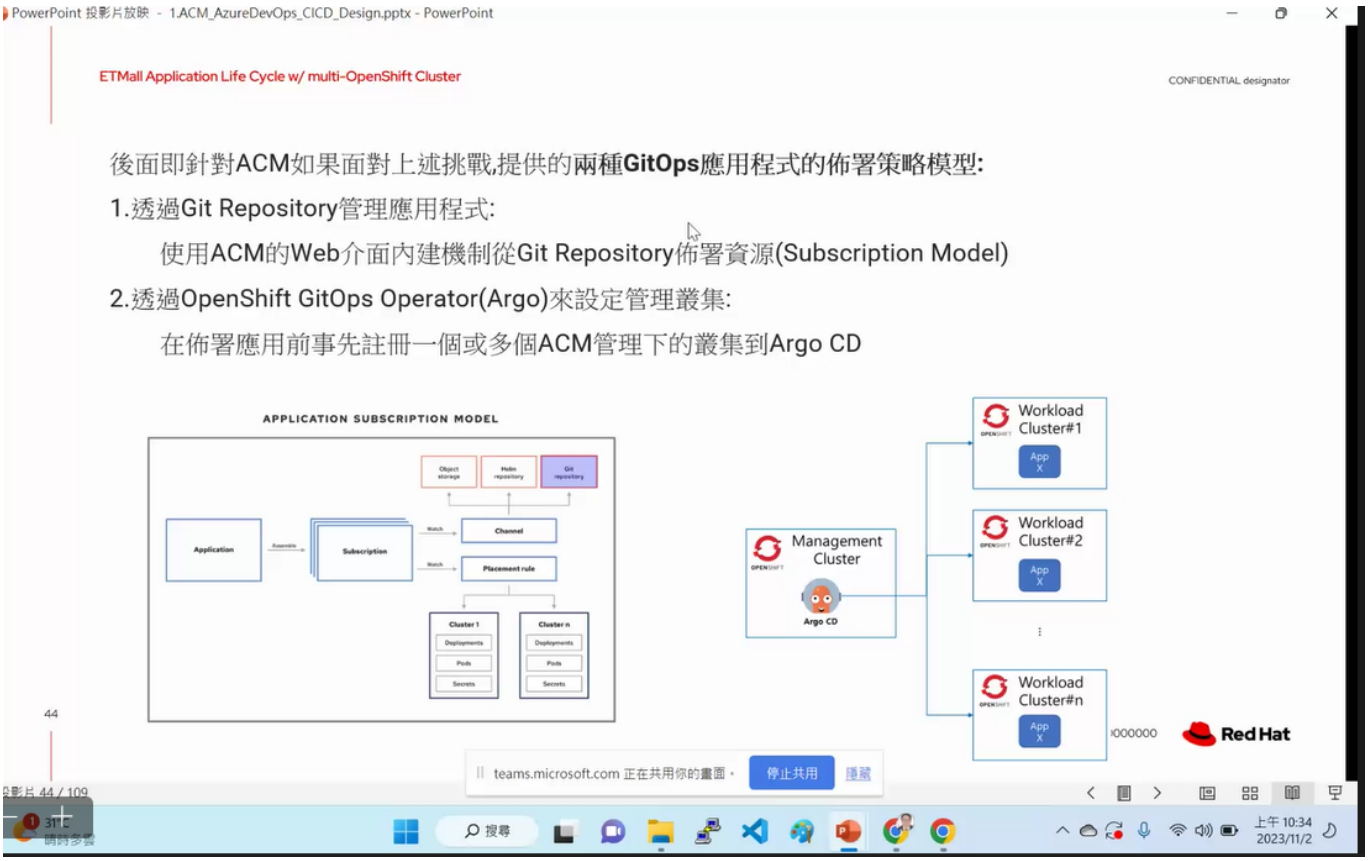
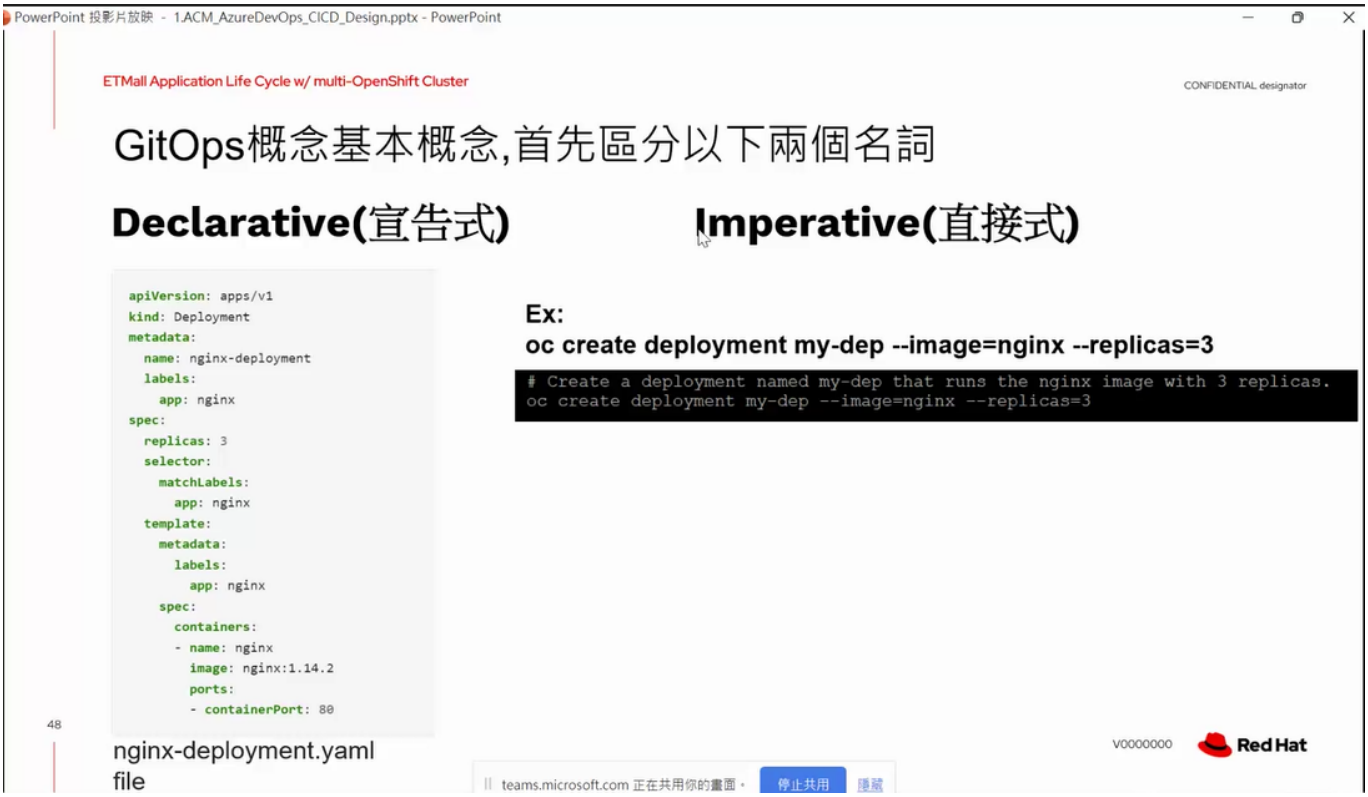
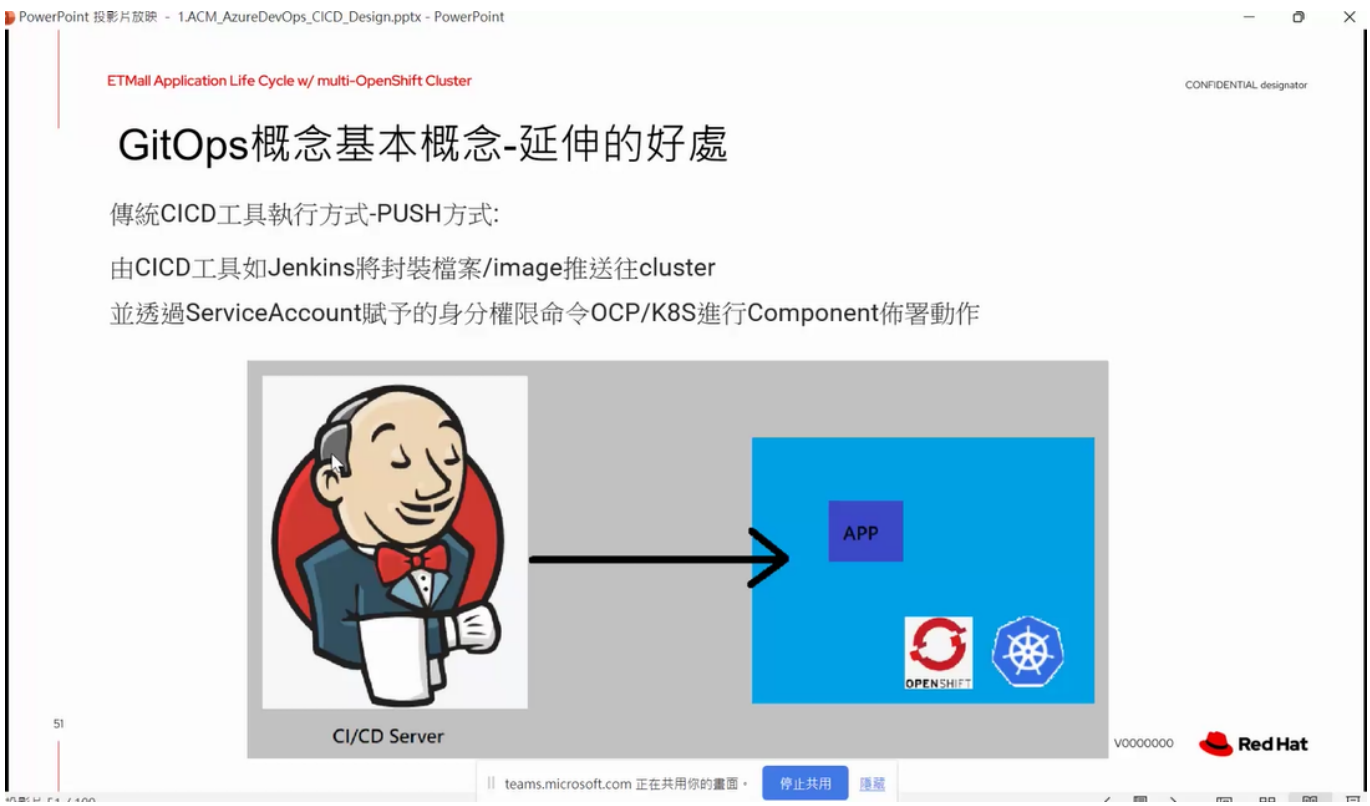


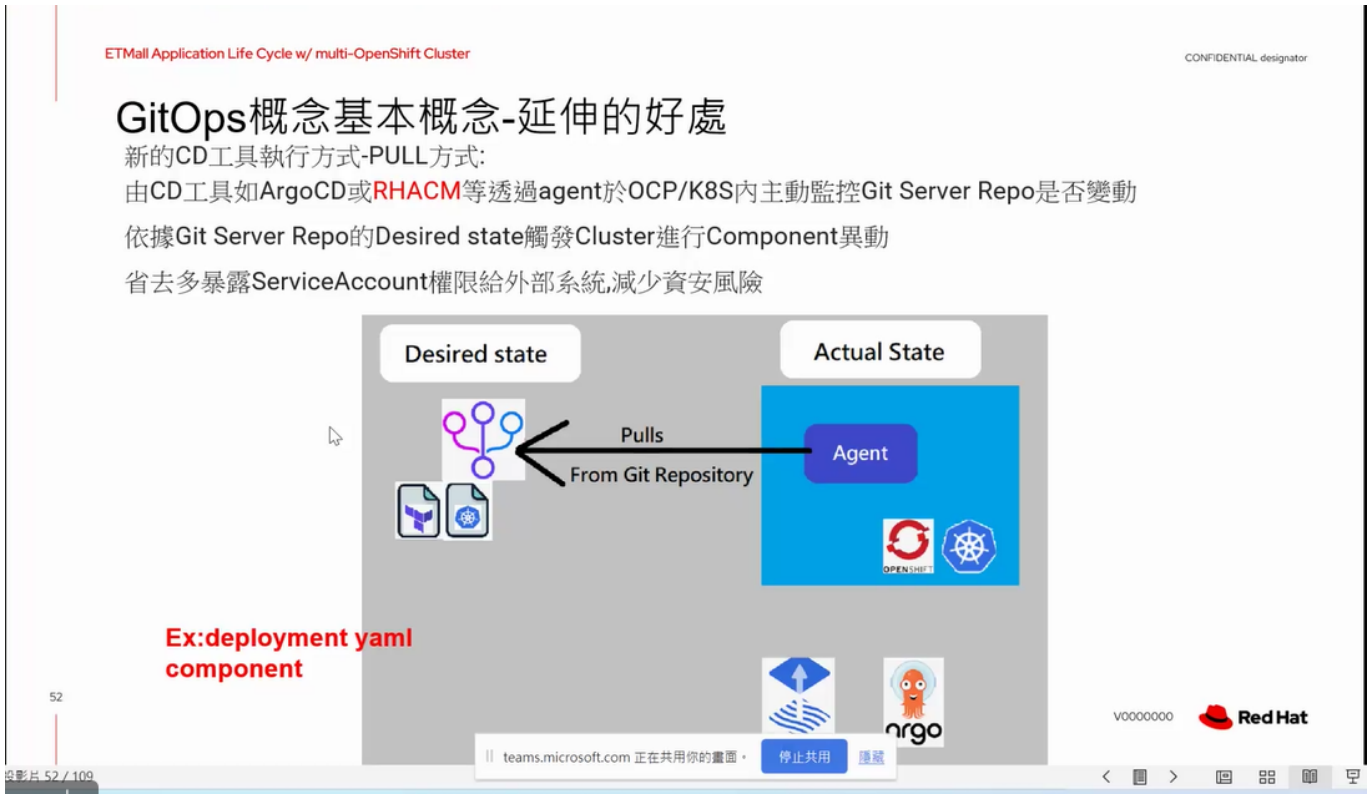
- ACM 有自有的subscription model



- 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

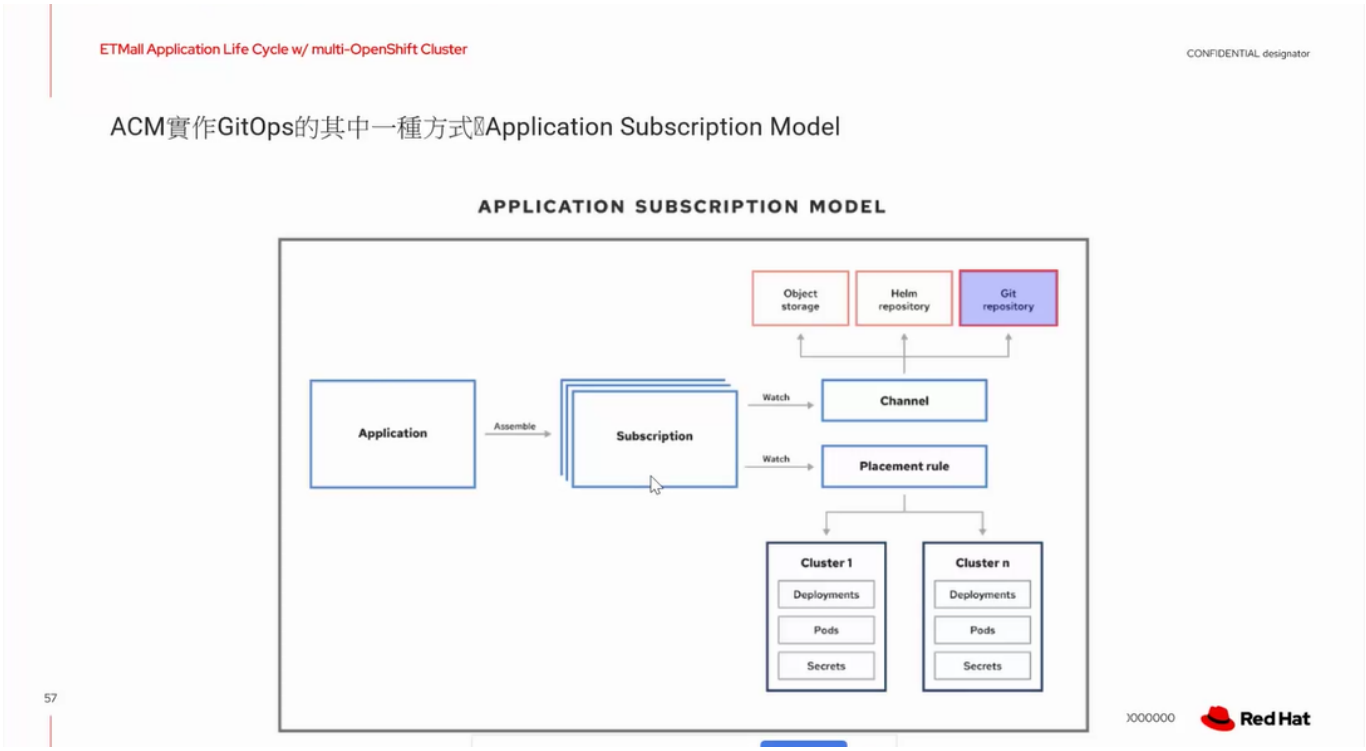

PowerPoint 投影片放映 - 1.ACM_AzureDevOps_CICD_Design.pptx - PowerPoint
ETMall Application Life Cycle w/ multi-OpenShift Cluster
CONFIDENTIAL designator
GitOps概念基本概念
GitOps : Infrastructure as Code Done Right
Infrastructure as Code or X as Code
•Infrastructure as Code
•Policy as Code
•Networking as Code
•Security as Code
•Configuration as Code
•...etc
50
V0000000
Red Hat
teams.microsoft.com 正在共用你的畫面。
停止共用
隱藏
投影片 50 / 109
搜尋
上午 10:38
2023/11/2



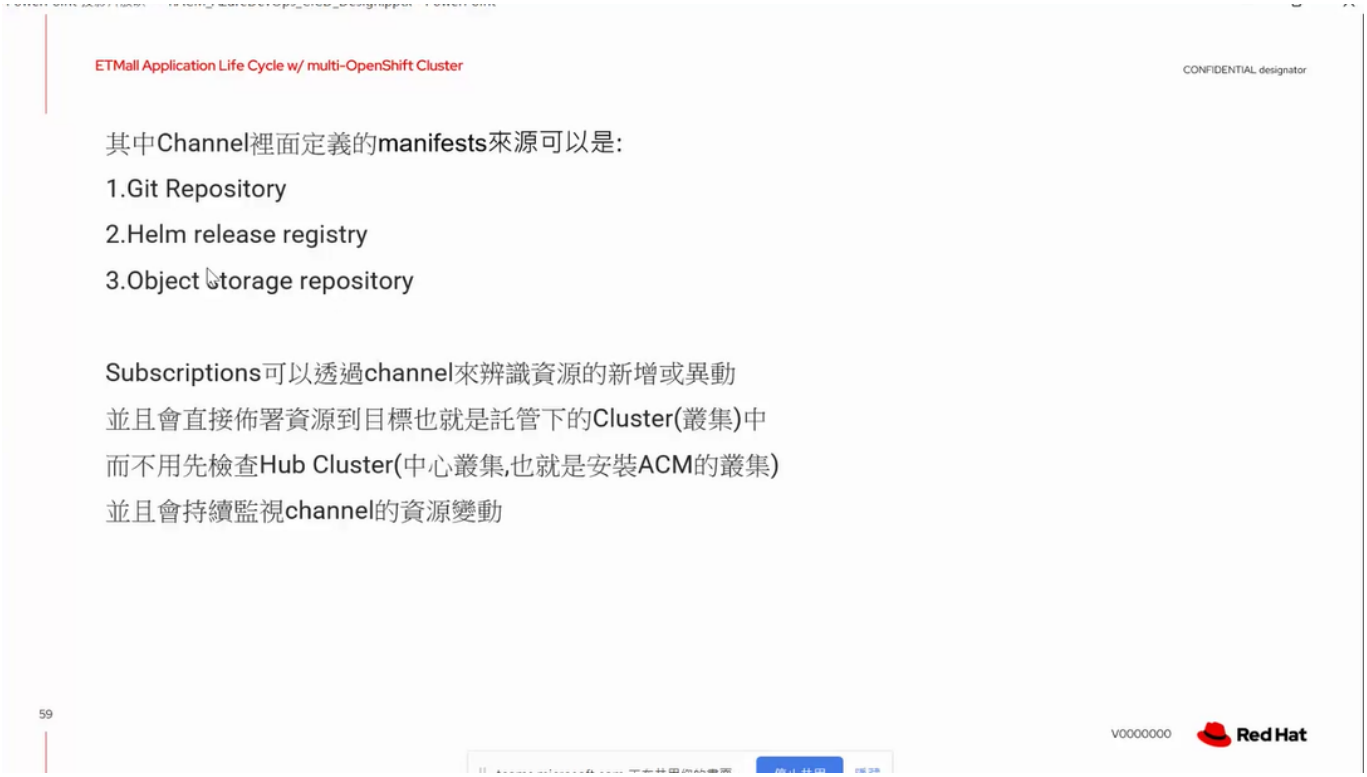


- push式 賦予權限給k8s
- pull式 k8s自己去取得權限

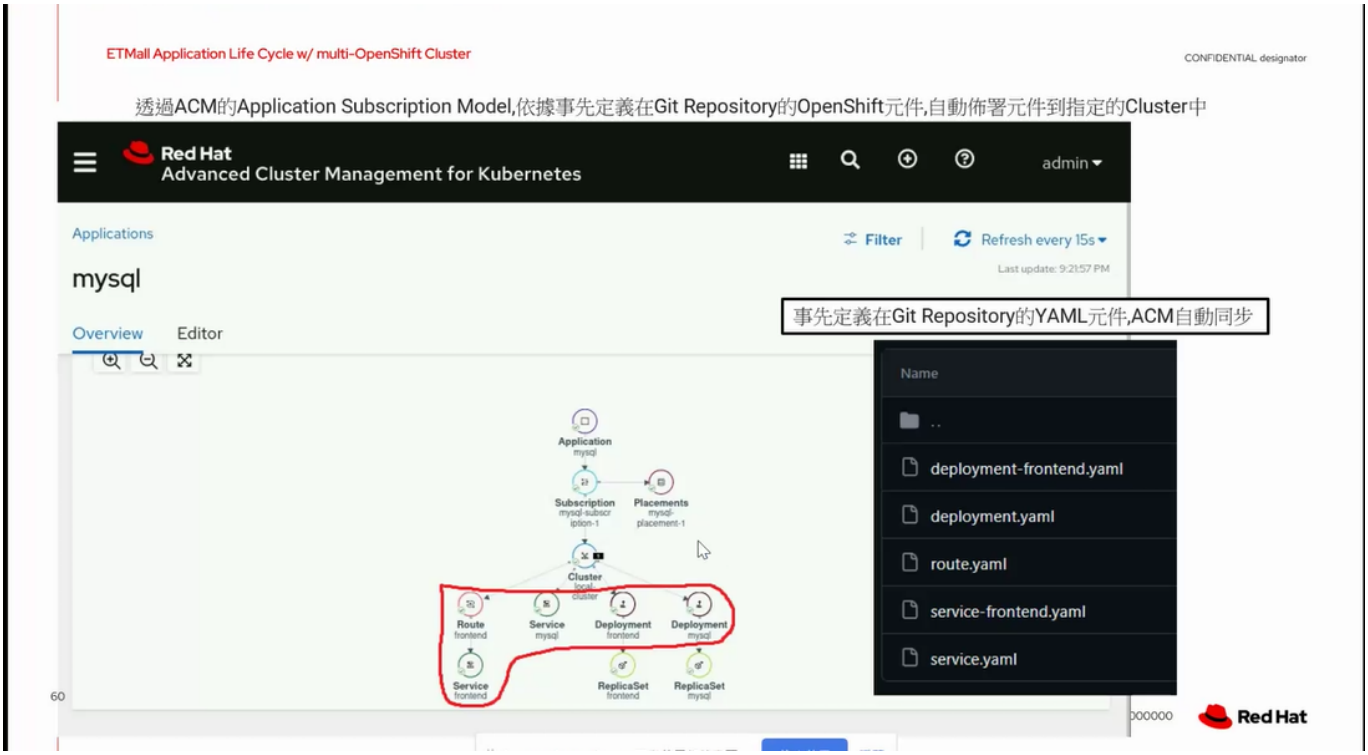
- 須嚴格遵守single source of truth
- 透過gitops來達成
- ACM Application Subscription Model



- 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
- plcaement 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



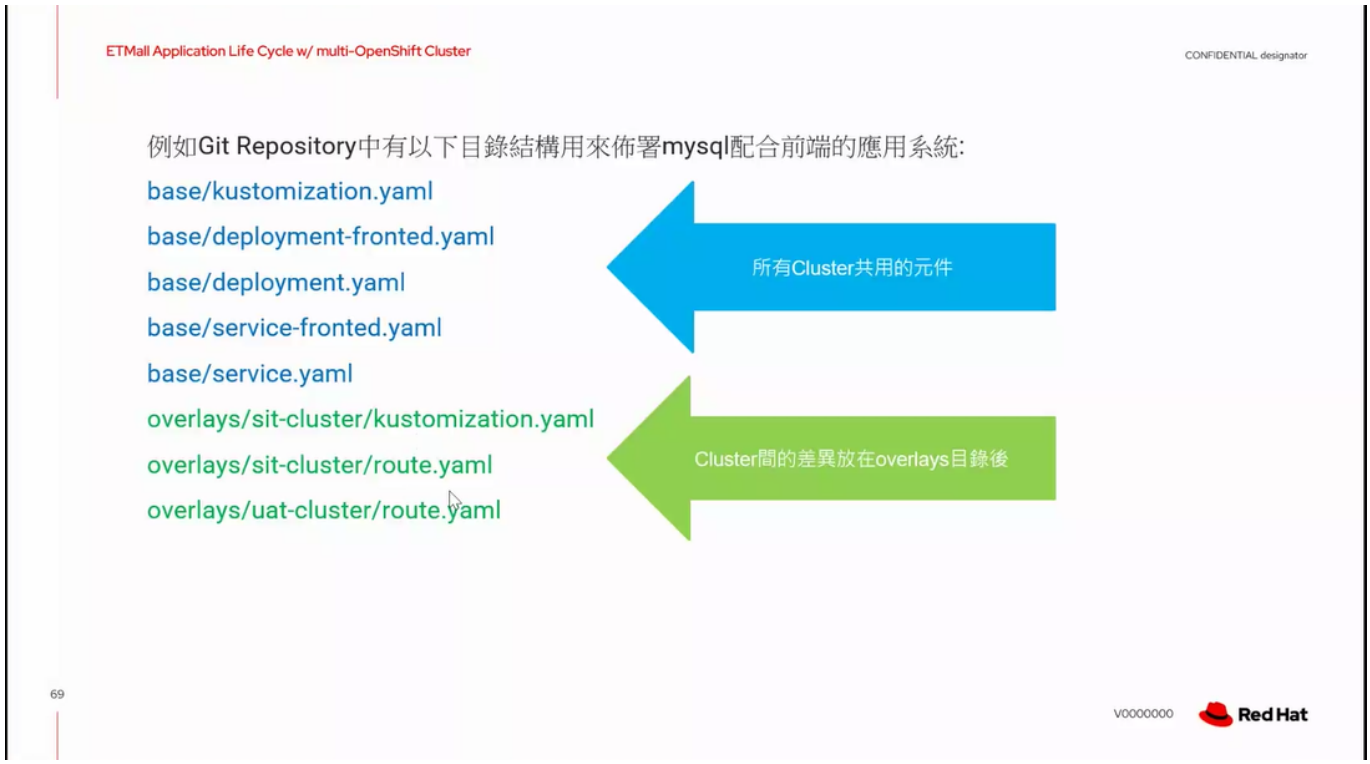
ACM subscrition model 範例

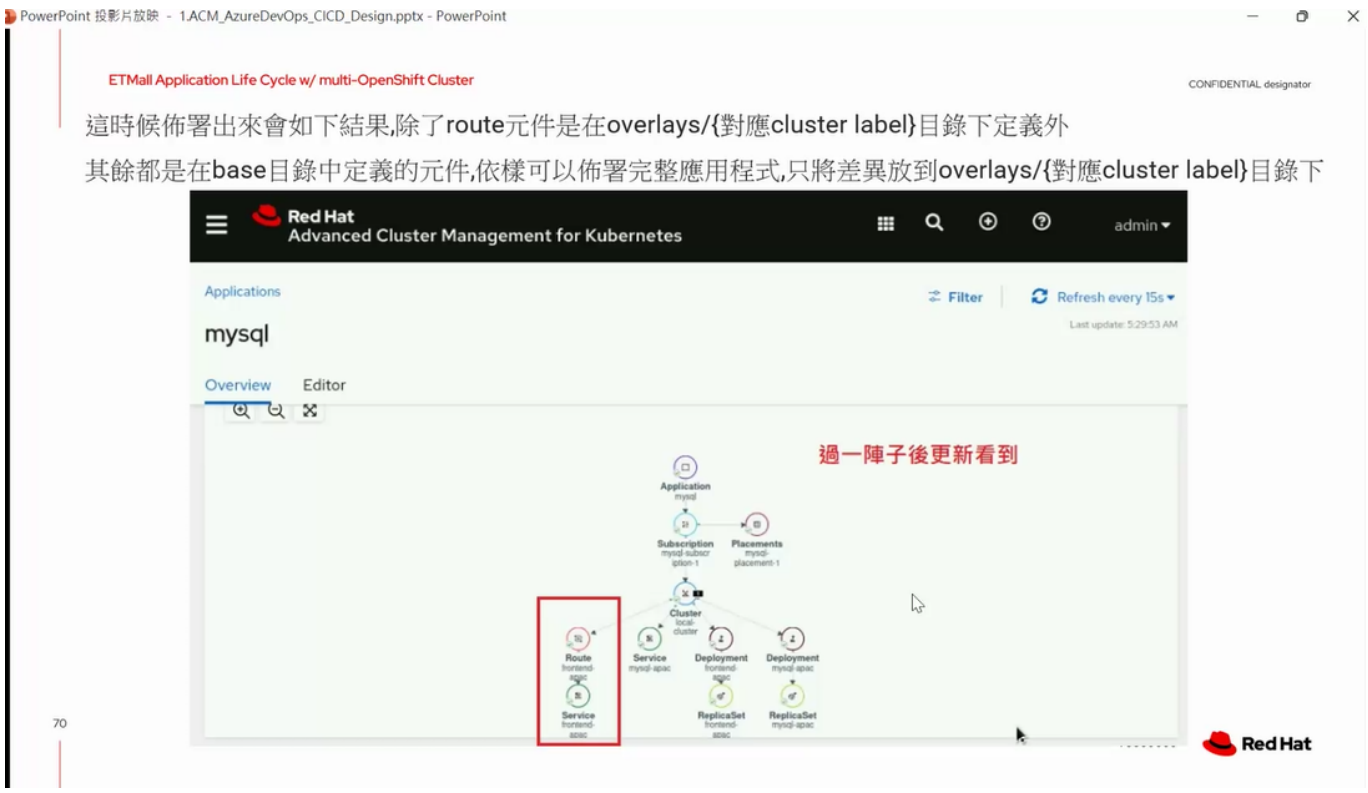


- kustomization.yaml


ETMall Application Life Cycle w/ multi-OpenShift Cluster
CONFIDENTIAL designator
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 DO480-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
新增整份
route.yaml
[student@workstation DO480-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
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
68
Red Hat

參考





- 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:

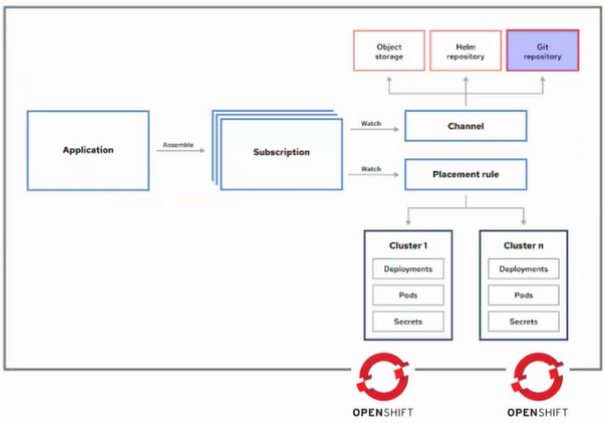
- Declarative desired state : Git Repository上即儲存了佈署的元件狀態都記錄在YAML檔案內
- Immutable desired state : 佈署後所有元件狀態都不可透過Git以外方式異動,並歷史異動同Git歷史
- Infrastructure as code : 所有運行有關的OpenShift元件基礎設施都儲存在Git Repository內
- Merge requests : 異動過程可配合Git審核機制
- Continuous integration (CI) : 如同程式碼常態異動一樣,持續整合佈署的OpenShift元件
- 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

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

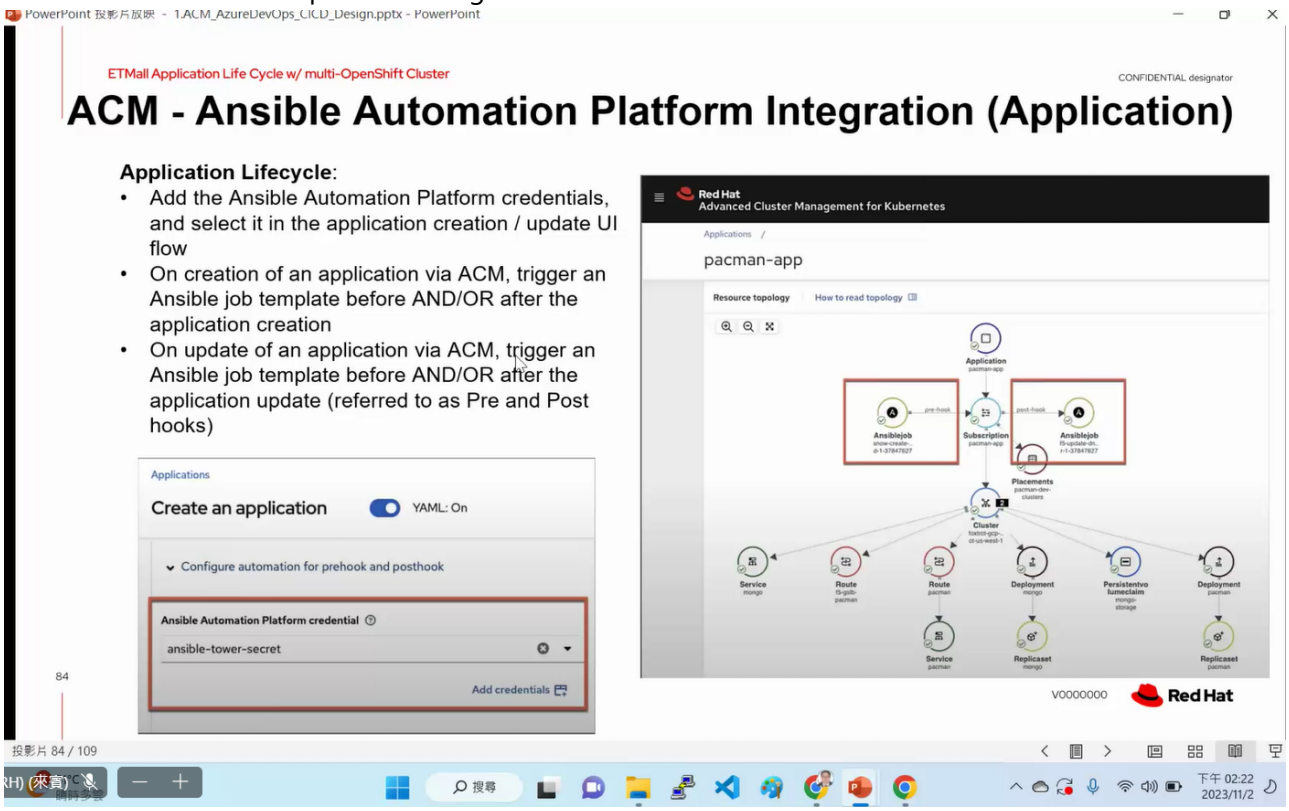
且透過Kustomize將不同叢集佈署的OpenShift元件簡化,不用重複撰寫

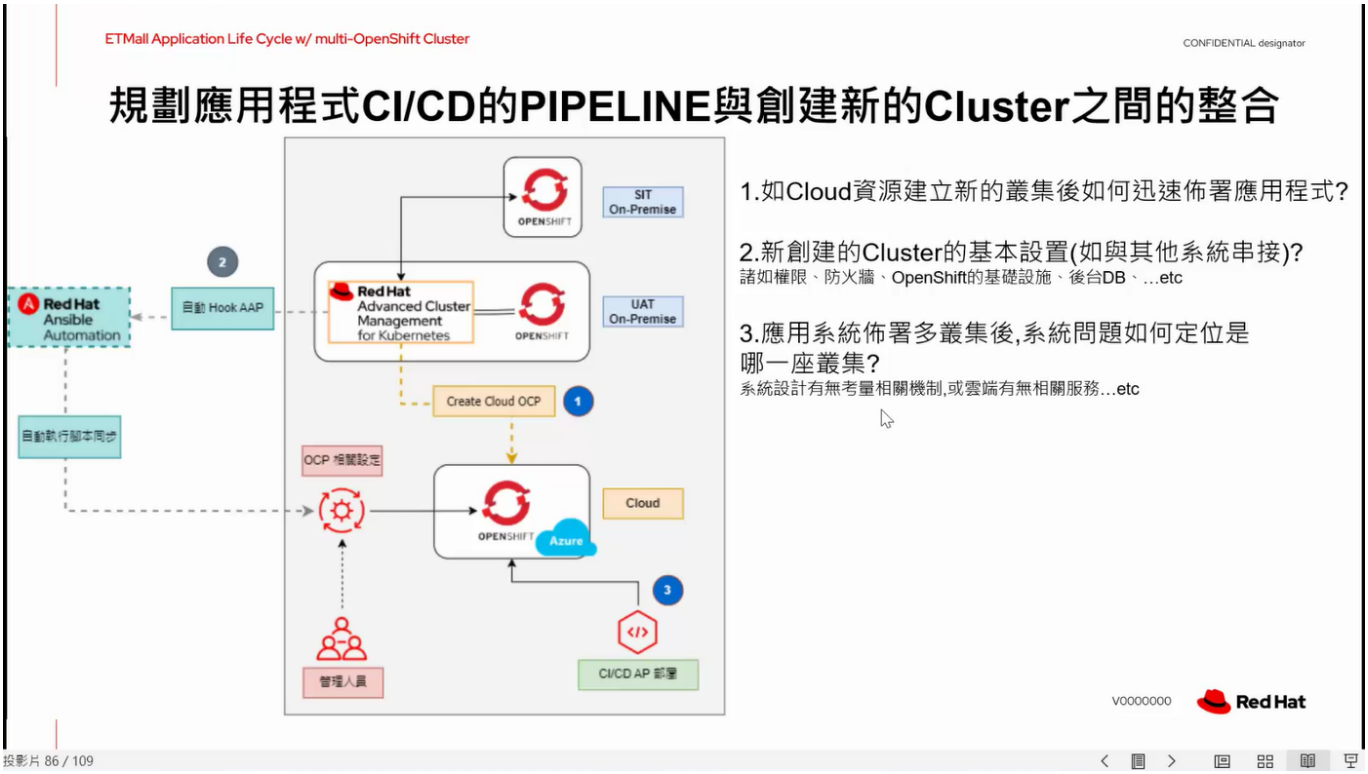
但是ACM的Subscription Model在佈署前若無法管理被管的Cluster或甚至與Git同步有問題
在佈署前就發現並處理完問題

APPLICATION SUBSCRIPTION MODEL

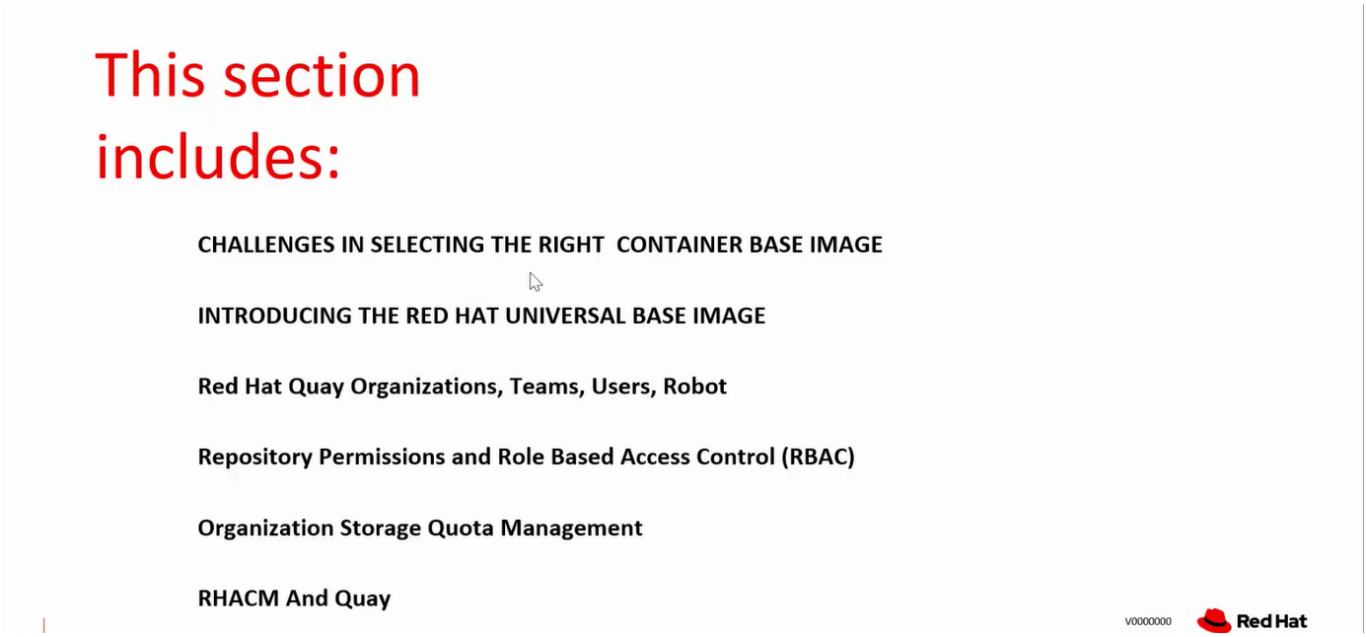


- 整合ansible Automation platform integration





Universal Base Image & 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

- C Library
- Core Utilities
- Size
- Life Cycle
- Compatibility
- Troubleshooting
- Technical Support
- ISV Support
- Distributability

Security

- Updates
- Tracking
- Security Response Team

Performance

- Automated
- Performance Engineering

V0000000 Red Hat

挑選base image 的考量

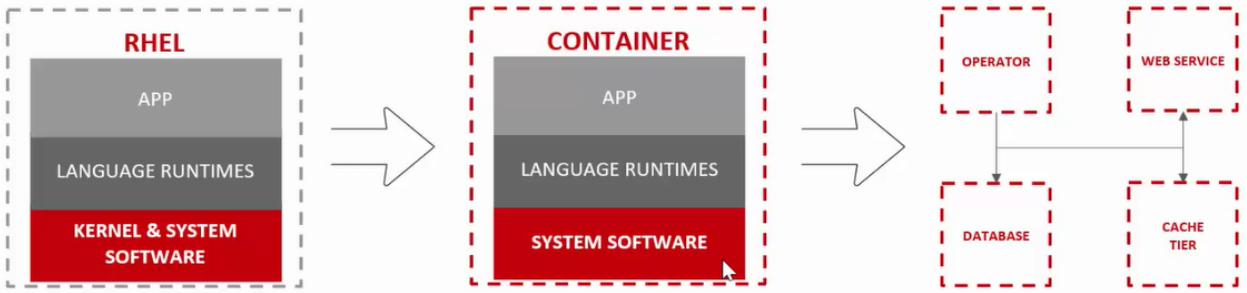
- 架構
- 安全性
- 效能

ETMall Red Hat Universal Base Image & Quay

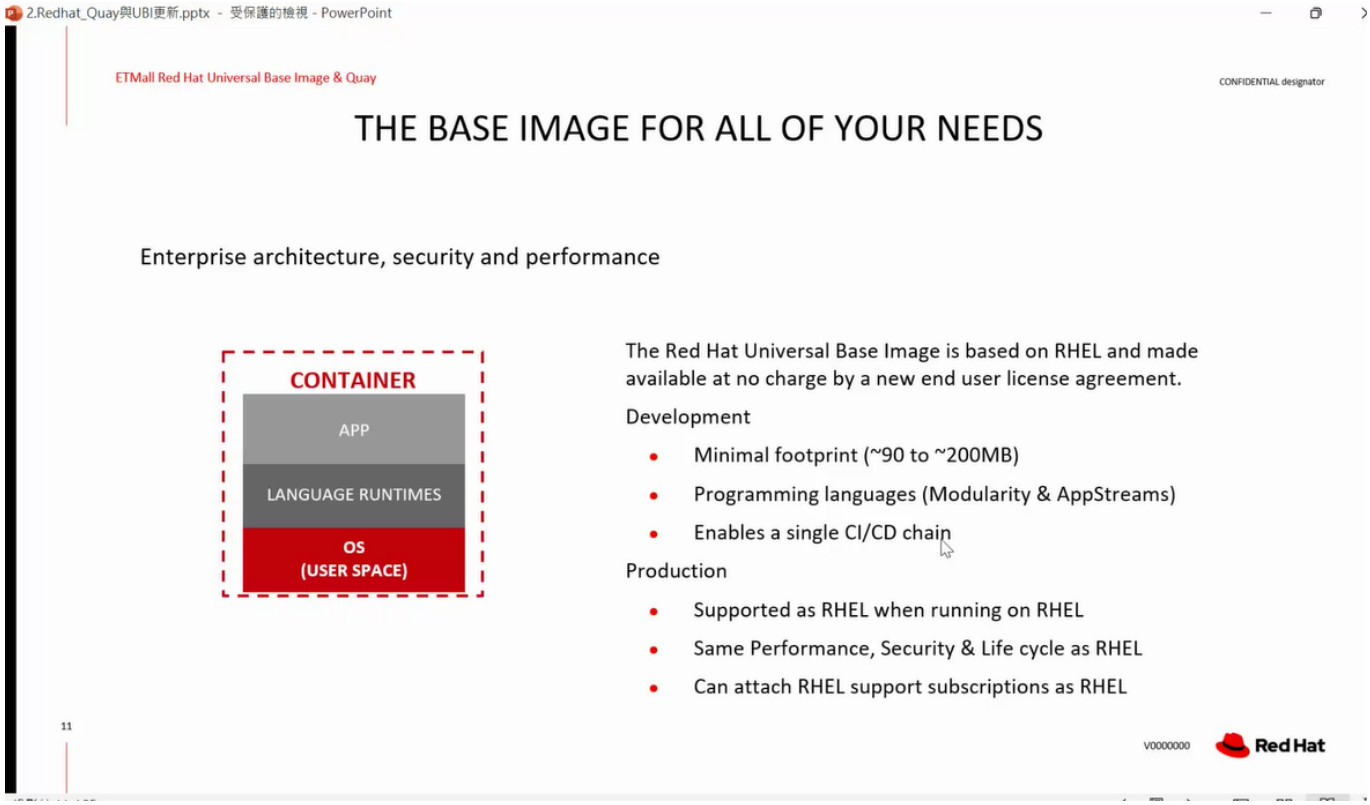
CONFIDENTIAL desi

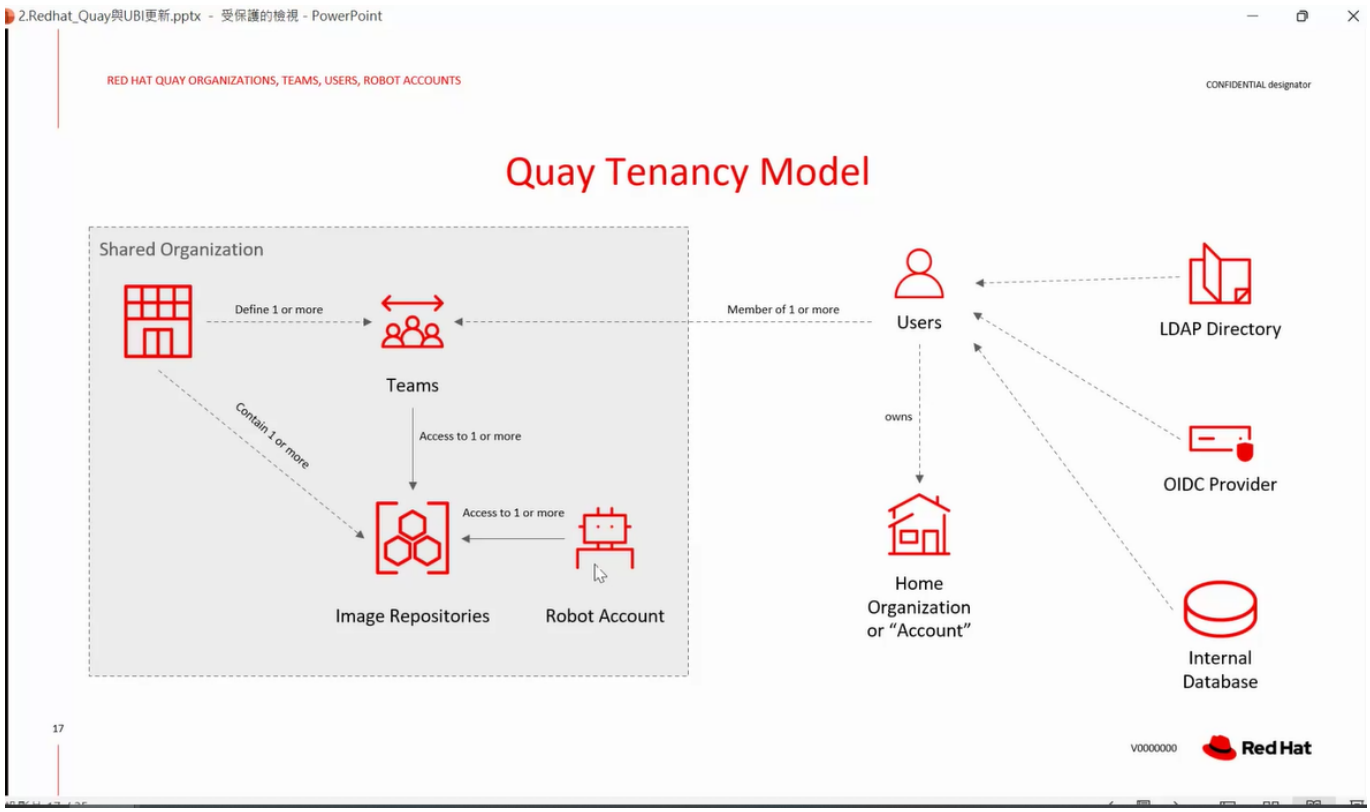
THE BASE IMAGE FOR ALL OF YOUR NEEDS

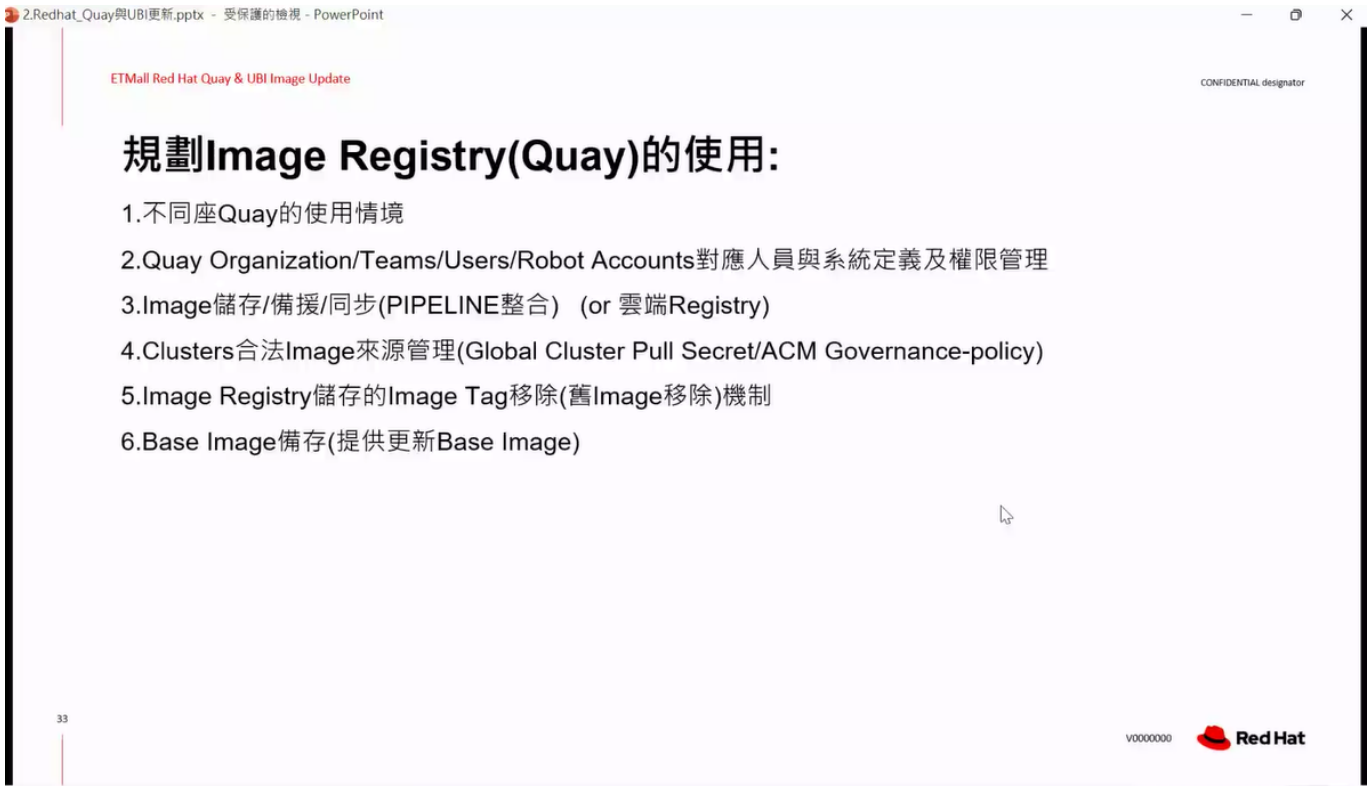
Bringing the value of RHEL to cloud native applications



Traditional Applications | Containerized Applications | 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
CONTAINER
APP
LANGUAGE RUNTIMES
OS
(USER SPACE)
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
Shared Organization
Define 1 or more
Teams
Contain 1 or more
Access to 1 or more
Access to 1 or more
Image Repositories
Robot Account
Member of 1 or more
Users
owns
Home
Organization
or "Account"
LDAP Directory
OIDC Provider
Internal
Database
17
V0000000
Red Hat


2.Redhat_Quay與UBI更新.pptx - 受保護的檢視 - 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