GSE

GUIDE
SHARE
EUROPE

UK Region

Modernize mainframes workloads with Red Hat OpenShift Platform

Fabio Alessandro "Fale" Locati
Principal Specialist Solution Architect @ Red Hat

November 2024

DV

GSE UK Conference 2024 Charities

Every year, speakers and delegates at the GSE UK Conference donate to the charities we help. This year is no exception and we aim to collect donations for two worthy causes: Air Ambulance UK and Muscular Dystrophy UK.

Please consider showing your appreciation by kindly donating a small sum to our charities this year!!

AIR AMBULANCES UK
SUPPORTING AIR AMBULANCE CHARITIES

MUSCULAR DYSTROPHY UK
OUR MUSCLES MATTER

About me

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


MAERSK
MAERSK SEALAND
BEACON
UASC
Hapag-Lloyd
MAERSK
MAERSK SEALAND


CMA CGM
TCLC
SINOTRANS


MSC


Safmarine
MAERSK

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

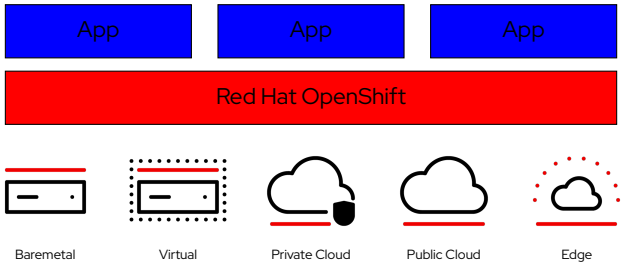
- **etcd**: a decentralised file storage database
- **api-server**: Kubernets API Control Plane
- **Control Plane node**: a node that controls the cluster by running *etcd* and *api-server*
- **Worker node**: a node that runs workload
- **Infrastructure node**: a node that runs additional system components

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 conumption models by 2023

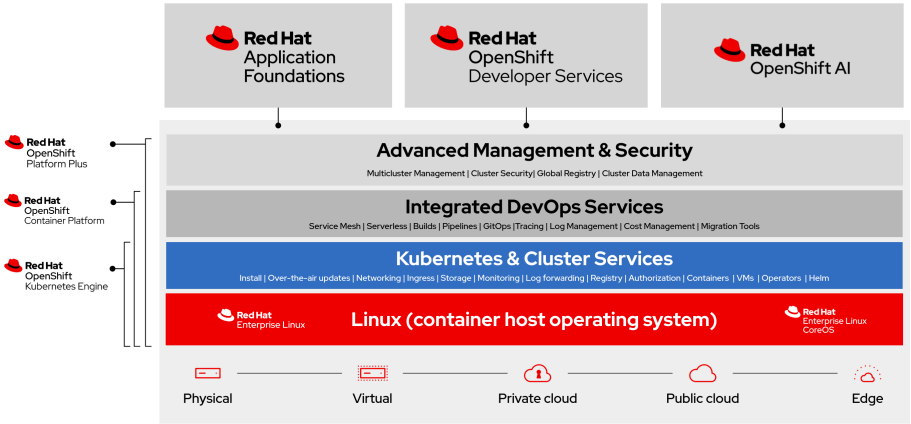
Sources: IDC Cloud Forecast, BCG, and McKinsey

Red Hat OpenShift - an abstraction platform



Red Hat

Red Hat OpenShift, much more than Kubernetes



Red Hat

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 initally 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

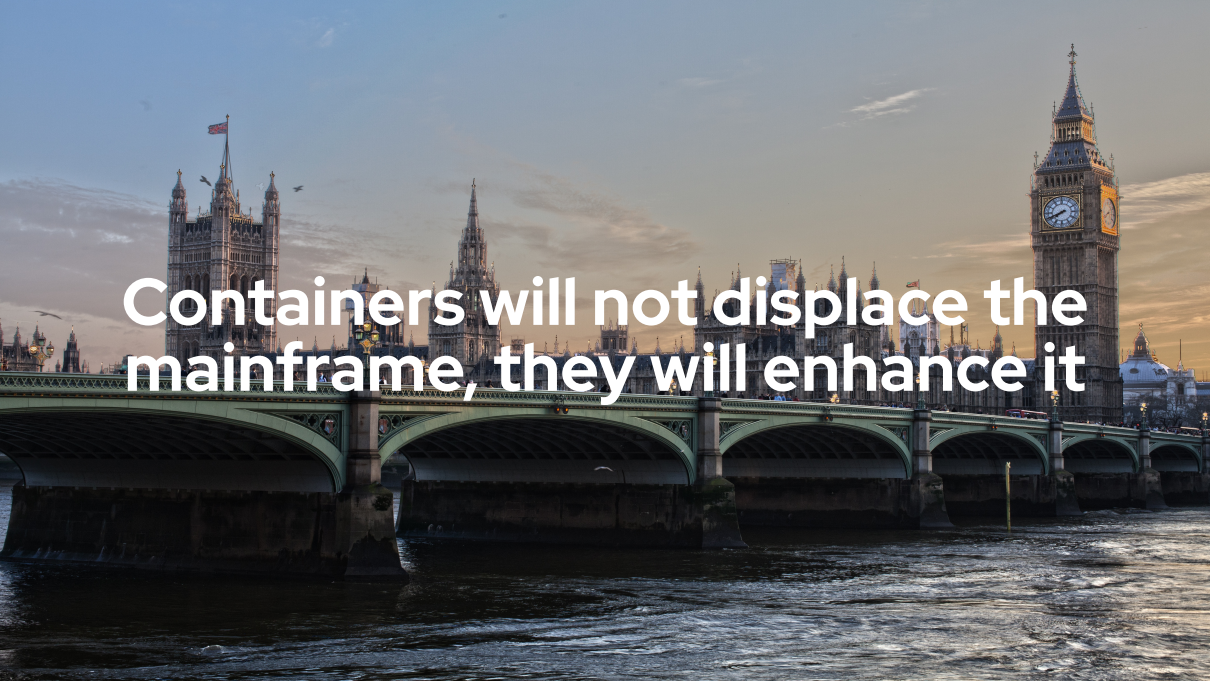
- Pro:
  - Full cluster availability
  - Full Flexibility placing applications and grow
  - Dedicated control nodes for High Availablity
- Cons:
  - Cluster IFL footprint highest. But will only slightly grow
  - Setup for Dev/Test env potentially oversized

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 ane 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

Please submit your session feedback!

- All done via the Whova App
- QR Code to the right to download the Whova App
- This session is DV