



A10 Device Package for Cisco Application Centric Infrastructure (ACI)

Step by Step Instructions for Deploying Rich Application Delivery and Security Capabilities in a Shared Infrastructure

Table of Contents

Overview.....	3
Deployment Prerequisites	3
Deployment Modes.....	3
The A10 Networks Device Package for Cisco ACI.....	3
Using Multiple Service Graphs for Larger ADC Configurations.....	9
Configuring Server Load Balancing Features on ACOS through APIC.....	12
Configuring the A10 ADC Using APIC XML	13
Summary	17
Appendix.....	18
XML Reference Guide	18
About A10 Networks.....	22

Disclaimer

This document does not create any express or implied warranty about A10 Networks or about its products or services, including but not limited to fitness for a particular use and noninfringement. A10 Networks has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks assumes no responsibility for its use. All information is provided “as-is.” The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks for current information regarding its products or services. A10 Networks’ products and services are subject to A10 Networks’ standard terms and conditions.

Overview

The A10 Networks® Device Package for Cisco Application Centric Infrastructure (ACI) allows integration of A10 Networks Advanced Core Operating System (ACOS®) devices with Cisco Application Policy Infrastructure Controller (Cisco APIC). Cisco ACI provides a centralized fabric control and automation framework capable of managing application policies. Within ACI, the Cisco APIC serves as the single point of automation and fabric element management in both physical and virtual environments. A10 Networks aCloud™ Services Architecture is integrated into the Cisco ACI application policy framework through a plugin also referred to as Device Package. The Device Package allows Cisco ACI to communicate with A10 Networks Application Delivery Controllers (ADC) and deploy policies automatically.

This deployment guide assumes that the user has basic knowledge of Cisco ACI, including how to configure Cisco APIC, and is familiar with A10 Networks ACOS 4.0 ADC functionality. As a pre-requisite for this integration, the user will have to configure the A10 Networks ADC device with a management IP address and administrator credentials to use the A10 ADC.

In this Deployment Guide, the workflow is based on Cisco APIC GUI driven steps for integrating the A10 ADC with APIC. If XML scripting is used for configuring APIC, a reference guide is given with sample scripts in the Appendix section of this guide. Should the user choose to define various parameters via script, a brief tutorial is given on this in the section "[Configuring the A10 ADC Using APIC XML.](#)"

Note that this integration will only be supported on A10 Networks hardware and virtual platforms running ACOS 4.0. The screenshot examples shown below are for Cisco APIC version 1.0 (1K).

Deployment Prerequisites

- A10 Networks Thunder ADC line of Application Delivery Controllers (physical and virtual appliances running ACOS 4.0)
- A10 Device Package for Cisco Application Centric Infrastructure (Cisco ACI)
- A10 Networks Advanced Core Operating System (ACOS)
- Cisco ACI with Application Policy Infrastructure (APIC) version 1.0 (1K)

Deployment Modes

- Single Service Graph Creation for Basic ADC Topology
- Multiple Service Graph Creation for Larger ADC Topologies

The A10 Networks Device Package for Cisco ACI

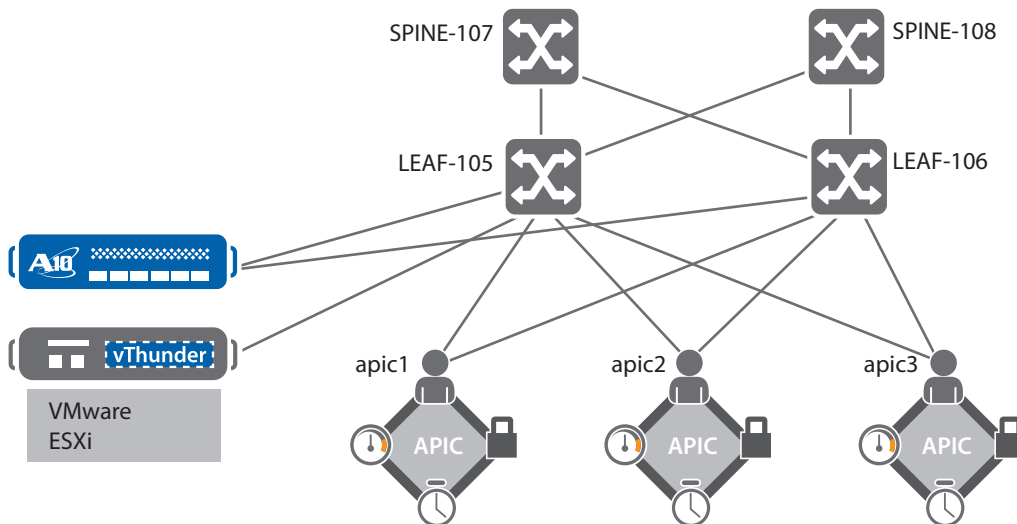
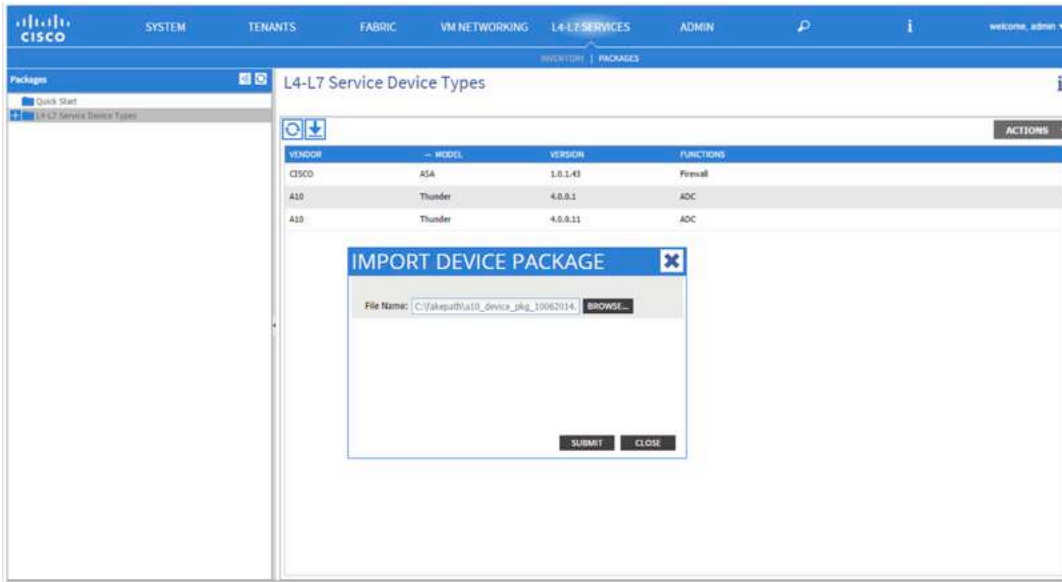


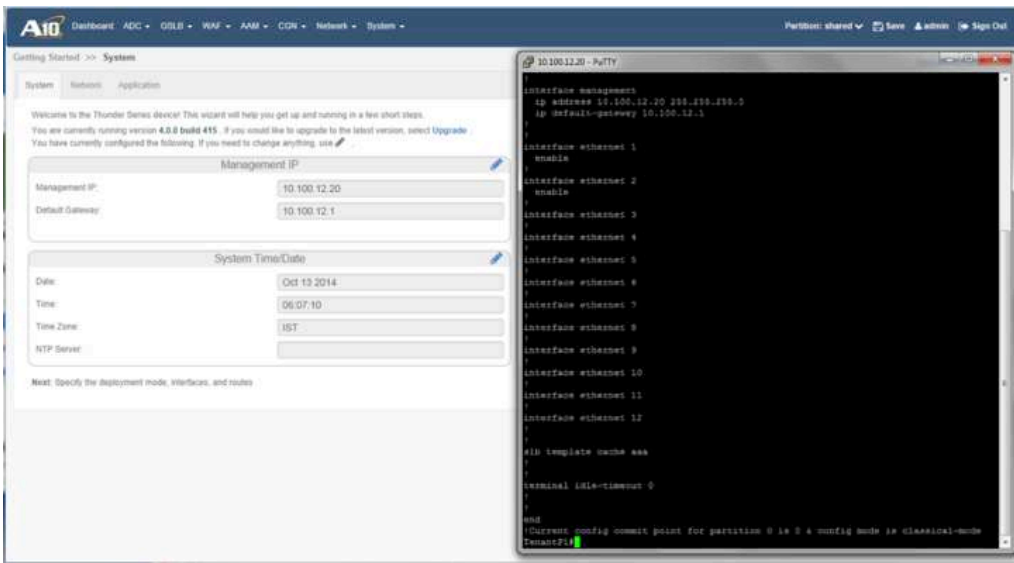
Figure 1. Physical topology of the Cisco APIC-A10 Networks A10 ADC integrated environment

These are the steps to add the device package, configure the A10 ADC and complete the rest of the configuration requirements.

- 1. Add the A10 Networks Device Package:** After logging onto Cisco APIC, the user will have to upload A10's Device Package in order to enable APIC to communicate with A10 Networks aXAPI® REST-based API and configure the A10 ADC. In the GUI, the user will go into "L4-L7 services" and click on "Packages," "expand," "L4-L7 Service Device Types," then click "Actions" to import the Device Package. The Device Package can be obtained from A10's software downloads page at <https://www.a10networks.com/support-axseries/downloads/downloads.php>. Upon upload, the package name will show up on the left-hand directory and in the table.



- 2. Configure the A10 ADC:** Next, the user will need to configure the management IP address of the A10 ADC in order to communicate and connect with the Cisco APIC. This can be done through the CLI.



- 3. Provision Tenants and Assign the A10 ADC:** After installing the A10 ACI package on APIC, the user will have to define a new tenant, create a network and go through the steps for tenant setup under the "Tenants" tab in the APIC GUI. While creating the tenant, the user will define the tenant network and create external and internal bridge domains relative to the A10 ADC in order to define the network topology. When the user configures bridge domains, a subnet for each domain will also need to be specified.

CREATE TENANT

STEP 1 > TENANT

1. TENANT 2. NETWORK

Tenant Identity

Specify tenant details

Name:

Description:

Tags:

Monitoring Policy:

Security Domains:

Select	Name	Description
<input type="checkbox"/>	all	
<input type="checkbox"/>	ingest	

STEP 2 > NETWORK

1. TENANT 2. NETWORK

Tenant Tenant_Demo

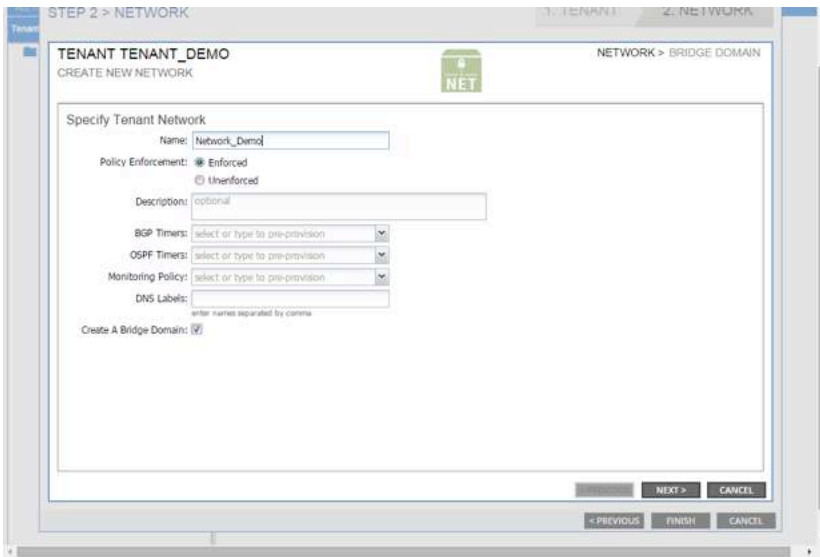
Create A Network

Click "+" to add network

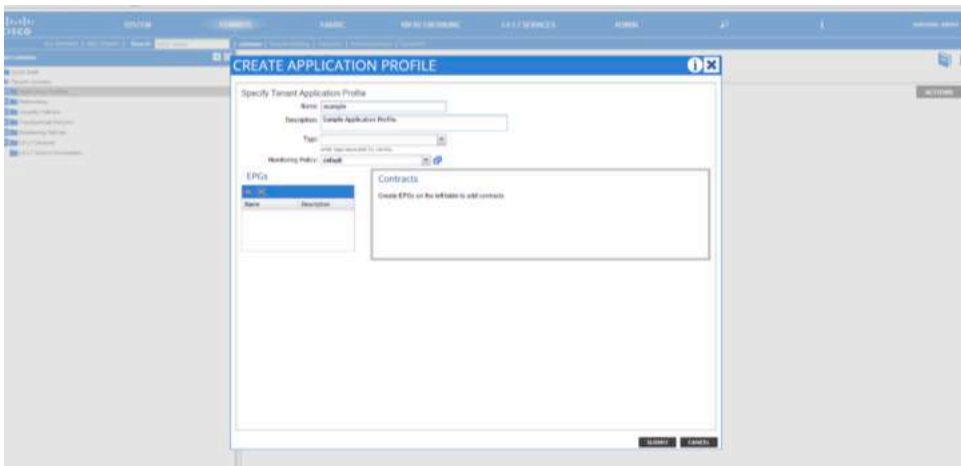


Take me to this tenant when I click finish

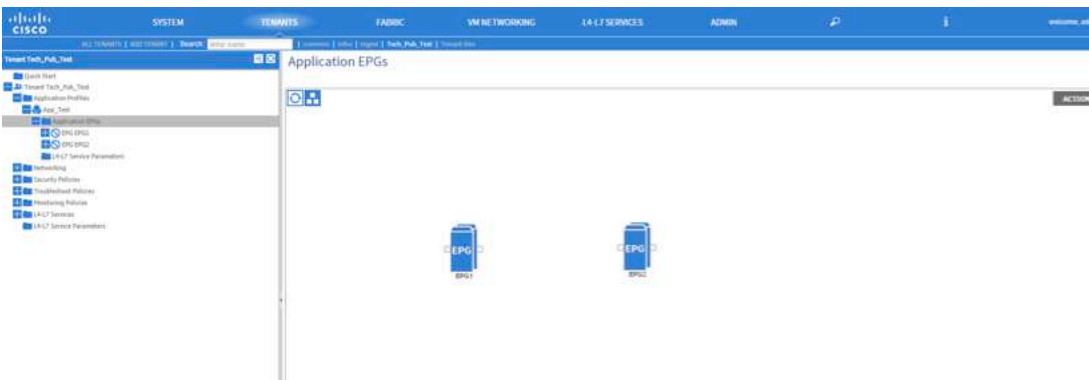
PREVIOUS FINISH CANCEL



- 4. Create the Application Profile:** The user will next have to create an application profile. Select the tenant name from the top task-pane under “Tenants,” and click on the folder underneath the tenant name in the left-hand pane titled “Application Profiles.” Click “Actions” to create an application profile.



- 5. Create Endpoint Groups (EPGs) within the Application Profile:** For the A10 Device Package to work successfully, the user will need to create two connectors – the consumer points towards the clients, and the provider points towards the server. This represents the client and real servers of a basic ADC topology. The user maps out this topology through creation of EPGs. When creating the application profile, the user will add EPGs in the box in the left-hand corner and fill out the appropriate fields in the pop-up form. This will include defining the bridge domains for the network.



6. Create the Device Cluster: To create a device cluster, select “Device Clusters” under “L4-L7 Services” and fill out the pop-up form. In the form, the user will have to name the device cluster and then choose the device type, device package and physical domains. For this current release, A10 will only support a single device for a device cluster.

For the device cluster, the user needs to define the following fields:

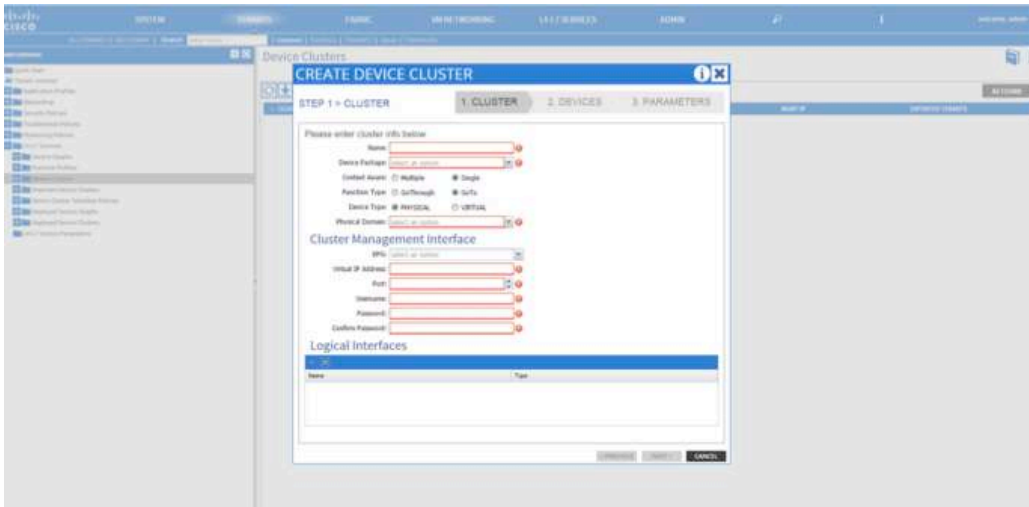
- Name
- Device Package
- Context Aware (currently A10 only supports “Single” context; “Multiple” context will be supported in a later release)
- Function Type
- Device Type
- Domain (this will depend on the device type; select physical domains if deploying physical A10 ADCs or VMM domain for an A10 Networks vThunder® ADC line of virtual appliances device)
- EPG (select the management EPG if previously configured)
- Virtual IP Address (this is the management IP address of the A10 ADC)
- Port (default is 443 unless the user wishes to change this)
- Username and password of the A10 ADC

Depending on the device type, users will have different domains available to them – either physical or virtual. The user will have to also define the IP of the device. Additionally the user will specify the logical interface, which is where the external and internal interface is defined for the cluster. It is important to define this correctly, as failure to establish the external and internal interface will prevent successful configuration of the A10 ADC from APIC.

After defining the Logical Interface, the user creates a concrete device, where the user specifies:

- Device name
- VM name
- vCenter name
- Management address
- Management port
- Username and password
- Interface

Defining the interface within the concrete device form is critically important, as this binds the logical interface to the concrete device. The interface name must be a number prefixed by the letter “e” to ensure communication between the syntax of the A10 ADC and the Cisco APIC. When the user creates a device cluster and configures the logical interface and concrete device, this allows for ACOS specific information to be registered within APIC, and only after defining this component can any ADC configurations made on APIC be pushed to the A10 ADC (as the management IP is given in the device cluster form). Under the configured device cluster, when the device state under Function Node is listed as “Stable,” the configuration of the A10 ADC has been implemented successfully.



- 7. Create Service Graphs:** To create the Service Graph, go to “L4-L7 Services” and select “Create Service Graph.” The user will have to name the graph and create connections between the endpoints that are defined as either L2 or L3.

After the Service Graph is defined, user must go to “Function Node” under the graph name. This is where interface configuration for the ACOS device will be defined. For the device configuration, there needs to be an external and internal interface defined with an IP address and netmask under “Device Config.” This will be done under the configuration parameters in the Function Node.

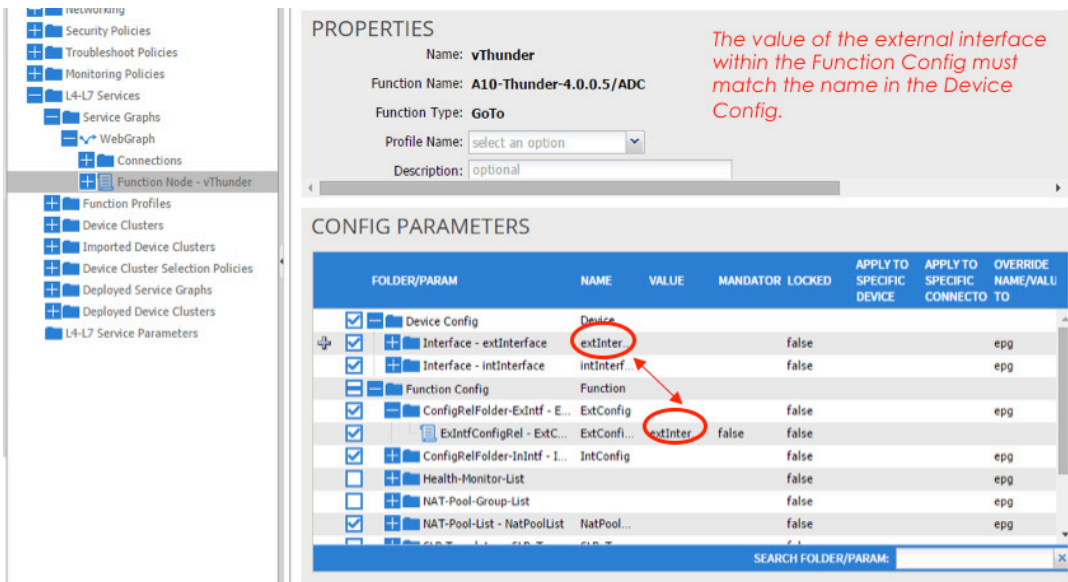
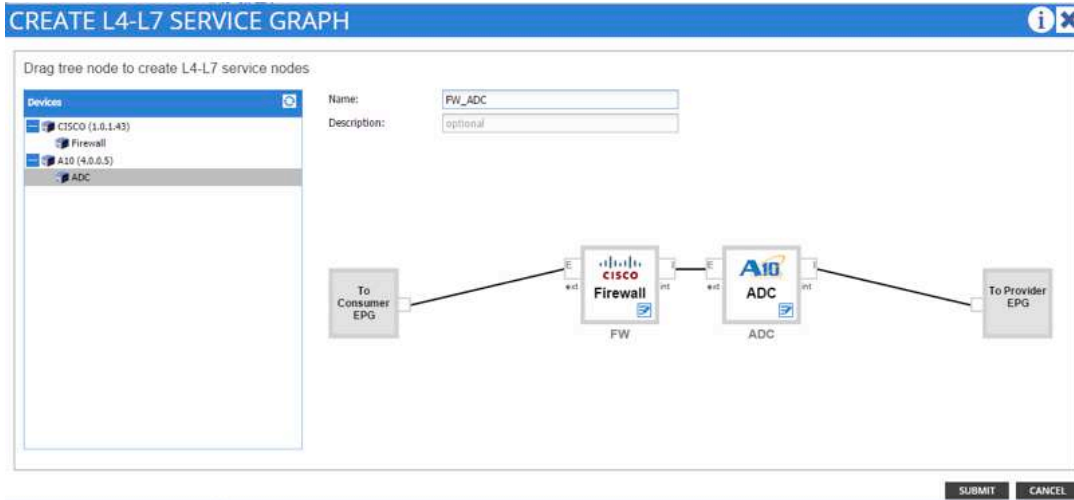
The next step is to define which network will communicate with the external/internal addresses within the “Function Config” folder under the Function Node. For communication to be successful, the user will have to set the value of the external interface to be the same as the name of the folder in which the external interface information resides in the “Device Config” section.

Note that the external and internal interfaces set in this directory represent virtual Ethernet interfaces. These will be referred to by VLANs generated by the APIC. If the customer is using a physical A10 Thunder ADC device, the APIC will configure tagged VLANs. If the customer is using a vThunder ADC appliance, the VLANs will be untagged.

The consumer endpoint group (EPG) is the client side, and the provider endpoint is the server side, with the A10 ADC sitting between the consumer and provider EPGs. The user will define this aspect further in later steps. Basic setup of a Service Graph is necessary to assign the service contracts and ensure communication between the endpoints.

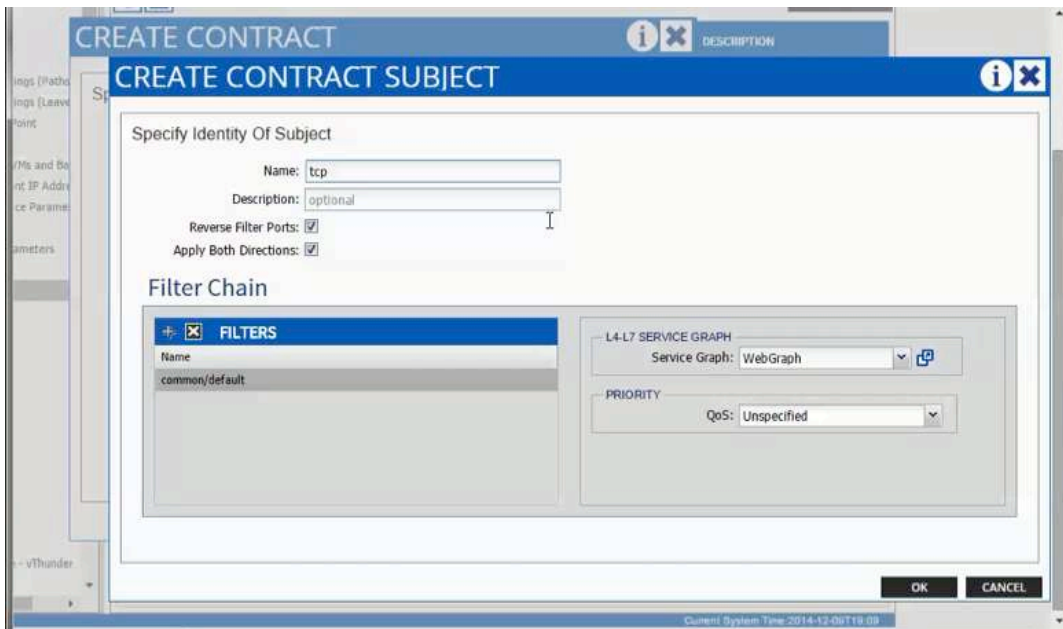
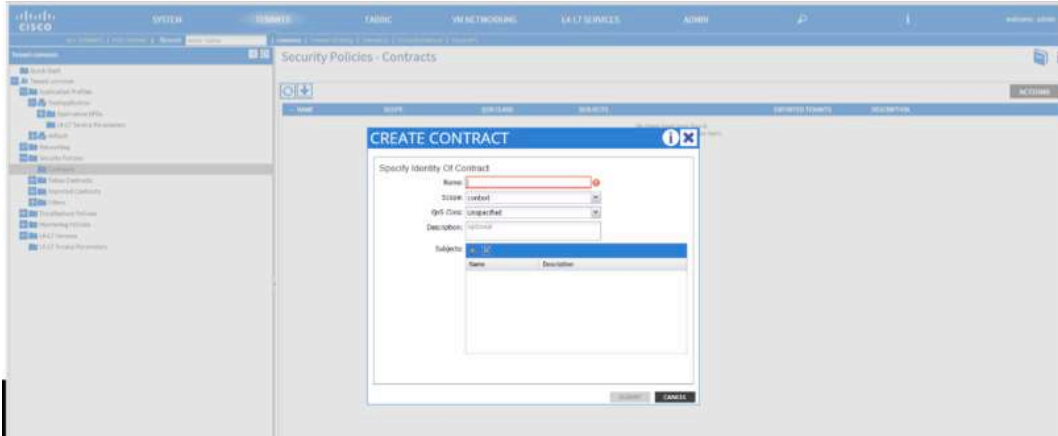
Using Multiple Service Graphs for Larger ADC Configurations

With larger ADC configurations, users may want to break them down into logically contained modules. For example, configurations that need to be applied on a given A10 device can be split along the lines of different virtual service instances. This will also simplify configuring and removing configuration of a virtual service without affecting other virtual services running on the same device.



In this scenario, include all of the virtual servers, service groups, servers and related configurations for a given virtual service instance inside the Function Config section of the Service Graph. Configurations such as interface and VLAN-related information that may be shared or common to other virtual services can be applied at the EPG level. These Service Graphs can then be combined to be deployed on the same device by including them in the contract specified between the EPGs. Refer to the next section on “Creating Security Contracts” for more details.

- 8. Create Security Contracts:** In order to assign the L4-L7 parameters of the A10 ADC to the tenants, the user must create security contracts to define the subjects and filters, and associate the contract with configured EPGs. To create the security contract profile, the user should click the folder titled “Security Policies” in the given tenants folder directory, and select “Actions” to create a security contract. When creating the contract, the user will have to create a contract subject in which the Service Graph name is given and the subject identity is defined (such as HTTP).

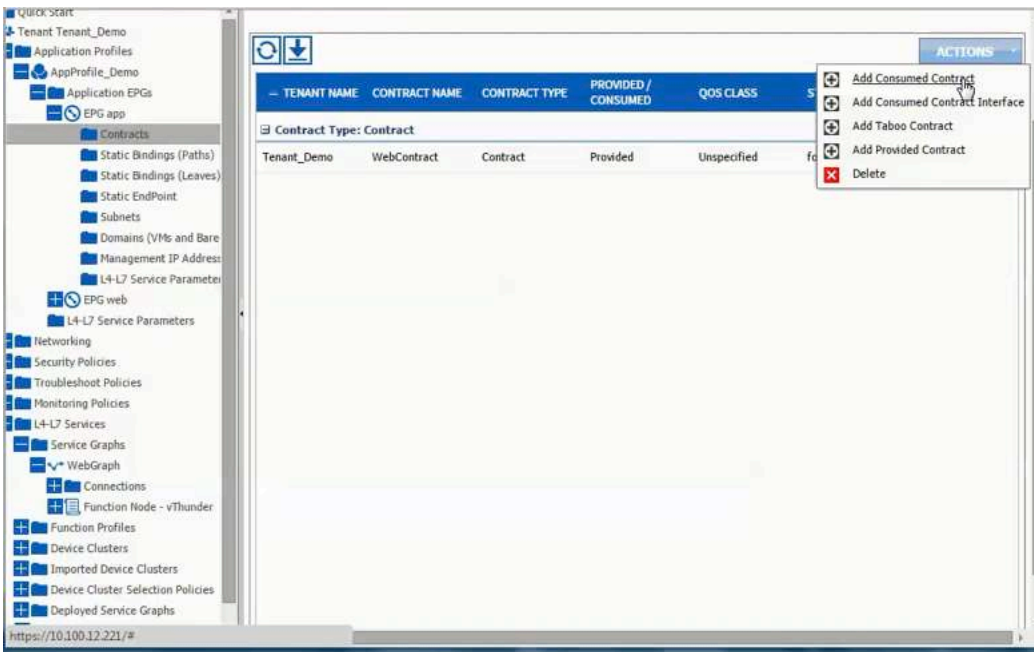


Note that when users define a subject identity, they can choose from multiple Service Graphs. It is suggested that for the section specifying filters, the user select default to allow for all traffic types. However, in this release there are a limited number of protocols the A10 ADC will support – please refer to the section titled [“Configuring Server Load Balancing Features on ACOS through APIC.”](#)

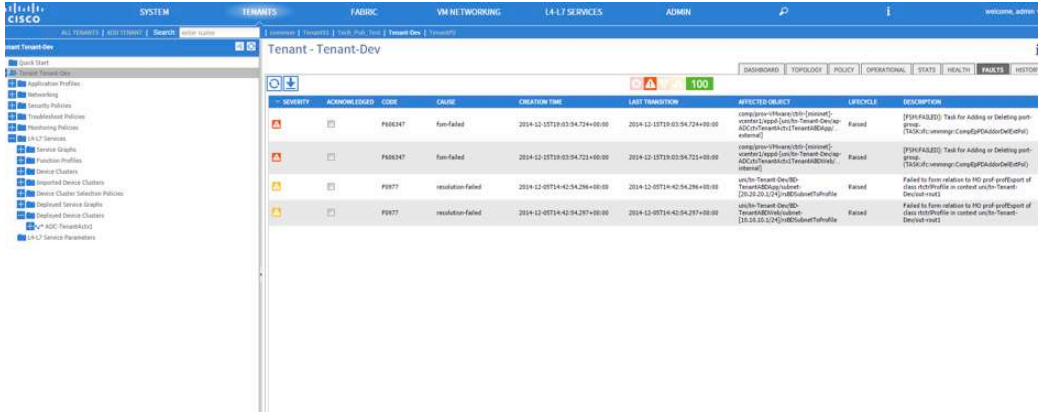
- 9. Creating the Device Cluster Policy:** In “L4-L7 Services,” provide the contract name, Service Graph name, node name and the device cluster in the form to create the “Logical Device Context.” Access this form by right-clicking “Device Cluster Selection Policies.”



10. Define the Consumer and Provider Endpoints: Go back to the EPG under Application Profiles and define the consumer endpoint by right-clicking on “EPG External” and selecting “Add Consumed Contract.”



11. How to Verify Graph Deployment and Device Functionality: To verify device configuration and network status, look at “Deployed Device Clusters” in the left-hand directory and check the section titled “Faults” in the window. To check for all faults, click on the tenant name in the top task-pane and click on the “Faults” section.



Configuring Server Load Balancing Features on ACOS through APIC

To configure server load balancing features on the A10 ADC, navigate to the Function Node section of the Service Graph. The Config Parameters section of this page contains all configurable ACOS features. Note that the only protocols currently supported for traffic filtering and load balancing are TCP, UDP, HTTP and Fast-HTTP virtual ports. Select the plus sign next to a folder to expand it and reveal all of the available parameters and sub-folders.

1. To add a Network Address Translation (NAT) pool, double-click on the NAT-Pool-List folder. You must specify a name for any folder that contains configuration items. This name is only used on the APIC, not on the A10 ADC. Click "Update."
2. Expand the NAT-Pool-List folder to reveal the NAT-Pool sub-folder. This folder represents the NAT pool object. Click the light blue plus sign to the right of the checkbox to add a new NAT pool to the configuration. Specify a name in the NAME column that will be used to identify the NAT pool on the APIC device (again, this does not necessarily correlate with the name on the A10 ADC configuration). Click "Update."
3. You can now specify the configurable parameters for the NAT pool. Configure the name of the NAT pool on the A10 ADC and double-click on the Pool-Name parameter. In the NAME column, specify "Pool-Name." In the value column, give the NAT pool a name such as "pool1" and click "Update." Repeat this process for each parameter, providing the name of the parameter in the NAME column, and the value in the VALUE column.
4. Click "SUBMIT" to complete the configuration.

The figure below shows an example configuration of a NAT pool.

Configuring the A10 ADC Using APIC XML

CONFIG PARAMETERS								
FOLDER/PARAM	NAME	VALUE	MANDATORY	LOCKED	APPLY TO SPECIFIC DEVICE	APPLY TO SPECIFIC CONNECTOR	OVERRIDE NAME/VALUE TO	
<input type="checkbox"/> + Device Config	Device							
<input type="checkbox"/> - Function Config	Function							
<input type="checkbox"/> + ConfigRelFolder-ExIntf				false			epg	
<input type="checkbox"/> + ConfigRelFolder-InIntf				false			epg	
<input type="checkbox"/> + Health-Monitor-List				false			epg	
<input type="checkbox"/> + NAT-Pool-Group-List				false			epg	
<input checked="" type="checkbox"/> - NAT-Pool-List - natpoolist	natpoolist			false			epg	
<input checked="" type="checkbox"/> + NAT-Pool - natpool	natpool			false			epg	
<input checked="" type="checkbox"/> End-Address - End-Address	End-Address	10.10.10.11	true	false				
<input checked="" type="checkbox"/> NetMask - NetMask	NetMask	255.255.25...	true	false				
<input checked="" type="checkbox"/> Pool-Name - Pool-Name	Pool-Name	pool1	true	false				
<input checked="" type="checkbox"/> Start-Address - Start-Address	Start-Address	10.10.10.10	true	false				
<input type="checkbox"/> + SLB-Templates				false			epg	
<input type="checkbox"/> + Server-List				false			epg	
<input type="checkbox"/> + Service-Group-List				false			epg	
<input type="checkbox"/> + Virtual-Server-List				false			epg	

Additionally, it is possible for the user to set server load balancing configurations to the functional node via an XML Script. The example below illustrates how the user would generate this configuration using a sample XML schema. All schemas can be downloaded from Cisco APIC by clicking on the arrow icon for any of the modules (i.e., should the user want to create a tenant, the schema can be downloaded from the GUI after clicking on the tenant page or generated in a custom fashion).

In this example, the user configures REST for the A10 Thunder ADC server load balancing configuration, which will create:

1. Three Real Servers: S1 / S2 / S3 with real ports of TCP80 / TCP22 / UDP53
2. Three Service Groups: sg-tcp80-1 / sg-tcp22-1 / sg-udp53-1
3. Two IP NAT pools: Pool1 / Pool2
4. One TCP template / one Virtual Server template
5. One Virtual Server: VS1

Note that this example was taken from the XML Script file for creating the Service Graph. When users implement a server load balancing configuration via XML, they will have to include this script within the same XML file for Service Graph creation (the file name is CreateGraphWithParams.xml).

1. Creating three real servers with port and protocol selection.

```
<vnsAbsFolder key="Server-List" name="ServerList">
  <!-- SLB Server 1-->
  <vnsAbsFolder key="Server" name="Server1" locked="true">
    <vnsAbsParam key="Name" name="ServerName" value="S1"/>
    <vnsAbsParam key="Host" name="ServerAddress" value="20.20.20.51"/>
    <vnsAbsFolder key="Port-List" name="PortList">
      <vnsAbsFolder key="Port" name="Port1">
        <vnsAbsParam key="Port-Number" name="Port" value="80"/>
        <vnsAbsParam key="Protocol" name="Protocol" value="tcp"/>
      </vnsAbsFolder>
      <vnsAbsFolder key="Port" name="Port2">
        <vnsAbsParam key="Port-Number" name="Port" value="22"/>
        <vnsAbsParam key="Protocol" name="Protocol" value="tcp"/>
      </vnsAbsFolder>
    </vnsAbsFolder>
  </vnsAbsFolder>
</vnsAbsFolder>
```

```

    <vnsAbsFolder key="Port" name="Port3">
      <vnsAbsParam key="Port-Number" name="Port" value="53"/>
      <vnsAbsParam key="Protocol" name="Protocol" value="udp"/>
    </vnsAbsFolder>
  </vnsAbsFolder>
</vnsAbsFolder>
<!-- SLB Server 2 -->
<vnsAbsFolder key="Server" name="Server2">
  <vnsAbsParam key="Name" name="ServerName" value="S2"/>
  <vnsAbsParam key="Host" name="ServerAddress" value="20.20.20.52"/>
  <vnsAbsFolder key="Port-List" name="PortList">

    <vnsAbsFolder key="Port" name="Port1">
      <vnsAbsParam key="Port-Number" name="Port" value="80"/>
      <vnsAbsParam key="Protocol" name="Protocol" value="tcp"/>
    </vnsAbsFolder>

    <vnsAbsFolder key="Port" name="Port2">
      <vnsAbsParam key="Port-Number" name="Port" value="22"/>
      <vnsAbsParam key="Protocol" name="Protocol" value="tcp"/>
    </vnsAbsFolder>

    <vnsAbsFolder key="Port" name="Port3">
      <vnsAbsParam key="Port-Number" name="Port" value="53"/>
      <vnsAbsParam key="Protocol" name="Protocol" value="udp"/>
    </vnsAbsFolder>
  </vnsAbsFolder>
</vnsAbsFolder>
<!-- SLB Server 3 -->
<vnsAbsFolder key="Server" name="Server3">
  <vnsAbsParam key="Name" name="ServerName" value="S3"/>
  <vnsAbsParam key="Host" name="ServerAddress" value="20.20.20.53"/>
  <vnsAbsFolder key="Port-List" name="PortList">

    <vnsAbsFolder key="Port" name="Port1">
      <vnsAbsParam key="Port-Number" name="Port" value="80"/>
      <vnsAbsParam key="Protocol" name="Protocol" value="tcp"/>
    </vnsAbsFolder>

    <vnsAbsFolder key="Port" name="Port2">
      <vnsAbsParam key="Port-Number" name="Port" value="22"/>
      <vnsAbsParam key="Protocol" name="Protocol" value="tcp"/>
    </vnsAbsFolder>

    <vnsAbsFolder key="Port" name="Port3">
      <vnsAbsParam key="Port-Number" name="Port" value="53"/>
      <vnsAbsParam key="Protocol" name="Protocol" value="udp"/>
    </vnsAbsFolder>
  </vnsAbsFolder>
</vnsAbsFolder>
</vnsAbsFolder>

```

2. Creating a service group and assigning previously configured real servers as group members.

```

<vnsAbsFolder key="Service-Group-List" name="ServiceGroupList">
  <vnsAbsFolder key="Service-Group" name="ServiceGroup_TCP80">
    <vnsAbsParam key="Name" name="SgName" value="sg-tcp80-1"/>
    <vnsAbsParam key="Protocol" name="SgProtocol" value="tcp"/>
    <vnsAbsFolder key="Member-List" name="MemberList">
      <vnsAbsFolder key="Member" name="SgMember1">
        <vnsAbsParam key="Name" name="Server" value="S1"/>
        <vnsAbsParam key="Port" name="Port" value="80"/>
      </vnsAbsFolder>
      <vnsAbsFolder key="Member" name="SgMember2">
        <vnsAbsParam key="Name" name="Server" value="S2"/>
        <vnsAbsParam key="Port" name="Port" value="80"/>
      </vnsAbsFolder>
      <vnsAbsFolder key="Member" name="SgMember3">
        <vnsAbsParam key="Name" name="Server" value="S3"/>
        <vnsAbsParam key="Port" name="Port" value="80"/>
      </vnsAbsFolder>
    </vnsAbsFolder>
  </vnsAbsFolder>
</vnsAbsFolder>
<vnsAbsFolder key="Service-Group" name="ServiceGroup_TCP22">
  <vnsAbsParam key="Name" name="SgName" value="sg-tcp22-1"/>
  <vnsAbsParam key="Protocol" name="SgProtocol" value="tcp"/>
  <vnsAbsFolder key="Member-List" name="MemberList">
    <vnsAbsFolder key="Member" name="SgMember1">
      <vnsAbsParam key="Name" name="Server" value="S1"/>
      <vnsAbsParam key="Port" name="Port" value="22"/>
    </vnsAbsFolder>
    <vnsAbsFolder key="Member" name="SgMember2">
      <vnsAbsParam key="Name" name="Server" value="S2"/>
      <vnsAbsParam key="Port" name="Port" value="22"/>
    </vnsAbsFolder>
    <vnsAbsFolder key="Member" name="SgMember3">
      <vnsAbsParam key="Name" name="Server" value="S3"/>
      <vnsAbsParam key="Port" name="Port" value="22"/>
    </vnsAbsFolder>
  </vnsAbsFolder>
</vnsAbsFolder>
<vnsAbsFolder key="Service-Group" name="ServiceGroup_UDP53">
  <vnsAbsParam key="Name" name="SgName" value="sg-udp53-1"/>
  <vnsAbsParam key="Protocol" name="SgProtocol" value="udp"/>
  <vnsAbsFolder key="Member-List" name="MemberList">
    <vnsAbsFolder key="Member" name="SgMember1">
      <vnsAbsParam key="Name" name="Server" value="S1"/>
      <vnsAbsParam key="Port" name="Port" value="53"/>
    </vnsAbsFolder>
    <vnsAbsFolder key="Member" name="SgMember2">
      <vnsAbsParam key="Name" name="Server" value="S2"/>
      <vnsAbsParam key="Port" name="Port" value="53"/>
    </vnsAbsFolder>
    <vnsAbsFolder key="Member" name="SgMember3">
      <vnsAbsParam key="Name" name="Server" value="S3"/>
      <vnsAbsParam key="Port" name="Port" value="53"/>
    </vnsAbsFolder>
  </vnsAbsFolder>
</vnsAbsFolder>
</vnsAbsFolder>
</vnsAbsFolder>

```

3. Creating a virtual server, with assigned ports, protocols and service groups.

```

<vnsAbsFolder key="Virtual-Server-List" name="VirtualServerList">
  <vnsAbsFolder key="Virtual-Server" name="VirtualServer1">
    <vnsAbsParam key="Name" name="VSName" value="VIP1"/>
    <vnsAbsParam key="Ip-Address" name="VSAddr" value="10.10.10.101"/>
    <vnsAbsFolder key="Port-List" name="Port-List">
      <vnsAbsFolder key="Port" name="VSPort_HTTP">
        <vnsAbsParam key="Port-Number" name="Port" value="80"/>
        <vnsAbsParam key="Protocol" name="Proto" value="http"/>
        <vnsAbsParam key="Service-Group" name="ServiceGroup"
value="sg-tcp80-1"/>
        <vnsAbsParam key="Pool" name="SNAT" value="Pool1"/>
      </vnsAbsFolder>
      <vnsAbsFolder key="Port" name="VSPort_Fast-Http">
        <vnsAbsParam key="Port-Number" name="Port" value="8080"/>
        <vnsAbsParam key="Protocol" name="Proto" value="fast-
http"/>
        <vnsAbsParam key="Service-Group" name="ServiceGroup"
value="sg-tcp80-1"/>
        <vnsAbsParam key="Pool" name="SNAT" value="Pool2"/>
      </vnsAbsFolder>
      <vnsAbsFolder key="Port" name="VSPort_TCP">
        <vnsAbsParam key="Port-Number" name="Port" value="22"/>
        <vnsAbsParam key="Protocol" name="Proto" value="tcp"/>
        <vnsAbsParam key="Service-Group" name="ServiceGroup"
value="sg-tcp22-1"/>
        <vnsAbsParam key="Template-Tcp" name="Tcp_template"
value="TCP_1"/>
      </vnsAbsFolder>
      <vnsAbsFolder key="Port" name="VSPort_UDP">
        <vnsAbsParam key="Port-Number" name="Port" value="53"/>
        <vnsAbsParam key="Protocol" name="Proto" value="udp"/>
        <vnsAbsParam key="Service-Group" name="ServiceGroup"
value="sg-udp53-1"/>
      </vnsAbsFolder>
    </vnsAbsFolder>
  </vnsAbsFolder>
</vnsAbsFolder>

```

4. Creating two IP NAT pools.

```

<vnsAbsFolder key="NAT-Pool-List" name="NatPoolList">
  <vnsAbsFolder key="NAT-Pool" name="NatPool1">
    <vnsAbsParam key="Pool-Name" name="PoolName" value="Pool1"/>
    <vnsAbsParam key="Start-Address" name="StartAddress"
value="10.10.10.151"/>
    <vnsAbsParam key="End-Address" name="EndAddress"
value="10.10.10.152"/>
    <vnsAbsParam key="NetMask" name="Netmask" value="255.255.255.0"/>
  </vnsAbsFolder>
  <vnsAbsFolder key="NAT-Pool" name="NatPool2">
    <vnsAbsParam key="Pool-Name" name="PoolName" value="Pool2"/>
    <vnsAbsParam key="Start-Address" name="StartAddress"
value="10.10.10.161"/>
  </vnsAbsFolder>
</vnsAbsFolder>

```



```

        <vnsAbsParam key="End-Address" name="EndAddress"
value="10.10.10.162"/>
        <vnsAbsParam key="NetMask" name="Netmask" value="255.255.255.0"/>
    </vnsAbsFolder>
</vnsAbsFolder>

```

5. Creating and assigning the server load balancing template to the virtual server with connection rate limiting.

```

<!-- SLB Templates -->
<vnsAbsFolder key="SLB-Templates" name="SLB_Templates">

    <!-- TCP template List -->
    <vnsAbsFolder key="TCP-Template-List" name="TTLList">
        <vnsAbsFolder key="TCP-Template" name="TCP_Template_Folder">
            <vnsAbsParam key="Name" name="Name" value="TCP_1"/>
            <vnsAbsParam key="Idle-Timeout" name="idle-timeout"
value="3000"/>
        </vnsAbsFolder>
    </vnsAbsFolder>

    <!-- Virtual Server Template List -->
    <vnsAbsFolder key="Virtual-Server-Template-List" name="VSTList">
        <vnsAbsFolder key="Virtual-Server-Template" name="Virtual-Server_
Template_Folder">
            <vnsAbsParam key="Name" name="Name" value="VS_1"/>
            <vnsAbsParam key="Conn-Limit" name="Conn-Limit" value="100"/>
            <vnsAbsParam key="Conn-Limit-Reset" name="Conn-Limit-Reset"
value="1"/>
            <vnsAbsParam key="Conn-Limit-No-Logging" name="Conn-Limit-No-
Logging" value="1"/>
            <vnsAbsParam key="Conn-Rate-Limit" name="Conn-Rate-Limit"
value="50"/>
        </vnsAbsFolder>
    </vnsAbsFolder>
</vnsAbsFolder>

```

For additional help on configuring APIC and resources for reference, please visit <http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html>.

For a list of all supported configuration objects and their parameters, see Cisco ACI Parameter Reference. Note that not all parameters are supported in the current release.

Summary

As businesses seek to make the data center more agile, the application-centric automation and virtualization of both hardware and software infrastructures become increasingly important. Cisco ACI builds the critical link between business-based requirements for applications and the infrastructure that supports them. The integration of A10 Networks line of Application Delivery Controllers with Cisco ACI and Cisco APIC provides enterprises with rich application delivery and security capabilities in a shared infrastructure. This integration can be leveraged to dynamically provision L4-L7 application services, and to ensure that SLAs and security requirements are consistently met. The A10 ADC platform can help customers accelerate, secure and optimize the performance of their data center applications and networks. The comprehensive joint solution can effectively transform the data center by enabling automatic provisioning of application delivery and security services with a significant increase in provisioning speed, business agility and reduced costs.

Appendix

XML Reference Guide

1. Creating a tenant.

```
<polUni>
  <!-- Create a Tenant: Tenant_Demo -->
  <fvTenant dn="uni/tn-Tenant_Demo" name="Tenant_Demo">
    <!-- Create a Network: Demo_Network -->
    <fvCtx name="Demo_Network"/>
    <!-- Create Bridge Domain: DemoBD_External -->
    <fvBD name="DemoBD_External">
      <fvRsCtx tnFvCtxName="Demo_Network" />
      <!-- Assign Subnet -->
      <fvSubnet ip="10.10.10.1/24">
        </fvSubnet>
      </fvBD>
    <fvBD name="DemoBD_Internal">
      <fvRsCtx tnFvCtxName="Demo_Network" />
      <fvSubnet ip="20.20.20.1/24">
        </fvSubnet>
      </fvBD>
    </fvTenant>
  </polUni>
```

2. Creating an application profile with EPGs, defining the consumer and provider endpoints.

```
<polUni>
  <fvTenant dn="uni/tn-Tenant_Demo" name="Tenant_Demo">
    <!-- Application Profile -->
    <fvAp dn="uni/tn-Tenant_Demo/ap-AppProfile_Demo" name="AppProfile_Demo">
      <!-- EPG 1 -->
      <fvAEPg dn="uni/tn-Tenant_Demo/ap-AppProfile_Demo/epg-web"
name="web">
        <fvRsBd tnFvBDName="DemoBD_External" />
        <fvRsDomAtt tDn="uni/vmmp-VMware/dom-ACI" />
        <fvRsCons tnVzBrCPName="WebContract">
          </fvRsCons>
        </fvAEPg>
      <!-- EPG 2 -->
      <fvAEPg dn="uni/tn-Tenant_Demo/ap-AppProfile_Demo/epg-app"
name="app">
        <fvRsBd tnFvBDName="DemoBD_Internal" />
        <fvRsDomAtt tDn="uni/vmmp-VMware/dom-ACI" />
        <fvRsProv tnVzBrCPName="WebContract"></fvRsProv>
        </fvAEPg>
      </fvAp>
    </fvTenant>
  </polUni>
```

3. Creating a device cluster.

```

<polUni>
  <fvTenant dn="uni/tn-Tenant_Demo" name="Tenant_Demo">
    <!-- Create Device Cluster -->
    <vnsLDevVip name="ADC" devtype="VIRTUAL">
      <vnsRsMDevAtt tDn="uni/infra/mDev-A10-Thunder-4.0.0.5"/>
      <vnsRsALDevToDomP tDn="uni/vmmp-VMware/dom-ACI"/>
      <vnsCMgmt name="devMgmt"
        host="10.100.12.92"
        port="443"/>
      <vnsCCred name="username"
        value="admin"/>
      <vnsCCredSecret name="password"
        value="a10"/>
    </vnsLDevVip>

    <!-- Create a Device Cluster Selection Policy -->
    <vnsLDevCtx ctrctNameOrLbl="WebContract" graphNameOrLbl="WebGraph"
nodeNameOrLbl="vThunder">
      <vnsRsLDevCtxToLDev tDn="uni/tn-Tenant_Demo/lDevVip-ADC"/>
      <vnsLIfCtx connNameOrLbl="internal">
        <vnsRsLIfCtxToLIf tDn="uni/tn-Tenant_Demo/lDevVip-ADC/lIf-
internal"/>
      </vnsLIfCtx>
      <vnsLIfCtx connNameOrLbl="external">
        <vnsRsLIfCtxToLIf tDn="uni/tn-Tenant_Demo/lDevVip-ADC/lIf-
external"/>
      </vnsLIfCtx>
    </vnsLDevCtx>
  </fvTenant>
</polUni>
<polUni>
  <fvTenant dn="uni/tn-Tenant_Demo" name="Tenant_Demo">
    <vnsLDevVip name="ADC">
      <!-- Create a concrete device -->
      <vnsCDev name="vThunder" vcenterName="vcenter1"
vmName="A10virtual">
        <!-- Create concrete interface. Note: the interface should be
e-[ETH]. ETH is the eth number which you'd like to use. Ex. you'd like to use
Physical Thunder interface 7, then the CIf name should be "e7" -->
        <vnsCIf name="e1" vnicName="Network adapter 2"/>
        <vnsCIf name="e2" vnicName="Network adapter 3"/>
        <vnsCMgmt name="devMgmt" host="10.100.12.92" port="443" />
        <vnsCCred name="username" value="admin" />
        <vnsCCredSecret name="password" value="a10" />
        <!-- Optional: To config Thunder Device hostname -->
        <vnsDevFolder key="hostname" name="Hostname">
          <vnsDevParam key="Value" name="Value" value="dcheng-APIC"
/>
        </vnsDevFolder>
      </vnsCDev>
    </vnsLDevVip>
  </fvTenant>
</polUni>

```

4. Creating a service graph.

```

<polUni>
  <fvTenant name="Tenant_Demo">
    <vnsAbsGraph name="WebGraph">

      <vnsAbsTermNodeCon name="Consumer">
        <vnsAbsTermConn name="Consumer">
          </vnsAbsTermConn>
        </vnsAbsTermNodeCon>

      <vnsAbsTermNodeProv name="Provider">
        <vnsAbsTermConn name="Provider">
          </vnsAbsTermConn>
        </vnsAbsTermNodeProv>

      <vnsAbsConnection name="CON1">
        <vnsRsAbsConnectionConns tDn="uni/tn-Tenant_Demo/AbsGraph-
WebGraph/AbsTermNodeCon-Consumer/AbsTConn"/>
        <vnsRsAbsConnectionConns tDn="uni/tn-Tenant_Demo/AbsGraph-
WebGraph/AbsNode-vThunder/AbsFConn-external"/>
      </vnsAbsConnection>

      <vnsAbsConnection name="CON2">
        <vnsRsAbsConnectionConns tDn="uni/tn-Tenant_Demo/AbsGraph-
WebGraph/AbsTermNodeProv-Provider/AbsTConn"/>
        <vnsRsAbsConnectionConns tDn="uni/tn-Tenant_Demo/AbsGraph-
WebGraph/AbsNode-vThunder/AbsFConn-internal"/>
      </vnsAbsConnection>

      <vnsAbsNode name="vThunder">
        <vnsRsDefaultScopeToTerm tDn="uni/tn-Tenant_Demo/AbsGraph-
WebGraph/AbsTermNodeProv-Provider/outtmnl"/>

        <vnsAbsFuncConn name="external">
          <vnsRsMConnAtt tDn="uni/infra/mDev-A10-Thunder-4.0.0.5/
mFunc-ADC/mConn-external"/>
        </vnsAbsFuncConn>

        <vnsAbsFuncConn name="internal">
          <vnsRsMConnAtt tDn="uni/infra/mDev-A10-Thunder-4.0.0.5/
mFunc-ADC/mConn-internal"/>
        </vnsAbsFuncConn>

      <!-- Configure Device Configuration -->
      <vnsAbsDevCfg>
        <vnsAbsFolder key="Interface" name="extInterface">
          <vnsAbsFolder key="InterfaceConfig"
name="externalIfCfg">
            <vnsAbsParam key="IPv4_Address" name="ipv4_
internal_addr" value="10.10.10.240"/>
            <vnsAbsParam key="IPv4_Netmask" name="ipv4_

```

```

internal_mask" value="255.255.255.0"/>
    </vnsAbsFolder>
  </vnsAbsFolder>

  <vnsAbsFolder key="Interface" name="intInterface">
    <vnsAbsFolder key="InterfaceConfig"
name="internalIfCfg">
      <vnsAbsParam key="IPv4_Address" name="ipv4_
external_addr" value="20.20.20.240"/>
      <vnsAbsParam key="IPv4_Netmask" name="ipv4_
external_mask" value="255.255.255.0"/>
      </vnsAbsFolder>
    </vnsAbsFolder>
  </vnsAbsDevCfg>

  <!-- Configure Function Configuration -->
  <vnsAbsFuncCfg>
    <vnsAbsFolder key="ConfigRelFolder-ExIntf" name="ExtConfig">
      <vnsAbsCfgRel key="ExIntfConfigRel" name="ExtConfigrel"
targetName="extInterface"/>
      <vnsRsCfgToConn tDn="uni/tn-Tenant_Demo/AbsGraph-
WebGraph/AbsNode-vThunder/AbsFConn-external"/>
    </vnsAbsFolder>

    <vnsAbsFolder key="ConfigRelFolder-InIntf" name="IntConfig">
      <vnsAbsCfgRel key="InIntfConfigRel" name="InConfigrel"
targetName="intInterface"/>
      <vnsRsCfgToConn tDn="uni/tn-Tenant_Demo/AbsGraph-
WebGraph/AbsNode-vThunder/AbsFConn-internal"/>
    </vnsAbsFolder>

  </vnsAbsNode>

</vnsAbsGraph>
</fvTenant>
</polUni>

```

5. Creating a security contract.

```

<polUni>
  <fvTenant dn="uni/tn-Tenant_Demo" name="Tenant_Demo">
    <!-- Create WebContract -->
    <vzBrCP name="WebContract">
      <vzSubj name="http" provMatchT="All" consMatchT="All">
        <vzRsSubjFiltAtt tnVzFilterName="default"/>
      </vzSubj>
    </vzBrCP>
  </fvTenant>
</polUni>

```

6. Creating a device cluster policy.

```

<!-- Create a Device Cluster Selection Policy -->
  <vnsLDevCtx ctrctNameOrLbl="WebContract" graphNameOrLbl="WebGraph"
nodeNameOrLbl="vThunder">
    <vnsRsLDevCtxToLDev tDn="uni/tn-Tenant_Demo/lDevVip-ADC"/>
    <vnsLIfCtx connNameOrLbl="internal">
        <vnsRsLIfCtxToLIf tDn="uni/tn-Tenant_Demo/lDevVip-ADC/lIf-
internal"/>
    </vnsLIfCtx>
    <vnsLIfCtx connNameOrLbl="external">
        <vnsRsLIfCtxToLIf tDn="uni/tn-Tenant_Demo/lDevVip-ADC/lIf-
external"/>
    </vnsLIfCtx>
  </vnsLDevCtx>
</fvTenant>
</polUni>
<polUni>
  <fvTenant dn="uni/tn-Tenant_Demo" name="Tenant_Demo">
    <vnsLDevVip name="ADC">

```

About A10 Networks

A10 Networks is a leader in application networking, providing a range of high-performance application networking solutions that help organizations ensure that their data center applications and networks remain highly available, accelerated and secure. Founded in 2004, A10 Networks is based in San Jose, California, and serves customers globally with offices worldwide. For more information, visit: www.a10networks.com

Corporate Headquarters

A10 Networks, Inc
 3 West Plumeria Ave.
 San Jose, CA 95134 USA
 Tel: +1 408 325-8668
 Fax: +1 408 325-8666
www.a10networks.com

Part Number: A10-DG-16143-EN-01
 Dec 2014

Worldwide Offices

North America
sales@a10networks.com
Europe
emea_sales@a10networks.com
South America
latam_sales@a10networks.com
Japan
jinfo@a10networks.com
China
china_sales@a10networks.com

Taiwan
taiwan@a10networks.com
Korea
korea@a10networks.com
Hong Kong
HongKong@a10networks.com
South Asia
SouthAsia@a10networks.com
Australia/New Zealand
anz_sales@a10networks.com

To learn more about the A10Thunder Application Service Gateways and how it can enhance your business, contact A10 Networks at: www.a10networks.com/contact or call to talk to an A10 sales representative.