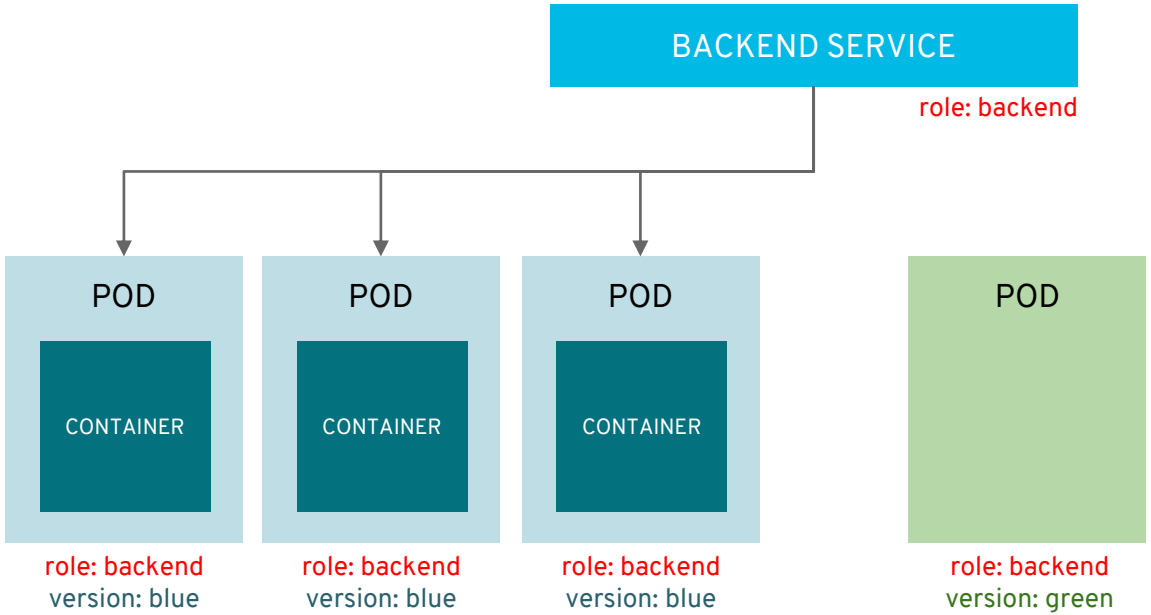
Rolling Upgrade



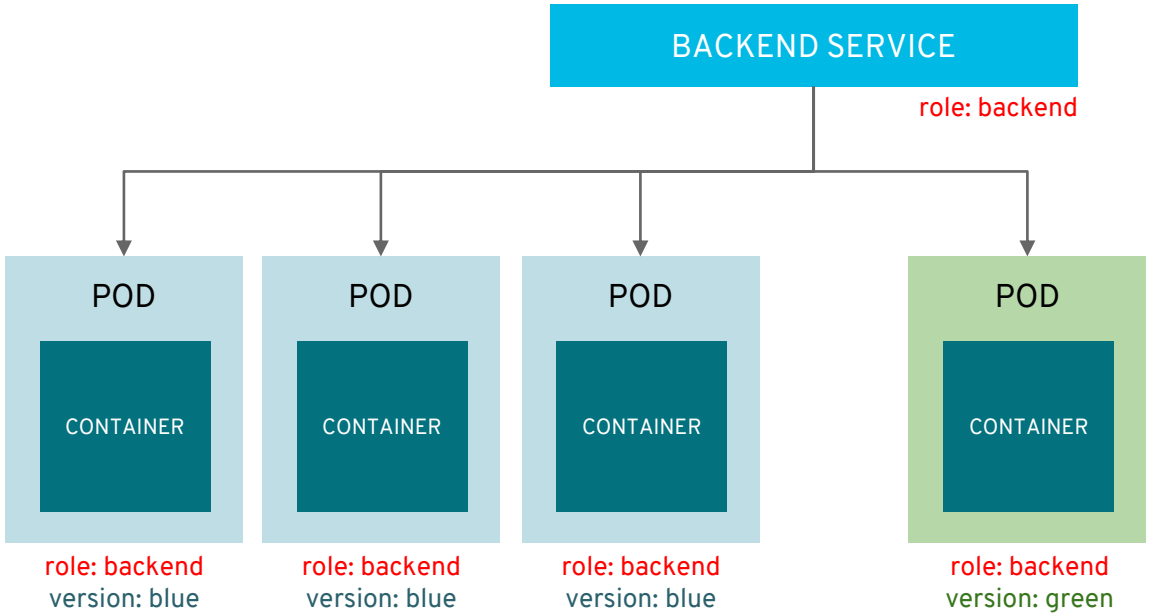
Rolling Upgrade



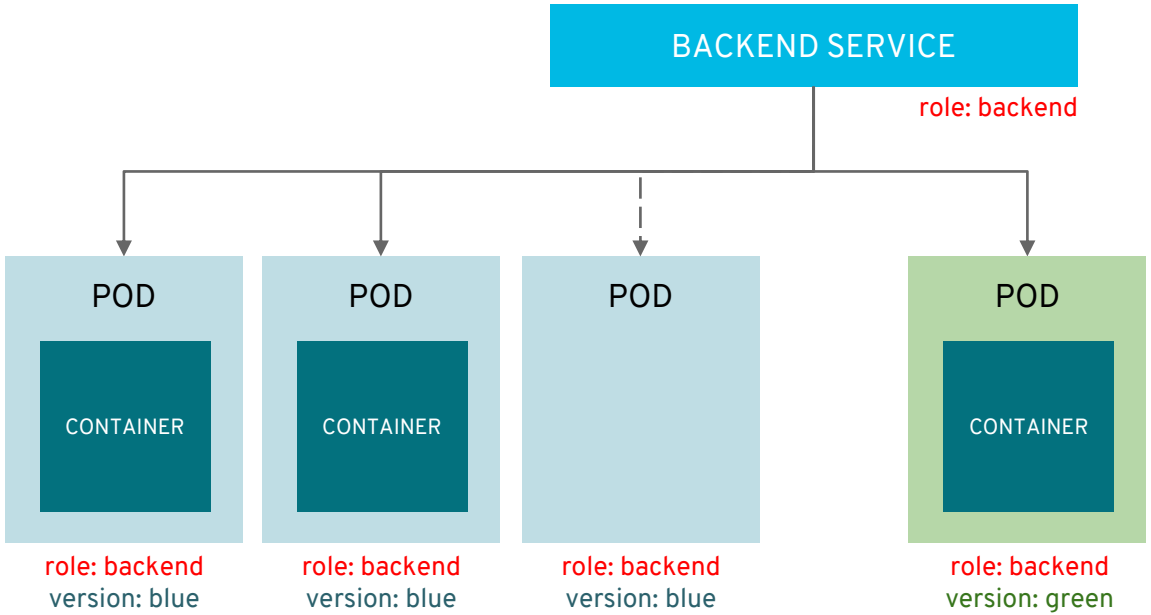
Rolling Upgrade



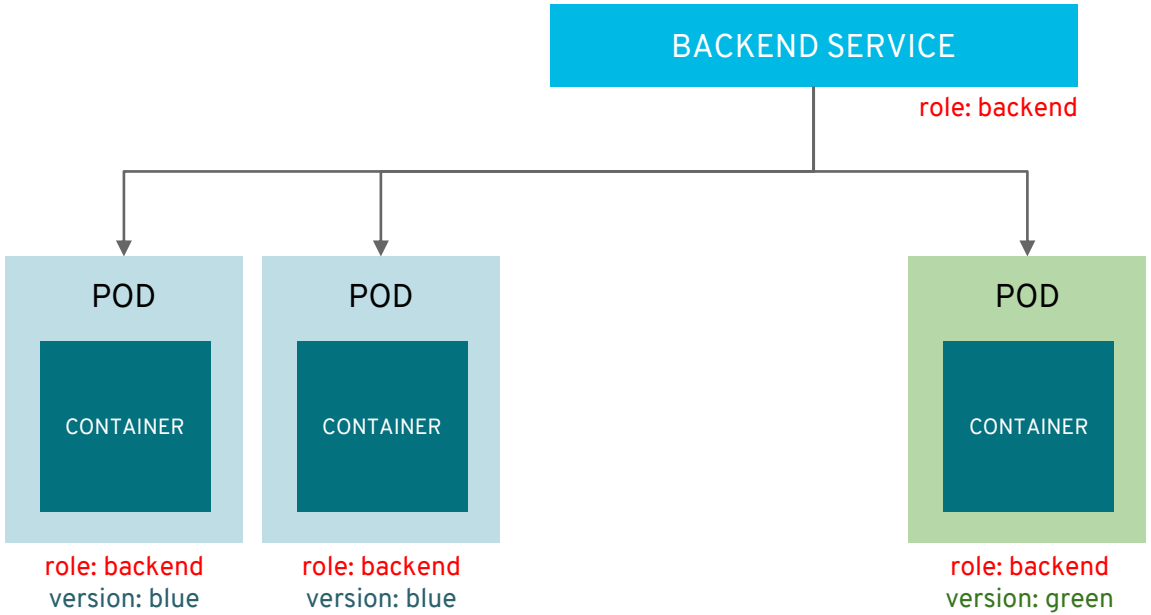
Rolling Upgrade



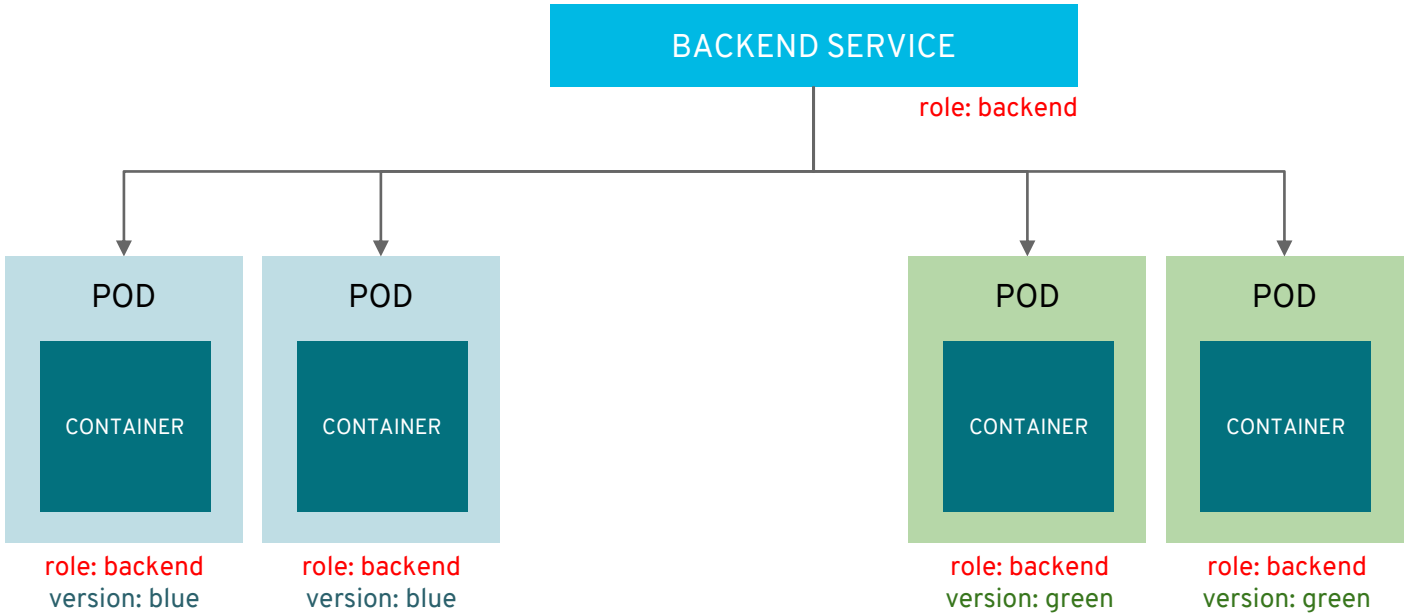
Rolling Upgrade



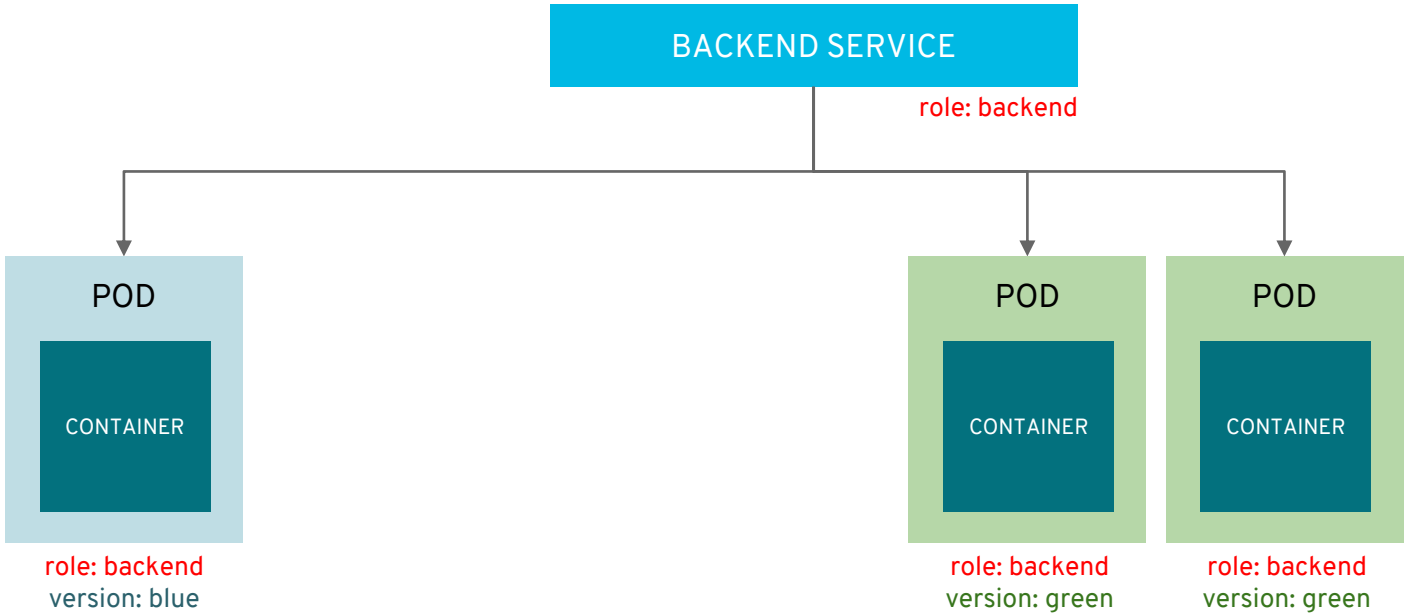
Rolling Upgrade



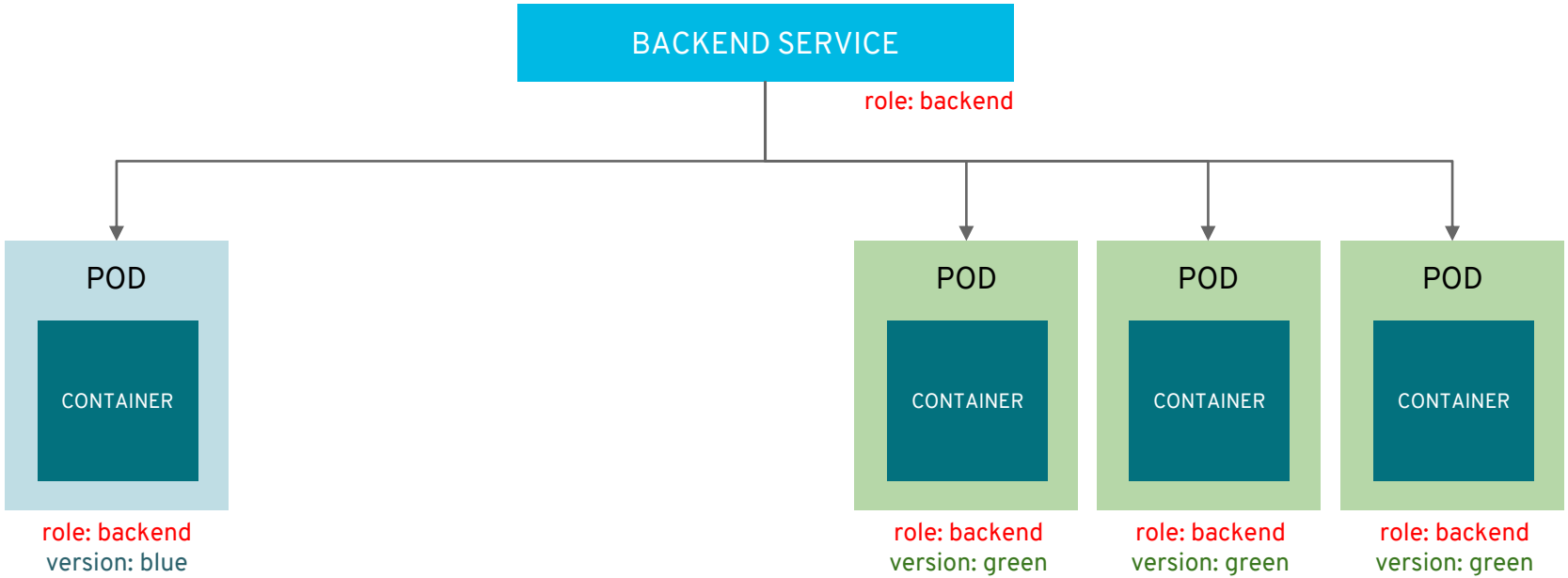
Rolling Upgrade



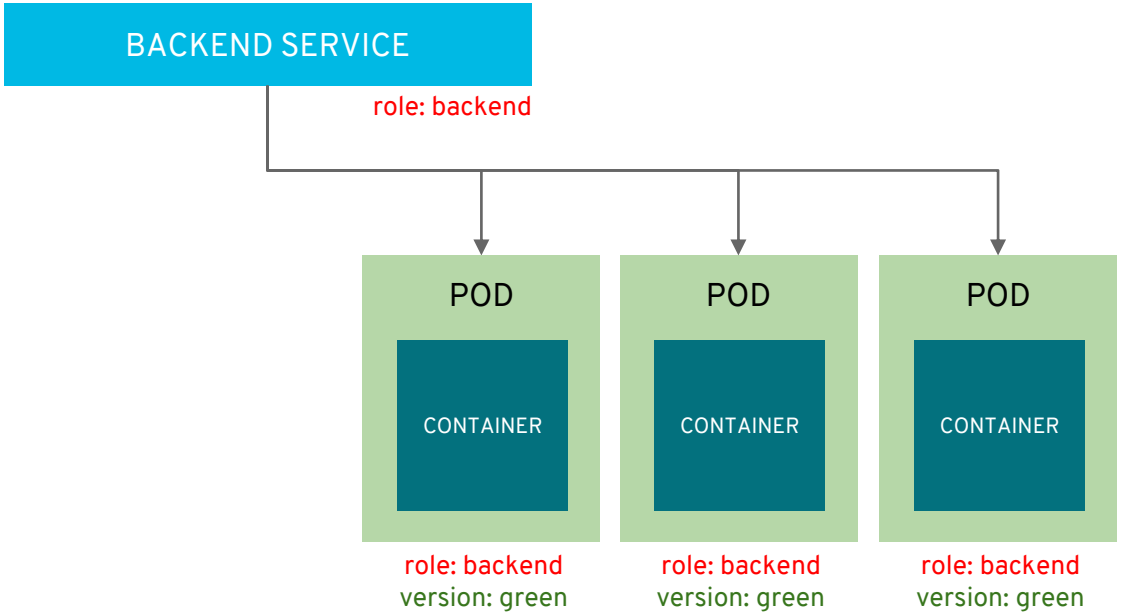
Rolling Upgrade

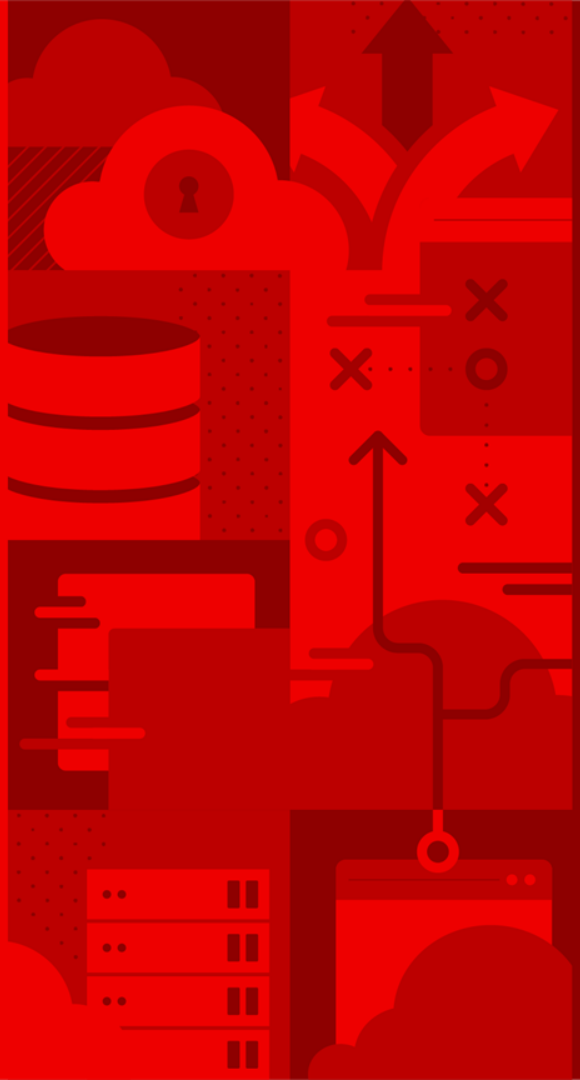


Rolling Upgrade



Rolling Upgrade



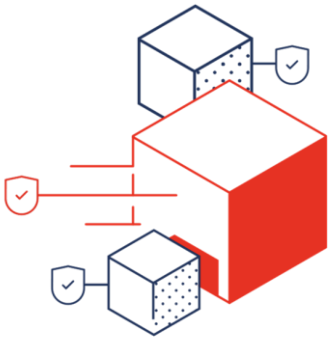


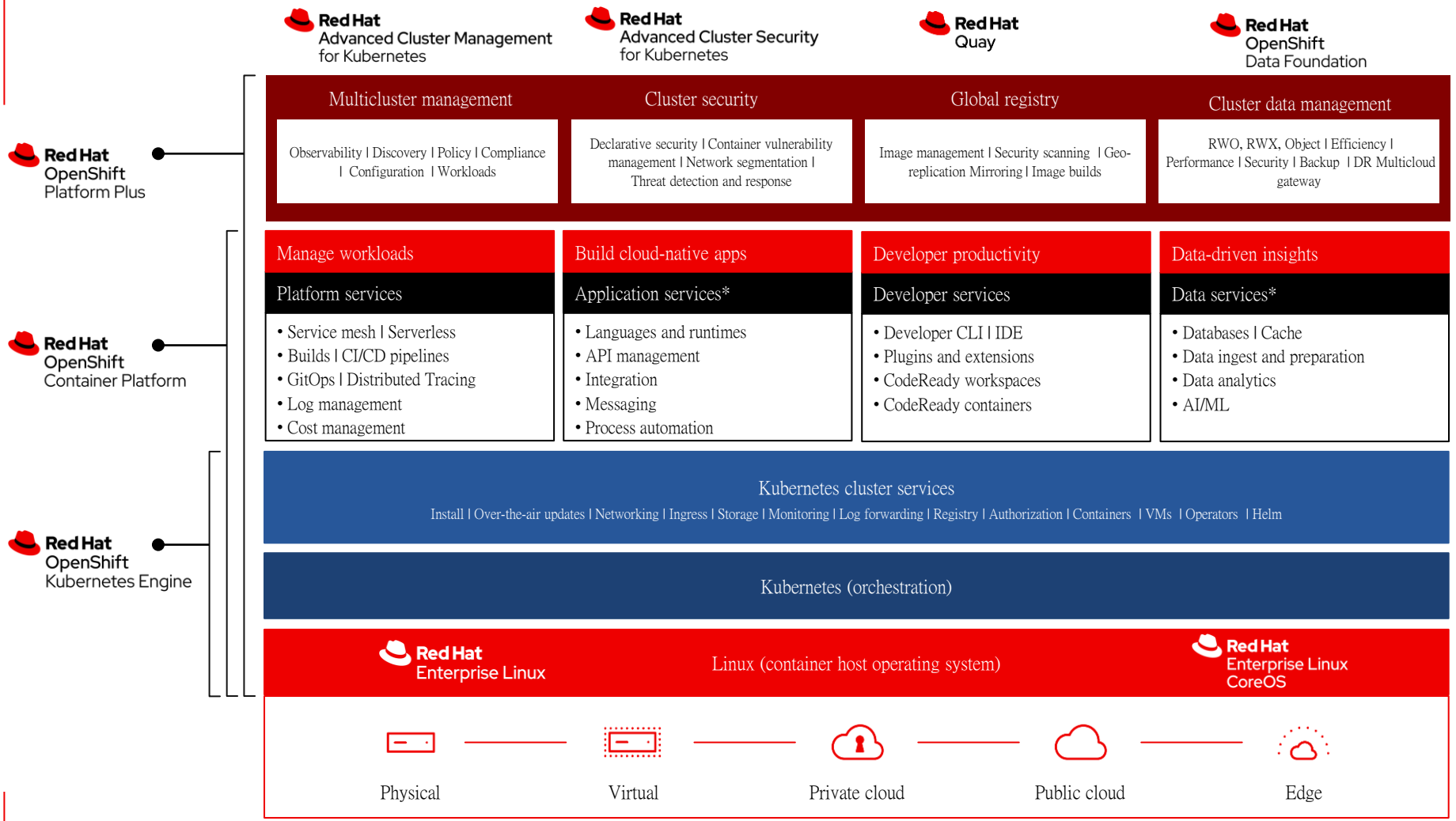
OpenShift Platform Plus

Red Hat

The OpenShift platform vision:

A single hybrid-cloud platform for enterprises to build, deploy, run and manage intelligent applications securely at scale.




Red Hat Advanced Cluster Management for Kubernetes
Red Hat Advanced Cluster Security for Kubernetes
Red Hat Quay
Red Hat OpenShift Data Foundation
Red Hat OpenShift Platform Plus
Multicluster management
Observability | Discovery | Policy | Compliance | Configuration | Workloads
Cluster security
Declarative security | Container vulnerability management | Network segmentation | Threat detection and response
Global registry
Image management | Security scanning | Geo-replication Mirroring | Image builds
Cluster data management
RWO, RWX, Object | Efficiency | Performance | Security | Backup | DR Multicloud gateway
Red Hat OpenShift Container Platform
Manage workloads
Platform services
• Service mesh | Serverless
• Builds | CI/CD pipelines
• GitOps | Distributed Tracing
• Log management
• Cost management
Build cloud-native apps
Application services*
• Languages and runtimes
• API management
• Integration
• Messaging
• Process automation
Developer productivity
Developer services
• Developer CLI | IDE
• Plugins and extensions
• CodeReady workspaces
• CodeReady containers
Data-driven insights
Data services*
• Databases | Cache
• Data ingest and preparation
• Data analytics
• AI/ML
Red Hat OpenShift Kubernetes Engine
Kubernetes cluster services
Install | Over-the-air updates | Networking | Ingress | Storage | Monitoring | Log forwarding | Registry | Authorization | Containers | VMs | Operators | Helm
Kubernetes (orchestration)
Red Hat Enterprise Linux
Linux (container host operating system)
Red Hat Enterprise Linux CoreOS
Physical
Virtual
Private cloud
Public cloud
Edge

Red Hat OpenShift Self-Managed Editions

Red Hat OpenShift Kubernetes

Essential enterprise Kubernetes infrastructure

Includes:

- Enterprise Kubernetes runtime
- RHEL CoreOS immutable container OS
- Administrator console
- OpenShift Virtualization

Red Hat OpenShift Container Platform

Complete application development platform

Adds:

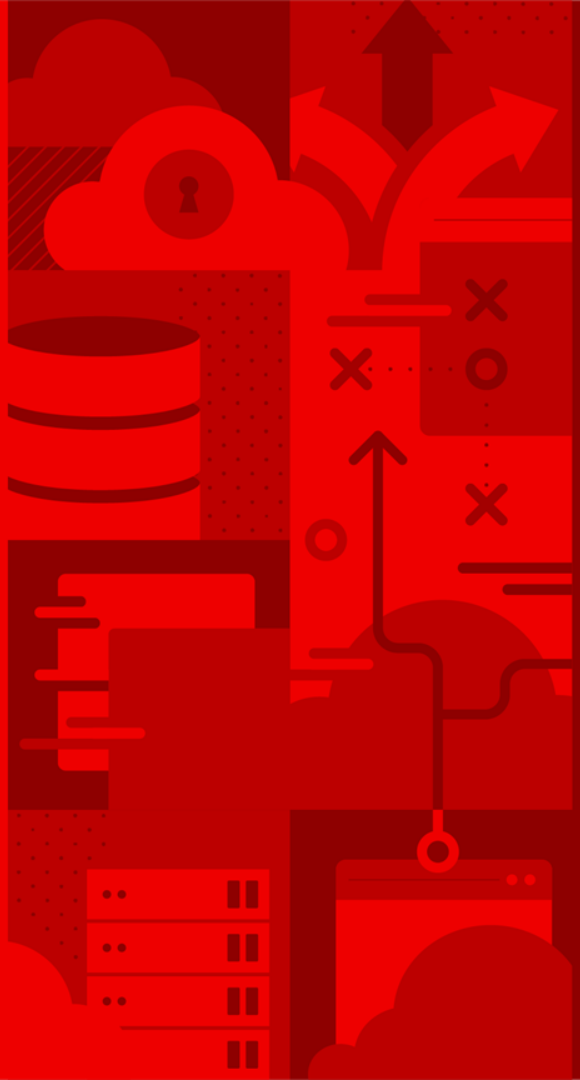
- Developer Console
- Log Mgt & Metering
- Serverless (Knative)
- Service Mesh (Istio)
- Pipelines & GitOps (Tekton, ArgoCD)
- Insights for OpenShift (Cost, Subscription, Advisor)

Red Hat OpenShift Platform Plus

Manageability and consistency across hybrid , multi cloud or on -prem data centers with advanced security & governance

Adds:

- Multicluster management
- Advanced observability and policy compliance
- Declarative security
- Threat detection and response
- Scalable global container registry
- Storage Management

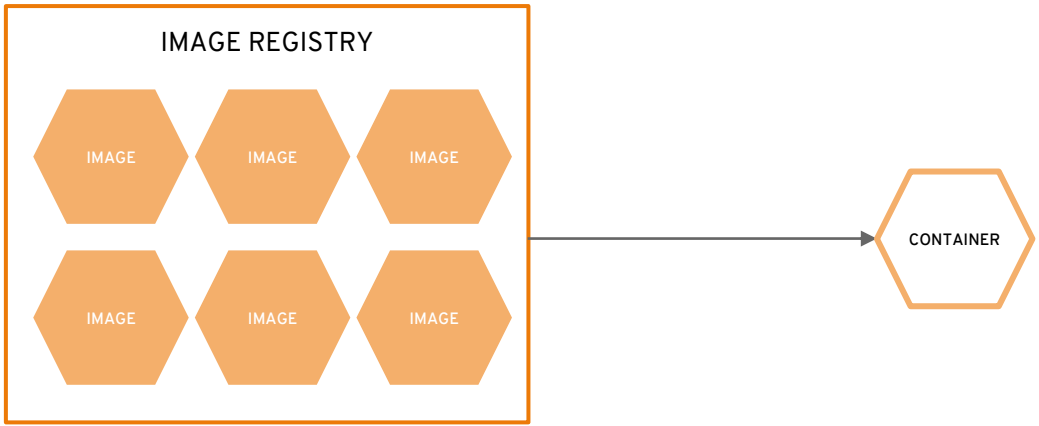


Kubernetes core concepts

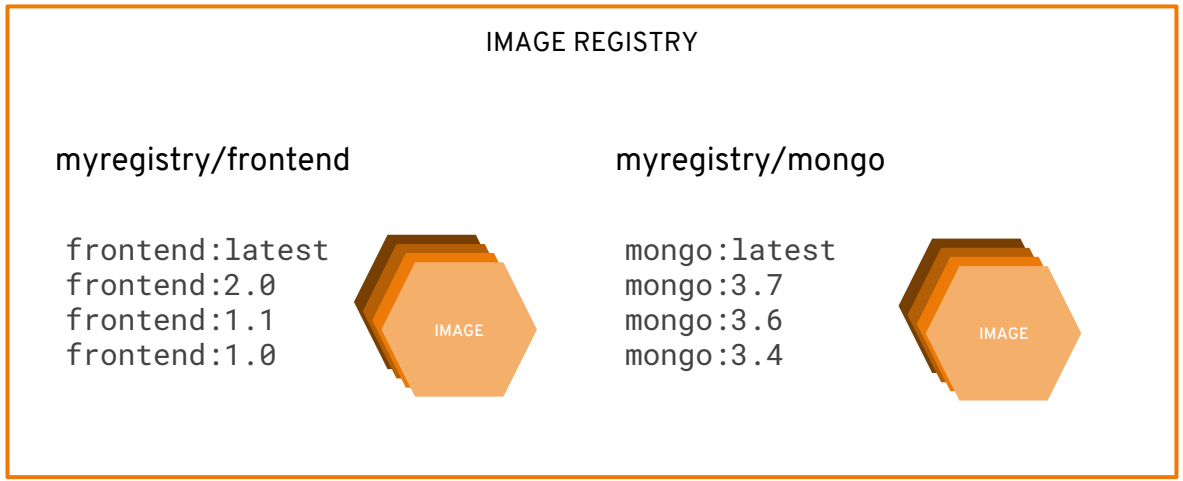
a container is the smallest compute unit

containers are created from container images

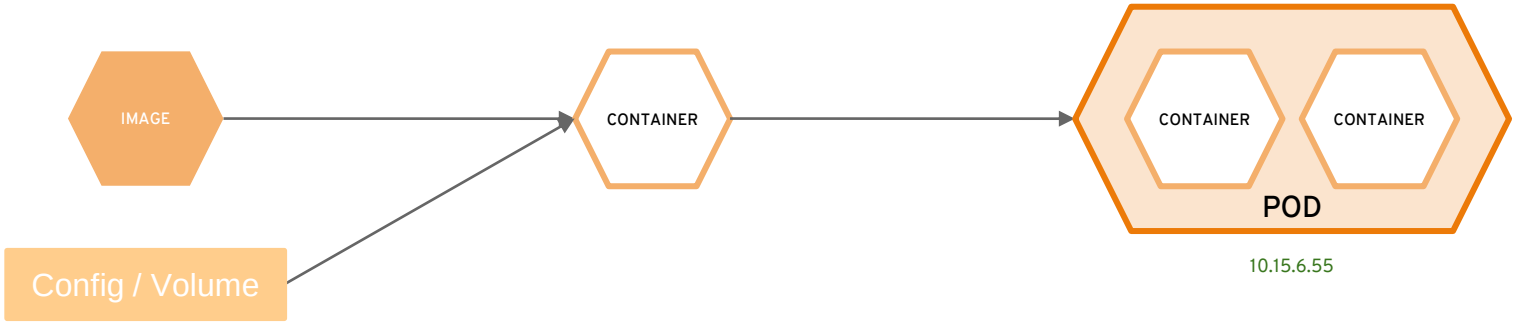
container images are stored in an image registry (鏡像倉庫)



an image repository contains all versions of an image in the image registry

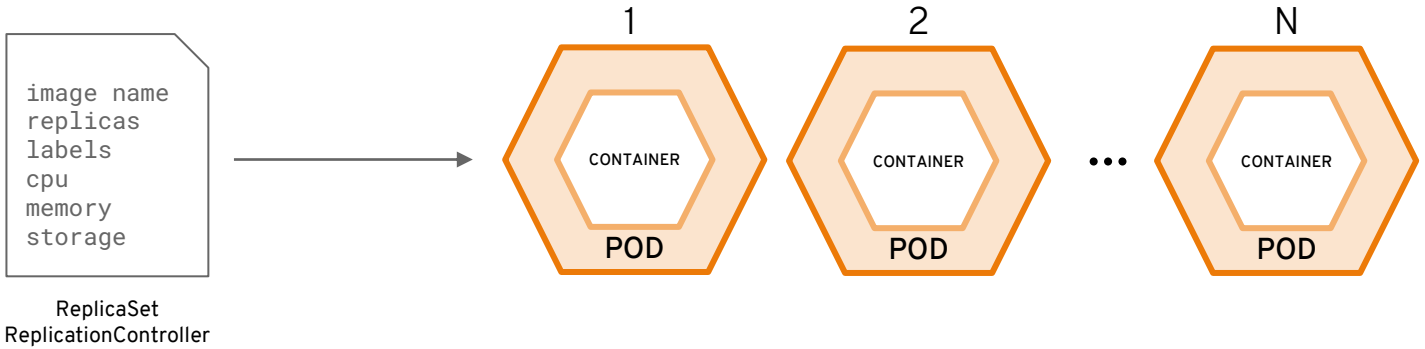


everything runs in pods

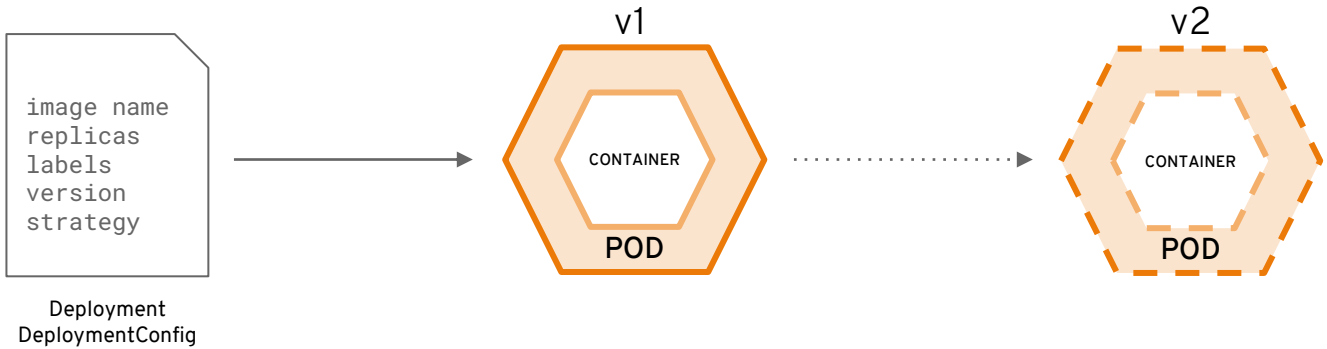


Pod 是容器部署及管理的最小單位

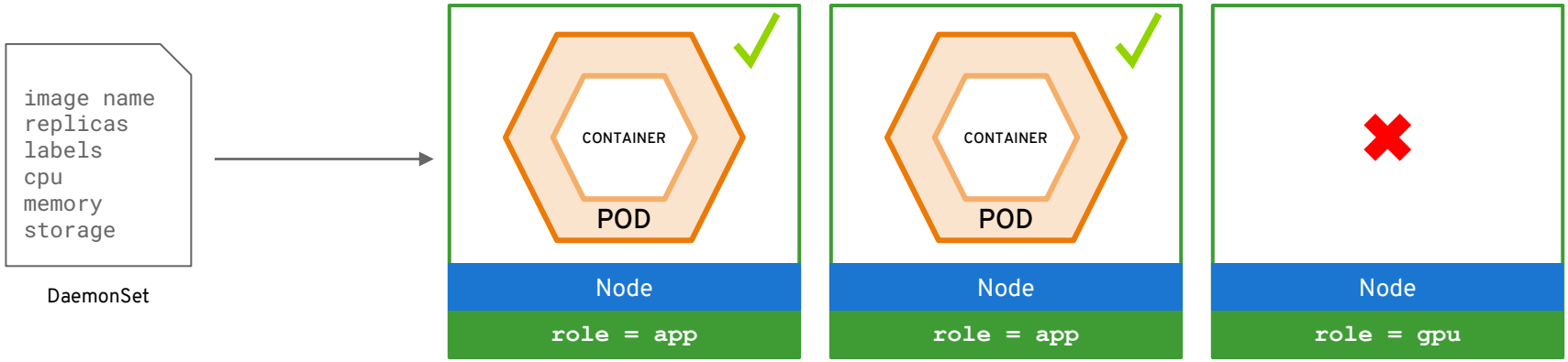
~~ReplicationControllers~~ & ReplicaSets 負責維持Pod在執行時期達到我們所指定的運行數量



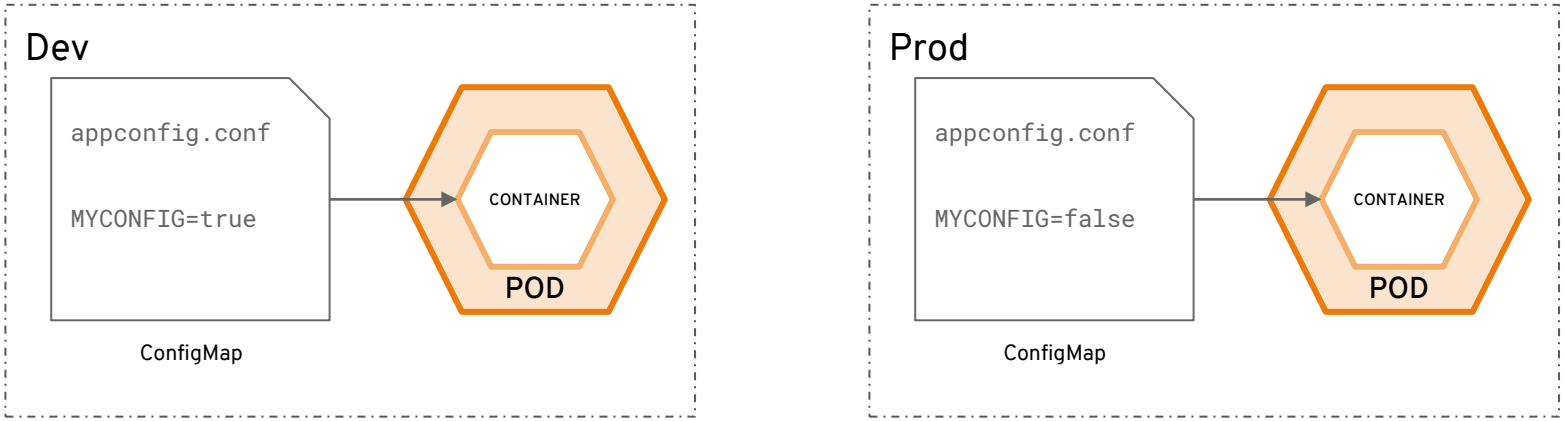
Deployments and ~~DeploymentConfigurations~~ 定義如何部署 Pods



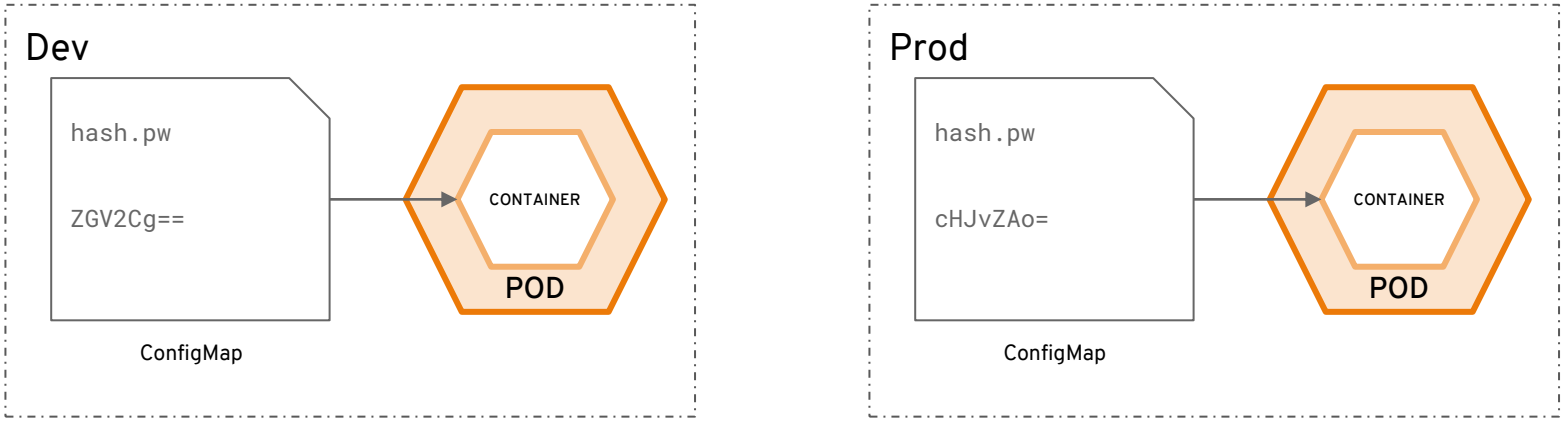
DaemonSet 定義在單一/複數節點上運行一至多個Pod



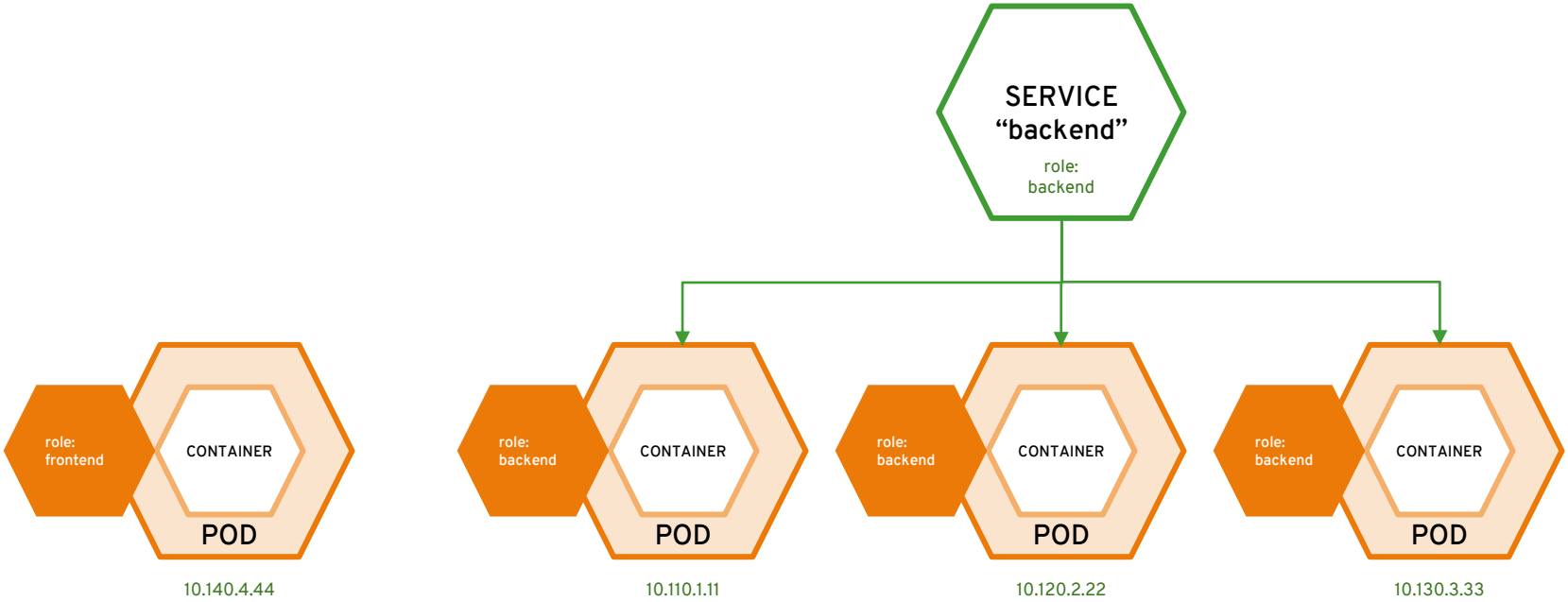
configmaps 讓設定資料和 image 內容分離增加



secrets 提供敏感資料不落地機制

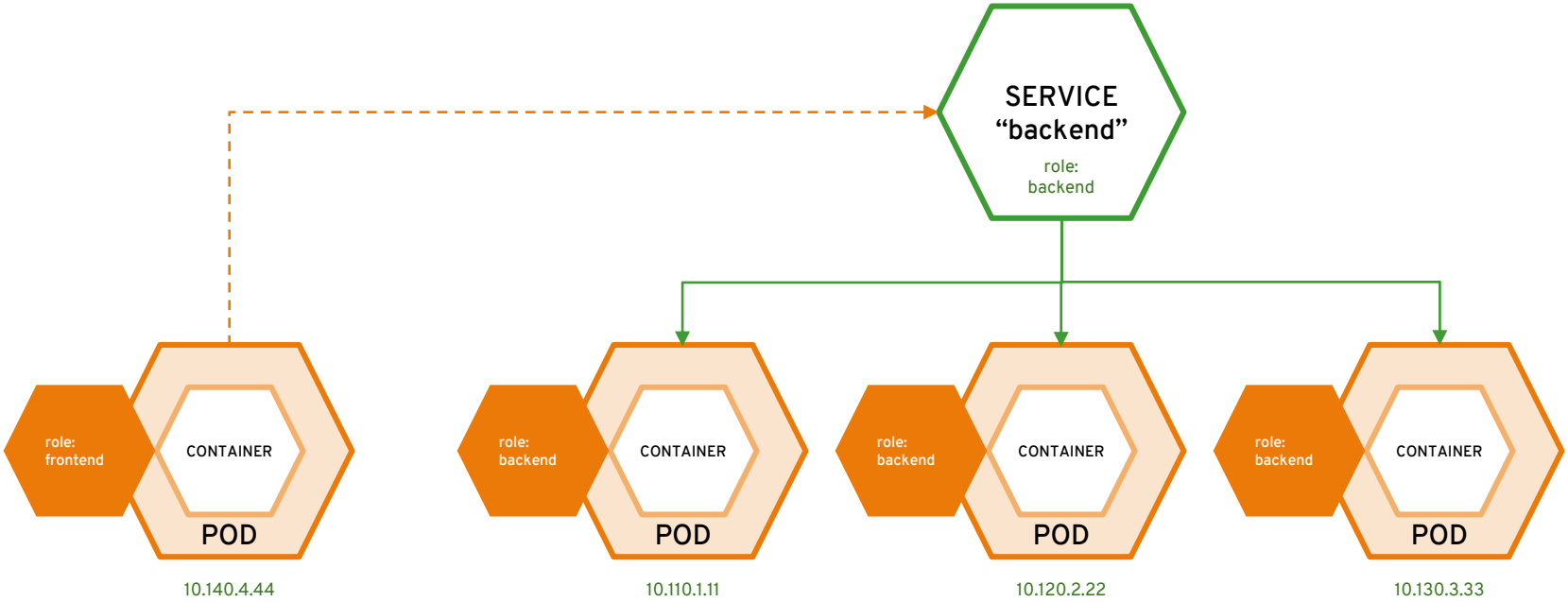


services 在 Pod 之間提供 load-balancing 和 service discovery 機制

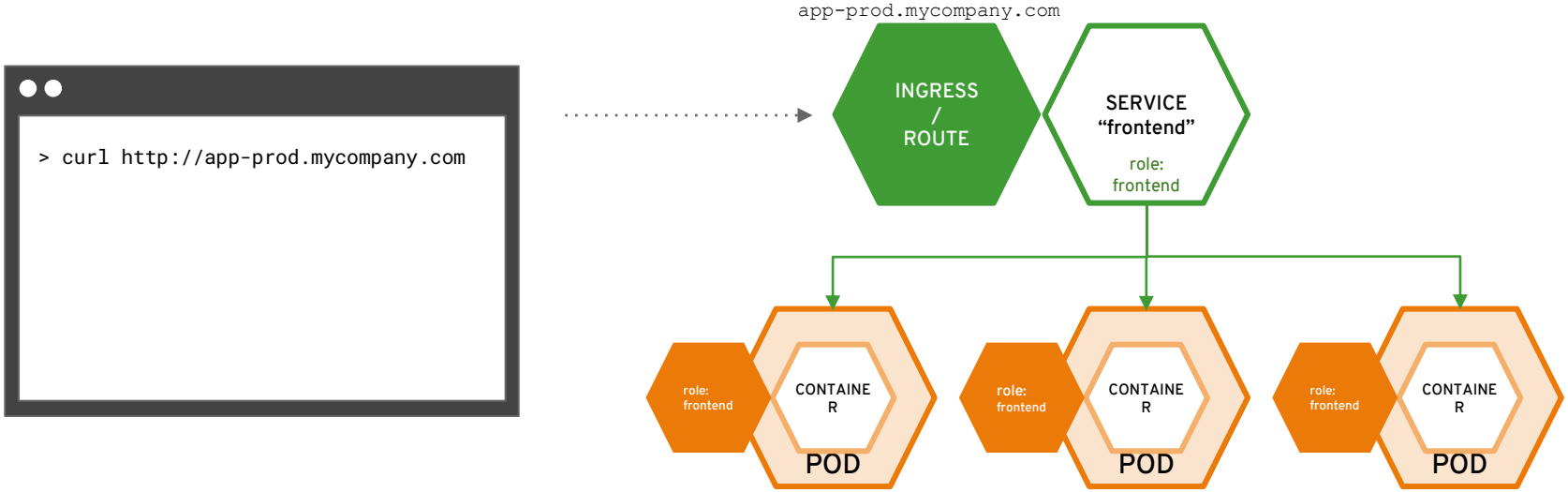


Red Hat

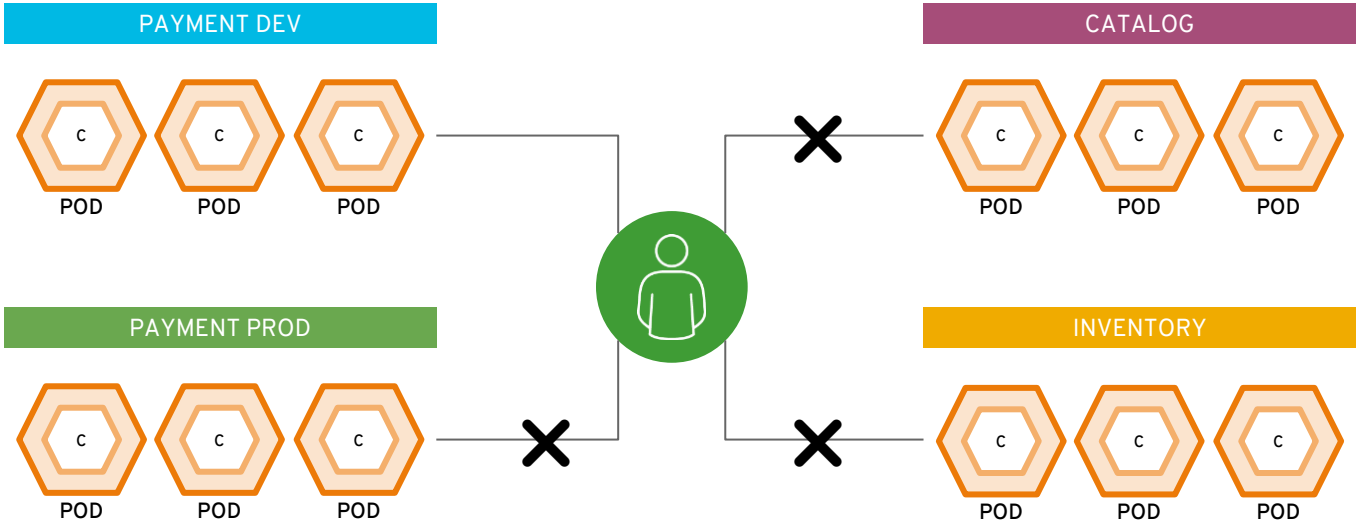
應用程式間可透過 services 溝通

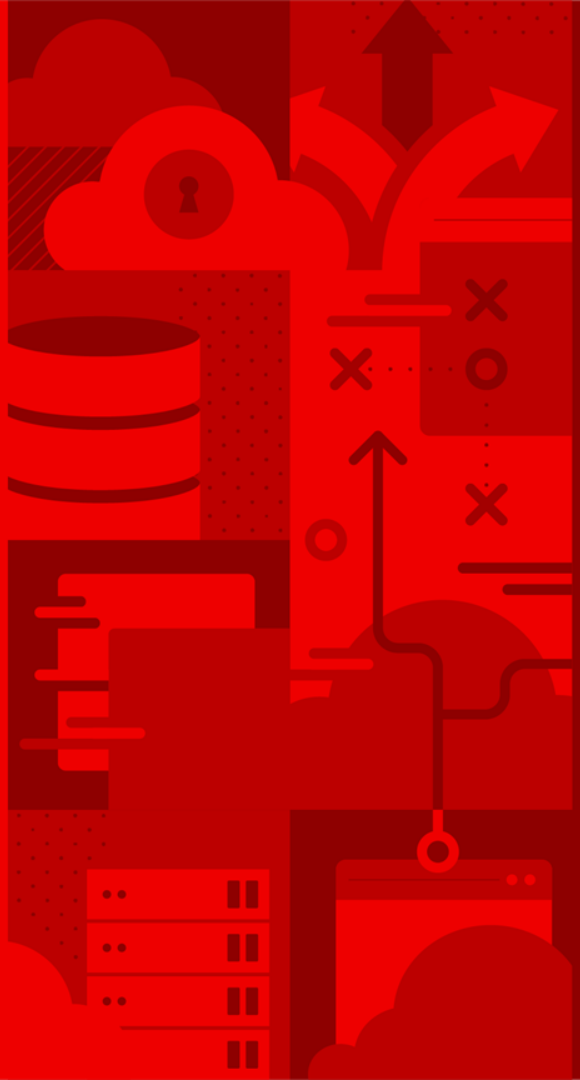


ingress/routes 提供服務對外 url 接口



projects 隔離不同部門/組織間的應用程式執行環境





OpenShift 4 Architecture

your choice of infrastructure

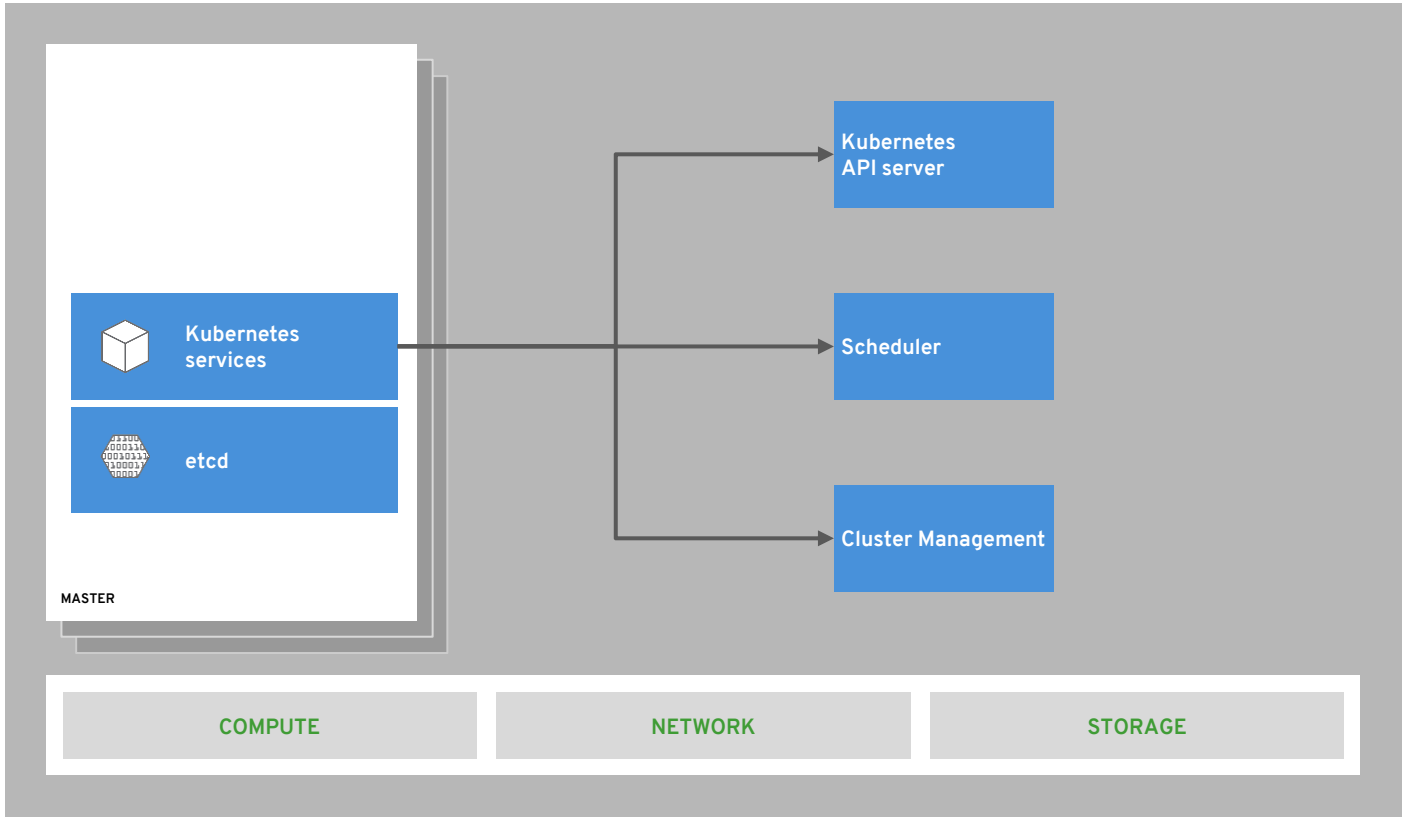
COMPUTE

NETWORK

STORAGE

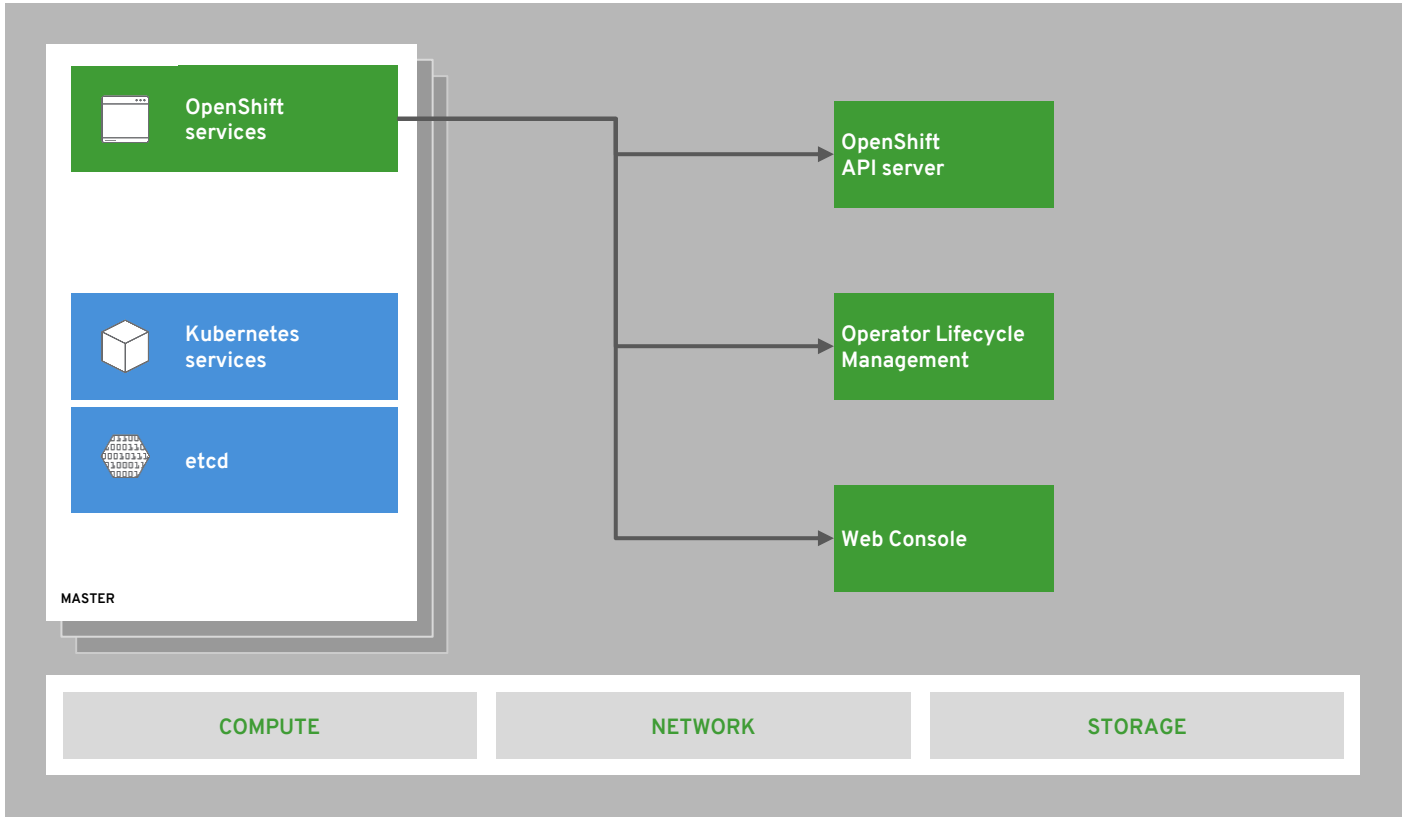
state of everything

core kubernetes components

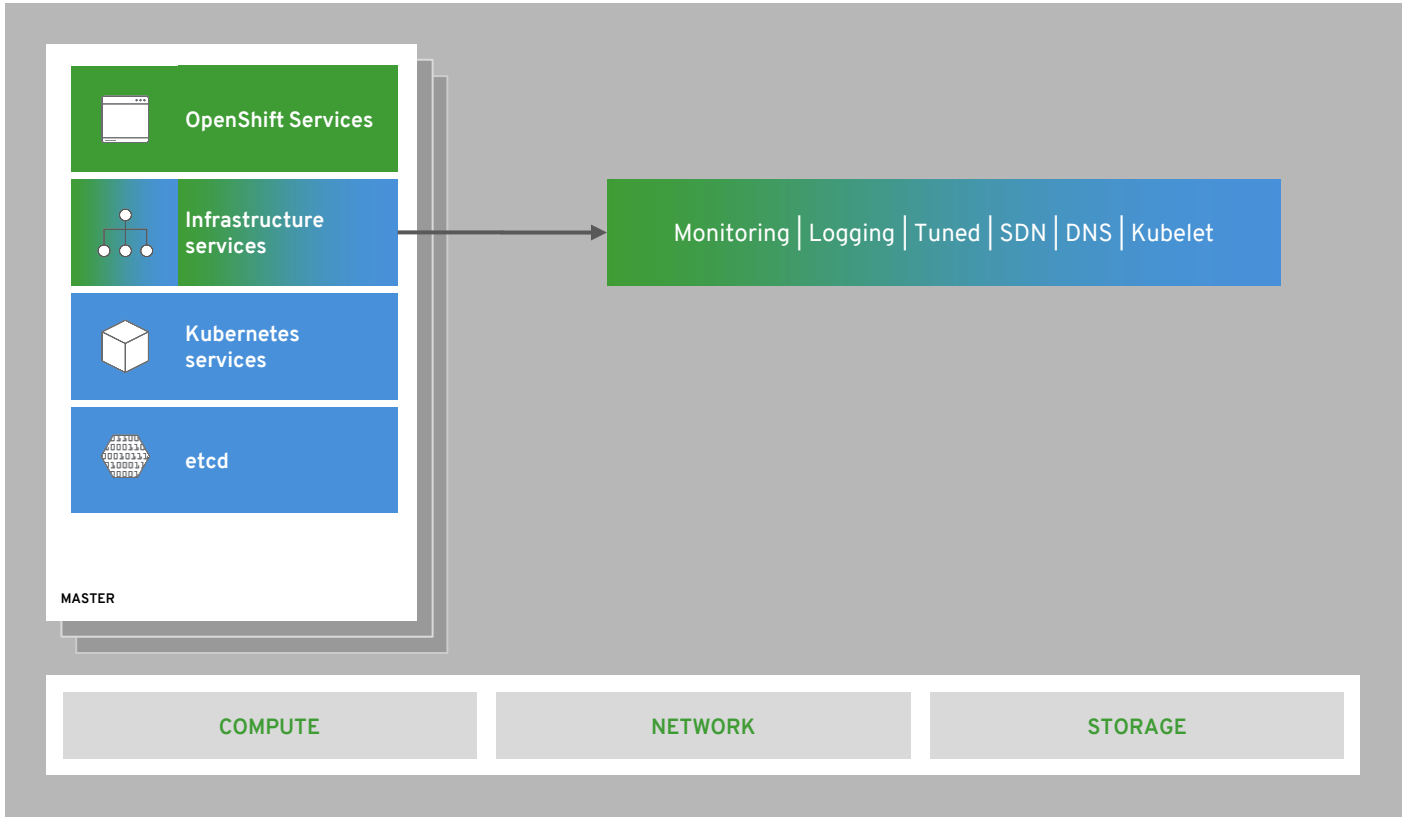


Red Hat

core OpenShift components

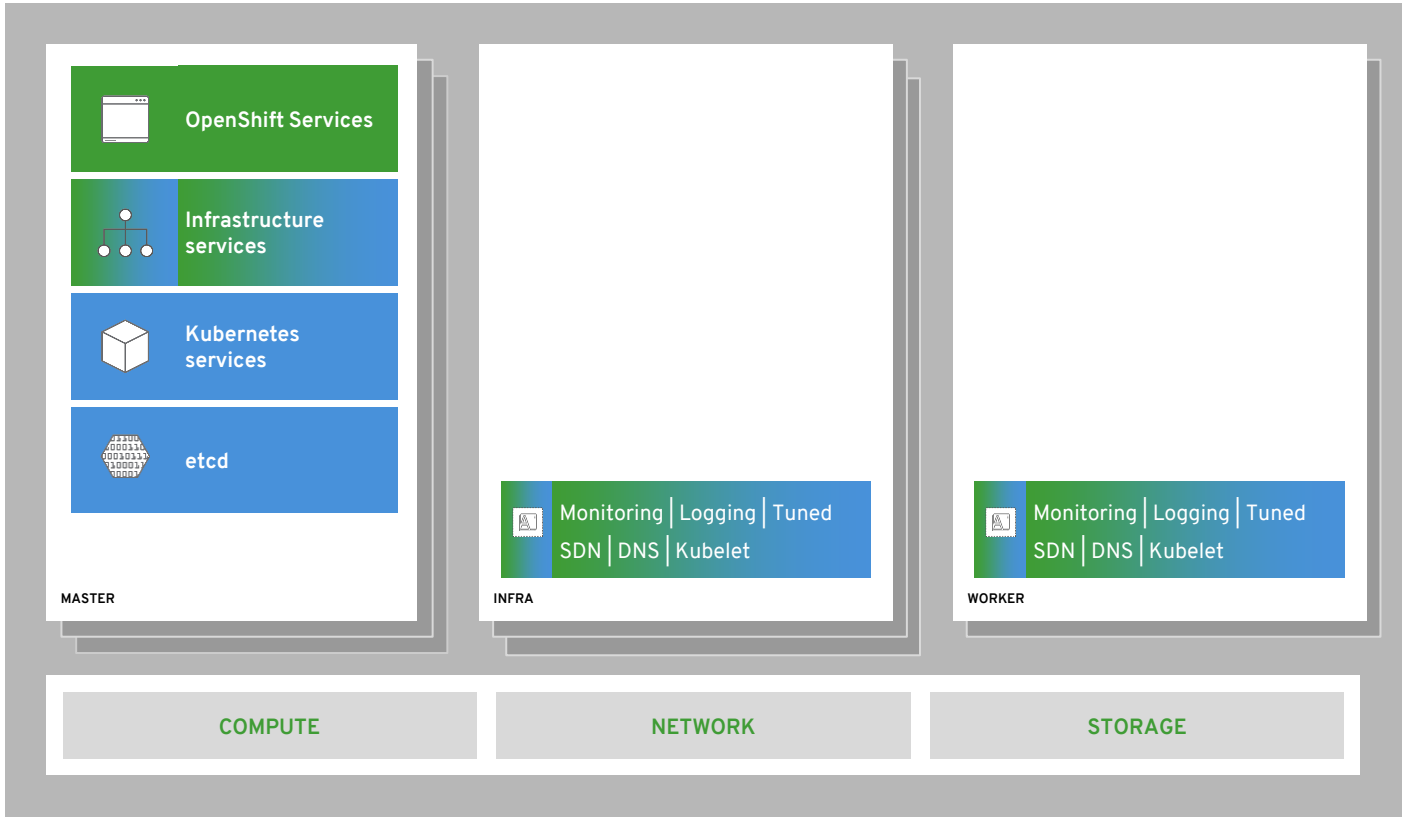


internal and support infrastructure services



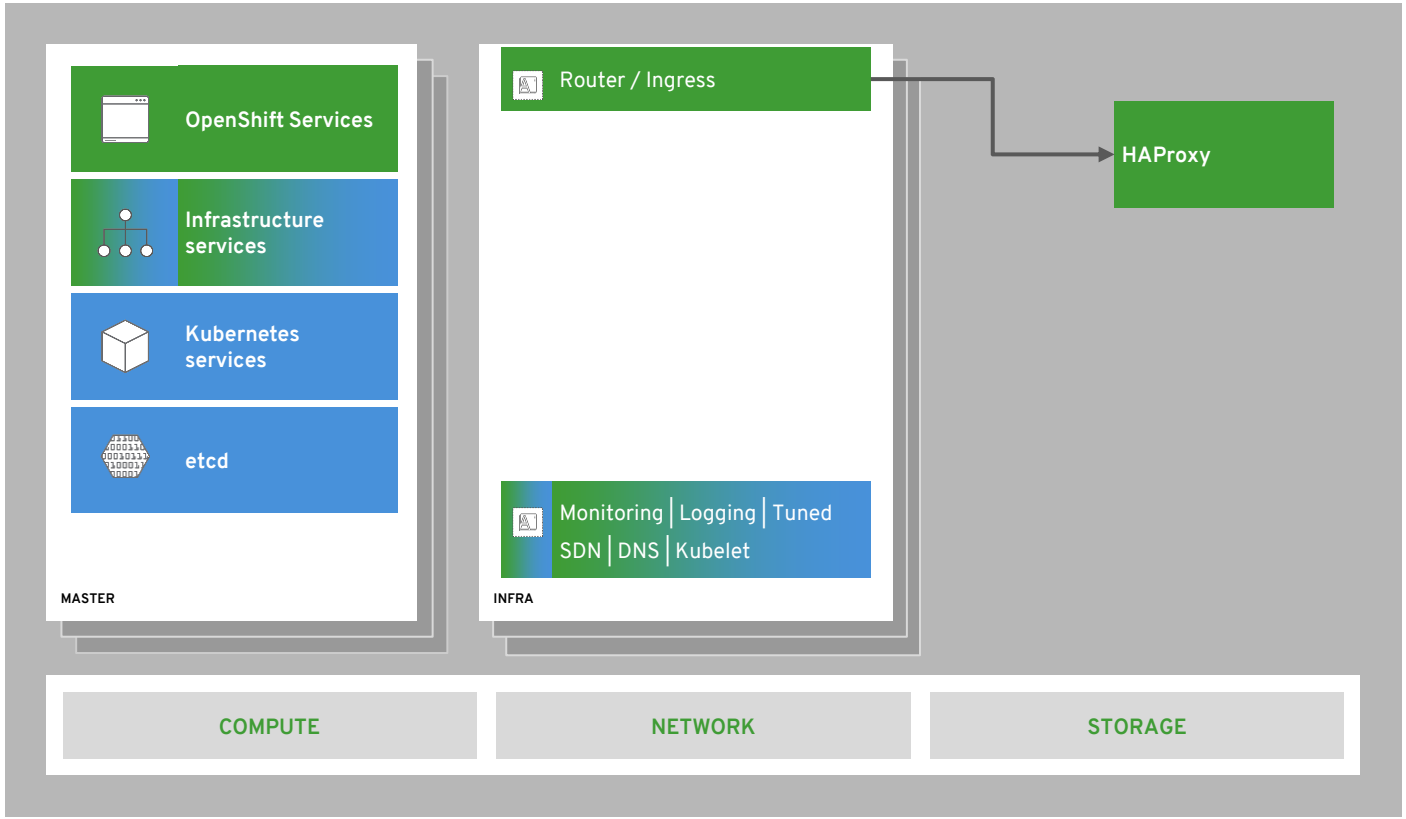
Red Hat

run on all hosts



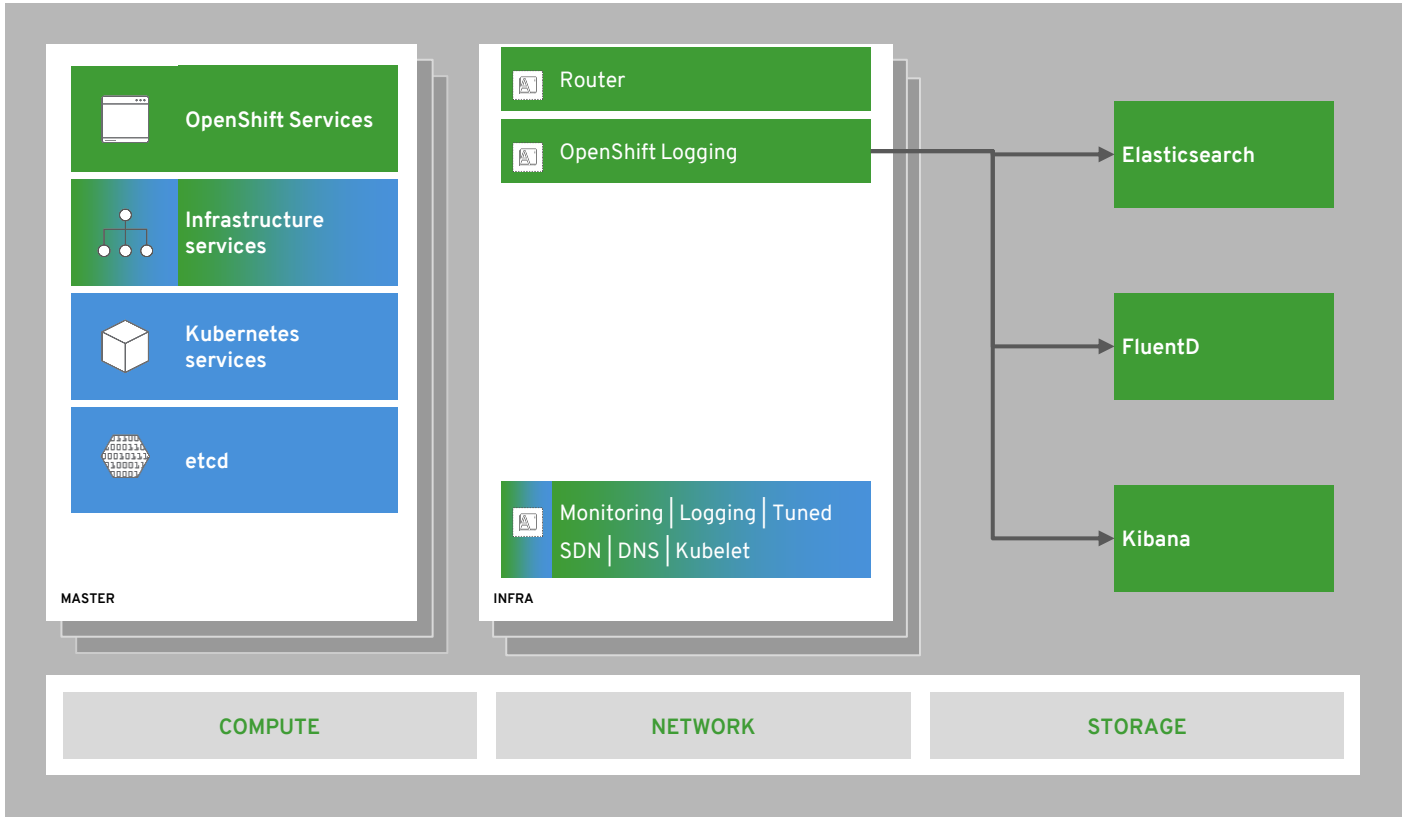
Red Hat

integrated routing



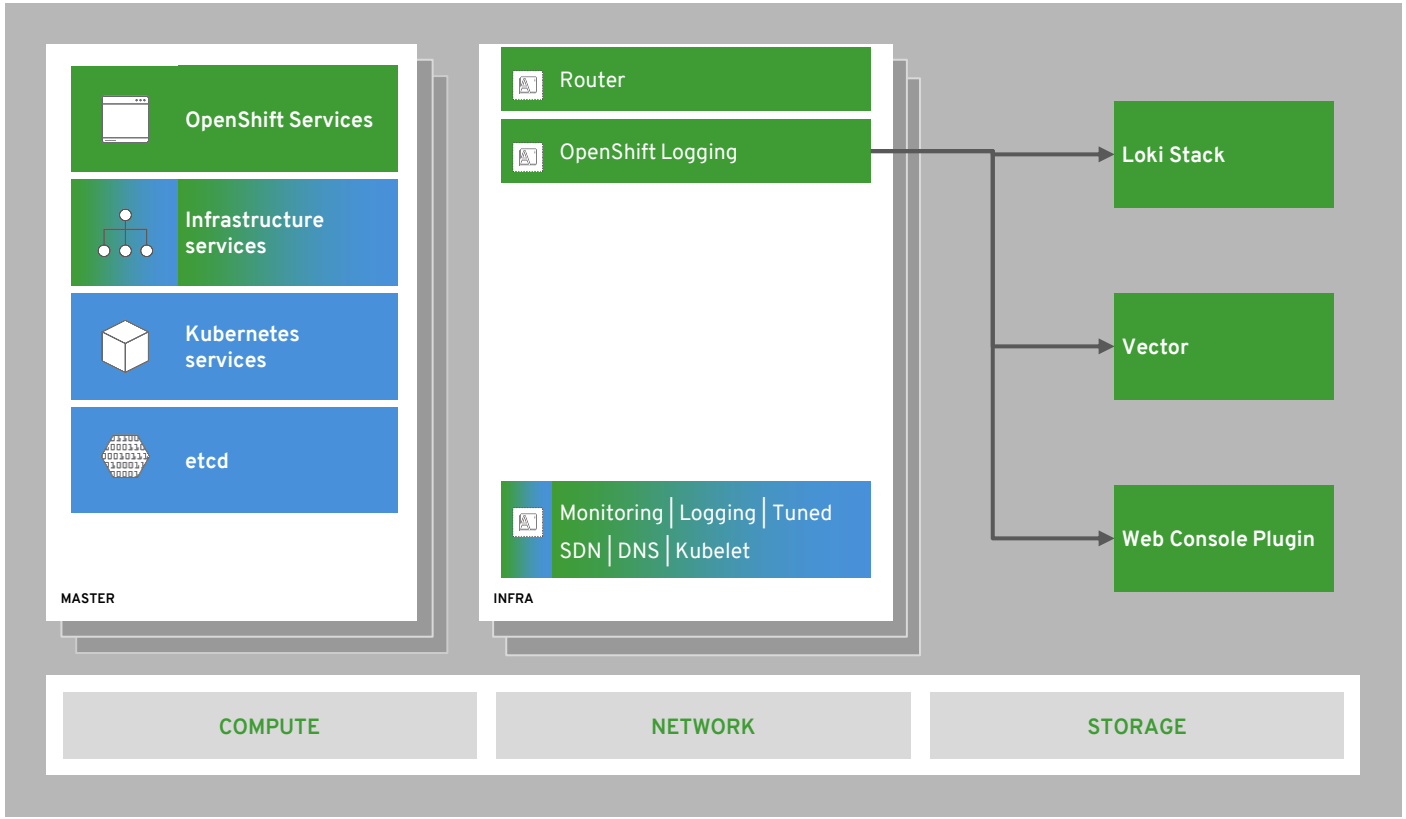
Red Hat

log aggregation (EFK)

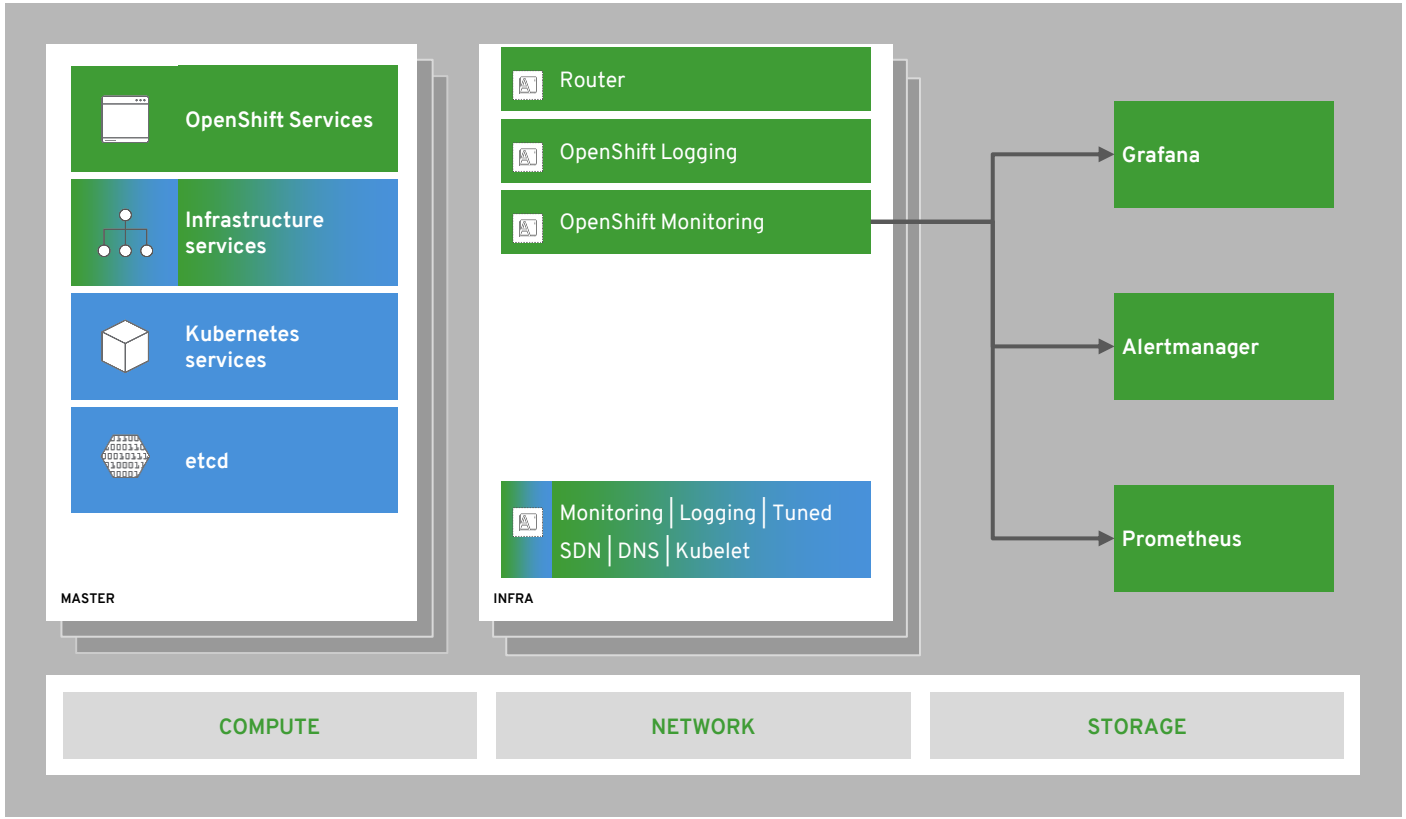


Red Hat

log aggregation (Loki)

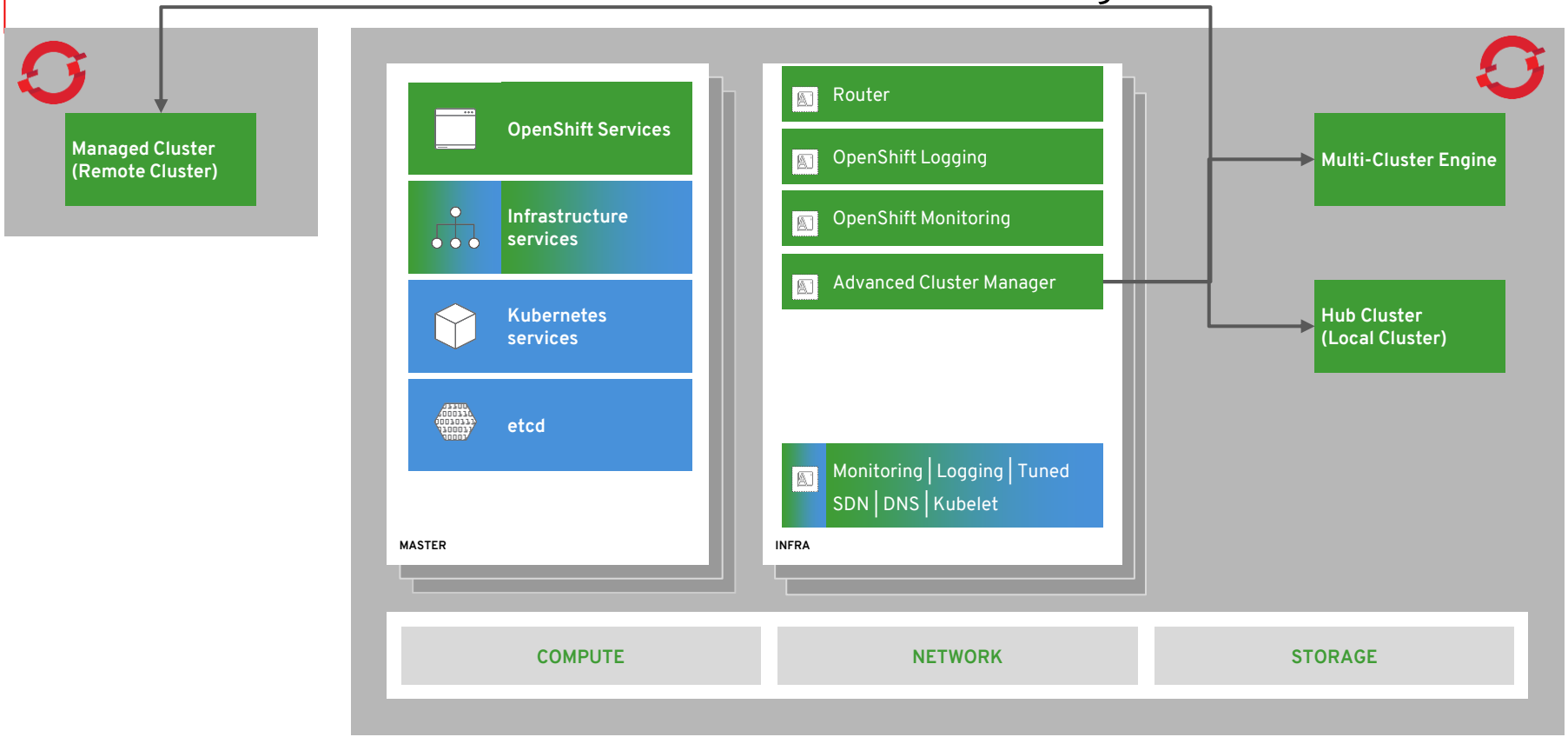


cluster monitoring

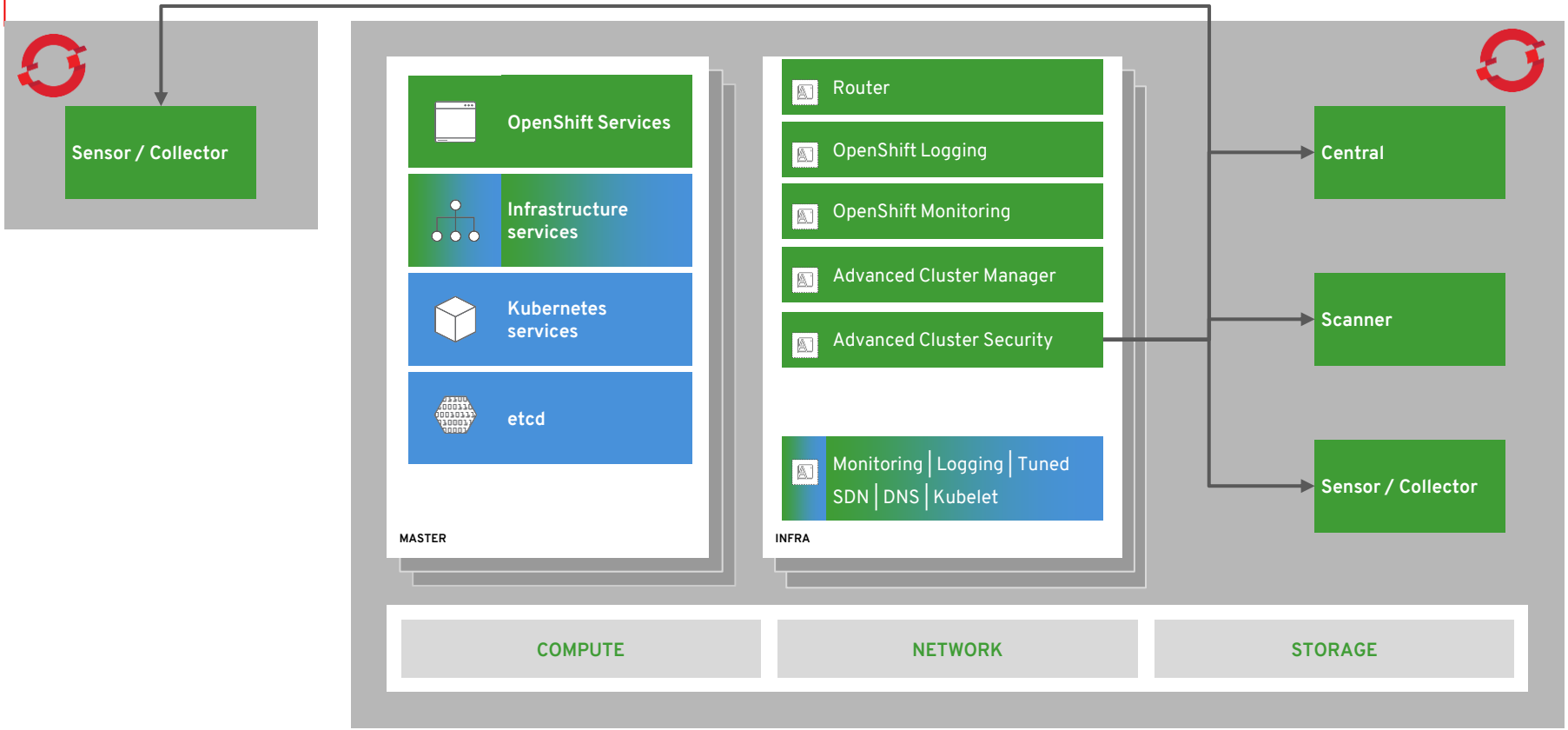


Red Hat

Multi-Cluster Management

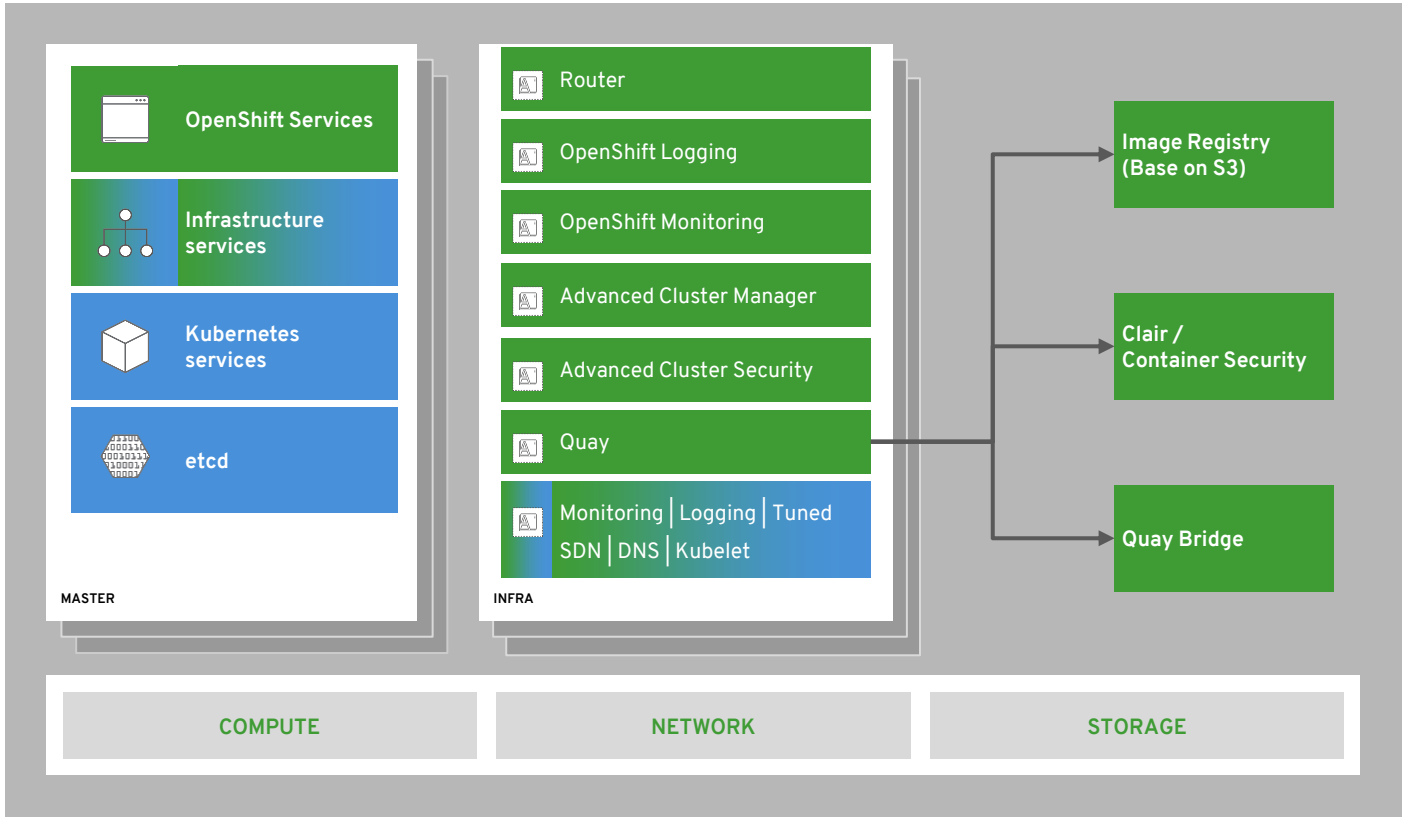


Cluster Security



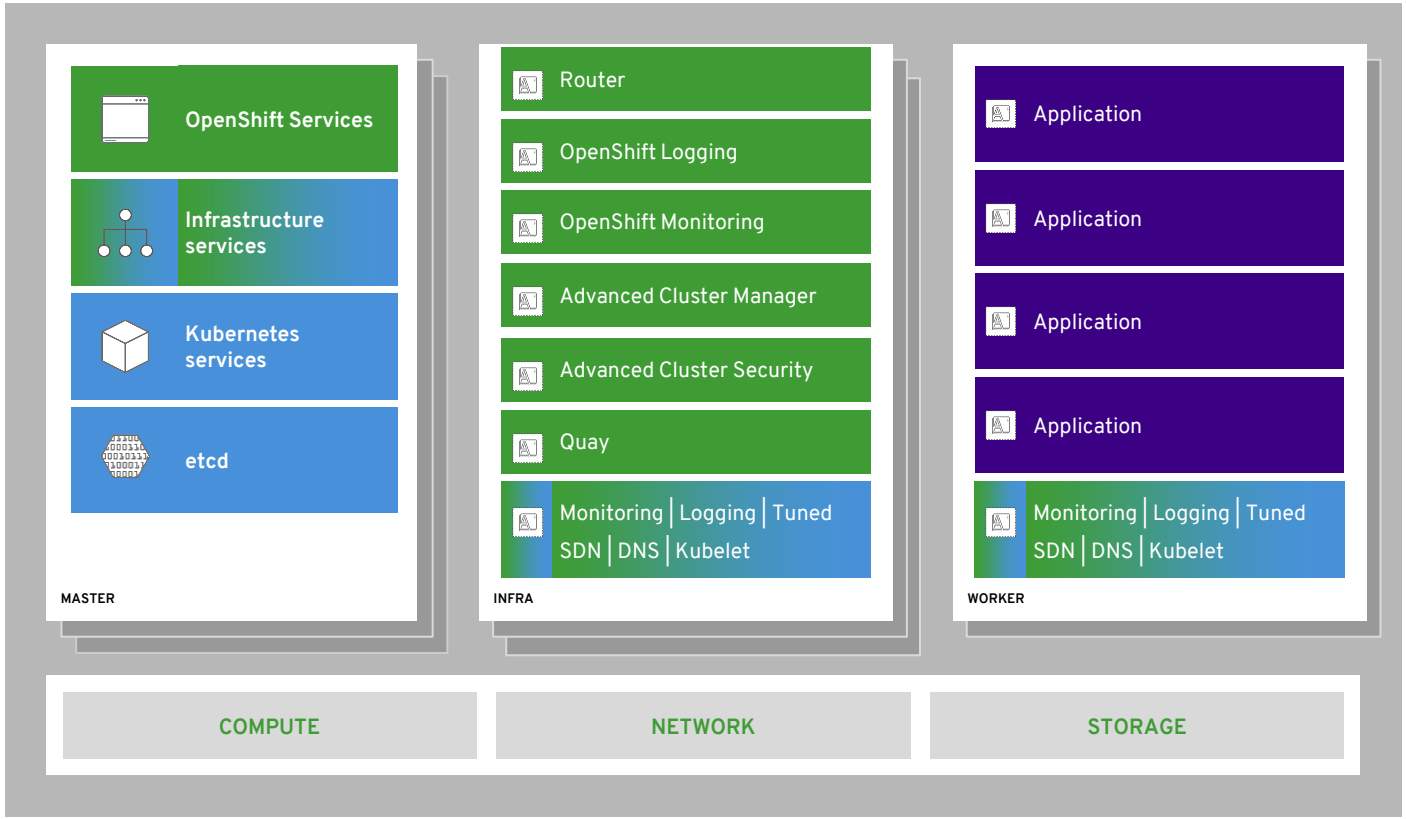
Red Hat

Image Registry



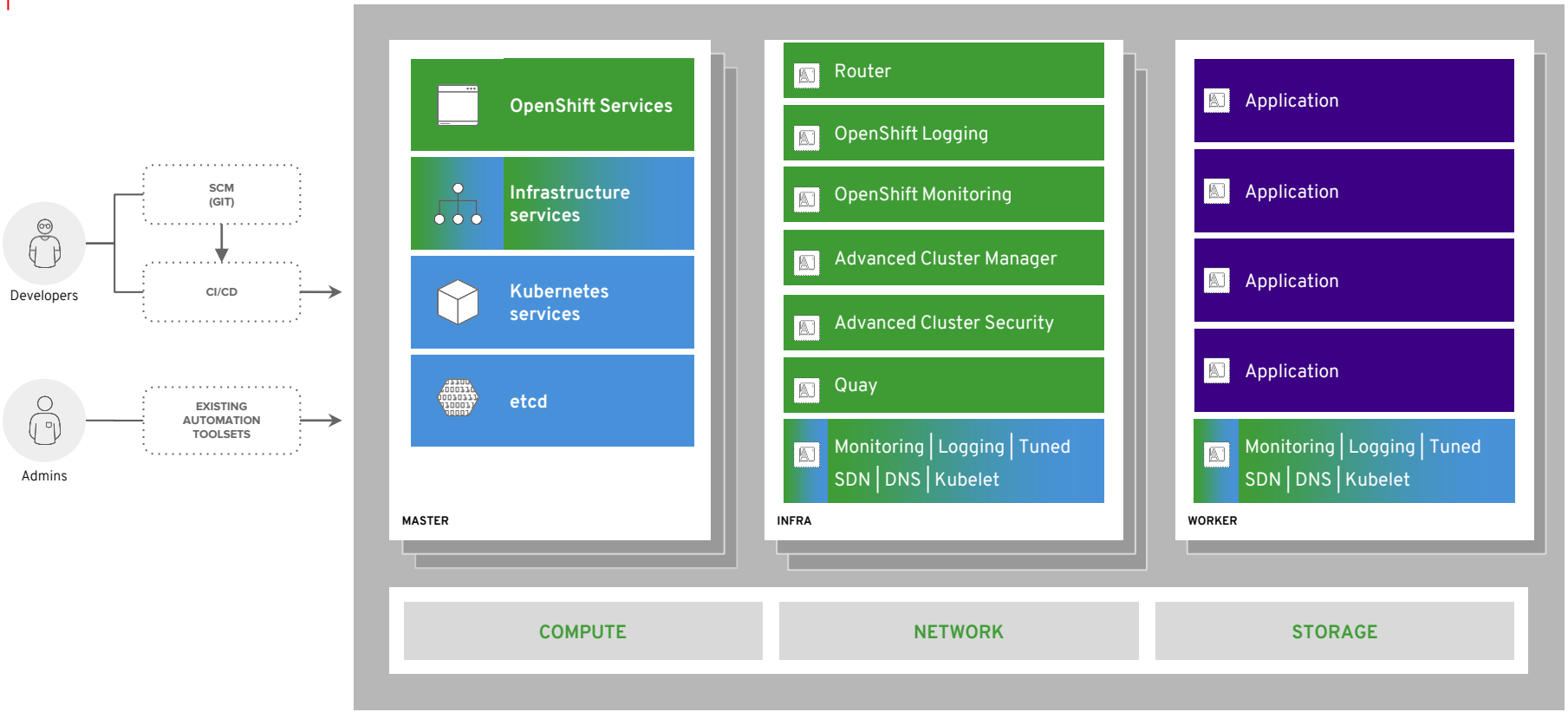
Red Hat

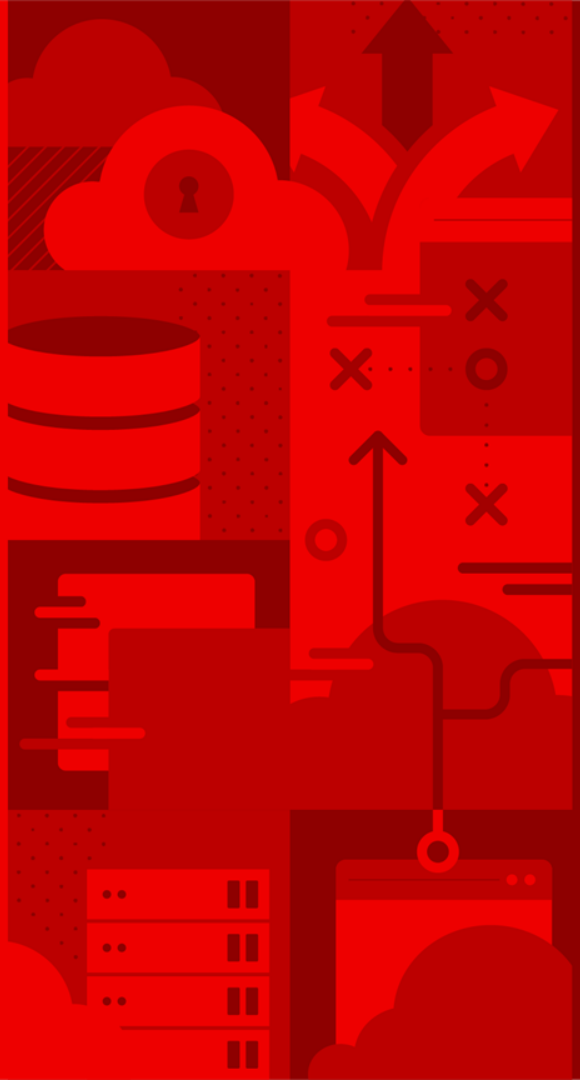
workers run workloads



Red Hat

dev and ops via web, cli, API, and IDE





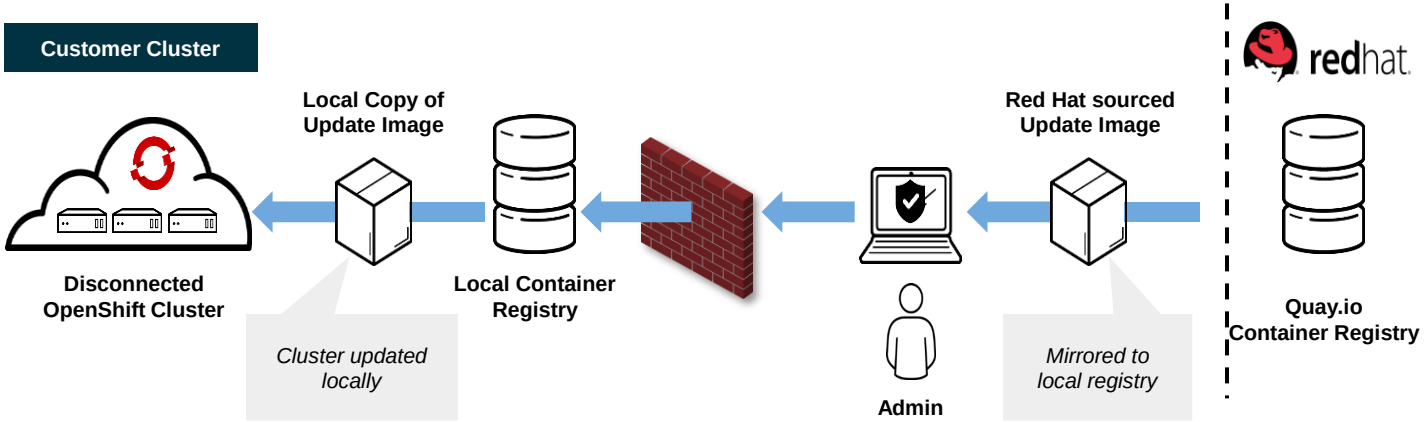
OpenShift lifecycle, installation & upgrades

Red Hat

OpenShift 4 installation

Installer and user-provisioned infrastructure, bootstrap, and more

離線環境安裝/升級



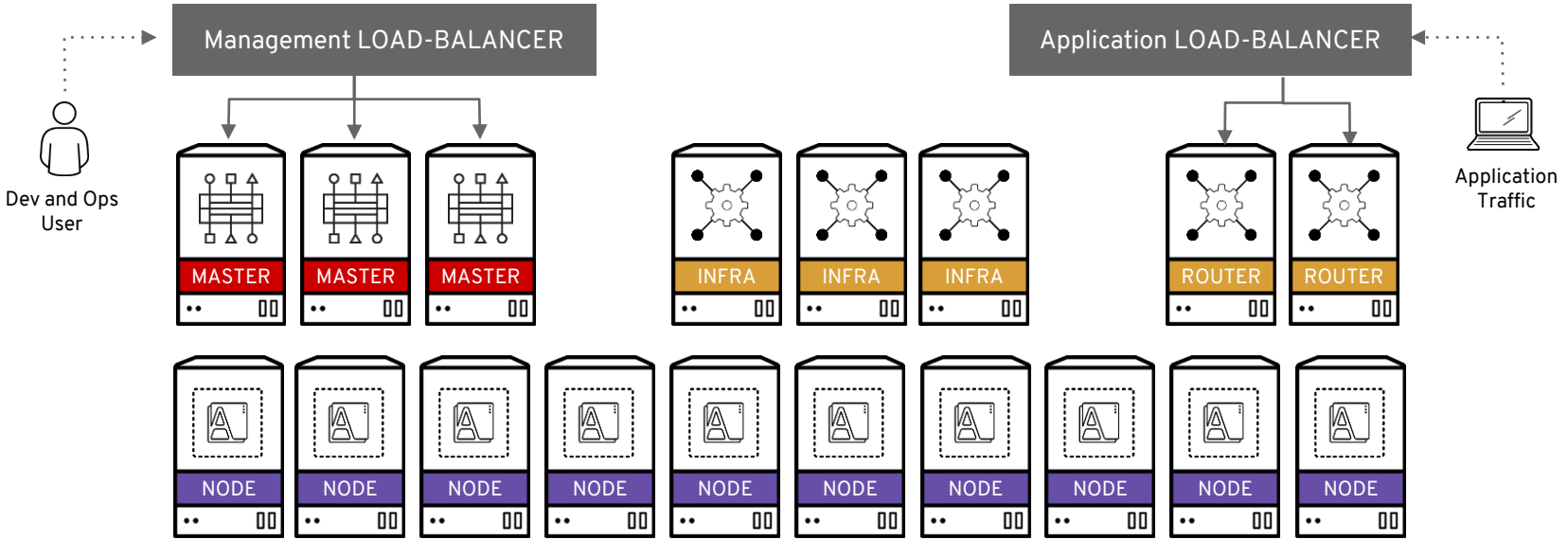
Red Hat

OpenShift Base Full HA Architecture

The base of platform full HA deployment.

Red Hat

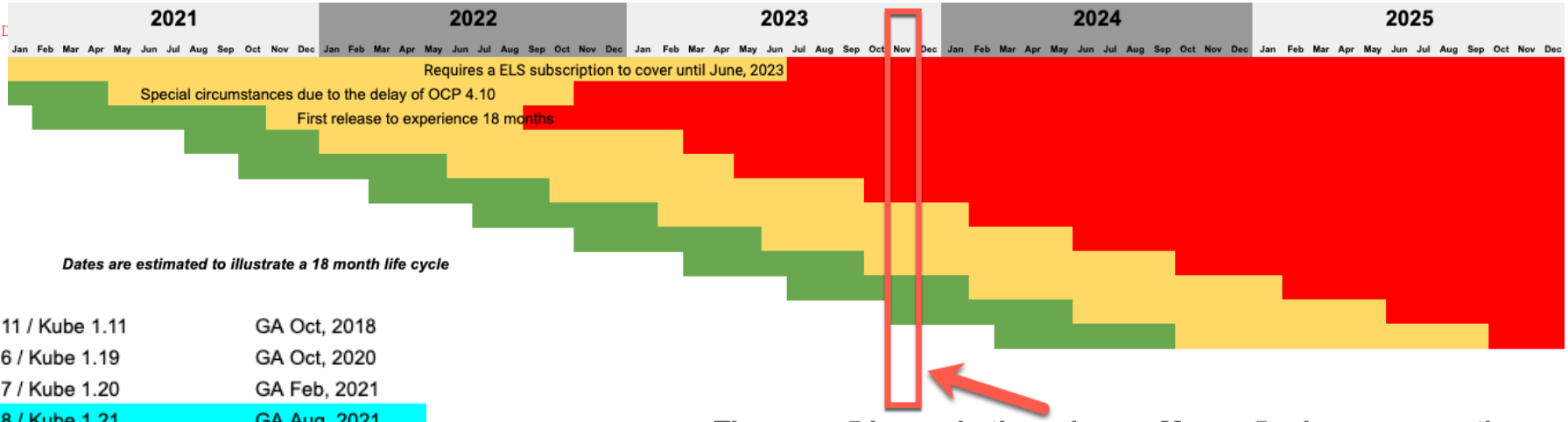
FULL HIGH-AVAILABILITY ARCHITECTURE



OpenShift 4 Lifecycle

Supported paths for upgrades and migrations

Red Hat



Dates are estimated to illustrate a 18 month life cycle

OCP 3.11 / Kube 1.11	GA Oct, 2018
OCP 4.6 / Kube 1.19	GA Oct, 2020
OCP 4.7 / Kube 1.20	GA Feb, 2021
OCP 4.8 / Kube 1.21	GA Aug, 2021
OCP 4.9 / Kube 1.22	GA Oct, 2021
OCP 4.10 / Kube 1.23	GA March, 2022
OCP 4.11 / Kube 1.24	GA July, 2022
OCP 4.12 / Kube 1.25	GA Nov, 2022
OCP 4.13 / Kube 1.26	GA March ,2023
OCP 4.14 / Kube 1.27	GA July, 2023
CCP 4.15 / Kube 1.28	GA Nov, 2023
OCP 4.16 / Kube 1.29	GA March, 2024

Full Support	
Maintenance Support	
Not Supported	
EUS Upgrade Procedure Available	

There are 5 boxes in the column. Means 5 releases are active.

OpenShift 版本

- 一年發布3個版本，與上游相同
- 與上游大約 Delay 一個版本

維護週期

- 開發週期，下一個版本 + 3個月
- 維護週期，最長大約 24 個月 (4.12 EUS only)

https://cloud.redhat.com/blog/time-is-on-your-side-a-change-to-the-openshift-4-lifecycle
https://cloud.redhat.com/blog/openshift-4-lifecycle-enhancements-introducing-an-extended-update-support-phase

Red Hat OpenShift

Life Cycle Dates

Version	General availability	Full support ends	Maintenance support ends	Extended update support ends
Full Support				
4.13	May 17, 2023	4.14 GA + 3 months	November 17, 2024	N/A
Maintenance Support				
4.12	January 17, 2023	August 17, 2023	July 17, 2024	January 17, 2025
4.11	August 10, 2022	April 17, 2023	February 10, 2024	N/A
4.10	March 10, 2022	November 10, 2022	September 10, 2023	N/A
End of life				
4.9	October 18, 2021	June 10, 2022	April 18, 2023	N/A
4.8	July 27, 2021	January 27, 2022	January 27, 2023	N/A

OpenShift 4.13

Channel details

stable-4.13

Latest version 4.13.9 Full support

fast-4.13

Latest version 4.13.11 Full support

-

No 4.13 EUS channel

candidate-4.13

Latest version 4.13.11

OpenShift 4.12

Channel details

stable-4.12

Latest version 4.12.30 Maintenance support

fast-4.12

Latest version 4.12.31 Maintenance support

eus-4.12

Latest version 4.12.30 Maintenance support

candidate-4.12

Latest version 4.12.32

OpenShift 4.11

Channel details

stable-4.11

Latest version 4.11.47 Maintenance support

fast-4.11

Latest version 4.11.48 Maintenance support

-

No 4.11 EUS channel

candidate-4.11

Latest version 4.11.48

https://console.redhat.com/openshift/releases

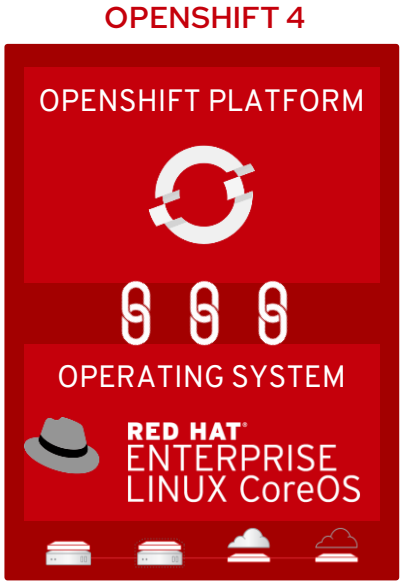
Red Hat Enterprise Linux CoreOS

The OpenShift operating system

Red Hat

Red Hat Enterprise Linux CoreOS

Controlled Immutability / Container Optimized



Role in OpenShift Ecosystem

- Versioned and validated for specific OpenShift version
- 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
  - Multiple machine pools can be created
- Continuously monitoring for configuration drift
- Deploy the File Integrity operator to monitor for changes to files

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

A lightweight, OCI-compliant container runtime

Minimal and Secure Architecture

Optimized for Kubernetes

Runs any OCI-compliant image (including docker)

CRI-O Support in OpenShift

CRI-O tracks and versions identical to Kubernetes, simplifying support permutations

CRI-O 1.24 	Kubernetes 1.24 	OpenShift 4.11 
CRI-O 1.25 	Kubernetes 1.25 	OpenShift 4.12 
CRI-O 1.26 	Kubernetes 1.26 	OpenShift 4.13 

podman

A docker-compatible CLI for containers

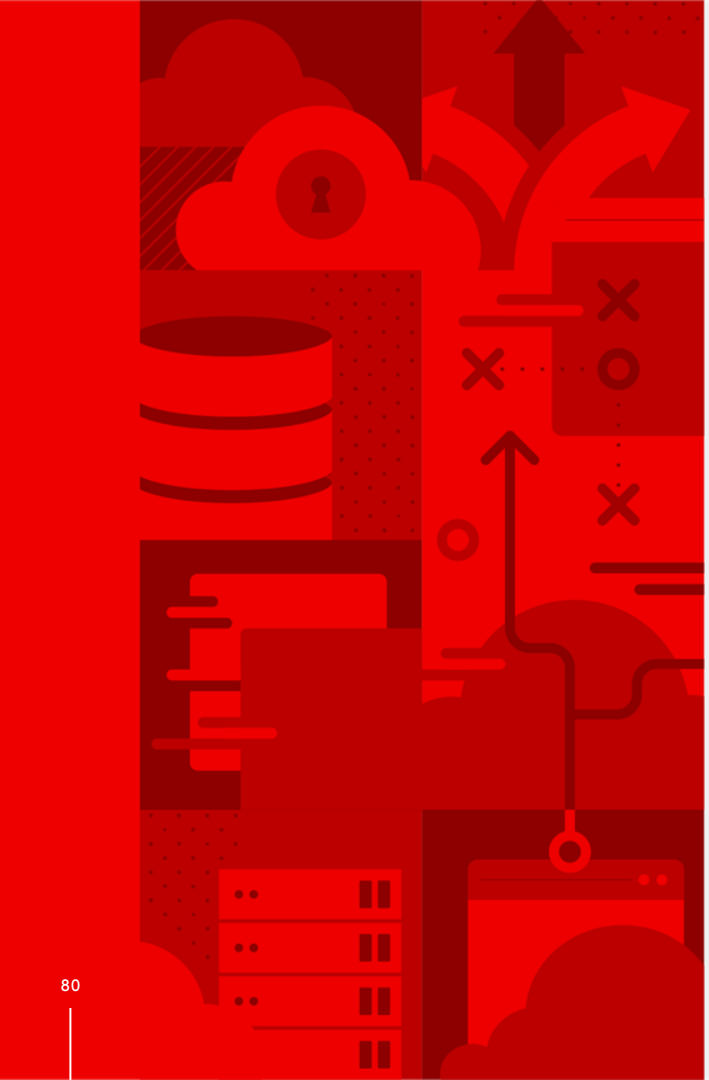
- Remote management API via Varlink
- Image/container tagging
- Advanced namespace isolation

buildah

buildah

Secure & flexible OCI container builds

- Integrated into OCP build pods
- Performance improvements for knative enablement
- Image signing improvements



Take a Break~

Q&A

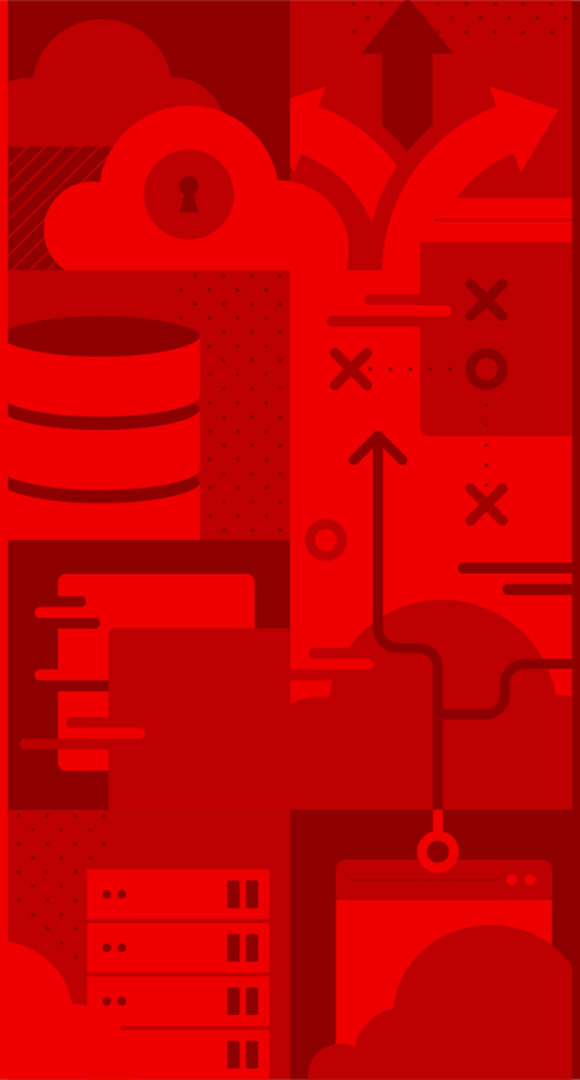
linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat



Operations and infrastructure deep dive

Red Hat

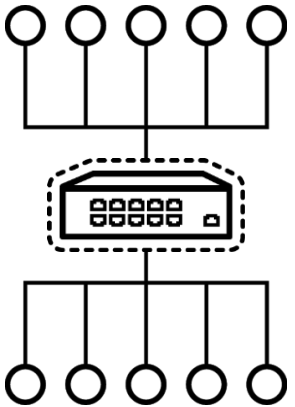
OpenShift Networking

Software-Defined Networking based on openVSwitch technology to provide pods communication and isolation.

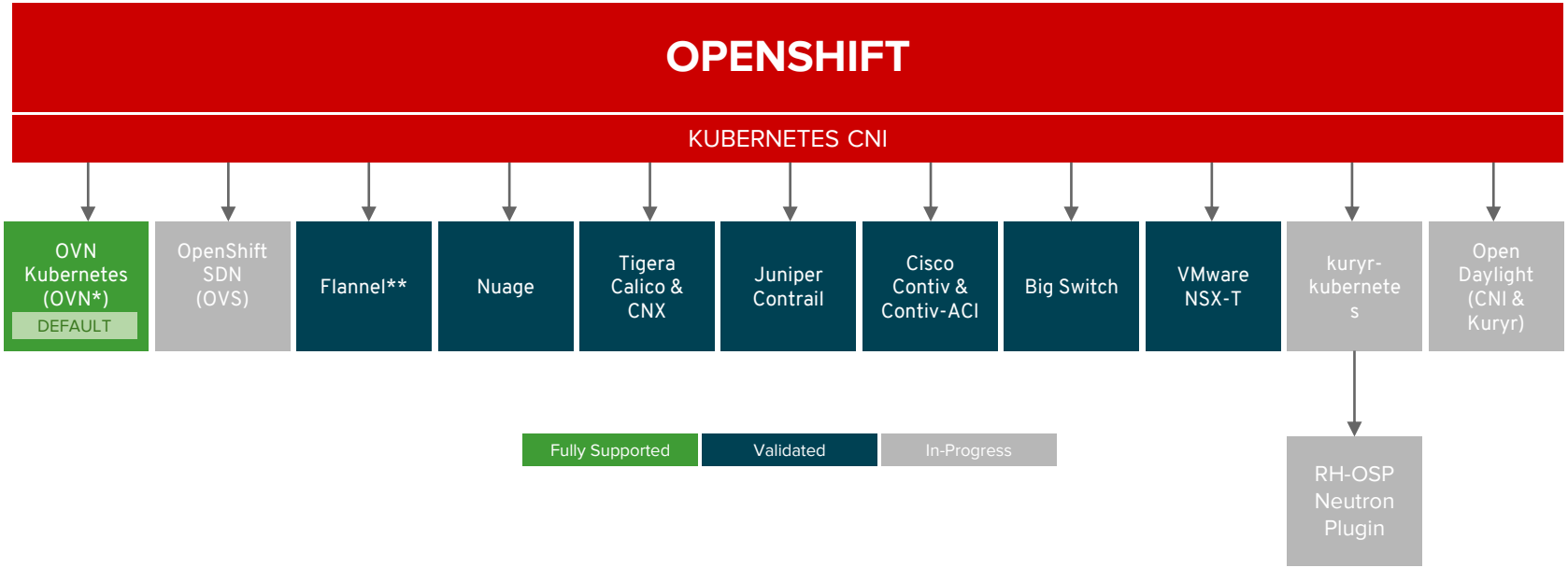
Red Hat

OPENSHIFT NETWORKING

- Built-in **internal DNS** to reach services by name
- Split DNS is supported via SkyDNS
  - Master answers DNS queries for internal services
  - Other name servers serve the rest of the queries
- **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



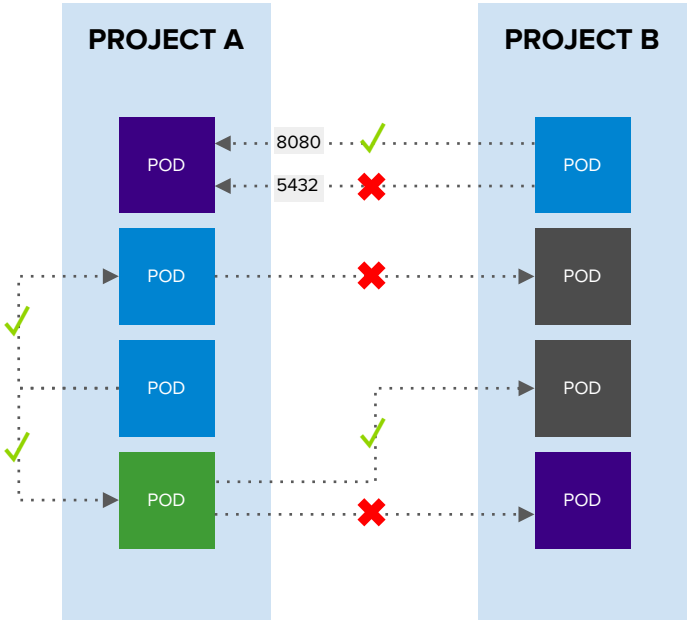
OPENSHIFT NETWORK PLUGINS



* As default in OCP 4.12
** Flannel is minimally verified and is supported only and exactly as deployed in the OpenShift on OpenStack reference architecture

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

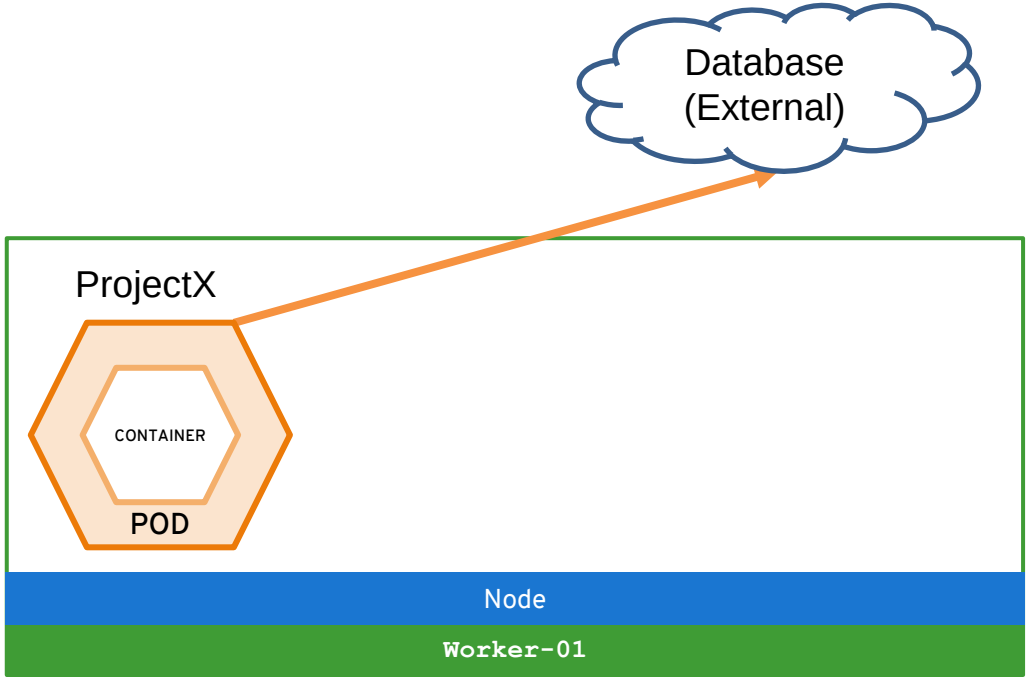
```
apiVersion: extensions/v1beta1
kind: NetworkPolicy
metadata:
  name: allow-to-purple-on-8080
spec:
  podSelector:
    matchLabels:
      color: purple
  ingress:
  - ports:
    - protocol: tcp
      port: 8080
```


Red Hat

Egress IP

來源IP:
10.10.0.11

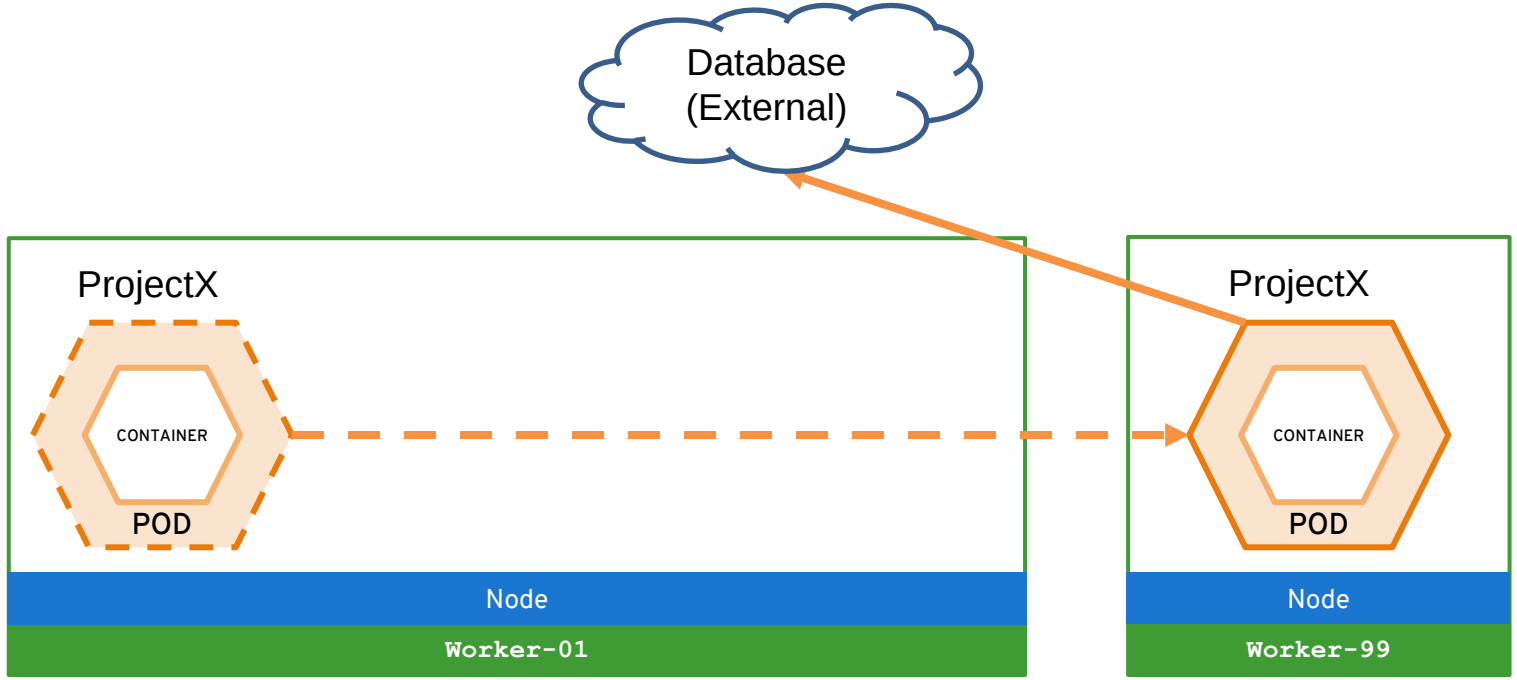
預設行為 (沒用 Egress IP)



Manage IP: 10.10.0.11

來源IP:
10.10.0.99

預設行為 (沒用 Egress IP)

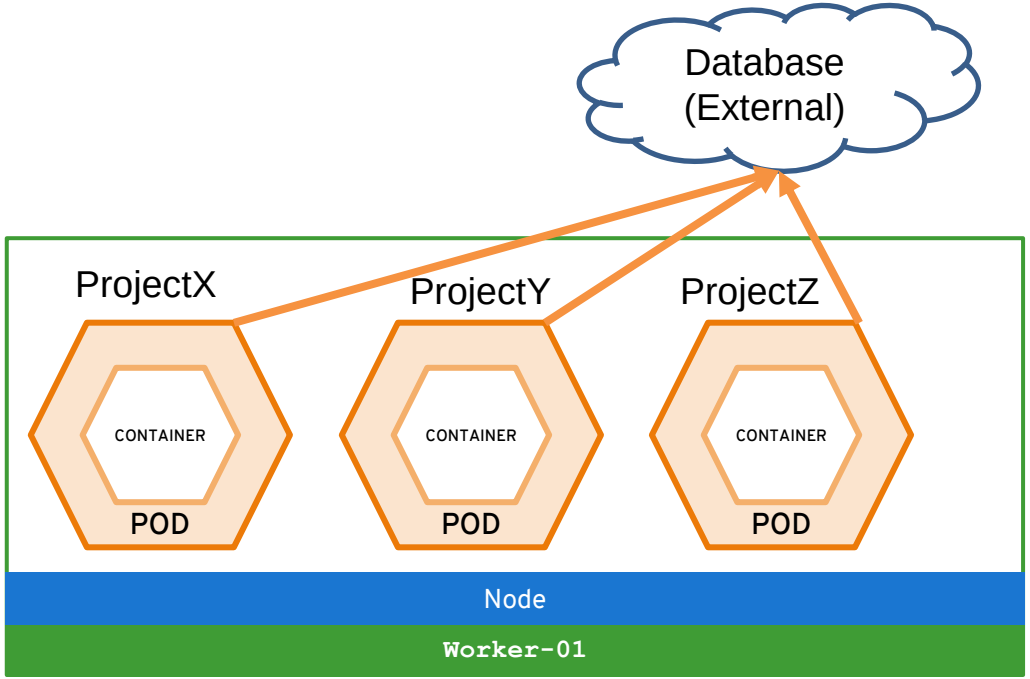


Manage IP: 10.10.0.11

Manage IP: 10.10.0.99

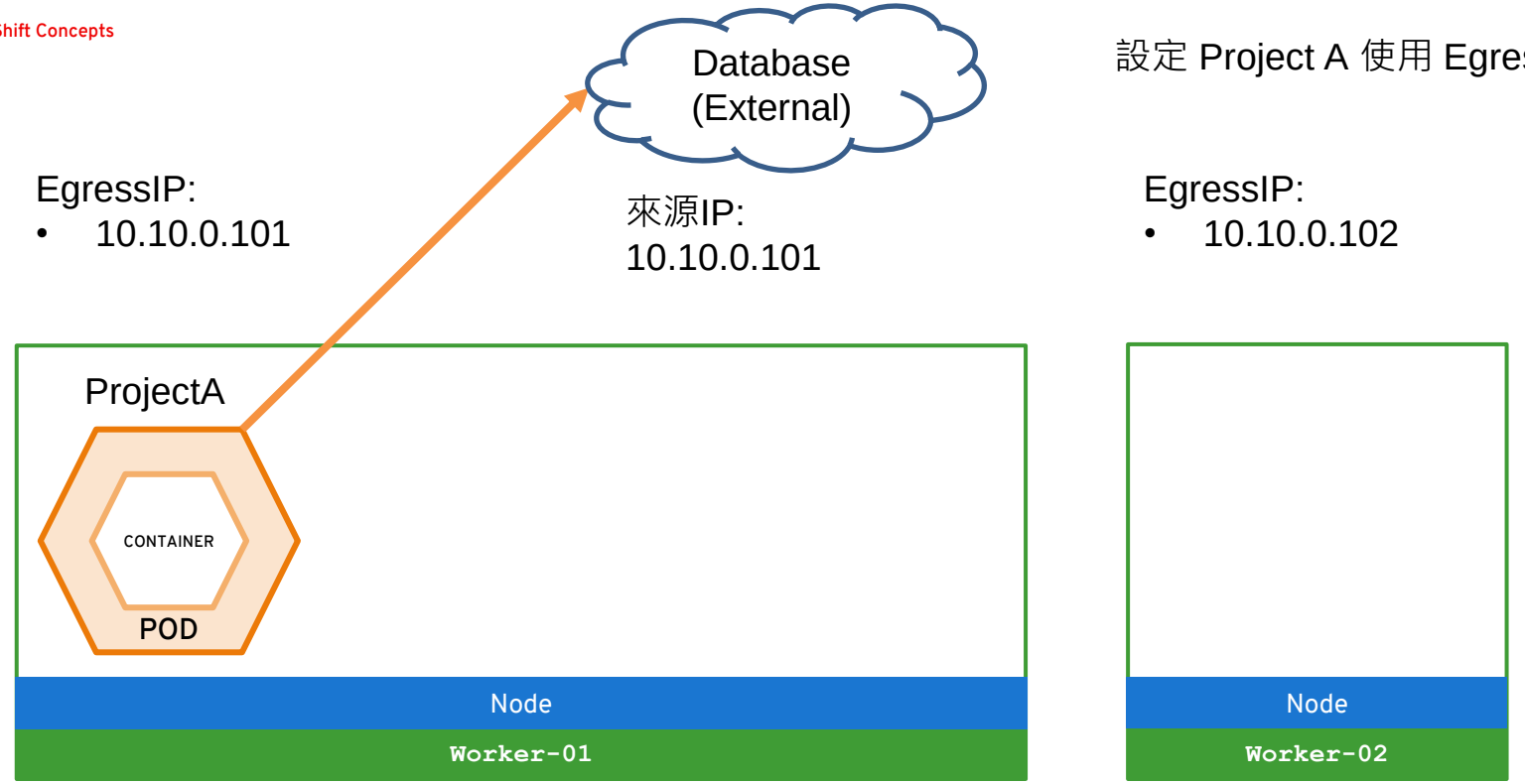
來源IP:
10.10.0.11

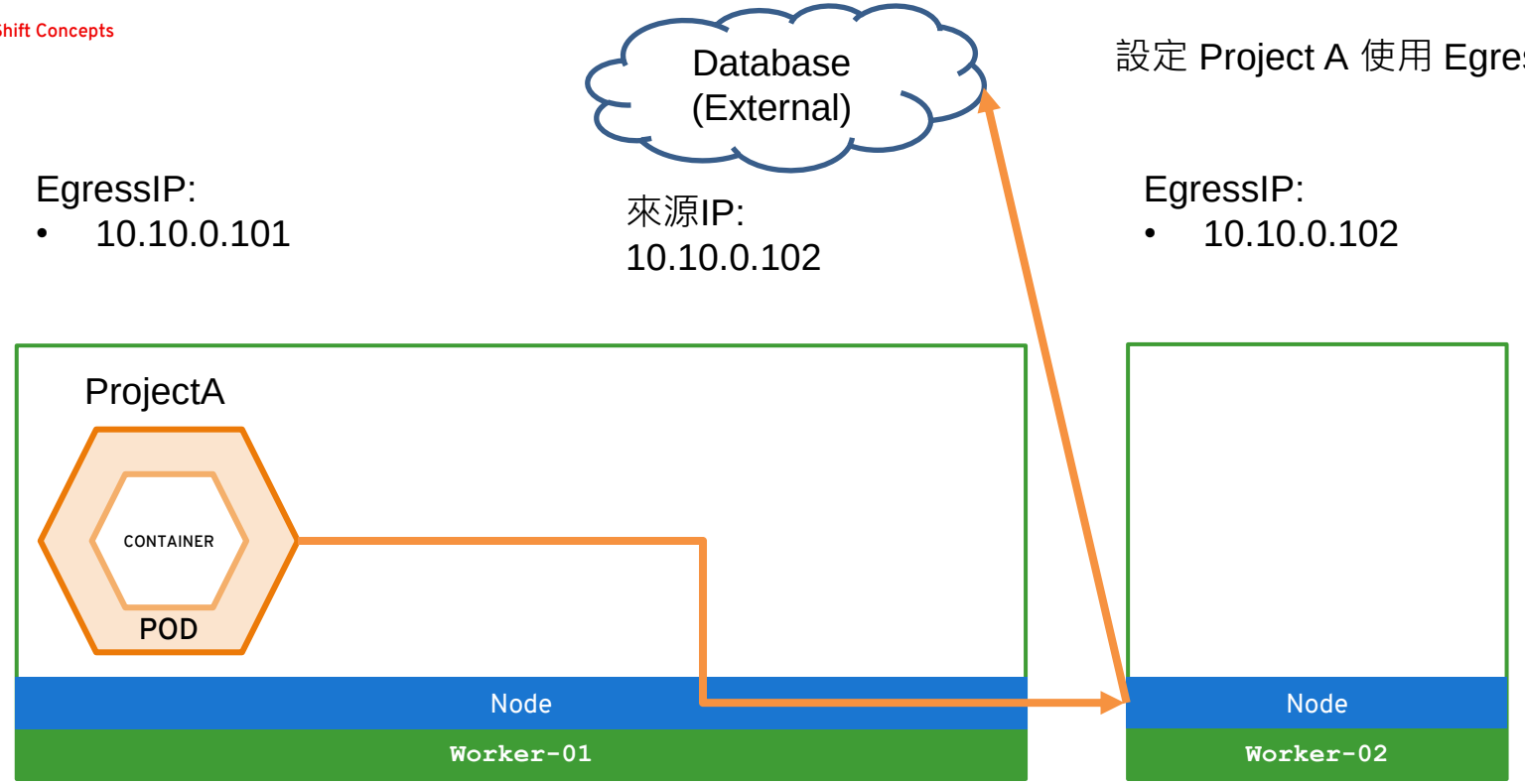
預設行為 (沒用 Egress IP)



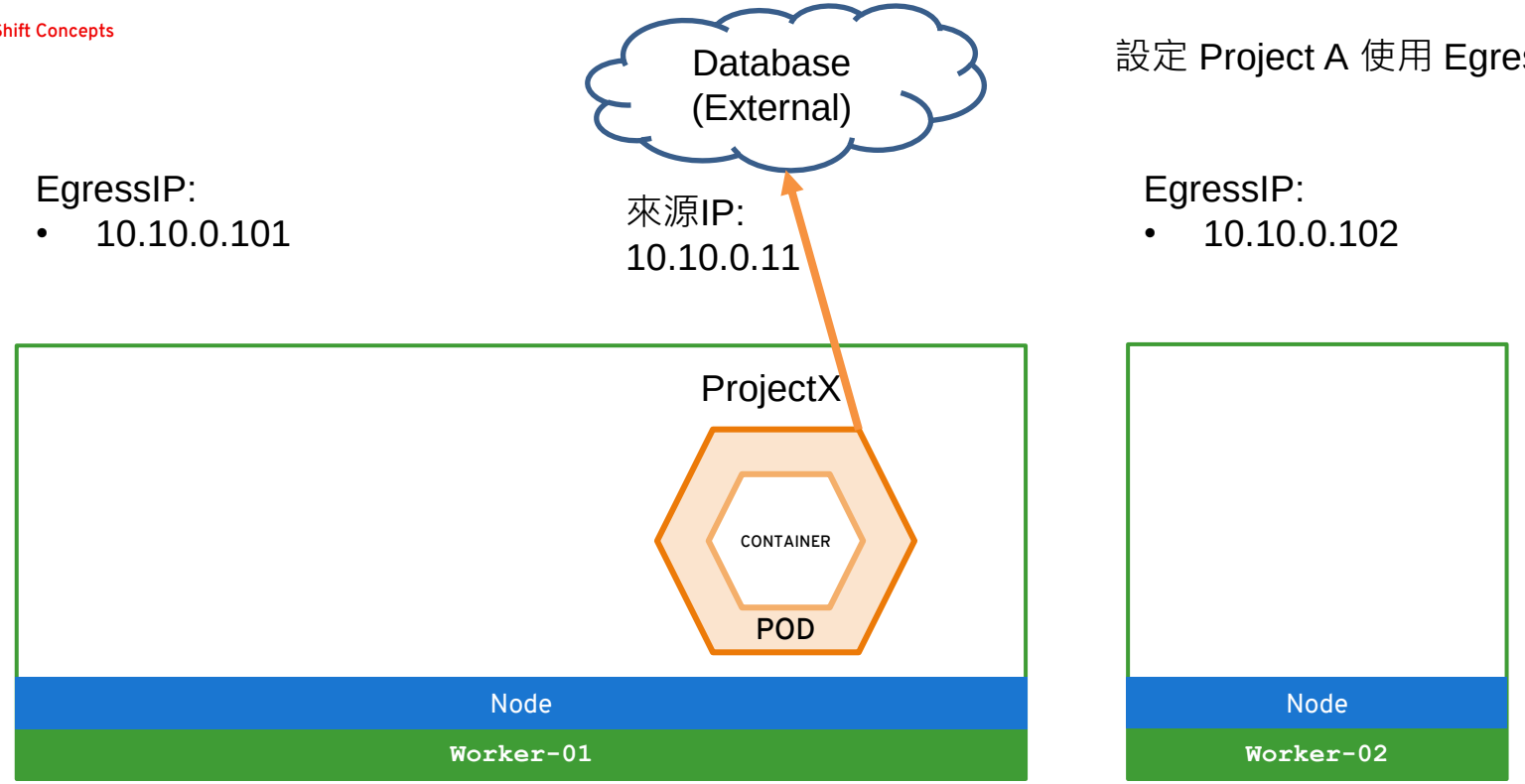
Manage IP: 10.10.0.11

設定 Project A 使用 Egress IP

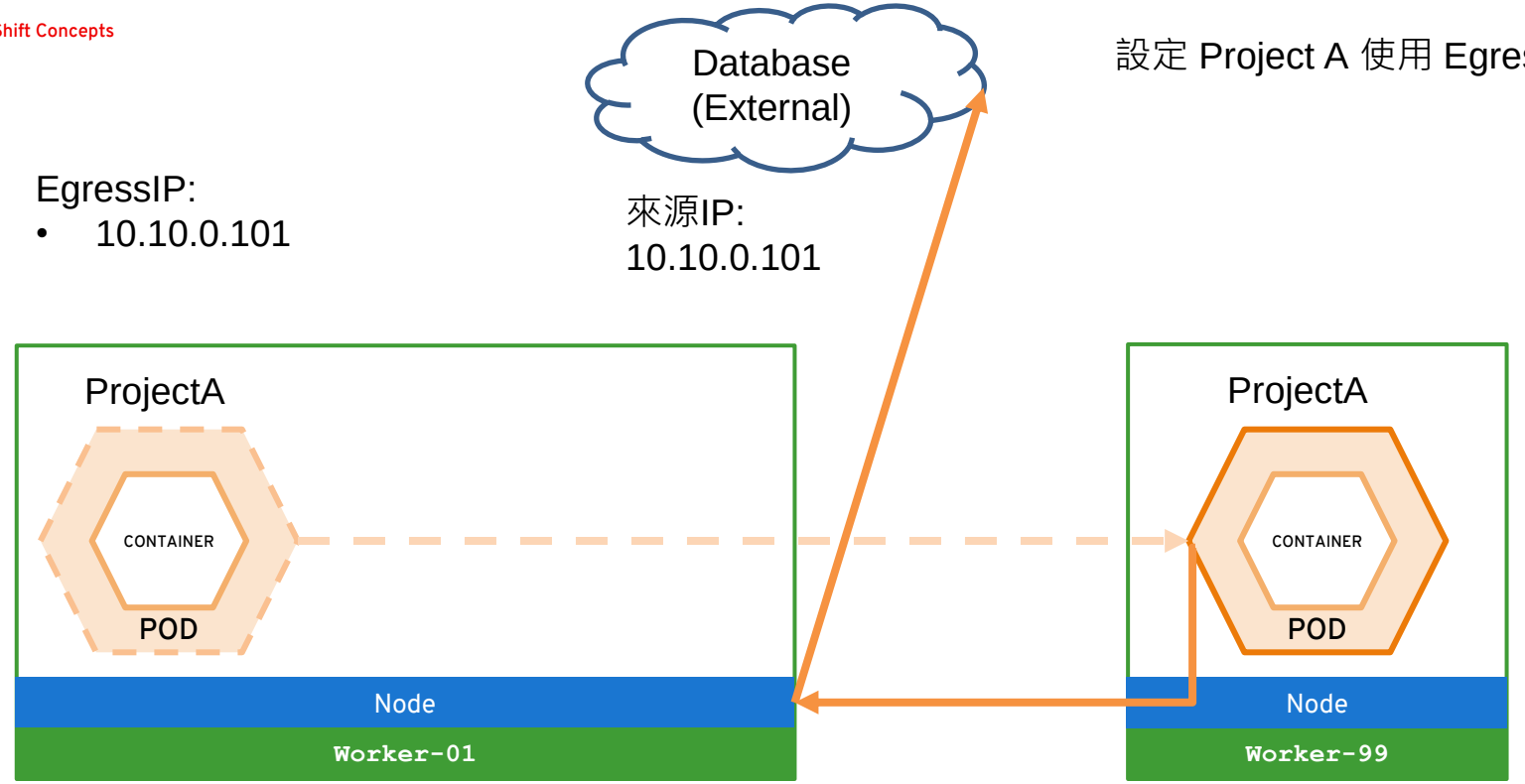


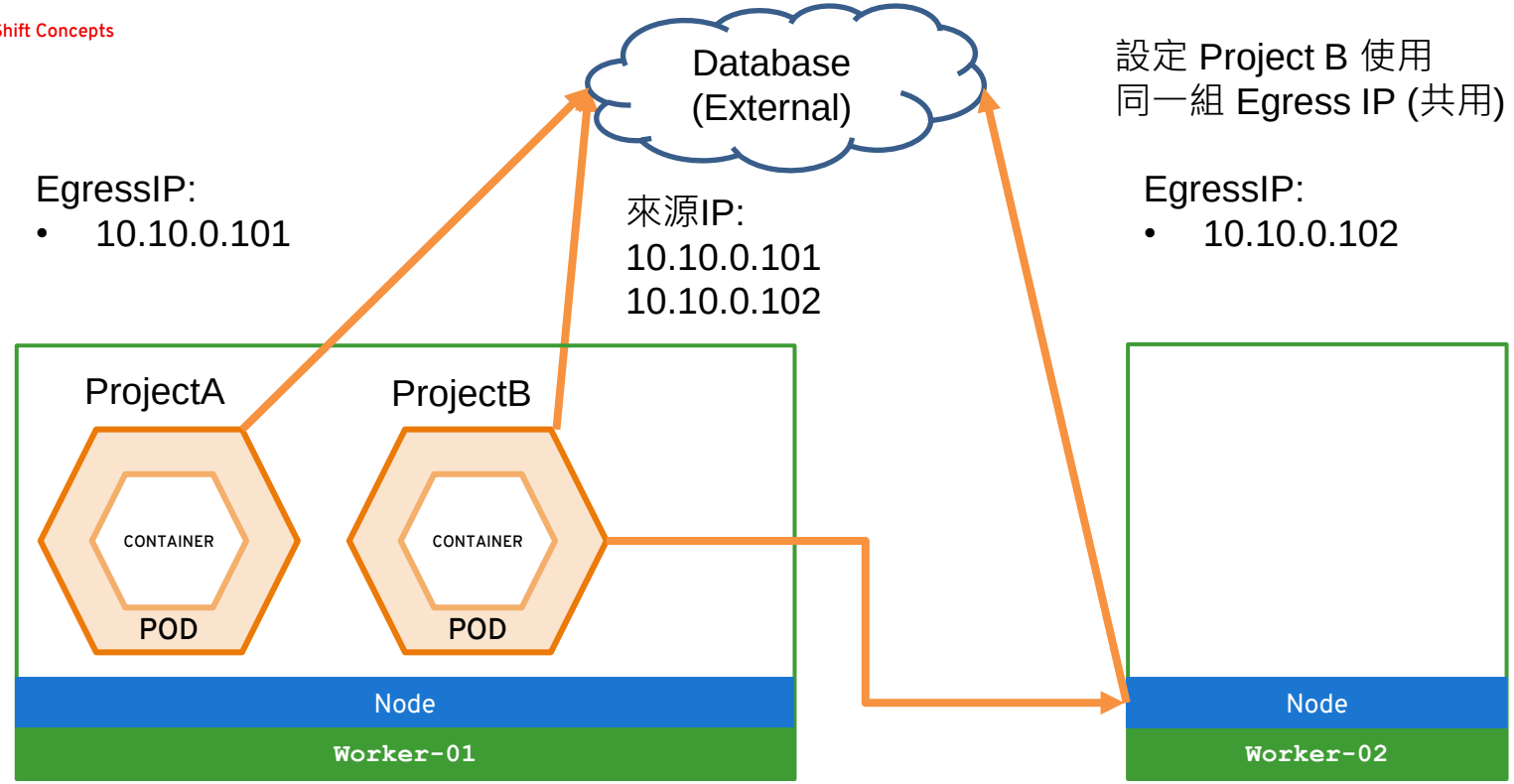

Database
(External)
設定 Project A 使用 Egress IP
EgressIP:
• 10.10.0.101
來源IP:
10.10.0.102
EgressIP:
• 10.10.0.102
ProjectA
CONTAINER
POD
Node
Worker-01
Node
Worker-02
Manage IP: 10.10.0.11
Manage IP: 10.10.0.12

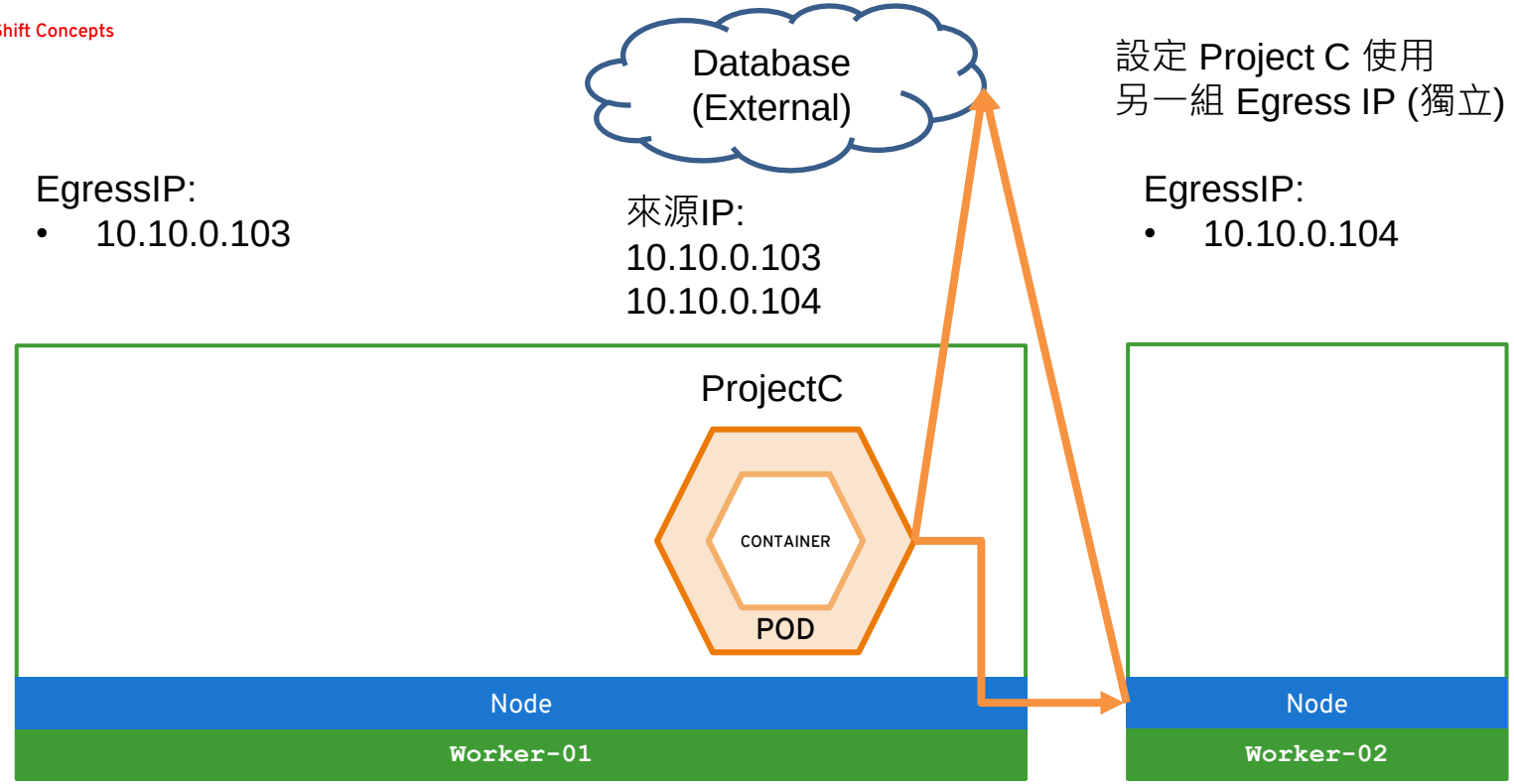
設定 Project A 使用 Egress IP



設定 Project A 使用 Egress IP




Database
(External)
設定 Project B 使用
同一組 Egress IP (共用)
EgressIP:
• 10.10.0.101
來源IP:
10.10.0.101
10.10.0.102
EgressIP:
• 10.10.0.102
ProjectA
ProjectB
CONTAINER
POD
CONTAINER
POD
Node
Worker-01
Node
Worker-02
Manage IP: 10.10.0.11
Manage IP: 10.10.0.12


Database
(External)
設定 Project C 使用
另一組 Egress IP (獨立)
EgressIP:
• 10.10.0.103
來源IP:
10.10.0.103
10.10.0.104
EgressIP:
• 10.10.0.104
ProjectC
CONTAINER
POD
Node
Worker-01
Node
Worker-02
Manage IP: 10.10.0.11
Manage IP: 10.10.0.12

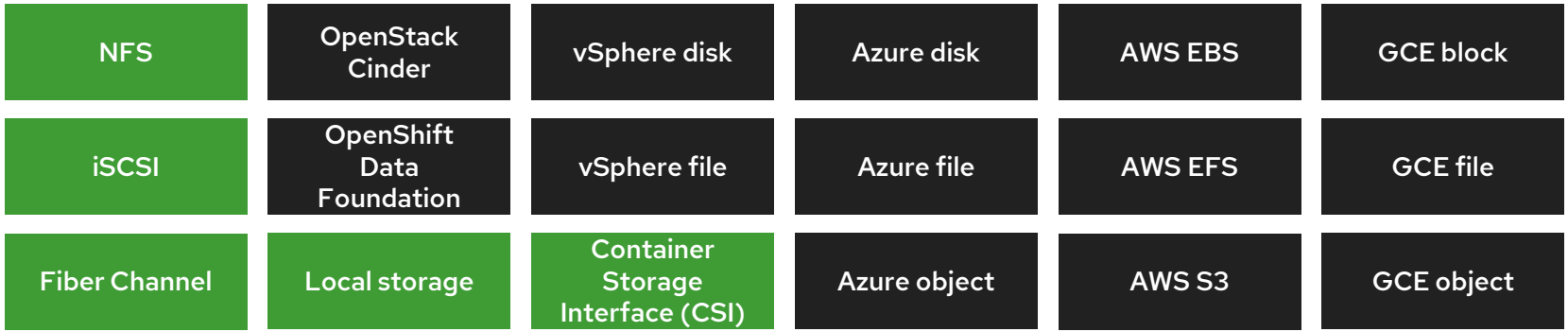
Persistent Storage

Connecting real-world storage to your containers to enable stateful applications

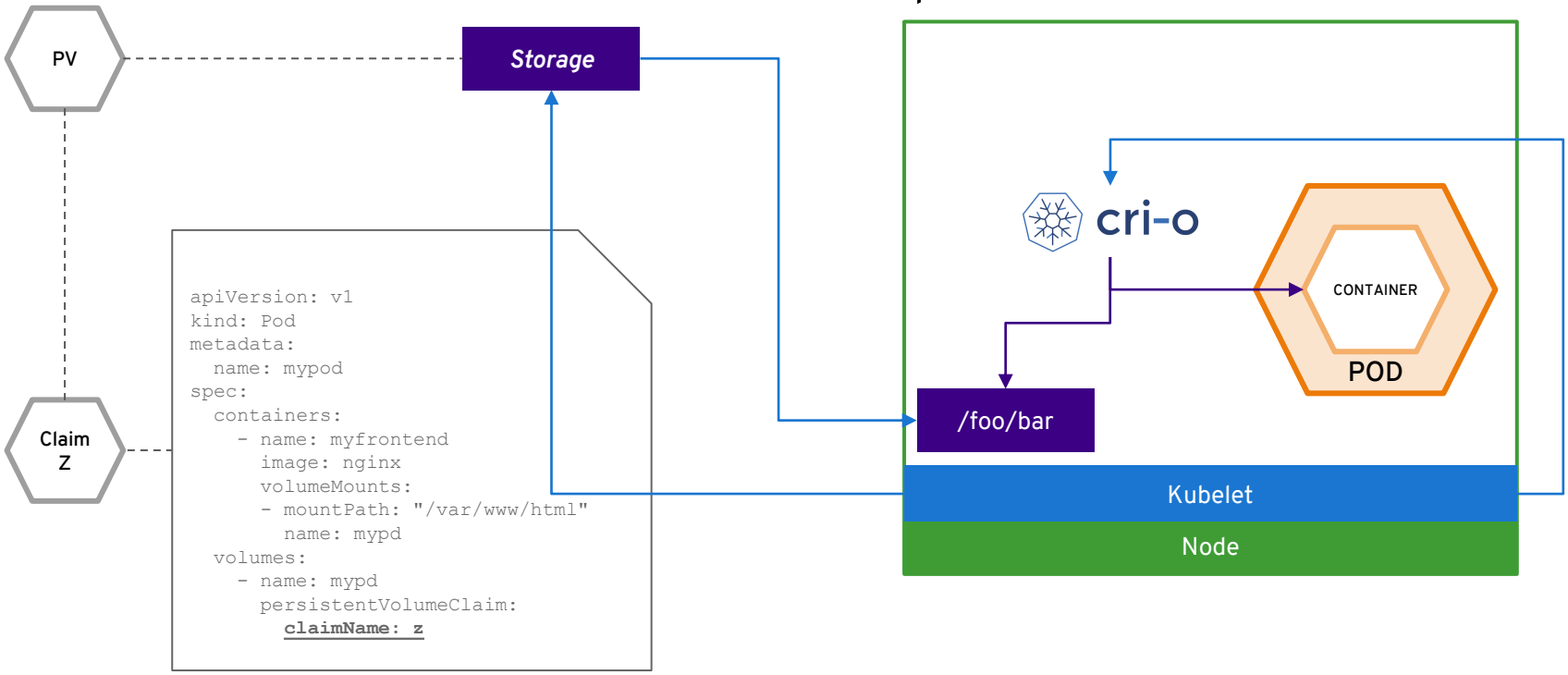
Red Hat

PERSISTENT STORAGE

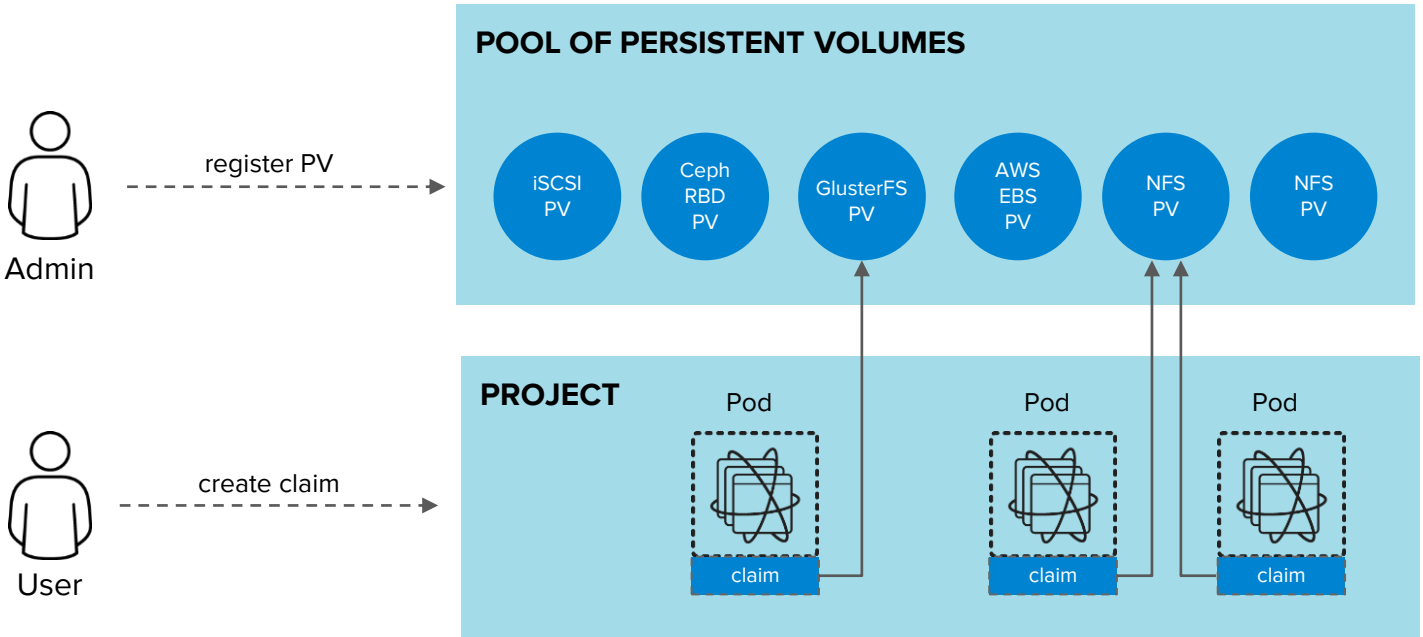
- Persistent Volume (PV) is tied to a piece of network storage
- Provisioned by an administrator (static or dynamically)
- Allows admins to describe storage and users to request storage
- Assigned to pods based on the requested size, access mode, labels and type



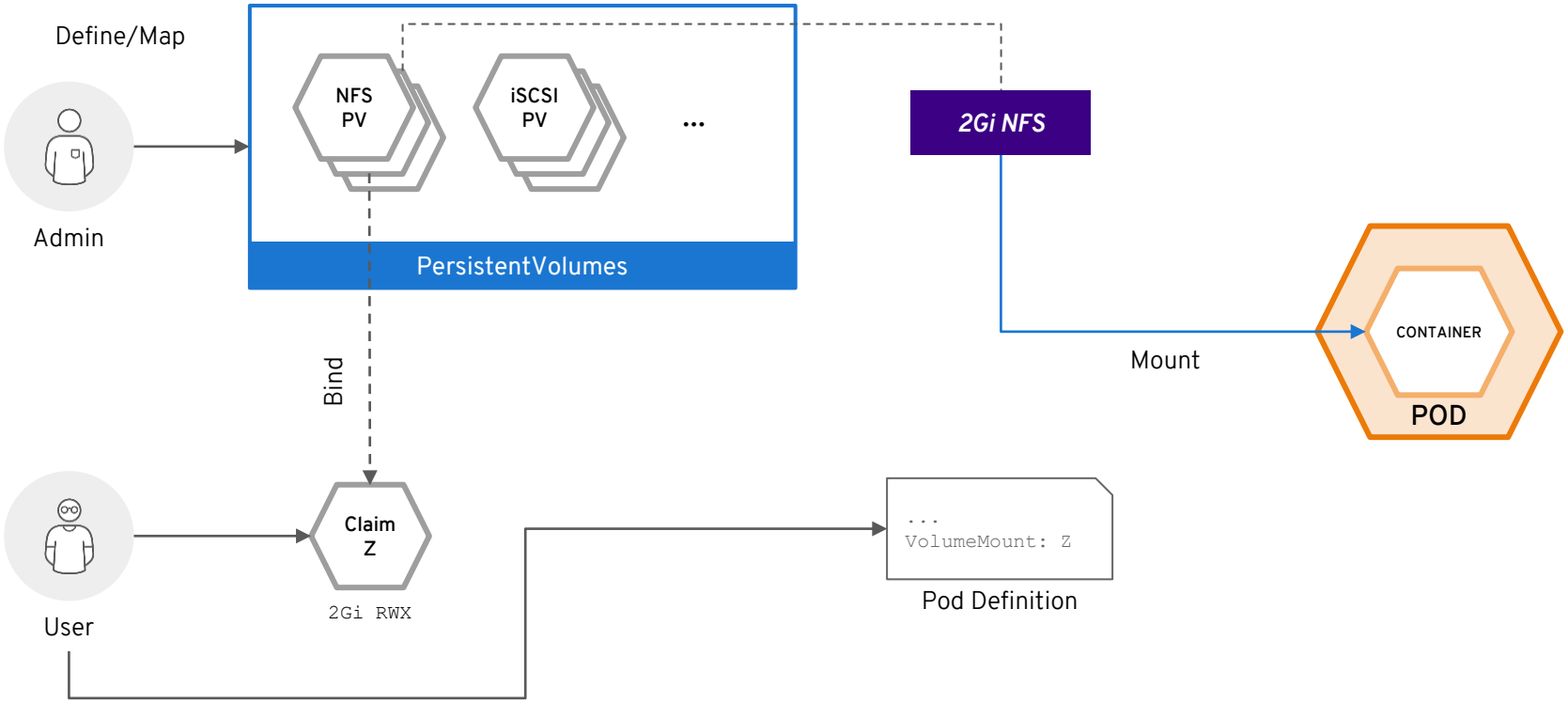
PV Consumption



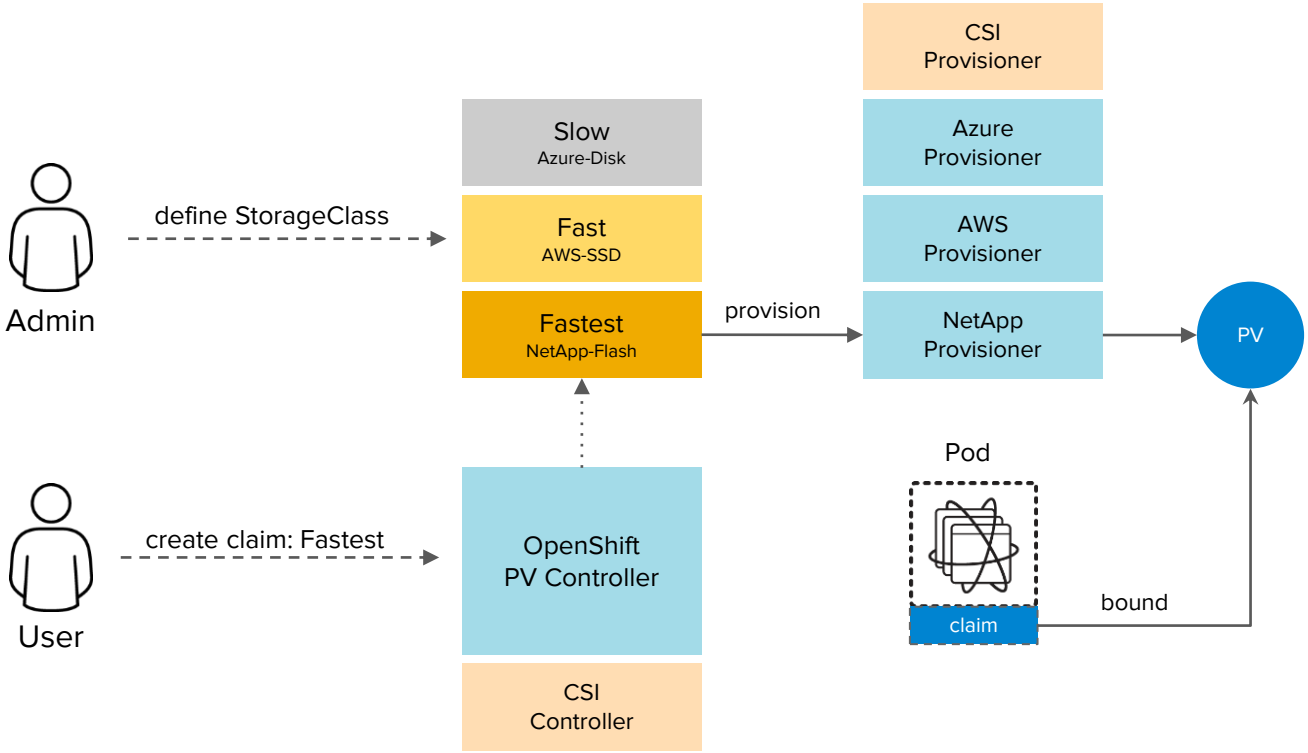
PERSISTENT STORAGE



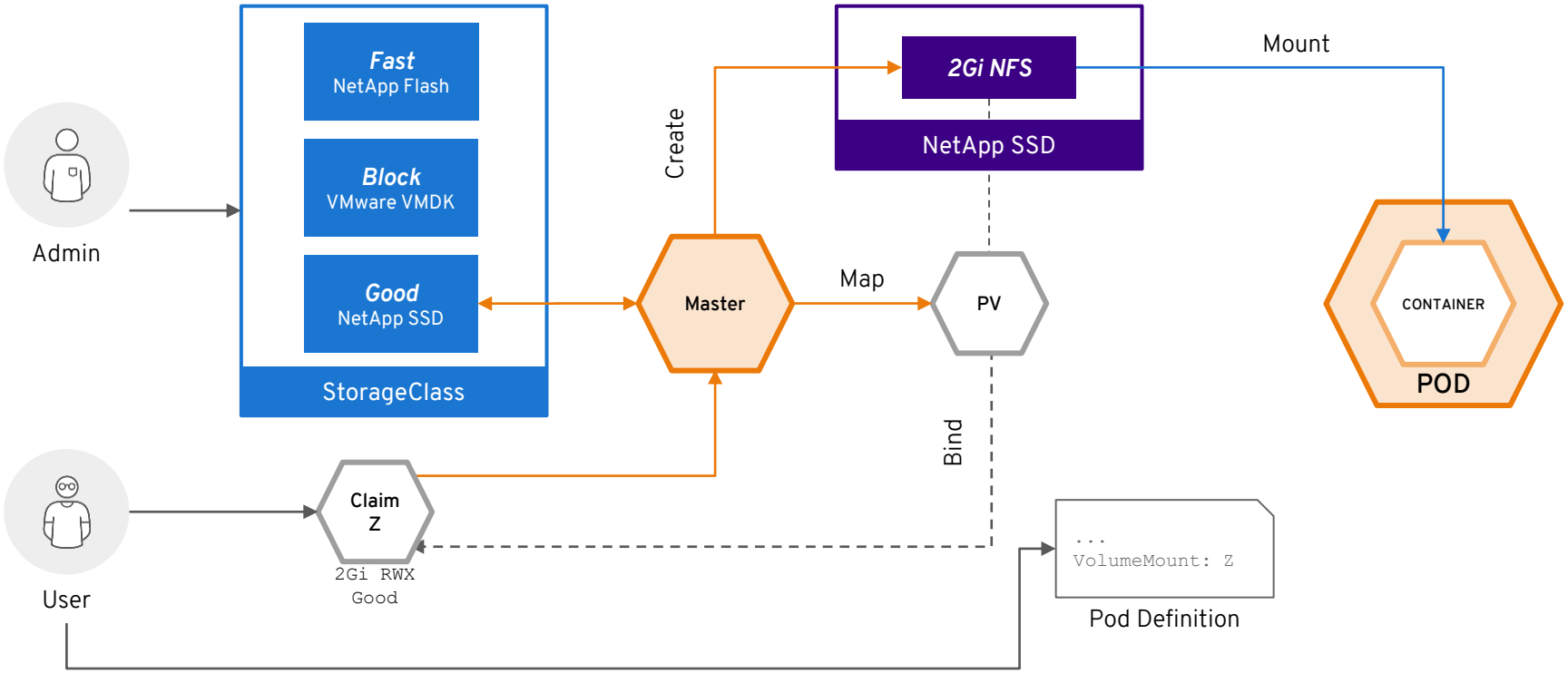
Static Storage Provisioning



DYNAMIC VOLUME PROVISIONING



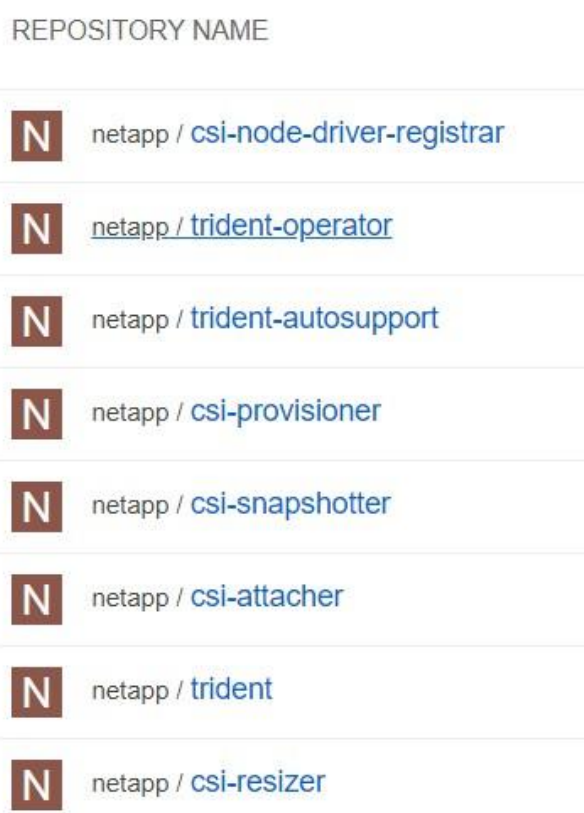
Dynamic Storage Provisioning



NetApp Astra Trident (CSI)

Red Hat

- docker.io/netapp/trident:23.07.0
- docker.io/netapp/trident-autosupport:23.07
- registry.k8s.io/sig-storage/csi-provisioner:v3.5.0
- registry.k8s.io/sig-storage/csi-attacher:v4.3.0
- registry.k8s.io/sig-storage/csi-resizer:v1.8.0
- registry.k8s.io/sig-storage/csi-snapshotter:v6.2.2
- registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.8.0
- docker.io/netapp/trident-operator:23.07.0 (optional)

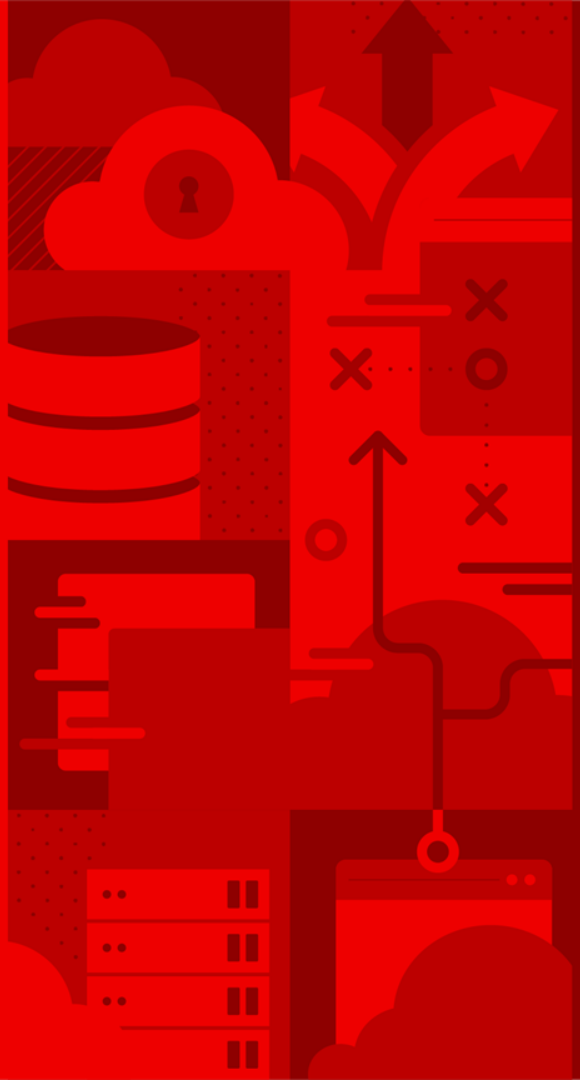


```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: password
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

- Need NetApp Admin / Password

- ontap-san for iSCSI
- ontap-nas for NFS

- S3 直接設定URL，不透過 Trident



Issue Overview

Issue Overview

OpenShift Installation

- Offline Installation (**version 4.12.30**)
- Support Node (VM, 1site)
  - Bastion * 1
  - HA Proxy * 2
- OpenShift Node (VM, 1site)
  - Bootstrap * 0
  - Master * 3
  - Infra (Router) * 2
  - Infra (Logging) * 3
  - Infra (Quay / ACM / ACS) * 2
- OpenShift Node (Bare Metal, Default Max Pods = 250, **Tested Max Pods = 500**)
  - SIT+Dev Worker * 3
  - UAT Worker * 2

Issue Overview

OpenShift Installation

- OperatorHub / Marketplace
  - **OpenShift Logging (EFK) 5.7**
  - **ElasticSearch Operator**
  - Loki Operator
  - **Red Hat Advanced Cluster Manager 2.8**
  - **Red Hat Advanced Cluster Security for Kubernetes 4.1**
  - **Red Hat Quay 3.9**
  - Red Hat Quay Container Security Operator
  - Red Hat Quay Bridge Operator
  - **KEDA / OpenShift Distributed Tracing (Jaeger / OTLP)**
  - ~~Local Volume~~
  - ~~Red Hat OpenShift Data Foundation~~
  - ~~Red Hat OpenShift Service Mesh~~
- 3-rd Operator
  - **NetApp Astra Trident 23.07**

Issue Overview

Networking

- DNS Record
  - 正解 (A Record, User Site)
  - 反解 (PTR Record, Server Site)
    master-01.sit.etmall.ocp ←→ 10.100.10.101

  - Base Domain: **etmall.ocp**
  - Cluster Name(for SIT/UAT): **sit / uat**
    Cluster Name(for Production): **prod / dr**

  - Wildcard: ***.apps.<Cluster Name>.<Base Domain>**
    console-openshift-console.apps.sit.etmall.ocp

- **LoadBalancer (採用 HAProxy 軟體的方式)**
- Ingress (Hostname 指向 LoadBalancer)
- **EgressIP (連到 DB 的 IP 白名單)**
- Firewall (SIT / UAT 之間的防火牆 for ACM / ACS)

Issue Overview

Security

- Security Configuration Implement
  - Firewall
  - TLS
  - SELinux
  - Security Context Constraints
  - **Container Image Scan**
- User / Group / RBAC
- **用 OAuth 保存帳號 (Azure AD)**
  - **https://cloud.redhat.com/blog/openshift-blog-aro-aad**
- 備案：使用 LDAP 保存帳號 (本地AD)

SSL Certificate

- 自簽CA or CA import (X509v3 Subject Alternative Name)
- **Mirror Registry needs TLS**
- Ingress TLS offload
- HAProxy SNI Backend (need HAProxy 1.8+ / RHEL 8+)
- Certificate Rotate

Issue Overview

Node Sizing

- **CSI use NetApp Astra Trident (iSCSI ~~/ NFS~~)**
- Storage Provision
- Monitoring / Logging (Disk Sizing)

- Node Sizing
  - How Many Applications?
  - How Much CPU / Memory Usage per Application?
  - How Many Replicas for High Available?
  - N + 1 ?

- Networking IP Sizing
- Network Segment (VM / Data Usage)
- OpenShift SDN (Pod / Service Usage)

Need to Pay Attention

- CSI use NetApp Astra Trident **(3rd Operator)**

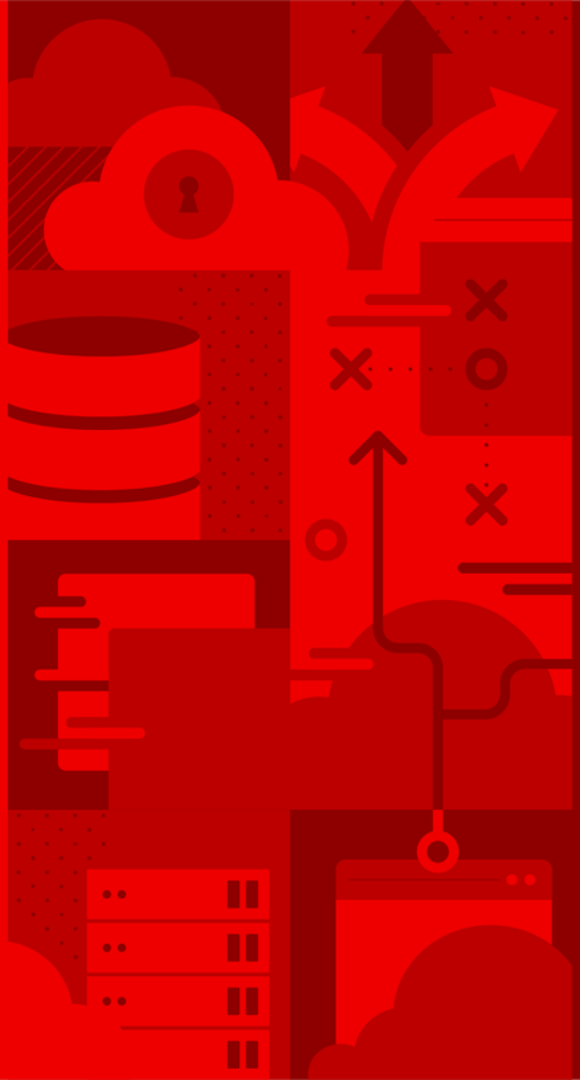
Issue Overview

Development

- Image Registry (Quay)
  - UAT
  - SIT
- Image Scan
  - 網路白名單
  - CVE 資訊下載

Monitoring & Alerting

- **Logging 採用 EFK / Loki ?**
- Mail Alert 寄送名單
- Webhook to MS Teams ?

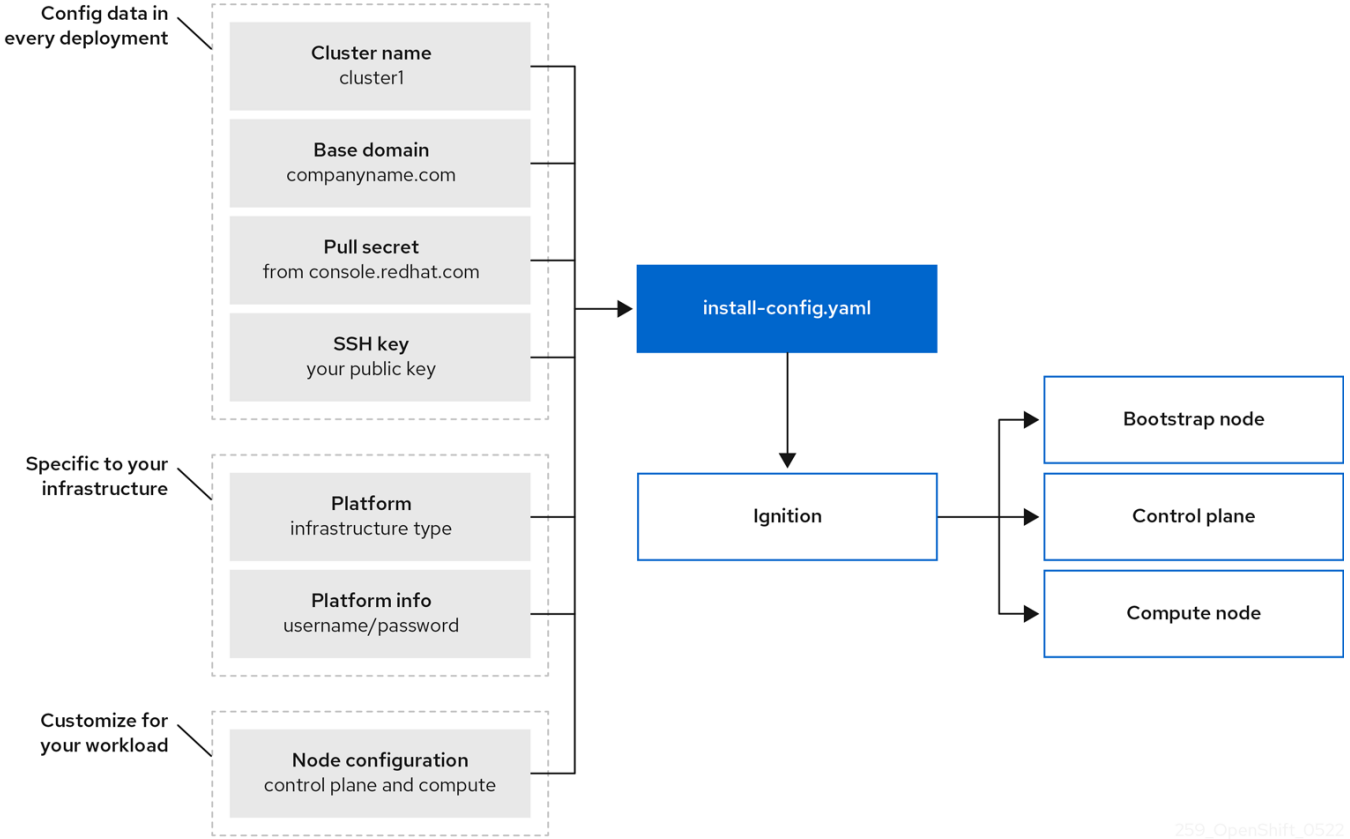


OpenShift 4 Installation

Red Hat

Offline Installation

- OpenShift **4.12.30** (Issued: 2023-08-23)
  - Issued: 2023-08-23
  - EUS (Extended Update Support) Due to Jan 17, 2025
- Need Local Registry
  - Would be installed **in Bastion**
  - TLS Enabled
  - Need Pull-Secret
  - Keep OpenShift image & Operators image
  - **Mirror Registry** (micro Quay only for OCP usage)
- Need Bootstrap Node
  - Would be removed after install
- Remote Installation ?


Config data in every deployment
Cluster name
cluster1
Base domain
companyname.com
Pull secret
from console.redhat.com
SSH key
your public key
Specific to your infrastructure
Platform
infrastructure type
Platform info
username/password
Customize for your workload
Node configuration
control plane and compute
install-config.yaml
Ignition
Bootstrap node
Control plane
Compute node

Offline Installation

- RHEL 9 / RHCOS
  - **採用 VMware vSphere 做虛擬化**
  - **所有 RHEL 皆採用 9.2 Release**
  - RHCOS Kernel Patch，將跟著 OCP 版本更新，進行修補
  - **RHCOS 4.12 Base on RHEL 8.6**
  - **RHCOS 4.13 Base on RHEL 9.2**
- OpenShift Node Role
  - 建議相同 Role 的 Node，放置於**不同的實體主機**
  - Master**必須使用SSD**
  - Logging建議使用SSD

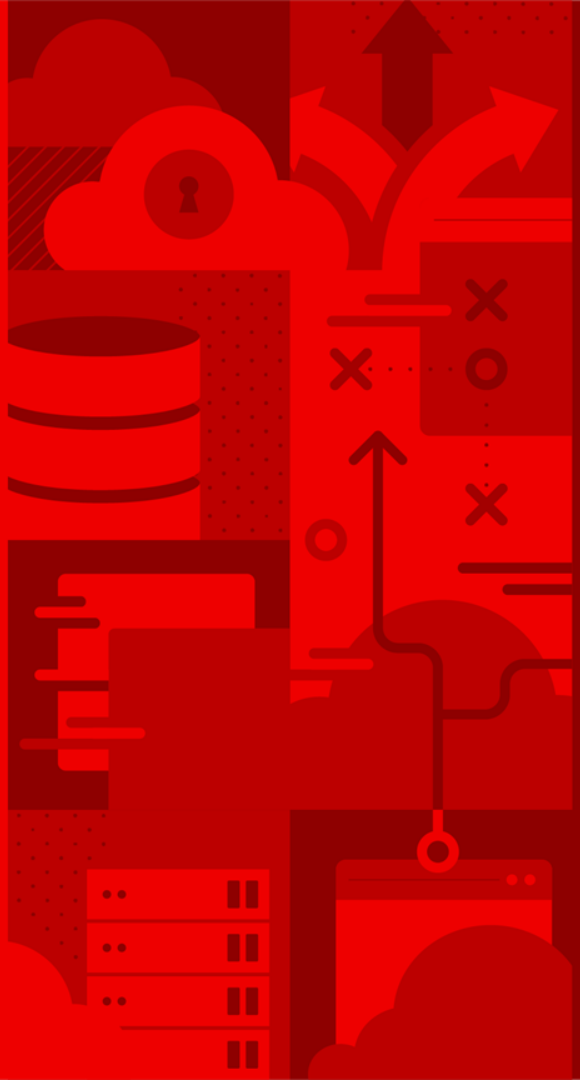
Offline Installation

- OperatorHub / Marketplace
  - **OpenShift Logging (EFK) 5.7**
  - **ElasticSearch Operator**
  - Loki Operator

  - **Red Hat Advanced Cluster Manager 2.8**
  - **Red Hat Advanced Cluster Security for Kubernetes 4.1**
  - **Red Hat Quay 3.9**
  - Red Hat Quay Container Security Operator
  - Red Hat Quay Bridge Operator
  - **KEDA / OpenShift Distributed Tracing (Jaeger / OTLP)**

  - ~~Local Volume~~
  - ~~Red Hat OpenShift Data Foundation~~
  - ~~Red Hat OpenShift Service Mesh~~

- 3-rd Operator
  - **NetApp Astra Trident 23.07**



Networking

Red Hat

Networking

- DNS Record
  - 用途：TLS 加密、Reverse Proxy (**多個 Application 共用同一組入口IP**)
  - 正解 (A Record, User Site & Admin)
    - **Node Name (需與反解對應)**
    - API URL
    - Web Console
    - Applications
  - 反解 (PTR Record, Server Site)
    - aka Server name
    - master-01.sit.etmall.ocp ←→ 10.100.10.101
    - worker-01.uat.etmall.ocp ←→ 10.100.20.201

Networking

- DNS Record
  - 用途：TLS 加密、Reverse Proxy (多個 Application 共用同一組入口IP)
  - Base Domain: **etmall.ocp**
  - Cluster Name: **sit / uat**
  - Wildcard: ***.apps.<Cluster Name>.<Base Domain>**

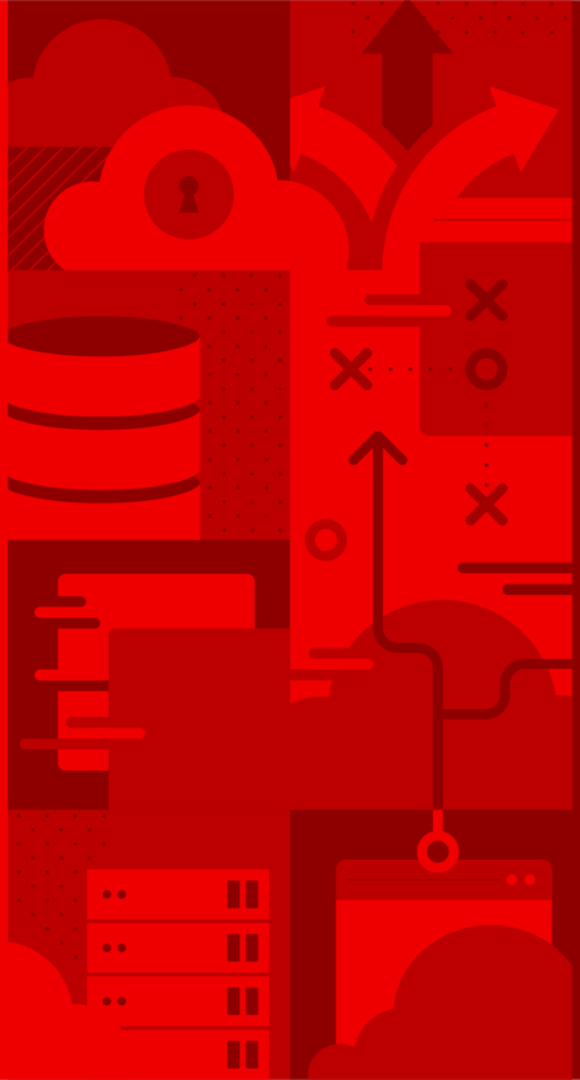
    https://**console-openshift-console.apps.sit.etmall.ocp**
    https://**central-rhacs-operator.apps.uat.etmall.ocp**

Networking

- LoadBalancer
  - HA Proxy *2 (for External User, tcp/80, tcp/443)
    - In: Everyone
    - Out: OpenShift Router
    - 對外憑證，放在這台設備上
    - Web Console 也會透過這個 LB
  - HA Proxy *2 (for Admin & oc command, tcp/6443)
    - In: Only Admin (Bastion)
    - Out: OpenShift Master
  - HA Proxy *2 (for OCP internal Usage, tcp/6443, tcp 22623)
    - In: OpenShift Node
    - Out: OpenShift Master

Networking

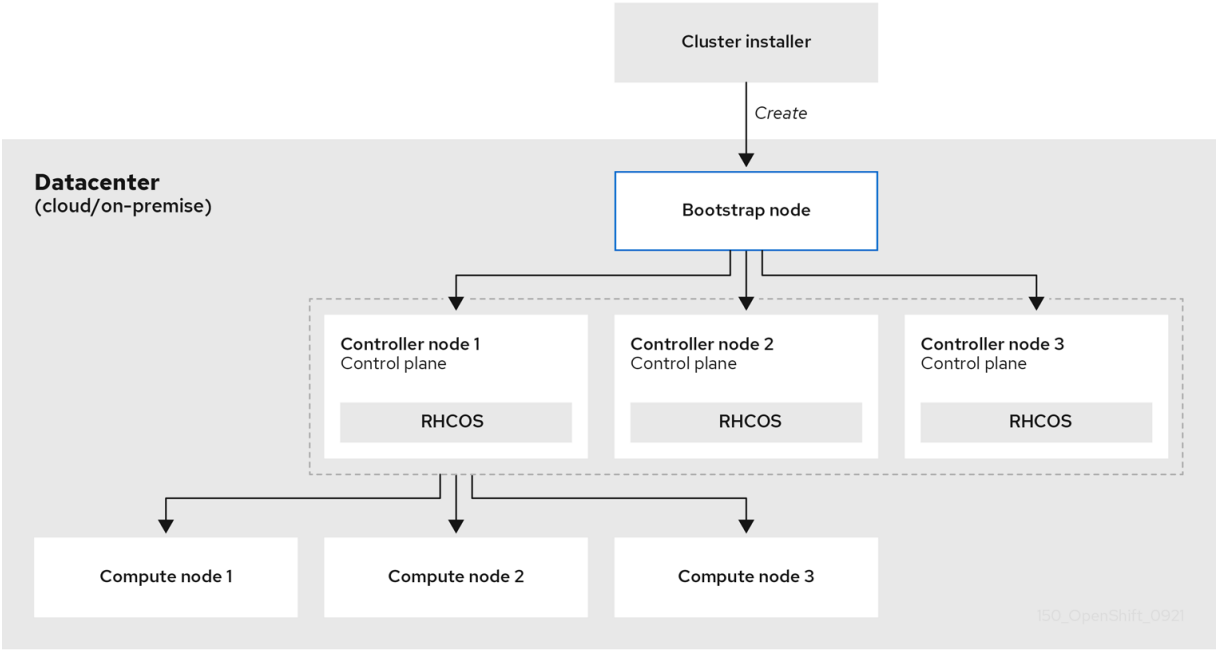
- Ingress / Router
  - 進入叢集的流量 (來自使用者的流量)
  - Same Hostname Must in Same Project
    - Route Admission Policy
      - ~~InterNamespaceAllowed~~
  - **Router Node * 2**
  - Support Protocol
    - HTTP
    - HTTPs
    - HTTP/2
    - Web Socket
    - gRPC
  - **80 / 443 Port (TCP/UDP)**
- EgressIP
  - 離開叢集的流量 (往外部系統，例如資料庫)
  - Keep Same EgressIP in Same Project
  - 必須與 Host 同網段



Node Sizing

Node Sizing

- VM Sizing
  - Support Node (VM)
    - Bastion * 1
    - HA Proxy * 2
  - OpenShift Node (VM)
    - Bootstrap * 0
    - Master * 3
    - Infra (Router) * 2
    - Infra (Logging) * 3
    - Infra (Quay / ACM / ACS) * 2
  - OpenShift Node (Bare Metal)
    - SIT+Dev Worker * 3
    - UAT Worker * 2



Red Hat

Node Sizing

- Network Sizing
  - 建議使用**全新的網段**
  - 建議位於 Internal Server Zone 之服務，放置於**同一個 L2 network**
  - RHEL / RHCOS 主機網段
    - SIT = 10.100.10.0/24
    - UAT = 10.100.20.0/24
    - **Data = 10.200.10.0/24 (SIT/UAT共用)**
    - **MGMT = 10.200.20.0/24 (BMC 使用)**
  - Cluster Network Segment (Pod IP)
    - SIT = 10.128.0.0/16
    - UAT = 10.129.0.0/16
  - Service Network Segment
    - SIT = 172.30.0.0/16
    - UAT = 172.31.0.0/16
  - Host Prefix
    - SIT = 22 (1024 Pod IP per Node)
    - UAT = 22 (1024 Pod IP per Node)

Node Sizing

- Network Sizing
  - **SIT IP Total = 9 (VM) + 2 (LB VIP) + 3 (Worker) + 2 (Egress IP) = 16**
  - **UAT IP Total = 10 (VM) + 2 (LB VIP) + 2 (Worker) + 2 (Egress IP) = 16**

  - VM IPs (SIT, total 9)
    - Master / Bootstrap * 4
    - Infra * 2 (Router)
    - Bastion / HA Proxy * 3
  - Virtual IPs
    - 2 per Cluster
  - Worker
    - SIT/Dev * 3
    - Data Lan * 3
  - Egress IP
    - **Egress IP * 2 (共用)**
    - **放置於 Router Node**

  - VM IPs (UAT, total 10)
    - Master / Bootstrap * 4
    - Infra * 3 (Router * 2 + EFK * 1)
    - Bastion / HA Proxy * 3
  - Virtual IPs
    - 2 per Cluster
  - Worker
    - UAT * 2
    - Data Lan * 3 (Worker *2 + Infra * 1)
  - Egress IP
    - **Egress IP * 2 (共用)**
    - **放置於 Router Node**

Node Sizing

- Storage Provisioner
  - 3rd-Provider
    - **NetApp Astra Trident**
  - iSCSI (via Trident ontap-san)
    - RWO (Read Write Once)
    - Block
    - **Monitoring、Logging(EFK)** 使用
    - ACS DB 使用
  - NFS (via Trident ontap-nas)
    - RWX (Read Write Many)
    - Filesystem
  - S3
    - Loki
    - Quay
    - 直接設定 URL，不透過 Trident

Node Sizing

- Worker Sizing
  - 需要多少 CPU / Memory / Disk ?
  - 一共需要多少台主機

  - How Many Applications?
  - How Much Resource Usage per Application?
  - How Many Replicas for High Available?
  - N + 1 ?

  - 假設 20 隻程式，各需要 4GB的RAM，每隻程式需要 2份副本，每台 Worker 有 32GB 的 RAM
    - 20 * 4GB * 2 / 32GB + 1 = 5 + 1 = 6台主機
  - 假設 5 隻程式，各需要 0.5Core vCPU，每隻程式需要 3份副本，每台 Worker 有4Core 的 vCPU
    - 5 * 0.5 * 3 / 4 + 1 = 1.875 + 1 = 3 台主機

Node Sizing

- Worker Sizing
  - 需要多少 CPU / Memory / Disk ?
  - 一共需要多少台主機

  - How Many Applications?
  - How Much Resource Usage per Application?
  - How Many Replicas for High Available?
  - N + 1 ?

  - 取決於 SLA 能容許的範圍 (只能多、不能少)

    - 2台 worker 至少預留 50% 資源
    - 3台 worker 至少預留 35% 資源
    - 4台 worker 至少預留 25% 資源
    - 5台 worker 至少預留 20% 資源
    - .....

Node Sizing

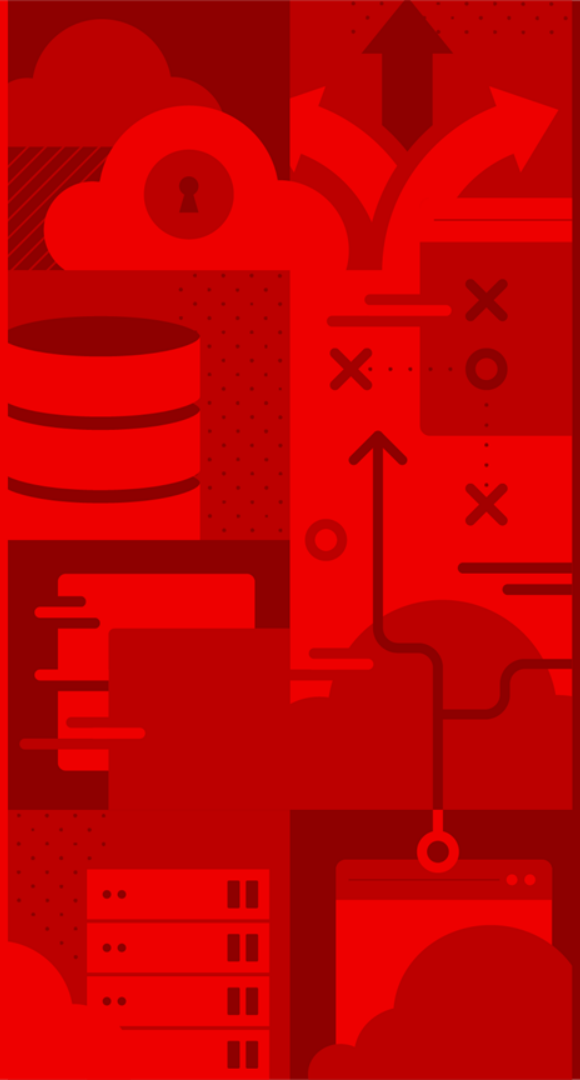
- Application Quota
  - 一定要設定 **Resource Limit**
  - Resource Quota
    - Request (最小值)
    - Limit (最大值)
  - **Java 程式的情況**
    - CPU
      - 達到 Limit 就會速度變慢
      - Over Commit
    - Memory
      - 達到 Limit 會被系統 Kill 掉
      - **給多少用多少，作為Cache用途**

Node Sizing

- Disk Sizing
  - Bastion:
    - **Mirror Registry 使用**
    - 300GB
  - RHCOS:
    - 100GB for OS
    - 300GB for Worker Node
  - Master Node
    - **建議 SSD**
    - **IOPS: 3,600 ~ 16,000**
    - Throughput: 60MBps ~ 250 MBps

Node Sizing

- Disk Sizing
  - Monitoring / Metrics:
    - 100GB per Infra Node
    - 經驗上，100GB per Infra Node， 可存放約 15天
  - Logging / EFK:
    - **建議使用SSD / SAS**
    - Infrastructure + Audit
      - 50GB per Node per Day
    - Application
      - **1 GB per Pod per Day**
    - Total
      - 50GB * **3 (Days)** = 150GB per Node
      - 1GB * 20 (Apps) * **7 (Days)** * 2 (Replicas) * 0.66 (per EFK Node) = 185 GB per Node
      - **335 GB per Node**



Security

Red Hat

OpenShift Security

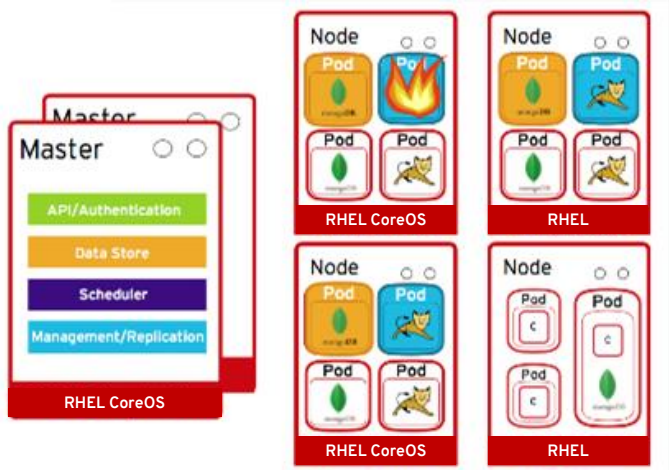
Features, mechanisms and processes for container and platform isolation

Red Hat

SECURING THE CONTAINER PLATFORM

Security Features Include

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



Red Hat

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
```

Security (OpenShift)

- SCC (Security Context Constraints)
  - 預設不允許 root (UID = 0) 執行
  - 預設不允許修改 kernel
  - 預設不允許掛載 HostPath (宿主機上的檔案)

  - **不同 Project 之間，預設將產生不同的 Random UID/GID**
    - 避免不同Projct之間的 Container，可以讀取對方的資料

  - Custom SCC
    - **Priority Must <10**
    - 建議用新增的方式，**不要修改預設的SCC**
    - 建議使用 ServiceAccount 及 RoleBinding 進行授權

IDENTITY AND ACCESS MANAGEMENT

OpenShift includes an OAuth server, which does three things:

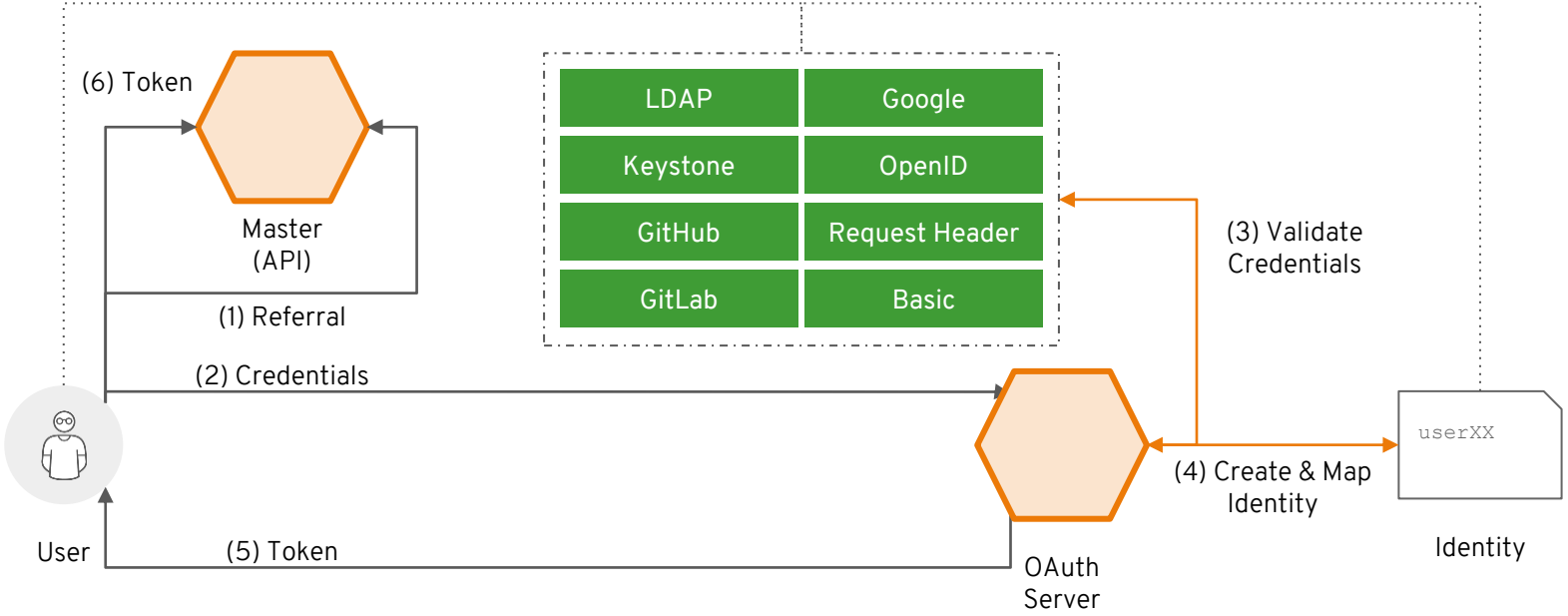
- Identifies the person requesting a token, using a configured identity provider
- Determines a mapping from that identity to an OpenShift user
- Issues an OAuth access token which authenticates that user to the API

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)

Identity and Access Management



RESTRICT ACCESS BY NEED TO KNOW

Role		Users					
		admin1	admin2	developer1	developer2	tester1	tester2
		Groups					
		admin-group		developer-group		tester-group	
Project	project-cicd	admin		edit		view	
	project-uat	admin		view		edit	

For more information see: Managing RBAC in OpenShift

RBAC 預設 Roles

admin：可管理除了 project quota 外其它所有資源

basic-user：可查詢 project 和用户的基本資料

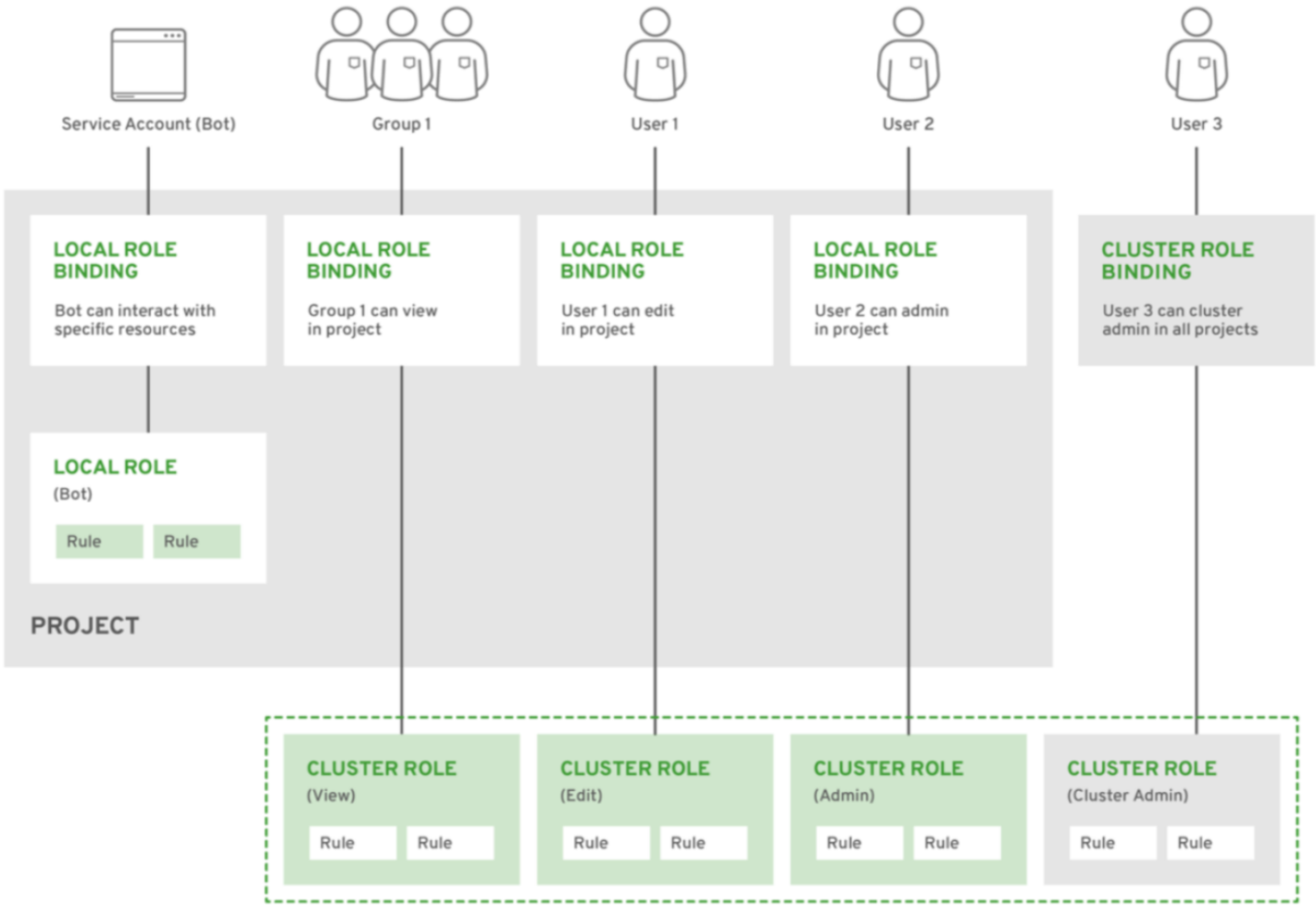
cluster-admin：集群管理者

cluster-status：可以查詢集群狀態

edit：可修改除了 Role 和 Binding 外的其它資源

self-provisioner：可以新增 project

view：不能進行任何更改，但是可以查看


Service Account (Bot)
Group 1
User 1
User 2
User 3
LOCAL ROLE BINDING
Bot can interact with specific resources
LOCAL ROLE BINDING
Group 1 can view in project
LOCAL ROLE BINDING
User 1 can edit in project
LOCAL ROLE BINDING
User 2 can admin in project
CLUSTER ROLE BINDING
User 3 can cluster admin in all projects
LOCAL ROLE
(Bot)
Rule
Rule
PROJECT
CLUSTER ROLE
(View)
Rule
Rule
CLUSTER ROLE
(Edit)
Rule
Rule
CLUSTER ROLE
(Admin)
Rule
Rule
CLUSTER ROLE
(Cluster Admin)
Rule
Rule

Security (RBAC)

- User
  - 使用 OAuth 保存帳號 (Azure AD)
    - https://cloud.redhat.com/blog/openshift-blog-aro-aad
  - 備案：使用 LDAP 保存帳號 (本地AD)
- Group
  - admin (cluster-admin成員)
  - Other?
- ServiceAccount
  - 程式使用
  - 作為程式 Lifecycle 的權限控管
  - 可搭配 RBAC進行 SCC 、 Pull-Secret 、 Image Stream 的授權
  - Azure DevOps 使用 (pipeline agent) ?

CVE-2021-33909

Public on July 20, 2021

Important Impact
What does this mean?

7.0
CVSS v3 Base Score
CVSS Score Breakdown

Description

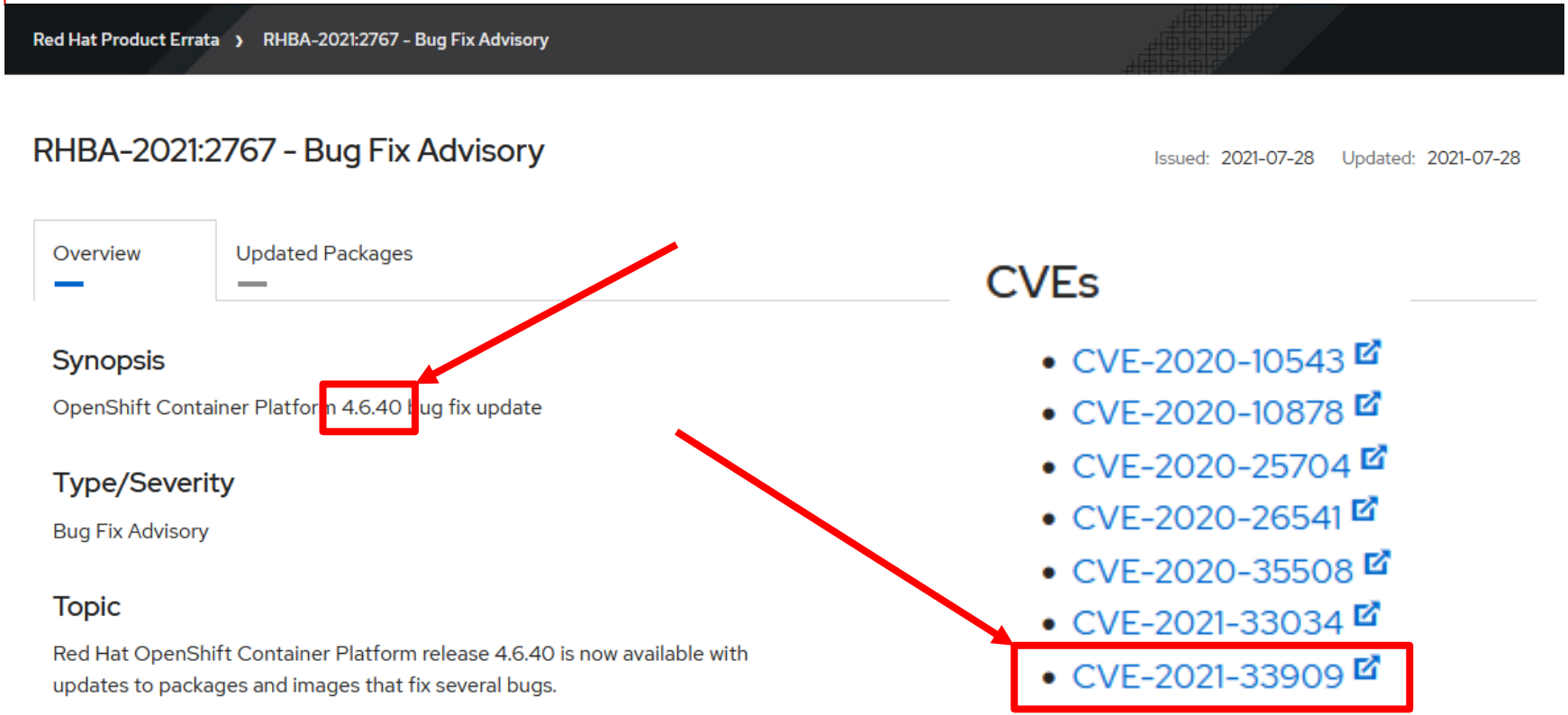
An out-of-bounds write flaw was found in the Linux kernel's seq_file in the Filesystem layer. This flaw allows a local attacker with a user privilege to gain access to out-of-bound memory, leading to a system crash or a leak of internal kernel information. The issue results from not validating the size_t-to-int conversion prior to performing operations. The highest threat from this vulnerability is to data integrity, confidentiality and system availability.

Vulnerability Response

https://access.redhat.com/security/vulnerabilities/RHSB-2021-006

Additional Information

- Bugzilla 1970273: CVE-2021-33909 kernel: size_t-to-int conversion vulnerability in the filesystem layer
- CWE-787: Out-of-bounds Write
- FAQ: Frequently asked questions about CVE-2021-33909



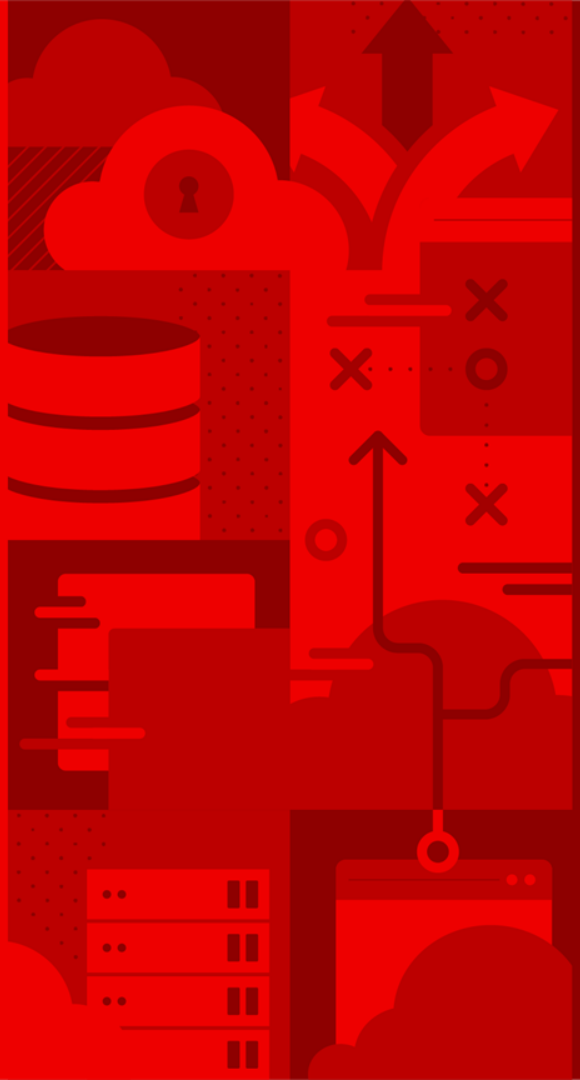
https://access.redhat.com/security/cve/CVE-2021-33909

Security (Base OS)

- Control / Data Plane
- End-to-End Encryption

- RHEL 9 (Bastion / HAProxy)
  - Firewall
  - **SELinux**
  - **Registry use TLS & Pull-Secret**
  - NTP / Chrony

- RHCOS (RedHat CoreOS)
  - Minimal
  - **Immutable (不可變的)**
  - Without Password
  - Without Package Manager (yum / dnf)
  - Best for Container Usage
  - 集中管理
    - Machine Config Pool
  - **Rolling Upgrade**

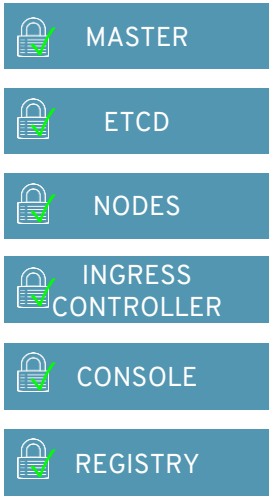


SSL Certificate

Red Hat

CERTIFICATE MANAGEMENT

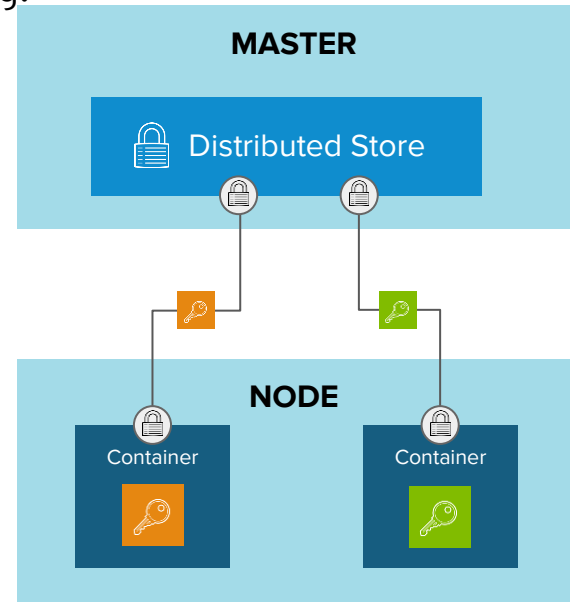
- Certificates are used to provide secure connections to
  - master and nodes
  - Ingress controller and registry
  - etcd
- Certificate rotation is automated
- Optionally configure external endpoints to use custom certificates
- For example:
  Requesting and Installing Let's Encrypt Certificates for 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*
- Never rest on the nodes



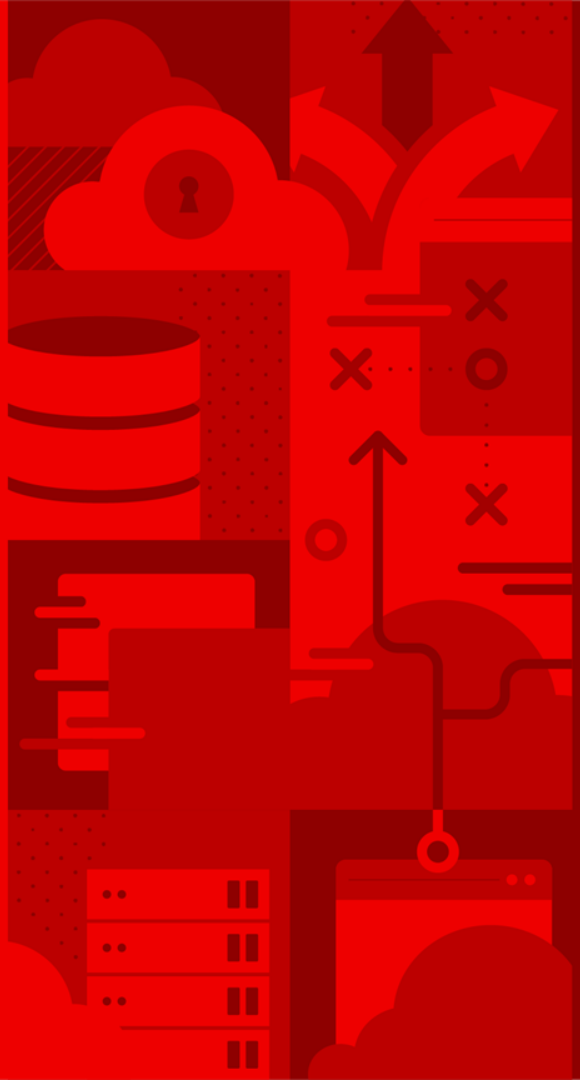
* Full disk encryption is supported for AWS, VSphere. Encrypting the etcd datastore is not available in OCP 4.1. Targeted for OCP 4.3.

SSL Certificate

- CA import
  - Need X509v3 **SAN (Subject Alternative Name)**
    - OpenShift 4.6 後，要求之憑證格式
  - **Registry 使用 自簽CA** ~~/客戶核發的憑證~~
- OpenShift Route (Edge / Re-encrypt)
  - 預設自動設定 HTTP Header
    - X-Forward-For (提供**真實的來源IP**)
  - 強制 TLS v1.2
  - 關閉不安全的加密演算法
    (會隨著 OpenShift 更新，同步更新Router的安全性)

SSL Certificate

- HAProxy SNI Backend
  - **SNI (Server Name Indication)**
    - 提供給多個不同的 HTTPs Backend，去共用同一個 SLB IP (Frontend)
  - **Need HAProxy 1.8 以上**，才支援此功能
    - **RHEL 8 => HAProxy 1.8**
    - **RHEL 9 => HAProxy 2.4**
- LoadBalancer
  - HA Proxy *2 (Active-Standby)
  - **對外憑證，放在這組設備上**
- Certificate Rotate
  - 第一次安裝，24hr即過期
  - 時間到期後，OpenShift內部溝通用的憑證，會**自動更新 (Rotate)**



OpenShift Logging

Logging 5.7 for OpenShift 4.12

<< NEW >>

Vector as alternate collector

Loki as alternate log store

Major updates and features

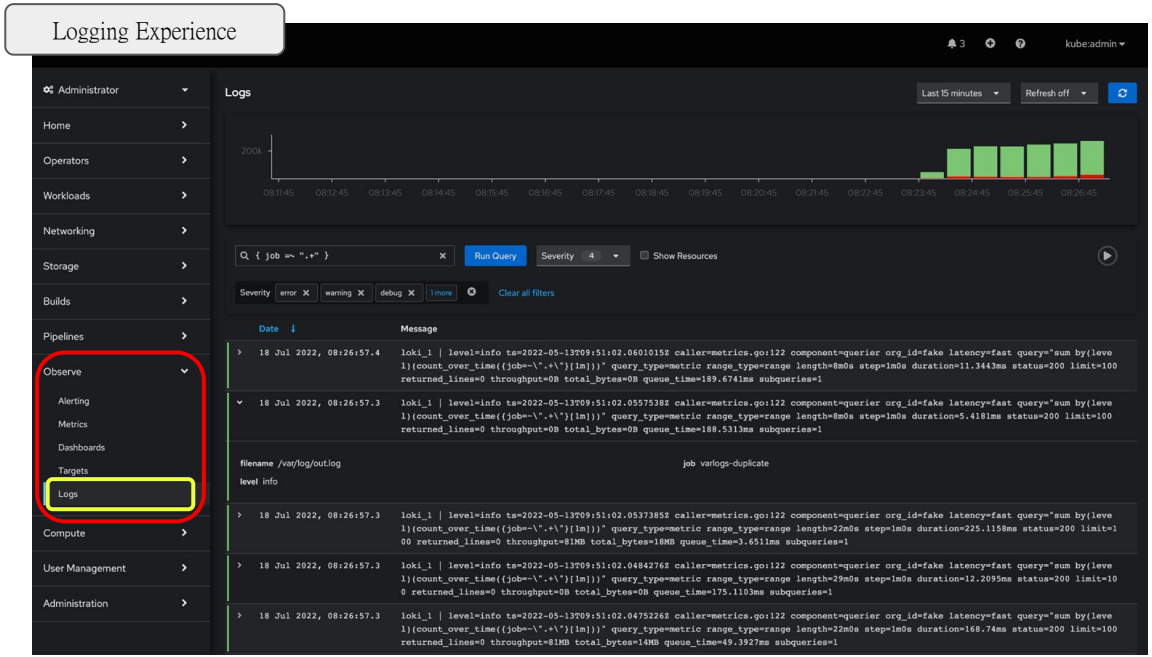
- maxUnavailable of 'collector' daemonset reducing upgrade time
- Log exploration natively inside the OpenShift Console
- Upgrade fluent to ruby 2.7 and latest dependencies
- Pod labels for k8s are preserved
- Support Cloudwatch output for Vector
- CloudWatch log forwarding add-on supports STS installations

PM: Roger Florén

Logging 5.5: OpenShift Logging UI Experience

OpenShift Console Logging Experience

- Continue to work towards a consistent and simplified Observability User Experience by introducing a logging view in the console:
  - **Observe > Logs**: exposes log information from the underlying storage via an API, queried by the console to retrieve contextualized logs



Red Hat

EFK Overview

Components

- **Elasticsearch:** OpenShift 用來保存Log的資料庫，並用來執行 Log 分析
- **Fluentd:** 用來提取 Log ，並傳送至 Elasticsearch 保存
- **Kibana:** 用來查看 Elasticsearch 的 Web UI
- **Curator:** Cron Job ，定期清理與替換過舊的 Log (OCP 4.7 之後移除)
- **Index Management:** Cron Job，由 ElasticSearch Operator 負責維護，分為 App、Audit、Infra 三種

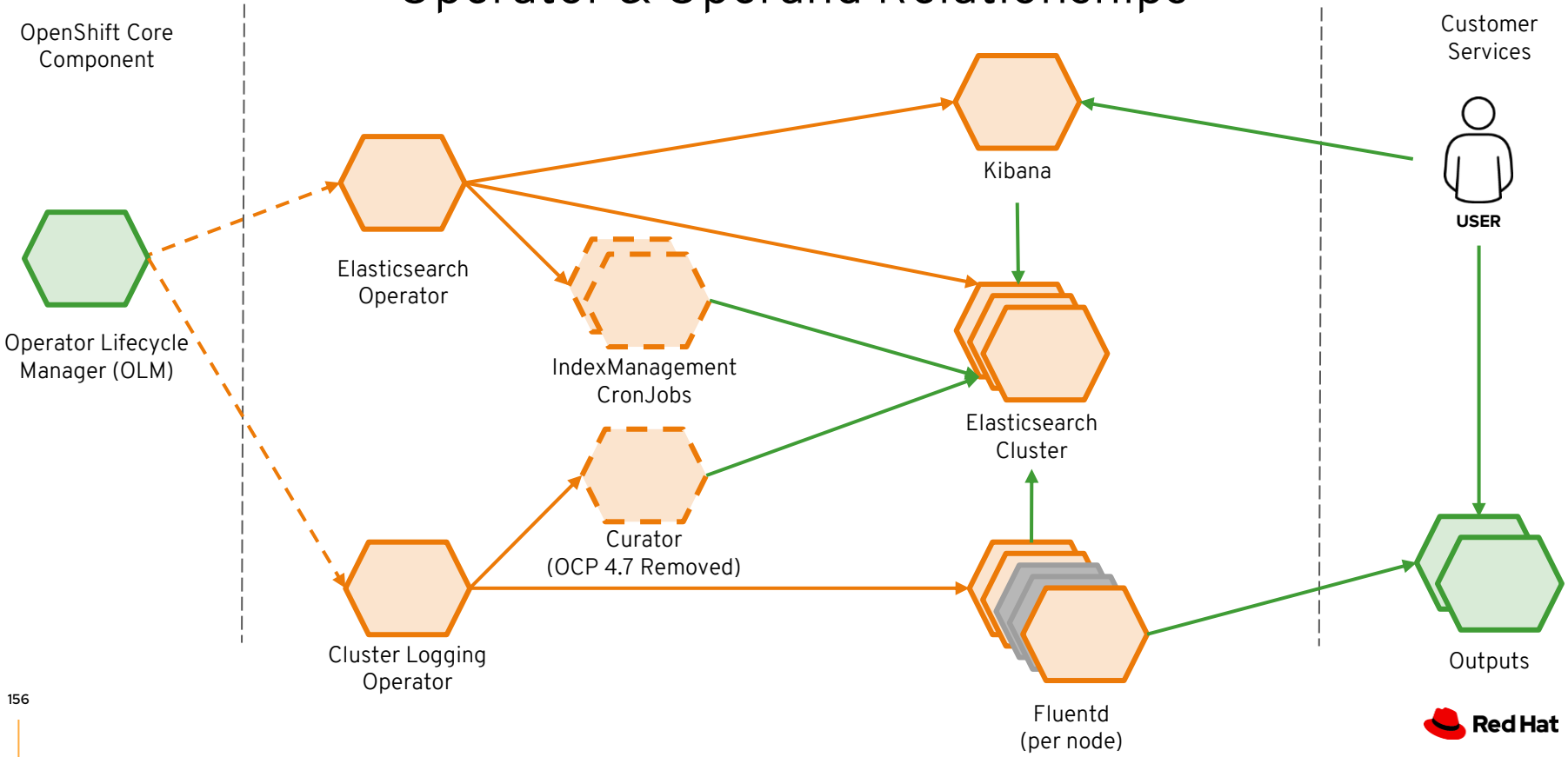
Access control

- Cluster administrators 可以查看所有的 log
- Users 只能查看，他們自己相關的 projects 的 log

Ability to forward logs elsewhere

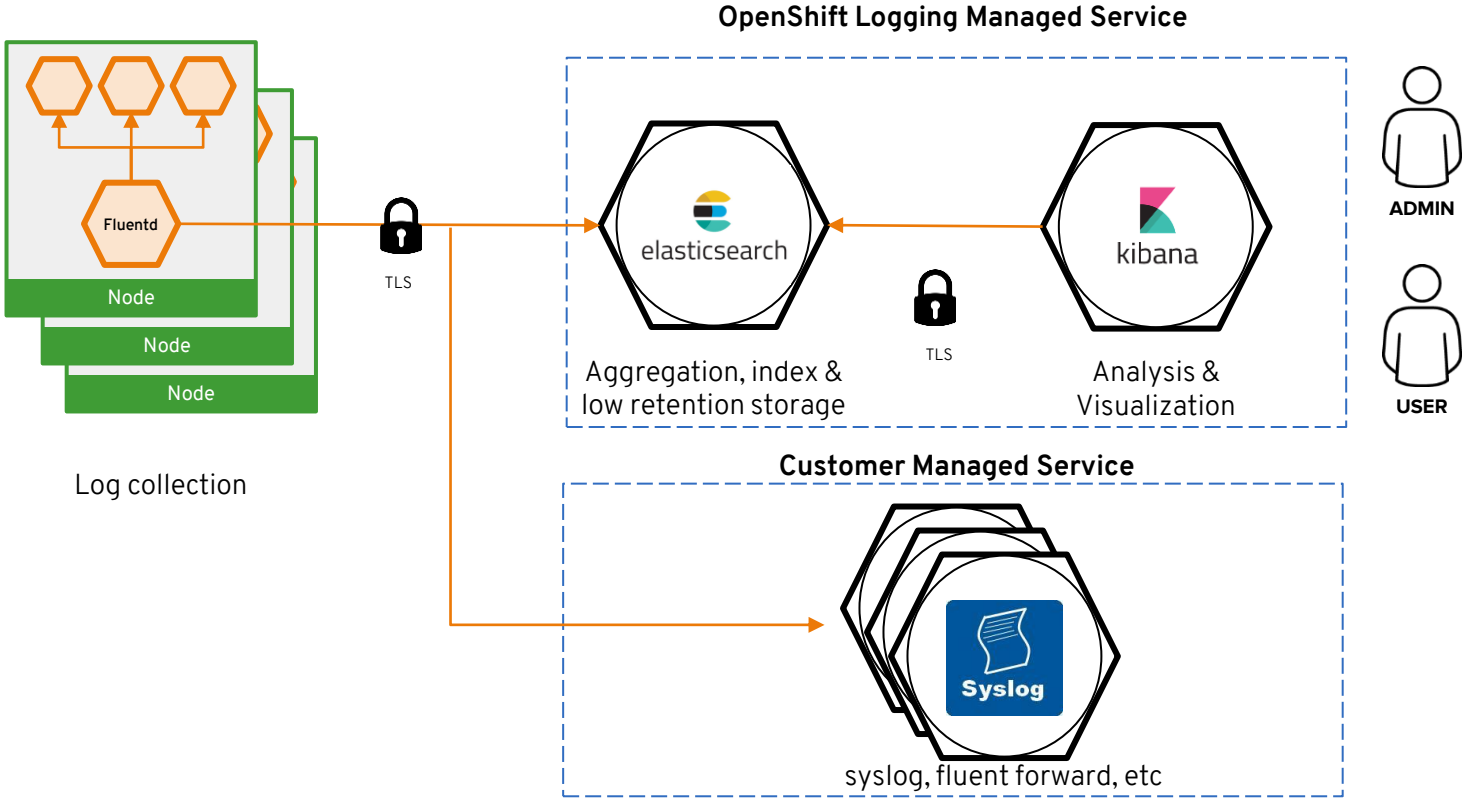
- External elasticsearch, Splunk, syslog, etc

Operator & Operand Relationships



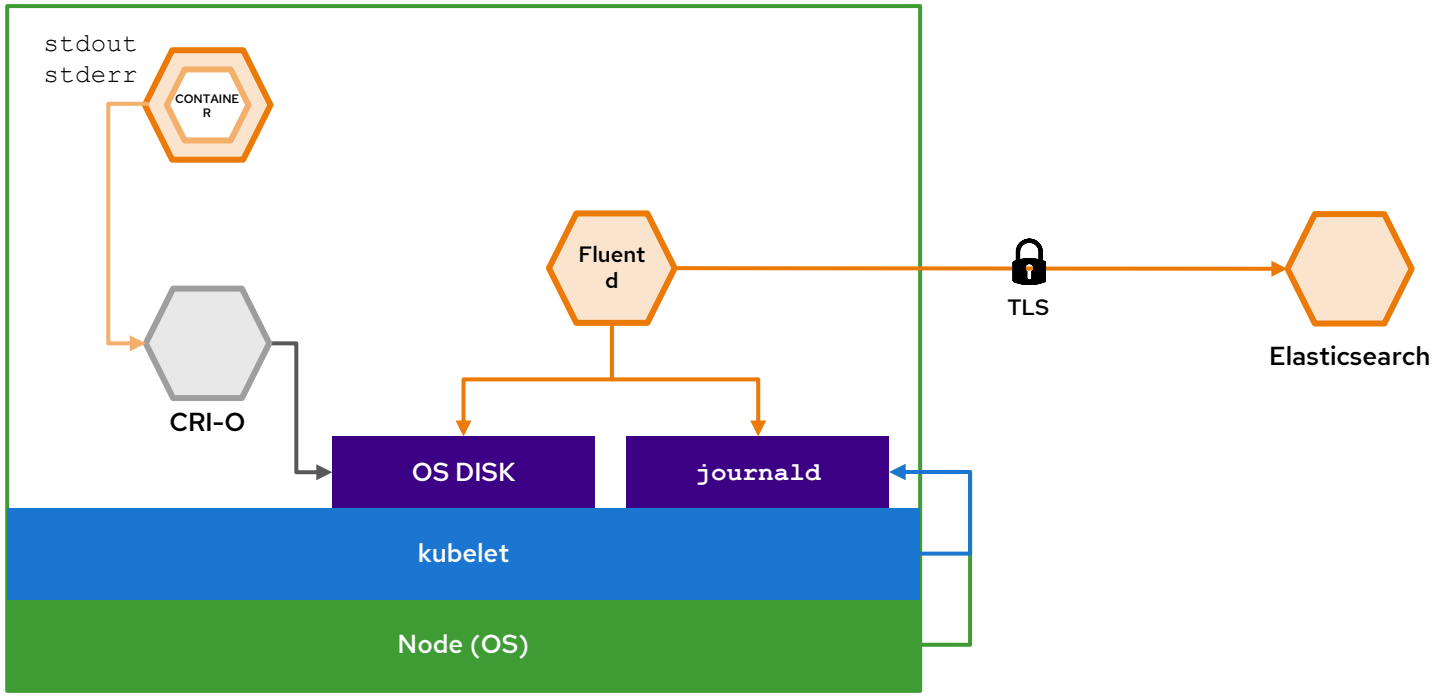
Red Hat

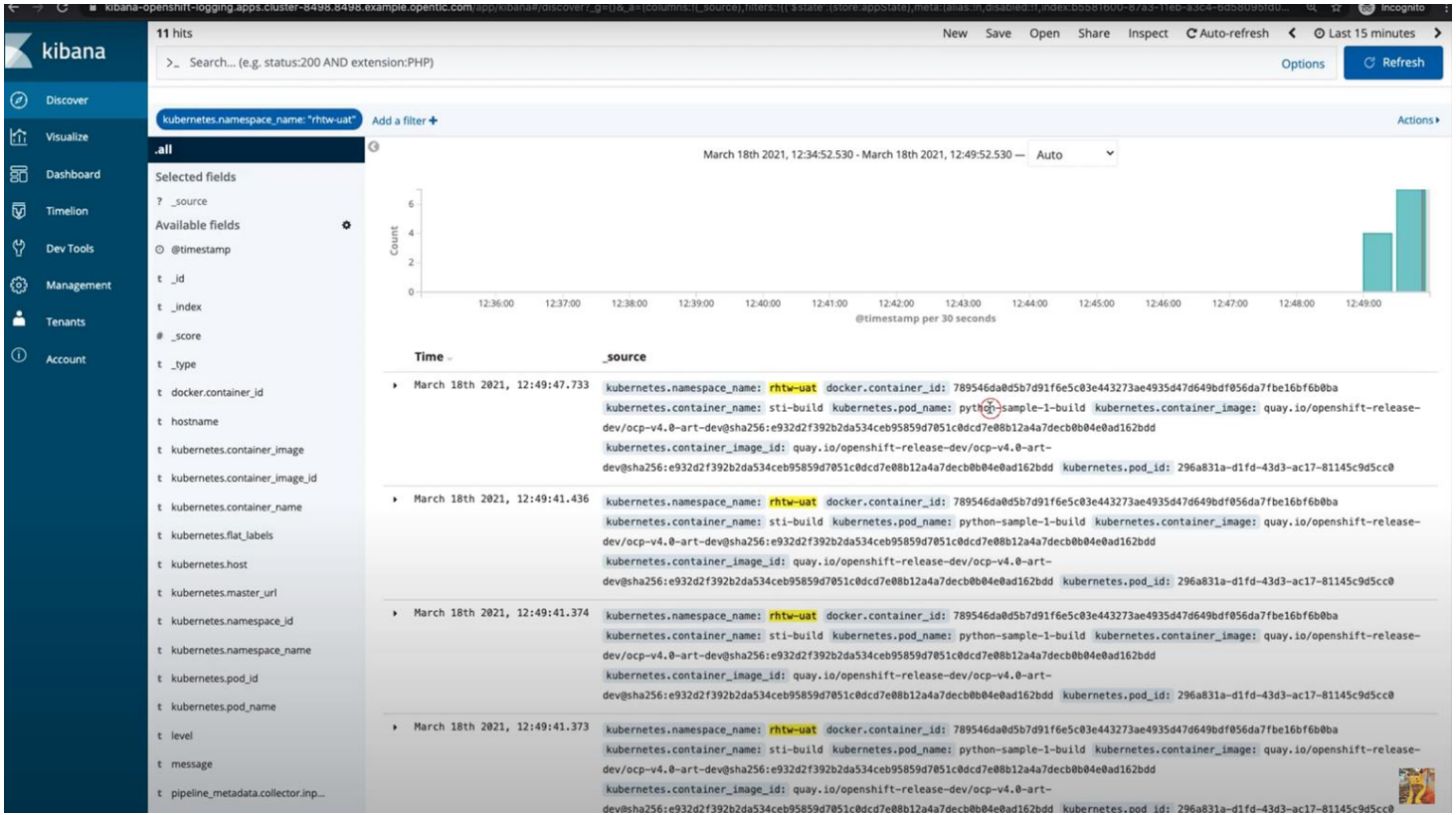
Log data flow in OpenShift (cluster-wide)



Red Hat

Log data flow in OpenShift (node-wide)





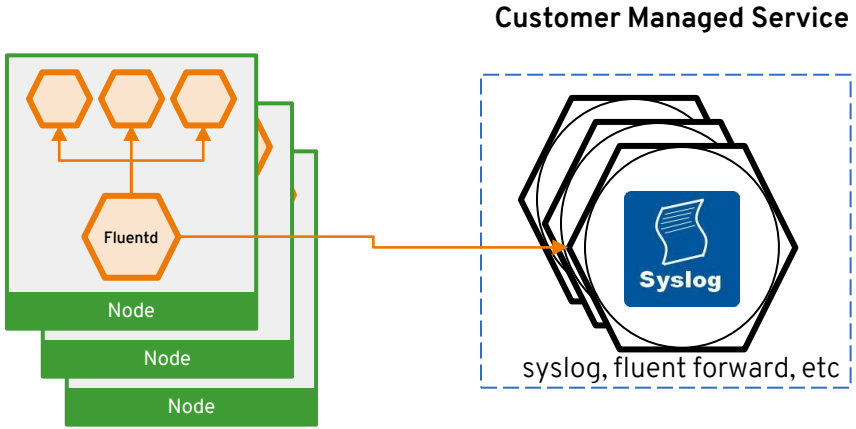
Source: https://youtu.be/HXkB5qH_518
https://docs.openshift.com/container-platform/4.6/logging/cluster-logging-deploying.html

Red Hat

Log Forwarder

Red Hat

提供平台 Infra Log 轉發至第三方系統

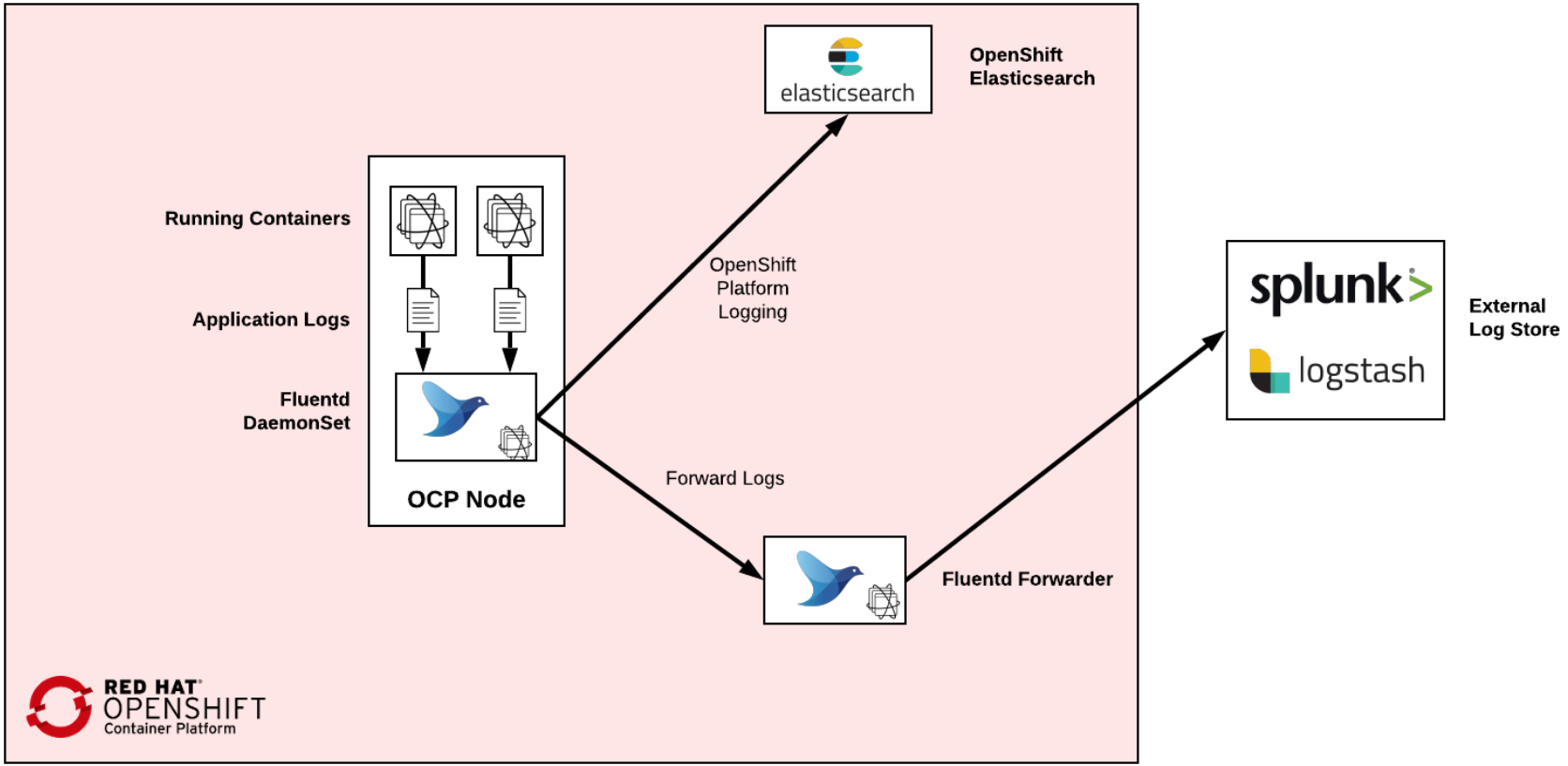


Log collection

- OpenShift Logging 5.7 ，Log 轉發支援以下第三方系統：
  - Amazon CloudWatch
  - Elasticsearch
  - fluentd
  - logstash
  - Loki Stack
  - kafka
  - Syslog
  - Google Cloud Logging
  - Splunk HTTP Event Collector (HEC)

Table 1. Logging 5.7 outputs

Output	Protocol	Tested with	Fluentd	Vector
Cloudwatch	REST over HTTP(S)		✓	✓
Elasticsearch v6		v6.8.1	✓	✓
Elasticsearch v7		v7.12.2, 7.17.7	✓	✓
Elasticsearch v8		v8.4.3	✓	✓
Fluent Forward	Fluentd forward v1	Fluentd 1.14.6, Logstash 7.10.1	✓	
Google Cloud Logging				✓
HTTP	HTTP 1.1	Fluentd 1.14.6, Vector 0.21	✓	✓
Kafka	Kafka 0.11	Kafka 2.4.1, 2.7.0, 3.3.1	✓	✓
Loki	REST over HTTP(S)	Loki 2.3.0, 2.7	✓	✓
Splunk	HEC	v8.2.9, 9.0.0		✓
Syslog	RFC3164, RFC5424	Rsyslog 8.37.0-9.el7	✓	✓



備註： 此配圖為 OpeShift Logging 4.6，新版 OpenShift Logging 5.6 已原生支援 splunk 轉發 Log

https://cloud.redhat.com/blog/forwarding-logs-to-splunk-using-the-openshift-log-forwarding-api

```yaml
apiVersion: "logging.openshift.io/v1"
kind: ClusterLogForwarder
metadata:
  name: instance 1
  namespace: openshift-logging 2
spec:
  outputs:
   - name: elasticsearch-insecure 3
     type: "elasticsearch" 4
     url: http://elasticsearch.insecure.com:9200 5
   - name: elasticsearch-secure
     type: "elasticsearch"
     url: https://elasticsearch.secure.com:9200
     secret:
        name: es-secret 6
  pipelines:
   - name: application-logs 7
     inputRefs: 8
     - application
     - audit
     outputRefs:
     - elasticsearch-secure 9
     - default 10
     labels:
       logs: application 11
   - name: infrastructure-audit-logs 12
     inputRefs:
     - infrastructure
     outputRefs:
     - elasticsearch-insecure
     labels:
       logs: audit-infra
```

Forward 至 外部 elastic search 端點設定

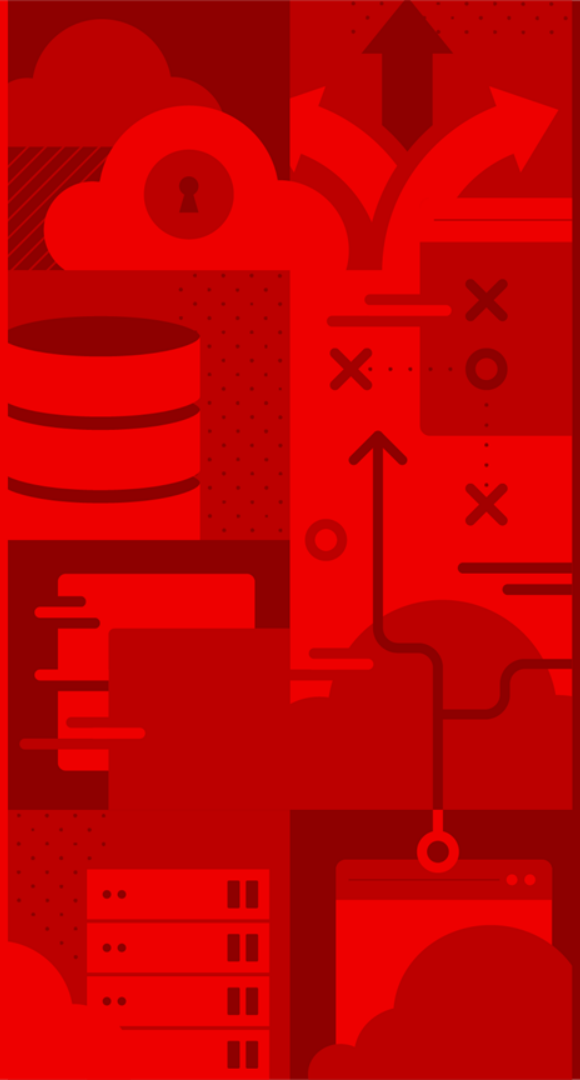
定義系統哪些 log 要用哪個 forward 設定

Log 類型

- `application`. Container logs generated by user applications running in the cluster, except infrastructure container applications.
- `infrastructure`. Container logs from pods that run in the `openshift*`, `kube*`, or `default` projects and journal logs sourced from node file system.
- `audit`. Logs generated by the node audit system (auditd) and the audit logs from the Kubernetes API server and the OpenShift API server.

Ref: https://docs.openshift.com/container-platform/4.6/logging/cluster-logging-external.html#cluster-logging-collector-log-forward-es_cluster-logging-external

Red Hat



OpenShift Monitoring

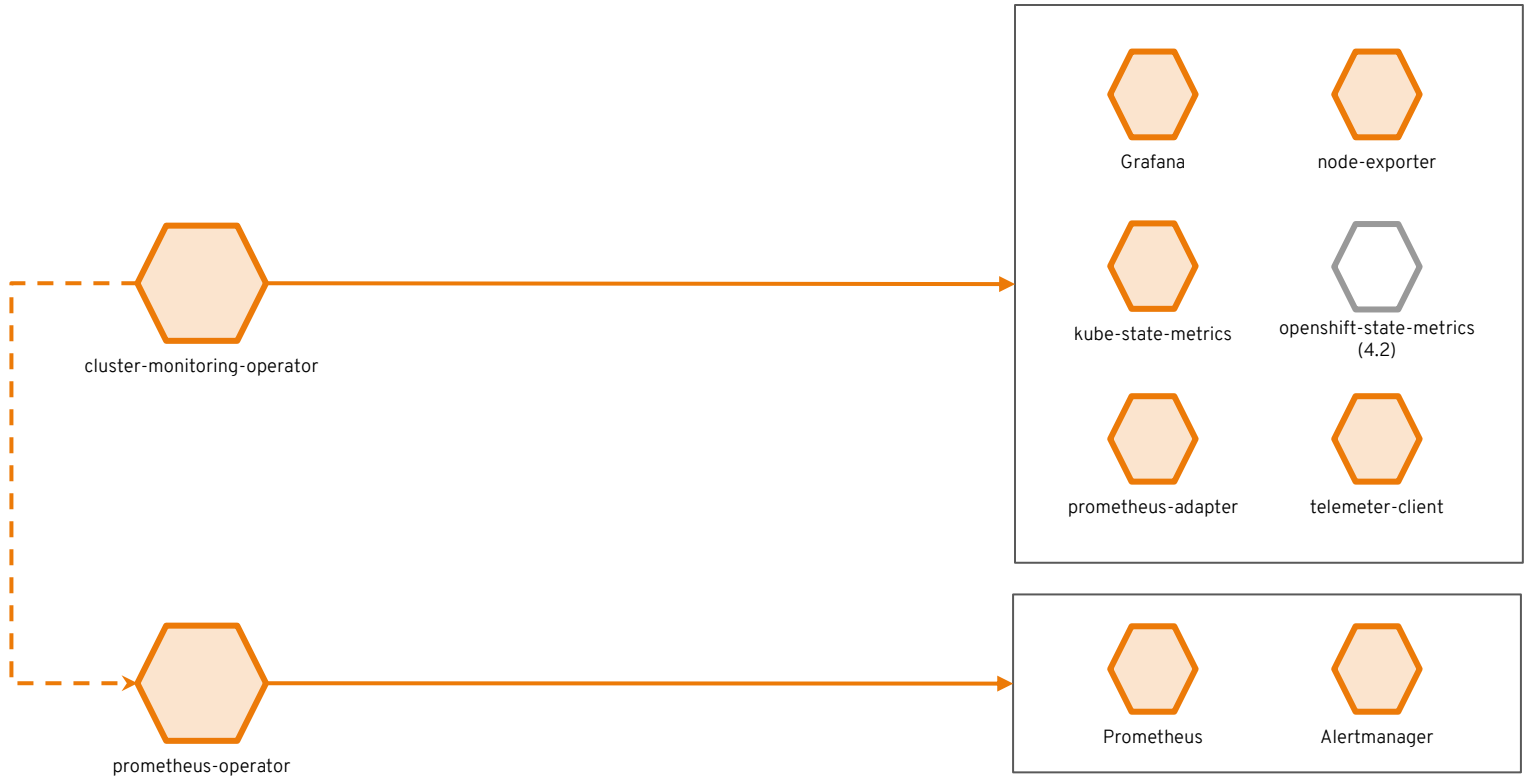
Red Hat

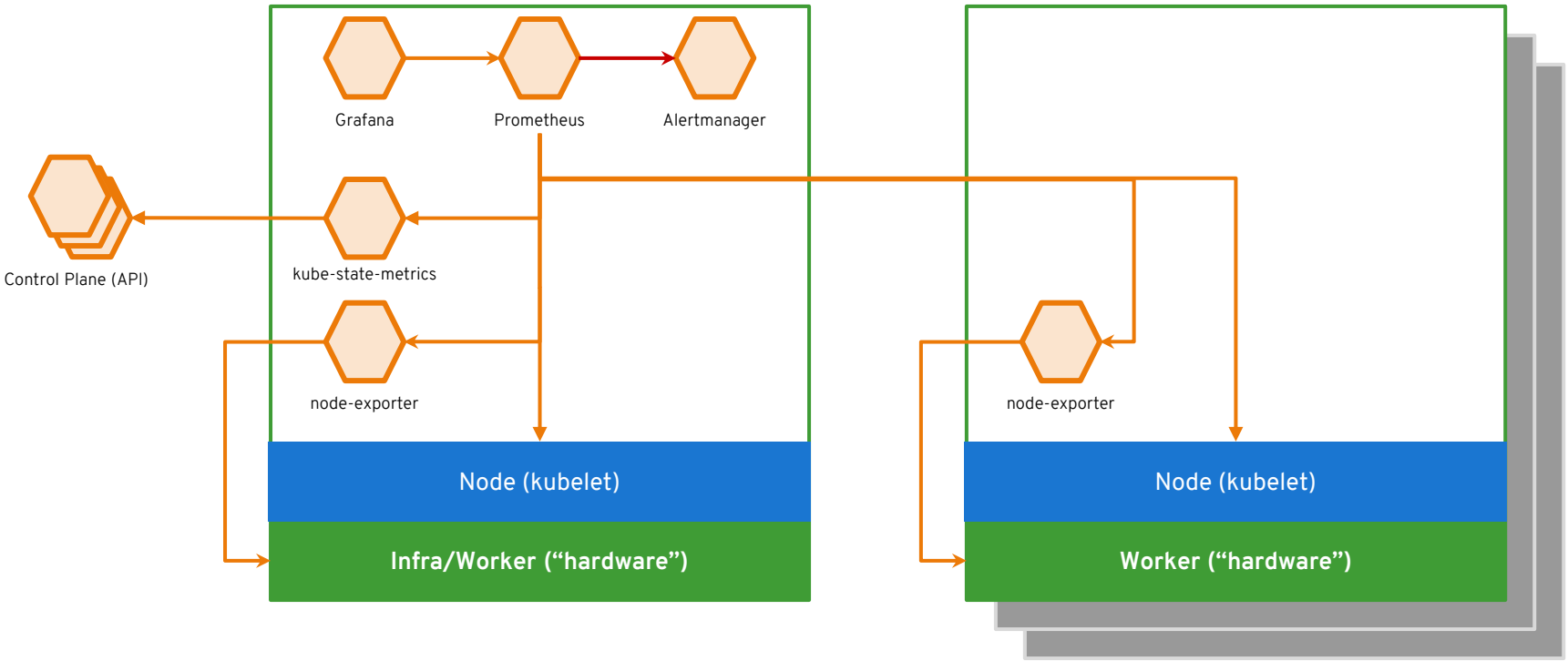
OpenShift Cluster Monitoring

Metrics collection and storage via Prometheus, an open-source monitoring system time series database.

Alerting/notification via Prometheus' Alertmanager, an open-source tool that handles alerts send by Prometheus.

Metrics visualization via Grafana, the leading metrics visualization technology.


Grafana
node-exporter
kube-state-metrics
openshift-state-metrics
(4.2)
prometheus-adapter
telemeter-client
cluster-monitoring-operator
prometheus-operator
Prometheus
Alertmanager

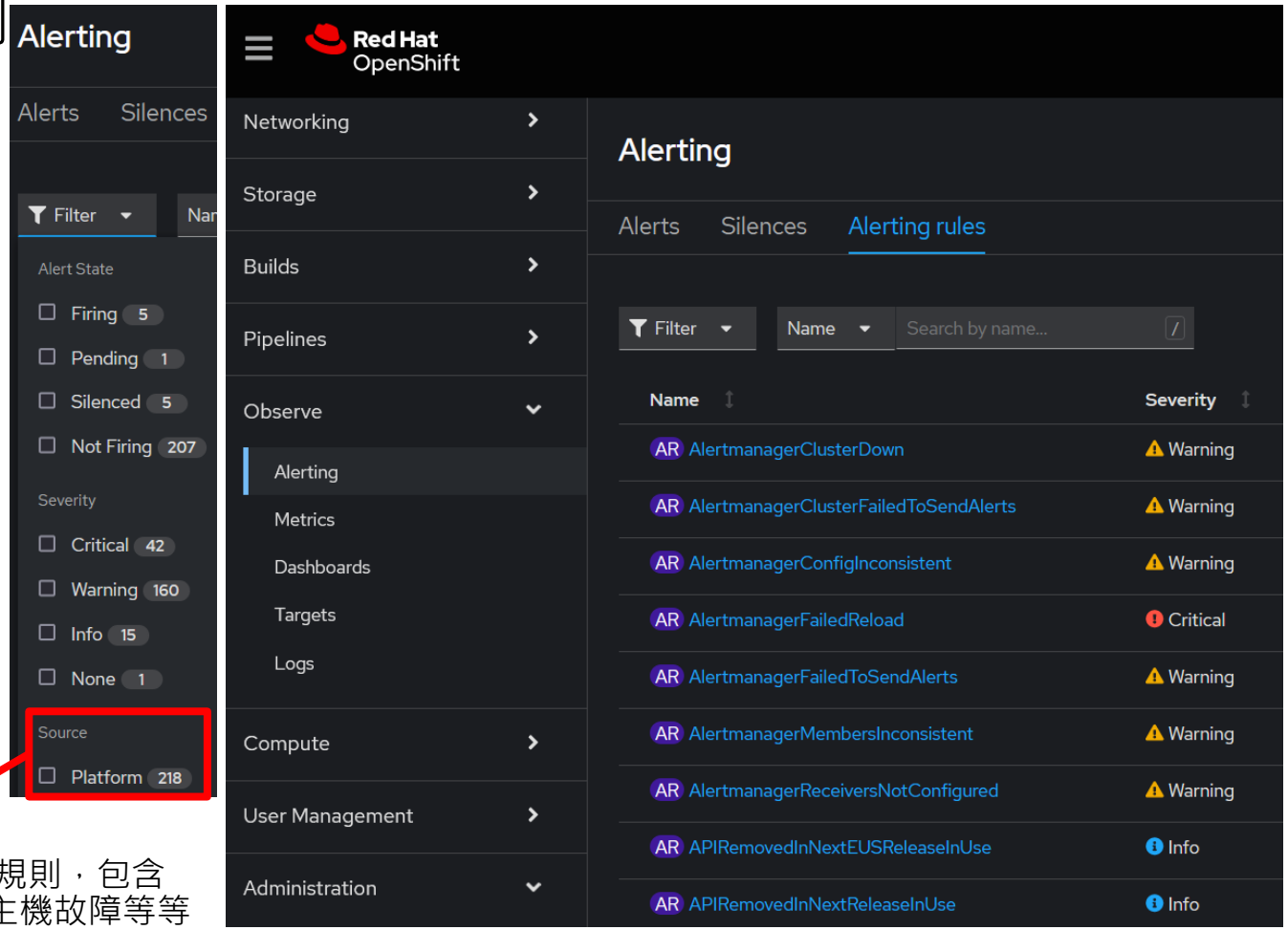

Grafana
Prometheus
Alertmanager
Control Plane (API)
kube-state-metrics
node-exporter
node-exporter
Node (kubelet)
Node (kubelet)
Infra/Worker (“hardware”)
Worker (“hardware”)

OpenShift 4.10 後
取消外部 Grafana 元件，
功能與 Web Console 整合

1. Red Hat OpenShift 4 提供官方內建及既有 Grafana 儀表板，並無提供支援客製化能力。
2. 若需客製化 Grafana 儀表板需求，則需要利用 Grafana Operator 另外部署 Grafana Dashboard 並客製化 Dashboard，並採集既有 Prometheus Datasource。

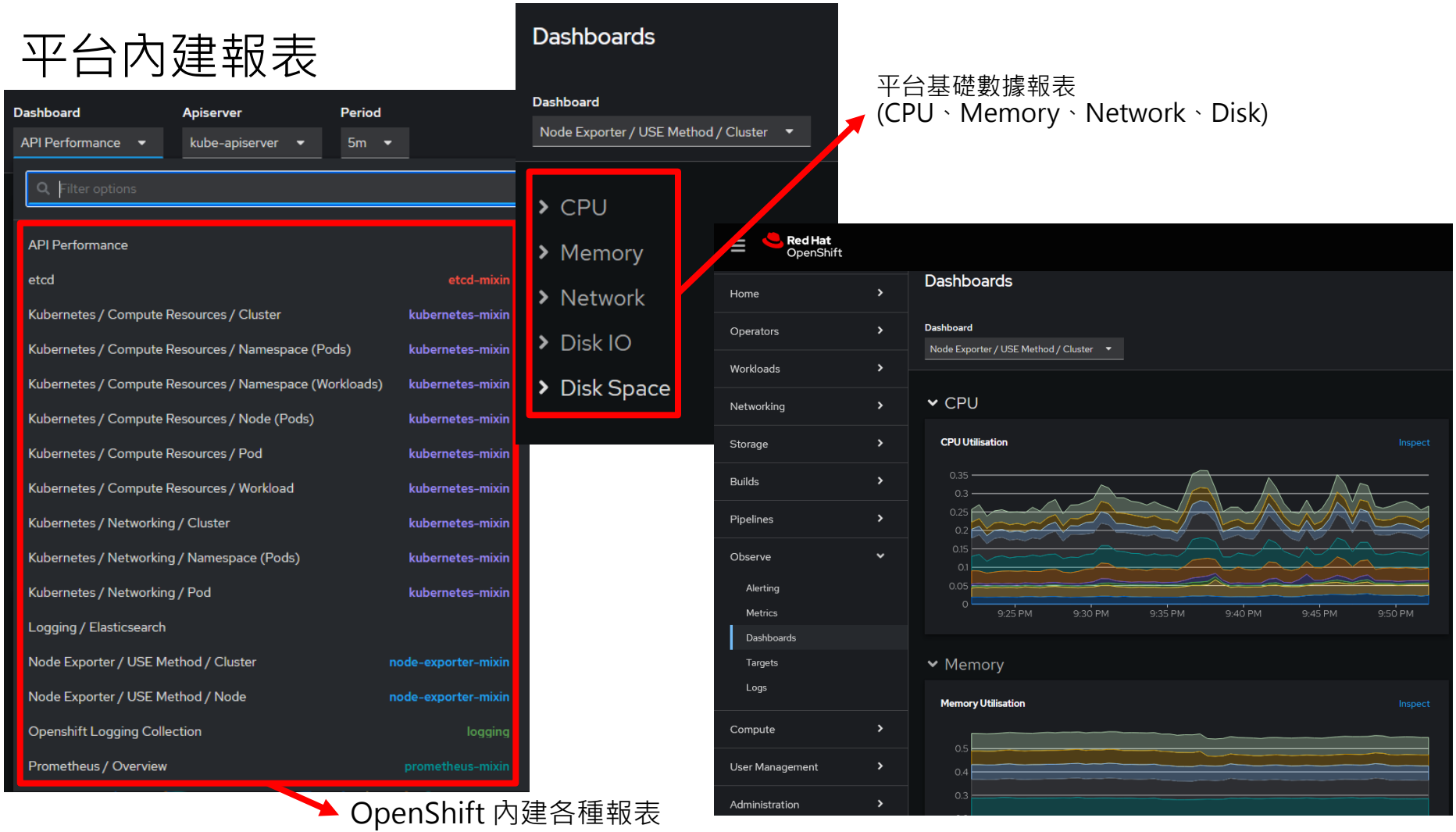
Source: https://docs.openshift.com/container-platform/4.6/monitoring/understanding-the-monitoring-stack.html
Source:https://www.openshift.com/blog/openshift-4-3-alertmanager-configuration

平台內建告警規則



OpenShift 內建 200 多條告警規則，包含
CPU、Memory、Disk、平台主機故障等等

平台內建報表



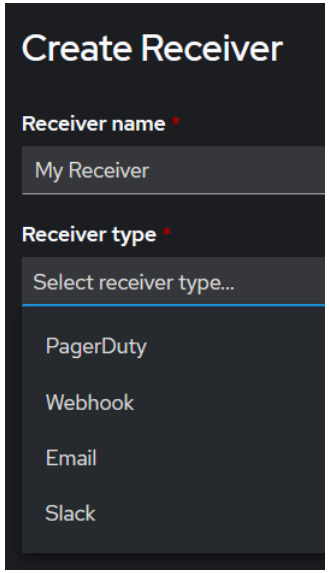
平台基礎數據報表
(CPU、Memory、Network、Disk)

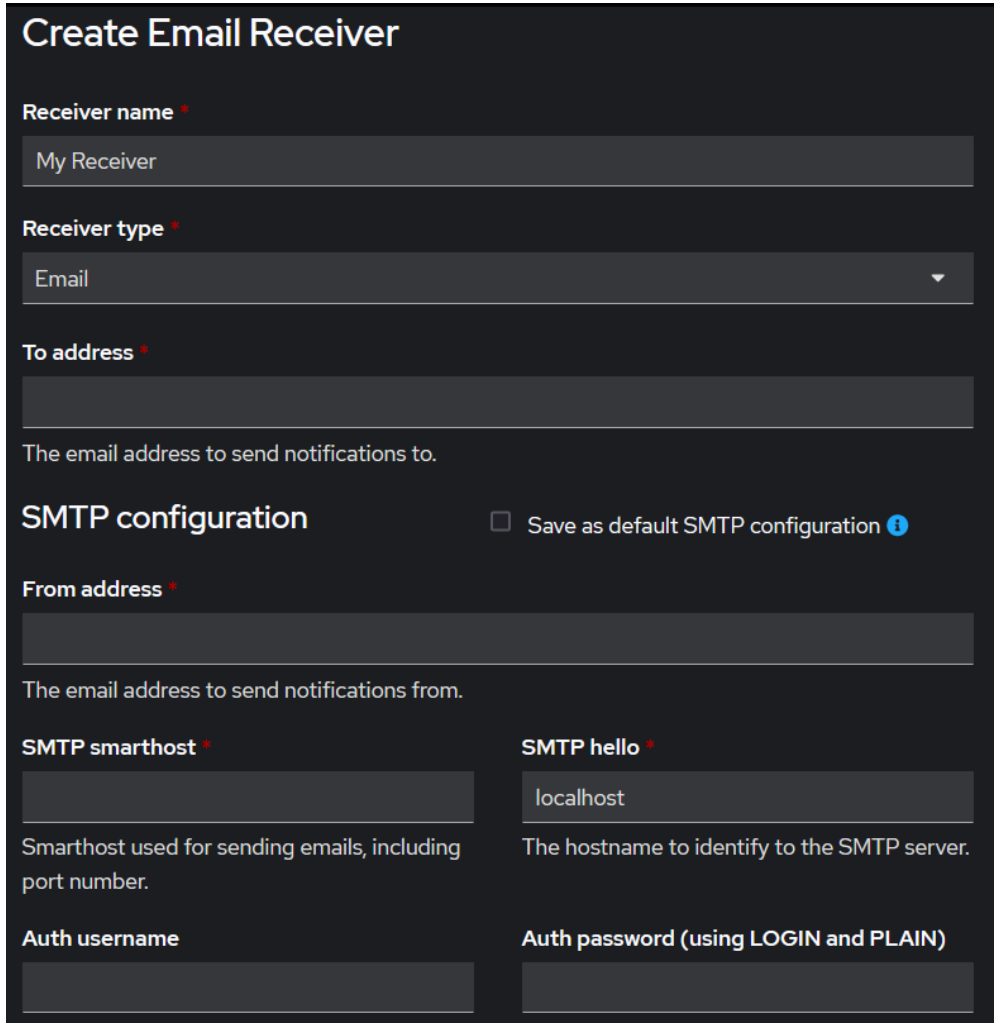
OpenShift 內建各種報表

平台告警通知

支援多種告警通知機制

- Email
- Webhook
- PageDuty
- Slack





Issue Overview

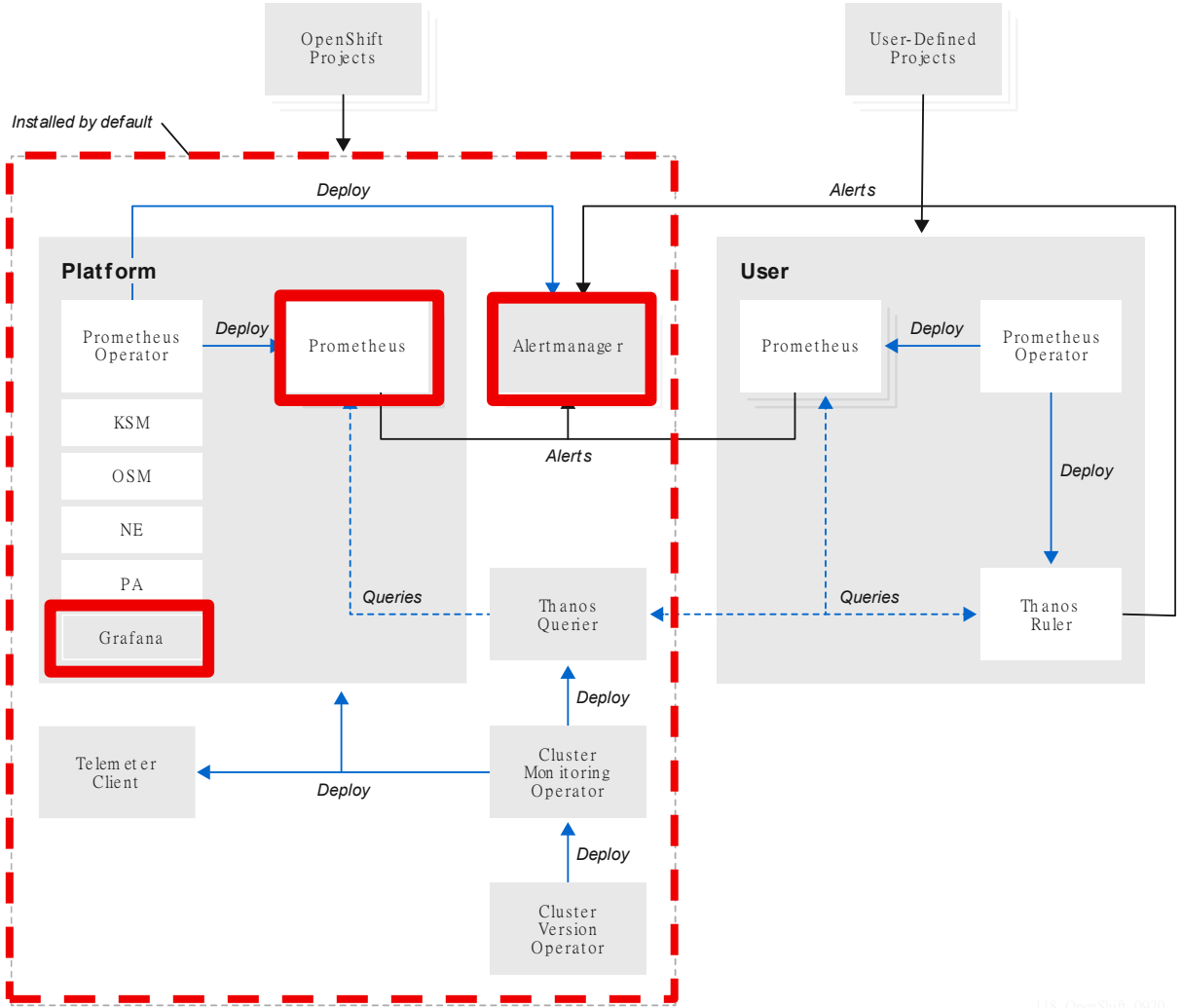
Alerting

- Mail Alert 寄送名單
- Webhook to MS Teams
  - https://prometheus.io/docs/alerting/latest/configuration/#webhook_config

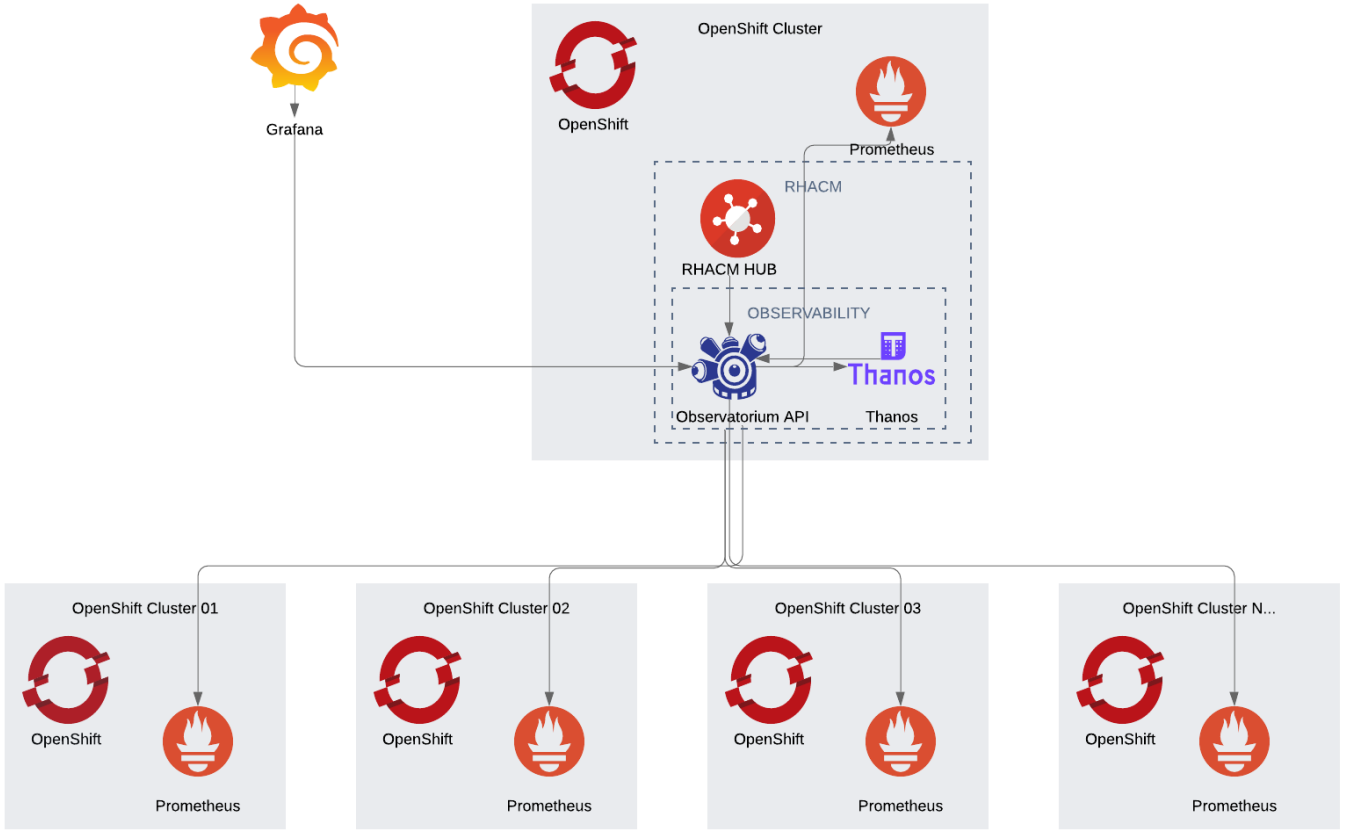
- 格式轉換程式 (Community)
  - https://github.com/prometheus-msteams/prometheus-msteams

The Alertmanager will send HTTP POST requests in the following JSON format to the configured endpoint:

```
{
  "version": "4",
  "groupKey": <string>,              // key identifying the group of alerts (e.g. to deduplicate)
  "truncatedAlerts": <int>,          // how many alerts have been truncated due to "max_alerts"
  "status": "<resolved|firing>",
  "receiver": <string>,
  "groupLabels": <object>,
  "commonLabels": <object>,
  "commonAnnotations": <object>,
  "externalURL": <string>,           // backlink to the Alertmanager.
  "alerts": [
    {
      "status": "<resolved|firing>",
      "labels": <object>,
      "annotations": <object>,
      "startsAt": "<rfc3339>",
      "endsAt": "<rfc3339>",
      "generatorURL": <string>,      // identifies the entity that caused the alert
      "fingerprint": <string>        // fingerprint to identify the alert
    },
    ...
  ]
}
```


OpenShift Projects
User-Defined Projects
Installed by default
Deploy
Alerts
Platform
Prometheus Operator
Deploy
Prometheus
Alertmanager
KSM
OSM
NE
PA
Grafana
Alerts
Queries
Thanos Querier
User
Prometheus
Deploy
Prometheus Operator
Deploy
Queries
Thanos Ruler
Deploy
Telemeter Client
Deploy
Cluster Monitoring Operator
Deploy
Cluster Version Operator

Multi-Cluster Monitoring (with RHACM)



Red Hat

Monitoring Sizing Considerations

Various tests were performed for different scale sizes. The Prometheus database grew, as reflected in the table below.

Number of Nodes	Number of Pods	Prometheus storage growth per day	Prometheus storage growth per 15 days	RAM Space (per scale size)	Network (per tsdb chunk)
50	1800	6.3 GB	94 GB	6 GB	16 MB
100	3600	13 GB	195 GB	10 GB	26 MB
150	5400	19 GB	283 GB	12 GB	36 MB
200	7200	25 GB	375 GB	14 GB	46 MB

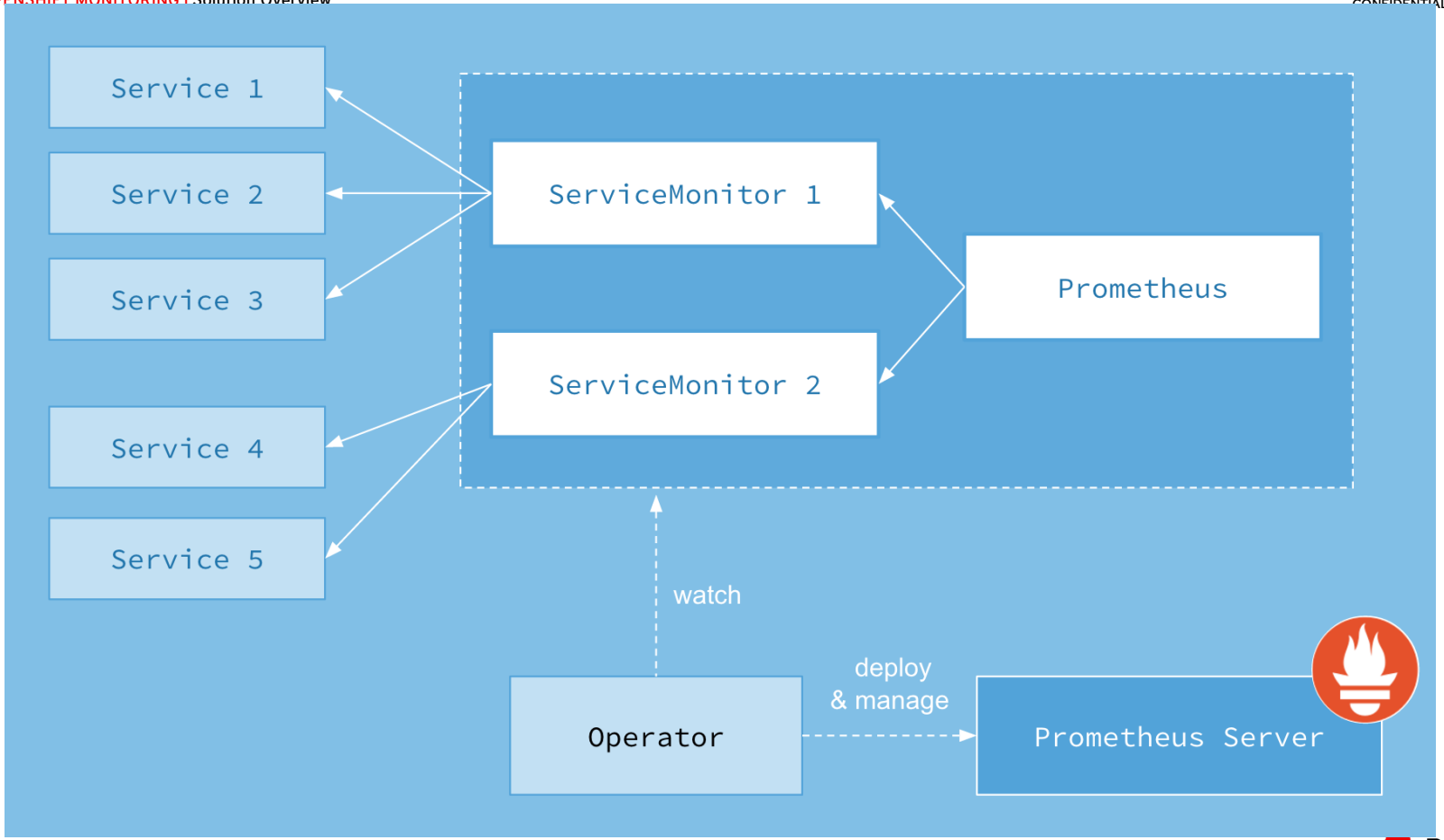
Custom Metric Autoscaler (Technology Preview)

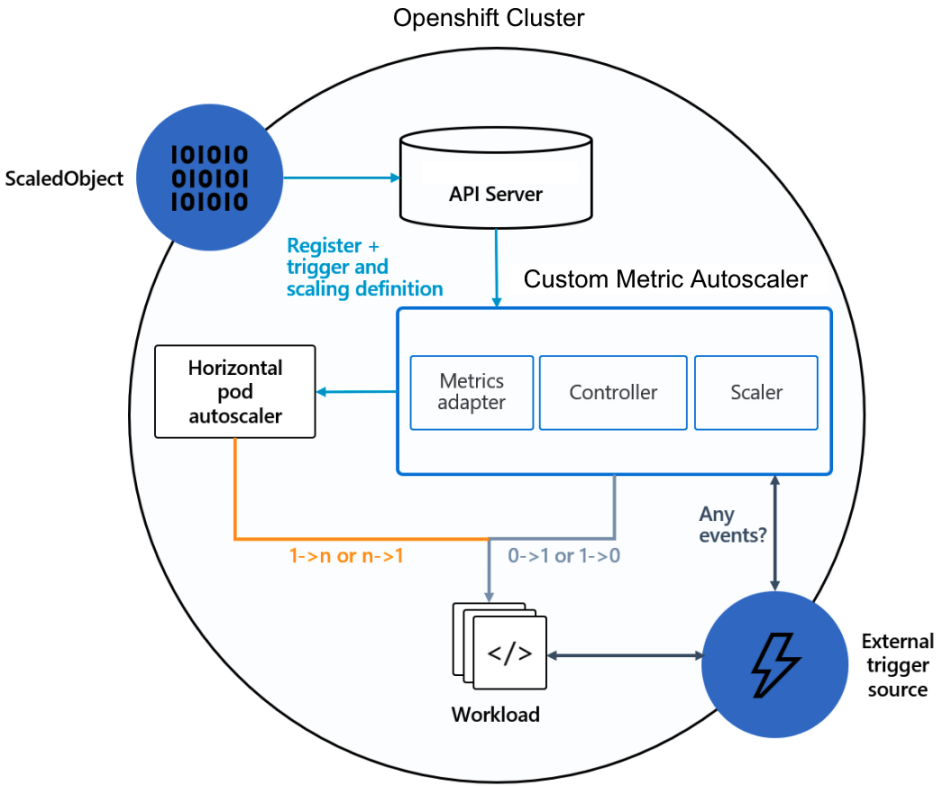
Scale workloads horizontally based on custom metrics

- Custom Metric Autoscaler is built on CNCF project KEDA
- Use Scalers example Prometheus , Apache Kafka and many more on which custom metric autoscaler can scale based on
- Manages workloads to scale to 0
- Registers itself as k8s Metric Adapter
- Provides metrics for Horizontal Pod Autoscaler (HPA) to scale on

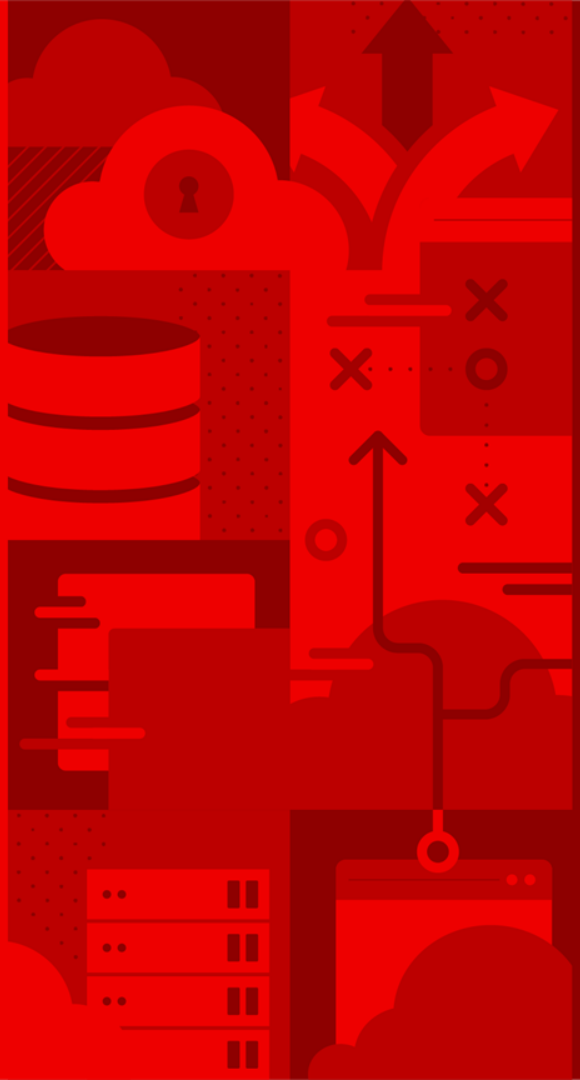
Product Manager: Gaurav Singh

```
[I have no name!@login-service-566d75fcfc-xc66t ~]$ curl -s localhost:9000
# HELP jmx_config_reload_failure_total Number of times configuration have failed to be reloaded.
# TYPE jmx_config_reload_failure_total counter
jmx_config_reload_failure_total 0.0
# HELP jmx_exporter_build_info A metric with a constant '1' value labeled with the version of the JMX exporter.
# TYPE jmx_exporter_build_info gauge
jmx_exporter_build_info{version="0.17.0",name="jmx_prometheus_javaagent",} 1.0
# HELP jvm_gc_collection_seconds Time spent in a given JVM garbage collector in seconds.
# TYPE jvm_gc_collection_seconds summary
jvm_gc_collection_seconds_count{gc="Copy",} 1838.0
jvm_gc_collection_seconds_sum{gc="Copy",} 22.31
jvm_gc_collection_seconds_count{gc="MarkSweepCompact",} 9.0
jvm_gc_collection_seconds_sum{gc="MarkSweepCompact",} 1.221
# HELP jmx_config_reload_success_total Number of times configuration have successfully been reloaded.
# TYPE jmx_config_reload_success_total counter
jmx_config_reload_success_total 0.0
# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 673.16
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.654628506147E9
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds 49.0
```





https://cloud.redhat.com/blog/custom-metrics-autoscaler-on-openshift
https://docs.openshift.com/container-platform/4.11/nodes/cma/nodes-cma-autoscaling-custom.html



Needs Attention

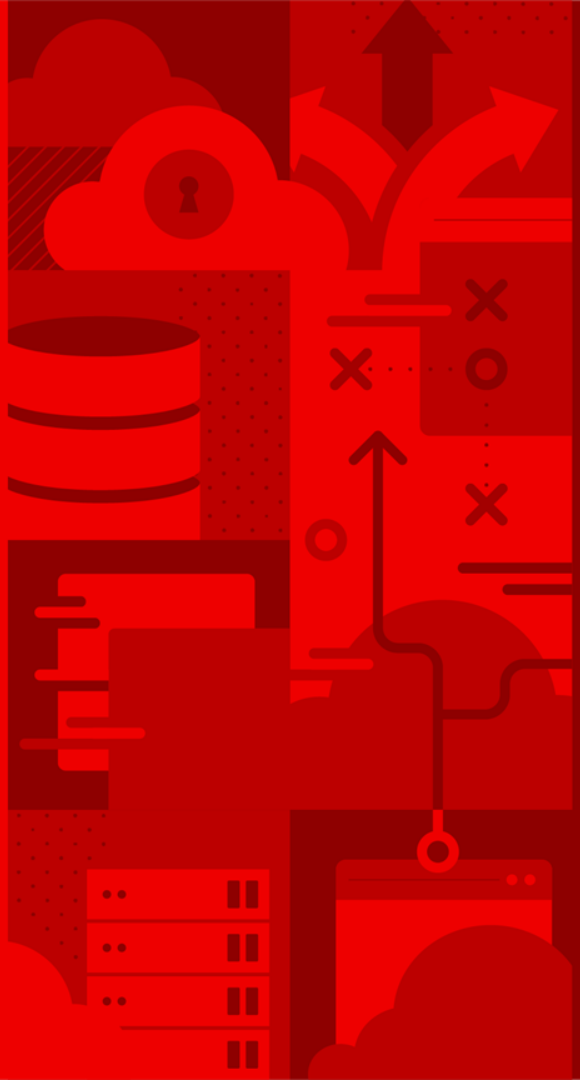
Red Hat

Needs Attention

- CSI use NetApp Astra Trident
  - iSCSI / NFS / S3
- Load Balancer
  - **採用 HA Proxy 軟體的方式 (Active-Standby)**
- Monitoring & Alerting
  - Webhook to MS Teams
- 帳號認證
  - **採用 OAuth (Azure AD)**
  - 備案: LDAP (本地AD)
- Azure DevOps 介接
  - **採用 Service Account Token 提供給 Azure DevOps 認證**
  - **需打通 ADO agent 至 OpenShift API Server** (https://api.sit.etmall.ocp:6443)

Needs Attention

- **Infra Node 不獨立，與 Worker Node 共用**
  - **OpenShift Logging 採用 Loki Stack + Vector 建置**
  - OpenShift Log Forwarder 將 AP Log 導向客戶自行維運之EFK，**僅保留 Infra 以及 Audit Log**
  - UAT 若採用 EFK，需多 1台 VM 作為 Infra Node (IBM 協助東森建置於OCP中)
- Routine Task
  - Who will monitor cluster?
  - Who will receive Mail Alert?
- Documentation
  - **Design Document**
  - **OpenShift Installation Steps**



Development

Container Tools

Red Hat

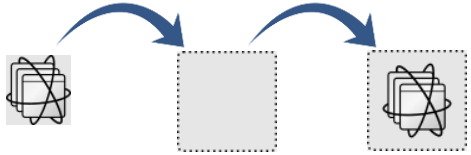
CONTAINERS TOOLS

Providing stability, flexibility and performance with containers and images

- Conform to the OCI image and runtime specifications
- Create, run, and manage, Linux Containers with an enterprise life cycle.
- Daemon-less
- Rootless capable
- OS-native container tooling
- Separation of concerns
- Part of Red Hat Enterprise Linux.
  Available, fully supported at no additional costs
- Check spare slides for labs shortcut

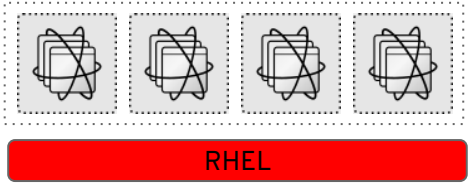
```
#yum module install container-tools
```

buildah



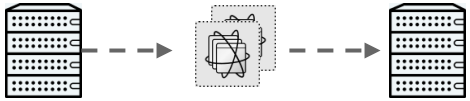
Build OCI/docker Images

podman



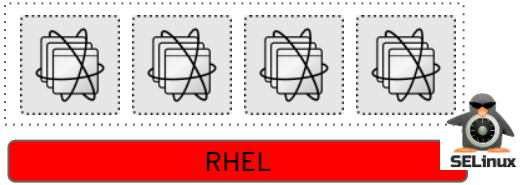
run, manage, debug containers

skopeo



Inspect, copy, & sign Images

udica



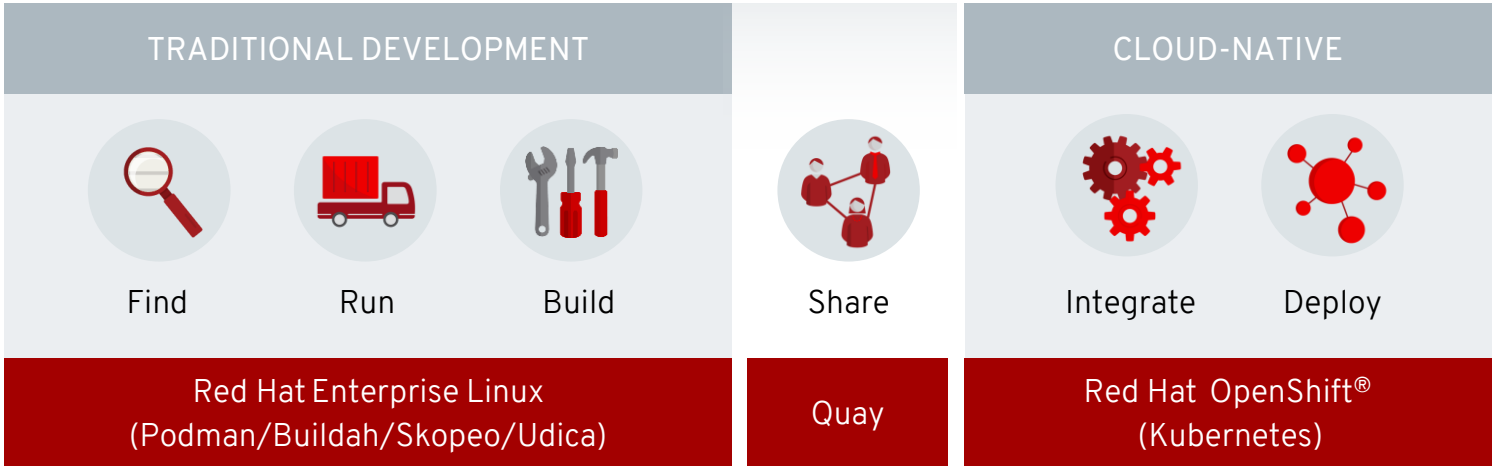
SELinux policy for containers made easy

OPEN CONTAINER INITIATIVE

Red Hat

The Journey

It can start everywhere

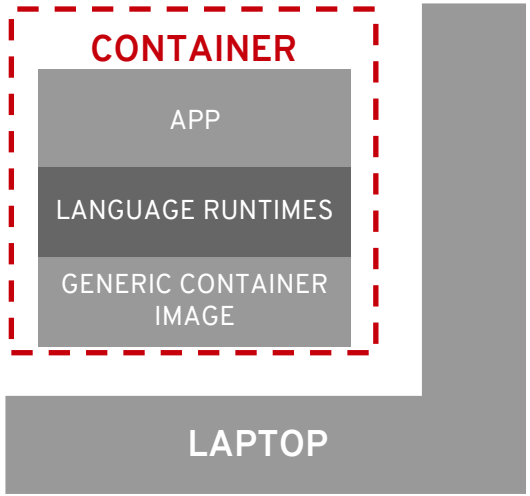


Red Hat

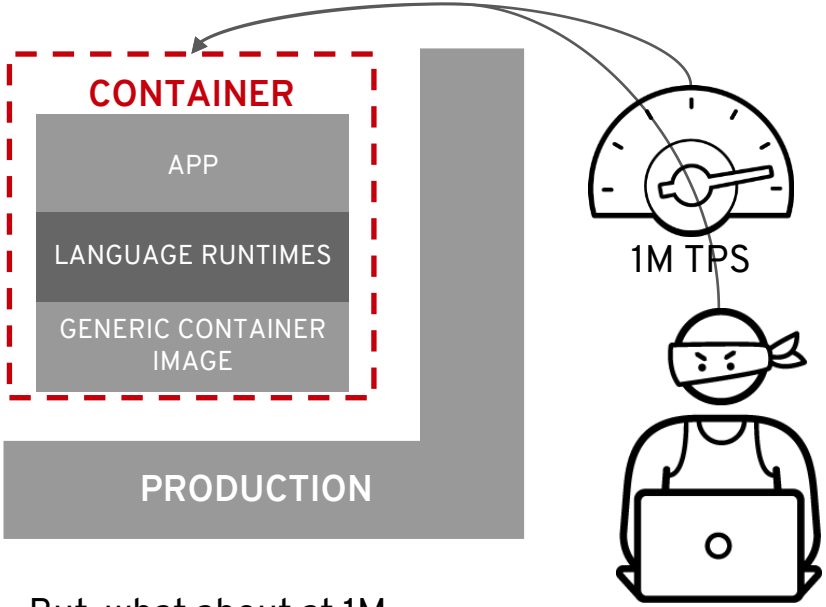
Red Hat Universal Base Image

Red Hat

Red Hat Universal Base Image



Works on my laptop



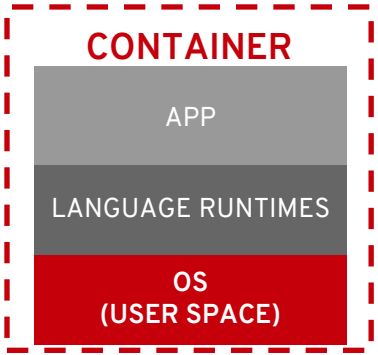
But, what about at 1M transactions per second & security ?

Red Hat

Red Hat Universal Base Image

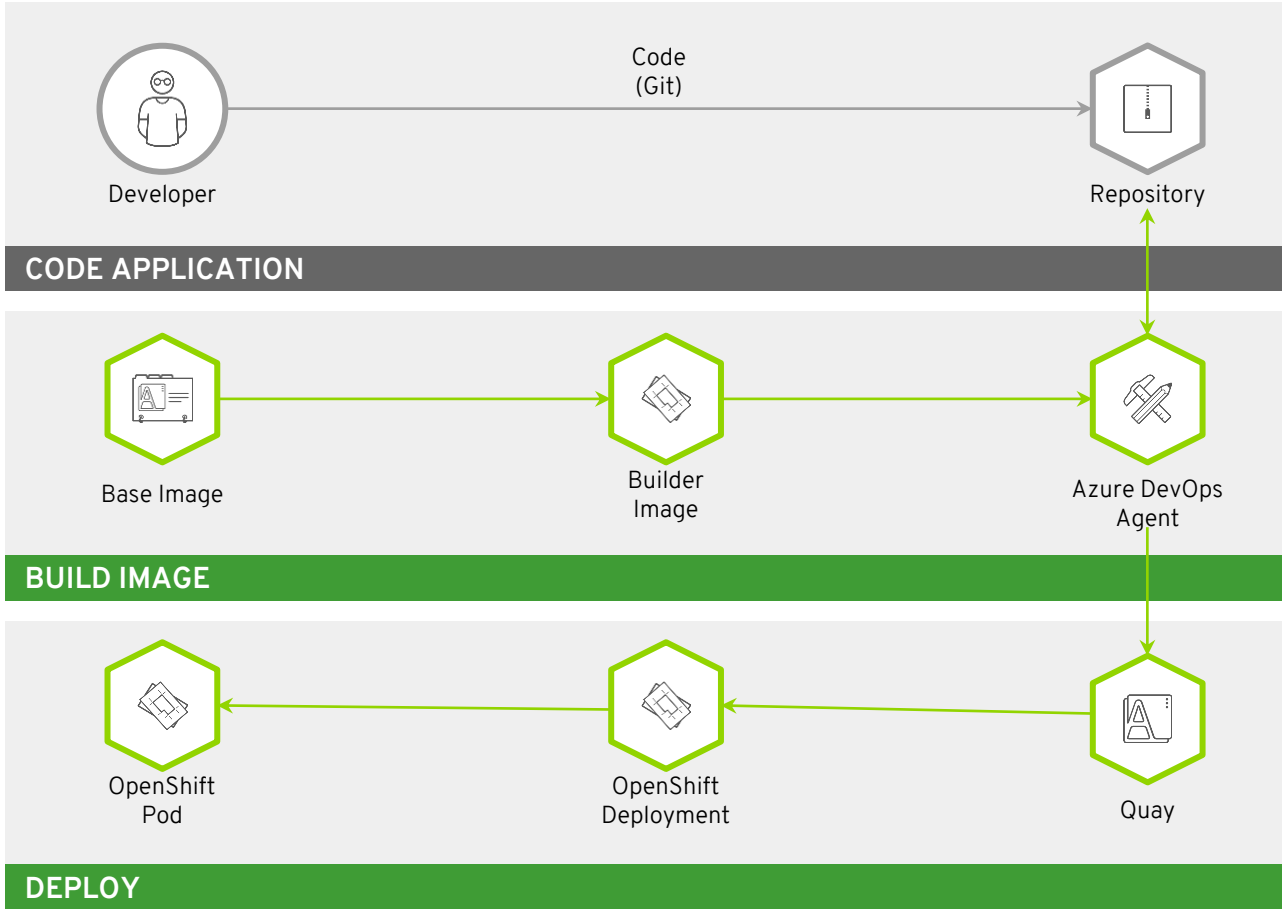
The image for all your needs

- Based on RHEL binaries
- Made available at no charge by a new end user license agreement.
- Development
  - Minimal footprint (~90 to ~200MB)
  - Programming languages (Modularity & AppStreams)
  - Enables a single CI/CD chain
- Production
  - Supported as RHEL when running on RHEL
  - Same Performance, Security & Life cycle as RHEL
  - Can attach RHEL support subscriptions as RHEL
- 3 flavors : Minimal, std, Init



- Standardize Your own deps os image, certified & compliant ST
- freely distribute to your teams, partners and contractor that run image on the OS of their choice
- Once app built, and shipped as container in your Red Hat env:
  - it is ST compliant
  - your get full support back

Red Hat


Developer
Code
(Git)
Repository
CODE APPLICATION
Base Image
Builder
Image
Azure DevOps
Agent
BUILD IMAGE
OpenShift
Pod
OpenShift
Deployment
Quay
DEPLOY

Red Hat Quay

The enterprise-grade image registry

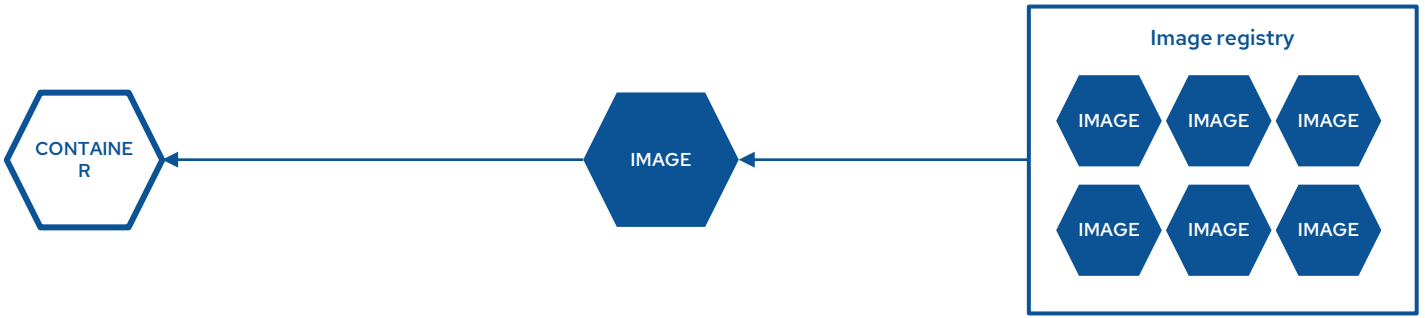
Source of truth for you container images distribution

Red Hat

A container is the smallest compute unit.

Containers are created from container images.

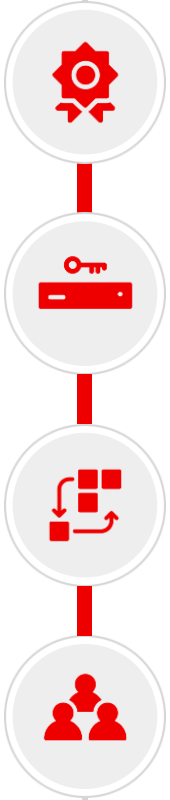
Container images are stored in an image registry.



If you're doing containers and Kubernetes, you need a container image registry.

Red Hat Quay goes far beyond that.

Red Hat Quay



Industry-leading, **trusted, and open source registry platform** operating at scale since 2014

Built to **efficiently manage content** under governance and security **controls** globally

Runs **everywhere**, easy to **integrate** and **automate** but works best with **OpenShift**

Developed in **collaboration** with a broad open source, customer, and ecosystem **community**

One product, flexible purchasing models

Enterprise-grade
container registry

Full-featured enterprise product
Runs both on-prem or on public cloud
Deployed and operated via Kubernetes operators
Flexible configuration options
Maintained by the customer
Premium Red Hat Support

SELF-MANAGED

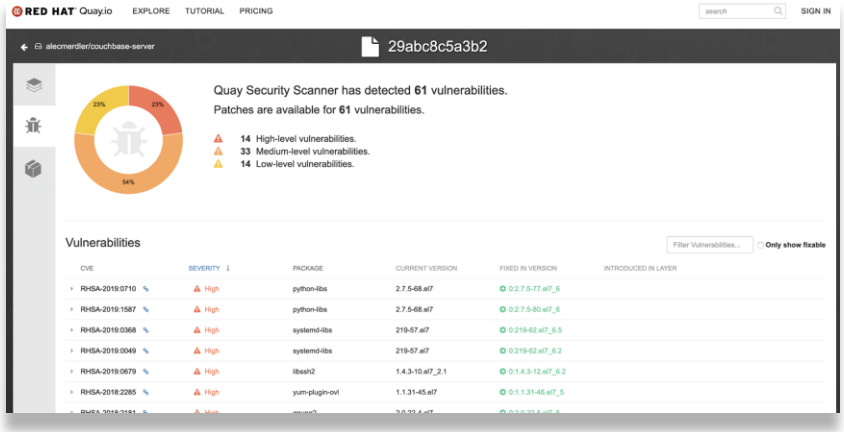
Hosted, multi-tenant
SaaS offering

Free unlimited public repositories
Private repositories available via plans
Self-service access
Monthly credit card payment
Maintained by Red Hat
Support included in paid plans

HOSTED SERVICE

Clair Overview

- Clair is an open source tool for static analysis of vulnerabilities in application containers
- Developed by CoreOS for Quay and it’ s massive scale usage at Quay.io
- Used by various other projects and third party products
- Upstream Repositories: https://github.com/quay/clair



Red Hat

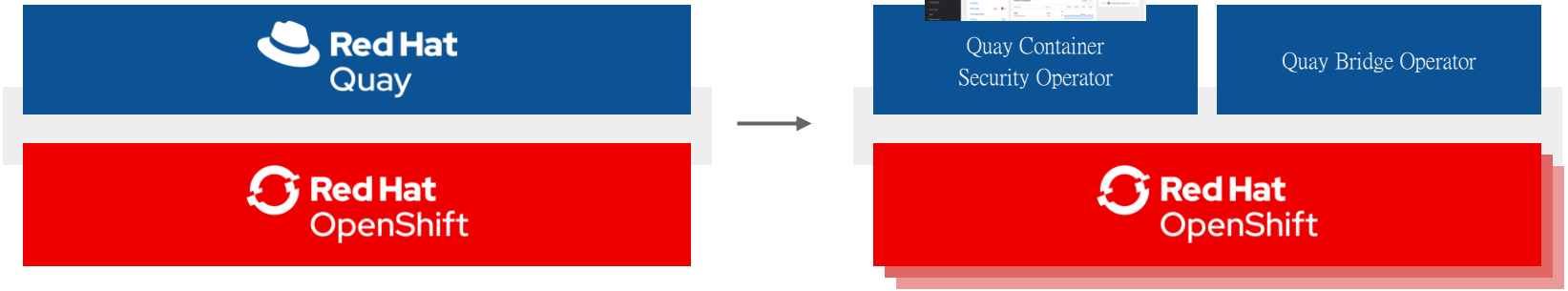
Red Hat Quay works best with OpenShift

Red Hat Quay runs on any infrastructure but **runs best on OpenShift**

The **Quay Operator** ensures seamless deployment and management of Quay running on OpenShift

CSO brings Quay / Clair vulnerability data into the OpenShift Console

The **Quay Bridge Operator** ensures seamless integration and user experience for using Quay **with** OpenShift

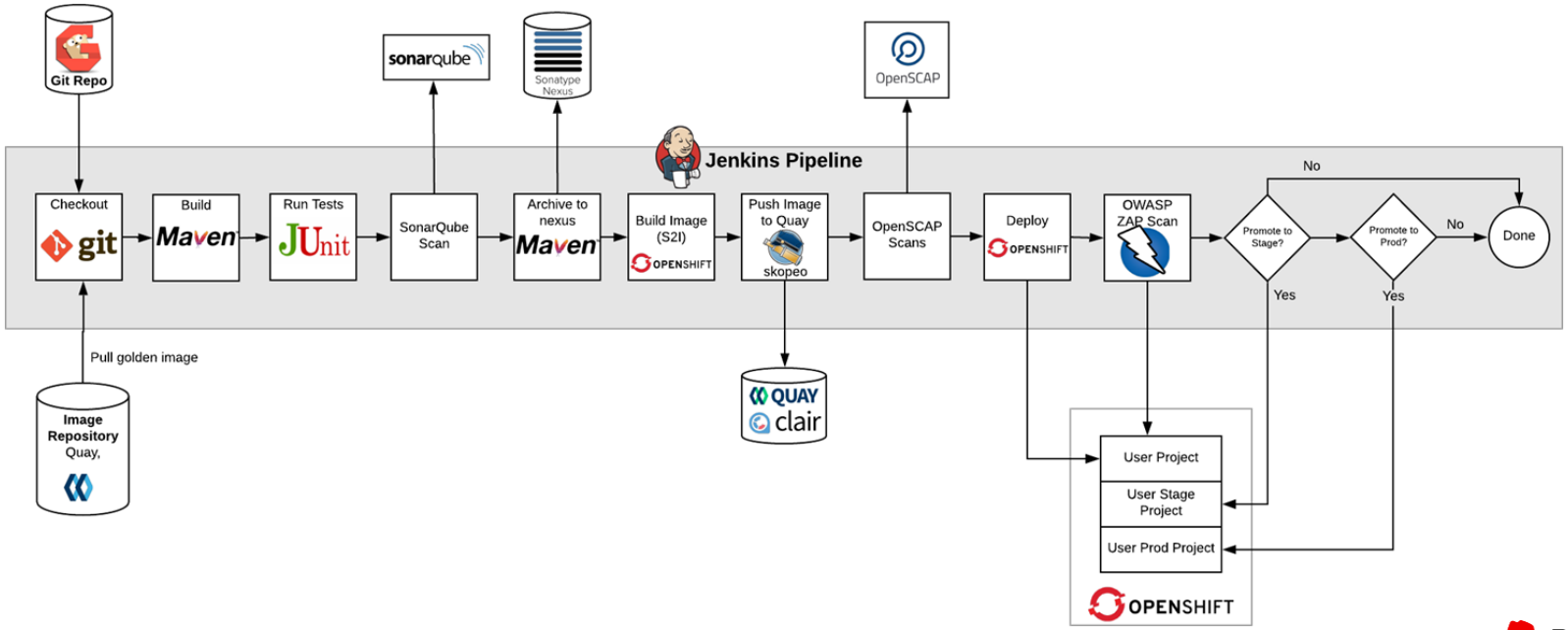


Quay serves content to **one or many OpenShift clusters**, wherever they're running.

With or without using the OpenShift internal registry but leveraging all OpenShift capabilities.

Quay and OpenShift - Pipeline Integration

Quay and Clair can be integrated into existing CI/CD pipelines - exemple



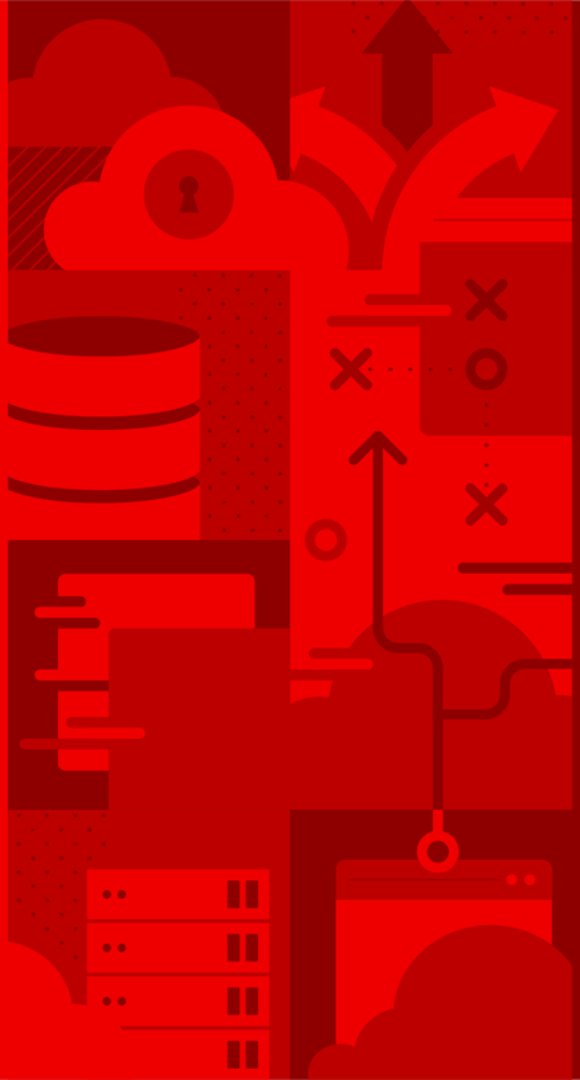
Development

- Image
  - **Image Registry (Quay)**
    - UAT
    - SIT
  - Image Scan
    - 網路白名單
    - CVE 資訊下載
  - Image Layer
    - Worker 會使用更多空間，用以暫時存放 Image
    - 建議採用，共用 Library的 image 架構
  - 建議所有環境 (SIT/UAT/Prod)，皆採用同一份 Image

Development

- Deploy Application
  - Scale Number of Replicas
  - Modify Image Version
  - Modify ConfigMap / Secret

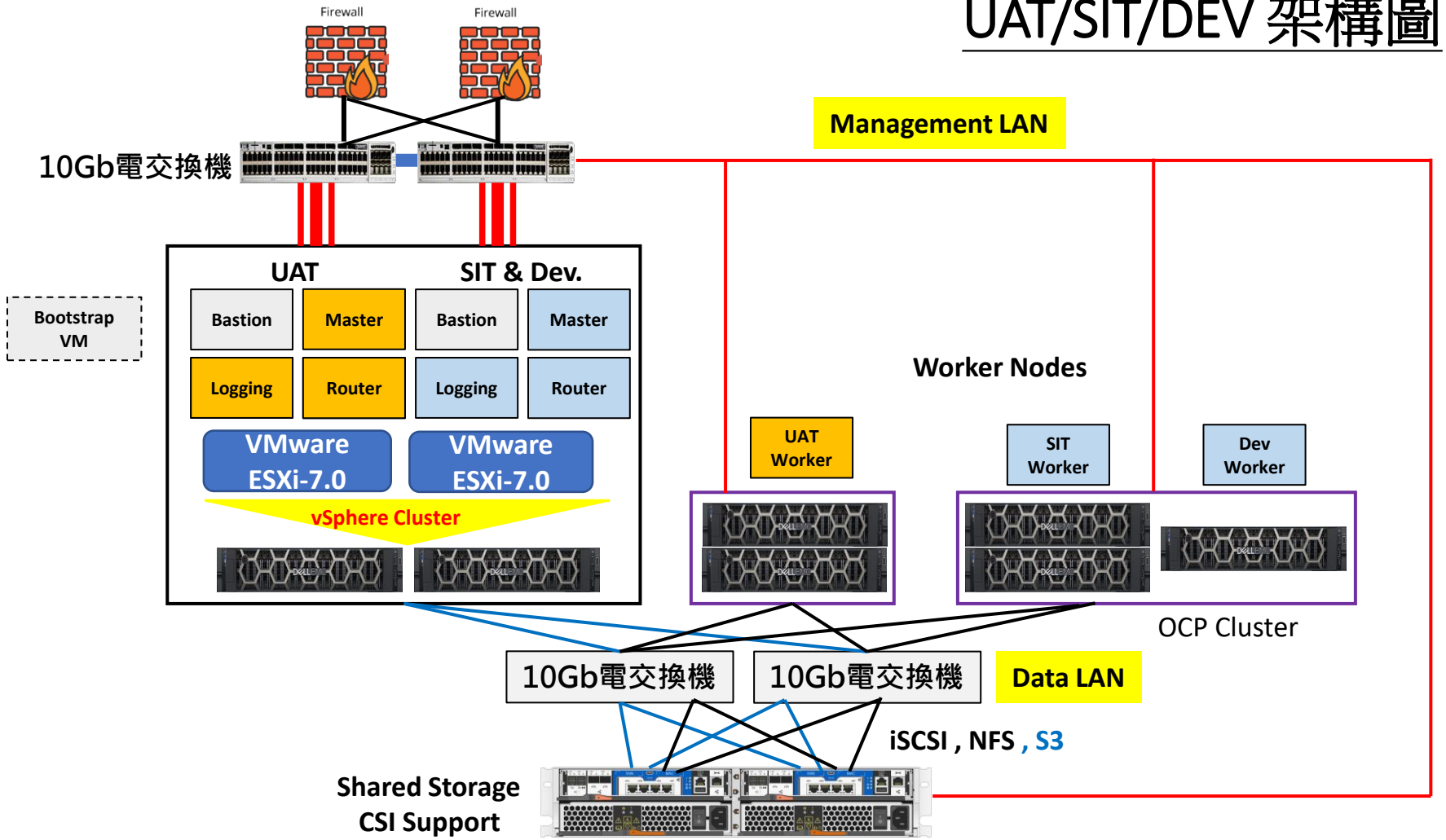
  - Create new Project
  - Restart Program (Delete Pod)



Resource Requirement with Infra Node

Red Hat

UAT/SIT/DEV 架構圖



OpenShift 資源 (SIT / Dev VM)

虛擬平台 (SIT/Dev)		OS	Nodes #	vCPU	Memory	Storage
Bastion 部署管理機	VM	RHEL 9.2	1	4	16	300
HAProxy (Load Balancer)	VM	RHEL 9.2	2	2	8	50
Bootstrap	VM	RHCOS	0	4	16	100
Master Nodes	VM	RHCOS	3	4	16	100
Infra nodes (Quay + ACS Sensor)	VM	RHCOS	2	10	24	100
Router Nodes	VM	RHCOS	2	2	8	100
Log Node (ELK) Node	VM	RHCOS	3	4	32	100

OpenShift 資源 (UAT VM)

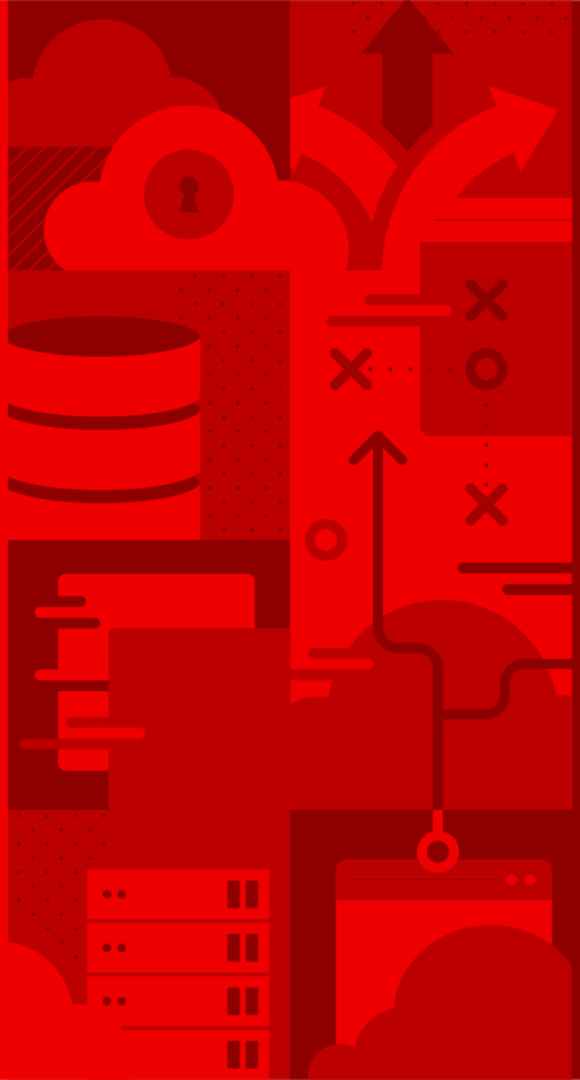
虛擬平台 (UAT)		OS	Nodes #	vCPU	Memory	Storage
Bastion 部署管理機	VM	RHEL 9.2	1	4	16	300
HAProxy (Load Balancer)	VM	RHEL 9.2	2	2	8	50
Bootstrap	VM	RHCOS	0	4	16	100
Master Nodes	VM	RHCOS	3	4	16	100
Infra nodes (Quay + ACM + ACS)	VM	RHCOS	2	32	70	100
Router Nodes	VM	RHCOS	2	2	8	100
Log Node (ELK) Node	VM	RHCOS	3	4	32	100

OpenShift 資源 (Pods)

OCP 額外使用 (已記入 Infra Node)		Cluster	Nodes #	vCPU	Memory	Storage
EFK	Pod	UAT + SIT	6	2	16	300
Prometheus	Pod	UAT + SIT	4	1	8	100
ACM	Pod	UAT	1	6	12	
ACS Central	Pod	UAT	1	4	8	
ACS Central DB	Pod	UAT	1	8	16	100
ACS Scanner	Pod	UAT	1	2	4	
ACS Scanner DB	Pod	UAT	1	2	4	100
ACS Sensor	Pod	UAT + SIT	2	4	8	
ACS Adminssion Controller	Pod	UAT + SIT	2	0.5	0.5	
ACS Collector	Pod	UAT + SIT	2	2.75	5	
Quay	Pod	UAT + SIT	2	2	8	TBD

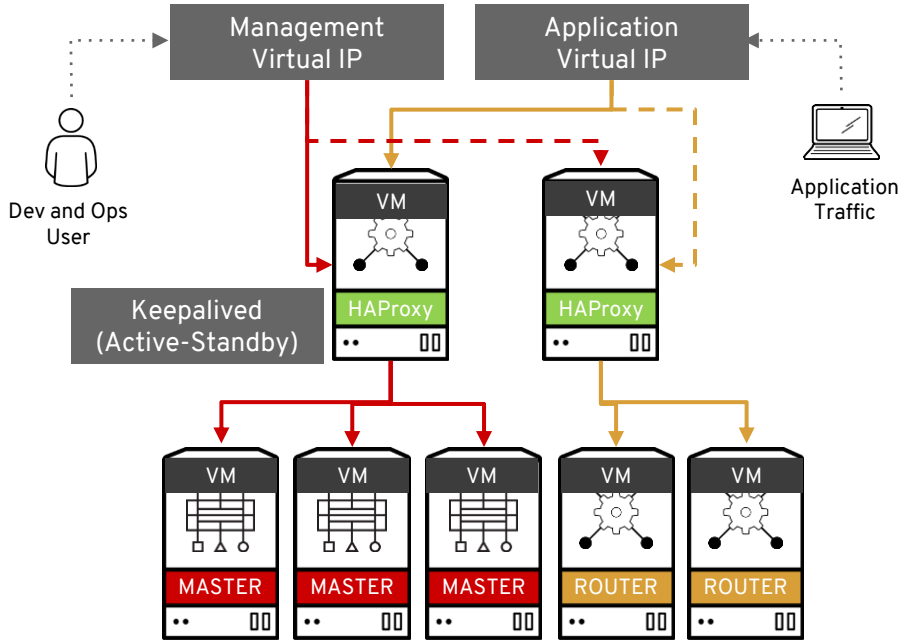
OpenShift 資源 (Other)

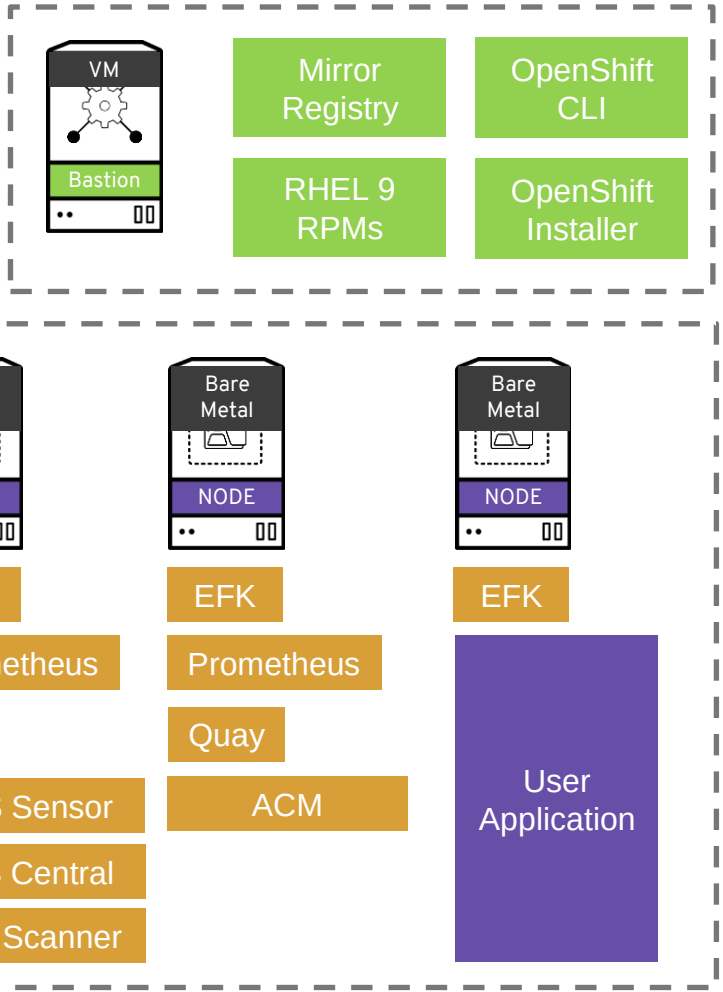
虛擬平台 (Other)		OS	Nodes #	vCPU	Memory	Storage
vCenter	VM	x	1	x	x	x
實體機 (Worker)		OS	Nodes #	vCPU	Memory	Storage
UAT Worker	Bare Metal	RHCOS	2	128	256	300+
SIT / Dev Worker	Bare Metal	RHCOS	3	128	256	300+
VM Host	Bare Metal	ESXi 7.0	2	96	256	x



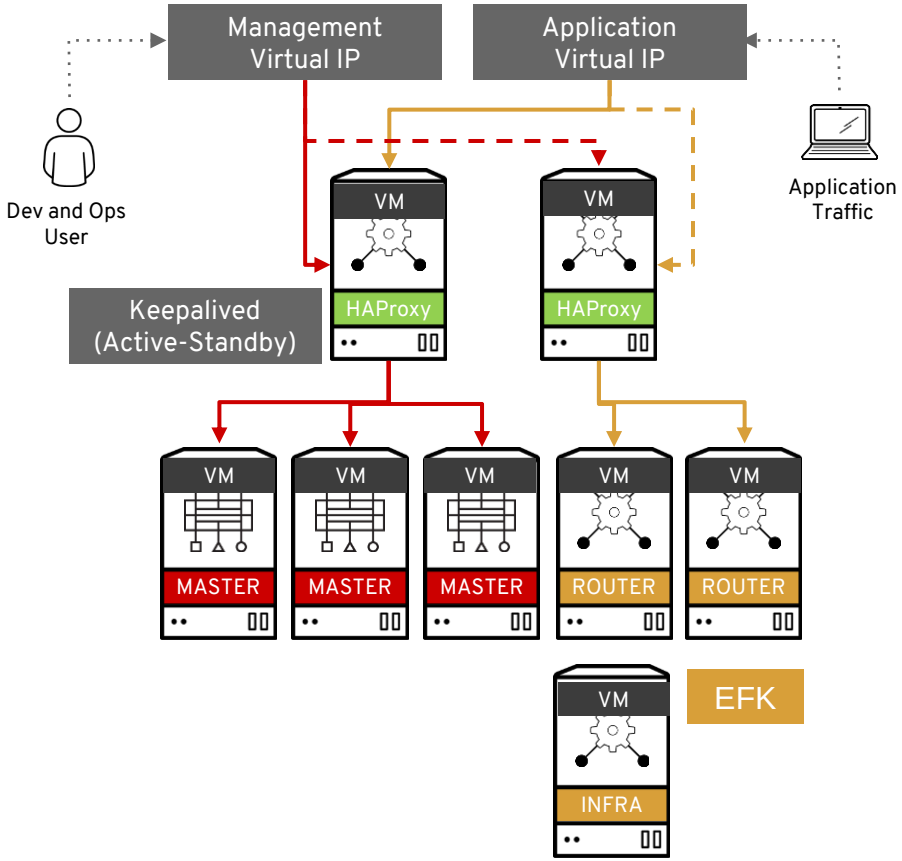
Resource Requirement without Infra Node

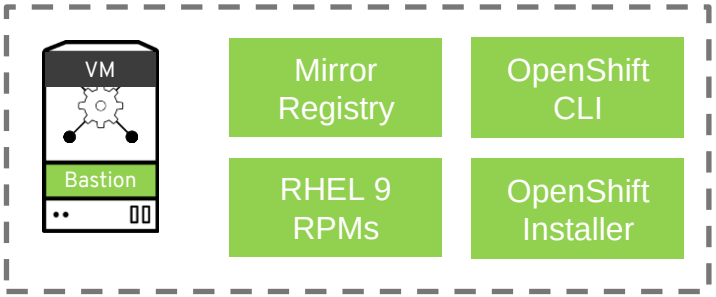
OpenShift 元件 (SIT/Dev)

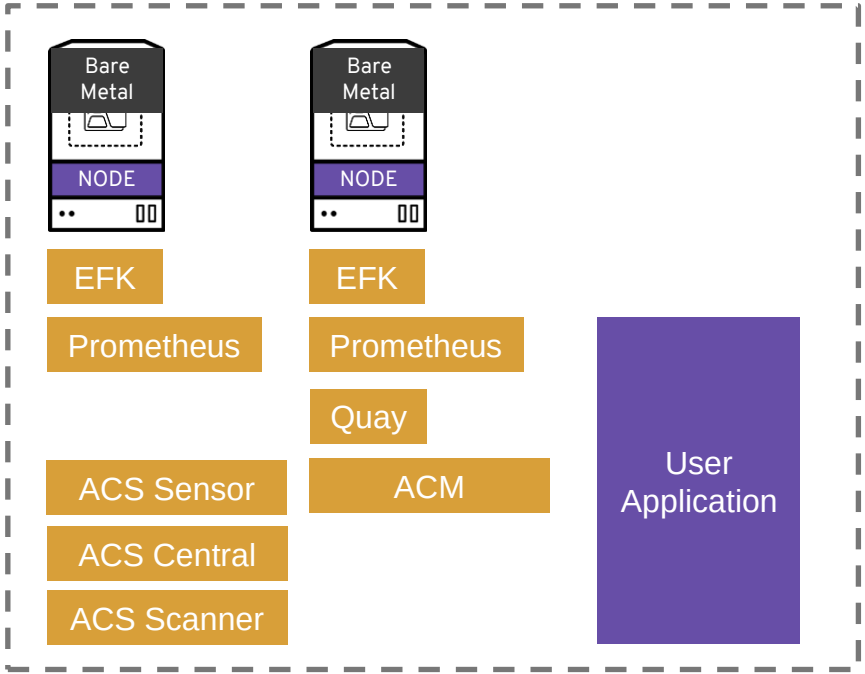




OpenShift 元件 (UAT)







OpenShift 資源 (SIT / Dev VM)

虛擬平台 (SIT/Dev)		OS	Nodes #	vCPU	Memory	Storage
Bastion 部署管理機	VM	RHEL 9.2	1	4	16	300
HAProxy (Load Balancer)	VM	RHEL 9.2	2	2	8	50
Bootstrap	VM	RHCOS	0	4	16	100
Master Nodes	VM	RHCOS	3	4	16	100
Router Nodes	VM	RHCOS	2	2	8	100
Worker node占用 (Quay + ACS Sensor + EFK)	x	x	x	22	144	x

OpenShift 資源 (UAT VM)

虛擬平台 (UAT)		OS	Nodes #	vCPU	Memory	Storage
Bastion 部署管理機	VM	RHEL 9.2	1	4	16	300
HAProxy (Load Balancer)	VM	RHEL 9.2	2	2	8	50
Bootstrap	VM	RHCOS	0	4	16	100
Master Nodes	VM	RHCOS	3	4	16	100
Router Nodes	VM	RHCOS	2	2	8	100
Log Node (EFK) Node	VM	RHCOS	1	4	24	100
Worker node占用 (Quay + ACM + ACS + EFK)	x	x	x	40	134	x

OpenShift 資源 (Pods)

OCP 額外使用 (已記入 Infra Node)		Cluster	Nodes #	vCPU	Memory	Storage	Type
EFK	Pod	UAT + SIT	6	2	16	300	iSCSI
Loki Stack	Pod	UAT + SIT	4	36	63	500	S3
Prometheus	Pod	UAT + SIT	4	1	8	100	iSCSI
ACM	Pod	UAT	1	6	12	x	x
ACS Central	Pod	UAT	1	4	8	x	x
ACS Central DB	Pod	UAT	1	8	16	100	iSCSI
ACS Scanner	Pod	UAT	1	2	4	x	x
ACS Scanner DB	Pod	UAT	1	2	4	100	iSCSI
ACS Sensor	Pod	UAT + SIT	2	4	8	x	x
ACS Adminssion Controller	Pod	UAT + SIT	2	0.5	0.5	x	x
ACS Collector	Pod	UAT + SIT	2	2.75	5	x	x
Quay	Pod	UAT + SIT	2	2	8	500	S3

OpenShift 資源 (Other)

虛擬平台 (Other)		OS	Nodes #	vCPU	Memory	Storage
vCenter	VM	x	1	x	x	x
實體機 (Worker)		OS	Nodes #	vCPU	Memory	Storage
UAT Worker	Bare Metal	RHCOS	2	128	256	300+
SIT / Dev Worker	Bare Metal	RHCOS	3	128	256	300+
VM Host	Bare Metal	ESXi 7.0	2	96	256	x

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.

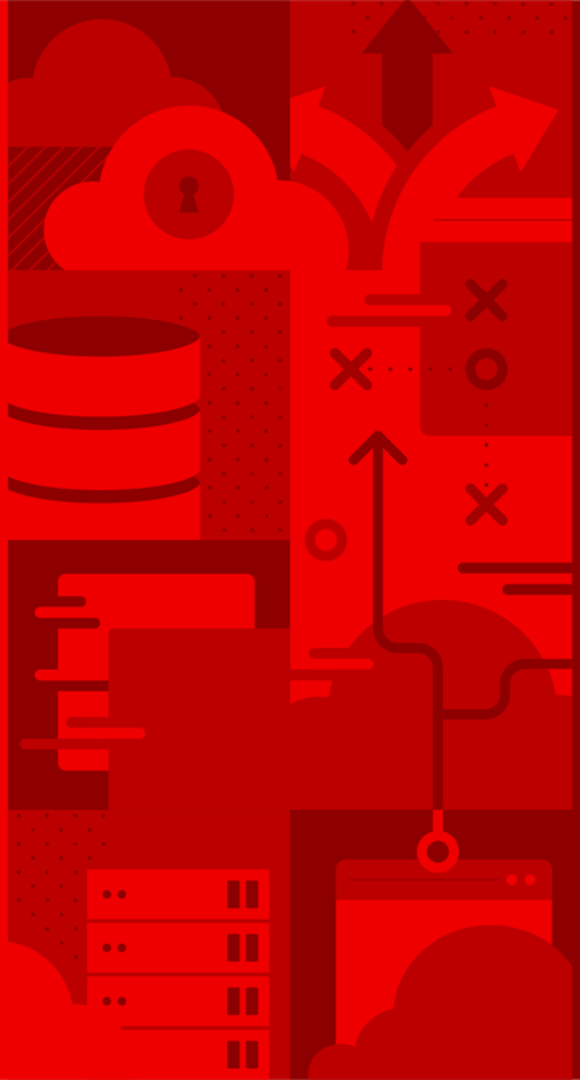

linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat



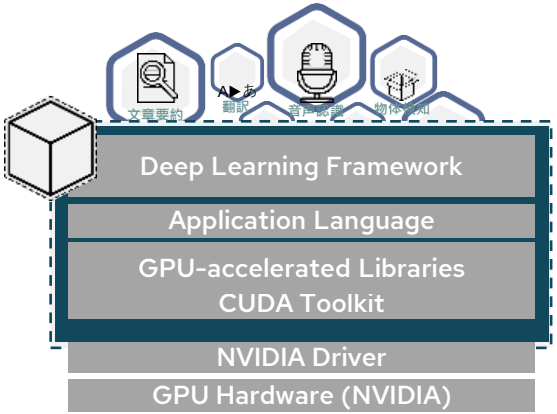
NVIDIA GPU Operator

Red Hat

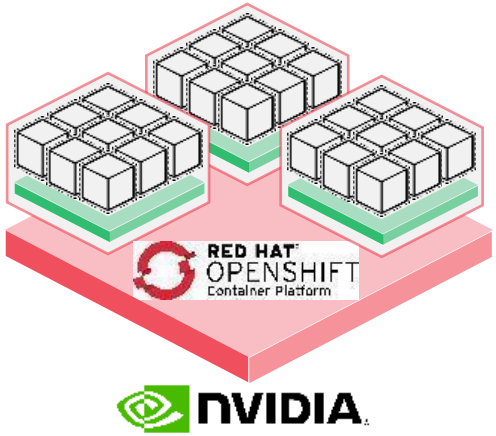
Benefits of running GPU workloads on Container Platform

- Adopting GPU accelerated hardware requires the configuration of multiple software components i.e. drivers, container runtimes, other libraries that are difficult and prone to errors. [1]
- The layered structure of container images isolate dependencies between host OS libraries the NVIDIA driver, and are highly compatible with change management and continuous integration (CI) processes.

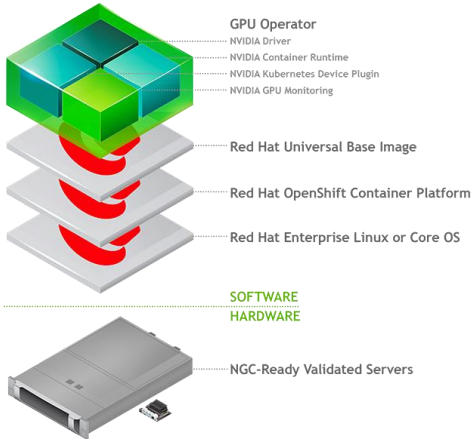
Virtualization Platform



Container Platform (Kubernetes, OpenShift)



NVIDIA GPU Driver



- Easier to change versions of applications and dependent libraries in containers.
- Highly compatible with CI through continuous build and automated testing.
- NVIDIA GPU Driver allows the allocation of GPU resources at the container level.

[1] https://cloud.redhat.com/blog/business-centric-ai/ml-with-kubernetes-part-3-gpu-acceleration

Red Hat