F5 Private Cloud Solutions Documentation

F5 Networks, Inc.

Aug 28, 2019

Agility 2019 Hands-on Lab Guide

Contents:

1

Class 1: F5 Private Cloud Solutions for Openstack

1.1 Getting Started

During this lab you will learn how to:

- Install the F5 LBaaS Agent using Ansible
- Deploy Basic L4-L7 services using LBaaS
- Deploy enhanced L4-L7 services using ESD.

1.1.1 About OpenStack

OpenStack provides an Open Source Infrastructure As-A Service (IaaS) solution that provides a framework for provisioning Network, Compute, and Storage in an automated and repeatable manner.

1.1.2 About LBaaS

Load Balancing As-A Service (LBaaS) is a community standard around providing Load Balancing as a standardized service within OpenStack. The current version, LBaaS v2, provides basic L4-L7 capabilities.

1.1.3 About F5 & OpenStack

F5 can be deployed in two ways in an OpenStack environment. The two methods are a **undercloud** or **overcloud** deployment. It is possible to use one or both of these methodologies when deploying F5 & OpenStack.

Undercloud: LBaaS

Undercloud commonly refers to a deployment where the BIG-IP device (physical or virtual) is outside of the OpenStack environment. Typically this is done with physical hardware to provide a multi-tenant environment and used with LBaaS.

Overcloud: HEAT

Overcloud refers to a deployment where a BIG-IP Virtual Edition (VE) is provisioned within a tenant network as a virtual machine within OpenStack Nova. In this scenario the BIG-IP is in a similar topology to other tenant virtual machines. When deploying in overcloud OpenStack HEAT templates (automation templates) are commonly used to deploy the BIG-IP device. A customer can manage the BIG-IP device through traditional methods, HEAT templates, and/or other automation templates.

It is also possible to deploy a BIG-IP VE in an overcloud deployment and use LBaaS. In this deployment you are limited by the number of interfaces currently supported on BIG-IP VE can use (9 data & 1 mgmt).

1.1.4 Under or Over?

The decision to use one method or both will depend on customer requirements. An undercloud deployment using LBaaS is well suited to providing basic services that can be provided in a multi-tenant manner. Overcloud is well suited to providing access to features and functions that may not be exposed via LBaaS or provide per-tenant services.

1.2 Lab Topology

The current Lab Environment looks like the following:

You will be connecting via RDP to a Windows host to perform all the steps in this lab.

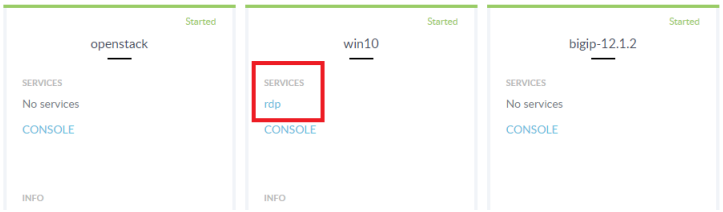
1.2.1 Lab Components

The following table lists VLANS, IP Addresses and Credentials for all components:

Component	VLAN/IP Address(es)	Credentials
Windows RDP Host	• 10.0.10.50	`student`/[Viewable in Ravello]
OpenStack	• 10.0.10.10	`student` / [SSH Key]
BIG-IP	• 10.0.10.20	`admin` / `admin`

1.2.2 Connecting to the Lab Environment

Please follow the instructions provided by the instructor to start your lab and access your jump host by clicking on this “rdp” host link.



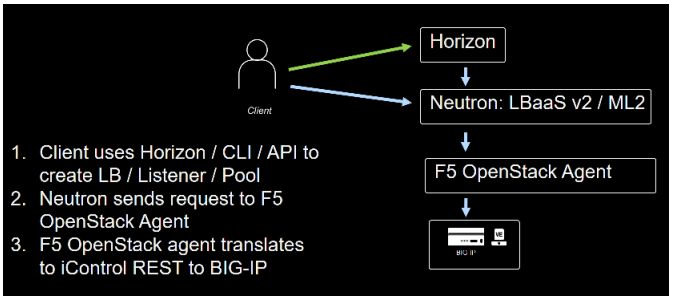
Note: All work for this lab will be performed exclusively from the Windows jumphost. No installation or interaction with your local system is required.

1.3 Installing the F5 LBaaS Agent

Two pieces of software are required to use F5 BIG-IP with OpenStack LBaaS.

1. F5 LBaaS Driver
2. F5 OpenStack Agent.

The F5 LBaaS driver communicates with F5 OpenStack Agent that will then use F5 iControl REST to update the BIG-IP configuration.



The following lab will first guide you through using both the OpenStack GUI/CLI.

You will then install the required software via an Ansible automation script.

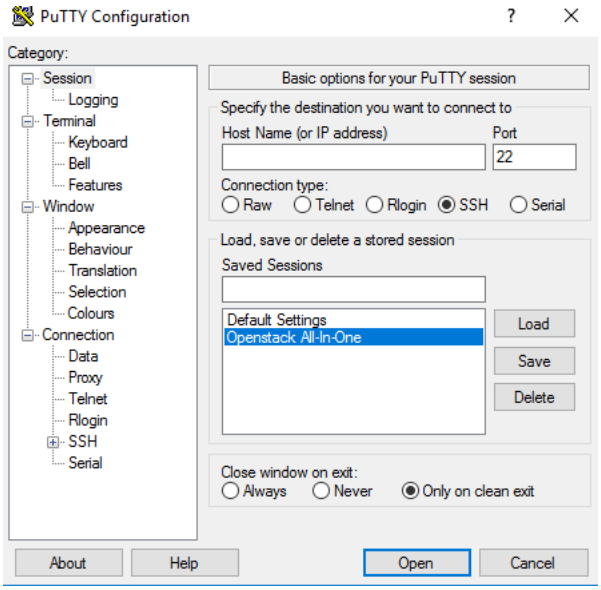
1.3.1 Login to OpenStack CLI

Verify OpenStack environment

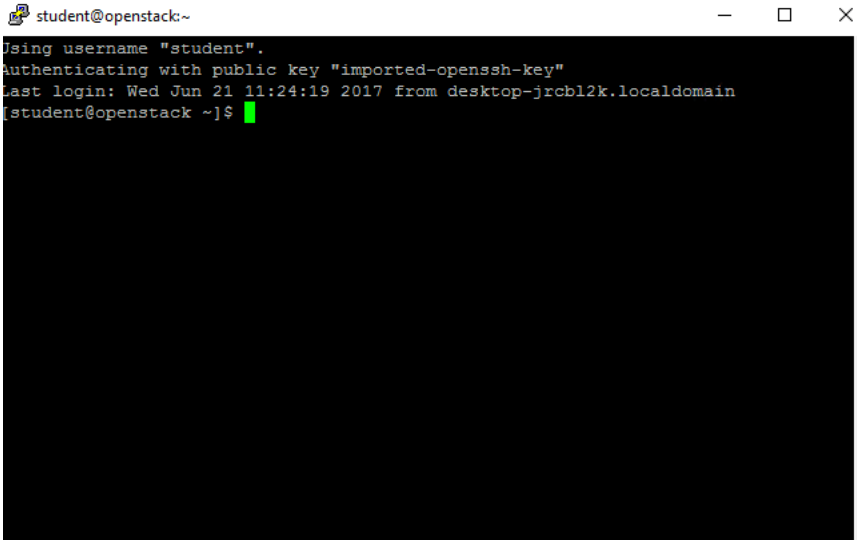
The first exercise is to use the OpenStack CLI to verify the environment.

First Launch Putty from the Desktop.

Click on “OpenStack All-In-One”, select load, then click “Open”.



If you fail to connect on the first try, try again. When you connect you should see.



Type `source keystonerc_demo`. The prompt should change from:

```
[student@openstack ~]$
```

To:

```
[student@openstack ~(keystone_demo)]$
```

Run `neutron subnet-list` and you should see

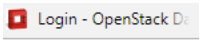
```
[student@openstack ~(keystone_demo)]$ neutron subnet-list
+----------------------+-----------------+---------------+----------------------+
| id                   | name            | cidr          | allocation_pools     |
+----------------------+-----------------+---------------+----------------------+
| 0c03d2f4-a60a-4869-  | internal-subnet | 10.1.100.0/24 | {"start":            |
| a378-e46e5cf47eac    |                 |               | "10.1.100.100",      |
|                      |                 |               | "end":               |
|                      |                 |               | "10.1.100.200"}      |
+----------------------+-----------------+---------------+----------------------+
```

Please ask for assistance if you do not see the correct output. Leave this window open, it will be used throughout the lab.

1.3.2 Deploy Backend Instances

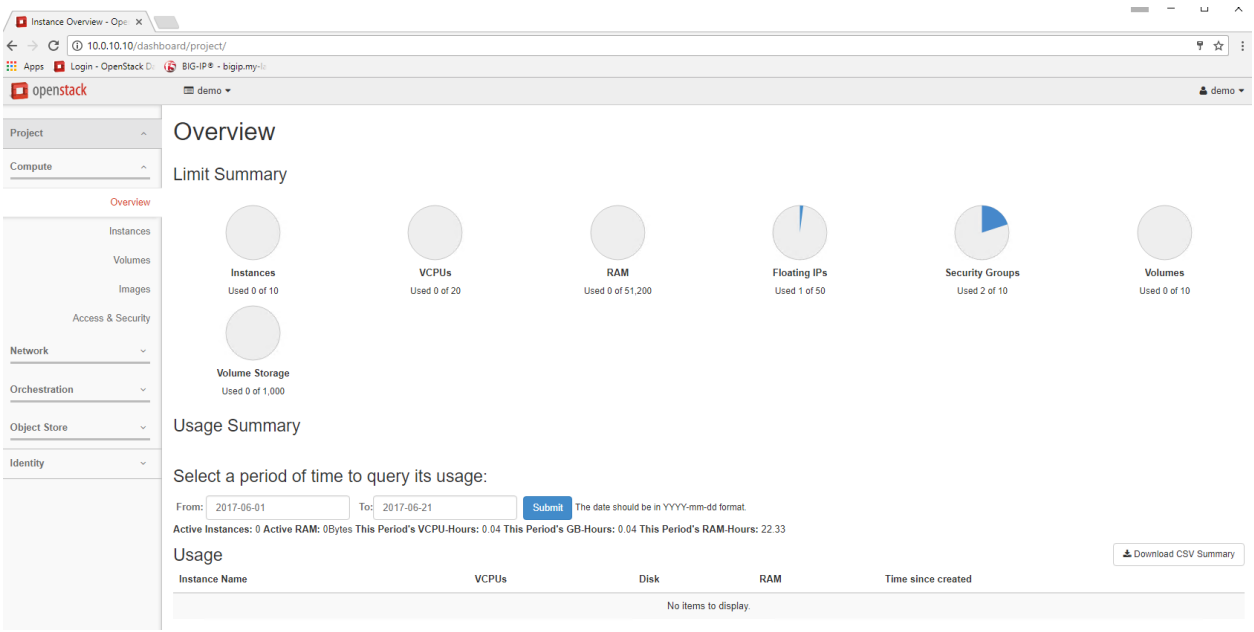
During the previous excercise we made use of the OpenStack CLI. OpenStack also has a web gui, Horizon, that can be used. The following will deploy two backend web servers that will be used later in the lab.

Launch Google Chrome and click on the "Login – OpenStack. . . " bookmark.



Login to Horizon (OpenStack GUI) using the username: demo, password: demo

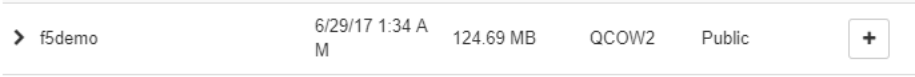
You should see.



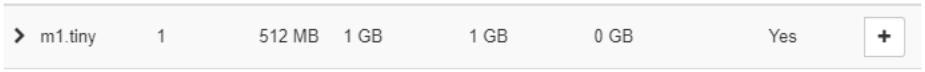
Click on “Instances” and then “Launch Instance” (top right of page).

For the Instance name specify “server” for the count enter “2”. Then click next.

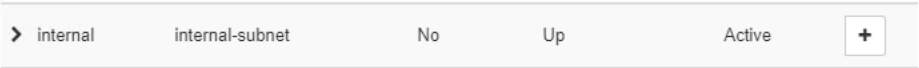
Click on the “+” next to “f5demo”.Then click next.



Click on the “+” next to “m1.tiny”. Then click next.

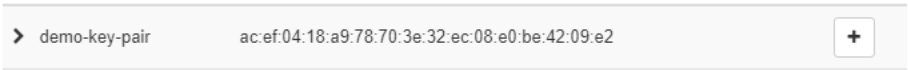


Click on the “+” next to “internal” Network. This step should have been completed for you since the internal network is the only network available. Then click on next TWICE until you are on the Security Groups tab.

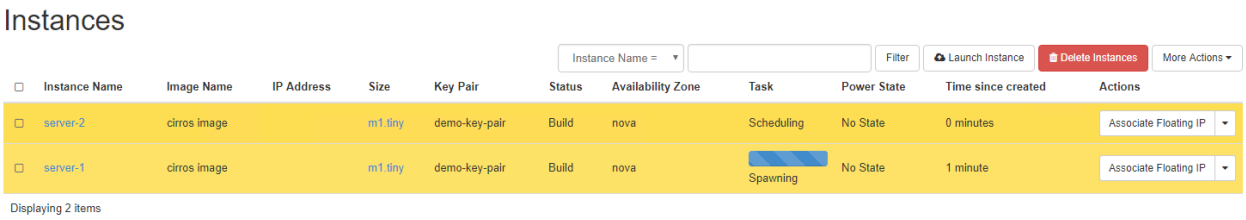


On the Security Groups tab click on the “+” next to “default-allow-all”. Then click next.

Click on the "+" next to "demo-key-pair" and the click on "Launch Instance". This step should have been completed for you since the demo-key-pair is the only available key pair.



You should now see them starting.



Once the instance status is "active" on "server-1" then "Log" you should see

1.3.3 Install Driver/Agent

Complete directions for installing the Driver/Agent can be found at: http://f5-openstack-lbaasv2-driver.readthedocs.io/en/mitaka/map_quick-start-guide.html

During this lab we will be using Ansible (a Systems/Network automation tool) to automate the installation. The Ansible module that is being used in this lab can be found at: https://github.com/f5devcentral/f5-openstack-ansible

Install via Ansible

Open your Putty Window (Directions in *Login to OpenStack CLI* if you closed the Window).

Change your directory by typing `cd f5-openstack-ansible/playbooks`

```
[student@openstack ~]$cd f5-openstack-ansible/playbooks/
```

Now run

```
ansible-playbook -i hosts --extra-vars '{"remote_user":"student"}' agent_driver_
↪deploy.yaml
```

You should see.

```
TASK [configure_lbaasv2_agent : Start F5 OpenStack Agent] *********************
changed: [10.0.10.10]

TASK [Restart neutron-metadata-agent] *****************************************
changed: [10.0.10.10]

TASK [Restart neutron-dhcp-agent] *********************************************
changed: [10.0.10.10]

TASK [Restart neutron-l3-agent] ***********************************************
changed: [10.0.10.10]

TASK [Restart neutron-openvswitch-agent] **************************************
changed: [10.0.10.10]

TASK [Restart Neutron] ********************************************************
changed: [10.0.10.10]

PLAY RECAP ********************************************************************
10.0.10.10                 : ok=23   changed=20   unreachable=0    failed=0

[student@openstack playbooks]$
```

Change back to your home directory by typing `cd`.

Now type `source keystonerc_admin` and you should see a prompt that looks like:

```
[student@openstack ~(keystone_admin)]$
```

Expand the window to full screen and type. `neutron agent-list`

```
student@openstack:~
[student@openstack ~(keystone_admin)]$ neutron agent-list
+--------------------------------------+--------------------+-----------------------------------------------+-------------------+-------+----------------+---------------------------+
| id                                   | agent_type         | host                                          | availability_zone | alive | admin_state_up | binary                    |
+--------------------------------------+--------------------+-----------------------------------------------+-------------------+-------+----------------+---------------------------+
| 20c8f652-3e0c-4d26-9f21-52e9fba62b95 | Metadata agent     | openstack.my-lab                              |                   | :-)   | True           | neutron-metadata-agent    |
| 2528f3eb-e9b6-4336-926e-5e921e71cbd9 | L3 agent           | openstack.my-lab                              | nova              | :-)   | True           | neutron-l3-agent          |
| 299fe424-dd7d-4533-ad33-eae83b49f90a | DHCP agent         | openstack.my-lab                              | nova              | :-)   | True           | neutron-dhcp-agent        |
| 5f92bcf6-bd14-474e-a5f2-07864623e421 | Loadbalancerv2 agent | openstack.my-lab:3b6ed4b9-082f-58e7-9023-0d98339b8962 |                   | :-)   | True           | f5-oslbaasv2-agent        |
| 6401a932-d0d2-4e64-a7de-0bf91c87e397 | Open vSwitch agent | openstack.my-lab                              |                   | :-)   | True           | neutron-openvswitch-agent |
| ff731693-b615-40aa-a937-96abaef3e913 | Metering agent     | openstack.my-lab                              |                   | :-)   | True           | neutron-metering-agent    |
+--------------------------------------+--------------------+-----------------------------------------------+-------------------+-------+----------------+---------------------------+
[student@openstack ~(keystone_admin)]$
```

There should be a table that contains the following information.

agent_type	alive	admin_state_up	binary
Loadbalancerv2 agent	:-)	True	f5-oslbaasv2-agent

Now type `source keystonerc_demo` to restore your prompt to the demo user.

```
[student@openstack ~(keystone_demo)]$
```

1.4 Deploying Basic L4-L7 Services using LBaaS

There's multiple ways of provisioning F5 Services via OpenStack LBaaS including.

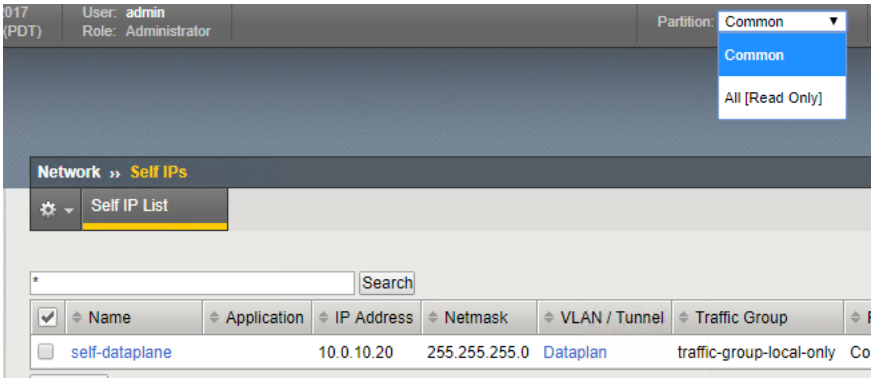
1. Horizon GUI
2. CLI
3. OpenStack API

Today we will be covering the first two options.

1.4.1 Lab 1.4: Deploy L4-L7 via Horizon

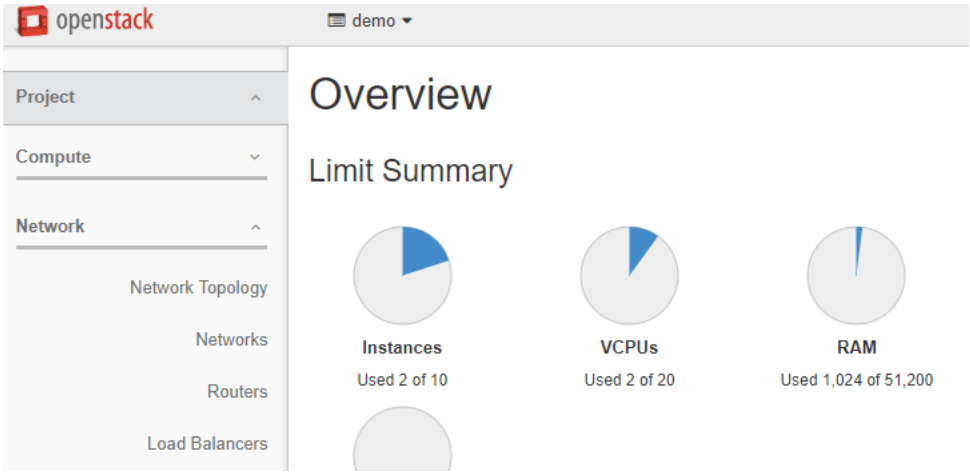
The F5 LBaaS integration will configure the Networking on the BIG-IP to connect to the OpenStack network. From Chrome click on the “BIG-IP” bookmark and login with the credentials “admin / admin”. Observe that there is only a single partition “Common”.

Also note only one self-ip in Route Domain 0.

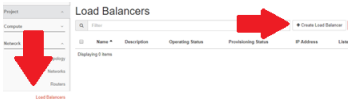


Switch back to the OpenStack Horizon tab inside Chrome and do a forced refresh (Shift+[Reload]).

You should now see a new menu item under “Network”.



If you do not see “Load Balancers” verify that the Loadbalancer Agent is running from the previous lab. Click on the “Load Balancers” menu item, then click on “+Create Load Balancer”.



Complete the following information.

Important: Make sure to use the values below and not the GUI defaults!

Load Balancer Details

name	value
Name	lb1
Subnet	internal-subnet

Listener Details

name	value
Name	listener1
Protocol	HTTP
Port	80

Pool Details

name	value
Name	pool1
Method	ROUND_ROBIN

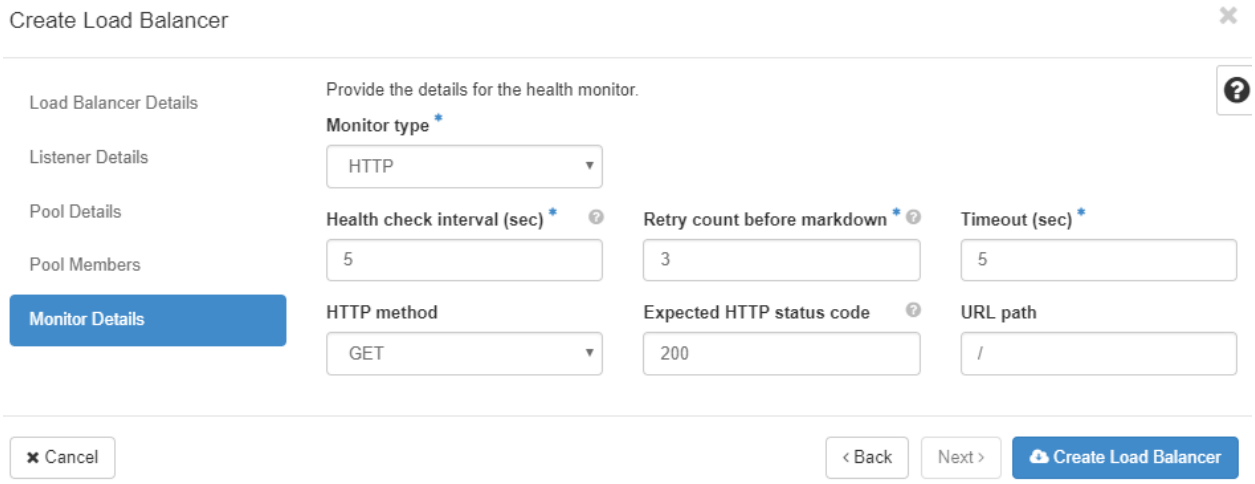
Pool Members

name	port
server-1	80
server-2	80

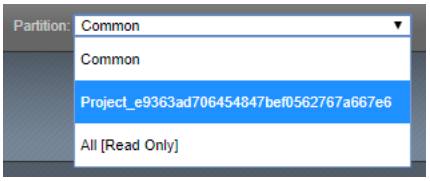
Monitor type

name	value
Monitor type	HTTP

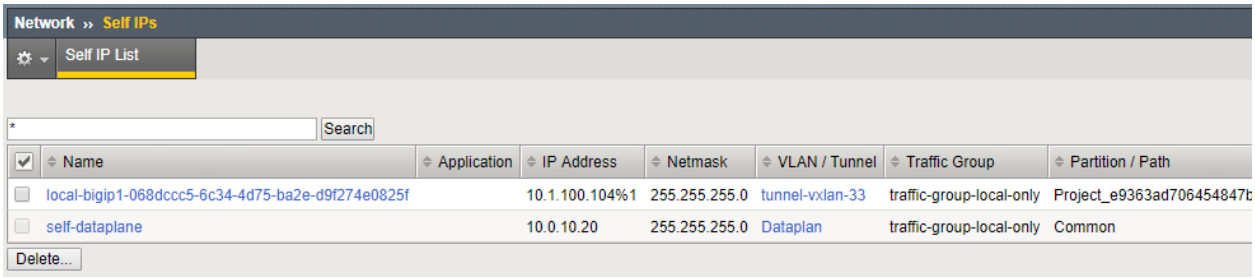
Then click on "Create Load Balancer"



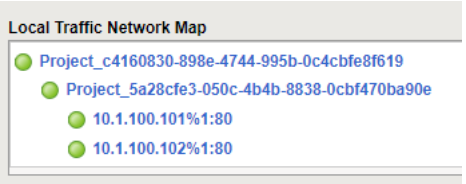
On the BIG-IP take a look at the Partition. You should see that a new partition was created.



Change to that partition and inspect the Self IPs items under Network. You should see that a VXLAN tunnel that was created connected to the tenant network. Verify the tenant network is the internal network from viewing the `neutron subnet-list` command you ran in the previous lab.



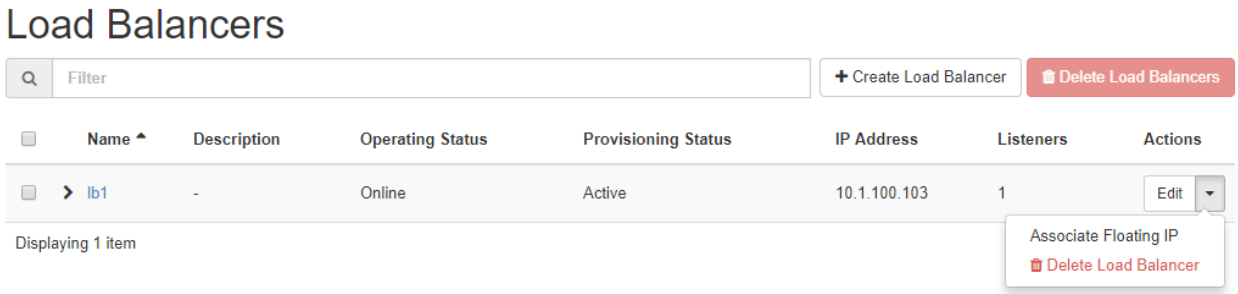
Under Network Map you will see the entries that were created by LBaaS via the Horizon Panel.



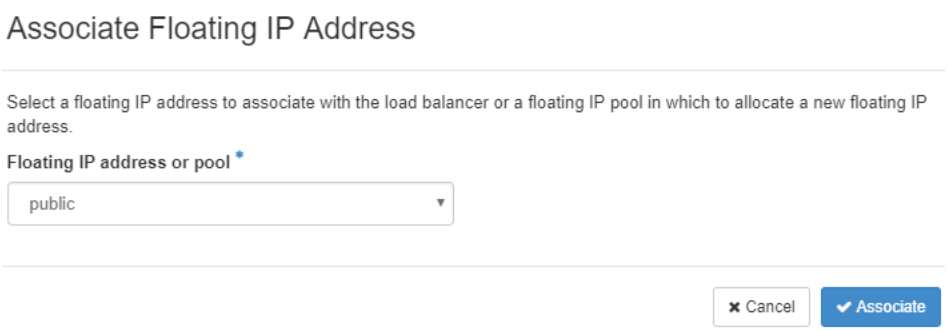
Observe that the BIG-IP Pool name uses the OpenStack Pool ID from the load balancer configuration. Horizon>Network>Load Balancers>lb1>Listeners>Listener 1 – Default Pool ID

(yours will differ in value from the example).

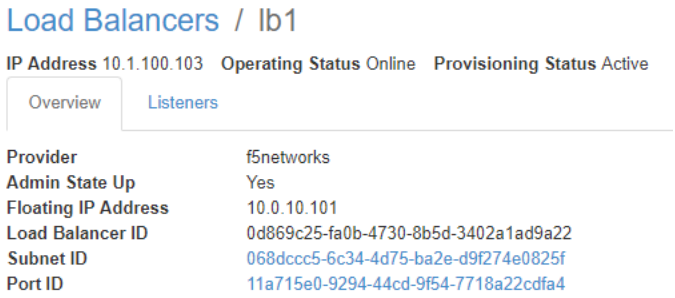
To test this configuration we will need to add a Floating IP to be able to access the Tenant Subnet externally. On the main “Load Balancers” page, click on the downward arrow next to “Edit” and select “Associate Floating IP”



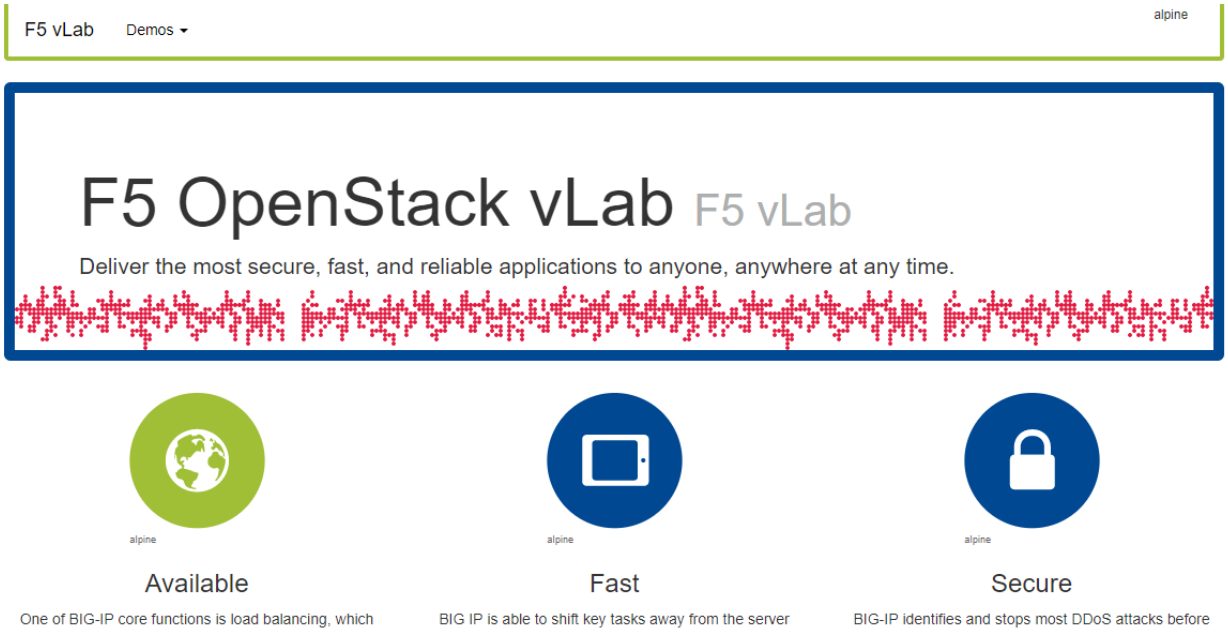
Specify the “public” pool.



And click “Associate”. Click on “lb1” and you will see the Floating IP Address.



Enter this value into the Chrome URL and you should see (colors may vary, there’s a chance they may be the same).



Adding “/simple.shtml” you can see the Server IP and see the service being load balanced.

1.4.2 Lab 1.5: Deploy L4-L7 via CLI

In addition to using the Horizon GUI panel you can also provision LBaaS via the command-line. From your Putty window run the following commands:

See also `Text file of commands`.

```
neutron lbaas-loadbalancer-create --name lb2 internal-subnet

neutron lbaas-listener-create --name listener2 --loadbalancer lb2 --protocol TCP --
↪protocol-port 22

neutron lbaas-pool-create --name pool2 --lb-algorithm ROUND_ROBIN --listener␣
↪listener2 --protocol TCP

neutron lbaas-member-create --subnet internal-subnet --address 10.1.100.101 --
↪protocol-port 22 pool2

neutron lbaas-member-create --subnet internal-subnet --address 10.1.100.102 --
↪protocol-port 22 pool2

neutron lbaas-healthmonitor-create --delay 3 --type TCP --max-retries 3 --timeout 3 --
↪pool pool2
```

Verify on the BIG-IP that you see the new Virtual Server deployed.

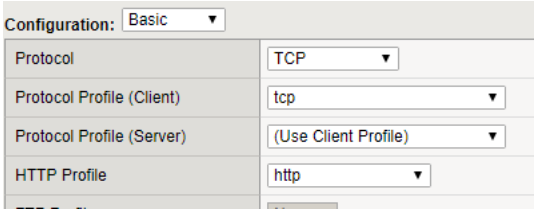
1.5 Deploying Enhanced L4-L7 Services using ESD

LBaaS only provides a subset of the capabilities of an F5 BIG-IP. The following exercise will demonstrate how to provide a way to extend LBaaS through the use of custom policies.

1.5.1 Deploy Enhanced L4-L7 via ESD

In addition to supporting LBaaS v2 capabilities, the F5 OpenStack LBaaS integration can support Enhanced Service Definitions to expose F5 specific capabilities. The following exercise will modify the TCP profiles that we created on our first listener.

First take a look at the existing TCP configuration on the BIG-IP. Observe that it is using the default TCP profile.



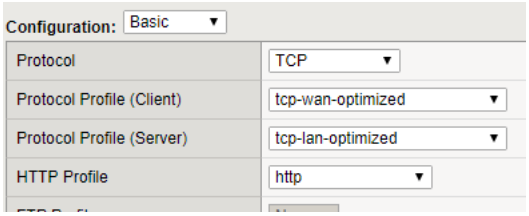
From your Putty window run.

```
neutron lbaas-l7policy-create --listener listener1 --name esd_demo_3 --action REJECT
```

You should see the following output.

```
[student@openstack ~(keystone_demo)]$ neutron lbaas-l7policy-create --listener l
istener1 --name esd_demo_3 --action REJECT
Created a new l7policy:
+------------------+--------------------------------------+
| Field            | Value                                |
+------------------+--------------------------------------+
| action           | REJECT                               |
| admin_state_up   | True                                 |
| description      |                                      |
| id               | 8336d41d-3c00-47ea-9d03-437632ea3423 |
| listener_id      | f023d320-fbb7-4a9d-a8ec-1b6ea53f1e45 |
| name             | esd_demo_3                           |
| position         | 1                                    |
| redirect_pool_id |                                      |
| redirect_url     |                                      |
| rules            |                                      |
| tenant_id        | e9363ad706454847bef0562767a667e6     |
+------------------+--------------------------------------+
```

Refresh your window on the BIG-IP and you will see that the TCP profile has changed.



Now from your Putty window run.

```
cat /etc/neutron/services/f5/esd/demo.json
```

You will see the definition that we referenced earlier.

```
  "esd_demo_3": {
    "lbaas_ctcp": "tcp-wan-optimized",
    "lbaas_stcp": "tcp-lan-optimized"
  }
}
[student@openstack ~(keystone_demo)]$
```

In addition to TCP profiles you can also add iRules, Local Traffic Policies, client/server SSL profiles, and modify session persistence.

2 Class 2: Deploying Cisco APIC with F5 iWorkflow and BIG-IP

About This Solution

The **Cisco Application Policy Infrastructure Controller** (Cisco APIC) is the unifying point of automation and management for the **Cisco Application Centric Infrastructure** (Cisco ACI™) fabric. The Cisco APIC provides centralized access to all fabric information, optimizes the application lifecycle for scale and performance, supporting flexible application provisioning across physical and virtual resources.

For additional information, visit www.cisco.com/go/apic.

About This Demonstration

This preconfigured demonstration includes:

- Scenario 1: Deploy Service Graph using F5 iApps in Cisco ACI with F5 iWorkflow
- Scenario 2: Modify L4 – L7 deployed graph parameters
- Scenario 3: Remove APIC Service Graph
- Scenario 4: Using POSTMAN REST client to deploy service graph

There are two options to complete each lab task

(1) Using iWorkflow and APIC UI – Scenario 1

(2) Using POSTMAN REST client (APIC Only) – Scenario 4

The goal of ACI is to accelerate application deployment by building L4-L7 policy into Cisco ACI model. We recommend using the REST client model as the most effective way to execute the APIC portion of the lab; for BIG-IP and iWorkflow, please continue to use the UI. You are encouraged to use the UI screen shots as a reference to the tasks executed by POSTMAN.

2.1 Demonstration Requirements

Required	Optional
• Laptop	• Cisco AnyConnect

2.1.1 Demonstration Configuration

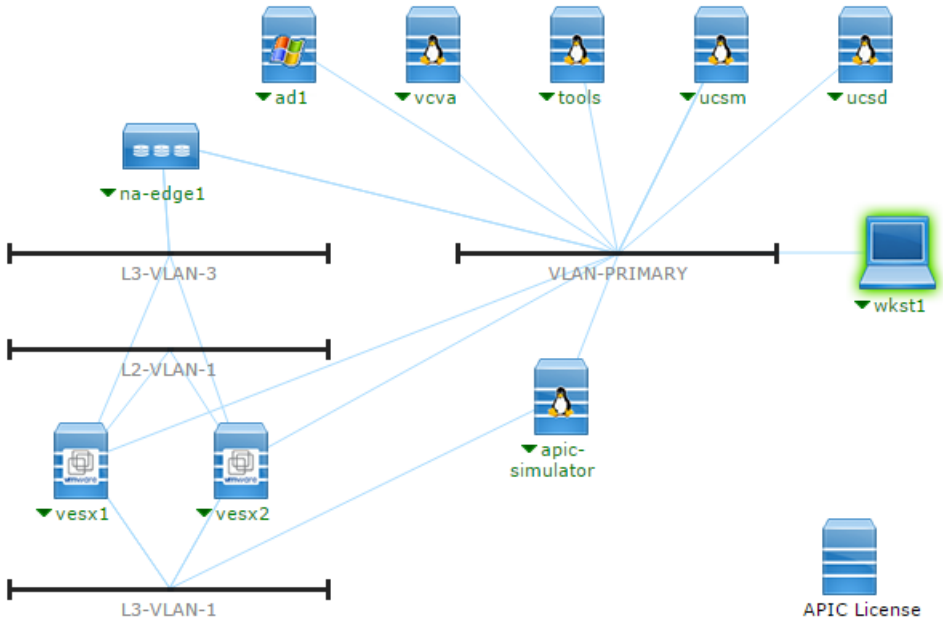
This demonstration contains preconfigured users and components to illustrate the scripted scenarios and features of this Cisco solution. All access information needed to complete the demonstration scenario, is located in the **Topology** and **Servers** menus of your **active demonstration**, and throughout this script.

- **Topology Menu**. Click on any server in the topology to display the available server options and credentials.
- **Servers Menu**. Click on ⊞ or ▶ next to any server name to display the available server options and credentials.

2.1.2 Demonstration Topology

The following is the virtual demonstration topology, which consists of the following virtual machines:

- APIC Simulator – version 2.1(1h)
  - APIC1, APIC2, APIC3
  - Leaf1 and Leaf2
  - Spine1 and Spine2
- VMware Virtual Center Server 5.5 Appliance
- F5 iWorkflow – release 2.0.2
- F5 BIG-IP – release 12.0.0 HF4
- VMware ESXi 5.5 Host 1
- VMware ESXi 5.5 Host 2
- Workstation – Windows 8
- NetApp EDGE Storage Appliance – ONTAP 8.2
- Linux Tools Repository (Ubuntu 12.04)

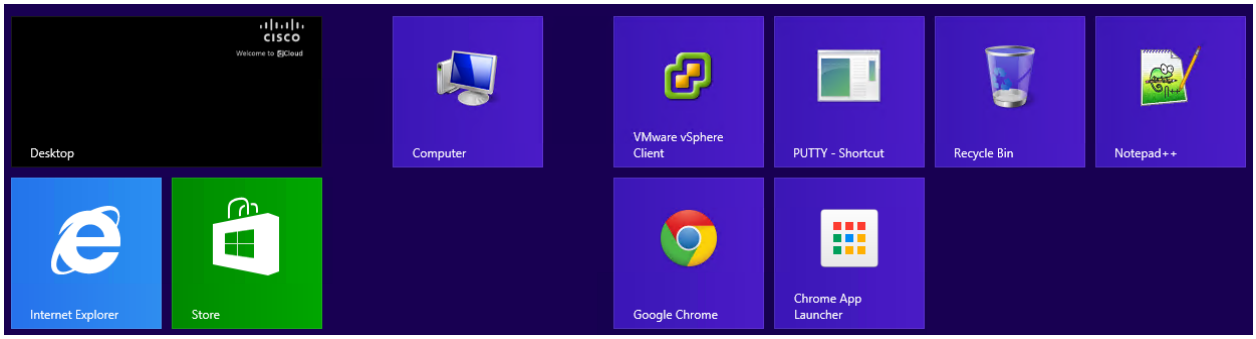


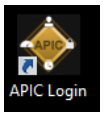
This demonstration contains preconfigured users and components to illustrate the scripted scenarios and features. All access information needed to complete the scripted scenarios is located in the **Topology** and **Servers** menus of your **active demonstration**, and throughout this script.

2.1.3 Demonstration Preparation

Follow the steps below to schedule and configure your environment.

1. Browse to dcloud.cisco.com, choose the location closest to you, and then login with your Cisco.com credentials.
2. Schedule a session. [Show Me How].
3. Test your bandwidth from the demonstration location before performing any scenario. [Show Me How]
4. Verify your session has a status of **Active** under **My Demonstrations** on the **My Dashboard** page in the Cisco dCloud UI.
5. It may take up to **15 minutes for your demo to become active**.
6. Access the workstation named **wkst1** located at **198.18.133.36** and login using the following credentials: Username: **dcloud\demouser**, Password: **C1sco12345**.
7. **Option 1**: **(Preferred)** Use **Cisco AnyConnect** [Show Me How] and the **local RDP client** on your laptop [Show Me How].
   - Accept any certificates or warnings.
   - From the **Start** menu, click **Desktop**.
8. **Option 2**: Use the **Cisco dCloud Remote Desktop client with HTML5**. [Show Me How]
   - Accept any certificates or warnings.
   - From the **Start** menu, click **Desktop**.
9. Start Menu

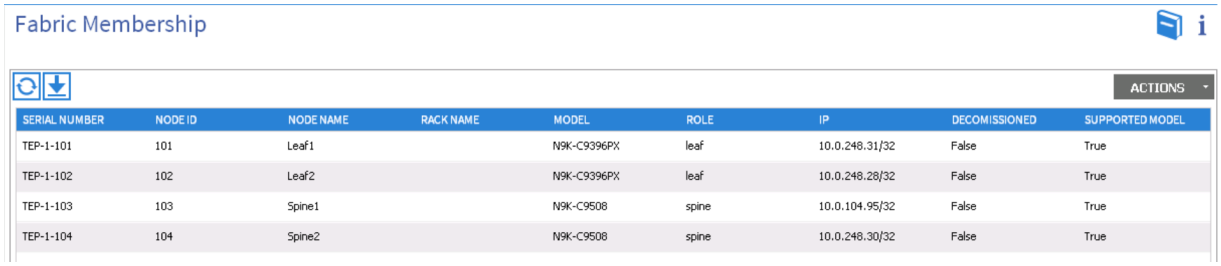


10. The fabric discovery is automatically started at demo setup. Double-click the **APIC Login** icon 

and login (`admin/C1sco12345`).
11. Select **Fabric** from the top menu.
12. Select **Inventory** from the top sub-menu.
13. In the left menu, click **Fabric Membership** and check that you have the 4 devices populated as shown in Figure 3. (IP addresses may vary.)

Note: The fabric discovery can take up to 15 minutes to complete. If you login before 15 minutes have passed, all devices may not be fully discovered.

1. Completed Fabric Membership

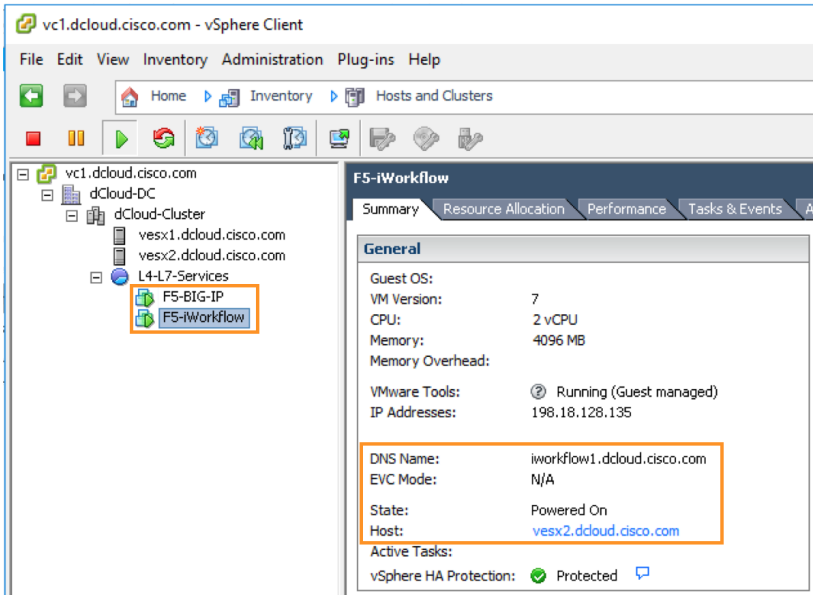


SERIAL NUMBER	NODE ID	NODE NAME	RACK NAME	MODEL	ROLE	IP	DECOMISSIONED	SUPPORTED MODEL
TEP-1-101	101	Leaf1		N9K-C9396PX	leaf	10.0.248.31/32	False	True
TEP-1-102	102	Leaf2		N9K-C9396PX	leaf	10.0.248.28/32	False	True
TEP-1-103	103	Spine1		N9K-C9508	spine	10.0.104.95/32	False	True
TEP-1-104	104	Spine2		N9K-C9508	spine	10.0.248.30/32	False	True

Note: To demonstrate Fabric Discovery, reset the APIC Simulator (see *Appendix A*.) If only TEP-1-101 is present at login, see *Appendix B* to discover the Fabric.

1. Double-click the **VI Login** icon 

and login with the following credentials: Username: **demouser**, Password: **C1sco12345**. (If password is grayed out, click **Login**.)
2. Check that the **F5 iWorkflow and BIG-IP** virtual machine is present and running as below.
3. Virtual Center Inventory



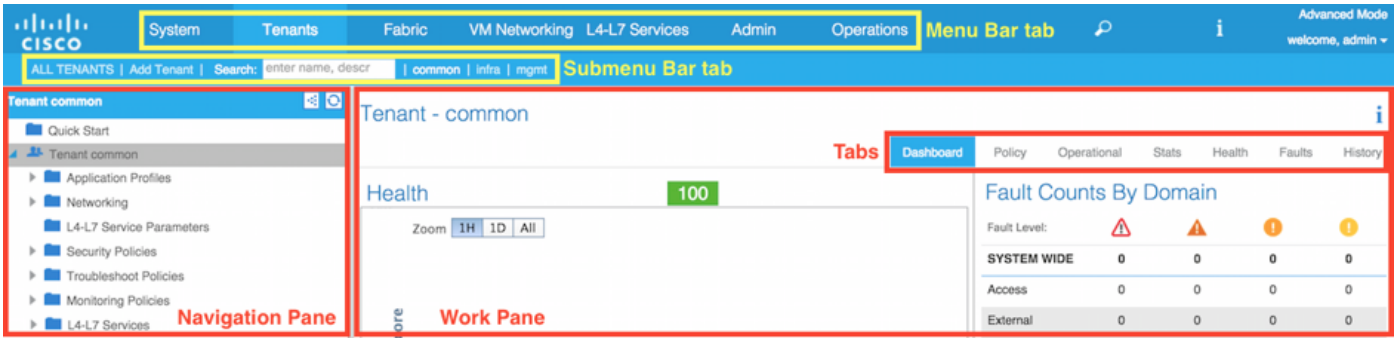
Note: If the F5 BIG-IP and iWorkflow VMs are not present in the L4-L7 Services Resource Pool, *add it manually*.

2.2 Deploy Service Graph using F5 iApps in Cisco ACI with F5 iWorkflow

2.2.1 Overview

Cisco Application Centric Infrastructure (ACI) technology provides the capability to insert Layer 4 through Layer 7 (L4-L7) functions using an approach called a service graph. One of Cisco ACI's changes to the operation model with the service graph function is that a configuration now includes not only the network connectivity consisting of VLANs, IP addresses, etc., but also the configuration of access control lists, load-balancing rules, etc., on service appliances, such as the firewalls and load balancers. This approach differs from the traditional operation model of service insertion. Prior to Cisco ACI, the fabric configuration would have consisted only of connectivity for firewalls and load balancers. With Cisco ACI, the service graph configuration includes the ability to push configuration of firewalls and load balancers from ACI.

2.2.2 APIC GUI Layout



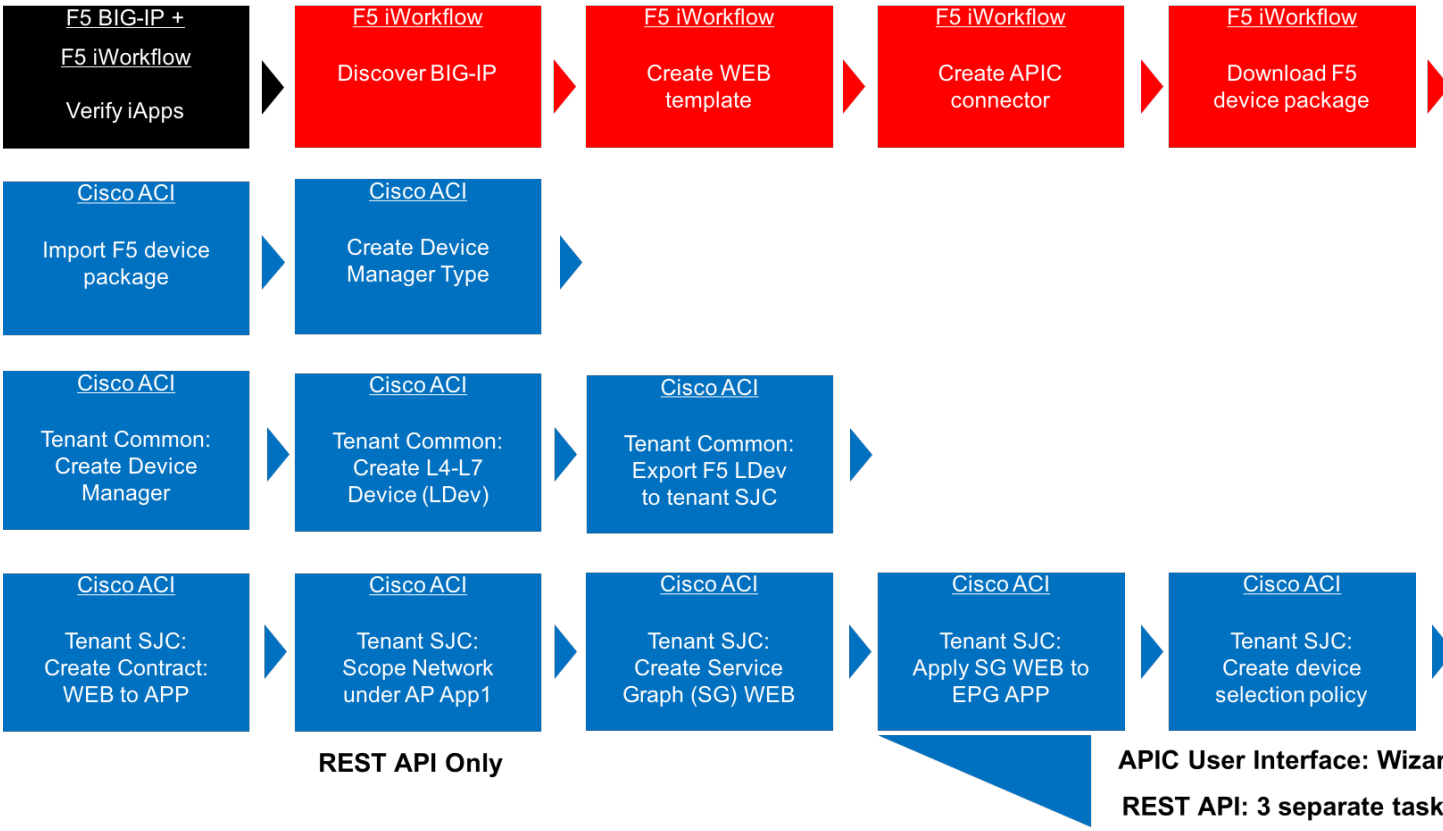
The top of the GUI screen is the Menu bar tab, the middle of the GUI is the Submenu bar tab, the bottom left of the GUI screen is the Navigation Pane, and the middle-right of the GUI is the Work Pane.

2.2.3 F5 iWorkflow and Cisco ACI Lab

The goal of this lab is to demonstrate a WEB application deployment that has L4-L7 ADC requirements in ACI environment. Using F5 iWorkflow service catalog model, the WEB application ADC requirements are defined in iWorkflow service catalog template using F5 iApps technology. Thru F5 dynamic device package, this service catalog is imported into ACI. In Cisco ACI, when deploy application WEB, administrator can now pick WEB template to apply ADC functionality to application WEB.

To achieve this scenario, you will configure ACI L4-L7 service insertion in managed mode with device manager using F5 BIG-IP VE Virtual ADC and F5 iWorkflow orchestration + automation platform using User Interface.

2.2.4 F5 iWorkflow and Cisco ACI Lab Flow Chart



2.2.5 BIG-IP – Verify the F5 BIG-IP iApps

F5 iApps is a user-customized framework for deploying application, providing a flexible way to automate tasks and templatize F5 virtual server configurations.

The iApps must be imported into F5 BIG-IP in order to allow F5 iWorkflow to create an application template based on this iApps. In this step, we will verify the iApps is already exist in F5 BIG-IP.

Log into the F5 BIG-IP with the following username and password from the web browser:

BIG-IP: `https://198.18.128.130`

Username: `admin`

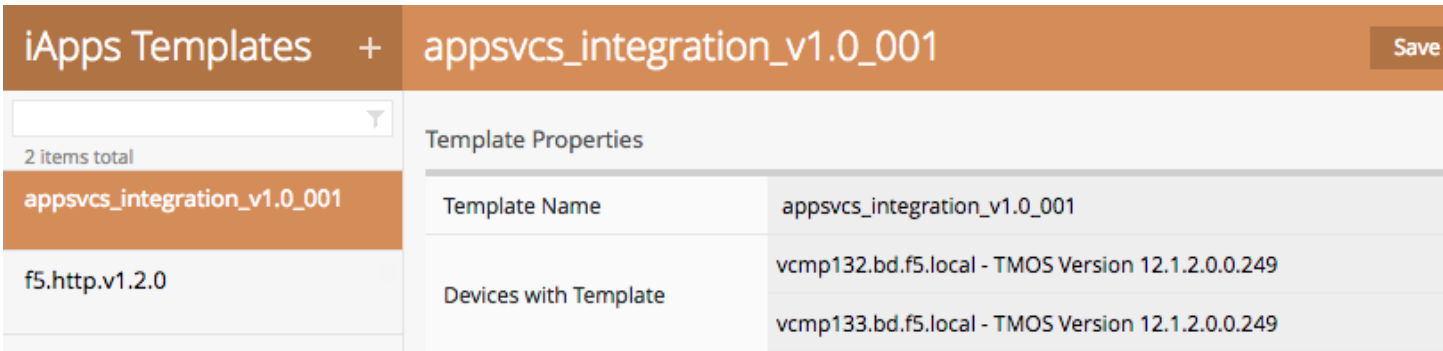
Password: `C1sco12345`

After you have logged into the F5 BIG-IP GUI. In the Navigation pane, click the iApps -> Templates. You should see the iApps template **appsvcs_integration_v1.0_001** pre-loaded into the F5 BIG-IP:

Note: Up to iWorkflow release 2.0.2, iApps to be used by iWorkflow / APIC integration must be exist in BIG-IP in order for iWorkflow to be discovered. Beginning iWorkflow release 2.1.0, user import iApps into iWorkflow and iWorkflow will push the iApps to BIG-IP

2.2.6 iWorkflow – Set up the F5 iWorkflow Clouds and Services

F5 iApps template is **ALREADY** added in iWorkflow:



F5 iWorkflow Clouds and Services allows administrator to create a cloud connector to Cisco APIC by generating a customized device package that contains the service catalog. It is also where administrator can manage service catalog life cycle.

In this step, we will configure F5 iWorkflow prior to Cisco ACI integration.

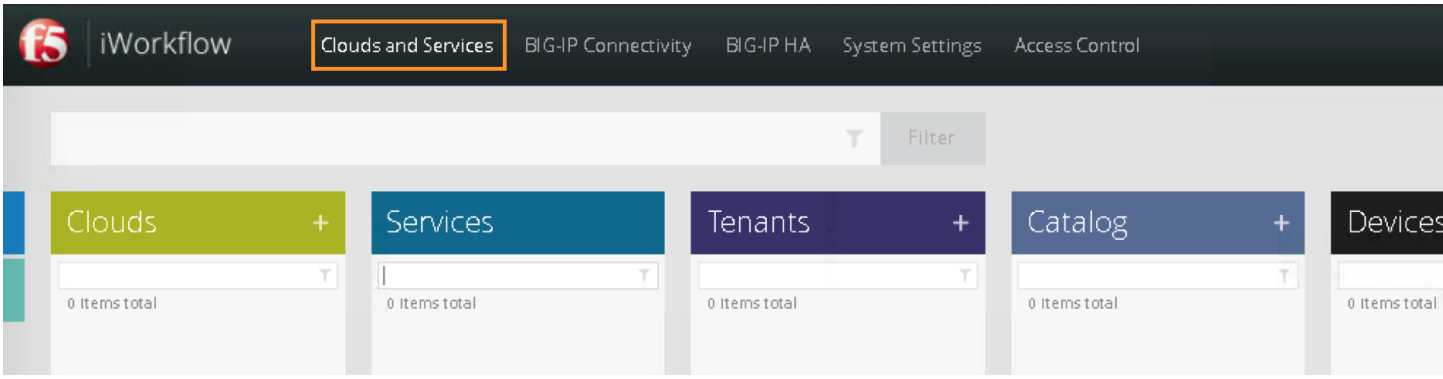
Log into the F5 iWorkflow `198.18.128.135` with the following username and password from the web browser:

iWorkflow: `https://198.18.128.135`

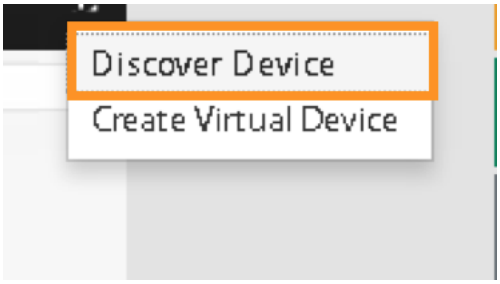
Username: `admin`

Password: `C1sco12345`

After you have logged into the F5 iWorkflow GUI. Click on “Clouds and Services”, select “+” Devices



Register F5 BIG-IP by selecting “Discover Device”



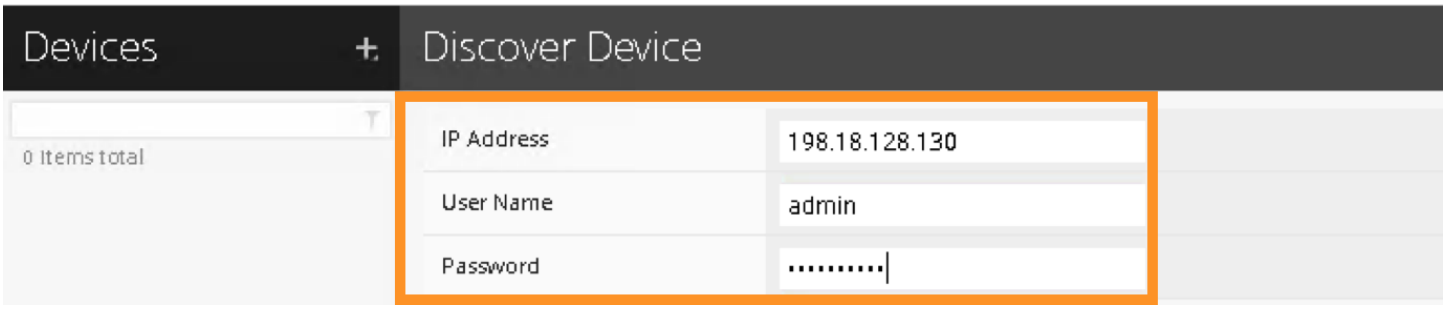
Register the F5 BIG-IP by using the BIG-IP’s IP address and credential as the following:

IP Address: `198.18.128.130`

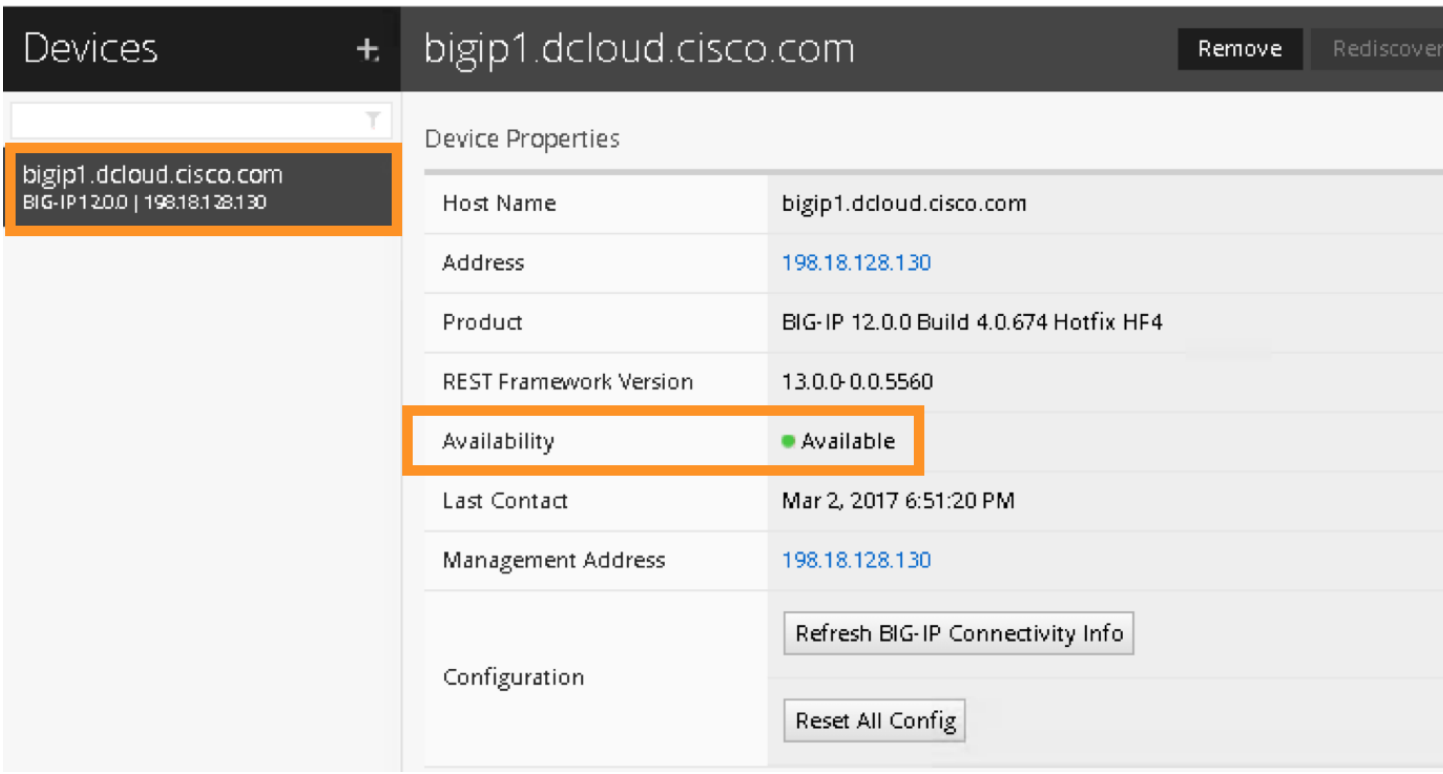
Username: `admin`

Password: `C1sco12345`

Click Save to register the BIG-IP device:



You can now double click the registered BIG-IP and verify its status. It should say “Available” when the BIG-IP is communicating with the iWorkflow:

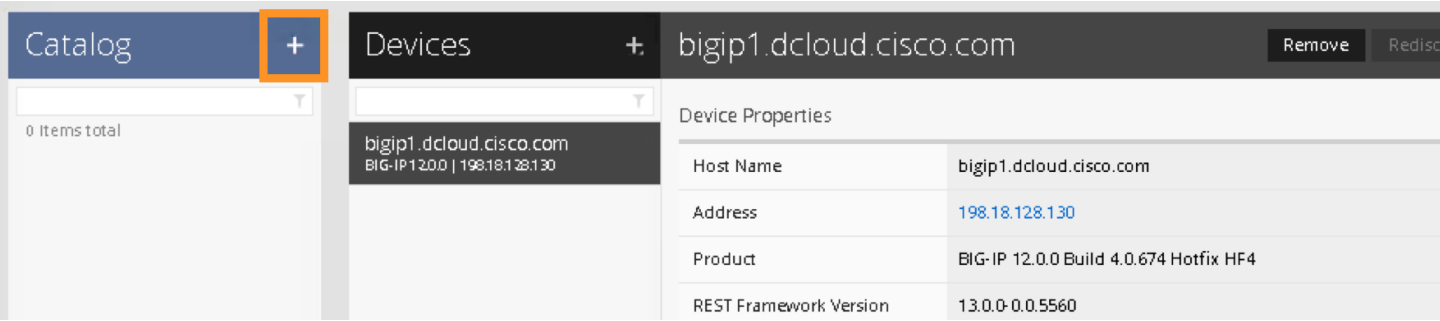


2.2.7 iWorkflow – Create WEB application template in iWorkflow Catalog

After BIG-IP is successfully discovered by iWorkflow, the iApps reside on BIG-IP are now exposed to iWorkflow.

In this step, we will create a WEB application template based on iApps in iWorkflow Cloud Catalog. We can specify the WEB application F5 virtual server requirements here and build it into a template.

Move your mouse to the left or right side of the screen and the Cloud Catalog menu should appear, click "+" to add a template



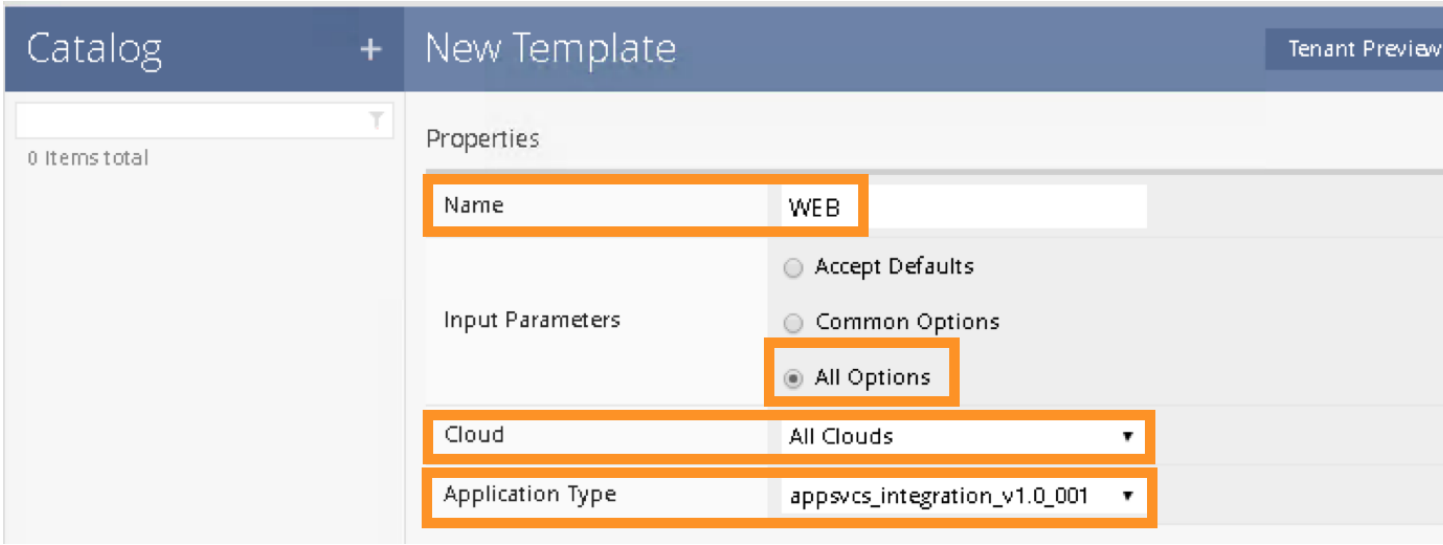
A New Template screen will appear. Enter and select the following in the New Template:

Name: `WEB`

Input Parameters: `All Options`

Cloud: `All Clouds`

Application Type: `appsvcs_integration_v1.0_001`



Note: Only field marked "Tenant Editable" will be visible in Cisco APIC

You can now edit all the available options that need to be included with this template.

Expand the Virtual Server Listener & Pool Configuration by clicking the >. Scroll down and CHECK the following to make them Tenant Editable. What this does is allow the parameters expose to Cisco APIC thru F5 device package. Administrator has total control over what is exposed via a custom device package (this reduces the complexity). It is highly recommended to expose only what is needed to APIC:

`pool__addr`: this is the VIP

`pool__port`: this is the VIP listening port

Note: By default, this iApp allows VIP as tenant editable field. When you check VIP listening port as tenant editable, iWorkflow will highlight it.

Virtual Server Listener & Pool Configuration 14/14

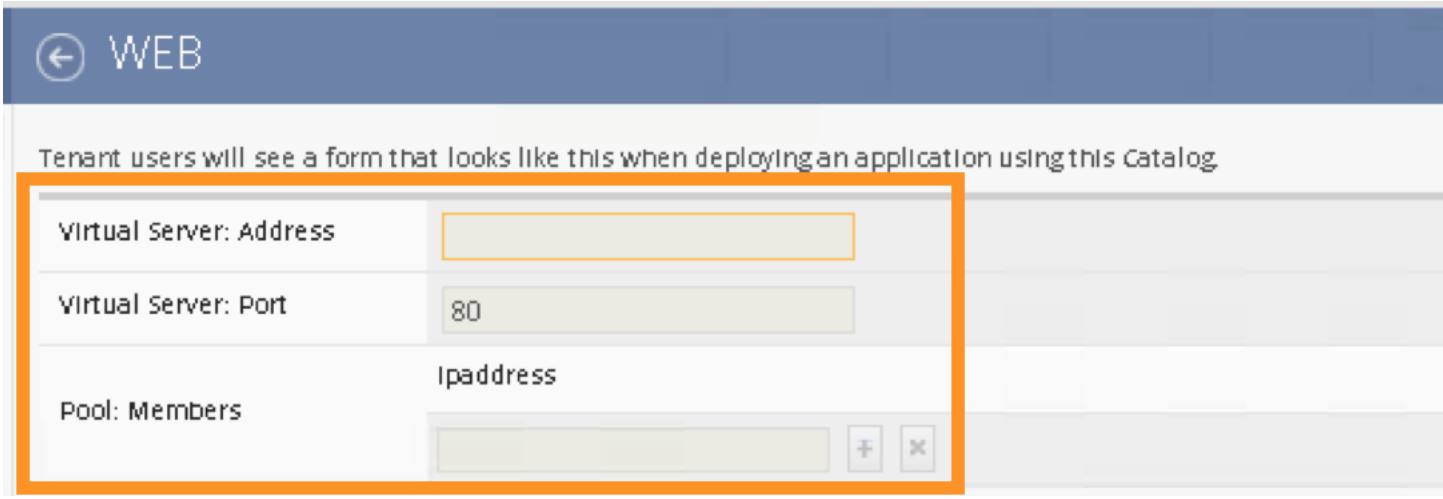
Name	Description	Default Value	Tenant Editable
pool_AdvOptions	Pool: Advanced Options		☐
pool_Description	Pool: Description	pooldescr	☐
pool_LbMethod	Pool: Load Balancing Method	round-robin	☐
pool_MemberDef...	Pool: Member Default Port	80	☐
pool_Monitor	Pool: Health Monitor	/Common/http	☐
pool_Name	Pool: Name		☐
pool_addr	Virtual Server: Address		☑
pool_mask	Virtual Server: Mask	255.255.255.255	☐
pool_port	Virtual Server: Port	80	☑

Click “Tenant Preview” to review the parameters will be visible in Cisco APIC:

New Template Tenant Preview Save Cancel

You should only see 3 parameters:

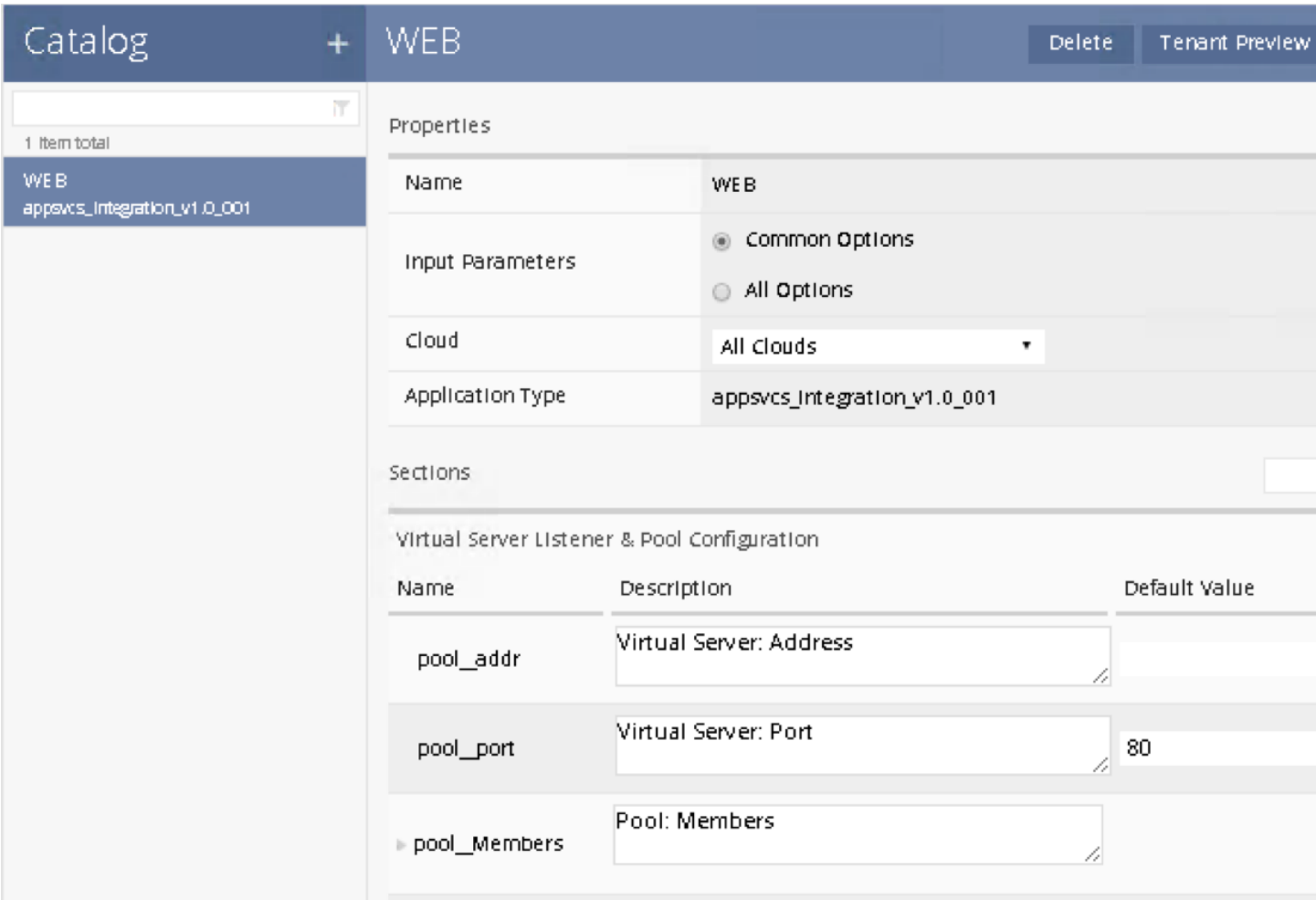
```
Virtual Server:  Address
Virtual Server:  Port
Pool:  Members
```



Click  to go back, then “Save”

Notice a new application template now under iWorkflow Cloud Catalog. The “Save” operation will also update the F5 iWorkflow Cloud APIC device package with the updated service catalog.

This service catalog is ready to be consumed by Cisco APIC.

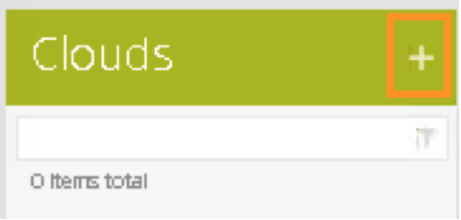


2.2.8 iWorkflow – Create F5 iWorkflow APIC device package

The next step is to create the iWorkflow Cloud APIC Connectors which will generate a custom device package that contains iWorkflow service catalog. The template we created in the previous step will appear in APIC as a service function.

Move your mouse to the left / right side of the screen to make the Clouds menu to appear.

To create a new Connectors, move the mouse to the Clouds menu and the + should appear.

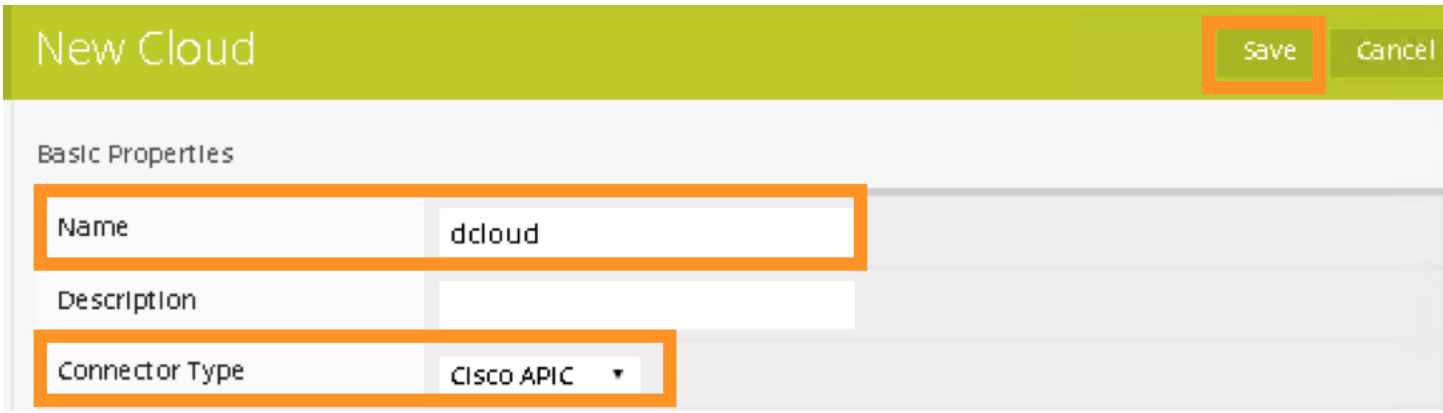


Click “+” to create a new Cloud Connector:

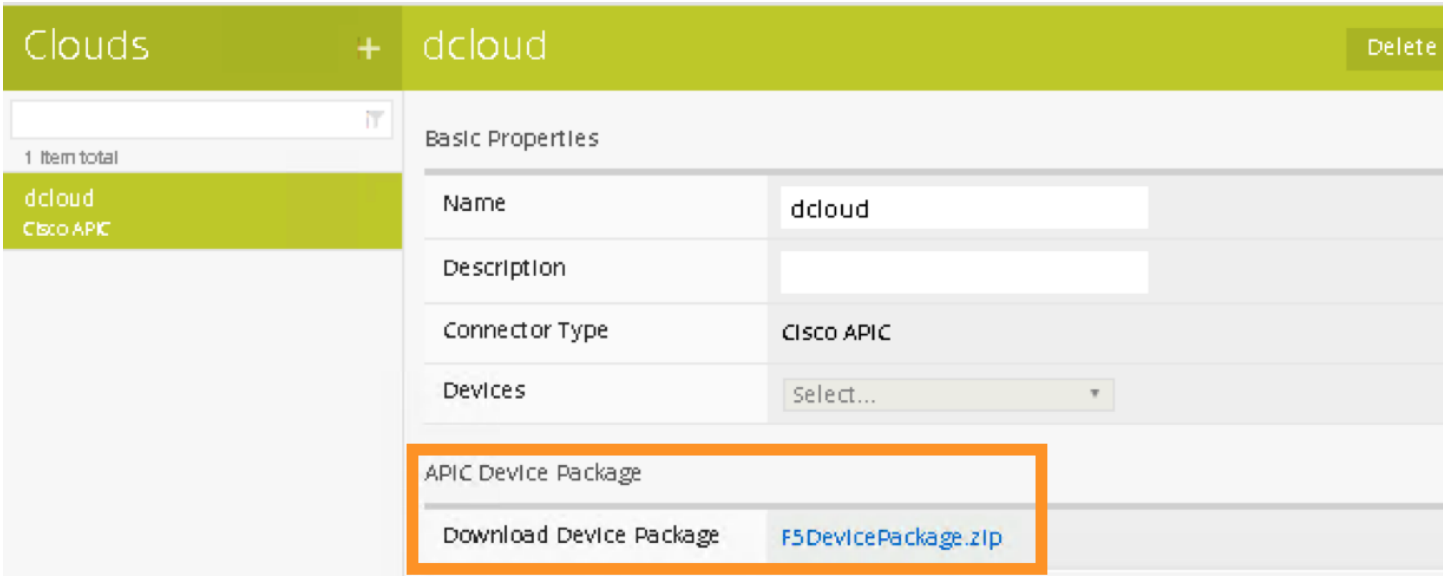
Name: `dcloud`

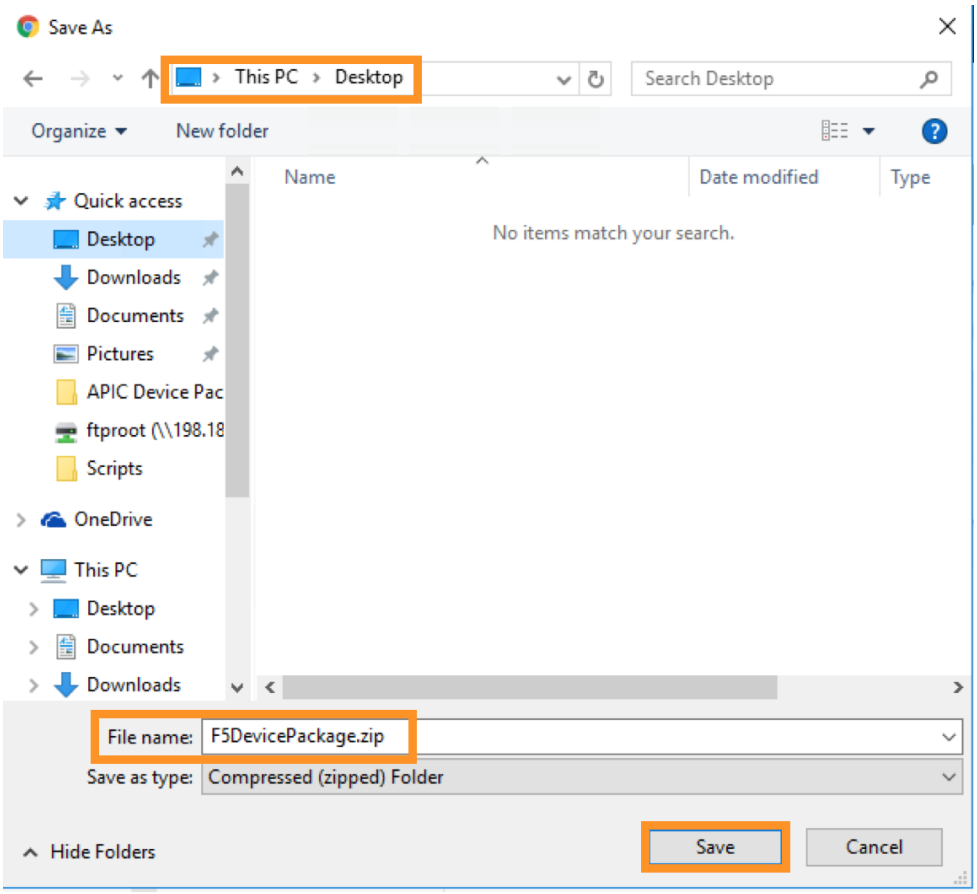
Connector Type: `Cisco APIC`

Click “Save” to finish



Double Click the dcloud connector, you can download this customized device package that contains iWorkflow Catalog to your desktop.





We now complete the configuration steps on iWorkflow necessary prior to F5 ACI integration.

2.2.9 APIC – Import the Custom Device Package

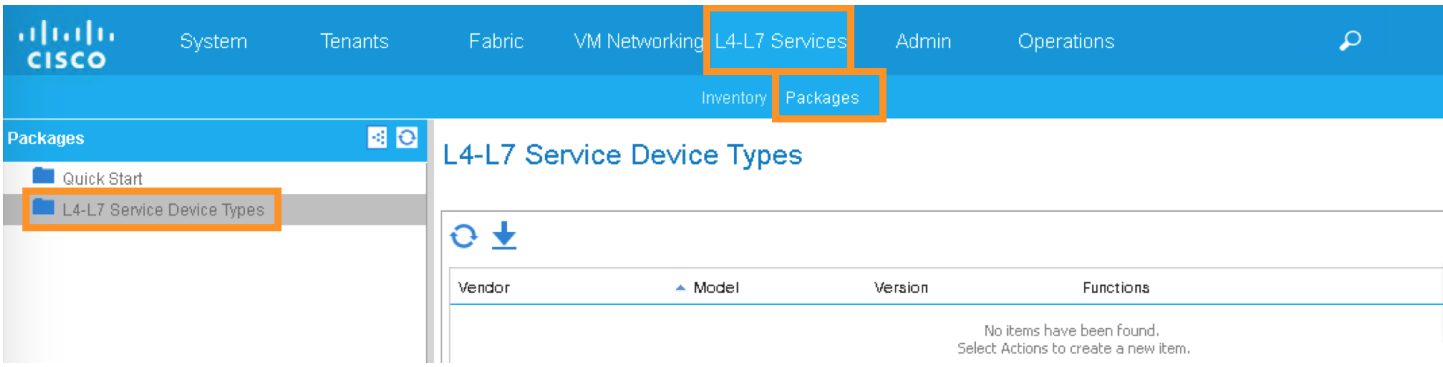
Starting here, you will use Cisco APIC to perform the workflow in deploying the WEB application, with the integration of F5 iWorkflow and BIG-IP, user can apply WEB application L4-L7 requirements within APIC policy model, reducing significant amount of operation complexity.

In this step, you will import the customized device package generated by F5 iWorkflow into Cisco APIC. This will allow the iWorkflow service catalog available in Cisco APIC. The device package serves as a conduit to facilitate communications between F5 iWorkflow and BIG-IP.

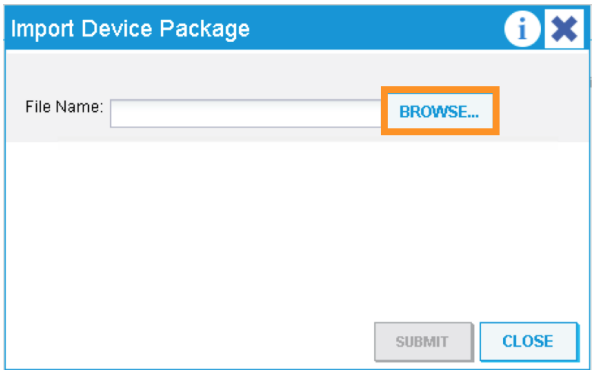
Switch to your APIC GUI and click the following to import the device package:

L4-L7 Services -> Packages -> L4-L7 Service Device Type

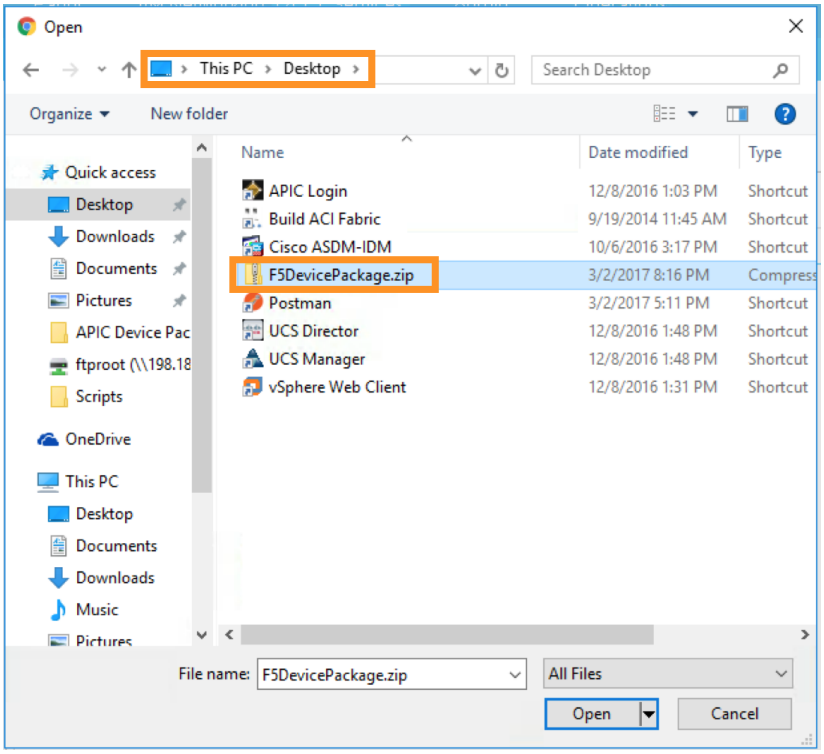
Click the ACTIONS button at the Work pane and choose IMPORT DEVICE PACKAGE



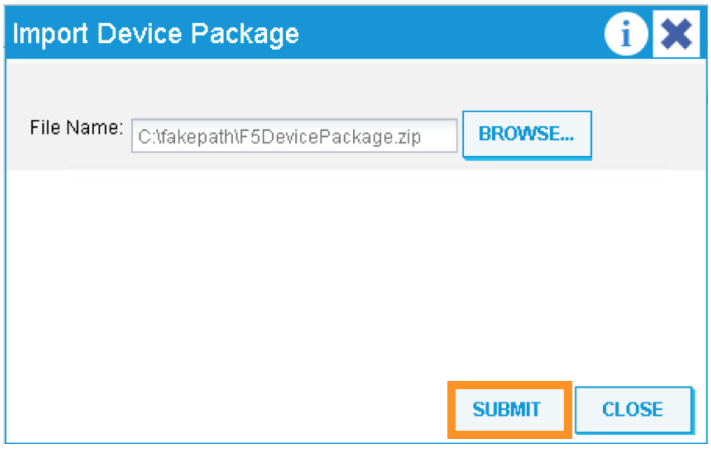
A new pop-up should appear to allow you to choose the device package to be installed, click “Browse”:



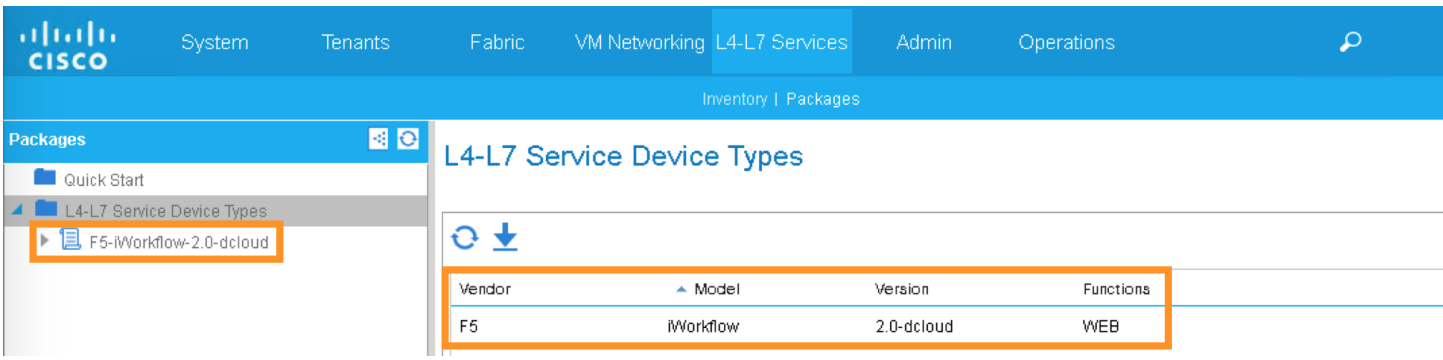
Go to Desktop and select F5DevicePackage.zip



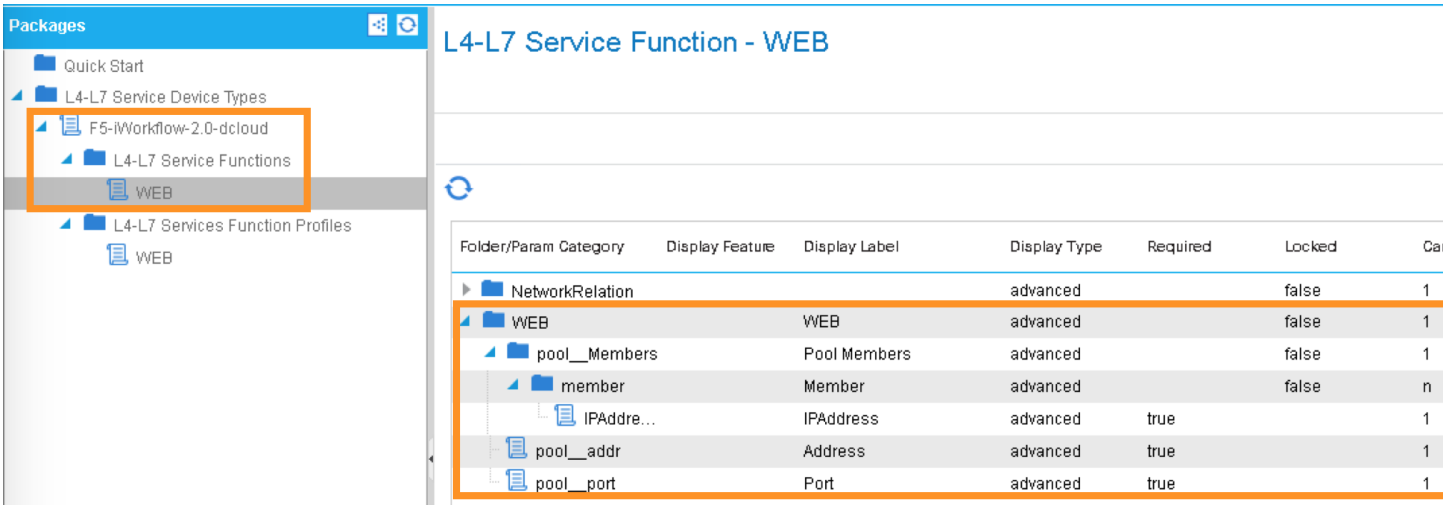
Click “Submit”



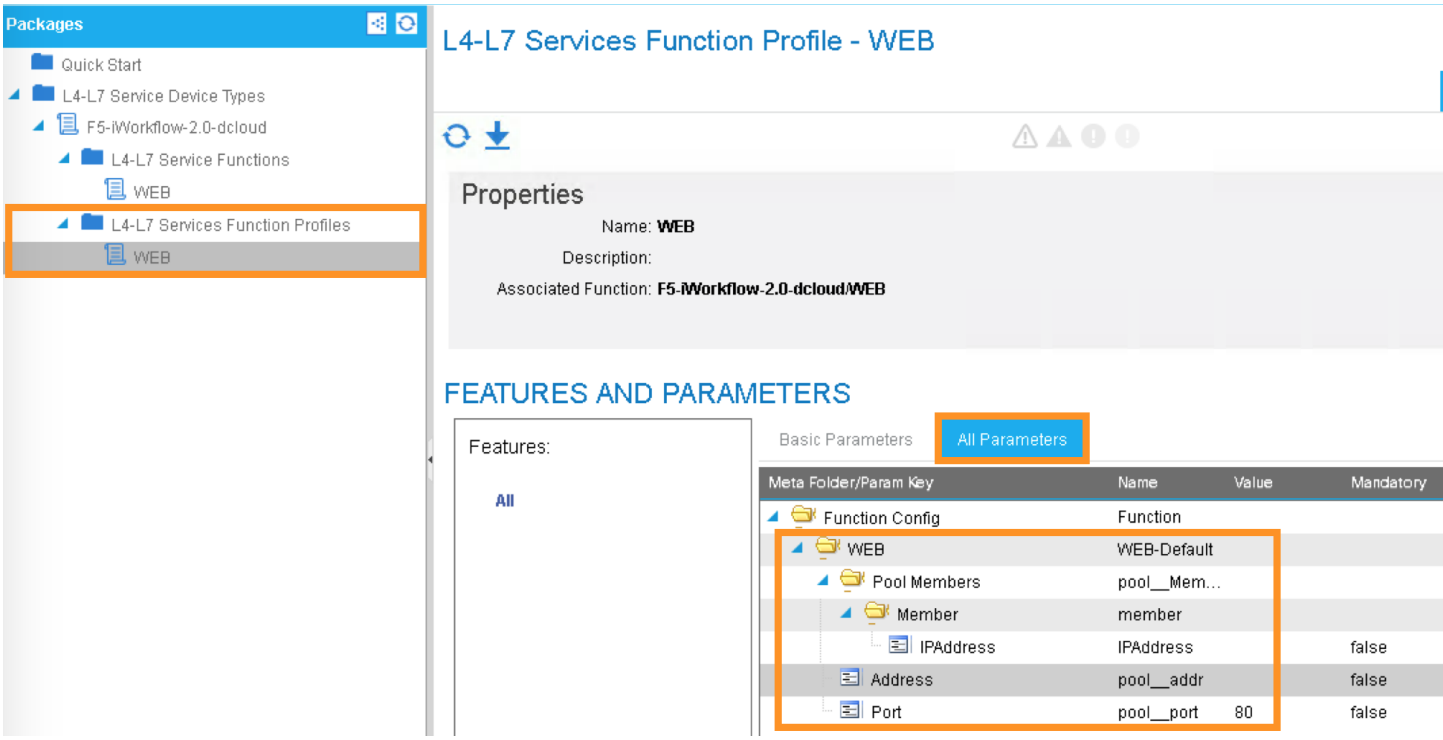
Now F5 device package is imported into APIC



Expand the Device Package, notice Service Function “WEB” is equivalent to iWorkflow Catalog template “WEB”. Under Operational, parameters visible in APIC are the “Tenant Editable” parameters in iWorflow:



Under Function Profiles, you can see if there is any default value assigned to the parameters:



2.2.10 APIC – Create APIC L4-L7 Device Manager under L4-L7 Services

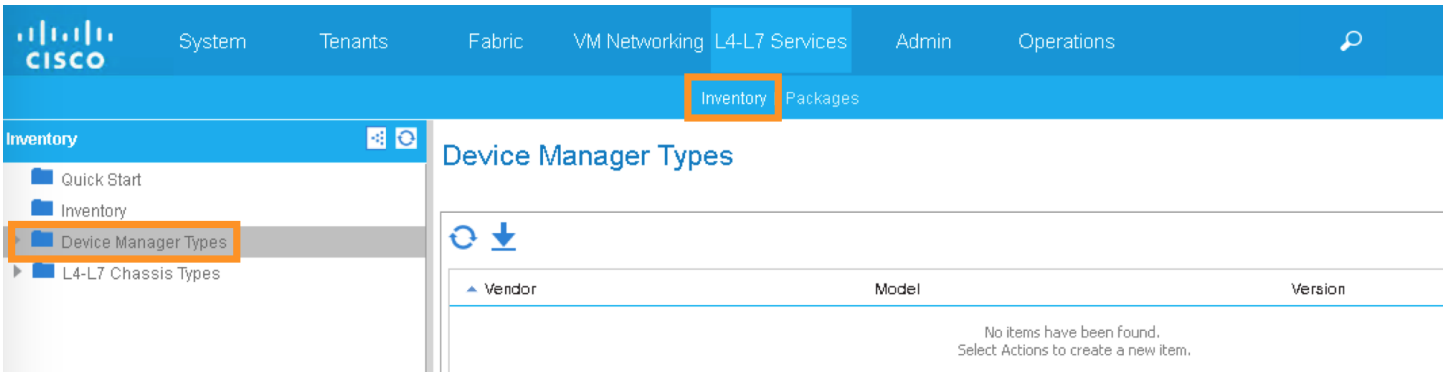
In order to integrate F5 iWorkflow cluster into Cisco APIC L4-L7 devices, we use Cisco APIC device manager feature to define and specify F5 iWorkflow.

From APIC perspective, F5 iWorkflow is a “device manager” managing the F5 BIG-IP ADC (both physical and virtual form factors).

We will first define the device manager type. In the APIC GUI, click the following to configure the Device Manager Type:

L4-L7 Services -> Inventory -> Device Manager Type

Click the ACTIONS button at the Work pane and choose Create Device Manager Type



A new pop-up should appear to allow you to enter the device manager information. Enter the following information:

Vendor: `F5`

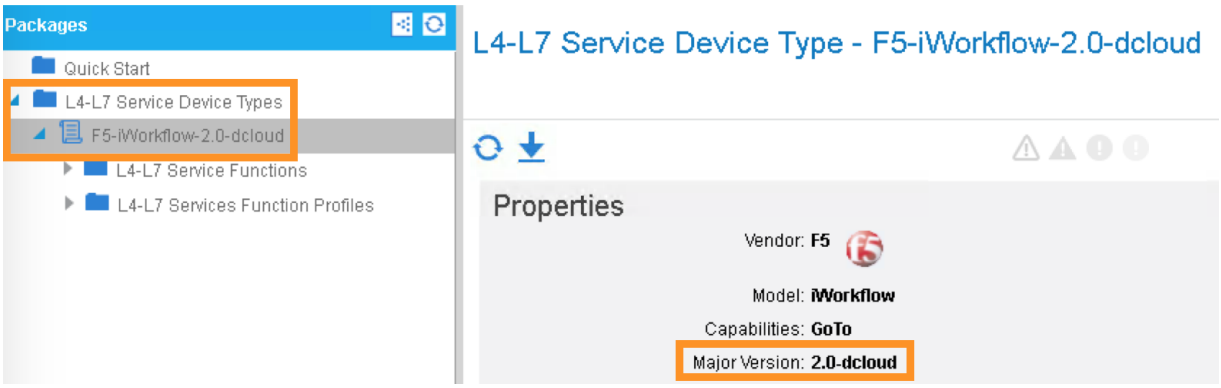
Model: `iWorkflow`

Version: `2.0-dcloud`

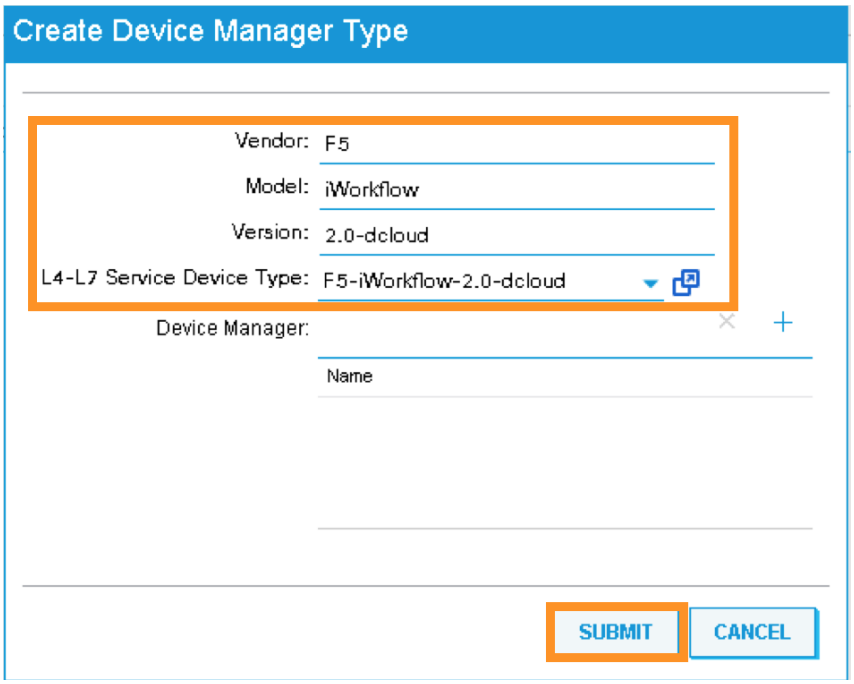
L4-L7 Service Device Type: `F5-iWorkflow-2.0-dcloud`

Device Manager: Leave this field empty

Note: It is extremely import to match the Version number with the major version of the device package



Click SUBMIT to accept the configuration.



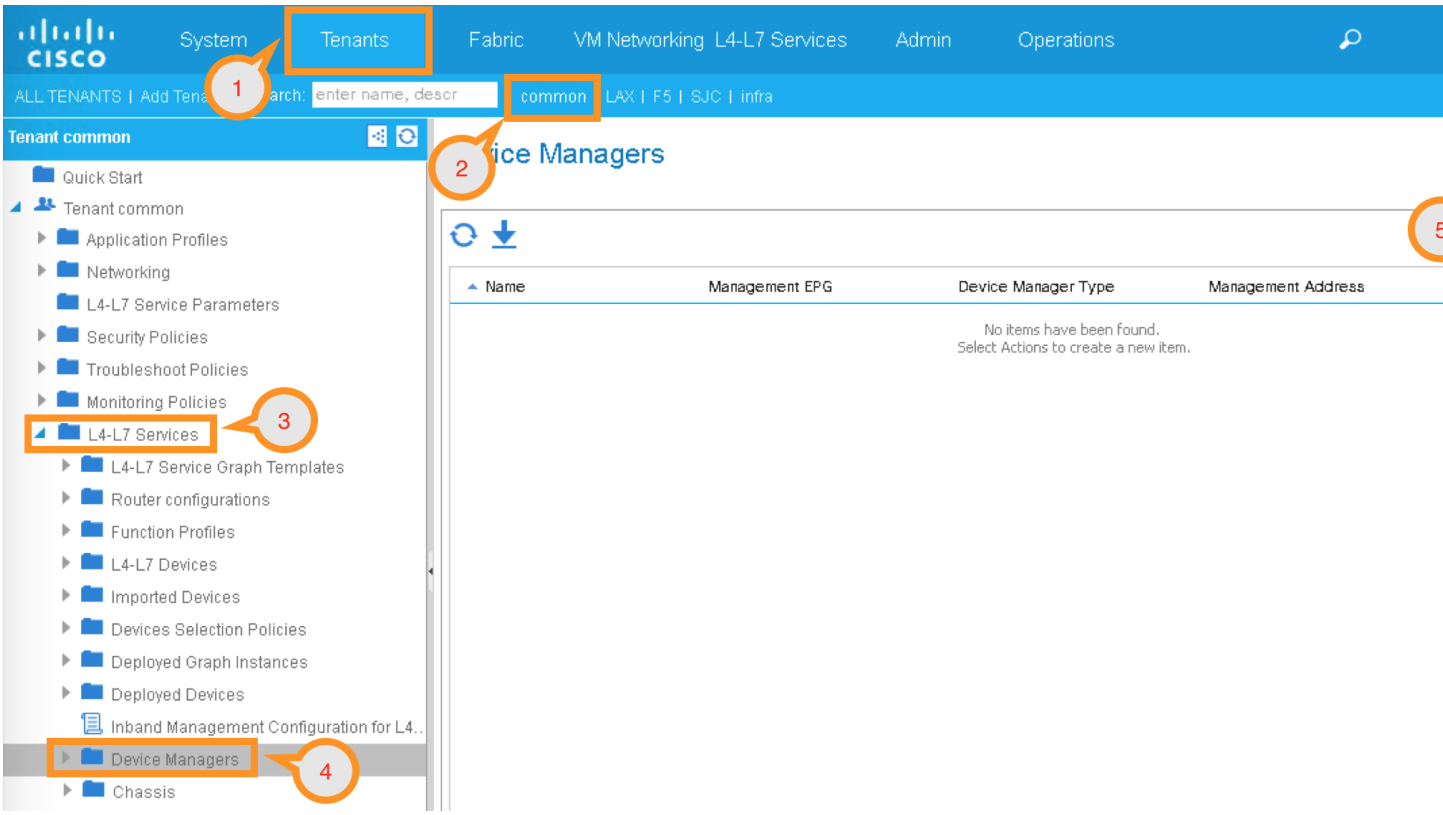
The Device Manager Type is now configured and we can now associate this device manager type with a device manager.

2.2.11 APIC – Create Device Manager under Tenant Common

To create a device manager, navigate to your tenant common to create a new L4-L7 Device Manager by clicking the following:

Tenants Common -> L4-L7 Services -> Device Managers

In the Work pane, click: ACTIONS -> Create Device Manager



A new pop-up should appear to allow you to Create Device Manager in your tenant. You will specify F5 iWorkflow management IP here and associate it with the device manager type created in the previous step. Enter the following information:

Device Manager Name: dcloud-device-manager

Management EPG: Leave this field empty since we use OOB to communicate

Device Manager Type: `F5-iWorkflow-2.0-dcloud`

Click the + to enter the iWorkflow management IP for device manager Management connectivity:

Host: `198.18.128.135`

Port: `443`

Click UPDATE to accept.

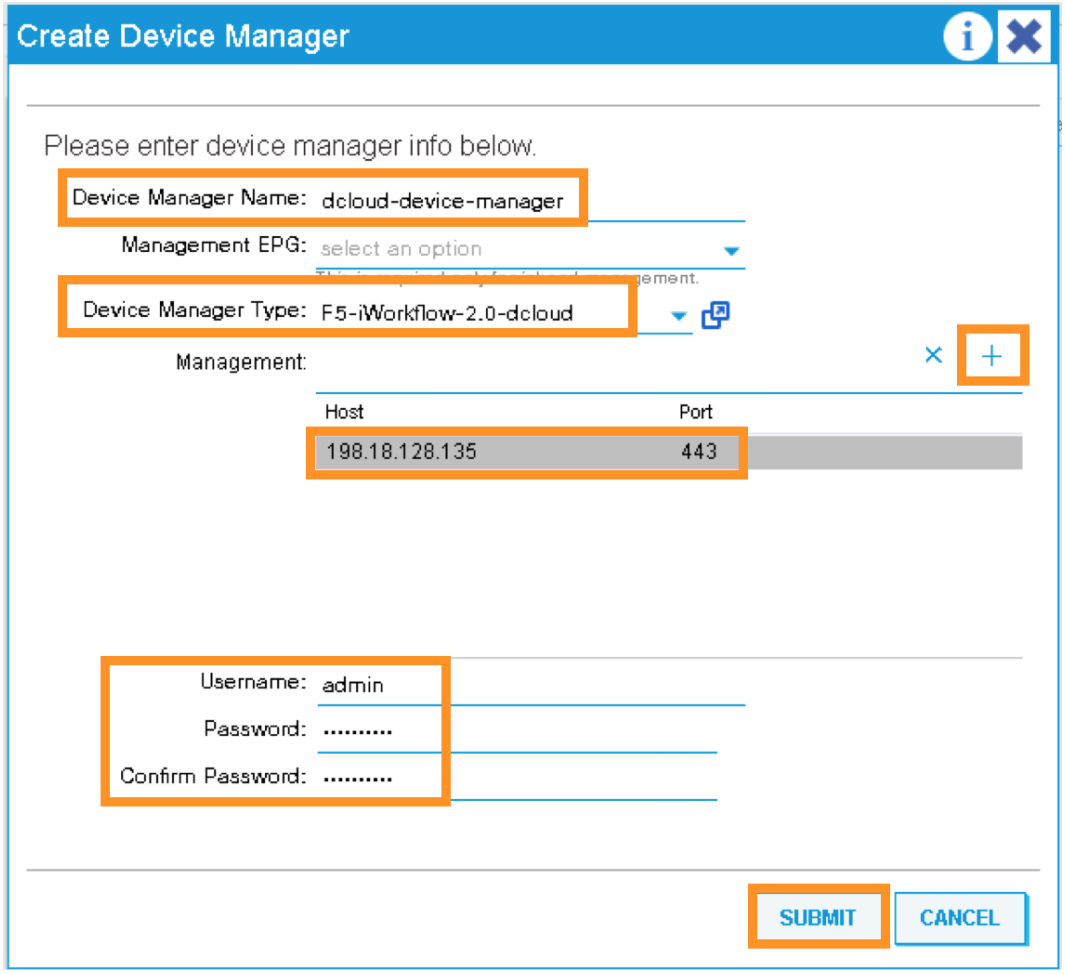
Enter the Device Manager's login credential:

Username: `admin`

Password: `C1sco12345`

Confirm Password: `C1sco12345`

Click SUBMIT to accept the configuration.



This complete the steps to create APIC L4-L7 device manager. We will use this device manager in the next step when creating APIC L4-L7 device.

2.2.12 APIC – Create the L4-L7 Device

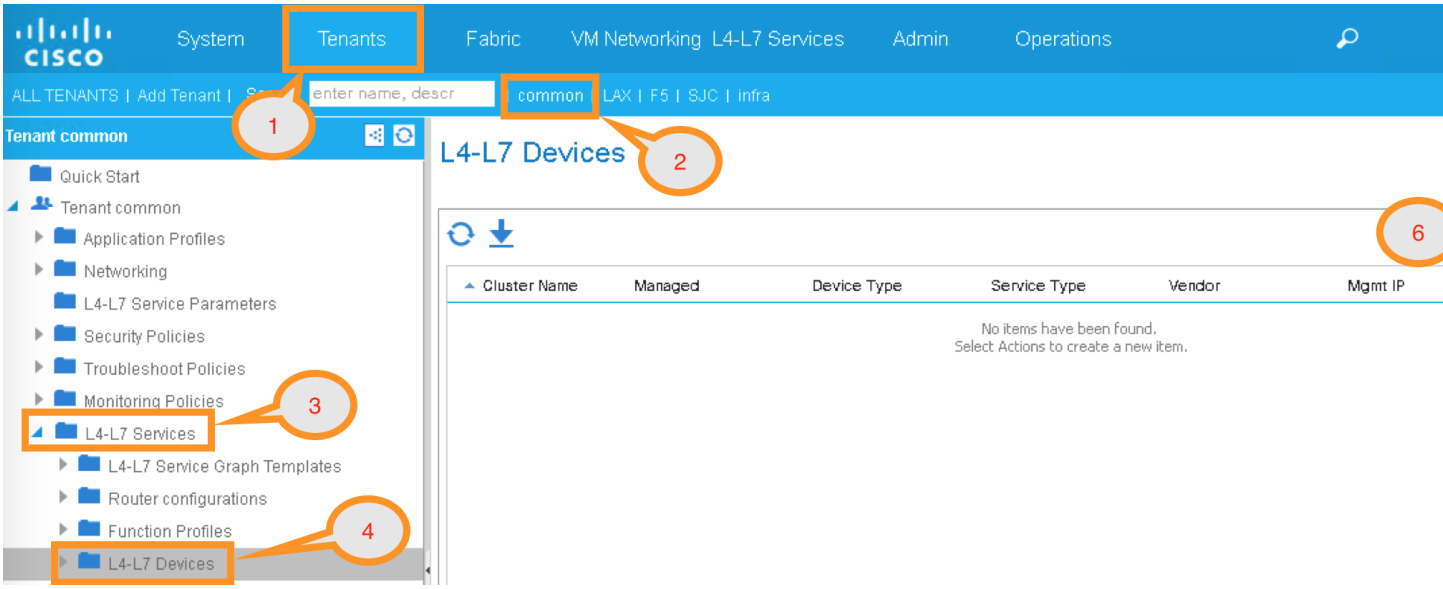
In this step, we will create an APIC L4-L7 device, this is the logical construct that contains F5 BIG-IP and iWorkflow information. You will see in the later steps on how to build an APIC service graph using this L4-L7 device.

Navigate to your tenant to create a new L4-L7 Device by clicking the following:

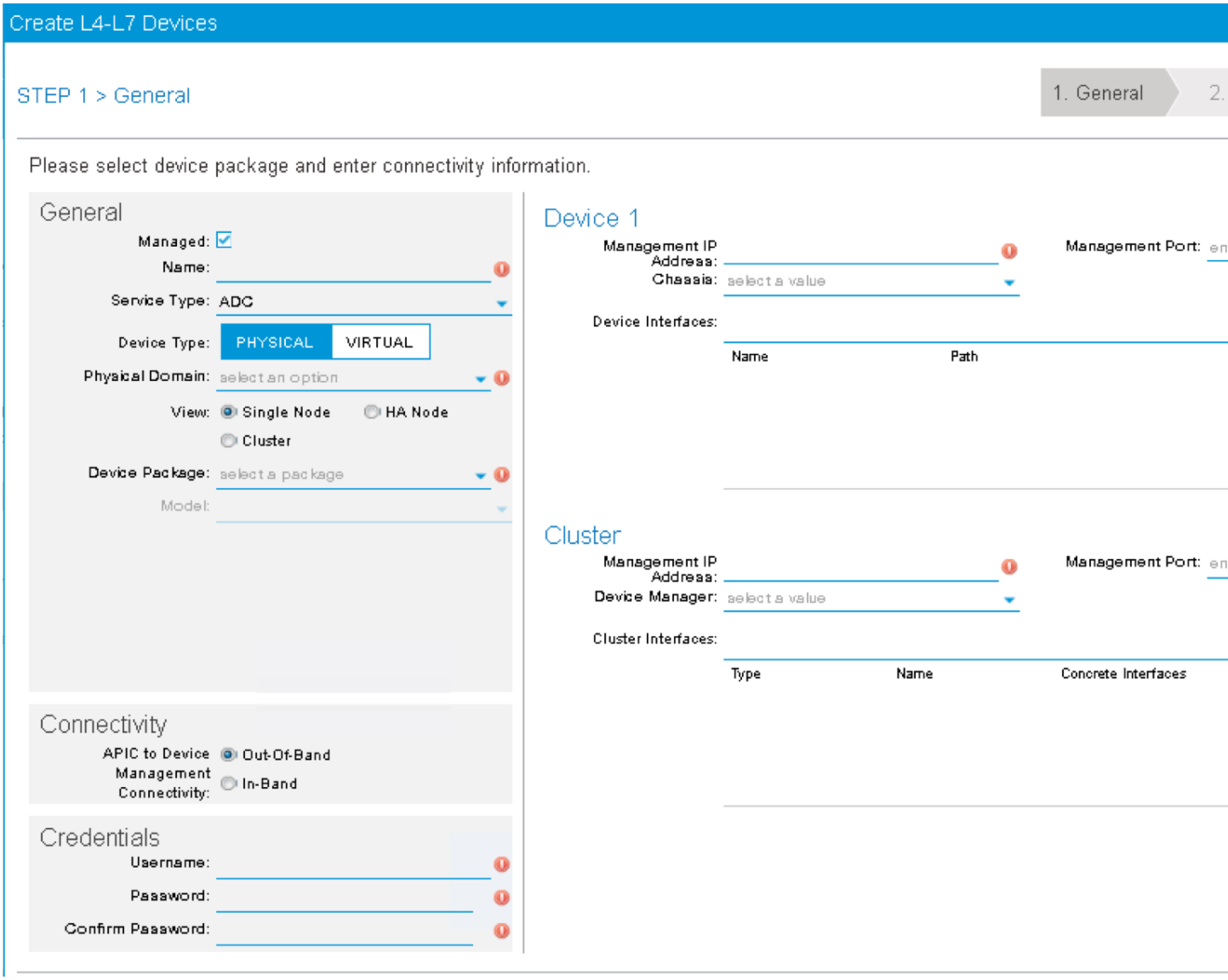
Tenants Common -> L4-L7 Services -> L4-L7 Devices

In the Work pane, click:

ACTIONS -> Create L4-L7 Devices



A new window should appear for you to create the L4-L7 Devices.



In the Create L4-L7 Devices window, enter the following:

Managed: `CHECK`

Name: `F5-BIG-IP`

Service Type: `ADC`

Device Type: `Virtual`

VMM Domain, click the down arrow to select: `My-vCenter`

Mode: `Single Node`

Device Package: `F5-iWorkflow-2.0-dcloud`

Model: `Unknown (Manual)`

Context Aware: `Single`

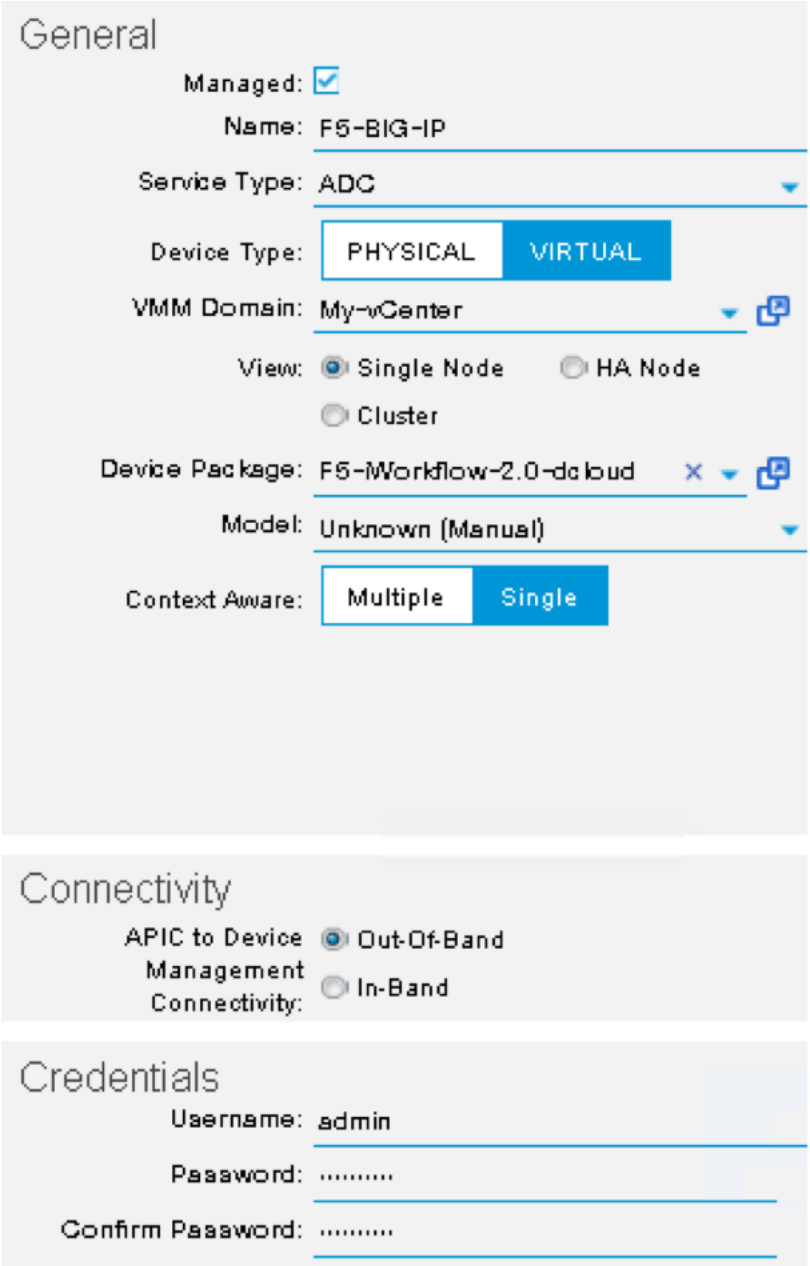
APIC to Device Management Connectivity: `Out-Of-Band`

Username: `admin`

Password: `C1sco12345`

Confirm Password: `C1sco12345`

After completion, it should look like:



What did I configure?

Managed: this means this L4-L7 device will be managed by Cisco APIC to be used in L4-L7 service insertion

Name: User defined name of the L4-L7 device

Service Type: Firewall or ADC, F5 BIG-IP is considered an ADC device

Device Type: Physical or Virtual, we use BIG-IP Virtual Edition in this lab

VMM Domain: If device type is virtual, select the VMM domain for this L4-L7 device, the VMM domain

contains BIG-IP VE virtual machine

Mode: Single or HA, in this lab, only one BIG-IP VE, so select Single Node

Device Package: Drop down menu, pick the device package dcloud

Model: Choose Unknown(Manual) giving you flexibility to enter any F5 BIG-IP interface convention

Context Aware: Single Context device can be used by only 1 tenant; where Multi Context device can be shared among multiple tenants. In the case of virtual, we will select single context

APIC to Device Management Connectivity: All management connections are out-of-band in this lab Credentials: F5 BIG-IP admin credentials

On the right-hand side of the wizard, in the Device 1, enter the following:

Management IP Address: `198.18.128.130`

VM: Click the down arrow and select `dcloud-DC/F5-BIG-IP`

Management Port: `https`

Click the + to add a Device Interface:

Name: `1_1`

VNIC: `Network adapter 2`

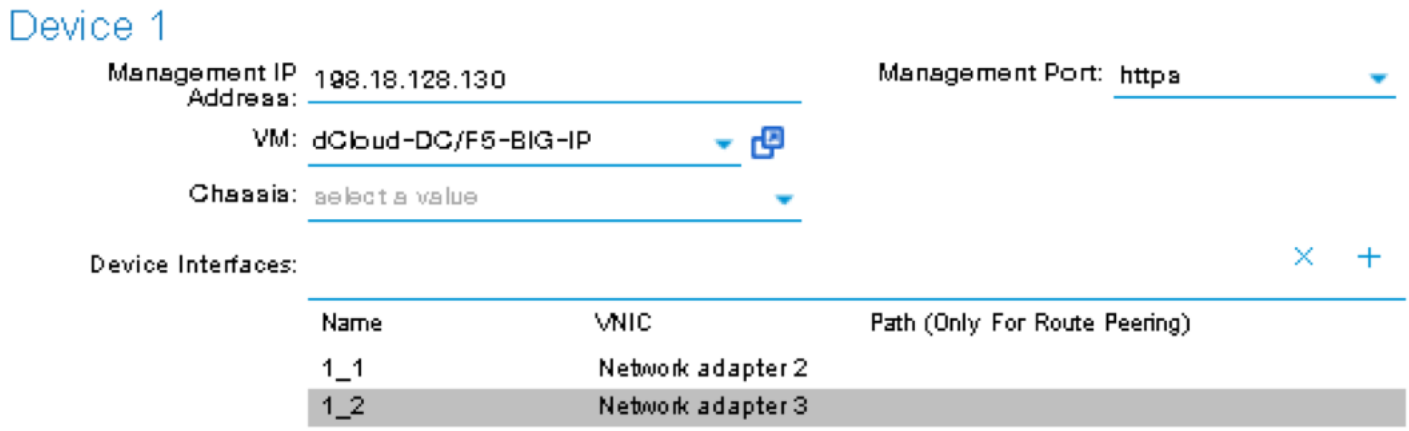
Click UPDATE to accept the Device Interface configuration.

Click the + to add 2nd Device Interface:

Name: `1_2`

VNIC: `Network adapter 3`

Click UPDATE to accept the Device Interface configuration.



What did I configure?

Under Device 1, enter the BIG-IP VE management IP and management port of https (443)

Since this is a BIG-IP VE cluster, the VM field is visible and based on the VMM domain specified earlier, pick the VM for this L4-L7 device.

Device Interfaces: specify the BIG-IP VE interface to be used in data plane. We are configuring physical 2-arm in this lab, two BIG-IP interfaces are specified in this cluster. Notice the interface naming is 1_1, which is equivalent to interface 1.1 of BIG-IP. "_" is used instead of "." is because APIC does not allow "." as parameter value.

Next part of the configuration is L4-L7 device cluster information.

By default, APIC will populate Device 1's management IP as the Cluster Management IP. In this lab, since we are going to use the iWorkflow to manage BIG-IP, the Cluster IP will be changed to the iWorkflow's IP. The device will eventually ignore this setting and it will use the Device Manager information configured earlier to establish communication.

Management IP Address: `198.18.128.135`

Management Port: `https`

Device Manager: `common/dcloud-device-manager`

Click the + to add the 1st Logical Interface:

Type: `consumer`

Name: `External`

Concrete Interface: `Device1/1_1`

Click UPDATE to accept the consumer interface configuration.

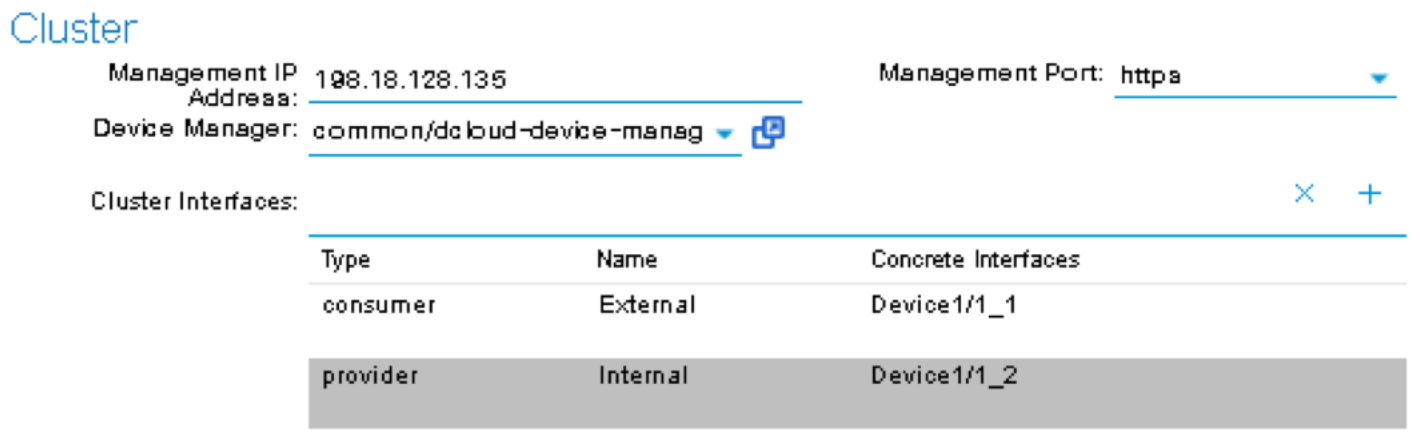
Click the + to add the 2nd Logical Interface:

Type: `provider`

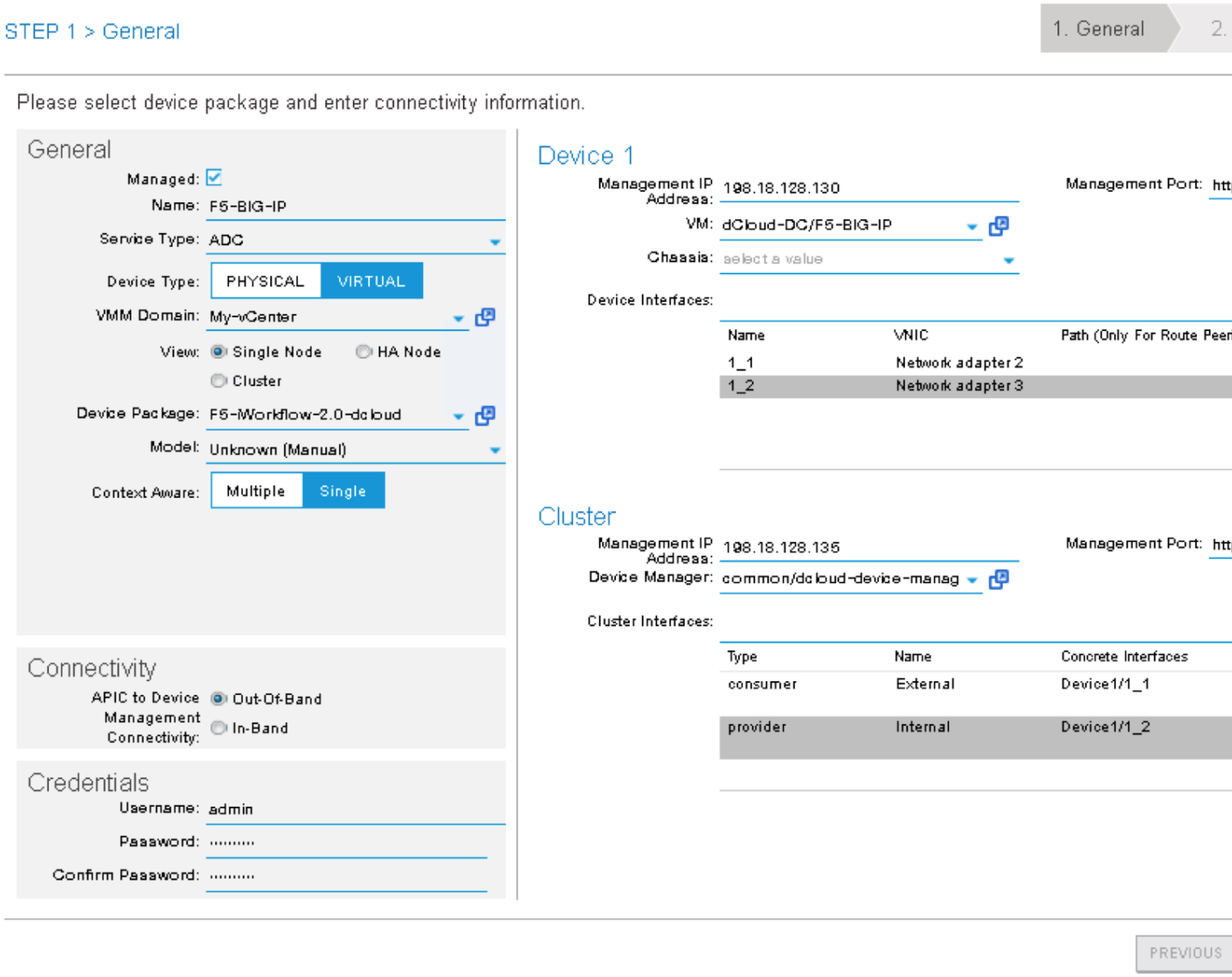
Name: `Internal`

Concrete Interface: `Device1/1_2`

Click UPDATE to accept the consumer interface configuration.



Make sure all L4-L7 Devices parameters are entered correctly, click "NEXT"

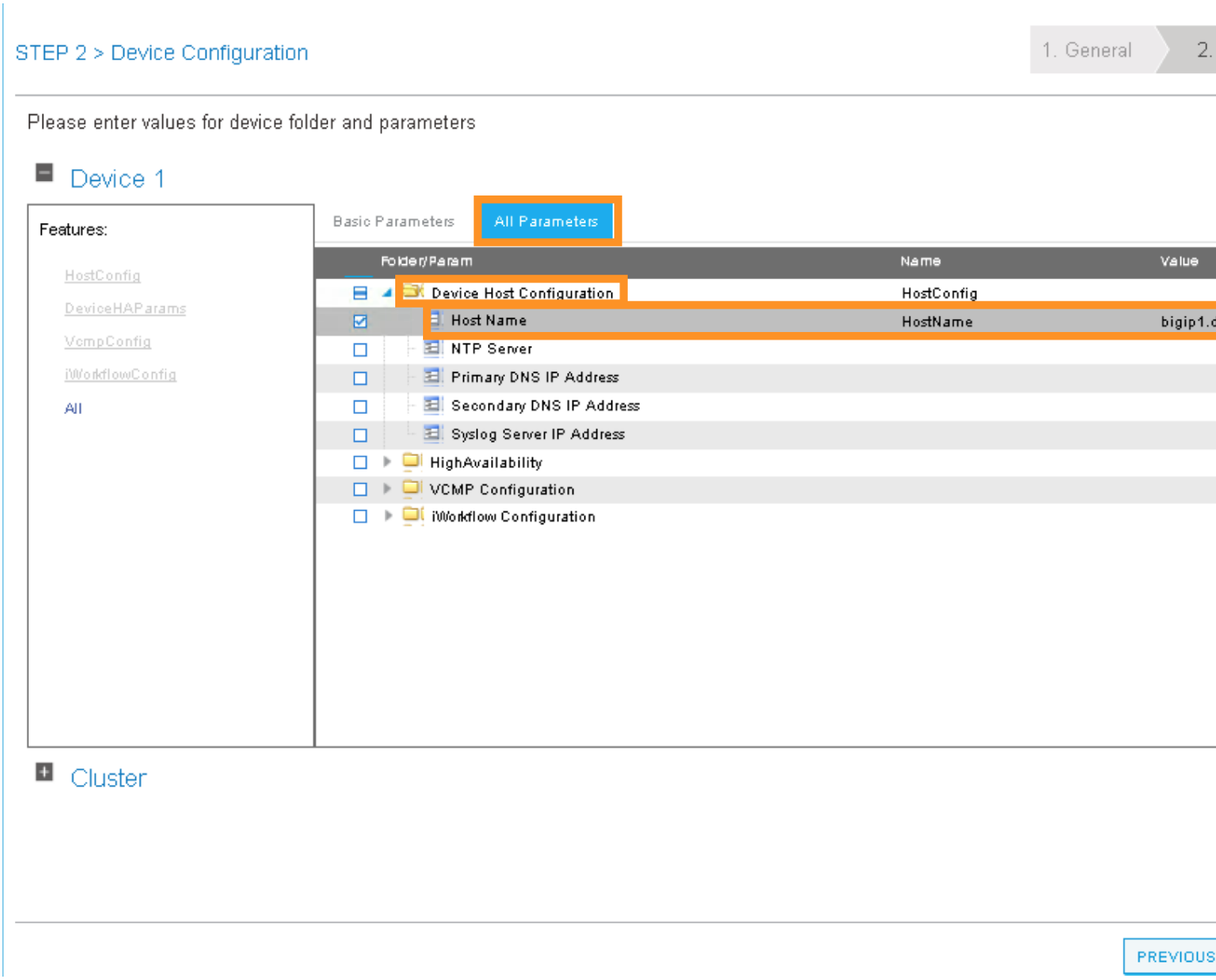


STEP2, Device Configuration. We would like to set up some basic information on the BIG-IP by choosing the All Parameters tab.

Click > to expand the field Device Host Configuration and enter the following parameters and click UPDATE to save the change:

Host Name: `bigip1.dcloud.cisco.com`

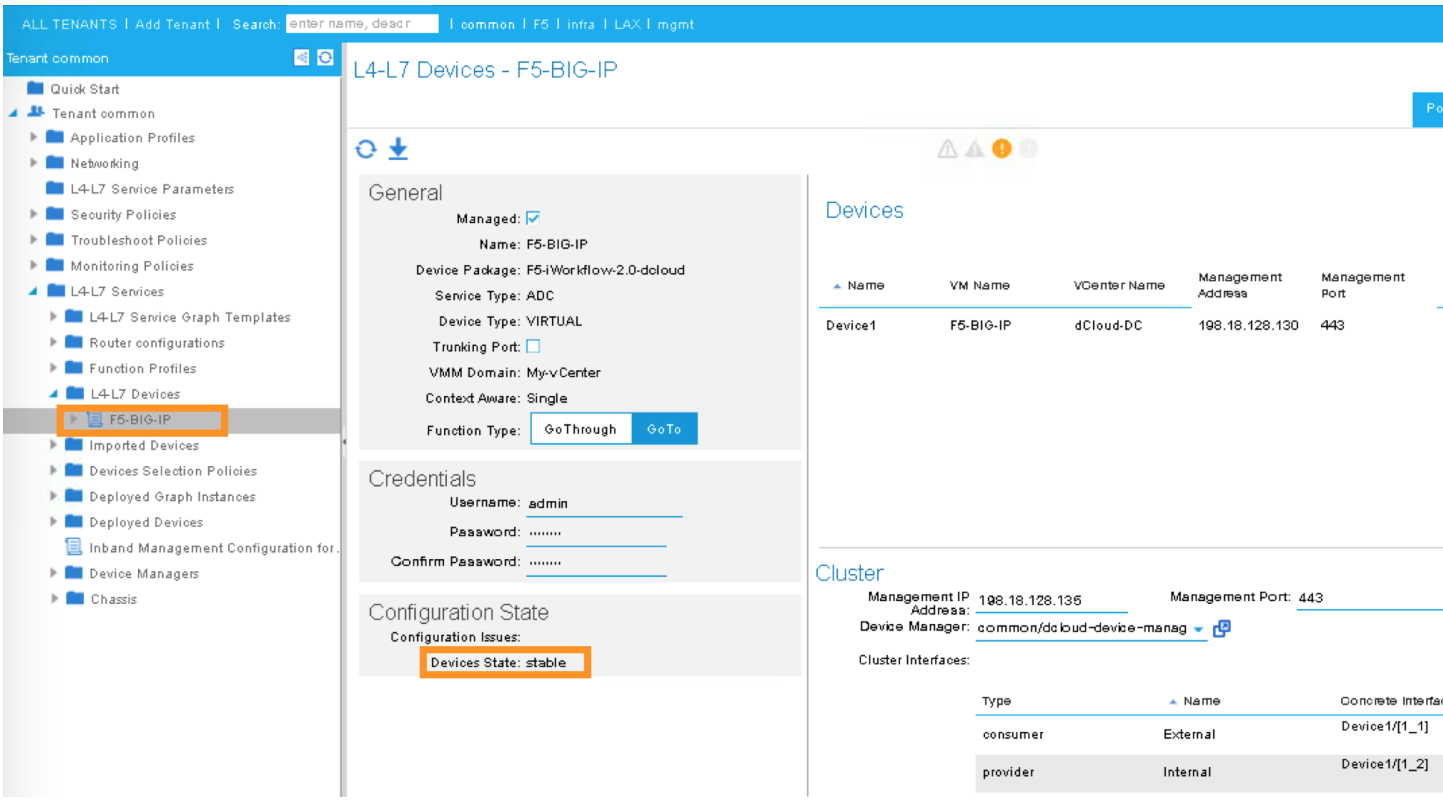
Click “FINISH”



Navigate to the newly created L4-L7 Device to verify its Configuration State is stable:

Tenants common ->L4-L7 Services -> L4-L7 Devices -> F5-BIG-IP

In the Work pane, ensure the Configuration State is stable, if the device is not stable, click the Faults tab and ensure no faults or all the faults are in clearing state.

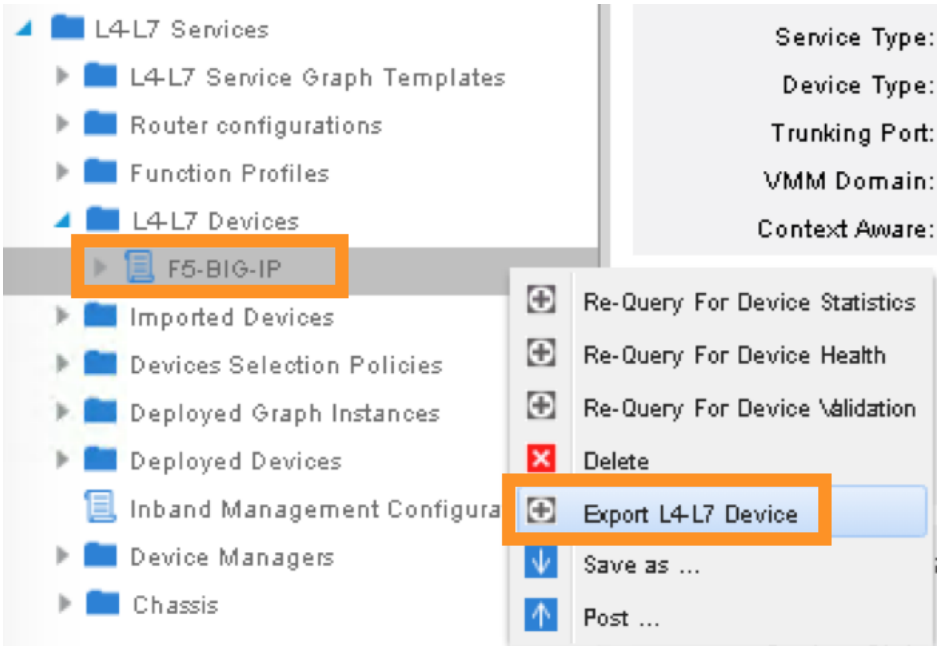


We now complete the configuration of the ACI L4-L7 device, and we will use this device when creating L4-L7 Service Graph Template in the next step.

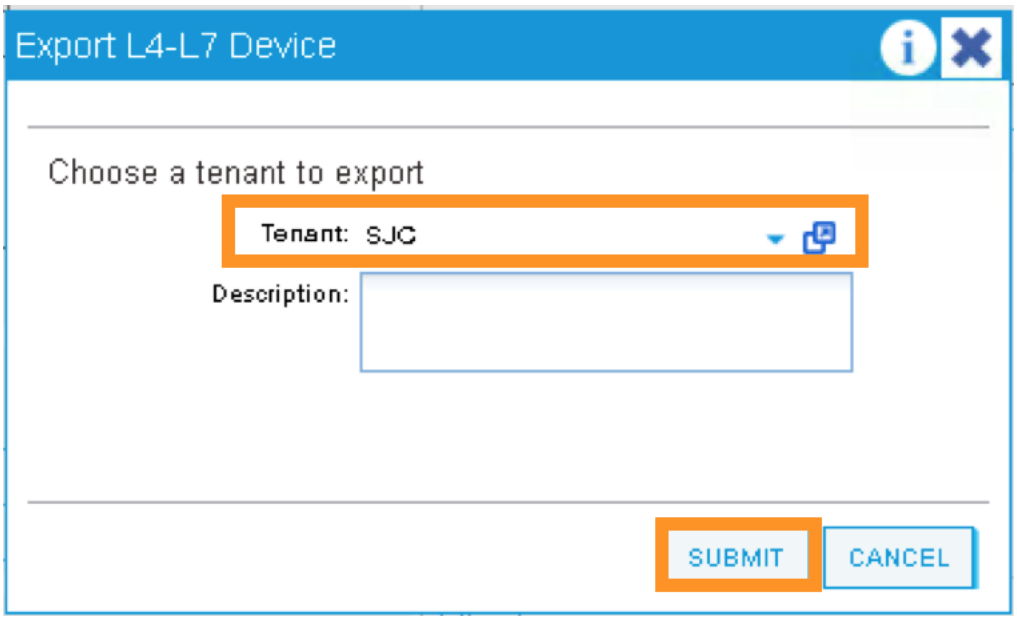
2.2.13 APIC – Export L4-L7 Device to Tenant

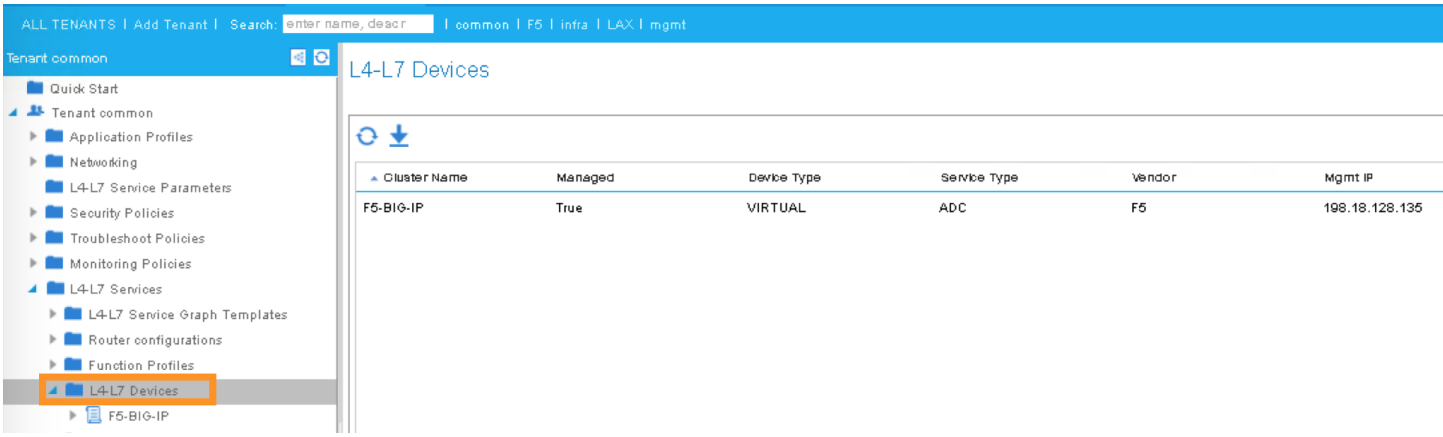
Export F5-BIG-IP L4-L7 device as a resource to another tenant where application profile is configured.

Right click on F5-BIG-IP, and select “Export L4-L7 Device”



Drop down and select tenant “SJC”, the “SUBMIT”

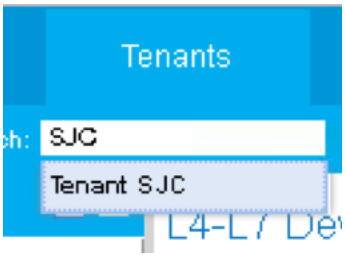




2.2.14 APIC – Create L4-L7 Service Graph Templates

An APIC L4-L7 Service Graph Template is an abstract object allowing L4-L7 configuration build into ACI policy model. In this step, you will create a service graph template and add L4-L7 device you created in the previous step, then select the WEB service function for this graph.

Go to Tenant SJC by typing “SJC” in the Tenant search box

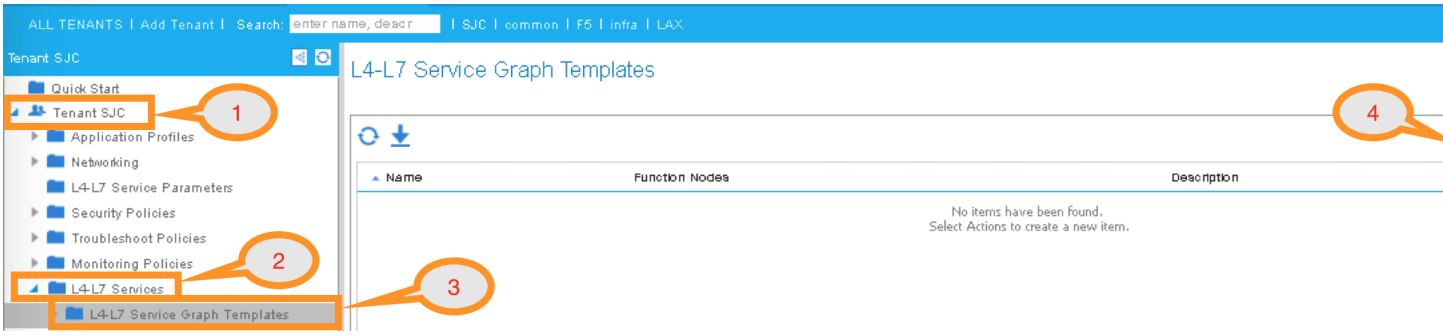


To create a new Service Graph Template, click the following in the navigation pane:

Tenants SJC -> L4-L7 Services -> L4-L7 Service Graph Template

In the Work pane:

ACTIONS -> Create L4-L7 Service Graph Template



In the new window, enter the following:

Graph Name: WEB

Graph Type: Create a New One (should be the default)

Now, drag the Device Clusters to the right side of the window into the graph. You should be able to place the Node “SJC/F5-BIG-IP (Imported Managed)” between the Consumer EPG and the Provider EPG.

When this graph template is deployed, the traffic will be redirected to the F5 BIG-IP of this device cluster automatically by Cisco ACI.

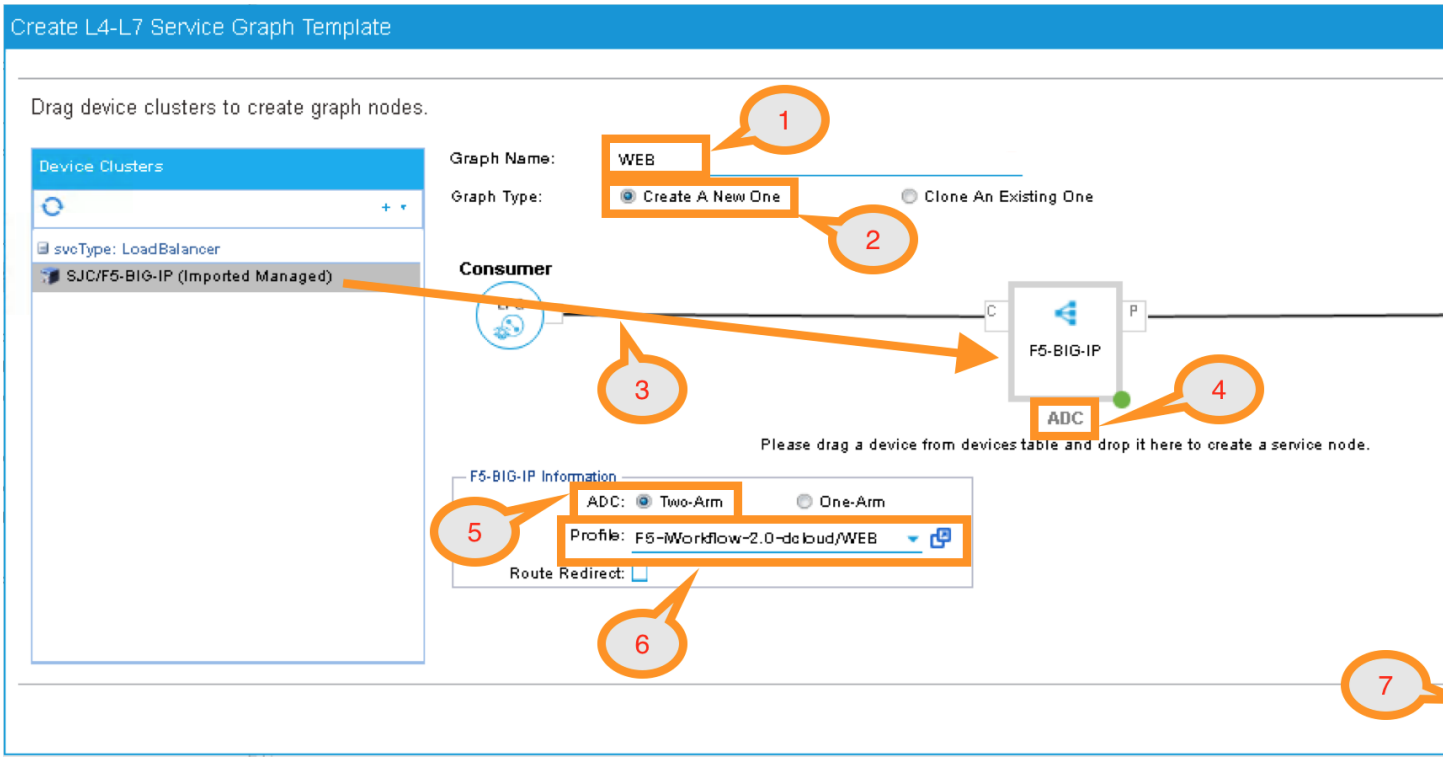
Double click the word N1 under the Node to change the name to ADC.

Under F5-BIG-IP Information, click the Two-Arm option for this graph.

Select the Profile: F5-iWorkflow-2.0–dcloud/WEB <- this coming from the F5 device package

This is WEB application template that we created earlier.

Click “SUBMIT”



2.2.15 APIC – Deploy the Service Graph (EPG and Contract selection)

The new ADC L4-L7 Service Graph Template is now created and we are ready to deploy the BIG-IP with the pre-created web and app EPG.

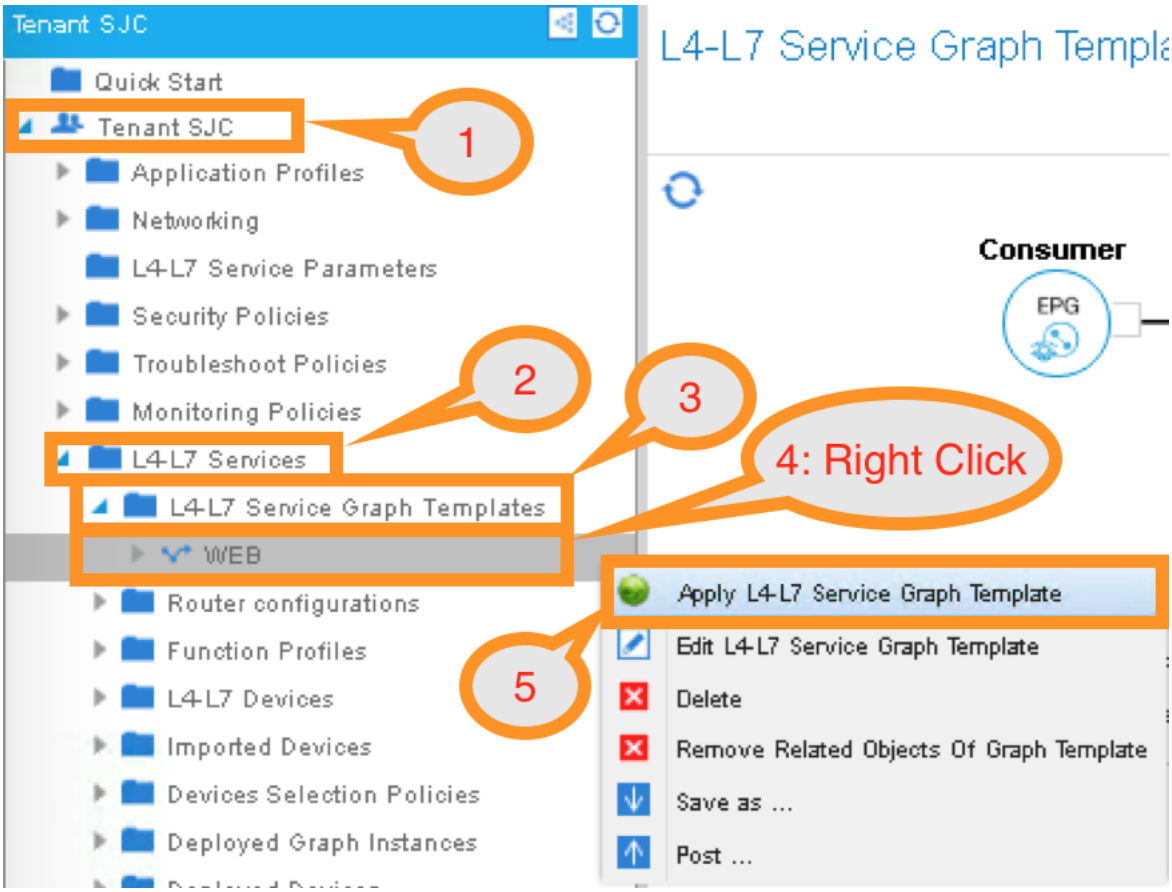
In this step, we are deploying WEB graph, connecting between the web tier and the app tier. Inside contract between the web and app EPG, we will assign the service graph template created in the previous step, this will provide F5 BIG-IP ADC functionality to APP tier.

To deploy the service graph, click the following in the Navigation pane of your tenant:

Tenants SJC -> L4-L7 Services -> L4-L7 Service Graph Template

Select the Service Graph Template you just created from the Work pane. Right click and choose the option to

Apply L4-L7 Service Graph Template



In the new window, you will have the ability to choose which EPGs the Service Graph will be inserted in between.

Select the following for the EPG information:

Consumer EPG / External Network: `SJC/App1/epg-web`

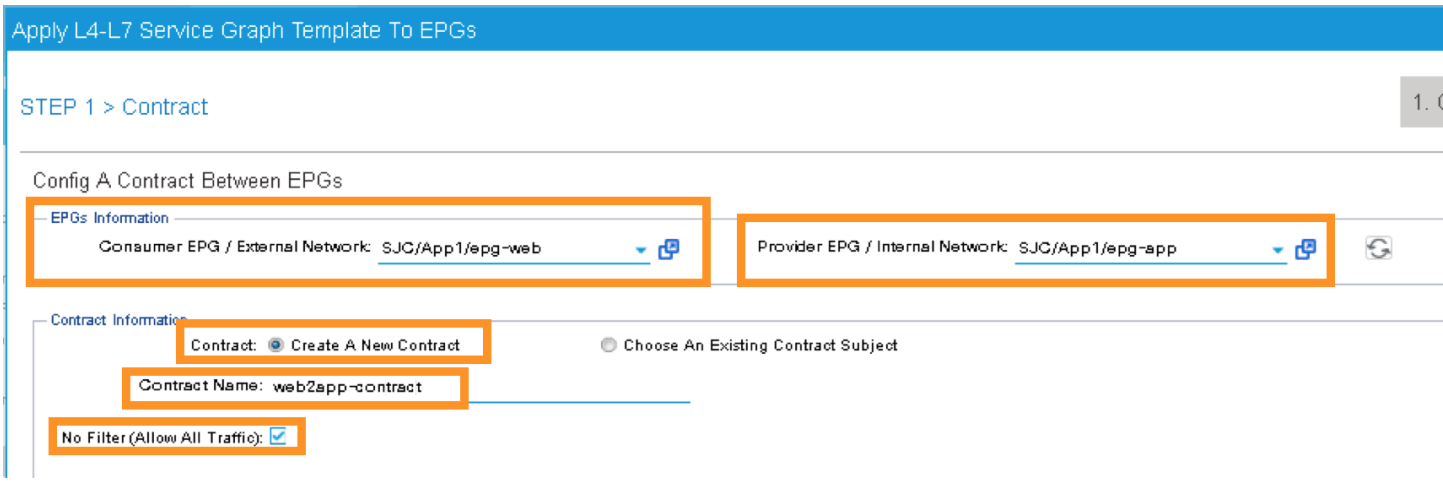
Provider EPG / External Network: `SJC/App1/epg-app`

Under Contract Information, use the option to create a new Contract:

Create a New Contract: `SELECTED`

Contract Name: `web2app-contract`

No Filter (Allow All Traffic): `CHECKED`



Click NEXT to continue to the next screen.

2.2.16 APIC – Deploy the Service Graph (Connectivity to Fabric)

A new window to apply the service graph template will now appear. This window will show the Service Graph Template that you created earlier.

In addition to the Service Graph Template, there are some options that need to be selected to deploy the BIG-IP with a Service Graph. Under the SJC/WEB Information, you need to choose the appropriate connector information:

Under the Connector, choose the following:

Type: `General`

BD: `SJC/SJCBDWeb`

Cluster Interface: `External`

We use the External interface for the communication between the BIG-IP and the Web servers. The Web servers belong to Web EPG, which tied to the SJCBDWeb Bridge Domain.

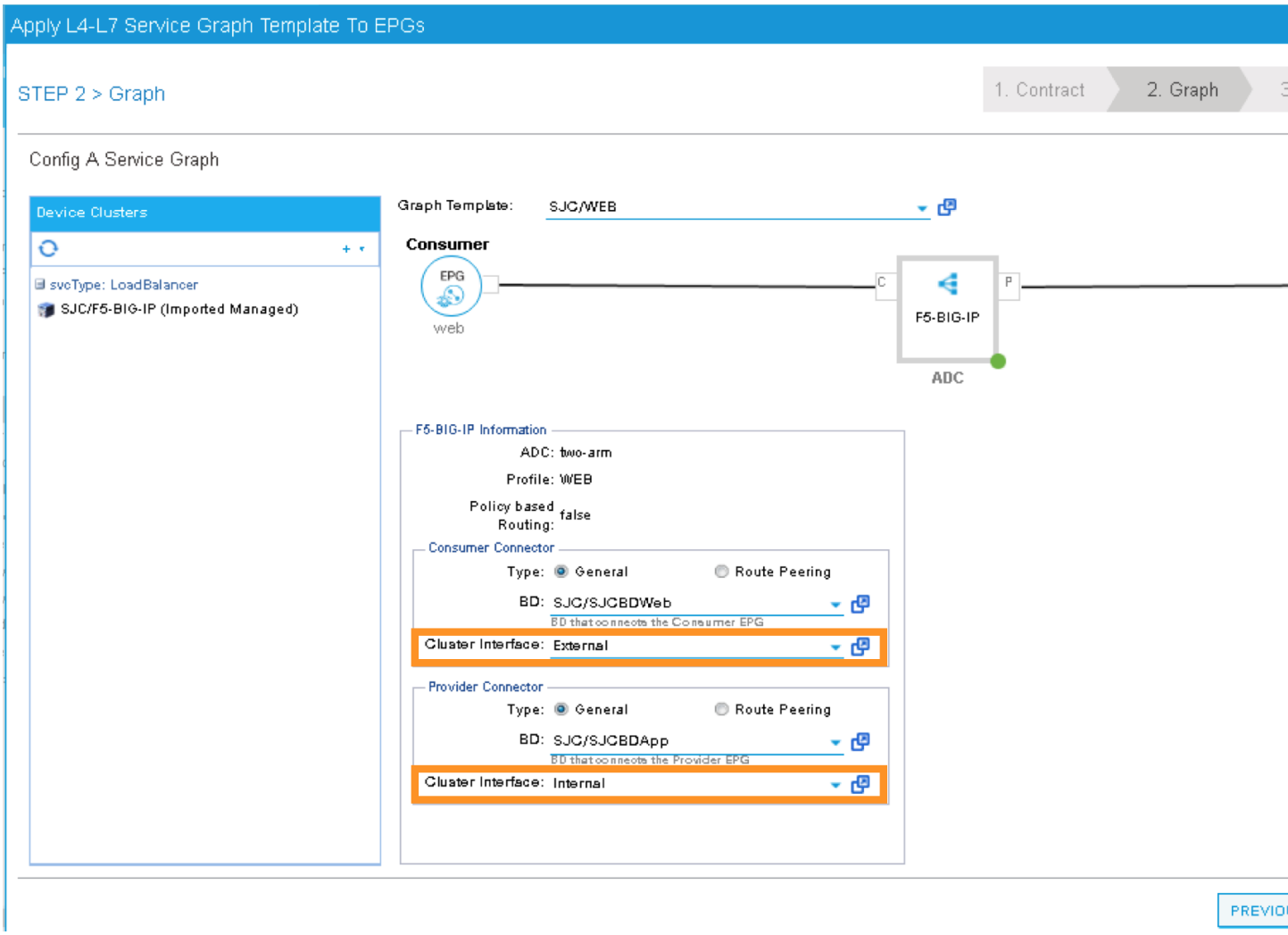
Type: `General`

BD: `SJC/SJCBDApp`

Cluster Interface: internal

We use the Internal interface for the communication between the BIG-IP and the App servers. The App servers belong to App EPG, which tied to the SJCBDApp Bridge Domain.

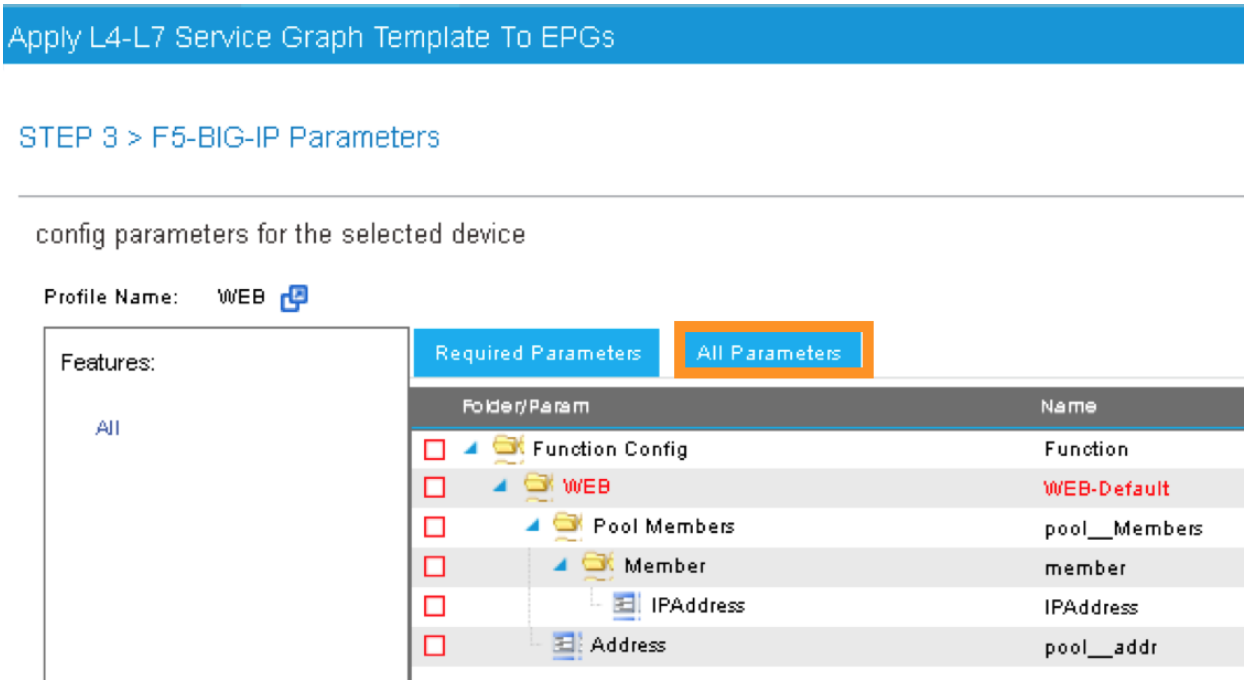
Click NEXT to continue to the next screen.



2.2.17 APIC – Deploy the Service Graph (BIG-IP Parameters)

A new window for the BIG-IP parameters will now appear. In this window, you will have the ability to modify the parameters to be deployed to the BIG-IP. Let us modify some parameters to push the Service Graph into the BIG-IP.

Under Feature, it should be selected All. Parameters should be All Parameters.



Once you click the All Parameters tab, the folder and parameters will appear. To edit the parameter, you need to expand the parameter by clicking the > and double the field to change the parameter's name and value. Let us edit the following parameters:

Under Device Config

Press > to expand the Network configuration folder

Press > to expand the folder ExternalSelfIP

Double click the parameter Enable Floating? and select No as the value

Click UPDATE to apply

Double click the parameter External Self IP Address and enter 10.10.10.130 as the value

Click UPDATE to apply

Double click the parameter External Self IP Netmask and enter 255.255.255.0 as the value

Click UPDATE to apply

Double click the parameter Port Lockdown and select Default as the value

Click UPDATE to apply

Press > to expand the folder InternalSelfIP

Double click the parameter Enable Floating? and select No as the value

Click UPDATE to apply

Double click the parameter Internal Self IP Address and enter 192.168.10.130 as the value

Click UPDATE to apply

Double click the parameter Internal Self IP Netmask and enter 255.255.255.0 as the value

Click UPDATE to apply

Double click the parameter Port Lockdown and select Default as the value

Click UPDATE to apply

Required Parameters | All Parameters

Folder/Param	Name	Value
Device Config	Device	
Network	Network	
ExternalSelfIP	ExternalSelfIP	
Enable Floating?	Floating	NO
External Self IP	SelfIPAddress	10.10.10.130
Port Lockdown	PortLockdown	DEFAULT
Self IP Netmask	SelfIPNetmask	255.255.255.0
InternalSelfIP	InternalSelfIP	
Enable Floating?	Floating	NO
Internal Self IP Address	SelfIPAddress	192.168.10.130
Internal Self IP Netmask	SelfIPNetmask	255.255.255.0
Port Lockdown	PortLockdown	DEFAULT

Device config is BIG-IP device level configuration, like self-IP and default route. Resource configured in the device config will be used by Function Config

Assign Device Config “Network” to Function Config “NetworkRelation”

Note: It is extremely important to assign Network to NetworkRelation, fail to perform this step will result in graph deployment failure, as there will not be any network resource associated with the graph

Function Config	Function	
NetworkRelation	NetworkRelation	
Select Network	NetworkRel	Network

The above step associates the network information under device config to the BIG-IP virtual server.

Apply at deployment WEB service graph configuration under Function Config

Press > to expand the WEB configuration folder

Double click on the name and delete Default

Click UPDATE to apply

Press > to expand the Pool Members folder

Press > to expand the Member folder

Double click to enter value into the IPAddress field: `192.168.10.150`

Click UPDATE to apply

Back to the WEB configuration folder

Double click to enter value into the Address field (`pool__addr`): `10.10.10.100`

Click UPDATE to apply

Double click the parameter Port field (`pool__port`): `80`

Click UPDATE to apply

WEB	WEB	
Pool Members	pool__Members	
Member	member	
IPAddress	IPAddress	192.168.10.150
Address	pool__addr	10.10.10.100
Port	pool__port	80

Function config is BIG-IP virtual server level configuration. We define the WEB service catalog parameters here, as well as associating the device level network config to this virtual server.

Make sure both the device config and function config are correct

Device Config

Device Config	Device	
Network	Network	
ExternalSelfIP	ExternalSelfIP	
Enable Floating?	Floating	NO
External Self IP	SelfIPAddress	10.10.10.130
Port Lockdown	PortLockdown	DEFAULT
Self IP Netmask	SelfIPNetmask	255.255.255.0
InternalSelfIP	InternalSelfIP	
Enable Floating?	Floating	NO
Internal Self IP Address	SelfIPAddress	192.168.10.130
Internal Self IP Netmask	SelfIPNetmask	255.255.255.0
Port Lockdown	PortLockdown	DEFAULT
Route		
SNAT Pool		

Function Config

Function Config	Function	
NetworkRelation	NetworkRelation	
Select Network	NetworkRel	Network
WEB	WEB	
Pool Members	pool__Members	
Member	member	
IPAddress	IPAddress	192.168.10.150
Address	pool__addr	10.10.10.100
Port	pool__port	80

Click “FINISH” to deploy the graph

Apply L4-L7 Service Graph Template To EPGs

STEP 3 > F5-BIG-IP Parameters

1. Contract 2. Graph

config parameters for the selected device

Profile Name: WEB

Features:

All

Required Parameters All Parameters

Folder/Param	Name	Value	Apply To Specific Device
Device Config	Device		
Network	Network		
Function Config	Function		
NetworkRelation	NetworkRelation		
Select Network	NetworkRel	Network	
WEB	WEB		
Pool Members	pool__Members		
Member	member		
IPAddress	IPAddress	192.168.10.150	
Address	pool__addr	10.10.10.100	
Port	pool__port	80	

RED indicators parameters needed to be updated and GREEN indicates parameters will be summitted to the provider EPG.

PREVIO

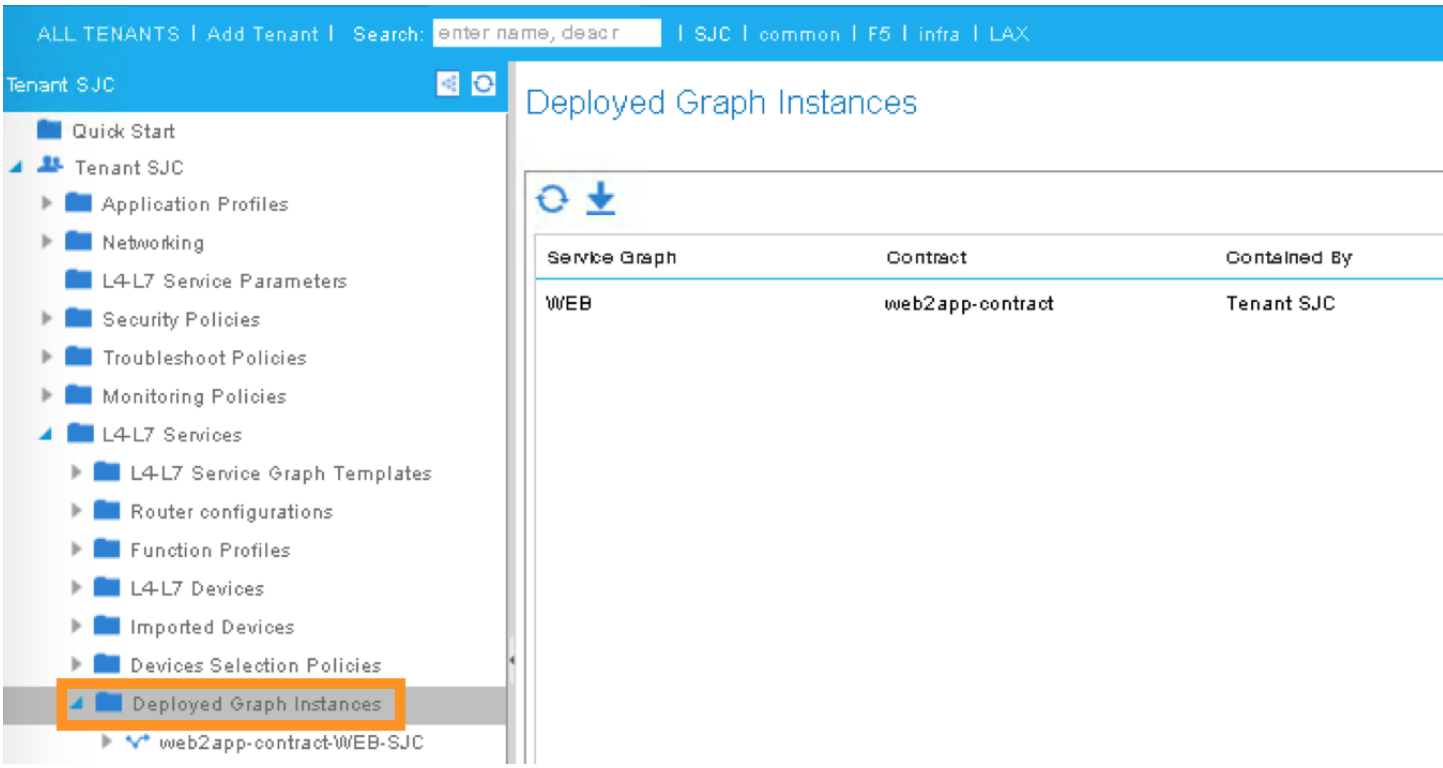
2.2.18 APIC – Verifying WEB application deployment

APIC: Verifying the service graph deployment

You can now verify if APIC has deployed the service graph correctly. First, navigate the following:

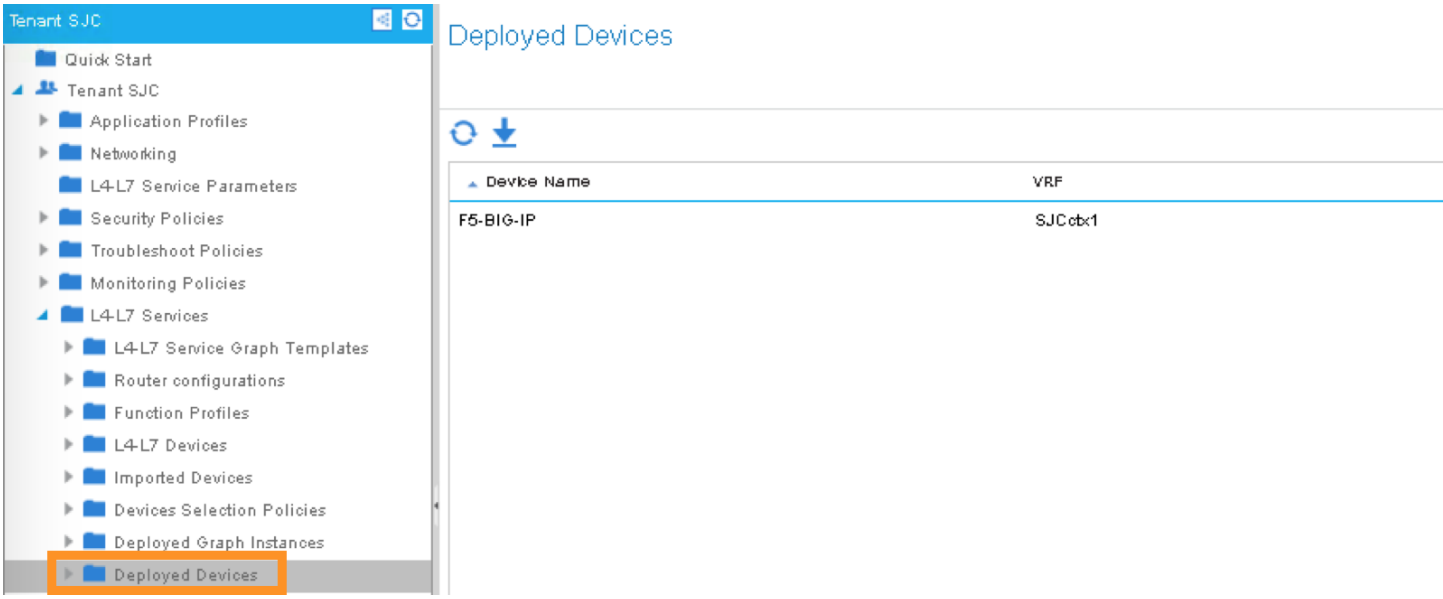
Tenant SJC -> L4-L7 Services -> Deployed Graph Instances

You should be able to see a screen similar to the following. The State should say “applied”



Tenant SJC -> L4-L7 Services -> Deployed Devices

You should be able to see a screen similar to the following. The State should say “allocated”



Make sure there is no faults to the deployment:

2.2.19 iWorkflow – Verifying the template deployment

Once the service graph is deployed in Cisco APIC, administrator can also view application status in F5 iWorkflow.

Log into the F5 iWorkflow 198.18.128.135 with the following username and password from the web browser (if the previous session has timed out):

iWorkflow: `https://198.18.128.135`

Username: `admin`

Password: `C1sco12345`

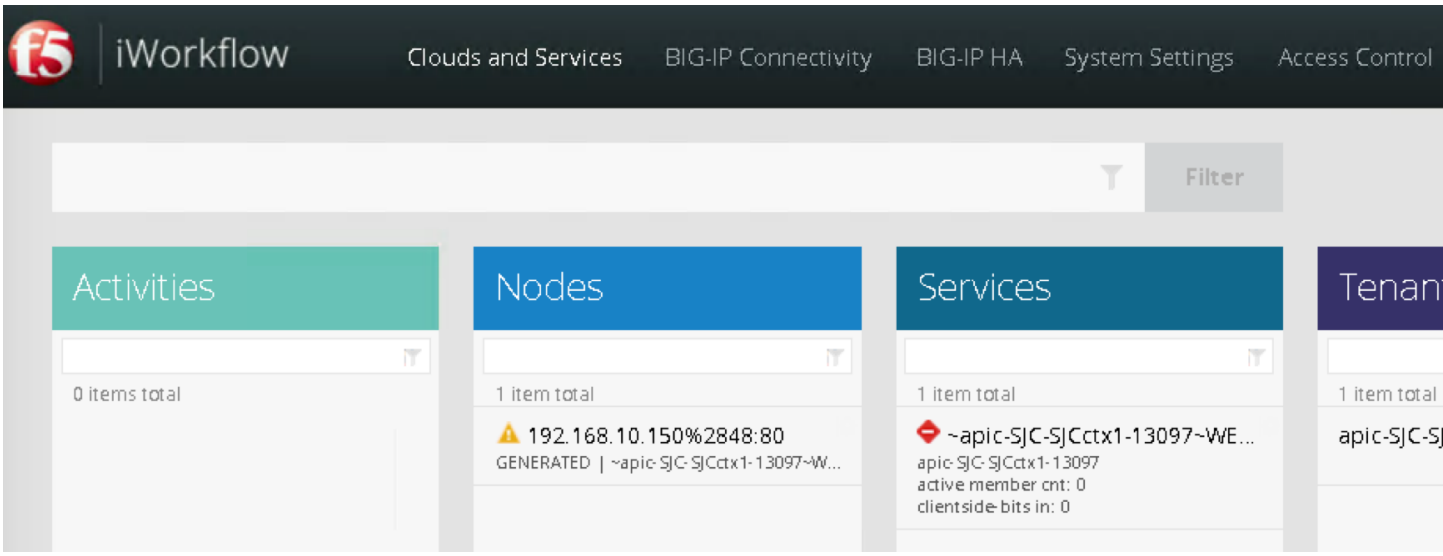
Under the iWorkflow Cloud and Services. In the Work pane, under:

Services: graph deployment status

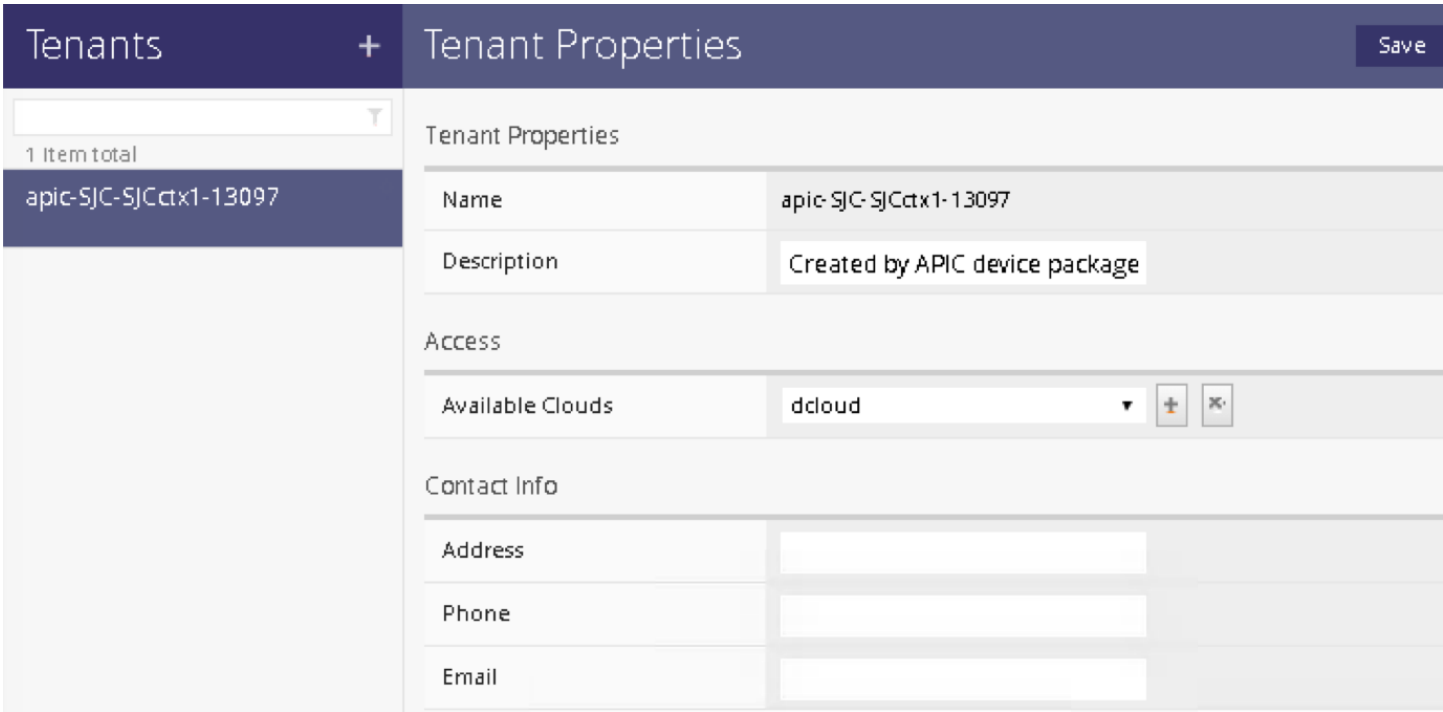
Tenant: APIC tenant information

Nodes: pool members information

Notice the graph is “unhealthy” because no servers are available to the BIG-IP virtual server. This is expected because dCloud only validate control plane, as a result, BIG-IP data plane validation to the servers failed.

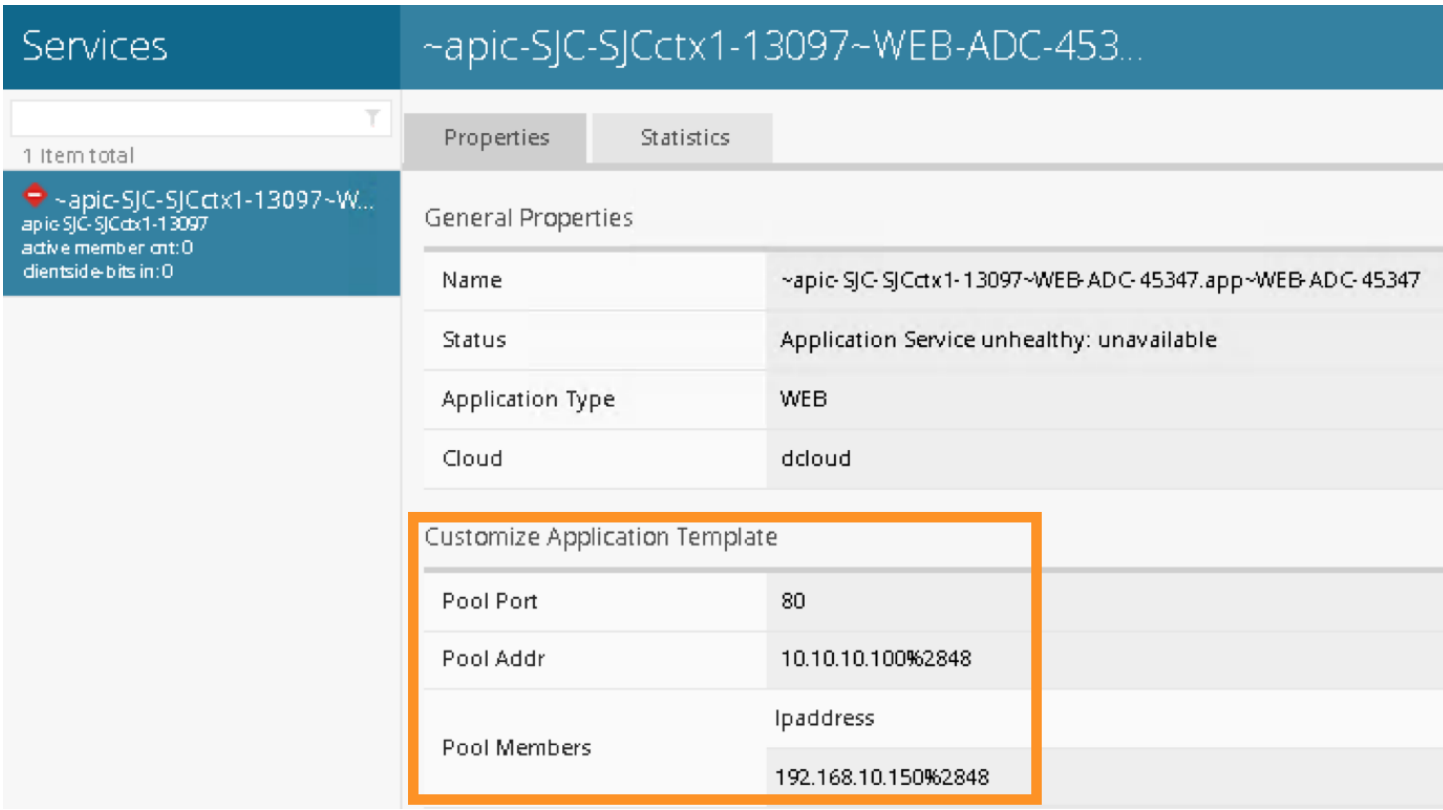


Tenants:



Services

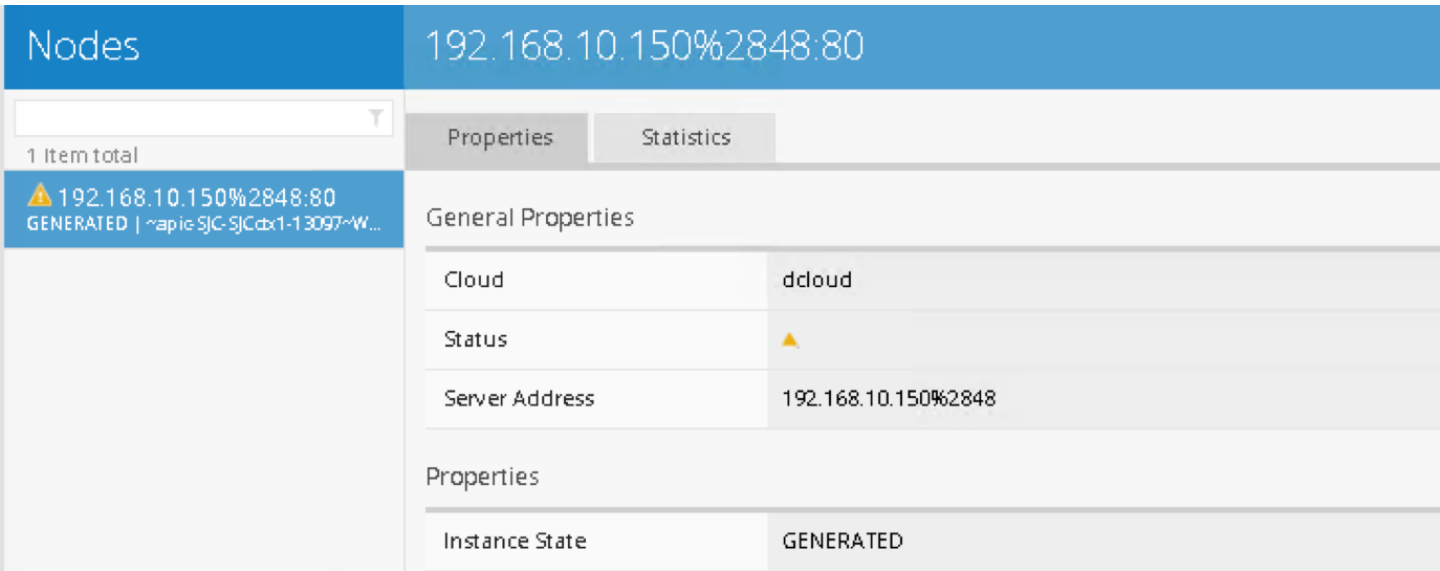
Notices “Customize Application Template” contains the fields visible in APIC. User input the values from APIC.



In case Customize Application Template is empty, please check back in a few minutes until the resource is refresh

Nodes:

This the member IP entered through APIC.



2.2.20 BIG-IP – Verifying Application Services (Virtual Server) deployment

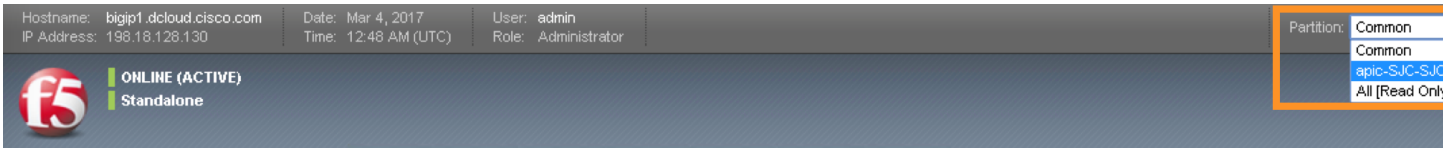
Log into the F5 BIG-IP `198.18.128.130` with the following username and password from the web browser (if the previous session has timed out):

BIG-IP: `https://198.18.128.130`

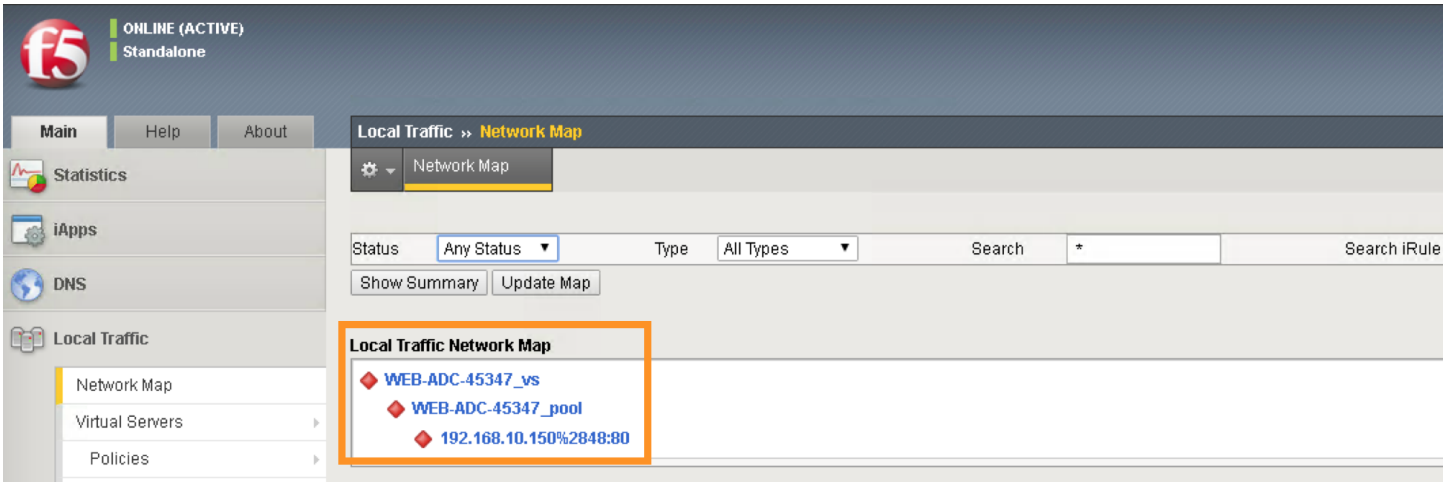
Username: `admin`

Password: `C1sco12345`

On the Main menu, click Local Traffic -> Network Map. Then on the top right corner, next to the Log out button, click the drop down to select the newly created Partition (please note that this reflects the APIC Virtual Device ID):

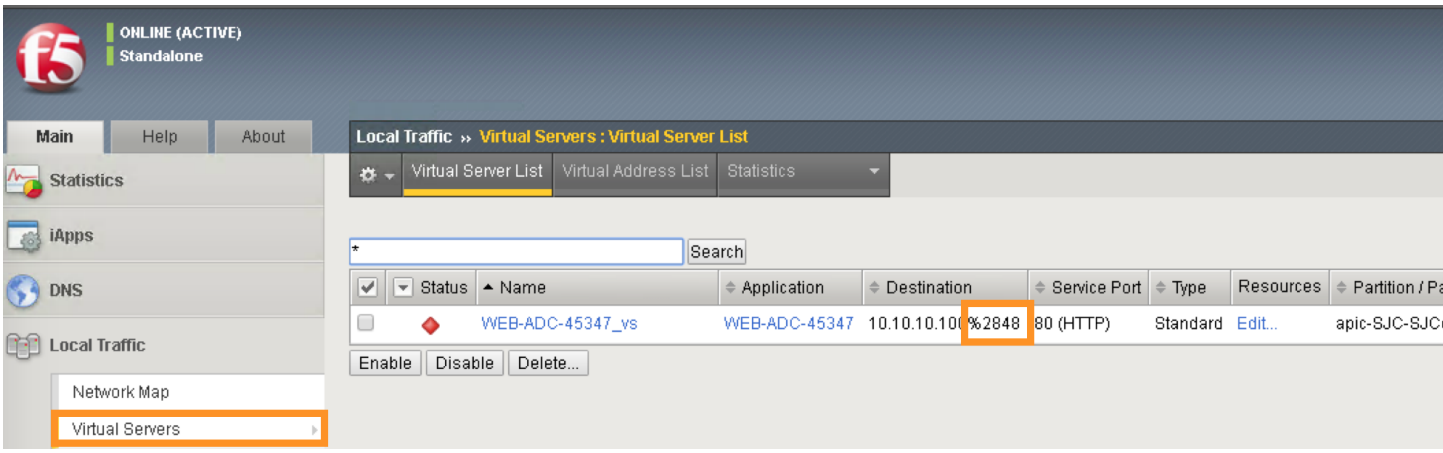


Once you are in the partition, click Local Traffic -> Network Map. You should be able to see the virtual server is configured along with its pool and pool members.

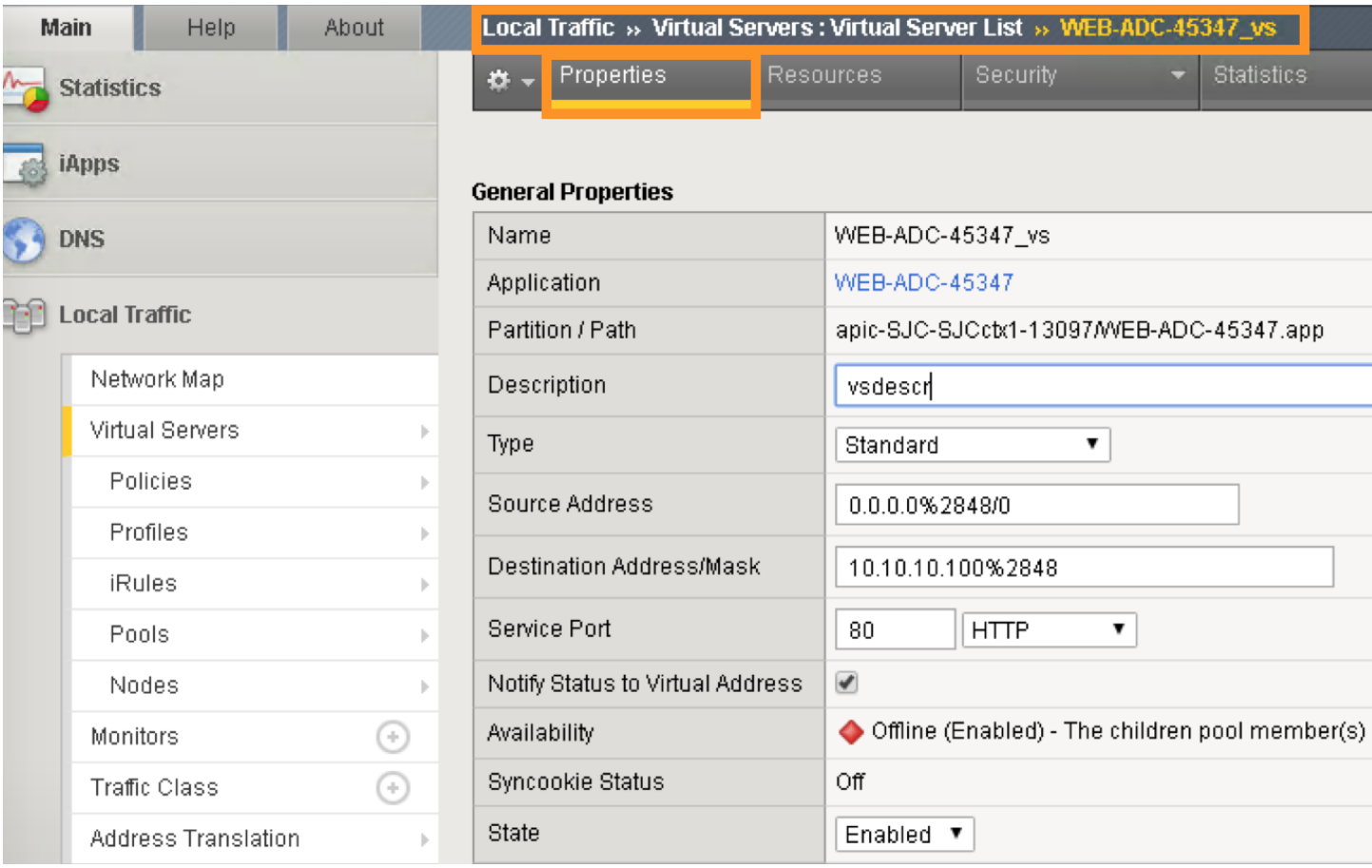


On the right Navigation menu, click the Local Traffic -> Virtual Servers and you should be able to see the brief Virtual IP information. You can see that the VIP is currently listening on HTTP port `80`.

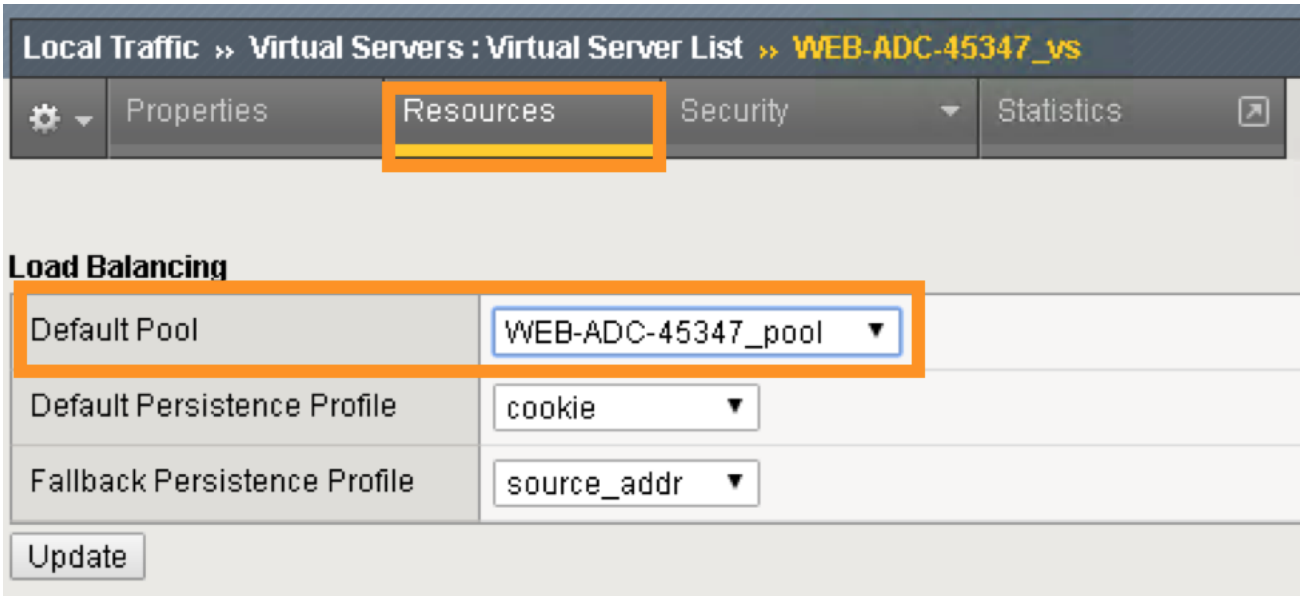
The number (in this example, 2848) after the % mark represents the route domain (RD) number. There will be a RD number assign to each APIC partition, which equivalent to an ACI L3 VRF. This allows BIG-IP to provide multi-tenancy support in ACI environment.



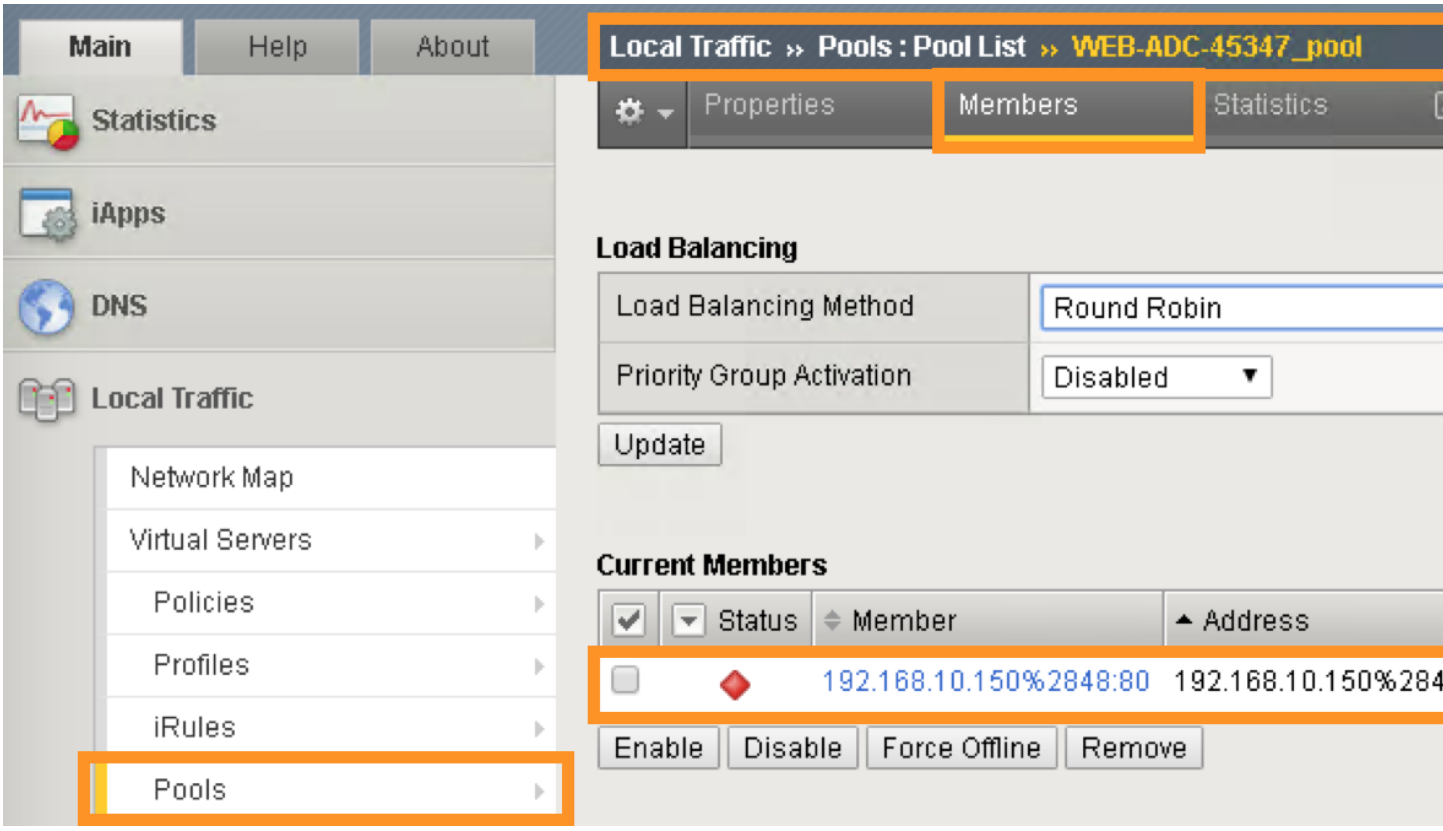
In the Virtual Server List, click the Name in the hyperlink and you will see the Property of the Virtual Server with more detailed information. The configured the parameters will appear here.



Click on “Resource”, notice the pool name being used



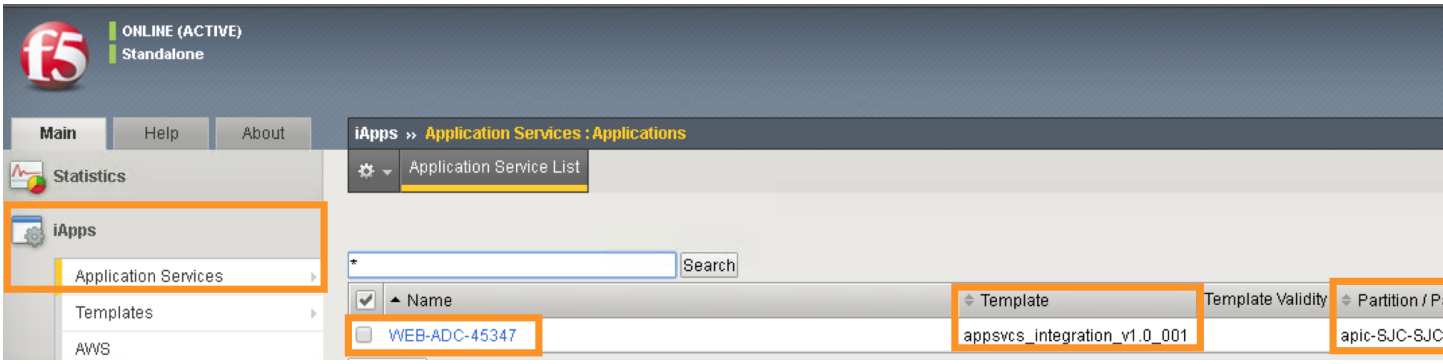
Click Local Traffic -> Pools and you should see the brief information of the real server pool information:



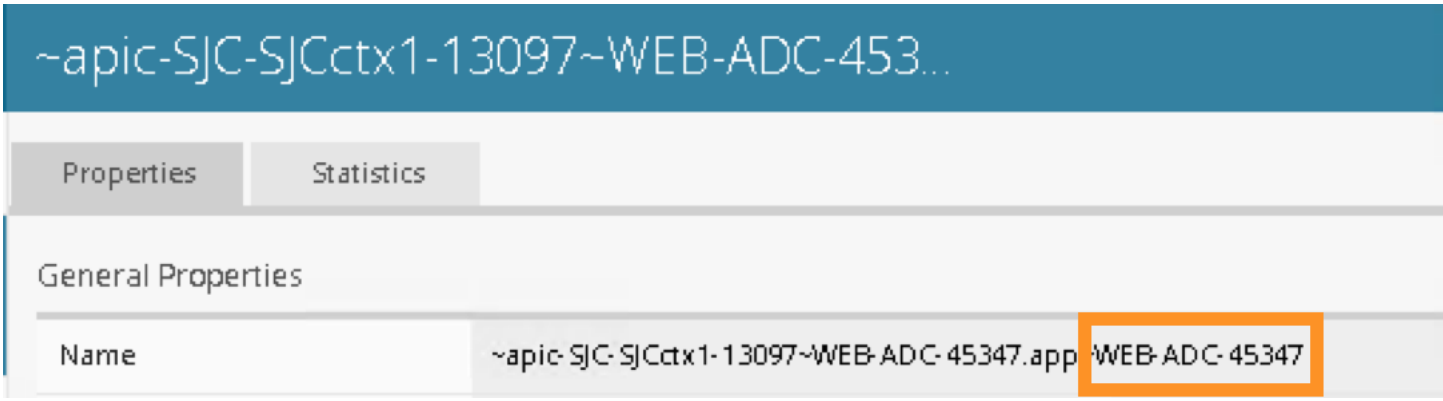
Go back to the Navigation pane and click the iApps -> Application Services. Notice the name of the Application Services is same as the Services name in iWorkflow.

Template is the iApps template that associated with this application service

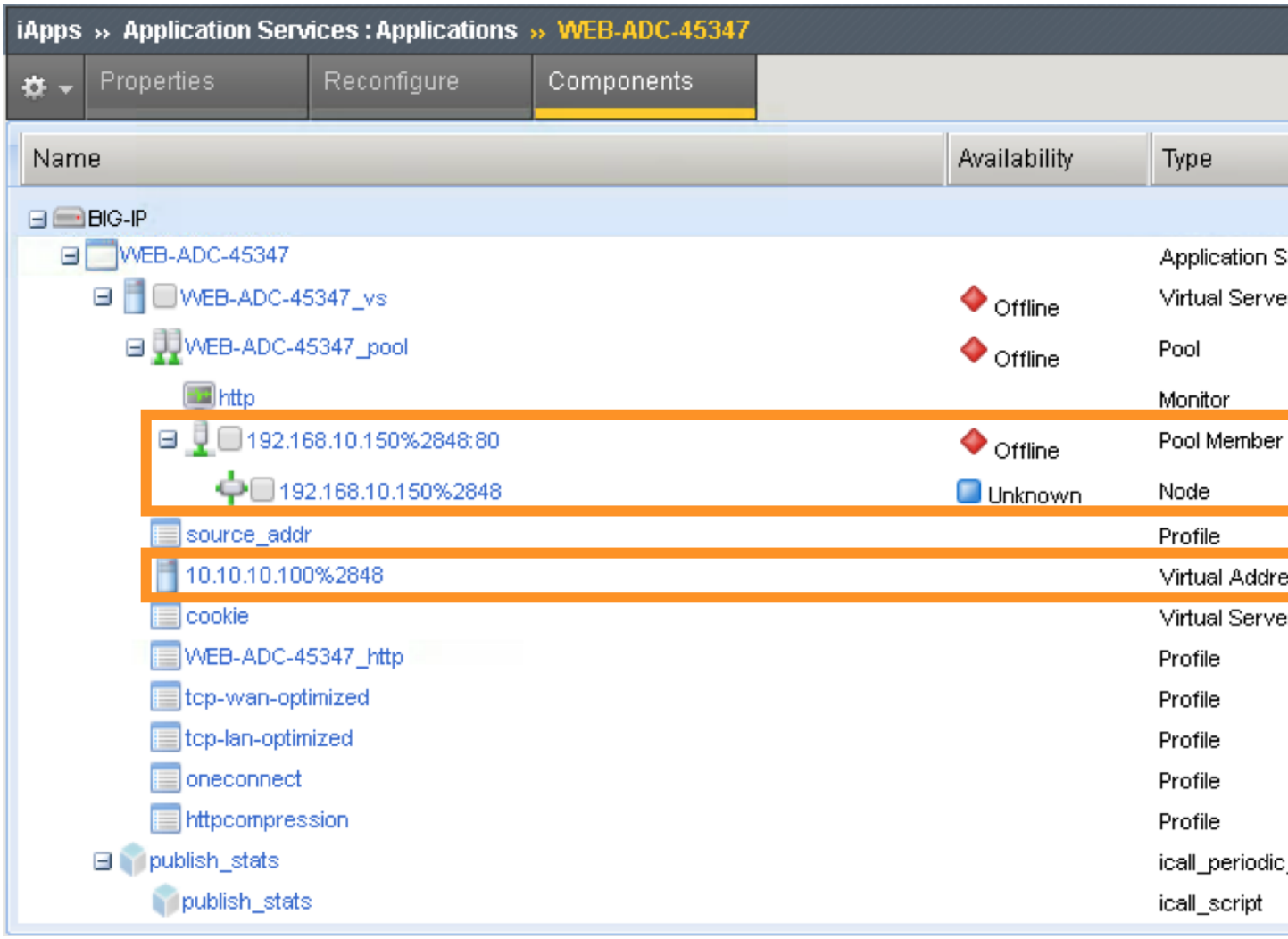
Partition/Path is the APIC created partition and the name of the application service



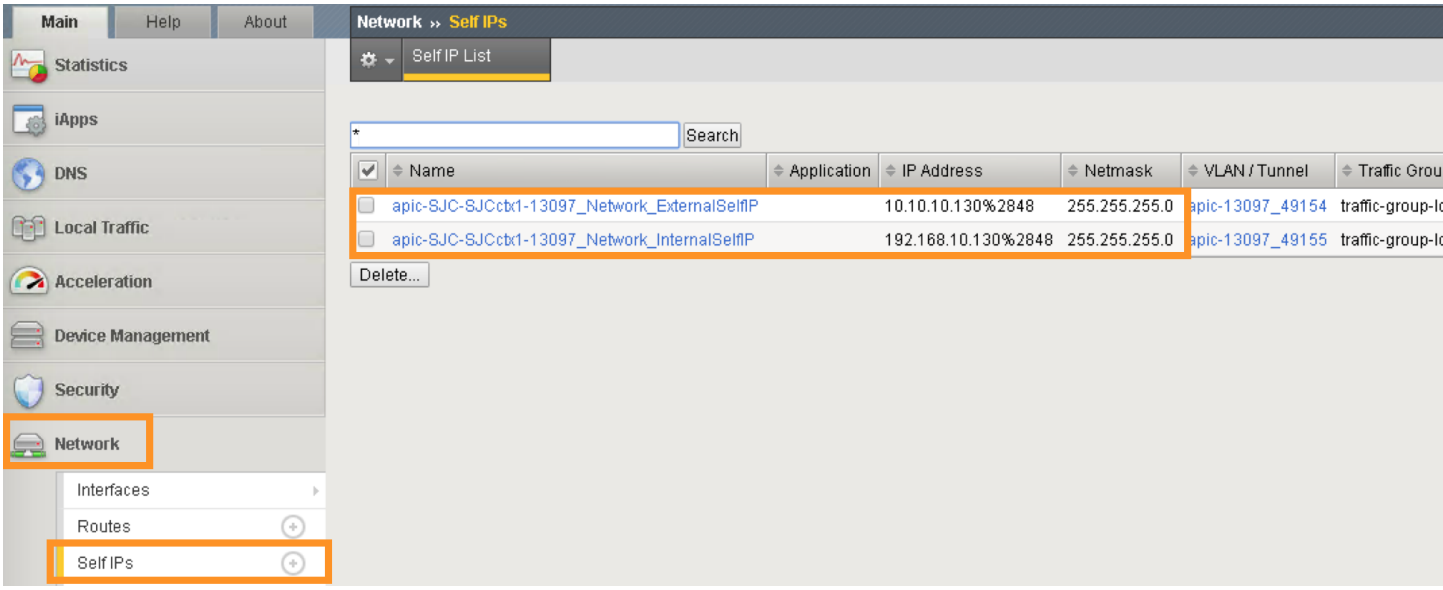
F5 iWorkflow service name



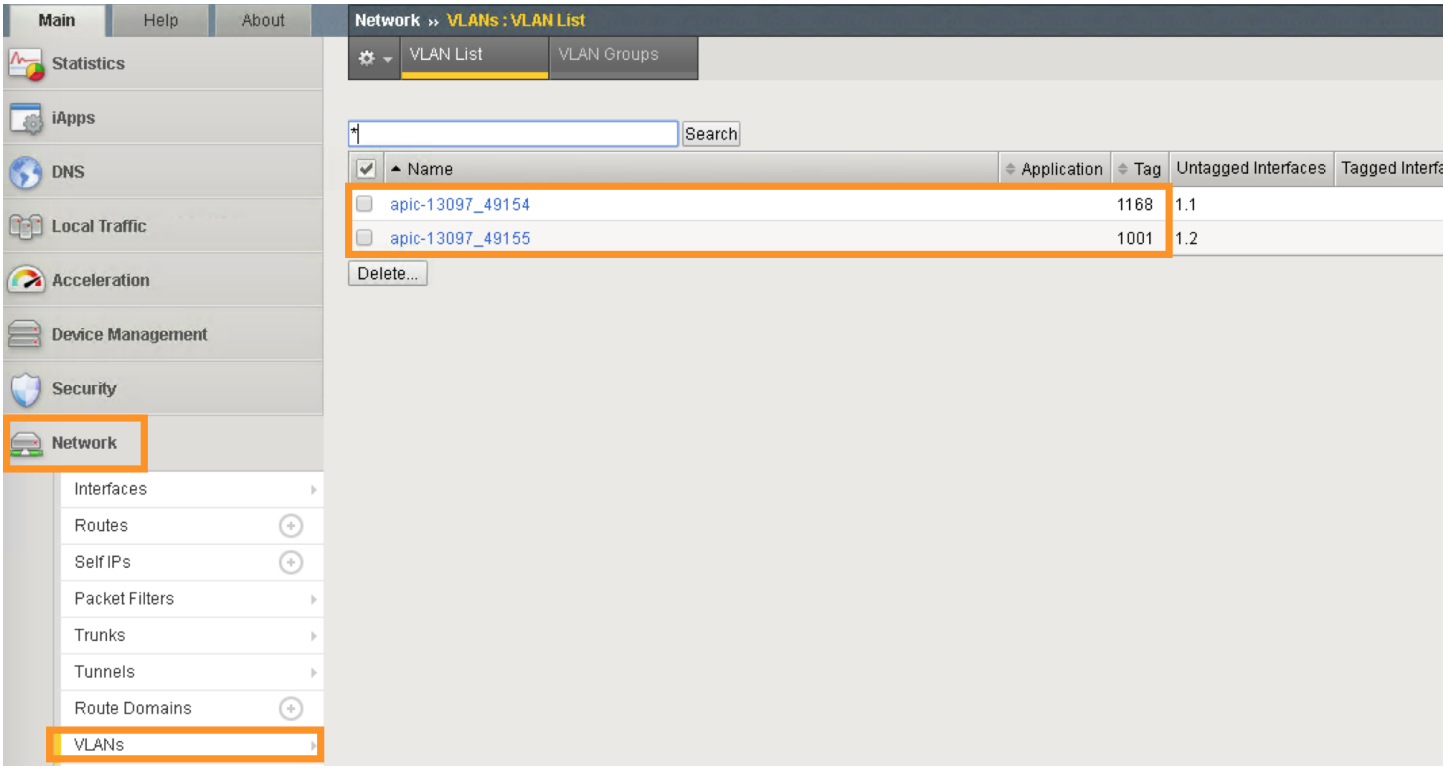
Click the application service name will direct to the Application Services Components. By using iApps template, you can configure a full features virtual server by specifying customized parameters exposed to APIC. Only the highlighted ones are entered by APIC, the rest of the virtual servers features are built inside the iApps template.



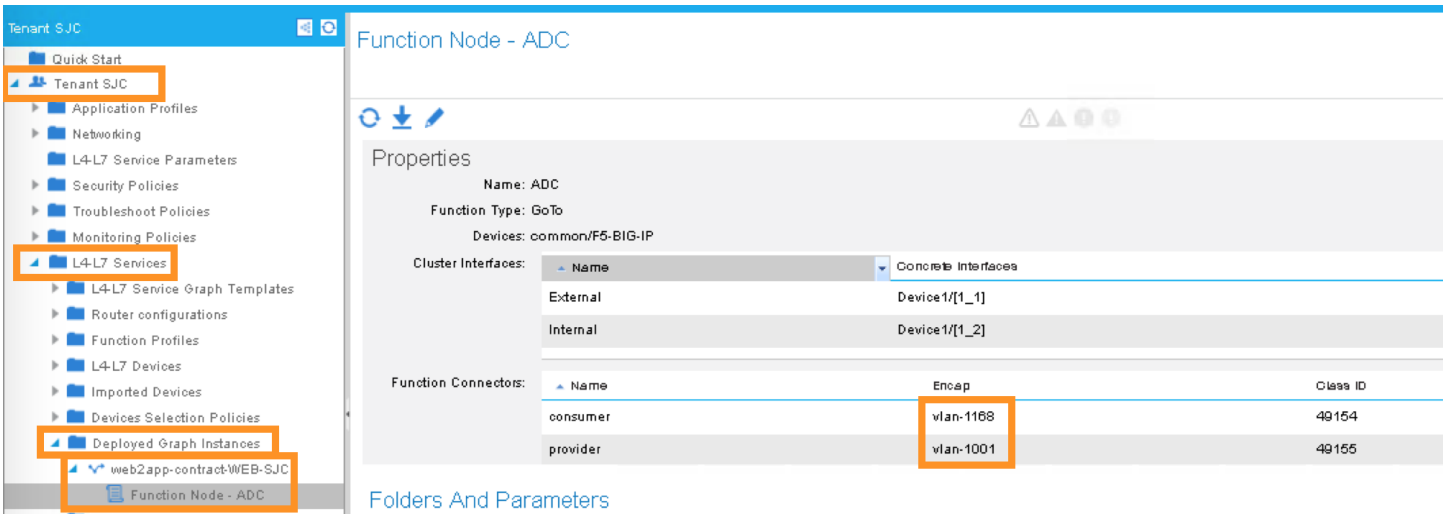
Network -> Self IP configuration from APIC



VLAN information imported from APIC:



Same VLAN tags are being assigned in APIC

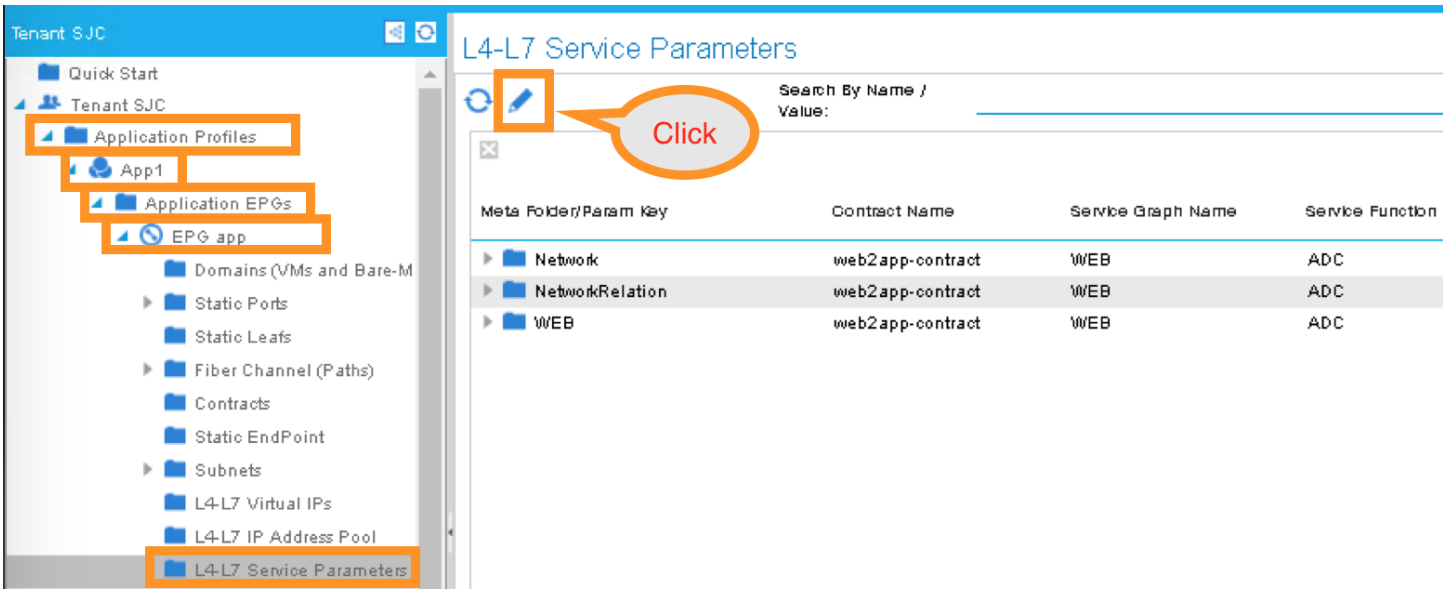


This concludes Scenario 1 "Deploy Service Graphs in Cisco ACI using F5 iWorkflow" lab.

2.3 Modify L4 – L7 deployed graph parameters

User can modify deployed graph parameters, only parameters mark "Tenant Editable" in iWorkflow can be changed in APIC. Once a graph is deployed, user need to go under Application Profiles / EPG level in order to make changes to deployed graph parameters. The deployed graph parameters reside under the provider EPG, in this case, it is the app EPG.

Go to APIC Tenant SJC -> Application Profiles -> App1 -> Application EPGs -> EPG app -> L4-L7 Service Parameters, click the pen button:



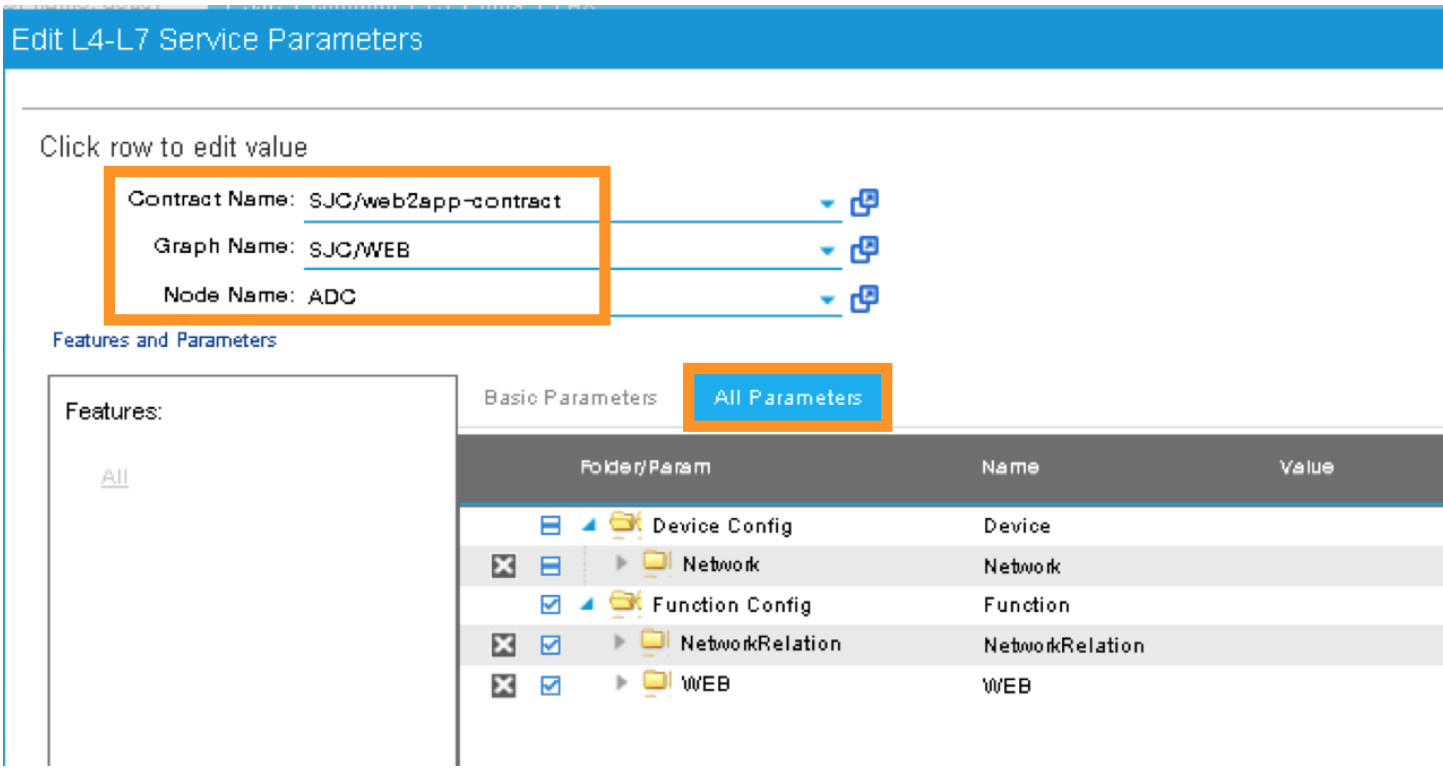
Select the following:

Contract Name: `SJC/web2app-contract`

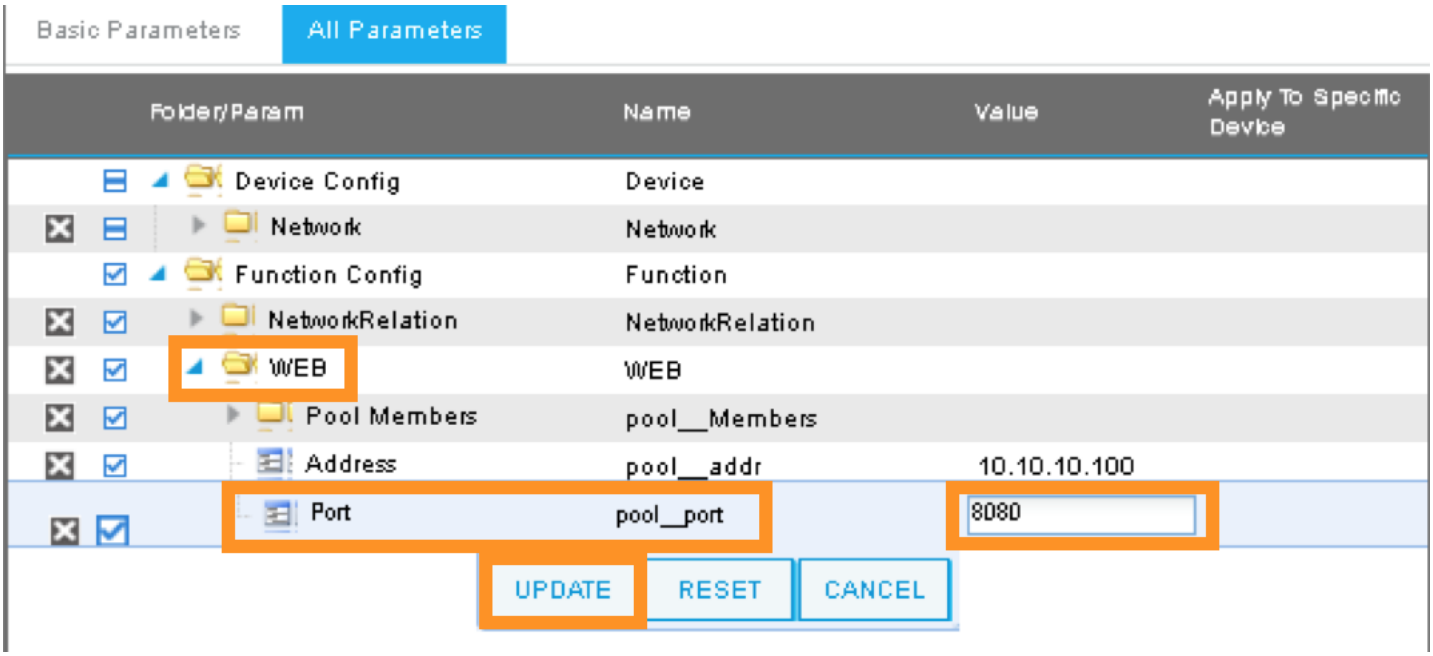
Graph Name: `SJC/WEB`

Node Name: `ADC`

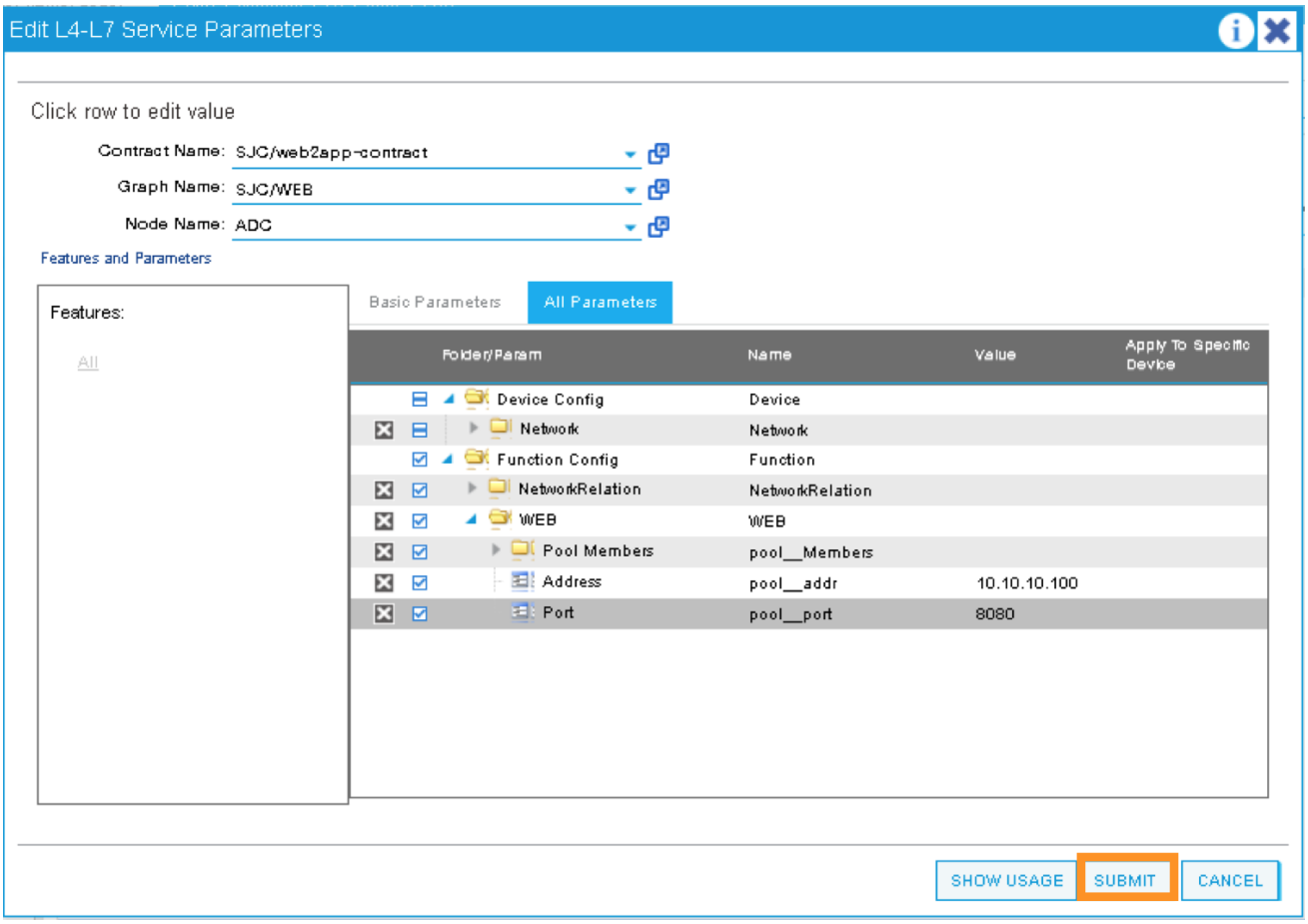
Then click “All Parameters”



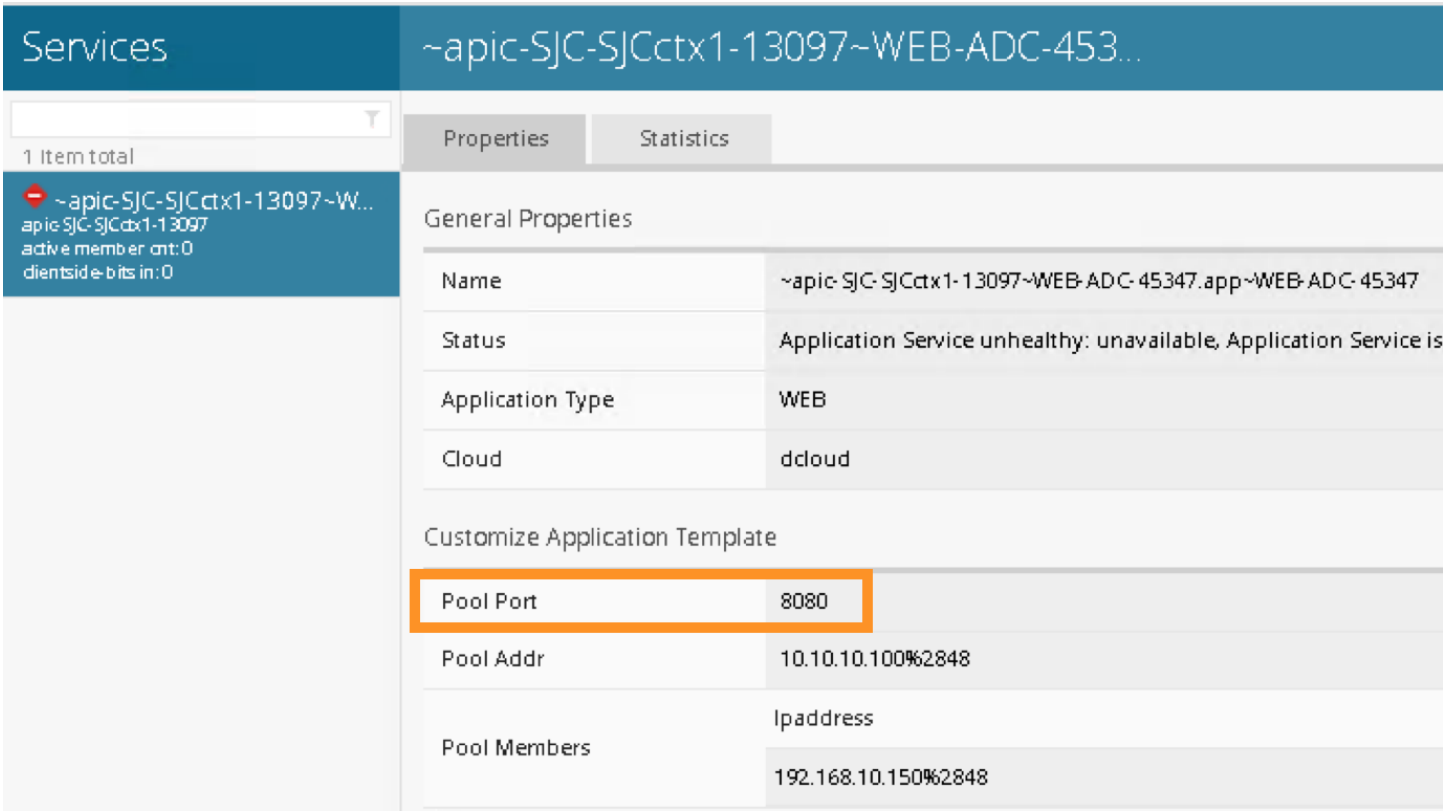
Expand `WEB` folder, double click on `pool__port`, change the value from `80` to `8080`, then “UPDATE”



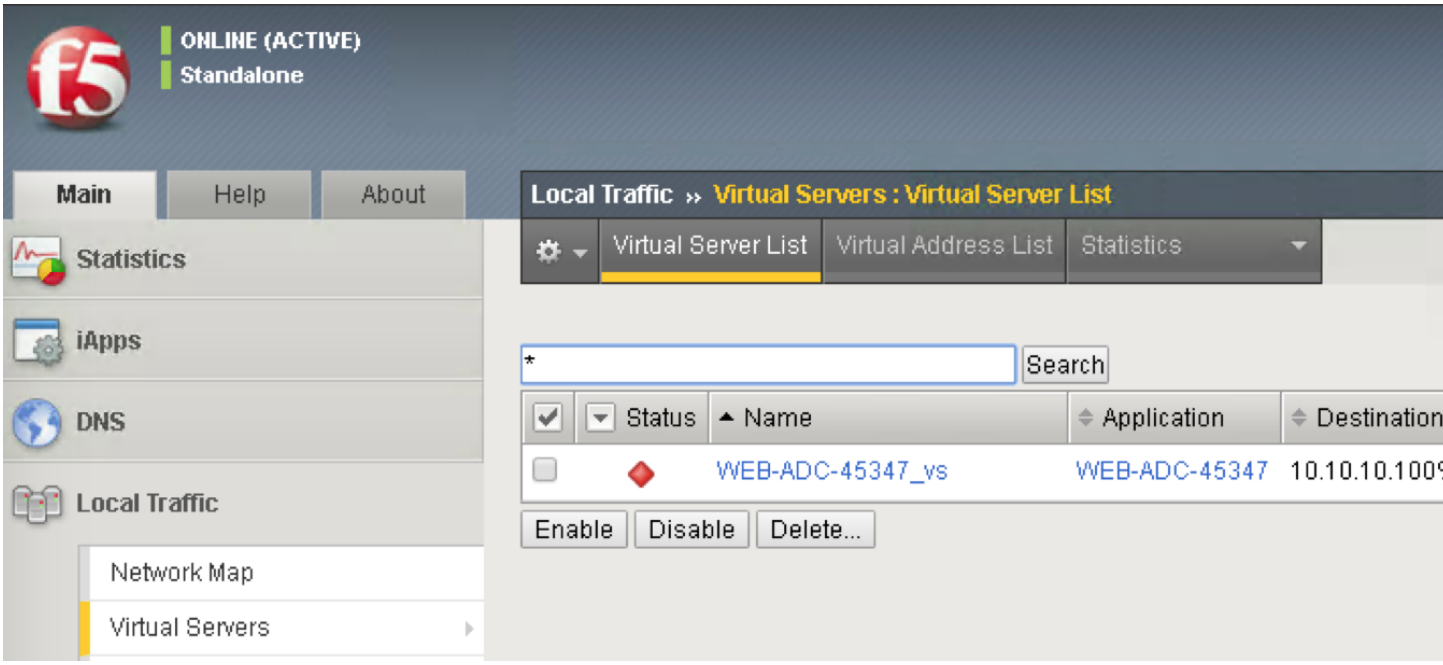
Then “SUBMIT”



Notice on iWorkflow, under Services, the port value is updated to 8080



BIG-IP virtual server reflects the same configuration update



This concludes Scenario 2 "Modify L4 – L7 deployed graph parameters" lab.

2.4 Remove APIC Service Graph

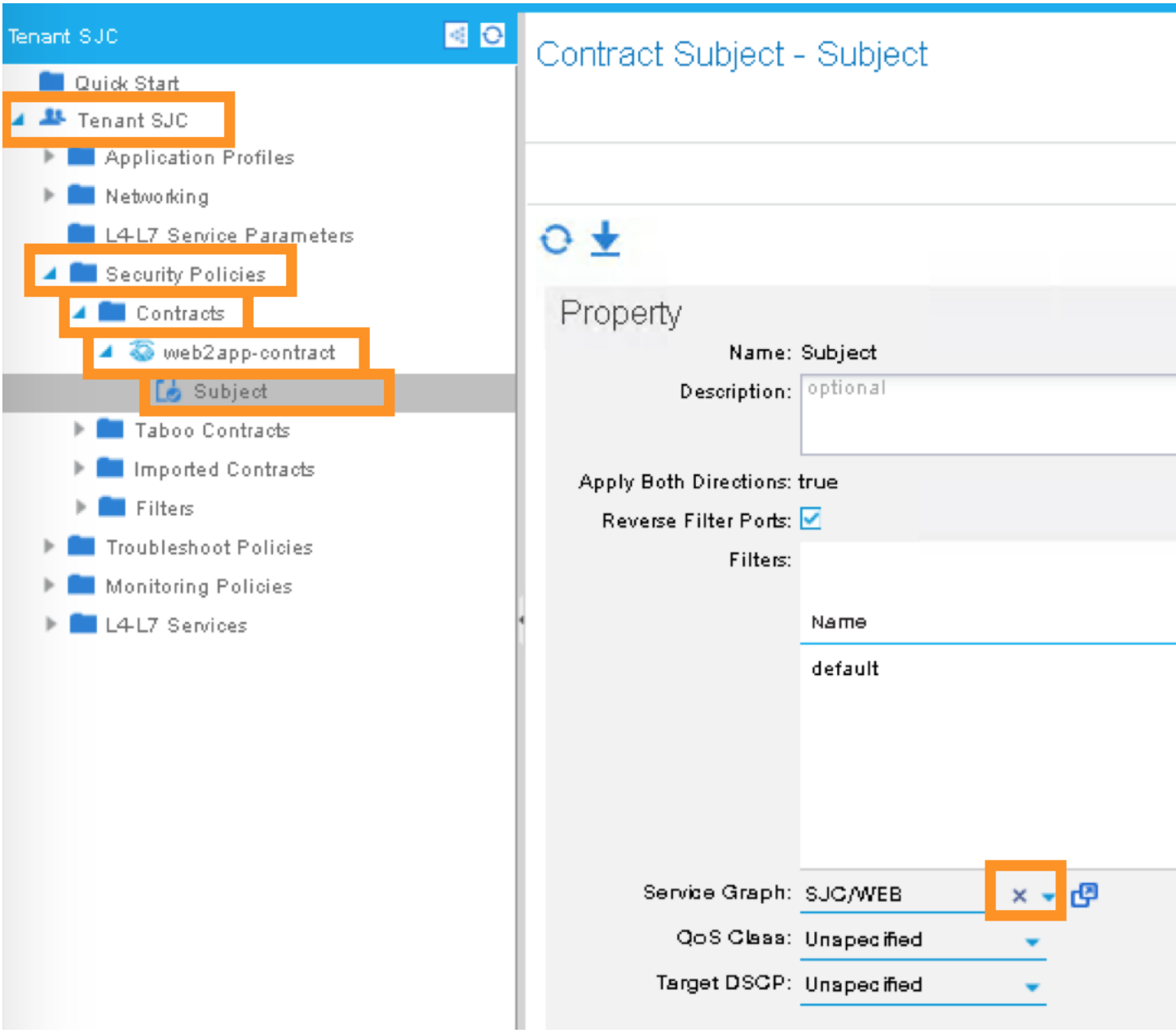
2.4.1 APIC – Remove Only Service Graph Deployment

The easiest way to remove a service graph deployment, which is same as removing virtual server from the BIG-IP, yet remain all the EPG and device selection policy parameters for easy re-deployment is to un-associate a service graph under the contract subject.

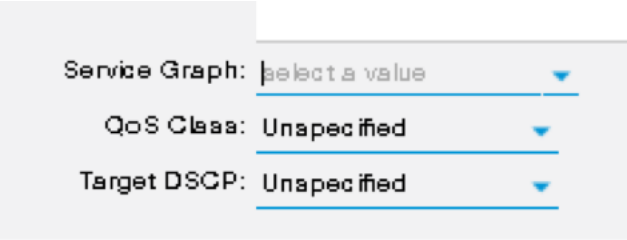
Go to the contract subject by clicking the following:

Tenants SJC -> Security Policies -> Contracts -> web2app-contract -> Subject

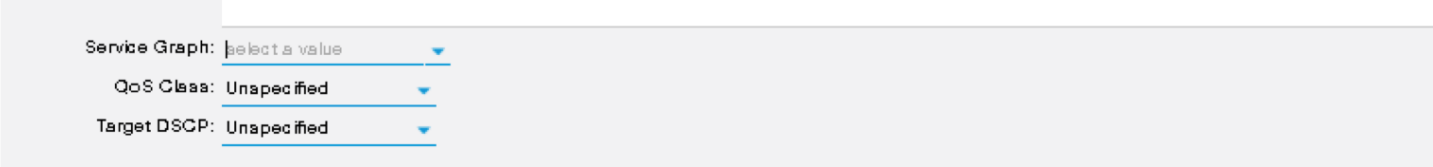
Move the mouse to Service Graph and hover near the drop-down menu, you will see “X”, click “X” and graph will be removed from contract subject:



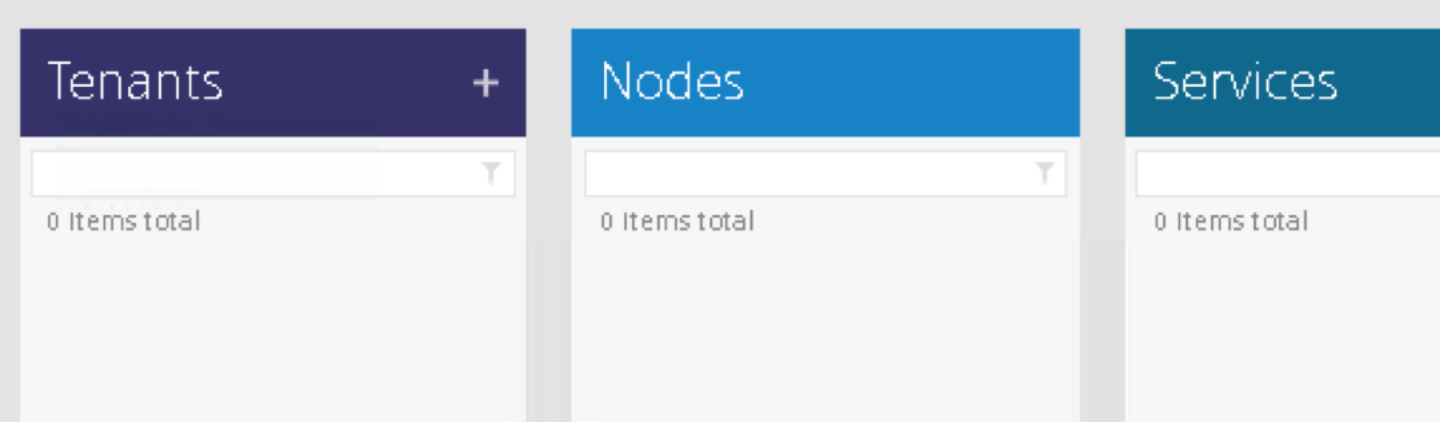
Click “X”, the service graph SJC/WEB will disappear:



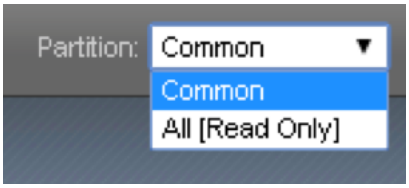
Click “SUBMIT”



Notice iWorkflow: Tenant, Service and Node are empty:

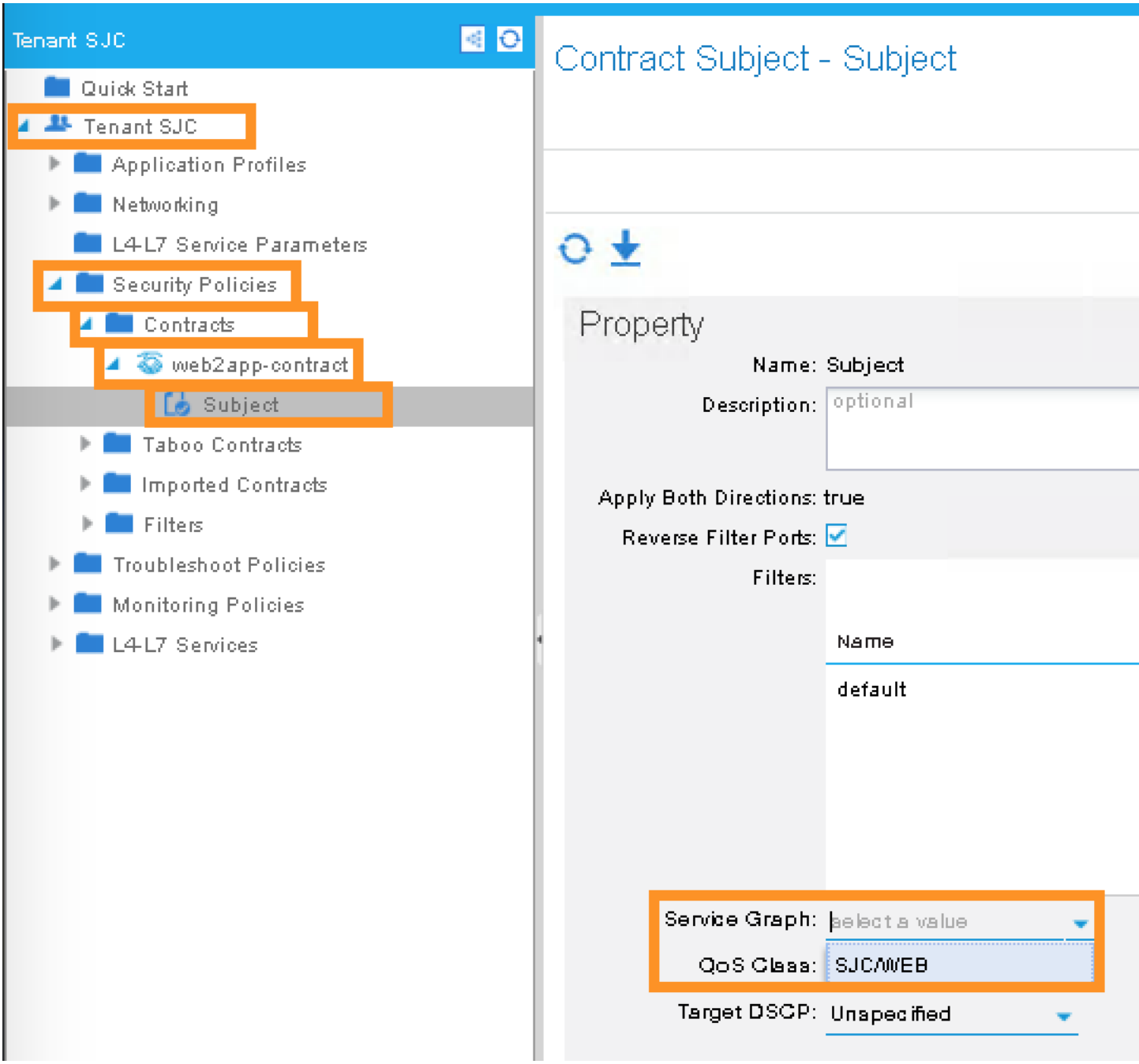


BIG-IP, the partition is removed, including all virtual servers and network related configurations:

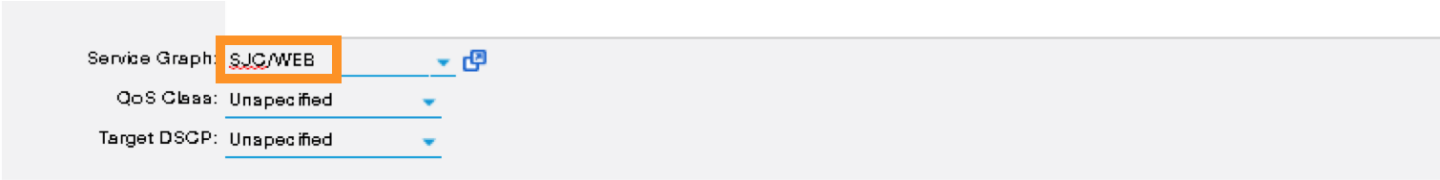


2.4.2 APIC – Re-deploy Service Graph

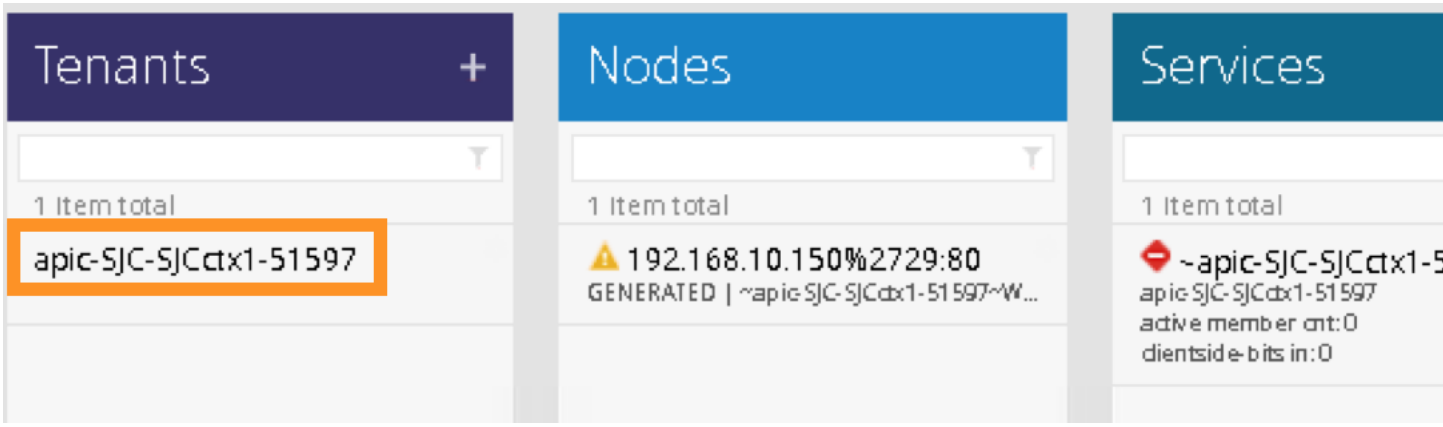
In order to re-deploy the same graph, simply go to contract subject and re-associate SJC/WEB under Service Graph:

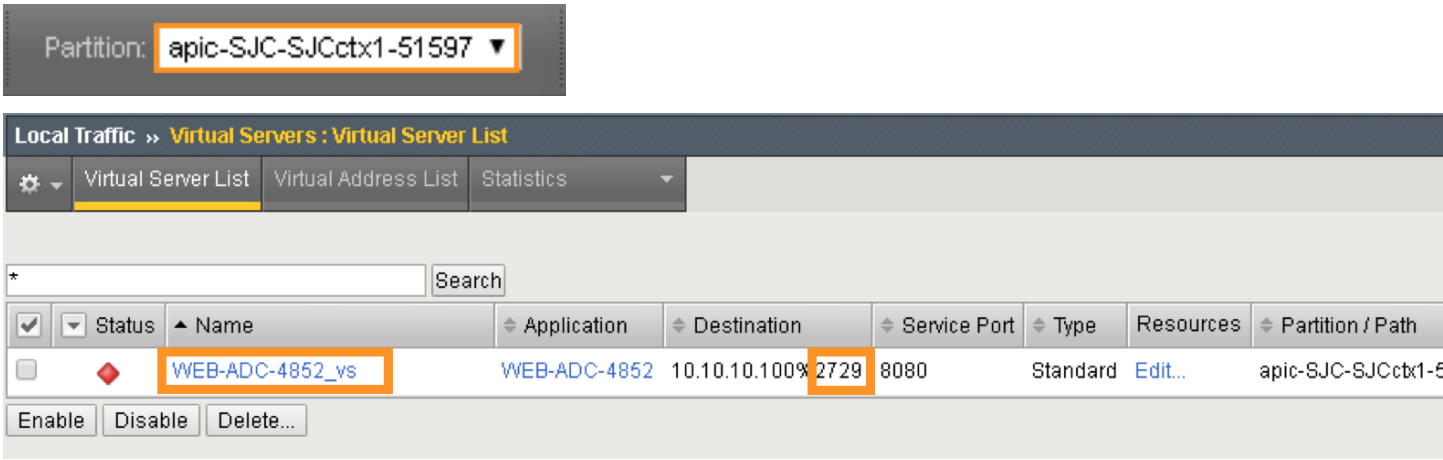


Click “SUBMIT”



You will see the Application Service is redeployed in iWorkflow and BIG-IP





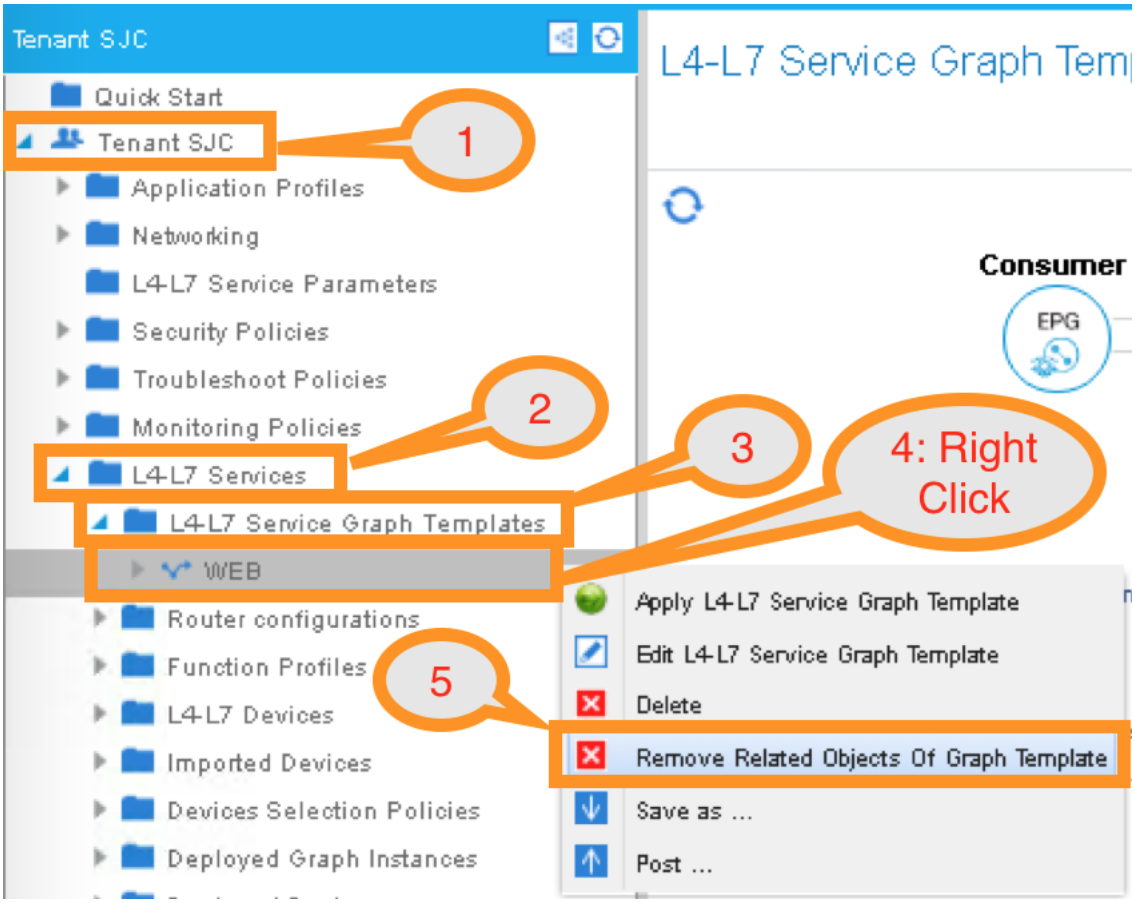
Notice the tenant VID, graph ID and the RD values are different from previous deployment.

2.4.3 APIC – Remove all graph associated objects

If you want to clean up all the related objects of the deployed graph template, go to:

Tenants SJC ->L4-L7 Services -> L4-L7 Service Graph Templates, right click on the graph template WEB, then select

“Removed Related Objects of Graph Template”



Select:

Contract: web2app-contract

Provider EPF: App1/app

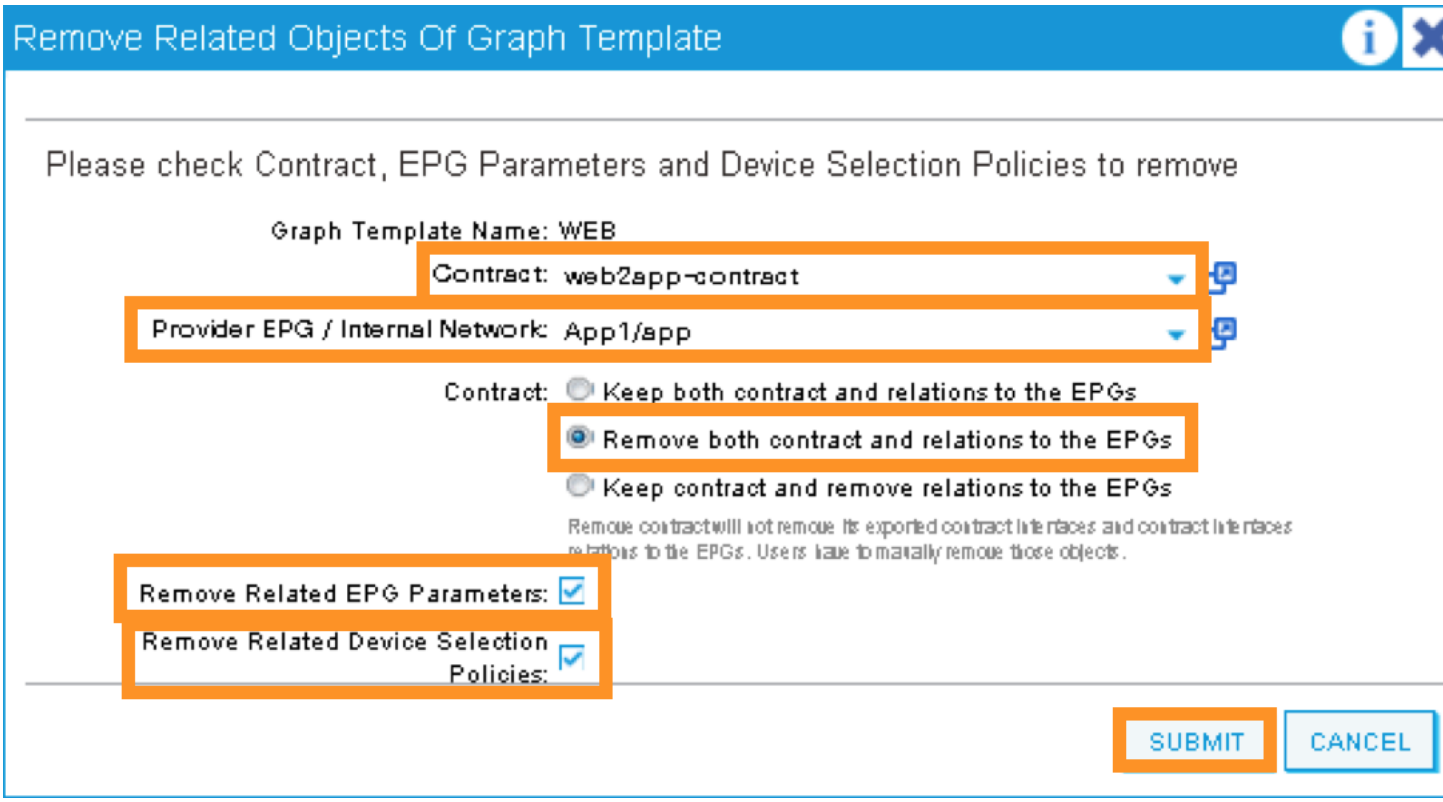
Radio button: “remove both contracts and relations to the EPGs”

Check box:

Remove related EPF parameters <- this will remove all L4-L7 parameters of this particular contract/graph/node under EPG

Remvoe related device selection policies <- this will remove connectivity policy of this particular contract/graph/node

Click “SUBMIT”

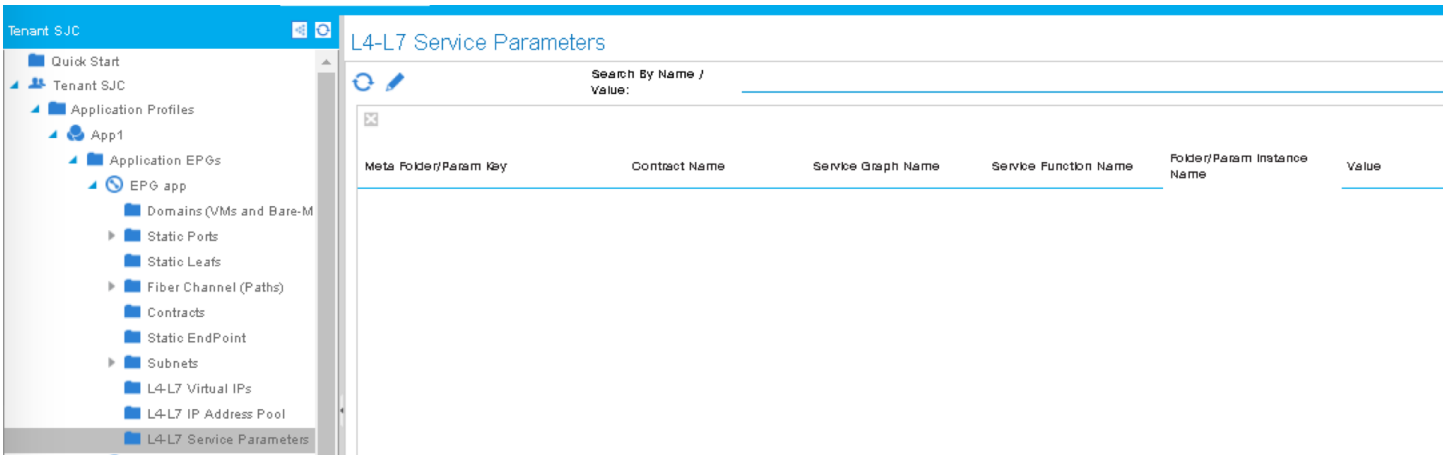


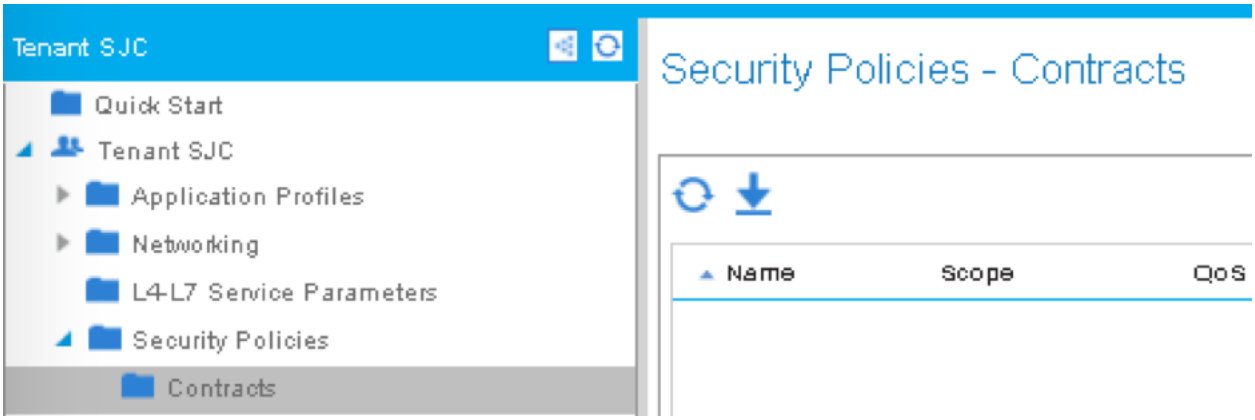
Notice on APIC:

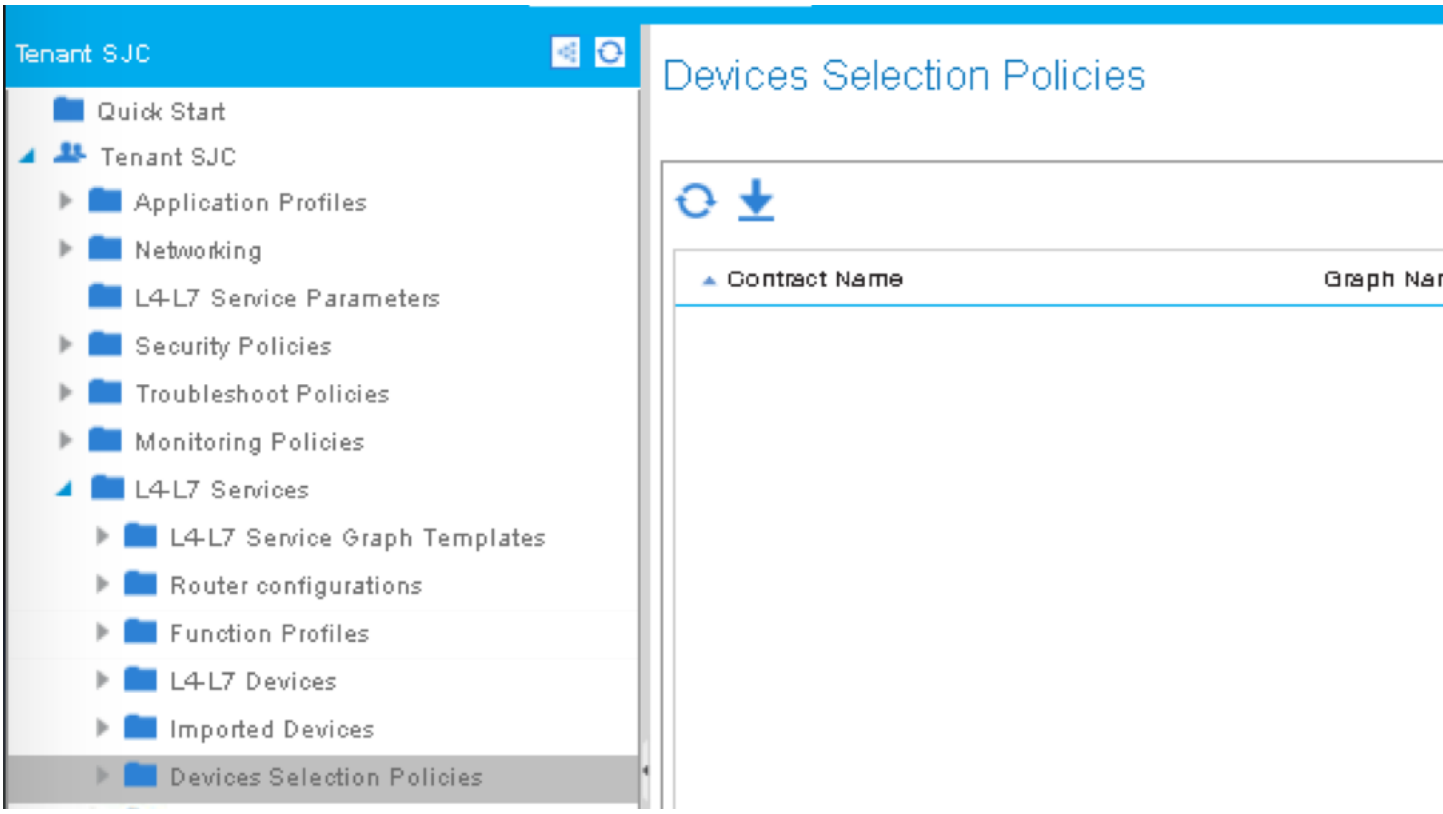
EPG app: related L4-L7 Services Parameters are removed

Related Devices Selection Policies is removed

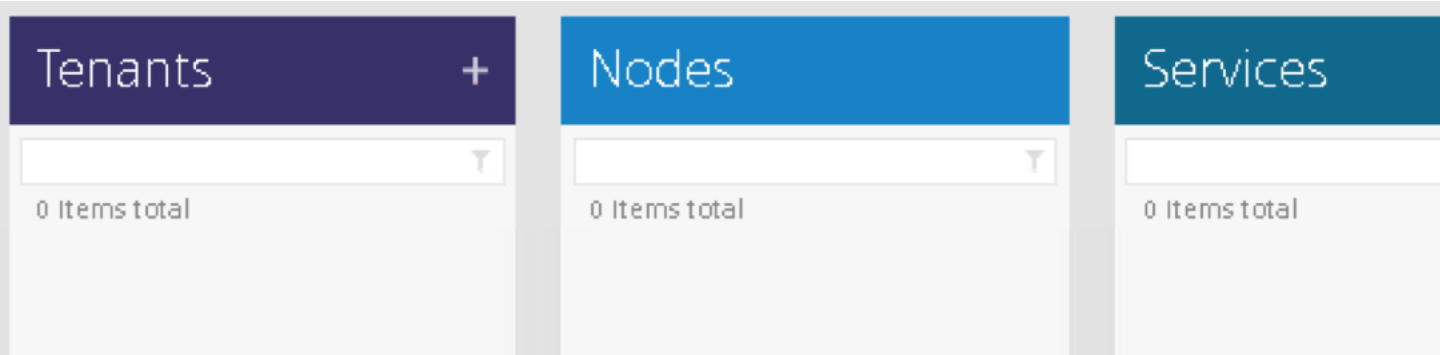
Related contract is removed



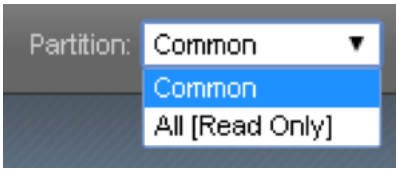




F5 iWorkflow configuration related to APIC tenant and service graph is un-configured



BIG-IP is also clean:

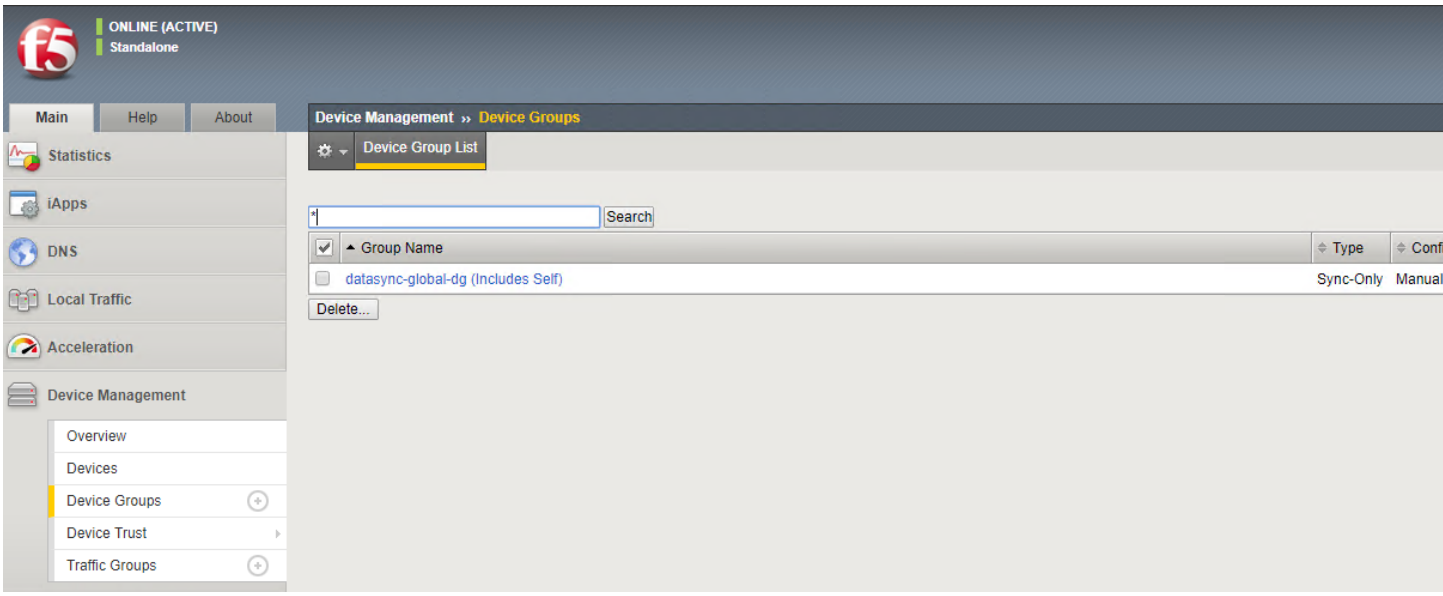


2.4.4 APIC – Remove L4-L7 Devices from Tenant Common

Remove the L4-L7 logical device cluster from common tenant.

Tenant Common->L4-L7 Services -> L4-L7 devices -> , right click on the logical device cluster and click delete

This will also delete the device group from the BIG-IP (no device group correcponding to the logcail device cluster present anymore)



2.4.5 APIC – Remove Device Manager from Tenant Common

Remove the device manager from common tenant.

Tenant Common->L4-L7 Services -> L4-L7 devices -> Device Managers-> ‘dcloud-device-manager, right click on the device manager and click delete

2.4.6 APIC – Remove Device Manager Type from L4-L7 Services

Remove the device manager type from L4-L7 services

Go to L4-L7 Services -> Inventory -> Device manager types , right click on the device manager and click delete

vThis conclude Scenario 3 “Remove APIC Service Graph” lab.

2.5 Using POSTMAN REST client to deploy service graph

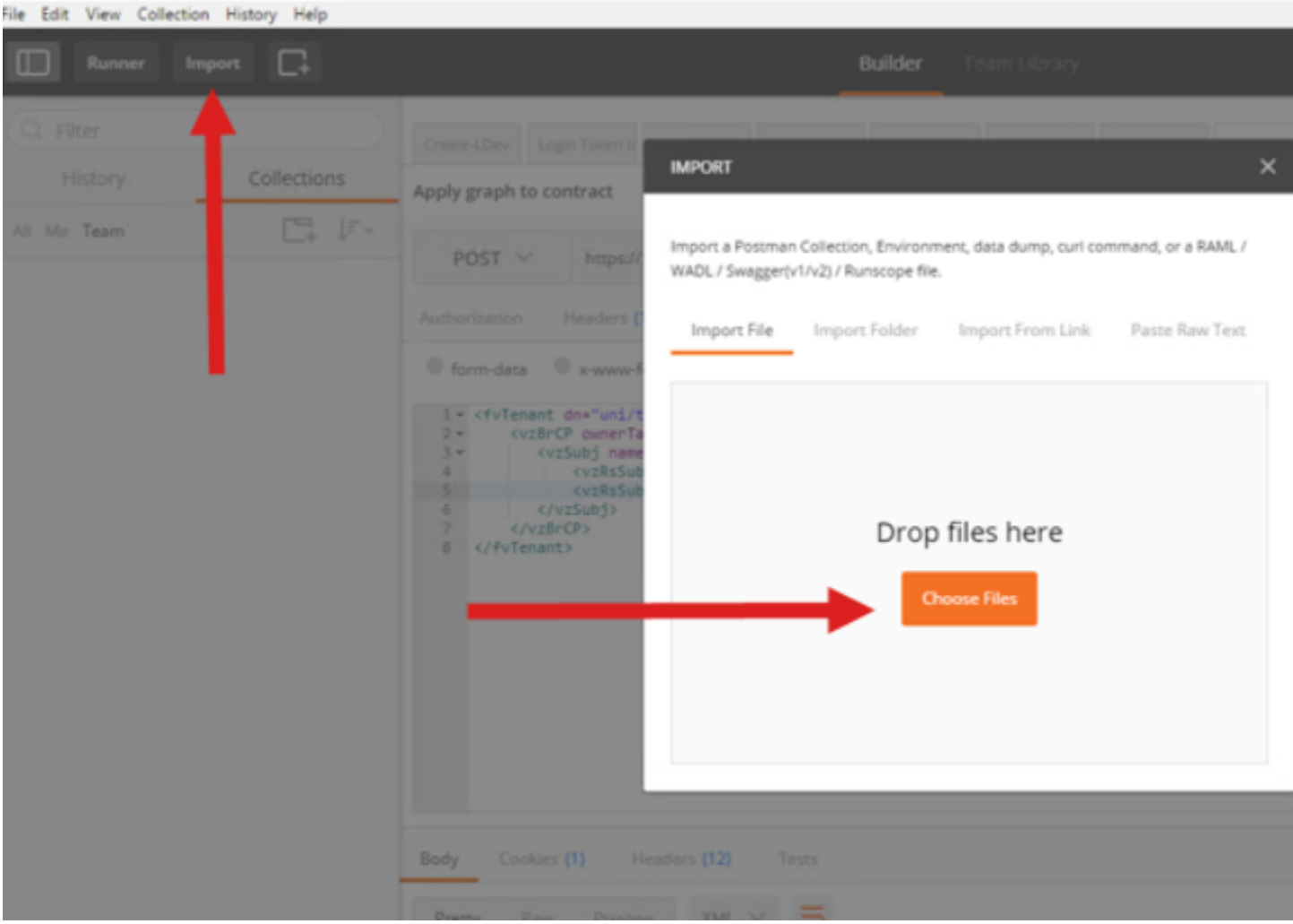
Launch POSTMAN from desktop

Import the POSTMAN collection

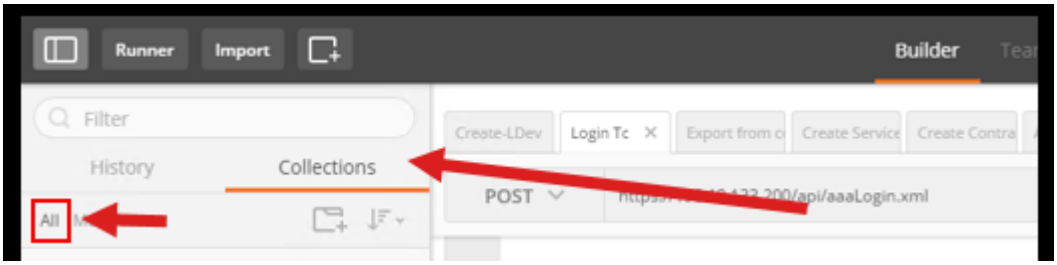
The JSON collection if saved on your desktop - 'dCloud-F5-iWorkflow-App-iApps-Final.postman_collection.json'

Click on Collection->Import

Click on the 'Choose Files' button and browse to the json collection and import it



The POSTMAN collection will be loaded in your POSTMAN window:

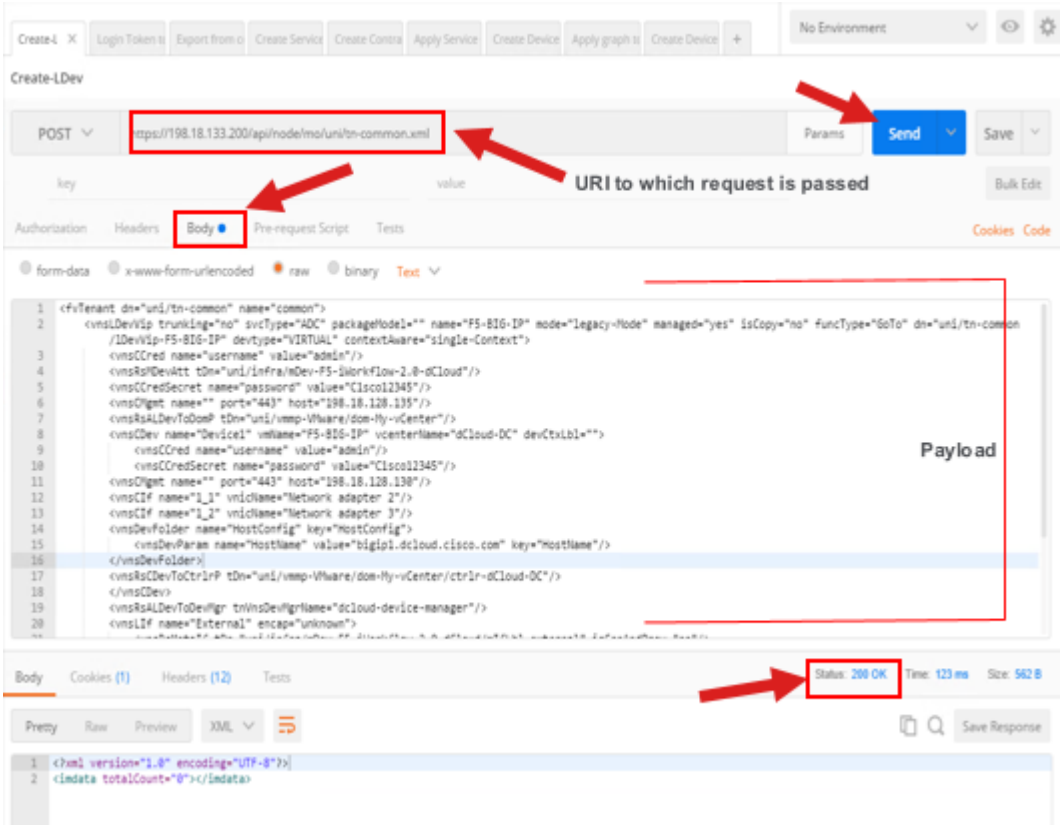


To view what each API call executed, click on the POST requests

Click on the Body to view the payload being passed

Click the Send button to execute the request

Check the status at the bottom of the window to see if the request got executed successfully (200 OK)



Note: Device package install, device manager configuration has already been done, POSTS are from the point of when a graph is to be created

Run each postman POST and then see the corresponding object created on the APIC

1. *Login Token to APIC* – Used for authentication to the APIC. The response to the POST operation will contain an authentication token. Subsequent operations on the REST API will use this token value to authenticate future requests.
2. *CreateDeviceManagerType* – Used to create a device manger type under L4-L7 services->Inventory
3. *CreateDeviceManager-Common* – Will create a device manager which has iWorkflow credentials under tenant common
4. *Create-Ldev-Common*– Creates a logical device cluster on the APIC in tenant common
5. *Export from Common to SJC tenant* – Exports the LDev from common tenant to SJC tenant
6. *Scope Network under AP* – This will scope the network parameters like self IP/route under the application profiles
7. *Create contract* – Creates a contract to be used in tenant SJC
8. Assign contract to web EPG

9. Assign contract to app EPG
10. *Create service graph template* – Creates the service graph template to be used
11. *Apply service graph template* – Specifies the parameters (virtual server/pool. Pool members etc.) to be configured for this particular graph
12. *Create device selection policy* – Creates a device selection policy (This construct gets created automatically when using the UI, this is an extra step needed when using automation)
13. *Apply graph to contact* – Attach the graph to the contract

This conclude Scenario 4 "POSTMAN REST client" lab.

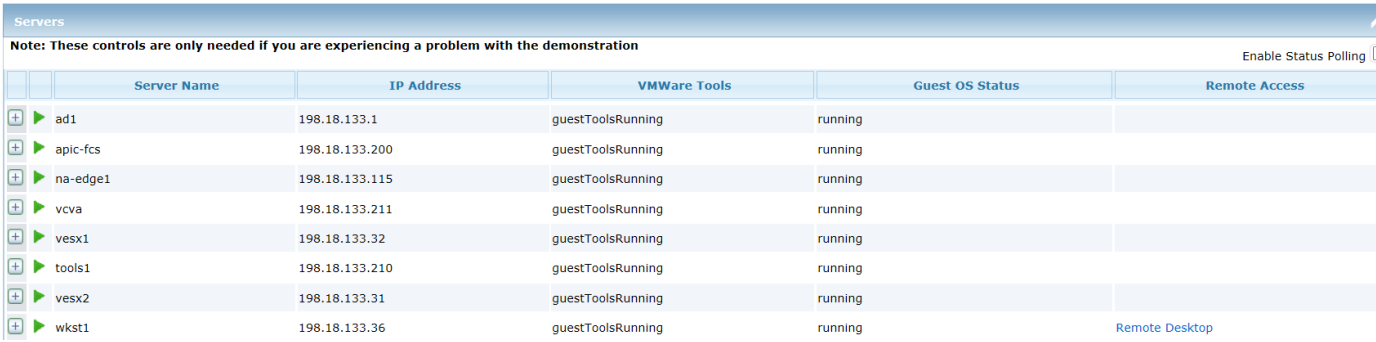
A. Reset APIC Simulator

APIC Fabric Members are created by default, so that the demonstration can begin with the creation of the APIC objects.

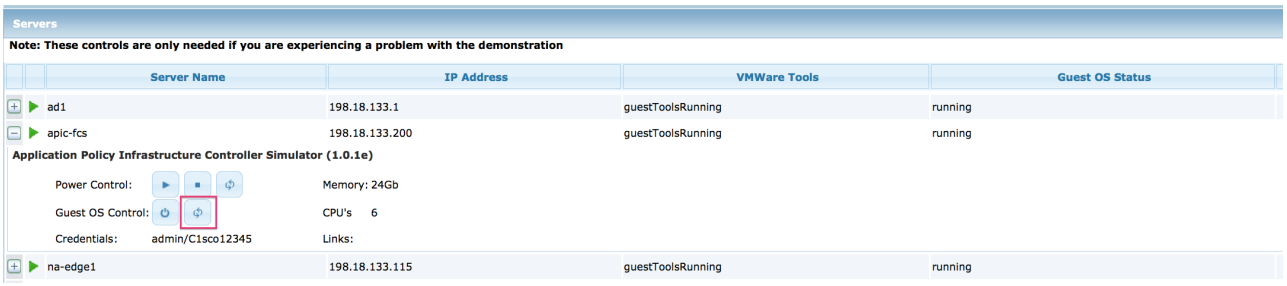
If you want to demonstrate the fabric discovery, reboot the **apic-fcs** via Guest OS Control as follows:

1. From the Demo Dashboard, click **Servers**.
2. Servers Tab

3. From the **Servers** list, click the  next to **apic-fcs**.



4. Click the **Reboot** button in **Guest OS Control** to restart the server.



Note: It will take up to 5 minutes before you can login and rebuild the Fabric using one of the Fabric Discovery methods in *Appendix B*.

A. Fabric Discovery

If they are not configured, use one of the three methods below to configure:

Method	Automation Level	Explanation	Completion Time
Script Configuration	High	Skip the configuration steps and discover the APIC Fabric automatically, as shown in *Configure APIC Fabric Using Scripts*.	1 minute, followed by 15 minutes to build the fabric
Wizard Configuration	Medium	Set up the APIC Fabric using the Postman–REST client, as shown in *Configure APIC Fabric Using Postman–REST Client*.	5 minutes, followed by 15 minutes to build the fabric

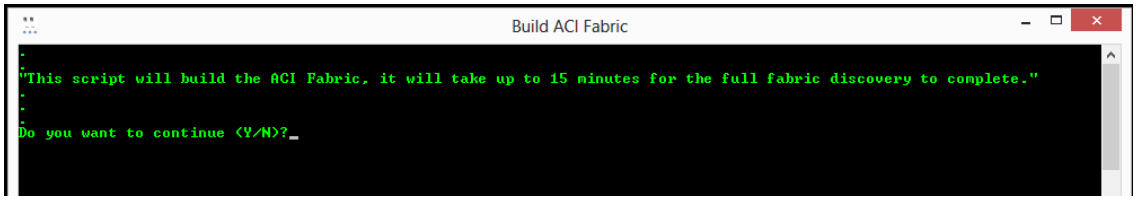
Note: The full fabric discovery can take up to 15 minutes. The apic3 controller will be discovered after all the devices are discovered. You can check monitor the progress by selecting **Topology** from the **Inventory** pane in the APIC GUI. While the discovery is taking place, you can complete *Scenario 1*, which ends in the APIC Topology window showing the discovered elements.

Demonstration Steps

2.5.1 Configure APIC Fabric Using Scripts

1. From the demonstration workstation, click the **Build ACI Fabric** icon. 

2. Type **Y <Enter>** at the **Do you want to continue (Y/N)?** prompt. The script will begin building the fabric, which will take about 15 minutes.
3. Build ACI Fabric Script



4. Type **Y <Enter>** at the **Do you want to continue (Y/N)?** prompt. The script will begin building the F5, which will complete before the ACI fabric is set up.

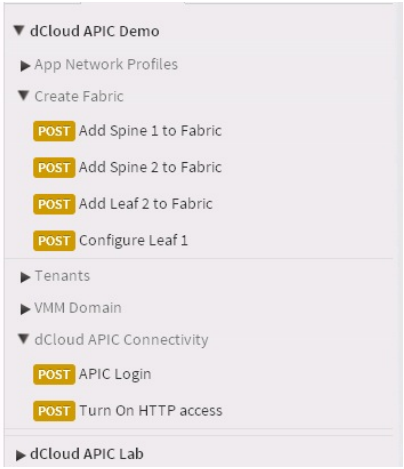
2.5.2 Configure APIC Fabric Using Postman–REST Client

1. From the demonstration workstation, launch '**APIC Login**', and then log in to the **Application Policy Infrastructure Controller** with the following credentials: Username: **admin**, Password: **C1sco12345**.
2. From the menu bar, click **Fabric**.
3. From the sub-menu bar, click **Inventory**.
4. In the left-pane, choose **Fabric Membership**.
5. Review the current members of the Fabric.
6. Fabric Membership

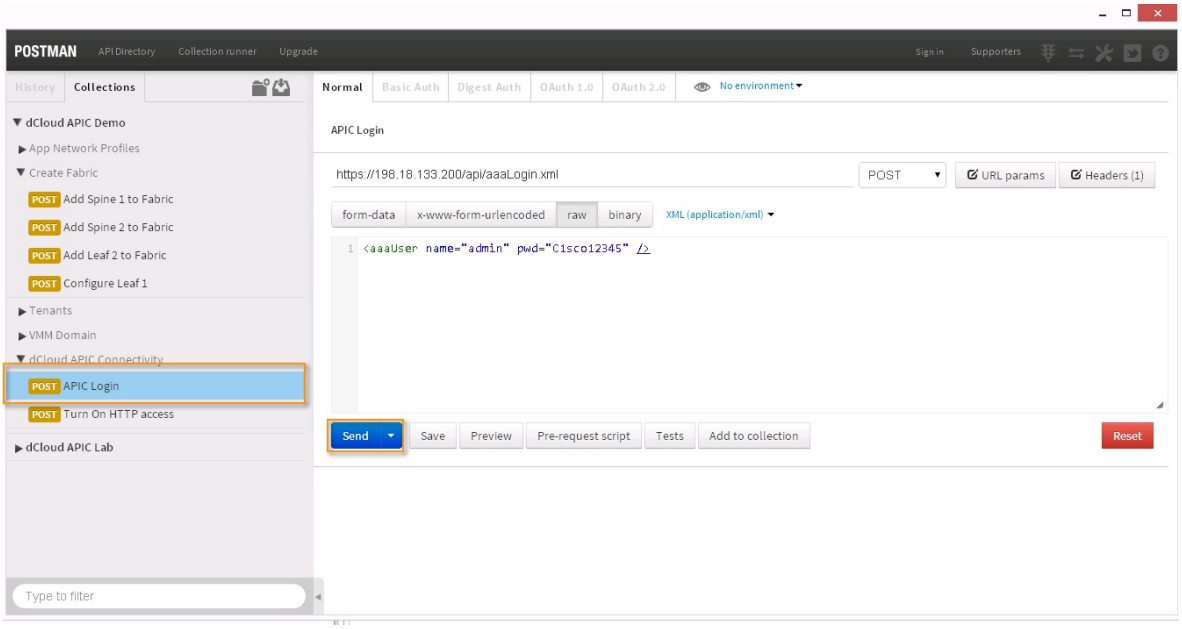
7. Launch the **Postman – REST Client** [] from the taskbar. You are automatically be logged in. This is where you will register the switches for the APIC.

Note: If you get a status of **403 Forbidden** while performing the activity in this scenario, review the text below for more information on the error. If you see **Token was invalid (Error: Token timeout)**, this means that your session has timed out. You will need to launch the **APIC Login** POST [POST APIC Login] and then proceed with the next POST.

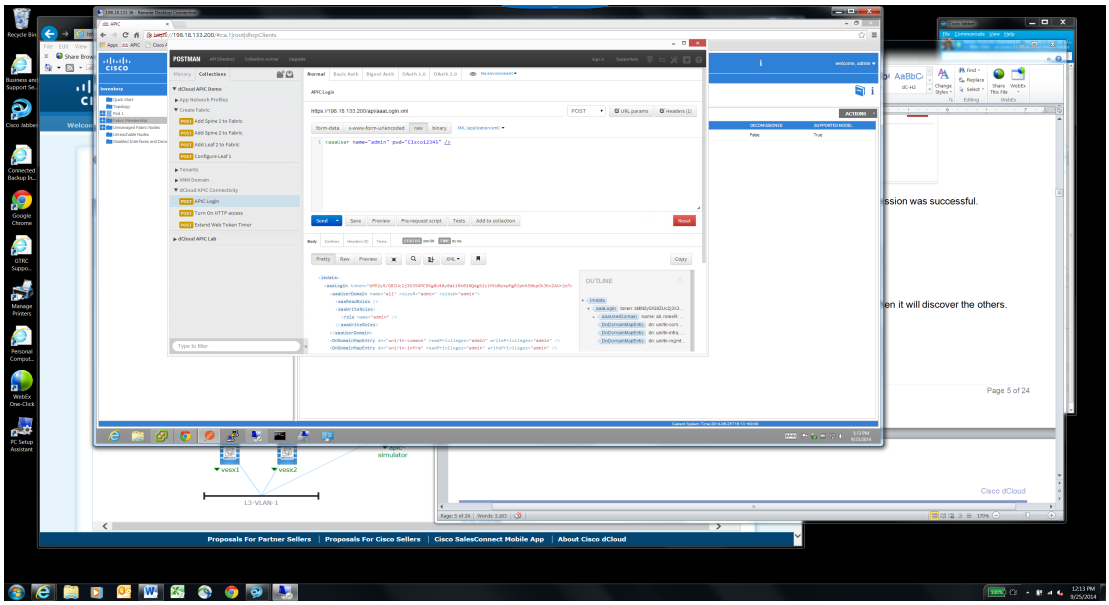
8. In the left-pane, click the arrow [▶] next to **dCloud APIC Demo**, and then click the arrow next to **Create Fabric** and **dCloud APIC Connectivity**.
9. dCloud APIC Demo



10. Go to **dCloud APIC Connectivity** and then choose **APIC Login**. Click **Send** to connect to the APIC.
11. APIC Login and Send



12. Review the **Status** of the submission. A result of **200 OK** means the submission was successful.
13. Status



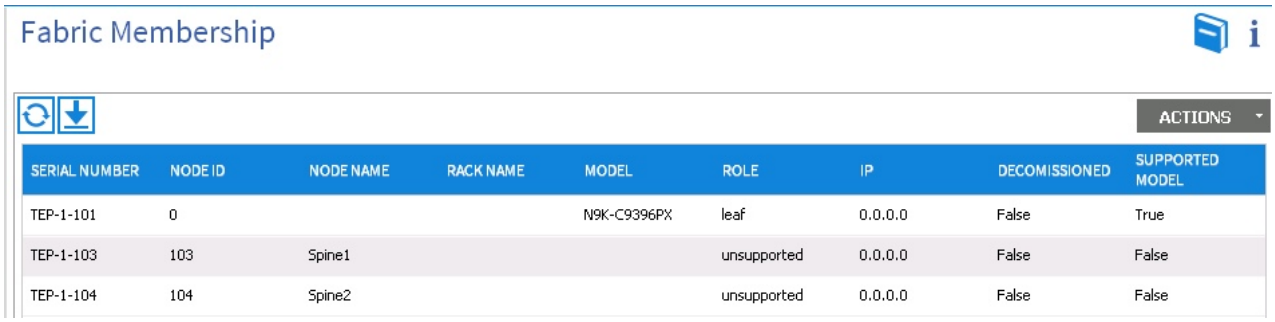
14. Go to **Create Fabric**.
15. Choose the **Add Spine1 to Fabric** post. Click **Send** to configure the first spine,a and then it will discover the others.
16. Review the status of the submission.
17. In the APIC application window, you can see Spine1 is now part of the Fabric Membership.
18. Fabric Membership

Fabric Membership

ACTIONS

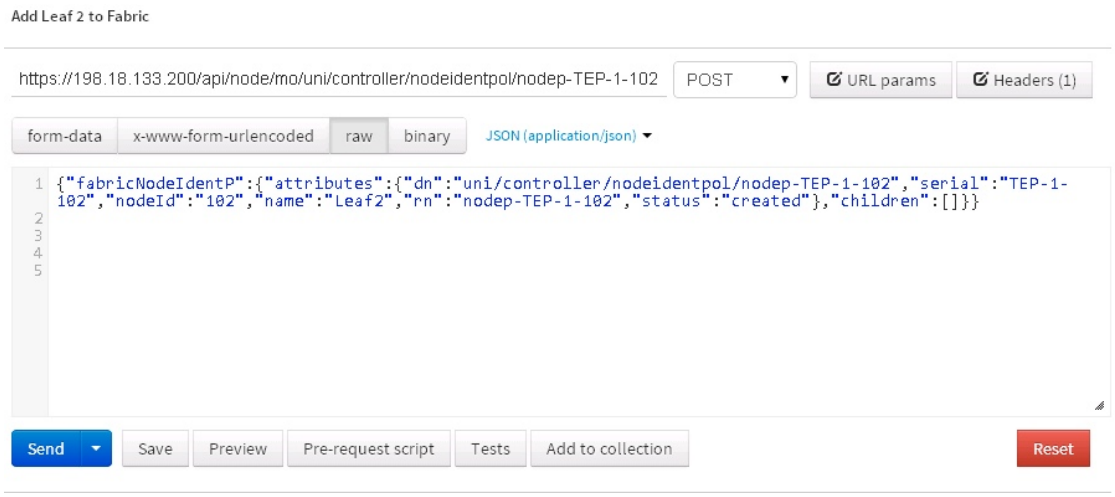
SERIAL NUMBER	NODE ID	NODE NAME	RACK NAME	MODEL	ROLE	IP	DECOMISSIONED	SUPPORTED MODEL
TEP-1-101	0			N9K-C9396PX	leaf	0.0.0.0	False	True
TEP-1-103	103	Spine1			unsupported	0.0.0.0	False	False

19. Go to the **Postman – REST Client** window.
20. Under **Create Fabric**, choose the **Add Spine2 to Fabric** post and then click **Send** to configure the second spine.
21. Review the status of the submission.
22. In the APIC window, you can see Spine2 is now part of the Fabric Membership.
23. Fabric Membership

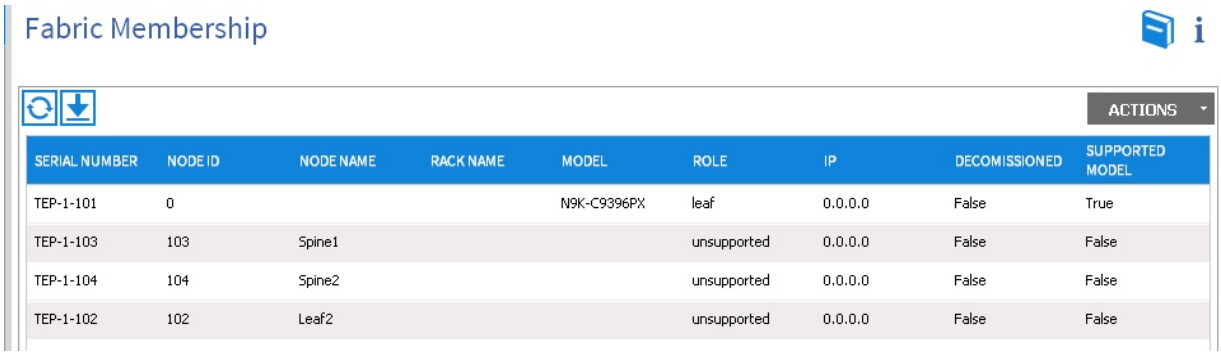


SERIAL NUMBER	NODE ID	NODE NAME	RACK NAME	MODEL	ROLE	IP	DECOMISSIONED	SUPPORTED MODEL
TEP-1-101	0			N9K-C9396PX	leaf	0.0.0.0	False	True
TEP-1-103	103	Spine1			unsupported	0.0.0.0	False	False
TEP-1-104	104	Spine2			unsupported	0.0.0.0	False	False

24. Go to the **Postman – REST Client** window.
25. Under **Create Fabric**, choose the **Add Leaf2 to Fabric** post.
26. Review the command for this post and you can see that it:
    - Looks for the serial number (TEP-1-102)
    - Sets up the serial number for node 102
    - Names Leaf2
27. Add Leaf2 to Fabric

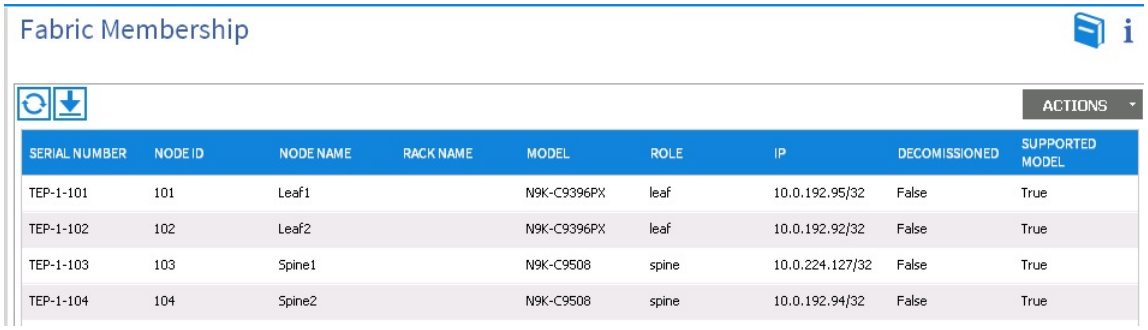


28. Click **Send**.
29. Review the status of the submission.
30. In the **APIC window**, you can see **Leaf2** is now part of the **Fabric Membership**.
31. Fabric Membership



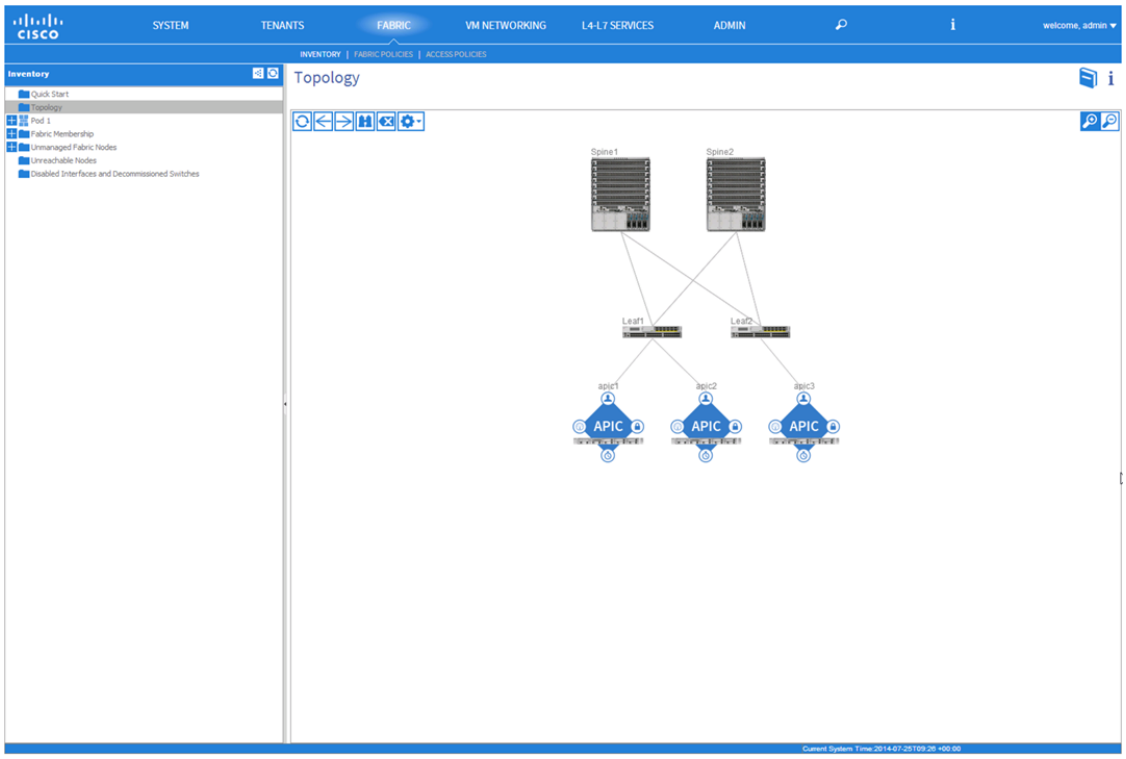
SERIAL NUMBER	NODE ID	NODE NAME	RACK NAME	MODEL	ROLE	IP	DECOMISSIONED	SUPPORTED MODEL
TEP-1-101	0			N9K-C9396PX	leaf	0.0.0.0	False	True
TEP-1-103	103	Spine1			unsupported	0.0.0.0	False	False
TEP-1-104	104	Spine2			unsupported	0.0.0.0	False	False
TEP-1-102	102	Leaf2			unsupported	0.0.0.0	False	False

32. Go to the **Postman – REST** Client window.
33. Under **Create Fabric**, choose the **Configure Leaf 1 to Fabric** post, which will update the first member of the Fabric.
34. Click **Send**.
35. Review the status of the submission.
36. In the **APIC window**, you can see that **Node ID** and **Node Name** have been set for serial number TEP-1-101.
37. As it discovers Leaf1, an IP address is allocated.
38. The discovery will continue until it finds all of the links to the other members and populates the IP Addresses.
39. Fabric Membership



SERIAL NUMBER	NODE ID	NODE NAME	RACK NAME	MODEL	ROLE	IP	DECOMISSIONED	SUPPORTED MODEL
TEP-1-101	101	Leaf1		N9K-C9396PX	leaf	10.0.192.95/32	False	True
TEP-1-102	102	Leaf2		N9K-C9396PX	leaf	10.0.192.92/32	False	True
TEP-1-103	103	Spine1		N9K-C9508	spine	10.0.224.127/32	False	True
TEP-1-104	104	Spine2		N9K-C9508	spine	10.0.192.94/32	False	True

40. Wait for discovery to finish. In the APIC window, select **Fabric > Inventory** from the main menu. Click **Topology** and demonstrate that the entire fabric has been discovered and is included in the topology.
41. Fabric Discovery Topology



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

3

Class 3: Automation of Cisco APIC and F5 BIG-IP using Ansible

Pre-requisites

- iApps to be used for service insertion is already present on the iWorkflow

All pre-requisites are already satisfied for this lab. We DO NOT need to do the above

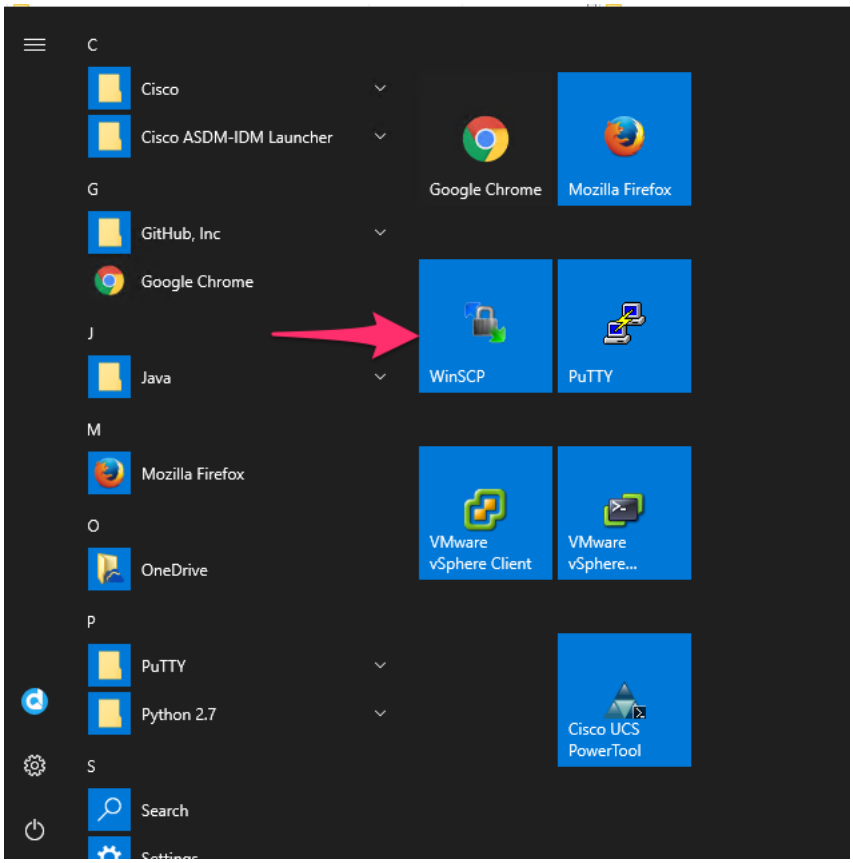
3.1 Lab Topology

3.1.1 Install Ansible

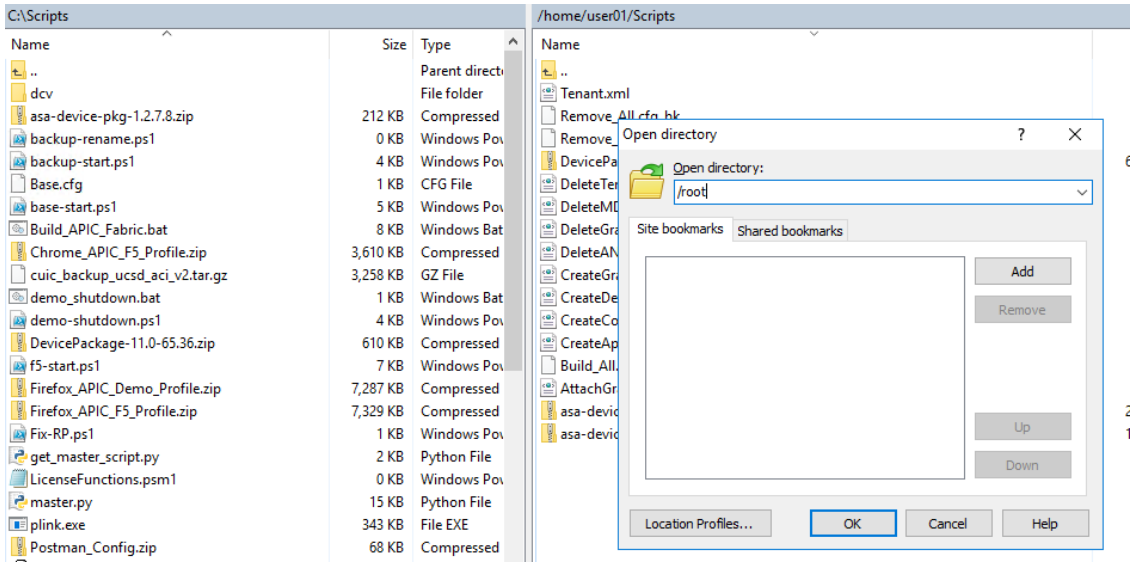
- On dCloud once logged into RDP, open Putty and go to server 'Tools' (root/C1sco12345). Run the following commands to install Ansible
  - `pip install --upgrade pip`
  - `yum install openssl-devel`
  - `yum install python-devel`
  - `yum install gcc`
  - `pip install cryptography`
  - `pip install ansible`
- Once ansible is installed successfully, run following command from /root directory
  - `export ANSIBLE_LIBRARY=/root/library`

3.1.2 Environment setup

- Download `ansible_automation_files.tar` from `https://tinyurl.com/y9zvj6nl` to desktop
- Open WinSCP, click on with windows startup button and then click WinSCP



- On WinSCP
  - Hostname: `tools.dcloud.cisco.com`
  - Port: `22`
  - Click on the EDIT button to change username and password
    - Username: `root`
    - Password: `C1sco12345`
  - Click Save
  - Click login
  - In the right hand pane click on the `/home/user01/Scripts` tab, change it to `/root`

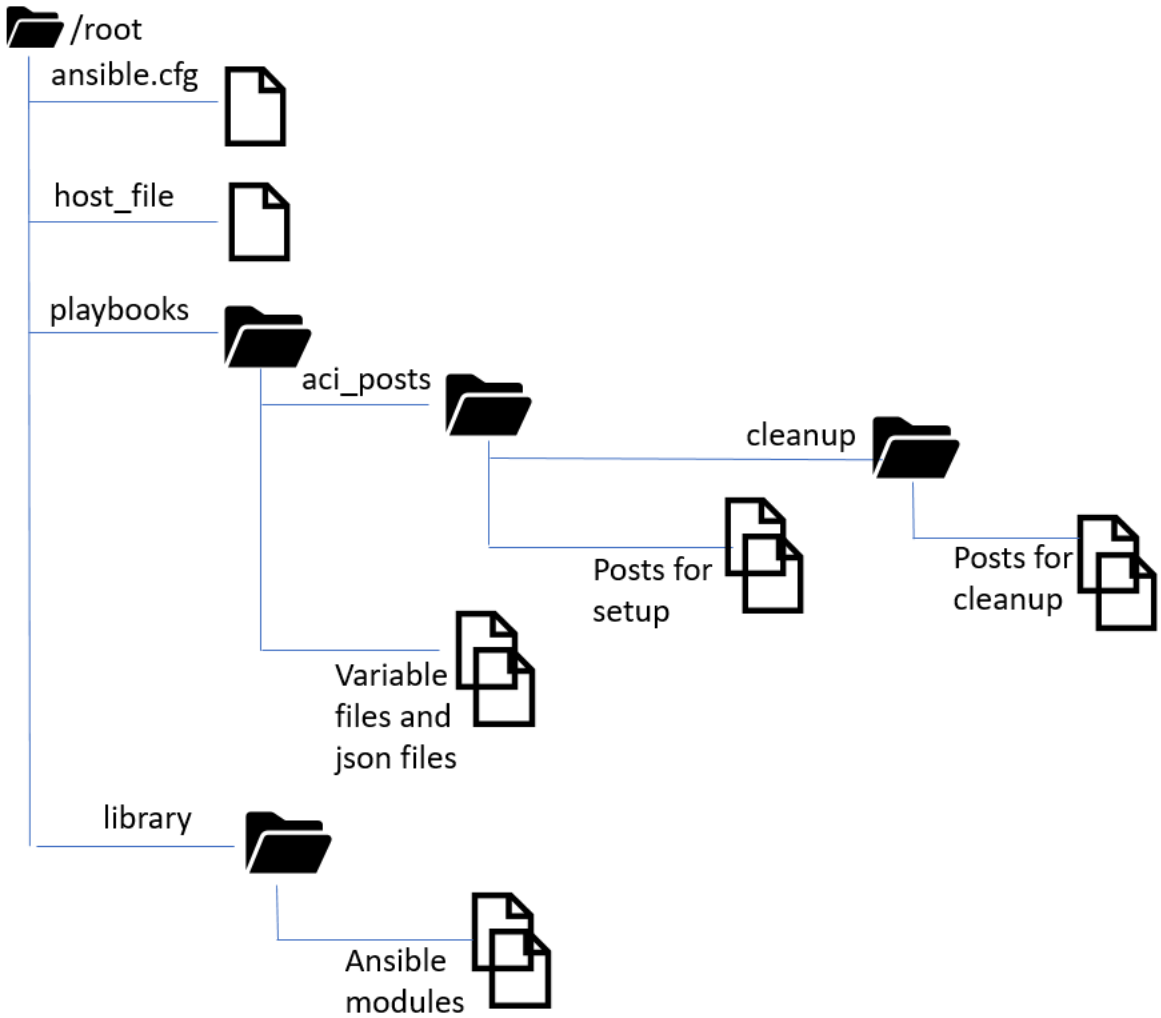


  - Click OK
  - Similarly change the left hand pane from `C:\Scripts` to `C:\Users\demouser\Desktop`
  - Copy the download tar file from the desktop to the root directory on the ansible host
- SSH to the 'Tools' host using Putty
  - Username: `root`
  - Password: `C1sco12345`
  - Untar the `ansible_automation_files.tar` file using command:

    ```
    tar xvf ansible_automation_files.tar
    ```

3.1.3 Directory structure

All the files and folders are under /root directory itself. Let's take a look at the files and directories. This is for reading and familiarizing yourself with the playbooks and files we are going to use. No task to be performed in this section



- File `ansible.cfg`
  - Ansible configuration file where you can set ansible environment variables, for more information refer to link http://docs.ansible.com/ansible/intro_configuration.html
- File `host_file`
  - This file is the ansible inventory file, which stored information about the host(s) that we want to run the playbook against, and variable information pertaining to those hosts. For more information about the inventory file refer to link http://docs.ansible.com/ansible/intro_inventory.html#inventory
  - The host file is specific to your environment
  - Sample `host_file` for the dCloud environment

```
[iworkflow]
198.18.128.135

[iworkflow:vars]
username=admin
password=C1sco12345

[apic]
198.18.133.200
```

(continues on next page)

(continued from previous page)

```
[apic:vars]
username=admin
password=C1sco12345
```

- Directory `playbooks` – This directory contains
  - All the playbooks we are going to run in this lab
    - `iworkflow_setup.yaml` – Configure setting on iWorkflow
    - `aci_tenant_setup.yaml` – Create a tenant and related parameters on APIC
    - `logical_device_cluster.yaml` – Create a logical device cluster on APIC (this enabled communication of APIC with BIG-IP)
    - `service_insertion.yaml` - Configure service insertion on APIC
    - `aci_delete_service.yaml` – Clean up of the configuration done on APIC
  - The variable file which we are going to edit to customize it to our needs
    - This is a sample input to the variable file, you can modify it to fit your environment

bigip_ip	`198.18.128.130`
bigip_username	`admin`
bigip_password	`C1sco12345`
bigip_hostname	`bigip1.dcloud.cisco.com`
iworkflow_ip	`198.18.128.135`
iworkflow_username	`admin`
iworkflow_password	`C1sco12345`
tenant_name	`Demo`
context_name	`{{tenant_name}}_ctx1`
app_profile_name	`App_profile`
provider_bd_name	`{{tenant_name}}_BDApp`
provider_ip	`192.168.10.220`
provider_mask	`24`
provider_epg_name	`prov_EPG_app`
consumer_bd_name	`{{tenant_name}}_BDWeb`
consumer_ip	`10.10.10.220`
consumer_mask	`24`
consumer_epg_name	`cons_EPG_web`
contract_name	`web2app-demo-contract`
filter_name	`{{contract_name}}_filter`
subject_name1	`http`
subject_name2	`https`
iworkflow_servicetemplate_name	`SimpleHTTP`
devicePackage_name	`dCloudConnector`
downloaded_devicePackage_name	`F5DevicePackageSimple`
logicalDeviceCluster_name	`StandaloneBIGIP`
SGtemplate_name	`SimpleHTTP_ServiceGraphTemplate`

Continued on next page

Table 1 – continued from previous page

external_selfip	`10.10.10.120`
external_netmask	`255.255.255.0`
internal_selfip	`192.168.10.120`
internal_netmask	`255.255.255.0`
vip_ip	`10.10.10.100`
vip_port	`80`
poolMember_ip	`192.168.10.140`
lb_method	`round-robin`

- Directory `aci_posts`
  - This directory has all the aci posts we are going to execute on the APIC
  - Each post is a j2 (jinja2) template file. This template file contains variables which are going to be substituted at run time from information present in the variable file. The XML file then created after the substitution will be then run on the APIC
- JSON blob for creating a service template on iWorkflow
- Directory `library`
  - This contains the python files which are responsible for running code for modules. For this lab we have the one aci module `aci_rest.py` which will be used to run the posts on the APIC

3.2 Module 1: L4-7 Services with Cisco APIC and BIG-IP

3.2.1 Lab 1: Customize files to fit the environment

Let's take a look at the `host_file` and `variable_file` and fill it out.

3.2.2 Lab 2: Executing the playbooks

iWorflow

Let's login to iWorkflow and have a look at the configuration before we run the playbook

Let us first execute the playbook on iWorkflow. This playbook will perform the following tasks

- Discover device
- Create a cloud connector
- Create a service template – Parameters than are tenant editable on this template are
  - Virtual IP
  - Virtual Port
  - Load balancing method
  - Pool members

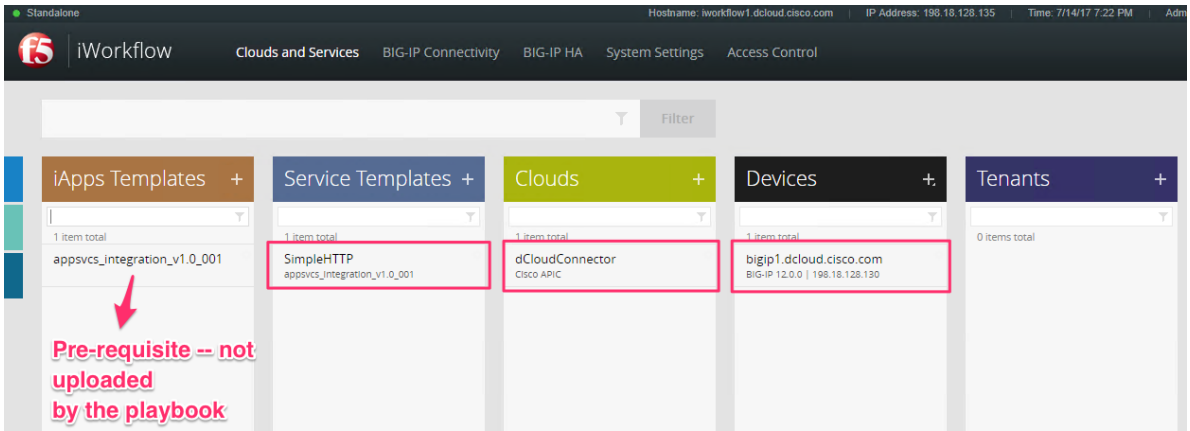
To execute the playbook run command:

1. SSH to the "Tools" host

2. Go to the `/root` directory
3. `ansible-playbook --step playbooks/iworkflow_setup.yaml`

Note: The playbook will be run step by step, after the first task device discovery, make sure before you go to the next step the device is discovered correctly and the BIG-IP is in a healthy state

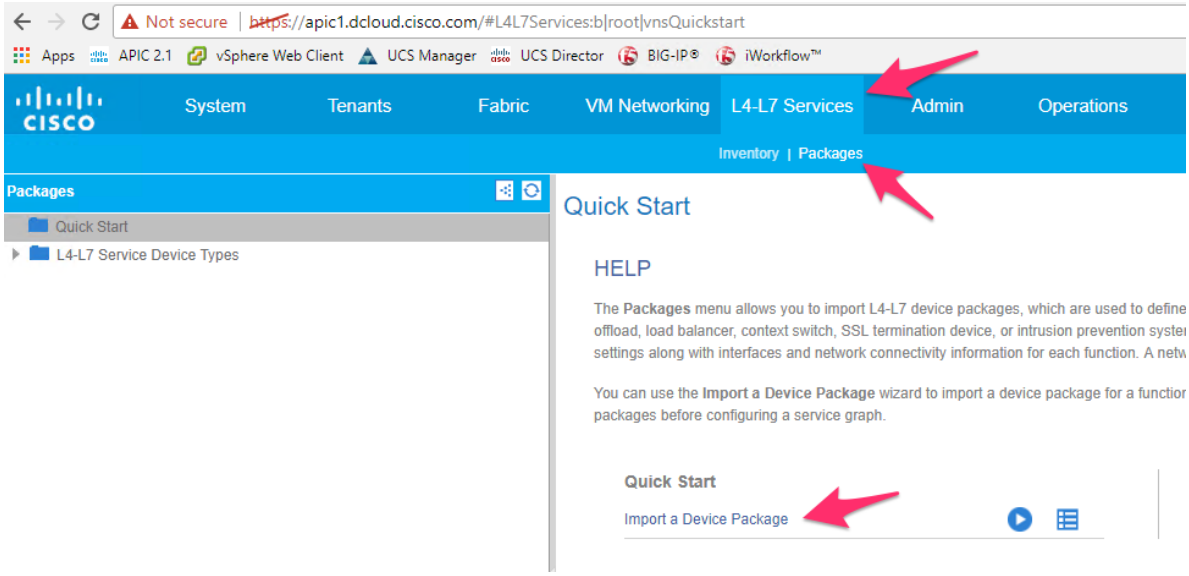
4. The following gets created on iWorkflow after playbook(`iworkflow_setup.yaml`) execution



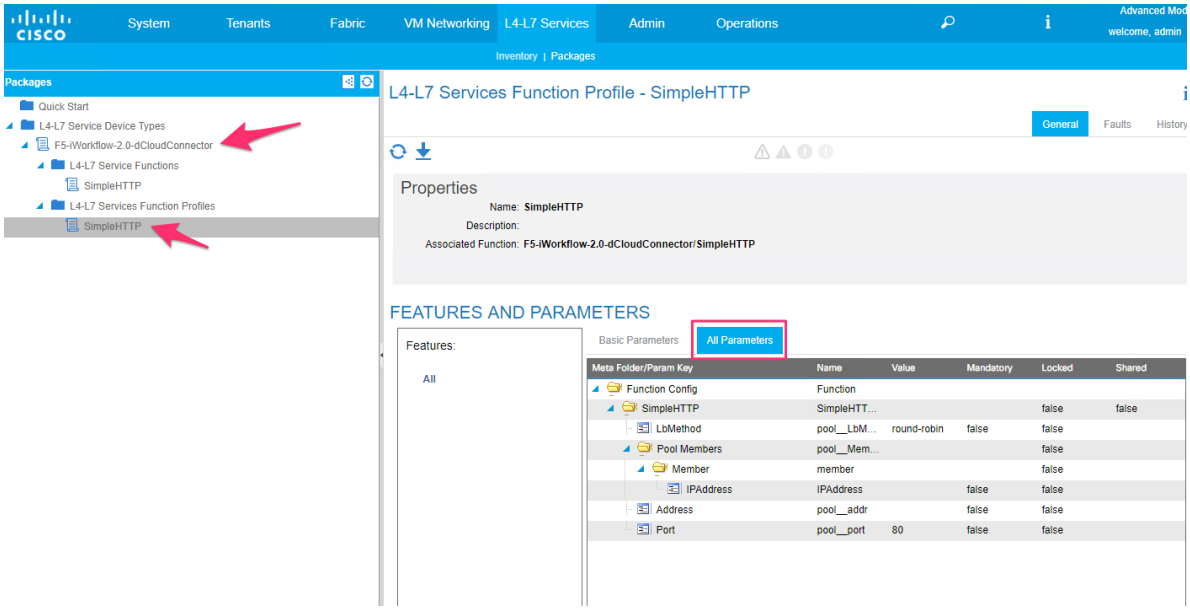
5. Once executed the playbook downloads the device package to the playbooks directory. Open WinSCP again and move this downloaded device package to desktop on your dcloud environment. Name of the device package is picked up from the variable file**

Manual step to upload Device Package to APIC

1. Go to APIC UI, login with `admin/C1sco12345`
2. Click on L4-L7 services->Packages->Import a device package



3. Click on Browse and then select the device package present on the desktop
4. Once uploaded, you can view the device package contents on the left-hand side of the pane



APIC

Let's login to APIC and have a look at the configuration before we run the playbook

Let us now execute the playbooks on APIC.

1. Log back into the 'Tools' host, go to the `/root` directory

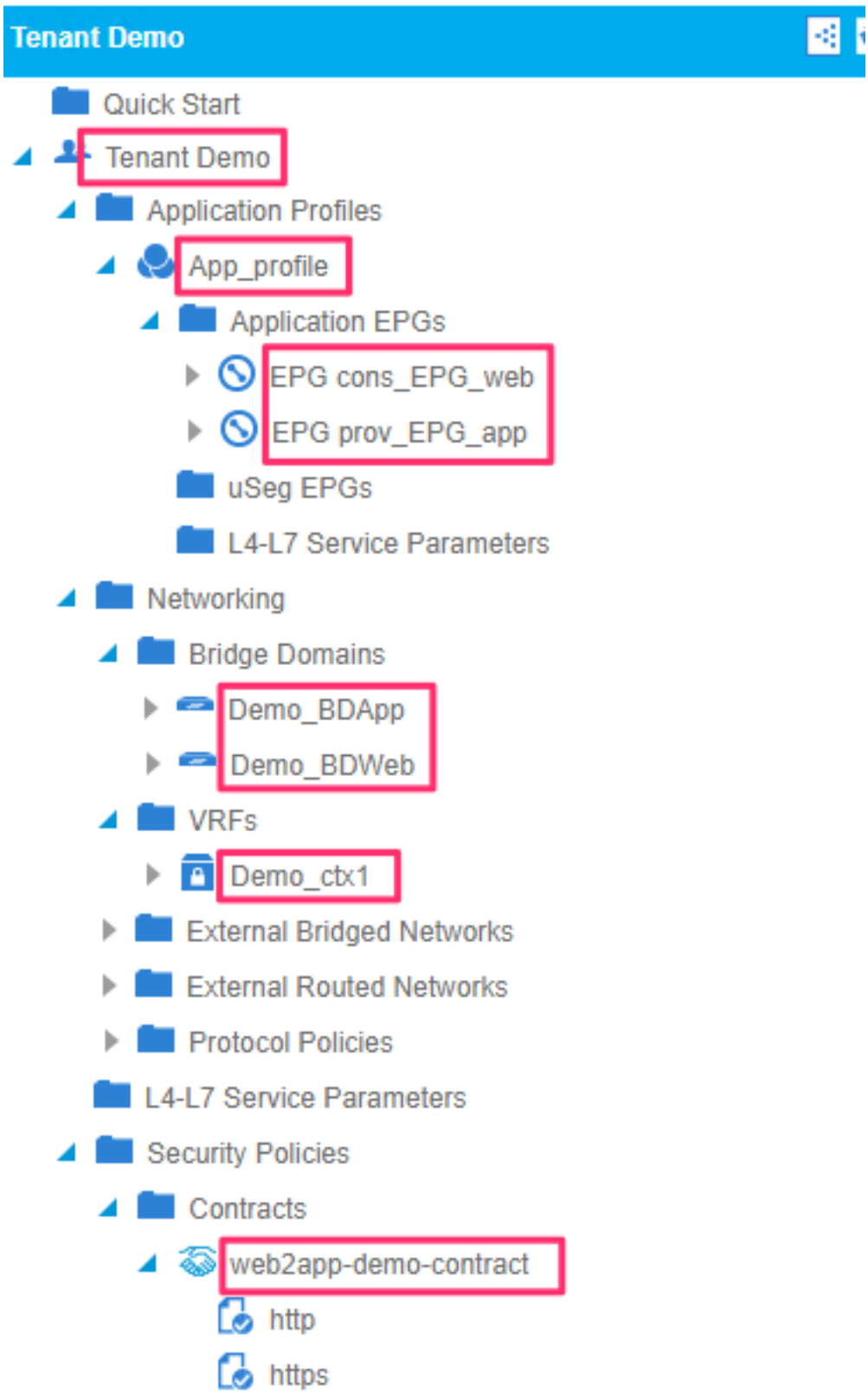
   Playbook `aci_tenant_setup.yaml` – this playbook will perform the following tasks

   - Create a tenant
   - Create a Private Context
   - Create two bridge domains
   - Create an application profiles
   - Create two EPG (End Point Groups)
   - Create a contract

2. To execute the playbook run command

   ```
   ansible-playbook --step playbooks/aci_tenant_setup.yaml
   ```

3. The following gets created on APIC after playbook (`aci_tenant_setup.yaml`) execution



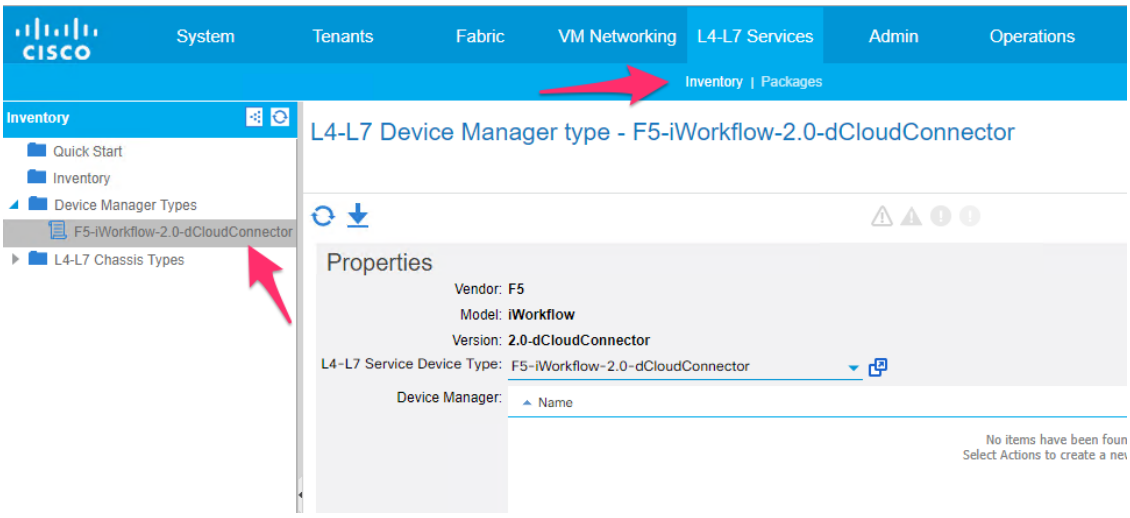
4. Playbook `logical_device_cluster.yaml` – this playbook will perform the following tasks
   - Create a device manager type

- Create a device manager in tenant common
- Create a logical device cluster in tenant common

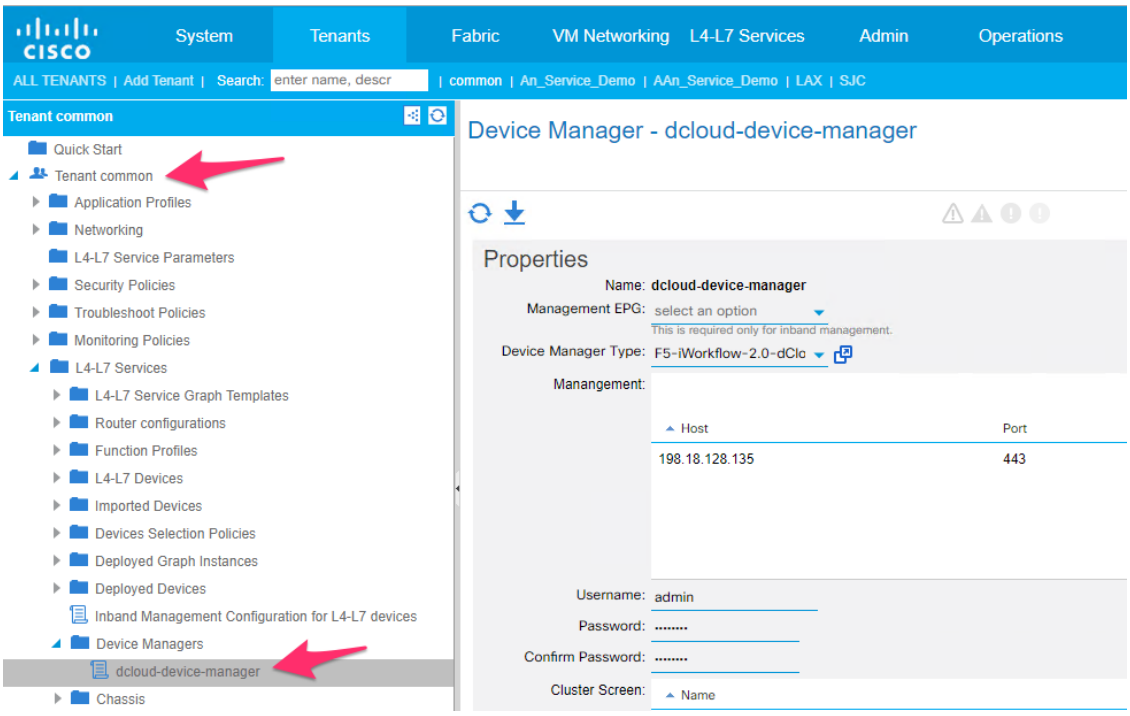
5. To execute the playbook run command

```
ansible-playbook --step playbooks/logical_device_cluster.yaml
```

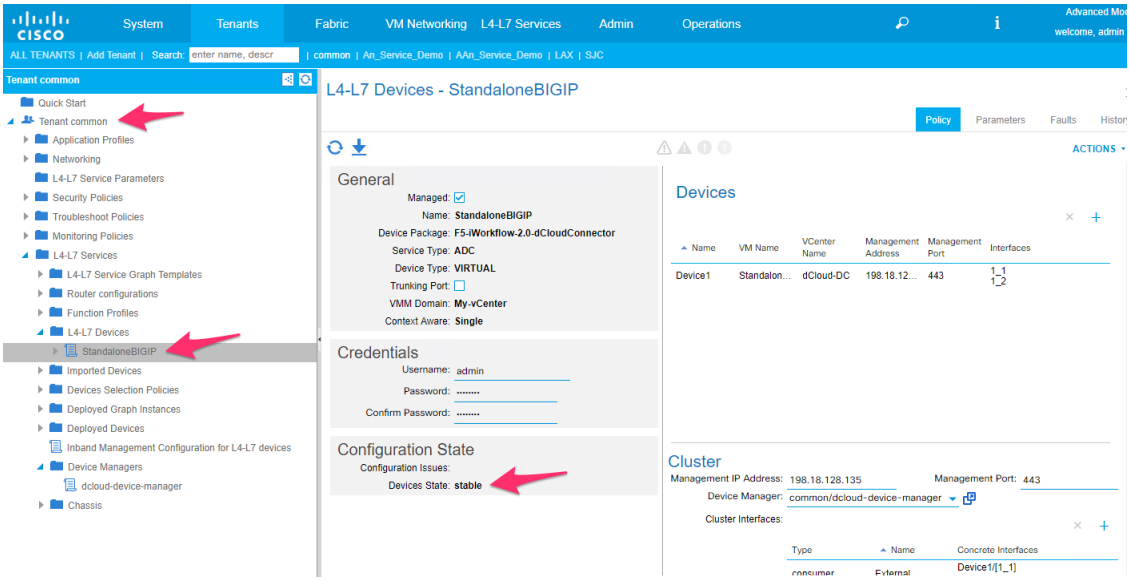
6. The following gets created on APIC after playbook (`logical_sevice_cluster.yaml`) execution
   1. Device Manager Type under L4-L7 services->Inventory->Device Manager Types



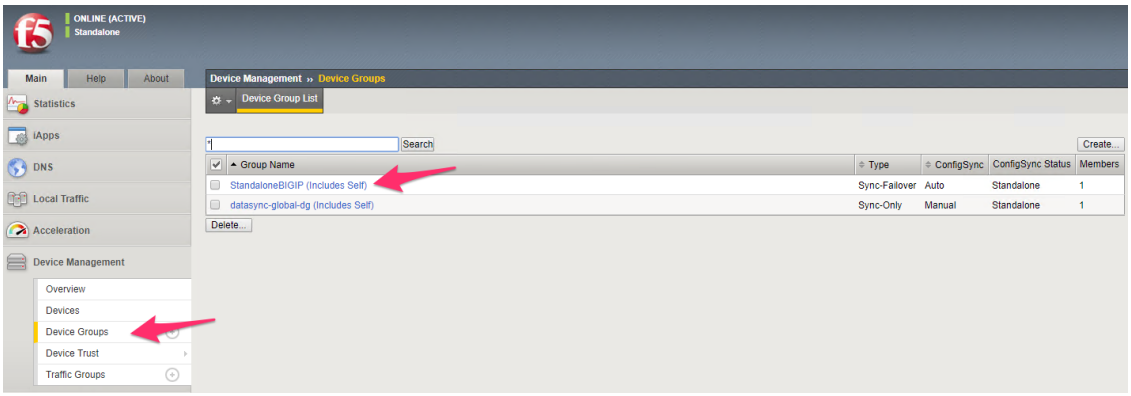
   2. Device Manager under Tenant common ->L4-L7 services->Device Managers



   3. Logical device cluster under tenant common -> L4-L7 Devices. **Make sure before proceeding to the next step that your logical device cluster is in 'Stable' state**



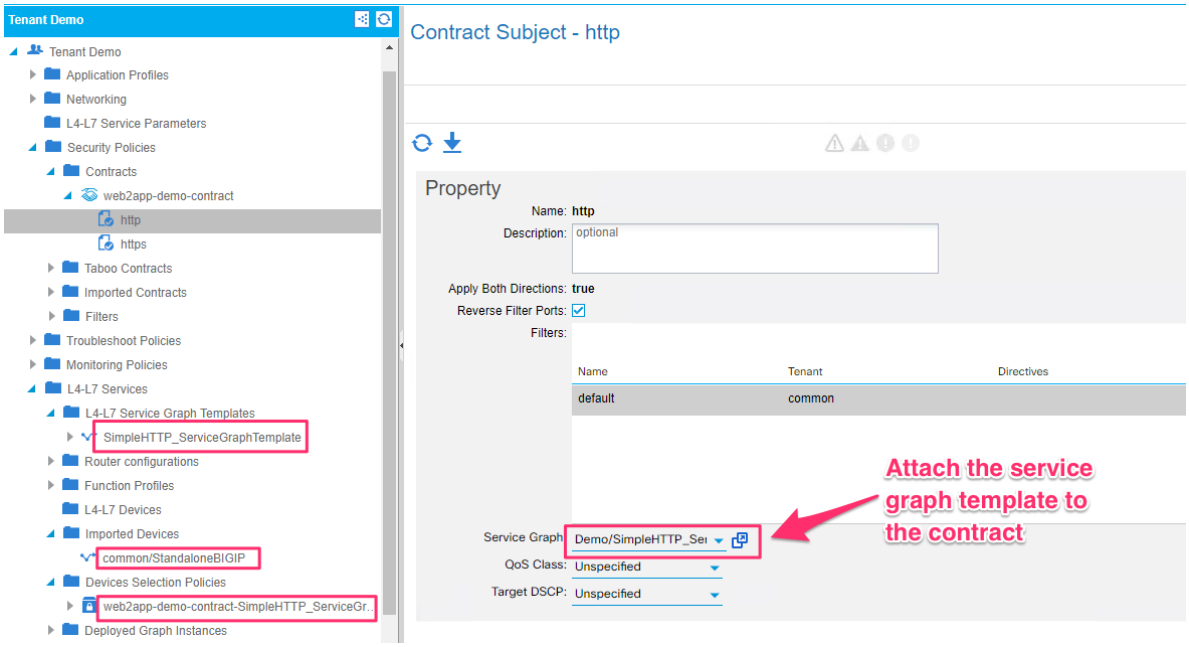
4. On the BIG-IP a device group will be created which has the same name as that of the logical device cluster



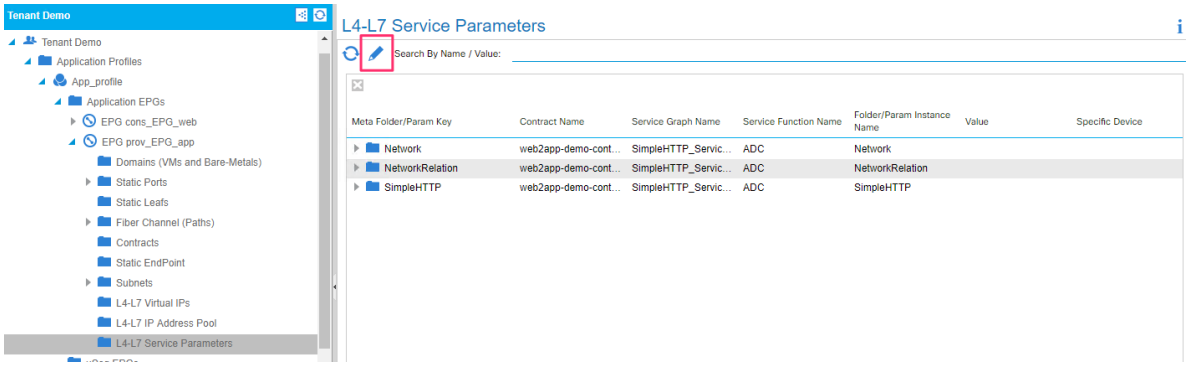
7. Playbook `service_insertion.yaml` - this playbook will perform the following tasks
   - Export the logical device cluster from tenant common to user tenant
   - Create a service graph template
   - Assign L4-L7 BIG-IP parameters (VIP, Port etc.) to the graph
   - Create a device selection policy
   - Then attach the service graph template to the contract
8. To execute the playbook run command

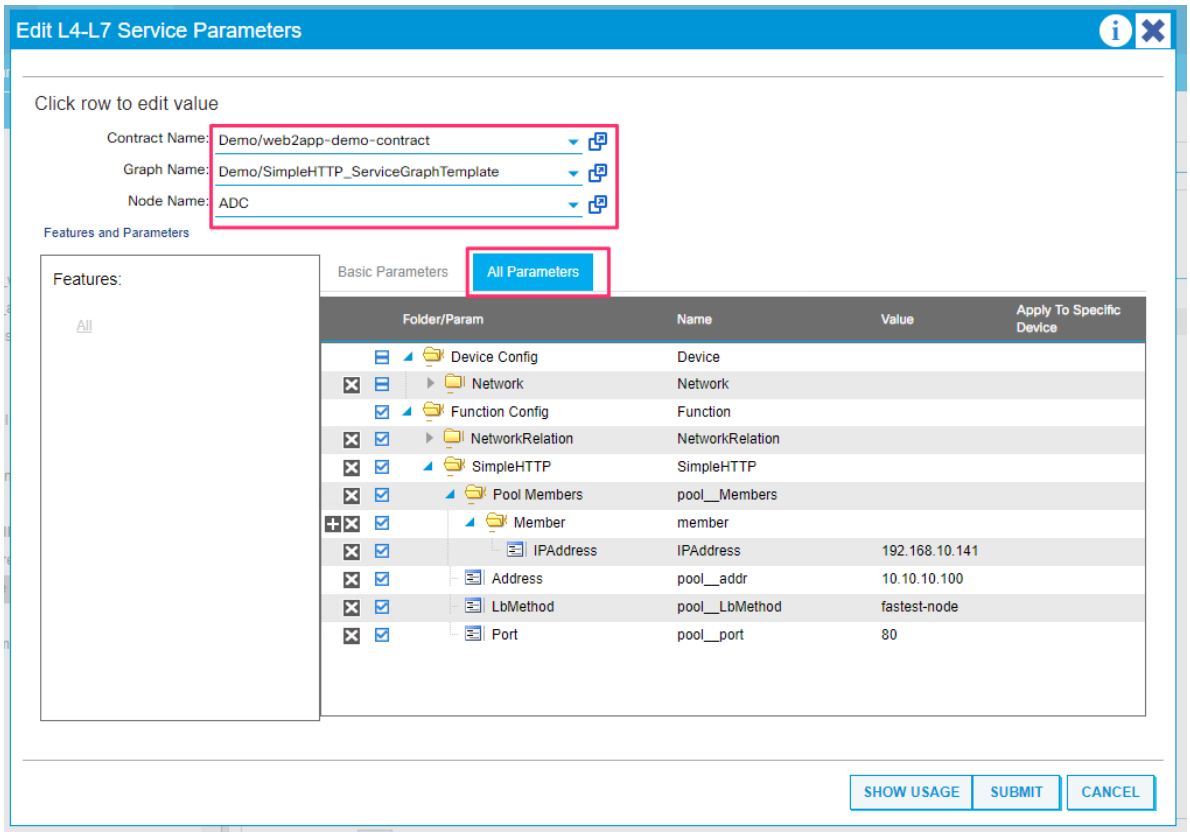
   ```
   ansible-playbook --step playbooks/service_insertion.yaml
   ```

9. The following gets created on APIC after playbook (`service_insertion.yaml`) execution



10. You can view the BIG-IP parameters that get configured under provider EPG. Click on the pencil edit button, select the appropriate graph/contract and node. Click on the 'all parameters' tab to view all the details

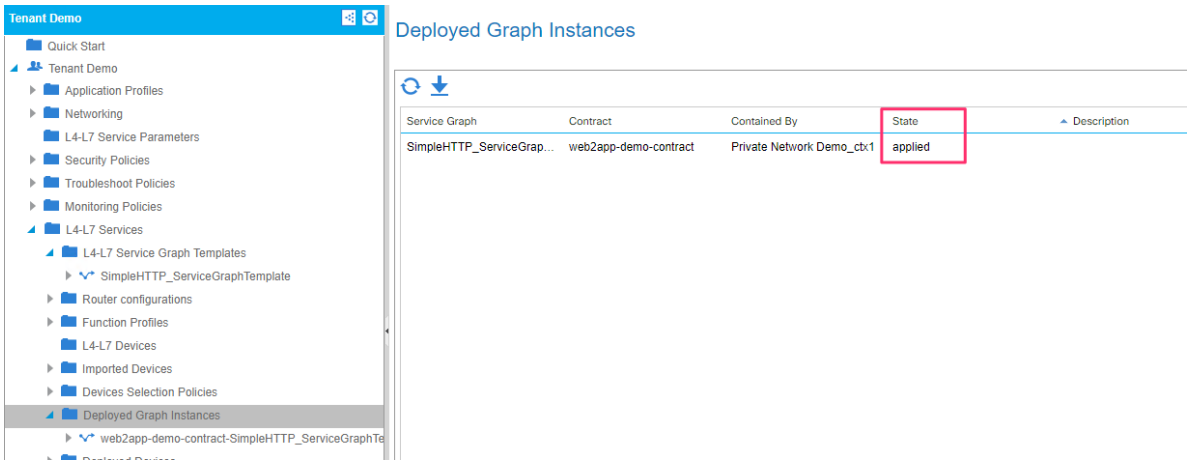




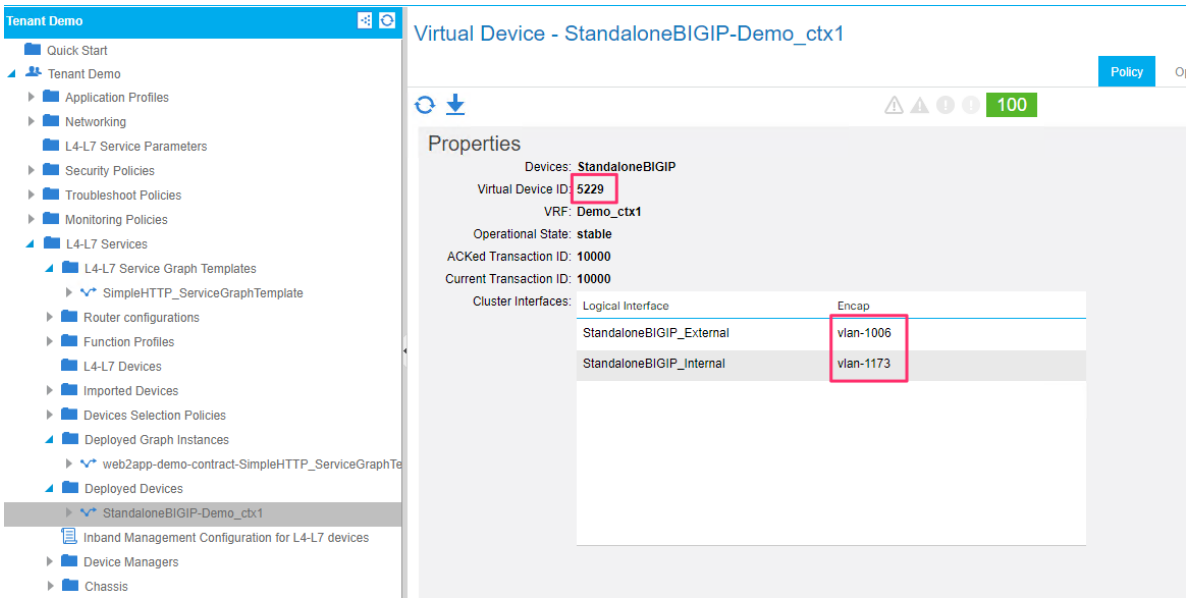
Verify

Verify successful deployment of network and application parameters on the APIC, iWorkflow, BIG-IP

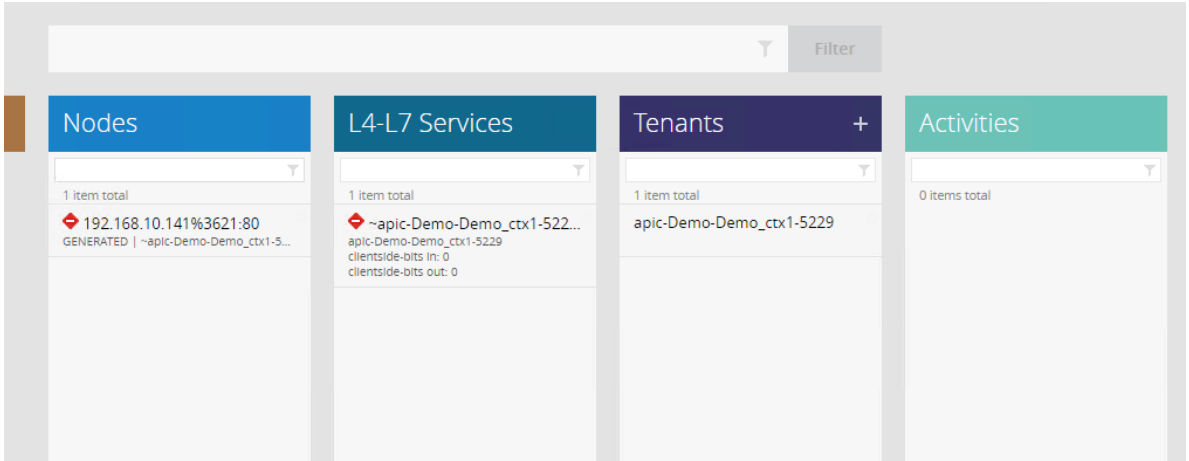
1. On the APIC make sure the graph is deployed and the state is 'applied'



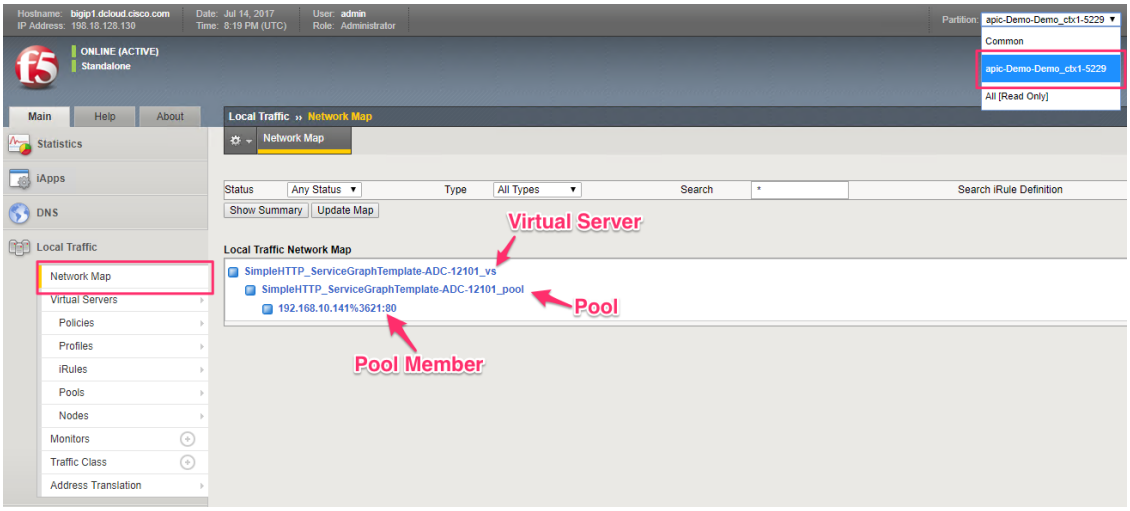
2. View the deployed devices tab and take node of the Virtual device ID. This will be the identified on the BIG-IP with which you can associate the partition created on the BIG-IP to the graph deployed on the APIC. Also keep note of the VLAN tags



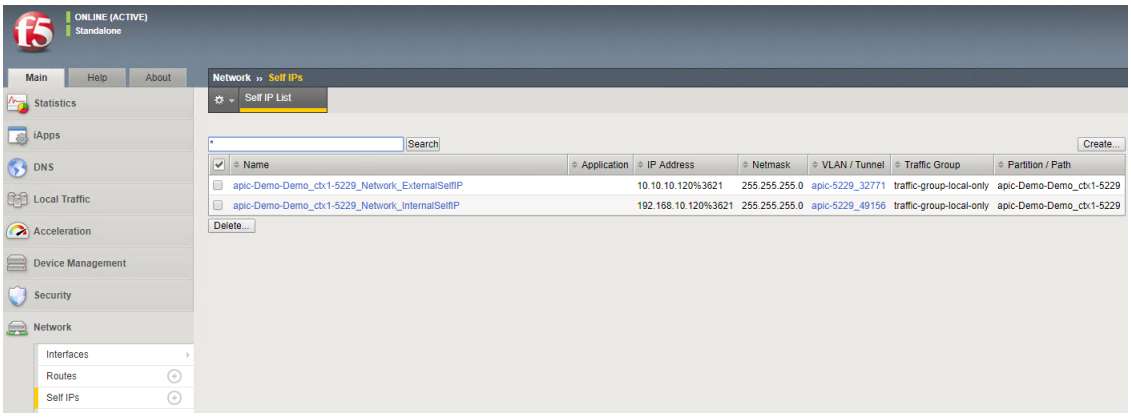
3. On the iWorkflow, make sure there is a
   - Tenant created which will map to a BIG-IP partition
   - A L4-L7 service which will map to the virtual server configured on the BIG-IP
   - Nodes are created which map to the node members created on the BIG-IP

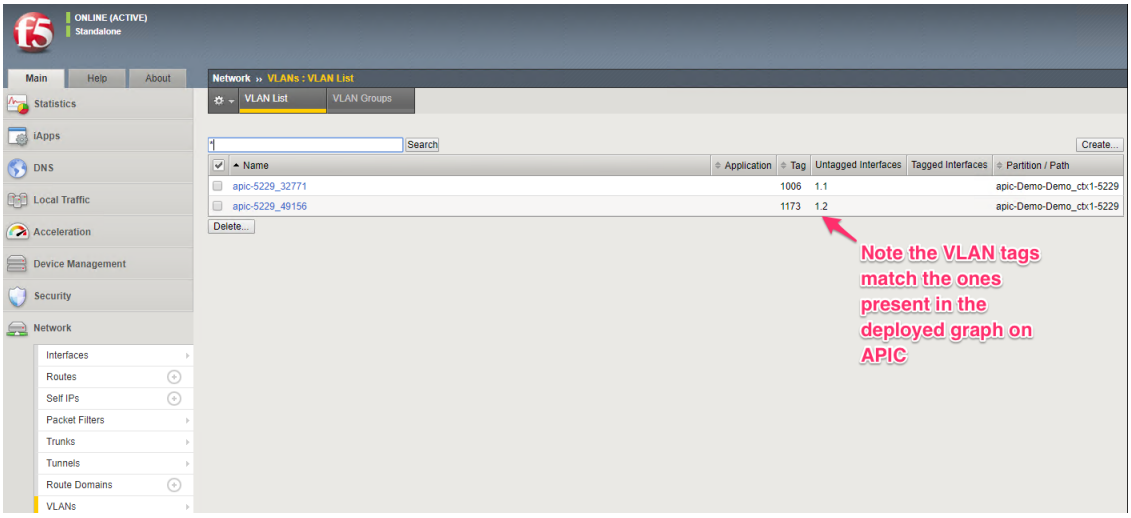


4. On the BIG-IP make sure a partition is created (note the partition is referencing the Virtual ID generated by APIC).
   - Click on 'Network Map' to get a unified view of the objects deployed on the BIG-IP. To see individual objects, click on the appropriate tab from the left hand pane



- To view network related parameters, click on the 'Network' tab and then view the Self IP's and the VLAN information. The Self IP information is user driver (part of the service graph). The VLAN information is dynamically generated by APIC which is configured on the BIG-IP





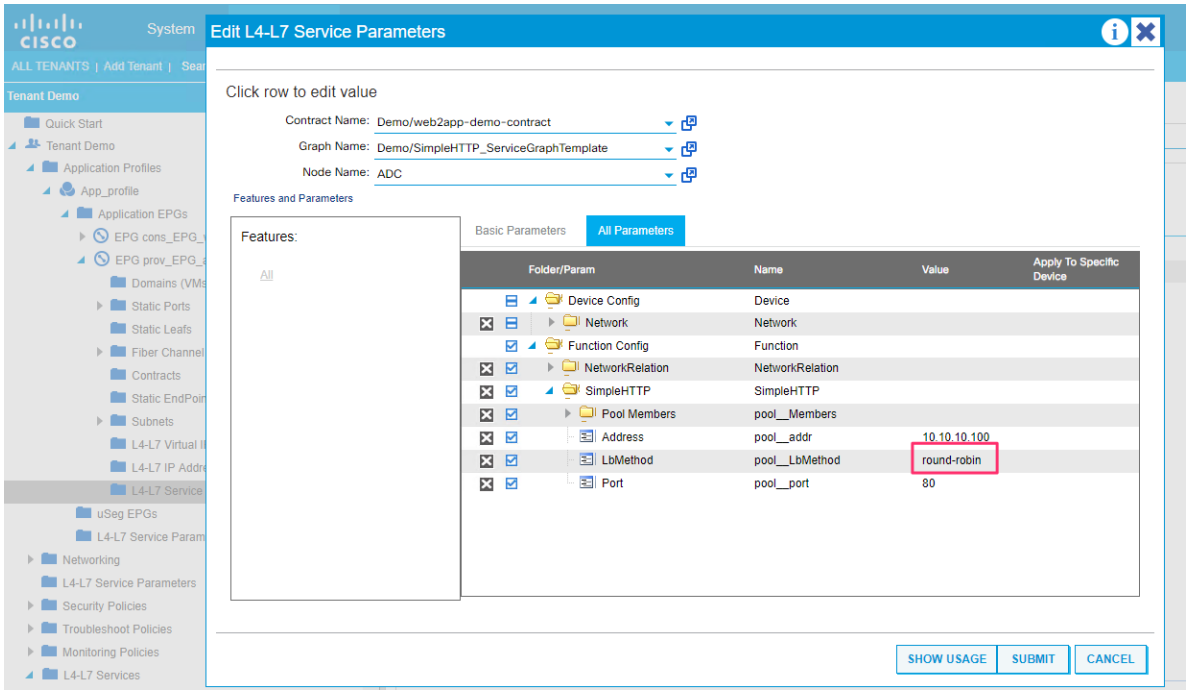
3.2.3 Lab 3: Making modifications to the service graph

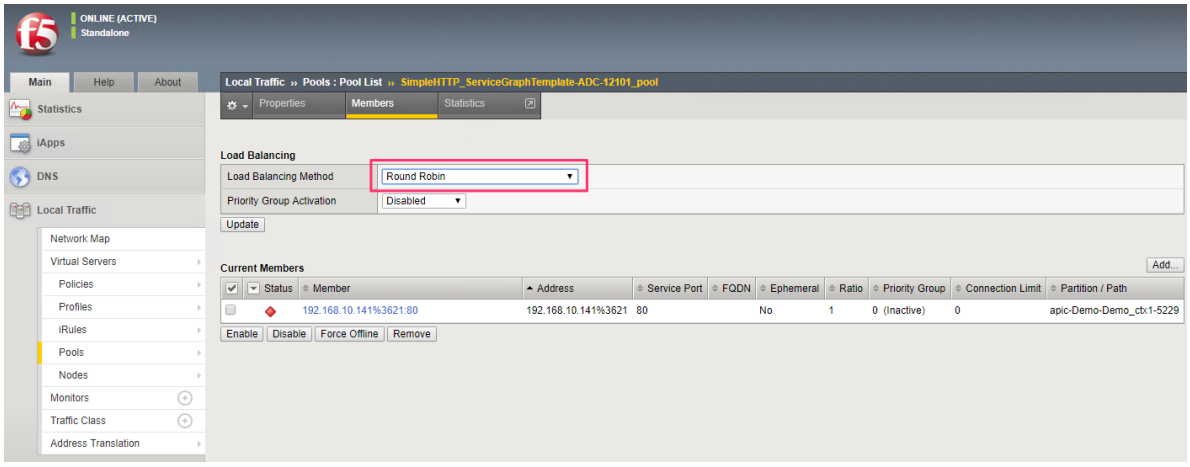
Playbook `modify_parameters.yaml` - this playbook will perform the following task

- Changes the load balancing method to the desired load balancing method (input taken from the variable file)

1. Open the variable file placed under `/root/playbooks/variable_file.yaml` and change the `lb_method` parameter from `round-robin` to `fastest-node`

   Before modification:

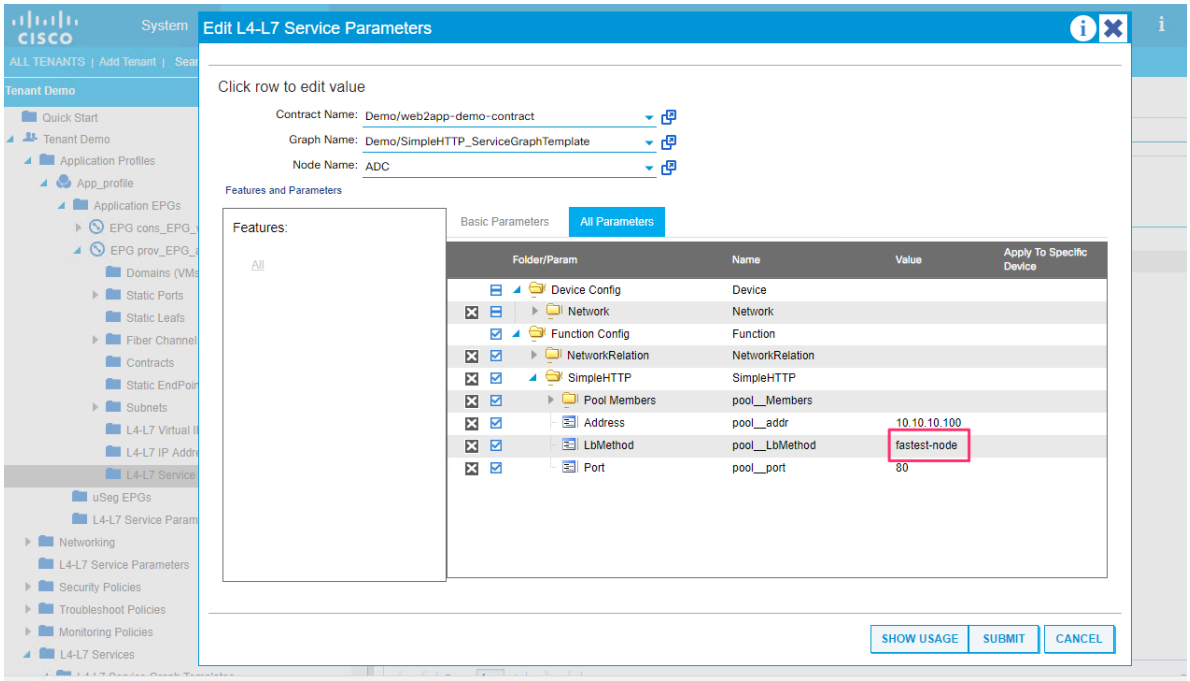


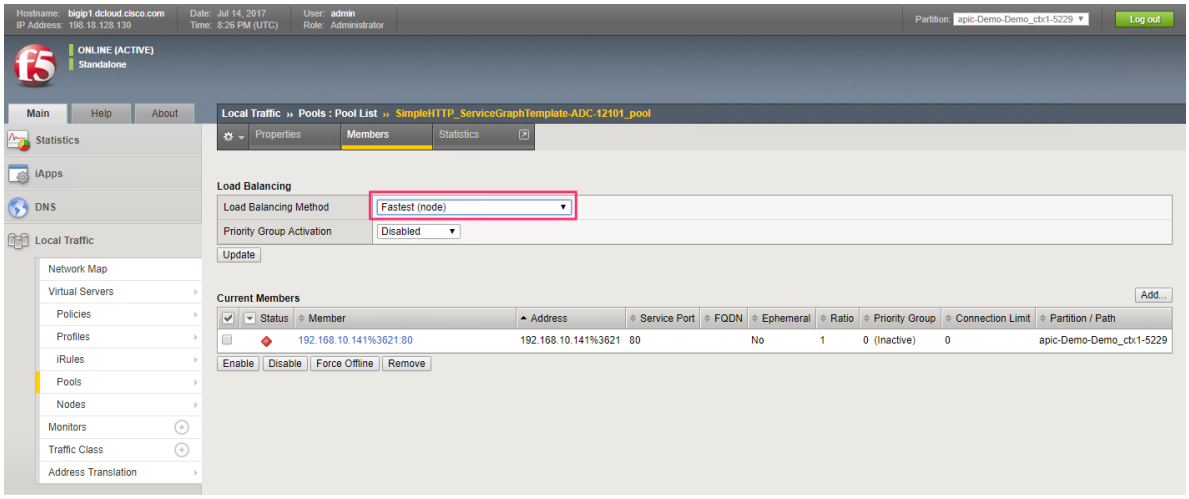


2. To execute the playbook run command

   `ansible-playbook --step playbooks/ modify_parameters.yaml`

   After running the playbook for modification:





3.2.4 Lab 4: Deleting the service

Playbook `aci_delete_service.yaml` - this playbook will perform the following tasks

- Detach the service graph from the contract
  - This will delete the partition created on the BIG-IP (thus deleting all the objects that belong to that partition)
- Delete the device selection policy
- Delete the BIG-IP parameters which are present under the provider End Point Group (EPG). Remove the provided as well as consumed contracts from the EPG's
- Delete the service graph template
- Delete the contract

- Delete the logical device cluster
  - This will delete the device group that is created on the BIG-IP
- Delete the device manager from tenant common
- Delete the device manager type under L4-L7 Services

1. To execute the playbook run command

   ```
   ansible-playbook --step playbooks/ aci_delete_service.yaml
   ```

   After execution of this playbook, the BIG-IP will be in a clean state. There will be no partition on the BIG-IP pertaining to the service graph and there will be no device group pertaining to the logical device cluster