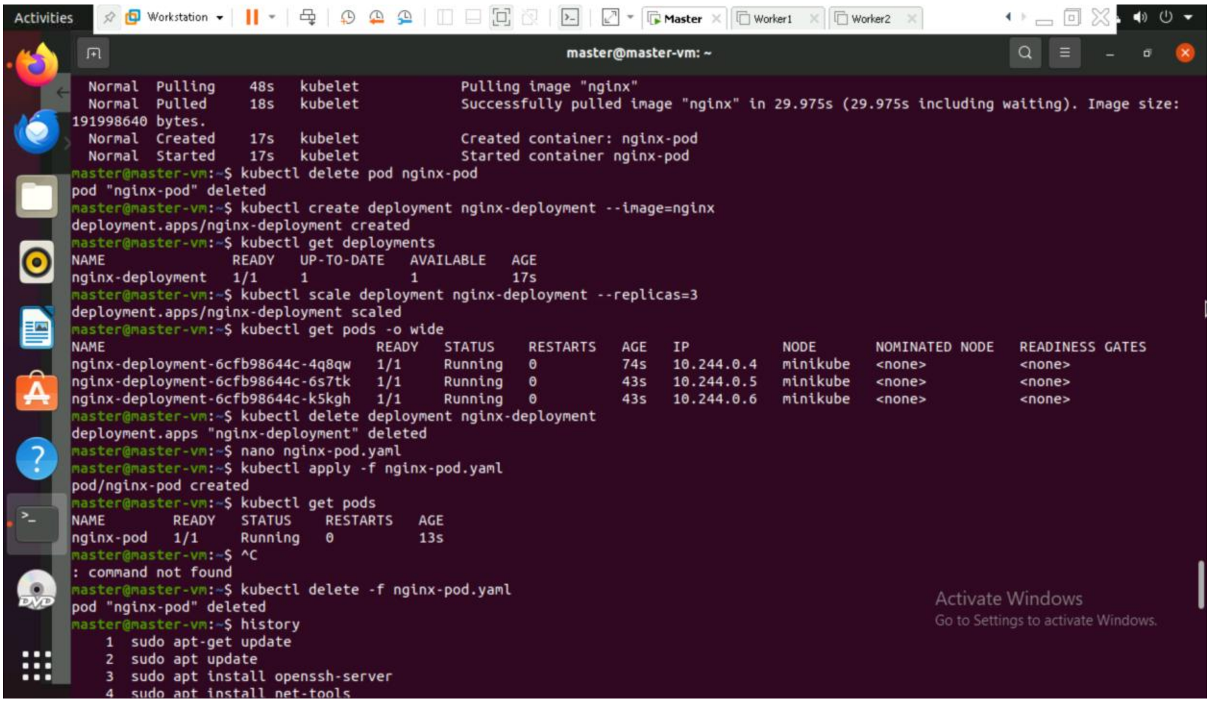
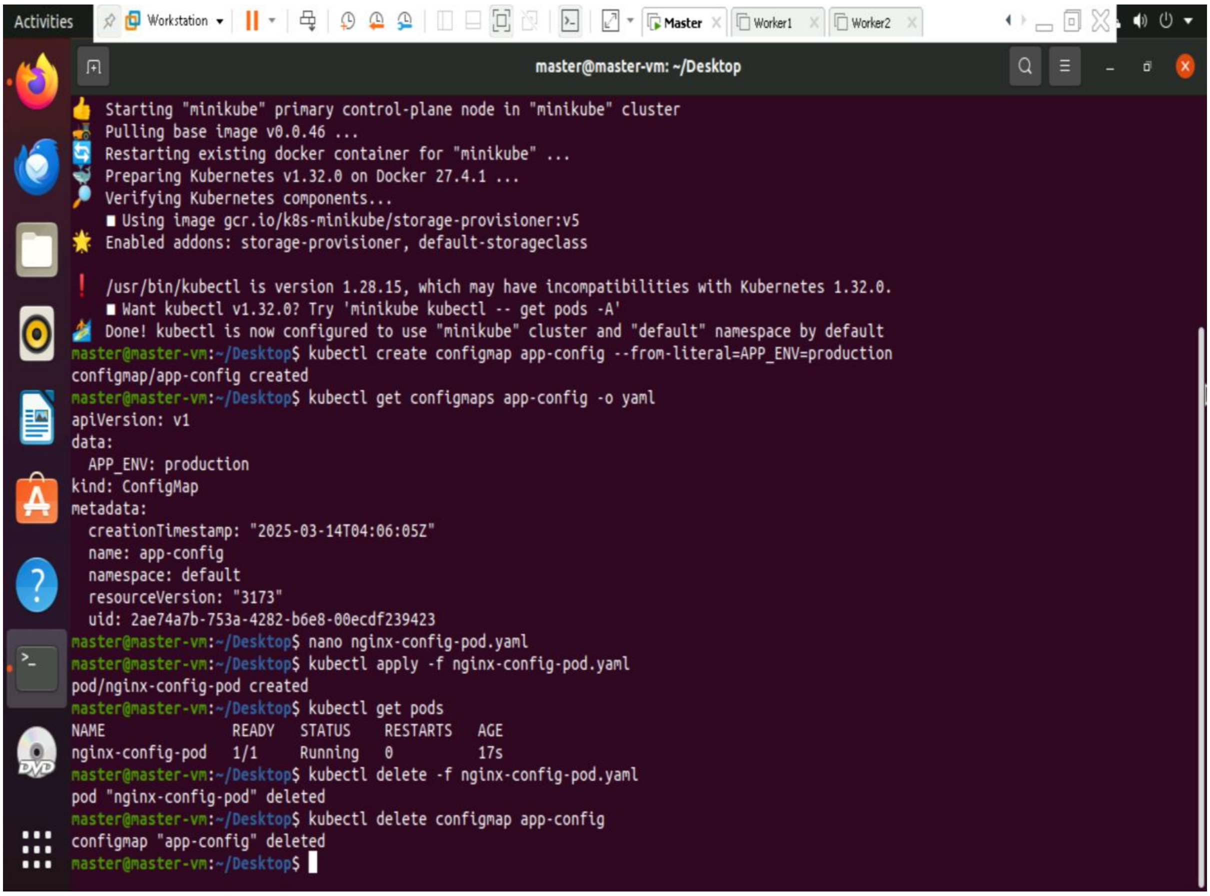
3)Create a Pod using a YAML Manifest



```
  Normal  Pulling    48s    kubelet             Pulling image "nginx"
  Normal  Pulled     18s    kubelet             Successfully pulled image "nginx" in 29.975s (29.975s including waiting). Image size:
191998640 bytes.
  Normal  Created    17s    kubelet             Created container: nginx-pod
  Normal  Started    17s    kubelet             Started container nginx-pod
master@master-vm:~$ kubectl delete pod nginx-pod
pod "nginx-pod" deleted
master@master-vm:~$ kubectl create deployment nginx-deployment --image=nginx
deployment.apps/nginx-deployment created
master@master-vm:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1     1            1           17s
master@master-vm:~$ kubectl scale deployment nginx-deployment --replicas=3
deployment.apps/nginx-deployment scaled
master@master-vm:~$ kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
nginx-deployment-6cfb98644c-4q8qw   1/1     Running   0          74s   10.244.0.4   minikube   <none>           <none>
nginx-deployment-6cfb98644c-6s7tk   1/1     Running   0          43s   10.244.0.5   minikube   <none>           <none>
nginx-deployment-6cfb98644c-k5kgh   1/1     Running   0          43s   10.244.0.6   minikube   <none>           <none>
master@master-vm:~$ kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
master@master-vm:~$ nano nginx-pod.yaml
master@master-vm:~$ kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
master@master-vm:~$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          13s
master@master-vm:~$ ^C
: command not found
master@master-vm:~$ kubectl delete -f nginx-pod.yaml
pod "nginx-pod" deleted
master@master-vm:~$ history
    1  sudo apt-get update
    2  sudo apt update
    3  sudo apt install openssh-server
    4  sudo apt install net-tools
```

4)Create and Use a ConfigMap



```
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass

❗  /usr/bin/kubectl is version 1.28.15, which may have incompatibilities with Kubernetes 1.32.0.
    ▪ Want kubectl v1.32.0? Try 'minikube kubectl -- get pods -A'
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
master@master-vm:~/Desktop$ kubectl create configmap app-config --from-literal=APP_ENV=production
configmap/app-config created
master@master-vm:~/Desktop$ kubectl get configmaps app-config -o yaml
apiVersion: v1
data:
  APP_ENV: production
kind: ConfigMap
metadata:
  creationTimestamp: "2025-03-14T04:06:05Z"
  name: app-config
  namespace: default
  resourceVersion: "3173"
  uid: 2ae74a7b-753a-4282-b6e8-00ecdf239423
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   1/1     Running   0          17s
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$ 
```