

- ACM 有自有的subscription model



- imperative **直接式** 直接用cli建立
- declarative **宣告式** 透過yaml檔案建立 可做版控



PowerPoint 播影片放映 - 1.ACM_AzureDevOps_CICD_Design.pptx - PowerPoint

ETMall Application Life Cycle w/ multi-OpenShift Cluster

CONFIDENTIAL designator

GitOps概念基本概念-延伸的好處

傳統CICD工具執行方式-PUSH方式:

由CICD工具如Jenkins將封裝檔案/image推送往cluster
並透過ServiceAccount賦予的身分權限命令OCP/K8S進行Component佈署動作

CI/CD Server

APP

OPENSHIFT

Kubernetes

Red Hat

51

teams.microsoft.com 正在共用你的畫面。 停止共用 重新

V0000000

ETMall Application Life Cycle w/ multi-OpenShift Cluster

CONFIDENTIAL designator

GitOps概念基本概念-延伸的好處

新的CD工具執行方式-PULL方式:

由CD工具如ArgoCD或RHACM等透過agent於OCP/K8S內主動監控Git Server Repo是否變動
依據Git Server Repo的Desired state觸發Cluster進行Component異動
省去多暴露ServiceAccount權限給外部系統,減少資安風險

Desired state

Actual State

Pulls

From Git Repository

Agent

OPENSHIFT

Kubernetes

Argo

Ex:deployment yaml component

52

teams.microsoft.com 正在共用你的畫面。 停止共用 重新

V0000000

Red Hat

- push式 賦予權限給k8s
- pull式 k8s自己去取得權限

PowerPoint 投影片放映 - 1.ACM_AzureDevOps_CICD_Design.pptx - PowerPoint

ETMall Application Life Cycle w/ multi-OpenShift Cluster

CONFIDENTIAL, designator

GitOps概念基本概念

Benefit:

- 1.更容易Rollback:
 - 如上以Pull式的方式,Agent會隨時syncing changes
 - 持續地比對git Repository的內容是否有差異
 - 例如改動造成了環境問題,就可以透過git revert進行修復!
 - 讓環境退回(還原)到前一次可以運行的狀態!
2. Single source of Truth:
 - 即使IaC內容都在許多人各自本機上修改
 - 但是實際會造成異動的都是Git Repository上的那一份
 - 也就是實際環境僅參照到Git Repository的Code
 - 這也在管理上更加統一
- 3.增加安全性:
 - 各種人員不再能夠自行隨意且直接地異動環境(如個人筆電就連上去改)
 - 僅限定由CD Pipeline工具才能做出實際的異動
 - 但是人員將可透過Git Flow進行變更後最終觸發CD Pipeline
 - 而實際能透過Git Flow變更的人員或是過程覆核的人員還可以限制
 - 這將得到以下益處:
 - 1.Less Permissions to Manage(最小化管理)
 - 2.More Secure Environment(更高的環境安全性)

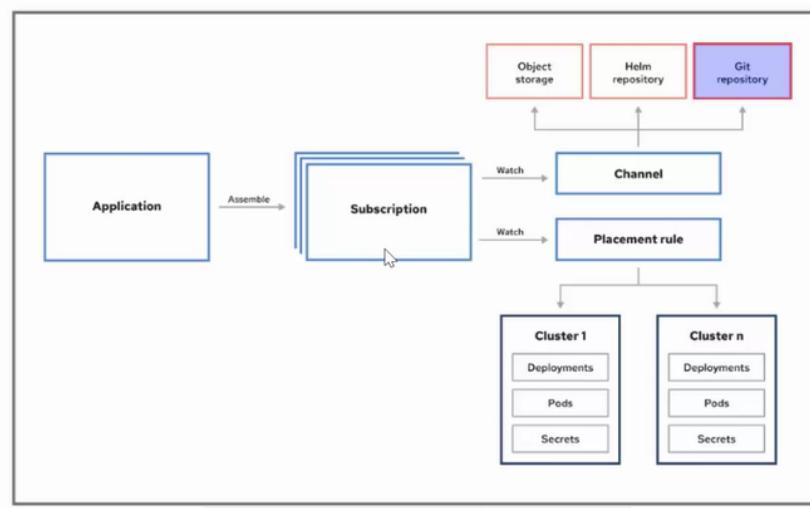
55 V0000000 Red Hat

- 須嚴格遵守single source of truth
- 透過gitops來達成
- ACM Application Subscription Model

ETMall Application Life Cycle w/ multi-OpenShift Cluster CONFIDENTIAL, designator

ACM實作GitOps的其中一種方式 Application Subscription Model

APPLICATION SUBSCRIPTION MODEL



57

V0000000 Red Hat

- 在創建時需要指定要sync的 repo 位置
- repo 帳號暴露給 k8s, 讓k8s可以去pull
- placement rule 紀錄要部屬到哪個cluster
- channel 紀錄要部屬到哪個namespace
- subscription 紀錄要部屬到哪個application

ETMall Application Life Cycle w/ multi-OpenShift Cluster

CONFIDENTIAL, designator

其中Channel裡面定義的manifests來源可以是:

- 1.Git Repository
- 2.Helm release registry
- 3.Object storage repository

Subscriptions可以透過channel來辨識資源的新增或異動

並且會直接佈署資源到目標也就是託管下的Cluster(叢集)中

而不用先檢查Hub Cluster(中心叢集,也就是安裝ACM的叢集)

並且會持續監視channel的資源變動

59

V0000000

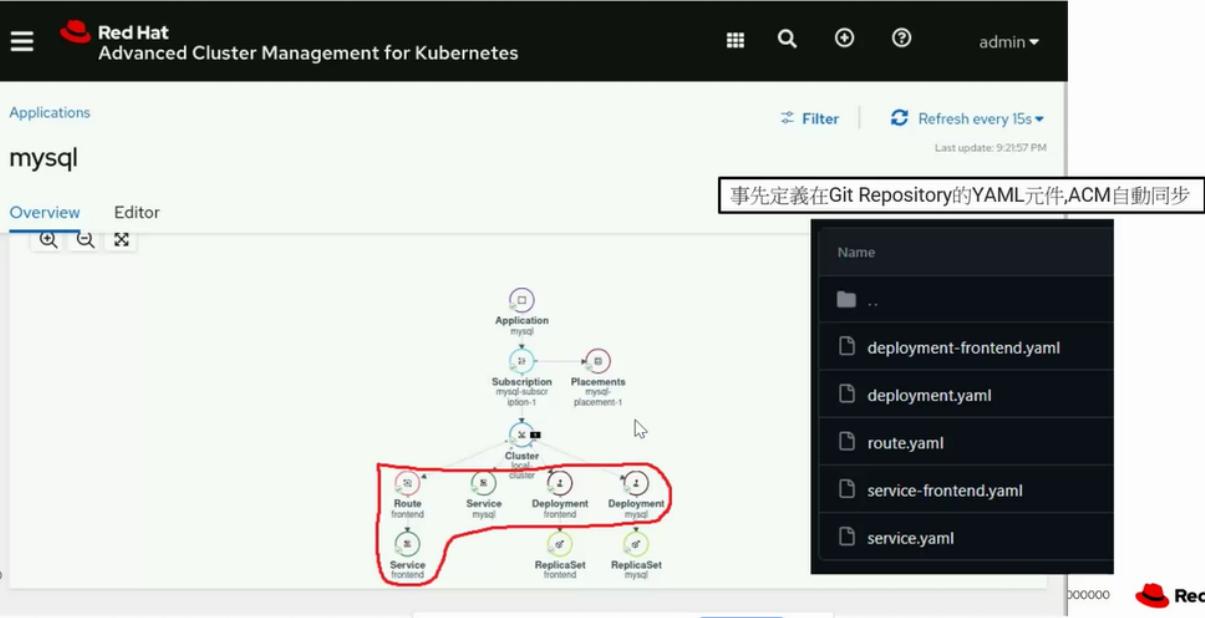


ACM subscription model 範例

ETMall Application Life Cycle w/ multi-OpenShift Cluster

CONFIDENTIAL, designator

透過ACM的Application Subscription Model,依據事先定義在Git Repository的OpenShift元件,自動佈署元件到指定的Cluster中



- kustomization.yaml

之後在overlays/sit-cluster目錄下新增route YAML檔並調整該目錄下kustomization.yaml

```
[student@workstation D0480-apps]$ vim overlays/apac/route.yaml
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  labels:
    app: todonodejs
    name: frontend
    name: frontend
spec:
  host: todo.apps.ocp4.example.com
  path: "/todo"
  to:
    kind: Service
    name: frontend-apac
    weight: 100
  wildcardPolicy: None

[student@workstation D0480-apps]$ vim overlays/apac/kustomization.yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
bases:
  - ../../base
nameSuffix: -apac
images:
  - name: mysql
    newName: registry.redhat.io/rhel8/mysql-80
    newTag: 1-152
-
```

新增整份
route.yaml

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
bases:
  - ../../base
nameSuffix: -apac
images:
  - name: mysql
    newName: registry.redhat.io/rhel8/mysql-80
    newTag: 1-152
resources:
  - route.yaml
```

Red Hat

68

參考

例如Git Repository中有以下目錄結構用來佈署mysql配合前端的應用系統:

```
base/kustomization.yaml
base/deployment-fronted.yaml
base/deployment.yaml
base/service-fronted.yaml
base/service.yaml
overlays/sit-cluster/kustomization.yaml
overlays/sit-cluster/route.yaml
overlays/uat-cluster/route.yaml
```

所有Cluster共用的元件

Cluster間的差異放在overlays目錄後

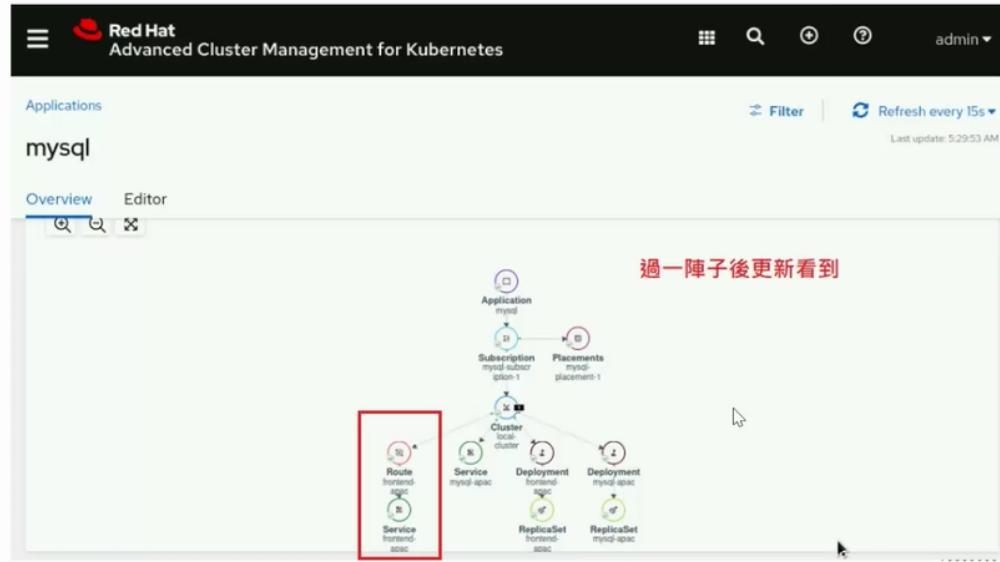
69

v0000000 Red Hat

ETMall Application Life Cycle w/ multi-OpenShift Cluster

CONFIDENTIAL designator

這時候佈署出來會如下結果,除了route元件是在overlays/{對應cluster label}目錄下定義外
其餘都是在base目錄中定義的元件,依樣可以佈署完整應用程式,只將差異放到overlays/{對應cluster label}目錄下



70

Red Hat

- kustomize 會將所有的yaml檔案合併成一個

經由上面介紹ACM的Subscription Model及進階透過 Kustomize結構目錄佈署,做以下整理:

該方式有遵守GitOps原則,並有以下好處:

- Declarative desired state : Git Repository上即儲存了佈署的元件狀態都記錄在YAML檔案內
- Immutable desired state : 佈署後所有元件狀態都不可透過Git以外方式異動,並歷史異動同Git歷史
- Infrastructure as code : 所有運行有關的OpenShift元件基礎設施都儲存在Git Repository內
- Merge requests : 異動過程可配合Git審核機制
- Continuous integration (CI) : 如同程式碼常態異動一樣,持續整合佈署的OpenShift元件
- 減少暴露ServiceAccount到Cluster之外的風險

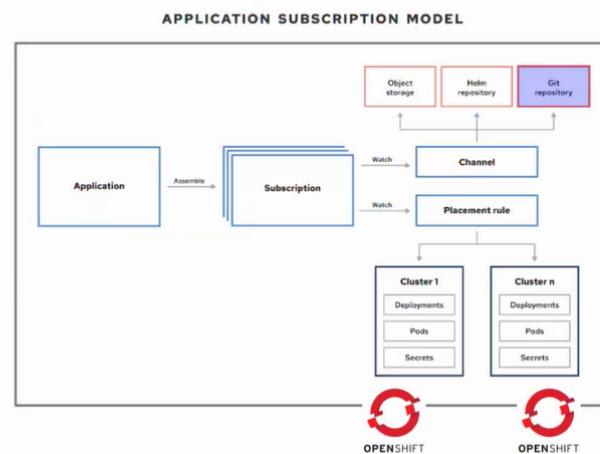
(注意,上述的CI指的是OpenShift/Kubernetes元件的CI,而非單指應用程式)



且透過Kustomize將不同叢集佈署的OpenShift元件簡化,不用重複撰寫

V0000000 Red Hat

但是ACM的Subscription Model在佈署前若無法管理被管的Cluster或甚至與Git同步有問題
在佈署前就發現並處理完問題



- 整合ansible Automation platform integration

ET-Mall Application Life Cycle w/ multi-OpenShift Cluster

ACM - Ansible Automation Platform Integration (Application)

Application Lifecycle:

- Add the Ansible Automation Platform credentials, and select it in the application creation / update UI flow
- On creation of an application via ACM, trigger an Ansible job template before AND/OR after the application creation
- On update of an application via ACM, trigger an Ansible job template before AND/OR after the application update (referred to as Pre and Post hooks)

Create an application

Configure automation for prehook and posthook

Ansible Automation Platform credential: ansible-tower-secret

Red Hat Advanced Cluster Management with Ansible Automation Platform

CONFIDENTIAL. designator

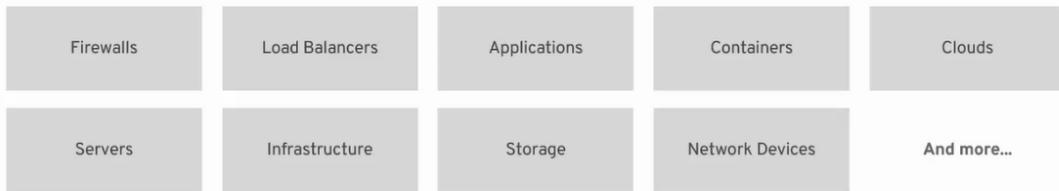
HOW WILL ET-MALL USE ANSIBLE with RHACM?

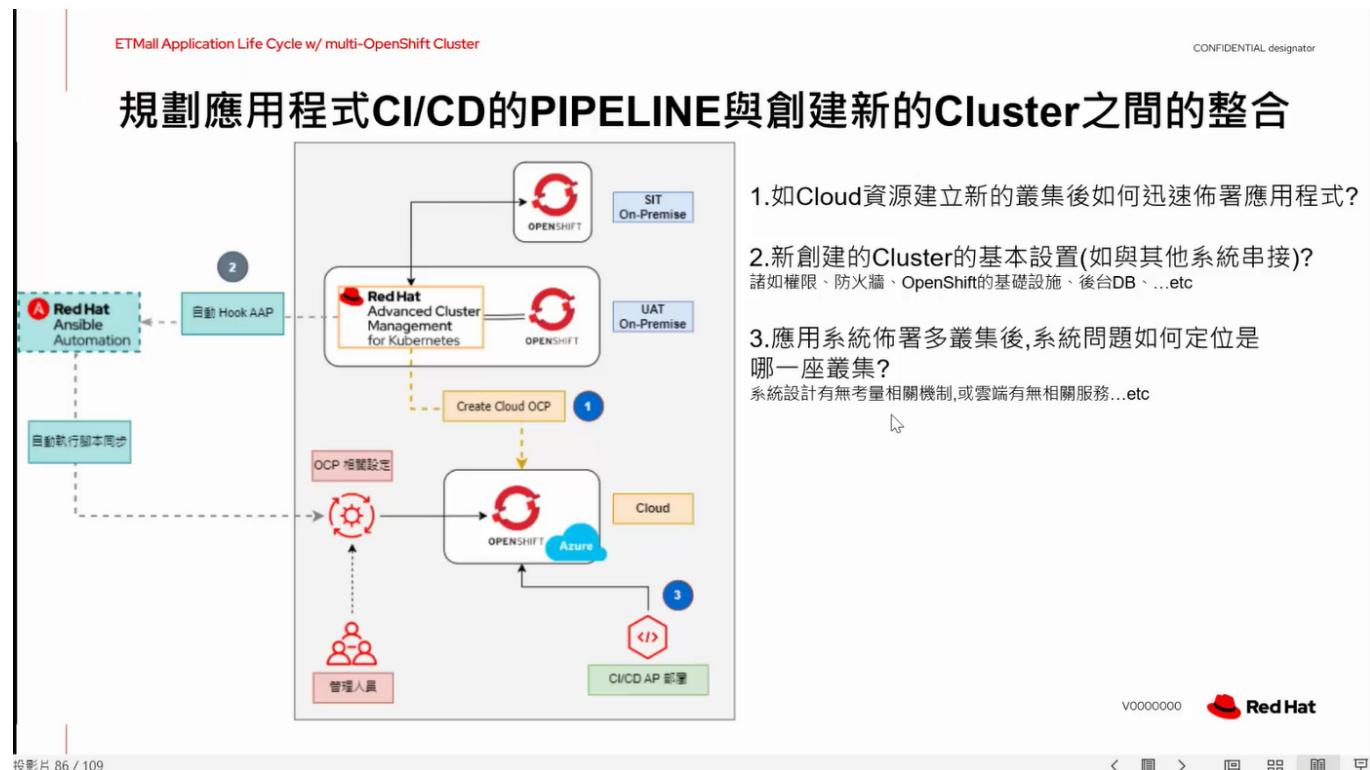
Automate the deployment and management of your entire IT footprint.

Do this...



On these...





Universal Base Image & Quay

This section
includes:

CHALLENGES IN SELECTING THE RIGHT CONTAINER BASE IMAGE

INTRODUCING THE RED HAT UNIVERSAL BASE IMAGE

Red Hat Quay Organizations, Teams, Users, Robot

Repository Permissions and Role Based Access Control (RBAC)

Organization Storage Quota Management

RHACM And Quay

- 挑選適合的base image
- Red UBI 介紹
- Quay for org
- RBAC

There is some standard criteria that can help

Architecture	Security	Performance
<ul style="list-style-type: none"> • C Library • Core Utilities • Size • Life Cycle • Compatibility • Troubleshooting • Technical Support • ISV Support • Distributability 	<ul style="list-style-type: none"> • Updates • Tracking • Security Response Team 	<ul style="list-style-type: none"> • Automated • Performance • Engineering



V0000000



挑選base image 的考量

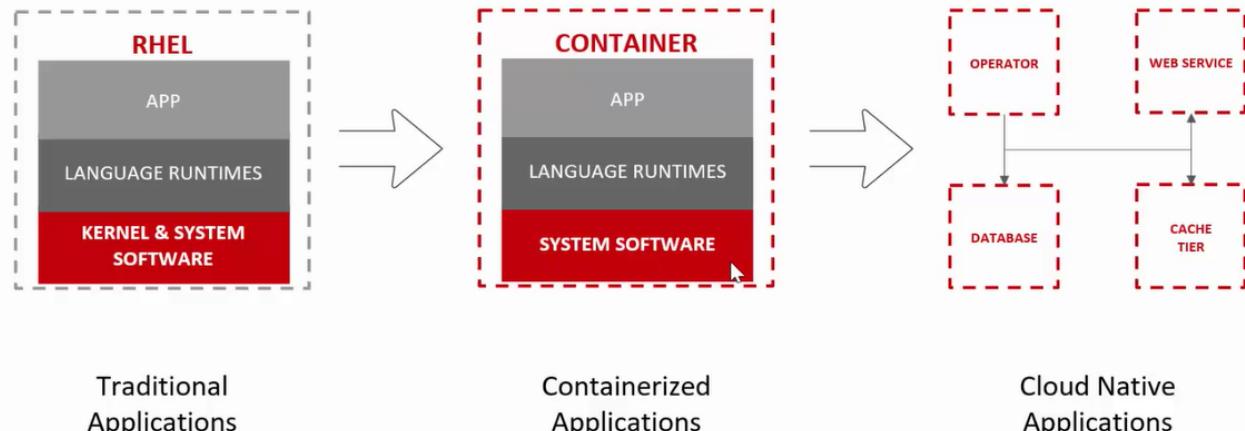
- 架構
- 安全性
- 效能

ETMall Red Hat Universal Base Image & Quay

CONFIDENTIAL design

THE BASE IMAGE FOR ALL OF YOUR NEEDS

Bringing the value of RHEL to cloud native applications

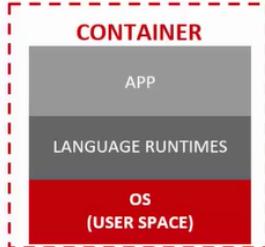


2.Redhat_Quay與UBI更新.pptx - 受保護的檢視 - PowerPoint

ETMall Red Hat Universal Base Image & Quay CONFIDENTIAL designator

THE BASE IMAGE FOR ALL OF YOUR NEEDS

Enterprise architecture, security and performance



The Red Hat Universal Base Image is based on RHEL and 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

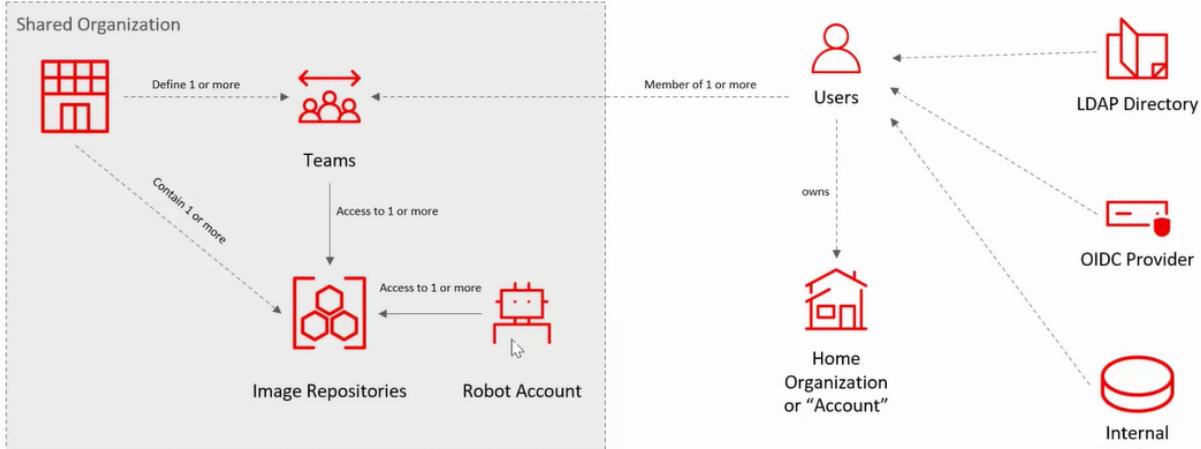
- Supported as RHEL when running on RHEL
- Same Performance, Security & Life cycle as RHEL
- Can attach RHEL support subscriptions as RHEL

11 V0000000 Red Hat

2.Redhat_Quay與UBI更新.pptx - 受保護的檢視 - PowerPoint

RED HAT QUAY ORGANIZATIONS, TEAMS, USERS, ROBOT ACCOUNTS CONFIDENTIAL designator

Quay Tenancy Model



17 V0000000 Red Hat

2.Redhat_Quay與UBI更新.pptx - 愛保護的檢視 - PowerPoint

ETMall Red Hat Quay & UBI Image Update

CONFIDENTIAL designator

規劃Image Registry(Quay)的使用:

- 1.不同座Quay的使用情境
- 2.Quay Organization/Teams/Users/Robot Accounts對應人員與系統定義及權限管理
- 3.Image儲存/備援/同步(PIPELINE整合) (or 雲端Registry)
- 4.Clusters合法Image來源管理(Global Cluster Pull Secret/ACM Governance-policy)
- 5.Image Registry儲存的Image Tag移除(舊Image移除)機制
- 6.Base Image備存(提供更新Base Image)

33

V0000000 Red Hat