MDONS Validation Procedure vs User Documentations

ONAP Setup/Readiness

Configuring MSB

1. MSB UI discovery which will be in https://<cluster external IP>:30283/iui/microservices/default.html URL.

2. In order to create a service from UUI, the SO service create API path must be registered in MSB.
Click on service register and add the details below
- Name - so-serviceInstances
- URL - /onap/so/infra/e2eServiceInstances/v3
- Version - v3
- Protocol - REST
- Load balance: round-robin
- VisualRange - Insystem
- Host IP - will be SO Nodeport ip address and
- Port - 8080

2. To create SDC registration in MSB, do the following:

Click on service register and add the details below

- Name - sdc
- URL - /sdc/v1
- Version - v1
- Protocol - REST
- Load balance: round-robin
- VisualRange - Insystem
- Enable SSL- Select
- Host IP - will be SDC-be-external Nodeport ip address and
- Port - 8443

SO config update

MariaDB [catalogdb]> select * from service;

+--------------------------------------+------------------+----------------------------------+--------------+--------------------------+--------------------
+--------------------------------------+--------------+--------------+----------------------+-----------------+-----------------+----------------+----------------------------
+----------------------+--------------+----------------+-----------------+--------------------+---+-----------------+
| MODEL_UUID | MODEL_NAME | MODEL_INVARIANT_UUID | MODEL_VERSION | DESCRIPTION | CREATION_TIMESTAMP |
TOSCA_CSAR_ARTIFACT_UUID | SERVICE_TYPE | SERVICE_ROLE | ENVIRONMENT_CONTEXT | WORKLOAD_CONTEXT |
SERVICE_CATEGORY | RESOURCE_ORDER | OVERALL_DISTRIBUTION_STATUS | ONAP_GENERATED_NAMING | NAMING_POLICY |
CONTROLLER_ACTOR | CDS_BLUEPRINT_NAME | CDS_BLUEPRINT_VERSION | SKIP_POST_INSTANTIATION_CONFIGURATION |
service_function |
+--------------------------------------+------------------+----------------------------------+--------------+--------------------------+--------------------
+--------------------------------------+--------------+--------------+----------------------+-----------------+-----------------+----------------+----------------------------
+----------------------+--------------+----------------+-----------------+--------------------+---+-----------------+
| 89148094-786d-484b-9a06-c4af73856497 | L1 Access Service | b4e77117-783a-4b53-9688-ca2d7cd48748 | 1.0 | Layer 1 E2E Access Service | 2020-06-
26 17:42:01 | f8529f4d-8dc0-487b-9fcd-2df62ca7712f | MDONS_OTN | | General_Revenue-Bearing | Production | E2E Service | L1 UNI-UNI | NULL | 1 | |
NULL | NULL | NULL | 0 | |
+--------------------------------------+------------------+----------------------------------+--------------+--------------------------+--------------------
+--------------------------------------+--------------+--------------+----------------------+-----------------+-----------------+----------------+----------------------------
+----------------------+--------------+----------------+-----------------+--------------------+---+-----------------+

Here the service template is distributed with MODEL_VERSION = 1.0 and RESOURCE_ORDER=L1 UNI-UNI. These are cross referenced as below in catalog DB tables:

Catalog DB Table name	Column Name 1	Column Name 2
service	RESOURCE_ORDER	MODEL_VERSION
vnf_resource	MODEL_NAME	MODEL_VERSION
vnf_resource_customization	NF_ROLE	-
vnf_recipe	NF_ROLE	VERSION_STR

By default, the VNF Recipe entries are created with VERSION_STR="2.0" and NF_ROLE without "L1" prefix as in table below when deploying SO.

+----+------------+----------------+-------------+-------------+----------------------------------+---+---------------+----------------
+---------------------+-------------+
| id | NF_ROLE | ACTION | SERVICE_TYPE | VERSION_STR | DESCRIPTION | ORCHESTRATION_URI | VNF_PARAM_XSD | RECIPE_TIMEOUT |
CREATION_TIMESTAMP | VF_MODULE_ID |
+----+------------+----------------+-------------+-------------+----------------------------------+---+---------------+----------------
+---------------------+-------------+
| 15 | UNI-UNI | createInstance | NULL | 2.0 | OTN UNI-UNI resource create recipe | /mso/async/services/CreateSDNCNetworkResource | NULL | 180 |
2017-10-05 18:52:03 | NULL |
| 16 | UNI-UNI | deleteInstance | NULL | 2.0 | OTN UNI-UNI resource delete recipe | /mso/async/services/DeleteSDNCNetworkResource | NULL | 180 |
2017-10-05 18:52:03 | NULL |
| 17 | UNI-ENNI | createInstance | NULL | 2.0 | OTN UNI-ENNI resource create recipe | /mso/async/services/CreateSDNCNetworkResource | NULL | 180 |
2017-10-05 18:52:03 | NULL |
| 18 | UNI-ENNI | deleteInstance | NULL | 2.0 | OTN UNI-ENNI resource delete recipe | /mso/async/services/DeleteSDNCNetworkResource | NULL | 180 |
2017-10-05 18:52:03 | NULL |
+----+------------+----------------+-------------+-------------+----------------------------------+---+---------------+----------------
+---------------------+-------------+

In case the Service Template does not have same name for RESOURCE_ORDER as NF_ROLE(like in this case L1 UNI-UNI vs UNI-UNI) , then the vnf_recipe table has to be updated. The queries to update are as below:

MariaDB [catalogdb]>update vnf_recipe set VERSION_STR="1.0" where NF_ROLE like "UNI%";

MariaDB [catalogdb]>update vnf_recipe set NF_ROLE="L1 UNI-UNI" where NF_ROLE ="UNI-UNI";

MariaDB [catalogdb]>update vnf_recipe set NF_ROLE="L1 UNI-ENNI" where NF_ROLE ="UNI-ENNI";

MDONS Design Time

Service Design

Refer to this link for MDONS design time for MDONS_OTN service design and distribution.

Topology Discovery

Domain Controller (DC) is registered by adding the DC entry into AAI from a rest client or from command line. Assume the domain controller is up and running at <controller IP> and <controller port>.

Import DC Certificates to SDNC

Before register the DC to trigger the topology discovery, DC certificates needs to be imported into SDNC. But the way of importing in Frankfurt release is different from Guilin after Java 8 to 11 migration.

Only if the 3rd party domain controller is SSL-enabled, this CA importing step needs to be proceeded.

Frankfurt

1. Login to sdnc controller container from rancher vm
   kubectl exec -it -n onap dev-sdnc-sdnc-0 bash
2. use 'vi' to modify files.
3. cd to the directories that has the keystores and truststores
4. bash-4.4# **cd /opt/onap/sdnc/data/stores/**

   a) check if there is a file named **truststore.onap.client.jks_org** using 'ls' command.

   b) If the file exist, do the steps in c) . If the file doesn't exist, do the steps in d)

   c) bash-4.4$ **rm -f truststore.onap.client.msa.jks**
   bash-4.4$ **rm -f truststore.onap.client.tapi.jks**
   bash-4.4$ **cp truststore.onap.client.jks_org truststore.onap.client.tapi.jks**
   bash-4.4$ **cp truststore.onap.client.jks_org truststore.onap.client.msa.jks**

   d) back up the existing truststore file. truststore.onap.client.tapi.jks is used for the virtuoranc TAPI instance

   bash-4.4# **cp truststore.onap.client.jks truststore.onap.client.tapi.jks**

   Make a copy for the virtuoranc MSA instance. Please do not use any other names for these files.

   bash-4.4# **cp truststore.onap.client.jks truststore.onap.client.msa.jks**

   bash-4.4# ls
   sdnc.p12 truststore.onap.client.msa.jks

   truststore.onap.client.jks truststore.openecomp.client.jks
   truststore.onap.client.jks_org
5. Get the Server Certificate using the below command and save it to a file (Make sure to not copy any extra space). Will have to do Ctrl+C after the command returns output. Make sure to remove the files tmp/vnc.crt, tmp/msa.crt and tmp/vnc1.crt if they already exist before copying the certificate.
   a) For Virtuora MSA instance: **openssl s_client -connect <controller IP>:<controller port>**

   root@demo-sdnc-sdnc-0:/opt/onap/sdnc/data/stores# **cat > /tmp/msa.crt**

   b) For Virtuora TAPI 1 Instance: **openssl s_client -connect <controller IP>:<controller port>**

   root@demo-sdnc-sdnc-0:/opt/onap/sdnc/data/stores# **cat > /tmp/vnc.crt**

c) For Virtuora TAPI 2 Instance: **openssl s_client -connect <controller IP>:<controller port>**

root@demo-sdnc-sdnc-0:/opt/onap/sdnc/data/stores# **cat /tmp/vnc1.crt**

6. Import the server certificate to the truststore and enter yes for 'Trust this certificate? [no]: ' when prompted.

   bash-4.4# **keytool -importcert -file /tmp/msa.crt -alias msa_key -keystore truststore.onap.client.msa.jks -storepass adminadmin**

   bash-4.4# **keytool -importcert -file /tmp/vnc.crt -alias vnc_key -keystore truststore.onap.client.tapi.jks -storepass adminadmin**

   bash-4.4# **keytool -importcert -file /tmp/vnc1.crt -alias vnc1_key -keystore truststore.onap.client.tapi.jks -storepass adminadmin**

   The output could be something like the following:

Command Output

```
Owner: CN=virtuoranc-57bdd8c4bf-t6g84, OU=FNC, O=Fujitsu Network Communications Inc, L=Richardson, ST=Texas,
C=US
Issuer: CN=virtuoranc-57bdd8c4bf-t6g84, OU=FNC, O=Fujitsu Network Communications Inc, L=Richardson, ST=Texas,
C=US
Serial number: 2e88f579
Valid from: Thu March 30 01:11:30 GMT 2020 until: Wed May 30 01:11:30 GMT 2020
Certificate fingerprints:
MD5: AA:BF:02:DB:EE:02:8E:B0:2D:3D:89:82:A9:1E:E4:59
SHA1: 2E:4D:6A:90:FB:6B:E1:B9:29:4F:C4:36:E0:AD:B7:50:60:37:57:ED
SHA256: AD:39:89:30:53:E2:F0:F3:FA:A0:38:BC:63:41:2F:92:6B:D0:14:DD:52:BF:C9:1B:E9:E2:BE:FA:46:C3:32:3E
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 70 AE 9D 70 11 32 3C 34 BB 34 D7 E8 0C F5 80 AE p..p.2<4.4......
0010: 9C 6E 59 7F .nY.
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
```

7. After the certificate is added, use keytool list to check if it exists

   bash-4.4# **keytool -list -keystore truststore.onap.client.msa.jks -storepass adminadmin | grep msa_key**
   msa_key, Mar 30, 2020, trustedCertEntry,

   bash-4.4# **keytool -list -keystore truststore.onap.client.tapi.jks -storepass adminadmin | grep vnc_key**
   vnc_key, Mar 30, 2020, trustedCertEntry,

   bash-4.4# **keytool -list -keystore truststore.onap.client.tapi.jks -storepass adminadmin | grep vnc1_key**
   vnc1_key, Mar 30, 2020, trustedCertEntry,

Guilin

Refer to SDNC-1420 solution in the comment portion if ONAP Guilin+ release is deployed.

REST API

Register Domain Controller

```
curl -X -k PUT https://{{WorkerIP}}:30233/aai/{{AAIVersion}}/external-system/esr-thirdparty-sdnc-list/esr-thirdparty-sdnc/<controller name>

Request body:

{
"thirdparty-sdnc-id": "<controller name>",
"location": "Core",
"product-name": "VirtuoraNetworkController",
"esr-system-info-list": {
"esr-system-info":[ {
"esr-system-info-id": "<controller name>",
"system-name": "<controller name>",
"type": "TAPI",
"vendor": "Fujitsu",
"version": "V2",
"service-url": "https://<controller IP>:<controller port>",
"user-name": "<user name>",
"password": "<password>",
"system-type": "Controller",
"protocol": "RESTAPI",
"ssl-cacert": "example-ssl-cacert-val-20589",
"ssl-insecure": "true",
"ip-address": "<controller IP>",
"port": "<controller port>",
"cloud-domain": "example-cloud-domain-val-76077",
"default-tenant": "example-default-tenant-val-71148",
"passive": "true",
"remote-path": "example-remotepath-val-5833",
"system-status": "example-system-status-val-23435"
}]
}
}
```

where "workerIP" is ONAP cluster external IP address.

This command will trigger the TAPI or OpenRoadM topology discovery DGs to be called in SDNC.

Inter-domain Link Provision

Refer to this link for Inter Domain Link Provision for OTN service creation across multiple domains managed by single ONAP.

MDONS Run Time

(Note: One reference for use case run time could be the MDONS demo video posted here. The demo link - 'April 8th demo' - is in Team Accomplishment session.)

OTN Service Creation

OTN Service Deletion

Close Loop

- MDONS Close Loop Approach in R7

MDONS Over Releases

Frankfurt

- Multi-domain Optical Network Services

Guilin

- MDONS Extension in R7

References