OSIRIS – Software User's Guide

Document – OSR-SUG-Rev01
Prototype Project Agreement (PPA) No. 01/IWRP Base Agreement No. 2021-562
Project Title - IWRP-21-PAC-0163-008
Submitted – 25 Jul 2024

Prepared For:
Marine Corps Tactical Systems Support Activity
Agreements Officer Representative (AOR)

Prepared By:
Lockheed Martin Rotary and Mission Systems
9500 Godwin Drive
Manassas, VA 20110-4116
Cage Code: 52088

Table of Contents

List of Figures

List of Tables

Document Change History

Rev	Date	Change Author	Change Summary
Rev 01	25 Jul 2024	Kyle Clark	Initial Release

1 PURPOSE AND SCOPE

This document covers general interactions with the OSIRIS system including starting, stopping, configuration, and debug. This document aims to educate new users to a satisfactory proficiency level to perform many basic system tasks and serves as a reference for existing users. This document assumes the OSIRIS system has already been deployed and configured.

This document does not provide an exhaustive education in 5G private networks nor is it a deployment guide.

2 ASSUMPTIONS

This document assumes basic familiarity with 5G concepts and architecture including the primary components, 3GPP interfaces, and traffic flows. This document also assumes users a familiar with the Linux Command Line Interface (CLI) and Kubernetes. This document will provide all the information required to control the OSIRIS system. Credentials for each system component will be delivered with this document for system access. This document assumes all baseband equipment has been powered on.

Note: All commands should be executed as a privileged user unless otherwise stated
Note: The "$" represents a command line input – do not type the $, it is an identifier of a command line input (CLI).

3 MOBAXTERM AND SERVER ACCESS

3.1 OVERVIEW

This section provides an overview of MobaXterm – what it is, what it is used for, and how to use it to access the various components of the OSIRIS. MobaXterm is a key piece of software used by the Lockheed Martin team to interact with the OSIRIS. It provides a number of important remote networking tools such as SSH, RDP, SFTP, etc. As part of the delivered system, the management OSIRIS Windows Getac laptop comes with MobaXterm installed and configured for accessing the various components of both the Nomadic Tower and Mobile Relay. This machine will serve as the primary human machine interface (HMI).

3.2 SERVER ACCESS

The image below shows the home page of the MobaXterm application when it is launched. Outlined in red are the pre-configured SSH sessions. These sessions are named after the various components of the OSIRIS system. To access a component of the system, simple double click on the corresponding session. This will spawn a SSH session into the respective server.

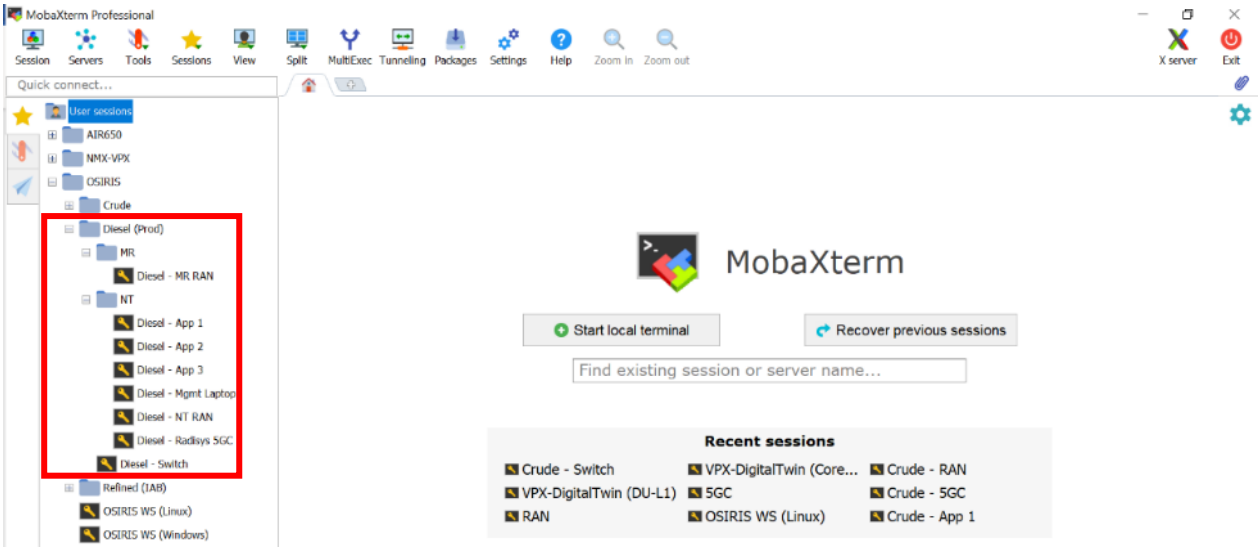

Figure 3.2-1: MobaXterm Home Page

4 GNU SCREEN

GNU Screen is a terminal multiplexor. In other words, GNU Screen provides an alternative method for creating and managing additional shell sessions. With GNU Screen, we can avoid opening many different MobaXterm sessions. Instead, we need only a single MobaXterm session for each server.

The original shell session opened via MobaXterm is referred to as the parent terminal. The additional shell sessions created using GNU Screen are the child terminals or virtual consoles. They're identical to a typical console with the added benefit of being able to be terminated without effecting other terminals. Below is a screen cheat sheet. More information about screen can be found here: https://www.gnu.org/software/screen/.

Table 4-1: GNU Screen Commands

Purpose	Screen Command
List Current Screen Sessions	screen -ls
Create New Screen Session	screen -S *new_session_name*
Delete Existing Screen Session	screen -XS *session_name* quit
Attach to Existing Screen Session	screen -r *session_name*
Exist a Screen Session	CTRL + A, D

5 USER DEFINED ALIASES

In Linux, an alias is a short, easy to remember name for a command or sequence of commands. There are a number of user defined aliases on the Nomadic Tower gNB and the Mobile Relay gNB. The user of this guide should become familiar with the available aliases to increase system operation efficiency.

To list all aliases currently defined on the system (any Linux system), use the following command…
$ alias

It is encouraged to view the available user defined aliases on each server (e.g. the gNB, the 5GC, the Data Network hosts, etc.).

6 SYSTEM INITIALIZATION

This section outlines the process to verify the primary components of the OSIRIS system are healthy and that the 5G system is ready to be started.

6.1 DATA NETWORK

6.1.1 Overview

This section is an overview of the DN architecture and captures how to assess the state of the Kubernetes-hosted Data Network to ensure all applications are healthy.

The Data Network (DN) is a Kubernetes Cluster consisting of one or many Kubernetes Server Nodes. The Data Network hosts a set of pre-installed infrastructure applications and services and can be used to install and run additional Kubernetes, Helm, Docker or Linux Services. These applications and services are accessible to 5G user equipment (UE) connected to the 5G system. The DN applications can also be accessed by anyone connected to the local area network (LAN) – for instance, by another server hosting an application.

When covering the DN, this guide will not provide an exhaustive education in Kubernetes or containerization. It is expected that operators have some level of familiarity with Kubernetes and it's utilities.

6.1.2 Cluster and Application Health – Kubectl

When the DN servers are powered on and complete boot sequences, the Kubernetes workloads are expected to initialize to a healthy state. To confirm this, we can use Kubectl. Kubectl is the command line tool for communicating with Kubernetes. Read more about Kubectl here (https://kubernetes.io/docs/reference/kubectl/).

To view the state of the DN infrastructure, access the Kubernetes Master Node (i.e. App 1) server using MobaXterm. Refer to Section 3: MobaXterm and Server Access for help.

After an SSH session has been opened to the App 1 server, use the following command…

$ kubectl get pods -A

```
root@app-node-1:/home/kclark# kubectl get pods -A
NAMESPACE              NAME                                                      READY   STATUS      RESTARTS        AGE
kube-system            helm-install-traefik-fwmb4                                0/1     Completed   0               60d
kube-system            helm-install-traefik-crd-zpjlq                            0/1     Completed   0               60d
harbors                harbor-portal-69df948cd6-jgn8b                            1/1     Running     2 (20d ago)     60d
kubeapps               kubeapps-internal-apprepository-controller-b64f45b7b-pvmdz 1/1     Running     2 (20d ago)     31d
harbors                harbor-redis-0                                            1/1     Running     2 (20d ago)     60d
harbors                harbor-database-0                                         1/1     Running     2 (20d ago)     60d
harbors                harbor-registry-d5c5b44dc-lhvwg                           2/2     Running     4 (20d ago)     60d
kubeapps               kubeapps-internal-dashboard-78dbd54b7d-k5v8l              1/1     Running     2 (20d ago)     31d
kubernetes-dashboard   dashboard-metrics-scraper-58d67b7d97-hcwd7                1/1     Running     0               20d
harbors                harbor-core-84966db8d7-stll6                              1/1     Running     6 (20d ago)     60d
kubeapps               kubeapps-786cdd5c85-x7xws                                 1/1     Running     7 (20d ago)     31d
kubeapps               chart-museum-6b66d76897-ccr9r                             1/1     Running     2 (20d ago)     25d
taks                   takserver-5f5d856b95-4bx5j                                1/1     Running     0               20d
kubernetes-dashboard   kubernetes-dashboard-8599d7db88-ztdwg                     1/1     Running     2 (20d ago)     45d
```

Figure 6.1-1: Example Data Network Status

Above is only an example screenshot. The important check here is that your system shows only "Completed" or "Running" in the "STATUS" column. This indicates all workloads are healthy.

Note: If any containers are in an "exited", "ImagePullBackOff", or "CrashLoopBackOff" state, there is likely an issue with one of the workloads, and debugging will be needed.

6.1.3 Cluster and Application Healthy – Kubernetes Dashboard

Please refer to Section 15:DATA NETWORK APPLICATION: TAK SERVER.

6.2 5G CORE NETWORK

This section captures assessing the state of the Docker-hosted Radisys 5GC software. All of the 5GC Network Virtual Functions (NVFs) are hosted as separate docker containers. A set of startup processes are executed when the 5GC server powers on. Those startup processes ensure the Docker containers start properly. This section covers the confirmation that those startup processes executed successfully and that the 5GC NVFs are all in a healthy state.

To view the state of the 5GC NVFs, access the 5GC server using MobaXterm. Refer to Section 3: MobaXterm and Server Access for help.

After an SSH session has been opened to the 5GC server, use the following command…

$ docker ps -a

```
root@node1:/home/kclark# docker ps -a
CONTAINER ID   IMAGE                      COMMAND                  CREATED      STATUS       PORTS                                                                  NAMES
fb7c0b73849f   upfsp:5.0.0                "/bin/bash -c 'sysct…"   4 weeks ago  Up 2 hours                                                                          5gc-upfsp
b3c8cb3a4bec   amfcomm:5.0.0              "./amfcomm"              4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-amfcomm
68a25a99a158   amfloc:5.0.0               "./amfloc"               4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-amfloc
aefcd8b6ffc9   amfueidgen:5.0.0           "./amfueidgen"           4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-amfueidgen
33bce73aad59   ee:5.0.0                   "./ee"                   4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-amfee
59f8461f324f   gateway:5.0.0              "./gateway"              4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-amfgw
9317742bb897   amfn2iwf:5.0.0             "/bin/bash -c 'sysct…"   4 weeks ago  Up 2 hours   8082/tcp, 0.0.0.0:38412->38412/sctp, :::38412->38412/sctp              5gc-amfn2iwf
6be65b3790a5   amfgnbmgr:5.0.0            "./gnbmgr"               4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-amfgnbmgr
dedaa9297853   upffp:5.0.0                "/bin/bash -c 'sysct…"   4 weeks ago  Up 2 hours                                                                          5gc-upffp
a16df2ab358c   udr:5.0.0                  "./udr_bin"              4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-udr
129fcfb059a3   udsf-notify:5.0.0          "./udsf-notify"          4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-udsf-notify
93759fe5ae3d   udsf:5.0.0                 "./udsf"                 4 weeks ago  Up 2 hours   0.0.0.0:8082->8082/tcp, :::8082->8082/tcp                              5gc-udsf
18880ce5655b   nssfnsselection:5.0.0      "./nssf"                 4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-nssf
6aac6d0c2c39   config-operator:5.0.0      "python configOperat…"   4 weeks ago  Up 2 hours                                                                          5gc-config-operator
7356b980727e   gui:5.0.0                  "/docker-entrypoint.…"   4 weeks ago  Up 2 hours   0.0.0.0:80->80/tcp, :::80->80/tcp                                      5gc-gui
c1cb365eaaa6   appserver:5.0.0            "python startPython.…"   4 weeks ago  Up 2 hours                                                                          5gc-appserver
9aca4fcefb2a   smfnidd:5.0.0              "./smfnidd"              4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-smfnidd
5637dc771dd0   smfee:5.0.0                "./smfee"                4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-smfee
daa5f5b6d731   gateway:5.0.0              "./gateway"              4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-smfgw
8102a66f5566   smfn4iwf:5.0.0             "./smfn4iwfnew"          4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-smfn4iwf
dcd034482603   smfipm:5.0.0               "./smfipmnew"            4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-smfipm
2a8455f58dfb   smfpdusession:5.0.0        "./smfpdusessionnew"     4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-smfpdusession
cbded9574aee   udm:5.0.0                  "./udm_bin"              4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-udm
06992d5c18f3   db-watch:5.0.0             "./db-watch"             4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-db-watch
5b6a3f72b6fc   ausfv2:5.0.0               "./ausfv2"               4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-ausf
6622456e9e3c   nrfnfm:5.0.0               "./nrf"                  4 weeks ago  Up 2 hours   0.0.0.0:8083->8082/tcp, :::8083->8082/tcp                              5gc-nrfnfm
4aa59647f14f   timer:5.0.0                "./timer"                4 weeks ago  Up 2 hours   8082/tcp                                                               5gc-timer
a372d274b51b   oamsysrepo:5.0.0           "bash -c /usr/local/…"   4 weeks ago  Up 2 hours   0.0.0.0:830->830/tcp, :::830->830/tcp                                  5gc-oamsysrepo
1b86c79df455   elasticsearch:7.17.8       "/bin/tini -- /usr/l…"   4 weeks ago  Up 2 hours   9200/tcp, 9300/tcp                                                     5gc-elasticsearch
3ab5f4bc57da   mongo:6.0.8                "docker-entrypoint.s…"   4 weeks ago  Up 2 hours   27017/tcp                                                              5gc-mongodb
```

Figure 6.2-1: 5GC NVF State

Confirm that each entry in the “STATUS” column displays “Up X hours”.
Note: If any containers are in an “exited” state, debugging will be needed.

6.3 5G GNODEB (NT AND MR)

This section captures assessing the initial state of the gNodeB. A set of startup processes are executed when the gNodeB server powers on. Those startup processes ensure the initial network configuration is implemented and that the precision timing protocol processes are started. This section covers the confirmation that those startup processes executed successfully, the gNodeB network interfaces are configured properly, and that the system is synchronized. Access the gNodeB server using MobaXterm. Refer to Section 3: MobaXterm and Server Access for help.

The following steps should be performed on the Nomadic Tower and the Mobile Relay.

After an SSH session has been opened, use an alias to determine if the network interface card is synchronized to the 1 Pulse Per Second (PPS) signal provided via our GPS connection. To check the state of the clock generation unit (CGU) embedded with the Intel E810T network interface card (NIC), use the following command...

$ checktiming

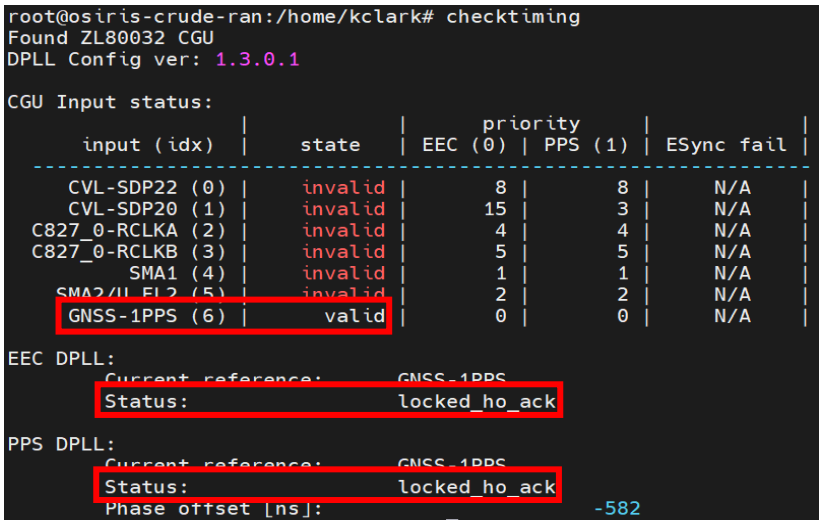

Figure 6.3-1: Clock Generation Unit (CGU) State

Confirm your output matches the above. Specifically paying attention to the values outlined in red. Confirm from the example above that..

1) System is receiving valid 1 PPS input from GNSS receiver
2) Digital Phase Locked Loop (DPLL) state is in hold over

Note: Your “status” for the DPLLs may be “locked_ho_acq” - this is acceptable
Note: If your output does not match, debugging may be needed.

Confirm the CGU in the E810T is synchronized, then confirm the startup processes were executed successfully when the server powered on. To do this, verify that three screen sessions exist on the system. Use the following command…

$ screen -ls

```
root@osiris-crude-ran:/home/5gran/RAN/ActiveConfigs# sls
There are screens on:
        6554.nic1ptp    (06/27/2024 06:58:22 PM)        (Detached)
        6550.phc2sys    (06/27/2024 06:58:21 PM)        (Detached)
        6548.ts2phc     (06/27/2024 06:58:21 PM)        (Detached)
3 Sockets in /run/screen/S-root.
```

Figure 6.3-2: List of Screen Sessions

See below for a brief explanation for each screen session.

- ts2phc – Synchronizes Physical Hardware Clock (PHC) in NIC to Timestamps from GNSS Module
- phc2sys – Synchronizes System Clock on Server to NIC PHC
- nic1ptp – Sends PTP to the Radio Units

Note: If your output does not contain all three screen sessions, debugging may be needed.

6.4 RADIO UNITS (ALL)

This section captures the initialization of the Benetel RAN650 radio unit. The initialization process occurs in two steps. First, the Benetel attempts to synchronize to the PTP packets sent from the gNodeB (nic1ptp screen session in the previous section). After PTP synchronization is successful, the Benetel goes through a configuration process where attributes such as Center Frequency, Bandwidth, TDD Slot Pattern, etc. are established.

The Benetel radio units are NOT connected directly to the nomadic tower switch. Instead, each Benetel at the Nomadic Tower and Mobile Relay are connected to their respective gNodeB. This is important because this means the Benetel radios are accessed from the Nomadic Tower gNodeB or the Mobile Relay gNodeB i.e. SSH cannot be directly accessed from the Benetel radios from the management laptop.

The following steps should be performed for each radio at the Nomadic Tower and the Mobile Relay.

Access the gNodeB server using MobaXterm. Refer to Section 3: MobaXterm and Server Access for help. You can start with the Nomadic Tower first, then move to the Mobile Relay.

After an SSH session has been opened, use an alias to access each radio unit. Use the following alias to access Radio Unit 1. Then tail a log file to determine synchronization state.

$ ru1
$ tail -f /var/log/pcm4l

Monitor the file until the following output is displayed…

```
RE::Debug: 2020-02-07 15:55:20 247940640 ns [2, Supervisor] Enter frequency locked state.
RE::Debug: 2020-02-07 15:55:20 249083720 ns [2, Supervisor] Set warm start DCO valid to true
RE::Debug: 2020-02-07 15:55:20 250053320 ns [2, Supervisor] Tracker#0:
RE::Debug: 2020-02-07 15:55:20 250811140 ns [2, Supervisor]     Master port ID: 50:7c:6f:ff:fe:30:fb:12.2
RE::Debug: 2020-02-07 15:55:20 251696680 ns [2, Supervisor]     Current reference master: Yes
RE::Debug: 2020-02-07 15:55:20 252311320 ns [2, Supervisor]     Freq locked: Yes
RE::Debug: 2020-02-07 15:55:20 252879240 ns [2, Supervisor]     Time locked: No
RE::SyncAnalysis: 2020-02-07 15:55:20 260247920 ns [2, Supervisor] (3066) LO state: 'Frequency Locked' to 'Time Locked'    Event: 'LO time locked'
.
RE::Debug: 2020-02-07 15:55:20 261063140 ns [2, Supervisor] Enter time locked state
RE::SyncAnalysis: 2020-02-07 15:55:20 261612620 ns [2, Supervisor] (3563) Holdover Qualification Timeout: 0 seconds.
RE::Debug: 2020-02-07 15:55:20 262219960 ns [2, Supervisor] Tracker#0:
RE::Debug: 2020-02-07 15:55:20 262747820 ns [2, Supervisor]     Master port ID: 50:7c:6f:ff:fe:30:fb:12.2
RE::Debug: 2020-02-07 15:55:20 263240900 ns [2, Supervisor]     Current reference master: Yes
RE::Debug: 2020-02-07 15:55:20 263830880 ns [2, Supervisor]     Freq locked: Yes
RE::Debug: 2020-02-07 15:55:20 264385960 ns [2, Supervisor]     Time locked: Yes
RE::SyncAnalysis: 2020-02-07 15:55:20 264899320 ns [0, Main] (3564) Holdover In-Spec Qualified.
RE::SyncAnalysis: 2024-07-01 18:39:21 013078105 ns [3, Tracker#0] (3045) UL packet rate (0.57) is different from the nominal rate (1.00).
RE::SyncAnalysis: 2024-07-01 18:39:21 645102456 ns [3, Tracker#0] (3240) offset: -1.0 ns    delay: 289.0 ns
```

Figure 6.4-1: Benetel RU Synchronization Status

Once the output is displayed, the Benetel radios have been successfully synchronized to our PTP signal. Next, confirm the configuration of the Benetel is completed correctly.

$ tail -f /tmp/logs/radio_status

Monitor the file until the following output is displayed…

```
[INFO] Modifying the TDD pattern from default to custom
[INFO] Transmission enabled (4x4)
[INFO] Radio bringup complete
 19:34:08 up 6 min,  load average: 0.33, 0.37, 0.20
[INFO] Launching o_ru_app
```

Figure 6.4-2: Benetel RU Configuration State

Reference each systems alias for accessing other radios.

7 SYSTEM START

Once all system components have been verified and are healthy the cell can be started Fortunately, this process is very simple. No action is required on the DN or 5GC. This section assumes successful completion of Section 6. If it has not been, please return to that section and complete it.

7.1 GNODEB SYSTEM START

We must first access the gNodeB server using MobaXterm. Refer to Section 3 MobaXterm and Server Access for help. You can start with the Nomadic Tower first, then move to the Mobile Relay. Feel free to use your existing SSH session to the gNodeB if one is already open.

These steps will begin RF transmission!! Be aware of HERP/HERO guidelines for safe distancing from transmitters!!

The following steps should be performed on the Nomadic Tower gNB and the Mobile Relay gNB.

After an SSH session has been opened, use an alias to access the startup scripts directory.

$ startup
$ ./StartAll.sh
$ screen -ls

Confirm three new screen sessions have been created…

```
root@osiris-crude-ran:/home/5gran/RAN/StartupScripts# sls
There are screens on:
        24701.du        (07/01/2024 07:36:22 PM)        (Detached)
        24697.l1        (07/01/2024 07:36:22 PM)        (Detached)
        24692.cu        (07/01/2024 07:36:22 PM)        (Detached)
        24317.nic1ptp   (07/01/2024 06:45:16 PM)        (Detached)
        24311.phc2sys   (07/01/2024 06:45:16 PM)        (Detached)
        24307.ts2phc    (07/01/2024 06:45:16 PM)        (Detached)
6 Sockets in /run/screen/S-root.
```

Figure 7.1-1: List of Screen Sessions after System Start

See below for a brief explanation for each screen session.

- cu – Centralized Unit executable
- l1 – High Physical Layer executable
- du – Distributed Unit executable

If your output does not contain these three new screen sessions, debugging may be needed.

7.2 SYSTEM START CONFIRMATION

After the StartAll script has been execute, confirm that the system started properly. The Benetel radios should be started. From the gNodeB, SSH to the Benetel radios. Reference the aliases on the system to access each Benetel radio – below is an example for radio 1.

The following steps should be performed on each radio at the Nomadic Tower and the Mobile Relay.

$ ru1

$ kpi.sh

```
root@benetelru:~# kpi.sh
 SAMPLE_TIME        | RX_TOTAL | RX_ON_TIME | RX_EARLY | RX_LATE | RX_ON_TIME_C | RX_EARLY_C | RX_LATE_C | TX_TOTAL
 20:55:09.345402    | 64586    | 54560      | -0       | -0      | 9976         | -0         | -0        | 67368
 20:55:10.346029    | 64318    | 54340      | -0       | -0      | 10008        | -0         | -0        | 67212
 20:55:11.345337    | 64471    | 54492      | -0       | -0      | 10004        | -0         | -0        | 67188
 20:55:12.347792    | 64439    | 54296      | -0       | -0      | 10000        | -0         | -0        | 67392
 20:55:13.344078    | 64415    | 54312      | -0       | -0      | 9988         | -0         | -0        | 66978
 20:55:14.344499    | 64473    | 54448      | -0       | -0      | 9996         | -0         | -0        | 67230
```

Figure 7.2-1: Benetel eCPRI Packet Counters

The above output shows packet counters for the enhanced common public radio interface (eCPRI). Confirm that your output looks similar. Exactly matching this output is not necessary. However, it is important that you have “0” in every EARLY and LATE column – like the output shown above. This tells us that the Benetel is receiving and sending all eCPRI traffic on time.

Next, confirm that the Benetel is transmitting RF energy using the following command…

$ TXMeanPower

```
root@benetelru:~# TXMeanPower
TX 1 TSSI: 6.50269 dBm
TX 2 TSSI: 5.20406 dBm
TX 3 TSSI: 7.05467 dBm
TX 4 TSSI: 4.35421 dBm
```

Figure 7.2-2: Benetel Transmit Power

Confirm that your output looks similar to this. Your values for TX 1 – TX 4 may be larger or smaller, but the important thing is that there is RF energy! Note: If your output does not match, debugging may be needed.

8 SYSTEM SHUTDOWN

This section captures the steps to stop 5G system transmission and shutting down the baseband equipment including the data network, 5G core, gNB, and radios.

8.1 5G CELL SHUTDOWN

First access the gNodeB server using MobaXterm. Refer to Section 3 MobaXterm and Server Access for help. You can start with the Nomadic Tower first, then move to the Mobile Relay. Feel free to use your existing SSH session to the gNodeB if one is already open.

The following steps should be performed on the Nomadic Tower gNB and the Mobile Relay gNB.

After an SSH session has been opened, use an alias to access the startup scripts directory.
$ startup
$ ./KillAll.sh
$ screen -ls

```
root@osiris-crude-ran:/home/5gran/RAN/ActiveConfigs# sls
There are screens on:
        6554.nic1ptp    (06/27/2024 06:58:22 PM)        (Detached)
        6550.phc2sys    (06/27/2024 06:58:21 PM)        (Detached)
        6548.ts2phc     (06/27/2024 06:58:21 PM)        (Detached)
3 Sockets in /run/screen/S-root.
```

Figure 8.1-1: List of Screen Sessions after System Stop

Confirm your output matches the above - the cu, l1, and du Screen sessions are destroyed. If any of the cu, l1, or du Screen sessions persist, execute the *KillAll.sh* script again!

If you intend to continue testing, do not complete Section 8.2 Server Shutdown. The Server Shutdown should only be completed when testing for the day has been completed.

8.2 SERVER SHUTDOWN

Shutting down each of the servers in the system is simple. The following command should be executed for each system component in the following order – Data Network (Application Node 1-3), 5G Core, and gNB.

Access each component in the order specified above using MobaXterm. Refer to Secion 3: MobaXterm and Server Access for help. Execute the following command…

$ sudo shutdown now

This command will turn each server off. After 5 minutes, remove each server from power. The Benetel does not have a shutdown procedure. To turn it off, simply remove it from power.

9 SYSTEM CONFIGURATION

This section outlines the primary touch points for configuring each of the 5G system components. Detailed guidance on configuration modification will be provided in supporting vendor documentation for each component respectively.

9.1 5G CORE NETWORK

The Radisys 5GCN has a single location for configuration of all NVFs. This is referred to as their Enhanced Management System (EMS). It is a browser based graphical user interface (GUI) that exposes configuration options for each NVF.

To access the EMS GUI, open Google Chrome on the management laptop. Navigate to http://12.12.3.11/ems

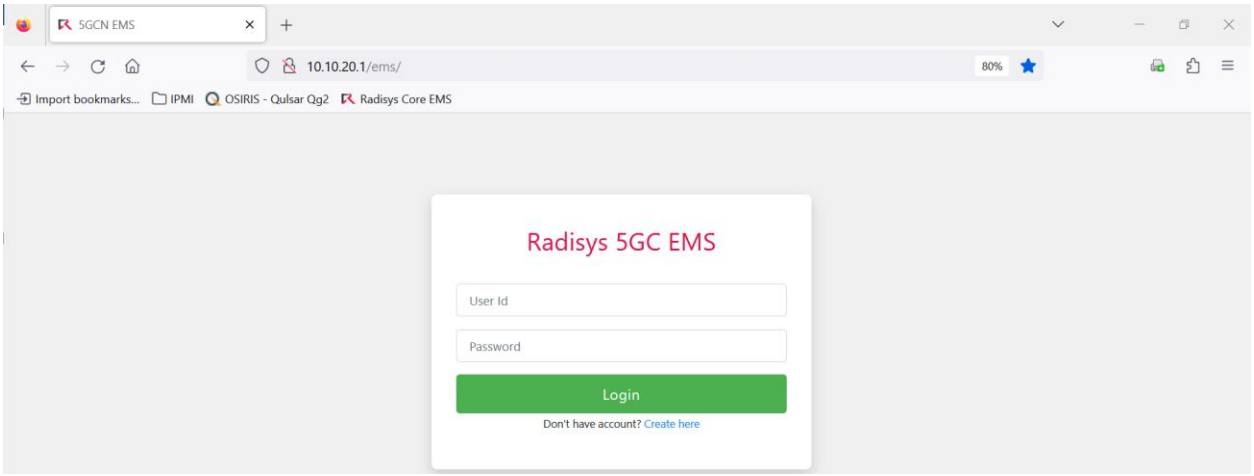

Figure 9.1-1: Radisys 5GCN EMS GUI

After signing in with the provided credentials, each NVF is displayed on the left hand column. These NVF sections all contain the various parameters required to configure the 5GCN including things like PLMN, NSSAI, N3/N6 IP address, etc. **There is A LOT of information in these configuration settings.** They should not be modified unless the user has a strong understanding of 5G concepts.

The Radisys user guide *Trillium_5GCN_EMS_User_Guide_R5.0.0_v1.pdf* provides detailed information about the configurations available in the EMS GUI. It is strongly recommended that operators of this system familiarize themselves with the available configuration.

9.2 5G GNB (NT AND MR)

As previously discussed, the OSIRIS 5G gNB is composed of three primary components, the Centralized Unit (CU), the Distributed Unit (DU), and the High Physical Layer (L1). These three components have various files associated with them that allow a user to configure 5G parameter, network interface information, CPU core pinning, and much more. This section will outline the location of those files and provide high level descriptions for each configuration file.

For convenience, each of the primary configurations have been symbolically linked to a single, common location to save the operator time when performing configuration modification. Using the following alias, the user can access the configuration files…

$ configs
$ ll

```
root@osiris-crude-ran:/home/5gran/RAN/ActiveConfigs# configs
root@osiris-crude-ran:/home/5gran/RAN/ActiveConfigs# ll
total 460
drwxr-xr-x  3 root root  4096 Jul  2 15:16 ./
drwxr-xr-x 10 root root  4096 Jun 11 20:23 ../
drwxr-xr-x  2 root root  4096 Jun 17 15:27 06172024_StableBackup/
-rwxr-xr-x  1 root root 76734 Jun 17 15:36 oam_3gpp_cell_cfg_mu1_1cell_flexran_3352-26MHz_BalancedSlot.xml*
-rwxr-xr-x  1 root root 76734 Jun 17 15:37 oam_3gpp_cell_cfg_mu1_1cell_flexran_3352-26MHz_DLHeavySlot.xml*
-rwxr-xr-x  1 root root 76734 Jun 20 16:38 oam_3gpp_cell_cfg_mu1_1cell_flexran_3352-26MHz_ULHeavySlot.xml*
-rwxr-xr-x  1 root root 76734 Jun 21 17:26 oam_3gpp_cell_cfg_mu1_1cell_flexran.xml*
-rwxr-xr-x  1 root root 71193 Jun 18 21:07 oam_3gpp_cu_sa_1du_1cell_flexran.xml*
-rwxr-xr-x  1 root root 39277 Jun 21 15:25 phycfg_xran.xml*
-rw-r--r--  1 root root  5832 Jun 19 20:03 sys_config_cu.txt
-rw-r--r--  1 root root  8183 Jun 19 20:04 sys_config_du.txt
-rwxr-x--x  1 root root 13946 Jun 17 15:22 xrancfg_sub6.xml*
```

Figure 9.2-1: gNB Active Configs Directory

Each of these files represents a configuration touch point for one of the primary 5G gNB components. The table below summarizes the relationship.

Table 9-1: gNB Configuration Files

File	Component	Purpose
oam_3gpp_cu_sa_1du_1cell_flexran.xml	CU	5G Parameters
sys_config_cu.txt	CU	Network/Hardware Information
oam_egpp_cell_cfg_mu1_1cell_flexran.xml	DU	5G Parameters
sys_config_du.txt	DU	Network/Hardware Information
xrancfg_sub6.xml	L1	Radio and 5G Parameters
phycfg_xran.xml	L1	Network/Hardware Information

Provided with this document are the *Trillium_5G_NR_gNB_Solution_OAM_Guide_R4.0.1_v0.1.pdf* and the *571741-flexran-refsol-l1-xml-cfg-ug_v22p11.pdf*. These user guides provides detailed information about the configurations available in the CU/DU and FlexRAN software, respectively.

9.3 RADIO UNITS (ALL)

The Benetel RAN650 radio unit (as of firmware version 1.0.2) has two primary configuration files. The table below summarizes the purpose of each.

Table 9-2: Benetel RU Configuration Files

File	Component	Purpose
/usr/sbin/radio_setup_a.sh	RU	Network/Hardware Information

/etc/ru_config.cfg	RU	5G Parameters

The radio_setup_a.sh file need not be modified after system deployment. This guide highlights its existence for awareness purposes. In this file, information such as the DU media access control (MAC) address and the DU virtual local area network (VLAN) address are configured.

The ru_config.cfg file contains configuration such as the multiple-input multiple-output (MIMO) configuration and the PRACH preamble format.

Other parameters such as center frequency, bandwidth, time division duplex (TDD) configuration, and synchronization mode are modified through scripts created by the Benetel team. Use the following command to view the available helper scripts…

$ benetel-help

```
root@benetelru:~# benetel-help
+-----------------------------------+-----------------------------------+
| Command                           | Description                       |
+-----------------------------------+-----------------------------------+
| O-RU Configuration                |                                   |
+-----------------------------------+-----------------------------------+
| cat /etc/benetel-rootfs-version   | Shows software version installed  |
| ifconfig eth0                     | Shows O-RU MAC address            |
| gaincontrol                       | Sets O-RU output power            |
| frequencycontrol                  | Sets O-RU center frequency        |
| set_bandwidth                     | Sets O-RU bandwidth               |
| vi /etc/ru_config.cfg             | Edits O-RU config file            |
| vi /etc/tdd.xml                   | Edits TDD config file             |
| vi /etc/eaxc.cfg                  | Edits eAxC ID config file         |
| setSyncModePtp                    | Sets sync mode to PTP             |
| setSyncModeGps                    | Sets sync mode to GPS/1PPS        |
| oru_dpd_reset                     | Resets DPD algorithm              |
| oru_fem_reset                     | Resets FEM                        |
+-----------------------------------+-----------------------------------+
| O-RU Status/Statistics            |                                   |
+-----------------------------------+-----------------------------------+
| benetel-help                      | Shows this help menu              |
| cat /tmp/logs/ru_information      | Shows O-RU product info           |
| cat /var/syncmon/sync-state       | Shows sync status (0 = locked)    |
| cat /var/log/pcm4l                | Shows PTP logs                    |
| cat /etc/ru-sync-mode             | Shows current sync mode           |
| cat /tmp/logs/radio_status        | Shows radio bringup log           |
| oru_status                        | Shows current radio status        |
| oru_dpd_stats                     | Shows DPD status                  |
| kpi.sh                            | Shows Key Performance Indicators  |
| RXMeanPower                       | Shows current O-RU mean RX power  |
| TXMeanPower                       | Shows current O-RU mean TX power  |
| reportRuStatus                    | Shows CSR status for debug        |
| oru_vlan_mac_info                 | Shows DU VLAN and MAC settings    |
+-----------------------------------+-----------------------------------+
```

Figure 9.3-1: Benetel RU Help Menu

This section covers an outline of the primary configuration touch points for the Benetel RU. Provided with this document is the *'Software User Guide for RANx50 CAT-A O-RUs (Dec 2023).pdf'*. This user guide provides detailed information about the configurations available in the RU.

10 SYSTEM LOGGING

This section will outline the primary locations for logs generated by each of the 5G system components. Detailed explanation of all available logging information will be provided in supporting vendor documentation for each component respectively.

10.1 DATA NETWORK

10.2 5G CORE NETWORK

As previously discussed, the 5GCN is composed of Docker-hosted, containerized NVFs. Each NVF has it's own logging that can be accessed through the docker CLI utility. Using the following command template, an operator can view log from any NVF…

Template: docker logs *<NVF Pod Name>*
$ docker logs 5gc-amfcomm

The *<NVF Pod Name>* can be substituted for the name of any docker pod running. This command will generate output to the console that contains the logs for the particular NVF.

The table below lists out a list of different scripts you can access on the 5GC. A script is a small piece of software or program that will automate some process for the user. On the right hand side is a description of what each script does. All these scripts live on the 5GC and can only be executed there. These scripts allow the user to easily restart the 5GC workloads and view the logging of a number NVFs.

Table 10-1: 5GC Logging Scripts

5GC Scripts	Purpose
5gc_restart.sh	Restarts the 5GC
getAMFLogs.sh	Helper script to print and follow AMF logs
getPDULogs.sh	Helper script to print and follow SMF PDU logs
getUPFLogs.sh	Helper script to print and follow UPF logs

10.3 5G GNODEB (NT AND MR)

As previously discussed, the OSIRIS 5G gNB is composed of three primary components, the Centralized Unit (CU), the Distributed Unit (DU), and the High Physical Layer (L1). These three components have various logs files associated with them are produced during system operation. This section will outline the location of those files and provide high level descriptions for each log file.

Table 10-2: gNB Log Files

File	Component	Purpose
console_cu.log	CU	Initialization Logs EGTPU Logs UDP Statistics PDCP Statistics SDAP Statistics
cu_stats_<date>.txt	CU	Same as console_cu.log

File	Component	Purpose
cu_<date>_part_<num>.txt	CU	Detailed Module Logs
console_du.log	DU	Initialization Logs Downlink Statistics Uplink Statistics Failure Statistics BLER Statistics Cell Throughput Statistics
du_stats_<date>.txt	DU	Same as console_du.log
du_<date>_part_<num>.txt	DU	Detailed Module Logs
console_l1.log	L1	Initialization Logs Latency Statistics eCPRI Packet Statistics Downlink Throughput Uplink Throughput Uplink BLER FEC Statistics Core Utilization
PhyStats	L1	Layer 2 and Layer 1 API Comms

The following table outlines where each file can be found.

Table 10-3: gNB Log File Locations

Component	Location
CU	/home/5gran/RAN/cuFiles/bin/
DU	/home/5gran/RAN/duFiles/bin/
L1	/home/5gran/RAN/l1Files/L12211/bin/nr5g/gnb/l1/

Currently, no documentation exists to provide a complete overview of all available logging from the vendors. A request has been made for such documentation to be provided by Radisys and Intel. In the meantime, it is suggested the operator familiarizes themselves with each of the log files discussed in this section.

10.4 RADIO UNITS (ALL)

The available logging on the Benetel RAN 650 radio unit is neatly captured in Figure 9.3-1: Benetel RU Help Menu.

11 NETWORK SWITCH

Documentation for configuring the network switch is provided in the Cisco documentation.

12 DATA NETWORK OVERVIEW

The Data Network (DN) is a Kubernetes Cluster consisting of one or many Kubernetes Server Nodes. The Data Network hosts a set of pre-installed infrastructure applications and services and can be used to install and run additional Kubernetes, Helm, Docker or Linux Services. These applications and services are accessible to UEs connected to the 5G system. This section of the guide will describe the Kubernetes Cluster, the pre-installed applications and services and how to access and use them, and how to use the Cluster to install additional Services.

12.1 DATA NETWORK ARCHITECTURE

The Data Network consists of one Master Node and may have 0, 1, or many additional worker nodes. Worker nodes can be added or removed to the cluster with instructions from the other Infrastructure guide without compromising the functions or services provided in the Cluster.

Each Node in the Data Network Cluster has the following utilities:

- Docker version 24.0.5
- ContainerD/CTR version 1.7.2
- Kubernetes version 1.28.5+k3s1
- Helm version 3.14.1

The Master Node has additional responsibilities and is responsible for the overall health, coordination and function of the Data Network cluster. The Master Node is running a Bind9 Server in Docker which is used as the primary DNS server for all of the Kubernetes Nodes including the Master Node. It is also installed as the Master for Kubernetes and as such is essential for the Data Node Cluster functionality and contains the Master Kubernetes Access Key (KUBECONFIG). The Master Node also runs a local HTTP Docker Registry with no credentials with the sole purpose of hosting the images required to run the full access-controlled Harbor registry in Kubernetes.

12.2 USING THE DATA NETWORK

The Data Network is designed primarily as a Helm and Kubernetes Applications Server. You can interact with the cluster using the provided and installed infrastructure services – which are individually explained later in the guide - or using them directly via command line utilities.

To be able to fully utilize the distributed cluster as well as utilize the pre-installed infrastructure utilities that are installed on the cluster, it is recommended to use Helm and Kubernetes to deploy additional workloads. However Docker is also installed and usable on the Data Network Nodes.

12.2.1 Using the Data Network via Command Line Utilities

Helm and Kubernetes are able to access the Cluster if they have the Key (furthermore referred to as KUBECONFIG). The Master Node is configured to use the Root KUBECONFIG which has full read/write permissions for all resources across all namespaces in the Kubernetes Cluster. The default root KUBECONFIG exists at /etc/rancher/k3s/k3s.yaml on the Master Server. The additional nodes will not by default have the Master KUBECONFIG present on them. For security and control purposes, it may be preferable to create additional KUBECONFIG keys with

more limited permissions for user access, which is explained in the separate K3S User Management guide. This will require the creator of the key to grant read/write permissions to specific objects and namespaces. If you choose to use the Master KUBECONFIG for access, care should be taken.

By precedent, user access is configured by placing a KUBECONFIG file as the ~/.kube/config file and setting the KUBECONFIG environment variable for that user to =~/.kube/config

The cluster is interacted with through the KUBECONFIG key, and interacting with Kubernetes will interact with the entire cluster including all of the additional Nodes. Using the KUBECONFIG contained on the Master (or any KUBECONFIG) will show all resources deployed and available through all of the Nodes in the cluster. And installing things using that key will deploy things to the entire cluster. Kubernetes will choose automatically to what physical node the workloads are deployed based on resource availability and load balancing. Kubernetes will also automatically Load Balance network requests to the physical node where the workload is deployed to. That means that if you use a KUBECONFIG that points to the Master to deploy a workload, it may actually get deployed to a worker node. And if you use the Master IP and a NodePort (for TCP) or ClusterIP ExternalIP (for UDP) Service, or Ingress Service to access that service, that network request will automatically get proxied to the correct worker node where that workload is actually running. This is explained here to help to reduce confusion as to why the Master IP or DNS name is used to access deployed Services or within the KUBECONFIG itself to interact with resources not necessarily deployed on the Master Server.

The cluster is accessible to be interacted with by any server that has a valid KUBECONFIG and has network access to that server and is available to reach the Kubernetes Server access port (port 6443). That means that if your local windows or linux computer can reach the Master Node IP and the firewall / networking rules allow connectivity to port 6443, then you can remotely control, access, or deploy workloads to the Kubernetes Cluster. In order to access the cluster, you will need a Kubectl utility, helm utility, and to set the KUBECONFIG for those services. Then you will be able to execute kubectl / helm commands like you were on the server itself. By default, KUBECONFIG keys that you create or will have their server IP set to 127.0.0.1 - but to use them remotely, you will just need to update the KUBECONFIG file to have the correct remotely accessible IP of the server.

12.2.2 Pre-installed Infrastructure Apps

The following is the list of pre-installed infrastructure applications/services and their access ports. These services are all accessed by using the IP address of the Master Node.

Table 12-1: Application URL Access

Application/Server	Port
Bind9 DNS Web Server	*master-node-ip-address:*10000
Kubernetes Dashboard Web Server	*master-node-ip-address:*30080
Kubeapps Web Server	*master-node-ip-address:*30250

Application/Server	Port
Openspeedtest Web Server	*master-node-ip-address:*30125
TAK Server Web Server	*master-node-ip-address:*30015
Harbor Web Server	osiris-app1.lmosiris.com:80
Iperf3 Server	*master-node-ip-address:*30030 or 30040

Each of the infrastructure apps have an installation directory with management/monitoring utilities and capabilities that exists on the Master Node at /root/kubernetes-apps.

Each of the installed applications has additional README.md information located in their installation directories. For example, for more information on TAK Server, see the /root/kubernetes-apps/standalone-TAKSERVER-4-9-23-withtranslators/takserver/README.md file.
Each of the installation applications has an included set of installation and monitoring utilities located at /root/kubernetes-apps/<APPLICATION>/ on the Master Node:

- get-ns-all.sh - get all of the currently existing namespaces on the Cluster
- get-pods-all.sh - get the pods installed for only this application
- get-svc-all.sh - get the services installed for only this application
- install-all.sh - install only this application
- reinstall-all.sh - reinstall only this application
- uninstall-all.sh - uninstall only this application

For more information, see the included INSTALLATION-INSTRUCTIONS.README for each application. These scripts are helper scripts that execute kubectl and helm commands, and interacting with those application installs using the helm or kubectl binaries is also a valid way to manage the system applications if you are comfortable using those utilities. But using the included helper utilities is recommended for managing the pre-installed infrastructure applications and services.

Additionally, applications can be monitored, managed, installed and uninstalled, profiled, logged and more using the included infrastructure applications Kubeapps and Kubernetes Dashboard described below.

13 DATA NETWORK APPLICATION: KUBEAPPS

Kubeapps is an in-cluster web-based application that enables users with a one-time installation to deploy, manage, and upgrade applications on a Kubernetes cluster.

For additional information – see https://kubeapps.dev/

Note: If you are attempting to reinstall Kubeapps and experience a namespace being stuck terminating – see the included documentation at /root/kubernetes-apps/standalone-Kubeapps/kubeapps/README.md

13.1 USING KUBEAPPS

Create token and login to Kubeapps
$ kubectl -n kubernetes-dashboard create token admin-user

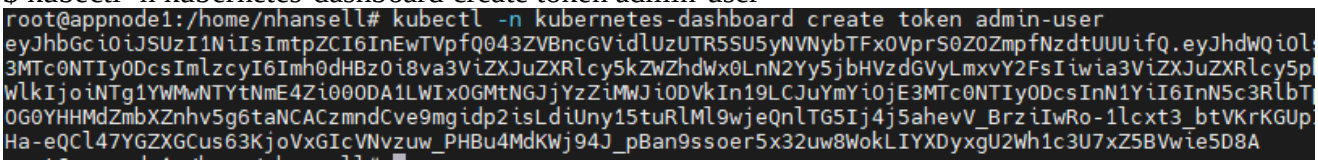

Figure 13.1-1: Kubernetes Token

Note: If you use a token from a user that doesn't have the correct permissions, then Kubeapps will spin at the getting cluster information stage.

Below will be images that walk through the process of logging into Kubeapps and explores different webpages available on the GUI.

Enter token generated on App-node-1 into the Kubeapps login page to login:

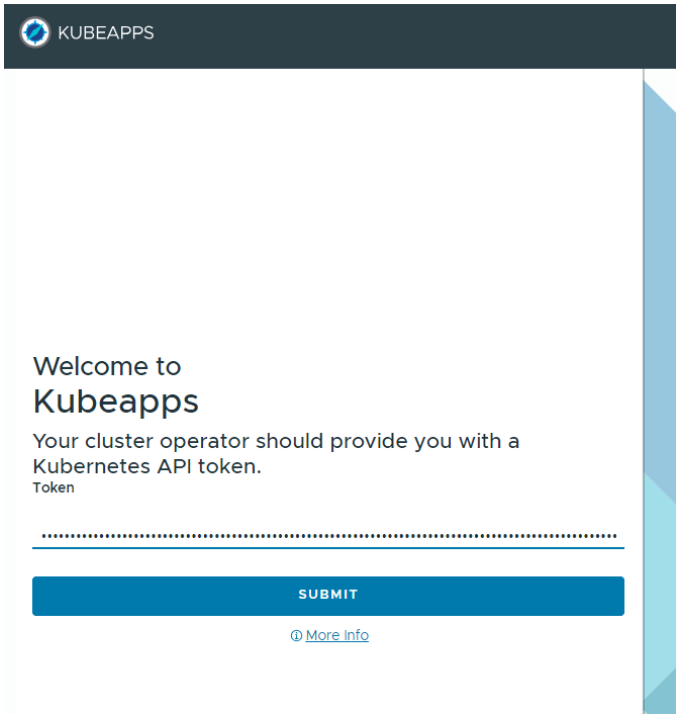

Figure 13.1-2: Kubeapps Token Login

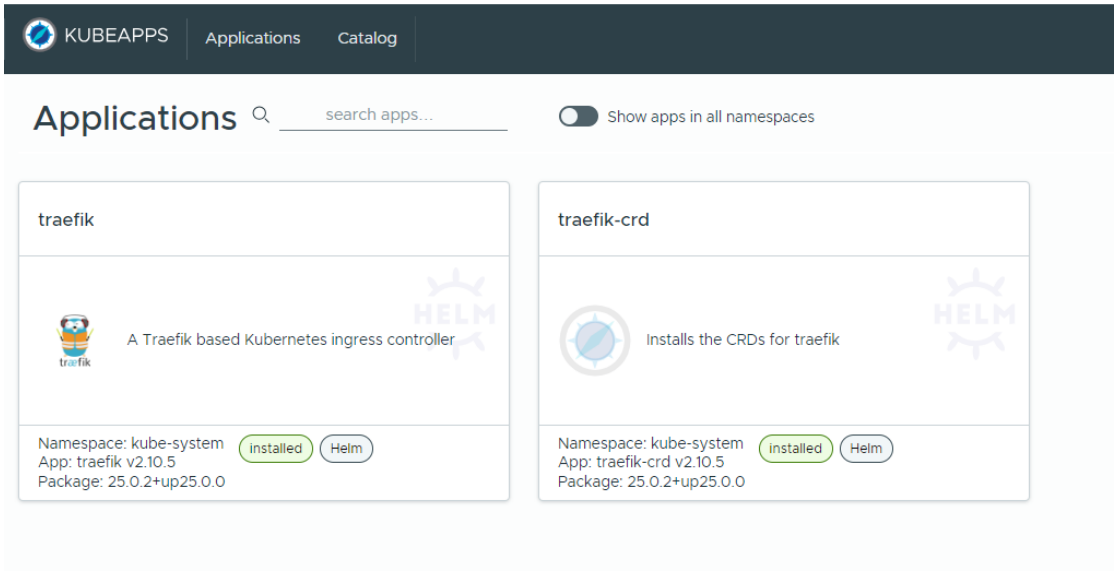

Figure 13.1-3: Kubeapps Main Page

Toggle the show all apps button to show all apps on available on the k3s cluster, and that can be terminated from the main page:

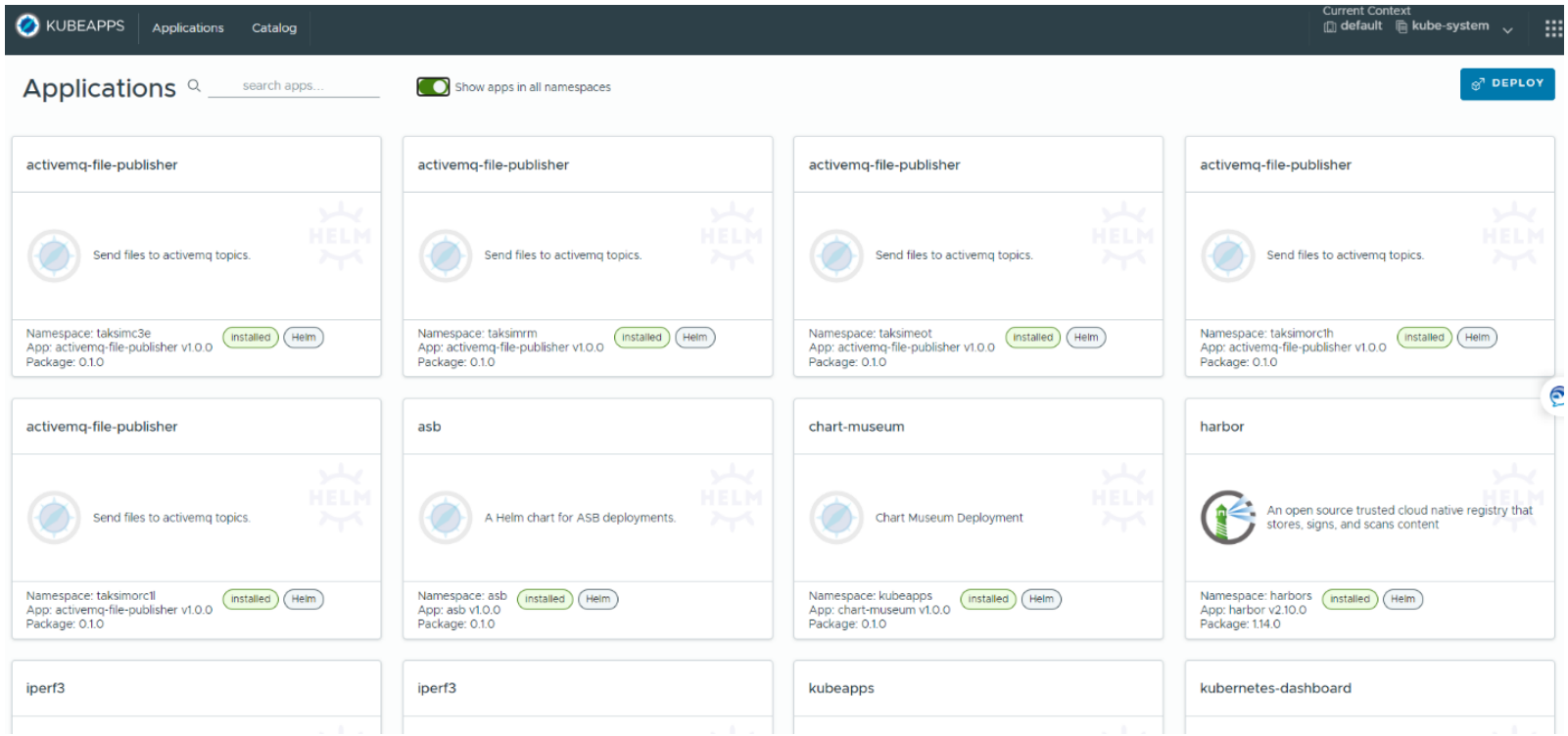

Figure 13.1-4: Kubeapps Application page

Click on the “Catalog” button at the top left of the screen to view all the apps are able to be deployed or redeployed from Kubeapps. All these apps have their Helm charts saved in a repository linked to Kubeapps and that enables Kubeapps to redeploy the apps:

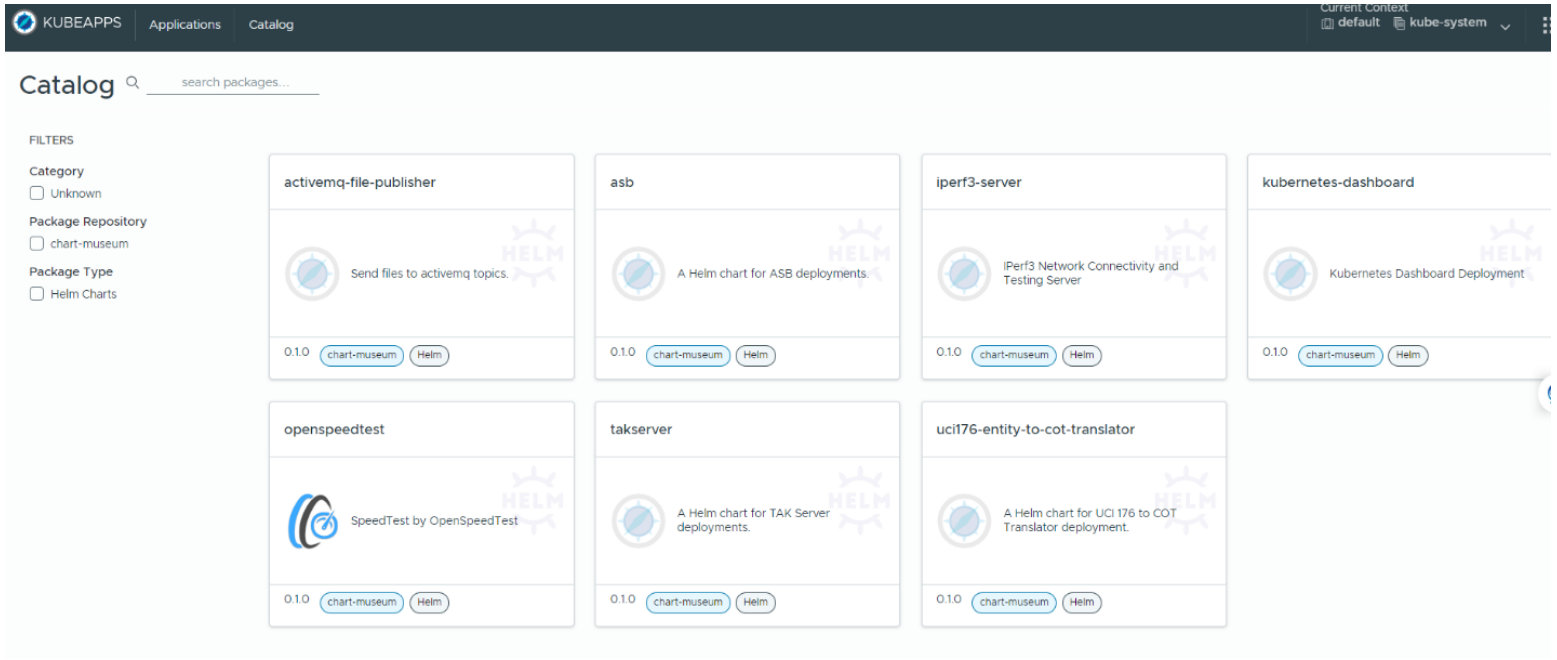

Figure 13.1-5: Kubeapps Category Page

Select the openspeedtest app to view it in more details and have to option to deploy it:

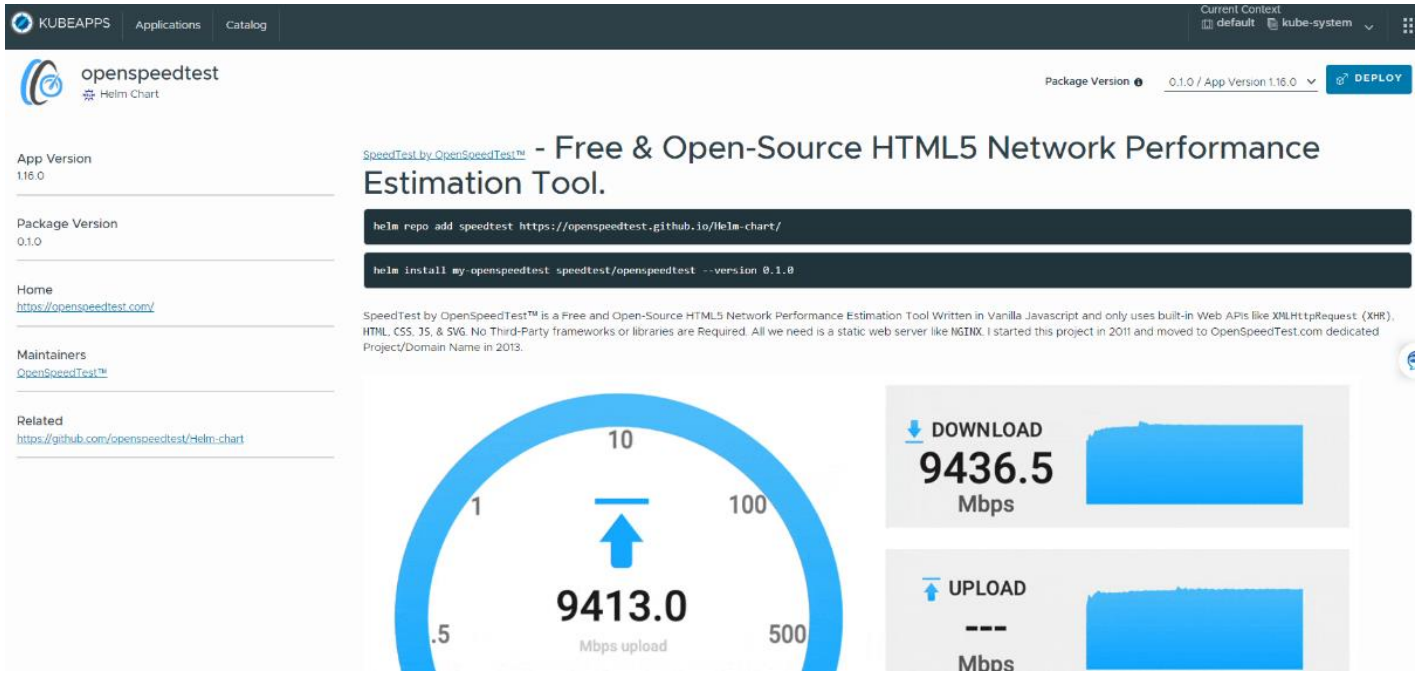

Figure 13.1-6: Kubeapps Openspeedtest App Page

Kubeapps - Deploy an Application:

This section reviews the steps taken to deploy an application on Kubeapps.

Click the “Current Context” drop down menu from the top right of the screen, change the “Namespace” to “openst” for this example using Openspeedtest. Use the namespace corresponding to app attempting to be deployed:

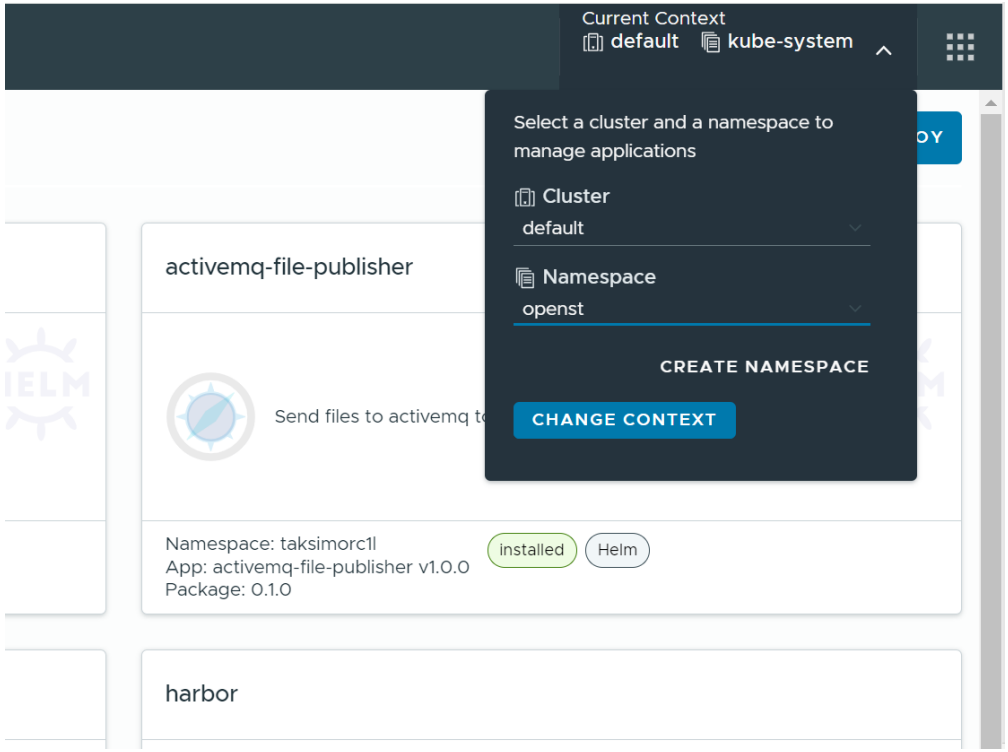

Figure 13.1-7: Kubeapps Change Context

After Current Context is changed to the namespace of the app that is being attempted to be deployed, in this case it is openspeedtest, navigate to Categories and the app attempting to be deployed, then click the blue deploy button at the top right of the screen:

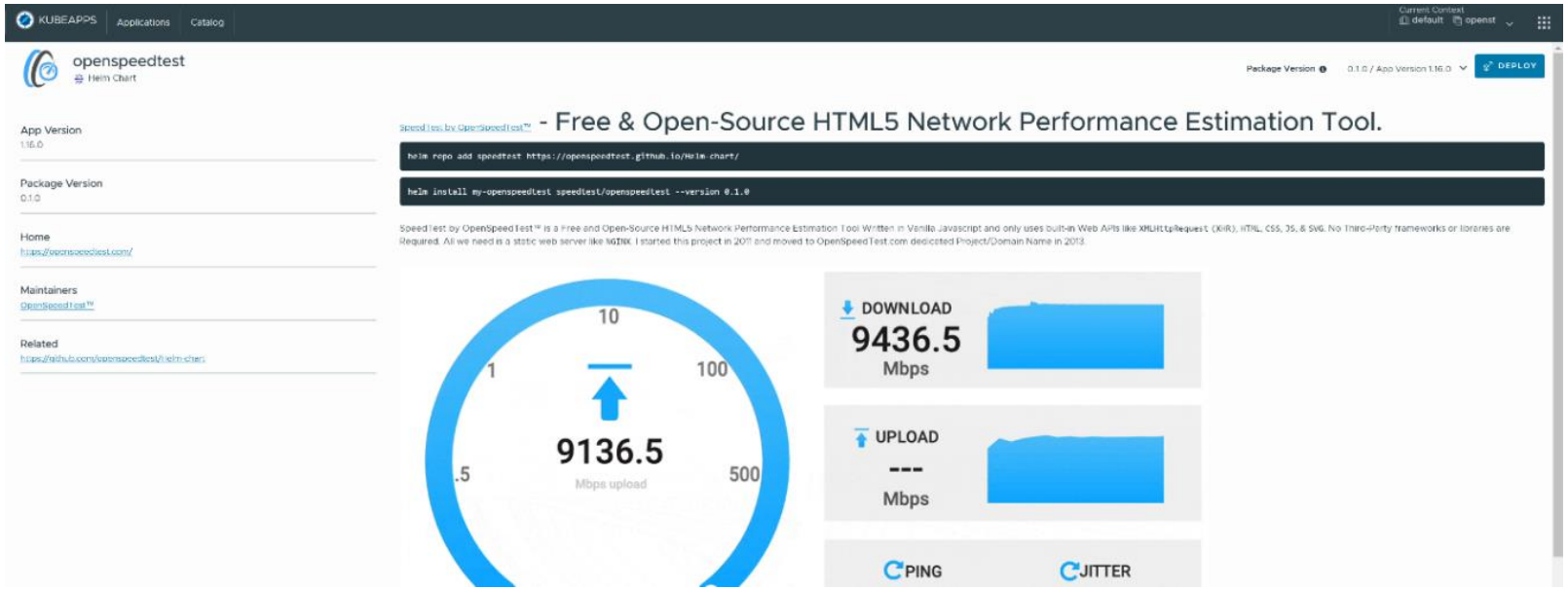

Figure 13.1-8: Kubeapps Application Page

Give the app being deployed and name and scroll to the bottom of the screen and click the deploy button:

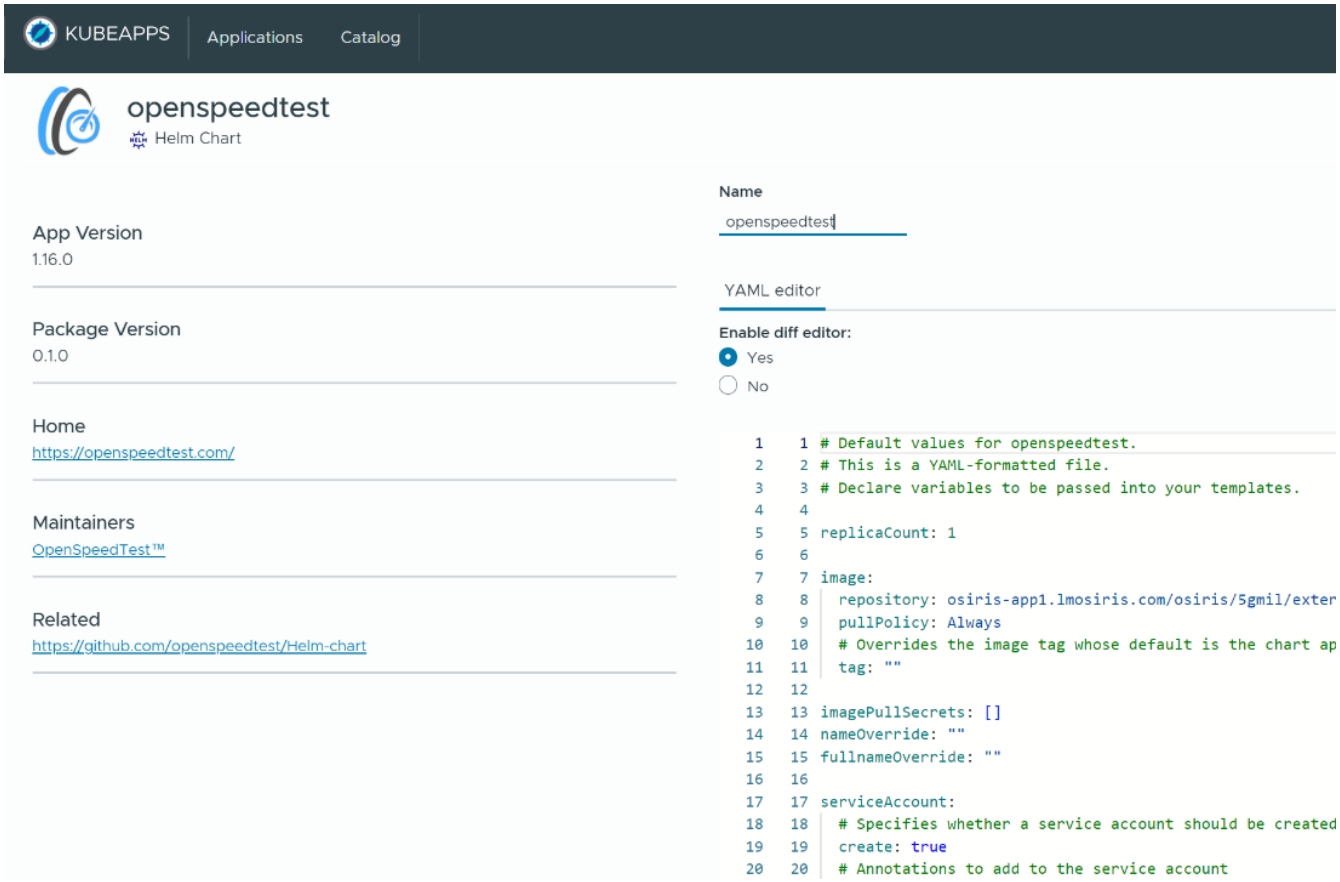

Figure 13.1-9: Kubeapps Application Deployment - Page 1

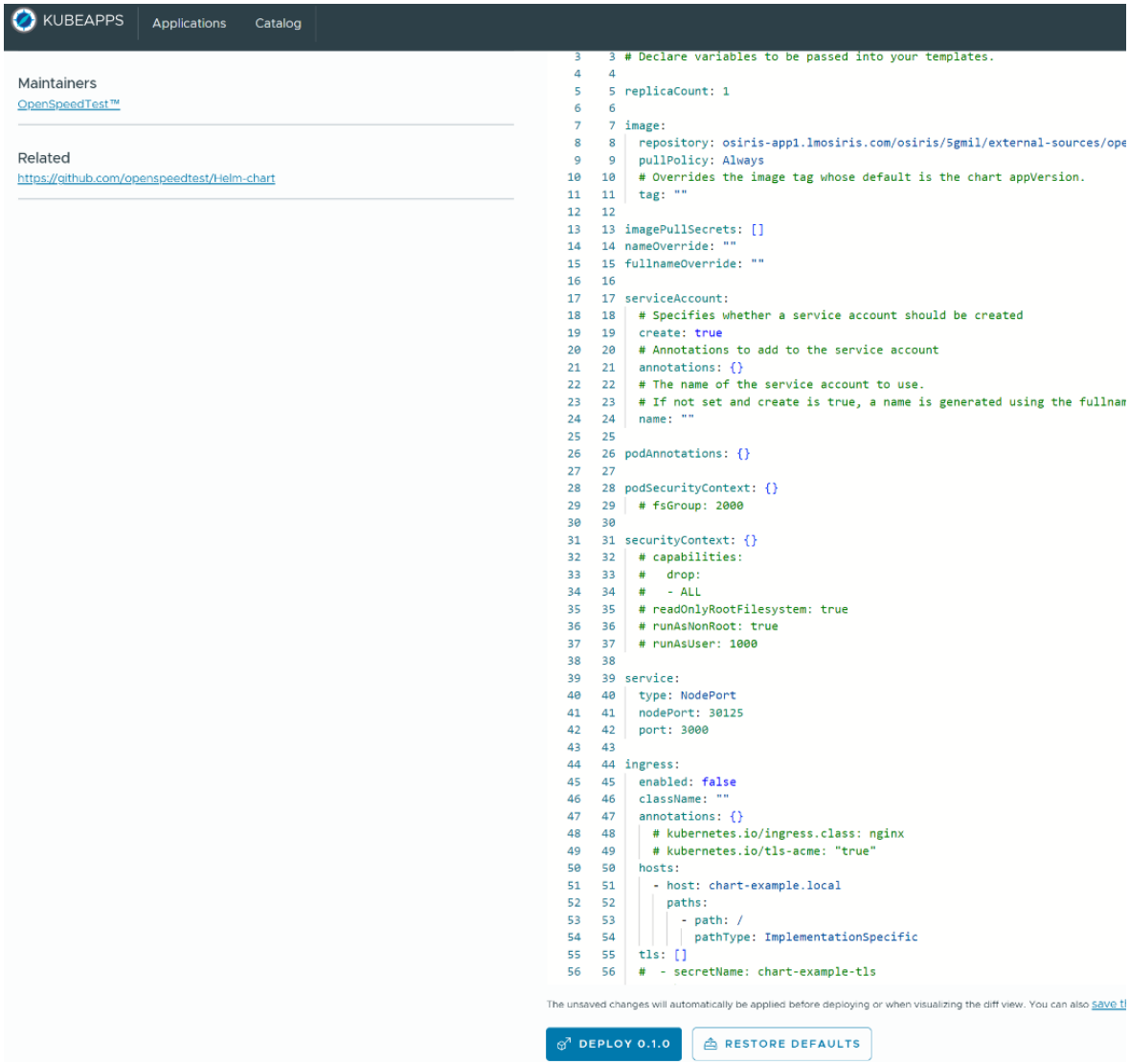

Figure 13.1-10: Kubeapps Application Deployment Page 2

The application should be successfully deployed, and the screen should change and show the deployment was successful. Under from this screen the status of the application can be seen and the option to delete the application available at the top right of the screen:

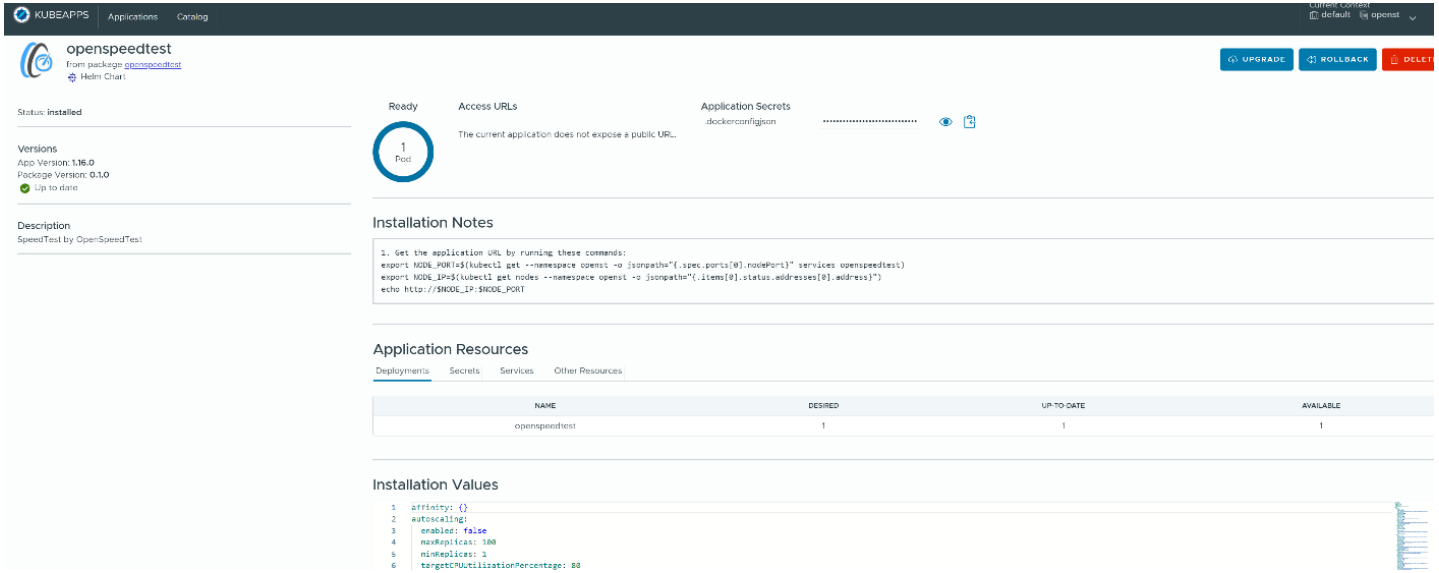

Figure 13.1-11: Kubeapps Application Deployed Page

To verify that the application is running correctly the user can check the main screen of Kubeapps and see if the application has been populated there or the k3s cluster can be accessed and the user can check to see the pod has been spawned and is running:

To check on Kubeapps, click the “Applications” button on the top left of the screen and toggle on the all-namespace button, search the apps listed to see if it has populated, in this example Openspeedtest has populated:

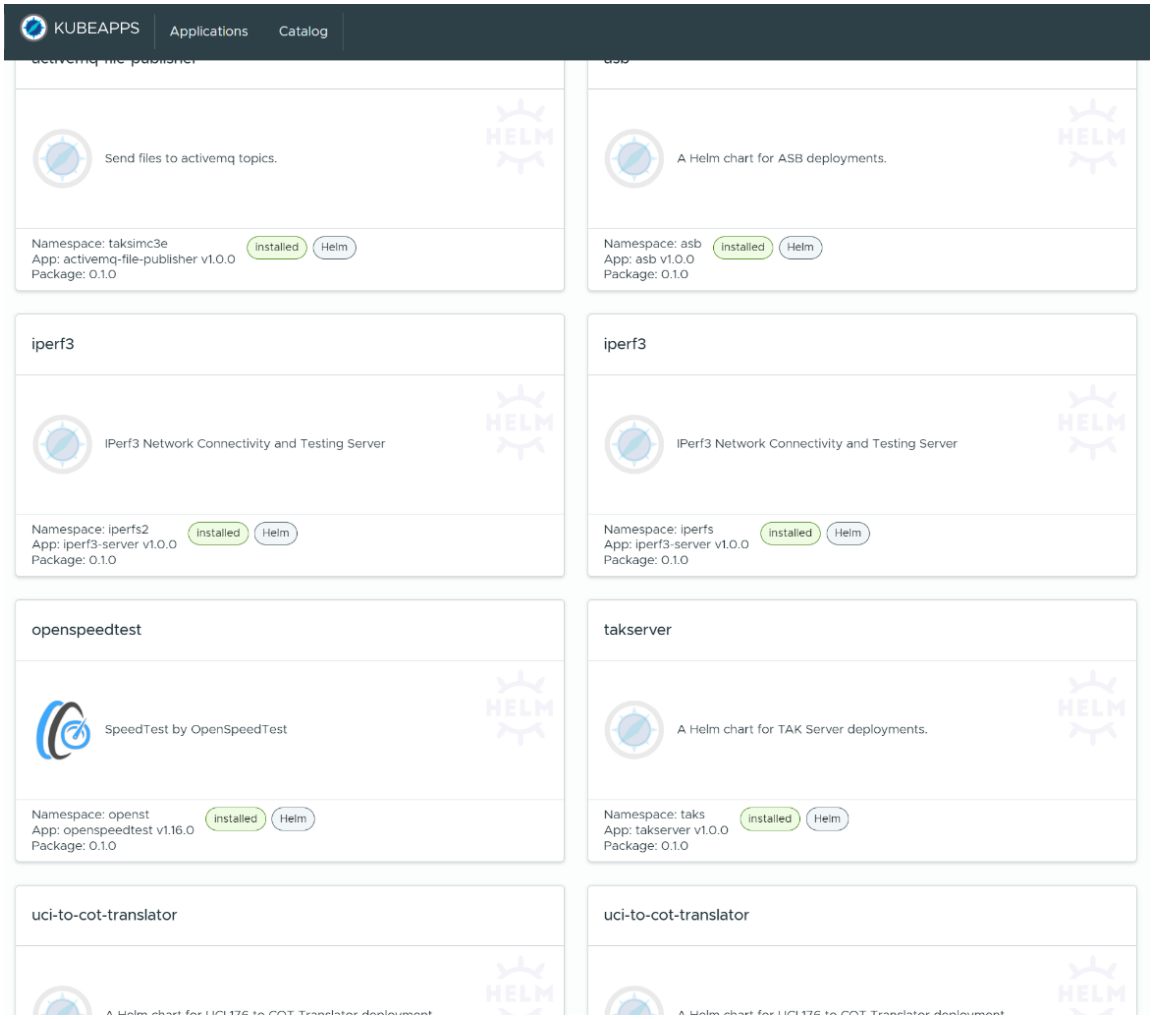

Figure 13.1-12: Kubeapps Application Main Page

To check the k3s cluster access the app-node-1 and execute the get all services command as shown below and look for the namespace and app name of your application:

```
root@appnode1:/home/nhansell# kubectl get svc -A
NAMESPACE              NAME                              TYPE           CLUSTER-IP      EXTERNAL-IP            PORT(S)
default                kubernetes                        ClusterIP      10.43.0.1       <none>                 443/TCP
kube-system            kube-dns                          ClusterIP      10.43.0.10      <none>                 53/UDP,53/TCP,9153/TCP
kube-system            metrics-server                    ClusterIP      10.43.23.78     <none>                 443/TCP
harbors                harbor-core                       ClusterIP      10.43.248.51    <none>                 80/TCP
harbors                harbor-database                   ClusterIP      10.43.73.73     <none>                 5432/TCP
harbors                harbor-jobservice                 ClusterIP      10.43.185.74    <none>                 80/TCP
harbors                harbor-portal                     ClusterIP      10.43.163.174   <none>                 80/TCP
harbors                harbor-redis                      ClusterIP      10.43.224.49    <none>                 6379/TCP
harbors                harbor-registry                   NodePort       10.43.57.149    <none>                 5000:30583/TCP,8080:30966/TCP
harbors                harbor-trivy                      ClusterIP      10.43.169.182   <none>                 8080/TCP
iperfs                 iperf3-server-udp                 ClusterIP      10.43.137.36    10.10.20.2             30030/UDP
iperfs                 iperf3-server-tcp                 NodePort       10.43.108.234   10.10.20.2             5201:30030/TCP
iperfs2                iperf3-server-udp                 ClusterIP      10.43.121.2     10.10.20.2             30040/UDP
iperfs2                iperf3-server-tcp                 NodePort       10.43.108.222   10.10.20.2             5201:30040/TCP
kubeapps               kubeapps-postgresql-hl            ClusterIP      None            <none>                 5432/TCP
kubeapps               kubeapps-internal-kubeappsapis    ClusterIP      10.43.240.236   <none>                 8080/TCP
kubeapps               kubeapps-postgresql               ClusterIP      10.43.176.127   <none>                 5432/TCP
kubeapps               kubeapps-internal-dashboard       ClusterIP      10.43.85.123    <none>                 8080/TCP
kubeapps               kubeapps                          NodePort       10.43.179.224   <none>                 80:30250/TCP
kubeapps               chart-museum                      NodePort       10.43.48.6      <none>                 8080:30230/TCP
kubernetes-dashboard   dashboard-metrics-scraper         ClusterIP      10.43.23.4      <none>                 8000/TCP
kubernetes-dashboard   kubernetes-dashboard              NodePort       10.43.34.30     <none>                 443:30080/TCP
taks                   tak-database                      NodePort       10.43.33.113    <none>                 5432:30260/TCP
taks                   takserver                         NodePort       10.43.87.252    <none>                 8080:30904/TCP,8089:30014/TCP,8
asb                    asb                               ClusterIP      10.43.202.225   <none>                 8161/TCP,61616/TCP
asb                    asb-73                            NodePort       10.43.155.143   <none>                 8161:31678/TCP,61616:30010/TCP
kube-system            traefik                           LoadBalancer   10.43.165.1     10.10.20.2,10.10.20.3  80:30191/TCP,443:32618/TCP
openst                 openspeedtest                     NodePort       10.43.137.246   <none>                 3000:30125/TCP
root@appnode1:/home/nhansell#
```

Figure 13.1-13: Data Network ‘get services’ Command

Check that the pod was created and running for the application by running the get all pods command, the grep command can be used to search the output of all pods command narrowing the output to specific namespaces or app names as shown below. Search the output of the command to verify the pod hosting your application was created.

```
root@appnode1:/home/nhansell# kubectl get pods -A  | grep openst
openst                 openspeedtest-7cbff95f5b-ds8q9                    1/1     Running     0              2m47s
root@appnode1:/home/nhansell#
```

Figure 13.1-14: Data Network 'get pods' command

14 DATA NETWORK APPLICATION: BIND9 DOMAIN NAME SERVER (DNS)

BIND is an open source DNS software system including an authoritative server, a recursive resolver and related utilities.

For more information, see https://bind9.net/

14.1 USING BIND9

The Bind9 Server runs as a container on the Master Node. The software exposes a GUI where DNS entries can be added and modified. The function of the Bind9 DNS is to take IP addresses, example 10.10.20.2 and give them a fully qualified domain name (FQDN) (i.e. app-node-1.lmosiris.com) and when that IP address is trying to be accessed either by k3s or a remote user, the DNS name can be used instead of the IP address. More info about Bind9 dns can be found here: https://bind9.readthedocs.io/en/latest/#

The Bind9 home page should load but be blank press “Servers” tab on left hand side to access the OSIRIS DNS:

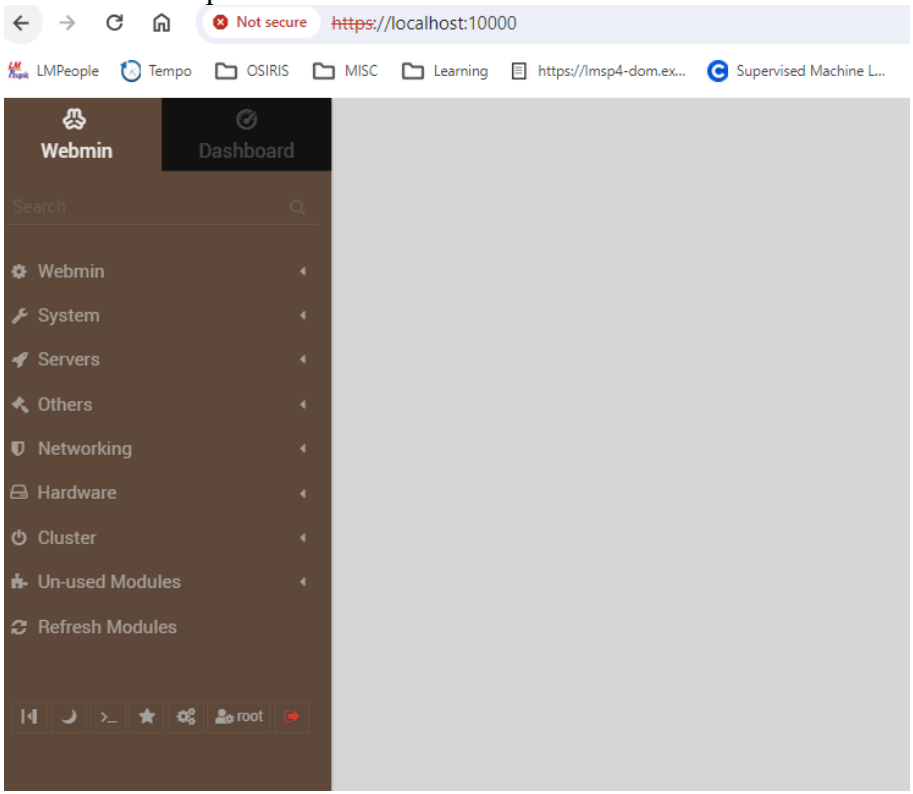

Figure 14.1-1: Bind9 Home Page

Click bind9 dns server:

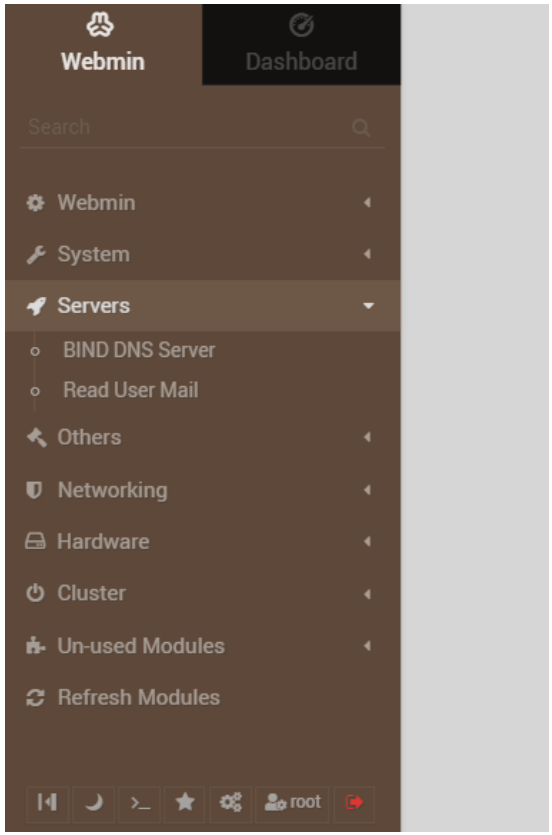

Figure 14.1-2: Bind9 Home Page Side Menu

Create Master Zone:
Select 'Create master zone'
Fill out form data, click 'Create', and then 'Apply configuration' in the top-right corner

Figure 14.1-3: Bind9 Create Master Zone

Select lmosiris.com icon:

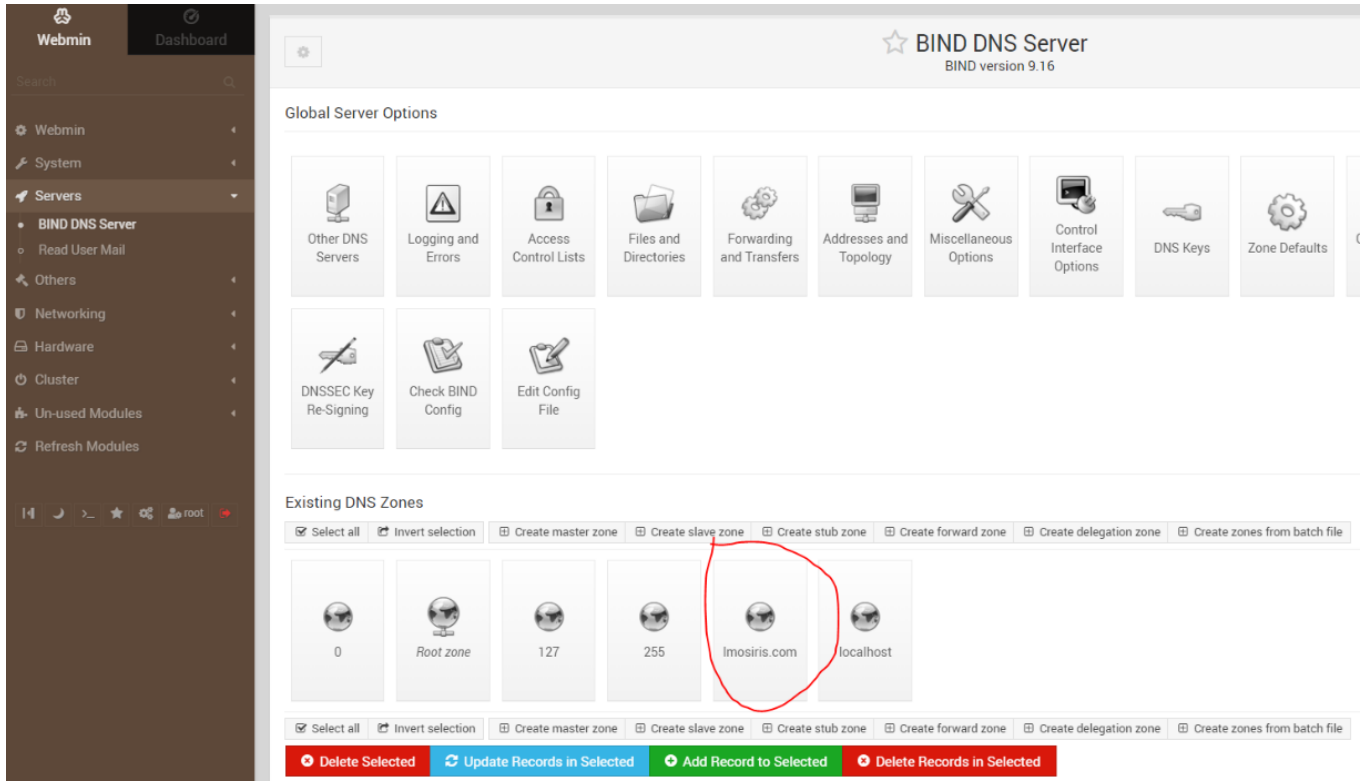

Figure 14.1-4: Bind9 Servers Page

Click addresses:

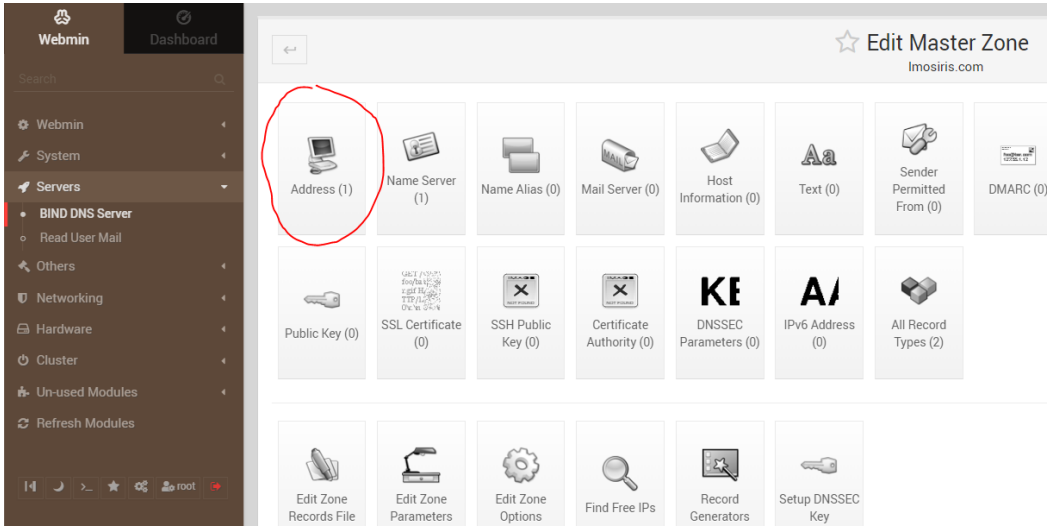

Figure 14.1-5: Bind9 Address Selection

From this page the user can now add, delete and modify DNS from this page:

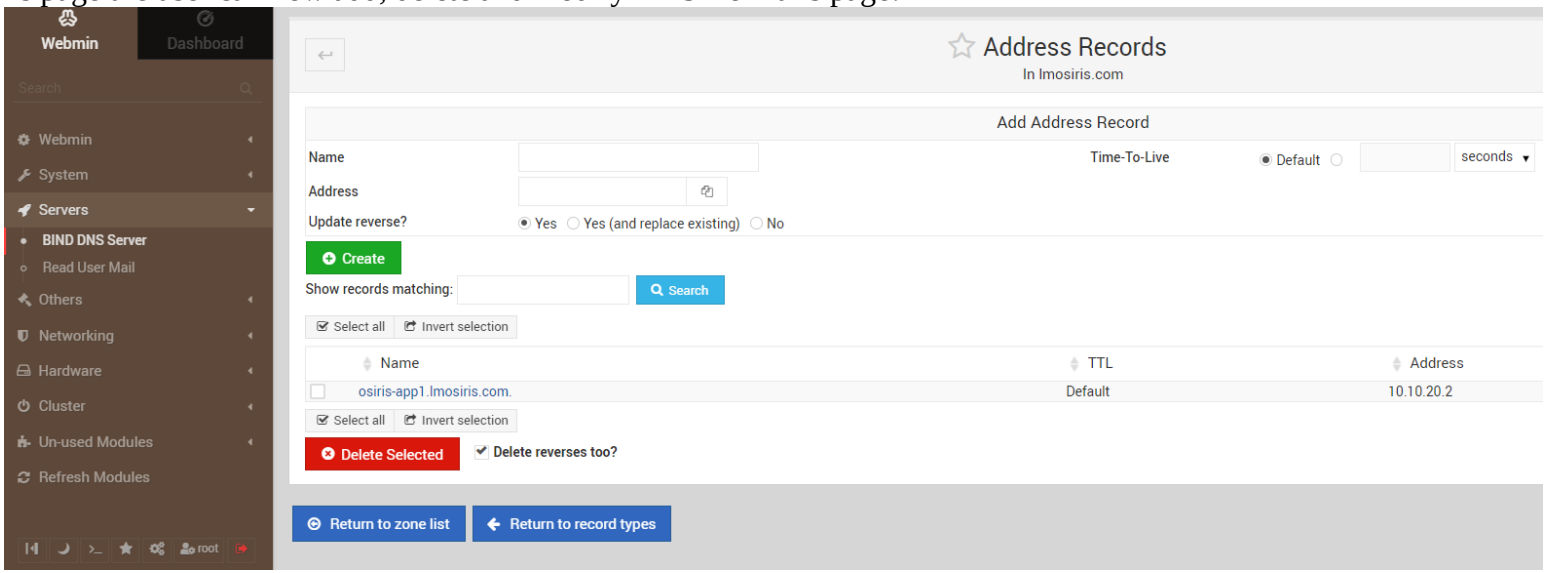

Figure 14.1-6: Bind9 lmosiris.com Page

Disable systemd-resolved and set host/app-node-1 dns to use Bind9
Disable system-resolved
Shows system status

```
$ systemctl status systemd-resolved.service
```

Will stop system-resolved from updating the resolv.conf file which would break kubernetes

```
$ systemctl stop system-resolved
$ systemctl disable system-resolved
```

Configure dns server to host
edit /etc/resolv.conf
set the only entry in the file to "nameserver <HOST-IP>"
In this example <HOST-IP> would be 10.10.20.2

```
#search lmms.lmco.com atc.lmco.com ast.lmco.com
#nameserver 192.132.207.1 #MOST RECENTLY VERIFIED
#nameserver 192.132.207.2
#nameserver 129.197.217.100
nameserver 10.10.20.2
~
```

Figure 14.1-7: Data Network /etc/resolv.conf

15 DATA NETWORK APPLICATION: KUBERNETES DASHBOARD

Dashboard is a web-based Kubernetes user interface. You can use Dashboard troubleshoot your containerized application and manage the cluster resources. You can use Dashboard to get an overview of applications running on your cluster, as well as for creating or modifying individual Kubernetes resources (such as Deployments, Jobs, DaemonSets, etc).

For more information – see https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/

15.1 USING THE KUBERNETES DASHBOARD

Refer to Table 12-1 for GUI Access.
To login into Dashboard go to app-node-1 and generate a token:
Token generation:

```
root@app-node-1:/home/nhansell# kubectl -n kubernetes-dashboard create token admin-user
eyJhbGciOiJSUzI1NiIsImtpZCI6InEwTVpfQ043ZVBncGVidlUzUTR5SU5yNVNybTFxOVprS0ZOZmpfNzdtUUUif
kzMjAzNjQsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJ
iNTg1YWMwNTYtNmE4Zi00ODA1LWIxOGMtNGJjYzZiMWJiODVkIn19LCJuYmYiOjE3MTkzMjAzNjQsInN1YiI6InN5
h9ZxUTEWj-2HgDLdOy5_SO4hNgYPBpwpNxpPueJq4cjaEBEjF4bc7B-Z0MKXXIj61dUfJaSefOCV1n4P5QsxegLKI
LvBr8iuDzDI4ZPfWDwvpSAcqjqby8rTyRqRUkZmV6oxxZQVOJmOCjIJV3WodTdW3nR2esS10I4gILMePUjzBQ
```

Figure 15.1-1: Kubernetes Dashboard Token

Go back to browser where Dashboard Gui is open, enter in token and sign in:

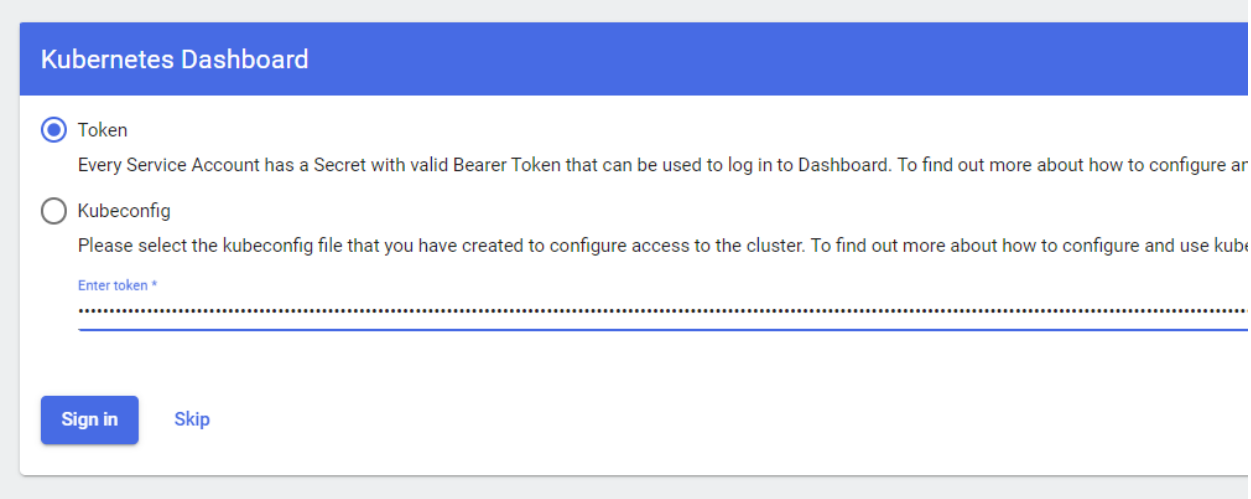

Figure 15.1-2: Kubernetes Dashboard Login w/ Token

Once successfully logged in change the namespaces to “All namespaces” in the top right, view image below:

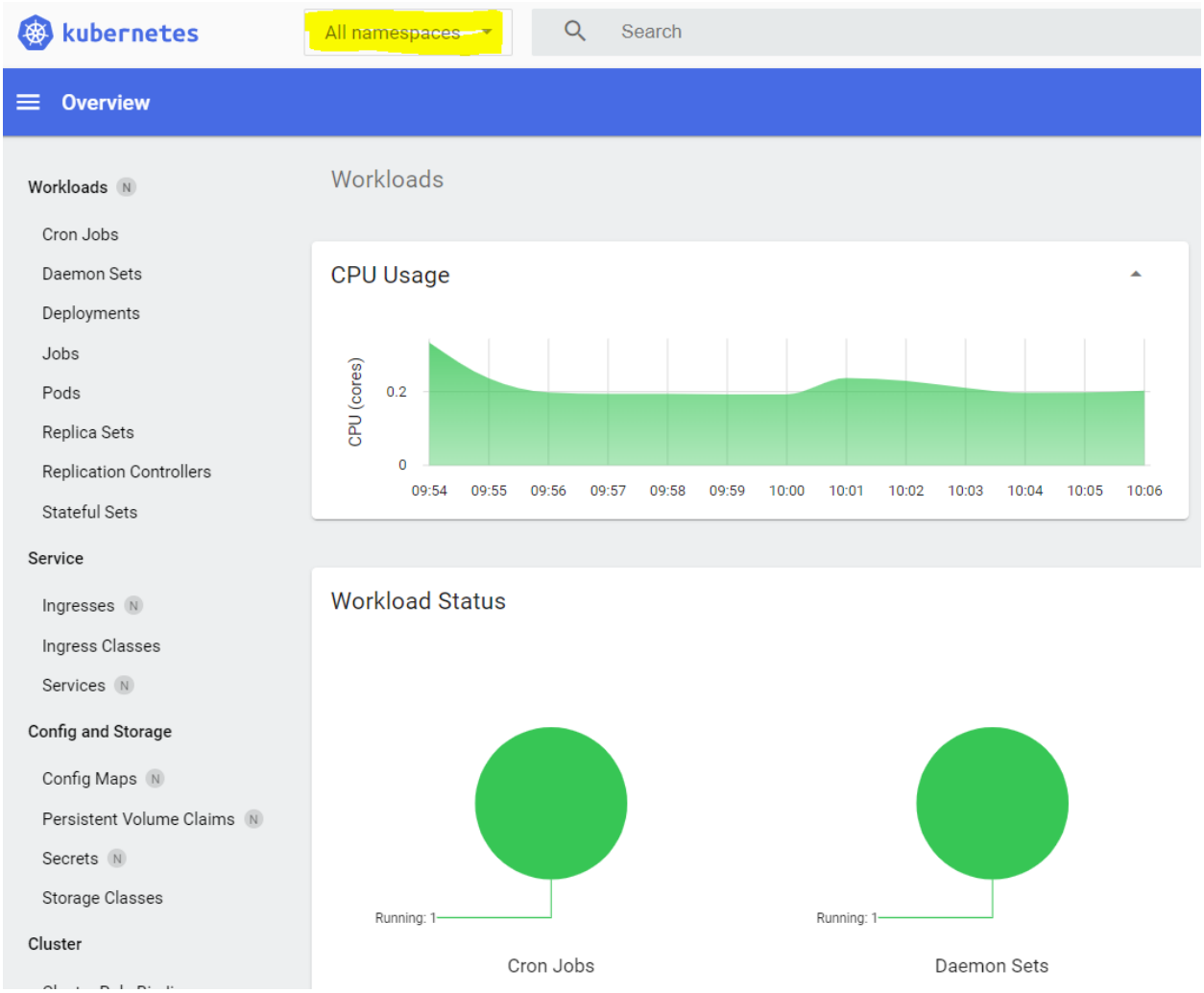

Figure 15.1-3: Kubernetes Dashboard Main Page

16 DATA NETWORK APPLICATION: OPENSPEEDTEST

Openspeedtest is a tool used to test network performance. A test is started but clicking the "Start" button and Openspeedtest will attempt to max out the UL and DL of the network. Openspeedtest can be accessed from a UE(smartphone) or from computer connected to the network. Refer to Table 12-1 for GUI Access to access Openspeedtest.

For more information – see https://openspeedtest.com/

16.1 USING OPENSPEEDTEST

Homepage:

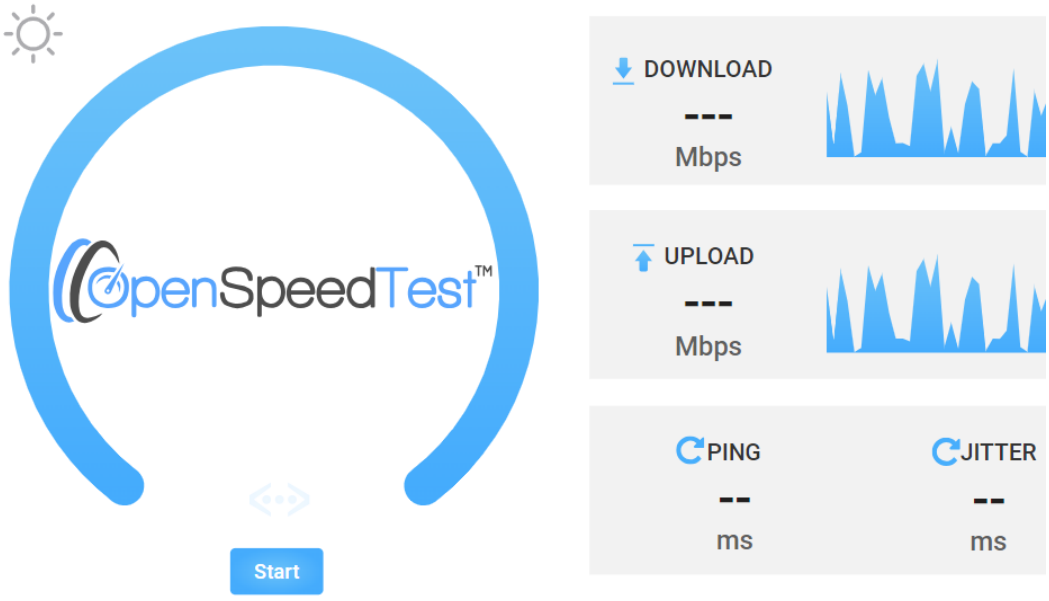

Figure 16.1-1: Openspeedtest Home Page

Example test:

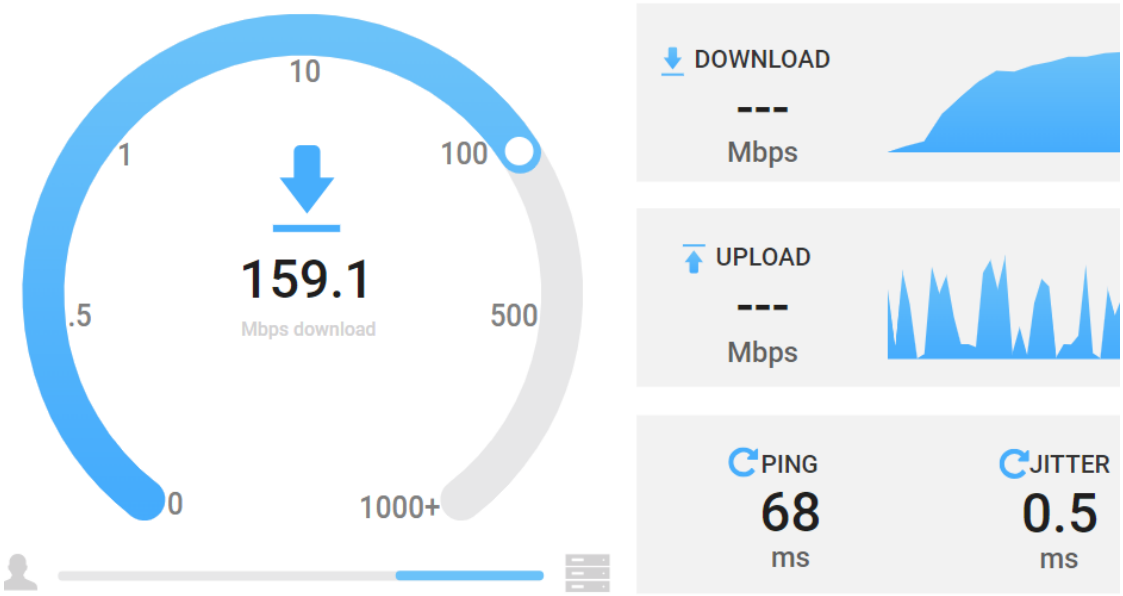

Figure 16.1-2: Openspeedtest Running Test

Example test result:

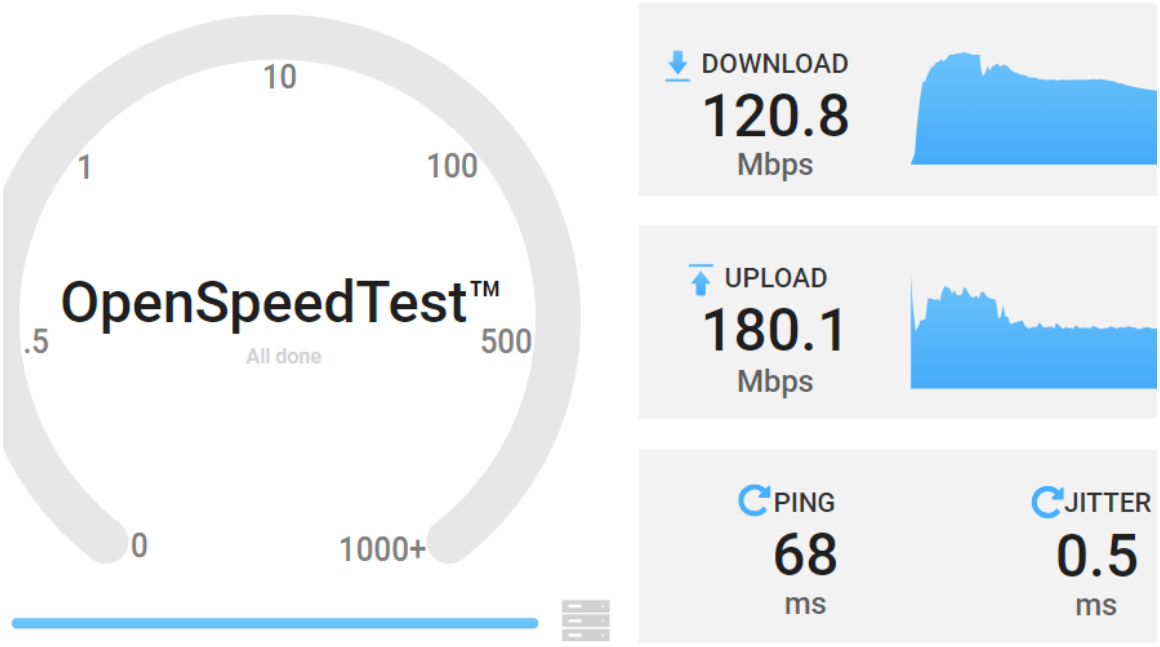

Figure 16.1-3: Openspeedtest Completed Test

17 DATA NETWORK APPLICATION: TAK SERVER

The Team Awareness Kit for Android was originally developed by the Air Force Research Laboratory (AFRL) and is now maintained by a Joint Product Center. ATAK is an Android smartphone geospatial infrastructure and situational awareness app. It allows for precision targeting, surrounding land formation intelligence, situational awareness, navigation, and data sharing. All the Android variants of TAK are virtually identical and all are interoperable with each other and with other TAK products.

For more information – see https://tak.gov/

Refer to Table 12-1 to access TAK Server (use https://)

17.1 USING TAK SERVER

TAK server login:

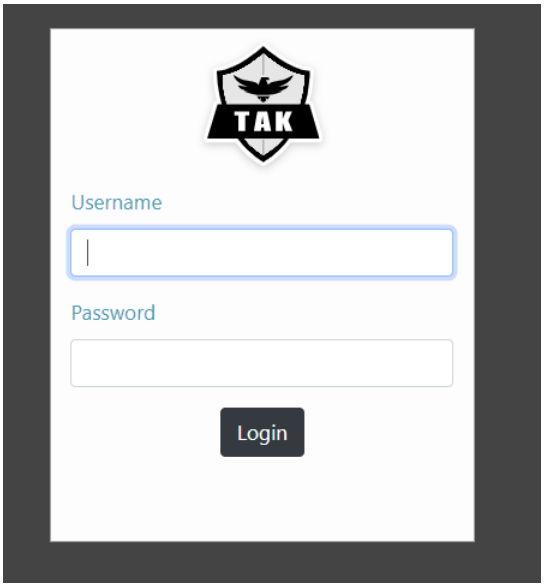

Figure 17.1-1: Tak Server Login Page

Login Information
Two users exist currently:
An Admin user, that has access to the admin console and to make configuration changes to TAK Server
User: superuser
Password: UnnecessarilyLongPassword123!
A Normal user, that has access to log in and view Webtak, as well as make SSL Connections with TAK Clients.
User: nmxma
Password: 5GNetModXMissionApps123!
Home Page:

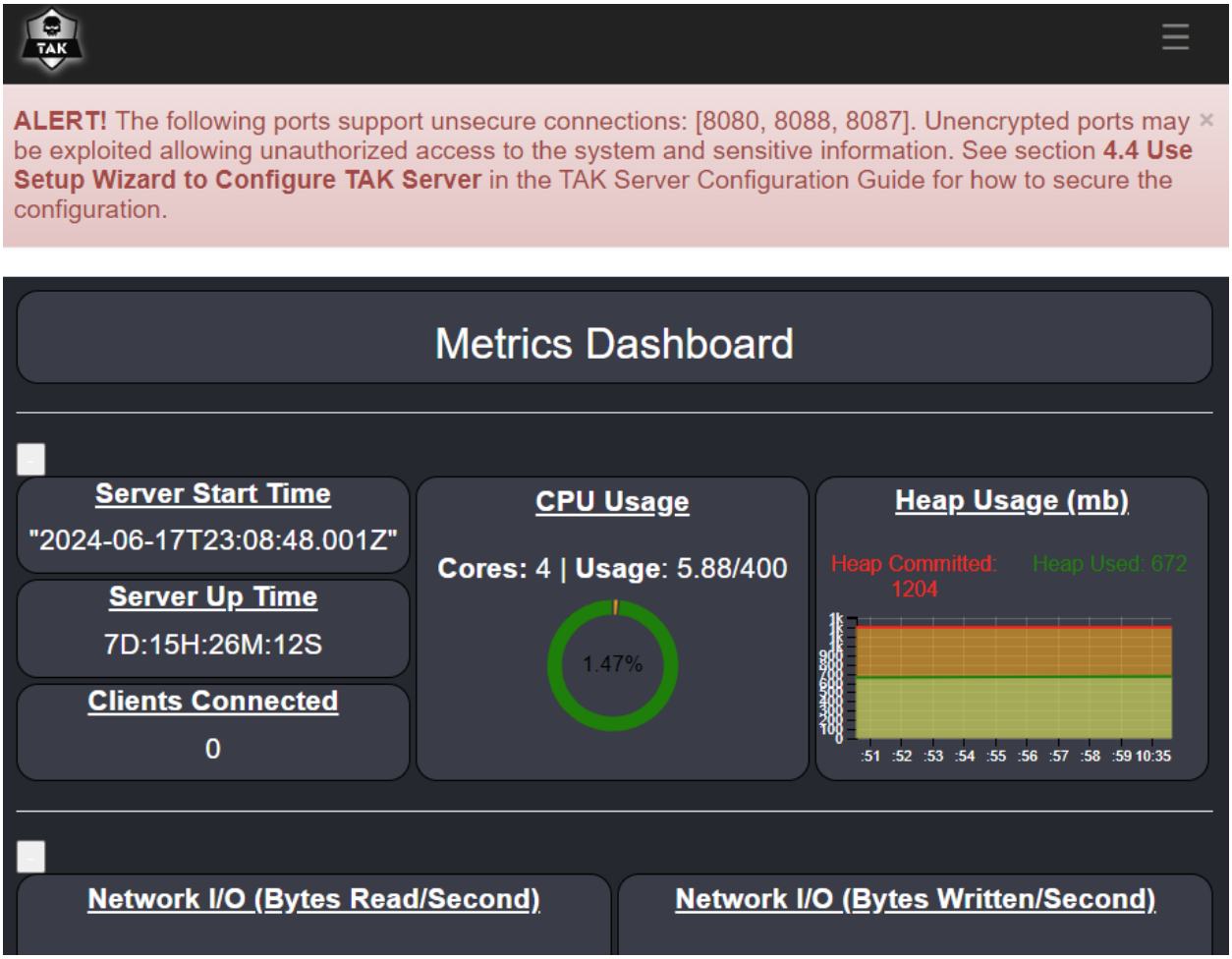

Figure 17.1-2: Tak Server Home Page

WebTAK allows a user to access the situational awareness view through the browser. To access WebTAK directly, log in using the nmxma user which does not have permissions to access the management dashboard.

ATAK and WinTAK can also be used to access the situational awareness view. To make a connection to the TAK Server, Add a Server Connection using either of those clients:

Figure 17.1-3: Tak Server Connection Page

For more information – including how to use the provided certificates to connect using SSL – see the TAK Server README.md at /root/kubernetes-apps/standalone-TAKSERVER-4-9-23-withtranslators/takserver/README.md on the Master Node or see the standalone TAK Server document.

18 DATA NETWORK APPLICATION: IPERF

iPerf, is a tool for network performance measurement and tuning. There are two iPerf servers running on the Data Network. iPerf does not have a GUI like other apps, instead the server address is used when running a test, for example with a UE such as S23 Samsung phone:

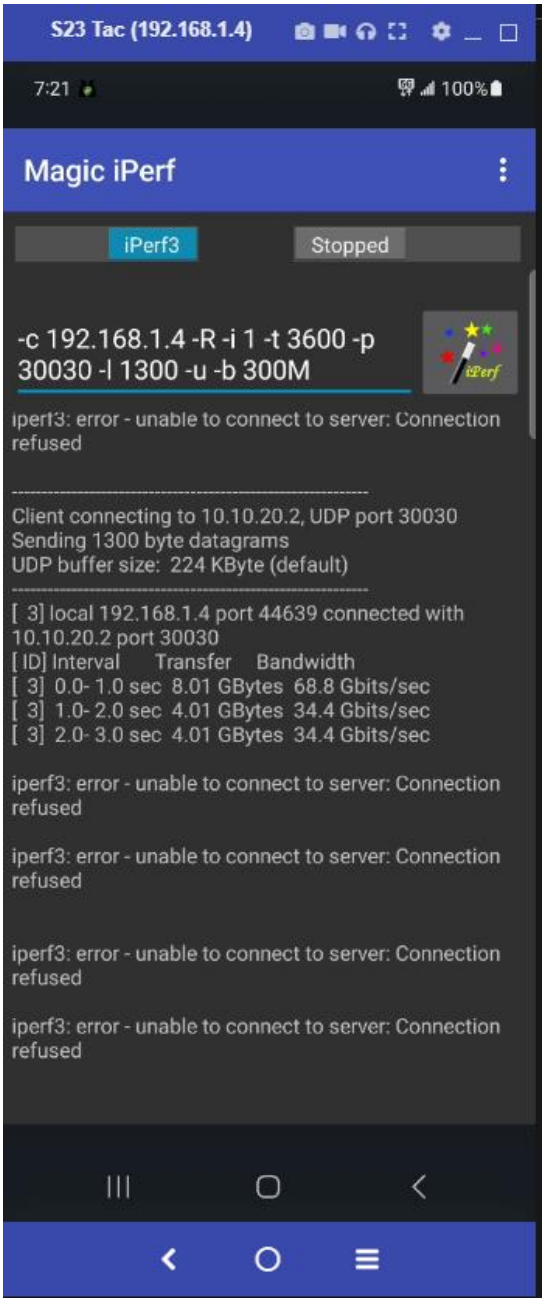

Figure 18-1: Magic iPerf Test Page

19 DATA NETWORK APPLICATION: HARBOR

Harbor is an open source registry that secures artifacts with policies and role-based access control, ensures images are scanned and free from vulnerabilities, and signs images as trusted. Harbor, a CNCF Graduated project, delivers compliance, performance, and interoperability to help you consistently and securely manage artifacts across cloud native compute platforms like Kubernetes and Docker. Harbor serves as the Image Repository for Applications installed to the Data Network.

For more information – see https://goharbor.io/

19.1 USING HARBOR

Harbor is available using http://osiris-app1.lmosiris.com

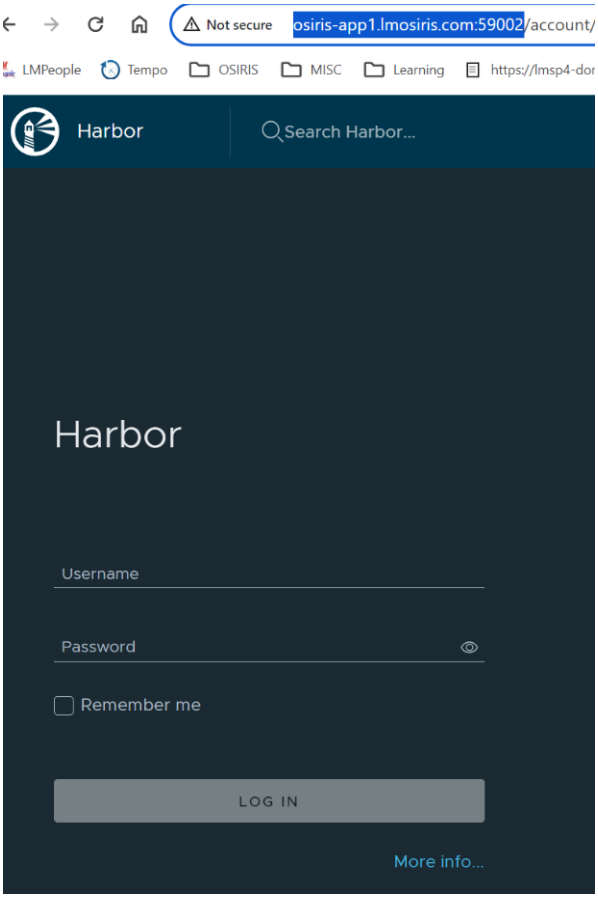

Figure 19.1-1: Harbor Login Page

Enter UN: admin PW:Harbor12345 click login, if entered successfully it should load the Harbor GUI

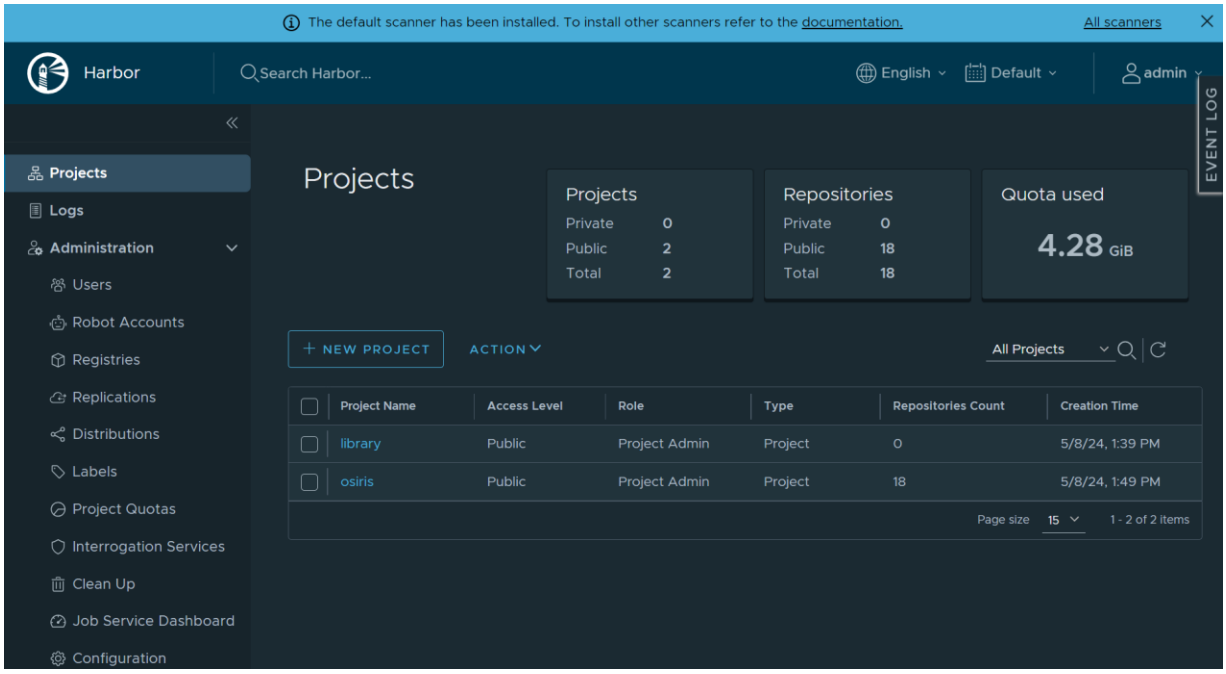

Figure 19.1-2: Harbor Projects Main Page

To create a new project click the NEW PROJECT button, give your project a name and click OK:

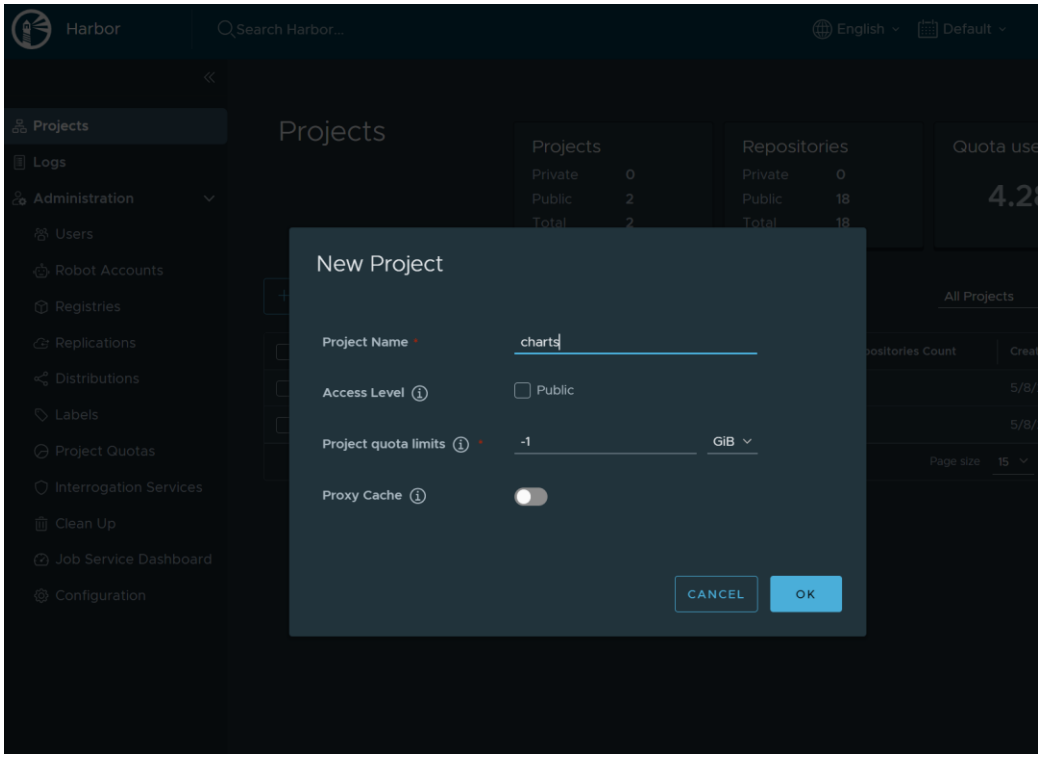

Figure 19.1-3: Harbor New Project Page

A green success banner should appear at the top and the new project should be populated in the list in the middle:

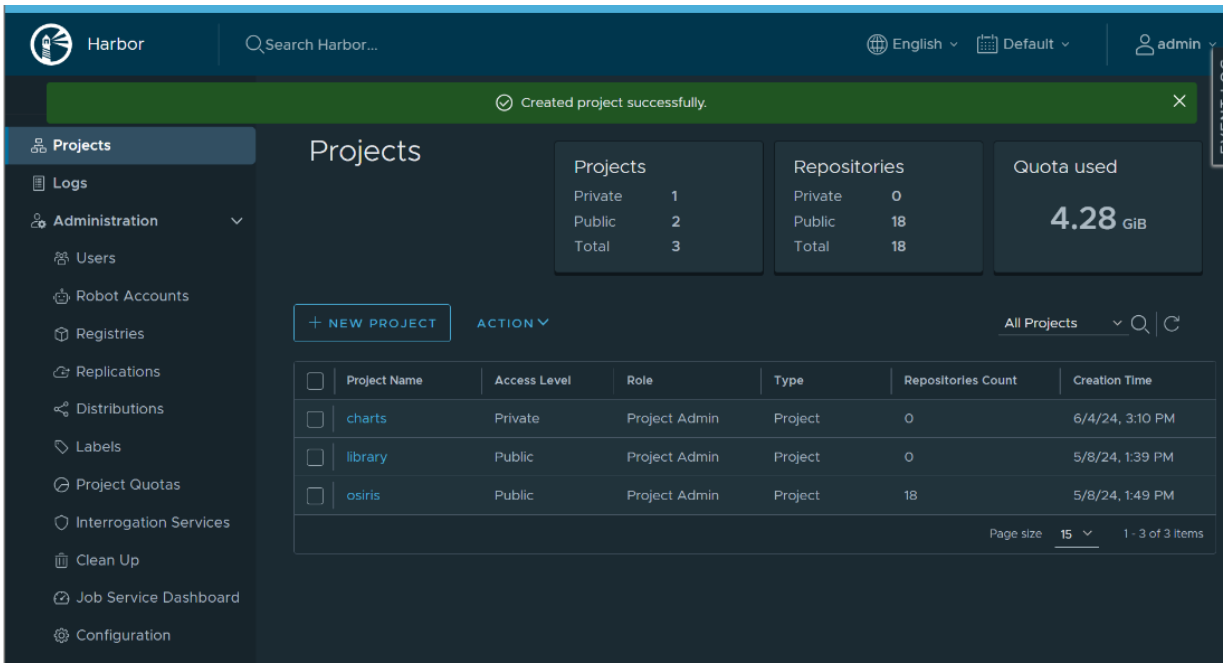

Figure 19.1-4: Harbor Project Creation Successful

Log into Harbor with Docker and push an image
To test the harbor registry, log into Harbor using Docker
$ Docker login osiris-app1.lmosiris.com
Creds: admin / Harbor12345
From your list of images – pick any image that you have locally:
$ Docker images

```
root@appnode1:~/kubernetes-apps# docker images
REPOSITORY                                                                                                                                TAG                 IMAGE ID       CREATED         SIZE
osiris-app1.lmosiris.com/osiris/5gmil/mission-apps/site/container-images/activemq-file-publisher/activemq-file-publisher                 msnapps-develop     ae87eecbef0b   2 weeks ago     483MB
registry.gitlab.us.lmco.com:443/5gmil/mission-apps/site/container-images/activemq-file-publisher/activemq-file-publisher                 msnapps-develop     ae87eecbef0b   2 weeks ago     483MB
osiris-app1.lmosiris.com/osiris/5gmil/external-sources/bitnami/kubeapps-dashboard                                                         2.10.0-debian-12-r3 c385158fa937   2 weeks ago     230MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/bitnami/kubeapps-dashboard                                                         2.10.0-debian-12-r3 c385158fa937   2 weeks ago     230MB
osiris-app1.lmosiris.com/osiris/5gmil/external-sources/bitnami/kubeapps-asset-syncer                                                      2.10.0-debian-12-r3 5a7ea2905b73   2 weeks ago     41.2MB
bitnami/kubeapps-asset-syncer                                                                                                             2.10.0-debian-12-r3 5a7ea2905b73   2 weeks ago     41.2MB
osiris-app1.lmosiris.com/osiris/5gmil/external-sources/bitnami/kubeapps-apprepository-controller                                          2.10.0-debian-12-r3 d6b52bb85730   2 weeks ago     43.6MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/bitnami/kubeapps-apprepository-controller                                          2.10.0-debian-12-r3 d6b52bb85730   2 weeks ago     43.6MB
osiris-app1.lmosiris.com/osiris/5gmil/external-sources/bitnami/kubeapps-apis                                                              2.10.0-debian-12-r4 8777dbbba225   2 weeks ago     481MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/bitnami/kubeapps-apis                                                              2.10.0-debian-12-r4 8777dbbba225   2 weeks ago     481MB
osiris-app1.lmosiris.com/osiris/5gmil/external-sources/bitnami/nginx                                                                      1.26.0-debian-12-r0 40acfaabbf77   2 weeks ago     184MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/bitnami/nginx                                                                      1.26.0-debian-12-r0 40acfaabbf77   2 weeks ago     184MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/bitnami/postgresql                                                                 16.3.0-debian-12-r4 93dde1ce4ed0   2 weeks ago     334MB
osiris-app1.lmosiris.com/osiris/5gmil/external-sources/bitnami/postgresql                                                                 16.3.0-debian-12-r4 93dde1ce4ed0   2 weeks ago     334MB
10.10.20.2:5000/5gmil/external-sources/goharbor/redis-photon                                                                              v2.10.0             5083850c5206   5 months ago    165MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/goharbor/redis-photon                                                              v2.10.0             5083850c5206   5 months ago    165MB
10.10.20.2:5000/5gmil/external-sources/goharbor/trivy-adapter-photon                                                                      v2.10.0             5873942a56be   5 months ago    478MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/goharbor/trivy-adapter-photon                                                      v2.10.0             5873942a56be   5 months ago    478MB
10.10.20.2:5000/5gmil/external-sources/goharbor/harbor-registryctl                                                                        v2.10.0             7a3b7d7d972c   5 months ago    149MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/goharbor/harbor-registryctl                                                        v2.10.0             7a3b7d7d972c   5 months ago    149MB
10.10.20.2:5000/5gmil/external-sources/goharbor/registry-photon                                                                           v2.10.0             9baecb934ded   5 months ago    83.4MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/goharbor/registry-photon                                                           v2.10.0             9baecb934ded   5 months ago    83.4MB
10.10.20.2:5000/5gmil/external-sources/goharbor/harbor-jobservice                                                                         v2.10.0             4960b98063d3   5 months ago    140MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/goharbor/harbor-jobservice                                                         v2.10.0             4960b98063d3   5 months ago    140MB
10.10.20.2:5000/5gmil/external-sources/goharbor/harbor-core                                                                               v2.10.0             00c9a2f5729c   5 months ago    168MB
registry.gitlab.us.lmco.com:443/5gmil/external-sources/goharbor/harbor-core                                                               v2.10.0             00c9a2f5729c   5 months ago    168MB
```

Figure 19.1-5: Docker Images Command

Retag and push the image to the Harbor registry: (example below)
$ docker tag registry.gitlab.us.lmco.com:443/5gmil/external-sources/iperf osiris-app1.lmosiris.com/osiris/5gmil/external-sources/iperf3
$ Docker push osiris-app1.lmosiris.com/osiris/5gmil/external-sources/iperf3

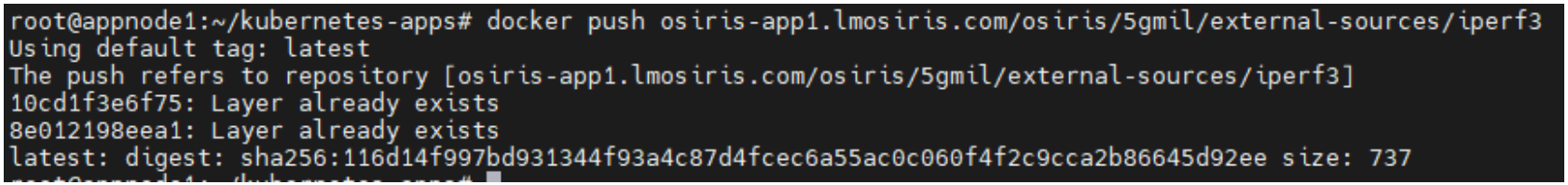

Figure 19.1-6: Docker Push Command

20 VENDOR MANUALS AND OPERATORS GUIDES

Table 20-1 lists the Vendor Manuals available for Phase 2.

Table 20-1: Software Vendor Manuals

Software		Description	Vendor Manual
5GCN	5G Core	Admin Guides	Trillium_5G_NR_Kubernetes_Platform_Installation_Guide_R4.2.0_v2.pdf Trillium_5GCN_EMS_User_Guide_R5.0.0_v1.pdf Trillium_5GCN_Quick_Installation_Guide_R5.0.0_v1.pdf Trillium_5GCN_Solution_CNF_User_Guide_R5.0.0_v1.pdf Trillium_5GCN_Solution_Description_Guide_R5.0.0_v1.pdf Trillium_5GCN_Solution_OAM_User_Guide_R5.0.0_v1.pdf Trillium_5GCN_Solution_Release_Notes_R5.0.0_v0.1.pdf
gNodeB	Radisys Centralized Unit (CU)	Trillium 5G NR CU Guides & Specs, Config. Parameters and Handover calls	Handover_Calls_Flow.zip PICS-5GRAN_4.0.7.xlsx Radisys_5G_NR_Feature_Matrix_Q4_2023.xlsx Trillium_5G_NR_gNB_CU_Solution_Description_Guide_R4.0.1_v1.pdf Trillium_5G_NR_gNB_CU_Solution_x86_User_Guide_R4.0.7_v1.pdf Trillium_5G_NR_gNB_CU_Solution_YANG_Configuration_Parameter_Reference_R4.0.0_v1.xlsx Trillium_5G_NR_gNB_Solution_FlexRAN_OTA_Release_Note_R4.0.7_v1.pdf Trillium_5G_NR_gNB_Solution_Functional_Specification_R4.0.0_v1.pdf Trillium_5G_NR_gNB_Solution_OAM_Guide_R4.0.1_v0.1.pdf
	Radisys Distributed Unit (DU)	Trillium 5G NR DU Guides & Specs, Config. Parameters and Handover calls	Handover_Calls_Flow.zip PICS-5GRAN_4.0.7.xlsx Radisys_5G_NR_Feature_Matrix_Q4_2023.xlsx

Software		Description	Vendor Manual
			Trillium_5G_NR_gNB_DU_Solution_Description_Guide_R4.0.1_v1.pdf Trillium_5G_NR_gNB_DU_Solution_x86_User_Guide_R4.0.7_v1.pdf Trillium_5G_NR_gNB_DU_Solution_YANG_Configuration_Parameter_Reference_R4.0.0_v1.xlsx Trillium_5G_NR_gNB_Solution_FlexRAN_OTA_Release_Note_R4.0.7_v1.pdf Trillium_5G_NR_gNB_Solution_Functional_Specification_R4.0.0_v1.pdf Trillium_5G_NR_gNB_Solution_OAM_Guide_R4.0.1_v0.1.pdf
	High Physical Layer (FlexRAN)	FlexRAN_22.11	575822-2.11-flexran-ref-sw-release-note-v22.11.pdf 637859_38.211_22_07_Rev0_5.xlsx 637860_38.212_22.11_Rev0_6.xlsx 645964_FlexRAN_rel_announcement_v22_11.pdf 737897_38.213_22_07_Rev0_1.xlsx 737898_38.214_22_07_Rev0_1.xlsx intel5G-Edge-Innovation-Handbook.pdf
	High Physical Layer (FlexRAN)	FlexRAN_22.11 Additional docs	570228-3.0-flexran-reference-solution-l1-user_guide_v20_08.pdf 571741-flexran-refsol-l1-xml-cfg-ug_v22p11.pdf 571742-flexran-reference-solution-l2-l1-api-spec-v22.07.pdf 572002-flexran-sdk-user-guide-and-api-reference-v22.11.pdf 572007-FlexRAN_FWK_API_Doxygen_v21-07.zip 570 572318-FlexRAN_LTE_RefPHY_Doxygen_documentation_v22.11.zip 575822-2.11-flexran-ref-sw-release-note-v22.11.pdf 575891-flexran-and-mobile-edge-compute-mec-platform-setup-guide-rev1-4.pdf 576423-flexran-reference-solution-l2-l1-nfapi-specification.pdf 576898-4.5_flexran-reference-arch-framework-progGdv22.11.pdf

Software		Description	Vendor Manual
			603575-15.0-flexran-ref-solution-5g-nr-l2-l1-api-v22.11.pdf 603576-9.0- flexran-5g-new-radio-ref-solution-l1-v21.03.pdf 603577-FlexRAN_5GNR_RefPHY_Doxygen_documentation_v22.11.zip 603578-3.0-flexran-5g-nr-fpga-user-guide-v19.03.pdf 611268-13.0-xRAN Front Haul-22.11.pdf 737775_FlexRAN_Ref_Sol_Cloud Native Setup InstGd v22.11.pdf FlexRan_performance_report_22.11_update.zip
Radio Unit	Benetel Radios	Support for Open Radio Access Network (O-RAN). Features include: • Four Transmit and Four Receive (4T4R) external antennas • Up to 4 x 5W output power • 100 MHz of 5G bandwidth • Choice of several bands	00104969_Benetel O-RU System Software Upgrade App Note [CUSM Plane V3.4].pdf L1 interface specification doc_v8.0.pdf Benetel_CAT-A_O-RU_V1.0.2 Release Note.pdf RAN650_Install_and_Bring_up_Guide_Rev2.7.pdf Software User Guide for RANx50 CAT-A O-RUs (Dec 2023).pdf

ACRONYMS

The list of Acronyms is located at

- Baseline Data Library