Lab 4

By Sarah Martin C00257967

# **Learning Kubernetes Basics**

## **Part 1: Create A Cluster**

I started minikube in my terminal.

A screenshot of a computer program

Description automatically generated

Installed kubectl.

A black background with white text

Description automatically generated

Opened the minikube dashboard.

A black screen with white text

Description automatically generated

I used the kubectl create command to create a Deployment that manages a Pod and then viewed the Deployment.

A black screen with a black background

Description automatically generated

I used this command to get the list of pods.

A black background with white text

Description automatically generated

I used this command to view the cluster commands

A black screen with white text

Description automatically generated

I viewed the kubectl configuration.

A screenshot of a computer

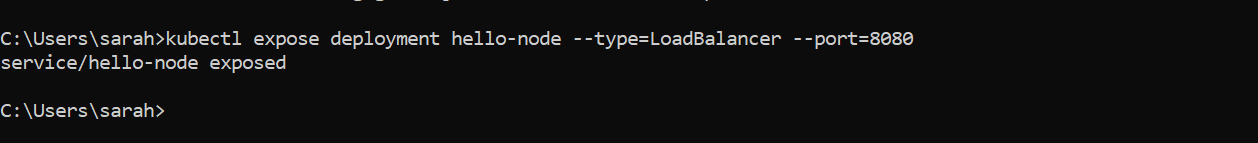
Description automatically generated

I then viewed the application logs for a container in a pod.

A black background with white text

Description automatically generated

I used this command to expose the Pod to the public internet using the kubectl expose command



I then viewed the service I created

A black screen with white text

Description automatically generated

This opened up a browser window on my laptop that serves the app and shows the app's response

A computer screen shot of a black screen

Description automatically generated

The browser window:

A white background with black text

Description automatically generated

Enable Addons:

List of the currently supported addons.

A screenshot of a computer screen

Description automatically generated

Example of enabling an addon such as metrics-server.

A black screen with white text

Description automatically generated

I view the Pod and Service I created by installing the addon:

A black screen with white text

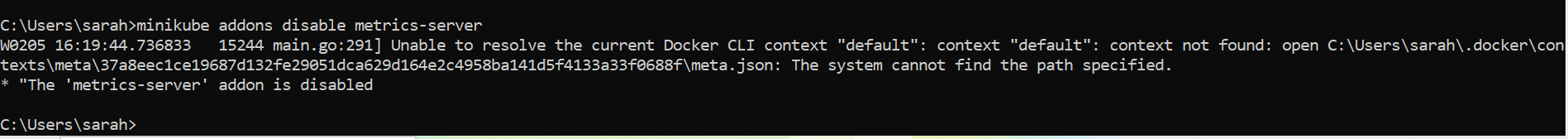
Description automatically generated

the output from metrics-server

A black background with white text

Description automatically generated

I disabled metrics-server.

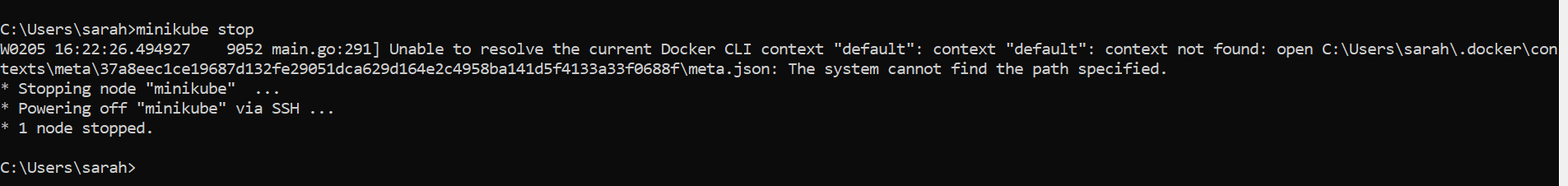


I cleaned up the resources I created in my cluster.

A screen shot of a computer

Description automatically generated

I then stopped the minikube cluster.



# **Part 2: Deploy An App**

I checked the kubectl version.

A black screen with white text

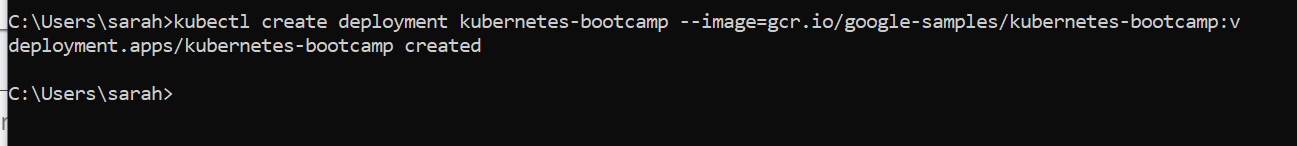
Description automatically generated

Viewed the nodes in the cluster.

A black screen with white text

Description automatically generated

Deployed the first app on Kubernetes with the kubectl create deployment command.

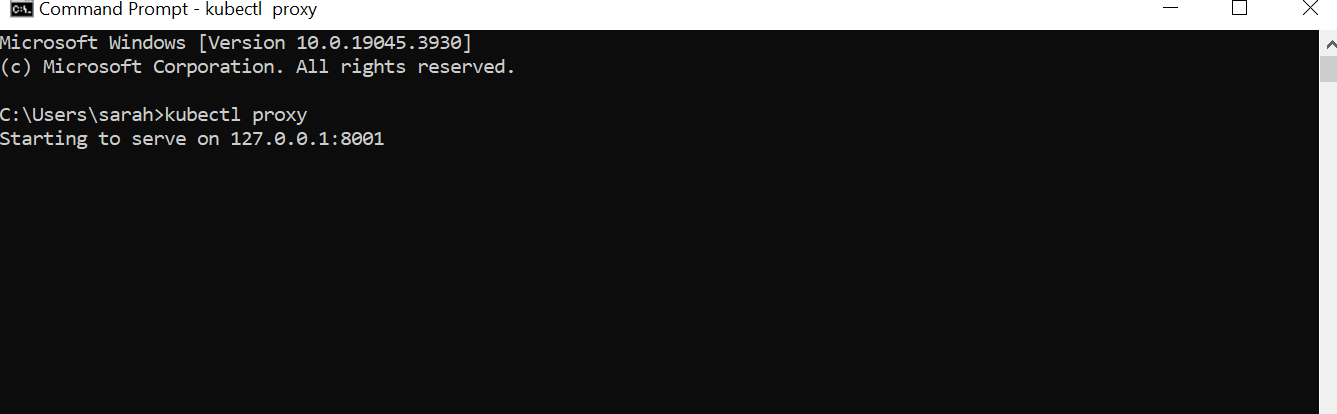


List of deployments.

A black screen with white text

Description automatically generated

Open up a new terminal to start a proxy.



can see all those APIs hosted through the proxy endpoint.

A computer screen with white text

Description automatically generated

First I had to get the pod name and then I had to store the pod name in a variable called POD\_NAME.

A screenshot of a computer

Description automatically generated

I can now access the Pod through the proxied API

A computer screen with text on it

Description automatically generated

## **Part 3: Viewing Pods and Nodes**

I used the get pods command to view the existing pods.

A screenshot of a computer screen

Description automatically generated

I used the describe pods command to view the pods details.

A screen shot of a computer

Description automatically generated

I then viewed the output of the application by running a curl request.

A screenshot of a computer program

Description automatically generated

A black screen with white text

Description automatically generated

I then viewed the container logs.

A screen shot of a computer

Description automatically generated

Listed the environment variables.

A computer screen with white text

Description automatically generated

Started a bash session.

A black background with white text

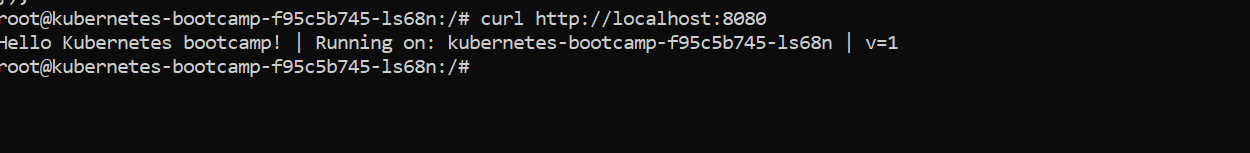
Description automatically generated

Viewed the server.js file.

A computer screen with white text

Description automatically generated

Checked that the application is up and running by using the curl command.



## **Part 4: Expose Your App Publicly**

Used the get pods command to look for existing pods. I then used the get services command to list the current services from the cluster.

A screen shot of a computer

Description automatically generated

I used the expose command to create a new service and expose it to external traffic. I then used the get services command to show the a running service called Kubernetes-bootcamp.

A screen shot of a computer

Description automatically generated

Found what port was opened externally.

A screenshot of a computer

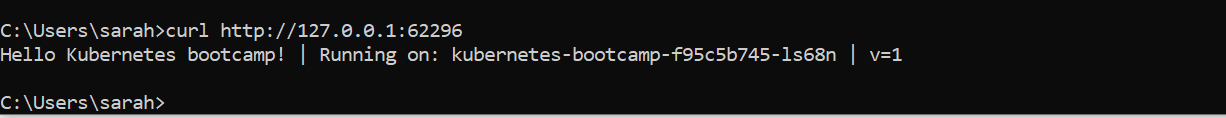
Description automatically generated

Since Im running minikube with Docker Desktop as the container driver, I ran this command in a separate terminal window.

A computer screen with white text

Description automatically generated

Then used the given URL to access the app.



I used the describe deployment command to view the automatically created label for the pod.

A computer screen with white text

Description automatically generated

I used this label to query the list of pods. I used the get pods command with -l as a parameter and followed by the label values. I did the same to list the existing services.

A screen shot of a computer

Description automatically generated

I applied a new label. I checked it with the describe pod command.

A screen shot of a computer

Description automatically generated

I then queried the list of pods using the new label.

A black screen with white text

Description automatically generated

I then deleted the service.

A screen shot of a computer

Description automatically generated

I then confirmed the service was gone.

A black background with white text

Description automatically generated

## **Part 5: Scaling Your App**

Used the get deployments command to list my deployments.

A black background with white text

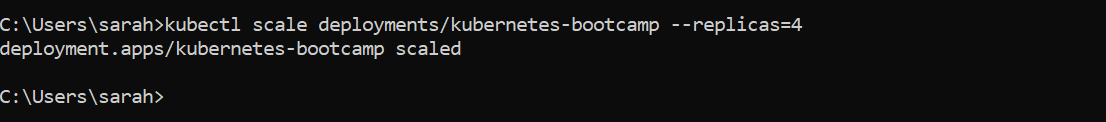
Description automatically generated

I viewed the ReplicaSet created by the deployment

A black screen with white text

Description automatically generated

I then scaled the deployment to four replicas.



I viewed the deployments.

A black screen with white text

Description automatically generated

I then checked if the number of pods changed. There are four pods now.

A black background with many small squares

Description automatically generated with medium confidence

Checked to see if the change was registered in the deployment events log.

A screenshot of a computer

Description automatically generated

I found the exposed IP and Port.

A black screen with white text

Description automatically generated

I used the curl command to the exposed IP address and port. I executed the command multiple times and noted that we hit a different pod with every request.

A screen shot of a computer code

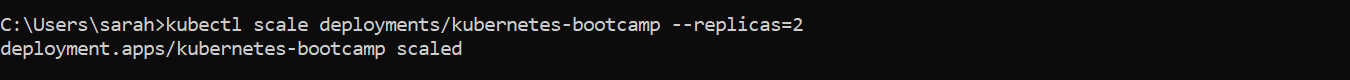
Description automatically generated

Listed the deployments.

A black screen with white text

Description automatically generated

I scaled down the deployments to two replicas by running the scale command.



The number of replicas decreased to 2. I listed the number of pods with get pods command.

A black background with many small letters

Description automatically generated with medium confidence

## **Part 6: Update Your App**

I listed the running pods and viewed the current image version of the app and looked for the image field.

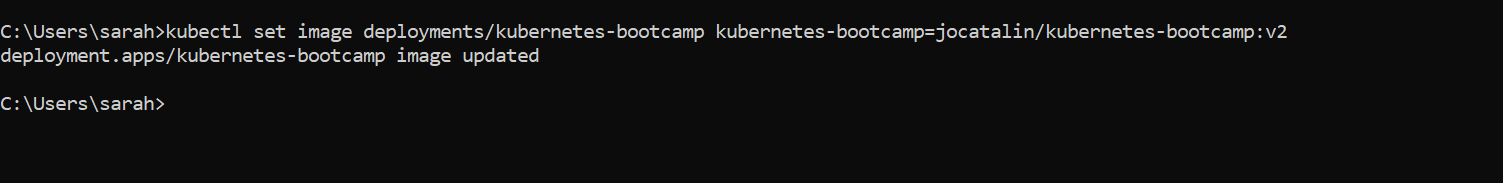
A screen shot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

I updated the image of the application to version 2 by using the set image command.



I then checked the status of the new Pods and viewed the one terminating with the get pods command.

A screenshot of a computer program

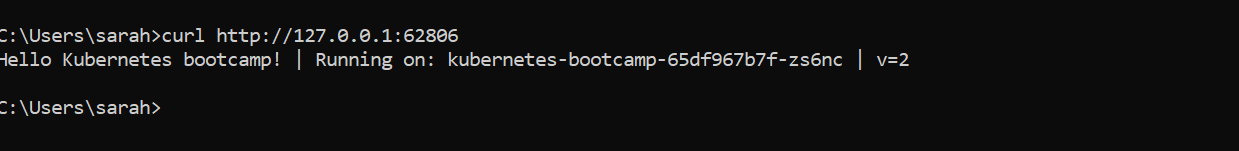
Description automatically generated

I checked if the app is running and found the exposed IP address and port.

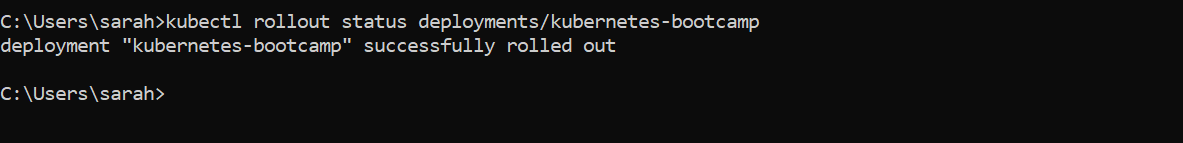
A screenshot of a computer program

Description automatically generated

I then did a curl to the exposed IP and port.



I confirmed the update by running the rollout status command.



I then viewed the current image of the app by running the describe pods command.

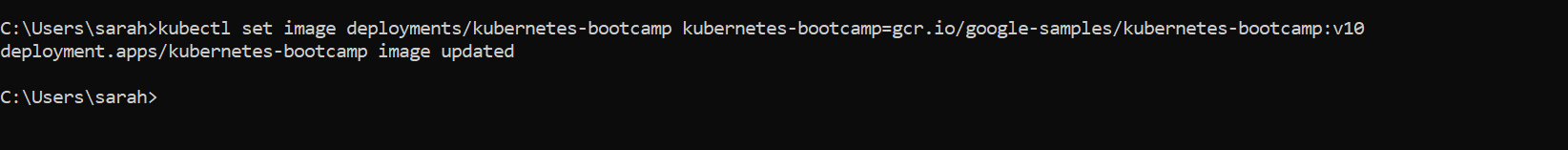
A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

I performed another update and deployed an image tagged with v10.



I ran the get deployments command to see the status of the deployment but the output doesn’t list the desired number of available pods. I then ran the get pods command to list all pods.

A black screen with white text

Description automatically generated

I then ran the describe pods command to get insight into the problem of some of the pods having the status of ImagePullBackOff. I noticed in the events section of the output of the effected pods that the v10 image version did not exist in the repository.

A screenshot of a computer

Description automatically generated

A computer screen with text on it

Description automatically generated

I rolled back the deployment to the last working version. I used the get pods command to list the pods and used the describe pods command. The deployment is now using a stable version of the app (v2).

A black screen with white text

Description automatically generated

A computer screen with text

Description automatically generated

I then cleaned up the local cluster.

A black screen with white text

Description automatically generated