

# Openshift Container Platform

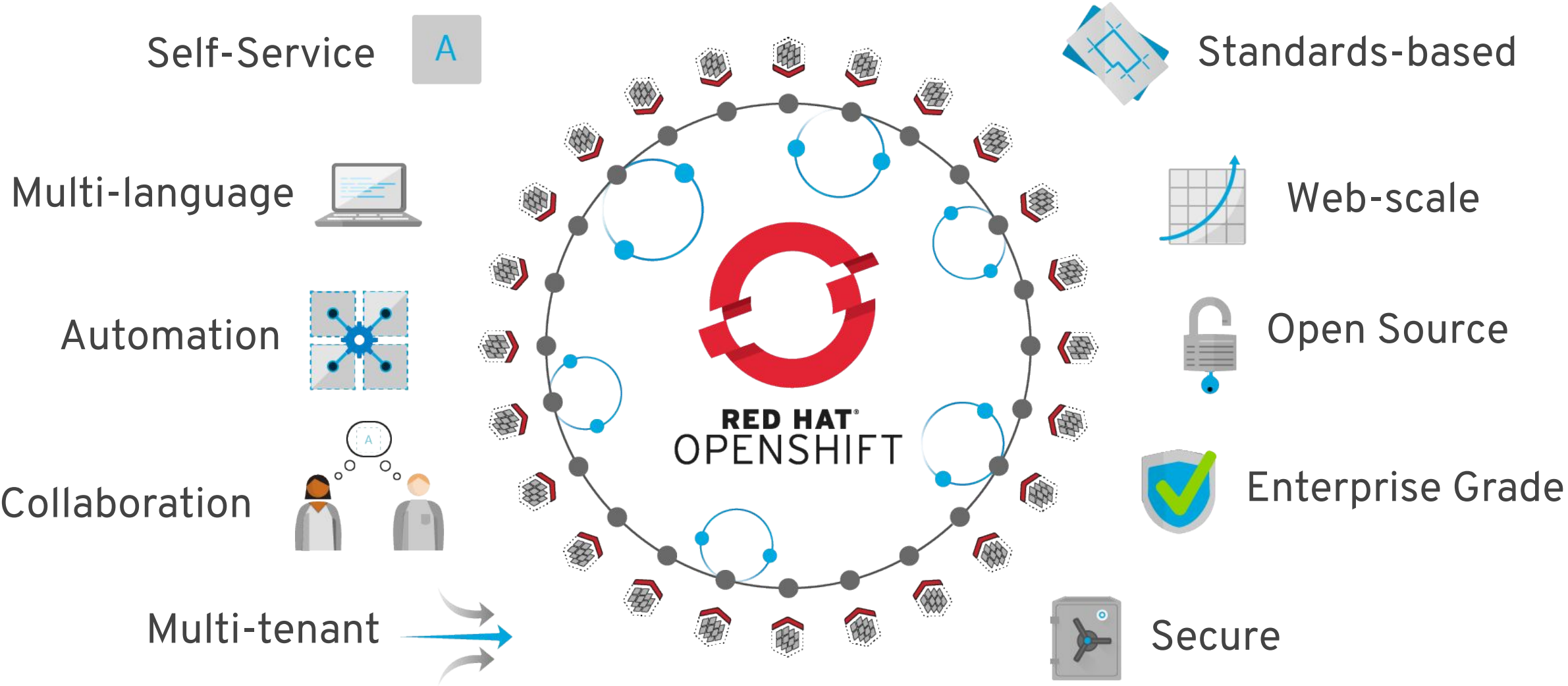
## Workshop OCP Technical Overview

By : Oren Oichman

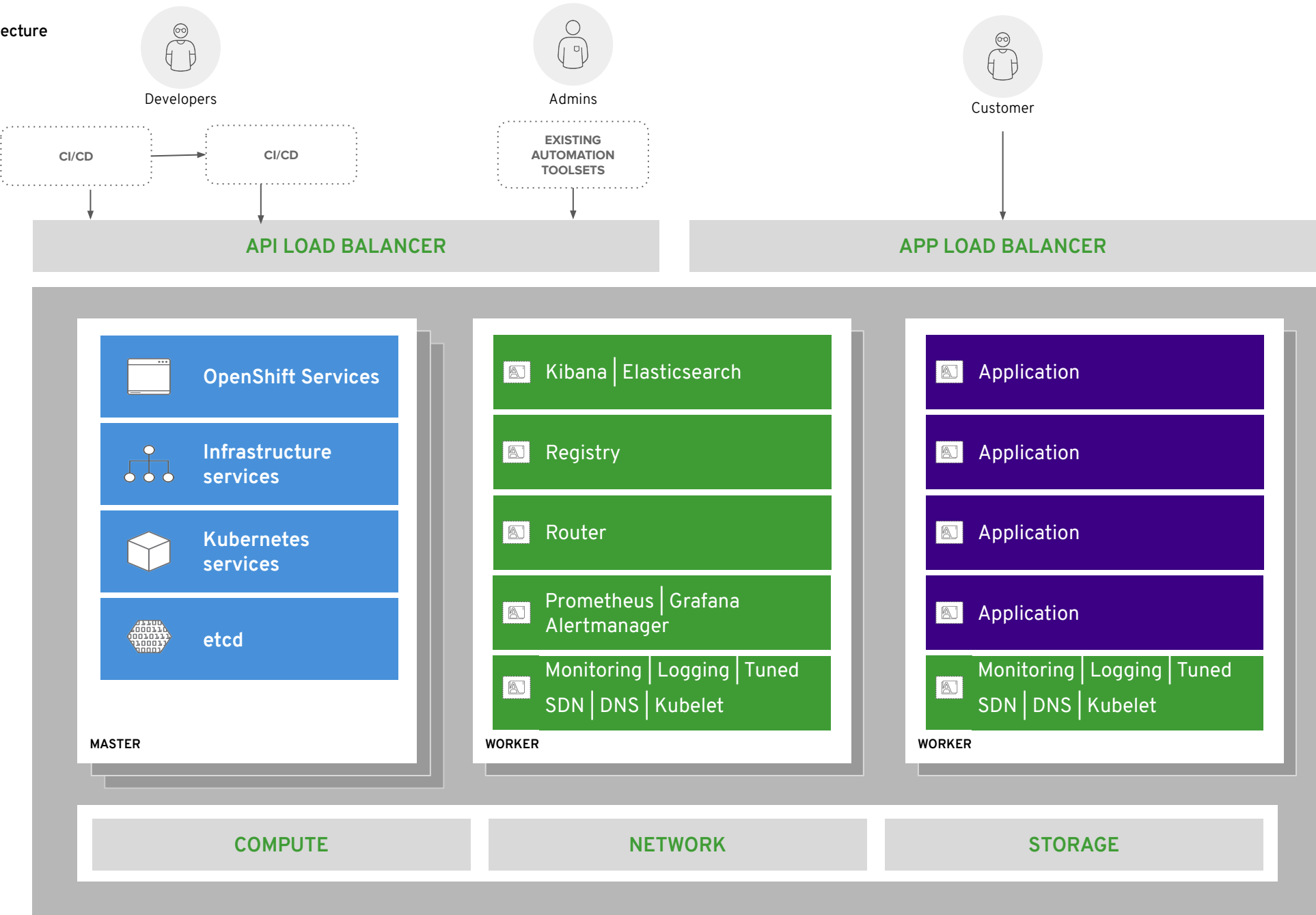
Title: Senior Solution Architect

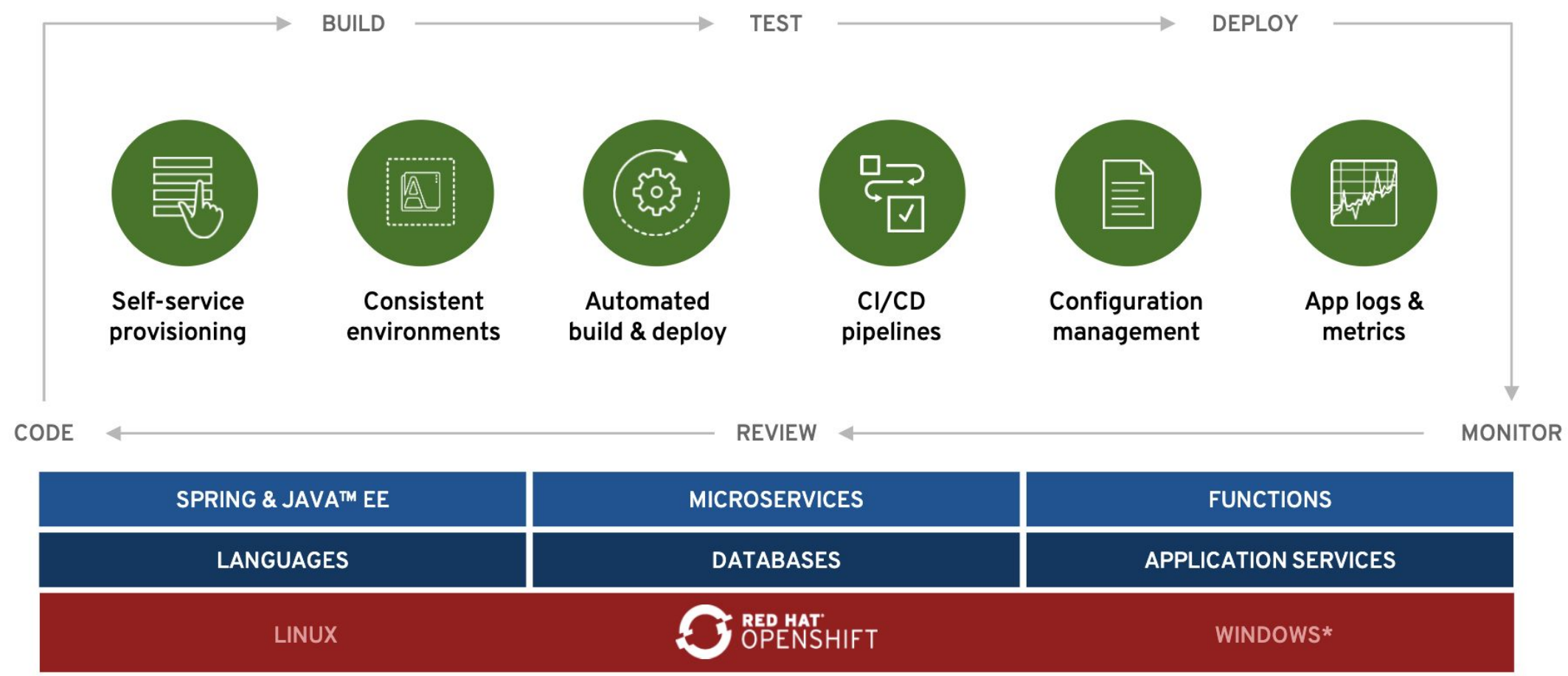
Email : [ooichman@redhat.com](mailto:ooichman@redhat.com)

IRC : [two\\_oes/ooichman](https://www.irccloud.com/app/invite/?invite=two_oes/ooichman)



# OpenShift Architecture





\*Coming soon



# cri-o

Openshift Container engine

supports any OCI compatible runtime.



# Kubelet

Kubernetes primary node agent

makes sure that containers are running in a Pod.

# CRI-O Support in OpenShift

CRI-O tracks and versions identical to Kubernetes, simplifying support permutations

CRI-O 1.18



Kubernetes 1.18



OpenShift 4.5



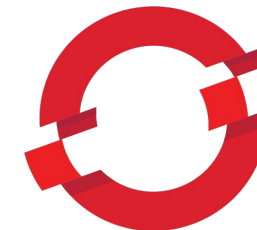
CRI-O 1.19



Kubernetes 1.19



OpenShift 4.6



CRI-O 1.20



Kubernetes 1.20



OpenShift 4.7

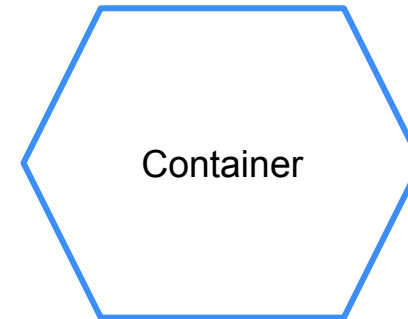


# Basic Concepts



# Containers

A container is the smallest compute unit



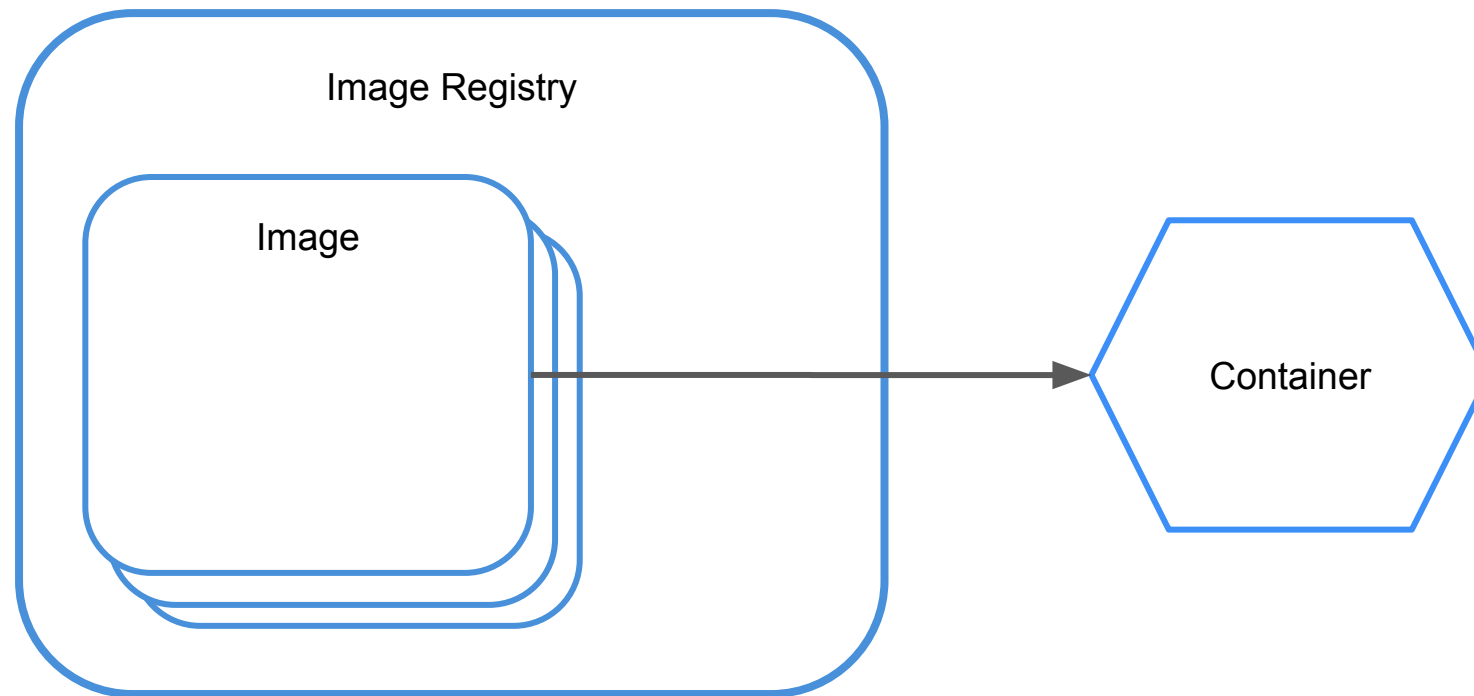
# Containers

Containers are created from container images



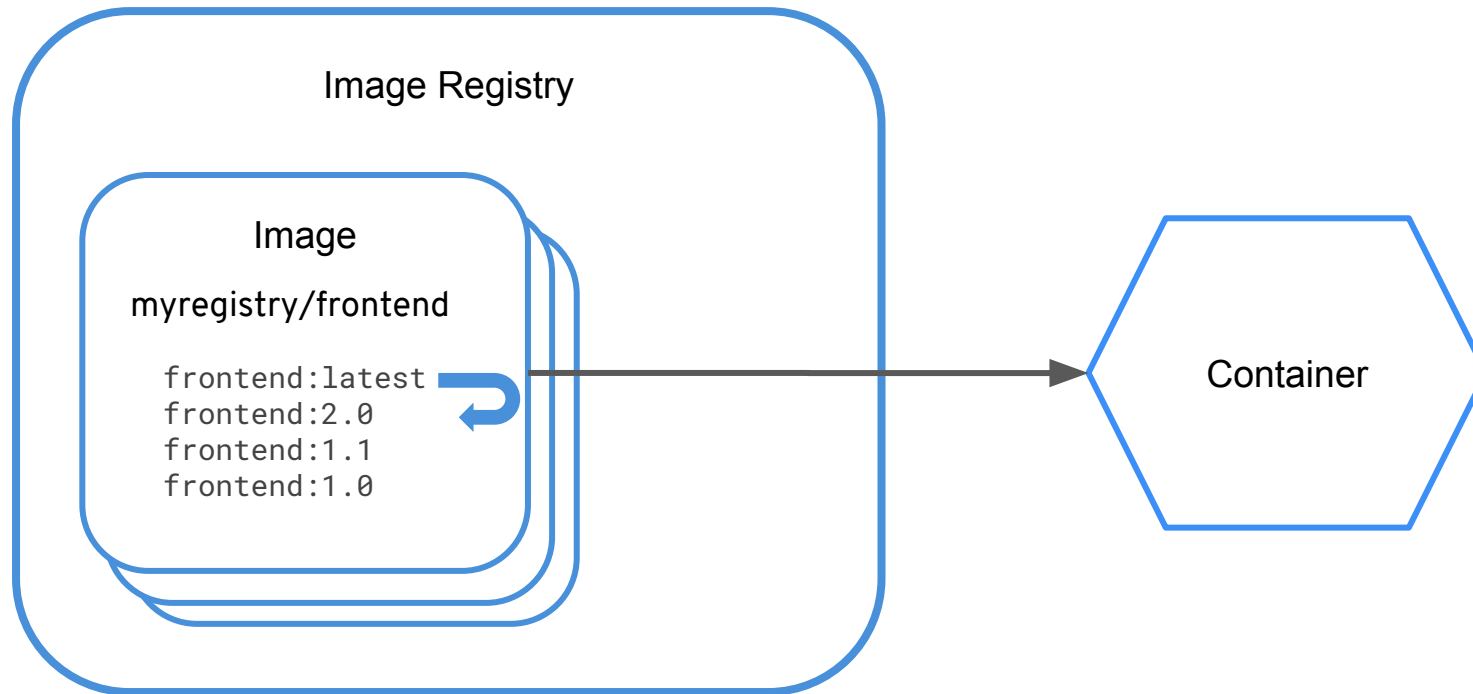
# Containers

Container images are stored in an image registry



# Containers

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

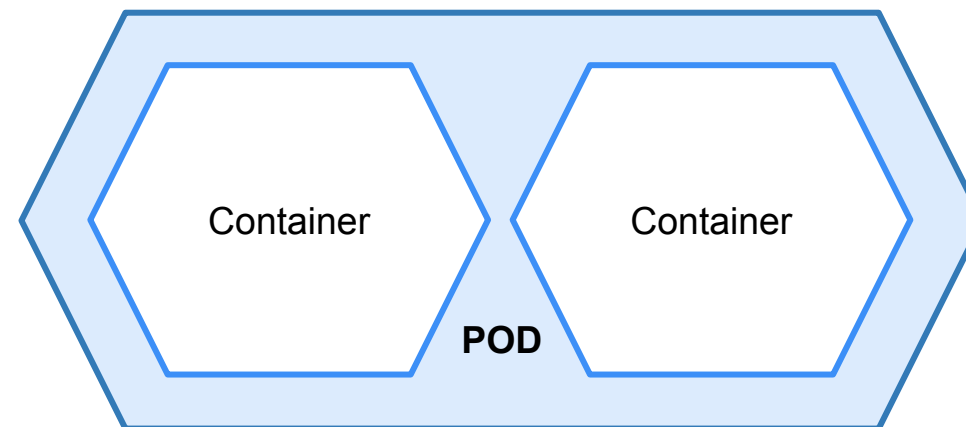


# Pods

The smallest compute unit that can be defined, deployed, and managed.

Rough equivalent of a machine instance (physical or virtual) to a container.

Pods provide CPU, memory, network and storage to a container

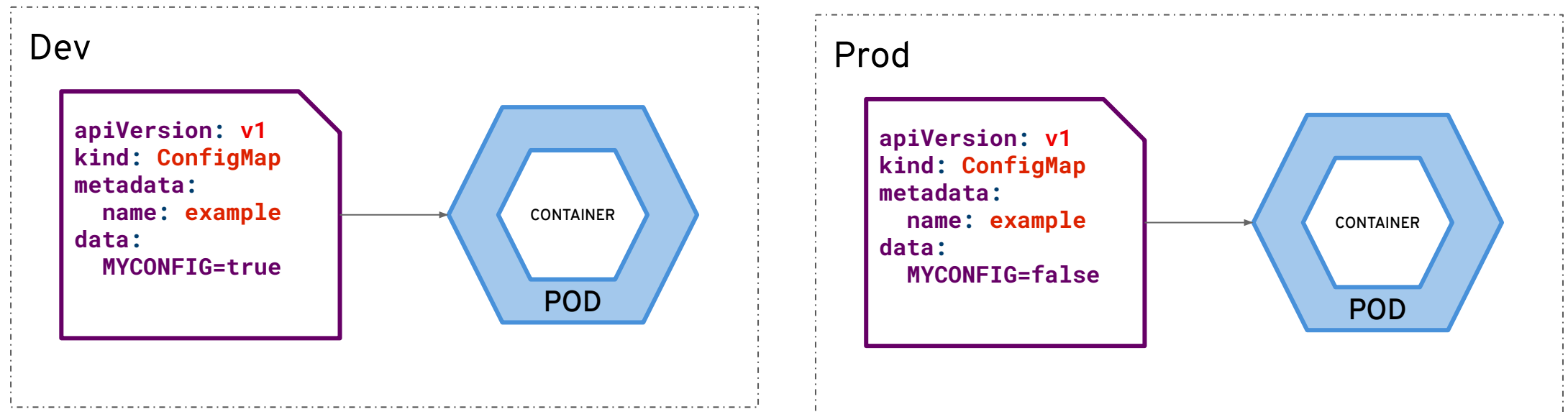


10.140.4.44

```
apiVersion: v1
kind: Pod
metadata:
  name: example
  labels:
    env: test
spec:
  containers:
  - name: containerone
    image: example-image
  - name: containertwo
```

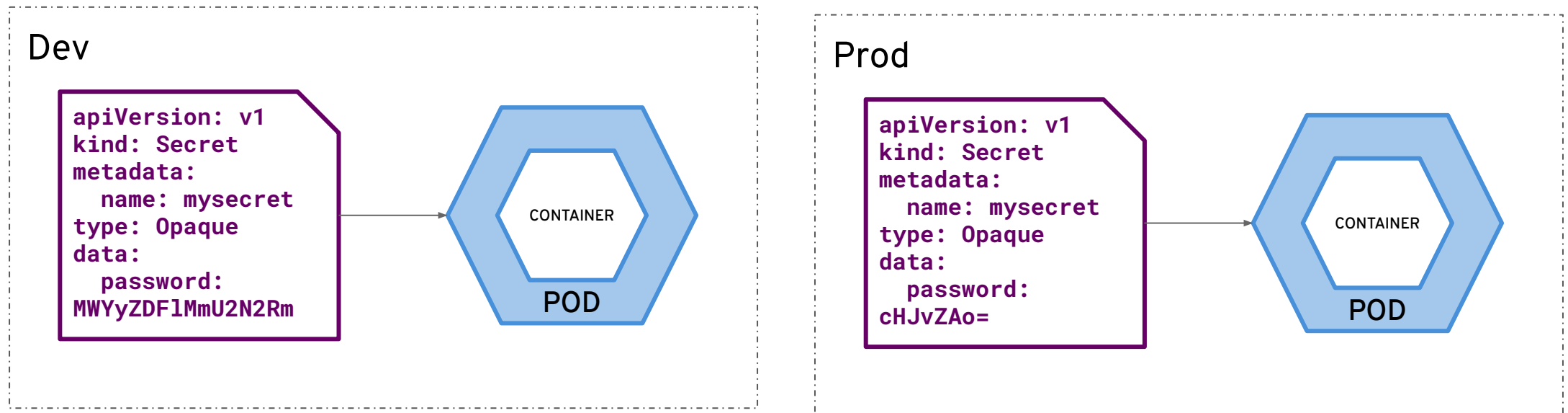
# Configmaps

Allow you to decouple configuration artifacts from image content.



# Secrets

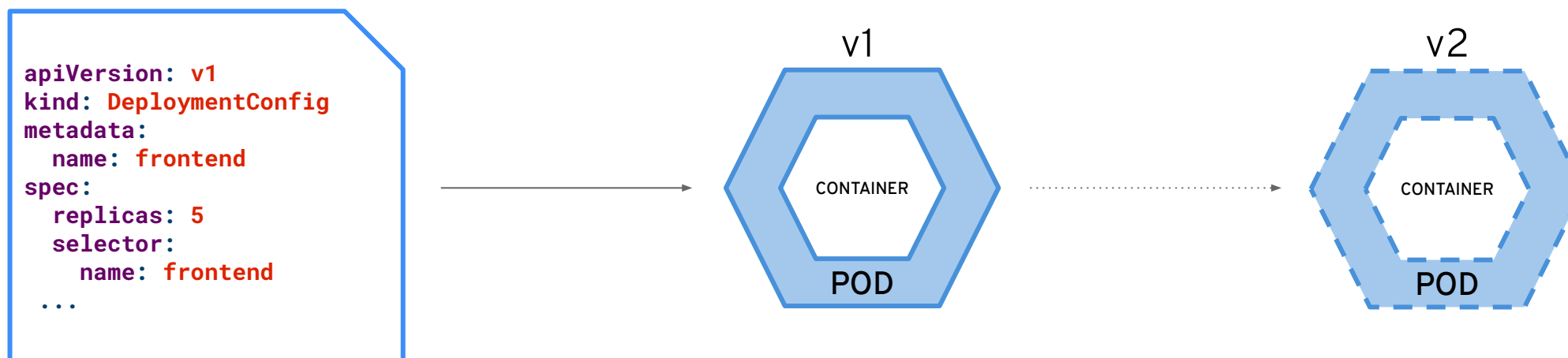
Provide a mechanism to hold sensitive information such as passwords



# Deployments and DeploymentConfigurations

Define how to roll out new versions of Pods

Each time a deployment is triggered, whether manually or automatically, a deployer Pod manages the deployment



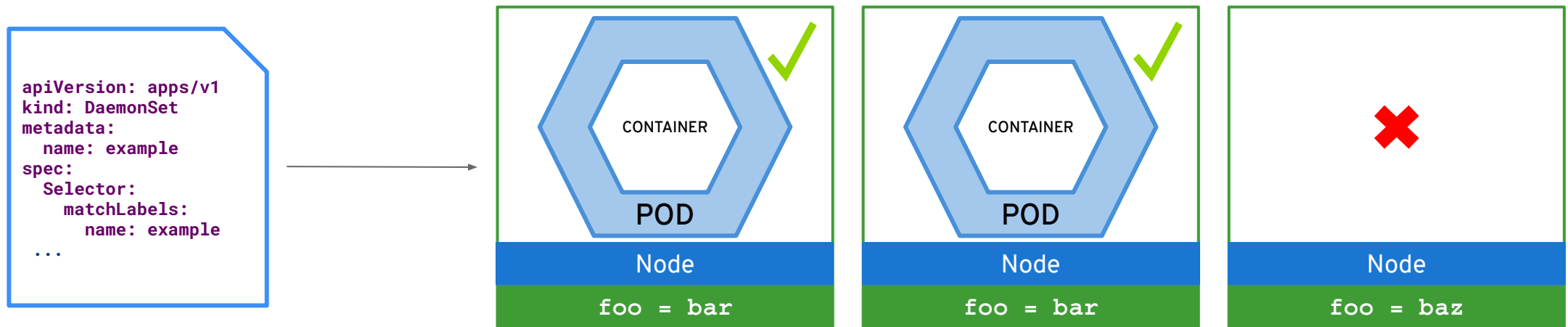


# Daemonset

Ensures that all  
(or some) nodes run a copy of  
a pod

Use cases:

- Cluster storage daemon
- Logs collection daemon (fluentd)
- Node monitoring daemon (node-exporter)
- Network daemon (dns, sdn, ovs)



# Statefulset

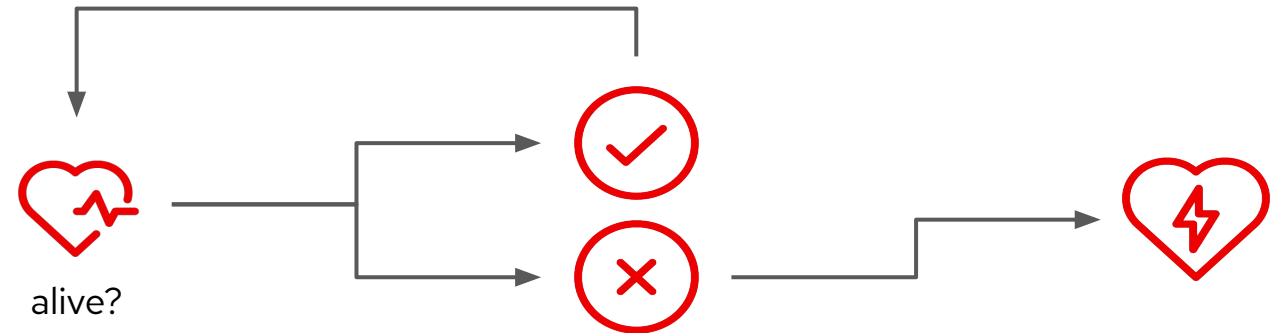
Manages the deployment and scaling of a set of Pods, and provides guarantees about the ordering and uniqueness of these Pods.

Used to manage stateful applications

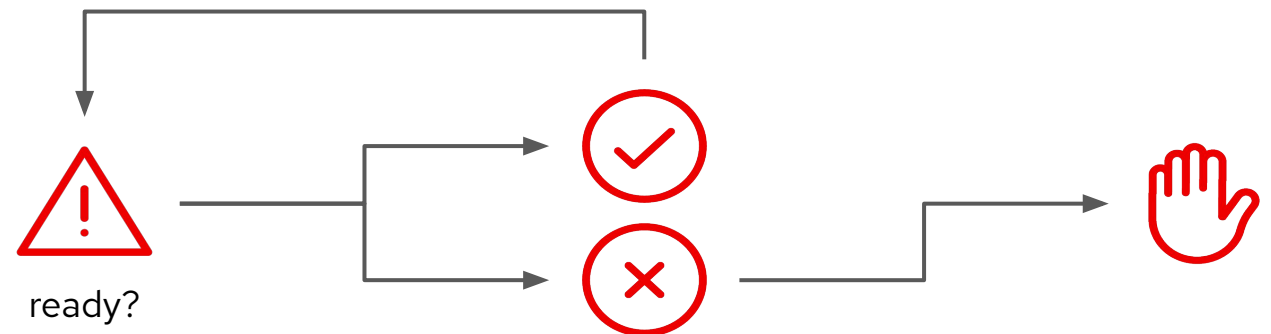
```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  Selector:
    matchLabels:
      app: example
  serviceName: "nginx"
  replicas: 3
  template:
    metadata:
      labels:
        app: example
    Spec:
  terminationGracePeriodSeconds: 10
  Containers: ...
```

# Liveness and Readiness

A Liveness checks determines if the container in which it is scheduled is still running.



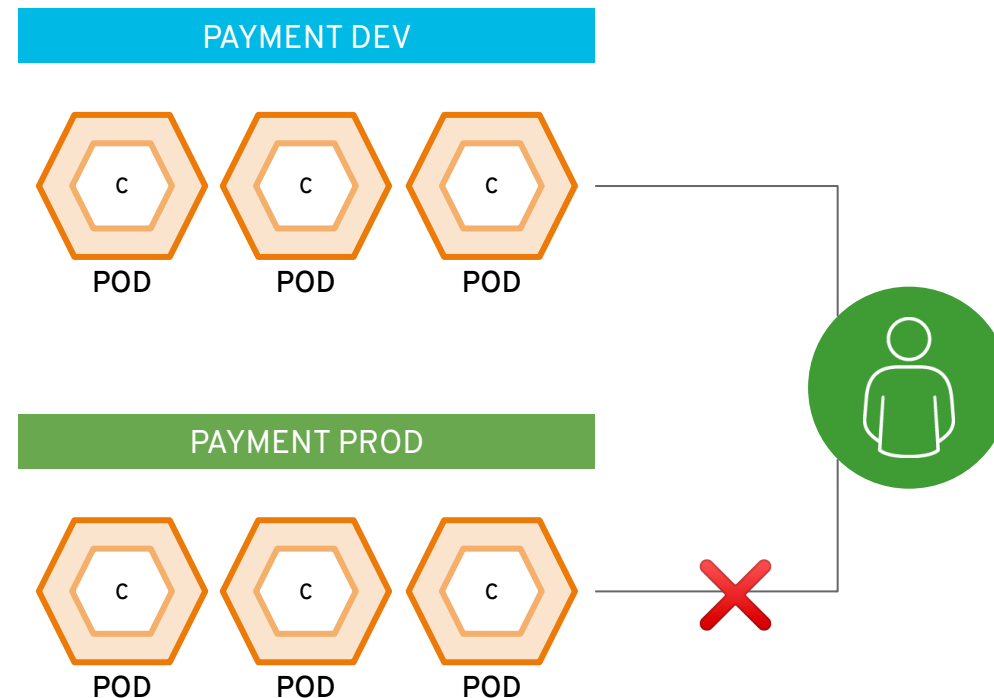
A Readiness check determines if the container in which it is scheduled is ready to service requests.



# Projects

A project allows a community of users to organize and manage their content in isolation from other communities.

Projects starting with openshift- and kube- host cluster components that run as Pods and other infrastructure components.





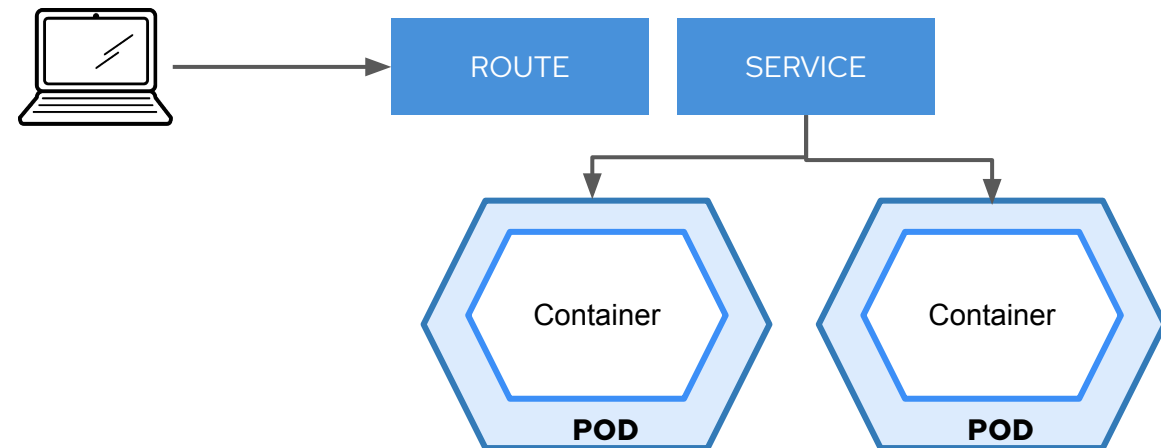
A dedicated Network  
Workshop will deal with  
networking details

# Networking

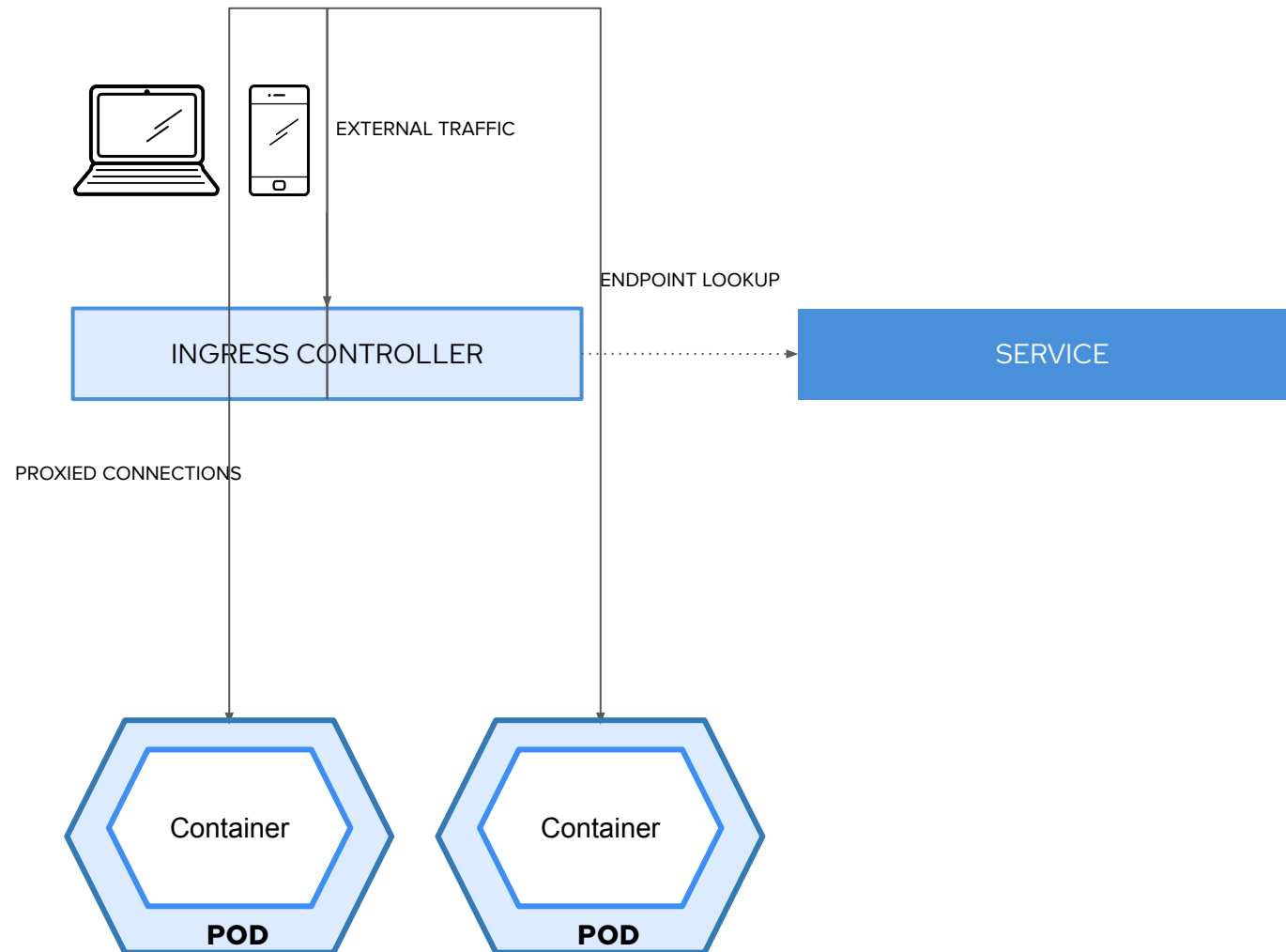
# Openshift Networking

Services provide internal load-balancing and service discovery across pods

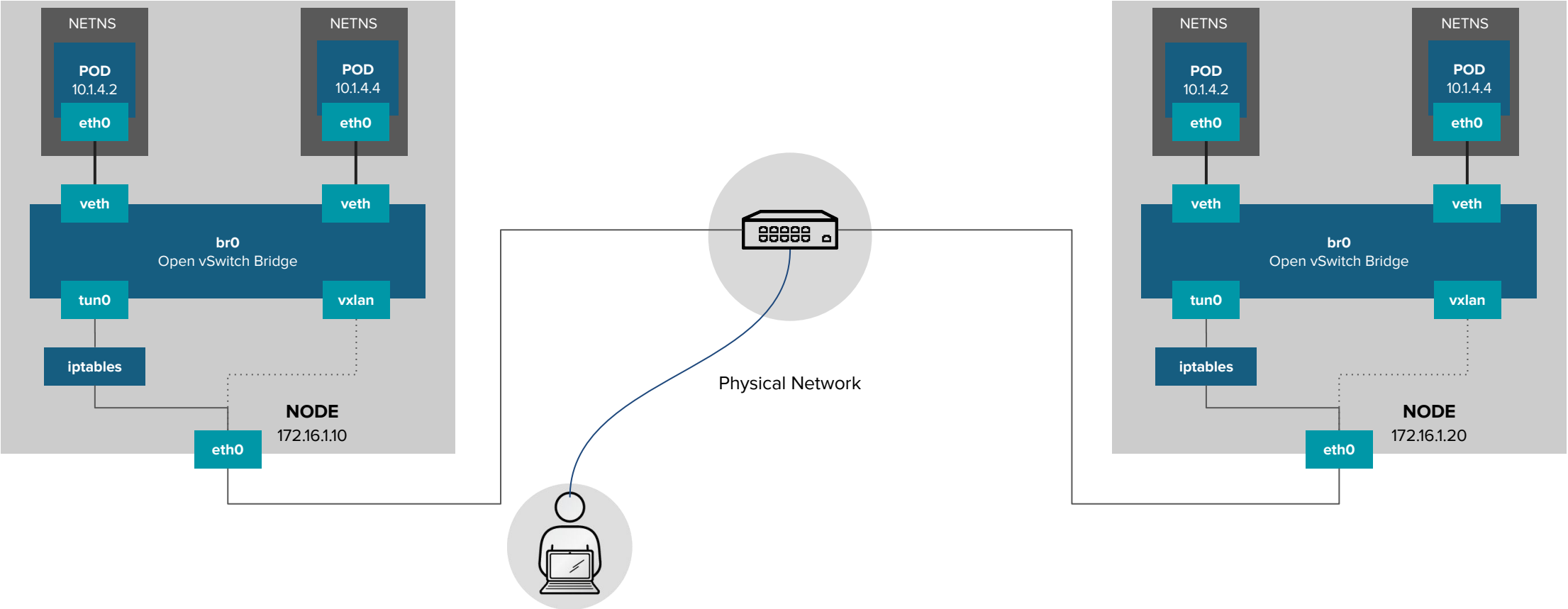
Routes make services accessible to clients outside the environment via real-world urls



# Openshift Networking



# Openshift SDN







# Begin Exercise 3