vFWCL instantiation, testing, and debuging

Pre-requisite

Setup the OOM Infrastructure; I've used OOM on Rancher in OpenStack

Running vFW demo - Close-loop

Video of onboarding

I had a hickup at the end, due to the fact I already had another vFW deployed, hence the ip it tried to assign was used. To fix this, I remove the existing stack.

Video of instantiation

I had a a hickup for the vFW_PG due to the fact I pre-loaded on the wrong instance. After realizing, all went well.

1. Let's start by running the **init** goal

   ```
   cd oom/kubernetes/robot
   $ ./demo-k8s.sh init
   ```

   Result:

```
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
```

2. Login into the VNC. Password is **password**

```
<kubernetes-vm-ip>:30211
```

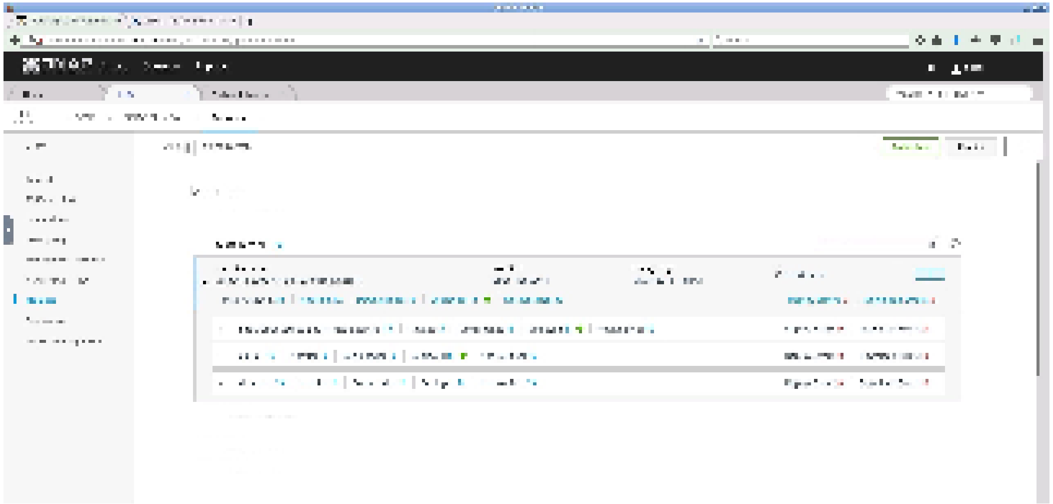
3. Open the browser and navigate to the ONAP Portal

   Login using the **Designer** user. cs0008/demo123456!

   ```
   http://portal.api.simpledemo.onap.org:8989/ONAPPORTAL/login.htm
   ```

4. Virutal Licence Model creation
   a. Open SDC application, click on the **OnBoard** tab.
      i. click **Create new VLM** (Licence Model)
         1. Use **onap** as Vendor Name, and enter a description
         2. clicksave
         3. click **Licence Key Group** and **Add Licence KeyGroup**, then fill in the required fields
         4. click **Entitlements Pools** and **Add Entitlement Pool**, then fill in the required fields
         5. click **Feature Groups** and **Add Feature Group**, then fill in the required fields. Also, under the **Entitlement Pools** tab, drag the created entitlement pool to the left. Same for the **License Key Groups**
         6. click **Licence Aggreements** and **Add Licence Agreement**, then fill in the required fields. Under the tab **Featur es Groups**, drag the feature group created previously.
         7. then **check-in** and **submit**
         8. go back to **OnBoard** page
5. Vendor Software Product onboarding and testing
   a. **click Create a new VSP**
      i. First we create the vFW sinc; give it a name, i.e. vFW_SINC. Select the Vendor (onap) and the Category (Firewall) and give it a description.
      ii. Click on the warning, and add a licence model
      iii. Get the zip package: vfw-sinc.zip
      iv. Click on overview, and import the zip
      v. Click **Proceed to validation** then **check-in** then **submit**
   b. **click Create a new VSP**
      i. Then we create the vFW packet generator; give it a name, i.e. vFW_PG. Select the Vendor (onap) and the Category (Firewall) and give it a description.
      ii. Click on the warning, and add a licence model
      iii. Get the zip package: vfw_pg.zip
      iv. Click on overview, and import the zip
      v. Click **Proceed to validation** then **check-in** then **submit**
   c. **Go to SDC home.** Click on the top right icone, with the orange arrow.
      i. Import the VSP one by one
      ii. Submit for both testing
   d. Logout and Login as the **tester**: **jm0007/demo123456!**
   e. Go to the SDC portal
   f. Test and accept the two VSP
6. Service Creation
   a. Logout and login as the **designer**: **cs0008/demo123456!**
   b. Go to the SDC home page
   c. Click **Add a Service**
   d. Fill in the required field
   e. Click **Create**
   f. Click on the **Composition** left tab
   g. In the search bar, type "vFW" to narrow down the created VSP, and drag them both.
   h. Then click **Submit for Testing**
7. Service Testing
   a. Logout and Login as the **tester**: **jm0007/demo123456!**
   b. Go to the SDC portal
   c. Test and accept the service
8. Service Approval
   a. Logout and Login as the **govener**: **gv0001/demo123456!**
   b. Go to the SDC portal
   c. Approve the service
9. Service Distribution
   a. Logout and Login as the **operator**: **op0001/demo123456!**
   b. Go to the SDC portal
   c. Distribute the service
   d. Click on the left tab monitor and click on arrow to open the distribution status
   e. Wait until everything is disitributed (green tick)

Expected output:



10. Service Instance creation:
    a. Logout and Login as the **user**: **demo/demo123456!**
    b. Go to the VID portal
    c. Click the **Browse SDC Service Models** tab
    d. Click **Deploy** on the service to deploy
    e. Fiil in the required filed, call it vFW_Service for instance. Once done, this will redirect you to a new screen
    f. Click **Add VNF**, and select the vFW_SINC VNF first
    g. Fill in the required field. Call it vFW_SINC_VNF, for instance.
    h. Click **Add VNF**, and select the vFW_PG_VNF first
    i. Fill in the required field. Call it vFW_PG_VNF, for instance.
11. SDNC preload:
    a. Then go to the SDNC Admin portal and create an account

```
<kubernetes-host-ip>:30201/signup
```

    b. Login into the SDNC admin portal

```
<kubernetes-host-ip>:30201/login
```

    c. Click **Profiles** then **Add VNF Profile**
        i. The **VNF Type** is the string that looks like this: VfwPg..base_vpkg..module-0 It can be copy/paste from VID, when attempting to create the VF-Module
        ii. Enter 100 for the **Availability Zone Count**
        iii. Enter vFW for **Equipement Role**
    d. Repeat the same for the other VNF

    e. Pre-load the vFW SINC. Mind the following values:

service-type: it's the service instance ID of the service instance created step 9
vnf-name: the name to give to the VF-Module. The same name will have to be re-use when creating the VF-Module
vnf-type: Same as the one used to add the profile in SDNC admin portal
generic-vnf-name: The name of the created VNF, see step 9f
vfw_name_0: is the same as the generic-vnf-name
generic-vnf-type: Can be find in VID, please see video if not found.
dcae_collector_ip: Has to be the IP address of the dcaedoks00 VM
Make sure **image_name**, **flavor_name**, **public_net_id**, **onap_private_net_id**, **onap_private_subnet_id**, **key_name** and **pub_key** reflect your environment

```
curl -X POST \

http://<kubernetes-host-ip>:30202/restconf/operations/VNF-API:preload-vnf-topology-operation \
  -H 'accept: application/json' \
  -H 'authorization: Basic
YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrM2VIbGNzZTJnQXc4NHZhb0dHbUp2VXky
VQ==' \
  -H 'content-type: application/json' \
  -H 'x-fromappid: API client' \
  -d '{
  "input": {
    "vnf-topology-information": {
      "vnf-topology-identifier": {
        "service-type": "34992be5-b38c-46da-96b2-553e60f9c24b",
        "vnf-name": "vFW_SINC_Module",
        "vnf-type": "VfwSinc..base_vfw..module-0",
        "generic-vnf-name": "vFW_SINC_VNF",
        "generic-vnf-type": "vFW_SINC 0"
      },
      "vnf-assignments": {
        "availability-zones": [

        ],
        "vnf-networks": [

        ],
        "vnf-vms": [

        ]
      },
      "vnf-parameters": [
        {
          "vnf-parameter-name": "image_name",
          "vnf-parameter-value": "trusty"
        },
        {
          "vnf-parameter-name": "flavor_name",
          "vnf-parameter-value": "m1.medium"
        },
        {
          "vnf-parameter-name": "public_net_id",
          "vnf-parameter-value":
"d87ff178-3eb7-44df-a57b-84636dbdc817"
        },
        {
          "vnf-parameter-name": "unprotected_private_net_id",
          "vnf-parameter-value": "zdfw1fwl01_unprotected"
        },
        {
          "vnf-parameter-name": "unprotected_private_subnet_id",
          "vnf-parameter-value": "zdfw1fwl01_unprotected_sub"
        },
        {
          "vnf-parameter-name": "protected_private_net_id",
          "vnf-parameter-value": "zdfw1fwl01_protected"
        },
        {
          "vnf-parameter-name": "protected_private_subnet_id",
          "vnf-parameter-value": "zdfw1fwl01_protected_sub"
        },
        {
          "vnf-parameter-name": "onap_private_net_id",
          "vnf-parameter-value": "oam_onap_k0H4"
        },
        {
          "vnf-parameter-name": "onap_private_subnet_id",
          "vnf-parameter-value": "oam_onap_k0H4"
        },
        {
          "vnf-parameter-name": "unprotected_private_net_cidr",
          "vnf-parameter-value": "192.168.10.0/24"
        },
        {
          "vnf-parameter-name": "protected_private_net_cidr",
          "vnf-parameter-value": "192.168.20.0/24"
        },
        {
          "vnf-parameter-name": "onap_private_net_cidr",
          "vnf-parameter-value": "10.0.0.0/16"
        },
        {
          "vnf-parameter-name": "vfw_private_ip_0",
          "vnf-parameter-value": "192.168.10.100"
        },
        {
          "vnf-parameter-name": "vfw_private_ip_1",
          "vnf-parameter-value": "192.168.20.100"
        },
        {
          "vnf-parameter-name": "vfw_private_ip_2",
          "vnf-parameter-value": "10.0.100.5"
        },
        {
          "vnf-parameter-name": "vpg_private_ip_0",
          "vnf-parameter-value": "192.168.10.200"
        },
        {
          "vnf-parameter-name": "vsn_private_ip_0",
          "vnf-parameter-value": "192.168.20.250"
        },
        {
          "vnf-parameter-name": "vsn_private_ip_1",
          "vnf-parameter-value": "10.0.100.4"
        },
        {
          "vnf-parameter-name": "vfw_name_0",
          "vnf-parameter-value": "vFW_SINC_VNF"
        },
        {
```

```
          "vnf-parameter-name": "vsn_name_0",
          "vnf-parameter-value": "zdfw1fwl01snk01"
        },
        {
          "vnf-parameter-name": "vnf_id",
          "vnf-parameter-value": "vFirewal_vSink_demo_app"
        },
        {
          "vnf-parameter-name": "vf_module_id",
          "vnf-parameter-value": "vFirewall_vSink"
        },
        {
          "vnf-parameter-name": "dcae_collector_ip",
          "vnf-parameter-value": "10.195.200.38"
        },
        {
          "vnf-parameter-name": "dcae_collector_port",
          "vnf-parameter-value": "8080"
        },
        {
          "vnf-parameter-name": "repo_url_blob",
          "vnf-parameter-value":
"https://nexus.onap.org/content/sites/raw"
        },
        {
          "vnf-parameter-name": "repo_url_artifacts",
          "vnf-parameter-value":
"https://nexus.onap.org/content/groups/staging"
        },
        {
          "vnf-parameter-name": "demo_artifacts_version",
          "vnf-parameter-value": "1.1.1"
        },
        {
          "vnf-parameter-name": "install_script_version",
          "vnf-parameter-value": "1.1.1"
        },
        {
          "vnf-parameter-name": "key_name",
          "vnf-parameter-value": "onap_key_k0H4"
        },
        {
          "vnf-parameter-name": "pub_key",
          "vnf-parameter-value": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCmuLf5dnvDS4hiwmXYg2YtgByeAj8ZoH5t
oGPNENIr9uIhgRclPWb5HSIDzhFLKy9K9Z1ht5XZEkzAcslSIKkodZlVYyucG/Qw
qLlN8N05EMLVm6TudjUp/j/VDvavSgp/xzIDsdHuhQZ8VHRE88mKzsTA4jPFp4s4
Ic8eCes4nrydMrlbxeLjV3/+/xc77StQ7hDMaBlJX8xztgHRodxIQmMBWwb/4YSx
jTbO0cwi4XYlRXzFPY7vmO2VDRhfaOVtyv8Pw6a3AaqIP6CR0z6QgbLYjtiFbWmh
KQ+0qUfJeb0Kkc7Deok7x58a3mHkhswGS1aJLCaHC/W1b7n6C+lv
adetalhouet@bell.corp.bce.ca"
        },
        {
          "vnf-parameter-name": "cloud_env",
          "vnf-parameter-value": "openstack"
        }
      ]
    },
    "request-information": {
      "request-id": "robot12",
      "order-version": "1",
      "notification-url": "openecomp.org",
      "order-number": "1",
      "request-action": "PreloadVNFRequest"
    },
    "sdnc-request-header": {
      "svc-request-id": "robot12",
      "svc-notification-url":
"http://openecomp.org:8080/adapters/rest/SDNCNotify",
      "svc-action": "reserve"
    }
  }
}'
```

Expected result:

```
{
    "output": {
        "svc-request-id": "robot12",
        "response-code": "200",
        "ack-final-indicator": "Y"
    }
}
```

f. Pre-load the vFW PG. Mind the following values:

service-type: it's the service instance ID of the service instance created step 9
vnf-name: the name to give to the VF-Module. The same name will have to be re-use when creating the VF-Module
vnf-type: Same as the one used to add the profile in SDNC admin portal
generic-vnf-name: The name of the created VNF, see step 9h
vpg_name_0: is the same as the generic-vnf-name
generic-vnf-type: Can be find in VID, please see video if not found.
Make sure **image_name**, **flavor_name**, **public_net_id**, **onap_private_net_id**, **onap_private_subnet_id**, **key_name** and **pub_key** reflect your environment

```
curl -X POST \

http://<kubernetes-host-ip>:30202/restconf/operations/VNF-API:pr
eload-vnf-topology-operation \
  -H 'accept: application/json' \
  -H 'authorization: Basic
YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrM2VIbGNzZTJnQXc4NHZhb0dHbUp2VXky
VQ==' \
  -H 'content-type: application/json' \
  -H 'x-fromappid: API client' \
  -d '{
  "input": {
    "vnf-topology-information": {
      "vnf-topology-identifier": {
        "service-type": "df6e075a-119a-4790-a470-2474a692e3ce",
        "vnf-name": "vFW_PG_Module",
        "vnf-type": "VfwPg..base_vpkg..module-0",
        "generic-vnf-name": "vFW_PG_VNF",
        "generic-vnf-type": "vFW_PG 0"
      },
      "vnf-assignments": {
        "availability-zones": [

        ],
        "vnf-networks": [

        ],
        "vnf-vms": [
        ]
      },
      "vnf-parameters": [
        {
          "vnf-parameter-name": "image_name",
          "vnf-parameter-value": "trusty"
        },
        {
          "vnf-parameter-name": "flavor_name",
          "vnf-parameter-value": "m1.medium"
        },
        {
          "vnf-parameter-name": "public_net_id",
          "vnf-parameter-value":
"d87ff178-3eb7-44df-a57b-84636dbdc817"
        },
        {
          "vnf-parameter-name": "unprotected_private_net_id",
          "vnf-parameter-value": "zdfw1fwl01_unprotected"
        },
        {
          "vnf-parameter-name": "unprotected_private_subnet_id",
          "vnf-parameter-value": "zdfw1fwl01_unprotected_sub"
        },
        {
          "vnf-parameter-name": "onap_private_net_id",
          "vnf-parameter-value": "oam_onap_k0H4"
        },
        {
          "vnf-parameter-name": "onap_private_subnet_id",
          "vnf-parameter-value": "oam_onap_k0H4"
        },
        {
          "vnf-parameter-name": "unprotected_private_net_cidr",
          "vnf-parameter-value": "192.168.10.0/24"
        },
        {
          "vnf-parameter-name": "protected_private_net_cidr",
          "vnf-parameter-value": "192.168.20.0/24"
        },
        {
          "vnf-parameter-name": "onap_private_net_cidr",
          "vnf-parameter-value": "10.0.0.0/16"
        },
        {
          "vnf-parameter-name": "vfw_private_ip_0",
          "vnf-parameter-value": "192.168.10.100"
        },
        {
          "vnf-parameter-name": "vpg_private_ip_0",
          "vnf-parameter-value": "192.168.10.200"
        },
        {
          "vnf-parameter-name": "vpg_private_ip_1",
          "vnf-parameter-value": "10.0.80.2"
        },
        {
          "vnf-parameter-name": "vsn_private_ip_0",
          "vnf-parameter-value": "192.168.20.250"
        },
        {
          "vnf-parameter-name": "vpg_name_0",
          "vnf-parameter-value": "vFW_PG_VNF"
        },
        {
          "vnf-parameter-name": "vnf_id",
          "vnf-parameter-value": "vPacketGen_demo_app"
        },
        {
          "vnf-parameter-name": "vf_module_id",
          "vnf-parameter-value": "vPacketGen"
        },
        {
          "vnf-parameter-name": "repo_url_blob",
          "vnf-parameter-value":
"https://nexus.onap.org/content/sites/raw"
        },
        {
          "vnf-parameter-name": "repo_url_artifacts",
          "vnf-parameter-value":
"https://nexus.onap.org/content/groups/staging"
        },
        {
          "vnf-parameter-name": "demo_artifacts_version",
          "vnf-parameter-value": "1.1.1"
        },
        {
          "vnf-parameter-name": "install_script_version",
          "vnf-parameter-value": "1.1.1"
        },
        {
          "vnf-parameter-name": "key_name",
          "vnf-parameter-value": "vfw_key"
        },
        {
          "vnf-parameter-name": "pub_key",
          "vnf-parameter-value": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCmuLf5dnvDS4hiwmXYg2YtgByeAj8ZoH5t
oGPNENIr9uIhgRclPWb5HSIDzhFLKy9K9Z1ht5XZEkzAcslSIKkodZlVYyucG/Qw
qLlN8N05EMLVm6TudjUp/j/VDvavSgp/xzIDsdHuhQZ8VHRE88mKzsTA4jPFp4s4
Ic8eCes4nrydMrlbxeLjV3/+/xc77StQ7hDMaBlJX8xztgHRodxIQmMBWwb/4YSx
jTbO0cwi4XYlRXzFPY7vmO2VDRhfaOVtyv8Pw6a3AaqIP6CR0z6QgbLYjtiFbWmh
KQ+0qUfJeb0Kkc7Deok7x58a3mHkhswGS1aJLCaHC/W1b7n6C+lv
adetalhouet@bell.corp.bce.ca"
        },
        {
          "vnf-parameter-name": "cloud_env",
          "vnf-parameter-value": "openstack"
        }
      ]
    },
    "request-information": {
      "request-id": "robot12",
      "order-version": "1",
      "notification-url": "openecomp.org",
      "order-number": "1",
      "request-action": "PreloadVNFRequest"
    },
    "sdnc-request-header": {
      "svc-request-id": "robot12",
      "svc-notification-url":
"http://openecomp.org:8080/adapters/rest/SDNCNotify",
      "svc-action": "reserve"
```

```
        }
    }
}'
```

Expected result:

```
{
    "output": {
        "svc-request-id": "robot12",
        "response-code": "200",
        "ack-final-indicator": "Y"
    }
}
```

g. Create the VF-Module for vFW_SINC
    i. The instance name must be the **vnf-name** setup in the preload phase.
    ii. After a few minutes, the stack should be created.

h. Create the VF-Module for vFW_PG
    i. The instance name must be the **vnf-name** setup in the preload phase.
    ii. After a few minutes, the stack should be created.

Close loop

1. Run heatbridge robot tag to tell AAI about the relationaship between the created HEAT stack (SINC one) and the service instance id.

   To run this, you need:
   - the heat stack name of the vSINC
   - the service instance id

```
$ ./demo-k8s.sh heatbridge vFW_SINC_Module
82678348-2f42-4ee7-bd29-0ef24b5e4bca vFW
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 ...
=====================================================================
=========
Run Heatbridge :: Try to run heatbridge
| 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/heatbridge/output.xml
Log:     /share/logs/demo/heatbridge/log.html
Report:  /share/logs/demo/heatbridge/report.html
```

2. Upload operational policy: this is to tell policy that for this specific instance, we should apply this policy.
    a. Retrieve from MSO Catalog the **modelInvariantUuid** for the **vFW_PG**. Specify in the bellow request the **service-model-name**, as defined step 5.c.

```
curl -X GET \

'http://<kubernetes-host>:30223/ecomp/mso/catalog/v2/serviceVnfs
?serviceModelName=<service-model-name>' \
  -H 'Accept: application/json' \
  -H 'Authorization: Basic
SW5mcmFQb3J0YWxDbGllbnQ6cGFzc3dvcmQxJA==' \
  -H 'Content-Type: application/json' \
  -H 'X-FromAppId: Postman' \
  -H 'X-TransactionId: get_service_vnfs'
```

Based on the payload bellow, result would be: **86a1bdd8-1f59-4796-bf30-3002108068f6**

```
{
    "serviceVnfs": [
        {
            "modelInfo": {
                "modelName": "vFW_PG",
                "modelUuid":
"7af8882e-f732-405f-b48b-38b6403654ea",
                "modelInvariantUuid":
"86a1bdd8-1f59-4796-bf30-3002108068f6",
                "modelVersion": "1.0",
                "modelCustomizationUuid":
"a2521929-d6da-46cc-9a62-ca3b6c3cef9b",
                "modelInstanceName": "vFW_PG 0"
            },
            "toscaNodeType": "org.openecomp.resource.vf.VfwPg",
            "nfFunction": "",
            "nfType": "",
            "nfRole": "",
            "nfNamingCode": "",
            "vfModules": [
                {
                    "modelInfo": {
                        "modelName":
"VfwPg..base_vpkg..module-0",
                        "modelUuid":
"54a98442-52e3-46e8-8b40-193f04e92ff7",
                        "modelInvariantUuid":
"9c6c0369-a9c1-4419-94c9-aabf6250fc87",
                        "modelVersion": "1",
                        "modelCustomizationUuid":
"35595818-2e09-4ad2-b6ce-2ffc263489af"
                    },
                    "isBase": true,
                    "vfModuleLabel": "base_vpkg",
                    "initialCount": 1,
                    "hasVolumeGroup": true
                }
            ]
        },
        {
            "modelInfo": {
                "modelName": "vFW_SINC",
                "modelUuid":
"b8cc7acf-eba8-4ddb-950a-be52a96b28c8",
                "modelInvariantUuid":
"edd473e1-7d08-4cf1-be31-0d705017f644",
                "modelVersion": "1.0",
                "modelCustomizationUuid":
"c890203f-44a0-4c43-aadb-250d8f6c54b0",
                "modelInstanceName": "vFW_SINC 0"
            },
            "toscaNodeType": "org.openecomp.resource.vf.VfwSinc",
            "nfFunction": "",
            "nfType": "",
            "nfRole": "",
            "nfNamingCode": "",
            "vfModules": [
                {
                    "modelInfo": {
                        "modelName":
"VfwSinc..base_vfw..module-0",
                        "modelUuid":
"605ef192-e190-4043-97be-31a0d64a2f8e",
                        "modelInvariantUuid":
"858e065b-7491-4c70-91e6-109a65c6102d",
                        "modelVersion": "1",
                        "modelCustomizationUuid":
"acf94576-fe00-43ec-b9f9-0f8748e44c0a"
                    },
                    "isBase": true,
                    "vfModuleLabel": "base_vfw",
                    "initialCount": 1,
                    "hasVolumeGroup": true
                }
            ]
        }
    ]
}
```

b. Under

```
oom/kubernetes/policy/script
```

invoke the script as follow:

```
Usage: update-vfw-op-policy.sh <k8s-host> <policy-pdp-node-port>
<policy-drools-node-port> <resource-id>

./update-vfw-op-policy.sh 10.195.197.53 30220 30221
86a1bdd8-1f59-4796-bf30-3002108068f
```

Result can look like, with debug enable (/bin/bash -x)

```
$ ./update-vfw-op-policy.sh 10.195.197.53 30220 30221
86a1bdd8-1f59-4796-bf30-3002108068f
+ '[' 4 -ne 4 ']'
+ K8S_HOST=10.195.197.53
+ POLICY_PDP_PORT=30220
+ POLICY_DROOLS_PORT=30221
+ RESOURCE_ID=86a1bdd8-1f59-4796-bf30-3002108068f
+ echo

+ echo

+ echo 'Removing the vFW Policy from PDP..'
Removing the vFW Policy from PDP..
+ echo

+ echo

+ curl -v -X DELETE --header 'Content-Type: application/json'
--header 'Accept: text/plain' --header 'ClientAuth:
cHl0aG9uOnRlc3Q=' --header 'Authorization: Basic
dGVzdHBkcDphbHBoYTEyMw==' --header 'Environment: TEST' -d '{
  "pdpGroup": "default",
  "policyComponent" : "PDP",
  "policyName": "com.BRMSParamvFirewall",
  "policyType": "BRMS_Param"
}' http://10.195.197.53:30220/pdp/api/deletePolicy
*   Trying 10.195.197.53...
* TCP_NODELAY set
* Connected to 10.195.197.53 (10.195.197.53) port 30220 (#0)
> DELETE /pdp/api/deletePolicy HTTP/1.1
> Host: 10.195.197.53:30220
> User-Agent: curl/7.54.0
> Content-Type: application/json
> Accept: text/plain
> ClientAuth: cHl0aG9uOnRlc3Q=
> Authorization: Basic dGVzdHBkcDphbHBoYTEyMw==
> Environment: TEST
> Content-Length: 128
>
* upload completely sent off: 128 out of 128 bytes
< HTTP/1.1 200 OK
< Server: Apache-Coyote/1.1
< Content-Type: text/plain;charset=ISO-8859-1
< Content-Length: 91
< Date: Wed, 20 Dec 2017 20:17:22 GMT
<
* Connection #0 to host 10.195.197.53 left intact
Transaction ID: af030f0c-0c2b-43a1-b1ec-6abf4ca73799 --The policy
was successfully deleted.+ sleep 20



+ echo

+ echo

+ echo 'Updating vFW Operational Policy ..'
Updating vFW Operational Policy ..
+ echo

+ curl -v -X PUT --header 'Content-Type: application/json'
--header 'Accept: text/plain' --header 'ClientAuth:
cHl0aG9uOnRlc3Q=' --header 'Authorization: Basic
dGVzdHBkcDphbHBoYTEyMw==' --header 'Environment: TEST' -d '{
  "policyConfigType": "BRMS_PARAM",
  "policyName": "com.BRMSParamvFirewall",
  "policyDescription": "BRMS Param vFirewall policy",
  "policyScope": "com",
  "attributes": {
    "MATCHING": {
      "controller": "amsterdam"
    },
    "RULE": {
      "templateName": "ClosedLoopControlName",
      "closedLoopControlName":
"ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
      "controlLoopYaml":
"controlLoop%3A%0D%0A++version%3A+2.0.0%0D%0A++controlLoopName%3
A+ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a%0D%
0A++trigger_policy%3A+unique-policy-id-1-modifyConfig%0D%0A++tim
eout%3A+1200%0D%0A++abatement%3A+false%0D%0A+%0D%0Apolicies%3A%0
D%0A++-+id%3A+unique-policy-id-1-modifyConfig%0D%0A++++name%3A+m
odify+packet+gen+config%0D%0A++++description%3A%0D%0A++++actor%3
A+APPC%0D%0A++++recipe%3A+ModifyConfig%0D%0A++++target%3A%0D%0A+
+++++%23+TBD+-+Cannot+be+known+until+instantiation+is+done%0D%0A
++++++resourceID%3A+86a1bdd8-1f59-4796-bf30-3002108068f%0D%0A+++
+++type%3A+VNF%0D%0A++++retry%3A+0%0D%0A++++timeout%3A+300%0D%0A
++++success%3A+final_success%0D%0A++++failure%3A+final_failure%0
D%0A++++failure_timeout%3A+final_failure_timeout%0D%0A++++failur
e_retries%3A+final_failure_retries%0D%0A++++failure_exception%3A
+final_failure_exception%0D%0A++++failure_guard%3A+final_failure
_guard"
    }
  }
}' http://10.195.197.53:30220/pdp/api/updatePolicy
*   Trying 10.195.197.53...
* TCP_NODELAY set
* Connected to 10.195.197.53 (10.195.197.53) port 30220 (#0)
> PUT /pdp/api/updatePolicy HTTP/1.1
> Host: 10.195.197.53:30220
> User-Agent: curl/7.54.0
> Content-Type: application/json
> Accept: text/plain
> ClientAuth: cHl0aG9uOnRlc3Q=
> Authorization: Basic dGVzdHBkcDphbHBoYTEyMw==
> Environment: TEST
> Content-Length: 1327
> Expect: 100-continue
>
< HTTP/1.1 100 Continue
* We are completely uploaded and fine
< HTTP/1.1 200 OK
< Server: Apache-Coyote/1.1
< Content-Type: text/plain;charset=ISO-8859-1
< Content-Length: 149
< Date: Wed, 20 Dec 2017 20:17:42 GMT
<
* Connection #0 to host 10.195.197.53 left intact
Transaction ID: 20f4e273-d193-466c-8cce-ee643a854f5f --Policy
with the name com.Config_BRMS_Param_BRMSParamvFirewall.2.xml was
successfully updated. + sleep 5
+ echo

+ echo

+ echo 'Pushing the vFW Policy ..'
Pushing the vFW Policy ..
+ echo

+ echo

+ curl -v --silent -X PUT --header 'Content-Type:
application/json' --header 'Accept: text/plain' --header
'ClientAuth: cHl0aG9uOnRlc3Q=' --header 'Authorization: Basic
dGVzdHBkcDphbHBoYTEyMw==' --header 'Environment: TEST' -d '{
  "pdpGroup": "default",
  "policyName": "com.BRMSParamvFirewall",
  "policyType": "BRMS_Param"
}' http://10.195.197.53:30220/pdp/api/pushPolicy
*   Trying 10.195.197.53...
* TCP_NODELAY set
* Connected to 10.195.197.53 (10.195.197.53) port 30220 (#0)
> PUT /pdp/api/pushPolicy HTTP/1.1
> Host: 10.195.197.53:30220
> User-Agent: curl/7.54.0
> Content-Type: application/json
> Accept: text/plain
> ClientAuth: cHl0aG9uOnRlc3Q=
> Authorization: Basic dGVzdHBkcDphbHBoYTEyMw==
> Environment: TEST
> Content-Length: 99
>
* upload completely sent off: 99 out of 99 bytes
< HTTP/1.1 200 OK
< Server: Apache-Coyote/1.1
< Content-Type: text/plain;charset=ISO-8859-1
< Content-Length: 162
< Date: Wed, 20 Dec 2017 20:17:48 GMT
<
* Connection #0 to host 10.195.197.53 left intact
Transaction ID: e8bc4ae1-d0b0-483e-b1ba-871486661240 --Policy
'com.Config_BRMS_Param_BRMSParamvFirewall.2.xml' was successfully
pushed to the PDP group 'default'.+ sleep 20
+ echo

+ echo

+ echo 'Restarting PDP-D ..'
Restarting PDP-D ..
+ echo

+ echo

++ kubectl --namespace onap-policy get pods
++ sed 's/ .*//'
++ grep drools
+ POD=drools-870120400-5b5k1
+ kubectl --namespace onap-policy exec -it drools-870120400-5b5k1
-- bash -c 'source /opt/app/policy/etc/profile.d/env.sh && policy
stop && sleep 5 && policy start'
Defaulting container name to drools.
Use 'kubectl describe pod/drools-870120400-5b5k1' to see all of
the containers in this pod.
[drools-pdp-controllers]



 L []: Stopping Policy Management... Policy Management (pid=5452)
is stopping... Policy Management has stopped.
[drools-pdp-controllers]
```

```
 L []: Policy Management (pid 5722) is running
+ sleep 20
+ echo

+ echo

+ echo 'PDP-D amsterdam maven coordinates ..'
PDP-D amsterdam maven coordinates ..
+ echo

+ echo

+ curl -vvv --silent --user @1b3rt:31nst31n -X GET
http://10.195.197.53:30221/policy/pdp/engine/controllers/amsterd
am/drools
+ python -m json.tool
*   Trying 10.195.197.53...
* TCP_NODELAY set
* Connected to 10.195.197.53 (10.195.197.53) port 30221 (#0)
* Server auth using Basic with user '@1b3rt'
> GET /policy/pdp/engine/controllers/amsterdam/drools HTTP/1.1
> Host: 10.195.197.53:30221
> Authorization: Basic QDFiM3J0OjMxbnN0MzFu
> User-Agent: curl/7.54.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Wed, 20 Dec 2017 20:18:49 GMT
< Content-Type: application/json
< Content-Length: 382
< Server: Jetty(9.3.14.v20161028)
<
{ [382 bytes data]
* Connection #0 to host 10.195.197.53 left intact
{
    "alive": true,
    "artifactId": "policy-amsterdam-rules",
    "brained": true,
    "groupId": "org.onap.policy-engine.drools.amsterdam",
    "locked": false,
    "modelClassLoaderHash": 665564874,
    "recentSinkEvents": [],
    "recentSourceEvents": [],
    "sessionCoordinates": [

"org.onap.policy-engine.drools.amsterdam:policy-amsterdam-rules:
0.6.0:closedloop-amsterdam"
    ],
    "sessions": [
        "closedloop-amsterdam"
    ],
    "version": "0.6.0"
}
```

```
+ echo

+ echo

+ echo 'PDP-D control loop updated ..'
PDP-D control loop updated ..
+ echo

+ echo

+ curl -v --silent --user @1b3rt:31nst31n -X GET
http://10.195.197.53:30221/policy/pdp/engine/controllers/amsterd
am/drools/facts/closedloop-amsterdam/org.onap.policy.controlloop
.Params
+ python -m json.tool
*   Trying 10.195.197.53...
* TCP_NODELAY set
* Connected to 10.195.197.53 (10.195.197.53) port 30221 (#0)
* Server auth using Basic with user '@1b3rt'
> GET
/policy/pdp/engine/controllers/amsterdam/drools/facts/closedloop
-amsterdam/org.onap.policy.controlloop.Params HTTP/1.1
> Host: 10.195.197.53:30221
> Authorization: Basic QDFiM3J0OjMxbnN0MzFu
> User-Agent: curl/7.54.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Wed, 20 Dec 2017 20:18:50 GMT
< Content-Type: application/json
< Content-Length: 3565
< Server: Jetty(9.3.14.v20161028)
<
{ [1207 bytes data]
* Connection #0 to host 10.195.197.53 left intact
[
    {
        "closedLoopControlName":
"ControlLoop-vCPE-48f0c2c3-a172-4192-9ae3-052274181b6e",
        "controlLoopYaml":
"controlLoop%3A%0D%0A++version%3A+2.0.0%0D%0A++controlLoopName%3
A+ControlLoop-vCPE-48f0c2c3-a172-4192-9ae3-052274181b6e%0D%0A++t
rigger_policy%3A+unique-policy-id-1-restart%0D%0A++timeout%3A+36
00%0D%0A++abatement%3A+true%0D%0A+%0D%0Apolicies%3A%0D%0A++-+id%
3A+unique-policy-id-1-restart%0D%0A++++name%3A+Restart+the+VM%0D
%0A++++description%3A%0D%0A++++actor%3A+APPC%0D%0A++++recipe%3A+
Restart%0D%0A++++target%3A%0D%0A++++++type%3A+VM%0D%0A++++retry%
3A+3%0D%0A++++timeout%3A+1200%0D%0A++++success%3A+final_success%
0D%0A++++failure%3A+final_failure%0D%0A++++failure_timeout%3A+fi
nal_failure_timeout%0D%0A++++failure_retries%3A+final_failure_re
tries%0D%0A++++failure_exception%3A+final_failure_exception%0D%0
A++++failure_guard%3A+final_failure_guard"
    },
    {
        "closedLoopControlName":
"ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a",
        "controlLoopYaml":
"controlLoop%3A%0D%0A++version%3A+2.0.0%0D%0A++controlLoopName%3
A+ControlLoop-vFirewall-d0a1dfc6-94f5-4fd4-a5b5-4630b438850a%0D%
0A++trigger_policy%3A+unique-policy-id-1-modifyConfig%0D%0A++tim
eout%3A+1200%0D%0A++abatement%3A+false%0D%0A+%0D%0Apolicies%3A%0
D%0A++-+id%3A+unique-policy-id-1-modifyConfig%0D%0A++++name%3A+m
odify+packet+gen+config%0D%0A++++description%3A%0D%0A++++actor%3
A+APPC%0D%0A++++recipe%3A+ModifyConfig%0D%0A++++target%3A%0D%0A+
+++++%23+TBD+-+Cannot+be+known+until+instantiation+is+done%0D%0A
++++++resourceID%3A+86a1bdd8-1f59-4796-bf30-3002108068f%0D%0A+++
+++type%3A+VNF%0D%0A++++retry%3A+0%0D%0A++++timeout%3A+300%0D%0A
++++success%3A+final_success%0D%0A++++failure%3A+final_failure%0
D%0A++++failure_timeout%3A+final_failure_timeout%0D%0A++++failur
e_retries%3A+final_failure_retries%0D%0A++++failure_exception%3A
+final_failure_exception%0D%0A++++failure_guard%3A+final_failure
_guard"
    },
    {
        "closedLoopControlName":
"ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3",
        "controlLoopYaml":
"controlLoop%3A%0D%0A++version%3A+2.0.0%0D%0A++controlLoopName%3
A+ControlLoop-vDNS-6f37f56d-a87d-4b85-b6a9-cc953cf779b3%0D%0A++t
rigger_policy%3A+unique-policy-id-1-scale-up%0D%0A++timeout%3A+1
200%0D%0A++abatement%3A+false%0D%0Apolicies%3A%0D%0A++-+id%3A+un
ique-policy-id-1-scale-up%0D%0A++++name%3A+Create+a+new+VF+Modul
e%0D%0A++++description%3A%0D%0A++++actor%3A+SO%0D%0A++++recipe%3
A+VF+Module+Create%0D%0A++++target%3A%0D%0A++++++type%3A+VNF%0D%
0A++++retry%3A+0%0D%0A++++timeout%3A+1200%0D%0A++++success%3A+fi
nal_success%0D%0A++++failure%3A+final_failure%0D%0A++++failure_t
imeout%3A+final_failure_timeout%0D%0A++++failure_retries%3A+fina
l_failure_retries%0D%0A++++failure_exception%3A+final_failure_ex
ception%0D%0A++++failure_guard%3A+final_failure_guard"
    },
    {
        "closedLoopControlName":
"ControlLoop-VOLTE-2179b738-fd36-4843-a71a-a8c24c70c55b",
        "controlLoopYaml":
"controlLoop%3A%0D%0A++version%3A+2.0.0%0D%0A++controlLoopName%3
A+ControlLoop-VOLTE-2179b738-fd36-4843-a71a-a8c24c70c55b%0D%0A++
trigger_policy%3A+unique-policy-id-1-restart%0D%0A++timeout%3A+3
600%0D%0A++abatement%3A+false%0D%0A+%0D%0Apolicies%3A%0D%0A++-+i
d%3A+unique-policy-id-1-restart%0D%0A++++name%3A+Restart+the+VM%
0D%0A++++description%3A%0D%0A++++actor%3A+VFC%0D%0A++++recipe%3A
+Restart%0D%0A++++target%3A%0D%0A++++++type%3A+VM%0D%0A++++retry
%3A+3%0D%0A++++timeout%3A+1200%0D%0A++++success%3A+final_success
%0D%0A++++failure%3A+final_failure%0D%0A++++failure_timeout%3A+f
inal_failure_timeout%0D%0A++++failure_retries%3A+final_failure_r
etries%0D%0A++++failure_exception%3A+final_failure_exception%0D%
0A++++failure_guard%3A+final_failure_guard"
    }
]
```

3. Mount APPC
    a. Get the VNF instance ID, either through VID or through AAI. Bellow the AAI request

```
curl -X GET \
  https://<kubernetes-host>:30233/aai/v8/network/generic-vnfs/ \
  -H 'Accept: application/json' \
  -H 'Authorization: Basic QUFJOkFBSQ==' \
  -H 'Content-Type: application/json' \
  -H 'X-FromAppId: Postman' \
  -H 'X-TransactionId: get_generic_vnf'
```

In the result, search for the **vFW_PG_VNF** and get it's vnf-id. In the payload bellow, it would be **e6fd60b4-f436-4a21-963c-cc9060127633**

```
{
    "generic-vnf": [
        {
            "vnf-id": "9663a27e-8fbe-4fde-bc33-064ae45caee6",
            "vnf-name": "vFW_SINC_VNF",
            "vnf-type": "vFW_Service/vFW_SINC 0",
            "service-id": "75af21a4-6519-4505-b418-134e9e836023",
            "prov-status": "PREPROV",
            "orchestration-status": "Created",
            "in-maint": false,
            "is-closed-loop-disabled": false,
            "resource-version": "1513788953961",
            "persona-model-id":
"edd473e1-7d08-4cf1-be31-0d705017f644",
            "persona-model-version": "1.0",
            "relationship-list": {
                "relationship": [
                    {
                        "related-to": "service-instance",
                        "related-link":
"https://10.195.197.53:30233/aai/v8/business/customers/customer/
Demonstration/service-subscriptions/service-subscription/vFWCL/s
ervice-instances/service-instance/63b55891-ebc6-40bf-b884-2e2427
280a83",
                        "relationship-data": [
                            {
                                "relationship-key":
"customer.global-customer-id",
                                "relationship-value":
"Demonstration"
                            },
                            {
                                "relationship-key":
"service-subscription.service-type",
                                "relationship-value": "vFWCL"
                            },
                            {
                                "relationship-key":
"service-instance.service-instance-id",
                                "relationship-value":
"63b55891-ebc6-40bf-b884-2e2427280a83"
                            }
                        ],
                        "related-to-property": [
                            {
                                "property-key":
"service-instance.service-instance-name",
                                "property-value":
"vFWServiceInstance-20-12"
                            }
                        ]
                    }
                ]
            },
            "vf-modules": {
                "vf-module": [
                    {
                        "vf-module-id":
"fc8ba83f-2ebf-4066-bb3a-f581667f77da",
                        "vf-module-name": "vFW_SINC_Module",
                        "heat-stack-id":
"vFW_SINC_Module/09b1a25e-4ef0-4490-9b05-d79c00c7d218",
                        "orchestration-status": "active",
                        "is-base-vf-module": true,
                        "resource-version": "1513790007998",
                        "persona-model-id":
"858e065b-7491-4c70-91e6-109a65c6102d",
                        "persona-model-version": "1"
                    }
                ]
            }
        },
        {
            "vnf-id": "e6fd60b4-f436-4a21-963c-cc9060127633",
            "vnf-name": "vFW_PG_VNF",
            "vnf-type": "vFW_Service/vFW_PG 0",
            "service-id": "75af21a4-6519-4505-b418-134e9e836023",
            "prov-status": "PREPROV",
            "orchestration-status": "Created",
            "in-maint": false,
            "is-closed-loop-disabled": false,
            "resource-version": "1513788903856",
            "persona-model-id":
"86a1bdd8-1f59-4796-bf30-3002108068f6",
            "persona-model-version": "1.0",
            "relationship-list": {
                "relationship": [
                    {
                        "related-to": "service-instance",
                        "related-link":
"https://10.195.197.53:30233/aai/v8/business/customers/customer/
Demonstration/service-subscriptions/service-subscription/vFWCL/s
ervice-instances/service-instance/63b55891-ebc6-40bf-b884-2e2427
280a83",
                        "relationship-data": [
                            {
                                "relationship-key":
"customer.global-customer-id",
                                "relationship-value":
"Demonstration"
                            },
                            {
                                "relationship-key":
"service-subscription.service-type",
                                "relationship-value": "vFWCL"
                            },
                            {
                                "relationship-key":
"service-instance.service-instance-id",
                                "relationship-value":
"63b55891-ebc6-40bf-b884-2e2427280a83"
                            }
                        ],
                        "related-to-property": [
                            {
                                "property-key":
"service-instance.service-instance-name",
                                "property-value":
"vFWServiceInstance-20-12"
                            }
                        ]
                    }
                ]
            },
            "vf-modules": {
                "vf-module": [
                    {
                        "vf-module-id":
"c2fed873-263c-46b5-bb95-4dfaf6c02410",
                        "vf-module-name": "vFW_PG_Module",
                        "heat-stack-id":
"vFW_PG_Module/850e84a4-6cee-405c-8058-7f3fa25ca42e",
                        "orchestration-status": "active",
                        "is-base-vf-module": true,
                        "resource-version": "1513791913543",
                        "persona-model-id":
"9c6c0369-a9c1-4419-94c9-aabf6250fc87",
                        "persona-model-version": "1"
```

```
        }
    ]
}
```

```
        }
    ]
}
```

b. Get the public IP address of the Packet Generator from your deployment.
c. In the bellow curl request, replace <vnf-id> with the VNF ID retrieved at step 2.a (it needs to be updated at two places), and replace <vnf-ip> with the ip retrieved at step 2.b.

```
curl -X PUT \

http://<kubernetes-host>:30230/restconf/config/network-topology:
network-topology/topology/topology-netconf/node/<vnf-id> \
  -H 'Accept: application/xml' \
  -H 'Authorization: Basic
YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrM2VIbGNzZTJnQXc4NHZhb0dHbUp2VXky
VQ==' \
  -H 'Content-Type: text/xml' \
  -d '<node xmlns="urn:TBD:params:xml:ns:yang:network-topology">
   <node-id><vnf-id></node-id>
   <host
xmlns="urn:opendaylight:netconf-node-topology"><vnf-ip></host>
   <port
xmlns="urn:opendaylight:netconf-node-topology">2831</port>
   <username
xmlns="urn:opendaylight:netconf-node-topology">admin</username>
   <password
xmlns="urn:opendaylight:netconf-node-topology">admin</password>
   <tcp-only
xmlns="urn:opendaylight:netconf-node-topology">false</tcp-only>
</node>'
```

If you want to verify the NETCONF connection has successfully being established, use the following request (replace <vnd-id> with yours

```
curl -X GET \

http://<kubernetes-host>:30230/restconf/operational/network-topo
logy:network-topology/topology/topology-netconf/node/<vnf-id> \
  -H 'Accept: application/json' \
  -H 'Authorization: Basic
YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrM2VIbGNzZTJnQXc4NHZhb0dHbUp2VXky
VQ=='
```

Result should be:

```
{
    "node": [
        {
            "node-id": "e6fd60b4-f436-4a21-963c-cc9060127633",
            "netconf-node-topology:available-capabilities": {
                "available-capability": [
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"urn:ietf:params:netconf:capability:exi:1.0"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"urn:ietf:params:netconf:capability:candidate:1.0"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"urn:ietf:params:netconf:base:1.1"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"urn:ietf:params:netconf:base:1.0"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-restconf?revision=2013-10-19)
ietf-restconf"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netconf:mdsal:n
otification?revision=2015-08-03)netconf-mdsal-notification"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring?revision=2
010-10-04)ietf-netconf-monitoring"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:TBD:params:xml:ns:yang:network-topology?revision=2013-07-1
2)network-topology"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-interfaces?revision=2014-05-0
8)ietf-interfaces"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-access-control-list?revision=
2016-07-08)ietf-access-control-list"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:honeycomb:params:xml:ns:yang:eid:mapping:context?revision=
2016-08-01)eid-mapping-context"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:rest:con
nector?revision=2014-07-24)opendaylight-rest-connector"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:binding?
revision=2013-10-28)opendaylight-md-sal-binding"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netconf:northbo
und:ssh?revision=2015-01-14)netconf-northbound-ssh"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:core:spi
:entity-ownership-service?revision=2015-08-10)opendaylight-entit
y-ownership-service"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-inet-types?revision=2013-07-1
5)ietf-inet-types"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:mdsal:core:general-entity?
revision=2015-09-30)odl-general-entity"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:protocol:framew
ork?revision=2014-03-13)protocol-framework"
```

```
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:common?r
evision=2013-10-28)opendaylight-md-sal-common"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:sal:restconf:event:subscription?revision=2014-07-08)sal-re
mote-augment"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netconf:northbo
und:notification?revision=2015-08-06)netconf-northbound-notifica
tion"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-yang-types?revision=2010-09-2
4)ietf-yang-types"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:inmemory-datast
ore-provider?revision=2014-06-17)opendaylight-inmemory-datastore
-provider"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netty?revision=
2013-11-19)netty"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:binding:
impl?revision=2013-10-28)opendaylight-sal-binding-broker-impl"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:sal:restconf:se
rvice?revision=2015-07-08)sal-restconf-service"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:config?revision
=2013-04-05)config"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:vpp:classifier?revision=20
15-06-03)vpp-classifier"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:v3po?revision=2015-01-05)v
3po"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:core:gen
eral-entity?revision=2015-08-20)general-entity"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:netconf:notification:1.0?revision=2008-
07-14)notifications"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:sample-plugin?revision=201
6-09-18)sample-plugin"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:core:spi
:config-dom-store?revision=2014-06-17)opendaylight-config-dom-da
tastore"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:core:spi
:operational-dom-store?revision=2014-06-17)opendaylight-operatio
nal-dom-datastore"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:config:netconf:
northbound:impl?revision=2015-01-12)netconf-northbound-impl"
```

```
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:vpp:nsh?revision=2016-06-2
4)vpp-nsh"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netconf:north:m
apper?revision=2015-01-14)netconf-northbound-mapper"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-yang-types?revision=2013-07-1
5)ietf-yang-types"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:honeycomb:params:xml:ns:yang:naming:context?revision=2016-
05-13)naming-context"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:iana-if-type?revision=2014-05-08)i
ana-if-type"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:vpp:vlan?revision=2015-05-
27)vpp-vlan"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netconf:mdsal:m
apper?revision=2015-01-14)netconf-mdsal-mapper"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:vpp:classifier?revision=20
16-09-09)vpp-classifier-context"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-ip?revision=2014-06-16)ietf-i
```

```
p"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:remote?r
evision=2014-01-14)sal-remote"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:lisp?revision=2016-05-20)l
isp"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-inet-types?revision=2010-09-2
4)ietf-inet-types"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:threadpool?revi
sion=2013-04-09)threadpool"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:md:sal:dom?revi
sion=2013-10-28)opendaylight-md-sal-dom"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netconf:northbo
und:tcp?revision=2015-04-23)netconf-northbound-tcp"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:v3po:context?revision=2016
-09-09)v3po-context"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:netmod:notification?revision=2008-07-14
)nc-notifications"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
```

```
"(urn:ietf:params:xml:ns:netconf:base:1.0?revision=2011-06-01)ie
tf-netconf"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ieee:params:xml:ns:yang:dot1q-types?revision=2015-06-26)do
t1q-types"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netconf:mdsal:m
onitoring?revision=2015-02-18)netconf-mdsal-monitoring"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(instance:identifier:patch:module?revision=2015-11-21)instance-
identifier-patch-module"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring-extension?
revision=2013-12-10)ietf-netconf-monitoring-extension"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:netconf:northbo
und:notification:impl?revision=2015-08-07)netconf-northbound-not
ification-impl"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:yang:extension:yang-ext?revision=2013-07-09)y
ang-ext"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-packet-fields?revision=2016-0
7-08)ietf-packet-fields"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-lisp-address-types?revision=2
015-11-05)ietf-lisp-address-types"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:config:netconf:
northbound?revision=2015-01-14)netconf-northbound"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:ietf-netconf-notifications?revisio
n=2012-02-06)ietf-netconf-notifications"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:TBD:params:xml:ns:yang:network-topology?revision=2013-10-2
1)network-topology"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:ietf:params:xml:ns:yang:rpc-context?revision=2013-06-17)rp
c-context"
                    },
                    {
                        "capability-origin": "device-advertised",
                        "capability":
"(urn:opendaylight:params:xml:ns:yang:controller:config:netconf:
auth?revision=2015-07-15)netconf-auth"
                    }
                ]
            },
            "netconf-node-topology:host": "10.195.200.32",
            "netconf-node-topology:unavailable-capabilities": {},
            "netconf-node-topology:connection-status":
"connected",
            "netconf-node-topology:port": 2831
```

```
        }
    ]
}
```

d. Using NETCONF, let's get the current streams being active in our Packet Generator. The number of stream will change along the time, this is the result of close-loop policy. When the traffic goes over a certain treashold, DCAE will publish an event on the unauthenticated.DCAE_CL_OUTPUT topic that will be picked up by APPC, that will send a NETCONF request to the paquet generator to ajust the traffic it's sending.

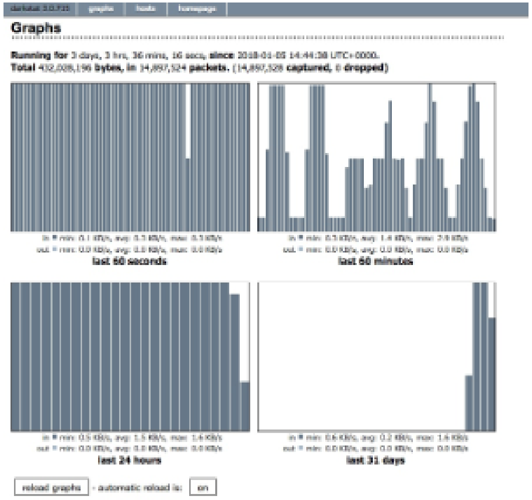
```
curl -X GET \

http://10.195.197.53:30230/restconf/config/network-topology:network-topology/topology/topology-netconf/node/e6fd60b4-f436-4a21-963c-cc9060127633/yang-ext:mount/sample-plugin:sample-plugin/pg-streams \
  -H 'Accept: application/json' \
  -H 'Authorization: Basic YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrM2VIbGNzZTJnQXc4NHZhb0dHbUp2VXkyVQ=='
```

4. Browse to the zdfw1fwl01snk01 on port 667 to see a graph representing the traffic being received:

```
http://<zdfw1fwl01snk01>:667/
```

As you can see in the bellow grah, looking at the top right square, we can see the first two fluctuations are going from very low to very high. This is when close-loop isn't running.
Once close-loop is running, you'll have somw medium bars.



Graphs

Running for 3 days, 3 hrs, 36 mins, 16 secs, **since** 2018-01-05 14:44:38 UTC+0000.
Total 432,028,156 **bytes, in** 14,897,524 **packets.** (14,897,528 **captured,** 0 **dropped**)

darkstat 3.0.719
graphs
hosts
homepage
in ■ min: 0.1 KB/s, avg: 0.3 KB/s, max: 0.3 KB/s
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s
last 60 seconds
in ■ min: 0.3 KB/s, avg: 1.4 KB/s, max: 2.9 KB/s
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s
last 60 minutes
in ■ min: 0.5 KB/s, avg: 1.5 KB/s, max: 1.6 KB/s
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s
last 24 hours
in ■ min: 0.6 KB/s, avg: 0.2 KB/s, max: 1.6 KB/s
out ■ min: 0.0 KB/s, avg: 0.0 KB/s, max: 0.0 KB/s
last 31 days
reload graphs
- automatic reload is:
on

5. Check the events sent by Virtual Event Collector (VES) to Threeshold Crossing Analytic (TCA) app:

```
curl -X GET \

http://<K8S_IP>:3904/events/unauthenticated.SEC_MEASUREMENT_OUTPUT/gr
oup1/C1 \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/cambria'
```

The VES resides in the VNF itself, whereas the TCA is an application running on Cask. A DCAE component.

6. Check the events sent by TCA on unauthenticated.DCAE_CL_OUTPUT:

```
curl -X GET \
  http://<K8S_IP>:3904/events/unauthenticated.DCAE_CL_OUTPUT/group1/C1
\
  -H 'Accept: application/json' \
  -H 'Content-Type: application/cambria'
```

Those events are the resulting of the TCA application, e.g. TCA has noticed an event was crossing a given threeshold, hence is sending a message of that particular topic. Then Policy will grab this event and perform the appropriate action, as defined in the Policy. In the case of vFWCL, Policy will send an event on the APPC_CL topic, that APPC will consume. This will trigger a NETCONF request to the packet generator to adjust the traffic.

I hope everything worked for you, if not, please leave a comment. Thanks