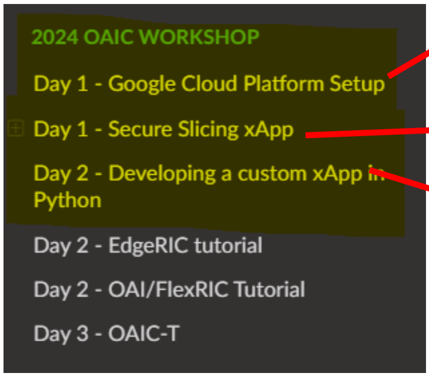
Tutorial link:

https://openaicellular.github.io/oaic/



2024 OAIC WORKSHOP

Day 1 - Google Cloud Platform Setup

Day 1 - Secure Slicing xApp

Day 2 - Developing a custom xApp in Python

Day 2 - EdgeRIC tutorial

Day 2 - OAI/FlexRIC Tutorial

Day 3 - OAIC-T

Required steps for creating VM on Google cloud. I have tried installation on VCL but the setup didn't work. Google cloud is the safe alternative.

This tutorial contains many the prerequisites.

This tutorial explains installation process for a dummy xApp that assumes the interference and change MCS for BS and doesn't contain the model. There might be other missing pieces as well.

The following are the notes I made while doing the setup. Please refer to the tutorial for all the steps. I am summarizing the portions of tutorial and some changes that we are supposed to do.

Legends:

Green text: Follow all the steps
Blue text: Follow the notes
Red text: Ignore as these steps are not needed

============================Day 1 - Google Cloud Platform Setup=====================

============================Day 1 - Secure Slicing xApp: ============================

Install packages

Setup - Near-Realtime RIC

Setup - Srslte with E2 Agent- srsLTE might be redundant but the first part (velichkov_s1ap_plus_option_group) is definitely required. Need to check if the srsLTE installation creates problem with running srsRAN.

Setup - Nginx Web Server

Setup - Secure Slicing xApp

Setup - 5G Network

Deploying the SS xApp

Run only these:

```
export KONG_PROXY=`sudo kubectl get svc -n ricplt -l app.kubernetes.io/name=kong -o jsonpath='{.items[0].spec.clusterIP}'`
```

```
export E2MGR_HTTP=`sudo kubectl get svc -n ricplt --field-selector metadata.name=service-ricplt-e2mgr-http -o jsonpath='{.items[0].spec.clusterIP}'`
```

```
export APPMGR_HTTP=`sudo kubectl get svc -n ricplt --field-selector metadata.name=service-ricplt-appmgr-http -o jsonpath='{.items[0].spec.clusterIP}'`
```

```
export E2TERM_SCTP=`sudo kubectl get svc -n ricplt --field-selector metadata.name=service-ricplt-e2term-sctp-alpha -o jsonpath='{.items[0].spec.clusterIP}'`
```

```
export ONBOARDER_HTTP=`sudo kubectl get svc -n ricplt --field-selector metadata.name=service-ricplt-xapp-onboarder-http -o jsonpath='{.items[0].spec.clusterIP}'`
```

```
export RTMGR_HTTP=`sudo kubectl get svc -n ricplt --field-selector metadata.name=service-ricplt-rtmgr-http -o jsonpath='{.items[0].spec.clusterIP}'`
```

=========================Day 2 - Developing a custom xApp in Python=================

Setup-Compiling E2-like srsRAN:

Before running the mentioned commands, move to oaic directory.

```
cd ~/oaic     (1)
```

Do NOT run all the commands mentioned in tutorial together [do not simply copy and paste all commands together].

“sudo srsran_install_configs.sh user –force” writes config files and hence ask user “y/n” during installation. The subsequent command “cd ../../” somehow interfere with previous command and results in “not writing” the required files.

Solution:

Run these together: (2)

```
cd srsRAN-e2
mkdir build
export SRS=`realpath .`
cd build
cmake ../ -DCMAKE_BUILD_TYPE=RelWithDebInfo \
   -DENABLE_E2_LIKE=1 \
   -DENABLE_AGENT_CMD=1 \
   -DRIC_GENERATED_E2AP_BINDING_DIR=${SRS}/e2_bindings/E2AP-v01.01 \
   -DRIC_GENERATED_E2SM_KPM_BINDING_DIR=${SRS}/e2_bindings/E2SM-KPM \
   -DRIC_GENERATED_E2SM_GNB_NRT_BINDING_DIR=${SRS}/e2_bindings/E2SM-GNB-NRT
```

```
make -j`nproc`
sudo make install
sudo srsran_install_configs.sh user –force
```

after pressing “y” run:

```
cd ../../   (3)
```

Setup-Creating RAM filesystem

Development => Later, we need to refer this section for making changes and adding AI model

Deployment-Building the Docker image

change directory first: ~/oaic/ric-app-ml-e2like [this contains the dockerfile]

Deployment-Creating the xApp config

Deployment-Finding local IP address

Deployment-Configuring the Nginx Web server

Deployment-Hosting the config files

Deployment-create onboard URL file

Deployment-onboard and deploy the xApp

Connecting to srsRAN

Please remember that this step requires running individual commands in new terminals.

In every terminal, first change directory:

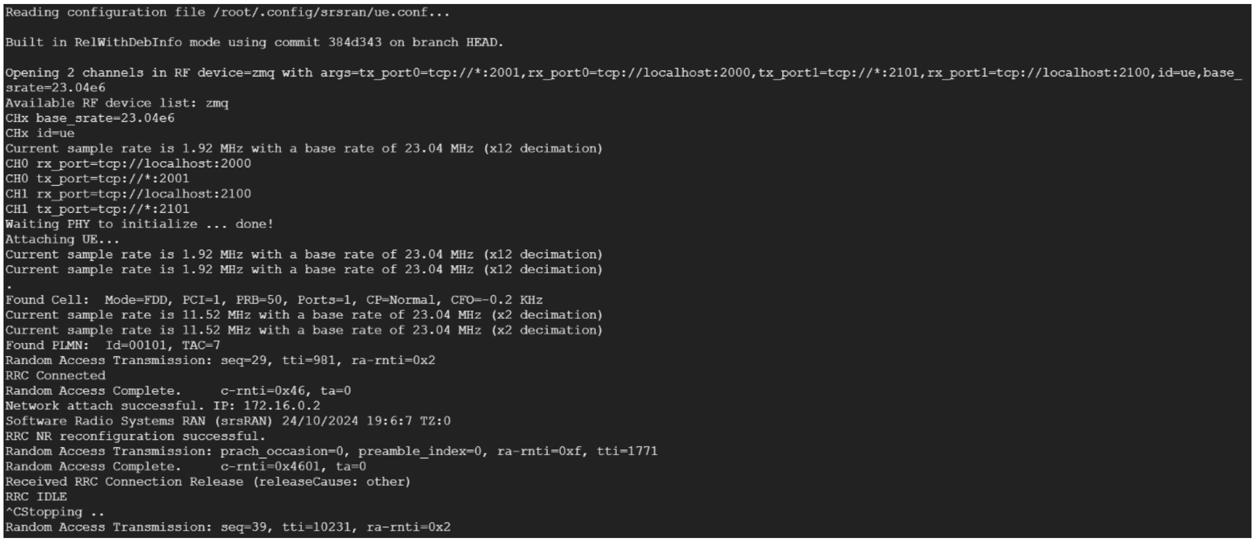
```
cd ~/oaic
cd srsRAN-e2/build
```

check the value of variable “HOST_IP” and “XAPP_PORT” wherever needed.

To exit: sudo killall -s9 srsepc srsenb srsue

I did not see the expected output at this stage. Please try at your end. I started iperf server for base station and iperf client for UE side, however I didn’t observe the expected data flow between them and client returns time out. When I tried to run the experiment again by stopping and restarting base station (srsenb), I am getting the error I posted on piazza.

Even in my second attempt, I cannot send uplink data and connection is closed with following message at UE side:



```
Reading configuration file /root/.config/srsran/ue.conf...

Built in RelWithDebInfo mode using commit 384d343 on branch HEAD.

Opening 2 channels in RF device=zmq with args=tx_port0=tcp://*:2001,rx_port0=tcp://localhost:2000,tx_port1=tcp://*:2101,rx_port1=tcp://localhost:2100,id=ue,base_
srate=23.04e6
Available RF device list: zmq
CHx base_srate=23.04e6
CHx id=ue
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://localhost:2000
CH0 tx_port=tcp://*:2001
CH1 rx_port=tcp://localhost:2100
CH1 tx_port=tcp://*:2101
Waiting PHY to initialize ... done!
Attaching UE...
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
.
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 decimation)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Found PLMN:  Id=00101, TAC=7
Random Access Transmission: seq=29, tti=981, ra-rnti=0x2
RRC Connected
Random Access Complete.     c-rnti=0x46, ta=0
Network attach successful. IP: 172.16.0.2
Software Radio Systems RAN (srsRAN) 24/10/2024 19:6:7 TZ:0
RRC NR reconfiguration successful.
Random Access Transmission: prach_occasion=0, preamble_index=0, ra-rnti=0xf, tti=1771
Random Access Complete.     c-rnti=0x4601, ta=0
Received RRC Connection Release (releaseCause: other)
RRC IDLE
^CStopping ..
Random Access Transmission: seq=39, tti=10231, ra-rnti=0x2
```