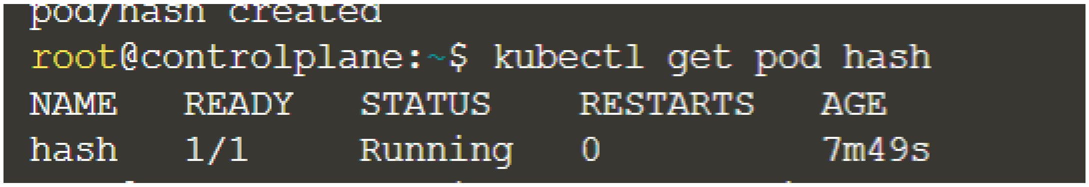
Namespace: default

Inspecting an existing Pod for mutability
Making the container immutable
Verifying the correct runtime behavior

Inspecting the Pod
Inspect the Pod named hash in the default namespace. Make sure that the Pod transitions into the Running status.

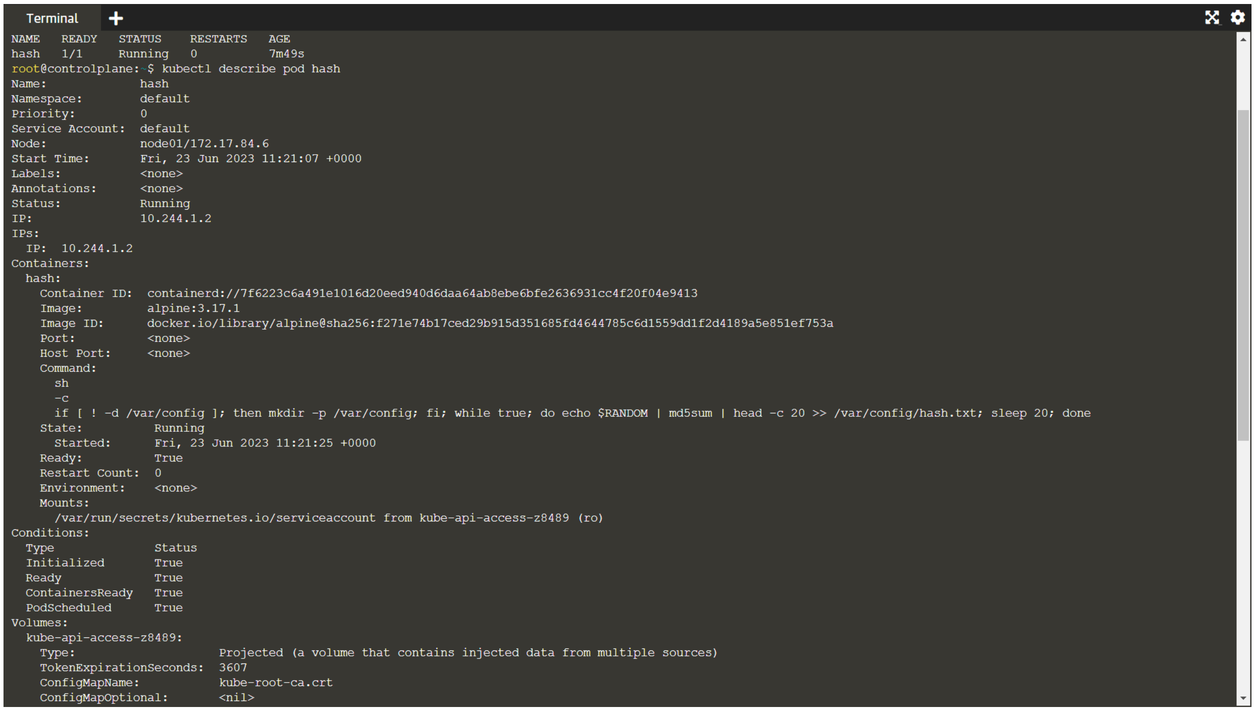
Open an interactive shell to the Pod and find the file created by the container process. Have a look at the contents of the file.

kubectl get pod hash



```
pod/hash created
root@controlplane:~$ kubectl get pod hash
NAME    READY   STATUS     RESTARTS   AGE
hash    1/1     Running    0          7m49s
```

kubectl describe pod hash



```
NAME   READY   STATUS    RESTARTS   AGE
hash   1/1     Running   0          7m49s
root@controlplane:~$ kubectl describe pod hash
Name:             hash
Namespace:        default
Priority:         0
Service Account:  default
Node:             node01/172.17.84.6
Start Time:       Fri, 23 Jun 2023 11:21:07 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.1.2
IPs:
  IP:  10.244.1.2
Containers:
  hash:
    Container ID:  containerd://7f6223c6a491e1016d20eed940d6daa64ab8ebe6bfe2636931cc4f20f04e9413
    Image:         alpine:3.17.1
    Image ID:      docker.io/library/alpine@sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a
    Port:          <none>
    Host Port:     <none>
    Command:
      sh
      -c
      if [ ! -d /var/config ]; then mkdir -p /var/config; fi; while true; do echo $RANDOM | md5sum | head -c 20 >> /var/config/hash.txt; sleep 20; done
    State:          Running
      Started:      Fri, 23 Jun 2023 11:21:25 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-z8489 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-z8489:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
```

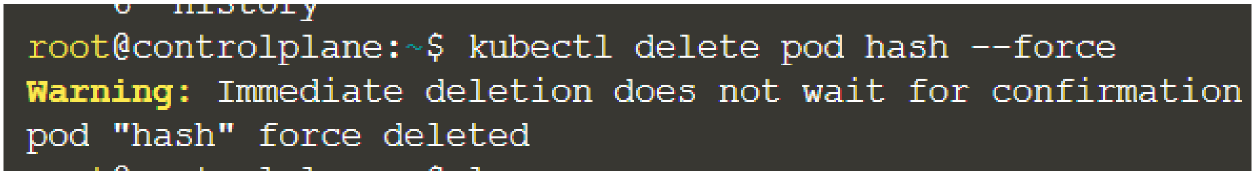
kubectl exec -it hash -- /bin/sh

history

Making the Container Immutable

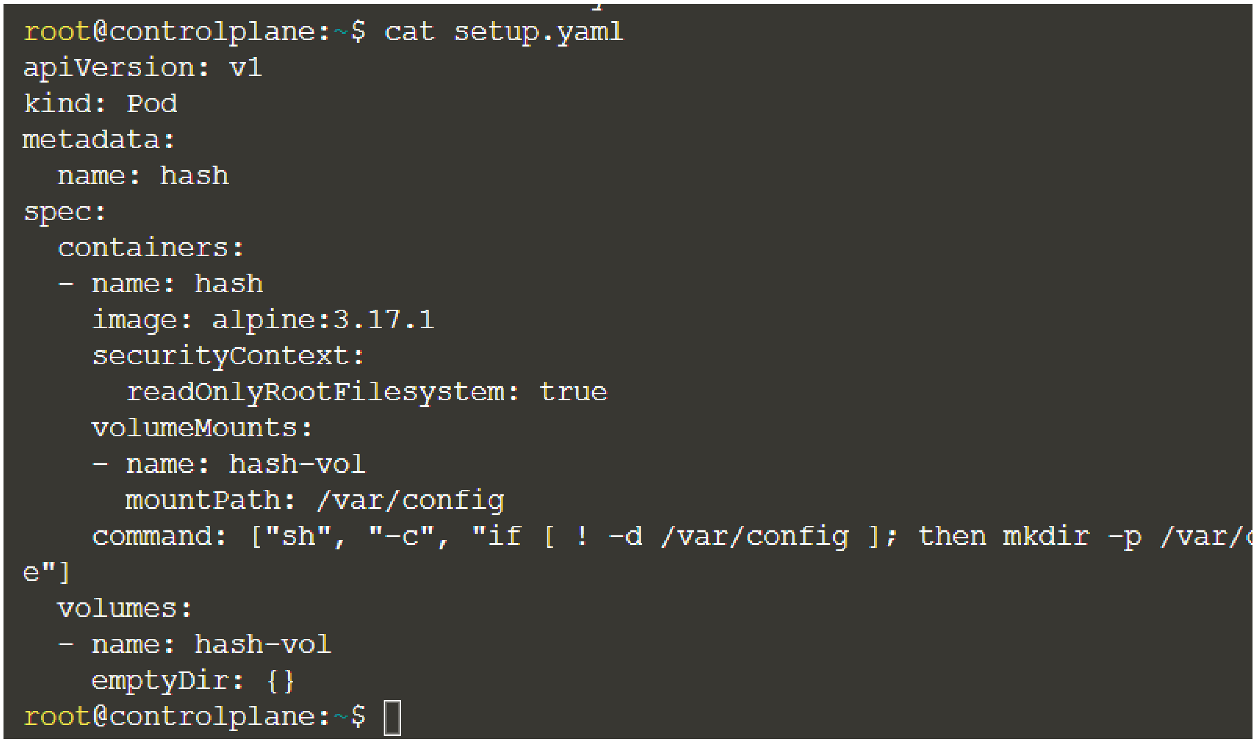
The container running in the Pod is considered mutable. Make relevant changes to the Pod so that its container can be considered immutable.

You can't modify a running Pod to make it immutable. You will first have to delete it.
$ kubectl delete pod hash --force



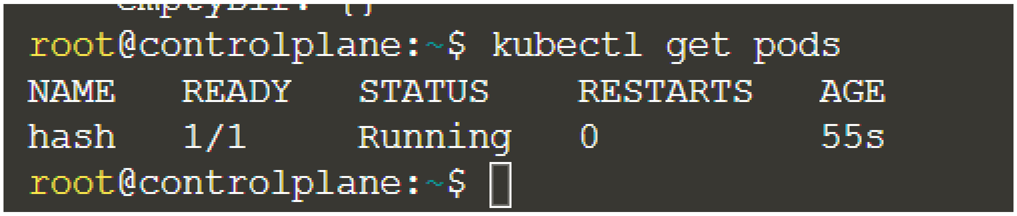
```
root@controlplane:~$ kubectl delete pod hash --force
Warning: Immediate deletion does not wait for confirmation
pod "hash" force deleted
```

You can modify the existing setup.yaml file to make the container immutable. You have to set the root filesystem to read-only access, and mount a Volume to the path /var/config to allow writing to the file named hash.txt.



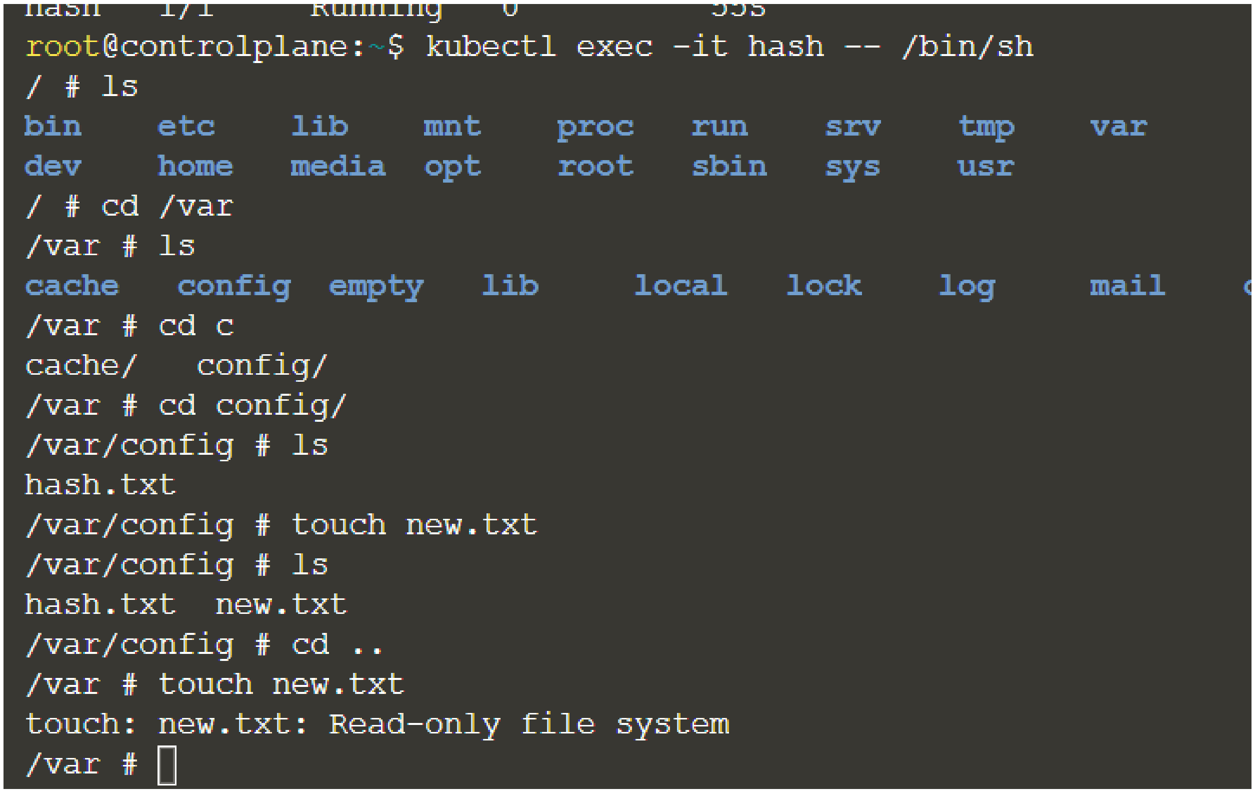
```
root@controlplane:~$ cat setup.yaml
apiVersion: v1
kind: Pod
metadata:
  name: hash
spec:
  containers:
  - name: hash
    image: alpine:3.17.1
    securityContext:
      readOnlyRootFilesystem: true
    volumeMounts:
    - name: hash-vol
      mountPath: /var/config
    command: ["sh", "-c", "if [ ! -d /var/config ]; then mkdir -p /var/c
e"]
  volumes:
  - name: hash-vol
    emptyDir: {}
root@controlplane:~$
```

Create the Pod from the modified setup.yaml file:
kubectl apply -f setup.yaml



```
    emptyDir: {}
root@controlplane:~$ kubectl get pods
NAME    READY    STATUS     RESTARTS    AGE
hash    1/1      Running    0           55s
root@controlplane:~$ 
```

You cannot write any more files to directories other than /var/config. To check, open an interactive shell and try to create a file in a read-only directory. As you will see, the attempt will render an error message.



```
hash    1/1      Running    0           55s
root@controlplane:~$ kubectl exec -it hash -- /bin/sh
/ # ls
bin    etc    lib    mnt    proc   run    srv    tmp    var
dev    home   media  opt    root   sbin   sys    usr
/ # cd /var
/var # ls
cache   config  empty   lib     local   lock    log     mail    
/var # cd c
cache/   config/
/var # cd config/
/var/config # ls
hash.txt
/var/config # touch new.txt
/var/config # ls
hash.txt  new.txt
/var/config # cd ..
/var # touch new.txt
touch: new.txt: Read-only file system
/var # 
```