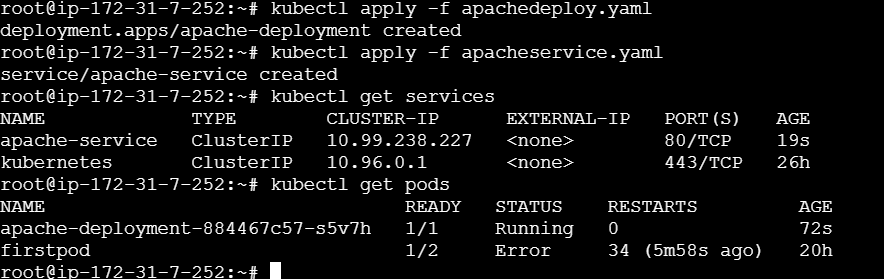
**KUBERNETES TASK 3**

1. Create a ClusterIP service for an Apache web server pod.



To apply the services or pods use this command

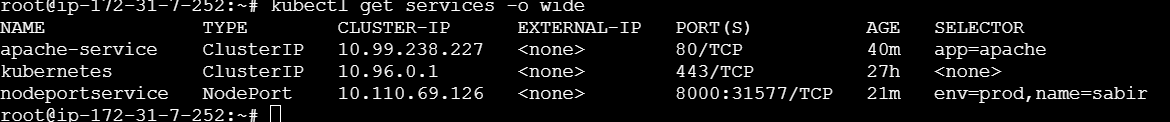
Kubectl apply -f apacheservice.yaml

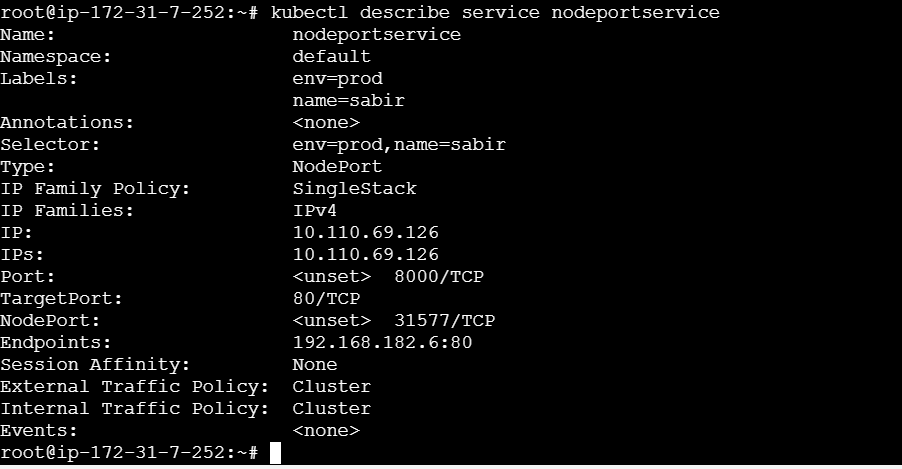


1. Expose an Nginx pod externally using a NodePort service.

To create nodePort service

kubectl expose pod firstpod --type=NodePort --port=8000 --target-port=80 --name nodeportservice

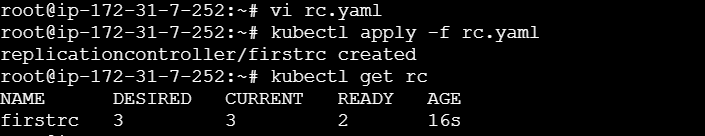




1. Deploy a ReplicationController to maintain 3 replicas of an Nginx pod.

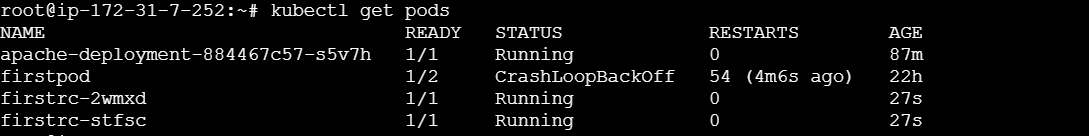
vi rc.yaml to create rc --- kubectl apply -f rc.yaml





To show all the pods

kubectl get pods



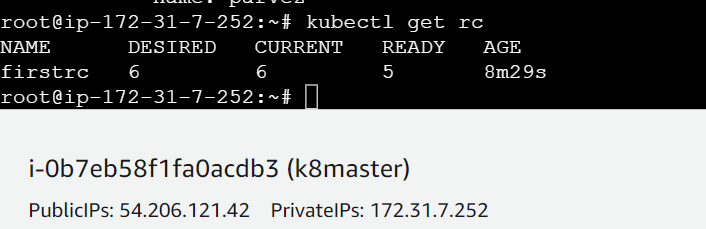
1. Scale the Replication Controller from 3 replicas to 5 replicas.

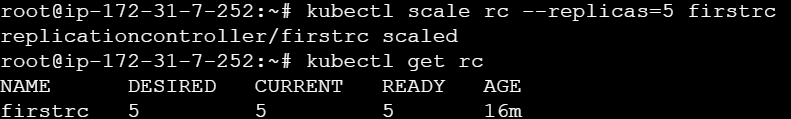
To scale this rc we need to execute the command

Kubectl scale rc –replicas=6 firstrc



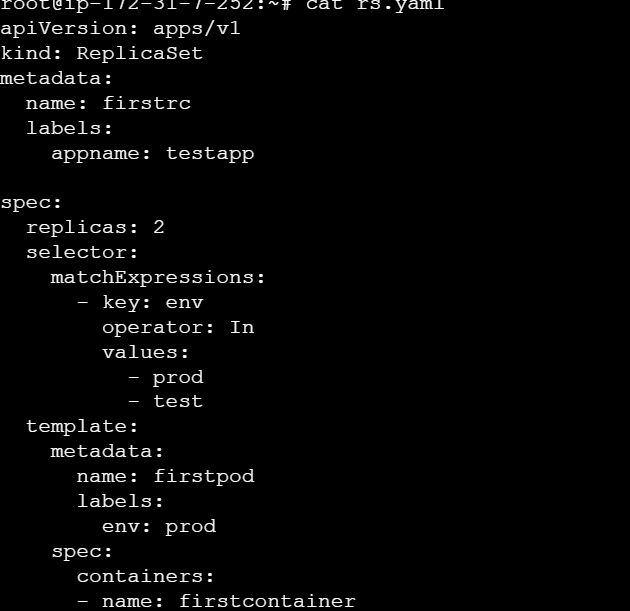
Kubectl get rc

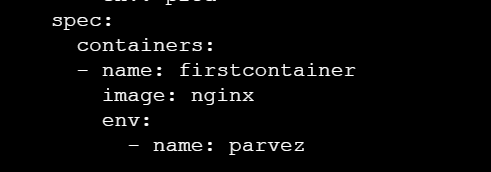




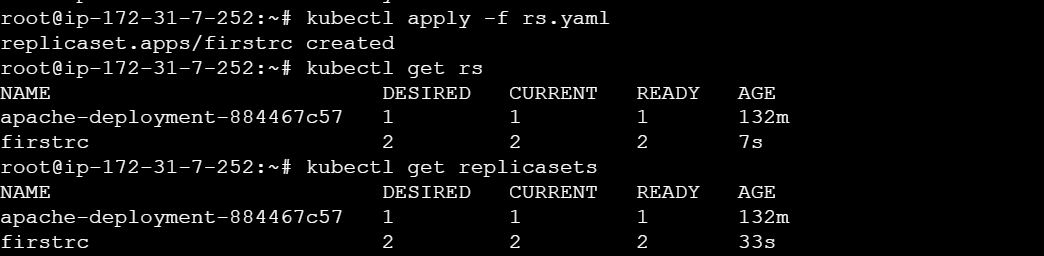
1. Create a ReplicaSet to manage pods based on multiple labels (prod and test).

Create vi rs.yaml rs replicasets



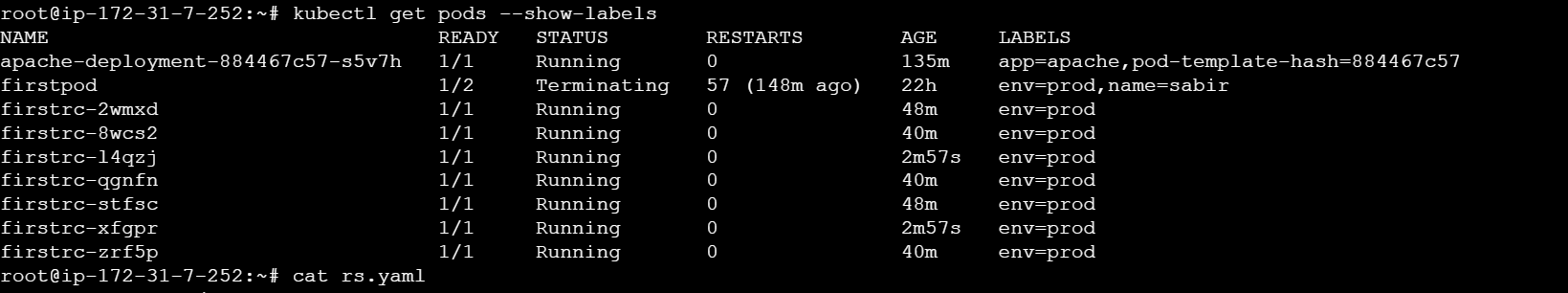


Use this commands to create and display or show the replica sets which have been created

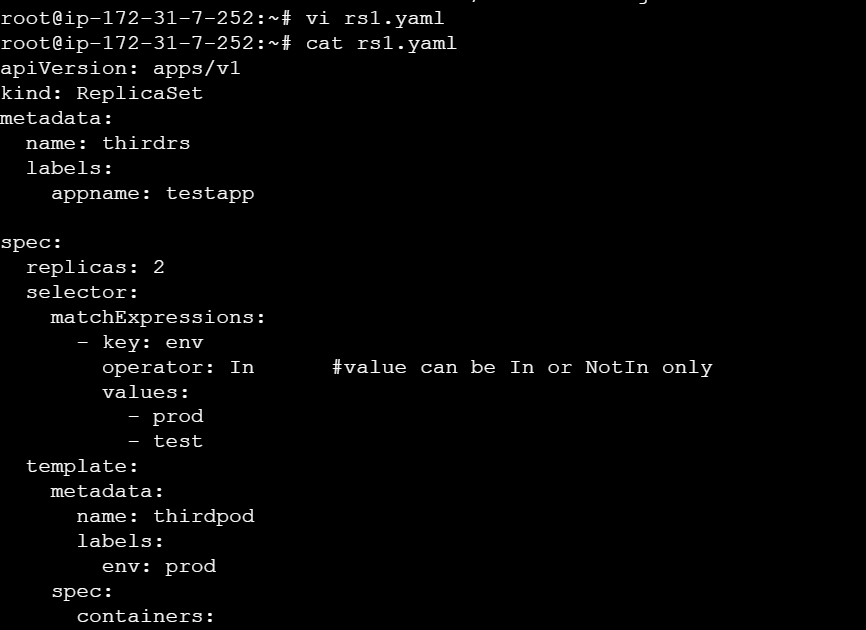


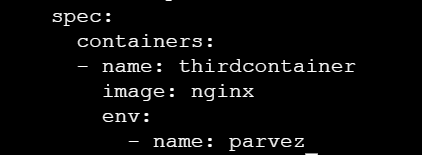
To show the labels we need to execute this command

kubectl get pods –show-labels



1. Deploy a ReplicaSet that excludes pods with the label backend.



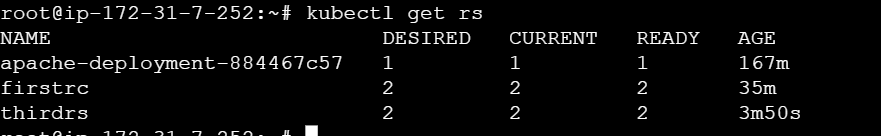


To create we need to execute this commad because it will over ride

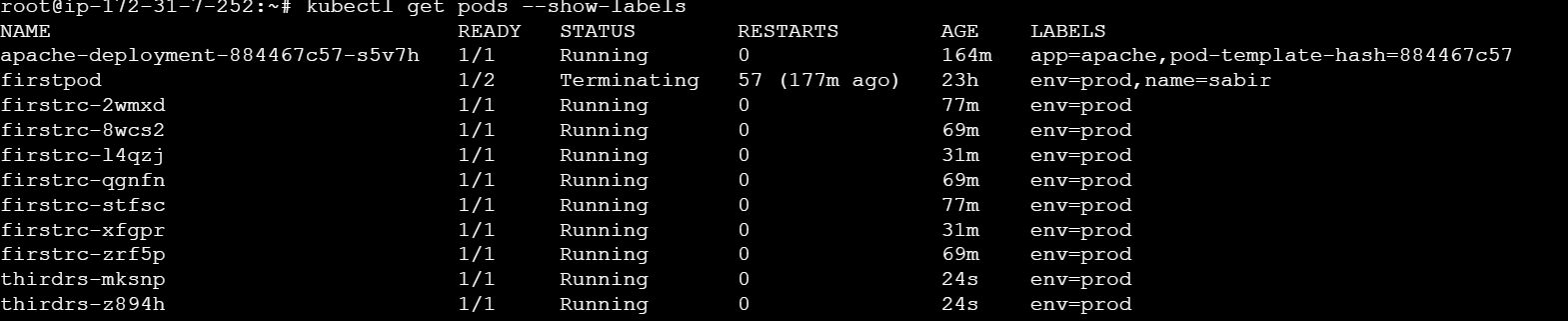
kubectl create -f rs1.yaml



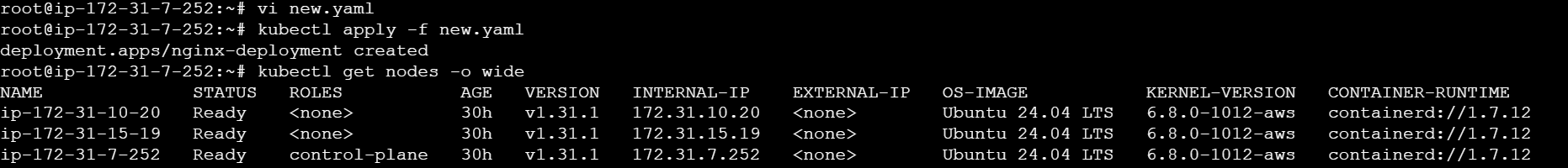
Kubectl get rs

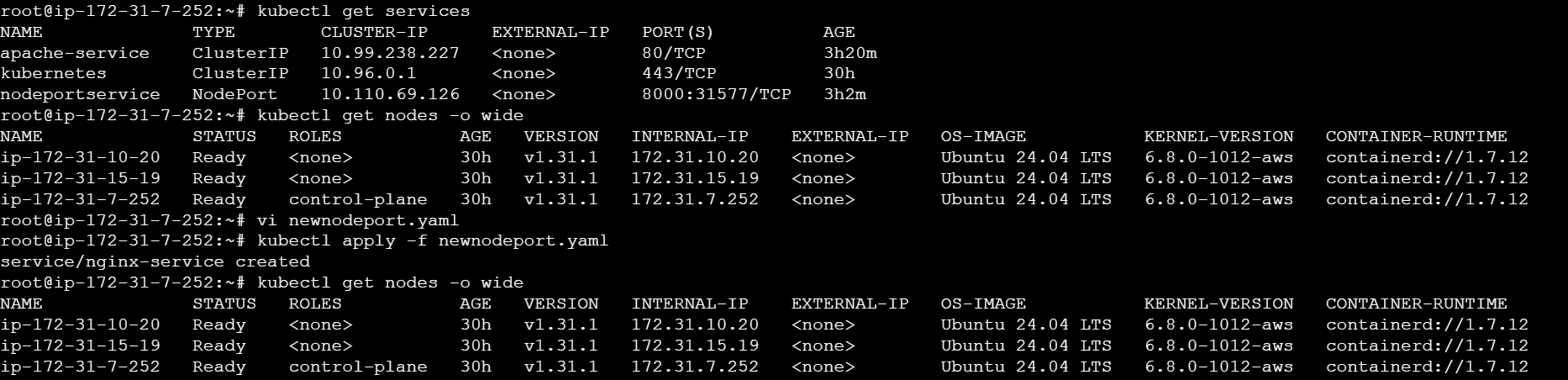


Kubectl get pods –show-labels



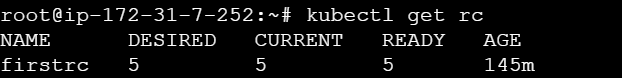
1. Test load balancing across multiple pods using a NodePort service.





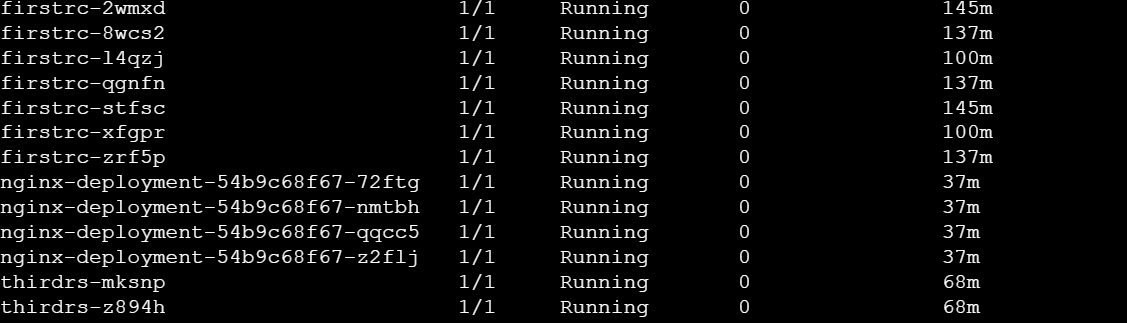
1. Delete a Replication Controller without affecting the running pods.

Before deleting we have on r c names firstrc



We have running pods

Kubectl get pods



To delete the rc we need to execute this command

Kubectl delete rc rcname –cascade=false

