| A black and white logo  Description automatically generated with low confidence | INTERNATIONAL TELECOMMUNICATION UNION  **TELECOMMUNICATION** **STANDARDIZATION SECTOR**  STUDY PERIOD 2022-2024 | | **Focus Group on AI Native Networks** | |
| --- | --- | --- | --- | --- |
| **AINN-I-xx** | |
| **Original: English** | |
| **Question(s):** | | N/A | Virtual, TBD 2024 | |
| **INPUT DOCUMENT** | | | | |
| **Source:** | | *Tenali* | | |
| **Title:** | | *Tenali Team - Report on* *ITU WTSA Hackathon 2024 – A midnight Robbery* | | |
| **Contact:** | | Raman | | E-mail: name@gmail.com |

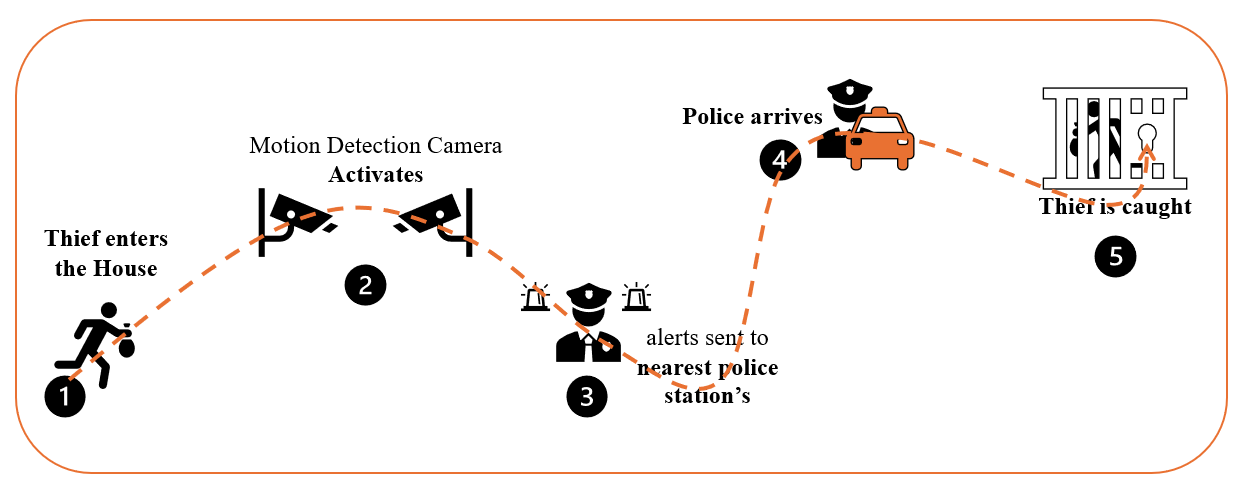
| **Abstract:** | This document contains the submission of a report for Tenali Team towards ITU WTSA Hackathon 2024 for use case *A midnight Robbery.* |
| --- | --- |

## Use case introduction: **“A midnight Robbery”**

Use case scenario involves a corporate / home surveillance. Night vision cameras are deployed in a high security area (home/enterprise). As an intruder enters the high security area, a motion detected workflow is triggered including object detection, authentication, authorization and alarm and intruder tracking. High priority connection resources are allocated to the surveillance connection till the intrusion situation is resolved with law enforcement involvement.

| *At midnight, when everyone is in deep sleep.* ***The thief moved silently****, thinking he was unnoticed. But as soon as he stepped inside, a motion-detection camera hidden in the corner sprang to life.*  *The camera, equipped with advanced video surveillance and QoS optimization, immediately began tracking the thief's every move. High-priority resources were allocated to ensure the camera's feed remained crystal clear, even in the low light. Within seconds,* ***an alert was sent to the nearest police station, along with live footage of the intruder****.*  *The thief is still unaware that he is getting caught by camera and alerts have been received to police.* |
| --- |

Consider the scene map below:



Phase 1: “*Thief Enters the House, Motion Detection Camera Activates”*: Intrusion is detected by motion detection cameras. The images/videos need to be processed for authentication and authorization.

Phase 2: “*Alert Sent to Nearest Police Station”*: Law enforcement is alerted in case of intrusion is confirmed. Continuous monitoring feeds are to be sent to law enforcement officers.

Note: If the police do not respond on time, then it will capture and give the live tracking of the thief by activating nearby cameras and base stations.

Phase 3: “*Continuous Monitoring happens and live tracking details sent to police”*: Officers arrive from local police stations and are provided with high priority connection on the scene.

Phase 4: “*Thief is Caught by Police”*: Intrusion situation is resolved, the intruder is apprehended and normal call priorities are restored.

## use case requirements

Requirement-1: It is critical that image detection, video analysis models need to be selected and deployed in the cloud or edge. Images/videos from the cameras need to be sent with low latency and high resolution to the models for inference and processing.

Requirement-2: It is critical that continuous allocation of resources for monitoring videos and images be sent to law enforcement.

Requirement-3: It is critical that mobility tracking for high priority users and location based prioritized resource allocation for high priority users for specific locations.

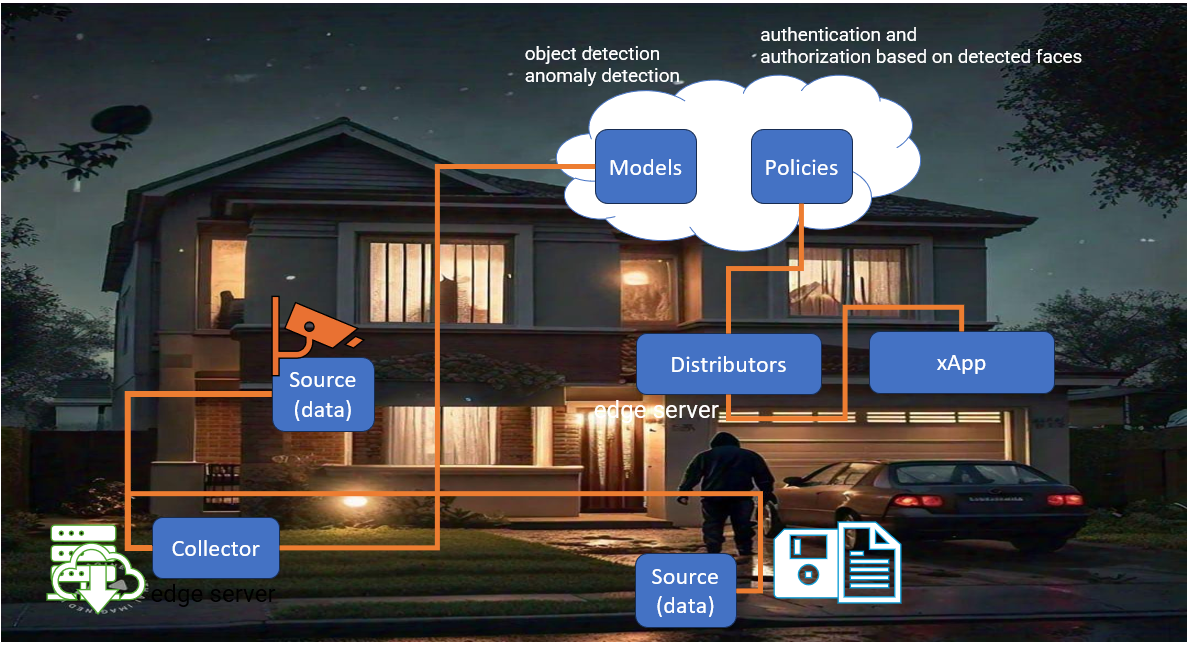
Requirement-4: It is critical to trigger based normalization and re-distribution of resources to low priority users once the emergency situation is resolved.

## PS1: pipeline design

* AI /ML Concept used is event analysis and anomaly detection
* **In Relation with ITU Y.3172** – for **submission**

**Requirements for this type of application?**

* SRC of data: Camera
* Collector: edge server
* Models: object detection, anomaly detection
* Policies: authentication and authorization based on detected faces
* Distributors : edge server
* Model inference Application (SINK): xApp



"AI generated pic from [Meta AI](https://www.meta.ai/?utm_source=ai_meta_site&utm_medium=web&utm_content=AI_nav&utm_campaign=April_moment)”

## PS2: xApp design

* Open RAN concept used is QoS optimization for video surveillance services
* **What is the role of xApp?**
* The xApp provides specific functions in the RAN such as resource allocation.
* This resource allocation mechanism needs to be validated before deploying in the field.
* **What is the role of Sandbox?**
* To verify and optimize the xApp based resource allocation, operators may use AI/ML and experiment with several strategies before deploying in the field.

**xApp design**:

code:

# Send an E2-like request to ask nodeB to send I/Q data

1 conn.send(f"E2-like request at {datetime.now().strftime('%H:%M:%S')}".encode('utf-8’))

2 # Example:The data in this case is I/Q data sourced from the RU (radio unit).

3 # This section of code will receive enough I/Q data to make one 10ms spectrogram.

4 data = conn.recv(16384)

5 if data:

6 log\_info(self, f"Receiving I/Q data...")

7 while len(data) < spectrogram\_size:

8 data += conn.recv(16384)

9# Point our global variable to the I/Q data we just received, and use our machine learning

10model to make a prediction.

11 current\_iq\_data = data

12 result = run\_prediction(self)

13 time.sleep(0.5)

14

15 # If there is interference, send a command to turn on adaptive MCS.

16 # This is a feature in srsRAN that we can leverage. When we turn it off, we set the MCS to a 17 #fixed value.

18 if result == 'Interference’:

19l og\_info(self, "Interference signal detected, sending control message to enable adaptive

20MCS")

21 conn.send(cmds['ENABLE\_ADAPTIVE\_MCS'])

22 elif result in ('5G', 'LTE'): #and last\_cmd == cmds['BASE\_STATION\_OFF’]:

23 log\_info(self, "Interference signal no longer detected, sending control message to disable 24

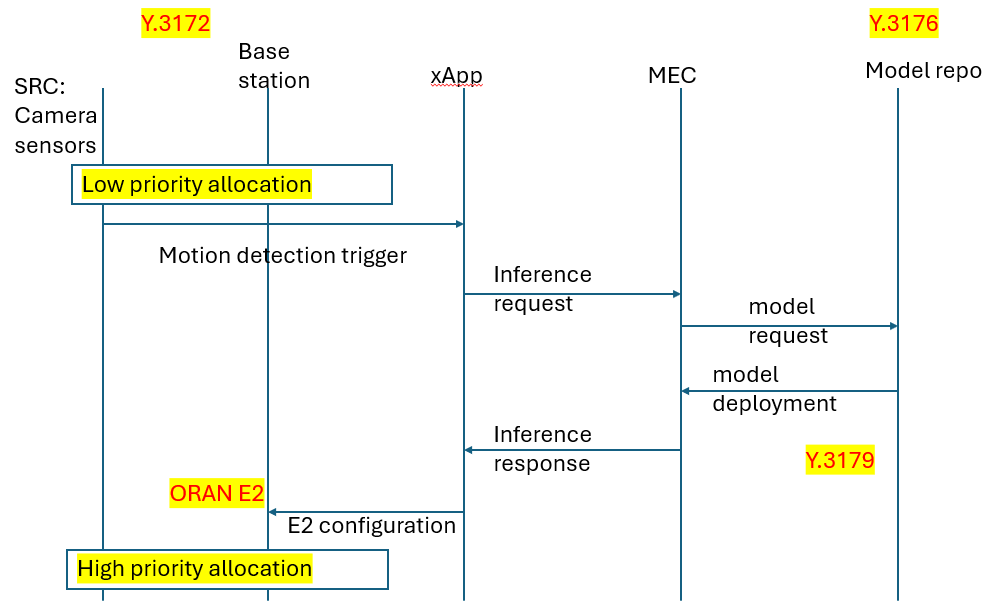
24 adaptive MCS")

25 conn.send(cmds['DISABLE\_ADAPTIVE\_MCS'])

**comments**:

1. configure the image / video tx by camera UEs
2. trigger based on model inference. continuous resource allocation for camera UEs.
3. trigger based on location of Law enforcement UEs. Allocate high priority for law enforcement UEs.
4. trigger based on edge server, reallocate resources for low priority UEs.

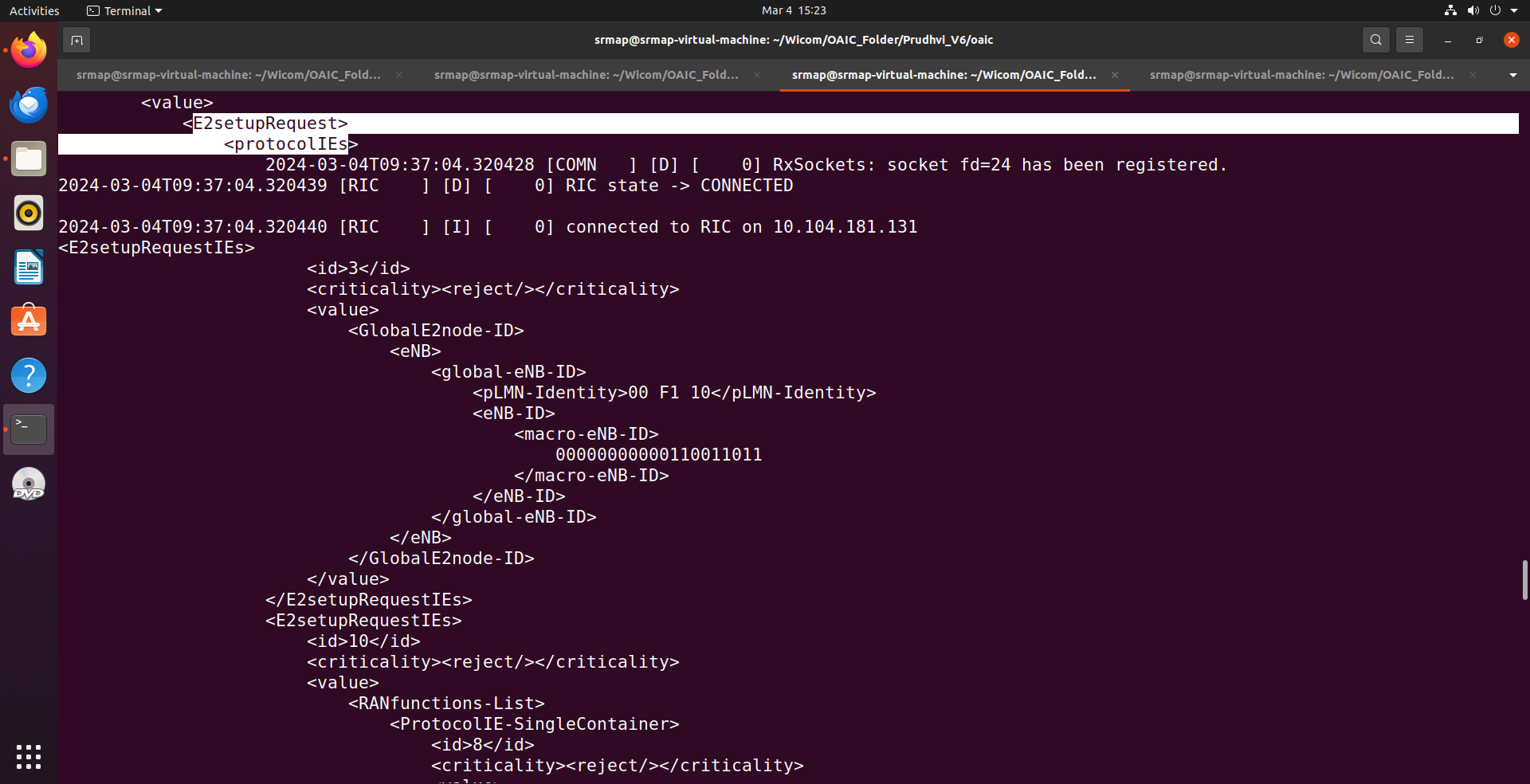
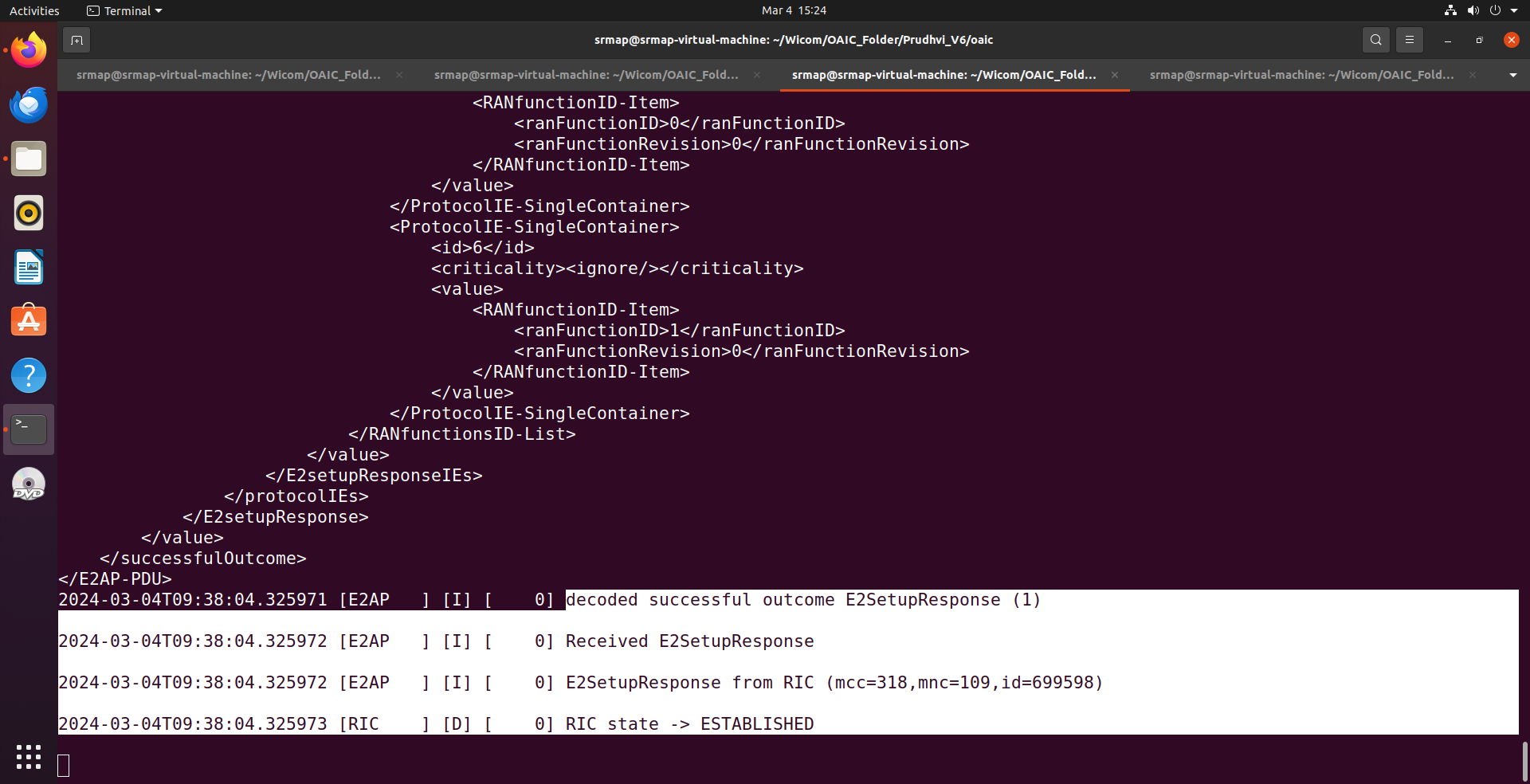
## Relation to Standards

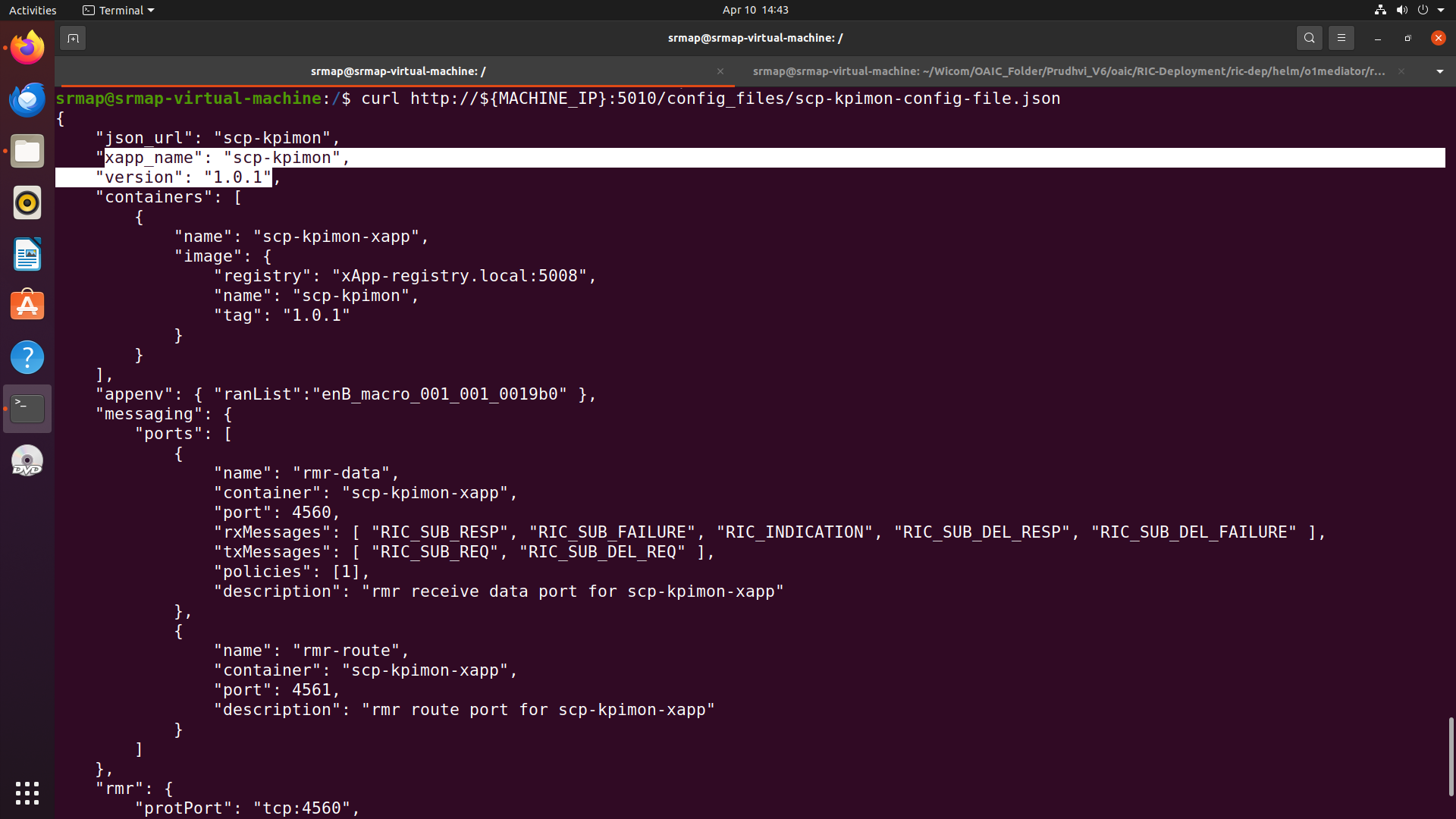


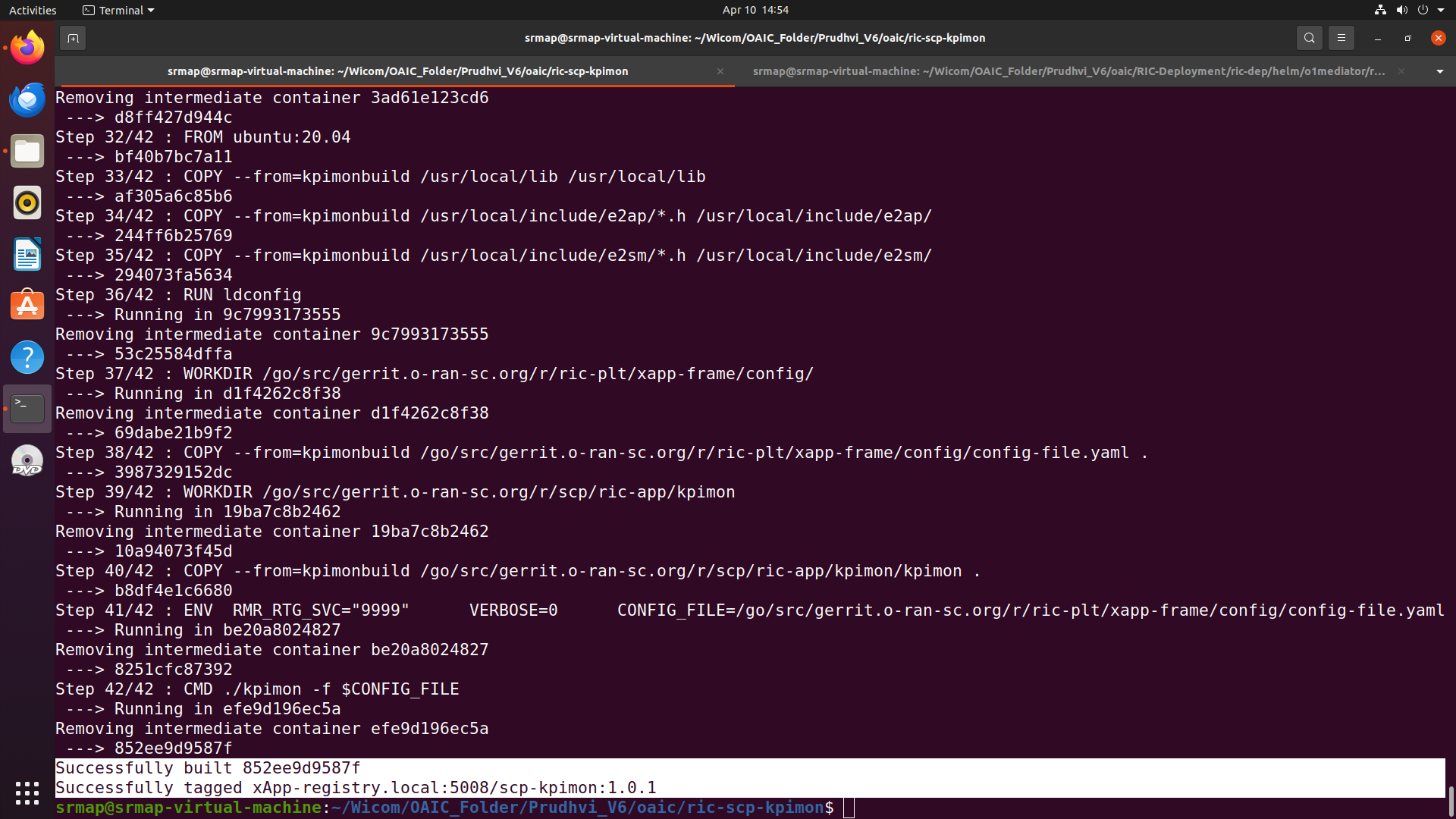
## Code submission details

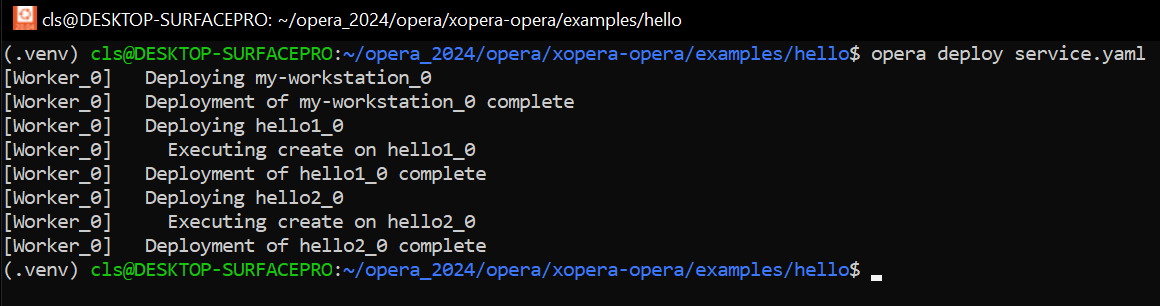
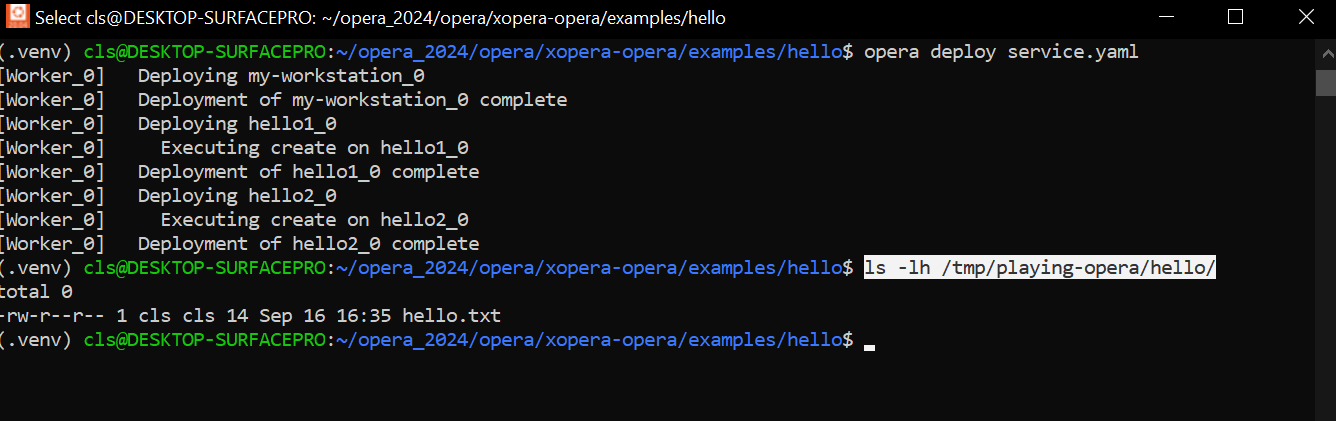
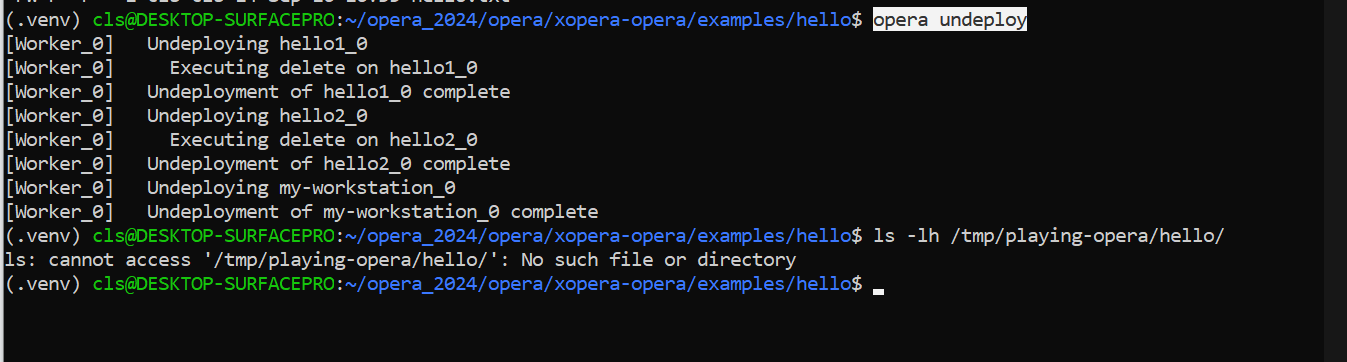
1. commit app.py or other files to your github repo.
2. refer to mentoring session 2 for details.
3. link to mentoring session-2: [ITU WTSA-24 AI Bharat 5G/6G Sandbox Hackathon Mentoring Session #2 - Zoom](https://itu.zoom.us/rec/play/Soe_lOKdxvDDkjXH8RDoCHxrwLpufRk2YB3448MKg2-yl9d-ZY9c4BUekVxxh1w-cemhr1KXteQLHPIT.3zHIN7xRxXgOU1Vi?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Fitu.zoom.us%2Frec%2Fshare%2Fn3RoIgMzGb9H4SFEK7hDKWseyaoJDwNQScRokjmul8l5evqdmYcRNV7cWYsBFjho.vIPyVp6N8X3GWAXH)
4. github repo link here: <https://github.com/CrashingGuru/ITU_WTSA_HACKATHON/tree/main/Example-4/Doc>

## Self-Testing results

* OAIC: [Getting Started — OAIC 0.1 documentation (openaicellular.github.io)](https://openaicellular.github.io/oaic/quickstart.html)
* ****
* ****





* xOpera: Ref:<https://xlab-si.github.io/xopera-docs/>
* 
* 
* 

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_