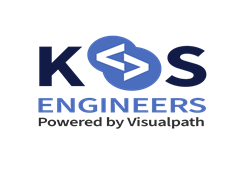
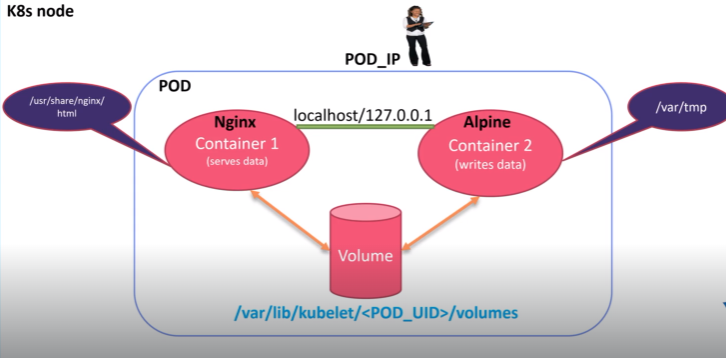
****

**Creation of Multi container pod with shared volume**

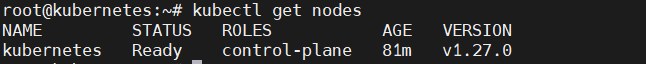
Let us assume, On a kubernetes cluster, we have a node in which a multi container pod is deployed. This pod will be allocated with the ip address to get accessed by the users and has two containers, one volume in it. One container is built on nginx image for serving the data and another one is built on the alpine image for writing the data. The requests by the user will come to the nginx application which is listening to 80 port number and alpine container will not have any application running but it is meant to run some of the shell scripts (like while loop) to write the data into the /var/tmp directory by creating files like index.html inside the directory. The data written in the /var/tmp directory will be mounted to the volume under /var/lib/kubelet/<pod\_id>/volumes directory. Since the date is stored under the volume as we use emptyDir in the pod yaml file. Also nginx application related data is stored under /usr/share/nginx/html which is also mounted to the same volume.

As data or files stored under /var/tmp directory and /usr/share/nginx/html directory are mounted to the same volume to replicate the concept of shared volume. Both the containers can communicate to each other on localhost ip address, But to communicate from outside we use pod ip address with the application port number(by default nginx runs on 80 port number). The data which is written by the alpine container is shared by the nginx application though the shared volume.



Let us see how these multi container pods are implemented practically on the terminal.

First see the node status through the command kubectl get nodes, it has to be in ready state so that pods can be deployed on it.



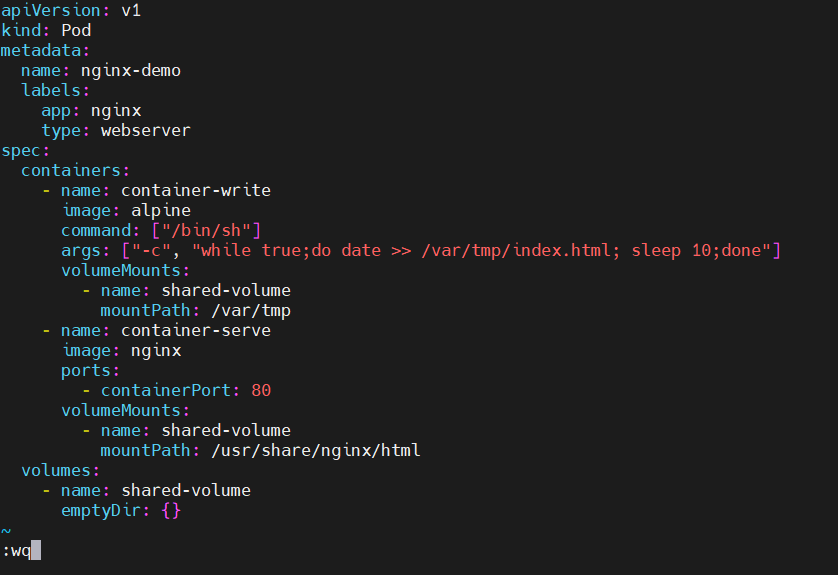
Create a yaml file using vim file editor.



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.



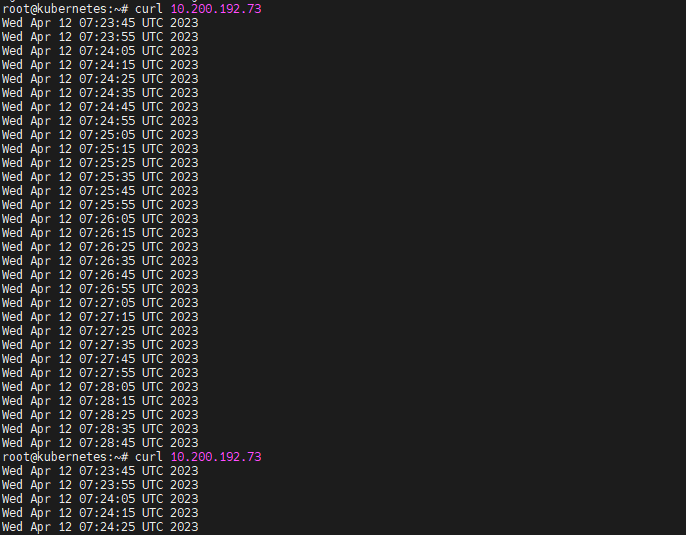
Kubectl create -f file\_name command will create the pod according to the specified file. Hence the pod is created.



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.



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.



To know the complete information about the pod, run the command kubectl describe pod pod\_name. Here Under mounts section both /usr/share/nginx/html and /var/tmp are mounted to the same volume so that whatever the data written in the index.html file of alpine container will be accessed by the nginx container as they have the shared volume in the multi container pod.

