

Tutorial: Creating a Service Instance from a Design Model

- [TODO:](#)
- [Creating a Service Instance](#)
 - [Run Robot demo.sh init](#)
- [Deploy Service Instance in VID](#)
- [Run Robot demo.sh preload of DemoModule](#)
 - [Preload Flow](#)
- [Add a VF Module in VID](#)
 - [Option 1: REST call to MSO](#)
 - [Option 2: VID GUI](#)
 - [Create VF Module - polling hangs - vFW VMs are created though](#)
 - [Watch VF VM stack creation](#)
- [Run Robot demo.sh appc on DemoModule to mount the Traffic Generator](#)
- [Control Loop Flows](#)
- [Error Handling](#)
 - [Handle MSO Failure on RAM Quota exceeded](#)
 - [Handle MSO Failure on Create VF Module](#)
 - [Handle outdated vFW \(201702xx\) zip causing Traffic Generation not to start](#)
 - [Fix: Use the 1.0.0 template in Nexus - or the updated one on this wiki](#)
 - [Issue:](#)

TODO:

20171120: Brian Freeman has commented on R1 changes - need to verify these in a live system before posting here

VNF preload is now part of VID in a checkbox - so we don't require the sdnc rest call as part of demo.sh preload

Robot in OOM is run in oom/kubernetes/robot now

Creating a Service Instance

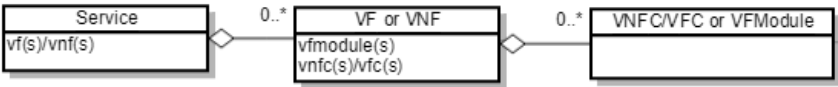
In this tutorial we show how to take a service design that has been distributed and create a running instance of the service.

SDNC preload fragment

```
"service-type": "11819dd6-6332-42bc-952c-1a19f8246663",
"vnf-name": "DemoModule2",

above is the vf-module (3 of 3 in the diagram below)
"vnf-type": "Vsp..base_vfw..module-0",
"generic-vnf-name": "vFWDemoVNF",

above is the vnf (2 of 3)
"generic-vnf-type": "vsp 0"
```



To simplify this we are going to use scripts (with some selenium robot scripts) to create the design, pre-load customer and network information, and orchestrate parts of the virtual firewall closed loop example. The following steps assume that you have completed and understand basic concepts from the [setting up the platform](#) and [using the portal](#) tutorials.

Let's start by finding the IP Address of vm1-robot in the Rackspace list of servers. Use this vm1-robot IP address, your Rackspace private key, and the PuTTY client to login to vm1-robot as root.

Note: The current default LCP Region is IAD - to use DFW switch the example zip in the last section - currently though we have hardcoding that must be fixed: [DOC-6](#) - Getting issue details... STATUS

```
osx$ ssh-add onap_rsa
```

```
osx$ ssh root@104.130.170.232
```

Run Robot demo.sh init

At the command prompt type

```
root@vm1-robot:~# cd /opt
```

```
root@vm1-robot:/opt# ./demo.sh init
```

Wait for all steps to complete (will take 60-120 sec) as shown below

```
root@vml-robot: /opt
Using username "root".
Authenticating with public key "imported-openssh-key" from agent
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

32 packages can be updated.
19 updates are security updates.

*** System restart required ***
Last login: Fri Mar  3 21:13:55 2017 from 144.160.5.25
root@vml-robot:~# cd /opt
root@vml-robot:/opt# ./demo.sh init
Starting Xvfb on display :89 with res 1280x1024x24
Executing robot tests at log level TRACE
=====
OpenECOMP ETE
=====
OpenECOMP ETE.Robot
=====
OpenECOMP ETE.Robot.Testsuites
=====
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestration Test ...
=====
Initialize Customer And Models | PASS |
-----
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestrat... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot.Testsuites | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
Output:  /share/logs/demo/InitDemo/output.xml
Log:     /share/logs/demo/InitDemo/log.html
Report:  /share/logs/demo/InitDemo/report.html
root@vml-robot:/opt#
```

If you want to see the details of what ran, you can open report.html in a browser. (located within openecompete_container docker container)

```
root@vm1-robot:/opt# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f99954f00ab2	nexus3.onap.org:10001/openecomp/testsuite:1.0-STAGING-latest	"lighttpd -D -f /e..."	19 hours ago	Up 19 hours	0.0.0.0:88->88/tcp	openecomp_container

```
root@vm1-robot:/opt# docker exec -it openecomp_container bash
```

```
root@f99954f00ab2:/# cat /share/logs/demo/InitDemo/
```

```
log.html  output.xml  report.html
```

Deploy Service Instance in VID

From the ONAP portal, login to the VID application using demo user, browse to locate the demo SDC Service Models, and Deploy an instance of the service you created - not the pre-populated demoVFW.

Action	UUID	Invariant UUID	Name	Version	Category	Distribution Status	Last Updated By	Tosca Model
Deploy	e3cce766-d901-41e8-aa8d-1f890fa6c151	4f3c45f0-da9e-4774-926d-a92407942aca	demoVFW	1.0	Network L1-3	DISTRIBUTED	jm0007	
Deploy	76229bf3-28d0-45f3-92cd-9ecb59107ef4	95105c19-615f-435b-ad91-80d933f84435	demoVLB	1.0	Network L1-3	DISTRIBUTED	jm0007	
Deploy	12171814-7a33-46eb-b00d-df5749744e53	35d01ae9-be5a-42e8-b78f-43d3639a536d	Service	1.0	Network L4+	DISTRIBUTED	jm0007	

~~(Note: deploy your "service" above - not demoVFW or demoVLB - these 2 are leftover pre-population artifacts of the init script and will be removed)~~

Use the generated demoVFW above (you don't need to onboard/distribute your own)

Fill in the information (Instance Name=DemoInstance, Demonstration, vFW) for a Service Instance as shown below and press Confirm.

Create Service Instance

Service Name: service
Service Invariant UUID: dd322b17-d21f-4ec7-9ee6-6e76736952e7
Service Version: 1.0
Service UUID: 55cf72d4-ec11-41cc-8001-48fe6b1a4ae9
Service Description: service
Service Category: Network L4+

User Provided Data (* indicates required field)

Instance Name: *	<input type="text" value="DemoInstance"/>
Subscriber Name: *	<input type="text" value="Select Subscriber Name"/>
Service Type: *	<input type="text"/>
Suppress Rollback on Failure:	<input type="checkbox"/>

Enter Data and **Confirm** to
Create **Service Instance**

Cancel to Return to Previous Page.
Data entered will be lost

Confirm

Cancel

adjust above for project and owning entity and vFWCL/vSNK

Wait for a response and close the window

Status: COMPLETE - Service Instance has been created successfully.

100 %

```
05/26/17 17:35:46 HTTP Status: OK (200)
{
  "request": {
    "requestId": "7c020f8c-8f6f-4e7a-ac3a-cd426473a7f5",
    "startTime": "Thu, 25 May 2017 21:35:34 GMT",
    "requestScope": "service",
    "requestType": "createInstance",
    "requestDetails": {
      "modelInfo": {
        "modelCustomizationName": null,
        "modelInvariantId": "35d01ae9-be5a-42e8-b78f-43d3639a536d",
        "modelType": "service",
        "modelNameVersionId": "12171814-7a33-46eb-b00d-df5749744e53",
        "modelName": "Service",
        "modelVersion": "1.0"
      },
      "requestInfo": {
        "billingAccountNumber": null,
        "callbackUrl": null,
        "correlator": null,
        "orderNumber": null,
        "productFamilyId": null,
        "orderVersion": null,
        "source": "VID",
        "instanceName": "DemoInstance",
        "suppressRollback": false
      }
    }
  }
}
```

Close

You should now see a service instance displayed.

SUBSCRIBER: Demonstration

SERVICE TYPE: vFW

SERVICE INSTANCE ID: 54260621-d9d7-4ffc-b73d-513c0084c228

Service Instance Name: DemoInstance

SERVICE INSTANCE: DemoInstance

?

Add VNF

×

Add a Virtual Network Function under the Service Instance in VID

Add a VNF using the drop down button, complete, and enter the following information. The tenant and LCP region drop down choices may be different for your Rackspace account. Both IAD and DFW support heat templates.

SUBSCRIBER: Demonstration

SERVICE TYPE: vFW

SERVICE INSTANCE ID: 54260621-d9d7-4ffc-b73d-513c0084c228

Service Instance Name: DemoInstance

SERVICE INSTANCE: DemoInstance

?

Add VNF

×

VSP

Create Virtual Network Function

Service Name: Service
Subscriber Name: Demonstration
Service Instance Name: DemoInstance
Model Name: VSP
Model Invariant UUID: 65c82ea8-1ee2-4eba-bcac-63dc44db36ee
Model Version: 1.0
Model UUID: 3c94372e-2563-48ef-9bcf-bbd6c5fccd5d

User Provided Data ( indicates required field)

Instance Name: *	<input type="text" value="DemoVNF"/>
Product Family: *	<input type="text" value="vFW"/>
LCP Region: *	<input type="text" value="IAD"/>
Tenant: *	<input type="text" value="1035199"/>
Suppress Rollback on Failure:	<input type="checkbox"/>

Enter Data and **Confirm** to
Create **Virtual Network Function**

Cancel to Return to Previous Page.

Data entered will be lost

Confirm

Cancel

Wait for and close the response window.

Status: COMPLETE - Vnf has been created successfully.

100 %

```
05/26/17 17:37:43 HTTP Status: OK (200)
{
  "request": {
    "requestId": "9ac35e07-6bac-4fa1-86fd-66cfbc4eb29a",
    "startTime": "Thu, 25 May 2017 21:37:32 GMT",
    "requestScope": "vnf",
    "requestType": "createInstance",
    "requestDetails": {
      "modelInfo": {
        "modelCustomizationName": "VSP 1",
        "modelInvariantId": "65c82ea8-1ee2-4eba-bcac-63dc44db36ee",
        "modelType": "vnf",
        "modelNameVersionId": "3c94372e-2563-48ef-9bcf-bbd6c5fccd5d",
        "modelName": "VSP",
        "modelVersion": "1.0"
      },
      "requestInfo": {
        "billingAccountNumber": null,
        "callbackUrl": null,
        "correlator": null,
        "orderNumber": null,
        "productFamilyId": "cc772cc8-e04b-49c6-9313-94d48c1b2df1",
        "orderVersion": null,
        "source": "VID",
        "instanceName": "DemoVNF",
        "suppressRollback": false
      }
    }
  },
}
```

Close

Run Robot demo.sh preload of DemoModule

Return to the PuTTY/ssh window and type the command to load VNF configuration information

DO NOT Navigate from the Deploy page before adding the VF Module below - or you will need to search for it - then hit edit

```
./demo.sh preload <vnf_name> <module_name>
example
./demo.sh preload DemoVNF DemoModule
```

Wait for the results as shown below


```

root@vml-robot:/opt# ./demo.sh preload DemoVNF DemoModule
Starting Xvfb on display :89 with res 1280x1024x24
Executing robot tests at log level TRACE
=====
OpenECOMP ETE
=====
OpenECOMP ETE.Robot
=====
OpenECOMP ETE.Robot.Testsuites
=====
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestration Test ...
=====
Preload VNF | PASS |
-----
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestrat... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot.Testsuites | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
Output: /share/logs/demo/PreloadDemo/output.xml
Log: /share/logs/demo/PreloadDemo/log.html
Report: /share/logs/demo/PreloadDemo/report.html

```

For any error/debug logs looks into

```
/opt/eteshare/logs/demo/PreloadDemo/output.xml
```

Preload Flow

see overall [Tutorial: Verifying and Observing a deployed Service Instance#vFirewallFlow](#)

demo.sh calls runTags.sh in the docker container in robot - which runs the robot test framework scripts starting with PreLoad VNF in demo.robot

```

Preload VNF
  Preload User Model    ${VNF_NAME}    ${MODULE_NAME}

```

which calls demo_preload.robot (although it screen scrapes the Service ID previously from VID - in this call it does a rest call to VID to package up the vm modules list) and calls SDNC preload with these (the diagram needs a 25-2 for robot to VID as well for this sub-step)

```

Preload User Model
  Login To VID GUI
  ${vf_modules}=    Get Module Names from VID    ${invariantUUID}
  Preload Vnf    ${service_instance_id}    ${vnf_name}    ${vnf_type}    ${vf_module_name}    ${vf_modules}
${service}    demo

```

which calls sdngc_interface.robot (logs in and posts to sdnc/mobility/addVnfProfile)

```

Preload Vnf Profile
  Login To SDNGC Admin GUI
  Go To    ${SDNGC_ADMIN_VNF_PROFILE_URL}
  Click Button    xpath=//button[@data-target='#add_vnf_profile']
  Click Button    xpath=//button[contains(., 'Submit')]

```

which calls the post form

```
<form name="addForm" role="form" action="/mobility/addVnfProfile" method="POST">
```

<http://sdnc:8843/mobility/getVnfProfile>

which runs a backend DB insert operation on SDNC (calling the DB directly here is likely not advised, also there is no parameter checking on the resultant SQL, we should also be using an ORM framework)

```
/sdnc-oam/admportal/mobility.js

router.post('/addVnfNetwork', csp.checkAuth, function(req,
res){
  var sql = "INSERT INTO VNF_NETWORKS (vnf_type,
network_role) VALUES ("
+ "'" + req.body.nf_vnf_type + "'," + "'" + req.body.
nf_network_role + "'"");
  tasks.push( function(callback) { dbRoutes.executeSQL(sql,
req,res,callback); } );
});
```

Add a VF Module in VID

Option 1: REST call to MSO

POST to http://{{mso_ip}}:8080/ecom/mso/infra/serviceInstances/v2/<id>/vnfs/<id>/vfModules - see [UCA-20 OSS JAX-RS 2 Client](#)

Option 2: VID GUI

Add a VF Module using the drop down button.

The screenshot shows a web interface for managing service instances. At the top, there's a header bar with three sections: 'SUBSCRIBER: Demonstration', 'SERVICE TYPE: vFW', and 'SERVICE INSTANCE ID: 54260621-d9d7-4ffc-b73d-513c0084c228'. Below this, a text field displays 'Service Instance Name: DemoInstance'. A main content area has a light blue background. It features a dropdown menu labeled 'SERVICE INSTANCE: DemoInstance' with a green checkmark icon. To the right of the dropdown is a blue information icon and a green button labeled 'Add VNF'. Below this, a status bar shows 'VNF: DemoVNF | TYPE: Service/VSP 1 | ORCH STATUS: Created'. To the right of the status bar are a blue information icon, a green button labeled 'Add Volume Group', a green button labeled 'Add VF-Module', and a red button with a white 'X' icon.

Fill in information for the VF module (service name = Service) and confirm.

Create VF Module

Service Name: Service
Subscriber Name: Demonstration
Service Instance Name: DemoInstance
Model Name: Vsp..base_vfw..module-0
Model Invariant UUID: 0f6bfe93-bfc5-45aa-8fbb-83e5c4c4c82d
Model Version: 1
Model UUID: 8aed7a60-1542-4678-8c69-40708b0a8a27

User Provided Data (* indicates required field)

Instance Name: *	<input type="text" value="DemoModule"/>
LCP Region: *	<input type="text" value="IAD"/>
Tenant: *	<input type="text" value="1035199"/>
Suppress Rollback on Failure:	<input type="checkbox"/>

Enter Data and **Confirm** to
Create **VF Module**

Cancel to Return to Previous Page.
Data entered will be lost

Confirm

Cancel

Create VF Module - polling hangs - vFW VMs are created though

Maximum number of poll attempts exceeded

Eventually you will see a (red-herring) poll timeout - we need to adjust the wait time and # of retries here - anyway the 3 VM's are up (with pings but not necessarily with 200 health checks on the processes)

see [UCA-19](#) - Getting issue details... **STATUS**

Status: Error

Maximum number of poll attempts exceeded

```
05/26/17 17:43:50 HTTP Status: OK (200)
{
  "request": {
    "requestId": "c79c54e6-d8e5-42aa-8e13-c06228bf1368",
    "startTime": "Thu, 25 May 2017 21:41:58 GMT",
    "requestScope": "vModule",
    "requestType": "createInstance",
    "requestDetails": {
      "modelInfo": {
        "modelCustomizationName": null,
        "modelInvariantId": "0f6bfe93-bfc5-45aa-8fbb-83e5c4c4c82d",
        "modelVersion": "1"
      }
    }
  }
}
```

Cloud Servers

Northern Virginia (AD)

Filter Servers

STATUS
Active (20)

TYPE
Next Generation (20)

IMAGE

Create Server Create Stack Delete Server

Search 20 servers...

Name	Tags	IP Address	Monitoring
demoVFW01fw		172.99.75.45	
demoVFW01pgn		172.99.75.46	
demoVFW01lank		172.99.69.13	
vm1-aal		104.239.249.72	

Select close, and later cancel (for now) - as the VMs are actually up

Status: Error



Maximum number of poll attempts exceeded

```
05/26/17 17:43:50 HTTP Status: OK (200)
{
  "request": {
    "requestId": "c79c54e6-d8e5-42aa-8e13-c06228bf1368",
    "startTime": "Thu, 25 May 2017 21:41:58 GMT",
    "requestScope": "vfModule",
    "requestType": "createInstance",
    "requestDetails": {
      "modelInfo": {
        "modelCustomizationName": null,
        "modelInvariantId": "0f6bfe93-bfc5-45aa-8fbb-83e5c4c4c82d",
        "modelType": "vfModule",
        "modelNameVersionId": "8aed7a60-1542-4678-8c69-40708b0a8a27",
        "modelName": "Vsp..base_vfw..module-0",
        "modelVersion": "1"
      },
      "requestInfo": {
        "billingAccountNumber": null,
        "callbackUrl": null,
        "correlator": null,
        "orderNumber": null,
        "productFamilyId": null,
        "orderVersion": null,
        "source": "VID",
        "instanceName": "DemoModule",
        "suppressRollback": false
      }
    }
  },
}
```

Close

For now cancel the Create VF Module dialog (the VMs were created)

Create VF Module

Service Name: Service
Subscriber Name: Demonstration
Service Instance Name: DemoInstance
Model Name: Vsp..base_vfw..module-0
Model Invariant UUID: 0f6bfe93-bfc5-45aa-8fbb-83e5c4c4c82d
Model Version: 1
Model UUID: 8aed7a60-1542-4678-8c69-40708b0a8a27

User Provided Data (* indicates required field)


Instance Name: *	<input type="text" value="DemoModule"/>
LCP Region: *	<input type="text" value="IAD"/>
Tenant: *	<input type="text" value="1035199"/>
Suppress Rollback on Failure:	<input type="checkbox"/>

Enter Data and **Confirm** to
Create **VF Module**

Cancel to Return to Previous Page.
Data entered will be lost

Watch VF VM stack creation

Watch as the 3 VMs for the VF start to come up on Rackspace (dialog is still up)



Rackspace Cloud

More Products

Supportobrienlabs

DashboardServersOrchestrationNetworkingStorageDatabasesBackups

Cloud Servers

Northern Virginia (IAD)

Filter Servers

STATUS

Active (17)

Building (3)

TYPE

Next Generation (20)

IMAGE

Ubuntu 14.04 LTS (Trus... (13)

Booted From Volume (4)

Ubuntu 16.04 LTS (Xeni... (3)

FLAVOR

4 GB General Purpose v1 (7)

8 GB Performance (5)

15 GB Compute v1 (3)

2 GB General Purpose v1 (2)

15 GB I/O v1 (1)

more

Create ServerCreate StackDelete Server

Search 20 servers...

	Name	Tags	IP Address	Monitoring
	demofw01fwl		172.99.75.45	
	demofw01pgn		172.99.75.46	
	demofw01snk		172.99.69.13	
	vm1-aai		104.239.249.72	
	vm1-appc		162.242.218.203	
	vm1-dcae-controller		146.20.110.39	
	vm1-dns-server		104.130.170.150	
	vm1-message-router		162.209.124.181	
	vm1-mso		104.130.170.156	
	vm1-policy		104.239.249.17	
	vm1-portal		104.130.31.25	
	vm1-robot		104.130.170.237	
	vm1-sdc		104.239.249.15	
	vm1-sdnc		104.130.170.232	
	vm1-vid		104.130.170.142	
	zldciad4vicdap00		104.239.168.61	
	zldciad4vicdap01		162.242.235.70	
	zldciad4vicdap02		104.130.239.90	
	zldciad4vicoll00		146.20.110.155	
	zldciad4vipstg00		146.20.110.226	

Note: Openstack users with RegionOne may see failures here. Looks into the below ticker to update MSO docker container /shared/mso-docker.json file with RegionOne settings. For logs use

```
docker logs -f testlab_mso_1
```

Browse our new vFW service

▲ Browse SDC Service Models

Admin

Search for Existing Service Instances

▲ Browse SDC Service Models

Admin

Filter:

View/Edit	Global Customer ID	Subscriber Name	Service Type	Service Instance Name	Service Instance ID
View/Edit	Demonstration	Demonstration	vFW	Demolnstance	54260621-d9d7-4ffc-b73d-513c0084c228

View/Edit Service Instance

SUBSCRIBER: Demonstration

SERVICE TYPE: vFW

SERVICE INSTANCE ID: 54260621-d9d7-4ffc-b73d-513c0084c228

Service Instance Name: Demolnstance

▼ SERVICE INSTANCE: Demolnstance

Info

Add VNF

▼ VNF: DemoVNF | TYPE: Service/VSP 1 | ORCH STATUS: Created

Info

Add Volume Group

Add VF-Module

VFMODULE: DemoModule | TYPE: vf-module | ORCH STATUS: active

Info

✕

Verify VNF Profile

create an account on SDNC <http://sdnc-ip:8843/signup>

login <http://sdnc-ip:8843/login>

Check VNF Profile in Profile menu

AdminPortal

SLA

Profiles

Preload Tools

User Admin

Logout

VNF Profile

Add VNF Profile

Showing 1 to 1 of 1 entries

File input

Choose File No file chosen

Choose a file to upload.

Upload File


*VNF_TYPE	AVAILABILITY_ZONE_COUNT	EQUIPMENT_ROLE	Action
Vsp..base_vfw..module-0	999	robot-ete-1	<div>Delete</div>

Previous

1

Next



Wait for the response and close the window as was done in prior steps. The VF Module creation can also be viewed as a stack in Rackspace as shown below.


Rackspace Cloud
More Products ▾
Support ▾
obrienlabs ▾

Dashboard
Servers
Orchestration
Networking
Storage
Databases
Backups
+ MANAGED INFRASTRUCTURE

Stacks

Create Stack
Delete Stack
Northern Virginia (IAD) ▾

	Name ▲	Type	Condition	Region
<input type="checkbox"/>	 DemoModule	(Not Specified)	Deployed on May 25, 2017 - 9:42 PM EDT	Northern Virginia (IAD)
<input type="checkbox"/>	 ONAP24	(Not Specified)	Deployed on May 25, 2017 - 5:33 PM EDT	Northern Virginia (IAD)

Run Robot demo.sh appc on DemoModule to mount the Traffic Generator

To complete the service instance we will run one more script that mounts the Traffic Generator on the Application Controller to enable policy driven configuration changes. Return to the PuTTY window, type the command and wait for the response as shown below.

```
./demo.sh appc DemoModule
```

To summarize: here are all 3 orchestration assistance runs (init, preload, appc) between interleaved Service, VNF and VF-Module UI actions - to summarize


```
root@vm1-robot:/opt# ./demo.sh init
Starting Xvfb on display :89 with res 1280x1024x24
Executing robot tests at log level TRACE
=====
OpenECOMP ETE
=====
OpenECOMP ETE.Robot
=====
OpenECOMP ETE.Robot.Testsuites
=====
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestration Test ...
=====
Initialize Customer And Models | PASS |
=====
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestrat... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot.Testsuites | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
Output: /share/logs/demo/InitDemo/output.xml
Log: /share/logs/demo/InitDemo/log.html
Report: /share/logs/demo/InitDemo/report.html
root@vm1-robot:/opt# ./demo.sh preload DemoVNF DemoModule
Starting Xvfb on display :89 with res 1280x1024x24
Executing robot tests at log level TRACE
=====
OpenECOMP ETE
=====
OpenECOMP ETE.Robot
=====
OpenECOMP ETE.Robot.Testsuites
=====
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestration Test ...
=====
Preload VNF | PASS |
=====
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestrat... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot.Testsuites | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
Output: /share/logs/demo/PreloadDemo/output.xml
Log: /share/logs/demo/PreloadDemo/log.html
Report: /share/logs/demo/PreloadDemo/report.html
root@vm1-robot:/opt# ./demo.sh appc DemoModule
Starting Xvfb on display :89 with res 1280x1024x24
Executing robot tests at log level TRACE
=====
OpenECOMP ETE
=====
```

```

OpenECOMP ETE.Robot
=====
OpenECOMP ETE.Robot.Testsuites
=====
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestration Test ...
=====
Create APPC Mount Point | PASS |
=====
OpenECOMP ETE.Robot.Testsuites.Demo :: Executes the VNF Orchestrat... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot.Testsuites | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE.Robot | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
OpenECOMP ETE | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====

```

see the vFW sink page on the snk VM - to view traffic generation stats

demofw01snk

172.99.69.13





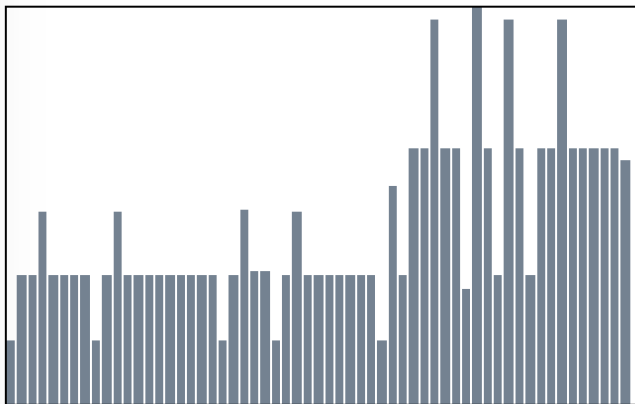
.245.148:667

[Apps](#) [Amdocs](#) [ONAP w](#) [AWS](#) [Azure](#) [GCE](#) [obrienlabs](#) [Rackspace](#) [LucidCha](#)[darkstat 3.0.715](#) [graphs](#) [hosts](#) [homepage](#)

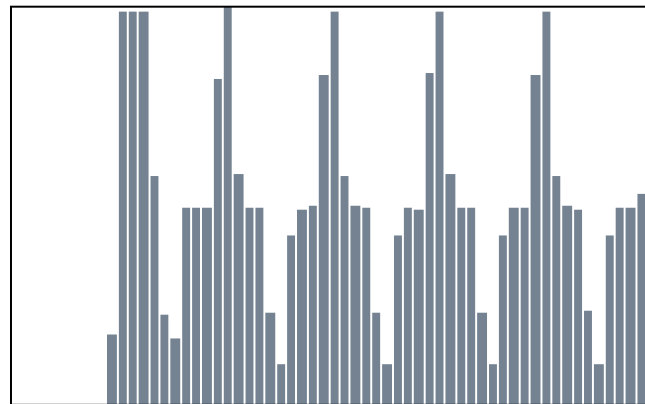
Graphs

Running for 49 mins, 53 secs, since 2017-06-08 18:47:49 UTC+0000.

Total 4,689,793 bytes, in 161,717 packets. (161,758 captured, 0 dropped)



last 60 seconds



last 60 minutes



last 24 hours




last 31 days

- automatic reload is:

Note: the overrides in the env are not picked up for the network/IPs as well - these are the sample py defaults - a JIRA is open

[UCA-17](#) - Getting issue details...

STATUS

Rackspace

Rackspace Cloud


More Products ▾

DashboardServersOrchestrationNetworkingStorageDatabasesBack

[◀ Back to Servers List](#)

CLOUD SERVER

demofwl01fwl



⚙️ Actions ▾

Server Details


Server Status

Active

ID

b9b8567a-9df8-434a-ba54-5dbe4d4e5a36

System Image

 Ubuntu 14.04 LTS (Trusty Tahr) (PVHVM) · [Rebuild...](#)

Flavor

4 GB General Purpose v1 · [Resize...](#)

SSH Keyname

vfw_keydemo ⓘ

Disk Configuration

Manual ⓘ

Monitoring Agent

Not installed, host checks unavailable · [How To Install Agent](#) ⓘ

Region

Northern Virginia (IAD)

Reverse DNS

0 Records · [Add Record...](#)

Created Date

May 25, 2017 - 5:42 PM EDT

Last Updated






May 25, 2017 - 5:43 PM EDT

Visualize

[📈](#) [Default Graphs](#) ⓘ

Networks

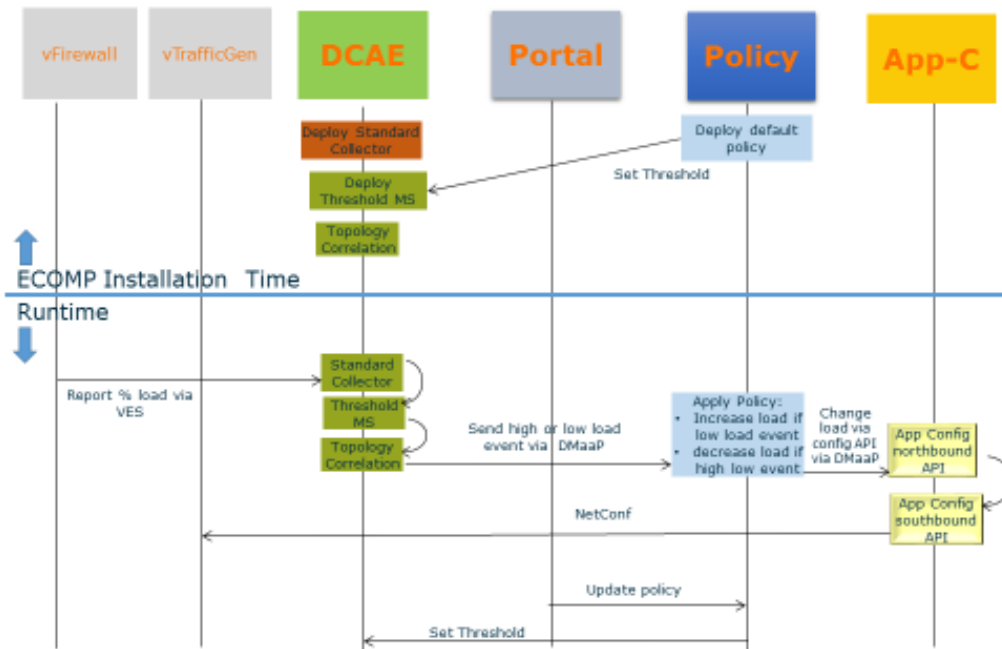
Add Network ⓘ

Name	IPv4	IPv6
 PublicNet (Internet)	172.99.75.45	2001:4802:7805:104:be76:4eff:fe20:54ab
 ServiceNet (Rackspace IAD)	None	None
 demofwl_protected	192.168.120.100	None
 demofwl_unprotected	192.168.110.100	None
 oam_ecomp_Pr2N	10.1.0.11	None

Control Loop Flows

The platform and virtual function interactions in the control loop are summarized here.

vFirewall Control Loop Flows



[Tutorial: Verifying and Observing a deployed Service Instance#vFirewallFlow](#)

Error Handling

Handle MSO Failure on RAM Quota exceeded

A default rackspace account is only allocated 128G but we will require

```
"requestState": "FAILED",
"statusMessage": "Received vfModuleException from VnfAdapter: category='INTERNAL' message='Exception
during create VF 0 : Stack error (CREATE_FAILED):

Resource CREATE failed: Forbidden: resources.vsn_0: Quota exceeded for ram:

Requested 4096, but already used 130048 of 131072 ram (HTTP 403) (Request-ID: req-02439f0a-c8a4-4fe7-8d79-
d6fa99bd4f57) - stack successfully deleted' rolledBack='true'",
"percentProgress": 100,
"finishTime": "Tue, 23 May 2017 22:19:11 GMT"
```

Handle MSO Failure on Create VF Module

Update: 20170523: looks like the template defaults are not being modified - and pass through

[UCA-17](#) - Getting issue details...

STATUS

This type of MSO failure means the demo VNF was retried with the same defaults - a retry will just fail on another IP conflict for the port.

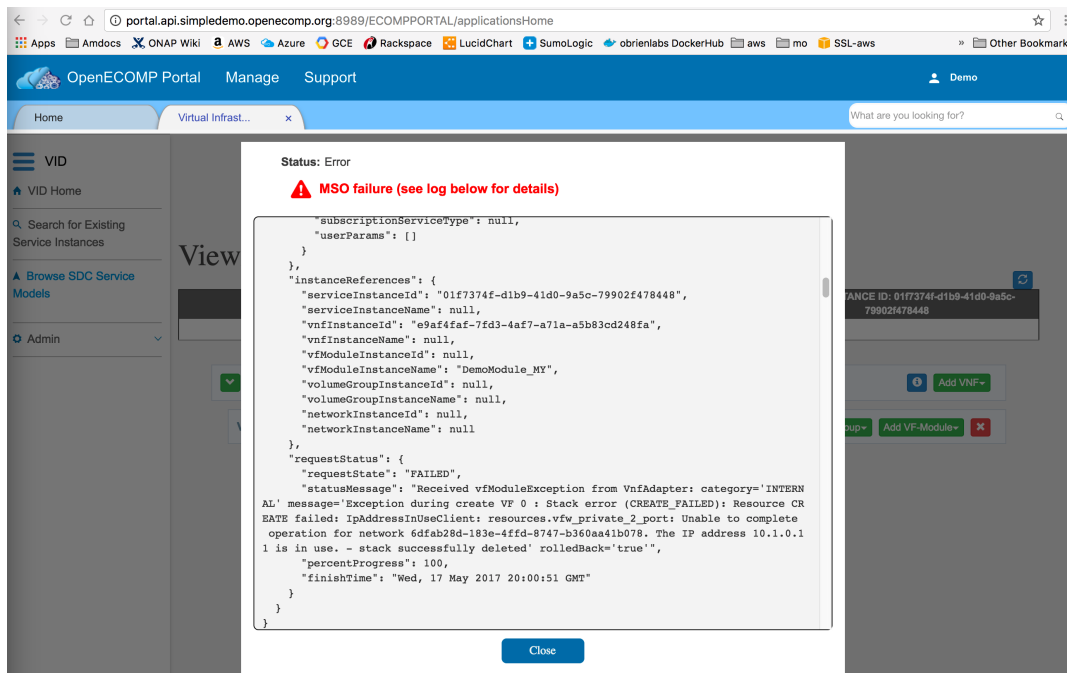
The openstack tenant also happens to have other instances of the VNF that look like they are causing a port resource contention - clean/reset your VM's for now.

```
onap@server-01:~/onap$ openstack port list | grep ip_address=10.1.0.
```

```
| 6d4c9ef9-ceec-4c62-85b1-fa6f2de34256 | FirewallSvcModule-vfw_private_2_port-ewvqxhjdm2tv | BC:76:4E:20:57:DB | ip_address='10.1.0.11', subnet_id='5a4808b2-2fca-40ab-ba43-10d21a9e5b64' | ACTIVE |
```

```
| 7861e542-600f-4bfa-96d0-47e1be19331d | FirewallSvcModule-vpg_private_1_port-ctu2jymvh2yr | BC:76:4E:20:3B:75 | ip_address='10.1.0.12', subnet_id='5a4808b2-2fca-40ab-ba43-10d21a9e5b64' | ACTIVE |
```

```
| b22e7d79-58e6-4c16-8acc-f1a4c358c8c9 | FirewallSvcModule-vsn_private_1_port-xit2fdnpz2yd | BC:76:4E:20:3B:63 | ip_address='10.1.0.13', subnet_id='5a4808b2-2fca-40ab-ba43-10d21a9e5b64' | ACTIVE |
```



```
05/18/17 15:05:57 HTTP Status: OK (200)
```

```
{
  "request": {
    "requestId": "6c0afeaf-42a4-4628-9312-2305e533f673",
    "startTime": "Wed, 17 May 2017 19:04:56 GMT",
    "requestScope": "vfModule",
    "requestType": "createInstance",
    ...
    "requestStatus": {
      "requestState": "FAILED",
      "statusMessage": "Received vfModuleException from VnfAdapter: category='INTERNAL' message='Exception during create VF 0 : Stack error (CREATE_FAILED): Resource CREATE failed: IPAddressInUseClient: resources.vsn_private_1_port: Unable to complete operation for network 6dfab28d-183e-4ffd-8747-b360aa41b078. The IP address 10.1.0.13 is in use. - stack successfully deleted' rolledBack='true'",
      "percentProgress": 100,
      "finishTime": "Wed, 17 May 2017 19:05:48 GMT"
    }
  }
}
```

Handle outdated vFW (201702xx) zip causing Traffic Generation not to start

Fix: Use the 1.0.0 template in Nexus - or the updated one on this wiki

1) The vFW zip attached to the onap.org wiki that we were using will not work with 1.0.0-SNAPSHOT or 1.0.0 (disabled/replacing it) – we are using the official yaml now from 1.0.0 - this fixes the userdata bootstrap script on the PGN instance – where nexus pulls of TG scripts was failing (why the demo did not work in the past) – we now use (with modified ssh key, ips and networks)

Heat template:

<https://nexus.onap.org/content/sites/raw/org.openecomp.demo/heat/vFW/1.0.0/>

Scripts to verify on the pgn VM:

<https://nexus.onap.org/content/sites/raw/org.openecomp.demo/vnfs/vfw/1.0.0/>

After this we were able to run `./demo.sh appc` – to start the TG

Issue:

We are currently having issues with the traffic generator - ~~both starting the stream and also actually echino to the VM (looks like the ssh key in the env is not picked up)~~

Fix: the repo URL in the vFW zip has changed to

#repo_url: <https://ecomp-nexus:8443/repository/raw/org.openecomp.simpdemo>

repo_url: <https://nexus.onap.org/content/sites/raw/org.openecomp.demo/vnfs/vfw/1.0.0-SNAPSHOT>

Check your TG VM and look for scripts in `/config` like the following that should have been copied over

`wget --user=$REPO_USER --password=$REPO_PASSWD $REPO_URL/v_firewall_init.sh`

Also the private key for the 3 vFW VM's is in `/testsuite/robot/assets/keys/robot_ssh_private_key.pvt`

```
root@vml-robot:/opt# curl -X PUT -H "Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/json" -H "Cache-Control: no-cache" -d '{"pg-streams":{"pg-stream":{"id":"fw_udp1","is-enabled":"true"},"id":"fw_udp2","is-enabled":"true"},"id":"fw_udp3","is-enabled":"true"},"id":"fw_udp4","is-enabled":"true"},"id":"fw_udp5","is-enabled":"true"}}}' "http://172.99.75.46:8183/restconf/config/sample-plugin/sample-plugin/pg-streams"
curl: (7) Failed to connect to 172.99.75.46 port 8183: Connection refused
root@vml-robot:/opt# ssh root@172.99.75.46
The authenticity of host '172.99.75.46 (172.99.75.46)' can't be established.
ECDSA key fingerprint is SHA256:Aph2vfiuPQtSC43pnJwXfKcenQZtCUaV7XQX6wpVyok.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.99.75.46' (ECDSA) to the list of known hosts.
root@172.99.75.46's password: 
```

TODO: 20181023 during the Academic Conference : the SDNC preload checkbox does not actually run the preload robot script - still need a manual preload via the rest call in [Vetted vFirewall Demo - Full draft how-to for F2F and ReadTheDocs](#) - it just tells SO to pull in data from SDNC

Install the vFWCL first because it has the network

to do repeated instantiations - adjust the network values in the preload-vnf-topology-operation rest call - being automated in casablanca - 92,96, put the right service-type (Service Instance ID - top right in the gui)