

# Controller Design Studio - Instantiation Testing Strategy

## Assumption:

- That the regression test suite will cover the existing use case for vFW instantiation using the preload mechanism.
- The CDS initiative will add a new test suite for testing the vFW instantiation use case with the automated assignment mechanism using the self-service flow integrated with CDS for Casablanca Release.

## Action Items:

- Yang to check the existing robot script.
- Yang to enhance the robot script for the option 2 use case if SO code is stable. Else we will focus on Option 1.
- Yuriy to follow on the action items for the SO Payload, GR-API Payload and Ingestion of Tosca and Blueprint information.
- Yuriy to follow up with Brian on the SB07 lab setup. Are the minimal set of component available for option 2?
  - [SDC, DMAAP/UEB, SO, Policy Manager, SDN-C , A&A] ONAP OOM instance is instantiated
- Yang to estimate the work effort required for the vFW instantiation use case.
  - Based on the effort Yang will might request additional effort and/or work with CDS team to help with the robot script implementation.
- Yuriy to create the JIRA US for the testing use case.

Steps	Description	Reference	NOTES
<b>Perquisite</b>			
1	[APP-C, SDN-C] ONAP OOM instance is instantiated		
2	Controller Blueprint mS instance is instantiated		
<b>Activate Controller Blueprint</b>			

1	Script to use the API to create a controller blueprint for vFW.		<ul style="list-style-type: none"> <li>Yuriy to verify with Brinda on the Payload API needed for the blueprint creation and activation.</li> </ul>
2	Script to Activate and Publish the Controller Blueprint		<ul style="list-style-type: none"> <li>Yuriy to verify with Brinda on the Payload API needed for the blueprint creation and activation.</li> </ul>
3	Script to Verify the Controller blueprint is persisted in the DB		
Steps	Description	Reference	NOTES
<b>Perquisite</b>			
1	[SDC, SO, DMAAP, Policy Manager, SDN-C , A&AI] ONAP OOM instance is instantiated		<ul style="list-style-type: none"> <li>Yuriy to follow up with Dan T/Lalena if its enough to have SDC/DMAAP and SDNC for ingestion of the tosca model?</li> <li>Brian Freeman will help with OPEN-LAB spin of the ONAP instance needed for the vFW use case. <ul style="list-style-type: none"> <li>OPEN LAB-- SB07</li> </ul> </li> </ul>
2	vFW Heat is preloaded with the SDNC Model information		
3	SDC TOSCA is ingested in SDNC for vFW		<ul style="list-style-type: none"> <li>Verify the VF-Model table has all the information related to the SDNC Model Name, Artifact Name and Model Version.</li> <li>Yuriy to work with Marco.</li> </ul>
4	Netbox mS instance is instantiated		
5	Naming-generation mS instance is instantiated		
6	Controller Blueprint mS instance is instantiated		
7	Naming Policy is preloaded in Policy Manager for vFW		
8	Controller Blueprint is preloaded in mS DB for vFW		
<b>Assign Create, &amp; Activate</b>			
1	Script that triggers Macro instantiation vFW request to SO using assign, create an activate building block.		<p>Yuriy to request SO team on the Payload for the MACRO Instantiation request.</p> <ul style="list-style-type: none"> <li>Verify the Resource Accumulator is triggered to return the controller blueprint to GR-API</li> <li>Verify the Naming mS generate the name per vFW Naming Policy for the VNF Module</li> <li>Verify the Resource Accumulator is triggered to return the controller blueprint to GR-API</li> <li>Verify the Naming mS generate the name per vFW Naming Policy for the VF Module</li> <li>Verify the Netbox Capability assigns an IP Address</li> <li>Verify A&amp;AI has VNFC inventory.</li> </ul>

2	Script that copies MD-SAL instance for vFW		
Steps	Description	Reference	NOTES
<b>Perquisite</b>			
1	[SDC, DMAAP/UEB, Policy Manager, SDN-C , A&AI] - ONAP OOM instance is instantiated		<ul style="list-style-type: none"> <li>Yuriy to follow up with Dan T/Lalena if its enough to have SDC/DMAAP and SDNC for ingestion of the tosca model?</li> <li>Brian Freeman will help with OPEN-LAB spin of the ONAP instance needed for the vFW use case.</li> <li>OPEN LAB-- SB07</li> </ul>
2	vFW Heat is preloaded with the SDNC Model information		
3	SDC TOSCA is ingested in SDNC for vFW		<ul style="list-style-type: none"> <li>Verify the VF-Model table has all the information related to the SDNC Model Name, Artifact Name and Model Version.</li> <li>Yuriy to work with Marco.</li> </ul>
4	Netbox mS instance is instantiated		
5	Naming-generation mS instance is instantiated		
6	Controller Blueprint mS instance is instantiated		
7	Naming Policy is preloaded in Policy Manager for vFW		
8	Controller Blueprint is preloaded in mS DB for vFW		<ul style="list-style-type: none"> <li>Yuriy to check with Brinda as part of the mS spin up if we can preload the Controller Blueprint for vFW use case.</li> </ul>
9	Pre-create A&AI object for service , generic VNF and VF Module.		
<b>Assignment</b>			
1	Script that triggers GR-API for Service Assign		
2	Script that triggers GR-API for VNF Assign		
3	Script that triggers GR-API for VF Module Assign		<ul style="list-style-type: none"> <li>Verify the Resource Accumulator is triggered to return the controller blueprint to GR-API</li> <li>Verify the Naming mS generate the name per vFW Naming Policy for the VNF Module</li> </ul>
4	Script that triggers copy of MD-SAL Config tree for VNF and VF Module for vFW		<ul style="list-style-type: none"> <li>Verify the Resource Accumulator is triggered to return the controller blueprint to GR-API</li> <li>Verify the Naming mS generate the name per vFW Naming Policy for the VF Module</li> <li>Verify the Netbox Capability assigns an IP Address</li> <li>Verify A&amp;AI has VNFC inventory.</li> </ul>
<b>Activate</b>			
1	Script that triggers GR-API for VF Module Activate		
2	Script that triggers GR-API for VNF Module Activate		

3	Script that triggers GR-API for Service Module Activate		
4	Script that triggers copy of MD-SAL Config tree for VNF and VF Module for vFW		

Steps	Description	Reference	NOTES
<b>Perquisite</b>			
1	[Policy Manager, SDN-C , A&AI] - ONAP OOM instance is instantiated		<ul style="list-style-type: none"> <li>Yuriy to follow up with Dan T/Lalena if its enough to have SDC/DMAAP and SDNC for ingestion of the toasca model?</li> <li>Brian Freeman will help with OPEN-LAB spin of the ONAP instance needed for the vFW use case.</li> <li>OPEN LAB-- SB07</li> </ul>
2	SDC TOSCA is ingested in SDNC for vFW		<ul style="list-style-type: none"> <li>Verify the VF-Model table has all the information related to the SDNC Model Name, Artifact Name and Model Version.</li> <li>Yuriy to work with Marco.</li> </ul>
3	Netbox mS instance is instantiated		
4	Naming-generation mS instance is instantiated		
5	Controller Blueprint mS instance is instantiated		
6	Naming Policy is preloaded in Policy Manager for vFW		
7	Controller Blueprint is preloaded in mS DB for vFW		<ul style="list-style-type: none"> <li>Yuriy to check with Brinda as part of the mS spin up if we can preload the Controller Blueprint for vFW use case.</li> </ul>
8	Pre-create A&AI object for service , generic VNF and VF Module.		
<b>Assignment</b>			
1	Script that triggers GR-API for Service Assign		
2	Script that triggers GR-API for VNF Assign		
3	Script that triggers GR-API for VF Module Assign		<ul style="list-style-type: none"> <li>Verify the Resource Accumulator is triggered to return the controller blueprint to GR-API</li> <li>Verify the Naming mS generate the name per vFW Naming Policy for the VNF Module</li> </ul>
4	Script that triggers copy of MD-SAL Config tree for VNF and VF Module for vFW		<ul style="list-style-type: none"> <li>Verify the Resource Accumulator is triggered to return the controller blueprint to GR-API</li> <li>Verify the Naming mS generate the name per vFW Naming Policy for the VF Module</li> <li>Verify the Netbox Capability assigns an IP Address</li> <li>Verify A&amp;AI has VNFC inventory.</li> </ul>
<b>Activate</b>			
1	Script that triggers GR-API for VF Module Activate		
2	Script that triggers GR-API for VNF Module Activate		
3	Script that triggers GR-API for Service Module Activate		
4	Script that triggers copy of MD-SAL Config tree for VNF and VF Module for vFW		


Controller Blueprint Artifact and DDs for testing vFW Use Case: <https://gerrit.onap.org/r/#/c/69421/>

**Alexis to Retest**

Steps	Description	Status	NOTEs
1	Robot Script to Trigger SO for Macro Instantiation Request	Complete	
2	SO Triggers A&AI for Service Creation	Complete	
3	SO Trigger SDNC for Service Assign	Complete	
4	SO Triggers A&AI for VNF Object Creation	Complete	
5	SO Triggers SDNC for VNF Assign	Complete	
6	SDNC Triggers Resource Accumulator mS	Complete	
7	SDNC Trigger Naming mS	Complete	
8	Naming mS retrieves Policy Instance from Policy Manager	Complete	
9	Naming mS Generates the name and returns the output to SDNC	Complete	
10	SDNC Triggers Netbox IP Capability	NA	
11	Netbox Assign IP and Returns the values to SDNC	NA	
12	SDNC Triggers A&AI Updates	SKIPPED	
13	SDNC returns status to SO	Completed	
	NOTE Steps 14 to 21 will be repeated for base and each add on vf module (base template, vFW, vPG, vSN)	Pending	
14	SO Triggers SDNC with VF Module assign		base_template- Pass, vFW - Pass  vSNK-naming policy is missing vPG- MS call for get Policy is failing Policy
15	SDNC Triggers Resource Accumulator mS	Completed	
16	SDNC Trigger Naming mS	Completed	
17	Naming mS retrieves Policy Instance from Policy Manager	Pending	vSNK-naming policy is missing  vPG- MS call for get Policy is failing Policy
18	SDNC Triggers Netbox IP Capability	Completed	
19	Netbox Assign IP and Returns the values to SDNC	Completed	
20	SDNC Triggers A&AI Updates	SKIPPED	
21	SDNC returns status to SO	Failed	SO is not handling fallout correctly, Although SDNC return error code 500 so triggers SDNC for VF Module Assignment. JIRA created and assigned to Steve S.

	DescriptionDesc	Environment
--	-----------------	-------------

Yang	<p>1. Testing in SB05 Lab Instance 2. Current testing is failing in SDNC. Checking on the logs for failure</p> <p>Robot Script integration with SDC for naming-policy update, nf role etc.. <b>[Completed]</b></p> <p>Robot Script integration with policy manager for storing the naming recipe as part of the onap instantiation. <b>[Completed]</b></p> <p>Robot Script to update to pass the onap_private_subnet_id at the vnf level. <b>[Completed]</b></p> <p>protected_private_subnet_id</p> <p>Still Issue with the current environment. Being worked.</p> <p><b>SDC Distribution is failing. No Release docker that cant control the version. Blocking CDS testing. <a href="#">SDC Distribution has been patched and fixed. Yang is working on the distribution of the latest vFW csar package after the SB05 reinstall.</a></b></p> <p><b>Do we want to resolve via input or env file.</b></p> <ul style="list-style-type: none"> <li>• The only drawback for env file is that it has to be reworked for every onap instance that is created.</li> <li>• Yang to follow up with Marco.</li> <li>• Does the value have to be unique for ONAP instance.</li> <li>• Casablanca focus the testing with SO starting point.</li> <li>• Dublin release focus the testing with VID as the starting point.</li> </ul> <p><b>Question:</b></p> <p>Can the Macro instantiation use case be triggered with VID. Yuriy: Yes it should be covered. Yang to share the logs for the VID Failure.</p>	Integration Testing
------	---	---------------------

Alexis	<p>1. Fixing issue in DD, Blueprint, and Velocity. (Auto Generated for Blueprint should be a target for Dublin)</p> <p>2. High Level status:</p> <ul style="list-style-type: none"> <li>○ Service Assign Passed</li> <li>○ VNF Assign Passed</li> <li>○ Base Assign Passed</li> <li>○ vFW Module Assign Passed</li> <li>○ vPG Module Assign Passed</li> <li>○ vSN Module Assign Passed</li> </ul> <p><b>A&amp;AI Capability for putting in updates for the generic vnf object and vf module objects. Pat is working on the issue and will submit by 10/12/18.</b></p> <p><b>Alexis: Retest the use case for manual naming assignment.</b></p> <p><b>NOTE:</b></p> <p>In General all the capabilities are passing for naming-generation capability, netbox capability, all the MD-SAL and DD resource retrieval is passing.</p> <p>3. Completed all the Assignment</p> <p><b>4. Now SO is trigger the Create-Activate BB but its failing.</b></p> <p><a href="#">SO-1107</a> - Max Benjamin shall upstream the SO fix. [Completed]</p> <p><b>11/16 - Latest SO JIRA:</b>  <a href="#">SO-1144</a> - [SO] Assign Service fails with Null Pointer Expection <span>CLOSED</span></p> <p>SDNC: SO failed b/c A&amp;AI didn't have the self-links. SDNC didn't update the self links (A&amp;AI ) as part of the assignment capability.</p> <p>SO issue: the cloudOwner is being hardcode as att-aic. Steve S is looking into the fix.</p> <p>SO is retrying the failed request over and over again in the loop. – Steve S has said a fix is ready and available.</p> <p>SO Issue: Failing need to be check why. Latest issue.</p> <p><b><u>Plan is to have Alexis start testing in SB05. 10/16.</u></b></p> <p><b>October 23, 2018:</b></p> <p>SO to OOF pairwise testing issue is blocking instantiation use cases progress. Possible workaround is to use OOF emulator that needs to be integrated with SO.</p> <p>OOF implementation in GR-API maybe incomplete. <b>Need meeting with SO team to make progress.</b></p> <p>OOF implementation with the VNF-API is coupled to the vCPE use case.</p> <p>Moved the cloud param from VF Module to VNF level such as image_name, key_name, and pub_key for shared resourced.</p> <p>Resolved a bug in the heat that overriding the key_name value provide by ONAP. Marco committed and merge the changes.</p>	Local Testing
--------	--	---------------

Retrieve the Model Information from SO Database. Update the get command with the name.

```
curl -X GET \
  'http://10.195.197.73:31883/ecom/mso/catalog/v2/serviceVnfs?serviceName=vFW_CDS' \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'X-FromAppId: AAI' \
  -H 'X-TransactionId: get_aai_subscr'
```

SO Payload:

```
{
  "requestDetails": {
    "subscriberInfo": {
      "globalSubscriberId": "Demonstration"
    },
    "requestInfo": {
      "suppressRollback": true,
      "productFamilyId": "a9a77d5a-123e-4ca2-9eb9-0b015d2ee0fb",
      "requestorId": "adt",
      "instanceName": "test-1234",
      "source": "VID"
    },
    "cloudConfiguration": {
      "lcpCloudRegionId": "RegionOne",
      "tenantId": "7320ec4a5b9d4589ba7c4412ccfd290f"
    },
    "requestParameters": {
      "subscriptionServiceType": "vFW",
      "userParams": [
        {
          "service": {
            "instanceParams": [],
            "instanceName": "Service_E2E_vFW",
            "resources": {
              "vnfs": [
                {
                  "modelInfo": {
                    "modelName": "d26a74bc-3fdb-4fd2-ad55",
                    "modelVersionId": "69a7d967-fe68-49d2-ad6b-fc40f5f5b31c",
                    "modelInvariantUuid": "c397d40f-cd2a-4b19-add6-e12ea9ecf4ab",
                    "modelVersion": "2.0",
                    "modelCustomizationId": "2cc242bc-ff38-4bac-9650-5felb54f5aeb",
                    "modelInstanceName": "d26a74bc-3fdb-4fd2-ad55 0"
                  },
                  "cloudConfiguration": {
                    "lcpCloudRegionId": "RegionOne",
                    "tenantId": "demo"
                  },
                  "platform": {
                    "platformName": "test"
                  },
                  "lineOfBusiness": {
                    "lineOfBusinessName": "someValue"
                  },
                  "productFamilyId": "a9a77d5a-123e-4ca2-9eb9-0b015d2ee0fb",
                  "instanceName": "vFirewall 0",
                  "instanceParams": [],
                  "vfModules": [
                    {
```



```

        "modelInfo": {
            "modelName": "D26a74bc3fdb4fd2Ad55..base_template..module-0",
            "modelVersionId": "e3c095c3-a620-4828-a689-7132936b90ff",
            "modelInvariantUuid": "ac36a6de-0f92-4ee2-8b39-1b0dba0058f5",
            "modelVersion": "1",
            "modelCustomizationId": "1f8ff6cf-640b-432f-9dec-1352b7e1f862"
        },
        "instanceName": "vfirewall10..Vfirewall..base_template..module-0",
        "instanceParams": [
            {}
        ]
    },
    {
        "modelInfo": {
            "modelName": "D26a74bc3fdb4fd2Ad55..vfw..module-3",
            "modelVersionId": "086223d6-352c-47d3-abb4-213711fc5188",
            "modelInvariantUuid": "d9e4a4e0-c0d3-4bd3-89cc-c6b95af29f6d",
            "modelVersion": "1",
            "modelCustomizationId": "a6751179-1762-4771-8ccb-541516c53cf2"
        },
        "instanceName": "vfirewall10..Vfirewall..vfw..module-0",
        "instanceParams": [
            {}
        ]
    },
    {
        "modelInfo": {
            "modelName": "D26a74bc3fdb4fd2Ad55..vpg..module-2",
            "modelVersionId": "6a18492b-4c9d-47a5-a8eb-34b085db0eb0",
            "modelInvariantUuid": "74d0a390-c76f-42fd-ae5d-dc9bea9b79a5",
            "modelVersion": "1",
            "modelCustomizationId": "a505c501-0264-4835-821e-006d603691a9"
        },
        "instanceName": "vfirewall10..Vfirewall..vpg..module-0",
        "instanceParams": [
            {}
        ]
    },
    {
        "modelInfo": {
            "modelName": "D26a74bc3fdb4fd2Ad55..vsn..module-1",
            "modelVersionId": "37272aa7-aa29-4772-a1f9-2a67c5860df6",
            "modelInvariantUuid": "ba2e53f1-73ca-40c2-97fe-albc74c278ba",
            "modelVersion": "1",
            "modelCustomizationId": "8b59f644-flac-4a57-8ddb-c6680bc5a22c"
        },
        "instanceName": "vfirewall10..Vfirewall..vsn..module-0",
        "instanceParams": [
            {}
        ]
    }
]
},
{
    "modelInfo": {
        "modelVersion": "1.0",
        "modelVersionId": "88da85c-d9e8-4f73-b837-3a72a431622b",
        "modelInvariantId": "7242c3f8-c5d8-4069-be3c-0db8bf4caa4d",
        "modelName": "vFW_CDS",
        "modelType": "service"
    }
}
},
{
    "aLaCarte": false
},
{
    "project": {
        "projectName": "Project-Demonstration"
    },
    "owningEntity": {

```

```
    "owningEntityId": "33a8b609-1cfe-4d19-8dc2-5b95b921dd1e",  
    "owningEntityName": "OE-Demonstration"  
  },  
  "modelInfo": {  
    "modelVersion": "1.0",  
    "modelVersionId": "88da85c-d9e8-4f73-b837-3a72a431622b",  
    "modelInvariantId": "7242c3f8-c5d8-4069-be3c-0db8bf4caa4d",  
    "modelName": "vFW_CDS",  
    "modelType": "service"  
  }  
}
```