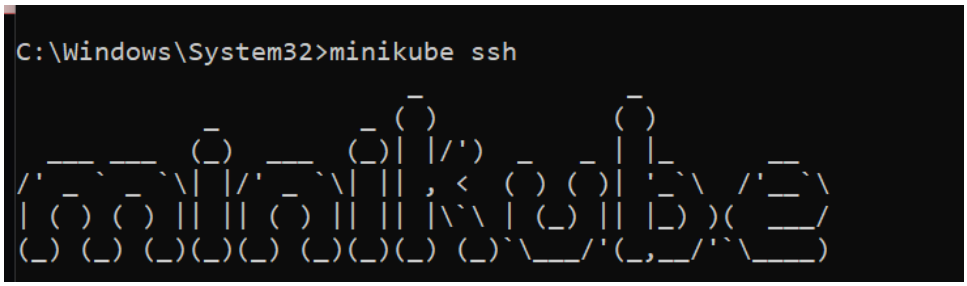
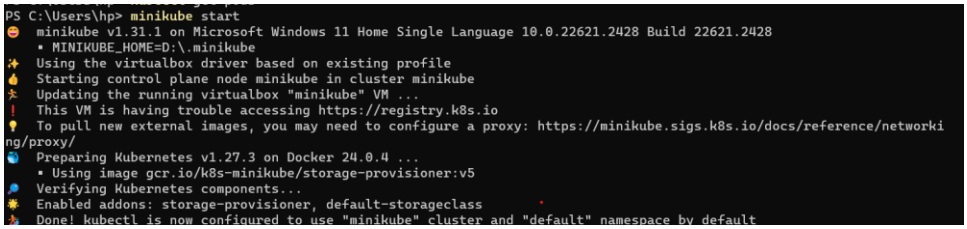
Lab Exercise 7– Creating Pods in Kubernetes

Below is a lab exercise that will help you understand and practice creating pods in Kubernetes:

Task 1: Start Kubernetes in Docker-Desktop

- Start Kubernetes service in Docker-Desktop





Task 2: Creating a Simple Pod

- Create a simple YAML manifest file named pod.yaml to define a basic Pod in Kubernetes. An example of the file content is as follows:

```
apiVersion: v1

kind: Pod

metadata:

  name: my-pod

spec:

  containers:

  - name: my-container

    image: nginx
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

hp@t4r MINGW64 ~/Desktop/ACO-LAB-2021-25-SUBMISSION/R2142211343/Exp7 (main)
$ touch pod.yaml
```

- Apply the Pod configuration using the following command:

```
kubectl apply -f pod.yaml
```

```
PS C:\Users\hp\Desktop\ACO-LAB-2021-25-SUBMISSION\R2142211343\Exp7> kubectl apply -f .\pod.yaml
pod/my-pod created
```

Check the status of the Pod using the following command:

```
kubectl get pods
```

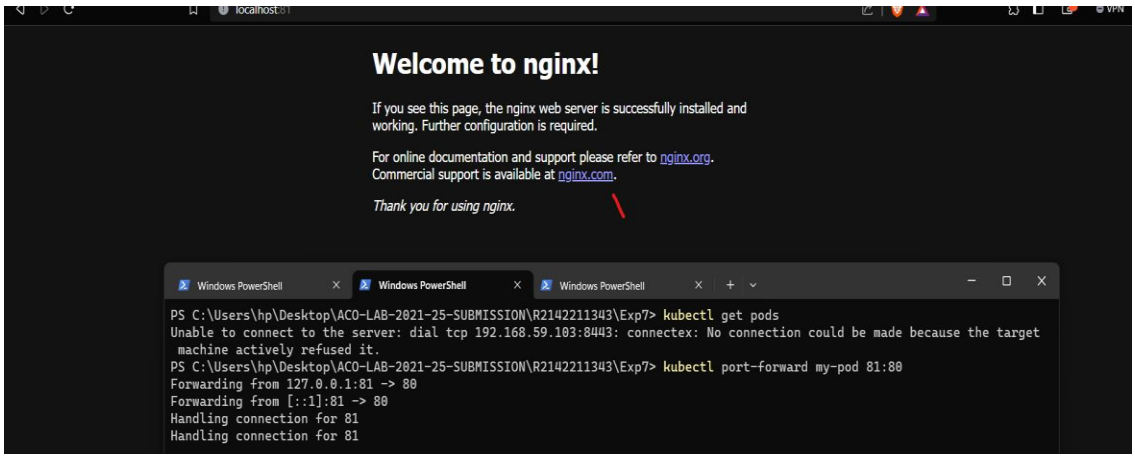
```
PS C:\Users\hp\Desktop\ACO-LAB-2021-25-SUBMISSION\R2142211343\Exp7> kubectl get pods
NAME     READY   STATUS             RESTARTS         AGE
my-pod   0/1     Pending            0                0s
mydb     0/1     CrashLoopBackOff   22 (4m21s ago)   84d
```

Task 3: Accessing the Pod

Access the Pod by using port forwarding to the container. Run the following command:

```
kubectl port-forward my-pod 81:80
```

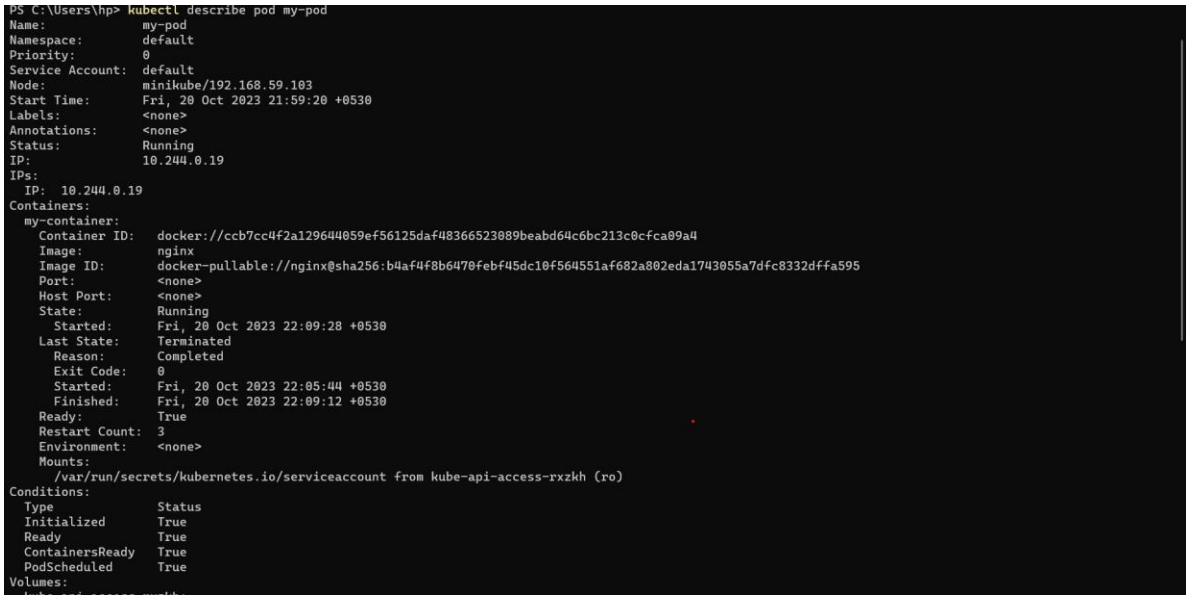
Access the Nginx server running in the Pod by opening a web browser and navigating to http://localhost:81.



Task 4: Exploring Pod Details

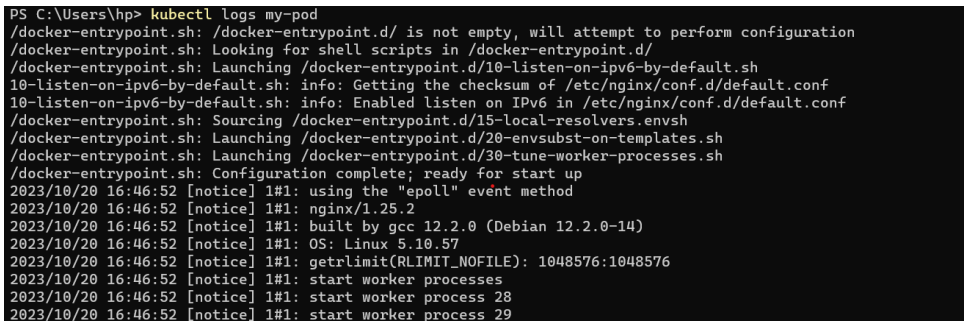
Retrieve detailed information about the Pod using the following command:

```
kubectl describe pod my-pod
```



Check the logs of the Pod to understand its behavior using the following command:

```
kubectl logs my-pod
```



Task 5: Deleting the Pod

Delete the Pod using the following command:

```
kubectl delete pod my-pod
```

Verify that the Pod has been deleted by running the kubectl get pods command.

Task 6: Advanced Pod Configuration

- Experiment with advanced Pod configuration options such as environment variables, volume mounts, resource limits, and labels.
- Update the Pod manifest file and apply the changes to the Kubernetes cluster.

```yaml
apiVersion: v1

kind: Pod

metadata:

  name: my-second-pod

spec:
  containers:
  - name: my-container2
    image: httpd
    ports:
    - containerPort: 30001
      protocol: TCP
```

Task 7: Cleanup

Delete any remaining Pods, services, and deployments created during the exercise using the appropriate kubectl delete commands.

Task 8: Documentation and Best Practices

Document your findings and the best practices for creating and managing Pods in Kubernetes.

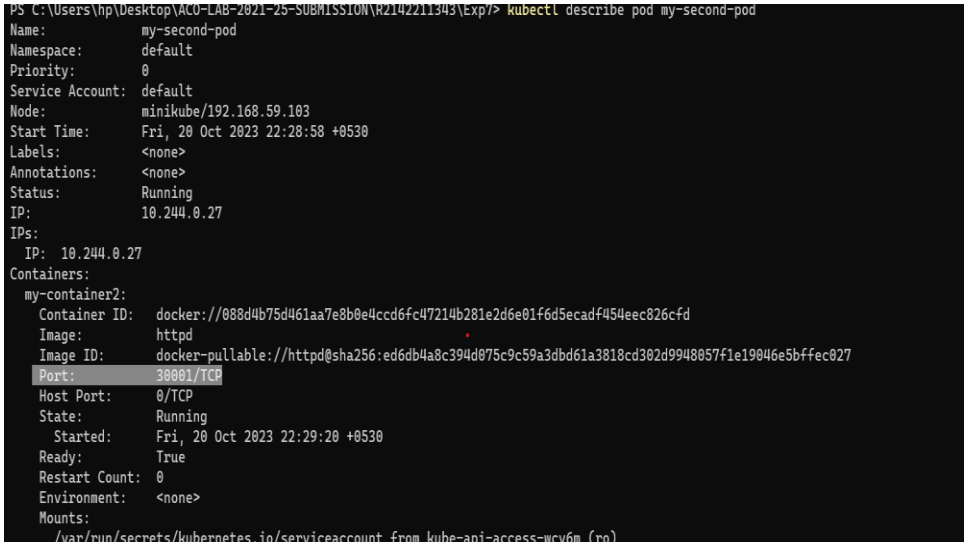
Through this exercise, you'll gain a better understanding of how to create, manage, and interact with Pods in Kubernetes. Adjust the exercise based on your specific use case and requirements.

We will try to assign a port in a new yaml file and try to find out how a port can be assigned through yaml file

```yaml
apiVersion: v1

kind: Pod

metadata:

  name: my-second-pod

spec:
  containers:
  - name: my-container2
    image: httpd
    ports:
    - containerPort: 30001
      protocol: TCP
```

After creating a new pod with image of apache web server.We will try to find out the port we assigned using kubectl describe

```
PS C:\Users\hp\Desktop\ACO-LAB-2021-25-SUBMISSION\R2142211343\Exp7> kubectl apply -f .\pod2.yml
pod/my-second-pod created
```



Here we can see that the port number that we assigned in the Yaml file was 30001 and after using kubectl describe we can see that when the pod is being used its listening in nport number 30001