**ASSIGNMENT – 5**

**INFORMATION MANAGEMENT**

NAME: JAHNAVI ANGATI (ja54632), KARTHICK VEL KATHIRVEL (kk37347)

Due Date: 11-05-2023

**Goal:**

A hand on experience with Kubernetes to complete as a homework to learn the basics of Kubernetes with an interactive tutorial.

**Provider:**

Kubernetes (Official Kubernetes Website)

**Duration:**

Self-paced, typically 3 hours

**Link:**

https://kubernetes.io/docs/tutorials/stateless-application/hello-minikube/

**Instructions:**

1. Set up a local Kubernetes cluster using Minikube.
2. In the website's Tutorial tab, go through the 'Hello Minikube'
3. In the Tutorial tab's 'Learn Kubernetes Basics' go through the

* 'Create a Cluster'
* 'Deploy an App'
* 'Explore your app'
* 'Expose your App'
* 'Scale your App'

1. Obtain screenshots showing that each step above is done, paste into a file and upload the file.
2. **Set up a local Kubernetes cluster using Minikube**

*minikube dashboard*

A screenshot of a computer

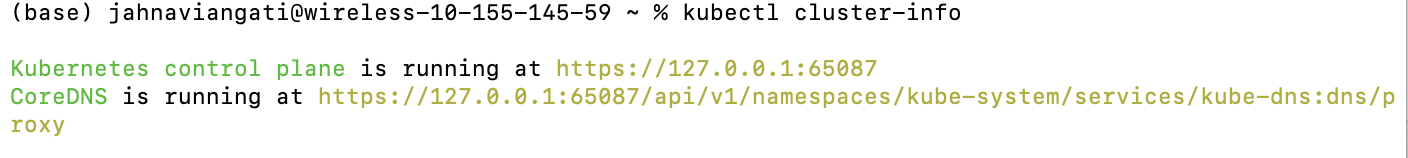
Description automatically generated

1. **Hello Minikube: 'Create a Cluster'**

**A diagram of a cluster diagram

Description automatically generated**

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

**

A screenshot of a computer

Description automatically generated

A computer screen shot of a white screen

Description automatically generated

*kubectl expose deployment hello-node --type=LoadBalancer --port=8080*

A screenshot of a phone

Description automatically generated

*minikube addons list*

A screenshot of a computer

Description automatically generated

*minikube addons enable metrics-server*



View the Pod and Service created by installing that addon:

A white paper with black text

Description automatically generated

1. **Deploy an App**

**A diagram of a software application

Description automatically generated with medium confidence**

*kubectl create deployment kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1*

A screenshot of a computer

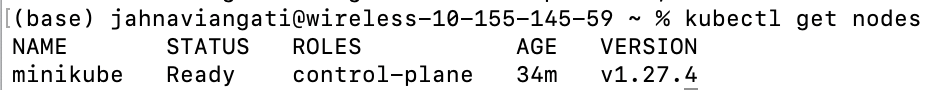
Description automatically generated

*export POD\_NAME=$(kubectl get pods -o go-template --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}')*

*echo Name of the Pod: $POD\_NAME*



*kubectl get nodes*

**

1. **Explore your App**

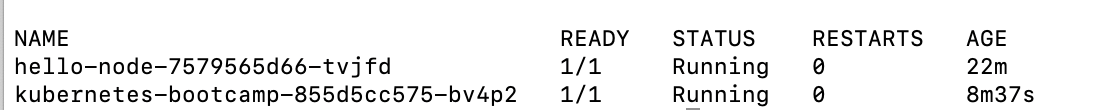
**A diagram of different types of cubes

Description automatically generated**

**A screenshot of a diagram

Description automatically generated**

*kubectl get pods*

****

*kubectl describe pods*

**A screenshot of a computer

Description automatically generated**

1. **Expose your App Publicly**

**A diagram of a application

Description automatically generated**

A screenshot of a computer

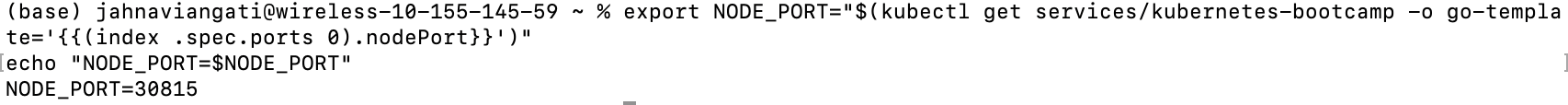
Description automatically generated

We have now a running Service called kubernetes-bootcamp. Here we see that the Service received a unique cluster-IP, an internal port and an external-IP (the IP of the Node).

*kubectl describe services/kubernetes-bootcamp*

A screenshot of a computer program

Description automatically generated



A screenshot of a computer program

Description automatically generated

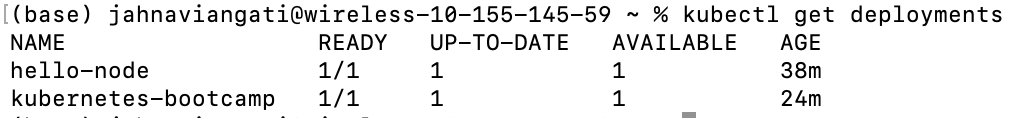
A close-up of a white background

Description automatically generated

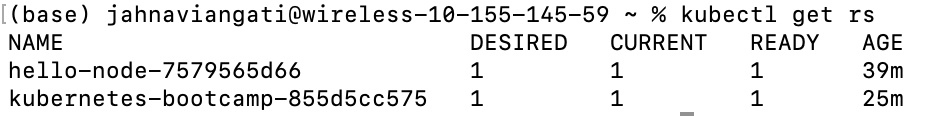
1. **Scale your App**

**A diagram of a service

Description automatically generated**



* NAME lists the names of the Deployments in the cluster.
* READY shows the ratio of CURRENT/DESIRED replicas
* UP-TO-DATE displays the number of replicas that have been updated to achieve the desired state.
* AVAILABLE displays how many replicas of the application are available to your users.
* AGE displays the amount of time that the application has been running.



Let’s scale the Deployment to 4 replicas. We’ll use the kubectl scale command, followed by the Deployment type, name and desired number of instances:

A white background with black text

Description automatically generated

Checking if the pods have changed:

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Update your App**

**A diagram of a service

Description automatically generated**

To update the image of the application to version 2, use the set image subcommand, followed by the deployment name and the new image version:

A screenshot of a computer program

Description automatically generated

Verifying the Update:

A screenshot of a computer

Description automatically generated



Rollback an update:

A screenshot of a computer

Description automatically generated

Cleaning up local cluster:

