<table>
<tr><td>College Name</td><td colspan="4">University of Hertfordshire
SEGi College <u>Subang Jaya</u></td></tr>
<tr><td>Programme Name</td><td colspan="4">BACHELOR OF <u>CYBERSECURITY AND NETWORKS</u></td></tr>
<tr><td rowspan="2">Module Name</td><td rowspan="2">CYBERSECURITY</td><td>Module Code</td><td>6COM1040</td></tr>
<tr><td>Semester</td><td>SEP 2025</td></tr>
<tr><td>Module Leader</td><td>DR ANESHKUMAR THANGAVELOO</td><td rowspan="2">Assessment Type</td><td rowspan="2">PROJECT PROGRESS REPORT</td></tr>
<tr><td>Lecturer Name</td><td>MS NUR DIANA MADINAH</td></tr>
<tr><td>Student’s declaration</td><td colspan="4">I hereby certify that this assignment is my own work and where materials have been used from other resources, they have been properly acknowledged. I also understand I will face the possibility of failing the module if the content of this assignment is plagiarized.
<table>
<tr><th>No.</th><th>Name</th><th>Student ID</th><th>Signature / Initial</th></tr>
<tr><td>1</td><td>TAN JIAJIAN</td><td>SCSJ1901781</td><td>TAN</td></tr>
</table></td></tr>
<tr><td>Release Date</td><td></td><td>Submission Due Date</td><td>8/12/2025</td><td rowspan="2">Marks obtained:</td></tr>
<tr><td>Date Received</td><td></td><td>Student’s work assessed by / date</td><td></td></tr>
</table>

Module Leader's Feedback.

Contents

List of Figure

3. Implementation of Security Measures

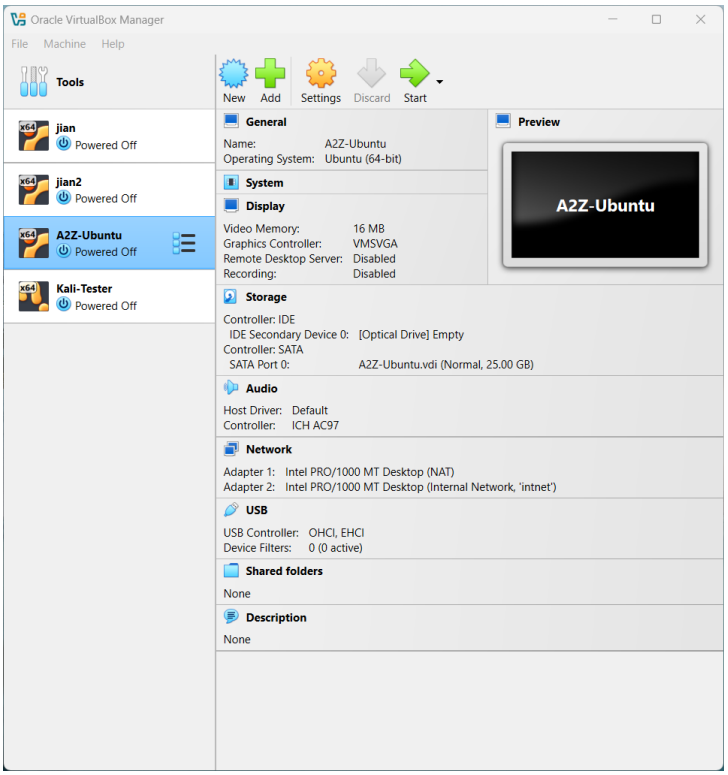

Figure 1

Figure 2 is A2Z-Ubuntu virtual machine.

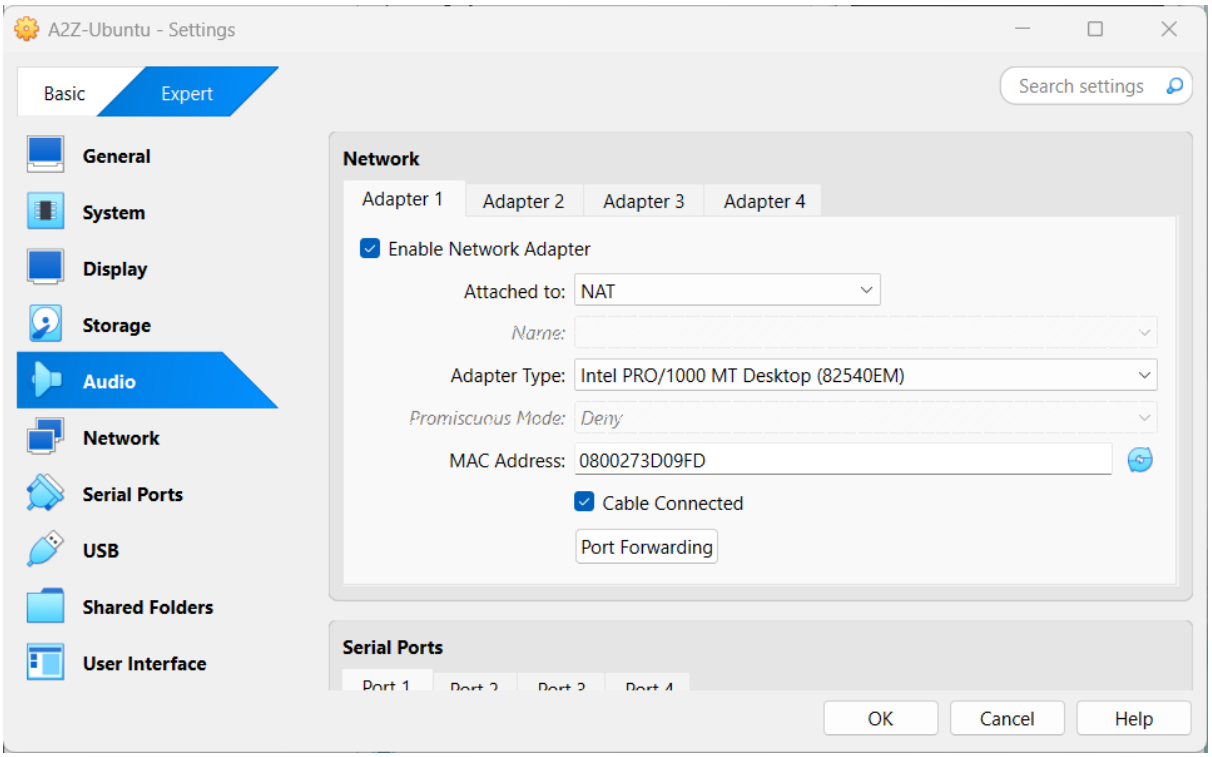

Figure 2

Figure 3 is Adapter 1 of A2Z-Ubuntu virtual machine, I set it as NAT.

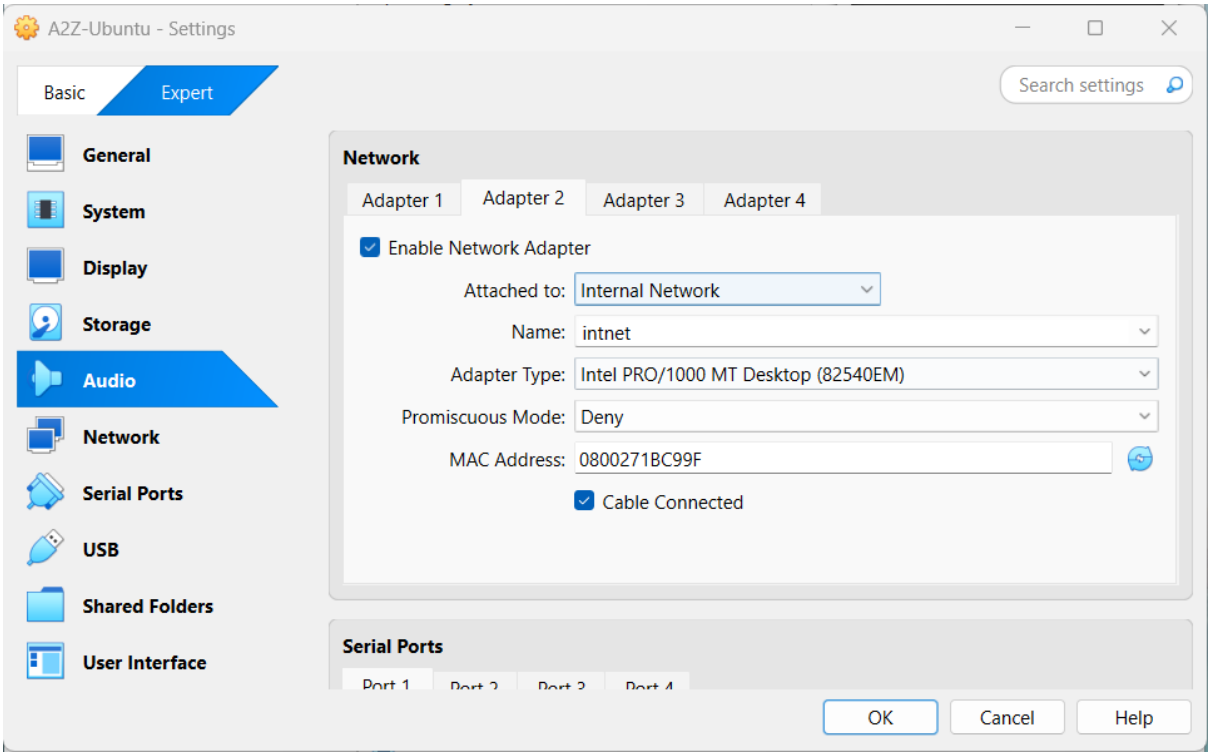

Figure 3

Figure 4 is Adapter 2 of A2Z-Ubuntu virtual machine, I set it as Internal Network because need to set an IP address to connect with Kali-Linux virtual machine. I set the IP address as 192.168.10.1.

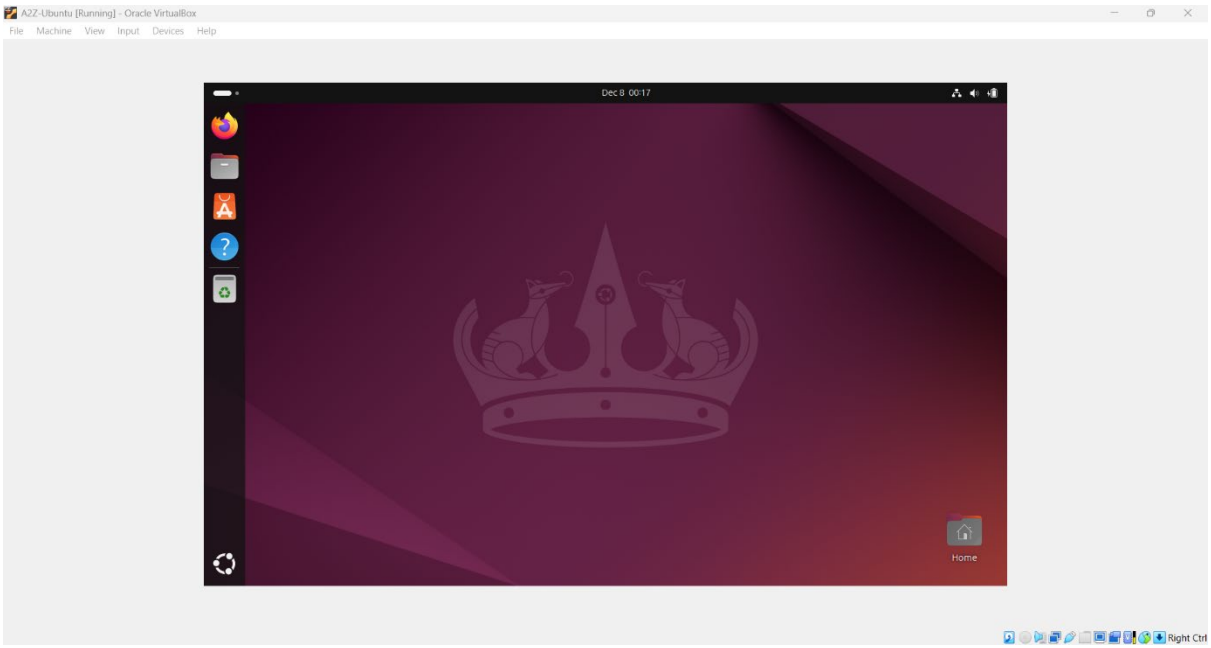

Figure 4

Figure 5 is A2Z-Ubuntu virtual machine home screen.

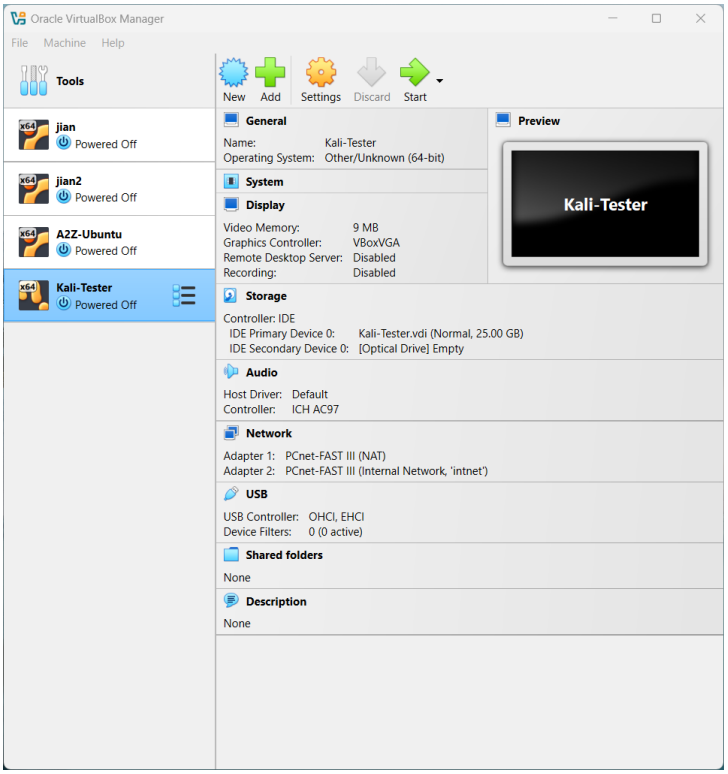

Figure 5

Figure 6 is Kali-Tester virtual machine.

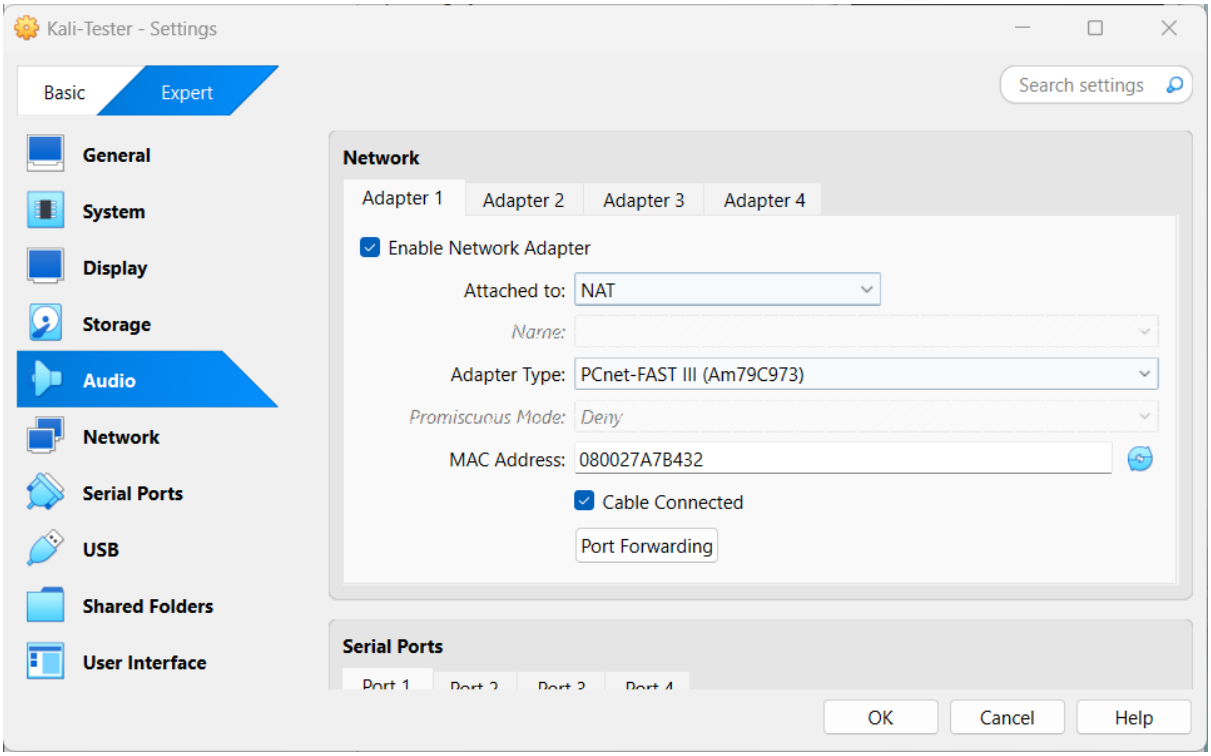

Figure 6

Figure 7 is Adapter 1 of Kali-Tester virtual machine, I set it as NAT.

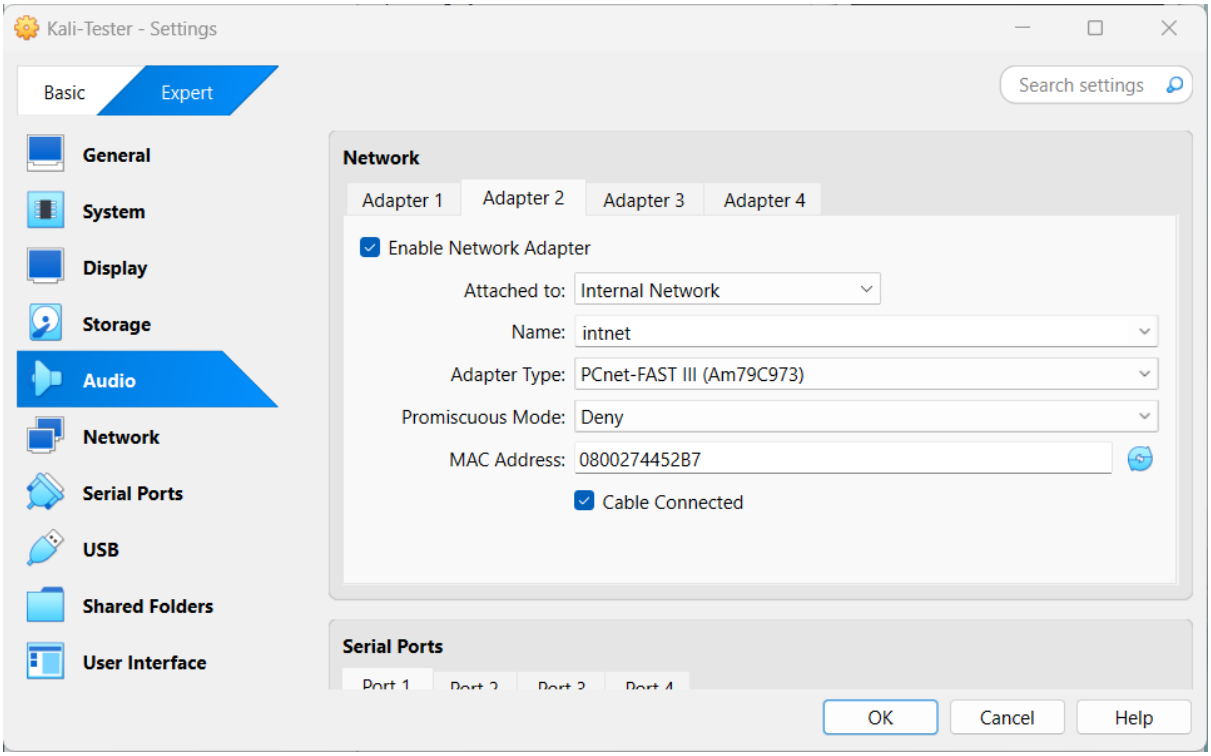

Figure 7

Figure 8 is Adapter 2 of Kali-Tester virtual machine, I set it as Internal Network because need to set an IP address to connect with Ubuntu virtual machine. I set the IP address as 192.168.10.11.

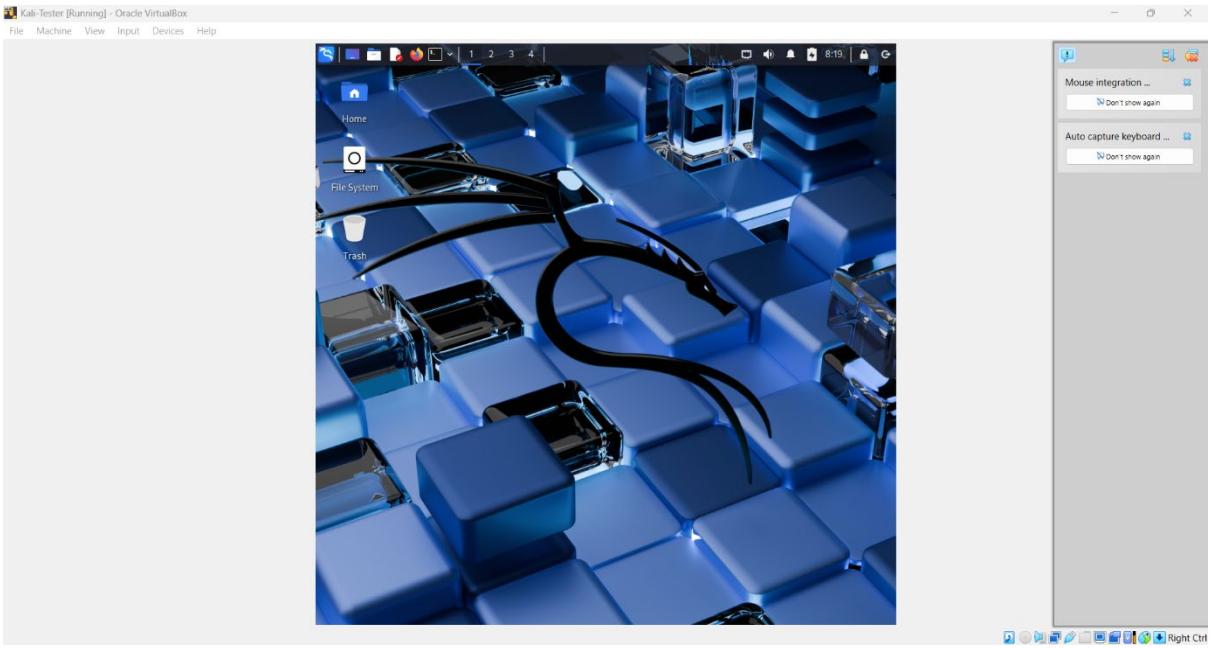

Figure 8

Figure 9 is Kali-Tester virtual machine home screen.

```
jiajan@jiajian-VirtualBox:~$ sudo apt update
[sudo] password for jiajan:
Hit:1 http://my.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://my.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://my.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
203 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Figure 9

Figure 10 is sudo apt update command to refreshes the system's list of available software packages from online repositories.

```
jiajan@jiajian-VirtualBox:~$ sudo apt install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc
  ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 203 not upgraded.
Need to get 80.4 MB of archives.
After this operation, 312 MB of additional disk space will be used.
Get:1 http://my.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1
[65.6 kB]
Get:2 http://my.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.
7.1-1ubuntu2 [33.9 kB]
Get:3 http://my.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd am
d64 1.7.28-0ubuntu1~24.04.1 [38.4 MB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 runc amd64 1.3
```

Figure 10

Figure 11 sudo apt install -y docker.io command is to download the docker.

```
jiajan@jiajian-VirtualBox:~$ sudo systemctl enable docker
jiajan@jiajian-VirtualBox:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: e>
     Active: active (running) since Sun 2025-12-07 18:59:29 +08; 1min 25s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 5961 (dockerd)
      Tasks: 10
     Memory: 20.8M (peak: 21.3M)
        CPU: 333ms
     CGroup: /system.slice/docker.service
             └─5961 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.360>
Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.656>
Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.711>
Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.711>
Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.717>
Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.717>
Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.737>
Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.741>
Dec 07 18:59:29 jiajian-VirtualBox dockerd[5961]: time="2025-12-07T18:59:29.741>
Dec 07 18:59:29 jiajian-VirtualBox systemd[1]: Started docker.service - Docker >
```

Figure 11

Figure 12 sudo systemctl enable docker to start the docker application container engine and sudo systemctl status docker is check whether docker engine is active or not.

```
jiajan@jiajian-VirtualBox:~$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 203 not upgraded.
```

Figure 12

Figure 13 sudo apt install ufw is installed the ufw firewall.

```
jiajan@jiajian-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Figure 13

Figure 14 sudo ufw enable is open the ufw firewall.

```
jiajan@jiajian-VirtualBox:~$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
jiajan@jiajian-VirtualBox:~$ sudo ufw allow 443/tcp
Rule added
Rule added (v6)
```

Figure 14

Figure 15 sudo ufw allow 80 & 443 tcp is only allow required ports.

```
jiajan@jiajian-VirtualBox:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
jiajan@jiajian-VirtualBox:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Figure 15

Figure 16 sudo ufw default deny incoming is deny anything else from outside and sudo ufw default allow outgoing is allow anything else from inside.

```
jiajan@jiajian-VirtualBox:~$ sudo ufw allow from 192.168.10.11 to any port 22
Rule added
jiajan@jiajian-VirtualBox:~$ sudo ufw deny 22/tcp
Rule added
Rule added (v6)
```

Figure 16

Figure 17 sudo ufw allows from 192.168.10.11 to any port 22 and sudo ufw deny 22/tcp is restrict SSH so only Kali-Tester IP addresses can access.

```
jiajan@jiajian-VirtualBox:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW IN    Anywhere
443/tcp                    ALLOW IN    Anywhere
22                         ALLOW IN    192.168.10.11
22/tcp                     DENY IN     Anywhere
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443/tcp (v6)               ALLOW IN    Anywhere (v6)
22/tcp (v6)                DENY IN     Anywhere (v6)
```

Figure 17

Figure 18 sudo ufw status verbose is check firewall status.

```
jiajan@jiajian-VirtualBox:~$ sudo docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
0e4bc2bd6656: Pull complete
b5feb73171bf: Pull complete
108ab8292820: Pull complete
53d743880af4: Pull complete
77fa2eb06317: Pull complete
192e2451f875: Pull complete
de57a609c9d5: Pull complete
Digest: sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c52a42
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

Figure 18

Figure 19 sudo docker pull nginx is pull the open-source web server image.

```
jiajan@jiajian-VirtualBox:~$ sudo docker run -d --name webserver -p 80:80 nginx
20043ec978317e7deec6220d33dbba6c771ac00bd924d3364f7af245189c47cb
```

Figure 19

Figure 20 sudo docker run -d --name webserver -p 80:80 nginx is run container with correct port mapping. After running the container, from Kali browser type http://192.168.10.1 will shows “Welcome to nginx!”, so the server is working.

```
jiajan@jiajian-VirtualBox:~$ sudo docker pull vulnerables/web-dvwa
Using default tag: latest
latest: Pulling from vulnerables/web-dvwa
3e17c6eae66c: Pull complete
0c57df616dbf: Pull complete
eb05d18be401: Pull complete
e9968e5981d2: Pull complete
2cd72dba8257: Pull complete
6cff5f35147f: Pull complete
098cffd43466: Pull complete
b3d64a33242d: Pull complete
Digest: sha256:dae203fe11646a86937bf04db0079adef295f426da68a92b40e3b181f337daa7
Status: Downloaded newer image for vulnerables/web-dvwa:latest
docker.io/vulnerables/web-dvwa:latest
```

Figure 20

Figure 21 sudo docker pull vulnerables/web-dvwa is pull the DVWA docker image.

```
jiajan@jiajian-VirtualBox:~$ sudo docker run -d --name dvwa -p 8080:80 vulnerabl
es/web-dvwa
1d84e9072e264b45c06cbb1df46e906d75143e8d1d42e7509aac7af810362ed9
```

Figure 21

Figure 22 is run DVWA container.

Next, type the sudo apt-get install snort -y to install the snort.

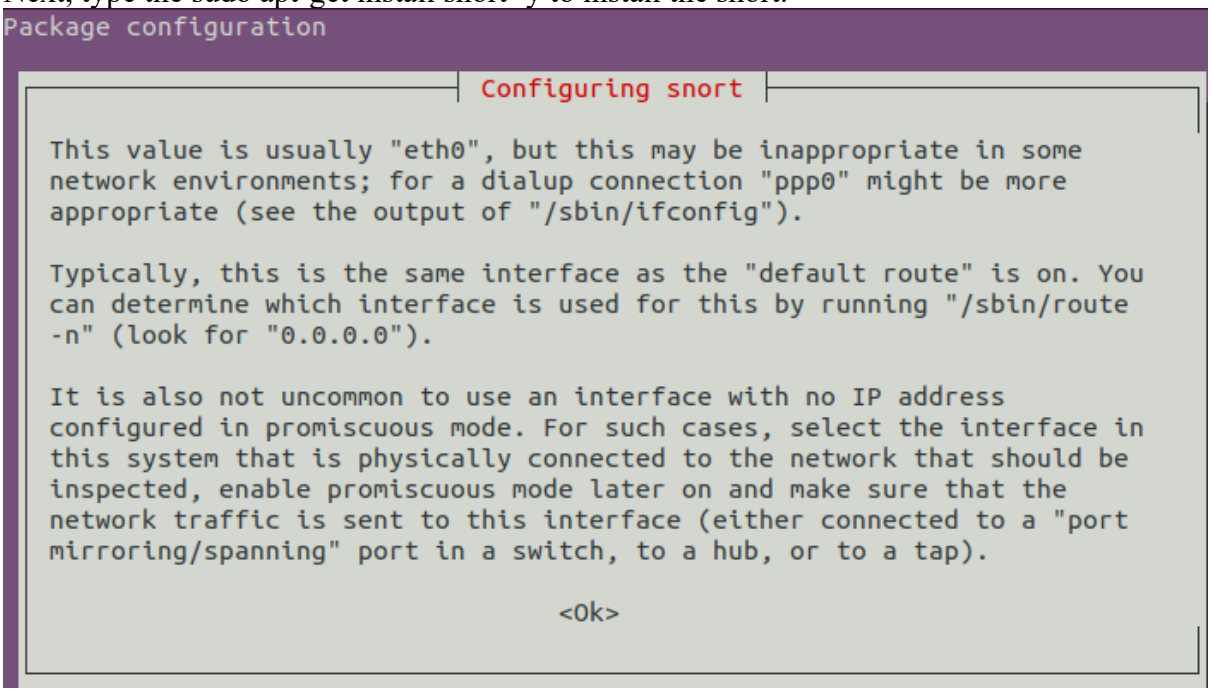

Figure 22

Figure 23 is after type the command, the system shows package configuration window to configure the snort.

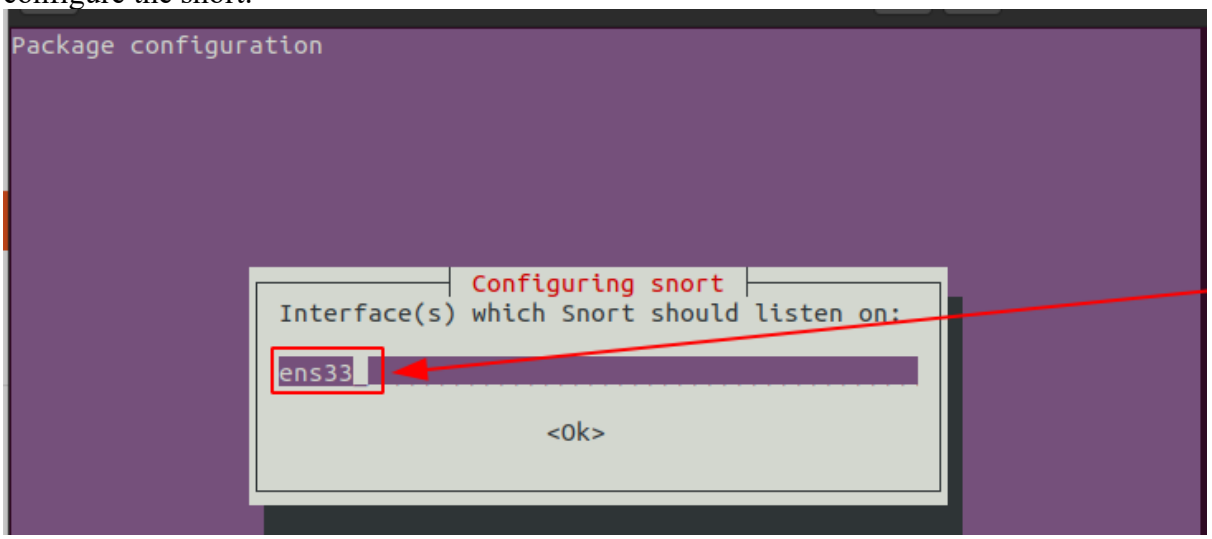

Figure 23

Figure 24 is typing the interface which Snort should listen on, in this part I put the interface as enp0s8.

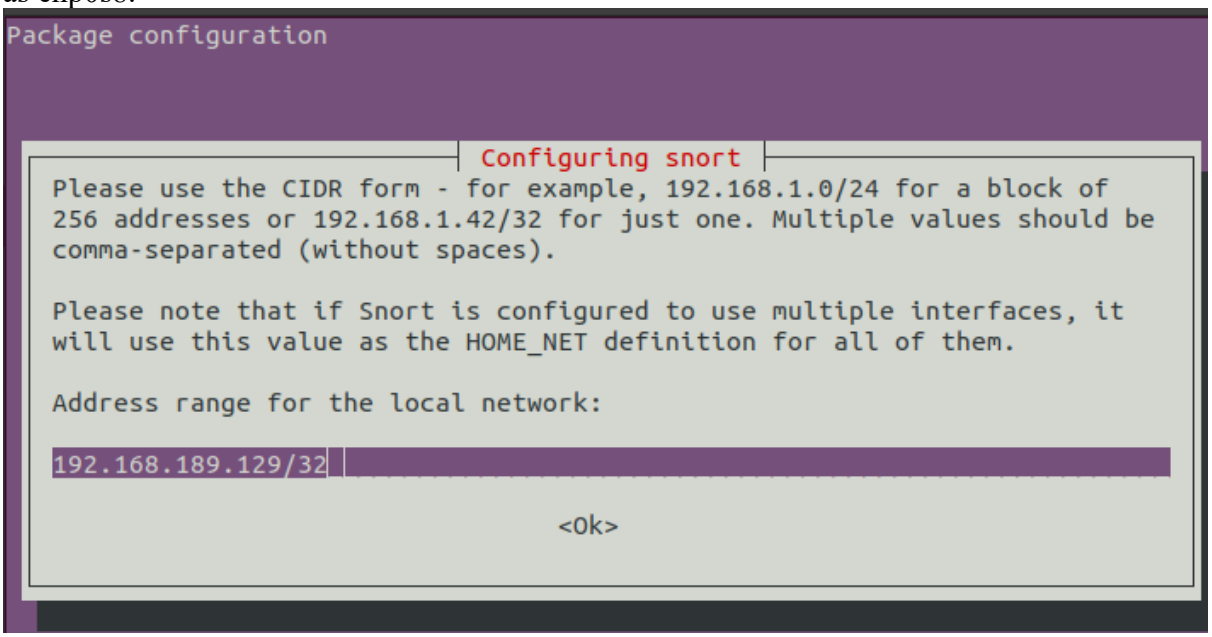

Figure 24

Figure 25 is typing the IP address for the local network, at this part I put the IP address as 192.168.10.1/24.

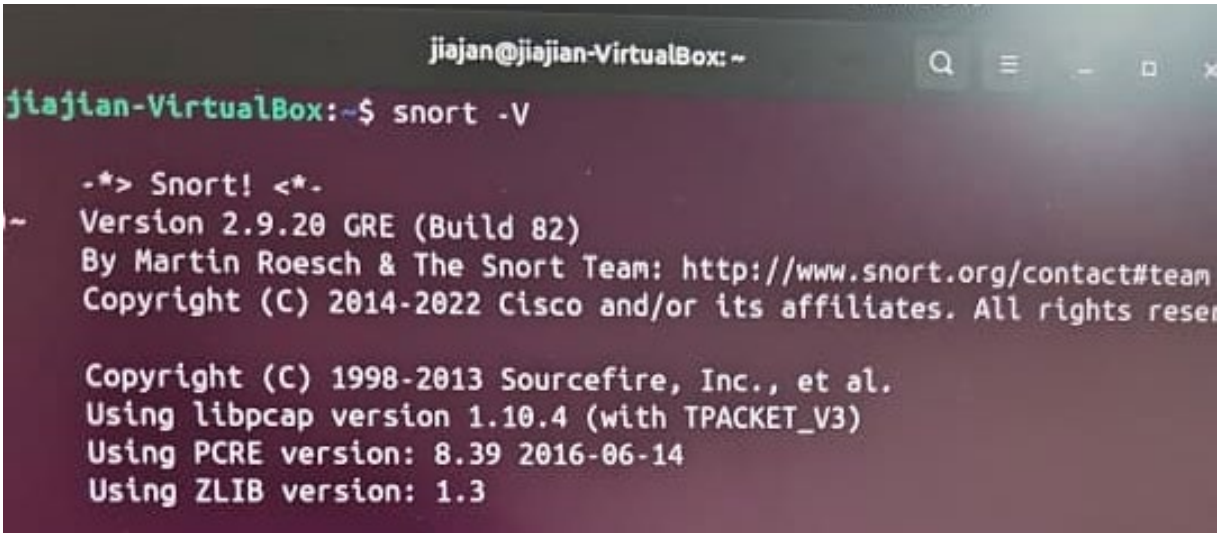

Figure 25

Figure 26 is after setting all the interface and IP address, snort -V is to determine which version of the Snort is installed.

```
jiajan@jiajian-VirtualBox:~$ sudo nano /etc/snort/snort.conf
[sudo] password for jiajan:
```

Figure 26

Figure 27 is a I should type the Ubuntu virtual machine's IP address as a HOME_NET, so I type this command.

```
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 192.168.10.1/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET
```

Figure 27

Figure 28 shows the HOME_NET already set as 192.168.10.1/24.

```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules
```

Figure 28

Figure 29 shows The RULE_PATH variable in the snort.conf file determines the location of the snort rule files.

```
# site specific rules
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:

#include $RULE_PATH/app-detect.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules
include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/experimental.rules
#include $RULE_PATH/exploit-kit.rules
include $RULE_PATH/exploit.rules
```

Figure 29

Figure 30 shows all the community rules.

```
jiajan@jiajian-VirtualBox:~$ sudo snort -T -i ens33 -c /etc/snort/snort.conf
Running in Test mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1741 1830
2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777
7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300
8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002
55555 ]
```

Figure 30

Figure 31 is a command that can test the configuration. If make any mistakes in the configuration file will show error.

```
Total snort Fixed Memory Cost - MaxRss:103956
Snort successfully validated the configuration!
Snort exiting
```

Figure 31

Figure 32 shows if everything is configured correctly, will shows a “Snort successfully validated the configuration!” message.

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.

alert icmp any any -> $HOME_NET any (msg:"ICMP Detection Rule"; sid:1000001;)
alert tcp any any -> $HOME_NET 22 (msg:"SSH Connection Attempts"; sid:1000002)
alert tcp any any -> $HOME_NET 80 (msg:"Command Execution Attempt"; content:"GET"; content:"/etc/passwd"; sid:1000003;)
```

Figure 32

Figure 33 shows I set three alerts, first is generate an alert when an ICMP traffic is received from any source IP Address to Ubuntu. Second is create a snort rule to detect SSH connection attempts and last create a snort rule to detect a command execution attack which contains /etc/passwd in the http GET request.