

Kubernetes Lab

Create a Cluster

Here we are starting Kubernetes on our machine. Before running any commands we need to ensure that our docker application is running first. Then we can run the following command:

```
minikube start
```

```
denis.pk@Deniss-MacBook-Air ~ % minikube start
🐳 minikube v1.32.0 on Darwin 14.3.1 (arm64)
🔔 minikube 1.33.0 is available! Download it: https://github.com/kubernetes/minikube/releases/tag/v1.33.0
💡 To disable this notice, run: 'minikube config set WantUpdateNotification false'

🌟 Using the docker driver based on existing profile

🚫 Exiting due to PROVIDER_DOCKER_NOT_RUNNING: "docker version --format <no value>--<no value>:<no value>" exit status 1: Cannot connect to the Docker daemon at unix:///Users/denis.pk/.docker/run/docker.sock. Is the docker daemon running?
💡 Suggestion: Start the Docker service
📖 Documentation: https://minikube.sigs.k8s.io/docs/drivers/docker/

denis.pk@Deniss-MacBook-Air ~ %
```

In a separate tab we will run the following command to start the minikube dashboard

```
minikube dashboard
```

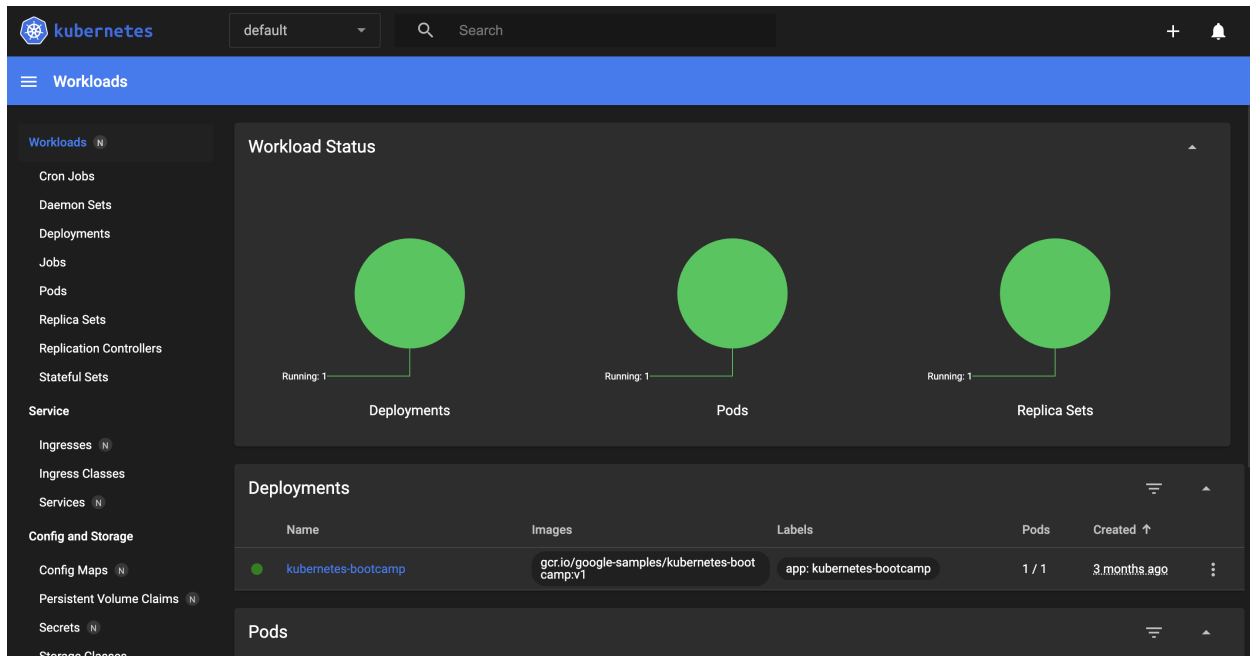
```

Enabling dashboard ...
  ■ Using image docker.io/kubernetes/metrics-scraper:v1.0.8
  ■ Using image docker.io/kubernetes/dashboard:v2.7.0
💡 Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube addons enable metrics-server

🤖 Verifying dashboard health ...
🚀 Launching proxy ...
🤖 Verifying proxy health ...
🔗 Opening http://127.0.0.1:50070/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...

```



Deploy an App

When we run the following command, we are creating our deployment cluster:

```
kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080
```

```

denis.pk@Deniss-MacBook-Air ~ % kubectl create deployment hello-node --image=
registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=
8080
deployment.apps/hello-node created

```

In order to view our deployment, we can run the following command:

```
kubectl get deployments
```

```
denis.pk@Deniss-MacBook-Air ~ % kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
hello-node	1/1	1	1	75s

To view our pod we can run the following command:

```
kubectl get pods
```

```
denis.pk@Deniss-MacBook-Air ~ % kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
hello-node-ccf4b9788-hw9q9	1/1	Running	0	6m23s

Running the following command shows us the application logs for a container in a pod:

```
kubectl logs hello-node-ccf4b9788-hw9q9
```

```
denis.pk@Deniss-MacBook-Air ~ % kubectl logs hello-node-ccf4b9788-hw9q9
I0509 16:35:46.004764      1 log.go:195] Started HTTP server on port 8080
I0509 16:35:46.005378      1 log.go:195] Started UDP server on port 8081
```

Explore Your App

Expose Your App Publicly

Scale Your App

Update Your App