5G Use Case Lab

User Manual

Disclaimer

Copyright

Confidentiality and Disclaimer

The contents of this document are strictly and highly confidential and are meant only for the perusal of the recipient to whom it has been delivered by W4S. No part of this document can be copied and or shared with any person who has not been authorised by W4S. The contents of the document are subject to revision without notice due to constant and continued progress in methodology, design and manufacturing. Further W4S is not providing any representation and/or warranty in respect of the contents of this document and W4S shall have no liability for any error or damage of any kind resulting from any use of this document.

Table Of Contents

1. About This Document 6

1.1. Introduction 6

1.2. Purpose 6

1.3. Scope 6

1.4.Intended Audience 6

1.5. Document Conventions 7

1.6. Terms and Abbreviations 7

2. NMS Overview 9

3. Steps For Using NMS 10

4. CORE Overview 12

4.1. Prerequisites for Core Implementation 13

4.1.1. Core Hardware Specification 13

4.1.2. Core Software Specification 13

4.1.3. Core Network Specification 14

4.1.4. Core Server Implementation 15

4.1.5. CORE Nodes 16

4.1.6. 5G Core Features 18

5. CORE Workflow 20

6. Hardware Configuration for CORE 21

7. Software Configuration for Pre-Configured CORE 22

7.1. Steps for Pre-Configured CORE 22

7.2. 5G Core Images 23

8. gNodeB Overview 24

8.1. Pre-requisites for gNodeB Implementation 25

8.1.1. gNodeB Hardware Specification 25

8.1.2. gNodeB Software Specification 26

8.1.3. gNodeB Network Specification 26

9. Hardware Configuration for RAN 29

10. Software Configuration for Pre-Configured RAN 30

10.1. Steps for Pre-Configured RAN 30

11. Configuration For Non Pre-Configured RAN 31

11.1. Assigning Static IP 31

11.2 Configuration for updation of GRUB 33

11.3. Ethernet Driver 34

11.4. Tuned Parameters 35

11.5. Ring Parameters and CPU frequency 35

11.6. Virtual Functions Creation 36

11.7. Dpdk Installation 37

11.8. Linuxptp Installation 37

11.9. Docker Installation 37

11.10. Create services for ptp4l and phc2sys 38
11.11. Configuration for Images 40
11.11.1 Check gNB Process Status 41
11.12. gNB Logs Parameters 41
11.12.1.gNB Log File Parameters Overview 42
11.12.1.1. Block Error Rate Overview 42
11.12.1.2. Radio Network Temporary Identifier Overview 42
11.12.1.3. Reference Signal Received Power Overview 43
11.12.1.4. Modulation and Coding Scheme Overview 43
11.12.1.5. Signal-to-Noise Ratio Overview 44
11.12.1.6. Logical Channel Identity Overview 44
11.12.1.7. Channel Quality Indicator Overview 44
11.12.1.8. Maximum Transmission Power Overview 45
11.12.1.9. Precoding Matrix Indicator Overview 45
11.12.1.10. Rank Indicator Overview 45
11.12.1.11. Medium Access Control Overview 45
11.13.   Configuration for Network slicing 45
11.13.1.Configuration for Network Slice 1 45
11.13.2. Configuration for Network Slice 2 49
11.13.3.Configuration for Network Slice 3 54
11.14. Configuration for 1x1 58
11.15. Configuration for 2x2 72
12. Mobile Edge Compute 87
12.1. MEC Hardware Specification 88
12.2. Software Specification 88
12.3. Network Specification 88
13. Commands for MEC 90
13.1. Route Configuration for Mec Applications 90
13.2. Camera 90
13.2.1 Camera Connection through LAN 90
13.2.2. Wireless Camera Connection 90
13.2.3. Camera Usage Steps 91
13.3. Drone 91
14. RU Overview 92
14.1. Key Specifications: 92
15. Configuration for RU 93
15.1. Steps for Pre-Configured RU 93
16. APN configuration 96
16.1.Configuration for IoT Gateway 96
16.2. Configuration for Camera 97
16.3. Configuration for CPE 98

16.4. Configuration for Evaluation Board 101

16.5. Configuration for Drone 107

17. Configuration For Sensors 108

17.1. IoT Setup 108

17.2. Configuration for NPK Sensor 109

17.3. Configuration for Light, TDS and Temperature and Humidity Sensors 110

17.4. Configuration steps for Thingsboard 111

17.5. Configuration Steps for AR/VR 114

17.6. Configuration Steps for iDronam 114

17.7. Configuration Steps for Sparsh Camera 117

18. Firewall Configuration 122

18.1. Zones 122

18.2. Configuration for Policy 123

18.3. Rules 124

18.4. Port Forwarding 127

19. Troubleshooting for CORE 129

19.1. Logs for CORE 129

20. Troubleshooting for RAN 130

20.1. Configuration of Wireshark for capturing RAN Logs: 130

20.2. Capturing Logs For RAN 132

20.3. Troubleshooting for IP Route 133

21. Troubleshooting for the Drivers 135

22. Troubleshooting for RU 137

22.1. Troubleshooting for LPRU configuration 138

23. Troubleshooting for IoT Sensors 140

23.1. Non-Working sensors 140

23.2. Finding the IP of the MEC Server 141

Reference Links 143

1. About This Document

1.1. Introduction

This guide is for Core, gNodeB and RU Configuration and user guidance.

1.2. Purpose

This document provides Core, gNodeB and RU configuration which includes hardware, software management and configuration for Core, gNodeB, RU.

1.3. Scope

This document specifies the configuration supported by Core, gNodeB and RU.

1.4. Intended Audience

The intended target groups for this document are:

- Software Installation/Support Engineers.

1.5. Document Conventions

The different conventions used in this document are explained in the following table:

Table 1: Document Conventions

Convention	Description
Note:	Note provides information about important features or instructions. This appears with a background.
Courier New font	File and directory names are represented in Courier New font.
Bold font	Any option that needs to be selected or typed in the user interface is represented using bold font.
<home directory>	Command variables, the values of which you must supply.

1.6. Terms and Abbreviations

Sl.No	Abbreviations	Longform
1.	DU	Distributed Unit
2.	CU	Central Unit
3.	UE	User Equipment
4.	GPS	Global Positioning System
5.	SRS	Sounding Reference Signal
6.	CQI	Channel Quality Indicator

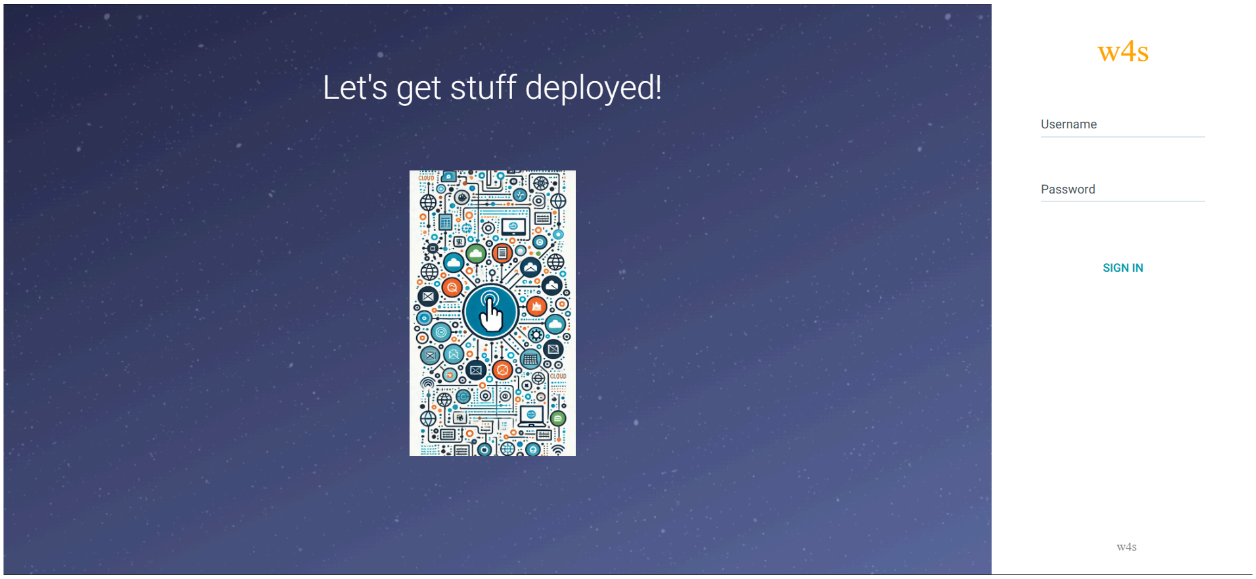
7.	NGAP	Next Generation Application Protocol
8.	SCTP	Stream Control Transmission Protocol
9.	RF	Radio Frequency
10.	RU	Radio Unit
11.	SCS	Subcarrier Spacing
12.	RRC	Radio Resource Control
13.	RNTI	Radio Network Temporary Identifier
14.	RSRP	Reference Signal Received Power
15.	RSRQ	Reference Signal Received Quality
16.	SINR	Signal-to-Interference-plus-Noise Ratio
17.	MCS	Modulation and Coding Scheme
18.	BLER	Block Error Rate
19.	PDSCH	Physical Downlink Shared Channel
20.	PUSCH	Physical Uplink Shared Channel
21.	PUCCH	Physical Uplink Control Channel

2. NMS Overview

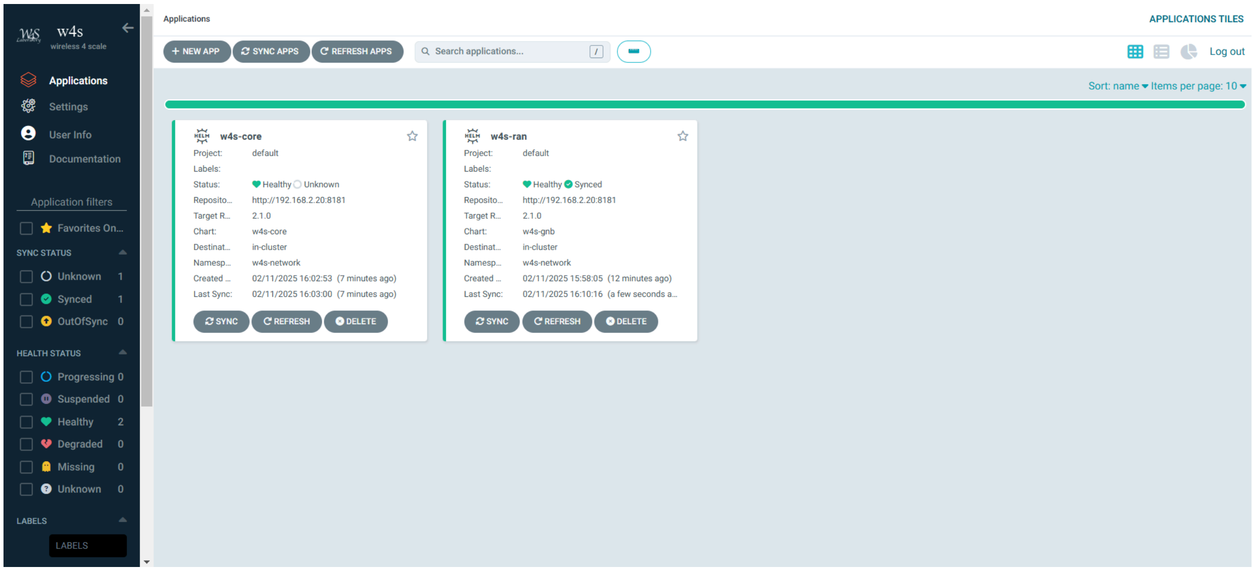
The Network Management System (NMS) plays a crucial role in 5G private networks by ensuring seamless monitoring, control, and optimization of network operations. It enhances network performance, connectivity, and security, making it essential for industries that rely on reliable and high-performance communication infrastructure. One of its primary functions is network monitoring and fault management, where it continuously oversees network health, detects issues such as hardware failures or interference, and triggers alerts for proactive maintenance. Additionally, NMS contributes to performance optimization by minimizing latency and maximizing reliability, particularly for mission-critical applications like industrial automation and healthcare.

3. Steps For Using NMS

1. Use Institute IP 192.168.x.xx:8080 to open NMS in the browser.
2. Use authorized credentials to login into NMS.



3. After Logging in→Applications→gride icon to see health and synced status of Core and Ran.



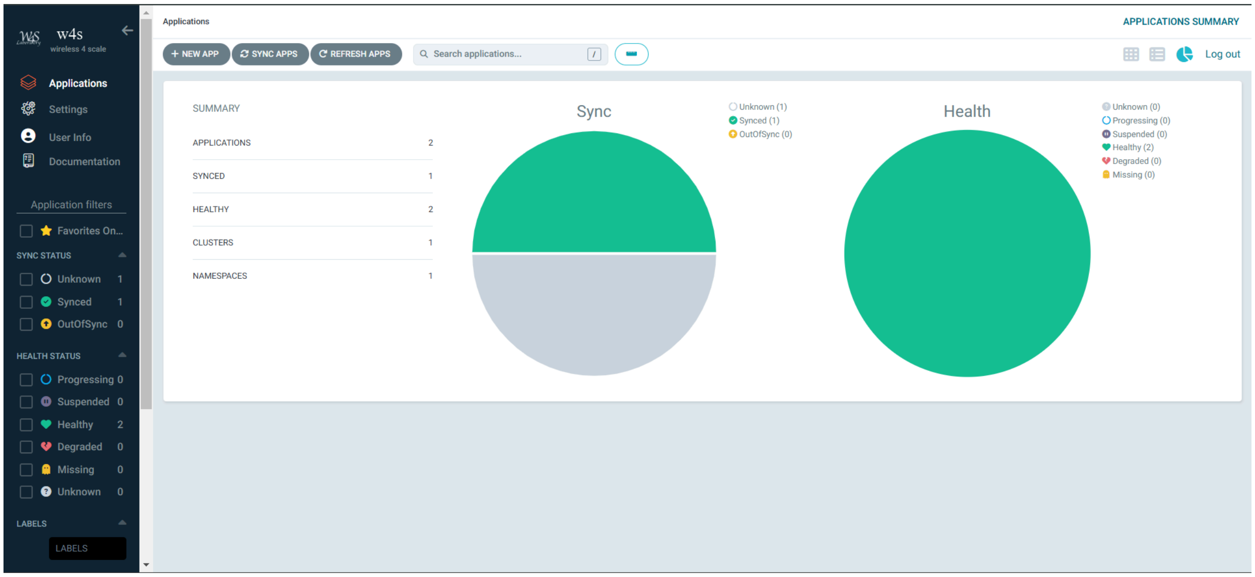
w4s-core

Project: default
Labels:
Status: Healthy Unknown
Reposito... http://192.168.2.20:8181
Target R... 2.1.0
Chart: w4s-core
Destinat... in-cluster
Namesp... w4s-network
Created ... 02/11/2025 16:02:53 (7 minutes ago)
Last Sync: 02/11/2025 16:03:00 (7 minutes ago)

w4s-ran

Project: default
Labels:
Status: Healthy Synced
Reposito... http://192.168.2.20:8181
Target R... 2.1.0
Chart: w4s-gnb
Destinat... in-cluster
Namesp... w4s-network
Created ... 02/11/2025 15:58:05 (12 minutes ago)
Last Sync: 02/11/2025 16:10:16 (a few seconds a...

4. Applications→pie chart icon for seeing summary, sync and health of Core and Ran.

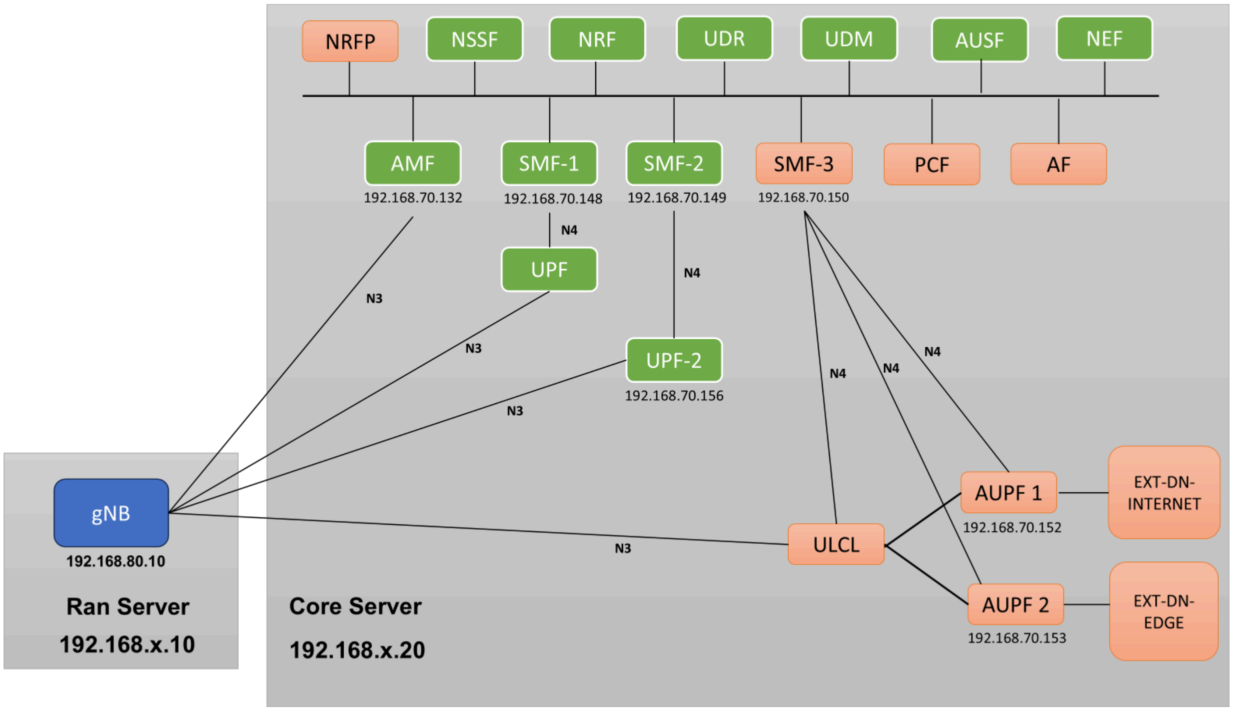


SUMMARY

APPLICATIONS	2
SYNCED	1
HEALTHY	2
CLUSTERS	1
NAMESPACES	1

4. CORE Overview

The 5G Core is designed using a service-based architecture (SBA) to support the full potential of 5G, including high data throughput, low latency, and robust scalability. It incorporates various key components such as the Access and Mobility Management Function (AMF) for managing device registration and mobility, the Session Management Function (SMF) for session control, and the User Plane Function (UPF) for data traffic management. The 5GC also includes other critical elements like the Policy Control Function (PCF) for policy enforcement, the Network Slice Selection Function (NSSF) for enabling network slicing, and the Authentication Server Function (AUSF) for secure user authentication. Together, these functions facilitate advanced 5G features like network slicing making 5G NR SA ideal for complex use cases such as autonomous driving, smart cities, and industrial IoT applications. The standalone architecture's independence from legacy 4G networks enables operators to unlock the full capabilities of 5G, paving the way for enhanced mobile broadband (eMBB) and the future of seamless digital connectivity.



4.1. Prerequisites for Core Implementation

4.1.1. Core Hardware Specification

Server Type	CORE
Processor Model	Intel® Xeon Gold 3.0 GHz 5317 x 2
Memory(RAM)	32 GB DDR4 x 2
Storage Capacity	1.92 TB x 2
BIOS Settings	default
Operating System	Ubuntu 22.04 Server Edition
Kernel	Realtime Kernel

4.1.2. Core Software Specification

Software Package	CORE
Docker	>= Version 26
Docker-compose	Version 1.29.2

4.1.3. Core Network Specification

Server Type	**CORE**
Server IP	192.168.x.20
Netmask /Gateway :192.168.x.1	255.255.255.0
Ip forwarding configuration	Ip forwarding enabled

4.1.4. Core Server Implementation

CORE Implementation	Details
Deployment Type	containerized Core
Implementation Language	C , C++
Thread Management	Dynamic
Interfaces	Standard Interfaces N1 , N2 , N3 , N4 , N6 , N9 , NRF are used for intercommunication via sbi.
Architecture	Service Based Architecture
Slices	Slice 1 : SST :128 SD:128 Slice 2 : SST :1 SD:1 Slice 3 : SST :130 SD:130
APN configuration Slice wise	Slice 1 : 5glabs1 Slice 2 : 5glabs2 Slice 3 : 5glab

Core nodes Implemented	AMF , NSSF , SMF , UPF , UDM , UDR , NRF , PCF ,EXT-DN, IMS , AUSF , ULCL , NEF, AF

4.1.5. CORE Nodes

CORE Nodes	Container Name	Introduction
AMF	w4s-amf	Responsible for managing user access, user authentication, session management and mobility within the 5G network
AUSF	w4s-ausf	Responsible for verifying the identity of users and ensuring secure access to network resources. The AUSF interacts with the UDM (Unified Data Management) to retrieve authentication data and supports various authentication methods to enhance security.
UDM	w4s-udm	Tasked with managing user data and subscription information. It stores and retrieves user profiles, subscription details, and authentication credentials. The UDM ensures consistent and efficient handling of user data across different network functions, enabling personalized and seamless service delivery.
UDR	w4s-udr	The UDR acts as a centralized repository for storing and managing network data. It provides a unified storage solution for various types of data, including subscriber profiles, network configuration, and service-related information. The UDR ensures data consistency and availability across the 5G core network.

SMF	w4s-smf-slice1 w4s-smf-slice2 w4s-smf-slice3	The SMF is responsible for establishing, modifying, and releasing sessions within the 5G network. It manages session continuity, quality of service (QoS), and bearer paths. The SMF coordinates with the UPF (User Plane Function) to ensure efficient data transfer and service delivery.
NSSF	w4s-nssf	The NSSF facilitates network slicing by selecting the appropriate network slice for a given user session. It manages network slice selection based on the service requirements and user profile.
PCF	w4s-pcf	The PCF provides policy and charging control functions across the 5G network. It enforces policies related to service quality, resource allocation, and charging rules
NRF	w4s-nrf w4s-nrfp	The NRF acts as a centralized registry, enabling the registration and discovery of Network Functions (NFs) to facilitate efficient service-based communication. The NRF ensures secure interactions by authenticating and authorizing NFs and optimizing network performance through load balancing and resilience mechanisms.
IMS	w4s-ims	IMS delivers multimedia services over IP networks. It provides a standardized approach to delivering services such as voice, video, and messaging.
EXT-DN	w4s-ext-dn-internet w4s-ext-dn-edge	Ext-dn provides external internet world access to the devices connected to the private 5g network

UPF	w4s-upf-slice1 w4s-upf-slice2 w4s-upf-ulcl w4s-upf-aupf1 w4s-upf-aupf2	The UPF is responsible for handling user plane data traffic. It manages the routing and forwarding of user data packets between the gNB (gNodeB) and external networks.
NEF	w4s-nef	The NEF a secure gateway between the 5G Core Network and external applications or systems. It enables controlled and secure access to the network's services and capabilities.
AF	w4s-af	The AF works with PCF to define QoS requirements and traffic management rules. It integrates application services like IoT, streaming, and AR with the network's capabilities.

4.1.6. 5G Core Features

Multi Segment Offering	Built for multiple use cases ranging from 5G device/application/network testing, building 5G labs, and enterprise private networks
Network Slicing	Enables isolation , independent scalability , flexible deployments & configuration of the network as per the needs of the different services.
Policy Control	Policy based upf selection for internet or edge access
UE integration	Integration with COTS UEs,5g modems .

Management	The 5G Core components are deployed in containers as fully orchestrated by Container orchestrators .

5. CORE Workflow

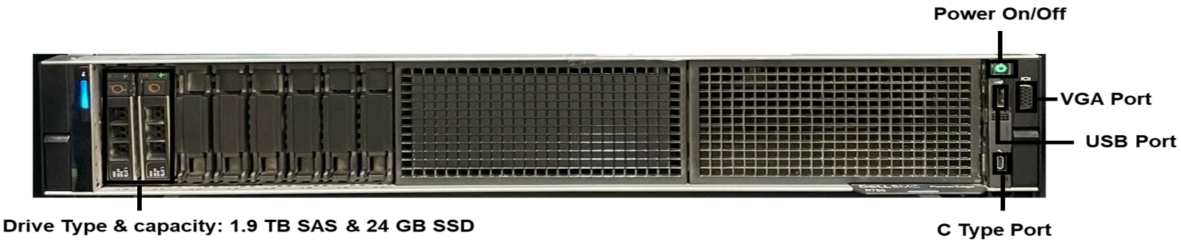
The Core network and Service Management and Orchestration (SMO) layer collaboratively manage connectivity and service delivery. The 5G Core (5GC) is responsible for essential functions including user registration, authentication, session management, mobility handling, and data routing between user devices and external networks. Core components such as the Access and Mobility Management Function (AMF), Session Management Function (SMF), and User Plane Function (UPF) facilitate efficient connectivity, seamless mobility, and high-performance service access.

One of 5GC primary functions is **user registration and authentication**, which ensures that only authorized users can access the network. This process is managed by the **Unified Data Management (UDM)** and **Authentication Server Function (AUSF)**, which verify user identities and enable secure access.

Another crucial function of the 5GC is **session management**, handled by the **Session Management Function (SMF)**. This component establishes, maintains, and terminates user sessions while assigning IP addresses and configuring Quality of Service (QoS) parameters to ensure optimal network performance. Additionally, the **Access and Mobility Management Function (AMF)** is responsible for **mobility management**, enabling seamless transitions between network cells and ensuring continuous connectivity, even when users move between different coverage areas.

The **User Plane Function (UPF)** plays a key role in **data routing and forwarding** by directing user traffic between mobile devices and external networks, such as the internet or private enterprise networks. This ensures efficient data transmission while maintaining low latency and optimized traffic flow. Furthermore, the 5G Core supports **network slicing**, which allows multiple virtual networks to operate on the same physical infrastructure. This feature enables differentiated services tailored to specific use cases, such as ultra-reliable low-latency communication (URLLC) for autonomous systems and enhanced mobile broadband (eMBB) for high-speed streaming.

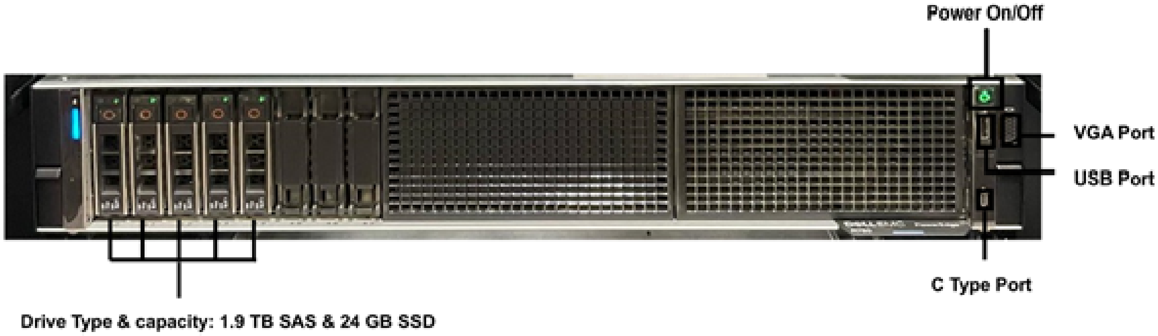
6. Hardware Configuration for CORE



The image above provides a detailed overview of the functionality of each switch, along with the corresponding drive type and capacity utilized in the hardware.

Check that UPS power is ON and input power is available for the servers.

Switch ON the Core ON/OFF button and MEC ON/OFF button.



MEC

7. Software Configuration for Pre-Configured CORE

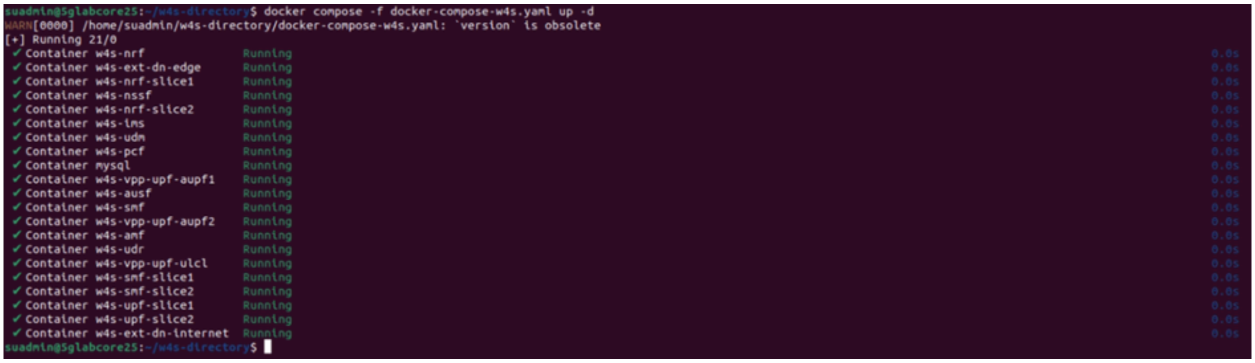
7.1. Steps for Pre-Configured CORE

1.Open the terminal window in MEC.

2.Log in into Core: ssh suadmin@192.168.xx.20(xx refers to subnet based on institute)
Password: Admin@123

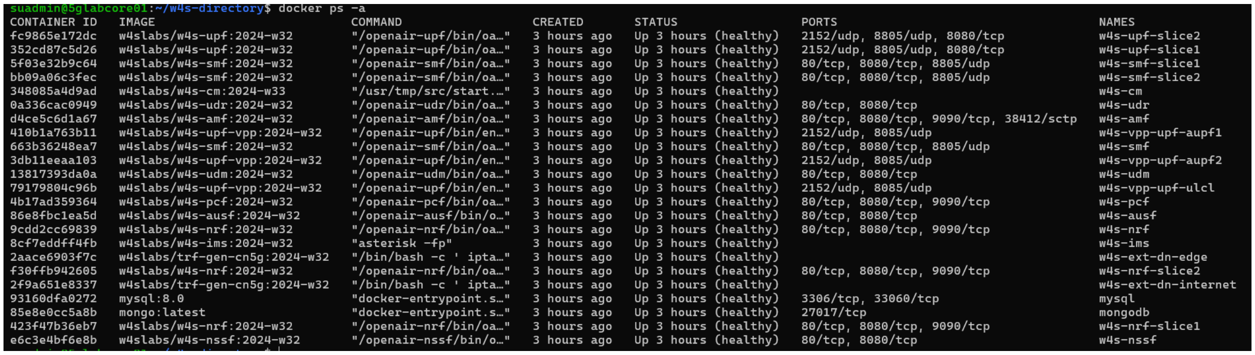
3.Change the directory using **cd w4s-directory** command.

4.To start the core services execute **docker compose -f docker-compose-w4s.yaml up -d** command.



```
suadmin@5glabcore25:~/w4s-directory$ docker compose -f docker-compose-w4s.yaml up -d
WARN[0000] /home/suadmin/w4s-directory/docker-compose-w4s.yaml: `version` is obsolete
[+] Running 21/0
 ✔ Container w4s-nrf               Running                                                              0.0s
 ✔ Container w4s-ext-dn-edge       Running                                                              0.0s
 ✔ Container w4s-nrf-slice1        Running                                                              0.0s
 ✔ Container w4s-nssf              Running                                                              0.0s
 ✔ Container w4s-nrf-slice2        Running                                                              0.0s
 ✔ Container w4s-ims               Running                                                              0.0s
 ✔ Container w4s-udm               Running                                                              0.0s
 ✔ Container w4s-pcf               Running                                                              0.0s
 ✔ Container mysql                 Running                                                              0.0s
 ✔ Container w4s-vpp-upf-aupf1     Running                                                              0.0s
 ✔ Container w4s-ausf              Running                                                              0.0s
 ✔ Container w4s-smf               Running                                                              0.0s
 ✔ Container w4s-vpp-upf-aupf2     Running                                                              0.0s
 ✔ Container w4s-amf               Running                                                              0.0s
 ✔ Container w4s-udr               Running                                                              0.0s
 ✔ Container w4s-vpp-upf-ulcl      Running                                                              0.0s
 ✔ Container w4s-smf-slice1        Running                                                              0.0s
 ✔ Container w4s-smf-slice2        Running                                                              0.0s
 ✔ Container w4s-upf-slice1        Running                                                              0.0s
 ✔ Container w4s-upf-slice2        Running                                                              0.0s
 ✔ Container w4s-ext-dn-internet   Running                                                              0.0s
suadmin@5glabcore25:~/w4s-directory$ 
```

5.Check the status of all the services using **docker ps -a** command.



```
suadmin@5glabcore01:~/w4s-directory$ docker ps -a
CONTAINER ID   IMAGE                            COMMAND                  CREATED       STATUS                 PORTS                                      NAMES
fc9865e172dc   w4slabs/w4s-upf:2024-w32         "/openair-upf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   2152/udp, 8805/udp, 8080/tcp               w4s-upf-slice2
352cd87c5d26   w4slabs/w4s-upf:2024-w32         "/openair-upf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   2152/udp, 8805/udp, 8080/tcp               w4s-upf-slice1
5f03e32b9c64   w4slabs/w4s-smf:2024-w32         "/openair-smf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp, 8805/udp                 w4s-smf-slice1
bb09a06c3fec   w4slabs/w4s-smf:2024-w32         "/openair-smf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp, 8805/udp                 w4s-smf-slice2
348085a4d9ad   w4slabs/w4s-cm:2024-w33          "/usr/tmp/src/start.…"   3 hours ago   Up 3 hours (healthy)                                              w4s-cm
0a336cac0949   w4slabs/w4s-udr:2024-w32         "/openair-udr/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp                           w4s-udr
d4ce5c6d1a67   w4slabs/w4s-amf:2024-w32         "/openair-amf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp, 9090/tcp, 38412/sctp     w4s-amf
410b1a763b11   w4slabs/w4s-upf-vpp:2024-w32     "/openair-upf/bin/en…"   3 hours ago   Up 3 hours (healthy)   2152/udp, 8085/udp                         w4s-vpp-upf-aupf1
663b36248ea7   w4slabs/w4s-smf:2024-w32         "/openair-smf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp, 8805/udp                 w4s-smf
3db11eeaa103   w4slabs/w4s-upf-vpp:2024-w32     "/openair-upf/bin/en…"   3 hours ago   Up 3 hours (healthy)   2152/udp, 8085/udp                         w4s-vpp-upf-aupf2
13817393da0a   w4slabs/w4s-udm:2024-w32         "/openair-udm/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp                           w4s-udm
79179804c96b   w4slabs/w4s-upf-vpp:2024-w32     "/openair-upf/bin/en…"   3 hours ago   Up 3 hours (healthy)   2152/udp, 8085/udp                         w4s-vpp-upf-ulcl
4b17ad359364   w4slabs/w4s-pcf:2024-w32         "/openair-pcf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp, 9090/tcp                 w4s-pcf
86e8fbc1ea5d   w4slabs/w4s-ausf:2024-w32        "/openair-ausf/bin/o…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp                           w4s-ausf
9cdd2cc69839   w4slabs/w4s-nrf:2024-w32         "/openair-nrf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp, 9090/tcp                 w4s-nrf
8cf7eddff4fb   w4slabs/w4s-ims:2024-w32         "asterisk -fp"           3 hours ago   Up 3 hours (healthy)                                              w4s-ims
2aace6903f7c   w4slabs/trf-gen-cn5g:2024-w32    "/bin/bash -c ' ipta…"   3 hours ago   Up 3 hours (healthy)                                              w4s-ext-dn-edge
f30ffb942605   w4slabs/w4s-nrf:2024-w32         "/openair-nrf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp, 9090/tcp                 w4s-nrf-slice2
2f9a651e8337   w4slabs/trf-gen-cn5g:2024-w32    "/bin/bash -c ' ipta…"   3 hours ago   Up 3 hours (healthy)                                              w4s-ext-dn-internet
93160dfa0272   mysql:8.0                        "docker-entrypoint.s…"   3 hours ago   Up 3 hours (healthy)   3306/tcp, 33060/tcp                        mysql
85e8e0cc5a8b   mongo:latest                     "docker-entrypoint.s…"   3 hours ago   Up 3 hours (healthy)   27017/tcp                                  mongodb
423f47b36eb7   w4slabs/w4s-nrf:2024-w32         "/openair-nrf/bin/oa…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp, 9090/tcp                 w4s-nrf-slice1
e6c3e4bf6e8b   w4slabs/w4s-nssf:2024-w32        "/openair-nssf/bin/o…"   3 hours ago   Up 3 hours (healthy)   80/tcp, 8080/tcp                           w4s-nssf
```

To check Log’s for core: docker logs -f <container_name> for example: docker logs -f w4s-amf

6.Execute **docker-compose -f docker-compose-w4s.yaml down** command to shut down the core services.

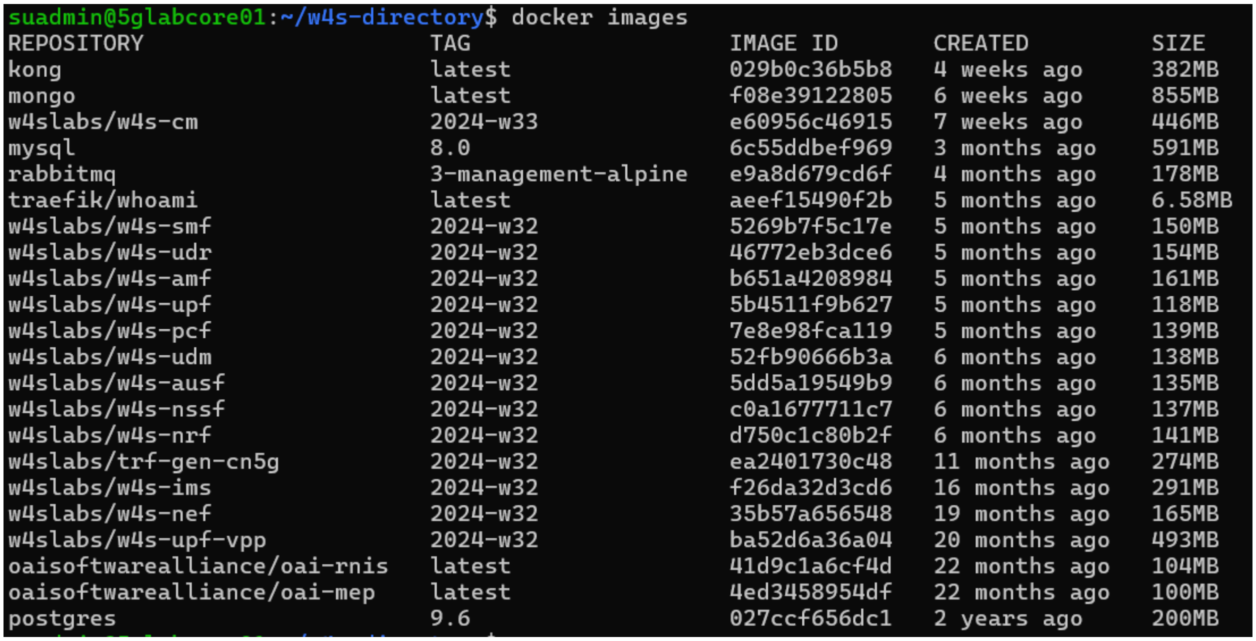
7.For Loging out from Core execute **exit** command.

7.2. 5G Core Images

The core Network functions are deployed as containerized applications which are run on the core server via a container orchestration service like docker. To view the container images which are used to deploy the container for the different network functions execute the below mentioned command.

Execute the below command on core server:

docker images

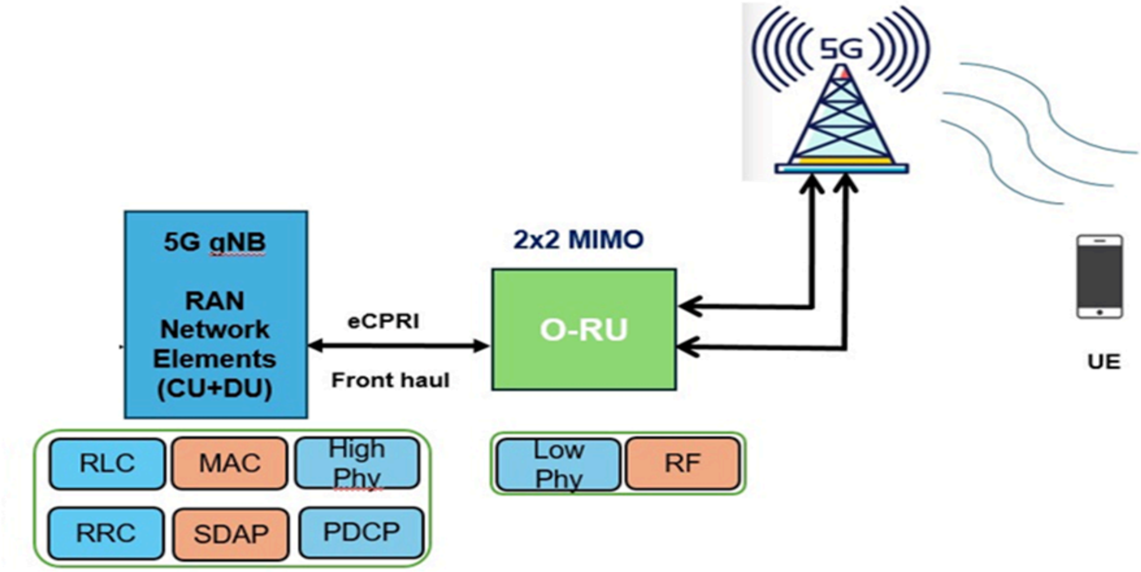


```
suadmin@5glabcore01:~/w4s-directory$ docker images
REPOSITORY                     TAG                    IMAGE ID       CREATED          SIZE
kong                           latest                 029b0c36b5b8   4 weeks ago      382MB
mongo                          latest                 f08e39122805   6 weeks ago      855MB
w4slabs/w4s-cm                 2024-w33               e60956c46915   7 weeks ago      446MB
mysql                          8.0                    6c55ddbef969   3 months ago     591MB
rabbitmq                       3-management-alpine    e9a8d679cd6f   4 months ago     178MB
traefik/whoami                 latest                 aeef15490f2b   5 months ago     6.58MB
w4slabs/w4s-smf                2024-w32               5269b7f5c17e   5 months ago     150MB
w4slabs/w4s-udr                2024-w32               46772eb3dce6   5 months ago     154MB
w4slabs/w4s-amf                2024-w32               b651a4208984   5 months ago     161MB
w4slabs/w4s-upf                2024-w32               5b4511f9b627   5 months ago     118MB
w4slabs/w4s-pcf                2024-w32               7e8e98fca119   5 months ago     139MB
w4slabs/w4s-udm                2024-w32               52fb90666b3a   6 months ago     138MB
w4slabs/w4s-ausf               2024-w32               5dd5a19549b9   6 months ago     135MB
w4slabs/w4s-nssf               2024-w32               c0a1677711c7   6 months ago     137MB
w4slabs/w4s-nrf                2024-w32               d750c1c80b2f   6 months ago     141MB
w4slabs/trf-gen-cn5g           2024-w32               ea2401730c48   11 months ago    274MB
w4slabs/w4s-ims                2024-w32               f26da32d3cd6   16 months ago    291MB
w4slabs/w4s-nef                2024-w32               35b57a656548   19 months ago    165MB
w4slabs/w4s-upf-vpp            2024-w32               ba52d6a36a04   20 months ago    493MB
oaisoftwarealliance/oai-rnis   latest                 41d9c1a6cf4d   22 months ago    104MB
oaisoftwarealliance/oai-mep    latest                 4ed3458954df   22 months ago    100MB
postgres                       9.6                    027ccf656dc1   2 years ago      200MB
```

8. gNodeB Overview

gNB, also known as gNodeB, is a pivotal element within the 5G cellular network architecture. It plays a critical role in facilitating the next generation of wireless communication by providing the essential infrastructure to support high-speed data transfer, low latency, and massive connectivity.

Consisting of integrated Central Unit (CU) and Distributed Unit (DU), and Radio Unit (RU), its architecture is designed to uphold high data rates, low latency, and massive connectivity. gNB leverages advanced technologies such as MIMO and network slicing to ensure an optimized and versatile network that meets diverse application requirements, thereby paving the way for future innovations.



8.1. Pre-requisites for gNodeB Implementation

8.1.1. gNodeB Hardware Specification

Server Type	**gNB**
Processor Model	Intel® Core i7 11700k
Memory(RAM)	32 GB DDR4
Storage Capacity	500 GB
BIOS Settings	Hyper Threading , sr-iov , intel c-state off , xml profile 1
Operating System	Ubuntu 22.04 Server Edition
Kernel	Realtime Kernel

8.1.2. gNodeB Software Specification

Software Package	gNB
Docker	>= Version 26
linuxptp	1.9.2-1ubuntu0.1

8.1.3. gNodeB Network Specification

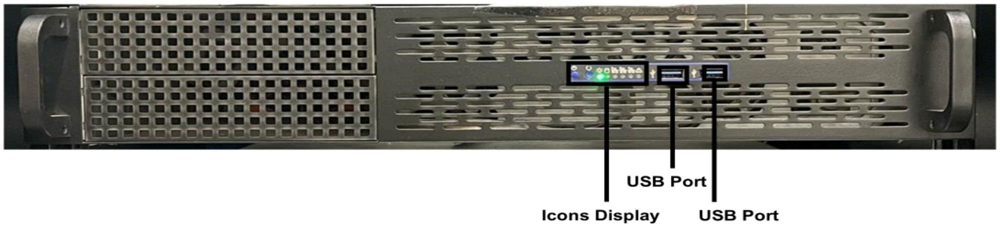
Server Type	gNB
Server IP	192.168.x.10
Netmask Gateway :192.168.x.1	255.255.255.0
Ip forwarding configuration	Ip forwarding enabled
Ip route configuration	192.168.70.0/24 via 192.168.x.20,192.168.72.0/24 via 192.168.x.20

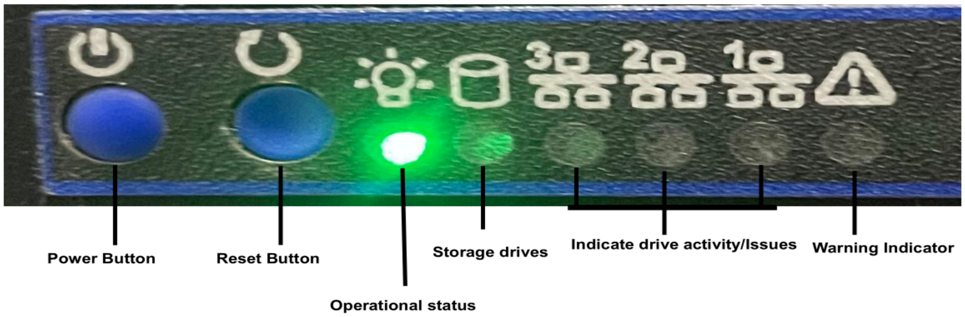
6.1.4. gNodeB Server Implementation

gNB Implementation	Details
Deployment Type	Containerized RAN
Implementation Language	C , C++
Container IP for RAN	192.168.80.10
Thread Management	Thread for RAN are assigned to exclusive isolated cores
AMF Connection	192.168.80.10
Slices	Slice 1 : SST :128 SD:128 Slice 2 : SST :1 SD:1 Slice 3 : SST :130 SD:130
UPF Connection	192.168.80.10:2152
Architecture	Monolithic
Split Type	7.2 Split
Interface b/w gNB & RU	2 Virtual Function on the gNB interface

Synchronization	Linuxptp (ptp4l & phc2sys) on gNB server with RU

9. Hardware Configuration for RAN





The images above illustrate the icons and their respective functions within the hardware.

10. Software Configuration for Pre-Configured RAN

10.1. Steps for Pre-Configured RAN

1. Log in gNB by executing **ssh suadmin@192.168.xx.10** (xx refers to the subnet based on institute) command.
Password: Admin@123

2. Execute RU steps for Login in LPRU.

NOTE: Please Refer to Steps for Pre-Configured RU.

3.To Start gNB: sudo systemctl start ran **OR** suadmin@5glabgnb:~$ sudo docker run -d --net w4s-network --ip 192.168.80.10 -v /home/suadmin/T.2x2.oran.fhi.7.2.conf:/opt/oai-gnb/etc/gnb.conf -v /dev/hugepages:/dev/hugepages -v /tmp:/tmp --cgroup-parent ran.slice --cap-add SYS_ADMIN --cap-add IPC_LOCK --cap-add SYS_NICE –env USE_ADDITIONAL_OPTIONS="--sa --opt.type pcap --opt.path /tmp/ng-ran.pcap --thread-pool 3,4,5,6,7" --device /dev/vfio:/dev/vfio/ --env TZ=Asia/Kolkata --name w4s-gnb w4slabs/w4s-gnb-fhi-e:2024-w32 taskset -c 10 /opt/oai-gnb/bin/nr-softmodem -O /opt/oai-gnb/etc/gnb.conf

4. Use command **docker ps** to check the status of all the services.

5.Open gNB Logs by using **docker logs -f w4s-gnb** command.

6. gNB Stop Command: sudo systemctl stop ran **OR** docker stop w4s-gnb && docker rm w4s-gnb

7. To logout from gNB execute the **exit** command.

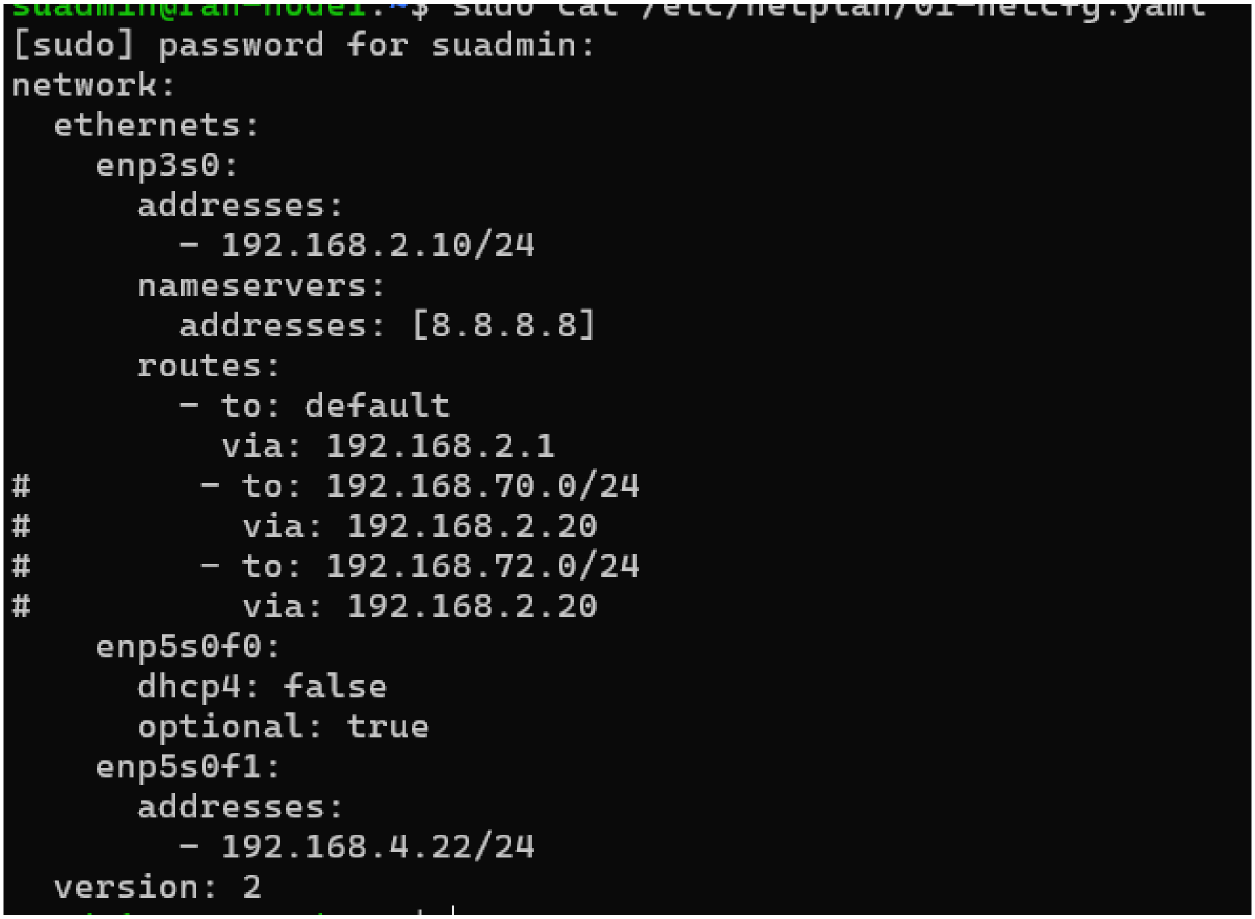
11. Configuration For Non Pre-Configured RAN

Log in gNB: ssh suadmin@192.168.xx.10 (xx refers for the subnet based on institute)
Password: Admin@123

11.1. Assigning Static IP

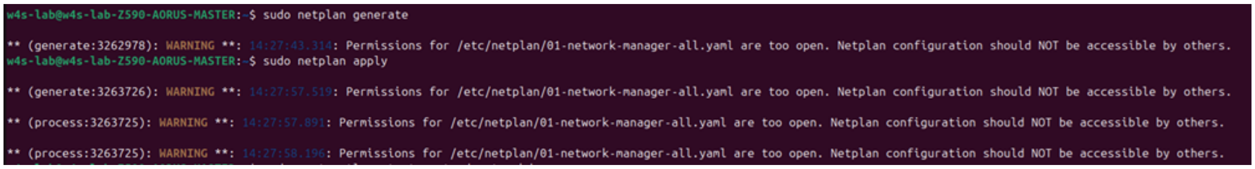
Edit the file in (Ubuntu) **/etc/netplan/01-netcfg.yaml**

** (the static ip of the interface has to be in the lab specific subnet 192.168.2.0/24 is an example)



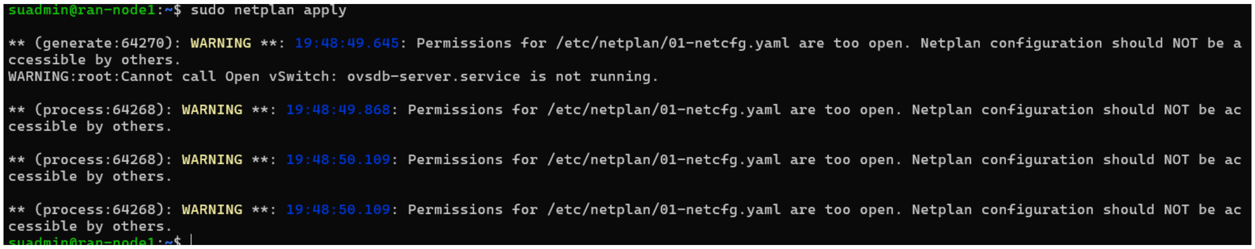
```
suadmin@ran-node1:~$ sudo cat /etc/netplan/01-netcfg.yaml
[sudo] password for suadmin:
network:
  ethernets:
    enp3s0:
      addresses:
        - 192.168.2.10/24
      nameservers:
        addresses: [8.8.8.8]
      routes:
        - to: default
          via: 192.168.2.1
#        - to: 192.168.70.0/24
#          via: 192.168.2.20
#        - to: 192.168.72.0/24
#          via: 192.168.2.20
    enp5s0f0:
      dhcp4: false
      optional: true
    enp5s0f1:
      addresses:
        - 192.168.4.22/24
  version: 2
```

Now type “**sudo netplan generate**” command,



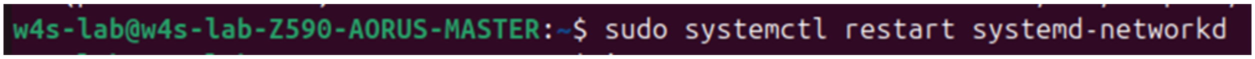
```
w4s-lab@w4s-lab-Z590-AORUS-MASTER:~$ sudo netplan generate

** (generate:3262978): WARNING **: 14:27:43.314: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.
w4s-lab@w4s-lab-Z590-AORUS-MASTER:~$ sudo netplan apply

** (generate:3263726): WARNING **: 14:27:57.519: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:3263725): WARNING **: 14:27:57.891: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:3263725): WARNING **: 14:27:58.196: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.
```

After this type “**sudo netplan apply**” command as stated below:



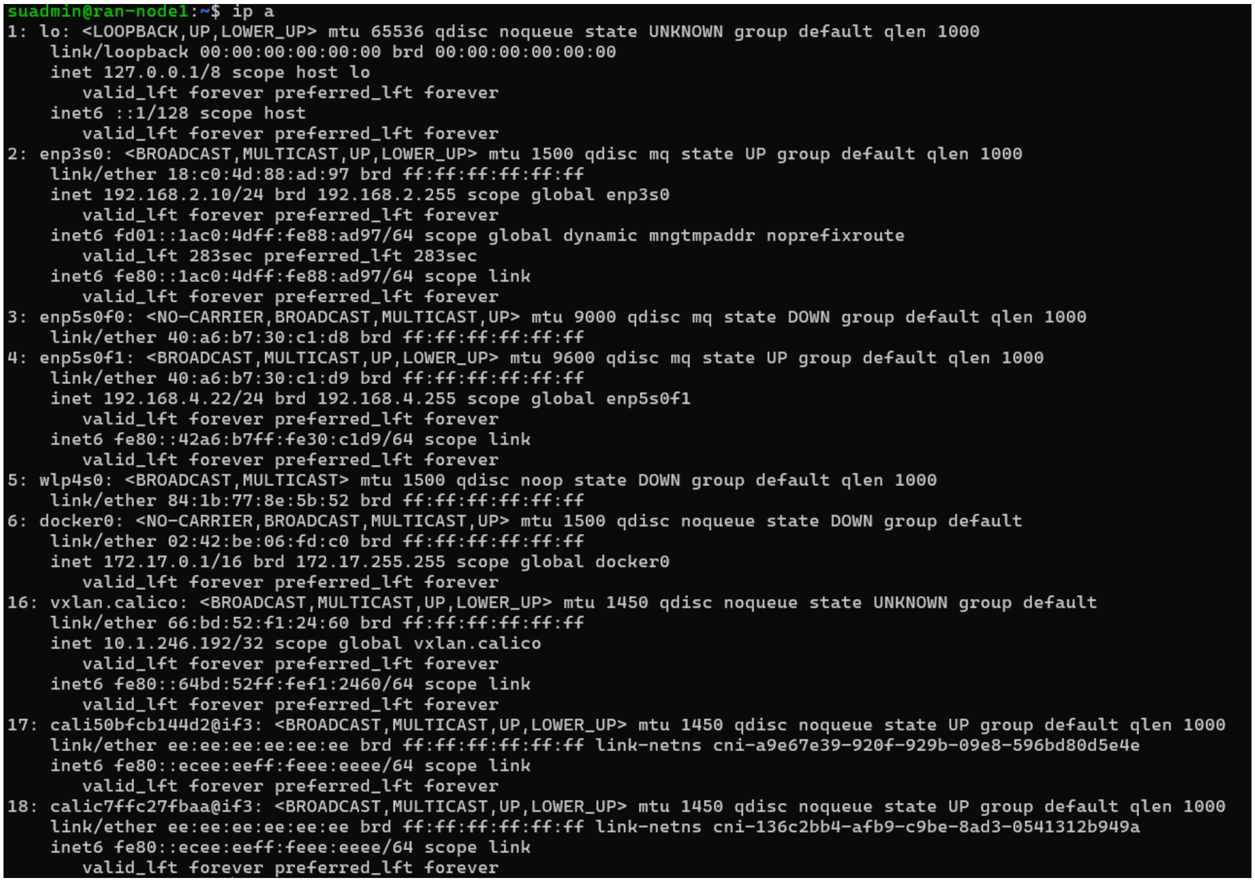
```
suadmin@ran-node1:~$ sudo netplan apply

** (generate:64270): WARNING **: 19:48:49.645: Permissions for /etc/netplan/01-netcfg.yaml are too open. Netplan configuration should NOT be a
ccessible by others.
WARNING:root:Cannot call Open vSwitch: ovsdb-server.service is not running.

** (process:64268): WARNING **: 19:48:49.868: Permissions for /etc/netplan/01-netcfg.yaml are too open. Netplan configuration should NOT be ac
cessible by others.

** (process:64268): WARNING **: 19:48:50.109: Permissions for /etc/netplan/01-netcfg.yaml are too open. Netplan configuration should NOT be ac
cessible by others.

** (process:64268): WARNING **: 19:48:50.109: Permissions for /etc/netplan/01-netcfg.yaml are too open. Netplan configuration should NOT be ac
cessible by others.
suadmin@ran-node1:~$
```

Now execute “**sudo systemctl restart systemd-networkd**” command.



```
w4s-lab@w4s-lab-Z590-AORUS-MASTER:~$ sudo systemctl restart systemd-networkd
```

Now, check the IP address by using “ip a” command to confirm the configuration.

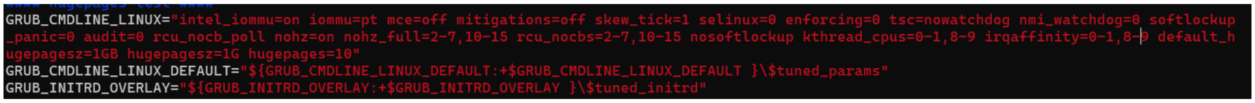


```
suadmin@ran-node1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 18:c0:4d:88:ad:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.10/24 brd 192.168.2.255 scope global enp3s0
       valid_lft forever preferred_lft forever
    inet6 fd01::1ac0:4dff:fe88:ad97/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 283sec preferred_lft 283sec
    inet6 fe80::1ac0:4dff:fe88:ad97/64 scope link
       valid_lft forever preferred_lft forever
3: enp5s0f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 9000 qdisc mq state DOWN group default qlen 1000
    link/ether 40:a6:b7:30:c1:d8 brd ff:ff:ff:ff:ff:ff
4: enp5s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9600 qdisc mq state UP group default qlen 1000
    link/ether 40:a6:b7:30:c1:d9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.22/24 brd 192.168.4.255 scope global enp5s0f1
       valid_lft forever preferred_lft forever
    inet6 fe80::42a6:b7ff:fe30:c1d9/64 scope link
       valid_lft forever preferred_lft forever
5: wlp4s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 84:1b:77:8e:5b:52 brd ff:ff:ff:ff:ff:ff
6: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:be:06:fd:c0 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
16: vxlan.calico: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN group default
    link/ether 66:bd:52:f1:24:60 brd ff:ff:ff:ff:ff:ff
    inet 10.1.246.192/32 scope global vxlan.calico
       valid_lft forever preferred_lft forever
    inet6 fe80::64bd:52ff:fef1:2460/64 scope link
       valid_lft forever preferred_lft forever
17: cali50bfcb144d2@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UP group default qlen 1000
    link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netns cni-a9e67e39-920f-929b-09e8-596bd80d5e4e
    inet6 fe80::ecee:eeff:feee:eeee/64 scope link
       valid_lft forever preferred_lft forever
18: calic7ffc27fbaa@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UP group default qlen 1000
    link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netns cni-136c2bb4-afb9-c9be-8ad3-0541312b949a
    inet6 fe80::ecee:eeff:feee:eeee/64 scope link
       valid_lft forever preferred_lft forever
```

11.2 Configuration for updation of GRUB

The following commands given below are to be executed:

vim /etc/default/grub

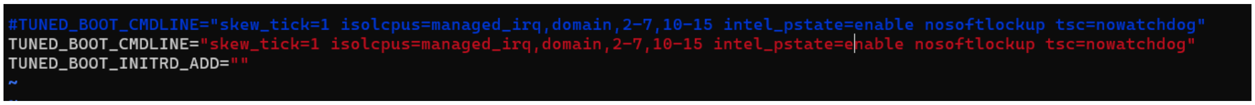


```
#### hugepages test ####
GRUB_CMDLINE_LINUX="intel_iommu=on iommu=pt mce=off mitigations=off skew_tick=1 selinux=0 enforcing=0 tsc=nowatchdog nmi_watchdog=0 softlockup
_panic=0 audit=0 rcu_nocb_poll nohz=on nohz_full=2-7,10-15 rcu_nocbs=2-7,10-15 nosoftlockup kthread_cpus=0-1,8-9 irqaffinity=0-1,8-9 default_h
ugepagesz=1GB hugepagesz=1G hugepages=10"
GRUB_CMDLINE_LINUX_DEFAULT="${GRUB_CMDLINE_LINUX_DEFAULT:+$GRUB_CMDLINE_LINUX_DEFAULT }\$tuned_params"
GRUB_INITRD_OVERLAY="${GRUB_INITRD_OVERLAY:+$GRUB_INITRD_OVERLAY }\$tuned_initrd"
```

Open respective **path-** ***/etc/default/grub*** to edit the grub boot parameters for cpu and core isolation***.***

Update GRUB using the following steps:

1.1. check and update pstate enable in **/etc/tuned/bootcmdline**



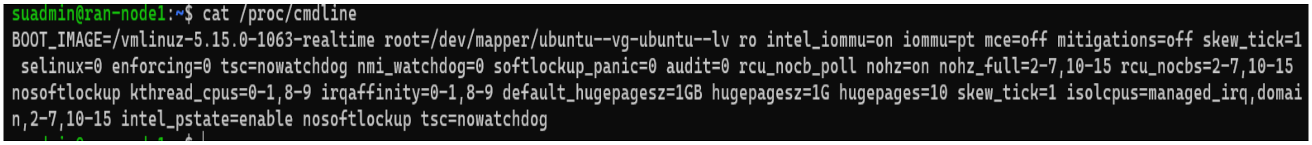
```
#TUNED_BOOT_CMDLINE="skew_tick=1 isolcpus=managed_irq,domain,2-7,10-15 intel_pstate=enable nosoftlockup tsc=nowatchdog"
TUNED_BOOT_CMDLINE="skew_tick=1 isolcpus=managed_irq,domain,2-7,10-15 intel_pstate=enable nosoftlockup tsc=nowatchdog"
TUNED_BOOT_INITRD_ADD=""
~
```

1.2. sudo update-grub

1.3. reboot

1.4. Check that the result of **cat /proc/cmdline** matches:

> BOOT_IMAGE=/vmlinuz-5.15.0-1063-realtime root=/dev/mapper/ubuntu--vg-ubuntu--lv ro intel_iommu=on iommu=pt mce=off mitigations=off skew_tick=1 selinux=0 enforcing=0 tsc=nowatchdog nmi_watchdog=0 softlockup_panic=0 audit=0 rcu_nocb_poll nohz=on nohz_full=2-7,10-15 rcu_nocbs=2-7,10-15 nosoftlockup kthread_cpus=0-1,8-9 irqaffinity=0-1,8-9 default_hugepagesz=1GB hugepagesz=1G hugepages=10 skew_tick=1 isolcpus=managed_irq,domain,2-7,10-15 intel_pstate=enable nosoftlockup tsc=nowatchdog



```
suadmin@ran-node1:~$ cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-5.15.0-1063-realtime root=/dev/mapper/ubuntu--vg-ubuntu--lv ro intel_iommu=on iommu=pt mce=off mitigations=off skew_tick=1
 selinux=0 enforcing=0 tsc=nowatchdog nmi_watchdog=0 softlockup_panic=0 audit=0 rcu_nocb_poll nohz=on nohz_full=2-7,10-15 rcu_nocbs=2-7,10-15
nosoftlockup kthread_cpus=0-1,8-9 irqaffinity=0-1,8-9 default_hugepagesz=1GB hugepagesz=1G hugepages=10 skew_tick=1 isolcpus=managed_irq,domai
n,2-7,10-15 intel_pstate=enable nosoftlockup tsc=nowatchdog
```

11.3. Ethernet Driver

Download Ethernet Driver From:
wget https://downloadmirror.intel.com/832290/i40e-2.26.8.tar.gz

To untar execute **tar -xvzf** command.

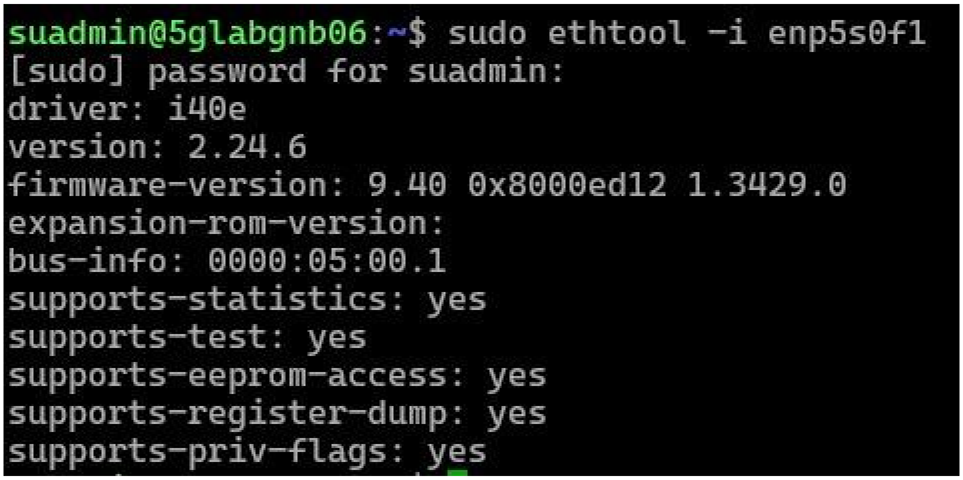
To remove the previous driver version use the **sudo rmmod i40e** command.

Change the directory to src in the driver by executing the **cd/home/suadmin/driver_x710/i40e-2.26.8/src** command.

Install the driver by the **sudo make install** command.

Insert the driver in the kernel by using **Sudo modprobe i40e** command.

Check the driver version via the **sudo ethtool -i enp5s0f1** command.



```
suadmin@5glabgnb06:~$ sudo ethtool -i enp5s0f1
[sudo] password for suadmin:
driver: i40e
version: 2.24.6
firmware-version: 9.40 0x8000ed12 1.3429.0
expansion-rom-version:
bus-info: 0000:05:00.1
supports-statistics: yes
supports-test: yes
supports-eeprom-access: yes
supports-register-dump: yes
supports-priv-flags: yes
```

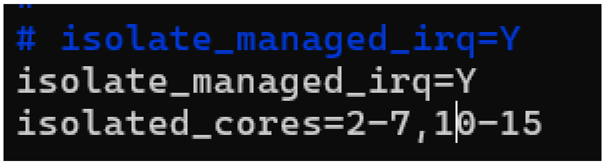
11.4. Tuned Parameters

sudo apt install tuned -y

sudo vim /etc/tuned/realtime-variables.conf

isolate_managed_irq=Y

isolated_cores=2-7,10-15'



```
# isolate_managed_irq=Y
isolate_managed_irq=Y
isolated_cores=2-7,10-15
```

sudo apt install build-essential -y

sudo tuned-adm profile realtime

11.5. Ring Parameters and CPU frequency

sudo apt install linux-tools-common -y

sudo apt install linux-tools-5.15.0-1058-realtime -y

sudo apt install linux-cloud-tools-5.15.0-1058-realtime -y

sudo timedatectl set-ntp false

sudo cpupower idle-set -D 0

```
sudo cpupower frequency-set -g performance

sudo ldconfig

sudo ip link set mtu 9000 “interface_name”

sudo ip link set mtu 9000 “interface_name”

sudo sysctl -w net.core.rmem_max=62500000

sudo sysctl -w net.core.wmem_max=62500000

sudo sysctl -w net.core.rmem_default=62500000

sudo sysctl -w net.core.wmem_default=62500000

sudo sysctl -w net.core.default_qdisc=fq

sudo sysctl -w net.ipv4.tcp_congestion_control=bbr

sudo ip route add 192.168.xx.0/24 via 192.xxx.xxx.xx

sudo iptables -P FORWARD ACCEPT

sudo sysctl net.ipv4.conf.all.forwarding=1

sudo cpupower frequency-set -g performance -u 4.70GHz -d 4.70GHz

sudo cpupower idle-set -D 0

sudo ethtool -G enp2s0f1 tx 4096 rx 4096

sudo ethtool -G “interface_name” tx 4096 rx 4096

sudo sh -c 'echo "1" > /sys/module/vfio/parameters/enable_unsafe_noiommu_mode'

sudo ethtool -G “interface_name” tx 4096 rx 4096

sudo ifconfig “interface_name” mtu 8870
```

11.6. Virtual Functions Creation

```
sudo  'echo "0" > /sys/class/net/enp2s0f1/device/sriov_numvfs'

sudo  'echo "2" > /sys/class/net/(ethernet interface)/device/sriov_numvfs'
sudo ip link set dev “interface_name” vf 0 mac 00:11:22:33:44:66 trust on vlan 100 spoofchk off mtu 8870

sleep 1

sudo ip link set dev “interface_name” vf 1 mac 00:11:22:33:44:66 trust on vlan 100 spoofchk off mtu 8870
```

11.7. Dpdk Installation

```
sudo apt install wget xz-utils libnuma-dev -y

cd /home/suadmin

wget http://fast.dpdk.org/rel/dpdk-20.11.9.tar.xz

sudo apt install meson -y

tar xvf dpdk-20.11.9.tar.xz && cd dpdk-stable-20.11.9

meson build

ninja -C build

sudo ninja install -C build

sudo rm -rf dpdk-20.11.9.tar.xz
```

11.8. Linuxptp Installation

```
sudo apt install linuxptp-y
```

11.9. Docker Installation

Open a bash shell to install docker and related tools

```
a) sudo apt install -y git net-tools putty

b) sudo apt install -y apt-transport-https ca-certificates curl software-properties-common

c) curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs)  stable"

d) sudo apt update
```

e) sudo apt install -y docker docker-ce

f) curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o/usr/share/keyrings/docker-archive-keyring.gpg echo “deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs)stable” | sudo tee /etc/apt/sources.list.d/docker.list >/dev/null

g) sudo apt-get update

h) # Add your username to the docker group, otherwise you will have to run in sudo mode.

- sudo usermod -a -G docker $(whoami)

i) reboot

disable ptp if required to install docker:

Sudo systemctl stop ptp4l

Sudo systemctl stop phc2sys

Sudo timedatectl set-ntp true

hwclock -w

Restart ptp after docker installation

11.10. Create services for ptp4l and phc2sys

Create a service file with below contents in the following path

Service file path /usr/lib/systemd/system/ptp4l.service

```
[Unit]

Description=Precision Time Protocol (PTP) service

After=network-online.target

Wants=network-online.target
```

```
[Service]

Type=simple

EnvironmentFile=-/etc/sysconfig/ptp4l

ExecStart=/usr/bin/taskset -c 2 /usr/sbin/ptp4l $OPTIONS

Slice=ran.slice


[Install]

WantedBy=multi-user.target
```

Service file path /usr/lib/systemd/system/phc2sys.service

```
 [Unit]
 Description=Synchronize system clock or PTP hardware clock (PHC)
 After=ntpdate.service ptp4l.service

[Service]
 Type=simple
 EnvironmentFile=-/etc/sysconfig/phc2sys
 ExecStart=/usr/bin/taskset -c 2 /usr/sbin/phc2sys $OPTIONS
 Slice=ran.slice

[Install]
 WantedBy=multi-user.target
```

For enabling these services on startup execute following commands

```
sudo systemctl daemon-reload
sudo systemctl enable ptp4l.service
sudo systemctl enable phc2sys.service
sudo systemctl start ptp4l.service
sudo systemctl start phc2sys.service
```

11.11. Configuration for Images

Login into docker

Follow the following steps:

1. sudo systemctl set-property ran.slice AllowedCPUs=2-7,10-15

2. Edit the file /usr/lib/systemd/system/docker.service and add

Slice=ran.slice

3. sudo docker network create --driver bridge --subnet=192.168.80.0/24 w4s-network

4. Make the config file in /home/suadmin/T.2x2.oran.fhi.7.2.conf with the content as mentioned at step 5.

The bus id can be found out via the command:

```
sudo lshw -c network -businfo
```

check vf device address and update line:

```
dpdk_devices = ("vf1 busid", "vf2 busid");
```

Login to RU check RU MAC and update:

```
ru_addr = ("ru mac for eth1", "ru mac for eth1");
```

5. docker run -d --net w4s-network --ip 192.168.80.10 -v /home/suadmin/T.2x2.oran.fhi.7.2.conf:/opt/oai-gnb/etc/gnb.conf -v /dev/hugepages:/dev/hugepages -v /tmp:/tmp --cgroup-parent ran.slice --cap-add SYS_ADMIN --cap-add IPC_LOCK --cap-add SYS_NICE --env USE_ADDITIONAL_OPTIONS="--sa --thread-pool 3,4,5,6,7" --device /dev/vfio:/dev/vfio/ --env TZ=Asia/Kolkata --name w4s-gnb w4slabs/w4s-gnb-fhi-e:2024-w33 taskset -c 10 /opt/oai-gnb/bin/nr-softmodem -O /opt/oai-gnb/etc/gnb.conf

6. Check that w4s-gnb container is up and healthy

7. Stop w4s container

8. Login to RU

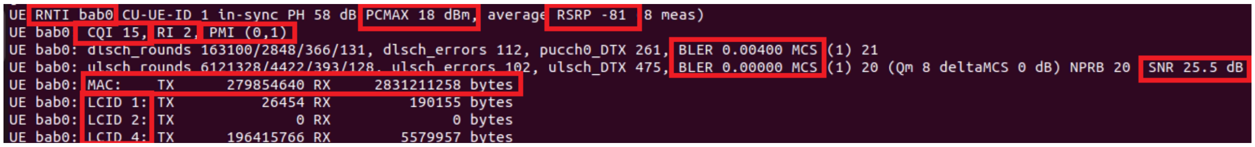
9. Start w4s-gnb again.

NOTE: Please refer to RU Overview for configuration details.

11.11.1 Check gNB Process Status

docker ps

11.12. gNB Logs Parameters



```
UE RNTI bab0 CU-UE-ID 1 in-sync PH 58 dB PCMAX 18 dBm, average RSRP -81  8 meas)
UE bab0  CQI 15, RI 2, PMI (0,1)
UE bab0: dlsch_rounds 163100/2848/366/131, dlsch_errors 112, pucch0_DTX 261, BLER 0.00400 MCS (1) 21
UE bab0: ulsch_rounds 6121328/4422/393/128, ulsch_errors 102, ulsch_DTX 475, BLER 0.00000 MCS (1) 20 (Qm 8 deltaMCS 0 dB) NPRB 20  SNR 25.5 dB
UE bab0: MAC:    TX      279854640 RX     2831211258 bytes
UE bab0: LCID 1: TX          26454 RX         190155 bytes
UE bab0: LCID 2: TX              0 RX              0 bytes
UE bab0: LCID 4: TX      196415766 RX        5579957 bytes
```

UE RNTI 1ef1 CU-UE-ID 1 in-sync PH 44 dB PCMAX 15 dBm, average **RSRP** -89 (8 meas)

UE 1ef1: CQI 15, RI 2, PMI (0,1)

UE 1ef1: dlsch_rounds 314/2/0/0, dlsch_errors 0, pucch0_DTX 0, BLER 0.02407 MCS (1) 26

UE 1ef1: ulsch_rounds 1981/38/1/0, ulsch_errors 0, ulsch_DTX 5, BLER 0.04202 MCS (1) 27 (Qm 8 deltaMCS 0 dB) NPRB 5 SNR 26.5 dB

```
UE 1ef1: MAC:          TX        318488 RX              290851 bytes

UE 1ef1: LCID 1: TX            631 RX                     836 bytes

UE 1ef1: LCID 2: TX            0 RX                         0 bytes

UE 1ef1: LCID 4: TX            136412 RX              76742 bytes
```

11.12.1.gNB Log File Parameters Overview

11.12.1.1. Block Error Rate Overview

The Block Error Rate (BLER) is a crucial performance metric used to evaluate the reliability of data transmission. It measures the efficiency of data transmission over the air interface and denotes the percentage of transmitted data blocks (or packets) that are received with errors and cannot be successfully decoded.

A high BLER indicates poor link quality, necessitating retransmissions via Hybrid Automatic Repeat Request (HARQ), which can increase latency and decrease throughput. Whereas, a low BLER implies efficient communication but may lead to underutilization of resources if achieved at the cost of overly conservative link adaptation.

The factors that influence BLER are the Signal-to-Noise Ratio (SNR), channel condition, Modulation and Coding Scheme (MCS), and Resource Management.

BLER (Block Error Rate) is a crucial metric in 5G networks that maintains robust and efficient communication by balancing reliability with resource utilization. By monitoring and optimizing BLER, 5G networks can provide the different QoS levels needed for various applications, ensuring a seamless and dependable user experience.

11.12.1.2. Radio Network Temporary Identifier Overview

The Radio Network Temporary Identifier (RNTI) is an essential component that enables efficient communication between UEs and the gNB. RNTIs uniquely identify UEs, manage signaling, allocate resources, and maintain secure communications. They play a crucial role in ensuring optimized spectrum usage, reducing interference, and protecting user privacy by using temporary identifiers rather than permanent ones, mitigating risks of tracking or unauthorized interception.

RNTI management in 5G is dynamic and adaptable, supporting the network's need to manage high-speed, low-latency communication. UEs may be assigned to multiple RNTIs simultaneously, each mapped to a particular role, such as dedicated communication or system information reception.

11.12.1.3. Reference Signal Received Power Overview

Reference Signal Received Power (RSRP) is a crucial metric used in 5G networks to evaluate the signal strength and quality of a specific cell. It measures the average power of the reference signals transmitted by a base station (gNB) over a given resource element.

RSRP value serves as a critical parameter for network performance assessment, handover decisions, and optimization processes, directly influencing the user experience in terms of connectivity and data throughput. RSRP measurement ensures robust connectivity by enabling the selection of the most suitable base station for a user's device and enable intelligent resource management and beamforming techniques.

RSRP plays an essential role in load balancing and interference coordination while ensuring efficient utilization of network resources and enhancing overall system capacity. Moreover, RSRP values are crucial for network operators to optimize coverage planning, assess the effectiveness of signal propagation, and fine-tune gNB deployments.

11.12.1.4. Modulation and Coding Scheme Overview

The Modulation and Coding Scheme (MCS) enables dynamic adjustment of data transmission rates to ensure efficient use of radio resources and maintain robust communication under varying channel conditions. The MCS defines how data is modulated for transmission and the level of error-correcting codes applied, balancing trade-offs between data rate, reliability, and signal quality.

The MCS has 2 Key concepts:

Modulation: defines how many bits can be carried by a single Resource Element (**RE)** irrespective of whether it’s a useful bit or parity bits. The supported modulation is QPSK, 16 QAM, 64 QAM, and 256 QAM.

Code Rate: the ratio between useful bit and total transmitted bit (Useful + Redundant Bits). These Redundant bits are added for Forward Error Correction (FEC). In other words, the ratio between the number of information bits at the top of the Physical layer and the number of bits that are mapped to PDSCH at the bottom of the Physical layer.

11.12.1.5. Signal-to-Noise Ratio Overview

Signal-to-Noise Ratio (SNR) measures the power ratio of a signal, representing useful information, to the power of background noise, which is unwanted interference. This ratio is usually expressed in decibels (dB).

High SNR enables the use of higher-order modulation schemes. It also ensures consistent connectivity and minimizes errors in signal transmission, crucial for mission-critical applications**.**

11.12.1.6. Logical Channel Identity Overview

Logical Channel Identity (LCID) is a unique identifier that distinguishes different logical channels within a single User Equipment (UE). Each logical channel corresponds to a specific type of traffic, such as signaling or user data, and is associated with a Quality of Service (QoS) requirement.

LAC ID 1 typically represents a high-priority or densely populated area within the network, such as urban centers, business districts, or areas with heavy user traffic. These locations require optimized mobility management due to frequent handovers and high network congestion.

LAC ID 2 represents suburban or less congested areas where mobile traffic is moderate compared to primary urban zones. These areas may have different location update policies, allowing for slightly longer update intervals to balance efficiency and signaling overhead.

LAC ID 4 is generally associated with larger geographic areas that cover rural, remote, or low-density regions. In these areas, network signaling efficiency is prioritized over frequent updates. The network may configure LAC ID 4 with extended periodic location update timers to reduce unnecessary signaling and optimize resource usage while ensuring continued service availability.

11.12.1.7. Channel Quality Indicator Overview

Channel Quality Indicator (CQI) is a key metric in wireless communication that represents the quality of the radio channel as perceived by the UE. It is reported to the network to facilitate adaptive modulation and coding (AMC), enabling dynamic adjustments in transmission parameters to optimize throughput and maintain reliable connectivity.

11.12.1.8. Maximum Transmission Power Overview

Maximum Transmission Power (Pcmax) refers to the highest power level at which a device, such as a UE or a base station, is allowed to transmit. This parameter is crucial for managing network interference, conserving battery life, and ensuring compliance with regulatory power limits. Pcmax is determined by network configurations, device capabilities, and environmental conditions, and it plays a significant role in balancing coverage and capacity within a cellular network.

11.12.1.9. Precoding Matrix Indicator Overview

The Precoding Matrix Indicator (PMI) is a feedback parameter used in Multiple Input Multiple Output (MIMO) systems to enhance spectral efficiency. PMI provides information about the preferred precoding matrix, which is used to optimize the spatial transmission of signals.

11.12.1.10. Rank Indicator Overview

Rank Indicator (RI) signifies the number of spatial layers that can be effectively used for data transmission in MIMO systems. It is reported by the UE to the network, helping determine whether a single-stream (Rank 1) or multi-stream (higher rank) transmission should be used. A higher RI generally indicates that multiple independent data streams can be transmitted simultaneously, improving spectral efficiency and overall system capacity.

11.12.1.11. Medium Access Control Overview

Medium Access Control (MAC) layer is a fundamental component of the wireless protocol stack that governs how network resources are allocated and how data packets are transmitted between devices. It manages scheduling, prioritization, and error correction mechanisms, ensuring efficient communication and minimal latency. The MAC layer operates between the physical layer and higher-layer protocols.

11.13.   Configuration for Network slicing

11.13.1.Configuration for Network Slice 1

Below are the commands for the network slice 1 configuration:

```
# Log level for all the NFs
log_level:
  general: debug
# If you enable registration, the other NFs will use the NRF discovery mechanism
register_nf:
  general: yes
http_version: 2
############## SBI Interfaces
nfs:
  smf:
          host: w4s-smf-slice1
          sbi:
          port: 8080
          api_version: v1
          interface_name: eth0
          n4:
          interface_name: eth0
          port: 8805
  nrf:
          host: w4s-nrf-slice1
          sbi:
          port: 8080
          api_version: v1
```

```
        interface_name: eth0
  upf:
        host: w4s-upf-slice1
        sbi:
        port: 8080
        api_version: v1
        interface_name: eth0
        n3:
        interface_name: eth0
        port: 2152
        n4:
        interface_name: eth0
        port: 8805
        n6:
        interface_name: eth0
        n9:
        interface_name: eth0
        port: 2152

# anchor is set to re-use slice config in SMF
snssais:
  - &slice1
        sst: 128
```

```
          sd: 000080 # in hex


smf:
  support_features:
    use_local_subscription_info: yes # Use infos from local_subscription_info or from UDM
    use_local_pcc_rules: yes # Use infos from local_pcc_rules or from PCF
  ue_dns:
          primary_ipv4: "172.21.3.100"
          secondary_ipv4: "8.8.8.8"
  # the DNN you configure here should be configured in "dnns"
  # follows the SmfInfo datatype from 3GPP TS 29.510
  smf_info:
    sNssaiSmfInfoList:
          - sNssai: *slice1
        dnnSmfInfoList:
          - dnn: "5glabs1"
  local_subscription_infos:
          - single_nssai: *slice1
          dnn: "5glabs1"
          qos_profile:
          5qi: 5
        session_ambr_ul: "50Mbps"
        session_ambr_dl: "100Mbps"
```

```
upf:
  support_features:
    enable_bpf_datapath: no          # If "on": BPF is used as datapath else simpleswitch is used, DEFAULT= off
           enable_snat: yes          # If "on": Source natting is done for UE, DEFAULT= off
  remote_n6_gw: localhost            # Dummy host since simple-switch does not use N6 GW
  upf_info:
    sNssaiUpfInfoList:
           - sNssai: *slice1
        dnnUpfInfoList:
           - dnn: "5glabs1"


## DNN configuration
dnns:
  - dnn: "5glabs1"
           pdu_session_type: "IPV4"
           ipv4_subnet: "12.0.0.0/24"
```

11.13.2. Configuration for Network Slice 2

Below are the parameters for the network slice 2 configuration:

```
 # Log level for all the NFs

log_level:

  general: debug


# If you enable registration, the other NFs will use the NRF discovery mechanism

register_nf:

  general: yes


http_version: 2


############## SBI Interfaces

nfs:

  smf:

           host: w4s-smf-slice2

           sbi:

           port: 8080

           api_version: v1

           interface_name: eth0

           n4:

           interface_name: eth0

           port: 8805

  nrf:

           host: w4s-nrf-slice2
```

```
          sbi:
          port: 8080
          api_version: v1
          interface_name: eth0
  upf:
          host: w4s-upf
          sbi:
          port: 8080
          api_version: v1
          interface_name: eth0
          n3:
          interface_name: eth0
          port: 2152
          n4:
          interface_name: eth0
          port: 8805
          n6:
          interface_name: eth0
          n9:
          interface_name: eth0
          port: 2152

# anchor is set to re-use slice config in SMF
```

```
snssais:
  - &slice2
          sst: 1
          sd: 000001 # in hex

smf:
  support_features:
          use_local_subscription_info: yes # Use infos from local_subscription_info or from UDM
    use_local_pcc_rules: yes # Use infos from local_pcc_rules or from PCF
  ue_dns:
          primary_ipv4: "172.21.3.100"
          secondary_ipv4: "8.8.8.8"
  # the DNN you configure here should be configured in "dnns"
  # follows the SmfInfo datatype from 3GPP TS 29.510
  smf_info:
    sNssaiSmfInfoList:
          - sNssai: *slice2
        dnnSmfInfoList:
          - dnn: "5glabs2"
  local_subscription_infos:
          - single_nssai: *slice2
          dnn: "5glabs2"
          qos_profile:
```

```
          5qi: 5
        session_ambr_ul: "20Mbps"
        session_ambr_dl: "400Mbps"

upf:
  support_features:
    enable_bpf_datapath: no         # If "on": BPF is used as datapath else simpleswitch is used, DEFAULT= off
          enable_snat: yes          # If "on": Source natting is done for UE, DEFAULT= off
  remote_n6_gw: localhost           # Dummy host since simple-switch does not use N6 GW
  upf_info:
    sNssaiUpfInfoList:
          - sNssai: *slice2
        dnnUpfInfoList:
          - dnn: "5glabs2"

## DNN configuration
dnns:
  - dnn: "5glabs2"
          pdu_session_type: "IPV4"
          ipv4_subnet: "13.0.0.0/24"
```

11.13.3.Configuration for Network Slice 3

Below are the parameters for network Slice 3 configuration:

```
# Log level for all the NFs
log_level:
  general: debug


# If you enable registration, the other NFs will use the NRF discovery mechanism
register_nf:
  general: yes


http_version: 2


############## SBI Interfaces
nfs:
  smf:
          host: w4s-smf
          sbi:
          port: 8080
          api_version: v1
          interface_name: eth0
          n4:
          interface_name: eth0
          port: 8805
  pcf:
          host: w4s-pcf
          sbi:
          port: 8080
          api_version: v1
          interface_name: eth0
  nrf:
          host: w4s-nrf
          sbi:
          port: 8080
          api_version: v1
          interface_name: eth0

# anchor is set to re-use slice config in SMF

snssais:
  - &slice3
          sst: 130
          sd: 000082 # in hex

smf:
  ue_mtu: 1500
  support_features:
   use_local_subscription_info: yes # Use infos from local_subscription_info or from UDM

   use_local_pcc_rules: no # Use infos from local_pcc_rules or from PCF

  # this is an example how you can enable/disable UPF features while the profile comes from NRF, host has to match with NRF profile

  upfs:

          - host: ulcl.node.5gcn.mnc01.mcc001.5glabindia.in

          config:

       enable_usage_reporting: yes

          - host: aupf1.node.5gcn.mnc01.mcc001.5glabindia.in

          config:

       enable_usage_reporting: no

          - host: aupf2.node.5gcn.mnc01.mcc001.5glabindia.in

          config:

       enable_usage_reporting: no

  ue_dns:

          primary_ipv4: "172.21.3.100"

          secondary_ipv4: "8.8.8.8"

#  ims:

#          pcscf_ipv4: "127.0.0.1"

  # the DNN you configure here should be configured in "dnns"

  # follows the SmfInfo datatype from 3GPP TS 29.510

  smf_info:

   sNssaiSmfInfoList:
          - sNssai: *slice3
        dnnSmfInfoList:
          - dnn: "5glab"
  local_subscription_infos:
          - single_nssai: *slice3
          dnn: "5glab"
          qos_profile:
          5qi: 9

## DNN configuration
dnns:
  - dnn: "5glab"
          pdu_session_type: "IPV4"
#         ipv4_subnet: "14.0.0.0/24"
          ipv4_subnet: "12.1.1.0/25"

#nssf:
#  slice_config_path: /openair-nssf/etc/nssf_slice_config.yaml

pcf:
  local_policy:
    policy_decisions_path: /openair-pcf/policies/policy_decisions
          pcc_rules_path: /openair-pcf/policies/pcc_rules
```

```
traffic_rules_path: /openair-pcf/policies/traffic_rules
```

11.14. Configuration for 1x1

```
Active_gNBs = ( "gNB-OAI");

# Asn1_verbosity, choice in: none, info, annoying

Asn1_verbosity = "none";


gNBs =

(

 {

   ////////// Identification parameters:

   gNB_ID    =  0xe00;

   gNB_name  =  "gNB-OAI";


   // Tracking area code, 0x0000 and 0xfffe are reserved values

   tracking_area_code  = 0xa000;

# plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ( { sst = 1; }); });

# plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 130 ,sd = 130 }) });

# plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 128 ,sd = 128 }) });

   plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 1 ,sd = 1 }) });


   //nr_cellid = 12345678L;
```

```
    nr_cellid = 1;

    ////////// Physical parameters:

    pdsch_AntennaPorts_XP = 1;
  # pdsch_AntennaPorts_N1 = 1;
    maxMIMO_layers     = 1;
    pusch_AntennaPorts          = 1;
    do_CSIRS                 = 1;
    do_SRS                   = 0 ;
    sib1_tda                 = 15;

    enable_sdap = 1;                 # SDAP configuration test
    pdcch_ConfigSIB1 = (
    {
            controlResourceSetZero = 11;
            searchSpaceZero = 0;
    }
    );

    servingCellConfigCommon = (
    {
   #spCellConfigCommon
  physCellId                                = 0;

  # n_TimingAdvanceOffset                              = 0;

#  downlinkConfigCommon

  #frequencyInfoDL

  # this is 3450.72 MHz (center frequency)

  absoluteFrequencySSB                                 = 640704;

  dl_frequencyBand                                = 78;

  # this is 3401.58 MHz

  dl_absoluteFrequencyPointA                           = 640128;

  #scs-SpecificCarrierList

         dl_offstToCarrier                             = 0;

# subcarrierSpacing

# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120

         dl_subcarrierSpacing                          = 1;

         dl_carrierBandwidth                           = 273;

  #initialDownlinkBWP

  #genericParameters

  initialDLBWPlocationAndBandwidth                     = 1099; #38.101-1 Table
5.3.2-1

  #

# subcarrierSpacing

# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120

         initialDLBWPsubcarrierSpacing                        = 1;
  #pdcch-ConfigCommon
        initialDLBWPcontrolResourceSetZero                = 11;
        initialDLBWPsearchSpaceZero                       = 0;

 #uplinkConfigCommon
  #frequencyInfoUL
  ul_frequencyBand                             = 78;
  #scs-SpecificCarrierList
  ul_offstToCarrier                            = 0;
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
  ul_subcarrierSpacing                         = 1;
  ul_carrierBandwidth                          = 273;
  pMax                                    = 23;
  #initialUplinkBWP
  #genericParameters
        initialULBWPlocationAndBandwidth                  = 1099;
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
        initialULBWPsubcarrierSpacing                     = 1;
  #rach-ConfigCommon
        #rach-ConfigGeneric
        prach_ConfigurationIndex                     = 159;
```

```
#prach_msg1_FDM
#0 = one, 1=two, 2=four, 3=eight
        prach_msg1_FDM                              = 0;
        prach_msg1_FrequencyStart                   = 0;
        zeroCorrelationZoneConfig                   = 15;
        preambleReceivedTargetPower                      = -90;
#preamblTransMax (0...10) = (3,4,5,6,7,8,10,20,50,100,200)
        preambleTransMax                            = 6;
#powerRampingStep
# 0=dB0,1=dB2,2=dB4,3=dB6
        powerRampingStep                            = 1;
#ra_ReponseWindow
#1,2,4,8,10,20,40,80
        ra_ResponseWindow                           = 5;
#ssb_perRACH_OccasionAndCB_PreamblesPerSSB_PR
#1=oneeighth,2=onefourth,3=half,4=one,5=two,6=four,7=eight,8=sixteen
    ssb_perRACH_OccasionAndCB_PreamblesPerSSB_PR          = 3;
#oneHalf (0..15) 4,8,12,16,...60,64
    ssb_perRACH_OccasionAndCB_PreamblesPerSSB             = 15;
#ra_ContentionResolutionTimer
#(0..7) 8,16,24,32,40,48,56,64
        ra_ContentionResolutionTimer                     = 7;
        rsrp_ThresholdSSB                           = 19;
```

```
#prach-RootSequenceIndex_PR

#1 = 839, 2 = 139

        prach_RootSequenceIndex_PR                  = 2;

        prach_RootSequenceIndex                     = 1;

        # SCS for msg1, can only be 15 for 30 kHz < 6 GHz, takes precendence over the one derived from prach-ConfigIndex

        #

        msg1_SubcarrierSpacing                  = 1,

# restrictedSetConfig

# 0=unrestricted, 1=restricted type A, 2=restricted type B

        restrictedSetConfig                     = 0,


# this is the offset between the last PRACH preamble power and the Msg3 PUSCH, 2 times the field value in dB

        msg3_DeltaPreamble                      = 6;

        p0_NominalWithGrant                     = -90;


# pucch-ConfigCommon setup :

# pucchGroupHopping

# 0 = neither, 1= group hopping, 2=sequence hopping

        pucchGroupHopping                       = 0;

        hoppingId                           = 0;

        p0_nominal                           = -70;

# ssb_PositionsInBurs_BitmapPR
# 1=short, 2=medium, 3=long
  ssb_PositionsInBurst_PR                           = 2;
  ssb_PositionsInBurst_Bitmap                       = 0x1;

# ssb_periodicityServingCell
# 0 = ms5, 1=ms10, 2=ms20, 3=ms40, 4=ms80, 5=ms160, 6=spare2, 7=spare1
  ssb_periodicityServingCell                 = 2;

# dmrs_TypeA_position
# 0 = pos2, 1 = pos3
  dmrs_TypeA_Position                        = 0;

# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
  subcarrierSpacing                          = 1;


 #tdd-UL-DL-ConfigurationCommon
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
  referenceSubcarrierSpacing                        = 1;
  # pattern1
  # dl_UL_TransmissionPeriodicity
```

```
   # 0=ms0p5, 1=ms0p625, 2=ms1, 3=ms1p25, 4=ms2, 5=ms2p5, 6=ms5, 7=ms10

   dl_UL_TransmissionPeriodicity                               = 5;

   nrofDownlinkSlots                                   = 3;

   nrofDownlinkSymbols                                 = 6;

   nrofUplinkSlots                                   = 1;

   nrofUplinkSymbols                                   = 4;


  ssPBCH_BlockPower                                         = -25;#0;

  }


  );



   # ------- SCTP definitions

   SCTP :

   {

         # Number of streams to use in input/output

         SCTP_INSTREAMS  = 2;

         SCTP_OUTSTREAMS = 2;

   };



# ////////// AMF parameters:
```

```
# amf_ip_address = ({ ipv4 = "172.21.6.5"; });

#

# NETWORK_INTERFACES :

# {

#         GNB_IPV4_ADDRESS_FOR_NG_AMF          = "172.21.18.20/22";

#         GNB_IPV4_ADDRESS_FOR_NGU             = "172.21.18.20/22";

#         GNB_PORT_FOR_S1U                     = 2152; # Spec 2152

#   };

          ////////// AMF parameters:

          amf_ip_address      = ( { ipv4       = "192.168.70.132";

                        ipv6    = "192:168:30::17";

                        active  = "yes";

                        preference = "ipv4";

                        }

                   );


          NETWORK_INTERFACES :

          {

#         GNB_INTERFACE_NAME_FOR_NG_AMF              = "enp3s0";

          GNB_IPV4_ADDRESS_FOR_NG_AMF          = "192.168.80.10";

#         GNB_INTERFACE_NAME_FOR_NGU           = "enp3s0";

          GNB_IPV4_ADDRESS_FOR_NGU             = "192.168.80.10";

          GNB_PORT_FOR_S1U                     = 2152; # Spec 2152
```

```
              };
  }
);

#e2_agent = {
##  near_ric_ip_addr = "192.168.71.164";
#  near_ric_ip_addr = "192.168.70.164";
#  sm_dir = "/usr/local/lib/flexric/"
#}

MACRLCs = (
{
  num_cc                      = 1;
  tr_s_preference                   = "local_L1";
  tr_n_preference                   = "local_RRC";
  pusch_TargetSNRx10          = 250;
  pucch_TargetSNRx10          = 200;
 # dl_bler_target_upper=.35;
 # dl_bler_target_lower=.15;
#  ul_bler_target_upper=.35;
#  ul_bler_target_lower=.15;
  pusch_FailureThres            = 100;
  ulsch_max_frame_inactivity  = 0;
  ul_max_mcs                    = 20;
  dl_max_mcs                    = 26;
  min_grant_prb                 = 20;
  min_grant_mcs                 = 9;
}
);

L1s = (
{
  num_cc = 1;
  tr_n_preference = "local_mac";
  prach_dtx_threshold = 130;
  pucch0_dtx_threshold = 80;
  pusch_dtx_threshold = 20;#-100;
#thread_pool_size = 8;
  tx_amp_backoff_dB = 10;
  L1_rx_thread_core = 11;
  L1_tx_thread_core = 12;
  phase_compensation = 0; # needs to match O-RU configuration
}
);

RUs = (
{
  local_rf       = "no";
  nb_tx          = 1;
  nb_rx          = 1;
  att_tx         = 0
  att_rx         = 0;
  bands          = [78];
  max_pdschReferenceSignalPower = -27;
  max_rxgain                    = 75;
  sf_extension                  = 0;
  eNB_instances  = [0];
  ru_thread_core = 14;
  sl_ahead       = 5;
  ##beamforming 1x2 matrix: 1 layer x 2 antennas
  bf_weights = [0x00007fff, 0x0000,0x00007fff, 0x0000];
  clock_src = "internal";
  tr_preference  = "raw_if4p5"; # important: activate FHI7.2
  do_precoding = 0; # needs to match O-RU configuration
}
);

rfsimulator :
{
  serveraddr = "server";
  serverport = "4043";
  options = (); #("saviq"); or/and "chanmod"
  modelname = "AWGN";
  IQfile = "/tmp/rfsimulator.iqs";
};

security = {
  # preferred ciphering algorithms
  # the first one of the list that an UE supports in chosen
  # valid values: nea0, nea1, nea2, nea3
  ciphering_algorithms = ( "nea0" );

  # preferred integrity algorithms
  # the first one of the list that an UE supports in chosen
  # valid values: nia0, nia1, nia2, nia3
  integrity_algorithms = ( "nia2", "nia0" );

  # setting 'drb_ciphering' to "no" disables ciphering for DRBs, no matter
  # what 'ciphering_algorithms' configures; same thing for 'drb_integrity'
  drb_ciphering = "yes";
  drb_integrity = "no";
};
```

```
log_config : {
  global_log_level = "info";
  hw_log_level  = "info";
  phy_log_level = "info";
  mac_log_level          = "info";
  rlc_log_level   = "info";
  pdcp_log_level   = "info";
  rrc_log_level   = "info";
  ngap_log_level   = "info";
  f1ap_log_level   = "info";
  sdap_log_level   = "info";
};


fhi_72 = {
  dpdk_devices = ("0000:05:0a.0", "0000:05:0a.1");
  system_core = 3;
  io_core = 15;
  worker_cores = (13);
  du_addr = ("00:11:22:33:44:66", "00:11:22:33:44:66");
  ru_addr = ("98:ae:71:01:91:6b", "98:ae:71:01:91:6b");   #check and update from 192.168.4.50 (LPRU)
  mtu = 9216; # check if xran uses this properly
  file_prefix = "fhi_72";
  fh_config = ({
   Tadv_cp_dl = 125;
   T2a_cp_dl = (285, 429); # (min, max)
   T2a_cp_ul = (285, 429); # (min, max)
   T2a_up = (125, 428); # (min, max)
   Ta3 = (130, 170); # (min, max)
   T1a_cp_dl = (285, 470); # (min, max)
   T1a_cp_ul = (285, 429); # (min, max)
   T1a_up = (125, 350); # (min, max)
   Ta4 = (110, 180); # (min, max)
   ru_config = {
   iq_width = 9;
   iq_width_prach = 9;
#          fft_size = 12;
   };
   prach_config = {
   eAxC_offset = 2;     #change for MIMO and SISO switching
   };
  });
};
```

11.15. Configuration for 2x2

```
Active_gNBs = ( "gNB-OAI");
```

```
  # Asn1_verbosity, choice in: none, info, annoying
Asn1_verbosity = "none";


gNBs =
(
 {
    ////////// Identification parameters:
    gNB_ID       =  0xe00;
    gNB_name  =  "gNB-OAI";


    // Tracking area code, 0x0000 and 0xfffe are reserved values
    tracking_area_code  = 0xa000;
#    plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ( { sst = 1; }); });
#    plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 130 ,sd = 130 }) });
#    plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 128 ,sd = 128 }) });
    plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 1 ,sd = 1 }) });


    //nr_cellid = 12345678L;
    nr_cellid = 1;


    ////////// Physical parameters:


    pdsch_AntennaPorts_XP = 2;   # change this to switch from SISO to MIMO
#    pdsch_AntennaPorts_N1 = 2;  # ? #
```

```
   maxMIMO_layers              = 2;  # change this to switch from SISO to MIMO
   pusch_AntennaPorts          = 2;  # change this to switch from SISO to MIMO
   do_CSIRS              = 1;
   do_SRS                = 0 ;
   sib1_tda                    = 15;

   enable_sdap = 1;            # SDAP configuration test
   pdcch_ConfigSIB1 = (
     {
       controlResourceSetZero = 11;
       searchSpaceZero = 0;
     }
   );

   servingCellConfigCommon = (
         {
 #spCellConfigCommon

     physCellId                                        = 0;
    # n_TimingAdvanceOffset                                        = 0;
#  downlinkConfigCommon
   #frequencyInfoDL
     # this is 3450.72 MHz (center frequency)
     absoluteFrequencySSB                          = 640704;
     dl_frequencyBand                           = 78;
     # this is 3401.58 MHz
     dl_absoluteFrequencyPointA                           = 640128;
     #scs-SpecificCarrierList
       dl_offstToCarrier                        = 0;
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
       dl_subcarrierSpacing                     = 1;
       dl_carrierBandwidth                      = 273;
    #initialDownlinkBWP
     #genericParameters
      initialDLBWPlocationAndBandwidth                    = 1099; #38.101-1 Table 5.3.2-1
      #
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
       initialDLBWPsubcarrierSpacing                      = 1;
     #pdcch-ConfigCommon
       initialDLBWPcontrolResourceSetZero                 = 11;
       initialDLBWPsearchSpaceZero                        = 0;

  #uplinkConfigCommon
    #frequencyInfoUL
     ul_frequencyBand                           = 78;
     #scs-SpecificCarrierList
    ul_offstToCarrier                                = 0;
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    ul_subcarrierSpacing                             = 1;
    ul_carrierBandwidth                              = 273;
    pMax                                       = 23;
   #initialUplinkBWP
    #genericParameters
     initialULBWPlocationAndBandwidth                        = 1099;
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
     initialULBWPsubcarrierSpacing                           = 1;
    #rach-ConfigCommon
     #rach-ConfigGeneric
      prach_ConfigurationIndex                       = 159;
#prach_msg1_FDM
#0 = one, 1=two, 2=four, 3=eight
      prach_msg1_FDM                                 = 0;
      prach_msg1_FrequencyStart                              = 0;
      zeroCorrelationZoneConfig                      = 15;
      preambleReceivedTargetPower                            = -90;
#preamblTransMax (0...10) = (3,4,5,6,7,8,10,20,50,100,200)
      preambleTransMax                               = 6;
#powerRampingStep
```

```
# 0=dB0,1=dB2,2=dB4,3=dB6
        powerRampingStep                                  = 1;
#ra_ReponseWindow
#1,2,4,8,10,20,40,80
        ra_ResponseWindow                                 = 5;
#ssb_perRACH_OccasionAndCB_PreamblesPerSSB_PR
#1=oneeighth,2=onefourth,3=half,4=one,5=two,6=four,7=eight,8=sixteen
        ssb_perRACH_OccasionAndCB_PreamblesPerSSB_PR                      = 3;
#oneHalf (0..15) 4,8,12,16,...60,64
        ssb_perRACH_OccasionAndCB_PreamblesPerSSB                         = 15;
#ra_ContentionResolutionTimer
#(0..7) 8,16,24,32,40,48,56,64
        ra_ContentionResolutionTimer                            = 7;
        rsrp_ThresholdSSB                                 = 19;
#prach-RootSequenceIndex_PR
#1 = 839, 2 = 139
        prach_RootSequenceIndex_PR                              = 2;
        prach_RootSequenceIndex                                 = 1;
        # SCS for msg1, can only be 15 for 30 kHz < 6 GHz, takes precendence over the one derived from prach-ConfigIndex
        #
        msg1_SubcarrierSpacing                            = 1,
# restrictedSetConfig
# 0=unrestricted, 1=restricted type A, 2=restricted type B
        restrictedSetConfig                               = 0,
# this is the offset between the last PRACH preamble power and the Msg3 PUSCH, 2 times the
field value in dB

    msg3_DeltaPreamble                          = 6;

    p0_NominalWithGrant                         = -90;


# pucch-ConfigCommon setup :

# pucchGroupHopping

# 0 = neither, 1= group hopping, 2=sequence hopping

    pucchGroupHopping                           = 0;

    hoppingId                               = 0;

    p0_nominal                               = -70;

# ssb_PositionsInBurs_BitmapPR

# 1=short, 2=medium, 3=long

   ssb_PositionsInBurst_PR                      = 2;

   ssb_PositionsInBurst_Bitmap                        = 0x1;


# ssb_periodicityServingCell

# 0 = ms5, 1=ms10, 2=ms20, 3=ms40, 4=ms80, 5=ms160, 6=spare2, 7=spare1

   ssb_periodicityServingCell                   = 2;


# dmrs_TypeA_position

# 0 = pos2, 1 = pos3

   dmrs_TypeA_Position                          = 0;
```

```
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    subcarrierSpacing                              = 1;


  #tdd-UL-DL-ConfigurationCommon
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    referenceSubcarrierSpacing                             = 1;
    # pattern1
    # dl_UL_TransmissionPeriodicity
    # 0=ms0p5, 1=ms0p625, 2=ms1, 3=ms1p25, 4=ms2, 5=ms2p5, 6=ms5, 7=ms10
    dl_UL_TransmissionPeriodicity                          = 5;
    nrofDownlinkSlots                              = 3;
    nrofDownlinkSymbols                            = 6;
    nrofUplinkSlots                             = 1;
    nrofUplinkSymbols                              = 4;

  ssPBCH_BlockPower                                  = -25;#0;
  }

  );
```

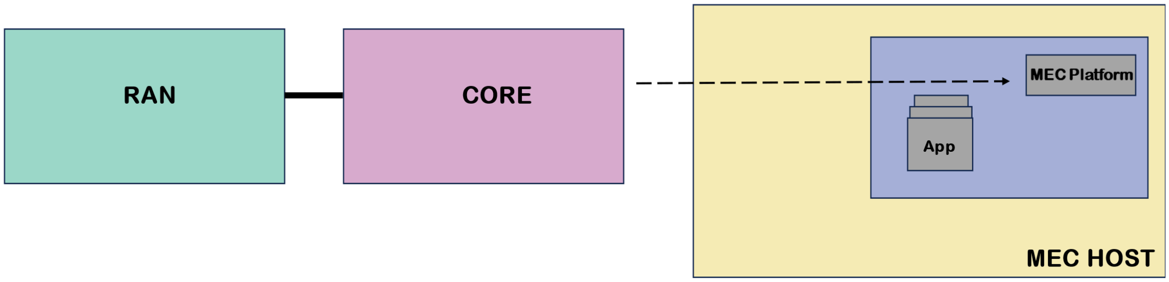
```
        # ------- SCTP definitions
    SCTP :
        {
      # Number of streams to use in input/output
      SCTP_INSTREAMS  = 2;
      SCTP_OUTSTREAMS = 2;
    };


#   ////////// AMF parameters:
#       amf_ip_address = ({ ipv4 = "172.21.6.5"; });
#
#   NETWORK_INTERFACES :
#   {
#      GNB_IPV4_ADDRESS_FOR_NG_AMF            = "172.21.18.20/22";
#      GNB_IPV4_ADDRESS_FOR_NGU               = "172.21.18.20/22";
#      GNB_PORT_FOR_S1U                       = 2152; # Spec 2152
#  };
      ////////// AMF parameters:
      amf_ip_address           = ( { ipv4       = "192.168.70.132";
                                     ipv6     = "192:168:30::17";
                                     active  = "yes";
                                     preference = "ipv4";
                                     }
                        );
        NETWORK_INTERFACES :

        {

#        GNB_INTERFACE_NAME_FOR_NG_AMF            = "enp3s0";

        GNB_IPV4_ADDRESS_FOR_NG_AMF              = "192.168.80.10";

#        GNB_INTERFACE_NAME_FOR_NGU               = "enp3s0";

        GNB_IPV4_ADDRESS_FOR_NGU                 = "192.168.80.10";

        GNB_PORT_FOR_S1U                         = 2152; # Spec 2152

        };

  }

);


#e2_agent = {

##  near_ric_ip_addr = "192.168.71.164";

#  near_ric_ip_addr = "192.168.70.164";

#  sm_dir = "/usr/local/lib/flexric/"

#}


MACRLCs = (

{

  num_cc                     = 1;

  tr_s_preference                  = "local_L1";

  tr_n_preference                  = "local_RRC";

  pusch_TargetSNRx10          = 250;
  pucch_TargetSNRx10         = 200;
 # dl_bler_target_upper=.35;
 # dl_bler_target_lower=.15;
#  ul_bler_target_upper=.35;
#  ul_bler_target_lower=.15;
  pusch_FailureThres         = 100;
  ulsch_max_frame_inactivity  = 0;
  ul_max_mcs                 = 20;
  dl_max_mcs                 = 26;
  min_grant_prb              = 20;
  min_grant_mcs              = 9;
}
);

L1s = (
{
  num_cc = 1;
  tr_n_preference = "local_mac";
  prach_dtx_threshold = 130;
  pucch0_dtx_threshold = 80;
  pusch_dtx_threshold = 20;#-100;
#thread_pool_size = 8;
  tx_amp_backoff_dB = 10;
  L1_rx_thread_core = 11;
  L1_tx_thread_core = 12;
  phase_compensation = 0; # needs to match O-RU configuration
}
);
RUs = (
{
  local_rf        = "no";
  nb_tx           = 2;   # change this to switch from SISO to MIMO
  nb_rx           = 2;   # change this to switch from SISO to MIMO
  att_tx          = 0
  att_rx          = 0;
  bands           = [78];
  max_pdschReferenceSignalPower = -27;
  max_rxgain                    = 75;
  sf_extension                  = 0;
  eNB_instances  = [0];
  ru_thread_core = 14;
  sl_ahead        = 5;
  ##beamforming 1x2 matrix: 1 layer x 2 antennas
  bf_weights = [0x00007fff, 0x0000,0x00007fff, 0x0000];
  clock_src = "internal";
  tr_preference  = "raw_if4p5"; # important: activate FHI7.2
  do_precoding = 0; # needs to match O-RU configuration
}
```

```
);

rfsimulator :
{
  serveraddr = "server";
  serverport = "4043";
  options = (); #("saviq"); or/and "chanmod"
  modelname = "AWGN";
  IQfile = "/tmp/rfsimulator.iqs";
};

security = {
  # preferred ciphering algorithms
  # the first one of the list that an UE supports in chosen
  # valid values: nea0, nea1, nea2, nea3
  ciphering_algorithms = ( "nea0" );

  # preferred integrity algorithms
  # the first one of the list that an UE supports in chosen
  # valid values: nia0, nia1, nia2, nia3
  integrity_algorithms = ( "nia2", "nia0" );

  # setting 'drb_ciphering' to "no" disables ciphering for DRBs, no matter
  # what 'ciphering_algorithms' configures; same thing for 'drb_integrity'
  drb_ciphering = "yes";
  drb_integrity = "no";
};


log_config : {
  global_log_level = "info";
  hw_log_level          = "info";
  phy_log_level         = "info";
  mac_log_level         = "info";
  rlc_log_level  = "info";
  pdcp_log_level   = "info";
  rrc_log_level = "info";
  ngap_log_level   = "info";
  f1ap_log_level   = "info";
  sdap_log_level   = "info";
};


fhi_72 = {
  dpdk_devices = ("0000:05:0a.0", "0000:05:0a.1");
  system_core = 3;
  io_core = 15;
  worker_cores = (13);
  du_addr = ("vf1 mac", "vf2 mac");
  ru_addr = ("ru mac for eth1", "ru mac for eth1");   #check and update from 192.168.4.50 (LPRU)
  mtu = 9216; # check if xran uses this properly

  file_prefix = "fhi_72";

  fh_config = ({

    Tadv_cp_dl = 125;

    T2a_cp_dl = (285, 429); # (min, max)

    T2a_cp_ul = (285, 429); # (min, max)

    T2a_up = (125, 428); # (min, max)

    Ta3 = (130, 170); # (min, max)

    T1a_cp_dl = (285, 470); # (min, max)

    T1a_cp_ul = (285, 429); # (min, max)

    T1a_up = (125, 350); # (min, max)

    Ta4 = (110, 180); # (min, max)

    ru_config = {

      iq_width = 9;

      iq_width_prach = 9;

#       fft_size = 12;

    };

    prach_config = {

      eAxC_offset = 2;   #change for MIMO and SISO switching

    };

  });

};
```

12. Mobile Edge Compute

Mobile Edge Compute (MEC) is a transformative technology that redefines how network services are delivered by bringing computing, storage, and data processing capabilities closer to the end-user at the network's edge. MEC plays a pivotal role in reducing latency, enhancing application performance, and enabling real-time data processing.

By strategically localizing computational resources near base stations or access points, MEC effectively minimizes the necessity for data to traverse extended network paths to centralized cloud data centers. This approach leads to enhanced response times and more efficient utilization of bandwidth.



12.1. MEC Hardware Specification

Server Type	MEC
Processor Model	Intel® Xeon Gold 3.0 GHz 5317 x 2
Memory(RAM)	32 GB DDR4 x 2
Storage Capacity	1.92 TB x 6
BIOS Settings	default
Operating System	Ubuntu 22.04
Kernel	Realtime Kernel

X = Lab specific subnet

12.2. Software Specification

Software Package	MEC
Docker	>= Version 26
Docker-compose	Version 1.29.2
linuxptp	NA

12.3. Network Specification

Server Type	MEC
Server IP	192.168.x.30
Netmask /Gateway :192.168.x.1	255.255.255.0
Ip forwarding configuration	Ip forwarding enabled
Ip route configuration	

13. Commands for MEC

1. Power on the server
2. Enter the password.
3. Follow different devices documents for different MEC applications for example camera analytics, Drone streaming etc.

13.1. Route Configuration for Mec Applications

On Core

Sudo ip route add 12.0.0.0/24 via 192.168.70.155

On MEC

Sudo ip route add 12.0.0.0/24 via 192.168.x.20

13.2. Camera

13.2.1 Camera Connection through LAN

1. Power on the camera
2. Connect LAN and MEC through a static ip.
3. Execute **cd camera-analytics** command on MEC to check the connection.

13.2.2. Wireless Camera Connection

1. Power on the camera
2. Connect camera via P5N
3. Execute **ping CAMERA IP** on MEC to check the connection.

13.2.3. Camera Usage Steps

1. For object detection

```
make object
```

2. For face detection

```
make detect
```

Note: If the user face is not added to the dataset it will show "Unknown", for that there is one optional step to add face to the dataset.

```
make sample
```

3. When asked enter the name of the person

4. The Camera will start, make sure to face the camera in the center location to add sample, once done the camera UI will close.

5. For face prediction

```
make predict
```

13.3. Drone

view the webpage on - **http://(mecip):5000**

for viewing drone stream enter link - **rtsp://(drone ip):10000/drone_cam**

NOTE: Please refer to Configuration iDronam section under Configuration for sensors.

14. RU Overview

Low Power Radio Unit (LPRU) is a single-board Optical to Radio interface solution for 5G low-power RU application, The RU features a digital and RF section on a single board, enhancing 5G network coverage and permitting multiple-stage cascading.

14.1. Key Specifications:

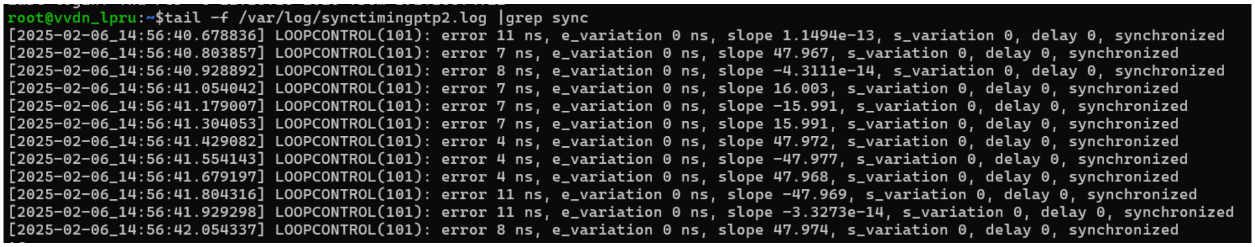
- Deployment Type: Indoor RU
- Radio Configuration: 1 CC, 100 MHz, 2Tx2R, n78 (Center Frequency 3.6 GHz)
- BandWidth: 100MHz
- Mode: Single Numerology, SCS 30KHz
- Antenna: External
- Fronthaul: 10G over SFP+
- Powered using PoE++ Type 3 (IEEE PoE standard, 802.3bt)
- RF Power: 4x250mW
- O-RAN Split: 7-2a
- Internal Antenna Gain Range: 6-9dBi

15. Configuration for RU

15.1. Steps for Pre-Configured RU

1. Login into RU: ssh root@192.168.4.50
    Password: vvdn

2. After login:
 tail -f /var/log/synctimingptp2.log |grep sync



```
root@vvdn_lpru:~$tail -f /var/log/synctimingptp2.log |grep sync
[2025-02-06_14:56:40.678836] LOOPCONTROL(101): error 11 ns, e_variation 0 ns, slope 1.1494e-13, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:40.803857] LOOPCONTROL(101): error 7 ns, e_variation 0 ns, slope 47.967, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:40.928892] LOOPCONTROL(101): error 8 ns, e_variation 0 ns, slope -4.3111e-14, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:41.054042] LOOPCONTROL(101): error 7 ns, e_variation 0 ns, slope 16.003, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:41.179007] LOOPCONTROL(101): error 7 ns, e_variation 0 ns, slope -15.991, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:41.304053] LOOPCONTROL(101): error 7 ns, e_variation 0 ns, slope 15.991, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:41.429082] LOOPCONTROL(101): error 4 ns, e_variation 0 ns, slope 47.972, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:41.554143] LOOPCONTROL(101): error 4 ns, e_variation 0 ns, slope -47.977, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:41.679197] LOOPCONTROL(101): error 4 ns, e_variation 0 ns, slope 47.968, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:41.804316] LOOPCONTROL(101): error 11 ns, e_variation 0 ns, slope -47.969, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:41.929298] LOOPCONTROL(101): error 11 ns, e_variation 0 ns, slope -3.3273e-14, s_variation 0, delay 0, synchronized
[2025-02-06_14:56:42.054337] LOOPCONTROL(101): error 8 ns, e_variation 0 ns, slope 47.974, s_variation 0, delay 0, synchronized
```

You will get a message that the system is synchronized. If it is not, kindly allow some time for it to synchronize automatically.

2.1 Command to make ru_compression-enable.sh
vi ru_compression-enable.sh
#!/bin/bash
mw.l a0010024 1919
Press **esc**
Then, press **shift+wq**, you will get **:wq**
Press **enter**
chmod +x ru_compression-enable.sh for making this file executable

> NOTE: Only Use 2.1 Commands if not Pre-Configured.

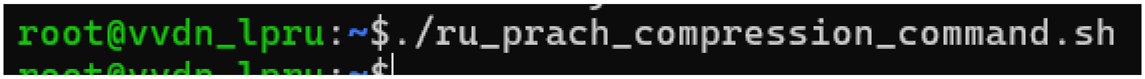
Configuration for 1x1 xml

```
<vvdn_lpru_config>
    <du_mac_address>00:11:22:33:44:66</du_mac_address>
    <cu_plane_vlan>100</cu_plane_vlan>
    <dl_compression_method>1</dl_compression_method>
    <dl_compression_value>9</dl_compression_value>
    <ul_compression_method>1</ul_compression_method>
    <ul_compression_value>9</ul_compression_value>
    <num_prb>273</num_prb>
    <prach_layer0_PCID>1</prach_layer0_PCID>
    <prach_layer1_PCID>5</prach_layer1_PCID>
    <prach_layer2_PCID>6</prach_layer2_PCID>
    <prach_layer3_PCID>7</prach_layer3_PCID>
    <pxsch_layer0_PCID>0</pxsch_layer0_PCID>
    <pxsch_layer1_PCID>2</pxsch_layer1_PCID>
    <pxsch_layer2_PCID>3</pxsch_layer2_PCID>
    <pxsch_layer3_PCID>4</pxsch_layer3_PCID>
 </vvdn_lpru_config>
```

Configuration for 2x2 xml

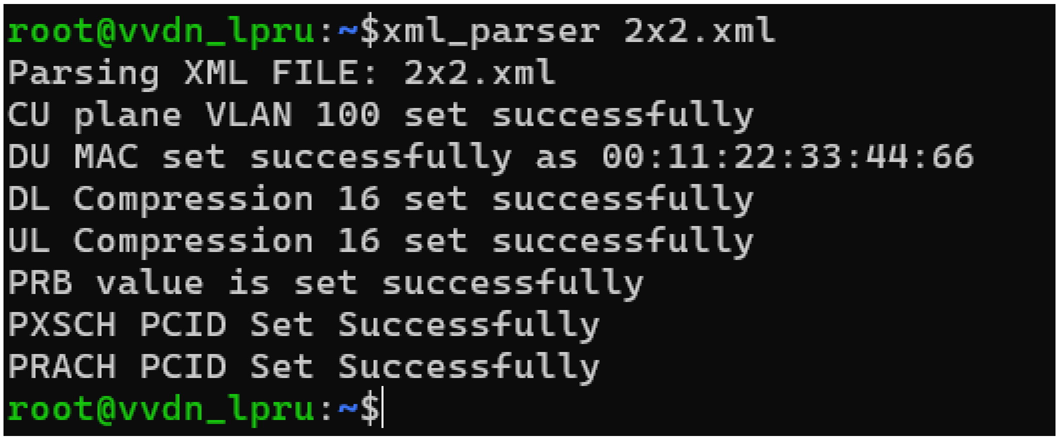
```
<vvdn_lpru_config>

   <du_mac_address>00:11:22:33:44:66</du_mac_address>
    <cu_plane_vlan>100</cu_plane_vlan>
    <dl_compression_method>1</dl_compression_method>
    <dl_compression_value>9</dl_compression_value>
    <ul_compression_method>1</ul_compression_method>
    <ul_compression_value>9</ul_compression_value>
    <num_prb>273</num_prb>
    <prach_layer0_PCID>2</prach_layer0_PCID>
    <prach_layer1_PCID>3</prach_layer1_PCID>
    <prach_layer2_PCID>6</prach_layer2_PCID>
    <prach_layer3_PCID>7</prach_layer3_PCID>
    <pxsch_layer0_PCID>0</pxsch_layer0_PCID>
    <pxsch_layer1_PCID>1</pxsch_layer1_PCID>
    <pxsch_layer2_PCID>4</pxsch_layer2_PCID>
    <pxsch_layer3_PCID>5</pxsch_layer3_PCID>
 </vvdn_lpru_config>
```

3. Run this executable by: **./ru_compression-enable.sh**



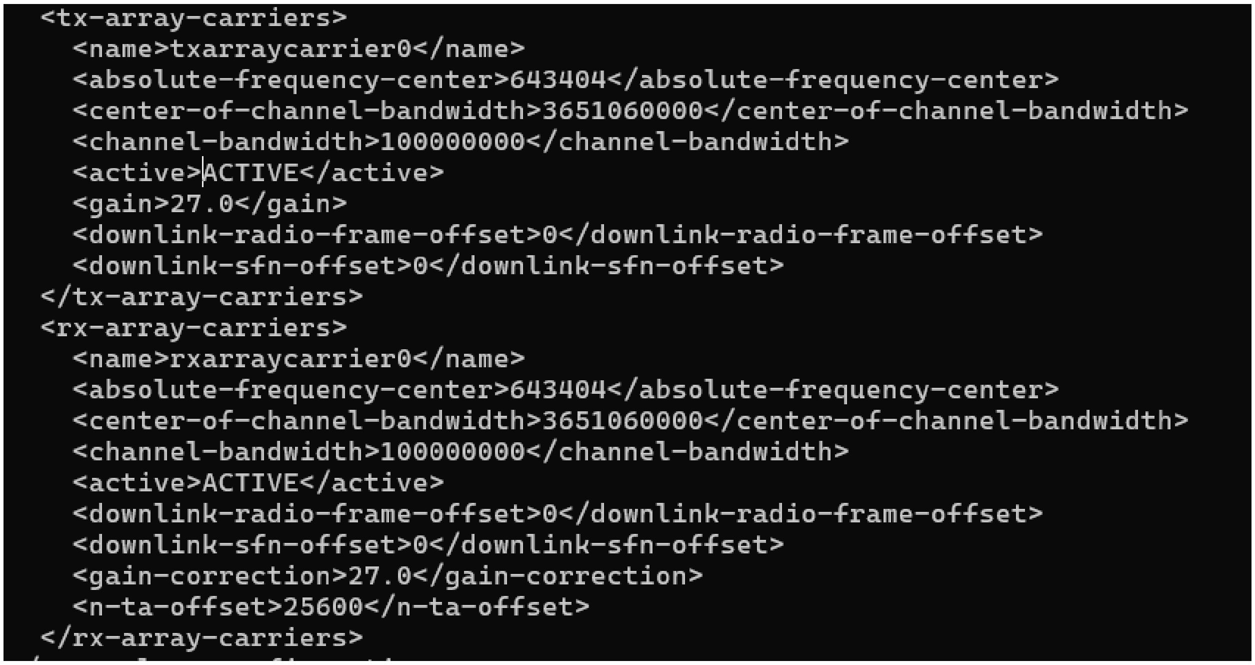
```
root@vvdn_lpru:~$./ru_prach_compression_command.sh
root@vvdn_lpru:~$
```

4. xml_parser 2x2.xml (2x2.xml is file name)



```
root@vvdn_lpru:~$xml_parser 2x2.xml
Parsing XML FILE: 2x2.xml
CU plane VLAN 100 set successfully
DU MAC set successfully as 00:11:22:33:44:66
DL Compression 16 set successfully
UL Compression 16 set successfully
PRB value is set successfully
PXSCH PCID Set Successfully
PRACH PCID Set Successfully
root@vvdn_lpru:~$
```

5. sysrepocfg --edit=vi -d running
   Then search for INACTIVE like this - /INACTIVE and then press enter
   Then press i then remove IN and make it ACTIVE
   After 9 lines there is one more INACTIVE make it ACTIVE
   Then press esc
   Type **:wq** to save the changes
   Then press enter



```
  <tx-array-carriers>
    <name>txarraycarrier0</name>
    <absolute-frequency-center>643404</absolute-frequency-center>
    <center-of-channel-bandwidth>3651060000</center-of-channel-bandwidth>
    <channel-bandwidth>100000000</channel-bandwidth>
    <active>ACTIVE</active>
    <gain>27.0</gain>
    <downlink-radio-frame-offset>0</downlink-radio-frame-offset>
    <downlink-sfn-offset>0</downlink-sfn-offset>
  </tx-array-carriers>
  <rx-array-carriers>
    <name>rxarraycarrier0</name>
    <absolute-frequency-center>643404</absolute-frequency-center>
    <center-of-channel-bandwidth>3651060000</center-of-channel-bandwidth>
    <channel-bandwidth>100000000</channel-bandwidth>
    <active>ACTIVE</active>
    <downlink-radio-frame-offset>0</downlink-radio-frame-offset>
    <downlink-sfn-offset>0</downlink-sfn-offset>
    <gain-correction>27.0</gain-correction>
    <n-ta-offset>25600</n-ta-offset>
  </rx-array-carriers>
```

6. Logout Command from RU

>> exit

16. APN configuration

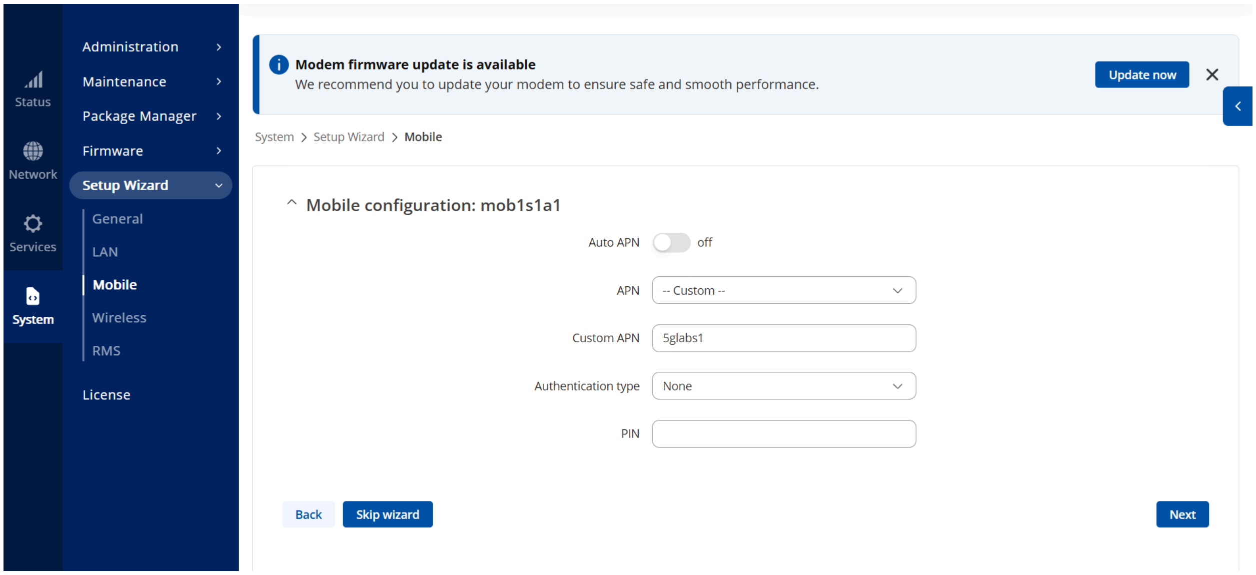
16.1.Configuration for IoT Gateway

1. Power ON the rutx50 and connect it to a pc or laptop via wifi (the credentials will be at the back of the rutx) or LAN.

2. Access the RUTX web configuration page by inputting the IP address: 192.168.1.1 in a web browser. The login credentials for RUTX are
Username: admin
Password :

*(for first access input password described on the back of the device and then change it to your desired password)

3. In the GUI of rutx50 navigate to System -> Setup Wizard -> Mobile



Modem firmware update is available
We recommend you to update your modem to ensure safe and smooth performance.

Mobile configuration: mob1s1a1

Auto APN	off
APN	-- Custom --
Custom APN	5glabs1
Authentication type	None
PIN	

The following settings are to be done as shown in the above image

Auto apn	=	off
APN	=	custom
Custom APN	=	5glabs1
Authentication Type	=	None

PIN =

4. Press Next twice till you reach the RMS setting page (do not configure anything here) press finish.

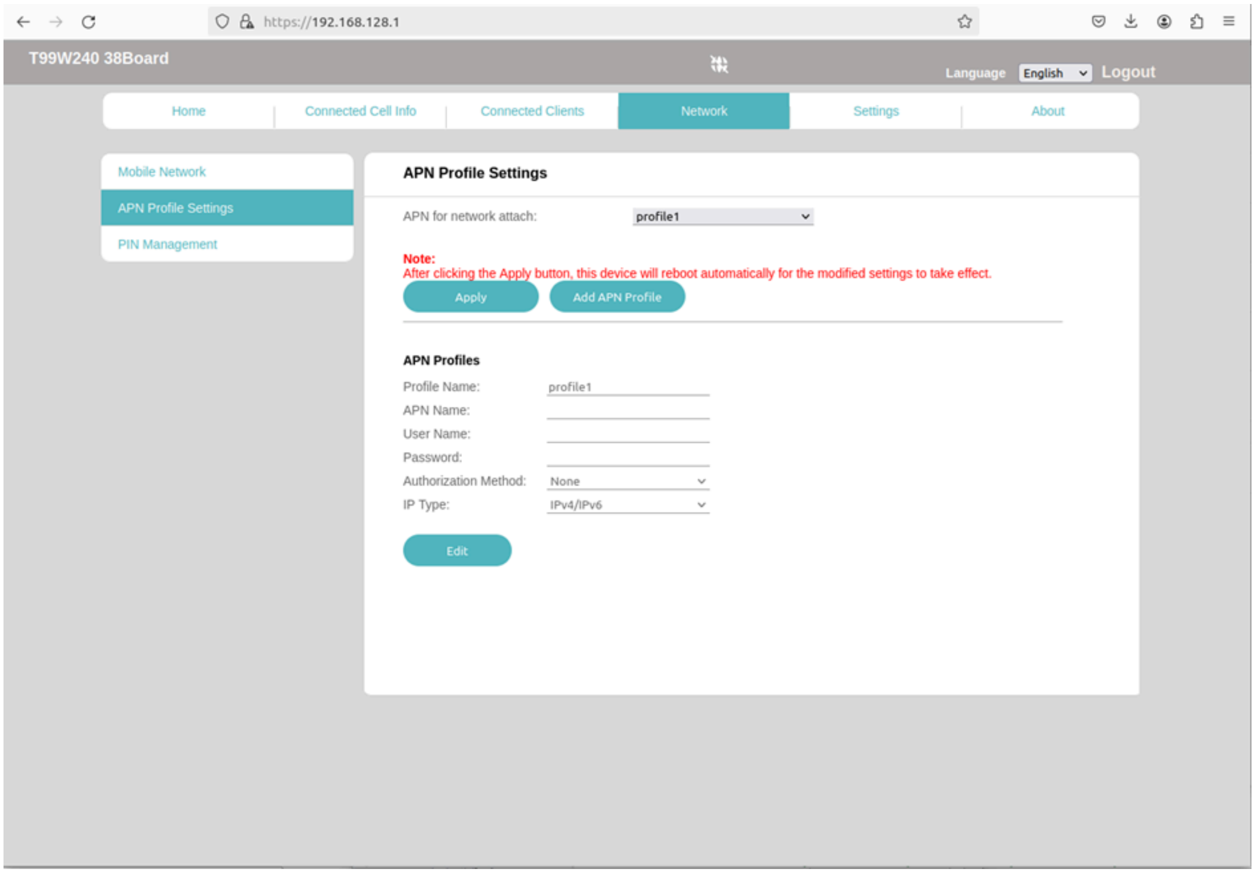
16.2. Configuration for Camera

1. Connect the camera to a laptop or pc after powering it on via an ethernet cable or USB (port can be accessed after opening the lid of the camera)

2. Open Firefox web browser and go to the Cameras web gui by inputting the following ip in the web browser : 192.168.128.1

3. Press Login and enter (admin) to access the following settings

4. Go to Network -> APN Profile Settings



APN Profile Settings

APN for network attach: profile1

Note:
After clicking the Apply button, this device will reboot automatically for the modified settings to take effect.

APN Profiles

Profile Name: profile1
APN Name:
User Name:
Password:
Authorization Method: None
IP Type: IPv4/IPv6

5. Click on Add APN Profile and add the following details

Profile Name:	w4slab
APN Name:	5glabs1
User Name:	
Password:	
Authorization Method:	None
IP Type:	ipv4

6. Press OK to save the apn profile

7. Then switch the apn profile by toggling from profile 1 to 5glabs1in the APN for network attach file.

8. Click apply and press OK

9. After successful selection of manually configured apn, do delete old apn

10. GO to SETTINGS -> PORT MAPPING -> create rule and map port 554 of wan to port 554 of lan on lan ip 192.168.128.10

16.3. Configuration for CPE

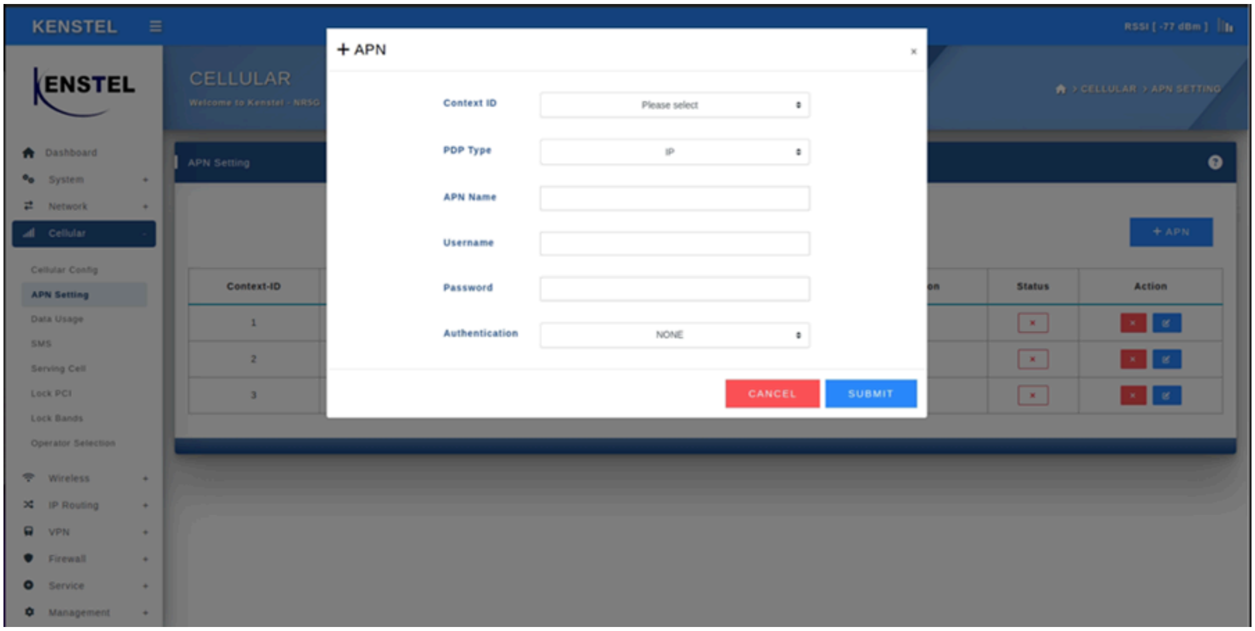
1. Power ON the Kenstel CPE and connect it to a PC or laptop via Wi-Fi (the credentials will be on the back of it) or LAN.

2. Access the web UI by entering the ip address: 51.0.0.1 in a web browser.

3. The login credentials for Kenstel CPE are Username: root, Password: root

4. After successful login into CPE UI

5. Go to Cellular > APN Setting > Click on +APN



+ APN

Context ID: Please select

PDP Type: IP

APN Name:

Username:

Password:

Authentication: NONE

6. A dialog box will appear with blank details

7. Fill out mandatory fields as shown below

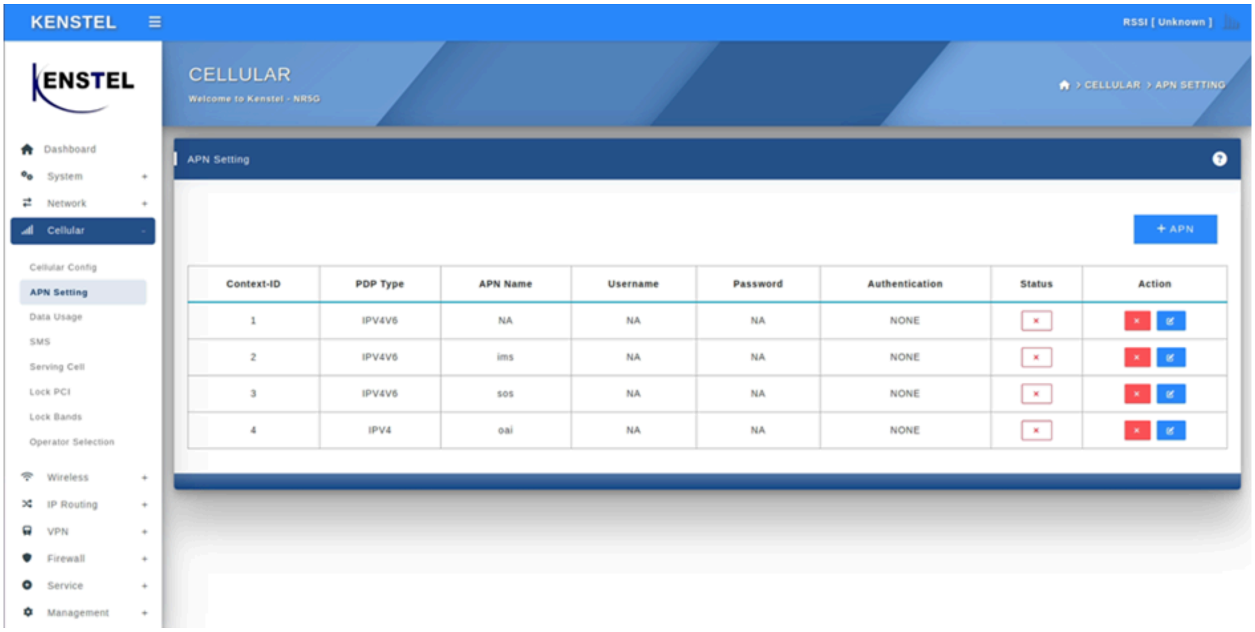
Context id 4

Pdp type: ipv4

Apn name 5glabs1

Authentication: none

Rest are blank



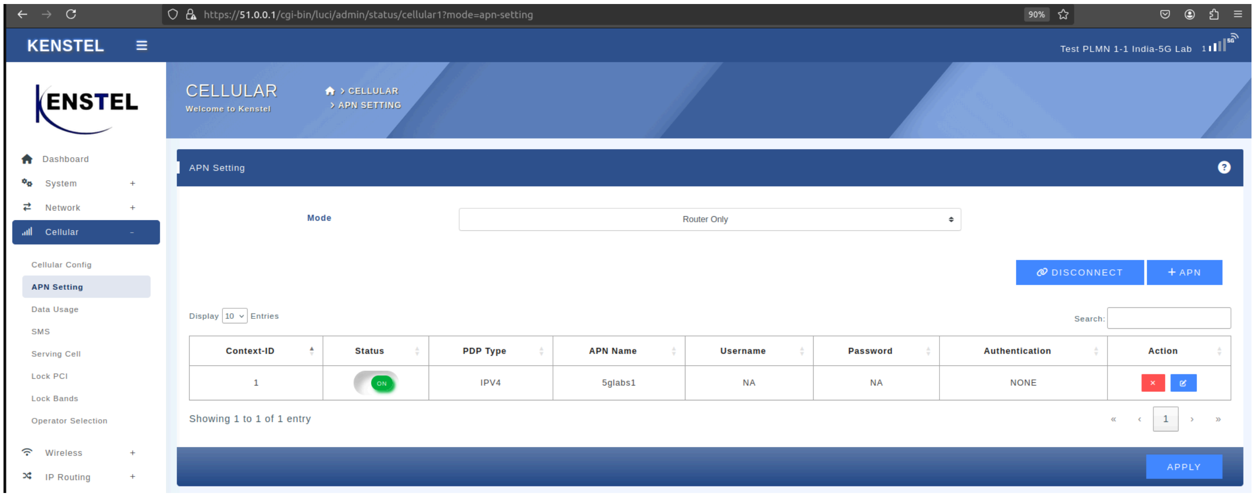
Context-ID	PDP Type	APN Name	Username	Password	Authentication	Status	Action
1	IPV4V6	NA	NA	NA	NONE	×	× ✎
2	IPV4V6	ims	NA	NA	NONE	×	× ✎
3	IPV4V6	sos	NA	NA	NONE	×	× ✎
4	IPV4	oai	NA	NA	NONE	×	× ✎

8. After filling out all details click on Submit button

9. Data has been updated successfully

10. After adding the APN delete the old APN manually which consists of ims and sos

11. After refreshing the page, you can check your newly configured APN in the list.



Mode: Router Only

Context-ID	Status	PDP Type	APN Name	Username	Password	Authentication	Action
1	ON	IPV4	5glabs1	NA	NA	NONE	× ✎

Showing 1 to 1 of 1 entry

16.4. Configuration for Evaluation Board

Power on and Startup

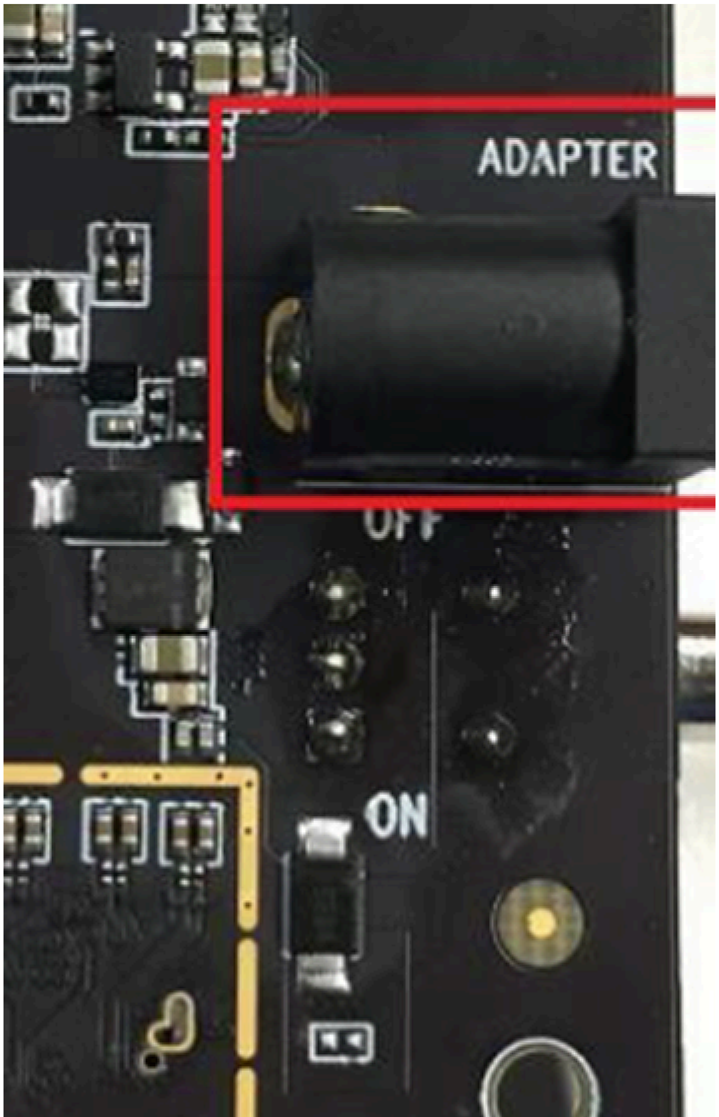
Add Eval board drivers on the machine.

Insert Private 5G SIM in the Eval board (If not already present).

Anubhav evaluation board can be powered using the Supplied adaptor or USB cable (on-side USB Port.)

Set the power switch to the left.

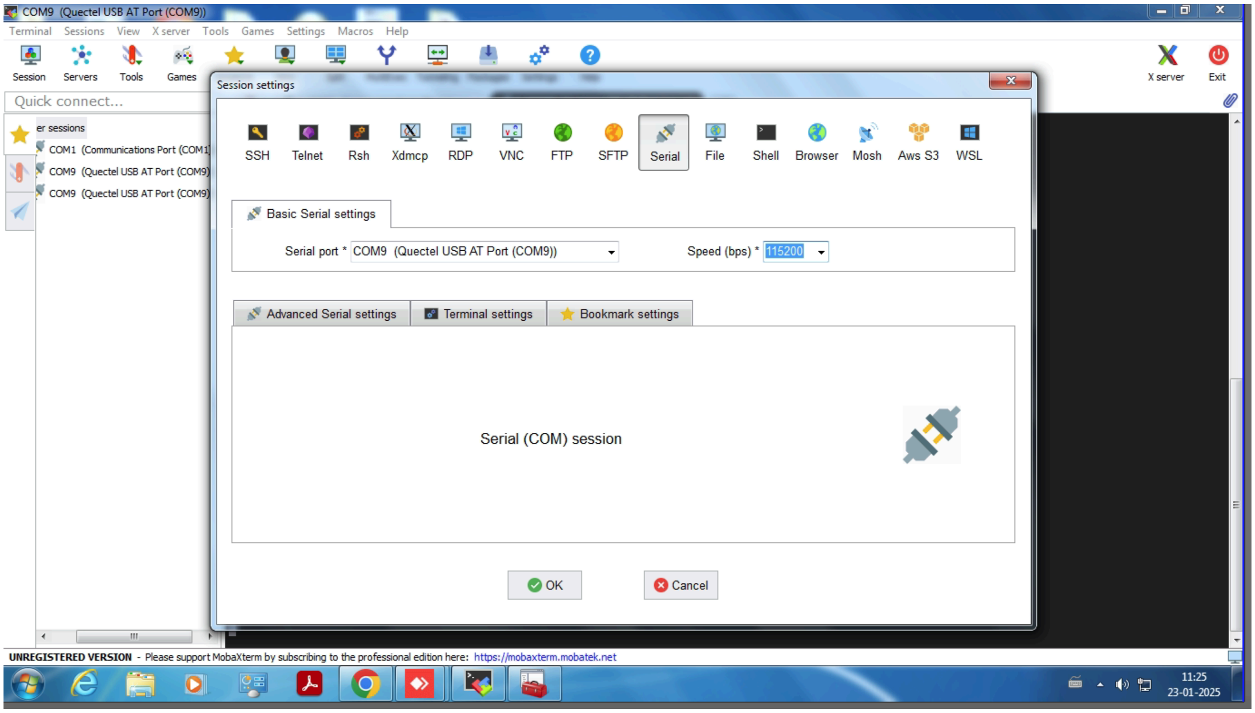
Press the reset button for 10 seconds.



Open the terminal using terminal emulation SW like minicom or Moba Xterm. (Check the right com port in the device manager).

Moba Xterm

Go to Device Manager→At Port→Session Setting→serial option→choose serial port AT Port and select speed.



COM9 (Quectel USB AT Port (COM9))

Session settings

Basic Serial settings

Serial port * COM9 (Quectel USB AT Port (COM9))

Speed (bps) * 115200

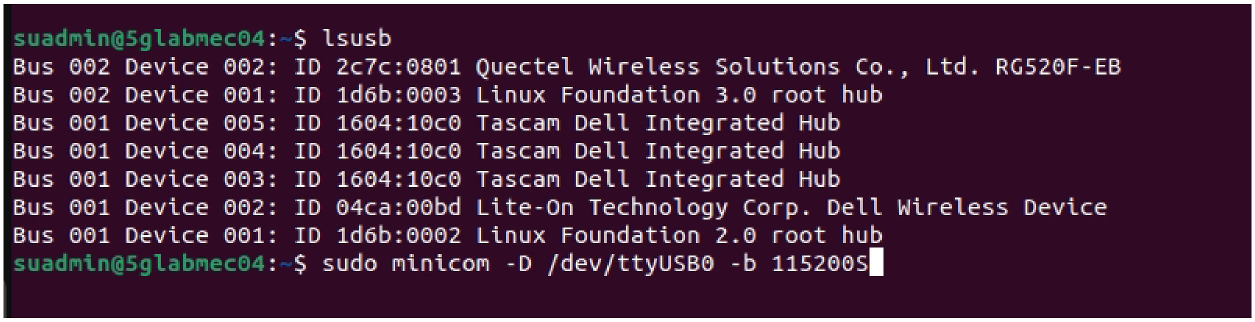
Serial (COM) session

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

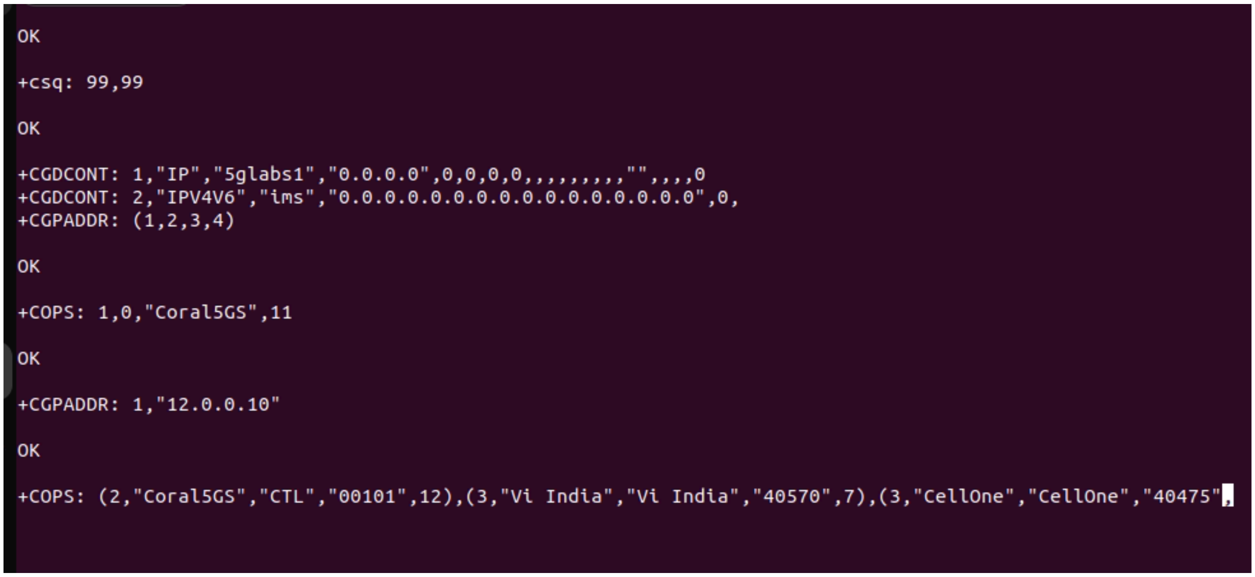
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games
X server Exit
Quick connect...
er sessions
COM1 (Communications Port (COM1
COM9 (Quectel USB AT Port (COM9)
COM9 (Quectel USB AT Port (COM9)
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL
Advanced Serial settings
Terminal settings
Bookmark settings
OK
Cancel
11:25
23-01-2025

Minicom

Open Terminal on Mec→execute **lsusb** command→execute **sudo minicom -D /dev/ttyUSB0 -b 115200S** for setting the speed.

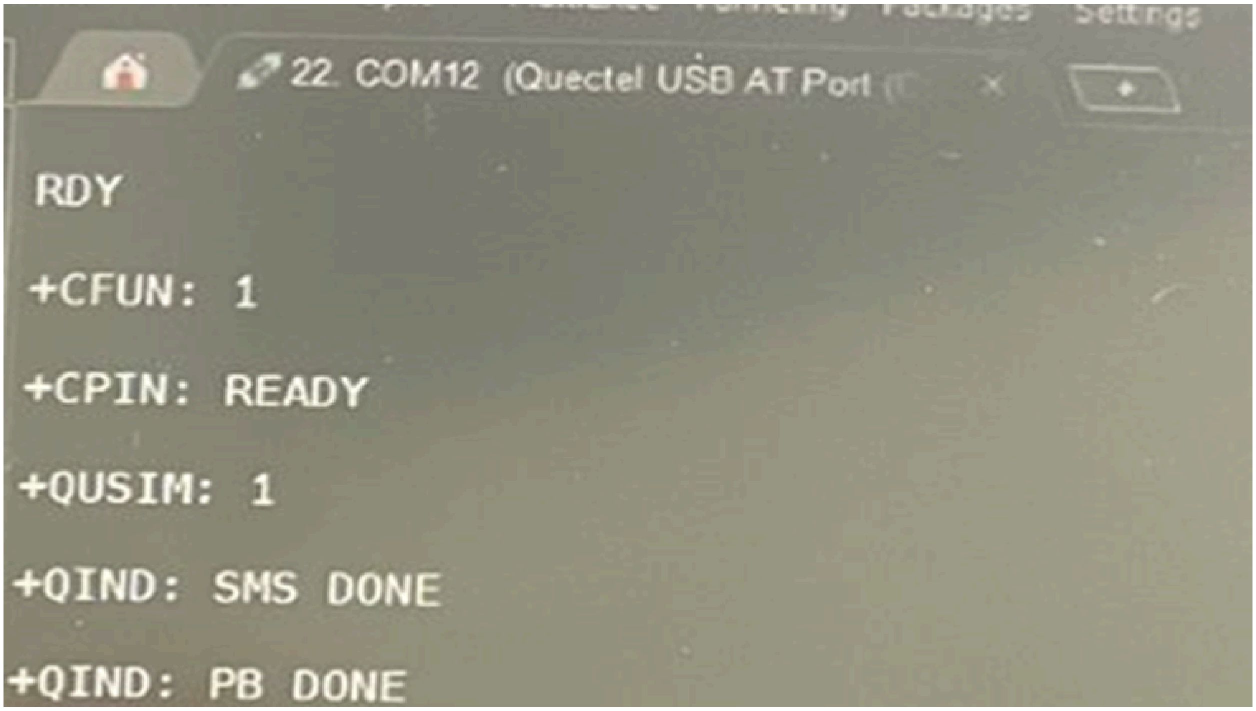


```
suadmin@5glabmec04:~$ lsusb
Bus 002 Device 002: ID 2c7c:0801 Quectel Wireless Solutions Co., Ltd. RG520F-EB
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 1604:10c0 Tascam Dell Integrated Hub
Bus 001 Device 004: ID 1604:10c0 Tascam Dell Integrated Hub
Bus 001 Device 003: ID 1604:10c0 Tascam Dell Integrated Hub
Bus 001 Device 002: ID 04ca:00bd Lite-On Technology Corp. Dell Wireless Device
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
suadmin@5glabmec04:~$ sudo minicom -D /dev/ttyUSB0 -b 115200S
```



```
OK

+csq: 99,99

OK

+CGDCONT: 1,"IP","5glabs1","0.0.0.0",0,0,0,0,,,,,,,,,"",,,,0
+CGDCONT: 2,"IPV4V6","ims","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,
+CGPADDR: (1,2,3,4)

OK

+COPS: 1,0,"Coral5GS",11

OK

+CGPADDR: 1,"12.0.0.10"

OK

+COPS: (2,"Coral5GS","CTL","00101",12),(3,"Vi India","Vi India","40570",7),(3,"CellOne","CellOne","40475",
```

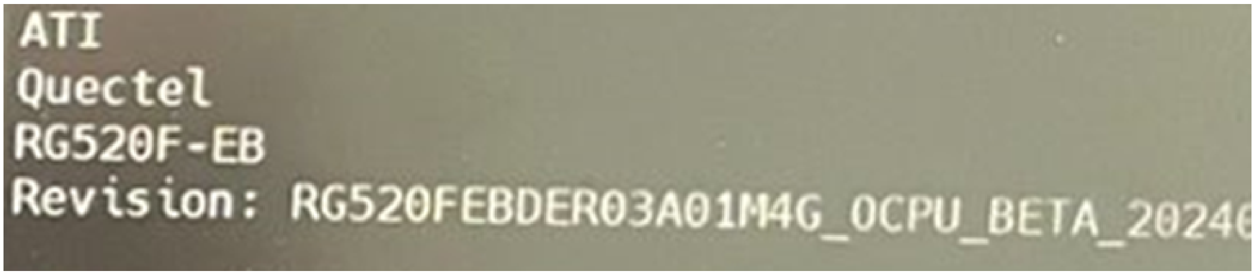
It shall display the following when ready to take commands.



```
RDY

+CFUN: 1

+CPIN: READY

+QUSIM: 1

+QIND: SMS DONE

+QIND: PB DONE
```

Basic Network and Configuration Commands

ATI // Returns Information Model number and firmware number.



```
ATI
Quectel
RG520F-EB
Revision: RG520FEBDER03A01M4G_OCPU_BETA_2024
```

AT+CIMI   //returns IMSI number

AT+COPS=? //Displays list of available network, Check network "00101" is available

//if ERROR in AT+COPS=?. This can also be done in other error states or to deregister and register again.

AT+CFUN=0 //Returns OK, Switch UE to minimum functionality.

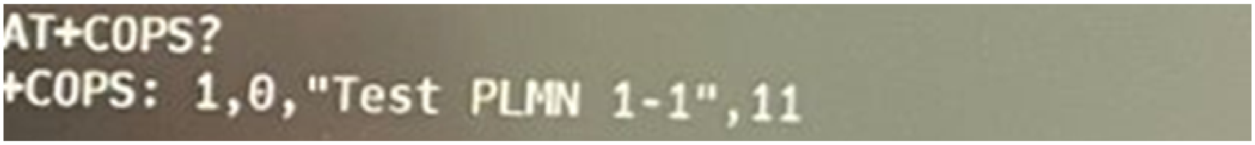
AT+CFUN=1 //Returns OK, Switch UE to Full functionality.

AT+CGDCONT? // Displays list of APNs, check apn to configure as per Network Slice.

AT+CGDCONT=1,"IP","5glabs1" // set apn to "5glabs1"

AT+QNWPREFCFG= "mode_pref",NR5G //Returns OK, Set eval board to work in NR5G mode only

AT+COPS? // Displays details of the registered network, if only 0 (automatic selection) or 1 (Manual selection), the Eval board is not registered on the network



```
AT+COPS?
+COPS: 1,0,"Test PLMN 1-1",11
```

//AT+COPS=1,2,"00101" // To register manually

AT+CGPADDR=1 // Check IP address is assigned, if 0.0.0.0 means PDU session is not established, use CFUN=0 & CFUN=1, if issue persists check Network logs.

Check that QsoftAP is available in wifi networks.

Password is 1xxxxx

Device and Module Info

Get Firmware and Device Information: AT+CGMR

Display firmware version.

Check 5G MIMO Status: AT+QNWCFG="nr5g_mimo"

Check if 5G MIMO is enabled.

Enable 5G MIMO: AT+QNWCFG="nr5g_mimo",1

Enable 5G MIMO.

Interface and Peripheral Commands

Check Supported Configurations: AT+QCFG=?

List supported configurations and options.

Network-related Configuration Options: AT+QNWCFG=?

List network-related configuration options (e.g., nr5g_cdrx, nr5g_tx_pwr, nr5g_mimo).

SD Card Commands

See the details of the SD Card using ADB, Follow the given instructions precisely:

1. Install ADB in your system

2. After that connect your PC to the 5G Evaluation Board

3. Access the command prompt and launch adb and enter the command to show all devices(adb devices).

4.Access the 5G Evaluation Board and see the SD card details.

Raspberry Pi Configuration

Connect your 5G Evaluation Board with the Raspberry Pi using type c data cable.

After that connect your PC to the Raspberry Pi using Putty and by using the Static IP method/configuration. Make sure your pc has same network with Raspberry pi, to take the ssh of Raspberry pi

After accessing the Raspberry Pi enter the creditionals if any and switch to super user using the command (sudo su).

After that Run the command to see the active USB drives in the Raspberry Pi and after then after that run the command:

```
-minicom -D /dev/ttyUSB<USB Drive Connected> -b 115200
```

After this you will be able to control your 5G Evaluation Board via Raspberry Pi.

AT Command Sequence for Activating GPS on Quectel Module

1. Set "Auto GPS" to Enable GPS Automatically

```
AT+QGPSCFG="autogps",1
OK
```

2. Activate GPS

```
AT+QGPS=1
OK
```

16.5. Configuration for Drone

Configuration Steps:

1. **Insert 5G Nano SIM:** Insert the 5G nano SIM card into the drone's SIM slot.

2. **Power On Drone:** Turn on the drone and ensure it is powered up properly.

3. **Open Terminal Connection on MEC Server:** Connect to the drone's modem via a terminal emulator such as Minicom or PuTTY.

4. **Configure PDP Context:** Set up the Packet Data Protocol (PDP) context for internet connectivity using the following commands:

   ```
   AT+CGDCONT=1,"IPV4V6","oai"
   ```

5. **Verify PDP Context:** Check the configuration using:

   ```
   AT+CGDCONT?
   ```

6. **Check Network Registration Status:** Ensure the modem is registered on the network:

   ```
   AT+C5GREG?
   ```

   A response of **`+C5GREG: 0,1`** **i**ndicates successful registration.

7. **Obtain IP Address:** Check the assigned IP address:

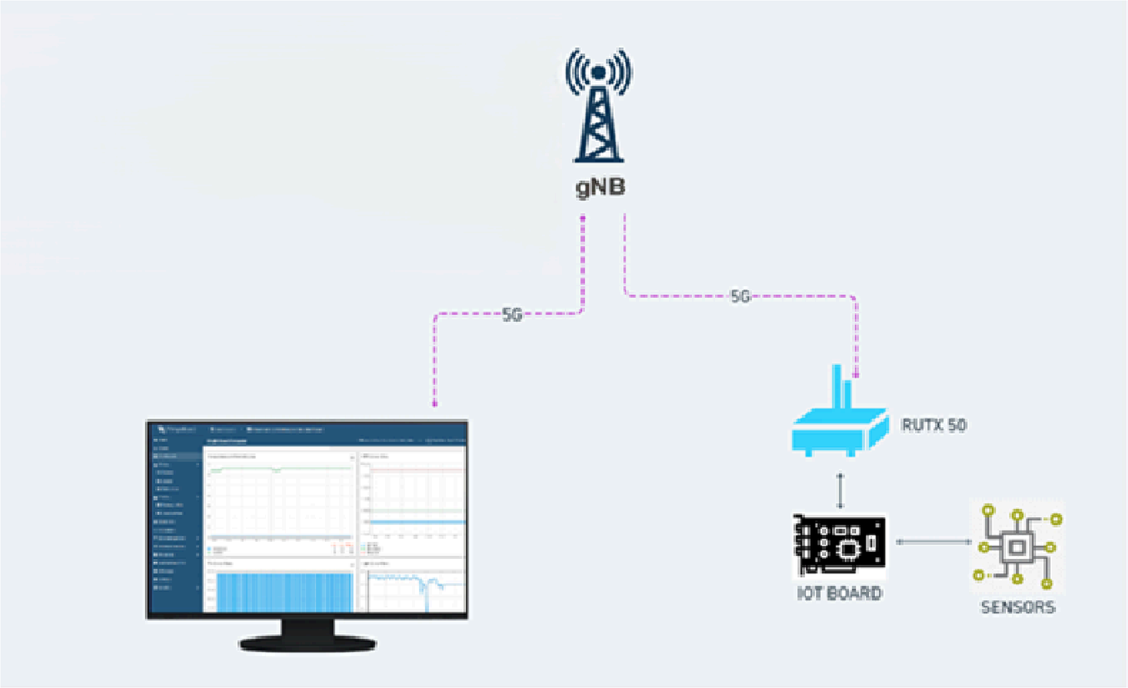
   ```
   AT+CGPADDR=1
   ```

8. **Check Network Information:** Get network information to ensure 5G connection:

   ```
   AT+COPS?

   AT+QNWINFO
   ```

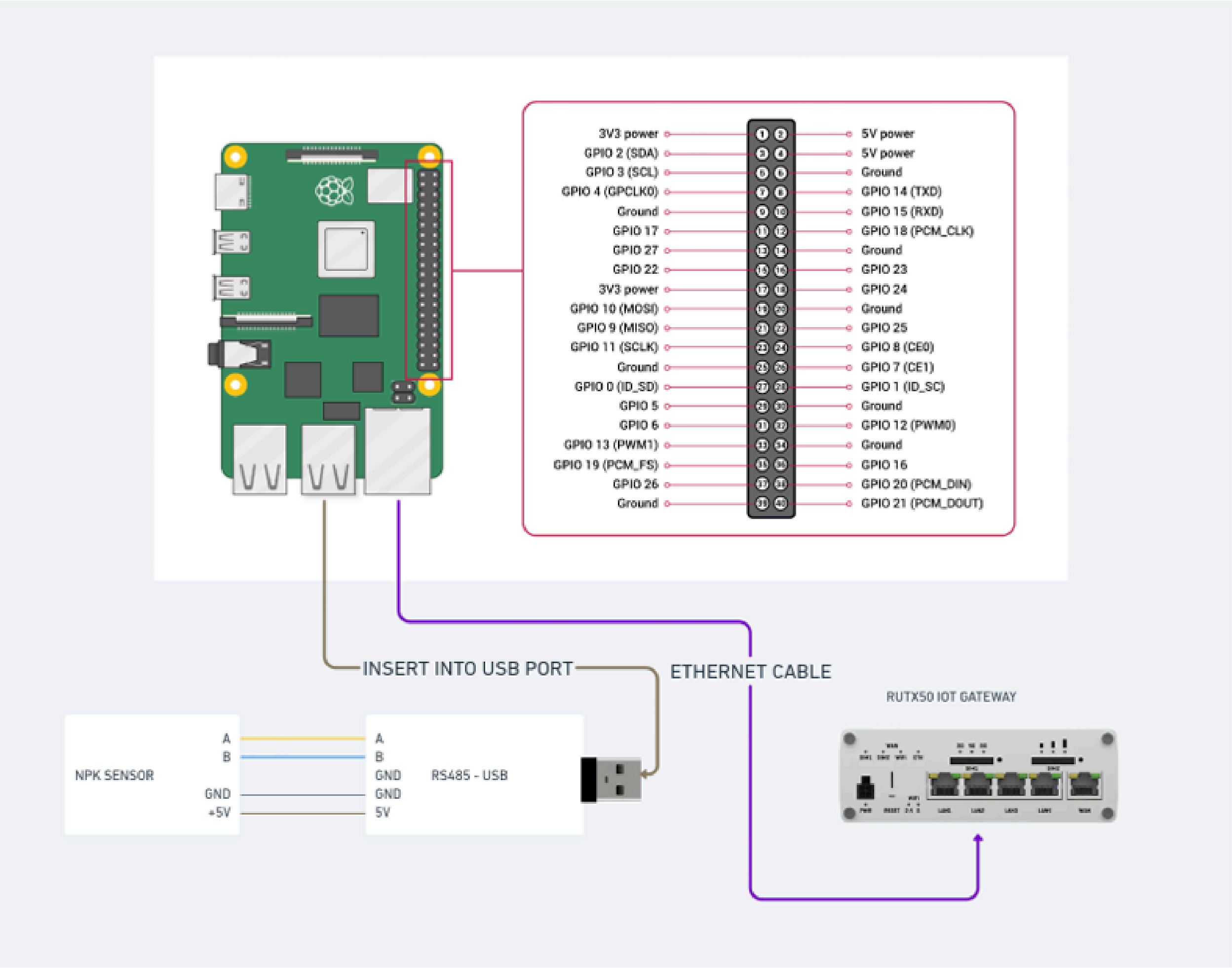
17. Configuration For Sensors

17.1. IoT Setup



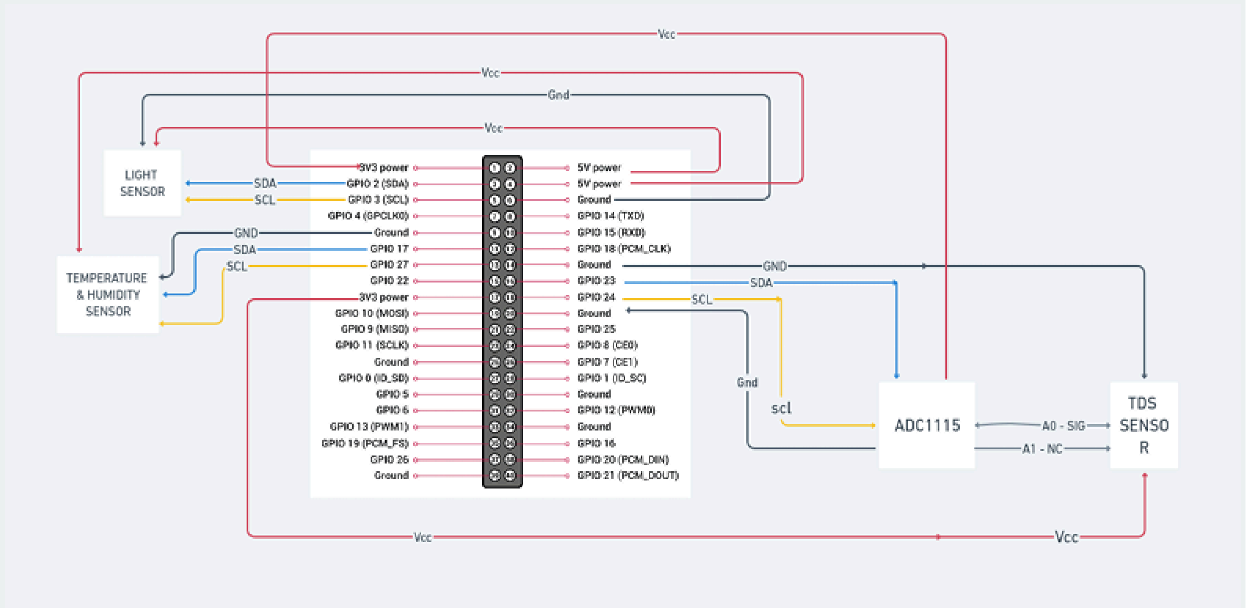
To establish a connection with the gNB, connect through the 5G network interface to the RUTX 50. From the RUTX 50, a connection to the IoT board can be established, enabling communication with IoT sensors.

17.2. Configuration for NPK Sensor



Insert USB into the microprocessor to establish the connection with the NPK sensor and connect IoT Gateway through an ethernet cable to the microprocessor.

17.3. Configuration for Light, TDS and Temperature and Humidity Sensors



15.3.1. Steps for configuration for Raspberry PI

1. ssh w4s@ip_address
2. sudo apt-get update
3. sudo apt-get upgrade
4. sudo vim /boot/config.txt

add the following lines

```
dtparam=i2c_arm=on
#dtparam=i2s=on
dtparam=spi=on
```

#i2c bus-3

(password:Gateway@123)

```
dtoverlay=i2cgpio,bus=3,i2c_gpio_delay_us=1,i2c_gpio_sda=17,i2c_gpio_scl=27
```

#i2c bus-4

```
dtoverlay=i2cgpio,bus=4,i2c_gpio_delay_us=1,i2c_gpio_sda=23,i2c_gpio_scl=24
```

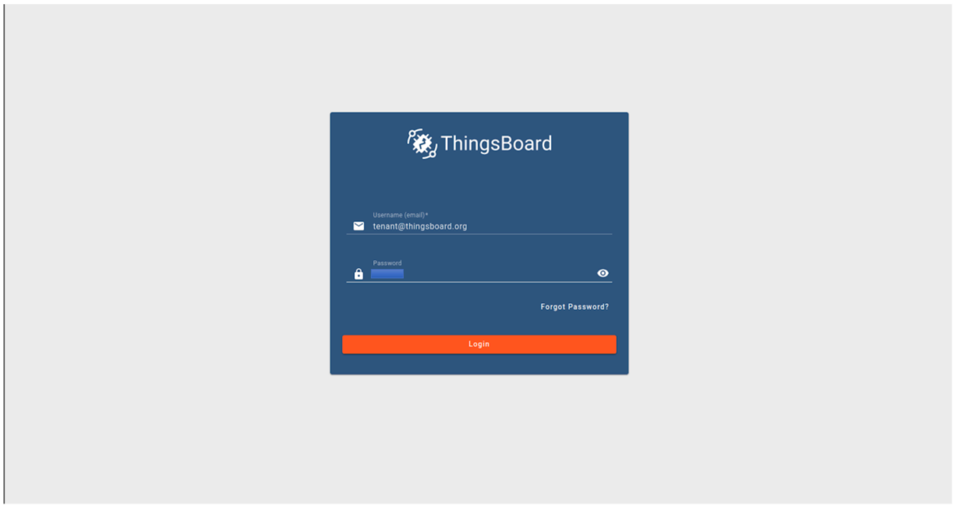
sudo reboot to apply the changes

17.4. Configuration steps for Thingsboard

1. Open Thingsboard dashboard in browser on MEC
   192.168.X.30:9090

2. Log in to Thingsboard via the following credentials

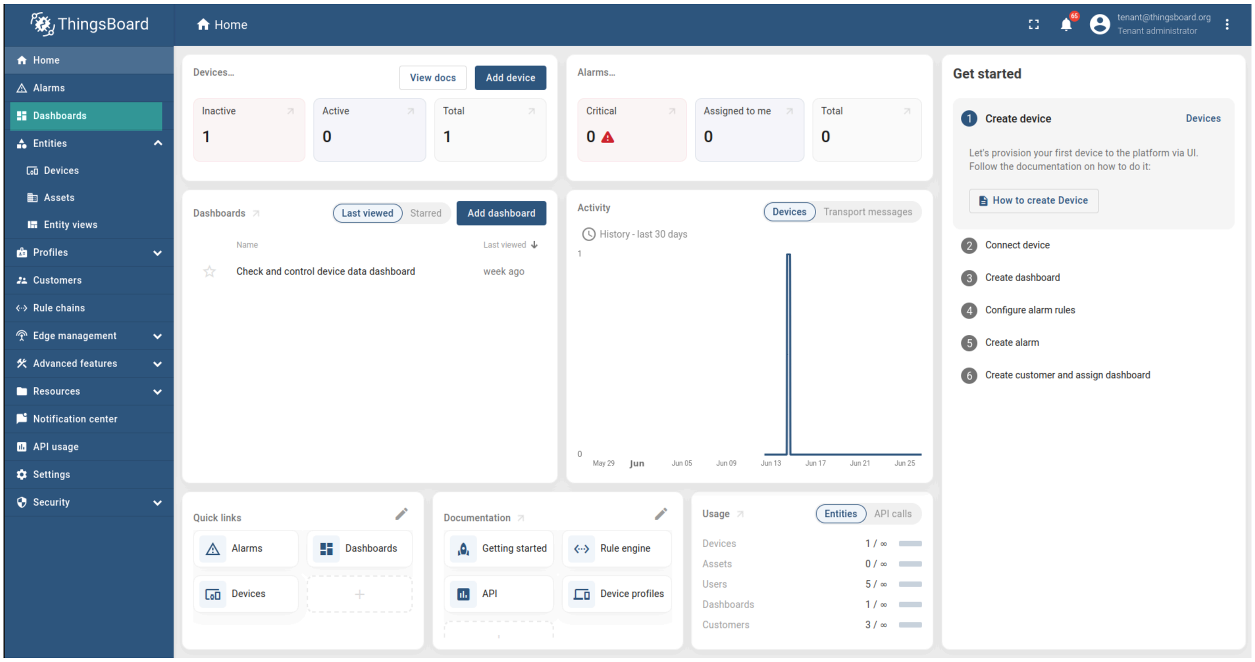
Username :
Password :



Username (email)* tenant@thingsboard.org

Password [redacted]

3. Go to the dashboard after logging into Thingsboard.

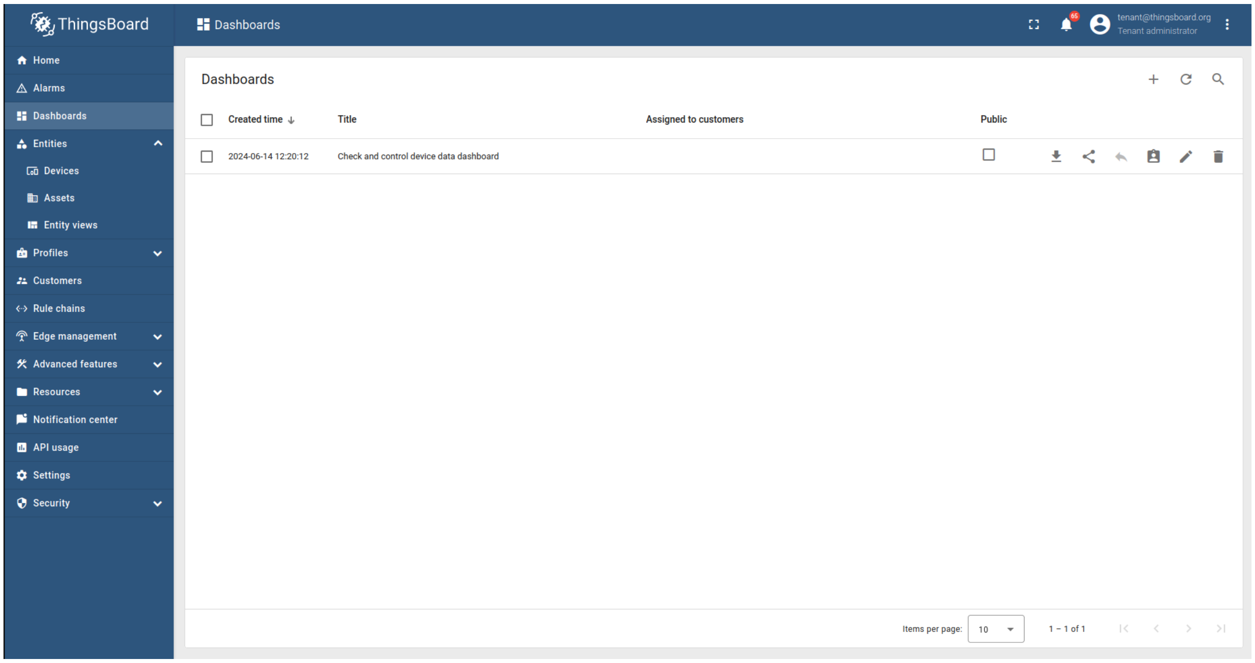


Let's provision your first device to the platform via UI. Follow the documentation on how to do it:

Name	Last viewed
Check and control device data dashboard	week ago

Usage	
Devices	1 / ∞
Assets	0 / ∞
Users	5 / ∞
Dashboards	1 / ∞
Customers	3 / ∞

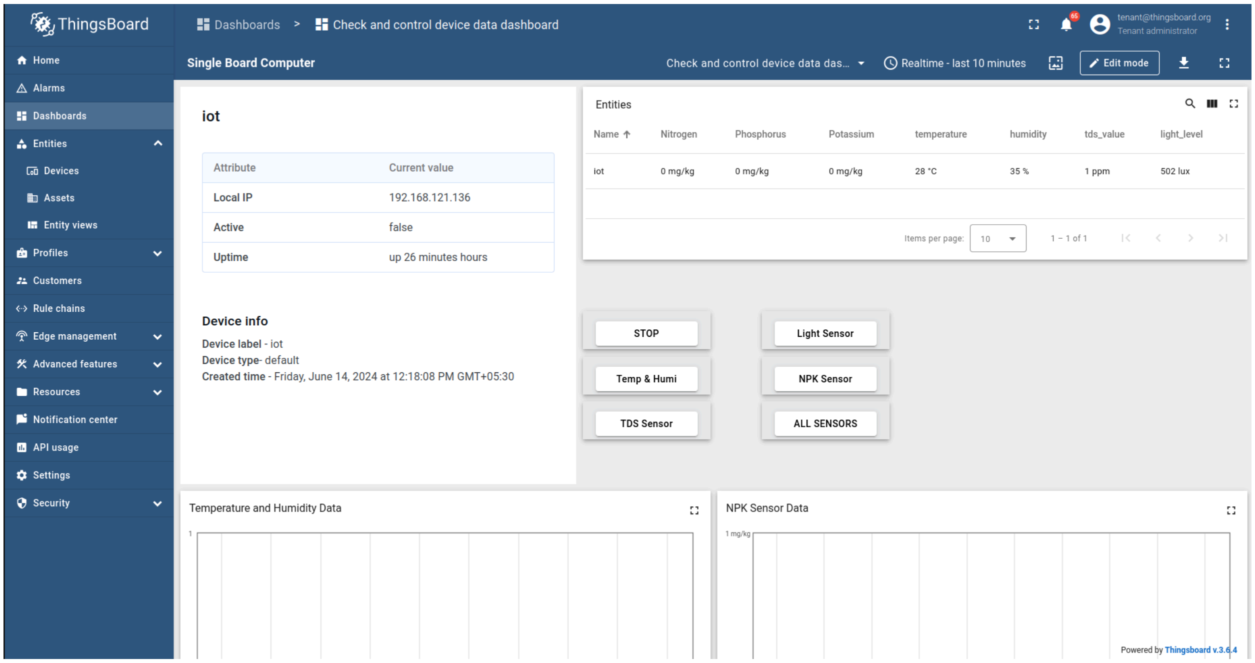
4. Now click on the Mentioned Dashboard in the fields.



Dashboards

□	Created time ↓	Title	Assigned to customers	Public
□	2024-06-14 12:20:12	Check and control device data dashboard		□

5. Acquire Sensor Readings from this Dashboard via pressing the buttons assigned to the mentioned sensors in the Dashboard.



iot

Attribute	Current value
Local IP	192.168.121.136
Active	false
Uptime	up 26 minutes hours

Device info

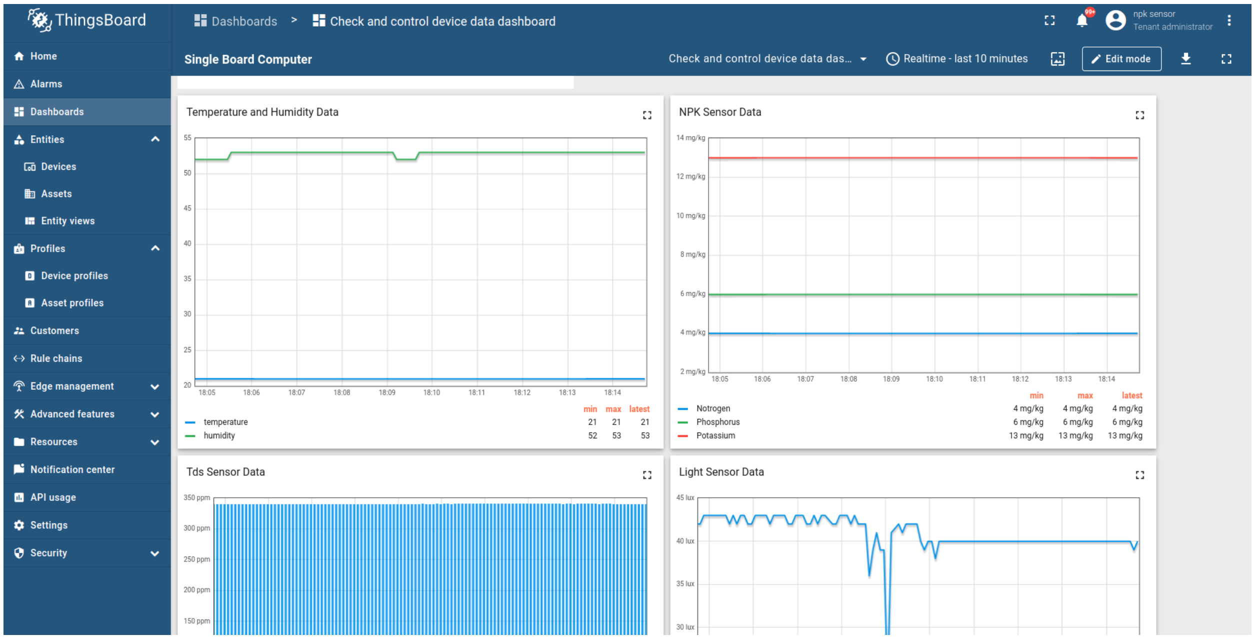
Device label - iot
Device type- default
Created time - Friday, June 14, 2024 at 12:18:08 PM GMT+05:30

Entities

Name ↑	Nitrogen	Phosphorus	Potassium	temperature	humidity	tds_value	light_level
iot	0 mg/kg	0 mg/kg	0 mg/kg	28 °C	35 %	1 ppm	502 lux

Items per page: 10 1 – 1 of 1

The sensor data will be observable in the dashboard.

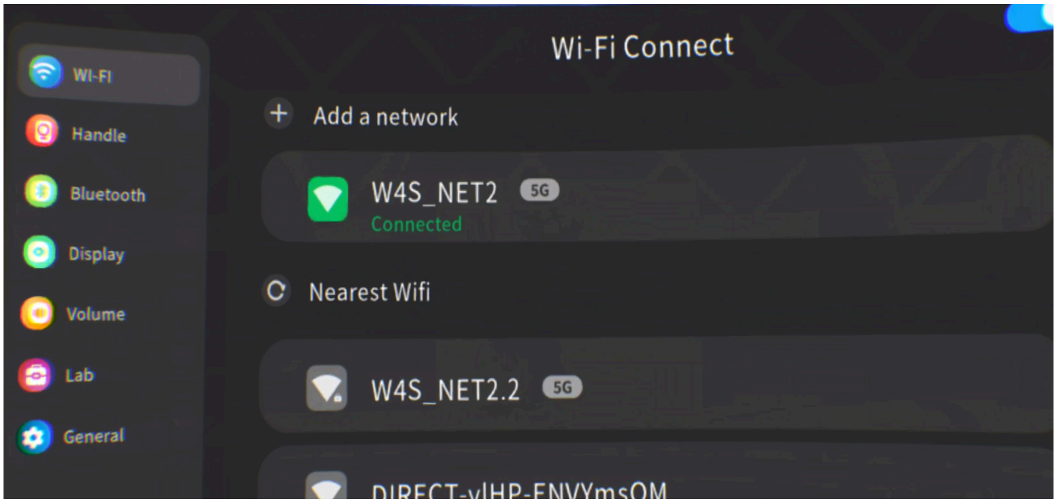


	min	max	latest
temperature	21	21	21
humidity	52	53	53

	min	max	latest
Notrogen	4 mg/kg	4 mg/kg	4 mg/kg
Phosphorus	6 mg/kg	6 mg/kg	6 mg/kg
Potassium	13 mg/kg	13 mg/kg	13 mg/kg

17.5. Configuration Steps for AR/VR

1. Power on your headset after wearing it.
2. Navigate to settings→wifi
3. Choose wifi i.e. from IoT gateway, cpe, evaluation board etc.(anyone of them)



4. Open resource center and run provided inbuilt application. Enjoy AR/VR experience.

17.6. Configuration Steps for iDronam

1. Charge the battery in LiPo 3A configuration till it reaches 100%
2. Insert the nano SIM card with the chip facing upwards.
3. Place the drone in an open space with a minimum area of 3m x 3m
4. Insert the charged battery into the drone.
5. Pre-requisites:

   Core and Ran should be already running.

6. Install iDronam App for Linux:

a. Log in to the MEC server using SSH:

```
ssh suadmin@192.168.x.30
```

b. Install the iDronam Enterprise app:

```
sudo snap install idronam-enterprise --beta
```

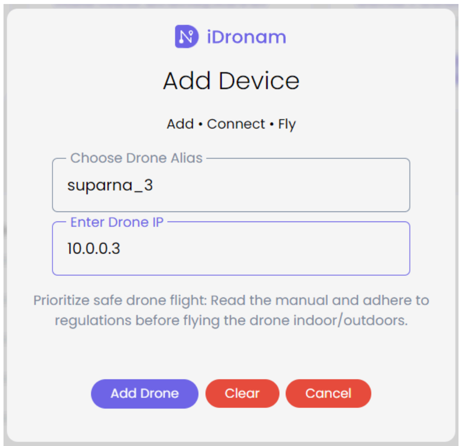
c. Connect the necessary interfaces for joystick functionality:

```
sudo snap connect idronam-enterprise:raw-usb
sudo snap connect idronam-enterprise:joystick
```

d. Open the iDronam Enterprise app:

```
idronam-enterprise
```

7. Use the iDronam App to add the device and then connect the drone.



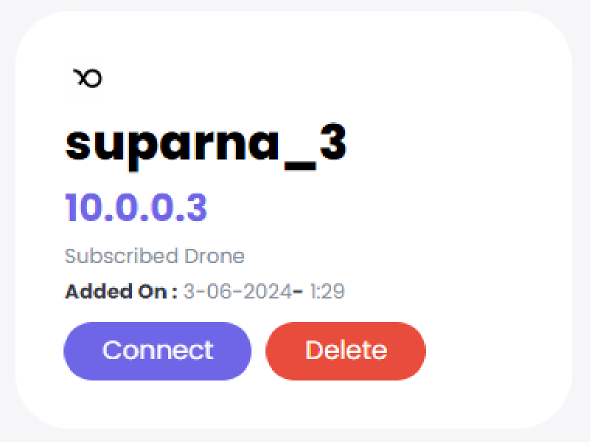
Add Device

Add • Connect • Fly

Choose Drone Alias: suparna_3

Enter Drone IP: 10.0.0.3

Prioritize safe drone flight: Read the manual and adhere to regulations before flying the drone indoor/outdoors.

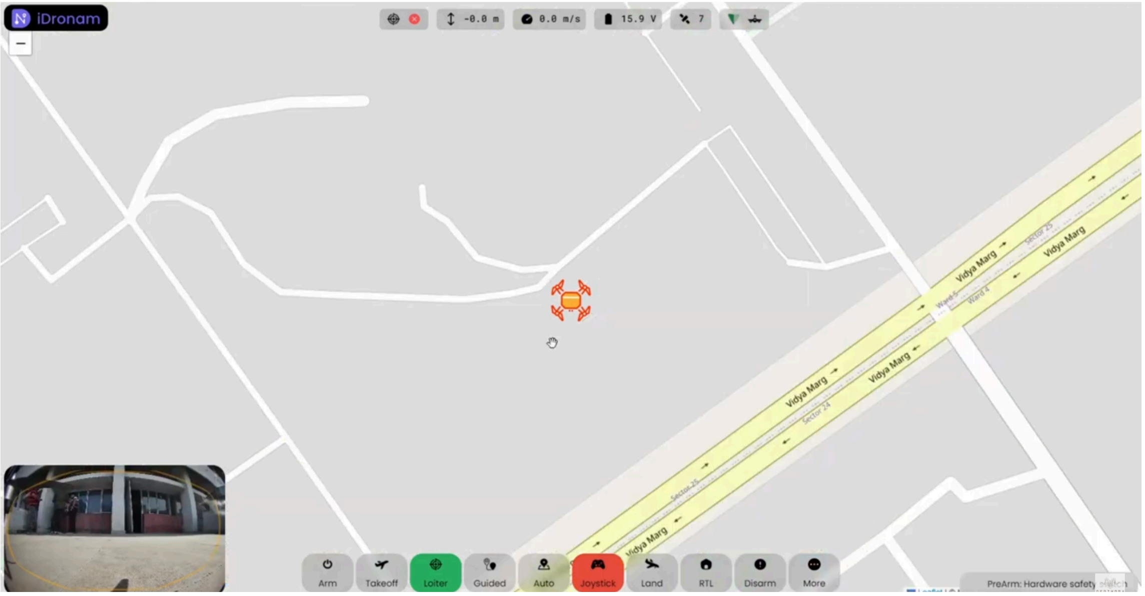

suparna_3

10.0.0.3

Subscribed Drone

Added On : 3-06-2024- 1:29

8. See the drone telemetry data and live feed at the bottom left of the screen.



9. Go to the "More Options" menu in the iDronam Software

    1. Turn on Indoor Mode
    2. Turn off Fencing
    3. Turn on Avoidance to enable obstacle detection and avoidance
    4. Turn off safety

10. Arm the drone by using the joystick on your remote control

    1. Hold the left joystick down
    2. Press the A button to arm the drone. You should hear a confirmation sound or see a light indicating the drone is armed

11. Control the drone by moving both joysticks

    1. To increase or decrease the altitude, push the left joystick up or down
    2. To move in the horizontal plane, push the right joystick in that direction

12. To land the drone, always press the X button on your joystick and never push the left of the joystick down.

13. In case of an emergency, immediately push the left joystick down and press the B button to disarm the drone and stop the motors.

17.7. Configuration Steps for Sparsh Camera

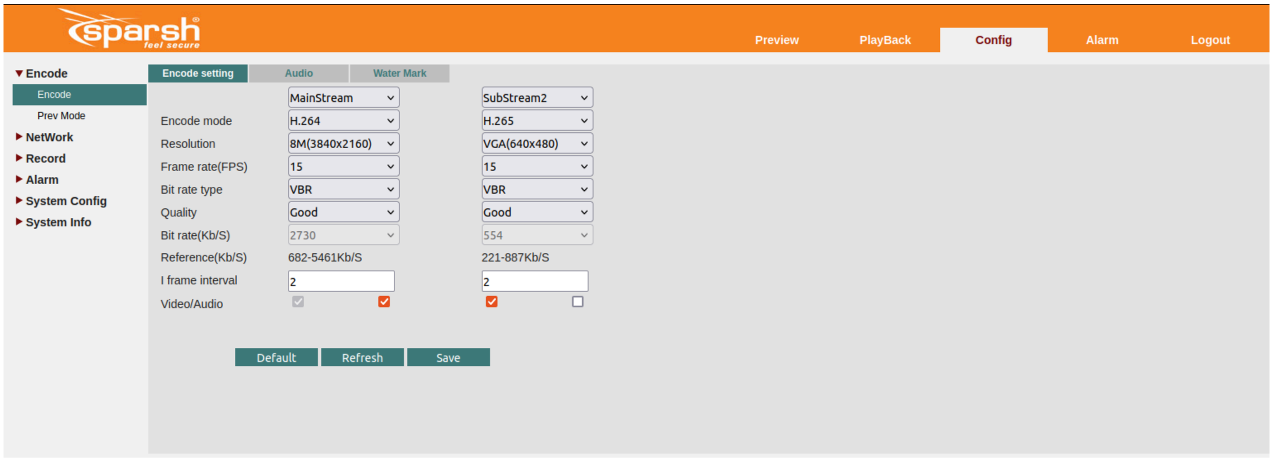
15.7.1 Steps for setting features:

1. Connect the camera to a laptop or pc after powering it on via an ethernet cable or USB (port can be accessed after opening the lid of the camera)
2. Open Firefox web browser and go to the Cameras web gui by inputting the following ip in the web browser : 192.168.128.10
3. Username - admin

Password - admin123

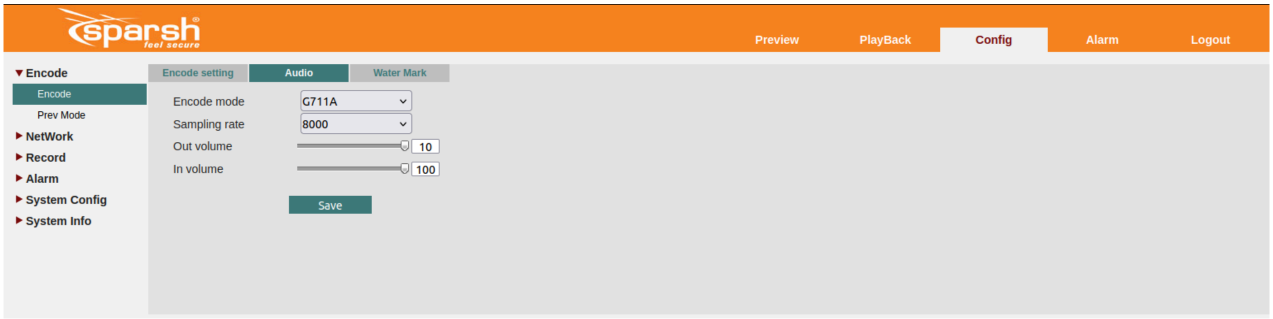
4. Check following settings as per your requirement

Encode Setting



	MainStream	SubStream2
Encode mode	H.264	H.265
Resolution	8M(3840x2160)	VGA(640x480)
Frame rate(FPS)	15	15
Bit rate type	VBR	VBR
Quality	Good	Good
Bit rate(Kb/S)	2730	554
Reference(Kb/S)	682-5461Kb/S	221-887Kb/S
I frame interval	2	2
Video/Audio	☑ ☑	☑ ☐

Audio



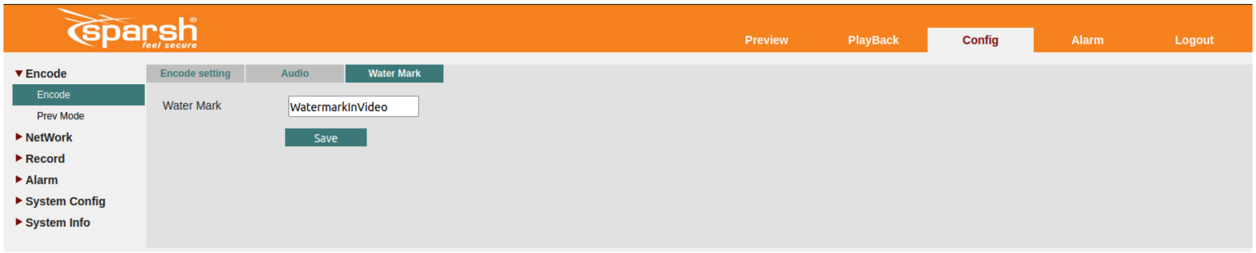
Encode mode G711A

Sampling rate 8000

Out volume 10

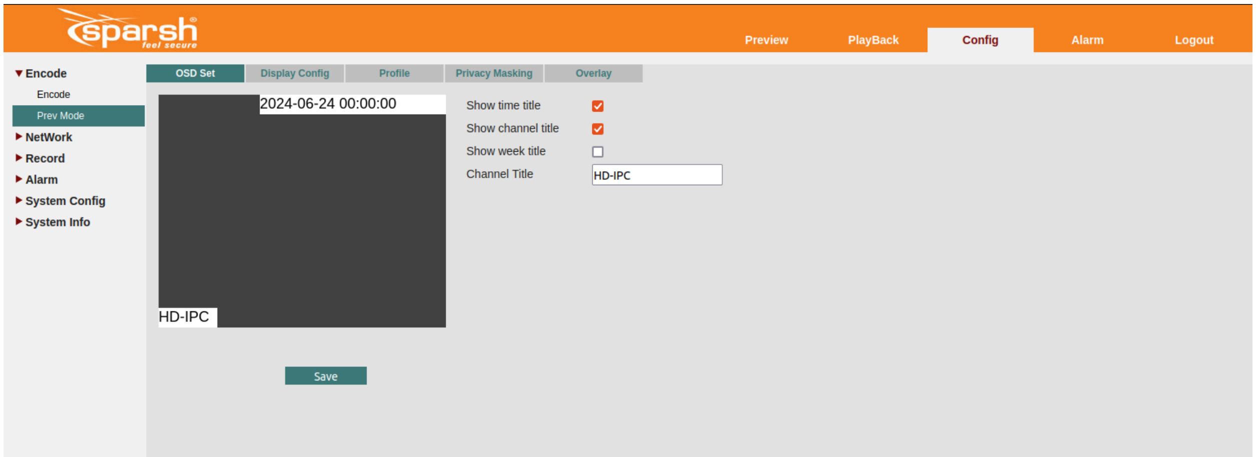
In volume 100

Watermark



Water Mark: WatermarkInVideo

OSD Set



Show time title ☑

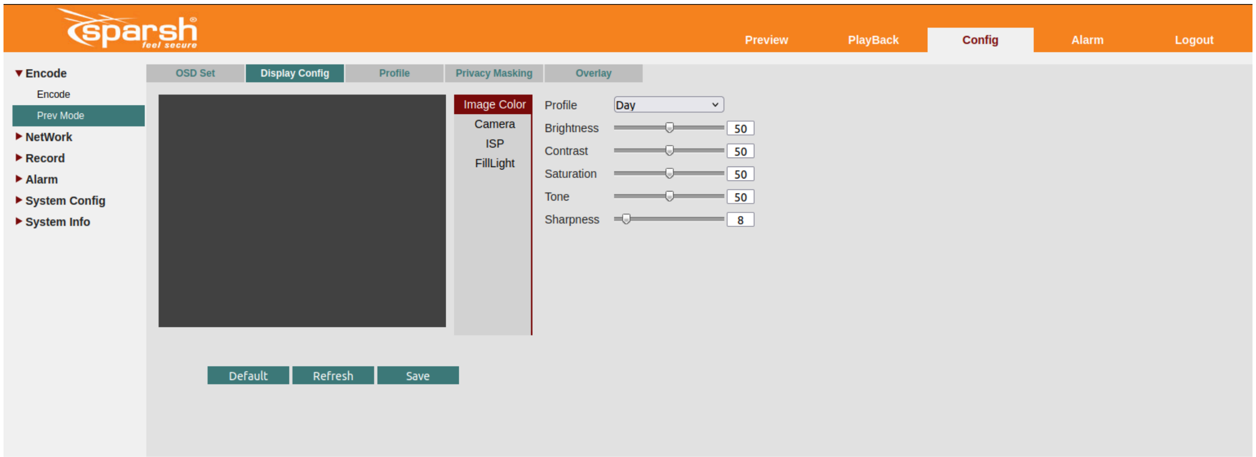
Show channel title ☑

Show week title □

Channel Title: HD-IPC

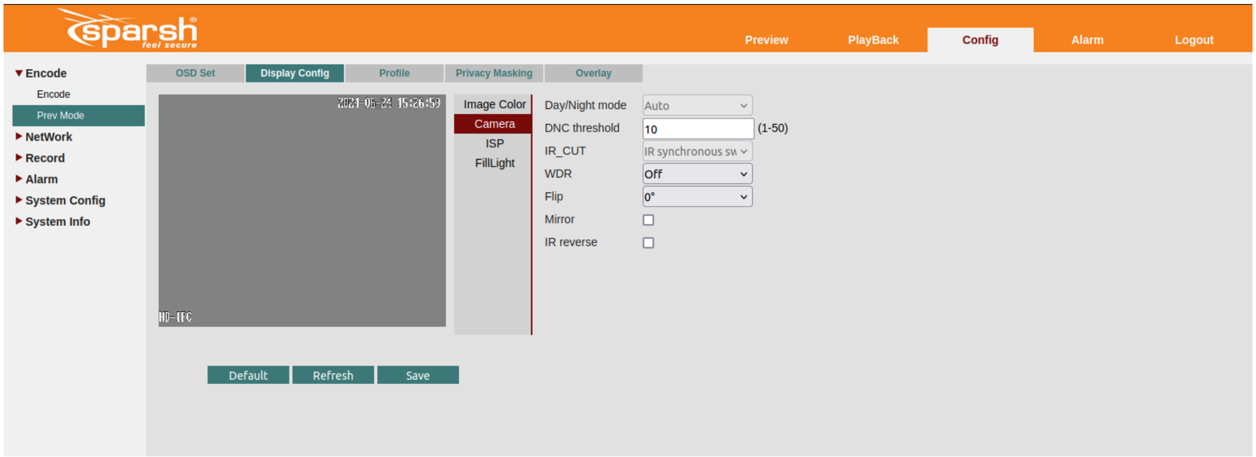
Display Config

Image Color



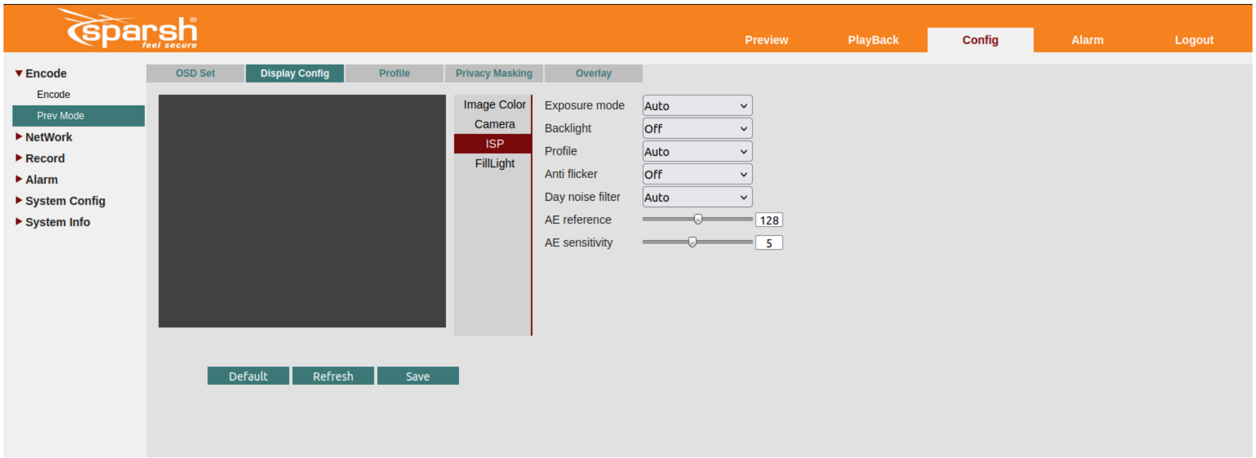
Profile	Day
Brightness	50
Contrast	50
Saturation	50
Tone	50
Sharpness	8

Camera



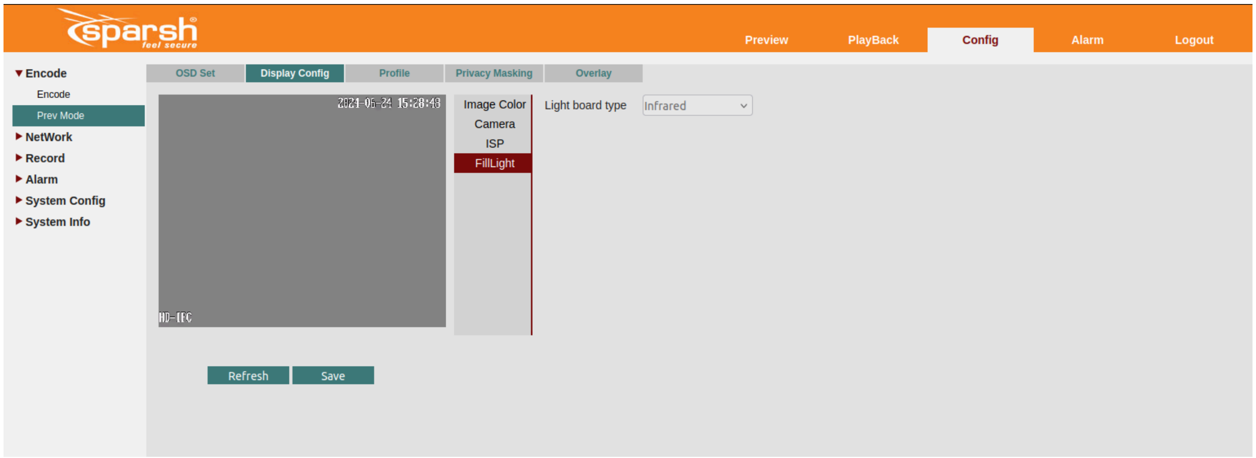
Day/Night mode: Auto
DNC threshold: 10 (1-50)
IR_CUT: IR synchronous sw
WDR: Off
Flip: 0°
Mirror: ☐
IR reverse: ☐

ISP



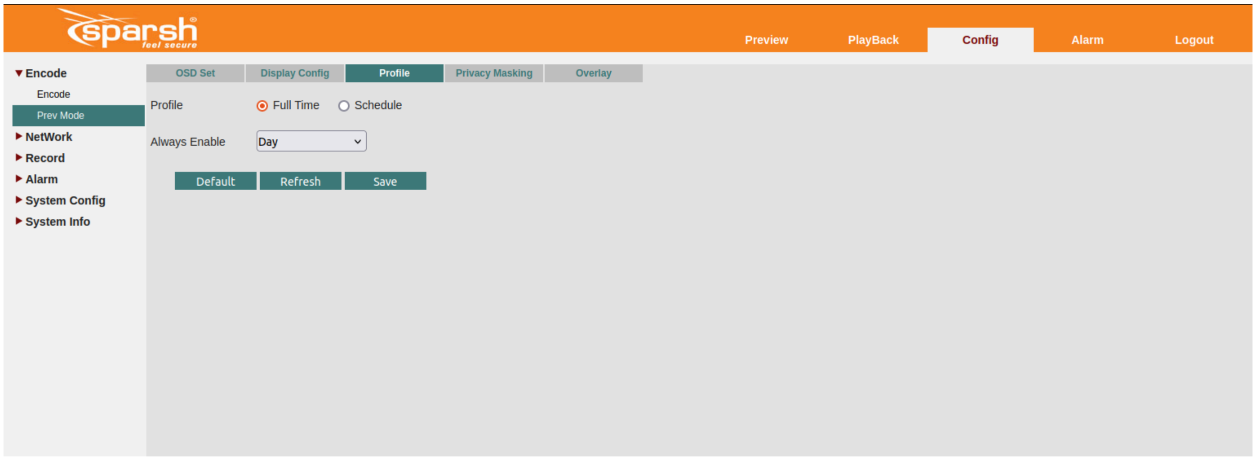
Exposure mode: Auto
Backlight: Off
Profile: Auto
Anti flicker: Off
Day noise filter: Auto
AE reference: 128
AE sensitivity: 5

FillLight



Light board type: Infrared

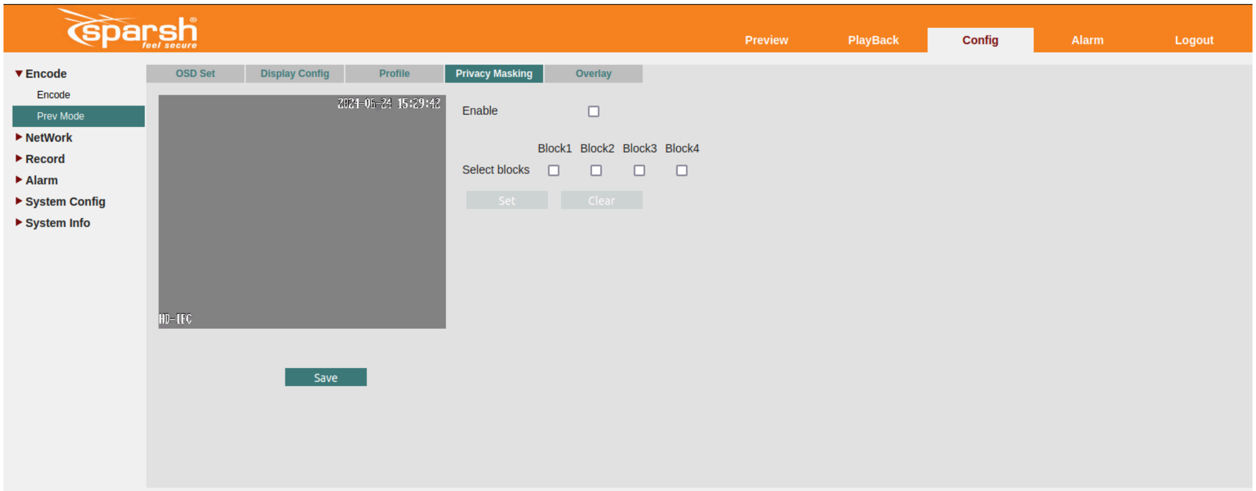
Profile



Profile ● Full Time ○ Schedule

Always Enable Day

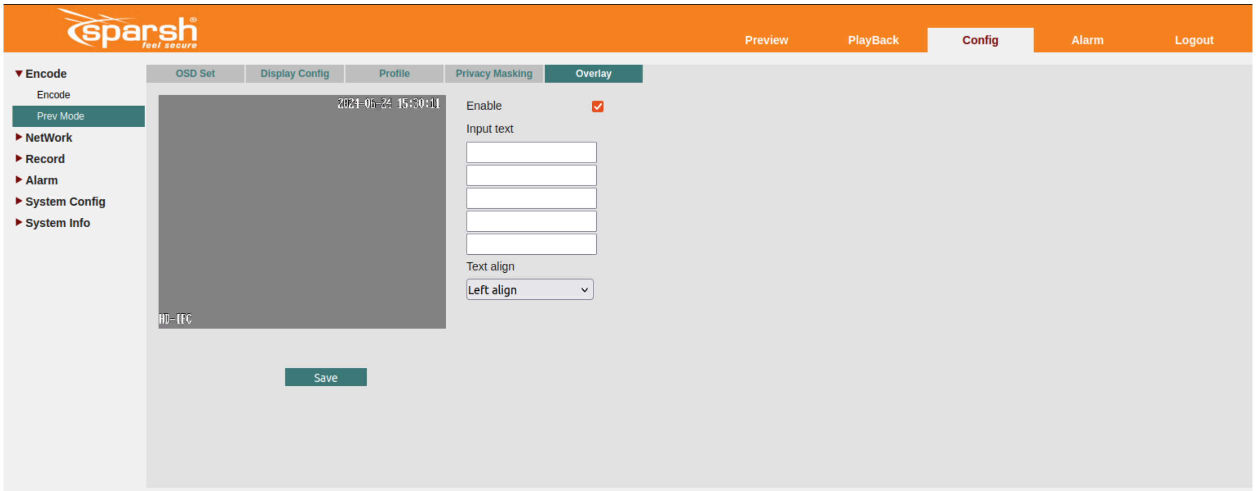
Privacy Masking



Enable □

	Block1	Block2	Block3	Block4
Select blocks	□	□	□	□

Overlay



Enable ☑

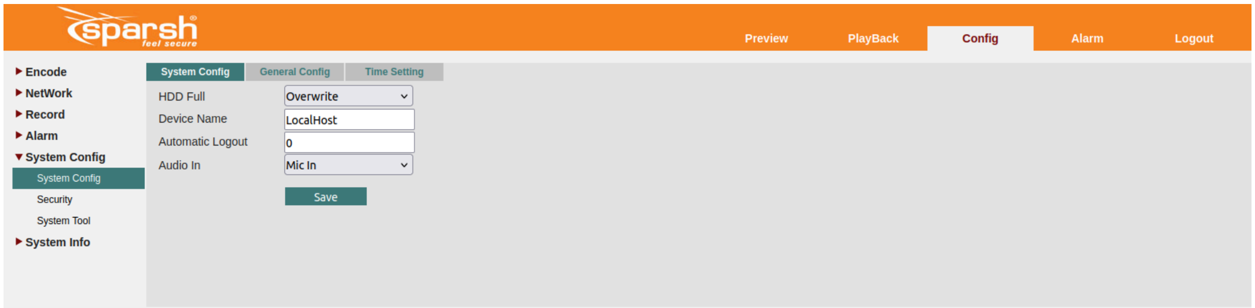
Input text

Text align

Left align

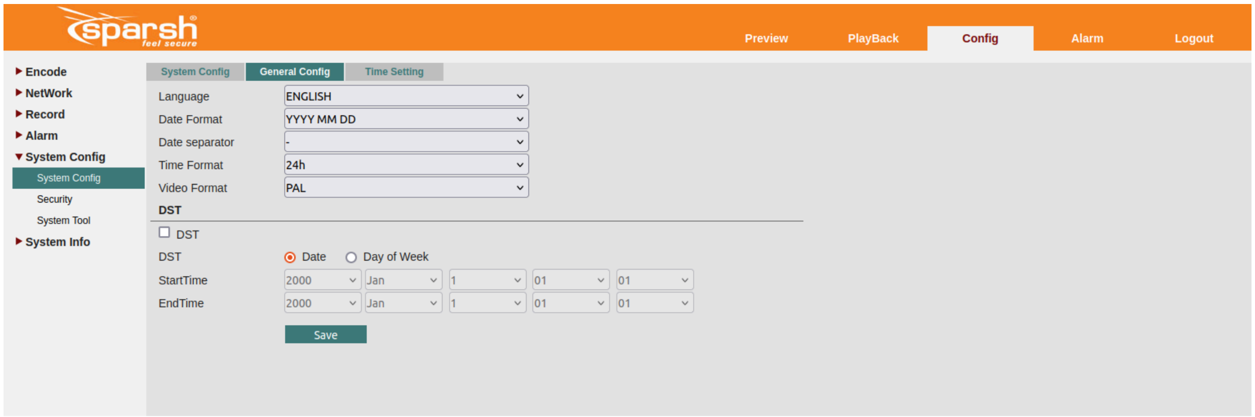
System Config

System Config



HDD Full: Overwrite
Device Name: LocalHost
Automatic Logout: 0
Audio In: Mic In

General Config



Language: ENGLISH
Date Format: YYYY MM DD
Date separator: -
Time Format: 24h
Video Format: PAL

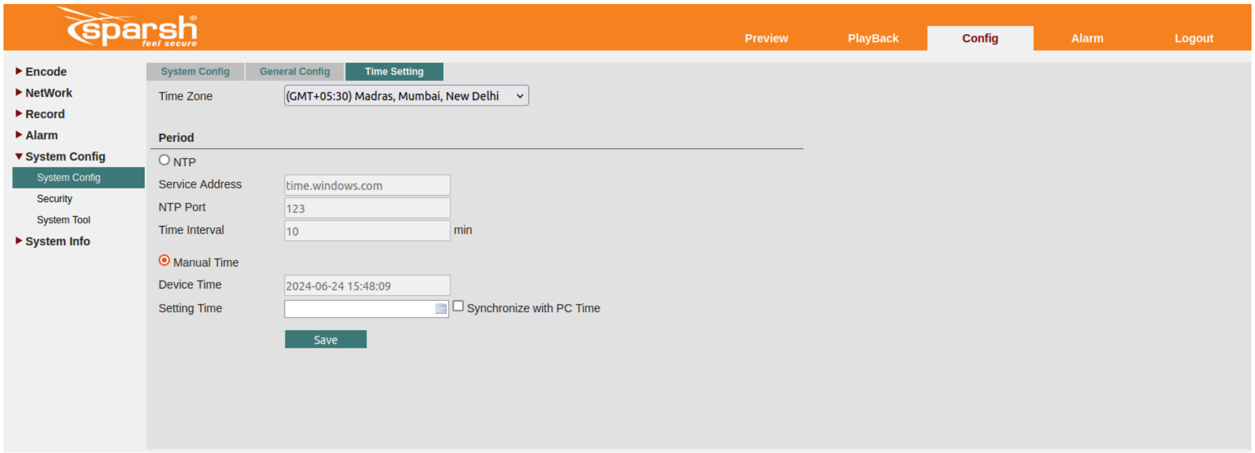
DST

□ DST

DST: ● Date ○ Day of Week

StartTime: 2000 Jan 1 01 01
EndTime: 2000 Jan 1 01 01

Time Setting



Time Zone: (GMT+05:30) Madras, Mumbai, New Delhi

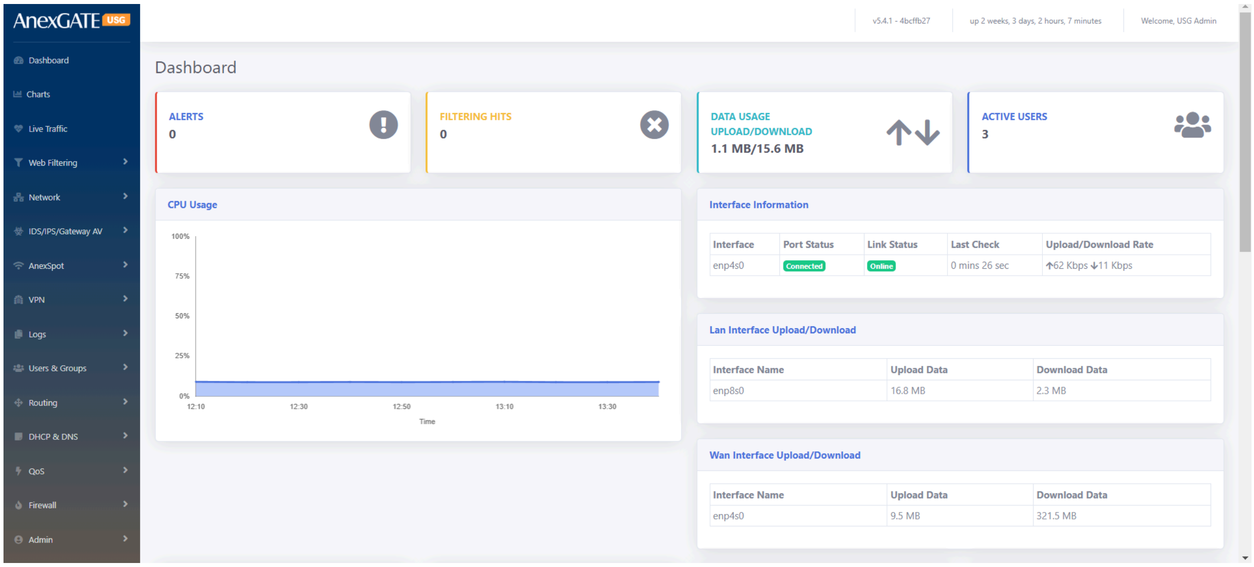
Period

○ NTP
Service Address: time.windows.com
NTP Port: 123
Time Interval: 10 min

● Manual Time
Device Time: 2024-06-24 15:48:09
Setting Time: □ Synchronize with PC Time

18. Firewall Configuration

1. Copy and paste the appropriate IP address into your web browser to log in.
2. After opening the Anexgate site on the browser login with respective Username and password.



ALERTS 0

FILTERING HITS 0

DATA USAGE UPLOAD/DOWNLOAD 1.1 MB/15.6 MB

ACTIVE USERS 3

Interface Information

Interface	Port Status	Link Status	Last Check	Upload/Download Rate
enp4s0	Connected	Online	0 mins 26 sec	↑62 Kbps ↓11 Kbps

Lan Interface Upload/Download

Interface Name	Upload Data	Download Data
enp8s0	16.8 MB	2.3 MB

Wan Interface Upload/Download

Interface Name	Upload Data	Download Data
enp4s0	9.5 MB	321.5 MB

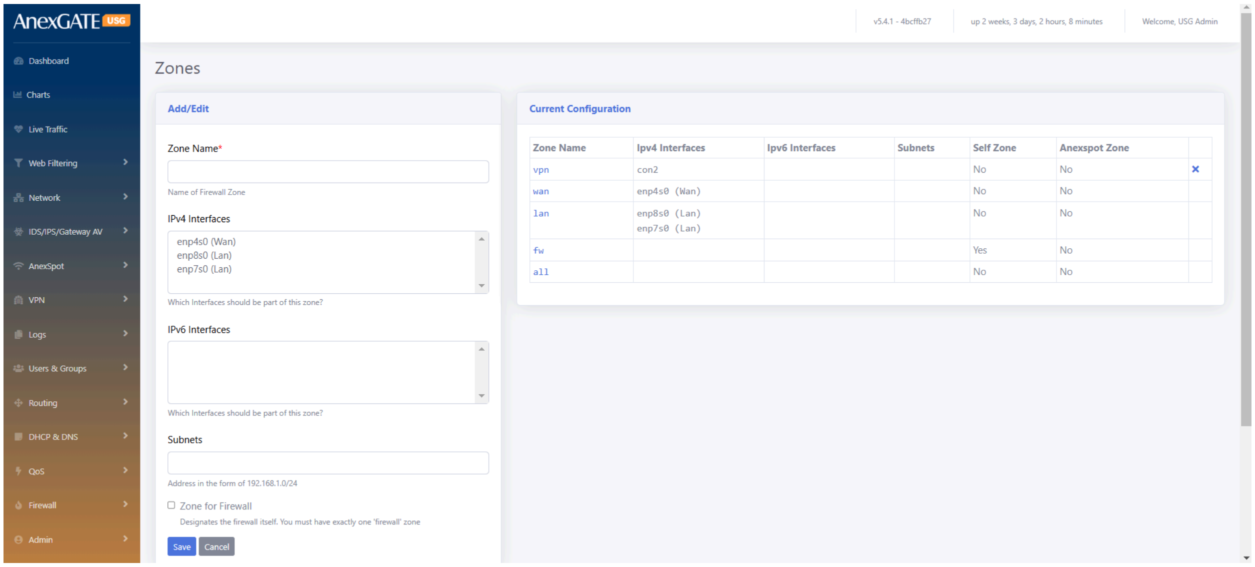
18.1. Zones

To configure Firewall Zones, Go to **Firewall** → **Zones.**

Configuration Parameters

ELEMENTS	DESCRIPTION
Zone Name	Enter the name of the zone for the interfaces you want to assign.
Interfaces	Select the physical interfaces (WAN-enp1s0, LAN-enp2s0 etc..)
Subnets	Enter the network subnet of the interface if required to specify. (Example: 192.168.100.0/24)

Zone for Firewall	Select this if the Zone is to be made for Firewall.
Allow AnexSPOT	Allows AnexSPOT to work for the Zone.
SAVE	Save the settings to configure the Zone.



Zones

Add/Edit

Zone Name*

Name of Firewall Zone

IPv4 Interfaces

enp4s0 (Wan)
enp8s0 (Lan)
enp7s0 (Lan)

Which Interfaces should be part of this zone?

IPv6 Interfaces

Which Interfaces should be part of this zone?

Subnets

Address in the form of 192.168.1.0/24

□ Zone for Firewall

Designates the firewall itself. You must have exactly one 'firewall' zone

Save Cancel

Current Configuration

Zone Name	Ipv4 Interfaces	Ipv6 Interfaces	Subnets	Self Zone	Anexspot Zone	
vpn	con2			No	No	×
wan	enp4s0 (Wan)			No	No	
lan	enp8s0 (Lan) enp7s0 (Lan)			No	No	
fw				Yes	No	
all				No	No	

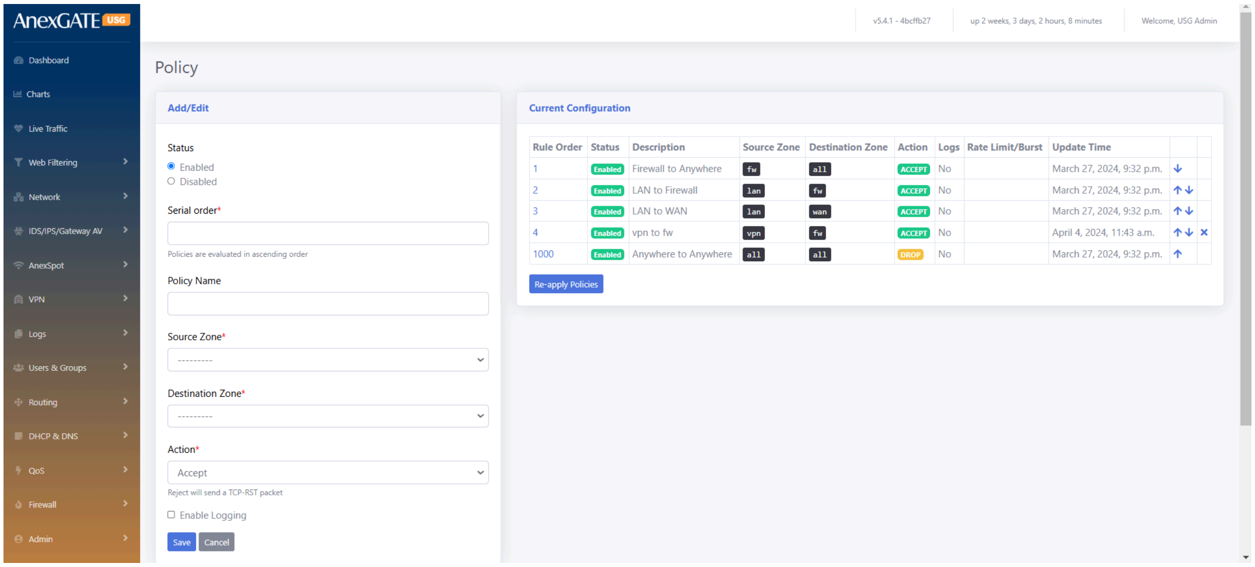
18.2. Configuration for Policy

To configure Firewall Policy, Go to **Firewall** → **Policy.**

Configuration Parameters

ELEMENTS	DESCRIPTION
Status	Enable or Disable the policy.
Serial Order	Enter a serial number to categorise policies.
Policy Name	Enter the name of the Firewall Policy.
Source Zone	Select the zone from the dropdown list configured in Firewall >Zones as the Source Zone.

Destination Zone	Select the zone from the dropdown list configured in Firewall >Zones as the Destination Zone.
Action	• ACCEPT – Select this to allow communication between Zones. • DROP – Select this to drop communication between Zones. • REJECT – Select this to drop as well as send a TCP-RST packet to the sender.
Enable Logging	Select this to get logs generated by the Firewall Policy.
SAVE	Save the configuration settings of the Firewall policy.



Policy

Add/Edit

Status

● Enabled
○ Disabled

Serial order*

Policies are evaluated in ascending order

Policy Name

Source Zone*

Destination Zone*

Action*

Accept

Reject will send a TCP-RST packet

□ Enable Logging

Save Cancel

Current Configuration

Rule Order	Status	Description	Source Zone	Destination Zone	Action	Logs	Rate Limit/Burst	Update Time		
1	Enabled	Firewall to Anywhere	fw	all	ACCEPT	No		March 27, 2024, 9:32 p.m.	↓	
2	Enabled	LAN to Firewall	lan	fw	ACCEPT	No		March 27, 2024, 9:32 p.m.	↑↓	
3	Enabled	LAN to WAN	lan	wan	ACCEPT	No		March 27, 2024, 9:32 p.m.	↑↓	
4	Enabled	vpn to fw	vpn	fw	ACCEPT	No		April 4, 2024, 11:43 a.m.	↑↓	×
1000	Enabled	Anywhere to Anywhere	all	all	DROP	No		March 27, 2024, 9:32 p.m.	↑	

Re-apply Policies

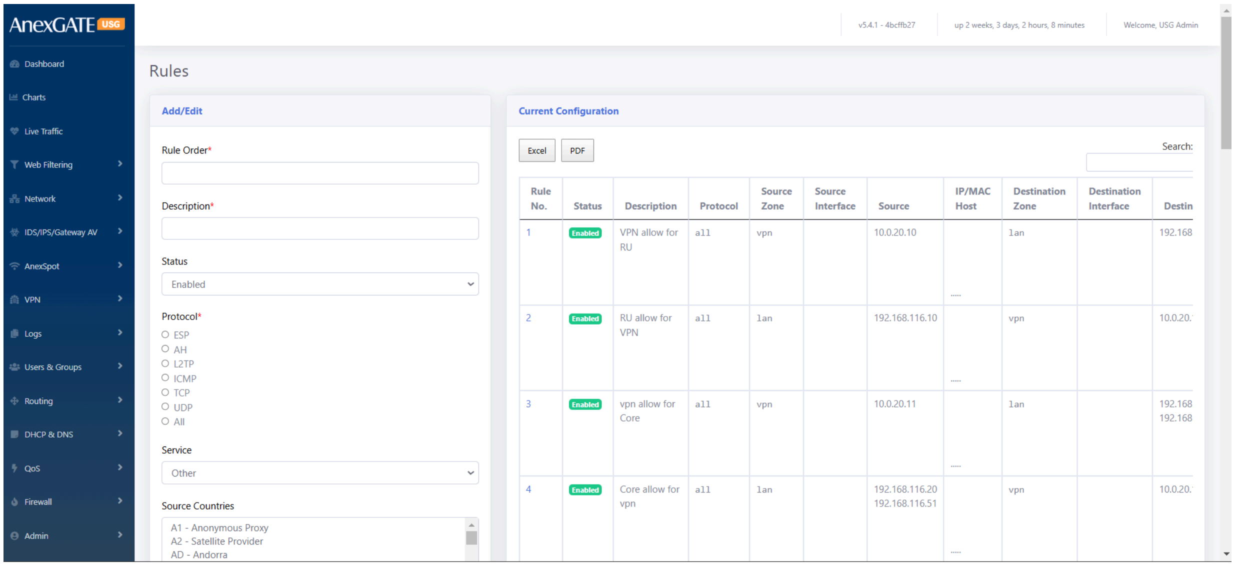
18.3. Rules

To configure Firewall Rules, Go to **Firewall→Rules.**

Configuration Parameters

ELEMENTS	DESCRIPTION
Rule Order	Enter Rule Order in terms of numerical order.
Description	Add a description to the Firewall rule
Status	Enable or Disable status of the Firewall rule.
Protocol	Select Network Protocol (TCP, UDP, ICMP etc..) to allow or drop traffic.
Service	Select Services from Dropdown list to apply the firewall rule.
Source Countries	Select Source country for the firewall rule to allow or drop traffic.
Source Zone	Select Source Zone from dropdown list to which the rule applies.
Source Interface	Select Source Interface if there are multiple interfaces and need only a single interface to which the rule applies.
Source	Enter Souce IP or MAC of the host to which the rule applies.
Source Port	Enter Source Port of the host to which the rule applies.
Destination Countries	Select Destination Countries for the firewall rule to allow or drop traffic.
Destination Zone	Select Destination Zone from dropdown list to which the rule applies.
Destination Interface	Select Destination Interface if there are multiple interfaces and need only a single interface to which the rule applies.

Destination	Enter Destination of the host to which the rule gets applied.
Destination Port	Enter Destination Port of the host to which the rule gets applied.
What to do?	Options to choose from: • Accept – Allows access • Drop – Discards access • Reject – Denies Access and ICMP Port Unreachable will be sent to the source.
Time Based Rule	Enable or Disable to schedule the rule.
From/To Date	Enter From/To Date for the rule to be applied.
From/To Time	Enter From/To Time for the rule to be applied.
Days of Week	Select Days in a week for the rule to be applied.
AnexSPOT Bypass	Click to Bypass AnexSPOT restriction for rule.
Proxy Bypass	Click to Bypass Proxy if proxy server is enabled.
Enable Logging	Click to enable traffic logging for the rule i.e, traffic allowed or dropped by the Firewall rule.
Keywords	Assign a keyword to comply it to a firewall rule.
SAVE	Save the configuration settings for the firewall rule.



Rules

Rule Order*

Description*

Status: Enabled

Protocol*
○ ESP
○ AH
○ L2TP
○ ICMP
○ TCP
○ UDP
○ All

Service: Other

Source Countries
A1 - Anonymous Proxy
A2 - Satellite Provider
AD - Andorra

Rule No.	Status	Description	Protocol	Source Zone	Source Interface	Source	IP/MAC Host	Destination Zone	Destination Interface	Destin
1	Enabled	VPN allow for RU	all	vpn		10.0.20.10	.....	lan		192.168
2	Enabled	RU allow for VPN	all	lan		192.168.116.10	.....	vpn		10.0.20.
3	Enabled	vpn allow for Core	all	vpn		10.0.20.11	.....	lan		192.168 192.168
4	Enabled	Core allow for vpn	all	lan		192.168.116.20 192.168.116.51	.....	vpn		10.0.20.

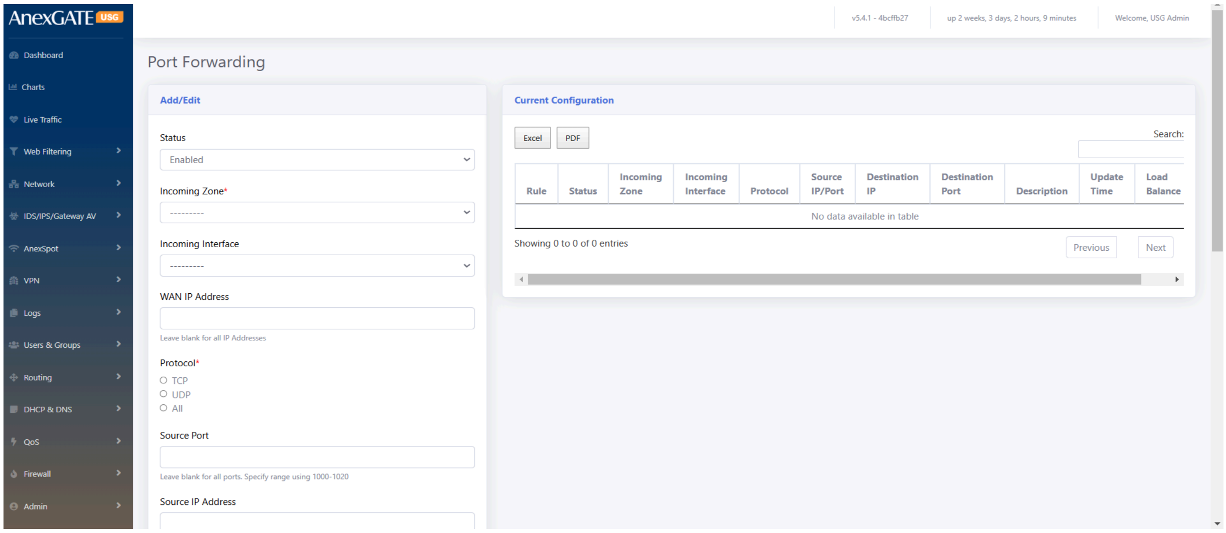
18.4. Port Forwarding

To configure Port Forwarding, Go to **Firewall → Port Forwarding.**

Configuration Parameters

ELEMENTS	DESCRIPTION
Status	Enable or Disable the Port Forwarding rule.
Incoming Zone*	Select WAN as Incoming Zone. (To do Hair Pin NAT, select LAN as Incoming Zone)
Incoming Interface	Select WAN interface if there are multiple interfaces. Leaving this parameter blank will be considered as to include all WAN interfaces configured.
WAN IP Address	Specify WAN IP Address for Port Forwarding if there is no interface selected.
Protocol	Select Transport Layer Protocol TCP/UDP or ALL protocols to be

	allowed.
Source Port	Specify Port Number for TCP/UDP protocol (0-65535).
Source IP Address	Enter Source IP Address to be Port Forwarded from.
Destination Zone*	Select LAN Zone.
Destination IP Address	Select LAN IP for port forwarding.
Destination Port	Specify Specific Port Number to listen on.
Service	Select the service from the list. (Example: FTP, PPTP, SNMP,TFTP)
Hairpin NAT	Select this option to allow the internal network hosts to reach the port forwarded services using the public IP, locally.
Enable Logging	Click to enable traffic logging for the rule i.e, traffic allowed or dropped. Logs can be seen at, Logs > Traffic Logs > NAT Table
Description	Add a description to the Port Forwarding rule.
SAVE	Save the configuration settings of the Port Forwarding rule.



Port Forwarding

Add/Edit

Status: Enabled

Incoming Zone*: ---------

Incoming Interface: ---------

WAN IP Address:

Leave blank for all IP Addresses

Protocol*

○ TCP
○ UDP
○ All

Source Port:

Leave blank for all ports. Specify range using 1000-1020

Source IP Address:

Current Configuration

Excel | PDF | Search:

Rule	Status	Incoming Zone	Incoming Interface	Protocol	Source IP/Port	Destination IP	Destination Port	Description	Update Time	Load Balance
No data available in table										

Showing 0 to 0 of 0 entries

Previous | Next

19. Troubleshooting for CORE

19.1. Logs for CORE

Steps for Capturing Logs for CORE:

1. Open the terminal window in MEC.

2.Log in into Core: ssh suadmin@192.168.xx.20(xx refers to subnet based on institute)
Password: Admin@123

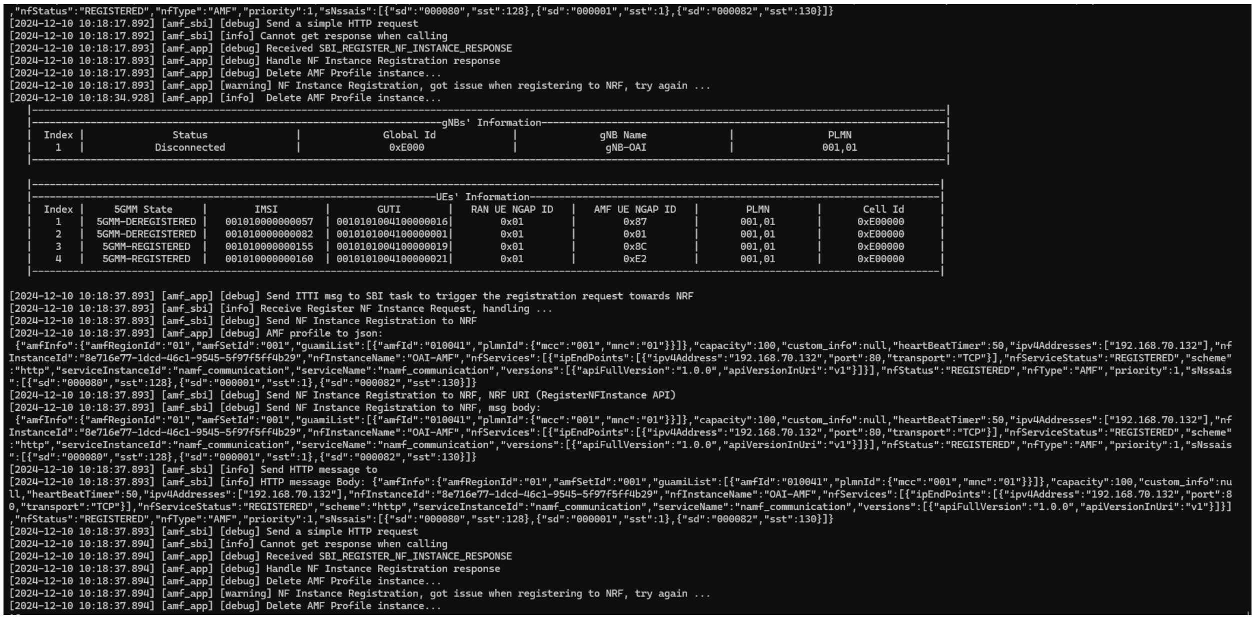
3. Change the directory using **cd w4s-directory** command.

4. To start the core services execute the **docker compose docker-compose-w4s.yaml up -d** command.

5 . Check the status of all the services using **docker ps** command.

6. Execute **docker logs -f w4s-amf** command to view the logs.

7. To save logs ; Run Docker logs w4s-amf>[file name which needs to be saved]



```
,"nfStatus":"REGISTERED","nfType":"AMF","priority":1,"sNssais":[{"sd":"000080","sst":128},{"sd":"000001","sst":1},{"sd":"000082","sst":130}]}
[2024-12-10 10:18:17.892] [amf_sbi] [debug] Send a simple HTTP request
[2024-12-10 10:18:17.892] [amf_sbi] [info] Cannot get response when calling
[2024-12-10 10:18:17.893] [amf_app] [debug] Received SBI_REGISTER_NF_INSTANCE_RESPONSE
[2024-12-10 10:18:17.893] [amf_app] [debug] Handle NF Instance Registration response
[2024-12-10 10:18:17.893] [amf_app] [debug] Delete AMF Profile instance...
[2024-12-10 10:18:17.893] [amf_app] [warning] NF Instance Registration, got issue when registering to NRF, try again ...
[2024-12-10 10:18:34.928] [amf_app] [info]  Delete AMF Profile instance...
   |----------------------------------------------------------------------------------------------------------------------------------------------------|
   |----------------------------------------------------------------------gNBs' Information----------------------------------------------------------------------|
   | Index |              Status              |              Global Id              |              gNB Name              |              PLMN              |
   |   1   |           Disconnected           |                0xE000               |               gNB-OAI              |             001,01             |
   |----------------------------------------------------------------------------------------------------------------------------------------------------|

   |---------------------------------------------------------------------------------------------------------------------------------------------------|
   |---------------------------------------------------------------------UEs' Information---------------------------------------------------------------------|
   | Index |      5GMM State      |      IMSI      |      GUTI      |  RAN UE NGAP ID  |  AMF UE NGAP ID  |      PLMN      |      Cell Id      |
   |   1   | 5GMM-DEREGISTERED |  001010000000057 | 0010101004100000016|       0x01       |       0x87       |     001,01     |     0xE00000     |
   |   2   | 5GMM-DEREGISTERED |  001010000000082 | 0010101004100000001|       0x01       |       0x01       |     001,01     |     0xE00000     |
   |   3   |  5GMM-REGISTERED  |  001010000000155 | 0010101004100000019|       0x01       |       0x8C       |     001,01     |     0xE00000     |
   |   4   |  5GMM-REGISTERED  |  001010000000160 | 0010101004100000021|       0x01       |       0xE2       |     001,01     |     0xE00000     |
   |---------------------------------------------------------------------------------------------------------------------------------------------------|

[2024-12-10 10:18:37.893] [amf_app] [debug] Send ITTI msg to SBI task to trigger the registration request towards NRF
[2024-12-10 10:18:37.893] [amf_sbi] [info] Receive Register NF Instance Request, handling ...
[2024-12-10 10:18:37.893] [amf_sbi] [debug] Send NF Instance Registration to NRF
[2024-12-10 10:18:37.893] [amf_app] [debug] AMF profile to json:
 {"amfInfo":{"amfRegionId":"01","amfSetId":"001","guamiList":[{"amfId":"010041","plmnId":{"mcc":"001","mnc":"01"}}]},"capacity":100,"custom_info":null,"heartBeatTimer":50,"ipv4Addresses":["192.168.70.132"],"nf
InstanceId":"8e716e77-1dcd-46c1-9545-5f97f5ff4b29","nfInstanceName":"OAI-AMF","nfServices":[{"ipEndPoints":[{"ipv4Address":"192.168.70.132","port":80,"transport":"TCP"}],"nfServiceStatus":"REGISTERED","scheme"
:"http","serviceInstanceId":"namf_communication","serviceName":"namf_communication","versions":[{"apiFullVersion":"1.0.0","apiVersionInUri":"v1"}]}],"nfStatus":"REGISTERED","nfType":"AMF","priority":1,"sNssais
":[{"sd":"000080","sst":128},{"sd":"000001","sst":1},{"sd":"000082","sst":130}]}
[2024-12-10 10:18:37.893] [amf_sbi] [debug] Send NF Instance Registration to NRF, NRF URI (RegisterNFInstance API)
[2024-12-10 10:18:37.893] [amf_sbi] [debug] Send NF Instance Registration to NRF, msg body:
 {"amfInfo":{"amfRegionId":"01","amfSetId":"001","guamiList":[{"amfId":"010041","plmnId":{"mcc":"001","mnc":"01"}}]},"capacity":100,"custom_info":null,"heartBeatTimer":50,"ipv4Addresses":["192.168.70.132"],"nf
InstanceId":"8e716e77-1dcd-46c1-9545-5f97f5ff4b29","nfInstanceName":"OAI-AMF","nfServices":[{"ipEndPoints":[{"ipv4Address":"192.168.70.132","port":80,"transport":"TCP"}],"nfServiceStatus":"REGISTERED","scheme"
:"http","serviceInstanceId":"namf_communication","serviceName":"namf_communication","versions":[{"apiFullVersion":"1.0.0","apiVersionInUri":"v1"}]}],"nfStatus":"REGISTERED","nfType":"AMF","priority":1,"sNssais
":[{"sd":"000080","sst":128},{"sd":"000001","sst":1},{"sd":"000082","sst":130}]}
[2024-12-10 10:18:37.893] [amf_sbi] [info] Send HTTP message to
[2024-12-10 10:18:37.893] [amf_sbi] [info] HTTP message Body: {"amfInfo":{"amfRegionId":"01","amfSetId":"001","guamiList":[{"amfId":"010041","plmnId":{"mcc":"001","mnc":"01"}}]},"capacity":100,"custom_info":nu
ll,"heartBeatTimer":50,"ipv4Addresses":["192.168.70.132"],"nfInstanceId":"8e716e77-1dcd-46c1-9545-5f97f5ff4b29","nfInstanceName":"OAI-AMF","nfServices":[{"ipEndPoints":[{"ipv4Address":"192.168.70.132","port":8
0,"transport":"TCP"}],"nfServiceStatus":"REGISTERED","scheme":"http","serviceInstanceId":"namf_communication","serviceName":"namf_communication","versions":[{"apiFullVersion":"1.0.0","apiVersionInUri":"v1"}]}]
,"nfStatus":"REGISTERED","nfType":"AMF","priority":1,"sNssais":[{"sd":"000080","sst":128},{"sd":"000001","sst":1},{"sd":"000082","sst":130}]}
[2024-12-10 10:18:37.893] [amf_sbi] [debug] Send a simple HTTP request
[2024-12-10 10:18:37.894] [amf_sbi] [info] Cannot get response when calling
[2024-12-10 10:18:37.894] [amf_app] [debug] Received SBI_REGISTER_NF_INSTANCE_RESPONSE
[2024-12-10 10:18:37.894] [amf_app] [debug] Handle NF Instance Registration response
[2024-12-10 10:18:37.894] [amf_app] [debug] Delete AMF Profile instance...
[2024-12-10 10:18:37.894] [amf_app] [warning] NF Instance Registration, got issue when registering to NRF, try again ...
[2024-12-10 10:18:37.894] [amf_app] [debug] Delete AMF Profile instance...
```

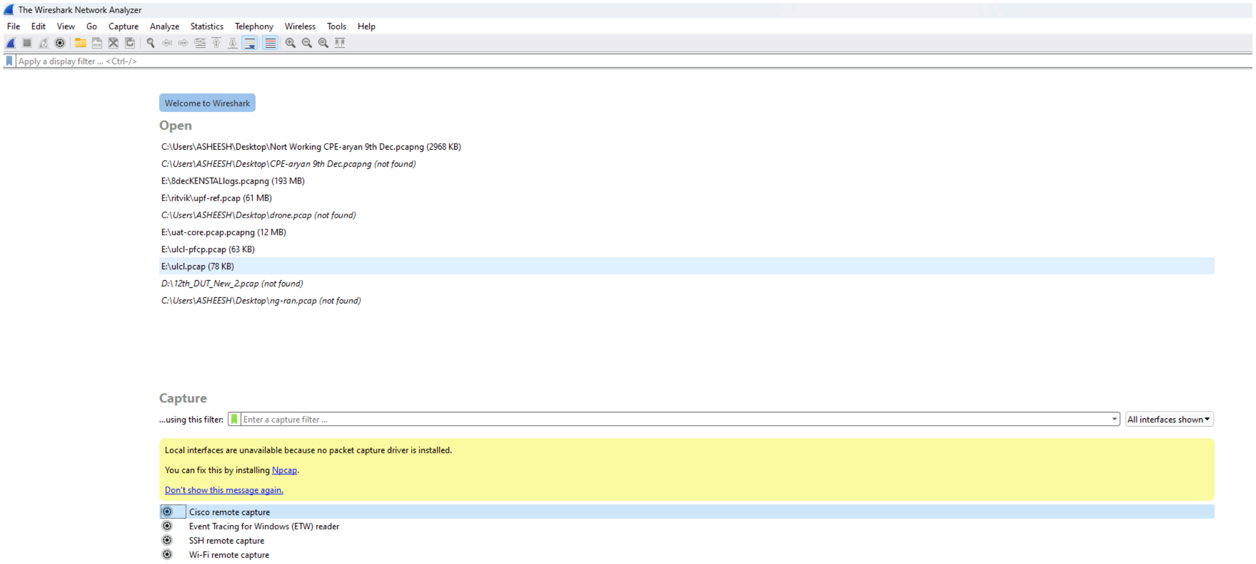
NOTE: Please refer to Solutions manual for more Troubleshooting solutions for CORE

20. Troubleshooting for RAN

20.1. Configuration of Wireshark for capturing RAN Logs:

Steps to configure the Wireshark before capturing RAN Logs:

1. Open wireshark in your system.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Open
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown

C:\Users\ASHEESH\Desktop\Nort Working CPE-aryan 9th Dec.pcapng (2968 KB)
C:\Users\ASHEESH\Desktop\CPE-aryan 9th Dec.pcapng (not found)
E:\8decKENSTALlogs.pcapng (193 MB)
E:\ritvik\upf-ref.pcap (61 MB)
C:\Users\ASHEESH\Desktop\drone.pcap (not found)
E:\uat-core.pcap.pcapng (12 MB)
E:\ulcl-pfcp.pcap (63 KB)
E:\ulcl.pcap (78 KB)
D:\12th_DUT_New_2.pcap (not found)
C:\Users\ASHEESH\Desktop\ng-ran.pcap (not found)

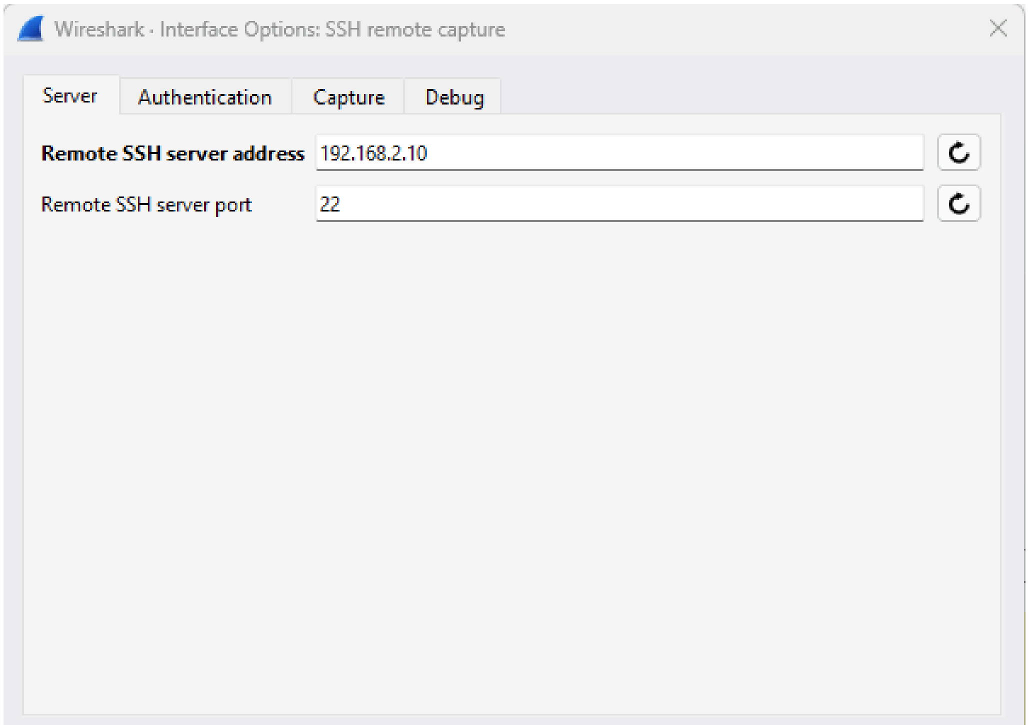
Local interfaces are unavailable because no packet capture driver is installed.

You can fix this by installing Npcap.

Don't show this message again.

- Cisco remote capture
- Event Tracing for Windows (ETW) reader
- SSH remote capture
- Wi-Fi remote capture

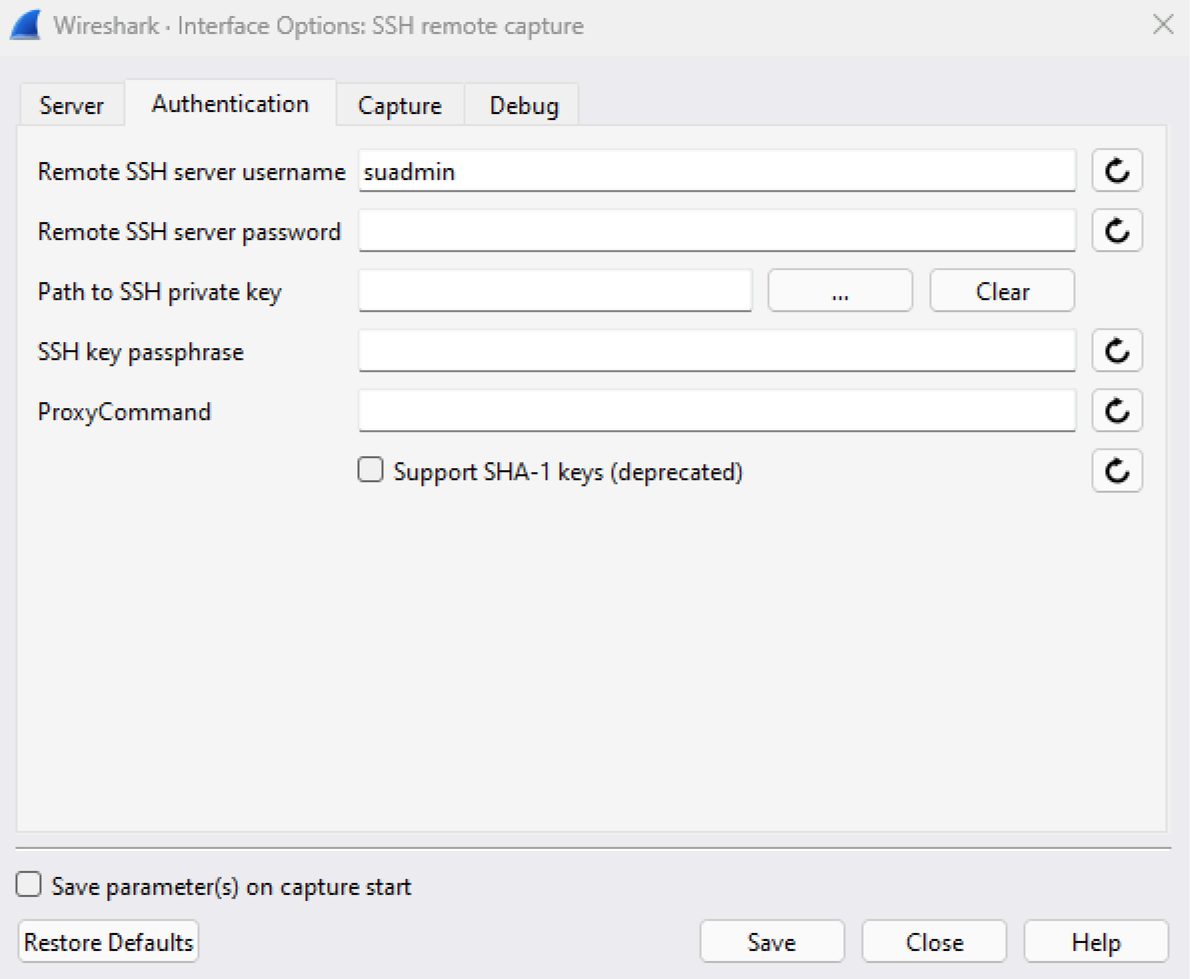
2. To capture particular RAN logs enter its IP address in the Remote SSH server address.



Remote SSH server address 192.168.2.10

Remote SSH server port 22

3. Enter the Remote ssh server username in the Authentication.



Wireshark · Interface Options: SSH remote capture

Remote SSH server username: suadmin

Remote SSH server password:

Path to SSH private key:

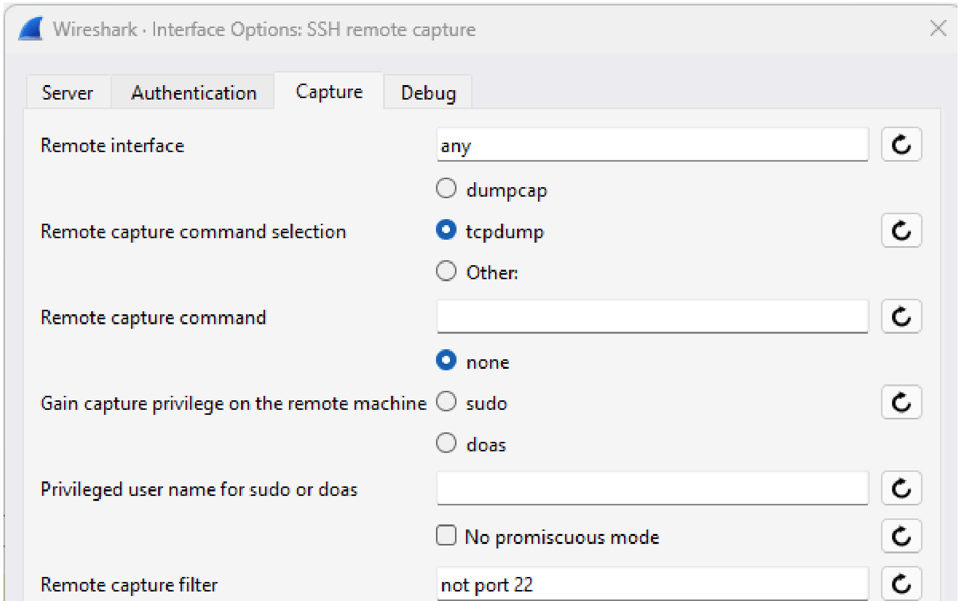
SSH key passphrase:

ProxyCommand:

□ Support SHA-1 keys (deprecated)

□ Save parameter(s) on capture start

4. Set the following options as shown in the image below for the capture.



Wireshark · Interface Options: SSH remote capture

Remote interface: any

Remote capture command selection: ○ dumpcap ● tcpdump ○ Other:

Remote capture command:

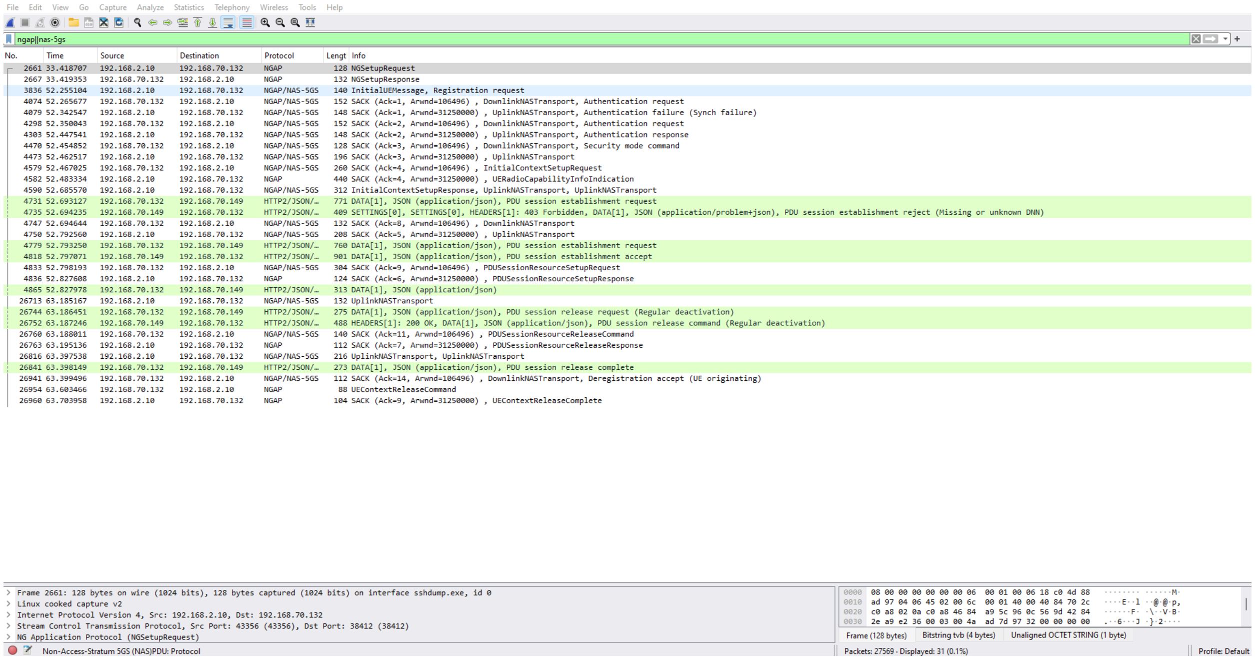
Gain capture privilege on the remote machine: ● none ○ sudo ○ doas

Privileged user name for sudo or doas:

□ No promiscuous mode

Remote capture filter: not port 22

5. Use ngap||nas-5gs filter to see 5g sa call flows as in the below image:



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ngap||nas-5gs
```

```
No.    Time      Source          Destination     Protocol       Lengt Info
 2661 33.418707 192.168.2.10    192.168.70.132  NGAP            128 NGSetupRequest
 2667 33.419353 192.168.70.132  192.168.2.10    NGAP            132 NGSetupResponse
 3836 52.255104 192.168.2.10    192.168.70.132  NGAP/NAS-5GS    140 InitialUEMessage, Registration request
 4074 52.265677 192.168.70.132  192.168.2.10    NGAP/NAS-5GS    152 SACK (Ack=1, Arwnd=106496) , DownlinkNASTransport, Authentication request
 4079 52.342547 192.168.2.10    192.168.70.132  NGAP/NAS-5GS    148 SACK (Ack=1, Arwnd=31250000) , UplinkNASTransport, Authentication failure (Synch failure)
 4298 52.350043 192.168.70.132  192.168.2.10    NGAP/NAS-5GS    152 SACK (Ack=2, Arwnd=106496) , DownlinkNASTransport, Authentication request
 4303 52.447541 192.168.2.10    192.168.70.132  NGAP/NAS-5GS    148 SACK (Ack=2, Arwnd=31250000) , UplinkNASTransport, Authentication response
 4470 52.454852 192.168.70.132  192.168.2.10    NGAP/NAS-5GS    128 SACK (Ack=3, Arwnd=106496) , DownlinkNASTransport, Security mode command
 4473 52.462517 192.168.2.10    192.168.70.132  NGAP/NAS-5GS    196 SACK (Ack=3, Arwnd=31250000) , UplinkNASTransport
 4579 52.467025 192.168.70.132  192.168.2.10    NGAP/NAS-5GS    260 SACK (Ack=4, Arwnd=106496) , InitialContextSetupRequest
 4582 52.483334 192.168.2.10    192.168.70.132  NGAP            440 SACK (Ack=4, Arwnd=31250000) , UERadioCapabilityInfoIndication
 4590 52.685570 192.168.2.10    192.168.70.132  NGAP/NAS-5GS    312 InitialContextSetupResponse, UplinkNASTransport, UplinkNASTransport
 4731 52.693127 192.168.70.132  192.168.70.149  HTTP2/JSON/…    771 DATA[1], JSON (application/json), PDU session establishment request
 4735 52.694235 192.168.70.149  192.168.70.132  HTTP2/JSON/…    409 SETTINGS[0], SETTINGS[0], HEADERS[1]: 403 Forbidden, DATA[1], JSON (application/problem+json), PDU session establishment reject (Missing or unknown DNN)
 4747 52.694644 192.168.70.132  192.168.2.10    NGAP/NAS-5GS    132 SACK (Ack=8, Arwnd=106496) , DownlinkNASTransport
 4750 52.792560 192.168.2.10    192.168.70.132  NGAP/NAS-5GS    208 SACK (Ack=5, Arwnd=31250000) , UplinkNASTransport
 4779 52.793250 192.168.70.132  192.168.70.149  HTTP2/JSON/…    760 DATA[1], JSON (application/json), PDU session establishment request
 4818 52.797071 192.168.70.149  192.168.70.132  HTTP2/JSON/…    901 DATA[1], JSON (application/json), PDU session establishment accept
 4833 52.798193 192.168.70.132  192.168.2.10    NGAP/NAS-5GS    304 SACK (Ack=9, Arwnd=106496) , PDUSessionResourceSetupRequest
 4836 52.827608 192.168.2.10    192.168.70.132  NGAP            124 SACK (Ack=6, Arwnd=31250000) , PDUSessionResourceSetupResponse
 4865 52.827978 192.168.70.132  192.168.70.149  HTTP2/JSON/…    313 DATA[1], JSON (application/json)
26713 63.185167 192.168.2.10    192.168.70.132  NGAP/NAS-5GS    132 UplinkNASTransport
26744 63.186451 192.168.70.132  192.168.70.149  HTTP2/JSON/…    275 DATA[1], JSON (application/json), PDU session release request (Regular deactivation)
26752 63.187246 192.168.70.149  192.168.70.132  HTTP2/JSON/…    488 HEADERS[1]: 200 OK, DATA[1], JSON (application/json), PDU session release command (Regular deactivation)
26760 63.188011 192.168.70.132  192.168.2.10    NGAP/NAS-5GS    140 SACK (Ack=11, Arwnd=106496) , PDUSessionResourceReleaseCommand
26763 63.195136 192.168.2.10    192.168.70.132  NGAP            112 SACK (Ack=7, Arwnd=31250000) , PDUSessionResourceReleaseResponse
26816 63.397538 192.168.2.10    192.168.70.132  NGAP/NAS-5GS    216 UplinkNASTransport, UplinkNASTransport
26841 63.398149 192.168.70.132  192.168.70.149  HTTP2/JSON/…    273 DATA[1], JSON (application/json), PDU session release complete
26941 63.399496 192.168.70.132  192.168.2.10    NGAP/NAS-5GS    112 SACK (Ack=14, Arwnd=106496) , DownlinkNASTransport, Deregistration accept (UE originating)
26954 63.603466 192.168.70.132  192.168.2.10    NGAP             88 UEContextReleaseCommand
26960 63.703958 192.168.2.10    192.168.70.132  NGAP            104 SACK (Ack=9, Arwnd=31250000) , UEContextReleaseComplete
```

```
> Frame 2661: 128 bytes on wire (1024 bits), 128 bytes captured (1024 bits) on interface sshdump.exe, id 0
> Linux cooked capture v2
> Internet Protocol Version 4, Src: 192.168.2.10, Dst: 192.168.70.132
> Stream Control Transmission Protocol, Src Port: 43356 (43356), Dst Port: 38412 (38412)
> NG Application Protocol (NGSetupRequest)
```

```
0000  08 00 00 00 00 00 00 06  00 01 00 06 18 c0 4d 88   ········ ······M·
0010  ad 97 04 06 45 02 00 6c  00 01 40 00 40 84 70 2c   ····E··l ··@·@·p,
0020  c0 a8 02 0a c0 a8 46 84  a9 5c 96 0c 56 9d 42 84   ······F· ·\··V·B·
0030  2e a9 e2 36 00 03 00 4a  ad 7d 97 32 00 00 00 00   .··6···J ·}·2····
```

Frame (128 bytes)
Bitstring tvb (4 bytes)
Unaligned OCTET STRING (1 byte)
Non-Access-Stratum 5GS (NAS)PDU: Protocol
Packets: 27569 · Displayed: 31 (0.1%)
Profile: Default

20.2. Capturing Logs For RAN

Steps for Capturing Logs for RAN:

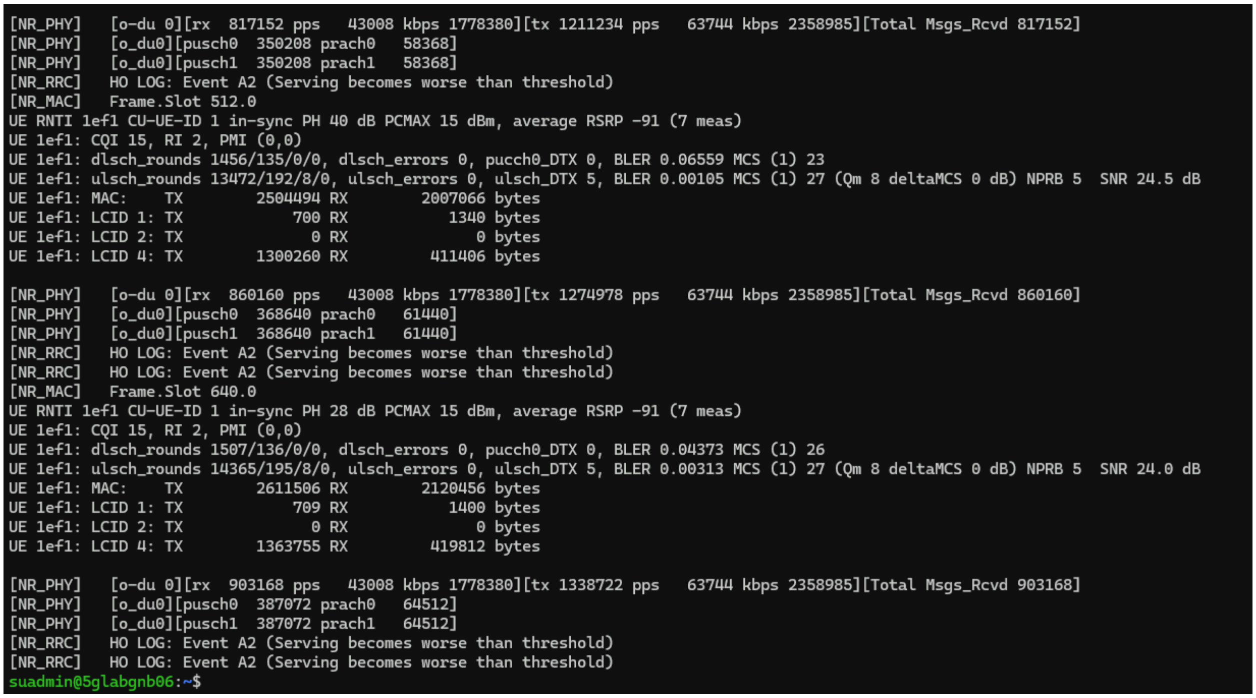
1. Log in to gNB by executing **ssh suadmin@192.168.xx.10** (here xx refers for the subnet based on institute) command.
Password: Admin@123

2. Execute the RU steps for login to LPRU.

NOTE: Please Refer to Steps for Pre-Configured RU.

3. Use the command **docker ps -a** to check the status of all the services.

4. Execute **Tail -f gnb.log** command to capture the logs.

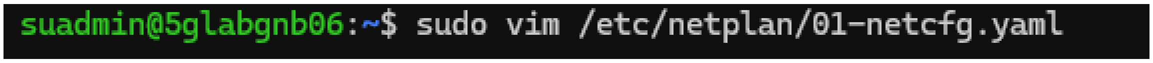


```
[NR_PHY]   [o-du 0][rx  817152 pps   43008 kbps 1778380][tx 1211234 pps   63744 kbps 2358985][Total Msgs_Rcvd 817152]
[NR_PHY]   [o_du0][pusch0  350208 prach0   58368]
[NR_PHY]   [o_du0][pusch1  350208 prach1   58368]
[NR_RRC]   HO LOG: Event A2 (Serving becomes worse than threshold)
[NR_MAC]   Frame.Slot 512.0
UE RNTI 1ef1 CU-UE-ID 1 in-sync PH 40 dB PCMAX 15 dBm, average RSRP -91 (7 meas)
UE 1ef1: CQI 15, RI 2, PMI (0,0)
UE 1ef1: dlsch_rounds 1456/135/0/0, dlsch_errors 0, pucch0_DTX 0, BLER 0.06559 MCS (1) 23
UE 1ef1: ulsch_rounds 13472/192/8/0, ulsch_errors 0, ulsch_DTX 5, BLER 0.00105 MCS (1) 27 (Qm 8 deltaMCS 0 dB) NPRB 5  SNR 24.5 dB
UE 1ef1: MAC:    TX        2504494 RX        2007066 bytes
UE 1ef1: LCID 1: TX            700 RX           1340 bytes
UE 1ef1: LCID 2: TX              0 RX              0 bytes
UE 1ef1: LCID 4: TX        1300260 RX         411406 bytes

[NR_PHY]   [o-du 0][rx  860160 pps   43008 kbps 1778380][tx 1274978 pps   63744 kbps 2358985][Total Msgs_Rcvd 860160]
[NR_PHY]   [o_du0][pusch0  368640 prach0   61440]
[NR_PHY]   [o_du0][pusch1  368640 prach1   61440]
[NR_RRC]   HO LOG: Event A2 (Serving becomes worse than threshold)
[NR_RRC]   HO LOG: Event A2 (Serving becomes worse than threshold)
[NR_MAC]   Frame.Slot 640.0
UE RNTI 1ef1 CU-UE-ID 1 in-sync PH 28 dB PCMAX 15 dBm, average RSRP -91 (7 meas)
UE 1ef1: CQI 15, RI 2, PMI (0,0)
UE 1ef1: dlsch_rounds 1507/136/0/0, dlsch_errors 0, pucch0_DTX 0, BLER 0.04373 MCS (1) 26
UE 1ef1: ulsch_rounds 14365/195/8/0, ulsch_errors 0, ulsch_DTX 5, BLER 0.00313 MCS (1) 27 (Qm 8 deltaMCS 0 dB) NPRB 5  SNR 24.0 dB
UE 1ef1: MAC:    TX        2611506 RX        2120456 bytes
UE 1ef1: LCID 1: TX            709 RX           1400 bytes
UE 1ef1: LCID 2: TX              0 RX              0 bytes
UE 1ef1: LCID 4: TX        1363755 RX         419812 bytes

[NR_PHY]   [o-du 0][rx  903168 pps   43008 kbps 1778380][tx 1338722 pps   63744 kbps 2358985][Total Msgs_Rcvd 903168]
[NR_PHY]   [o_du0][pusch0  387072 prach0   64512]
[NR_PHY]   [o_du0][pusch1  387072 prach1   64512]
[NR_RRC]   HO LOG: Event A2 (Serving becomes worse than threshold)
[NR_RRC]   HO LOG: Event A2 (Serving becomes worse than threshold)
suadmin@5glabgnb06:~$
```

20.3. Troubleshooting for IP Route

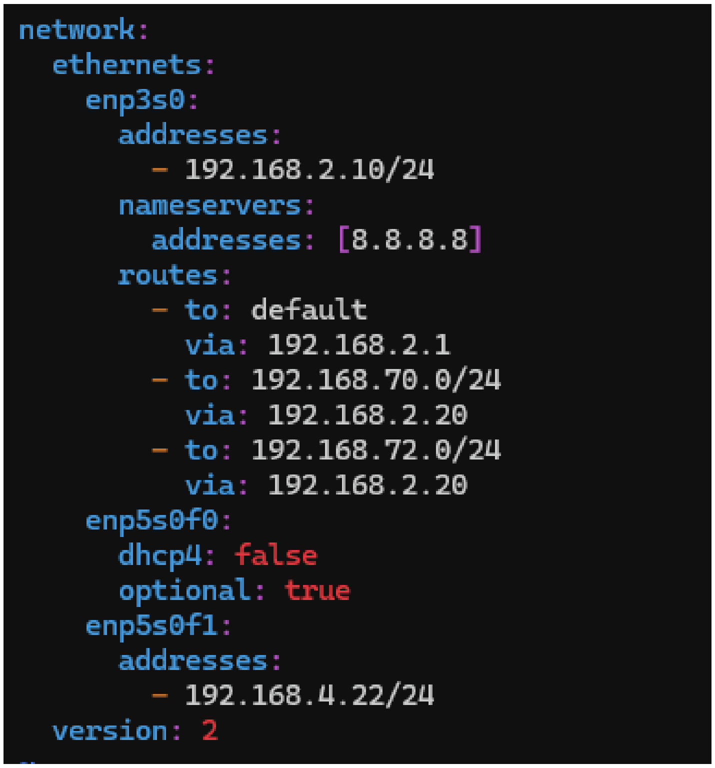
When an IP route is not properly configured, errors may appear in the gNB logs. We can implement both static and dynamic routing to facilitate communication between different core containers and establish connections with external 5G devices.

Use the following file path:



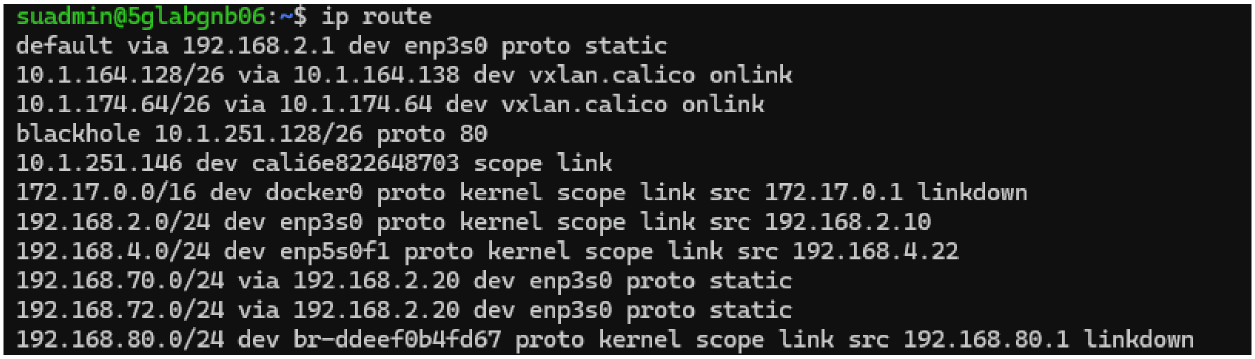
```
suadmin@5glabgnb06:~$ sudo vim /etc/netplan/01-netcfg.yaml
```

To add IP's permanently; we make routes in the netplan directory as follows :



```
network:
  ethernets:
    enp3s0:
      addresses:
        - 192.168.2.10/24
      nameservers:
        addresses: [8.8.8.8]
      routes:
        - to: default
          via: 192.168.2.1
        - to: 192.168.70.0/24
          via: 192.168.2.20
        - to: 192.168.72.0/24
          via: 192.168.2.20
    enp5s0f0:
      dhcp4: false
      optional: true
    enp5s0f1:
      addresses:
        - 192.168.4.22/24
  version: 2
```

Alternatively, **sudo ip route** command to add 192.168.x.x./24 via 192.168.x.20 or 192.168.x.10 (gnb).

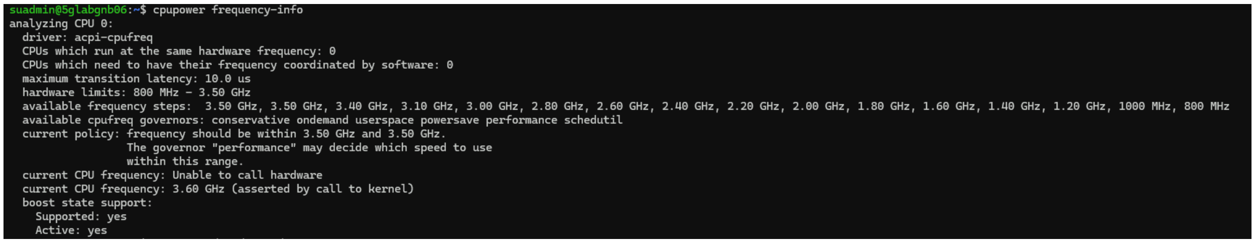


```
suadmin@5glabgnb06:~$ ip route
default via 192.168.2.1 dev enp3s0 proto static
10.1.164.128/26 via 10.1.164.138 dev vxlan.calico onlink
10.1.174.64/26 via 10.1.174.64 dev vxlan.calico onlink
blackhole 10.1.251.128/26 proto 80
10.1.251.146 dev cali6e822648703 scope link
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
192.168.2.0/24 dev enp3s0 proto kernel scope link src 192.168.2.10
192.168.4.0/24 dev enp5s0f1 proto kernel scope link src 192.168.4.22
192.168.70.0/24 via 192.168.2.20 dev enp3s0 proto static
192.168.72.0/24 via 192.168.2.20 dev enp3s0 proto static
192.168.80.0/24 dev br-ddeef0b4fd67 proto kernel scope link src 192.168.80.1 linkdown
```

NOTE: Please refer to the solutions manual for more Troubleshooting solutions for RAN

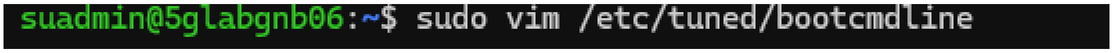
21. Troubleshooting for the Drivers

1. To ensure optimal performance, please execute the command **cpupower frequency-info** whenever the driver name is displayed, except in the case of pstate.



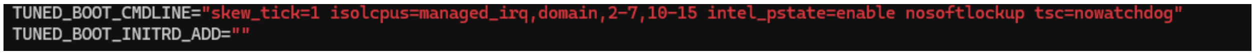
```
suadmin@5glabgnb06:~$ cpupower frequency-info
analyzing CPU 0:
  driver: acpi-cpufreq
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 10.0 us
  hardware limits: 800 MHz - 3.50 GHz
  available frequency steps:  3.50 GHz, 3.50 GHz, 3.40 GHz, 3.10 GHz, 3.00 GHz, 2.80 GHz, 2.60 GHz, 2.40 GHz, 2.20 GHz, 2.00 GHz, 1.80 GHz, 1.60 GHz, 1.40 GHz, 1.20 GHz, 1000 MHz, 800 MHz
  available cpufreq governors: conservative ondemand userspace powersave performance schedutil
  current policy: frequency should be within 3.50 GHz and 3.50 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency: Unable to call hardware
  current CPU frequency: 3.60 GHz (asserted by call to kernel)
  boost state support:
    Supported: yes
    Active: yes
```

2. Make Changes in the following directory:



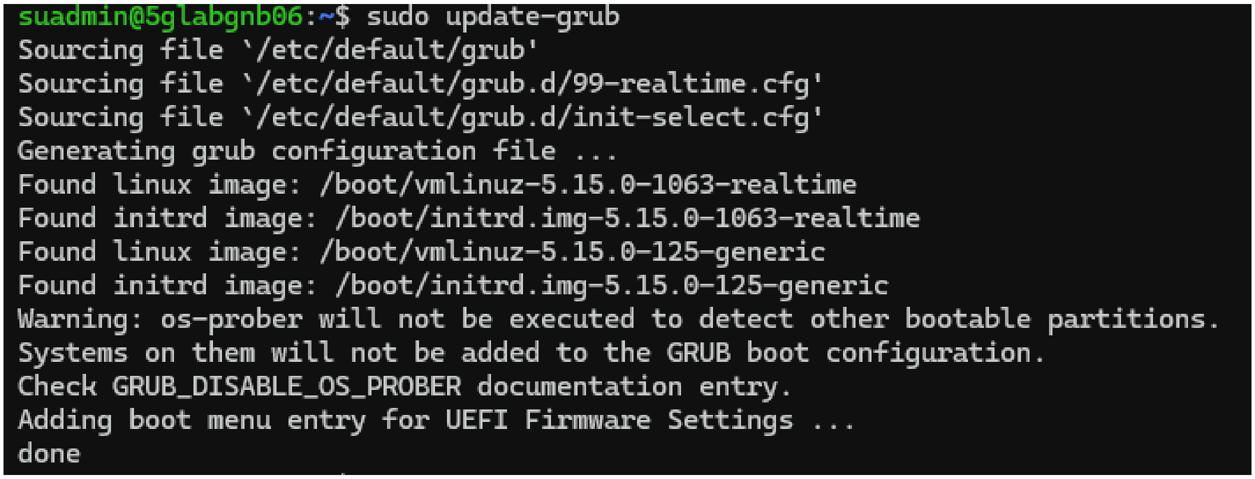
```
suadmin@5glabgnb06:~$ sudo vim /etc/tuned/bootcmdline
```

3. Replace disable with enable as shown in the following image.



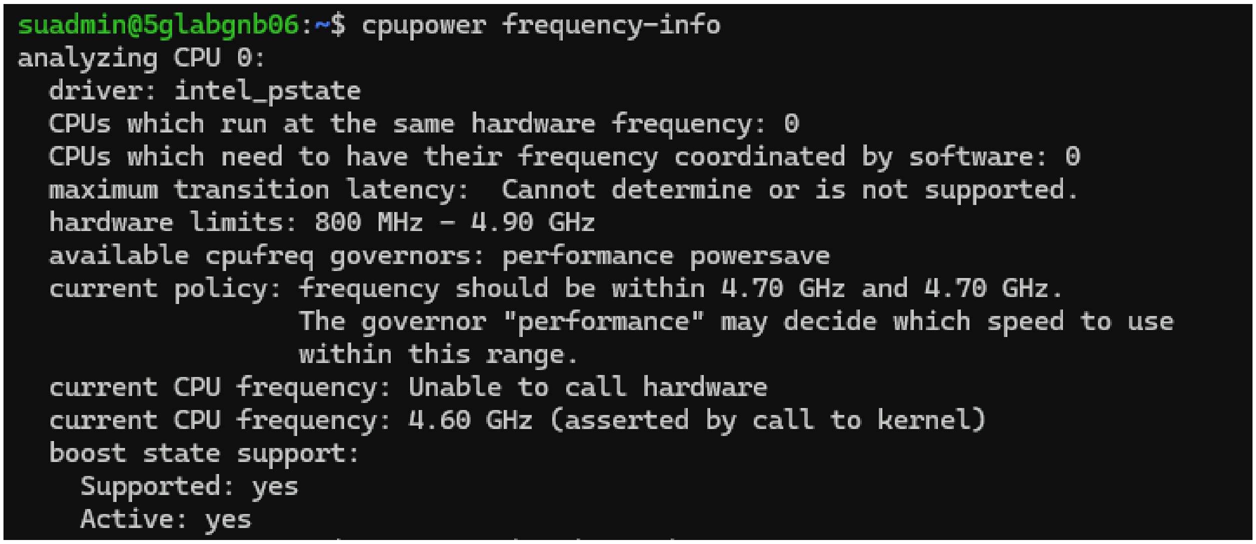
```
TUNED_BOOT_CMDLINE="skew_tick=1 isolcpus=managed_irq,domain,2-7,10-15 intel_pstate=enable nosoftlockup tsc=nowatchdog"
TUNED_BOOT_INITRD_ADD=""
```

4. Execute **sudo update-grub** command.



```
suadmin@5glabgnb06:~$ sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/99-realtime.cfg'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.0-1063-realtime
Found initrd image: /boot/initrd.img-5.15.0-1063-realtime
Found linux image: /boot/vmlinuz-5.15.0-125-generic
Found initrd image: /boot/initrd.img-5.15.0-125-generic
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
```

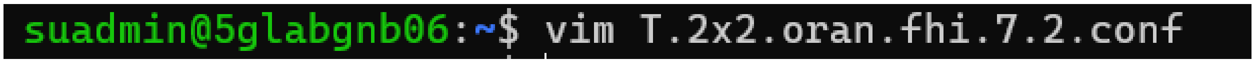
We can see the intel pstate by cross checking it by **cpupower frequency-info** command.



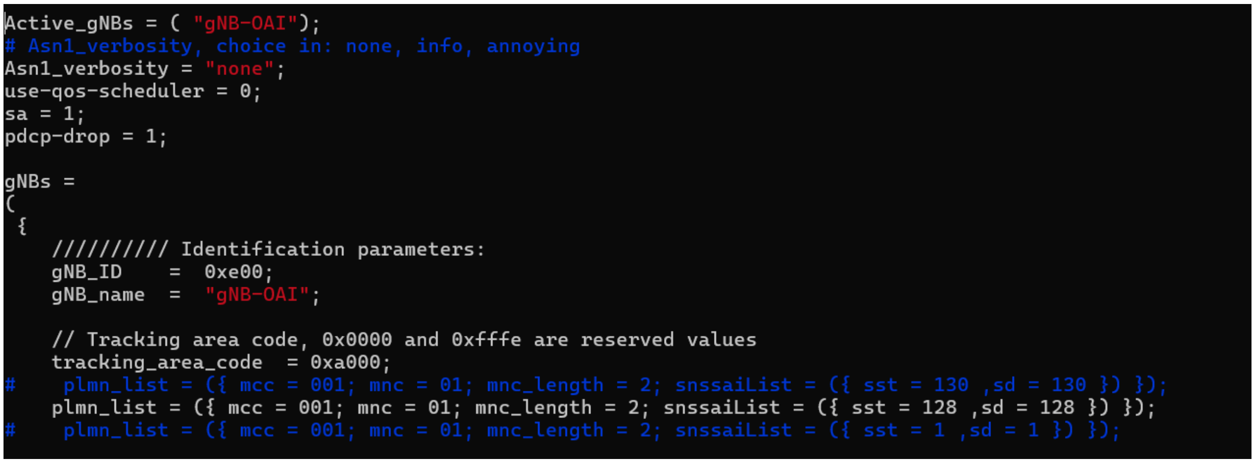
```
suadmin@5glabgnb06:~$ cpupower frequency-info
analyzing CPU 0:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency:  Cannot determine or is not supported.
  hardware limits: 800 MHz - 4.90 GHz
  available cpufreq governors: performance powersave
  current policy: frequency should be within 4.70 GHz and 4.70 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency: Unable to call hardware
  current CPU frequency: 4.60 GHz (asserted by call to kernel)
  boost state support:
    Supported: yes
    Active: yes
```

22. Troubleshooting for RU

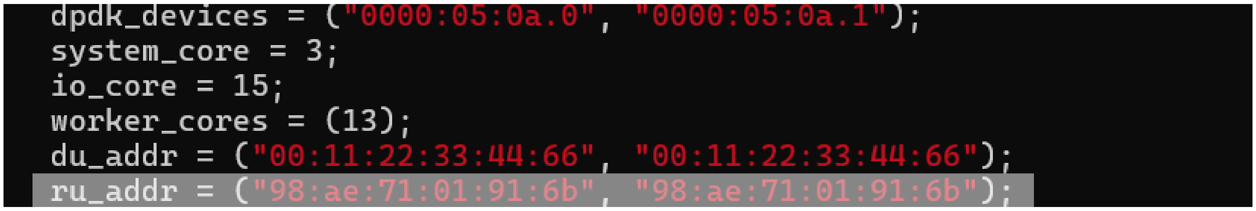
1. Login into gNB
2. Login into lpru and cross-check RU mac-address by “ip a” command and put same value in T.2x2.oran.fhi.7.2.conf file



```
suadmin@5glabgnb06:~$ vim T.2x2.oran.fhi.7.2.conf
```

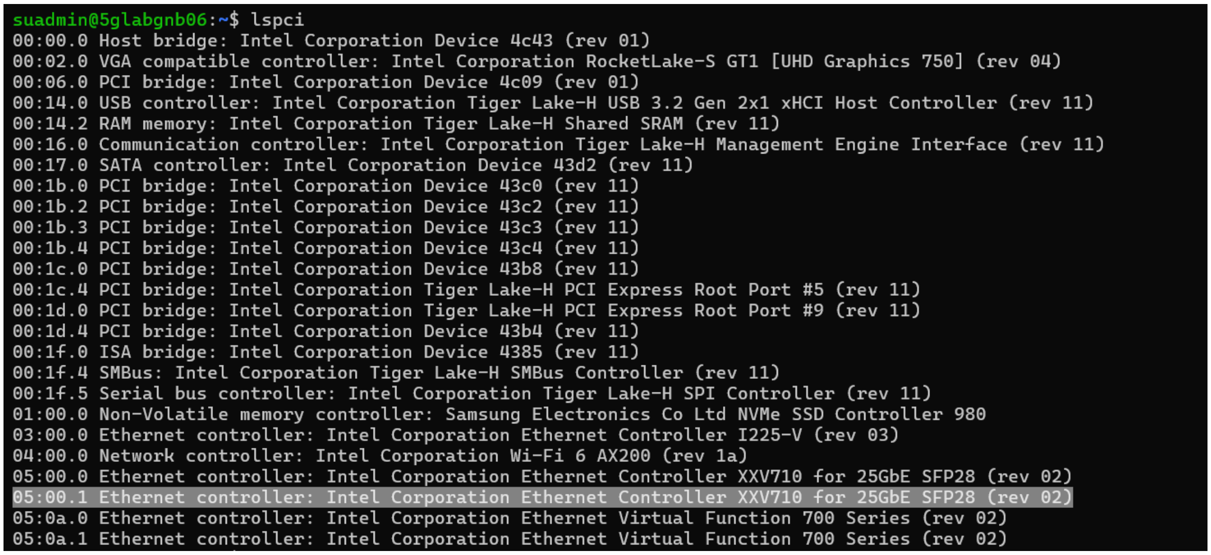


```
Active_gNBs = ( "gNB-OAI");
# Asn1_verbosity, choice in: none, info, annoying
Asn1_verbosity = "none";
use-qos-scheduler = 0;
sa = 1;
pdcp-drop = 1;

gNBs =
(
 {
    ////////// Identification parameters:
    gNB_ID    =  0xe00;
    gNB_name  =  "gNB-OAI";

    // Tracking area code, 0x0000 and 0xfffe are reserved values
    tracking_area_code  = 0xa000;
#    plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 130 ,sd = 130 }) });
    plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 128 ,sd = 128 }) });
#    plmn_list = ({ mcc = 001; mnc = 01; mnc_length = 2; snssaiList = ({ sst = 1 ,sd = 1 }) });
```



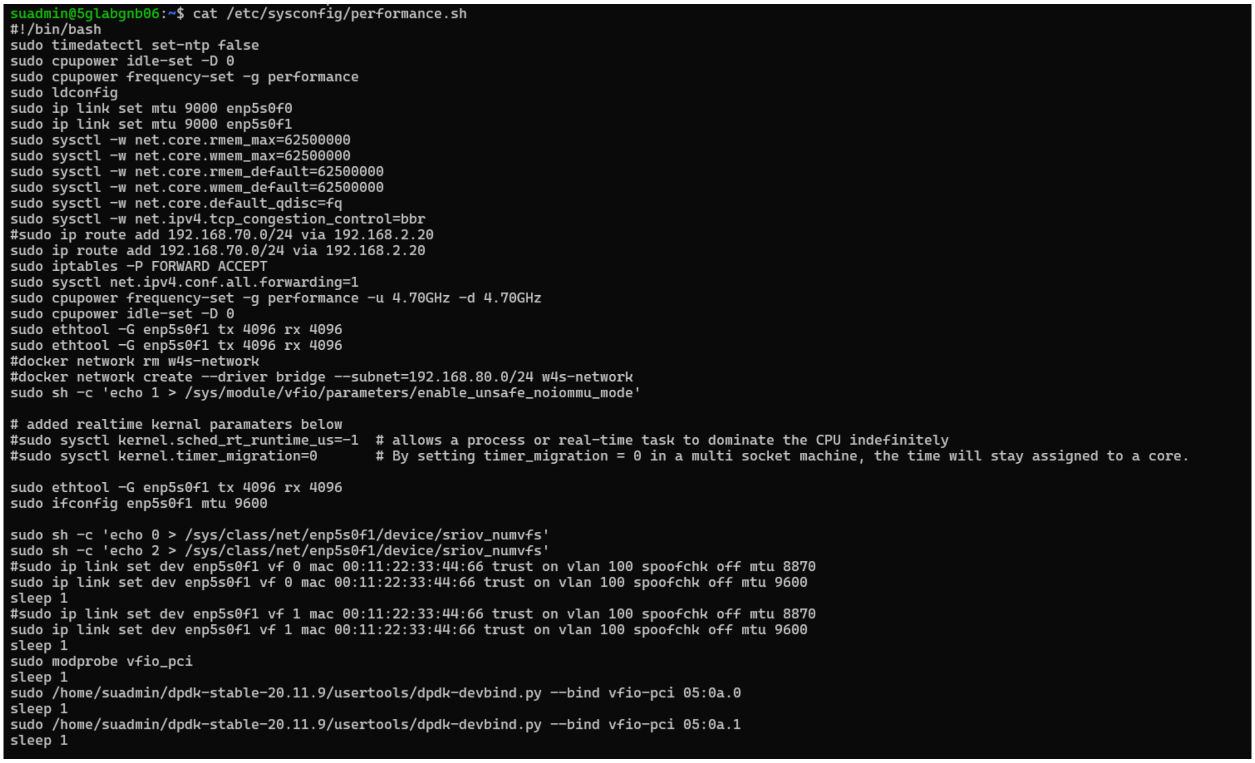
```
    dpdk_devices = ("0000:05:0a.0", "0000:05:0a.1");
    system_core = 3;
    io_core = 15;
    worker_cores = (13);
    du_addr = ("00:11:22:33:44:66", "00:11:22:33:44:66");
    ru_addr = ("98:ae:71:01:91:6b", "98:ae:71:01:91:6b");
```

3.The device configuration needs to be cross-checked by the command **lspci** in ran where it will show the dpdk devices number bind to the ethernet card pcie slot.



```
suadmin@5glabgnb06:~$ lspci
00:00.0 Host bridge: Intel Corporation Device 4c43 (rev 01)
00:02.0 VGA compatible controller: Intel Corporation RocketLake-S GT1 [UHD Graphics 750] (rev 04)
00:06.0 PCI bridge: Intel Corporation Device 4c09 (rev 01)
00:14.0 USB controller: Intel Corporation Tiger Lake-H USB 3.2 Gen 2x1 xHCI Host Controller (rev 11)
00:14.2 RAM memory: Intel Corporation Tiger Lake-H Shared SRAM (rev 11)
00:16.0 Communication controller: Intel Corporation Tiger Lake-H Management Engine Interface (rev 11)
00:17.0 SATA controller: Intel Corporation Device 43d2 (rev 11)
00:1b.0 PCI bridge: Intel Corporation Device 43c0 (rev 11)
00:1b.2 PCI bridge: Intel Corporation Device 43c2 (rev 11)
00:1b.3 PCI bridge: Intel Corporation Device 43c3 (rev 11)
00:1b.4 PCI bridge: Intel Corporation Device 43c4 (rev 11)
00:1c.0 PCI bridge: Intel Corporation Device 43b8 (rev 11)
00:1c.4 PCI bridge: Intel Corporation Tiger Lake-H PCI Express Root Port #5 (rev 11)
00:1d.0 PCI bridge: Intel Corporation Tiger Lake-H PCI Express Root Port #9 (rev 11)
00:1d.4 PCI bridge: Intel Corporation Device 43b4 (rev 11)
00:1f.0 ISA bridge: Intel Corporation Device 4385 (rev 11)
00:1f.4 SMBus: Intel Corporation Tiger Lake-H SMBus Controller (rev 11)
00:1f.5 Serial bus controller: Intel Corporation Tiger Lake-H SPI Controller (rev 11)
01:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller 980
03:00.0 Ethernet controller: Intel Corporation Ethernet Controller I225-V (rev 03)
04:00.0 Network controller: Intel Corporation Wi-Fi 6 AX200 (rev 1a)
05:00.0 Ethernet controller: Intel Corporation Ethernet Controller XXV710 for 25GbE SFP28 (rev 02)
05:00.1 Ethernet controller: Intel Corporation Ethernet Controller XXV710 for 25GbE SFP28 (rev 02)
05:0a.0 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
05:0a.1 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
```

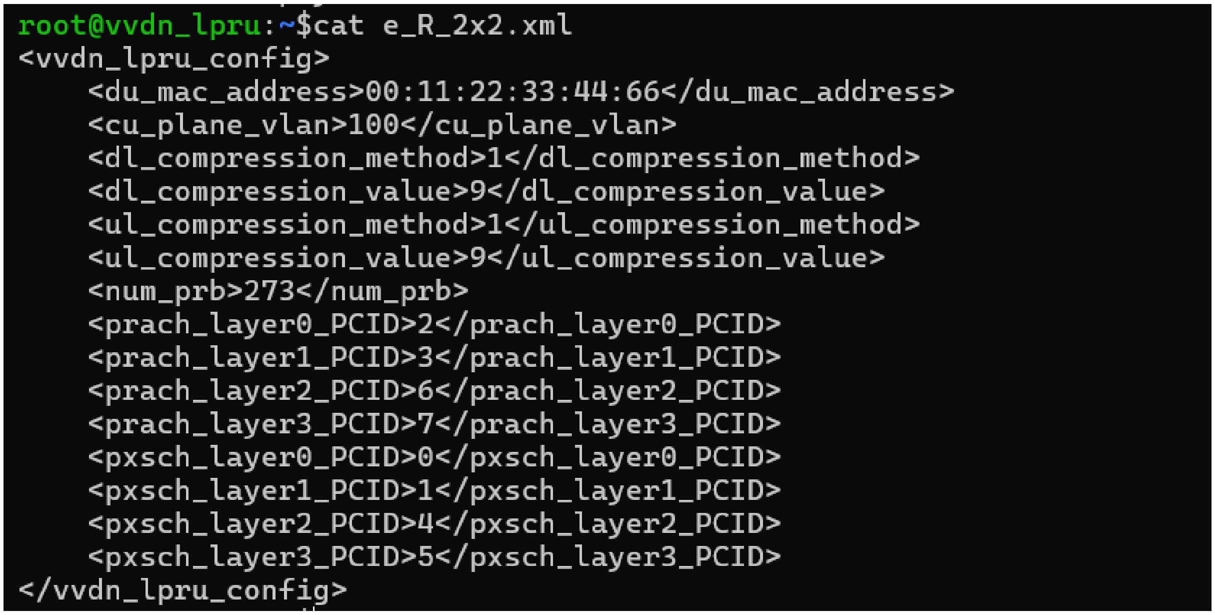
4. In the performance.sh file which is located in the directory : sudo vim /etc/sysconfig/performance.sh



```
suadmin@5glabgnb06:~$ cat /etc/sysconfig/performance.sh
#!/bin/bash
sudo timedatectl set-ntp false
sudo cpupower idle-set -D 0
sudo cpupower frequency-set -g performance
sudo ldconfig
sudo ip link set mtu 9000 enp5s0f0
sudo ip link set mtu 9000 enp5s0f1
sudo sysctl -w net.core.rmem_max=62500000
sudo sysctl -w net.core.wmem_max=62500000
sudo sysctl -w net.core.rmem_default=62500000
sudo sysctl -w net.core.wmem_default=62500000
sudo sysctl -w net.core.default_qdisc=fq
sudo sysctl -w net.ipv4.tcp_congestion_control=bbr
#sudo ip route add 192.168.70.0/24 via 192.168.2.20
sudo ip route add 192.168.70.0/24 via 192.168.2.20
sudo iptables -P FORWARD ACCEPT
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo cpupower frequency-set -g performance -u 4.70GHz -d 4.70GHz
sudo cpupower idle-set -D 0
sudo ethtool -G enp5s0f1 tx 4096 rx 4096
sudo ethtool -G enp5s0f1 tx 4096 rx 4096
#docker network rm w4s-network
#docker network create --driver bridge --subnet=192.168.80.0/24 w4s-network
sudo sh -c 'echo 1 > /sys/module/vfio/parameters/enable_unsafe_noiommu_mode'

# added realtime kernal paramaters below
#sudo sysctl kernel.sched_rt_runtime_us=-1  # allows a process or real-time task to dominate the CPU indefinitely
#sudo sysctl kernel.timer_migration=0       # By setting timer_migration = 0 in a multi socket machine, the time will stay assigned to a core.

sudo ethtool -G enp5s0f1 tx 4096 rx 4096
sudo ifconfig enp5s0f1 mtu 9600

sudo sh -c 'echo 0 > /sys/class/net/enp5s0f1/device/sriov_numvfs'
sudo sh -c 'echo 2 > /sys/class/net/enp5s0f1/device/sriov_numvfs'
#sudo ip link set dev enp5s0f1 vf 0 mac 00:11:22:33:44:66 trust on vlan 100 spoofchk off mtu 8870
sudo ip link set dev enp5s0f1 vf 0 mac 00:11:22:33:44:66 trust on vlan 100 spoofchk off mtu 9600
sleep 1
#sudo ip link set dev enp5s0f1 vf 1 mac 00:11:22:33:44:66 trust on vlan 100 spoofchk off mtu 8870
sudo ip link set dev enp5s0f1 vf 1 mac 00:11:22:33:44:66 trust on vlan 100 spoofchk off mtu 9600
sleep 1
sudo modprobe vfio_pci
sleep 1
sudo /home/suadmin/dpdk-stable-20.11.9/usertools/dpdk-devbind.py --bind vfio-pci 05:0a.0
sleep 1
sudo /home/suadmin/dpdk-stable-20.11.9/usertools/dpdk-devbind.py --bind vfio-pci 05:0a.1
sleep 1
```

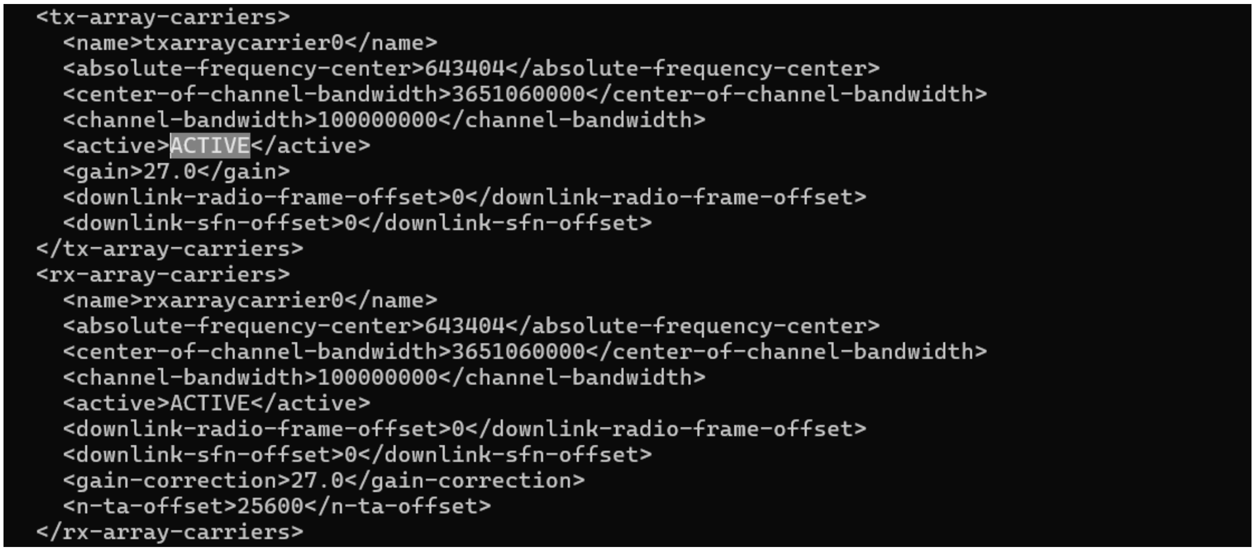
22.1. Troubleshooting for LPRU configuration

1. Use the **cat e_R_2x2.xml** Command to check the configuration for the LPRU.

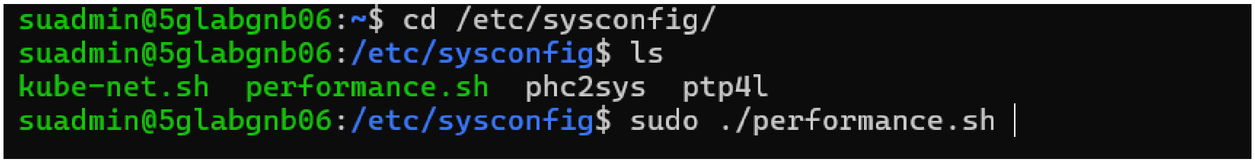


```
root@vvdn_lpru:~$cat e_R_2x2.xml
<vvdn_lpru_config>
    <du_mac_address>00:11:22:33:44:66</du_mac_address>
    <cu_plane_vlan>100</cu_plane_vlan>
    <dl_compression_method>1</dl_compression_method>
    <dl_compression_value>9</dl_compression_value>
    <ul_compression_method>1</ul_compression_method>
    <ul_compression_value>9</ul_compression_value>
    <num_prb>273</num_prb>
    <prach_layer0_PCID>2</prach_layer0_PCID>
    <prach_layer1_PCID>3</prach_layer1_PCID>
    <prach_layer2_PCID>6</prach_layer2_PCID>
    <prach_layer3_PCID>7</prach_layer3_PCID>
    <pxsch_layer0_PCID>0</pxsch_layer0_PCID>
    <pxsch_layer1_PCID>1</pxsch_layer1_PCID>
    <pxsch_layer2_PCID>4</pxsch_layer2_PCID>
    <pxsch_layer3_PCID>5</pxsch_layer3_PCID>
</vvdn_lpru_config>
```

2. Execute only 2x2_xml file (which can be crosschecked at no. 273 in its file) in RU and run the ran (2024-w33)



```
  <tx-array-carriers>
    <name>txarraycarrier0</name>
    <absolute-frequency-center>643404</absolute-frequency-center>
    <center-of-channel-bandwidth>3651060000</center-of-channel-bandwidth>
    <channel-bandwidth>100000000</channel-bandwidth>
    <active>ACTIVE</active>
    <gain>27.0</gain>
    <downlink-radio-frame-offset>0</downlink-radio-frame-offset>
    <downlink-sfn-offset>0</downlink-sfn-offset>
  </tx-array-carriers>
  <rx-array-carriers>
    <name>rxarraycarrier0</name>
    <absolute-frequency-center>643404</absolute-frequency-center>
    <center-of-channel-bandwidth>3651060000</center-of-channel-bandwidth>
    <channel-bandwidth>100000000</channel-bandwidth>
    <active>ACTIVE</active>
    <downlink-radio-frame-offset>0</downlink-radio-frame-offset>
    <downlink-sfn-offset>0</downlink-sfn-offset>
    <gain-correction>27.0</gain-correction>
    <n-ta-offset>25600</n-ta-offset>
  </rx-array-carriers>
```



```
suadmin@5glabgnb06:~$ cd /etc/sysconfig/
suadmin@5glabgnb06:/etc/sysconfig$ ls
kube-net.sh  performance.sh  phc2sys  ptp4l
suadmin@5glabgnb06:/etc/sysconfig$ sudo ./performance.sh
```

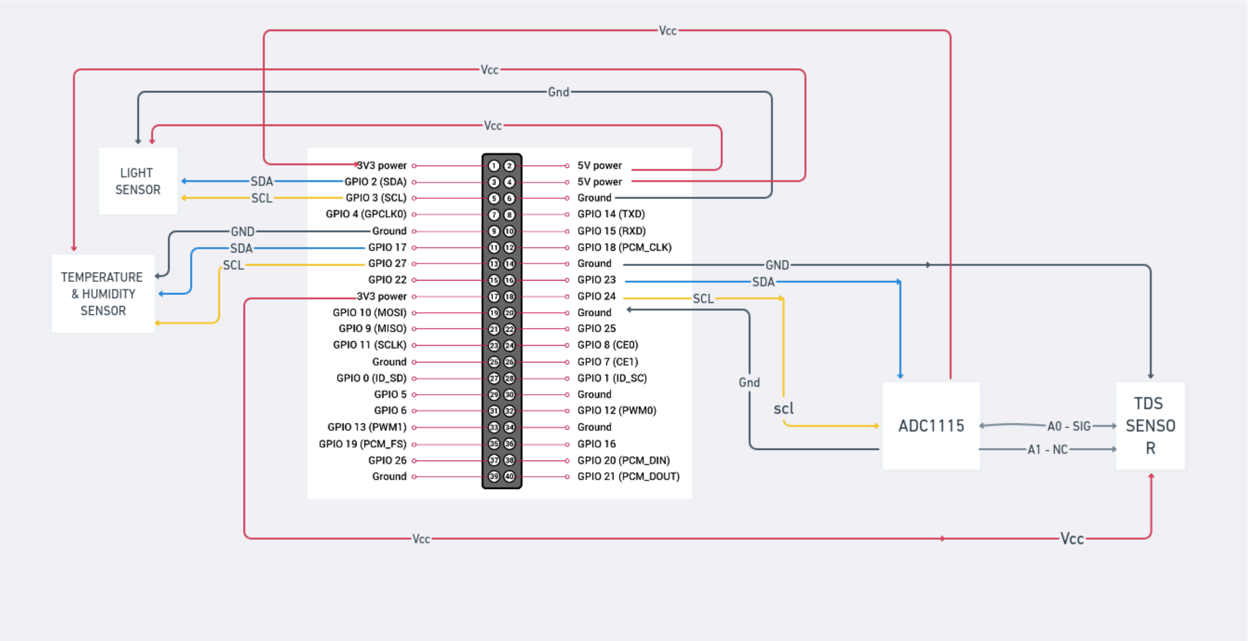
NOTE: Please refer to the solutions manual for more Troubleshooting solutions for RU

23. Troubleshooting for IoT Sensors

23.1. Non-Working sensors

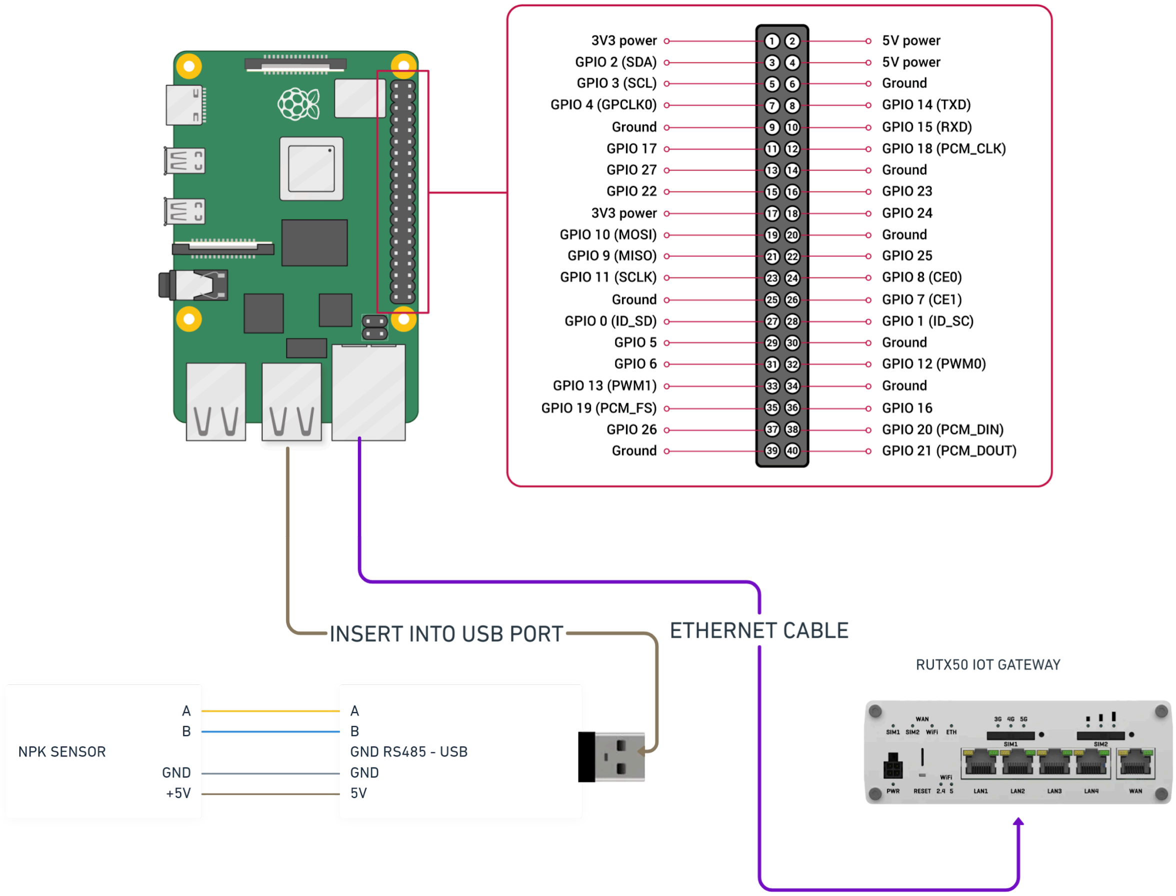
Check the wiring and connections via the below configuration:

Step 1 : Connect the sensors with the Iot board according to the wiring diagram below . Connect the NPK Sensor via a USB port to the Iot board .



Step 2: Connect the IoT board to RUTX via a LAN cable.

Step 3: Power on the IoT board and RUTX.The sensor data will be sent to the IoT application on the MEC server.

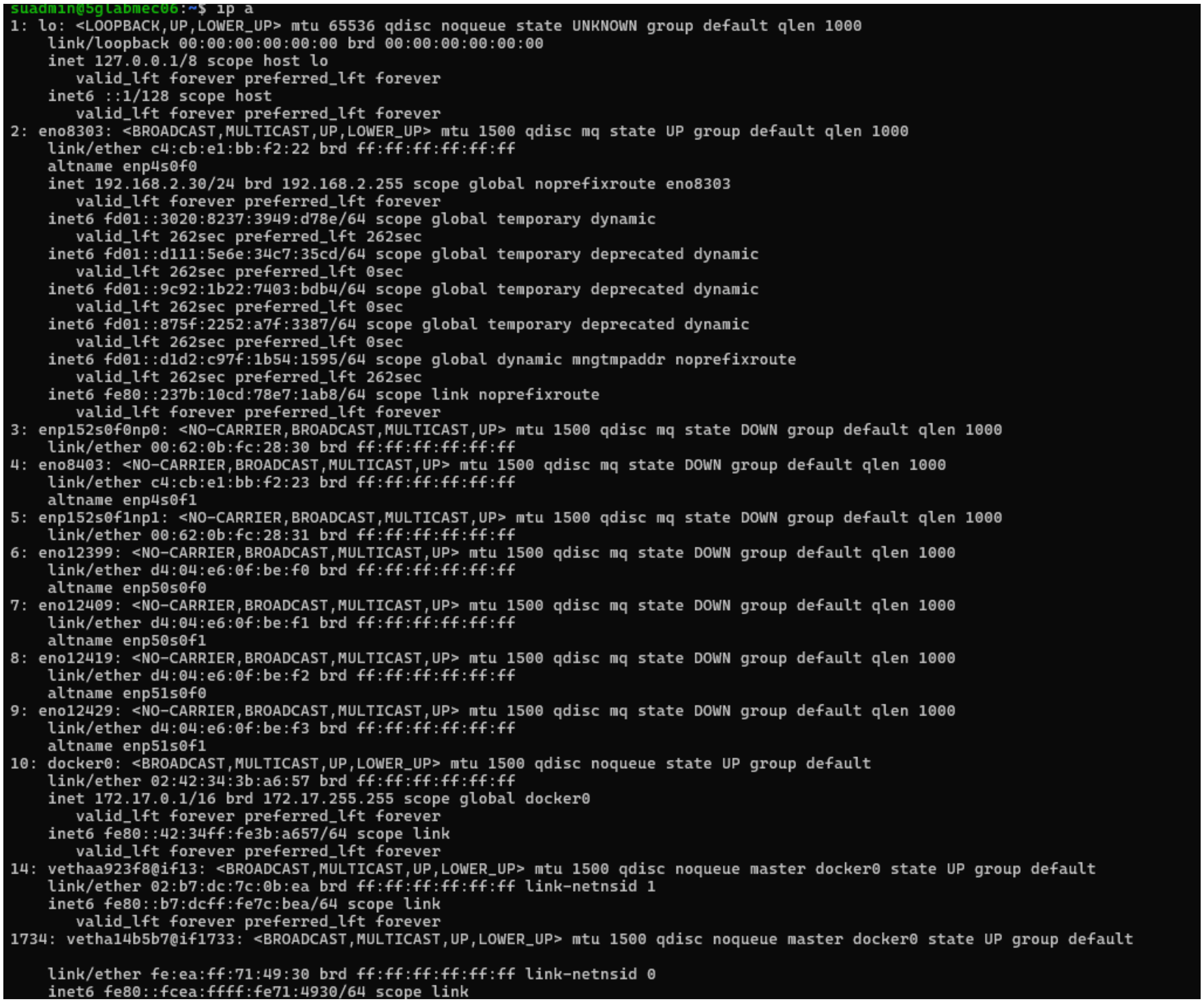

3V3 power 1 2 5V power
GPIO 2 (SDA) 3 4 5V power
GPIO 3 (SCL) 5 6 Ground
GPIO 4 (GPCLK0) 7 8 GPIO 14 (TXD)
Ground 9 10 GPIO 15 (RXD)
GPIO 17 11 12 GPIO 18 (PCM_CLK)
GPIO 27 13 14 Ground
GPIO 22 15 16 GPIO 23
3V3 power 17 18 GPIO 24
GPIO 10 (MOSI) 19 20 Ground
GPIO 9 (MISO) 21 22 GPIO 25
GPIO 11 (SCLK) 23 24 GPIO 8 (CE0)
Ground 25 26 GPIO 7 (CE1)
GPIO 0 (ID_SD) 27 28 GPIO 1 (ID_SC)
GPIO 5 29 30 Ground
GPIO 6 31 32 GPIO 12 (PWM0)
GPIO 13 (PWM1) 33 34 Ground
GPIO 19 (PCM_FS) 35 36 GPIO 16
GPIO 26 37 38 GPIO 20 (PCM_DIN)
Ground 39 40 GPIO 21 (PCM_DOUT)
INSERT INTO USB PORT
ETHERNET CABLE
RUTX50 IOT GATEWAY
NPK SENSOR
A
B
GND
+5V
A
B
GND RS485 - USB
GND
5V
WAN
SIM1 SIM2 WiFi ETH
3G 4G 5G
SIM1
SIM2
PWR RESET 2.4 5
WiFi
LAN1 LAN2 LAN3 LAN4 WAN

23.2. Finding the IP of the MEC Server

Log in to the MEC Server and open the Firefox web browser then input the following in the url search bar at the top

192.168.x.30:9090 (check the subnet of your mec server via **ip a** command in the terminal)

For example, the ip a command shows the ip 192.168.2.30 in the output below.



```
suadmin@5glabmec06:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno8303: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether c4:cb:e1:bb:f2:22 brd ff:ff:ff:ff:ff:ff
    altname enp4s0f0
    inet 192.168.2.30/24 brd 192.168.2.255 scope global noprefixroute eno8303
       valid_lft forever preferred_lft forever
    inet6 fd01::3020:8237:3949:d78e/64 scope global temporary dynamic
       valid_lft 262sec preferred_lft 262sec
    inet6 fd01::d111:5e6e:34c7:35cd/64 scope global temporary deprecated dynamic
       valid_lft 262sec preferred_lft 0sec
    inet6 fd01::9c92:1b22:7403:bdb4/64 scope global temporary deprecated dynamic
       valid_lft 262sec preferred_lft 0sec
    inet6 fd01::875f:2252:a7f:3387/64 scope global temporary deprecated dynamic
       valid_lft 262sec preferred_lft 0sec
    inet6 fd01::d1d2:c97f:1b54:1595/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 262sec preferred_lft 262sec
    inet6 fe80::237b:10cd:78e7:1ab8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp152s0f0np0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether 00:62:0b:fc:28:30 brd ff:ff:ff:ff:ff:ff
4: eno8403: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether c4:cb:e1:bb:f2:23 brd ff:ff:ff:ff:ff:ff
    altname enp4s0f1
5: enp152s0f1np1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether 00:62:0b:fc:28:31 brd ff:ff:ff:ff:ff:ff
6: eno12399: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether d4:04:e6:0f:be:f0 brd ff:ff:ff:ff:ff:ff
    altname enp50s0f0
7: eno12409: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether d4:04:e6:0f:be:f1 brd ff:ff:ff:ff:ff:ff
    altname enp50s0f1
8: eno12419: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether d4:04:e6:0f:be:f2 brd ff:ff:ff:ff:ff:ff
    altname enp51s0f0
9: eno12429: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether d4:04:e6:0f:be:f3 brd ff:ff:ff:ff:ff:ff
    altname enp51s0f1
10: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:34:3b:a6:57 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:34ff:fe3b:a657/64 scope link
       valid_lft forever preferred_lft forever
14: vethaa923f8@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether 02:b7:dc:7c:0b:ea brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::b7:dcff:fe7c:bea/64 scope link
       valid_lft forever preferred_lft forever
1734: vetha14b5b7@if1733: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default

    link/ether fe:ea:ff:71:49:30 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::fcea:ffff:fe71:4930/64 scope link
```

Reference Links

Teltonika (IoT Gateway)

https://drive.google.com/drive/folders/1uVe29Didf0-PSPnSRIb1AdlzT_q40561

Evaluation Board

https://drive.google.com/drive/folders/1KBIkPjCHgzdvitOxl6EnvhrCHYr9i6pr

Sparsh Camera

https://drive.google.com/drive/folders/1jmtDA5i05j3RRb6M2R_ZVsMh2tfn_Ex1

Kenstel CPE

https://drive.google.com/drive/folders/1ygIlXAIkFjxaMr676uwRVPUvHM3aHBNa

Ajna AR/VR

https://drive.google.com/drive/folders/1JWNksbGrYm9uH9yDRuIqw34Jn46MtPa0

Suparna 5G Mini Drone

https://drive.google.com/drive/folders/1L6Xw51kU8uH-B9RQW-4KpCellqnalkqg

Anexgate Firewall

https://drive.google.com/drive/folders/1yMkIZZE2B7ciPqCyeLxDn4Bg8bAXE4gH

VVDN RU

https://drive.google.com/drive/folders/1g_2KoXSS2lS3rWWCyyRTK8jWVj3ym_7n