

ONAP 5G Network Slicing Quick Guide

1. Introduction	2
2. ONAP deployment and image changes on Guilin/Honolulu.	2
3. SDC Template creation	3
4. Core Network and Radio Access Network NSSMF simulator	11
4.1 Download CN and RAN NSSMF simulator	11
4.2 Register Core Network (CN) NSSMF simulator with ESR	12
4.3 Register RAN NSSMF simulator with ESR	13
4.4 Running CN NSSMF simulator in ONAP cluster	15
4.5 Running RAN NSSMF simulator in ONAP cluster	16
5. RestAPIs on NSSMF simulator.	16
5.1 CN NSSMF simulator RestAPIs	16
5.2 RAN NSSMF simulator RestAPIs	17
6. Use case UI (UUI)	17
6.1 Operational steps for 5G network slices	17
7. References	23

Version	Remarks
1.0	Bring up guide for ONAP network slicing

1. Introduction

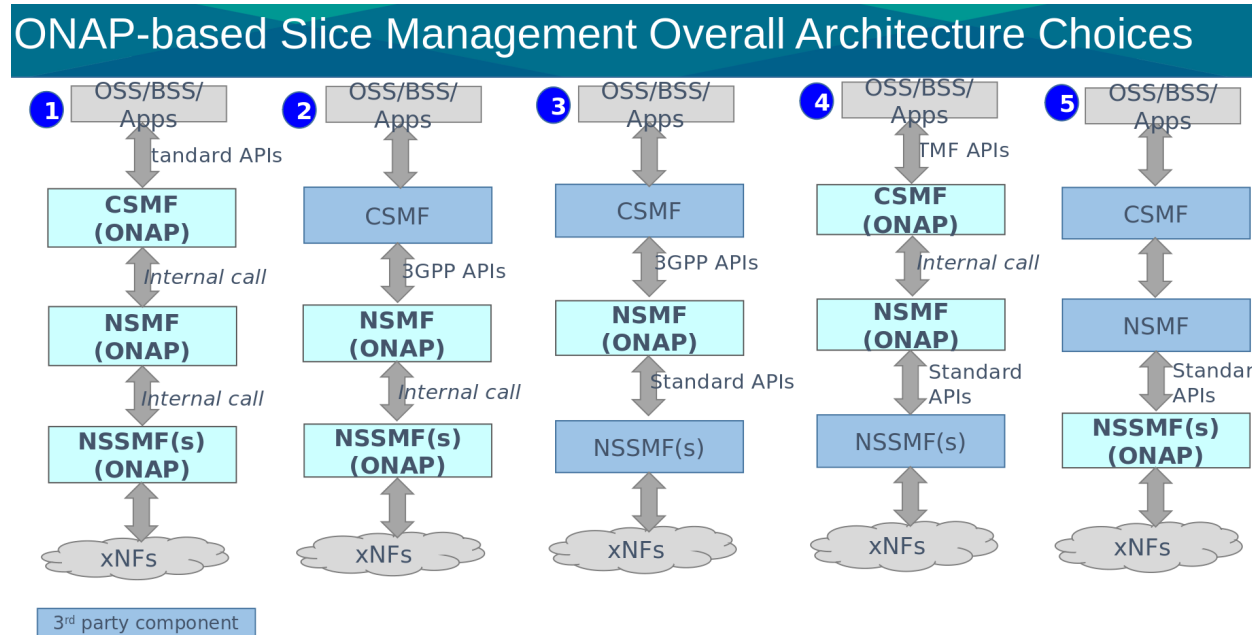


Figure-Intro-1: ONAP slice architecture choices

ONAP has provided various architecture choices for slice management. In this quick reference guide, we will uncover steps required to bring up ONAP cluster with architecture choice#4 as shown in *Figure-Intro-1* above.

2. ONAP deployment and image changes on Guilin/Honolulu.

Here are the steps.

- For ONAP Guilin and Honolulu release, if one doesn't want to exercise the transport domain of the slice, we need to disable the transport slice. We use the below patch in the SO helm charts to perform the same.

```
--- a/kubernetes/so/components/so-bpmn-infra/resources/config/overrides/override.yaml
+++ b/kubernetes/so/components/so-bpmn-infra/resources/config/overrides/override.yaml
@@ -160,6 +160,8 @@ mso:
   endpoint: https://oof-osdf.{{ include "common.namespace" . }}:8698
   timeout: PT30M
   workflow:
+   TnNssmf:
+     enableSDNCNetworkConfig: 'false'
   CreateGenericVNFV1:
```

aai:
volume-group:

- b. For ONAP Honolulu release as of 22nd July we need to apply the below patch.
<https://gerrit.onap.org/r/c/oom/+121768>

We need to add an override value to SDNC, otherwise SDNC will not work. (Note: This can be done by deploying SDNC separately at the end with the command line parameter)

Adding override:

global:
cmpv2Enabled: false

- c. After the deployment is complete, we have to follow the steps mentioned in the below wiki. We need to make sure that the image versions of different components are matching with what is specified in the wiki.

<https://wiki.onap.org/display/DW/Setup+related+issues>

3. SDC Template creation

We use the below wiki link for the details on how to build the slicing template.

<https://wiki.onap.org/display/DW/Template+Design+for+Option2>

Below are some additional screenshots for reference.

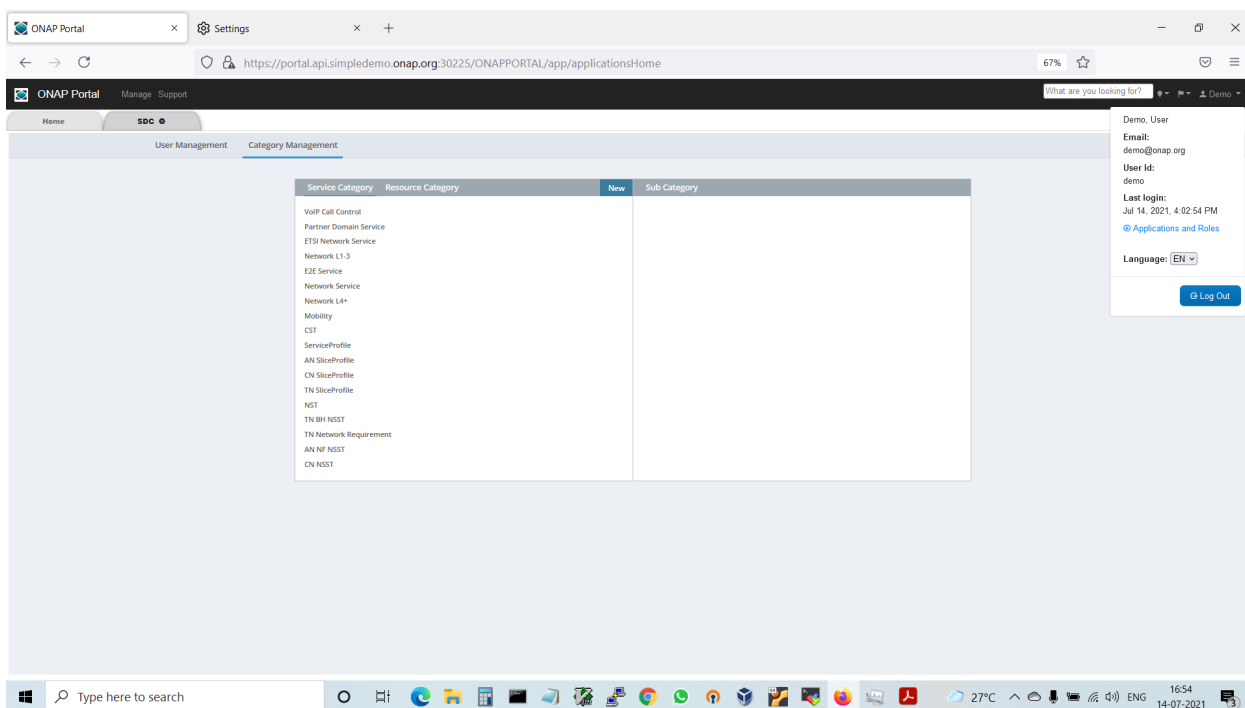


Figure-SDC-1: Creating new service category in SDC

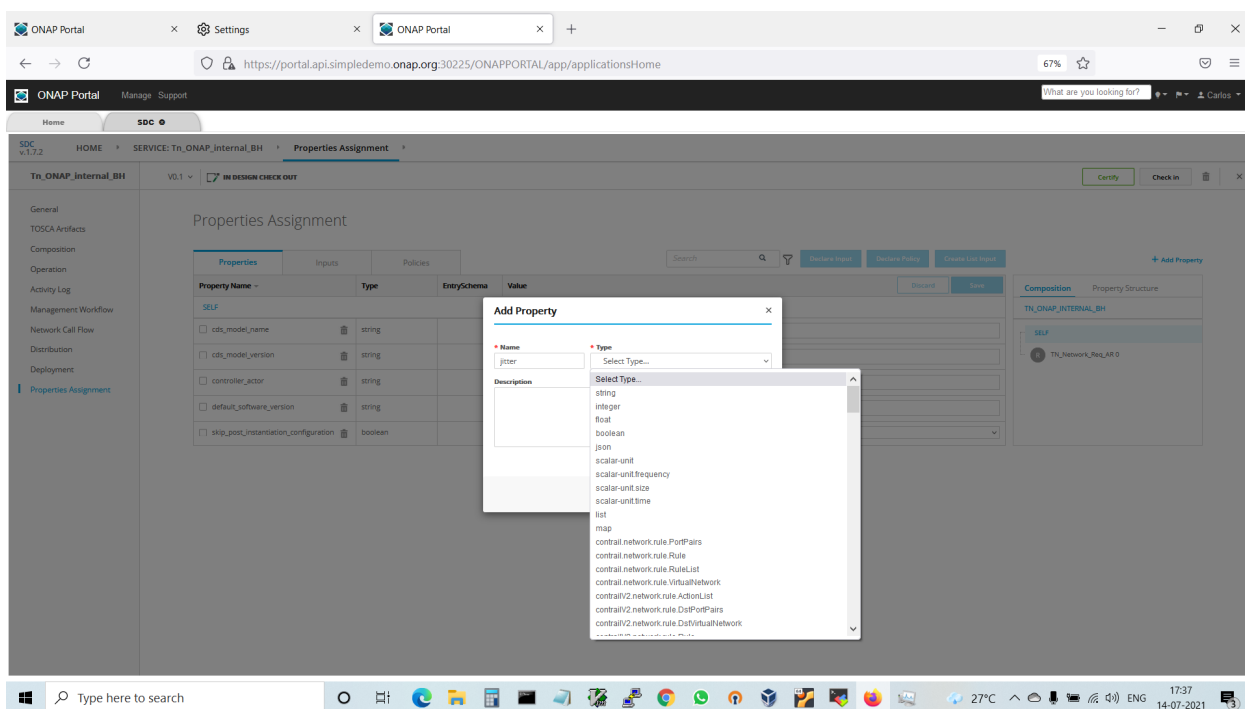


Figure-SDC-2: Adding a property attribute to selected service in SDC

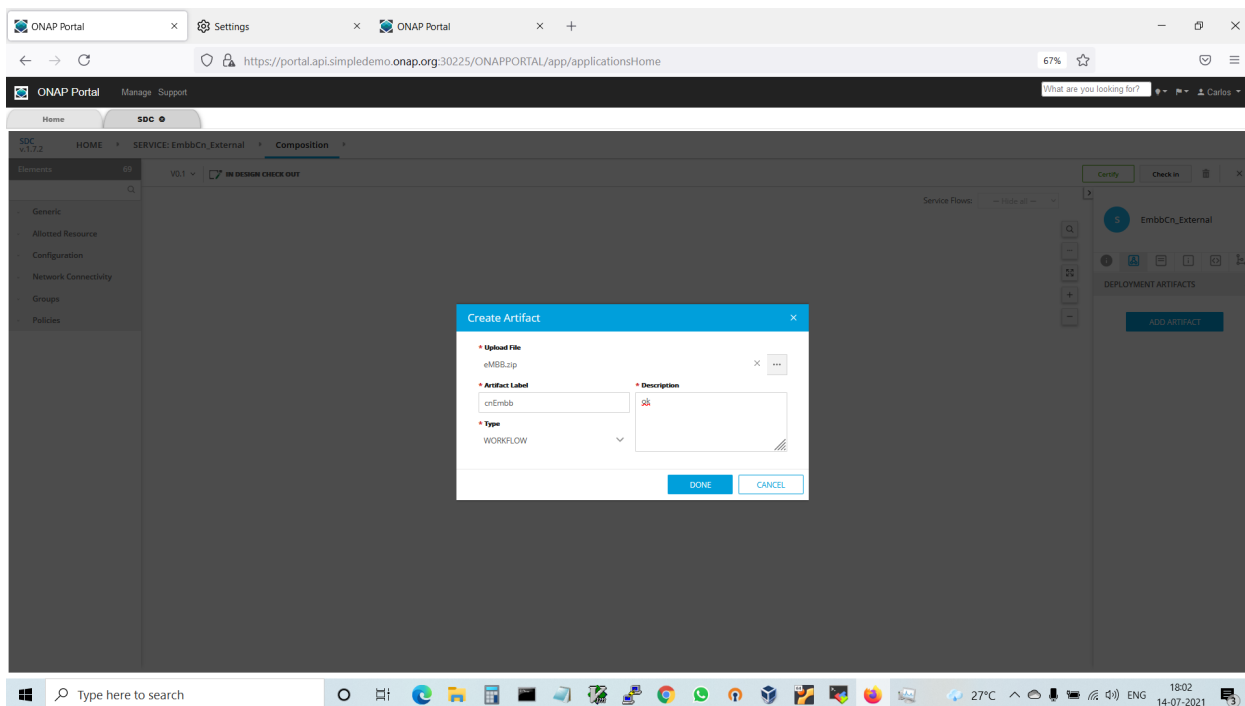


Figure-SDC-3: Create Artifact in SDC for service eMBB

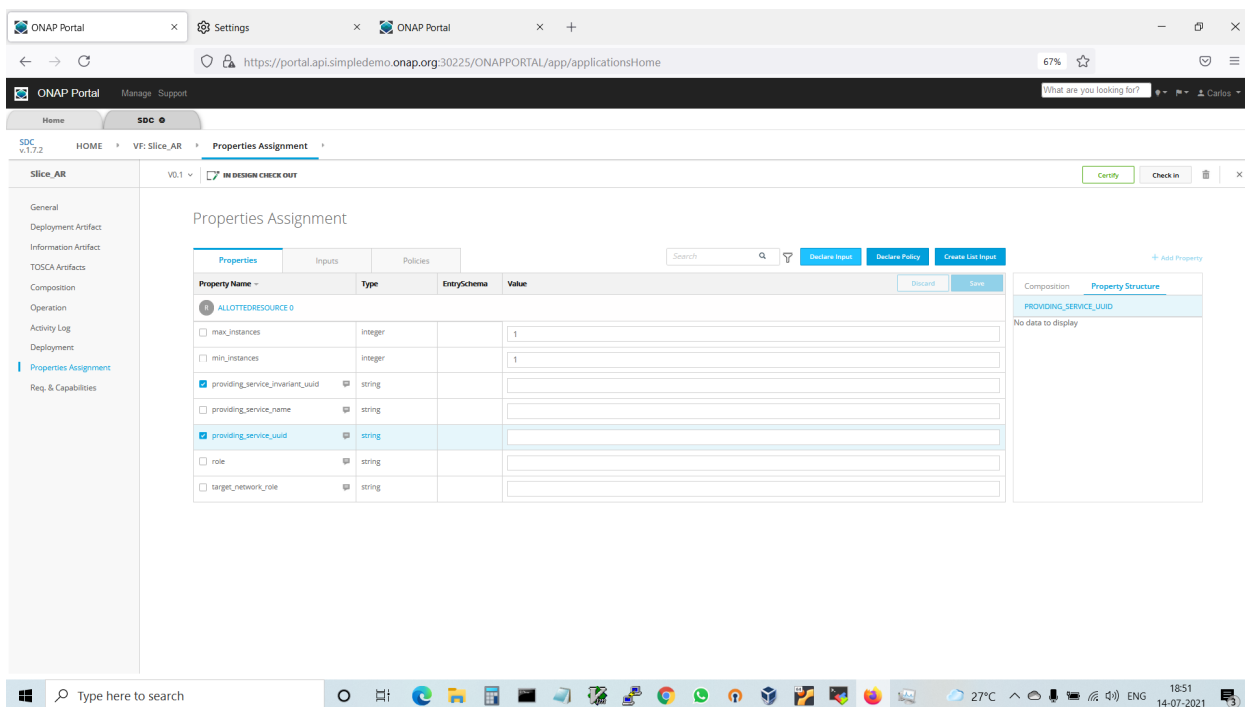


Figure-SDC-4: Allotted Resource property assignment

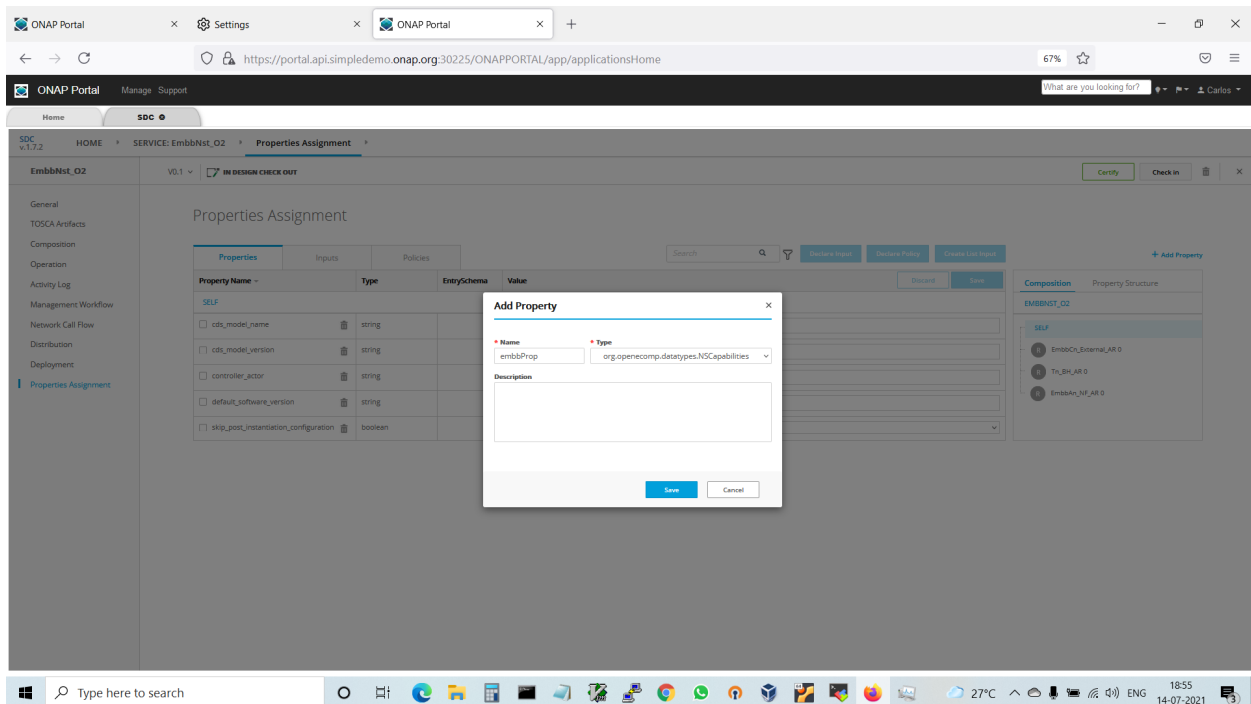


Figure-SDC-5: Add property to eMBB NST service template

Sometimes you may see the error like below, this can be safely ignored. It is due to network connectivity issues.

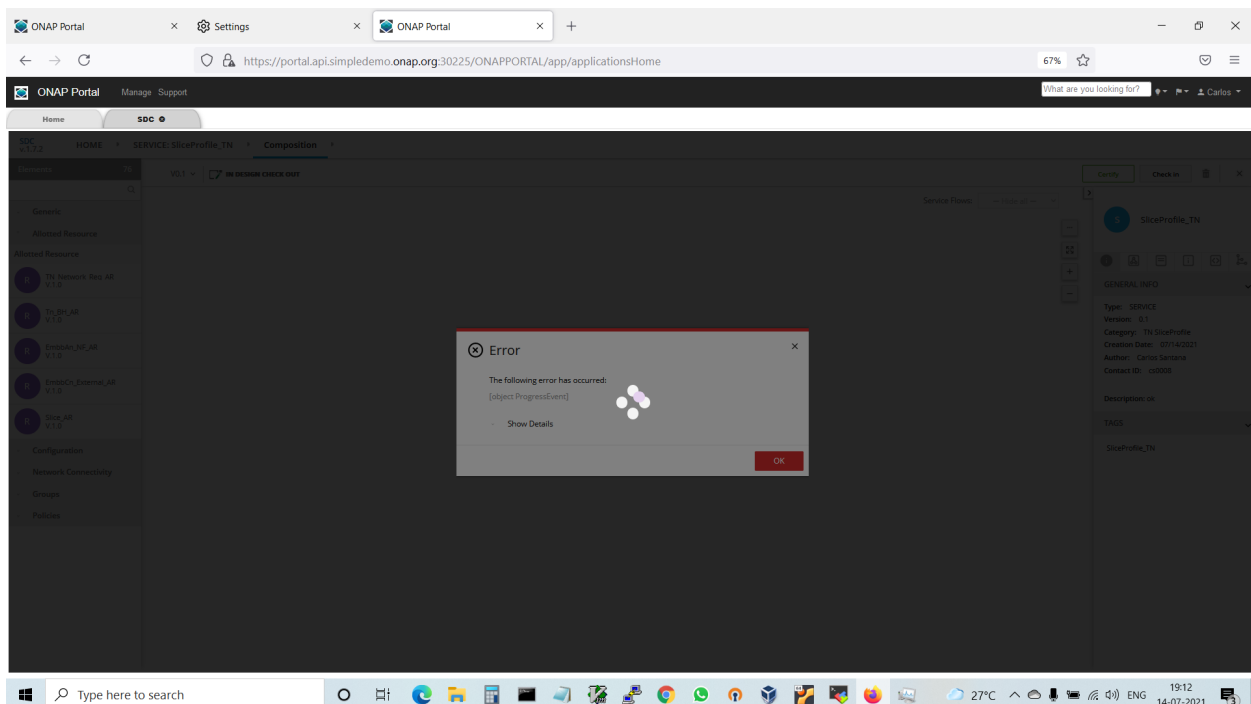


Figure-SDC-6: Error screen

In some cases distribution fails. Like below.

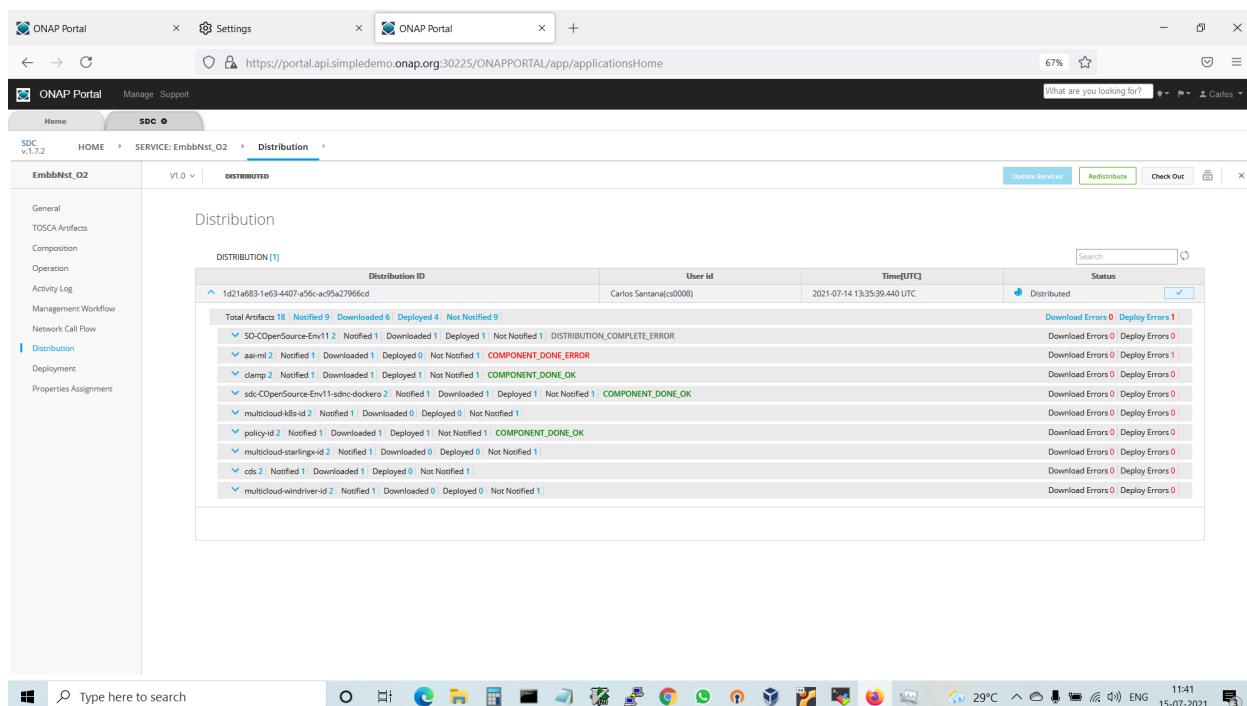


Figure-SDC-7: Distribution failed screen for eMBB service for NST

In such a case, make sure that the below components are added to AAI. The below steps ensure that all the necessary components are added to AAI.

```
curl --user AAI:AAI -X PUT -H X-FromAppId:AAI -H X-TransactionId:get_aai_subscr -H
Accept:application/json -H Content-Type:application/json -k -d '{ "global-customer-id":"5GCustomer",
"subscriber-name":"5GCustomer", "subscriber-type":"INFRA"}'
https://10.43.71.18:8443/aai/v21/business/customers/customer/5GCustomer
```

```
curl --user AAI:AAI -X PUT -H X-FromAppId:AAI -H X-TransactionId:get_aai_subscr -H
Accept:application/json -H Content-Type:application/json -k
https://10.43.71.18:8443/aai/v21/business/customers/customer/5GCustomer/service-subscriptions/service
-subscription/5G
```

```
curl --user AAI:AAI -X PUT -H "X-FromAppId:AAI" -H "X-TransactionId:get_aai_subscr" -H
"Accept:application/json" -H "Content-Type:application/json" -k -d '{ "model-invariant-id":
"8f50cc5d-3fb4-493f-90a6-b1273383ec8d", "model-type": "resource", "model-vers": { "model-ver": [ {
"model-version-id": "4d804d99-5e82-4ba4-a4a8-badf25497df8", "model-name": "Slice_AR",
"model-version": "1.0" } ] } } }
```

"https://10.43.71.18:8443/aai/v21/service-design-and-creation/models/model/8f50cc5d-3fb4-493f-90a6-b1273383ec8d"

```
curl --user AAI:AAI -X PUT -H "X-FromAppld:AAI" -H "X-TransactionId:get_aai_subscr" -H
"Accept:application/json" -H "Content-Type:application/json" -k -d '{ "model-invariant-id":
"6a37bb9e-0741-416e-97ca-aa3e276db74a", "model-type": "service", "model-vers": { "model-ver": [ {
"model-version-id": "e26c11c0-1625-47a0-a2e8-6ae1732870a1", "model-name": "EmbbNst_O2",
"model-version": "1.0" } ] } }'
"https://10.43.71.18:8443/aai/v21/service-design-and-creation/models/model/6a37bb9e-0741-416e-97ca-aa3e276db74a" | python -m json.tool
```

```
curl --user AAI:AAI -X PUT -H "X-FromAppld:AAI" -H "X-TransactionId:get_aai_subscr" -H
"Accept:application/json" -H "Content-Type:application/json" -k -d '{ "model-invariant-id":
"3da37387-c5c3-4c7e-bf2b-2b361b34290d", "model-type": "resource", "model-vers": { "model-ver": [ {
"model-version-id": "d858b41c-e97f-4e16-8d59-26e10b170231", "model-name": "EmbbAn_NF_AR",
"model-version": "1.0" } ] } }'
"https://10.43.71.18:8443/aai/v21/service-design-and-creation/models/model/3da37387-c5c3-4c7e-bf2b-2b361b34290d" | python -m json.tool
```

```
curl --user AAI:AAI -X PUT -H "X-FromAppld:AAI" -H "X-TransactionId:get_aai_subscr" -H
"Accept:application/json" -H "Content-Type:application/json" -k -d '{ "model-invariant-id":
"34063b6e-3c24-4a96-bbd6-5466fecb2efc", "model-type": "resource", "model-vers": { "model-ver": [ {
"model-version-id": "879300a0-1354-4f1f-9676-39332df4cfa9", "model-name": "Tn_BH_AR",
"model-version": "1.0" } ] } }'
"https://10.43.71.18:8443/aai/v21/service-design-and-creation/models/model/34063b6e-3c24-4a96-bbd6-5466fecb2efc" | python -m json.tool
```

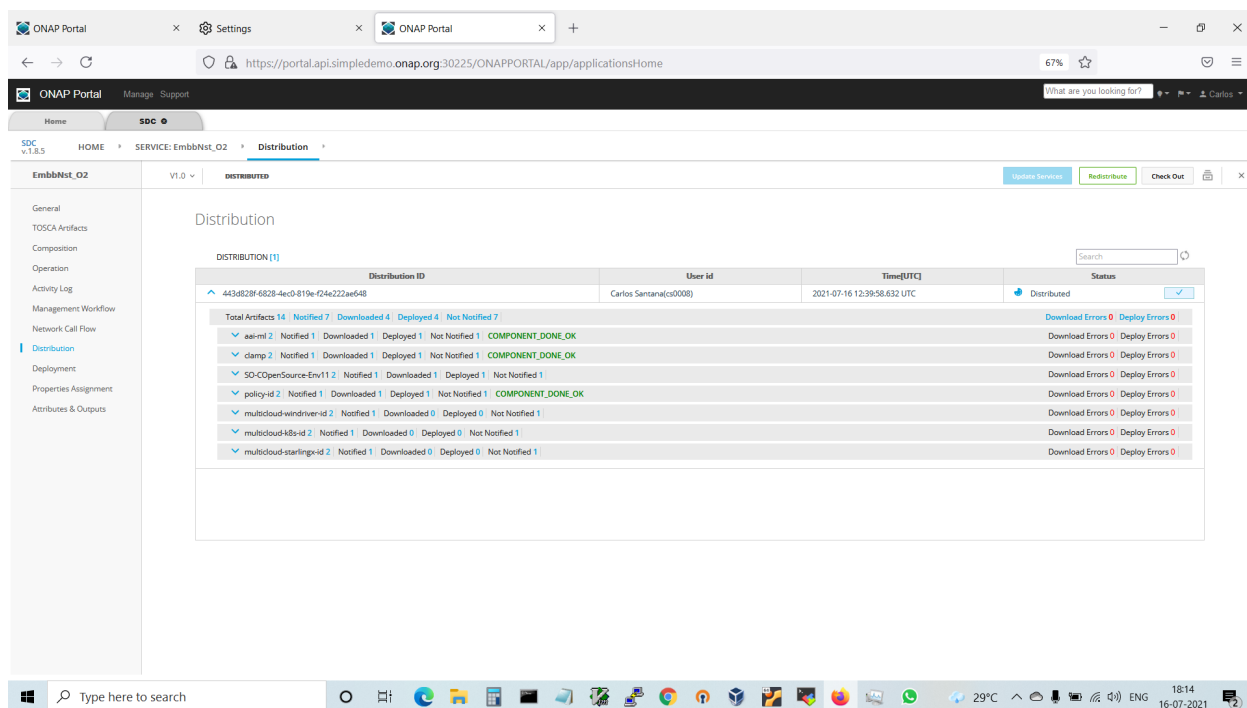


Figure-SDC-8: Distribution successful screen for eMBB service for NST

Policy update in OOF pod. Use the below commands to update the policy.

```
python3 policy_utils.py create_policy_types policy_types
python3 policy_utils.py generate_nsi_policies EmbbNst_O2
python3 policy_utils.py create_and_push_policies gen_nsi_policies
python3 policy_utils.py generate_nssi_policies EmbbAn_NF minimize latency
python3 policy_utils.py create_and_push_policies gen_nssi_policies
python3 policy_utils.py generate_nssi_policies Tn_ONAP_internal_BH minimize latency
python3 policy_utils.py create_and_push_policies gen_nssi_policies
python3 policy_utils.py generate_nssi_policies EmbbCn_External minimize latency
python3 policy_utils.py create_and_push_policies gen_nssi_policies
```

When the above policy update command is performed, one may see various warning messages and logs generated as shown below for reference. These can be ignored.

```
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-api'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
```

```
warnings.warn(
Policy type created
```

```
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-api'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
```

```
warnings.warn(
```

Policy type created

```

/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-api'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
Policy vnfPolicy_EmbbNst_O2_non_shared created
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-pap'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
b{"message": "Use the policy status url to fetch the latest status. Kindly note that when a policy is
successfully undeployed, it will no longer appear in policy status
response.", "uri": "/policy/pap/v1/policies/status"}
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-api'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
Policy minimizeLatency_EmbbNst_O2 created
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-pap'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
b{"message": "Use the policy status url to fetch the latest status. Kindly note that when a policy is
successfully undeployed, it will no longer appear in policy status
response.", "uri": "/policy/pap/v1/policies/status"}
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-api'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
Policy vnfPolicy_EmbbNst_O2 created
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-pap'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
b{"message": "Use the policy status url to fetch the latest status. Kindly note that when a policy is
successfully undeployed, it will no longer appear in policy status
response.", "uri": "/policy/pap/v1/policies/status"}
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning: Unverified
HTTPS request is being made to host 'policy-api'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
Policy maximizeCost_EmbbNst_O2 created

```

4. Core Network and Radio Access Network NSSMF simulator

In this section we shall cover following items about CN and RAN NSSMF simulator

- How to download the CN and RAN NSSMF simulator.
- Registering CN and RAN NSSMF simulator with ESR.
- How to run the NSSMF simulators for CN and RAN.

4.1 Download CN and RAN NSSMF simulator

- One can download the community built Core Network (CN) NSSMF simulator by following the below steps.
 - Perform wget to download the software on the master node.
 - `wget`
<https://wiki.onap.org/download/attachments/81397712/cn-nssmf-simulator-docker-compose.tar.gz>
 - Untar the download file into a directory
 - `tar -xvf cn-nssmf-simulator-docker-compose.tar.gz`
 - One can verify following files been present
 cn-nssmf-simulator-docker-compose/
 cn-nssmf-simulator-docker-compose/application.properties
 cn-nssmf-simulator-docker-compose/simulator-0.0.1-SNAPSHOT.jar
 cn-nssmf-simulator-docker-compose/dcae/
 cn-nssmf-simulator-docker-compose/docker-compose.yml
 cn-nssmf-simulator-docker-compose/luk.keystore
 - Check and configure `server.port` value in application.properties file matches the `ports` value in docker-compose.yml
 - Snippet of application.properties file is shown below:
`$ cat application.properties`
server.port=60053
 notifyurl=http://192.168.122.145:30472/v1/pm/notification
 ftppath=sftp://root:oom@192.168.122.145:22/home/ubuntu/dcae/PM.tar.gz
 z
 fixeddelay=900000
 filepath=/app/dcae
 amffilepath=/app/dcae/AMF.xml.gz
 upffilepath=/app/dcae/UPF.xml.gz
 - Snippet of docker-compose.yml file is shown below:
`$ cat docker-compose.yml`
 version: '3'
 services:

```
cn-simulator-docker-compose:
  image: openjdk:8-jre-slim
  container_name: cn-simulator-test-1
  ports:
    - "60053:60053"
  restart: always
  volumes:
    - ./app
  working_dir: /app
  entrypoint: java -jar simulator-0.0.1-SNAPSHOT.jar
```

- One can download the community built RAN NSSMF simulator by following the below steps.
 - a. Download the integration repository in the master node of ONAP deployment.
 - `git clone https://gerrit.onap.org/r/integration`
 - b. Install the requirements of RAN NSSMF.
 - `cd integration/test/mocks/ran-nssmf-simulator`
 - `sudo apt update`
 - `sudo apt install python3-pip`
 - `sudo pip3 install -r requirements.txt`

4.2 Register Core Network (CN) NSSMF simulator with ESR

- Follow the below steps to register CN NSSMF simulator with ONAP external registry system (ESR). The IP address (192.168.122.145) is an example referenced and needs to be replaced with the appropriate master node IP address of the ONAP deployment cluster.
 - a. Create a placeholder for resource id: 12345 in esr-thirdparty-sdnc
 - `curl -v -k -X PUT "https://192.168.122.145:30233/aai/v19/external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/12345" -H 'Accept: application/json' -H 'X-FromAppId: AAI' -H 'X-TransactionId: 1'`
 - b. Add the resource details as below:
 - `curl -v -k -X PUT "https://192.168.122.145:30233/aai/v19/external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/12345/esr-system-info-list/esr-system-info/12345" \ -H 'Accept: application/json' \ -H 'X-FromAppId: AAI' \ -H 'X-TransactionId: 1' \ -H 'Content-Type: application/json' \ -d '{`

```
"system-name": "E2E",
"vendor": "huawei",
"type": "cn",
"service-url": "",
"user-name": "",
"password": "",
"system-type": "thirdparty_SDNC",
"ip-address": "192.168.200.145",
"port": "60053"
}
```

- c. Verify the above addition of resource in ESR is reflecting correctly or not by performing a GET operation as shown below:

- `curl -v -k -X GET`
`"https://192.168.122.145:30233/aai/v19/external-system/esr-thirdparty-sd`
`nc-list/esr-thirdparty-sdnc/12345/esr-system-info-list/esr-system-info/1234`
`5" \`
`-H 'Accept: application/json' \`
`-H 'X-FromAppId: AAI' \`
`-H 'X-TransactionId: 1'`
- Result output from the above command will look like this with appropriate *resource-version* value:

```
{
  "esr-system-info-id": "12345",
  "system-name": "E2E",
  "type": "cn",
  "vendor": "huawei",
  "service-url": "",
  "user-name": "",
  "password": "",
  "system-type": "thirdparty_SDNC",
  "ip-address": "192.168.122.51",
  "port": "60053",
  "resource-version": "1618387830497"
}
```

4.3 Register RAN NSSMF simulator with ESR

- Follow the below steps to register RAN NSSMF simulator with ONAP external registry system (ESR). The IP address (192.168.122.145) is an example referenced and needs to be replaced with the appropriate master node IP address of the ONAP deployment cluster.

- a. Check if already the RAN NSSMF simulator is register by any chance with resource name as "sdnc-an-01"
 - `curl -s -k -u "AAI:AAI" -H "X-FromAppId: testApp" -H "X-TransactionId: 1001" -H "Accept: application/json" -X GET`
<https://192.168.122.145:30233/aai/v20/external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/sdnc-an-01/esr-system-info-list>
 - If not registered, you will see a sample output like below:


```
{
  "requestError": {
    "serviceException": {
      "messageId": "SVC3001",
      "text": "Resource not found for %1 using id %2 (msg=%3) (ec=%4)",
      "variables": [
        "GET",

"external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/sdnc-an-01/esr-system-info-list",
        "Node Not Found:No Node of type esr-system-info found at: external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/sdnc-an-01/esr-system-info-list",
        "ERR.5.4.6114"
      ]
    }
  }
}
```
- b. Add esr-thirdparty-sdnc to ESR.
 - `curl -s -k -u "AAI:AAI" -H "X-FromAppId: testApp" -H "X-TransactionId: 1001" -H "Content-Type: application/json" -H "Accept: application/json" -X PUT`
<https://192.168.122.145:30233/aai/v20/external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/sdnc-an-01> -d @sdnc-an-01.json
 - Use JSON payload in the file: sdnc-an-01.json as shown below:


```
{
  "thirdparty-sdnc-id": "sdnc-an-01"
}
```
- c. Add esr-system-info (RAN NSSMF) to ESR:
 - `curl -s -k -u "AAI:AAI" -H "X-FromAppId: testApp" -H "X-TransactionId: 1001" -H "Content-Type: application/json" -H "Accept: application/json" -X PUT`
<https://192.168.122.145:30233/aai/v20/external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/sdnc-an-01/esr-system-info-list/esr-system-info/nssmf-an-01> -d@nssmf-an-01.json
 - Use JSON payload in the file: nssmf-an-01.json as shown below:

```
{
  "esr-system-info-id": "nssmf-an-01",
  "type": "an",
  "vendor": "huawei",
  "system-type": "thirdparty-sdnc",
  "ip-address": "192.168.122.145",
  "port": "8443",
  "user-name": "admin",
  "password": "123456",
  "ssl-cacert": "test.ca"
}
```

d. Validate esr-system-info added is reflecting or not in ESR:

- `curl -s -k -u "AAI:AAI" -H "X-FromAppId: testApp" -H "X-TransactionId: 1001" -H "Accept: application/json" -X GET https://192.168.122.145:30233/aai/v20/external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/sdnc-an-01/esr-system-info-list`

- The above command will generate an output like below if successfully registered with ESR.
- Result output:

```
{
  "esr-system-info": [
    {
      "esr-system-info-id": "nssmf-an-01",
      "type": "an",
      "vendor": "huawei",
      "user-name": "admin",
      "password": "123456",
      "system-type": "thirdparty-sdnc",
      "ssl-cacert": "test.ca",
      "ip-address": "10.200.132.27",
      "port": "8443",
      "resource-version": "1618391040155"
    }
  ]
}
```

4.4 Running CN NSSMF simulator in ONAP cluster

- Follow the below steps to run the CN NSSMF simulator
 - a. Goto directory `cn-nssmf-simulator-docker-compose` as part of download procedure mentioned above.
 - b. Perform command:

- *docker-compose up*

4.5 Running RAN NSSMF simulator in ONAP cluster

- Follow the below steps to run the RAN NSSMF simulator
 - a. Goto directory *ran-nssmf-simulator* as part of download procedure mentioned above. i.e.
 - *cd integration/test/mocks/ran-nssmf-simulator*
 - b. Perform command:
 - *sudo python3 main.py*
 - Sample reference output is shown below:

```
aarna@anod-master:~/ran-nssmf/integration/test/mocks/ran-nssmf-simulator$ sudo python3 main.py
* Serving Flask app 'RanNssmfSimulator.MainApp' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on https://192.168.122.145:8443/ (Press CTRL+C to quit)
```

5. RestAPIs on NSSMF simulator.

In this section we are listing down the RestAPIs defined in the NSSMF simulator for Core Network and Radio Access Network domains

5.1 CN NSSMF simulator RestAPIs

Slice Events	Operation Type	Slice Domain	URI
Allocate NSSI	POST	CN	<code>/api/rest/provMns/v1/NSS/SliceProfiles</code>
Activate NSSI	PUT	CN	<code>/api/rest/provMns/v1/NSS/<string:snssai>/activation</code>
Deactivate NSSI	PUT	CN	<code>/api/rest/provMns/v1/NSS/<string:snssai>/deactivation</code>

DeAllocate NSSI	DELETE	CN	"/api/rest/provMns/v1/NSS/SliceProfiles/<string:sliceProfileId>"
GetJobStatus	GET	CN	"/api/rest/provMns/v1/NSS/jobs/<string:jobId>"

Table-1: NBI Interface exposed from CN NSSMF towards NSMF

5.2 RAN NSSMF simulator RestAPIs

Slice Events	Operation Type	Slice Domain	URI
Allocate NSSI	POST	AN	"/ObjectManagement/NSS/SliceProfiles"
DeAllocate NSSI	DELETE	AN	"/ObjectManagement/NSS/SliceProfiles/<string:sliceProfileId>"

Table-2: NBI Interface exposed from RAN NSSMF towards NSMF

6. Use case UI (UUI)

In this section, we shall cover details of slice service creation process and essentially showcase the lifecycle management of network slicing inclusive of following:

- Slice Creation
- Slice Activation
- Slice Deactivation
- Slice Termination

6.1 Operational steps for 5G network slices

- Navigate to UUI home page using portal link: <https://hostIP:30283/iui/usecaseui/#/home> as shown below in *Figure-uui-1*.

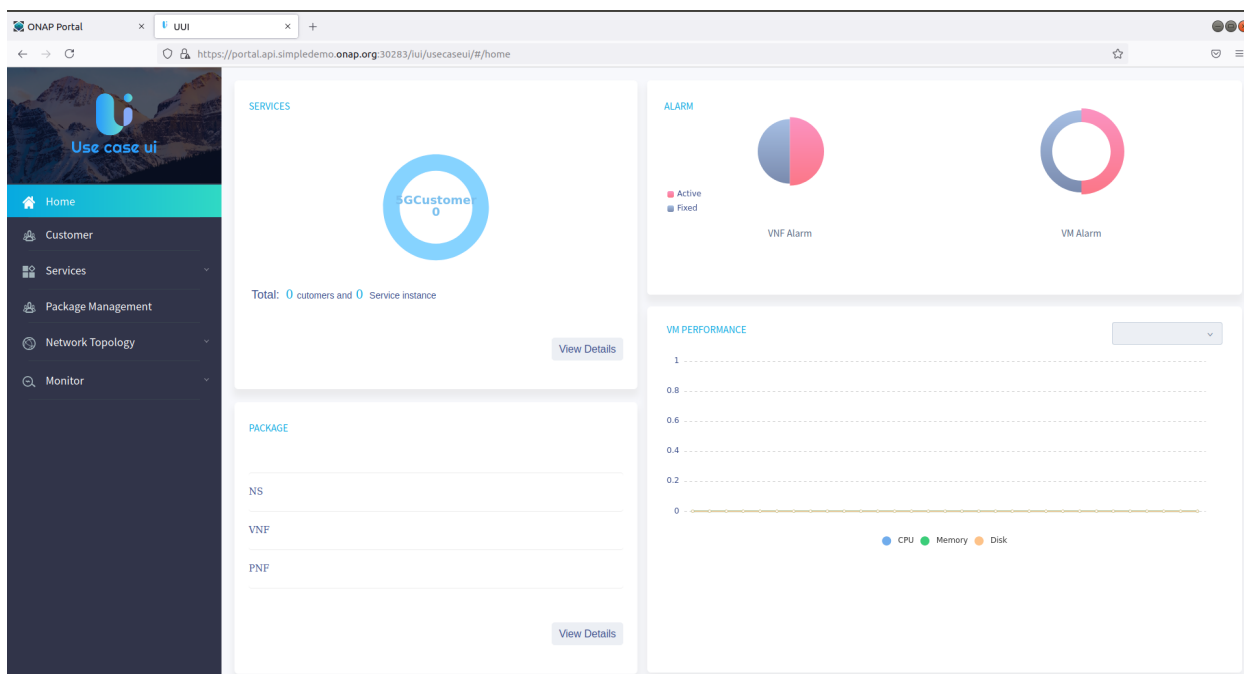


Figure-uu1-1: Home page of UII

- Navigate to 5G Slice Management under Services, as shown below in Figure-uu1-2.

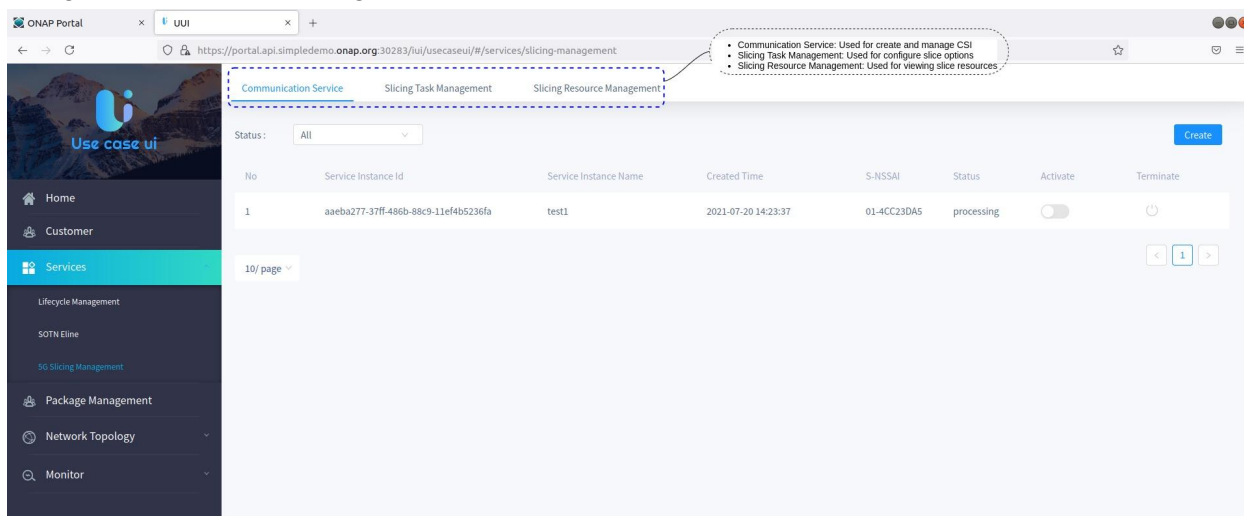


Figure-uu1-2: 5G Slice Management

- Slice Creation steps
 - a. Click 'Create' as shown in Figure-uu1-3 and fill the required parameters as shown in Figure-uu1-4 and Figure-uu1-5.

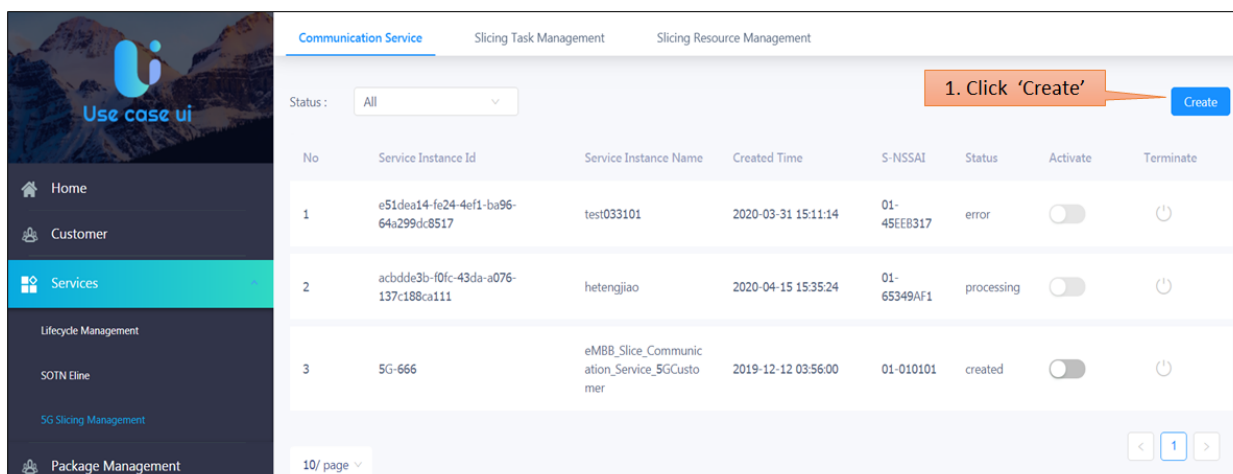


Figure-uu1-3: Slice Creation

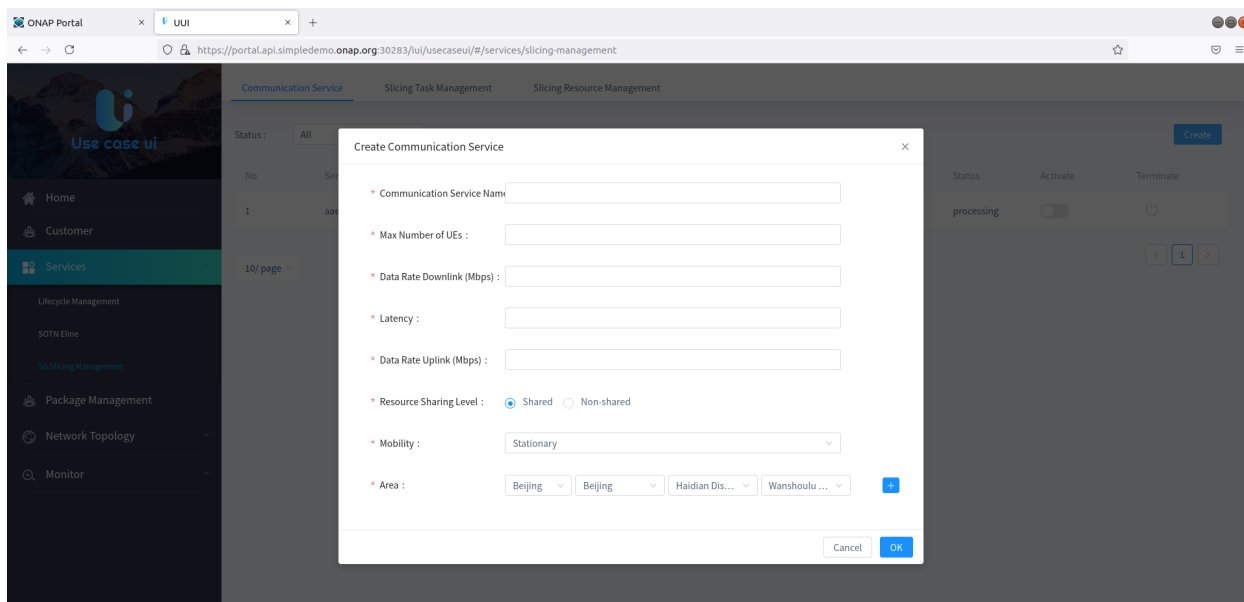


Figure-uu1-4: Create communication service

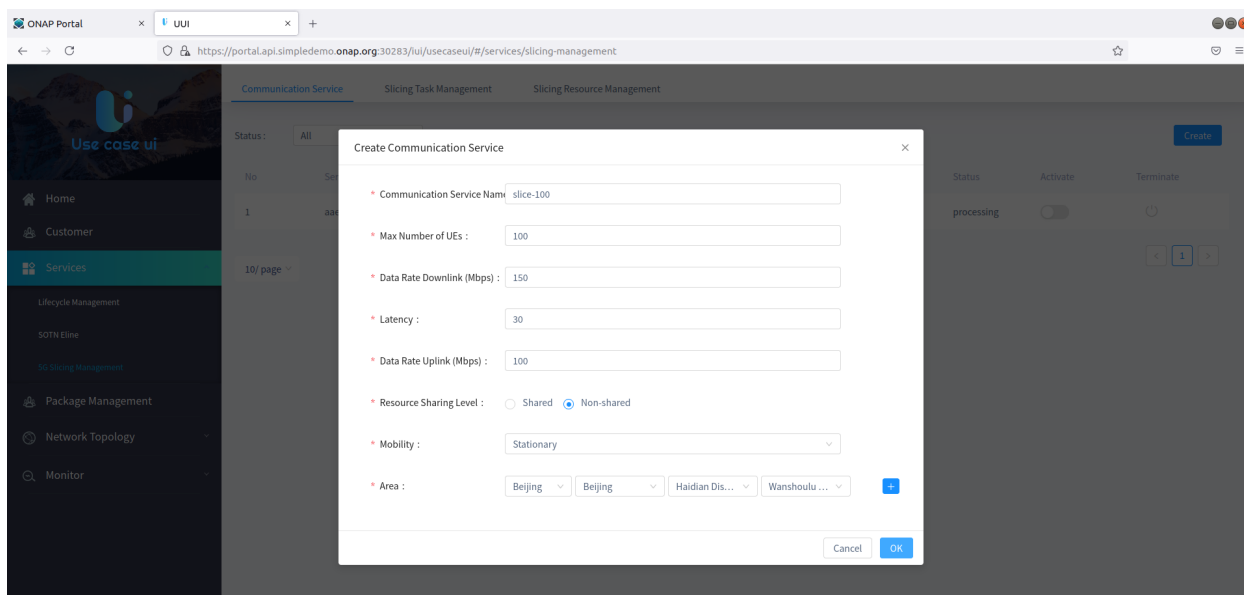


Figure-uu-5: Fill the parameters for communication service

- b. Switch to the *Slice Task Management* tab and click *Process Task* when the *Status* of the newly created task is 'WaitingToConfirm' as shown in Figure-uu-6.

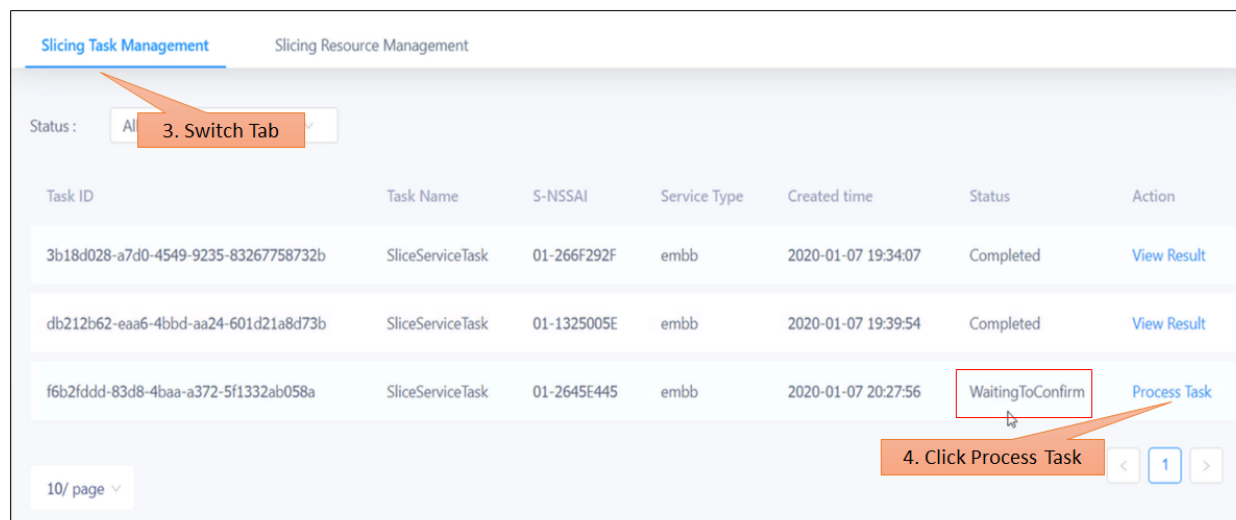


Figure-uu-6: Slice Task Management

- c. Decide whether to use a shared slice or create a new slice NSI by default as shown in Figure-uu-7. Fill in subdomain parameters for the AN and CN domain as per your requirement along with the Endpoint required by TN, refer Figure-uu-8 and Figure-uu-9
- An: Access Network
 - Tn: Transport Network

■ Cn: Core Network

The screenshot shows the 'New slice NSI' form. It includes sections for 'Matching NST Information', 'Matching Shared NSI', and 'Matching Shared NSSI'. The 'Matching Shared NSSI' section has three rows for 'An', 'Tn', and 'Cn', each with 'NSSI ID' and 'NSSI Name' fields. Callouts indicate: 5. Select a shared NSI(Optional) pointing to the 'NSSI ID' dropdown in the 'Matching Shared NSI' section; 6. Update domain parameters (Optional) pointing to the 'NSSI Name' field in the 'Cn' row; and 7. Click 'OK' to create pointing to the 'OK' button at the bottom right.

Figure-uu1-7: New slice NSI

The screenshot shows the 'Matching Shared NSSI' form. It has three rows for 'An', 'Tn', and 'Cn', each with 'NSSI ID' and 'NSSI Name' fields. Callouts indicate: 'Click to set Endpoint of AN' pointing to the 'NSSI Name' field in the 'An' row; and 'Click to set Endpoint of CN' pointing to the 'NSSI Name' field in the 'Cn' row. The 'OK' button is visible at the bottom right.

Figure-uu1-8: Navigate AN and CN to fill Endpoint details

Script Name :

Overall User Density :

* Endpoint :

* Coverage :

Figure-uu-9: Fill in Endpoint details

- d. Once the creation step is completed, switch to *Slice Resource Management* tab to view the NSI and NSSI resource information. Refer below Figure-uu-10.

Slicing Business Management Slicing Instance Management Slicing Subnet Instance Management

Status :

Service Instance Id	Service Instance Name	Service Type	Context	Status	Detail
NSSI-C-35F-HDBNJ-NSSMF-01-A-ZX	nssi_cnTest012801	embb	cn	deactivated	View Detail
2bd1fc60-0938-4617-ae93-401685921ae6	nssi_tnTest012801	embb	tn	deactivated	View Detail
8d8bbdf7df924c49aa502ff00bcf1b07	nssi_anTest012801	embb	an	activated	View Detail

Figure-uu-10: Slice Resource Management

- Steps for Slice Activation/Deactivation/Termination operation.
By default, when the slice is created it remains in status as Deactivation. Refer below Figure-uu-11 showing other operations (Activation/Deactivation/Termination).

Communication Service Slicing Task Management Slicing Resource Management

Status : All Create

No	Service Instance Id	Service Instance Name	Created Time	S-NSSAI	Status	Activate	Terminate
1	a3a12c73-1a54-4fe8-94e9-cbb08f6aef3	Test012801	2021-01-28 11:43:39	01-5E0E5291	deactivated	☐	⏻

Figure-uu1-11: Activation/Deactivation/Termination operations

7. References

ONAP Setup related Issues

- <https://wiki.onap.org/x/uj6LBQ>

ONAP SDC

- [Template Design for Option-2](#) (Guilin)
- [Design Service Template](#) (Frankfurt)

ONAP Use case UI

- <https://wiki.onap.org/display/DW/UI+Operation+Guidance+for+Guilin>
- <https://wiki.onap.org/display/DW/UI+Operation+Guidance+for+5G+Network+Slicing>