DEPLOYMENT GUIDE

Private Cloud Solution Package for Cisco Networking

June 26, 2017

Contents

Contents, continued

Introduction

Today's software-defined economy requires businesses to move faster than their competitors. Speed and agility are critical to keeping up with competitive demands for new applications, as well as maintaining existing infrastructure. Because an inability to scale and service networks can lead to escalating costs and increased time-to-service, many IT groups deploy private clouds to help them respond aggressively to business needs.

F5 and Cisco have partnered to provide a full-stack, end-to-end software-defined networking and policy-driven solution to accelerate the journey to the cloud. By integrating the F5® BIG-IP® platform, Cisco® Application Centric Infrastructure (Cisco ACI™), and F5 iWorkflow™, the two companies offer a market-leading solution that provides automation and orchestration up and down the stack—from layer 2 through layer 7.

Optimizes Applications with Cisco's Networking Architecture

Combining the advantages of the F5 and Cisco deployment models, organizations can deploy versatile, elastic network and application services—ultimately leading to quicker and more successful application rollouts. F5 and Cisco share a fabric-based approach that lets customers use their choice of physical, virtual, and cloud solutions to provide an environment that best suits their needs. This integrated solution enhances Cisco environments with comprehensive L2–7 policy controls to address application performance, scale, security, and orchestration.

The adoption of Cisco ACI with the Application Policy Infrastructure Controller (APIC) continues to gain traction in the market. Since the APIC 1.2 release in 2016, Cisco has streamlined how ADCs are connected to the fabric. The F5 Dynamic Device Package with F5 iWorkflow for Cisco APIC uses programmability and orchestration APIs, which enable customers to configure application policies and requirements for F5 appliances across L2–7 fabrics. This ensures that applications receive the services and resources they require throughout the network, while also enabling organizations to automate systems for further efficiency and cost savings.

Networking for Cloud

With F5's industry-leading architecture, organizations can deploy multi-tenant solutions in their private cloud leveraging F5 virtual appliances, cloud ready iSeries hardware platforms across L2–7 fabrics, and Cisco ACI multi-tenancy capabilities. With the ability to configure application policies throughout L2–7 for each tenant applications, enterprises have granular control over how resources are deployed and prioritized to support software-defined networking. This frees IT teams from tying specific devices or resources to individual applications, while preserving the ability to isolate services if needed for compliance or other business requirements.

Accelerate Application Deployment with Cisco APIC and F5 iWorkflow

Some analysts believe that OPEX costs are doubling every eight years. This data is based on historical trends, and doesn't necessarily take into consideration the forthcoming explosions in applications and data resulting from technological shifts like the Internet of Things (IoT). Even so, it's no wonder that almost every study done on IT budgets pegs operating expenses—the "keep the lights on and apps running" kind of operating expenses—at upward of 70%

of the total budget. Something, obviously, must change, and change radically. Cloud, DevOps and SDN all point organizations in the same direction: operationalization through automation, orchestration, and ultimately, integration via open, standards-based APIs and protocols.

That's the goal of Cisco's Application Centric Infrastructure (ACI) strategy, which seeks to address the challenges in scaling networks and services not only from a technology perspective, but from a people perspective. One reason that so much of IT budgets is spent on operations is that the configuration of the network is spread across tens and hundreds and sometimes thousands of myriad network devices. From layer 2 to layer 7, organizations use a veritable cornucopia of network and application services to deliver the applications upon which business relies.

Deploying an application can take days, weeks, or months, because of the coordination required across not just the devices themselves (whether they are virtual or physical makes no difference as configuration is agnostic with respect to form factor), but across what are increasingly siloed IT teams: operations, security and networking.

While the magnitude of the tectonic shifts in technology today has never been more disruptive, organizations can't simply start over from scratch. This means implementing a hybrid model that can bridge the gap between the existing and the new. It's essential to insulate production applications from the massive disruption that comes with these seismic changes around the way organizations build and manage IT today. Private cloud and Colo networking are excellent options to bridge the gap as part of the hybrid approach.

The F5 private cloud solution package for Cisco Networking is that bridge. It's the abstraction layer that provides the capabilities of delivering yesterday's applications while enabling tomorrow's architecture. By integrating with Cisco ACI or Cisco Nexus 9000 Series Switches, F5 allows customers to operationalize the entire network and start migrating to the policy-based, application-driven network architectures so necessary to succeed in a software-defined economy—without compromising on the security, performance, or availability of both existing and new applications.

F5 Private Cloud Solution Package for Cisco Networking

The F5 private cloud solution package for Cisco Networking includes deployment scenarios that have been validated, certified, and documented by Cisco and F5, such as the following:

1. Cisco ACI Service Manager Mode (Managed)—Maintain L2–7 automation while providing operational flexibility with native management consoles experiences by integrating iWorkflow with APIC in Service Manager Mode.
2. Cisco ACI Network Policy Mode (Unmanaged)—Gain flexibility for the networking administrator to only configure the provider and consumer VLANs through the APIC management console while allowing the application administrator to orchestrate the F5 L4–7 polices via Ansible playbooks.
3. Cisco 9000 NX-OS (Standalone)—Allow for BIG-IP configuration of L4-7 polices in a Cisco Nexus standalone environment. The application administrator can also orchestrate the F5 L4–7 polices via Ansible playbooks

Using these models, organizations can deploy a BIG-IP multi-tenant private cloud with Cisco using orchestration via APIC and Ansible playbooks. The private cloud package represents deployment models of the most common scenarios found in existing network environments. The F5 solution validates these deployment models based on tests utilizing the Cisco ACI service manage mode with device package, unmanaged mode without a device package and Cisco Nexus 9000 standalone environments using Ansible playbooks for comprehensive L2–7 policy controls. This enables organizations to rapidly deploy the F5 private cloud solution for Cisco Networking, accelerating the migration of existing workloads to a private cloud utilizing BIG-IP i5800 ADC devices and deployment of a BIG-IP VE instance within an L2–7 fabric.

Implementing BIG-IP Local Traffic Manager

In these deployment models, BIG-IP users implement BIG-IP Local Traffic Manager (LTM) L4–L7 services through service insertion, unmanaged, and standalone architectures in a private cloud. The use case leverages standard F5 L4-7 load balancers, listeners, pools, members, monitors, and L7 policy and rules.

BIG-IP features tested include BIG-IP LTM standard virtual servers, client TLS decryption, server context re-encryption, HTTP profiles, multiple pools, cookie persistence, multiple iRules associations, and monitored pool members. Pool member state and virtual service statistics are collected through networking APIs.

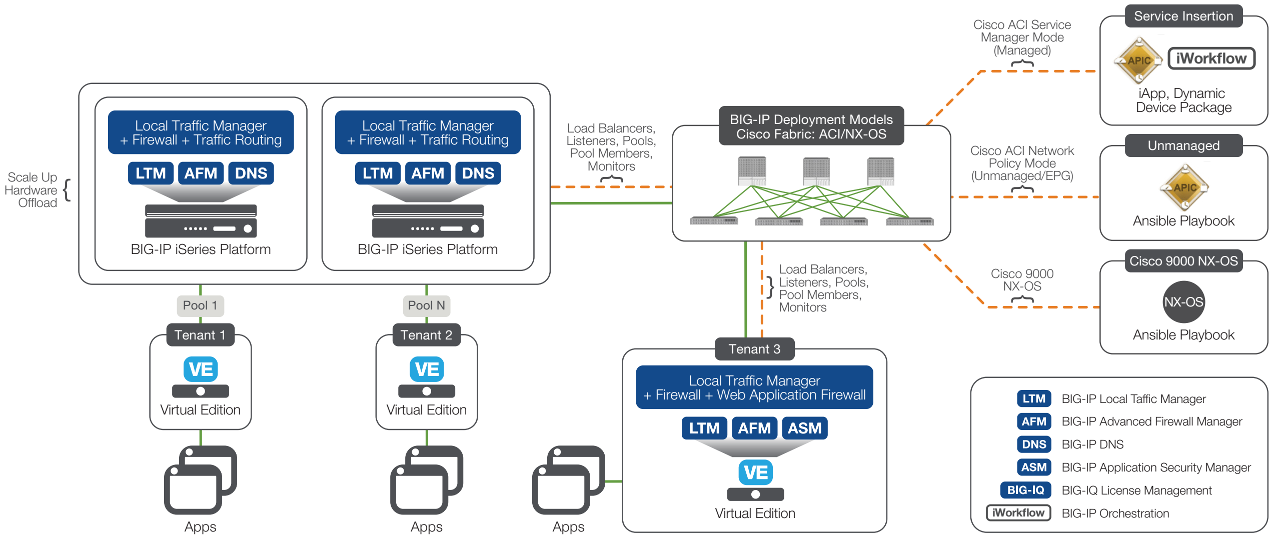

Scale Up Hardware Offload
Local Traffic Manager + Firewall + Traffic Routing
LTM
AFM
DNS
BIG-IP iSeries Platform
Local Traffic Manager + Firewall + Traffic Routing
LTM
AFM
DNS
BIG-IP iSeries Platform
Pool 1
Pool N
Tenant 1
VE
Virtual Edition
Apps
Tenant 2
VE
Virtual Edition
Apps
Apps
Load Balancers, Listeners, Pools, Pool Members, Monitors
BIG-IP Deployment Models Cisco Fabric: ACI/NX-OS
Load Balancers, Listeners, Pools, Pool Members, Monitors
Tenant 3
Local Traffic Manager + Firewall + Web Application Firewall
LTM
AFM
ASM
VE
Virtual Edition
Cisco ACI Service Manager Mode (Managed)
Service Insertion
APIC
iWorkflow
iApp, Dynamic Device Package
Cisco ACI Network Policy Mode (Unmanaged/EPG)
Unmanaged
APIC
Ansible Playbook
Cisco 9000 NX-OS
Cisco 9000 NX-OS
NX-OS
Ansible Playbook
LTM BIG-IP Local Taffic Manager
AFM BIG-IP Advanced Firewall Manager
DNS BIG-IP DNS
ASM BIG-IP Application Security Manager
BIG-IQ BIG-IQ License Management
iWorkflow BIG-IP Orchestration

Figure 1: F5 deployment model architecture

In Figure 1, the deployment models deliver the agility to deploy a multi-tier architecture using both the BIG-IP multi-tenant iSeries and BIG-IP VE ADCs. The BIG-IP hardware devices in the diagram below are cloud-ready i5800 ADCs, while the BIG-IP VE tenants are software ADCs, which utilize the F5 BIG-IQ Centralized Management license manager for manual licensing of the VEs and provisioning. The BIG-IP VE adds additional application-specific services for security. These additional polices can be enabled via Ansible playbooks.

Deployment Scenarios Documented in This Guide

This document is intended for use by network architects and engineers to aid in developing solutions for Cisco ACI and F5 L4-L7 service insertion and automation. This document discusses three deployment scenarios:

1. How to deploy F5 BIG-IP Service Insertion with the Cisco® Application Centric Infrastructure (Cisco ACI™) using a customized device package generated by F5 iWorkflow. F5 iWorkflow manages application service catalog life cycle; and thru its capability to dynamically generate device package, this service catalog can be represented in Cisco ACI environment.

2. How to deploy F5 BIG-IP with Cisco® Application Centric Infrastructure (Cisco ACI™) where Layer 2-3 configuration for F5 BIG-IP is configured through the Cisco ACI environment. Advanced application configuration is done directly on the BIG-IP using Ansible for orchestration.

3. How to deploy F5 BIG-IP with Cisco® Nexus 9000 Series and F5 BIG-IP both are configured using Ansible.

This document defines the deployment recommendations for Private Cloud Solution Package for Cisco Networking validated and certified by Cisco. The document has been certified with the following hardware and software components and versions:

Solution Package	Medium Size
F5 iSeries	i5800 x 2
iSeries SW Modules	LTM, DNS, AFM
Virtual Edition – 200M	8
Virtual Edition – 25M	8
Virtual Edition Software Modules	LTM, ASM, AFM
Orchestration	Cisco APIC and Ansible
Services	40 Hour Engagement per deployment scenario
Support	Premium Support
Customer Documentation	• Solution Architecture • Deployments Guide

Table 1: Private Cloud Solution Packages for Cisco Networking: iSeries + VE + SW Solution-Engineered, Tested, and Certified

Components of Offering	Quantity	Detail
I5800 Better	i5800 x 2	Provides desired network packaging of LTM, DNS and AFM
200M “App Services” VE	8	Provides desired tenant packaging of LTM + ASM +AFM. This packaging is only available within the Private Cloud Offering
25M “App Services” VE	8	Provides desired tenant packaging of LTM + ASM +AFM. This packaging is only available within the Private Cloud Offering
BIG-IQ VE “S”	2	Included free as part of offering - BIG-IQ VE License Manager is needed for the VE licensing. The full Centralized Management is not needed.
iWorkflow VE “Max” –	3	Included free as part of offering
Premium Support	For all of the above	
Consulting	40 hours	1 week of consulting/scoping

Table 2: Private Cloud Solution Package for Cisco Networking: i5800M Offering and Build of Material

Product Validated	Version	Detail
F5 iWorkflow	2.2	Non-clustered
iApp	F5.http	Download
BIG-IP	12.1.2	Compatibly Matrix
Cisco APIC	2.2(2i) a.k.a Danube MR1	
Cisco NX-OS Standalone	7.0.(3)	
Ansible	2.3	

Table 3: Tested and certified software components and versions

Cisco APIC Overview

Main Characteristics of Cisco ACI

- Simplified automation by an application-based policy model
- Centralized management, automation, and orchestration
- Mixed-workload and migration optimization
- Secure and scalable multitenant environment
- Extensibility and openness: open source, open APIs, and open software flexibility for development and operations (DevOps) teams and ecosystem partner integration
- Investment protection (for both staff and infrastructure resources)

The APIC acts as a central point of configuration management and automation for L4-L7 services and tightly coordinates service delivery, serving as the controller for network automation (Figure 2). A service appliance (device) performs a service function defined in the service graph. One or more service appliances may be required to render the services required by a service graph. A single service device can perform one or more service functions.

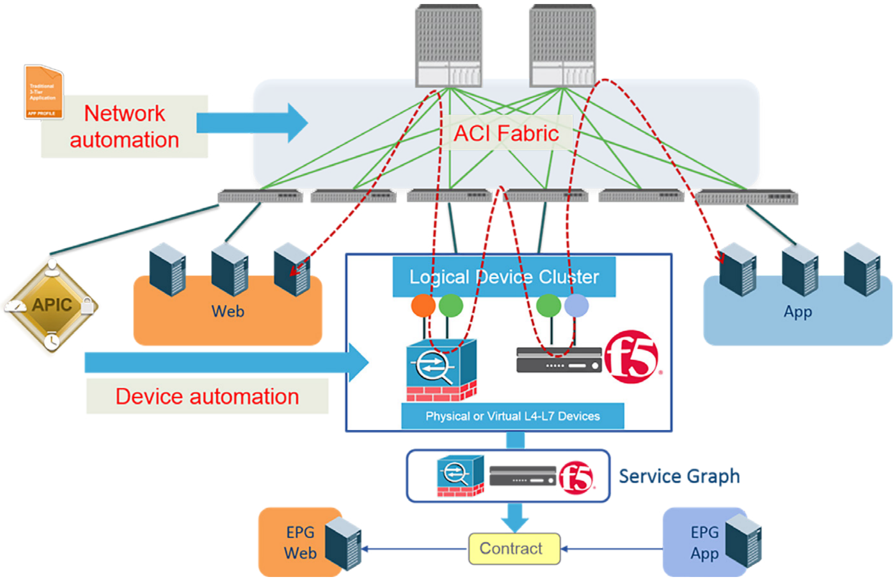


Figure 2: Cisco APIC: Central Point of Configuration Management and Automation

The APIC enables the user to define a service graph or chain of service functions in the form of an abstract graph: for example, a graph of the web application firewall (WAF) function, the load-balancing function, and the network firewall function. The graph defines these functions based on a user-defined policy. One or more service appliances may be needed to render the services required by the service graph. These service appliances are integrated into the APIC using southbound APIs built into a device package that contains the XML schema of the F5 device model. This schema defines the software version, functions provided by BIG-IP LTM (SSL termination, Layer 4 server load balancing [SLB], etc.), parameters required to configure each function, and network connectivity details. It also includes Python scripts that map APIC events to function calls to BIG-IP.

The joint solution uses the Cisco Nexus® 9000 Series Switches (Figure 3):

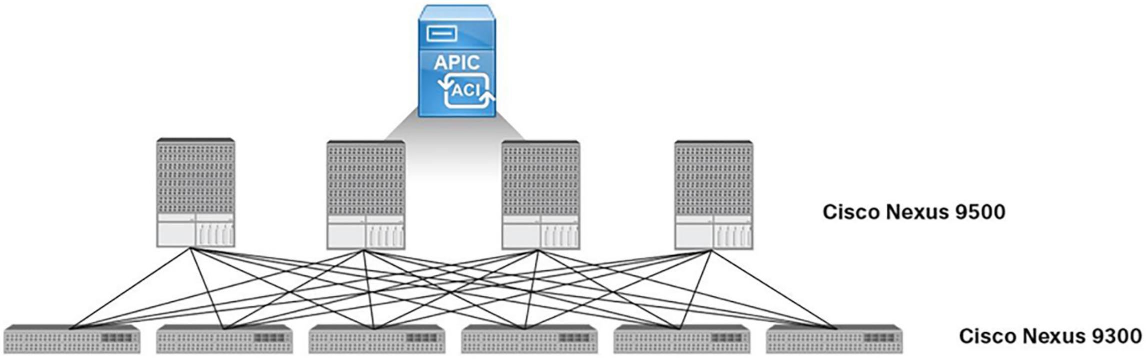


Figure 3: Cisco ACI Solution

- The solution described in this document requires the following components:
- Spine switches: The spine provides the mapping database function and connectivity among leaf switches.
- Leaf switches: The leaf switches provide physical and server connectivity and policy enforcement.
- APIC: The controller is the point of configuration for policies and the place at which statistics are archived and processed to provide visibility, telemetry, application health information, and overall management for the fabric. The APIC is a physical server appliance like a Cisco UCS® C220 M3 Rack Server,

The designs in this document have been validated on APIC Version 2.2(2). Release notes.

Cisco ACI configuration reference.

More information about Cisco ACI hardware and software releases.

F5 iWorkflow

iWorkflow enables organizations to accelerate the deployment of applications and services while reducing exposure to operational risk. iWorkflow is a multi-tenant platform for deploying application delivery policies onto BIG-IP devices. Presented using services catalogues, iWorkflow tenants deploy highly-configurable, administrator-defined application services templates. Using these service templates (called F5 iApps®), you avoid operational delay, risk, and complexity while simplifying application delivery management.

F5 iWorkflow version 2.2 is being used in this deployment guide. F5 iWorkflow 2.2.0 Knowledge Center.

F5 iApps

F5 iApp™ is a powerful set of features in the BIG-IP system that provides a new way to architect application delivery in the data center, and it includes a holistic, application-centric view of how applications are managed and delivered inside, outside, and beyond the data center. The iApp template for HTTP applications acts as the single-point interface for building, managing, and monitoring these servers. For more information on iApp, see the White Paper, F5 iApp: Moving Application Delivery Beyond the Network.

The iApp template for HTTP will be managed via iWorkflow. Installing and configuring the iApp template for HTTP is covered later in this Guide (see iWorkflow Import F5 HTTP iApps).

F5 BIG-IP

F5's next-generation, cloud-ready Application Delivery Controller (ADC) platform provides DevOps-like agility with the scale, security depth, and investment protection needed for both established and emerging apps. The new F5® BIG-IP® iSeries appliances deliver quick and easy programmability, ecosystem-friendly orchestration, and record breaking, software-defined hardware performance. As a result, customers can accelerate their private clouds and secure critical data at scale while lowering TCO and future-proofing their application infrastructure.

The Advantages of F5 BIG-IP i5800 Hardware

The BIG-IP iSeries platform perfectly blends software and hardware innovations that balance the need for performance, scalability, and agility. The F5 TMOS® operating system provides total visibility, flexibility, and control across all application delivery services. With TMOS, organizations can intelligently adapt to the diverse and evolving requirements of applications and networks. Other unique or patented hardware and software innovations enable the BIG-IP iSeries platform to offer unmatched capabilities:

F5 TurboFlex™ optimization technology: Field-programmable gate arrays (FPGAs), tightly integrated with CPUs, memory, TMOS, and software, provide specific packet-flow optimizations, L4 offload, support for private cloud tunneling protocols, and denial-of-service (DoS) protection. These hardware optimizations not only improve performance, but free CPU capacity for other app delivery and security tasks as well. Only BIG-IP iSeries appliances feature TurboFlex performance profiles—user-selectable, pre-packaged optimizations that provide different performance characteristics depending on the business need.

- L4 offload enables unsurpassed throughput and reduced loads on software.
- Unique per-virtual-IP/application SYN flood protection ensures that if one application is under attack, others are not affected. Only F5 ADCs implement hardware-based SYN cookies in L4 and full proxy L7 mode.
- More than 100 types of DoS attacks can be detected and mitigated in hardware, hugely increasing the attack size that can be absorbed compared to software-only implementations.
- Network virtualization and overlay protocol processing (such as VXLAN and NVGRE tunneling) increases traffic processing capacity.
- UDP traffic processing increases throughput and reduces both latency and jitter, improving VoIP or streaming media performance.
- Best-in-market SSL performance accelerates SSL/TLS adoption by offloading costly SSL processing and speed key exchange and bulk encryption. BIG-IP iSeries solutions include hardware acceleration of ECC ciphers, enabling forward secrecy. In addition, the ability to achieve an SSL Labs A+ rating with one click reduces SSL configuration complexity and errors.
- BIG-IP platforms offer maximum hardware compression, enabling cost-effective offloading of traffic compression processing to improve page load times and reduce bandwidth utilization.

- Enterprise class SSD (solid state drive) technology on select BIG-IP platforms improves performance and reliability, saves power, and reduces heat generation and noise.
- Efficiency features include 80 Plus Platinum certified power supplies as well as front-panel touchscreen LCD management, remote boot and multi-boot support, and USB support.

Figure 4: BIG-IP i5800 ADC appliance

BIG-IP Virtual Editions

F5® BIG-IP® virtual editions (VEs) are virtual application delivery controllers (vADCs) that can be deployed on all leading hypervisors and cloud platforms running on commodity servers. BIG-IP VEs deliver all the same market-leading application delivery services -- including advanced traffic management, acceleration, DNS, firewall, and access management -- as F5 purpose-built hardware. VE software images are downloadable and portable between on-premises virtualized data centers, public, and hybrid cloud environments. With BIG-IP virtual editions and F5 BIG-IQ® Centralized Management solutions, you can rapidly provision consistent application services across the data center and into the cloud.

BIG-IP TMOS Specifications

The BIG-IP i5800 devices are installed with TMOS 12.1.2 licensed with LTM, AFM, and DNS services. The initial configuration should be implemented to match the deployment architecture Installation Guide for Active/Standby HA pair configuration. Instructions for configuring the iSeries are provided later in this Guide (see APIC: Create L4-L7 Device with F5 vCMP Guests). The TMOS Virtual Edition is a version 12.1.2, licensed with LTM, ASM, AFM, installed on VMware. Common certificate, keys, and profiles for LTM deployment are installed on the BIG-IP devices.

Centralized Management and Licensing with BIG-IQ

BIG-IQ Centralized Management is an intelligent framework for centrally managing F5 application delivery and security solutions. It provides a single pane of glass to manage and deploy all F5 devices, including central management for key BIG-IP modules including BIG-IP® Local Traffic Manager™ (LTM), BIG-IP® Application Security Manager™ (ASM), BIG-IP® Advanced Firewall Manager™ (AFM). Use BIG-IQ Centralized Management for device tracking; image and configuration backup; centralized reporting and alerting; BIG-IP VE license management; and to ensure consistent security and traffic management policies across your infrastructure.

BIG-IQ's VE license management enables you to automate large-scale virtual ADC deployments in private clouds. With BIG-IQ License Manager, you can spin up and provision individual VEs, or groups of VEs, from a single license pool on demand. When resource requirements decrease, spin down the VE and return it to the license pool for future use.

F5 Virtual Edition Software Modules

- BIG-IP® Local Traffic Manager™ (LTM) VE is a virtual Application Delivery Controller. It enables you to deliver network services in a reliable, secure, and optimized way. This VNF provides Layer 4–7 load-balancing and Layer 7 traffic management—allowing you to optimize and offload other network resources, including value-added services (VAS) platforms and other VNFs in your network.
- BIG-IP® DNS VE is a virtual DNS. It secures your DNS infrastructure through high-performance DNS services, scales DNS responses to survive volume increases and distributed denial-of-service (DDoS) attacks, and ensures high availability of your global applications and services. This VNF also provides DNS scalability and delivery offload to your LDNS infrastructure. By delivering faster response times for content being accessed by fixed and mobile devices, vDNS offers higher subscriber QoE.
- BIG-IP Advanced Firewall Manager (AFM) is a high-performance, stateful, full-proxy network firewall designed to guard data centers against incoming threats that enter the network on the most widely deployed protocols—including HTTP/S, SMTP, DNS, and FTP. By aligning firewall policies with the applications they protect, BIG-IP AFM streamlines application deployment, security, and monitoring. More information on BIG-IP AFM.
- BIG-IP ASM protects the HTTP applications your business relies on with an agile, certified web application firewall and comprehensive, policy-based web application security. Offering threat assessment and mitigation, visibility, and almost limitless flexibility, BIG-IP ASM helps you secure your HTTP applications. More information on BIG-IP Application Security Manager.

Note: In this deployment guide, we use BIG-IQ LM to license the VE tenants manually. BIG-IQ Centralized Management and Licensing will be featured in later use cases and deployment guides.

Ansible

Ansible is installed on a server. For details on how to install refer to the following links:

Installation documentation
Getting started with Ansible
Ansible module support

Cisco Datacenter Github:
Ansible Modules
Sample Playbooks

Cisco Cloud Center

Future testing of the F5 private cloud package for Cisco will expand to include Cisco CloudCenter (formerly CliQr) to more securely deploy and manage applications in private cloud environments. This application-centric cloud management solution helps modernize your private cloud application deployment to your service offering.

More information regarding Cisco Cloud center.

Deployment Information

This section covers the three deployment scenarios test and documented in this deployment guide.

1. Cisco ACI Service Manager Mode (Managed)
2. Cisco ACI Network Policy Mode (Unmanaged) while allowing the Application administrator to orchestrater the F5 L4-7
3. Cisco 9000 NX-OS (Standalone)

The network architecture we used to validate this deployment guides uses multiple tiers and tenants. To achieve multi-tenancy the F5 i5800 is configured with Virtualized Clustered Multiprocessing (vCMP). vCMP is a feature of the BIG-IP® system that allows you to run multiple instances of the BIG-IP software on a single hardware platform. This will allow you to configure instances of the BIG-IP software for Cisco ACI Service Manager Mode (Managed) and another instance of the BIG-IP software for Cisco ACI Network Policy Mode (Unmanaged). In the validation environment 4 vCMP guests have been created on 5800(1) and 5800(2) [Set HA between the 2 vCMP guests]:

1 HA guest pair for service manager tenant1A & tenant1B
1 HA guest pair for service manager tenant2A & tenant 2B
1 HA guest pair for unmanaged mode tenant1A & tenant 1B
1 HA guest pair for unmanaged mode tenant2A & tenant2B

4 BIG-IP VE's have been created (to showcase the web to app tier communication). We will show only 1 tenant in each company:

SM_Tenant1A_VE
SM_Tenant2A_VE
UM_Tenant1A_VE
UM_Tenant2A_VE

Cisco ACI Service Manager Mode (Managed)

F5 iWorkflow + Cisco ACI Supported Features

- vCMP HA Support with APIC Chassis Manager
    - Virtualized Clustered Multiprocessing (vCMP) is a feature of the BIG-IP® system that allows you to run multiple instances of the BIG-IP software on a single hardware platform
    - Thru APIC Chassis Manager, each F5 BIG-IP vCMP guest that act as concrete device can now be associated with a chassis, in F5 case, vCMP host. This will allow vCMP guests HA across two vCMP hosts
- iWorkflow HA Support with APIC Device Manager
    - Thru Cisco APIC Device Manager, a cluster of iWorkflow instances can be associated with the APIC Logical Device Cluster management, providing HA protection on iWorkflow / cluster level
- Dynamic endpoint attach and detach
    - Endpoints either can be pre-specified into corresponding EPGs (statically at any time) or can be added dynamically as they are attached to the Cisco ACI. Endpoints are tracked by a special endpoint registry mechanism of the policy repository. This tracking gives the Cisco APIC visibility into the attached endpoints.

APIC passes this information to the BIG-IP. From the BIG-IP's point of view this end point attached is a member of a pool and hence converts the APIC call that the device package receives into an addition of a member into a particular pool

F5 iWorkflow Dynamic Device Package

F5 iWorkflow dynamic device package is created based on F5 iApps template. The iApps template will determine features supported in the device package. F5 device package (similar to a plug-in) allows APIC simplified integration with virtually no disruption to existing service architectures. F5 device package shown below is two pieces: the device model (an XML file) and a device script (written in Python). The device model describes in a, APIC-consumable format what functions are available in the device script. The device package model is extensible, which consumes iApps from iWorkflow for deploying services-based, template driven configurations for L4-7 parameters configured via the APIC console.

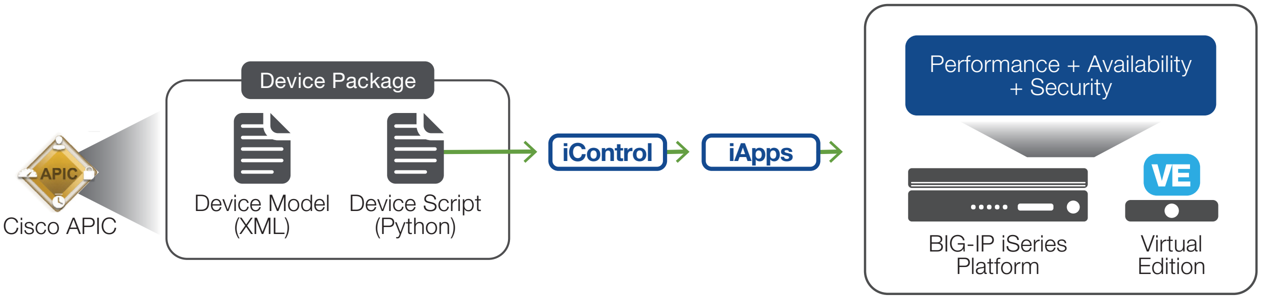


Figure 5: F5 Device Package using Open, Standards-based API

Cisco ACI L4-L7 Service Insertion Prerequisites

Review the design guide before proceeding to the L4-L7 service insertion configuration. The design guide will help you determine the L2-L3 networking elements and topology required for your use case.

Elements need to be preconfigured before you begin the L4-L7 service insertion.

F5 BIG-IP Prerequisites

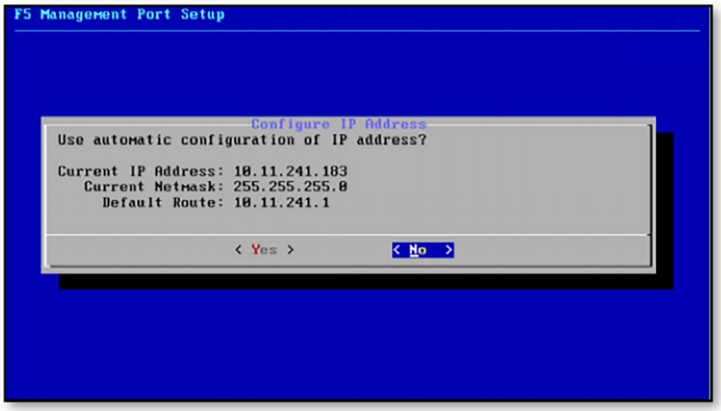
Before BIG-IP can be used for L4-L7 service insertion, it needs access to the management network and it needs to be licensed out of band.

To learn more about how to assign a management IP address to BIG-IP, please see K15040: Configuring and displaying the management IP address for the BIG-IP system.

One method for accessing the management IP address, if you have console access to BIG-IP, is to use the config command,

1. Log in to the console.

2. Run the command **config** on the command line. A wizard will appear to help you assign the management IP address.



```
F5 Management Port Setup

                     Configure IP Address
 Use automatic configuration of IP address?

 Current IP Address: 10.11.241.183
    Current Netmask: 255.255.255.0
      Default Route: 10.11.241.1

                 < Yes >          < No >
```

Figure 6: BIG-IP management IP config

3. Click **<No>** and continue with the wizard to enter you own IP address and netmask and default route.

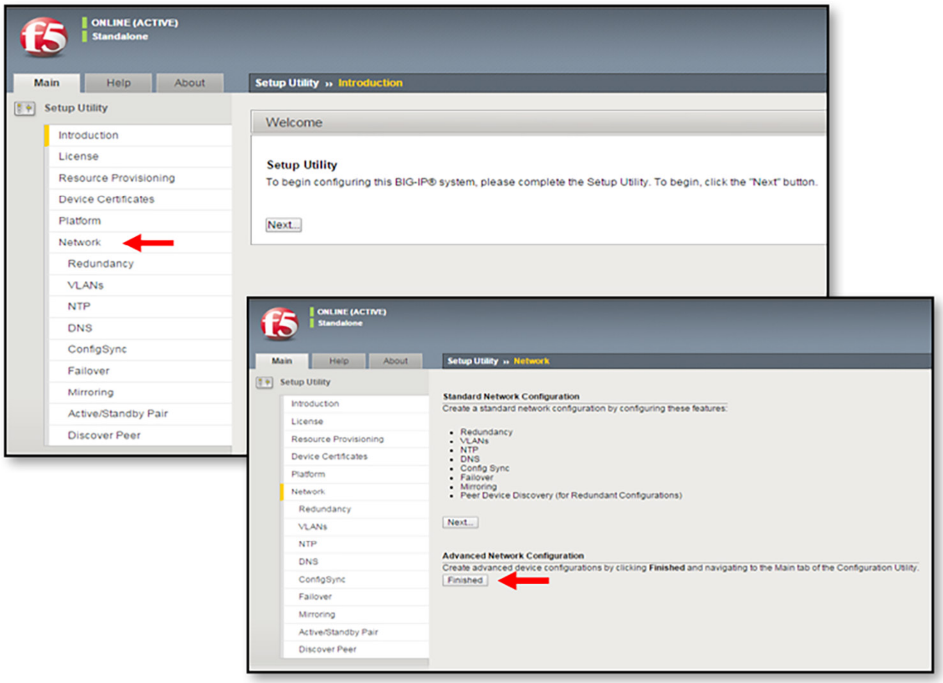
Install the License

You will need a registration key to apply the license.

From the GUI, follow these steps:

1. Log in to the GUI at https://<mgmt_ip_address>. The default username is **admin**, and the default password is **admin**.

2. From the Setup Utility menu, choose Network. Then click **Finished**.



Setup Utility

To begin configuring this BIG-IP® system, please complete the Setup Utility. To begin, click the "Next" button.

Standard Network Configuration

Create a standard network configuration by configuring these features:

- Redundancy
- VLANs
- NTP
- DNS
- Config Sync
- Failover
- Mirroring
- Peer Device Discovery (for Redundant Configurations)

Advanced Network Configuration

Create advanced device configurations by clicking **Finished** and navigating to the Main tab of the Configuration Utility.

Figure 7: BIG-IP Initial Setup Utility

3. After the configuration is saved, choose **System** from the main menu. Then choose **License**.
4. Apply the registration key to license the system and follow the wizard.

Click the following links for more information about license installation:

- License installation using the command-line interface (CLI)
- License installation using the GUI

F5 iWorkflow

F5 iWorkflow is a virtual appliance: Download iWorkflow 2.2.0.

F5 iWorkflow is control plane only in Cisco ACI integration, please ensure F5 iWorkflow, F5 BIG-IP and Cisco APIC management connectivity (either OOB or inband) is established.

Since F5 iWorkflow is control plane only, it does not need to be integrated with Cisco ACI VMM.

Please follow F5 iWorkflow documentation for initial bring up and licensing.

F5 iWorkflow supports standalone mode or 3-peers cluster. For production environment, F5 recommends 3-peers cluster iWorkflow configuration in order to provide high availability. Please refer to iWorkflow High Availability guide for detail in configuring iWorkflow cluster.

Cisco ACI

Integrating the Virtual Machine Manager When Using Virtual Machines

The APIC is a single-pane manager that automates the entire networking configuration for all virtual and physical workloads, including access policies and L4-L7 services. In the case of vCenter, all the networking functions of the VMware Virtual Distributed Switch (VDS) and port groups are performed using the APIC. The only task that a vCenter administrator needs to perform in vCenter is to place the virtual network interface cards (vNICs) in the appropriate groups that the APIC created.

More information about how to integrate vCenter and manage virtual machine domains and connectivity.

Defining Fabric Access Policies to Communicate with F5 Hardware

Access policies govern the operation of switch access ports that provide connectivity to resources such as storage, computing, Layer 2 and Layer 3 (bridged and routed) connectivity, virtual machine hypervisors, L4-L7 devices, and so on. If a tenant requires interface configurations other than those provided in the default link, Cisco Discovery Protocol, Link Layer Discovery Protocol (LLDP), Link Aggregation Control Protocol (LACP), or Spanning Tree Protocol, an administrator must configure access policies to enable such configurations on the access ports of the leaf switches.

The following entities need to be configured to add a new device to the fabric:

- Physical and external domains
- VLAN pool
- Attachable access entity profile
- Interface policies
- Switch policies

Detailed explanation of the steps for configuring these entities (look in the fabric connectivity section).

Creating Tenants

A Tenant is a logical container of a folder for application policies. The container can represent an actual tenant, an organization, or a domain or can just be used for conveniences of organizing information. A tenant represents a unit of isolation from a policy perspective, but it does not represent a private network. A context is representation of a private later 3 namespace or layer 3 network. It is a unit of isolation in the ACI framework. A tenant can rely on several contexts. Context can be declared within a tenant or can be in the "Common Tenant." In this deployment guide 8 tenants have been created on the APIC shown in the Figure 8.

SM_Tenant1A
SM_Tenant1B
SM_Tenant2A
SM_Tenant2B
UM_Tenant1A
UM_Tenant1B
UM_Tenant2A
UM_Tenant2B

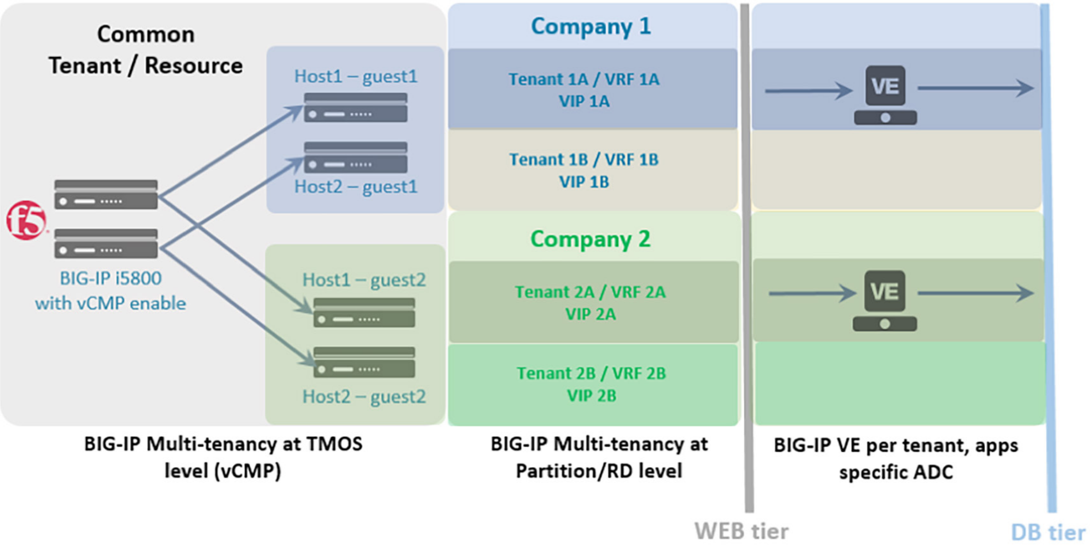


Figure 8: Verification Lab setup showing multi-tenancy

Create the VRF and Bridge Domain

A VRF (Virtual Routing and Forwarding) is a unique Layer 3 forwarding and application policy domain that provides IP address space isolation for tenants.

A bridge domain represents a Layer 2 forwarding construct within the fabric.

In BIG-IP, a tenant in the APIC maps to a partition in BIG-IP with a unique route domain (RD). A contextual VRF instance is represented in BIG-IP as a route domain.

For a typical deployment for one tenant, define:

- One private network
- One or more bridge domains
    - Two-arm Service Graph: two bridge domains (one representing the client subnet, and one representing the server subnet).
    - One-arm Service Graph: one bridge domain (this bridge domain can either be external, internal or F5 BIG-IP subnet)

Create the Application Profile and EPG

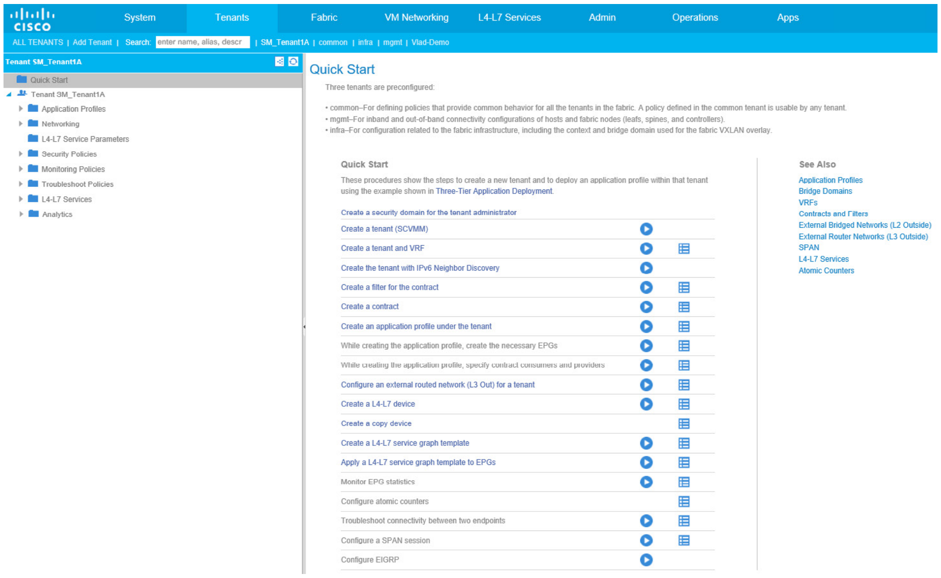
Application profiles contain one or more EPGs. Modern applications contain multiple components. For example, an e-commerce application might require a web server, a database server, data located in a SAN, and access to outside resources that enable financial transactions. The application profile contains as many (or as few) EPGs as necessary that are logically related to the capabilities of an application.

EPGs can be organized according to any of the following:

- The application they provide (such as SAP applications)
- The function they provide (such as infrastructure)
- Where they are in the structure of the data center (such as the DMZ)
- Whatever organizing principle that a fabric or tenant administrator chooses to use

Detailed explanation of EPG configuration steps.

For Cisco ACI terminology, go to Cisco APIC Tenant space, go to **Quick Start**:



Quick Start

Three tenants are preconfigured:

• common–For defining policies that provide common behavior for all the tenants in the fabric. A policy defined in the common tenant is usable by any tenant.
• mgmt–For inband and out-of-band connectivity configurations of hosts and fabric nodes (leafs, spines, and controllers).
• infra–For configuration related to the fabric infrastructure, including the context and bridge domain used for the fabric VXLAN overlay.

Quick Start

These procedures show the steps to create a new tenant and to deploy an application profile within that tenant using the example shown in Three-Tier Application Deployment.

Create a security domain for the tenant administrator
Create a tenant (SCVMM)
Create a tenant and VRF
Create the tenant with IPv6 Neighbor Discovery
Create a filter for the contract
Create a contract
Create an application profile under the tenant
While creating the application profile, create the necessary EPGs
While creating the application profile, specify contract consumers and providers
Configure an external routed network (L3 Out) for a tenant
Create a L4-L7 device
Create a copy device
Create a L4-L7 service graph template
Apply a L4-L7 service graph template to EPGs
Monitor EPG statistics
Configure atomic counters
Troubleshoot connectivity between two endpoints
Configure a SPAN session
Configure EIGRP

See Also

Application Profiles
Bridge Domains
VRFs
Contracts and Filters
External Bridged Networks (L2 Outside)
External Router Networks (L3 Outside)
SPAN
L4-L7 Services
Atomic Counters

There are many resources available. Click on the book icon, APIC terminologies are available:

Configure Cisco ACI Service Manager Mode (Managed)

1. Adding BIG-IP Devices to the iWorkflow Inventory
2. iWorkflow Import F5 HTTP iApps
3. iWorkflow Cloud: Device Discovery
4. iWorkflow Cloud: Create Service Template
5. iWorkflow Cloud: Dynamic Creation of Customized F5 Device Package
6. APIC: Import F5 Device Package
7. APIC: Device Manager Configuration
8. APIC: Chassis Manager Configuration
9. APIC: Create L4-L7 Device with F5 vCMP Guests
10. APIC: Create Service Graph Template
11. APIC: Create Two-Arm Service Graph Template
12. APIC: Deploy Two-Arm Service Graph Template

iWorkflow Requirements for iApps

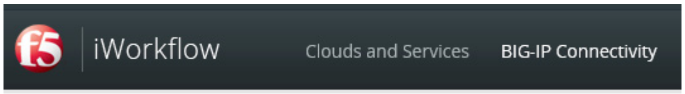
iWorkflow requires an iApps template to be named and versioned. An iApps template is fundamentally a code script which has been developed by a qualified engineering team and tested to deliver specific functionality at a specific scale. Just like any software product, an iApps templates is expected to be improved over time to address defects and to deliver improved functionality and scale. These improvements are delivered as ascending versions of the template. The iApps template name is treated as a stable identifier that will not change across versions. Each version of an iApps template can optionally include a BIG-IP compatibility matrix that defines clearly the minimum, maximum and known incompatible versions of BIG-IP.

iWorkflow requires a JSON representation of the iApps APL API. This JSON representation of the iApps template is the basis for all service templating and provisioning that iWorkflow performs with a version of an iApps template. Ideally, this JSON representation will be provided by the iApps author along with the release of an iApps version. The iApps author can use iWorkflow to generate this JSON document or can do so using their iApp build process. If this JSON representation is not published by the iApp author than the iWorkflow GUI can be used to populate it during the iApps template import process. In this deployment guide, we will be using the iWorkflow GUI to populate the iApps template during the import process. You can download the f5.http supported iApp via F5 Downloads. You need to discover the BIG-IP® devices before adding the F5 HTTP iApp. For background knowledge on developing iApps using APL and TCL, review the BIG-IP iApps Developer's Guide. This will familiarize you with the concepts required to build an iApps template that can provision BIG-IP application services.

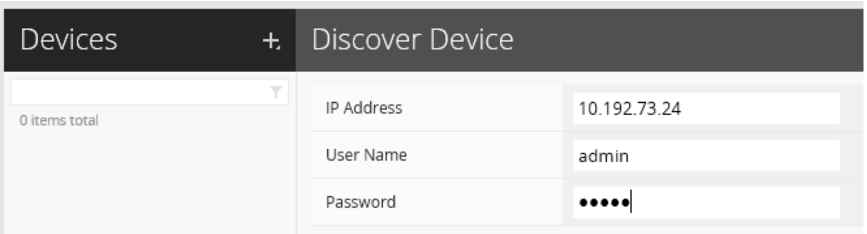
Adding BIG-IP Devices to the iWorkflow Inventory

After you license and perform the initial configuration for the iWorkflow™ system, you can discover BIG-IP® devices. By registering F5 BIG-IP under iWorkflow, iWorkflow will update the BIG-IP REST framework, ensuring reliable communications between iWorkflow and BIG-IP

Upon log into iWorkflow, go to BIG-IP Connectivity:



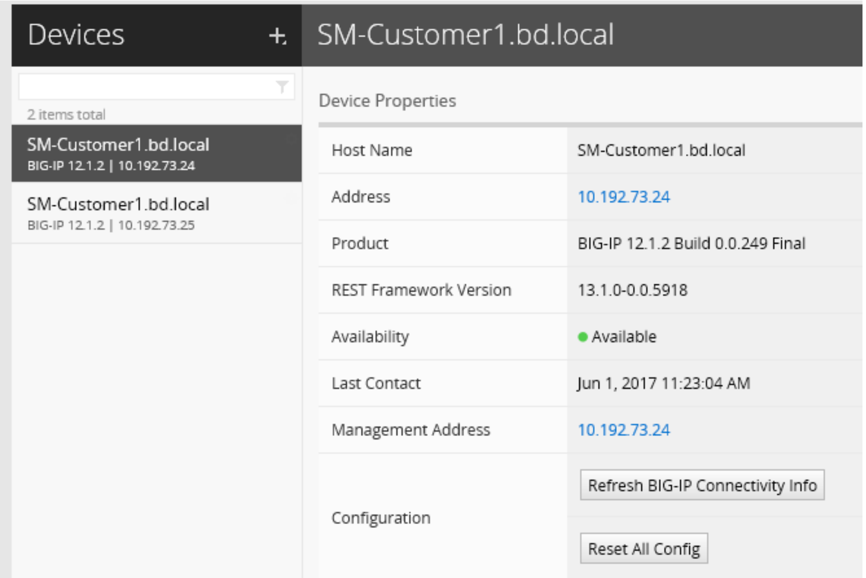
Under BIG-IP Connectivity, select Devices ->, click **+ Discover Devices**. Enter the BIG-IP management IP and login credentials. For BIG-IP HA setup, make sure discover both BIG-IP. In this deployment guide the BIG-IPs i5800 is configured for multitenancy using vCMP guest's. We are using the **SM_Tenant1A** BIG-IP management IP:



Discover Device

IP Address	10.192.73.24
User Name	admin
Password	●●●●●

Both BIG-IP devices should have been added to iWorkflow. Upon successfully discovery, BIG-IP availability become "Available." Once available you can import the F5 HTTP iApp.



SM-Customer1.bd.local
BIG-IP 12.1.2 | 10.192.73.24

SM-Customer1.bd.local
BIG-IP 12.1.2 | 10.192.73.25

SM-Customer1.bd.local

Device Properties

Host Name	SM-Customer1.bd.local
Address	10.192.73.24
Product	BIG-IP 12.1.2 Build 0.0.249 Final
REST Framework Version	13.1.0-0.0.5918
Availability	● Available
Last Contact	Jun 1, 2017 11:23:04 AM
Management Address	10.192.73.24
Configuration	Refresh BIG-IP Connectivity Info Reset All Config

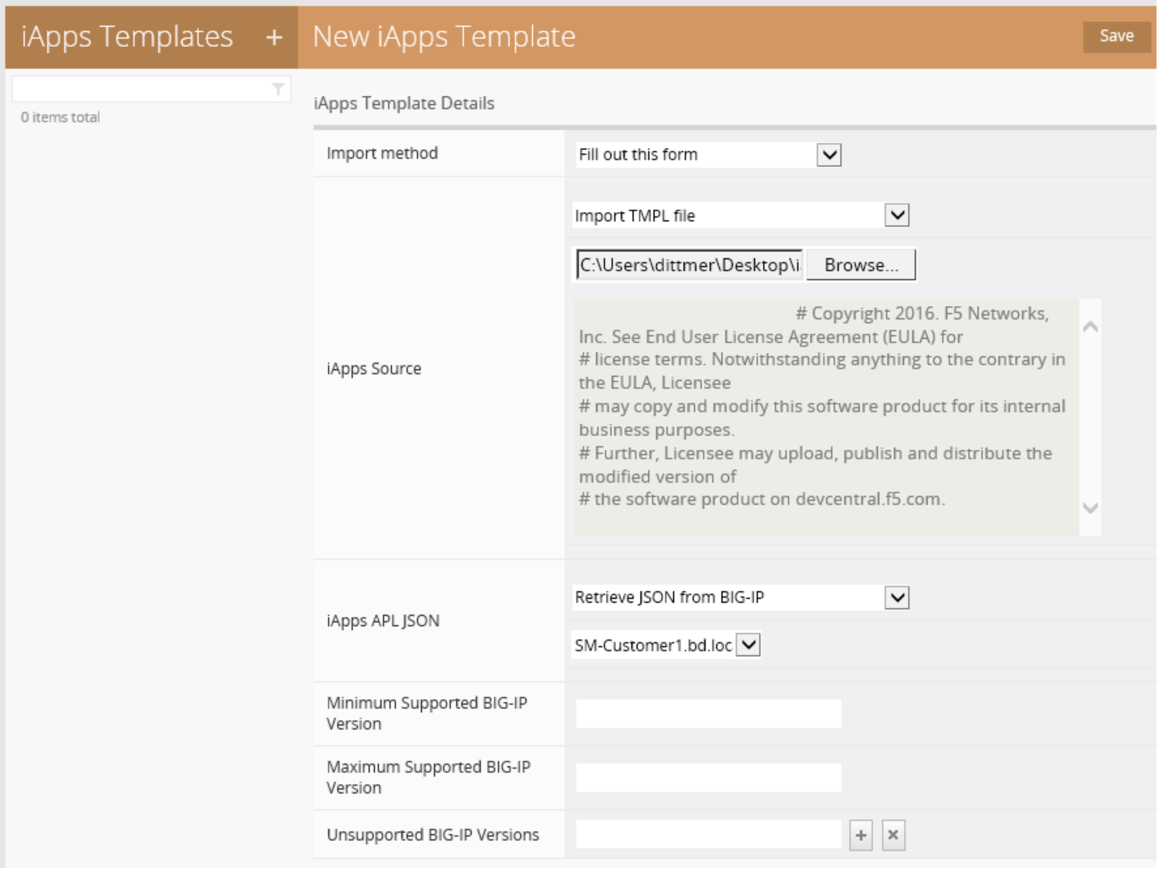
iWorkflow Import F5 HTTP iApps

iWorkflow requires a basic understanding about some key parts of the iApps template's APL API. This is referred to as the Service Tier information and represents a mapping of key APL variables, tables and table columns. This mapping enables iWorkflow to provide a better presentation layer for L4-L7 Service management and to collect helpful health and usage statistics from the BIG-IP application services. This Service Tier information can be provided in a number of ways. Ideally, the iApp author will provide along with the release of an iApps version in the form of a JSON file that includes the iApps TMPL file content, APL JSON representation and the Service Tier Information. If this Service Tier information is not published by the iApp author than the iWorkflow GUI can be used to populate it during the iApps template import process.

In the Clouds and Services tab on the top, go to iApps Templates ->, click **+…**

1. Select the f5.http.v1.2.0.tmpl iApp
2. Select the dropdown tab to retrieve the JSON from the BIG-IP.

Select the correct BIG-IP from the device list. Ignore the Service Tier Information. Configuration of the Service Tier information will be done in the Service Template configuration.



iApps Template Details

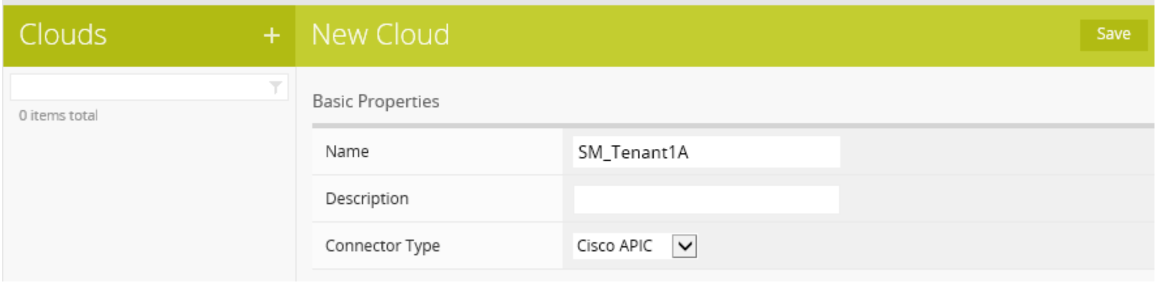
Import method	Fill out this form
iApps Source	Import TMPL file C:\Users\dittmer\Desktop\i Browse... # Copyright 2016. F5 Networks, Inc. See End User License Agreement (EULA) for # license terms. Notwithstanding anything to the contrary in the EULA, Licensee # may copy and modify this software product for its internal business purposes. # Further, Licensee may upload, publish and distribute the modified version of # the software product on devcentral.f5.com.
iApps APL JSON	Retrieve JSON from BIG-IP SM-Customer1.bd.loc
Minimum Supported BIG-IP Version	
Maximum Supported BIG-IP Version	
Unsupported BIG-IP Versions	

iWorkflow Cloud: Device Discovery

Create the iWorkflow Cloud APIC Connectors which will generate a custom device package that contains iWorkflow service catalog. Go to iWorkflow Cloud menu, click **+**:

Name: <user_defined>, for this example, use **SM_Tenant1A**

Connector Type: **Cisco APIC**

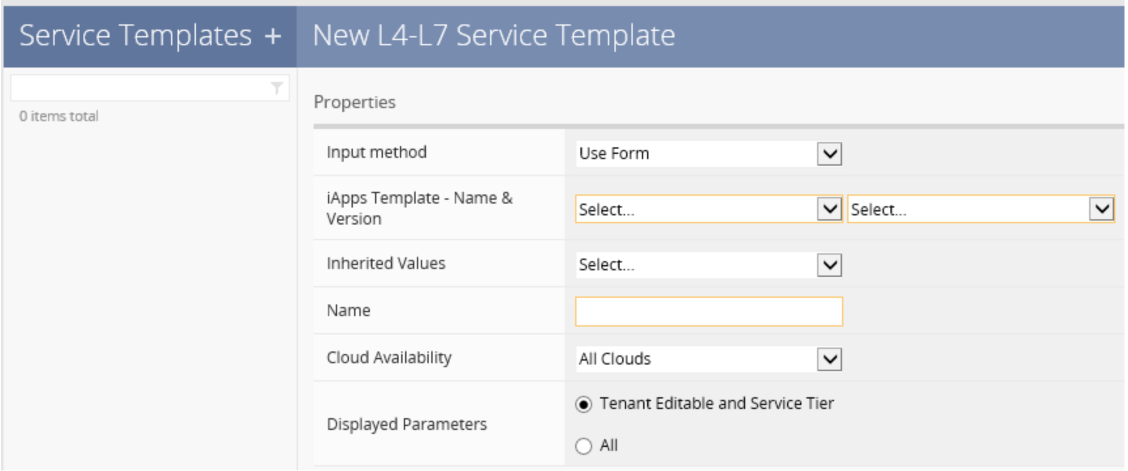


Name	SM_Tenant1A
Description	
Connector Type	Cisco APIC

iWorkflow Cloud: Create Service Template

After BIG-IP is successfully discovered by iWorkflow, and the iApps reside on iWorkflow. Create an service template inside iWorkflow Service Catalog. User can specify F5 virtual server requirements and build them into a template. Please note this deployment illustrate a common example, user can customize the template based on F5 virtual server requirements.

Select iWorkflow Cloud and Services. When the Service Template menu appears on the screen, click + to configure the new L4-L7 Service Template:

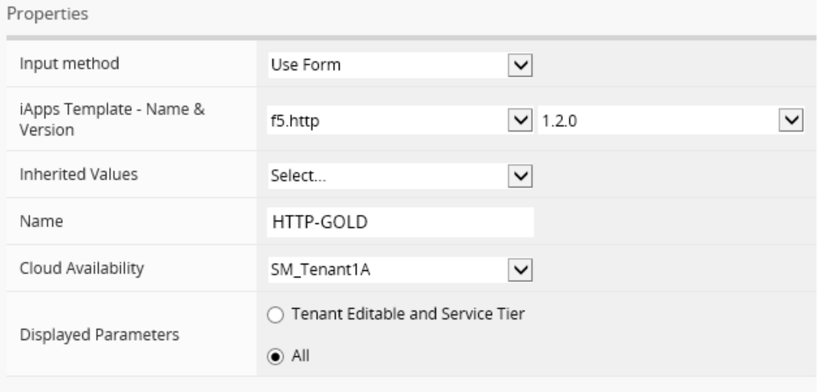


Input method	Use Form
iApps Template - Name & Version	Select... Select...
Inherited Values	Select...
Name	
Cloud Availability	All Clouds
Displayed Parameters	● Tenant Editable and Service Tier ○ All

Name: user defined name for the template, recommendation would be type of template for multiple application use, for example: HTTP-Gold.

Input Parameters:

- iApps Template – Name & Version: f5.http 1.2.0
- Cloud Availabilty – SM_Tenent1a
- Display Parameters: All
- All Options: All parameters available for edit

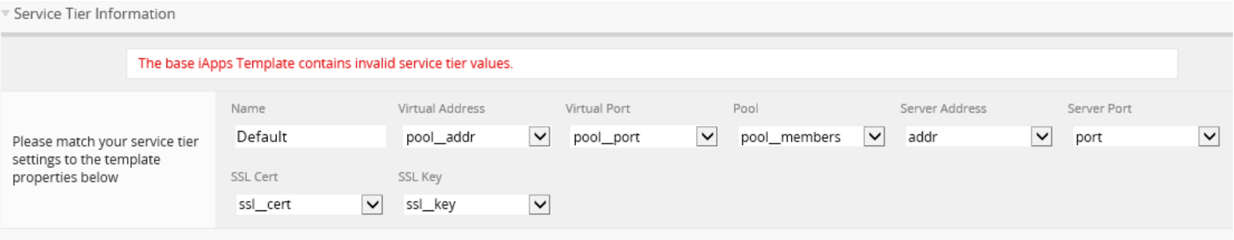


Properties

Input method	Use Form
iApps Template - Name & Version	f5.http 1.2.0
Inherited Values	Select...
Name	HTTP-GOLD
Cloud Availability	SM_Tenant1A
Displayed Parameters	○ Tenant Editable and Service Tier ● All

Configure the Service Tier information from the following diagram:

Service Tier Information	
Name	Default
Virtual Address	pool_addr
Virtual Port	pool_port
Pool	pool_members
Pool Server Address	addr
Pool Server Port	port
SSL Cert	ssl_cert
SSL Key	ssl_key



Service Tier Information

The base iApps Template contains invalid service tier values.

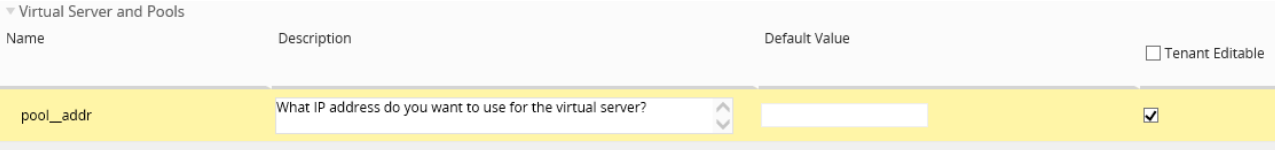
Please match your service tier settings to the template properties below

Name	Virtual Address	Virtual Port	Pool	Server Address	Server Port
Default	pool__addr	pool__port	pool__members	addr	port

SSL Cert	SSL Key
ssl__cert	ssl__key

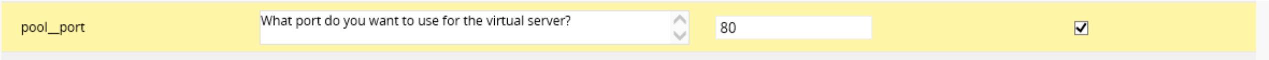
Configure the Virtual Server and Pools Tenant editable fields so how in the APIC GUI:

Virtual Server and Pools	
pool__addr	Tenant Editable
pool__port	Tenant Editable
pool__members	
addr	Tenant Editable
port	Tenant Editable

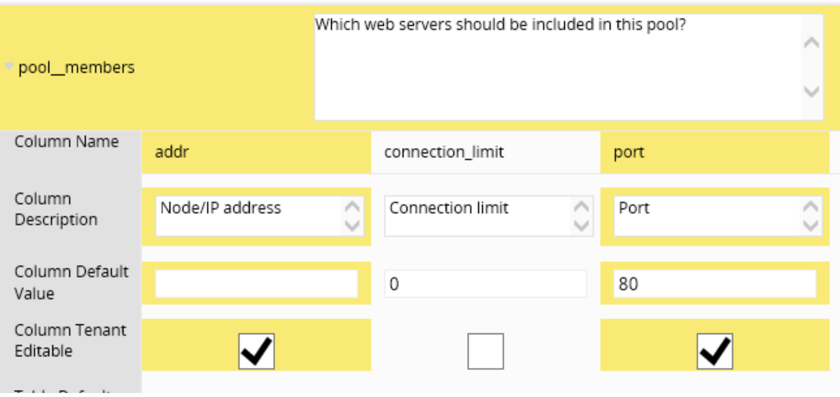


Virtual Server and Pools

Name	Description	Default Value	□ Tenant Editable
pool__addr	What IP address do you want to use for the virtual server?		☑



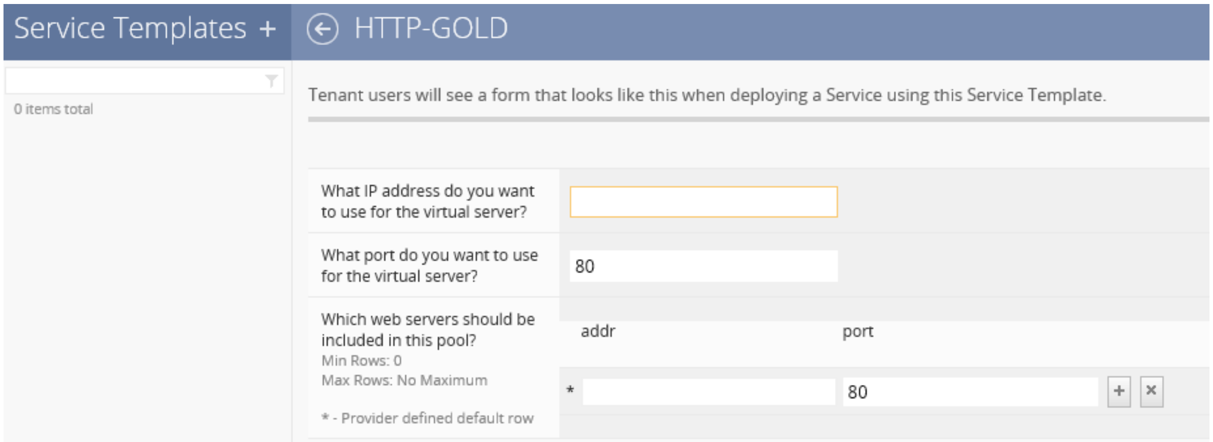
pool__port	What port do you want to use for the virtual server?	80	☑



pool__members — Which web servers should be included in this pool?

Column Name	addr	connection_limit	port
Column Description	Node/IP address	Connection limit	Port
Column Default Value		0	80
Column Tenant Editable	☑	□	☑

Click **Tenant Preview** to see the default value (provider template) of this iApps, those parameters are considered "Tenant Editable" and will be exposed to Cisco APIC thru device package.



Tenant users will see a form that looks like this when deploying a Service using this Service Template.

What IP address do you want to use for the virtual server?		
What port do you want to use for the virtual server?	80	
Which web servers should be included in this pool? Min Rows: 0 Max Rows: No Maximum * - Provider defined default row	addr	port
	*	80

Only VIP, Ports and Pool Member are tenant editable.

Click ← to go back for editing

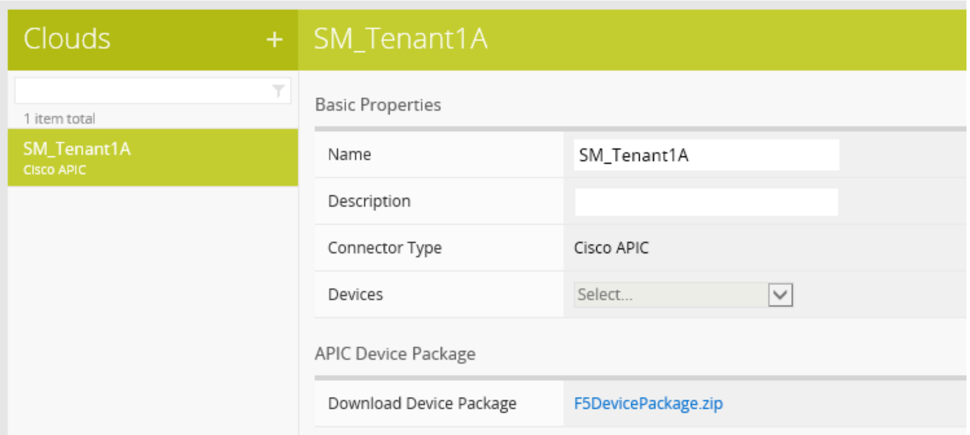
Click Save to complete the template:

As expected, only "Virtual Server: Port" parameter is added to the list of parameters exposing to APIC

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

iWorkflow Cloud: Dynamic Creation of Customized F5 Device Package

Double click on the Clouds Connector **SM_Tenant1A**, notice there is a link to download a customized F5 Device Package that contains this iWorkflow Service Catalog:



SM_Tenant1A

Basic Properties

Name	SM_Tenant1A
Description	
Connector Type	Cisco APIC
Devices	Select...

APIC Device Package

Download Device Package	F5DevicePackage.zip

Click the Download Device Package **F5DevicePackage.zip** to save the custom device package to a location that is accessible by Cisco APIC.

The configuration steps on iWorkflow necessary prior to F5 ACI integration are completed.

APIC: Import F5 Device Package

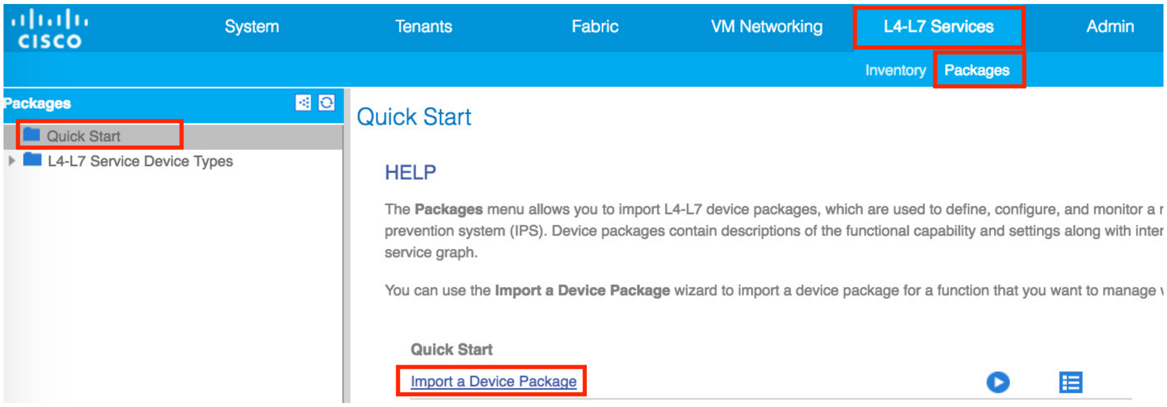
Thru Cisco APIC, user can perform the workflow in deploying the HTTP application, with the integration of F5 iWorkflow and BIG-IP, user can apply HTTP application L4-L7 requirements within APIC policy model, reducing significant amount of operation complexity.

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.

Using APIC GUI and click the following to import the device package:

L4-L7 Services -> Packages

Under Quick Start, click **Import a Device Package**



Quick Start

HELP

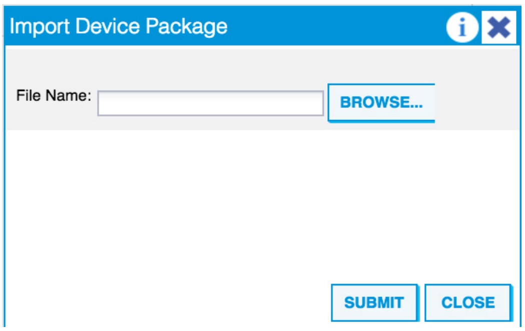
The **Packages** menu allows you to import L4-L7 device packages, which are used to define, configure, and monitor a r prevention system (IPS). Device packages contain descriptions of the functional capability and settings along with inter service graph.

You can use the **Import a Device Package** wizard to import a device package for a function that you want to manage \

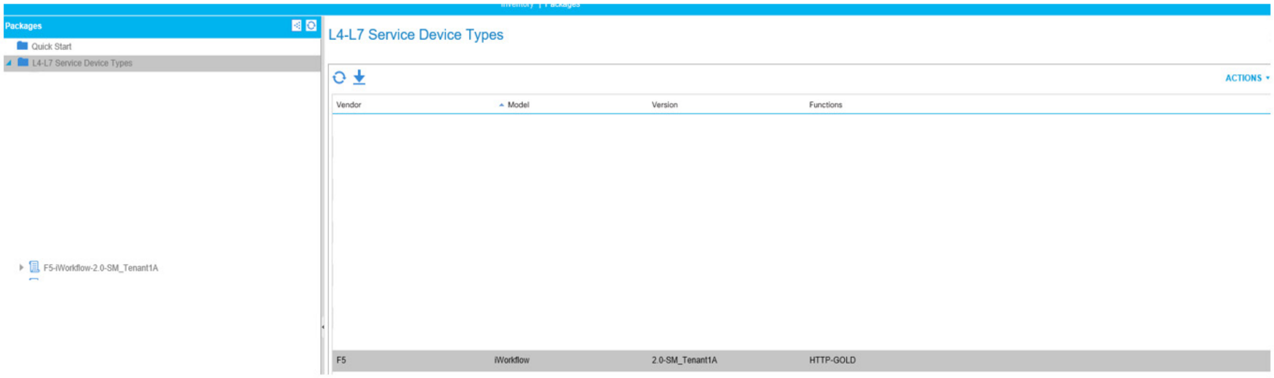
Quick Start

Import a Device Package

A new pop-up should appear to allow you to choose the device package to be installed:

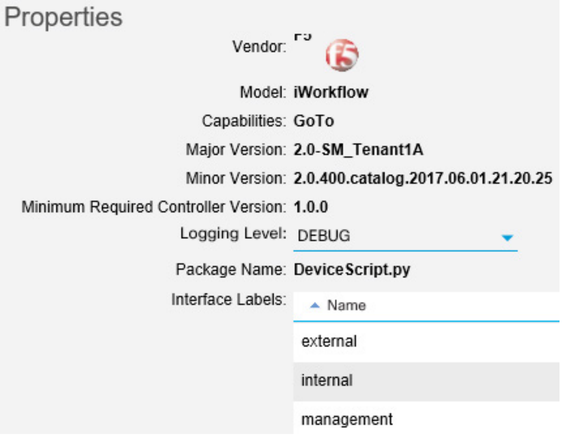


Click **BROWSE** and choose the previous downloaded device package - F5DevicePackage.zip and click **SUBMIT** to import the device package.



L4-L7 Service Device Types

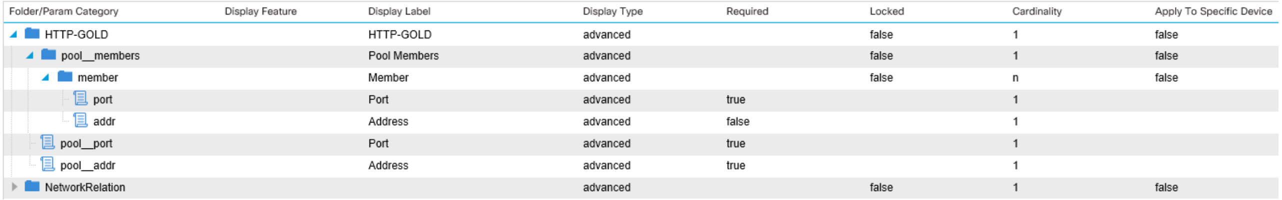
Vendor	Model	Version	Functions
F5	iWorkflow	2.0-SM_Tenant1A	HTTP-GOLD



Properties

Vendor:	F5
Model:	iWorkflow
Capabilities:	GoTo
Major Version:	2.0-SM_Tenant1A
Minor Version:	2.0.400.catalog.2017.06.01.21.20.25
Minimum Required Controller Version:	1.0.0
Logging Level:	DEBUG
Package Name:	DeviceScript.py
Interface Labels:	Name external internal management

Click on **OPERATION**, notice only tenant editable parameters of the template is visible in Cisco APIC:



Folder/Param Category	Display Feature	Display Label	Display Type	Required	Locked	Cardinality	Apply To Specific Device
HTTP-GOLD		HTTP-GOLD	advanced		false	1	false
pool__members		Pool Members	advanced		false	1	false
member		Member	advanced		false	n	false
port		Port	advanced	true		1	
addr		Address	advanced	false		1	
pool__port		Port	advanced	true		1	
pool__addr		Address	advanced	true		1	
NetworkRelation			advanced		false	1	false

APIC: Device Manager Configuration

To integrate F5 iWorkflow cluster into Cisco APIC L4-L7 devices, using Cisco APIC device manager feature to define and specify F5 iWorkflow.

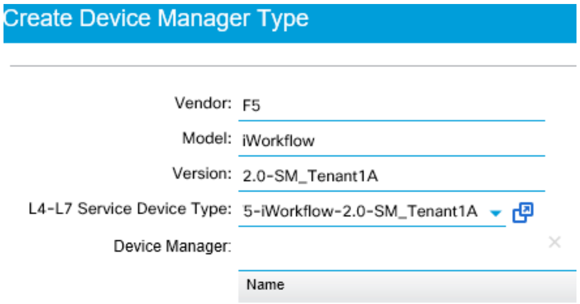
From APIC perspective, F5 iWorkflow is a “device manager” managing the F5 BIG-IP ADC (both physical and virtual form factors).

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

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

A new pop up window will appear

- Vendor: F5 (this is the vendor info of this device manager)
- Model: iWorkflow (product model)
- Version: 2.0-SM_Tenant1A (it is extremely important to state the version value 2.0-<name of the connector specify in iWorkflow>)
- L4-L7 Service Device Type: F5-iWorkflow-2.0-VNG-iW (select the device package)
- Device Manager: Leave this field empty



Create Device Manager Type

Vendor: F5

Model: iWorkflow

Version: 2.0-SM_Tenant1A

L4-L7 Service Device Type: 5-iWorkflow-2.0-SM_Tenant1A

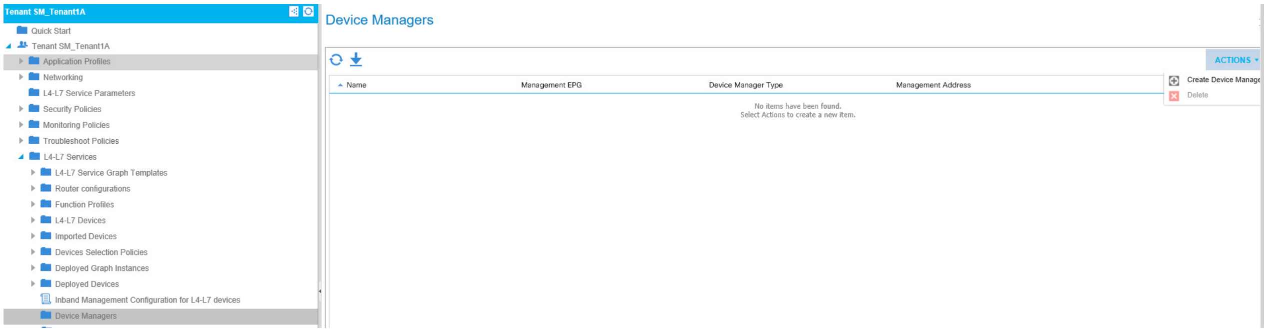
Device Manager:

Name

Click **SUBMIT**. The Device Manager Type is now created and can be associated with a Device Manager. Go to the APIC tenant where the L4-L7 device will be created. In this example, go to Tenant SM_Tenant1A:

Tenants SM_Tenant1A -> L4-L7 Services -> Device Managers

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

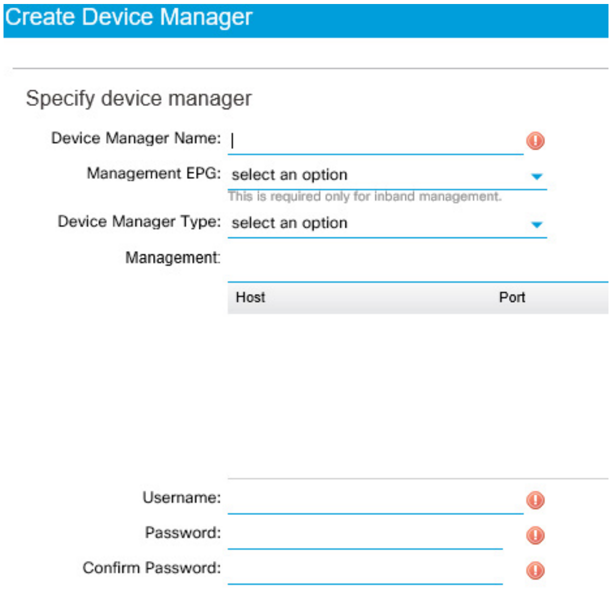


Device Managers

Name	Management EPG	Device Manager Type	Management Address

No items have been found.
Select Actions to create a new item.

A new pop up appears:



Create Device Manager

Specify device manager

Device Manager Name: |

Management EPG: select an option

This is required only for inband management.

Device Manager Type: select an option

Management:

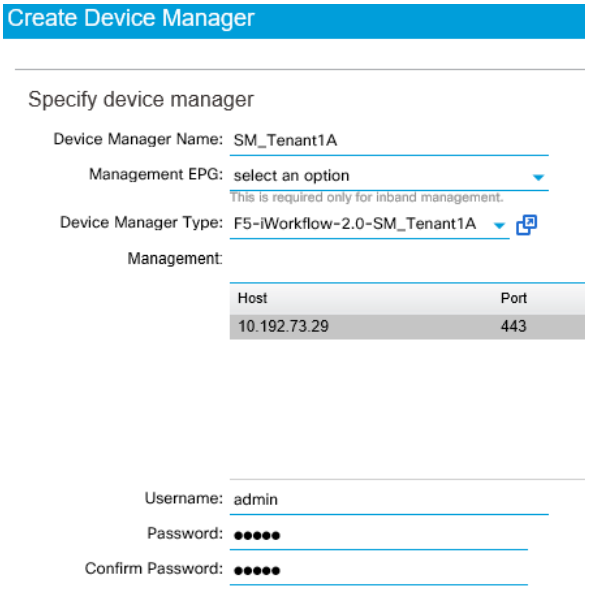
Host	Port

Username:

Password:

Confirm Password:

- Device Manager Name: User defined
- Management EPG: Leave it blank (only use for inband management)
- Device Manager Type: Select the type created
- Management: adding each iWorkflow management IP
- Username: iWorkflow admin username
- Password / Confirm Password: iWorkflow admin password (Notice, all three iWorkflow virtual appliance must have the same admin password)



Create Device Manager

Specify device manager

Device Manager Name: SM_Tenant1A

Management EPG: select an option

This is required only for inband management.

Device Manager Type: F5-iWorkflow-2.0-SM_Tenant1A

Management:

Host	Port
10.192.73.29	443

Username: admin

Password: ●●●●●

Confirm Password: ●●●●●

Click **SUBMIT**.

This complete the steps to create APIC L4-L7 device manager. This device manager will be used later when creating APIC L4-L7 device.

APIC: Chassis Manager Configuration

(**Note:** F5 vCMP guest only which is what we are using in this deployment guide)

This is a setup for a logical device cluster, following is what is being used/setup:

- HA – setup HA through the device package, not out of band
- vCMP chassis – 2 vCMP chassis, 1 host on each chassis
- Device manager – 1 iWorkflow present in the device manager

IP addresses in our validation setup

- VCMP host – **10.192.73.91** and **10.192.73.92**
- Vcmp guest – **10.192.73.24** and **10.192.73.25**
- Device manager chassis – 102.192.73.29

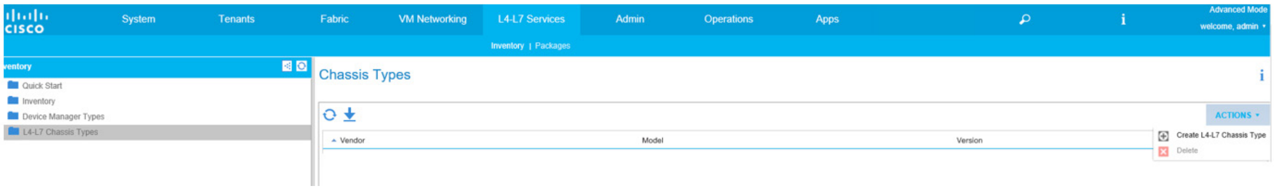
Since APIC L4-L7 Device using F5 vCMP guest, user will need to configure APIC chassis manager in order to specify the vCMP host information.

From APIC perspective, F5 BIG-IP vCMP host is a "chassis manager" managing the F5 BIG-IP vCMP guest.

First define the chassis manager type. In the APIC GUI, click the following to configure the Chassis Manager Type:

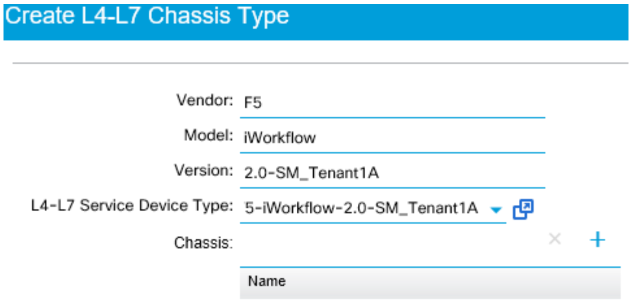
L4-L7 Services -> Inventory -> Chassis Manager Type

Click the **ACTIONS** button at the Work pane and choose **Create Chassis Manager Type**:



A new pop up window will appear

- Vendor: F5 (this is the vendor info of this device manager)
- Model: iWorkflow (product model)
- Version: 2.0-VNG-iW (it is extremely important to state the version value 2.0-<name of the connector specify in iWorkflow>)
- L4-L7 Service Device Type: F5-iWorkflow-2.0-VNG-iW (select the device package)
- Chassis: Leave this field empty



Create L4-L7 Chassis Type

Vendor: F5

Model: iWorkflow

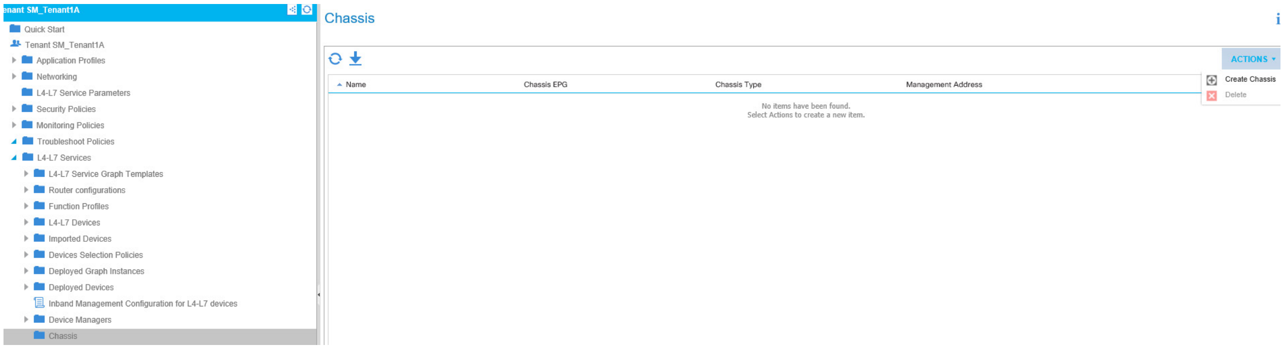
Version: 2.0-SM_Tenant1A

L4-L7 Service Device Type: 5-iWorkflow-2.0-SM_Tenant1A

Chassis:

Name

Click **SUBMIT**. The Chassis Manager Type is now created and can be associated with a Chassis. Go to the APIC tenant where the L4-L7 device will be created. In this example, go to **Tenant SM_Tenant1A**:

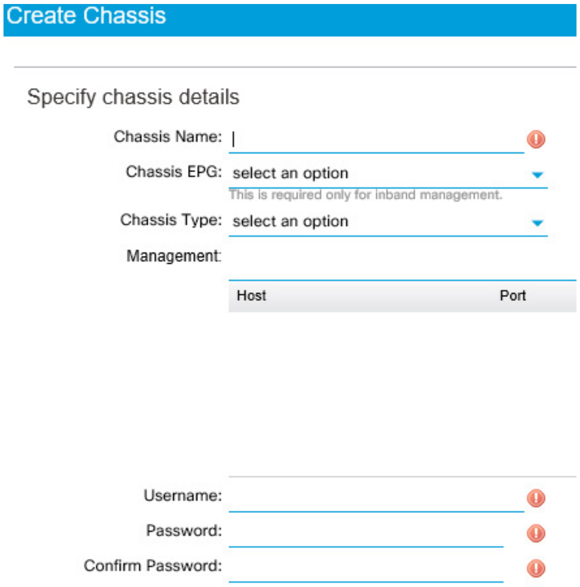


Chassis

Name	Chassis EPG	Chassis Type	Management Address

No items have been found.
Select Actions to create a new item.

A new pop up window will appear:



Create Chassis

Specify chassis details

Chassis Name:

Chassis EPG: select an option

This is required only for inband management.

Chassis Type: select an option

Management:

Host	Port

Username:

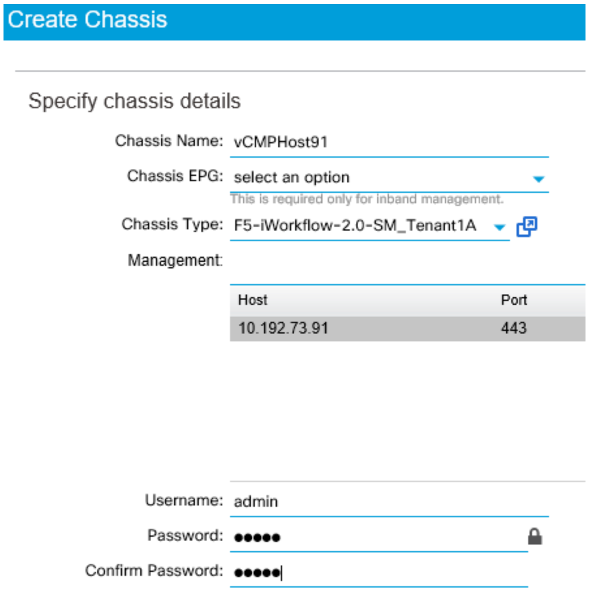
Password:

Confirm Password:

- Chassis Name: User defined
- Management EPG: Leave it blank (only use for inband management)
- Chassis Type: Select the type created

- Management: vCMP host management IP
- Username: vCMP host admin username
- Password / Confirm Password: vCMP host admin password

Duplicate this step for 2nd vCMP host in HA environment.


Create Chassis

Specify chassis details

Chassis Name: vCMPHost91

Chassis EPG: select an option

This is required only for inband management.

Chassis Type: F5-iWorkflow-2.0-SM_Tenant1A

Management:

Host	Port
10.192.73.91	443

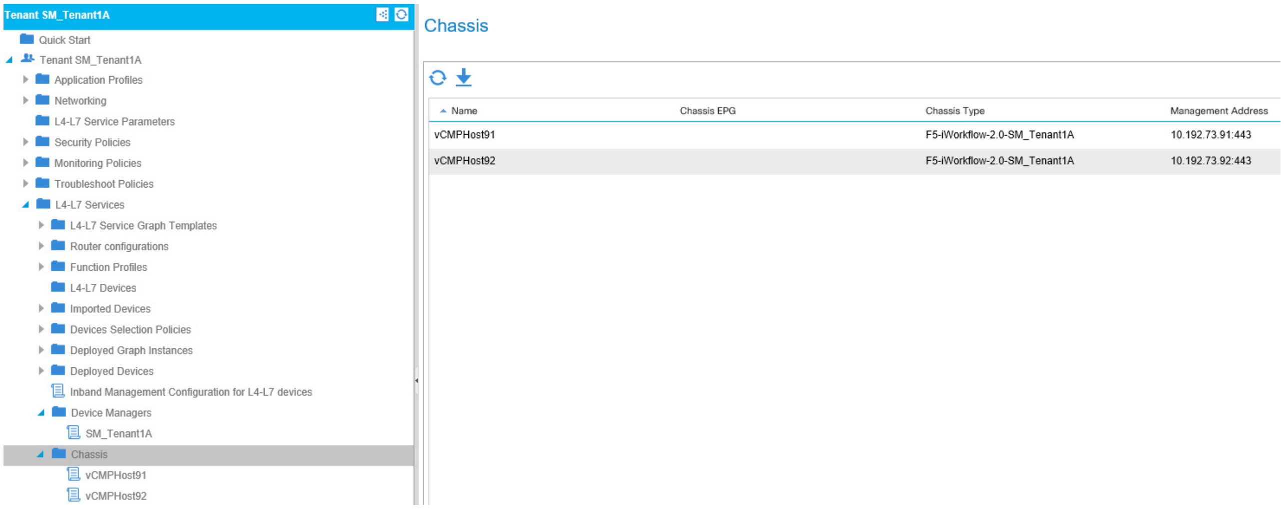
Username: admin

Password: ●●●●●

Confirm Password: ●●●●●

Click **SUBMIT**.

This completes the steps to create APIC L4-L7 chassis manager. This chassis will be used later when creating APIC L4-L7 device that uses F5 vCMP guests as concrete devices.



Chassis

Name	Chassis EPG	Chassis Type	Management Address
vCMPHost91		F5-iWorkflow-2.0-SM_Tenant1A	10.192.73.91:443
vCMPHost92		F5-iWorkflow-2.0-SM_Tenant1A	10.192.73.92:443

APIC: Create L4-L7 Device with F5 vCMP Guests

A L4-L7 device (also known as a logical device cluster, LDev) contains one or more devices (also known as concrete devices) that act as a logical entity, it also references F5 iWorkflow information. A L4-L7 device has logical interfaces, which describe the interface information for the logical device cluster. During service graph template rendering, function node connectors are associated with logical interfaces. The APIC allocates the network resources (VLAN or Virtual Extensible LAN [VXLAN]) for a function node connector during service graph template instantiation and rendering and programs the network resources on the logical interfaces.

An administrator can set up a maximum of two concrete devices for a single logical device clusters in the active-standby mode.

The logical device cluster has information about BIG-IP credentials that the APIC will use to communicate with BIG-IP.

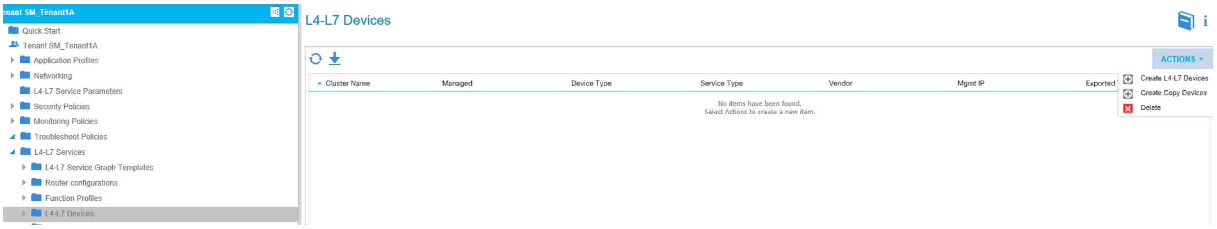
The logical device cluster can be created in tenant common or in your created tenant. The advantage of creating it in tenant common is that this logical device cluster can then be exported to multiple tenants and used by multiple tenants.

In this example, which is multi-tenant scenario, create the L4-L7 device in tenant common and export the L4-L7 device to user-defined tenant. Navigate to Tenant SM_Tenant1A to create a new L4-L7 Device by clicking the following:

Tenants SM_Tenant1A -> L4-L7 Services -> L4-L7 Devices

In the Work pane, click:

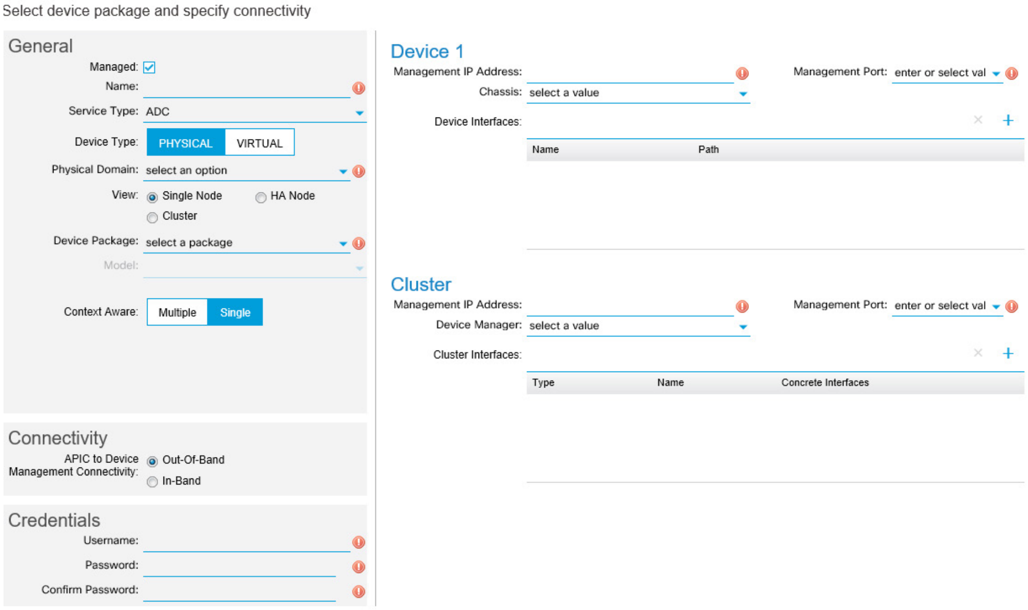
ACTIONS -> Create L4-L7 Devices



Cluster Name	Managed	Device Type	Service Type	Vendor	Mgmt IP	Exported

No items have been found.
Select Actions to create a new item.

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



Select device package and specify connectivity

General

Managed: ☑
Name:
Service Type: ADC
Device Type: [✓ PHYSICAL] VIRTUAL
Physical Domain: select an option
View: ● Single Node ○ HA Node ○ Cluster
Device Package: select a package
Model:
Context Aware: Multiple [✓ Single]

Connectivity

APIC to Device Management Connectivity: ● Out-Of-Band ○ In-Band

Credentials

Username:
Password:
Confirm Password:

Device 1

Management IP Address:
Management Port: enter or select val
Chassis: select a value
Device Interfaces:

Name	Path

Cluster

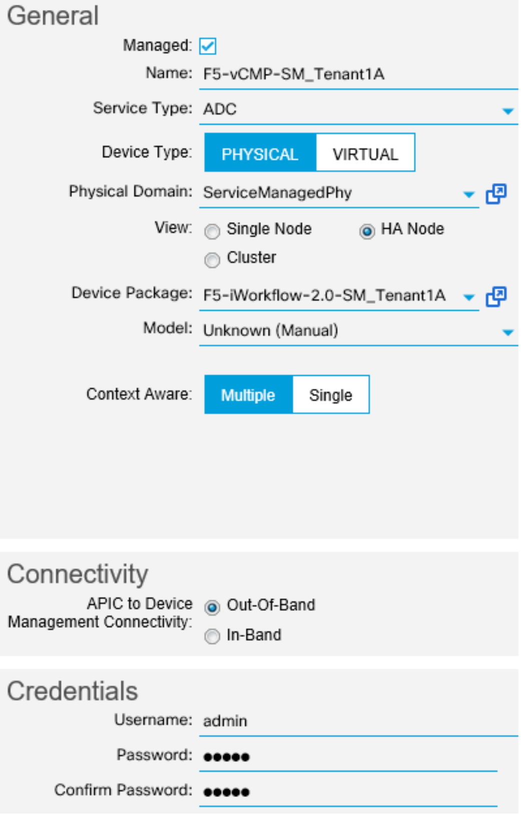
Management IP Address:
Management Port: enter or select val
Device Manager: select a value
Cluster Interfaces:

Type	Name	Concrete Interfaces

In the scenario where a pair of vCMP guests are used as concrete devices for APIC L4-L7 Devices:

Field Description	What does it mean?	Value use in this example
Managed	Managed: this L4-L7 device is managed and configured by Cisco APIC Unmanaged: L4-L7 device configuration is done by user	**Check**
Name	User defined L4-L7 Device name	**F5-vCMP-SM_TenantA**
Service Type	Firewall or ADC. F5 BIG-IP is considered an ADC device	**ADC**
Device Type	Physical or Virtual form factor	**Physical**
Physical Domain	Physical domain contains dynamic service insertion VLAN pool	**ServiceManagedPhy** (configured under APIC Fabric)
Mode	Is L4-L7 Device a Single Node or HA Cluster?	**HA Node**
Device Package	Name of the device package associated with this L4-L7 Device, available from pull-down menu	**F5-iWorkflow-2.0-SM_Tenant1A**
Model	BIG-IP Generic / BIG-IP VE Generic / Unknown: Pre-defined BIG-IP Interface name or manual input	Choose Unknown(Manual) provide flexibility to enter any F5 BIG-IP interface convention
Context Aware	Single Context device can be used by only 1 tenant + VRF; where Multi Context device can be shared among multiple tenants + VRFs	**Multiple**
APIC to Device Management Connectivity	Out-Of-Band or In-Band management	**Out-Of-Band**
Username	BIG-IP admin username	**admin**
Password	BIG-IP admin password	**admin**
Confirmed Password	Confirm BIG-IP admin password	**admin**

After completion, it should look like:



General

Managed: ☑
Name: F5-vCMP-SM_Tenant1A
Service Type: ADC
Device Type: PHYSICAL | VIRTUAL
Physical Domain: ServiceManagedPhy
View: ○ Single Node ● HA Node ○ Cluster
Device Package: F5-iWorkflow-2.0-SM_Tenant1A
Model: Unknown (Manual)
Context Aware: Multiple | Single

Connectivity

APIC to Device Management Connectivity: ● Out-Of-Band ○ In-Band

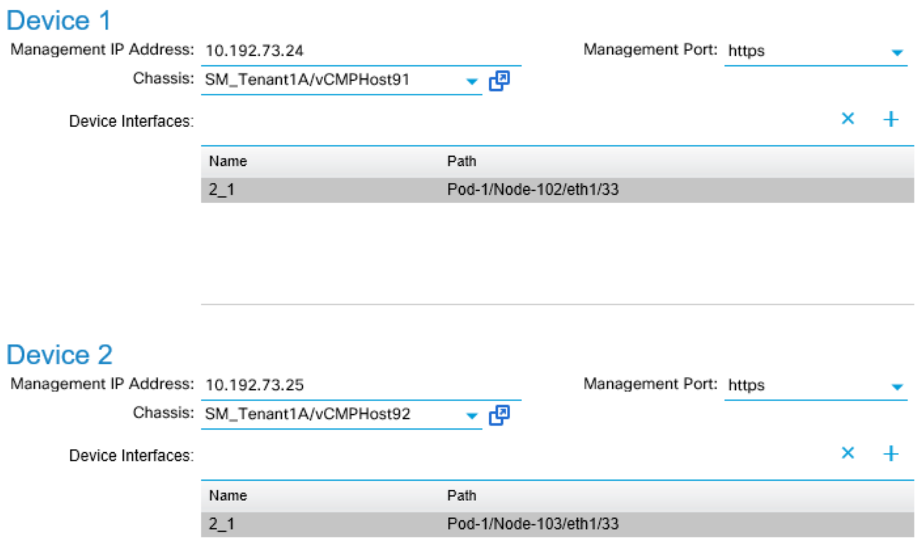
Credentials

Username: admin
Password: ●●●●●
Confirm Password: ●●●●●

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

Field Description	What does it mean?	Value use in this example
Management IP Address	Concrete device management IP address	**10.192.73.24** for vCMP guest 1 **10.192.73.25** for vCMP guest 2
Management Port	HTTP or HTTPS	**HTTPS**
Chassis	Optional: if chassis manager is used, select chassis. In F5 BIG-IP integration, this field identify the vCMP host of the vCMP guest	**vCMPHost91** for vCMP guest 1 **vCMPHost92** for vCMP guest 2
Device Interfaces Name	Specify the physical interface connect between Cisco ACI and BIG-IP	**2_1** for vCMP guest 1 **2_1** for vCMP guest 2 This value must match with BIG-IP interface or trunk name The "Name" is either BIG-IP interface, like 1_1, 1/1_1 or trunk name. Notice that "_" is used instead of "." to specify BIG-IP interface, this is due to APIC use "." as object delimiter
Device Interfaces Path	Specify the Cisco ACI interface (node and interface name) that connect to the device interface	Cisco ACI corresponding VPC

Repeat the same for Device 2, which would be vCMP guest #2:


Device 1

Management IP Address: 10.192.73.24

Management Port: https

Chassis: SM_Tenant1A/vCMPHost91

Device Interfaces:

Name	Path
2_1	Pod-1/Node-102/eth1/33

Device 2

Management IP Address: 10.192.73.25

Management Port: https

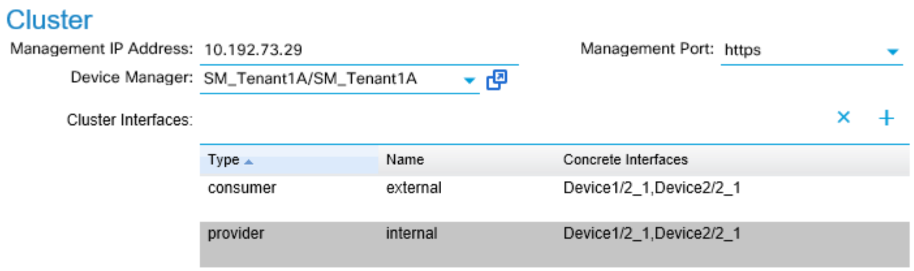
Chassis: SM_Tenant1A/vCMPHost92

Device Interfaces:

Name	Path
2_1	Pod-1/Node-103/eth1/33

Under the Cluster, which specify the L4-L7 device info. Notice Device 1 management IP is pre-populated as the cluster management IP. Change the cluster management IP to one of the iWorkflow management IP. Since Device Manager will be used for this cluster, the cluster management IP will be ignored and "Device Manager" information will be used to establish communication.

Field Description	What does it mean?	Value use in this example
Management IP Address	Concrete device management IP address	**10.192.73.29**
Management Port	HTTP or HTTPS	**HTTPS**
Device Manager	In F5 iWorkflow integration, Device Manager is mandatory and it specify iWorkflow cluster info	**F5-iWorkflow-2.0-SM_Tenant1A**
Cluster Interfaces	Map L4-L7 device logical consumer (external) and provider (internal) interfaces with the device(s) Interface(s), ensuring traffic will flow from abstract layer to rendering layer	In this example, since a single VPC is connected between BIG-IP and APIC, this VPC will serve as both **consumer** and **provider** interfaces.
Type	consumer / provider – APIC key to identify the role of the interface	Create one entry for consumer and one entry for provider
Name	User defined	F5 recommend using external (consumer) and internal (provider) to match F5 terminology
Concrete Interfaces	The physical interfaces of the concrete device (BIG-IP) that will be used for external or internal or both roles	Pick both device 1 and device 2 VPC. Repeat same for provider interface. Available as drop-down menu item


Cluster

Management IP Address: 10.192.73.29

Management Port: https

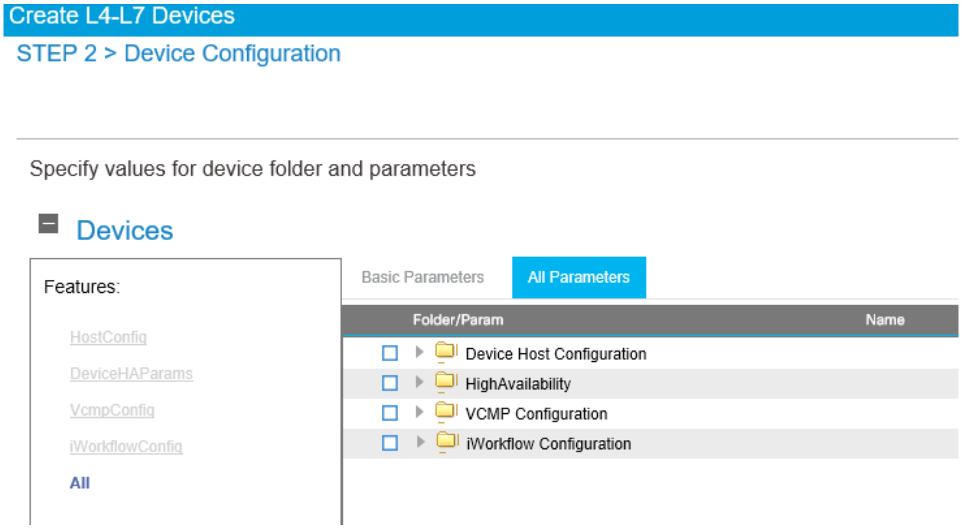
Device Manager: SM_Tenant1A/SM_Tenant1A

Cluster Interfaces:

Type	Name	Concrete Interfaces
consumer	external	Device1/2_1,Device2/2_1
provider	internal	Device1/2_1,Device2/2_1

Click **NEXT** to move the Concrete Device configuration.

Click **ALL PARAMETERS** to enter BIG-IP specific information:


Create L4-L7 Devices

STEP 2 > Device Configuration

Specify values for device folder and parameters

Devices

Features:

- HostConfig
- DeviceHAParams
- VcmpConfig
- iWorkflowConfig
- All

Basic Parameters | All Parameters

Folder/Param	Name
□ Device Host Configuration	
□ HighAvailability	
□ VCMP Configuration	
□ iWorkflow Configuration	

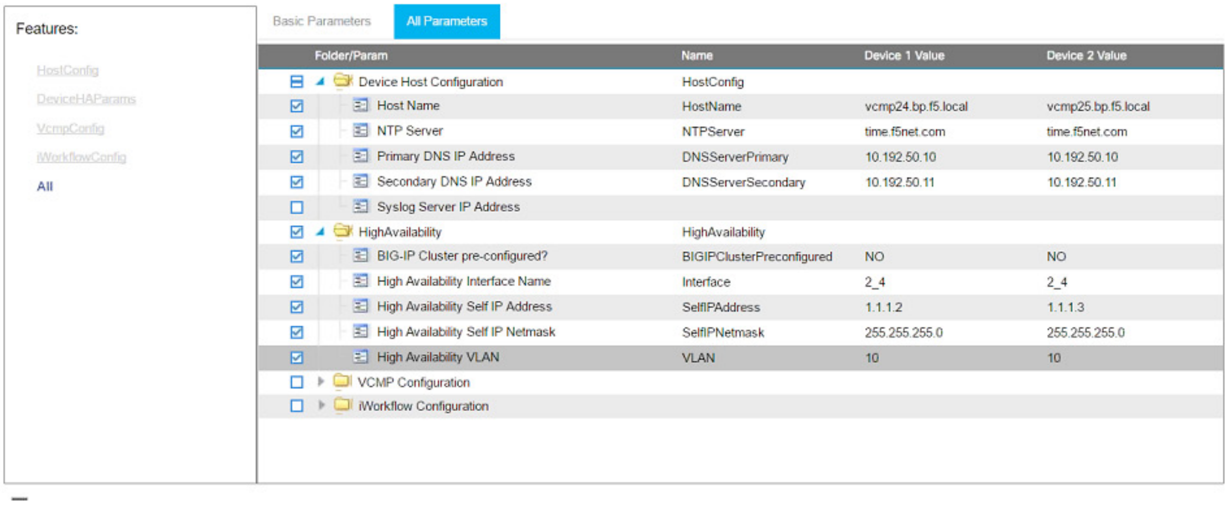
Enter Device Host configuration, like hostname, NTP, DNS information:

Field Description	What does it mean?	Value use in this example
Host Name	BIG-IP host name in FQDN format	**vcmp24.bp.f5.local** **vcmp25.bp.f5.local**
NTP Server	NTP server IP, ensuring APIC, BIG-IP and iWorkflow use the same BTP server	**time.f5net.com**
Primary DNS IP Address	Primary DNS	**10.192.50.10**
Secondary DNS IP Address	Secondary DNS	**10.192.50.11**
Syslog Server IP Address	Syslog Server IP	

Enter High Availability information, like HA interfaces, IP, marks, VLAN (locally significant, not manage by APIC):

Field Description	What does it mean?	Value use in this example
BIG-IP Cluster pre-configured	It means is the BIG-IP cluster to be formed by APIC or outside of APIC	**NO**
High Availability Interface Name	BIG-IP Interface maintain HA heartbeat	**2_4** for both
High Availability Self IP Address	Local significant HA interface IP address	**1.1.1.2** and **1.1.1.3** (one for each BIG-IP)
High Availability Self IP Netmask	HA interface netmask	**255.255.255.0**
High Availability VLAN	HA VLAN (local significant, not visible or managed by APIC)	**10**

Since APIC “Chassis Manager” and “Device Manager” features are being used, leave VCMP and iWorkflow configuration blank:



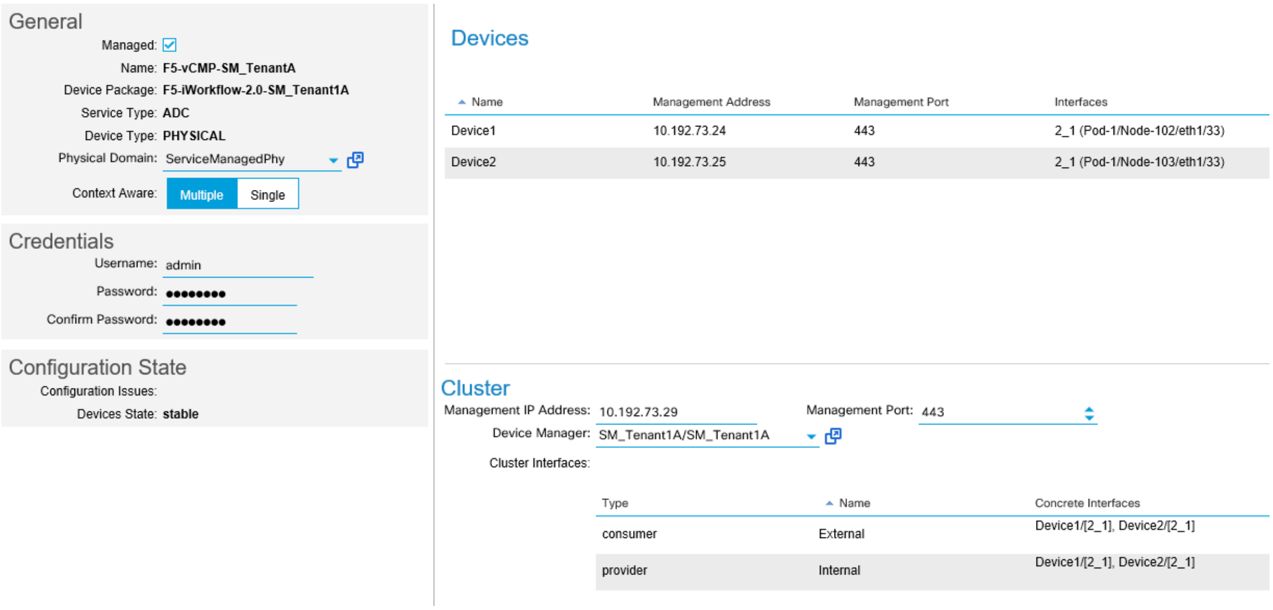
	Folder/Param	Name	Device 1 Value	Device 2 Value
⊟	Device Host Configuration	HostConfig		
☑	Host Name	HostName	vcmp24.bp.f5.local	vcmp25.bp.f5.local
☑	NTP Server	NTPServer	time.f5net.com	time.f5net.com
☑	Primary DNS IP Address	DNSServerPrimary	10.192.50.10	10.192.50.10
☑	Secondary DNS IP Address	DNSServerSecondary	10.192.50.11	10.192.50.11
☐	Syslog Server IP Address			
☑	HighAvailability	HighAvailability		
☑	BIG-IP Cluster pre-configured?	BIGIPClusterPreconfigured	NO	NO
☑	High Availability Interface Name	Interface	2_4	2_4
☑	High Availability Self IP Address	SelfIPAddress	1.1.1.2	1.1.1.3
☑	High Availability Self IP Netmask	SelfIPNetmask	255.255.255.0	255.255.255.0
☑	High Availability VLAN	VLAN	10	10
☐	VCMP Configuration			
☐	iWorkflow Configuration			

Click **FINISH** to complete the L4-L7 Device configuration.

A few minutes may be needed for all the configuration to be completed and the high-availability cluster to become stable. After the configuration is completed, navigating to the newly created L4-L7 Device to verify its Configuration State is stable:

Tenants common -> L4-L7 Services -> L4-L7 Devices -> <L4-L7 Device Name>

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.



General

Managed: ☑
Name: F5-vCMP-SM_TenantA
Device Package: F5-iWorkflow-2.0-SM_Tenant1A
Service Type: ADC
Device Type: PHYSICAL
Physical Domain: ServiceManagedPhy
Context Aware: Multiple | Single

Credentials

Username: admin
Password: ●●●●●●●●
Confirm Password: ●●●●●●●●

Configuration State

Configuration Issues:
Devices State: stable

Devices

Name	Management Address	Management Port	Interfaces
Device1	10.192.73.24	443	2_1 (Pod-1/Node-102/eth1/33)
Device2	10.192.73.25	443	2_1 (Pod-1/Node-103/eth1/33)

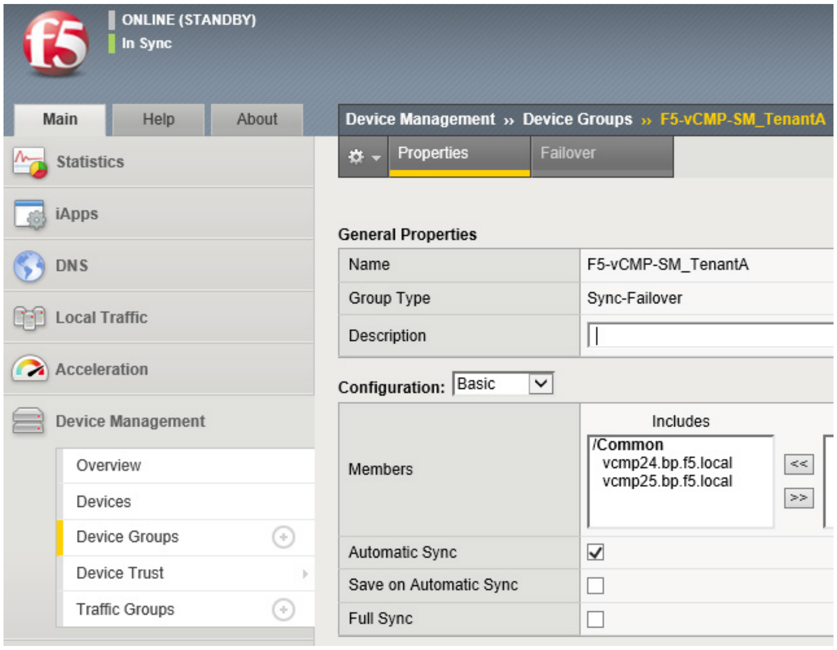
Cluster

Management IP Address: 10.192.73.29
Management Port: 443
Device Manager: SM_Tenant1A/SM_Tenant1A
Cluster Interfaces:

Type	Name	Concrete Interfaces
consumer	External	Device1/[2_1], Device2/[2_1]
provider	Internal	Device1/[2_1], Device2/[2_1]

Log into BIG-IP and confirm that the device group has been formed (it will contain one or two BIG-IP members depending on whether BIG-IP is deployed in standalone mode or high-availability mode).

The device should be online, active, and synchronized (only if you are deploying BIG-IP in high -availability mode).



Device Management » Device Groups » F5-vCMP-SM_TenantA

General Properties

Name	F5-vCMP-SM_TenantA
Group Type	Sync-Failover
Description	

Configuration: Basic

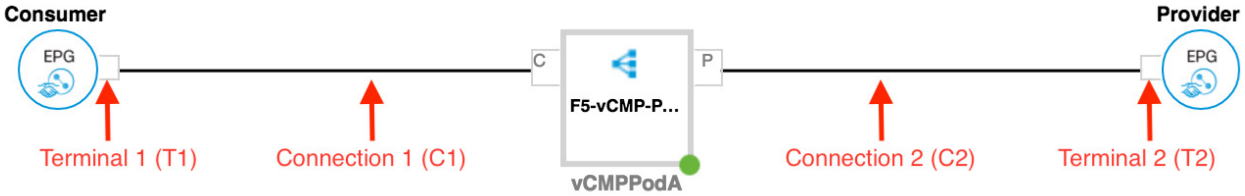
Members	Includes /Common vcmp24.bp.f5.local vcmp25.bp.f5.local
Automatic Sync	☑
Save on Automatic Sync	□
Full Sync	□

APIC: Create Service Graph Template

An APIC L4-L7 Service Graph Template is an abstract object allowing L4-L7 configuration build into ACI policy model. An APIC L4-L7 Service Graph Template that utilize ADC as function node has option to create an one-arm or two-arm graph. APIC ADC one-arm and two-arm graph is a logical construct, user must ensure physical connectivity between Cisco ACI fabric and BIG-IP (physical 1-arm or inline) can support the logical one-arm or two-arm graph.

APIC ADC Graph Type (Logical)	Number of VLANs to be configured on BIG-IP	BIG-IP connect to ACI (Physical)	Supported?
One-Arm	1	1-arm (as a stick)	Yes
		2-arm (inline)	Yes, use Internal Interface
Two-Arm	2 (external and internal)	1-arm (as a stick)	Yes, if link can trunk multiple VLANs
		2-arm (inline)	Yes

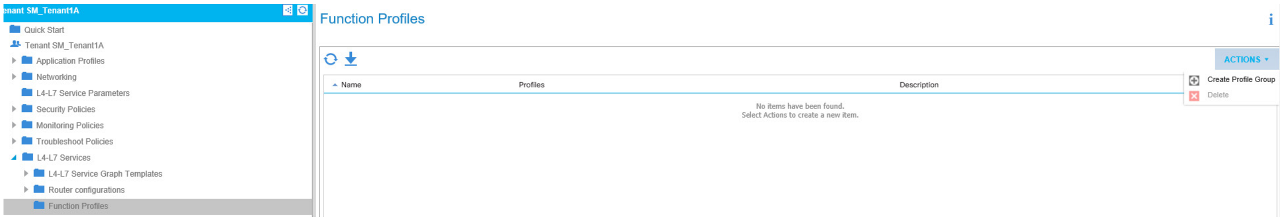
Understanding APIC service graph template, consider a 1-node graph in this example:



To create a new Function Profile, click the following in the navigation pane:

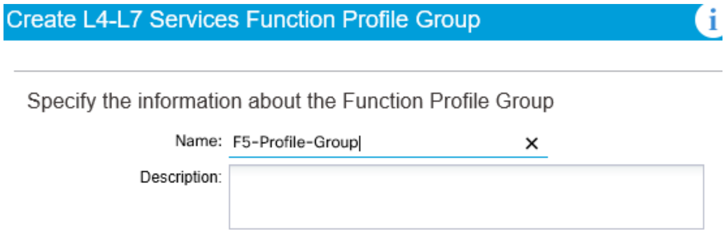
Tenants <user-defined_tenant_name> -> L4-L7 Services -> Function Profiles

Right Click on **Function Profile** and select **Create Profile Group**:



No items have been found.
Select Actions to create a new item.

Enter the name of the function profile group (in this example, F5-Profile-Group), then **SUBMIT**:



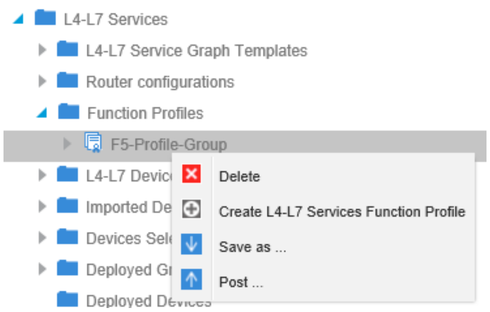
Create L4-L7 Services Function Profile Group

Specify the information about the Function Profile Group

Name: F5-Profile-Group

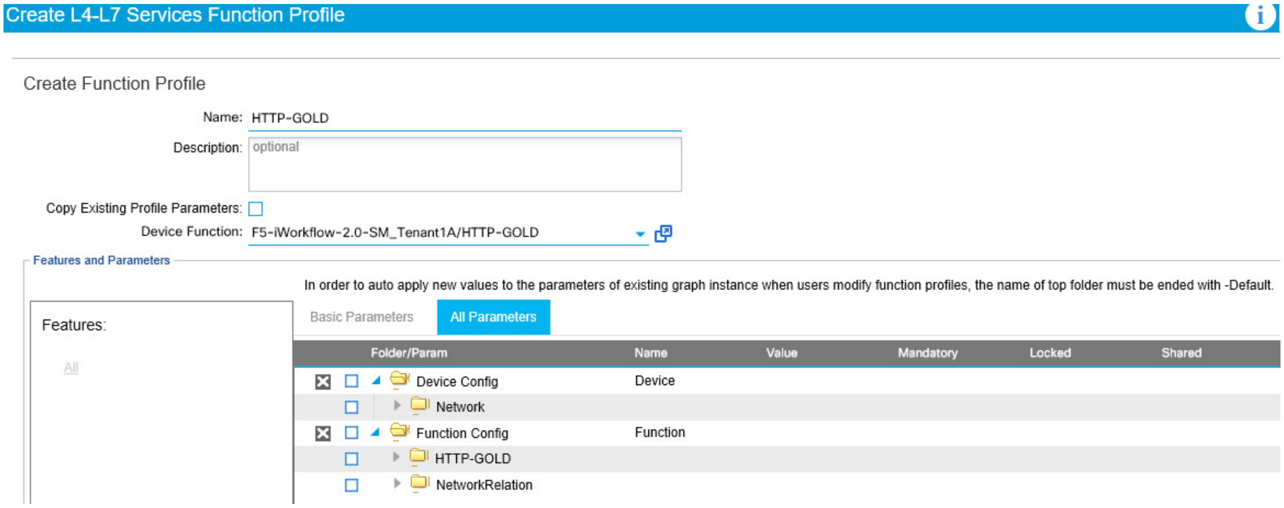
Description:

Navigate to the **F5-Profile-Group**, then right click, select **Create L4-L7 Services Function Profile**:



Field Description	What does it mean?	Value use in this example
Name	User defined name of the function profile	**HTTP-GOLD**
Copy Existing Profile Parameters	If box is checked, pick an existing function profile as base values If box is not checked, pick a service function	**Not Checked**
Profile	Select existing function profile	N/A
Device Function	Select existing service function	**F5-iWorkflow-2.0-SM_Tenant1A/ HTTP-Gold**

Under **Features and Parameters**, select **All Parameters**:


Create L4-L7 Services Function Profile

Create Function Profile

Name: HTTP-GOLD

Description: optional

Copy Existing Profile Parameters: □

Device Function: F5-iWorkflow-2.0-SM_Tenant1A/HTTP-GOLD

Features and Parameters

In order to auto apply new values to the parameters of existing graph instance when users modify function profiles, the name of top folder must be ended with -Default.

Features:

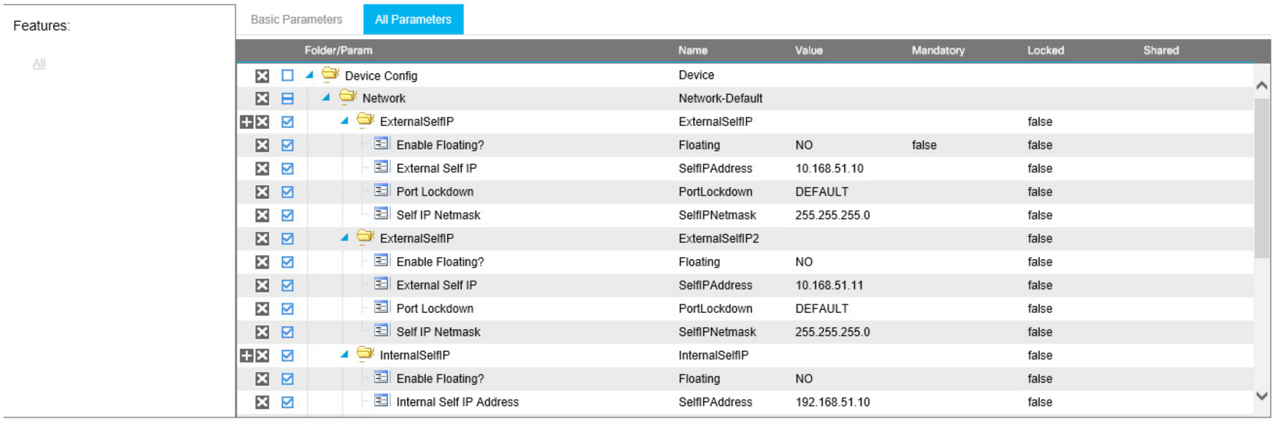
All

Basic Parameters | All Parameters

	Folder/Param	Name	Value	Mandatory	Locked	Shared
☒ □	Device Config	Device				
□	Network					
☒ □	Function Config	Function				
□	HTTP-GOLD					
□	NetworkRelation					

User can now pre-configured parameters for the **HTTP-Gold** service function. This is commonly used to pre-configure network parameters.

Here is an example of a function profile with all networking elements configured ahead of time. You need to add another ExternalSelf2 AND InternalSelf2 interfaces for HA:



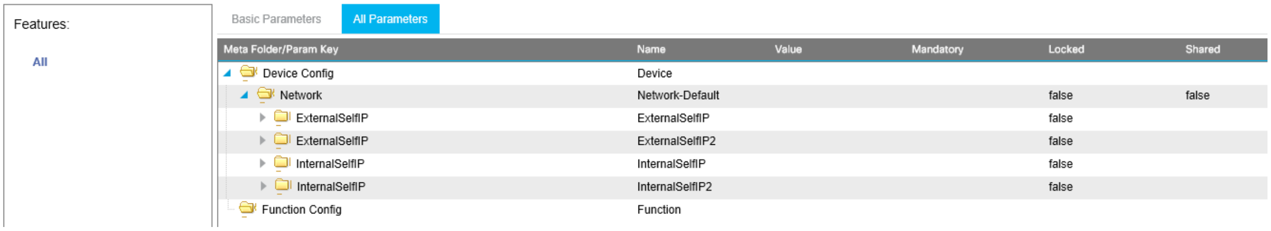
Features:

All

Basic Parameters | All Parameters

	Folder/Param	Name	Value	Mandatory	Locked	Shared
☒ □	Device Config	Device				
☒ ⊟	Network	Network-Default				
⊞ ☒ ☑	ExternalSelfIP	ExternalSelfIP			false	
☒ ☑	Enable Floating?	Floating	NO	false	false	
☒ ☑	External Self IP	SelfIPAddress	10.168.51.10		false	
☒ ☑	Port Lockdown	PortLockdown	DEFAULT		false	
☒ ☑	Self IP Netmask	SelfIPNetmask	255.255.255.0		false	
☒ ☑	ExternalSelfIP	ExternalSelfIP2			false	
☒ ☑	Enable Floating?	Floating	NO		false	
☒ ☑	External Self IP	SelfIPAddress	10.168.51.11		false	
☒ ☑	Port Lockdown	PortLockdown	DEFAULT		false	
☒ ☑	Self IP Netmask	SelfIPNetmask	255.255.255.0		false	
⊞ ☒ ☑	InternalSelfIP	InternalSelfIP			false	
☒ ☑	Enable Floating?	Floating	NO		false	
☒ ☑	Internal Self IP Address	SelfIPAddress	192.168.51.10		false	

FEATURES AND PARAMETERS



Features:

All

Basic Parameters | All Parameters

Meta Folder/Param Key	Name	Value	Mandatory	Locked	Shared
Device Config	Device				
Network	Network-Default			false	false
ExternalSelfIP	ExternalSelfIP			false	
ExternalSelfIP	ExternalSelfIP2			false	
InternalSelfIP	InternalSelfIP			false	
InternalSelfIP	InternalSelfIP2			false	
Function Config	Function				

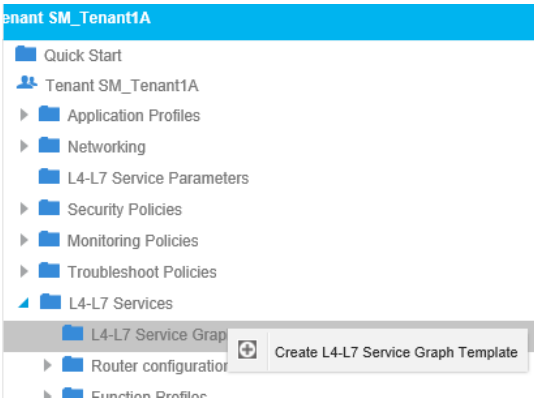
APIC: Create Two-Arm Service Graph Template

Inside the service graph template, user can drag-and-drop the L4-L7 device(s) into the template to provide Firewall or ADC functionality. In this example, the imported L4-L7 device from previous step will be used to provide ADC functionality to a new service graph template. This service graph template is created to provide HTTP-GOLD function for the WEB EPG.

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

Tenants <user-defined_tenant_name> -> L4-L7 Services -> L4-L7 Service Graph Template

Right Click on **L4-L7 Service Graph Template** and select **Create L4-L7 Service Graph Template**:



New pop up window will appear. In the new window, enter the following:

- Graph Name: <HTTP-GOLD>
- Graph Type: Create a New One (should be the default)

Drag the device cluster from the right side of the window into the graph, place it in between the consumer and provider EPG.

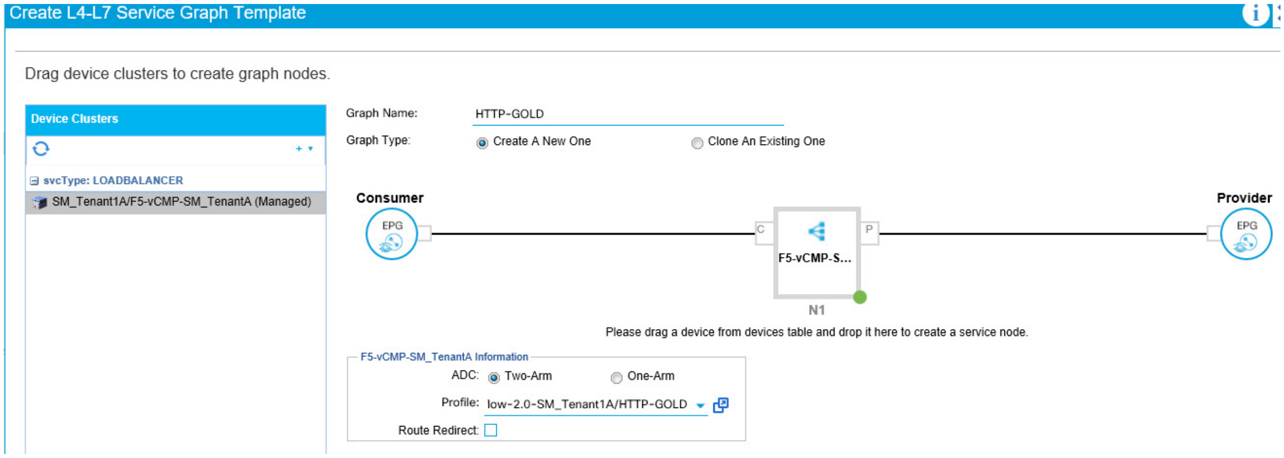
When this graph template is deployed, the traffic will be redirected to the F5 BIG-IP of this device cluster automatically by Cisco ACI.

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

Under **<L4-L7_Device_Name>** Information (in this example, F5-vCMP-SM_Tenanta), click the Two-Arm option for this graph.

- Select the Profile: **F5-iWorkflow-2.0–SM_Tenant1A/HTTP-GOLD**

This is the application template created in iWorkflow:



Create L4-L7 Service Graph Template

Drag device clusters to create graph nodes.

Device Clusters

svcType: LOADBALANCER
SM_Tenant1A/F5-vCMP-SM_TenantA (Managed)

Graph Name: HTTP-GOLD
Graph Type: ● Create A New One ○ Clone An Existing One

Please drag a device from devices table and drop it here to create a service node.

F5-vCMP-SM_TenantA Information
ADC: ● Two-Arm ○ One-Arm
Profile: low-2.0-SM_Tenant1A/HTTP-GOLD
Route Redirect: □

Click **SUBMIT** to complete this task.

APIC: Deploy Two-Arm Service Graph Template

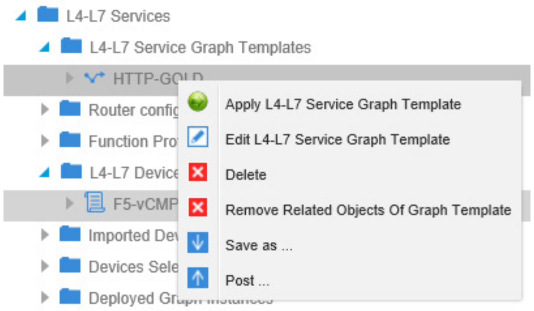
APIC Service Graph Template is considered abstract object, it is not yet deployed between a pair of EPG and no configuration has yet to apply on F5 BIG-IP.

In this example, the service graph template “HTTP-Gold” created in the previous step will be deployed between APIC L3out EPG (clients coming from outside) to the Web EPG (Web tier) with no persistence requirement. Notice the EPG function, in this example Web tier (HTTP), match with the L4-L7 service function “HTTP-Gold” inside the service graph.

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

Tenants <user-defined_tenant_name> -> L4-L7 Services -> L4-L7 Service Graph Template

Select the Service Graph Template **HTTP-No-Persistence-Graph** from the Left-hand-side work pane. Right click and choose the option to: **Apply L4-L7 Service Graph Template**:



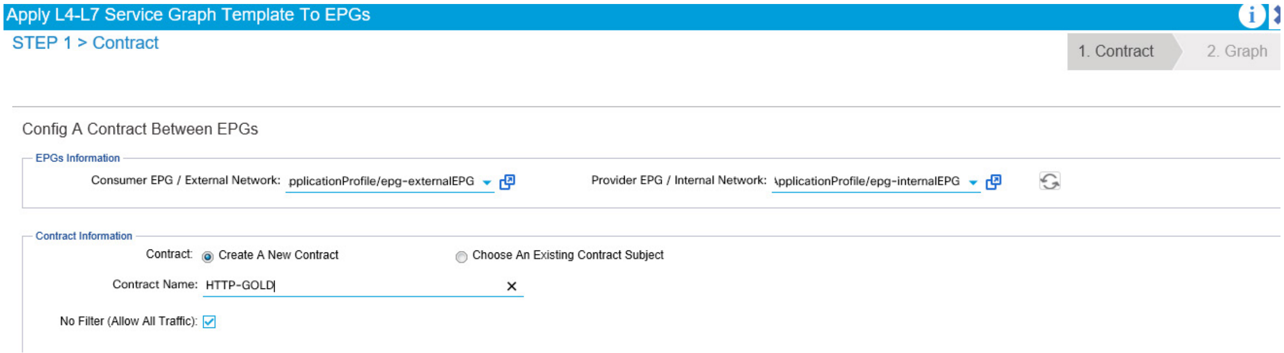
In the new window, select the EPGs the Service Graph will be inserted in between.

Select the following for the EPG information:

Field Description	What does it mean?	Value use in this example
Consumer EPG / External Network	this EPG consume service	**SM_Tenant1A/ApplicationProfile/epg-externalEGP**
Provider EPG / External Network	this EPG provide service	**SM_Tenant1A/ApplicationProfile/epg-internalEGP**

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

Field Description	What does it mean?	Value use in this example
Contract	Using a new contract or existing contract	CHECK **Create A New Contract**
Contract Name	User defined contract Name	**HTTP-GOLD**
No Filter (Allow All Traffic)	Does this contract allow all traffic?	**CHECK**


Apply L4-L7 Service Graph Template To EPGs

STEP 1 > Contract

1. Contract 2. Graph

Config A Contract Between EPGs

EPGs Information

Consumer EPG / External Network: pplicationProfile/epg-externalEPG

Provider EPG / Internal Network: \pplicationProfile/epg-internalEPG

Contract Information

Contract: ● Create A New Contract ○ Choose An Existing Contract Subject

Contract Name: HTTP-GOLD

No Filter (Allow All Traffic): ☑

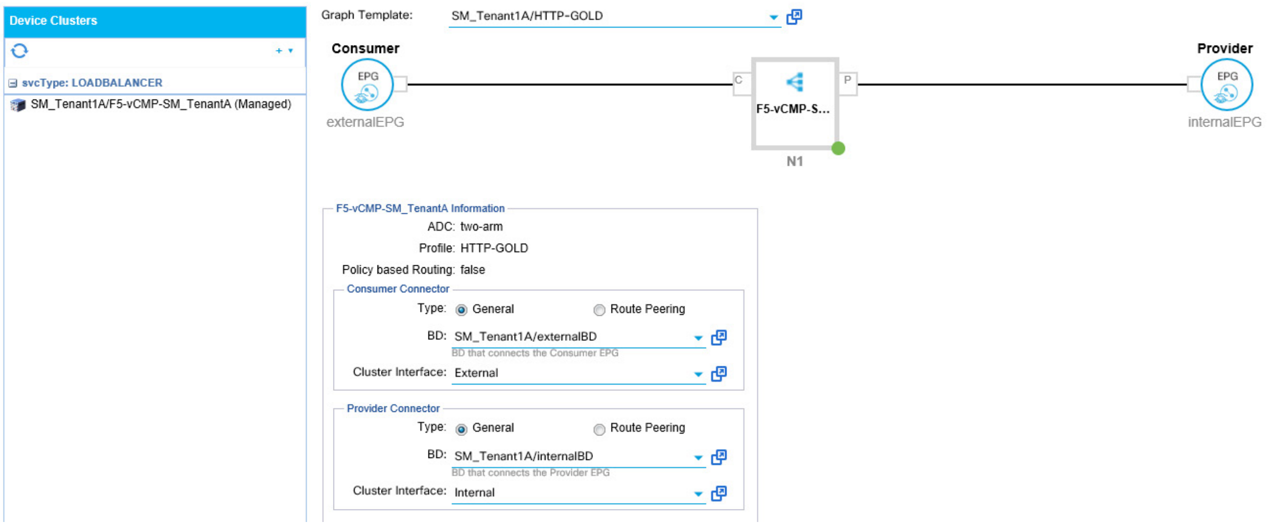
Click **NEXT** to go to STEP 2.

STEP 2, user can apply Service Graph specific parameters. For Two-Arm graph, there is 2 connectors, user needs to map the connector with the L4-L7 Device cluster interface, this is particularly important, as the graph connector is associated with a APIC bridge domain (BD), which contains subnet information. This mapping will provide connectivity between the BIG-IP and backend servers, ensuring BIG-IP monitor to work as expected.

Under Connector:

Field Description	What does it mean?	Value use in this example
Type	General / Route Peering – select “General” for BD, “Route Peering” for L3 Ext Net	**General**
BD (if Type is General)	the bridge domain that connects the two devices	**SM_Tenant1A/externalBD/ SM_Tenant1A/ internalBD**
L3 Ext Network (if Type is Route Peering)	select L3 network for dynamic routing	N/A
Cluster Interface	map L4-L7 Device cluster interface (external or internal) to the connector	**External/Internal**

Config A Service Graph



Graph Template: SM_Tenant1A/HTTP-GOLD

F5-vCMP-SM_TenantA Information

ADC: two-arm
Profile: HTTP-GOLD
Policy based Routing: false

Consumer Connector

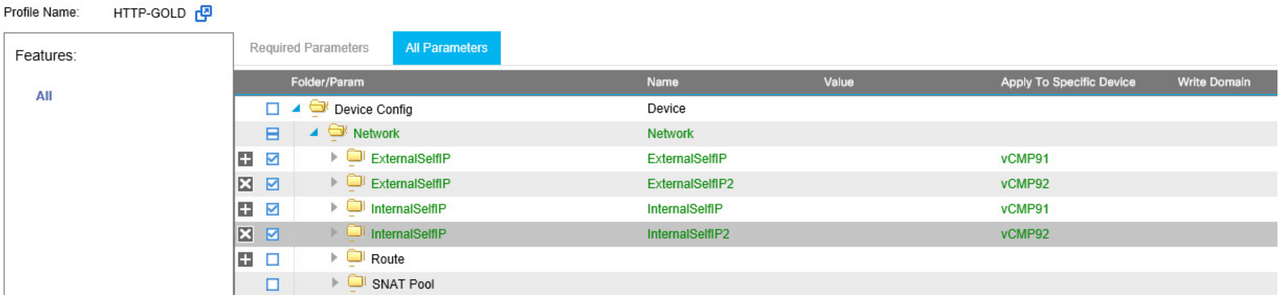
Type: ● General ○ Route Peering
BD: SM_Tenant1A/externalBD
BD that connects the Consumer EPG
Cluster Interface: External

Provider Connector

Type: ● General ○ Route Peering
BD: SM_Tenant1A/internalBD
BD that connects the Provider EPG
Cluster Interface: Internal

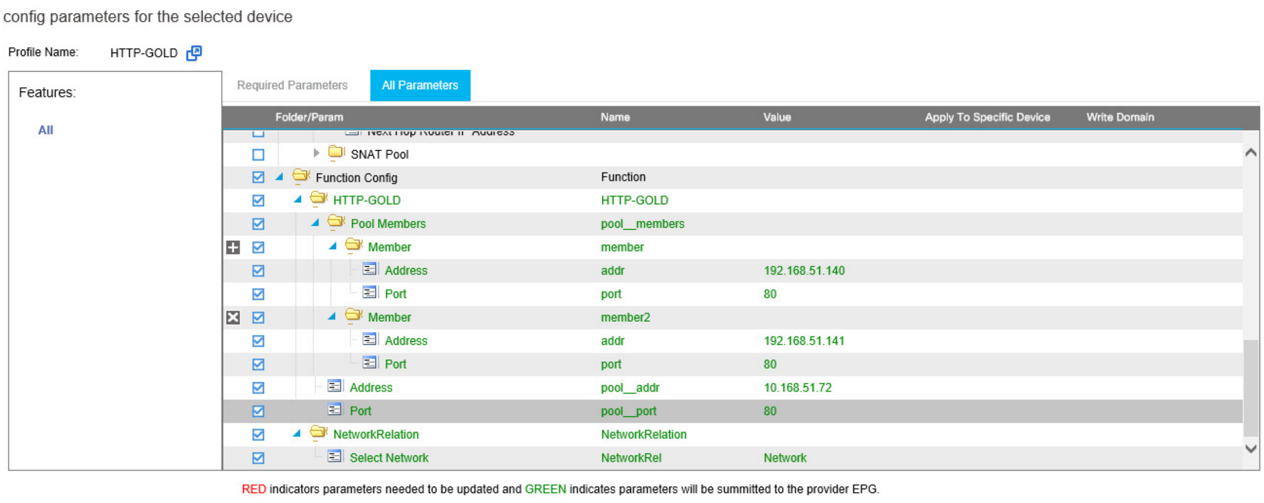
Click **NEXT** to go to STEP 3.

STEP 3, graph specific parameters for the L4-L7 Device. In this step, user can enter F5 virtual server specific parameters based on the template of the iWorkflow service catalog.



Profile Name: HTTP-GOLD

Folder/Param	Name	Value	Apply To Specific Device	Write Domain
□ Device Config	Device			
⊟ Network	Network			
☑ ExternalSelfIP	ExternalSelfIP		vCMP91	
☑ ExternalSelfIP	ExternalSelfIP2		vCMP92	
☑ InternalSelfIP	InternalSelfIP		vCMP91	
☑ InternalSelfIP	InternalSelfIP2		vCMP92	
□ Route				
□ SNAT Pool				



config parameters for the selected device

Profile Name: HTTP-GOLD

Features:

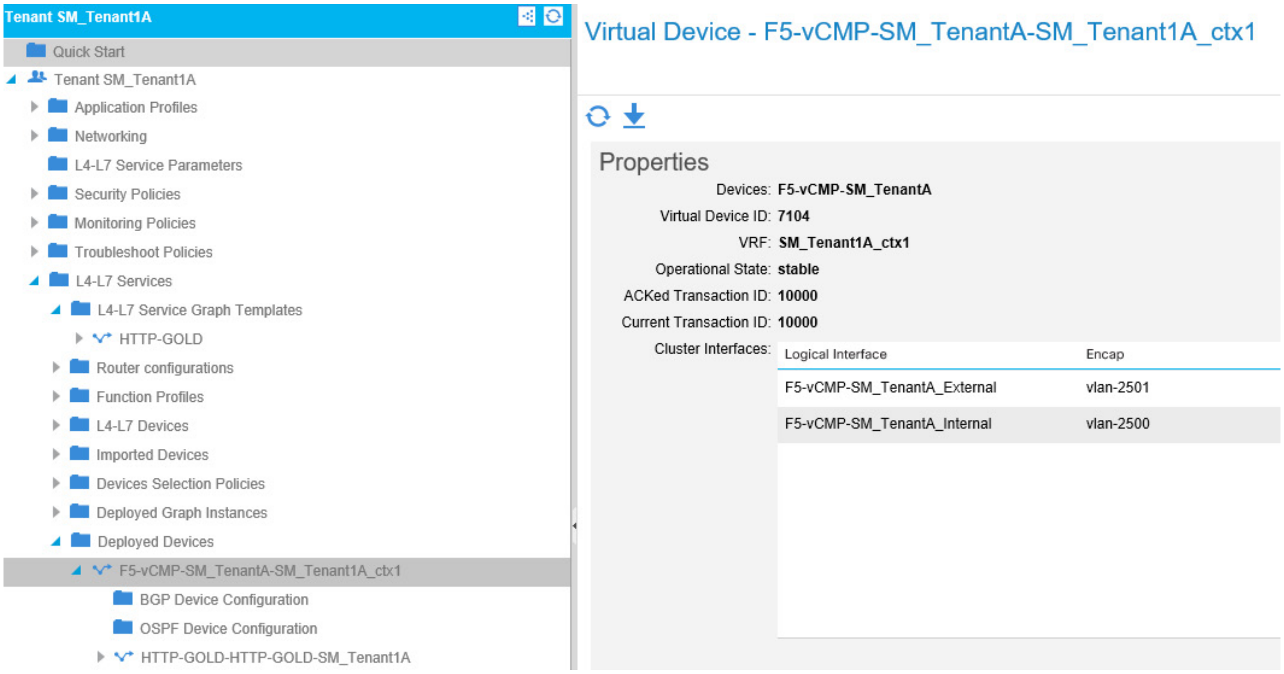
All

		Folder/Param	Name	Value	Apply To Specific Device	Write Domain
	□	Next Hop Router IP Address				
	□	SNAT Pool				
	☑	Function Config	Function			
	☑	HTTP-GOLD	HTTP-GOLD			
	☑	Pool Members	pool__members			
⊞	☑	Member	member			
	☑	Address	addr	192.168.51.140		
	☑	Port	port	80		
☒	☑	Member	member2			
	☑	Address	addr	192.168.51.141		
	☑	Port	port	80		
	☑	Address	pool__addr	10.168.51.72		
	☑	Port	pool__port	80		
	☑	NetworkRelation	NetworkRelation			
	☑	Select Network	NetworkRel	Network		

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

Click **FINISH** to deploy the graph.

Cisco APIC assign 2 VLAN for 2-arm graph deployment:



Tenant SM_Tenant1A

- Quick Start
- Tenant SM_Tenant1A
  - Application Profiles
  - Networking
  - L4-L7 Service Parameters
  - Security Policies
  - Monitoring Policies
  - Troubleshoot Policies
  - L4-L7 Services
    - L4-L7 Service Graph Templates
      - HTTP-GOLD
    - Router configurations
    - Function Profiles
    - L4-L7 Devices
    - Imported Devices
    - Devices Selection Policies
    - Deployed Graph Instances
    - Deployed Devices
      - F5-vCMP-SM_TenantA-SM_Tenant1A_ctx1
        - BGP Device Configuration
        - OSPF Device Configuration
        - HTTP-GOLD-HTTP-GOLD-SM_Tenant1A

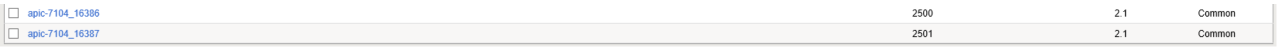
Virtual Device - F5-vCMP-SM_TenantA-SM_Tenant1A_ctx1

Properties

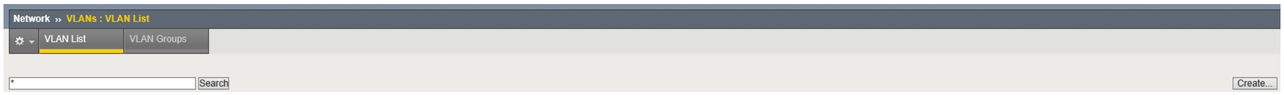
Devices: **F5-vCMP-SM_TenantA**
Virtual Device ID: **7104**
VRF: **SM_Tenant1A_ctx1**
Operational State: **stable**
ACKed Transaction ID: **10000**
Current Transaction ID: **10000**
Cluster Interfaces:

Logical Interface	Encap
F5-vCMP-SM_TenantA_External	vlan-2501
F5-vCMP-SM_TenantA_Internal	vlan-2500

F5 BIG-IP VLAN matches with Cisco APIC:

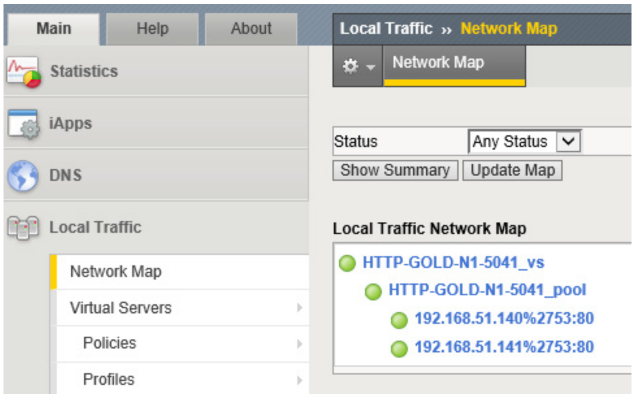


□	apic-7104_16386	2500	2.1	Common
□	apic-7104_16387	2501	2.1	Common



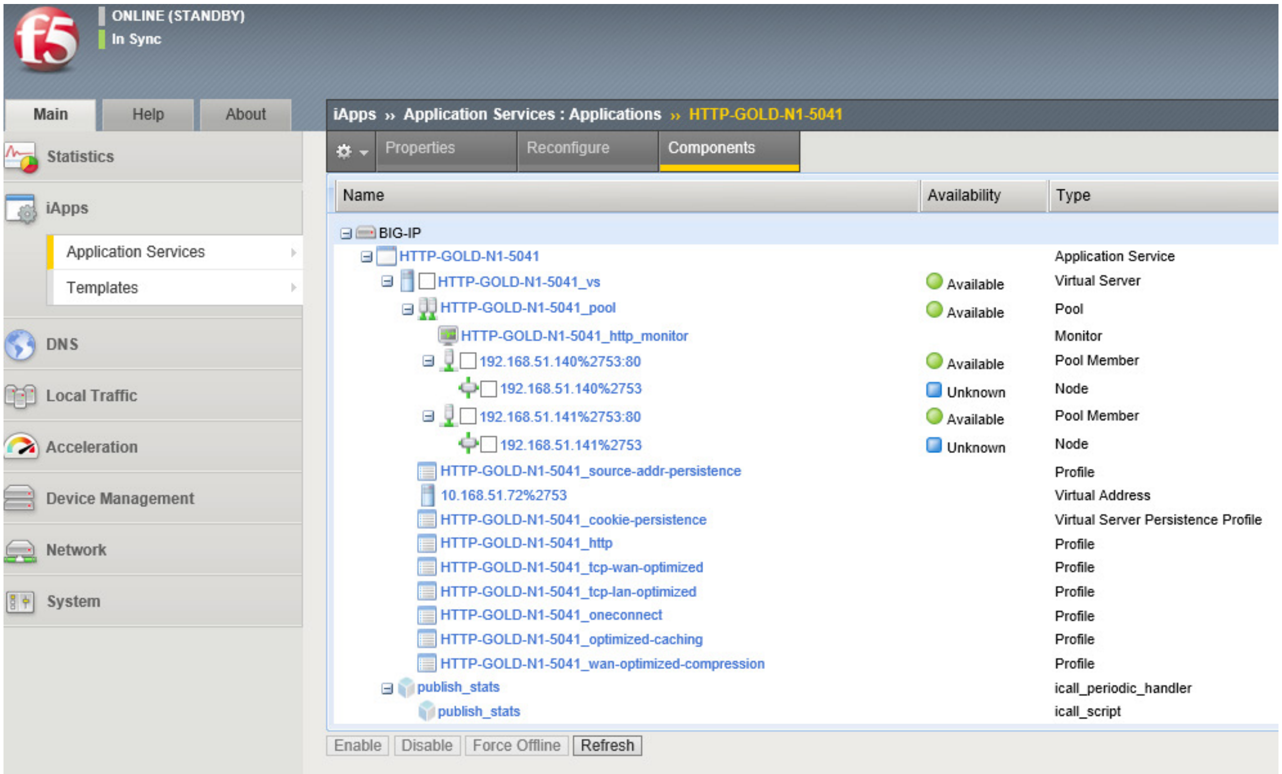
Network » VLANs : VLAN List

When virtual server is initially deployed in BIG-IP, it is unavailable due to the Pool is empty:



Local Traffic Network Map

- HTTP-GOLD-N1-5041_vs
  - HTTP-GOLD-N1-5041_pool
    - 192.168.51.140%2753:80
    - 192.168.51.141%2753:80

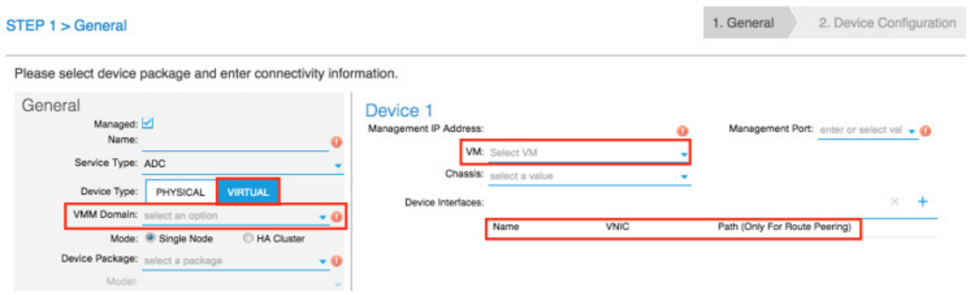


Name	Availability	Type
BIG-IP		
HTTP-GOLD-N1-5041		Application Service
HTTP-GOLD-N1-5041_vs	Available	Virtual Server
HTTP-GOLD-N1-5041_pool	Available	Pool
HTTP-GOLD-N1-5041_http_monitor		Monitor
192.168.51.140%2753:80	Available	Pool Member
192.168.51.140%2753	Unknown	Node
192.168.51.141%2753:80	Available	Pool Member
192.168.51.141%2753	Unknown	Node
HTTP-GOLD-N1-5041_source-addr-persistence		Profile
10.168.51.72%2753		Virtual Address
HTTP-GOLD-N1-5041_cookie-persistence		Virtual Server Persistence Profile
HTTP-GOLD-N1-5041_http		Profile
HTTP-GOLD-N1-5041_tcp-wan-optimized		Profile
HTTP-GOLD-N1-5041_tcp-lan-optimized		Profile
HTTP-GOLD-N1-5041_oneconnect		Profile
HTTP-GOLD-N1-5041_optimized-caching		Profile
HTTP-GOLD-N1-5041_wan-optimized-compression		Profile
publish_stats		icall_periodic_handler
publish_stats		icall_script

Repeat the same step to create APIC L4-L7 Device using F5 BIG-IP Virtual Edition (VE), there are some slight differences in the configuration where the BIG-IP virtual machines and vCenter information must be provided when creating the L4-L7 Devices.

APIC chassis manager feature does not apply to VE.

In this example, creating an APIC L4-L7 Device using a single BIG-IP VE as concrete device. APIC L4-L7 Device Type Virtual:



STEP 1 > General

1. General | 2. Device Configuration

Please select device package and enter connectivity information.

General

Managed: ☑
Name:
Service Type: ADC
Device Type: PHYSICAL | VIRTUAL
VMM Domain: select an option
Mode: ● Single Node ○ HA Cluster
Device Package: select a package
Model:

Device 1

Management IP Address:
VM: Select VM
Chassis: select a value
Management Port: enter or select val
Device Interfaces:

Name	VNIC	Path (Only For Route Peering)

Repeat the same step to create APIC L4-L7 Service Graph Template is an abstract object allowing L4-L7 configuration build into ACI policy model.

Cisco ACI Network Policy Mode (Unmanaged)

In the Unmanaged Mode deployment model, F5 BIG-IP is not managed by Cisco APIC. All F5 BIG-IP configurations must be completed by the user, including network and virtual server configurations.

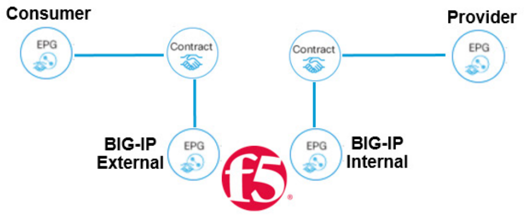
In unmanaged mode, there is no device package requirement, as a result, no service insertion.

Regardless service insertion or unmanaged mode, physical connectivity between F5 BIG-IP and Cisco ACI of using single port, trunk or VPC is supported.

Prior to APIC release 1.2(1*), EPG mode is used to attach F5 BIG-IP to ACI fabric without service insertion. There are subtle differences between EPG mode and Unmanaged mode, Figure below illustrate the differences between the two models. We are ONLY discussing unmanaged mode in this document, we will NOT be covering deployment of EPG mode.

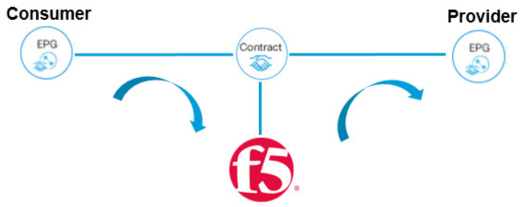
EPG Mode

- **No service graph representation**
- **Manual bindings of VLAN and contract to EPG**
- **Manual configuration to steer service traffic**
- **EPG and BD assignment to BIG-IP**
- **2 contracts:**
  - Consumer EPG -> BIG-IP EPG
  - BIG-IP EPG -> Provider EPG
- **No APIC L4-L7 objects requires**



Unmanaged Mode

- **Service graph representation**
- **Automatic bindings of VLAN and contract**
- **Automatic steer service traffic**
- **BIG-IP configure as unmanaged ACI L4-L7 device, only connectivity to ACI is specified**
- **1 contract**
- **Requires APIC L4-L7 objects**



Deployment Workflow (App tier to Web tier)

1. APIC: Create Unmanaged L4-L7 Device
2. APIC: Create Service Graph Template using Unmanaged Device
3. APIC: Deploy Service Graph Template using Unmanaged Device
4. BIG-IP: Configure Network Parameters
5. BIG-IP: Configure Virtual Servers

APIC: Create Unmanaged L4-L7 Device

Due to the nature that unmanaged device, VLANs are statically bind, therefore, unmanaged L4-L7 device should be created under user-defined tenant.

As each customer L2-L3 network requirements are different, please consult Cisco ACI team on network design and the necessary network configuration for network stitching between Cisco ACI to F5 BIG-IP to server farm.

Similar to non-ACI environment, in unmanaged mode, F5 BIG-IP administrator would expect VLAN tag and subnet / IP addresses information from the network administrator.

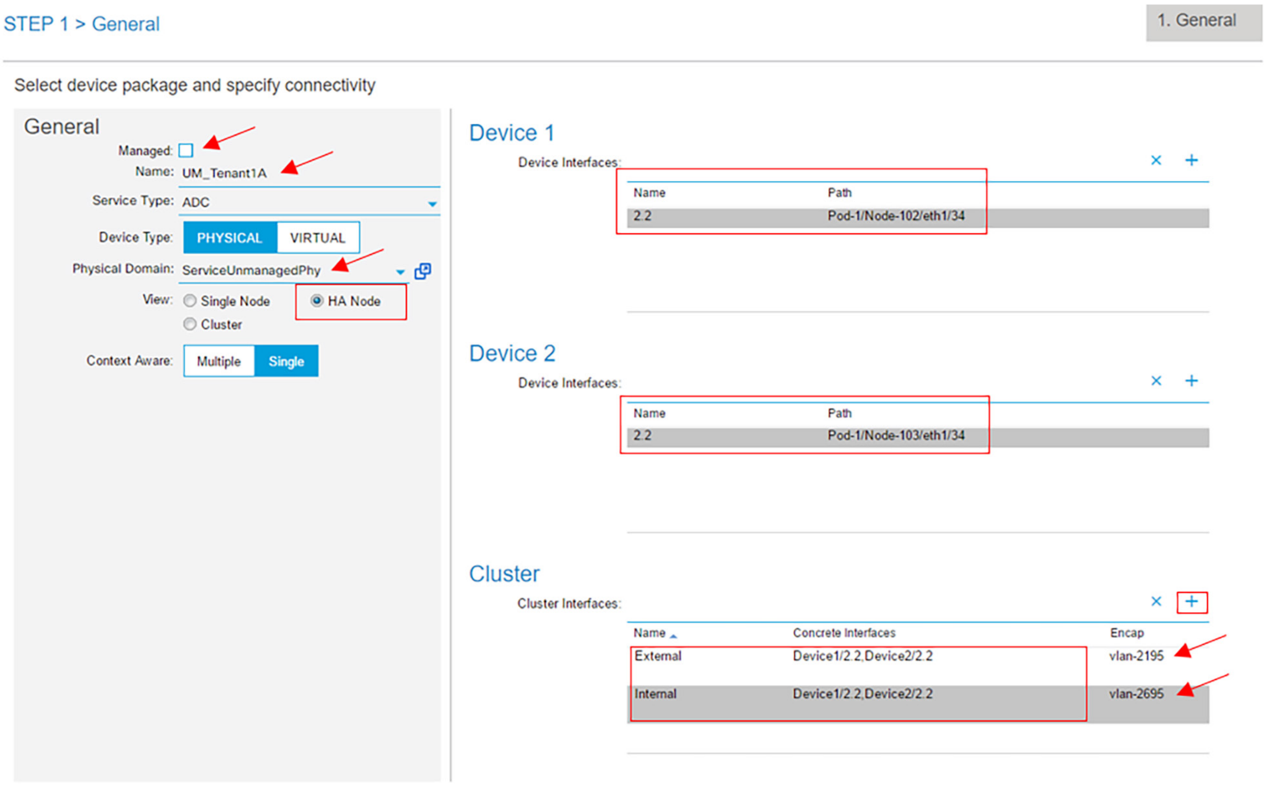
Create Unmanaged L4-L7 device under user-defined (UM_Tenant1A) tenant, go to:

Tenant UM_Tenant1A -> L4-L7 Services -> L4-L7 Devices

A new wizard pop up:

Field Description	What does it mean?	Value use in this example
Managed	Managed: this L4-L7 device is managed and configured by Cisco APIC Unmanaged: L4-L7 device configuration is done by user	**UNCHECK**
Name	User defined L4-L7 Device name	**UM_Tenant1A**
Service Type	Firewall or ADC F5 BIG-IP is considered an ADC device	**ADC**
Device Type	Physical or Virtual form factor	**Physical**
Physical Domain	Physical domain contains static VLAN pool for L4-L7 service in unmanaged mode	**ServiceUnmanagedPhy** (pre-configured under APIC Fabric)
Mode	Is L4-L7 Device a Single Node or HA Cluster?	**HA Cluster**
Function Type	GoTo or GoThrough GoTo: L4-L7 device as next-hop GoThrough: L4-L7 device act like bump-in-the-wire	**GoTo**
Device Interfaces Name	F5 BIG-IP Interface connect to ACI	**2.2**
Device Interface Path	ACI fabric interface connecting to the corresponding BIG-IP interface	Node-102/eth1/34 (Device 1) Node-103/eth1/34 (Device 2)
Cluster Interface Name	User-defined	External and Internal

Concrete Interfaces	Which device interface will be used for external and interface traffic?	Select both Device 1 and Device 2 interface – since only 1 link between ACI and BIG-IP, this link serve both external and internal traffic
Encap	Static binding VLAN information	vlan-2695 – internal VLAN value APIC send to BIG-IP vlan-2195 – external VLAN value APIC send to BIG-IP



STEP 1 > General

1. General

Select device package and specify connectivity

General

Managed: □

Name: UM_Tenant1A

Service Type: ADC

Device Type: [✓ PHYSICAL] VIRTUAL

Physical Domain: ServiceUnmanagedPhy

View: ○ Single Node ● HA Node ○ Cluster

Context Aware: Multiple [✓ Single]

Device 1

Device Interfaces:

Name	Path
2.2	Pod-1/Node-102/eth1/34

Device 2

Device Interfaces:

Name	Path
2.2	Pod-1/Node-103/eth1/34

Cluster

Cluster Interfaces:

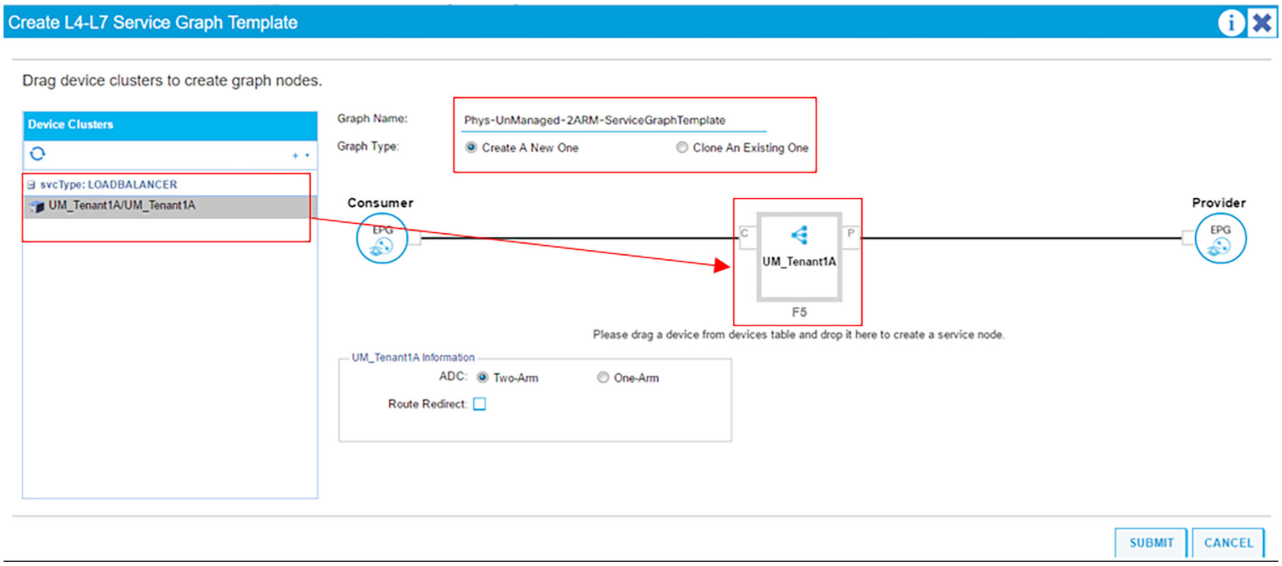
Name	Concrete Interfaces	Encap
External	Device1/2.2,Device2/2.2	vlan-2195
Internal	Device1/2.2,Device2/2.2	vlan-2695

Click **FINISH**.

Notice that no BIG-IP information (management IP, login, password, etc.) is provided. Only connectivity is defined in unmanaged L4-L7 Device.

APIC: Create Service Graph Template using Unmanaged Device

Create an unmanaged two-arm service graph—**Phys-UnManaged-2ARM-ServiceGraphTemplate**



Create L4-L7 Service Graph Template

Drag device clusters to create graph nodes.

Device Clusters

⊟ svcType: LOADBALANCER

UM_Tenant1A/UM_Tenant1A

Graph Name: Phys-UnManaged-2ARM-ServiceGraphTemplate

Graph Type: ● Create A New One ○ Clone An Existing One

Please drag a device from devices table and drop it here to create a service node.

UM_Tenant1A Information

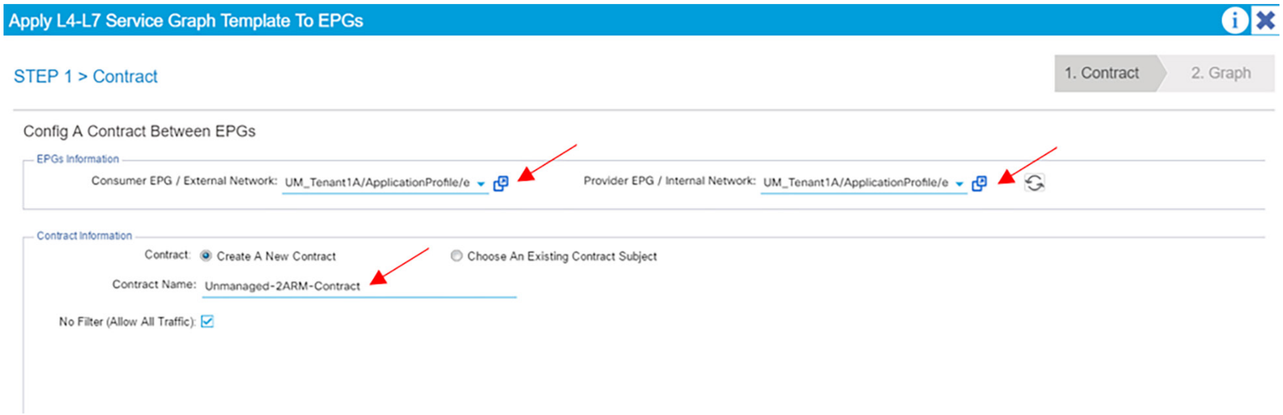
ADC: ● Two-Arm ○ One-Arm

Route Redirect: □

APIC: Deploy Service Graph Template using Unmanaged Device

Deploy the graph template between two EPG

- Consumer: externalEPG (Application Tier)
- Provider: internalEPG (Web Tier)
- New Contract: Unmanaged-2ARM-Contract:



Apply L4-L7 Service Graph Template To EPGs

STEP 1 > Contract

Config A Contract Between EPGs

EPGs Information

Consumer EPG / External Network: UM_Tenant1A/ApplicationProfile/e

Provider EPG / Internal Network: UM_Tenant1A/ApplicationProfile/e

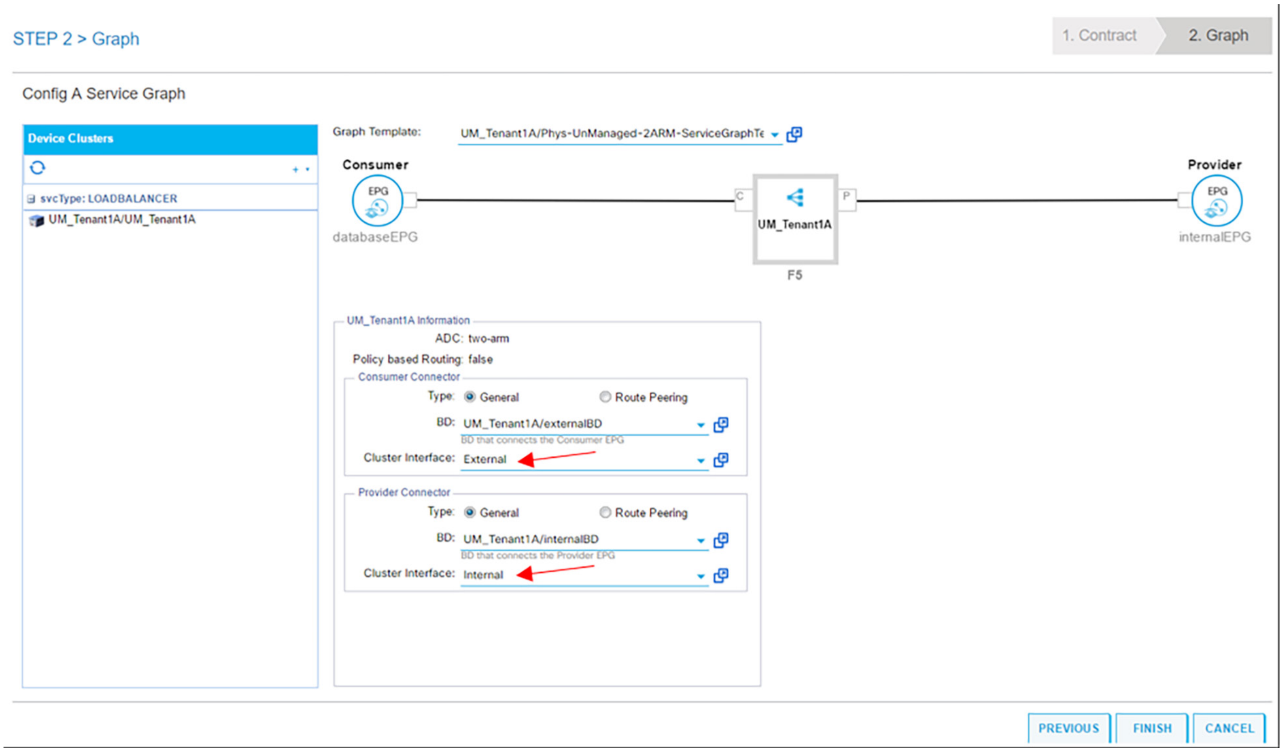
Contract Information

Contract: ● Create A New Contract ○ Choose An Existing Contract Subject

Contract Name: Unmanaged-2ARM-Contract

No Filter (Allow All Traffic): ☑

Map cluster interface with the BD. **Note:** Different network topologies and requirements will result in different network settings, please consult Cisco ACI team.



STEP 2 > Graph

Config A Service Graph

Graph Template: UM_Tenant1A/Phys-UnManaged-2ARM-ServiceGraphTe

Device Clusters: svcType: LOADBALANCER — UM_Tenant1A/UM_Tenant1A

UM_Tenant1A Information

ADC: two-arm

Policy based Routing: false

Consumer Connector

Type: ● General ○ Route Peering

BD: UM_Tenant1A/externalBD (BD that connects the Consumer EPG)

Cluster Interface: External

Provider Connector

Type: ● General ○ Route Peering

BD: UM_Tenant1A/internalBD (BD that connects the Provider EPG)

Cluster Interface: Internal

Click **FINISH**

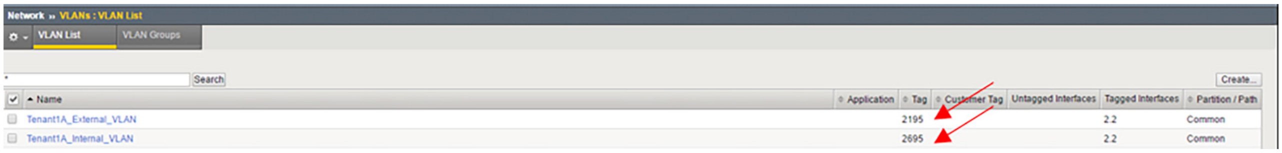
Notice, there is no F5 BIG-IP related configuration.

The above steps will complete the network stitching on the ACI side, where VLAN tags, in this example 2195 and 2695, will be used in the external and internal paths that connect to the BIG-IP.

BIG-IP: Configure Network Parameters

F5 BIG-IP administrator need to configure both network and virtual server elements on BIG-IP.

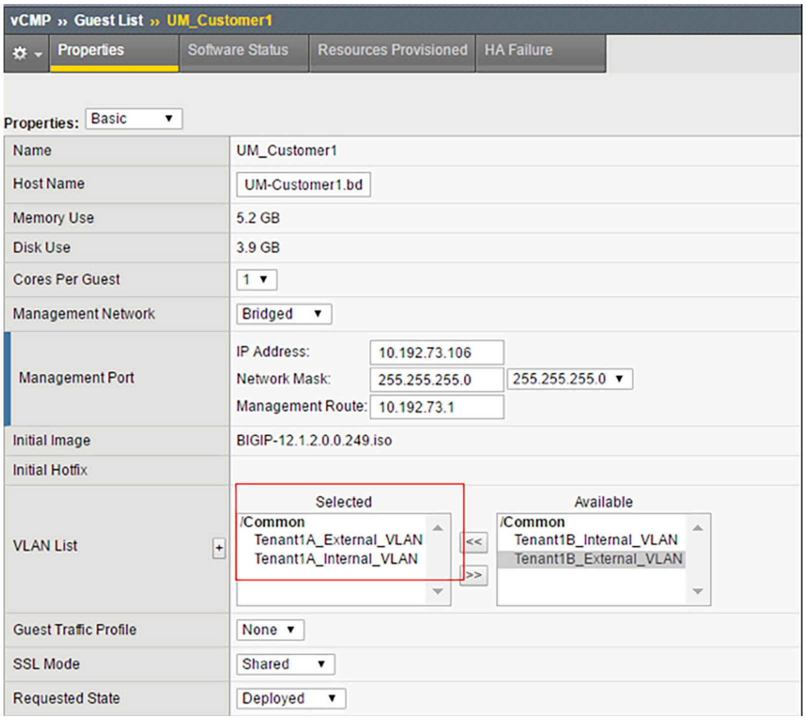
In this example, vCMP is used, create VLAN on vCMP host. Notice the tag matches with APIC VLAN encap value, also tagging on the uplink that connect BIG-IP to ACI:



Network » VLANs : VLAN List

Name	Application	Tag	Customer Tag	Untagged Interfaces	Tagged Interfaces	Partition / Path
Tenant1A_External_VLAN		2195			2.2	Common
Tenant1A_Internal_VLAN		2695			2.2	Common

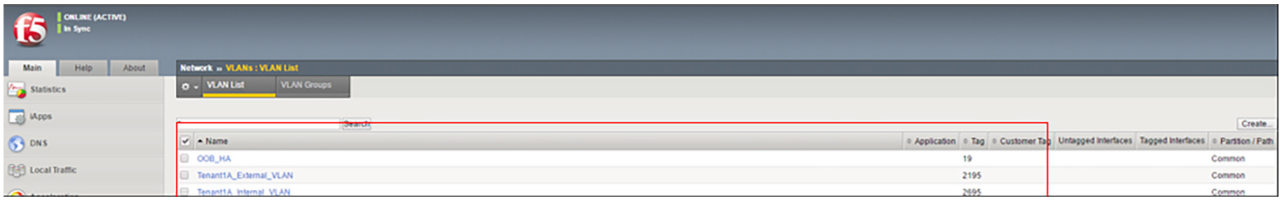
Add this VLAN to the vCMP guest:



Properties: Basic

Name	UM_Customer1
Host Name	UM-Customer1.bd
Memory Use	5.2 GB
Disk Use	3.9 GB
Cores Per Guest	1
Management Network	Bridged
Management Port	IP Address: 10.192.73.106 Network Mask: 255.255.255.0 255.255.255.0 Management Route: 10.192.73.1
Initial Image	BIGIP-12.1.2.0.0.249.iso
Initial Hotfix	
VLAN List	Selected: /Common Tenant1A_External_VLAN Tenant1A_Internal_VLAN Available: /Common Tenant1B_Internal_VLAN Tenant1B_External_VLAN
Guest Traffic Profile	None
SSL Mode	Shared
Requested State	Deployed

vCMP guest VLAN list will have VLAN information from vCMP host:



Name	Application	Tag	Customer Tag	Untagged Interfaces	Tagged Interfaces	Partition / Path
OOB_HA		19				Common
Tenant1A_External_VLAN		2195				Common
Tenant1A_Internal_VLAN		2695				Common

Repeat the same for standby BIG-IP

High availability on the BIG-IP is setup out of band. Configure external and internal self IPs and floating self IPs based on network requirements and subnet information from network administrator.

Upon completion, the network traffic re-direction that stitches consumer EPG <-> Cisco ACI <-> F5 BIG-IP <-> provider EPG is established.

BIG-IP: Configure Virtual Servers

F5 BIG-IP administrator can configure virtual server on BIG-IP based on application requirements. Please refer to F5 BIG-IP configuration guide on virtual server configurations.

Deployment Workflow using Ansible (App tier to Web tier)

Cisco ACI has streamlines how ADC's can be connected to the fabric. One of the methods as we have discussed earlier is of connecting devices is called the Unmanaged mode.

Use Ansible as a tool to accomplish automating the ACI fabric to achieve L2-L3 stitching as well as automating the BIG-IP to achieve end-to-end application deployment

Before running the below Ansible playbook following needs to be setup:

- BIG-IP is licensed and is physically connected to the ACI fabric
- Access policies are configured on the APIC which enable the BIG-IP and Cisco ACI fabric to communicate
- Tenant, Private network, Bridge domain, Application profile and EPG's are created on APIC
- Values such as Self-IP and VLAN information needs to be known

In the example below we are configuring the BIG-IP hardware in HA mode which one interface attached to the fabric. But for configuration below the one physical interface will be treated as two logical interfaces on the BIG-IP.

HA on the BIG-IP is setup manually, once the BIG-IP is configured as an HA cluster then networking and application configuration can be done on the BIG-IP using Ansible:

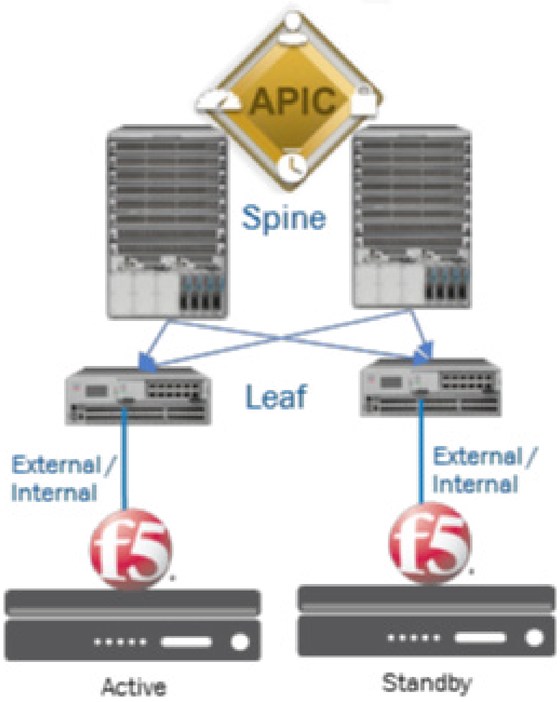


Figure 9: BIG-IP is configured as an HA cluster

Ansible Playbook

The playbook will include a variable file. The values will be substituted in the playbook which will run against the BIG-IP vCMP Host, BIG-IP vCMP Guest and Cisco APIC.

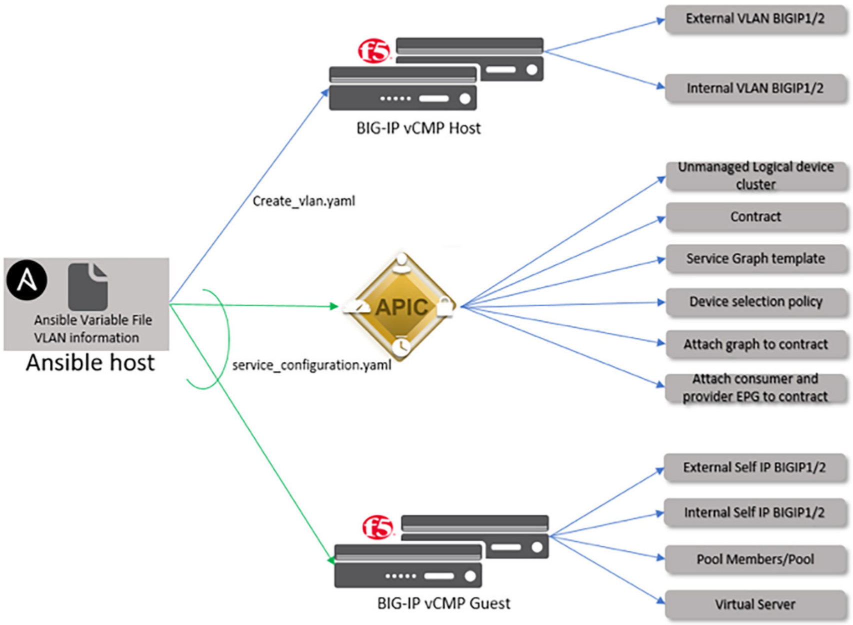


Figure 10: Ansible Playbook diagram

Multi-tenancy is achieved on the BIG-IP by using different subnets for different tenants

Field in white -> Pre-created.

Entry fields will be created by the ansible playbook.

Field Description	What does it mean?	Value use in this example
Managed	Managed: this L4-L7 device is managed and configured by Cisco APIC Unmanaged: L4-L7 device configuration is done by user	UNCHECK
consumerBD_name	Consumer bridge domain name on the APIC	externalBD
providerBD_name	Provider bridge domain name on the APIC	internalBD
appProfile_name	Application profile name on the APIC	ApplicationProfile
consumerEPG_name	Consumer End point group name on the APIC	externalEPG
providerEPG_name	Provider End point group name on the APIC	internalEPG
logicalDeviceCluster_name	Logical device cluster name on the APIC	UM_Tenant2A
device1_interface1_connectionPath	Physical connection path of BIG-IP1 to the APIC	pod-1/paths-102/pathep-[eth1/34]
device2_interface1_connectionPath	Physical connection path of BIG-IP1 to the APIC	pod-1/paths-102/pathep-[eth1/34]
SGtemplate_name	Service graph template name	Phys-UnManaged-2ARM-ServiceGraphTemplate

contract_name	Contract name	Unmanaged-2ARM-Contract
physical_domain_name	Physical domain to which the logical device cluster will be associated to	ServiceUnmanagedPhy
consumer_vlan_name	Consumer VLAN name on the BIG-IP	External_VLAN
consumer_vlan	Consumer VLAN - Will be statically bound to the logical device cluster - Will be created on the BIG-IP	2197
provider_vlan_name	Provider VLAN name on the BIG-IP	Internal_VLAN
provider_vlan	Provider VLAN - Will be statically bound to the logical device cluster - Will be created on the BIG-IP	2697
vip_port , vip_ip	Virtual IP address and port created on the BIG-IP	10.168.57.72, 80
pool_members (host/port)	Pool members and port on which the pool members are listening on created on the BIG-IP	192.168.57.140, 80 192.168.57.141, 80
bigip1_selfip_information (name/address/netmask)	Self IP address name, IP and netmask created on the BIG-IP1	External-SelfIP, 10.168.57.10, 255.255.255.0 Internal-SelfIP, 192.168.57.10, 255.255.255.0
bigip2_selfip_information (name/address/netmask)	Self IP address name, IP and netmask created on the BIG-IP2	External-SelfIP, 10.168.57.11, 255.255.255.0 Internal-SelfIP, 192.168.57.11, 255.255.255.0

```
Variable File:
tenant_name: “UM_Tenant2A”

consumerBD_name: “externalBD”
providerBD_name: “internalBD”

appProfile_name: “ApplicationProfile”
consumerEPG_name: “externalEPG”
providerEPG_name: “internalEPG”

logicalDeviceCluster_name: “UM_Tenant2A”

device1_interface1_connectionPath: “pod-1/paths-102/pathep-[eth1/34]”
device1_interface2_connectionPath: “”

device2_interface1_connectionPath: “pod-1/paths-103/pathep-[eth1/34]”
device2_interface2_connectionPath: “”

SGtemplate_name: “Phys-UnManaged-2ARM-ServiceGraphTemplate”
contract_name: “Unmanaged-2ARM-Contract”

physical_domain_name: “ServiceUnmanagedPhy”
consumer_vlan_name: “External_VLAN”
consumer_vlan: “2197”

provider_vlan_name: “Internal_VLAN”
provider_vlan: “2697”

vip_port: “80”
```

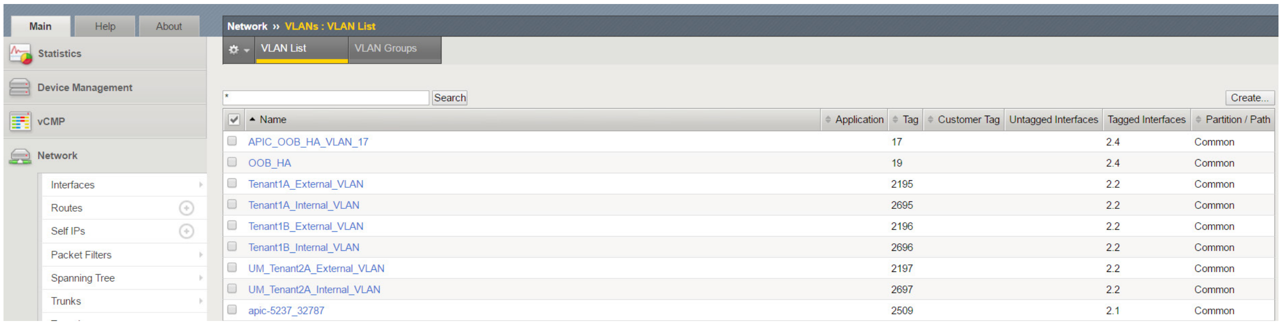
```
vip_ip: “10.168.57.72”

pool_members:
- port: “80”
  host: “192.168.57.140”
- port: “80”
  host: “192.168.57.141”

bigip1_selfip_information:
- name: ‘External-SelfIP’
  address: ‘10.168.57.10’
  netmask: ‘255.255.255.0’
  vlan: “External_VLAN”
- name: ‘Internal-SelfIP’
  address: ‘192.168.57.10’
  netmask: ‘255.255.255.0’
  vlan: ‘Internal_VLAN’

bigip2_selfip_information:
- name: ‘External-SelfIP’
  address: ‘10.168.57.11’
  netmask: ‘255.255.255.0’
  vlan: “External_VLAN”
- name: ‘Internal-SelfIP’
  address: ‘192.168.57.11’
  netmask: ‘255.255.255.0’
  vlan: ‘Internal_VLAN’
```

Playbook: create_vlan.yaml which will be run against the vCMP Host

```
- name: BIG-IP1 vCMP Host Setup
  hosts: unmanaged_vcmpHost_bigip1
  connection: local
  gather_facts: false

  vars_files:
    - variable_file.yaml

  tasks:

  - name: Add Provider VLAN - vCMP Host
    bigip_vlan:
      server: “{{inventory_hostname}}”
      user: “admin”
      password: “admin”
      name: “{{tenant_name}}_{{consumer_vlan_name}}”
      tag: “{{consumer_vlan}}”
      tagged_interfaces:
      - “2.2”
      validate_certs: “no”
    delegate_to: localhost

  - name: Add Consumer VLAN - vCMP Host
    bigip_vlan:
      server: “{{inventory_hostname}}”
      user: “admin”
      password: “admin”
      name: “{{tenant_name}}_{{provider_vlan_name}}”
      tag: “{{provider_vlan}}”
      tagged_interfaces:
      - “2.2”
      validate_certs: “no”
    delegate_to: localhost
```

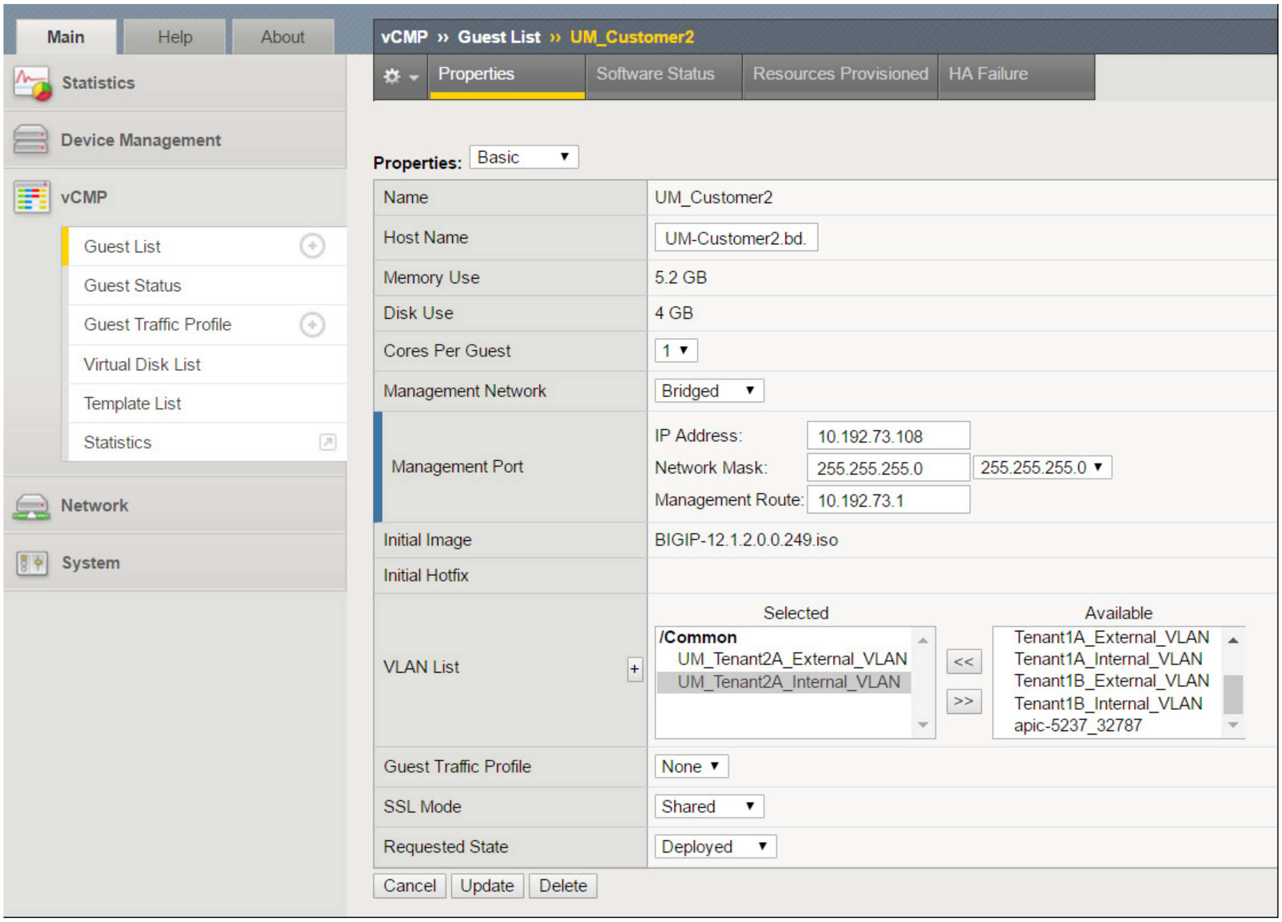
```
- name: BIG-IP2 vCMP Host Setup
  hosts: unmanaged_vcmpHost_bigip2
  connection: local
  gather_facts: false

  vars_files:
    - variable_file.yaml

  tasks:

  - name: Add Provider VLAN - vCMP Host
    bigip_vlan:
      server: “{{inventory_hostname}}”
      user: “admin”
      password: “admin”
      name: “{{tenant_name}}_{{consumer_vlan_name}}”
      tag: “{{consumer_vlan}}”
      tagged_interfaces:
      - “2.2”
      validate_certs: “no”
    delegate_to: localhost

  - name: Add Consumer VLAN - vCMP Host
    bigip_vlan:
      server: “{{inventory_hostname}}”
      user: “admin”
      password: “admin”
      name: “{{tenant_name}}_{{provider_vlan_name}}”
      tag: “{{provider_vlan}}”
      tagged_interfaces:
      - “2.2”
      validate_certs: “no”
    delegate_to: localhost
```

Following VLANS gets configured on the vCMP Hosts:



☑	Name	Application	Tag	Customer Tag	Untagged Interfaces	Tagged Interfaces	Partition / Path
□	APIC_OOB_HA_VLAN_17		17			2.4	Common
□	OOB_HA		19			2.4	Common
□	Tenant1A_External_VLAN		2195			2.2	Common
□	Tenant1A_Internal_VLAN		2695			2.2	Common
□	Tenant1B_External_VLAN		2196			2.2	Common
□	Tenant1B_Internal_VLAN		2696			2.2	Common
□	UM_Tenant2A_External_VLAN		2197			2.2	Common
□	UM_Tenant2A_Internal_VLAN		2697			2.2	Common
□	apic-5237_32787		2509			2.1	Common

Add the VLAN’s from the vCMP Host to the vCMP guest (this is a manual step).



Properties: Basic

Name	UM_Customer2
Host Name	UM-Customer2.bd.
Memory Use	5.2 GB
Disk Use	4 GB
Cores Per Guest	1
Management Network	Bridged
Management Port	IP Address: 10.192.73.108 Network Mask: 255.255.255.0 255.255.255.0 Management Route: 10.192.73.1
Initial Image	BIGIP-12.1.2.0.0.249.iso
Initial Hotfix	
VLAN List	Selected: /Common UM_Tenant2A_External_VLAN UM_Tenant2A_Internal_VLAN Available: Tenant1A_External_VLAN Tenant1A_Internal_VLAN Tenant1B_External_VLAN Tenant1B_Internal_VLAN apic-5237_32787
Guest Traffic Profile	None
SSL Mode	Shared
Requested State	Deployed

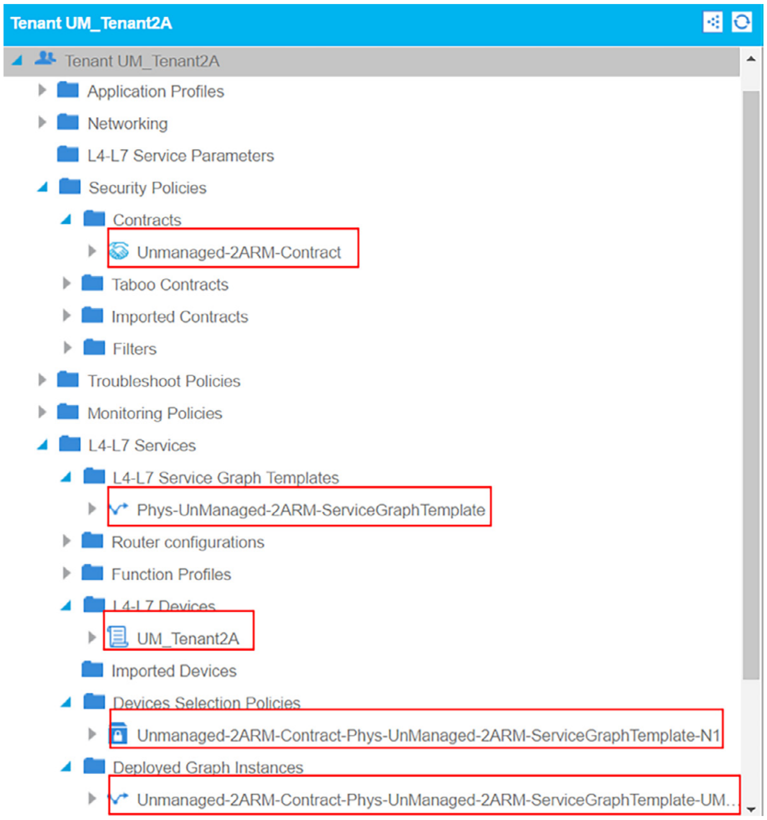
`Playbook: service_configuration.yaml` which will be run against the Cisco APIC and the BIG-IP vCMP guests. (Check the jinga2 (*.j2) files used in this example in the appendix)

```
- name: ACI Setup
  hosts: aci
  connection: local
  gather_facts: false

  vars_files:
    - variable_file.yaml

  tasks:

  - name: Create XML POSTS from templates
    template: src={{ item.src }} dest={{ item.dest }}
    with_items:
      - { src: ‘create_unmanged_ldev.j2’,  dest: ‘create_unmanged_ldev.xml’ }
      - { src: ‘contract.j2’, dest: ‘contract.xml’ }
      - { src: service_graph_template.j2’, dest: ‘service_graph_template.xml’}
      - { src: ‘deviceSelectionPolicy.j2’, dest: ‘deviceSelectionPolicy.xml’}
      - { src: ‘apply_graph.j2’, dest: ‘apply_graph.xml’}
      - { src: ‘attach_cons_prov_contract.j2’, dest: ‘attach_cons_prov_contract.xml’}

  - name: Execute POSTS
    aci_rest:
      action: "post"
      uri: "/api/node/mo/uni/tn-{{tenant_name}}.xml"
      config_file: "{{ item }}"
      host: "{{inventory_hostname}}"
      username: admin
      password: cisco123
    with_items:
     - "create_unmanged_ldev.xml"
     - "contract.xml"
     - "service_graph_template.xml"
     - "deviceSelectionPolicy.xml"
     - "apply_graph.xml"
     - "attach_cons_prov_contract.xml"

- name: BIG-IP1 vCMP guest Setup
  hosts: unmanaged_Cust2_bigip1
  connection: local
  gather_facts: false

  vars_files:
    - variable_file.yaml

  tasks:

  - name: Configure SELF-IP
    bigip_selfip:
     server: "{{inventory_hostname}}"
     user: "admin"
     password: "admin"
     validate_certs: False
     name: "{{tenant_name}}_{{item.name}}"
     address: "{{item.address}}"
     netmask: "{{item.netmask}}"
     vlan: "{{tenant_name}}_{{item.vlan}}"
    with_items: "{{ bigip1_selfip_information }}"
    delegate_to: localhost

  - name: Create pool
    bigip_pool:
      server: "{{inventory_hostname}}"
      user: "admin"
      password: "admin"
      state: "present"
      name: "{{tenant_name}}_http-pool"
      lb_method: "round-robin"
      monitors: "/Common/http"
      monitor_type: "and_list"
      quorum: 1
    delegate_to: localhost

  - name: Add Pool members
    bigip_pool:
      server: "{{inventory_hostname}}"
      user: "admin"
      password: "admin"
      name: "{{tenant_name}}_http-pool"
      host: "{{item.host}}"
      port: "{{item.port}}"
      validate_certs: False
    with_items: "{{pool_members}}"
    delegate_to: localhost

  - name: Add Virtual Server
```

```
    bigip_virtual_server:
      server: “{{inventory_hostname}}”
      user: “admin”
      password: “admin”
      name: “{{tenant_name}}_http_vs”
      destination: “{{vip_ip}}”
      port: “{{vip_port}}”
      enabled_vlans:
       - “{{tenant_name}}_{{consumer_vlan_name}}”
       - “{{tenant_name}}_{{provider_vlan_name}}”
      profiles_both: “http”
      pool: “{{tenant_name}}_http-pool”
      snat: “automap”
      validate_certs: False
    delegate_to: localhost

- name: BIG-IP2 vCMP guest Setup
  hosts: unmanaged_Cust2_bigip2
  connection: local
  gather_facts: false

  vars_files:
    - variable_file.yaml

  tasks:

  - name: Configure SELF-IP
    bigip_selfip:
     server: “{{inventory_hostname}}”
     user: “admin”
     password: “admin”
     validate_certs: False
     name: “{{tenant_name}}_{{item.name}}”
     address: “{{item.address}}”
     netmask: “{{item.netmask}}”
     vlan: “{{tenant_name}}_{{item.vlan}}”
    with_items: “{{ bigip2_selfip_information }}”
    delegate_to: localhost
```

For multi-tenancy, you can change the variable file to represent a different tenant and different VLAN/Self IP values

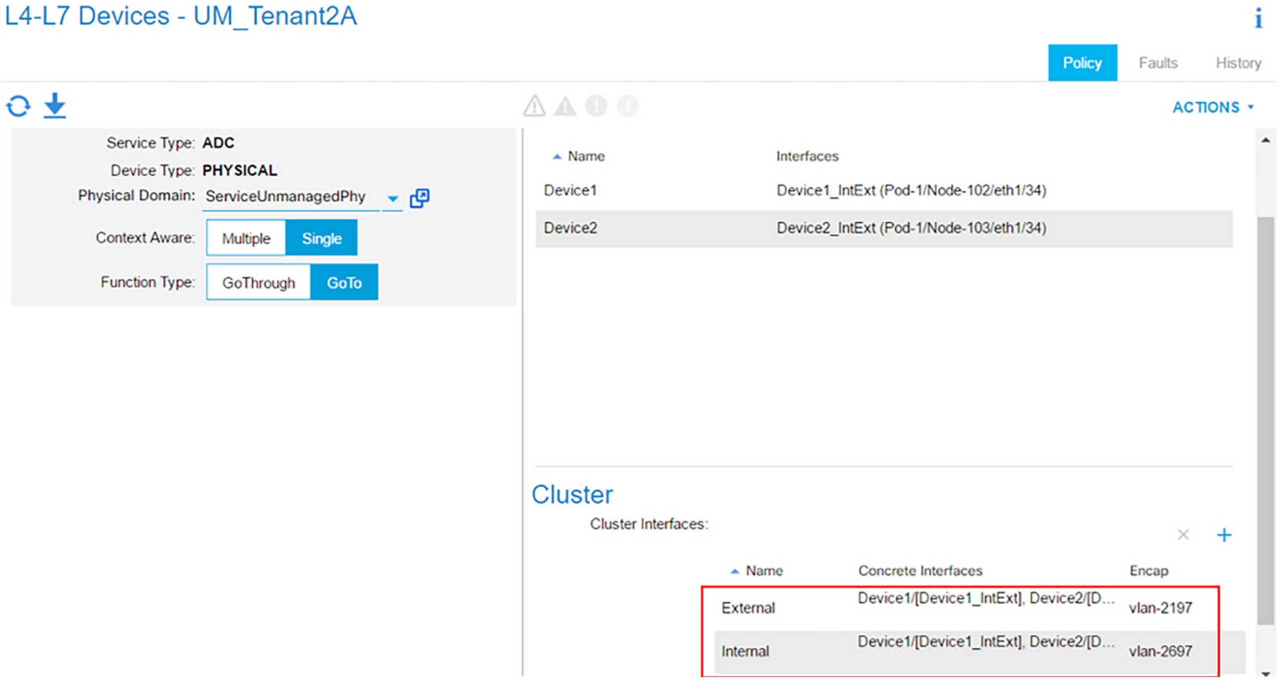
APIC each tenant will be configured as follows:



Tenant UM_Tenant2A

- Tenant UM_Tenant2A
  - Application Profiles
  - Networking
  - L4-L7 Service Parameters
  - Security Policies
    - Contracts
      - Unmanaged-2ARM-Contract
    - Taboo Contracts
    - Imported Contracts
    - Filters
  - Troubleshoot Policies
  - Monitoring Policies
  - L4-L7 Services
    - L4-L7 Service Graph Templates
      - Phys-UnManaged-2ARM-ServiceGraphTemplate
    - Router configurations
    - Function Profiles
    - L4-L7 Devices
      - UM_Tenant2A
    - Imported Devices
    - Devices Selection Policies
      - Unmanaged-2ARM-Contract-Phys-UnManaged-2ARM-ServiceGraphTemplate-N1
    - Deployed Graph Instances
      - Unmanaged-2ARM-Contract-Phys-UnManaged-2ARM-ServiceGraphTemplate-UM...

Logical device cluster (has VLANS 2197 and 2697):



L4-L7 Devices - UM_Tenant2A

Service Type: ADC
Device Type: PHYSICAL
Physical Domain: ServiceUnmanagedPhy
Context Aware: Multiple [✓ Single]
Function Type: GoThrough [✓ GoTo]

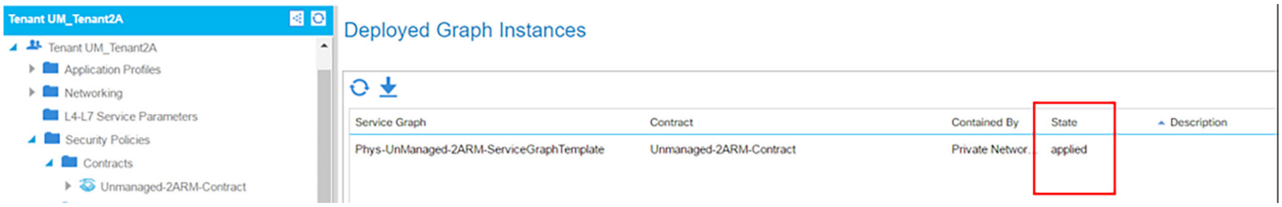
Name	Interfaces
Device1	Device1_IntExt (Pod-1/Node-102/eth1/34)
Device2	Device2_IntExt (Pod-1/Node-103/eth1/34)

Cluster

Cluster Interfaces:

Name	Concrete Interfaces	Encap
External	Device1/[Device1_IntExt], Device2/[D...	vlan-2197
Internal	Device1/[Device1_IntExt], Device2/[D...	vlan-2697

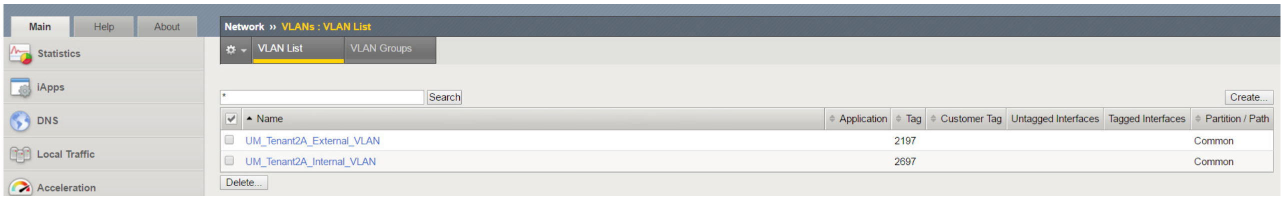
Make sure the device is in deployed correctly, Tenant-> L4-L7 services -> Deployed Graph:



Service Graph	Contract	Contained By	State	Description
Phys-UnManaged-2ARM-ServiceGraphTemplate	Unmanaged-2ARM-Contract	Private Networ...	applied	

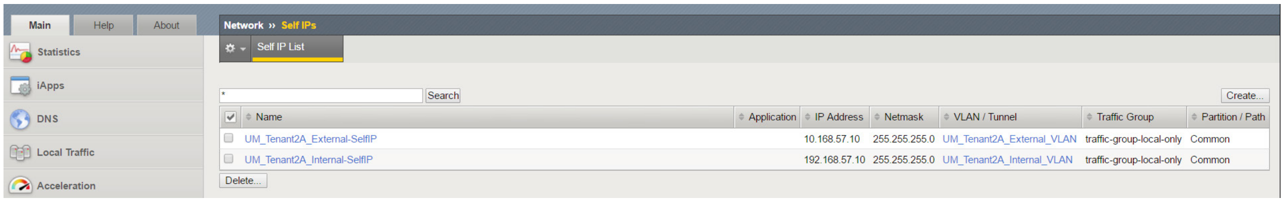
BIG-IP each tenant will be configured with values as follows:

VLANs: (2197 and 2697):



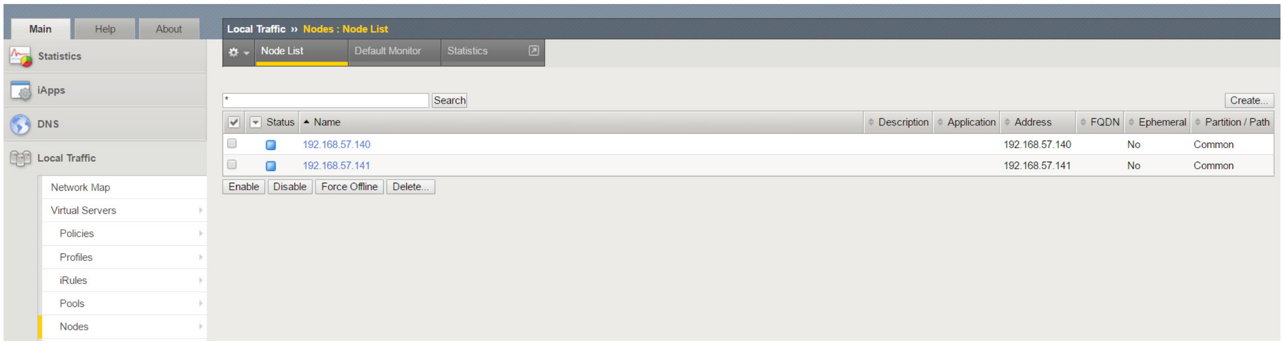
Name	Application	Tag	Customer Tag	Untagged Interfaces	Tagged Interfaces	Partition / Path
UM_Tenant2A_External_VLAN		2197				Common
UM_Tenant2A_Internal_VLAN		2697				Common

SELF-IPs:



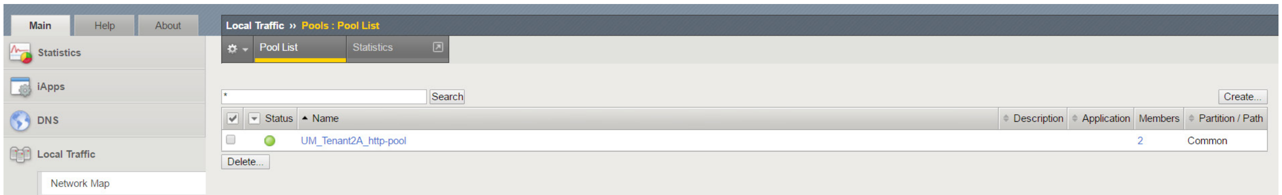
Name	Application	IP Address	Netmask	VLAN / Tunnel	Traffic Group	Partition / Path
UM_Tenant2A_External-SelfIP		10.168.57.10	255.255.255.0	UM_Tenant2A_External_VLAN	traffic-group-local-only	Common
UM_Tenant2A_Internal-SelfIP		192.168.57.10	255.255.255.0	UM_Tenant2A_Internal_VLAN	traffic-group-local-only	Common

Pool Members:



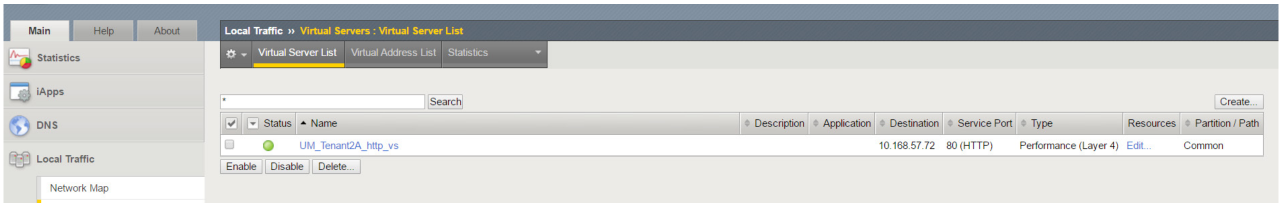
Status	Name	Description	Application	Address	FQDN	Ephemeral	Partition / Path
■	192.168.57.140			192.168.57.140		No	Common
■	192.168.57.141			192.168.57.141		No	Common

Pools:



Status	Name	Description	Application	Members	Partition / Path
●	UM_Tenant2A_http-pool			2	Common

Virtual Servers:



☑	Status	Name	Description	Application	Destination	Service Port	Type	Resources	Partition / Path
□	●	UM_Tenant2A_http_vs			10.168.57.72	80 (HTTP)	Performance (Layer 4)	Edit...	Common

Deployment Workflow (Web Tier to DB Tier)

The workflow to setup a BIG-IP in unmanaged mode between the Web and the DB tier would be the similar as above, however we are using a BIG-IP virtual edition at this layer and following will some of the differences.

Deployment Workflow Progress:

1. APIC: Create Unmanaged Virtual L4-L7 Device
2. APIC: Create Service Graph Template
3. APIC: Deploy Service Graph Template
4. BIG-IP: Configure Network Parameters
5. BIG-IP: Configure Virtual Servers

APIC: Create Unmanaged Virtual L4-L7 Device

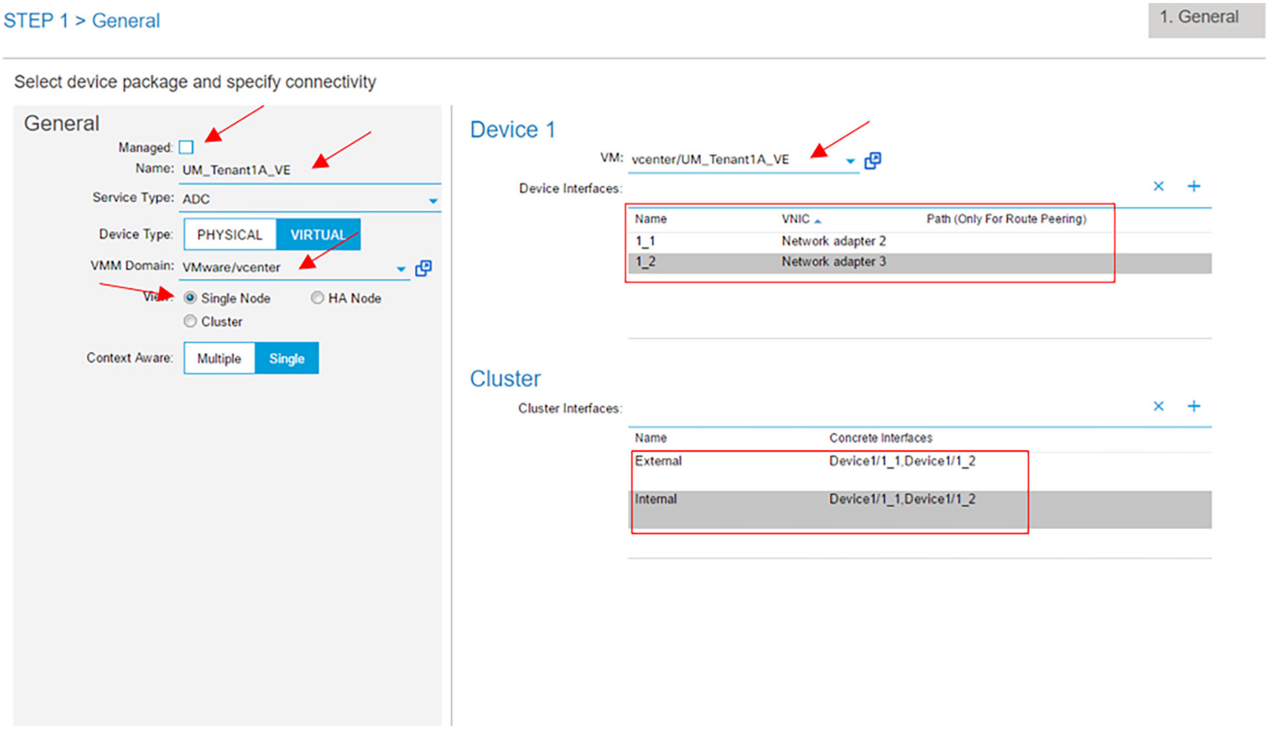
Create Unmanaged L4-L7 device under user-defined (UM_Tenant1A) tenant, go to:

Tenants UM_Tenant1A -> L4-L7 Services -> L4-L7 Devices, right click and go to 'Create L4-L7' devices

For Unmanaged BIG-IP VE, configure as follow:

- Uncheck the box "Managed"
- Name: Name of the Unmanaged BIG-IP VE cluster
- Service Type: ADC
- Device Type: Virtual
- VMM Domain: Select the VMM domain already integrated in APIC where BIG-IP VE is hosted
- View: Single or HA, in this example, it is a standalone BIG-IP VE
- VM: Select the BIG-IP VE virtual machine from drop-down menu
- Device Interface: BIG-IP interface 1.1 = Network Adaptor 2; BIG-IP interface 1.2 = Network Adaptor 3 (NOTE: If route peering is required, then Path must be specified. Path is where the hypervisor physically connected to the ACI fabric)
- Cluster Interfaces: Logical Interfaces (External and Internal) map to the BIG-IP VE interfaces (NOTE: in the case of Unmanaged VE, no static VLAN binding is specified in L4-L7 Device)

Click **FINISH** when completed.



STEP 1 > General

1. General

Select device package and specify connectivity

General

Managed: □
Name: UM_Tenant1A_VE
Service Type: ADC
Device Type: PHYSICAL [✓ VIRTUAL]
VMM Domain: VMware/vcenter
View: ● Single Node ○ HA Node ○ Cluster
Context Aware: Multiple [✓ Single]

Device 1

VM: vcenter/UM_Tenant1A_VE

Device Interfaces:

Name	VNIC	Path (Only For Route Peering)
1_1	Network adapter 2	
1_2	Network adapter 3	

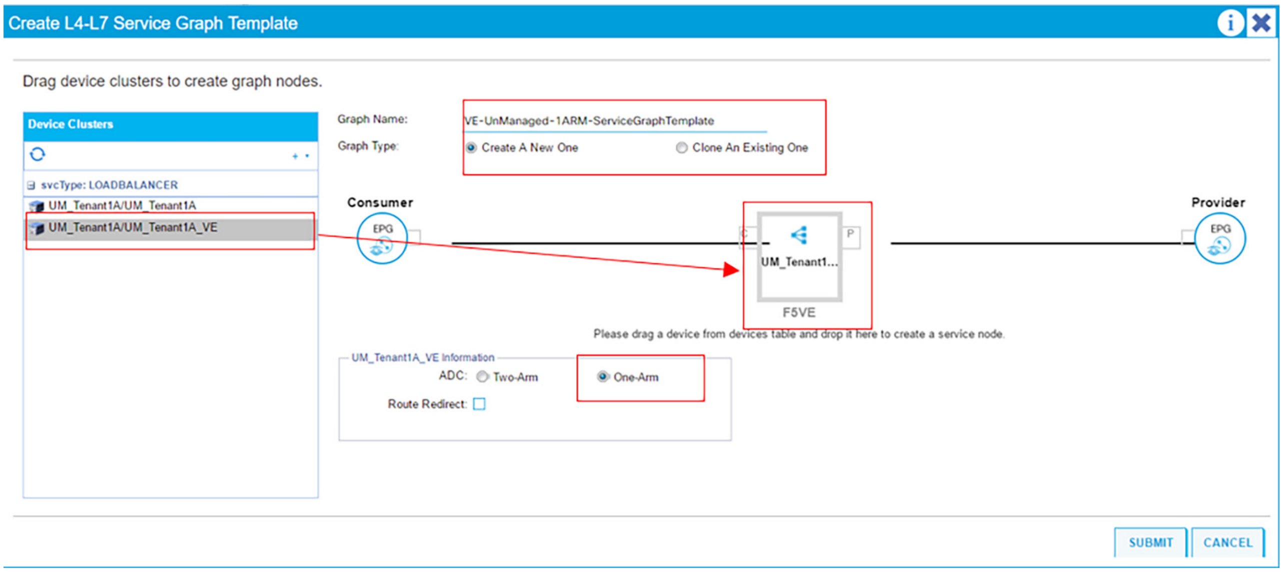
Cluster

Cluster Interfaces:

Name	Concrete Interfaces
External	Device1/1_1,Device1/1_2
Internal	Device1/1_1,Device1/1_2

APIC: Create Service Graph Template

Create a generic 1-arm service graph that uses the unmanaged VE cluster



Create L4-L7 Service Graph Template

Drag device clusters to create graph nodes.

Device Clusters

svcType: LOADBALANCER
UM_Tenant1A/UM_Tenant1A
UM_Tenant1A/UM_Tenant1A_VE

Graph Name: VE-UnManaged-1ARM-ServiceGraphTemplate
Graph Type: ● Create A New One ○ Clone An Existing One

Please drag a device from devices table and drop it here to create a service node.

UM_Tenant1A_VE Information
ADC: ○ Two-Arm ● One-Arm
Route Redirect: □

APIC: Deploy Service Graph Template

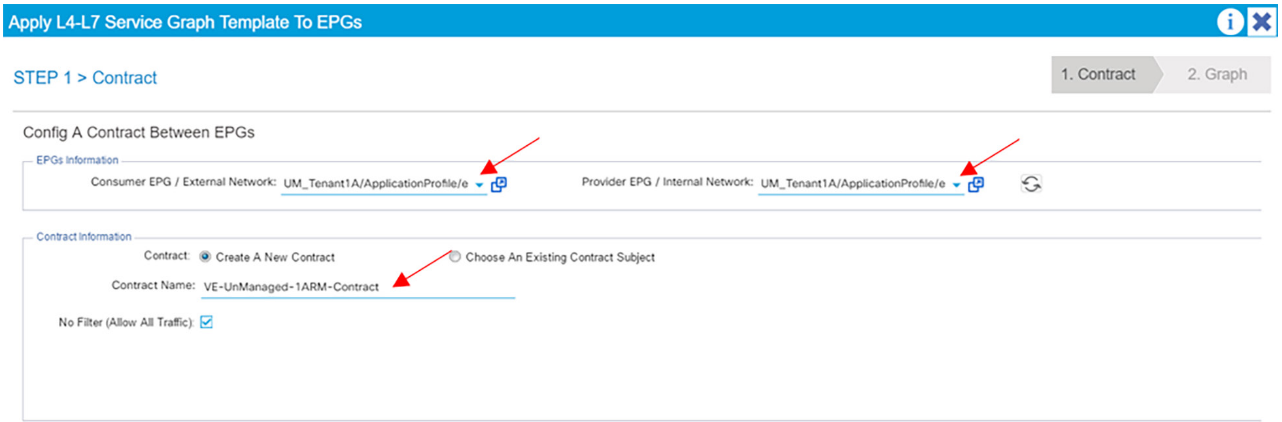
Select the consumer and provider EPG, in this example:

Consumer: InternalEPG (Web Tier)

Provider: databaseEPG (DB Tier)

Provide a new contract name

Click **NEXT**



Apply L4-L7 Service Graph Template To EPGs

STEP 1 > Contract

Config A Contract Between EPGs

EPGs Information

Consumer EPG / External Network: UM_Tenant1A/ApplicationProfile/e

Provider EPG / Internal Network: UM_Tenant1A/ApplicationProfile/e

Contract Information

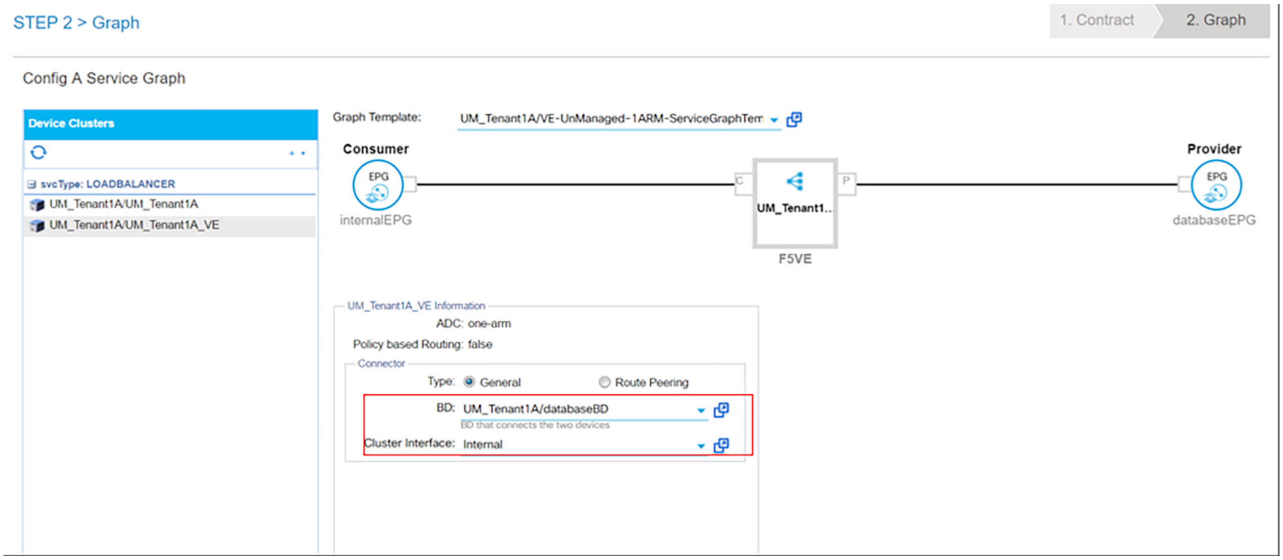
Contract: ● Create A New Contract ○ Choose An Existing Contract Subject

Contract Name: VE-UnManaged-1ARM-Contract

No Filter (Allow All Traffic): ☑

Select the Cluster Interface that is tied to the BD. In this example, the BD is the provider EPG - Database BD and it is associated with the Internal Cluster Interface.

Click **FINISH**:



STEP 2 > Graph

Config A Service Graph

Device Clusters

svcType: LOADBALANCER
UM_Tenant1A/UM_Tenant1A
UM_Tenant1A/UM_Tenant1A_VE

Graph Template: UM_Tenant1A/VE-UnManaged-1ARM-ServiceGraphTem

UM_Tenant1A_VE Information

ADC: one-arm

Policy based Routing: false

Connector

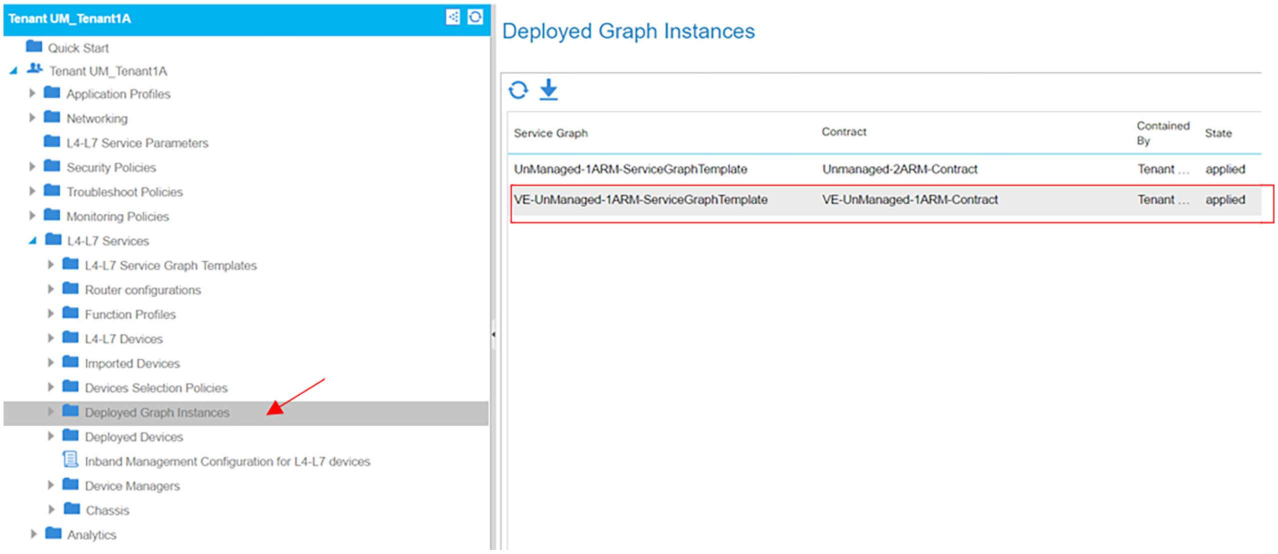
Type: ● General ○ Route Peering

BD: UM_Tenant1A/databaseBD

BD that connects the two devices

Cluster Interface: Internal

Verify the graph is deployed successfully:



Tenant UM_Tenant1A

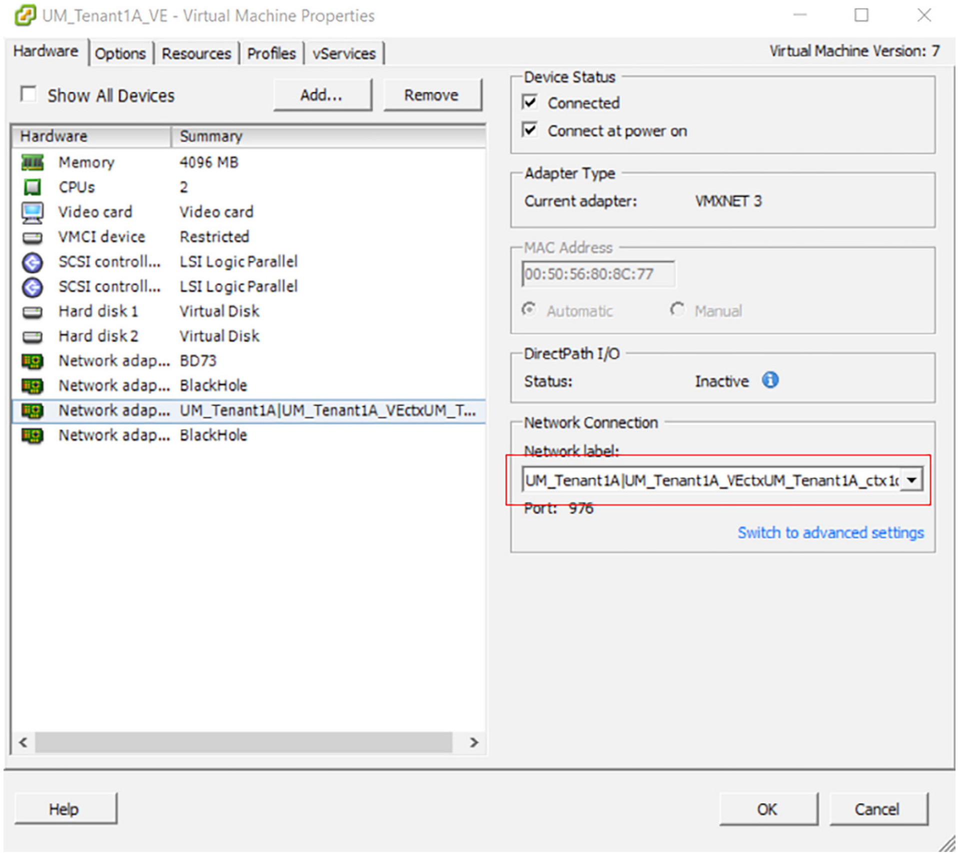
- Quick Start
- Tenant UM_Tenant1A
  - Application Profiles
  - Networking
  - L4-L7 Service Parameters
  - Security Policies
  - Troubleshoot Policies
  - Monitoring Policies
  - L4-L7 Services
    - L4-L7 Service Graph Templates
    - Router configurations
    - Function Profiles
    - L4-L7 Devices
    - Imported Devices
    - Devices Selection Policies
    - Deployed Graph Instances
    - Deployed Devices
    - Inband Management Configuration for L4-L7 devices
    - Device Managers
    - Chassis
  - Analytics

Deployed Graph Instances

Service Graph	Contract	Contained By	State
UnManaged-1ARM-ServiceGraphTemplate	Unmanaged-2ARM-Contract	Tenant ...	applied
VE-UnManaged-1ARM-ServiceGraphTemplate	VE-UnManaged-1ARM-Contract	Tenant ...	applied

BIG-IP: Configure Network Parameters

In this example, the BIG-IP VE is deployed in VMware environment, go to the vCenter that manage the BIG-IP VE, notice APIC has assigned a port-group to BIG-IP interface 1.2 (Internal Interface), which is VNIC3:



UM_Tenant1A_VE - Virtual Machine Properties

Hardware | Options | Resources | Profiles | vServices

Virtual Machine Version: 7

☐ Show All Devices Add... Remove

Hardware	Summary
Memory	4096 MB
CPUs	2
Video card	Video card
VMCI device	Restricted
SCSI controll...	LSI Logic Parallel
SCSI controll...	LSI Logic Parallel
Hard disk 1	Virtual Disk
Hard disk 2	Virtual Disk
Network adap...	BD73
Network adap...	BlackHole
Network adap...	UM_Tenant1A|UM_Tenant1A_VEctxUM_T...
Network adap...	BlackHole

Device Status
☑ Connected
☑ Connect at power on

Adapter Type
Current adapter: VMXNET 3

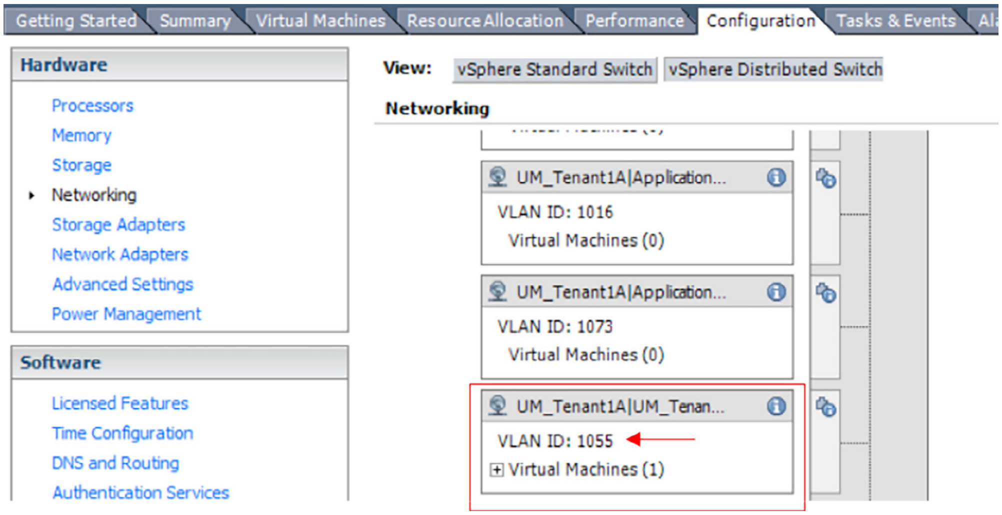
MAC Address
00:50:56:80:8C:77
● Automatic ○ Manual

DirectPath I/O
Status: Inactive

Network Connection
Network label:
UM_Tenant1A|UM_Tenant1A_VEctxUM_Tenant1A_ctx1c
Port: 976
Switch to advanced settings

Help OK Cancel

Go to the APIC generated distributed virtual switch. Notice VLAN 1055 is assigned to this port-group. VLAN 1055 is part of the VLAN pool values assigned to the APIC VMM domain.



UM_Tenant1A|Application...
VLAN ID: 1016
Virtual Machines (0)

UM_Tenant1A|Application...
VLAN ID: 1073
Virtual Machines (0)

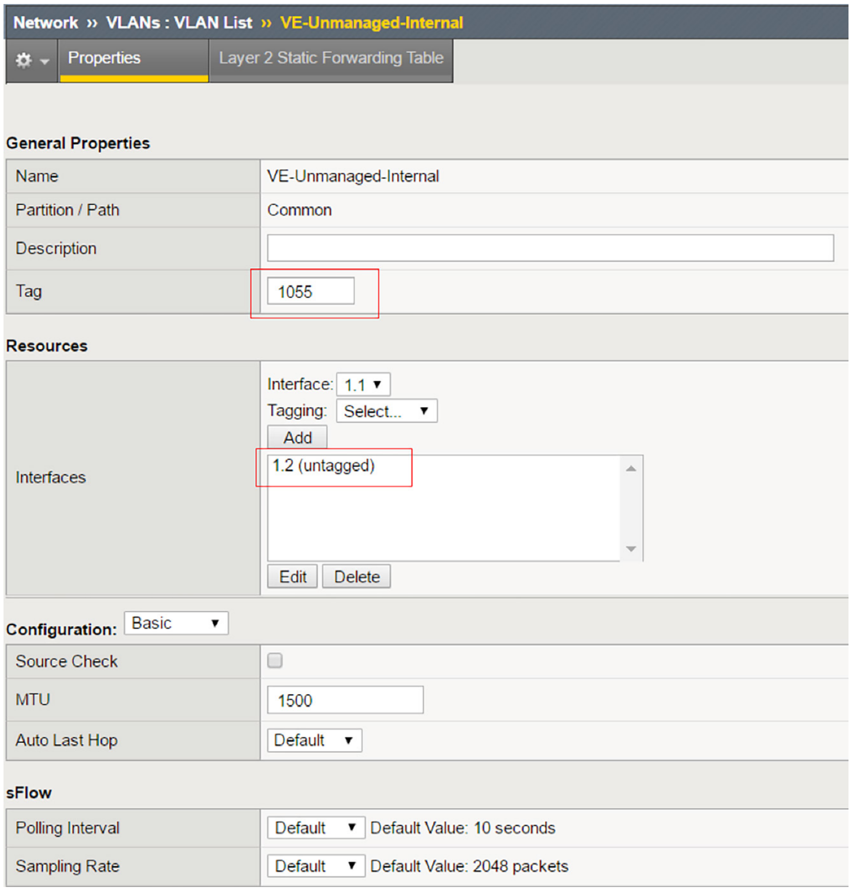
UM_Tenant1A|UM_Tenan...
VLAN ID: 1055
Virtual Machines (1)

Network administrator can now pass this VLAN tag value and self IP value to the F5 administrator for BIG-IP network configuration.

Configure unmanaged BIG-IP VE VLAN. Based on vCenter port-group VLAN, in this example:

VLAN tag: 1055

Interface: 1.2 Untagged – 1.2 is the internal interface, tagging at the VMware distributed virtual switch level:


Network » VLANs : VLAN List » VE-Unmanaged-Internal
Properties
Layer 2 Static Forwarding Table
Add
Edit
Delete

General Properties

Name	VE-Unmanaged-Internal
Partition / Path	Common
Description	
Tag	1055

Resources

Interfaces	Interface: 1.1 Tagging: Select... 1.2 (untagged)

Configuration: Basic

Source Check	□
MTU	1500
Auto Last Hop	Default

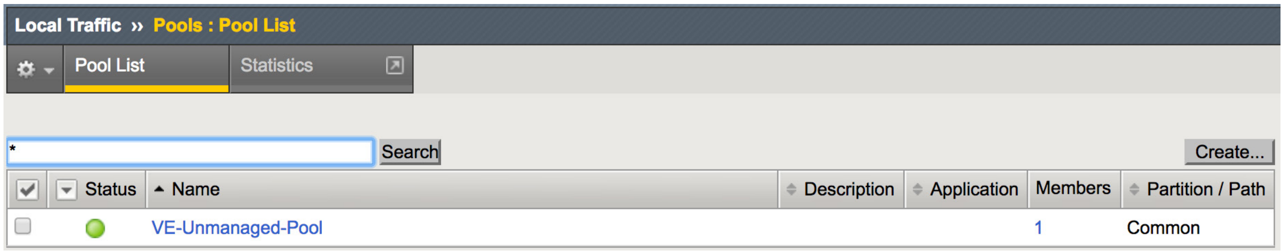
sFlow

Polling Interval	Default Default Value: 10 seconds
Sampling Rate	Default Default Value: 2048 packets

Configure BIG-IP self IP and default gateway (if needed) based on the APIC Network/BD configuration.

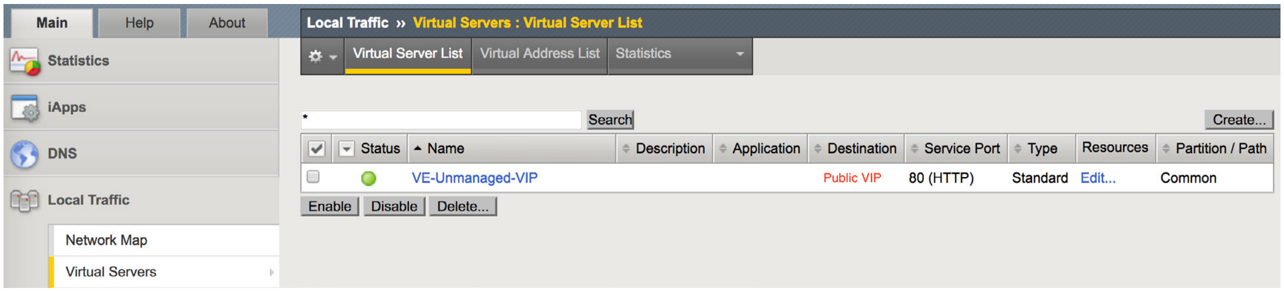
BIG-IP: Configure Virtual Servers

Create Pool and add pool members, verify pool member(s) is available:



☑	Status	Name	Description	Application	Members	Partition / Path
□	●	VE-Unmanaged-Pool			1	Common

Create virtual server to load balance the backend DB servers, utilize the pool created above.



☑	Status	Name	Description	Application	Destination	Service Port	Type	Resources	Partition / Path
□	●	VE-Unmanaged-VIP			Public VIP	80 (HTTP)	Standard	Edit...	Common

Ansible playbook above can be used to configuration Self-IP/VLAN/Pools/Virtual Servers etc on the BIG-IP Virtual Edition as well.

Cisco 9000 NX-OS (Standalone)

The Cisco Nexus 9000 Series offers high-performance data center switches that include both fixed-configuration and modular models. It has a variety of physical interfaces that include 1-, 10-, 40-, and, 100-Gbps connectivity to all types of devices in a typical data center or cloud environment. The Cisco Nexus 9000 Series has two modes of operation: one with Cisco® NX-OS Software and the other with Cisco Application Centric Infrastructure (Cisco ACI). This section focuses in detail on Ansible integration with NX-OS and F5 BIG-IP.

F5 BIG-IP Application Delivery Controllers (ADC) appliances and Virtual Edition products can use Ansible for DevOps. With introduction of Open NXOS, a common Ansible infrastructure can be used to provision and manage Cisco Nexus 9000 fabric and F5 BIG-IP devices in the Data Center.

Problem

- L2-L7 layer configuration is independent of each other causing problems in maintaining a consistency, reliable and automatable deployment
- Managing scalability of applications

Solution

- Glue which stitches L2 to L7 services using BIG-IP & Cisco Data Center fabric
- Use Enhance features of Ansible for F5 BIG-IP and Cisco NX-OS 9000 series Integration.
    - N9K switch configuration provisioning: Cookie cutter configuration for Scaled Deployment
    - Common VLAN provisioning for Layer 3 separation

Advantage

- Single playbook for provisioning as well as managing the entire application stack
- Accelerated application deployments with reliability, security, and consistent scalable network and Layer 4 through 7 services
- Improved deployment speed for services enabling consistent automation and orchestration of critical services to support business

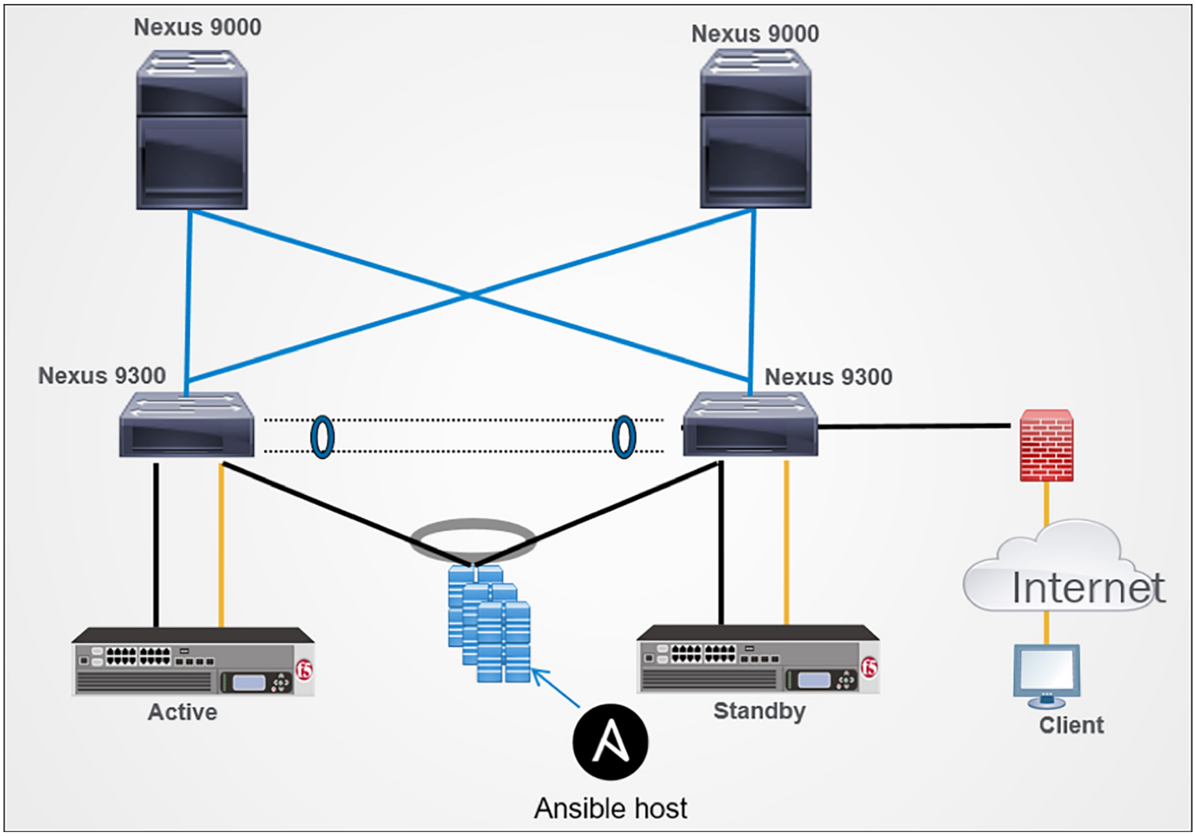


Figure 11: Cisco Nexus 9300 Ansible architecture

Common Playbook

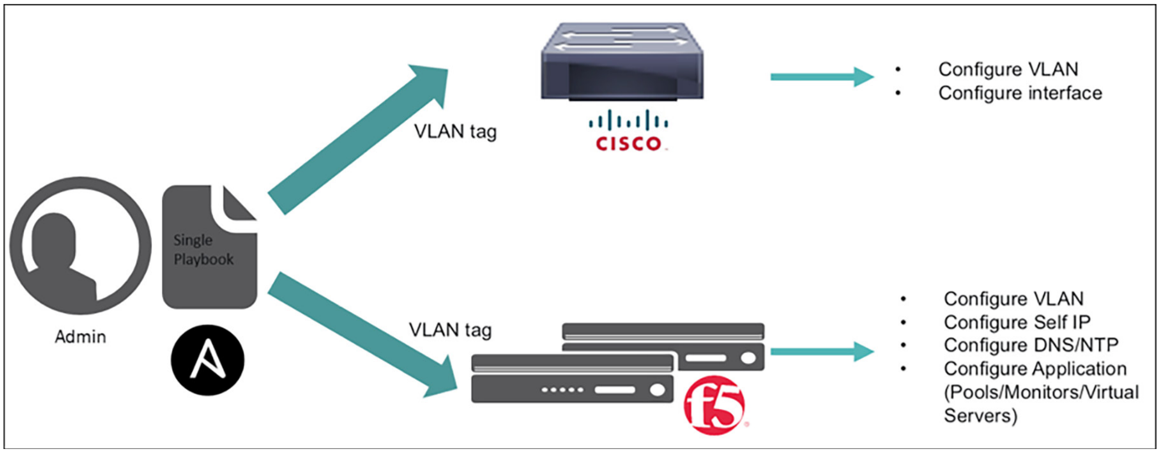


Figure 12: Common Playbook for Ansible

Application Deployment

L2-L7 configuration to load balance an application across web servers

```
Command to run playbook. We are passing the VLAN information as part of the command line argument
but this information can be passed in a variable file as well
```

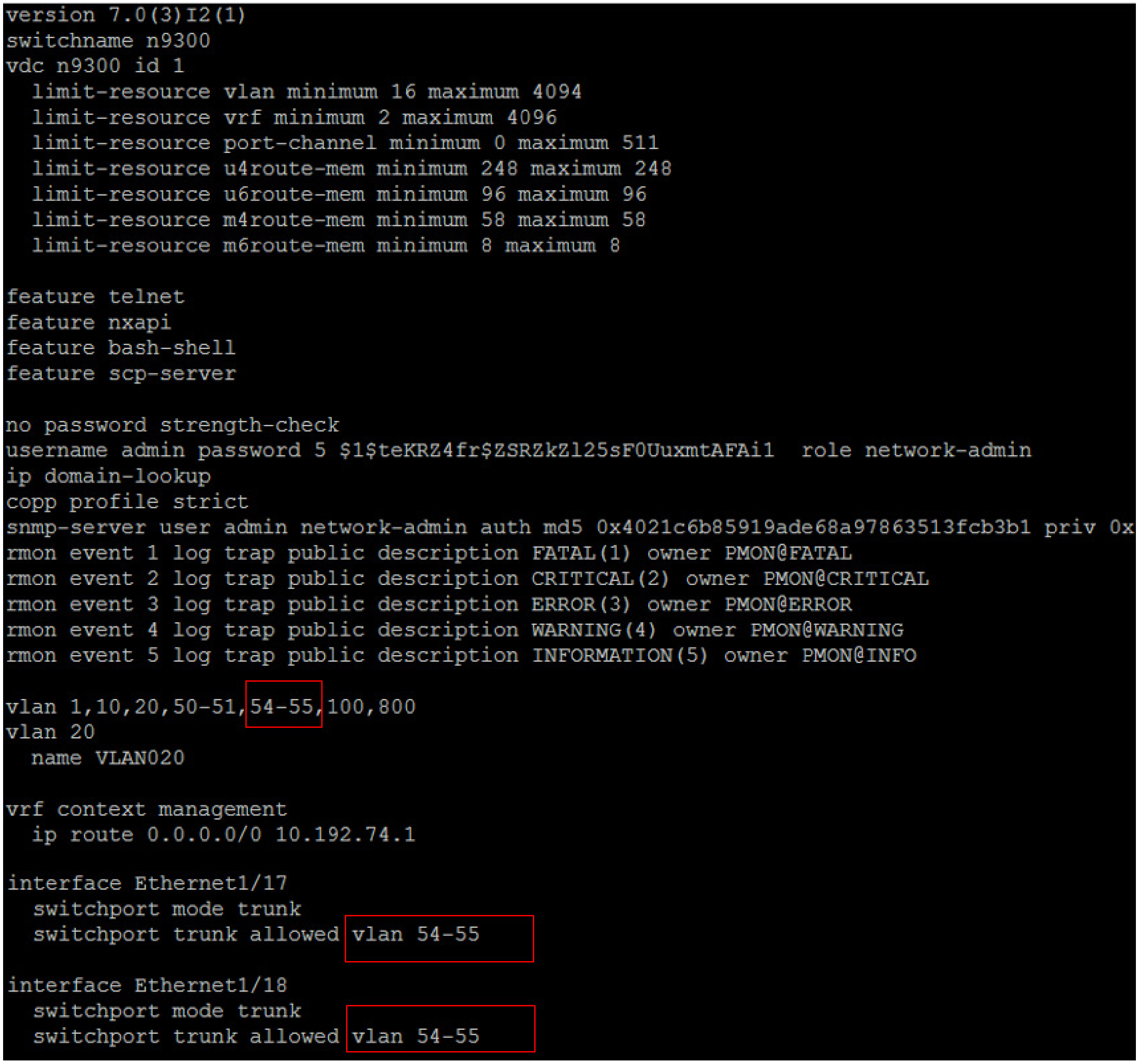
`ansible-playbook playbooks/cisco.yaml --extra-vars "vlan_external=54 vlan_internal=55"`

```
Playbook:
---
- name: Configure Cisco n9K
  hosts: cisco
  connection: local
  gather_facts: true

  tasks:

  - name: Setup VLAN
    nxos_vlan:
     username: admin
     password: cisco123
     transport: nxapi
     host: "{{ inventory_hostname }}"
     vlan_id: "{{ item }}"
     state: present
    with_items:
     - "{{ vlan_external }}"
     - "{{ vlan_internal }}"

  - name: Configure vlan on Eth1/17
    nxos_switchport:
     interface: eth1/17
     mode: trunk
     username: admin
     password: cisco123
     transport: nxapi
     host: "{{ inventory_hostname }}"
     trunk_vlans: “{{ item }}”
    with_items:
     - “{{ vlan_external }}”
     - “{{ vlan_internal }}”

  - name: Configure vlan on Eth1/18
    nxos_switchport:
     interface: eth1/18
     mode: trunk
     username: admin
     password: cisco123
     transport: nxapi
     host: “{{ inventory_hostname }}”
     trunk_vlans: “{{ item }}”
    with_items:
     - “{{ vlan_external }}”
     - “{{ vlan_internal }}”

- name: Configure BIG-IP1
  hosts: bigip1_cisco
  connection: local
  tasks:

  - name: Configure VLANs on the BIG-IP1
    bigip_vlan:
        server: “{{ inventory_hostname }}”
        user: “{{ username }}”
        password: “{{ password }}”
        validate_certs: False
        name: “{{ item.name }}”
        tag: “{{ item.tag }}”
        tagged_interface: “{{ item.interface }}”
    with_items:
        - name: ‘External’
          tag: “{{ vlan_external }}”
          interface: ‘1.1’
        - name: ‘Internal’
          tag: “{{ vlan_internal }}”
          interface: ‘1.1’
    delegate_to: localhost

  - name: Configure SELF-IPs on the BIG-IP1
    bigip_selfip:
        server: “{{ inventory_hostname }}”
        user: “{{ username }}”
        password: “{{ password }}”
        validate_certs: False
        name: “{{ item.name }}”
        address: “{{ item.address }}”
        netmask: “{{ item.netmask }}”
        vlan: “{{ item.vlan }}”
        allow_service: “{{item.allow_service}}”
    with_items:
        - name: ‘External-SelfIP’
          address: ‘10.10.10.10’
          netmask: ‘255.255.255.0’
          vlan: ‘External’
          allow_service: ‘default’
        - name: ‘Internal-SelfIP’
          address: ‘192.10.10.10’
          netmask: ‘255.255.255.0’
          vlan: ‘Internal’
          allow_service: ‘default’
    delegate_to: localhost
```

```
- name: Configure BIG-IP2
  hosts: bigip2_cisco
  connection: local
  tasks:

  - name: Configure VLANs on the BIG-IP2
    bigip_vlan:
        server: “{{ inventory_hostname }}”
        user: “{{ username }}”
        password: “{{ password }}”
        validate_certs: False
        name: “{{ item.name }}”
        tag: “{{ item.tag }}”
        tagged_interface: “{{ item.interface }}”
    with_items:
        - name: ‘External’
          tag: “{{ vlan_external }}”
          interface: ‘1.1’
        - name: ‘Internal’
          tag: “{{ vlan_internal }}”
          interface: ‘1.1’
    delegate_to: localhost

  - name: Configure SELF-IPs on the BIG-IP2
    bigip_selfip:
        server: “{{ inventory_hostname }}”
        user: “{{ username }}”
        password: “{{ password }}”
        validate_certs: False
        name: “{{ item.name }}”
        address: “{{ item.address }}”
        netmask: “{{ item.netmask }}”
        vlan: “{{ item.vlan }}”
        allow_service: “{{item.allow_service}}”
    with_items:
        - name: ‘External-SelfIP’
          address: ‘10.10.10.11’
          netmask: ‘255.255.255.0’
          vlan: ‘External’
          allow_service: ‘default’
        - name: ‘Internal-SelfIP’
          address: ‘192.10.10.11’
          netmask: ‘255.255.255.0’
          vlan: ‘Internal’
          allow_service: ‘default’
    delegate_to: localhost

- name: Onboard BIG-IP1 and BIG-IP2
  hosts: bigips_cisco
  connection: local
  tasks:

  - name: Configure NTP server on BIG-IP
    bigip_device_ntp:
     server: “{{ inventory_hostname }}”
     user: “{{ username }}”
     password: “{{ password }}”
     ntp_servers: “172.27.1.1”
     validate_certs: False
    delegate_to: localhost

  - name: Manage SSHD setting on BIG-IP
```

```
    bigip_device_sshd:
     server: “{{ inventory_hostname }}”
     user: “{{ username }}”
     password: “{{ password }}”
     banner: “enabled”
     banner_text: “ Welcome to BIG-IP”
     validate_certs: False
    delegate_to: localhost

  - name: Manage BIG-IP DNS settings
    bigip_device_dns:
     server: “{{ inventory_hostname }}”
     user: “{{ username }}”
     password: “{{ password }}”
     name_servers: “8.8.8.8”
     search: “local”
     ip_version: “4”
     validate_certs: False
    delegate_to: localhost

- name: Add Pool members
    bigip_pool:
      server: “{{inventory_hostname}}”
      user: “admin”
      password: “admin”
      name: “http-pool”
      host: “192.10.10.100”
      port: “80”
      validate_certs: False
    delegate_to: localhost

  - name: Add Virtual Server
    bigip_virtual_server:
      server: “{{inventory_hostname}}”
      user: “admin”
      password: “admin”
      name: “http_vs”
      destination: “10.10.10.100”
      port: “80”
      enabled_vlans:
       - “Internal”
       - “External”
      profiles_both: “http”
      pool: “http-pool”
      snat: “automap”
      validate_certs: False
    delegate_to: localhost
```

Configuration on the NX-OS



```
version 7.0(3)I2(1)
switchname n9300
vdc n9300 id 1
  limit-resource vlan minimum 16 maximum 4094
  limit-resource vrf minimum 2 maximum 4096
  limit-resource port-channel minimum 0 maximum 511
  limit-resource u4route-mem minimum 248 maximum 248
  limit-resource u6route-mem minimum 96 maximum 96
  limit-resource m4route-mem minimum 58 maximum 58
  limit-resource m6route-mem minimum 8 maximum 8

feature telnet
feature nxapi
feature bash-shell
feature scp-server

no password strength-check
username admin password 5 $1$teKRZ4fr$ZSRZkZl25sF0UuxmtAFAi1  role network-admin
ip domain-lookup
copp profile strict
snmp-server user admin network-admin auth md5 0x4021c6b85919ade68a97863513fcb3b1 priv 0x
rmon event 1 log trap public description FATAL(1) owner PMON@FATAL
rmon event 2 log trap public description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 log trap public description ERROR(3) owner PMON@ERROR
rmon event 4 log trap public description WARNING(4) owner PMON@WARNING
rmon event 5 log trap public description INFORMATION(5) owner PMON@INFO

vlan 1,10,20,50-51,54-55,100,800
vlan 20
  name VLAN020

vrf context management
  ip route 0.0.0.0/0 10.192.74.1

interface Ethernet1/17
  switchport mode trunk
  switchport trunk allowed vlan 54-55

interface Ethernet1/18
  switchport mode trunk
  switchport trunk allowed vlan 54-55
```

Configuration on the BIG-IP

VLAN/Self IP/Pool and Virtual servers will be configured.

Appendix—The F5 Solution

Ansible Templates

create_unmanged_ldev.j2

```
<vnsLDevVip trunking=”no” svcType=”ADC” packageModel=”” name=”{{logicalDeviceCluster_name}}”
mode=”legacy-Mode” managed=”no” isCopy=”no” funcType=”GoTo” dn=”uni/tn-{{tenant_name}}/lDevVip-
{{logicalDeviceCluster_name}}” devtype=”PHYSICAL” contextAware=”single-Context”>
   <vnsRsALDevToPhysDomP tDn=”uni/phys-{{physical_domain_name}}” />
   <vnsCDev name=”Device2” vmName=”” vcenterName=”” devCtxLbl=””>
      <vnsCCred name=”username” value=”a” />
      <vnsCCredSecret name=”password” />
      <vnsCIf name=”Device2_IntExt” vnicName=””>
         <vnsRsCIfPathAtt tDn=”topology/{{device2_interface1_connectionPath}}” />
      </vnsCIf>
   </vnsCDev>
   <vnsCDev name=”Device1” vmName=”” vcenterName=”” devCtxLbl=””>
      <vnsCCred name=”username” value=”a” />
      <vnsCCredSecret name=”password” />
      <vnsCIf name=”Device1_IntExt” vnicName=””>
         <vnsRsCIfPathAtt tDn=”topology/{{device1_interface1_connectionPath}}” />
      </vnsCIf>
   </vnsCDev>
   <vnsLIf name=”External” encap=”vlan-{{consumer_vlan}}”>
      <vnsRsCIfAttN tDn=”uni/tn-{{tenant_name}}/lDevVip-{{logicalDeviceCluster_name}}/cDev-Device1/
cIf-[Device1_IntExt]” />
      <vnsRsCIfAttN tDn=”uni/tn-{{tenant_name}}/lDevVip-{{logicalDeviceCluster_name}}/cDev-Device2/
cIf-[Device2_IntExt]” />
   </vnsLIf>
   <vnsLIf name=”Internal” encap=”vlan-{{provider_vlan}}”>
      <vnsRsCIfAttN tDn=”uni/tn-{{tenant_name}}/lDevVip-{{logicalDeviceCluster_name}}/cDev-Device1/
cIf-[Device1_IntExt]” />
      <vnsRsCIfAttN tDn=”uni/tn-{{tenant_name}}/lDevVip-{{logicalDeviceCluster_name}}/cDev-Device2/
cIf-[Device2_IntExt]” />
   </vnsLIf>
</vnsLDevVip>
```

Contract.j2

```
<fvTenant dn=”uni/tn-{{tenant_name}}” name=”{{tenant_name}}”>
    <vzBrCP ownerTag=”” ownerKey=”” name=”{{contract_name}}” descr=”” scope=”context”
prio=”unspecified”>
        <vzSubj name=”http” descr=”” prio=”unspecified” revFltPorts=”yes” provMatchT=”AtleastOne”
consMatchT=”AtleastOne”>
            <vzRsSubjFiltAtt tnVzFilterName=”default”/>
        </vzSubj>
        <vzSubj name=”https” descr=”” prio=”unspecified” revFltPorts=”yes” provMatchT=”AtleastOne”
consMatchT=”AtleastOne”>
            <vzRsSubjFiltAtt tnVzFilterName=”default”/>
        </vzSubj>
    </vzBrCP>
</fvTenant>
```

service_graph_template.j2

```
<vnsAbsGraph uiTemplateType=”UNSPECIFIED” ownerTag=”” ownerKey=”” name=”{{SGtemplate_name}}”
dn=”uni/tn-{{tenant_name}}/AbsGraph-{{SGtemplate_name}}” descr=””>
   <vnsAbsTermNodeCon ownerTag=”” ownerKey=”” name=”T1” descr=””>
```

```
      <vnsAbsTermConn ownerTag=”” ownerKey=”” name=”1” descr=”” attNotify=”no” />
      <vnsInTerm name=”” descr=”” />
      <vnsOutTerm name=”” descr=”” />
   </vnsAbsTermNodeCon>
   <vnsAbsTermNodeProv ownerTag=”” ownerKey=”” name=”T2” descr=””>
      <vnsAbsTermConn ownerTag=”” ownerKey=”” name=”1” descr=”” attNotify=”no” />
      <vnsInTerm name=”” descr=”” />
      <vnsOutTerm name=”” descr=”” />
   </vnsAbsTermNodeProv>
   <vnsAbsConnection ownerTag=”” ownerKey=”” name=”C1” descr=”” unicastRoute=”yes”
directConnect=”no” connType=”external” connDir=”provider” adjType=”L2”>
      <vnsRsAbsConnectionConns tDn=”uni/tn-{{tenant_name}}/AbsGraph-{{SGtemplate_name}}/AbsNode-N1/
AbsFConn-consumer” />
      <vnsRsAbsConnectionConns tDn=”uni/tn-{{tenant_name}}/AbsGraph-{{SGtemplate_name}}/
AbsTermNodeCon-T1/AbsTConn” />
   </vnsAbsConnection>
   <vnsAbsConnection ownerTag=”” ownerKey=”” name=”C2” descr=”” unicastRoute=”yes”
directConnect=”no” connType=”external” connDir=”provider” adjType=”L2”>
      <vnsRsAbsConnectionConns tDn=”uni/tn-{{tenant_name}}/AbsGraph-{{SGtemplate_name}}/
AbsTermNodeProv-T2/AbsTConn” />
      <vnsRsAbsConnectionConns tDn=”uni/tn-{{tenant_name}}/AbsGraph-{{SGtemplate_name}}/AbsNode-N1/
AbsFConn-provider” />
   </vnsAbsConnection>
   <vnsAbsNode ownerTag=”” ownerKey=”” name=”N1” descr=”” shareEncap=”no” sequenceNumber=”0”
routingMode=”unspecified” managed=”no” isCopy=”no” funcType=”GoTo” funcTemplateType=”ADC_TWO_ARM”>
      <vnsAbsFuncConn ownerTag=”” ownerKey=”” name=”consumer” descr=”” attNotify=”no” />
      <vnsAbsFuncConn ownerTag=”” ownerKey=”” name=”provider” descr=”” attNotify=”no” />
      <vnsRsNodeToLDev tDn=”uni/tn-{{tenant_name}}/lDevVip-{{logicalDeviceCluster_name}}” />
   </vnsAbsNode>
</vnsAbsGraph>
```

deviceSelectionPolicy.j2

```
<vnsLDevCtx nodeNameOrLbl=”N1” name=”” graphNameOrLbl=”{{SGtemplate_name}}” dn=”uni/tn-{{tenant_
name}}/ldevCtx-c-{{contract_name}}-g-{{SGtemplate_name}}-n-N1” descr=”” ctrctNameOrLbl=”{{contract_
name}}”>
   <vnsRsLDevCtxToLDev tDn=”uni/tn-{{tenant_name}}/lDevVip-{{logicalDeviceCluster_name}}” />
   <vnsLIfCtx name=”” descr=”” permitLog=”no” connNameOrLbl=”provider”>
      <vnsRsLIfCtxToBD tDn=”uni/tn-{{tenant_name}}/BD-{{providerBD_name}}” />
      <vnsRsLIfCtxToLIf tDn=”uni/tn-{{tenant_name}}/lDevVip-{{logicalDeviceCluster_name}}/lIf-
Internal” />
   </vnsLIfCtx>
   <vnsLIfCtx name=”” descr=”” permitLog=”no” connNameOrLbl=”consumer”>
      <vnsRsLIfCtxToBD tDn=”uni/tn-{{tenant_name}}/BD-{{consumerBD_name}}” />
      <vnsRsLIfCtxToLIf tDn=”uni/tn-{{tenant_name}}/lDevVip-{{logicalDeviceCluster_name}}/lIf-
External” />
   </vnsLIfCtx>
</vnsLDevCtx>
```

apply_graph.j2

```
<fvTenant dn=”uni/tn-{{tenant_name}}” name=”{{tenant_name}}”>
   <vzBrCP descr=”” dn=”uni/tn-{{tenant_name}}/brc-{{contract_name}}” name=”{{contract_name}}”
ownerKey=”” ownerTag=”” prio=”unspecified” scope=”context” targetDscp=”unspecified”>
      <vzSubj consMatchT=”AtleastOne” descr=”” name=”http” prio=”unspecified”
provMatchT=”AtleastOne” revFltPorts=”yes” targetDscp=”unspecified”>
         <vzRsSubjFiltAtt directives=”” tnVzFilterName=”default” />
         <vzRsSubjGraphAtt tnVnsAbsGraphName=”{{SGtemplate_name}}”/>
      </vzSubj>
   </vzBrCP>
</fvTenant>
```

attach_cons_prov_contract.j2

```
<fvAp prio="unspecified" ownerTag="" ownerKey="" name="{{appProfile_name}}" dn="uni/tn-{{tenant_
name}}/ap-{{appProfile_name}}" descr="">
   <fvAEPg prio="unspecified" name="{{providerEPG_name}}" descr="" prefGrMemb="exclude"
pcEnfPref="unenforced" matchT="AtleastOne" isAttrBasedEPg="no" fwdCtrl="">
      <fvRsProv prio="unspecified" matchT="AtleastOne" tnVzBrCPName="{{contract_name}}" />
   </fvAEPg>
   <fvAEPg prio="unspecified" name="{{consumerEPG_name}}" descr="" prefGrMemb="exclude"
pcEnfPref="unenforced" matchT="AtleastOne" isAttrBasedEPg="no" fwdCtrl="">
      <fvRsCons prio="unspecified" tnVzBrCPName="{{contract_name}}" />
   </fvAEPg>
</fvAp>
```

F5 Networks, Inc. 401 Elliott Avenue West, Seattle, WA 98119 888-882-4447 www.f5.com

Americas
info@f5.com

Asia-Pacific
apacinfo@f5.com

Europe/Middle-East/Africa
emeainfo@f5.com

Japan
f5j-info@f5.com

Solutions for
an application world.