
HOW ONAP ORCHESTRATES A CNF TO WRCP

bin.yang@windriver.com

Contents

Overview	3
Phase 1: WRCP installation and provisions.....	4
Phase 2: CNF development and validation	5
Phase 3: ONAP installation and provisions	8
Approach 1: deploy the whole stack from OpenStack, including Heat stack, k8s cluster, and ONAP instance	8
Approach 2: deploy the ONAP instance over WRCP (StarlingX 3.0)	9
Step 1) Deploy nfs-server-provisioner	9
Step 2) Prepare namespace for ONAP deployment.....	11
Step 3) Deploy ONAP with integration override values.....	12
Step 4) Verify and checkpoints	12
ONAP provisions: Update SO configurations.....	14
ONAP Health Check.....	15
ONAP provisions: Populate demonstration data	15
Access to ONAP portals.....	15
Phase 4: Register WRCP instance to ONAP.....	17
Step 1: Create a SO Cloud Site	17
Step 2: Create an AAI Cloud Region along with complex	17
Step 4: Add service type “cfw-k8s”:.....	24
Step 5: Associate subscription to Cloud Region:.....	25
Phase 5: CNF onboarding and service design	27
Step 1: create a tar ball for the CNF helm chart	27

Step 2: Wrap helm chart tar ball into a dummy heat template artifact.....	27
Step 3: Onboard the VSP artifact 'cfwsriov1_vsp.zip' into ONAP SDC	28
Phase 6: Service Instantiation hence CNF instantiation and validation	29
Step 1, Create service instance	29
Step 2, Add node instance 'cfw1vf1' for service instance 'cfw1'	30
Step 3, Preload VNF topology	30
Step 4, prepare supplement data file	32
Step 5, Add VF Module	34
Step 6: validate the CNF deployment	34
Phase 7: Service deletion hence CNF deletion.....	36
Attachments	37
Attachment 1: integration-override.yaml.....	37
Attachment 2: onap-oom-lite.env	44
Attachment 3: base_dummy.yaml.....	52
Attachment 4: base_dummy.env.....	54
Attachment 5: Example output of ONAP Health Check.....	55
Attachment 6: Example output of populating ONAP demo data	60
Attachment 7: Dump ONAP components status	62
Debug Tips.....	68
Tip 1: Tear down ONAP instance:	68
Tip 2: Postman collections	68

Overview

This is a detailed How-To document to illustrate how users could leverage ONAP to orchestrate CNF to WRCP 19.12 instance.

The general deployment topology is depicted as diagram below:

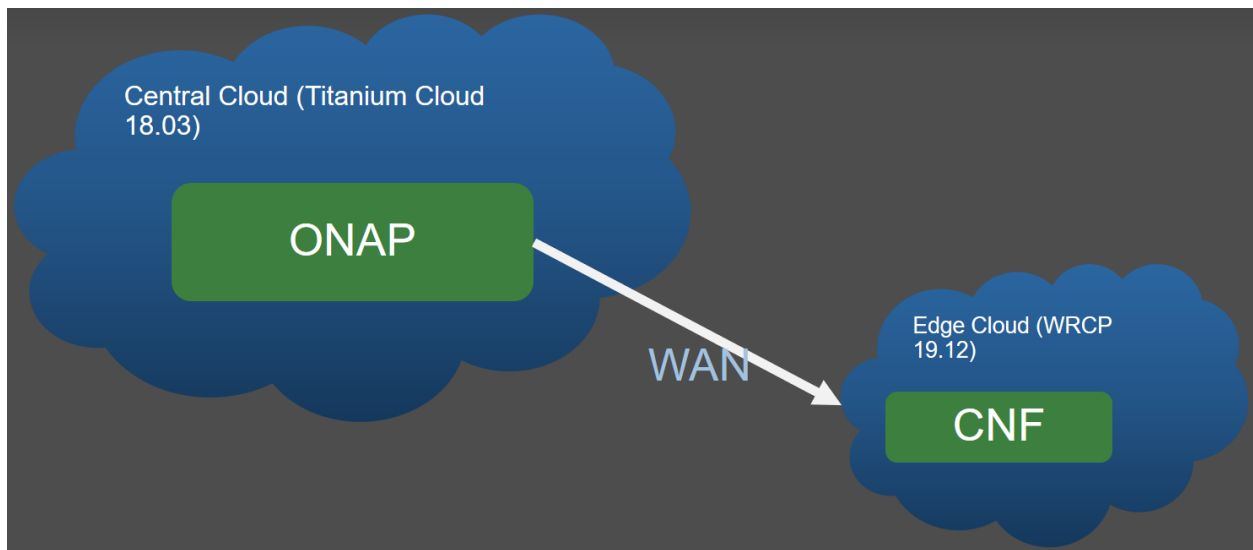


Diagram 1: ONAP and the cloud hosting it, WRCP, CNF topology

The comprehensive workflow consists of following phases:

- Phase 1: WRCP 19.12 installation and provisions
- Phase 2: CNF development and validation
- Phase 3: ONAP installation and provisions
- Phase 4: Register WRCP instance to ONAP
- Phase 5: CNF onboarding and service design
- Phase 6: Service Instantiation hence CNF instantiation and validation
- Phase 7: Service deletion hence CNF deletion

Phase 1: WRCP installation and provisions

WRCP (19.12 or later version) installation could be Duplex, and Standard type. WRCP Distributed Cloud mode is also supported.

Note: AIO Simplex is insufficient for ONAP deployment due to the limitation of maximum 110 pods per worker node

Due to CNF requiring multiple networking plane, the WRCP must be provisioned with:

- Datatnetwork backend by SRIOV netdevice (at least 2 vlan ID to support 2 network planes of the cFW use case)
 - In case This SRIOV netdevice is not available, host netdevice passthrough could be used as well.
 - veth pair could also be used for demonstration purpose, with constraint that all pods should be scheduled to the same worker node
- Hugepage-2M: 512x2M for each NUMA node for worker nodes
- Dedicated Tenant and user with admin role
- Kubernetes service account with clusterrolebindings and privileges to operate various resource, including namespace, etc.

Phase 2: CNF development and validation

The CNF should be developed and validated over WRCP 19.12 directly (without ONAP's orchestration)

The example CNF is containerized Firewall use case (referred as cFW in context below):

<https://gerrit.onap.org/r/gitweb?p=multicloud/k8s.git;a=tree;f=starlingx/demo;h=44ab83ca5c5c9f01082695b1aa9a6e71fdaeec20;hb=HEAD>

It consists of 3 pods, connected through 2 network planes. The topology is depicted as below:

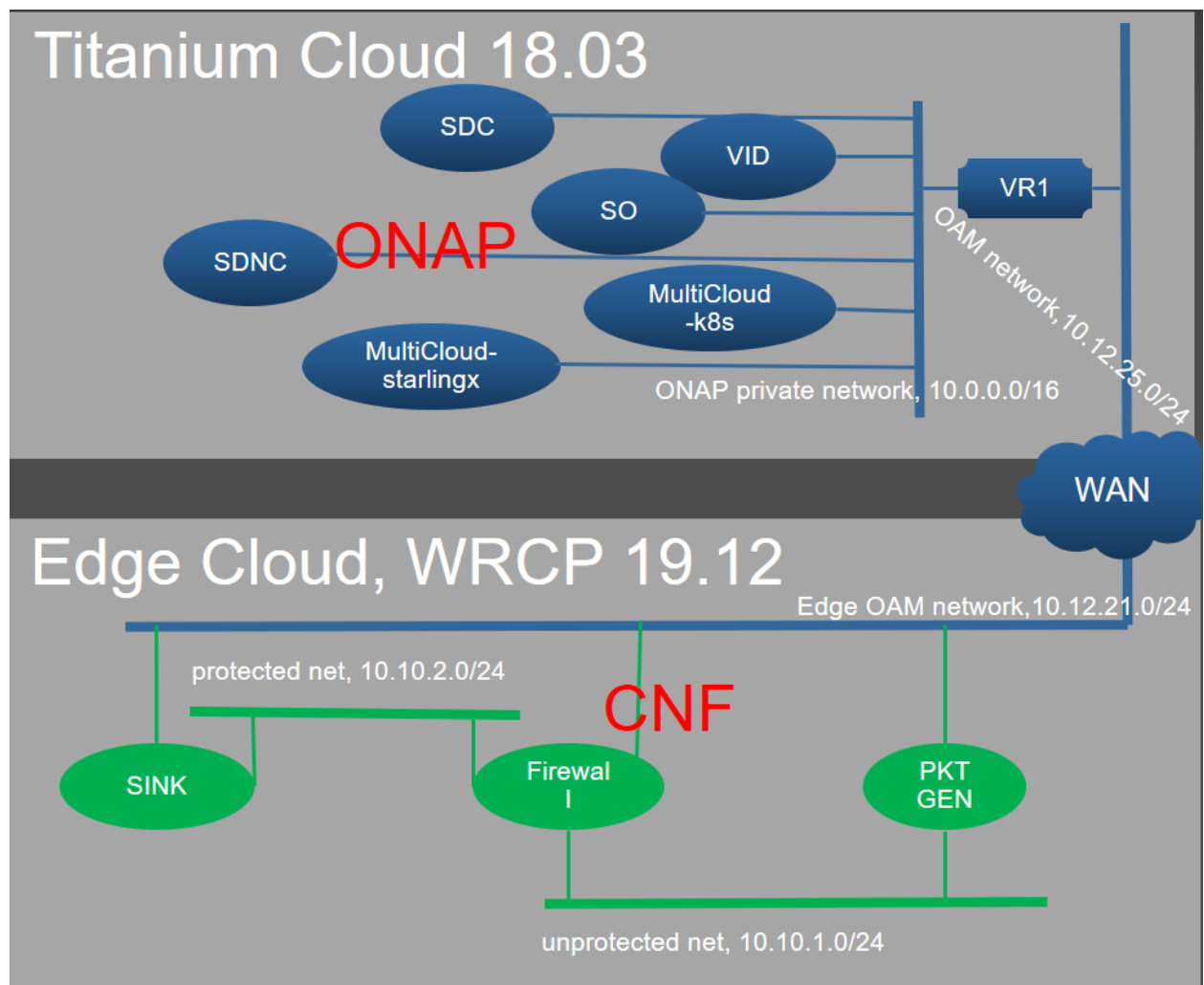


Diagram2: ONAP and cFW components

Use helm to validate the CNF, e.g. cFW:

```
$ git clone "https://gerrit.onap.org/r/multicloud/k8s"
```

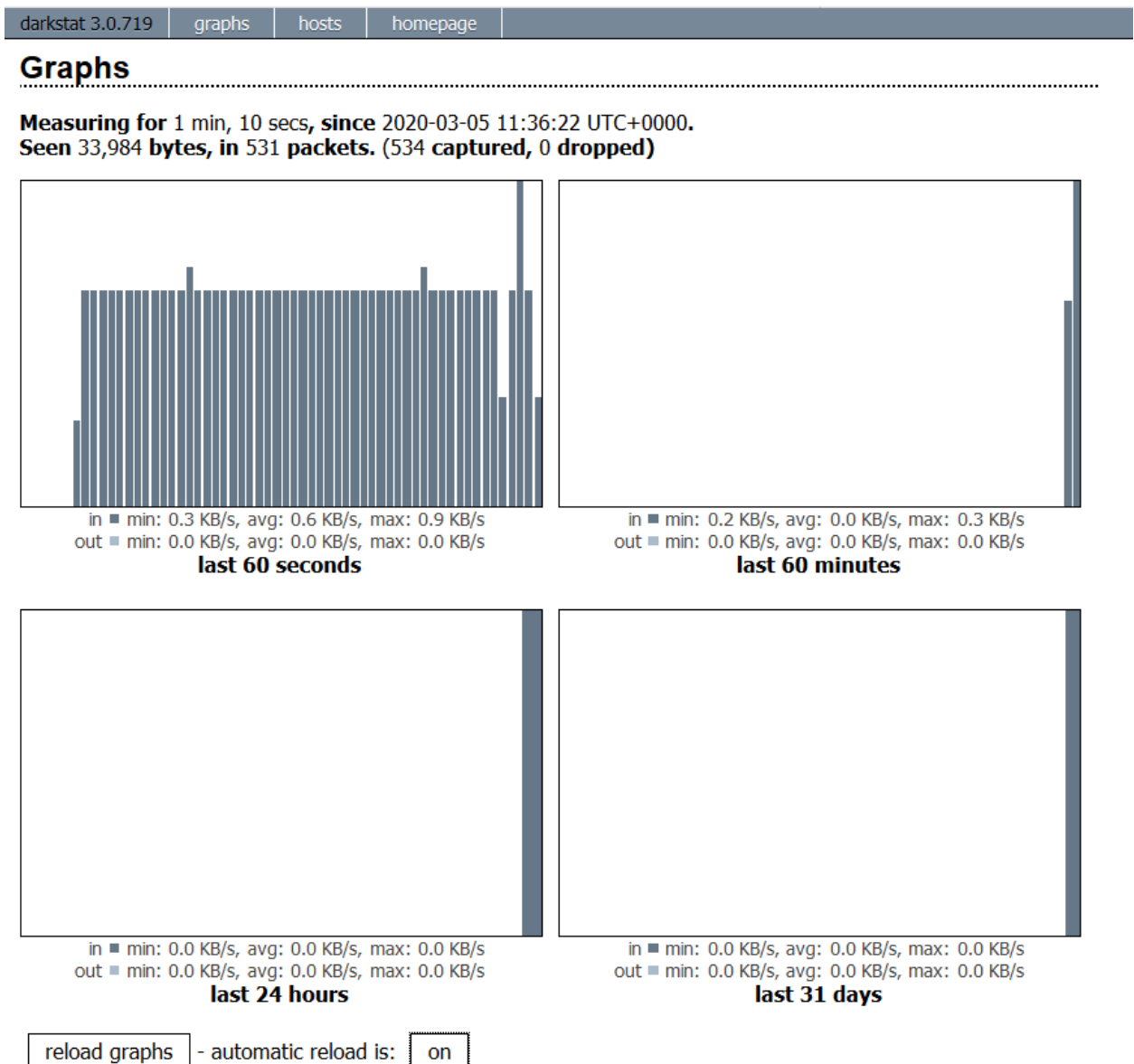
```
$ cd k8s/starlingx/demo
```

```
$ helm install firewall-sriov -n cfw1
```

To validate of the deployed cFW, monitoring the traffic over SINK pod with your browser (Chrome, Firefox) to open following url (replace NODE_IP with any worker node IP of the WRCP instance):

[http://\\$NODE_IP:30667/](http://$NODE_IP:30667/)

You should observe traffic diagram likes below:



Snapshot 1: sink traffic monitor page

Phase 3: ONAP installation and provisions

Deploy ONAP instance with el alto release, with overrides to update multicloud services

There are 2 alternative approaches to deploy ONAP instance:

Approach 1: deploy the whole stack from OpenStack, including Heat stack, k8s cluster, and ONAP instance

prepare: download openrc file from openstack horizon, e.g. VIM-openrc

update the attached [onap-oom-lite.env](#) with openstack instance's information.

ONAP deployment might take hours so it is better to initiate the process by screen terminal:

```
$ screen -R onapdeploy

$ git clone https://git.onap.org/integration

$ cd integration/deployment/heat/onap-rke/

$ copy the attached file as onap-oom-lite.env

$ source <openrc file>

$ ./scripts/deploy.sh -n 15 -s mctest1 -i elalto -o elalto
onap-oom-lite.env
```

You should observe following message from screen terminal while bootstrap process is accomplished.

```
Cloud-init v. 18.2 running 'modules:final' at Mon, 02 Mar 2020 03:24:21 +0000. Up 18.87 seconds.
Cloud-init v. 18.2 finished at Mon, 02 Mar 2020 04:17:14 +0000. Datasource DataSourceOpenStack [net,ver=2]. Up 3132.34 seconds
+ exit 0
```


Snapshot 2: ONAP deployment bootstrap accomplished

Approach 2: deploy the ONAP instance over WRCP (StarlingX 3.0)

WRCP can also host ONAP. Due to kubernetes version difference , the upstream ONAP oom charts will be updated with following aspects:

- 1) Change the API versions (for deployment, statefulset, etc.) to “apps/v1”.
- 2) Use StorageClass “nfs” (by stable/nfs-server-provisioner) for PVC requiring ReadWriteMany access mode.
- 3) Use default StorageClass “general” (by ceph rbd-provisioner) for PVC without requiring ReadWriteMany access mode.
- 4) Fixing bug

The changes above introduce so many patches to oom charts, for now, they are maintained over the forked oom repos:

<https://github.com/biny993/oom/tree/elalto-wrcp19.12>

<https://github.com/biny993/aai-oom/tree/elalto-wrcp19.12>

<https://github.com/biny993/testsuite-oom/tree/elalto-wrcp19.12>

*Note: The resource usage by ONAP installation with default override values
(<https://github.com/biny993/oom/blob/elalto-wrcp19.12/integration-override.yaml>) :*

CPU: 11 cores; Memory: 32GB, Persistence Volume: 80GB

Step 1) Deploy nfs-server-provisioner

Create ServiceAccount:

```
kubectl -n default create sa ben
```

```
kubectl create clusterrolebinding ben-admin --clusterrole cluster-admin --serviceaccount  
default:ben
```

Clone the helm charts:

```
git clone https://github.com/helm/charts.git
```

```
cd charts
```

Change the default values as below:

```
diff --git a/stable/nfs-server-provisioner/values.yaml b/stable/nfs-server-provisioner/values.yaml
index 6da0e1327..75807cdac 100644
--- a/stable/nfs-server-provisioner/values.yaml
+++ b/stable/nfs-server-provisioner/values.yaml
@@ -35,7 +35,7 @@ service:
   externalIPs: []

persistence:
- enabled: false
+ enabled: true

## Persistent Volume Storage Class
## If defined, storageClassName: <storageClass>
@@ -47,7 +47,7 @@ persistence:
   # storageClass: "-"

  accessMode: ReadWriteOnce
- size: 1Gi
+ size: 180Gi

## For creating the StorageClass automatically:
storageClass:
@@ -77,7 +77,7 @@ storageClass:

## For RBAC support:
rbac:
- create: true
+ create: false

## Ignored if rbac.create is true
##
@@ -95,4 +95,13 @@ nodeSelector: {}

tolerations: []

-affinity: {}
+affinity:
+ nodeAffinity:
+   requiredDuringSchedulingIgnoredDuringExecution:
+     nodeSelectorTerms:
+       - matchExpressions:
+         - key: kubernetes.io/hostname
+           operator: In
+         values:
```

```
+ - controller-0
+ - controller-1
```

now apply the helm charts:

```
cd stable
helm install nfs-server-provisioner --name nfsserver1 -f nfs-server-provisioner/values.yaml
```

check the storageclass "nfs":

```
$ kubectl get pods -n default
NAME                                READY STATUS RESTARTS AGE
nfsserver1-nfs-server-provisioner-0 1/1   Running 0      6d14h

$ kubectl get sc
NAME             PROVISIONER              AGE
general (default) ceph.com/rbd              75d
nfs              cluster.local/nfsserver1-nfs-server-provisioner 8d
```

Step 2) Prepare namespace for ONAP deployment

ssh to WRCP controller node, perform following commands:

```
kubectl create ns onap
```

```
source /etc/platform/openrc
```

```
cat <<EOF > rbd-namespaces.yaml
classes:
- additionalNamespaces: [default, kube-public, onap]
  chunk_size: 64
  crush_rule_name: storage_tier_ruleset
  name: general
  pool_name: kube-rbdkube-system
  replication: 1
  userId: ceph-pool-kube-rbd
  userSecretName: ceph-pool-kube-rbd
EOF
```

```
system helm-override-update --values rbd-namespaces.yaml platform-integ-apps rbd-provisioner
kube-system
```

```
system application-apply platform-integ-apps
```

Step 3) Deploy ONAP with integration override values

clone the following repo to local host, then copy it to WRCP controller node

```
git clone --recurse-submodules https://github.com/biny993/oom.git -b elalto-wrcp19.12
```

```
tar -zcvf oom-elalto-wrcp1912.tgz oom
scp oom-elalto-wrcp1912.tgz sysadmin@<WRCP OAM IP>:~/
```

login to WRCP controller node, perform following command over WRCP controller node
helm serve &

```
tar -zxvf oom-elalto-wrcp1912.tgz
cd oom
rsync -avt kubernetes/helm/plugins ~/.helm/
sed -i "\/^enabled:/a\  echo sleep 30s\n  sleep 30s" ~/.helm/plugins/deploy/deploy.sh
sed -i 's/for subchart in \*/for subchart in aaf cassandra mariadb-galera dmaap */'
~/.helm/plugins/deploy/deploy.sh

cd kubernetes
make all
cd ..

helm deploy dev local/onap -f ./kubernetes/onap/resources/environments/public-cloud.yaml -
f ./integration-override.yaml --namespace onap

rsync -avt kubernetes/helm/plugins ~/.helm/
```

Step 4) Verify and checkpoints

check statefulset, all should be ready

```
$ kubectl -n onap get sts
```

check jobs, all should be completed

```
$ kubectl -n onap get jobs.batch
```

Check pods states, all pods should be either in completed or running state, except the following one:

```
$ kubectl -n onap get pod
dev-aai-aai-graphgraph-67fdb7db7f-p8sxf      0/1   ImagePullBackOff   0      30m
```

Check PVC, all PVC should be in bound state

\$ kubectl -n onap get pvc

[Attachement 7: Dump ONAP components status](#)

ONAP provisions: Update SO configurations

(Note, this step can be skipped in case ONAP is deployed with approach 2 above)

----ONAP SO VNF Adapter Rest API endpoint version shall be set to version “v2”

```
$ kubectl -n onap get configmap | grep so-so-bpmn-infra-app-configmap
```

```
$ kubectl -n onap edit configmap dev-so-so-bpmn-infra-app-configmap
```

in the section “vnf”, modify the rest endpoint:

vnf:

endpoint: http://so-openstack-adapter.onap:8087/services/VnfAdapter

rest:

- endpoint: http://so-openstack-adapter.onap:8087/services/rest/v1/vnfs

+ endpoint: http://so-openstack-adapter.onap:8087/services/rest/v2/vnfs

volume-groups:

rest:

endpoint: http://so-openstack-adapter.onap:8087/services/rest/v1/volume-groups

```
$ kubectl get po -n onap |grep bpmn-infra
```

```
$ kubectl -n onap delete pod dev-so-so-bpmn-infra-65945c685d-cfw92
```

check if pods restarted

```
$ kubectl -n onap get po | grep so-so
```

ONAP Health Check

```
$ cd oom/kubernetes/robot/
```

```
$ ./ete-k8s.sh onap health
```

Please refer to the example output of ONAP health check: [Attachment 5: Example output of ONAP Health Check](#)

ONAP provisions: Populate demonstration data

```
$ cd oom/kubernetes/robot/
```

```
$ ./demo-k8s.sh onap init
```

Now wait about half an hour for completion of the demo data population: [Attachement 6: Example output of populating ONAP demo data](#)

In case of failure of init script, you may need populate necessary data , by curl command, refer to [Tip 2: Postman collections](#) for help curl commands in postman collection: requests 0,1,2,3,4, and populate demo VNFs by: `$./demo-k8s.sh onap distribute`

Access to ONAP portals

Update hosts with following entries: <e.g. assume 10.12.6.76 is a k8s cluster node IP>

```
10.12.6.76 portal.api.simpLEDemo.onap.org
```

```
10.12.6.76 vid.api.simpLEDemo.onap.org
```

```
10.12.6.76 sdc.api.fe.simpLEDemo.onap.org
```

```
10.12.6.76 sdc.api.be.simpLEDemo.onap.org
```

```
10.12.6.76 sdc.workflow.plugin.simpLEDemo.onap.org
```

```
10.12.6.76 sdc.dcae.plugin.simpLEDemo.onap.org
```

```
10.12.6.76 portal-sdk.simpLEDemo.onap.org
```

```
10.12.6.76 policy.api.simpLEDemo.onap.org
```

10.12.6.76 aai.api.sparky.simpdemo.onap.org

10.12.6.76 cli.api.simpdemo.onap.org

10.12.6.76 msb.api.discovery.simpdemo.onap.org

10.12.6.76 msb.api.simpdemo.onap.org

10.12.6.76 clamp.api.simpdemo.onap.org

10.12.6.76 so.api.simpdemo.onap.org

10.12.6.76 sdnc.api.simpdemo.onap.org

10.12.6.76 so-monitoring

Use a browser (Chrome, Firefox) to open the following URL, and input username/password:

<https://portal.api.simpdemo.onap.org:30225/ONAPPORTAL/login.htm>

Here is the list of users with roles, passwords defaults to “demo123456!”:

Role	User ID	Password	
designer	cs0008	demo123456!	
tester	jm0007	demo123456!	
governance Rep	gv0001	demo123456!	
ops	op0001	demo123456!	
admin	demo	demo123456!	

Note 1: The first time to access applications of ONAP portal might end up with certification error, to workaround that, you need browse the following urls and add them as security exceptions.

1, <https://sdc.api.fe.simpdemo.onap.org:30207/sdc1/portal#!/adminDashboard>

2, <https://vid.api.simpdemo.onap.org:30200/vid/welcome.htm>

Note 2: in case that so-monitoring GUI fails to show up, use browser to open the following url directly:

<http://so-monitoring:30224/>

Phase 4: Register WRCP instance to ONAP

Registering WRCP to ONAP demands following information and multiple manual steps via curl command or postman:

- Keystone endpoint URL for OpenStack API access, along with Project(Tenant) name, Domain name, User ID, Password
- Tiller endpoint URL and Service account Token for Kubernetes API access
- Figure out a cloud region ID, e.g. Cloud Owner = WRCP2, Cloud Region ID = STXRegionOne

Step 1: Create a SO Cloud Site

```
$ kubectl -n onap get pod | grep mariadb-galera

$ kubectl -n onap exec -ti dev-mariadb-galera-mariadb-galera-0 sh

mysql --user=so_admin --password=so_Admin123

USE catalogdb

select * from cloud_sites;

INSERT INTO cloud_sites(ID, REGION_ID, IDENTITY_SERVICE_ID, CLOUD_VERSION, CLLI,
ORCHESTRATOR) values("STXRegionOne", "STXRegionOne", "DEFAULT_KEystone", "2.5",
"My_Complex", "multicloud");

select * from cloud_sites;

quit

$ exit
```

Step 2: Create an AAI Cloud Region along with complex

Post following RestAPI to Create AAI Complex :

```
$ curl -X PUT \

https://aai.api.sparky.simplesdemo.onap.org:30233/aai/v16/cloud-
infrastructure/complexes/complex/My_Complex \
```

```
-H 'Accept: application/json' \
-H 'Authorization: Basic QUJJOtFBSQ==' \
-H 'Cache-Control: no-cache' \
-H 'Content-Type: application/json' \
-H 'Real-Time: true' \
-H 'X-FromAppId: jimmy-postman' \
-H 'X-TransactionId: 9999' \
-d '{
  "physical-location-id": "My_Complex",
  "data-center-code": "example-data-center-code-val-5556",
  "complex-name": "My_Complex",
  "identity-url": "example-identity-url-val-56898",
  "physical-location-type": "example-physical-location-type-val-7608",
  "street1": "example-street1-val-34205",
  "street2": "example-street2-val-99210",
  "city": "Beijing",
  "state": "example-state-val-59487",
  "postal-code": "100000",
  "country": "example-country-val-94173",
  "region": "example-region-val-13893",
  "latitude": "39.9042",
  "longitude": "106.4074",
  "elevation": "example-elevation-val-30253",
  "lata": "example-lata-val-46073"
}' -k
```

For **standalone** WRCP instance (compared to sub-cloud of WRCP Distributed Cloud), post following RestAPI request to ONAP MSB endpoint via curl command or postman (replace those placeholder marked by <>):

```
$ CLOUD_OWNER=WRCP2

$ CLOUD_REGIONID=STXRegionOne

$ curl -X PUT \

https://aai.api.sparky.simplesdemo.onap.org:30233/aai/v16/cloud-infrastructure/cloud-regions/cloud-
region/${CLOUD_OWNER}/${CLOUD_REGIONID} \

-H 'Accept: application/json' \

-H 'Authorization: Basic QUfJOkFBSQ==' \

-H 'Cache-Control: no-cache' \

-H 'Content-Type: application/json' \

-H 'Postman-Token: 8b9b95ae-91d6-4436-90fa-69cb4d2db99c' \

-H 'Real-Time: true' \

-H 'X-FromAppId: jimmy-postman' \

-H 'X-TransactionId: 9999' \

-d '{

  "cloud-owner": "WRCP2",

  "cloud-region-id": "STXRegionOne",

  "cloud-type": "openstack",

  "owner-defined-type": "t1",

  "cloud-region-version": "starlingx",

  "complex-name": "My_Complex",

  "cloud-zone": "CloudZone",

  "sriov-automation": false,

  "identity-url": "",

  "cloud-extra-info": "{\"openstack-region-id\":\"RegionOne\", \"k8s-apiserver\":\"https://<wrcp
controller IP>:6443\", \"k8s-apitoken\":\"<service account token>\"}"
```

```

"relationship-list": {
  "relationship": [
    {
      "related-to": "complex",
      "relationship-label": "org.onap.relationships.inventory.LocatedIn",
      "related-link": "/aai/v16/cloud-infrastructure/complexes/complex/My_Complex",
      "relationship-data": [
        {
          "relationship-key": "complex.physical-location-id",
          "relationship-value": "My_Complex"
        }
      ]
    }
  ]
},
"esr-system-info-list": {
  "esr-system-info": [
    {
      "esr-system-info-id": "55f97d59-6cc3-49df-8e69-926565f00055",
      "service-url": "http://<wrcp controller IP>:5000/v3",
      "user-name": "<OpenStack username>",
      "password": "<Openstack user pass>",
      "system-type": "VIM",
      "ssl-insecure": true,
      "cloud-domain": "Default",
      "default-tenant": "<OpenStack Project/Tenant name, e.g. onap-sb-01>"
    }
  ]
}

```

```
]
}
}' -k
```

For **sub-cloud** of WRCP Distributed Cloud, post following RestAPI request to ONAP MSB endpoint via curl command or postman (replace those placeholder marked by <>):

```
CLOUD_OWNER=WRCP2
```

```
CLOUD_REGIONID=STXRegionOne
```

```
OPENSTACK_REGIONID=Alameda-0000
```

```
curl -X PUT \
```

```
https://aai.api.sparky.simplesdemo.onap.org:30233/aai/v16/cloud-infrastructure/cloud-
regions/cloud-region/${CLOUD_OWNER}/${CLOUD_REGIONID} \
```

```
-H 'Accept: application/json' \
```

```
-H 'Authorization: Basic QUfJOkFBSQ==' \
```

```
-H 'Cache-Control: no-cache' \
```

```
-H 'Content-Type: application/json' \
```

```
-H 'Postman-Token: 8b9b95ae-91d6-4436-90fa-69cb4d2db99c' \
```

```
-H 'Real-Time: true' \
```

```
-H 'X-FromAppId: jimmy-postman' \
```

```
-H 'X-TransactionId: 9999' \
```

```
-d '{
```

```
  "cloud-owner": "WRCP2",
```

```
  "cloud-region-id": "STXRegionOne",
```

```
  "cloud-type": "openstack",
```

```
  "owner-defined-type": "t1",
```

```
  "cloud-region-version": "starlingx",
```

```

"complex-name": "My_Complex",

"cloud-zone": "CloudZone",

"sriov-automation": false,

"identity-url": "",

"cloud-extra-info": {"openstack-region-id": "Alameda-0000", "k8s-apiserver": "https://<wrcp
sub-cloud public OAM IP>:6443", "k8s-apitoken": "<service account token>", "isystem":
{"software_version": "19.12"}},

"relationship-list": {
  "relationship": [
    {
      "related-to": "complex",
      "relationship-label": "org.onap.relationships.inventory.LocatedIn",
      "related-link": "/aai/v16/cloud-infrastructure/complexes/complex/My_Complex",
      "relationship-data": [
        {
          "relationship-key": "complex.physical-location-id",
          "relationship-value": "My_Complex"
        }
      ]
    }
  ]
},

"esr-system-info-list": {
  "esr-system-info": [
    {
      "esr-system-info-id": "55f97d59-6cc3-49df-8e69-926565f00055",
      "service-url": "http://<System Controller public OAM IP>:5000/v3",
      "user-name": "<OpenStack username>",

```

```

    "password": "<Openstack user pass>",
    "system-type": "VIM",
    "ssl-insecure": true,
    "cloud-domain": "Default",
    "default-tenant": "<OpenStack Project/Tenant name, e.g. onap-sb-01>"
  }
]
}
}' -k

```

Step 3: Trigger MultiCloud registration process:

```

$ curl -X POST \
https://msb.api.discovery.simplesdemo.onap.org:30283/api/multicloud-
starlingx/v1/${CLOUD_OWNER}/${CLOUD_REGIONID}/registry \
-H 'Accept: application/json' \
-H 'Cache-Control: no-cache' \
-H 'Content-Type: application/json' -k

```

Step 4: associate subscription with Cloud Region

Create Customer if necessary (e.g. democustomer1) :

```

$ curl -k --location --request PUT
'https://aai.api.sparky.simplesdemo.onap.org:30233/aai/v16/business/customers/customer/democustomer1'
\
--header 'Authorization: Basic QUFJOkFBSQ==' \
--header 'X-FromAppId: AAI' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \

```

```
--header 'X-TransactionId: 808b54e3-e563-4144-a1b9-e24e2ed93d4f' \

--data-raw '{

  "global-customer-id": "democustomer1",

  "subscriber-name": "democustomer1",

  "subscriber-type": "INFRA"

}'
```

Step 4: Add service type “cfw-k8s”:

```
$ curl -k --location --request PUT 'https://aai.api.sparky.simpdemo.onap.org:30233/aai/v16/service-design-
and-creation/services/service/cfw-k8s' \
```

```
--header 'Authorization: Basic QUFJOkFBSQ==' \

--header 'X-FromAppId: AAI' \

--header 'Accept: application/json' \

--header 'X-TransactionId: 808b54e3-e563-4144-a1b9-e24e2ed93d4f' \

--header 'Content-Type: application/json' \

--data-raw '{

  "service-id": "cfw-k8s",

  "service-description": "cfw-k8s"

}'
```

```
$ curl -k --location --request PUT
'https://aai.api.sparky.simpdemo.onap.org:30233/aai/v16/business/customers/customer/democustomer1/
service-subscriptions/service-subscription/cfw-k8s' \
```

```
--header 'Authorization: Basic QUFJOkFBSQ==' \

--header 'X-FromAppId: AAI' \

--header 'Accept: application/json' \

--header 'Content-Type: application/json' \
```



```
--header 'X-TransactionId: 808b54e3-e563-4144-a1b9-e24e2ed93d4f' \

--data-raw '{

    "service-id": "cfw-k8s"

}'
```

Step 5: Associate subscription to Cloud Region:

```
$ curl -k --location --request PUT
'https://aai.api.sparky.simplesdemo.onap.org:30233/aai/v16/business/customers/customer/democustomer1/
service-subscriptions/service-subscription/cfw-k8s/relationship-list/relationship' \

--header 'Authorization: Basic QUFJOkFBSQ==' \

--header 'X-FromAppId: AAI' \

--header 'Accept: application/json' \

--header 'Content-Type: application/json' \

--header 'X-TransactionId: 808b54e3-e563-4144-a1b9-e24e2ed93d4f' \

--data-raw '{

    "related-to": "tenant",

    "related-link": "/aai/v16/cloud-infrastructure/cloud-regions/cloud-
region/WRCP2/STXRegionOne/tenants/tenant/fd32fdd20ff5467ebef2de63468eb2e4",

    "relationship-data": [

        {

            "relationship-key": "cloud-region.cloud-owner",

            "relationship-value": "WRCP2"

        },

        {

            "relationship-key": "cloud-region.cloud-region-id",

            "relationship-value": "STXRegionOne"

        },

    ],

}
```

```
{  
  "relationship-key": "tenant.tenant-id",  
  "relationship-value": "fd32fdd20ff5467ebef2de63468eb2e4"  
}  
],  
"related-to-property": [  
  {  
    "property-key": "tenant.tenant-name",  
    "property-value": "onap-sb-01"  
  }  
]  
}'
```

Phase 5: CNF onboarding and service design

Refer to Phase 2, which develops and validates the CNF (cFW in this case), the following steps should be followed to onboard it to ONAP for service design.

Step 1: create a tar ball for the CNF helm chart

```
$ cd k8s/starlingx/demo  
  
$ CNF_NAME=" cfwsriov1"  
  
$ CNF_ARTIFACT_NAME="${CNF_NAME}_cloudtech_k8s_charts.tgz"  
  
$ tar -czvf $CNF_ARTIFACT_NAME  firewall-sriov/
```

Step 2: Wrap helm chart tar ball into a dummy heat template artifact

Copy the attachment files here: [base_dummy.yaml](#) , [base_dummy.env](#)

then apply following commands:

```
$ cat <<EOF> MANIFEST.json  
  
{  
  
  "name": "",  
  
  "description": "",  
  
  "data": [  
  
    {  
  
      "file": "base_dummy.yaml",  
  
      "type": "HEAT",  
  
      "isBase": "true",  
  
      "data": [  
  
        {
```

```

        "file": "base_dummy.env",
        "type": "HEAT_ENV"
    }
]
},
{
    "file": "${CNF_ARTIFACT_NAME}",
    "type": "CLOUD_TECHNOLOGY_SPECIFIC_ARTIFACTS"
}
]
}
EOF

```

```

$ zip ${CNF_NAME}_vsp.zip base_dummy.env base_dummy.yaml MANIFEST.json
$CNF_ARTIFACT_NAME

```

Now you have the VSP artifact named 'cfwsriov1_vsp.zip' ready for onboarding to SDC

Step 3: Onboard the VSP artifact 'cfwsriov1_vsp.zip' into ONAP SDC

Browser open URL: <https://portal.api.simpLEDemo.onap.org:30225/ONAPPORtal/login.htm>

Open SDC application from ONAP portal, onboard the artifact "cfwsriov1_vsp.zip" as VSP, import it as a VF 'Cfwsriov1' (follow the instructions: <https://docs.onap.org/en/elalto/guides/onap-user/design/vfcreation/index.html>).

Create Service 'cfwsvc1' and add the VF 'cfwsriov1', test the service model, approve it, and distribute it (follow the instructions: <https://docs.onap.org/en/elalto/guides/onap-user/design/service-design/index.html>).

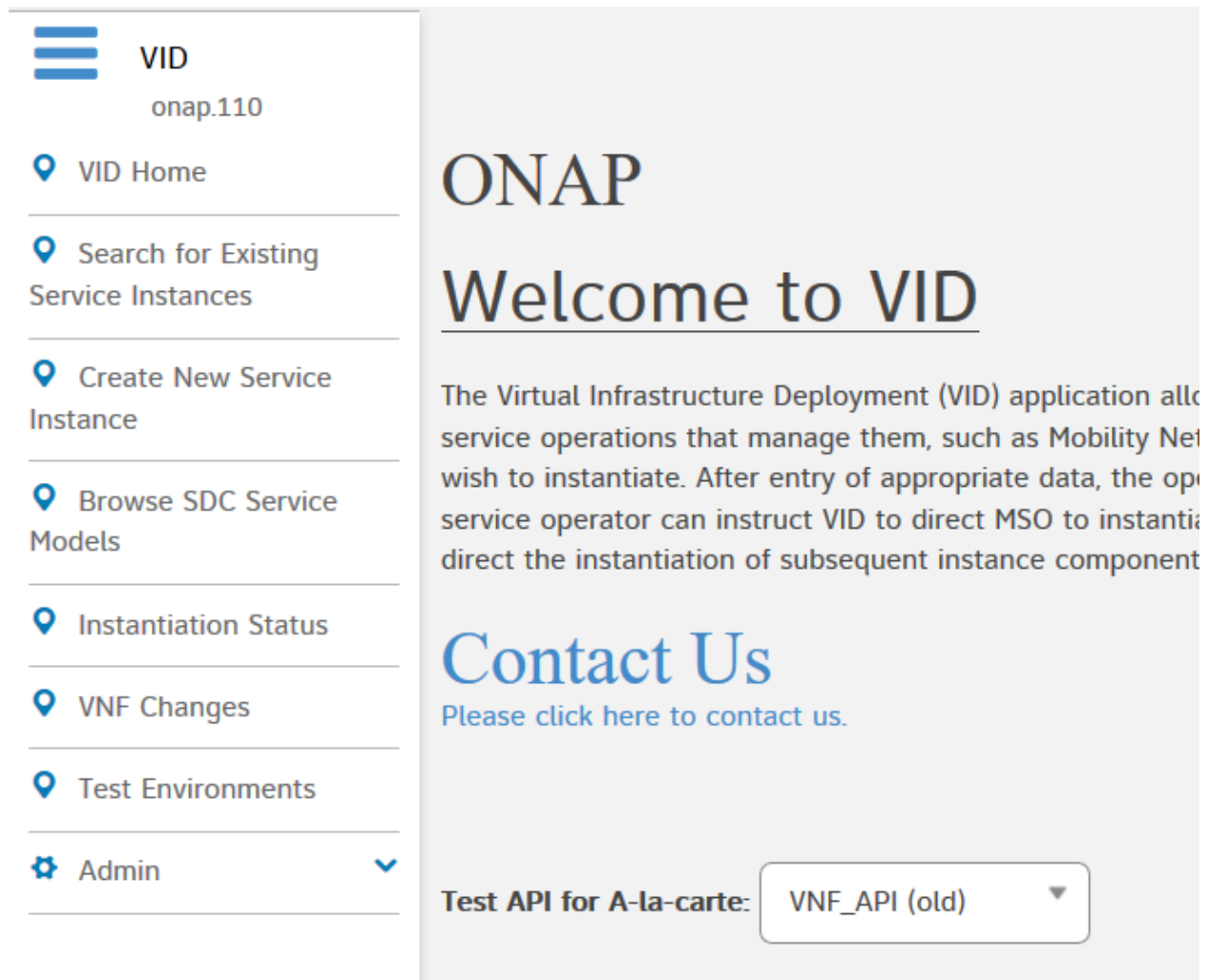
Phase 6: Service Instantiation hence CNF instantiation and validation

Once the service 'cfwsvc1' is distributed, you can instantiate it through VID application of ONAP portal.

Refer to <https://docs.onap.org/en/elalto/submodules/vid.git/docs/instantiate.html> for detailed instruction

Step 1, Create service instance

Create service instance 'cfw1' from service model 'cfwsvc1', make sure you select "VNF_API(old)" from the VID index page,



The screenshot displays the ONAP VID application interface. On the left is a sidebar with a menu icon and the text 'VID onap.110'. Below this are several menu items, each preceded by a location pin icon: 'VID Home', 'Search for Existing Service Instances', 'Create New Service Instance', 'Browse SDC Service Models', 'Instantiation Status', 'VNF Changes', 'Test Environments', and 'Admin' (which has a dropdown arrow). The main content area on the right has a light gray background. At the top, it says 'ONAP' in large letters, followed by 'Welcome to VID' in a larger, bold font. Below this is a paragraph of text: 'The Virtual Infrastructure Deployment (VID) application allows service operations that manage them, such as Mobility Net, to wish to instantiate. After entry of appropriate data, the operator can instruct VID to direct MSO to instantiate or direct the instantiation of subsequent instance components.' Further down is a section titled 'Contact Us' with the text 'Please click here to contact us.' At the bottom of the main area, there is a label 'Test API for A-la-carte:' followed by a dropdown menu currently showing 'VNF_API (old)'.

Then browse Service Models, select "cfwsvc1", click "deploy" button, assign name with 'cfw1'

Step 2, Add node instance 'cfw1vf1' for service instance 'cfw1'

Add node instance 'cfw1vf1' for service instance 'cfw1', click the menu "add node instance" to add generic vnf: 'cfw1vf1': input the generic vnf name, select the cloud region: WRCP2_STXRegion, and the tenant 'onap-sb-01', then click confirm button , wait for its completion

Step 3, Preload VNF topology

Preload VNF topology for VF Module 'cfw1vf1vfmsriov1' via curl command or postman

```
$ curl -k --location --request POST
'https://sdnc.api.simpledemo.onap.org:30267/restconf/operations/VNF-API:preload-vnf-topology-operation'
\

--header 'Accept: application/json' \

--header 'Content-Type: application/json' \

--header 'X-TransactionId: 0a3f6713-ba96-4971-a6f8-c2da85a3176e' \

--header 'X-FromAppId: API client' \

--header 'Authorization: Basic
YWRtaW46S3A4Yko0U1hzek0wV1hsaGFrM2VlbGNzZTJnQXc4NHZhb0dHbUp2VXkyVQ==' \

--data-raw '{
  "input": {
    "request-information": {
      "notification-url": "onap.org",
      "order-number": "1",
      "order-version": "1",
      "request-action": "PreloadVNFRequest",
      "request-id": "test"
    },
    "sdnc-request-header": {
      "svc-action": "reserve",
      "svc-notification-url": "http://onap.org:8080/adapters/rest/SDNCNotify",
      "svc-request-id": "test"
    }
  }
}
```

```

    },
    "vnf-topology-information": {
      "vnf-assignments": {
        "availability-zones": [],
        "vnf-networks": [],
        "vnf-vms": []
      },
      "vnf-parameters": [],
      "vnf-topology-identifier": {
        "generic-vnf-name": "cfw1vf1",
        "generic-vnf-type": "cfwsvc1/cfwsriov1 0",
        "service-type": "4ed97904-f3bc-48d9-a729-561dd9f83262",
        "vnf-name": " cfw1vf1vfmsriov1'",
        "vnf-type": "Cfwsriov1..base_dummy..module-0"
      }
    }
  }
}
}'

```

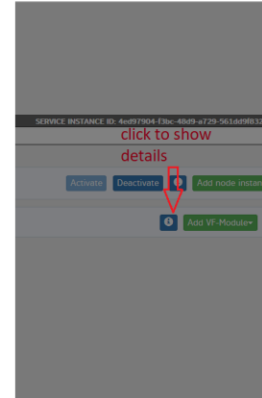
Check the following example about how to populate the VNF topology request body:

```
curl -k -location -request POST 'https://sdmc.api.simplifiedmomo.comap.org:30267/testcont/open' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'X-TransactionId: ba5f711b-ba56-4971-a6f8-c22a85a317de' \
--header 'X-FromAppId: API client' \
--header 'X-Authentication: Basic VXRjM2Q5MzR4bWVudD1hbnRkblU1bG9hGFRMPWVhcnV2b3R0cXZlG4NHRhZG90b3R0b3R0' \
--data-raw '{
  "input": {
    "request-information": {
      "notification-url": "cnap.org",
      "order-number": "1",
      "order-version": "1",
      "request-action": "PreloadONTRequest",
      "request-id": "test"
    },
    "sdmc request header": {
      "svc-action": "reserve",
      "svc-notification-url": "http://cnap.org:8030/adapters/test/SDMNotify",
      "svc-request-id": "test"
    },
    "vnf-topology-information": {
      "vnf-assignments": {
        "availability-zones": [],
        "vnf-networks": [],
        "vnf-vms": []
      },
      "vnf-parameters": [],
      "vnf-topology-identifier": {
        "generic-vnf-name": "cf5bcv4vtf",
        "generic-vnf-type": "cf5bcv1/cf5bcv1 0",
        "service-type": "4e5b1b/4-e-5bc-4929-a23-3a6059b83262",
        "vnf-name": "cf5bcv4vtfvmln1r",
        "vnf-type": "cf5bcv1_base_dummy_module-0"
      }
    }
  }
}
```

Service Instance Details

Subscriber Name:	democustomer1
Service Instance ID:	fed97904-f3bc-48d9-a729-561dd9f83262
Service Type:	vfw-k8s
Model Name:	chw1srv
Model Version:	1.0

Virtual Network Function Details	
VNF ID:	74434183-726e-4dd8-8ed6-48a3e9e570a
VNF Name:	cfwsvc4v1
VNF Type:	cfw1srv/cfwsvriov 1
Service ID:	vw-k8s
Prov Status:	PREPROV
Orchestration Status:	Created
In Maint:	false
Is Closed Loop Disabled:	false
Resource Version:	1583145128851
Model ID:	4d972da-e651-4a0f-9506-5dab8160c428
Model Version ID:	cce930a0-231a-4d4d-8793-3ac2978a63



Snapshot 3: populate VNF Topology request data, generic VNF and service type


```
% -k --location --request POST 'https://sdmc.api.simplenemo.omap.org:30267/restcon/ops/  
header: Accept: application/json"  
header: Content-Type: application/json"  
header: X-transactionid: da3f7e13-ba56-491f-a6f8-c3aa85a137ee"  
header: X-FrontAppId: AFI client"  
header: Authorization: Basic VWR=ZM46S3A4Ykd0MTI0ZWk0VHlwZXQwPmF0bWVkb3R0c3R0dGZlNHRhbnRpbG9y  
data-csv: "  
"input": {  
  "request-information": {  
    "notification-url": "cnap.crg",  
    "order-number": "1",  
    "order-version": "1",  
    "request-action": "PreloadVNFRequest",  
    "request-id": "test"  
  },  
  "sdnc request header": {  
    "svc-action": "reserve",  
    "svc-notification-url": "http://cnap.crg:8030/adapters/vres//SDNCNotify",  
    "svc-request-id": "test"  
  },  
  "vnf-topology-information": {  
    "vmn-assignments": {  
      "availability-zones": [],  
      "vnf-networks": [],  
      "vnf-vms": []  
    },  
    "vnf-parameters": [],  
    "vnf-topology-identifier": {  
      "generic-vnf-name": "Cfswscv4vfl",  
      "generic-vnf-type": "Cfswscvl/Cfswscivl 0",  
      "service-type": "4ed979/4-23dc-4789-9-3c3-beleday85g-62",  
      "vnf-name": "Cfswscv4vlfmlr",  
      "vnf-type": "Cfswscivl.base.dummy.module-0"
```

```

Create VF Module -- a la carte

Service Name:                cfw1srv
Subscriber Name:             democustomer1
Service Instance Name:      cfwsv4c
Model Name:                  cfwsvr1_base_dummy_module-0
Model Invariant UUID:        4b251f34-4434-4ba2-bd99-29701c06548f
Model Version:               1
Model UUID:                  26254fac-14f9-4dfe-9903-18bb822d7f3b
Model Customization UUID:    432bc1f3-9300-41ae-a665-a1804ac28dc6

```



User Provided Data (* indicates required field)

Instance Name: *

LCP Region: *

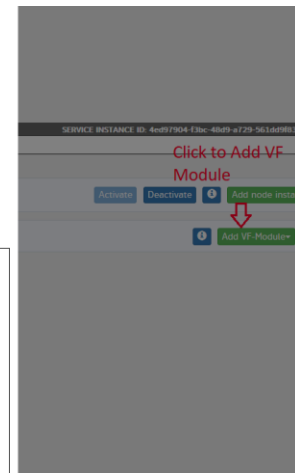
Tenant: *

Suppress Rollback on Failure: ☒

SDN-C Pre-Load: ☐

Upload Supplementary Data file: ☒

Supplementary Data file (JSON format): supplemental2.json



Snapshot 4: populate VNF Topology request data, VF module part

Note, do not click “Confirm” button until runs to step 5 below.

Step 4, prepare supplement data file

Prepare `override_values.yaml` which override the helm charts' values, encode the file content and put it into `supplemental1.json` for VF Module 'cfw1vf1vfmsriov1'

```
$ cat << EOF > override values.yaml
```

global:

unprotectedNetProviderVlan: 29

protectedNetProviderVlan: 30

nodeAffinity:

- label:

- labelkey: sriovdp

- op: In

- labelvalues:

- enabled

- label:

- labelkey: kube-cpu-mgr-policy

- op: In

- labelvalues:

- static

EOF

\$ OVERRIDE_VALUES_YAML_BASE64=`cat override_values.yaml | base64 -w 0`

\$ cat <<EOF > supplemental1.json

```
[
{
  "name": "definition-name",
  "value": "Cfwsriov1..base_dummy..module-0"
},
{
  "name": "definition-version",
  "value": "1"
},
]
```

```

{
  "name": "profile-name",
  "value": "p1"
},
{
  "name": "template_type",
  "value": "heat"
},
{
  "name": "override_values_yaml_base64",
  "value": "$OVERRIDE_VALUES_YAML_BASE64"
}
]
EOF

```

Step 5, Add VF Module

Add VF Module 'cfw1vf1vfmsriov1' for node instance 'cfw1vf1'

From the popup dialog window, input the VF module name, select the cloud region: WRCP2_STXRegion, and the tenant 'onap-sb-01', check on the following options:

- SDN-C Pre-Load
- Upload Supplementary Data file

Then upload the file 'supplemental1.json', click "confirm" button to start the VF module creation process and wait for its completion. This process will orchestrate cFW helm charts to WRCP instance. Hence you can check if the workload is deployed over WRCP instance with kubectl commands as well.

Step 6: validate the CNF deployment

Execute kubectl to check if deployments are there, check the pods status

```
$ kubectl get deployments -o wide --all-namespaces | grep firewall
```

```
$ kubectl get pods -o wide --all-namespaces | grep firewall
```

Monitor the traffic over SINK pod via browser: [http://\\$NODE_IP:30667/](http://$NODE_IP:30667/), you should observe the traffic statistics from the page. Refer to snapshot 1 for details.

Phase 7: Service deletion hence CNF deletion

With VID application GUI, Navigate to the deployed service instance 'cfw1' , delete VF module first, then delete node instance, then delete service.

Attachments

Attachment 1: integration-override.yaml

global:

repository: nexus3.onap.org:10001

pullPolicy: IfNotPresent

robot:

enabled: true

flavor: large

appcUsername: "appc@appc.onap.org"

appcPassword: "demo123456!"

openStackKeyStoneUrl: "<keystone endpoint, e.g. http://10.12.25.2:5000>"

openStackPublicNetId: "<tenant network UUID for public access,e.g. 971040b2-7059-49dc-b220-4fab50cb2ad4>"

openStackTenantId: "<tenant ID of openstack user, e.g. 0e148b76ee8c42f78d37013bf6b7b1ae>"

openStackUserName: "<tenant username, e.g. demo>"

openStackUserDomain: "Default"

openStackProjectName: "VIM"

ubuntu14Image: "<glance image name for ubuntu14, e.g. ubuntu-14-04-cloud-amd64>"

ubuntu16Image: "<glance image name for ubuntu16, e.g. ubuntu-16-04-cloud-amd64>"

openStackPrivateNetId: "eda70926-a53f-458c-a621-40a64e72643d"

openStackPrivateSubnetId: "4ef0889a-406f-488b-934e-52b3ad6aef3a"

openStackPrivateNetCidr: "10.0.0.0/16"

openStackSecurityGroup: "aa534410-959e-4c40-9480-b3ae5ec1d8d8"

openStackOamNetworkCidrPrefix: "10.0"

dcaeCollectorIp: "10.12.6.149"

kubernetesExternallp: "10.12.6.149"

vnfPubKey: "<public key for ssh access to vnf>"

demoArtifactsVersion: "1.6.0-SNAPSHOT"

demoArtifactsRepoUrl: "https://nexus.onap.org/content/repositories/releases"

scriptVersion: "1.6.0-SNAPSHOT"

nfsIpAddress: "10.12.6.253"

config:

openStackEncryptedPasswordHere: "bbaef6cd76625ab9eb60deedeae7dbb9"

openStackSoEncryptedPassword: ""

so:

enabled: true

so-catalog-db-adapter:

config:

openStackUserName: "<tenant username, e.g. demo>"

openStackKeyStoneUrl: "<keystone endpoint, e.g. http://10.12.25.2:5000>"

openStackEncryptedPasswordHere: ""

so-bpmn-infra:

config:

appc:

enabled: true

replicaCount: 3

config:

enableClustering: true

openStackType: "OpenStackProvider"

openStackName: "OpenStack"

openStackKeyStoneUrl: "<keystone endpoint, e.g. http://10.12.25.2:5000/v2.0>"

openStackServiceTenantName: "VIM"

openStackDomain: "Default"

openStackUserName: "<tenant username, e.g. demo>"

openStackEncryptedPassword: "<tenant user password>"

mariadb-galera:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

sdnc:

enabled: true

replicaCount: 3

config:

enableClustering: true

mariadb-galera:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

aai:

enabled: true

liveness:

initialDelaySeconds: 120

aai-data-router:

liveness:

initialDelaySeconds: 120

aai-sparky-be:

liveness:

initialDelaySeconds: 120

aai-spike:

liveness:

initialDelaySeconds: 120

aai-cassandra:

replicaCount: 3

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

portal:

enabled: true

portal-cassandra:

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

vid:

enabled: true

aaf:

enabled: true

cassandra:

enabled: true

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

clamp:

enabled: true

cli:

enabled: true

consul:

enabled: true

contrib:

enabled: true

dcaegen2:

enabled: false

dmaap:

enabled: true

dmaap-dr-prov:

mariadb:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

esr:

enabled: true

log:

enabled: false

log-logstash:

replicaCount: 1

sniro-emulator:

enabled: true

oof:

enabled: true

oof-has:

music:

music-cassandra:

replicaCount: 3

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

music-tomcat:

replicaCount: 1

mariadb-galera:

enabled: true

liveness:

initialDelaySeconds: 180

periodSeconds: 60

modeling:

enabled: true

mariadb-galera:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

msb:

enabled: true

multicloud:

enabled: true

image: onap/multicloud/framework:1.5.1

multicloud-starlingx:

image: onap/multicloud/openstack-starlingx:1.5.5

multicloud-k8s:

image: onap/multicloud/k8s:0.5.0

nbi:

enabled: false

policy:

enabled: true

pomba:

enabled: false

sdc:

enabled: true

sdc-cs:

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

sdc-be:

liveness:

initialDelaySeconds: 120

periodSeconds: 120

timeoutSeconds: 15

readiness:

initialDelaySeconds: 120

periodSeconds: 120

timeoutSeconds: 15

sdc-fe:

livenessProbe:

initialDelaySeconds: 120

periodSeconds: 120

timeoutSeconds: 15

readinessProbe:

initialDelaySeconds: 120

periodSeconds: 120

timeoutSeconds: 15

uui:

enabled: false

vfc:

enabled: false

mariadb-galera:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

vnfsdk:

enabled: false

cds:

enabled: true

Attachment 2: onap-oom-lite.env

parameters:

ubuntu_1804_image: <glance image name for ubuntu16, e.g. ubuntu-18.04>

apt_proxy: ""

docker_proxy: nexus3.onap.org:10001

nfs_vm_flavor: m1.lm.xlarge

k8s_vm_flavor: m1.lm.xlarge

orch_vm_flavor: m1.lm.medium

public_net_id: <tenant network UUID for public access,e.g. 971040b2-7059-49dc-b220-4fab50cb2ad4>

oam_network_cidr: 10.0.0.0/16

oam_ext_network_cidr: 10.100.0.0/16

integration_gerrit_branch: master

helm_deploy_delay: 30s

integration_override_yaml: >

global:

repository: __docker_proxy__

pullPolicy: IfNotPresent

robot:

enabled: true

flavor: large

appcUsername: "appc@appc.onap.org"

appcPassword: "demo123456!"

openStackKeyStoneUrl: "<keystone endpoint, e.g. http://10.12.25.2:5000>"

openStackPublicNetId: "__public_net_id__"

openStackTenantId: "\${OS_PROJECT_ID}"

openStackUserName: "\${OS_USERNAME}"

openStackUserDomain: "\${OS_USER_DOMAIN_NAME}"

openStackProjectName: "\${OS_PROJECT_NAME}"

ubuntu14Image: "<glance image name for ubuntu14, e.g. ubuntu-14-04-cloud-amd64>"

ubuntu16Image: "<glance image name for ubuntu16, e.g. ubuntu-16-04-cloud-amd64>"

openStackPrivateNetId: "__oam_network_id__"

openStackPrivateSubnetId: "__oam_subnet_id__"

openStackPrivateNetCidr: "__oam_network_cidr__"

openStackSecurityGroup: "__sec_group__"

openStackOamNetworkCidrPrefix: "10.0"

dcaeCollectorIp: "__k8s_01_vm_ip__"

kubernetesExternalIp: "__k8s_01_vm_ip__"

vnfPubKey: "<public key for ssh access to vnf>"

demoArtifactsVersion: "1.6.0-SNAPSHOT"

demoArtifactsRepoUrl: "https://nexus.onap.org/content/repositories/releases"

scriptVersion: "1.6.0-SNAPSHOT"

nfsIpAddress: "__nfs_ip_addr__"

config:

openStackEncryptedPasswordHere: "\${OS_PASSWORD_ENCRYPTED_FOR_ROBOT}"

openStackSoEncryptedPassword: "\${OS_PASSWORD_ENCRYPTED}"

so:

enabled: true

so-catalog-db-adapter:

config:

openStackUserName: "\${OS_USERNAME}"

openStackKeyStoneUrl: "<keystone endpoint, e.g. http://10.12.25.2:5000/v2.0>"

openStackEncryptedPasswordHere: "\${OS_PASSWORD_ENCRYPTED}"

appc:

enabled: true

replicaCount: 3

config:

enableClustering: true

openStackType: "OpenStackProvider"

openStackName: "OpenStack"

openStackKeyStoneUrl: "<keystone endpoint, e.g. http://10.12.25.2:5000/v2.0>"

openStackServiceTenantName: "\${OS_PROJECT_NAME}"

openStackDomain: "\${OS_USER_DOMAIN_NAME}"

openStackUserName: "\${OS_USERNAME}"

openStackEncryptedPassword: "\${OS_PASSWORD}"

mariadb-galera:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

sdnc:

enabled: true

replicaCount: 3

config:

enableClustering: true

mariadb-galera:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

aai:

enabled: true

liveness:

initialDelaySeconds: 120

aai-data-router:

liveness:

initialDelaySeconds: 120

aai-sparky-be:

liveness:

initialDelaySeconds: 120

aai-spike:

liveness:

initialDelaySeconds: 120

aai-cassandra:

replicaCount: 3

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

portal:

enabled: true

portal-cassandra:

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

vid:

enabled: true

aaf:

enabled: true

cassandra:

enabled: true

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

clamp:

enabled: true

cli:

enabled: true

consul:

enabled: true

contrib:

enabled: true

dcaegen2:

enabled: false

dmaap:

enabled: true

dmaap-dr-prov:

mariadb:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

esr:

enabled: true

log:

enabled: false

log-logstash:

replicaCount: 1

sniro-emulator:

enabled: true

oof:

enabled: true

oof-has:

music:

music-cassandra:

replicaCount: 3

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

music-tomcat:

replicaCount: 1

mariadb-galera:

enabled: true

liveness:

initialDelaySeconds: 180

periodSeconds: 60

modeling:

enabled: true

mariadb-galera:

liveness:

initialDelaySeconds: 180

periodSeconds: 60

msb:

enabled: true

multicloud:

enabled: true

image: onap/multicloud/framework:1.5.1

multicloud-starlingx:

image: onap/multicloud/openstack-starlingx:1.5.5

multicloud-k8s:

image: onap/multicloud/k8s:0.5.0

nbi:

enabled: false

policy:

enabled: true

pomba:

enabled: false

sdc:

enabled: true

sdc-cs:

liveness:

periodSeconds: 120

readiness:

periodSeconds: 60

sdc-be:

liveness:

initialDelaySeconds: 120

periodSeconds: 120

timeoutSeconds: 15

readiness:

initialDelaySeconds: 120

periodSeconds: 120

```
    timeoutSeconds: 15
sdc-fe:
  livenessProbe:
    initialDelaySeconds: 120
    periodSeconds: 120
    timeoutSeconds: 15
  readinessProbe:
    initialDelaySeconds: 120
    periodSeconds: 120
    timeoutSeconds: 15
uui:
  enabled: false
vfc:
  enabled: false
mariadb-galera:
  liveness:
    initialDelaySeconds: 180
    periodSeconds: 60
vnfsdk:
  enabled: false
cds:
  enabled: true
```

Attachment 3: base_dummy.yaml

```
# #=====LICENSE_START=====
```

##

Copyright (C) 2020 Wind River System Inc.

SPDX-License-Identifier: Apache-2.0

##

##=====LICENSE_END=====

heat_template_version: 2016-10-14

description: Heat template to deploy dummy VNF

parameters:

dummy_name_0:

type: string

label: name of vm

description: Dummy name

vnf_id:

type: string

label: id of vnf

description: Provided by ONAP

vnf_name:

type: string

label: name of vnf

description: Provided by ONAP

vf_module_id:

type: string

label: vnf module id

description: Provided by ONAP

dummy_image_name:

type: string

label: Image name or ID

description: Dummy image name

dummy_flavor_name:

type: string

label: flavor

description: Dummy flavor

resources:

dummy_0:

type: OS::Nova::Server

properties:

name: { get_param: dummy_name_0 }

image: { get_param: dummy_image_name }

flavor: { get_param: dummy_flavor_name }

metadata: { vnf_name: { get_param: vnf_name }, vnf_id: { get_param: vnf_id }, vf_module_id:
{ get_param: vf_module_id }}

Attachment 4: base_dummy.env

parameters:

vnf_id: PROVIDED_BY_ONAP

vnf_name: PROVIDED_BY_ONAP

vf_module_id: PROVIDED_BY_ONAP

dummy_name_0: dummy_1_0

dummy_image_name: dummy

dummy_flavor_name: dummy.default

Attachment 5: Example output of ONAP Health Check

```
oom/kubernetes/robot$ ./ete-k8s.sh onap health > h.1
```

```
oom/kubernetes/robot$ cat h.1
```

Starting Xvfb on display :90 with res 1280x1024x24

Executing robot tests at log level TRACE

=====
Testsuites

=====
Testsuites.Health-Check :: Test that ONAP components are available via basi...
=====

Basic A&AI Health Check | PASS |

Basic AAF Health Check | PASS |

Basic AAF SMS Health Check | PASS |

Basic APPC Health Check | FAIL |

ConnectionError: HTTPSConnectionPool(host='appc.onap', port=8443): Max retries exceeded with url: /restconf/operations/SLI-API:healthcheck (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f90095faad0>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic CLI Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='cli.onap', port=8080): Max retries exceeded with url: / (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f900445f190>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic CLAMP Health Check | FAIL |

ConnectionError: HTTPSConnectionPool(host='clamp.onap', port=8443): Max retries exceeded with url: /restservices/clds/v1/healthcheck (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f9004d21710>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic DCAE Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='dcae-healthcheck.onap', port=80): Max retries exceeded with url: /healthcheck (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f9004467bd0>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic DMAAP Data Router Health Check | PASS |

Basic DMAAP Message Router Health Check | PASS |

Basic DMAAP Bus Controller Health Check With Basic Auth | PASS |

Basic External API NBI Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='nbi.onap', port=8080): Max retries exceeded with url: /nbi/api/v4/status (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f9004d21a10>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Log Elasticsearch Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='log-es.onap', port=9200): Max retries exceeded with url: / (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f900445f210>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Log Kibana Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='log-kibana.onap', port=5601): Max retries exceeded with url: / (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f90061a7110>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Log Logstash Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='log-ls-http.onap', port=9600): Max retries exceeded with url: / (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f9004d2ef90>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Microservice Bus Health Check | PASS |

Basic Multicloud API Health Check | PASS |

Basic Multicloud-pike API Health Check | PASS |

Basic Multicloud-starlingx API Health Check | PASS |

Basic Multicloud-titanium_cloud API Health Check | PASS |

Basic Multicloud-vio API Health Check | PASS |

Basic Multicloud-k8s API Health Check | PASS |

Basic OOF-Homing Health Check | FAIL |

ConnectionError: HTTPSConnectionPool(host='oof-has-api.onap', port=8091): Max retries exceeded with url: /v1/plans/healthcheck (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f9004441090>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic OOF-SNIRO Health Check | FAIL |

ConnectionError: HTTPSConnectionPool(host='oof-osdf.onap', port=8698): Max retries exceeded with url: /api/oof/v1/healthcheck (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f9004441e50>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic OOF-CMSO Health Check | FAIL |

ConnectionError: HTTPSConnectionPool(host='oof-cmsd.onap', port=8080): Max retries exceeded with url: /cmsd/v1/health?checkInterfaces=true (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f9004badad0>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Policy Health Check | FAIL |

ConnectionError: HTTPSConnectionPool(host='drools.onap', port=6969): Max retries exceeded with url: /healthcheck (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f9004456450>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba AAI-context-builder Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='pomba-aaictxbuilder.onap', port=9530): Max retries exceeded with url: /aaicontextbuilder/health (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f9004be3e10>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba SDC-context-builder Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='pomba-sdcctxbuilder.onap', port=9530): Max retries exceeded with url: /sdcontextbuilder/health (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f90043eb110>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba Network-discovery-context-builder Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='pomba-networkdiscoveryctxbuilder.onap', port=9530): Max retries exceeded with url: /ndcontextbuilder/health (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f90043eb410>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba Service-Decomposition Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='pomba-servicedecomposition.onap', port=9532): Max retries exceeded with url: /service-decomposition/health (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f90043f2b10>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba Network-Discovery-MicroService Health Check | FAIL |

ConnectionError: HTTPSConnectionPool(host='pomba-networkdiscovery.onap', port=9531): Max retries exceeded with url: /health (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f9004407090>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba Pomba-Kibana Health Check | FAIL |

ConnectionError: HTTPSConnectionPool(host='pomba-kibana.onap', port=5601): Max retries exceeded with url: / (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f9004441fd0>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba Elastic-Search Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='pomba-es.onap', port=9200): Max retries exceeded with url: / (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f9004414290>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba Sdnc-Context-Builder Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='pomba-sdncctxbuilder.onap', port=9530): Max retries exceeded with url: /sdnccontextbuilder/health (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f90043eb650>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Pomba Context-Aggregator Health Check | FAIL |

ConnectionError: HTTPConnectionPool(host='pomba-contextaggregator.onap', port=9529): Max retries exceeded with url: /health (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f900439db10>: Failed to establish a new connection: [Errno -2] Name or service not known',))

Basic Portal Health Check | PASS |

Basic SDC Health Check (DMaaP:UP)
| PASS |

Basic SDNC Health Check | PASS |

Basic SO Health Check | PASS |

Basic UseCaseUI API Health Check | FAIL |
502 != 200

Basic VFC catalog API Health Check | FAIL |
502 != 200

Basic VFC emsdriver API Health Check | FAIL |
502 != 200

Basic VFC gvnfmdriver API Health Check | FAIL |
502 != 200

Basic VFC huaweivnfmdriver API Health Check | FAIL |
502 != 200

Basic VFC jujuvnfmdriver API Health Check | FAIL |
502 != 200

Basic VFC multivimproxy API Health Check | FAIL |
502 != 200

Basic VFC nokiav2driver API Health Check | FAIL |
502 != 200

Basic VFC nslcm API Health Check | FAIL |
502 != 200

Basic VFC resmgr API Health Check | FAIL |
502 != 200

Basic VFC vnflcm API Health Check | FAIL |
502 != 200

Basic VFC vnfmgr API Health Check | FAIL |
502 != 200

Basic VFC vnfres API Health Check | FAIL |
502 != 200

Basic VFC workflow API Health Check | FAIL |
502 != 200

Basic VFC ztesdncdriver API Health Check | FAIL |
502 != 200

Basic VFC ztevnfmdriver API Health Check | FAIL |
502 != 200

Basic VID Health Check | PASS |

Basic VNFSDK Health Check | FAIL |
ConnectionError: HTTPConnectionPool(host='refrepo.onap', port=8702): Max retries exceeded with url: /onapapi/vnfsdk-marketplace/v1/PackageResource/healthcheck (Caused by

```
NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f90042c86d0>: Failed to
establish a new connection: [Errno -2] Name or service not known',))
```

```
-----
Basic Holmes Rule Management API Health Check | FAIL |
502 != 200
```

```
-----
Basic Holmes Engine Management API Health Check | FAIL |
502 != 200
```

```
-----
Basic Multicloud-fcaps API Health Check | PASS |
```

```
-----
Basic Modeling genericparser API Health Check | FAIL |
502 != 200
```

```
-----
Basic CDS Health Check | FAIL |
ConnectionError: HTTPConnectionPool(host='cds-blueprints-processor-http.onap', port=8080):
Max retries exceeded with url: /api/v1/execution-service/health-check (Caused by
NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f9004bc8750>: Failed to
establish a new connection: [Errno -2] Name or service not known',))
```

```
-----
Testsuites.Health-Check :: Test that ONAP components are available... | FAIL |
61 critical tests, 19 passed, 42 failed
61 tests total, 19 passed, 42 failed
```

```
=====
Testsuites | FAIL |
61 critical tests, 19 passed, 42 failed
61 tests total, 19 passed, 42 failed
```

```
=====
Output: /share/logs/0000_ete_health/output.xml
Log: /share/logs/0000_ete_health/log.html
Report: /share/logs/0000_ete_health/report.html
```

Attachement 6: Example output of populating ONAP demo data

```
$ ./demo-k8s.sh onap init
```

```
Number of parameters:
```

```
2
```

```
KEY:
```

```
init
```

```
++ kubectl --namespace onap get pods
```

```
++ sed 's/ .*//'
```

```

++ grep robot

+ POD=dev-robot-robot-65cd75cc96-r9xqc

++ dirname ./demo-k8s.sh

+ DIR=.

+ SCRIPTDIR=scripts/demoscript

+ ETEHOME=/var/opt/ONAP

+ '[' ']'

++ kubectl --namespace onap exec dev-robot-robot-65cd75cc96-r9xqc -- bash -c 'ls -1q /share/logs/ |
wc -l'

+ export GLOBAL_BUILD_NUMBER=1

+ GLOBAL_BUILD_NUMBER=1

++ printf %04d 1

+ OUTPUT_FOLDER=0001_demo_init

+ DISPLAY_NUM=91

+ VARIABLEFILES='-V /share/config/robot_properties.py'

+ kubectl --namespace onap exec dev-robot-robot-65cd75cc96-r9xqc -- /var/opt/ONAP/runTags.sh -
V /share/config/robot_properties.py -d /share/logs/0001_demo_init -i InitDemo --display 91

Starting Xvfb on display :91 with res 1280x1024x24

Executing robot tests at log level TRACE

=====

Testsuites

=====

Testsuites.Demo :: Executes the VNF Orchestration Test cases including setu...

=====

Initialize Customer And Models                                | PASS |
-----

Initialize SO Openstack Identity For V3                        | PASS |
-----

```

Testsuites.Demo :: Executes the VNF Orchestration Test cases inclu... | PASS |

2 critical tests, 2 passed, 0 failed

2 tests total, 2 passed, 0 failed

=====

Testsuites | PASS |

2 critical tests, 2 passed, 0 failed

2 tests total, 2 passed, 0 failed

=====

Output: /share/logs/0001_demo_init/output.xml

Log: /share/logs/0001_demo_init/log.html

Report: /share/logs/0001_demo_init/report.html

Attachement 7: Dump ONAP components status

\$ kubectl -n onap get sts

NAME	READY	AGE
dev-aaf-aaf-sms-quorumclient	3/3	41m
dev-aaf-aaf-sms-vault	1/1	41m
dev-cassandra-cassandra	3/3	41m
dev-consul-consul-server	3/3	29m
dev-dmaap-dbc-pg	2/2	39m
dev-dmaap-dmaap-dr-db	2/2	39m
dev-dmaap-dmaap-dr-node	1/1	39m
dev-dmaap-message-router	1/1	39m
dev-dmaap-message-router-kafka	3/3	39m
dev-dmaap-message-router-zookeeper	3/3	39m
dev-mariadb-galera-mariadb-galera	3/3	40m
dev-multicloud-multicloud-k8s-etcd	1/1	26m
dev-multicloud-multicloud-k8s-mongo	1/1	26m
dev-sdnc-nengdb	1/1	23m
dev-sdnc-sdnc	3/3	23m
dev-vid-vid-mariadb-galera	1/1	21m

\$ kubectl -n onap get jobs.batch

NAME	COMPLETIONS	DURATION	AGE
dev-aaf-aaf-sms-preload	1/1	3m19s	41m
dev-aaf-aaf-sshsm-distcenter	1/1	40s	41m

dev-aaf-aaf-sshsm-testca	1/1	56s	41m
dev-aai-aai-graphadmin-create-db-schema	1/1	109s	30m
dev-aai-aai-traversal-update-query-data	1/1	4m3s	30m
dev-portal-portal-db-config	1/1	8m11s	25m
dev-sdc-sdc-be-config-backend	1/1	17m	24m
dev-sdc-sdc-cs-config-cassandra	1/1	2m44s	24m
dev-sdc-sdc-dcae-be-tools	1/1	22m	24m
dev-sdc-sdc-es-config-elasticsearch	1/1	2m21s	24m
dev-sdc-sdc-onboarding-be-cassandra-init	1/1	2m49s	24m
dev-sdc-sdc-wfd-be-workflow-init	1/1	2m56s	24m
dev-sdnc-sdnc-dbinit-job	1/1	25s	23m
dev-so-so-mariadb-config-job	1/1	15s	22m
dev-vid-vid-galera-config	1/1	51s	21m

\$ kubectl -n onap get pod

NAME	READY	STATUS	RESTARTS	AGE
dev-aaf-aaf-cass-85c487dfb5-9vqdn	1/1	Running	0	41m
dev-aaf-aaf-cm-796979df57-p8t8f	1/1	Running	0	41m
dev-aaf-aaf-fs-74b94d67fb-s6vrq	1/1	Running	0	41m
dev-aaf-aaf-gui-79f5584b44-82t7s	1/1	Running	0	41m
dev-aaf-aaf-locate-7d6b56ff64-7g55p	1/1	Running	0	41m
dev-aaf-aaf-oauth-78cd4f9cd9-sz9vz	1/1	Running	0	41m
dev-aaf-aaf-service-5cf9c4fdf4-lsv6w	1/1	Running	0	41m
dev-aaf-aaf-sms-7855754576-2r5pb	1/1	Running	0	41m
dev-aaf-aaf-sms-preload-b2hrn	0/1	Completed	0	41m
dev-aaf-aaf-sms-quorumclient-0	1/1	Running	0	41m
dev-aaf-aaf-sms-quorumclient-1	1/1	Running	0	41m
dev-aaf-aaf-sms-quorumclient-2	1/1	Running	0	41m
dev-aaf-aaf-sms-vault-0	2/2	Running	1	41m
dev-aaf-aaf-sshsm-distcenter-jbwqs	0/1	Completed	0	41m
dev-aaf-aaf-sshsm-testca-6qgx4	0/1	Completed	0	41m
dev-aai-aai-7744d85957-qwhwd	1/1	Running	0	30m
dev-aai-aai-babel-5cb45654f4-5vwqj	2/2	Running	0	30m
dev-aai-aai-data-router-b98c775c6-cqsbv	2/2	Running	0	30m
dev-aai-aai-elasticsearch-5cb6b5f588-tcscq	1/1	Running	0	30m
dev-aai-aai-graphadmin-5597fc59b8-s28lv	2/2	Running	0	30m
dev-aai-aai-graphadmin-create-db-schema-8fq9c	0/1	Completed	0	30m
dev-aai-aai-graphgraph-67fdb7db7f-p8sxf	0/1	ImagePullBackOff	0	30m
dev-aai-aai-modelloader-77bb578995-l5zrt	2/2	Running	0	30m
dev-aai-aai-resources-6879867dc4-dc84h	2/2	Running	0	30m
dev-aai-aai-schema-service-66499c6fd9-ltl8z	2/2	Running	0	30m
dev-aai-aai-search-data-7d78b7bffd-bf4pb	2/2	Running	0	30m
dev-aai-aai-sparkly-be-94f6b77d6-qjznm	2/2	Running	0	30m
dev-aai-aai-traversal-6d75c9c9f5-qk5ff	2/2	Running	0	30m
dev-aai-aai-traversal-update-query-data-t6d8b	0/1	Completed	0	30m
dev-cassandra-cassandra-0	1/1	Running	0	41m
dev-cassandra-cassandra-1	1/1	Running	0	37m
dev-cassandra-cassandra-2	1/1	Running	0	35m

dev-consul-consul-9bfcd7669-qxldh	1/1	Running	0	29m	
dev-consul-consul-server-0	1/1	Running	0	29m	
dev-consul-consul-server-1	1/1	Running	0	28m	
dev-consul-consul-server-2	1/1	Running	0	28m	
dev-dmaap-dbc-pg-0	1/1	Running	0	39m	
dev-dmaap-dbc-pg-1	1/1	Running	0	39m	
dev-dmaap-dbc-pgpool-857685c9bd-lfv5r	1/1	Running	0	39m	
dev-dmaap-dbc-pgpool-857685c9bd-mmzsl	1/1	Running	0	39m	
dev-dmaap-dmaap-bc-56f7b66ff9-sgpvx	1/1	Running	0	39m	
dev-dmaap-dmaap-dr-db-0	1/1	Running	0	39m	
dev-dmaap-dmaap-dr-db-1	1/1	Running	0	38m	
dev-dmaap-dmaap-dr-node-0	2/2	Running	0	39m	
dev-dmaap-dmaap-dr-prov-69746cf966-2kwb7	2/2	Running	0	39m	
dev-dmaap-message-router-0	1/1	Running	0	39m	
dev-dmaap-message-router-kafka-0	1/1	Running	0	39m	
dev-dmaap-message-router-kafka-1	1/1	Running	0	39m	
dev-dmaap-message-router-kafka-2	1/1	Running	0	39m	
dev-dmaap-message-router-mirrormaker-6cc59dd8cd-9gglb	1/1	Running	0	39m	
dev-dmaap-message-router-zookeeper-0	1/1	Running	0	39m	
dev-dmaap-message-router-zookeeper-1	1/1	Running	0	39m	
dev-dmaap-message-router-zookeeper-2	1/1	Running	0	39m	
dev-mariadb-galera-mariadb-galera-0	1/1	Running	0	40m	
dev-mariadb-galera-mariadb-galera-1	1/1	Running	0	39m	
dev-mariadb-galera-mariadb-galera-2	1/1	Running	0	38m	
dev-msb-kube2msb-86cdf8db6c-z5wkm	1/1	Running	0	27m	
dev-msb-msb-consul-fc98d9574-9d2hd	1/1	Running	0	27m	
dev-msb-msb-discovery-6bf79d47dd-gpqtq	2/2	Running	0	27m	
dev-msb-msb-eag-8596f7b584-pgmq7	2/2	Running	0	27m	
dev-msb-msb-iag-75774ff4bf-mfn6x	2/2	Running	0	27m	
dev-multicloud-multicloud-6bd7767884-ljkb6	2/2	Running	0	26m	
dev-multicloud-multicloud-azure-7bd74b7697-tpzhs	2/2	Running	0	26m	
dev-multicloud-multicloud-fcaps-596b58945-7sj8t	3/3	Running	0	26m	
dev-multicloud-multicloud-k8s-6dc84c5679-h2h2f	2/2	Running	2	26m	
dev-multicloud-multicloud-k8s-etcd-0	1/1	Running	0	26m	
dev-multicloud-multicloud-k8s-mongo-0	1/1	Running	0	26m	
dev-multicloud-multicloud-lenovo-6dfb58c96c-qj228	2/2	Running	0	26m	
dev-multicloud-multicloud-pike-77b578cf96-ld2tw	2/2	Running	0	26m	
dev-multicloud-multicloud-starlingx-77fb6984b-x6xbw	3/3	Running	0	26m	
dev-multicloud-multicloud-vio-68f89576bb-c6cvw	2/2	Running	0	26m	
dev-multicloud-multicloud-windriver-7f886b8787-shjqt	3/3	Running	0	26m	
dev-portal-portal-app-7778b4c6c4-7mkhp	2/2	Running	0	25m	
dev-portal-portal-cassandra-7949bff6f6-kp7ph	1/1	Running	0	25m	
dev-portal-portal-db-5bb5dbf8dc-n2nzh	1/1	Running	0	25m	
dev-portal-portal-db-config-lqn9h	0/2	Completed	0	25m	
dev-portal-portal-sdk-7fb88c57c-pbz88	2/2	Running	0	25m	
dev-portal-portal-widget-7c944bf9b-2mwtw	1/1	Running	0	25m	
dev-portal-portal-zookeeper-bf44d644f-d76jh	1/1	Running	0	25m	
dev-robot-robot-65cd75cc96-r9xqc	1/1	Running	0	24m	

dev-sdc-sdc-be-855ff6b44-jz5wh	2/2	Running	0	24m
dev-sdc-sdc-be-config-backend-9b5gd	0/1	Completed	0	24m
dev-sdc-sdc-cs-config-cassandra-nvhj7	0/1	Completed	0	24m
dev-sdc-sdc-dcae-be-5744545b65-wb5lw	2/2	Running	0	24m
dev-sdc-sdc-dcae-be-tools-pjrlc	0/1	Completed	0	3m52s
dev-sdc-sdc-dcae-dt-689997dfd6-bg62s	2/2	Running	0	24m
dev-sdc-sdc-dcae-fe-5488765bbf-tb6kk	2/2	Running	0	24m
dev-sdc-sdc-dcae-tosca-lab-7b4b46dc67-wcj22	2/2	Running	0	24m
dev-sdc-sdc-es-5c9788797-2wtw8	1/1	Running	0	24m
dev-sdc-sdc-es-config-elasticsearch-ggpvv	0/1	Completed	0	24m
dev-sdc-sdc-fe-68964747d5-zwxsn	2/2	Running	0	24m
dev-sdc-sdc-kb-655f754957-cwkg8	1/1	Running	0	24m
dev-sdc-sdc-onboarding-be-7656998885-kgbgs	2/2	Running	0	24m
dev-sdc-sdc-onboarding-be-cassandra-init-jrg9h	0/1	Completed	0	24m
dev-sdc-sdc-wfd-be-5884945bbb-q6wxh	1/1	Running	0	24m
dev-sdc-sdc-wfd-be-workflow-init-bvz4g	0/1	Completed	0	24m
dev-sdc-sdc-wfd-fe-6b85c4d87f-5npqq	2/2	Running	0	24m
dev-sdnc-nengdb-0	1/1	Running	0	23m
dev-sdnc-network-name-gen-7cf8b96dc4-ckmvw	1/1	Running	0	23m
dev-sdnc-sdnc-0	2/2	Running	0	23m
dev-sdnc-sdnc-1	2/2	Running	0	23m
dev-sdnc-sdnc-2	2/2	Running	0	23m
dev-sdnc-sdnc-ansible-server-587d8ffdfb-zk6m6	1/1	Running	0	23m
dev-sdnc-sdnc-dbinit-job-6bft2	0/1	Completed	0	23m
dev-sdnc-sdnc-dgbuilder-846dc6856-76cvb	1/1	Running	0	23m
dev-sdnc-sdnc-dmaap-listener-6bbc5bbd84-lbs66	1/1	Running	0	23m
dev-sdnc-sdnc-ueb-listener-6bd57b7f8-hqnnd	1/1	Running	3	23m
dev-so-so-58b5bd7f6-dctxd	1/1	Running	0	22m
dev-so-so-bpmn-infra-58c79778f6-g85bw	1/1	Running	0	22m
dev-so-so-catalog-db-adapter-84d9d75df-ms82j	1/1	Running	0	22m
dev-so-so-mariadb-config-job-2vrt9	0/1	Completed	0	22m
dev-so-so-monitoring-549567c8fd-m7jrjz	1/1	Running	0	22m
dev-so-so-openstack-adapter-6b9f76cf45-gvrf8	1/1	Running	0	22m
dev-so-so-request-db-adapter-7898f95f4f-22ksx	1/1	Running	0	22m
dev-so-so-sdc-controller-5b69c5fbdf-ffwln	1/1	Running	0	22m
dev-so-so-sdnc-adapter-76464f9cf4-96z6s	1/1	Running	0	22m
dev-so-so-vfc-adapter-cf9854c5b-26xq9	1/1	Running	0	22m
dev-so-so-vnfm-adapter-6ff57c9b9-q94fb	1/1	Running	0	22m
dev-vid-vid-757cb484f5-bxp68	2/2	Running	0	21m
dev-vid-vid-galera-config-vrjsx	0/1	Completed	0	21m
dev-vid-vid-mariadb-galera-0	1/1	Running	0	21m

\$ kubectl -n onap get pvc

NAME	STATUS	VOLUME	CAPACITY
ACCESS MODES STORAGECLASS AGE			
cassandra-data-dev-cassandra-cassandra-0		Bound	pvc-14f03b91-57fc-41d1-952f-ac1e56612fe7 2Gi RWO
	general	41m	

cassandra-data-dev-cassandra-cassandra-1					Bound	pvc-384b7492-1fbe-
4c3b-a7d7-8742e0b84abc	2Gi	RWO	general	38m		
cassandra-data-dev-cassandra-cassandra-2					Bound	pvc-4784401a-ca30-
4ca7-9c16-12efdc885383	2Gi	RWO	general	35m		
dev-aaf-aaf-cass-pvc					Bound	pvc-8ff2b9a7-82d2-41be-9b87-
5c1df1e4175e	20Gi	RWO	general	41m		
dev-aaf-aaf-config-pvc					Bound	pvc-985a9d65-1b42-41ac-bc4d-
41fd1ca6feeb	2Gi	RWX	nfs	41m		
dev-aaf-aaf-hello-pvc					Bound	dev-aaf-aaf-hello-pv
40M	RWX	nfs	41m			
dev-aaf-aaf-sms					Bound	pvc-3d475666-6218-4922-84c7-
1df24538941a	1Gi	RWO	general	41m		
dev-aaf-aaf-sms-quorumclient					Bound	pvc-c027d313-f829-467c-
bd76-90ff3f7ac8d0	10Mi	RWX	nfs	41m		
dev-aaf-aaf-sms-vault					Bound	pvc-fc830bd3-92fc-433e-baae-
fc7f9a198079	2Gi	RWO	general	41m		
dev-aaf-aaf-sshsm-data					Bound	pvc-c3a4d829-7dde-4386-9535-
803a55a81a37	10Mi	RWO	general	41m		
dev-aaf-aaf-sshsm-dbus					Bound	pvc-bd829e56-ee80-421c-9792-
e65afd7a7f61	10Mi	RWO	general	41m		
dev-aaf-aaf-sshsm-distcenter					Bound	pvc-bec97275-d4af-4b19-8b56-
e70b4e5006a6	10Mi	RWX	nfs	41m		
dev-aaf-aaf-status-pvc					Bound	pvc-23a07d9f-0195-47b7-af68-
d62d55769eee	2M	RWX	nfs	41m		
dev-dmaap-dbc-pg-data-dev-dmaap-dbc-pg-0					Bound	pvc-199d62d7-6902-
45b6-9087-ae01d55e13cd	1Gi	RWO	nfs	39m		
dev-dmaap-dbc-pg-data-dev-dmaap-dbc-pg-1					Bound	pvc-637a50dc-69f1-
4c6a-b81f-0754ff8eced3	1Gi	RWO	nfs	39m		
dev-dmaap-dmaap-dr-db					Bound	pvc-bbbd729d-ee79-4ddc-9e07-
badc553464df	1Gi	RWO	general	39m		
dev-dmaap-dmaap-dr-db-data-dev-dmaap-dmaap-dr-db-0					Bound	pvc-ef35fd56-
a1ae-48d5-a5f6-b8f723557011	1Gi	RWO	general	39m		
dev-dmaap-dmaap-dr-db-data-dev-dmaap-dmaap-dr-db-1					Bound	pvc-8b25dd54-
3a7f-4216-91d4-96986d2a9246	1Gi	RWO	general	38m		
dev-dmaap-dmaap-dr-node-event-logs-pvc-dev-dmaap-dmaap-dr-node-0					Bound	pvc-
3f108efa-37c6-4d17-ab49-991651528aab	1Gi	RWO	general	39m		
dev-dmaap-dmaap-dr-node-spool-data-pvc-dev-dmaap-dmaap-dr-node-0					Bound	pvc-
da3b7074-e8b8-49cf-b935-8e2fad7d7d4b	1Gi	RWO	general	39m		
dev-mariadb-galera-mariadb-galera					Bound	pvc-6c930af8-26e8-434c-
8569-9cbe6541b45b	2Gi	RWO	general	40m		
dev-mariadb-galera-mariadb-galera-data-dev-mariadb-galera-mariadb-galera-0					Bound	pvc-
b6381754-d614-4a0c-a5c5-91927632b1f6	2Gi	RWO	general	40m		
dev-mariadb-galera-mariadb-galera-data-dev-mariadb-galera-mariadb-galera-1					Bound	pvc-
f6930394-d56a-44a3-9f04-34ac7821b068	2Gi	RWO	general	39m		
dev-mariadb-galera-mariadb-galera-data-dev-mariadb-galera-mariadb-galera-2					Bound	pvc-
c0022aed-e374-4ec8-a13d-ce487a6d8a96	2Gi	RWO	general	38m		
dev-multicloud-multicloud-k8s-etcd-data-dev-multicloud-multicloud-k8s-etcd-0					Bound	pvc-
da164d94-1f38-4610-84b5-6e504aad6bc9	1Gi	RWO	general	26m		

dev-multicloud-multicloud-k8s-mongo-data					Bound	pvc-5650aea3-643a-4dc5-9c32-77b319154997	1Gi	RWO	general	26m
dev-multicloud-multicloud-windriver					Bound	pvc-2fc129d6-129f-4b4e-8bb1-6f38c53fa425	5Gi	RWO	general	26m
dev-portal-portal-cassandra					Bound	pvc-3544fb81-7ad4-4441-b940-2bf388cba448	2Gi	RWO	general	25m
dev-portal-portal-db					Bound	pvc-5634d597-dd46-48d6-9e65-c26b8b975eaa	2Gi	RWO	general	25m
dev-robot-robot					Bound	pvc-a7eff314-abd6-4caf-b760-782486cae333	2Gi	RWX	nfs	25m
dev-sdc-sdc-es					Bound	pvc-5a7554bd-7ce2-4f41-b99c-e2cb42c55ccd	2Gi	RWO	general	24m
dev-sdc-sdc-onboarding-be-cert					Bound	pvc-9400daf5-5d19-4216-9ca9-73be724bcfac	10Mi	ROX	general	24m
dev-sdnc-nengdb					Bound	pvc-c28b2ff2-8b30-4fd2-85b9-9603bfa56fb4	2Gi	RWO	general	23m
dev-sdnc-nengdb-data-dev-sdnc-nengdb-0					Bound	pvc-5593360e-a862-4ace-ab18-c5ebfbe7d8c6	2Gi	RWO	general	23m
dev-sdnc-sdnc-mdsal-dev-sdnc-sdnc-0					Bound	pvc-7e04d9ad-a47d-4772-80bc-795af24ad88c	1Gi	RWO	nfs	23m
dev-sdnc-sdnc-mdsal-dev-sdnc-sdnc-1					Bound	pvc-121b2cb7-7b04-4aee-a57a-58aeed359f3	1Gi	RWO	nfs	23m
dev-sdnc-sdnc-mdsal-dev-sdnc-sdnc-2					Bound	pvc-cd35b783-0ec5-4a3b-9504-7abe0fea74a1	1Gi	RWO	nfs	23m
dev-sdnc-sdnc-pvc-certs					Bound	pvc-c26ca8d3-eba4-40a7-b025-31ede7a95751	50Mi	RWX	nfs	23m
kafka-data-dev-dmaap-message-router-kafka-0					Bound	pvc-5c8a5d7b-fceb-47eb-ad06-087c2d55b351	2Gi	RWO	nfs	39m
kafka-data-dev-dmaap-message-router-kafka-1					Bound	pvc-13d14c61-e000-4e4d-a5ef-b5a5e674e2f2	2Gi	RWO	nfs	39m
kafka-data-dev-dmaap-message-router-kafka-2					Bound	pvc-04b36e03-de9e-4b5c-a255-b5fdd3d4f638	2Gi	RWO	nfs	39m
zookeeper-data-dev-dmaap-message-router-zookeeper-0					Bound	pvc-e81f3e06-3d49-49bb-a45f-5cf34a5c8253	2Gi	RWO	general	39m
zookeeper-data-dev-dmaap-message-router-zookeeper-1					Bound	pvc-dc529ac4-9625-4c65-8abe-979c6020b504	2Gi	RWO	general	39m
zookeeper-data-dev-dmaap-message-router-zookeeper-2					Bound	pvc-dd482df9-8087-4d57-af8f-6607d6dcc86f	2Gi	RWO	general	39m

Debug Tips

Tip 1: Tear down ONAP instance:

```
helm list |grep dev
helm del --purge dev-aaf dev-aai dev-cassandra dev-consul dev-dmaap dev-msb dev-multicloud
dev-portal dev-robot dev-sdc dev-sdnc dev-so dev-vid dev-mariadb-galera dev
```

```
kubectl -n onap get jobs.batch | cut -d ' ' -f1 | xargs kubectl delete jobs.batch
```

```
kubectl -n onap get pvc | cut -d ' ' -f1 | xargs kubectl delete pvc
```

Tip 2: Postman collections

Postman imports following collection and environment to postman:

<https://github.com/biny993/oom/tree/elalto-wrcp19.12/postman>

Postman Requests	Intention	Involved Environment Variables
0,Get Example Cloud-Regions	Validate the postman connection to ONAP AAI service	ONAP_SERVICE_IP must be set with K8S IP for accessing ONAP services
1,declare Owning-Entity in AAI	Declare Owning entity	
2,Declare platform	Declare platform	
3,Declare lineOfBusiness	Declare line of business	
4,Declare project	Declare project	
5, Create a Complex	Create a Complex to AAI	CLOUD_COMPLEX must be set appropriately
6, Create a Cloud-Region	Create a Cloud Region to AAI	CLOUD_OWNER, CLOUD_REGIONID, K8S_APISERVER, K8S_APITOKEN, OS_REGION_ID, OS_AUTH_URL, OS_USERNAME, OS_PASSWORDOS_PROJECT_NAME must be set appropriately
7, Register To Multicloud	Trigger MultiCloud Registration process for a Cloud Region	
8, Get a k8s connectivity-info	Check if k8s endpoint is registered to multicloud-k8s service	
9, Create a Customer	Create a subscription customer	global-customer-id must be set appropriately
10, Create service type	Create service type	global-service-type must be set appropriately
11, Put a Customer subscription	Create customer subscription	
12, Put a Customer subscription relationship	Associate customer with subscription	

13, preload for VFmodule	Preload VNF topology to SDNC	gvnf_instance_name, gvnf_model_name, service_instance_uuid, vfmodule_instance_name, vfmodule_model_name must be set appropriately
--------------------------	------------------------------	--