



# RAN PM Installation and Setup Guide

This documentation provides step-by-step instructions for installing and setting up the **RAN PM** project, including its dependencies, containerized services, and verification steps. Each command is explained with comments for clarity.

## 1 Install Docker

To install Docker, follow these steps:

```
# Update the package list
sudo apt update

# Install required dependencies
sudo apt install apt-transport-https ca-certificates curl software-properties-common

# Add Docker's official GPG key
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearm  
or -o /usr/share/keyrings/docker-archive-keyring.gpg  
  
# Add the Docker repository to APT sources  
echo "deb [arch=$(dpkg --print-architecture) 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  
  
# Update package lists again  
sudo apt update  
  
# Install Docker  
sudo apt install docker-ce docker-ce-cli containerd.io  
  
# Start the Docker service  
sudo systemctl start docker  
  
# Enable Docker to start on boot  
sudo systemctl enable docker
```

## 2 Install Docker Compose

```
# Download the latest version of Docker Compose  
sudo curl -L "https://github.com/docker/compose/releases/latest/download/d  
ocker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compos  
e  
  
# Give execution permission to the Docker Compose binary  
sudo chmod +x /usr/local/bin/docker-compose  
  
# Verify installation  
docker-compose --version
```

## 3 Install Additional Prerequisites

Install **Helm, jq, OpenSSL, and Java (for Keytool)**:

```
# Add Helm's official GPG key
curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null

# Install transport-https for secure package downloads
sudo apt-get install apt-transport-https --yes

# Add the Helm repository to APT sources
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list

# Update package lists again
sudo apt-get update

# Install Helm, jq, and OpenSSL
sudo apt-get install helm jq openssl

# Install OpenJDK 21 (required for Java-based tools like Keytool)
sudo apt install openjdk-21-jre-headless
```

## 4 Clone the RAN PM Repository

```
# Clone the RAN PM repository from Gerrit
git clone "https://gerrit.o-ran-sc.org/r/nonrtric/plt/ranpm"

# Navigate into the cloned repository
cd ranpm
```

## 5 Build Components

### Build HTTPS Server

```
# Navigate to the HTTPS server directory  
cd ranpm/https-server  
  
# Build the HTTPS server without pushing the image to a registry  
.build.sh no-push
```

### Build PM rApp

```
# Navigate to the PM rApp directory  
cd ranpm/pm-rapp  
  
# Build the PM rApp without pushing the image to a registry  
.build.sh no-push
```

## 6 Deploy the RAN PM Containers

```
# Navigate to the Docker project directory  
cd ranpm/docker-proj  
  
# Set up and deploy the required Docker containers  
sudo ./docker-setup.sh
```

It takes time to download images and starting containers at the end it should look like this:

```

Activities Terminal
hamo@octo:~/Desktop/RAN_PM_1/ranpm/docker-proj
m 9:09 مارس 24 ⚡
httpsca:crt: OK
httpsca:crt: OK
Generating cert and key for server pm-https-server-8
subjectC = SE, ST = .., O = ERIC, OU = ERIC, CN = pm-https-server-8, emailAddress = a@example.com
Verifying cert towards ca cert
httpsca:crt: OK
httpsca:crt: OK
Generating cert and key for server pm-https-server-9
subjectC = SE, ST = .., O = ERIC, OU = ERIC, CN = pm-https-server-9, emailAddress = a@example.com
Verifying cert towards ca cert
httpsca:crt: OK
httpsca:crt: OK
Generating cert and key for server pm-https-server-10
subjectC = SE, ST = .., O = ERIC, OU = ERIC, CN = pm-https-server-10, emailAddress = a@example.com
Verifying cert towards ca cert
httpsca:crt: OK
httpsca:crt: OK
DONE!
Starting 1 dfc
Updating dfc1 truststore
Importing https ca cert to dfc1 truststore
Certificate was added to keystore
WARN[0000] /home/hamo/Desktop/RAN_PM_1/ranpm/docker-proj/docker-compose-dfc_gen.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
Docker Compose is running!
Container dfc-auth-token-file-dfc-1 Started
Container dfc1 Started
Starting producers
Container pm-auth-token-file-pm-producer-1 Started 0.3s
Container kafka-producer-pm-xml2json-0 Started 0.3s
Container pm-producer-json2kafka Started 0.3s
Starting http servers
WARN[0000] /home/hamo/Desktop/RAN_PM_1/ranpm/docker-proj/docker-compose-pm-https_gen.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
Docker Compose is running!
Container pn-https-server-8 Started 0.9s
Container pn-https-server-1 Started 0.8s
Container pn-https-server-3 Started 0.8s
Container pn-https-server-7 Started 0.8s
Container pn-https-server-10 Started 0.7s
Container pn-https-server-3 Started 0.8s
Container pn-https-server-6 Started 0.8s
Container pn-https-server-2 Started 0.9s
Container pn-https-server-4 Started 0.6s
Deleting files in shared volumes
script-home/ scripts
Deleting 0 file in /home/hamo/Desktop/RAN_PM_1/ranpm/docker-proj/shared-volume
DONE!
[hamster] ~/Desktop/RAN_PM_1/ranpm/docker-proj $ 

```

## 7 Start InfluxDB

```
# Navigate to the Docker project directory
cd ranpm/docker-proj
```

```
# Run the setup script for InfluxDB
sudo ./pmlog-setup.sh
```

It should look like this:

```

Activities Terminal
hamo@octo:~/Desktop/RAN_PM_1/ranpm/docker-proj
2.6.1: Pulling from library/influxdb
2ec5c85e47d1: Pull complete
a031d98e08e4: Pull complete
0f32858a30: Pull complete
a5c51a2a2a: Pull complete
fe0f6e0bcb8: Pull complete
2f1b9b55880: Pull complete
449a0a0a0a: Pull complete
f5eac85494ab: Pull complete
1c1cb330ff7: Pull complete
03a9a177805: Pull complete
Digest: sha256:c3431c094df12ef899ac39034e7b3acb90a6f4d87bd98102ca985041caa8
Status: Downloaded newer image for influxdb:2.6.1
docker.io/library/influxdb:2.6.1
Pulling ntnrtric-pmlog (nonrtric-sc.org:10801/o-ran-sc/nonrtric-pmlog:1.0.0)
1.0.0: Pulling from o-ran-sc/nonrtric-pmlog
759700526078: Already exists
3e18cd7391e: Pull complete
403a2a2a2a: Pull complete
3fb07df74970: Pull complete
aabbef4f9311: Pull complete
ad31e872940f: Pull complete
03a9a177805: Pull complete
03a9a177805: Pull complete
14959994497e: Pull complete
90bcbb5126c6: Pull complete
3bdd58a8ac0d: Pull complete
3f3d974532d4: Pull complete
897133333333: Pull complete
1be828720ef8: Pull complete
Digest: sha256:b031ec15751cfcfa31f2028edc7a129fd45136d578acd7ea2b2cc8dd74c2d26
nexus3.o-ran-sc.org:10801/o-ran-sc/nonrtric-pmlog:1.0.0
WARN[0808] /home/hamo/Desktop/RAN_PM_1/ranpm/docker-proj/docker-compose-influxdb_gen.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[1] Container influxdb2-0 started
Waiting for influxdb2-0 ready
XNEM+APxKcJ...-bkC3Pj0Mp179jwZATGBuvk4oD5AVB32W7tzQuK8eRtbRGs66VQn7zUGZZou9zUdU-4G-A==

Attempting to create clients ntnrtric-pmlog for realm: nonrtric-realm
Creating client ntnrtric-pmlog
Client token: eyJhbGciOi...
OK
Admin token: eyJhbGciOi...
OK
Attempting to generate secret for clients ntnrtric-pmlog in realm nonrtric-realm
Client Id for client ntnrtric-pmlog in realm nonrtric-realm: 01659947-a2d-496a-9307-0ae41da0f1e4
Creating secret
Client secret for client ntnrtric-pmlog in realm nonrtric-realm: ZFCFR0Kj53SKKFAKuunmcgolFbBebvq
0...
WARN[0808] /home/hamo/Desktop/RAN_PM_1/ranpm/docker-proj/docker-compose-pmlog_gen.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[2] Running 2/2
Container pmlog-auth-token-file-pmlog-0-1 started
Container pmlog-0-1 started
* [root] -> /root/RAN_PM_1/ranpm/docker-proj $ 

```

## 8 Verify Running Containers

```
# List all running and stopped containers
sudo docker ps -a
```

## Important Container Ports

- **Redpanda Console** → 8780
- **InfluxDB** → 8086
- **Control Panel** → 8088
- **Keycloak** → 8462
- **VEscollector** → 8080
- **MinIO** → 9001

It should look like this:

## 9 Create a Kafka Topic in Redpanda

```
# Navigate to the scripts directory inside the Docker project
cd ranpm/docker-proj/scripts
```

```
# Push sample data to a Kafka topic
sudo ./push-to-file-ready-topic.sh 1 2 3 4 5
```

(*The numbers refer to: <node-count> <num-of-events> <node-name-base> <file-extension> <num-servers>*)

## ◆ Accessing Different Services

# 1 Admin Tool for Keycloak

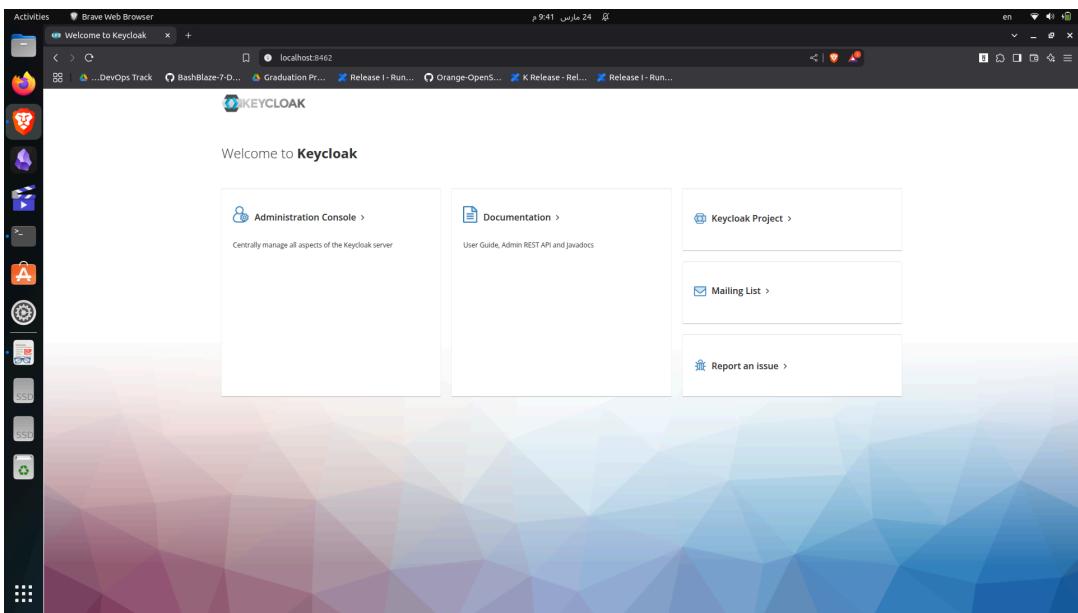
Keycloak is an **identity and access management (IAM) solution** used for authentication and authorization.

- Open a browser and go to:

<http://localhost:8462>

- Default credentials:

- **Username:** admin
  - **Password:** admin



## 2 Redpanda Console

Redpanda is a **high-performance Kafka-compatible streaming platform**. The **Redpanda Console** allows you to **view topics, consumers, and messages**.

- Open a browser and go to:

```
http://localhost:8780
```

After running all of these go to the localhost of redpandas to view the topics :

Redpanda

Overview

Topics

Schema Registry

Consumer Groups

Security

Quotas

Connectors

Reassign Partitions

Create Topic

Name	Partitions	Replicas	CleanupPolicy	Size
collected-file	1	1	delete	869 B
file-ready	1	1	delete	2.74 kB
json-file-ready-kp	1	1	delete	273 B
json-file-ready-kpadp	1	1	delete	273 B
pmreports	1	1	delete	2.27 kB

Total 5 items < 1 > 50 / page

Redpanda Console (Platform Version v23.0) (built March 22, 2023) 8987CD4

## Example of files in topics :

Offset Partition Timestamp Key Value

1 0 24/03/2025, 21:15:28 - {"productName": "RnNode", "vendorName": "Ericsson", "lastEpochMicrosec": 1742841926000000, "sourceName": "3-0"}

Key Value Headers

None (0 B) JSON (380 B) 1 Headers

Value

17 items

"productName": "RnNode", "vendorName": "Ericsson", "lastEpochMicrosec": 1742841926000000, "sourceName": "3-0", "startEpochMicrosec": 1742841926000000, "timeZoneOffset": "UTC+000", "compression": "gzip", "fileFormatType": "org.3gpp.32.435#measCollect", "fileFormatVersion": "V10", "name": "/3-0/A20208524\_1839+0200\_1845+0200\_3-0-1.4", "changeIdentifier": "PM\_MEAS\_FILES", "objectStoreBucket": "ropfiles"}

0 0 24/03/2025, 21:15:28 - {"productName": "RnNode", "vendorName": "Ericsson", "lastEpochMicrosec": 1742841926000000, "sourceName": "3-0"}

Key Value Headers

None (0 B) JSON (380 B) 1 Headers

Value

17 items

"productName": "RnNode", "vendorName": "Ericsson", "lastEpochMicrosec": 1742841926000000, "sourceName": "3-0", "startEpochMicrosec": 1742841926000000, "timeZoneOffset": "UTC+000", "compression": "gzip", "fileFormatType": "org.3gpp.32.435#measCollect", "fileFormatVersion": "V10", "name": "/3-0/A20208524\_1839+0200\_1845+0200\_3-0-0.4", "changeIdentifier": "PM\_MEAS\_FILES", "objectStoreBucket": "ropfiles"}

Save Messages

Total 2 items < 1 > 20 / page

## 3 MinIO Web

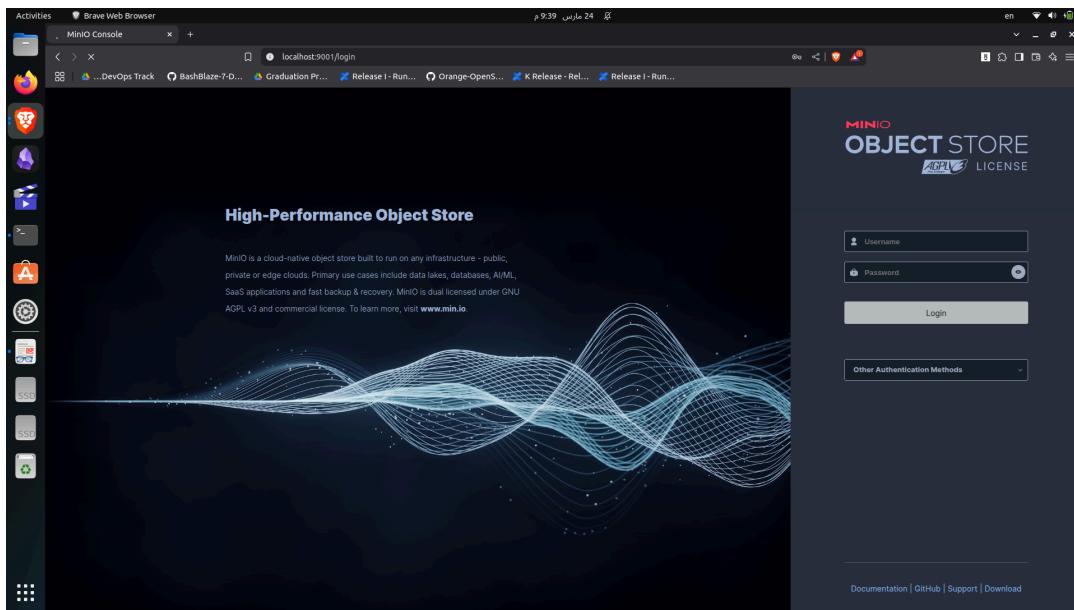
MinIO provides **object storage similar to Amazon S3**, used for **storing and retrieving files**.

- Open a browser and go to:

```
http://localhost:9001
```

- Default credentials:

- **Username:** admin
- **Password:** adminadmin



The screenshot shows the MinIO web interface's Object Browser. On the left, a sidebar lists 'User' and 'Administrator' sections. Under 'Administrator', 'Buckets' is selected, showing three buckets: 'pm-files.json', 'pm-files.json-locks', and 'r0files'. Each bucket has a row with columns for 'Name', 'Objects', 'Size', and 'Access'. The 'pm-files.json' bucket contains 2 objects, is 18.5 KB in size, and has R/W access.

This screenshot shows the contents of the 'pm-files.json' bucket. The top bar indicates the bucket was created on March 24, 2025, at 21:13:21 (GMT+2), has PRIVATE access, and contains 18.5 KB of 2 objects. The list shows two objects: 'A20250324.1815+0200-1830+0200\_3-0-0.4kafka-producer-pm-xml2json-0.json.gz' and 'A20250324.1830+0200-1845+0200\_3-0-1.4kafka-producer-pm-xml2json-0.json.gz', both modified today at 21:15 and each 9.2 KB in size.

## 4 InfluxDB Web Interface

InfluxDB is a **time-series database** used for storing performance metrics and logs.

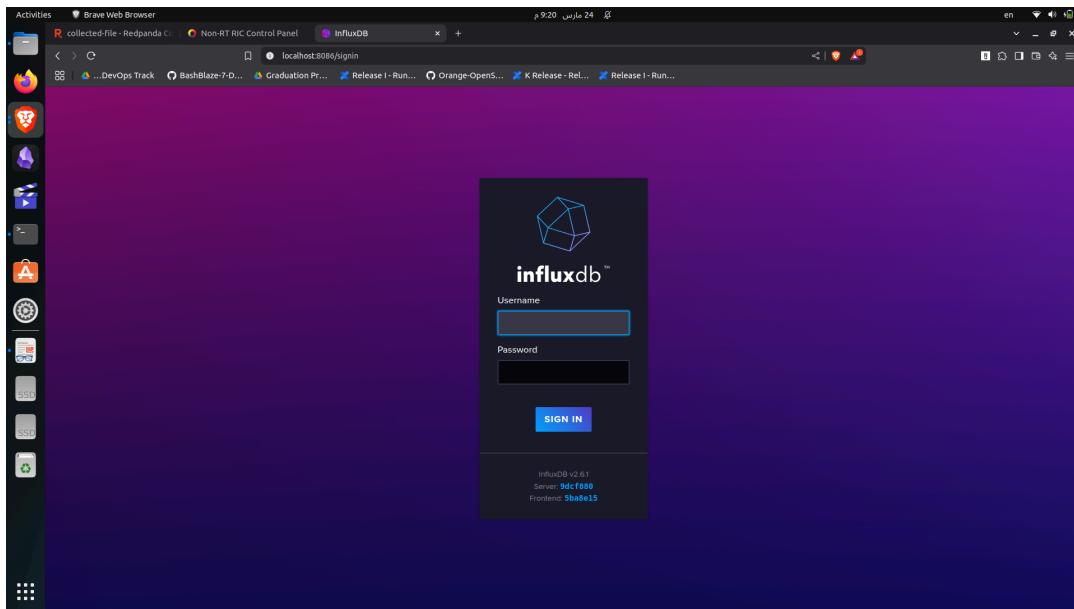
- Open a browser and go to:

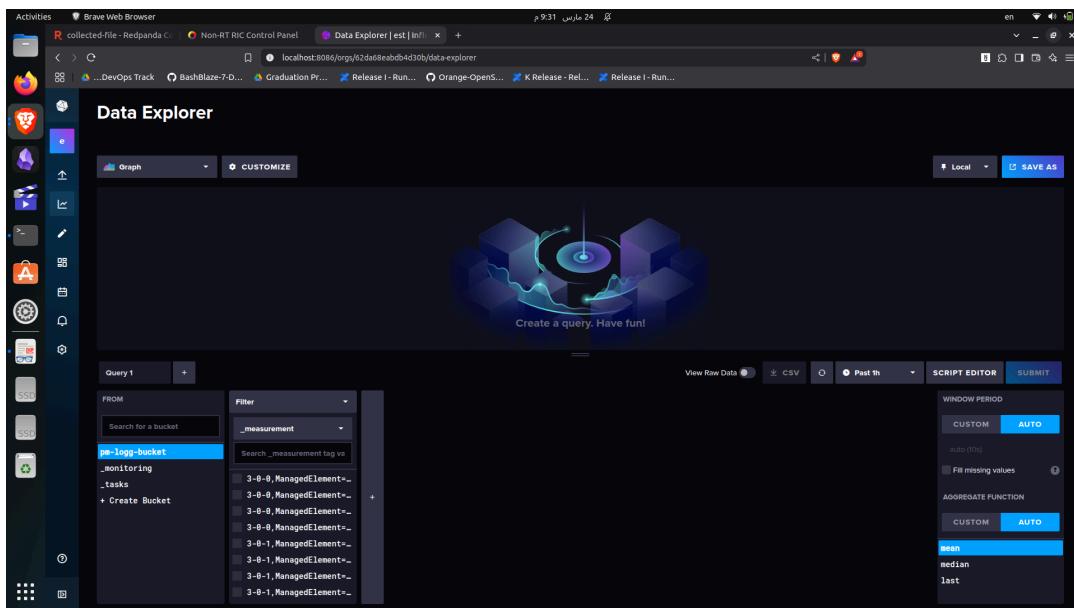
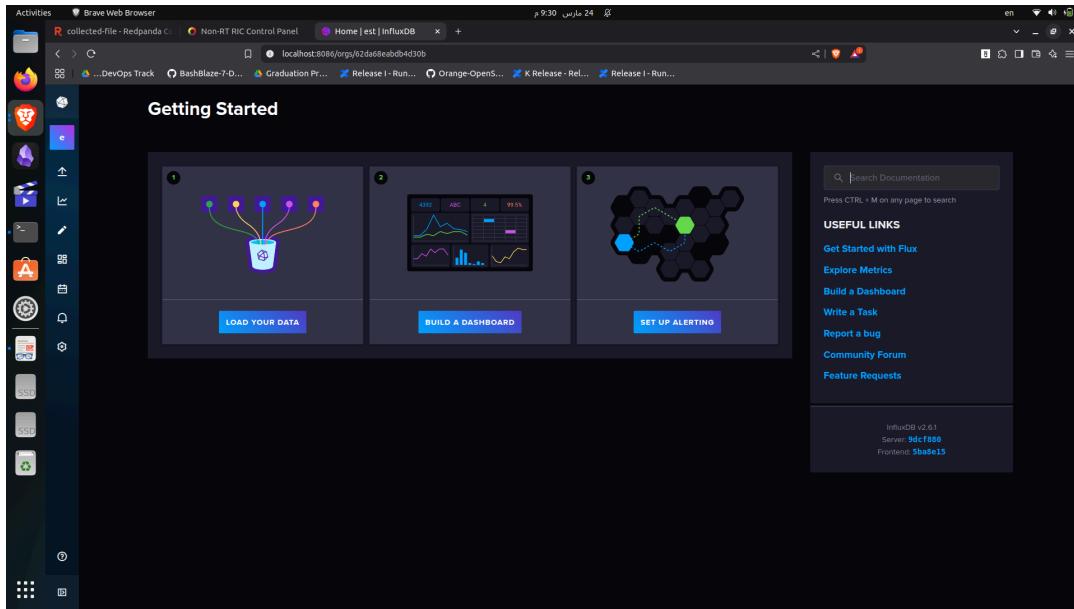
http://localhost:8086

- Default credentials:

- **Username:** admin

- **Password:** mySuP3rS3cr3tT0keN



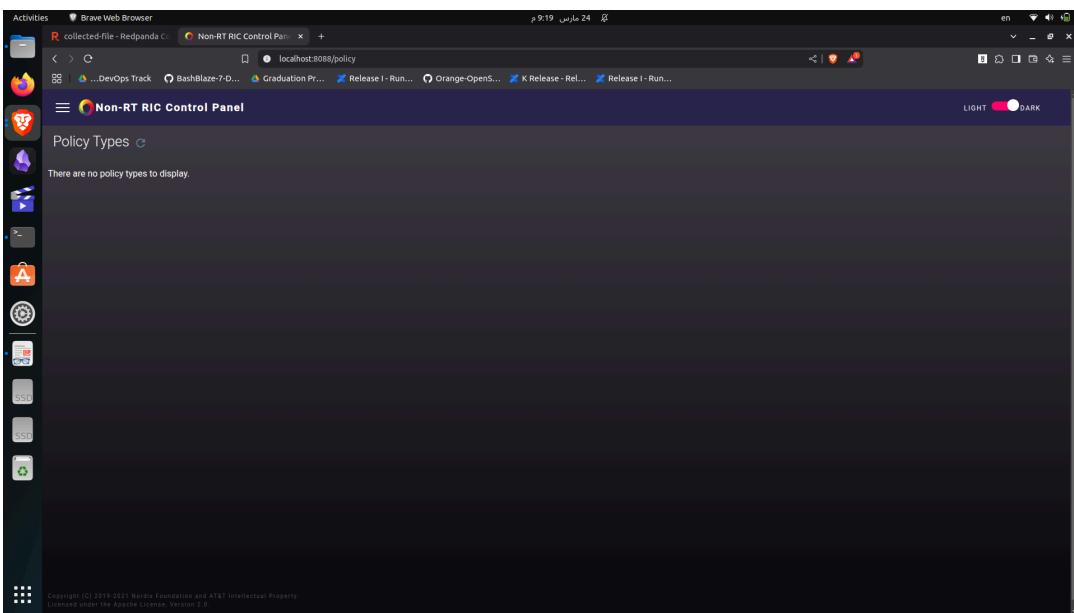
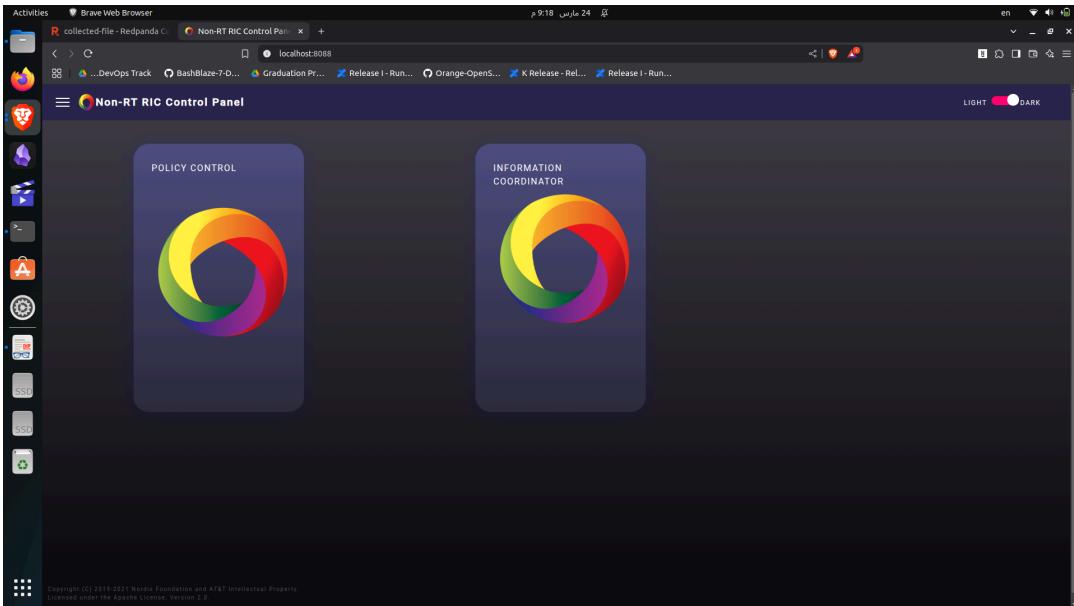


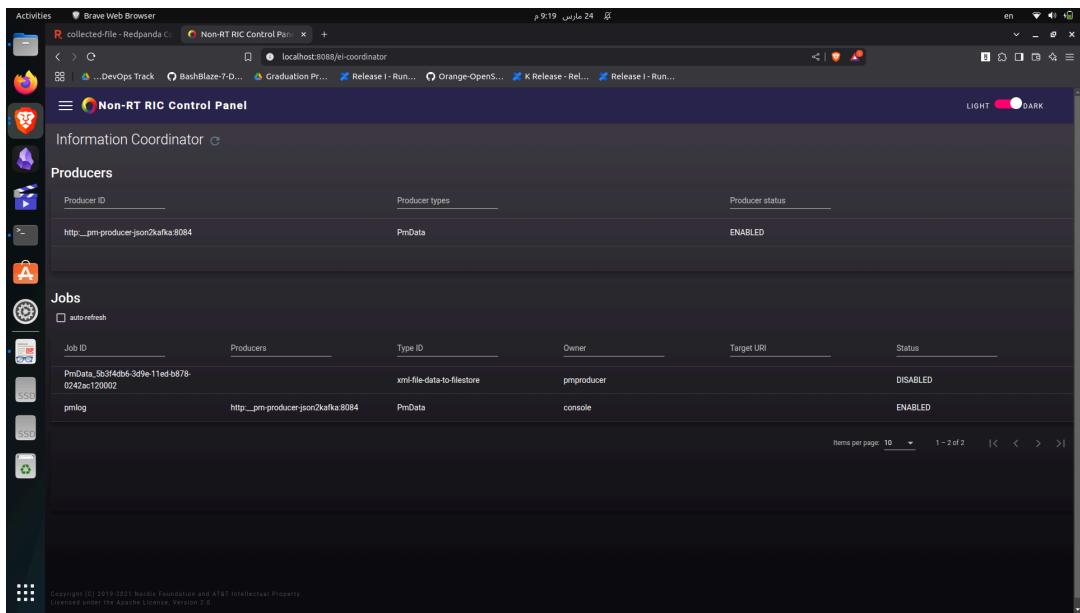
## 5 Control Panel

The Control Panel provides **administrative functions for managing the platform**.

- Open a browser and go to:

<http://localhost:8088>





## 6 VEScollector

VEScollector is used for **collecting telemetry and performance data** from the system. It plays a crucial role in **monitoring** and **data analysis**.

- Open a browser and go to:

```
http://localhost:8080
```

