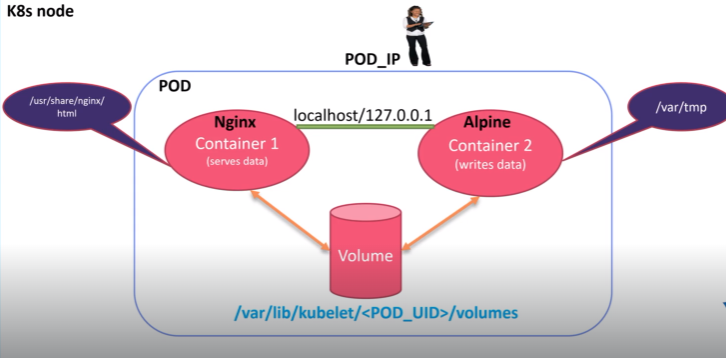
**Handling containers of a pod with crictl (restart the containers in a pod):**

When the pod is deleted, data of the pod will be lost as volume attached to the pod will also get deleted parallelly. Restarting the container solely depends upon the container runtime which is getting used in the backend(in this case it is containerd). Hence restarting the container should not lose the volume attached to it. Application serving the data may not be delayed due to the container restart.

This scenario is handled to prove that volumes attached to the containers in a pod are not container specific rather they are involved as a pod specific. As when the pod is deleted, volume inside it is also deleted but when the container is removed, volume attached to it is not removed.



To look after it practically, let us open a terminal.

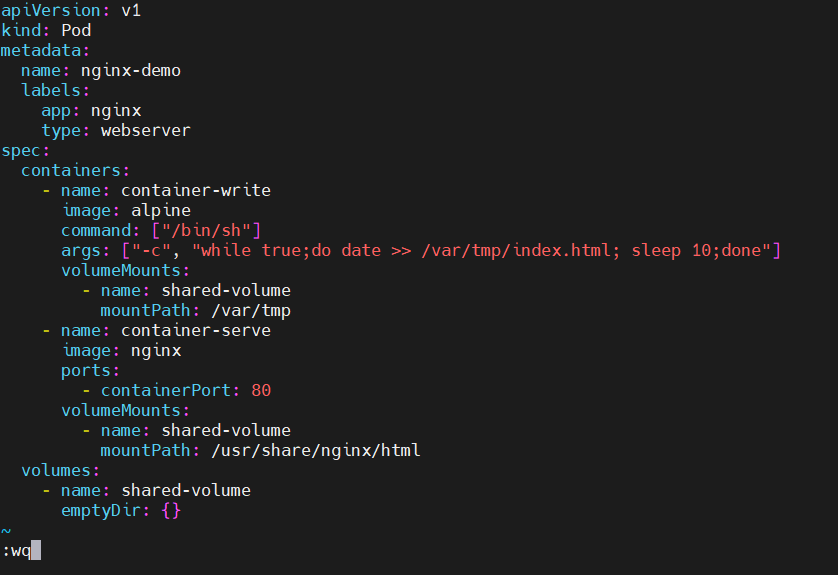
Create a file with vim file editor to create multiple containers in a pod.



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.

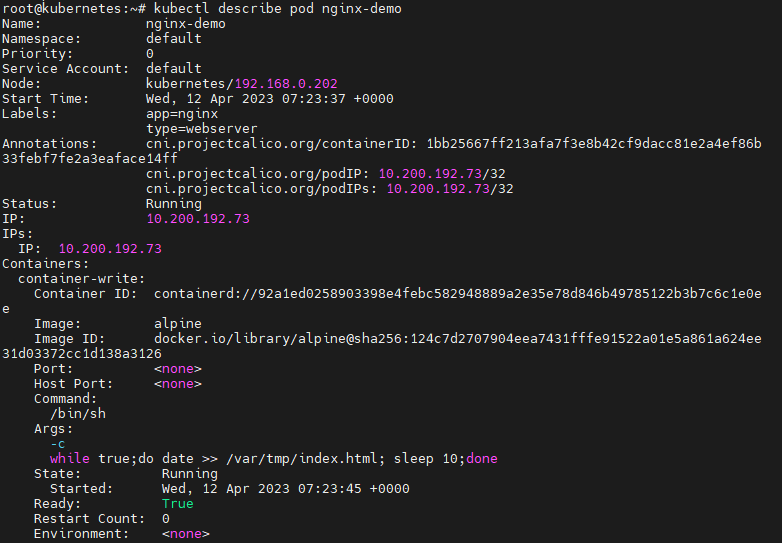
Let us now create one more container with the name as container-serve which is created with the nginx image meant to serve the data and the container port is 80 , while mounting the path /usr/share/nginx/html directory to the shared volumes which is defined under volume section along with emptyDir. Finally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

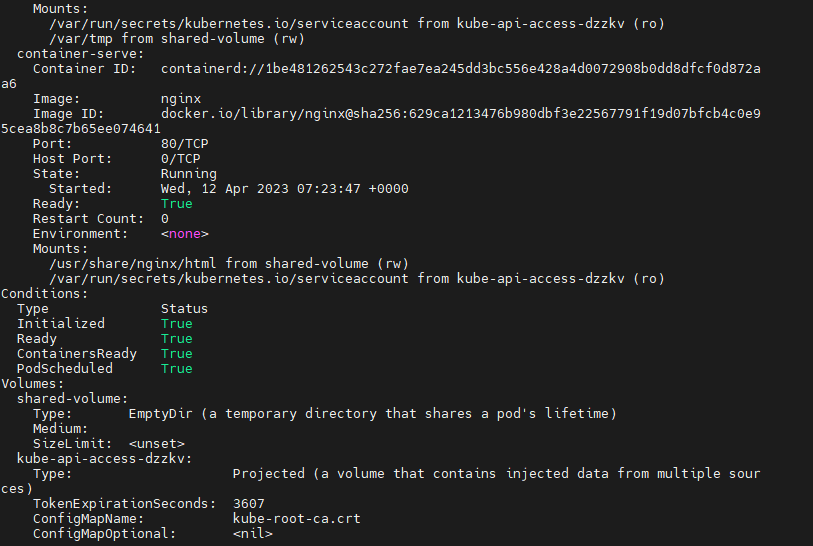


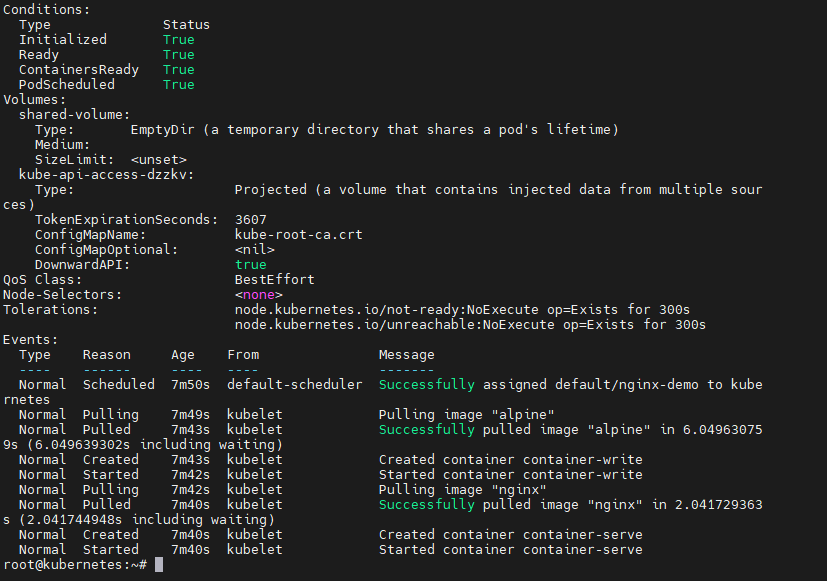
Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.



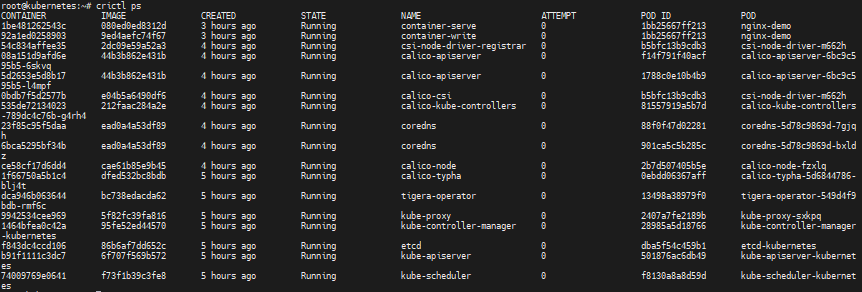
To know the complete information about the pod, run the command kubectl describe pod pod\_name. Here the restart count for both nginx container and alpine container is zero as they are not restarted till now.







As we are using containerd in the backend, we need to use crictl ps command which is a command line interface for CRI- compatible container runtimes and shows all the containers running in the kubernetes cluster.

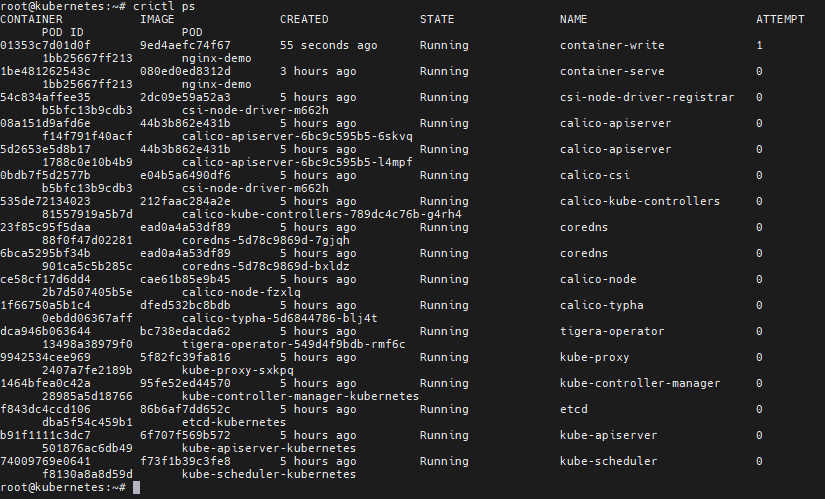


crictl stop container\_id command will stop that particular container. In this case, container-write is stopped.

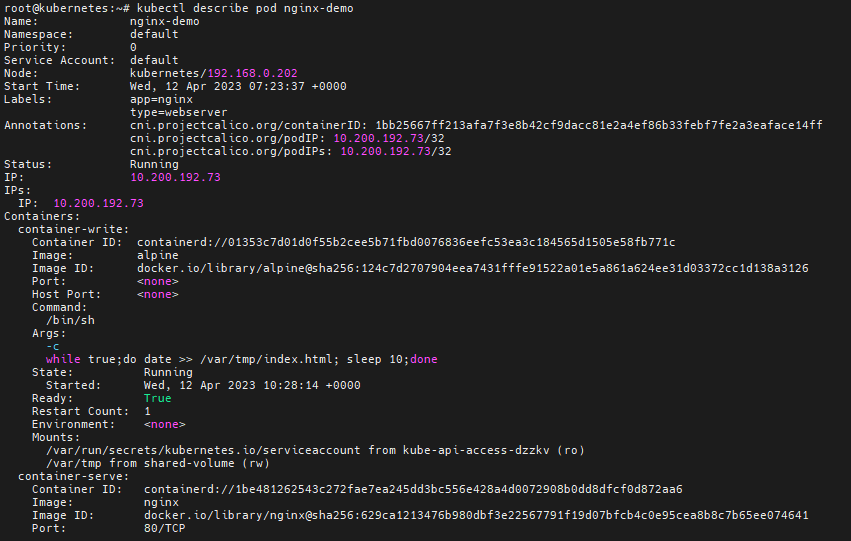
crictl rm container\_id command will remove that particular container. Here container-write is removed.

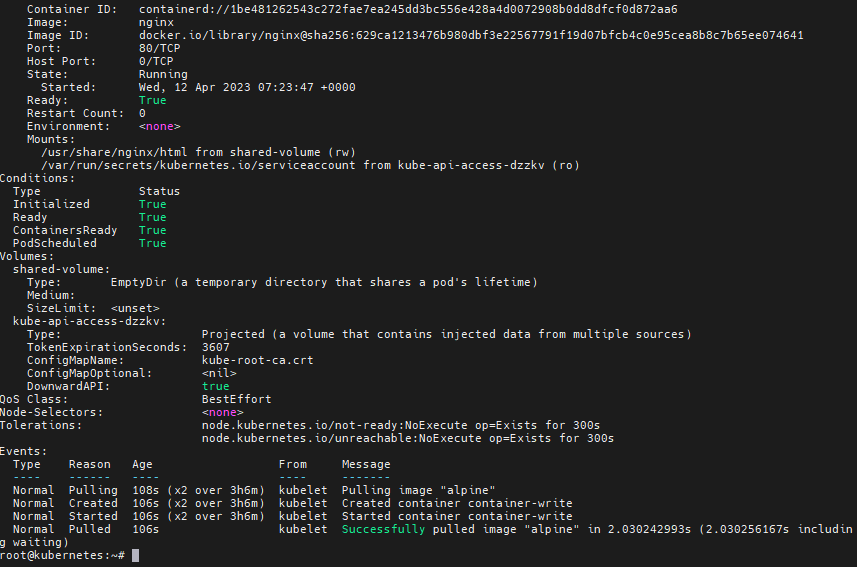


Soon after container-write deletion when we execute crictl ps command again the container is restarted with the same name immediately within the seconds, but volume does not get affected.

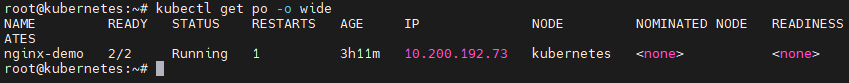


When we run kubectl describe pod pod\_name command to check the restart count of container-write which is created on alpine image shows as one as it is restarted for one time.

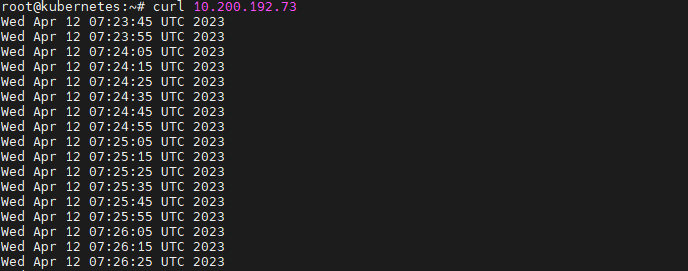




Kubectl get pods -o wide command will give the pod details like status, age, pod ip address and on which node that particular pod is created.

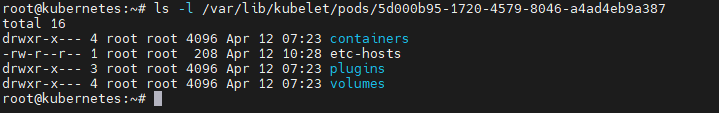


Curl the pod ip address to check whether the while loop passed in arguments of the pod file(i.e date command output) is executed for every 10 seconds.

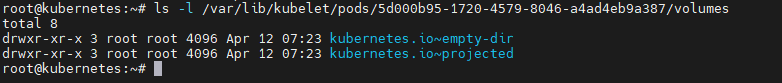


In kubernetes, volumes in pods are default stored under /var/lib/kubelet/pods/pod\_uid.To list the files under that particular path ls -l command is used.

Note: kubectl get pod pod\_name -o yaml | grep uid command will give the uid of the pod.



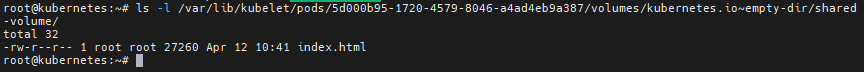
ls -l /var/lib/kubelet/pods/uid/volumes command is used to list the files and directories present under that particular path.



In the /var/lib/kubelet/pods/pod\_uid/volumes/kubernetes.io~empty-dir, the shared volume is stored



In this /var/lib/kubelet/pods/pod\_uid/volumes/kubernetes.io~empty-dir/shared-volume, the index.html is placed.



kubectl delete pod pod\_name command will delete the specified pod.



Thus when we check the volume in this path /var/lib/kubelet/pods/pod\_uid, volumes running inside that deleted pod will be deleted automatically .

