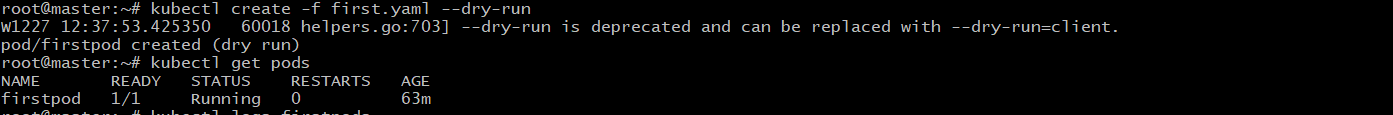
KUBERNETES[TASK-02]

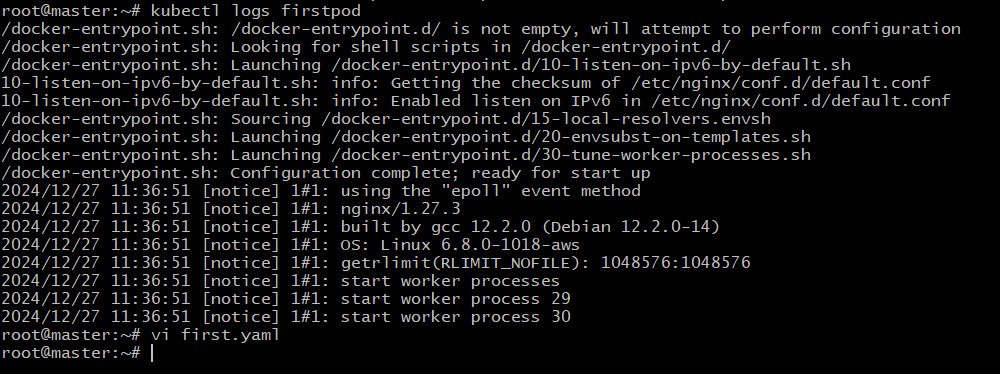
**1) Create a Simple Pod Using YAML Task: Write a YAML file to create a Pod named firstpod with an nginx container. Verify the Pod creation using kubectl get pods and check the logs of the container using kubectl logs firstpod.**



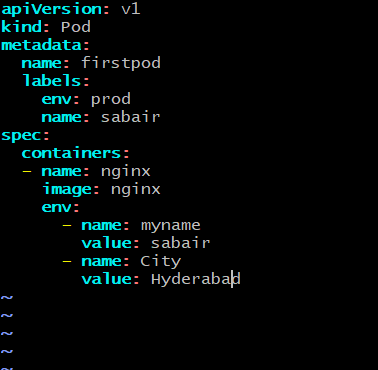
Verify the Pod creation using kubectl get pods:

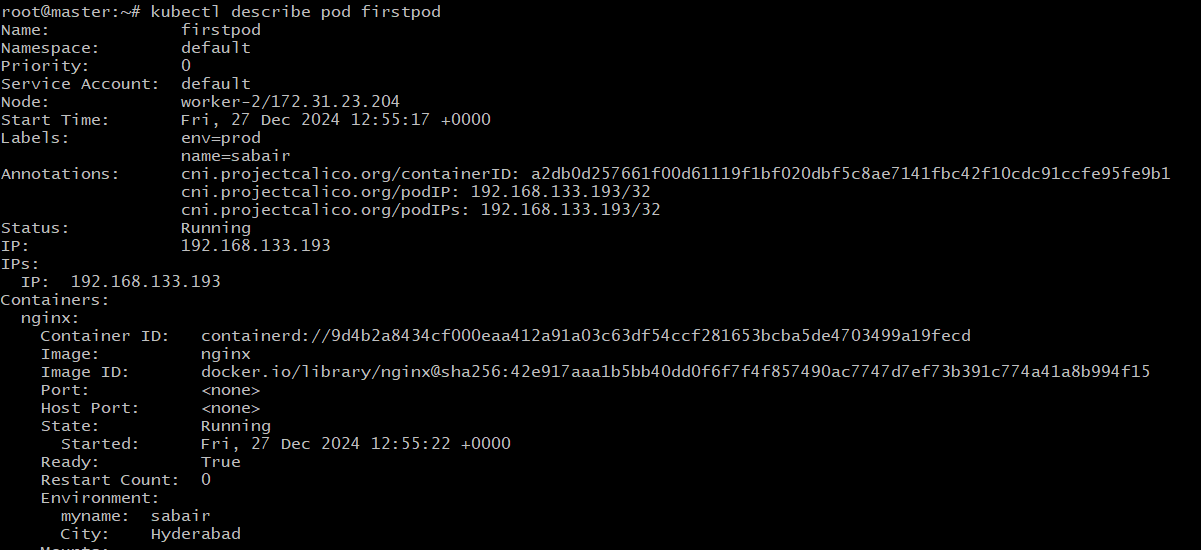


check the logs of the container using kubectl logs firstpod.

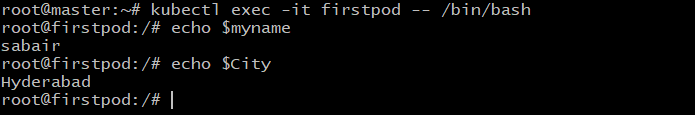


**2) Set Environment Variables in a Pod Task: Modify the YAML file to include environment variables myname: sabair and City: Hyderabad. Deploy the Pod and use kubectl exec <pod\_name> -- env to check if the environment variables are set properly.**



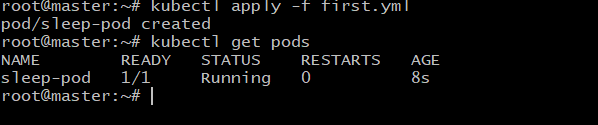


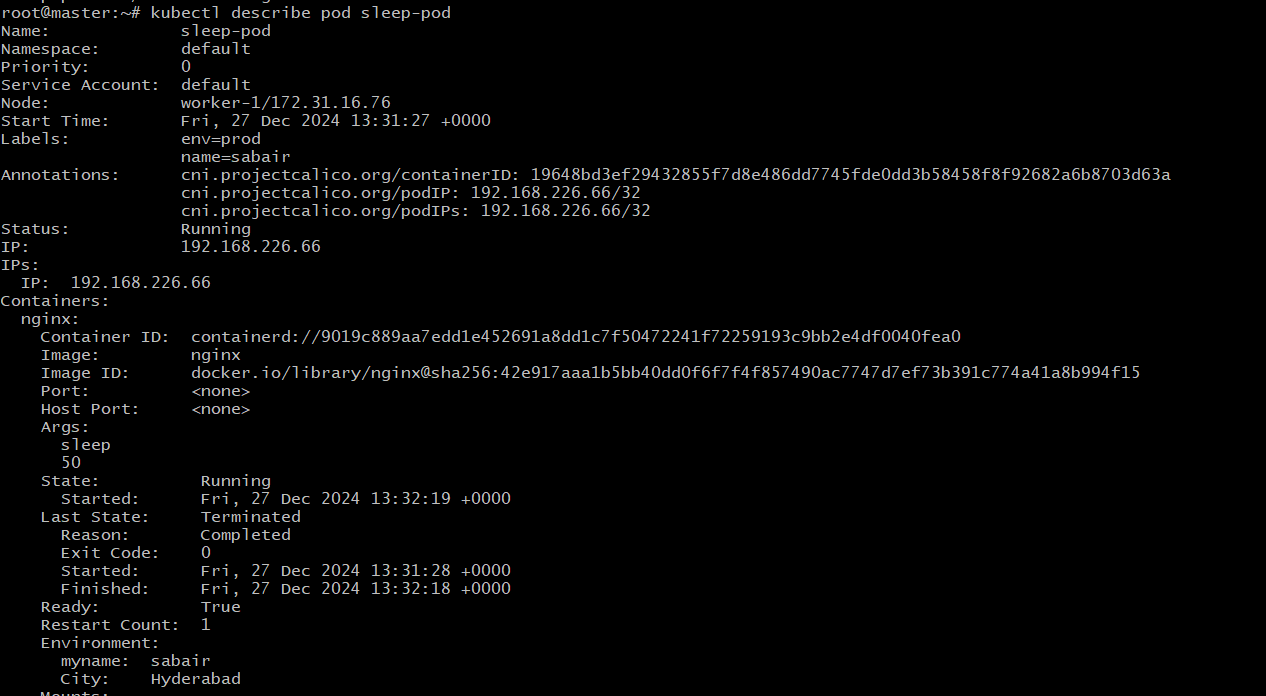
**Deploy the Pod and use kubectl exec <pod\_name> -- env to check if the environment variables are set properly**



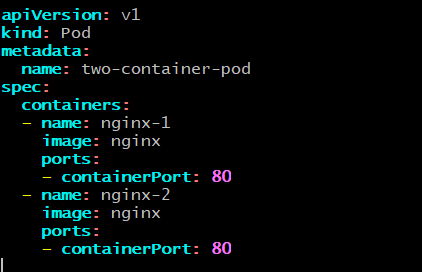
**3) Deploy a Pod with Commands (Args) in YAML Task: Modify the YAML file to add args that instruct the container to sleep for 50 seconds. Deploy the Pod and use kubectl describe pod to verify the args are correctly passed to the container.**

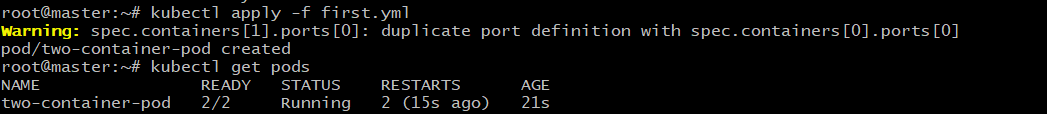




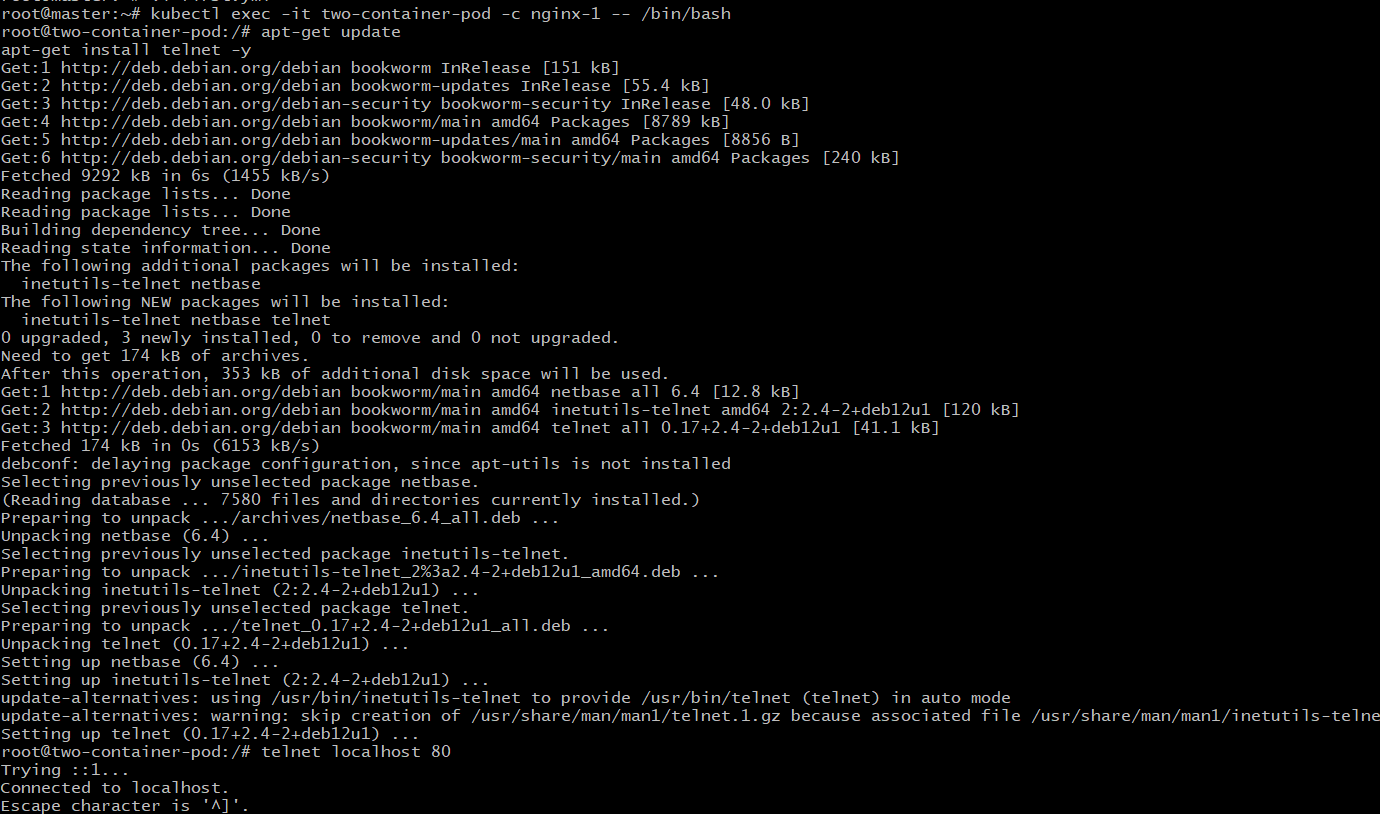


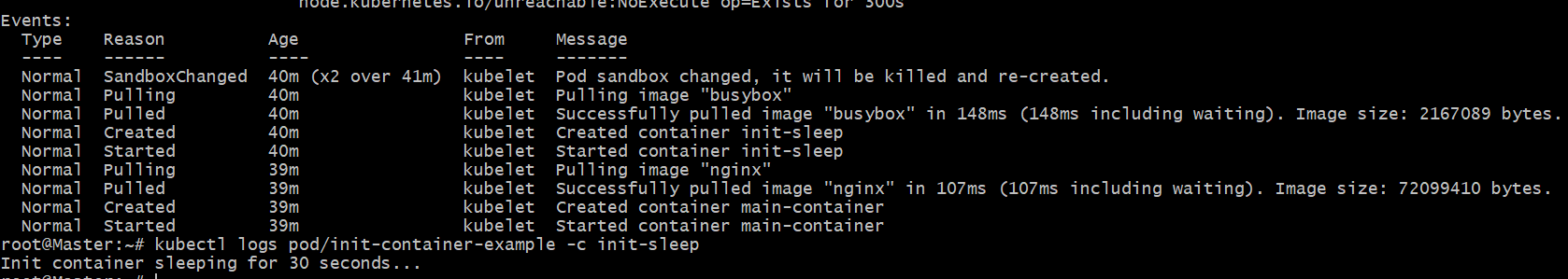
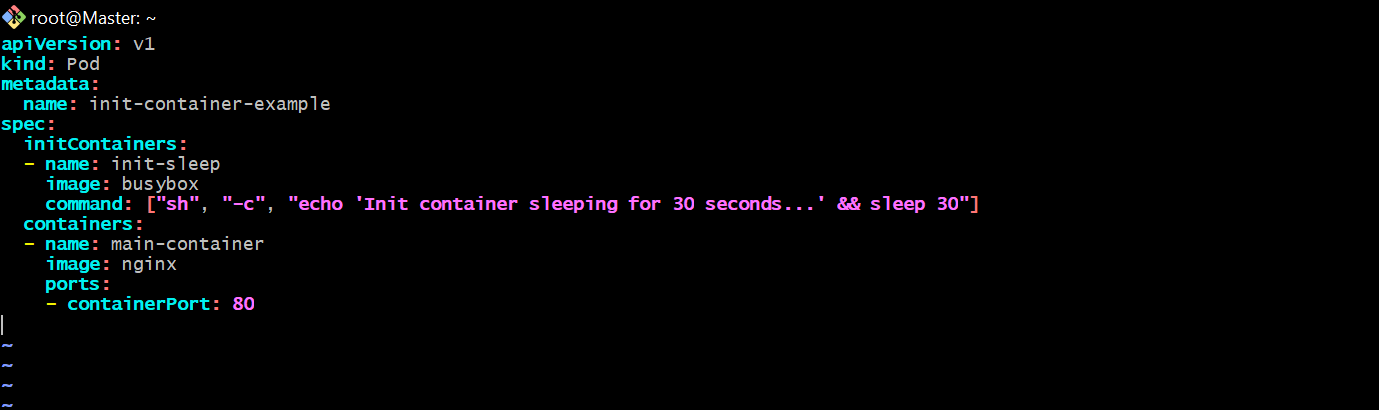
**4) Create a Pod with Two Containers Task: Create a YAML file to define a Pod with two nginx containers inside. Use kubectl exec to access both containers and verify that both containers can communicate through the same network (e.g., using telnet between them).**

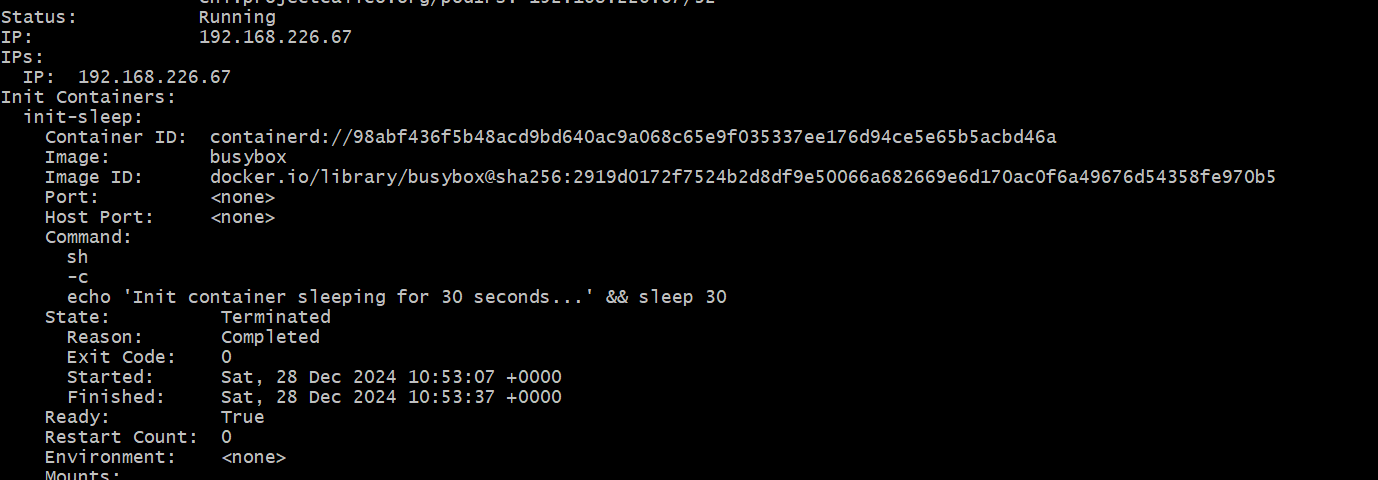




**Using kubectl exec to access both containers and verifying that both containers can communicate through the same network (e.g., using telnet between them).**



**5) Set Up an Init Container in a Pod Task: Modify the YAML to include an init container that sleeps for 30 seconds before the main containers start. Verify the init container's execution using kubectl describe pod and check the logs to confirm its completion.** 



**6) Run a Dry Run Command to Generate YAML Task: Use the kubectl run nginx --image=nginx --dry-run=client -o yaml command to generate a Pod YAML definition. Modify the generated YAML to suit specific requirements (e.g., labels or environment variables) and deploy it.**

***kubectl run nginx --image=nginx --dry-run=client -o yaml > nginx-pod.yaml***

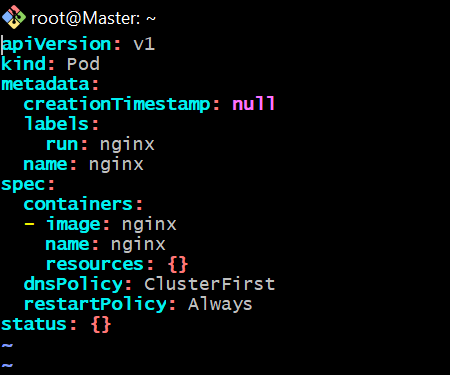
--dry-run=client: Simulates the creation of the resource without actually creating it.

-o yaml: Outputs the resource definition in YAML format.

> nginx-pod.yaml: Redirects the output to a file named nginx-pod.yaml.

Step-2[opening the ni nginx.pod for editing]

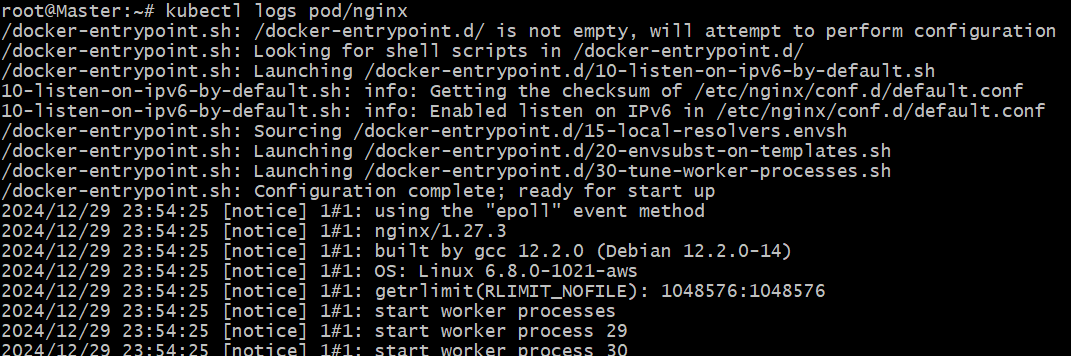
vi nginx-pod.yaml

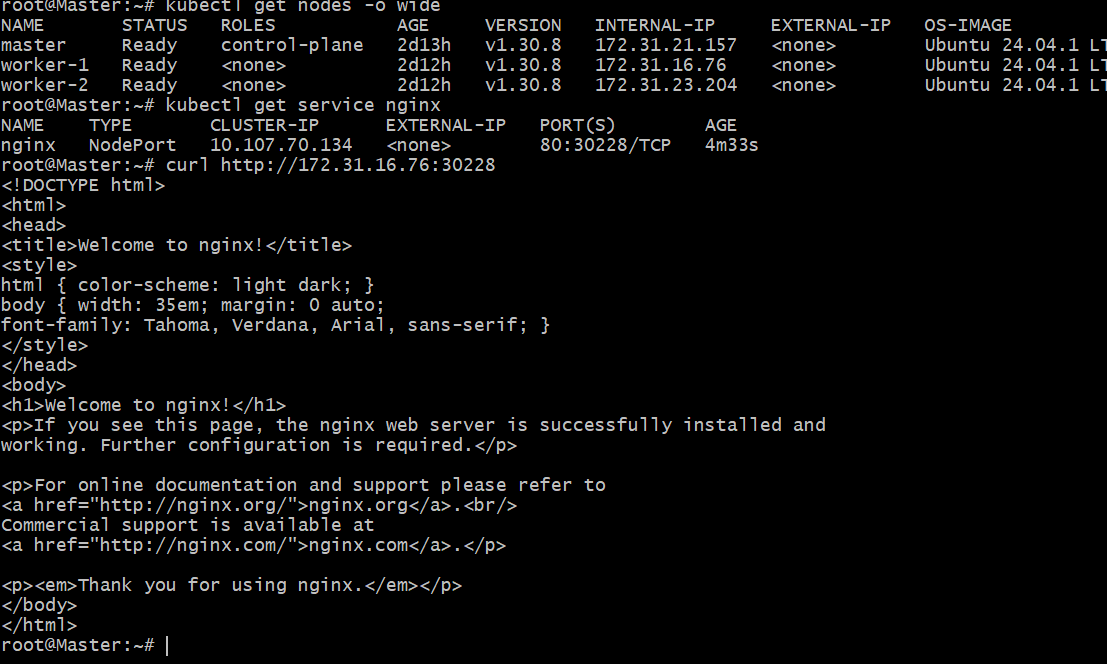


Modifying the code



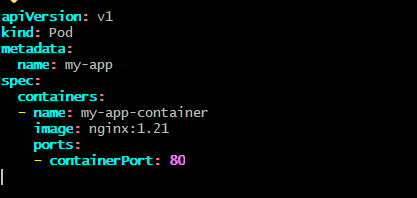
Logs

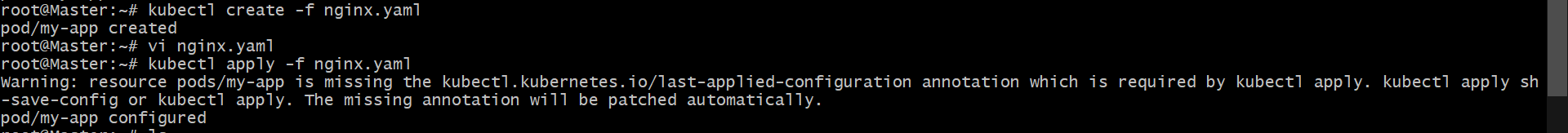




**7) Use kubectl apply vs kubectl create Task: Create a YAML file to define a Pod. First, deploy it using kubectl create -f <file\_name>.yml and then modify the YAML (e.g., change the image version). Use kubectl apply to redeploy and verify the difference between both commands.**

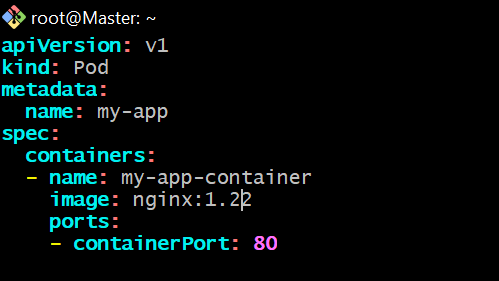
Created a yaml file with name nginx.yaml





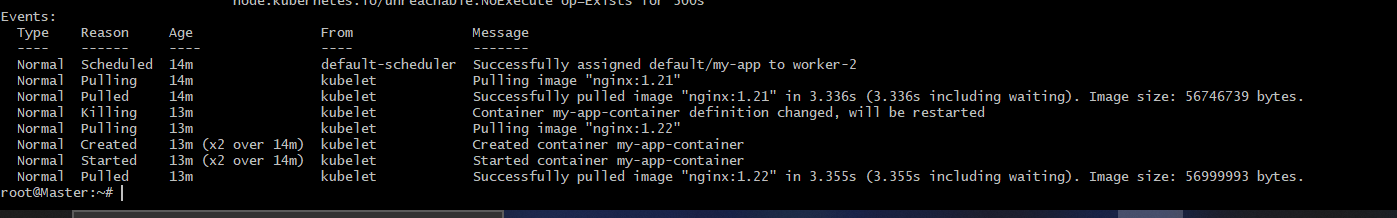
Execute the command “kubectl create nginx.pod”

Modifing file with the updated file

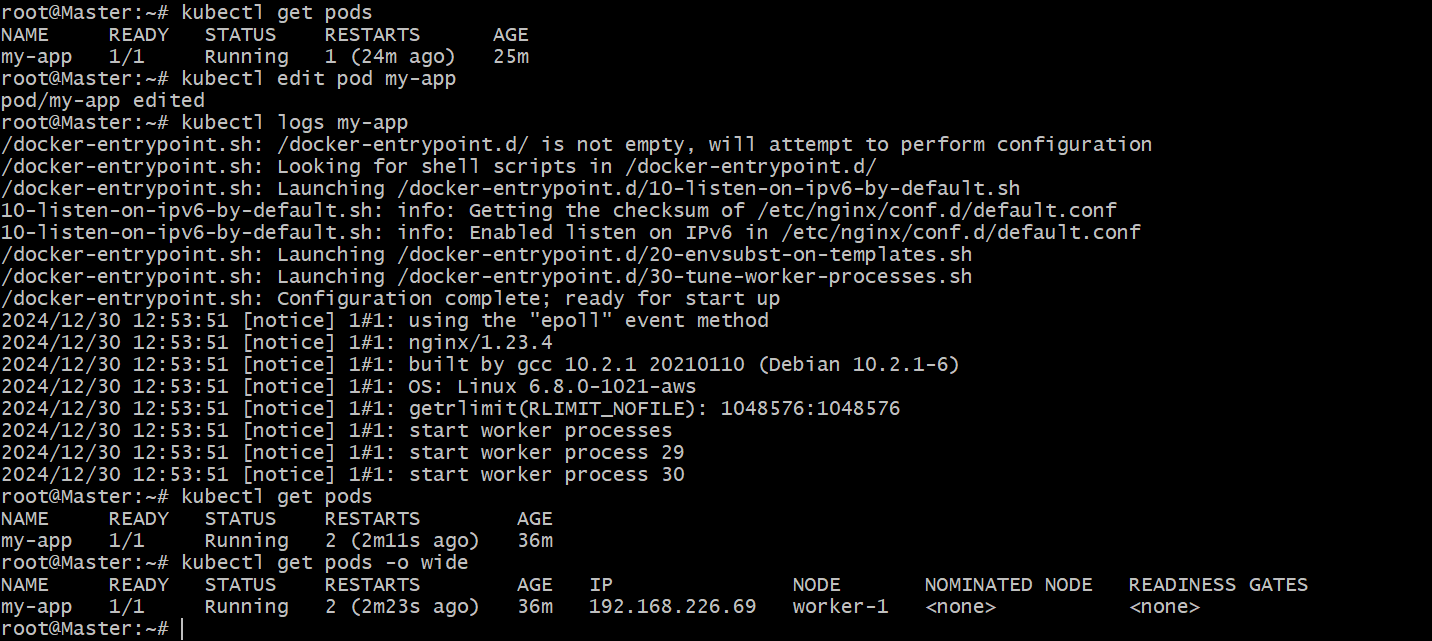


Using the command:”kubectl apply nginx.pod”

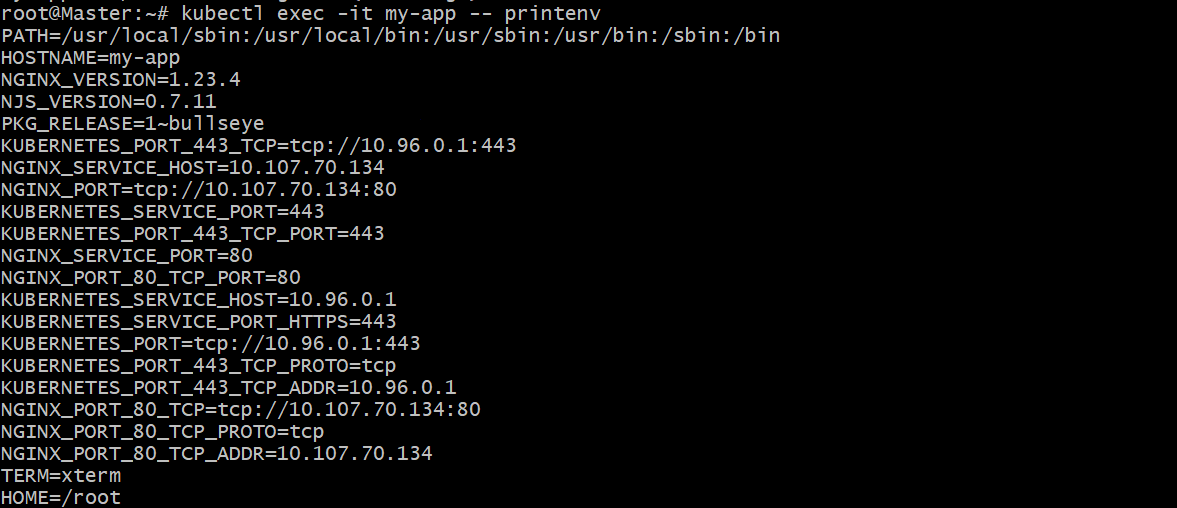
Next, “kubectl describe pod”



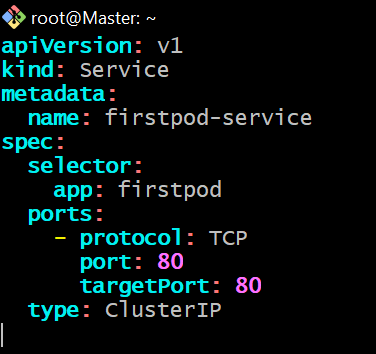
**8) Edit an Existing Pod Configuration Task: Use kubectl edit pod <pod\_name> to modify the running Pod's environment variables or image. After making the changes, verify if they took effect by** **checking the container logs or environment variables using kubectl exec.**

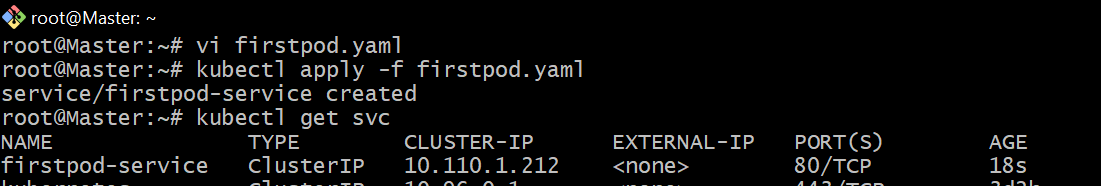


checking the container logs or environment variables using kubectl exec.

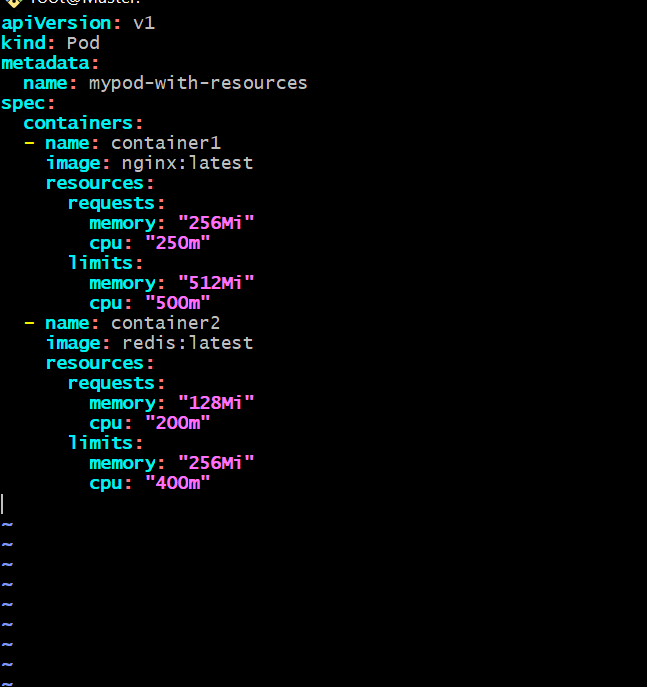


**9) Expose a Pod Using a Service Task: Create a YAML file to expose your firstpod using a Service (ClusterIP). Ensure that your service is exposing the Pod on port 80 and verify it using kubectl get svc.**





**10) Pod with Resource Limits and Requests Task: Add resource requests and limits to the containers in your YAML file. Specify CPU and memory requests/limits for both containers and deploy the Pod. Use kubectl describe pod to verify if the resource configurations are correctly applied.**



**kubectl describe pod**

