

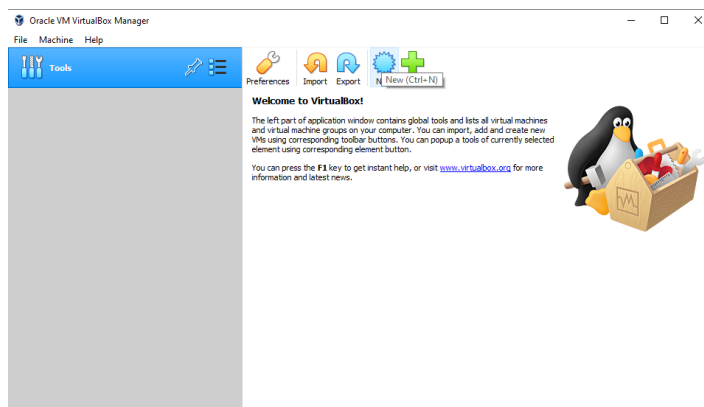
How to build OpenCore and OpenRAN for 5G SA & 5G NSA using Virtualbox VM on Linux Ubuntu OS

● Open5GS

a. Virtualbox preparation

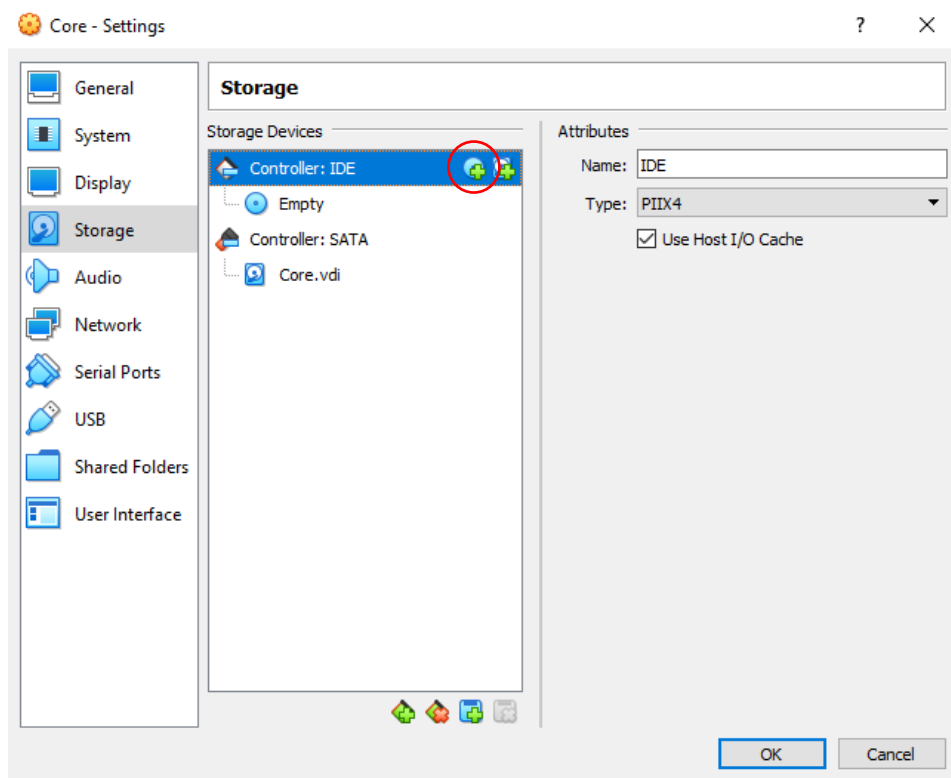
First, you need to install virtualbox and download linux Ubuntu version 18.04.6 [We use version because when this doc is created some support application only support until Linux Ubuntu version 18.04.6] LTS (Bionic Beaver) go to this link: [Ubuntu 18.04.6 LTS \(Bionic Beaver\)](https://ubuntu.com/download/server). You can choose desktop or server, in this case I use desktop image.

b. VM preparation



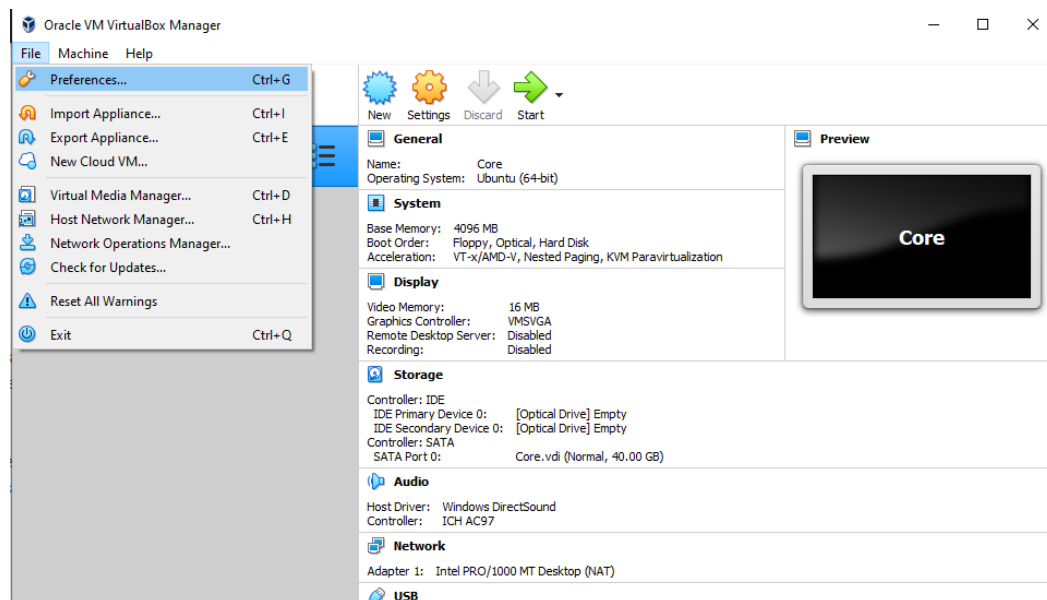
Click new, then a pop up will appear. You can name it as you want, because this VM will be used for core, I will name it **core**. For the machine folder you can choose where you want to save this VM files (it is recommend to use disk with memory with free space about 210Gbps for all this VM or if only for core you can prepare space about 70 Gbps). Then for OS please choose **Linux Ubuntu with 64 bit**. Next for RAM it's recommend to set minimum 4Gb. For all next option you can just follow the default choices. Memory minimum can be set 70 Gb. Then click create. Nice!, now you have created a VM. Double click the VM to turn it on, choose Linux Ubuntu image that you have download. For the next step just follow the procedure for Ubuntu Installation.

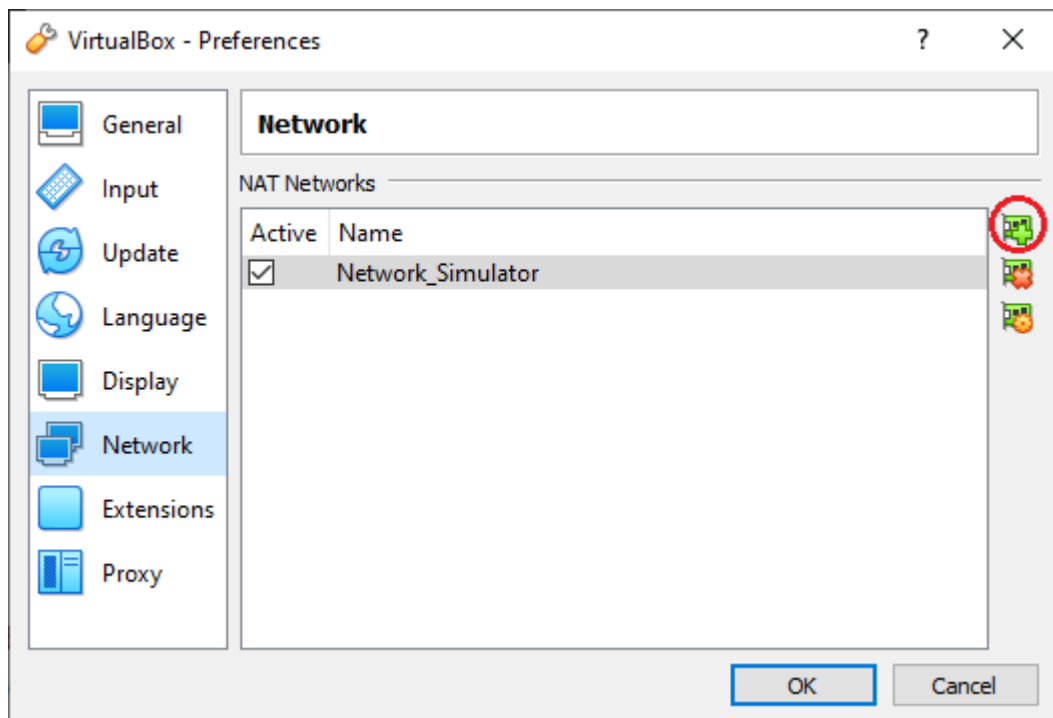
If you miss the OS installation part, you can manually choose from Settings > Storage > click the red circle > click Add, to add Ubuntu 64 image > click ok > turn on the VM. After this you can start the Linux Ubuntu Installation.



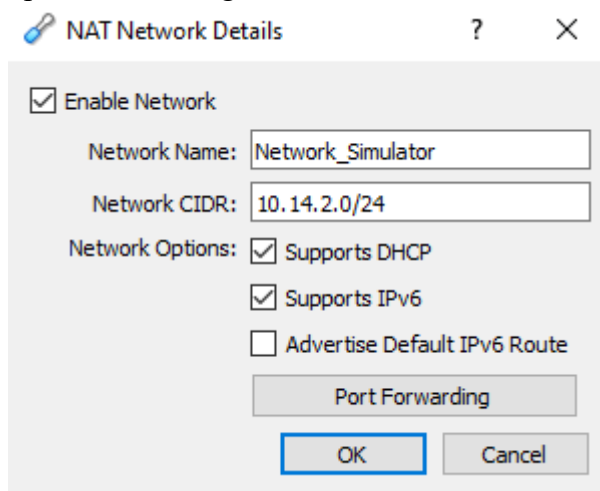
1. Network Setting for VMs

- a) Please make sure all VMs already turned down.
- b) Go to global setting in Virtualbox, click File > Preferences > Network and add a new NAT network > Click symbol on red circle to create it

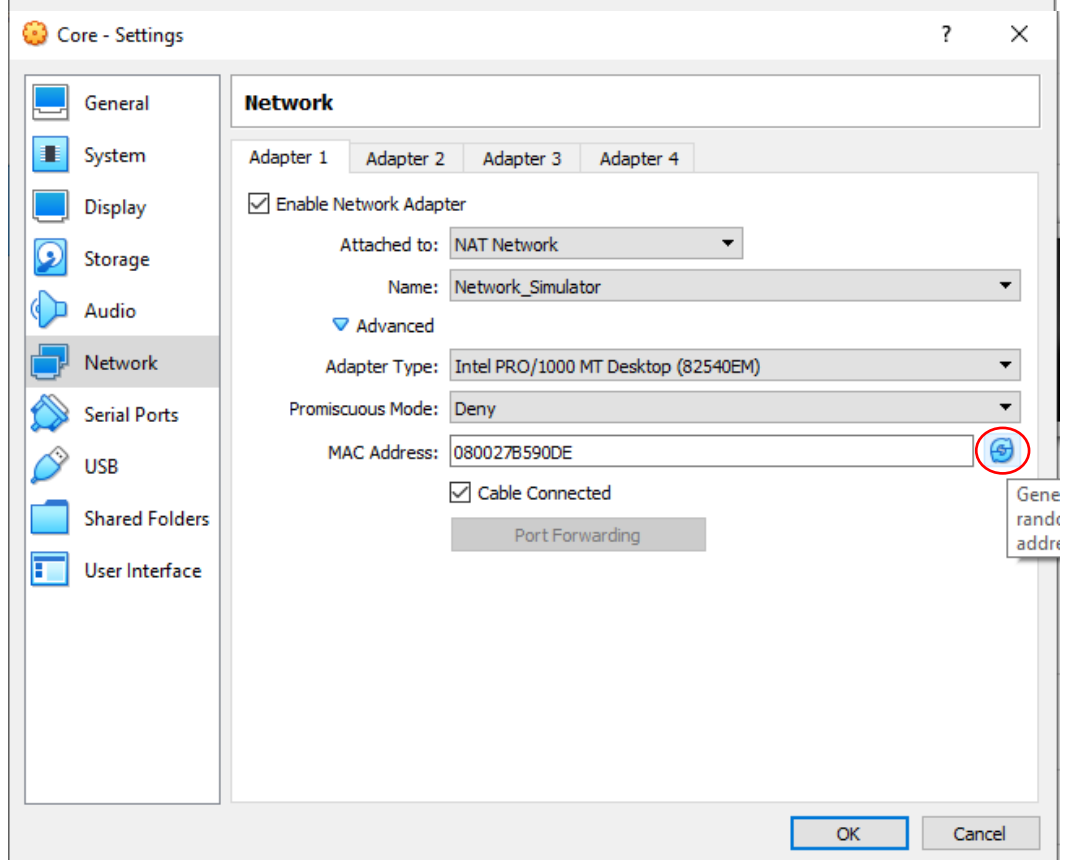
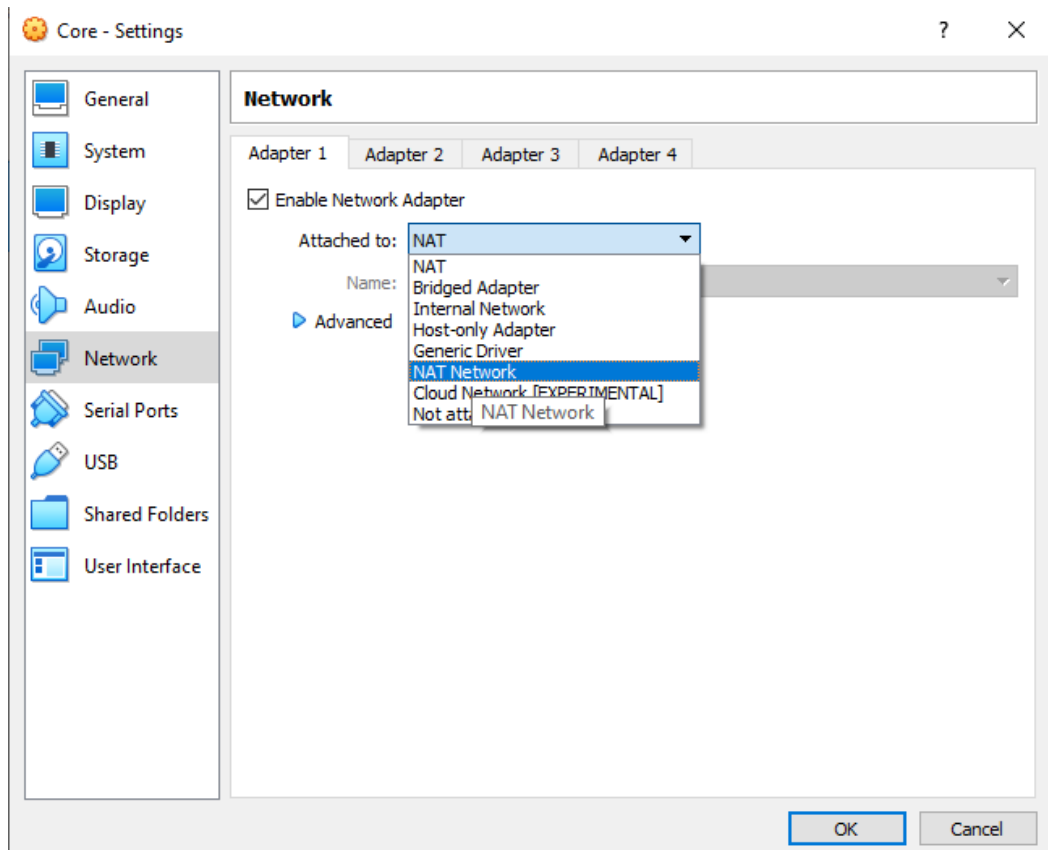




- c) Go to setting by click third symbol from top, Enable IPv6, keep all other options, and assign a reasonable name and network CIDR.



- d) Then move to each VM: go to Settings > Network and select NAT network that we have just created. For each VM, please make sure click refresh to avoid same MAC address and IP.



c. Open5GS Installation

I make this tutorial from these two sources:

[Katharina Kohls \(kkohls.org\)](https://kkohls.org)

[Building Open5GS from Sources | Open5GS](#)

You need to know in this simulation, for the open5gs I still run this program using test script in this path: /home/lab/open5gs/build/tests/app/ follow guide from [Katharina Kohls \(kkohls.org\)](https://kkohls.org).

1. Installation Open5Gs

- Turn On the VM, open Linux Ubuntu Terminal then change to superuser by using this command:
`sudo -i`
[insert your password]
- Update the VM using this command:
`sudo apt update`
- Install and enable mongodb:
`sudo apt install mongodb`
`sudo systemctl start mongodb`
`sudo systemctl enable mongodb` [this command to make mongodb running everytime you turn on the VM]
- Install requirements:
- `sudo apt install python3-pip python3-setuptools python3-wheel ninja-build build-essential flex bison git libsctp-dev libgnutls28-dev libgcrypt-dev libssl-dev libidn11-dev libmongoc-dev libbson-dev libyaml-dev libnghttp2-dev libmicrohttpd-dev libcurl4-gnutls-dev libnghttp2-dev libtins-dev meson libtalloc-dev cppcheck clang-tidy libsocket++1 python3-dev libbsd-arc4random-perl libkqueue-dev libssl-dev socket`
`sudo update`
`sudo upgrade`
- Build Open5Gs from sources
`cd /home/[pc name]`
`git clone https://github.com/open5gs/open5gs`
`cd open5gs`
`meson build --prefix=`pwd`/install`

```
Core (check 2) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Min 11:39
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM
root@core5gs-VirtualBox: /home/core5gs/open5gs

File Edit View Search Terminal Help
Has header "sys/socket.h": YES
Configuring upf-config.h using configuration
Native dependency libtins found: YES 3.4
Has header "net/if.h": YES
Has header "netinet/ip.h": YES
Has header "netinet/ip6.h": YES
Has header "netinet/ip_icmp.h": YES
Has header "netinet/icmp6.h": YES
Configuring smf-config.h using configuration
Configuring open5gs-dbctl using configuration
Compiler for C supports arguments -Wno-missing-prototypes -Wno-missing-declarat
ions -Wno-discarded-qualifiers -Wno-incompatible-pointer-types -Wno-unused-vari
able -Wno-deprecated-declarations: YES
Configuring test-config-private.h using configuration
Message:
prefix: /home/core5gs/open5gs/install
libdir: /home/core5gs/open5gs/install/lib/x86_64-
linux-gnu
bindir: /home/core5gs/open5gs/install/bin
sysconfdir: /home/core5gs/open5gs/install/etc
localstatedir: /home/core5gs/open5gs/install/var
source code location: /home/core5gs/open5gs
compiler: gcc
debugging support: debug

Build targets in project: 96
Found ninja-1.8.2 at /usr/bin/ninja
root@core5gs-VirtualBox: /home/core5gs/open5gs#
```

ninja -C build

```
Core (check 2) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Min 11:56
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM
root@core5gs-VirtualBox: /home/core5gs/open5gs

File Edit View Search Terminal Help
Has header "netinet/ip6.h": YES
Has header "netinet/ip_icmp.h": YES
Has header "netinet/icmp6.h": YES
Configuring smf-config.h using configuration
Configuring open5gs-dbctl using configuration
Compiler for C supports arguments -Wno-missing-prototypes -Wno-missing-declarat
ions -Wno-discarded-qualifiers -Wno-incompatible-pointer-types -Wno-unused-vari
able -Wno-deprecated-declarations: YES
Configuring test-config-private.h using configuration
Message:
prefix: /home/core5gs/open5gs/install
libdir: /home/core5gs/open5gs/install/lib/x86_64-
linux-gnu
bindir: /home/core5gs/open5gs/install/bin
sysconfdir: /home/core5gs/open5gs/install/etc
localstatedir: /home/core5gs/open5gs/install/var
source code location: /home/core5gs/open5gs
compiler: gcc
debugging support: debug

Build targets in project: 96
Found ninja-1.8.2 at /usr/bin/ninja
root@core5gs-VirtualBox: /home/core5gs/open5gs# sudo ./build/tests/attach/attach
sudo: ./build/tests/attach/attach: command not found
root@core5gs-VirtualBox: /home/core5gs/open5gs# sudo ninja -C build
ninja: Entering directory `build'
[3233/3233] Linking target tests/non3gpp/non3gpp.
root@core5gs-VirtualBox: /home/core5gs/open5gs#
```

- Run this command for the internal ip configuration
[make sure you already in path open5gs]

```
sudo ./misc/netconf.sh
```

- Run this test command:

```
./build/tests/attach/attach
```

```
root@core5gs-VirtualBox:/home/core5gs/open5gs# sudo ./build/tests/attach/attach
s1setup-test      : \
SUCCESS
guti-test         : SUCCESS
auth-test         : SUCCESS
idle-test         : SUCCESS
emm-status-test   : SUCCESS
ue-context-test   : SUCCESS
reset-test        : SUCCESS
issues-test       : SUCCESS
All tests passed.
```

```
./build/tests/registration/registration
```

```
root@core5gs-VirtualBox:/home/core5gs/open5gs# ./build/tests/registration/registration
guti-test         : SUCCESS
auth-test         : SUCCESS
idle-test         : SUCCESS
dereg-test        : /08/28 12:10:02.930: [amf] ERROR: [5:0] Error Indication
(..src/amf/nsmf-handler.c:786)
|Line 755: expected <41>, but saw <9>
|08/28 12:10:02.930: [amf] ERROR: Invalid Action[0] (../src/amf/ngap-handler.c:
1469)
\08/28 12:10:02.931: [amf] ERROR: Cannot find AMF-UE Context [27] (../src/amf/n
gap-handler.c:840)
08/28 12:10:02.931: [amf] ERROR: Cannot find AMF-UE Context [27] (../src/amf/ng
ap-handler.c:597)
08/28 12:10:02.931: [gmm] ERROR: Unknown message [103] (../src/amf/gmm-sm.c:123
6)
|08/28 12:10:02.932: [amf] ERROR: Cannot find PDU Session ID [5] (../src/amf/ng
ap-handler.c:1622)
^C
```

[This is okay, from acetcom the one who make this repo said that this is a bug, in my case I just ignore and continue to the next step. All functions still work fine]

```
ninja -C build test
```

```
root@core5gs-VirtualBox:/home/core5gs/open5gs# ninja -C build test
ninja: Entering directory `build'
[0/1] Running all tests.
1/13 open5gs:unit / core          OK          6.95 s
2/13 open5gs:unit / crypt         OK          0.09 s
3/13 open5gs:app / sctp           OK          0.01 s
4/13 open5gs:unit / unit          OK          0.04 s
5/13 open5gs:5gc / registration  TIMEOUT 30.04 s
6/13 open5gs:5gc / vonr          OK          11.52 s
7/13 open5gs:5gc / slice         OK          2.29 s
8/13 open5gs:epc / attach        OK          10.89 s
9/13 open5gs:epc / volte         OK          13.16 s
10/13 open5gs:epc / csfb         OK          7.19 s
11/13 open5gs:app / 310014       OK          6.44 s
12/13 open5gs:app / handover     OK          14.13 s
13/13 open5gs:epc / non3gpp      OK          7.17 s

OK:      12
FAIL:    1
SKIP:    0
TIMEOUT: 1
```

```
cd build
```

```
meson test -v
```

```

9/13 open5gs:epc / volte OK 13.18 s
mo-idle-test : SUCCESS
mt-idle-test : SUCCESS
mo-active-test : SUCCESS
mt-active-test : SUCCESS
mo-sms-test : SUCCESS
mt-sms-test : SUCCESS
crash-test : SUCCESS
All tests passed.
10/13 open5gs:epc / csfb OK 7.21 s
epc-test : SUCCESS
All tests passed.
11/13 open5gs:app / 310014 OK 6.15 s
epc-x2-test : SUCCESS
epc-s1-test : SUCCESS
5gc-xn-test : SUCCESS
5gc-n2-test : SUCCESS
All tests passed.
12/13 open5gs:app / handover OK 12.58 s
epdg-test : SUCCESS
All tests passed.
13/13 open5gs:epc / non3gpp OK 7.10 s

OK: 12
FAIL: 1
SKIP: 0
TIMEOUT: 1

```

ninja install

```

root@core5gs-VirtualBox: /home/core5gs/open5gs/build
File Edit View Search Terminal Help
th.join("/home/core5gs/open5gs/install/etc/freeDiameter", os.path.split("config
s/freeDiameter/hss.cert.pem")[1])) else False;'
Running custom install script '/usr/bin/python3 -c import os; import shutil; sh
util.copy("configs/freeDiameter/hss.key.pem", "/home/core5gs/open5gs/install/et
c/freeDiameter") if not os.environ.get("DESTDIR") and not os.path.isfile(os.pat
h.join("/home/core5gs/open5gs/install/etc/freeDiameter", os.path.split("config
s/freeDiameter/hss.key.pem")[1])) else False;'
Running custom install script '/usr/bin/python3 -c import os; import shutil; sh
util.copy("configs/freeDiameter/smf.cert.pem", "/home/core5gs/open5gs/install/e
tc/freeDiameter") if not os.environ.get("DESTDIR") and not os.path.isfile(os.pa
th.join("/home/core5gs/open5gs/install/etc/freeDiameter", os.path.split("config
s/freeDiameter/smf.cert.pem")[1])) else False;'
Running custom install script '/usr/bin/python3 -c import os; import shutil; sh
util.copy("configs/freeDiameter/smf.key.pem", "/home/core5gs/open5gs/install/et
c/freeDiameter") if not os.environ.get("DESTDIR") and not os.path.isfile(os.pat
h.join("/home/core5gs/open5gs/install/etc/freeDiameter", os.path.split("config
s/freeDiameter/smf.key.pem")[1])) else False;'
Running custom install script '/usr/bin/python3 -c import os; import shutil; sh
util.copy("configs/freeDiameter/pcrf.cert.pem", "/home/core5gs/open5gs/install/
etc/freeDiameter") if not os.environ.get("DESTDIR") and not os.path.isfile(os.p
ath.join("/home/core5gs/open5gs/install/etc/freeDiameter", os.path.split("confi
gs/freeDiameter/pcrf.cert.pem")[1])) else False;'
Running custom install script '/usr/bin/python3 -c import os; import shutil; sh
util.copy("configs/freeDiameter/pcrf.key.pem", "/home/core5gs/open5gs/install/e
tc/freeDiameter") if not os.environ.get("DESTDIR") and not os.path.isfile(os.pa
th.join("/home/core5gs/open5gs/install/etc/freeDiameter", os.path.split("config
s/freeDiameter/pcrf.key.pem")[1])) else False;'

```

Next, we will install the WebUI for user registration in HSS. Using this command:

```

cd ../..
sudo apt install curl
curl -fsSL https://deb.nodesource.com/setup_14.x |
sudo -E bash -
sudo apt install nodejs
cd open5gs/webui/
npm ci --no-optional

```



```

root@core5gs-VirtualBox:~# cd /home/core5gs/open5gs/webui/
root@core5gs-VirtualBox:/home/core5gs/open5gs/webui# npm ci --no-optional

> uglifyjs-webpack-plugin@0.4.6 postinstall /home/core5gs/open5gs/webui/node_modules/uglifyjs-webpack-plugin
> node lib/post_install.js

```

Congrats! You have finished the Open5Gs installation

d. UERANSIM Installation

- Make new VM for UERANSIM, do it like in VM preparation step and for VM network configuration you can start from step **d**.

- Build UERANSIM from sources

```

sudo apt install git make gcc g++ libsctp-dev lksctp-tools iproute2
sudo apt install snap
sudo snap install cmake --classic

```

```

root@core5gs-VirtualBox:/home/core5gs# sudo snap install cmake --classic
cmake 3.24.1 from Crascit ✓ installed

```

- Get the sources and build:

```

git clone https://github.com/aligungr/UERANSIM
cd UERANSIM
make

```

```

make[3]: Entering directory '/home/core5gs/UERANSIM/cmake-build-release'
[ 99%] Building CXX object CMakeFiles/nr-ue.dir/src/ue.cpp.o
[ 99%] Linking CXX executable nr-ue
make[3]: Leaving directory '/home/core5gs/UERANSIM/cmake-build-release'
[ 99%] Built target nr-ue
make[3]: Entering directory '/home/core5gs/UERANSIM/cmake-build-release'
make[3]: Leaving directory '/home/core5gs/UERANSIM/cmake-build-release'
make[3]: Entering directory '/home/core5gs/UERANSIM/cmake-build-release'
[100%] Building CXX object CMakeFiles/devbnd.dir/src/binder.cpp.o
[100%] Linking CXX shared library libdevbnd.so
make[3]: Leaving directory '/home/core5gs/UERANSIM/cmake-build-release'
[100%] Built target devbnd
make[3]: Entering directory '/home/core5gs/UERANSIM/cmake-build-release'
make[3]: Leaving directory '/home/core5gs/UERANSIM/cmake-build-release'
make[3]: Entering directory '/home/core5gs/UERANSIM/cmake-build-release'
[100%] Building CXX object CMakeFiles/nr-cli.dir/src/cli.cpp.o
[100%] Linking CXX executable nr-cli
make[3]: Leaving directory '/home/core5gs/UERANSIM/cmake-build-release'
[100%] Built target nr-cli
make[2]: Leaving directory '/home/core5gs/UERANSIM/cmake-build-release'
make[1]: Leaving directory '/home/core5gs/UERANSIM/cmake-build-release'
cp cmake-build-release/nr-gnb build/
cp cmake-build-release/nr-ue build/
cp cmake-build-release/nr-cli build/
cp cmake-build-release/libdevbnd.so build/
cp tools/nr-binder build/
UERANSIM successfully built.
root@core5gs-VirtualBox:/home/core5gs/UERANSIM#

```

- Congrats! You have successfully installed UERANSIM.

e. Open5GS Configuration

- Install package to check VM's IP with this command:

```
sudo apt install net-tools
ifconfig
```

```
root@core5gs-VirtualBox:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.14.2.10 netmask 255.255.255.0 broadcast 10.14.2.255
    inet6 fe80::1030:8aa5:49e8:723 prefixlen 64 scopeid 0x20<link>
    inet6 fd17:625c:f037:2:f38e:691c:ac11:655a prefixlen 64 scopeid 0x0<glo
lobal>
    inet6 fd17:625c:f037:2:dc:9b17:a457:9b86 prefixlen 64 scopeid 0x0<glo
bal>
    ether 08:00:27:bf:58:70 txqueuelen 1000 (Ethernet)
    RX packets 19 bytes 3283 (3.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 110 bytes 13346 (13.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 156 bytes 12257 (12.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 156 bytes 12257 (12.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

IP in interface enp0s3 (10.14.2.10) will be used for next configuration

- Make a backup of the Open5Gs original configuration:
cp /home/core5gs[VM's name]/open5gs/build/configs/sample.yaml /home/core5gs[VM's name]/open5gs/build/configs/sample.yaml.backup
- Then, configure the IP of AMF NGAP by using this command:
cd /home/core5gs[VM's name]/open5gs/build/configs
nano sample.yaml [To open, view, and edit the file]

```
GNU nano 2.9.3 sample.yaml

amf:
  sbi:
    - addr: 127.0.0.5
      port: 7777
  ngap:
    - addr: 127.0.0.5
```

Change this parameter to:

```
amf:
  sbi:
    - addr: 127.0.0.5
      port: 7777
  ngap:
    - addr: 10.14.2.10 [IP from interface enp0s3]
```

Then, I tried to change some others parameters in core with this parameter:

```

guami:
  - plmn_id:
      mcc: 510
      mnc: 14
      amf_id:
        region: 2
        set: 1
  tai:
    - plmn_id:
        mcc: 510
        mnc: 14
        tac: 5400
  plmn_support:
    - plmn_id:
        mcc: 510
        mnc: 14
      s_nssai:
        - sst: 1

```

You can try to modify these parameter as you want: **PLMN ID** consist of **MCC** (country region for Indonesia) and **MNC** (Service provider's unique code), **TAC** (Type Allocation Code), and **S-NSSAI** (Single Network Slice Selection Assistance Information) this parameter is new param in 5G core consist of Consist of Slice/Service type (**SST**) and Slice Differentiator (**SD**). These three parameter are important parameter for integration between Core, HSS database, gNb, and UE (Simcard).

f. UERANSIM Configuration

- Go to config directory and create a copy of the configs:
`cd /home/ueran/UERANSIM/config`
`cp open5gs-gnb.yaml virtualbox_open5gs-gnb.yaml`
`cp open5gs-ue.yaml virtualbox_open5gs-ue.yaml`
- Make the following changes to the files:
`nano virtualbox_open5gs-gnb.yaml`
- First, put in the IP of this VM:

```

GNU nano 4.8                                virtualbox_open5gs-gnb.yaml                                Modified
mcc: '510'                                # Mobile Country Code value
mnc: '14'                                # Mobile Network Code value (2 or 3 digits)

nci: '0x000000010' # NR Cell Identity (36-bit)
idLength: 32      # NR gNB ID length in bits [22...32]
tac: 5400          # Tracking Area Code

linkIp: 10.14.2.6 # gNB's local IP address for Radio Link Simulation (Usually
ngapIp: 10.14.2.6 # gNB's local IP address for N2 Interface (Usually same wi
gtpIp: 10.14.2.6  # gNB's local IP address for N3 Interface (Usually same wi

# List of AMF address information
amfConfigs:
  - address: 10.14.2.10
    port: 38412

# List of supported S-NSSAIs by this gNB
slices:
  - sst: 1
    # sd: 011498

# Indicates whether or not SCTP stream number errors should be ignored.
ignoreStreamIds: true
linkIp: 10.14.2.6
ngapIp: 10.14.2.6
gtpIp: 10.14.2.6
Use IP from interface enp0s3, you can see it from ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.14.2.6 netmask 255.255.255.0 broadcast 10.14.2.255
        inet6 fd17:625c:f037:e02:103c:ccea:737e:9cdb prefixlen 64 scopeid 0x0
<global>
        inet6 fd17:625c:f037:e02:b2e7:25a:9224:489e prefixlen 64 scopeid 0x0<

```

- In the same file as before, change the IP of the AMF in file same as the Open5Gs NGAP IP:
amfConfigs:
- address: 10.14.2.10
port: 38412
- Start to build the gNb with our adjusted config file:
./build/nr-gnb -c config/virtualbox_open5gs-gnb.yaml
- Continue with the UE configuration:
nano config/virtualbox_open5gs-ue.yaml
- In this file, you need to change the gNb IP configuration in this UE's part. So, UE can connect to gNb:

```

# List of gNB IP addresses for Radio Link Simulation
gnbSearchList:
  - 10.14.2.6

gnbSearchList:
  - 10.14.2.6

```

This IP represent the gNb's IP, which is ip from interface enp0s3

- Finally, we need to register the UE in the core network. You can access and edit the required information from the UE configuration file, by using this command:
nano config/virtualbox_open5gs-ue.yaml

```

GNU nano 4.8          virtualbox open5gs-ue.yaml
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)
supi: 'imsi-510140000000001'
# Mobile Country Code value of HPLMN
mcc: '510'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '14'

# Permanent subscription key
key: '465B5CE8B199B49FAA5F0A2EE238A6BC'
# Operator code (OP or OPC) of the UE
op: 'E8ED289DEBA952E4283B54E88E6183CA'
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OPC'
# Authentication Management Field (AMF) value
amf: '8000'
# IMEI number of the device. It is used if no SUPI is provided
imei: '356938035643803'
# IMEISV number of the device. It is used if no SUPI and IMEI is provided
imeiSv: '4370816125816151'

# List of gNB IP addresses for Radio Link Simulation
gnbSearchList:
  - 10.14.2.6

# UAC Access Identities Configuration

```

which should by default contain the following pieces of information:

imsi: 510140000000001

key: 465B5CE8B199B49FAA5F0A2EE238A6BC

AMF: 8000

USIM: OPC

Operator Key: E8ED289DEBA952E4283B54E88E6183CA

You should configure these parameters same as parameters in HSS, so the attach process can be successfully accepted. Run the HSS via webUI in open5gs VM using this command:

```
cd /home/core5gs/open5gs/webui
```

```
npm run dev
```

```

> open5gs@2.4.7 dev /home/lab5g/open5gs/webui
> node server/index.js

> Using external babel configuration
> Location: "/home/lab5g/open5gs/webui/.babelrc"

DONE Compiled successfully in 2730ms 10:24:18 AM

Mongoose: accounts.count({}, {})
Mongoose: subscribers.ensureIndex({ imsi: 1 }, { unique: true, background: true })
Mongoose: accounts.ensureIndex({ username: 1 }, { unique: true, background: true })
> Ready on http://localhost:3000
Mongoose: accounts.findOne({ '$or': [ { username: 'admin' } ] }, { fields: { hash: 0, salt: 0 } })
> Building page: /

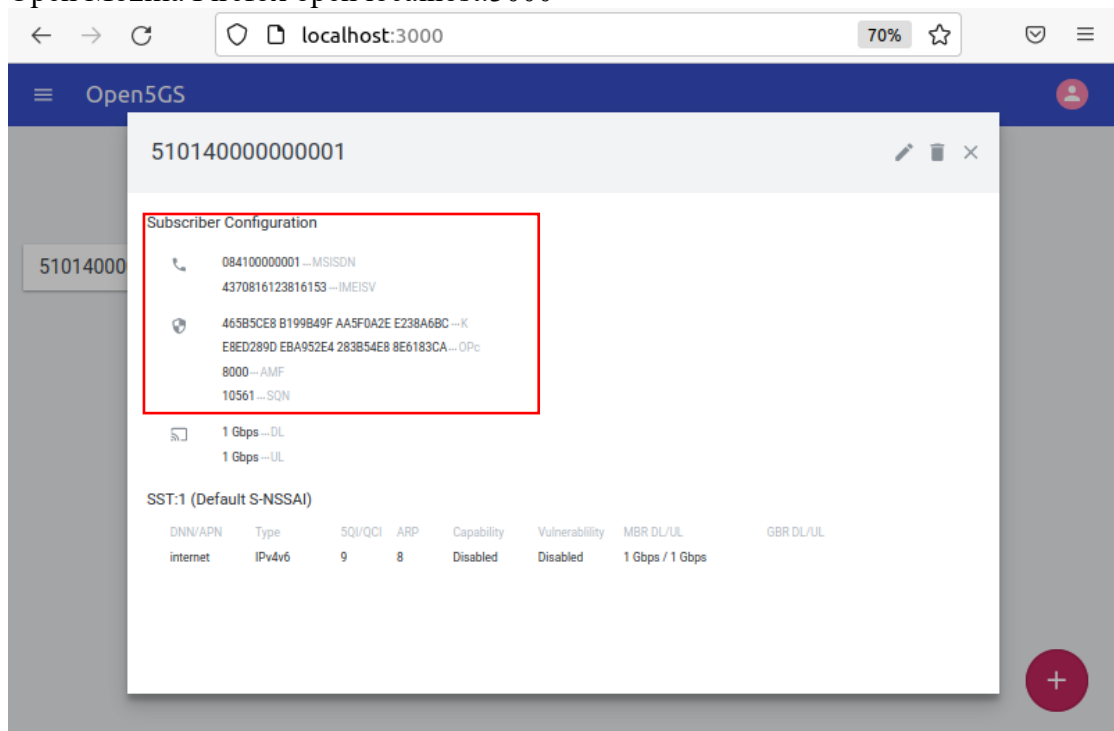
DONE Compiled successfully in 11025ms 10:25:47 AM

WAIT Compiling... 10:25:48 AM

Mongoose: accounts.findOne({ '$or': [ { username: 'admin' } ] }, { fields: { hash: 0, salt: 0 } })

```

Open Mozilla Firefox open localhost:3000



g. 5G SA Simulation

1. Turn on and run open5gs
 - a) Turn on the core5g VM (VM where you installed open5gs).
 - b) Open the terminal, then change to root user by using this command:

sudo -i

if already change to root user, before @ will be seen as root

root@lab5g-VirtualBox:

- c) Everytime turn on the core5g VM you need to execute this script for the core internal routing. By using this command:
`cd /home/core5g[VM's name]/open5gs`
`sudo ./misc/netconf.sh` [after executing this command there will be no output, but it's okay]
- d) Then execute this command to run all open5gs program:
`cd /home/core5g/open5gs/build/tests/app/ && ./app`
 There are three programs in this path: 5gc, epc, and app. 5gc is only to turn on 5G core function, epc is only to turn on 4G core function, and app will turn on both 4G and 5G core function.
- e) After all programs can be successfully run like this picture below, it means your core already working fine.

```
08/30 16:20:20.341: [udm] INFO: [f8146582-2844-41ed-b1bd-3380d80badfa] (NRF-notify) NF
registered (./src/udm/nnrf-handler.c:181)
08/30 16:20:20.341: [udm] INFO: [f8146582-2844-41ed-b1bd-3380d80badfa] (NRF-notify) NF
Profile updated (./src/udm/nnrf-handler.c:202)
08/30 16:20:20.341: [pcf] INFO: [f8146582-2844-41ed-b1bd-3380d80badfa] (NRF-notify) NF
registered (./src/pcf/nnrf-handler.c:181)
08/30 16:20:20.341: [pcf] INFO: [f8146582-2844-41ed-b1bd-3380d80badfa] (NRF-notify) NF
Profile updated (./src/pcf/nnrf-handler.c:202)
08/30 16:20:20.342: [udr] INFO: [f8146582-2844-41ed-b1bd-3380d80badfa] NF registered [H
eartbeat:10s] (./src/udr/nf-sm.c:216)
08/30 16:20:22.294: [sgwc] WARNING: PFCP[REQ] has already been associated (./src/sgwc/
pfcg-sm.c:209)
08/30 16:20:22.294: [sgwu] WARNING: PFCP[RSP] has already been associated (./src/sgwu/
pfcg-sm.c:202)
08/30 16:20:22.346: [smf] WARNING: PFCP[REQ] has already been associated (./src/smf/pf
cg-sm.c:213)
08/30 16:20:22.346: [upf] WARNING: PFCP[RSP] has already been associated (./src/upf/pf
cg-sm.c:207)
08/30 16:20:30.721: [scp] WARNING: [f7b55718-2844-41ed-992c-3329b4d80ccf] Retry to regi
stration with NRF (./src/scp/nf-sm.c:176)
08/30 16:20:30.721: [nrf] INFO: [f7b55718-2844-41ed-992c-3329b4d80ccf] NF registered [H
eartbeat:10s] (./src/nrf/nf-sm.c:192)
08/30 16:20:30.721: [scp] INFO: [f7b55718-2844-41ed-992c-3329b4d80ccf] NF registered [H
eartbeat:10s] (./src/scp/nf-sm.c:217)
```

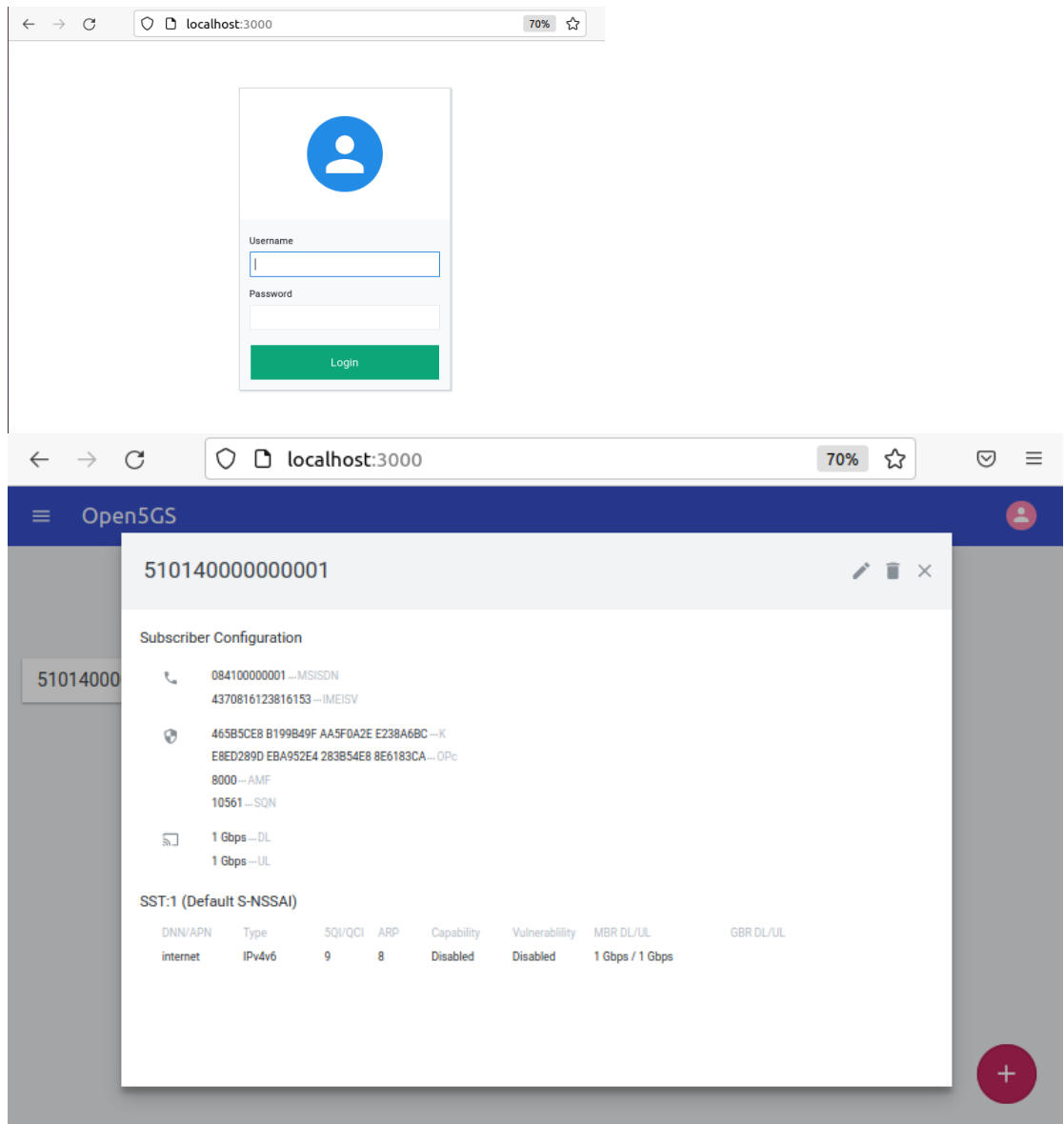
[Additional] To add and see SIM card information on HSS database

- a) Open new tab on terminal, then change the user to root.
- b) Run this command below:
`cd /home/core5g/open5gs/webui/ && npm run dev`
- c) If already appeared this message, it means the database already running successfully.

```
DONE Compiled successfully in 2729ms 4:41:47 PM

Mongoose: accounts.count({}, {})
Mongoose: subscribers.ensureIndex({ imsi: 1 }, { unique: true, background: true })
Mongoose: accounts.ensureIndex({ username: 1 }, { unique: true, background: true })
> Ready on http://localhost:3000
Mongoose: accounts.findOne({ '$or': [ { username: 'admin' } ] }, { fields: { hash: 0, s
alt: 0 } })
Client pings, but there's no entry for page: /
Mongoose: accounts.findOne({ '$or': [ { username: 'admin' } ] }, { fields: { hash: 0, s
alt: 0 } })
> Building page: /
```

- d) Then you can open Mozilla Firefox and go to link <http://localhost:3000> , then login using this:
 Username: admin
 Password: 1423
- e) After you successfully login, you can see the SIMcard database or add new SIMcard database.



2. Turn on and run UERANSIM (RAN & UE)

RAN

- Turn on ueran VM, then open terminal.
- Change the user to root by using this command:
sudo -i
- Change the path, then execute this command to run the RAN program:
**cd /home/ueran/UERANSIM/
./build/nr-gnb -c config/virtualbox_open5gs-gnb.yaml**
- Check in the ueran's terminal and core5g's terminal. If this message below appears, it means gNb to core already connected. This below message from

ueran's terminal

```
UERANSIM v3.2.6
[2022-08-31 10:37:50.153] [sctp] [info] Trying to establish SCTP connection...
(10.14.2.5:38412)
[2022-08-31 10:37:50.214] [sctp] [info] SCTP connection established (10.14.2.5:
38412)
[2022-08-31 10:37:50.214] [sctp] [debug] SCTP association setup ascId[3]
[2022-08-31 10:37:50.214] [ngap] [debug] Sending NG Setup Request
[2022-08-31 10:37:50.301] [ngap] [debug] NG Setup Response received
[2022-08-31 10:37:50.301] [ngap] [info] NG Setup procedure is successful
```

and this message below from core5g's terminal

```
08/30 16:20:20.342: [udr] INFO: [f8146582-2844-41ed-b1bd-3380d80badfa] NF registered [H
eartbeat:10s] (../src/udr/nf-sm.c:216)
08/30 16:20:22.294: [sgwc] WARNING: PFCP[REQ] has already been associated (../src/sgwc/
pfcf-sm.c:209)
08/30 16:20:22.294: [sgwu] WARNING: PFCP[RSP] has already been associated (../src/sgwu/
pfcf-sm.c:202)
08/30 16:20:22.346: [smf] WARNING: PFCP[REQ] has already been associated (../src/smf/pf
cp-sm.c:213)
08/30 16:20:22.346: [upf] WARNING: PFCP[RSP] has already been associated (../src/upf/pf
cp-sm.c:207)
08/30 16:20:30.721: [scp] WARNING: [f7b55718-2844-41ed-992c-3329b4d80ccf] Retry to regi
stration with NRF (../src/scp/nf-sm.c:176)
08/30 16:20:30.721: [nrf] INFO: [f7b55718-2844-41ed-992c-3329b4d80ccf] NF registered [H
eartbeat:10s] (../src/nrf/nf-sm.c:192)
08/30 16:20:30.721: [scp] INFO: [f7b55718-2844-41ed-992c-3329b4d80ccf] NF registered [H
eartbeat:10s] (../src/scp/nf-sm.c:217)
08/31 10:37:50.252: [amf] INFO: gNB-N2 accepted[10.14.2.6]:40284 in ng-path module (../
src/amf/ngap-sctp.c:113)
08/31 10:37:50.252: [amf] INFO: gNB-N2 accepted[10.14.2.6] in master_sm module (../src/
amf/amf-sm.c:660)
08/31 10:37:50.252: [amf] INFO: [Added] Number of gNBs is now 1 (../src/amf/context.c:8
77)
08/31 10:37:50.252: [amf] INFO: gNB-N2[10.14.2.6] max_num_of_ostreams : 10 (../src/amf/
amf-sm.c:699)
```

UE

- e) Open new terminal tab, and change the user to root.
- f) Change the path, the execute this command:
`cd /home/ueran/UERANSIM/build`
`./nr-ue -c ../config/virtualbox_open5gs-ue.yaml`
This command simulates one UE trying to attach to the BTS.
- g) Check the message in the current ueran's terminal. "Connection setup for PDU session[1] is successful, TUN interface ...", it means the UE has already successfully attached to the gNb and can get service from the core side.

```

[2022-08-31 10:47:16.346] [rrc] [debug] Sending RRC Setup Request
[2022-08-31 10:47:16.347] [rrc] [info] RRC connection established
[2022-08-31 10:47:16.347] [rrc] [info] UE switches to state [RRC-CONNECTED]
[2022-08-31 10:47:16.347] [nas] [info] UE switches to state [MM-REGISTER-INITIA
TED]
[2022-08-31 10:47:16.347] [nas] [info] UE switches to state [CM-CONNECTED]
[2022-08-31 10:47:16.398] [nas] [debug] Authentication Request received
[2022-08-31 10:47:16.410] [nas] [debug] Security Mode Command received
[2022-08-31 10:47:16.410] [nas] [debug] Selected integrity[2] ciphering[0]
[2022-08-31 10:47:16.434] [nas] [debug] Registration accept received
[2022-08-31 10:47:16.434] [nas] [info] UE switches to state [MM-REGISTERED/NORM
AL-SERVICE]
[2022-08-31 10:47:16.434] [nas] [debug] Sending Registration Complete
[2022-08-31 10:47:16.434] [nas] [info] Initial Registration is successful
[2022-08-31 10:47:16.434] [nas] [debug] Sending PDU Session Establishment Reque
st
[2022-08-31 10:47:16.434] [nas] [debug] UAC access attempt is allowed for ident
ity[0], category[M0_sig]
[2022-08-31 10:47:16.638] [nas] [debug] Configuration Update Command received
[2022-08-31 10:47:16.719] [nas] [debug] PDU Session Establishment Accept receiv
ed
[2022-08-31 10:47:16.720] [nas] [info] PDU Session establishment is successful
PSI[1]
[2022-08-31 10:47:16.770] [app] [info] Connection setup for PDU session[1] is s
uccessful, TUN interface[uesimtun0, 10.45.0.2] is up.

```

Message from ueran's terminal for UE

```

2022-08-31 10:37:50.214] [ngap] [debug] Sending NG Setup Request
2022-08-31 10:37:50.301] [ngap] [debug] NG Setup Response received
2022-08-31 10:37:50.301] [ngap] [info] NG Setup procedure is successful
2022-08-31 10:46:51.255] [rrc] [debug] UE[1] new signal detected
2022-08-31 10:47:04.114] [rrc] [debug] UE[2] new signal detected
2022-08-31 10:47:04.314] [rls] [debug] UE[1] signal lost
2022-08-31 10:47:09.329] [rls] [debug] UE[2] signal lost
Croot@lab5g-VirtualBox:/home/lab5g/UERANSIM# ./build/nr-gnb -c config/virtualb
ox_open5gs-gnb.yaml
UERANSIM v3.2.6
2022-08-31 10:47:12.406] [sctp] [info] Trying to establish SCTP connection...
(10.14.2.5:38412)
2022-08-31 10:47:12.417] [sctp] [info] SCTP connection established (10.14.2.5:
38412)
2022-08-31 10:47:12.418] [sctp] [debug] SCTP association setup ascId[4]
2022-08-31 10:47:12.418] [ngap] [debug] Sending NG Setup Request
2022-08-31 10:47:12.419] [ngap] [debug] NG Setup Response received
2022-08-31 10:47:12.419] [ngap] [info] NG Setup procedure is successful
2022-08-31 10:47:16.342] [rrc] [debug] UE[1] new signal detected
2022-08-31 10:47:16.347] [rrc] [info] RRC Setup for UE[1]
2022-08-31 10:47:16.347] [ngap] [debug] Initial NAS message received from UE[1]
2022-08-31 10:47:16.434] [ngap] [debug] Initial Context Setup Request received
2022-08-31 10:47:16.732] [ngap] [info] PDU session resource(s) setup for UE[1]
count[1]

```

Message from ueran's terminal for gNb

```

c:892)
08/31 10:47:16.660: [smf] INFO: [Added] Number of SMF-Sessions is now 1 (../src/smf/context.c:2972)
08/31 10:47:16.681: [app] WARNING: Try to discover [UDM] (../lib/sbi/path.c:114)
08/31 10:47:16.683: [smf] INFO: [f7f7a65e-2844-41ed-b5ac-39d76d8dffffa] (NF-discover) NF registered (../src/smf/nnrf-handler.c:294)
08/31 10:47:16.683: [smf] INFO: [f7f7a65e-2844-41ed-b5ac-39d76d8dffffa] (NF-discover) NF Profile updated (../src/smf/nnrf-handler.c:336)
08/31 10:47:16.702: [smf] INFO: UE SUPI[imsi-510140000000001] DNN[internet] IPv4[10.45.0.2] IPv6[] (../src/smf/npcf-handler.c:497)
08/31 10:47:16.702: [upf] INFO: [Added] Number of UPF-Sessions is now 1 (../src/upf/context.c:178)
08/31 10:47:16.718: [gtp] INFO: gtp_connect() [127.0.0.4]:2152 (../lib/gtp/path.c:60)
08/31 10:47:16.718: [upf] INFO: UE F-SEID[CP:0x1 UP:0x1] APN[internet] PDN-Type[1] IPv4[10.45.0.2] IPv6[] (../src/upf/context.c:397)
08/31 10:47:16.718: [upf] INFO: UE F-SEID[CP:0x1 UP:0x1] APN[internet] PDN-Type[1] IPv4[10.45.0.2] IPv6[] (../src/upf/context.c:397)
08/31 10:47:16.718: [gtp] INFO: gtp_connect() [127.0.0.7]:2152 (../lib/gtp/path.c:60)
08/31 10:47:16.736: [app] WARNING: Try to discover [AMF] (../lib/sbi/path.c:114)
08/31 10:47:16.736: [smf] INFO: [f7e1549e-2844-41ed-9019-1728820a55e6] (NF-discover) NF registered (../src/smf/nnrf-handler.c:294)
08/31 10:47:16.736: [smf] INFO: [f7e1549e-2844-41ed-9019-1728820a55e6] (NF-discover) NF Profile updated (../src/smf/nnrf-handler.c:336)
08/31 10:47:16.758: [gtp] INFO: gtp_connect() [10.14.2.6]:2152 (../lib/gtp/path.c:60)

```

Message from core5g's terminal

- h) You can try to test the UE service by trying to do ping from UE. First, you need to open new terminal tab, then execute this command:

```

sudo -i
cd /home /ueran/UERANSIM/build
./nr-binder 10.45.0.2 [UE's IP assigned from core] ping
google.com

```

```

root@lab5g-VirtualB... x root@lab5g-VirtualB... x root@lab5g-VirtualB... x
lab5g@lab5g-VirtualBox:~$ sudo -i
[sudo] password for lab5g:
root@lab5g-VirtualBox:~# cd /home/lab5g/UERANSIM/build/
root@lab5g-VirtualBox:/home/lab5g/UERANSIM/build# ./nr-binder 10.45.0.2 ping google.com
PING google.com (74.125.68.139) 56(84) bytes of data.
64 bytes from sc-in-f139.1e100.net (74.125.68.139): icmp_seq=1 ttl=102 time=33.1 ms
64 bytes from sc-in-f139.1e100.net (74.125.68.139): icmp_seq=2 ttl=102 time=31.1 ms
64 bytes from sc-in-f139.1e100.net (74.125.68.139): icmp_seq=3 ttl=102 time=48.7 ms
64 bytes from sc-in-f139.1e100.net (74.125.68.139): icmp_seq=4 ttl=102 time=35.1 ms

```

Check the UE's IP from this message below

```

[2022-08-31 10:47:16.638] [nas] [debug] Configuration Update Command received
[2022-08-31 10:47:16.719] [nas] [debug] PDU Session Establishment Accept received
[2022-08-31 10:47:16.720] [nas] [info] PDU Session establishment is successful PSI[1]
[2022-08-31 10:47:16.770] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.45.0.2] is up.

```

[Signalling Trace]

If you want to know the signalling trace between core to gNb or gNb to UE, you just need to open new terminal tab, change the user to root, then type "wireshark" and press enter. Wireshark needs to be opened in both VMs, so you can see signalling trace from the both side.

Sigtrace Example:

i. gNb connected to Core

No.	Time	Source	Destination	Protocol	Length	Info
19	6.808709674	10.14.2.6	10.14.2.5	SCTP	82	INIT
20	6.809141087	10.14.2.5	10.14.2.6	SCTP	306	INIT_ACK
21	6.809163711	10.14.2.6	10.14.2.5	SCTP	278	COOKIE_ECHO
22	6.809525064	10.14.2.5	10.14.2.6	SCTP	60	COOKIE_ACK
23	6.809930078	10.14.2.6	10.14.2.5	NGAP	134	NGSetupRequest
24	6.810260328	10.14.2.5	10.14.2.6	SCTP	62	SACK
25	6.810387735	10.14.2.5	10.14.2.6	NGAP	118	NGSetupResponse
26	6.810405662	10.14.2.6	10.14.2.5	SCTP	62	SACK

Frame 23: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface enp0s3, id 0
 Ethernet II, Src: PcsCompu_59:11:bd (08:00:27:59:11:bd), Dst: PcsCompu_b6:5b:62 (08:00:27:b6:5b:62)
 Internet Protocol Version 4, Src: 10.14.2.6, Dst: 10.14.2.5
 Stream Control Transmission Protocol, Src Port: 39838 (39838), Dst Port: 38412 (38412)
 NG Application Protocol

Offset	Hex	ASCII
0000	08 00 27 b6 5b 62 08 00 27 59 11 bd 08 00 45 02	...[b...Y...E...
0010	00 78 00 00 40 00 40 84 21 da 0a 0e 02 06 0a 0e	x...@...!.....
0020	02 05 9b 9e 96 0c b3 dd fd 6c 7e 7e 9d 7c 00 031... ...
0030	00 55 74 6e e7 f1 00 00 00 00 00 00 3c 00 15	Utn.....<...
0040	00 41 00 00 04 00 1b 00 09 00 15 f0 41 50 00 00	A.....AP...
0050	00 01 00 52 40 17 0a 00 55 45 52 41 4e 53 49 4d	..R@...UERANSIM
0060	2d 67 6e 62 2d 35 31 30 2d 31 34 2d 31 00 66 00	-gnb-510 -14-1 f...
0070	0d 00 00 00 15 18 00 15 f0 41 00 00 00 08 00 15A.....

Frame (134 bytes) Bitstring tvb (4 bytes) Unaligned OCTET STRING (1 byte)

wireshark_enp...1kTsq.pcapng Packets: 38 · Displayed: 38 (100.0%) · Dropped: 0 (0.0%) Profile: Default

Sigtrace from gNb's side.

No.	Time	Source	Destination	Protocol	Length	Info
140	5.665702133	10.14.2.6	10.14.2.5	SCTP	82	INIT
141	5.665761150	10.14.2.5	10.14.2.6	SCTP	306	INIT_ACK
142	5.666108856	10.14.2.6	10.14.2.5	SCTP	278	COOKIE_ECHO
143	5.666148704	10.14.2.5	10.14.2.6	SCTP	50	COOKIE_ACK
144	5.666805571	10.14.2.6	10.14.2.5	NGAP	134	NGSetupRequest
145	5.666883301	10.14.2.5	10.14.2.6	SCTP	62	SACK
146	5.667042297	10.14.2.5	10.14.2.6	NGAP	118	NGSetupResponse
147	5.667252207	10.14.2.6	10.14.2.5	SCTP	62	SACK

Frame 144: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface enp0s3, id 0
 Ethernet II, Src: PcsCompu_59:11:bd (08:00:27:59:11:bd), Dst: PcsCompu_b6:5b:62 (08:00:27:b6:5b:62)
 Internet Protocol Version 4, Src: 10.14.2.6, Dst: 10.14.2.5
 Stream Control Transmission Protocol, Src Port: 39838 (39838), Dst Port: 38412 (38412)
 NG Application Protocol

Offset	Hex	ASCII
0000	08 00 27 b6 5b 62 08 00 27 59 11 bd 08 00 45 02	...[b...Y...E...
0010	00 78 00 00 40 00 40 84 21 da 0a 0e 02 06 0a 0e	x...@...!.....
0020	02 05 9b 9e 96 0c b3 dd fd 6c 7e 7e 9d 7c 00 031... ...
0030	00 55 74 6e e7 f1 00 00 00 00 00 00 3c 00 15	Utn.....<...
0040	00 41 00 00 04 00 1b 00 09 00 15 f0 41 50 00 00	A.....AP...
0050	00 01 00 52 40 17 0a 00 55 45 52 41 4e 53 49 4d	..R@...UERANSIM
0060	2d 67 6e 62 2d 35 31 30 2d 31 34 2d 31 00 66 00	-gnb-510 -14-1 f...
0070	0d 00 00 00 15 18 00 15 f0 41 00 00 00 08 00 15A.....

Frame (134 bytes) Bitstring tvb (4 bytes) Unaligned OCTET STRING (1 byte)

wireshark_enp0s3_20220831110604_stziel.pcapng Packets: 173 · Displayed: 173 (100.0%) Profile: Default

Sigtrace from Core's side.

ii. UE connected to Core

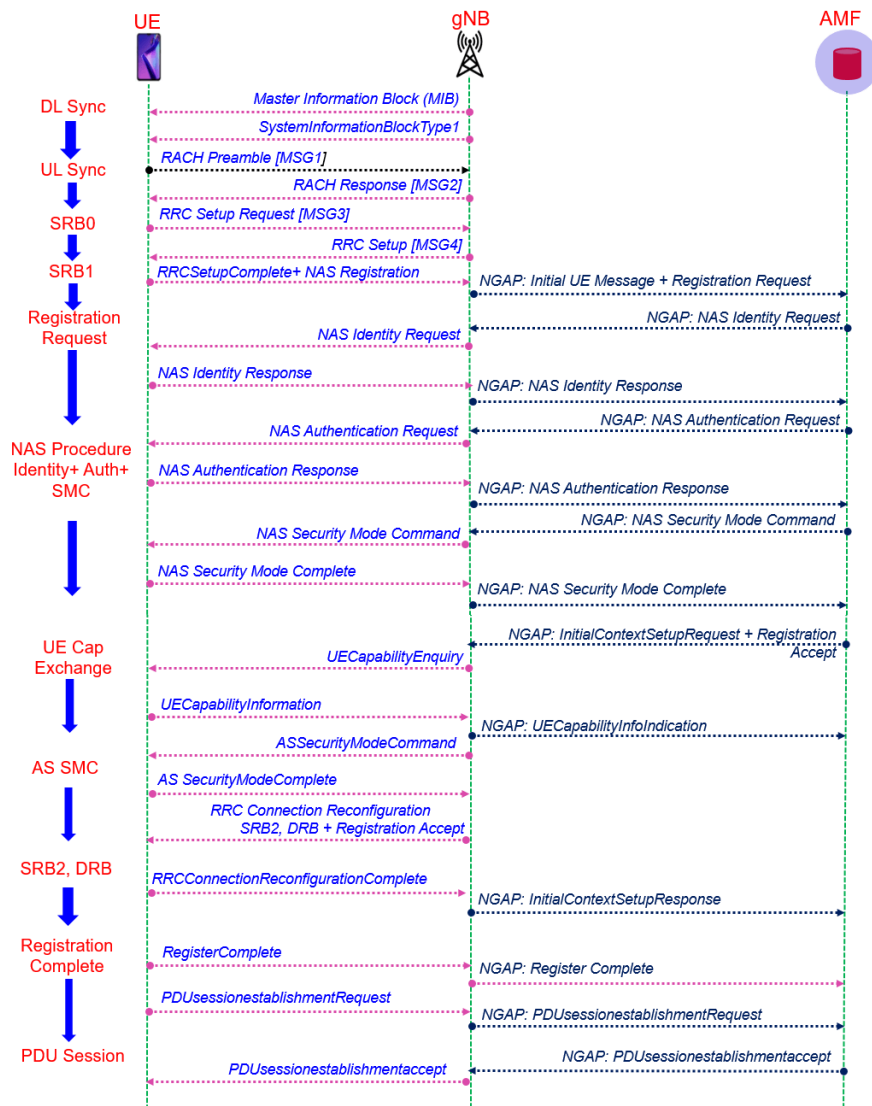
*enp0s3						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
23	12.386515976	10.14.2.6	10.14.2.5	NGAP/N...	138	InitialUEMessag
24	12.392882195	10.14.2.5	10.14.2.6	NGAP/N...	146	DownlinkNASTran
25	12.393289997	10.14.2.6	10.14.2.5	NGAP/N...	146	UplinkNASTransp
26	12.396143889	10.14.2.5	10.14.2.6	NGAP/N...	126	DownlinkNASTran
27	12.399970809	10.14.2.6	10.14.2.5	NGAP/N...	186	UplinkNASTransp
28	12.408525938	10.14.2.5	10.14.2.6	NGAP/N...	230	InitialContextS
29	12.408799802	10.14.2.6	10.14.2.5	NGAP	98	InitialContextS
30	12.412647469	10.14.2.5	10.14.2.6	SCPD	62	SACK
▶ Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface enp0s3, id 0 ▶ Ethernet II, Src: PcsCompu_b6:5b:62 (08:00:27:b6:5b:62), Dst: PcsCompu_59:11:bd (08:00:27:59:11:bd) ▶ Internet Protocol Version 4, Src: 10.14.2.5, Dst: 10.14.2.6 ▶ Stream Control Transmission Protocol, Src Port: 38412 (38412), Dst Port: 39838 (39838)						
0000 08 00 27 59 11 bd 08 00 27 b6 5b 62 08 00 45 02 ..'Y... '[b-E- 0010 00 5c a8 7c 40 00 40 84 79 79 0a 0e 02 05 0a 0e .\ @.@ yy..... 0020 02 06 96 0c 9b 9e be 38 49 7e e4 3a ab 4e 04 008 I~:~N.. 0030 00 3c 00 01 00 38 02 00 9b 9e 0a 0e 02 06 00 00 <...8..... 0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0050 00 00 00 00 00 00 00 00 00 00 fe 3f 3d 98 5a 09?=-Z- 0060 4d bb 00 00 00 00 00 00 00 00M.....						
wireshark_enp0s3_2022...1110819_sWCUwt.pcapng Packets: 78 · Displayed: 78 (100.0%) Profile: Default						

Sigtrace from gNb's side.

*enp0s3						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
13	11.106690525	10.14.2.6	10.14.2.5	NGAP/N...	138	InitialUEMessage, Registr
14	11.112776252	10.14.2.5	10.14.2.6	NGAP/N...	146	DownlinkNASTransport, Aut
15	11.113433263	10.14.2.6	10.14.2.5	NGAP/N...	146	UplinkNASTransport, Auth
16	11.116064438	10.14.2.5	10.14.2.6	NGAP/N...	126	DownlinkNASTransport
17	11.120307980	10.14.2.6	10.14.2.5	NGAP/N...	186	UplinkNASTransport
18	11.128432022	10.14.2.5	10.14.2.6	NGAP/N...	230	InitialContextSetupReques
19	11.128947186	10.14.2.6	10.14.2.5	NGAP	98	InitialContextSetupRespon
20	11.222665004	10.14.2.5	10.14.2.6	SCPD	62	SACK
▶ Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface enp0s3, id 0 ▶ Ethernet II, Src: PcsCompu_b6:5b:62 (08:00:27:b6:5b:62), Dst: RealtekU_12:35:00 (52:54:00:12:35:00) ▶ Internet Protocol Version 4, Src: 10.14.2.5, Dst: 143.244.33.72 ▶ Transmission Control Protocol, Src Port: 55200, Dst Port: 443, Seq: 1, Ack: 1, Len: 0						
0000 52 54 00 12 35 00 08 00 27 b6 5b 62 08 00 45 00 RT..5... '[b-E- 0010 00 28 41 8f 40 00 40 06 3b f2 0a 0e 02 05 8f f4 .(A@@.;..... 0020 21 48 d7 a0 01 bb f3 33 c8 e2 00 00 24 87 50 10 !H...3...\$.P.. 0030 f5 3c bd 69 00 00<1..						
wireshark_enp0s3_20220831110821_sY8sUs.pcapng Packets: 30 · Displayed: 30 (100.0%) Profile: Default						

Sigtrace from Core's side.

For reference, the signalling flow 5G SA is like this:



Congratulations! You have successfully set up and simulated 5G SA simulation from UE to the 5GC. If you want to explore more features beyond this document, you can open these two links:

https://kkohls.org/guides_open5gs.html

<https://open5gs.org/open5gs/docs/>

h. srsRAN Installation

This srsRAN installation guide is based on srsRAN installation with ZMQ Virtual Radio scenario from this link:

https://docs.srsran.com/en/latest/app_notes/source/zeromq/source/index.html

1. First you need to make new VM (ex: VM's name is srs) with the specification same as for core5g & ueran, then set up the VM's network configuration same as core5g & ueran
2. Install the required libraries with this command:

```
sudo apt-get install build-essential cmake libfftw3-dev libmbedtls-dev libboost-program-options-dev libconfig++-dev libsctp-dev libtool autoconf automake
```

3. Install the ZeroMQ development libraries installation can be installed with:

```
cd /home/srs/  
git clone https://github.com/zeromq/libzmq.git  
cd libzmq  
./autogen.sh  
./configure  
make  
sudo make install  
sudo ldconfig
```
4. Then you need install czmq, by using this command:

```
git clone https://github.com/zeromq/czmq.git  
cd czmq  
./autogen.sh  
./configure  
make  
sudo make install  
sudo ldconfig
```
5. Finally, you need to compile srsRAN (assuming you have already installed all the required dependencies). By using this command:

```
git clone https://github.com/srsRAN/srsRAN.git  
cd srsRAN  
mkdir build  
cd build  
cmake ../
```

Put extra attention in the “cmake ../” console output. Make sure you read the following line:

```
...  
-- FINDING ZEROMQ.  
-- Checking for module 'ZeroMQ'  
--   No package 'ZeroMQ' found  
-- Found libZEROMQ: /usr/local/include,  
/usr/local/lib/libzmq.so  
...  
-- libbladerF not found.  
-- FINDING SOAPY.  
-- Checking for module 'SoapySDR'  
--   No package 'SoapySDR' found  
-- libSOAPYSDR not found.  
-- FINDING ZEROMQ.  
-- Checking for module 'ZeroMQ'  
--   No package 'ZeroMQ' found  
-- Found libZEROMQ: /usr/local/include, /usr/local/lib/libzmq.so  
-- Boost version: 1.65.1  
-- Found the following Boost libraries:  
--   program_options  
-- SRSGUI_LIBRARIES SRSGUI_LIBRARIES-NOTFOUND  
-- SRSGUI_INCLUDE_DIRS SRSGUI_INCLUDE_DIRS-NOTFOUND  
-- Could NOT find SRSGUI (missing: SRSGUI_LIBRARIES SRSGUI_INCLUDE_DIRS)
```

make

```
.o
[ 97%] Building CXX object srsepc/src/mme/CMakeFiles/srsepc_mme.dir/nas.cc.o
[ 97%] Building CXX object srsepc/src/mme/CMakeFiles/srsepc_mme.dir/s1ap.cc.o
[ 97%] Building CXX object srsepc/src/mme/CMakeFiles/srsepc_mme.dir/s1ap_ctx_mn
gmt_proc.cc.o
[ 97%] Building CXX object srsepc/src/mme/CMakeFiles/srsepc_mme.dir/s1ap_erab_m
ngmt_proc.cc.o
[ 97%] Building CXX object srsepc/src/mme/CMakeFiles/srsepc_mme.dir/s1ap_mngmt_
proc.cc.o
[ 98%] Building CXX object srsepc/src/mme/CMakeFiles/srsepc_mme.dir/s1ap_nas_tr
ansport.cc.o
[ 98%] Building CXX object srsepc/src/mme/CMakeFiles/srsepc_mme.dir/s1ap_paging
.cc.o
[ 98%] Linking CXX static library libsrsepc_mme.a
[ 98%] Built target srsepc_mme
Scanning dependencies of target srsepc_hss
[ 98%] Building CXX object srsepc/src/hss/CMakeFiles/srsepc_hss.dir/hss.cc.o
[ 99%] Linking CXX static library libsrsepc_hss.a
[ 99%] Built target srsepc_hss
Scanning dependencies of target srsepc
[ 99%] Building CXX object srsepc/src/CMakeFiles/srsepc.dir/main.cc.o
[ 99%] Linking CXX executable srsepc
[ 99%] Built target srsepc
Scanning dependencies of target nr_phy_test
[100%] Building CXX object test/phy/CMakeFiles/nr_phy_test.dir/nr_phy_test.cc.o
[100%] Linking CXX executable nr_phy_test
[100%] Built target nr_phy_test
root@core5gs-VirtualBox: /home/core5gs/srsRAN/build#
```

```
sudo make install
srsran_install_configs.sh user
```

```
Installing srsRAN configuration files:
- Creating srsRAN config folder /root/.config/srsran
- Installing ue.conf.example in /root/.config/srsran/ue.conf
- Installing enb.conf.example in /root/.config/srsran/enb.conf
- Installing sib.conf.example in /root/.config/srsran/sib.conf
- Installing rr.conf.example in /root/.config/srsran/rr.conf
- Installing rb.conf.example in /root/.config/srsran/rb.conf
- Installing epc.conf.example in /root/.config/srsran/epc.conf
- Installing mbms.conf.example in /root/.config/srsran/mbms.conf
- Installing user_db.csv.example in /root/.config/srsran/user_db.csv
Done.
```

i. Open5Gs Configuration for 5G NSA

1. Open5Gs VM, then change user to root.
2. Change directory, then open and edit sample.yaml with this command:

```
cd /home/core5g/open5gs/build/configs
nano sample.yaml
```



```
GNU nano 4.8 sample.yaml
addr: 127.0.0.8

slap:
- addr: 10.14.2.10
gtpc:
- addr: 127.0.0.2
gummei:
  plmn_id:
    mcc: 510
    mnc: 14
    mme_gid: 2
    mme_code: 1
  tai:
    plmn_id:
      mcc: 510
      mnc: 14
    tac: 5300
  security:
    integrity_order : [ EIA2, EIA1, EIA0 ]
    ciphering_order : [ EEA0, EEA1, EEA2 ]

network_name:
```

Go to MME configuration section, then edit this parameter:

slap:

```
- addr: 10.14.2.10
```

gummei:

```
  plmn_id:
```

```
    mcc: 510
```

```
    mnc: 14
```

```
    mme_gid: 2
```

```
    mme_code: 1
```

tai:

```
  plmn_id:
```

```
    mcc: 510
```

```
    mnc: 14
```

```
    tac: 5300
```

j. srsRAN Configuration for 5G NSA

https://docs.srsran.com/en/latest/app_notes/source/zeromq/source/index.html

1. Create network namespace to receive IP pool from core's userplane:
`sudo ip netns add ue1`
using this command, you can verify the new "ue1" netns is exist:

```
sudo ip netns list
```

2. Open new terminal tab, then backup enb.conf, rr.conf, and ue.conf using this command:

```
cd /root/.config/srsran/
```

```
cp enb.conf enb.conf.backup
```

```
cp rr.conf rr.conf.backup
```

```
cp ue.conf ue.conf.backup
```

These configurations are default configuration after installation for 4G simulation.

3. Download three new enb.conf, rr.conf, and ue.conf from this link for 5G NSA simulation:

enb.conf:https://docs.srsran.com/en/latest/downloads/cbb9eed64bf03407cbb255828db3f5/enb_example.conf

rr.conf:

https://docs.srsran.com/en/latest/_downloads/b9b7009b9fda34684668e48dbfcc4f7d/rr_example.conf

ue.conf:

https://docs.srsran.com/en/latest/_downloads/c4c658121d530b085c7f1e53083e1d6c/ue_example.conf

after download these three files, please rename it first to enb.conf, rr.conf, and ue.conf.

The differences between original configuration & the new one:

-eNodeB configuration

device_name = zmq

device_args =

fail_on_disconnect=true,tx_port0=tcp://*:2000,rx_port0=tcp://localhost:2001,tx_port1=tcp://*:2100,rx_port1=tcp://localhost:2101,id=enb,base_srate=23.04e6

```
[rf]
#dl_arfcn = 3350
tx_gain = 80
rx_gain = 40

# Example for ZMQ-based operation with TCP transport for I/Q samples
device_name = zmq
device_args = fail_on_disconnect=true,tx_port0=tcp://*:2000,rx_port0=tcp://localhost:2001,tx_port1=tcp://*:2100,rx_port1=tcp://localhost:2101,id=enb,base_srate=23.04e6
```

-Radio configuration (rr.conf)

nr_cell_list =

```
(
{
    rf_port = 1;
    cell_id = 0x02; //you can modify this parameter as you
want (use hex format)
    tac = 0x0007; //you can modify this parameter as you
want, but need to be same with core configuration (use hex
format)
    pci = 500; //you can modify this parameter as you want
    root_seq_idx = 204;

    // TDD:
    //dl_arfcn = 634240;
    //band = 78;

    // FDD:
    dl_arfcn = 368500;
    band = 3;
}
);
```

```

nr_cell_list =
(
{
rf_port = 1;
cell_id = 0x02;
tac = 0x0007;
pci = 500;
root_seq_idx = 204;
// TDD:
//dl_arfcn = 634240;
//band = 78;
// FDD:
dl_arfcn = 368500;
band = 3;
}
);

```

- Put those three configurations to “/root/.config/srsran” with this command:

```

cp /home/srsran/Download/enb_example.conf
/root/.config/srsran/enb.conf
cp /home/srsran/Download/rr_example.conf
/root/.config/srsran/rr.conf
cp /home/srsran/Download/ue_example.conf
/root/.config/srsran/ue.conf

```

- Open and edit configuration files by using this command:

```

cd /root/.config/srsran/
nano enb.conf

```

```

GNU nano 2.9.3 /root/.config/srsran/enb.conf

# gtp_advertise_addr: IP address of eNB to advertise for
# s1c_bind_addr: Local IP address to bind for S1AP
# s1c_bind_port: Source port for S1AP connection (0)
# n_prb: Number of Physical Resource Blocks
# tm: Transmission mode 1-4 (TM1 default)
# nof_ports: Number of Tx ports (1 port default)
#
#####
[enb]
enb_id = 0x19B
mcc = 001
mnc = 01
mme_addr = 127.0.1.100
gtp_bind_addr = 127.0.1.1
s1c_bind_addr = 127.0.1.1
s1c_bind_port = 0
n_prb = 50
tm = 4
nof_ports = 2

```

Edit these parameters, as you want but must be same with core configuration:

```
mcc = 510
```

```
mnc = 14
```

```
mme_addr = 10.14.2.10 [open5gs IP]
```

```
gtp_bind_addr = 10.14.2.7 [srsRAN IP]
```

```

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.14.2.7 netmask 255.255.255.0 broadcast 10.14.2.255
    inet6 fd17:625c:f037:e02:a374:1456:501a:f59d prefixlen 64 scope

```

```
s1c_bind_addr = 10.14.2.7
```

```
GNU nano 4.8 /root/.config/srsran/enb.conf
#
#####
[enb]
enb_id = 0x19B
mcc = 510
mnc = 14
mme_addr = 10.14.2.10
gtp_bind_addr = 10.14.2.7
s1c_bind_addr = 10.14.2.7
s1c_bind_port = 0
n_prb = 50
#tm = 4
#nof_ports = 2
```

nano rr.conf

```
GNU nano 4.8 /root/.config/srsran/rr.conf
cell_list =
(
{
// rf_port = 0;
cell_id = 0x01;
tac = 0x14B4;
pci = 1;
// root_seq_idx = 204;
dl_earfcn = 3350;
//ul_earfcn = 21400;
ho_active = false;
```

4G anchor configuration, you can edit TAC parameter. In my case I try to use TAC 5300 in hex equals to 0x14B4

```
GNU nano 4.8 /root/.config/srsran/rr.conf
);
nr_cell_list =
(
{
rf_port = 1;
cell_id = 0x02;
tac = 0x14B4;
pci = 500;
```

5G cell configuration. 5G cell's TAC needs to be same as 4G anchor's TAC.

nano ue.conf

```
GNU nano 4.8 /root/.config/srsran/ue.conf
# imei: 15 digit International Mobile Station Equipment Identifier
# pin: PIN in case real SIM card is used
# reader: Specify card reader by it's name as listed in the reader module
#####
[usim]
mode = soft
algo = milenage
opc = E8ED289DEBA952E4283B54E88E6183CA
k = 465B5CE8B199B49FAA5F0A2EE238A6BC
imsi = 510140000000001
imei = 437081612381615
```

```
mode = soft
algo = milenage
opc = E8ED289DEBA952E4283B54E88E6183CA
k = 465B5CE8B199B49FAA5F0A2EE238A6BC
imsi = 510140000000001
imei = 437081612381615
```

You need to modify these parameter same as SIM card information in open5gs HSS.

k. 5G NSA Simulation

In this guide I integrate srsRAN with Open5gs, you can also simulate 5G NSA using core from srsRAN. If you want to try simulating 5G NSA using all program from srsRAN in one VM you can follow guide in this link

1. Turn on core5g VM, then change the user to root
2. Launch the open5gs with this command:
`cd /home/core5g/open5gs/build/tests/app/ && ./app`
3. Turn on the srs VM, then change the user to root
4. Launch the eNodeB by using this command:
`cd /home/srs/srsRAN/build
./srsenb/src/srsenb --rf.device_name=zmq --
rf.device_args="fail_on_disconnect=true,tx_port0=tcp://
:2000,rx_port0=tcp://localhost:2001,tx_port1=tcp://
:2100,rx_port1=tcp://localhost:2101,id=enb,base_srate=
23.04e6"`

Then, check the message in srsRAN terminal and open5gs terminal

```
Active RF plugins: librsran_rf_zmq.so
Inactive RF plugins:
--- Software Radio Systems LTE eNodeB ---

Couldn't open , trying /root/.config/srsran/enb.conf
Reading configuration file /root/.config/srsran/enb.conf...
Couldn't open sib.conf, trying /root/.config/srsran/sib.conf
Couldn't open rr.conf, trying /root/.config/srsran/rr.conf
Couldn't open rb.conf, trying /root/.config/srsran/rb.conf

Built in Release mode using commit ce8a3cae1 on branch master.

Opening 2 channels in RF device=zmq with args=fail_on_disconnect=true,tx_
port0=tcp://*:2000,rx_port0=tcp://localhost:2001,tx_port1=tcp://*:2100,rx
_port1=tcp://localhost:2101,id=enb,base_srate=23.04e6
Supported RF device list: zmq file
CHx base_srate=23.04e6
CHx id=enb
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decima
tion)
CH0 rx_port=tcp://localhost:2001
CH0 tx_port=tcp://*:2000
```

Message from srsRAN should be like this

```
08/31 16:16:38.693: [mme] INFO: eNB-S1 accepted[10.14.2.7]:54287 in s1_path module (../
src/mme/s1ap-sctp.c:114)
08/31 16:16:38.693: [mme] INFO: eNB-S1 accepted[10.14.2.7] in master_sm module (../src/
mme/mme-sm.c:153)
08/31 16:16:38.693: [mme] INFO: [Added] Number of eNBs is now 1 (../src/mme/mme-context
.c:1840)
08/31 16:16:38.693: [mme] INFO: eNB-S1[10.14.2.7] max_num_of_ostreams : 30 (../src/mme/
mme-sm.c:195)
```

Message from open5gs should be like this

5. Launch the UE simulation by using this command:
`sudo ./srsue/src/srsue --rf.device_name=zmq --
rf.device_args="tx_port0=tcp://*:2001,rx_port0=tcp://l
ocalhost:2000,tx_port1=tcp://*:2101,rx_port1=tcp://loc
alhost:2100,id=ue,base_srate=23.04e6" --gw.netns=ue1`

Then, check the message in the terminal of srsran and open5gs

```

root@lab5... x root@lab5... x root@lab5... x root@lab5... x
Waiting PHY to initialize ... done!
Attaching UE...
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decima
tion)
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decima
tion)
.
Found Cell: Mode=FDD, PCI=1, PRB=50, Ports=1, CP=Normal, CFO=-0.2 KHz
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decima
tion)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decima
tion)
Found PLMN: Id=51014, TAC=5300
Random Access Transmission: seq=19, tti=341, ra-rnti=0x2
RRC Connected
Random Access Complete. c-rnti=0x46, ta=0
Network attach successful. IP: 10.45.0.4
nTp) 31/8/2022 9:20:39 TZ:130
RRC NR reconfiguration successful.
Random Access Transmission: prach_occasion=0, preamble_index=0, ra-rnti=0
xf, tti=1131
Random Access Complete. c-rnti=0x4601, ta=0

```

Message from srsUE should be like this

```

==== eNodeB started ===
Type <t> to view trace
RACH: tti=341, cc=0, pci=1, preamble=19, offset=0, temp_crnti=0x46
User 0x46 connected
User 0x46 connected
User 0x46 connected
RACH: slot=1131, cc=0, preamble=0, offset=0, temp_crnti=0x4602

```

Message from srsRAN should be like this

```

context.c:891)
08/31 16:20:39.356: [gtp] INFO: gtp_connect() [127.0.0.4]:2123 (../lib/gtp/path.c:60)
08/31 16:20:39.356: [smf] INFO: [Added] Number of SMF-UEs is now 2 (../src/smf/context.
c:892)
08/31 16:20:39.356: [smf] INFO: [Added] Number of SMF-Sessions is now 2 (../src/smf/con
text.c:2972)
08/31 16:20:39.356: [smf] INFO: UE IMSI[510140000000001] APN[internet] IPv4[10.45.0.4]
IPv6[] (../src/smf/s5c-handler.c:255)
08/31 16:20:39.372: [upf] INFO: [Added] Number of UPF-Sessions is now 2 (../src/upf/con
text.c:178)
08/31 16:20:39.372: [gtp] INFO: gtp_connect() [127.0.0.6]:2152 (../lib/gtp/path.c:60)
08/31 16:20:39.372: [upf] INFO: UE F-SEID[CP:0x3 UP:0x3] APN[internet] PDN-Type[1] IPv4
[10.45.0.4] IPv6[] (../src/upf/context.c:397)
08/31 16:20:39.372: [upf] INFO: UE F-SEID[CP:0x3 UP:0x3] APN[internet] PDN-Type[1] IPv4
[10.45.0.4] IPv6[] (../src/upf/context.c:397)
08/31 16:20:39.373: [gtp] INFO: gtp_connect() [127.0.0.7]:2152 (../lib/gtp/path.c:60)
08/31 16:20:39.670: [emm] INFO: [510140000000001] Attach complete (../src/mme/emm-sm.c:
1018)
08/31 16:20:39.670: [emm] INFO: IMSI[510140000000001] (../src/mme/emm-handler.c:237
)
08/31 16:20:39.670: [emm] INFO: UTC [2022-08-31T09:20:39] Timezone[0]/DST[0] (../sr
c/mme/emm-handler.c:243)
08/31 16:20:39.670: [emm] INFO: LOCAL [2022-08-31T16:20:39] Timezone[25200]/DST[0]
(../src/mme/emm-handler.c:247)
08/31 16:20:39.671: [gtp] INFO: gtp_connect() [10.14.2.7]:2152 (../lib/gtp/path.c:60)

```

Message from Open5Gs should be like this

6. To make service flow from UE to core, you can try ping with this command:
`sudo ip netns exec ue1 ping 10.45.0.1`

(It is okay, if there is no reply message. You can check from srsRAN trace. If the brate value is not zero, it's means UE get the downlink packet from core)

[Additional]

7. Check trace from srsRAN, after srsRAN is running press "t" then enter. You will see this trace:

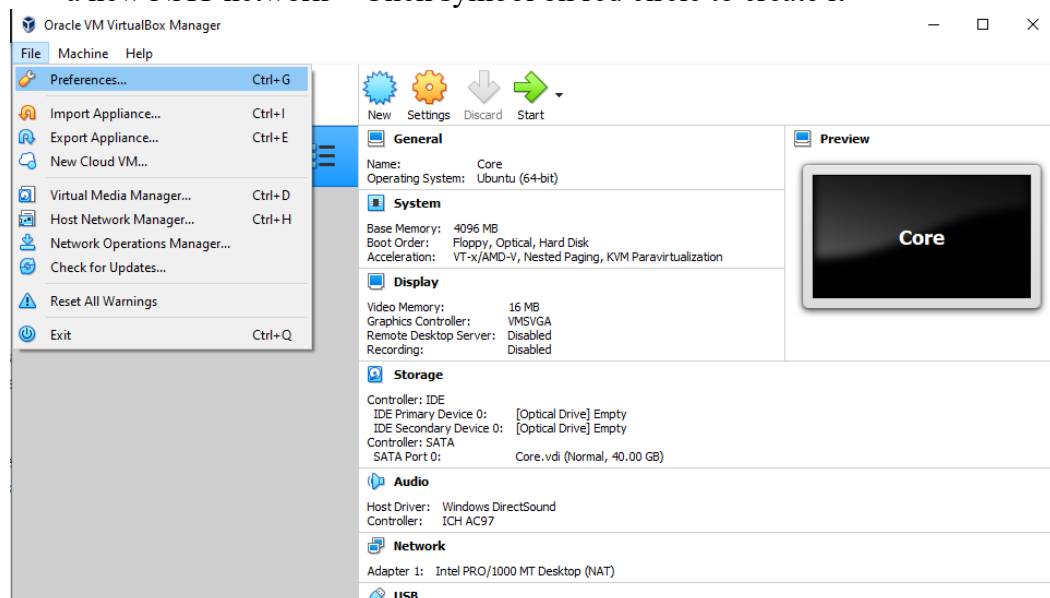
```
t
Enter t to stop trace.

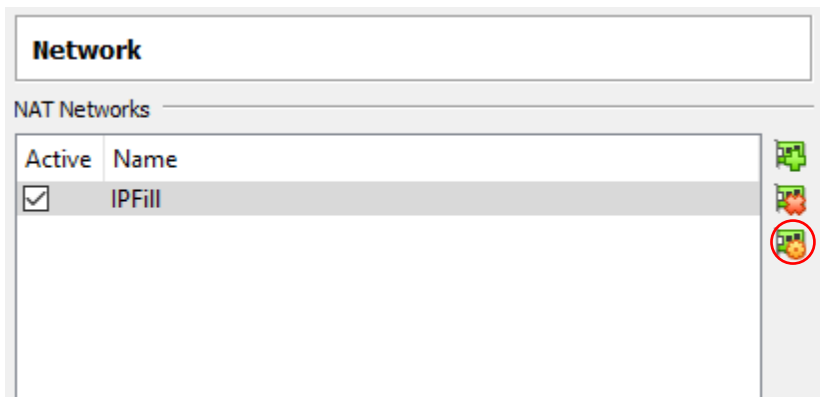
-----DL-----|-----
---UL-----
rat pci rnti cqi ri mcs brate ok nok (%) | pusch pucch phr mc
s brate ok nok (%) bsr
lte 1 47 15 0 0 0 0 0 0% | n/a 99.9 0
0 0 0 0 0% 0.0
nr 0 4603 n/a 0 28 62k 1 0 0% | 59.2 99.9 0 2
8 202k 3 0 0% 0.0
lte 1 47 15 0 0 0 0 0 0% | n/a 99.9 0
0 0 0 0 0% 0.0
nr 0 4603 n/a 0 28 61k 1 0 0% | 59.5 99.9 0 2
8 198k 3 0 0% 0.0
lte 1 47 15 0 0 0 0 0 0% | n/a 99.9 0
0 0 0 0 0% 0.0
nr 0 4603 n/a 0 28 63k 1 0 0% | 59.1 99.9 0 2
8 206k 3 0 0% 0.0
```

You can also check the trace from wireshark.

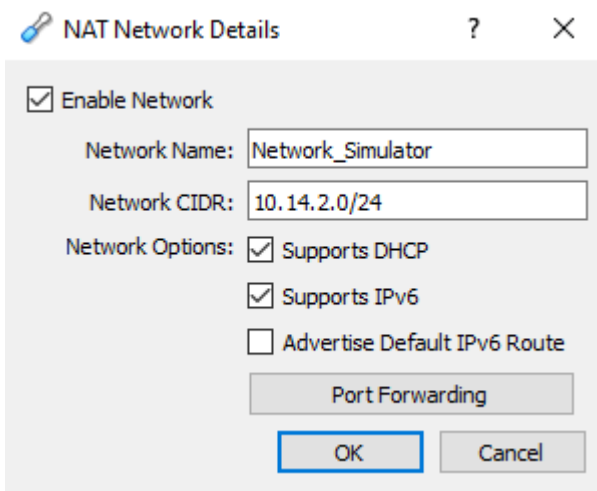
I. VM Port Forwarding Configuration

- a. Please make sure all VMs already turned down.
- b. Go to global setting in Virtualbox, click File > Preferences > Network and add a new NAT network > Click symbol on red circle to create it





c. Go to setting by click third symbol from top, Click port forwarding.



d. Go to setting by click third symbol from top, Click port forwarding. Add new configuration by clicking button with green plus icon. Then, make configuration like this configuration below

