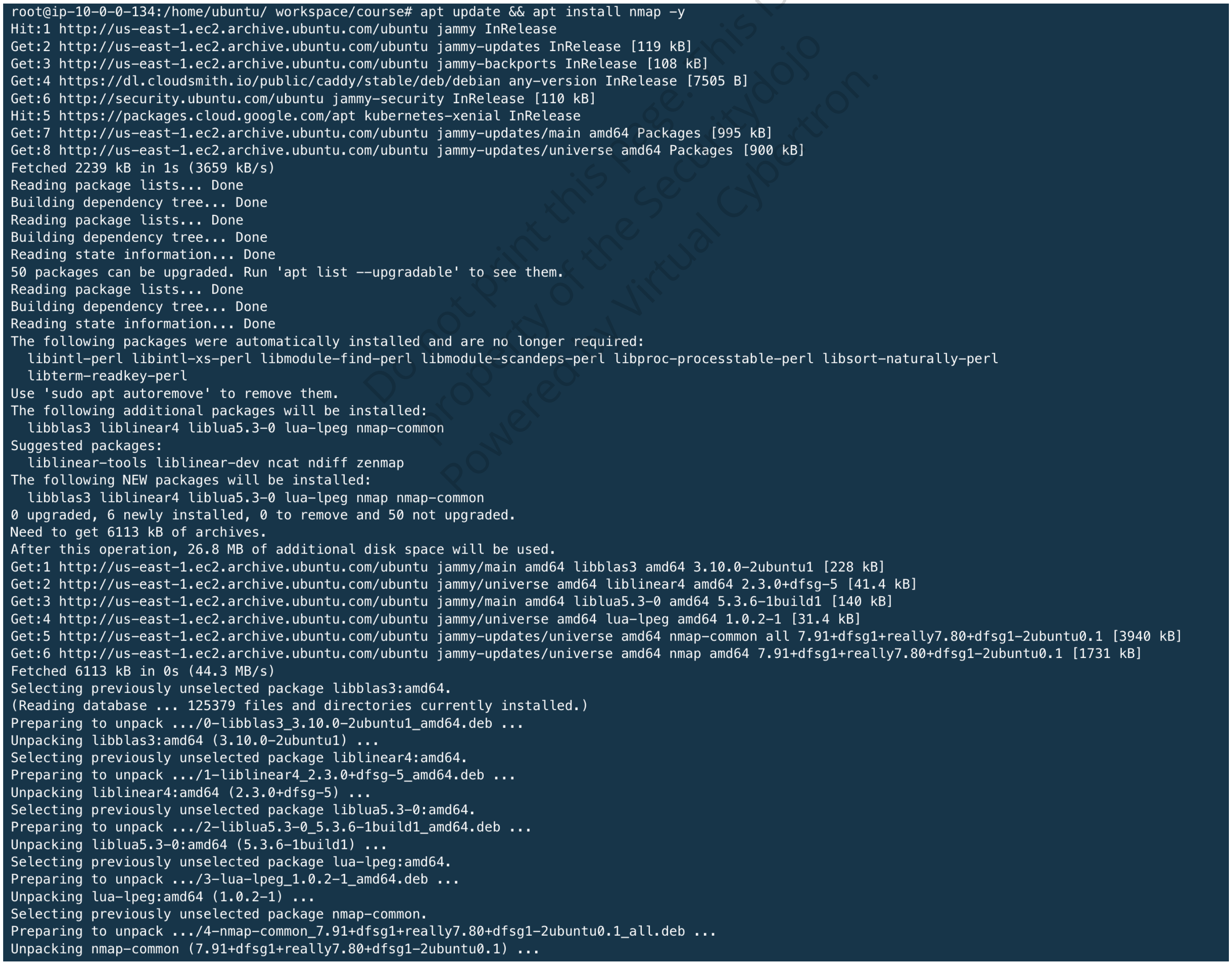
Lab: Internal Kubernetes Cluster Enumeration

Nmap Setup

Continue Using The Same Terminal

```
apt update && apt install nmap -y
```

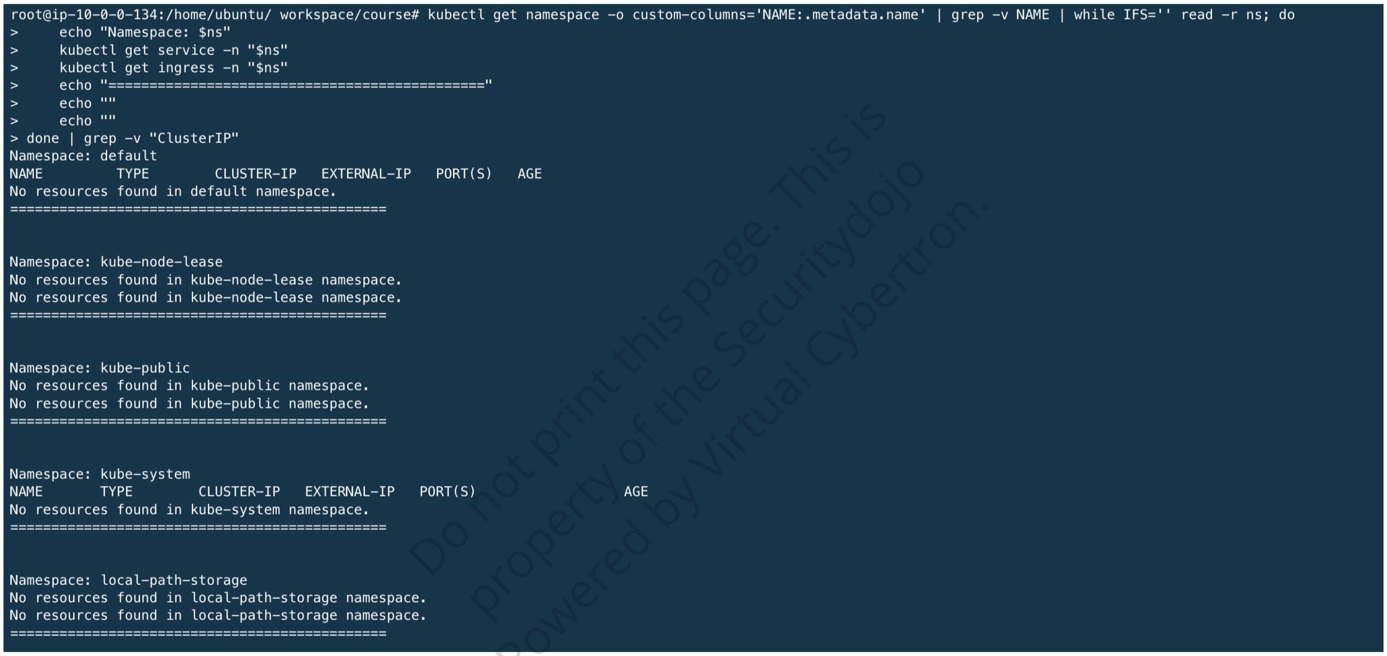


```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course# apt update && apt install nmap -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version InRelease [7505 B]
Get:6 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [995 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [900 kB]
Fetched 2239 kB in 1s (3659 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
50 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libintl-perl libintl-xs-perl libmodule-find-perl libmodule-scandeps-perl libproc-processtable-perl libsort-naturally-perl
  libterm-readkey-perl
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libblas3 liblinear4 liblua5.3-0 lua-lpeg nmap-common
Suggested packages:
  liblinear-tools liblinear-dev ncat ndiff zenmap
The following NEW packages will be installed:
  libblas3 liblinear4 liblua5.3-0 lua-lpeg nmap nmap-common
0 upgraded, 6 newly installed, 0 to remove and 50 not upgraded.
Need to get 6113 kB of archives.
After this operation, 26.8 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libblas3 amd64 3.10.0-2ubuntu1 [228 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 liblinear4 amd64 2.3.0+dfsg-5 [41.4 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1build1 [140 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 lua-lpeg amd64 1.0.2-1 [31.4 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap-common all 7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1 [3940 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap amd64 7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1 [1731 kB]
Fetched 6113 kB in 0s (44.3 MB/s)
Selecting previously unselected package libblas3:amd64.
(Reading database ... 125379 files and directories currently installed.)
Preparing to unpack .../0-libblas3_3.10.0-2ubuntu1_amd64.deb ...
Unpacking libblas3:amd64 (3.10.0-2ubuntu1) ...
Selecting previously unselected package liblinear4:amd64.
Preparing to unpack .../1-liblinear4_2.3.0+dfsg-5_amd64.deb ...
Unpacking liblinear4:amd64 (2.3.0+dfsg-5) ...
Selecting previously unselected package liblua5.3-0:amd64.
Preparing to unpack .../2-liblua5.3-0_5.3.6-1build1_amd64.deb ...
Unpacking liblua5.3-0:amd64 (5.3.6-1build1) ...
Selecting previously unselected package lua-lpeg:amd64.
Preparing to unpack .../3-lua-lpeg_1.0.2-1_amd64.deb ...
Unpacking lua-lpeg:amd64 (1.0.2-1) ...
Selecting previously unselected package nmap-common.
Preparing to unpack .../4-nmap-common_7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1_all.deb ...
Unpacking nmap-common (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
```

Hands on Enumerations

- Run the below command to get the ingress and service which might be accessible.

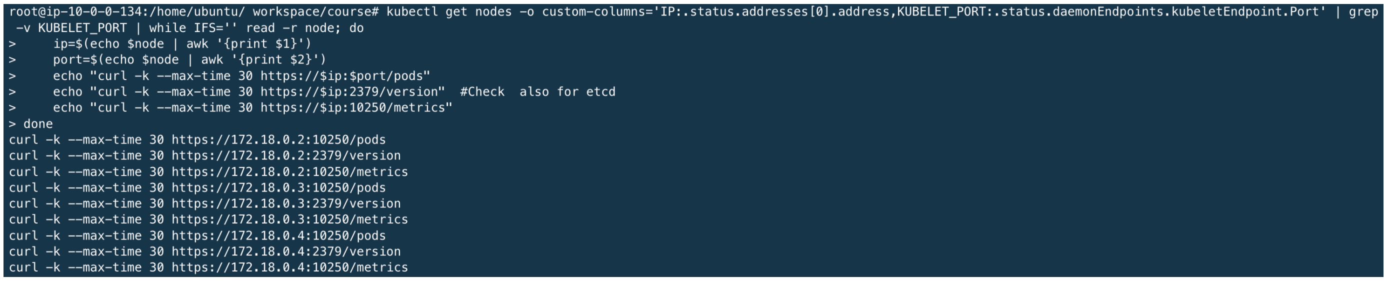
```
kubectl get namespace -o custom-columns='NAME:.metadata.name' | grep -v NAME |
while IFS='' read -r ns; do
    echo "Namespace: $ns"
    kubectl get service -n "$ns"
    kubectl get ingress -n "$ns"
    echo "=============================================="
    echo ""
    echo ""
done | grep -v "ClusterIP"
```



```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course# kubectl get namespace -o custom-columns='NAME:.metadata.name' | grep -v NAME | while IFS='' read -r ns; do
>     echo "Namespace: $ns"
>     kubectl get service -n "$ns"
>     kubectl get ingress -n "$ns"
>     echo "=============================================="
>     echo ""
>     echo ""
> done | grep -v "ClusterIP"
Namespace: default
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
No resources found in default namespace.
==============================================


Namespace: kube-node-lease
No resources found in kube-node-lease namespace.
No resources found in kube-node-lease namespace.
==============================================


Namespace: kube-public
No resources found in kube-public namespace.
No resources found in kube-public namespace.
==============================================


Namespace: kube-system
NAME       TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)                  AGE
No resources found in kube-system namespace.
==============================================


Namespace: local-path-storage
No resources found in local-path-storage namespace.
No resources found in local-path-storage namespace.
==============================================
```

- Check if node listing is allowed, then use it to list of kubelets endpoints via below command.

```
kubectl get nodes -o custom-
columns='IP:.status.addresses[0].address,KUBELET_PORT:.status.daemonEndpoints.k
ubeletEndpoint.Port' | grep -v KUBELET_PORT | while IFS='' read -r node; do
    ip=$(echo $node | awk '{print $1}')
    port=$(echo $node | awk '{print $2}')
    echo "curl -k --max-time 30 https://$ip:$port/pods"
    echo "curl -k --max-time 30 https://$ip:2379/version"  #Check  also for
etcd
    echo "curl -k --max-time 30 https://$ip:10250/metrics"
done
```

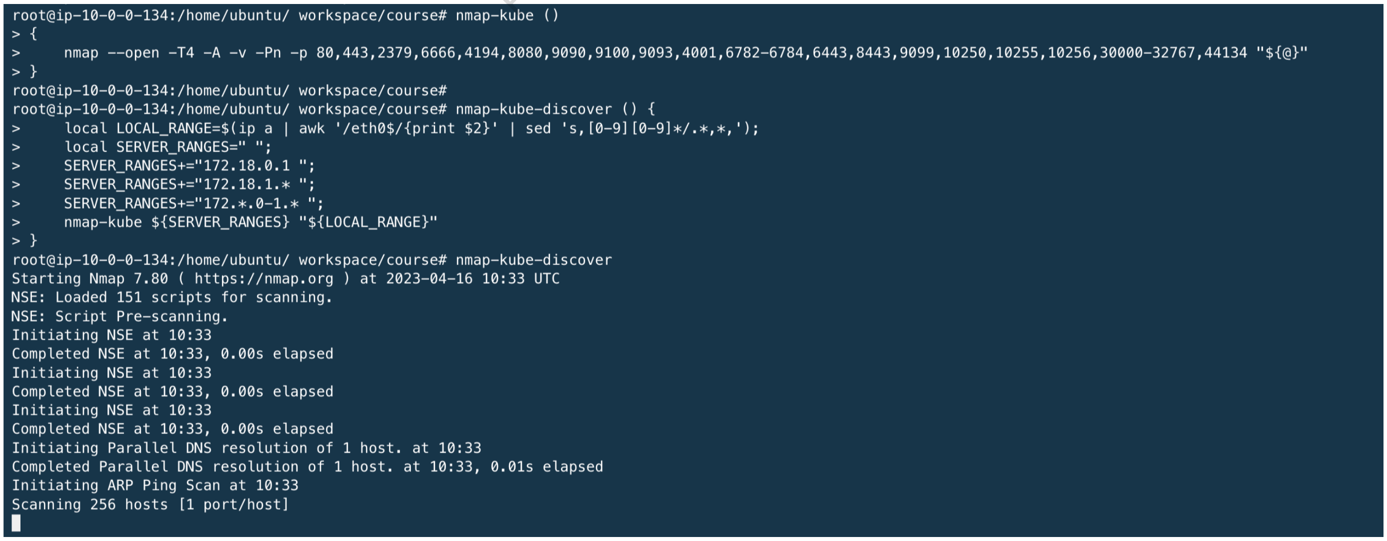


```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course# kubectl get nodes -o custom-columns='IP:.status.addresses[0].address,KUBELET_PORT:.status.daemonEndpoints.kubeletEndpoint.Port' | grep
 -v KUBELET_PORT | while IFS='' read -r node; do
>     ip=$(echo $node | awk '{print $1}')
>     port=$(echo $node | awk '{print $2}')
>     echo "curl -k --max-time 30 https://$ip:$port/pods"
>     echo "curl -k --max-time 30 https://$ip:2379/version"  #Check  also for etcd
>     echo "curl -k --max-time 30 https://$ip:10250/metrics"
> done
curl -k --max-time 30 https://172.18.0.2:10250/pods
curl -k --max-time 30 https://172.18.0.2:2379/version
curl -k --max-time 30 https://172.18.0.2:10250/metrics
curl -k --max-time 30 https://172.18.0.3:10250/pods
curl -k --max-time 30 https://172.18.0.3:2379/version
curl -k --max-time 30 https://172.18.0.3:10250/metrics
curl -k --max-time 30 https://172.18.0.4:10250/pods
curl -k --max-time 30 https://172.18.0.4:2379/version
curl -k --max-time 30 https://172.18.0.4:10250/metrics
```

- Use the bash script to scan the IP ranges of the kubernetes cluster for open ports.

```
nmap-kube ()
{
    nmap --open -T4 -A -v -Pn -p
80,443,2379,6666,4194,8080,9090,9100,9093,4001,6782-6784,6443,8443,9099,10250,1
0255,10256,30000-32767,44134 "${@}"
}

nmap-kube-discover () {
    local LOCAL_RANGE=$(ip a | awk '/eth0$/{print $2}' | sed 's,[0-9]
[0-9]*/.*,*,');
    local SERVER_RANGES=" ";
    SERVER_RANGES+="172.18.0.1 ";
    SERVER_RANGES+="172.18.1.* ";
    SERVER_RANGES+="172.*.0-1.* ";
    nmap-kube ${SERVER_RANGES} "${LOCAL_RANGE}"
}
nmap-kube-discover
```

Change the SERVER_RANGES based on Node IPs on the Kubernetes Cluster.



```
root@ip-10-0-0-134:/home/ubuntu/ workspace/course# nmap-kube ()
> {
>     nmap --open -T4 -A -v -Pn -p 80,443,2379,6666,4194,8080,9090,9100,9093,4001,6782-6784,6443,8443,9099,10250,10255,10256,30000-32767,44134 "${@}"
> }
root@ip-10-0-0-134:/home/ubuntu/ workspace/course#
root@ip-10-0-0-134:/home/ubuntu/ workspace/course# nmap-kube-discover () {
>     local LOCAL_RANGE=$(ip a | awk '/eth0$/{print $2}' | sed 's,[0-9][0-9]*/.*,*,');
>     local SERVER_RANGES=" ";
>     SERVER_RANGES+="172.18.0.1 ";
>     SERVER_RANGES+="172.18.1.* ";
>     SERVER_RANGES+="172.*.0-1.* ";
>     nmap-kube ${SERVER_RANGES} "${LOCAL_RANGE}"
> }
root@ip-10-0-0-134:/home/ubuntu/ workspace/course# nmap-kube-discover
Starting Nmap 7.80 ( https://nmap.org ) at 2023-04-16 10:33 UTC
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:33
Completed NSE at 10:33, 0.00s elapsed
Initiating NSE at 10:33
Completed NSE at 10:33, 0.00s elapsed
Initiating NSE at 10:33
Completed NSE at 10:33, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 10:33
Completed Parallel DNS resolution of 1 host. at 10:33, 0.01s elapsed
Initiating ARP Ping Scan at 10:33
Scanning 256 hosts [1 port/host]
█
```

Reference:

- cloud.hacktricks.xyz
- www.optiv.com