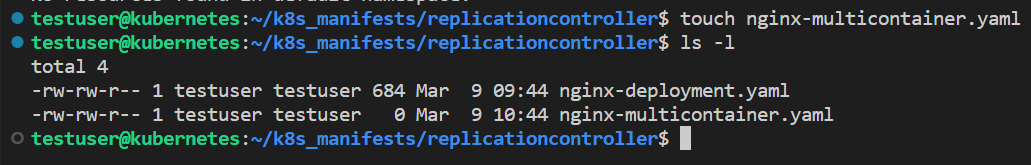
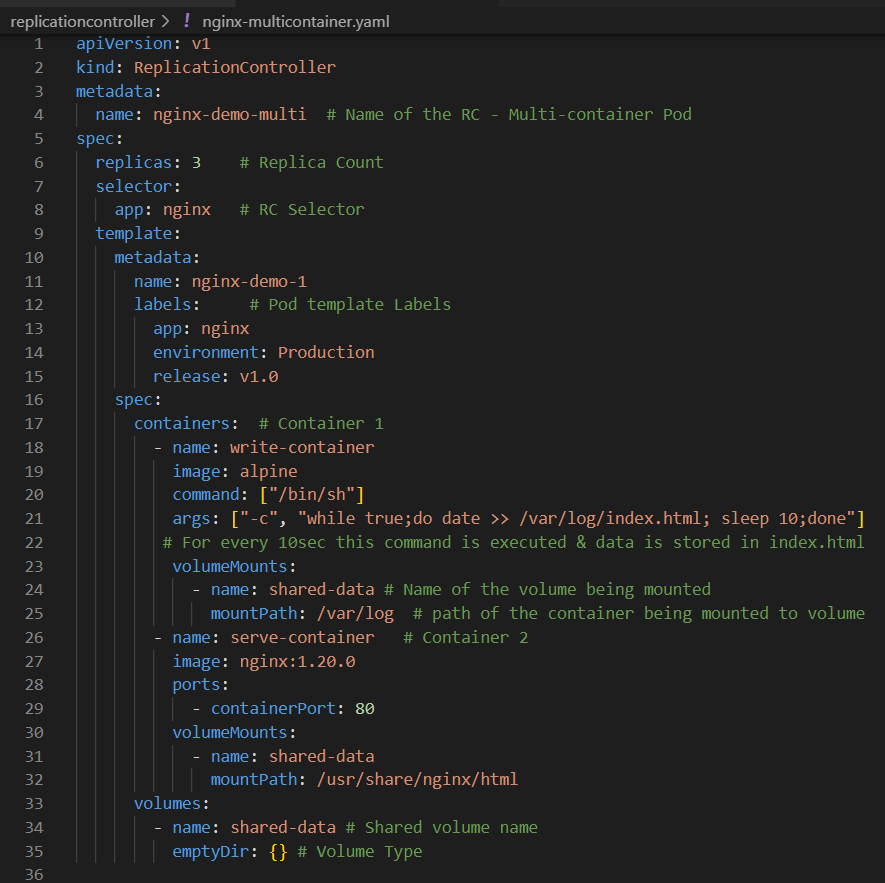
**RC – Scalein & Scaleout**

Create a new file to write a multi-container yaml file.



The RC identifies the Pods based on labels. That doesn’t mean all the labels mentioned in the Pod template metadata should be in the RC selector but atleast one should be matching.



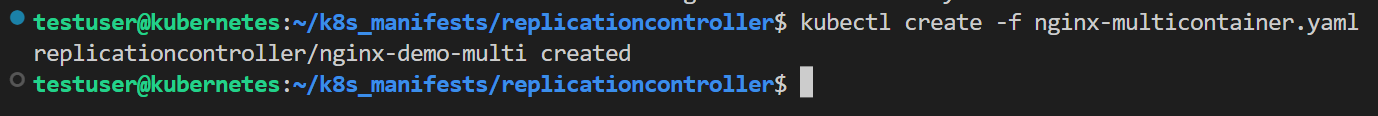
The **args: ["-c", "while true;do date >> /var/log/index.html; sleep 10;done"]** , will execute a data command for every 10sec in an infinite loop and store the logs in index.html file.

Volumemounts parameter defines the path inside the container that is being mounted to the Volume.

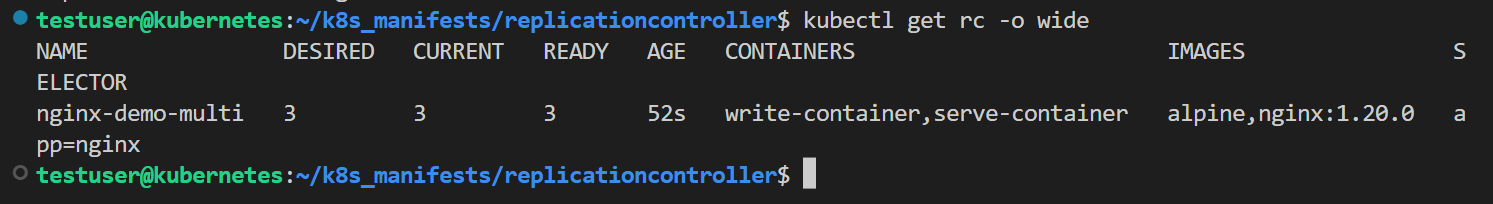
Make sure that the RC selector and the template spec labels are same. Also cross check that the volume name mentioned in both the containers is same as the volume that has to be shared.

If any mistakes are written in the yaml file. The Kube api-server doesn’t allow to create Pods because Kube-api server does Authentication, Authorization & schema validation (syntax check).

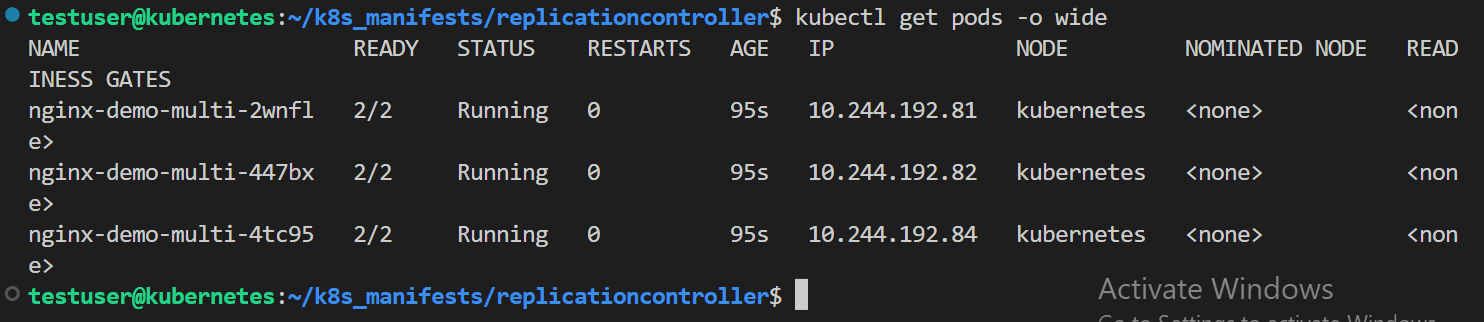
Create the RC using the written multi-container yaml file. Use the command – **kubectl create nginx-multicontainer.yaml** .



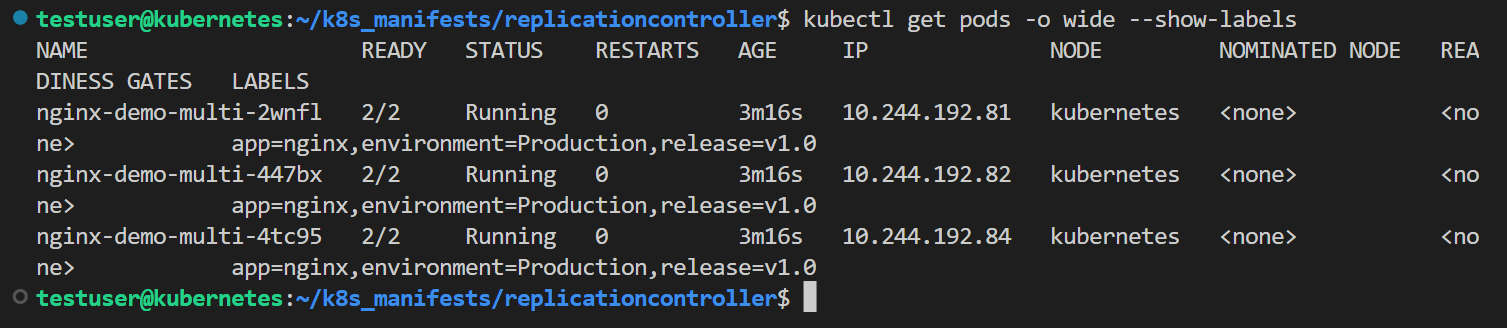
Check the status of the RC once created. Use the command – **kubectl get rc -o wide** . This will display all the metadata – names of the containers, images used for containers, selectors etc.



Also check the pods running status. Use the command – **kubectl get pods -o wide**.



To see the labels of the Pod use the command – **kubectl get pods -o wide** --**show-labels .** The labels of the pod should be matching with the selector of the RC. If they aren’t matched the RC can’t control the created Pods.

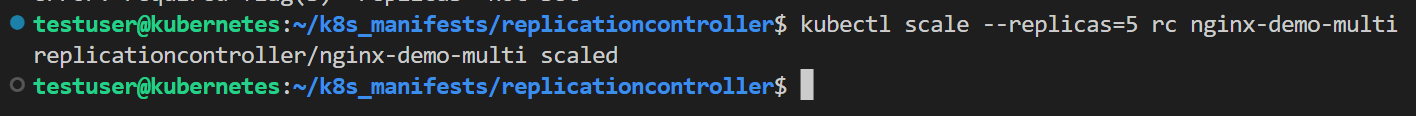


**Scalein & Scaleout concept**

Assume there are 3 Pods created by RC on which an application is running, but due to heavy load on the application, the available 3 pods aren’t able to handle the request. Here comes the necessity to increase the number of application pods availability i.e., the replica count should be increased. The scaleout concept has to be applied.

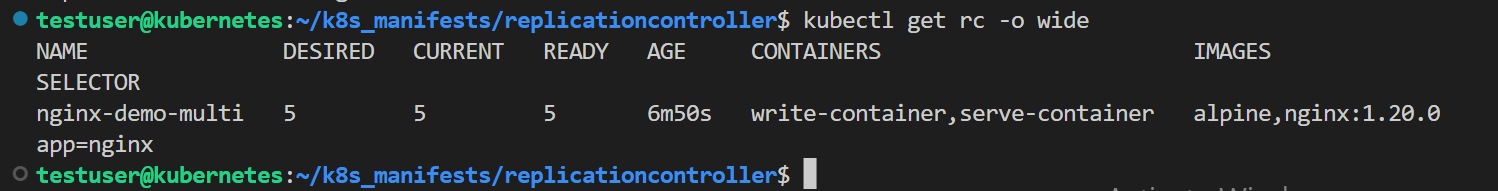
The scalein and scaleout concept can be applied to Pods in both imperative approach and declarative approach.

Let’s first use the imperative approach, the command is **kubectl scale** --**replicas=5 rc nginx-demo-multi** .

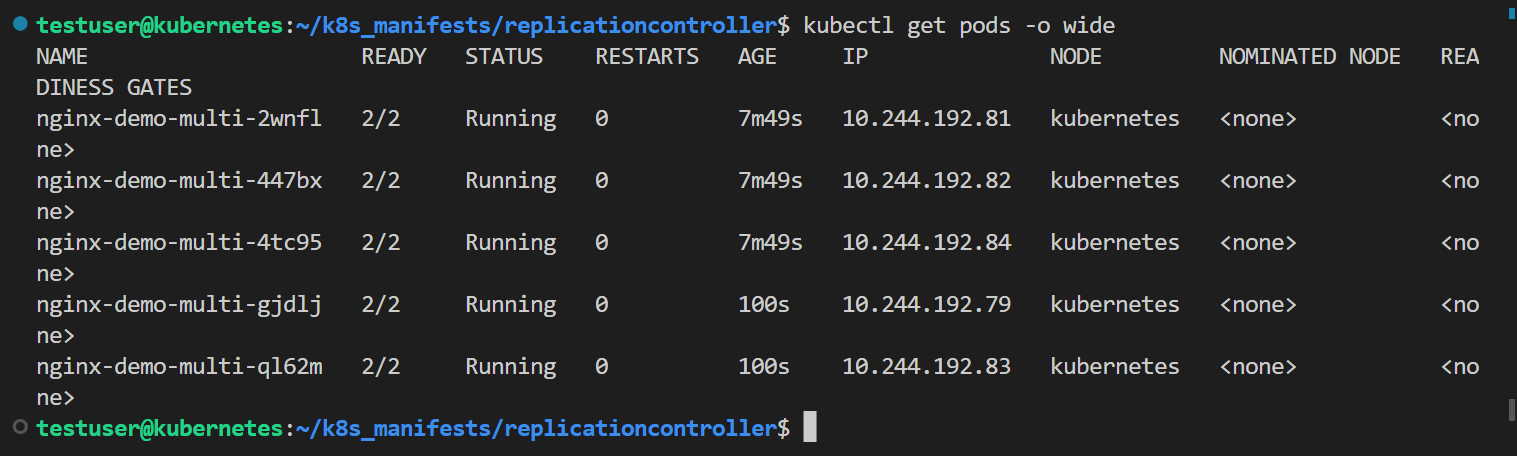


Note : when replicas=5 is given, the desired state is 5. Previously, there were 3 pods created from the yaml file, as now the desired state is 5 another 2 pods will be created not 5 new pods.

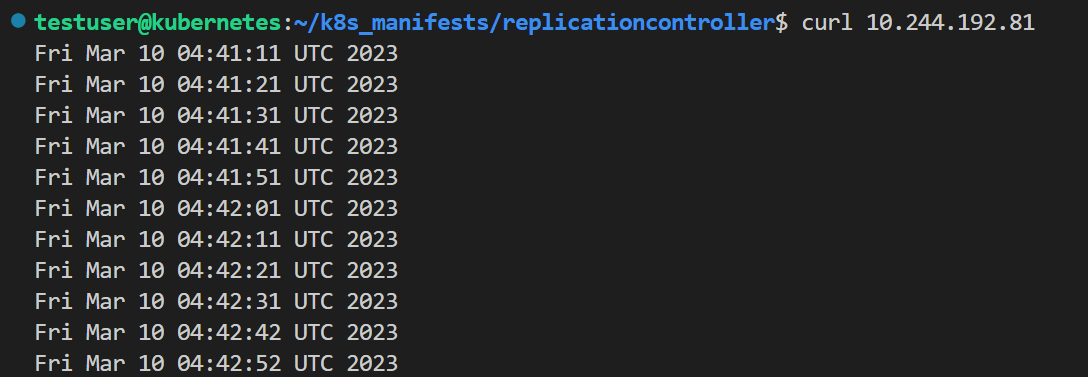
Check if the RC desired state changed to 5 or not. Use the command – **kubectl get rc -o wide**



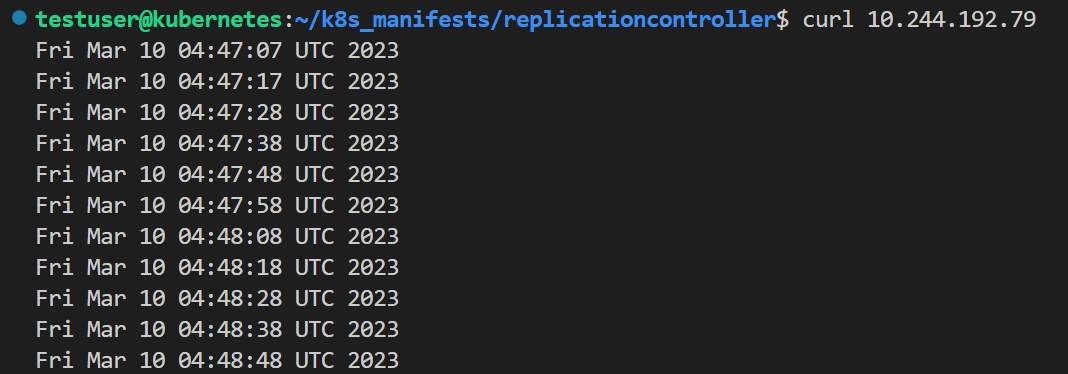
Also check the pods running status. Use the command - **kubectl get pods -o wide .** Every Pod has a unique IP address.



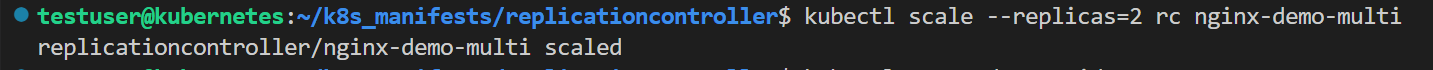
Check the contents of the application using the IP address. Use the command – **curl <IP of a POD>.** As we have used two containers (write-container & server-container) to which a volume is mounted. The contents displayed will be logs/ date commands running in the infinite while not the nginx default html page.



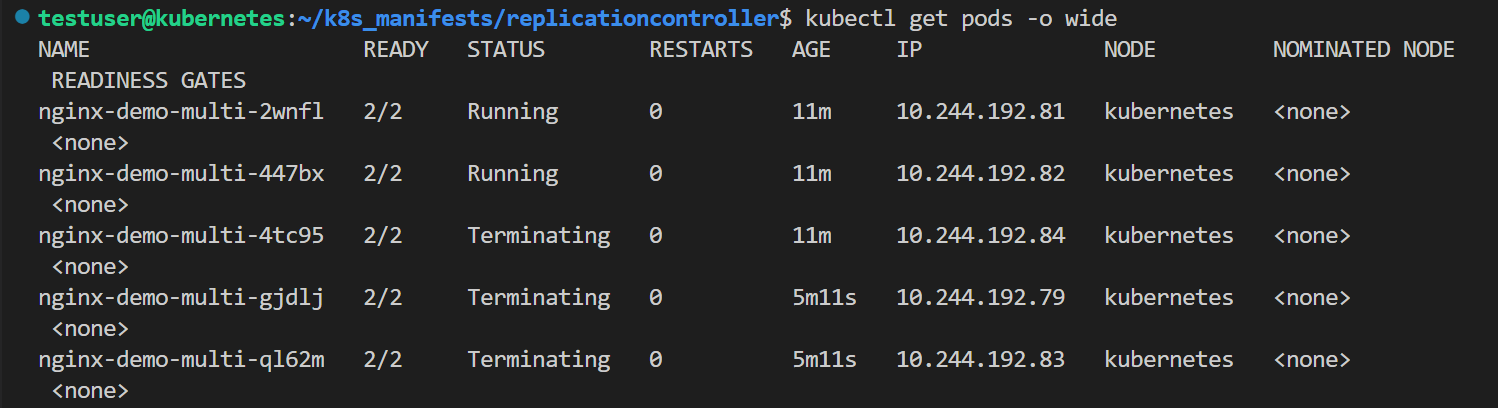
If you curl any Pod’s IP address all will show the same content – **curl <IP of POD2>**



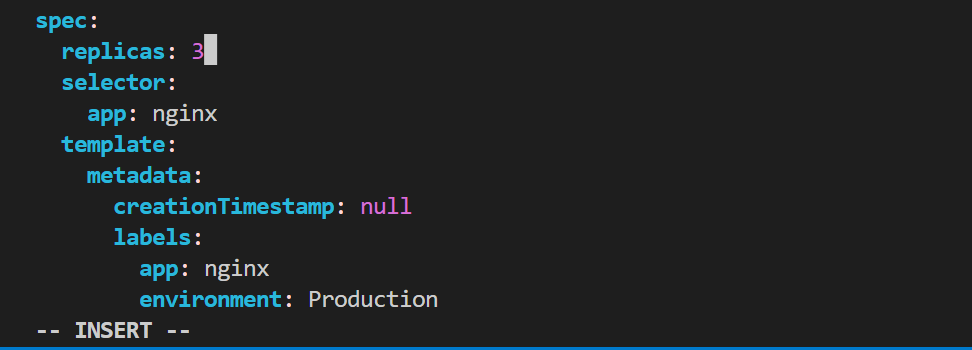
If the load on the application is decreases, the scalein also can be done. Use the command –   
**kubectl scale** --**replicas=2 rc nginx-demo-multi**

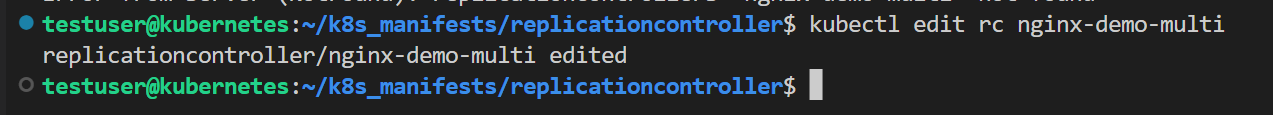


Use command – **kubectl get pods -o wide** to check what are the pods that are being terminated. The Pods will be deleted base on age (latest) and the number of pods running on that node (incase of multi-node cluster).

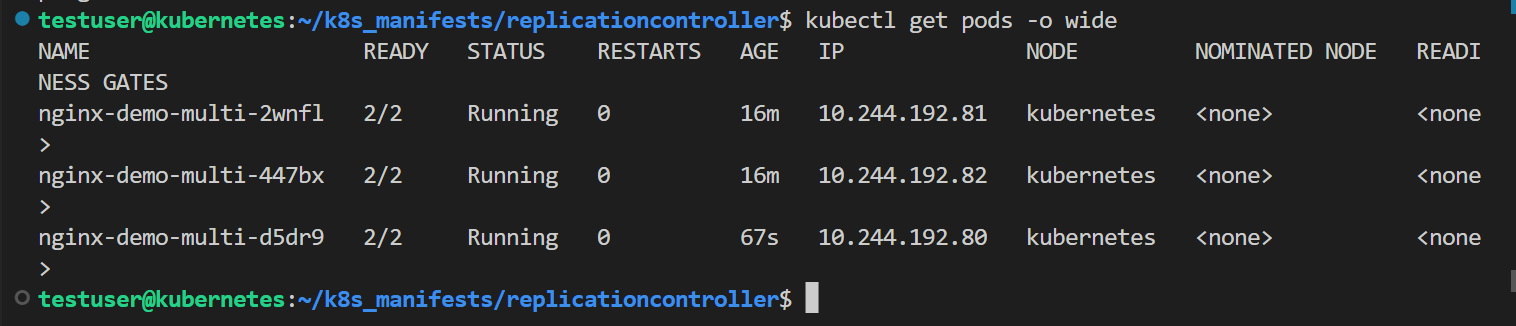


The declarative approach to scalein and scaleout is to edit the yaml file. Use the command -  
**kubectl edit rc nginx-demo-multi .** This will open the yaml file in vi editor to make changes. Change the replicas count to 3 and save the changes(:wq! & make edits to yaml file in insert mode).



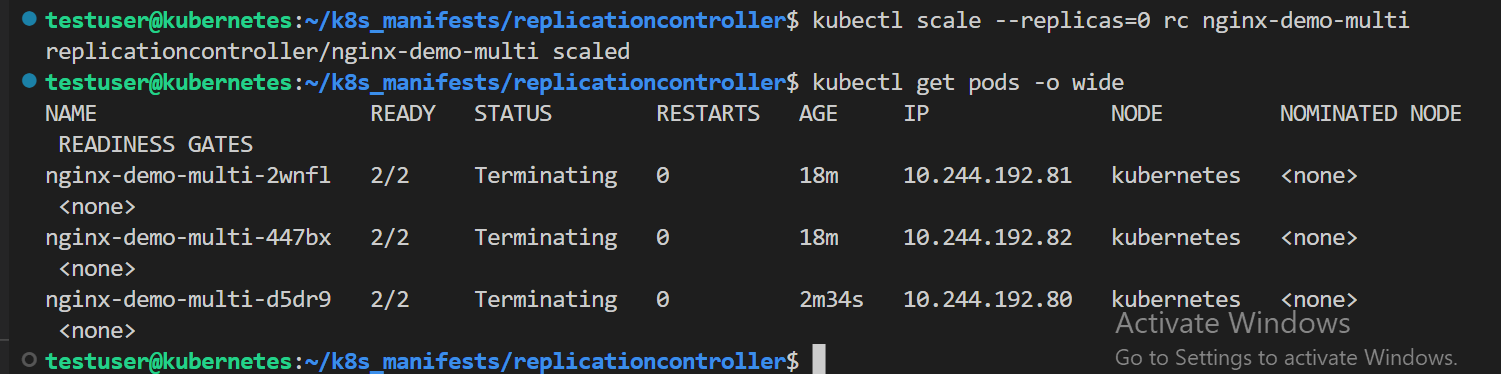


The Kube control manager will check for the match of desired state & actual state, then create or remove the pods accordingly. Check the Pods status - **kubectl get po -o wide**



**Note : The Imperative approach is the most recommended approach for scalein & scaleout.**

Without deleting the RC, all the Pods can be deleted making the replicas value equal to 0. Use the command - **kubectl scale** --**replicas=0 rc nginx-demo-multi.**



Now the Pod status – **kubectl get pods -o wide** will have zero Pods but the RC will be available, check the RC status too, use the command – **kubectl get rc -o wide** . If Pods are again needed just directly increase the replica count.

