THE KUBERNETES PLATFORM FOR BIG IDEAS

Ortwin Schneider
Middleware Solution Architect
Nov 2019

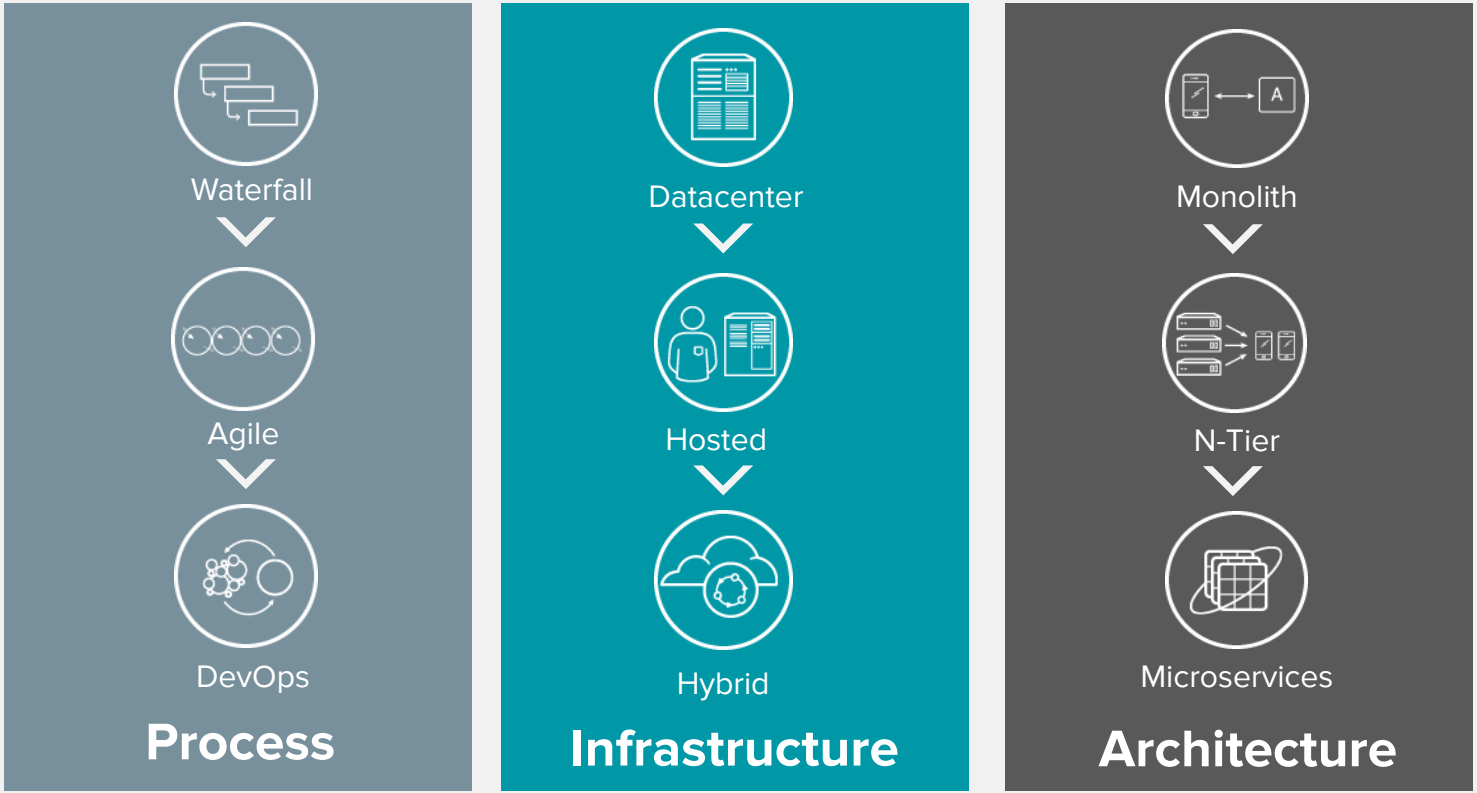

... so you want to do
containers and Kubernetes?


But Why?

"THE ONLY SUSTAINABLE ADVANTAGE YOU CAN HAVE OVER OTHERS IS AGILITY, THAT'S IT. BECAUSE NOTHING ELSE IS SUSTAINABLE."

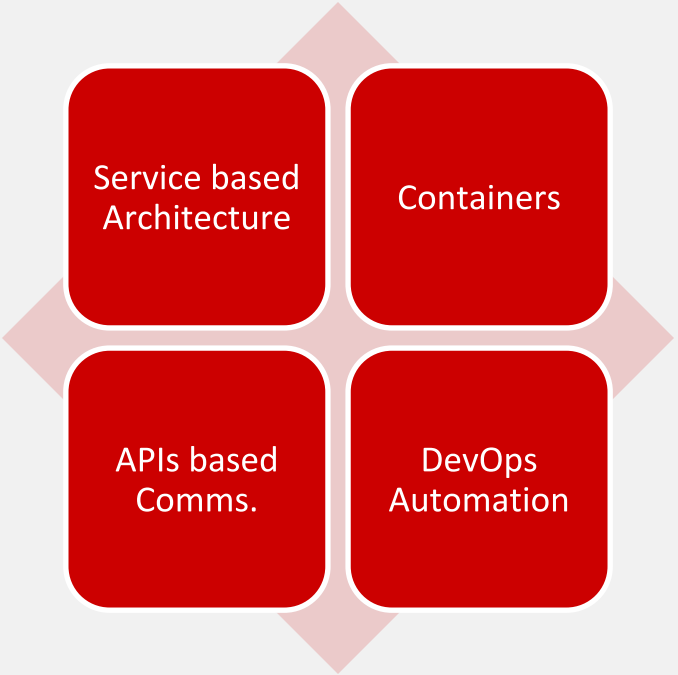
-Jeff Bezos, Founder Amazon

SOFTWARE DEVELOPMENT IS CHANGING



CLOUD NATIVE APPLICATION DEVELOPMENT

Red Hat Perspective



Cloud-native is an approach to building and running applications.

- Service based as in modular, loose coupling e.g Microservices
- Containers as in portable deployment and execution unit
- DevOps Automation as in using collaborative processes and automation from dev to deploy e.g CI/CD, Autoscale
- APIs based communication e.g no shared models, no direct linking

It is not about infrastructure like public, private or hybrid cloud

CLOUD NATIVE DEVELOPMENT IS ABOUT
RESPONDING TO CHANGE WITH

SPEED, RESILIENCY AND AGILITY

46X
More Deployment frequency

440X
Faster Lead time for changes

96X
Faster Mean time to recovery

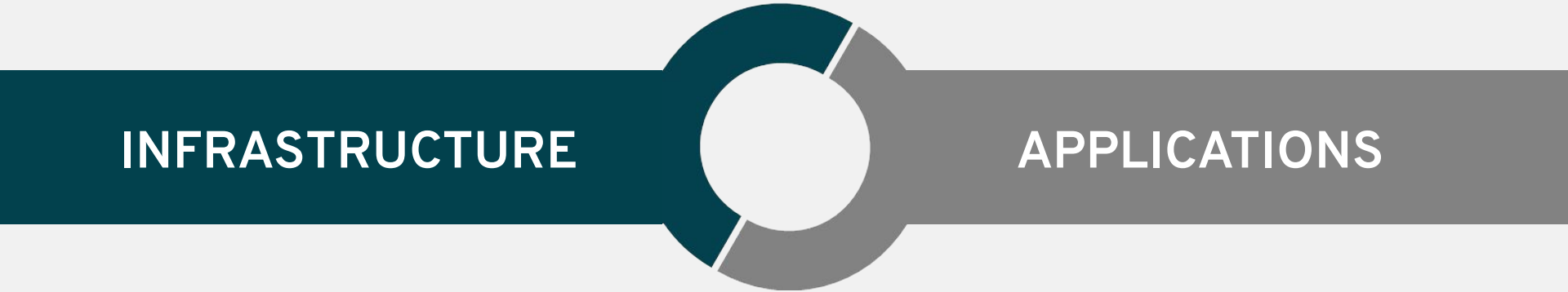
5X
Lower Change rate failure

2017 State of DevOps Report https://puppet.com/system/files/2017-09/2017-state-of-devops-report-puppet-dora.pdf


What are containers?

WHAT ARE CONTAINERS?

It Depends Who You Ask



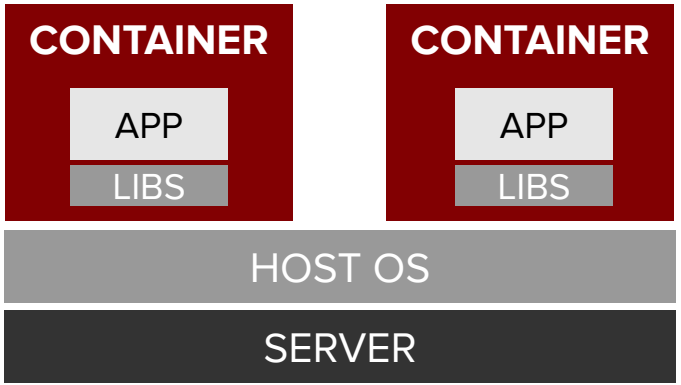
- Application processes on a shared kernel
- Simpler, lighter, and denser than VMs
- Portable across different environments

- Package apps with all dependencies
- Deploy to any environment in seconds
- Easily accessed and shared

Containers package applications with dependencies and isolate the runtime

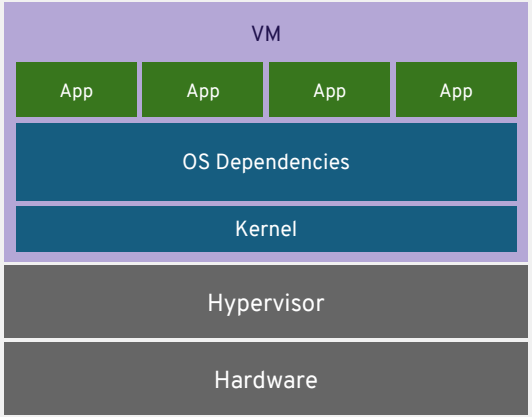
- Easy to deploy and portable across host systems
- Created from immutable, layered images
- Isolated from a host operating system.

  In RHEL, this is done through:
  - Control Groups (cgroups)
  - kernel namespaces
  - SELinux, sVirt, iptables
  - Docker



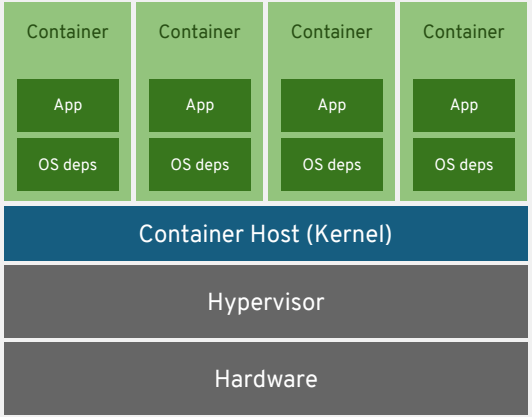
VIRTUAL MACHINES AND CONTAINERS

VIRTUAL MACHINES



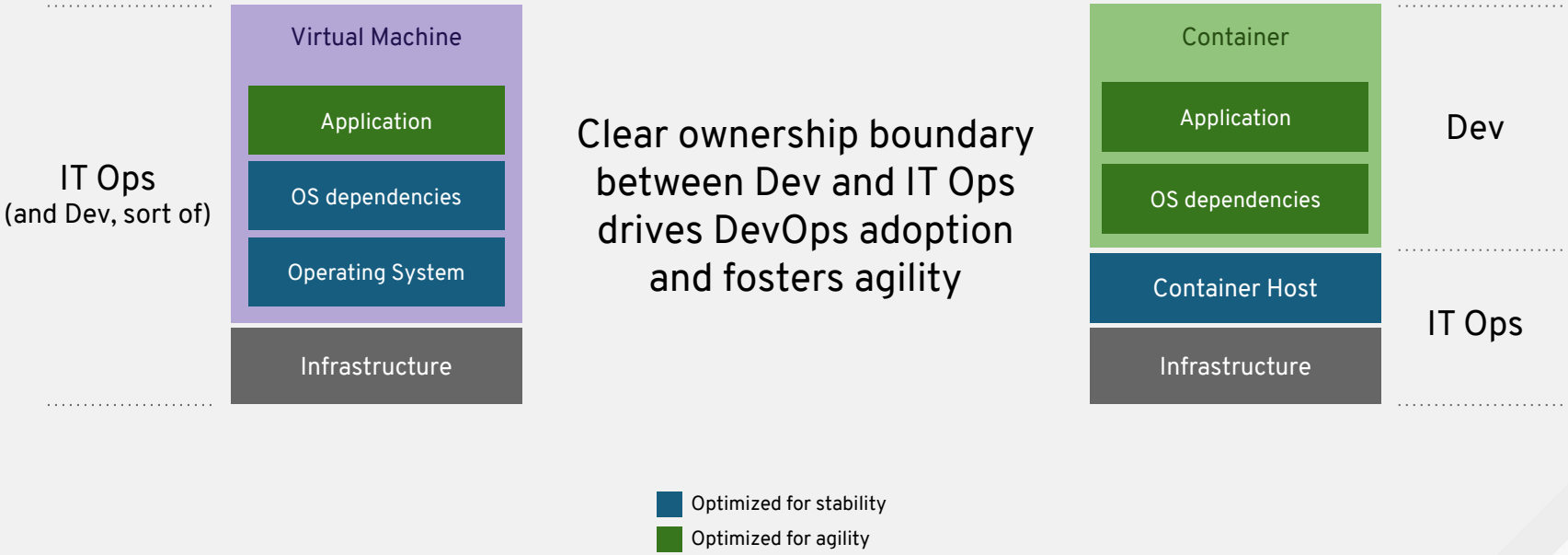
VM isolates the hardware

CONTAINERS



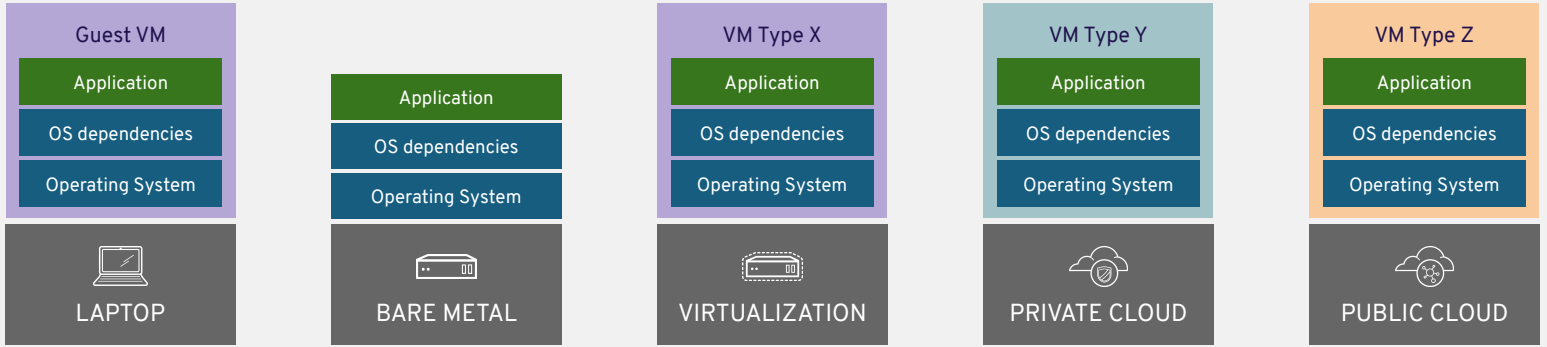
Container isolates the process

VIRTUAL MACHINES AND CONTAINERS



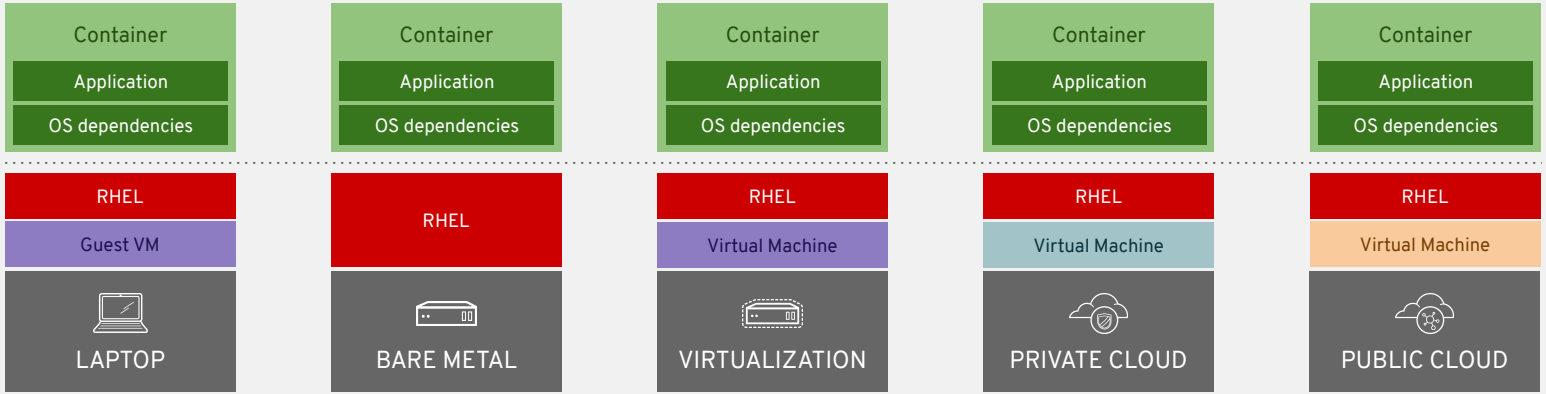
APPLICATION PORTABILITY WITH VM

Virtual machines are NOT portable across hypervisor and do NOT provide portable packaging for applications

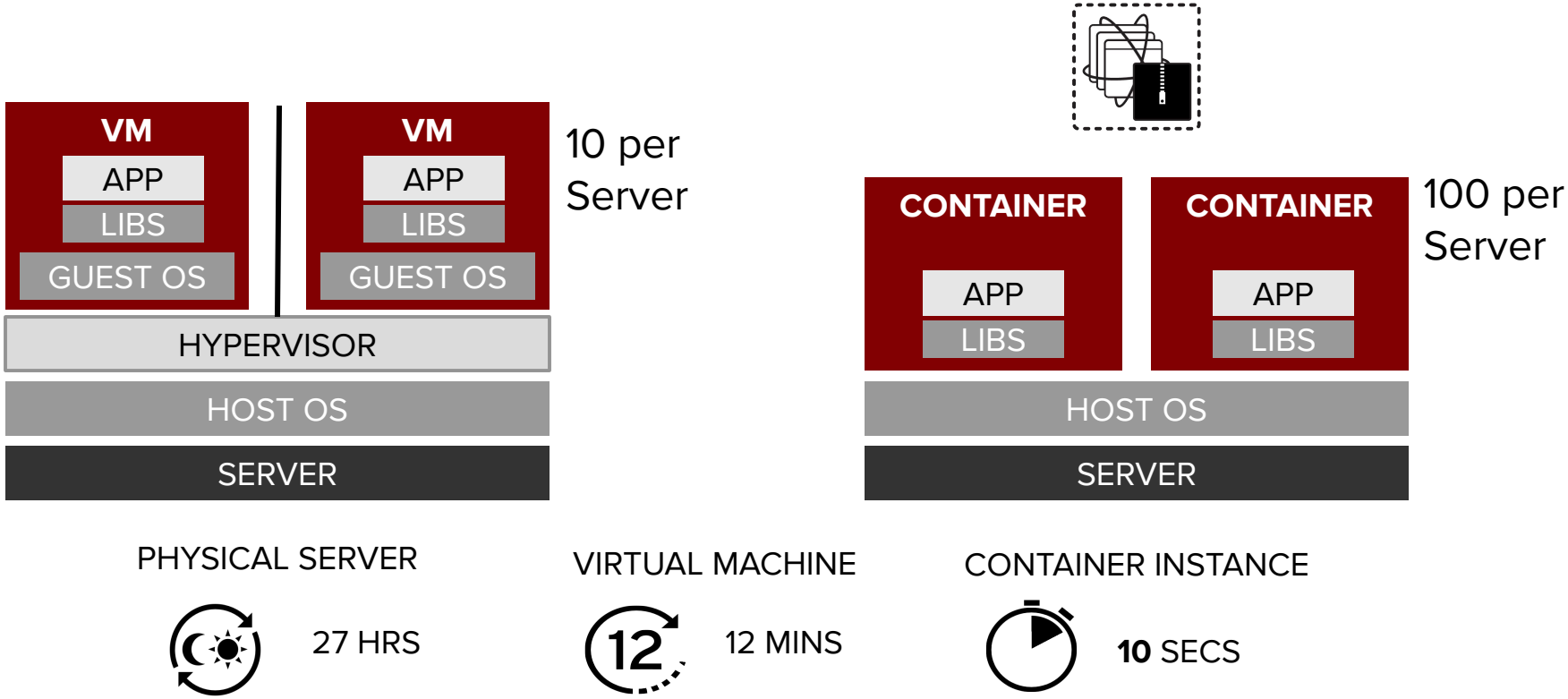


APPLICATION PORTABILITY WITH CONTAINERS

RHEL Containers + RHEL Host = Guaranteed Portability
Across Any Infrastructure



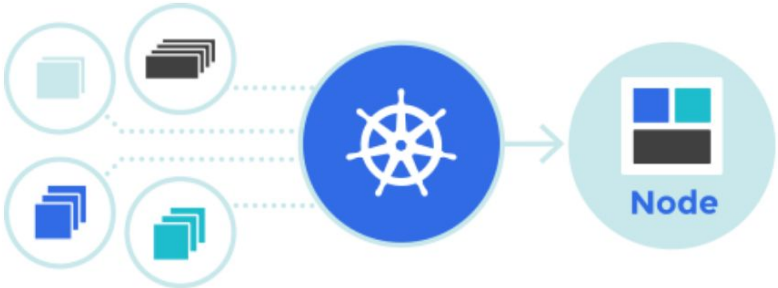
Containers provide high density and efficiency at the expense of isolation



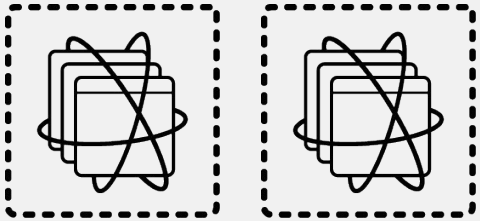

What is Kubernetes?

WHAT IS KUBERNETES?

- Orchestration of large amounts of running containers spread across a lot of hosts.
- “Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure.” [1]
- Open Sourced by google
- Kubernetes is a project not a product



WHY DO CONTAINERS NEED KUBERNETES?



CONTAINERIZED APPLICATIONS

kubernetes

MANAGE CONTAINERS SECURELY

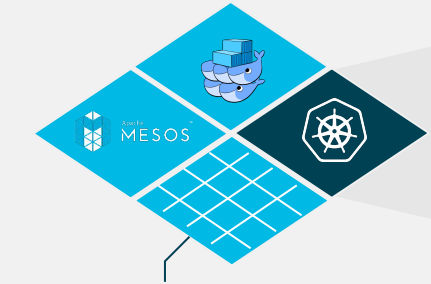
MANAGE CONTAINERS AT SCALE

INTEGRATE IT OPERATIONS

ENABLE HYBRID CLOUD

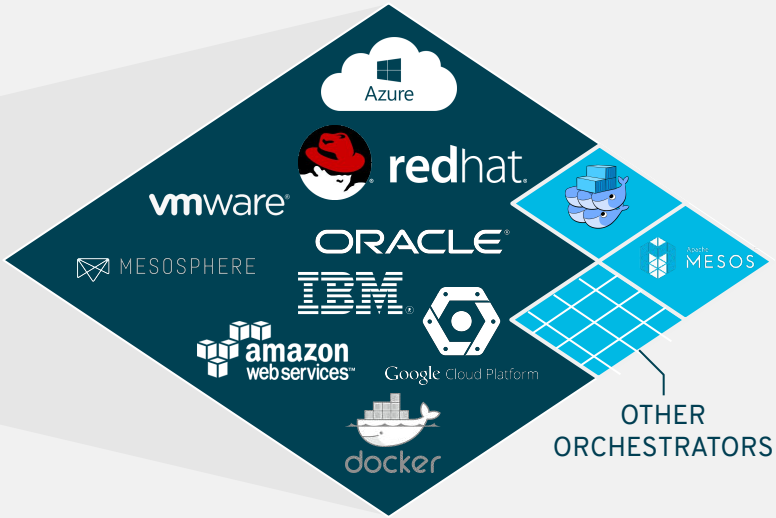
KUBERNETES IS THE CONTAINER ORCHESTRATION STANDARD

3 YEARS AGO
Fragmented landscape



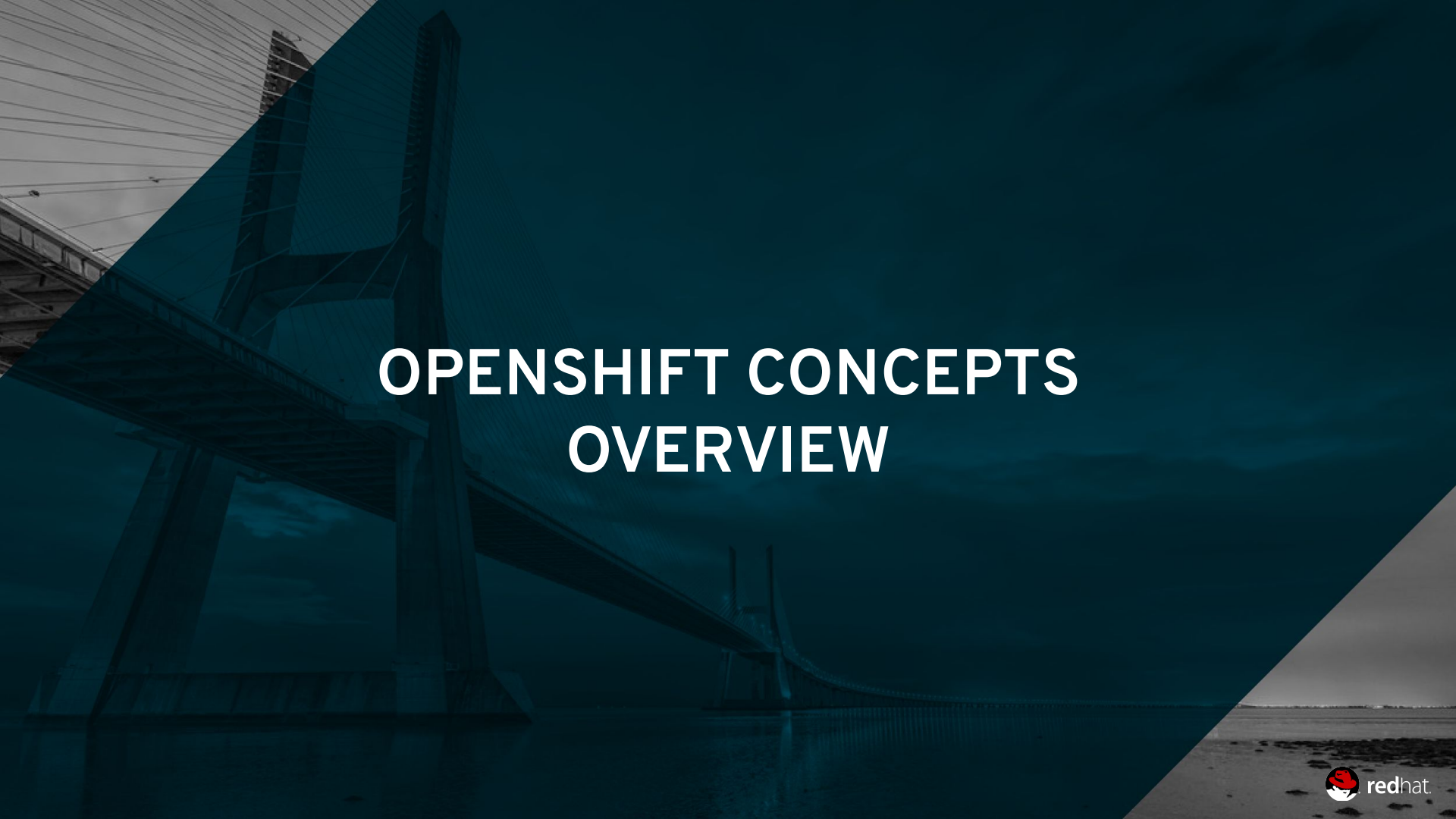
OTHER ORCHESTRATORS
(Cloud Foundry Diego,
Nomad, Blox, etc.)

TODAY
Kubernetes consolidation



OTHER
ORCHESTRATORS

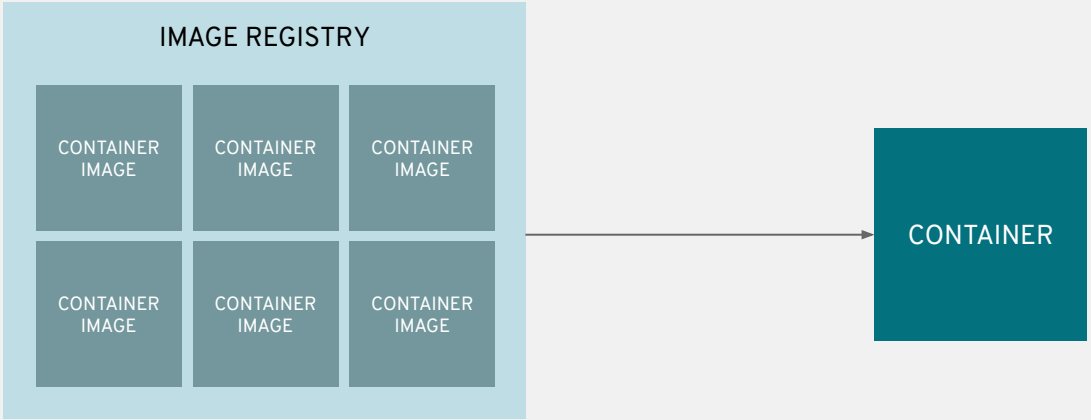
Red Hat bet early on Kubernetes. It has now become the dominant orchestration ecosystem


OPENSHIFT CONCEPTS
OVERVIEW
redhat.

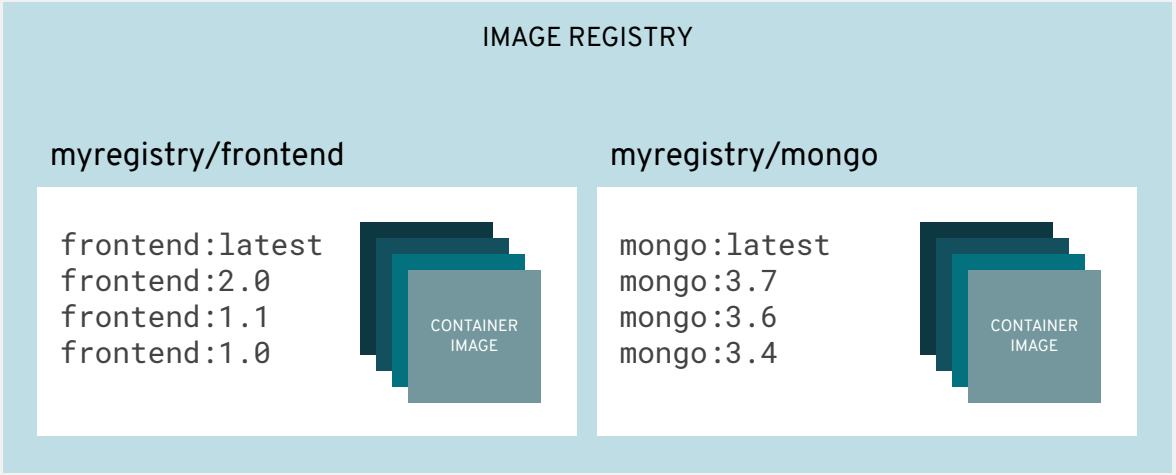
A container is the smallest compute unit

containers are created from container images

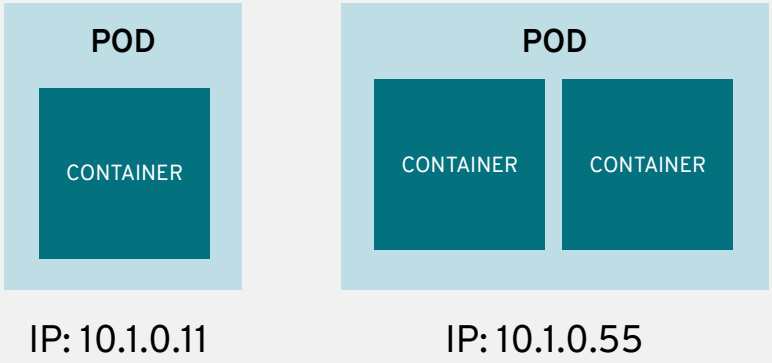
container images are stored in an image registry



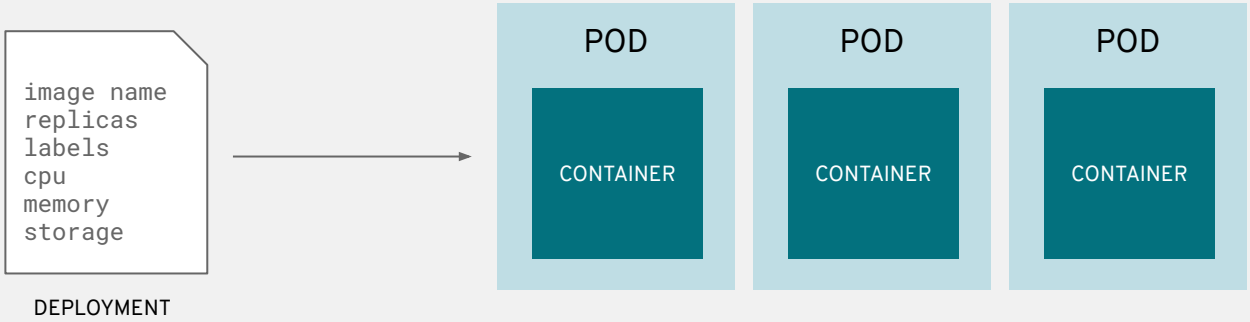
an image repository contains all versions of an image in the image registry



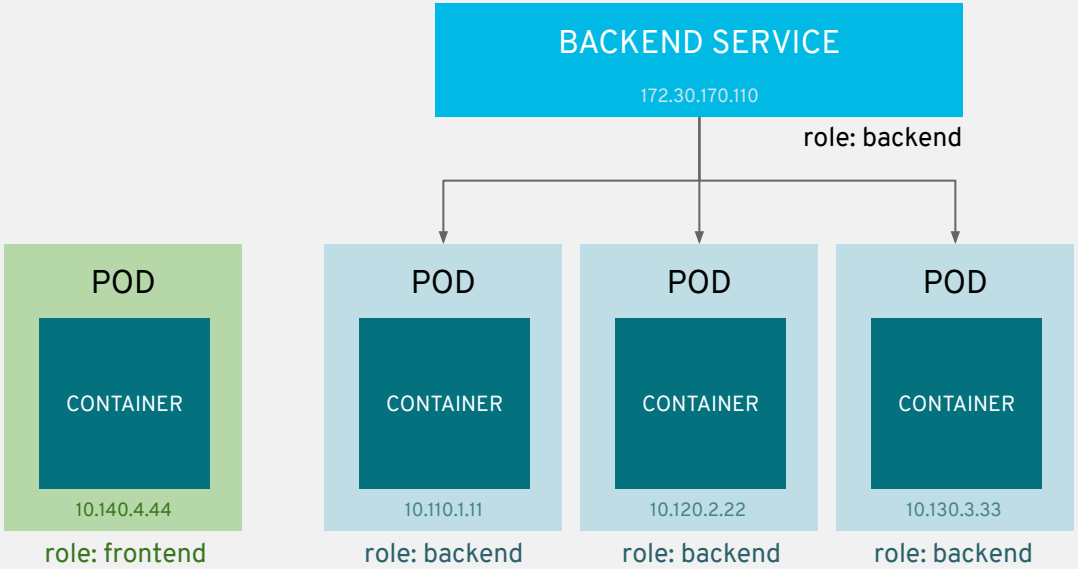
containers are wrapped in pods which are units of deployment and management



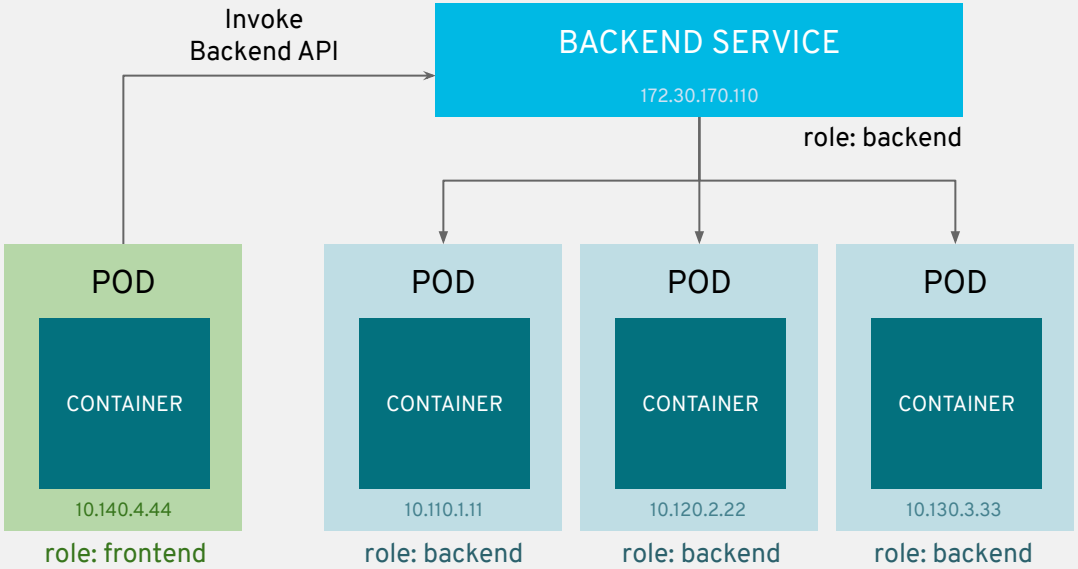
pods configuration is defined in a `deployment`



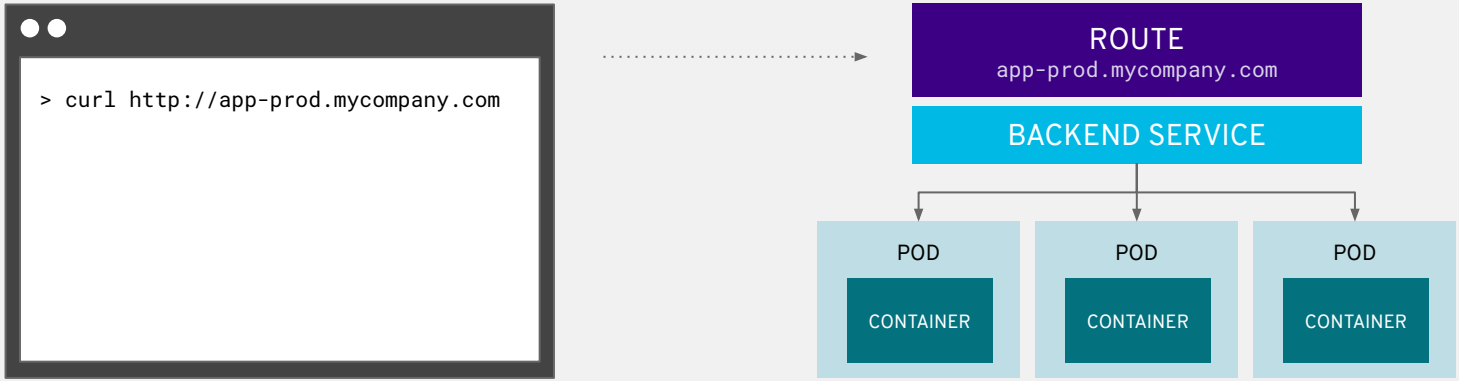
services provide internal load-balancing and service discovery across pods



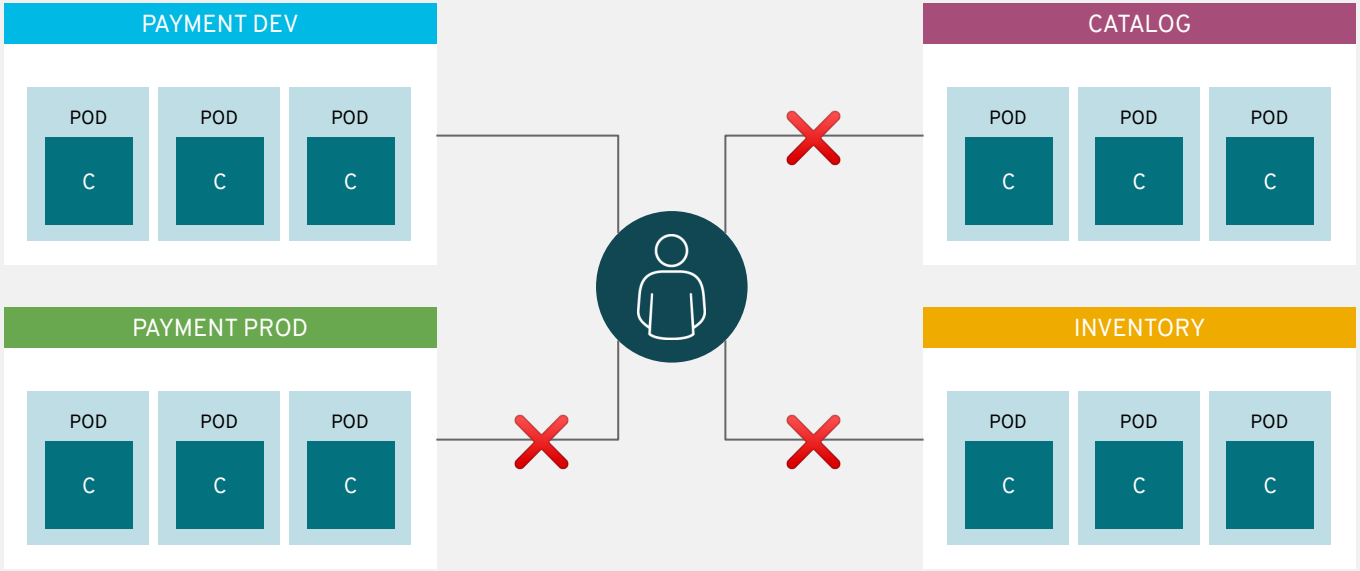
apps can talk to each other via services

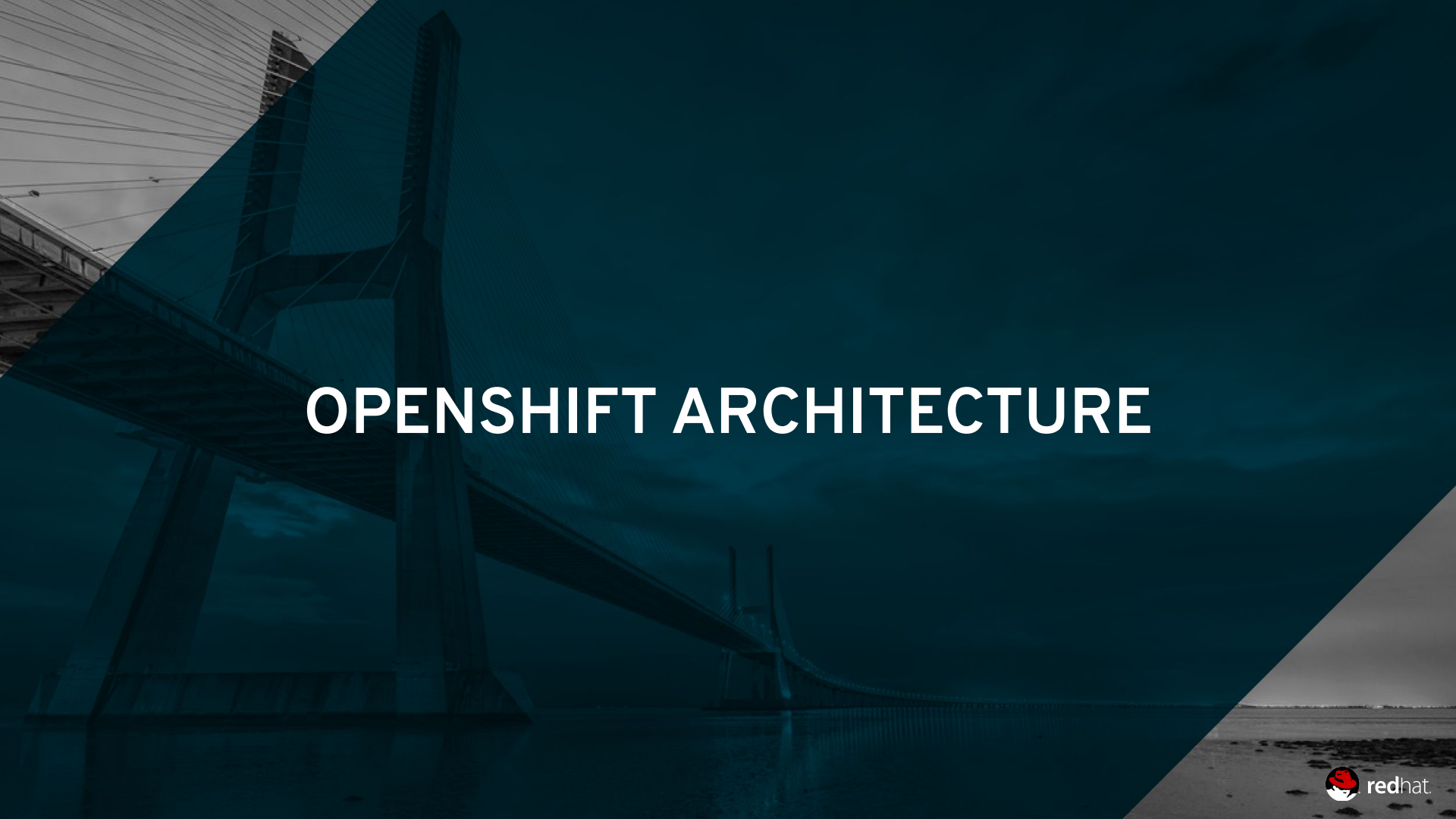


routes add services to the external load-balancer and provide readable urls for the app



projects isolate apps across environments, teams, groups and departments




OPENSHIFT ARCHITECTURE
redhat.

YOUR CHOICE OF INFRASTRUCTURE

Supported Infrastructures

OpenShift supported on RHEL and RHCOS

OpenShift 4.1 supported on AWS, VMware, and bare metal

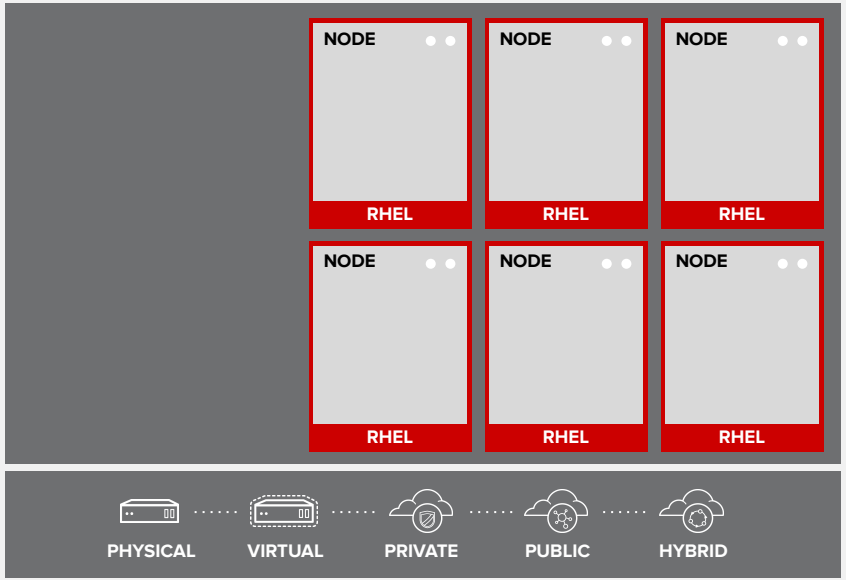
- Future releases to support more infrastructures

OpenShift 3.11 supported anywhere RHEL runs:

- On bare-metal physical machines or virtualized infrastructure, and in private or certified public clouds
  - Includes all supported virtualization platforms: Red Hat Virtualization, vSphere, Hyper-V
- On Red Hat OpenStack® Platform and certified public cloud providers like Amazon, Google, Azure
- x86 and IBM Power server architectures supported

Multi-cluster hybrid approach supports deploying OpenShift clusters on all of these infrastructures and federating them

NODES RHEL INSTANCES WHERE APPS RUN



Node Host

OpenShift runs on RHEL and RHCOS

OpenShift has two types of nodes:

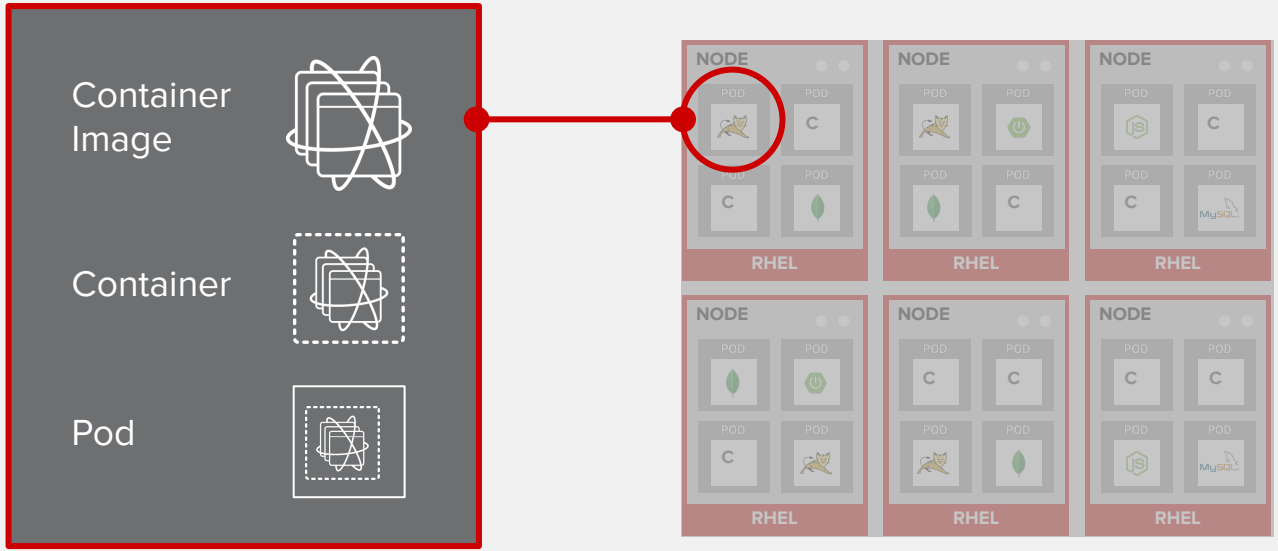
- Workers
- Masters

Nodes are:

- Instances of RHEL or RHCOS with OpenShift installed
- Where end-user applications run
- Orchestrated by master nodes

node daemon and other software run on all nodes

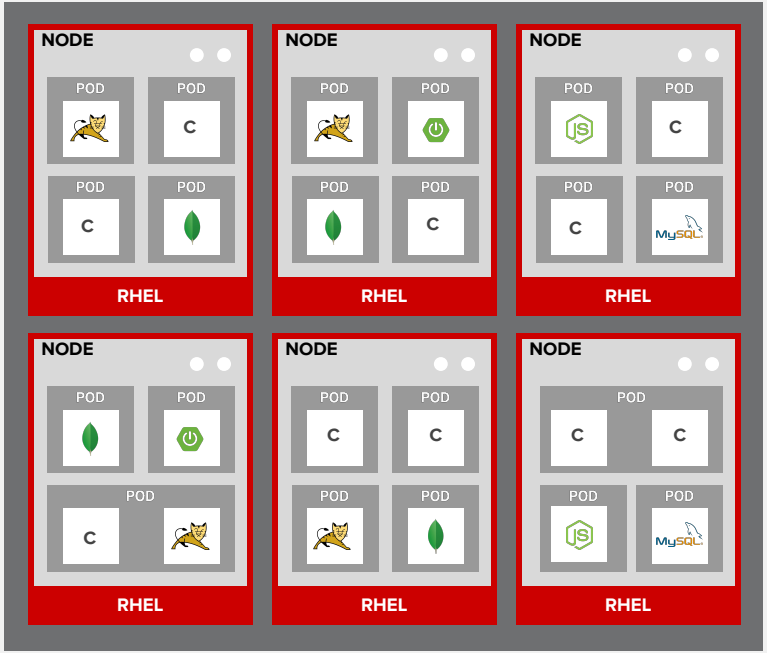
APPS RUN IN CONTAINERS



Container

- Application instances and components run in OCI-compliant containers
- OpenShift worker node can run many containers
- Node capacity is related to memory and CPU capabilities of underlying resources
- Hardware or virtualized

PODS ARE THE UNIT OF ORCHESTRATION



Pods

Orchestrated unit in OpenShift

- OpenShift schedules and runs all containers in pod on same node

Complex applications made up of many pods, each with own containers

- Interact externally and also with one another inside OpenShift environment

OpenShift runs container images in containers wrapped by meta object called "pod"

Possible to have multiple containers in single pod

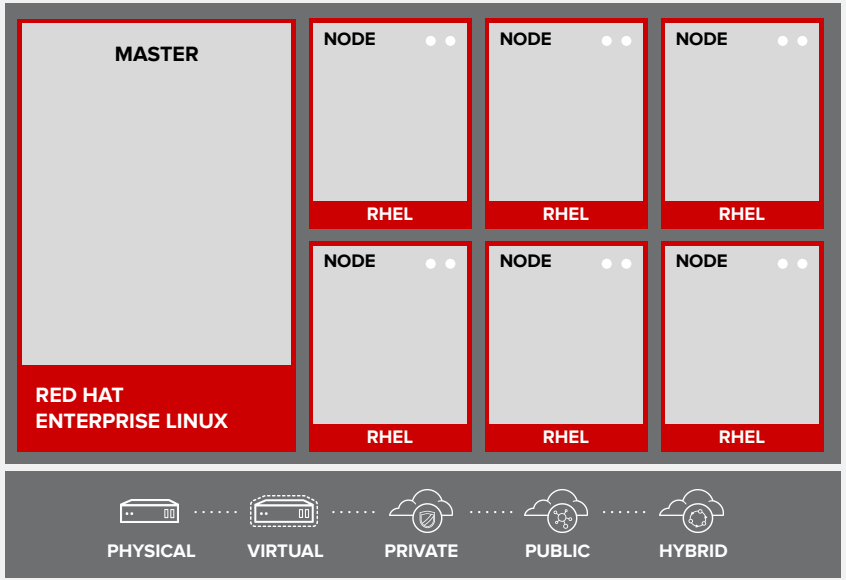
- Example: To support cluster features as sidecar containers

Most applications benefit from flexibility of single-container pod

- Different components such as application server and database generally not placed in single pod
- Allows for individual application components to be easily scaled horizontally

Application components are wired together by services

MASTERS ARE THE CONTROL PLANE



Master Nodes

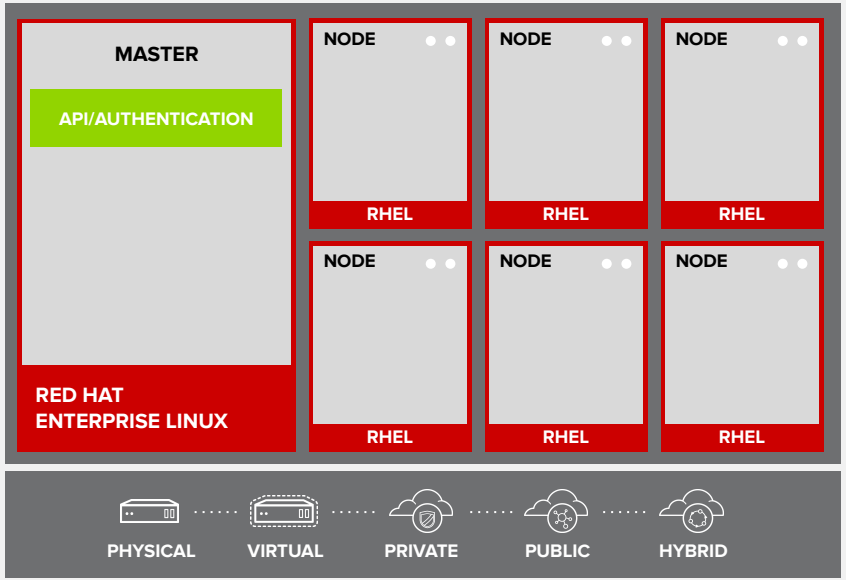
Instances of RHCOS

Primary functions:

- Orchestrate all activities on worker nodes
- Know and maintain state within OpenShift environment

Use multiple masters for high availability

API AND AUTHENTICATION



redhat.

Master API

Masters provide single API that all tooling and systems interact with

- All administration requests goes through this API

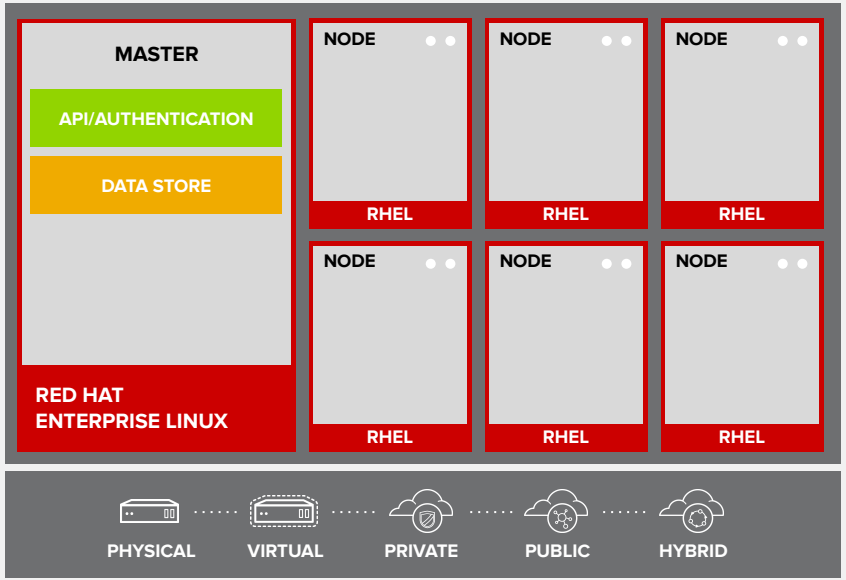
All API requests SSL-encrypted and authenticated

Authorizations handled via fine-grained role-based access control (RBAC)

Masters can be tied into external identity management systems

- Examples: LDAP, Active Directory, OAuth providers like GitHub and Google

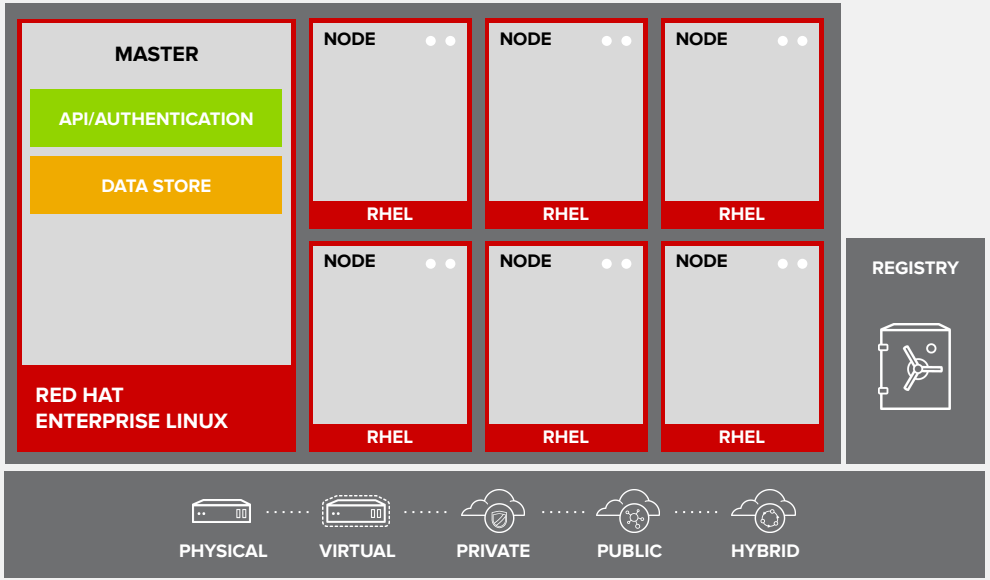
DESIRED AND CURRENT STATE



etcd

- Desired and current state held in data store that uses etcd as distributed key-value store
- etcd also holds RBAC rules, application environment information, non-application user data

INTEGRATED CONTAINER REGISTRY



redhat.

Container Registry

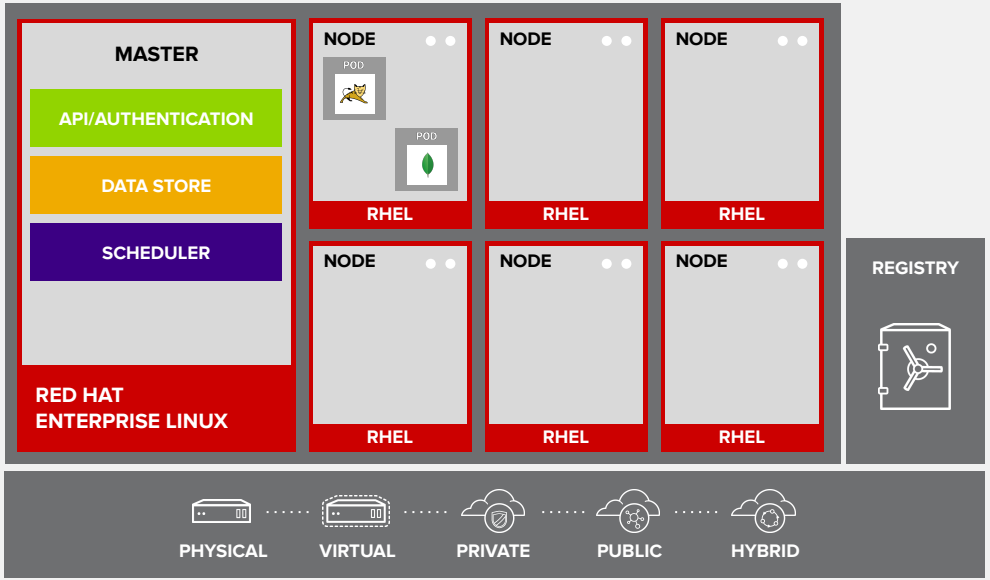
OpenShift Container Platform includes integrated container registry to store and manage container images

When new image pushed to registry, registry notifies OpenShift and passes along image information including:

- Namespace
- Name
- Image metadata

Various OpenShift components react to new image by creating new builds and deployments

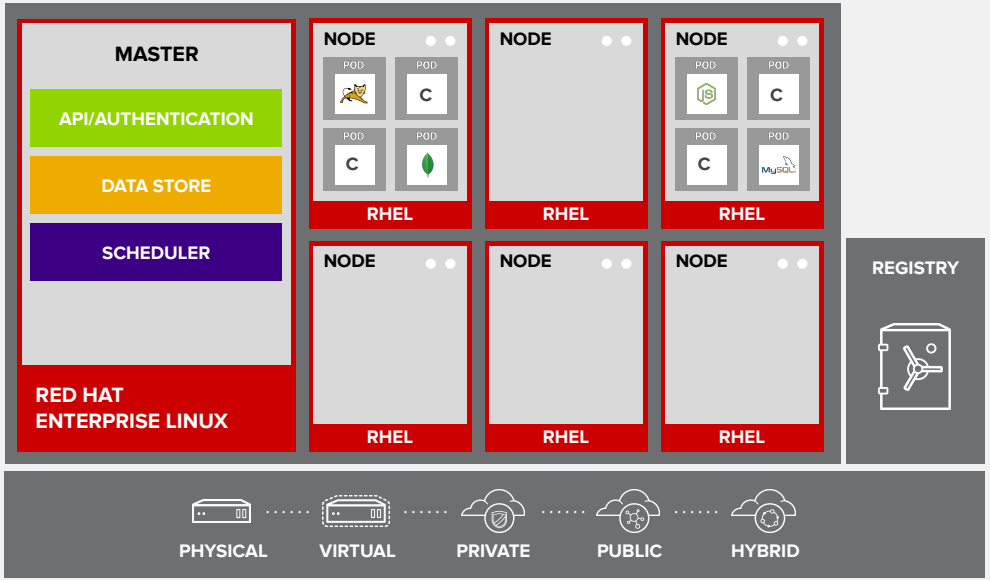
ORCHESTRATION AND SCHEDULING



Scheduler

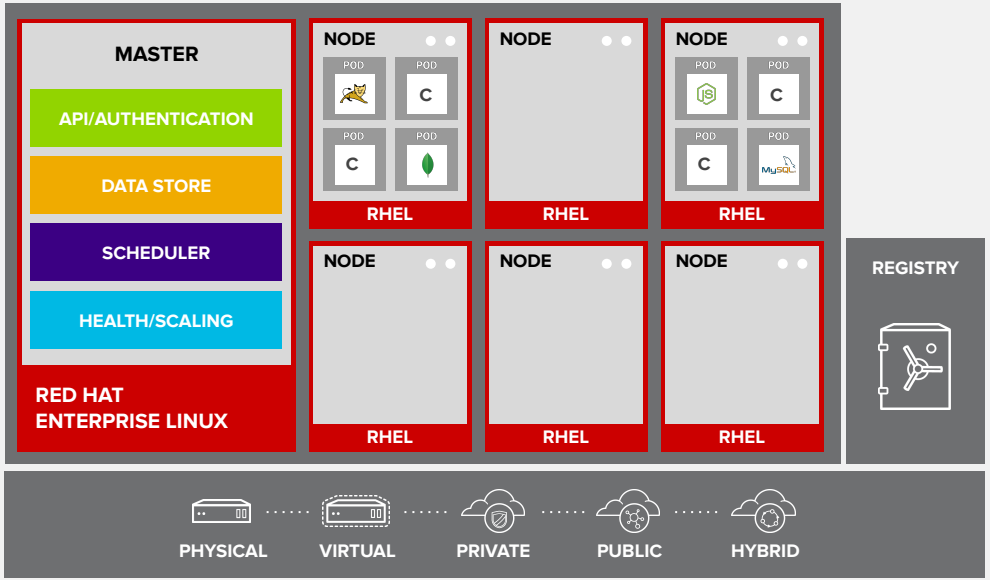
- Component responsible for determining pod placement
- Accounts for current memory, CPU, and other environment utilization when placing pods on worker nodes
- For application high availability, spreads pod replicas between worker nodes

PLACEMENT BY POLICY



redhat.

AUTOSCALING PODS



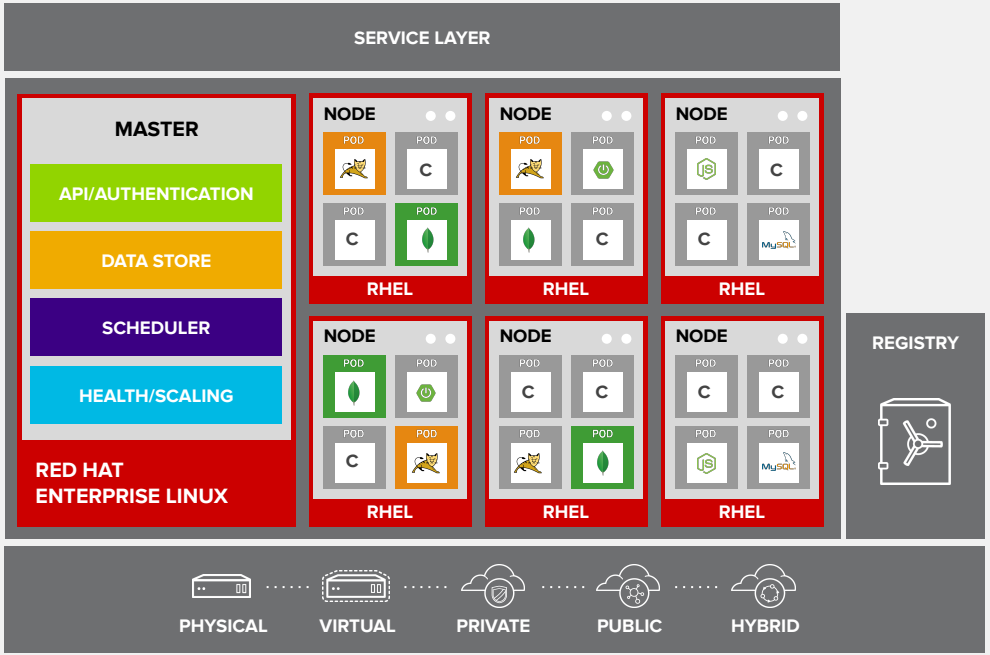
redhat.

Health and Scalinng

Masters monitor health of pods and automatically scale them

- User configures pod probes for liveness and readiness
- Pods automatically scaled based on CPU utilization metrics

SERVICE DISCOVERY



Services

Defines logical set of pods and access policy

- Provides permanent internal IP address and host name for other applications to use as pods are created and destroyed

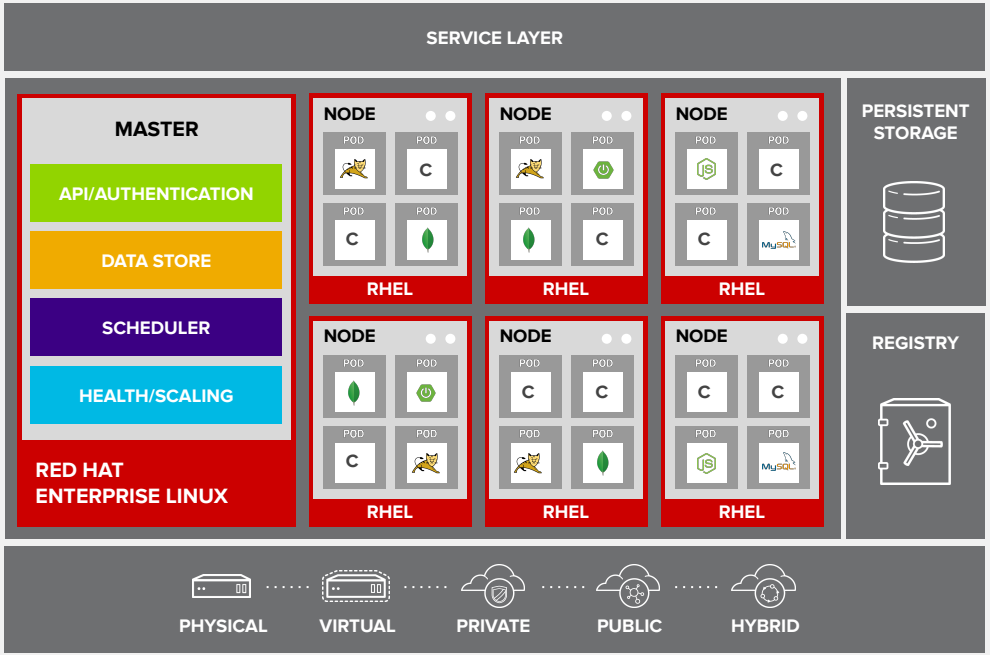
Service layer connects application components together

- Example: Front-end web service connects to database instance by communicating with its service

Services allow simple internal load balancing across application components

- OpenShift automatically injects service information into running containers for ease of discovery

PERSISTENT DATA IN CONTAINERS



redhat.

Persistent Data

Containers natively ephemeral

- Data not saved when containers restarted or created

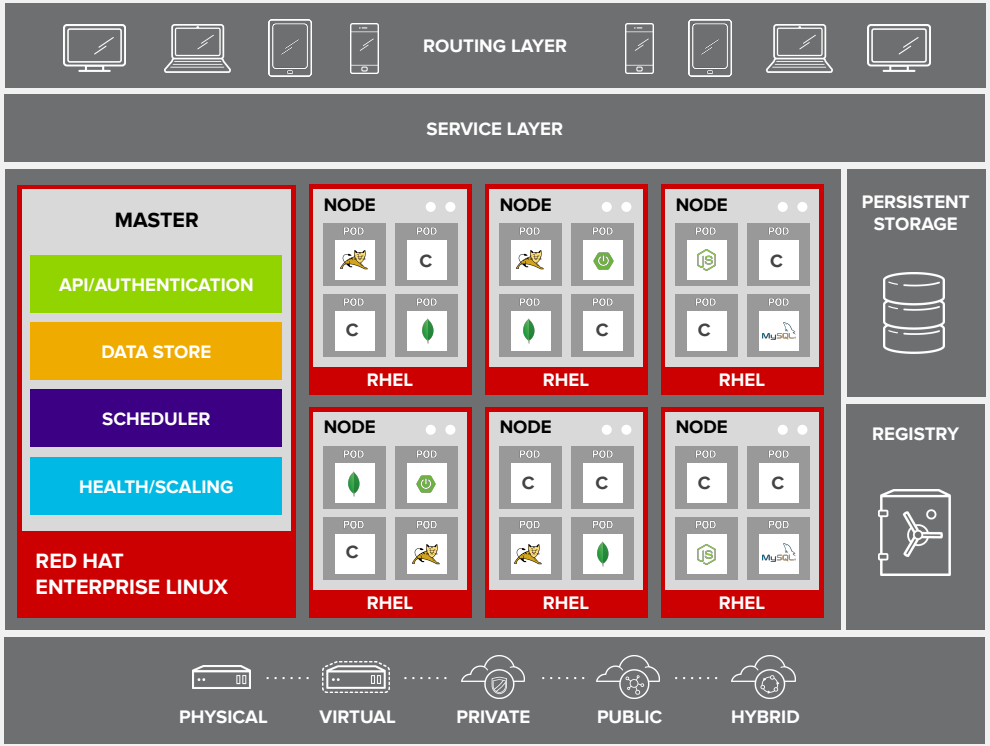
OpenShift provides persistent storage subsystem that automatically connects real-world storage to correct pods

- Allows use of stateful applications

OpenShift Container Platform provides wide array of persistent storage types including:

- Raw devices: iSCSI, Fibre Channel
- Enterprise storage: NFS
- Cloud-type options: Gluster®/Ceph®, AWS EBS, pDisk

ROUTING AND LOAD-BALANCING



Routing Layer

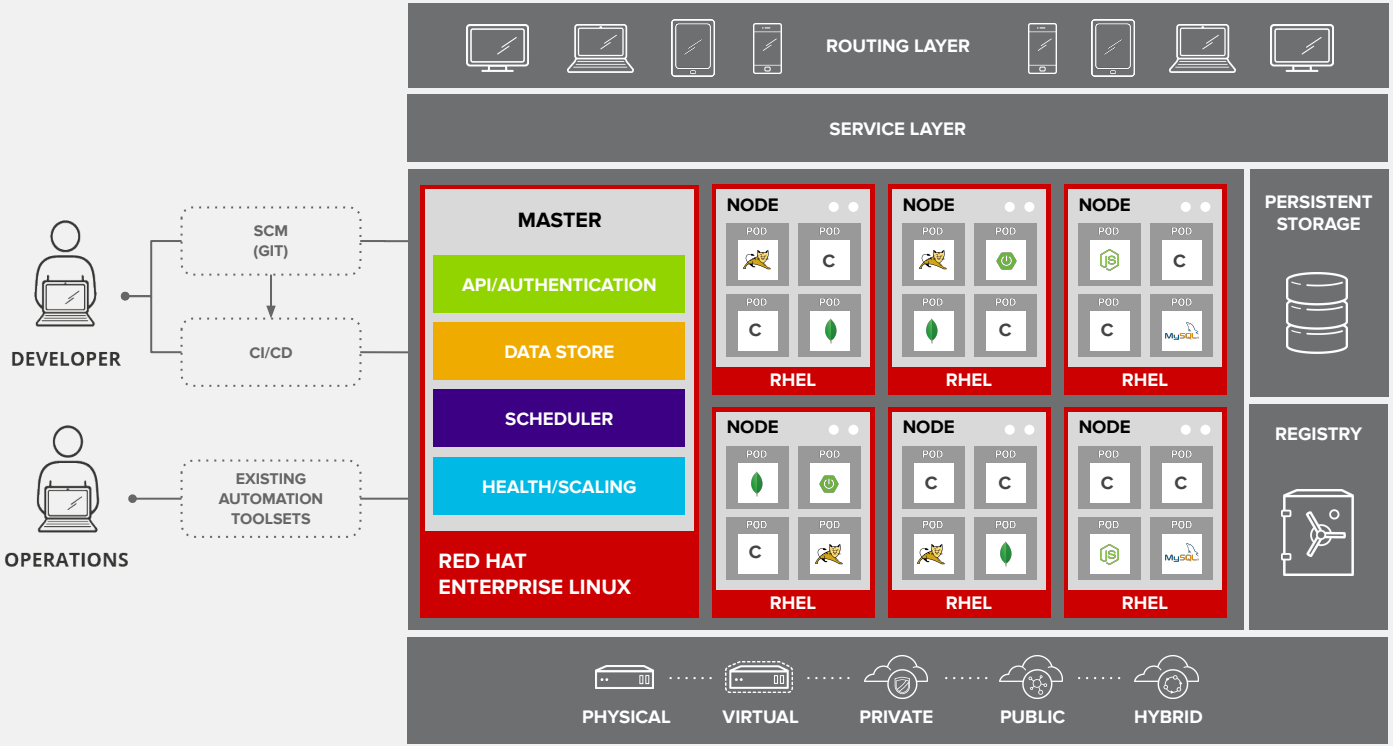
Provides external clients access to applications running inside OpenShift
Close partner to service layer
Runs in pods inside OpenShift
Provides:

- Automated load balancing to pods for external clients
- Load balancing and auto-routing around unhealthy pods

Routing layer pluggable and extensible

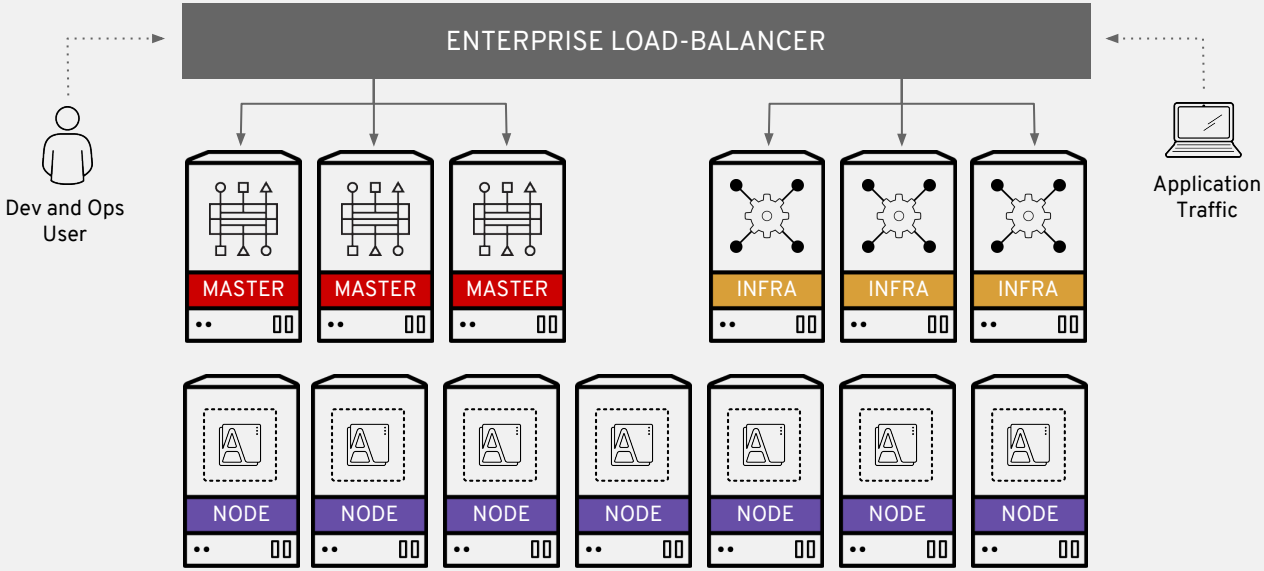
- Options include hardware or non-OpenShift software routers

ACCESS VIA WEB, CLI, IDE AND API



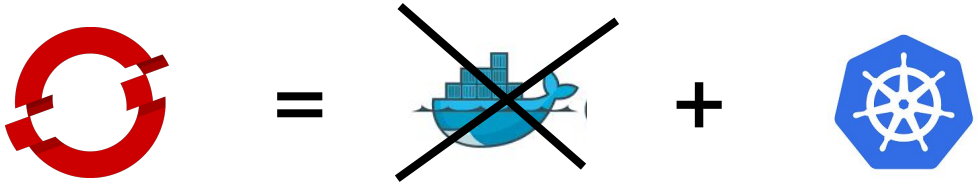
redhat.

FULL HIGH-AVAILABILITY ARCHITECTURE

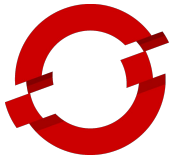



What is OpenShift?

made easy



made easy



=

+

made easy

OPEN CONTAINER INITIATIVE

- Docker, Red Hat et al. June 2015
- Two specifications
  - Image format
    - How to package an OCI Image with sufficient information to launch the application on the target platform
  - Runtime
    - How to launch a “filesystem bundle” that is unpacked on disk
- Version 1.0 of each released July 19th 2017
- Distribution spec started in April, 2018.

Red Hat Development Model

From Community to Enterprise: Upstream First

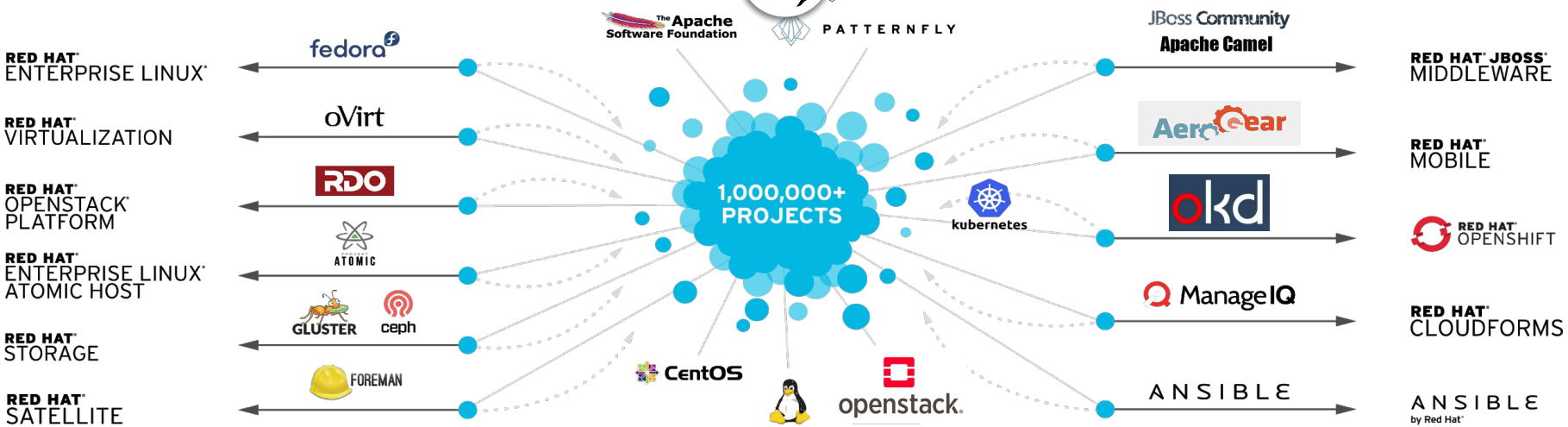
STABILIZE
(Enterprise supported products platforms, and solutions)

INTEGRATE
(community platforms)

PARTICIPATE
(upstream collaboration)

INTEGRATE
(community platforms)

STABILIZE
(Enterprise supported products platforms, and solutions)



RH0064-3

How Do We Deliver OpenShift?

okd

- Community Distribution of Kubernetes
- 100+ Integrations
- Align time with OSS trunk
- Integrate OSS projects
- Partner integration platform
- No-cost validations for innovation

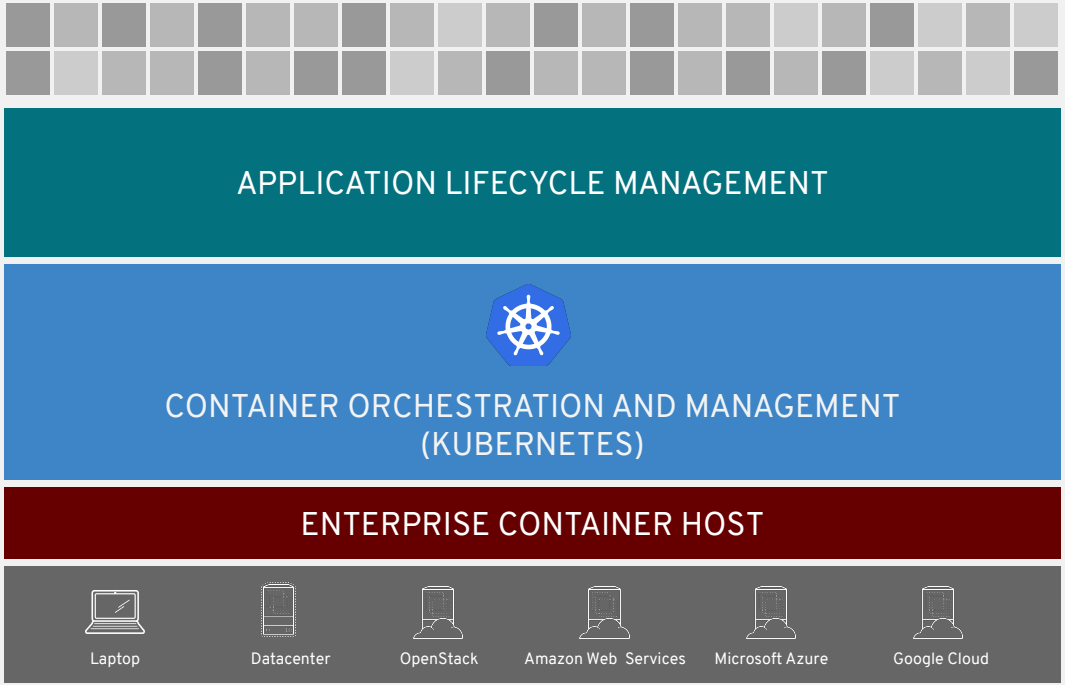
OPENSHIFT CONTAINER PLATFORM

OPENSHIFT DEDICATED

OPENSHIFT ONLINE

Red Hat

OPENSHIFT CONTAINER PLATFORM



ANY
CONTAINER

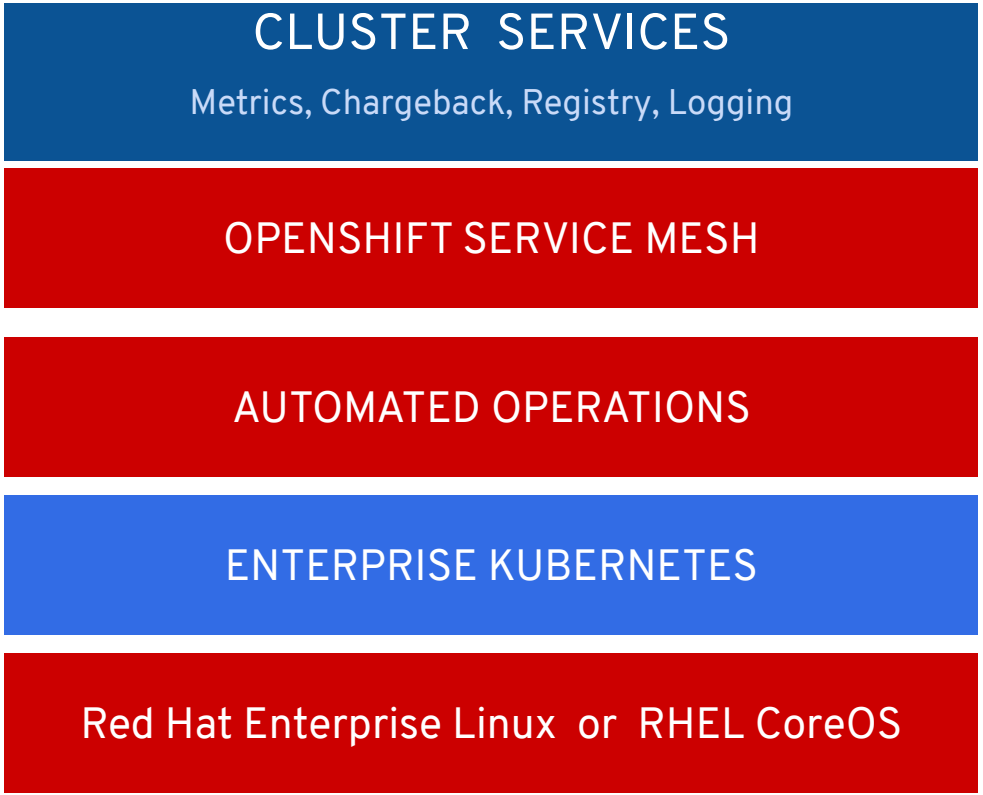
RED HAT®
OPENSHIFT

ANY
INFRASTRUCTURE

redhat.

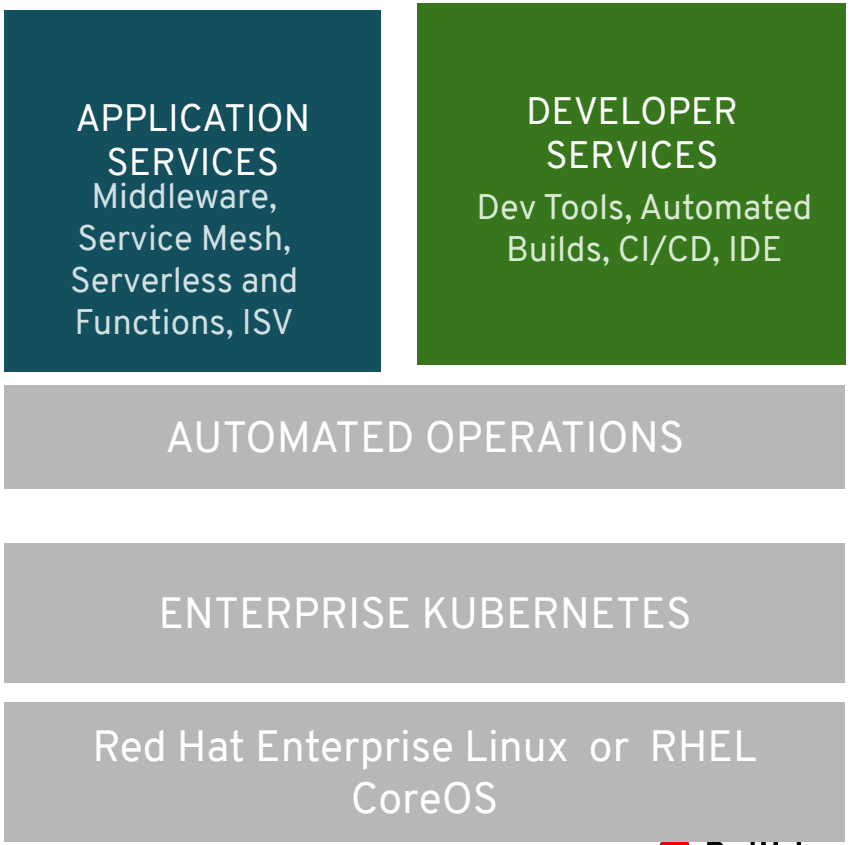
OPENSHIFT INCLUDES INFRASTRUCTURE AND OPERATIONS

1. Containers and Kubernetes provide multi-cloud portability.
2. Integrated SDN and Storage
3. Fully-automated Day-2 operations.
4. Integrated container registry or Red Hat Quay.
5. Integrated Service Mesh (Istio).
6. Integrated Logging, Monitoring, Metrics.
7. Integrated security throughout the stack.



App/dev services

1. Support for a broad set of applications on OpenShift.
2. Integrated Service Catalog and Marketplace
3. Integrated CI/CD pipelines.
4. Integrated Red Hat or IBM Middleware.
5. Integrated Service Mesh (Istio).
6. Integrated Serverless (Knative).
7. Integrated web-based IDE (Che).
8. Operators, Operator Lifecycle Mgmt, OpenShift Marketplace.



OPENSHIFT MAKES DOCKER UND KUBERNETES EASY TO USE

WHAT COMES IN KUBERNETES

- Container Scheduling on Multiple Hosts
- Self-healing
- Role Based Access Control
- Scaling
- Service Discovery
- Rolling Deploys and Rollbacks

WHAT OPENSHIFT ADDS OVER KUBERNETES

Ops:

- Software Defined Network
- Persistent Storage
- Container Native Storage (CNS / SDS)
- Log Aggregation and Analysis
- Monitoring | Telemetry
- Capacity Management
- Egress Routing for Enterprise integration
- Router Sharding
- Full Stack Support
- System Certifications and Patching
- …

Dev:

- Automatically Triggered Deployments (CICD)
- Integrated Customizable Pipelines (CICD)
- Build and Deployment Configurations
- Weighted AB Testing
- Stateful Workloads (Storage, StatefulSets)
- Workload Containerization
- Self-service
- User Experience
- …

Security:

- Container Security and Isolation (SELinux, etc)
- Multi-tenancy
- Secured Registries
- Automated Deployment Patching
- …

THE POWER OF THE OPENSHIFT ECOSYSTEM

RED HAT PORTFOLIO
Optimized for Containers

- RED HAT OPENSHIFT Application Runtimes
- RED HAT JBOSS WEB SERVER
- RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM
- RED HAT DATA GRID
- RED HAT AMQ
- RED HAT FUSE
- RED HAT MOBILE
- RED HAT ANSIBLE Engine
- RED HAT QUAY CONTAINER REGISTRY
- RED HAT DECISION MANAGER
- RED HAT PROCESS AUTOMATION MANAGER
- RED HAT 3SCALE API MANAGEMENT
- RED HAT OPENSHIFT Container Storage

THIRD-PARTY ISV
Red Hat Container Catalog (100s certified)

IBM, Microsoft, SAP, New Relic, Couchbase, Crunchy Enterprise PostgreSQL, Sysdig, f5, Veritas, Zabbix, NetApp, Sonatype, JFrog, dynatrace, nuagenetworks

CLOUD SERVICES
Open Service Broker

amazon web services, Microsoft Azure, Google Cloud

RED HAT OPENSHIFT

RED HAT ENTERPRISE LINUX ECOSYSTEM
Hardware, Virtualization, Cloud and Service Provider Certifications

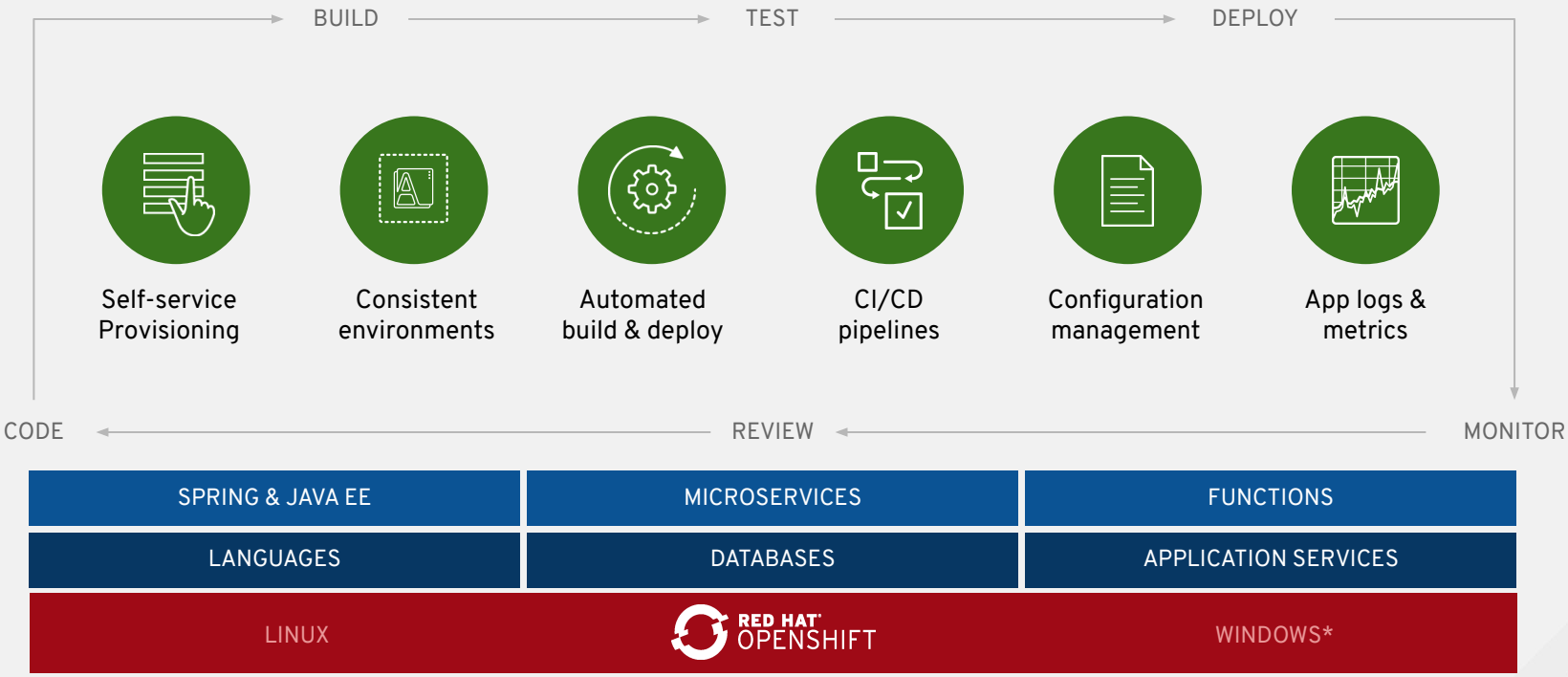
Developers want to be productive and have choice

Choice of architectures
Choice of programming languages
Choice of databases
Choice of application services
Choice of development tools
Choice of build and deploy workflows

They don't want to have to worry about the infrastructure.

Photo: rawpixel on Unsplash

HOW OPENSHIFT ENABLES
DEVELOPER PRODUCTIVITY



* coming soon

IT Operations needs secure, efficient and controlled processes

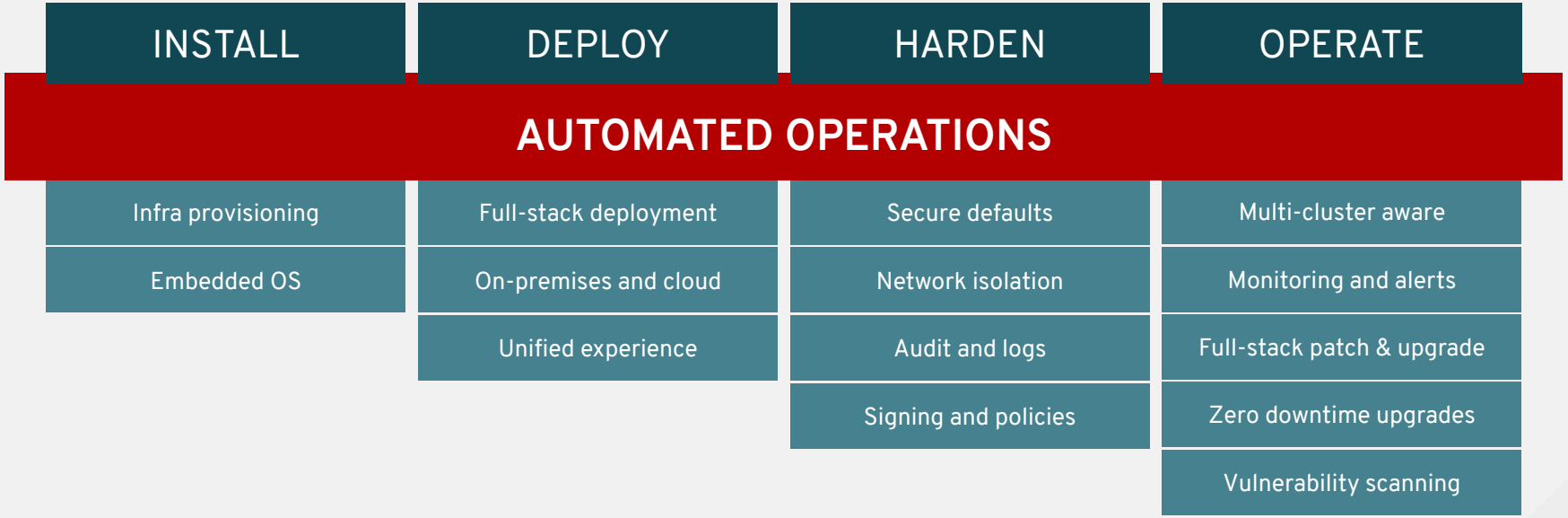
Automated* provisioning
Automated installations
Automated security scanning
Automated upgrades
Automated backups

And it needs to integrate with what you already have.

*coming soon

AUTOMATED CONTAINER OPERATIONS

Fully automated day-1 and day-2 operations

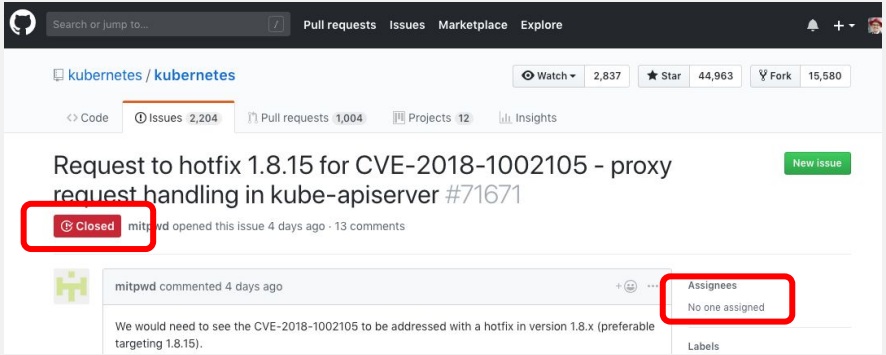


Community vs. Enterprise - Security Example

Dec 2018 - **Critical** (9.8 of 10) Kubernetes Security Vulnerability CVE-2018-1002105,

Kubernetes Community:

- Fixed **3** releases - (1.10, 1.11, 1.12)
- "All prior versions remain exposed and users should stop using them immediately"
- Requests to fix older versions are denied:



Enterprise OpenShift from Red Hat:

- Fixed **10** releases back to OpenShift V3.2 (which corresponds to kubernetes 1.2) :

Updates for Affected Products

Product	Package	Advisory/Update
OpenShift Container Platform v3.11	kubernetes	RHSA-2018:3537
OpenShift Container Platform v3.10	kubernetes	RHSA-2018:3549
OpenShift Container Platform v3.9	kubernetes	RHSA-2018:2908
OpenShift Container Platform v3.8	kubernetes	RHSA-2018:3551
OpenShift Container Platform v3.7	kubernetes	RHSA-2018:2906
OpenShift Container Platform v3.6	kubernetes	RHSA-2018:3598
OpenShift Container Platform v3.5	kubernetes	RHSA-2018:3624
OpenShift Container Platform v3.4	kubernetes	RHSA-2018:3752
OpenShift Container Platform v3.3	kubernetes	RHSA-2018:3754
OpenShift Container Platform v3.2	kubernetes	RHSA-2018:3742

OCP vs DIY Kubernetes

	Red Hat OpenShift 4	kubernetes
Container orchestration	✓ Kubernetes	✓ Kubernetes
Container image	✓ OCI-compliant/docker	✓ BYO OCI-compliant
Container runtime	✓ CRIO/docker	✓ BYO OCI-compliant engine
Container build	✓ RHCC/S2I/dockerfile	✗
Container registry	✓ Quay/OSS docker registry	✗
Container scanner	✓ OSCAP/Clair	✗
CI/CD automation	✓ Jenkins 2	✗
IDE	✓ Che	✗
Web UX	✓ Web console	✓ Web console
CLI UX	✓ oc/odo/kubectl	✓ kubectl
Service catalog	✓ Operators/OSB-API	✗
Secrets management	✓ Kubernetes Secrets	✓ Kubernetes Secrets
Supported/preferred runtime	✓ RHOAR (EAP, Spring, vert.x, node.js)	✗

OCP vs DIY Kubernetes

	Red Hat OpenShift 4	kubernetes
Service mesh	✓ Istio, Jaeger, Kiali, Profana	✗
Logging	✓ EFK	✗
Metrics	✓ Prometheus/Grafana	✗
Storage	✓ OpenShift Container Storage	✗
Network	✓ OVN	✗
Ingress	✓ Kubernetes Ingress/Routes	✓ Kubernetes Ingress
Ingress controller	✓ HA Proxy	✗
Egress	✓ Egress Router	✗
Authentication	✓ Kubernetes Auth/RH-IdM	✓ Kubernetes Auth
App isolation	✓ Kubernetes scheduler	✓ Kubernetes Scheduler
Infrastructure	✓ Bare metal, vSphere, KVM, OpenStack, AWS, GCP, Azure	✓ BYO Linux
Infra automation	✓ Ansible/Terraform/Operators	✗
Infra management	✓ Admin Console	✗
Operating system	✓ RHEL or RHEL CoreOS	✗
CNCF Certified Kubernetes	✓	✗

Red Hat

OCP vs DIY Kubernetes

		Red Hat OpenShift 4 Container Platform	okd Project	kubernetes Orchestration Engine
Stability	Long-term maintenance and support with backwards compatibility of bug fixes and security patches	●	○	○
	Security and feature backport	●	○	○
	Long-term support and release lifecycle management	●	○	○
	Config and settings tested for scale and enterprise use	●	○	○
	No need to understand, check, and maintain dependencies on arbitrary third-party project binaries	●	○	○
	Automated software updates, alerting, and management	●	○	○
Patches/ updates	Patches provided and maintained by Red Hat	●	○	○
	Certified hotfixes/patches and security updates	●	○	○
Certifications	Common Criteria certifications	● RHEL and RHEL Core OS	○	○

Red Hat

OCP vs DIY Kubernetes

		Red Hat OpenShift 4 Container Platform	okd Project	kubernetes Orchestration Engine
Security	Formalized enterprise security resolution process	●	○	○
	Security response team identifies, tracks and resolves issues	●	○	○
	Containers wrapped in SELinux custom contexts via MCS	●	◑	○
	Full SELinux container security Much stronger than the basic namespace protection	●	◑	○
	Single Sign-On support (OAuth, SAML)	●	◑	○
Support	SLA, defect escalation, end-of-life policy management	●	○	○
	Support portal	●	○	○
	Production support	●	○	○
	Product developers provide support	●	○	○
	Ability to file request for enhancements	●	○	○
	Access to project leads (Clayton Coleman, Jeremy Eder, etc)	●	○	○

Red Hat

OCP vs DIY Kubernetes

		Red Hat OpenShift 4 Container Platform	okd Project	kubernetes Orchestration Engine
Consulting, TAM and training certification	Expert consulting from Red Hat and certified partners	●	○	○
	Training content driven directly by those who wrote the code	●	○	○
	Training certification	●	○	○
Quality assurance	Scalability tested/hardened	●	○	○
	Certification for multiple cloud providers AWS, Google GCP, Microsoft Azure	●	○	○
Ecosystem	Third party ISV support: ex: applications running in containers and integrations	●	○	○
	Full stack integrations tested with Red Hat Cloud Suite*	●	○	○
	Full scalable SDN support via Open vSwitch (VXLAN managed by NetFlow rules)	●	○	○
	Dynamic storage provisioning for Persistent Volumes via EBS, Cinder, and GCE	●	◑	◑

*OpenStack, OpenShift, storage (Ceph and Gluster) and Ansible Tower

Red Hat

OCP vs DIY Kubernetes

		Red Hat OpenShift 4 Container Platform	okd Project	kubernetes Orchestration Engine
Developer productivity	Container development kit for developers	●	◑	○
	Atomic Registry with tested, secured, and supported images, e.g. EAP, BRMS, BPMS, Python, etc.	●	○	○
	Multi-tenant experience	●	◑	○
	Graphical experience	●	◑	○
	Innovation Labs quick start/support	●	○	○
	Influence product roadmap (RFE submissions)	●	○	○
	Access to project leads	●	○	○
	CI/CD integration - support for Jenkins, Git and most other industry standards	●	◑	○
	Production-grade container image registry (Quay)*	●	◑	○

*OpenShift's container image registry is much more feature-rich and capable than the base registry provided as part of Docker containers

Red Hat

OCP vs DIY Kubernetes

		Red Hat OpenShift 4 Container Platform	okd Project	kubernetes Orchestration Engine
Consulting, TAM and training	EFK (Elasticsearch, FluentD, Kibana - similar to ELK stack) for centralized stateless remote logging. Logging indexes manage themselves as you spin up new apps	●	●	●
	Hawkular/Heapster/Cassandra for performance metrics stack	●	●	○
	Role-based access control (RBAC)	●	●	○
Ecosystem	Load balancers - examples: support by F5, Avinetworks, Amazon ELB, and other major load balancer providers	●	●	○
	Development tools - certification to run plugins from app dev tool providers, example: Appdynamics	●	●	○
	Certification for multiple cloud providers, examples: Amazon AWS, Google GCE, Microsoft Azure	●	○	○

Red Hat

KUBERNETES CONFORMANCE

Interoperability at the API

Branch: master ▾ | k8s-conformance / v1.11 / openshift / | Create new file | Upload files | Find file | History

smarterclayton and taylorwaggoner OpenShift v1.11 conformance results (#385) — Latest commit 7615279 on Nov 28, 2018

..

PRODUCT.yaml	OpenShift v1.11 conformance results (#385)	3 months ago
README.md	OpenShift v1.11 conformance results (#385)	3 months ago
e2e.log	OpenShift v1.11 conformance results (#385)	3 months ago
junit_01.xml	OpenShift v1.11 conformance results (#385)	3 months ago
version.txt	OpenShift v1.11 conformance results (#385)	3 months ago

README.md

This conformance report is generated by the OpenShift CI infrastructure. The canonical source location for this test script is located at https://github.com/openshift/origin/blob/master/test/extended/conformance-k8s.sh

This file was generated by:

Commit 17bd01127a26f49de8b5a9f420e465de7c32dea8 Tag v3.11.0-63-g17bd011

To recreate these results

1. Install an OpenShift cluster
2. Retrieve a `.kubeconfig` file with administrator credentials on that cluster and set the environment variable KUBECONFIG

   export KUBECONFIG=PATH_TO_KUBECONFIG

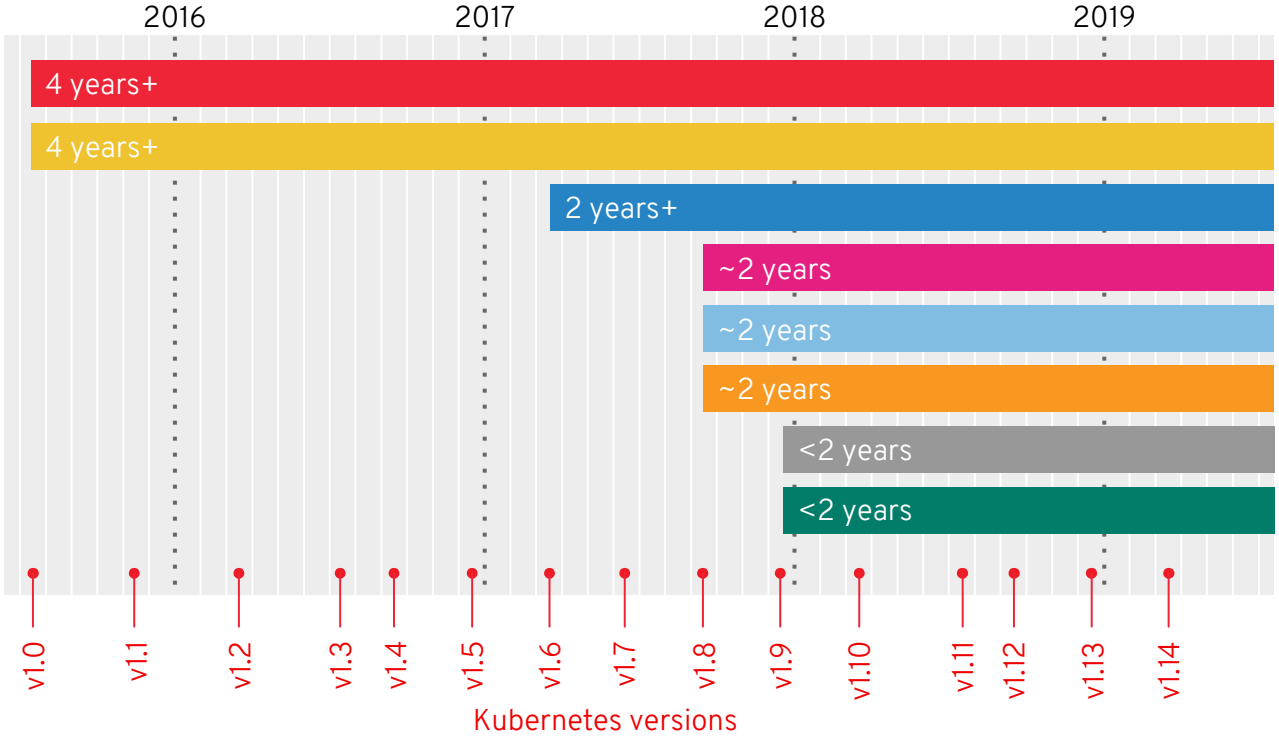
The conformance test on OpenShift install https://github.com/cncf/k8s-conformance/tree/master/v1.11/openshift https://github.com/cncf/k8s-conformance

OPENSHIFT IS KUBERNETES
FOR THE ENTERPRISE

Security fixes
100s of defect and performance fixes
200+ validated integrations
Middleware integrations
(container images, storage, networking, cloud services, etc)
9 year enterprise lifecycle management
Certified Kubernetes

Why Red Hat?

Red Hat has a strong history with Kubernetes



Red Hat
Google Cloud Platform
Microsoft Azure
MESOSPHERE
docker
aws
vmware
Pivotal

KUBERNETES SIGs & WGs- ENGINEERING LEADERSHIP

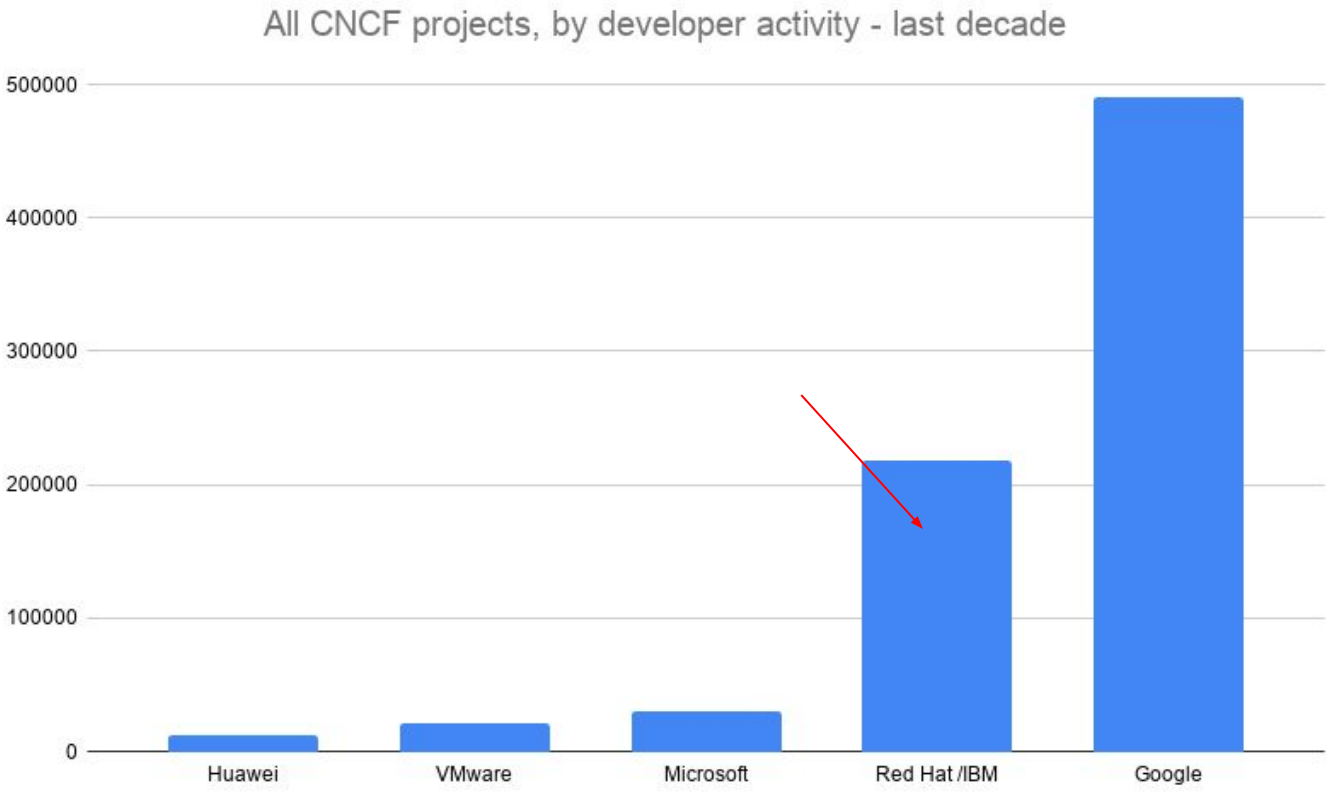
RED HAT LEAD or CO-LEAD

21 of 43 GROUPS

API MACHINERY	APPS	ARCHITECTURE	AUTH	AUTO SCALING	AWS
AZURE	BIG DATA	CLI	CLOUD PROVIDER	CLUSTER LIFECYCLE	CLUSTER OPS
CONTRIBUTOR EXPERIENCE	DOCS / GCP	IBM CLOUD	INSTRUMENTATION	MULTICLUSTER	NETWORK
NODE	OPENSTACK	PRODUCT MANAGEMENT	RELEASE	SCALABILITY	SCHEDULING
SERVICE CATALOG	STORAGE	TESTING	UI	VMware	WINDOWS
APP DEF	APPLY	COMPONENT STANDARD	CONTAINER IDENTITY	IoT EDGE	K8S INFRASTRUCTURE
KUBEADM ADOPTION	MACHINE LEARNING	MULTI TENANCY	POLICY	RESOURCE MANAGEMENT	SECURITY AUDIT

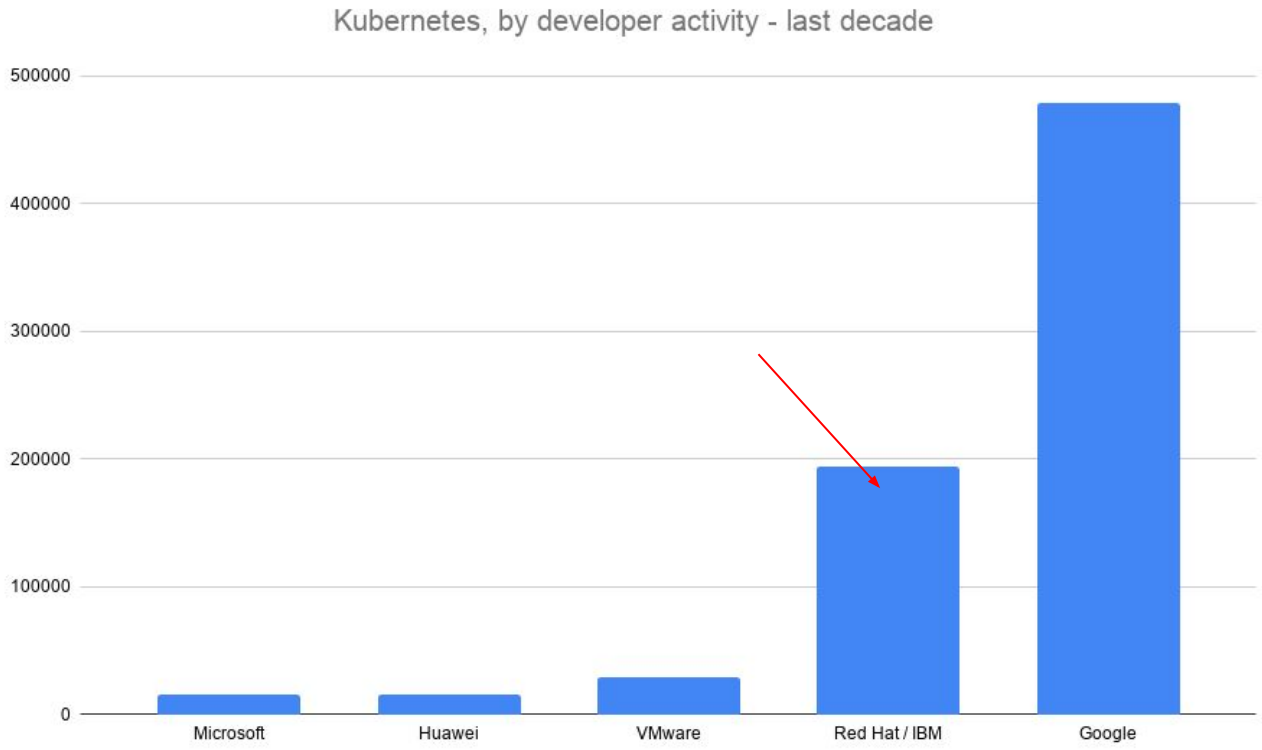
https://github.com/kubernetes/community/blob/master/sig-list.md

Cloud Native Computing Foundation



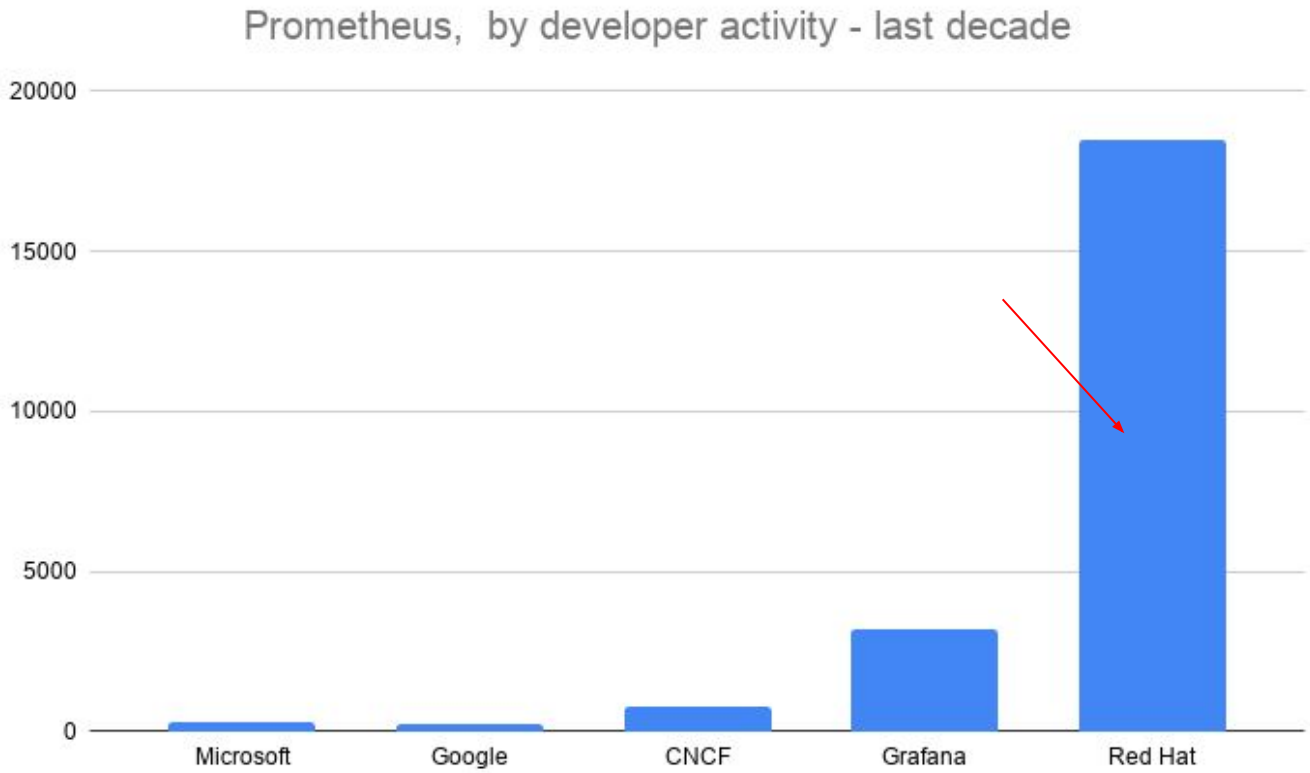
Source: CNCF (Oct, 2019)

Kubernetes activity by developer, last decade



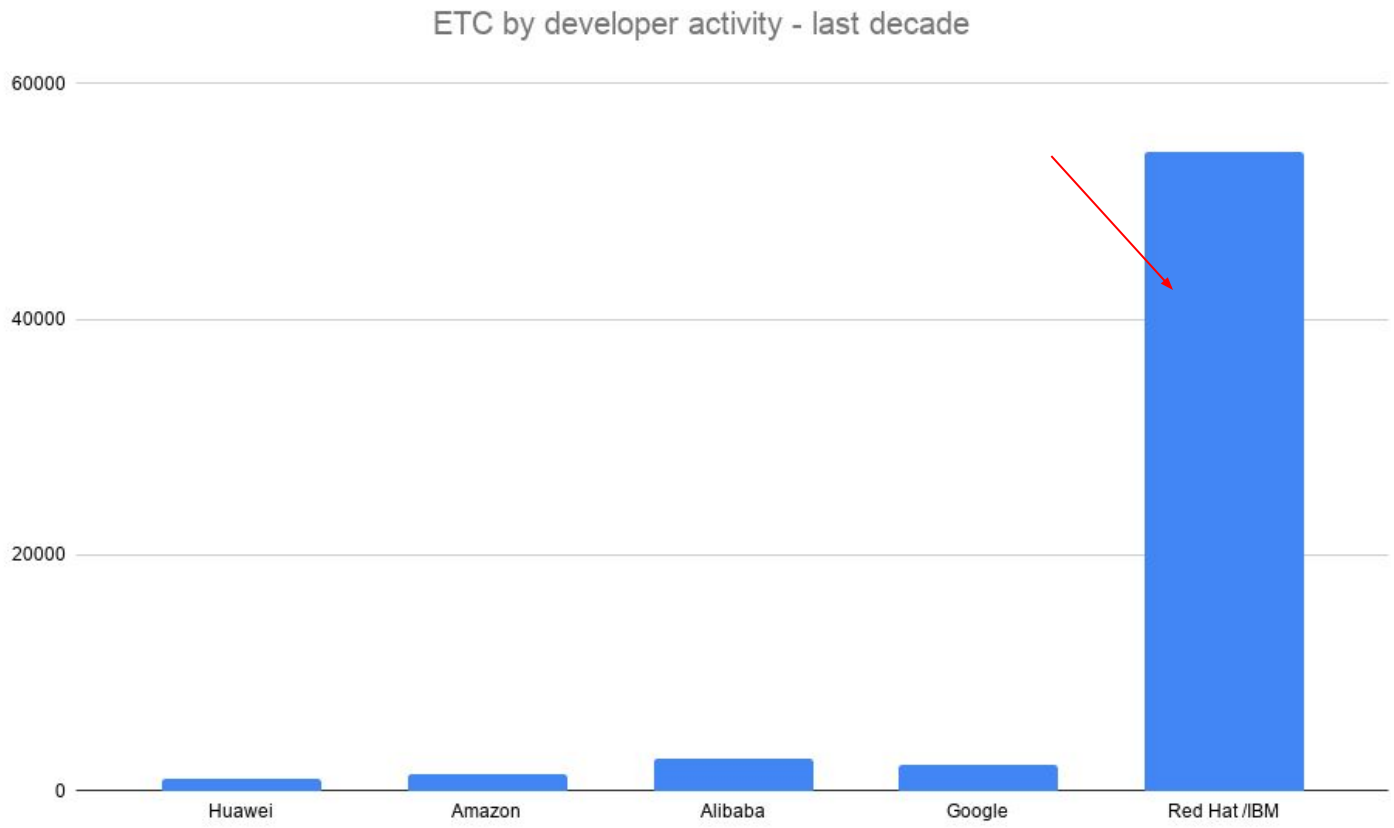
Source: CNCF (Oct, 2019)

Prometheus monitoring

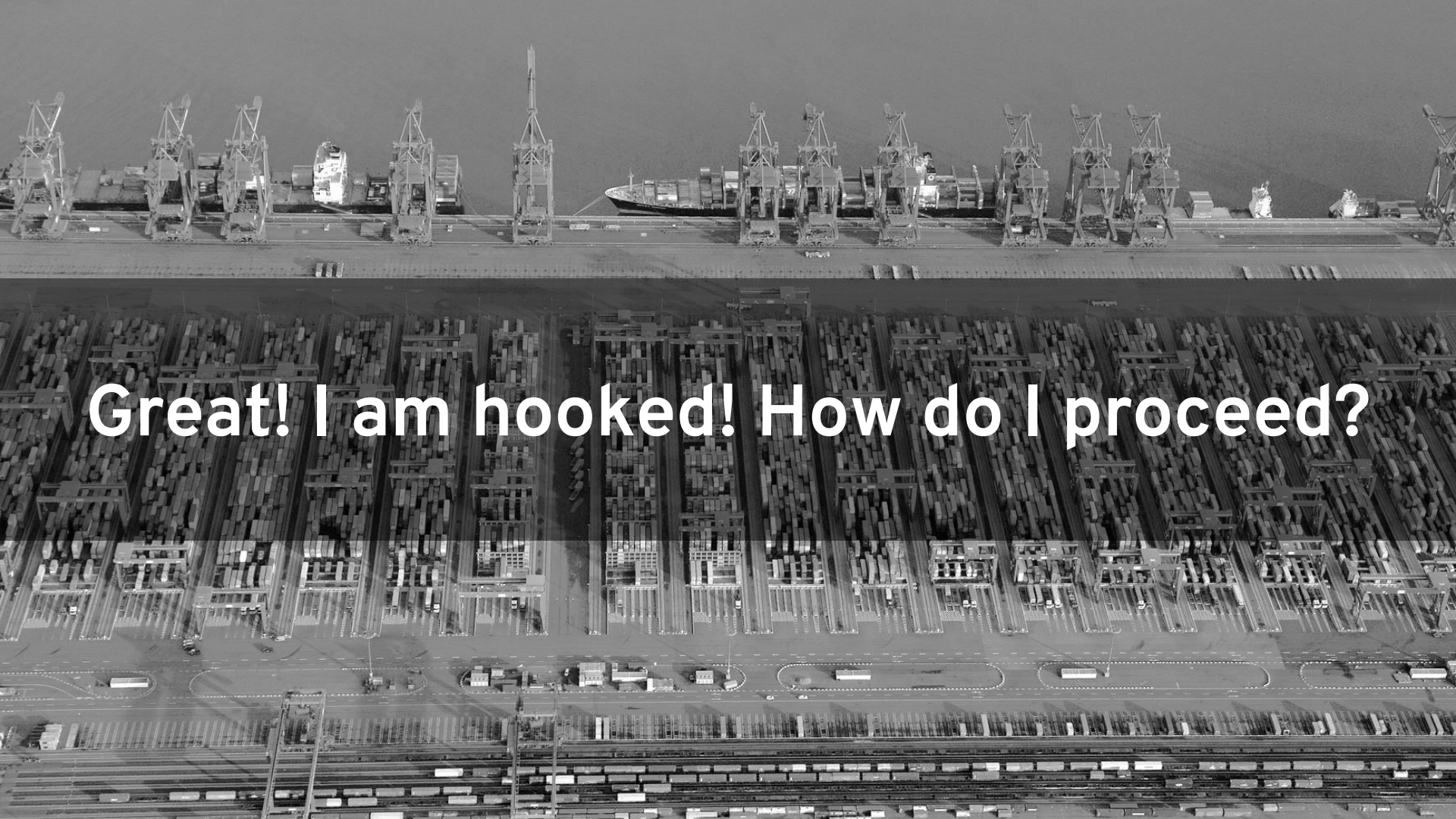


Source: CNCF (Jun, 2019)

ETC-D database

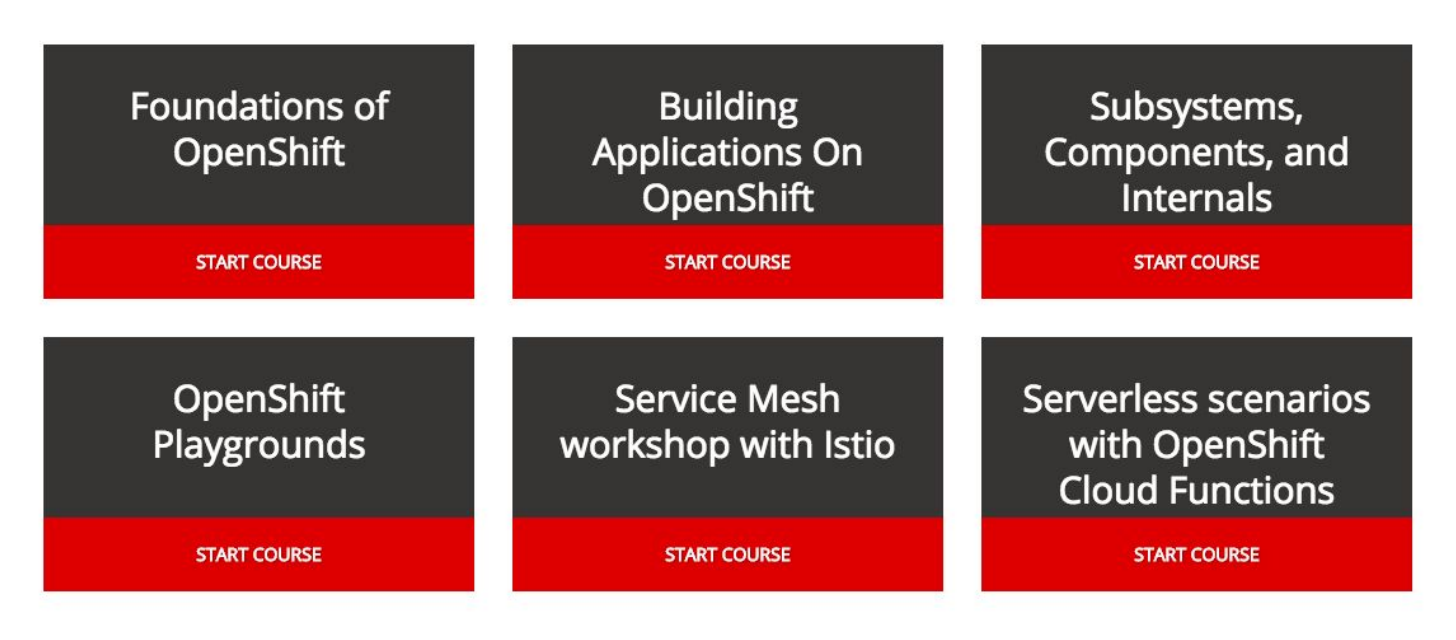


Source: CNCF (Oct, 2019)



Great! I am hooked! How do I proceed?

LEARN.OPENSHIFT.COM



Interactive Learning Scenarios provide you with a pre-configured OpenShift instance, accessible from your browser without any downloads or configuration.

RED HAT SERVICES FOR OPENSHIFT ADOPTION

RED HAT OPEN INNOVATION LABS

EXPERIMENT
Rapidly build prototypes, do DevOps, and be agile.

CATALYZE INNOVATION
Bring modern application development back to your team.

IMMERSE YOUR TEAM
Work side-by-side with experts in a residency-style engagement.

TO SHOW YOUR TEAMS HOW OPENSHIFT AND MODERN DEVELOPMENT PRACTICES CAN DRIVE INNOVATION: START WITH A 4- TO 12-WEEK LABS RESIDENCY

RED HAT CONTAINER ADOPTION PROGRAM

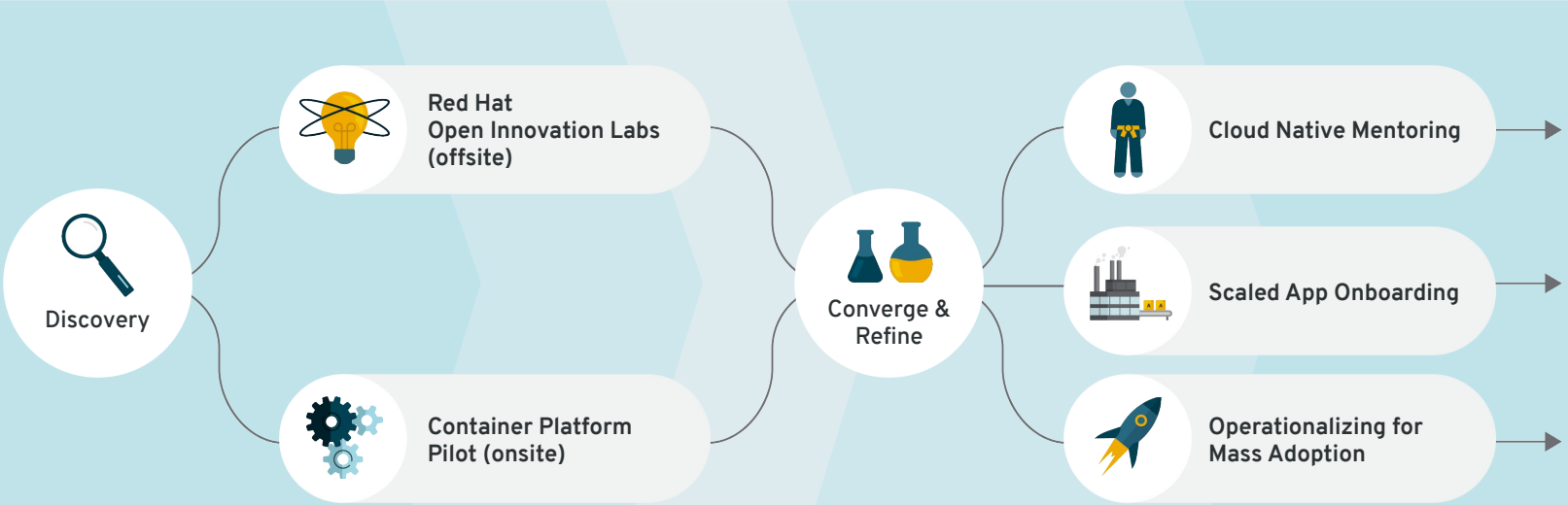
FRAMEWORK FOR SUCCESSFUL CONTAINER ADOPTION AND IT TRANSFORMATION:

Mentoring, training, and side-by-side collaboration to:

- Create a production platform and team to run it
- Create end-to-end container-driven deployment automation
- Scale application onboarding expertise
- Guide new Kubernetes-native development
- Align business with IT through **included Red Hat Open Innovation Labs**

TO BEGIN A COMPREHENSIVE PROGRAM (INCLUDING OPEN INNOVATION LABS): START WITH THE 12-WEEK RED HAT CONSULTING CONTAINER PLATFORM PILOT

CONTAINER ADOPTION PROGRAM



ESTIMATED PAYBACK TIME: 17 months [1]
TOTAL THREE-YEAR BENEFITS NPV: $10.1 M (USD) [1]

[1] The Total Economic Impact of Red Hat Consulting's Container Adoption Program and Open Innovation Labs, Forrester, 2018.

THANK YOU