INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

FOCUS GROUP ON AUTONOMOUS NETWORKS (FG-AN)

AN-I-163

Original: English

Question(s):	N/A	Virtual, 3-5 November 2021

INPUT DOCUMENT

Title:	Team “RAN-RIC-xApp” presentation – Build-a-thon – Activity-4- ORAN Control Loop Instantiation	
Contact:	Deena Mukundan	Email: deenamukund@gmail.com
Contact:	Divyani R Achari	Email: divyaniraj1d@gmail.com

Keywords: Build-a-thon, Model, Control Loop, xApp

Abstract: This contribution is a report on activities of team “RAN-RIC-xApp” towards the Build-a-thon hosted by ITU FG AN in ITU AI/ML in 5G Challenge (2021). It will cover the results obtained so far, and the open problems faced by the team.

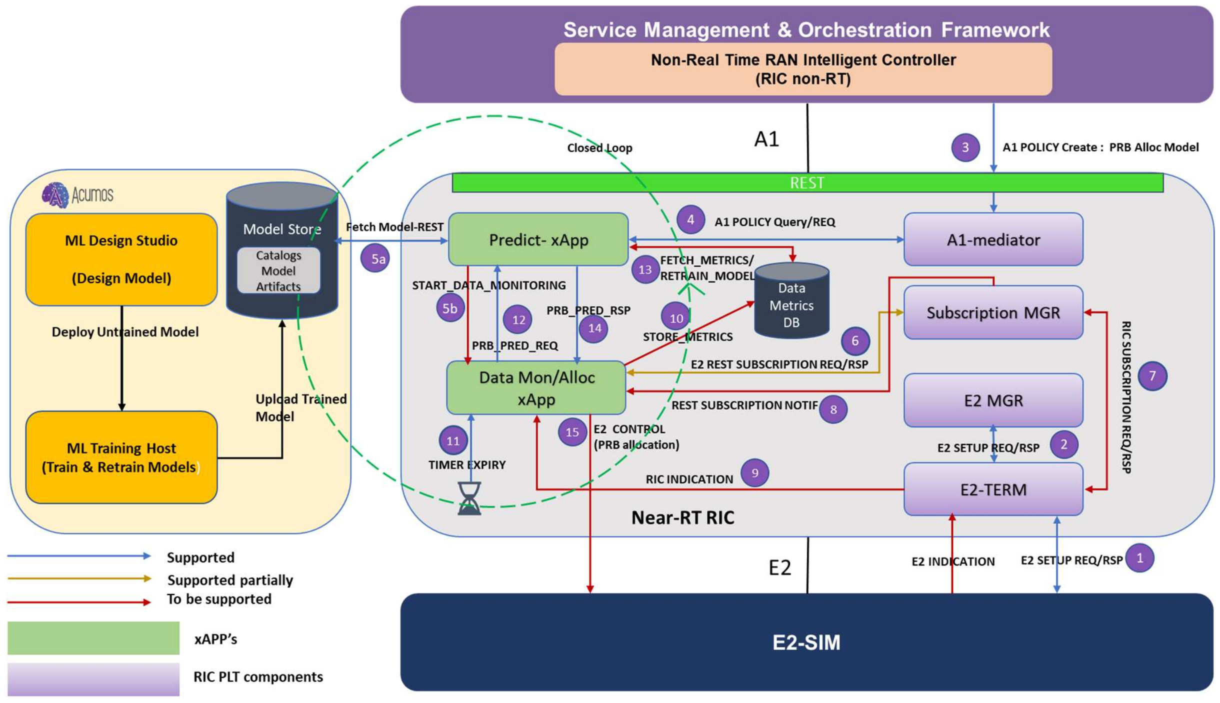
1. Introduction

As discussed in [Build-a-thon FG AN], the "RAN-RIC-xApp" team registered for the Build-a-thon problem statement and worked to produce proof of concept (PoC) demo. This document is the report from the build-a-thon activity from team "RAN-RIC-xApp" done as part of the ITU AI/ML in 5G Challenge.

2. Problem statement

1. Goal-1 According to the given requirements fetch model from Acumos
2. Goal-2 Deploy the model as xApp in ORAN. A pre-trained model might be used for this purpose

3. Solution Workflow



Note: E2 Indication is for future reference, currently data is not monitored via RIC Indication. In future, data needs to be monitored and sent to predict xApp.

Brief Work Summary:-

- RIC is UP and Running, RAN (E2-SIM in this case) is registered/associated with RIC
- RIC receives policy update from A1 for triggering closed loop PRB Allocation
- Fetch model based on the A1 Policy details
- Based on RAN data monitored, predict PRB utilisation [test data was used for POC instead of actual data from E2]
- Compute the PRB to be allocated and send E2 control message. PRB's are always reserved for Emergency Slide and additionally resources can be reallocated based on situational considerations
- Continuously monitor, evaluate and improve decision

Work Flow as defined in above figure :-

- Points 1&2 show E2 SIM is up and association with RIC is setup
- Point 3 nRT RIC receives A1 policy update to trigger closed loop monitoring
- Point 4 A1-mediator sends A1 Policy REQ to prbpred xApp
- Point 5a, Point 5b show the model is fetched from model store as per policy guidelines and prbpred instructs DataMon/Alloc xApp to start monitoring the data
- Point 6,7,8 shows the messaging done for subscribing to E2 for data
- Point 9 shows Data reception from E2 node. The received metrics are stored in metrics DB as in Point 10
- Upon timer expiry as in Point 11, request for Prediction is sent to prbpred xApp as in Point 12.
- prbpred uses ML model to predict the future utilisation. Based on new data model retraining may be done. Predicted values may be send to Datamon/Alloc xApp as in Points 13,14.
- DataMon/Alloc xApp computes the PRB to be allocated and sends E2 control message towards E2 as in Point 15

4. Activity Results

Disclaimer:

The pre-trained model, model specific implementation and PRB allocation ALG1 developed by Team "AUTOMATO" as part of this build-a-thon is re-used in the below xApps.
Please refer References [5] & [6] for details on report from **Team "AUTOMATO".** The credit for the model implementation, design of closed loop working and PRB allocation scheme goes to **Team "AUTOMATO**".

The primary focus of "RAN-RIC-xApp" team was as below: -

1) Develop understanding on ORAN-RIC Platform and bring up a fully functional setup
2) Understand, explore xApp deployment onto RIC Platform
3) Explore implementing different xApp variants (both reactive and proactive)
4) Explore messaging communication between xApp's and RIC components

As part of this following points was achieved: -

1) ORAN-RIC platform was brought up with Dawn release content
2) Understanding of XApp onboarding/deployment process and RIC platform components was achieved

   The below xApps are developed based on xApp Framework for Python. Separate xApp-descriptor files were defined detailing on the configuration, RX & TX messages supported.

   a. prbpred xApp – Implementation Details
      This xApp is responsible to receive A1_POLICY_REQ, save the policy details. Fetch the model from modelStore and save it. Based on timer trigger predict the future PRB utilisation and respond to alloc xApp for further processing.
      i. Upon Initialization does following
         1. registers for PRB_PRED_REQ (PRB Prediction Request) and A1_POLICY_REQ (A1 Policy Request)
         2. queries A1-mediator and gets the Policy details

      ii. As part of step 8,9 in the sequence diagram mentioned in Reference [4]. A specific policy was created which gives info on model and model version info to be used.

iii. Based on the policy details xApp fetches the model from model store and stores it locally
iv. Upon reception of PRB_PRED_REQ, based on the model fetched performs prediction of PRB utilisation for each slice and sends response to Alloc xApp
v. Following are the message types handled by this xApp
Outgoing Message types: - **"A1_POLICY_RESP", "PRB_PRED_RSP","A1_POLICY_QUERY"**
Incoming Message Types - **"PRB_PRED_REQ", "A1_POLICY_REQ"**

xApp Descriptor Details

```
{
        "xapp_name": "prbpredxapp",
        "version": "0.0.2",
        "containers": [
            {
                "name": "prbpredxapp",
                "image": {
                    "registry": "nexus3.o-ran-sc.org:10002",
                    "name": "o-ran-sc/ric-app-prbpred",
                    "tag": "0.0.2"
                }
            }
        ],
        "messaging": {
            "ports": [
                 {
                "name": "http",
                "container": "prbpredxapp",
                "port": 10003,
                "description": "http service"
                },
                {
                    "name": "rmr-data",
                    "container": "prbpredxapp",
                    "port": 4560,
                    "rxMessages": [
                        "A1_POLICY_REQ",
                        "PRB_PRED_REQ"
                    ],
                    "txMessages": [ "A1_POLICY_RESP",
"PRB_PRED_RSP","A1_POLICY_QUERY" ],
                    "policies": [20008],
                    "description": "rmr receive data port "
                },
                {
                    "name": "rmr-route",
                    "container": "prbpredxapp",
                    "port": 4561,
                    "description": "rmr route port "
                }
            ]
        },
        "rmr": {
            "protPort": "tcp:4560",
            "maxSize": 2072,
            "numWorkers": 1,
            "txMessages": [
                "PRB_PRED_RSP",
                "A1_POLICY_RESP",
                "A1_POLICY_QUERY"
            ],
            "rxMessages": [
                "PRB_PRED_REQ",
                "A1_POLICY_REQ"
            ],
            "policies": [20008]
        }
    }
```

b. Allocator xApp – Implementation Details

i. Upon Initialization registers with subscription mgr. for E2 information And starts timer to trigger PRB_PRED_REQ periodically.

ii. Based on predicted future PRB utilisation computes PRB to be allocated for emergency slice

iii. A simple algorithm for PRB allocation (ALG1 from References [5] & [6]) is used here. In addition, some PRB's are shown as reserved for Emergency/High priority events.
Below are the details: -
Assumption taken is total no of PRB's in the system is 100
Slice #1 and Slice#2 have been configured with 35 PRB's each.
30 PRB's are reserved for Emergency/High priority events

Based on the predicted PRB utilisation received for each slice, the actual Value of PRB utilised is computed.
Utilised_PRB_slice1 = PRB_ALLOC_SLICE1*(slice1_utilisation/100)
Utilised_PRB_slice2= PRB_ALLOC_SLICE2*(slice2_utilisation/100)

total_prb_avail = Total_PRB – (Utilised_PRB_slice1 + Utilised_PRB_slice2)
As this is Emergency event, the reserved PRB's are also made available.

iv. Alloc xApp sends the E2 control message to allocate the available PRB's from the above calculation.

v. Following are the message types handled by this xApp
Outgoing Message types: - **"PRB_PRED_REQ","RIC_HEALTH_CHECK_RESP"**
Incoming Message Types - **"PRB_PRED_RESP", "SUBSCRIPTION_REQ","RIC_HEALTH_CHECK_REQ"**

xApp Descriptor Details

```
        "xapp_name": "alloc",
        "version": "0.0.2",
        "containers": [
            {
                "name": "alloc",
                "image": {
                    "registry": "nexus3.o-ran-sc.org:10002",
                    "name": "o-ran-sc/ric-app-alloc",
                    "tag": "0.0.2"
                }
            }
        ],
        "messaging": {
            "ports": [
                 {
                "name": "http",
                "container": "alloc",
                "port": 10005,
                "description": "http service"
                },
                {
                    "name": "rmr-data",
                    "container": "alloc",
                    "port": 4560,
                    "txMessages":
["PRB_PRED_REQ","RIC_HEALTH_CHECK_RESP"],
                    "rxMessages": ["PRB_PRED_RESP",
"SUBSCRIPTION_REQ","RIC_HEALTH_CHECK_REQ"],
                    "policies": [],
                    "description": "rmr receive data port for alloc"
                },
                {
                    "name": "rmr-route",
                    "container": "alloc",
                    "port": 4561,
                    "description": "rmr route port for alloc "
                }
            ]
        },
        "rmr": {
            "protPort": "tcp:4560",
            "maxSize": 2072,
            "numWorkers": 1,
            "rxMessages": ["PRB_PRED_RESP"],
            "txMessages": ["PRB_PRED_REQ"],
            "policies": []
        },
        "controls": {
            "fileStrorage": false
         },
        "db" : {
            "waitForSdl": false
        }
}
```

Successful communication between the XApp's and other RIC platform components was achieved as part of this.

3) A Model Store was developed to mimic Acumos and had access to the pre-trained model
4) E2 SIM setup was brought up and was successfully registered with E2 component in RIC platform

Note: In the Dawn release, creation of A1 policy instance doesn't trigger A1 Policy create message towards xApp. This was confirmed by ORAN-RIC team

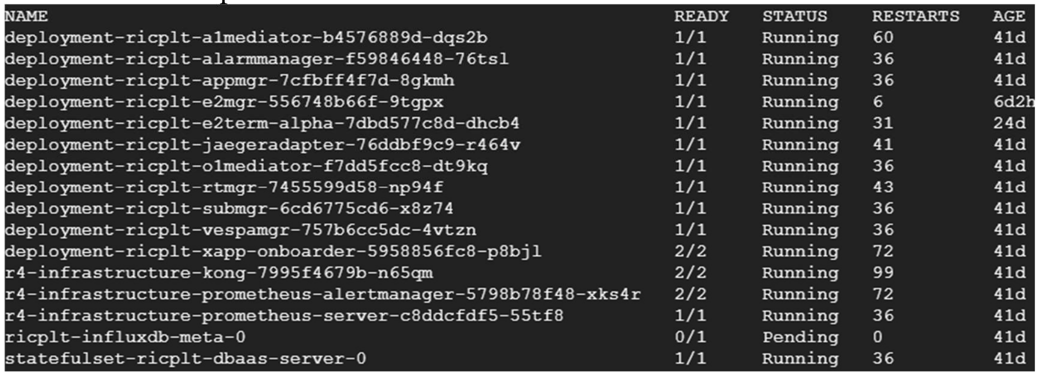
https://wiki.o-ran-sc.org/display/IAT/Traffic+Steering+Flows?focusedCommentId=41456537#comment-41456537

The workflow was modified to send timer-based event from alloc XApp to trigger PRB prediction. Once the A1 mediator sends CREATE/UPDATE messages to xApp when the policy instance is created, the prbpred xApp can store the model information and perform prediction based on the trigger.

Below is the sequence of message flows: -

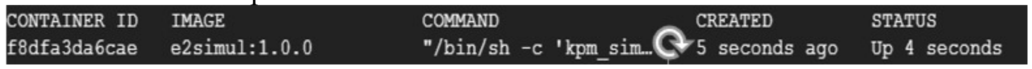
1) RIC Platform is brought up and then E2 SIM process is started. Upon start up, Simulator generates E2 Setup Request For each E2SM that is registered, includes RAN function definition. From the screenshot we can E2 Setup Request/Response. Points 1,2 in from the above fig are covered here

   a. RIC Platform Snapshot



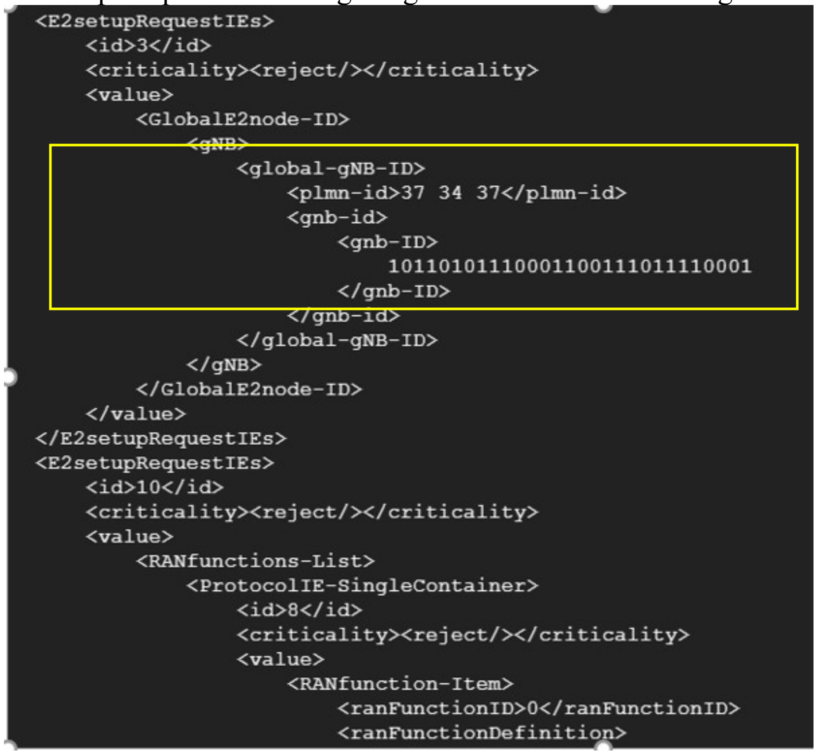
```
NAME                                                         READY   STATUS    RESTARTS   AGE
deployment-ricplt-a1mediator-b4576889d-dqs2b                 1/1     Running   60         41d
deployment-ricplt-alarmmanager-f59846448-76tsl               1/1     Running   36         41d
deployment-ricplt-appmgr-7cfbff4f7d-8gkmh                    1/1     Running   36         41d
deployment-ricplt-e2mgr-556748b66f-9tgpx                     1/1     Running   6          6d2h
deployment-ricplt-e2term-alpha-7dbd577c8d-dhcb4              1/1     Running   31         24d
deployment-ricplt-jaegeradapter-76ddbf9c9-r464v              1/1     Running   41         41d
deployment-ricplt-o1mediator-f7dd5fcc8-dt9kq                 1/1     Running   36         41d
deployment-ricplt-rtmgr-7455599d58-np94f                     1/1     Running   43         41d
deployment-ricplt-submgr-6cd6775cd6-x8z74                    1/1     Running   36         41d
deployment-ricplt-vespamgr-757b6cc5dc-4vtzn                  1/1     Running   36         41d
deployment-ricplt-xapp-onboarder-5958856fc8-p8bjl            2/2     Running   72         41d
r4-infrastructure-kong-7995f4679b-n65qm                      2/2     Running   99         41d
r4-infrastructure-prometheus-alertmanager-5798b78f48-xks4r   2/2     Running   72         41d
r4-infrastructure-prometheus-server-c8ddcfdf5-55tf8          1/1     Running   36         41d
ricplt-influxdb-meta-0                                       0/1     Pending   0          41d
statefulset-ricplt-dbaas-server-0                            1/1     Running   36         41d
```

   b. E2 SIM Process Snapshot

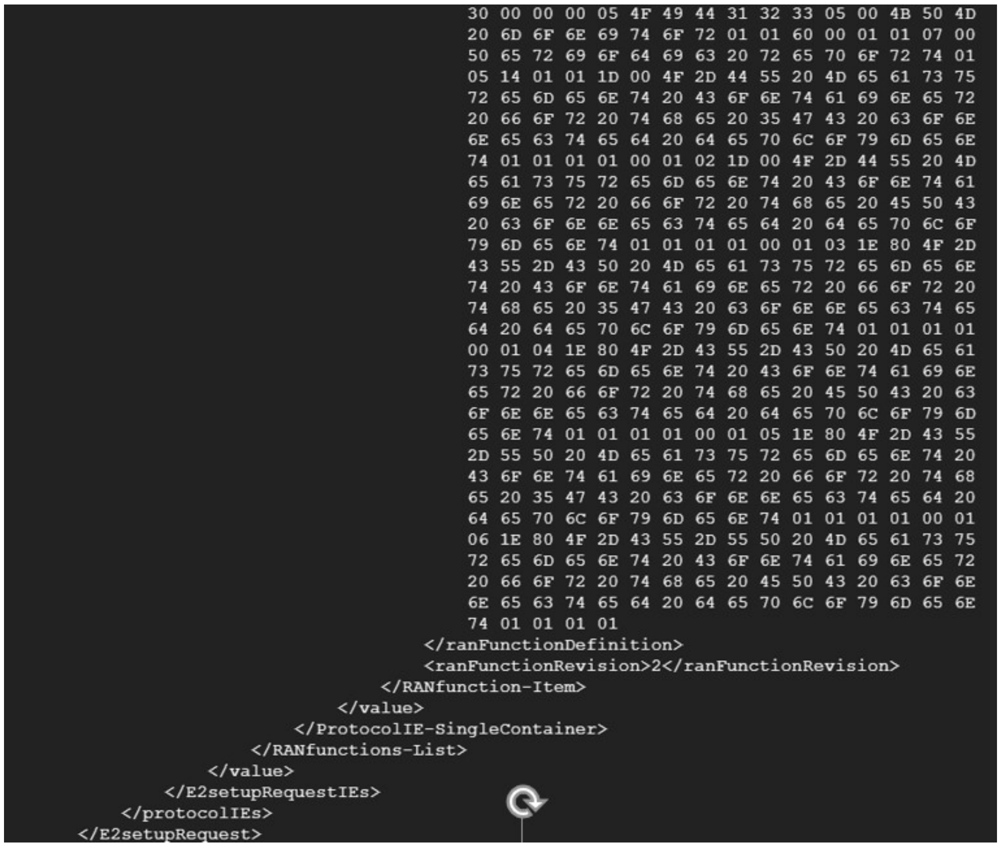


```
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS
f8dfa3da6cae   e2simul:1.0.0  "/bin/sh -c 'kpm_sim…   5 seconds ago   Up 4 seconds
```

   c. E2 Setup Request containing the gnb-Id and PLMN-id being sent from E2 interface

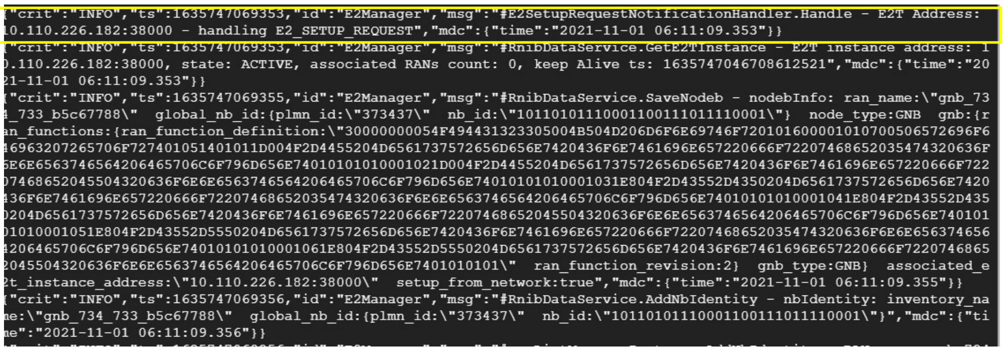


```
<E2setupRequestIEs>
    <id>3</id>
    <criticality><reject/></criticality>
    <value>
        <GlobalE2node-ID>
            <gNB>
                <global-gNB-ID>
                    <plmn-id>37 34 37</plmn-id>
                    <gnb-id>
                        <gnb-ID>
                            10110101110001100111011110001
                        </gnb-ID>
                    </gnb-id>
                </global-gNB-ID>
            </gNB>
        </GlobalE2node-ID>
    </value>
</E2setupRequestIEs>
<E2setupRequestIEs>
    <id>10</id>
    <criticality><reject/></criticality>
    <value>
        <RANfunctions-List>
            <ProtocolIE-SingleContainer>
                <id>8</id>
                <criticality><reject/></criticality>
                <value>
                    <RANfunction-Item>
                        <ranFunctionID>0</ranFunctionID>
                        <ranFunctionDefinition>
```



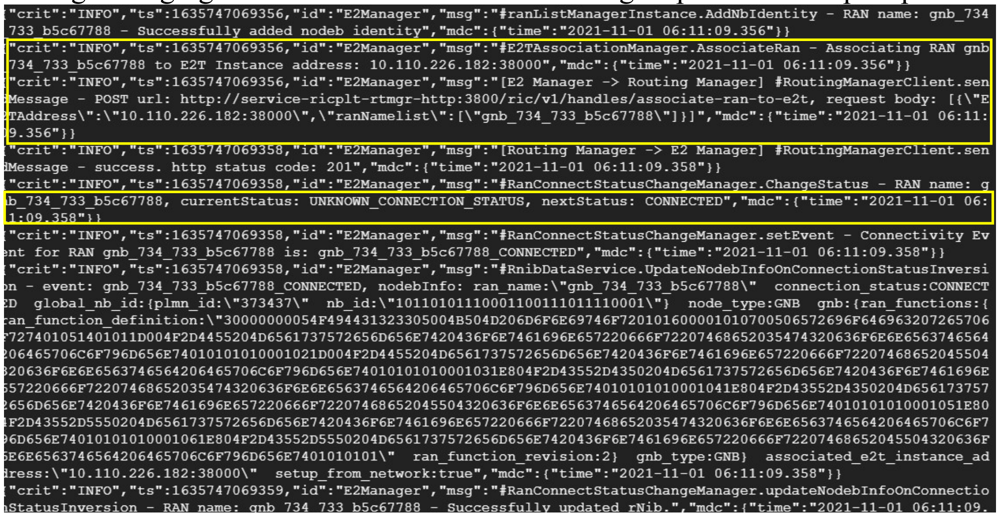
```
                    30 00 00 00 05 4F 49 44 31 32 33 05 00 4B 50 4D
                    20 6D 6F 6E 69 74 6F 72 01 01 60 00 01 01 07 00
                    50 65 72 69 6F 64 69 63 20 72 65 70 6F 72 74 01
                    05 14 01 01 1D 00 4F 2D 44 55 20 4D 65 61 73 75
                    72 65 6D 65 6E 74 20 43 6F 6E 74 61 69 6E 65 72
                    20 66 6F 72 20 74 68 65 20 35 47 43 20 63 6F 6E
                    6E 65 63 74 65 64 20 64 65 70 6C 6F 79 6D 65 6E
                    74 01 01 01 01 00 01 02 1D 00 4F 2D 44 55 20 4D
                    65 61 73 75 72 65 6D 65 6E 74 20 43 6F 6E 74 61
                    69 6E 65 72 20 66 6F 72 20 74 68 65 20 45 50 43
                    20 63 6F 6E 6E 65 63 74 65 64 20 64 65 70 6C 6F
                    79 6D 65 6E 74 01 01 01 01 00 01 03 1E 80 4F 2D
                    43 55 2D 43 50 20 4D 65 61 73 75 72 65 6D 65 6E
                    74 20 43 6F 6E 74 61 69 6E 65 72 20 66 6F 72 20
                    74 68 65 20 35 47 43 20 63 6F 6E 6E 65 63 74 65
                    64 20 64 65 70 6C 6F 79 6D 65 6E 74 01 01 01 01
                    00 01 04 1E 80 4F 2D 43 55 2D 43 50 20 4D 65 61
                    73 75 72 65 6D 65 6E 74 20 43 6F 6E 74 61 69 6E
                    65 72 20 66 6F 72 20 74 68 65 20 45 50 43 20 63
                    6F 6E 6E 65 63 74 65 64 20 64 65 70 6C 6F 79 6D
                    65 6E 74 01 01 01 01 00 01 05 1E 80 4F 2D 43 55
                    2D 55 50 20 4D 65 61 73 75 72 65 6D 65 6E 74 20
                    43 6F 6E 74 61 69 6E 65 72 20 66 6F 72 20 74 68
                    65 20 35 47 43 20 63 6F 6E 6E 65 63 74 65 64 20
                    64 65 70 6C 6F 79 6D 65 6E 74 01 01 01 01 00 01
                    06 1E 80 4F 2D 43 55 2D 55 50 20 4D 65 61 73 75
                    72 65 6D 65 6E 74 20 43 6F 6E 74 61 69 6E 65 72
                    20 66 6F 72 20 74 68 65 20 45 50 43 20 63 6F 6E
                    6E 65 63 74 65 64 20 64 65 70 6C 6F 79 6D 65 6E
                    74 01 01 01 01
                  </ranFunctionDefinition>
                  <ranFunctionRevision>2</ranFunctionRevision>
                </RANfunction-Item>
              </value>
            </ProtocolIE-SingleContainer>
          </RANfunctions-List>
        </value>
      </E2setupRequestIEs>
    </protocolIEs>
  </E2setupRequest>
```

d. E2 Mgr. receiving E2 Setup Request

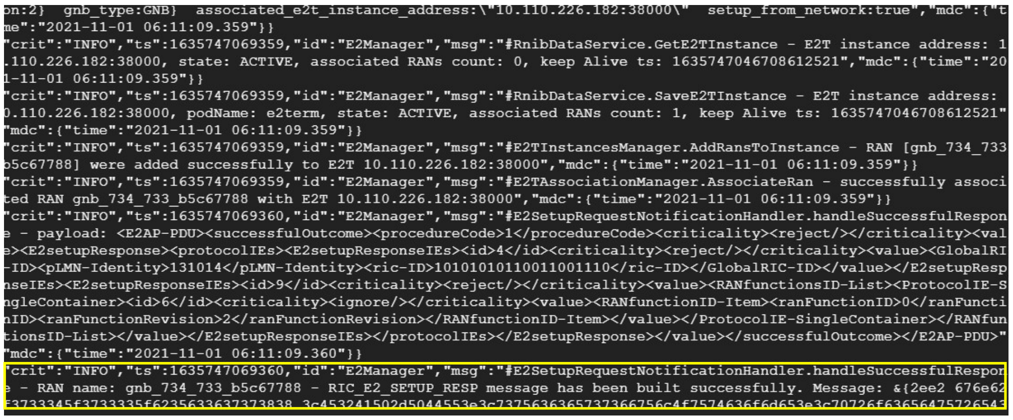


```
{"crit":"INFO","ts":1635747069353,"id":"E2Manager","msg":"#E2SetupRequestNotificationHandler.Handle - E2T Address:
10.110.226.182:38000 - handling E2_SETUP_REQUEST","mdc":{"time":"2021-11-01 06:11:09.353"}}
{"crit":"INFO","ts":1635747069353,"id":"E2Manager","msg":"#RnibDataService.GetE2TInstance - E2T instance address: 1
0.110.226.182:38000, state: ACTIVE, associated RANs count: 0, keep Alive ts: 1635747046708612521","mdc":{"time":"20
21-11-01 06:11:09.353"}}
{"crit":"INFO","ts":1635747069355,"id":"E2Manager","msg":"#RnibDataService.SaveNodeb - nodebInfo: ran_name:\"gnb_73
4_733_b5c67788\"  global_nb_id:{plmn_id:\"373437\"  nb_id:\"10110101110001100111011110001\"}  node_type:GNB  gnb:{r
an_functions:{ran_function_definition:\"30000000054F4944313233305004B504D206D6F6E69746F7201016000010107005065726 96F6
46963207265706F727401051401011D004F2D4455204D6561737572656D656E7420436F6E7461696E657220666F7220746865203547432063 6F
6E6E656374656420646570 6C6F796D656E74010101010001021D004F2D4455204D6561737572656D656E7420436F6E7461696E657220666F722
074686520455043 20636F6E6E656374656420646570 6C6F796D656E740101010100010 31E804F2D43552D4350204D6561737572656D656E7420
436F6E7461696E657220666F72207468652035474320636F6E6E656374656420646570 6C6F796D656E7401010101000104 1E804F2D43552D435
0204D6561737572656D656E7420436F6E7461696E657220666F7220746865204550432063 6F6E6E656374656420646570 6C6F796D656E740101
01010001051E804F2D43552D5550204D6561737572656D656E7420436F6E7461696E657220666F722074686520354743 20636F6E6E656374656
4206465706C6F796D656E7401010101000106 1E804F2D43552D5550204D6561737572656D656E7420436F6E7461696E657220666F72207468 65
2045504320636F6E6E656374656420646570 6C6F796D656E7401010101\"  ran_function_revision:2}  gnb_type:GNB}  associated_e
2t_instance_address:\"10.110.226.182:38000\"  setup_from_network:true","mdc":{"time":"2021-11-01 06:11:09.355"}}
{"crit":"INFO","ts":1635747069356,"id":"E2Manager","msg":"#RnibDataService.AddNbIdentity - nbIdentity: inventory_na
me:\"gnb_734_733_b5c67788\"  global_nb_id:{plmn_id:\"373437\"  nb_id:\"10110101110001100111011110001\"}","mdc":{"ti
me":"2021-11-01 06:11:09.356"}}
```

e. E2 Mgr changing the state as connected and sending response E2 setup response

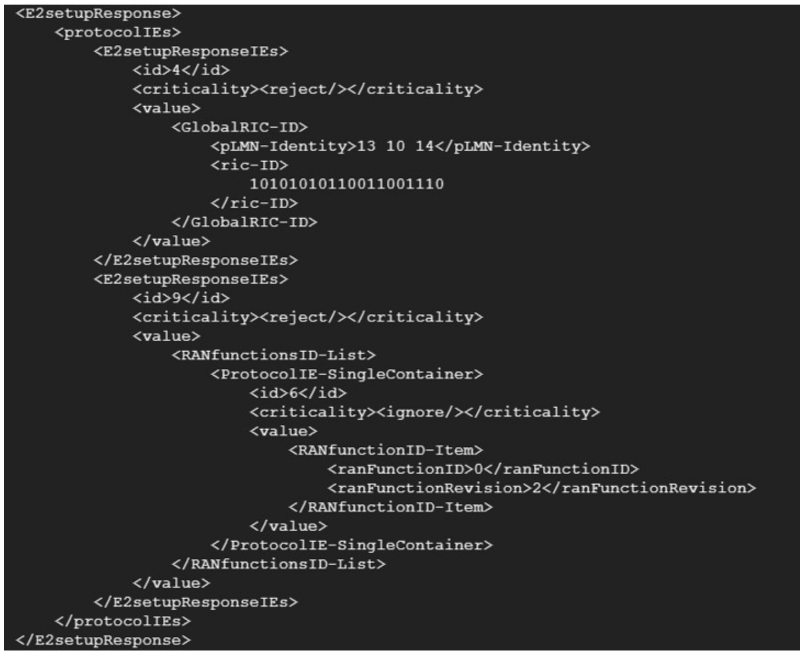


```
{"crit":"INFO","ts":1635747069356,"id":"E2Manager","msg":"#ranListManagerInstance.AddNbIdentity - RAN name: gnb_734
_733_b5c67788 - Successfully added nodeb identity","mdc":{"time":"2021-11-01 06:11:09.356"}}
{"crit":"INFO","ts":1635747069356,"id":"E2Manager","msg":"#E2TAssociationManager.AssociateRan - Associating RAN gnb
_734_733_b5c67788 to E2T Instance address: 10.110.226.182:38000","mdc":{"time":"2021-11-01 06:11:09.356"}}
{"crit":"INFO","ts":1635747069356,"id":"E2Manager","msg":"[E2 Manager -> Routing Manager] #RoutingManagerClient.sen
dMessage - POST url: http://service-ricplt-rtmgr-http:3800/ric/v1/handles/associate-ran-to-e2t, request body: [{\"E
2TAddress\":\"10.110.226.182:38000\",\"ranNamelist\":[\"gnb_734_733_b5c67788\"]}]","mdc":{"time":"2021-11-01 06:11:
09.356"}}
{"crit":"INFO","ts":1635747069358,"id":"E2Manager","msg":"[Routing Manager -> E2 Manager] #RoutingManagerClient.sen
dMessage - success. http status code: 201","mdc":{"time":"2021-11-01 06:11:09.358"}}
{"crit":"INFO","ts":1635747069358,"id":"E2Manager","msg":"#RanConnectStatusChangeManager.ChangeStatus - RAN name: g
nb_734_733_b5c67788, currentStatus: UNKNOWN_CONNECTION_STATUS, nextStatus: CONNECTED","mdc":{"time":"2021-11-01 06:
11:09.358"}}
{"crit":"INFO","ts":1635747069358,"id":"E2Manager","msg":"#RanConnectStatusChangeManager.setEvent - Connectivity Ev
ent for RAN gnb_734_733_b5c67788 is: gnb_734_733_b5c67788_CONNECTED","mdc":{"time":"2021-11-01 06:11:09.358"}}
{"crit":"INFO","ts":1635747069358,"id":"E2Manager","msg":"#RnibDataService.UpdateNodebInfoOnConnectionStatusInversi
on - event: gnb_734_733_b5c67788_CONNECTED, nodebInfo: ran_name:\"gnb_734_733_b5c67788\"  connection_status:CONNECT
ED  global_nb_id:{plmn_id:\"373437\"  nb_id:\"10110101110001100111011110001\"}  node_type:GNB  gnb:{ran_functions:{
ran_function_definition:\"30000000054F4944313233305004B504D206D6F6E69746F72010160000101070050657269 6F646963207265706
F727401051401011D004F2D4455204D6561737572656D656E7420436F6E7461696E657220666F72207468652035474320636F6E6E6563746564
206465706C6F796D656E74010101010001021D004F2D4455204D6561737572656D656E7420436F6E7461696E657220666F72207468652045504
320636F6E6E656374656420646570 6C6F796D656E740101010100010 31E804F2D43552D4350204D6561737572656D656E7420436F6E7461696E
657220666F72207468652035474320636F6E6E656374656420646570 6C6F796D656E7401010101000104 1E804F2D43552D4350204D656173757
2656D656E7420436F6E7461696E657220666F722074686520455043 20636F6E6E656374656420646570 6C6F796D656E7401010101000105 1E80
4F2D43552D5550204D6561737572656D656E7420436F6E7461696E657220666F72207468652035474320636F6E6E656374656420646570 6C6F7
96D656E7401010101000106 1E804F2D43552D5550204D6561737572656D656E7420436F6E7461696E657220666F722074686520455043 20636F
6E6E656374656420646570 6C6F796D656E7401010101\"  ran_function_revision:2}  gnb_type:GNB}  associated_e2t_instance_ad
dress:\"10.110.226.182:38000\"  setup_from_network:true","mdc":{"time":"2021-11-01 06:11:09.358"}}
{"crit":"INFO","ts":1635747069359,"id":"E2Manager","msg":"#RanConnectStatusChangeManager.updateNodebInfoOnConnectio
nStatusInversion - RAN name: gnb_734_733_b5c67788 - Successfully updated rNib.","mdc":{"time":"2021-11-01 06:11:09.
```



```
on:2}  gnb_type:GNB}  associated_e2t_instance_address:\"10.110.226.182:38000\"  setup_from_network:true","mdc":{"t
me":"2021-11-01 06:11:09.359"}}
"crit":"INFO","ts":1635747069359,"id":"E2Manager","msg":"#RnibDataService.GetE2TInstance - E2T instance address: 1
.110.226.182:38000, state: ACTIVE, associated RANs count: 0, keep Alive ts: 1635747046708612521","mdc":{"time":"20
1-11-01 06:11:09.359"}}
"crit":"INFO","ts":1635747069359,"id":"E2Manager","msg":"#RnibDataService.SaveE2TInstance - E2T instance address:
0.110.226.182:38000, podName: e2term, state: ACTIVE, associated RANs count: 1, keep Alive ts: 1635747046708612521"
"mdc":{"time":"2021-11-01 06:11:09.359"}}
"crit":"INFO","ts":1635747069359,"id":"E2Manager","msg":"#E2TInstancesManager.AddRansToInstance - RAN [gnb_734_733
b5c67788] were added successfully to E2T 10.110.226.182:38000","mdc":{"time":"2021-11-01 06:11:09.359"}}
"crit":"INFO","ts":1635747069359,"id":"E2Manager","msg":"#E2TAssociationManager.AssociateRan - successfully associ
ted RAN gnb_734_733_b5c67788 with E2T 10.110.226.182:38000","mdc":{"time":"2021-11-01 06:11:09.359"}}
"crit":"INFO","ts":1635747069360,"id":"E2Manager","msg":"#E2SetupRequestNotificationHandler.handleSuccessfulRespon
e - payload: <E2AP-PDU><successfulOutcome><procedureCode>1</procedureCode><criticality><reject/></criticality><val
e><E2setupResponse><protocolIEs><E2setupResponseIEs><id>4</id><criticality><reject/></criticality><value><GlobalRI
-ID><pLMN-Identity>131014</pLMN-Identity><ric-ID>10101010110011001110</ric-ID></GlobalRIC-ID></value></E2setupResp
nseIEs><E2setupResponseIEs><id>9</id><criticality><reject/></criticality><value><RANfunctionsID-List><ProtocolIE-S
ngleContainer><id>6</id><criticality><ignore/></criticality><value><RANfunctionID-Item><ranFunctionID>0</ranFuncti
nID><ranFunctionRevision>2</ranFunctionRevision></RANfunctionID-Item></value></ProtocolIE-SingleContainer></RANfun
tionsID-List></value></E2setupResponseIEs></protocolIEs></E2setupResponse></value></successfulOutcome></E2AP-PDU>"
"mdc":{"time":"2021-11-01 06:11:09.360"}}
"crit":"INFO","ts":1635747069360,"id":"E2Manager","msg":"#E2SetupRequestNotificationHandler.handleSuccessfulRespon
e - RAN name: gnb_734_733_b5c67788 - RIC_E2_SETUP_RESP message has been built successfully. Message: &{2ee2 676e62
f3733345f3733335f6235633637373838 3c453241502d5044553e3c7375636365737366756c4f7574636f6d653e3c70726f63656475726543
```

f. E2 Setup Response at E2 SIM end

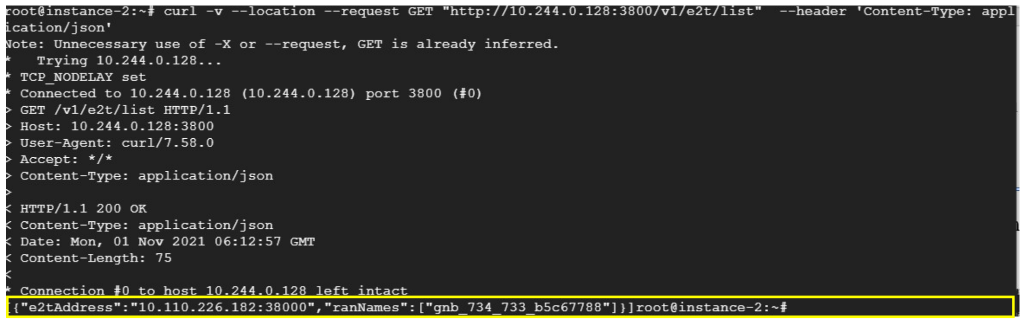


```
<E2setupResponse>
    <protocolIEs>
        <E2setupResponseIEs>
            <id>4</id>
            <criticality><reject/></criticality>
            <value>
                <GlobalRIC-ID>
                    <pLMN-Identity>13 10 14</pLMN-Identity>
                    <ric-ID>
                        10101010110011001110
                    </ric-ID>
                </GlobalRIC-ID>
            </value>
        </E2setupResponseIEs>
        <E2setupResponseIEs>
            <id>9</id>
            <criticality><reject/></criticality>
            <value>
                <RANfunctionsID-List>
                    <ProtocolIE-SingleContainer>
                        <id>6</id>
                        <criticality><ignore/></criticality>
                        <value>
                            <RANfunctionID-Item>
                                <ranFunctionID>0</ranFunctionID>
                                <ranFunctionRevision>2</ranFunctionRevision>
                            </RANfunctionID-Item>
                        </value>
                    </ProtocolIE-SingleContainer>
                </RANfunctionsID-List>
            </value>
        </E2setupResponseIEs>
    </protocolIEs>
</E2setupResponse>
```

g. E2 RAN registration can confirmed with response from below CURL command.

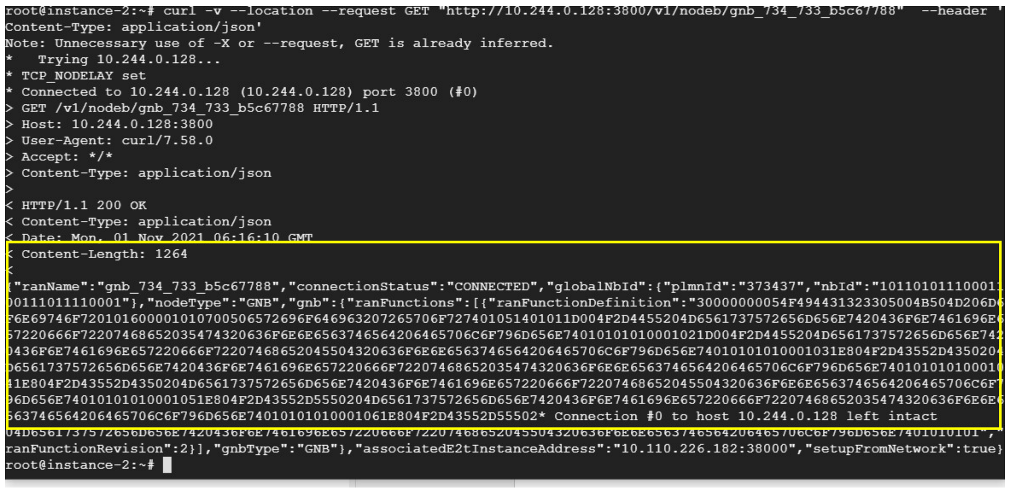
curl -v --location --request GET "http://10.244.0.128:3800/v1/e2t/list" --header 'Content-Type: application/json'

Result displays the e2Term instance and RAN name associated



```
root@instance-2:~# curl -v --location --request GET "http://10.244.0.128:3800/v1/e2t/list"  --header 'Content-Type: appl
ication/json'
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 10.244.0.128...
* TCP_NODELAY set
* Connected to 10.244.0.128 (10.244.0.128) port 3800 (#0)
> GET /v1/e2t/list HTTP/1.1
> Host: 10.244.0.128:3800
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Type: application/json
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Date: Mon, 01 Nov 2021 06:12:57 GMT
< Content-Length: 75
<
* Connection #0 to host 10.244.0.128 left intact
[{"e2tAddress":"10.110.226.182:38000","ranNames":["gnb_734_733_b5c67788"]}]root@instance-2:~#
```

h. Fetch the gNB details associated on trigger of GET request from E2mgr curl -v --location --request GET "http://10.244.0.128:3800/v1/nodeb/gnb_734_733_b5c67788" --header 'Content-Type: application/json'

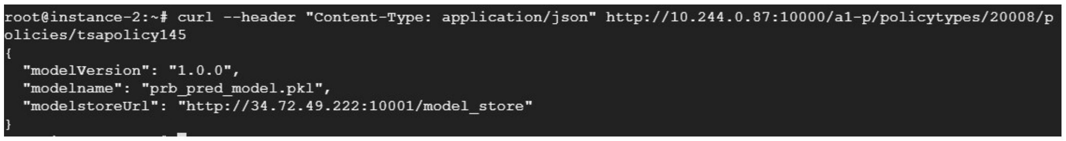


```
root@instance-2:~# curl -v --location --request GET "http://10.244.0.128:3800/v1/nodeb/gnb_734_733_b5c67788"  --header '
Content-Type: application/json'
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 10.244.0.128...
* TCP_NODELAY set
* Connected to 10.244.0.128 (10.244.0.128) port 3800 (#0)
> GET /v1/nodeb/gnb_734_733_b5c67788 HTTP/1.1
> Host: 10.244.0.128:3800
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Type: application/json
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Date: Mon, 01 Nov 2021 06:16:10 GMT
< Content-Length: 1264
<
{"ranName":"gnb_734_733_b5c67788","connectionStatus":"CONNECTED","globalNbId":{"plmnId":"373437","nbId":"10110101110001
001110111110001"},"nodeType":"GNB","gnb":{"ranFunctions":[{"ranFunctionDefinition":"30000000054F494431323305004B504D206D
F6E69746F72010160000101070050657269F646963207265706F727401051401011D004F2D4455204D6561737572656D656E7420436F6E7461696E
57220666F72207468652035474320636F6E6E656374656420646576C6F796D656E74010101010001021D004F2D4455204D6561737572656D656E742
0436F6E7461696E657220666F72207468652045504320636F6E6E656374656420646576C6F796D656E74010101010001031E804F2D43552D435020
D6561737572656D656E7420436F6E7461696E657220666F72207468652035474320636F6E6E656374656420646576C6F796D656E740101010100010
41E804F2D43552D4350204D6561737572656D656E7420436F6E7461696E657220666F72207468652045504320636F6E6E656374656420646576C6F7
96D656E74010101010001051E804F2D43552D5550204D6561737572656D656E7420436F6E7461696E657220666F72207468652035474320636F6E6E6
56374656420646576C6F796D656E74010101010001061E804F2D43552D55502* Connection #0 to host 10.244.0.128 left intact
04D6561737572656D656E7420436F6E7461696E657220666F72207468652045504320636F6E6E656374656420646576C6F796D656E7401010101","
ranFunctionRevision":2}],"gnbType":"GNB"},"associatedE2tInstanceAddress":"10.110.226.182:38000","setupFromNetwork":true}
root@instance-2:~#
```

2) A1- Policy and Policy Instances are created by executing following curl commands
   - curl -X PUT --header "Content-Type: application/json" -d @create1.json http://<a1med_ip>: <a1med_port>:/a1-p/policytypes/20008

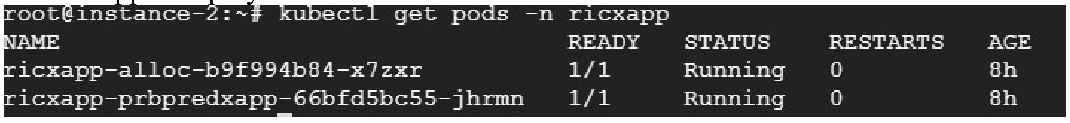
   - curl -X PUT --header "Content-Type: application/json" --data '{"modelVersion" : "1.0.0", "modelname":" prb_pred_model.pkl", "modelstoreUrl": "http:// <modelstoreIp>:10001/model_store"}' http://<a1med_ip>: <a1med_port>/a1-p/policytypes/20008/policies/tsapolicy145

   Below snapshot shows the policy instance is created with the values supplied in the curl command



```
root@instance-2:~# curl --header "Content-Type: application/json" http://10.244.0.87:10000/a1-p/policytypes/20008/p
olicies/tsapolicy145
{
  "modelVersion": "1.0.0",
  "modelname": "prb_pred_model.pkl",
  "modelstoreUrl": "http://34.72.49.222:10001/model_store"
}
```

3) Alloc xApp is deployed



```
root@instance-2:~# kubectl get pods -n ricxapp
NAME                                  READY   STATUS    RESTARTS   AGE
ricxapp-alloc-b9f994b84-x7zxr         1/1     Running   0          8h
ricxapp-prbpredxapp-66bfd5bc55-jhrmn  1/1     Running   0          8h
```

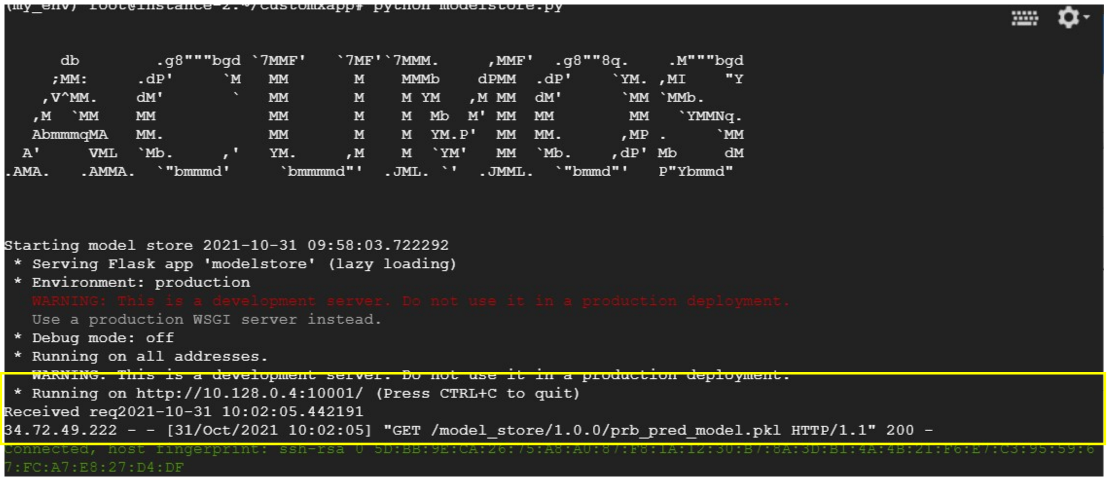
   Alloc xApp sends REST Subscription request to Subscription Manager

4) prbpred xApp is also deployed

   Above snapshot shows both xApps being deployed

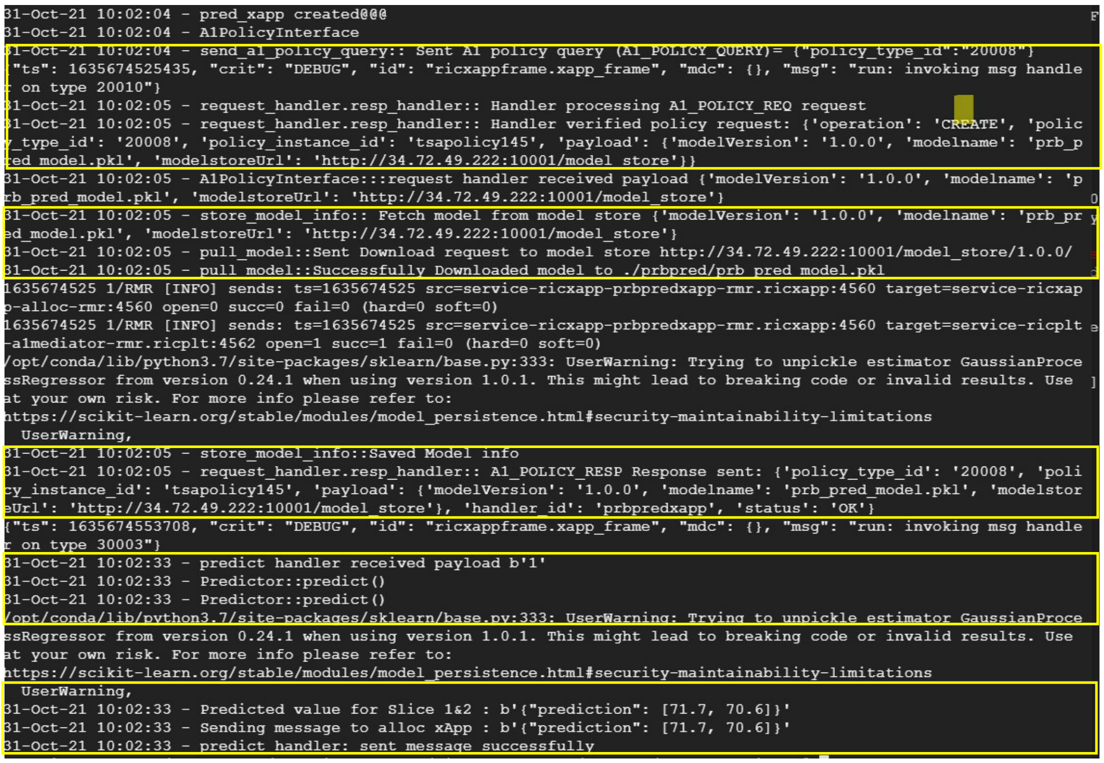
   Prbpred xApp sends query to A1 mediator on the policy details. It receives A1 POLICY REQ from A1 mediator

   a. Emulated Acumos Model Store listening on 10001 port, receives REST REQ for fetching model. Acumos functionality was emulated considering HW requirements in bringing up actual Acumos.



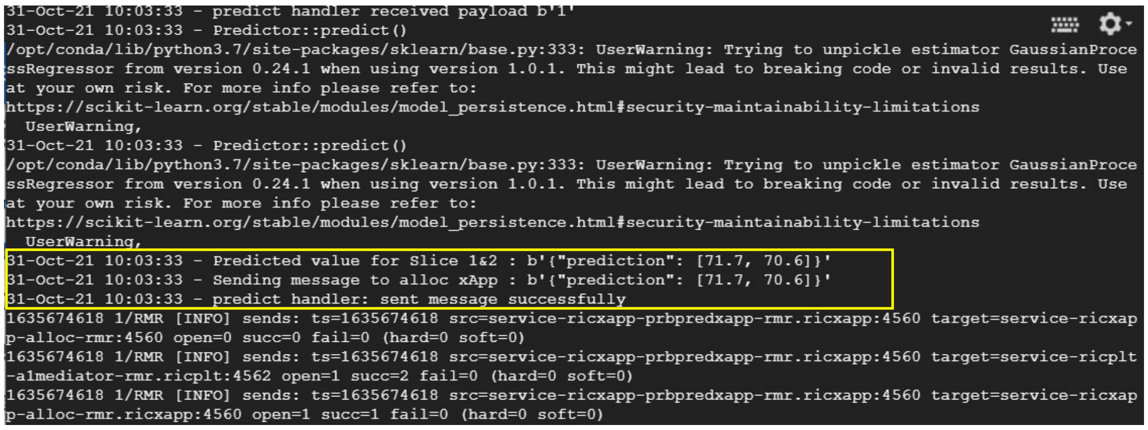
```
(my_env) root@instance-2:~/customxapp# python modelstore.py


     db        .g8"""bgd `7MMF'   `7MF'`7MMM.     ,MMF' .g8""8q.    .M"""bgd
    ;MM:     .dP'     `M   MM       M    MMMb    dPMM .dP'    `YM. ,MI    "Y
   ,V^MM.    dM'       `   MM       M    M YM   ,M MM dM'      `MM `MMb.
  ,M  `MM    MM            MM       M    M  Mb  M' MM MM        MM   `YMMNq.
  AbmmmqMA   MM.           MM       M    M  YM.P'  MM MM.      ,MP .     `MM
 A'     VML  `Mb.     ,'   YM.     ,M    M  `YM'   MM `Mb.    ,dP' Mb     dM
.AMA.   .AMMA. `"bmmmd'     `bmmmmd"'  .JML. `'  .JMML. `"bmmd"'   P"Ybmmd"


Starting model store 2021-10-31 09:58:03.722292
 * Serving Flask app 'modelstore' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://10.128.0.4:10001/ (Press CTRL+C to quit)
Received req2021-10-31 10:02:05.442191
34.72.49.222 - - [31/Oct/2021 10:02:05] "GET /model_store/1.0.0/prb_pred_model.pkl HTTP/1.1" 200 -
Connected, host fingerprint: ssh-rsa 0 5D:BB:9E:CA:26:75:A8:A0:87:F8:1A:12:30:B7:8A:3D:B1:4A:4B:21:F6:E7:C3:95:59:6
7:FC:A7:E8:27:D4:DF
```

b. Below Snapshot shows REST Req being sent to model store with the details received from A1 policy request. Model is successfully downloaded to prbpred container.



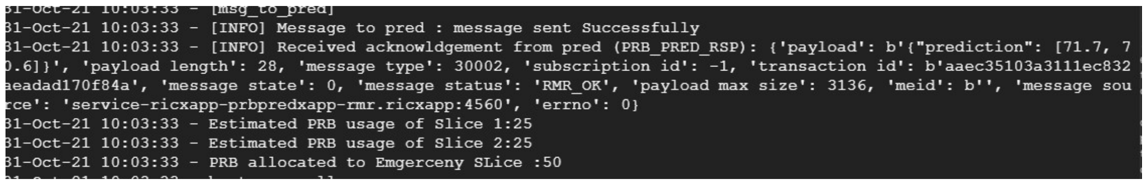
```
31-Oct-21 10:02:04 - pred_xapp created@@@
31-Oct-21 10:02:04 - A1PolicyInterface
31-Oct-21 10:02:04 - send_a1_policy_query:: Sent A1 policy query (A1_POLICY_QUERY)= {"policy_type_id":"20008"}
{"ts": 1635674525435, "crit": "DEBUG", "id": "ricxappframe.xapp_frame", "mdc": {}, "msg": "run: invoking msg handle
r on type 20010"}
31-Oct-21 10:02:05 - request_handler.resp_handler:: Handler processing A1_POLICY_REQ request
31-Oct-21 10:02:05 - request_handler.resp_handler:: Handler verified policy request: {'operation': 'CREATE', 'polic
y_type_id': '20008', 'policy_instance_id': 'tsapolicy145', 'payload': {'modelVersion': '1.0.0', 'modelname': 'prb_p
red_model.pkl', 'modelstoreUrl': 'http://34.72.49.222:10001/model_store'}}
31-Oct-21 10:02:05 - A1PolicyInterface:::request handler received payload {'modelVersion': '1.0.0', 'modelname': 'p
rb_pred_model.pkl', 'modelstoreUrl': 'http://34.72.49.222:10001/model_store'}
31-Oct-21 10:02:05 - store_model_info:: Fetch model from model store {'modelVersion': '1.0.0', 'modelname': 'prb_pr
ed_model.pkl', 'modelstoreUrl': 'http://34.72.49.222:10001/model_store'}
31-Oct-21 10:02:05 - pull_model::Sent Download request to model store http://34.72.49.222:10001/model_store/1.0.0/
31-Oct-21 10:02:05 - pull_model::Successfully Downloaded model to ./prbpred/prb_pred_model.pkl
1635674525 1/RMR [INFO] sends: ts=1635674525 src=service-ricxapp-prbpredxapp-rmr.ricxapp:4560 target=service-ricxap
p-alloc-rmr:4560 open=0 succ=0 fail=0 (hard=0 soft=0)
1635674525 1/RMR [INFO] sends: ts=1635674525 src=service-ricxapp-prbpredxapp-rmr.ricxapp:4560 target=service-ricplt
-a1mediator-rmr.ricplt:4562 open=1 succ=1 fail=0 (hard=0 soft=0)
/opt/conda/lib/python3.7/site-packages/sklearn/base.py:333: UserWarning: Trying to unpickle estimator GaussianProce
ssRegressor from version 0.24.1 when using version 1.0.1. This might lead to breaking code or invalid results. Use
at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
31-Oct-21 10:02:05 - store_model_info::Saved Model info
31-Oct-21 10:02:05 - request_handler.resp_handler:: A1_POLICY_RESP Response sent: {'policy_type_id': '20008', 'poli
cy_instance_id': 'tsapolicy145', 'payload': {'modelVersion': '1.0.0', 'modelname': 'prb_pred_model.pkl', 'modelstor
eUrl': 'http://34.72.49.222:10001/model_store'}, 'handler_id': 'prbpredxapp', 'status': 'OK'}
{"ts": 1635674553708, "crit": "DEBUG", "id": "ricxappframe.xapp_frame", "mdc": {}, "msg": "run: invoking msg handle
r on type 30003"}
31-Oct-21 10:02:33 - predict handler received payload b'1'
31-Oct-21 10:02:33 - Predictor::predict()
31-Oct-21 10:02:33 - Predictor::predict()
/opt/conda/lib/python3.7/site-packages/sklearn/base.py:333: UserWarning: Trying to unpickle estimator GaussianProce
ssRegressor from version 0.24.1 when using version 1.0.1. This might lead to breaking code or invalid results. Use
at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
31-Oct-21 10:02:33 - Predicted value for Slice 1&2 : b'{"prediction": [71.7, 70.6]}'
31-Oct-21 10:02:33 - Sending message to alloc xApp : b'{"prediction": [71.7, 70.6]}'
31-Oct-21 10:02:33 - predict handler: sent message successfully
```

5) On Timer expiry Alloc xApp sent PRB_PRED_REQ to prbpred xApp and Prbpred xApp receives PRB_PRED_REQ, performs prediction of the PRB's and sends message to Alloc XAPP



```
31-Oct-21 10:03:33 - predict handler received payload b'1'
31-Oct-21 10:03:33 - Predictor::predict()
/opt/conda/lib/python3.7/site-packages/sklearn/base.py:333: UserWarning: Trying to unpickle estimator GaussianProce
ssRegressor from version 0.24.1 when using version 1.0.1. This might lead to breaking code or invalid results. Use
at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
31-Oct-21 10:03:33 - Predictor::predict()
/opt/conda/lib/python3.7/site-packages/sklearn/base.py:333: UserWarning: Trying to unpickle estimator GaussianProce
ssRegressor from version 0.24.1 when using version 1.0.1. This might lead to breaking code or invalid results. Use
at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
31-Oct-21 10:03:33 - Predicted value for Slice 1&2 : b'{"prediction": [71.7, 70.6]}'
31-Oct-21 10:03:33 - Sending message to alloc xApp : b'{"prediction": [71.7, 70.6]}'
31-Oct-21 10:03:33 - predict handler: sent message successfully
1635674618 1/RMR [INFO] sends: ts=1635674618 src=service-ricxapp-prbpredxapp-rmr.ricxapp:4560 target=service-ricxap
p-alloc-rmr:4560 open=0 succ=0 fail=0 (hard=0 soft=0)
1635674618 1/RMR [INFO] sends: ts=1635674618 src=service-ricxapp-prbpredxapp-rmr.ricxapp:4560 target=service-ricplt
-a1mediator-rmr.ricplt:4562 open=1 succ=2 fail=0 (hard=0 soft=0)
1635674618 1/RMR [INFO] sends: ts=1635674618 src=service-ricxapp-prbpredxapp-rmr.ricxapp:4560 target=service-ricxap
p-alloc-rmr.ricxapp:4560 open=1 succ=1 fail=0 (hard=0 soft=0)
```

6) Alloc xApp receives PRB_PRED_RSP from prbpred xApp, computes the PRB to be allocated and sends control message to E2



```
31-Oct-21 10:03:33 - [msg_to_pred]
31-Oct-21 10:03:33 - [INFO] Message to pred : message sent Successfully
31-Oct-21 10:03:33 - [INFO] Received acknowldgement from pred (PRB_PRED_RSP): {'payload': b'{"prediction": [71.7, 7
0.6]}', 'payload length': 28, 'message type': 30002, 'subscription id': -1, 'transaction id': b'aaec35103a3111ec832
aeadad170f84a', 'message state': 0, 'message status': 'RMR_OK', 'payload max size': 3136, 'meid': b'', 'message sou
rce': 'service-ricxapp-prbpredxapp-rmr.ricxapp:4560', 'errno': 0}
31-Oct-21 10:03:33 - Estimated PRB usage of Slice 1:25
31-Oct-21 10:03:33 - Estimated PRB usage of Slice 2:25
31-Oct-21 10:03:33 - PRB allocated to Emgerceny SLice :50
```

4. Code Repository

[Github] https://github.com/deenammd/ITUML5GChallenge_BuildaThon

5. Open issues

a. ricplt-influxdb-meta-0 pod is in pending state in RIC platform. Tried all the suggestion as mentioned in RIC wiki but couldn't succeed
b. Not receiving Subscription response from Subscription Manager
c. Behaviour of A1 mediator sending A1 POLICY REQ to xApps when policy instance is CREATED/UPDATED is suppressed in Dawn release

6. Future activities

a. Build a multivariant timeseries model with monitored data and arrive at proper inference.
b. It is recommended that gNode/E2 interface has reserved resources for Emergency situations. Additionally, based on the situation resource reallocation from lower QOS based services should be explored.
c. Develop a user friendly webapp to onboard xApp's & trigger policy towards near-RT RIC and support visualisations
d. Extend the solution to self-learning Closed Loops with following capabilities: -
   i. Continuously perform Collection, Analytics, Decision and Actuation
   ii. Detect model performance and trigger a switch-over to another better performing model
   iii. Analyse and trigger different set of data/measurements for data analysis
e. Points for future study from FGAN-O-013: -
   i. How ML pipelines can be synchronized/managed across the edge and emergency responder devices?

ii. The split of inference tasks/model functionalities between edge and emergency responder devices

8. REFERENCES

1. [1] A. Dandekar, J.Schulz-Zander, H.Wissing, Fraunhofer HHI, “Use case and requirements for orchestration of AI/ML basedclosed loops to enable autonomous networks”, Fraunhofer HHI, Apr., 2021.
2. [Build-a-thon FG AN] ITU-T FG AN-I-146 “Proposal for a “Build-a-thon” for ITU AI/ML in 5G Challenge (second edition, 2021), aligned with FGAN WG3” https://extranet.itu.int/sites/itu-t/focusgroups/an/input/FGAN-I-114-R1.docx
3. [Build-a-thon Challenge] ITU-T AI/ML in 5G Challenge problem statement “ITU-ML5G-PS-014: Build-a-thon (PoC) Network resource allocation for emergency management based on closed loop analysis” https://challenge.aiforgood.itu.int/match/matchitem/45
4. https://github.com/ITU-build-a-thon/challenge-resources/blob/main/intro_tutorial.pdf
5. FGAN-153 **“Team AUTOMATO”** https://extranet.itu.int/sites/itu-t/focusgroups/an/_layouts/15/WopiFrame.aspx?sourcedoc=%7B85757552-DFBE-479A-A816-003AE91C2B22%7D&file=FGAN-I-155.docx&action=default
6. Pre-trained model and repository https://github.com/krcmehmet/ITUChallenge_BuildaThon_Activity4
7. Near Realtime RIC https://wiki.o-ran-sc.org/display/GS/Near+Realtime+RIC+Installation
8. https://wiki.o-ran-sc.org/display/ORANSDK/App+Writing+Guide
9. https://github.com/o-ran-sc
10. https://lists.o-ran-sc.org/g/main/topics
11. https://docbox.etsi.org/ISG/ZSM/Open/Drafts/009-3ed111_Cla_AdvTop/ZSM-009-3_Cla_AdvTopv010

Appendix-1 Steps to recreate demo setup

1) Near Realtime RIC Installation

System requirements: -

VM Minimum Requirements for RIC 22 –
OS: Ubuntu 18.04 LTS (Bionic Beaver)
CPU(s): 4
RAM: 16 GB
Storage: 160 GB

The following are the steps to be followed to set up the RIC platform –

1. Obtaining the deployment scripts and charts
   - sudo -i
   - git clone http://gerrit.o-ran-sc.org/r/it/dep -b bronze
   - cd dep
   - git submodule update --init --recursive --remote
2. Change the repository value in the values.yaml file in the /ric-dep/helm/infrastructure/subcharts/kong/. Refer to the following link https://gerrit.o-ran-sc.org/r/c/ric-plt/ric-dep/+/6502/1/helm/infrastructure/subcharts/kong/values.yaml
3. Generation of cloud-init script
   - cd tools/k8s/bin
   - ./gen-cloud-init.sh
4. Installation of Kubernetes, Helm, Docker
   - ./k8s-1node-cloud-init-k_1_16-h_2_17-d_cur.sh
   - sudo -i
   - kubectl get gods –all-namespaces (There should be 9 pods running in kube-system namespace.)
5. Helm3 installation steps
   - curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3
   - chmod 700 get_helm.sh
   - ./get_helm.sh
6. Deploy RIC using Recipe
   - cd dep/bin
   - ./deploy-ric-platform -f ../RECIPE_EXAMPLE/PLATFORM/example_recipe_oran_dawn_release.yaml
   - kubectl get pods -n ricplt (There should be ~16 pods running in the ricplt namespace)

     RIC platform pods

2) E2 Simulator setup

1. Install dependencies

   - sudo apt-get update
   - sudo apt-get install -y build-essential git cmake libsctp-dev lksctp-tools autoconf automake libtool bison flex libboost-all-devsudo apt-get clean
   - sudo apt-get clean

2. Build and Deploy E2Simulator

   - clone sim/e2-interface
   - mkdir build
   - cd build
   - cmake ..
   - make package
   - cmake .. -DDEV_PKG=1
   - make package
   - In the root directory of e2sim: Follow directions in README to produce the deb files
   - Since the deb files are not yet pushed to package cloud, we need to copy them
   - cp e2sim*deb ../e2sm_examples/kpm_e2sm
   - cd ../e2sm_examples/kpm_e2sm
   - Edit the Dockerfile at the bottom to have IP address of service-ricplt-e2term-sctp-alpha service
   - docker build .
   - docker tag <image id> e2simul:1.0.0
   - docker run -d e2simul:1.0.0

3) XApp installation steps

- docker run --rm -u 0 -it -d -p 8090:8080 -e DEBUG=1 -e STORAGE=local -e STORAGE_LOCAL_ROOTDIR=/charts -v $(pwd)/charts:/charts chartmuseum/chartmuseum:latest
- export CHART_REPO_URL=http://0.0.0.0:8090
- cd appmgr/xapp_orchestrater/dev/xapp_onboarder
- pip3 uninstall xapp_onboarder
- pip3 install ./
- dms_cli onboard --config_file_path=config.json --shcema_file_path=/root/dep/bin/appmgr/xapp_orchestrater/dev/docs/xapp_onboarder/guide/embedded-schema.json
- curl -X GET http://localhost:8090/api/charts | jq .
- dms_cli install --xapp_chart_name=<app-name> --version=1.0.0 --namespace=ricxapp
- kubectl get pods -n ricxapp (xapp should be up and running)

For specific details on onboarding prbpred and alloc x-app please refer README in https://github.com/deenammd/ITUML5GChallenge_BuildaThon