Dockerized VoLTE Setup

# Prepare SIM cards for VoLTE

* Nhóm chuẩn bị một SIM trắng có mã số 11 để thực hiện công việc thử nghiệm VoLTE. Cụ thể:
  + IMSI: 452041234567811
  + K: 465b5ce8b199b49faa5f0a2ee238a6bc
  + Op: 63BFA50EE6523365FF14C1F45F88737D
  + MSISDN: 0198765432100

# Build Open5GS, Kamailio with docker-compose

* Trước khi cài ta cần chuẩn bị docker-ce và docker-compose to run multiple container
* **Với docker-ce:**

Install using the repository

Setup the repository:

Install packages to allow apt to use a repository over HTTPS:

$sudo apt-get update

$sudo apt-get install \

ca-certificates \

curl \

gnupg \

lsb-release

Add Docker’s official GPS key:

$sudo mkdir -p /etc/apt/keyrings

$curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

Use the following command to set up the repository:

$echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Install Docker Engine:

$sudo apt-get update

$sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

Depend on the Ubuntu’s version, with Ubuntu 22.4, I had problem when installed …(check lai)

Check if successful by running: $sudo docker run hello-world

* **docker-compose:**

$curl -SL https://github.com/docker/compose/releases/download/v2.13.0/docker-compose-linux-x86\_64 -o /usr/local/bin/docker-compose

* Tiếp đến git clone repository và build các image tương ứng các thành phần trong khối Open5gs và IMS.
* Git clone từ trang web chứa các thư mục ta muốn cài đặt

$sudo git clone <https://github.com/herlesupreeth/docker_open5gs>

* Sau khi git clone từ github về ta bắt đầu vào từng các thư mục tương ứng để build tự động các image nhờ các Dockerfile.m

Build các khối trong open5gs

cd docker\_open5gs/base

docker build --no-cache --force-rm -t docker\_open5gs .

Build các khối trong ims

cd ../ims\_base

docker build --no-cache --force-rm -t docker\_kamailio .

Buid cho E-UTRAN

cd ../srslte

docker build --no-cache --force-rm -t docker\_srslte .

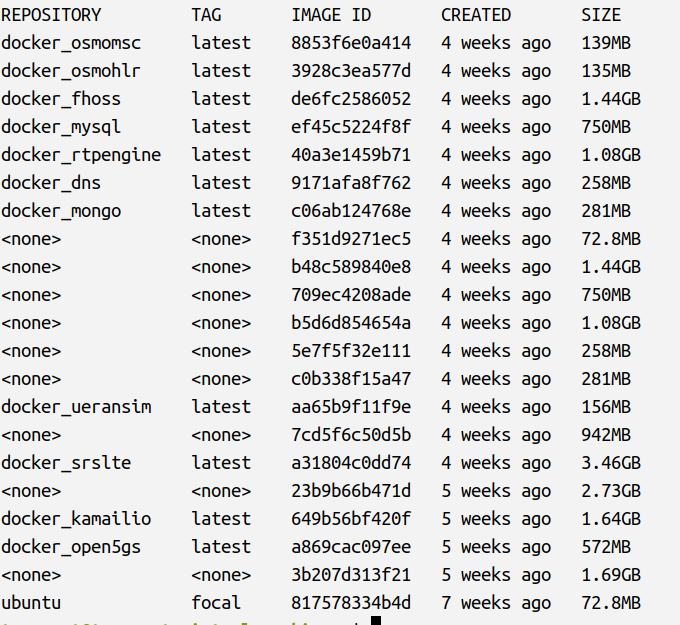
Build cho SIM

cd ../ueransim

docker build --no-cache --force-rm -t docker\_ueransim

Check công việc build các image đã thành công hay chưa, ta sử dụng câu lệnh

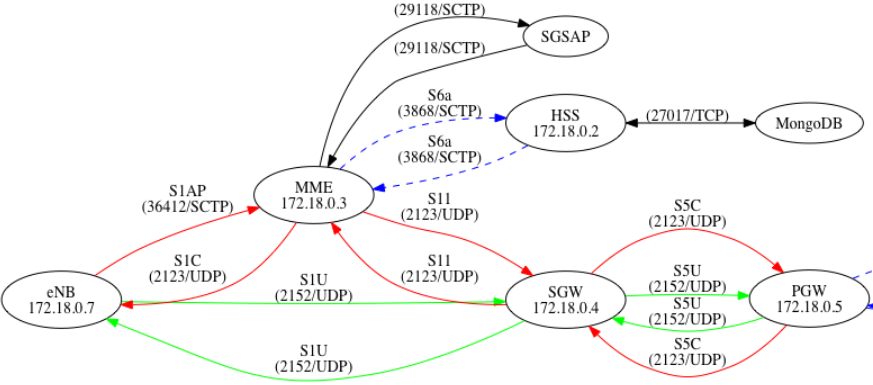
$sudo docker images



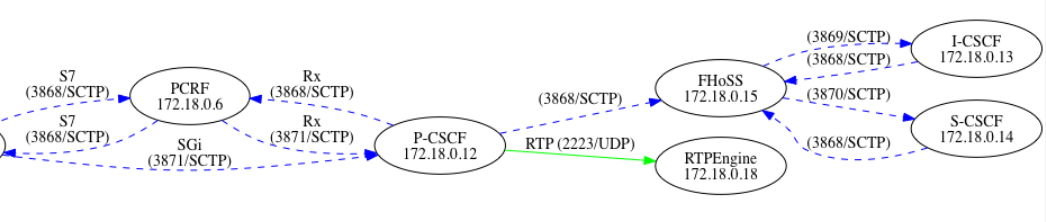
* + - * Việc cài đặt bước đầu đã hoàn thành

# Configuring your setup

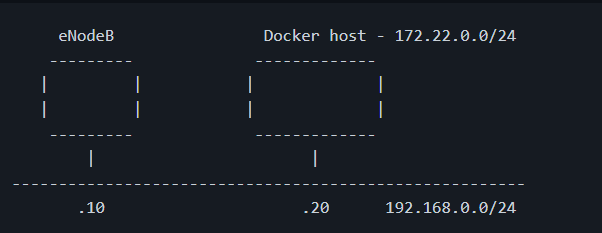
* Hầu hết việc cài đặt cấu hình cho toàn bộ khối, cụ thể là các IP là ở file .env
* Toàn bộ IP các khối được lưu trữ trong này, muốn thay đổi theo ý muốn ta chỉ cần config IP. Nhóm setup mặc định mà file đã tạo sẵn, cụ thể theo hình vẽ này:
  + Khối EnodeB và các EPC:



* Khối IMS



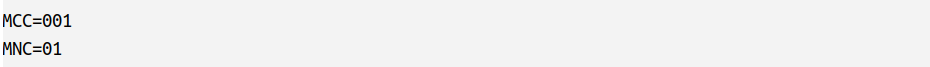
* + - * Đó là các địa chỉ IP các khối trong miền mạng ảo 172.22.0.0/24



* Đối với eNodeB, hãy đặt định tuyến tĩnh thành eNodeB cho các gói đi từ eNodeB đến mạng máy chủ Docker

#ip route add 172.22.0.0/24 via 192.168.0.20

* Nếu eNodeB không có chức năng thiết lập định tuyến tĩnh, hãy xây dựng mạng để Docker host trở thành GW mặc định.
* Tiếp theo cần lưu ý việc setup các tham số sau bởi nó sẽ ảnh hưởng tới việc kết nối test hệ thống khi bắt sóng điện thoại:

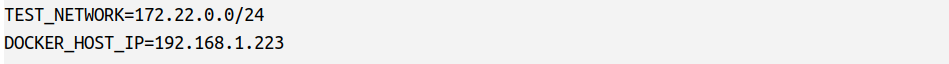


* MCC =452
* MNC = 004

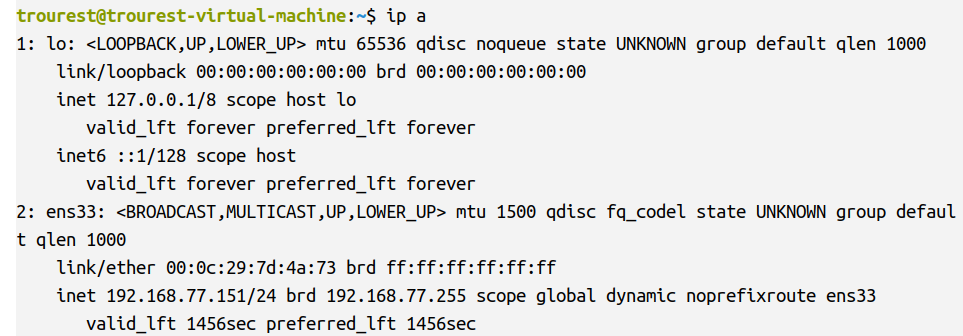
MCC : Mobile Country Code

MNC: Mobile Network Code

* + - * Các tham số được sử dụng xác thực duy nhất cho một thuê bao di động
* Ngoài ra còn các tham số khác như DOCKER\_HOST\_IP và SGWU\_ADVERTISE\_IP:



Để có thể check DOCKER\_HOST\_IP, ta dùng câu lệnh:



* Trong trường hợp eNB không chạy trong cùng một mạng/máy chủ docker với máy chủ đang chạy Core + IMS dockerized thì hãy làm thêm các bước bổ sung sau:
* Under mme section in docker compose file (docker-compose.yaml, nsa-deploy.yaml), uncomment the following part

...

# ports:

# - "36412:36412/sctp"

Under sgwu section in docker compose file (docker-compose.yaml, nsa-deploy.yaml), uncomment the following part

...

# ports:

# - "2152:2152/udp"

...

In docker compose file docker-compose.yaml, config

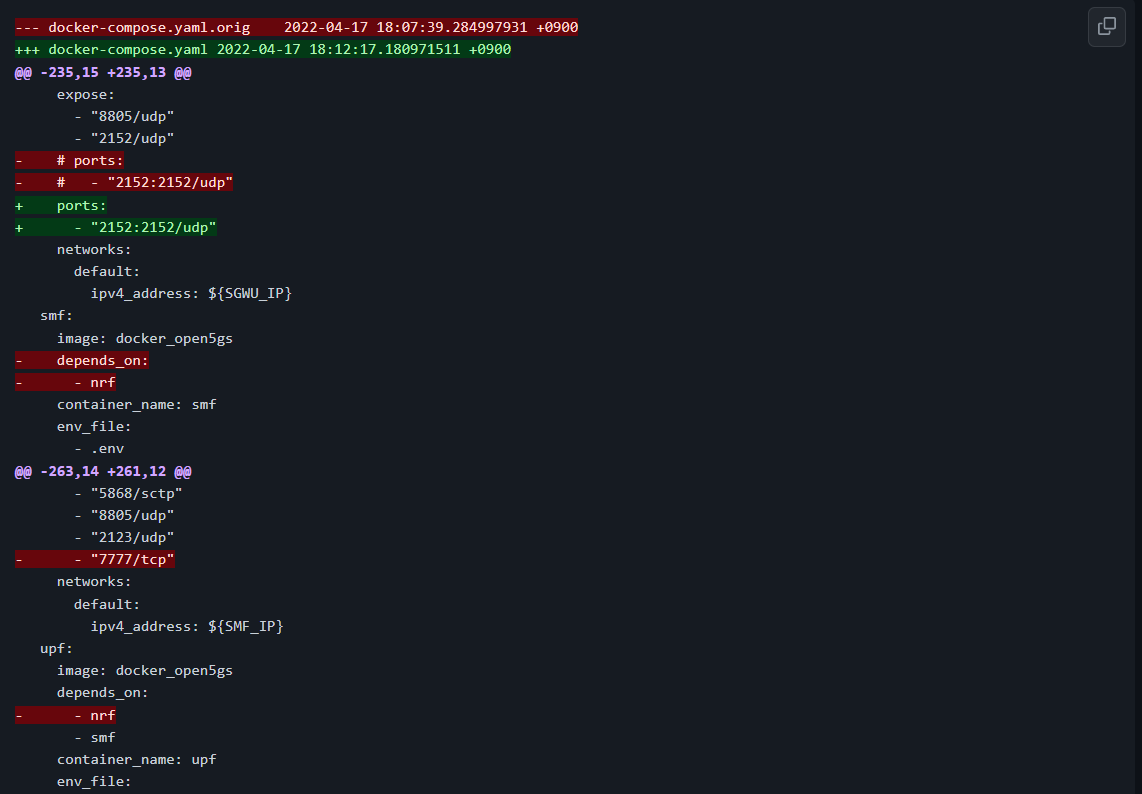
...

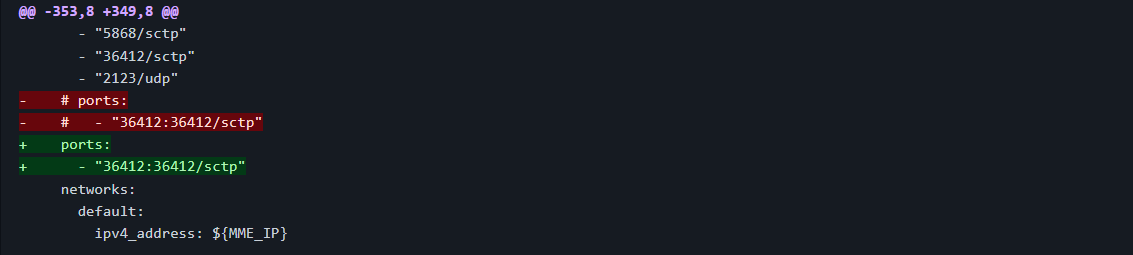
image: docker\_kamailio //change to open5gs\_kamailio

...

**Docker-compose.yaml**

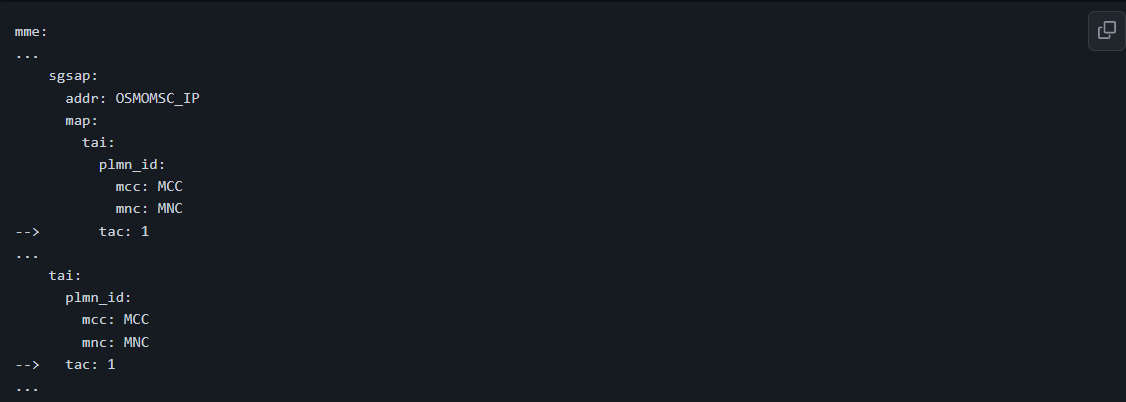
# Publish the following ports of SGW-U and MME. And remove the connection settings and dependencies to NRF





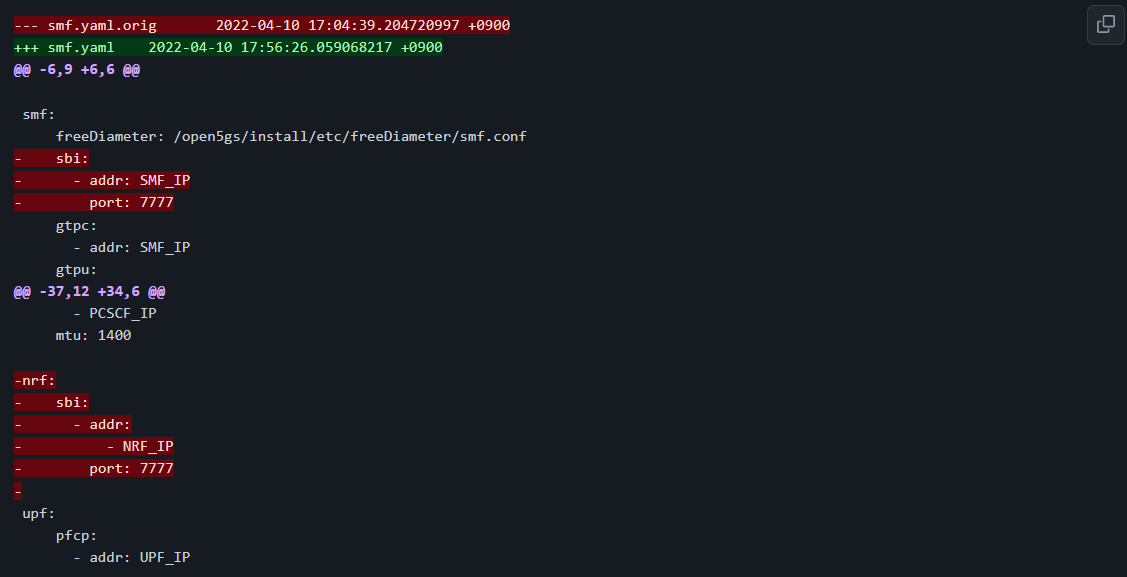
**mme/mme.yaml**

Change the following tac according to the TAC of eNodeB



**smf/smf.yaml**

Remove the binding on SBI and the connection settings to NRF



# Building 4G/5G Core + IMS related components images

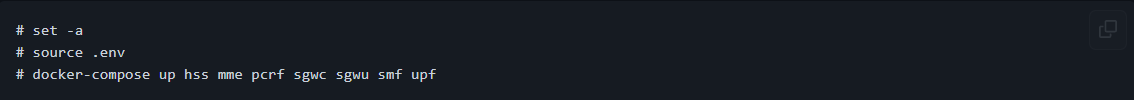
Terminal#1



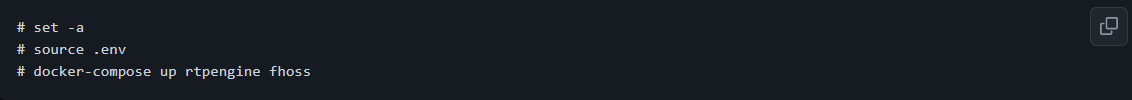
Terminal#2



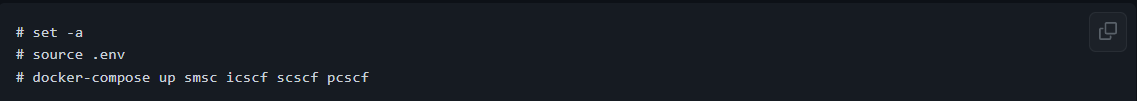
Terminal#3



Terminal#4

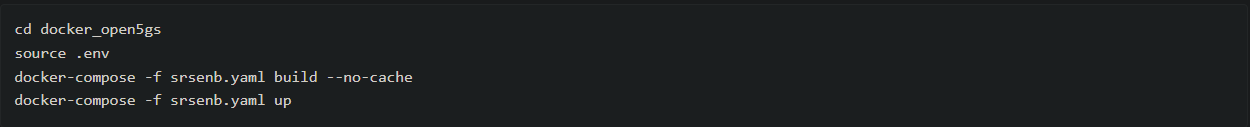


Terminal#5



# (Optional) Run srsENB in a separate container

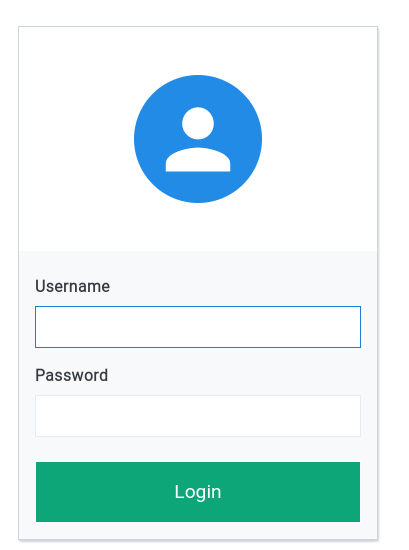
- Trong trường hợp nhóm sử dụng srsENB and B210.



# Configuration and register UE

## Đăng kí User trên WebUI

Ta mở địa chỉ mà WebUI khởi chạy qua cổng port TCP 3000 trong web brower, tại đó IP là của máy ảo đang chạy Open5gs containers. Khi đăng nhập, sử dụng tài khoản Username:admin và Password:1423



== Chụp ảnh đã đăng kí ở dây.

## Đăng kí IMSI và MSISDN với OsmoHLR

OsmoHLR, the Osmocom HLR is a minimal implementation of the Home Location Register, the subscriber database of a 2G/3G cellular network. It stores parameters such as the IMSI and MSISDN of each subscriber, together with the secret key material (Ki, K/OP/OPc) in order to generate authentication vectors.

OsmoHLR **does not support the classic TCAP/MAP protocol** but rather implements the more light-weight [GSUP](https://osmocom.org/projects/cellular-infrastructure/wiki/GSUP) protocol.

* Đầu tiên cần kiểm tra port TCP OsmoHLR ở đâu để ta sẽ thực thi trong container này. Sử dụng câu lệnh sau để kiểm tra. (Với các khối đã được chạy)









* Bắt đầu quá trình thực thi đăng kí trong container

Examples:

$ telnet localhost 4258

OsmoHLR> enable

OsmoHLR# subscriber imsi 452041234567811 create

% Created subscriber 452041234567811

ID: 1

IMSI: 452041234567811

MSISDN: none

OsmoHLR# subscriber imsi 452041234567811 update msisdn 0198765432100

% Updated subscriber IMSI='452041234567811' to MSISDN='0198765432100'

OsmoHLR# subscriber msisdn 0198765432100 update aud3g milenage k 465b5ce8b199b49faa5f0a2ee238a6bc op 63BFA50EE6523365FF14C1F45F88737D

OsmoHLR# subscriber msisdn 0198765432100 show

ID: 1 IMSI: 452041234567811

MSISDN: 0198765432100

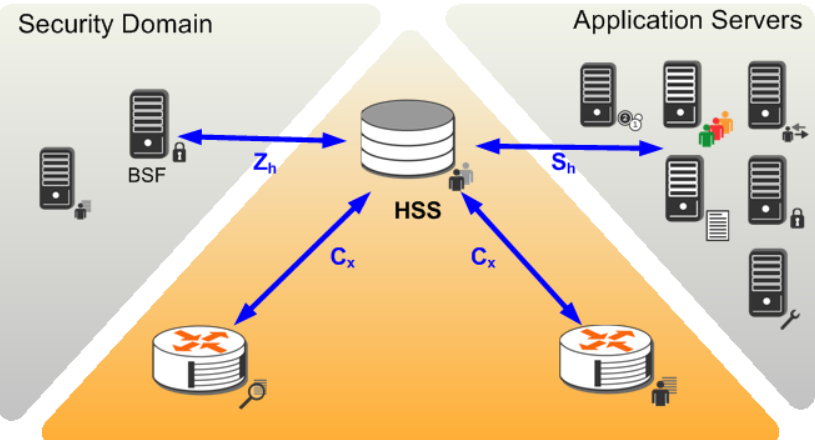
3G auth: MILENAGE

K=465b5ce8b199b49faa5f0a2ee238a6bc

OP=63BFA50EE6523365FF14C1F45F88737D

IND-bitlen=5

## Đăng kí IMS subscription với FhoSS



The FHoSS implementation defines the Java version of three interfaces:

* The Sh interface, which can be used by application servers to access the HSS
* The Zh interface
* The Cx interface, wich is used to communicate with I-CSCF and S-CSCF

Example: ??????

# Lỗi và cách khắc phục

* 1. Đã fix
  2. Hiện tại

1. Cách chạy

$sudo su

#cd docker\_open5gs

#source .env

#docker compose up

Open <http://3000>

Open wireshark to capture

Connect SDR circuit to this virtual machine then run command in host:

$cd docker\_open5gs

$srsenb