OAIC-C Installation

9 Aug 2023

Pratheek Upadhyaya
Dr. Joseph D. Gaeddert, PI

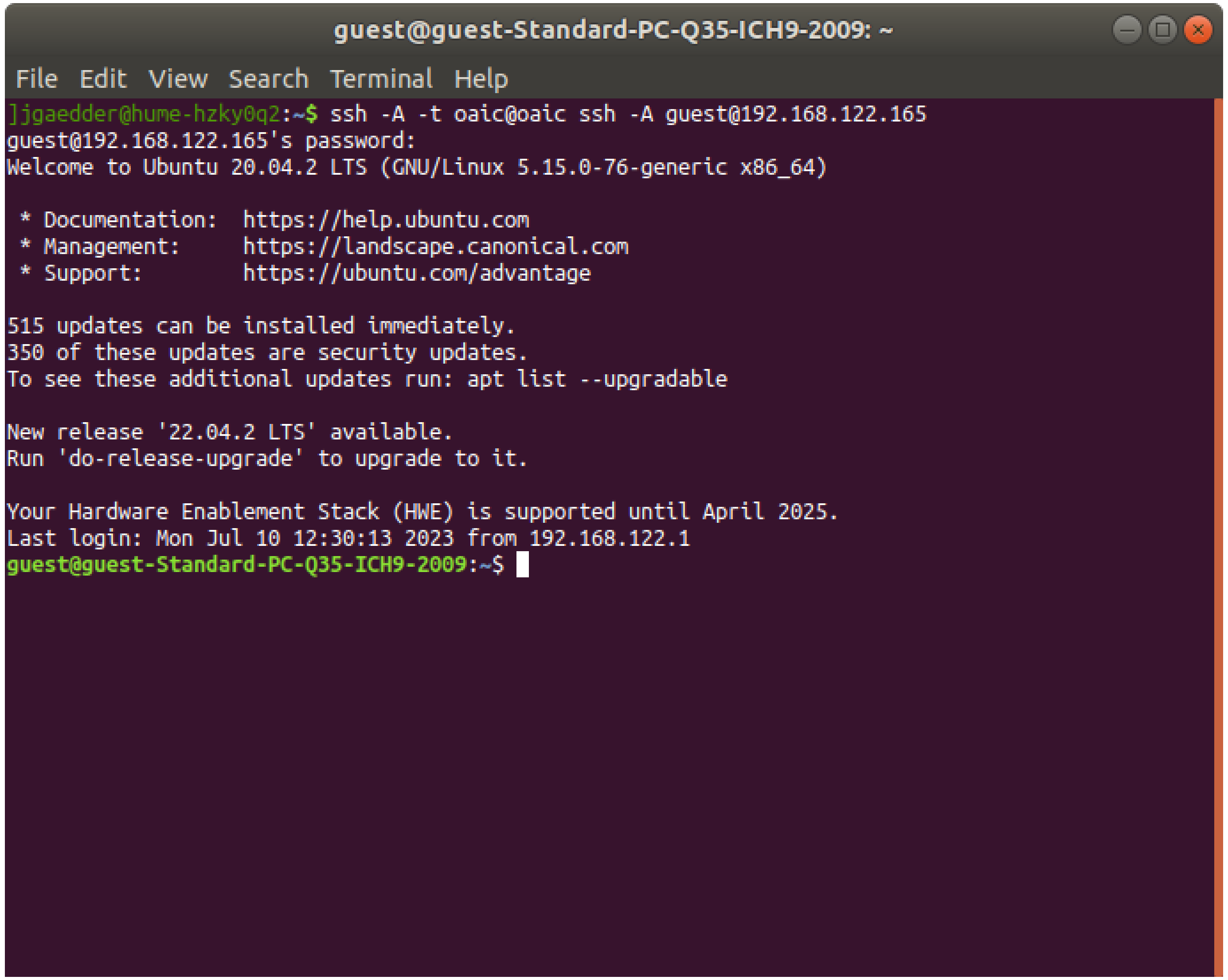
Open AI Cellular

National Science Foundation

Agenda for this Session

- Step 1: Setup (15 mins)
  - Clone repository
  - Install dependencies
- Step 2: O-RAN installation (45 mins)
- Step 3: srsRAN with e2 interface (30 mins)
- Step 4: Set up and Deploy 5G network (30 mins)

Connect to Virtual Machine


guest@guest-Standard-PC-Q35-ICH9-2009: ~
File Edit View Search Terminal Help

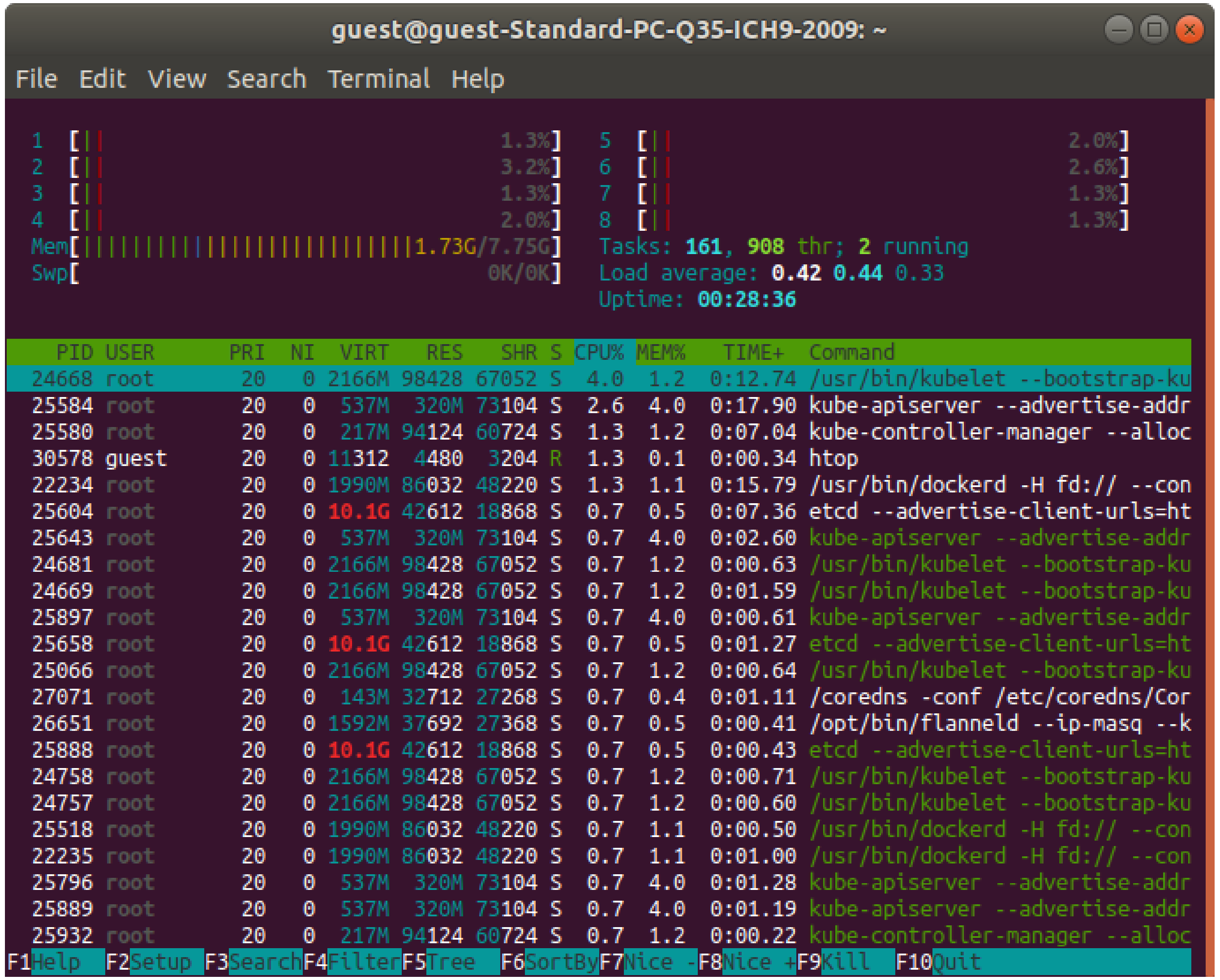
```
]jgaedder@hume-hzky0q2:~$ ssh -A -t oaic@oaic ssh -A guest@192.168.122.165
guest@192.168.122.165's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.15.0-76-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

515 updates can be installed immediately.
350 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Jul 10 12:30:13 2023 from 192.168.122.1
guest@guest-Standard-PC-Q35-ICH9-2009:~$
```

- We have a limited number of VMs running Ubuntu 20.04 on one of our servers
- Once provisioned, we will give you the IP address and account credentials
- IP addresses are assigned locally, so you will need to first connect through the main server
- You can do this with one command:

```
ssh -p 23401 -A -t guest@kermit.wireless.vt.edu \
  ssh -A guest@<provided.ip>
```

- The password for the server guest account is `guest123`
- The default password for the VM is `guest123`

Notes on Virtual Machines

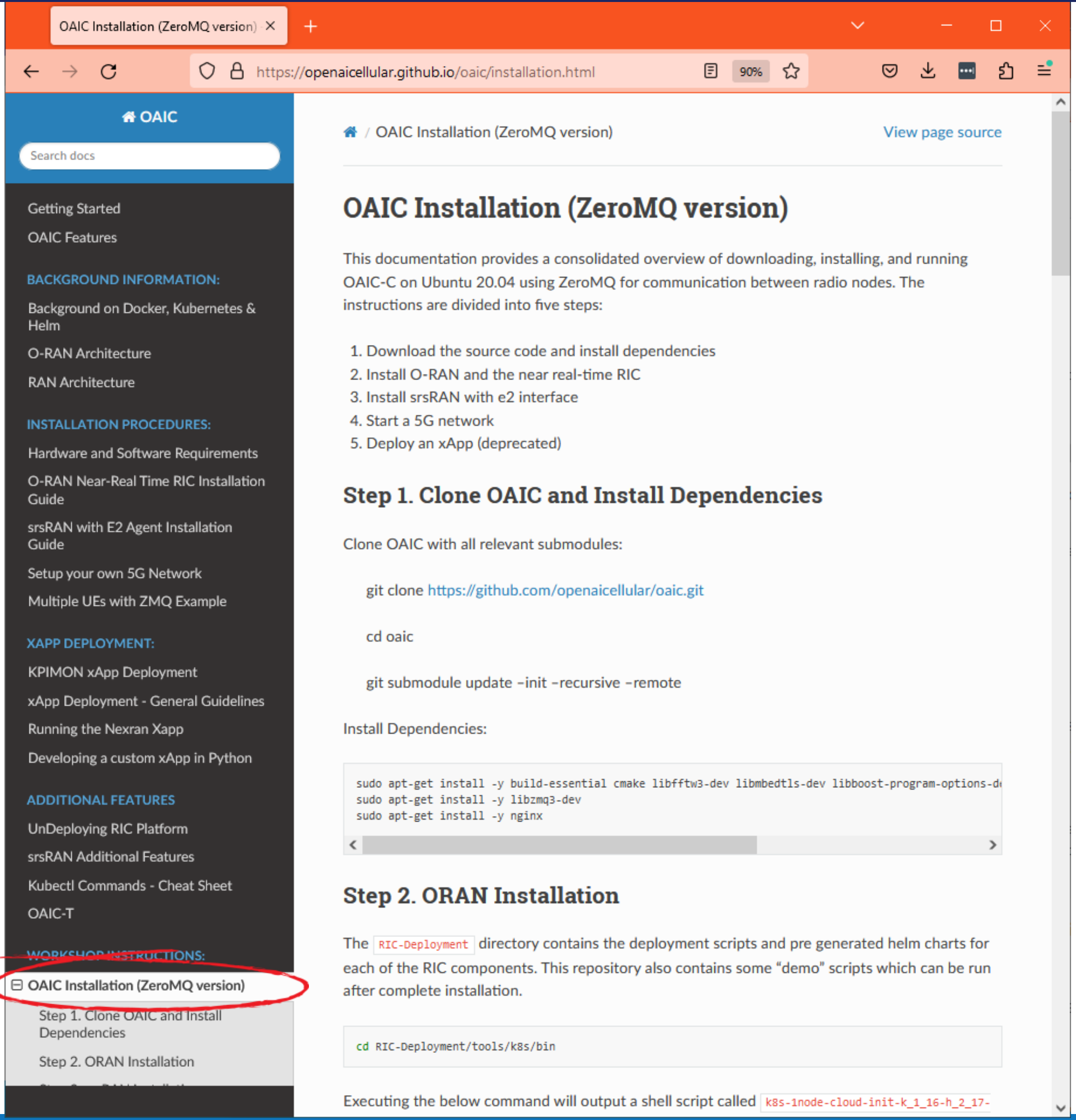


guest@guest-Standard-PC-Q35-ICH9-2009: ~

```
  1  [||                        1.3%]   5  [||                        2.0%]
  2  [||                        3.2%]   6  [||                        2.6%]
  3  [||                        1.3%]   7  [||                        1.3%]
  4  [||                        2.0%]   8  [||                        1.3%]
  Mem[||||||||||||||||||||||||||1.73G/7.75G]   Tasks: 161, 908 thr; 2 running
  Swp[                          0K/0K]   Load average: 0.42 0.44 0.33
                                         Uptime: 00:28:36

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  24668 root       20   0 2166M 98428 67052 S  4.0  1.2  0:12.74 /usr/bin/kubelet --bootstrap-ku
  25584 root       20   0  537M  320M 73104 S  2.6  4.0  0:17.90 kube-apiserver --advertise-addr
  25580 root       20   0  217M 94124 60724 S  1.3  1.2  0:07.04 kube-controller-manager --alloc
  30578 guest      20   0 11312  4480  3204 R  1.3  0.1  0:00.34 htop
  22234 root       20   0 1990M 86032 48220 S  1.3  1.1  0:15.79 /usr/bin/dockerd -H fd:// --con
  25604 root       20   0 10.1G 42612 18868 S  0.7  0.5  0:07.36 etcd --advertise-client-urls=ht
  25643 root       20   0  537M  320M 73104 S  0.7  4.0  0:02.60 kube-apiserver --advertise-addr
  24681 root       20   0 2166M 98428 67052 S  0.7  1.2  0:00.63 /usr/bin/kubelet --bootstrap-ku
  24669 root       20   0 2166M 98428 67052 S  0.7  1.2  0:01.59 /usr/bin/kubelet --bootstrap-ku
  25897 root       20   0  537M  320M 73104 S  0.7  4.0  0:00.61 kube-apiserver --advertise-addr
  25658 root       20   0 10.1G 42612 18868 S  0.7  0.5  0:01.27 etcd --advertise-client-urls=ht
  25066 root       20   0 2166M 98428 67052 S  0.7  1.2  0:00.64 /usr/bin/kubelet --bootstrap-ku
  27071 root       20   0  143M 32712 27268 S  0.7  0.4  0:01.11 /coredns -conf /etc/coredns/Cor
  26651 root       20   0 1592M 37692 27368 S  0.7  0.5  0:00.41 /opt/bin/flanneld --ip-masq --k
  25888 root       20   0 10.1G 42612 18868 S  0.7  0.5  0:00.43 etcd --advertise-client-urls=ht
  24758 root       20   0 2166M 98428 67052 S  0.7  1.2  0:00.71 /usr/bin/kubelet --bootstrap-ku
  24757 root       20   0 2166M 98428 67052 S  0.7  1.2  0:00.60 /usr/bin/kubelet --bootstrap-ku
  25518 root       20   0 1990M 86032 48220 S  0.7  1.1  0:00.50 /usr/bin/dockerd -H fd:// --con
  22235 root       20   0 1990M 86032 48220 S  0.7  1.1  0:01.00 /usr/bin/dockerd -H fd:// --con
  25796 root       20   0  537M  320M 73104 S  0.7  4.0  0:01.28 kube-apiserver --advertise-addr
  25889 root       20   0  537M  320M 73104 S  0.7  4.0  0:01.19 kube-apiserver --advertise-addr
  25932 root       20   0  217M 94124 60724 S  0.7  1.2  0:00.22 kube-controller-manager --alloc
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

- The server includes multiple VMs for workshop participants
  - 8 CPU cores
  - 8 GB RAM
  - 80 GB storage
- Lots of useful command-line tools already installed (`vim`, `htop`, `net-tools`, `tree`, `git`, `pip`, etc.)
- The `guest` user has sudo privileges for running certain commands

Installation Instructions (ZeroMQ Version)



OAIC Installation (ZeroMQ version)
https://openaicellular.github.io/oaic/installation.html
90%
OAIC
Search docs

Getting Started
OAIC Features

BACKGROUND INFORMATION:
Background on Docker, Kubernetes & Helm
O-RAN Architecture
RAN Architecture

INSTALLATION PROCEDURES:
Hardware and Software Requirements
O-RAN Near-Real Time RIC Installation Guide
srsRAN with E2 Agent Installation Guide
Setup your own 5G Network
Multiple UEs with ZMQ Example

XAPP DEPLOYMENT:
KPIMON xApp Deployment
xApp Deployment - General Guidelines
Running the Nexran Xapp
Developing a custom xApp in Python

ADDITIONAL FEATURES
UnDeploying RIC Platform
srsRAN Additional Features
Kubectl Commands - Cheat Sheet
OAIC-T

WORKSHOP INSTRUCTIONS:
OAIC Installation (ZeroMQ version)
Step 1. Clone OAIC and Install Dependencies
Step 2. ORAN Installation

OAIC Installation (ZeroMQ version) View page source

OAIC Installation (ZeroMQ version)

This documentation provides a consolidated overview of downloading, installing, and running OAIC-C on Ubuntu 20.04 using ZeroMQ for communication between radio nodes. The instructions are divided into five steps:

1. Download the source code and install dependencies
2. Install O-RAN and the near real-time RIC
3. Install srsRAN with e2 interface
4. Start a 5G network
5. Deploy an xApp (deprecated)

Step 1. Clone OAIC and Install Dependencies

Clone OAIC with all relevant submodules:

```
git clone https://github.com/openaicellular/oaic.git

cd oaic

git submodule update –init –recursive –remote
```

Install Dependencies:

```
sudo apt-get install -y build-essential cmake libfftw3-dev libmbedtls-dev libboost-program-options-d
sudo apt-get install -y libzmq3-dev
sudo apt-get install -y nginx
```

Step 2. ORAN Installation

The `RIC-Deployment` directory contains the deployment scripts and pre generated helm charts for each of the RIC components. This repository also contains some "demo" scripts which can be run after complete installation.

```
cd RIC-Deployment/tools/k8s/bin
```

Executing the below command will output a shell script called `k8s-1node-cloud-init-k_1_16-h_2_17-`

- A concise installation guide is set up on our website:
  - https://openaicellular.github.io/oaic/installation.html
  - (See screenshot at left)
  - This includes all the basic commands for installing OAIC-C from scratch on a base Ubuntu 20.04 image
- This tutorial will run through these instructions step by step
- Commands in this tutorial that you should run are highlighted in a green box:

```
ping www.openaicellular.org
```

Helpful Commands

- Changing directory (cd)
  - Enter a directory: `cd <directory name/path>`
  - Exit a directory: `cd ..`
  - Exit multiple directories: `cd ../../../..`
- List all files in current directory: *ls*
- Open a file: `vim <filename/path>`
  - Edit a file: Press `i`
  - Stop editing a file: `<Esc>`
  - Save a file: stop editing `<Esc>` and type `:w`
  - Exit a file: Stop editing `<Esc>` and type `:q`
  - Save and exit: Stop editing `<Esc>` and type `:wq`

Clone Repository

- OAIC is organized into several repositories
- “oaic.git” (https://github.com/openaicellular/oaic.git) is the top-level repo
- All of the supporting repositories are submodules that are pulled from oaic.git
- See the directory structure at right

```
git clone https://github.com/openaicellular/oaic.git
cd oaic
git submodule update --init --recursive --remote

tree -L 1 --dirsfirst
```

```
.
├── asn1c
├── docs
├── oaic-t
├── RIC-Deployment
├── ric-plt-e2
├── ric-scp-kpimon
├── srsRAN-e2
├── deployKPIMON.sh
├── generate_installation_script.py
├── LICENSE
├── makefile
├── README.md
├── requirements.txt
└── setup5GNetwork.sh
```

Install Dependencies

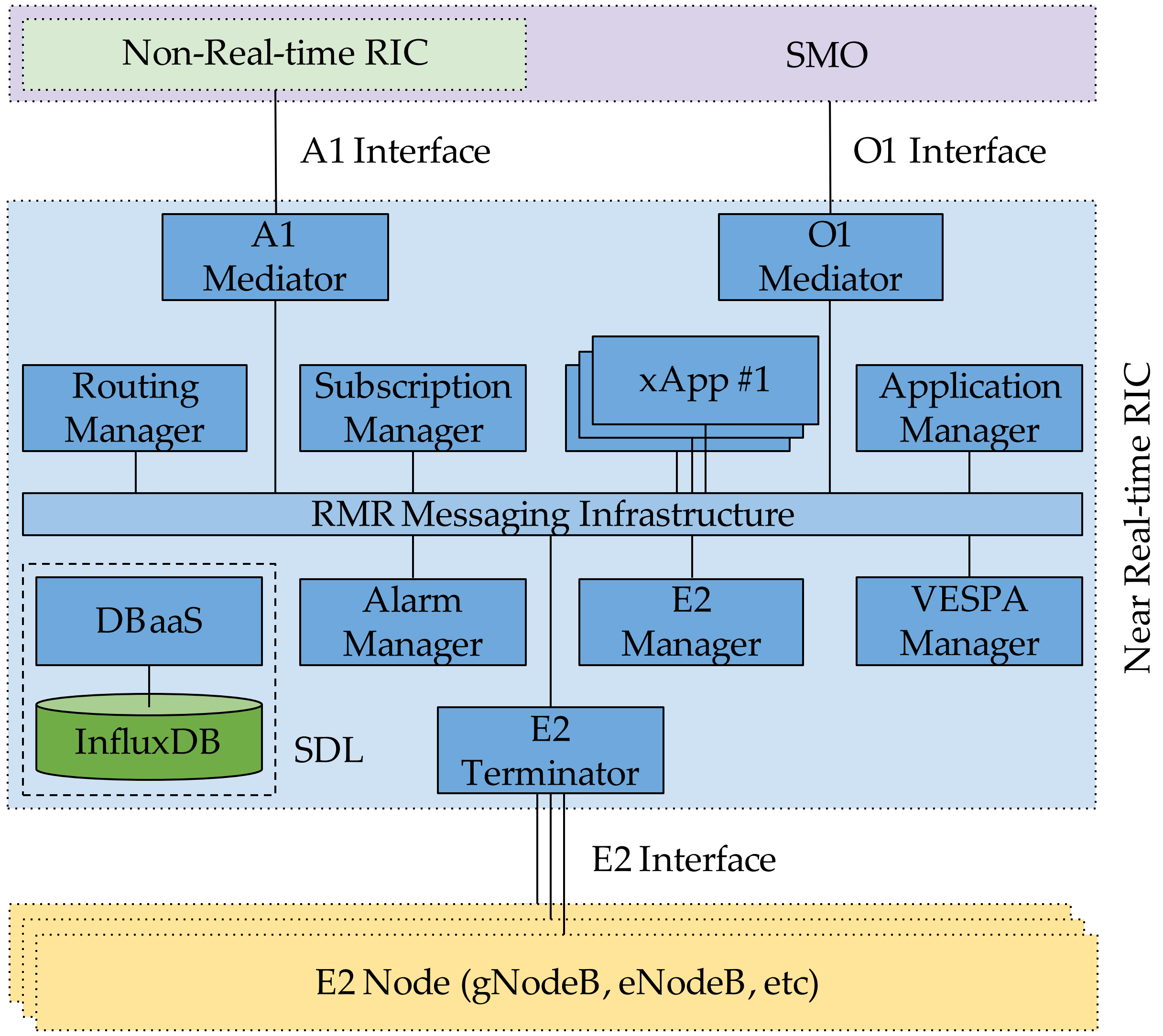
- OAIC is built on open-source software packages
- OAIC also relies on a number of open-source libraries and binaries
- For convenience, (most of) these can be installed up front using apt, a package management tool for Linux Debian and derivative distributions (such as Ubuntu)

```
sudo apt-get install -y build-essential cmake libfftw3-dev libmbedtls-dev
sudo apt-get install -y libzmq3-dev libboost-program-options-dev libconfig++-dev
sudo apt-get install -y nginx libsctp-dev libtool autoconf
```

Near-RT RIC Architecture

...Not a single piece of SW

- **Distributed** components
- **Isolated &** resource efficient design.
- **Microservice** architecture

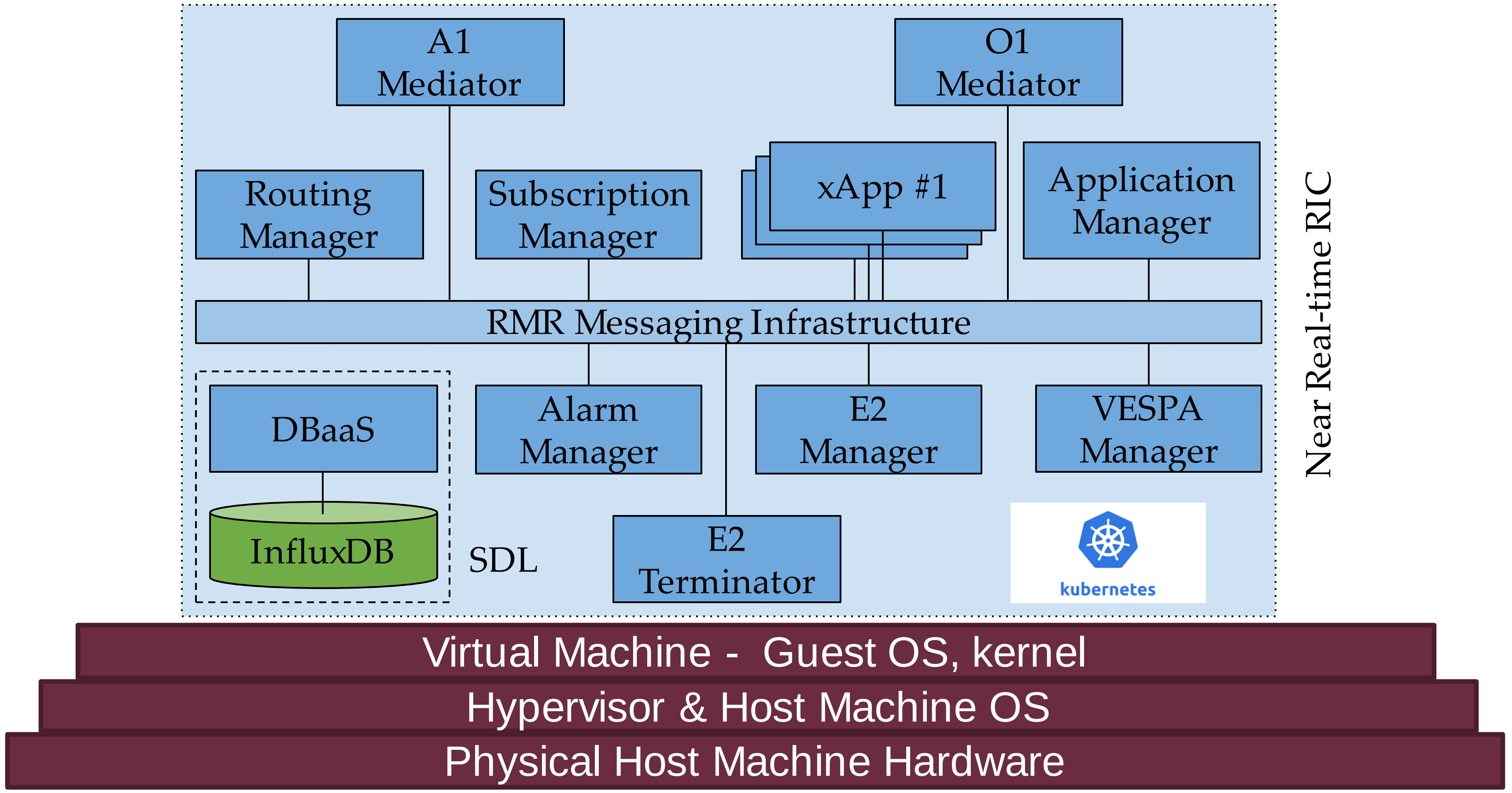


Credits: Dr. Joao Santos, CCI

Near-RT RIC Architecture

All of which:

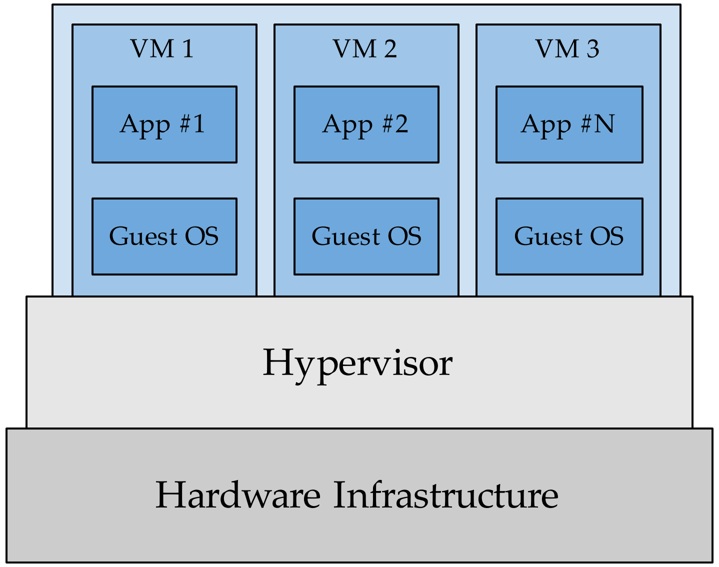
- Run as **Docker containers**
- Managed by a **Kubernetes cluster**

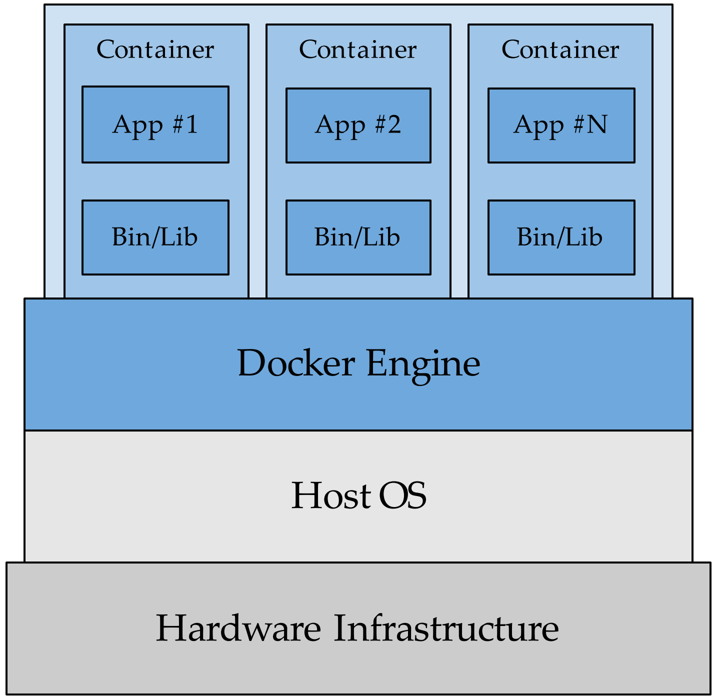


Credits: Dr. Joao Santos, CCI

Docker & Kubernetes

- Difference between Virtual Machines and Containers

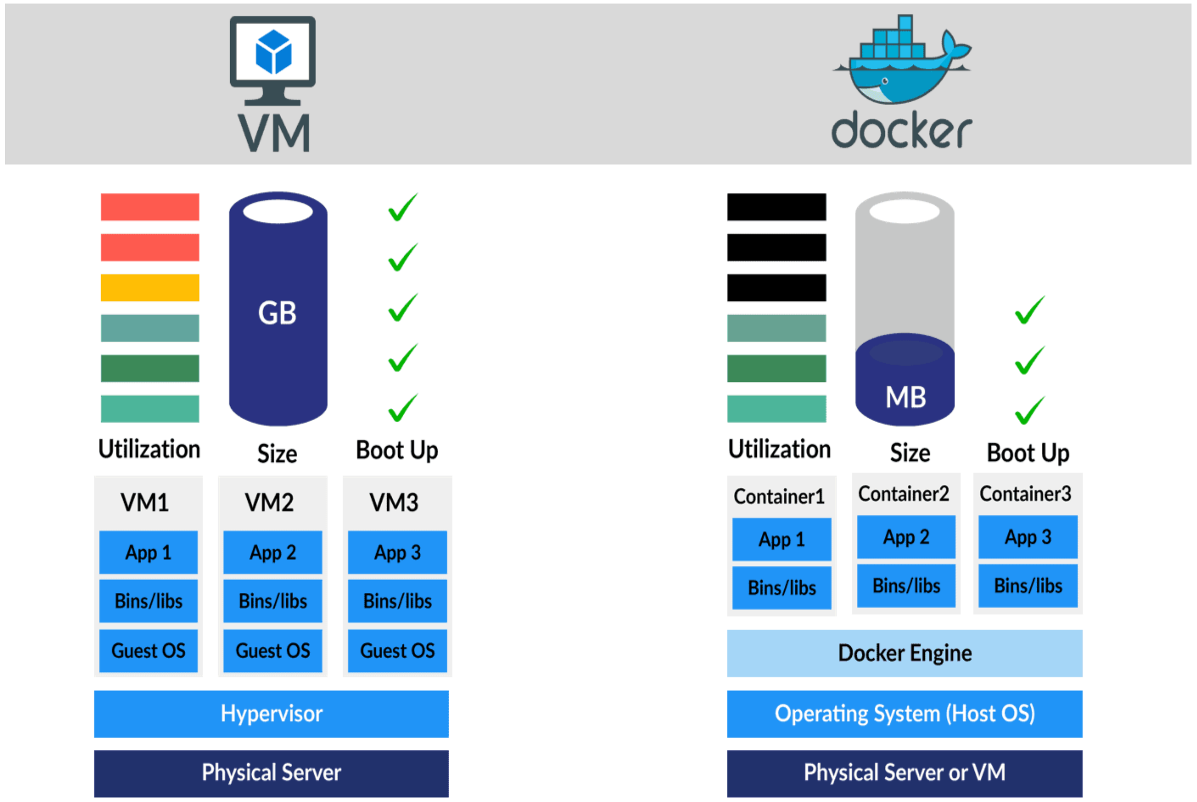




- Docker Containers are light weight while VMs are compute heavy.
- Isolation is better in VM due to dedicated resources, while docker uses the host OS kernel.
- Portability and efficiency of VMs is less compared to containers.

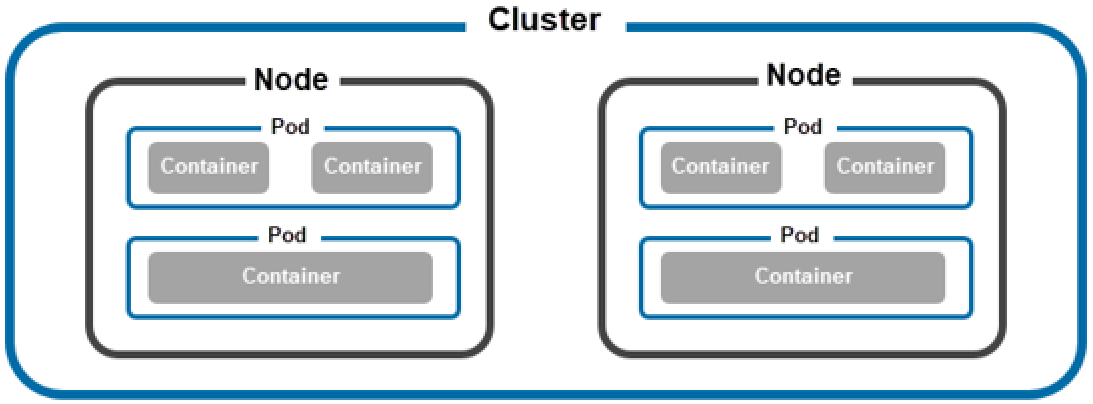
WHY Do We Need Docker?

- Each container can have different OS filesystem, use different libraries, and **run different applications**
- Isolated and secure environments.
  - *Portability and reproducibility.*
- Efficient Resource Usage.
  - *Mainly due to shared kernel with the host OS.*
- Scalability – Add or remove containers to handle usage variations.



Kubernetes

- Kubernetes **orchestrates** container deployments, their lifecycle and storage.
- **Kubernetes Pod:** A pod is a group of one or more container that run instances of an application.
- **Kubernetes Service:** Enables the group of pods to be assigned a name and unique IP address.
  - Expose an application deployed on a set of pods using a single endpoint.



We use only a limited number of features offered by Kubernetes – mainly resource management & stability.

Benefits of Using Kubernetes:

- Automated container orchestration and management
- Increased scalability and efficient resource management.
  - How is this different from the advantage docker provides in terms of scalability?
- Stability.

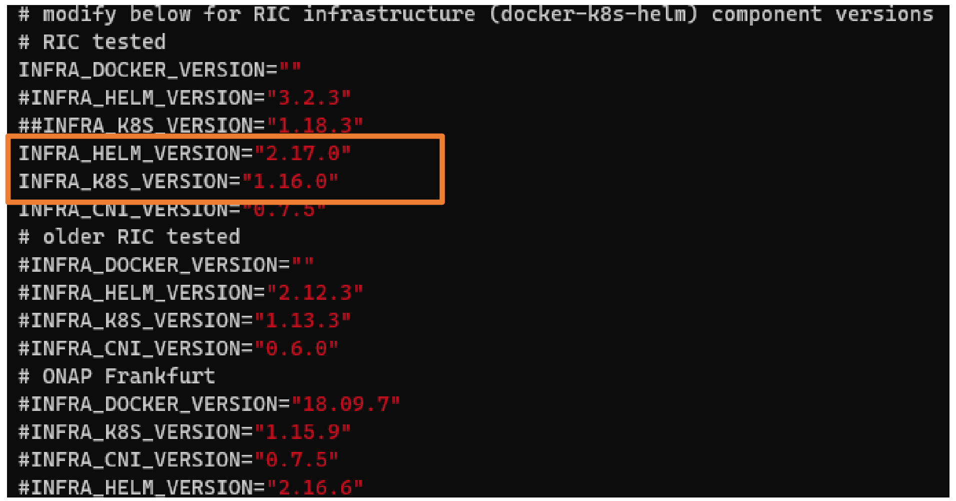
Exercise 1 : Install cloud computing platform

```
cd RIC-Deployment/tools/k8s/
tree -L 3 --dirsfirst
```

1 gen-cloud-init.sh script reads parameters from infra.rc, env.rc, openstack.rc

Task 1: Explore the file in **/etc/infra.rc**

Kubernetes version: 1.16
Helm Version: 2.17

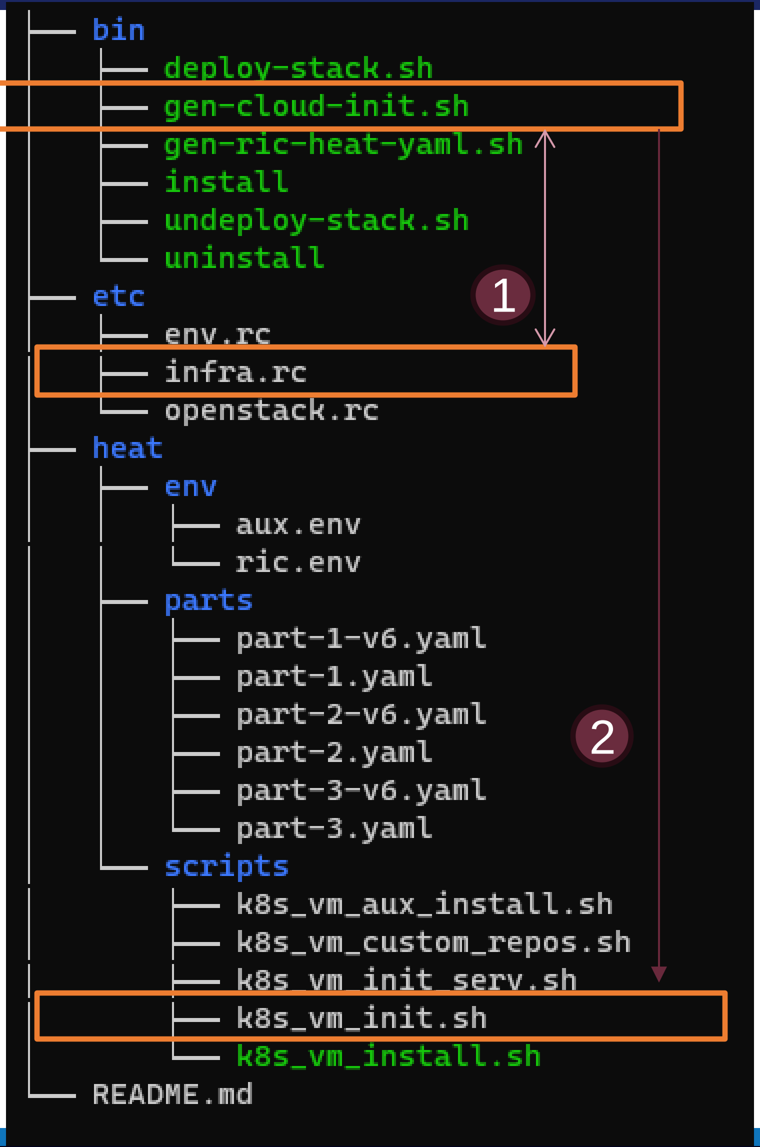


```
# modify below for RIC infrastructure (docker-k8s-helm) component versions
# RIC tested
INFRA_DOCKER_VERSION=""
#INFRA_HELM_VERSION="3.2.3"
##INFRA_K8S_VERSION="1.18.3"
INFRA_HELM_VERSION="2.17.0"
INFRA_K8S_VERSION="1.16.0"
INFRA_CNI_VERSION="0.7.5"
# older RIC tested
#INFRA_DOCKER_VERSION=""
#INFRA_HELM_VERSION="2.12.3"
#INFRA_K8S_VERSION="1.13.3"
#INFRA_CNI_VERSION="0.6.0"
# ONAP Frankfurt
#INFRA_DOCKER_VERSION="18.09.7"
#INFRA_K8S_VERSION="1.15.9"
#INFRA_CNI_VERSION="0.7.5"
#INFRA_HELM_VERSION="2.16.6"
```

```
cd etc/
vim infra.rc
Close the file (See  Tip)
cd ..
```

Tip

Exit a file – Press *:q*



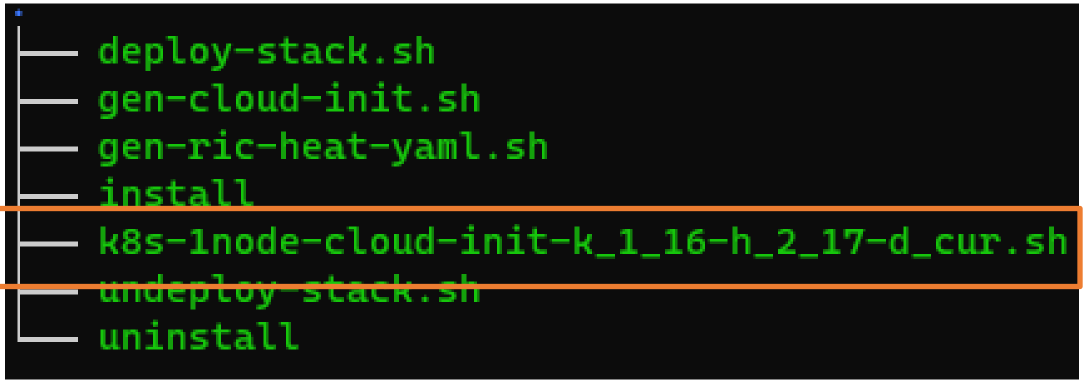
```
├── bin
│   ├── deploy-stack.sh
│   ├── gen-cloud-init.sh
│   ├── gen-ric-heat-yaml.sh
│   ├── install
│   ├── undeploy-stack.sh
│   └── uninstall
├── etc
│   ├── env.rc
│   ├── infra.rc
│   └── openstack.rc
├── heat
│   ├── env
│   │   ├── aux.env
│   │   └── ric.env
│   ├── parts
│   │   ├── part-1-v6.yaml
│   │   ├── part-1.yaml
│   │   ├── part-2-v6.yaml
│   │   ├── part-2.yaml
│   │   ├── part-3-v6.yaml
│   │   └── part-3.yaml
│   └── scripts
│       ├── k8s_vm_aux_install.sh
│       ├── k8s_vm_custom_repos.sh
│       ├── k8s_vm_init_serv.sh
│       ├── k8s_vm_init.sh
│       └── k8s_vm_install.sh
└── README.md
```

Exercise 1 : Install cloud computing platform

Task 2: Execute the installation script generation program *gen-cloud-init.sh*

```
cd bin/
./gen-cloud-init.sh
```

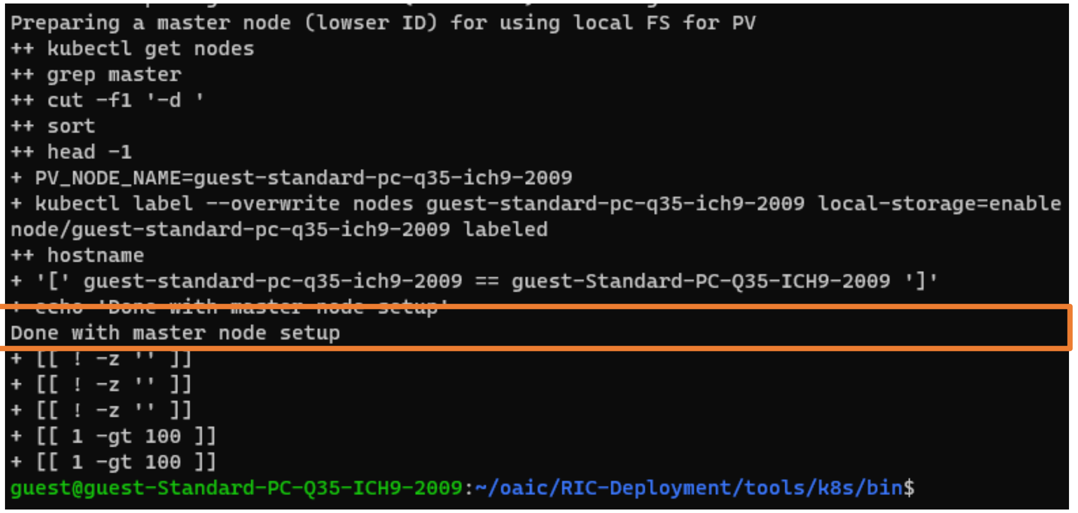
2 When executed *gen-cloud-init.sh* passes all parameters to **k8s_vm_init.sh** and an installation script is generated.



```
.
├── deploy-stack.sh
├── gen-cloud-init.sh
├── gen-ric-heat-yaml.sh
├── install
├── k8s-1node-cloud-init-k_1_16-h_2_17-d_cur.sh
├── undeploy-stack.sh
└── uninstall
```

Task 3: Execute the generated installation script *k8s-1node-cloud-init-k_1_16-h_2_17-d_cur.sh*

```
sudo ./k8s-1node-cloud-init-k_1_16-h_2_17-d_cur.sh
cd ../../../
```

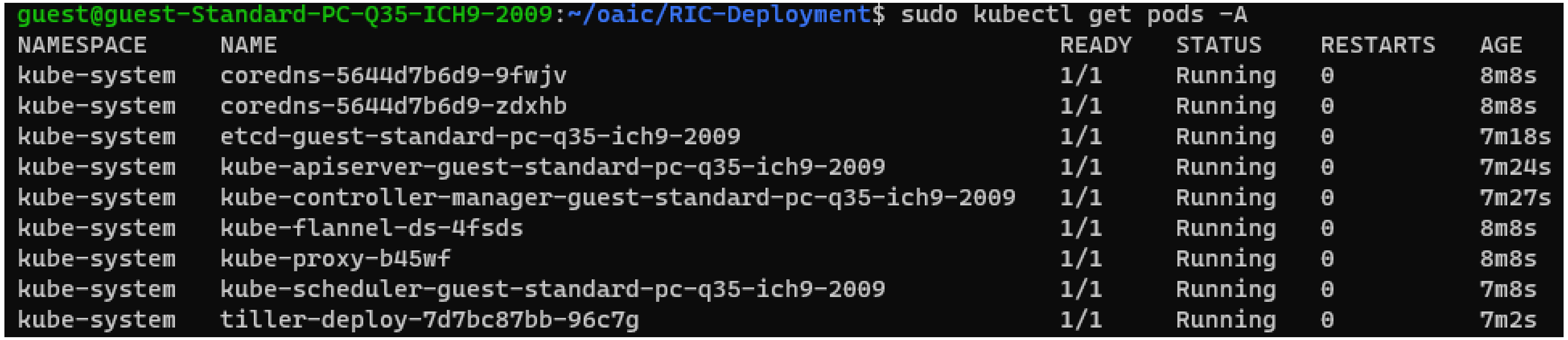


```
Preparing a master node (lowser ID) for using local FS for PV
++ kubectl get nodes
++ grep master
++ cut -f1 '-d '
++ sort
++ head -1
+ PV_NODE_NAME=guest-standard-pc-q35-ich9-2009
+ kubectl label --overwrite nodes guest-standard-pc-q35-ich9-2009 local-storage=enable
node/guest-standard-pc-q35-ich9-2009 labeled
++ hostname
+ '[' guest-standard-pc-q35-ich9-2009 == guest-Standard-PC-Q35-ICH9-2009 ']'
+ echo 'Done with master node setup'
Done with master node setup
+ [[ ! -z '' ]]
+ [[ ! -z '' ]]
+ [[ ! -z '' ]]
+ [[ 1 -gt 100 ]]
+ [[ 1 -gt 100 ]]
guest@guest-Standard-PC-Q35-ICH9-2009:~/oaic/RIC-Deployment/tools/k8s/bin$
```

Verify Docker Pods & Services are Running

- Verify all pods are deployed and running

```
sudo kubectl get pods -A
```

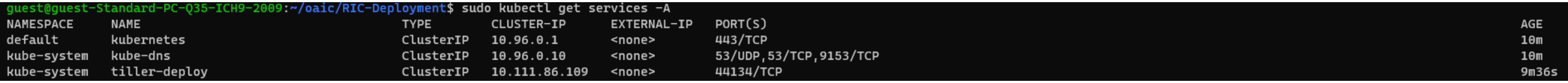


```
guest@guest-Standard-PC-Q35-ICH9-2009:~/oaic/RIC-Deployment$ sudo kubectl get pods -A
NAMESPACE     NAME                                                      READY   STATUS    RESTARTS   AGE
kube-system   coredns-5644d7b6d9-9fwjv                                  1/1     Running   0          8m8s
kube-system   coredns-5644d7b6d9-zdxhb                                  1/1     Running   0          8m8s
kube-system   etcd-guest-standard-pc-q35-ich9-2009                      1/1     Running   0          7m18s
kube-system   kube-apiserver-guest-standard-pc-q35-ich9-2009            1/1     Running   0          7m24s
kube-system   kube-controller-manager-guest-standard-pc-q35-ich9-2009   1/1     Running   0          7m27s
kube-system   kube-flannel-ds-4fsds                                     1/1     Running   0          8m8s
kube-system   kube-proxy-b45wf                                          1/1     Running   0          8m8s
kube-system   kube-scheduler-guest-standard-pc-q35-ich9-2009            1/1     Running   0          7m8s
kube-system   tiller-deploy-7d7bc87bb-96c7g                             1/1     Running   0          7m2s
```

We should have a total of 9 pods *"ready"* & *"running"*.

- Verify all services are running

```
sudo kubectl get services -A
```



```
guest@guest-Standard-PC-Q35-ICH9-2009:~/oaic/RIC-Deployment$ sudo kubectl get services -A
NAMESPACE     NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                                                                                          AGE
default       kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP                                                                                          10m
kube-system   kube-dns        ClusterIP   10.96.0.10      <none>        53/UDP,53/TCP,9153/TCP                                                                           10m
kube-system   tiller-deploy   ClusterIP   10.111.86.109   <none>        44134/TCP                                                                                        9m36s
```

We should have a total of **3** services running

Create Persistent Volume

What is Persistent Volume?

- A persistent volume (PV) is a Kubernetes resource that provides a way to store data that persists even when the pod that uses it is deleted.
  - The InfluxDB (database) uses persistent volumes to store data such as KPIs, xApp metrics etc.

- Create the *ricinfra* namespace

```
sudo kubectl create ns ricinfra
```

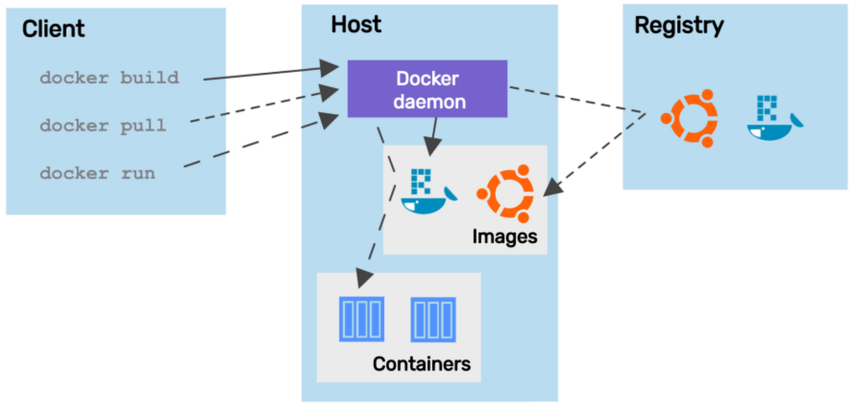
- Install the persistent storage volume

```
sudo helm install stable/nfs-server-provisioner --namespace ricinfra --name nfs-release-1
sudo kubectl patch storageclass nfs -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
sudo apt install nfs-common
```

Exercise 2: Docker Basics

Task 1 : Create a docker registry

- What is a docker registry?
  - A Docker registry is a storage and distribution system for named Docker images.
  - Here we instantiate the registry as a container which is running 24/7.



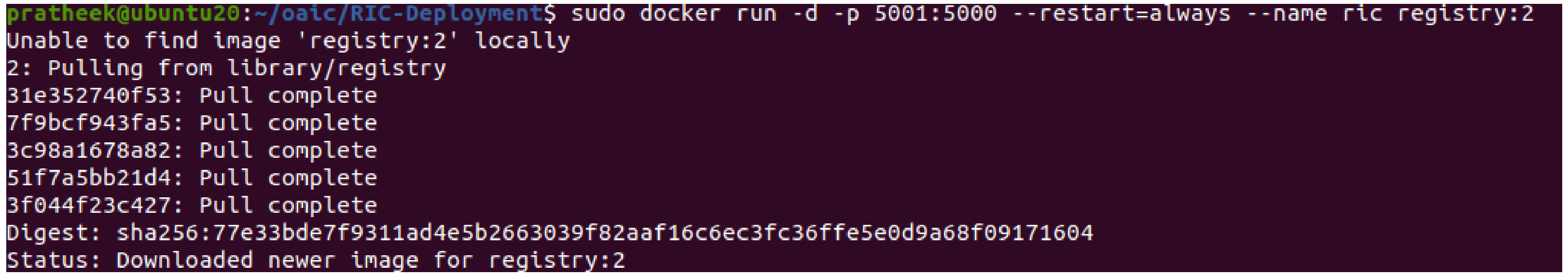
```
sudo docker run -d -p 5001:5000 --restart=always --name ric registry:2
```

Host machine & container port

Container name

Image name

version



```
pratheek@ubuntu20:~/oaic/RIC-Deployment$ sudo docker run -d -p 5001:5000 --restart=always --name ric registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
31e352740f53: Pull complete
7f9bcf943fa5: Pull complete
3c98a1678a82: Pull complete
51f7a5bb21d4: Pull complete
3f044f23c427: Pull complete
Digest: sha256:77e33bde7f9311ad4e5b2663039f82aaf16c6ec3fc36ffe5e0d9a68f09171604
Status: Downloaded newer image for registry:2
```

Docker Basics: Build and Push

Task 2 : Create a docker image with the modified E2 termination code (already provided).

- What is a docker image?

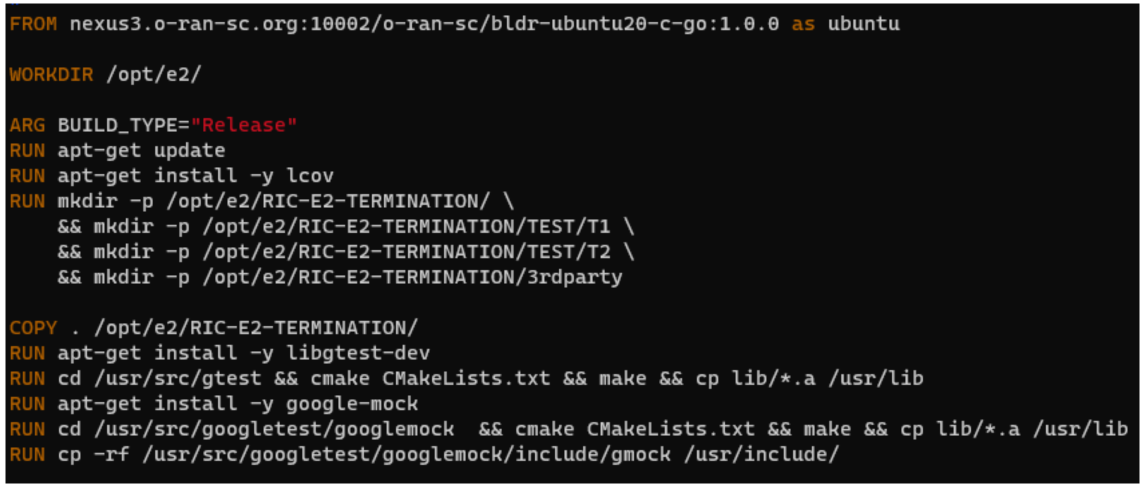
A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run

```
cd ~/oaic/ric-plt-e2/RIC-E2-TERMINATION
sudo docker build -f Dockerfile -t localhost:5001/ric-plt-e2:5.5.0 .
```

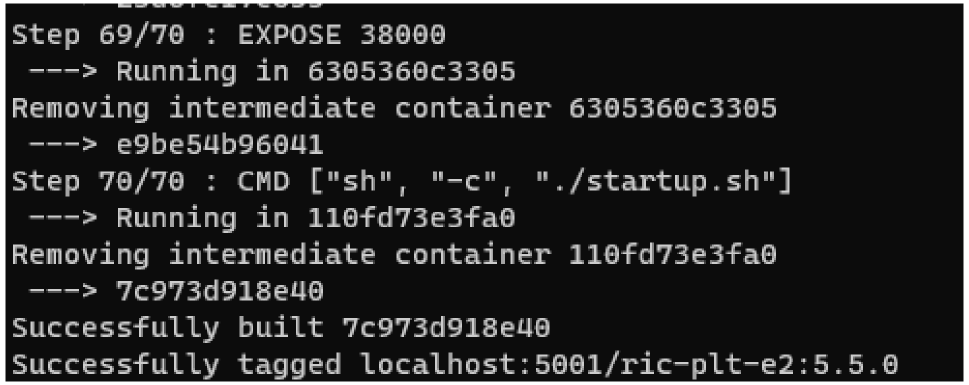
$$\underbrace{\texttt{localhost}}_{\text{IP address of our system}}:\underbrace{\texttt{5001}}_{\text{Network port number}}/\underbrace{\texttt{ric-plt-e2}}_{\text{Image name}}:\underbrace{\texttt{5.5.0}}_{\text{Version number}}$$

- Task 3 : Push the Created image to the registry

```
sudo docker push localhost:5001/ric-plt-e2:5.5.0
cd ../../
```



```
FROM nexus3.o-ran-sc.org:10002/o-ran-sc/bldr-ubuntu20-c-go:1.0.0 as ubuntu

WORKDIR /opt/e2/

ARG BUILD_TYPE="Release"
RUN apt-get update
RUN apt-get install -y lcov
RUN mkdir -p /opt/e2/RIC-E2-TERMINATION/ \
    && mkdir -p /opt/e2/RIC-E2-TERMINATION/TEST/T1 \
    && mkdir -p /opt/e2/RIC-E2-TERMINATION/TEST/T2 \
    && mkdir -p /opt/e2/RIC-E2-TERMINATION/3rdparty

COPY . /opt/e2/RIC-E2-TERMINATION/
RUN apt-get install -y libgtest-dev
RUN cd /usr/src/gtest && cmake CMakeLists.txt && make && cp lib/*.a /usr/lib
RUN apt-get install -y google-mock
RUN cd /usr/src/googletest/googlemock  && cmake CMakeLists.txt && make && cp lib/*.a /usr/lib
RUN cp -rf /usr/src/googletest/googlemock/include/gmock /usr/include/
```



```
Step 69/70 : EXPOSE 38000
 ---> Running in 6305360c3305
Removing intermediate container 6305360c3305
 ---> e9be54b96041
Step 70/70 : CMD ["sh", "-c", "./startup.sh"]
 ---> Running in 110fd73e3fa0
Removing intermediate container 110fd73e3fa0
 ---> 7c973d918e40
Successfully built 7c973d918e40
Successfully tagged localhost:5001/ric-plt-e2:5.5.0
```

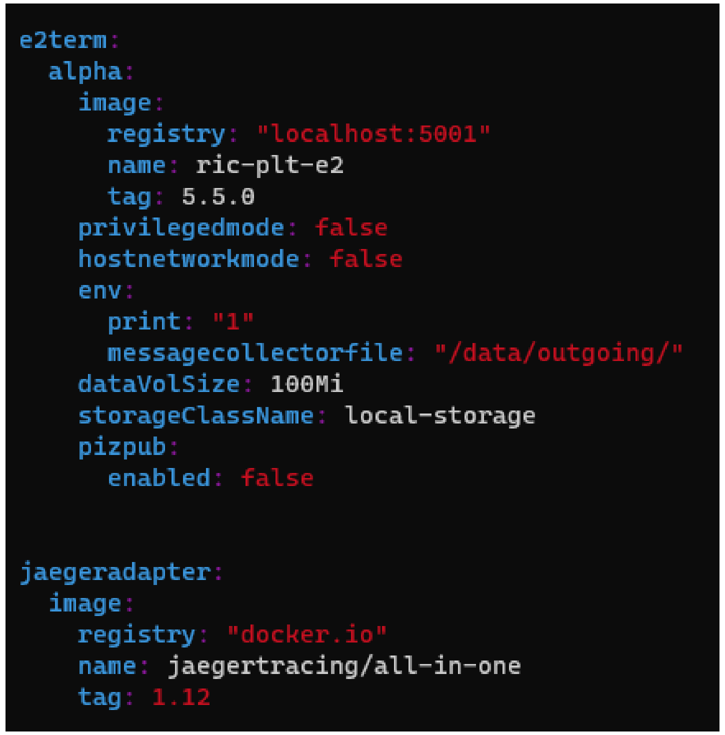
Exercise 3: Deploy the near-RT RIC

Task 1: Explore the Recipe file

- What is a Recipe file?

Recipe provides a customized specification for the components of a deployment group.

```
cd ~/oaic/RIC-Deployment/RECIPE_EXAMPLE/PLATFORM
vim example_recipe_oran_e_release_modified_e2.yaml
Close the file (See  Tip)
cd ~/oaic
```



```
e2term:
  alpha:
    image:
      registry: "localhost:5001"
      name: ric-plt-e2
      tag: 5.5.0
    privilegedmode: false
    hostnetworkmode: false
    env:
      print: "1"
      messagecollectorfile: "/data/outgoing/"
    dataVolSize: 100Mi
    storageClassName: local-storage
    pizpub:
      enabled: false


jaegeradapter:
  image:
    registry: "docker.io"
    name: jaegertracing/all-in-one
    tag: 1.12
```

Tip

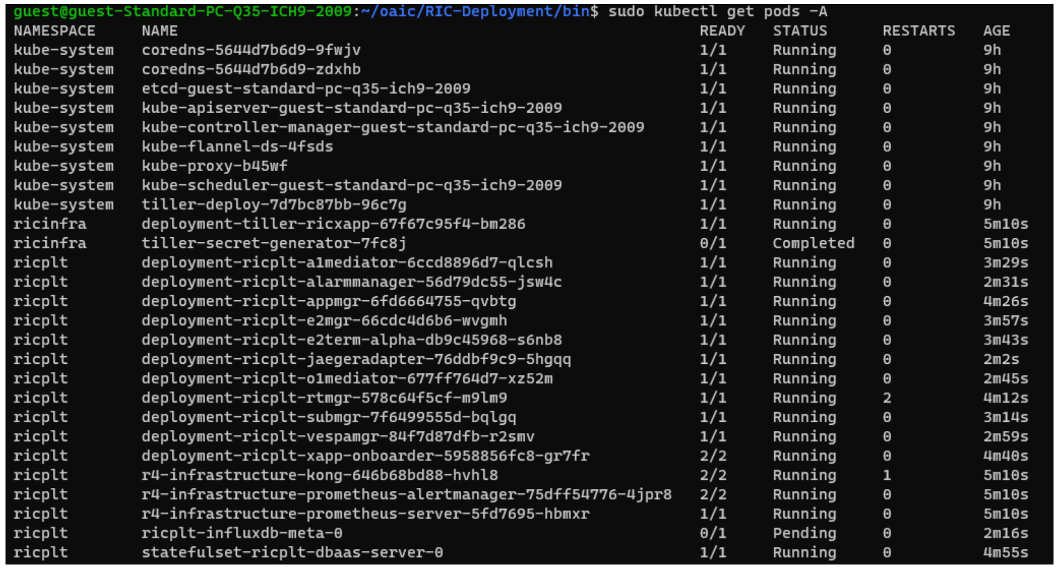
Exit a file – Press *:q*

Deploy the near-RT RIC (continued)

Task 2: Deploy the RIC Platform

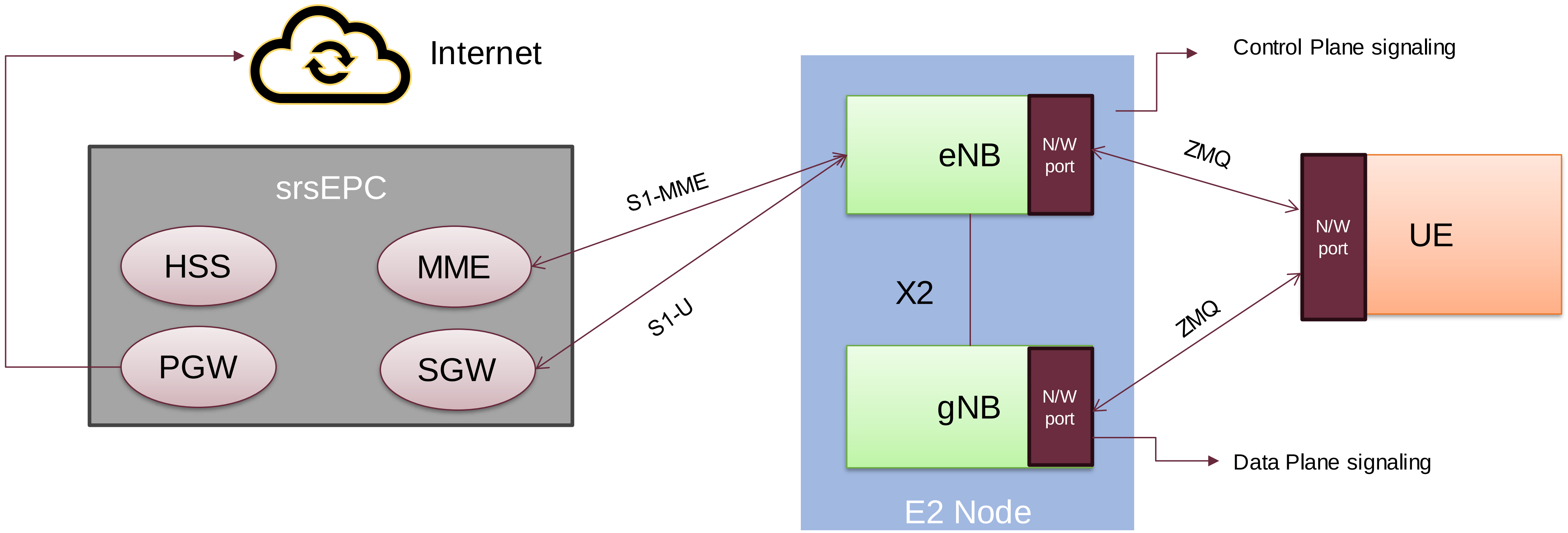
```
cd RIC-Deployment/bin

sudo ./deploy-ric-platform -f ../RECIPE_EXAMPLE/PLATFORM/example_recipe_oran_e_release_modified_e2.yaml
```

```
sudo kubectl get pods –A
sudo kubectl get services -A
```



```
guest@guest-Standard-PC-Q35-ICH9-2009:~/oaic/RIC-Deployment/bin$ sudo kubectl get pods -A
NAMESPACE     NAME                                                       READY   STATUS      RESTARTS   AGE
kube-system   coredns-5644d7b6d9-9fwjv                                   1/1     Running     0          9h
kube-system   coredns-5644d7b6d9-zdxhb                                   1/1     Running     0          9h
kube-system   etcd-guest-standard-pc-q35-ich9-2009                       1/1     Running     0          9h
kube-system   kube-apiserver-guest-standard-pc-q35-ich9-2009             1/1     Running     0          9h
kube-system   kube-controller-manager-guest-standard-pc-q35-ich9-2009    1/1     Running     0          9h
kube-system   kube-flannel-ds-4fsds                                      1/1     Running     0          9h
kube-system   kube-proxy-b45wf                                           1/1     Running     0          9h
kube-system   kube-scheduler-guest-standard-pc-q35-ich9-2009             1/1     Running     0          9h
kube-system   tiller-deploy-7d7bc87bb-96c7g                              1/1     Running     0          9h
ricinfra      deployment-tiller-ricxapp-67f67c95f4-bm286                 1/1     Running     0          5m10s
ricinfra      tiller-secret-generator-7fc8j                              0/1     Completed   0          5m10s
ricplt        deployment-ricplt-a1mediator-6ccd8896d7-qlcsh              1/1     Running     0          3m29s
ricplt        deployment-ricplt-alarmmanager-56d79dc55-jsw4c             1/1     Running     0          2m31s
ricplt        deployment-ricplt-appmgr-6fd6664755-qvbtg                  1/1     Running     0          4m26s
ricplt        deployment-ricplt-e2mgr-66cdc4d6b6-wvgmh                   1/1     Running     0          3m57s
ricplt        deployment-ricplt-e2term-alpha-db9c45968-s6nb8             1/1     Running     0          3m43s
ricplt        deployment-ricplt-jaegeradapter-76ddbf9c9-5hgqq            1/1     Running     0          2m2s
ricplt        deployment-ricplt-o1mediator-677ff764d7-xz52m              1/1     Running     0          2m45s
ricplt        deployment-ricplt-rtmgr-578c64f5cf-m9lm9                   1/1     Running     2          4m12s
ricplt        deployment-ricplt-submgr-7f6499555d-bqlgq                  1/1     Running     0          3m14s
ricplt        deployment-ricplt-vespamgr-84f7d87dfb-r2smv                1/1     Running     0          2m59s
ricplt        deployment-ricplt-xapp-onboarder-5958856fc8-gr7fr          2/2     Running     0          4m40s
ricplt        r4-infrastructure-kong-646b68bd88-hvhl8                    2/2     Running     1          5m10s
ricplt        r4-infrastructure-prometheus-alertmanager-75dff54776-4jpr8 2/2     Running     0          5m10s
ricplt        r4-infrastructure-prometheus-server-5fd7695-hbmxr          1/1     Running     0          5m10s
ricplt        ricplt-influxdb-meta-0                                     0/1     Pending     0          2m16s
ricplt        statefulset-ricplt-dbaas-server-0                          1/1     Running     0          4m55s
```

Step 3: Installing srsRAN with e2 interface



srsRAN 5G NSA Architecture – ZMQ Frontend

- Usually, eNodeB and UE are used with physical radios for over-the-air transmissions.
- Here we will use a virtual radio which uses the ZeroMQ networking library to transfer radio samples (I/Q samples) between eNB and UE.

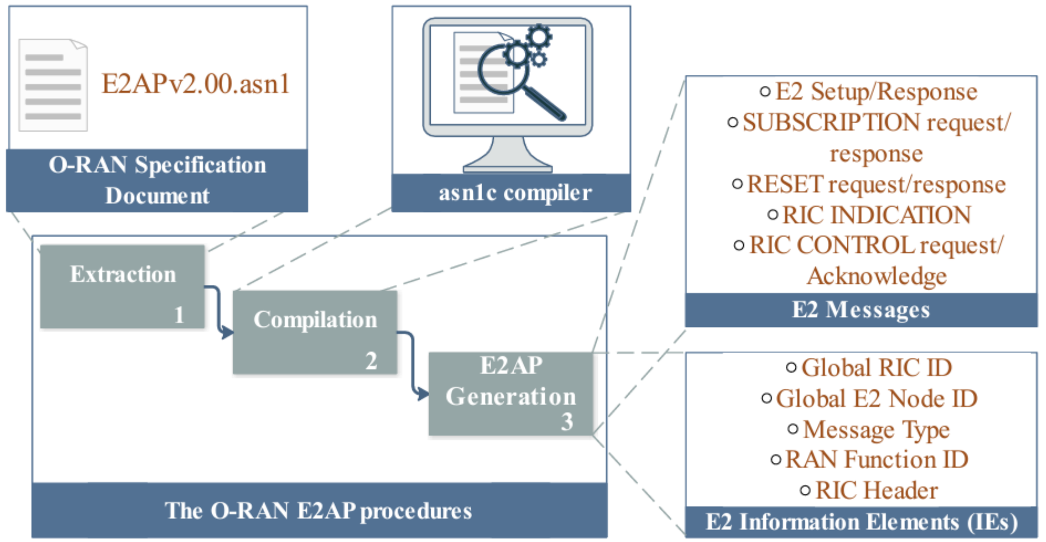
Asn1c Compiler Installation

What is ASN.1 (Abstract Syntax Notation.1)?

- ASN.1 is an interface description language (IDL) used for describing data transmitted by protocols, regardless of the underlying language implementation.
- This representation combined with standardization helps in achieving interoperability.

Why do we need the asn1 compiler?

- The compiler translates ASN.1 source specifications (developed by standardization bodies viz., 3GPP, O-RAN etc.) into C, C++, Java, Python, Go source code.
- Developers can use this code to translate the data they want to send/receive to/from the defined ASN.1 format.

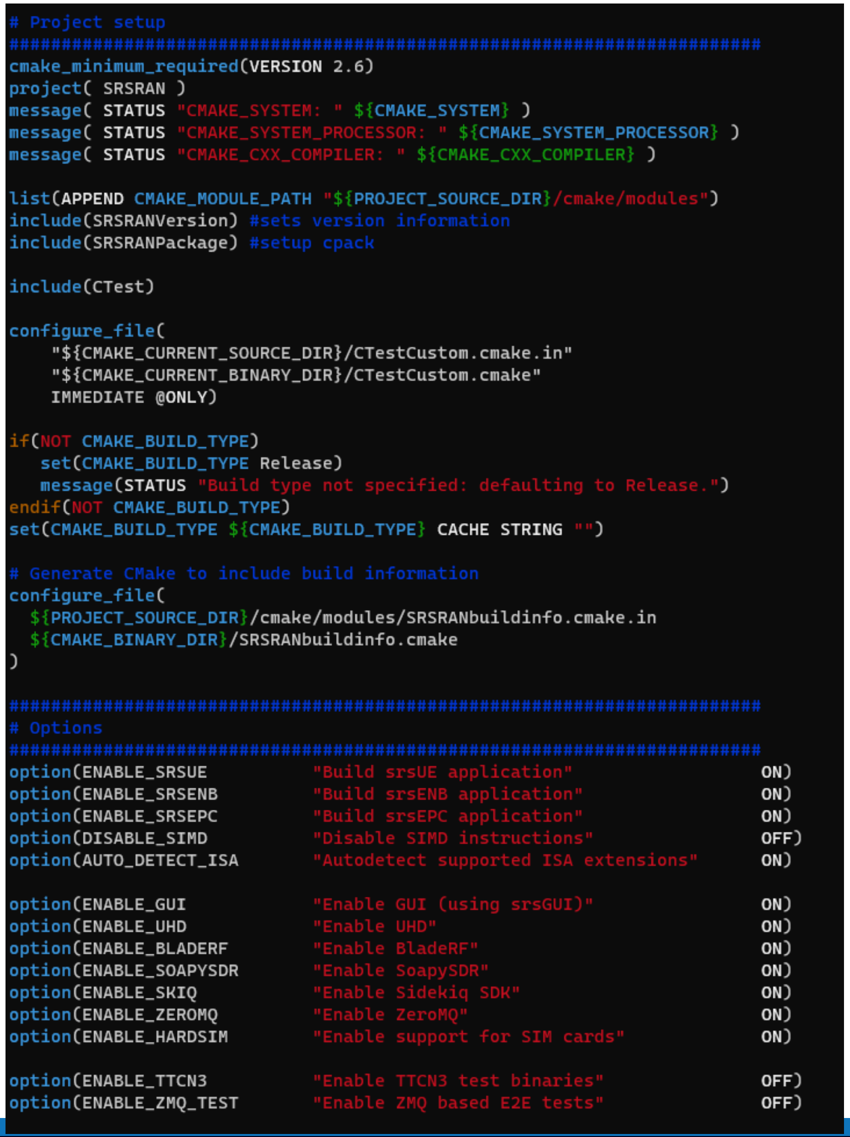


```
cd ../../asn1c
autoreconf -iv
./configure
make -j4
sudo make install
sudo ldconfig
cd ..
```

srsRAN installation

Compile and install srsRAN software stack:

```
cd srsRAN-e2
mkdir build
export SRS=`realpath .`
cd build
cmake ../ -DCMAKE_BUILD_TYPE=RelWithDebInfo \
    -DRIC_GENERATED_E2AP_BINDING_DIR=${SRS}/e2_bindings/E2AP-v01.01 \
    -DRIC_GENERATED_E2SM_KPM_BINDING_DIR=${SRS}/e2_bindings/E2SM-KPM \
    -DRIC_GENERATED_E2SM_GNB_NRT_BINDING_DIR=${SRS}/e2_bindings/E2SM-GNB-NRT
make -j`nproc`
sudo make install
sudo ldconfig
sudo srsran_install_configs.sh service
cd ../../
```



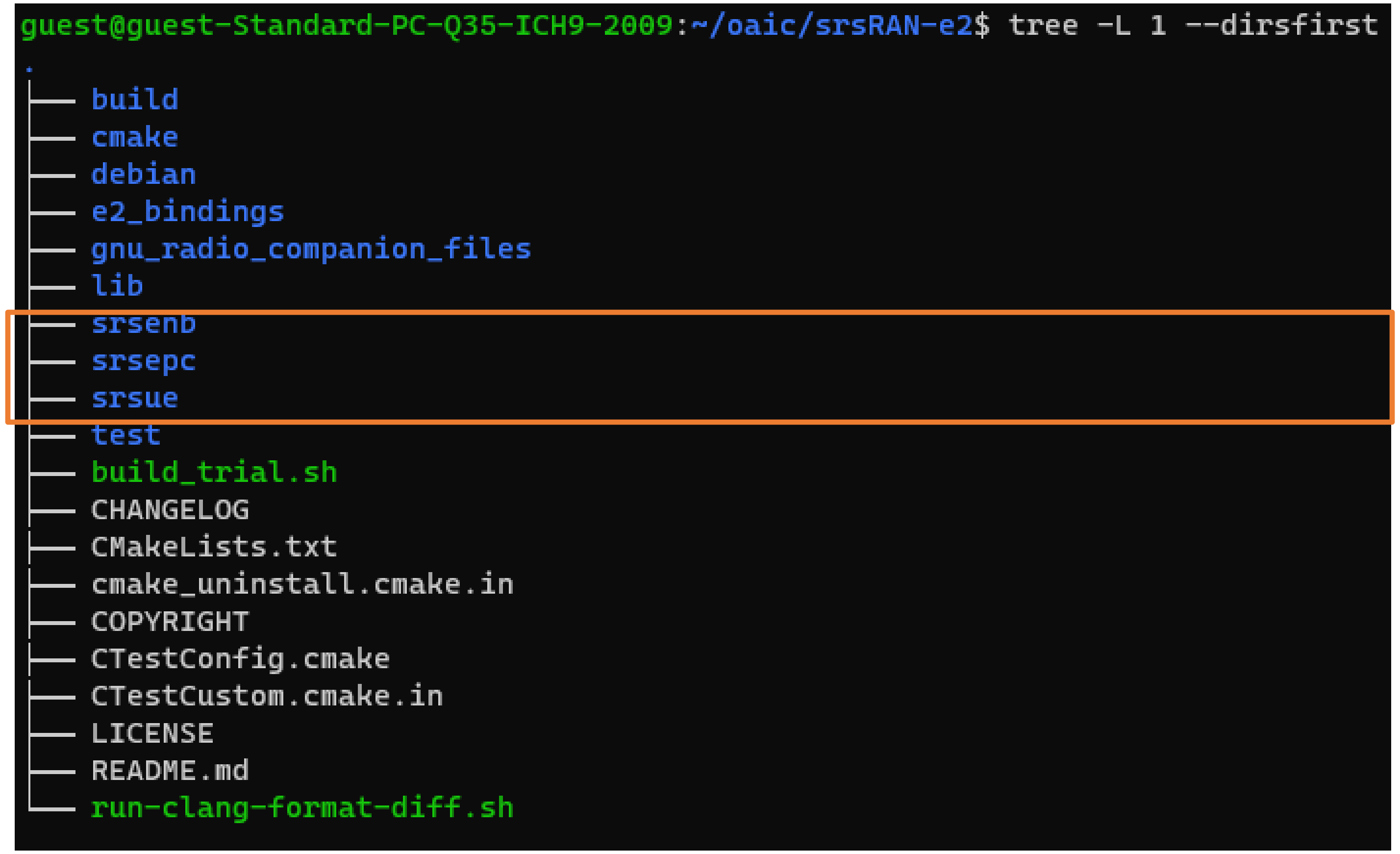
```
# Project setup
########################################################################
cmake_minimum_required(VERSION 2.6)
project( SRSRAN )
message( STATUS "CMAKE_SYSTEM: " ${CMAKE_SYSTEM} )
message( STATUS "CMAKE_SYSTEM_PROCESSOR: " ${CMAKE_SYSTEM_PROCESSOR} )
message( STATUS "CMAKE_CXX_COMPILER: " ${CMAKE_CXX_COMPILER} )

list(APPEND CMAKE_MODULE_PATH "${PROJECT_SOURCE_DIR}/cmake/modules")
include(SRSRANVersion) #sets version information
include(SRSRANPackage) #setup cpack

include(CTest)

configure_file(
    "${CMAKE_CURRENT_SOURCE_DIR}/CTestCustom.cmake.in"
    "${CMAKE_CURRENT_BINARY_DIR}/CTestCustom.cmake"
    IMMEDIATE @ONLY)

if(NOT CMAKE_BUILD_TYPE)
   set(CMAKE_BUILD_TYPE Release)
   message(STATUS "Build type not specified: defaulting to Release.")
endif(NOT CMAKE_BUILD_TYPE)
set(CMAKE_BUILD_TYPE ${CMAKE_BUILD_TYPE} CACHE STRING "")

# Generate CMake to include build information
configure_file(
  ${PROJECT_SOURCE_DIR}/cmake/modules/SRSRANbuildinfo.cmake.in
  ${CMAKE_BINARY_DIR}/SRSRANbuildinfo.cmake
)

########################################################################
# Options
########################################################################
option(ENABLE_SRSUE          "Build srsUE application"                ON)
option(ENABLE_SRSENB         "Build srsENB application"               ON)
option(ENABLE_SRSEPC         "Build srsEPC application"               ON)
option(DISABLE_SIMD          "Disable SIMD instructions"              OFF)
option(AUTO_DETECT_ISA       "Autodetect supported ISA extensions"    ON)

option(ENABLE_GUI            "Enable GUI (using srsGUI)"              ON)
option(ENABLE_UHD            "Enable UHD"                             ON)
option(ENABLE_BLADERF        "Enable BladeRF"                         ON)
option(ENABLE_SOAPYSDR       "Enable SoapySDR"                        ON)
option(ENABLE_SKIQ           "Enable Sidekiq SDK"                     ON)
option(ENABLE_ZEROMQ         "Enable ZeroMQ"                          ON)
option(ENABLE_HARDSIM        "Enable support for SIM cards"           ON)

option(ENABLE_TTCN3          "Enable TTCN3 test binaries"             OFF)
option(ENABLE_ZMQ_TEST       "Enable ZMQ based E2E tests"             OFF)
```

Step 4: Deploy 5G Network

- We will need a total of **four** terminals to trace the interaction between the near-RT RIC and the RAN.

- We will be observing the following processes

  1. The Core Network (EPC)
  2. The Base station (gNB)
  3. The User Equipment (UE)
  4. The traffic generator (e.g. ping or iPerf test)



```
guest@guest-Standard-PC-Q35-ICH9-2009:~/oaic/srsRAN-e2$ tree -L 1 --dirsfirst
.
├── build
├── cmake
├── debian
├── e2_bindings
├── gnu_radio_companion_files
├── lib
├── srsenb
├── srsepc
├── srsue
├── test
├── build_trial.sh
├── CHANGELOG
├── CMakeLists.txt
├── cmake_uninstall.cmake.in
├── COPYRIGHT
├── CTestConfig.cmake
├── CTestCustom.cmake.in
├── LICENSE
├── README.md
└── run-clang-format-diff.sh
```

Terminal 1: Deploy the EPC (Core Network)

- Here we will be using the Core Network software provided by SRS.
- The SRS base station (eNB/gNB) software is also compatible with third party Core Network solutions (Open5GS, MAGMA, etc.)
- Open a new window on the terminal. Let’s call this Terminal 2.
- Before we start the EPC, we need to create a separate network namespace for the UE since all components are running on the same machine.

1

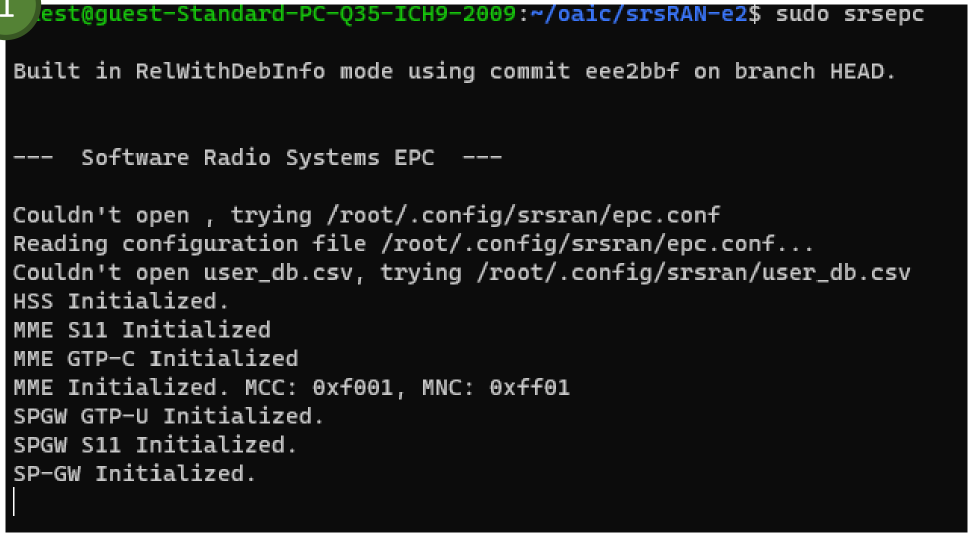
```
sudo ip netns add ue1
sudo ip netns list
```

- Start the EPC

1

```
sudo srsepc
```

1



```
est@guest-Standard-PC-Q35-ICH9-2009:~/oaic/srsRAN-e2$ sudo srsepc

Built in RelWithDebInfo mode using commit eee2bbf on branch HEAD.


---  Software Radio Systems EPC  ---

Couldn't open , trying /root/.config/srsran/epc.conf
Reading configuration file /root/.config/srsran/epc.conf...
Couldn't open user_db.csv, trying /root/.config/srsran/user_db.csv
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf001, MNC: 0xff01
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
```

Terminal 2: Deploy the en-gNB

Task 1: Get the IP address of the E2 Termination pod

- To connect the en-gNB to the near-RT RIC we should specify the IP address of the E2 Termination pod while instantiating the gNB.

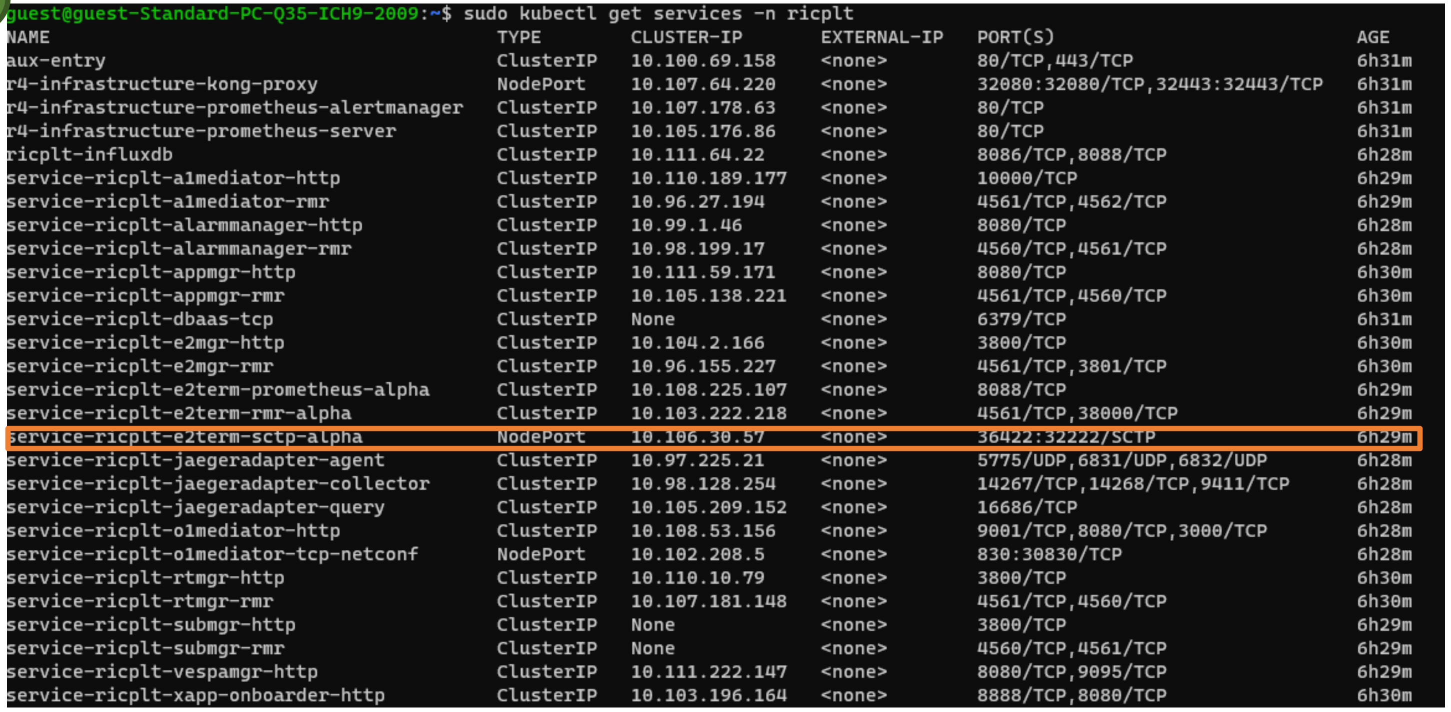
2

```
sudo kubectl get svc -n ricplt
```

Warning: This IP address will be different for each one of you! DO NOT COPY from the picture.

To automatically get the IP address,

2



```
guest@guest-Standard-PC-Q35-ICH9-2009:~$ sudo kubectl get services -n ricplt
NAME                                          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                              AGE
aux-entry                                     ClusterIP   10.100.69.158    <none>        80/TCP,443/TCP                       6h31m
r4-infrastructure-kong-proxy                  NodePort    10.107.64.220    <none>        32080:32080/TCP,32443:32443/TCP      6h31m
r4-infrastructure-prometheus-alertmanager     ClusterIP   10.107.178.63    <none>        80/TCP                               6h31m
r4-infrastructure-prometheus-server           ClusterIP   10.105.176.86    <none>        80/TCP                               6h31m
ricplt-influxdb                               ClusterIP   10.111.64.22     <none>        8086/TCP,8088/TCP                    6h28m
service-ricplt-a1mediator-http                ClusterIP   10.110.189.177   <none>        10000/TCP                            6h29m
service-ricplt-a1mediator-rmr                 ClusterIP   10.96.27.194     <none>        4561/TCP,4562/TCP                    6h29m
service-ricplt-alarmmanager-http              ClusterIP   10.99.1.46       <none>        8080/TCP                             6h28m
service-ricplt-alarmmanager-rmr               ClusterIP   10.98.199.17     <none>        4560/TCP,4561/TCP                    6h28m
service-ricplt-appmgr-http                    ClusterIP   10.111.59.171    <none>        8080/TCP                             6h30m
service-ricplt-appmgr-rmr                     ClusterIP   10.105.138.221   <none>        4561/TCP,4560/TCP                    6h30m
service-ricplt-dbaas-tcp                      ClusterIP   None             <none>        6379/TCP                             6h31m
service-ricplt-e2mgr-http                     ClusterIP   10.104.2.166     <none>        3800/TCP                             6h30m
service-ricplt-e2mgr-rmr                      ClusterIP   10.96.155.227    <none>        4561/TCP,3801/TCP                    6h30m
service-ricplt-e2term-prometheus-alpha        ClusterIP   10.108.225.107   <none>        8088/TCP                             6h29m
service-ricplt-e2term-rmr-alpha               ClusterIP   10.103.222.218   <none>        4561/TCP,38000/TCP                   6h29m
service-ricplt-e2term-sctp-alpha              NodePort    10.106.30.57     <none>        36422:32222/SCTP                     6h29m
service-ricplt-jaegeradapter-agent            ClusterIP   10.97.225.21     <none>        5775/UDP,6831/UDP,6832/UDP           6h28m
service-ricplt-jaegeradapter-collector        ClusterIP   10.98.128.254    <none>        14267/TCP,14268/TCP,9411/TCP         6h28m
service-ricplt-jaegeradapter-query            ClusterIP   10.105.209.152   <none>        16686/TCP                            6h28m
service-ricplt-o1mediator-http                ClusterIP   10.108.53.156    <none>        9001/TCP,8080/TCP,3000/TCP           6h28m
service-ricplt-o1mediator-tcp-netconf         NodePort    10.102.208.5     <none>        830:30830/TCP                        6h28m
service-ricplt-rtmgr-http                     ClusterIP   10.110.10.79     <none>        3800/TCP                             6h30m
service-ricplt-rtmgr-rmr                      ClusterIP   10.107.181.148   <none>        4561/TCP,4560/TCP                    6h30m
service-ricplt-submgr-http                    ClusterIP   None             <none>        3800/TCP                             6h29m
service-ricplt-submgr-rmr                     ClusterIP   None             <none>        4560/TCP,4561/TCP                    6h29m
service-ricplt-vespamgr-http                  ClusterIP   10.111.222.147   <none>        8080/TCP,9095/TCP                    6h29m
service-ricplt-xapp-onboarder-http            ClusterIP   10.103.196.164   <none>        8888/TCP,8080/TCP                    6h30m
```

2

```
export E2TERM_IP=`sudo kubectl get svc -n ricplt --field-selector metadata.name=service-ricplt-e2term-sctp-alpha -o jsonpath='{.items[0].spec.clusterIP}'`
echo $E2TERM_IP
```

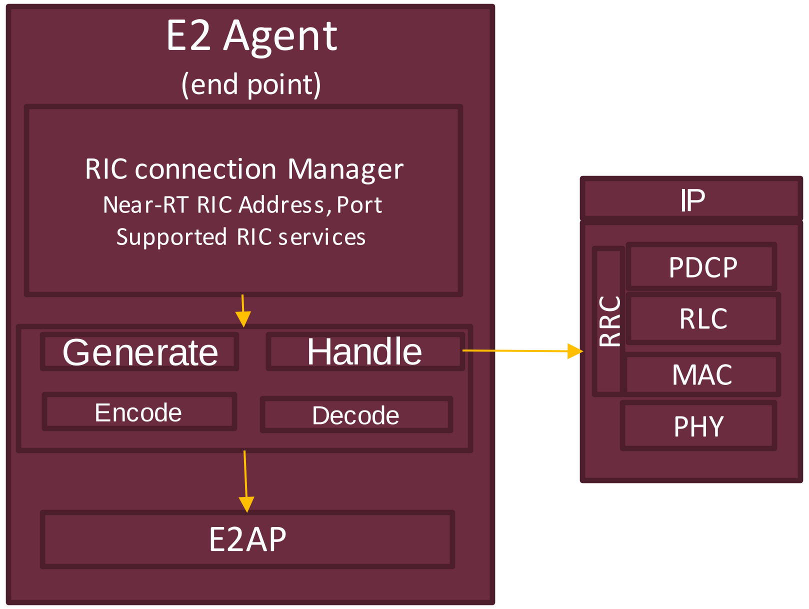
Terminal 2: Deploy the en-gNB (continued)

Task 2: Bring up the en-gNB

- Get the host Machine IP address

2

```
export E2NODE_IP=`hostname  -I | cut -f1 -d' '`
export E2NODE_PORT=5006
```



- Instantiate the en-gNB

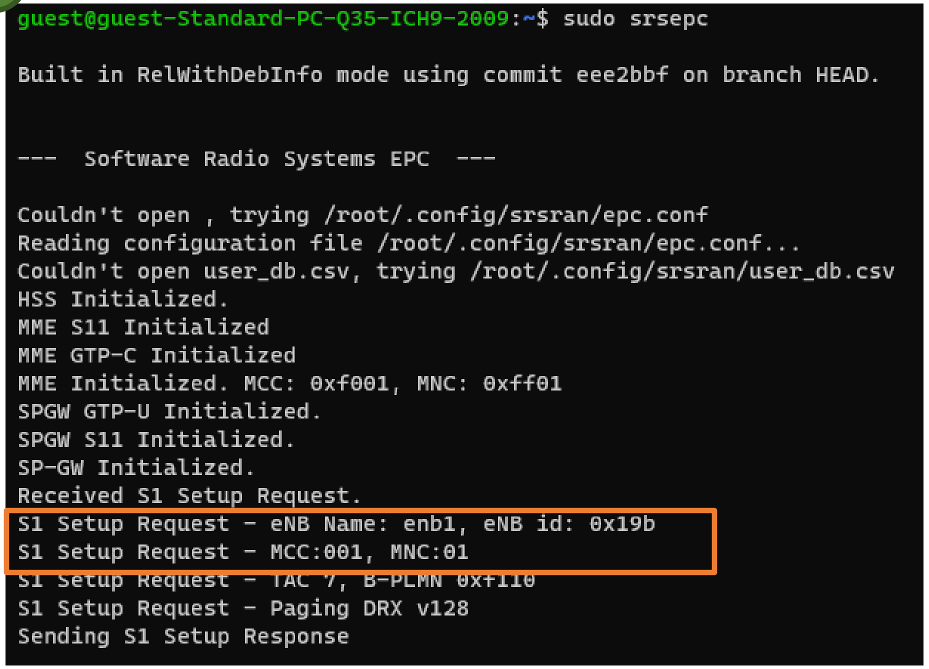
2

```
sudo srsenb --enb.n_prb=50 --enb.name=enb1 --enb.enb_id=0x19B --rf.device_name=zmq \
 --rf.device_args="fail_on_disconnect=true,tx_port0=tcp://*:2000,rx_port0=tcp://localhost:2001,tx_port1=tcp://*:2100,rx_port1=tcp://localhost:2101,id=enb,base_srate=23.04e6" \
 --ric.agent.remote_ipv4_addr=${E2TERM_IP} --log.all_level=warn --ric.agent.log_level=debug --log.filename=stdout --ric.agent.local_ipv4_addr=${E2NODE_IP} \
 --ric.agent.local_port=${E2NODE_PORT}
```

- Wait for about 30 seconds
- Observe the output on all the first two terminals

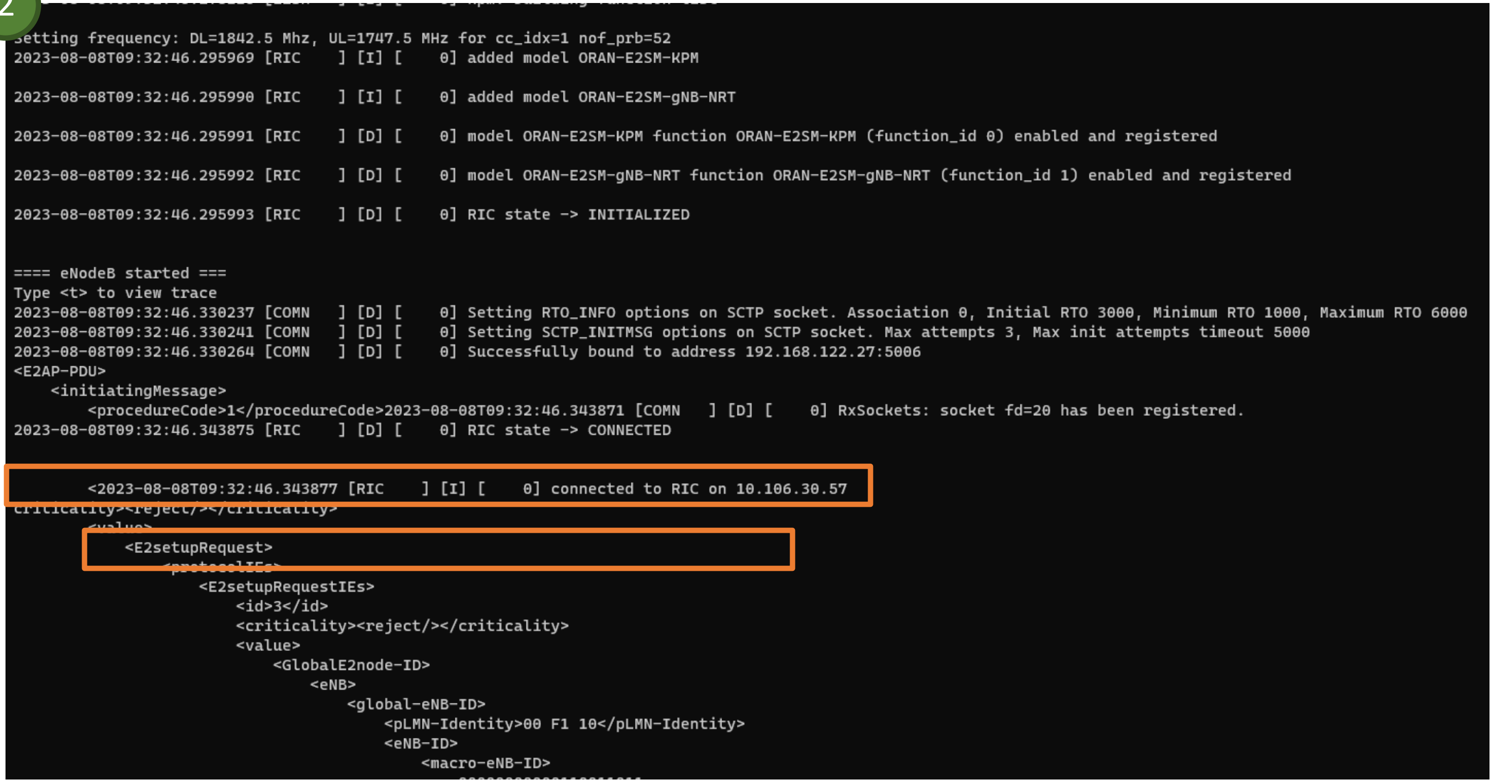
EPC and en-gNB Logs

1



```
guest@guest-Standard-PC-Q35-ICH9-2009:~$ sudo srsepc

Built in RelWithDebInfo mode using commit eee2bbf on branch HEAD.


---  Software Radio Systems EPC  ---

Couldn't open , trying /root/.config/srsran/epc.conf
Reading configuration file /root/.config/srsran/epc.conf...
Couldn't open user_db.csv, trying /root/.config/srsran/user_db.csv
HSS Initialized.
MME S11 Initialized
MME GTP-C Initialized
MME Initialized. MCC: 0xf001, MNC: 0xff01
SPGW GTP-U Initialized.
SPGW S11 Initialized.
SP-GW Initialized.
Received S1 Setup Request.
S1 Setup Request - eNB Name: enb1, eNB id: 0x19b
S1 Setup Request - MCC:001, MNC:01
S1 Setup Request - TAC 7, B-PLMN 0xf110
S1 Setup Request - Paging DRX v128
Sending S1 Setup Response
```

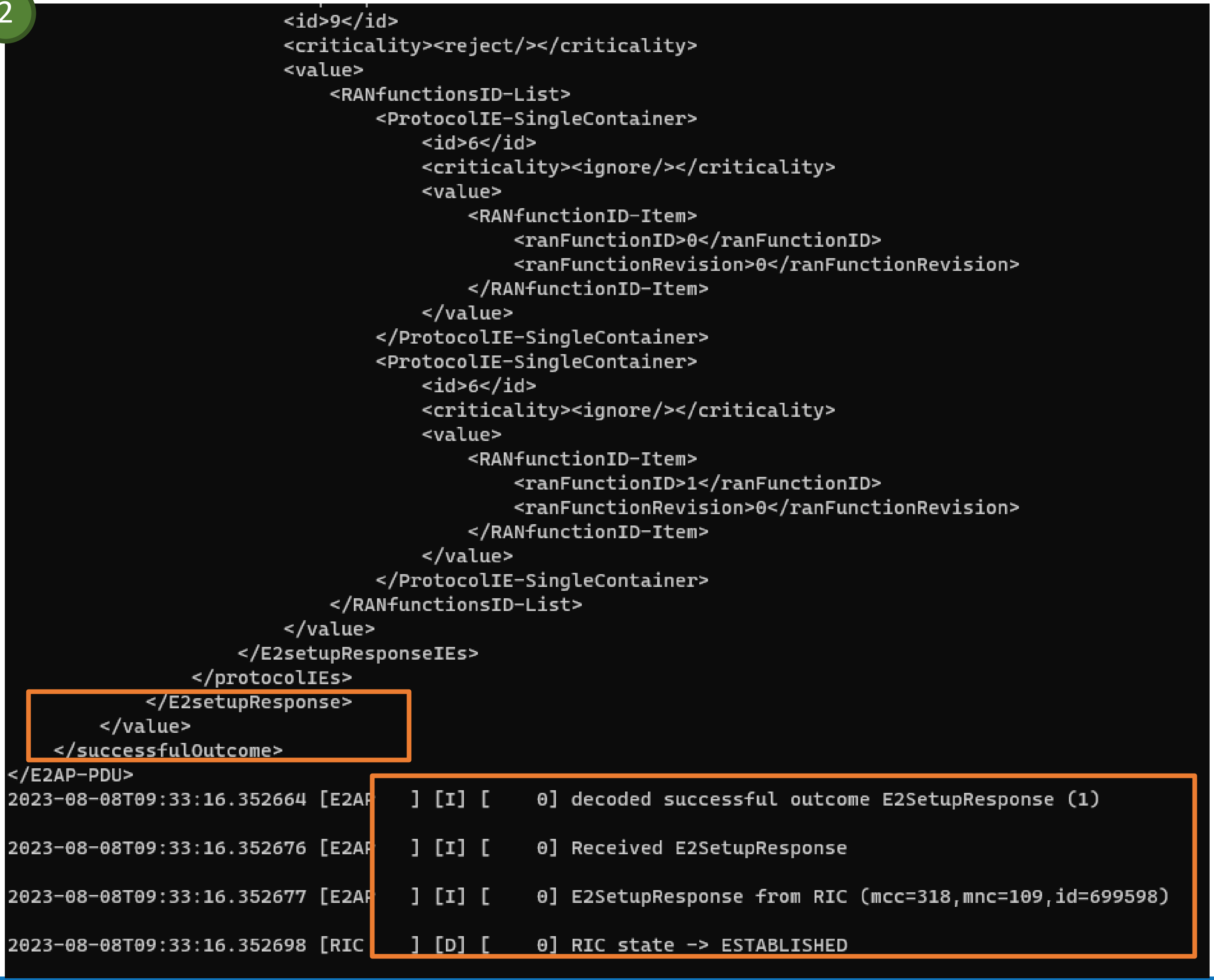
2



```
Setting frequency: DL=1842.5 Mhz, UL=1747.5 MHz for cc_idx=1 nof_prb=52
2023-08-08T09:32:46.295969 [RIC    ] [I] [    0] added model ORAN-E2SM-KPM

2023-08-08T09:32:46.295990 [RIC    ] [I] [    0] added model ORAN-E2SM-gNB-NRT

2023-08-08T09:32:46.295991 [RIC    ] [D] [    0] model ORAN-E2SM-KPM function ORAN-E2SM-KPM (function_id 0) enabled and registered

2023-08-08T09:32:46.295992 [RIC    ] [D] [    0] model ORAN-E2SM-gNB-NRT function ORAN-E2SM-gNB-NRT (function_id 1) enabled and registered

2023-08-08T09:32:46.295993 [RIC    ] [D] [    0] RIC state -> INITIALIZED


==== eNodeB started ===
Type <t> to view trace
2023-08-08T09:32:46.330237 [COMN   ] [D] [    0] Setting RTO_INFO options on SCTP socket. Association 0, Initial RTO 3000, Minimum RTO 1000, Maximum RTO 6000
2023-08-08T09:32:46.330241 [COMN   ] [D] [    0] Setting SCTP_INITMSG options on SCTP socket. Max attempts 3, Max init attempts timeout 5000
2023-08-08T09:32:46.330264 [COMN   ] [D] [    0] Successfully bound to address 192.168.122.27:5006
<E2AP-PDU>
    <initiatingMessage>
        <procedureCode>1</procedureCode>2023-08-08T09:32:46.343871 [COMN   ] [D] [    0] RxSockets: socket fd=20 has been registered.
2023-08-08T09:32:46.343875 [RIC    ] [D] [    0] RIC state -> CONNECTED


        <2023-08-08T09:32:46.343877 [RIC    ] [I] [    0] connected to RIC on 10.106.30.57
criticality><reject/></criticality>
        <value>
            <E2setupRequest>
                <protocolIEs>
                    <E2setupRequestIEs>
                        <id>3</id>
                        <criticality><reject/></criticality>
                        <value>
                            <GlobalE2node-ID>
                                <eNB>
                                    <global-eNB-ID>
                                        <pLMN-Identity>00 F1 10</pLMN-Identity>
                                        <eNB-ID>
                                            <macro-eNB-ID>
```

en-gNB Logs

2



```
                        <id>9</id>
                        <criticality><reject/></criticality>
                        <value>
                            <RANfunctionsID-List>
                                <ProtocolIE-SingleContainer>
                                    <id>6</id>
                                    <criticality><ignore/></criticality>
                                    <value>
                                        <RANfunctionID-Item>
                                            <ranFunctionID>0</ranFunctionID>
                                            <ranFunctionRevision>0</ranFunctionRevision>
                                        </RANfunctionID-Item>
                                    </value>
                                </ProtocolIE-SingleContainer>
                                <ProtocolIE-SingleContainer>
                                    <id>6</id>
                                    <criticality><ignore/></criticality>
                                    <value>
                                        <RANfunctionID-Item>
                                            <ranFunctionID>1</ranFunctionID>
                                            <ranFunctionRevision>0</ranFunctionRevision>
                                        </RANfunctionID-Item>
                                    </value>
                                </ProtocolIE-SingleContainer>
                            </RANfunctionsID-List>
                        </value>
                    </E2setupResponseIEs>
                </protocolIEs>
            </E2setupResponse>
        </value>
    </successfulOutcome>
</E2AP-PDU>
2023-08-08T09:33:16.352664 [E2AP   ] [I] [     0] decoded successful outcome E2SetupResponse (1)

2023-08-08T09:33:16.352676 [E2AP   ] [I] [     0] Received E2SetupResponse

2023-08-08T09:33:16.352677 [E2AP   ] [I] [     0] E2SetupResponse from RIC (mcc=318,mnc=109,id=699598)

2023-08-08T09:33:16.352698 [RIC    ] [D] [     0] RIC state -> ESTABLISHED
```

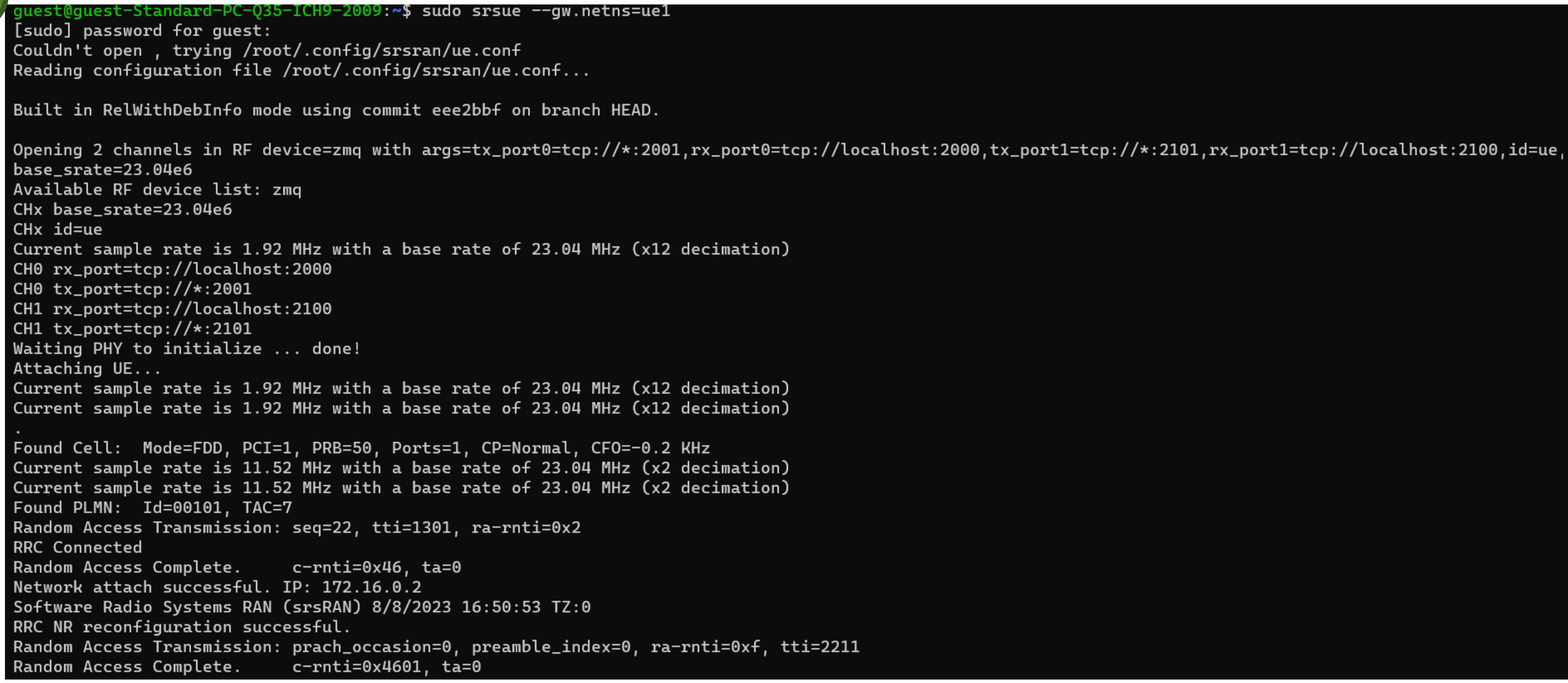
Terminal 3: Start the UE

Open a third terminal and start srsUE

3

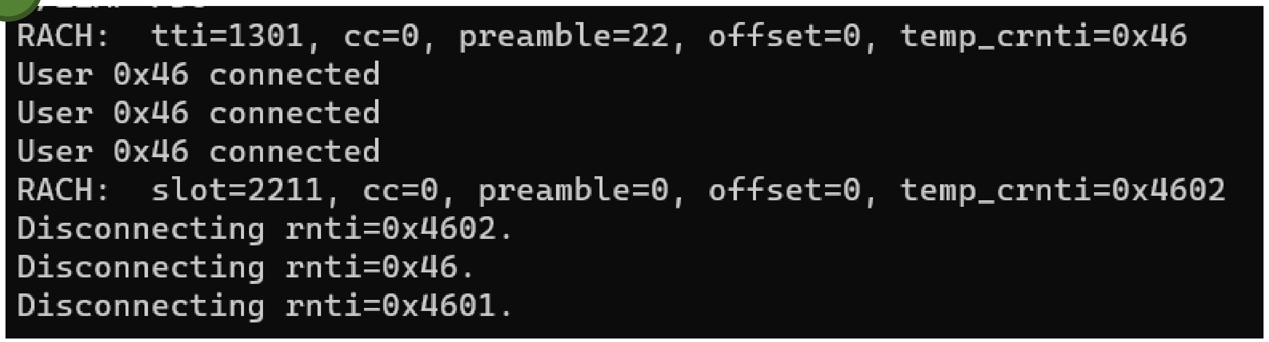
```
sudo srsue --gw.netns=ue1
```

3



```
guest@guest-Standard-PC-Q35-ICH9-2009:~$ sudo srsue --gw.netns=ue1
[sudo] password for guest:
Couldn't open , trying /root/.config/srsran/ue.conf
Reading configuration file /root/.config/srsran/ue.conf...

Built in RelWithDebInfo mode using commit eee2bbf on branch HEAD.

Opening 2 channels in RF device=zmq with args=tx_port0=tcp://*:2001,rx_port0=tcp://localhost:2000,tx_port1=tcp://*:2101,rx_port1=tcp://localhost:2100,id=ue,
base_srate=23.04e6
Available RF device list: zmq
CHx base_srate=23.04e6
CHx id=ue
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://localhost:2000
CH0 tx_port=tcp://*:2001
CH1 rx_port=tcp://localhost:2100
CH1 tx_port=tcp://*:2101
Waiting PHY to initialize ... done!
Attaching UE...
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
.
Found Cell:  Mode=FDD, PCI=1, PRB=50, Ports=1, CP=Normal, CFO=-0.2 KHz
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Found PLMN:  Id=00101, TAC=7
Random Access Transmission: seq=22, tti=1301, ra-rnti=0x2
RRC Connected
Random Access Complete.     c-rnti=0x46, ta=0
Network attach successful. IP: 172.16.0.2
Software Radio Systems RAN (srsRAN) 8/8/2023 16:50:53 TZ:0
RRC NR reconfiguration successful.
Random Access Transmission: prach_occasion=0, preamble_index=0, ra-rnti=0xf, tti=2211
Random Access Complete.     c-rnti=0x4601, ta=0
```

2



```
RACH:  tti=1301, cc=0, preamble=22, offset=0, temp_crnti=0x46
User 0x46 connected
User 0x46 connected
User 0x46 connected
RACH:  slot=2211, cc=0, preamble=0, offset=0, temp_crnti=0x4602
Disconnecting rnti=0x4602.
Disconnecting rnti=0x46.
Disconnecting rnti=0x4601.
```

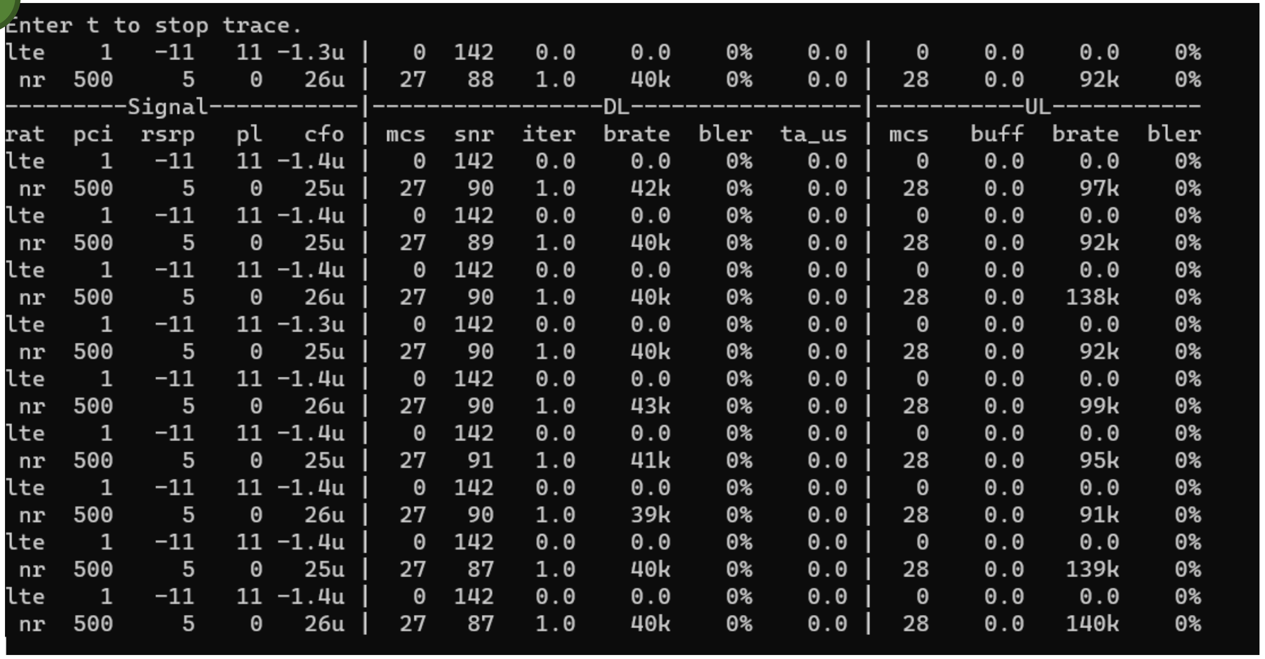
Terminal 4: Run traffic

Open a fourth terminal and check for connectivity

4

```
sudo ip netns exec ue1 ping 172.16.0.1 -c50
```

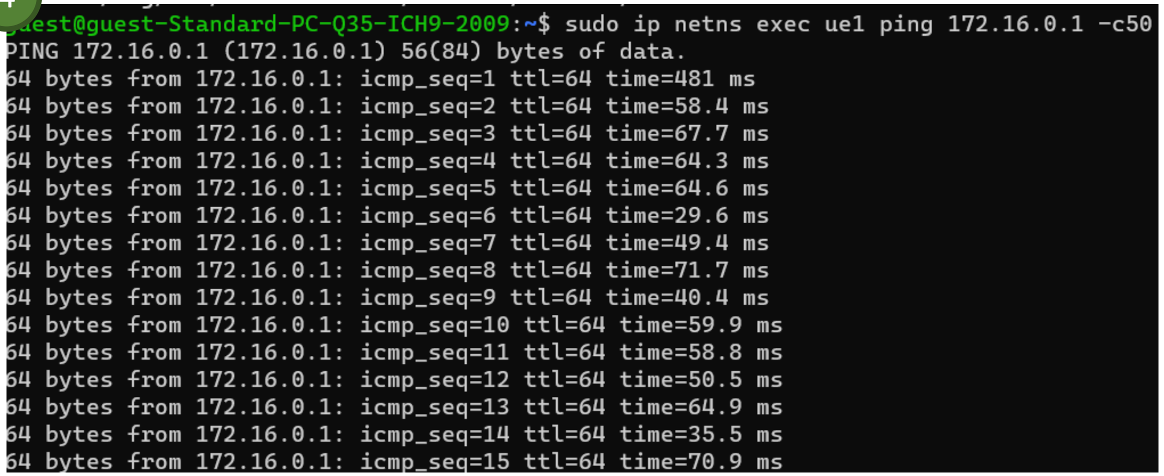
3



```
Enter t to stop trace.
lte     1   -11   11 -1.3u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   26u |  27   88   1.0    40k    0%    0.0 |  28   0.0    92k    0%
---------Signal-----------|-----------------DL-----------------|-----------UL-----------
rat   pci  rsrp   pl   cfo |  mcs  snr  iter  brate  bler  ta_us |  mcs  buff  brate  bler
lte     1   -11   11 -1.4u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   25u |  27   90   1.0    42k    0%    0.0 |  28   0.0    97k    0%
lte     1   -11   11 -1.4u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   25u |  27   89   1.0    40k    0%    0.0 |  28   0.0    92k    0%
lte     1   -11   11 -1.4u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   26u |  27   90   1.0    40k    0%    0.0 |  28   0.0   138k    0%
lte     1   -11   11 -1.3u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   25u |  27   90   1.0    40k    0%    0.0 |  28   0.0    92k    0%
lte     1   -11   11 -1.4u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   26u |  27   90   1.0    43k    0%    0.0 |  28   0.0    99k    0%
lte     1   -11   11 -1.4u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   25u |  27   91   1.0    41k    0%    0.0 |  28   0.0    95k    0%
lte     1   -11   11 -1.4u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   26u |  27   90   1.0    39k    0%    0.0 |  28   0.0    91k    0%
lte     1   -11   11 -1.4u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   25u |  27   87   1.0    40k    0%    0.0 |  28   0.0   139k    0%
lte     1   -11   11 -1.4u |   0  142   0.0    0.0    0%    0.0 |   0   0.0    0.0    0%
 nr   500     5    0   26u |  27   87   1.0    40k    0%    0.0 |  28   0.0   140k    0%
```

UE Console trace – Press "t" on UE Terminal (Terminal 3)

4



```
uest@guest-Standard-PC-Q35-ICH9-2009:~$ sudo ip netns exec ue1 ping 172.16.0.1 -c50
PING 172.16.0.1 (172.16.0.1) 56(84) bytes of data.
64 bytes from 172.16.0.1: icmp_seq=1 ttl=64 time=481 ms
64 bytes from 172.16.0.1: icmp_seq=2 ttl=64 time=58.4 ms
64 bytes from 172.16.0.1: icmp_seq=3 ttl=64 time=67.7 ms
64 bytes from 172.16.0.1: icmp_seq=4 ttl=64 time=64.3 ms
64 bytes from 172.16.0.1: icmp_seq=5 ttl=64 time=64.6 ms
64 bytes from 172.16.0.1: icmp_seq=6 ttl=64 time=29.6 ms
64 bytes from 172.16.0.1: icmp_seq=7 ttl=64 time=49.4 ms
64 bytes from 172.16.0.1: icmp_seq=8 ttl=64 time=71.7 ms
64 bytes from 172.16.0.1: icmp_seq=9 ttl=64 time=40.4 ms
64 bytes from 172.16.0.1: icmp_seq=10 ttl=64 time=59.9 ms
64 bytes from 172.16.0.1: icmp_seq=11 ttl=64 time=58.8 ms
64 bytes from 172.16.0.1: icmp_seq=12 ttl=64 time=50.5 ms
64 bytes from 172.16.0.1: icmp_seq=13 ttl=64 time=64.9 ms
64 bytes from 172.16.0.1: icmp_seq=14 ttl=64 time=35.5 ms
64 bytes from 172.16.0.1: icmp_seq=15 ttl=64 time=70.9 ms
```

QUESTIONS?

Open AI Cellular

National Science Foundation

THANK YOU

Backup

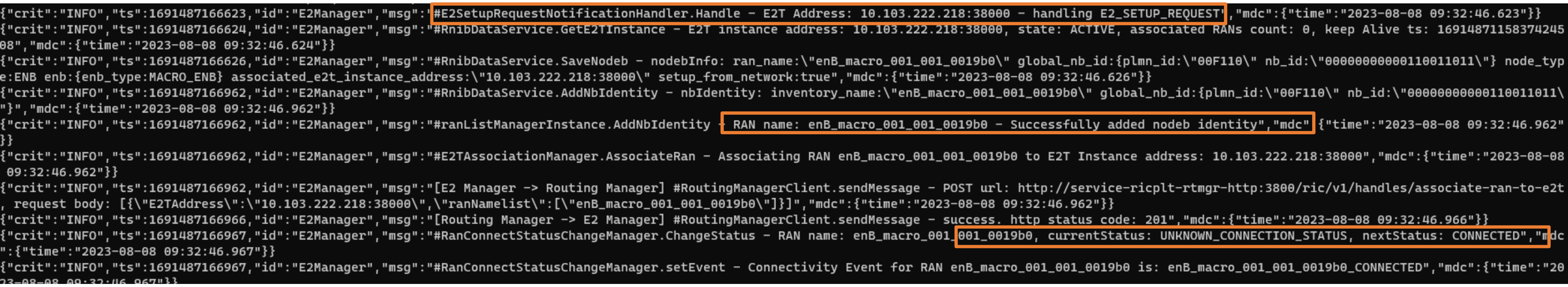
Notes + TODO

- Modifications to base VM:
  - Default screen resolution
  - Include terminal as shortcut
  - Remove extra stuff as favorites
  - Change background image
  - Shortcut to oaic installation on desktop
  - `sudo apt-get install net-tools vim openssh-server htop`
  - Enable ssh
    - Ssh timeout
- When trying to run apt-get install, getting error "could not get lock /var/lib/dpkg/lock-frontend": Reboot VM?
- Password-less sudo on VMs

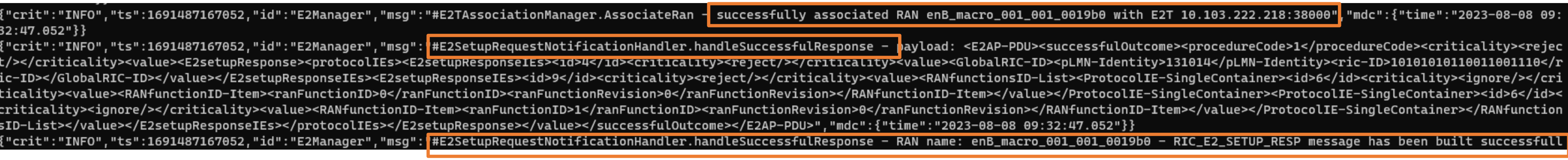
Troubleshooting

- Error “Could not get lock /var/lib/dpkg/lock-frontend”
  - “sudo killall apt apt-get”
- E2 Termination pod is not ready
  - sudo kubectl -n ricplt rollout restart deployment deployment-ricplt-e2term-alpha
- Find if a process is running
  - ps ax | grep <pname>
- Error “could not find a ready tiller pod”
  - Wait and try again (?)

E2 Manager Logs

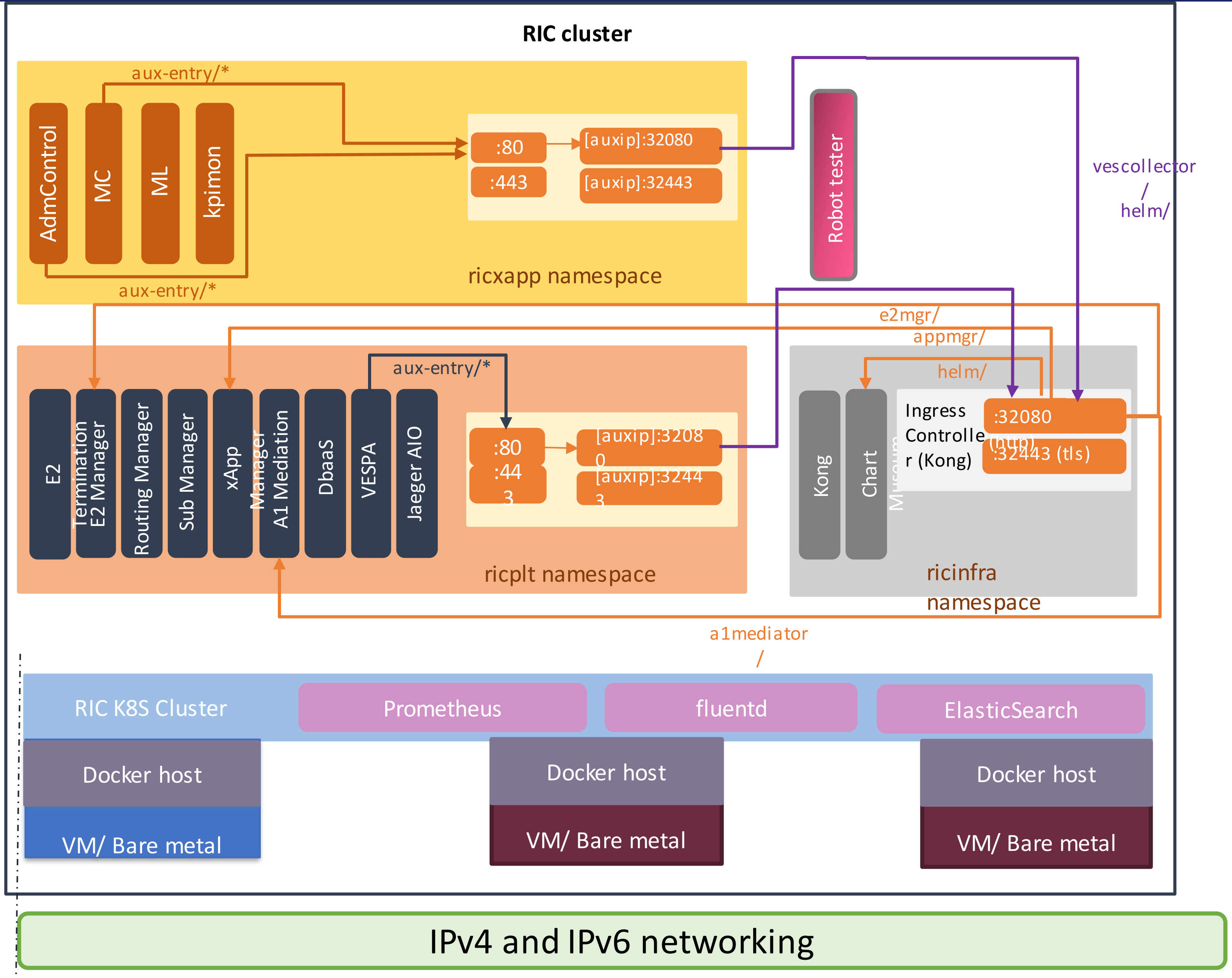


```
{"crit":"INFO","ts":1691487166623,"id":"E2Manager","msg":"#E2SetupRequestNotificationHandler.Handle - E2T Address: 10.103.222.218:38000 - handling E2_SETUP_REQUEST","mdc":{"time":"2023-08-08 09:32:46.623"}}
{"crit":"INFO","ts":1691487166624,"id":"E2Manager","msg":"#RnibDataService.GetE2TInstance - E2T instance address: 10.103.222.218:38000, state: ACTIVE, associated RANs count: 0, keep Alive ts: 16914871158374245
08","mdc":{"time":"2023-08-08 09:32:46.624"}}
{"crit":"INFO","ts":1691487166626,"id":"E2Manager","msg":"#RnibDataService.SaveNodeb - nodebInfo: ran_name:\"enB_macro_001_001_0019b0\" global_nb_id:{plmn_id:\"00F110\" nb_id:\"00000000000110011011\"} node_typ
e:ENB enb:{enb_type:MACRO_ENB} associated_e2t_instance_address:\"10.103.222.218:38000\" setup_from_network:true","mdc":{"time":"2023-08-08 09:32:46.626"}}
{"crit":"INFO","ts":1691487166962,"id":"E2Manager","msg":"#RnibDataService.AddNbIdentity - nbIdentity: inventory_name:\"enB_macro_001_001_0019b0\" global_nb_id:{plmn_id:\"00F110\" nb_id:\"00000000000110011011\
"}","mdc":{"time":"2023-08-08 09:32:46.962"}}
{"crit":"INFO","ts":1691487166962,"id":"E2Manager","msg":"#ranListManagerInstance.AddNbIdentity - RAN name: enB_macro_001_001_0019b0 - Successfully added nodeb identity","mdc":{"time":"2023-08-08 09:32:46.962"
}}
{"crit":"INFO","ts":1691487166962,"id":"E2Manager","msg":"#E2TAssociationManager.AssociateRan - Associating RAN enB_macro_001_001_0019b0 to E2T Instance address: 10.103.222.218:38000","mdc":{"time":"2023-08-08
 09:32:46.962"}}
{"crit":"INFO","ts":1691487166962,"id":"E2Manager","msg":"[E2 Manager -> Routing Manager] #RoutingManagerClient.sendMessage - POST url: http://service-ricplt-rtmgr-http:3800/ric/v1/handles/associate-ran-to-e2t
, request body: [{\"E2TAddress\":\"10.103.222.218:38000\",\"ranNamelist\":[\"enB_macro_001_001_0019b0\"]}]","mdc":{"time":"2023-08-08 09:32:46.962"}}
{"crit":"INFO","ts":1691487166966,"id":"E2Manager","msg":"[Routing Manager -> E2 Manager] #RoutingManagerClient.sendMessage - success. http status code: 201","mdc":{"time":"2023-08-08 09:32:46.966"}}
{"crit":"INFO","ts":1691487166967,"id":"E2Manager","msg":"#RanConnectStatusChangeManager.ChangeStatus - RAN name: enB_macro_001_001_0019b0, currentStatus: UNKNOWN_CONNECTION_STATUS, nextStatus: CONNECTED","mdc
":{"time":"2023-08-08 09:32:46.967"}}
{"crit":"INFO","ts":1691487166967,"id":"E2Manager","msg":"#RanConnectStatusChangeManager.setEvent - Connectivity Event for RAN enB_macro_001_001_0019b0 is: enB_macro_001_001_0019b0_CONNECTED","mdc":{"time":"20
23-08-08 09:32:46.967"}}
```



```
{"crit":"INFO","ts":1691487167052,"id":"E2Manager","msg":"#E2TAssociationManager.AssociateRan - successfully associated RAN enB_macro_001_001_0019b0 with E2T 10.103.222.218:38000","mdc":{"time":"2023-08-08 09:
32:47.052"}}
{"crit":"INFO","ts":1691487167052,"id":"E2Manager","msg":"#E2SetupRequestNotificationHandler.handleSuccessfulResponse - payload: <E2AP-PDU><successfulOutcome><procedureCode>1</procedureCode><criticality><rejec
t/></criticality><value><E2setupResponse><protocolIEs><E2setupResponseIEs><id>4</id><criticality><reject/></criticality><value><GlobalRIC-ID><pLMN-Identity>131014</pLMN-Identity><ric-ID>10101010110011001110</r
ic-ID></GlobalRIC-ID></value></E2setupResponseIEs><E2setupResponseIEs><id>9</id><criticality><reject/></criticality><value><RANfunctionsID-List><ProtocolIE-SingleContainer><id>6</id><criticality><ignore/></cri
ticality><value><RANfunctionID-Item><ranFunctionID>0</ranFunctionID><ranFunctionRevision>0</ranFunctionRevision></RANfunctionID-Item></value></ProtocolIE-SingleContainer><ProtocolIE-SingleContainer><id>6</id><
criticality><ignore/></criticality><value><RANfunctionID-Item><ranFunctionID>1</ranFunctionID><ranFunctionRevision>0</ranFunctionRevision></RANfunctionID-Item></value></ProtocolIE-SingleContainer></RANfunction
sID-List></value></E2setupResponseIEs></protocolIEs></E2setupResponse></value></successfulOutcome></E2AP-PDU>","mdc":{"time":"2023-08-08 09:32:47.052"}}
{"crit":"INFO","ts":1691487167052,"id":"E2Manager","msg":"#E2SetupRequestNotificationHandler.handleSuccessfulResponse - RAN name: enB_macro_001_001_0019b0 - RIC_E2_SETUP_RESP message has been built successfull
```

Near-RT RIC software Architecture


RIC cluster
aux-entry/*
AdmControl
MC
ML
kpimon
:80
:443
[auxip]:32080
[auxip]:32443
ricxapp namespace
aux-entry/*
Robot tester
vescollector/
helm/
e2mgr/
appmgr/
helm/
aux-entry/*
E2 Termination
E2 Manager
Routing Manager
Sub Manager
xApp Manager
A1 Mediation
DbaaS
VESPA
Jaeger AIO
:80
:443
[auxip]:32080
[auxip]:32443
ricplt namespace
Kong
Chart Museum
Ingress Controller (Kong)
:32080 (http)
:32443 (tls)
ricinfra namespace
a1mediator/
RIC K8S Cluster
Prometheus
fluentd
ElasticSearch
Docker host
VM/ Bare metal
Docker host
VM/ Bare metal
Docker host
VM/ Bare metal
IPv4 and IPv6 networking