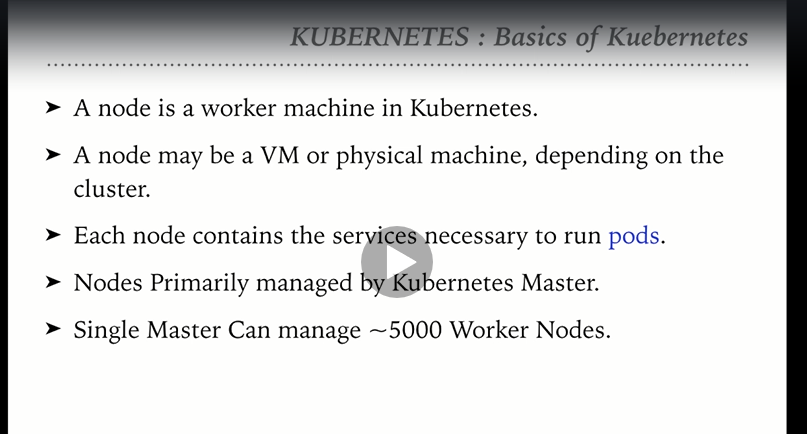
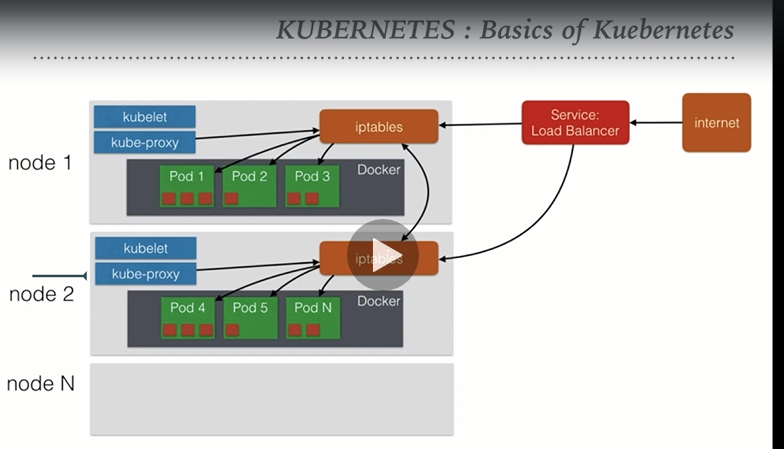
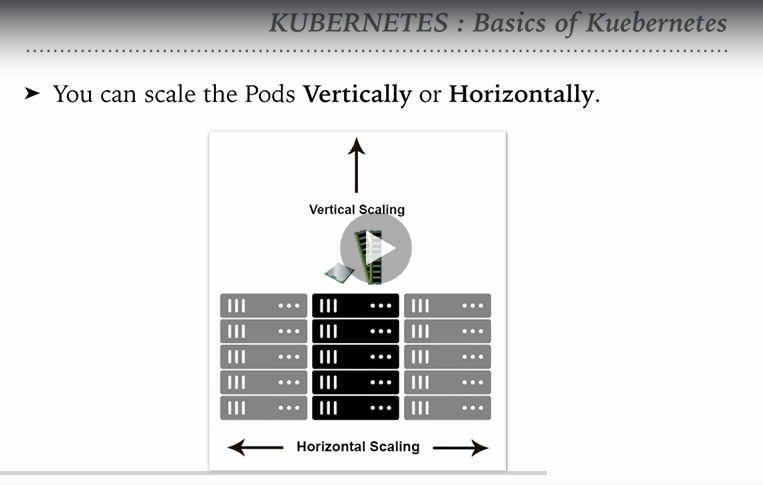
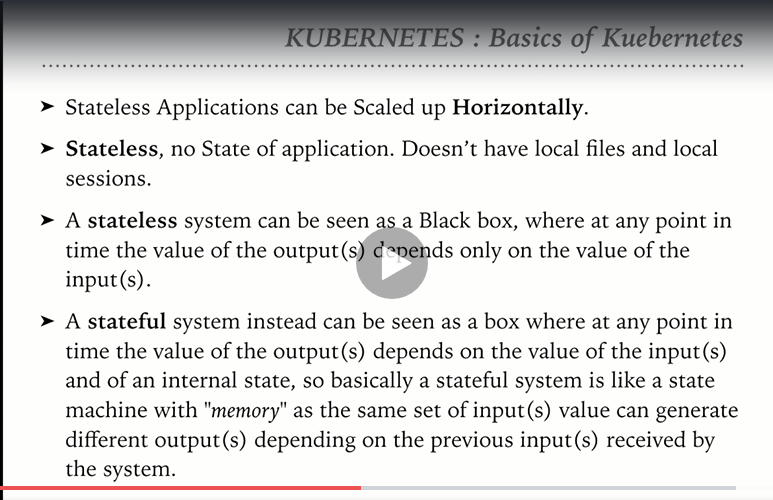
**Topic-1: Kubernetes NodeWorkflow**

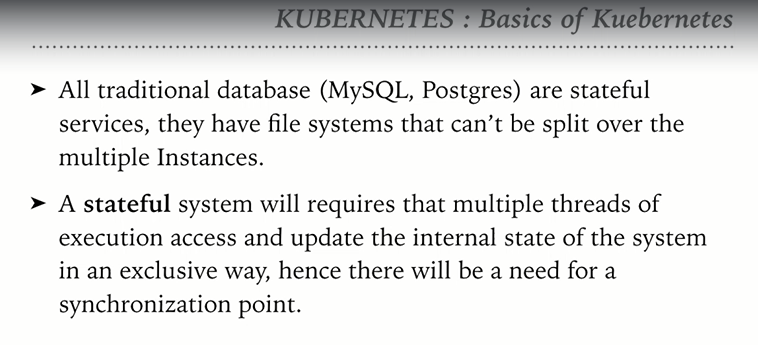


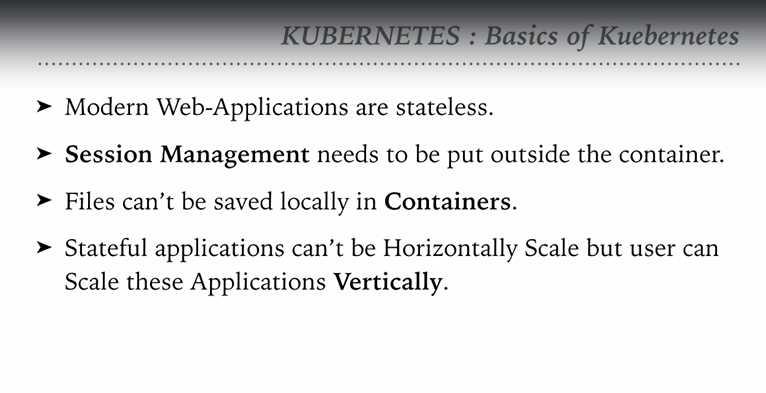
**t**

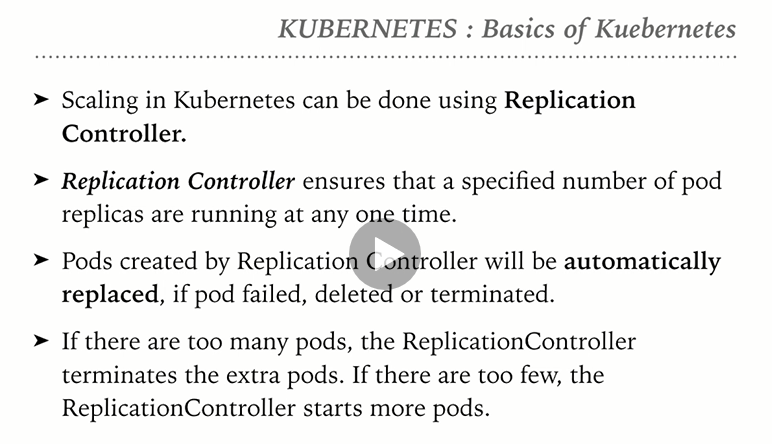
**Topic-2: Scaling PODs in Kubernetes**

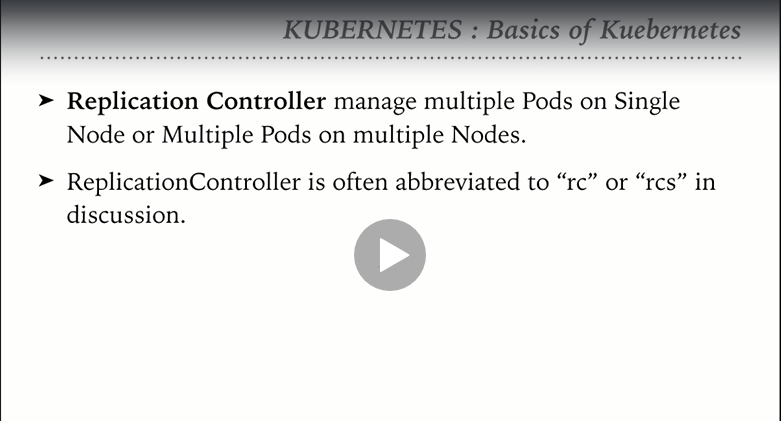




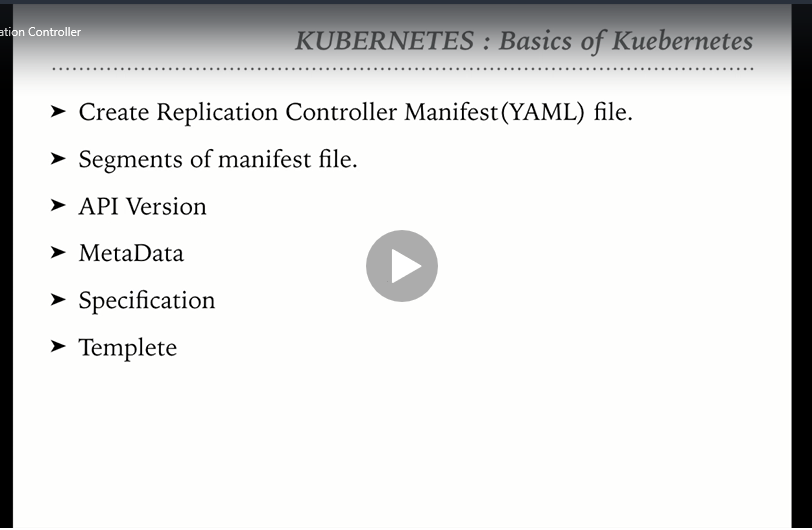


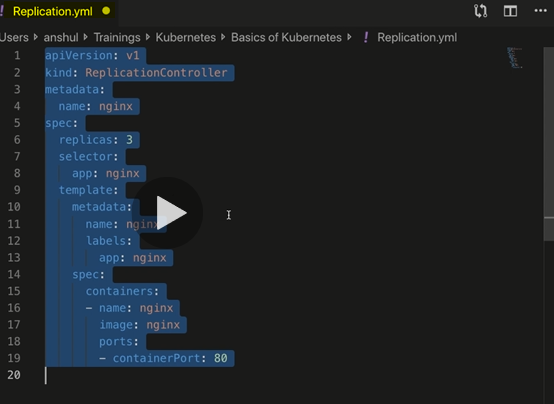




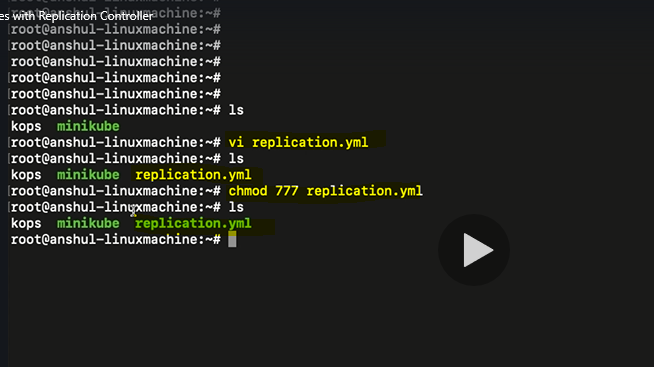


**Topic-3: Scaling pods with replication controller**

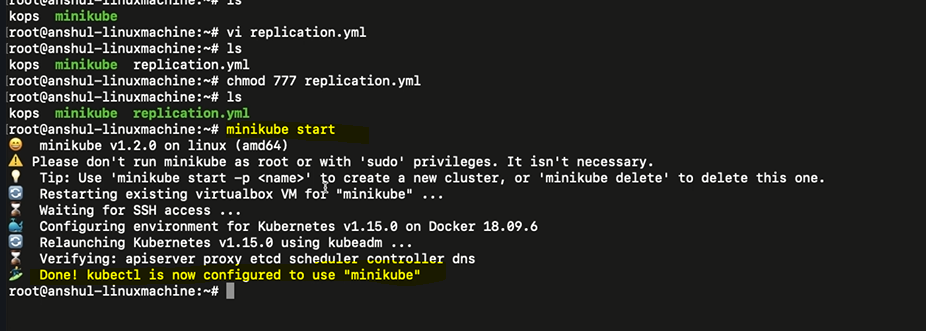


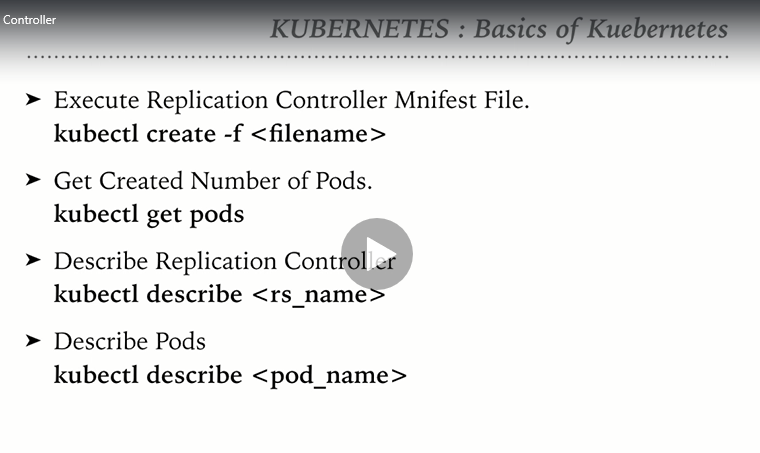


Create a Replication.yml file



Start minikube

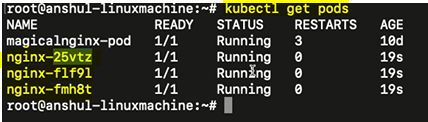




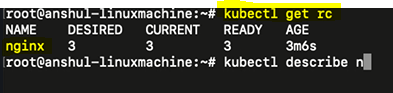
Create a replication controller



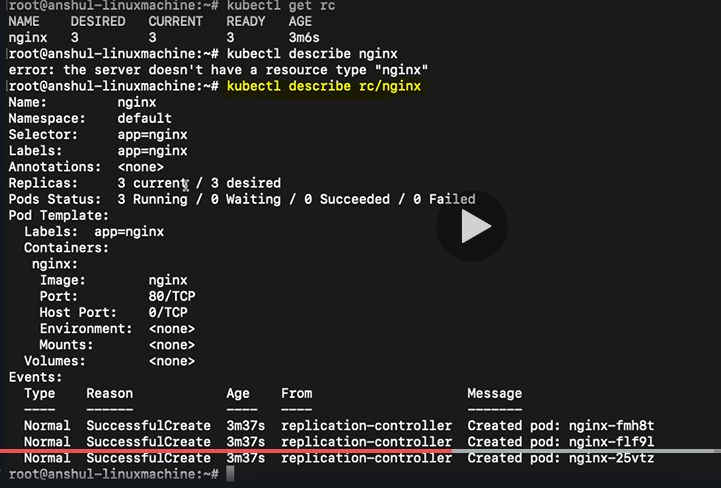
Get Pods : 3 pods were mentioned in the replication controller



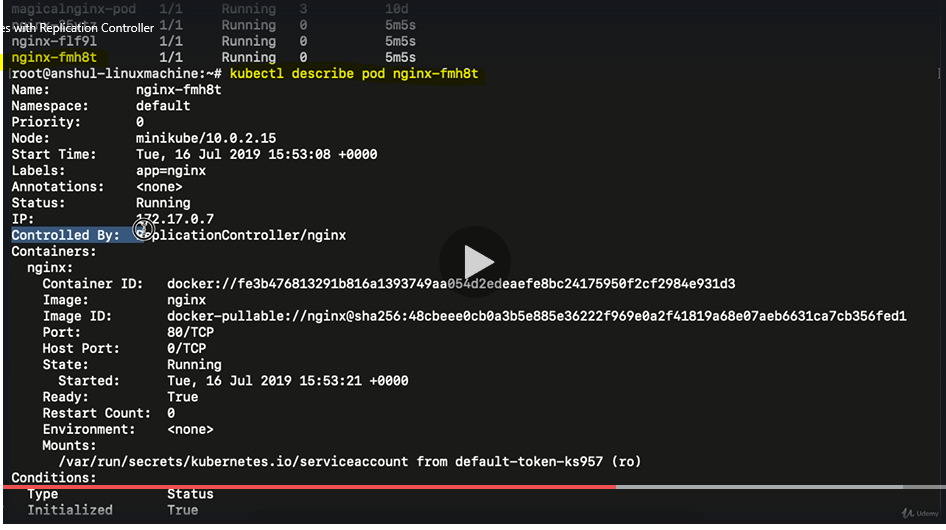
Get replication controller

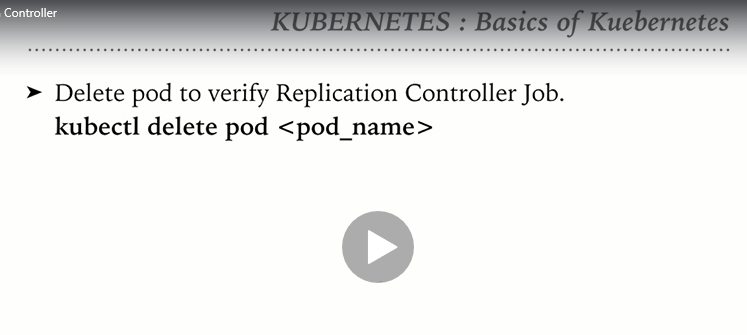


Describe replication controller

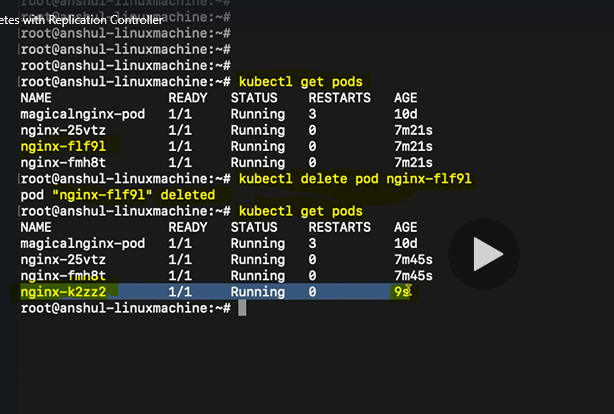


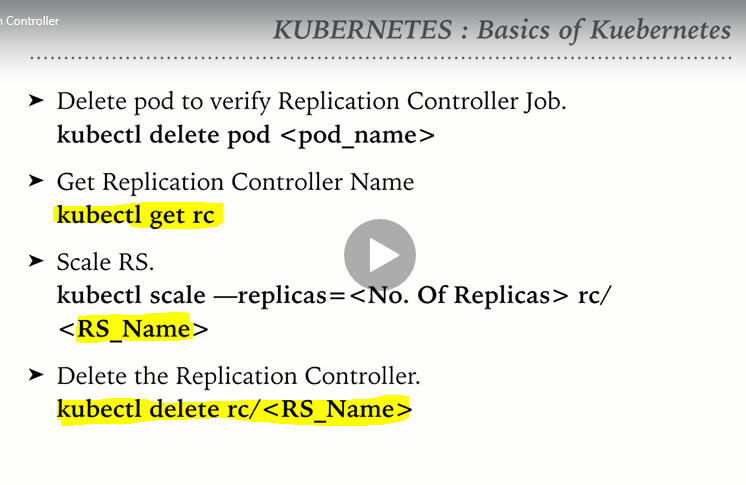
Describe pods





Delete a pod manually and verify if the Replication controller spin up a new pod

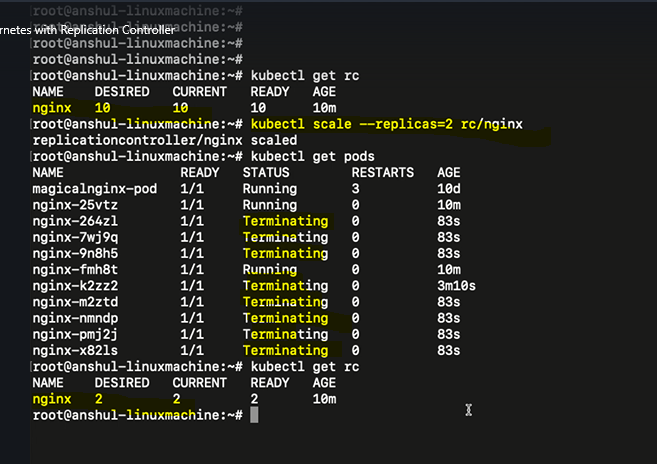




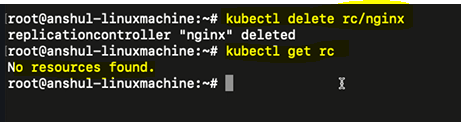
How to scale up the number of replicas in an existing RC:



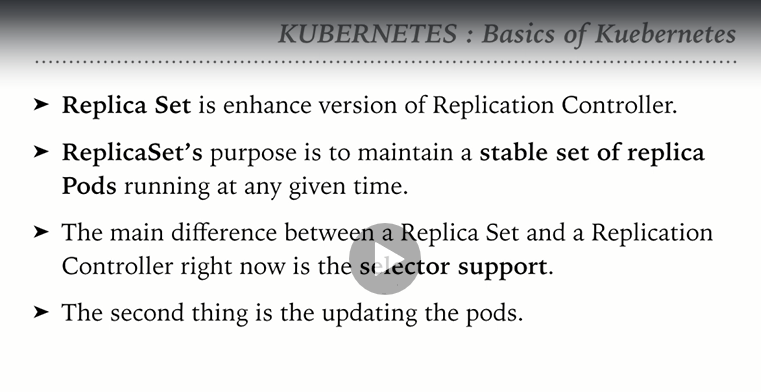
Scale down the replicas

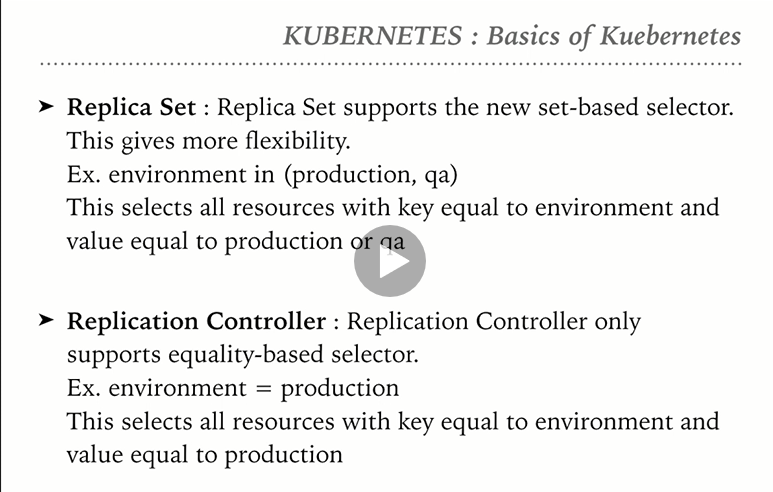


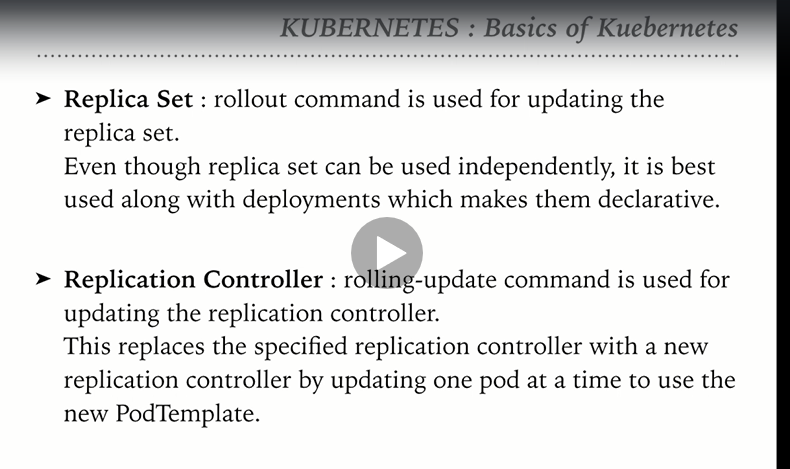
How to delete a replication controller



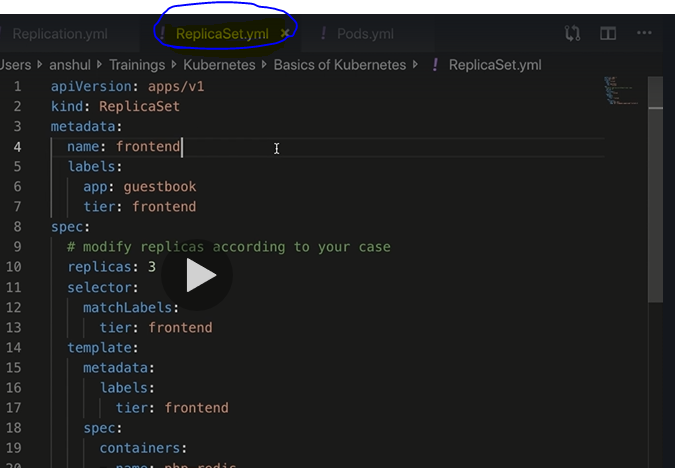
**Topic-4: Replica set in Kubernetes**







Create a Replicaset.yml file

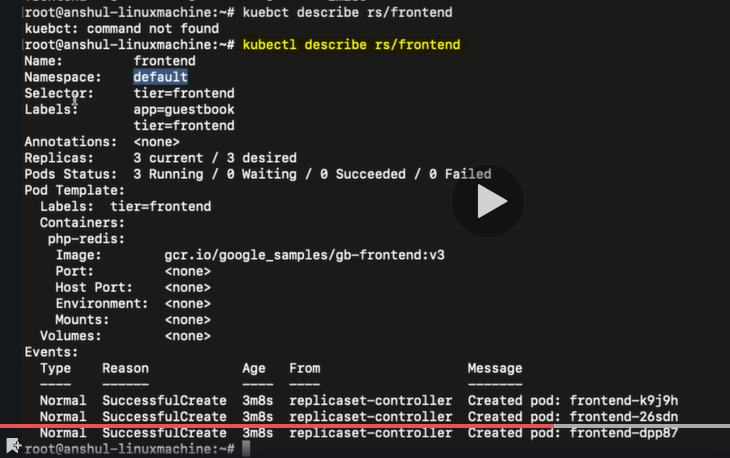


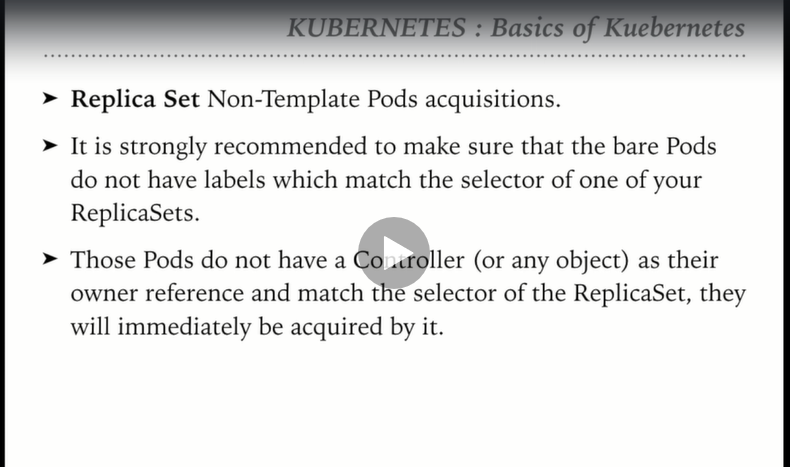
Create a replicaset and view the status:



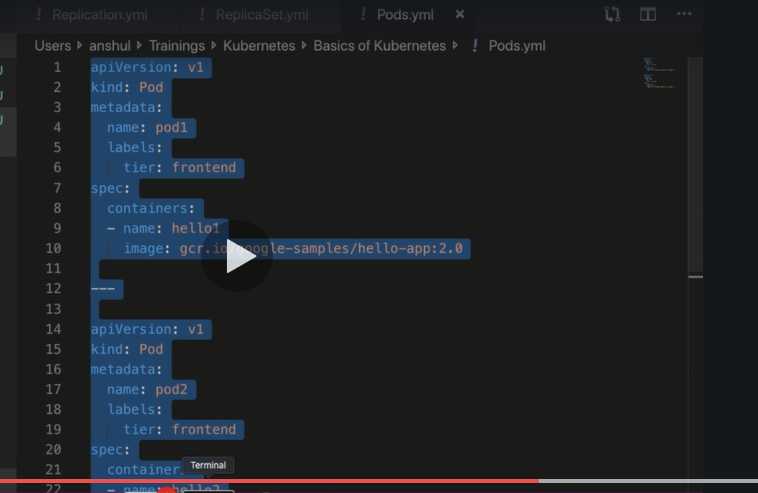
Note: Replicas refer to pods

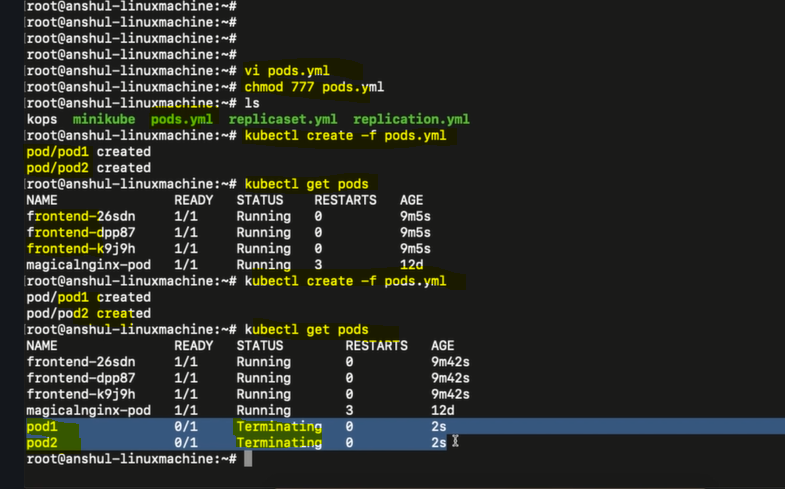
To describe replicaset



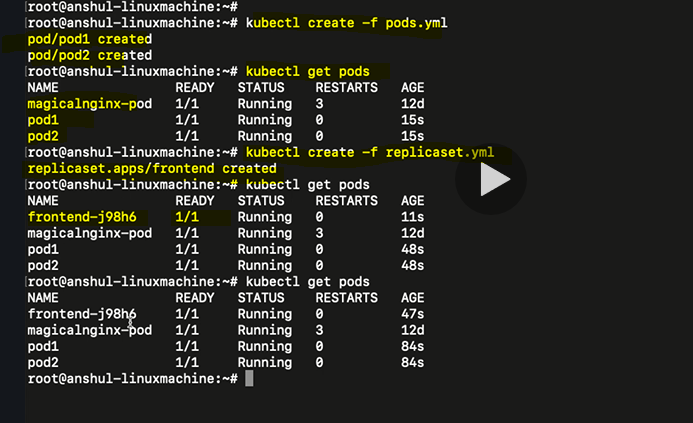


Create pod



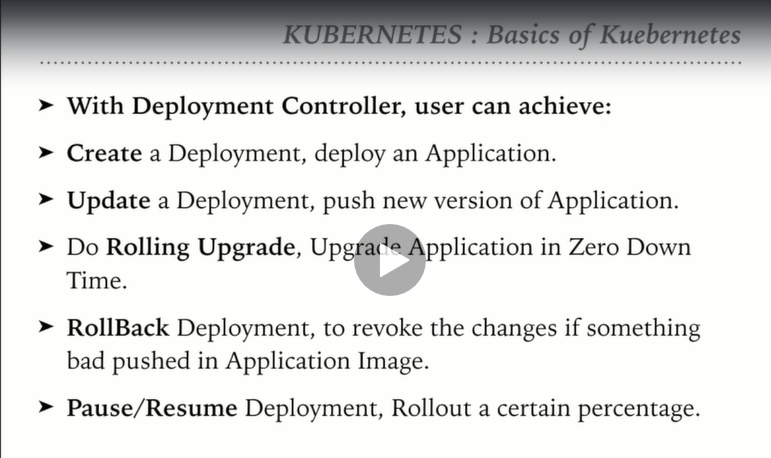
As the bare pods had the same label as that of the replica set then the bare pods are acquired by th e replica set. That is the reason, even though bear pods are cretaed, they are getting terminated

Create the bare pods first and then create the replicaset with the same label. We can onserve that only one pod of the replicas set is created as there are already 2 bare pods of the same specification are running.

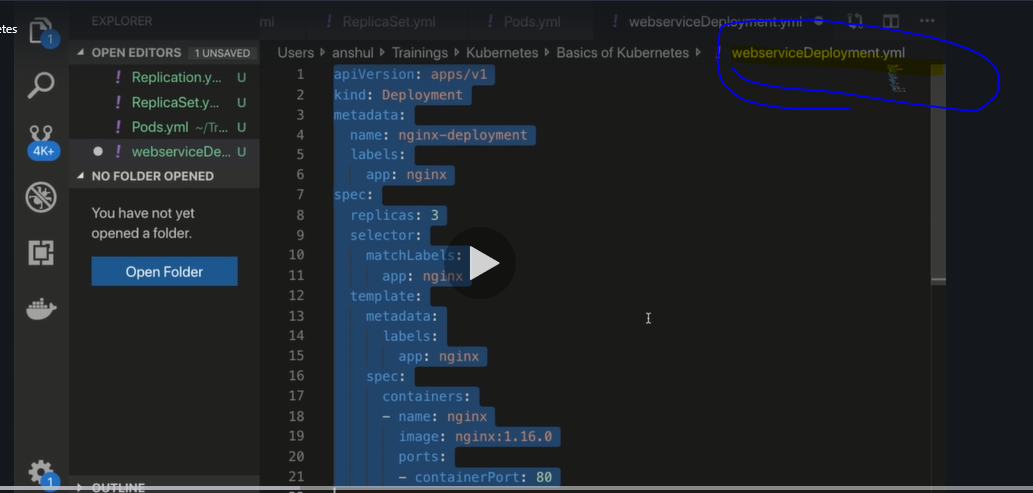


**Topic-5: Deployments in Kubernetes**



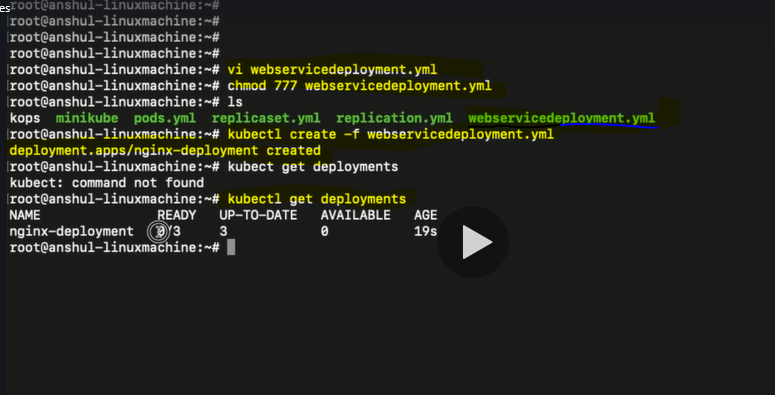


Create a Deployment manifest file i.e as shown below:

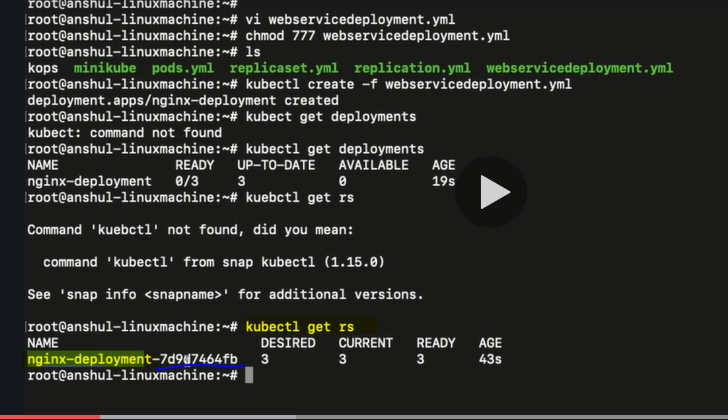




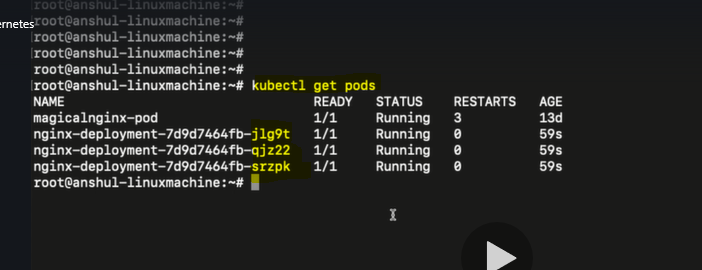
Commands to create deployment:



Verify the Replica set . Replica set is same as deployment appended with a random number which denotes the replica set



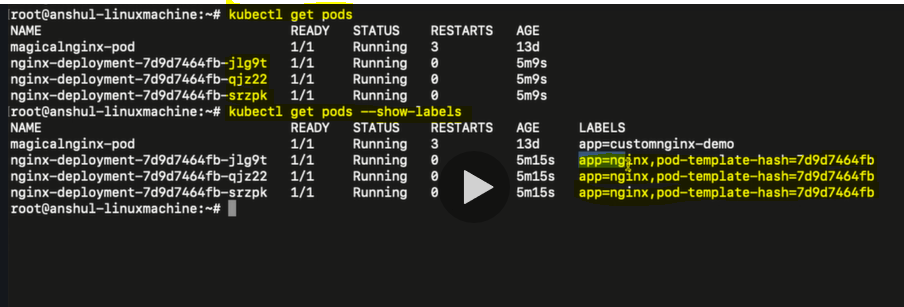
Verify the pods. Observe the names of the pods, they are basically same as replicaset appended with random number



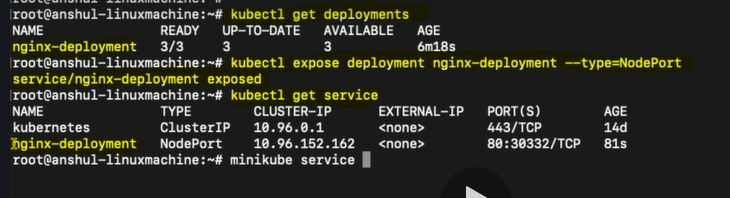
Describe the deployment



Show labels for pods



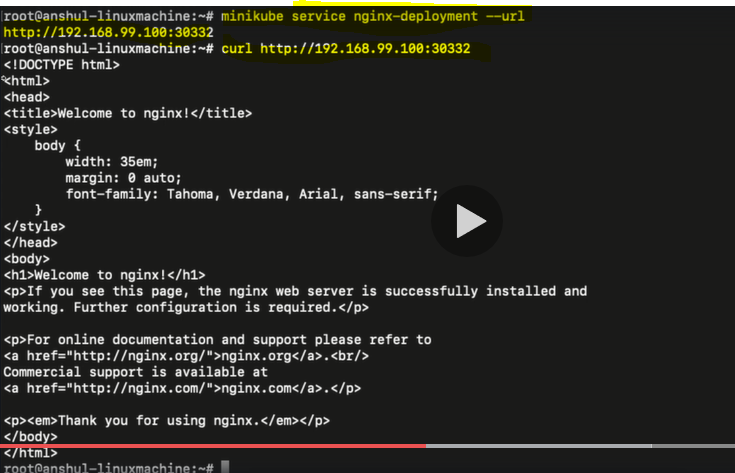
How to expose deployment . Deployment in turn creates service.



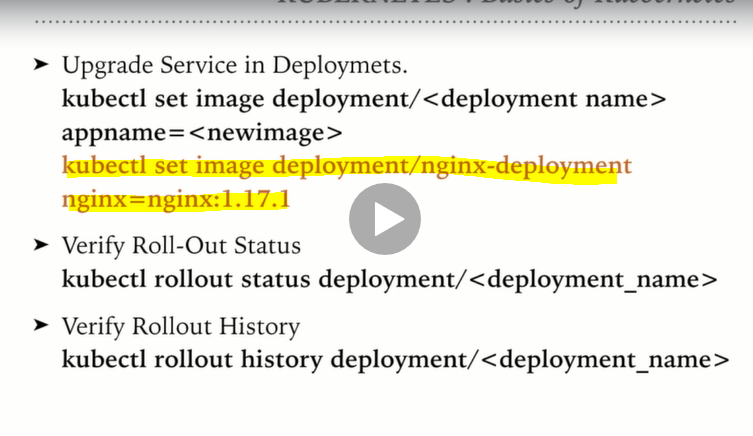
This service can be accessed either on minikube or on the AWS cluster or on the GCP

In the above screenshot, the CLUSTER-IP column is just the internal IP. We cannot access the service from internet using this IP.

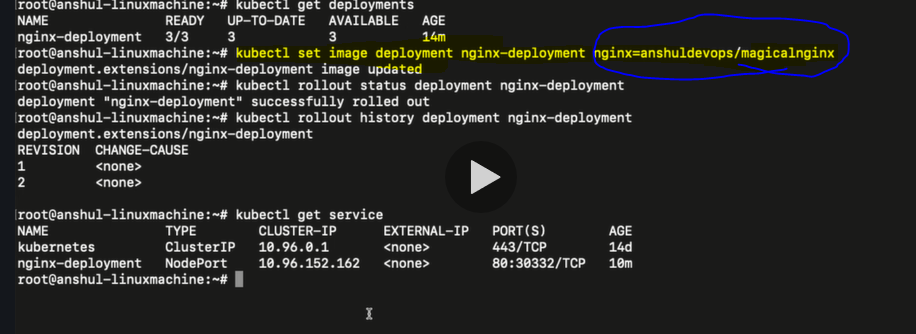
To access the service via internet, use the command highlighted below to get the IP address



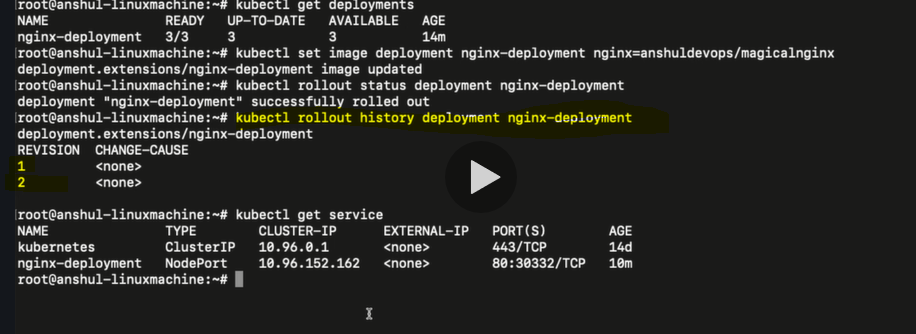
How to upgrade service in the deployment



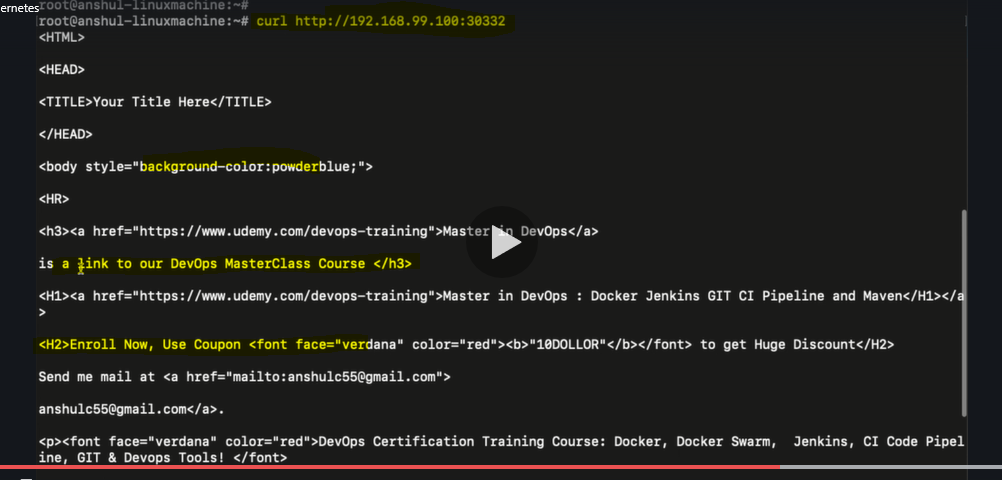
The below command is used to update the image from nginx:1.16.1 to the custom image magicalnginx



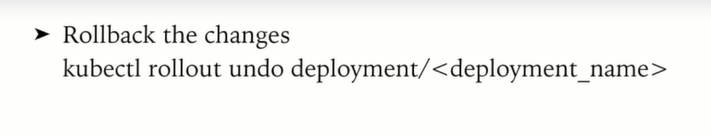
Also we can see that the history of services rolled out as explained below:

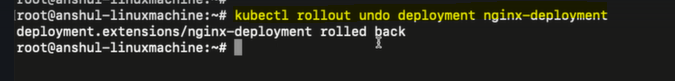


After the new image is rolled out,verify the service. The service should show the contents of magical nginx image

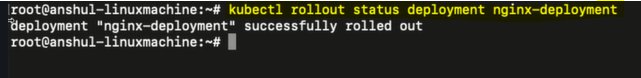


Now let us see, how to roll back the deployment to previous status

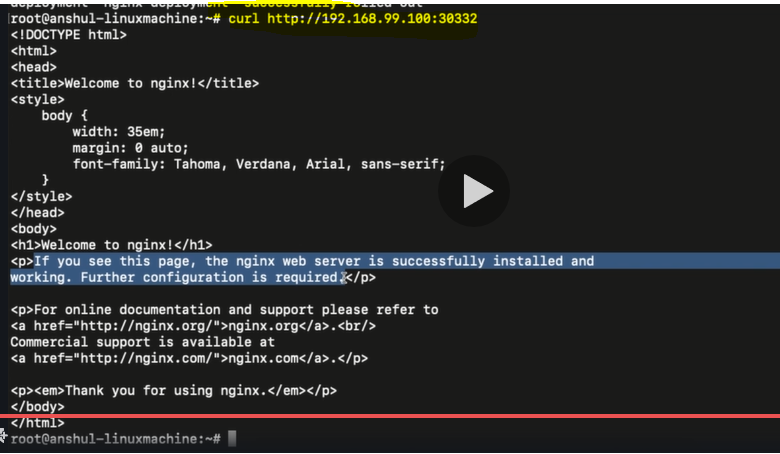




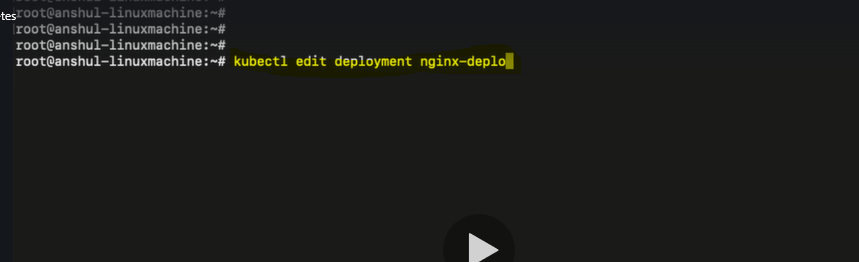
Verify if the deployment is successfully rolled back

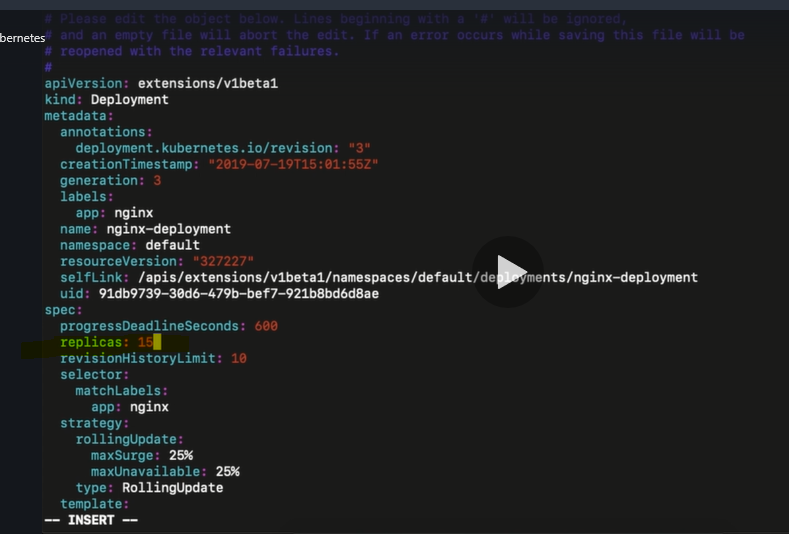


Verify that the service is rolled back to the older version of nginx image

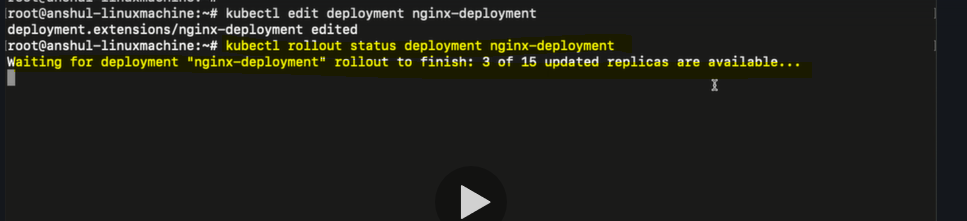


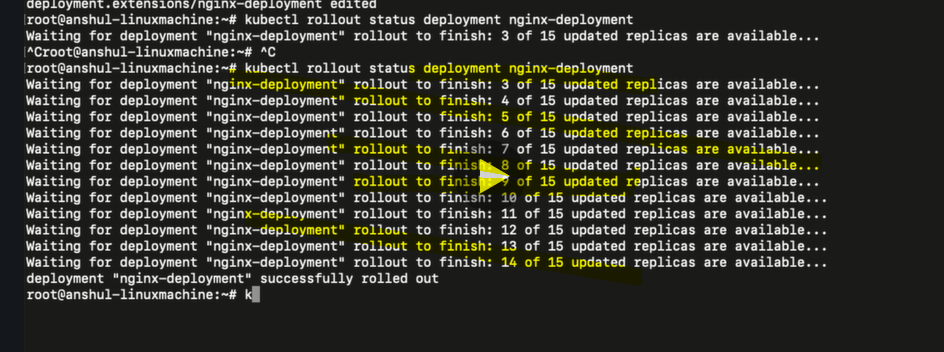
How to edit the running deployment



Edit the number of replicas from 3 to 15

Verify the status after editing the deployments





Verify if there are 15 pods created:

