DRAFT

Cloud Native 5G Super Blueprint

User Guide

**Table of Content**

[1.](#_heading=h.3znysh7) INTRODUCTION 3

[1.1 Purpose of the Document 3](#_heading=h.2et92p0)

[2.](#_heading=h.tyjcwt) ABBREVATIONS 3

[3.](#_heading=h.3dy6vkm) Cloud Native 5G Blueprint Network Architecture 4

[3.1 Architecture Diagram 4](#_heading=h.1t3h5sf)

[3.2 Architecture Explanation 4](#_heading=h.4d34og8)

[3.3 Work Flow 5](#_heading=h.2s8eyo1)

[4.](#_heading=h.17dp8vu) BUILD and Integration COMPONENTs 5

[4.1 What is ONAP and How it is Build? 5](#_heading=h.3rdcrjn)

[4.2 What is EMCO? 6](#_heading=h.26in1rg)

[4.3 What is Redhat Openshift and how it is deployed? 6](#_heading=h.lnxbz9)

[5.](#_heading=h.35nkun2) Integration COMPONENTs (Steps for Setting up the 5G Cloud Native Network) 6

[5.1 EMCO Installation Steps through Helm 7](#_heading=h.1ksv4uv)

[5.1.1 Deploy EMCO Modules 7](#_heading=h.44sinio)

[5.1.2 Deploy EMCO UI Modules 11](#_heading=h.2jxsxqh)

[5.2 Deploy OpenShift cluster 11](#_heading=h.z337ya)

[5.3 ALTRAN 5G Core preparation 11](#_heading=h.3j2qqm3)

[5.4 Orchestration of 5G Core 12](#_heading=h.1y810tw)

[5.4.1 Step by step Procedure for Altran 5G Core Orchestration 12](#_heading=h.4i7ojhp)

[5.4.2 Go to ONAP GUI to do Network Slicing. 21](#_heading=h.2xcytpi)

[5.5 Deploy gNB simulator 24](#_heading=h.1ci93xb)

[5.6 Testing 24](#_heading=h.3whwml4)

[5.6.1 Connection Request Test 25](#_heading=h.2bn6wsx)

[5.6.2 UE Authentication Test 26](#_heading=h.qsh70q)

[5.7 Steps to add EMCO UI Cluster and Controller Registration Flow 27](#_heading=h.3as4poj)

[5.7.1 sOnboard K8s cluster 27](#_heading=h.1pxezwc)

[5.7.2 Register Controllers 28](#_heading=h.49x2ik5)

[6.](#_heading=h.2p2csry) ISSUES ENCOUNTERED 29

[7.](#_heading=h.147n2zr) WORKAROUNDS 29

[8.](#_heading=h.3o7alnk) MITIGATIONS 29

[9.](#_heading=h.23ckvvd) LESSONS LEARNED 29

**PREFACE**

This Document defines the process and the steps performed in the Cloud Native 5G Super Blueprint demo held on October 2021.

Version Information

| **Title** | Cloud Native 5G Super Blueprint Documentation |
| --- | --- |
| **Document Ver. No.** | Draft 0.1 |
| **Approved By** |  |
| **Drafted by** |  |
| **Doc. Classification** |  |

Version History

| **Version #** | **Date** | **Author(s)** | **Approved by** | **Description** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# INTRODUCTION

## Purpose of the Document

This document details the build steps and the integration steps for each Proof-of-Concept component and also includes the issues encountered, workarounds, mitigations, and guidance for furthering the next evolution of the Proof-of-Concept.

# ABBREVATIONS

| **#** | **Abbreviations** | **Expansions** |
| --- | --- | --- |
| 1 | ONAP | Open Network Automation Platform |
| 2 | EMCO | Edge Multi Cluster Orchestrator |
| 3 | CNF | Cloud native Network Function |
| 4 | gNB | gNodeB |
| 5 | NGFW | Next-generation firewall |
| 6 | K8s | Kubernetes |
| 7 | UNH | University of New Hamster |
| 8 | NSSMF | Network Slice Subnet Management Function |
| 9 | RAN | Radio Access Network |
| 10 | UPF |  |

# Cloud Native 5G Blueprint Network Architecture

1. Architecture Diagram

Graphical user interface, application

Description automatically generated

1. Architecture Explanation

* In the above architecture diagram, the orchestrator components are ONAP and EMCO from LFN and these components run in the Kubernetes cloud in UNH (University of New Hamster) lab.
* From UNH lab it is connected to the second lab in Montreal over a SDWAN.
* The SDWAN components comes from Turnium.
* In the Montreal lab, the cloud layer or the NFVI is a Kubernetes distribution from Redhat Openshift.
* On Openshift there are several network functions, such as;
  + 5G Core from Capgemini engineering and Kaloom (UPF)
  + Nextgen firewall from A10.
  + RAN from three companies, Capgemini engineering, Intel and Genxcomm.
  + An optional bypass Gnb/UE Emulator from Rebaca.

1. Work Flow

Graphical user interface

Description automatically generated

* First user must connect the two data centers using SDWAN.
* Deploy the components that are listed below.
  + Deploy Redhat openshift.
  + Deploy ONAP and EMCO on Kubernetes.
  + Deploy Rebaca
  + Install the physical network functions which is the Genxcomm radio
  + Install the UPF
* After deploying all the above components, perform the below activities.
  + Onboard RAN, CORE and Nextgen Firewall Components (NGFWC)
  + Create network service
  + Orchestrate it on the Redhat Openshift
  + Create the E2E network slice using ONAP.

# BUILD and Integration COMPONENTs

1. What is ONAP and How it is Build?

* ONAP (Open Network Automation Platform) is an open-source, orchestration and automation framework and it is hosted by the Linux Foundation.
* ONAP enables service providers and developers to rapidly automate the instantiation and configuration of physical and virtual network functions and to support complete life cycle management activities.
* By unifying the resources of open-source members, ONAP enables the acceleration of the development of a vibrant ecosystem around a globally shared architecture and the implementation of network automation faster than any product could on its own.
* Click on below link to learn more about building the ONAP instance.
  + [Setting up ONAP](https://docs.onap.org/en/latest/guides/onap-operator/settingup/index.html)
  + [(OOM Quick Start Guide) The instructions to deploy ONAP](https://docs.onap.org/projects/onap-oom/en/latest/oom_quickstart_guide.html#quick-start-label)

1. What is EMCO?

* The Edge Multi-Cluster Orchestrator (EMCO) is a software framework for intent-based deployment of cloud-native applications to a set of Kubernetes clusters, spanning enterprise data centers, multiple cloud service providers and numerous edge locations.
* It is architected to be flexible, modular and highly scalable. It is aimed at various verticals, including telecommunication service providers.
* [Click here](https://gitlab.com/project-emco/core/emco-base/-/blob/main/docs/design/emco-design.md) to learn more about EMCO.

1. What is Redhat Openshift and how it is deployed?

* Red Hat OpenShift is a leading enterprise Kubernetes platform that enables a cloud-like experience wherever it is deployed.
* Whether it is in the cloud, on-premises or at the edge, Red Hat OpenShift gives the ability to choose where you build, deploy, and run applications through a consistent experience.
* Red Hat OpenShift’s full-stack automated operations and self-service provisioning for developers lets teams work together more efficiently to move ideas from development to production.
* [Click here](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html/installing/index) to learn more about Openshift deployment.

# Integration COMPONENTs (Steps for Setting up the 5G Cloud Native Network)

| Requirement | |
| --- | --- |
| Components | **Version** |
| ONAP |  |
| Openshift |  |
| EMCO |  |
| Abot | Release 7.0.0 |
| Altran 5G Core |  |

1. EMCO Installation Steps through Helm

| System Requirement for EMCO | |
| --- | --- |
| Item | **Description** |
| CPUs | 8 |
| RAM | 12GB |
| ROM | 200GB |
| NODE | Single node based Kubernetes cluster |

1. **Deploy EMCO Modules**
2. Clone the ONAP EMCO Repository.
3. For cloning the EMCO base repo follow the below link.

<https://gitlab.com/project-emco/core/emco-base.git>

A screenshot of a computer

Description automatically generated with medium confidence

1. EMCO modules deployment in Kubernetes cluster.
2. List the repo contents using the “**ls”** command.

A screenshot of a computer

Description automatically generated

1. User must set the export flag because the containers are used by the “**make deploy**” command. If the flag is not set, then it throws an error while performing “make Deploy”. Below command is used to set the export flag.

**export EMCODOCKERREPO=${container\_registry\_url}/**

A screenshot of a computer

Description automatically generated with medium confidence

1. “**make deploy**” command is used to perform pre requirements to the VM and download the necessary files.

Text

Description automatically generated

1. List the downloaded bin folder using the **ls** command.

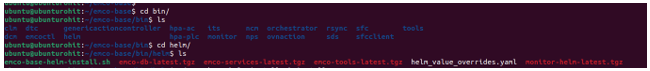
**Note:** After “make deploy” command is executed, the bin folder appear.

Text

Description automatically generated

1. Navigate to the below directory to get the emco-base-helm-install.sh file.

**(emco-base/bin/helm)**



1. Execute the below command for the creation of pods, deployments, services, namespace, etc.,

**./emco-base-helm-install.sh install**

Text

Description automatically generated with medium confidence

1. List all the services, pods, deployments under EMCO namespace using the below command.

**kubectl get all -n emco**

Text

Description automatically generated

1. Create the test directory and move to that directory using the below commands.

**Create:** mkdir emco-base/kud/tests

**Move:** cd emco-base/kud/tests

1. After moving to the test directory, download all the required repos.

**collectd.tar.gz**

wget <https://raw.githubusercontent.com/RohitArora7/collectd_prometheus/master/collectd.tar.gz>  
**collectd\_profile.tar.gz**  
wget   
https://raw.githubusercontent.com/RohitArora7/collectd\_prometheus/master/collectd\_profile.tar.gz

**prometheus-operator.tar.gz**

wget <https://raw.githubusercontent.com/RohitArora7/collectd_prometheus/master/prometheus-operator.tar.gz>

**prometheus-operator\_profile.tar.gz**

wget <https://raw.githubusercontent.com/RohitArora7/collectd_prometheus/master/prometheus-operator_profile.tar.gz>

1. Below screen shot shows all the downloaded repos and the **ls** command is used to list it.

A screenshot of a computer

Description automatically generated with medium confidence

1. List all the services created using below command.

**kubectl get svc -n emco**

Text

Description automatically generated

1. Navigate to the below path and edit the IPs (Associated to pod service) in **emco-cfg-remote.yaml** file.

**cd emco-base/src/tools/emcoctl/examples**

Graphical user interface, text

Description automatically generated

1. Edit the test.yaml file using the below command.

**cd emco-base/src/tools/emcoctl/examples**

**vim test.yaml**

Edit the following line numbers

39 /path/to/kubeCfg

111 /path/to/prometheus-operator.tar.gz

124 /path/to/collectd.tar.gz

150 /path/to/prometheus-operator\_profile.tar.gz

165 /path/to/collectd\_profile.tar.gz

i.e.,

39 /home/ubuntu/.kube/config

111 /home/ubuntu/emco-base/kud/test/prometheus-operator.tar.gz

124 /home/ubuntu/emco-base/kud/test/collectd.tar.gz

150 /home/ubuntu/emco-base/kud/test/prometheus-operator\_profile.tar.gz

165 /home/ubuntu/emco-base/kud/test/collectd\_profile.tar.gz

Text

Description automatically generated

1. Deploy the workload using emcoctl.
2. **Deploy EMCO UI Modules**
3. EMCOUI deployment.
4. Building GUI and guimiddleend docker images

docker build -t image\_name:version

cd guimiddleend   
docker build -t image\_name:version

1. Install EMCOUI using EMCOUI helm chart.
2. Access the UI: Open your browser and access http://<EMCO node IP>:30480/app
3. Deploy OpenShift cluster

* Openshift Kubernetes cluster is used for orchestrating the CNFs.

1. ALTRAN 5G Core preparation

* Contact Altran sales team for getting access to the CNFs and to the 5G cores helm charts.

1. Orchestration of 5G Core

* In Network Slicing Orchestration initially the 5G network service is created by going through the GUI of EMCO.
* Onboard the Kubernetes cluster, here it is the Redhat open shift cluster in the Kaloom lab.

1. **Step by step Procedure for Altran 5G Core Orchestration**
2. Select the clusters option from the left-hand side panel of EMCO and click on “**+Register Cloud Provider**” button, which open the Register Cloud Provider popup window.

Graphical user interface, application

Description automatically generated

1. In the Register Cloud Provider popup window, enter the provide name & the description and click on **Create** button.

Graphical user interface, text, application

Description automatically generated

1. A provider name is created in the clusters screen.

Graphical user interface, text, application

Description automatically generated

1. Click on the provider name and click on **Onboard Cluster** button, which popups an Onboard Cluster window.

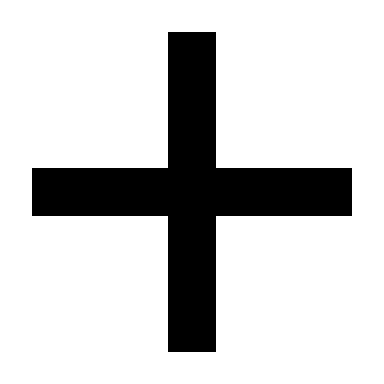
Graphical user interface, application

Description automatically generated

1. In the Onboard Cluster window, enter all the mandatory fields and click on **Submit** button.

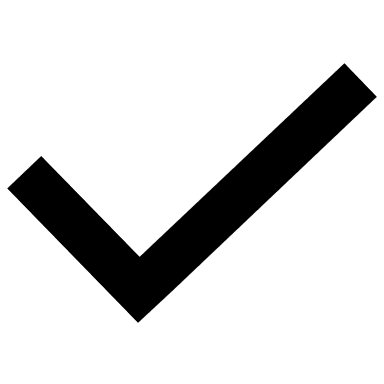
Graphical user interface, application

Description automatically generated

1. Once the submit button is clicked, a message displays as Cluster added MTL\_OC1 (As shown in below screen shot). Click on the “” button and add the label.

Graphical user interface, application, Teams

Description automatically generated

1. Enter the label name and click on the “” button as shown in below screen shot, which navigates to the Logical Clouds page.

Graphical user interface, application

Description automatically generated

* The logical cloud is created, which is a way to create a logical construct on top of the multiple physical cloud for multi-tenancy.

1. In the Logical Clouds page, click on the “**Create Logical Cloud”** button, which opens a Create Logical Cloud window.

Graphical user interface, application

Description automatically generated

1. In the Create Logical Cloud window, enter the values, select the cluster and click on **create** button.

Graphical user interface, application

Description automatically generated

1. LC1 logical cloud is created

Graphical user interface, application

Description automatically generated

* Add services using the 5G core and the next-gen firewall.
* Onboarding two applications, the next-gen core and the 5G core with its associated helm chart and day zero configuration overrides which concludes the design phase of the Network service that is created.

1. Select the **Services** option from the left panel (Highlighted in green) of EMCO and click on the **Add Services** button (Highlighted in red), which navigates to the Add Services page.

Graphical user interface, application

Description automatically generated

1. In the Add Services page, enter the name & description and click on “**+Add Application**” button, which popups the Add Application window.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. In the Add Application window, enter the application name & description and click on **Add** button, which navigates to the Add Service page.

Graphical user interface, text, application

Description automatically generated

1. In the Add Service page, click on the drop-down arrow button.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Upload the **App tgz file** & **Confi Override file** and click on **Add Application** button, which opens “Add Application” popup window.

Graphical user interface, text, application

Description automatically generated

1. In the Add Application popup window, enter the Application name & description and click on **Add** button, which navigates to the Add Service page.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. In the Add Service page, click on the drop-down arrow button.

Graphical user interface, application

Description automatically generated

1. Upload the App tgz file and Confi Override file.

Graphical user interface, application

Description automatically generated

* Here the instantiation is done. First the runtime instance is created where user describes where the application should be placed in terms of the indent.

1. Select the **Service Instances** option from the left side panel of EMCO and click on the “**Create Service Instance**” button, which navigates to the Create Service Instance page.

Graphical user interface, application

Description automatically generated

1. In the Create Service Instance page, enter the values in all the mandatory fields (To create the runtime instance) and click on **Next** button.

Graphical user interface, application, website

Description automatically generated

1. Click on the drop-down arrow (As shown in below screen shot, highlighted in Red)

Graphical user interface, application

Description automatically generated

1. Select the label.

**5G Core:**

Graphical user interface, application, website

Description automatically generated

**Firewall:**

Graphical user interface, application, website

Description automatically generated

1. Click on **submit** button, which navigates to the service instance page.

Graphical user interface, application

Description automatically generated

1. In the service instance page, click on the **instance** icon (Highlighted in red).

Graphical user interface, application

Description automatically generated

1. A popup message displays asking whether to instantiate or not. Click **ok**.

Graphical user interface, application

Description automatically generated

* After instantiating, the Instance icon is disabled (Highlighted in red in below screen shot) and a message displays as “Service ViNGC Instantiated”.

1. Click on the hyperlink (Highlighted in green), which navigates to the Service Instance Detail page.

Graphical user interface, application

Description automatically generated

1. In the Service Instance Detail page, the status of the GUI is shown.

Graphical user interface, application

Description automatically generated

1. For viewing the pods, use the below command.

**$ oc get pod -n ns-ngc**

Graphical user interface, text

Description automatically generated

1. **Go to ONAP GUI to do Network Slicing.**

* Slice parameters are configured by a custom Network Slice Subnet Management Function (NSSMF)
  + From data point of view first the slice is created in ONAP, where the Network Slice Subnet Management Function (NSSMF) goes between the ONAP and the 5G core.
* Video sessions initiated across good and bad slices.
* The NSSMF passes the parameters to the 5G core and two slices are created, one is the Good Slice with 5mb per second and the other is Bad Slice with 2.5mb per second.
* Slice SLA maintained by the A10 NGFW
* The SLAs are honored by the 5G core and the firewall.
* Video sessions terminated.

1. Create the communication service through the CSMF.
2. Click on the create button, which navigates to the “Create Communication Service” page.

Graphical user interface, application, Teams

Description automatically generated

1. In the “Create Communication Service” page, enter all the mandatory fields and click on **Ok** button.

Graphical user interface, application

Description automatically generated

* Two slices are created, a good slice with 5mb per second and the bad slice with 2.5 megabits per second.

1. Go to the Network service management function or the NSMF GUI, which shows the network slice instance that is selected, which is approved by the users with some overrides. Once it is done, the slices are pushed out.

Graphical user interface, application, Teams

Description automatically generated

1. Check whether the NSSMF (Network Slice Subnet Management Function) which was created is talking to the Altran 5G core and as per below screen shot it is communicating to the Altran 5g core.

Text

Description automatically generated

Text

Description automatically generated

1. Similarly, the NSSMF communicates to the Kaloom UPF over Netconf and when this completes the slices are created successfully

Text

Description automatically generated

1. In the quantitative aspect, it shows the bandwidth of the good slice and the bad slice.

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with medium confidence

1. Deploy gNB simulator

* What is the purpose of gNB simulator in the Cloud Native 5GS blueprint.
* Contact Rebaca for getting access to gNB simulator.

1. Testing

* Testing is done using Abot tool, which is a gNB and UE emulator. Here user runs two tests, they are;
* Connection Request test
* UE Authentication test

1. **Connection Request Test**

* The connection request test goes over the end to interface to the AMF and pass successfully.

1. Select the Feature File option from the left hand side panel of Abot and select the feature that needs to be tested.

Graphical user interface, text, application, email

Description automatically generated

1. Belowscreenshot show thatthe test is executed successfully.

Graphical user interface

Description automatically generated with low confidence

1. Click on Artefacts option, select aarna\_5gc🡪5G\_Test, it lists out tested item with its date and time. Select the required folder and view its features. Here the first 12 tests passed successfully for the connection request test.

Graphical user interface, text, application

Description automatically generated

1. **UE Authentication Test**
2. Select the Feature File option from the left hand side panel of Abot and select the feature that needs to be tested.

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated with low confidence

1. Below screen shot shows that the test execution has started and executed successfully.

Chart

Description automatically generated with medium confidence

1. Click on Artefacts option, select aarna\_5gc🡪5G\_Test, it lists out tested item with its date and time. Select the required folder and view its features. Here the 16 tests passed successfully for the UE Authentication test.

Graphical user interface, text, application

Description automatically generated

1. Steps to add EMCO UI Cluster and Controller Registration Flow
2. **sOnboard K8s cluster**
3. Before onboarding a K8s cluster, register a cluster provider by clicking on the 'Register Cluster Provider' button.

Graphical user interface, application

Description automatically generated

1. For adding a cluster under the cluster provider, click on the cluster provider drop down and click on 'Add Cluster' button, which pops up the Add Cluster window.

Graphical user interface, application

Description automatically generated

1. In the Add Cluster window (As shown in above screen shot) fill the fields, upload the kube config of the cluster and click create button.

**Note:** In case of OpenShift, specify the username/password in the 'User Data' section in a format as shown in the below example.

*{*  
*"cloud-region" : "oc",*  
*"cloud-owner" :  "mtl",*  
*"other-connectivity-list" : {*  
*"connectivity-records": [{*  
*"ssl-initiator": "false",*  
*"user-name": "kubeadmin",*  
*"password": <password>*  
*}]*  
*}*  
*}*

1. Once a cluster is added, add labels to it by clicking the plus icon in the labels column.  No need to add networks for this DEMO.
2. **Register Controllers**
3. Click on the Controllers from the panel on the left and add Rsync controller.

host: rsync

port: 9041

type: < leave blank, not required.>

priority: < leave blank, not required>

Graphical user interface, application

Description automatically generated

# ISSUES ENCOUNTERED

No known issues.

# WORKAROUNDS

No workarounds.

# MITIGATIONS

No mitigations.

# LESSONS LEARNED

* Need information
* For 5G core user must develop NSSMF for each core, because NSSMF is not generic. If user wants to do the same thing with another core like magma, then they must develop a new NSSMF layer.