# Launch Single Node Kubernetes Cluster

**CKA Syllabus:**

**Installation, Configuration and Validation**

Goal:

Design a Kubernetes cluster

Step 1:

Start Minikube

```
Terminal    Dashboard    +
Your Interactive Bash Terminal.

$ minikube version
minikube version: v1.8.1
commit: cbda04cf6bbe65e987ae52bb393c10099ab62014
$ minikube start --wait=false
* minikube v1.8.1 on Ubuntu 18.04
* Using the none driver based on user configuration
* Running on localhost (CPUs=2, Memory=2460MB, Disk=145651MB) ...
* OS release is Ubuntu 18.04.4 LTS
* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Launching Kubernetes ...
* Enabling addons: default-storageclass, storage-provisioner
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
```

Step 2:

Cluster info

```
$ kubectl cluster-info
Kubernetes master is running at https://172.17.0.22:8443
KubeDNS is running at https://172.17.0.22:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/p
roxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$ kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
minikube   Ready    master   34s   v1.17.3
$ kubectl create deployment first-deployment --image=katacoda/docker-http-server
deployment.apps/first-deployment created
```

Step 3:

Deploy Containers

```
$ kubectl create deployment first-deployment --image=katacoda/docker-http-server
deployment.apps/first-deployment created
```

The command below finds the allocated port and executes a HTTP request.

```
$ export PORT=$(kubectl get svc first-deployment -o go-template='{{range.spec.ports}}{{if .nodePort}
}{{.nodePort}}{{"\n"}}{{end}}{{end}}')
$ echo "Accessing host01:$PORT"
Accessing host01:31817
$ curl host01:$PORT
<h1>This request was processed by host: first-deployment-666c48b44-l8lrs</h1>
```

Step 4: Dashboard

Overview

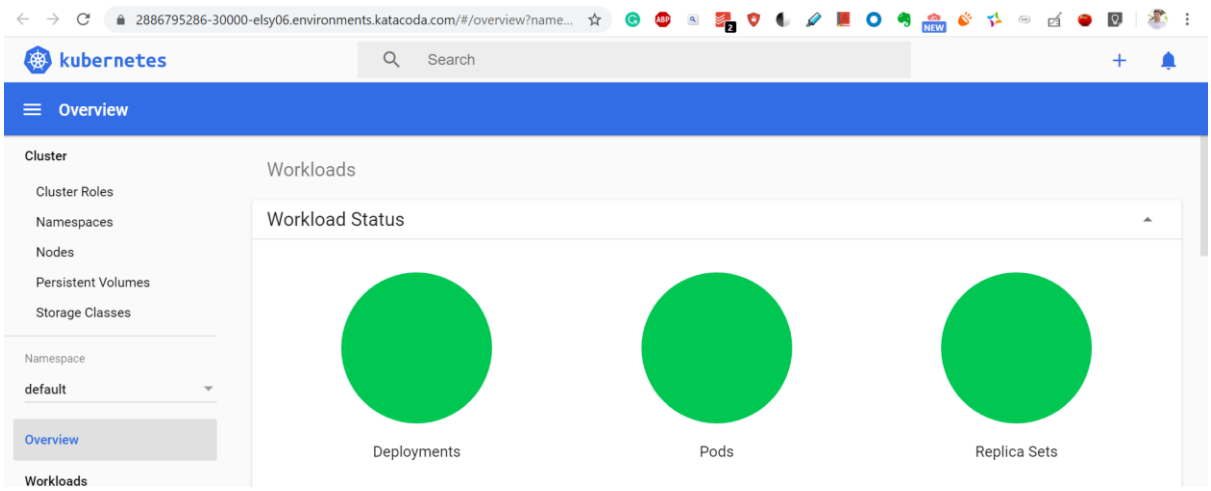


Fig: 1

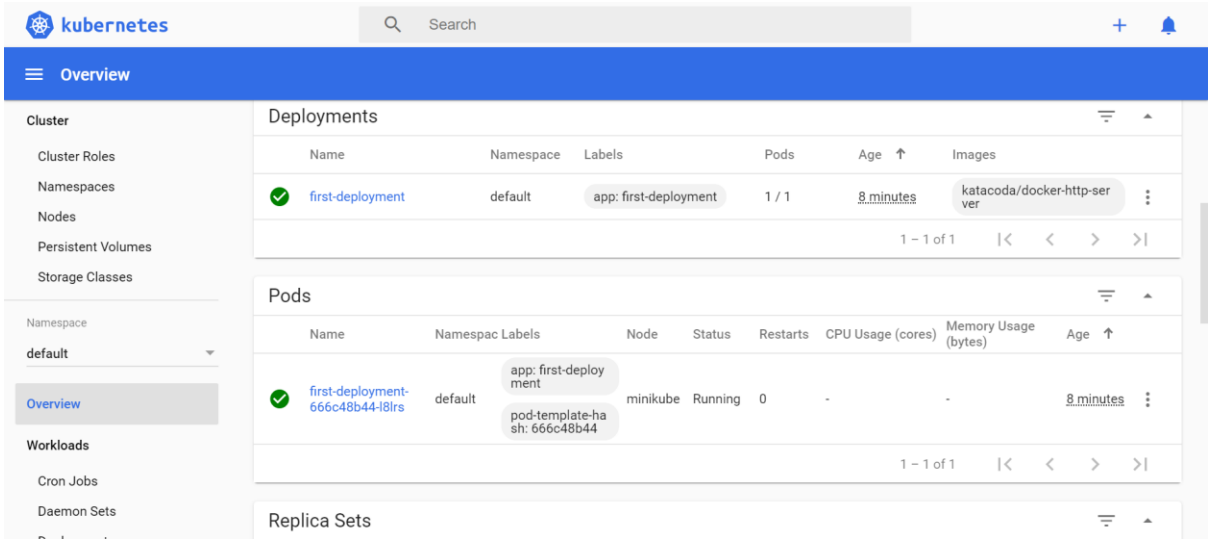

Fig: 2

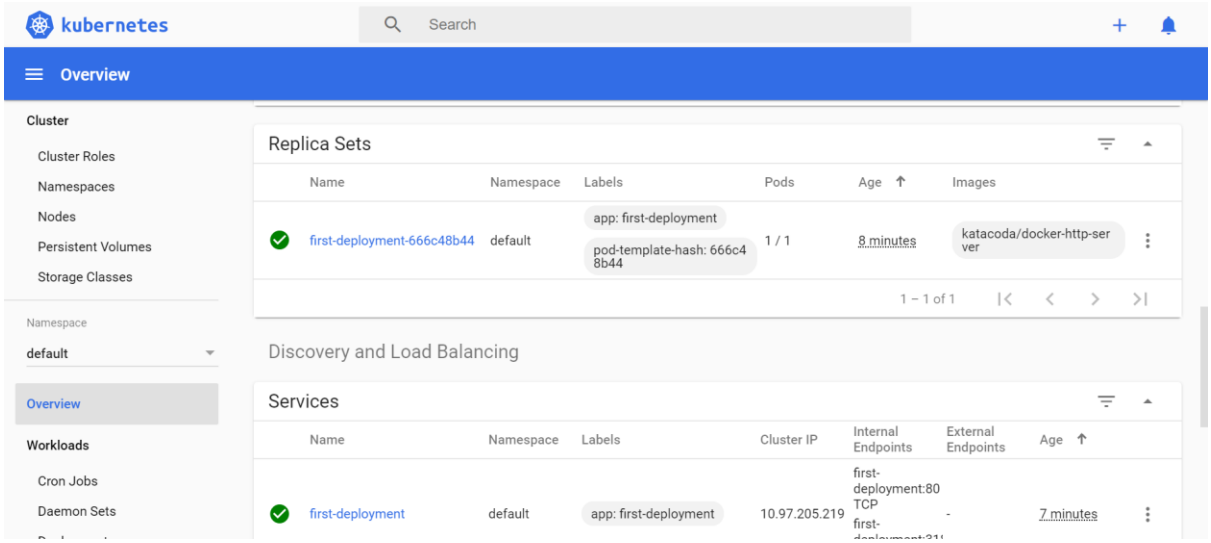


Fig: 3

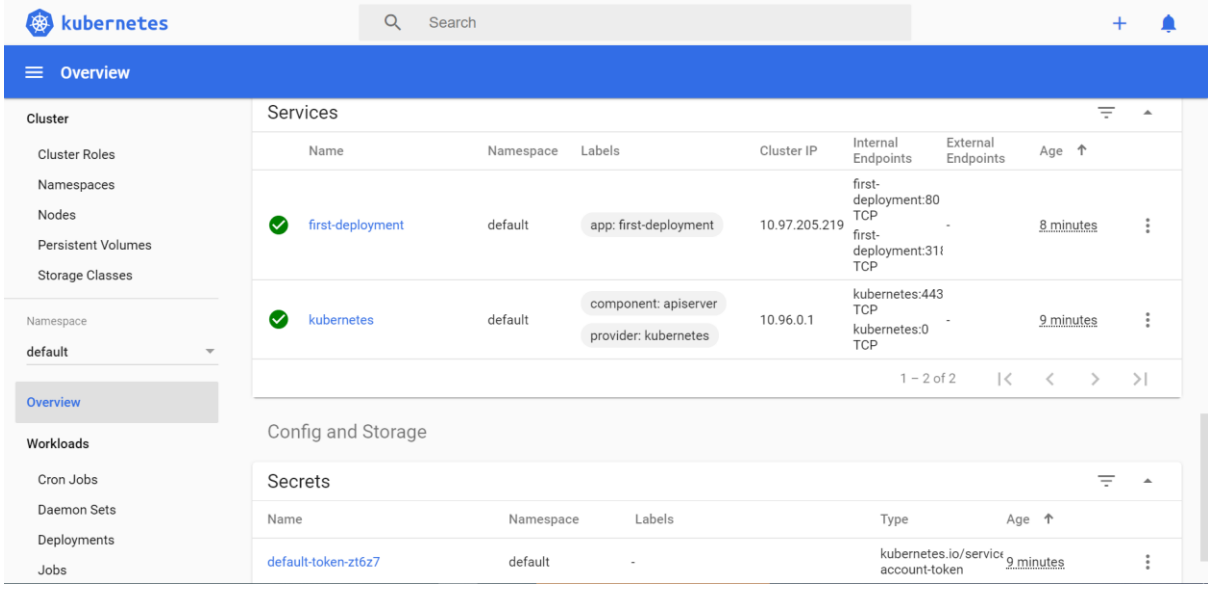


Fig: 4

Pods:

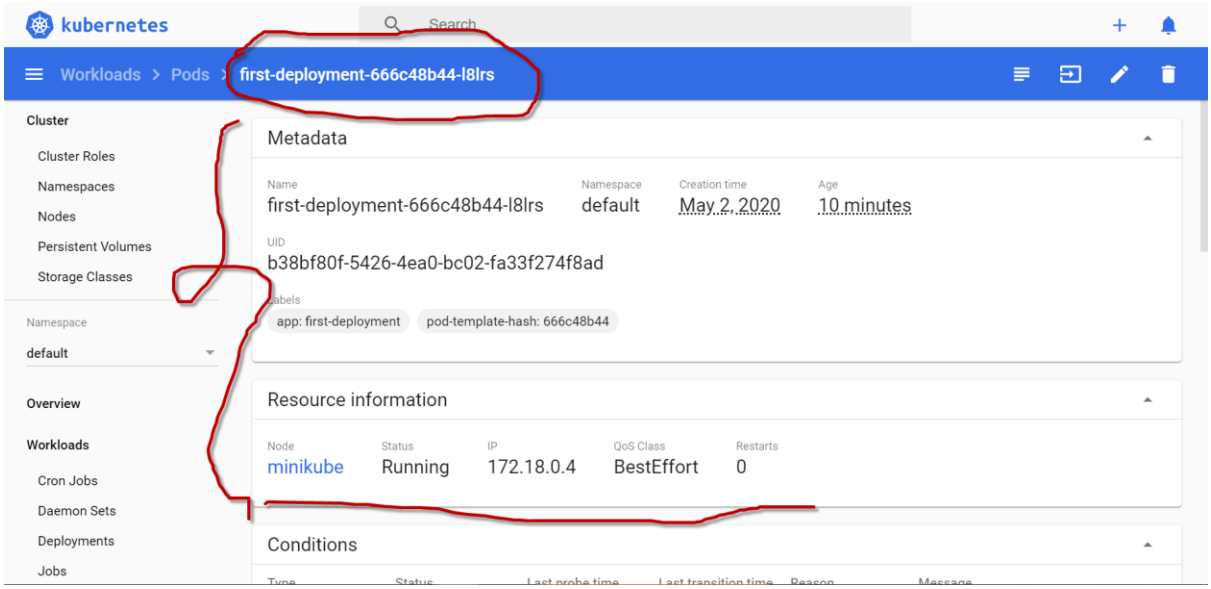


Fig: 5

Services:

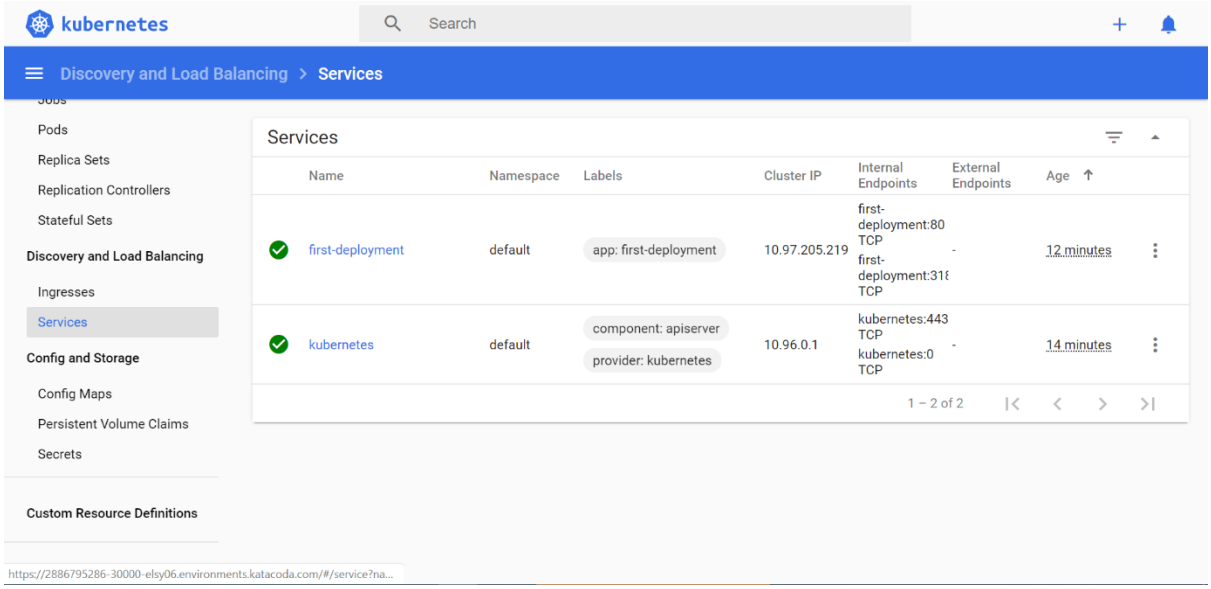


Fig: 6

Learning objectives:

**This scenario has explained how to launch a Single Node Kubernetes cluster. Using Minikube is ideal for development environments and testing Kubernetes locally.**

References:

https://www.katacoda.com/courses/kubernetes/launch-single-node-cluster

https://kubernetes.io/docs/setup/learning-environment/minikube/