

How-To: Register a VIM/Cloud Instance to ONAP

See : https://onap.readthedocs.io/en/latest/guides/onap-user/cloud_site/index.html

Following is maybe obsolete...

When a VIM/Cloud instance is used to deploy an ONAP instance, it will be registered to that ONAP instance by default. However, it is somehow tricky to add additional VIM/Cloud instance to ONAP instance with Amsterdam Release. This page is trying to share the practice to do so and the goal is to allow users deploy VNF/VF-Modules to the additional VIM/Cloud instance.

There are 3 places to store information of a VIM/Cloud instance: A&AI, SO, Robot. And it is very important to associate the VIM/Cloud instance with subscriber's service subscription, otherwise you will not able to deploy the subscriber's VNF/VF-Modules to that VIM/Cloud instance .

Prerequisite: Complex object and Subscribe object are created in the AAI

Complex object in A&AI represent the physical location of a VIM/Cloud instance, there are one-to-many association between them, so creating the complex object is out of scope of this practice. But I would like to share the POSTMAN script to do create complex named cli2 for your reference:

Create a Complex Object

```
PUT /aai/v11/cloud-infrastructure/complexes/complex/cli2 HTTP/1.1
Host: <AAI_VMI_IP>:8443
X-TransactionId: 9999
X-FromAppId: jimmy-postman
Real-Time: true
Authorization: Basic QUFJOkFBSQ==
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
Postman-Token: 734b5a2e-2a89-1cd3-596d-d69904bcda0a

{
  "physical-location-id": "cli2",
  "data-center-code": "example-data-center-code-val-6667",
  "complex-name": "cli2",
  "identity-url": "example-identity-url-val-28399",
  "physical-location-type": "example-physical-location-type-val-28399",
  "street1": "example-street1-val-28399",
  "street2": "example-street2-val-28399",
  "city": "example-city-val-28399",
  "state": "example-state-val-28399",
  "postal-code": "example-postal-code-val-28399",
  "country": "example-country-val-28399",
  "region": "example-region-val-28399",
  "latitude": "1111",
  "longitude": "2222",
  "elevation": "example-elevation-val-28399",
  "lata": "example-lata-val-28399"
}
```

Subscriber registration and subscription to service is out of scope of this practice, a default subscriber "Demonstration" and service subscriptions are populated by Robot service.

Here are the commands to find out subscribers and their subscriptions.

List subscribers

```
curl -X GET \
  https://<AAI_VMI_IP>:8443/aai/v11/business/customers \
  -H 'accept: application/json' \
  -H 'authorization: Basic QUFJOkFBSQ==' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: eb104518-1ce2-5fc8-a475-97890dbc36e0' \
  -H 'real-time: true' \
  -H 'x-fromappid: jimmy-postman' \
  -H 'x-transactionid: 9999'
```

List subscriptions of a subscriber

```
curl -X GET \
  'https://<AAI_VMI_IP>:8443/aai/v11/business/customers/customer/Demonstration?depth=all' \
  -H 'accept: application/json' \
  -H 'authorization: Basic QUFJOkFBSQ==' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: 87a35cc3-5411-1f0c-3caa-21c11f83b79a' \
  -H 'real-time: true' \
  -H 'x-fromappid: jimmy-postman' \
  -H 'x-transactionid: 9999'
```

1, Register VIM/Cloud instance into A&AI

The VIM/Cloud instance are represented as a cloud region object in A&AI. This representation will be used by VID,APPC, VFC, DCAEgen2, MultiCloud,etc.

Assuming you have a OpenStack ocata instance with Region ID is RegionTwo, Registering it into A&AI basically refers to following actions:

- Action 1: create a cloud region object

There are at least 2 approaches to create a cloud region object: Post curl commands to A&AI directly, or utilize ESR VIM registration portal.

Approach 1: curl commands to A&AI

e.g. Create a OpenStack Ocata instance as a cloud region identified by "CloudOwner/RegionTwo" into AAI,

Create a Cloud Region Object

```
curl -X PUT \
  'https://<AAI_VM1_IP>:8443/aai/v11/cloud-infrastructure/cloud-regions/cloud-region/CloudOwner/RegionTwo' \
  -H 'accept: application/json' \
  -H 'authorization: Basic QUFJOKFBSQ==' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: f7c57ec5-ac01-7672-2014-d8dfad883cea' \
  -H 'real-time: true' \
  -H 'x-fromappid: jimmy-postman' \
  -H 'x-transactionid: 9999' \
  -d '{
    "cloud-owner": "CloudOwner",
    "cloud-region-id": "RegionTwo",
    "cloud-type": "openstack",
    "owner-defined-type": "t1",
    "cloud-region-version": "ocata",
    "cloud-zone": "z1",
    "complex-name": "c1li2",
    "identity-url": "<keystone auth url, e.g. http://10.12.25.2:5000/v3> or <multicloud plugin service auth
url, e.g. http://10.0.14.1/api/multicloud-ocata/v0/CloudOwner_RegionTwo/identity/v2.0>"
    "sriov-automation": false,
    "cloud-extra-info": "",
    "tenants": {
      "tenant": [
        {
          "tenant-id": "48271fcc571449959d7d881a72f4ba10",
          "tenant-name": "onap-01"
        }
      ]
    },
    "esr-system-info-list":
    {
      "esr-system-info":
      [
        {
          "esr-system-info-id": "<uuid, e.g. 432ac032-e996-41f2-84ed-9c7a1766eb29>",
          "service-url": "keystone auth url, e.g. http://10.12.25.2:5000/v3",
          "user-name": "<your username>",
          "password": "<your password>",
          "system-type": "VIM",
          "ssl-cacert": "",
          "ssl-insecure": true,
          "cloud-domain": "<openstack domain name,e.g. Default>",
          "default-tenant": "onap-01"
        }
      ]
    }
  },'
```

Approach 2: utilize ESR VIM registration portal

open the portal by url: <http://<multi-service VM IP>/ui/aai-esr-gui/extsys/vim/vimView.html>

Click the "register button", the registration portal will show up.

Cloud Owner *	<input type="text" value="CloudOwner"/>
Cloud Region Id *	<input type="text" value="RegionTwo"/>
Cloud Type *	<input type="text" value="openstack"/>
Cloud Region Version *	<input type="text" value="ocata"/>
Owner Defined Type *	<input type="text" value="t1"/>
Cloud Zone *	<input type="text" value="z1"/>
Complex Name *	<input type="text" value="cli2"/>
username *	<input type="text" value="demo"/>
password	<input type="password" value="••••••••"/>
Auth Url	<input type="text" value="http://10.12.25.2:5000/v3"/>
ssl Cacert	<input type="text"/>
ssl Insecure	<input type="text" value="true"/>
Cloud Domain	<input type="text" value="Default"/>
Default Tenant	<input type="text" value="onap-01"/>
Cloud Extra Info	<input type="text"/>

Caveat: approach 2 which utilizes ESR VIM register portal will trigger MultiCloud to discover resources of VIM/Cloud instance and populate them into AAI. Right now with Amsterdam Release, it supports OpenStack ocata, Wind River Titanium Cloud, and VMware VIO. More options will be available in future releases.

- Action 2: associate this cloud region object with a complex object

This association will be important for ONAP Beijing Release when OOF needs to find out the location proximity of different cloud regions.

Associate cloud region with complex "cli2"

```
curl -X PUT \
  https://<AAI_VMI_IP>:8443/aai/v11/cloud-infrastructure/cloud-regions/cloud-region/CloudOwner/RegionTwo
/relationship-list/relationship \
-H 'accept: application/json' \
-H 'authorization: Basic QUFJOkFBSQ==' \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-H 'postman-token: e68fd260-5cac-0570-9b48-c69c512b034f' \
-H 'real-time: true' \
-H 'x-fromappid: jimmy-postman' \
-H 'x-transactionid: 9999' \
-d '{
{
  "related-to": "complex",
  "related-link": "/aai/v11/cloud-infrastructure/complexes/complex/cli2",
  "relationship-data": [
  {
    "relationship-key": "complex.physical-location-id",
    "relationship-value": "cli2"
  }
]
}'
```

- Action 3: associate this cloud region object with a subscriber's service subscription

With this association, this cloud region will be populated into list of selection for deploying VNF/VF-Modules from VID.

e.g. associate the cloud region "CloudOwner/RegionTwo" with subscriber "Demonstration" which subscribe to service "vFWCL"

Associate cloud region with a subscriber's subscription

```
curl -X PUT \
  https://<AAI_VMI_IP>:8443/aai/v11/business/customers/customer/Demonstration/service-subscriptions/service-
subscription/vFWCL/relationship-list/relationship \
  -H 'accept: application/json' \
  -H 'authorization: Basic QUFJOkFBSQ==' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: 4e6e55b9-4d84-6429-67c4-8c96144dlc52' \
  -H 'real-time: true' \
  -H 'x-fromappid: jimmy-postman' \
  -H 'x-transactionid: 9999' \
  -d '{
    "related-to": "tenant",
    "related-link": "/aai/v11/cloud-infrastructure/cloud-regions/cloud-region/CloudOwner/RegionTwo/tenants
/tenant/48271fcc571449959d7d881a72f4ba10",
    "relationship-data": [
      {
        "relationship-key": "cloud-region.cloud-owner",
        "relationship-value": "CloudOwner"
      },
      {
        "relationship-key": "cloud-region.cloud-region-id",
        "relationship-value": "RegionTwo"
      },
      {
        "relationship-key": "tenant.tenant-id",
        "relationship-value": "48271fcc571449959d7d881a72f4ba10"
      }
    ],
    "related-to-property": [
      {
        "property-key": "tenant.tenant-name",
        "property-value": "onap-01"
      }
    ]
  }
'
```

2, Register VIM/Cloud instance into SO

SO does not utilize the cloud region representation in A&AI. It stores information of the VIM/Cloud instances inside of the container as configuration file. So to add a VIM/Cloud instance to SO, you need attach to the SO service container and then update the configuration file "/etc/mso/config.d/cloud_config.json" accordingly.

Below is the template for adding a new cloud site and the associate identity service.

Add the VIM/Cloud info by updating cloud_sites

```
ubuntu@vm0-so:~$ sudo docker exec -it testlab_mso_1 bash
root@mso:/# apt-get install vim
root@mso:/# vi /etc/mso/config.d/cloud_config.json

{
  "cloud_config":
  {
    "identity_services":
    {
      "DEFAULT_KEYSTONE":
      {
        "identity_url": "http://10.12.25.2:5000/v2.0",
        "mso_id": "demonstrable",
        "mso_pass": "9f0933d3664e07961856781c9add38bc",
        "admin_tenant": "service",
        "member_role": "admin",
        "tenant_metadata": true,
        "identity_server_type": "KEYSTONE",
        "identity_authentication_type": "USERNAME_PASSWORD"
      },
      "DEFAULT_KEYSTONE2":
      {
        "identity_url": "<keystone auth url, support only keystone API v2.0>",
        "mso_id": "<username>",
        "mso_pass": "<encrypted password by http://<MSO VM IP>:8080/networks/rest/cloud/encryptPassword
/<password> >",
        "admin_tenant": "service",
        "member_role": "admin",
        "tenant_metadata": true,
        "identity_server_type": "KEYSTONE",
        "identity_authentication_type": "USERNAME_PASSWORD"
      }
    },
    "cloud_sites":
    {
      "RegionOne":
      {
        "region_id": "RegionOne",
        "clli": "RegionOne",
        "aic_version": "2.5",
        "identity_service_id": "DEFAULT_KEYSTONE"
      },
      "<cloud_site name, must align to cloud-region-id in AAI>":
      {
        "region_id": "cloud_site name, must align to cloud-region-id in AAI",
        "clli": "cloud_site name, must align to cloud-region-id in AAI",
        "aic_version": "2.5",
        "identity_service_id": "DEFAULT_KEYSTONE2"
      }
    }
  }
}
```

check the new config:

<http://<so-vm-ip>:8080/networks/rest/cloud/showConfig>

3, Change Robot service to operate with the VIM/Cloud instance

By default the Robot service supports the pre-populated cloud region of which the cloud-owner is "CloudOwner" and cloud-region-id is specified via the parameters "openstack_region" for deploying this ONAP instance. All cloud region information can be found at "/share/config/vm_properties.py" .

Robot service supports the hacking operations to populate subscribers, service models, and preload VNF data, bridge heat instance into AAI, as well as APPC mount. Most of operation are tightly coupled with the default cloud region. So if you would like to perform such hacking operations with additional cloud region, you'd better to launch another robot VM with corresponding cloud region parameters in order to get expected results.

Please refer to the template [additional_robot.zip](#) to create additional robot for new cloud region

Hint: Take care of following parameters which should reflect the new cloud region information, while others reflect the cloud region where ONAP runs

parameters reflecting info of new cloud region

```
openstack_tenant_id: PUT YOUR OPENSTACK PROJECT ID HERE

openstack_username: PUT YOUR OPENSTACK USERNAME HERE

openstack_api_key: PUT YOUR OPENSTACK PASSWORD HERE

openstack_region: PUT YOUR OPENSTACK Region ID HERE, e.g. RegionOne

keystone_url: PUT THE KEYSTONE URL HERE (do not include version number)

cloud_env: openstack
```

Launch robot VM by heat template:

Launch additional robot VM for new cloud region

```
openstack stack create -t additional_robot_vm.yaml -e additional_robot_vm.env onap.robot.1
```

To verify if the new robot service works well, you could check the ONAP service health from the new robot VM.

Verify the health of new Robot VM

*** ssh to Robot VM and perform following command, it is expected the same output as you can get from the original Robot VM***

```
sudo docker exec -it openecompete_container /var/opt/OpenECOMP_ETE/runTags.sh -i health h -d ./html -V /share
/config/integration_robot_properties.py -V /share/config/integration_preload_parameters.py -V /share/config
/vm_properties.py
```