

An abstract graphic on the left side of the slide, rendered in various shades of red. It features a vertical stack of server racks at the bottom, a cloud with a keyhole icon, a large upward-pointing arrow, and several curved arrows indicating flow or movement. There are also some 'X' and 'O' symbols scattered throughout the design.

Enterprise Kubernetes on AWS

Purpose of the Document


Use this document as an introduction to OpenShift Workshops to provide context to students who are still new to Kubernetes 2 key sections

Upfront introduce Kube Concepts

1. Introduction to Kube concepts (15 mins)
2. Introduction to Kube Architecture (5 mins)

(At halfway mark after break - Position the value of Openshift)

3. Introduction to Openshift (15 mins)



Kubernetes core concepts

a container is the smallest compute unit

The benefits

- Dealing with applications compatibility with the Libraries, Dependencies, OS and hardware across dev and test and prod.
- Containers are smaller, faster start up, Better utilize the OS.
- Lending themselves to be used microservices, distributed, decoupled, scalable applications



- Containers are lightweight mechanisms for **isolating** running processes so that they are limited to interacting with only their designated resources.
- Many application instances can be running in containers on a single host without visibility into each others' processes, files, network, and so on.

containers are created from container images



Note:

A container image is a binary package that encapsulates all of the files necessary to run a program inside of a container

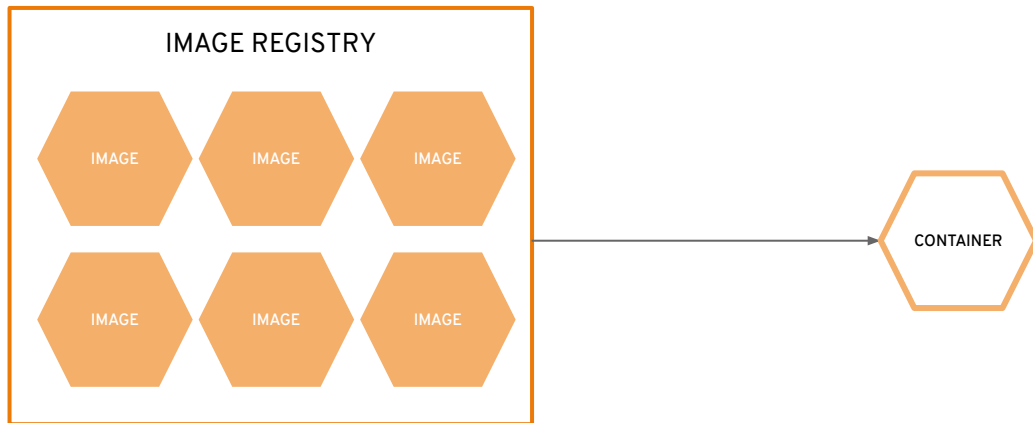
Container Layering

- └ container A: a base operating system only, such as RHEL
- └ container B: build upon #A, by adding Ruby v2.1.10
- └ container C: build upon #A, by adding Golang v1.6

container images are stored in an image registry

An image registry is a service for storing and retrieving container images.

Amazon Elastic Container Registry (ECR)
Red Hat Quay.
Azure Container Registry.
Docker hub.
Alibaba Container Registry.
Harbor.



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

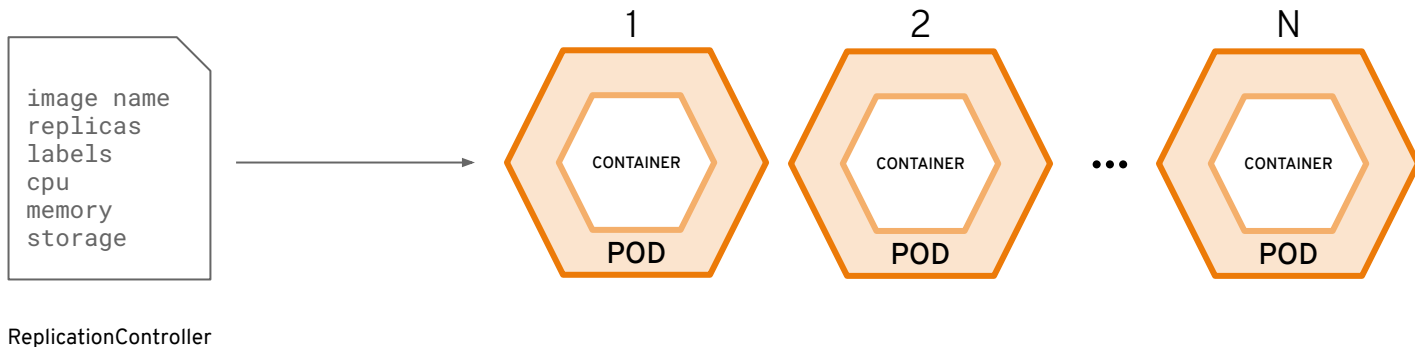


A Pod represents a collection of application containers and volumes running in the same execution environment. Pods, not containers, are the smallest deployable artifact in a Kubernetes cluster. This means all of the containers in a Pod always land on the same machine.

Applications running in the same Pod, share:

- the same IP address
- have the same hostname and can communicate
- Containers in different Pods running on the same node might as well be on different servers

ReplicationControllers ensure a specified number of pods are running at any given time

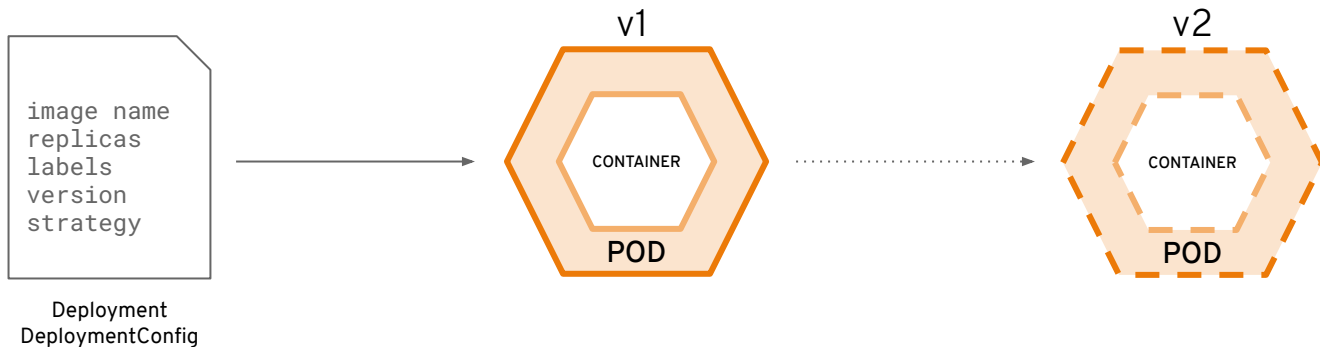


More often than not, you want multiple replicas of a container running at a particular time.

There are a variety of reasons for this type of Replication:

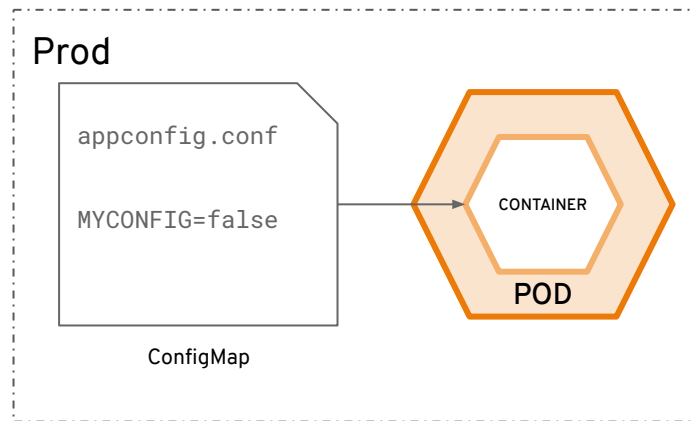
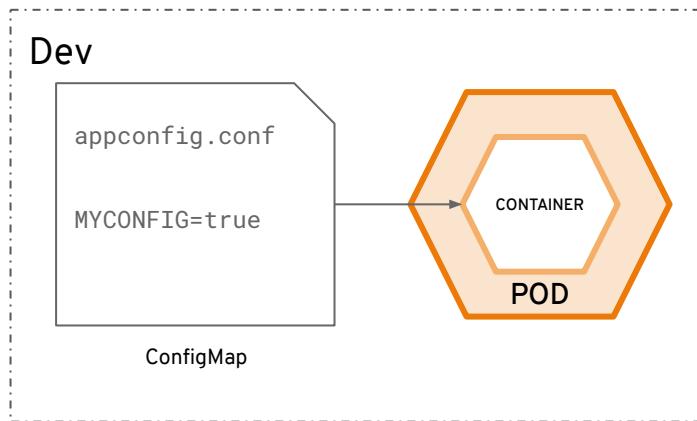
- Redundancy - Multiple running instances mean failure can be tolerated.
- Scale - Multiple running instances mean that more requests can be handled.
- Sharding - Different replicas can handle different parts of a computation in parallel

Deployments and DeploymentConfigurations define how to roll out new versions of Pods



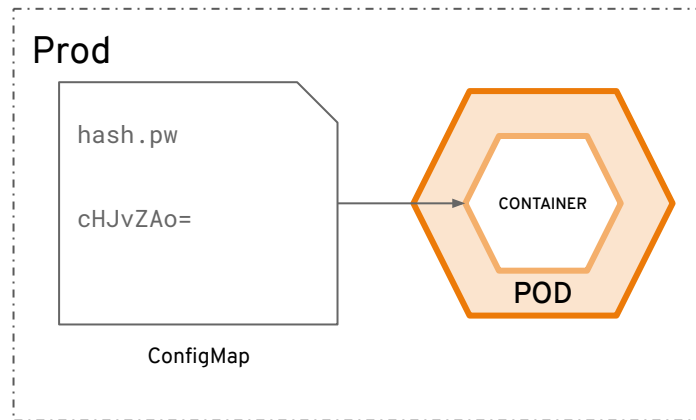
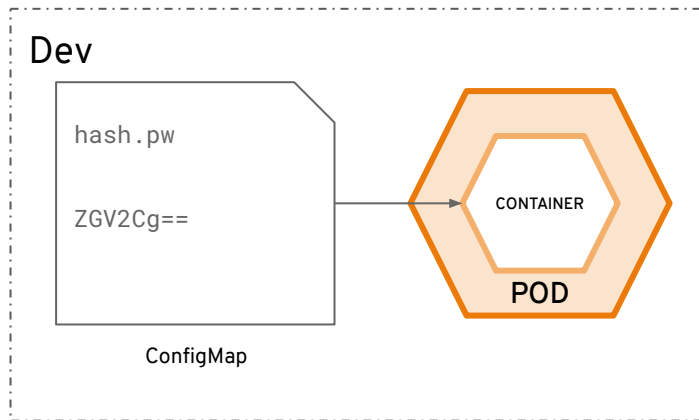
DeploymentConfigs & Deployments describe the desired state of a particular component of an application as a Pod template. Deployments create ReplicaSets, which orchestrate Pod lifecycles.

configmaps allow you to decouple configuration artifacts from image content

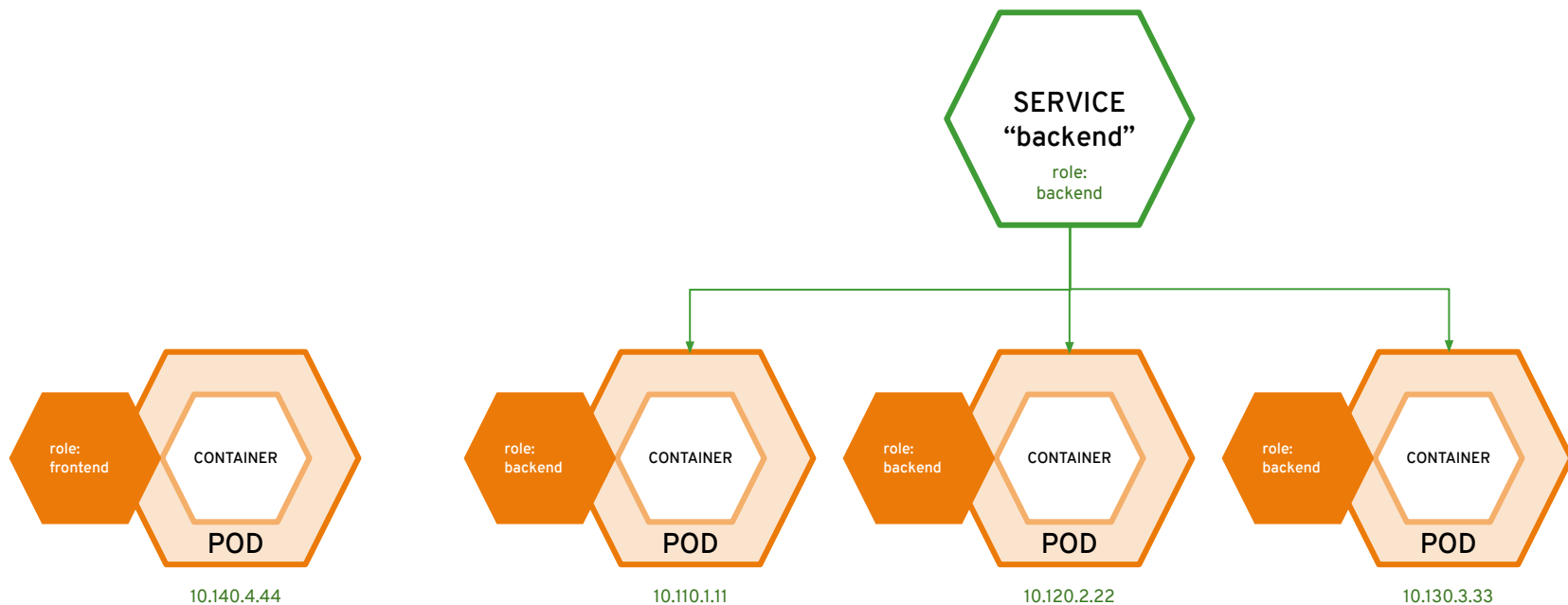


ConfigMaps are a great way to provide dynamic configuration in your application. They allow you to create a container image (and Pod definition) once and reuse it in different contexts. DEV vs PROD
Separating configuration from application code will make your applications more reliable and reusable.

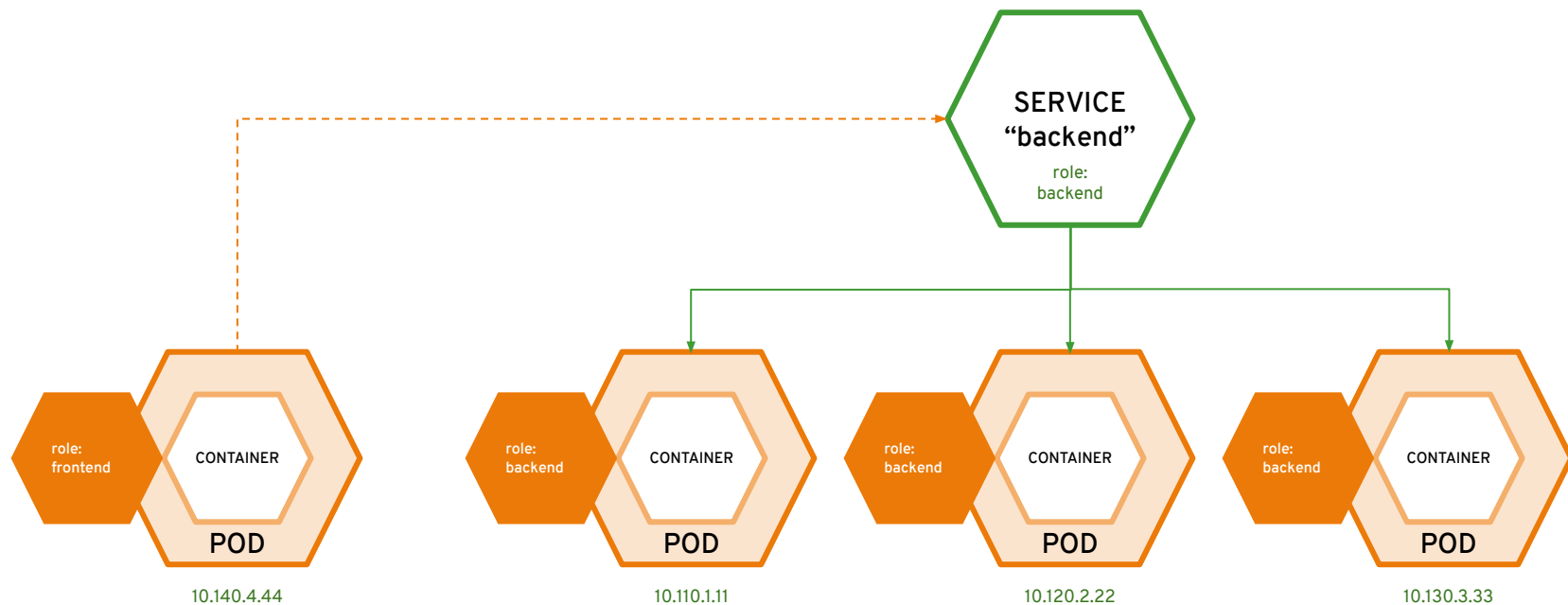
secrets provide a mechanism to hold sensitive information such as passwords



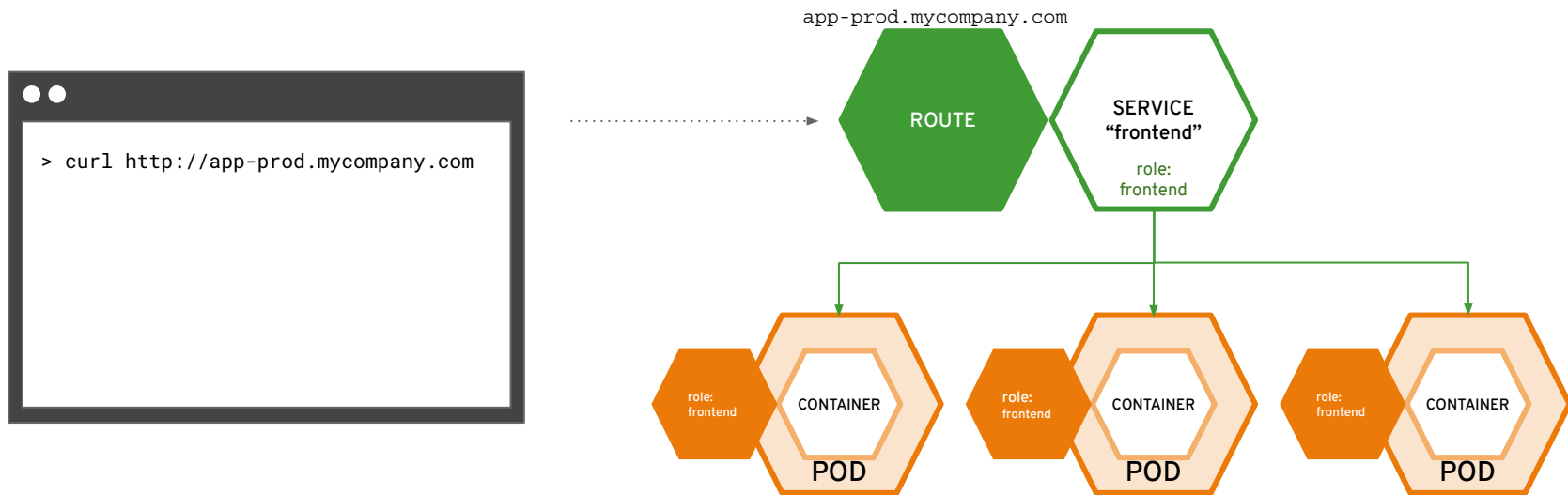
services provide internal load-balancing and service discovery across pods



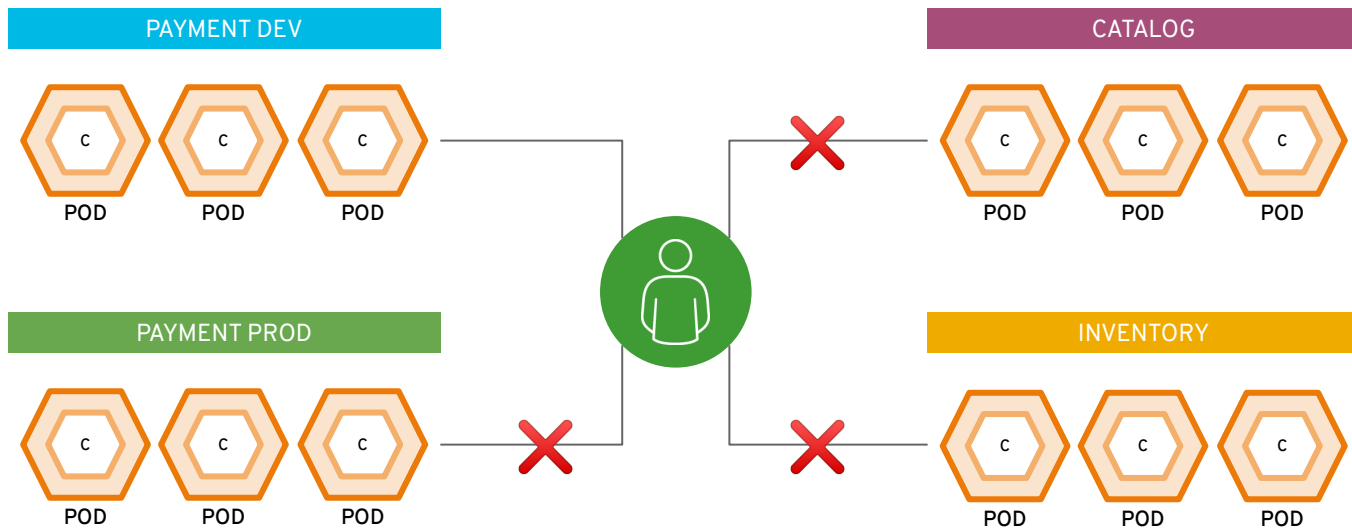
apps can talk to each other via services



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



projects isolate apps across environments, teams, groups and departments





Kube Architecture

starts with infrastructure

COMPUTE

NETWORK

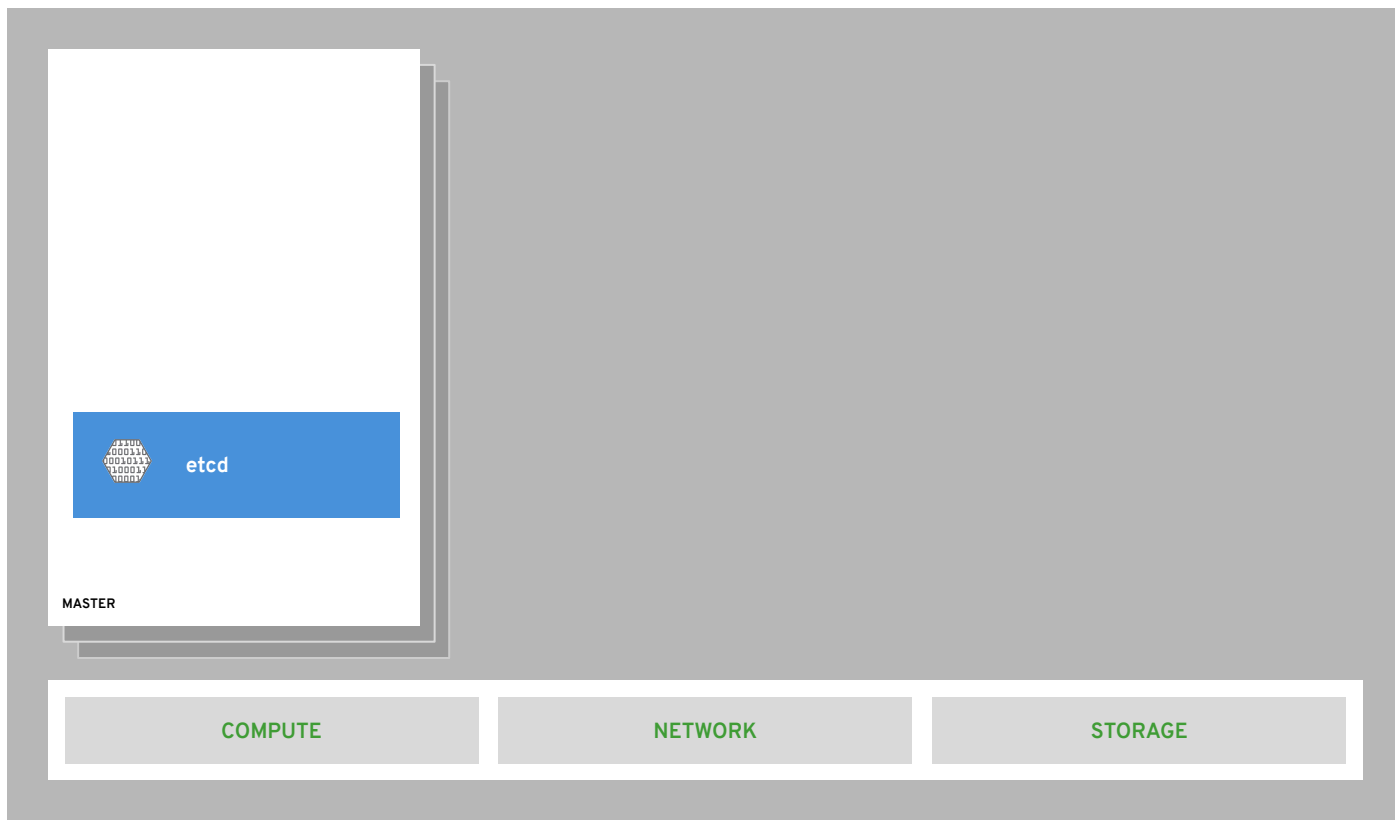
STORAGE

masters are the control plane



state of everything

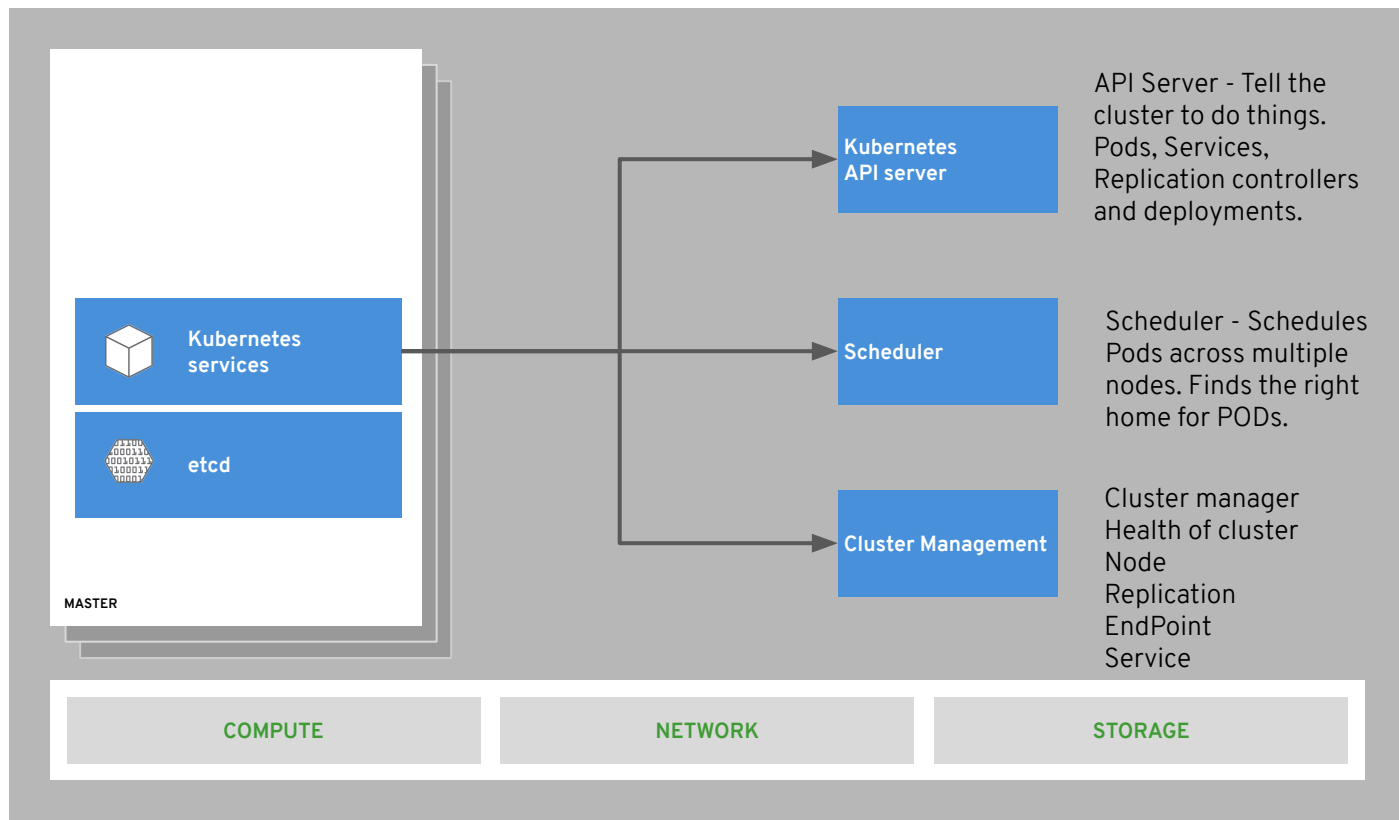
etcd is used to keep track of the state of everything in the cluster, from which users are logged in to where workload lives and more.



workers run workloads



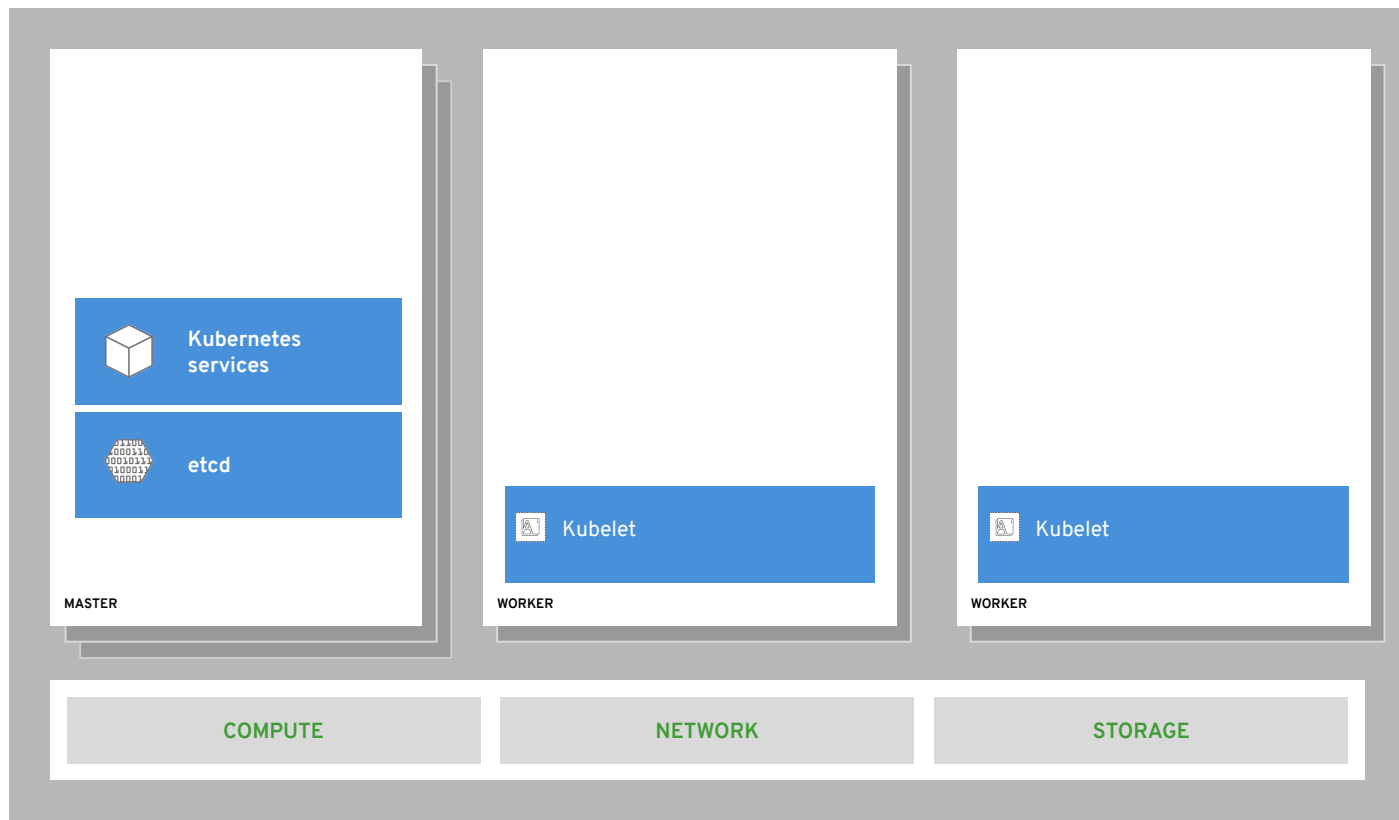
core kubernetes components



run on all hosts

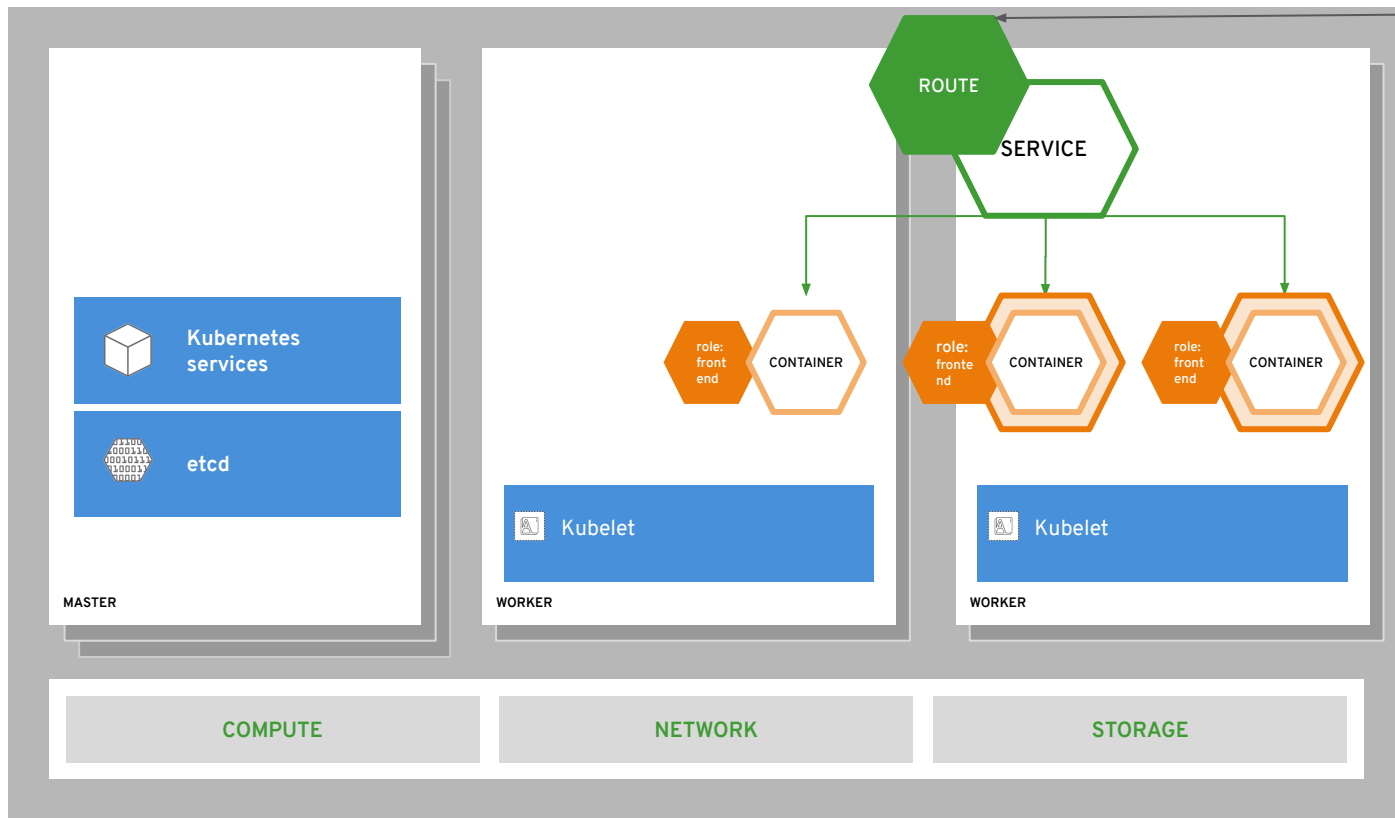
Agent Kubelet, tell me your PodSpec - manages the pods on a node ensuring they are running, healthy and have a good work-life balance.

Also enjoys hold hands with API Server



bring it back to me!

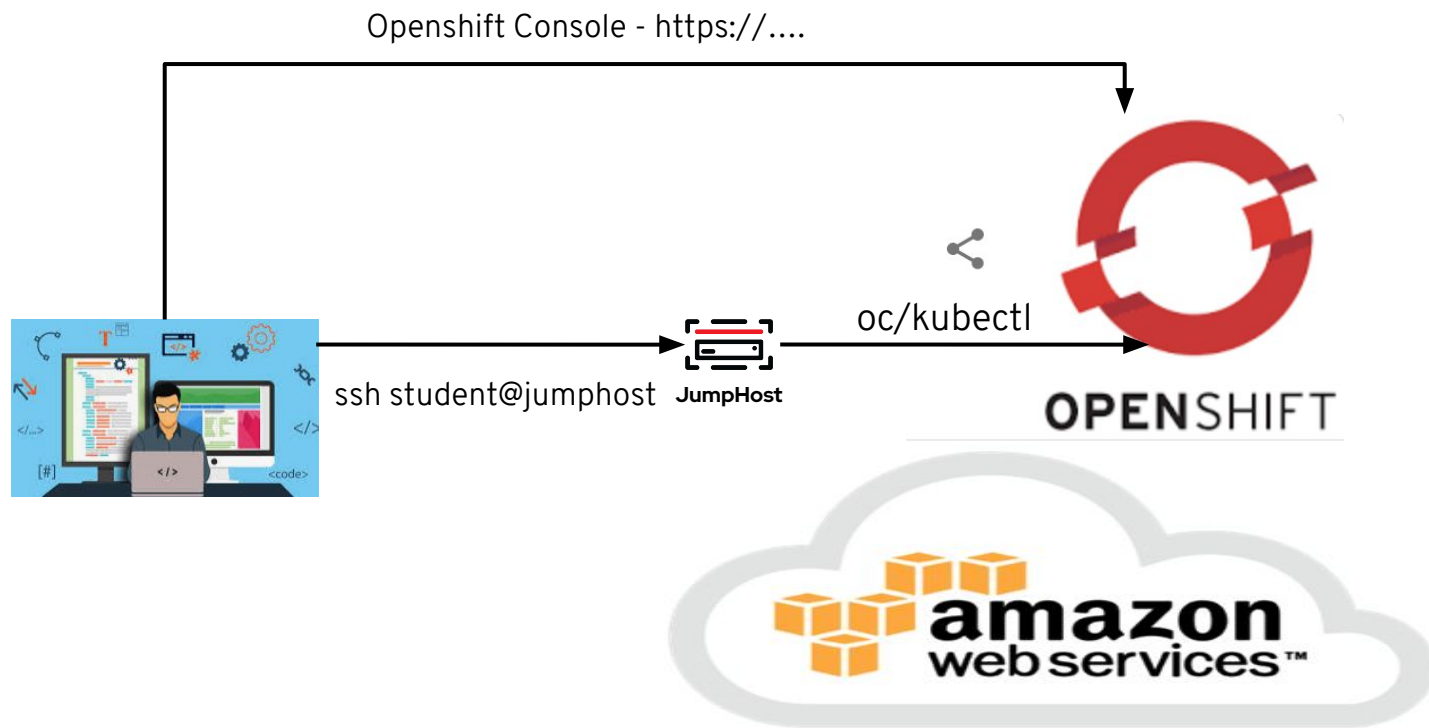
<https://www...>



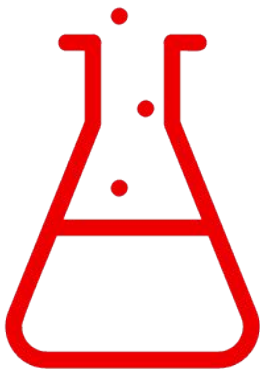


Workshopping

Environment



The Rules



Credentials to access **Lab 1** and **Lab 2** environments:

<https://bit.ly/2PN1O3x>

Retrieve your studentID/UserID

Retrieve how to ssh into the **Jumphost**

Retrieve the WebUI console for accessing the OpenShift cluster

The Connection

SSH Login - you will need an SSH Client

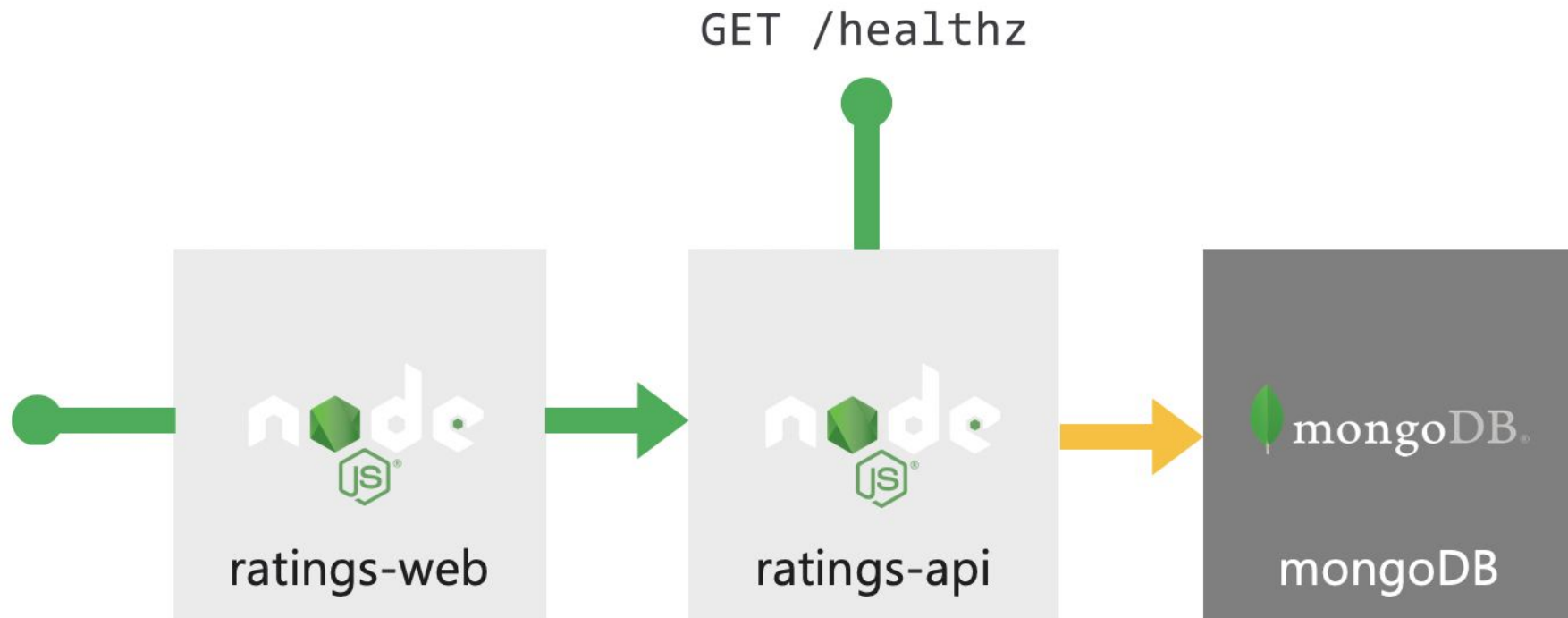
ssh student<add your number>@3.104.30.224

Password: **ansible**

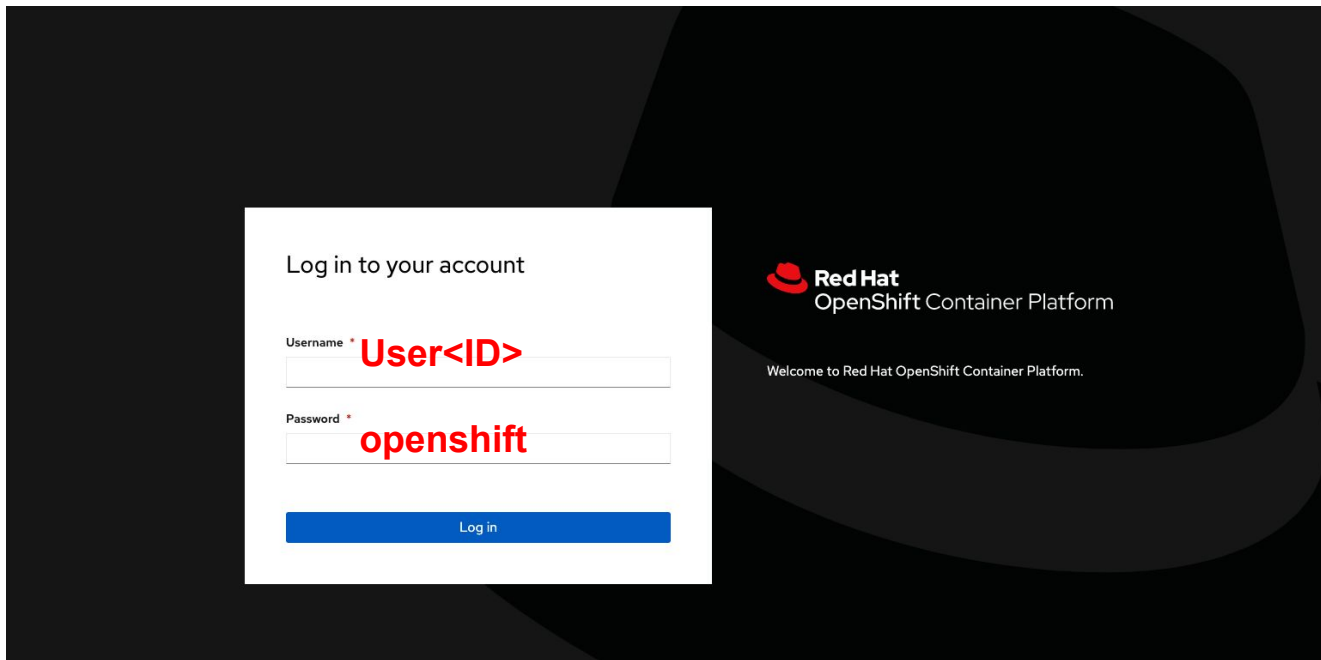
```
simondelord@sdelord-mac VM-Templates $ ssh student01@13.210.109.154
student01@13.210.109.154's password:
This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Mon Oct 12 20:46:01 2020 from 101.180.137.73
[student01@jump ~]$
```

Let's Kick It



WebUI Connection




The image shows the login interface for the Red Hat OpenShift Container Platform. It features a dark background with a large, faint OpenShift logo. On the left, a white login form is centered. The form has the title 'Log in to your account' and two input fields: 'Username' and 'Password'. The 'Username' field contains the text 'User<ID>' and the 'Password' field contains 'openshift'. Below these fields is a blue 'Log in' button. To the right of the form, the Red Hat logo and 'OpenShift Container Platform' text are displayed, followed by the welcome message 'Welcome to Red Hat OpenShift Container Platform.'

Log in to your account

Username * **User<ID>**

Password * **openshift**

Log in

 **Red Hat**
OpenShift Container Platform

Welcome to Red Hat OpenShift Container Platform.

Our First Step

The screenshot shows the Red Hat OpenShift console interface. In the top right, the 'User 1' dropdown menu is open, showing options to 'Copy Login Command' and 'Log out'. A 'Display Token' button is visible to the right. The main content area displays the 'Projects' page with a 'Create Project' button and a search filter. A large overlay window shows the API token and the login command.

Your API token is

5aywEd89sYwKrgVx3Vc1AdfA6dahGmH1jLUdY7aoKU4

Log in with this token

```
oc login --token=5aywEd89sYwKrgVx3Vc1AdfA6dahGmH1jLUdY7aoKU4 --server=https://api.jf7zmswh.southeastasia.aroapp.io:6443
```

Use this token directly as follows:

```
curl -H "Authorization: Bearer 5aywEd89sYwKrgVx3Vc1AdfA6dahGmH1jLUdY7aoKU4" https://api.jf7zmswh.southeastasia.aroapp.io:6443
```

You don't have any projects. You can try to create a new project, by running

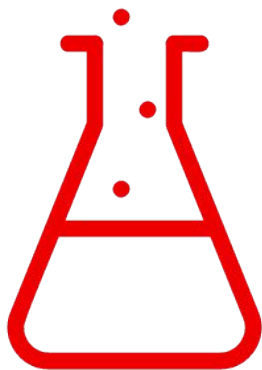
```
oc new-project <projectname>
```

```
[student01@jump ~]$
```

One Last Thing...
Everyone's lives depend on it...
For the next couple hours

```
oc new-project workshop<User#>
```

Lab Instructions

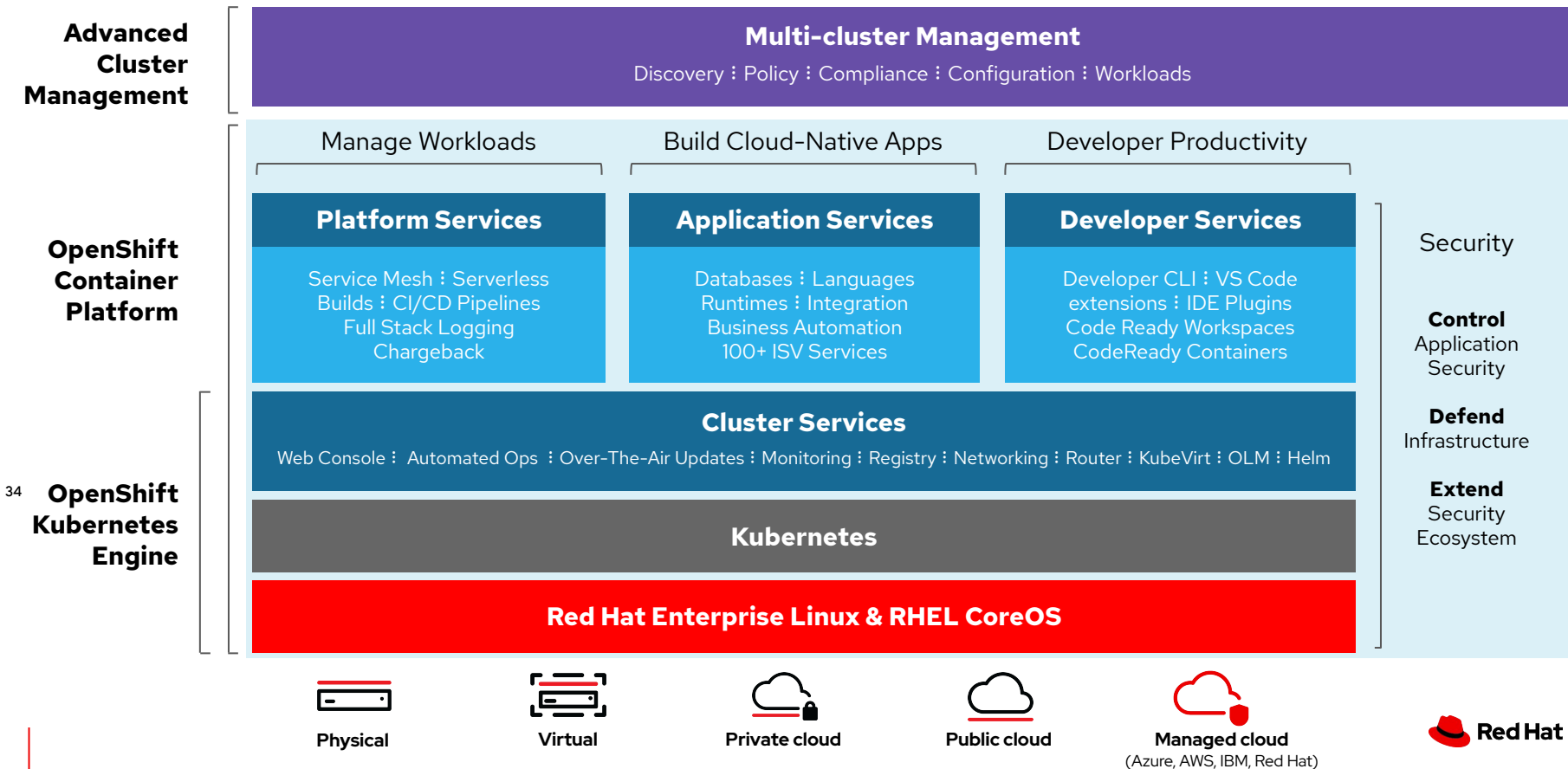


<https://bit.ly/3dbUabu>



What Openshift adds to Kubernetes

OpenShift Overview



Consume It How You Like

Developer Experience

Productivity

Enterprise Readiness

Red Hat OpenShift



Red Hat OpenShift
on Amazon
-or-
OpenShift
Dedicated

Azure Red Hat
OpenShift

OpenShift
Dedicated

Red Hat
OpenShift on IBM
Cloud

OpenShift
Container
Platform

Jointly supported by AWS
and Red Hat

Managed By Red Hat

OCP Customer Managed

Jointly Engineered,
Managed & supported by
Microsoft
and Red Hat


OCP Customer Managed

Managed By Red Hat

OCP Customer Managed

Managed by IBM
Supported by IBM and
Red Hat

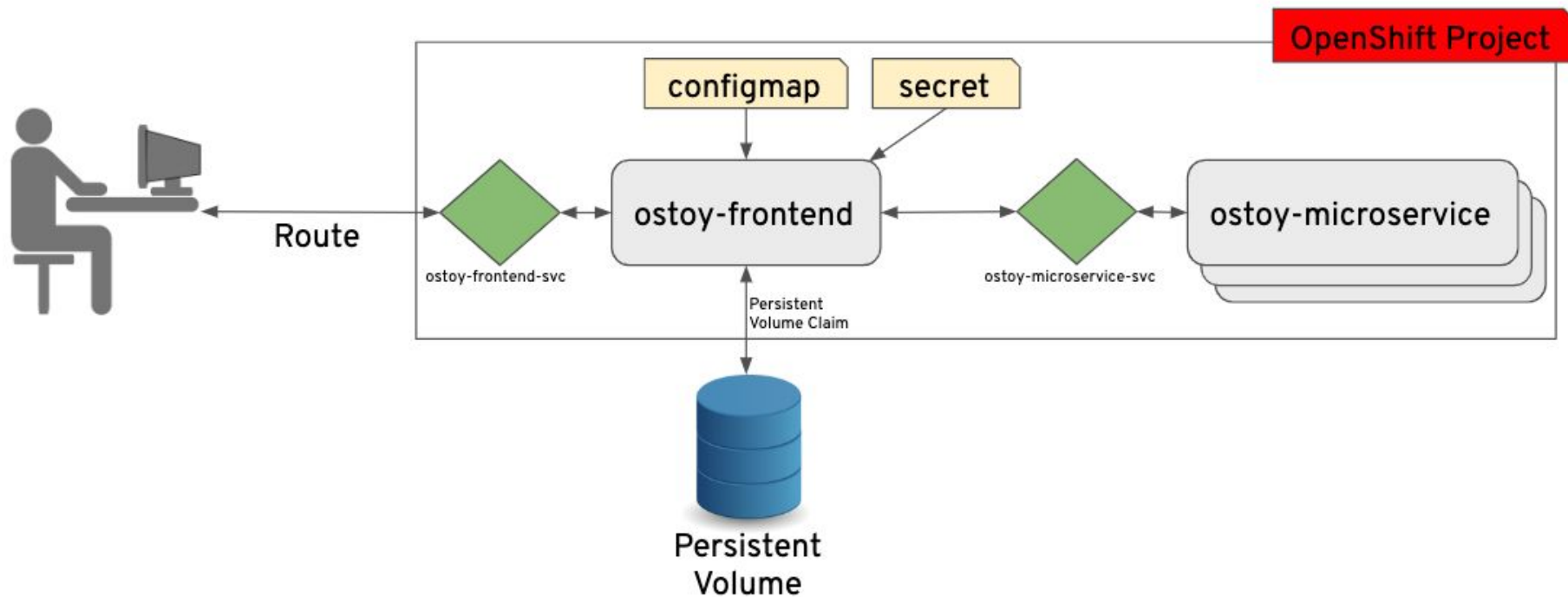
OCP Customer Managed



Lab 2

OSToy app

Let's Break Something



One Last Thing...
Lives are still on the line...
For an hour

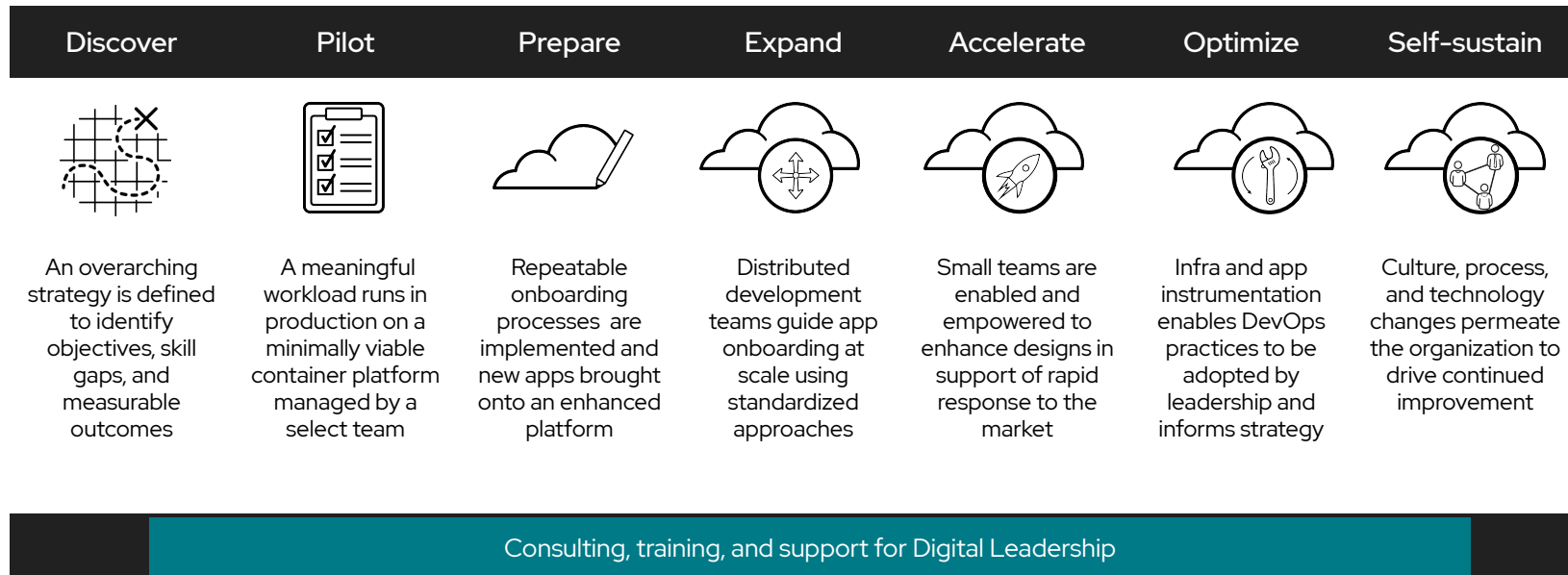
```
oc new-project ostoy<User#>
```



Before you go

Container Adoption Journey Map

Transformation in bite-sized chunks



People - Process - Technology



Your People

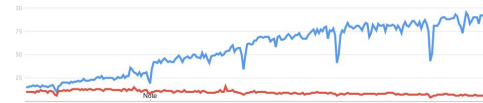
OpenShift Training Programs

[Here](#) is learning path

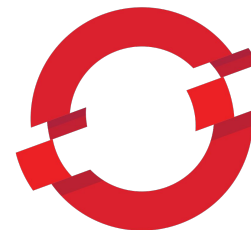


Red Hat Automation
Experts

OpenShift Consulting Services



OpenShift Container Platform



OPENSHIFT


Before you go


Please update you etherpad

1. What did you like:
2. What can we improve:
3. Let us know if you have any projects in mind and we can set you up with out specialist team for a a follow-up.

Thank you



 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

Red Hat is here to help

Responding to COVID-19 requires collaboration, transparency, and the free exchange of expertise.

Ways to contact us