

GSE | GUIDE  
SHARE  
EUROPE

W W W . G S E . O R G . U K



## Modernize mainframes with Red Hat OpenShift Platform

Fabio Alessandro Locati  
Red Hat

April 2024  
Session 3E



# About me

- ▶ Working in IT since 2004, mostly in consulting roles
- ▶ Author of 5 books
- ▶ EMEA Associate Principal Specialist Solution Architect @ Red Hat







# Containers advantages

- ▶ Less overhead
- ▶ Increased portability
- ▶ More consistent operation
- ▶ Greater efficiency
- ▶ Better application development

## Containers usecases

- ▶ “Lift and shift” existing applications into modern cloud architectures
- ▶ Refactor existing applications for containers
- ▶ Develop new container-native applications
- ▶ Provide better support for microservices architectures
- ▶ Provide DevOps support for continuous integration and deployment (CI/CD)
- ▶ Provide easier deployment of repetitive jobs and tasks

# Kubernetes advantages

- ▶ Service discovery and load balancing
- ▶ Storage orchestration
- ▶ Automated rollouts and rollbacks
- ▶ Automatic bin packing
- ▶ Self-healing
- ▶ Secret and configuration management

# Kubernetes naming

- ▶ **Container:** a group of processes with limited access to the system and resources, leveraging *cgroups*
- ▶ **Container Image:** a tar file containing all the required files and configurations to run a container
- ▶ **Pod:** a group of container
- ▶ **Service:** Kubernetes way to expose Pods ports over network
- ▶ **Persistent Volume:** a disk that is usable by a Pod
- ▶ **Config Map:** Kubernetes way to set configuration in Pods via file or *ENV\_VARS*
- ▶ **Secret:** Kubernetes way to store and inject secret strings



# Kubernetes components

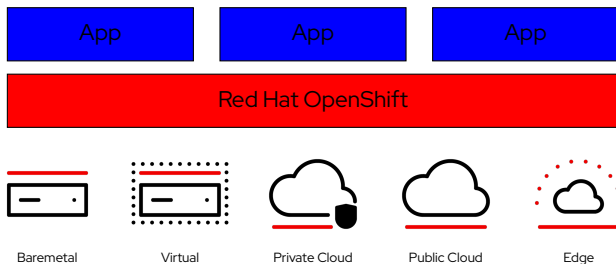
- ▶ **etcd**: a decentralised file storage database
- ▶ **api-server**: Kubernetes API Control Plane
- ▶ **Control Plane node**: a node that controls the cluster by running *etcd* and *api-server*
- ▶ **Worker node**: a node that runs workload
- ▶ **Infrastructure node**: a node that runs additional system components

## Hybrid and multicloud is the new normal

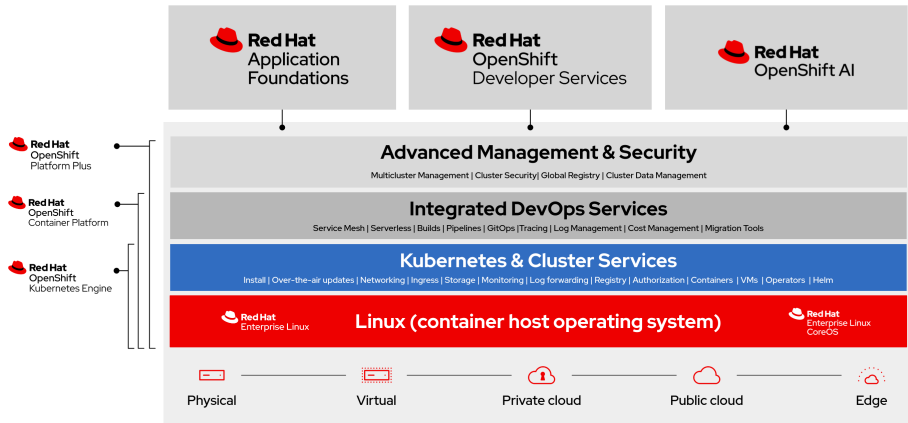
- ▶ 95% of enterprises will be using a mix of cloud models
- ▶ 31% of all compute capacity will be in a Private, on premises cloud in 2 years
- ▶ 60% of enterprises will utilize flexible consumption models by 2023

Sources: IDC Cloud Forecast, BCG, and McKinsey

# Red Hat OpenShift - an abstraction platform



# Red Hat OpenShift, much more than Kubernetes



# Installation options

## SNO

Cluster with a single node combining control and compute

- ▶ Pro:
  - ▶ Reduced IFL footprint
  - ▶ Compute can be added
  - ▶ Kubernetes functions available
  - ▶ Good for Dev/Test
- ▶ Cons:
  - ▶ No infrastructure High Availability
  - ▶ Updates contain service interruptions

## 3 Nodes

Cluster with initially three nodes combining control and compute

- ▶ Pro:
  - ▶ Reduced IFL footprint
  - ▶ Can grow by adding compute nodes
  - ▶ Cluster High Availability given
- ▶ Cons:
  - ▶ Cluster size might be limited for applications and grow cluster might be required

## Production

Cluster with a minimum of 3 control nodes and 2 compute nodes

- ▶ Pro:
  - ▶ Full cluster availability
  - ▶ Full Flexibility placing applications and grow
  - ▶ Dedicated control nodes for High Availability
- ▶ Cons:
  - ▶ Cluster IFL footprint highest. But will only slightly grow
  - ▶ Setup for Dev/Test env potentially oversized



# Multiarchitecture deployment options

- ▶ 4.14
  - ▶ A cluster with control planes and compute nodes on x86 architecture
  - ▶ With additional s390x compute nodes
- ▶ 4.15
  - ▶ A cluster with control planes and compute nodes on s390x architecture
  - ▶ With additional x86 compute nodes

# LinuxOne as Management Hub

- ▶ Hybrid Multi-Architecture Multi Cloud management
  - ▶ Using RH Advanced Cluster Management for Kubernetes
  - ▶ Single Pane of Glas
  - ▶ On-premise
  - ▶ Heterogeneous Kubernetes Container environments
  - ▶ Multi Cloud & Multi-Architecture
  - ▶ Including Kubernetes in public clouds
- ▶ Centralized integrated RH OpenShift Automation
  - ▶ Using OpenShift Pipelines
  - ▶ Across RH OpenShift environments



Containers will not displace the  
mainframe, they will enhance it

# Why Red Hat OpenShift on the Mainframe

- ▶ Application Development Consistency
- ▶ Leverage industry knowledge and tools
- ▶ Workload portability

# Why the Mainframe under Red Hat OpenShift

- ▶ Data gravity
- ▶ Low latency between LPARs
- ▶ Consolidation and TCO reduction
- ▶ Business Continuity
- ▶ Leverage Mainframe unique hardware capabilities



## HA and DR in Red Hat OpenShift and Mainframe

- ▶ OpenShift only handles Pod failures not Node failures
- ▶ OpenShift needs a majority of `etcd` nodes running to maintain cluster stability. If a majority of `etcd` nodes go down the recovery might need to be done manually
- ▶ Software-defined persistent storage alone cannot achieve zero RTO and zero RPO that mission critical stateful workloads demand
- ▶ The Mainframe's HA capabilities can ensure that OpenShift nodes do not go down while providing near zero RTO and zero RPO for stateful workloads when combined with external storage
- ▶ Does not need to be enabled for everything – can be partially enabled for workloads that require it

## Wrapping up

- ▶ It is key to focus on portable applications
- ▶ Containers can bring new tooling to Mainframe development
- ▶ The Mainframe is a great platform to run containerized workloads
- ▶ Red Hat OpenShift enables portable applications without giving up the specific platform optimizations

1 2 3 4 5 6 7 8 9

GSE | GUIDE  
SHARE  
EUROPE

W W W . G S E . O R G . U K