Red Hat Summit

Connect

Modernization strategies enabled by OpenShift Virtualization.

Lessons learned in the ATM sector.

Red Hat

Meet the Speaker

Paradigma

David Morales de Frías

Technical Product Manager

Modernization strategies and assessments.

MODERNIZATION STRATEGIES IN THE INDUSTRY (6 Rs)

	RETAIN	RETIRE	REHOST	REPLATFORM	REFACTOR	REPLACE
BUSINESS VALUE	LOW		MEDIUM	HIGH		
EFFORT	LOW		MEDIUM		HIGH	VERY HIGH

Do not touch

Legacy applications that do not have a business justification for a modernization effort due to decommissioning initiatives, technical debt, unsupported technologies…

.

Low Impact changes on deployment

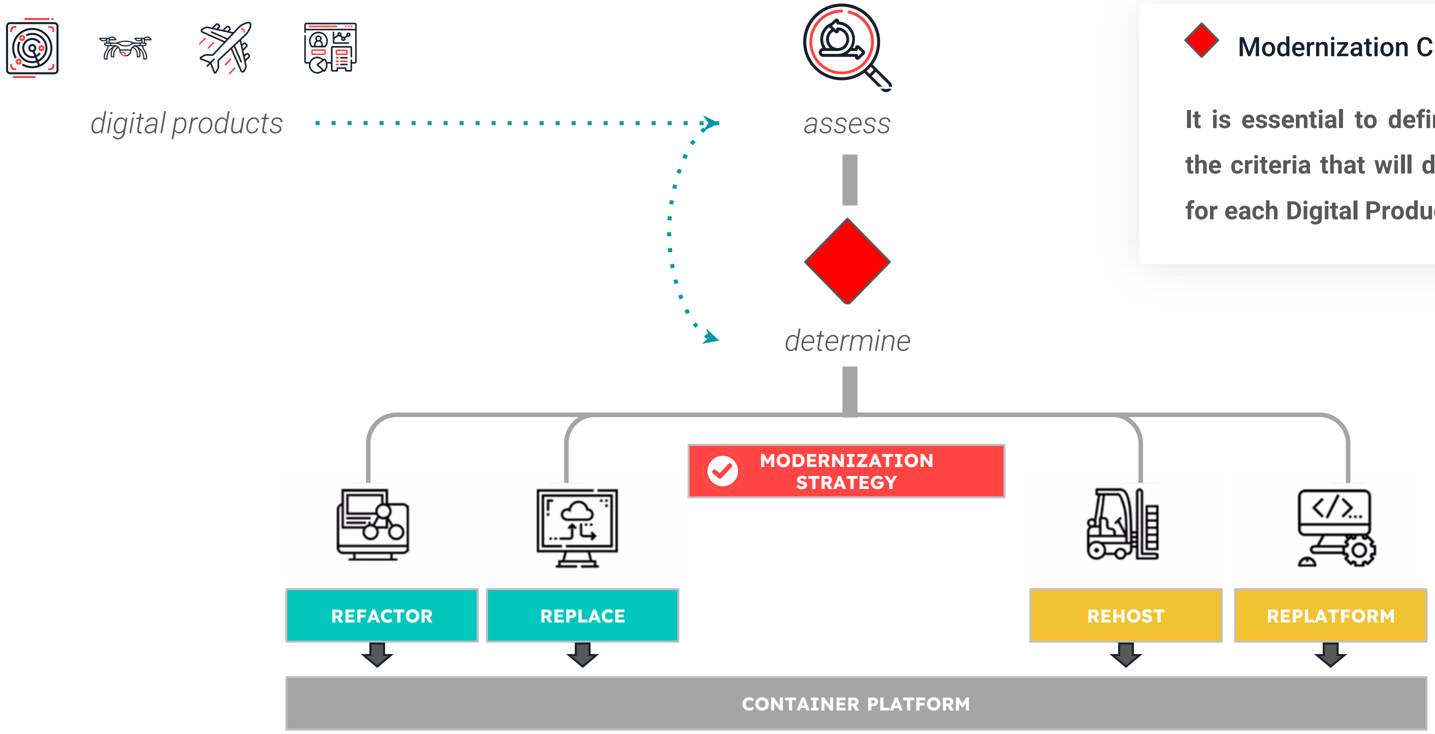
Active applications with a rigid architecture that can not be updated without a huge effort or high risk.

.

Core changes on the architecture

As part of the modernization process, the applications will be redesigned to adopt cloud native technologies and benefit from next-gen platforms.

CRITERIA TO SELECT THE RIGHT STRATEGY

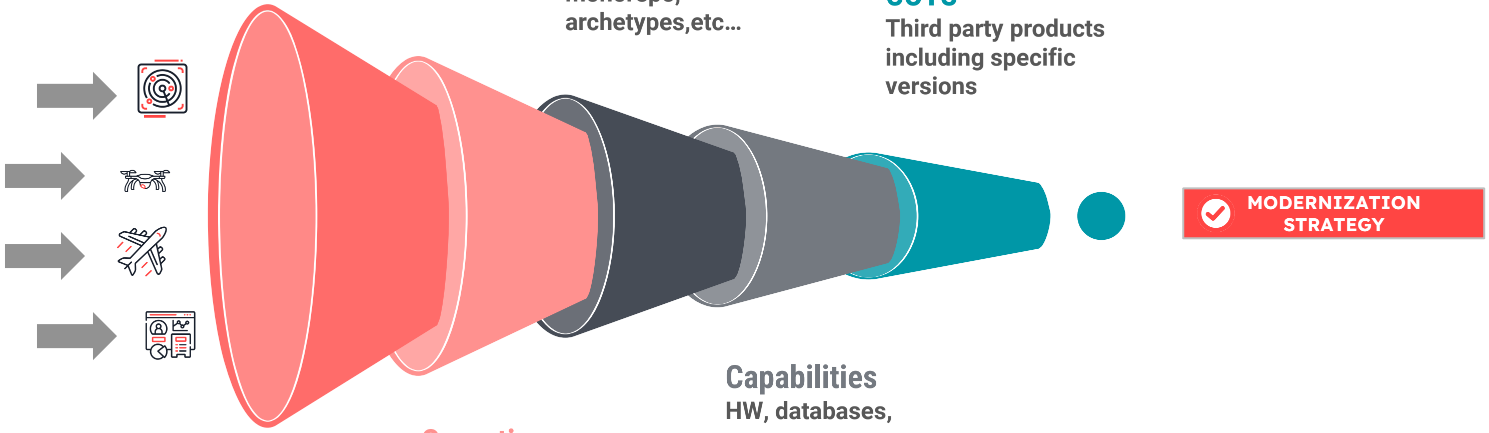


Modernization Criteria

It is essential to define a shared agreement on the criteria that will determine the best strategy for each Digital Product.

WHAT DIMENSIONS SHOULD BE CONSIDERED

digital products



Business
Stakeholders, Criticality and priority, legal or licensing requirements, level of investment

Operations
Deployment frequency, security, scalability, RTO, RPO, HA…

Architecture
Monolith or modular, monorepo, archetypes,etc…

Capabilities
HW, databases, message queues, streaming, caches, storage…

COTS
Third party products including specific versions

RED HAT: MIGRATION TOOLKIT FOR APPLICATIONS (MTA)

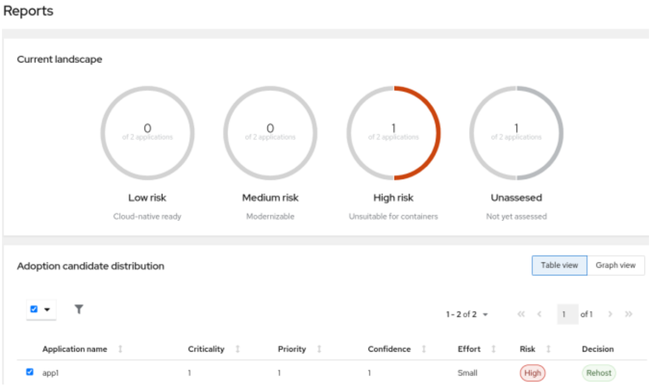
ASSESSMENT + TOOL

Tool to Evaluate cloud modernization readiness

With containerization-readiness, source-code analysis, and project management capabilities this tool can help your organization benefit from faster, safer legacy application modernization.

- **Risk assessment**
- **Decision**
- **Effort**
- **Priorities**

...



Reports

Current landscape

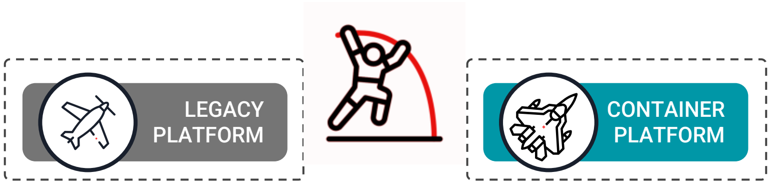
0 of 2 applications	0 of 2 applications	1 of 2 applications	1 of 2 applications
Low risk Cloud-native ready	Medium risk Modernizable	High risk Unsuitable for containers	Unassesed Not yet assessed

Adoption candidate distribution

☑	Application name	Criticality	Priority	Confidence	Effort	Risk	Decision
☑	app1	1	1	1	Small	High	Rehost

Platforms and Modernization strategies.

TWO DIFFERENT STRATEGIES FOR THE TRANSITION



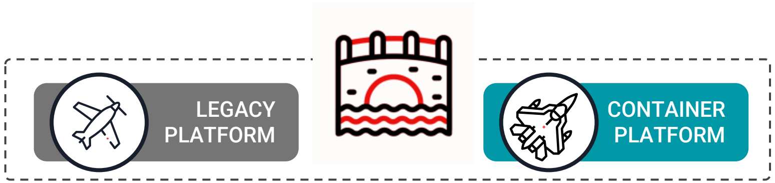
LEGACY PLATFORM

CONTAINER PLATFORM

REFACTOR
REPLACE

Isolated platforms.

Clear responsibilities and boundaries



LEGACY PLATFORM

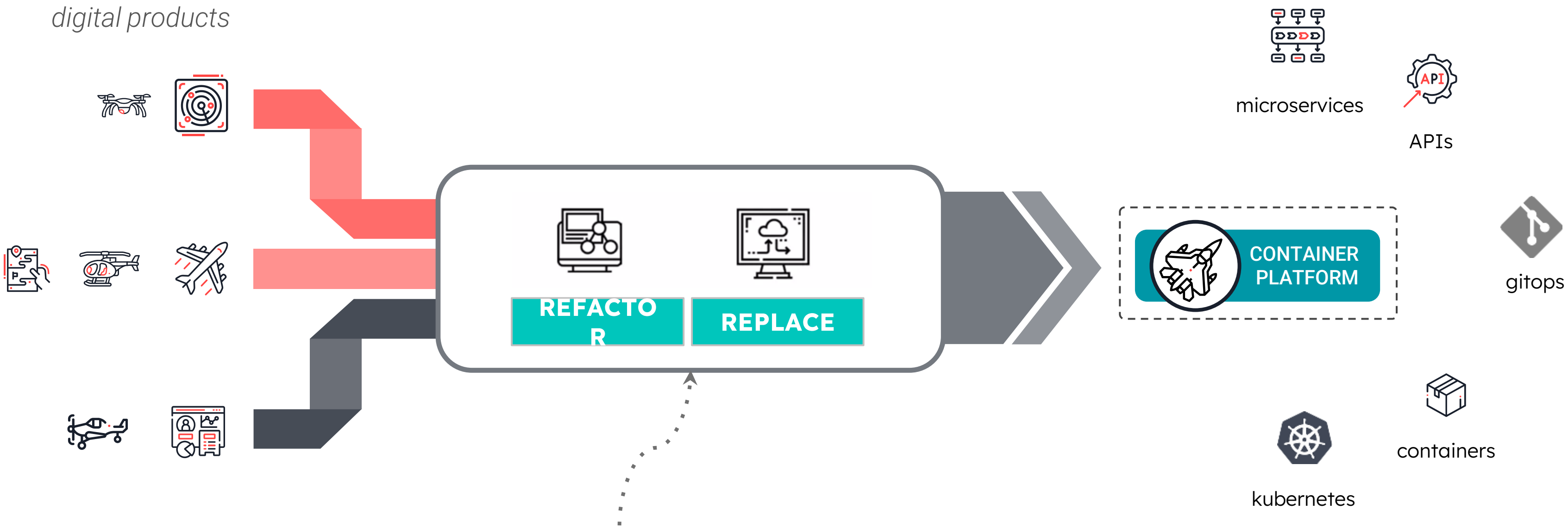
CONTAINER PLATFORM

PROGRESSIVE
MODERNIZATION

Integrated platforms
(1+1+bridge)

Responsibilities are shared.

ALL-IN-ONE TO CONTAINER PLATFORM

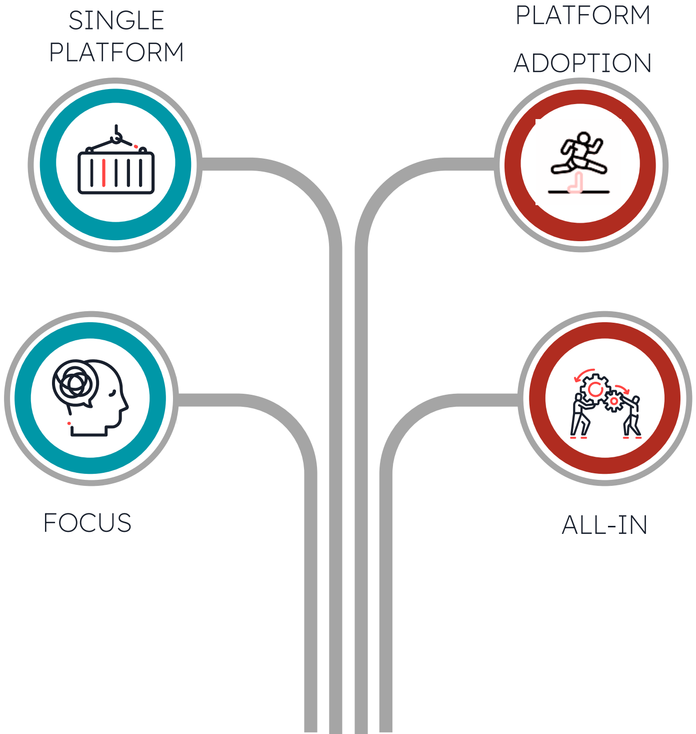


OpenShift focuses solely on containerization,
requiring applications to be
refactored or replaced to fit the platform.

BENEFITS & CHALLENGES

Dev & Ops focus on a single platform, with no overlap between legacy and modern workloads.

All efforts are invested into the modernization strategy



SINGLE PLATFORM

FOCUS

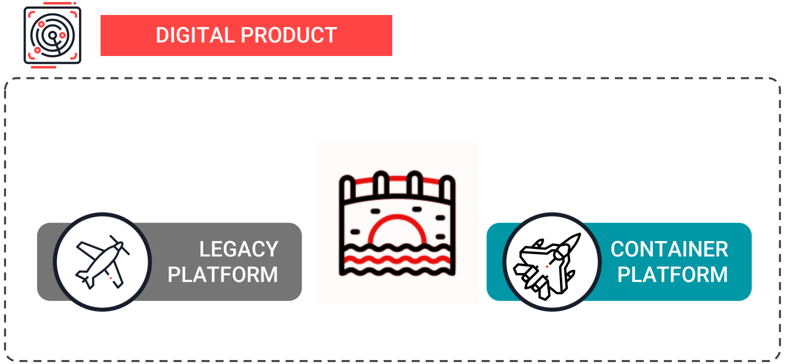
PLATFORM ADOPTION

ALL-IN

The platform adoption only takes place once the modernization process for each Digital Product is **complete,** delaying modern Ops.

Digital Products modernization require a **large investment** in time, resources, and expertise.

PROGRESSIVE MODERNIZATION



DIGITAL PRODUCT

LEGACY PLATFORM

CONTAINER PLATFORM

Modernization for a Digital Product will integrate legacy and new services over time until it is complete.

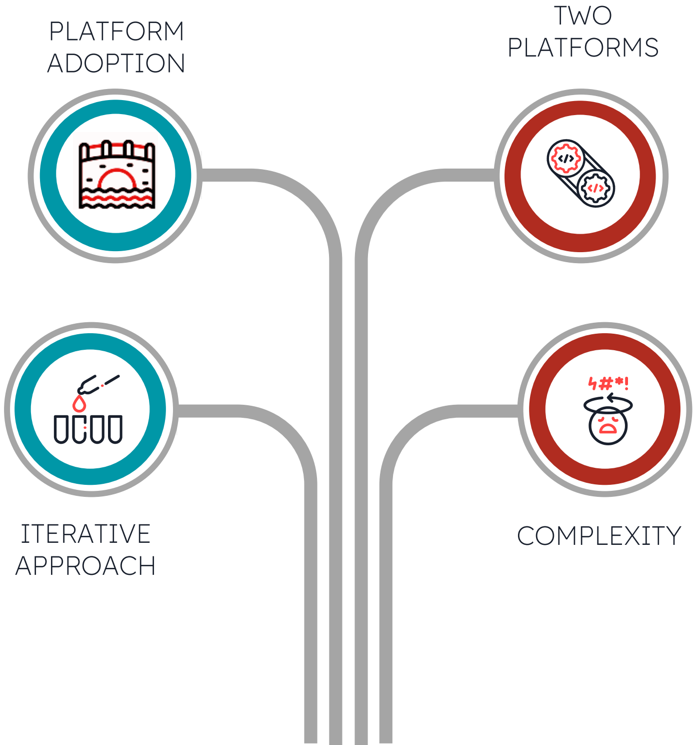
Certain design patterns and techniques enable a progressive modernization:

- ✓ Strangler Pattern
- ✓ Legacy Apification
- ✓ CDC and event driven
- ✓ …

BENEFITS & CHALLENGES

Platform Adoption takes place early but only for migrated services.

Learn and adjust over time mitigating the risks of a Big Bang.



PLATFORM ADOPTION

TWO PLATFORMS

ITERATIVE APPROACH

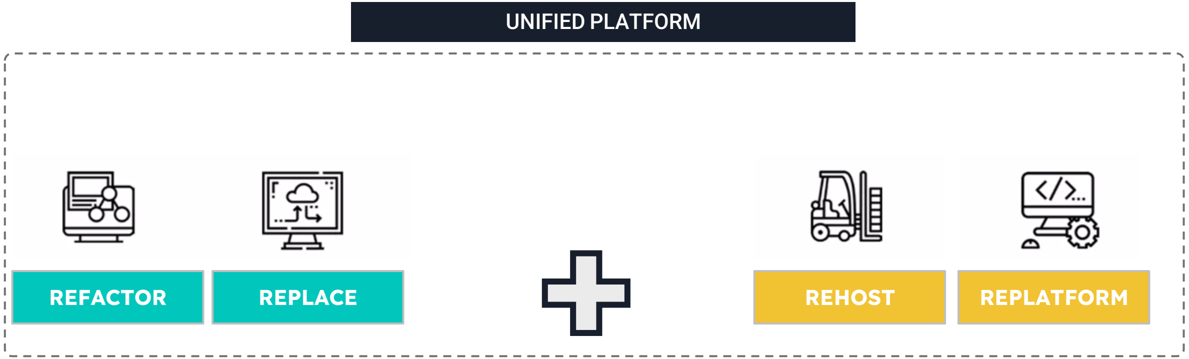
COMPLEXITY

Dev & Ops must manage two platforms and the bridge between them (networking, f.e)

Data and deployment must be synced, NFR must be managed twice.

LOOKING FOR ALTERNATIVES

Is there a solution where different modernization strategies and velocities can be deployed on a single platform?



UNIFIED PLATFORM

REFACTOR

REPLACE

REHOST

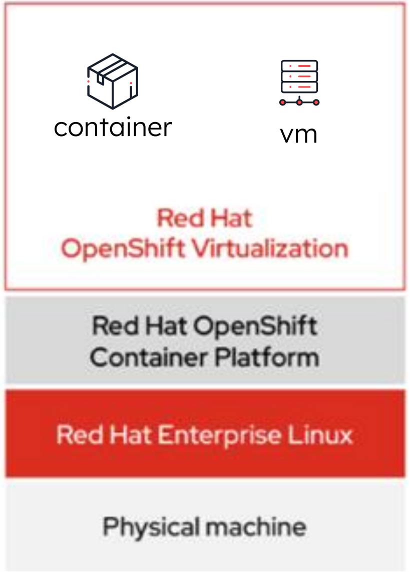
REPLATFORM

Introducing OpenShift Virtualization.

OPENSHIFT VIRTUALIZATION: A K8S OPERATOR

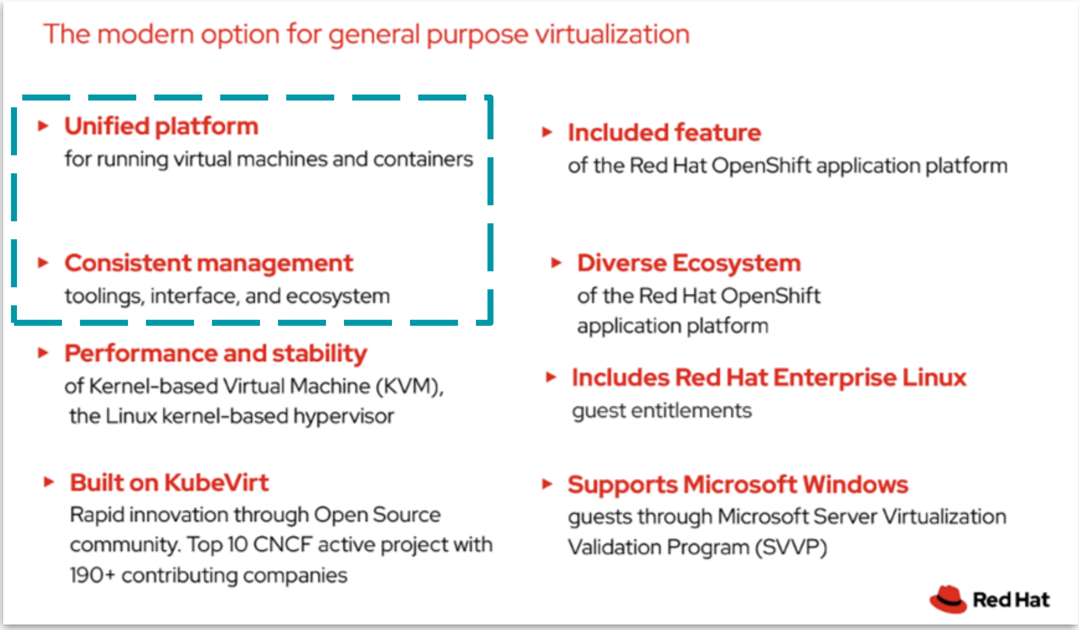
OpenShift Virtualization

A virtualization API and runtime for OpenShift, built on KubeVirt (open source), to run and manage virtual machines using a Kubernetes-native way



OPENSHIFT VIRTUALIZATION: BENEFITS

OpenShift Virtualization



The modern option for general purpose virtualization

- **Unified platform**
  for running virtual machines and containers
- **Consistent management**
  toolings, interface, and ecosystem
- **Performance and stability**
  of Kernel-based Virtual Machine (KVM), the Linux kernel-based hypervisor
- **Built on KubeVirt**
  Rapid innovation through Open Source community. Top 10 CNCF active project with 190+ contributing companies
- **Included feature**
  of the Red Hat OpenShift application platform
- **Diverse Ecosystem**
  of the Red Hat OpenShift application platform
- **Includes Red Hat Enterprise Linux**
  guest entitlements
- **Supports Microsoft Windows**
  guests through Microsoft Server Virtualization Validation Program (SVVP)

RED HAT: MIGRATION TOOLKIT FOR APPLICATIONS (MTA)

Supported Platforms

On Premise Bare Metal Servers

- Bare metal instances or servers offered by other cloud providers are not supported.
- OpenShift deployed on top of vmWare is not supported since it will lead to nested virtualization

AWS bare metal

- Amazon Web Services (AWS) bare-metal OpenShift Container Platform cluster.
- also supported on Red Hat OpenShift Service on AWS (ROSA) Classic clusters

Supported OS

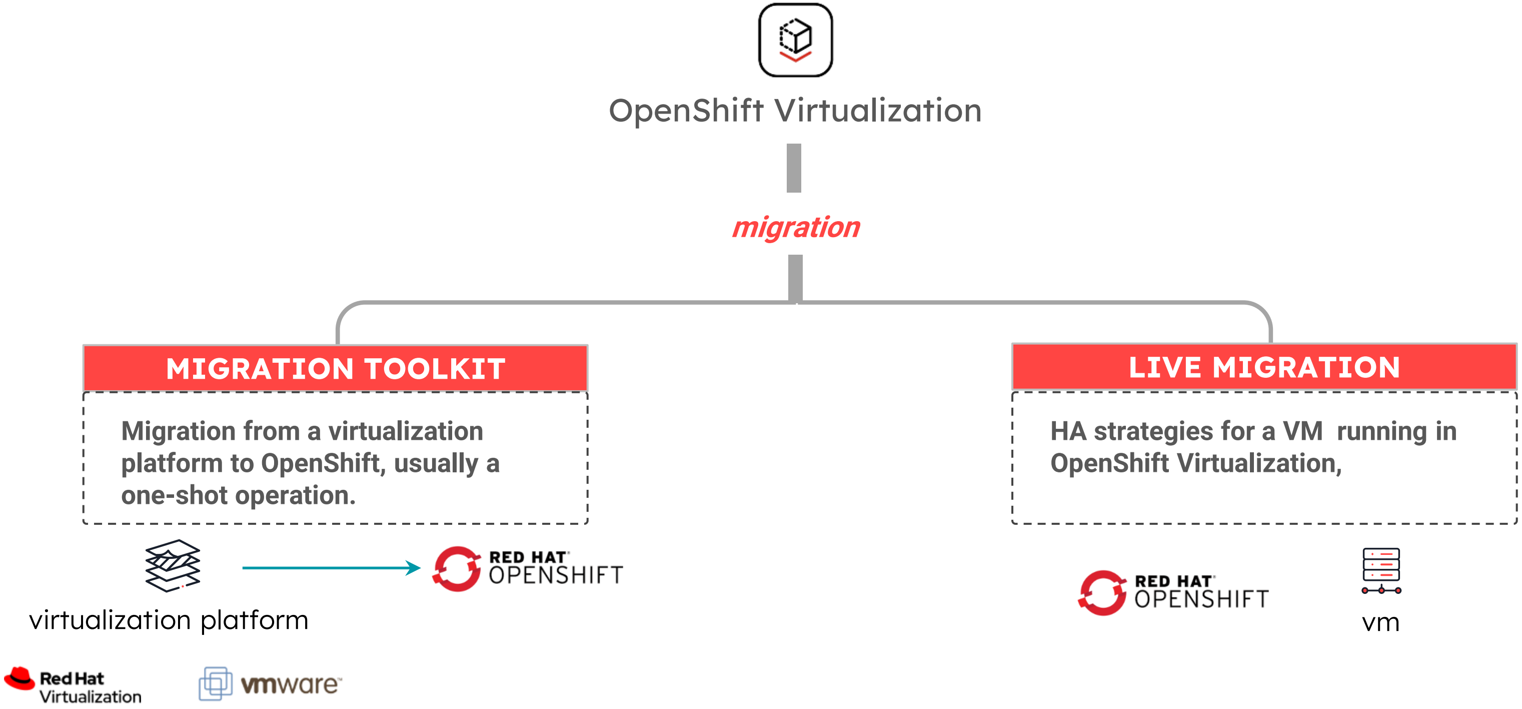
RHEL 7, 8, 9
Microsoft Windows 10, 11
Microsoft Server 2012 R2, 2016, 2019, 2022

OverHead

Memory overhead per infra node ≈ 150 MiB
Memory overhead per worker node ≈ 360 MiB

Additionally, OpenShift Virtualization environment resources require a total of 2179 MiB of RAM that is spread across all infrastructure nodes.

OPENSHIFT VIRTUALIZATION: MIGRATIONS



OpenShift Virtualization

migration

MIGRATION TOOLKIT

Migration from a virtualization platform to OpenShift, usually a one-shot operation.

LIVE MIGRATION

HA strategies for a VM running in OpenShift Virtualization,

MIGRATION TOOLKIT

virtualization platform

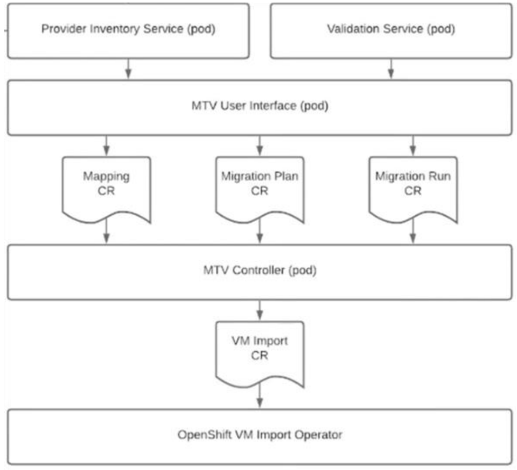
Red Hat Virtualization

vmware

MTV MIGRATION TOOLKIT

RED HAT OPENSHIFT

VIRTUAL MACHINE



A K8S operator to migrate workloads

The MTV ToolKit can help to migrate tens of VMs from a virtualization platform to OpenShift, creating the CRDs and mapping storage and networking between platforms.

OPENSHIFT VIRTUALIZATION: VMS AS PODS

VIRTUAL MACHINE CRD

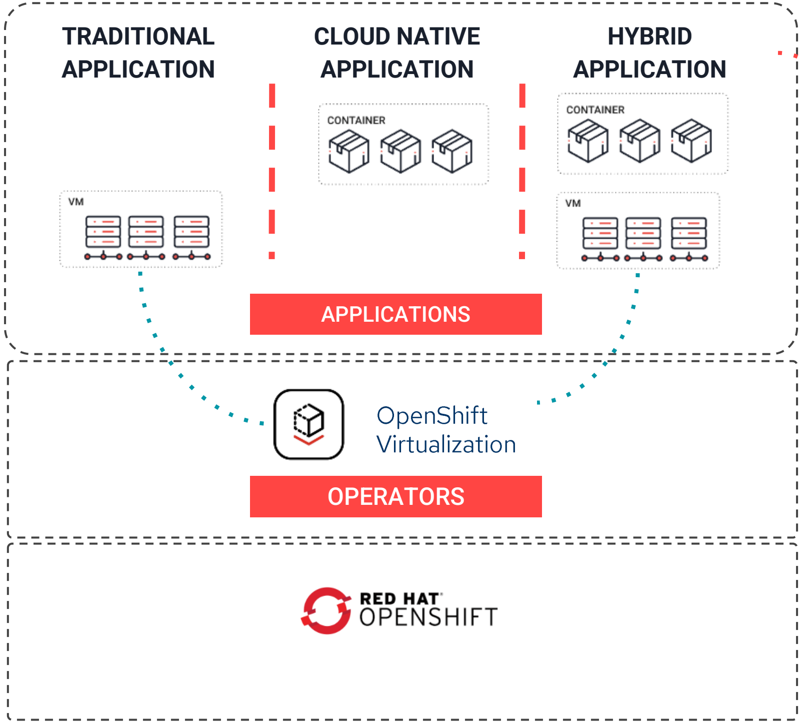
```
apiVersion: kubevirt.io/v1alpha3
kind: VirtualMachine
metadata:
  name: my-vm
spec:
    spec:
      domain:
        devices:
          disks:
          - disk:
              bus: virtio
            name: bootstrapdisk
        resources:
          requests:
            memory: 1G
      volumes:
      - name: bootstrapdisk
        persistentVolumeClaim:
          claimName: my-service-disk
```

OpenShift Virtualization expands the Kubernetes capacities to manage Virtual Machines, as a first level component.

Virtual Machines are provisioned in a declarative way, claiming storage or other infra resources, just like any other component in the cluster.

A new modernization approach.

A NEW MODERNIZATION APPROACH

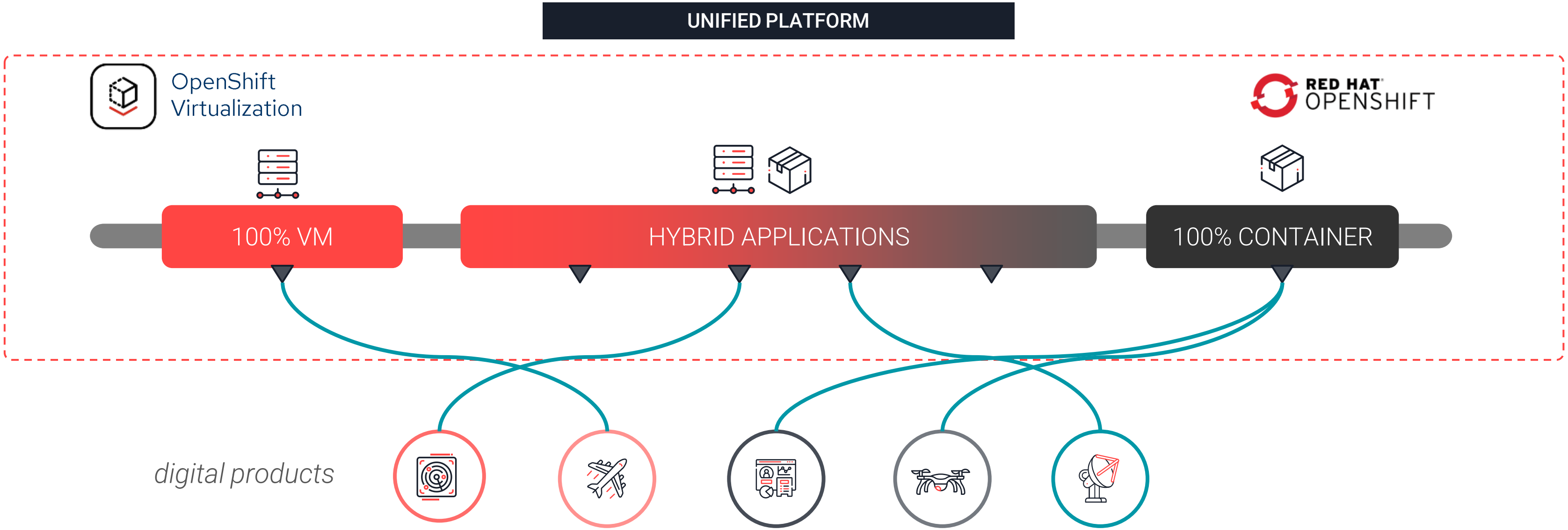


OpenShift Virtualization can support traditional applications where the workload is fully virtualized (escaping the vmWare license trap, f.e).

But the actual driver is a modernization strategy where applications can take advantage of both containers and VMs under a single platform.

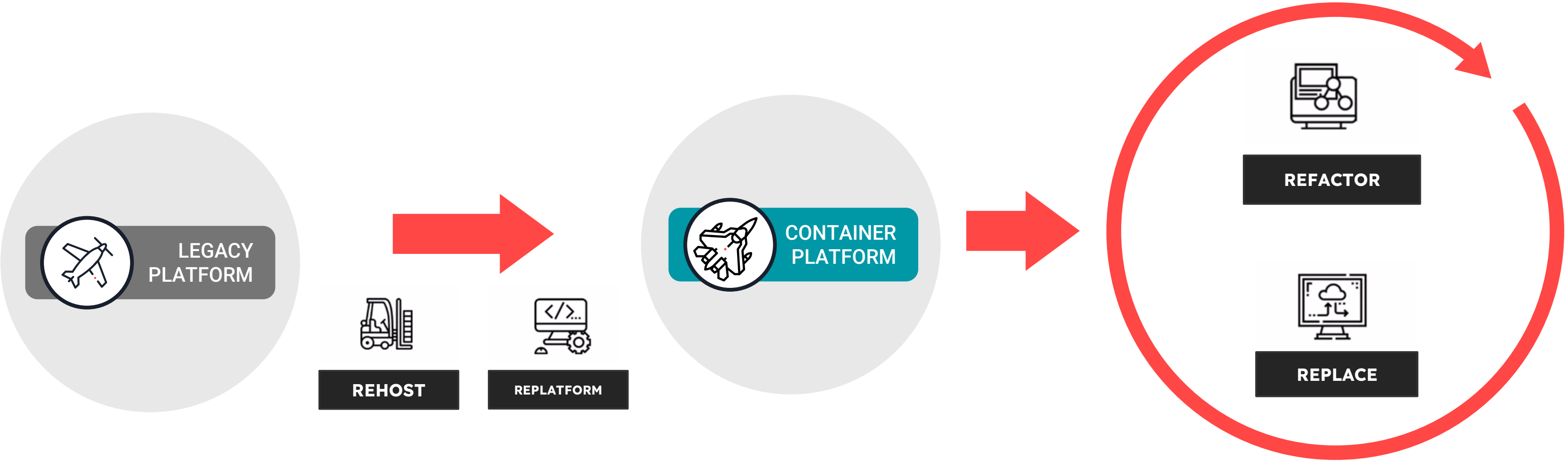
This approach enables the ability to deploy **Hybrid Applications** creating room for **different modernization velocities**.

FLEXIBLE MODERNIZATION PATHS IN ONE SINGLE PLATFORM



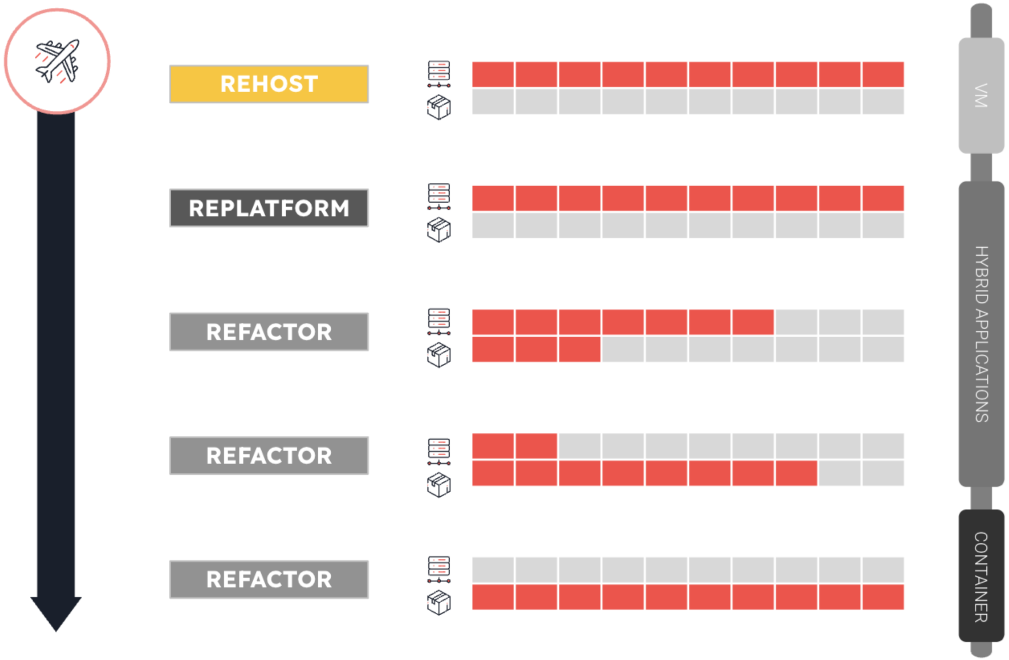
Leveraging a unified platform, it is now possible to on-board applications as early as possible and select the best modernization journey for each Digital Product.

MAXIMIZING PLATFORM ADOPTION UNDER SINGLE/MODERN OPS



Transfer all kind of workloads to the unified platform as early as possible to benefit from new Ops and enable adaptive modernization strategies.

ADAPTATIVE JOURNEY FOR A DIGITAL PRODUCT

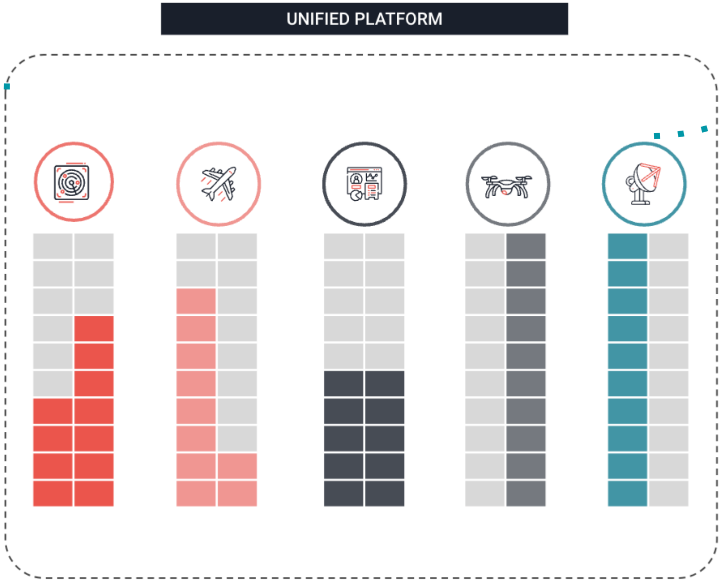


OpenShift Virtualization enables a flexible and adaptive journey towards the full-container solution, that can be tackled in different phases over time.

DIFFERENT MODERNIZATION STAGES COEXIST

OPS

Compact deployment package on a single platform leveraging a **unified and modern approach for Ops** (gitops, networking, observability, backups, HA, DR…) regardless of the modernization stage.



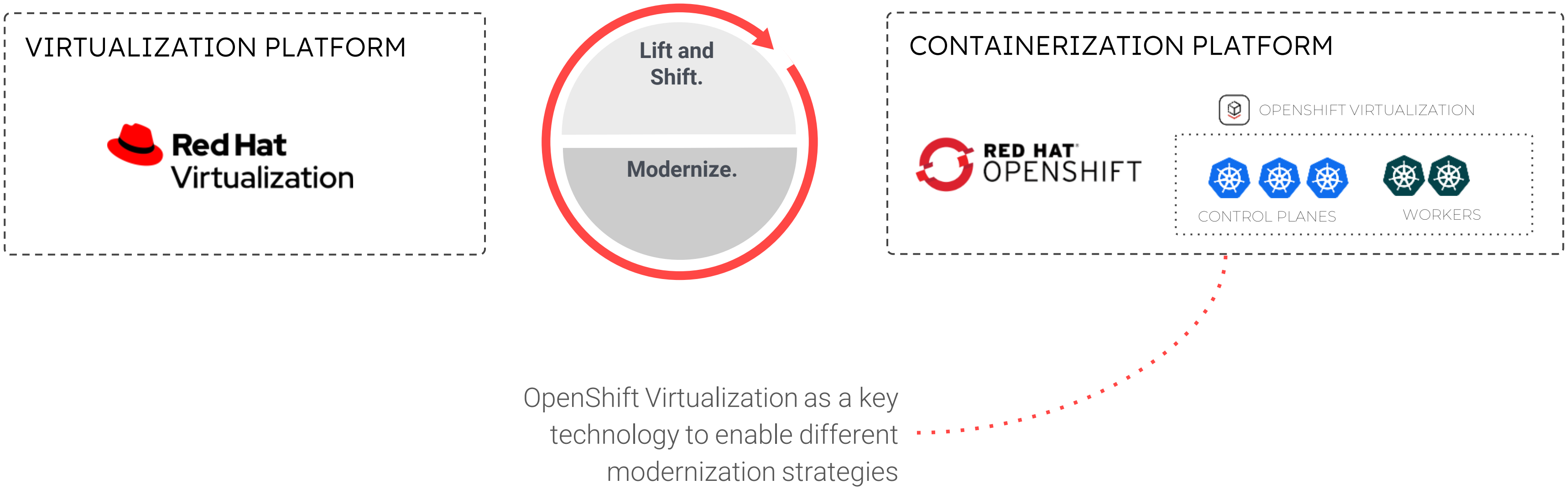
DEV

Under a single platform, different Digital Products will evolve independently creating room for development teams to modernize **without hard constraints or barriers.**

Platform Engineering
ATM - Openshift Virtualization
Use cases

Red Hat

FROM VMS TO A HYBRID PLATFORM.



VIRTUALIZATION PLATFORM

Red Hat Virtualization

Lift and Shift.

Modernize.

CONTAINERIZATION PLATFORM

RED HAT OPENSHIFT

OPENSHIFT VIRTUALIZATION

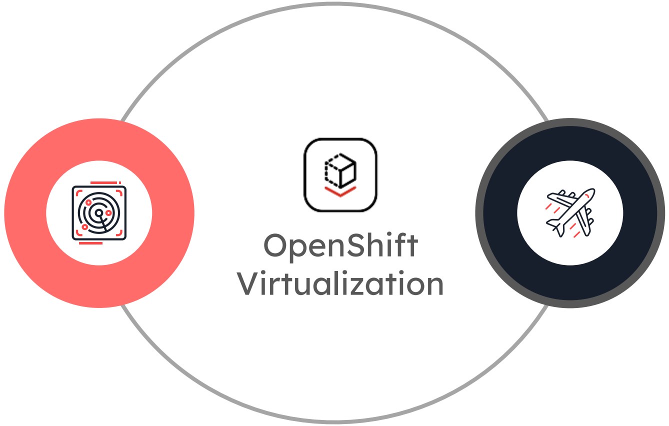
CONTROL PLANES

WORKERS

OpenShift Virtualization as a key technology to enable different modernization strategies

TWO TARGET SCENARIOS

OpenShift Virtualization as a **modernization strategy** for hybrid applications (containers, VMs)

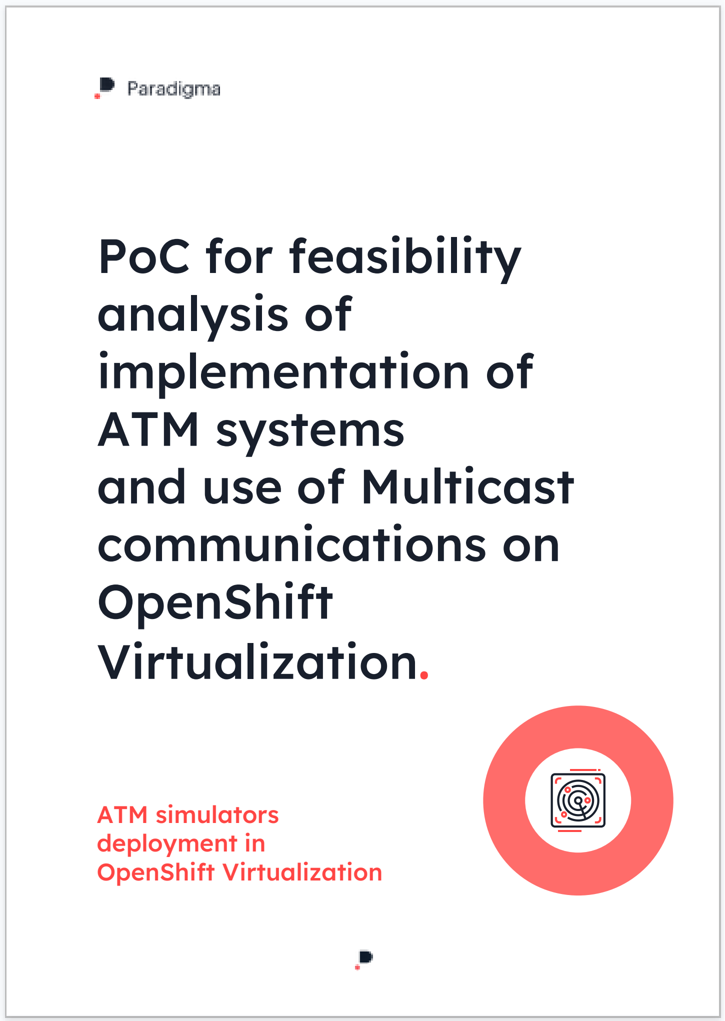


OpenShift Virtualization as a **virtualization platform** for virtualized applications

AIR TRAFFIC SYSTEM SIMULATOR.

GOAL

Assessing OpenShift Virtualization as a platform for Virtual Machine Hosting and Multicast Traffic in fully virtualized Air Traffic Management Simulations



Paradigma

PoC for feasibility analysis of implementation of ATM systems and use of Multicast communications on OpenShift Virtualization.

ATM simulators deployment in OpenShift Virtualization

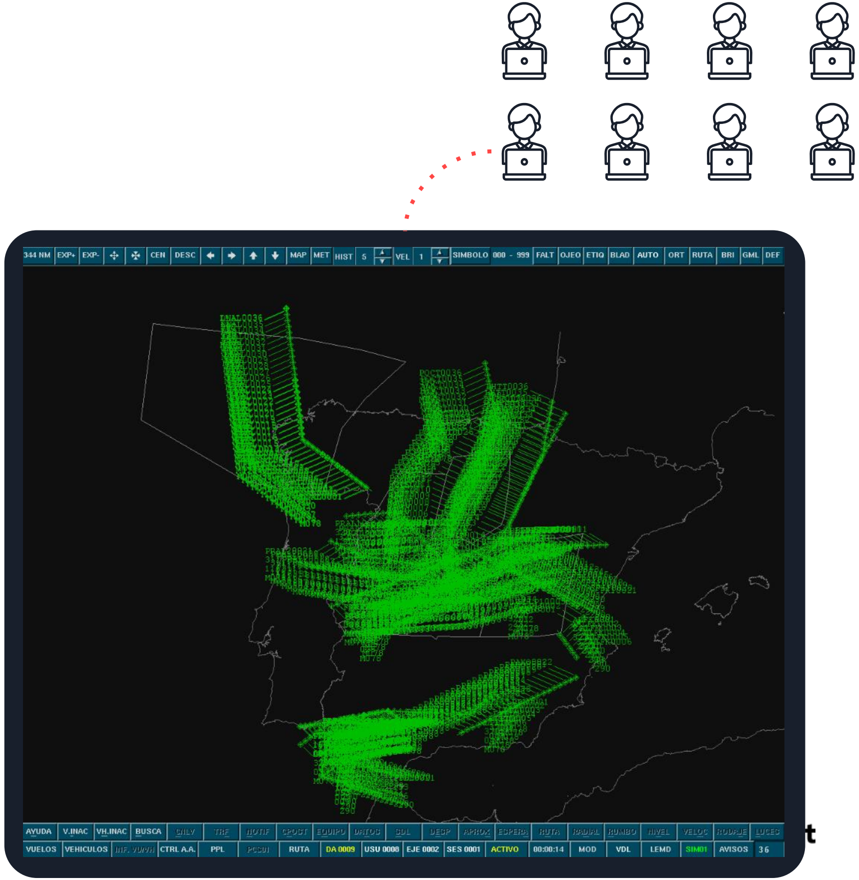
PRODUCT DESCRIPTION

USE CASES

- **Training** for new air traffic controllers until they are ready to manage live traffic.
- **Testing** new features or product versions to ensure they are safe for real-world operations.

SCENARIOS

- Each 'position' or seat for a controller **simulates** realistic air traffic scenarios in a controlled environment.
- The positions are fully **isolated** to prevent any collisions between controllers during training or testing.

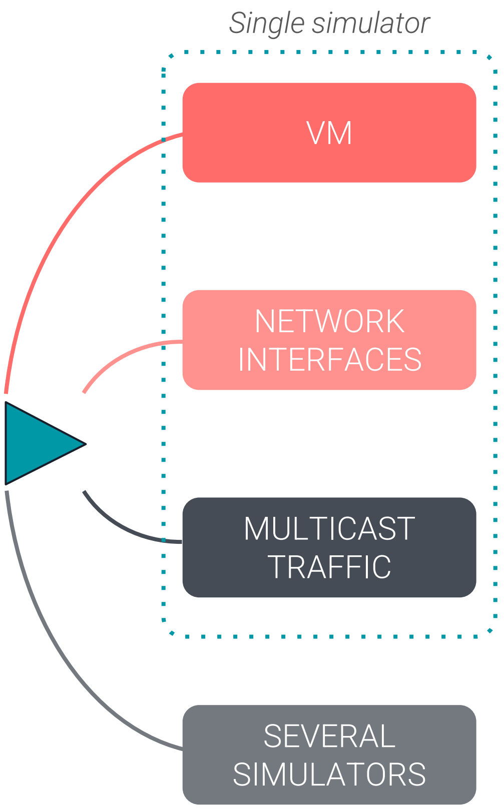


TESTED USE CASES

HYPOTHESIS

Can OpenShift Virtualization effectively run Air Traffic Simulators?

- ✓ Performance, Reliability
- ✓ Isolation
- ✓ Infrastructure overhead



Deploying VMs with RHEL6, a not supported OS.

VLANs are heavily used to segregate the different types of communication (e.g., control traffic, radar data).

Simulators generate large amounts of TCP, UDP multicast and UDP unicast traffic, vital for synchronizing different systems and maintaining accurate, real-time simulations.

Running several simulators in the same cluster to enable multiple seats or positions simultaneously.

HYBRID DIGITAL PRODUCT IN K8S.

GOAL

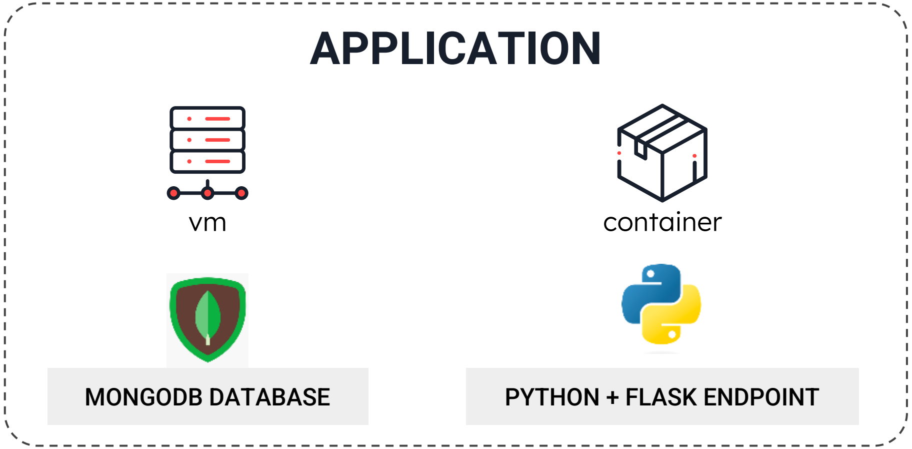
Assessing OpenShift Virtualization
as a strategy for
hybrid applications
combining containers and VMs
in a single deployment.

Paradigma

Deployment of hybrid Digital Products leveraging VMs, containers, K8S native resources and K8S operators.

Openshift and
OpenShift Virtualization
Capabilities
for application modernization

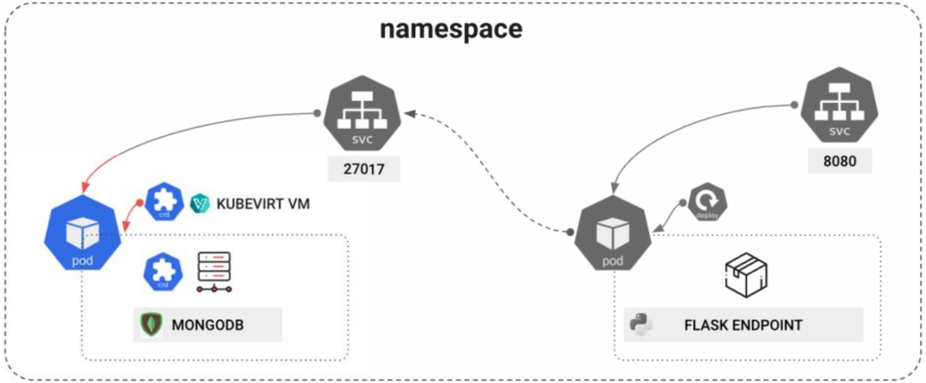
PROBLEM STATEMENT



A virtualized MongoDB database (v4.2.6) stores data read by a Python application deployed as a container.

The mongo database must be deployed as a replicaset with TLS enabled ensuring that only the Python application can connect to the database.

DEPLOYING COTS AS VMs

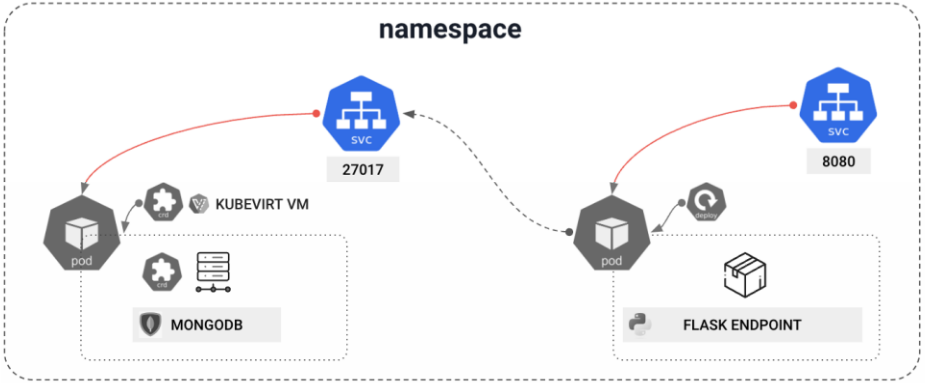


MongoDB will be deployed using VMs.

The OpenShift Virtualization operator (KubeVirt) provides a new “VirtualMachine” CRD to provision virtual machines in the Kubernetes Cluster.

This new K8S resource can be combined with other K8S native resources or even other K8S resources provided by other operators.

COMMUNICATION BETWEEN CONTAINERS AND VMS

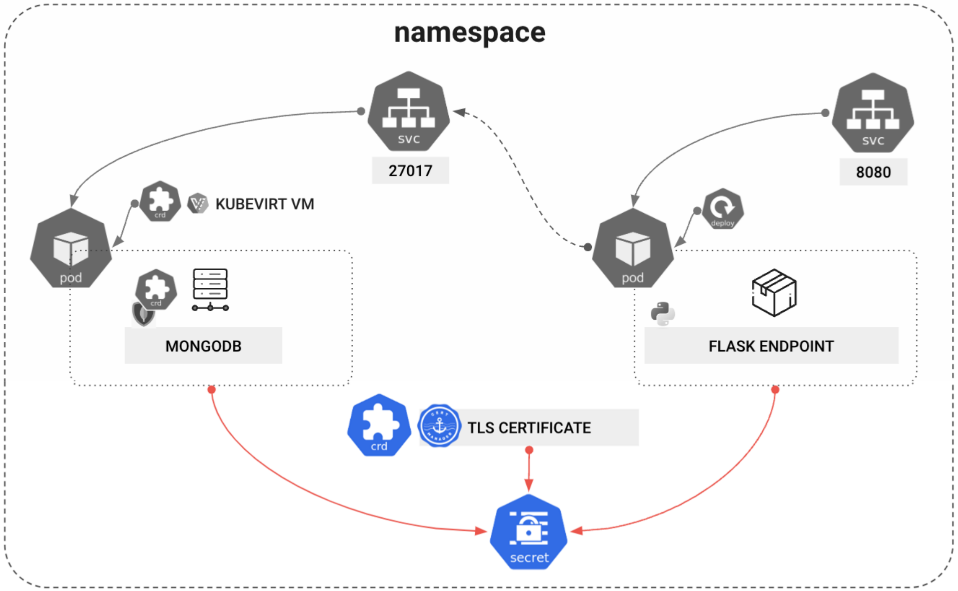


A Service will expose the Python endpoint to be consumed.

A different Service will allow the Python endpoint (Pod) to communicate with the MongoDB database.

A Service can be configured bind the Service to a set of Pods, determining which pods will receive the traffic that the service manages.

COMBINING VMS WITH OTHER K8S OPERATORS

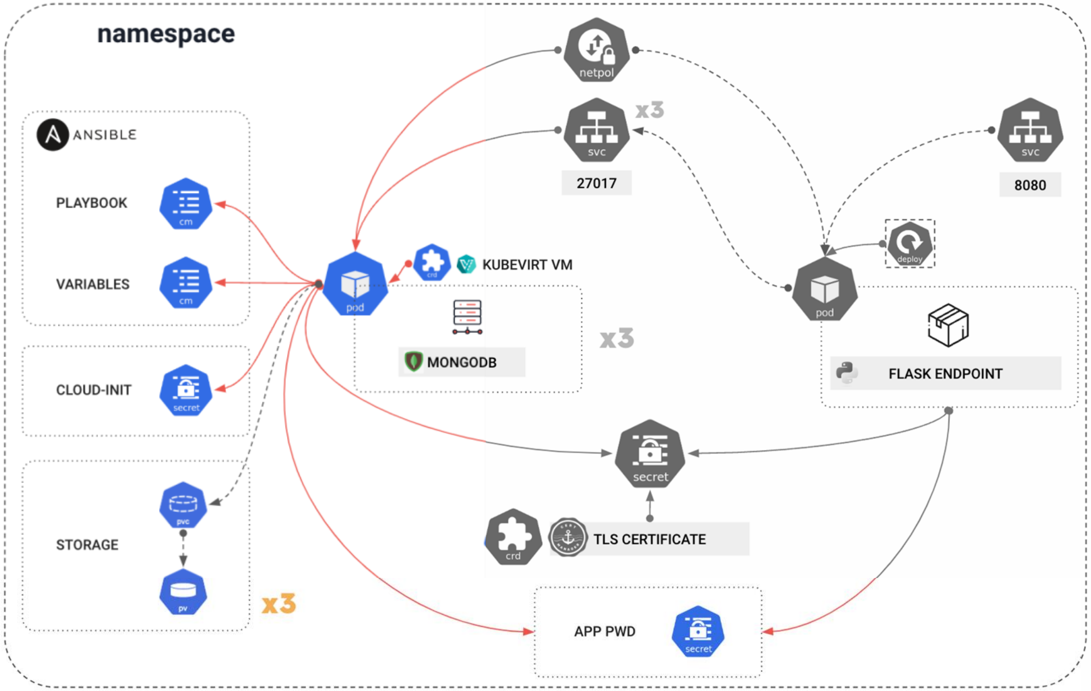


The cert-manager operator can automate the creation, renewal, and management of TLS certificates, ensuring secure connections for services without manual intervention.

Cert-manager will create a Secret that can be used to gather the certificate, private and public keys.

It is used to enable SSL in the MongoDB database and ensuring security in the connection from the endpoint.

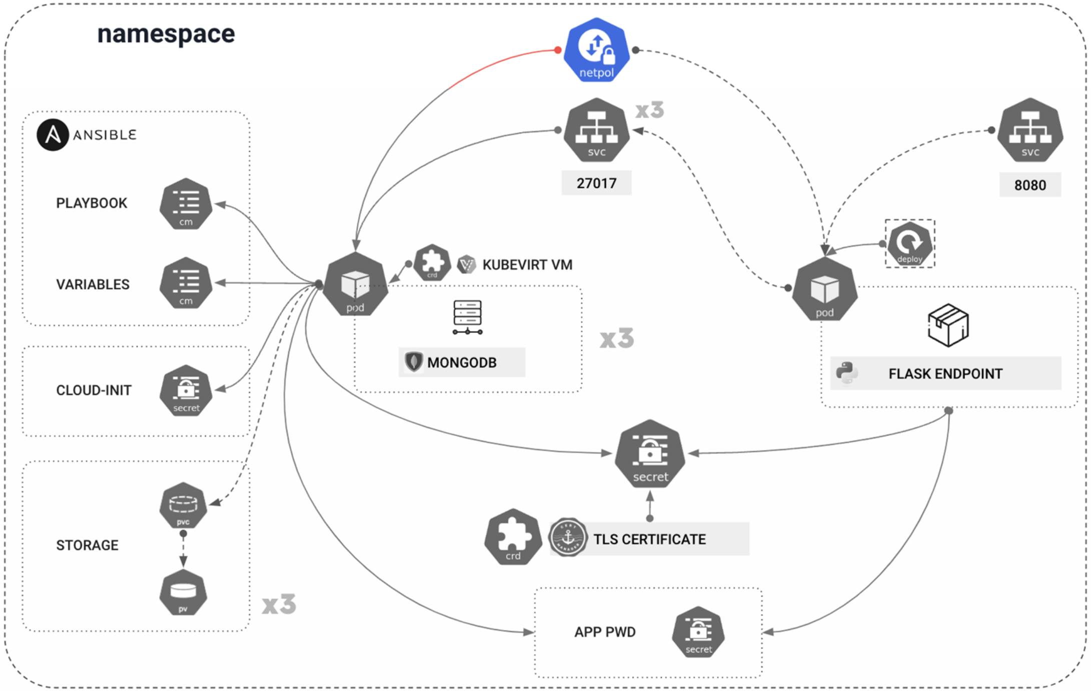
AUTOMATING AND ORCHESTRATING THE VMS PROVISION



To deploy a MongoDB replicaset with such a specific version, a combination of cloud-init and Ansible is required.

ConfigMaps and Secrets are used to push all the required provisioning assets to the Virtual Machine.

SECURING THE COMMUNICATION TO VMS.



A Network Policy is added to enforce the access rules.

It will prevent any other Pods in the same Namespace that don't meet the Ingress rule criteria from accessing the database.

Challenges & Lessons learned.

PREVENTING AND DISCOVERING BLOCKER ISSUES

01

CHALLENGE

Due to the limitations and constraints of OpenShift Virtualization, certain VMs may encounter issues running properly.

It is essential to anticipate blocking issues as early as possible

01 EARLY MIGRATION ASSESSMENT

02 CREATE TEST SCENARIOS

03 IDENTIFY INAPPROPRIATE USE CASES

DAY 2 CONSIDERATIONS FOR OPENSHIFT VIRTUALIZATION

02
CHALLENGE

While the Migration Toolkit helps accelerate the rehosting process to OpenShift Virtualization, there are key Day 2 challenges that need to be addressed:

Anticipate post-migration challenges while maintaining agility and reducing reliance on external virtualization platforms

01 VIRTUALIZATION PLATFORM DEPENDENCIES

02 GOLDEN IMAGES

03 FULL BOOTSTRAP PROCESS IN OPENSHIFT

NETWORKING SCENARIOS

03

CHALLENGE

Networking for VMs is more complex compared to containers, requiring advanced solutions for specific use cases and connectivity scenarios.

OpenShift Virtualization is equipped with powerful networking tools, while also addressing ongoing improvements for advanced scenarios

01 ADVANCED NETWORKING SCENARIOS

02 MULTUS AND NMSTATE

03 NETWORK OBSERVABILITY

ADOPTING DECLARATIVE MANAGEMENT FOR VMS

04
CHALLENGE

Although OpenShift Virtualization allows VM management via the OpenShift web console and SSH access, it is essential to shift towards a declarative management approach to ensure consistency.

The importance of transitioning to a code-first approach for VM management, aligning with modern infrastructure practices

01 MANAGE VMS AS CODE

02 LEVERAGE CLOUD-INIT & ANSIBLE

03 AVOID UIs OR DIRECT SSH ACCESS

AVOIDING MODERNIZATION STAGNATION

05
CHALLENGE

The inclusion of VMs helps ease the transition to OpenShift, but it requires a two-phase approach to modernization.

Focus on reducing reliance on VMs by migrating services to containers as part of the ongoing evolution, ensuring a path toward a cloud-native solution.

01 PHASE 1: DEPLOY ON THE NEW PLATFORM

02 PHASE 2: EVOLVE FROM VMS TO CONTAINERS

CHALLENGES OF NON-FUNCTIONAL REQUIREMENTS AND VMS

06
CHALLENGE

Deploying services on VMs introduces complexities in meeting critical Non-Functional Requirements (NFRs), such as Security, High Availability, Disaster Recovery or Observability

A concerted effort is required to align VM-based services with OpenShift's native tools to ensure comprehensive coverage of NFRs without duplicating effort or compromising on performance and reliability.

01 LEVERAGING PLATFORM CAPABILITIES

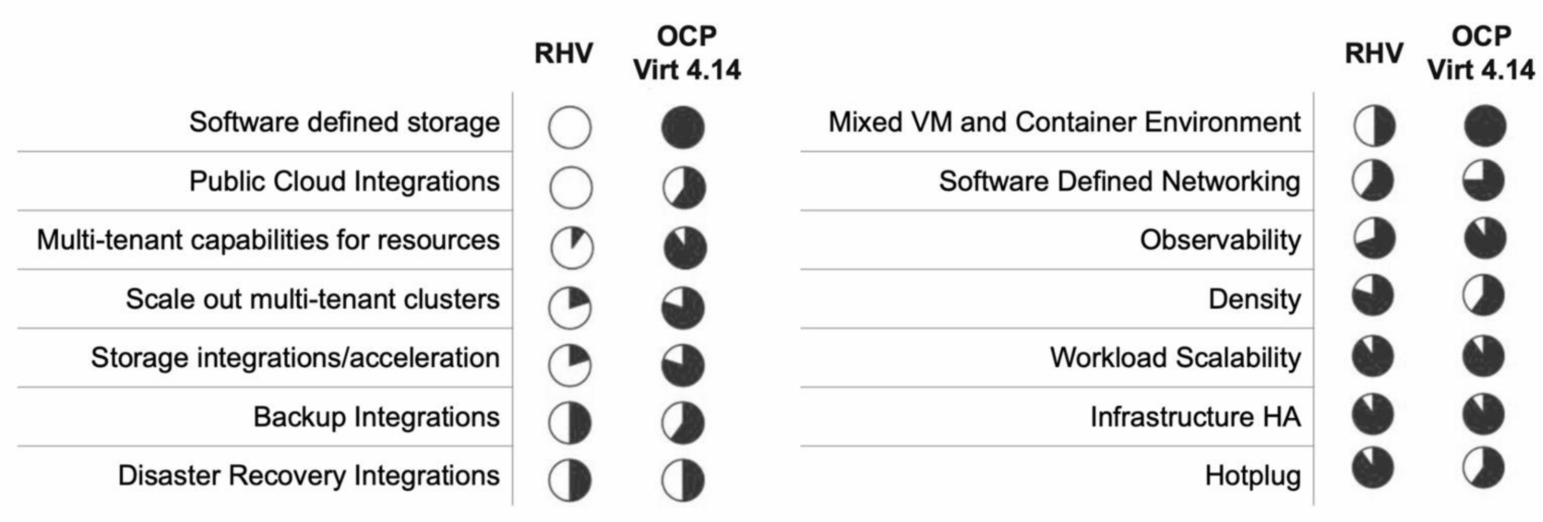
02 RISK OF INCOMPLETE NFR COVERAGE

FOLLOW THE ROADMAP

07

CHALLENGE

Follow closely the product enhancements, OpenShift Virtualization is gaining a lot of traction due to the VmWare license trap, mainly.



	RHV	OCP Virt 4.14		RHV	OCP Virt 4.14
Software defined storage	○	●	Mixed VM and Container Environment	◑	●
Public Cloud Integrations	○	◑	Software Defined Networking	◑	◕
Multi-tenant capabilities for resources	◔	◕	Observability	◕	◕
Scale out multi-tenant clusters	◔	◕	Density	◕	◑
Storage integrations/acceleration	◔	◕	Workload Scalability	◕	◕
Backup Integrations	◑	◑	Infrastructure HA	◕	◕
Disaster Recovery Integrations	◑	◑	Hotplug	◕	◑

Red Hat

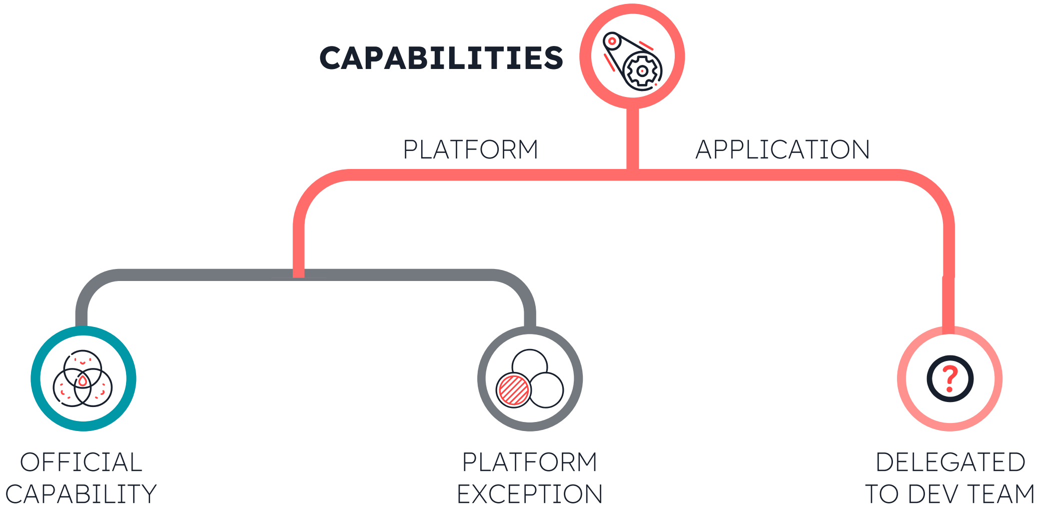
MATURITY AND RESPONSIBILITY MODEL

08
CHALLENGE

If your platform engineering initiative includes platform capabilities as a serve (databases, streaming, queues…) define a responsibility and maturity model.

OpenShift Virtualization enables teams to run core services on VMs overlapping the platform capabilities.

It is crucial to define a model to ensure that NFRs, security or service levels have clear owners.



Red Hat Summit

Connect

Thank you

in linkedin.com/company/red-hat

f facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat