Manual steps for CCVPN Integration Testing

Overall test status and blocking Issue

Task	Detailed step	Status	Reason	Current Handler
Service Distribution	we are facing issue in the service distribution due to SDC parser issue	Blocked using manual step2 can bypass this error	SDC-1955	Michael Lando
Client integration	Request from UUI is failing due to service template parse problem, SDC is requesting more logs/info on the defect.	Blocked using manual step3 can bypass this error	SDC-1958	Michael Lando
Link Management	Link discovery otn domain and link, create link, delete the link within otn domain,Create link to external onap otn domain. Can't delete the link to external onap otn domain	Blocked using manual step 1 can bypass this error	~~AAI-1923~~ - Problem deleting due to EdgeRules in CCVPN usecase Casablanca CLOSED manual delete workaround is documented in comments of JIRA case	Keong
SOTNInfra Service	Creation is successful from SO to SDNC and controllers	Done		Seshu
SDWan Infra Service	Creation is successful from SO to SDNC and controllers	Done		Seshu
Site Service	Creation with workaround is In Progress	Done		Seshu
Closed Loop flow	Tested with mocked objects, ready for the testing with real service. Found the sdnc closedloop DG call was overwritten as part of this merge: With the manual path things have worked for now.	Done	SDNC-540: This is found in the E2E testing and found the reason that the actual changes made were overwitten by another merge, would be correcting this for Casablanca Maintenance release. POLICY-1356: As per wiki (Policy on OOM), push-policied.sh script is used to install policies. but it is observed that CCVPN policy is not added in this script. So merged CCVPN policy using POLICY-1356 JIRA ticket. but policy is pushed by using push-policy_casablanca.sh script during integration test.	Gaurav Vidya

1. Link Management , can't delete the link to external onap otn domain
   For the manual steps provided by A&AI tea, we should follow the steps as follow

   the only way to delete is using the forceDeleteTool shell script in the graphadmin container.
   First we will need to find the vertex id, you should be able to get the id by making the following GET request.

   GET /aai/v14/network/ext-aai-networks/ext-aai-network/createAndDelete/esr-system-info/test-esr-system-info-id-val-0?format=raw

```
{
"results": [
{
"id": "20624",
"node-type": "pserver",
"url": "/aai/v13/cloud-infrastructure/pservers/pserver/pserverid14503-as988q",
"properties": {
}
}
]
}
```

Same goes for the ext-aai-network:

```
GET /aai/v14/network/ext-aai-networks/ext-aai-network/createAndDelete?format=raw
```

Retrieve the id from the above output as that will be the vertex id that you want to remove.

Run the following command multiple times for both the esr-system-info and ext-aai-network:

```
kubectl exec -it $(kubectl get pods -lapp=aai-graphadmin -n onap --template 'range .items.metadata.name"\n"end' | head -1) -n onap gosu aaiadmin /opt/app/aai-graphadmin/scripts/forceDeleteTool.sh -action DELETE_NODE -userId YOUR_ID_ANY_VALUE -vertexId VERTEX_ID
```

From the above, remove the YOUR_ID_ANY_VALUE and VERTEX_ID with your info.

Service Distribution Error
To overcome the Service distribution, the SO catalog has to be populated with the model information of the services and resources.
a) Referring to the Csar that is generated in the SDC designed as per the detailes mentioned in the below link: CCVPN Service Design
b) Download the Csar from SDC thus generated.
c) copy the csar to SO sdc controller pod and bpmn pod

```
  kubectl -n onap get pod|grep so
  kubectl -n onap exec -it dev-so-so-sdc-controller-c949f5fbd-qhfbl  /bin/sh
```

- mkdir -p null/ASDC/1 (for sdc controller pod)
- mkdir -p ASDC/1 (for bpmn pod)

```
  kubectl -n onap cp service-Sdwanvpninfraservice-csar.csar  dev-so-so-bpmn-infra-58796498cf-6pzmz:null/ASDC/1/service-Sdwanvpninfraservice-csar.csar

  kubectl -n onap cp service-Sdwanvpninfraservice-csar.csar  dev-so-so-bpmn-infra-58796498cf-6pzmz:ASDC/1/service-Sdwanvpninfraservice-csar.csar
```

d) populate model information to SO db
The DB scripts can be seen in step5

The same would also be applicable for the integration of the client to create the service and get the details.
Currently the testing has been performed using the postman calls to the corresponding APIs.

1. Client Integration :
   To overcome the Service Template Parser issue, Usecase-UI read local csar file and create request to SO component.
   a) Make an available csar file for CCVPN use case.
   b) Replace uuid of available files with what existing in SDC.
   c) Put available csar files in UUI local path (/home/uui).

2. Service Instantiation
   There are 3 services that will be required to be designed and instantiated for the CCVPN usecase.
      a. SDWanInfra
      b. SOTNInfra and
      c. Site Service

One can find more details of the services and their relationship under CCVPN Provisioning.

Its recommended to use the SO dockers of 1.3.4 version to avoid the issue SO-1249.

SO catalog has to be populated with the model information of Site service, SDWanInfra Service and SOTNInfra services

Manually copy the csar in the following path /app/ASDC/1/ to avoid the issue of csar missing exception SO-1248.

```
kubectl cp mso-infrastructure-bpmn/<CSAR_Name.csar onap/dev-so-so-bpmn-infra-54db5cd955-h7f5s:/app/ASDC/1/<CSAR_Name>.csar
```

This will enable the bpmn infra docker to be able to read the csar files.

The user needs to call the SO REST API either through postman or UUI the rest would be the usual process of the service instantiation flows.

Note: For the Site Service the model details needs to be updated manually as the SDC parser exception (SDC-1255) cant allow SO to provide the required information to sdnc.

Here we had to insert the following data to the sdnc database to continue with the operation.

Example SO request and SO DB insertion

JSON sample request for creating the services from SO

release-casablan..._collection.json

Catalog DB insertion details example, based on the csar generated.

SQL_Insertion_scripts.txt

5. Manual steps in closed loop Scenario:

Following steps were undertaken for the closed loop testing.

a. Give controller ip, username and password, trust store and key store file in restconf collector collector.properties

b. Updated DMAAP ip in cambria.hosts in DmaapConfig.json in restconf collector and run restconf collector

c. Followed the steps provided in this link(https://wiki.onap.org/display/DW/Holmes+User+Guide+-+Casablanca#HolmesUserGuide-Casablanca-Configurations) to push CCVPN rules to holmes

d. Followed the steps provided in this link(https://wiki.onap.org/display/DW/ONAP+Policy+Framework%3A+Installation+of+Amsterdam+Controller+and+vCPE+Policy) as reference to push CCVPN policies to policy module and updated sdnc.url, username and password in environment(/opt/app/policy/config/controlloop.properties.environment)

As per wiki (Policy on OOM), push-policied.sh script is used to install policies. but I observed that CCVPN policy is not added in this script. So merged CCVPN policy using POLICY-1356 JIRA ticket. but policy is pushed by using push-policy_casablanca.sh script during integration test.

It is found that the changes made were overwritten and hence had to patch the DG manually. This will be tracked by the JIRA SDNC-540.