**Pod Restart Policy:**

Pod restart policy comes under lifecycle management of pod and it is applicable on pod level but not on container level which means that it is written under the spec section of pod file rather under container section.

Restart policies that are supported for kubernetes are

| **Restart Policy** | **Description** |
| --- | --- |
| Never  –restart Never | Kubelet will not restart the containers exited with code >=0   * Even Though application fails, the kubelet will not try to start the containers. |
| Always (Default)  –restart Always | Default one. Kubelet will restart the containers exited with code >=0(failed or stopped due to any reason)   * Despite application failure, kubelet will restart them to avoid downtime. |
| OnFailure  –restart OnFailure | Kubelet will restart the containers only if they are failed with exit code > 0   * Nevertheless, the container stopped due to the failure with exit code >0, still we need to run an application. Then the kubelet will restart. |

* Kubelet restarts the container with exponential back-off delay of 10s, 20s,30s etc.

When the container goes down with an exit code >=0, then kubelet restarts them for every 10 seconds time interval till 5 mins. Until then the pod will be in an error state.

* Kubelet resets the backoff timer for the container, if no problem is found for 10 minutes.

Kubelet restarts the container after some seconds, thus the pod comes up. In this condition it will wait for 10 minutes and then the timer will be restarted. Again it will come back to zero, if the container goes down ,here it will wait for 40 seconds if the backoff timer is set to 40 seconds, then pod comes up. It will wait for 10 minutes to restart.

Let us see how pod restart policy is implemented practically on the terminal.

**Restart Policy: Always**

**Scenario-1**

Create a yaml file using vim file editor.

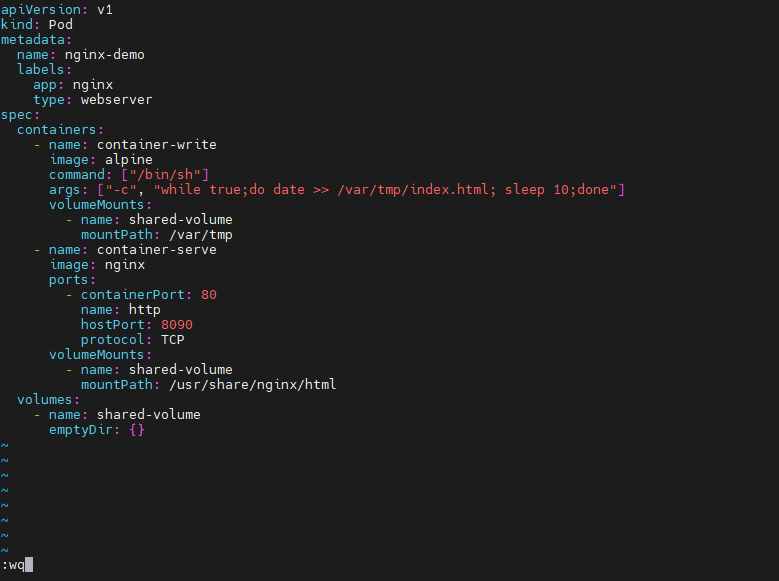


In this file, apiversion for pod is v1, kind is pod as we are trying to create a pod object and then name the pod under metadata, also label it(optional) with app and type.

Under specifications create the container by providing the name(container-write) using alpine image to write the data and run the bash shell. Now under args,we run a while loop to execute the date command for every 10 seconds, since date command output will be stored in /var/tmp/index.html. Volume mounts section is to mount the shared volume to the above container at the mount path /var/tmp.

Let us now create one more container with the name as container-serve which is created with the nginx image meant to serve the data and the container port is 80 , while mounting the path /usr/share/nginx/html directory to the shared volumes which is defined under volume section along with emptyDir. Finally save the file.

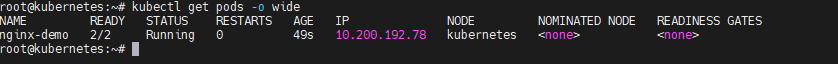
Here we have not set any restart policy, so it takes Always as a default policy.



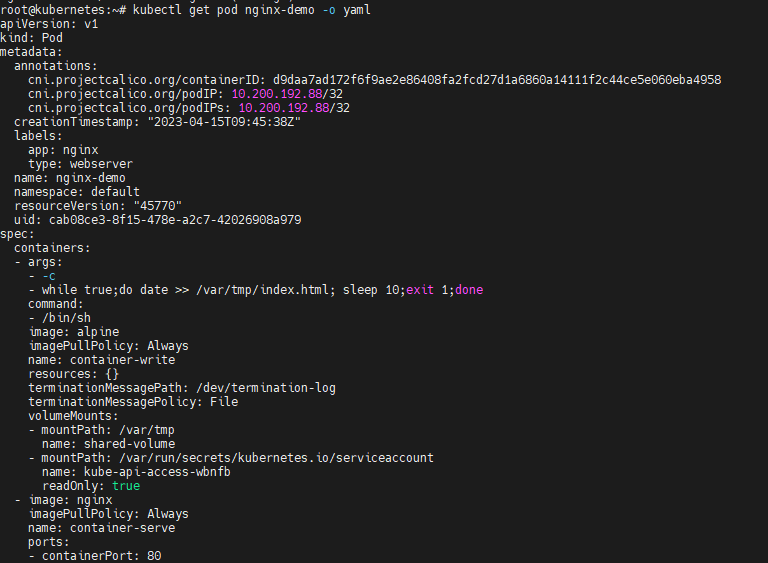
Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.

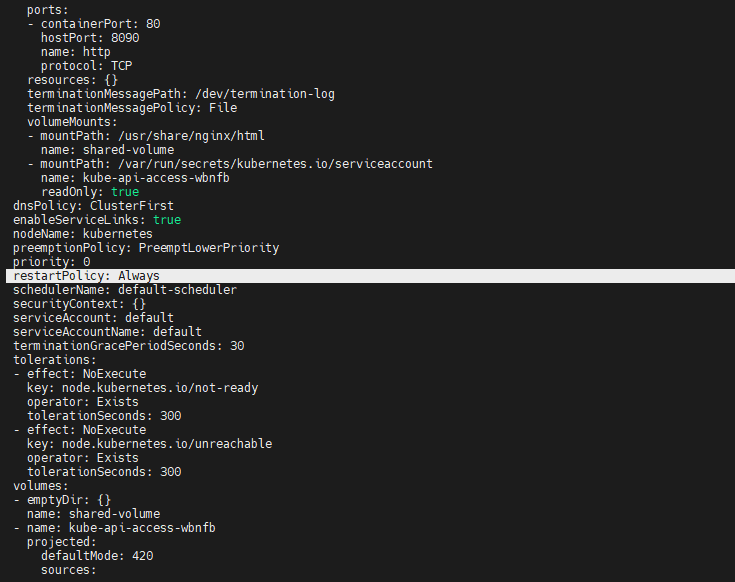


Kubectl get pods -o wide command will give the pod details like status, age, pod ip address and on which node that particular pod is created.



Kubectl get pods -o yaml command will give the pod details in yaml format. Here the restart policy is Always as it is default policy when we do not define any.





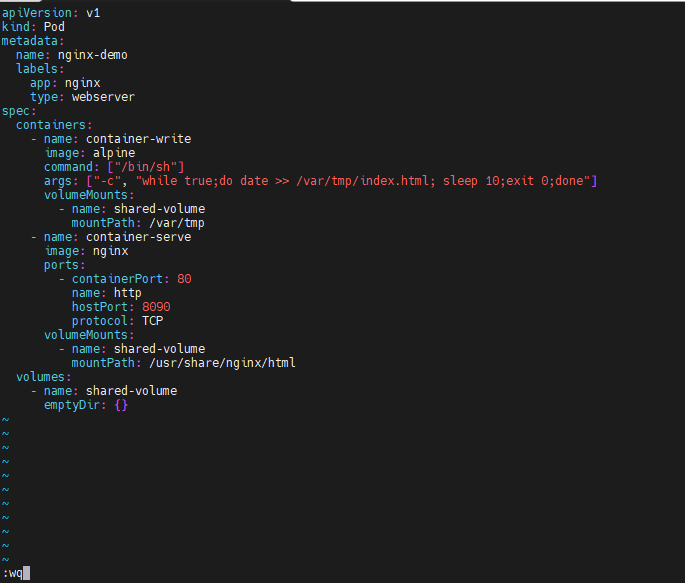
Kubectl delete -f file\_name command will delete the pod according to the specified file. Hence the pod is deleted.



**Scenario-2**

we modify the file with exit 0 in the while loop to check the condition where container exited without any error, hence it needs to get restarted.

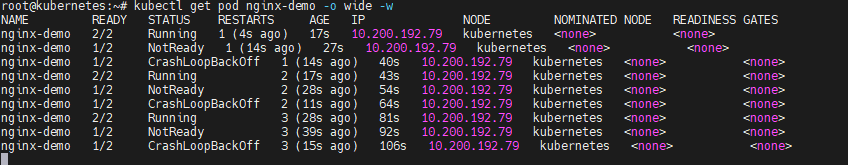




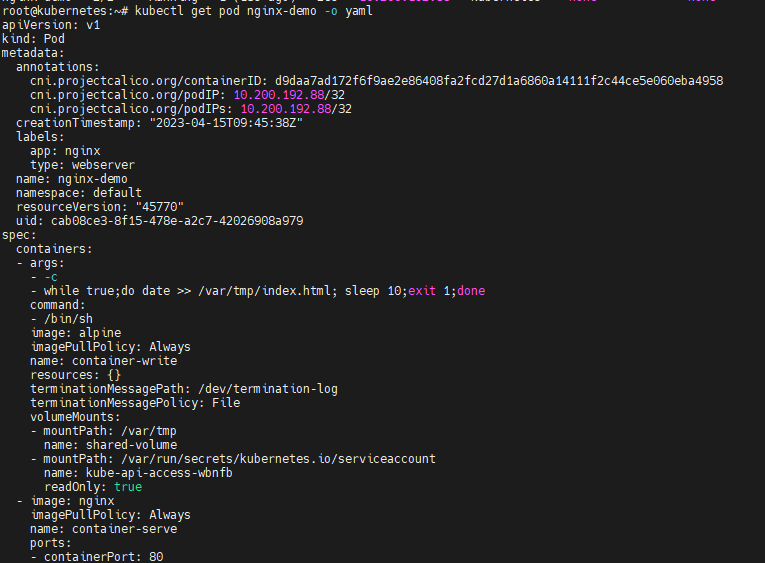
Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.

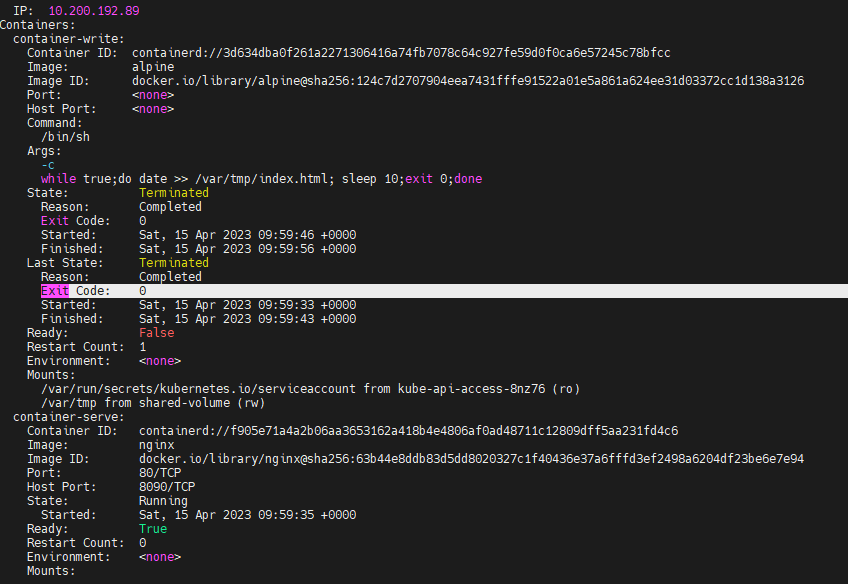


kubectl get pod nginx-demo -o wide -w command will give the detailed structure of pod restarting, where -w is for watching the pod cycle. Here pod is created then goes to the NotReady state and restarts as the policy is always.



Kubectl get pods -o yaml command will give the pod details in yaml format. Below exit code is taken as zero and restart count is one as it is restarted for one time.





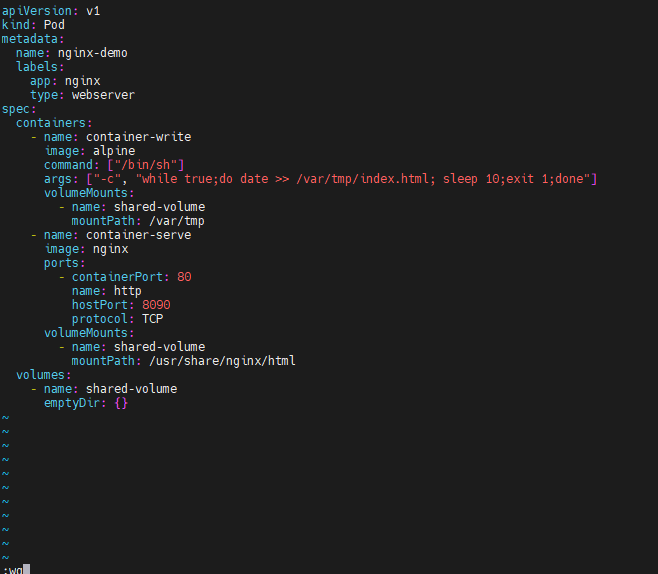
Kubectl delete -f file\_name command will delete the pod according to the specified file. Hence the pod is deleted.



**Scenario-3**

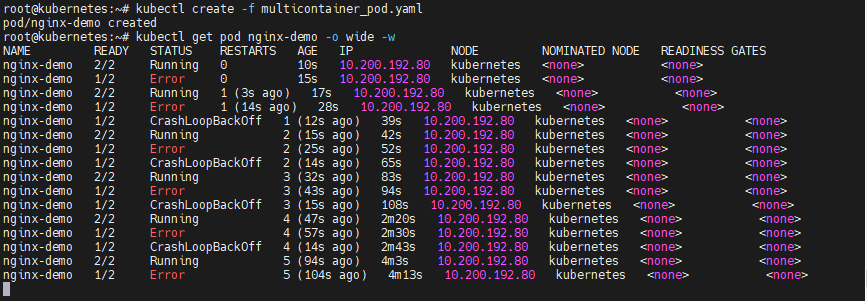
Let us modify the file with exit 1 in the while loop to check the condition where container exited with an error, hence it needs to get restarted.



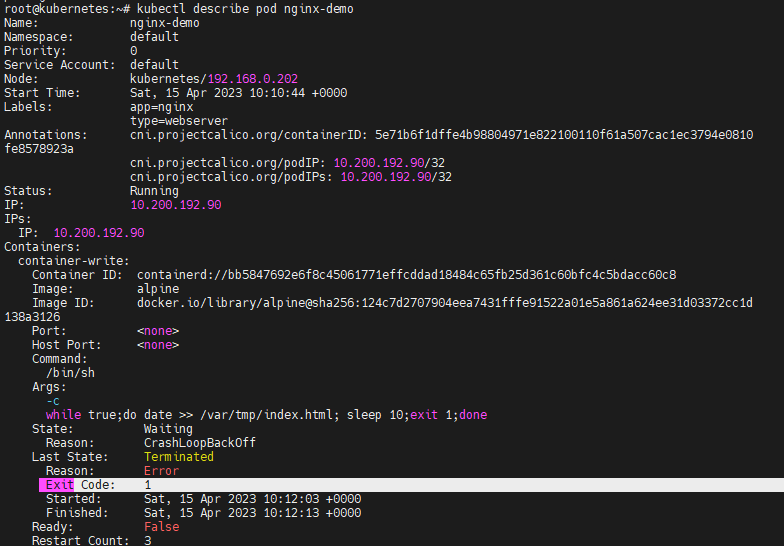


Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.

kubectl get pod nginx-demo -o wide -w command will give the detailed structure of pod restarting, where -w is for watching the pod cycle. Here pod is created then goes down with an error and restarts as the policy is always.



Kubectl describe pod nginx-demo command will give the pod details. Below exit code is one and restart count is three as it is restarted three times.



Kubectl delete -f file\_name command will delete the pod according to the specified file. Hence the pod is deleted.

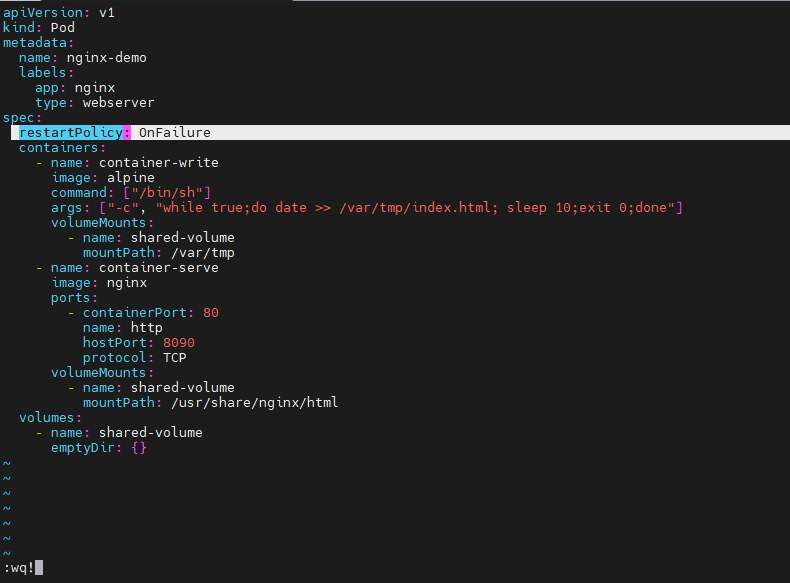


**Restart Policy: OnFailure**

**Scenario-1**

Let us set the restart policy as OnFailure and modify the file with exit 0 in the while loop to check the condition where the container exited without an error, hence it should not get restarted as it is not failed.





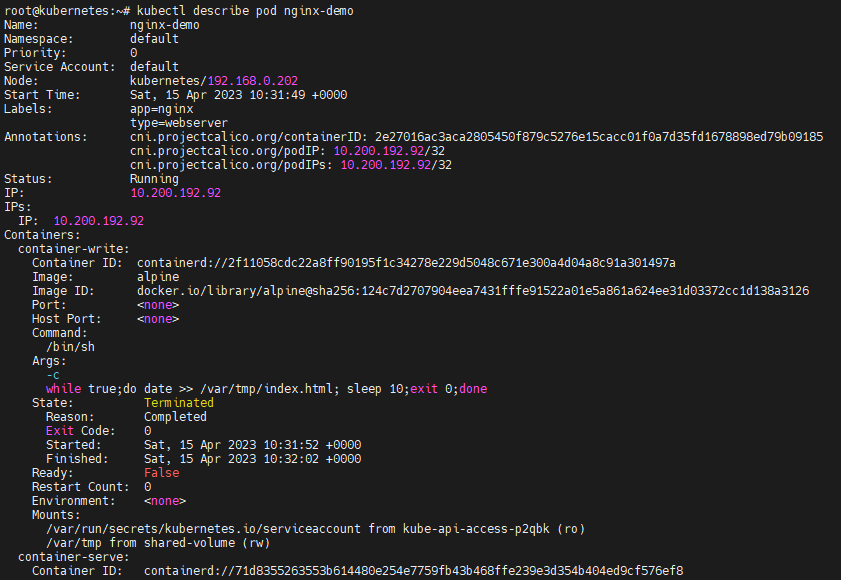
Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.



kubectl get pod nginx-demo -o wide -w command will give the detailed structure of pod restarting, where -w is for watching the pod cycle. Here pod status is NotReady as restart policy is OnFailure, the pod is exited with zero code which means it is not a failure. Thus the pod will not restart.



Kubectl describe pod nginx-demol command will give the pod details. Below exit code is zero and state is terminated but not restarted.



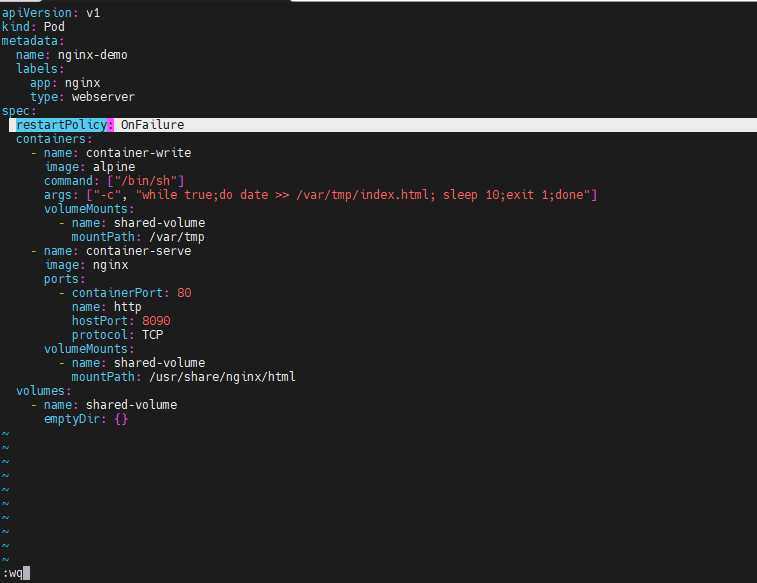
Kubectl delete -f file\_name command will delete the pod according to the specified file. Hence the pod is deleted.



**Scenario-2**

Let us set the restart policy as OnFailure and modify the file with non zero exit code in the while loop to check the condition where the container exited with an error, hence it should get restarted as pod is failed.

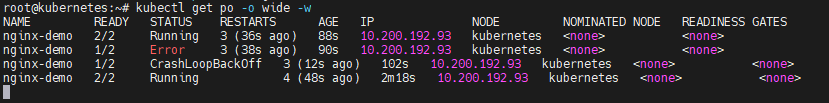




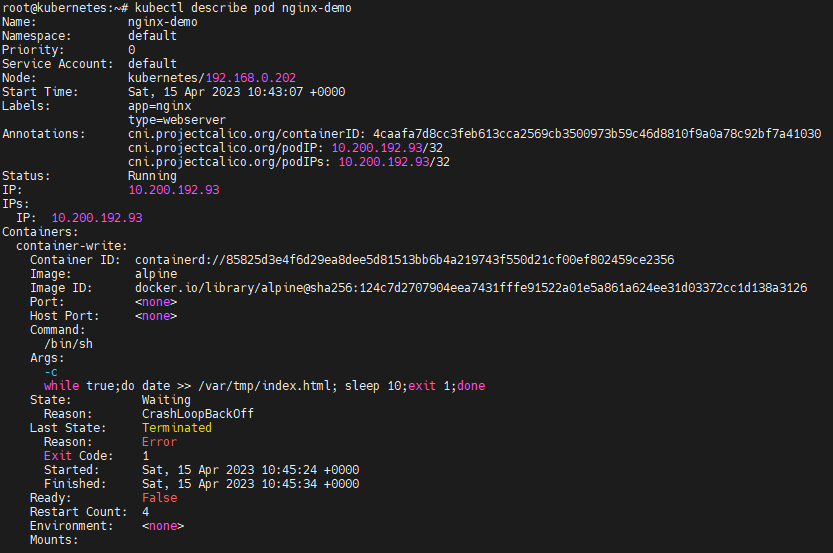
Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.



kubectl get pod nginx-demo -o wide -w command will give the detailed structure of pod restarting, where -w is for watching the pod cycle. Here pod throws an error and restarts again as restart policy is set to OnFailure, the pod is exited with non zero code which means it is a failure. Thus the pod restarted.



Kubectl describe pod nginx-demo command will give the pod details. Below exit code is one and restart count is four as it is restarted four times.



Kubectl delete -f file\_name command will delete the pod according to the specified file. Hence the pod is deleted.

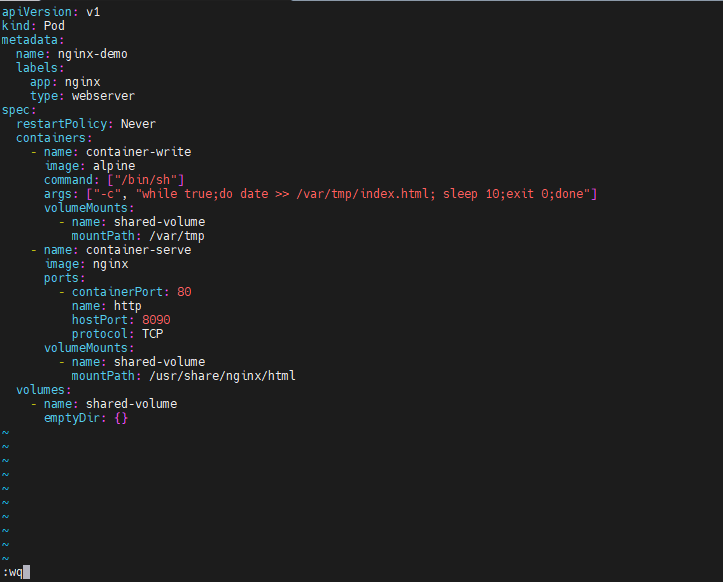


**Restart Policy: Never**

**Scenario-1**

Let us set the restart policy as Never and modify the file with zero exit code in the while loop to check the condition where the container exited without an error, hence it never gets restarted as the restart policy is set to Never.

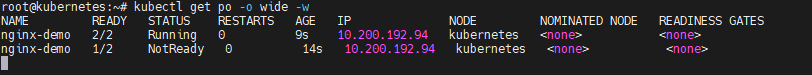




Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.



kubectl get pod nginx-demo -o wide -w command will give the detailed structure of pod restarting, where -w is for watching the pod cycle. Here the pod is in NotReady state and never restarts as restart policy is set to Never, though pod is exited with zero code without an error.



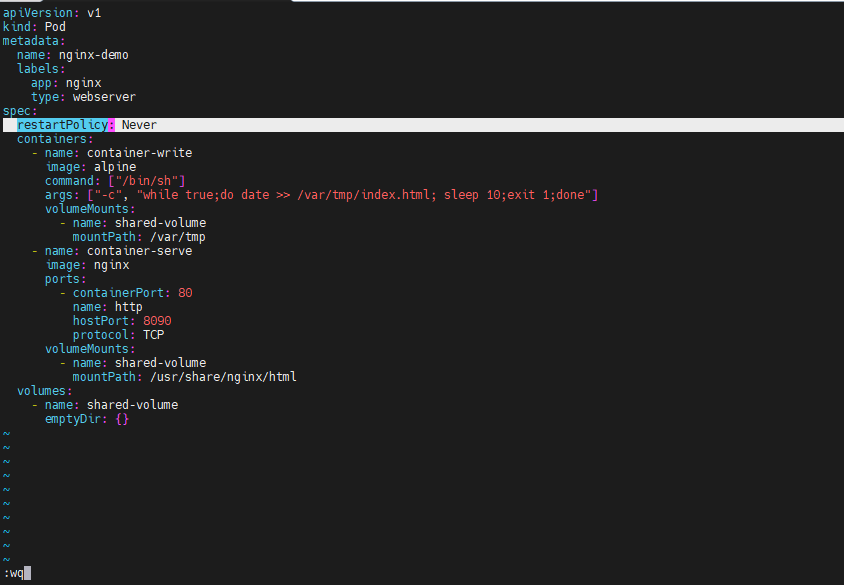
Kubectl delete -f file\_name command will delete the pod according to the specified file. Hence the pod is deleted.



**Scenario-2**

Let us set the restart policy as Never and modify the file with non zero exit code in the while loop to check the condition where the container exited with an error, hence it never gets restarted.

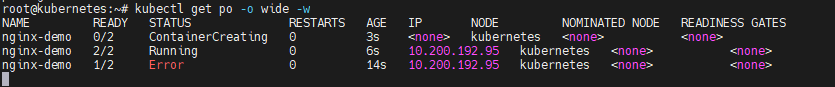




Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.



kubectl get pod nginx-demo -o wide -w command will give the detailed structure of pod restarting, where -w is for watching the pod cycle. Here the pod throws an error and never restarts as restart policy is set to Never, the pod is exited with non zero code.



Kubectl delete -f file\_name command will delete the pod according to the specified file. Hence the pod is deleted.

