redhat.

OpenShift on Microsoft Azure Workshop

Eric Webb, Cardinal Solutions

AGENDA

8:30 – 9:00	Breakfast
9:00 – 9:10	Welcome – Jeff Ekstrom, Red Hat
9:10 – 9:45	Intro to Docker, Containers, Kubernetes, Microservices – Eric Webb
9:45 – 10:00	Client tools setup and overview of labs 1-5 – Eric Webb
10:00 – 10:45	Labs 1-5
10:45 – 11:00	Break
11:00 – 11:30	OpenShift Core Concepts – Eric Webb
11:30 – 12:15	Lunch
12:15 – 1:00	OpenShift product options - Jeff Ekstrom
1:00 – 1:15	Break
1:15 – 1:20	Overview of labs 6-10
1:20 – 2:00	Labs
2:00 – 2:45	CI/CD with Jenkins - Samuel De Lima Tauil
2:45 – 2:55	Break
2:55 – 3:00	Overview of labs 11-15
3:00 - 3:40	Labs
3:40 - 3:50	Getting started w/ OpenShift on your Azure subscription – Eric Webb
3:50 - 4:00	Wrap-up, survey, & raffle - Eric Webb

cardinal

Cardinal provides creative technology solutions that transform client visions into compelling business and customer experiences.

- 1996 EST. IN
- 7 LOCATIONS
- 450 CONSULTANTS
- 65+M REVENUE
- NATIONAL FOOTPRINT

Cloud

Mobile

Data

Web

Microsoft cloudera Pivotal redhat sitecore MicroStrategy +ableau

MICROSOFT PARTNERSHIP

Microsoft Partner Since 2001

Microsoft National Solution Provider Program

Competencies

- Application Development
- Cloud Platform
- Cloud Productivity
- Collaboration and Content
- Data Analytics
- Data Platform
- DevOps
- Enterprise Mobility
- Project, Program, Portfolio Mgmt

2017 Partner of the Worldwide Year

Open Source on Azure

2017 East Region SMS&P Partner of the Year

2015 Central Region Partner of the Year

Build the Intelligent Cloud

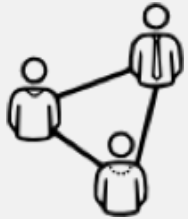

CONTAINERS AND MICROSERVICES:
BUSINESS VALUE OF DEVOPS
redhat.

Digital Transformation

There is Evolution in...

APPLICATIONS

New ways of developing, delivering and integrating applications



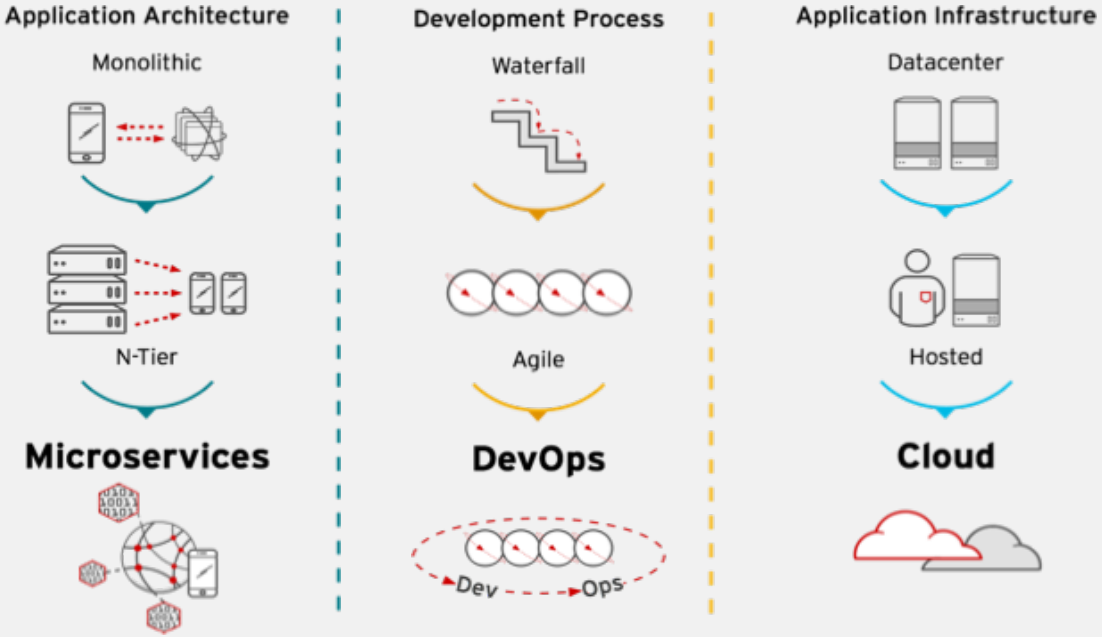
PROCESS

More agile processes across both IT and the business

INFRASTRUCTURE

Modernize existing and build new cloud based infrastructure

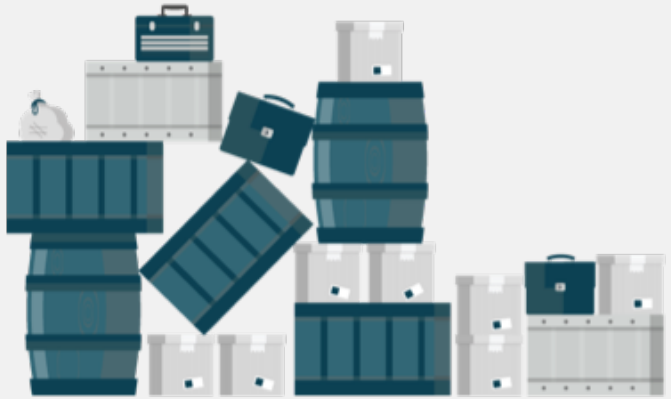
EVOLUTION OF APPS, INFRASTRUCTURE & PROCESS



THE PROBLEM

Existing applications require complicated installation and integration every time they are deployed leading to

- Slow service delivery
- Reduced service quality
- Frequent down times

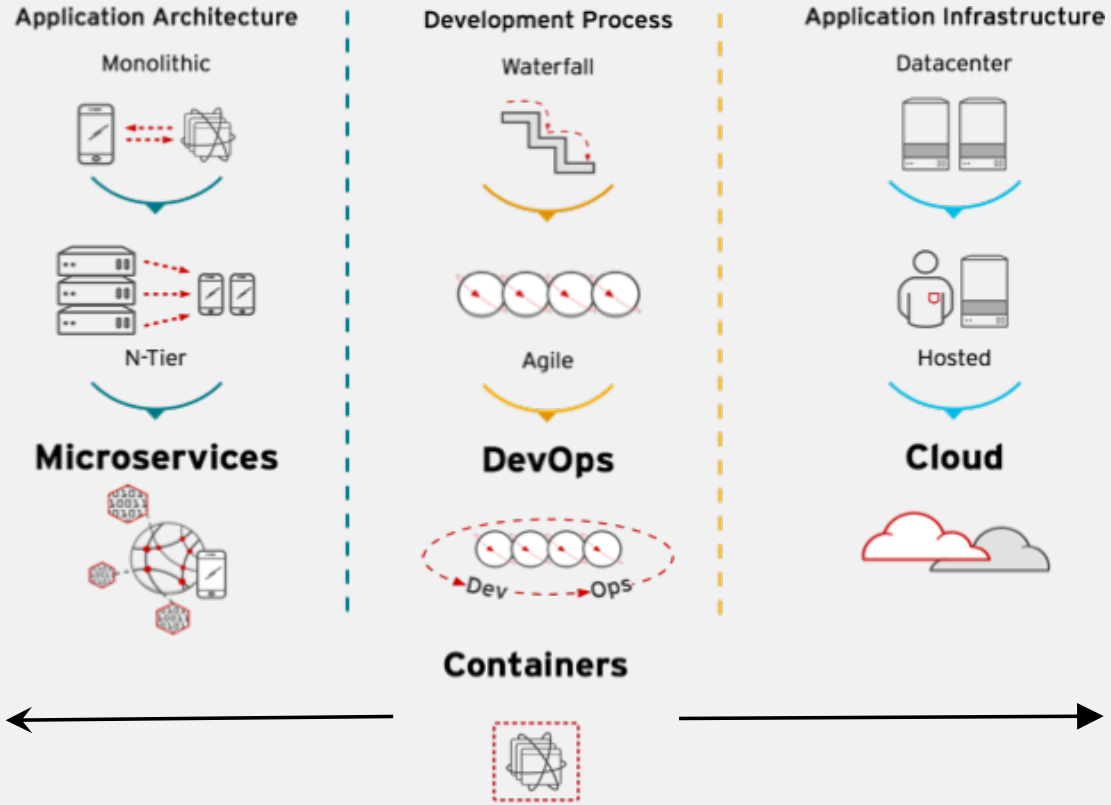


THE SOLUTION

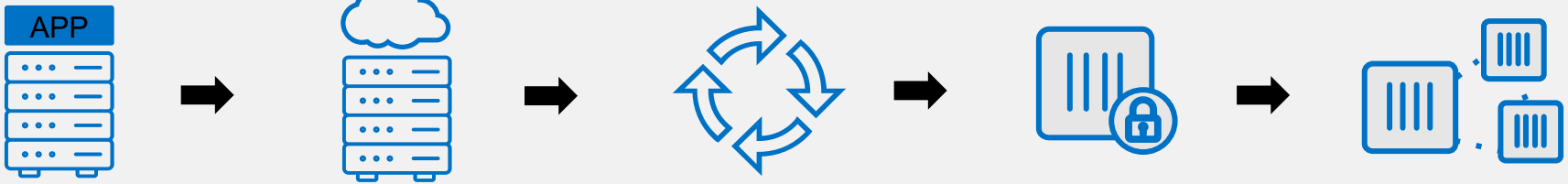
Adopting a container strategy will allow applications to be easily shared and deployed

- Consistent env and tools
- Predictable building blocks
- Faster deployment

CONTAINERS - TRANSFORM APPS, INFRASTRUCTURE & PROCESS



App Modernization



Existing Application

Modern Infrastructure

Move to the cloud as VMs or Containers or refresh HW.

Modern Methodologies

Implement CI/CD and automation.

Containerize Applications

Re-architect apps for scale with containers.

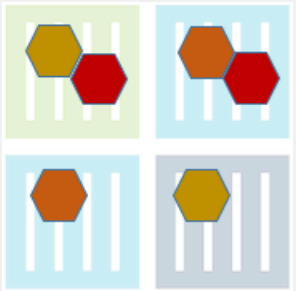
Modern Microservices

Add new services or start peeling off services from monolithic code.


ONE MISCONCEPTION…

CONTAINERS & MICROSERVICES

Microservices ≠ containers



Microservices is a application design paradigm

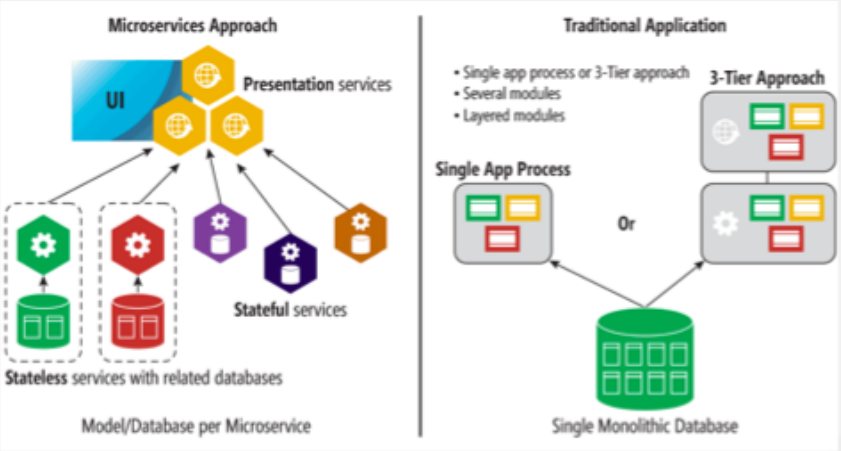
Containers are a implementation detail that can help

WHAT IS A MICROSERVICE?

Architecture style

Structures an application as a collection of loosely coupled services, which implement business

Each service is running in its own process and communicating with lightweight mechanisms like REST



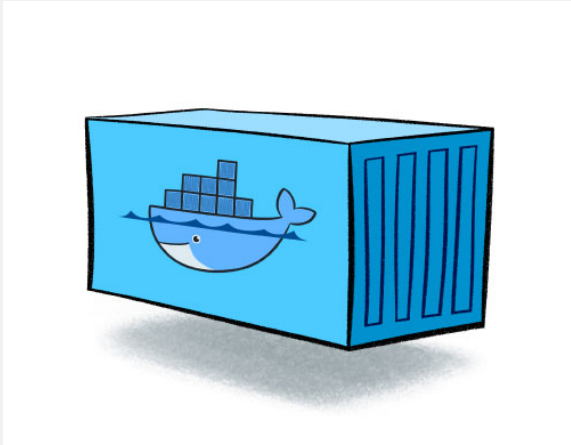
WHAT IS A CONTAINER?

Slice up the OS to run multiple apps on a single VM

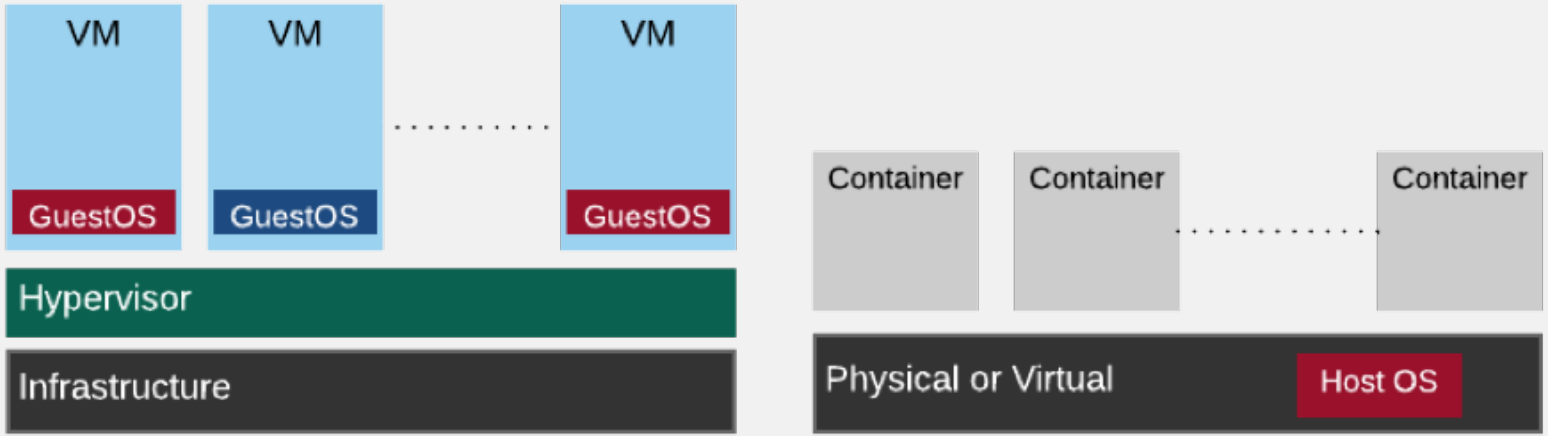
Every container has an isolated view

Shared kernel, very fast start-up, and repeatable execution

Cannot mix OS types on hosts and containers

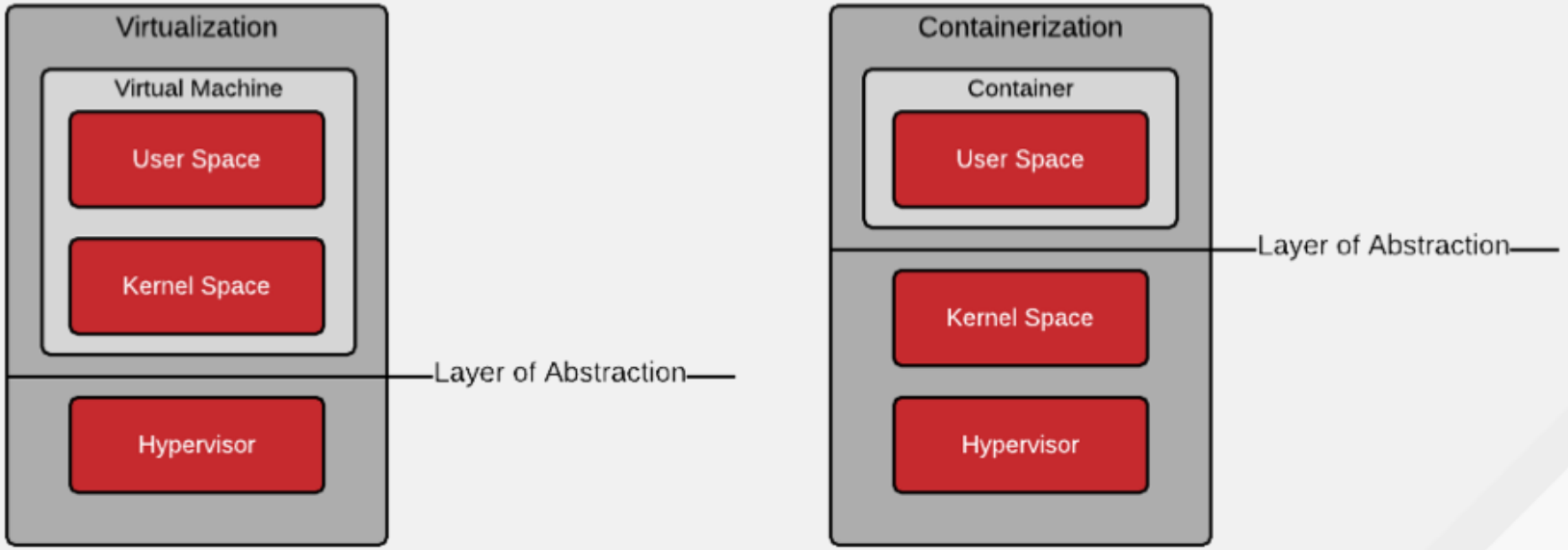


UNDERSTANDING CONTAINERS



Virtual Machines vs Containers

VIRTUALIZATION VS CONTAINERIZATION



WHAT ABOUT DOCKER?

Containers have been around for many years

Linux kernel: cgroups, namespaces

Docker Inc. did not invent them

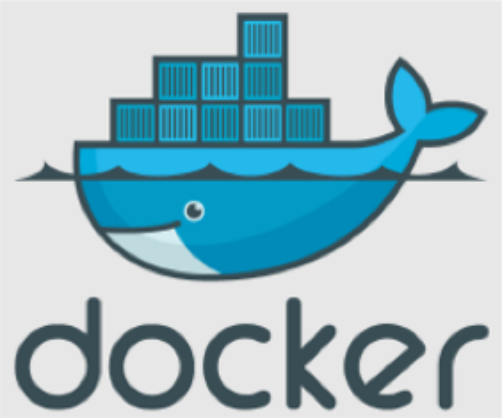
They created open source software to build and manage containers

Docker makes containers easy

Super easy. Fast learning curve

Docker is a container format and a set of tools

Docker CLI, Docker Engine, Docker Swarm, Docker Compose, Docker Machine



CONTAINERS ARE GREAT, RIGHT?

Containerized Apps on Docker Hub

900K

Image Pulls on Docker Hub

12B

Source: DockerCon 2017 Keynote

Container Adoption Rate in Production

35%

Source: RightScale 2017 State of the Cloud Report

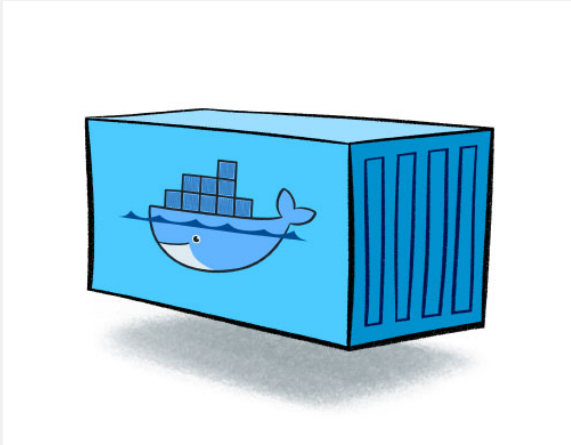
Why the Difference?

IT’S DIFFICULT!!!

CONTAINERS IN DEV

Docker makes it very easy to spin up containers on any OS (Windows, Mac, Linux) for development

It does not, however, help with the availability, scaling, and security requirements of production apps



WE NEED MORE THAN JUST CONTAINERS

Scheduling
Decide where to deploy containers

Lifecycle and health
Keep containers running despite failures

Discovery
Find other containers on the network

Monitoring
Visibility into running containers

Security
Control who can do what

Scaling
Scale containers up and down

Persistence
Survive data beyond container lifecycle

Aggregation
Compose apps from multiple containers


ORCHESTRATION

WHAT IS KUBERNETES?

Background

- "Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications”
- Schedules and runs application containers across a cluster of machines
- Kubernetes v1.0 released on July 21, 2015. Joe Beda, Brendan Burns, & Craig McLuckie

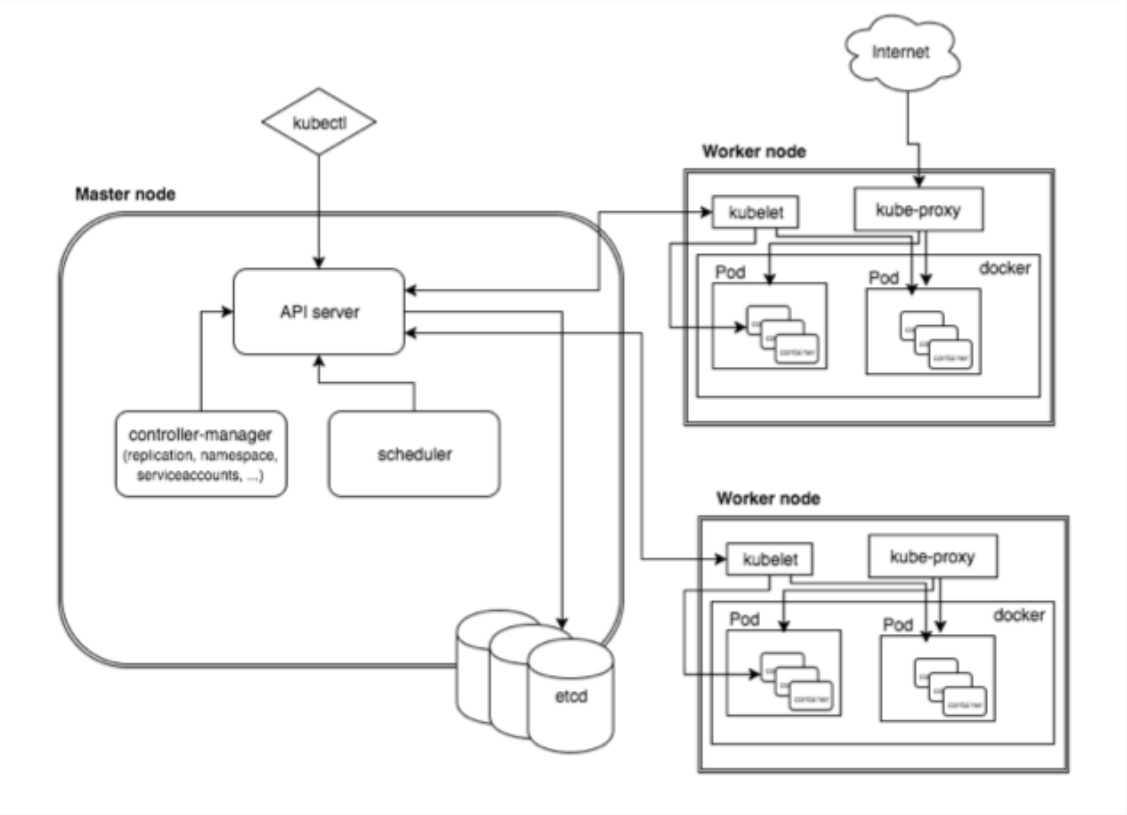
Key features

- Declarative infrastructure
- Self-healing
- Horizontal scaling
- Automated rollouts and rollbacks
- Service discovery and load balancing
- Storage orchestration
- Secret and configuration management

Kubernetes Resources

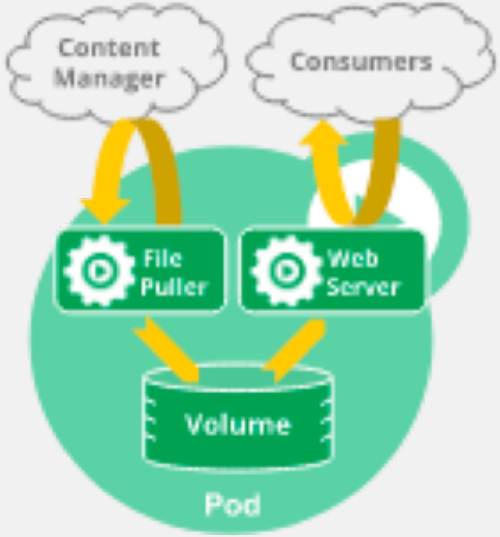
pod	deployment
service	replica set
ingress	daemon set, job
namespace	secret, config-map

Kubernetes Architecture



What is a Pod?

- Pod is the basic building block in kubernetes
- Pods are how containers are delivered
- Can be multiple containers (e.g. - side car)
- Encapsulates container(s), storage, network IP, and options on how to run

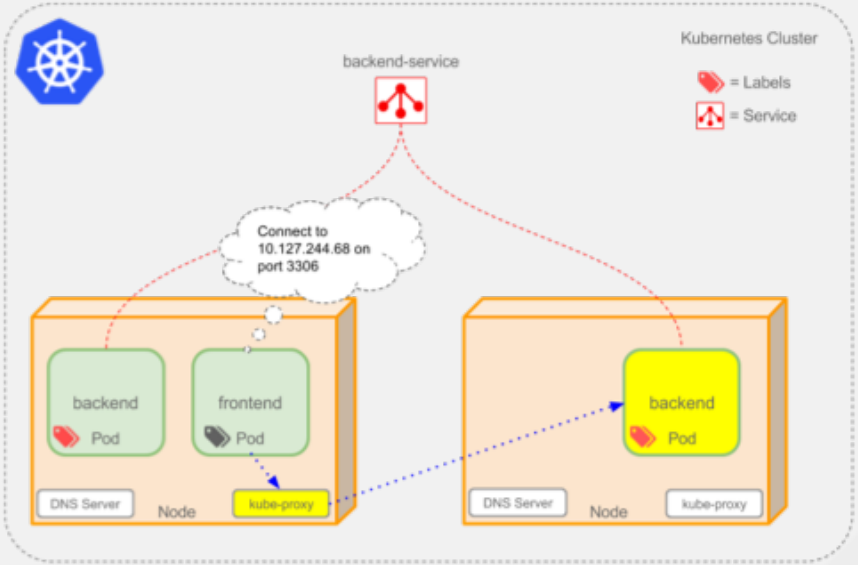


Manifest File: Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: redis-django
  labels:
    app: web
spec:
  containers:
    - name: key-value-store
      image: redis
      ports:
        - containerPort: 6379
    - name: frontend
      image: django
      ports:
        - containerPort: 8000
```

What is a Service?

- Defines a logical set of pods (your microservice)
- Essentially a virtual load balancer in front of pods



Manifest File: Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: web
  type: ClusterIP
  ports:
  - protocol: TCP
    port: 80
    targetPort: 9376
```

Service Types

ClusterIP:
Exposes the service on a cluster-internal IP. Choosing this value makes the service only reachable from within the cluster

NodePort:
Exposes the service on each Node's IP at a static port (the NodePort)
Connect from outside the cluster by requesting <NodeIP>:<NodePort>

LoadBalancer:
Exposes the service externally using a cloud provider's load balancer

Deployments

Provides declarative updates for Pods and Replica Sets

Deployment describes ”desired state”

Can:

- Create deployment to rollout ReplicaSet
- Declare new state for pods (e.g. – new imageTag)
- Rollback to earlier state
- Scale up/down
- Check rollout history
- Clean-up

Manifest File: Deployment

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: ocpweb-deploy
spec:
  replicas: 5
  template:
    metadata:
      labels:
        app: ocpweb
    spec:
      containers:
      - name: ocpweb
        image: ded101/ocpweb
        ports:
        - containerPort: 8080
```

Namespaces

Allow for multiple virtual clusters backed by the same physical cluster

Logical separation

Namespace used in FQDN of Kubernetes services

E.g. - <service-name>.<namespace-name>.svc.cluster.local

Every Kubernetes resource type is scoped to a namespace (except for nodes, persistentVolumes, etc.)

Intended for environments with many users, teams, projects

Labels and Selectors

Not related to CSS

Labels are key/value pairs for any API object in Kubernetes

”Label selectors” == queries against labels to match objects

Use cases:

- Associating pods to a service
- Pinning workloads to specific nodes
- Selecting a subset of resources


Labs 1-5
redhat.

Labs 1-5 Overview

Link to the labs:
http://labs.apps.ocp.cloudvillage.in/#/workshop/ocptigerteam/module/0-setting-up-client-tools

Labs

1. Setting up client tools
2. Create an App from a Docker image
3. Create an App using Docker build
4. Using Web Console
5. Creating an application using JBoss EAP builder image


OPENSHIFT OVERVIEW AND
ARCHITECTURE REVIEW
redhat.

Container application platform based on Docker and Kubernetes for building, distributing and running containers at scale

RED HAT CONTAINER SOLUTIONS

HOST RED HAT® ENTERPRISE LINUX® ATOMIC HOST

MANAGEMENT RED HAT® CLOUDFORMS

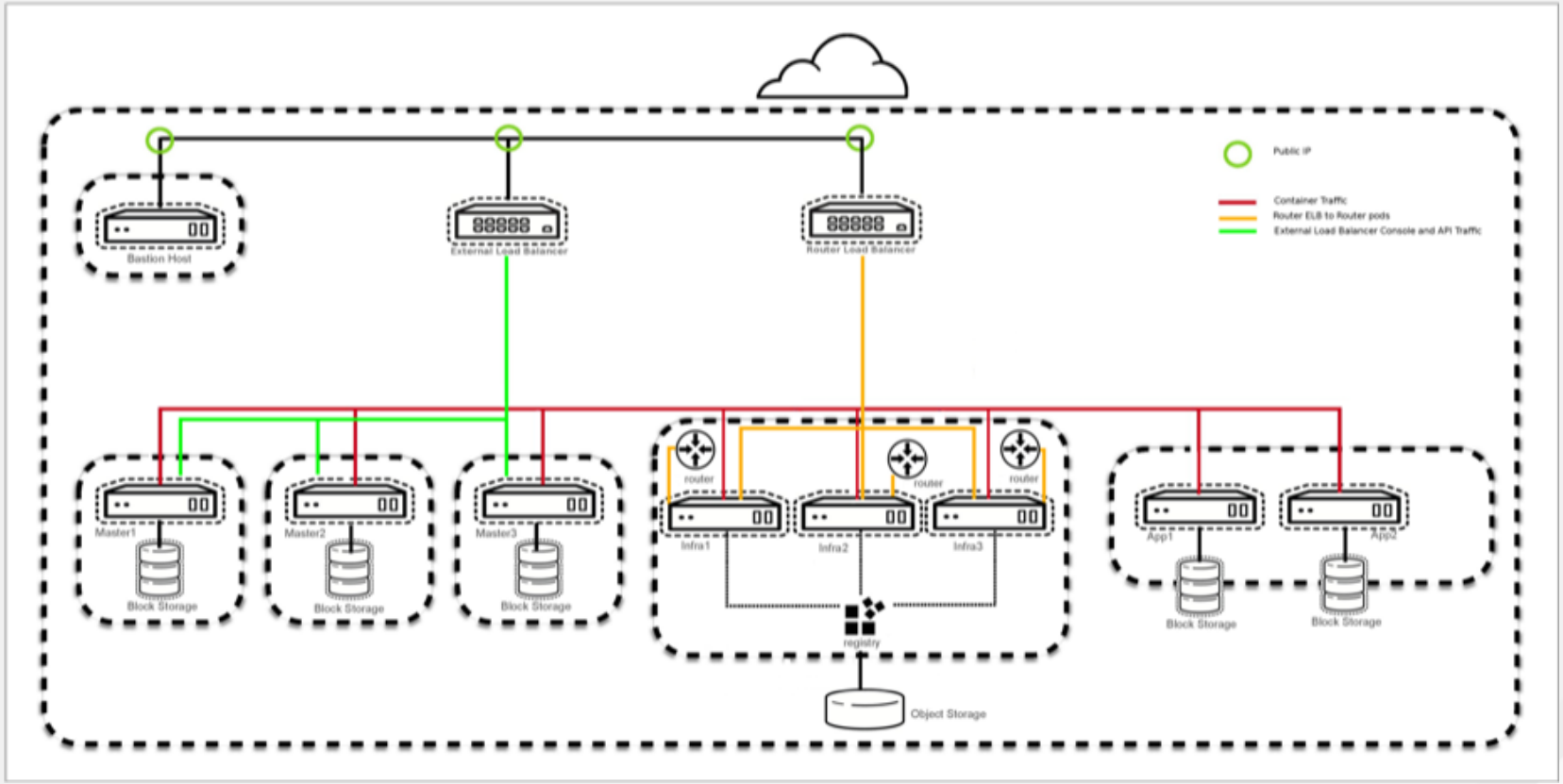
PLATFORM RED HAT® OPENSHIFT

STORAGE RED HAT® STORAGE

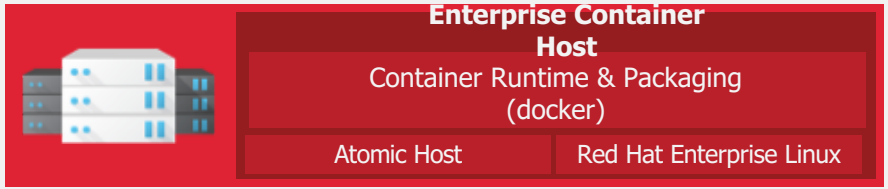

Self-Service
Multi-language
Automation
Collaboration
Multi-tenant
RED HAT
OPENSHIFT
Standards-based
Web-scale
Open Source
Enterprise Grade
Secure

Community Powered Innovation

Reference Architecture

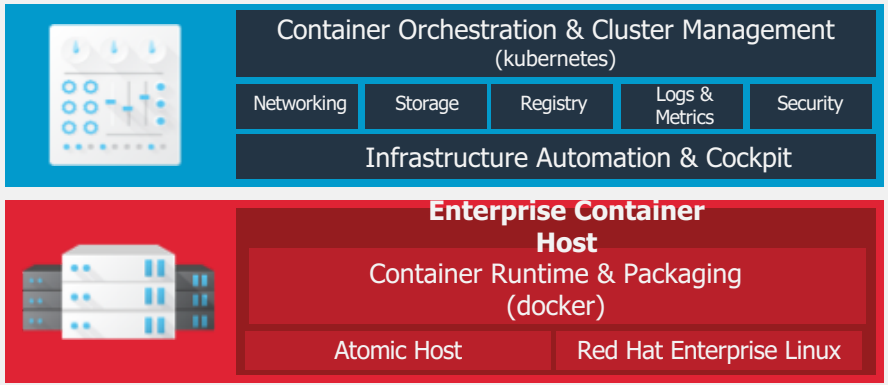


OPENSHIFT CONTAINER PLATFORM



Trusted by Fortune Global 500 companies

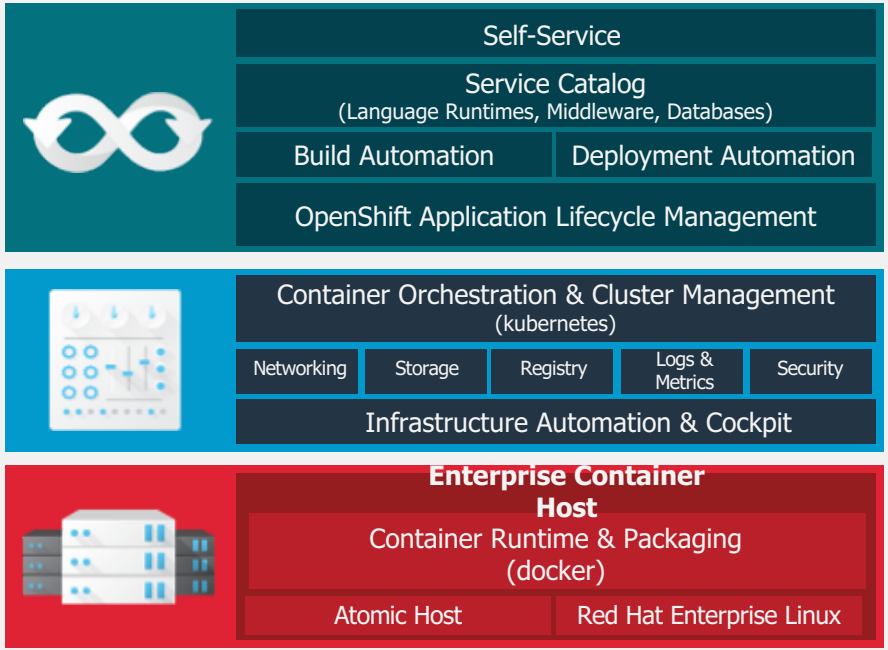
OPENSHIFT CONTAINER PLATFORM



Enterprise Kubernetes++ container orchestration

Trusted by Fortune Global 500 companies

OPENSHIFT CONTAINER PLATFORM

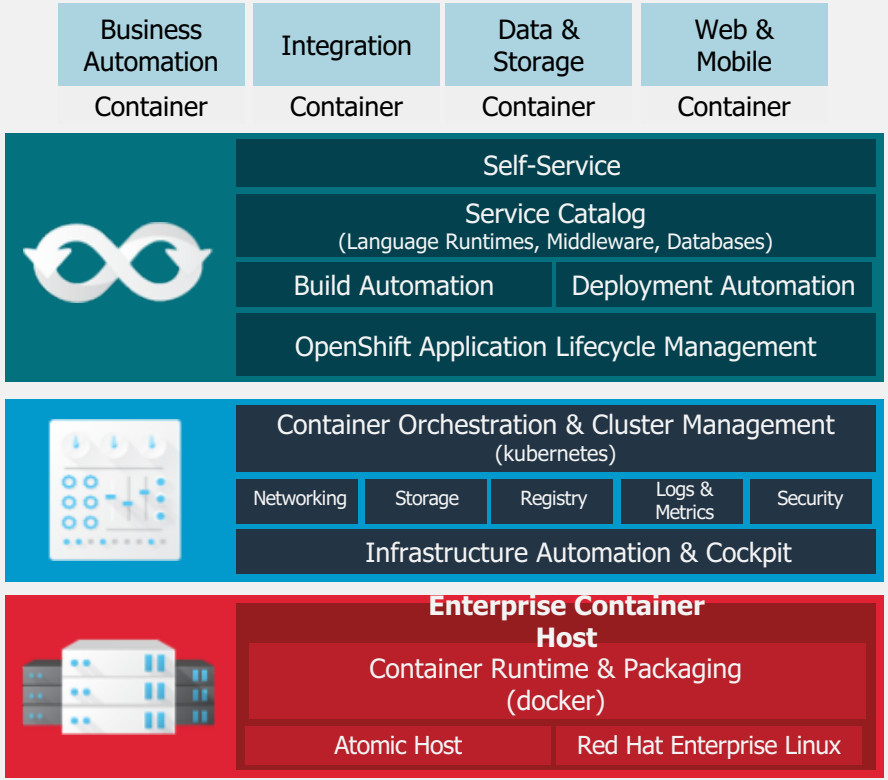


Developer Experience

Enterprise Kubernetes++ container orchestration

Trusted by Fortune Global 500 companies

OPENSHIFT CONTAINER PLATFORM



Traditional, stateful, and cloud-native apps

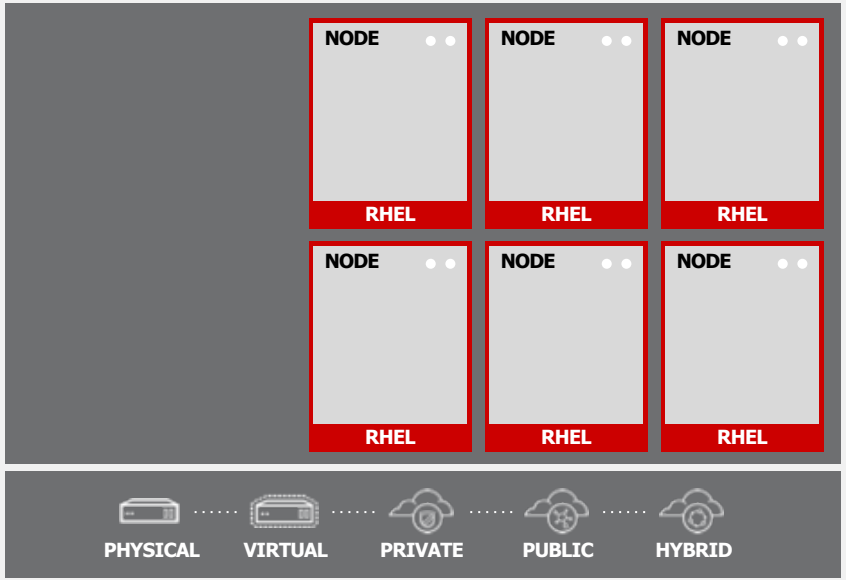
Developer Experience

Enterprise Kubernetes++ container orchestration

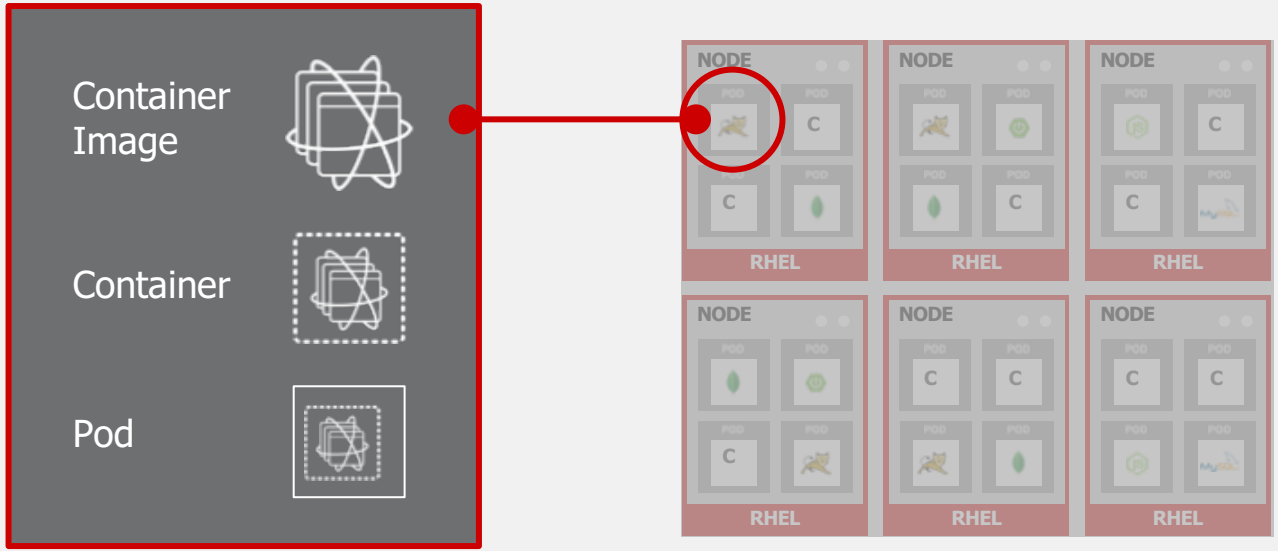
Trusted by Fortune Global 500 companies

YOUR CHOICE OF INFRASTRUCTURE

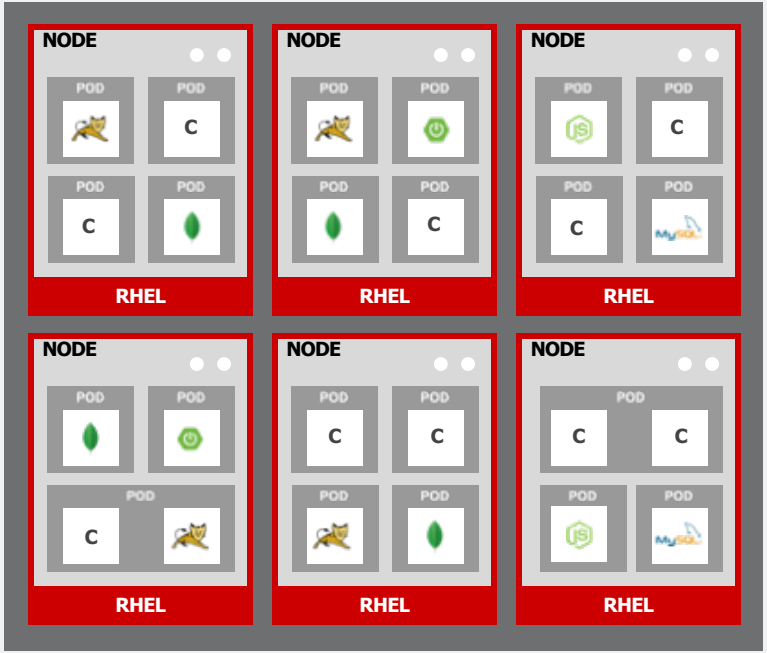
NODES RHEL INSTANCES WHERE APPS RUN



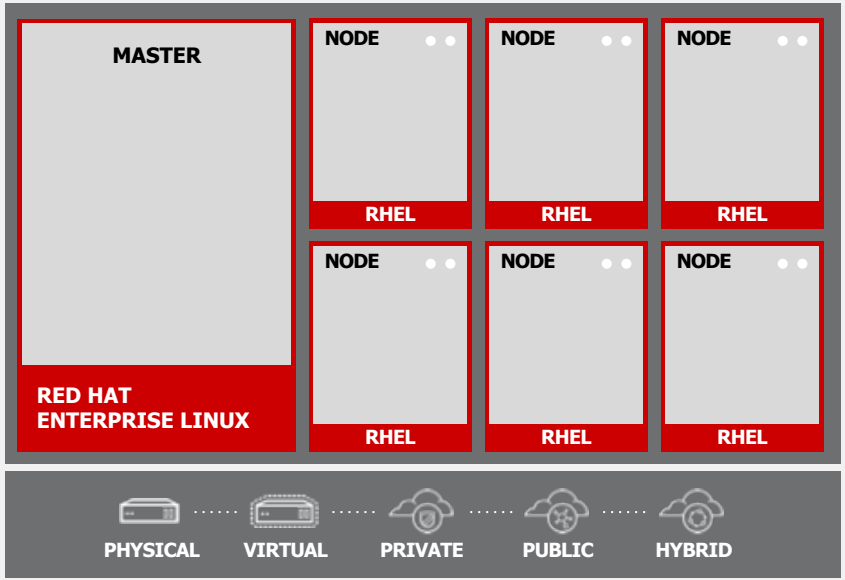
APPS RUN IN CONTAINERS



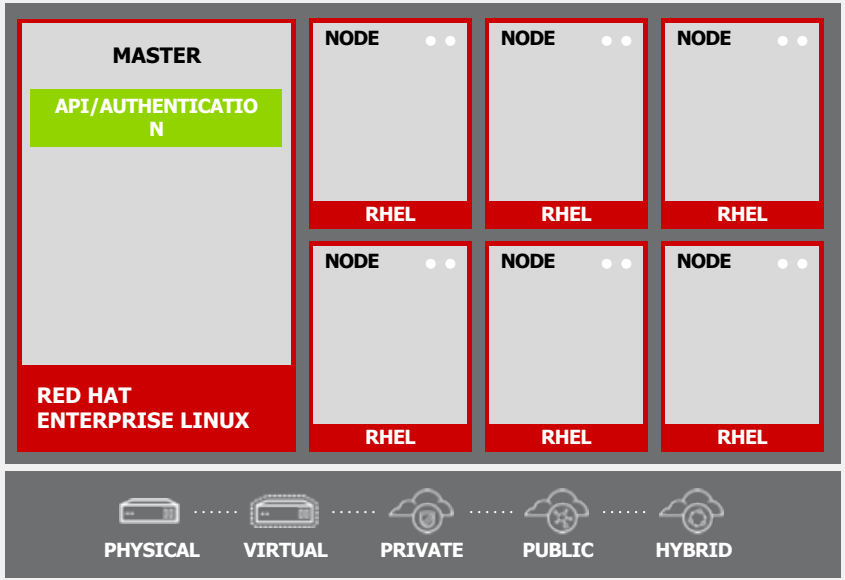
PODS ARE THE UNIT OF ORCHESTRATION



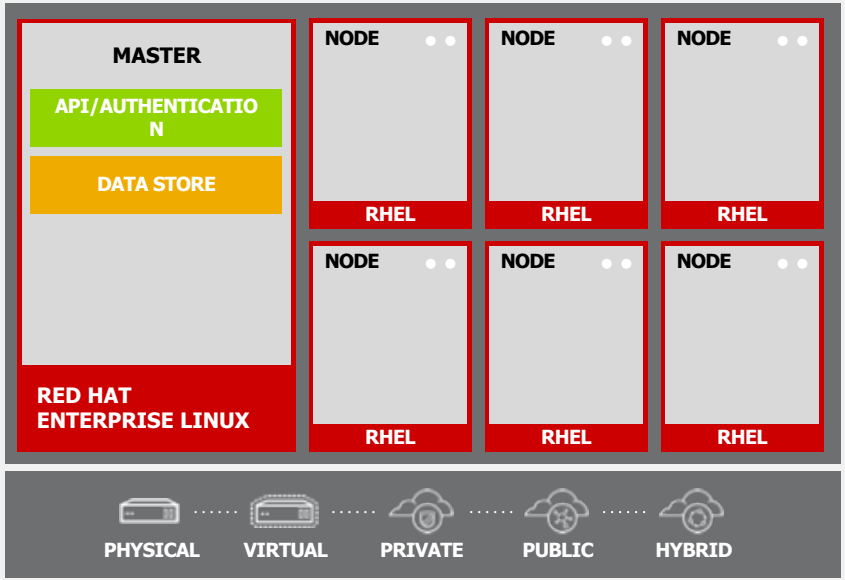
MASTERS ARE THE CONTROL PLANE



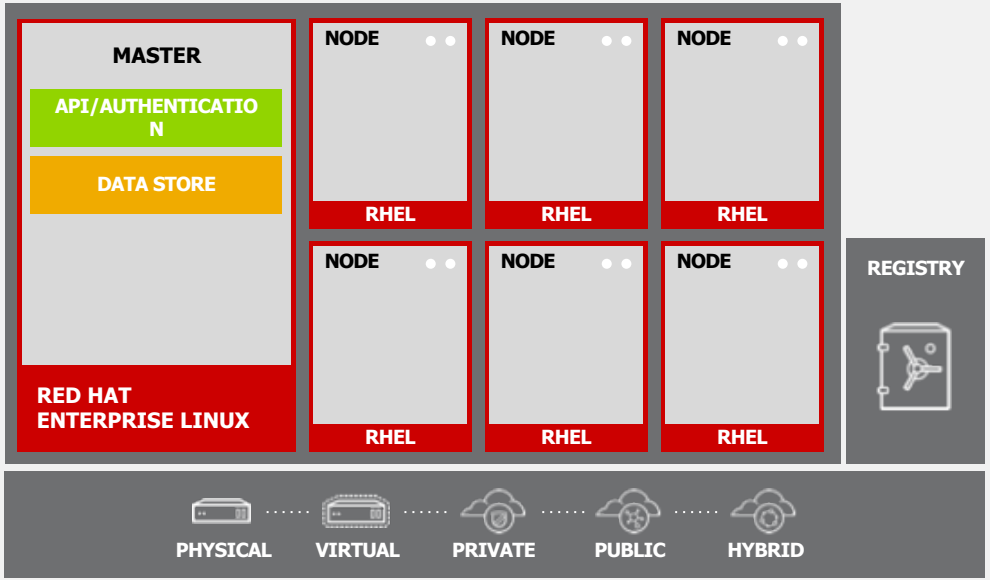
API AND AUTHENTICATION



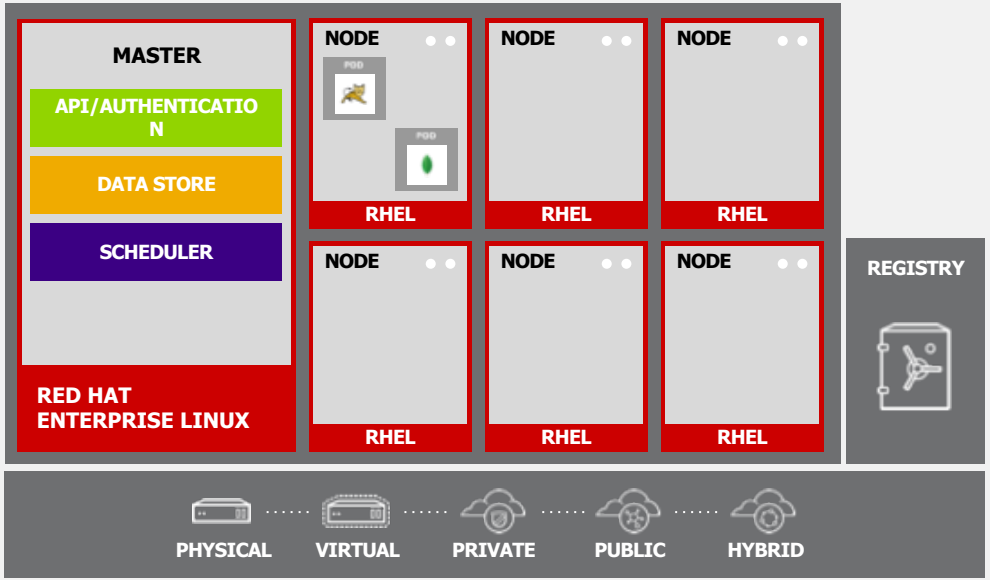
DESIRED AND CURRENT STATE



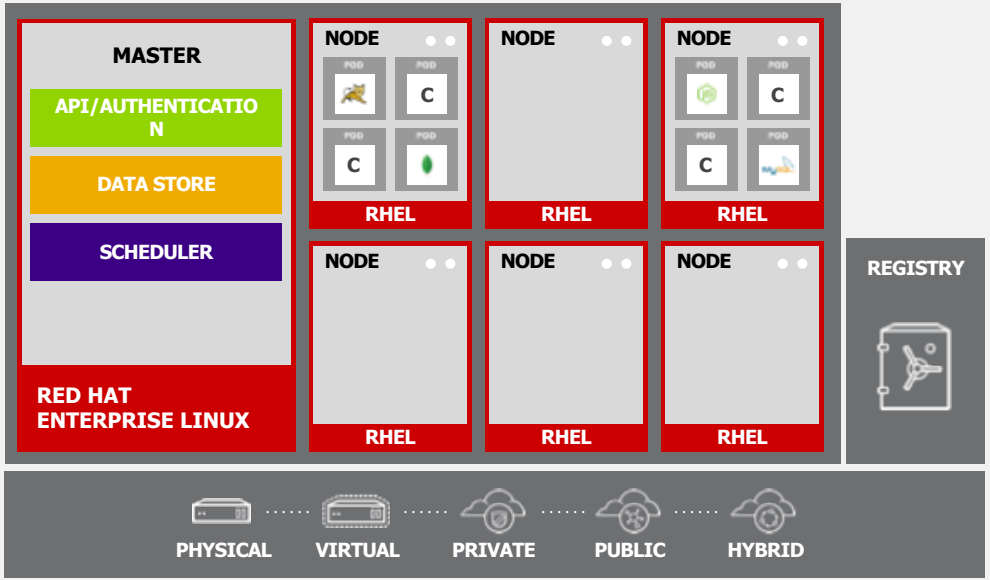
INTEGRATED CONTAINER REGISTRY



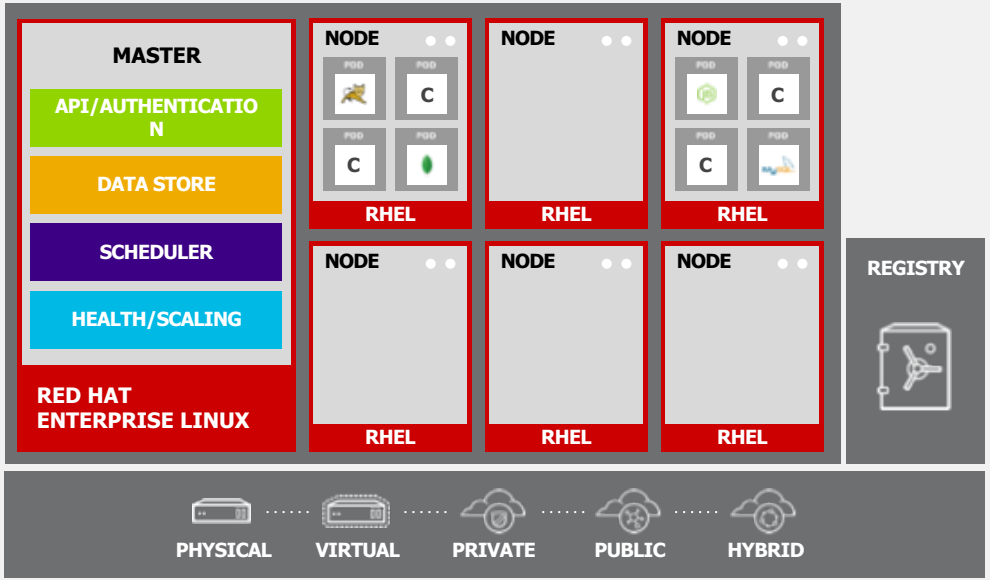
ORCHESTRATION AND SCHEDULING



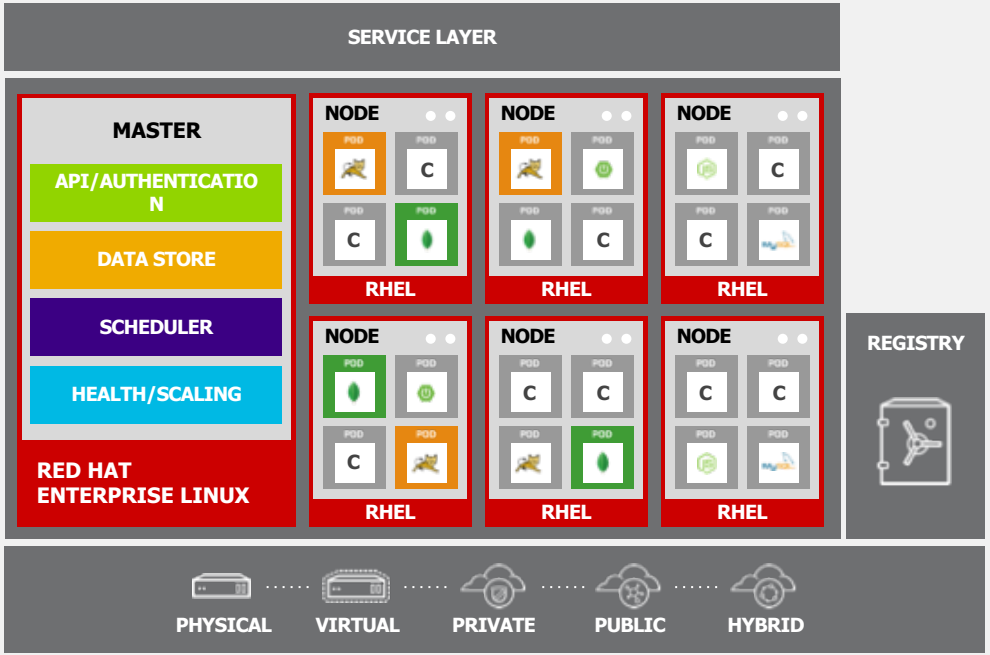
PLACEMENT BY POLICY



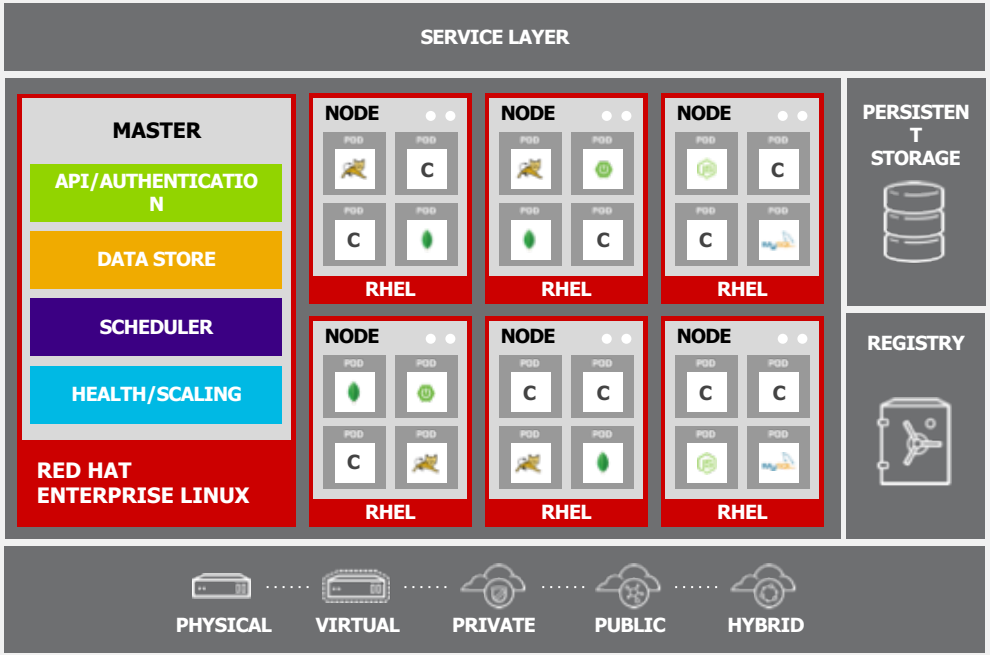
AUTOSCALING PODS



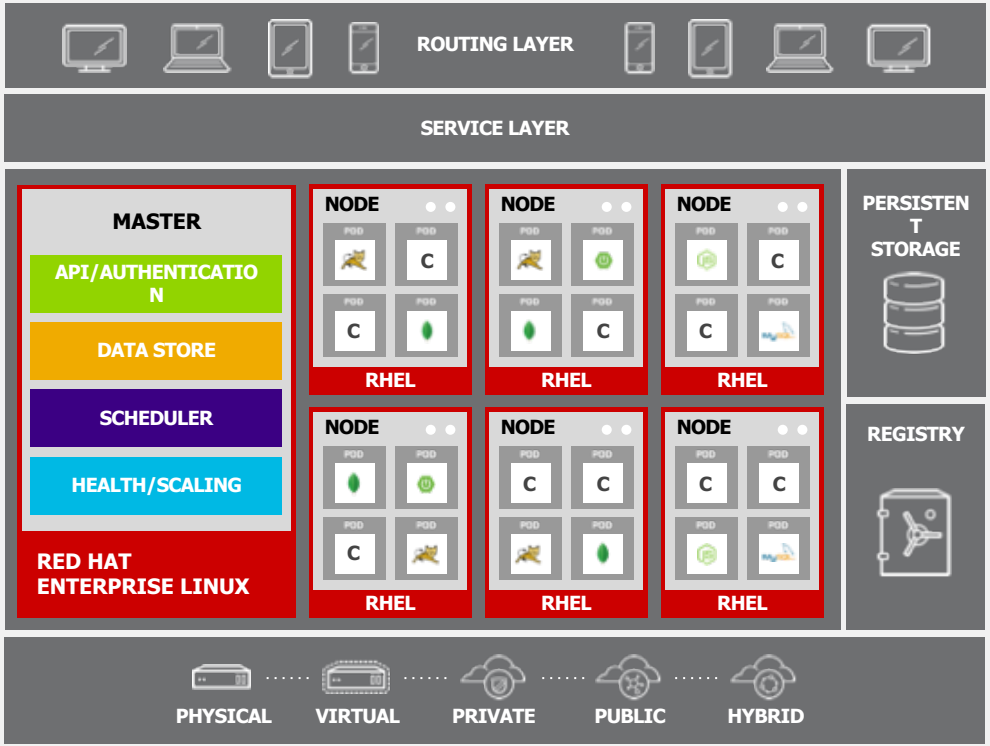
SERVICE DISCOVERY



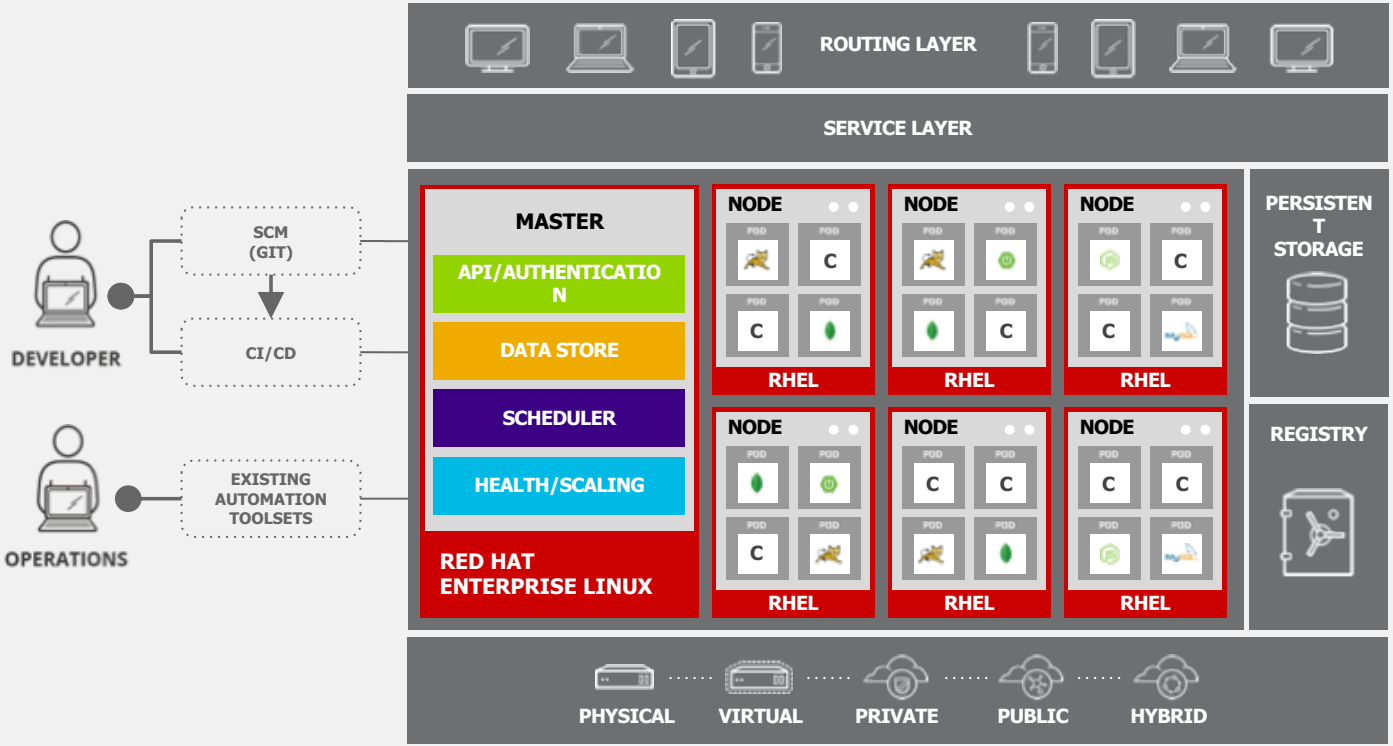
PERSISTENT DATA IN CONTAINERS

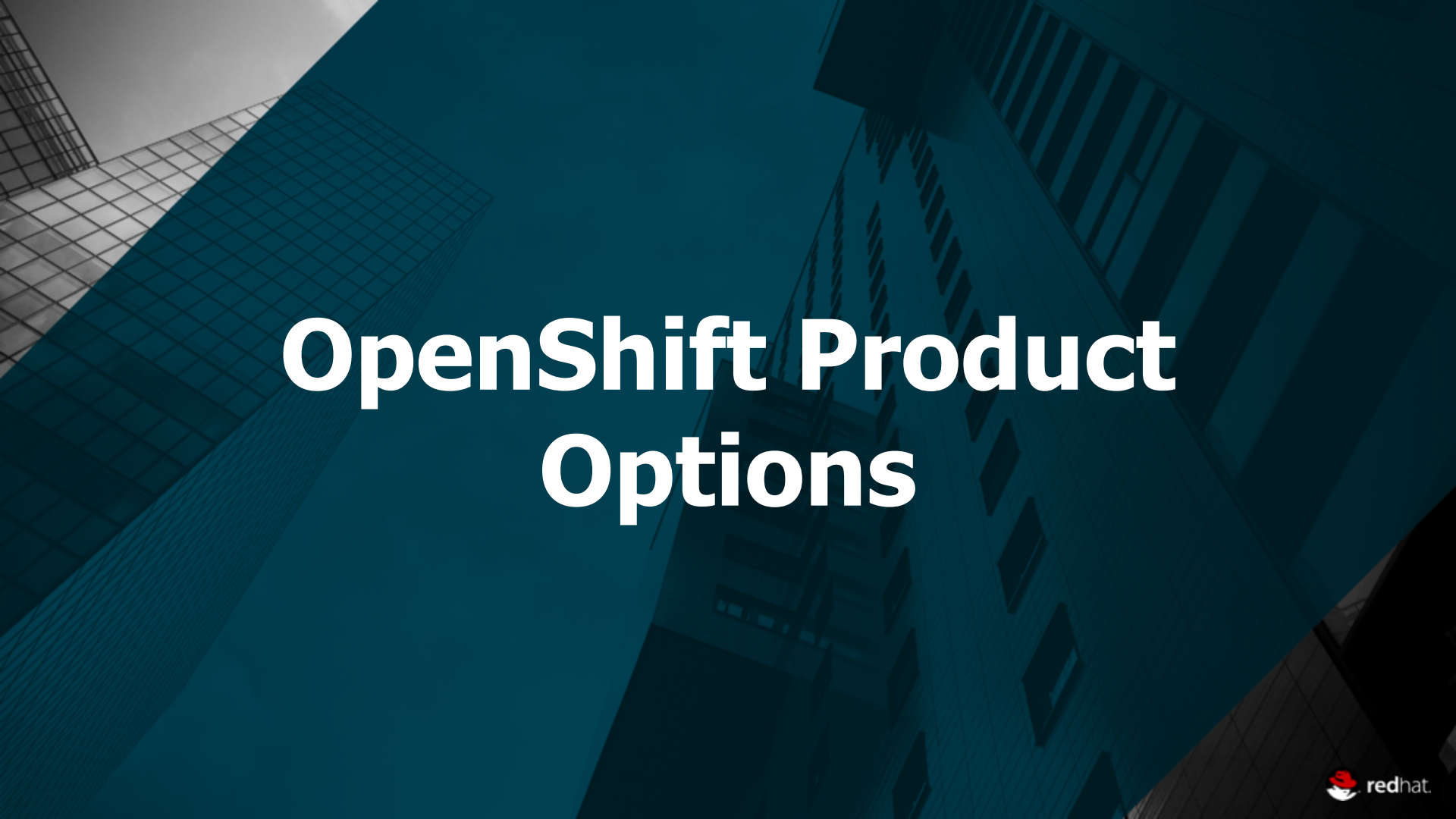


ROUTING AND LOAD-BALANCING



ACCESS VIA WEB, CLI, IDE AND API




OpenShift Product
Options

OpenShift Container Platform
Based on Docker and
Kubernetes for building,
distributing and running
containers at scale

This is the actual OpenShift bits

RED HAT OPENSHIFT DEDICATED

Enterprise Public Cloud

The power and flexibility of your own scalable OpenShift 3 cluster, installed and backed by the experience of Red Hat Engineering, Operations, and Support.

https://www.openshift.com/dedicated/

redhat.

Get Involved With Zero Commitment

Quickly build, host, and scale containerized applications in the public cloud, operated and supported by Red Hat.

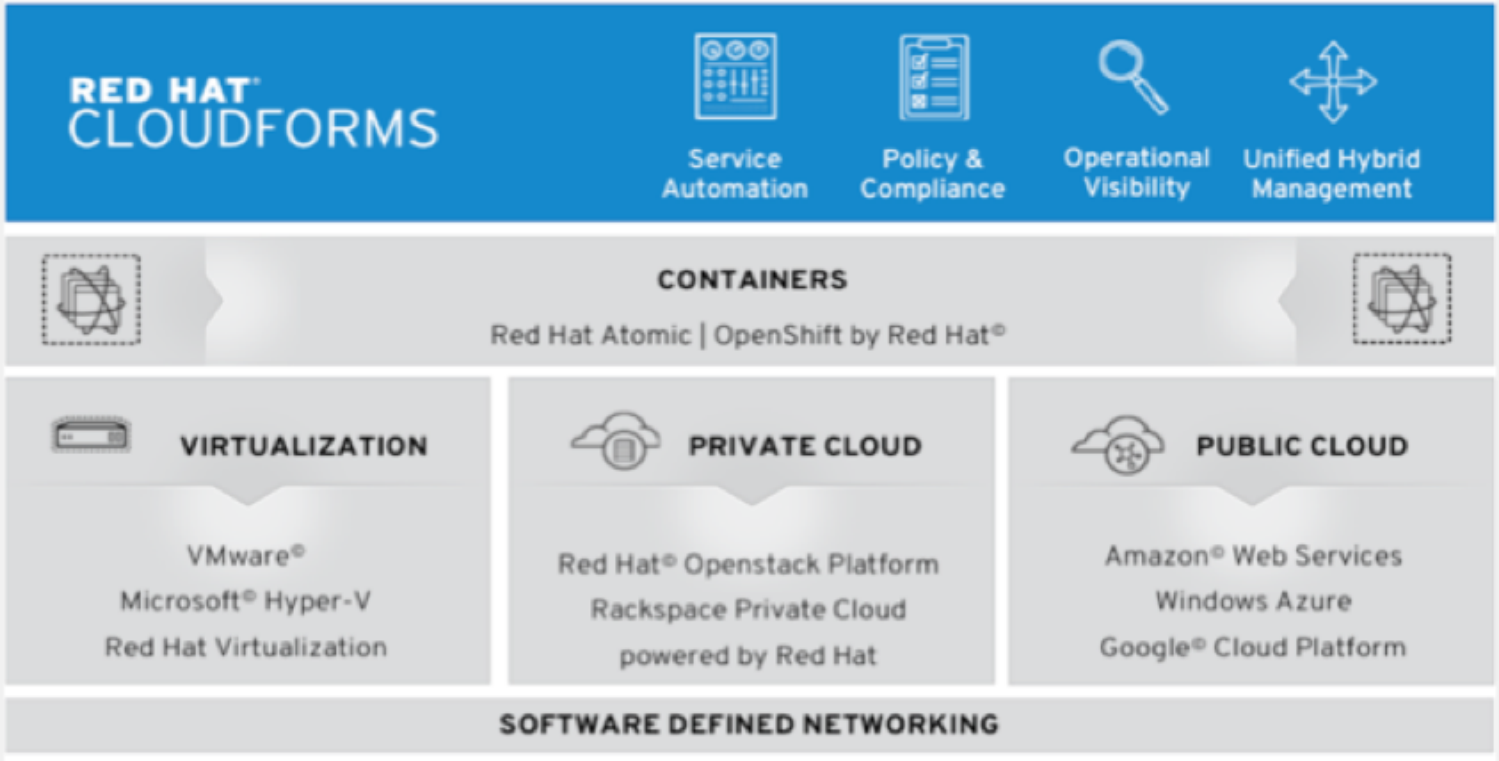
https://www.openshift.com/

Where users, partners, customers, and contributors come together to collaborate and work together on OpenShift.

https://commons.openshift.org/

Origin is the upstream community project that powers OpenShift.

https://openshift.org/


RED HAT®
CLOUDFORMS
Service Automation
Policy & Compliance
Operational Visibility
Unified Hybrid Management
CONTAINERS
Red Hat Atomic | OpenShift by Red Hat©
VIRTUALIZATION
VMware©
Microsoft© Hyper-V
Red Hat Virtualization
PRIVATE CLOUD
Red Hat© Openstack Platform
Rackspace Private Cloud
powered by Red Hat
PUBLIC CLOUD
Amazon© Web Services
Windows Azure
Google© Cloud Platform
SOFTWARE DEFINED NETWORKING

Cloud Forms

Key Product Features

AGENTLESS

EASY DEPLOYMENT

ANSIBLE AUTOMATION

SIMPLE, POWERFUL, AGENTLESS

MULTI-TENANCY AND RBAC

SEGMENT USER ACCESS,
FINE GRAINED ACCESS CONTROL

CONTINUOUS DISCOVERY

BROWN-FIELD MANAGEMENT,
INTEROPERATES WITH OTHER MGMT

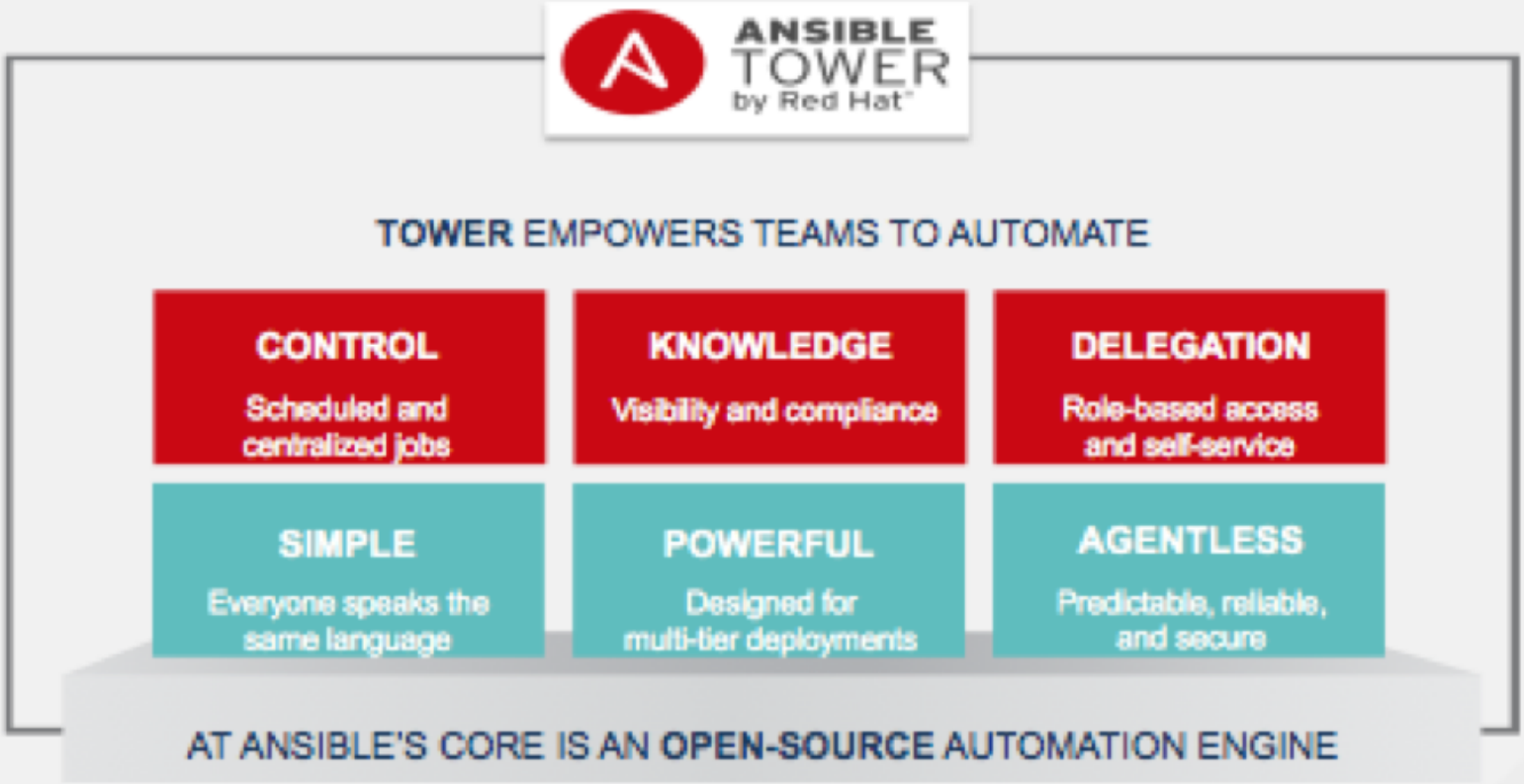
VIRTUAL APPLIANCE

EASY INSTALL, EASY MAINTENANCE

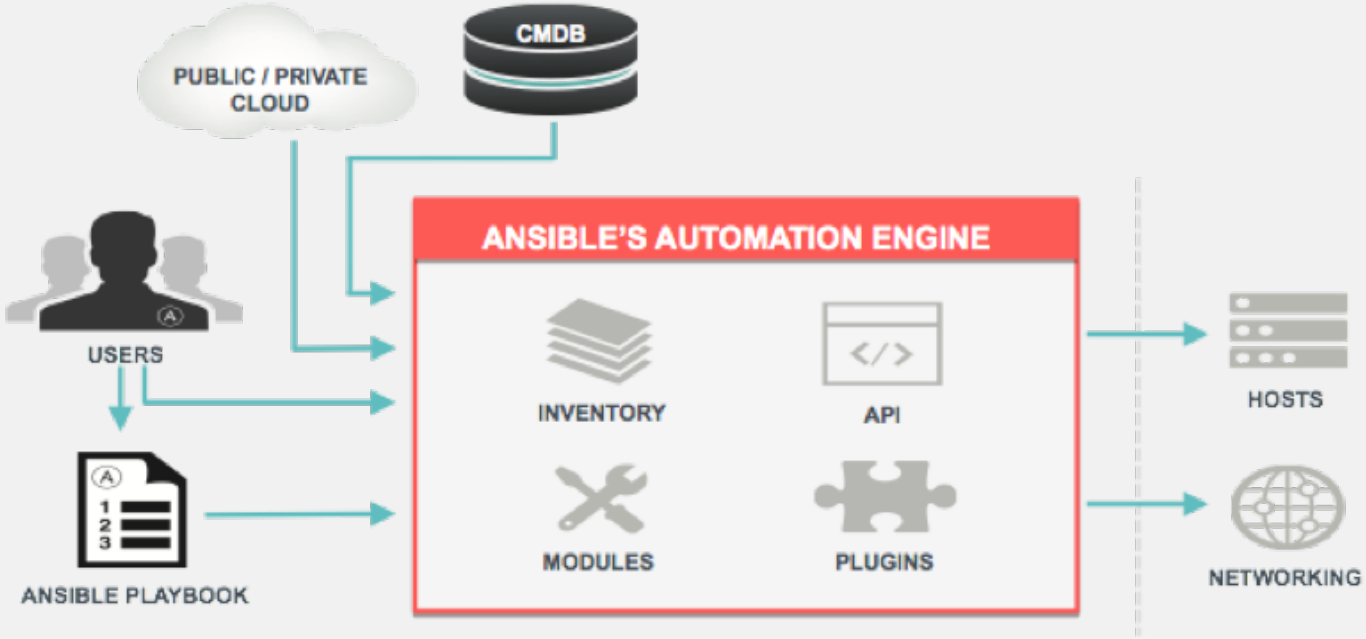
FEDERATED GLOBAL DEPLOYMENTS

HIGHLY SCALABLE, HIGHLY AVAILABLE
MULTI-REGION DEPLOYMENTS

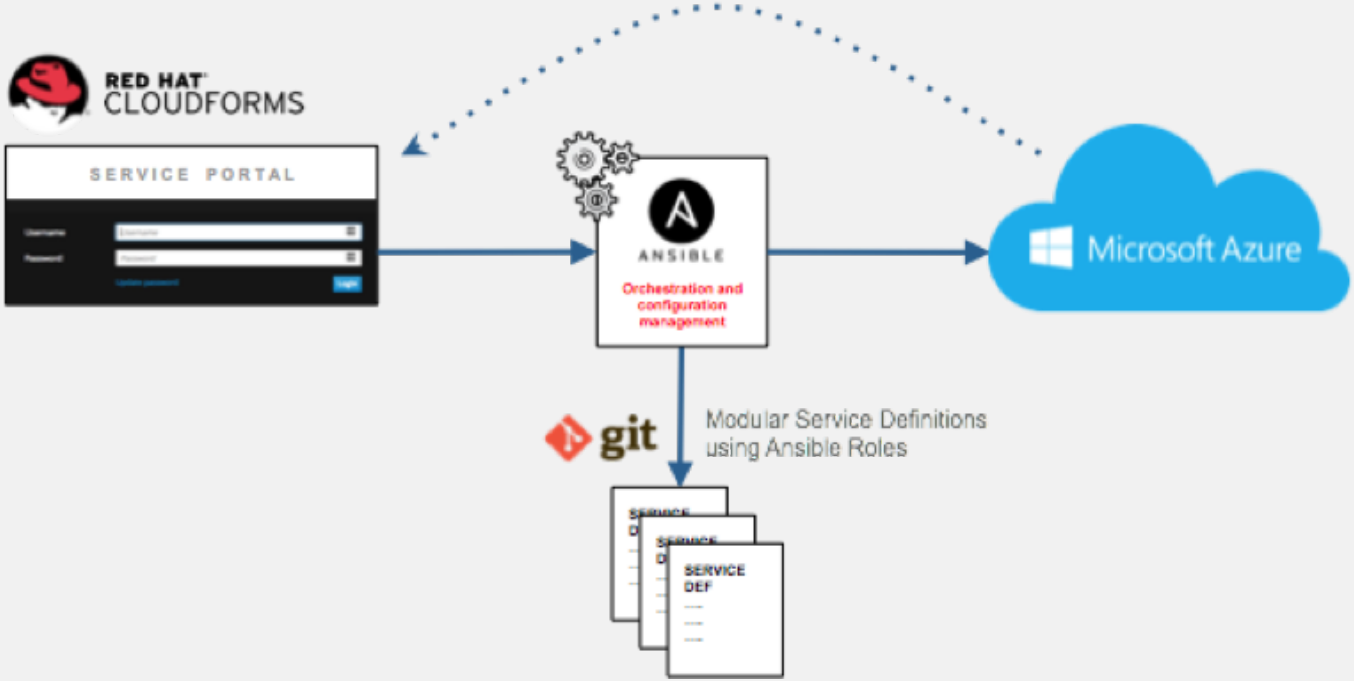
ANSIBLE TOWER



HOW ANSIBLE WORKS



Red Hat Management on Azure



Let’s take a break!


Labs 6-10
redhat.

Labs 6-10 Overview

Link to the labs:
http://labs.apps.ocp.cloudvillage.in/#/workshop/ocptigerteam/module/0-setting-up-client-tools

Labs

6. Using Templates
7. Scale up and Scale down and Idle the application instances
8. Binary Deployment of a war file
9. Using SSL in your application
10. Blue-Green Deployments

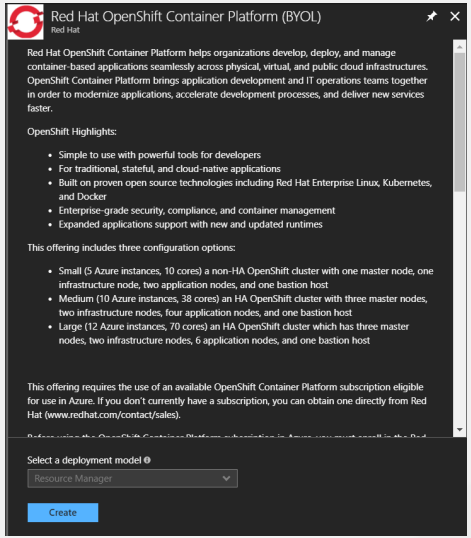

OPENSHIFT ON AZURE QUICK START
redhat.

Getting Started on Azure

Red Hat OpenShift Container Platform (BYOL)

This offering includes three configuration options:

- Small (5 Azure instances, 10 cores) a non-HA OpenShift cluster with one master node, one infrastructure node, two application nodes, and one bastion host
- Medium (10 Azure instances, 38 cores) an HA OpenShift cluster with three master nodes, two infrastructure nodes, four application nodes, and one bastion host
- Large (12 Azure instances, 70 cores) an HA OpenShift cluster which has three master nodes, two infrastructure nodes, 6 application nodes, and one bastion host

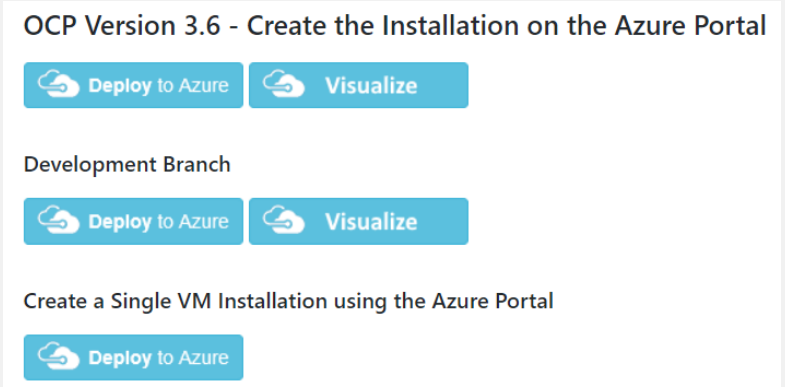


Getting Started on Azure

ARM Templates

https://github.com/openshift/openshift-ansible-contrib/tree/master/reference-architecture/azure-ansible

- Cant deploy full architecture or single instance VM
- Single click deploy or customize templates to fit your need




Labs 11-15
redhat.

Labs 11-15 Overview

Link to the labs:
http://labs.apps.ocp.cloudvillage.in/#/workshop/ocptigerteam/module/0-setting-up-client-tools

Labs

11. SCM Web Hooks
12. Rollback Applications
13. Code Promotion across Environments
14. Installing the JBoss Developer Studio
15. Deploy a SpringBoot Application

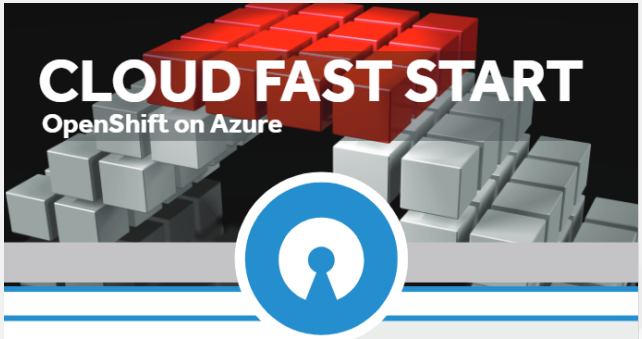
NEXT STEPS

Getting Started

Cardinal OpenShift Fast Start

Cardinal Solutions and Red Hat will guide you through a short-term, structured engagement that will:

- Educate you on OpenShift's capabilities and the value of using containers for development
- Help determine suitable use cases based on your current applications
- Deploy a working OpenShift cluster hosting your chosen workloads



redhat.

THANK YOU

plus.google.com/+RedHat

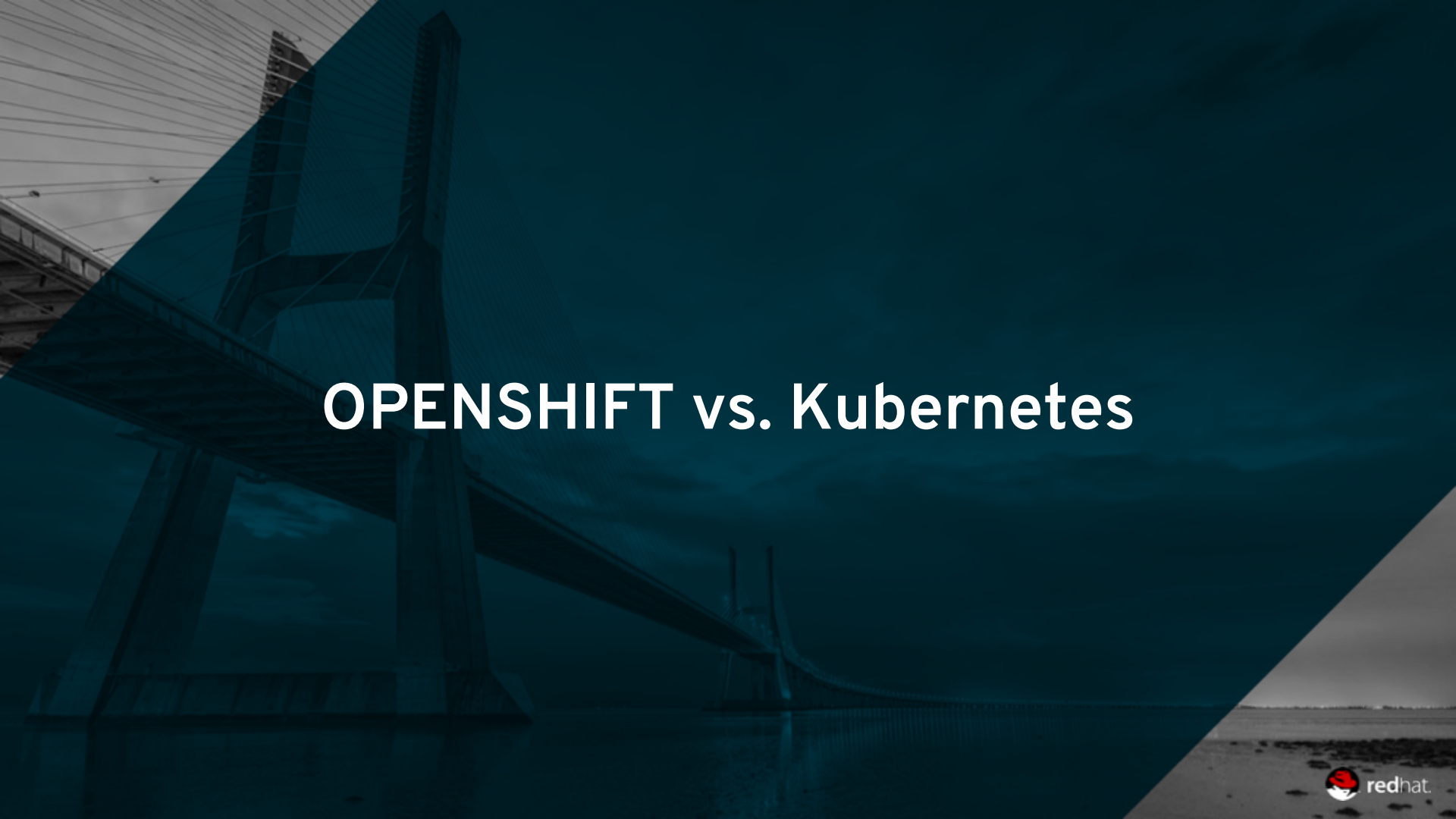
facebook.com/redhatinc

linkedin.com/company/red-hat

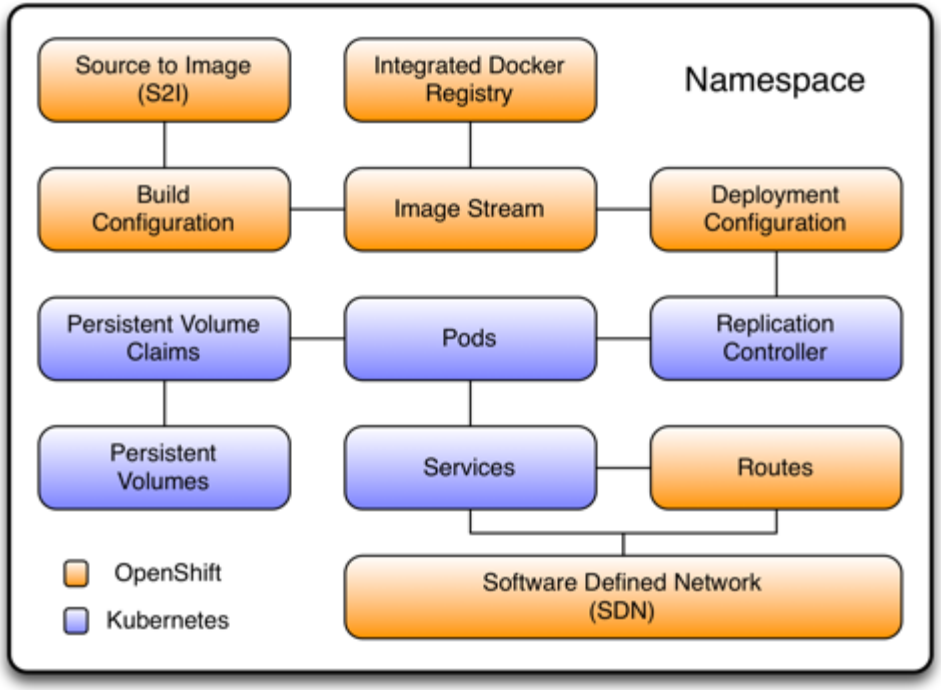
twitter.com/RedHatNews

youtube.com/user/RedHatVideos


APPENDIX
redhat.


OPENSHIFT vs. Kubernetes
redhat.

Component Comparison



Routers

A route exposes a service at a host name, like www.example.com, so that external clients can reach it by name.

Default Router in Openshift is an actual HAProxy container providing reverse proxy capabilities:

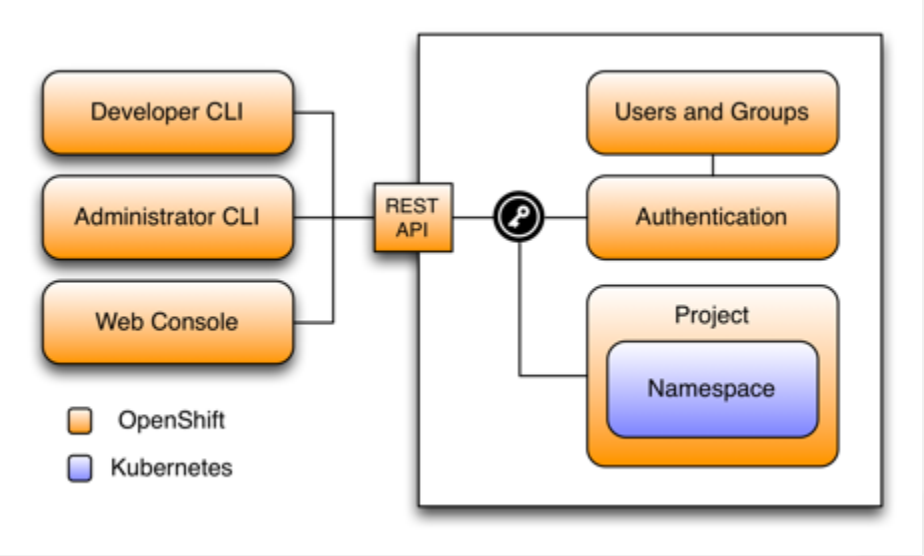
- SSL termination
- Sticky sessions
- Whitelist/blacklist

Projects

Effectively wraps a namespace, with access to the namespace being controlled via the project.

Access is controlled through an authentication and authorization model based on users and groups.

Provide the walls between namespaces, ensuring that users, or applications, can only see and access what they are allowed to.



Builds

Build system provides extensible support for build strategies that are based on selectable types specified in the build API

Docker build

- Invokes the docker build command using a repository with a Dockerfile and all required artifacts in it to produce a runnable image

Source-to-Image (S2I) build

- Framework that makes it easy to write images that take application source code as an input and produce a new image that runs the assembled application as output.

Custom build


OPENSHIFT LAB
redhat.


WHAT NEXT?
redhat.

Builds

Build system provides extensible support for build strategies that are based on selectable types specified in the build API

Docker build

- Invokes the docker build command using a repository with a Dockerfile and all required artifacts in it to produce a runnable image

Source-to-Image (S2I) build

- Framework that makes it easy to write images that take application source code as an input and produce a new image that runs the assembled application as output.

Custom build