

Setting static IP for VM (Ubuntu)

Author: Xu Jiachen

Email: liyingxujiachen@gmail.com e0402032@u.nus.edu

Abstract

Because the using a VM in the host with a smaller memory maybe cause the host to run slower. I want to introduce a way to solve this problem. To set a static IP for the VM, then we can use the browser of host to access the web Applications in VM, which are using the web server and have mapped their port to the host port

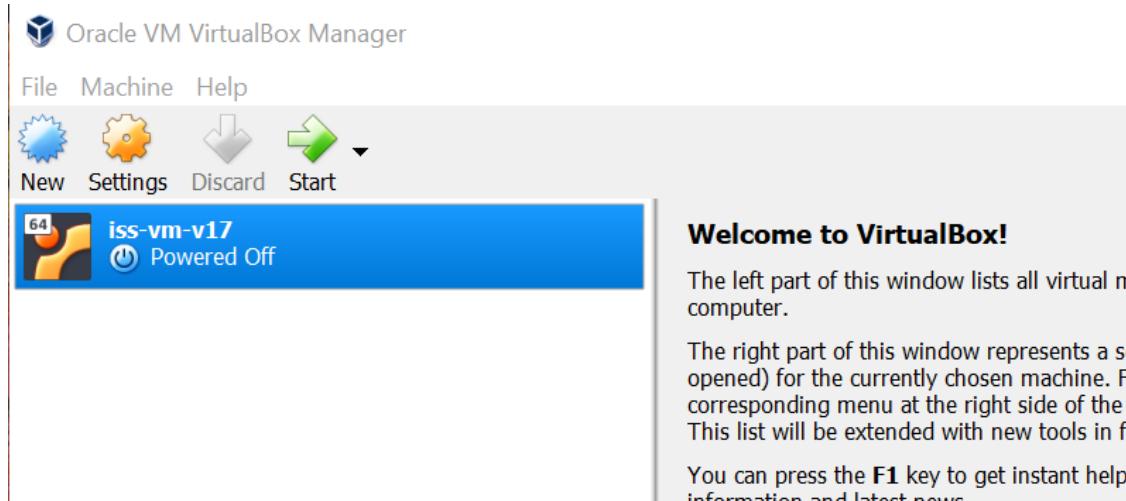
Tools

1. VirtualBox
2. VM (e.g ISS-VM 【Ubuntu】)
 - o account: iss-user
 - o password: iss-user

Procedure

Setting static IP

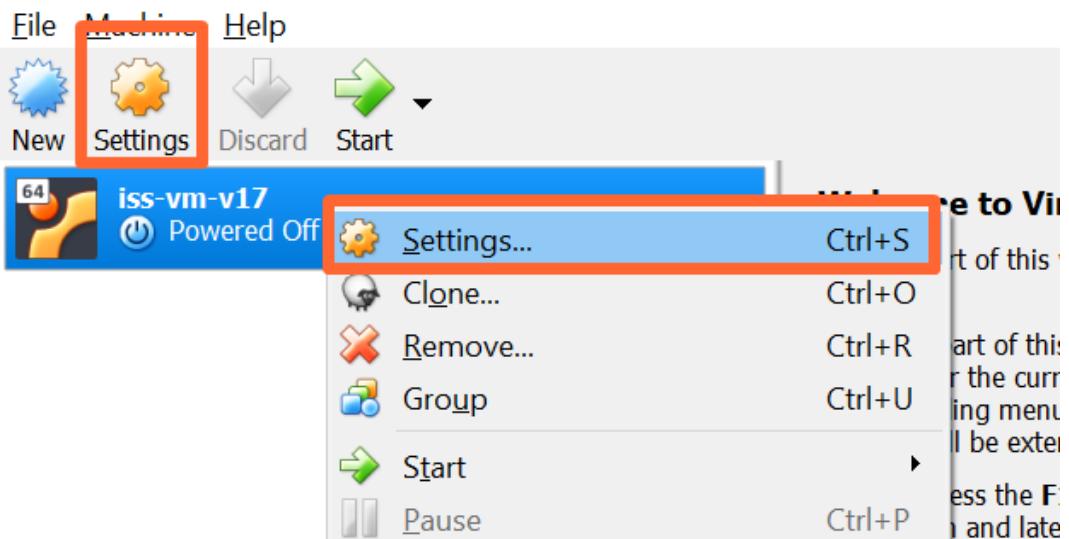
1. Import the VM into VirtualBox



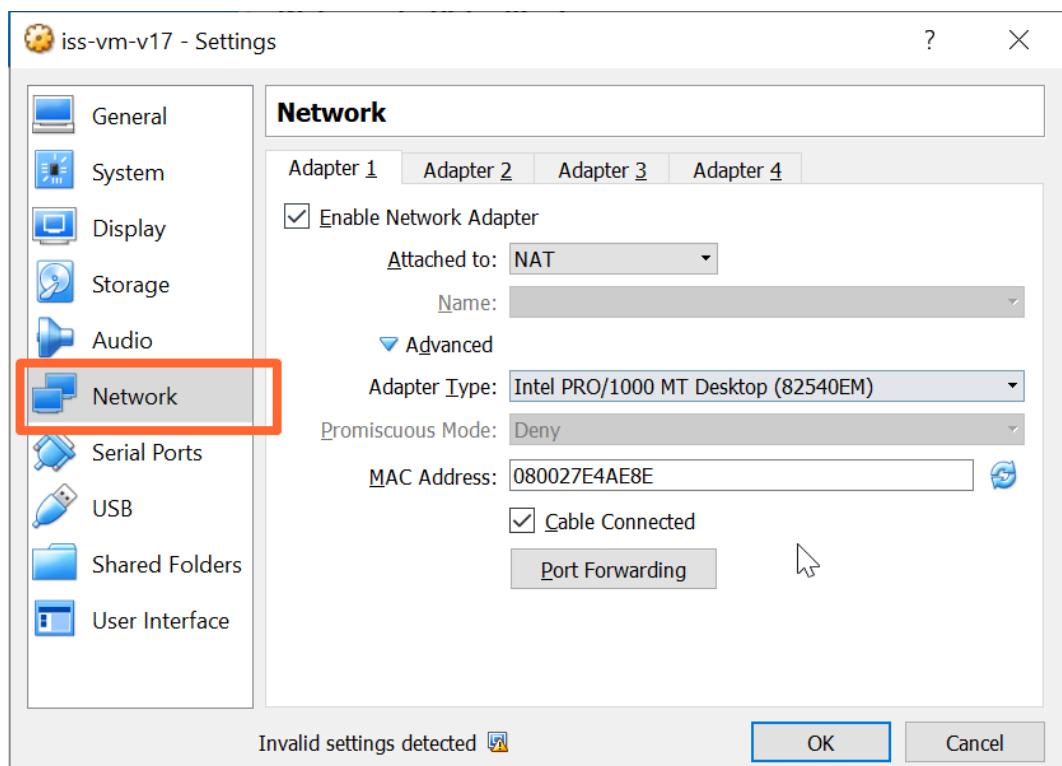
2. (the VM has been shutted down) Modify the Network configuration of the VM, add a host-only Network Adapter to the VM, set a static IP, and retain the original NAT mode adapter to ensure that the VM can access the Internet.

- o Select the VM to be configured, click the Settings button in the toolbar or click the right mouse button and select Settings from the pop-up menu.

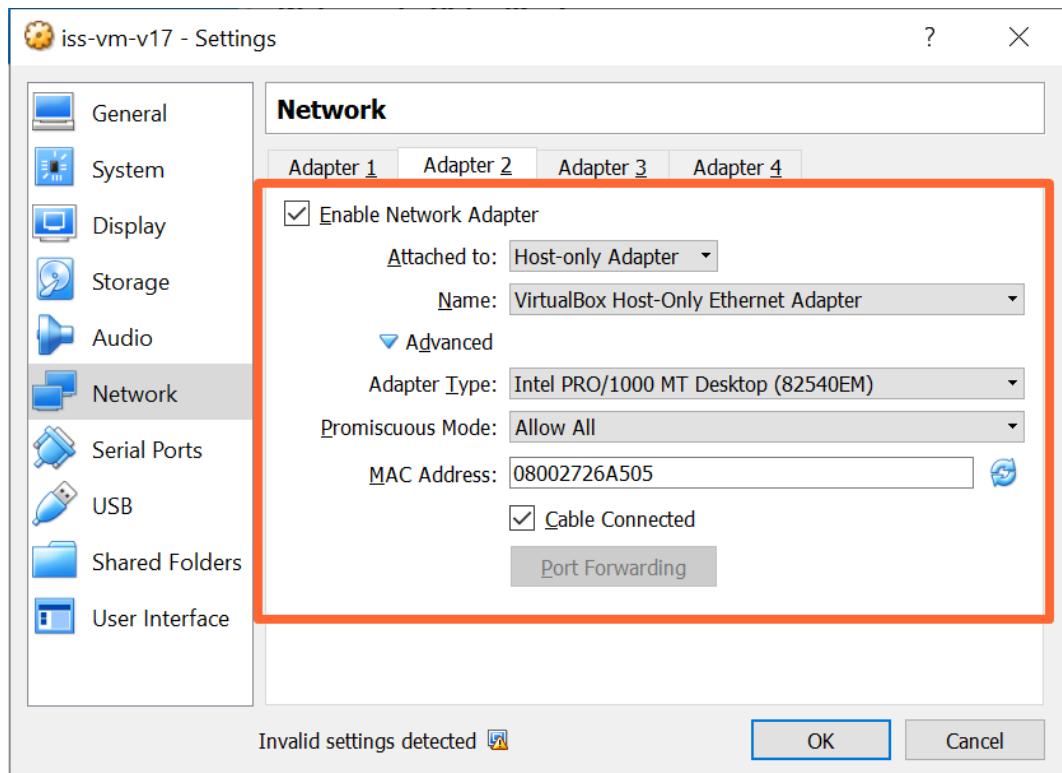
Oracle VM VirtualBox Manager



- Click Network button on the right side of the pop-up panel.



- For the current VM, an Adapter has been configured by default. It is set to NAT mode for connecting to the Internet. If VM does not have an adapter in that mode, please add this adapter for later port mapping.
- Click on the Adapter 2 tab (if the VM has already set the Adapter 2 for other purpose, you can use Adapter 3 or Adapter 4)
- Setting Adapter 2



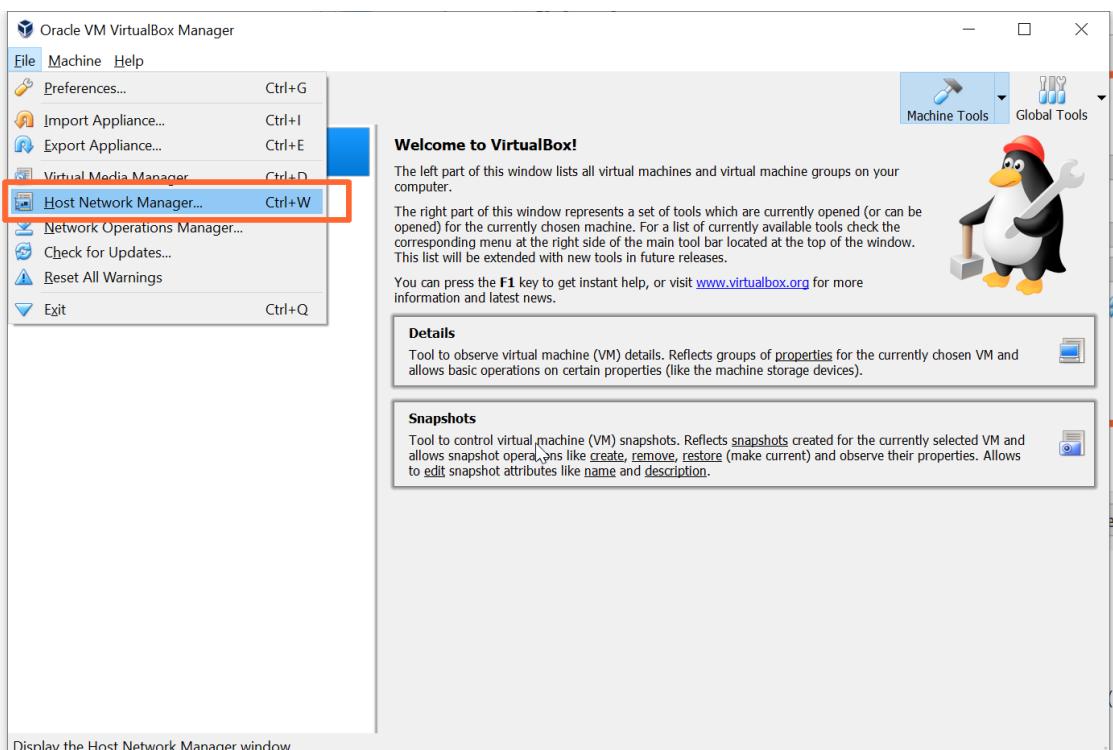
- Enable this Adapter
- Attached to: Host-only Adapter
- Name: VirtualBox Host-Only Ethernet Adapter

For the MAC, you need to add a local virtual network adapter to the global network management interface, and then do the above configuration (*).

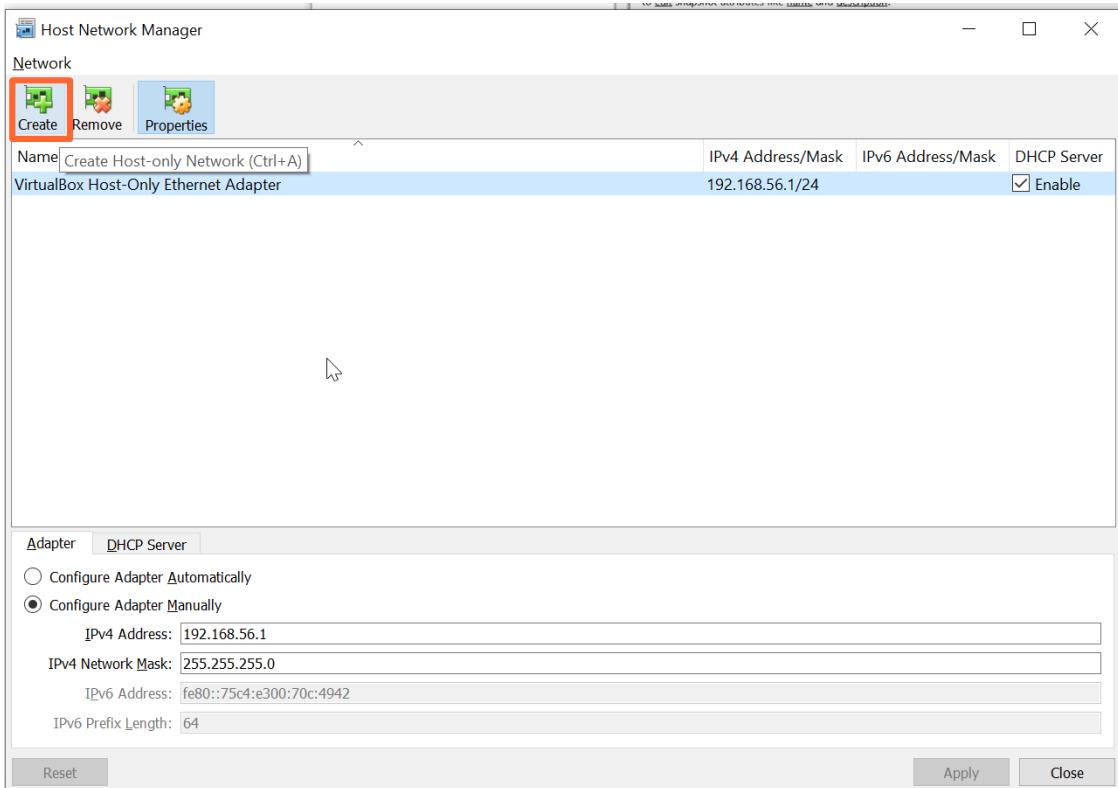
- Promiscuous Mode: Allow All
- Click the OK button

3. (*) Mac may not have Host-only Adapter by default. If not, please follow the steps below to create a local virtualnetwork adapter.

- Click the Files button located in the upper toolbar and select Host Network Manager from the pop-up menu.

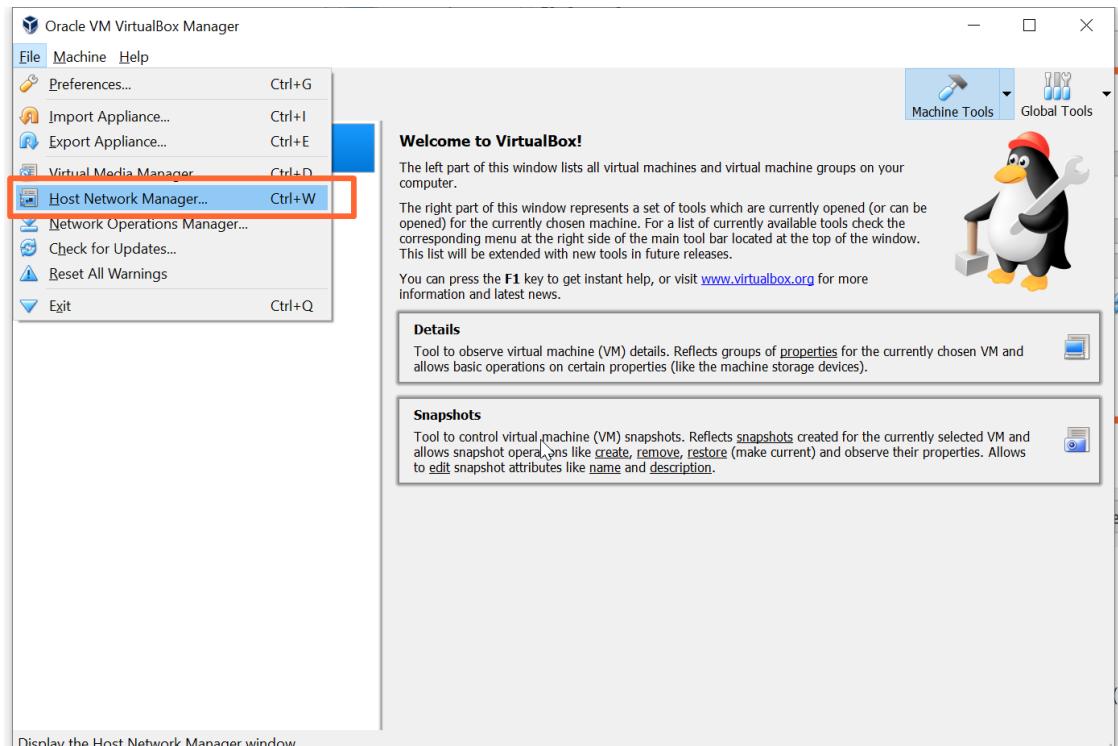


- In the pop-up panel, click the Create button in the upper toolbar to create a network adapter, then open the DHCP server of the network adapter.

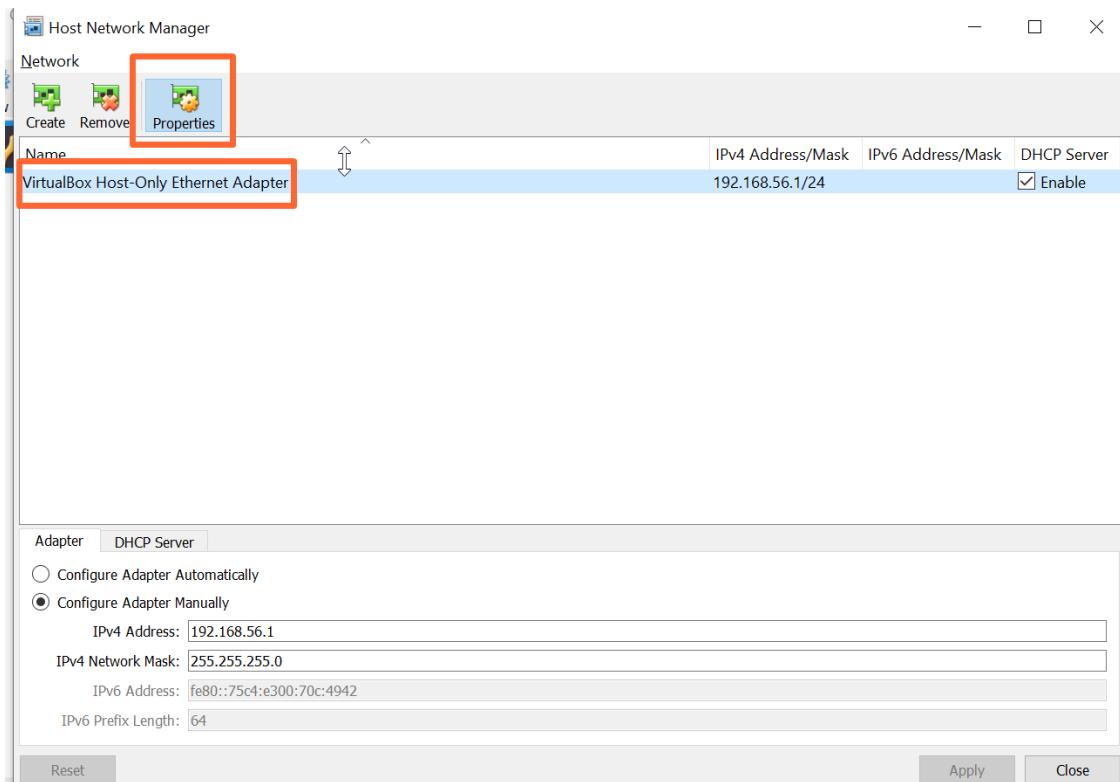


4. Check the range of IP addresses that can be used by the assigned adapter

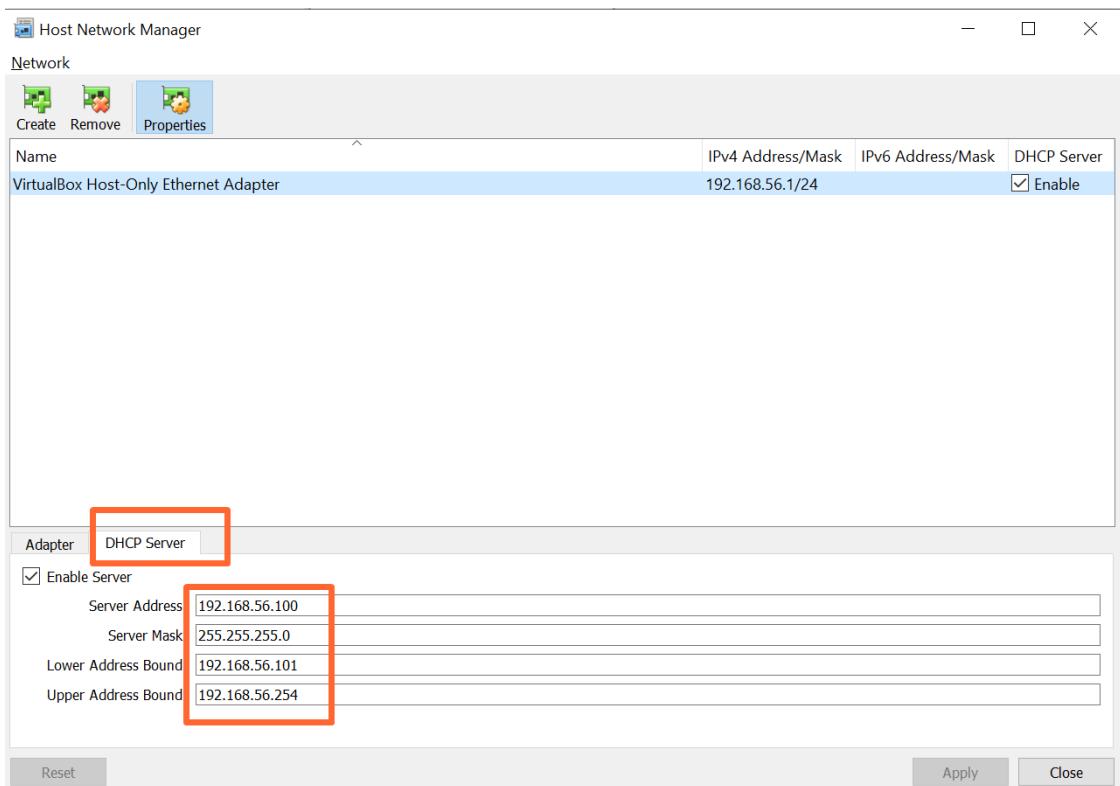
- Click the Files button in the VirtualBox menu bar and select Host Network Manager from the pop-up menu.



- In the pop-up panel, select the network adapter you just assigned to the virtual machine: VirtualBox Host-Only Ethernet Adapter, and click the Properties button located in the upper toolbar to check the properties of the current network adapter.

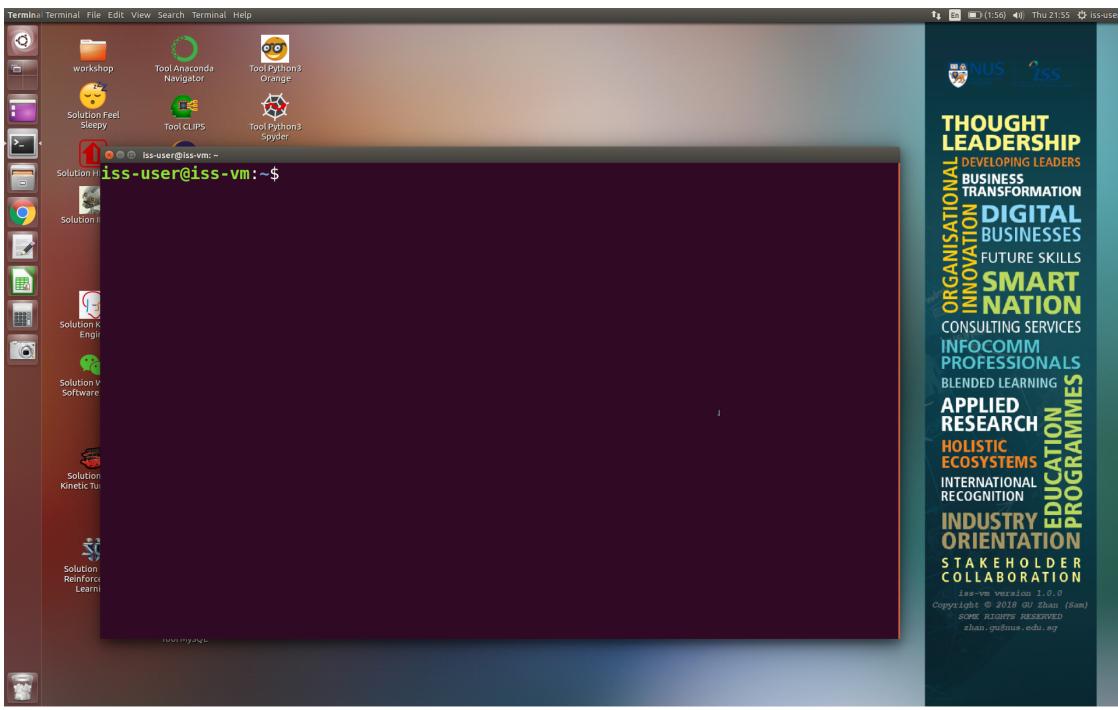


- Click the DHCP Server tab in the property bar to confirm the range of configurable IP addresses for the network adapter.



- Record the following information for setting a static IP address in VM
 - 1. Lower Address Bound: 192.168.56.101
 - 2. Upper Address Bound: 192.168.56.254
 - 3. Server Mask: 255.255.255.0

5. Start the VM and start a terminal window

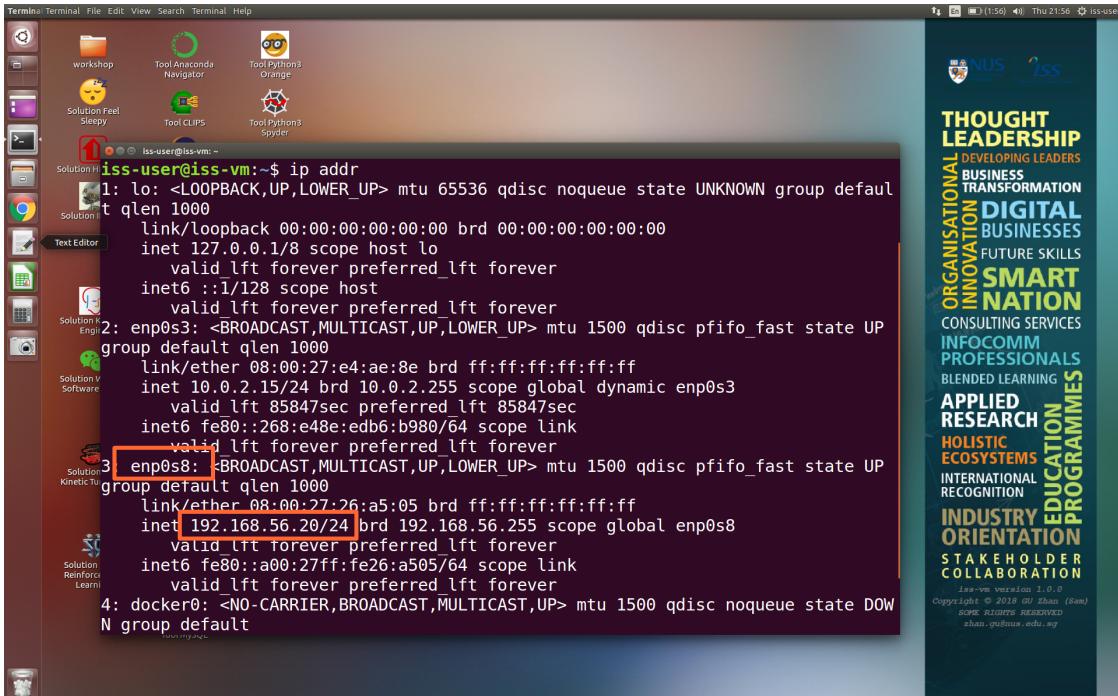


6. Use

```
ip addr
```

to check whether a network adapter has been added. The ip address is 192.168.56.X , then record the name of this network adapter.

192.168.56.101<IP of VM<192.168.56.254



7. Set a static IP address

- Use

```
sudo nano /etc/network/interfaces
```

to modify the configuration file of network adapters, add the following content

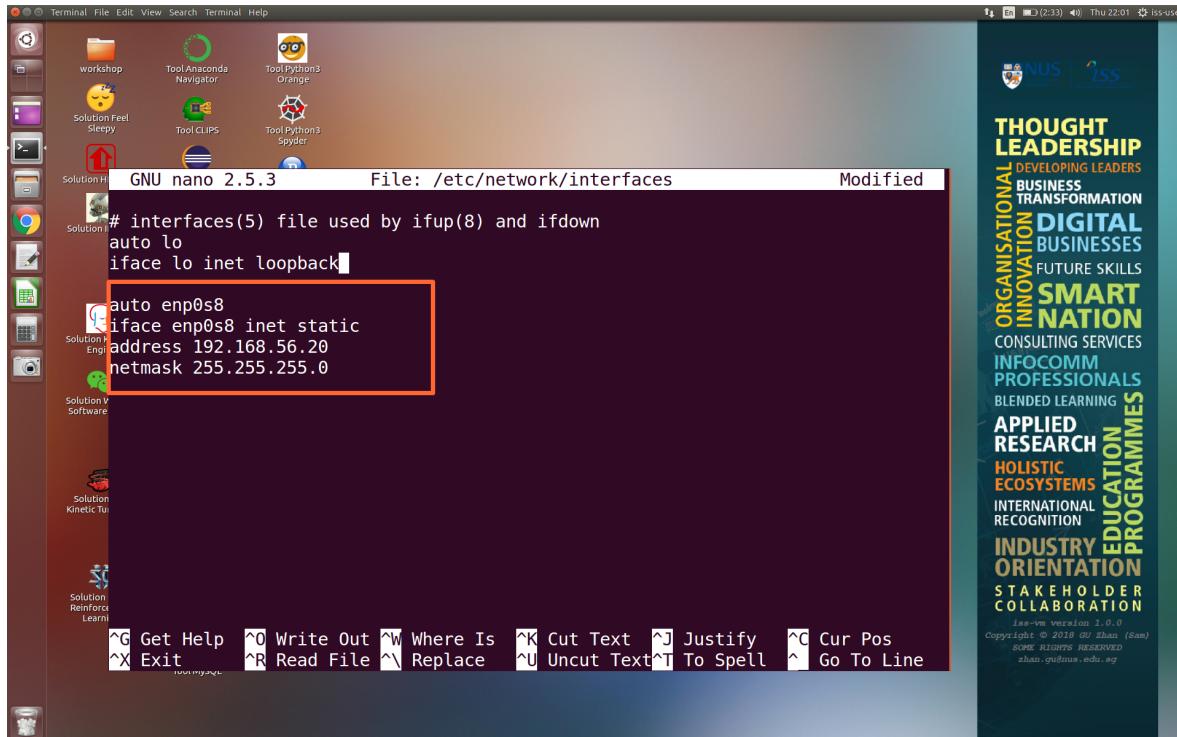
```
auto enp0s8

iface enp0s8 inet static

address 192.168.56.20

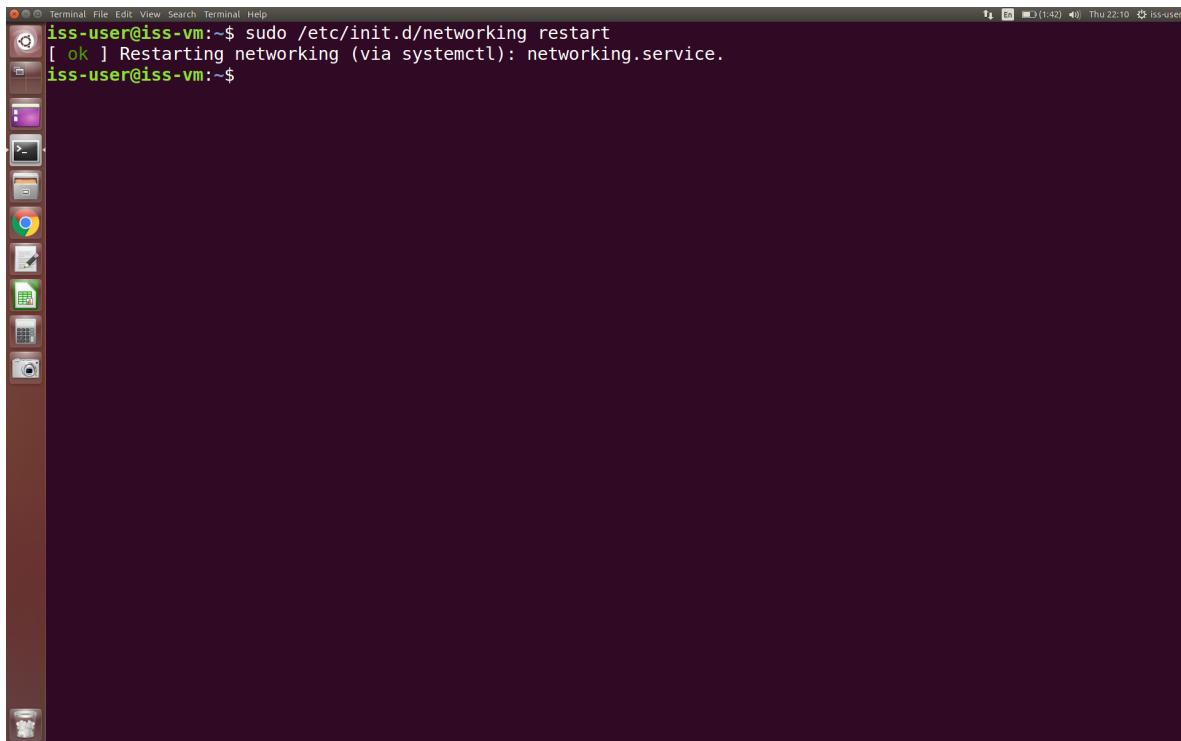
netmask 255.255.255.0
```

- The address here must be within the range of the previously configurable IP address of the network adapter (192.168.56.101--192.168.56.254)
- Save changes and exit the current file with **ctrl+x** and **y**



8. Restart the network with

```
sudo /etc/init.d/networking restart
```

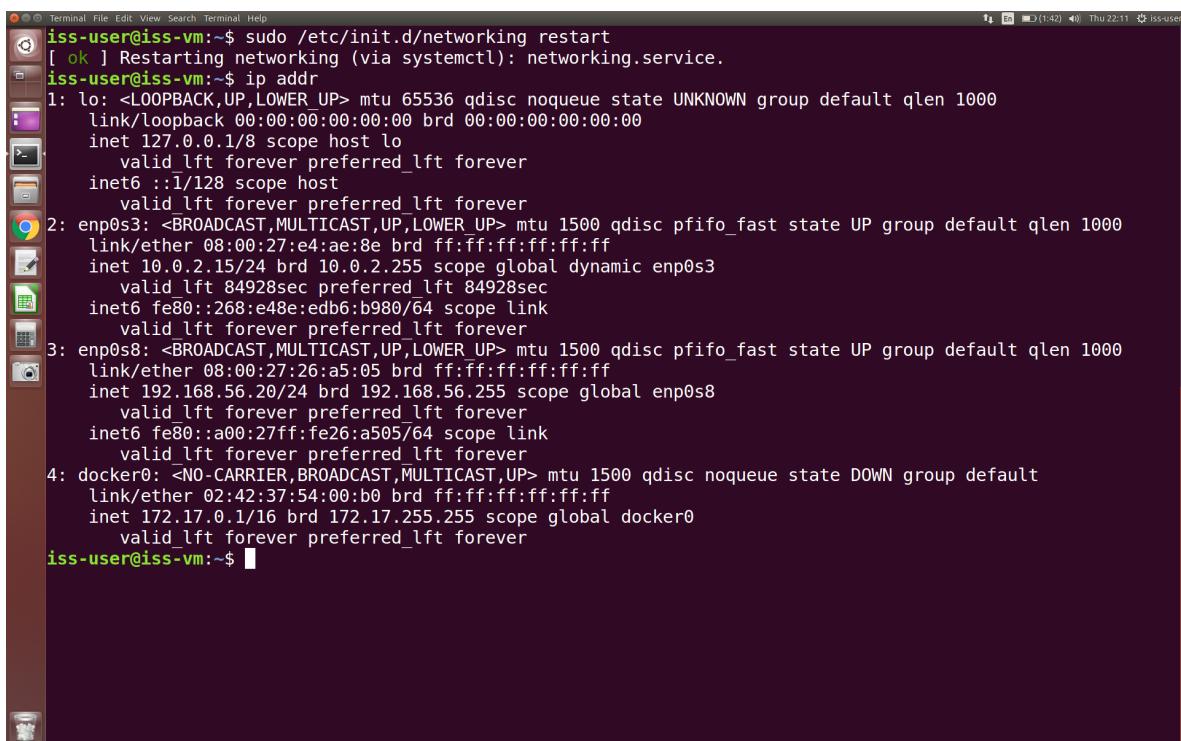
A screenshot of an Ubuntu desktop environment. On the left is a brown vertical dock containing icons for Dash, Home, Applications, and other system tools. A terminal window is open at the top, showing the command `sudo /etc/init.d/networking restart` being run and its success message. The desktop background is dark grey.

```
iss-user@iss-vm:~$ sudo /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
iss-user@iss-vm:~$
```

9. Use

```
ip addr
```

to check if the current static ip is configured.

A screenshot of an Ubuntu desktop environment showing the output of the `ip addr` command in a terminal window. The terminal shows the configuration for several network interfaces, including `lo`, `enp0s3`, `enp0s8`, and `docker0`. The `lo` interface has a static IP of `127.0.0.1/8`. The `enp0s3` and `enp0s8` interfaces have dynamic IP addresses assigned by DHCP. The `docker0` interface is shown as having a static IP of `172.17.0.1/16`.

```
iss-user@iss-vm:~$ sudo /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
iss-user@iss-vm:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:e4:ae:8e brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
            valid_lft 84928sec preferred_lft 84928sec
        inet6 fe80::2e4:aeff:fe00:64 scope link
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:26:a5:05 brd ff:ff:ff:ff:ff:ff
        inet 192.168.56.20/24 brd 192.168.56.255 scope global enp0s8
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe26:a505/64 scope link
            valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:37:54:00:b0 brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
iss-user@iss-vm:~$
```

10. Use the `ping 192.168.56.X` command on the host to see if you can ping the virtual machine.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.615]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\liyin>ping 192.168.56.20

Pinging 192.168.56.20 with 32 bytes of data:
Reply from 192.168.56.20: bytes=32 time=1ms TTL=64
Reply from 192.168.56.20: bytes=32 time<1ms TTL=64
Reply from 192.168.56.20: bytes=32 time<1ms TTL=64
Reply from 192.168.56.20: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

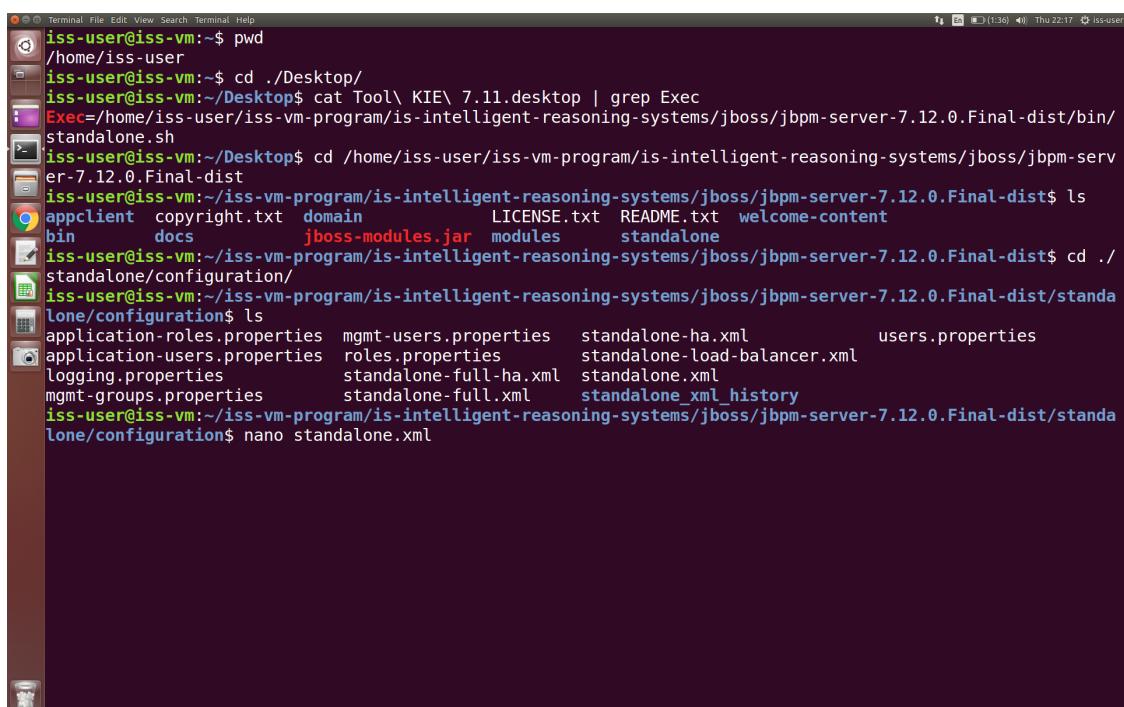
C:\Users\liyin>

```

11. VM static IP configuration is complete (The configured IP address may be the same as the original address assigned by DHCP Server, but after the static IP address is configured, the IP address of the VM will not be changed with the restart of the VM)

Testing with KIE

1. Modify the JBOSS configuration in KIE (because JBOSS can only be accessed from localhost by default, if you need to be accessed by the entire LAN, you need to modify its configuration file)
 - o Find the JBOSS configuration file in KIE



```

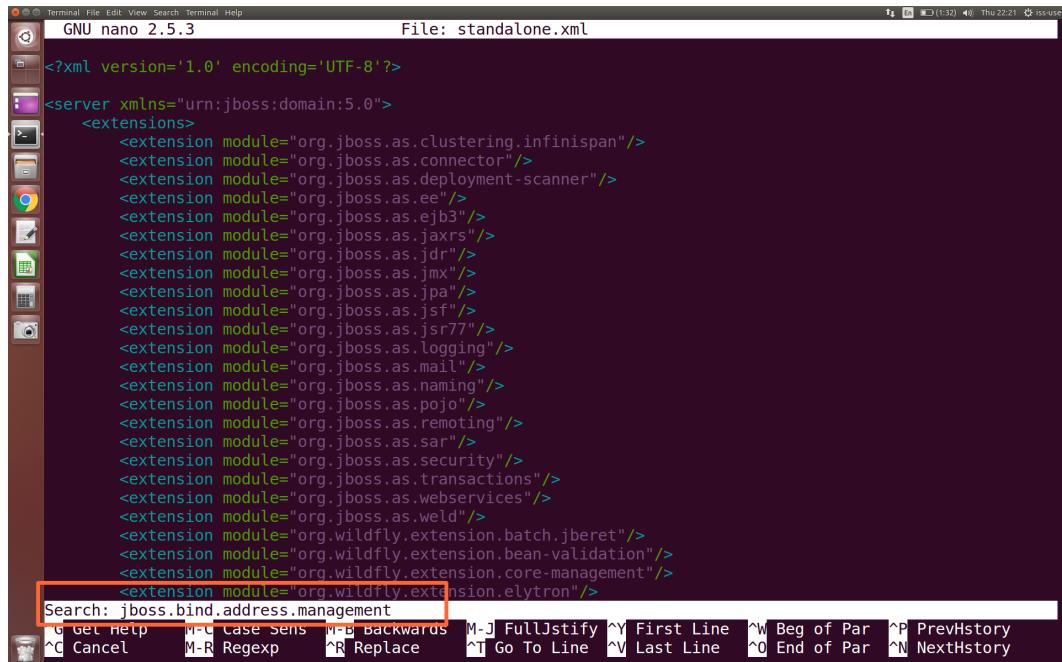
iss-user@iss-vm:~$ pwd
/home/iss-user
iss-user@iss-vm:~$ cd ./Desktop/
iss-user@iss-vm:~/Desktop$ cat Tool\ KIE\ 7.11.desktop | grep Exec
Exec=/home/iss-user/iss-vm-program/is-intelligent-reasoning-systems/jboss/jbpm-server-7.12.0.Final-dist/bin/
standalone.sh
iss-user@iss-vm:~/Desktop$ cd /home/iss-user/iss-vm-program/is-intelligent-reasoning-systems/jboss/jbpm-serv
er-7.12.0.Final-dist
iss-user@iss-vm:~/iss-vm-program/is-intelligent-reasoning-systems/jboss/jbpm-server-7.12.0.Final-dist$ ls
appclient  copyright.txt  domain           LICENSE.txt  README.txt  welcome-content
bin        docs            jboss-modules.jar  modules     standalone
iss-user@iss-vm:~/iss-vm-program/is-intelligent-reasoning-systems/jboss/jbpm-server-7.12.0.Final-dist$ cd ..
standalone/configuration/
iss-user@iss-vm:~/iss-vm-program/is-intelligent-reasoning-systems/jboss/jbpm-server-7.12.0.Final-dist/standa
lone/configuration$ ls
application-roles.properties  mgmt-users.properties  standalone-ha.xml          users.properties
application-users.properties  roles.properties       standalone-load-balancer.xml
logging.properties             standalone-full-ha.xml  standalone.xml
mgmt-groups.properties        standalone-full.xml   standalone_xml_history
iss-user@iss-vm:~/iss-vm-program/is-intelligent-reasoning-systems/jboss/jbpm-server-7.12.0.Final-dist/standa
lone/configuration$ nano standalone.xml

```

- o Modify the configuration file with

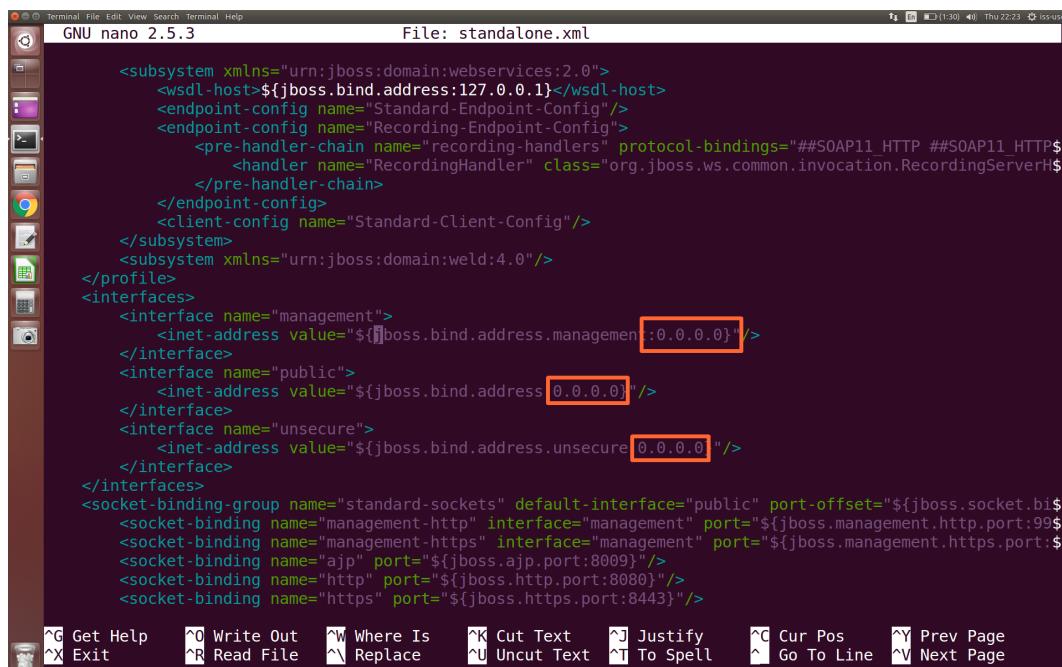
```
nano standalone.xml
```

- What needs to be modified is the IP address when JBOSS starts using standalone mode.
- Use Ctrl+W to search jboss.bind.address.management in the nano editor



```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns="urn:jboss:domain:5.0">
  <extensions>
    <extension module="org.jboss.as.clustering.infinispan"/>
    <extension module="org.jboss.as.connector"/>
    <extension module="org.jboss.as.deployment-scanner"/>
    <extension module="org.jboss.as.ee"/>
    <extension module="org.jboss.as.ejb3"/>
    <extension module="org.jboss.as.jaxrs"/>
    <extension module="org.jboss.as.jdr"/>
    <extension module="org.jboss.as.jmx"/>
    <extension module="org.jboss.as.jpa"/>
    <extension module="org.jboss.as.jsf"/>
    <extension module="org.jboss.as.jsr77"/>
    <extension module="org.jboss.as.logging"/>
    <extension module="org.jboss.as.naming"/>
    <extension module="org.jboss.as.pojo"/>
    <extension module="org.jboss.as.remoting"/>
    <extension module="org.jboss.as.sar"/>
    <extension module="org.jboss.as.security"/>
    <extension module="org.jboss.as.transactions"/>
    <extension module="org.jboss.as.webservices"/>
    <extension module="org.jboss.as.weld"/>
    <extension module="org.wildfly.extension.batch.jberet"/>
    <extension module="org.wildfly.extension.bean-validation"/>
    <extension module="org.wildfly.extension.core-management"/>
    <extension module="org.wildfly.extension.elytron"/>
  </extensions>
  Search: jboss.bind.address.management
  <!-- Get Help M-U Case Sensitive M-B Backwards M-J FullJustify ^Y First Line ^W Beg of Par ^P PrevHistory
  ^C Cancel M-R Regexp ^R Replace ^T Go To Line ^V Last Line ^O End of Par ^N NextHistory -->
```

- Modify the three IP addresses circled in the figure below to change them to **0.0.0.0**

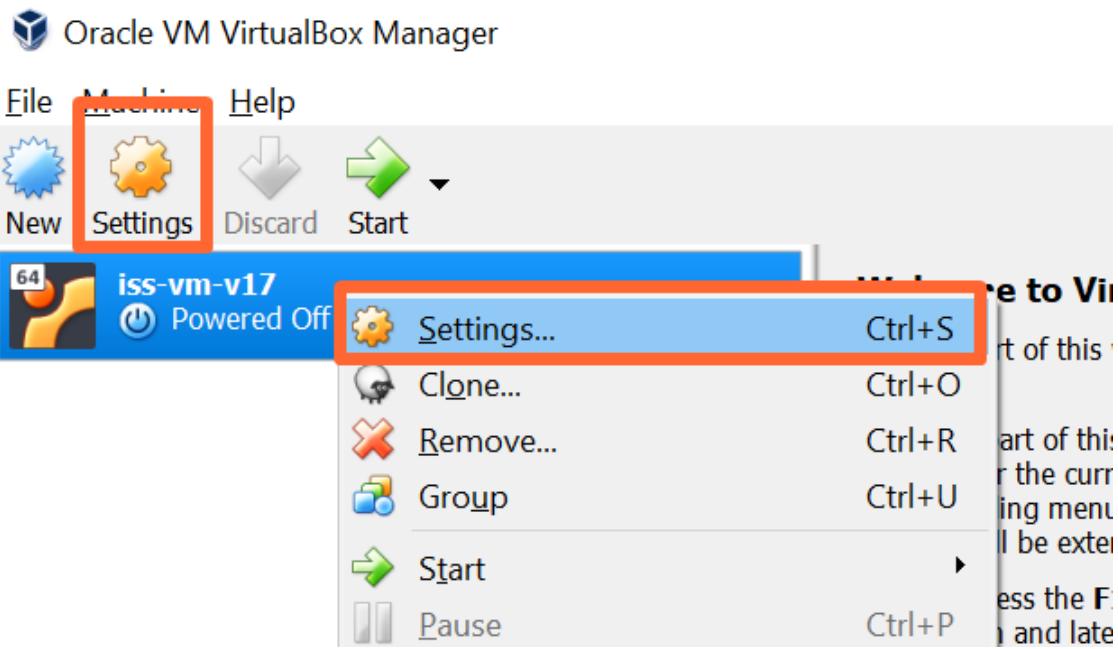


```
<subsystem xmlns="urn:jboss:domain:webservices:2.0">
  <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
  <endpoint-config name="Standard-Endpoint-Config"/>
  <endpoint-config name="Recording-Endpoint-Config">
    <pre-handler-chain name="recording-handlers" protocol-bindings="##SOAP11_HTTP ##SOAP11_HTTPS">
      <handler name="RecordingHandler" class="org.jboss.ws.common.invocation.RecordingServerHandler"/>
    </pre-handler-chain>
  </endpoint-config>
  <client-config name="Standard-Client-Config"/>
</subsystem>
<subsystem xmlns="urn:jboss:domain:weld:4.0"/>
</profile>
<interfaces>
  <interface name="management">
    <inet-address value="${jboss.bind.address.management}:0.0.0.0"/>
  </interface>
  <interface name="public">
    <inet-address value="${jboss.bind.address[0]}:0.0.0.0"/>
  </interface>
  <interface name="unsecure">
    <inet-address value="${jboss.bind.address.unsecure}0.0.0.0"/>
  </interface>
</interfaces>
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.binding.port-offset}">
  <socket-binding name="management-http" interface="management" port="${jboss.management.http.port:9990}"/>
  <socket-binding name="management-https" interface="management" port="${jboss.management.https.port:9993}"/>
  <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>
  <socket-binding name="http" port="${jboss.http.port:8080}"/>
  <socket-binding name="https" port="${jboss.https.port:8443}"/>
</socket-binding-group>
<!-- Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
  ^X Exit ^R Read File ^A Replace ^U Uncut Text ^T To Spell ^G Go To Line ^V Next Page -->
```

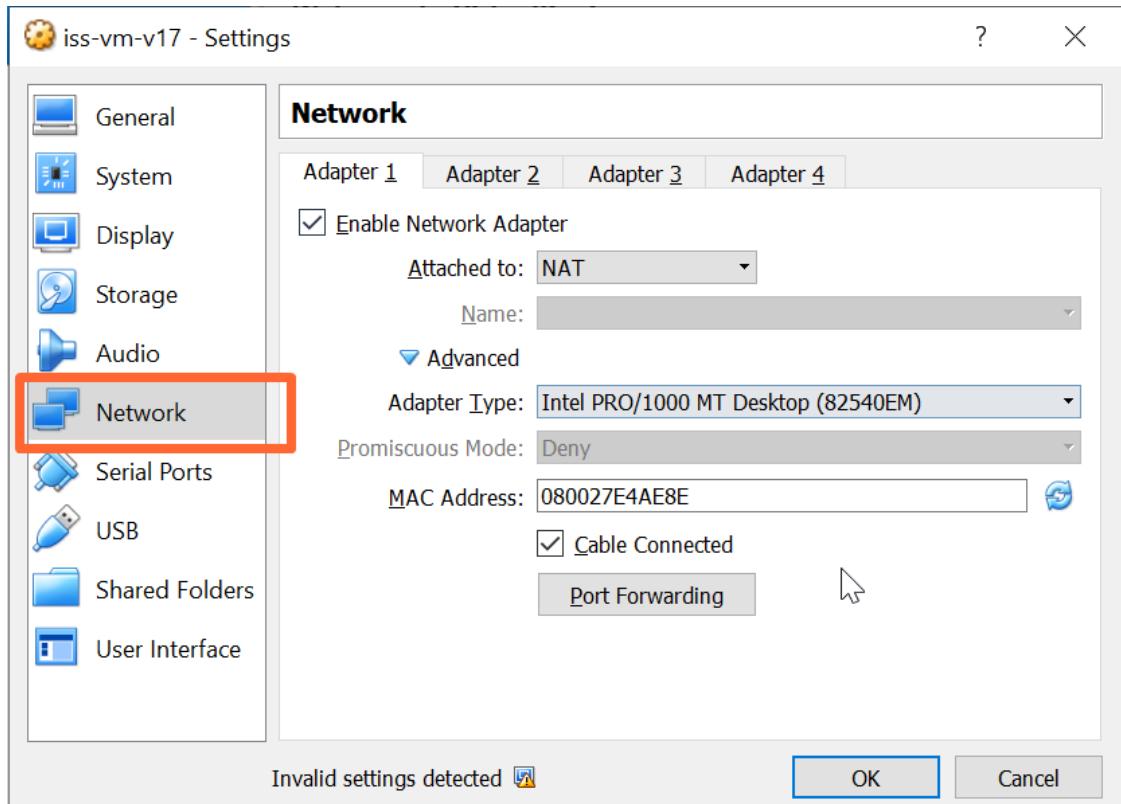
- Save the modified and current file with **ctrl+x** and **y**

2. Modify the port mapping in the virtual machine settings, and map the default port 8080 of JBOSS to port 8080 of the host (you can map different ports)

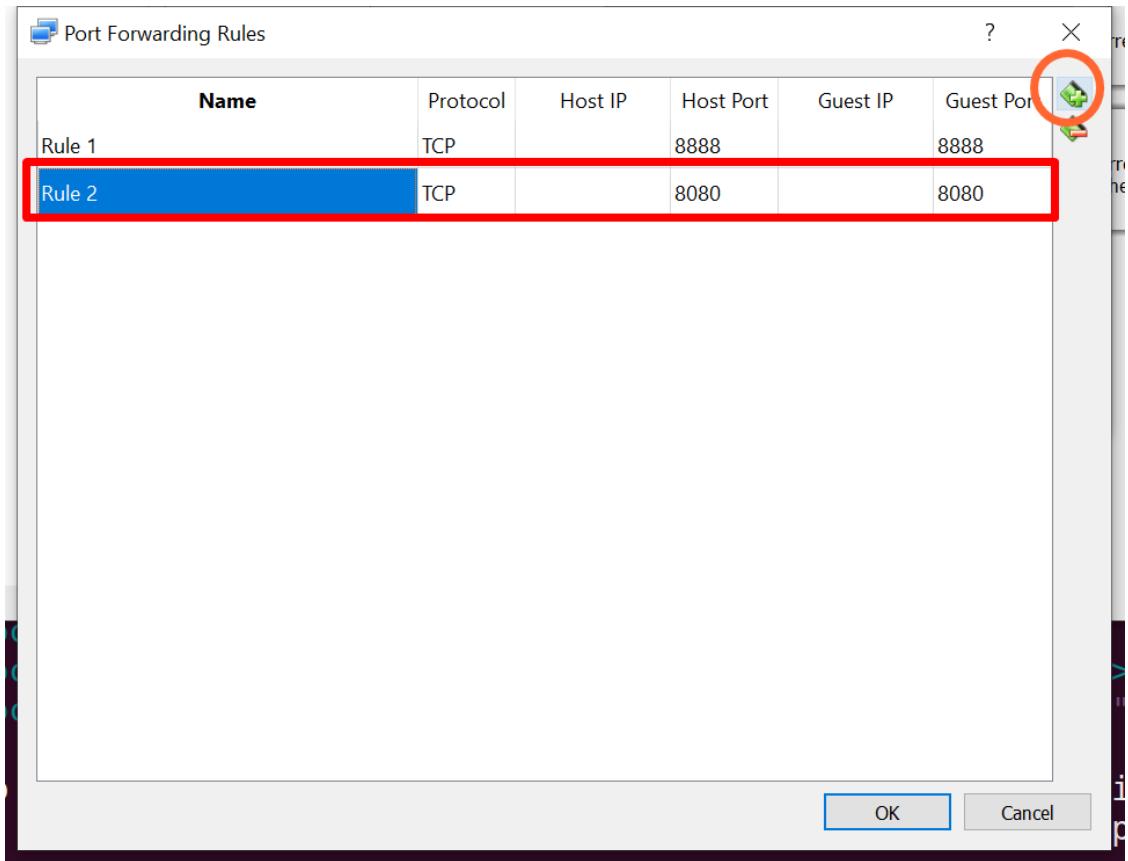
- Select the virtual machine to be configured, click the Settings button in the toolbar or select the virtual machine to be configured. Click the right mouse button and select Settings from the pop-up menu.



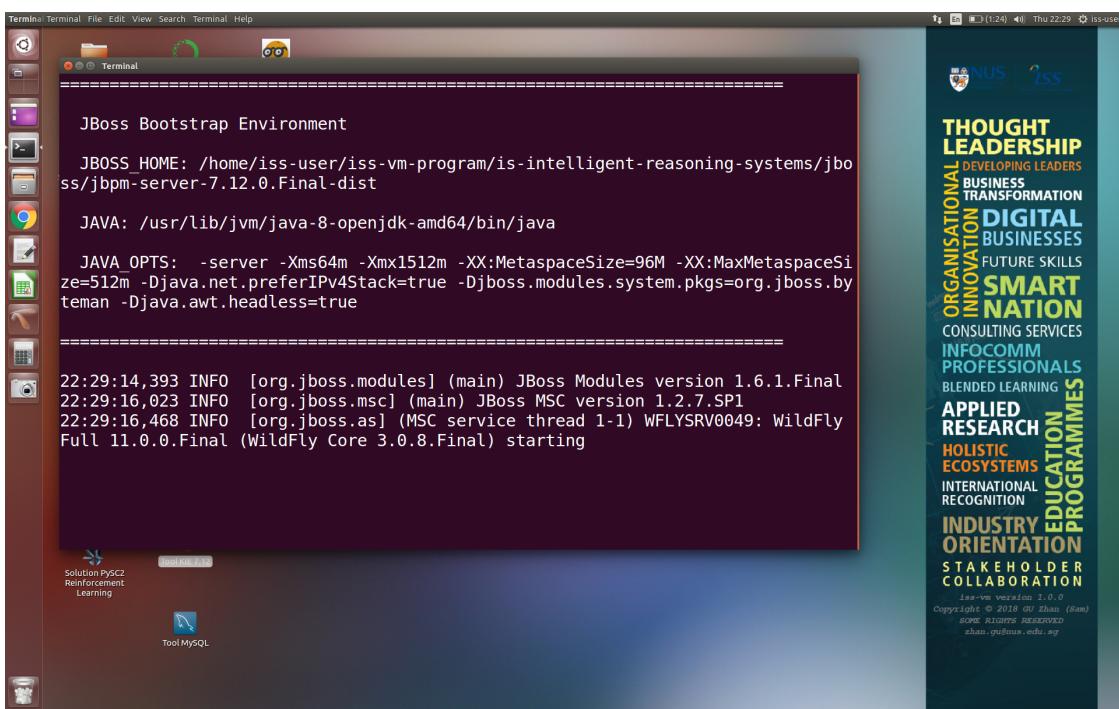
- Click Network on the right side of the pop-up panel.



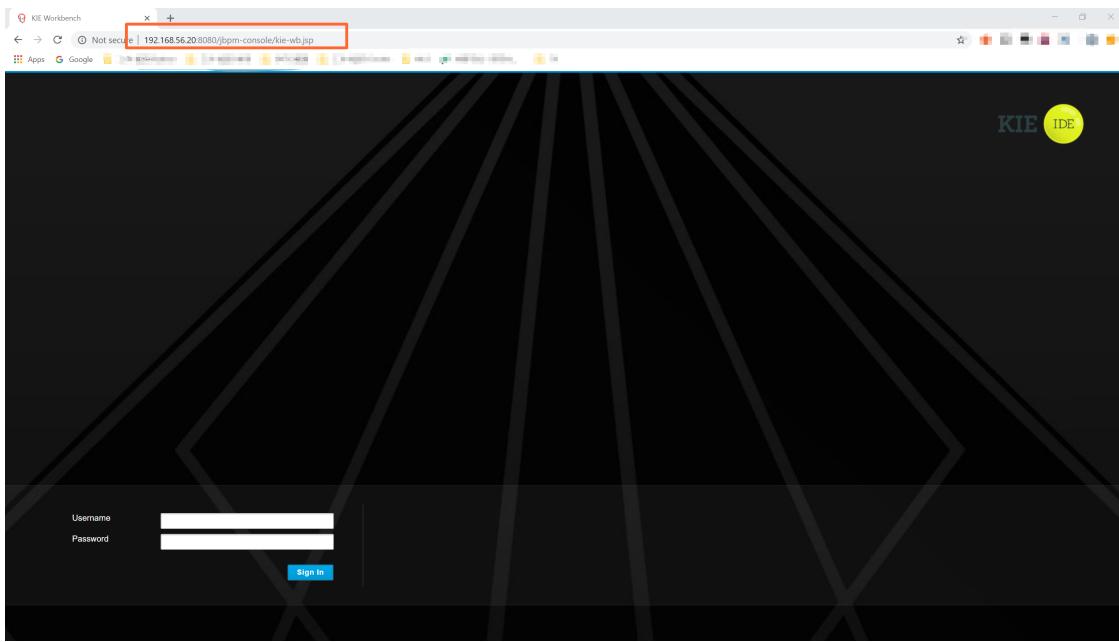
- In the network adapter using NAT mode, click the Port Forwarding button.
- Add a port mapping rule 8080 --- 8080 in the pop-up panel (the host mapping port could not conflict with the port occupied by the host's existing service)



- Launch the KIE shortcut located on the VM desktop



- Wait for the boot to complete, use the previously set IP and mapped port on the host to access (<http://192.168.56.20:8080/jbpm-console/>)



- Wbadmin login is successful, the test ends

