

KUBERNETES ASSIGNMENT -2

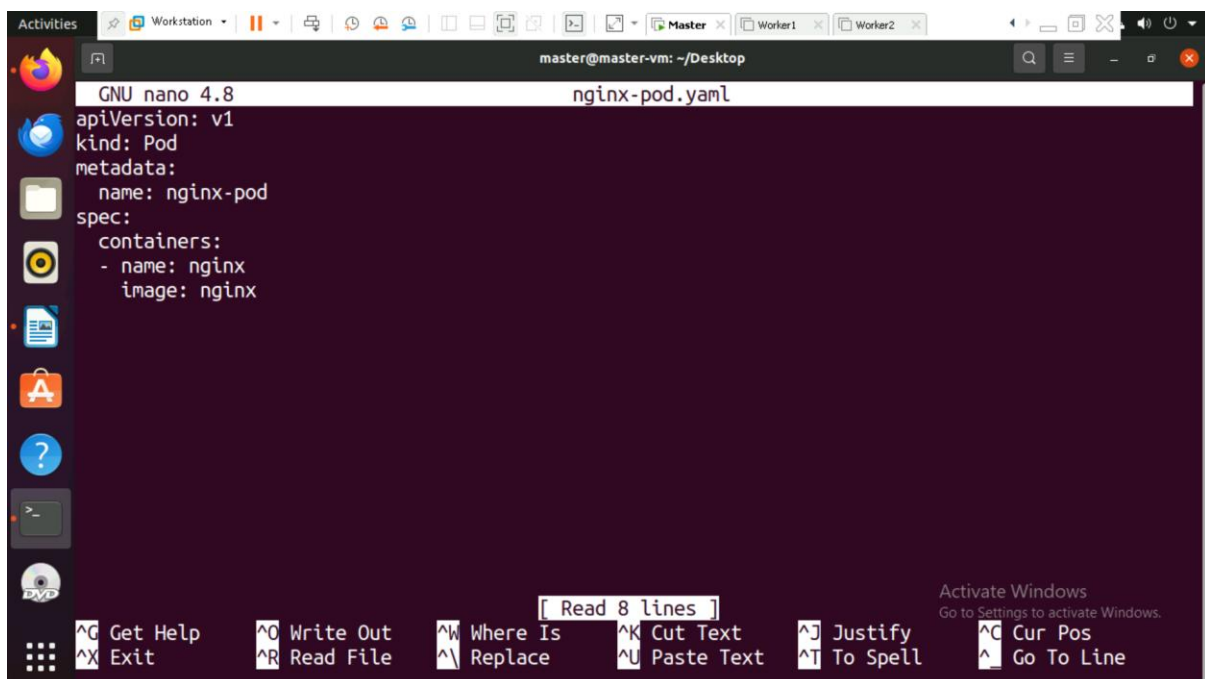
Exercise 3: Create a Pod using a YAML Manifest

Objective: Define and deploy a pod using YAML.

Steps & Commands:

1. Create a YAML file (nginx-pod.yaml):

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx
    image: nginx
```



2. **Apply the YAML file:**

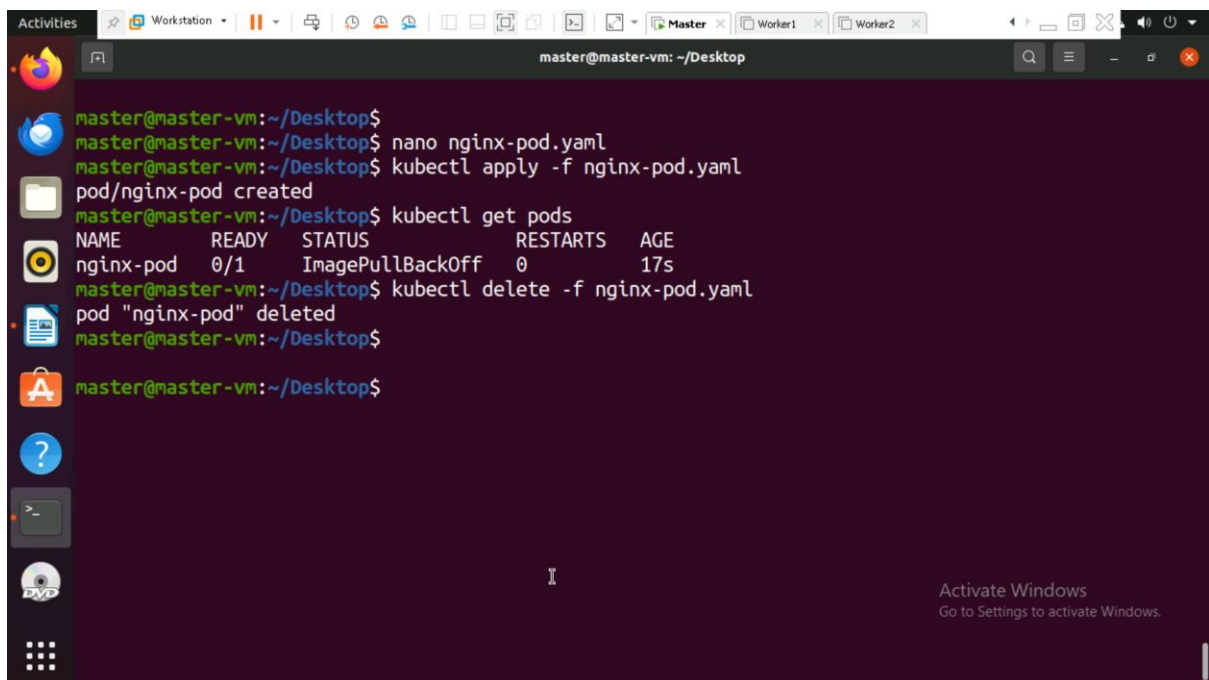
```
kubectl apply -f nginx-pod.yaml
```

3. **Check if the pod is running:**

```
kubectl get pods
```

4. **Delete the pod using YAML:**

```
kubectl delete -f nginx-pod.yaml
```



The screenshot shows a terminal window titled 'master@master-vm: ~/Desktop'. The user has executed the following commands and received the following output:

```
master@master-vm:~/Desktop$ nano nginx-pod.yaml
master@master-vm:~/Desktop$ kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
master@master-vm:~/Desktop$ kubectl get pods
NAME      READY   STATUS             RESTARTS   AGE
nginx-pod  0/1     ImagePullBackOff   0           17s
master@master-vm:~/Desktop$ kubectl delete -f nginx-pod.yaml
pod "nginx-pod" deleted
master@master-vm:~/Desktop$
```

The terminal window also shows a sidebar with application icons and a top bar with window management controls. A Windows watermark is visible in the bottom right corner.

Exercise 4: Create and Use a ConfigMap

Objective: Store environment variables in a ConfigMap and use it in a pod.

Steps & Commands:

1. **Create a ConfigMap:**

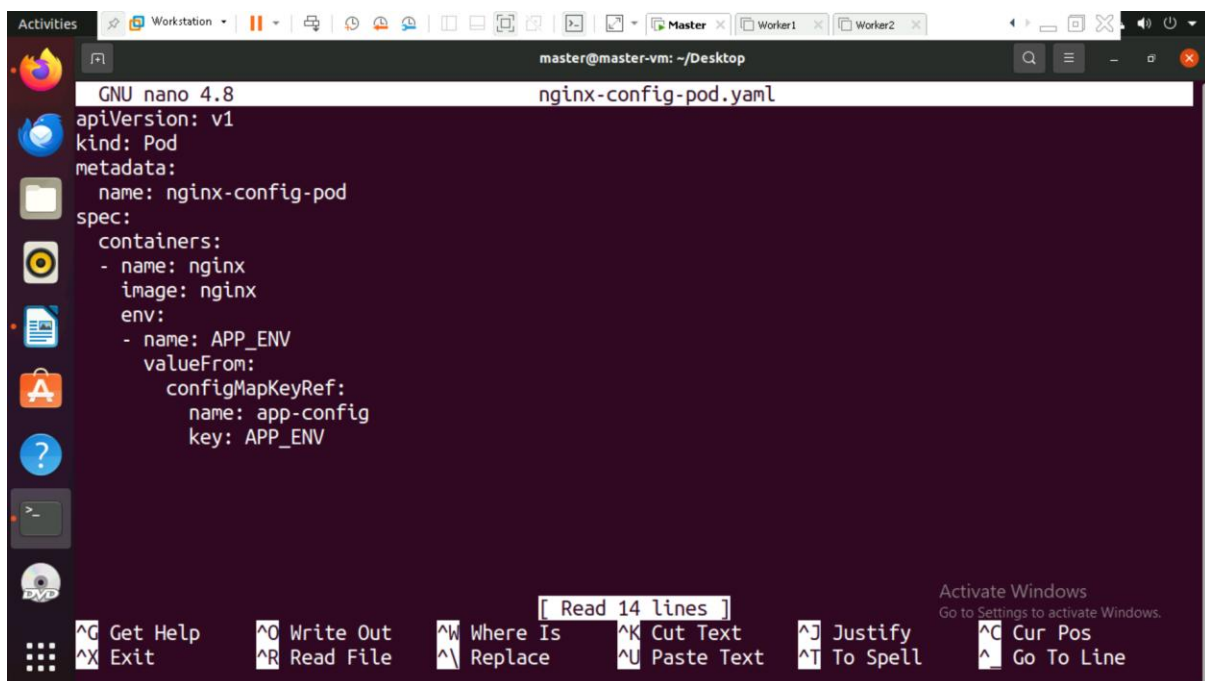
```
kubectl create configmap app-config --from-literal=APP_ENV=production
```

2. **Verify ConfigMap:**

```
kubectl get configmaps app-config -o yaml
```

3. Create a pod that uses the ConfigMap (nginx-config-pod.yaml):

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-config-pod
spec:
  containers:
  - name: nginx
    image: nginx
    env:
    - name: APP_ENV
      valueFrom:
        configMapKeyRef:
          name: app-config
          key: APP_ENV
```



4. Deploy the pod:

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

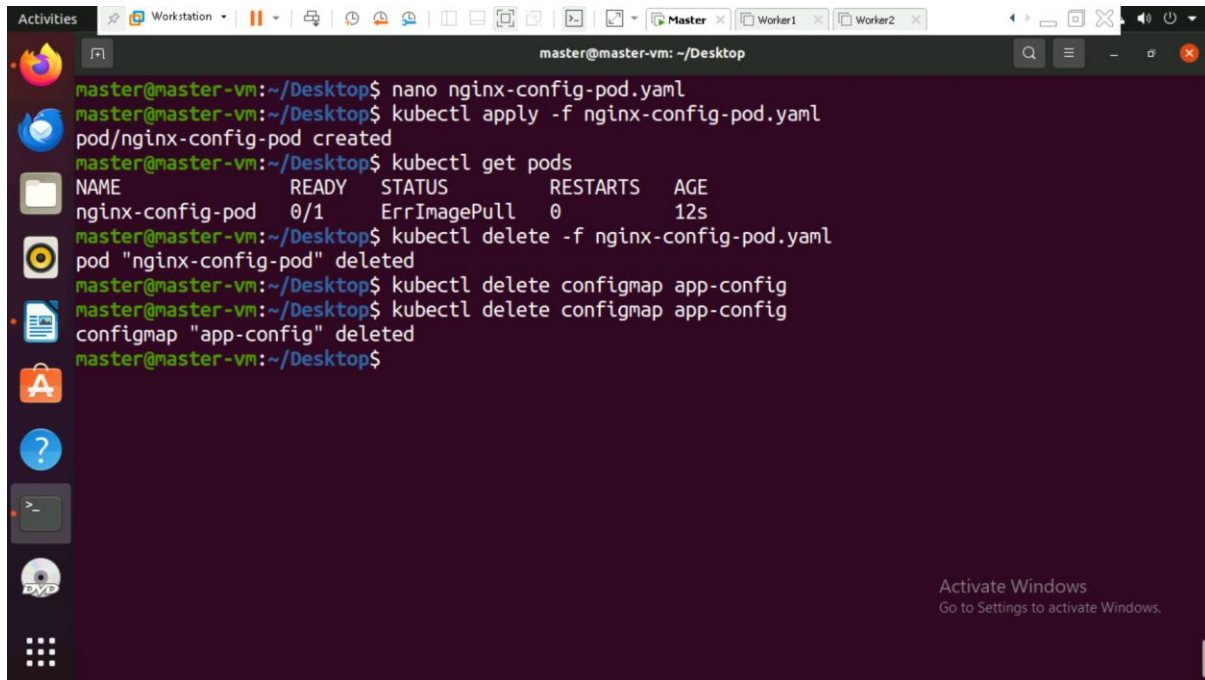
5. Check if the pod is running:

```
kubectl get pods
```

6. Delete the pod and ConfigMap:

```
kubectl delete -f nginx-config-pod.yaml
```

```
kubectl delete configmap app-config
```



A terminal window titled 'master@master-vm: ~/Desktop' showing the execution of several Kubernetes commands. The commands and their outputs are as follows:

```
master@master-vm:~/Desktop$ nano nginx-config-pod.yaml
master@master-vm:~/Desktop$ kubectl apply -f nginx-config-pod.yaml
pod/nginx-config-pod created
master@master-vm:~/Desktop$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-config-pod 0/1     ErrImagePull 0           12s
master@master-vm:~/Desktop$ kubectl delete -f nginx-config-pod.yaml
pod "nginx-config-pod" deleted
master@master-vm:~/Desktop$ kubectl delete configmap app-config
configmap "app-config" deleted
master@master-vm:~/Desktop$
```

The terminal output shows the creation of a pod, its status (ErrImagePull), and its subsequent deletion. It also shows the deletion of the ConfigMap 'app-config'. The terminal window has a dark purple background and a sidebar with application icons on the left. An 'Activate Windows' watermark is visible in the bottom right corner.