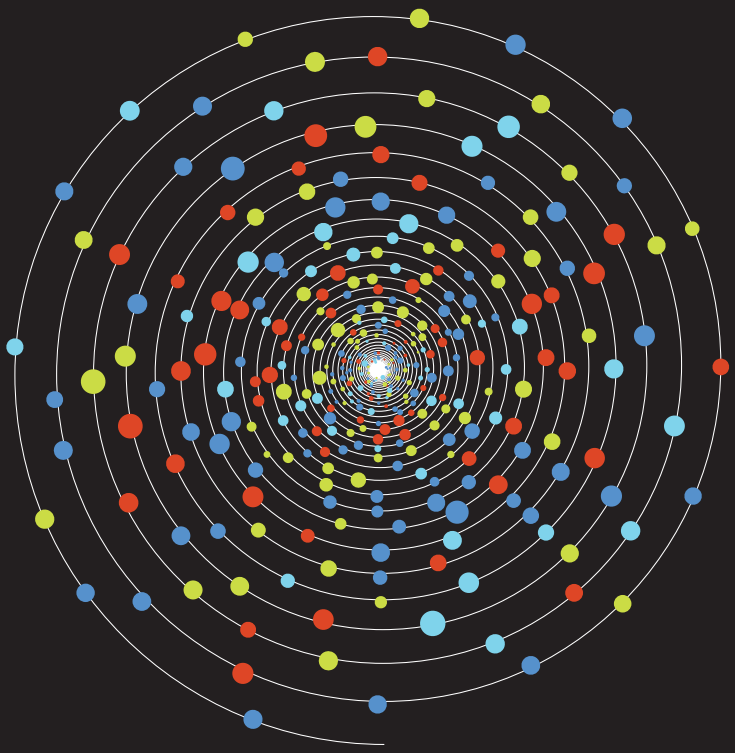
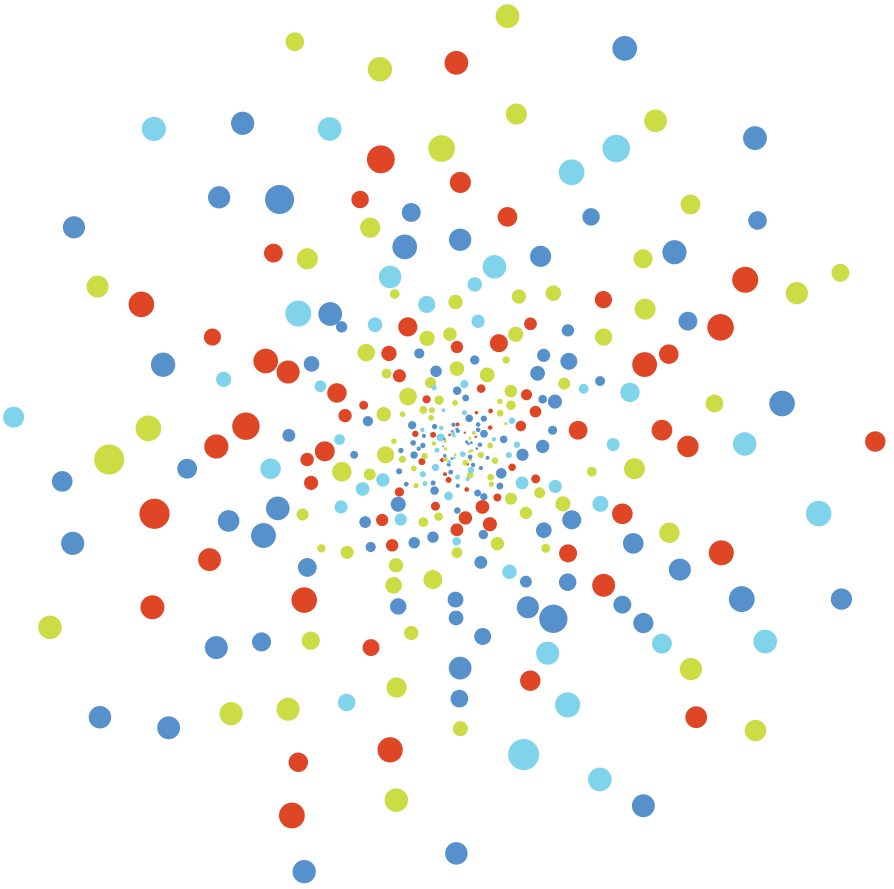


Operating Cisco Application Centric Infrastructure

Alejandra Sanchez, Andres Vega, Arvind Chari, Carly Stoughton, Christopher Stokes, Gabriel Fontenot, Jonathan Cornell, Ken Fee, Kevin Corbin, Lauren Malhoit, Loy Evans, Lucien Avramov, Paul Lesiak, Steven Lym, Rafael Muller, Robert Burns

Operating Cisco Application Centric Infrastructure



Copyright

Operating Cisco Application Centric Infrastructure

Alejandra Sanchez, Andres Vega, Arvind Chari, Carly Stoughton, Christopher Stokes, Gabriel Fontenot, Jonathan Cornell, Ken Fee, Kevin Corbin, Lauren Malhoit, Loy Evans, Lucien Avramov, Paul Lesiak, Rafael Muller, Robert Burns. Steven Lym

Privately published by Cisco Systems, Inc.

Warning and Disclaimer

This book is designed to provide information about Cisco ACI. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this document.

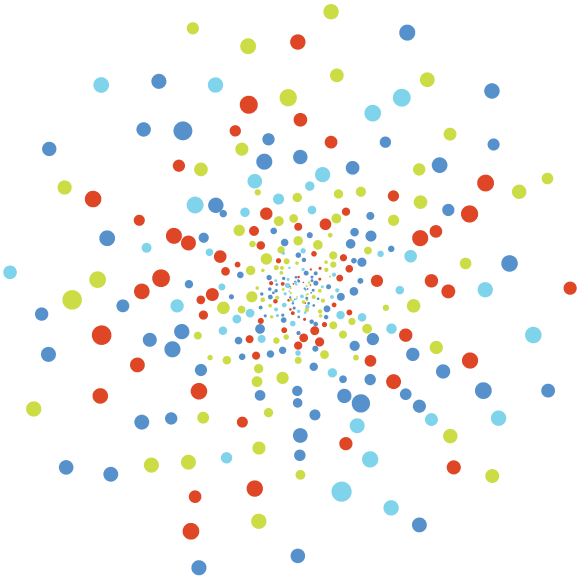
The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Feedback Information

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at ops-booksprint@cisco.com.

Contents

Prologue



Abstract

Cisco's Application Centric Infrastructure (ACI) provides powerful new ways to dynamically manage infrastructure in the modern world of IT automation and DevOps. Having the tools to change how infrastructure is built is one thing, but being able to effectively operate the infrastructure beyond the Day Zero build activities is crucial to long term effectiveness and efficiency. To effectively harness the power of ACI, organizations will need to understand how to incorporate ACI into their daily operations. This book examines some of the common operational activities that IT teams use to provide continued infrastructure operations and gives the reader exposure to the tools, methodologies, and processes that can be employed to support day 1+ operations within an ACI-based fabric.

Authors

This book represents a joint intense collaborative effort over the course of a week, with participation of a number of Cisco functional organizations including Cisco Advanced Services, Technical Services, Product Marketing, Solution Validation Testing, IT, and Sales.

Authors

Alejandra Sanchez - Customer Support Engineer, Technical Services
Andres Vega - Customer Support Engineer, Technical Services
Arvind Chari - Solutions Architect, Advanced Services
Carly Stoughton - Technical Marketing Engineer, INSBU
Christopher Stokes - Cisco IT Network Consulting Engineer
Gabriel Fontenot - Network Consulting Engineer, Advanced Services
Jonathan Cornell - Systems Engineer, Sales
Ken Fee - Consulting Architect, Business Technology Architects
Kevin Corbin - Technical Solutions Architect, Systems Engineering
Lauren Malhoit - Marketing Consultant, INSBU
Loy Evans - Consulting System Engineer, Sales
Lucien Avramov - Technical Marketing Engineer, INSBU
Paul Lesiak - Solutions Architect, Advanced Services
Rafael Muller - Technical Leader, Solution Validation Services
Robert Burns - Technical Leader, Technical Services
Steven Lym - Technical Writer, INSBU

Book Sprint Facilitation

Barbara Rühling
Faith Bosworth

Illustrations

Henrik van Leuwen

Book Production

Julien Taquet

Clean Up

Raewyn Whyte

Book Writing Methodology

The Book Sprint (booksprints.net) methodology was used for writing this book. The Book Sprint methodology is an innovative new style of cooperative and collaborative book production. Book Sprints are strongly facilitated and leverage team-oriented inspiration and motivation to rapidly deliver large amounts of well-authored and reviewed content, and incorporate it into a complete narrative in a short amount of time. By leveraging the input of many experts, the complete book was written in only five days, but involved hundreds of authoring person-hours, and included thousands of experienced engineering hours, allowing for extremely high quality in a very short production time period.

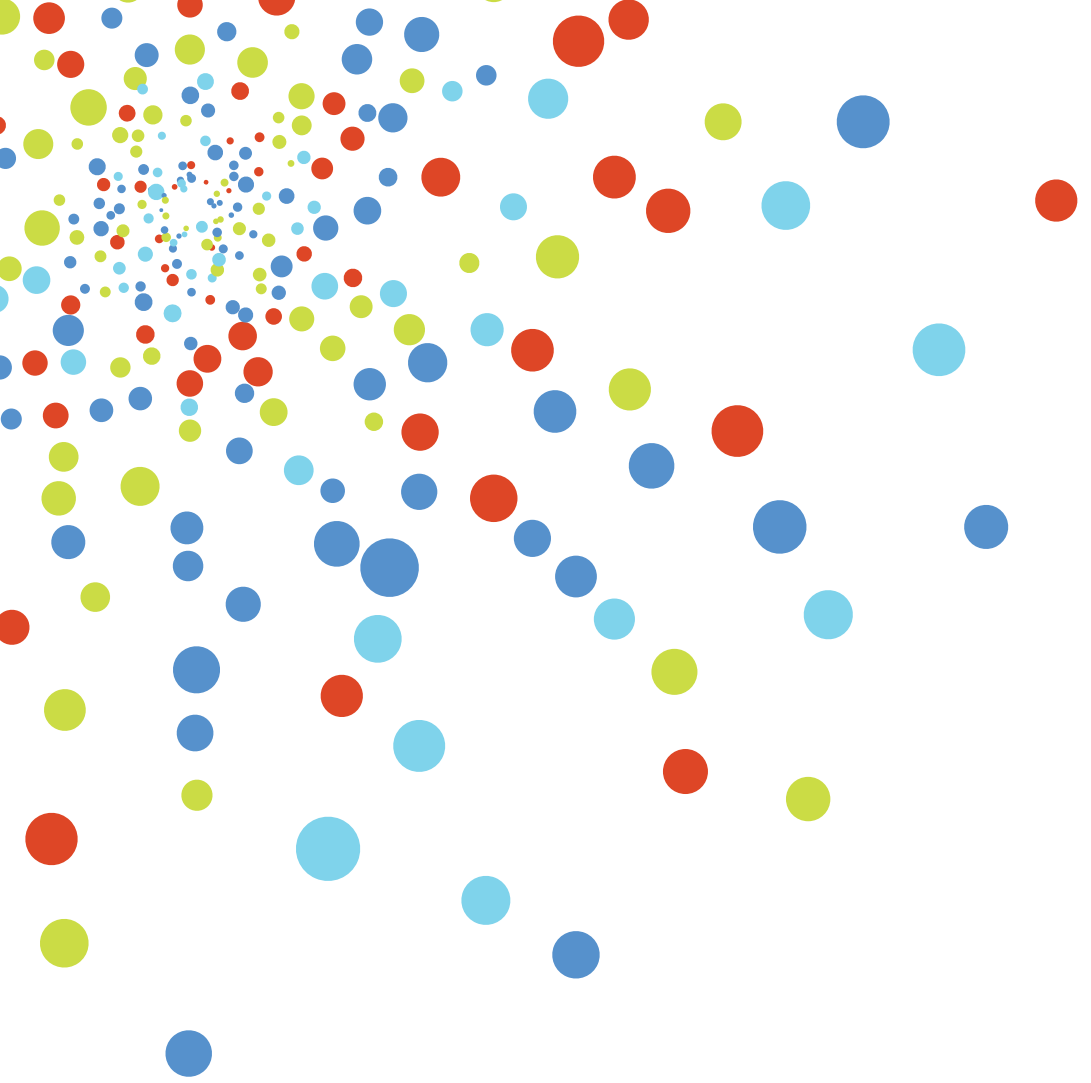
Hardware and Software Included in the Book

The Cisco Application Centric Infrastructure (ACI) combines hardware, software, and ASIC innovations into an integrated systems approach and provides a common management framework for network, application, security, and virtualization.

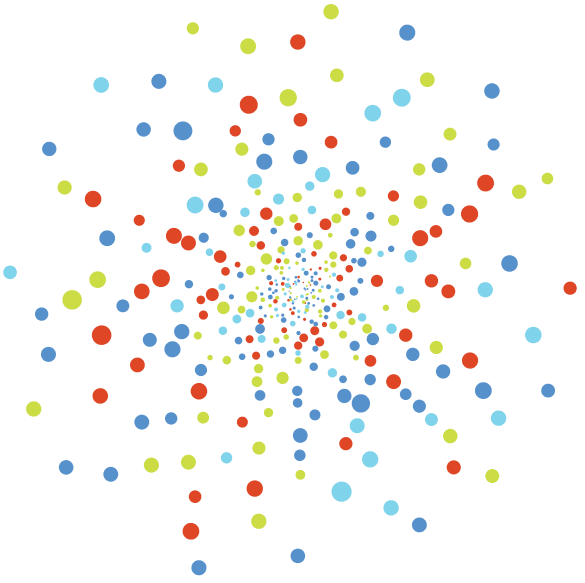
For the purpose of writing this book, the following hardware devices were used:

- Cisco Application Policy Infrastructure Controller (APIC)
- ACI Spine switches, including Cisco Nexus 9508 and 9336PQ
- ACI Leaf switches, including Cisco Nexus 9396PX, 9396TX, and 93128TX
- Cisco Application Virtual Switch (AVS)
- Cisco UCS B and C series servers
- Several models of switches and routers, including Cisco Nexus 5000 switches and Cisco Integrated Services Routers (ISR)
- A variety of hypervisors, including KVM, Microsoft Hyper-V, and VMware vSphere
- IXIA IxVM product family traffic generators

This book was written based on the Cisco ACI 1.1 software release.



Introduction



The Story of ACME Inc.

ACME Inc. is a multinational corporation that specializes in manufacturing, sales, and distribution of a diverse product portfolio, including rocket-powered roller skates, jet-propelled unicycles, and various explosive materials. These product groups operate as separate business groups within the company, and have previously maintained separate infrastructure and applications. They have largely focused on retail routes to market, but have recently decided to pursue a more direct-to-consumer business model due to intense pressure from new competitors who have dominated the online sales channels. In an effort to be more competitive, ACME has undertaken a project to build a mobile application platform to support ordering and logistics for product delivery to their customers for their entire portfolio.

Traditionally, ACME business units have leveraged third party software companies and commercially available software to meet their IT demands, but would like to create a more intimate relationship with their consumers and be able to take feedback on the platform directly from those users, while incorporating an ongoing improvement cycle so they can react to changing market dynamics in a more nimble fashion. Where they have used custom software in the past, they have leveraged a traditional infrastructure and software model that does not allow them to keep up with the changing requirements, and therefore ACME is looking for a new approach to both application and infrastructure lifecycle management. The application developers have been looking at new application development trends such as Continuous Delivery and Continuous Integration, and the new application platform is to be developed in this manner. To support this, the infrastructure components need to be capable of mapping to these new paradigms in a way that is not possible using traditional concepts.

One of the largest challenges ACME has historically faced is that operations and infrastructure has been an afterthought to product development. This has led to several situations where application deployments have meant long weekend hours for all of the teams, caused customer-impacting outages, and taken longer to accomplish than the business leaders would have liked. For this reason, ACME Inc. has decided to change by creating an environment where infrastructure artifacts are treated as part of the application, can be checked into version control, can be tested alongside the actual application, and can continually improve.

While ACME is intensely focused on delivering the new application platform in a timely manner, ACME is also interested in creating a foundation on which it can grow to deliver a common pool of infrastructure that is shared across all business groups and operated in a multi-tenant fashion to increase efficiency.

At an executive briefing, John Chambers, the CEO of Cisco Systems, told ACME:
"The world is changing. Every company is a technology company, and if you don't adapt, you'll get left behind."

As evidenced by the success of cloud platforms, such as Amazon Web Services (AWS) and Openstack, consumption models of technology delivery have the ability to adapt technology more quickly to rapid business requirements changes. This is the type of consumption that ACME Inc.'s business owners need. Control of operations is what operations groups are focused on, but control can be a barrier to a pure consumption model. Unless companies make investments in technologies that allow for consumption of automated components, the only other way to scale is by breaking the human level component, and few people would really choose to work for that type of company.

After analyzing current offers from various technology vendors, ACME Inc. selected Cisco Application Centric Infrastructure (ACI). The ability to abstract all physical and virtual infrastructure configuration into a single configuration that is consistent across dev, test, and prod environments, as well as portable across the various data center locations currently maintained by ACME, is highly desirable. ACI has been built from the ground up to change the substructure used to build network devices and protocols. Innovation at this level will provide more opportunities for expanding the tools with which users interact. This is where the fulcrum will tilt in the favor of IT and infrastructure being more dynamic, thus allowing IT to operate and manage at the speed of business. However, with a change of this nature comes fear, uncertainty, and doubt. This book will attempt to bring some level of comfort and familiarity with operations activities within an ACI fabric.

While ACME Inc. is a fictitious company, this is the true story of every company, and just important this is the story of the employees of those companies. Workers in the IT industry need to adapt to keep up with the rapid change of the business. However, this runs contrary to how most operations groups exist in the relationship between business and technology. Most IT operations groups invest a lot of time in the tools needed to deliver services today and there is an organic resistance to re-investing. The thought is, "Why fix what is already working?"

The Why, Who, What, When and How

Within ACI, ACME is looking to simplify how it operates infrastructure, but recognizes that this initiative, this application, and this infrastructure are new to ACME Inc. ACME must address fundamental questions including Who manages What, and How they go about their tasks. When different groups perform regular operations, and Where they go to perform these operations, are also considerations, but more tactical and point-in-time-relevant. This section discusses the relevant aspects of these monikers as it relates to ACI fabric operations and how a company such as ACME Inc. can divide the workload.

Why

"Why" is the most important aspect of what should be considered in operationalizing an ACI fabric. In the case of ACME Inc. the key success criteria is to streamline processes and procedures related to the deployment of infrastructure required to support the application initiatives. To achieve the desired result, a high degree of automation is required. Automation adds speed to repetitive tasks and eliminates errors or missed steps. Initially, automation can be a scary proposition for some of the key stakeholders, as it could be looked at as a threat to their own job. Quite the opposite, automation is about making work more enjoyable for all team members, allowing them the freedom to innovate and add value, while removing mundane, repetitive tasks. Looking at why an automated fabric is beneficial to an organization is important for setting expectations of return on investment. Also, looking at why an operational practice is done a specific way can help with framing the tools and processes that are employed.

Who

As with most organizations, ACME Inc. traditionally had different types of stakeholders involved in making any IT initiative successful, and each has a specific element of the infrastructure in which they have specific expertise and about which they care most. In any IT organization, these groups can have distinct organizational boundaries, but more likely the boundaries are blurred to some degree. Listed below are some characteristics of these groups, but keep in mind that some of these characteristics might be combined. At the macro level, the fact that these different organizations exist should

not be evident to the end-user. Instead, the entire organization should be seen as one team with a common goal of delivering value to their organization.

ACME's Development and Application Team is focused on the software and applications the company uses internally and the software that it delivers to its customers. The Application part of the team contains application owners and subject matter experts that ensure other business units are able to do their jobs by utilizing the business applications available. The Development part of the team will be writing the mobile application software platform. Both parts of this team will need to work closely with the other teams in this section to enable the best design, performance, and availability of applications for the end users.

ACME's Network Team is primarily focused on creating and managing networking constructs to forward packets at Layer 2 (MAC/switching) and Layer 3 (IP routing). The team is challenged with juggling application requirements, managing SLA, and assisting in the enforcement of information security, all while maintaining high levels of availability. What the team needs to know is how to configure the networking constructs, how to tie Layer 2 to Layer 3, how to verify forwarding, and how to troubleshoot network forwarding aspects in the fabric. With ACI, the team is the most interested in decoupling overloaded network constructs and returning to the specific network problems that the team was intended to solve, while allowing other groups to leverage their specific expertise to manipulate security and application level policies. The team is also interested in allowing more transparency in the performance of the network forwarding, and the team is making key metrics available on demand in a self-service capacity.

ACME's Storage Team is primarily focused on delivery of data storage resources to the organization. The storage team is concerned with protecting the data in terms of availability, as well as making sure that sensitive data is secure. The storage team has been very successful in maintaining very tight SLAs and has traditionally managed separate infrastructure for storage access. The capabilities provided by the ACI fabric allow them to confidently deploy newer IP-based storage and clustering technologies. The team is also very interested in being able to see how the storage access is performing and would like to be notified in the event of contention. The team typically has some specific requirements around QoS, multi-pathing, and so on. Historically, the team had to worry about delivering a storage fabric in addition to managing storage devices themselves. ACI will provide the storage team with the visibility they will require. These capabilities are primarily discussed in the monitoring sections.

The Compute and Virtualization Team at ACME Inc. is wrapping up a major initiative to virtualize the server farms that it is responsible for maintaining. The team also recently employed new configuration management tools to account for new workloads that fell outside of the virtualization effort to get similar agility for bare metal servers that the team gained from its virtualization efforts. This is timely as the application rollout will have both virtualized and non-virtualized workloads. Additionally, the application developers are increasingly interested in leveraging Linux container technologies to allow for even greater application portability. The Compute and Virtualization teams are interested in ACI for its ability to provide common access to physical and virtual servers, allowing the team to publish endpoint groups to virtualization clusters from a centralized place across multiple hypervisors. These capabilities are discussed further in the Fabric Connectivity chapter.

The Information Security Team at ACME Inc. has traditionally been engaged late in an application deployment process, and has been responsible for performing vulnerability assessment and data classification efforts. With the current project, the new application will be storing sensitive customer information, including credit card numbers. Due to the sensitivity of this information and the security aspects of the ACI fabric, the Information Security Team is able to provide input earlier in the process and avoid re-doing work because of security or compliance issues. The Information Security Team is interested in the operational aspects of the ACI security model as it relates to the following capabilities: tenancy, Role Based Access Control (RBAC), monitoring, and Layer 4 to Layer 7 services.

What

The aspect of "what" can be looked at in many different ways, but the main concept in the context of this book is what tools are used to manage operations of an ACI fabric. In a traditional network, you have some traditional tools, such as CLI and SNMP, to manage network operations, and these tools integrate into management platforms and configuration and management processes.

In ACI there are some elements of the traditional tools, but the fabric management is rooted in an abstracted object model that provides a more flexible base. With this base, the operator of the fabric can choose from multiple modes of management, such as GUI, CLI, API integration, programming, scripting, or some combination of these. How a tool is selected in ACI will often be a product of what is being done and the aspects of how the tool is used. For example, if an operations staff is trying to gather a bunch of information across a number of interfaces and switches or is managing the configura-

tion of many different objects at once, scripting might be more efficient, whereas simple dashboard monitoring might be more suited to a GUI.

When

"When" refers to when the teams listed above are involved in the planning. It is a good idea to involve the different teams early when building policies and processes for how the fabric is implemented and then managed. The collaborative nature of ACI allows for a high degree of parallelization of work flow. This is a key difference between ACI and traditional processes that were very serial in nature, resulting in a longer deployment time for applications and a higher mean-time to resolution when issues arise.

How

"How" answers the following basic questions:

- How does a networking person go about configuring the network forwarding?
- How does the compute team get information from the infrastructure to make optimal workload placement decisions?
- How do the application team track performance and usage metrics?
- How does a storage team track the access to storage subsystems and ensure that it is performant?

When "how" involves making a change to the configuration of an environment, an important consideration is change control. Change control is a fact of life in the mission-critical environments that ACI has been designed to support. The ACI policy model has been designed to reduce the overall size of a fault domain and provide a mechanism for incremental change. There are mechanisms for backup and restore that will be discussed in follow-on chapters. We will also discuss the model and which objects affect the tenants and the fabric as a whole.

An evaluation of current change control and continuous integration/delivery strategies is warranted as operational procedures evolve. Throughout this book we will highlight the methods and procedures to proactively and reactively manage the fabric.

As a baseline, most organizations are implementing some kind of structured change-control methodology to mitigate business risk and enhance system availability. There are a number of change/IT management principles (Cisco Lifecycle services, FCAPS,

and ITIL) that are good guides from which to start. A common sense approach to change management and continuous integration should be a premise that is discussed early in the design and implementation cycle before handing the fabric to the operations teams for day-to-day maintenance, monitoring, and provisioning. Training operations teams on norms (a stated goal of this book) is also key. Applying change management principles based on technology from five years ago would not enable the rapid deployment of technology that ACI can deliver.

The multi-tenant and role-based access control features inherent to the ACI solution allow the isolation or drawing of a very clean box around the scope and impact of the changes that can be made. For more details, see the RBAC section of this book.

Ultimately each change must be evaluated primarily in terms of both its risk and value to the business. A way to enable a low-overhead change management process is to reduce the risk of each change and increase its value. Continuous delivery does exactly this by ensuring that releases are performed regularly from early on in the delivery process, and ensuring that delivery teams are working on the most valuable thing they could be at any given time, based on feedback from users.

In the Information Management Systems world, there are three fundamental kinds of changes:

- Emergency changes
- Normal
- Standard

Emergency changes are by definition a response to some kind of technical outage (hardware, software, infrastructure) and are performed to restore service to affected systems.

Normal changes are those that go through the regular change management process, which starts with the creation of a request for change which is then reviewed, assessed, and then either authorized or rejected, and then (assuming it is authorized) planned and implemented. In an ACI environment a normal change could apply to anything within the following components:

- Fabric Policies (fabric internal and access will be discussed in detail later)
- Configuration objects in the Common tenant that are shared with all other tenants (things that affect the entire fabric)
    - Private Networks

    - Bridge Domains
    - Subnets
- Virtual Machine Manager (VMM) integrations
- Layer 4 to Layer 7 devices
    - Device packages
    - Creation of logical devices
    - Creation of concrete devices
- Layer 2 or Layer 3 external configuration
- Attachable Entity Profile (AEP) creation
- Server or external network attachment
- Changes to currently deployed contracts and filters that would materially change the way traffic flows

Standard changes are low-risk changes that are pre-authorized. Each organization will decide the kind of standard changes that they allow, who is allowed to approve them, the criteria for a change to be considered "standard", and the process for managing them. As with normal changes, they must still be recorded and approved. In the ACI environment some examples of "standard" changes could be:

- Tenant creation
- Application profile creation
- Endpoint group (EPG) creation
- Contracts scoped at a tenant level
- Layer 4 to Layer 7 service graphs
- Domain associations for EPGs

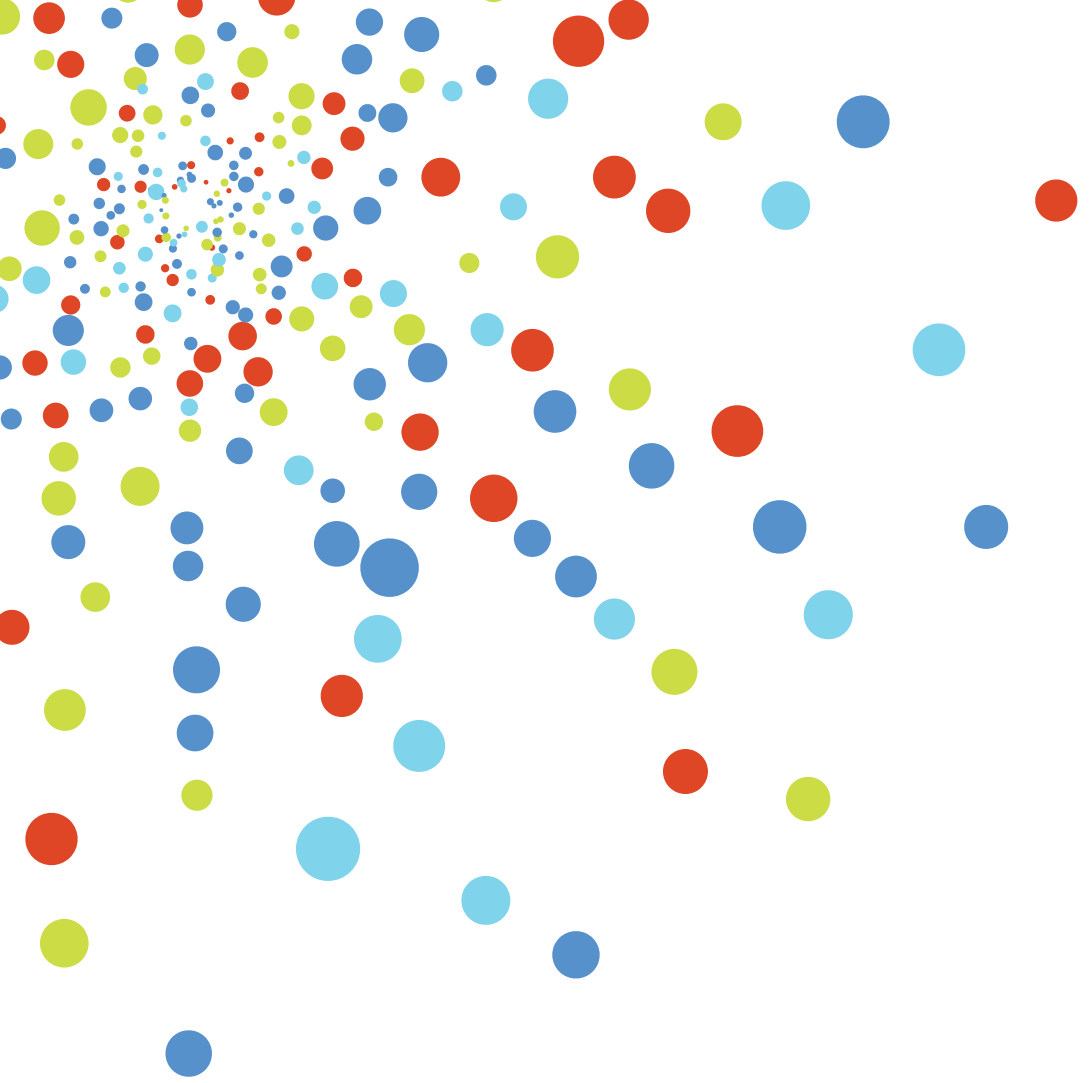
The items mentioned above are not intended to be all-inclusive, but are representative of common tasks performed day-to-day and week-to-week.

The ability to audit changes that are happening to the environment is a requirement for ACME Inc. APIC maintains an audit log for all configuration changes to the system. This is a key troubleshooting tool for "when something magically stops working". Immediate action should be to check the audit log as it will tell who made what change and when, correlating this to any faults that result from said change. This enables the change to be reverted quickly.

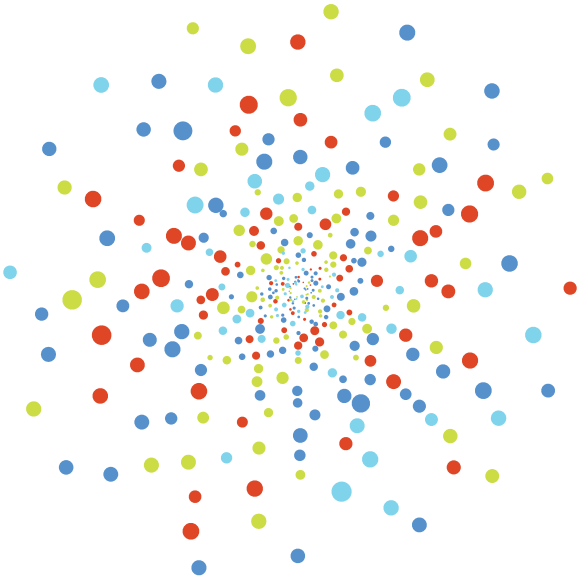
A more in-depth discussion of continuous delivery in the context of infrastructure management is outside of the scope of this book.

The remainder of this book answers these questions, providing you with a framework of how to take the concepts and procedures and apply them to similar initiatives within your organizations. The book is laid out in a specific order. However, ACI enables ACME Inc. to complete these tasks in parallel with the various stakeholders who are highlighted throughout, and this book illustrates how the stakeholders can work together in a more collaborative manner than they have in the past. While some scripting opportunities are called out throughout the book, there is a more in-depth section at the end that explains how to use the ACI API to automate most operational tasks. While organizational structures might be siloed into these teams, to provide the greatest value to the customer, user, and ultimately the business, the most important thing is for these groups to work *insieme* (together).

ACI addresses IT requirements from across the organization



Management



Section Content

- APIC Overview
- Configuring Management Protocols
  - Cisco Discovery Protocol
  - Link Layer Discovery Protocol
  - Time Synchronization and NTP
  - Out-of-Band Management NTP
  - In-Band Management NTP
  - Verify NTP Operation
    - Verifying that the NTP Policy Deployed to Each Node
  - Domain Name Services (DNS)
    - Verifying DNS Operation
- Role-Based Access Control
  - Multiple Tenant Support
  - User Roles
  - Security Domains
  - Creation of a Security Domain
  - Adding Users
    - Remote Authentication
- Import and Export Policies
  - Configuration Export (Backup)
  - Add a Remote Location (SCP)
  - Create a One Time Export Policy
  - Verify Export Policy was Successful
  - Extract and View Configuration Files
  - Configuration Import (Restore/Merge)

APIC Overview

There are a number of fundamental differences between the operations of traditional networking hardware and an ACI fabric. These differences serve to simplify the management greatly, reduce the number of touchpoints, and decouple the switching hardware from the desired configuration intent. These changes include:

- Single point of management controller-based architecture
- Stateless hardware
- Desired state-driven eventual consistency model

The single point of management within the ACI architecture is known as the Application Policy Infrastructure Controller (APIC). This controller provides access to all configuration, management, monitoring, and health functions. Having a centralized controller with an application programming interface (API) means that all functions configured or accessed through the fabric can be uniformly approached through a graphical user interface (GUI), command line interface (CLI), and application programming interface (API), with no risk of inconsistency between the various data interfaces. This results in a clean and predictable transition between the interfaces. The underlying interface for all access methods is provided through a REST-based API, which modifies the contents of a synchronized database that is replicated across APICs in a cluster and provides an abstraction layer between all of the interfaces.

This controller-based architecture also makes possible a stateless configuration model that decouples the hardware from the configuration running on it. This translates to an APIC cluster that manages individual fabric nodes of leaf and spine switches that derive their identity from what the controller defines as being the desired intent, and not from the serial number of the chassis, nor from a configuration file residing on the devices. Each node receives a unique node identifier, which allows for the device to download the correct configuration attributes from the controller. The device can also be substituted in a stateless fashion, meaning that hardware swaps can be faster, topology changes are less impactful, and network management is simplified.

The desired state model for configuration further complements these concepts of controller-based management and statelessness by taking advantage of a concept known as declarative control-based management, based on a concept known as the promise

theory. Declarative control dictates that each object is asked to achieve a desired state and makes a "promise" to reach this state without being told precisely how to do so. This stands in contrast with the traditional model of imperative control, where each managed element must be told precisely what to do, be told how to do it, and take into account the specific situational aspects that will impact its ability to get from its current state to the configured state. A system based on declarative control is able to scale much more efficiently than an imperative-based system, since each entity within the domain is responsible for knowing its current state and the steps required to get to the desired state, dictated by the managing controller.

Configuring Management Protocols

For the optimization of ACME's Cisco Application Centric Infrastructure (ACI) fabric, the network team created the following management protocol policies:

- Cisco Discovery Protocol (CDP) - These policies are primarily consumed by the network team, as CDP is the standard for their existing network equipment.
- Link Layer Discovery Protocol (LLDP) - Discovery protocols are no longer limited to just network devices. LLDP is a standards-based protocol for discovering topology relationships. ACME is deploying LLDP on servers, Layer 4 to Layer 7 services devices, and storage arrays. Therefore, these policies will be available for consumption by all of the teams.
- Network Time Protocol (NTP) - Maintaining accurate time across the application logs and infrastructure components is extremely important for being able to correlate events. ACME has existing NTP servers and will leverage them for maintaining time in their ACI fabric deployment. With ACI, maintaining accurate time is of increased importance as accurate time is a prerequisite for features such as atomic counters.
- Domain Name Services (DNS) - providing name resolution to fabric components consistent with the application and server teams. In addition to be able to resolve names within the fabric, the important ACI components are also registered with ACME's DNS server so teams can easily access them in their management tasks.

Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is a valuable source of information in troubleshooting physical and data-link layer issues. In the course of ACI operations, there may be times when you must provision a particular host-facing port manually with a CDP on or off policy. By default, the ACI fabric provides a default policy where CDP is disabled. As a recommended practice, manually create a “CDP-ENABLED" policy with CDP enabled, as well as a “CDP-DISABLED” policy with CDP disabled that can be referenced throughout all interface policy configurations. It is also important to note that CDP might be required for certain configurations, such as when to create switch profiles to connect to Cisco UCS B Series servers.

To create CDP policies:

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > CDP Interface**.
3. In the Work pane, choose **Actions > Create CDP Interface Policy**.
4. In the **Create CDP Interface Policy** dialog box, perform the following actions:
   a. In the **Name** field enter the name of the policy, such as "CDP-ENABLED".
   b. For the **Admin State** radio buttons, click **Enabled**.
   c. Click **Submit**.
5. In the Work pane, choose **Actions > Create CDP Interface Policy**.
6. In the **Create CDP Interface Policy** dialog box, perform the following actions:
   a. In the **Name** field enter the name of the policy, such as "CDP-DISABLED".
   b. For the **Admin State** radio buttons, click **Disabled**.
   c. Click **Submit**.

Your CDP policy is now ready for deployment to the ACI fabric interfaces.

Link Layer Discovery Protocol

Link Layer Discovery Protocol (LLDP) is a valuable source of information for troubleshooting physical and data-link layer issues. In the course of ACI operations, by default, the LLDP feature is enabled on the fabric. There might be times in the operation of the ACI fabric in which you must manually adjust the LLDP protocol to conform with interoperability requirements of end-host devices. For example, when connecting to Cisco Unified Computing System (UCS) Blade servers, you must disable LLDP.

Cisco recommends that you pre-provision LLDP enable and disable protocol policies to make future interface policy deployment decisions streamlined and easily configured.

To create an LLDP policy:

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > LLDP Interface**.
3. In the Work pane, choose **Actions > Create LLDP Interface Policy**.
4. In the **Create LLDP Interface Policy** dialog box, perform the following actions:

a. In the **Name** field, enter "LLDP-TX-ON-RX-ON".
b. For the **Receive State** radio buttons, click **Enabled**.
c. For the **Transmit State** radio buttons, click **Enabled**.
d. Click **Submit**.

5 In the Work pane, choose **Actions > Create LLDP Interface Policy**.

6 In the **Create LLDP Interface Policy** dialog box, perform the following actions:
a. In the **Name** field, enter "LLDP-TX-ON-RX-OFF".
b. For the **Receive State** radio buttons, click **Disabled**.
c. For the **Transmit State** radio buttons, click **Enabled**.
d. Click **Submit**.

7 In the Work pane, choose **Actions > Create LLDP Interface Policy**.

8 In the **Create LLDP Interface Policy** dialog box, perform the following actions:
a. In the **Name** field, enter "LLDP-TX-OFF-RX-ON".
b. For the **Receive State** radio buttons, click **Enabled**.
c. For the **Transmit State** radio buttons, click **Disabled**.
d. Click **Submit**.

9 In the **Create LLDP Interface Policy** dialog box, perform the following actions:
a. In the **Name** field, enter "LLDP-TX-OFF-RX-OFF".
b. For the **Receive State** radio buttons, click **Disabled**.
c. For the **Transmit State** radio buttons, click **Disabled**.
d. Click **Submit**.

Your LLDP policy is now ready for deployment to the ACI fabric interfaces.

Time Synchronization and NTP

Within the Cisco Application Centric Infrastructure (ACI) fabric, time synchronization is a crucial capability upon which many of the monitoring, operational, and troubleshooting tasks depend. Clock synchronization is important for proper analysis of traffic flows as well as for correlating debug and fault timestamps across multiple fabric nodes. An offset present on one or more devices can hamper the ability to properly diagnose and resolve many common operational issues. In addition, clock synchronization allows for the full utilization of the atomic counter capability that is built into the ACI upon which the application health scores depend. Nonexistent or improper configuration of time synchronization does not necessarily trigger a fault or a low health score. You

should configure time synchronization before deploying a full fabric or applications so as to enable proper usage of these features. The most widely adapted method for synchronizing a device clock is to use Network Time Protocol (NTP). For more information on atomic counters, see the Reactive Monitoring Tools chapter.

Prior to configuring NTP, consider what management IP address scheme is in place within the ACI fabric. There are two options for configuring management of all ACI nodes and Application Policy Infrastructure Controllers (APICs), in-band management and/or out-of-band management. Depending on which management option was chosen for the fabric, configuration of NTP will vary.

Another consideration in deploying time synchronization where the time source is located. The reliability of the source must be carefully considered when determining if you will use a private internal clock or an external public clock.

Out-of-Band Management NTP

When an ACI fabric is deployed with out-of-band management, each node of the fabric, inclusive of spines, leaves and all members of the APIC cluster, is managed from outside the ACI fabric. This IP reachability will be leveraged so that each node can individually query the same NTP server as a consistent clock source.

To configure NTP, a Date and Time policy must be created that references an out-of-band management endpoint group. Date and Time policies are confined to a single pod and must be deployed across all pods provisioned in the ACI fabric. As of the publishing of this document, only one pod per ACI fabric is allowed.

1. On the menu bar, choose **Fabric > Fabric Policies**.
2. In the Navigation pane, choose **Pod Policies > Policies**.
3. In the Work pane, choose **Actions > Create Date and Time Policy**.
4. In the **Create Date and Time Policy** dialog box, perform the following actions:
   a. Provide a name for the policy to easily distinguish between your environment's different NTP configurations, such as "Production-NTP".
   b. Click **Next**.
   c. Click the **+** sign to specify the NTP server information (provider) to be used.
   d. In the **Create Providers** dialog box, enter all relevant information, including the following fields: **Name**, **Description**, and **Minimum Polling Intervals**, and **Maximum Polling Intervals**.

i. If you are creating multiple providers, click the **Preferred** check box for the most reliable NTP source.
ii. In the **Management EPG** drop-down list, if the NTP server is reachable by all nodes on the fabric through out-of-band management, choose **Out-of-Band**. If you have deployed in-band management, see the "In-Band Management NTP" section.
iii. Click **OK**.

e. Repeat steps c and d for each provider that you want to create.

Your NTP policy is now ready for deployment to the ACI fabric nodes.

In-Band Management NTP

When an ACI Fabric is deployed with in-band management, consider the reachability of the NTP server from within the ACI in-band management network. In-band IP addressing used within the ACI fabric is, by definition, not reachable from anywhere outside the fabric. To leverage an NTP server external to the fabric with in-band management, you must construct a policy to enable this communication. The steps used to configure in-band management policies are identical to those used to establish an out-of-band management policy: the distinction is around how to allow the fabric to connect to the NTP server.

Once you have established a policy to allow communication, you can create the NTP policy as established in the "Out-of-Band Management NTP" section.

Verifying NTP Operation

Full verification of NTP functionality is best accomplished by leveraging both the ACI CLI and the APIC GUI. Start verifying time synchronization configuration by ensuring that all policies are configured properly inside of the APIC GUI. Policy configuration can be verified by these steps:

1 On the menu bar, choose **Fabric > Fabric Policies**.

2 In the Navigation pane, choose **Pod Policies > Policies > Date and Time > *ntp_policy* > *server_name***. The ***ntp_policy*** is the previously created policy.

3 In the Work pane, verify the details of the server.

Verifying that the NTP Policy Deployed to Each Node

To verify that the policy has been successfully deployed to each node, you should use the NX-OS Virtual Shell that is present in the APIC. To access the NX-OS Virtual Shell, open an SSH session to the out-of-band management IP interface of any of the APIC nodes. From this prompt, execute the "version" command to obtain a consolidated list of node hostnames.

1. SSH to an APIC in the fabric.
2. Press the **Tab** key twice after the entering the **attach** command to list all of the available node names:

```
admin@apic1:~> attach <Tab> <Tab>
```

3. Log in to one of the nodes using the same password that you used to access the APIC.

```
admin@apic1:~> attach node_name
```

4. View the NTP peer status:

```
leaf-1# show ntp peer-status
```

   A reachable NTP server has its IP address prefixed by an asterisk (*), and the delay is a non-zero value.

5. Repeat steps 3 and 4 for each node on the fabric.

Domain Name Services (DNS)

Setting up a DNS server allows the APIC to resolve various hostnames to IP addresses. This is useful when integrating VMM domains or other Layer 4 to Layer 7 devices and the hostname is referenced.

1. On the menu bar, choose **Fabric > Fabric Policies**.
2. In the Navigation pane, choose **Global Policies > DNS Profiles > default**.
3. In the Work pane, in the **Management EPG** drop-down list, choose the appropriate management EPG.

Note: The default is **default (Out-of-Band)**.

4. Click **+** next to **DNS Providers** to add a DNS provider.
   a. In the **Address** field, enter the provider address.
   b. In the **Preferred** field, click the check box if you want to have this address as the preferred provider.
      Note: You can have only one preferred provider.
   c. Click **Update**.
5. Repeat step 4 for each additional DNS provider.
6. Click **+** next to **DNS Domains** to add a DNS domain.
   a. In the **Name** field, enter the domain name, such as "cisco.com".
   b. In the Default field , click the check box to make this domain the default domain.
      Note: You can have only one domain name as the default.
   c. Click **Update**.
7. Repeat step 6 for each additional DNS Domain suffix.
8. Click **Submit**.

Verifying DNS Operation

1. Check the **resolv.conf** file from the APIC CLI:

```
admin@apic1:~> cat /etc/resolv.conf
# Generated by IFC
search cisco.com

nameserver 171.70.168.183

nameserver 173.36.131.10
```

2 Ping a host by DNS name that will be reachable from the APIC out-of-band management.

```
admin@apic1:~> ping cisco.com
PING cisco.com (72.163.4.161) 56(84) bytes of data.
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=1 ttl=241 time=42.7 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=2 ttl=241 time=42.4 ms
64 bytes from www1.cisco.com (72.163.4.161): icmp_seq=3 ttl=241 time=43.9 ms
^C
--- cisco.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2102ms
rtt min/avg/max/mdev = 42.485/43.038/43.903/0.619 ms
admin@apic1:~>
```

Role-Based Access Control

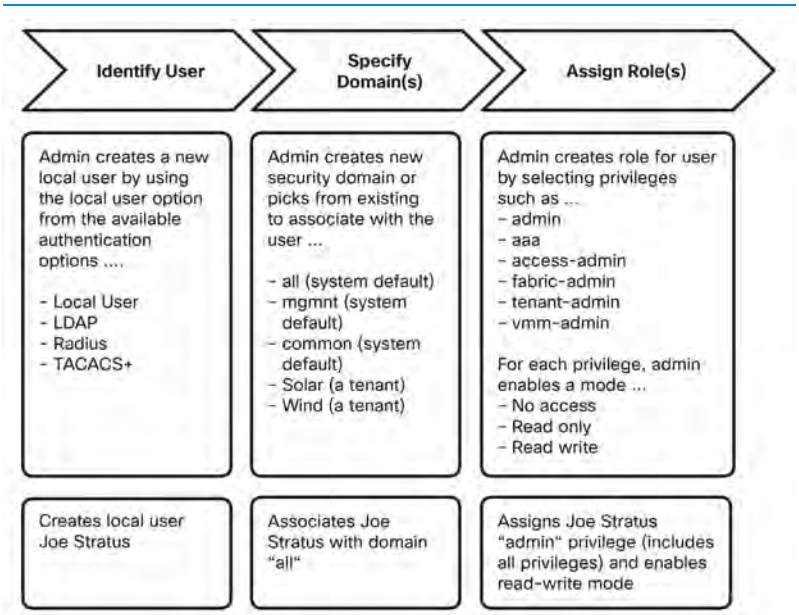
Role-based access control (RBAC) and tenancy are important concepts to master when operating a Cisco Application Centric Infrastructure (ACI) fabric. Role-based access and the concept of tenancy are two core foundations upon which the ACI management and policy models are built.

In an ACI fabric, the Cisco Application Policy Infrastructure Controller (APIC) grants access to all objects based on a user's established role in the configuration. Each user can be assigned to a set of roles, a privilege type, and a security domain. Think of this in terms of **Who**, **What**, and **Where**. The role is the **Who** (logical collection of privileges), privileges define **What** functions a user can perform, and the security domain defines **Where** the user can perform these actions. When combining these objects, you can implement very granular access control in the system.

The role classification is an established grouping of permissions. For example, a *fabric-admin* can have access to the entire fabric as well as to assigning other user roles and associate them to security domains, whereas a *tenant-admin* can only have access to the components within the tenant to which they're associated, and is unable to manage other user roles at a fabric wide level. ACME's IT organization fits perfectly into this access control model.

The APIC provides access according to a user's role through RBAC. An ACI fabric user is associated with the following:

- A set of roles
- For each role, privileges and privilege types (which can be "no access", "read-only", or "read-write")
- One or more security domain tags that identify the portions of the management information tree (MIT) that a user can access



Authenticated user creation lifecycle

The ACI fabric manages access privileges at the managed object (MO) level. A privilege is an MO that enables or restricts access to a particular function within the system. For example, **fabric-equipment** is a privilege flag. This flag is set by the APIC on all objects that correspond to equipment in the physical fabric.

APIC policies manage the access, authentication, and accounting (AAA) functions of the Cisco ACI fabric. The combination of user privileges, roles, and security domains with access rights inheritance enables administrators to configure AAA functions at the managed object level in a very granular fashion. These configurations can be implemented using the REST API, CLI, or GUI.

Multiple Tenant Support

A core APIC internal data access control system provides multi-tenant isolation and prevents information privacy from being compromised across tenants. Read/write restrictions prevent any tenant from seeing any other tenant's configuration, statistics,

faults, or event data. Unless the administrator assigns permissions to do so, tenants are restricted from reading fabric configuration, policies, statistics, faults, or events.

If a virtual machine management (VMM) domain is tagged as a security domain, the users contained in the security domain can access the corresponding VMM domain. For example, if a tenant named **solar** is tagged with the security domain called **sun** and a VMM domain is also tagged with the security domain called **sun**, then users in the s**olar** tenant can access the VMM domain according to their access rights.

User Roles

You can view the built-in user roles as well as create custom roles to meet specific requirements.

1. On the menu bar, choose **Admin > AAA.**
2. In the Navigation pane, choose **Security Management > Roles**.
   Note: From here, you can see each built-in role and the associated privileges. Additionally, you can create a custom role from the **Actions** menu.

Security Domains

The security domain concept is crucial to proper operation of the ACI fabric. By using security domains, users can be organized into various permission structures, most commonly applied to tenants. Using the tenancy capabilities of the ACI fabric in conjunction with properly configured security domains, it will be possible to completely separate configuration of application workloads while only providing access to those who need it.

A key concept to keep in mind when configuring security domains is that the configuration only applies at a tenant level. The policies cannot currently be applied to individual objects despite references at the object level inside of the RBAC screen in the APIC.

Cisco recommends that you configure security domains and access levels prior to deployment of tenants. Cisco does not recommend that you provide user access configuration by modifing permissions of the "all" domain. Changes in the "all" domain will affect access permissions for all users. If you need to make selective changes to allow access outside of a user's security domain, be sure to set up a discrete user access policy just for that communication.

Creation of a Security Domain

There are three security domains that are provisioned by default on the ACI fabric: "all", "common", and "mgmt". The "all" security domain usually includes access to everything within the management information tree (MIT). "Common" is usually used when there is a need for shared resources between tenants, such as DNS or directory services. The "mgmt" security domain is for management traffic policies. You can add security domains as necessary. For example, for a multi-tenant environment, a security domain would be created for each tenant. Users are then created and assigned to certain security domains, or tenants. RBAC policies are configured under the "Admin" tab of the APIC GUI.

For more information about the MIT, see the document at the following URL:

http://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/aci-fabric-controller/white-paper-c11-729586.html

To create a security domain:

1. On the menu bar, choose **Admin > AAA.**
2. In the Navigation pane, choose **Security Management >** S**ecurity Management**.
3. In the Work pane, choose **Actions > Create a Security Domain**.
4. In the **Create a Security Domain** dialog box, perform the following actions:
   a. Give the security domain a name and an optional description.

Adding Users

You might need to add new users within the ACI fabric. ACI can have local users manually entered by an admin, or use something such as LDAP, RADIUS, Active Directory, or TACACS+ to specify users that will be allowed to manage certain parts of the ACI network.

Configure Local Users:

1. On the menu bar, choose **Admin > AAA.**
2. In the Navigation pane, choose **AAA Authentication**.
3. In the Work pane, in the **Realm** drop-down list, choose **Local** and click **Submit** if **Local** is not already chosen.

4. In the Navigation pane, choose **Security Management**.
5. In the Work pane, choose **Actions > Create Local User**.
6. In the **Create Local User** dialog box, perform the following actions:
   a. Specify any security information that is necessary and click **Next**.
   b. Select the **roles** to be given to this user, such as Read/Write for admin or tenant admin, and click **Next**.
   c. Specify login information for the user and
7. Click **Finish**.

Remote Authentication

Creating remote user accounts in the ACI is similar to most other data center systems. If creating an LDAP account, then the LDAP provider must be configured first. The IP address or host name of the LDAP server is needed, along with the port it uses to communicate as well as any other relevant information that will allow connection to that server. The same is true for RADIUS and TACACS+. The next option is to configure groups from which it is allowed to read data and grab it for the purposes of selecting remote users granted access to the ACI network. Remote authentication is covered in detail in the *Cisco ACI Fundamentals Guide*.

Import and Export Policies

All of the stakeholders at ACME involved in the deployment and administration of the new mobile application platform need to know that they will be able to easily recover from any loss of configuration quickly and easily. Since all APIC policies and configuration data can be exported to create backups and tech support files for Disaster Recovery, troubleshooting, and offline analysis, ACI is a good choice on this front as well. As with all things APIC, a policy needs to be configured to export or backup the configuration policies. This can be done from any active and fully fit APIC within the ACI fabric. The backup and restore process does not require backup of individual components.

Backups are configurable through an export policy that allows either scheduled or immediate backups to a remote server (preferred) or, in the case where an external SCP/FTP server is not available, backups to be written to the local APIC file system.

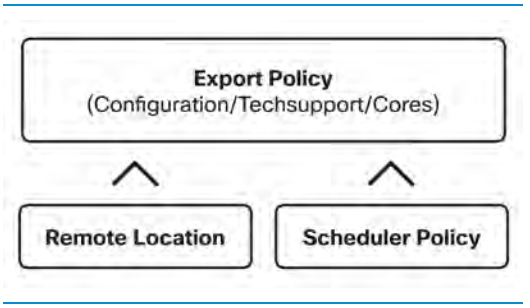
Backup/export policies can be configured to be run on-demand or based on a recurring schedule. Cisco recommends that a current Backup be performed before making any major system configuration changes or applying software updates. Configuration Import policies (policy restore) will be discussed later in this section.

By default, all policies and tenants are backed up, but the administrator can optionally specify only a specific subtree of the management information tree. This is useful in multi-tenant deployments where individual tenants wish to backup or restore their respective policies.

Another Export policy required on occasion is for gathering technical support information. If issues are encountered within the system, you might need to contact the Cisco Technical Assistance Center (TAC). Providing a technical support bundle will give Cisco support engineers the information needed to help identify and suggest remediation to issues. Technical support export policies can be configured to run on-demand or scheduled for recurring purposes. Technical support bundles are configured very similar to configuration export policies, and so the bundles will not be covered in detail here. For details on technical support export policies, see the *Cisco APIC Troubleshooting Guide* that is referenced in the Appendix of this book.

Configuration Export (Backup)

Since ACI is policy driven, more than a backup job is required. First, a remote location is created (the external server to which you want to export information), then an Export policy task, and optionally a Scheduler (for recurring tasks). The procedure below details how to create a remote location and export policy in ACI to transfer the configuration file bundle to an SCP server. The individual XML files can be extracted and viewed after the configuration bundle is transferred to the desktop. An extraction utility is needed to decompress the tar.gz file that is created.



Configuration Export Policy

Add a Remote Location (SCP)

1. On the menu bar, choose **Admin > Import/Export.**
2. In the Navigation pane, choose **Remote Locations**.
3. In the Work pane, choose **Actions > Create Remote Location**
4. In the **Create Remote Location** dialog box, perform the following actions:
   a. Enter a Remote location name
   b. Enter a Hostname/IP address
   c. Choose a Protocol
   d. Enter a Remote path
   e. Enter a Remote port
   f. Enter a Username
   g. Enter a Password
   h. Choose a Management EPG
      Note: default (Out-of-Band)
5. Click **Submit**.

Create a One Time Export Policy

The procedure below details a configuration export policy, but the procedure for a technical support export policy is very similar.

1. On the menu bar, choose **Admin > Import/Export**.
2. In the Navigation pane, choose **Export Policies > Configuration**.
3. In the Work pane, choose **Actions > Create Configuration Export Policy**
4. In the **Create Configuration Export Policy** dialog box, perform the following actions:
   a. Name = *Export_Policy_Name*
   b. Format = XML
   c. Start Now = Yes
   d. Export Destination = *Choose_the_Remote_location_created_above*
5. Click **Submit**.

Two optional configurations are applying a Scheduler policy if you want to setup a re-curring operation, and specifying a specific Distinguished Name (DN) if you want to backup only a subset of the Management Information Tree (MIT).

Verify Export Policy was Successful

1. On the menu bar, choose **Admin > Import/Export**.
2. In the Navigation pane, choose **Export Policies > Configuration > *Export_Name*.**
3. In the Work pane, choose the **Operational** tab.
   a. The **State** should change from "pending" to "success" when the export completes correctly.
   b. (Optional) Confirm on the SCP server that the backup filename exists.

Extract and View Configuration Files

A configuration export task exports a compressed bundle of individual XML files. These XML configuration files can be extracted and viewed on another workstation.

1. From the workstation where the exported bundle has been copied, select the archive utility of choice.
2. Navigate to the folder where the config export is located (tar.gz), highlight the file, and then select **Extract**.
3. Double-click one of the XML files to view the contents in a browser.
4. Examine the various XML configuration files for parameters that have been configured.

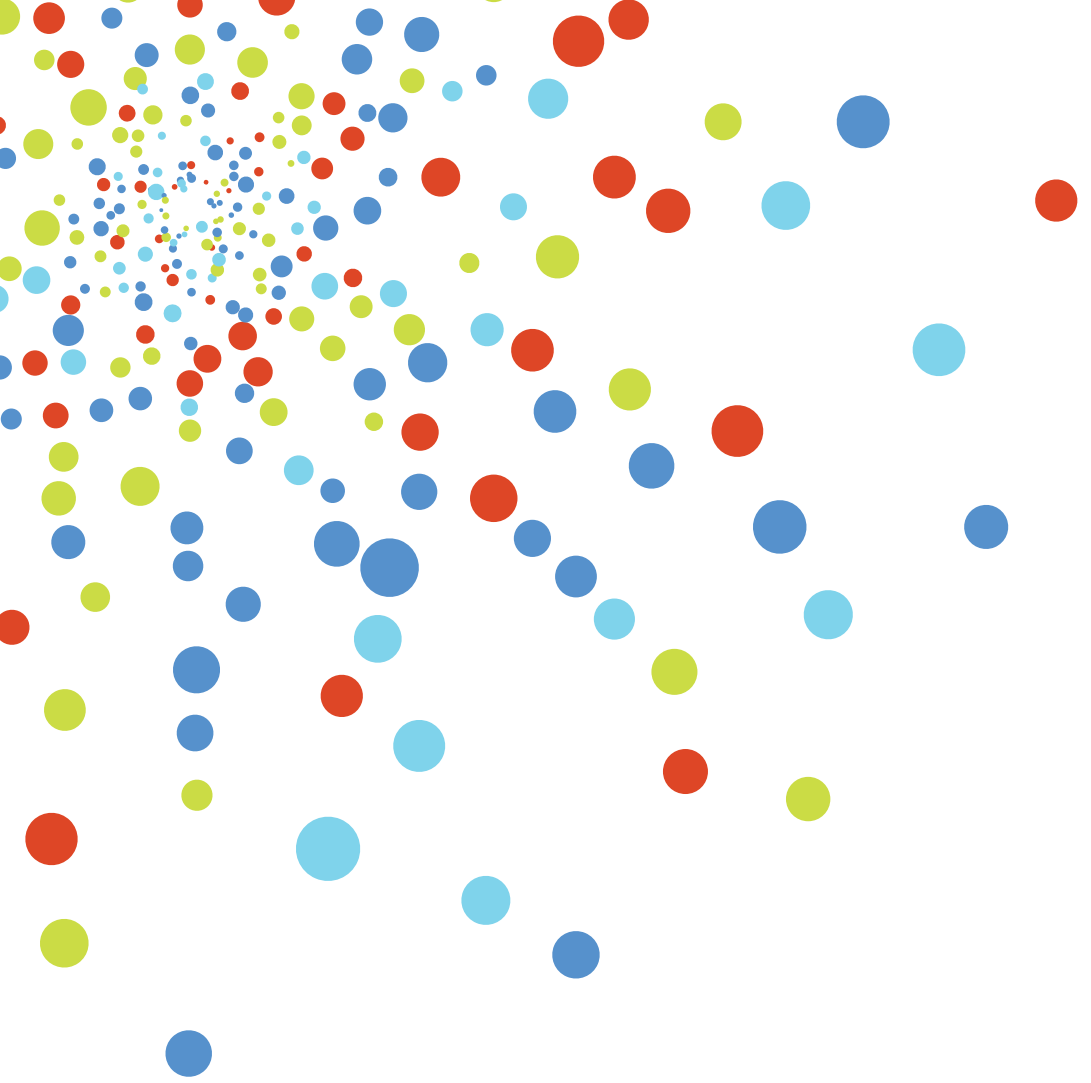
Configuration Import (Restore/Merge)

To restore a configuration from a previous backup:

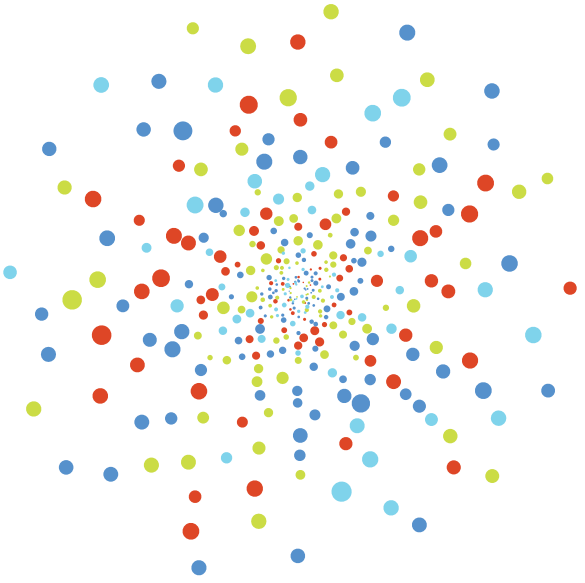
If the remote location does not exist, create a remote location per the "Adding a Remote Location (SCP)" section.

1. On the menu bar, choose **Admin > Import/Export**.
2. In the Navigation pane, choose **Import Policies > Configuration.**
3. In the Work pane, choose **Actions > Create Import Configuration Policy**.
4. In the **Create Import Configuration Policy** dialog box, perform the following actions:
   a. Enter a name and the filename for the import policy, such as "backup-file.tar.gz". Do not include the file path.
   b. Choose the import type:
      Merge - Will merge backup configuration with existing running configurations. Will not overrite any existing policies.
      Replace - Will replace all configurations with those only from the backup file.

c. The import mode must be specified when you attempt to perform a **Merge** import. The configuration data is imported per shard with each shard holding a certain part of the system configuration objects. The default is **best-effort**. The Replace Import Mode can be either:
best-effort - Each shard is imported, skipping any invalid objects
atomic - Attempts to import each shard, skipping those which contain an invalid configuration. A shard's entire configuration must be valid to be imported in this mode.
d. Choose **Start Now**.
e. Choose the **Remote Location** that was previously created in the "Adding a Remote Location (SCP)" section.



Upgrading and Downgrading Firmware



Section Content

Firmware Management

ACME Inc., in partnership with Cisco, has evaluated the requirements for their deployment based on the software features required, the support for the hardware platforms they have selected, and the maturity of the software releases. They have selected a target version of software for their deployment. Additionally, they have put a proactive plan in place to revisit this decision periodically to determine if future upgrades are required.

Firmware Versions

The software versions for ACI are listed in the following format:

```
major.minor(mntnc)
```

- *major* - Represents major changes in the product architecture, platform, or features content.
- *minor* - Represents a minor release with new software features.
- *mntnc* - Represents bug fixes to a feature release of APIC. This changes when there are fixes for product defects in the software, but no additional new features.

Example:

```
APIC version: 1.1(1d)
```

Both the software for the APIC and the fabric nodes are denoted by the same versioning scheme, For example, APIC release 1.0(2j) corresponds to switch software 11.0(2j). Release notes for the APIC versions reference the corresponding switch versions and vice-versa.

All components of the ACI infrastructure including the Cisco Application Policy Infrastructure Controller (APIC), leaf switches, and spine switches, should be on the same

version. While at the time of upgrading, disparate versions may exist between APIC and the switches, do not operate the fabric for extended periods of time in this state.

When considering the impact and risk of upgrading, you can assume that a maintenance version upgrade, such as 1.1.(1d) => 1.1(1d), will have less impact than a major/minor version upgrade, as there will be only bug fixes and no new features added.

Firmware Components

There are three main components that can be upgraded:

- Switches (leaf and spine)
- Application Policy Infrastructure Controller (APIC)
- Catalog firmware

Firmware Policies

Firmware Groups

Firmware Group policies on the APIC define which group of nodes on which firmware will be upgraded. For most deployments, a single firmware group is adequate. Control over which switches upgrade to the new version can be determined through maintenance groups.

Maintenance Groups

Maintenance Group policies define a group of switches that will be jointly upgraded to the associated firmware set. Maintenance groups can be upgraded on demand or according to a schedule, making it possible to defer an upgrade task to a business maintenance window.

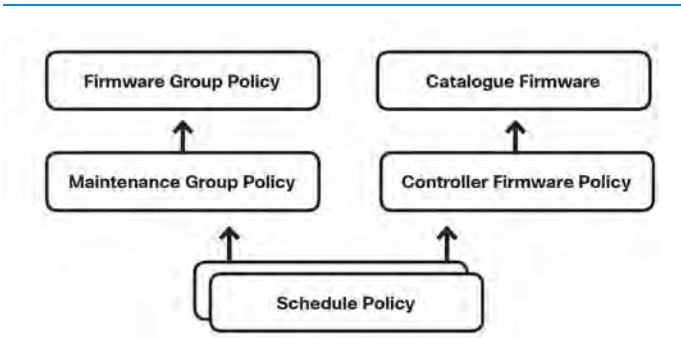
Controller Firmware

The APIC controller firmware policy always applies to all controllers in the cluster, but the upgrade is always done sequentially. The APIC GUI provides real-time status infor-

mation about firmware upgrades. Controller Firmware policies can be upgraded on demand or also according to a schedule.

Catalogue Firmware

Each firmware image includes a compatibility catalog that identifies supported switch models. The APIC maintains a catalog of the firmware images, switch types, and models that are allowed to use that firmware image. The APIC, which performs image management, has an image repository for compatibility catalogs, APIC controller firmware images, and switch images.



Firmware upgrade policy relationships

Upgrading and Downgrading Considerations

Before starting the upgrade or downgrade process, consider the following things:

- Before starting the upgrade process, your controllers should be in good health. On the menu bar, choose **System > Controllers**. In the navigation pane, choose **Controllers > *apic_controller_name* > Cluster**. Verify that the health state of all of the controllers in the cluster are **Fully Fit** before you proceed. To resolve issues for controllers that are not fully fit see the *Troubleshooting Cisco Application Centric Infrastructure* document:https://datacenter.github.io/aci-troubleshooting-book/
- Configuration backup. Before starting any upgrade, always export your configuration to an external source. For information about exporting configurations, see the Import and Export Policies chapter.
- Permissions. A user must have the fabric administrator role to perform firmware upgrade tasks.
- Verify free space. On the menu bar, choose **System > Controllers**. In the navigation pane, choose **Controllers >** *apic_controller_name* **> Storage.** Confirm that the **/firmware** partition is not filled beyond **75%**. If the partition is filled beyond 75%, you might be required to remove some unused firmware files from the repository to accommodate the compressed image as well as provide adequate space to extract the image. The APIC automatically extracts the image.
- Upgrade order. Typically, the controllers should be upgraded first, followed by the switch nodes. Always refer to the relevant release notes of the destination firmware version for any changes to this order.
- Maintenance windows. Although it is possible to upgrade the fabric without impacting the dataplane, you should perform an upgrade during a scheduled maintenance window according to your change control policy. This window should account for any unforseen issues that might arise during the upgrade, and allocate enough time to troubleshoot or perform a rollback.
- Maintenance groups. To help minimize the impact to hosts during an upgrade, you should set up at least two separate maintenance groups. A common separation is by odd and even node IDs. Assuming that your hosts are dual-connected to at least one odd and one even leaf node, there should not be any

impact to your hosts. Maintenance group creation is covered in detail later in the chapter. Another consideration is that your leaf VPC pairs should contain one odd and one even node.

- Upgrading a fabric with the Application Virtual Switch (AVS) deployed. The AVS software is not specifically tied to the APIC or switch software version.
- Device packages. Device packages are not always tied to the APIC software. You can confirm the device compatability for Layer 4 to Layer 7 devices using the online Application Centric Infrastructure (ACI) Compatability tool: http://www.cisco.com/web/techdoc/aci/acimatrix/matrix.html

Upgrading the Fabric

Downloading the Firmware Images

You must download both the controller software package and switch software package for the Application Policy Infrastructure Controller (APIC) from Cisco.com.

To download the firmware images using the APIC GUI:

1. On the menu bar, choose **Admin > Firmware.**
2. In the Navigation pane, choose **Fabric Node Firmware**.
   Note: In the Work pane, the list of all switches in the fabric and the status of when the firmware was last upgraded are displayed.
3. In the Navigation pane, choose **Download Tasks**.
4. In the Work pane, choose **Actions > Create Outside Firmware Source**.
5. In the **Create Outside Firmware Source** dialog box, perform the following actions:
   a. In the **Source Name** field, enter a name for the switch image, such as "apic_1.1.3d".
   b. For the **Protocol** radio buttons, click the **Secure copy** or **HTTP** radio button.
   c. In the URL field, enter the URL from where the image must be downloaded.
      - HTTP Example: http://192.168.0.50/aci-apic-dk9.1.1.3d.iso
      - SCP Example: 192.168.0.50:/tmp/aci-firmware/aci-apic-dk9.1.1.3d.iso
        - For SCP, enter your username and password.
   d. Click **Submit**.
6. In the Navigation pane, choose **Download Tasks**.
7. In the Work pane, choose the **Operational** tab to view the download status of the images.
8. Repeat the steps for the switch image.
9. Once the download reaches 100%, in the Navigation pane, choose **Firmware Repository**.

10 In the Work pane, the downloaded version numbers and image sizes are displayed.

To download the firmware images using the APIC CLI:

1 SSH to an APIC and log in as "admin".

2 Pull the software using SCP:

```
admin@apic1:~> scp
username@IP_address_with_the_image:/absolute_path_to_image_plus_image_filename &#10!
```

3 Place the image into the image repository:

```
admin@apic1:~> firmware add ver_no.iso &#10!
```

The firmware image *ver_no*.iso is added to the repository.

4 Verify the software has been added to the respository:

```
admin@ifc1:~> firmware list
Name : aci-apic-dk9.1.1.1d.bin
Type : controller
Version : 1.1(1d)
```

Upgrading the APIC Controller Software

The catalog firmware image is upgraded when an APIC controller image is upgraded. You do not need to upgrade the catalog firmware image separately.

To upgrade the APIC controller software using the GUI:

1 On the menu bar, choose **Admin > Firmware**.

2 In the Navigation pane, click **Controller Firmware**.

3 In the Work pane, choose **Actions > Upgrade Controller Firmware Policy**.

4 In the **Upgrade Controller Firmware Policy** dialog box, perform the following actions:

a. In the **Target Firmware Version** field, from the drop-down list, choose the image version to which you want to upgrade.
b. In the **Apply Policy** field, click the **Apply now** radio button. Alternately, you can apply a schedule policy if you wish to defer the task to a specific date/time.
c. Click **Submit** to complete the task.

The **Status** dialog box displays the Changes Saved Successfully message, and the upgrade process begins. The APIC controllers are upgraded serially so that the controller cluster is available during the upgrade.

5 Verify the status of the upgrade in the Work pane by clicking Controller Firmware in the Navigation pane.

The controllers upgrade in random order. Each APIC controller takes about 10 minutes to upgrade. Once a controller image is upgraded, it drops from the cluster, and it reboots with the newer version while the other APIC controllers in the cluster are still operational. Once the controller reboots, it joins the cluster again. Then the cluster converges, and the next controller image starts to upgrade. If the cluster does not immediately converge, and is not fully fit, the upgrade will wait until the cluster converges and is fully fit. During this period, a Waiting for Cluster Convergence message is displayed in the Status column for each APIC as it upgrades.

When the APIC controller that the browser is connected to is upgraded and it reboots, the browser displays an error message.

6 In the browser URL field, enter the URL for the APIC controller that has already been upgraded, and sign in to the APIC controller as prompted.

Upgrading the Switch Software Using the GUI

Before you upgrade the switches, the APICs must have completed upgrading and have a health state of **Fully Fit**.

To upgrade the switch software using the GUI:

1 On the menu bar, choose **Admin > Firmware**.

2 In the Navigation pane, choose **Fabric Node Firmware**.
Note: In the Work pane, the switches that are operating in the fabric are displayed.

3 If you have not created a firmware group, perform the following substeps:
a. In the Work pane, choose the **Policy** tab.
b. Choose **Actions > Create Firmware Group**.
c. In the **Create Firmware Group** dialog box, perform the following actions:
i. In the **Group Name** field, enter the name of the firmware group.
ii. In the **Target Firmware Version** drop-down list, choose the firmware version to which you will upgrade.
iii. In the **Group Node IDs** field, enter a comma-separated list or a range of node IDs to include in the group. For example, "101, 103-105, 108".
iv. Click **Submit**.

d. To verify that the firmware group was created, in the Navigation pane, choose **Fabric Node Firmware > Firmware Groups > *new_firmware_group***. The Work pane displays details about the firmware policy that was created earlier.

4 If you have not created maintenance groups, perform the following substeps:
a. In the Navigation pane, click **Maintenance Groups**.
Note: Cisco recommends that you create two maintenance groups for all of the switches. For example, create one group with the even-numbered devices and the other group with the odd-numbered devices.
b. In the Work pane, choose **Action > Create Maintenance Group**.
c. In the **Create Maintenance Group** dialog box, perform the following actions:
i. In the **Group Name** field, enter the name of the maintenance group. For example, "Even-Nodes".
ii. For the **Run Mode** radio buttons, click the **Pause Upon Upgrade Failure** radio button, which is the default mode.
iii. In the **Group Node IDs** field, enter a comma-separated list or a range of node IDs to include in the group. For example, "102, 104, 106, 108, 110".
iv. In the **Scheduler** drop-down list, you can choose to create a schedule for upgrading or leave the drop-down list blank so that you can upgrade on demand.
v. Click **Submit**.

vi. To verify that the maintenance group was created, in the Navigation pane, choose **Fabric Node Firmware > Maintenance Groups > *new_maintenance_group***, and click the name of the maintenance group that you created.
Note: The Work pane displays details about the maintenance policy.

vii. Repeat this step for the second maintenance group. For example, a group named "Odd-Nodes".

5 Right-click one of the maintenance groups that you created and choose **Upgrade Now**.

6 In the **Upgrade Now** dialog box, for **Do you want to upgrade the maintenance group policy now?**, click **Yes**.

7 Click **OK**.
Note: In the Work pane, the Status displays that all the switches in the group are being upgraded simultaneously. The default concurrency in a group is set at 20. Therefore, up to 20 switches at a time will get upgraded, and then the next set of 20 switches are upgraded. In case of any failures, the scheduler pauses and manual intervention is required by the APIC administrator. The switch upgrade takes up to 12 minutes for each group. The switches will reboot when they upgrade, connectivity drops, and the controllers in the cluster will not communicate for some time with the switches in the group. Once the switches rejoin the cluster after rebooting, you will see all the switches listed under the controller node. If there are any VPC configurations in the cluster, the upgrade process will upgrade only one switch at a time out of the two switches in a VPC domain.

8 In the Navigation pane, click **Fabric Node Firmware**.
Note: In the Work pane, view all of the switches that are listed. In the **Current Firmware** column, view the upgrade image details listed against each switch. Verify that the switches in the fabric are upgraded to the new image.

Upgrading the APIC Controller Software Using the CLI

The catalog firmware image is upgraded when an APIC controller image is upgraded. You do not need to upgrade the catalog firmware image separately. Cisco recommends that you to perform the firmware upgrade from the GUI. When you use the GUI, the APIC performs additional verification and integrity checks on the software image.

To upgrade the APIC controller software using the CLI:

1 List the current software in the respository that was previously downloaded.

Example:

```
admin@apic1:~> firmware list
Name : aci-apic-dk9.1.1.1d.bin
Type : controller
Version : 1.1(1d)
```

2 Upgrade the firmware on the controllers.

Example:

```
admin@apic1:~> firmware upgrade controllers ver_no.bin
```

The APIC controllers are upgraded serially so that the controller cluster is available during the upgrade. The upgrade occurs in the background.

3 Check the status of the upgrade.

Example:

```
admin@apic1:~> firmware upgrade status

Node-Id    Role          Current-      Target-           Upgrade-    Progress-Percent
                         Firmware      Firmware          Status      (if inprogress)
---------  ------------  ------------  ----------------- ----------  -----------------
1          controller    1.0(1.200)    apic-1.0(1.202a)  inqueue           0
2          controller    1.0(1.200)    apic-1.0(1.202a)  inqueue           0
3          controller    1.0(1.200)    apic-1.0(1.202a)  inprogress        0
```

The Upgrade-Status field will show "inqueue", "inprogress", or "completeok". If you see "unknown" in this field, that controller has probably upgraded and is rebooting.

When the APIC controller to which you have connected completes upgrading and reboots, you can close the SSH window.

Upgrading the Switch Software Using the CLI

Before you upgrade the switches, the APICs must have completed upgrading and have a health state of **Fully Fit**.

To upgrade the switch software using the CLI

1. Check that the output of the following command appears like the output shown below, with the correct version number:

   Example:

```
admin@apic1:~> firmware list

Name    : aci-n9000-dk9.11.1.1d.bin
Type    : switch
Version : 11.1(1d)
```

   The name changes from ".iso" to ".bin".

2. Upgrade the switches.

   Example:

```
admin@apic1:~> firmware upgrade switch node 101 ver_no.bin

Firmware Installation on Switch Scheduled
```

   You must upgrade each switch separately.

3. Check the upgrade status for the switch. The output that appears from the following command will appear like the following sample:

   Example:

```
admin@apic1:~> firmware upgrade status node node_id
 Node-Id    Role          Current-            Target-            Upgrade-    Progress-Percent
                          Firmware            Firmware           Status      (if inprogress)
 ---------  ----------  ------------------  ------------------ ----------  ------------------
 1017       leaf        n9000-11.0(1.869S1)  n9000-11.1(1d)     completeok      100
```

You can check the status of all nodes at once, by entering the firmware upgrade status command.

4 Repeat Steps 2 and 3 for each additional switch.

Verifying Cluster Convergence

You can monitor the progress of the cluster convergence after a scheduled maintenance. You view the **Controller Firmware** screen on the GUI, which presents you with a series of messages during the process of one cluster converging and then the next cluster. These messages are displayed in the **Status** field.

As the controller and switches move through the upgrade, you will see messages about the number of nodes queued and the number in the process of upgrading, as well as how many have upgraded successfully.

The following are the possible upgrade states for a node:

- NotScheduled: No upgrade is currently scheduled for this node.
- Scheduled: Upgrade is scheduled for this node.
- Queued: There is a currently active window (schedule) and the node is requesting permission to upgrade.
- Inprogress: Upgrade is currently in progress on this node.
- CompleteOK: Upgrade completed successfully.
- CompleteNOK: Upgrade failed on this node.
- Inretryqueue: Node is queued again for upgrade retry (5 attempts are made before declaring failure).

This may take a while. When all the clusters have converged successfully, you will see "No" in the **Waiting for Cluster Convergence** field of the **Controller Firmware** screen.

Troubleshooting Failures During the Upgrade Process

There is one scheduler per maintenance policy. By default, when an upgrade failure is detected, the scheduler pauses, and no more nodes in that group begin to upgrade. The scheduler expects manual intervention to debug any upgrade failures. Once manual intervention is complete, you must resume the paused scheduler.

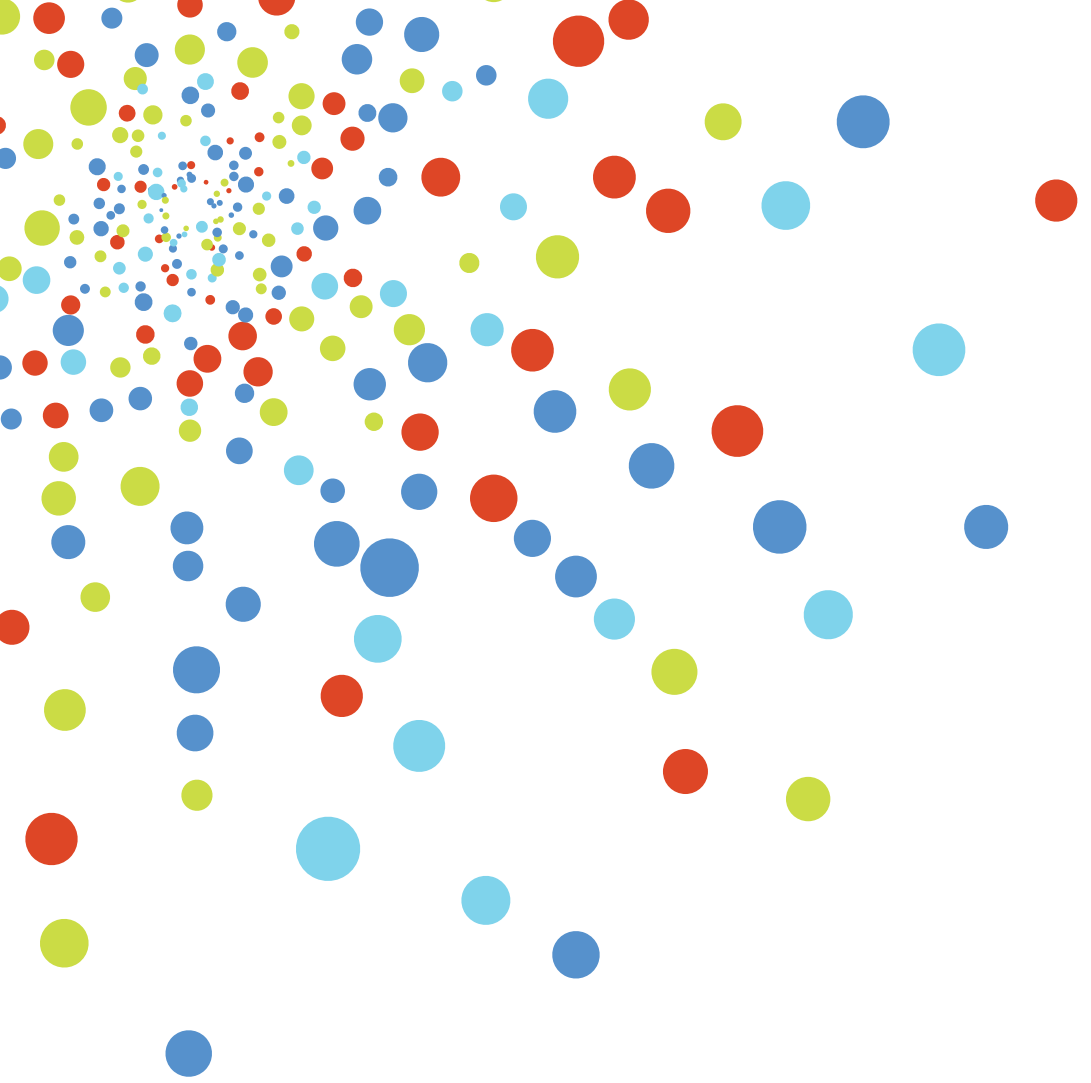
If you notice that switches are in the “queued” state, then check the following:

- Is the controller cluster healthy? The controller cluster must be healthy. If you see “waitingForClusterHealth = yes” in the API or "Waiting for Cluster Convergence" showing "Yes" in the GUI, that means the controller cluster is not healthy. Until the controller cluster is healthy, switches which have not already started their upgrade will be in “queued” state.
- Is the switch maintenance group paused? The group will be paused if any switch fails its upgrade.

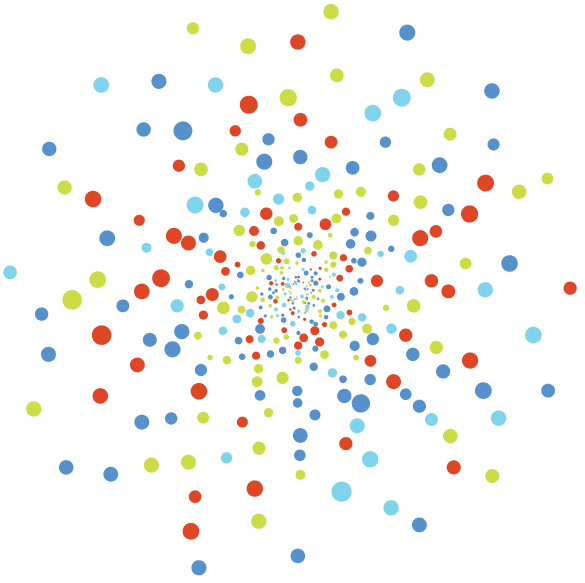
If the system takes longer than about 60 minutes for a switch to display “waitingForClusterHealth = no” in the API or "Waiting for Cluster Convergence" showing "No" in the GUI, you should work through the steps for verifying a pause in the scheduler.

For additional troubleshooting procedures, see the Troubleshooting Cisco Application Centric Infrastructure document at the following URL:

https://datacenter.github.io/aci-troubleshooting-book/



Fabric Connectivity



Section Content

- Understanding Fabric Policies
- Fabric – Access Policies
  - Domains
  - VLAN Pools
  - AEPs
  - Policy Types
    - Interface Policies
    - Switch Policies
    - Interface Policy Groups
    - Switch Policy Groups
    - Interface Profiles
    - Switch Profiles
  - Best Practices
- Adding New Devices to the Fabric
  - Sample Configuration
    - Creating VLAN Pools
    - Create VLAN Pool
    - Create Physical Domain
    - Create an Attachable Access Entity Profile (AEP)
    - Interface Policies
    - Interface Policy Groups
    - Interface Profile
    - Switch Profiles
  - Reusability
  - Sample vPC Creation
    - Create VLAN Pool
    - Create a Physical Domain

        - Create Access Entity Profile
        - Interface Policies
        - Switch Profile
        - Create vPC domain
        - Validate Operation of Configured vPC

- Server Connectivity
    - Cisco UCS B-Series Servers
    - Standalone Rack Mount Servers or Non-Cisco Servers

- Virtual Machine Networking
    - Understanding VM Networking in ACI
        - ACI VM Integration Workflow
    - VMware Integration
    - VMM Policy Model Interaction
    - Publishing EPGS to a VMM Domain
    - Connecting VMs to the EPG Port Groups on vCenter
    - Verifying Virtual Endpoint Learning
        - Verifying VM Endpoint Learning on the APIC from the CLI
    - VMware Integration Use Case

- Deploying the Application Virtual Switch
    - Prerequisites
    - Getting Started
    - Install the AVS VIB
    - Manual Installation
    - DHCP Relay
    - Attachable Access Entity Profile (AEP) and AVS
    - VMM Domains for vCenter
        - AVS Switching Modes
        - Create the VMM Domain for AVS
        - Verify AVS Deployment on vCenter
        - Add vSphere Hosts to the AVS

- Verify AVS on ESX
- VXLAN Load Balancing
- IGMP Snooping Policy for AVS

- External Connectivity
  - Extending ACI to External Layer 2
    - Extending an ACI Bridge Domain Outside of the Fabric
    - Extending Endpoint Groups Outside the ACI Fabric
  - Extending ACI to External Layer 3
    - Supported Routing Protocols
    - Configure MP-BGP Spine Route Reflectors
    - Layer 3 Integration Through Tenant Network with OSPF NSSA
    - External Layer 3 for Multiple Tenants

- Application Migration Use Case
  - Extending the Network to ACI

Understanding Fabric Policies

Now that ACME has been provisioned with ACI fabric and infrastructure space has been configured between the leaf and spine switches, access privileges, and basic management policies, it is time to start creating connectivity policies within the ACI fabric. The fabric tab in the APIC GUI is used to configure system-level features including, but not limited to, device discovery and inventory management, diagnostic tools, configuring domains, and switch and port behavior. The fabric pane is split into three sections: inventory, fabric policies, and access policies. Understanding how fabric and access policies configure the fabric is key for the ACME network teams, as they will be maintaining these policies for the purposes of internal connections between fabric leaf nodes, connections to external entities such as servers, networking equipment, and storage arrays. It is important that other teams such as server teams understand these concepts as well, as they will be interacting with them, particularly in the case of their build processes for adding additional capacity.

This chapter will review the key objects in the access policies subsection of the fabric tab – many of which are reusable; demonstrate how to add and pre-provision switches, and walk through the steps and objects required when new devices are added to the fabric to effectively operate an ACI fabric. While many policies are reusable, it is also key to understand the implications of deleting policies on the ACI fabric.

The access policies subsection is split into folders separating out different types of policies and objects that affect fabric behavior. For example, the interface policies folder is where port behavior is configured, like port speed, or whether or not to run protocols like LACP on leaf switch interfaces is set. Domains and AEPs are also created in the access policies view. The fabric access policies provide the fabric with the base configuration of the access ports on the leaf switches.

Fabric - Access Policies

Domains

EPGs are considered the “who” in ACI; contracts are considered the “what/when/why”; AEPs can be considered the "where" and domains can be thought

of as the “how” of the fabric. Different domain types are created depending on how a device is connected to the leaf switch. There are four different domain types: physical domains, external bridged domains, external routed domains, and VMM domains.

Physical domains are generally used for bare metal servers or servers where hypervisor integration is not an option.

External bridged domains are used for Layer 2 connections. For example, an external bridged domain could be used to connect an existing switch trunked-up to a leaf switch.

External routed domains are used for Layer 3 connections. For example, an external routed domain could be used to connect a WAN router to the leaf switch.

Domains act as the glue between the configuration done in the fabric tab to the policy model and EPG configuration found in the tenant pane. The fabric operator creates the domains, and the tenant administrators associate domains to EPGs.

For an in-depth whiteboard explanation on domains, check out the following video titled "How Devices Connect to the Fabric: Understanding Cisco ACI Domains": https://www.youtube.com/watch?v=_iQvoC9zQ_A

VLAN Pools

VLAN pools contain the VLANs used by the EPGs the domain will be tied to. A domain is associated to a single VLAN pool. VXLAN and multicast address pools are also configurable. VLANs are instantiated on leaf switches based on AEP configuration. Forwarding decisions are still based on contracts and the policy model, not subnets and VLANs.

AEPs

Attachable Access Entity Profiles (AEPs) can be considered the "where" of the fabric configuration, and are used to group domains with similar requirements. AEPs are tied to interface policy groups. One or more domains are added to an AEP. By grouping domains into AEPs and associating them, the fabric knows where the various devices in the domain live and the APIC can push the VLANs and policy where it needs to be. AEPs are configured under global policies.

Policy Types

Most of the policies folders have subfolders. For example, under the interface policies folder there are folders for configuration called policies, policy groups, and profiles.

Interface Policies

First, interface policies are created to dictate interface behavior, are are later tied to *interface policy groups*. For example, there should be a policy that dictates CDP is disabled, and a policy that dictates CDP is enabled - these can be reused as new devices are connected to the leaf switches.

Switch Policies

There are also policies for switches - for example, configuring vPC domains, which are called explicit vPC protection groups in the APIC GUI. Ideally, policies should be created once and reused when connecting new devices to the fabric. Maximizing reusability of policy and objects makes day-to-day operations exponentially faster and easier to make large-scale changes.

Interface Policy Groups

Interface policy groups are templates to dictate port behavior and are associated to an AEP. Interface policy groups use the policies described in the previous paragraph to specify how links should behave. These are also reusable objects as many devices are likely to be connected to ports that will require the same port configuration. There are three types of interface policy groups depending on link type: individual, Port Channel, and vPC. Note that the ports on the leaf switches default to 10GE, and a 1GE link level policy must be created for devices connected at that speed. Just like policies, there should ideally be a set of policy groups created once and reused as new devices are connected to the fabric. For example, there may be a policy that dictates 10GE, CDP-enabled. Note the interface policy groups simply dictate policy. Policy groups do not actually specify *where* the protocols and port behavior should be implemented. The "where" happens by associating one or more interface profiles to a switch profile, covered in the following paragraphs.

Switch Policy Groups

Switch policy groups allow leveraging of existing switch polices like Spanning Tree and monitoring policies.

Interface Profiles

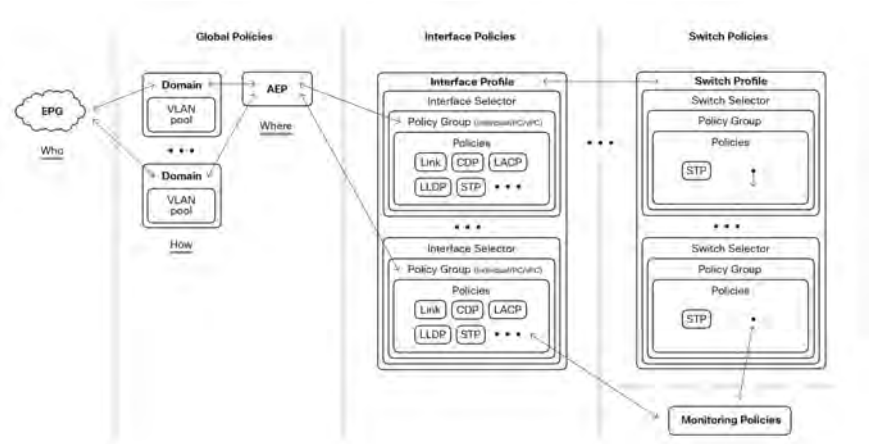
Interface profiles help tie the pieces together. Interface profiles contain blocks of ports - interface selectors - and are also tied to the interface policy groups described in the

previous paragraphs. Again, this is just an arbitrary port, such as e1/1, the profile must be associated to a specific switch profile (discussed in the next paragraph) to configure the ports.

Switch Profiles

Lastly, switch profiles allow the selection of one or more leaf switches and associate interface profiles to configure the ports on that specific node. This association pushes the configuration to the interface, and creates a Port Channel or vPC if one has been configured in the interface policy.

The following figure highlights the relationship between the various global, switch, and interface policies:



Relationships to allow a physical interface or interfaces to be attached to an EPG

Layer 2 Interface Policy

In Cisco ACI version 1.1, a new configurable Interface Policy was added to allow a per port-VLAN significance.

To connect devices to the ACI fabric we can use untagged traffic,VLAN encapsulation or VXLAN encapsulation.

In traditional networking one of the limitations related to VLAN encapsulation is scalability and re-usability due to the limit of 4096 VLANs in networking devices.

In ACI, with the default configuration (global), EPGs can use the same VLAN encapsulation as long as EPGs are bound to separate switches. This allows tenants to re-use VLAN encapsulation IDs thorough the fabric without allowing communication between tenants. However, global configuration assumes that tenants do not share leaf switches and therefore there is no VLAN overlapping within the same leaf.

- **Per Port-VLAN limitations and considerations**
  - When per port-VLAN is used, a port and VLAN pair (P,V) is registered internally instead of just a VLAN encapsulation ID. This increases the consumption of hardware resources at a per switch level.
  - Two EPGs belonging to a single Bridge Domain cannot share the same encapsulation ID on a given leaf switch.
  - It is expected that the port will flap when the Layer 2 interface policy changes between global and local. That is, traffic will get affected.

Best Practices

Cisco has established several best practices for fabric configuration. These are not requirements and might not work for all environments or applications, but might help simplify day-to-day operation of the ACI fabric.

- Policies
  - Reuse policies whenever possible. For example, there should be policies for LACP active/passive/off, 1GE port speed, and 10GE port speed.
  - When naming policies, use names that clearly describe the setting. For example, a policy that enables LACP in mode active could be called "LACP-Active". There are many "default" policies out of the box. However, it can be hard to remember what all the defaults are, which is why policies should be clearly named to avoid making a mistake when adding new devices to the fabric.
  - A set of interface policy groups should be created for each type of similar devices connected. For example, there can be a set of interface policy groups for all VMware ESXi servers connected via 10GE vPCs, and a different set of interface policy groups for all bare metal servers running

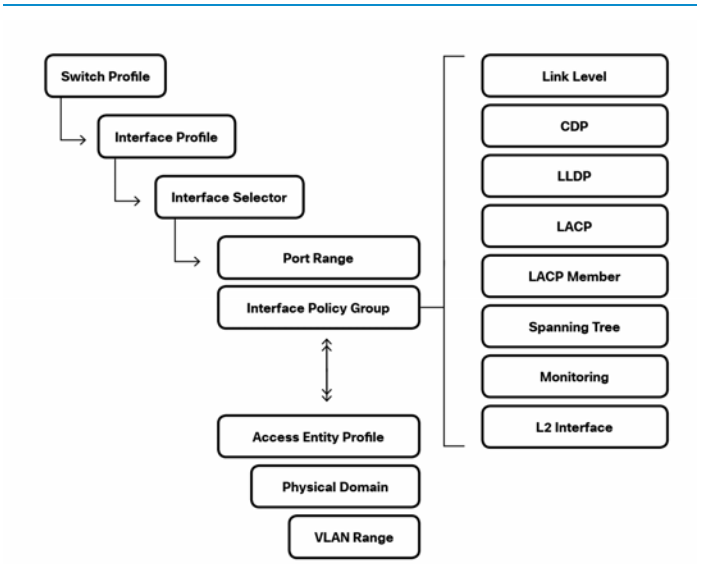
1GE with CDP disabled. Since interface policy groups are tied to a single AEP, each AEP will have its own set of interface policy groups.

- Create a switch profile for each leaf switch individually, and additionally, create a switch profile for each vPC pair (if using vPC).

- Domains
  - Build one physical domain per tenant for bare metal servers or servers without hypervisor integration requiring similar treatment.
  - Build one physical domain per tenant for external connectivity.
  - If a VMM domain needs to be leveraged across multiple tenants, a single VMM domain can be created and associated with all leaf ports where VMware ESXi servers are connected.
- AEPs
  - Multiple domains can be associated to a single AEP for simplicity's sake. There are some cases where multiple AEPs may need to be configured to enable the infrastructure VLAN, such as overlapping VLAN pools, or to limit the scope of the presence of VLANs across the fabric.

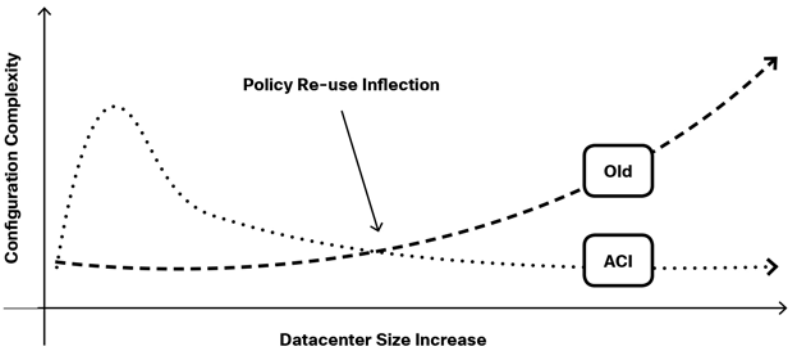
Adding New Devices to the Fabric

This section will demonstrate how to configure ACI to re-use the fabric access policies, simplifying day-to-day operation of the fabric. This section will walk through setting up profiles from scratch, with a focus on how to re-use these profiles across the fabric. As outlined in the previous section, these various profiles are linked together and have dependencies. The following diagram reiterates the object relationships:



Object relationships

Whereas a traditional command line interface on a switch generally requires a port-by-port confuguration, ACI allows definition of objects and policies that can be re-used. The re-usability of these policies makes it possible to replicate the configuration of a switch very easily. The following diagram depicts how this re-usability simplifies the operation of the fabric over time.



Policy Re-use

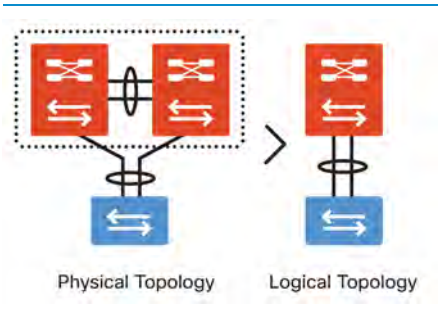
In any data center, the configuration of a couple of switches does not require many processes or automation. As the data center size increases, automation becomes more and more critical as it has a direct impact on the cost of business operations. In traditional networks, when changes that impact a large set of devices need to be made, the operator is faced with the cost of designing processes to manage these devices. These can be network management tools, scripts, or specialized applications. Leveraging the Cisco ACI policy model, an operator can leverage profiles to streamline the operation of adding devices and managing those devices. This is what is depicted as the policy re-use inflection point in the previous diagram.

Sample Configuration

The following sections will walk through sample configuration of setting up individually connected devices, Port Channel-connected devices, and vPC-connected devices from scratch, and will include a review of the objects as they are configured. These are the steps to be taken in the APIC GUI when new devices are connected to the leaf switches to ensure the access ports on the leaf switches have the right switchport configuration, and the verification steps to ensure proper configuration. The following steps represent the use case of adding a new bare metal server connected to a leaf switch.

Before getting into the configuration of vPC's, which are a popular server connectivity methodology, it is important to understand what vPC's are and how they are different from traditional methods of server connectivity. This section of the chapter attempts to clarify at a high level what vPC's are, the benefits they provide and how vPC's in the ACI fabric differ from how they are deployed on Cisco Nexus switches running NX-OS software.

At a high level, vPC extends link aggregation to two separate physical switches.



vPC Topology

In the figure above, a single server is dual homed to two different switches for redundancy. Without vPC's, the server will likely use an active-standby configuration, or a special configuration on the NIC driver or the kernel that allows it to intelligently load-balance traffic using an algorithm.

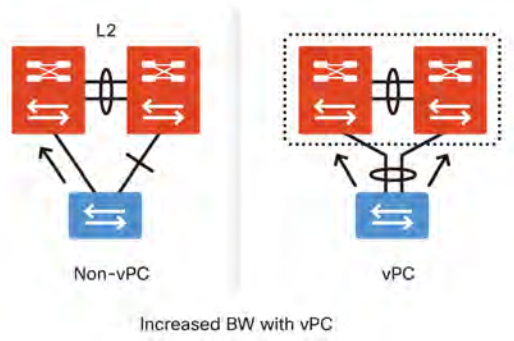
By configuring ports on two different switches as the same port-channel and using an inter-switch messaging channel (such as the inter-switch port-channel in the green box on the left hand side) to cover redundancy scenarios, we provide a logical topology that greatly simplifies server provisioning and management.

This allows for several key advantages from a server deployment perspective:

- You can create resilient Layer 2 topologies based on link aggregation
- You do not need STP
- You have increased bandwidth, as all links are actively forwarding
- Your server configurations are simplified since the configurations simply appears as port-channels without the need for special software, from a driver or kernel-tuning standpoint

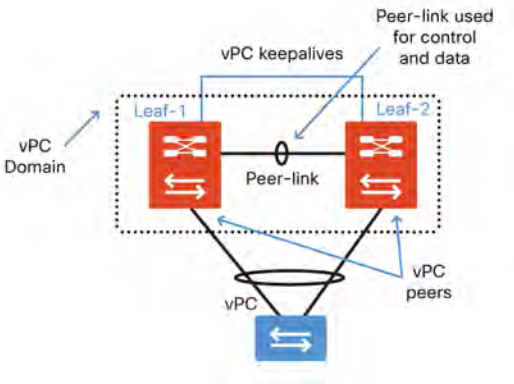
vPCs can also be used to connect other downstream devices, such as Cisco UCS fabric-interconnects, to provide similar benefits.

The figure below shows a single traditional Layer 2 switch connected to a VPC enabled Cisco switch pair.



Legacy connectivity compared to vPC

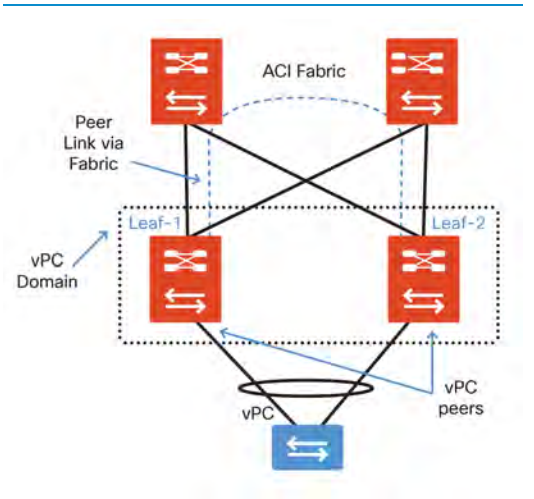
The components of a traditional vPC domain are illustrated below:



Traditional vPC topology

As illustrated above, in Cisco switching products running NX-OS software, vPC configurations need to be done manually by the operator and require a pair of dedicated "inter-switch" links also called a peer-link. There is also a peer-keepalive link, typically on the out-of-band management port, that is used to determine peer liveliness to detect a vPC peer-switch failure. Making configuration changes in such scenarios without the config-sync feature enabled may lead to scenarios where there are mismatched vPC parameters between the vPC primary and the vPC secondary switches that may cause partial connectivity loss during the change itself if a type-1 inconsistency is detected.

The ACI fabric greatly simplifies VPC configurations.



vPC topology in ACI

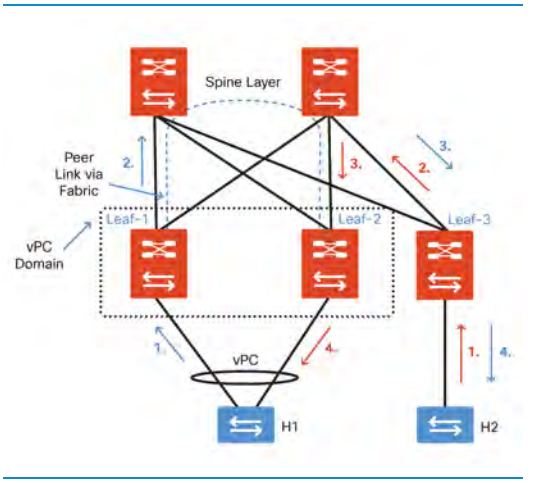
The key differences to note here are that relative to traditional vPC design, there is no requirement for setting up vPC peer-links. There are also no keepalives being sent on the management ports. The fabric itself serves as the peer-link. The rich interconnectivity between fabric nodes makes it unlikely that peers will have an active path between them.

Note that attempting to cable a leaf switch to another leaf switch will lead to a "wiring mismatch" fault in the GUI and result in a blacklisted port that will have to be manually recovered.

The following are some other key behavioral changes to vPC as it applies to the ACI fabric relative to classic vPC that are important for operators to understand:

- Configurations are automatically synchronized to avoid an error-free configuration by the APIC which is the central point of control for all configurations in the ACI fabric.
- In traditional vPC solution, the slave switch brings down all its vPC links if the MCT goes down.
- In the ACI fabric, it is very unlikely that all the redundant paths between vPC peers fail at the same time. Hence if the peer switch becomes unreachable, it is assumed to have crashed. The slave switch does not bring down vPC links.
- Role election still happens, peers assume master-slave roles.
- Role is used in case of vpc type-1 consistency failure. Slave switch brings down all its vPC ports. A list of type-1 parameters used for consistency checking for a given vPC domain specific to the ACI fabric are listed below.
- Global type-1 parameters:
  - STP
- Interface type-1 parameters:
  - STP: Only BPDU Guard is configurable
  - EthPM
  - Port speed
  - Duplex mode
  - Port mode
  - MTU
  - Native VLAN
    - PCM: Channel mode, static vs lacp
    - LACP: Lag ID

The following diagrams illustrate how the ACI fabric forwards traffic from a vPC domain to a non-vPC connected host in the fabric, and vice-versa.



vPC forwarding

Unicast packet flow H2 -> H1

1. H2 sends a pkt towards H1 on its link to S3.
2. S3 does a table lookup and routes with vPC Virtual IP (VIP).
3. Spine switch sees multiple routes for VIP and picks one of them (S2 in this case).
4. S2 delivers the pkt to locally attached host H1.

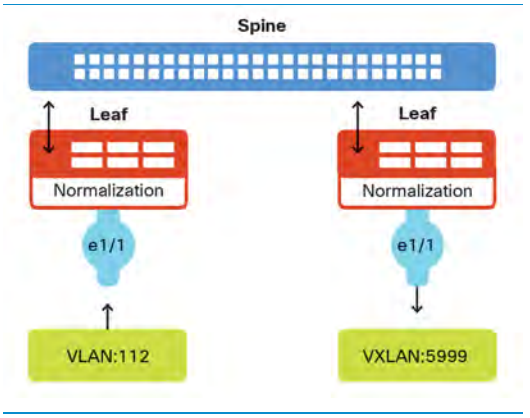
H1 -> H2

1. H1 sends a pkt towards H2 on one of its PC link (S1 in this case).
2. S1 does a table lookup and routes with IP of S3.
3. Spine switch routes to S3.
4. S3 delivers the pkt to locally attached host H2.

Creating VLAN Pools

In this example, configuring newly-connected bare metal servers first requires creation of a physical domain and then association of the domain to a VLAN pool. As mentioned in the previous section, VLAN pools define a range of VLAN IDs that will be used by the EPGs.

The servers are connected to two different leaf nodes in the fabric. Each server will be tagging using 802.1Q or VXLAN encapsulation. The range of VLANs used in the configuration example is 100-199. As depicted in the following figure, the ACI fabric can also act as a gateway between disparate encapsulation types such as untagged traffic, 802.1Q VLAN tags, VXLAN VNIDs, and NVGRE tags. The leaf switches normalize the traffic by stripping off tags and reapplying the required tags on fabric egress. In ACI, it is important to understand that the definition of VLANs as they pertain to the leaf switch ports is utilized only for identification purposes. When a packet arrives ingress to a leaf switch in the fabric, ACI has to know beforehand how to classify packets into the different EPGs, using identifiers like VLANs, VXLAN, NVGRE, physical port IDs, virtual port IDs.



Encapsulation normalization

Create VLAN Pool

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Pools > VLAN**.
3. In the Work pane, choose **Actions > Create VLAN Pool**.
4. In the **Create VLAN Pool** dialog box, perform the following actions:
   a. Define a meaningful name for the VLAN pool.
   b. Optionally, provide a description for the VLAN pool.
   c. There are two allocation modes: dynamic or static.
   Note: When dynamic allocation is selected, the APIC selects VLANs from the pool dynamically. This is common in VMM integration mode where the actual

VLAN ID is not important. Remember it's the EPG itself which policies are applied to. Static allocation is typically used when the pool will be referenced from a static source like a static path binding for an EPG for use with bare metal servers.

d. The encap blocks are used to define the range of VLANs in the VLAN pool. Note multiple ranges can be added to a single pool.

XML Object

```
<fvnsVlanInstP allocMode="static" childAction="" configIssues="" descr=""
dn="uni/infra/vlanns-[bsprint-vlan-pool]-static" lcOwn="local" modTs="2015-02-
23T15:58:33.538-08:00" monPolDn="uni/fabric/monfab-default" name="bsprint-vlan-pool"
ownerKey="" ownerTag="" status="" uid="8131">
    <fvnsRtVlanNs childAction="" lcOwn="local" modTs="2015-02-25T11:35:33.365-08:00"
rn="rtinfraVlanNs-[uni/l2dom-JC-L2-Domain]" status="" tCl="l2extDomP" tDn="uni/l2dom-
JC-L2-Domain"/>
    <fvnsRtVlanNs childAction="" lcOwn="local" modTs="2015-02-23T16:13:22.007-08:00"
rn="rtinfraVlanNs-[uni/phys-bsprint-PHY]" status="" tCl="physDomP" tDn="uni/phys-
bsprint-PHY"/>
    <fvnsEncapBlk childAction="" descr="" from="vlan-100" lcOwn="local" modTs="2015-02-
23T15:58:33.538-08:00" name="" rn="from-[vlan-100]-to-[vlan-199]" status="" to="vlan-
199" uid="8131"/>
</fvnsVlanInstP>
```

Create Physical Domain

A physical domain acts as the link between the VLAN pool and the Access Entity Profile (AEP). The domain also ties the fabric configuration to the tenant configuration, as the tenant administrator is the one who associates domains to EPGs, while the domains are created under the fabric tab. When configuring in this order, only the profile name and the VLAN pool are configured. The creation of the AEP and its association will be covered later in this section.

1 On the menu bar, choose **Fabric > Access Policies**.

2 In the Navigation pane, choose **Physical and External Domains > Physical Domains**.

3 In the Work pane, choose **Actions > Create Physical Domain**.

4 In the **Create Physical Domain** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Select the VLAN pool you just created.

XML Object

```
 <physDomP childAction="" configIssues="" dn="uni/phys-bsprint-PHY" lcOwn="local"
modTs="2015-02-23T16:13:21.906-08:00" monPolDn="uni/fabric/monfab-default"
name="bsprint-PHY" ownerKey="" ownerTag="" status="" uid="8131">
     <infraRsVlanNs childAction="" forceResolve="no" lcOwn="local" modTs="2015-02-
23T16:13:22.065-08:00" monPolDn="uni/fabric/monfab-default" rType="mo" rn="rsvlanNs"
state="formed" stateQual="none" status="" tCl="fvnsVlanInstP" tDn="uni/infra/vlanns-
[bsprint-vlan-pool]-static" tType="mo" uid="8131"/>
     <infraRsVlanNsDef childAction="" forceResolve="no" lcOwn="local" modTs="2015-02-
23T16:13:22.065-08:00" rType="mo" rn="rsvlanNsDef" state="formed" stateQual="none"
status="" tCl="fvnsAInstP" tDn="uni/infra/vlanns-[bsprint-vlan-pool]-static"
tType="mo"/>
     <infraRtDomP childAction="" lcOwn="local" modTs="2015-02-23T16:13:52.945-08:00"
rn="rtdomP-[uni/infra/attentp-bsprint-AEP]" status="" tCl="infraAttEntityP"
tDn="uni/infra/attentp-bsprint-AEP"/>
 </physDomP>
```

Create an Attachable Access Entity Profile (AEP)

The AEP links the physical domain and its VLAN Pool to the interface policies. The configuration for an AEP is straightforward.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Global Policies > Attached Acess Entity Profle**.
3. In the Work pane, choose **Actions > Create Attached Entity Profile**.
4. In the **Create Attached Entity Profile** dialog box, perform the following actions:
   a. Define the a meaningful name for the profile.
   b. Optionally, enter a description.
   c. Click **+** to associate the domain to the AEP.
   d. Select the physical domain that was previously configured.
5. Click **Next**.
6. Click **Submit**.

XML Object

```
 <infraAttEntityP childAction="" configIssues="" descr="" dn="uni/infra/attentp-
bsprint-AEP" lcOwn="local" modTs="2015-02-23T16:13:52.874-08:00"
monPolDn="uni/fabric/monfab-default" name="bsprint-AEP" ownerKey="" ownerTag=""
status="" uid="8131">
     <infraContDomP childAction="" lcOwn="local" modTs="2015-02-23T16:13:52.874-08:00"
rn="dompcont" status="">
         <infraAssocDomP childAction="" dompDn="uni/phys-bsprint-PHY" lcOwn="local"
modTs="2015-02-23T16:13:52.961-08:00" rn="assocdomp-[uni/phys-bsprint-PHY]" status=""/>
         <infraAssocDomP childAction="" dompDn="uni/l2dom-JC-L2-Domain" lcOwn="local"
modTs="2015-02-25T11:35:33.570-08:00" rn="assocdomp-[uni/l2dom-JC-L2-Domain]"
status=""/>
     </infraContDomP>
     <infraContNS childAction="" lcOwn="local" modTs="2015-02-23T16:13:52.874-08:00"
monPolDn="uni/fabric/monfab-default" rn="nscont" status="">
         <infraRsToEncapInstDef childAction="" deplSt="" forceResolve="no"
lcOwn="local" modTs="2015-02-23T16:13:52.961-08:00" monPolDn="uni/fabric/monfab-
default" rType="mo" rn="rstoEncapInstDef-[allocencap-[uni/infra]/encapnsdef-
[uni/infra/vlanns-[bsprint-vlan-pool]-static]]" state="formed" stateQual="none"
status="" tCl="stpEncapInstDef" tDn="allocencap-[uni/infra]/encapnsdef-
[uni/infra/vlanns-[bsprint-vlan-pool]-static]" tType="mo">
             <fabricCreatedBy childAction="" creatorDn="uni/l2dom-JC-L2-Domain"
deplSt="" domainDn="uni/l2dom-JC-L2-Domain" lcOwn="local" modTs="2015-02-
25T11:35:33.570-08:00" monPolDn="uni/fabric/monfab-default" profileDn="" rn="source-
[uni/l2dom-JC-L2-Domain]" status=""/>
             <fabricCreatedBy childAction="" creatorDn="uni/phys-bsprint-PHY" deplSt=""
domainDn="uni/phys-bsprint-PHY" lcOwn="local" modTs="2015-02-23T16:13:52.961-08:00"
monPolDn="uni/fabric/monfab-default" profileDn="" rn="source-[uni/phys-bsprint-PHY]"
status=""/>
         </infraRsToEncapInstDef>
     </infraContNS>
     <infraRtAttEntP childAction="" lcOwn="local" modTs="2015-02-24T11:59:37.980-08:00"
rn="rtattEntP-[uni/infra/funcprof/accportgrp-bsprint-AccessPort]" status=""
tCl="infraAccPortGrp" tDn="uni/infra/funcprof/accportgrp-bsprint-AccessPort"/>
     <infraRsDomP childAction="" forceResolve="no" lcOwn="local" modTs="2015-02-
25T11:35:33.570-08:00" monPolDn="uni/fabric/monfab-default" rType="mo" rn="rsdomP-
[uni/l2dom-JC-L2-Domain]" state="formed" stateQual="none" status="" tCl="l2extDomP"
tDn="uni/l2dom-JC-L2-Domain" tType="mo" uid="8754"/>
     <infraRsDomP childAction="" forceResolve="no" lcOwn="local" modTs="2015-02-
23T16:13:52.961-08:00" monPolDn="uni/fabric/monfab-default" rType="mo" rn="rsdomP-
[uni/phys-bsprint-PHY]" state="formed" stateQual="none" status="" tCl="physDomP"
tDn="uni/phys-bsprint-PHY" tType="mo" uid="8131"/>
 </infraAttEntityP>
```

Create Interface Policies

Next, define the interface profiles and showcase the re-usability of the fabric poli-

cies. Interface policies can be re-used as needed by different interface profile definition requirements. This section will illustrate creation of new profiles, but ideally there are already policies in place that can simply be selected.

Create Link Level Policies

Link level policies dictate configuration like the speed of ports.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > Link Level**.
3. In the Work pane, choose **Actions > Create Link Level Policy**.
4. In the **Create Link Level Policy** dialog box, perform the following actions:
   a. Define the meaningful name for the policy.
   b. Optionally, provide a description for the policy.
   c. Select the auto negotiation mode for the interface.
   d. Select the interface speed. Leaf switch ports default to 10GE.
   e. Change the de-bounce interval if required.
5. Click **Submit**.

XML Object

```
 <fabricHIfPol autoNeg="on" childAction="" descr="" dn="uni/infra/hintfpol-1G-Auto"
lcOwn="local" linkDebounce="100" modTs="2015-01-14T06:47:15.693-08:00" name="1G-Auto"
ownerKey="" ownerTag="" speed="1G" status="" uid="15374">
         <fabricRtHIfPol childAction="" lcOwn="local" modTs="2015-01-14T06:48:48.081-
08:00" rn="rtinfraHIfPol-[uni/infra/funcprof/accportgrp-UCS-1G-PG]" status=""
tCl="infraAccPortGrp" tDn="uni/infra/funcprof/accportgrp-UCS-1G-PG"/>
         <fabricRtHIfPol childAction="" lcOwn="local" modTs="2015-02-25T11:48:11.331-
08:00" rn="rtinfraHIfPol-[uni/infra/funcprof/accportgrp-L3-Example]" status=""
tCl="infraAccPortGrp" tDn="uni/infra/funcprof/accportgrp-L3-Example"/>
 </fabricHIfPol>
```

Create a CDP Interface Policy

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > CDP Interface**.
3. In the Work pane, choose **Actions > Create CDP Interface Policy**.
4. In the **Create CDP Interface Policy** dialog box, perform the following actions:
   a. Define a meaningful name for the policy such as 'CDP-Enable'.

b. Optionally, provide a description for the policy.
c. Choose either admin state **enabled** or **disabled**.

5 Click **Submit**.

XML Object

```
 <cdpIfPol adminSt="enabled" childAction="" descr="" dn="uni/infra/cdpIfP-CDP-Enable"
lcOwn="local" modTs="2015-01-14T06:47:25.470-08:00" name="CDP-Enable" ownerKey=""
ownerTag="" status="" uid="15374">
         <cdpRtCdpIfPol childAction="" lcOwn="local" modTs="2015-01-14T07:23:54.957-
08:00" rn="rtinfraCdpIfPol-[uni/infra/funcprof/accportgrp-UCS-10G-PG]" status=""
tCl="infraAccPortGrp" tDn="uni/infra/funcprof/accportgrp-UCS-10G-PG"/>
         <cdpRtCdpIfPol childAction="" lcOwn="local" modTs="2015-02-24T14:59:11.154-
08:00" rn="rtinfraCdpIfPol-[uni/infra/funcprof/accbundle-ACI-VPC-IPG]" status=""
tCl="infraAccBndlGrp" tDn="uni/infra/funcprof/accbundle-ACI-VPC-IPG"/>
         <cdpRtCdpIfPol childAction="" lcOwn="local" modTs="2015-01-14T06:48:48.081-
08:00" rn="rtinfraCdpIfPol-[uni/infra/funcprof/accportgrp-UCS-1G-PG]" status=""
tCl="infraAccPortGrp" tDn="uni/infra/funcprof/accportgrp-UCS-1G-PG"/>
         <cdpRtCdpIfPol childAction="" lcOwn="local" modTs="2015-02-24T11:59:37.980-
08:00" rn="rtinfraCdpIfPol-[uni/infra/funcprof/accportgrp-bsprint-AccessPort]"
status="" tCl="infraAccPortGrp" tDn="uni/infra/funcprof/accportgrp-bsprint-
AccessPort"/>
         <cdpRtCdpIfPol childAction="" lcOwn="local" modTs="2015-02-25T11:48:11.331-
08:00" rn="rtinfraCdpIfPol-[uni/infra/funcprof/accportgrp-L3-Example]" status=""
tCl="infraAccPortGrp" tDn="uni/infra/funcprof/accportgrp-L3-Example"/>
 </cdpIfPol>
```

Create an LLDP Interface Policy

1 On the menu bar, choose **Fabric > Access Policies**.

2 In the Navigation pane, choose **Interface Policies > Policies > LLDP Interface**.

3 In the Work pane, choose **Actions > Create LLDP Interface Policy**.

4 In the **Create LLDP Interface Policy** dialog box, perform the following actions:
a. Define a meaningful name for the policy.
b. Optionally, provide a description for the policy.
c. Choose the receive state.
d. Choose the transmit state.

5 Click **Submit**.

XML Object

```
 <lldpIfPol adminRxSt="enabled" adminTxSt="enabled" childAction=""
dn="uni/infra/lldpIfP-LLDP-Enable" lcOwn="local" modTs="2015-02-11T07:40:35.664-08:00"
name="LLDP-Enable" ownerKey="" ownerTag="" status="" uid="15374">       status=""
         <lldpRtLldpIfPol childAction="" lcOwn="local" modTs="2015-02-24T14:59:11.154-
08:00" rn="rtinfraLldpIfPol-[uni/infra/funcprof/accbundle-ACI-VPC-IPG]"
tCl="infraAccBndlGrp" tDn="uni/infra/funcprof/accbundle-ACI-VPC-IPG"    status=""
         <lldpRtLldpIfPol childAction="" lcOwn="local" modTs="2015-02-25T11:48:11.331-
08:00" rn="rtinfraLldpIfPol-[uni/infra/funcprof/accportgrp-L3-Example]"
tCl="infraAccPortGrp" tDn="uni/infra/funcprof/accportgrp-L3-Example"
 </lldpIfPol>
```

Create an LACP Interface Policy

lows
faces. These can be
ration requirements

Link Aggregation Control Protocol is part of an IEEE specification (802.3ad) that al
an optional negotiation protocol to be run on Port Channel inter
pre-emptively defined to be used as needed for the various configu
in the data center.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > LACP**
3. In the Work pane, choose **Actions > Create LACP Policy**.
4. In the **Create LACP Policy** dialog box, perform the following actions:
   a. Define a meaningful name for the policy.
   b. Optionally, provide a description for the policy.
   c. Select the LACP mode required for the server. Note if LACP is enabled on the leaf switch, LACP must also be enabled on the server or other connected device.
   d. Optionally, specify the minimum and maximum number of links in the Port Channel.
5. Click **Submit**.

XML Object

```
 <lacpLagPol childAction="" ctrl="fast-sel-hot-stdby,graceful-conv,susp-individual"
descr="" dn="uni/infra/lacplagp-LACP-Active" lcOwn="local" maxLinks="16" minLinks="1"
modTs="2015-02-24T11:58:36.547-08:00" mode="active" name="LACP-Active" ownerKey=""
ownerTag="" status="" uid="8131">
         <lacpRtLacpPol childAction="" lcOwn="local" modTs="2015-02-24T14:59:11.154-
08:00" rn="rtinfraLacpPol-[uni/infra/funcprof/accbundle-ACI-VPC-IPG]" status=""
tCl="infraAccBndlGrp" tDn="uni/infra/funcprof/accbundle-ACI-VPC-IPG"/>
 </lacpLagPol>
```

Create an LACP Member Profile (optional)

Optionally, the LACP member profile provides the ability to provide priority specifications to members of an LACP group.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > LACP Member**.
3. In the Work pane, choose **Actions > Create LACP Member Policy**.
4. In the **Create LACP Member Policy** dialog box, perform the following actions:
   a. Define a meaningful name for the policy.
   b. Optionally, provide a description for the policy.
   c. If required, change the priority.
   d. If required, change the transmit rate.
5. Click **Submit**.

Create a Spanning Tree Interface Policy (optional)

The Spanning Tree policy dictates the behavior of southbound leaf port Spanning Tree features. It is a common best practice to enable BPDU guard on interfaces connected to servers.

Note: ACI does not run Spanning Tree on the fabric between the leaves and spines. The Spanning Tree interface policy simply defines the port behavior.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > Spanning Tree Interface**.
3. In the Work pane, choose **Actions > Create Spanning Tree Interface Policy**.

4 In the **Create Spanning Tree Interface Policy** dialog box, perform the following actions:
   a. Define a meaningful name for the policy.
   b. Optionally, provide a description for the policy.
   c. Enable BPDU filter and/or BPDU guard.

5 Click **Submit**.

Create a Storm Control Policy (optional)

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. The traffic storm control feature can be used to prevent disruptions on ports by a broadcast, multicast, or unicast traffic storm on physical interfaces.

1 On the menu bar, choose **Fabric > Access Policies**.

2 In the Navigation pane, choose **Interface Policies > Policies > Storm Control**.

3 In the Work pane, choose **Actions > Create Storm Control Policy**.

4 In the **Create Storm Control Policy** dialog box, perform the following actions:
   a. Define a meaningful name for the policy.
   b. Optionally, provide a description for the policy.
   c. Specify how the control policy is to be applied, either through percentage of the total bandwidth or as a packet per second definition that matches the requirement for the data center

5 Click **Submit**.

Creating a Layer 2 Interface Policy to enable per port-VLAN

1 On the menu bar, choose **Fabric > Access Policies**.

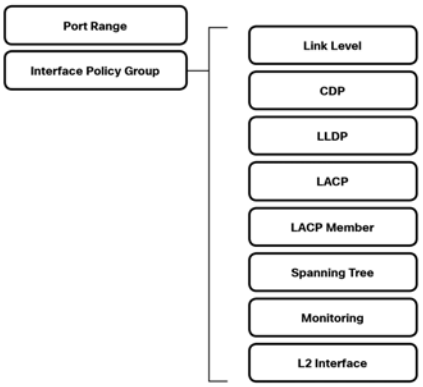
2 In the Navigation pane, choose **Interface Policies > Policies > L2 Interface**.

3 In the Work pane, choose **Actions > Create L2 Interface Policy**.

4 In the **Create L2 Interface Policy** dialog box, perform the following actions:
   a. Give the L2 Interface name and an optional description.
   b. Select VLAN scope to Port Local scope to enable per port-VLAN.

Create Interface Policy Groups

The interface policy groups comprise the interface policies as a functional group that is associated to an interface. The following diagram shows how previously created items are grouped under the policy group.



Policies contained in a policy group

Once all the interface policies have been defined, the individual policies can be brought together to form a policy group that will be linked to the interface profile. The policy group is defined from a master definition that encompasses being one of the following:

- Access Policy Group
- Port Channel Policy Group
- VPC Policy Group

Create Access Port Policy Group

The access port policy is defined for an individual link (non-Port Channel or vPC).

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policy Groups**.
3. In the Work pane, choose **Actions > Create Access Policy Group**.
4. In the **Create Access Policy Group** dialog box, perform the following actions:
   a. Define a meaningful name for the policy group.
   b. Optionally, provide a description for the policy group.
   c. Use the profiles created previously that are relevant for this policy group.
5. Click **Submit**.

Create Port Channel Interface Policy Group

Port Channeling also load-balances traffic across the physical interfaces that are members of the channel group. For every group of interfaces that needs to be configured into a port channel, a different policy group has to be created. This policy group defines the behaviour. For example, if ports 1/1-4 are to be configured into one port channel, and ports 1/5-8 into a separate port channel, each of those groups would require the creation of a separate policy group.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policy Groups**.
3. In the Work pane, choose **Actions > Create PC Interface Policy Group**.
4. In the **Create PC Interface Policy Group** dialog box, perform the following actions:
   a. Define a meaningful name for the policy group.
   b. Optionally, provide a description for the policy group.
   c. Select the policies created previously that are relevant for this PC policy group.
5. Click **Submit**.

Create VPC Interface Policy Group

Note: This object must be unique for each VPC created.

A virtual PortChannel (vPC) allows links that are physically connected to two different devices to appear as a single Port Channel to a third device. In the world of ACI, pairs of leaf switches may be configured in a vPC domain so that downstream devices can be active-active dual-homed.

For every group of interfaces that are to be configured into a vPC, a different interface policy group needs to be created. The vPC policy group contains both the definition for the behaviour of the port channel, and the identifier. For example, if ports 1/1-4 are to be configured into one vPC across two switches, and ports 1/5-8 into a separate vPC across two switches, each of those groups would require the definition of a separate policy group.

Note: For vPC you will also require a unique vPC domain definition between the two paired switches. More details to follow.

1. On the menu bar, choose **Fabric > Access Policies**.

2 In the Navigation pane, choose **Interface Policies > Policy Groups**.

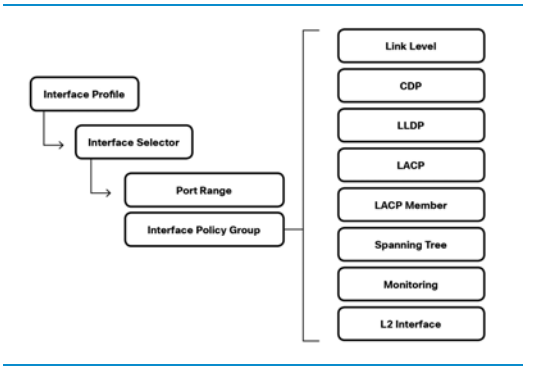
3 In the Work pane, choose **Actions > Create VPC Interface Policy Group**.

4 In the **Create VPC Interface Policy Group** dialog box, perform the following actions:
   a. Define a meaningful name for the policy group.
   b. Optionally, provide a description for the policy group.
   c. Choose the policies created previously that are relevant for this vPC policy group.

5 Click **Submit**.

Interface Profile

The interface profile in ACI links the policy groups that define how the interface is going to behave, and assigns them to specific ports via the concept of interface selector. In turn, the interface profile is eventually tied to a switch profile to specify which leaf switch the referenced ports should be configured. As we continue the process of defining the port profiles, you can observe how we have started at the bottom of this object tree configuring the different profiles. The purposes for all these individual policies that tie together is to maximize policy re-use.



Interface Profile links to Interface Selector and Interface Policy Group

The diagram in the previous section provides a visual description of what can be accomplished by grouping the policies that have been defined under the interface profile, and then assigned to ports with interface selectors and the access port policy groups.

Create Interface Profile

The interface profile is composed of two primary objects. The interface selector and the access port policy group. The interface selector defines what interfaces will apply the access port policy. The ports that share the same attributes can then be grouped under the same interface profile.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Profiles**.
3. In the Work pane, choose **Actions > Create Interface Profile**.
4. In the **Create Interface Profile** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Optionally, provide a description for the profile.
5. Click **Submit**.

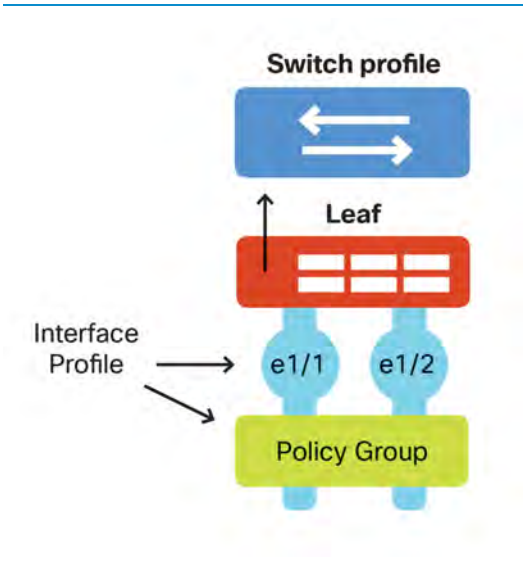
Next, add the interface selectors that are associated to this interface profile.

Create Interface Selector

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Profiles > *Name_of_Interface_Profile_Created***.
3. In the Work pane, choose **Actions > Create Access Port Selector**.
4. In the **Create Access Port Selector** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Optionally, provide a description for the profile.
   c. Enter the interface IDs.
   d. Choose the interface policy group that should be associated to these ports.
5. Click **Submit**.

Create Interface Profile for Port Channel

If a server has two or more uplinks to a leaf switch, the links can be bundled into a Port Channel for resiliency and load distribution. In order to configure this in ACI, create an interface policy group of type Port Channel to bundle the interfaces. Different Port Channels require different policy groups.



Port Channel Policy Group

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Profiles**.
3. In the Work pane, choose **Actions > Create Interface Profile**.
4. In the **Create Interface Profile** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Optionally, provide a description for the profile.
5. Click **Submit**.

Next, create an interface port selector. Since you will be configuring a Port Channel, the operator will add all of the interfaces required in the Port Channel interface. In this example interfaces Ethernet 1/1-2 will be configured in one Port Channel and interfaces Ethernet 1/3-4 will be configured in another Port Channel.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Profiles > *Name_of_Interface_Profile_Created***.
3. In the Work pane, choose **Actions > Create Access Port Selector**.

4 In the **Create Access Port Selector** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Optionally, provide a description for the profile.
   c. Enter interface IDs for the first port channel.
   d. Choose the interface policy group.

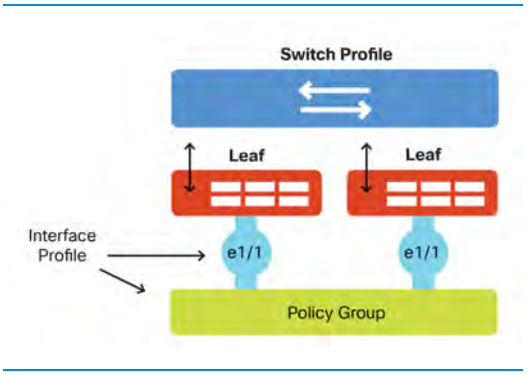
5 Click **Submit**.

6 Repeat this process for the second Port Channel (if you have another Port Channel to add).

Create an Interface Profile for Virtual Port Channel

A vPC domain is always made up of two leaf switches, and a leaf switch can only be a member of one vPC domain. In ACI, that means that the definition of the policies is significant between the two switches. The same policy can be reused betweeen the two switches, and through the vPC domain the pair association can be defined. vPC Switch domain members should be taken into consideration when configuring firmware maintenance groups. By keeping this in mind, firmware upgrades should never impact both vPC switch peers at the same time. More details on this can be found in the Upgrading and Downgrading Firmware section.

For this reason, a switch profile that would represent two separate switch IDs needs to be created. The relationship of these switches to the two ports in the same policy group is defined through the interface profile.



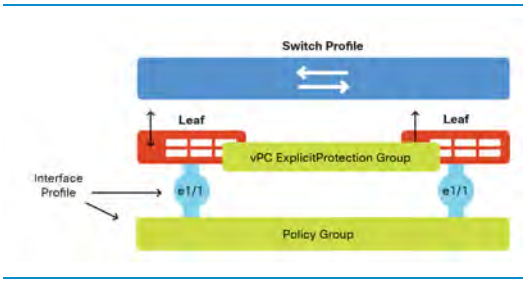
vPC Policy Group

The same process would have to be repeated for every grouped interface on each side that will be a member of the vPC.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Profiles**.
3. In the Work pane, choose **Actions > Create Interface Profile**.
4. In the **Create Interface Profile** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Optionally, provide a description for the profile.
5. Click **Submit**.
6. In the Navigation pane, choose **Interface Policies > Profiles > *Name_of_Interface_Profile_Created***.
7. In the Work pane, choose **Actions > Create Access Port Selector**.
8. In the **Create Access Port Selector** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Optionally, provide a description for the profile.
   c. Enter interface IDs.
   d. Select of the interface policy group to be used for the vPC port behavior.
9. Click **Submit**.

Create a vPC Domain for Virtual Port Channel

When configuring a vPC, there is one additional step to be configured once to put two leaf switches into the same vPC domain.



Creating a vPC Domain

1. On the menu bar, choose **Fabric > Access Policies**.

2 In the Navigation pane, choose **Switch Policies > VPC Domain > Virual Port Channel default**.

3 In the Work pane, choose **Actions > Explicit VPC Protection Group**.

4 In the **Explicit VPC Protection Group** dialog box, perform the following actions:
   a. Define a meaningful name for the vPC domain.
   b. Provide a unique ID to represent the vPC domain.
   c. Choose the first switch you want to be part of the vPC domain.
   d. Choose the second switch you want to be part of the vPC domain.

5 Click **Submit**.

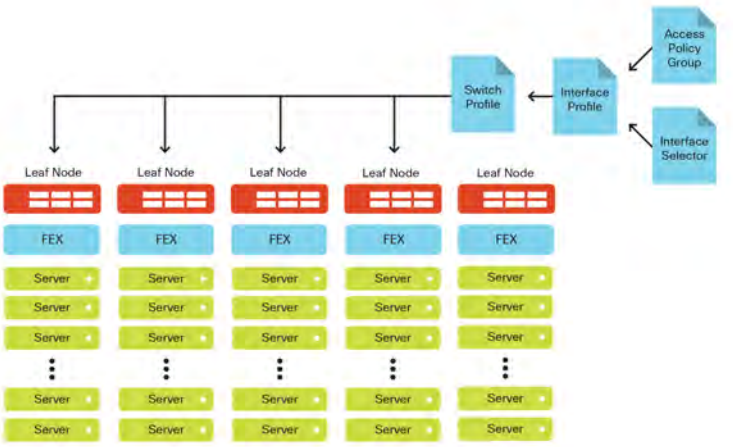
Switch Profiles

A switch profile groups all the interface profiles that define the behavior of its respective switch ports. A switch profile could be the definition of a single switch or it could be the definition of multiple switches. As a best practice, there should be a switch profile for each leaf switch, and an additional switch profile for each vPC domain pair of leaf switches.

The interface profiles that you have created can be associated to the switch through a single switch profile or they can be associated through different switch profiles. If you have various racks that are identical in the way the interface ports are configured, it could be beneficial to utilize the same switch profile. This would make it possible to modify the configuration of many switches during operations without having to configure each switch individually.

Reusability

The capability of policy reusability is crucial to re-emphasize from an operational perspective. If a profile has been defined to configure a port as 1GB speed for example, that profile can be reused for many interface policy groups. When looking at whole switch configurations, the re-usability of the profile can be extended to simplify data center operations and ensure compliance. The following figure illustrates the reusability of profiles across racks of switches.



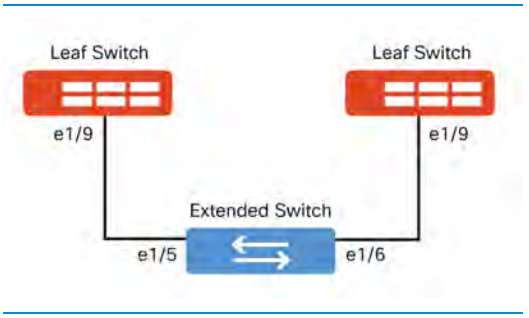
Policy re-use at scale

In the previous diagram, each of the top of rack switches is based on the same switch profile. If all these racks are configured in the same fashion (meaning they are wired in the same way) the same policies could be reused by simply assigning the switches to the same switch profile. It would then inherit the profile tree and be configured the exact same way as the other racks.

It is also important to be aware of the implication of deleting profiles. If a profile has been reused across many devices, make sure to check where it is being used before you delete the profile or policy.

Sample vPC Creation

The following procedure demonstrates what a vPC bringup looks like and also API POST configuration assesment of the vPC. The following topology will be configured:



Sample Topology

Create VLAN Pools

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Pools > VLAN**.
3. In the Work pane, choose **Actions > Create VLAN Pool**.
4. In the **Create VLAN Pool** dialog box, perform the following actions:
   a. Define a meaningful name for the pool.
   b. Optionally, provide a description for the pool.
   c. Click **Static Allocation** for the allocation mode.
   Note: For this example the pool will be from VLAN 100 to VLAN 199.

```
REST :: /api/node/class/fvnsVlanInstP.xml
CLI :: moquery -c fvnsVlanInstP
```

Create a Physical Domain

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Physical and External Domains > Physical Domains**.
3. In the Work pane, choose **Actions > Create Physical Domain**.
4. In the **Create Physical Domain** dialog box, perform the following actions:
   a. Define a meaningful name for the domain.
   b. Choose the VLAN pool that you previously created.

```
REST :: /api/node/class/physDomP.xml
CLI :: moquery -c physDomP
```

Create Access Entity Profile

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Global Policies > Attachable Access Entity Profles**.
3. In the Work pane, choose **Actions > Create Attached Entity Profile**.
4. In the **Create Attached Entity Profile** dialog box, perform the following actions:
   a. Define a meaningful name for the AEP.
   b. Provide a unique ID to represent the AEP.
   c. Click **+** to add a domain that will be associated to the interfaces.
   d. Choose the physical domain that you previously created.
5. Click **Next**.
6. Click **Submit**.

```
REST :: /api/node/class/infraAttEntityP.xml
CLI :: moquery -c infraAttEntityP
```

Interface Policies

Create Link Level Policy

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > Link Level**.
3. In the Work pane, choose **Actions > Create Link Level Policy**.
4. In the **Create Link Level Policy** dialog box, perform the following actions:
   a. Define a meaningful name for the pool.
   b. Optionally, provide a description for the policy.
   c. Choose the auto negotiation for the interface.
   d. Choose the speed to match the interface requirement.
5. Click **Submit**.

```
REST :: /api/node/class/fabricHIfPol.xml
CLI :: moquery -c fabricHIfPol
```

Create LACP Policy

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policies > LACP**.
3. In the Work pane, choose **Actions > Create LACP Policy**.
4. In the **Create LACP Policy** dialog box, perform the following actions:
   a. Define a meaningful name for the policy.
   b. Optionally, provide a description for the pool.
   c. In mode click on Active.
5. Click **Submit**.

```
REST :: /api/node/class/lacpLagPol.xml
CLI :: moquery -c lacpLagPol
```

Create vPC Interface Policy Group

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Policy Groups**.
3. In the Work pane, choose **Actions > Create vPC Interface Policy Group**.
4. In the **Create vPC Interface Policy Group** dialog box, perform the following actions:
   a. Define a meaningful name for the policy group.
   b. Optionally, provide a description for the policy group.
   c. Choose a Link Level Policy.
   d. Choose an LACP Policy.
      Note: LACP is recommended for vPCs. However, ensure LACP is configured on the device connected to the leaf switch.
   e. Choose an AEP to associate the policy group to.
5. Click **Submit**.

```
REST :: /api/node/class/infraAccBndlGrp.xml
CLI :: moquery -c infraAccBndlGrp
```

Create an Interface Profile

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Interface Policies > Profiles**.
3. In the Work pane, choose **Actions > Create Interface Profile**.
4. In the **Create Interface Profile** dialog box, perform the following actions:
   a. Define a meaningful name for the policy group.
   b. Optionally, provide a description for the policy group.
5. Click **Submit**.
6. In the Navigation pane, choose **Interface Policies > Profiles > Profiles > ACI-VPC-int-profile**.
7. In the Work pane, choose **Actions > Create Access Port Selector**.
8. In the **Create Access Port Selector** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Optionally, provide a description.
   c. Select the proper interfaces.
   d. Select an interface policy group.
9. Click **Submit**.

```
REST :: /api/node/class/infraAccPortP.xml
CLI :: moquery -c infraAccPortP
```

Create a Switch Profile

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Switch Policies > Profiles**.
3. In the Work pane, choose **Actions > Create Switch Profile**.
4. In the **Create Switch Profile** dialog box, perform the following actions:
   a. Define a meaningful name for the profile.
   b. Optionally, provide a description for the profile.
   c. In switch selectors click on the **+** symbol.
      i. Name: 103-104 (example node numbers).
      ii. Blocks: Select switch 103 and switch 104.

5 Click **Next.**

6 Select the previously created interface selector.

7 Click **Finish.**

```
REST :: /api/node/class/infraNodeP.xml
CLI :: moquery -c infraNodeP
```

Create vPC domain

1 On the menu bar, choose **Fabric > Access Policies**.

2 In the Navigation pane, choose **Switch Policies > VPC Domain > Virual Port Channel default**.

3 In the Work pane, choose **Actions > Explicit VPC Protection Group**.
   a. In the **Explicit VPC Protection Group** dialog box, perform the following actions:
   b. In the **Explicit VPC Protection Groups** section, click **+** to create a vPC protection group.
   c. Define a meaningful name for the vPC domain.
   d. Provide a unique ID for the vPC domain.
   e. Select the first switch ID that is part of this vPC pair: 103.
   f. Select the second switch ID that is part of this vPC pair: 104.

4 Click **Submit**.

```
REST :: /api/node/class/fabricExplicitGEp.xml
CLI :: moquery -c fabricExplicitGEp
```

Validate Operation of Configured vPC

1 On the menu bar, choose **Fabric > Inventory**.

2 In the Navigation pane, choose **POD 1 > Interfaces > vPC Interfaces**.

3 In the Work pane, there will be a table that shows the status of the vPC interface. If configured correctly, the status should be displayed and you should see successful establishment of the vPC domain.

You can also validate the operation of the vPC directly from the CLI of the switch itself.

If you connect to the console or the out of band management interface of the leaf node you should be able to see the status with the command show vpc.

```
Leaf-3# show vpc
Legend:
        (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 100
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : Disabled
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 1
Peer Gateway                      : Disabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Enabled (timeout = 240 seconds)
Operational Layer3 Peer           : Disabled

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1           up     -

vPC status
----------------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
1    Po1    up     success     success                    -
```

The following REST API call can be used to build vPCs and attach vPCs to static port bindings.

```
URL: https://{{apic-ip}}/api/policymgr/mo/.xml
 <polUni>
  <infraInfra>
    <!-- Switch Selector -->
    <infraNodeP name="switchProfileforVPC_201">
      <infraLeafS name="switchProfileforVPC_201" type="range">
        <infraNodeBlk name="nodeBlk" from_="201" to_="201"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-intProfileforVPC_201"/>
    </infraNodeP>
     <infraNodeP name="switchProfileforVPC_202">
      <infraLeafS name="switchProfileforVPC_202" type="range">
        <infraNodeBlk name="nodeBlk" from_="202" to_="202"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-intProfileforVPC_202"/>
    </infraNodeP>
         <!-- Interface Profile -->
    <infraAccPortP name="intProfileforVPC_201">
      <infraHPortS name="vpc201-202" type="range">
        <infraPortBlk name="vpcPort1-15" fromCard="1" toCard="1" fromPort="15"
toPort="15"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-intPolicyGroupforVPC"/>
      </infraHPortS>
    </infraAccPortP>
     <infraAccPortP name="intProfileforVPC_202">
      <infraHPortS name="vpc201-202" type="range">
        <infraPortBlk name="vpcPort1-1" fromCard="1" toCard="1" fromPort="1"
toPort="1"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-intPolicyGroupforVPC"/>
      </infraHPortS>
    </infraAccPortP>
         <!-- Interface Policy Group -->
    <infraFuncP>
      <infraAccBndlGrp name="intPolicyGroupforVPC" lagT="node">
        <infraRsAttEntP tDn="uni/infra/attentp-AttEntityProfileforCisco"/>
        <infraRsCdpIfPol tnCdpIfPolName="CDP_ON" />
        <infraRsLacpPol tnLacpLagPolName="LACP_ACTIVE" />
        <infraRsHIfPol tnFabricHIfPolName="10GigAuto" />
      </infraAccBndlGrp>
    </infraFuncP>
  </infraInfra>
</polUni>
  https://{{hostName}}/api/node/mo/uni.xml
 <polUni>
  <fvTenant descr="" dn="uni/tn-Cisco" name="Cisco" ownerKey="" ownerTag="">
      <fvAp descr="" name="CCO" ownerKey="" ownerTag="" prio="unspecified">
```

```
            <fvAEPg descr="" matchT="AtleastOne" name="Web" prio="unspecified">
                <fvRsPathAtt encap="vlan-1201" instrImedcy="immediate" mode="native"
tDn="topology/pod-1/protpaths-201-202/pathep-[vpc201-202]” />
            </fvAEPg>
            <fvAEPg descr="" matchT="AtleastOne" name="App" prio="unspecified">
                <fvRsPathAtt encap="vlan-1202" instrImedcy="immediate" mode="native"
tDn="topology/pod-1/protpaths-201-202/pathep-[vpc201-202]” />
            </fvAEPg>
        </fvAp>
    </fvTenant>
 </polUni>
```

Server Connectivity

Server connectivity is necessary for all application workloads to function properly on the Cisco Application Centric Infrastructure (ACI) fabric. The fabric connectivity requirements that are dictated by the server infrastructure must be carefully considered. In the case of Cisco Unified Computing System (UCS), fabric access policies must be provisoned to match these requirements. These policies are all governed by interface policy groups. ACME Inc. has several different models of servers in their data centers, such as Cisco UCS B and C series, as well as some third party servers that all need to be connected to the ACI fabric.

Cisco UCS B-Series Servers

If UCS B-series Fabric Interconnects are being connected to your ACI fabric and Cisco UCS, the Link Level Discovery Protocol (LLDP) must be disabled. Cisco Discovery Protocol (CDP) must be enabled on any ACI fabric interfaces that are connecting to UCS Fabric Interconnects. You can pre-provision these policies as part of the initial management tasks. See the Configuring Management Protocols chapter for more information.

When connecting UCS to the ACI fabric, the type of Layer 2 connection needed on the Fabric Interconnect facing ports must be determined first. A best practice is to leverage a virtual private cloud (vPC) to connect the UCS environment so as to create a multi-chassis etherchannel. In this scenario, individual link and fabric switch failures are mitigated to maintain a higher expected up time.

For more information on the process needed to configure links to UCS as either a vPC or a traditional port channel, see the Adding New Devices to the Fabric section.

Standalone Rack Mount Servers or Non-Cisco Servers

Any non-UCS server architecture can also be connected directly to the ACI fabric or to a Cisco Nexus 2000 Fabric Extender (FEX). When being connected to the ACI fabric, the kind of traffic expected out of the server links needs to be determined. If the workload is a bare metal server, traffic can be classified on a per port basis and associated AEPs and EPGs can be mapped appropriately to match the encapsulated traffic. If a supported hy-

pervisor is to be used, a Virtual Machine Manager (VMM) domain must be properly configured, and then associated with the corresponding ports on the fabric as a hypervisor that is facing through EPG and AEP mapping. The key is to map the expected traffic classification to the ports that are connected to the server infrastructure.

Utilizing a FEX is an alternative way to connect host devices into the ACI fabric. Restrictions that are present in NX-OS mode such that non-host-facing ports are not supported, are still true. Ports must only be connected to hosts, and connectivity to any other network device will not function properly. When utilizing a FEX, all host-facing ports are treated the same way as if they were directly attached to the ACI fabric.

Virtual Machine Networking

Understanding VM Networking in ACI

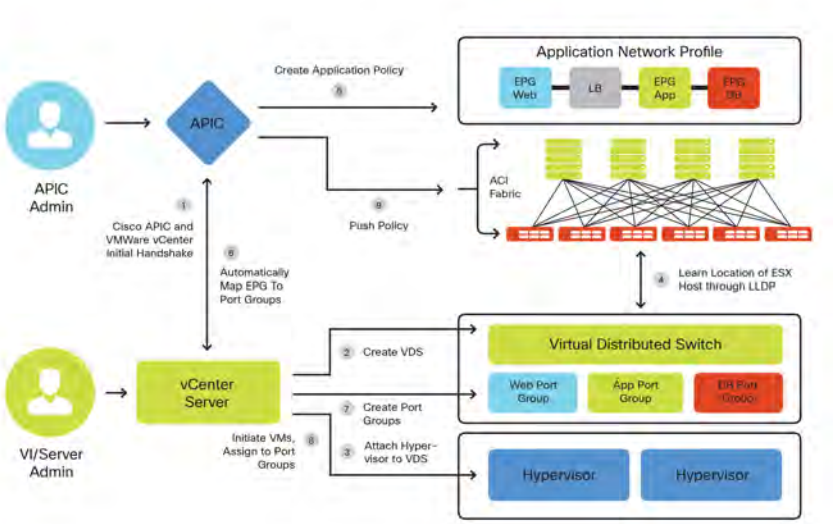
One of the most common uses of the Cisco Application Centric Infrastructure (ACI) will be to help manage and deploy applications in virtual environments. The ACI provides the ability to manage both virtual and physical endpoints with the same set of policies. This chapter will look at various operational tasks that will be performed throughout the daily operations.

The following list describes some virtual machine manager (VMM) system terms:

- A virtual machine controller is an external VMM entity, such as VMware vCenter, VMware vShield, and Microsoft Systems Center Virtual Machine Manager (SCVMM). The Application Policy Infrastructure Controller (APIC) communicates with the VMM to publish network policies that are applied to virtual workloads. A virtual machine controller administrator provides an APIC administrator with a virtual machine controller authentication credential; multiple controllers of the same type can use the same credential.
- Credentials represent the authentication credentials to communicate with virtual machine controllers. Multiple controllers can use the same credentials.
- A pool represents a range of traffic encapsulation identifiers, such as VLAN and VXLAN IDs, and multicast addresses. A pool is a shared resource and can be consumed by multiple domains, such as VMM and Layer 4 to Layer 7 services. A leaf switch does not support overlapping VLAN pools. Different overlapping VLAN pools must not be associated with the same attachable entity profile (AEP).
- The two types of VLAN-based pools are as follows:
    - Dynamic pools - Managed internally by the APIC to allocate VLANs for endpoint groups (EPGs). A VMware vCenter domain can associate only to a dynamic pool. This is the pool type that is required for VMM integration.
    - Static pools - The EPG has a relation to the domain, and the domain has a relation to the pool. The pool contains a range of encapsulated VLANs and VXLANs. For static EPG deployment, the user defines the interface and the encapsulation. The encapsulation must be within the range of a pool that is associated with a domain with which the EPG is associated.

When creating dynamic VLAN pools for VMM integration, the VLAN range must also be created on any intermediate devices, such as traditional switches or blade switches. This includes creating the VLANs on **Unified Computing System** (**UCS**).

ACI VM Integration Workflow



ACI VM Integration Workflow

For detailed information on how to deploy the VMware vSphere Distributed Switch with the Cisco APIC, see the following documents:

- *Cisco APIC Getting Started Guide*

VMware Integration

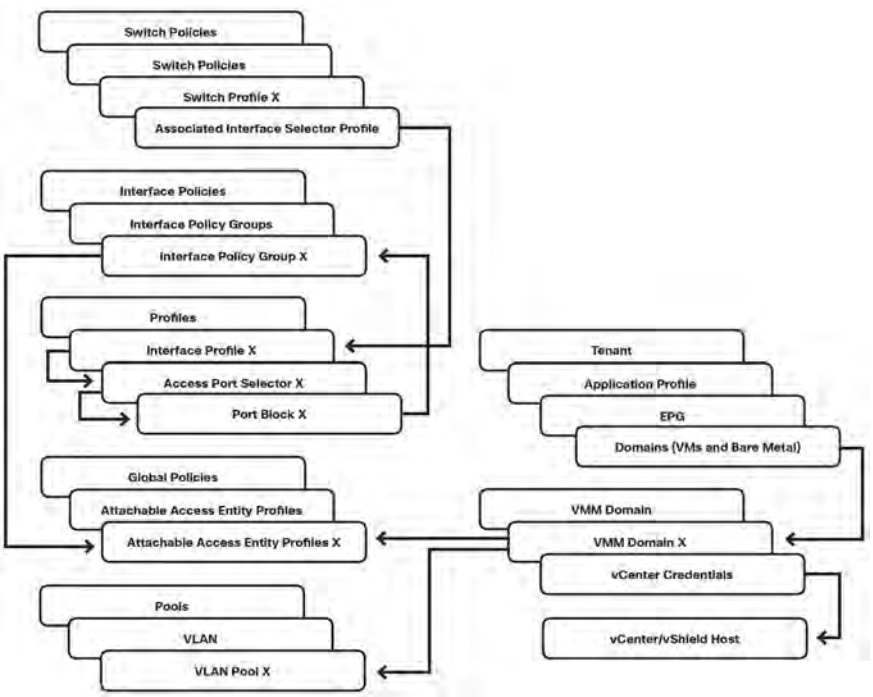
When integrating ACI into your VMware infrastructure you have two options for deploying networking. VMware domains can be deployed, leveraging the VMware vSphere Distributed Virtual Switch (DVS) or the Cisco Application Virtual Switch (AVS). Both provide similar basic virtual networking functionality; however, the AVS provides

additional capabilities, such as VXLAN and microsegmentation support. ACME Inc. has chosen to leverage the additional features provided by AVS. For organizations interested in using the standard DVS provided by VMware, please refer to the following documents:

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/getting-started/video/cisco_apic_create_vcenter_domain_profile_using_gui.html

VMM Policy Model Interaction

Shown below are some of the various ACI policies which are involved with setting up VM Integration. This serves as a reference for the ways the various policies are related to each other.



VMM Policy Model Interaction

Publishing EPGs to a VMM Domain

This section will detail how to publish an existing endpoint group (EPG) to a Virtual Machine Manager (VMM) domain. For details on how to create EPGs, see the Tenants section.

For an EPG to be pushed to a VMM domain, a domain binding within the tenant EPG must be created.

1. On the menu bar, choose **Tenants** > **ALL TENANTS**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane, choose ***Tenant_Name*** **> Application Profiles >** ***Application_Profile_Name*** **> Application EPGs >** ***Application_EPG_Name*** **> Domains (VMs and Bare-Metals)**.
4. In the Work pane, choose **Actions > Add VM Domain Association**.
5. In the **Add VM Domain Association** dialog box, choose the VMM Domain Profile that you previously created.
   a. For the **Deployment & Resolution Immediacy**, Cisco recommends keeping the default option of **On Demand**. This provides the best resource usage in the fabric by only deploying policies to Leaf nodes when endpoints assigned to this EPG are connected. There is no communication delay or traffic loss by keeping the default selections.
6. Click **Submit**.
   Note: The EPG will now be available as a Port Group to your VMM.

Connecting VMs to the EPG Port Groups on vCenter

1. Connect to your vCenter using the VI Client.
2. From the Host and Clusters view, right click on your Virtual Machine and select "Edit Settings".
3. Click on the Network Adapter, and in the Network Connection dropdown box select the Port Group which corresponds to your EPG. It should display in the format of
   [TENANT|APPLICATION_PROFILE|EPG|VMM_DOMAIN_PROFILE]

If you do not see your ACI EPG in the Network Connection list, it means one of the following:

- The VM is running on a host which is not attached to the Distributed Switch managed by the APIC.
- There may be a communication between your APIC and vCenter either through the OOB or INB management network.

Verifying Virtual Endpoint Learning

Once the VMs are connected to the appropriate port group/EPG, you should verify the APIC has learned your virtual end point.

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane, choose ***Tenant_Name*** **> Application Profiles >** ***Application_Profile_Name*** **> Application EPGs >** ***Application_EPG_Name***.
4. In the Work pane, choose the **Operational** tab.
   Note: The current tab should display CLIENT ENDPOINTS. All endpoints either virtual or physical will be displayed. From here you should be able to find your Virtual Machine by filtering the "Learning Source" column for rows with values of "Learned VMM".

Verifying VM Endpoint Learning on the APIC from the CLI

You can verify the same info as above from the CLI by using the 'moquery' (Managed Object Query) command and adding two filters. One for the Distinguished Name (DN) name of your EPG, and one for the Class Name of 'fvCEp' (Fabric Vector Client Endpoint)

```
moquery -c fvCEp --dn uni/tn-<TENANT_NAME>/ap-<APP_PROFILE_NAME>/epg-<EPG_NAME>
```

You can determine the DN of your EPG by right clicking on the EPG in the GUI, selecting "Save As" and looking at the XML object. From this file you will see the DN entry for the particular EPG:

```
<imdata totalCount="1"><fvAEPg uid="15374" triggerSt="triggerable" status=""
  scope="2588672" prio="unspecified" pcTag="49159" name="epg-od"
  monPolDn="uni/tn-common/monepg-default" modTs="2015-02-06T06:46:24.729+11:00"
  matchT="AtleastOne" lcOwn="local" dn="uni/tn-mb-tennant1/ap-mb-app-pro/epg-epg-od"
  descr="" configSt="applied" configIssues="" childAction=""/></imdata>
```

Next, use this DN with the moquery to return the list of client Enpoints for this EPG:

```
admin@apic1:~> moquery -c fvCEp --dn uni/tn-mb-tennant1/ap-mb-app-pro/epg-epg-od
Total Objects shown: 1

# fv.CEp
name          : 00:50:56:BB:8C:6A
childAction   :
dn            : uni/tn-mb-tennant1/ap-mb-app-pro/epg-epg-od/cep-00:50:56:BB:8C:6A
encap         : vlan-211
id            : 0
idepdn        :
ip            : 10.10.10.10
lcC           : learned,vmm
lcOwn         : local
mac           : 00:50:56:BB:8C:6A
mcastAddr     : not-applicable
modTs         : 2015-02-06T06:48:52.229+11:00
rn            : cep-00:50:56:BB:8C:6A
status        :
uid           : 0
uuid          :
```

VMware Integration Use Case

A VMWare administrator in ACME requests the network team to trunk a set of VLANs down to the ESX hosts to provide connectivity to their DVS switches. Rather than trunking VLANs on a per server basis, the network team decides to leverage a new methodology to be more agile and leverage the on-demand provisioning of resources where and when they are needed, as well as providing unlimited Layer 2 mobility for all the VM hosts within the ACI fabric.

To do so, the network admins work with the VMware admins to decide on a range of VLANs that will be provided dynamically by APIC to the ESX hosts that need them. They decide on an unused VLAN range of (600 - 800). This is their dynamic VLAN pool. Once this is decided, the APIC administrator proceeds to configure VMM integration in the APIC GUI by providing the vCenter credentials to APIC. APIC dynamically provisions all EPGs and makes them available to the ESX hosts as a port-group.

Note: The APIC does not automatically move VMNICs into the port-group. This allows VMware admins to maintain control and move virtual NICs into these port-groups on demand.

As the VMware admin provisions ESX hosts and selects the appropriate port-groups for VMs, the APIC dynamically communicates with vCenter to make EPGs available through port groups. The APIC also configures VLAN IDs on the leaf-switches as needed.

During a vMotion event, APIC is automatically informed of the VM move and then updates the endpoint tracking table to allow seamless communication. VMs are allowed to move anywhere within the ACI fabric with no restrictions other than those imposed by vCenter.

It is important to note that ACME can still choose to deploy traditional VLAN trunking down to VMware DVS switches by statically provisioning EPGs on a per-port basis, and still reap the advantages of the Layer 2-anywhere ACI fabric. However, ACME chose VMM integration as the preferred deployment model as it is the most effective method of breaking down organizational challenges, doing on-demand resource allocation, and getting enhanced visibility and telemetry into both the virtual and physical environments.

Deploying the Application Virtual Switch

Prerequisites

- All switch nodes have been discovered by the fabric
- INB or OOB management connectivity is configured.
- VMware vCenter is installed, configured, and available.
- One or more vSphere hosts are available for deployment to the AVS.
- A DNS server policy has been configured to enable connection to a VMM using a hostname.
- A dynamic VLAN pool has been created with enough VLAN IDs to accommodate one VLAN per EPG you plan on deploying to each VMM domain.

Getting Started

The AVS software was designed to operate independently of the APIC software version. This allows either device to be upgraded independently. Always refer to the AVS release notes to confirm if any special considerations may exist.

Just like any software, new versions of the AVS will be released to include new features and improvements. The initial AVS software released was version 4.2.1, followed by version 5.2.1. Refer to the *ACI Ecosystem Compatibility List* document to ensure your desired version of AVS is compatible with the APIC and vSphere versions being run.

The AVS package for either version will include vSphere Installation Bundles (VIBs). Each version of AVS software includes the VIB files for all supported vSphere versions. As of this publication there are two VIBs to support vSphere versions 5.1 and 5.5 (vSphere 5.0 is not supported). These can be downloaded from CCO at the following location:

Downloads Home > Products > Switches > Virtual Networking > Application Virtual Switch

The VIBs can be identified as follows:

```
AVS 4.2.1 Bundle
```

```
cross_cisco-vem-v165-4.2.1.2.2.3.0-3.1.1.vib
cross_cisco-vem-v165-4.2.1.2.2.3.0-3.2.1.vib
 AVS 5.2.1 Bundle
```

```
cross_cisco-vem-v172-5.2.1.3.1.3.0-3.1.1.vib
cross_cisco-vem-v172-5.2.1.3.1.3.0-3.2.1.vib
```

Install the AVS VIB

Before setting up the AVS configuration on the APIC, the AVS software must be installed in vSphere, referred to as the Virtual Ethernet Module (VEM). This can be achieved in a variety of ways, all of which are discussed in *Cisco Application Virtual Switch Installation Guide*. For a few hosts, this can easily be done manually, but for 10+ hosts it may be easier to leverage the Virtual Switch Update Manager (VSUM) to help automate the installation process.

Manual Installation

1. Copy the VIB file to a host. The easiest way to copy the VIB to the host is to leverage the VMware VI Client, navigate to the **Host > Configuration > Storage > *Datastore_X***. Right click on the desired datastore and select **Browse**. From here,VIB can be uploaded directly to the host's datastore.
2. SSH into the vSphere host on which the AVS VIB is to be installed.
3. Install or upgrade the VIB using the **esxcli** command:

   To install the AVS VIB:

```
esxcli software vib install -v /<path>/<vibname> --maintenance-mode --no-sig-check
```

To Upgrade an existing AVS VIB:

```
esxcli software vib update -v /<path>/<vibname> --maintenance-mode --no-sig-check
```

A sample output is shown below:

```
# esxcli software vib install -v /vmfs/volumes/datastore1/cross_cisco-vem-v172-
5.2.1.3.1.3.0-3.2.1.vib --maintenance-mode --no-sig-check
  Installation Result
    Message: Operation finished successfully.
    Reboot Required: false
    VIBs Installed: Cisco_bootbank_cisco-vem-v172-5.2.1.3.1.3.0-3.2.1
    VIBs Removed:
    VIBs Skipped:
 /vmfs/volumes/53cab6da-55209af3-0ef2-24e9b391de3e # vem version
 Running esx version -1623387 x86_64
 VEM Version: 5.2.1.3.1.3.0-3.2.1
 VSM Version:
 System Version: VMware ESXi 5.5.0 Releasebuild-1623387
```

4 Confirm the VEM is loaded and running.

```
# vem status

VEM modules are loaded

Switch Name      Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch0         3072        6           128               1500    VMNIC0
 VEM Agent (vemdpa) is running
```

DHCP Relay

The first task is to create a DHCP relay policy in the infra tenant on the APIC. This will allow the AVS to create a vmk Virtual Tunnel Endpoint (VTEP) interface on each host. This VTEP interface will be used for the OpenFlex control channel and/or the VXLAN tunnel source between the VTEP and ACI fabric.

If the default TEP Address pool was used during initial setup (10.0.0.0/16), this will allow the AVS VTEP interfaces to pull an address from this pool. The way APIC would do it is through over the infra vlan. For this reason it is critical to extend the Infra VLAN from a leaf through to each vSphere host that will host the AVS.

Create the Infra DHCP Relay Policy

Create a DHCP relay policy that will setup the DHCP server (the APIC in this case) and a label using the Infra Tenant and default EPG. The APIC IP addresses automatically function as the DHCP providers and **DO NOT** need to be explicitly added.

STEPS

1. On the menu bar, choose **Tenants** > **ALL TENANTS**.
2. In the Work pane, choose the **infra**.
3. In the Navigation pane, choose **infra > Networking > Protocol Policies > DHCP > Relay Policies.**
4. In the Work pane, choose **Actions** > **Create DHCP Relay Policy**.
5. In the **Create DHCP Relay Policy** dialog box, perform the following actions:
   a. Provide a name for the Relay policy, such as "avs-dhcp-relay-pol".
   b. Leave other fields blank and click **OK**.
6. Click **Submit**.

Create the DHCP Relay Label

1. On the menu bar, choose **Tenants** > **ALL TENANTS**.
2. In the Work pane, choose the **infra**.
3. In the Navigation pane, choose **infra** > **Networking** > **Bridge Domains** > **default** > **DHCP Relay Labels**.
4. In the Work pane, choose **Actions** > **Create DHCP Relay Policy**.
5. In the **Create DHCP Relay Policy** dialog box, perform the following actions:
   a. Change the Scope to **TENANT**.
   b. From the Name drop-down list, choose the DHCP Relay Policy that you created previously.
6. Click **Submit**.

Attachable Access Entity Profile (AEP) and AVS

An important component used by the AVS is the Attachable Entity Profile (AEP). Regardless of using an existing AEP or creating a new one, the **Enable Infrastructure VLAN** check box must be checked for the AEP policy. This is to ensure that the traffic of interest (DHCP request/offer can flow through the infrastructure VLAN to the AVS). The AEP defines which VLANs will be permitted on a host facing interface. This can be compared to a "switchport trunk allowed VLAN xxx" command in traditional networking. Referring back to the "VMM Policy Model Interaction" diagram from the "VM Networking Overview" chapter, the AEP is what ties the VMM domain to the physical interfaces where the vSphere hosts are connected. The AEP can be created on-the-fly during the creation of the VMM domain itself, but this guide will detail creating the AEP separately first.

Create a New AEP

1. On the menu bar, choose **Fabric** > **Access Policies**.
2. In the Navigation pane, choose **Global Policies** > **Attachable Access Entity Profile**.
3. In the Work pane, choose **Actions** > **Create Attachable Access Entity Profile**.
4. In the **Create Attachable Access Entity Profile** dialog box, perform the following actions:
   a. Fill in the AEP wizard information then click **Next**.
      i. Name: Provide any name to identify the AEP, such as "AVS-AEP".
      ii. Enable Infrastructure VLAN: Check this box.
      iii. Domains (VMs or Baremetal): Leave blank. This will be covered later in the ***Publishing EPGs to VMM Domains*** chapter.

   b. From the next page of the wizard, select the **Interface Policy Group** your AEP will be associated to. This procedure assumes your Interface Policy Group has already been created. Click the **All Interfaces** radio button for the desired Interface Policy Group.

Note: Interface Policy Group creation is covered elsewhere in this guide. Essentially the Interface Policy Group is a collection of Interface policies which define Interfaces Selectors and properties, such as speed/negotiation, LLDP, and CDP. See the "Adding New Devices to the Fabric" chapter for more detail on creating the interface policy group and interface profiles.

Modify an Exisiting AEP

1. On the menu bar, choose **Fabric** > **Access Policies**.
2. In the Navigation pane, choose **Global Policies** > **Attachable Access Entity Profile**.
   a. In the Navigation pane, choose the existing AEP
   b. In the Work pane, check the **Enable Infrastructure VLAN** check box.

Note: As mentioned early in this chapter, the Infrastructure VLAN is required for AVS communication to the fabric using the OpenFlex control channel.

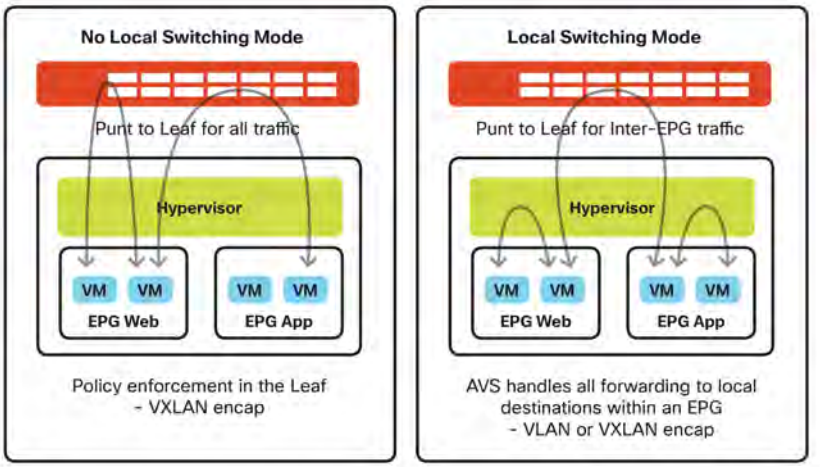
VMM Domains for vCenter

A Virtual Machine Manager (VMM) domain defines a virtual infrastructure that will be integrated into ACI. This allows the same policies applied to physical endpoints, to also be applied to virtual endpoints. vCenter VMM Domains are created using either the VMware DVS or Cisco AVS. You cannot change from one to the other. A new VMM Domain will be created from scratch to support AVS deployment.

AVS Switching Modes

The AVS can operate in the following switching modes:

- Local Switching: Supports VXLAN encapsulation or VLAN encapsulation.
  - This switching mode allows Inter-EPG traffic to be switched locally on the AVS.
- No Local Switch only supports VLAN encapsulation.
  - This switching mode sends all traffic (Inter-EPG included) to the Leaf switch.



AVS Switching Modes: Non-Local and Local switching mode

The decision between using VLAN or VXLAN encapsulation will mandate different VLAN extension requirements outside of the fabric. When using VXLAN encapsulation, only the infra VLAN is required to be extended to the AVS hosts. All traffic between the AVS uplinks and ACI fabric will be encapsulated by VXLAN and transfered using the infrastructure VLAN.

If VLAN encapsulation is preferred, you will need to ensure every VLAN in the VM Domain VLAN pool has been extended between the fabric and AVS host. This includes creating the VLANs on intermediate devices such as UCS and the vNICs for any AVS vSphere hosts.

Create the VMM Domain for AVS

Now that the DHCP server policy has been created and AEP created/modified, you can create the VMM domain for the AVS.

1. On the menu bar, choose **VM NETWORKING**.
2. In the Navigation pane, choose the **Policies** tab.
3. In the Work pane, choose **Actions > Create VCenter Domain**.

4 In the **Create vCenter Domain** dialog box, perform the following actions:

a. **Name**: This value will be used as the AVS "Switchname" displayed in vCenter.

b. **Virtual Switch**: **Cisco AVS**

c. **Switching Preference: <Choose Local or No Local Switching>**

- For **No Local Switching** mode:
  - Multicast Address: <Assign a multicast address to represent your AVS>
  - Multicast Address Pool: <Create a unique Multicast Address Pool large enough to include each AVS vSphere host.>
- For **Local Switching** mode:
  - Encapsulation: <Choose VLAN or VXLAN based on preference>
    - For **VLAN** Encapsulation
      - VLAN Pool: <Choose/Create a VLAN pool>
    - For **VXLAN** Encapsulation
      - Multicast Address: <Assign a multicast address to represent your AVS>
      - Multicast Address Pool: <Create a unique Multicast Address Pool large enough to include each AVS vSphere host.>

d. **Attachable Access Entity Profile**: <Choose the AEP previously created/modified>

e. **vCenter Credentials**: Create a credential set with administrator/root acces to vCenter

f. **vCenter**: Add the vCenter details.

- Name: Friendly name for this vCenter
- Hostname/IP Address: <DNS or IP Address of vCenter>
- DVS Version: vCenter Default
- Datacenter: <Enter the exact Datacenter name displayed in vCenter>
- Management EPG: <Set to oob or inb Management EPG>
- Associated Credentials: <Choose the Credential set previously created>
- Click **OK** to complete the creation of the vCenter.

5 Click **Submit**.

Verify AVS Deployment on vCenter

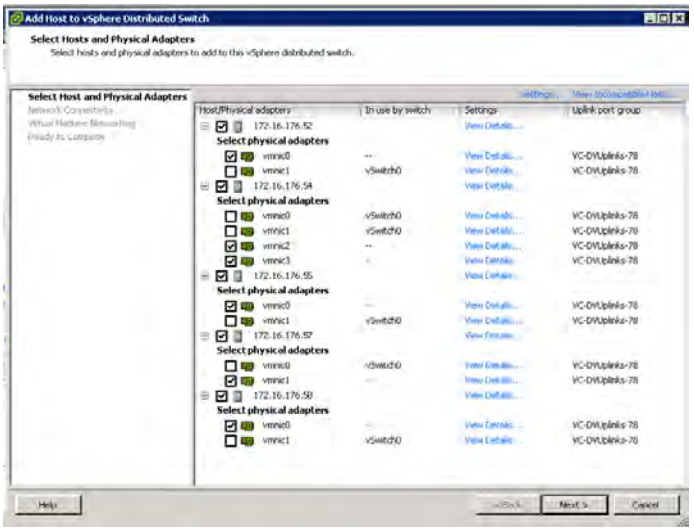
1. In the vCenter client, navigate to **HOME > INVENTORY > NETWORKING** and confirm a new Distributed Virtual Switch folder has been created.
2. Expand this folder to find your AVS, and a few default Port Groups including "**uplink**" and "**vtep**".

Add vSphere Hosts to the AVS

After the AVS has been created in vCenter, you then need to attach hosts to it. To do this you will need at least one unused physical interface (VMNIC) to act as the uplink on each host. AVS uplinks can not be shared with any other existing vSwitch or vDS.

1. From the vCenter client, navigate to **HOME** > **INVENTORY** > **NETWORKING**.
2. Right click on the newly created AVS switch (not the folder) and choose **Add Host...**.
3. In the **Add Host** dialog box, choose any vSphere hosts to add to the AVS, and select an unassigned VMNIC uplink.
4. Click **Next** until the wizard completes, skipping the migration of any virtual adapters or virtual machine networking at this time.

Note: For blade switch systems such as UCS, the VMNIC interface used must have all necessary VLANs allowed on the interface. In UCS terms, this requires the vNIC within the service profile to have all relevant VLANs active on the vNICs.



Adding Virtual host physical NICs to participate in Virtual Switch

5 Assuming the ACI fabric can reach the vSphere host over the infra VLAN, you should see a new vmk interface created on your distributed switch within vCenter and assigned to the 'vtep' port group. This vmk is your Virtual Tunnel Endpoint (vtep) interface and should have pulled a DHCP address from the APIC from the TEP subnet. As can be seen from the screenshot below, we see that the VMkernel port has received the IP address from the APIC. The APIC uses the same 10.0.0.0/16 pool that is created during the APIC setup to provision the IP address. This implies that we are ready for Opflex communication in between the AVS and the APIC.

```
~ # esxcfg-VMKNIC -l
 Interface  Port Group/DVPort   IP Family IP Address
Netmask         Broadcast       MAC Address       MTU     TSO MSS   Enabled Type
  vmk0       Management Network  IPv4      172.16.176.54
255.255.255.0   172.16.176.255  00:25:b5:00:00:29 1500    65535     true    STATIC
       vmk1       vmotion             IPv4      192.168.99.54
255.255.255.0   192.168.99.255  00:50:56:61:1c:92 1500    65535     true    STATIC
        vmk2       9                   IPv4      10.0.16.95
255.255.0.0     10.0.255.255    00:50:56:65:3d:b3 1500    65535     true    DHCP
```

Verify AVS on ESX

On the ESX command line, issue the '**vemcmd show openflex**' command.

Verify that the 'status: 12 (Active)' is seen as well as the switching mode. Also verify that the GIPO address is the same as the multicast address that was used while creating the VMM domain.

```
~ # vemcmd show openflex
Status: 12 (Active)
Dvs name: comp/prov-VMware/ctrlr-[AVS-TEST]-VC/sw-dvs-87
Remote IP: 10.0.0.30 Port: 8000
Infra vlan: 4093
FTEP IP: 10.0.0.32
Switching Mode: LS
NS GIPO: 225.127.1.1
```

Verify on the AVS host - there should be one multicast group per deployed EPG on the host. In the output below, there are three different Virtual Machines connected to different EPGs.

```
~ # vemcmd show epp multicast
Number of Group Additions 3
Number of Group Deletions 0
Multicast Address        EPP Ref Count
      225.0.0.58      1
      225.0.0.76      1
      225.0.0.92      1
```

These multicast addresses will correspond to the EPG details found in the APIC GUI under **Tenants** > *TenantX* > **Application Profiles** > *ApplicationProfileX* > **End Point Groups** > *EndPointGroupX* and click the **Operational Tab** > **Client End Points**.

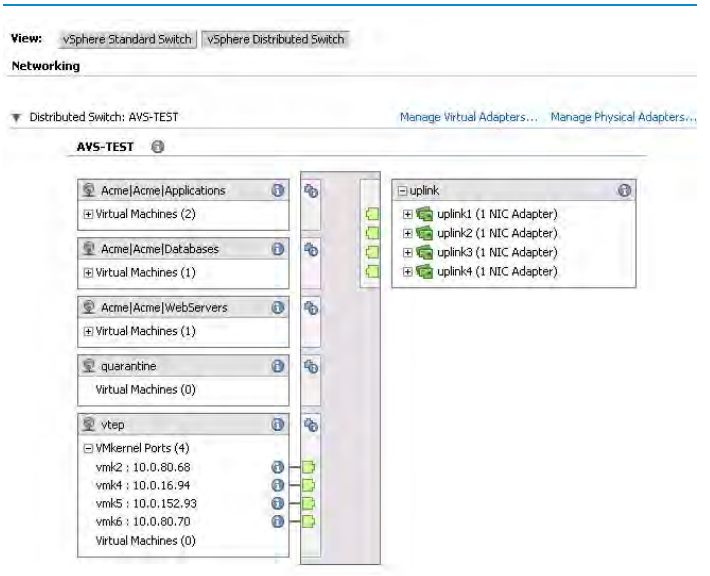
VXLAN Load Balancing

VXLAN load balancing is automatically enabled as soon as more than one VMKNIC is connected to the Cisco AVS. Each VMKNIC can use only one uplink port; you should have an equal number of VMKNICs and uplinks. A maximum of eight VMKNICs can be

attached to a Cisco AVS switch. Each of the VMKNICs that you create has its own software-based MAC address. In VXLAN load balancing, the VMKNICs provide a unique MAC address to packets of data that can then be directed to use certain physical NICs (VMNICs).

You need to have as many VMKNICs as the host has VMNICs, up to a maximum of eight. For example, if the host has five VMNICs, you need to add four VMKNICs to enable VXLAN load-balancing; the Cisco Application Policy Infrastructure Controller (APIC) already created one VMKNIC when the host was added to the distributed virtual switch (DVS).

In VMware vSphere Client, you will need to create an additional virtual adapter (VMK) for each AVS uplink. Each vmk interface created for the AVS should be attached to the **vtep** port group and configured for DHCP. In the screenshot below you can see the four VMNIC uplinks to the AVS and the four vmk virtual interfaces to provide equal load balancing traffic across all uplinks. Note that each VMK interface added is assigned a unique DHCP address from the fabric TEP pool.



Distributed Switch configured with APIC port groups

IGMP Snooping Policy for AVS

Cisco UCS-B Series Considerations with AVS Deployments

This section of the article will focus the necessary steps to enable AVS through the Cisco UCS-B series.

By default, USC-B FI has IGMP snooping enabled. Due to this, we have to configure an IGMP querier policy on the APIC. The IGMP snooping policy needs to be enabled on the infra tenant.

If we disable IGMP snooping on UCS or other intermediate blade switches, then IGMP policy is not needed since the blade switch will flood the multicast traffic on all the relevant ports.

Create an IGMP Snooping Policy for AVS

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose **infra**.
3. In the Navigation pane, choose **infr > Networking** > **Bridge Domain** > **default**.
   a. In the **IGMP Snoop Policy** drop-down list, choose **Create IGMP Snooping Policy**.
   b. Provide a name for the policy, such as "IGMP_Infra".
   c. Click the **Fast Leave** check box.
   d. Click the **Enable Querier** check box.
4. Click **Submit**.
   Note: Verify if IGMP snooping is working properly on the vSphere host CLI using '**vemcmd show epp multicast**' as shown above.

The alternate method would be to create an IGMP policy on UCS to disable IGMP snooping. This will cause flooding of the multicast traffic to all endpoints.

External Connectivity

Extending ACI to External Layer 2

As mentioned in the introduction of this book, ACME Inc. is a multinational company with multiple data centers. Therefore, ACME Inc. must configure some Layer 2 connectivity. This is necessary for extending Layer 2 connectivity to a Data Center Interconnect (DCI) platform, to further extend connectivity to a remote data center, or simply to extend a Layer 2 domain outside of the fabric to connect in an existing Layer 2 network in a non-ACI fabric.

Extending Endpoint Groups Outside the ACI Fabric

The simplest way to extend an endpoint group (EPG) outside of the ACI fabric is to statically assign a leaf port and VLAN ID to an existing endpoint group. After doing so, all of the traffic received on this leaf port with the configured VLAN ID will be mapped to the EPG, and as such, the configured policy for this EPG will be enforced. The endpoints do not need to be directly connected to the ACI leaf, as the traffic classification will be based on the encapsulation received on a port.

To assign a Layer 2 connection statically on an ACI leaf port to an EPG:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane, choose ***Tenant_Name*** **>** **Application Profiles >** ***App_Profile_Name*** **>** **Application EPGs >** ***EPG_Name*** **> Static Bindings (Paths)**.
4. In the Work pane, choose **Action > Deploy Static EPG on PC, vPC or Interface**.
5. In the **Deploy Static EPG on PC, vPC or Interface** dialog box, perform the following actions:
   a. In the **Path** field, specify a port as well as a VLAN ID.
   b. Click one of the **Deployment Immediacy** radio buttons. Deployment immediacy determines when the actual configuration will be applied on the leaf switch hardware. The immediacy also determines when the hardware resource, such as a VLAN resource and policy content-addressable memory

(CAM) to support the related contract for this EPG, will be consumed on the leaf switch. The option **Immediate** means that the EPG configuration and its related policy configuration will be programmed in the hardware right away. The option **On Demand** instructs the leaf switch to program the EPG and its related policy in the hardware only when traffic matching this policy is received for this EPG.

c. Click one of the **Mode** radio buttons. The mode option specifies whether the ACI leaf expects incoming traffic to be tagged with a VLAN ID or not.
   i. **Tagged -** The tagged option means that the leaf node expects incoming traffic to be tagged with the specified VLAN ID previously established. This is the default deployment mode. Choose this mode if the traffic from the host is tagged with a VLAN ID. Multiple EPGs can be statically bound to the same interface as long as the encap VLAN/VXLAN ID is unique.
   ii. **Untagged -** The untagged option means that the leaf expects untagged traffic without a VLAN ID. Similar to the **switchport access vlan *vlan_ID*** command, with this option you can only assign the interface to one EPG. This option can be used to connect a leaf port to a bare metal server whose network interface cards (NICs) typically generate untagged traffic. A port can have only one EPG statically bound to a port as untagged.
   iii. **802.1P -** The 802.1P option refers to traffic tagged with 802.1P headers. 802.1P mode is useful when its necessary to handle the traffic on one EPG as untagged to the interface (similar to the ***switchport trunk native vlan vlan_ID*** command), but (unlike the untagged mode) 802.1P will allow other 'tagged' EPGs to be statically bound to the same interface. Any traffic received on links with this mode classification will have the following conditions applied to them:

Encap Mode	Multiple VLANS Supported per Port	Supported Untagged Traffic	Common Use Case
Untagged	No	Yes	Bare Metal Server or PXE device (Non Hypervisor)
tagged	Yes	No	Hypervisor
802.1P	Yes	Yes	Hypervisor with PXE Devices

d. Create a physical domain and VLAN pool that are associated to this physical domain.
e. Associate the physical domain to the EPG in question.

f. Create an attachable access entity profile (AEP) to map the interfaces and policies together.

See the Adding New Devices to the Fabric section for more information on how to configure an AEP and a physical domain.

Extending an ACI Bridge Domain Outside of the Fabric

A Layer 2 outside connection is associated with a bridge domain and is designed to extend the whole bridge domain, not just an individual EPG under the bridge domain, to the outside network.

To accomplish an extension of the bridge domain to the outside, a Layer 2 outside connection must be created for the bridge domain. During this process, create a new external EPG to classify this external traffic. This new EPG will be part of the existing bridge domain. Classify any outside connections or endpoints into this new external EPG. With two separate EPGs, you will also need to select which traffic you would like to traverse between the two EPGs. Similar to the previous example of adding an endpoint to a pre-existing EPG, this method will also allow the endpoints to share the same subnet and default gateway.

To create an external Layer 2 domain:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane, choose ***Tenant_Name*** **> Networking > External Bridged Network**.
4. In the Work pane, choose **Action > Create Bridged Outside**.
5. In the **Create Bridged Outside** dialog box, perform the following actions:
   a. Associate the Layer 2 outside connection with the bridge domain and a VLAN ID. This VLAN must be configured on the external Layer 2 network. This Layer 2 outside connection will put this VLAN and the bridge domain of the ACI fabric under the same Layer 2 domain. This VLAN ID must be in the range of the VLAN pool that is used for the Layer 2 outside connection.
      i. For the **External Bridged Domain** drop-down list, create a Layer 2 domain if one does not already exist.

ii. While creating the Layer 2 domain, if it does not already exist, create a VLAN pool to associate to the VLAN on the Layer 2 outside connection. This is a means to specify the range of the VLAN IDs that will be used for creating a Layer 2 outside connection. This helps avoid any overlapping in the VLAN range between VLANs used for an EPG and those in use for a Layer 2 outside connection.

b. Add a Layer 2 border leaf node and Layer 2 interface for a Layer 2 outside connection.
c. After adding a Layer 2 border leaf and Layer 2 interface, click **Next** to start creating a Layer 2 EPG. Simply provide a name for the Layer 2 EPG. All of the traffic entering the ACI fabric with the designated VLAN (the VLAN ID provided in step 1) will be classified into this Layer 2 EPG.
d. Configure a contract to allow communication between your existing endpoints in the existing EPG and your new external Layer 2 EPG. In the Navigation pane, choose **External Bridged Networks > Networks** and specify a contract to govern this policy as the consumed contract. After specifying this contract as the provided contract for your internal EPG, the communication between this external Layer 2 EPG and your existing internal EPG will be allowed.
e. Create an AEP. This is a policy object that tells the APIC to allow certain encap (VLANs) on selected ports. For more information on how to create a Access Attachable Entity Profile, see the Adding New Devices to the Fabric section.

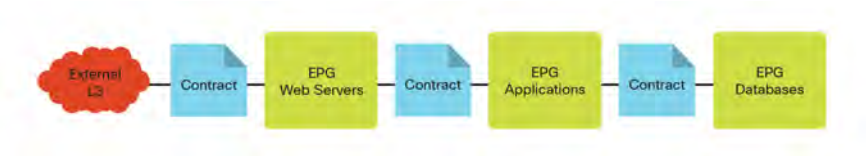
You should now have the desired reachability between the inside and outside Layer 2 segments.

Extending ACI to External Layer 3

The most important component of any application is the user or customer, which generally does not directly attach to the fabric, and therefore there must be connected to the external network. ACME must be able to connect to both their internal corporate backbone, as well as to the Internet to provide access to the mobile application. This integration is possible with Cisco Application Centric Infrastructure (ACI) at the tenant policy level. Layer 3 connectivity to a device like a router is known as an **External Routed Network**, and provides IP connectivity between a tenant private network and an external IP network. Each Layer 3 external connection is associated with one tenant

private network. The requirement of the Layer 3 external network is only needed when a group of devices in the application profile require Layer 3 connectivity to a network outside of the ACI fabric.

An application profile enables an operator to group the different components, or tiers, of an application into endpoint groups (EPGs). These application components might have requirements for external connectivity into them. The following figure shows part of a simplified fabric:



A sample application profile for a three-tiered application with contracts between the tiers

For example, web servers need a connection to the outside world for users to reach them. With ACI, the connectivity is defined by a contract to a defined external Layer 3 endpoint group. As the operator of the fabric, you can provide the tenant administrator with the ability to interface to an external Layer 3 connection in various ways by using a uniquely-defined Layer 3 construct for the tenant application profile or a shared common infrastructure.

External Layer 3 connections are usually established on the border leaf construct of the ACI. Any ACI leaf can become a border leaf. In large scale ACI designs it might be productive to have dedicated ACI leafs as border leafs. It is important to note that the spine nodes cannot have connections to external routers. A border leaf is simply terminology to refer to a leaf that happens to be connected to a Layer 3 device. Other devices, like servers, can still connect to the border leaves. In the ACI fabric, the external Layer 3 connection can be one of the following types:

1. Physical Layer 3 interface
2. Subinterface with 8021.Q tagging
3. Switched Virtual Interface (SVI)

Bridge domains contain one or more subnets. These subnets can be classified as private, public, or shared:

- Public - Indicates that this subnet will be advertised to the external router.
- Private - Indicates that this subnet will be contained only within the ACI fabric private network.
- Shared - Indicates that this subnet will be leaked to one or more private networks inside of the ACI fabric.

The following figure depicts the logic of public and private networks:

Application profile with external consumers, public and private networks annotated

With devices connecting through the external Layer 3 connection, the external network has learned of the internal ACI network 10.1.1.0/24, as it is advertised to the adjacent router through the Layer 3 external connection. For the private networks, ACI does not advertise the networks through the routing protocol to the adjacent Layer 3 router, and the networks are not reachable to devices external to the fabric.

In releases prior to version 1.1 of Cisco Application Policy Infrastructure Controller (APIC), the fabric only advertises subnets that are marked public in the associated bridge domain. Routes that are learned externally from the fabric are not advertised through other ports. This behavior is known as a non-transit fabric. In release version 1.1 and later, ACI is capable of acting as a transit network, and routes learned from one external Layer 3 connection can be advertised out to a different external Layer 3 connection, not just fabric routes.

The network team will provide the external Layer 3 connectivity for their tenants. One common mechanism is to use sub-interfaces on a router to create different Layer 3 domains since each tenant will likely not have their own external router.

Supported Routing Protocols

The following routing protocols are supported at time of writing:

- Static routes—You can define static routes to the outside world. Using static routes reduces the sizing and complexity of the routing tables in the leaf nodes, but increases administrator overhead. With static routes, you must also configure the static path back to the interanal network that you wish to be reachable from the outside world.
- OSPF NSSA—Using not-so-stubby area (NSSA) reduces the size of the Open Shortest Path First (OSPF) database and the need to maintain the overhead of the routing protocols with large tables of routes. With OSPF NSSA, the router learns only a summarization of routes, including a default path out of the fabric. OSPF NSSA advertises to the adjacent router the internal public subnets part of the Layer 3 external.
- iBGP peering leaf and external router—With internal Border Gateway Protocol (iBGP), ACI supports only one autonomous system (AS) number that has to match the one that is used for the internal Multiprotocol Border Gateway Protocol (MP-BGP) route reflector. Without MP-BGP, the external routes (static, OSPF, or BGP) for the Layer 3 outside connections are not propagated within the ACI fabric, and the ACI leaves that are not part of the border leaf does not have IP connectivity to any outside networks. Given that the same AS number is used for both cases, the user must find out the AS number on the router to which the ACI border leaf will connect, and use that AS number as the BGP AS number for the ACI fabric.

Configure MP-BGP Spine Route Reflectors

The ACI fabric route reflectors use multiprotocol border gateway protocol (MP-BGP) to distribute external routes within the fabric so a full mesh BGP topology is not required. To enable route reflectors in the ACI fabric, the fabric administrator must select at least one spine switch that will be a route reflector, and provide the autonomous system (AS) number for the fabric. Once route reflectors are configured, administrators can setup connectivity to external networks.

To connect external Layer 3 devices to the ACI fabric, the fabric infrastructure operator must configure a Route Reflector policy to designate which spines act as the route reflector(s). For redundancy purposes, configure more than one spine as a router reflector node.

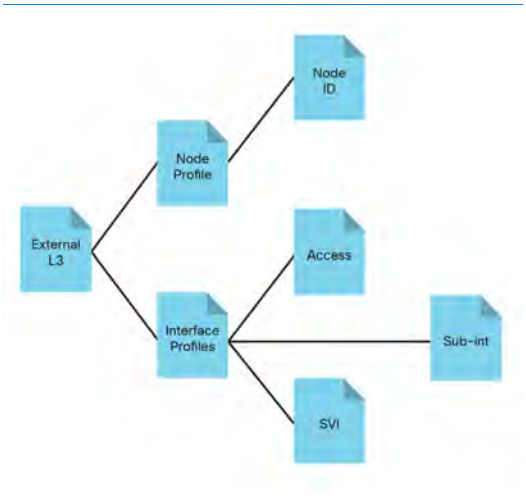
When a tenant requires a Layer 3 connection, the infrastructure operator configures the leaf node to which the WAN router is being connected as border leaf node, which pairs the border leaf node with one of the route reflector nodes as a BGP peer. After the route reflectors are configured, they can advertise routes in the fabric.

Each leaf node can store up to 4000 routes at time of writing. If a WAN router must advertise more than 4000 routes, the router should peer with multiple leaf nodes. The infrastructure operator configures each of the paired leaf nodes with the routes (or route prefixes) that the nodes can advertise.

To configure the Route Reflector policy:

1. On the menu bar, choose **Fabric > Fabric Policies**.
2. In the Navigation pane, choose **Pod Policies > Policies > BGP Route Deflector default**.
3. In the Work pane, perform the following actions:
   a. Change the **Autonomous System Number** to match the required number for your network.
   b. Add the two spine nodes that will be members of this reflector policy and
   c. Click **Submit**.
4. In the Navigation pane, choose **Pod Policies**.
5. In the Work pane, choose **Actions > Create Pod Policy Group**.
6. In the **Create Pod Policy Group** dialog box, perform the following actions:
   a. In the BGP Route Reflector Policy drop-down list, choose **default**.
   b. In the Navigation pane, choose **Pod Policies > Profiles > default**.
   c. In the Work pane, in the **Fabric Policy Group** drop-down list, choose **Create Pod Policy Group**.
   d. In the **Create Pod Policy Group** dialog box, **i**n the **Date Time Policy** drop-down list, choose **default**.
   e. In the **BGP Route Reflector Policy** drop-down list, choose **default**.
   f. Complete the remainder of the dialog box as appropriate to your setup.
7. Click **Submit**.

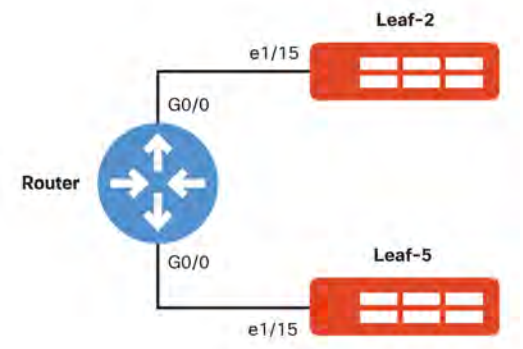
The following figure illustrates the objects and their relationships for external Layer 3 connections:



Object relationships for Layer 3 outside objects

Layer 3 Integration Through Tenant Network with OSPF NSSA

The following figure shows a simple integration of a Layer 3 external into ACI using OSPF:



Logical topology for an external OSPF router communicating with two border leafs

The setup includes a single router with two interfaces connected to leaf switches.

Note: See the "Adding New Devices to The Fabric" section to setup the access policies for the interfaces of the leaves that are connected to the router.

To integrate Layer 3 through a tenant network with OSPF/NSSA:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name*** **> Networking > External Routed Networks**.
4. In the Work pane, choose **Action > Create Routed Outside**.
5. In the **Create Routed Outside** dialog box, perform the following actions:
   a. In the **Name** field, enter a name for the profile.
   b. In the **Private Network** drop-down list, choose the private network for this tenant.
   c. Click the **OSPF** check box.
   d. In the **OSPF Area ID** field, enter the OSPF area ID, such as "1".
   e. In the **OSPF Area Control** section, click the **Send redistributed LSAs into NSSA area** check box.
   f. In the **OSPF Area Type** section, click the **NSSA Area** radio button.
   g. In the **Nodes and Interfaces Protocol Profiles** section, click **+** to add a profile.
   h. In the **Create Node Profile** dialog box, perform the following actions:
      i. In the Name field, enter a name for the profile.
      ii. In the **Nodes** section, click **+** to add a node.
      iii. In the **Select Node** dialog box, perform the following actions:
         1. In the **Node ID** drop-down list, choose a node, such as **Leaf-1**.
         2. In the **Router ID** field, enter the router's IP address as the ID, such as "10.0.1.1".
         3. Uncheck the **Router ID as Loopback Address** check box.
         4. In the **Loopback Addresses** section, click **+** to add a loopback address.
         5. Enter the loopback address, such as "10.254.254.1", and click **Update**.
         6. Click **OK**.

iv. In the **OSPF Interface Profiles** section, click **+** to create an OSPF interface profile.

v. In the **Create Interface Profile** dialog box, perform the following actions:

1. In the **Name** field, enter a name for the profile.
2. In the **OSPF Policy** drop-down list, choose **Create OSPF Interface Policy**. When defining the interaction with another OSPF router, you must specify the policy interaction. This document does not explain the different OSPF parameters.
3. In the **Create OSPF Interface Policy** dialog box, perform the following actions:
   a. In the **Name** field, enter a name for the OSPF policy, such as "OSPF-Point2Point".
   b. In the **Network Type** section, click the radio button that matches the adjacent router, such as **Point to Point**.
   c. Complete the remainder of the dialog box as appropriate to your setup.
   d. Click **Submit**.
4. In the **Interfaces** section, click on the **Routed Intefaces** tab.
5. Click the **+** sign to select a routed interface.
6. In the **Select Routed Interface** dialog box, perform the following actions:
   a. In the **Path** drop-down list, choose the interface on the leaf, such as e1/9 on Leaf-1.
   b. In the **IP Address** field, enter the IP address of the path that is attached to the layer 3 outside profile, such as "10.0.1.1/24".
   c. In the **MTU (bytes)** field, enter the maximum MTU of the external network, such as "1500" to match the example peering router.
   d. Complete the remainder of the dialog box as appropriate to your setup and click **OK**.
7. Click **OK**.

vi. Click **OK**.

i. Click **Next**.

j. In the **External EPG Networks** section, click **+** create an external network.

k. In the **Create External Network** dialog box, perform the following actions:

i. In the **IP Address** field, enter 0.0.0.0/0 to permit the learning of any subnet and click **OK**.

l. Click **Finish**.
Next, you must configure the external network EPG.

External Layer 3 for Multiple Tenants

In ACI, you can use various mechanisms to reuse the same external Layer 3 router for multiple tenants. If the adjacent router is a Cisco Nexus Series Switch with a Layer 2 trunk interface, the external Layer 3 connection can be configured to route via SVI. For routers capable of using sub-interfaces, those can be used to provide multiple external Layer 3 connection for multiple VRFs and/or tenants. The fabric operator can configure multiple external Layer 3 connections using either sub-interface or SVI and provide that to each tenant.

Application Migration Use Case

When operating the ACI fabric, there can be occasions when you will have to migrate workloads, servers or virtualization hosts from outside the ACI fabric, onto the fabric. One common example is when migrating from a traditional data center configuration over to a policy-driven data center using ACI. As ACME starts to use ACI in more of their data centers, it will be come necessary to perform these migrations. In this example, ACME must manage the migration of SVI interfaces as well as the policy allowing traffic to traverse a Layer 2 outside network and then on to the ACI fabric.

At a high level, you must start by configuring the Layer 2 outside network to allow traffic from the source VLAN to communicate with the same VLAN residing on the ACI fabric. You will also need to configure Layer 3 connectivity from the fabric out to the existing Layer 3 networks in preparation for full connectivity after SVI migration.

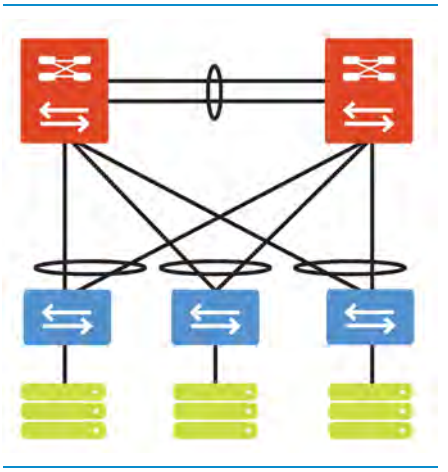
Further information and steps on how to create this Layer 2 and Layer 3 connectivity including the policy, can be found in the Fabric Connectivity chapter of this book.

Once they have successfuly established connectivity between the outside Layer 2 network (where the workload or host is coming from) and the existing internal fabric EPG, you can then start the migration process of moving application workloads onto the fabric. One key consideration should be when to switch over the SVI interfaces from the existing environment into the ACI fabric and when to start advertising routes to this SVI network. Assuming that the SVIs reside on the external Layer 2 network, Cisco recommends that you move the SVIs over to the ACI fabric once a majority of the hosts have been migrated over.

Extending the Network to ACI

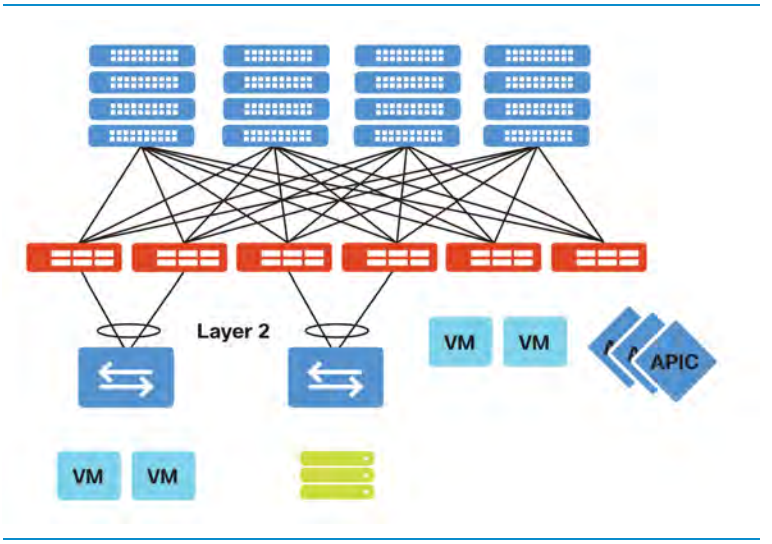
One of the ACME sites would like to migrate from the legacy data center architecture to the next generation ACI Fabric. They would like to migrate with minimal service interruption while taking advantage of ACI innovations where applicable. ACME would like to perform the migration in multiple stages.

The Legacy data center prior to migration:



Traditional pre-migration data center

The ACI data center following migration:



Post-migration ACI based data center topology

The first stage will provide connectivity from the legacy data center to the ACI fabric. In this state, you logically map a VLAN=EPG. The interconnect from the legacy network to the ACI fabric will be accomplished through standard Layer 2 extensions (VLAN/VXLAN).

Provide physical connectivity from the existing aggregation layer to the ACI border leafs. This connectivity can be accomplished in either the form of a Virtual Port Channel, Port Channel, or a single interface.

1. Provide a physical connection from aggregation switch #1 to the ACI border leaf #1.
2. Provide a physical connection from aggregation switch #2 to the ACI border leaf #1.

Note: Before connecting external physical connections into the fabric, the Fabric Access Policies for the access ports that you will be used for the DCI must be configured. For details on configuring the access policies please reference the Fabric Connectivty section of this book.

Configure the aggregation links as a Layer 2 trunk.

1. Trunk the VLAN representing the host connectivity. This allows for the host VLAN to be extended into the fabric.

In the APIC controller you will now configure a single tenant. The created tenant will represent the legacy data center into the ACI fabric.

1. On the menu bar, choose **Tenants > Add Tenant**.
2. In the **Create Tenant** dialog box perform the following actions:
   a. In the **Name** field enter a name for the tenant.
   b. Click **Next**.
3. Click **Finish**.

Configure a single private network.

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Networking** > **Private Networks**.

4 In the Work pane, choose **Actions > Create Private Network**.

5 In the **Create Private Network** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the private network.
   b. Click **Next**.
   c. In the **Name** field enter a name for the bridge domain.
   d. In the **Forwarding** drop-down list, choose **Custom**.
   e. For the **Layer 2 Unknown Unicast** radio buttons, click **Flood**.
   f. For the **Multi Destination Flooding** radio buttons, click **Flood in BD**.
   g. Click the **ARP Flooing** check box.

6 Click **Finish**.

Note: The concept of flooding the unknown unicast and arp within the Fabric is to allow for the Layer 2 semantics from the legacy data center to be extended into the ACI Fabric. When a host in the legacy data center sends an ARP request and/or floods an unknown unicast frame, the bridge domain will then mimic the behavior in the ACI Fabric. By default BPDU frames are flooded within EPG.

Configure a single application profile:

1 On the menu bar, choose **Tenants** > **ALL TENANTS**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose **Application Profiles**.

4 In the Work pane, choose **Actions > Create Application Profile**.

5 In the **Create Application Profile** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the Application Profile.
   b. Click **Submit**.

Configure a single endpoint group:

1 On the menu bar, choose **Tenants > ALL TENANTS**.

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Application Profiles >** ***Application_Profile_Name***.

4 In the Work pane, choose **Actions > Create Application EPG**.

5 In the **Create Application EPG** dialog box, perform the following actions:
   a. In the **Name** field, enter a name for the endpoint group.
   b. In the **Bridge Domain** field, choose the appropiate bridge domain.
   c. Click **Finish**.

Note: The EPG within the fabric will map to a single VLAN in the legacy data center. The use of a single VLAN per EPG will provide a path for a network-centric migration–minimizing impact while introducing fabric innovations.

Configure a VPC for the connectivity to the legacy data center. See the "Fabric Connectivity" section. Then, configure a static trunk binding using the VPC under the EPG, AcmeOutSide. The encapsulation VLAN should match the defined legacy data center VLAN definition.

1 On the menu bar, choose **Tenants > ALL TENANTS**.

2 In the Work pane, choose the ***Tenant_Name***.

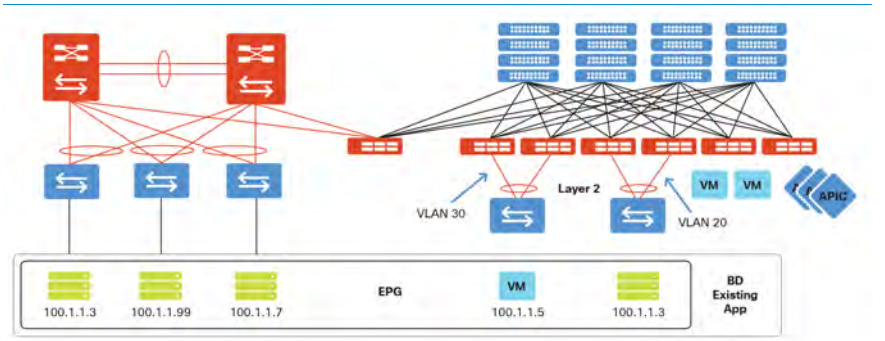
3 In the Navigation pane choose ***Tenant_Name*** **> Application Profiles >** ***Application_Profile_Name*** **> Application EPGs >** ***EPG_Name*** **> Static Bindings (Paths)**.

4 In the Work pane, choose **Actions > Deploy Static EPG on PC, VPC or Interface**.

5 In the **Deploy Static EPG on PC, VPC or Interface** dialog box, perform the following actions:
   a. Choose the **Path Type**.
   b. Choose the **Path**.
   c. Enter the encapsulation VLAN.
   d. Click **Submit**.

Following the Stage 1 migration, the Legacy host VLAN is now extended to the ACI fabric and all hosts from the Fabric point of view are in the EPG, AcmeOutSide. From the ACI Fabric perspective, the local VLAN number is not significant as it will be mapped to the tagged VLAN on the VPC toward the Legacy data center. This is what is known as normalization where ACI uses the tagged vlan to map external Layer 2 connections into ACI End Point Groups.

The connectivity noted below represents a BD=EPG=VLAN. The Layer 3 gateway for the ACI fabric and the legacy data center or provided by the Legacy data center.



An EPG as a VLAN datacenter deployment

The second stage of the migration takes advantage of the fabric policy model to define further the application modeling. During this stage, multiple EPGs will be created with APIC contracts that define reachability. The following list is an overview of the steps that you must perform:

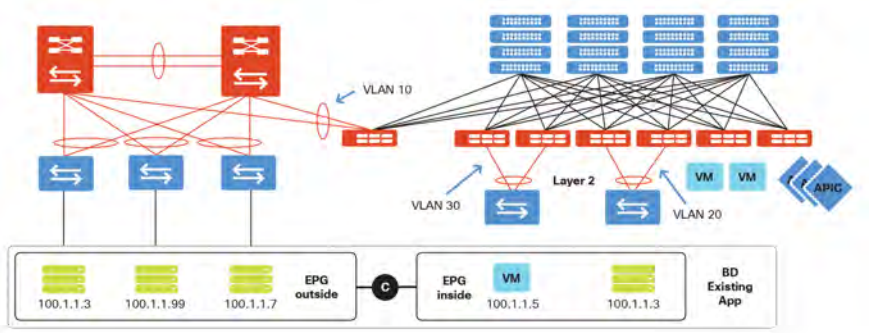
1 In the APIC:
   a. Configure additional Tenants (optional).
   b. Configure additional Private Networks (optional).
   c. Configure additional Bridge Domains (optional).
   d. Configure additional Application Profiles (optional).
   e. Configure additional End Point Groups.
      i. Create the endpoint group "AcmeInSide".
   f. Configure contracts for inter-EPG communications.
      i. For the **Scope** drop-down list, choose **Tenant**.
      ii. For the **Allow** drop-down list, choose **Any-Any**.

2 Begin migrating the host to the fabric EPG, "AcmeInSide".
   a. Create a static binding for each physical host to be migrated.
   b. Create a VMM domain to deploy the host within the ACI Fabric (optional).

   Note: Ensure the appropriate resources have been created to support the connectivity, VLAN pools, physical domain or VMM Domain, AEP, and switch or interface profiles. For more information, see the "Fabric Connectivty" section.

Following the stage 2 migration, the host connectivity across both the legacy data center and ACI fabric are now governed by the APIC policy (contracts).

Note: The Layer 3 gateway for the ACI fabric and the legacy data center are provided by the legacy data center.

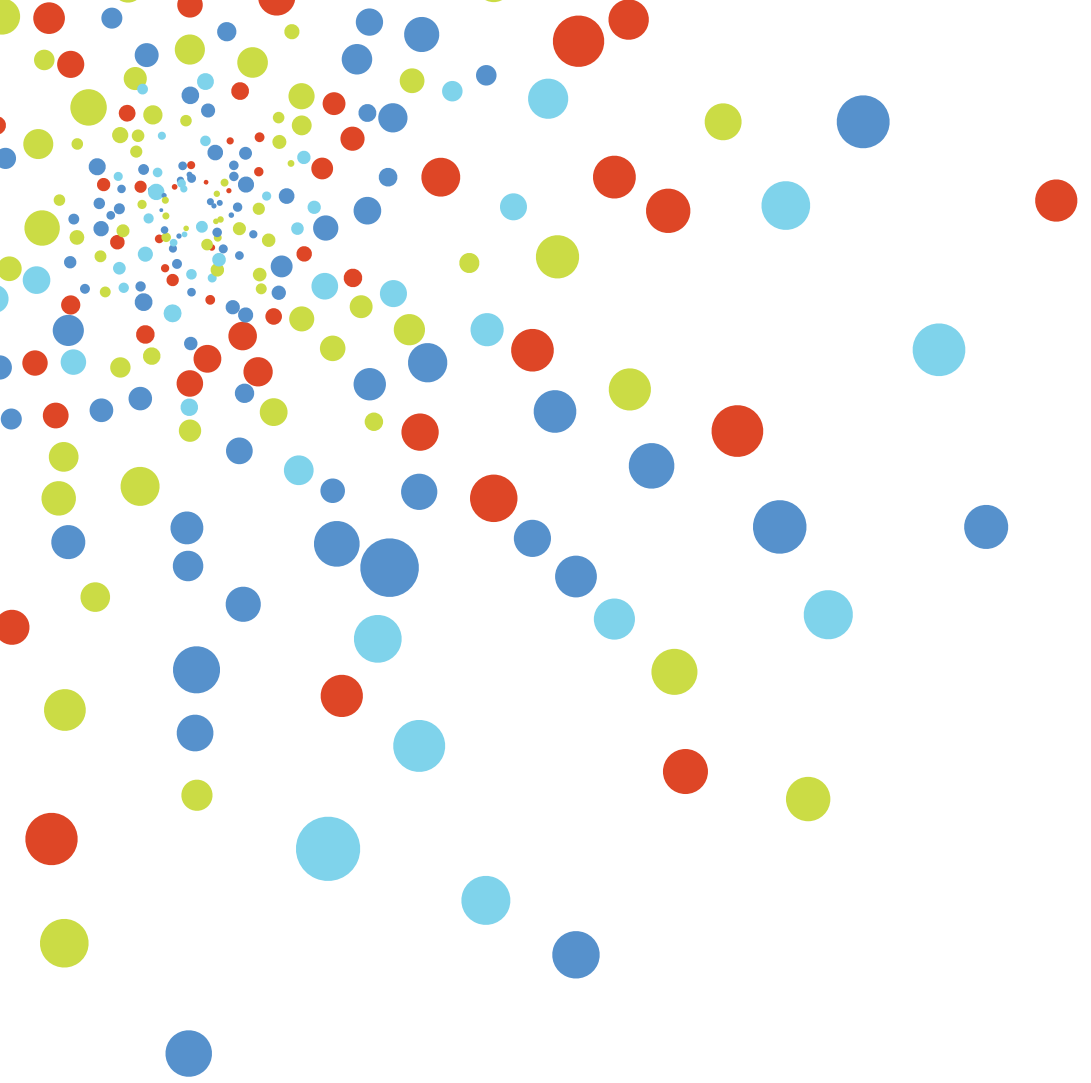


Datacenter migration with Layer 3 provided by existing DC and Layer 2 extended to new ACI datacenter

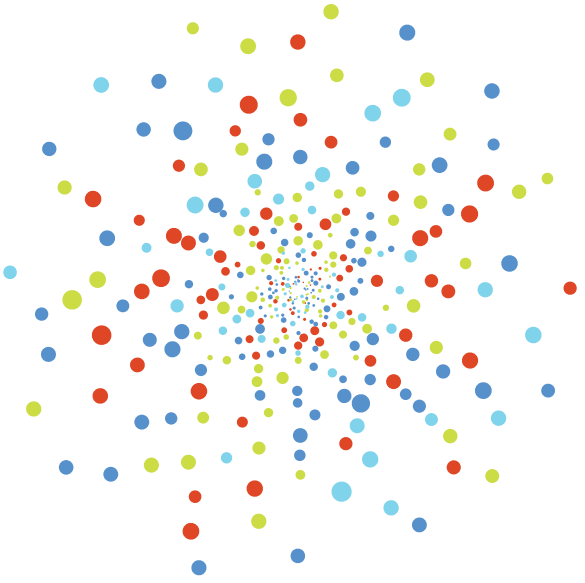
In the third stage of the migration the Layer 3 gateway will move from the legacy data center to the ACI Fabric. The following list is an overview of the steps that you must perform:

1 In the APIC:
   a. Configure a Layer 3 out.
   b. Migrate the gateway from the legacy data center to the fabric.
      i. In the **Bridge Domain** drop-down list, choose **AcmeBD**.
      ii. In the **Flood Layer 2** drop-down list, choose **Unknown Unicast**.
      iii. In the **ARP** drop-down list, choose **Flooding**.
      iv. In the **Unicast** drop-down list, choose **Routing**.

Note: The concept of unicast routing within the bridge domain allows for the configuration of the pervasive gateway across the fabric. The Layer 3 gateway for the ACI fabric and the legacy data center are provided by the ACI fabric.



Tenants



Section Content

- ACI Tenancy Models
- Application Profile
  - Application Profile Configuration
    - Create a New Application Profile
    - Modify Application Profile
    - Remove Application Profile
    - Verify Application Profile
- Endpoint Group
  - Endpoint Group Configuration
    - Create a New Endpoint Group
    - Modify Endpoint Group
    - Remove Endpoint Group
    - Verify Endpoint Group
- Endpoint
  - Verify Endpoint Group
- Private Network
  - Private Network Configuration Parameters
  - Creating a New Private Network
  - Modify Private Network
  - Remove Private Network
  - Verify Private Network

ACI Tenancy Models

ACME Inc. will be using tenancy for a couple of use cases. They will be using tenant constructs for the application lifecycle of their current deployment, maintaining a separate tenant for the resources that developers will be using to build the application, a tenant that will be used for the automated testing, and finally a production tenant. Additionally, as mentioned in the introduction, they are also looking to build an infrastructure which can be leveraged for similar initiatives in the future. Tenants will be used to draw virtual boundaries for different lines of business. The information security team will be able to integrate this into the corporate LDAP system, and prevent changes which would impact other groups.

ACI has been designed from the beginning to be "multi-tenant". This means different things to different people (much like the term Cloud) based on their perspective. In the case of a classic service provider, a tenant is a unique customer, while in a typical end-customer environment a tenant could be an operating group, business unit, application owner, etc.

The decision on how to leverage tenancy models is driven by a number of factors:

1. Overall IT operations and support models in your organization to manage application, networking, servers, security, and so on.
2. Separation of environments from a software development lifecycle perspective: Development, Quality Assurance, and Production.
3. Separation of duties by domain owner, such as web, app, and database owners.
4. Fault domain size and scope to limit the impact of failures, such as different business units.

In traditional networking environments, making a routing protocol change on a router or Layer 3 switch could potentially affect hundreds of unique VLANs/subnets. This introduces a warranted level of caution around change control and application impact. Leveraging the ACI policy model, the physical hardware is abstracted from the logical constructs. The tenant object gives us the ability to draw a box around the logical and concrete objects that we use to provide a unified view of the configuration dependencies for underlay and overlay networks.

A Tenant in the ACI Object model represents the highest-level object. Inside, you can differentiate between the objects that define the tenant networking, such as Private Networks, Bridge Domains and subnets; and the objects that define the tenant policies such as Application Profiles and Endpoint Groups.

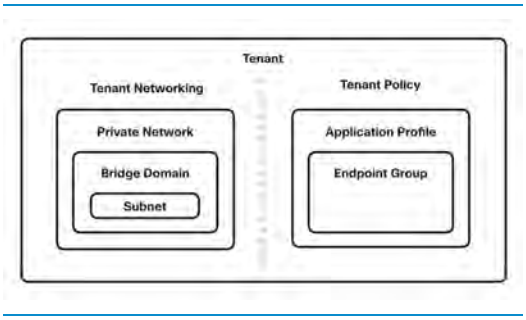
In ACI, the tenant policies are where you define applications. An application could consist of a combination of physical servers or VMs that we will call servers from now on. For example, a website could use a 3-tier application model, comprised of web servers, application servers and database servers. When a user browses the web site, they might actually be communicating with a virtual IP address on a load balancer that in turn can distribute the web request to a number of different web servers. The web servers in turn communicate with core applications that can be divided amongst several applications servers for load balancing or high availability purposes. Finally, the application servers communicate with the database which could also be a cluster of servers.

Each server is referred to as an Endpoint in ACI. Endpoints are classified in ACI to apply policies. You create endpoint groups with endpoints that share the same type of policies, such as with whom are they going to communicate and what type of communication or restrictions are required. Therefore, an application can be formed by several endpoint groups and they are grouped in an Application Profile.

The tenant networking is used to define networking policies and will be applied to the underlying hardware in a transparent way thanks to the layer of abstraction provided by ACI using private networks, bridge domains and subnets. In the next sections of this chapter these concepts will be covered in detail. Below you can find an illustration with the different objects that compound a tenant and how they are related.

Although the tenant networking and the tenant policies are defined separately, the networking policies used by an application are defined with a relationship between the Endpoint Groups and the Bridge Domain.

The following image shows all of the components that can be configured within a tenant. In the following sections each diagram shows the progress of how ACME Inc. adds each component.



Tenant Logical Model

There are 3 Tenants preconfigured in the system by defaut:

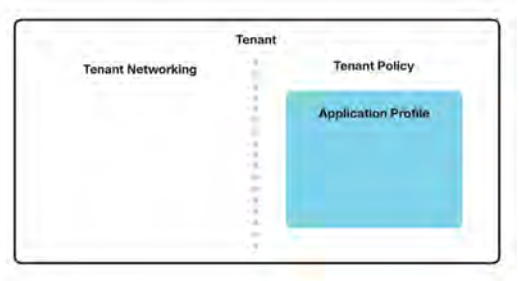
1. Common – a special tenant with the purpose of providing "common" services to other tenants in the ACI fabric. Global reuse is a core principle in the common tenant. Some examples of common services are:
   a. Shared Private Networks
   b. Shared Bridge Domains
   c. DNS
   d. DHCP
   e. Active Directory
2. Infra – The Infrastructure tenant that is used for all internal fabric communications, such as tunnels and policy deployment. This includes switch to switch (Leaf, Spine, Application Virtual Switch (AVS)) and switch to APIC. The infra tenant does not get exposed to the user space (tenants) and it has its own private network space and bridge domains. Fabric discovery, image management, and DHCP for fabric functions are all handled within this tenant.
3. Mgmt - The management tenant provides convenient means to configure access policies for fabric nodes. While fabric nodes are accessible and configurable through the APIC, they can also be accessed directly using in-band and out-of band connections. In-band and out-of-band policies are configured under the mgmt tenant:
   - In-Band Management Access
   - Out-of-Band Management Access

Application Profile

An application profile is a convenient logical container for multiple hosts (physical or virtual). You can create Application Profile containers based on a variety of criteria, such as what function the application provides, how the application looks from the end-user perspective, where they are located within the context of the data center, or any other logical grouping relative to the implementation. Application Profile servers are grouped in EPGs depending on the use of common policies.

Application Profiles provide a mechanism to understand groups of servers as a single application. This approach makes an ACI application aware and allows us to check the operational state for an application monitoring all the servers that are part of an application as a whole and become informed about relevant faults and health status for that particular application. Each Application Profile created can have a unique monitoring policy and QOS policy.

An Application Profile is a child object of the Tenant and a single Tenant can contain multiple Application Profiles.



Adding components to a Tenant - 1. Application Profile

Application Profile Configuration

Name - The name of the application profile.

Tags - A tag or metadata is a non-hierarchical keyword or term assigned to the fabric module.

Monitoring Policy - The monitoring policy name for the EPG semantic scope (optional).

Create a New Application Profile

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Application Profiles**.
4. In the Work pane, choose **Actions > Create Application Profile**.
5. In the **Create Application Profile** dialog box, perform the following actions:
   a. Enter an Application Profile **Name**.
   b. Enter a **TAG** (optional).
   c. Choose a **Monitoring Policy** (optional).
6. Click **Submit**.

Modify Application Profile

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Application Profiles >** ***Application_Profile.***
4. In the Work pane, choose **policy**.
5. In the **Create Application Profile** dialog box, perform the following actions:
   a. Enter an Application Profile **Name**.
   b. Enter an appropriate **TAG** (optional).
   c. Choose the **Monitoring Policy** (optional).
6. Click **Submit**.

Remove Application Profile

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name* > Application Profiles > *Application_Profile.***
4. In the Work pane, choose **Actions > Delete**.

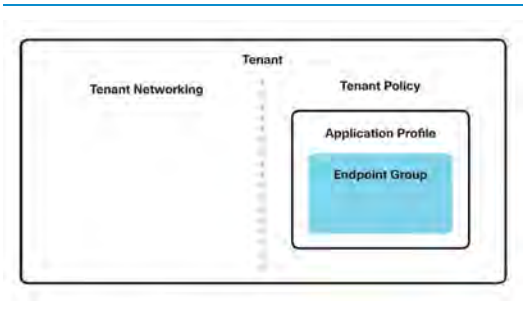
Verify Application Profile

```
REST :: /api/node/class/fvAp.xml
 CLI :: moquery -c fvAp
```

Endpoint Group

Endpoint Groups are used to create logical groupings of hosts or servers that perform similar functions within the fabric. Each Endpoint Group created can have a unique monitoring policy and QoS policy and must be associated with a Bridge Domain.

An Endpoint group is a child object of the Application Profile and an Application Profile can contain multiple Endpoint Groups. Each endpoint within an Endpoint Group is susceptible to the same policy in the Fabric.



Adding components to a tenant - 2. End Point Group in the Application Profile

All the Endpoints inside an EPG can communicate with each other. How communications between EPGs contracts will be required is governed by contracts and not traditional Layer 2/Layer 3 forwarding constructs. For example, Host-A in EPG-A can have the IP address/mask of 10.1.1.10/24 and Host B in EPG B can have the IP address/mask 10.1.1.20/24 (note that both hosts believe they are "in the same subnet"). In this case they would not be allowed to communicate unless a contract that permitted connectivity existed between EPG-A and EPG-B. Contracts will be explained in greater detail in a following section.

Note that there are some types of Endpoint Groups within the fabric that are not contained under Application Profiles such as, Application Endpoint Group, External Bridge Networks (aka Layer2 External), External Routed Networks (aka as Layer3 External) and Management Endpoint Groups. These Endpoint Groups might have special require-

ments, for example, in External Bridge Networks, MAC addresses of the endpoints are not learnt by the leaf switches.

Endpoint Groups are linked to Bridge Domains but they will receive a VLAN ID different from the bridge domain, unless Bridge Domain legacy mode is used.

It is important to understand that a single subnet can be extended across several EPGs. Each EPG is identified by an encapsulation VLAN or VXLAN so that the same subnet will be using different encapsulation IDs across the fabric. This concept is different from traditional networking.

Endpoint Group Configuration

Name - The name for the endpoint group.

Tag - A tag or metadata is a non-hierarchical keyword or term assigned to the fabric module.

Qos Class - The QoS priority class identifier.

The class can be:

- **Unspecified**
- **Level1**-Class 1 Differentiated Services Code Point (DSCP) value.
- **Level2**-Class 2 DSCP value.
- **Level3**-Class 3 DSCP value.

Custom Qos - The QoS traffic priority class identifier. The Custom class is a user-configurable DSCP value.

Bridge Domain - The name of the bridge domain associated with this object.

Monitoring Policy – The monitoring policy name for the EPG semantic scope (optional).

Associated Domain Profile – A source relation to an infrastructure domain profile associated with application endpoint groups.

Subnet - An Endpoint Group subnet is relevant when and only when configuring route leaking between VRF's/Private Network within a Tenant (optional).

Static Endpoint - The static client endpoint represents a silent client endpoint attached to the network (optional).

Create a New Endpoint Group

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Application Profiles >** ***Application_Profile*** **> Application EPGs**.
4. In the Work pane, choose **Actions > Create Application EPG.**
5. In the **Create Application EPG** dialog box, perform the following.
   a. Enter an **Application EPG Name**.
   b. Enter an **Tag** (optional).
   c. Enter an **Qos Class** (optional).
   d. Enter an **Custom Qos** (optional).
   e. Enter a **Bridge Domain Name**.
   f. Choose a **Monitoring Policy** (optional).
   g. Enter an **Associated Domain Profile Name**.
6. Click Finish.

Modify Endpoint Group

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Application Profiles >** ***Application_Profile*** **> Application EPGs** > ***Application_EPG***.
4. In the Work pane, select policy.
   a. Enter an **Application EPG Name**.
   b. Enter an **Tag** (optional).
   c. Enter an **Qos Class** (optional).
   d. Enter an **Custom Qos** (optional).

e. Enter a **Bridge Domain Name**.
f. Choose the appropriate **Monitoring Policy** if applicable (optional).
g. Enter an **Associated Domain Profile Name**.

5 Click Finish.

Remove Endpoint Group

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Application Profiles > *Application_Profile*** **>** **Application EPGs** > ***Application_EPG***..

4 In the Work pane, choose **Actions > Delete**.

Verify Endpoint Group

```
REST :: /api/node/class/fvAEPg.xml
CLI :: moquery -c fvAEPg
```

Endpoint

Endpoints are devices that are connected to the network either directly or indirectly. Endpoints have an address (identity), a location, and attributes, and can be either virtual or physical. Each endpoint has a path, an encapsulation, and a deployment Immediacy mode associated with it.

An Endpoint is a child object of the Endpoint Group and an Endpoint Group construct can contain multiple Endpoints. The Endpoints referenced within the fabric can be either static (defined within the APIC) or dynamic (automated by vCenter/Openstack).

You can add Static Endpoints by creating Static Bindings within the Endpoint Group. Below is an example of a static binding. See the VVM section for an example of a dynamic binding.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Application Profiles >** ***Application_Profile_Name*** **> Application EPGs >** ***EPG_Name*** **> Static Bindings (Paths)**.
4. In the Work pane, choose **Actions > Deploy Static EPG on PC, VPC or Interface**.
5. In the **Deploy Static EPG on PC, VPC or Interface** dialog box, perform the following actions:
   a. Choose the **Path Type**.
   b. Choose the **Path**.
   c. Enter the encapsulation VLAN.
   d. Click **Submit**.

In order to show the endpoints that are connected to the fabric under certain EPGs:

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation Pane choose ***Tenant_Name*** **> Application Profiles > *Application_Profile*** **> Application EPGs** > ***Application_EPG***.

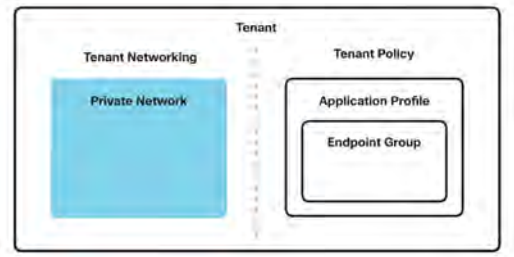
4 In the Work pane, choose **Operational**.

Verify Endpoint

```
REST :: /api/node/class/fvCEp.xml
CLI  :: moquery -c fvCEp
```

Private Network

A Private Network is also referred to as a VRF, private Layer 3 network, or context. It is a unique Layer 3 forwarding and application policy domain. Private networks are a child of the Tenant object. All of the endpoints within the Private Network must have unique IP addresses because it is possible to forward packets directly between these devices if the policy allows it. One or more bridge domains are associated with a private network.



Adding components to a Tenant - 3. Private Network as part of the Tenant Logical Model

The most common method to share Private Networks between tenants is through the common tenant. For more information about common tenants, see the overview section of this chapter. Private Networks created in the common tenant are shared globally within the fabric. However, a Private Network that is intended to be used by multiple tenants and is not created in the common tenant requires explicit configuration to be shared.

When there is a requirement to route traffic between separate Private Network instances, special consideration for subnet configuration is needed. This will be discussed in detail in the Bridge Domain and EPG configuration sections.

Private Network Configuration Parameters

Name - The name of the Private Network.

Policy Enforcement - The preferred policy control. The values can be **enforced** or **unenforced**. When enforced is chosen, contracts between EPGs are required to allow traffic. Unenforced allows all traffic within the Private Network. The default is **enforced**.

BGP Timers - Name of the BGP timers policy associated with this object.

OSPF Timers - Name of the OSPF timers policy associated with this object.

End Point Retention Policy - The end point retention policy name (optional).

Monitoring Policy - The monitoring policy name for the Tenant semantic scope (optional).

DNS Label - The network domain name label. Labels enable classifying which objects can and cannot communicate with one another (optional).

Creating a New Private Network

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Networking** > **Private Networks.**
4. In the Work pane, choose **Actions > Create Private Network.**
5. In the **Create Private Network** dialog box, perform the following actions:
   a. Enter an **Private Network Name**.
   b. Choose a **Policy Enforcement** (optional).
   c. Choose a **BGP Policy Name** (optional).
   d. Choose an **OSPF Policy Name** (optional).
   e. Choose an **End Point Retention Policy Name** (optional).
   f. Choose the appropriate **Monitoring Policy** if applicable (optional).
   g. Choose the appropriate **DNS Label** if applicable (optional).
6. Click **Finish**.

Modify Private Network

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **>** **Networking** > **Private Networks** > ***Private_Network.***
4. In the Work pane, select **policy.**
   a. Choosesa **Policy Enforcement** (optional).
   b. Choose a **BGP Policy Name** (optional).
   c. Choose an **OSPF Policy Name** (optional).
   d. Choose an **EIGRP Policy Name** (optional).
   e. Choose an **End Point Retention Policy Name** (optional).
   f. Choose the appropriate **Monitoring Policy** if applicable (optional).
   g. Choose the appropriate **DNS Label** if applicable (optional).
5. Click **Finish**.

Remove Private Network

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **>** **Networking** > **Private Networks** > ***Private_Network.***
4. In the Work pane, choose **Actions > Delete**.

Verify Private Network

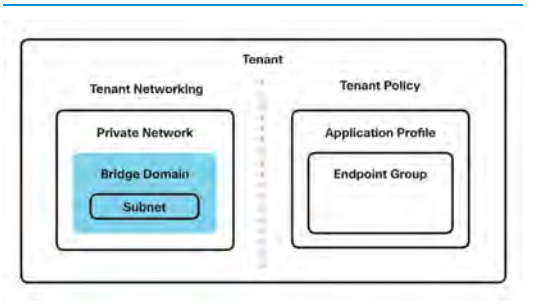
```
REST :: /api/node/class/fvCtx.xml
CLI :: moquery -c fvCtx
```

Bridge Domain

A Bridge Domain is the abstract representation of a Layer 2 forwarding domain within the fabric. A Bridge Domain is a child of the Tenant object and must be linked to a Private Network.

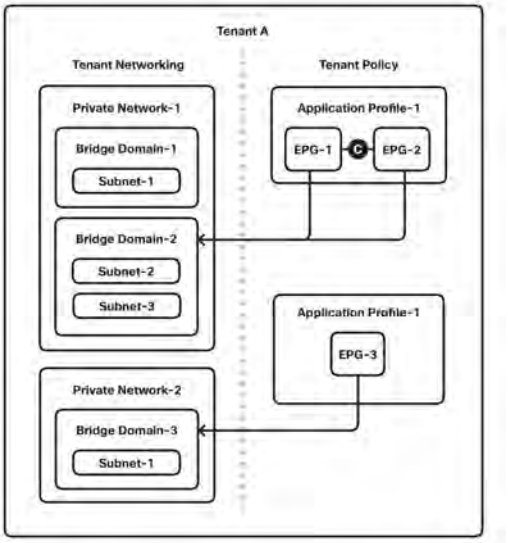
The bridge domain defines the unique Layer 2 MAC address space and a Layer 2 flood domain if flooding is enabled. While a Private Network defines a unique IP address space, that address space can consist of multiple subnets. Those subnets will be spread across one or more bridge domains contained in the Private Network.

Bridge domains will span all switches in which associated EPG are configured. A bridge domain can have multiple subnets. However, a subnet is contained within a single bridge domain.



Adding components to a Tenant - 4. Bridge Domain as part of the Tenant Application Profile

The following image provides an example of a tenant to show how bridge domains are contained inside of Private Networks and how they are linked to EPGs and the other elements.



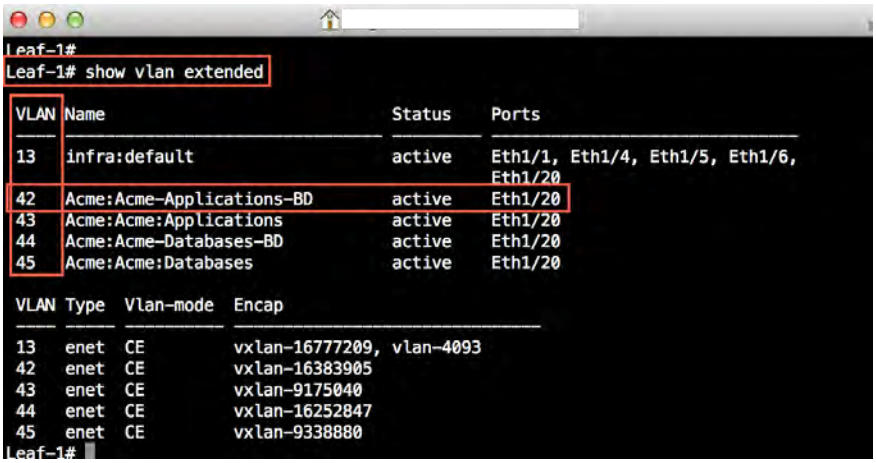
End Point Group as part of the Tenant Application Profile

It is important to understand that a bridge domain is NOT a VLAN, although it can act similar to a VLAN. You instead should think of it as a distributed switch, which, on a leaf, can be translated locally as a VLAN with local significance.

From a practical perspective, each Bridge Domain will exist in a particular leaf if there is a connected endpoint that belongs to that EPG. Each Bridge Domain receives a VLAN ID in the leaf switches. This VLAN ID is significant locally in the leaf switches and therefore it might be differ from one to other leaf switch.

The VLAN ID used is also called Platform Independent VLAN or PI VLAN. This VLAN concept is different from traditional networking and it is not used to forward traffic but as an identifier. Each PI VLAN is then linked to a VXLAN ID that will be used for forwarding purposes inside of the fabric.

In the following example, under the Tenant Acme, the bridge domain Acme-Applications-BD was assigned the PI VLAN ID 42 in the Leaf-1.



VLAN output from Leaf node

EPGs are also assigned with a PI VLAN ID that is locally significant in each leaf. This VLAN ID is different from the Bridge Domain. Therefore in ACI, several VLANs will be used for EPs inside on one bridge domain. For more details refer to the EP section in this chapter.

In some situations where traditional networking is required, for example during some migration processes, it might be required to have only one encapsulation VLAN per Bridge domain across all the leaf switches and EPGs that reference that Bridge Domain. For this situation, there is a feature called Bridge domain legacy mode.

Bridge domain in legacy mode is not recommended under ACI best practices. It is thought to use traditional networking within ACI but it limits the ACI features that can be used. EPGs contained in bridge domains in legacy mode cannot communicate with EPGs in ACI bridge domains.

When a Subnet is defined in a Bridge Domain, the leaf switches will be the default gateway for the EPGs using that subnet. If the EPGs have endpoints on multiple leaves, each leaf will configure the default gateway. In that way, the default gateway for the endpoints will always be the first switch of the fabric that is reached, also know as a pervasive gateway. This means that an SVI will be configured under the VRF that represents the private

network that the Bridge Domain is linked to. If a Bridge Domain has several subnets, there will be only one SVI per Bridge Domain but it will use secondary IP addresses.

Bridge Domain Configuration Parameters

Name - The name of the Bridge Domain.

Network - The associated layer 3 context.

Forwarding - Optimize/Custom

> L2 Unknown Unicast - The forwarding method for unknown layer 2 destinations. Default is Proxy.
>
> Unknown Multicast Flooding - The node forwarding parameter for unknown Multicast destinations.
>
> ARP Flooding - A property to specify whether ARP flooding is enabled. If flooding is disabled, unicast routing will be performed on the target IP address. Flooding is disabled by default but can be enabled by clicking the check box.
>
> Unicast Routing - The forwarding method based on predefined forwarding criteria (IP or MAC address). Unicast routing is enabled by default but can be disabled by clicking the check box.
>
> Config BD MAC Address - The MAC address of the bridge domain (BD) or switched virtual interface (SVI). Every BD by default takes the fabric wide default mac address.

IGMP Snoop Policy - The IGMP Snooping policy name. By examining (snooping) IGMP membership report messages from interested hosts, multicast traffic is limited to the subset of VLAN interfaces on which the hosts reside.

Associated L3 Outs - The name of the Layer 3 outside interface associated with this object.

L3 Out for Route Profile - The Layer 3 outside interface identifier controlling connectivity to outside networks.

Route Profile - The associated route profile name.

Monitoring Policy - The monitoring policy name for the Tenant semantic scope (optional).

Subnets - The IP address and mask of the default gateway. It will be configured in the leaf nodes which have EPGs in that bridge domain.

The network visibility of the subnet. The subnet is a portion of a network sharing a particular subnet address. The scope can be:

- **Shared** - Defines subnets under an endpoint group, with the Shared option configured, to route leak to other Tenants within the Fabric.
- **Public** - Defines subnets under a bridge domain, with the Public option configured, to share with Layer 3 outbound.
- **Private** - Defines subnets under a bridge domain, with the Private option configured, to only be used in that Tenant (will not be leaked). The default is **Private**.

Subnet Control - The control can be specific protocols applied to the subnet such as IGMP Snooping. The control can be:

- **Unspecified -** The default is **Unspecified**.
- **Querier IP** - Enables IGMP Snooping on the subnet.

DHCP Labels - The network domain name label.

Create a new Bridge Domain

1. On the menu bar, choose **Tenants** > **ALL TENANTS**.
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Networking** > **Bridge Domains**.
4. In the Work pane, choose **Actions > Create Bridge Domain**.
5. In the **Create Bridge Domain** dialog box, perform the following actions:
   a. Enter an **Bridge Domain Name**.
   b. Choose the **Network**.
   c. Choose the **Forwarding Semantics** (optional).

d. Choose the **IGMP Snoop Policy** (optional).
e. Choose the **Associated L3 Outs** (optional).
f. Choose the **L3 Out for Route Profile** (optional).
g. Choose the **Route Profile** (optional).
h. Choose the **Monitoring Policy** if applicable (optional).
i. Choose the **Subnets** (optional).
j. Choose the **DNS Label** if applicable (optional).

6 Click **Submit**.

Modify a Bridge Domain

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **>** **Networking** > **Bridge Domains** **>** ***Bridge_Domain_Name.***

4 In the Work pane, choose the **Policy** tab and perform the following actions:
a. Choose the **Network**.
b. Choose the **Forwarding Semantics** (optional).
c. Choose the **IGMP Snoop Policy** (optional).
d. Choose the **Associated L3 Outs** (optional).
e. Choose the **L3 Out for Route Profile** (optional).
f. Choose the **Route Profile** (optional).
g. Choose the **Monitoring Policy** if applicable (optional).
h. Choose the **Subnets** (optional).
i. Choose the **DNS Label** if applicable (optional).

5 Click **Finish**.

Remove a Bridge Domain

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation Pane choose ***Tenant_Name*** **>** **Networking** > **Bridge Domain** > ***Bridge Domain_Name.***

4 In the Work pane, choose **Actions > Delete**.

Verify Bridge Domain

```
REST :: /api/node/class/fvBD.xml
 CLI :: moquery -c fvBD
```

Tenant Networking Use Cases

Common Private Network for All Tenants

This use case may be typical for environments where an ACI administrator wishes to create multiple tenants, but place all within a single private network in the fabric.

This method has the following advantages and disadvantages:

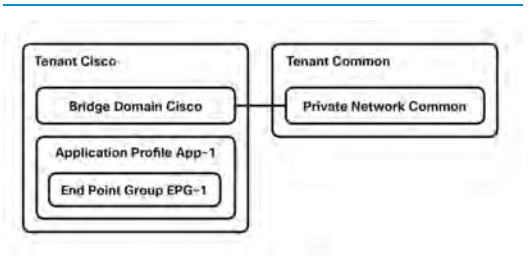
Advantages:

- Ability to use a single private network for all internal and external fabric connectivity
- No route leaking needed between EPGs in different VRFs
- Single Layer 3 Outside can be used by all tenants

Disadvantages:

- Changes to routing will impact all tenants

From a containment and relationship perspective, this topology looks as follows:



Common Private Network for all Tenants

To Configure the common Tenant private network:

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the **common**.

3. In the Navigation pane choose **common** > **Networking** > **Private Networks > default.**
4. In the Work pane, choose **policy.**
   a. Choose a **Policy Enforcement** (optional).
   b. Choosea **BGP Policy Name** (optional).
   c. Choose an **OSPF Policy Name** (optional).
   d. Choosean **End Point Retention Policy Name** (optional).
   e. Choose the appropriate **Monitoring Policy** if applicable (optional).
   f. Choosethe appropriate **DNS Label** if applicable (optional).
5. Click **Finish**.

The Tenant has been created. Now the network administrator will have to associate the common private network to the Tenant by first creating a bridge domain.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Bridge Domains.**
4. In the Work pane, choose **Actions** > **Create Bridge Domain.**
5. In the **Create Bridge Domain** dialog box, perform the following actions:
   a. Enter a Bridge Domain **Name.**
   b. Choose **network.**
   c. In the **Subnets** field, click **+**.
   d. In the **Gateway IP** field enter the IP address for this subnet.
   e. In the Scope field you have the option to select Private, Public and Shared. Note: By default the Private option is selected. For more information on what to select please reference the External Layer 3 section.
6. Click **OK**.
7. Click **Finish**.

The configuration for this use case can be applied via the following CLI configuration:

CLI : Tenant Cisco

```
# tenant
cd '/aci/tenants'
mocreate 'Cisco'
moconfig commit
# bridge-domain
cd '/aci/tenants/Cisco/networking/bridge-domains'
mocreate 'Cisco'
cd 'Cisco'
moset network 'default'
moconfig commit
# subnet
cd '/aci/tenants/Cisco/networking/bridge-domains/Cisco/subnets'
mocreate '172.16.0.1/24'
moconfig commit
# application-profile
cd '/aci/tenants/Cisco/application-profiles'
mocreate 'App1'
moconfig commit
# application-epg
cd '/aci/tenants/Cisco/application-profiles/App1/application-epgs'
mocreate 'EPG1'
cd 'EPG1'
moset bridge-domain 'Cisco'
moconfig commit
# criterion
cd '/aci/tenants/Cisco/application-profiles/App1/application-epgs/EPG1/vm-attributes-
criteria'
mocreate 'default'
moconfig commit
```

This configuration can also be applied using the following XML posted to the APIC REST API

XML : Tenant Cisco

```
<fvTenant name="Cisco">
    <fvBD arpFlood="no" multiDstPktAct="bd-flood" name="Cisco" unicastRoute="yes"
unkMacUcastAct="proxy" unkMcastAct="flood">
        <fvRsCtx tnFvCtxName="default"/>
        <fvSubnet ctrl="nd" descr="" ip="172.16.0.1/24" preferred="no"
scope="private"/>
    </fvBD>
    <fvAp name="App1">
        <fvAEPg matchT="AtleastOne" name="EPG1">
            <fvRsBd tnFvBDName="Cisco"/>
        </fvAEPg>
    </fvAp>
    <fvRsTenantMonPol tnMonEPGPolName=""/>
</fvTenant>
```

For many multi-tenant environments it is desirable to allow each tenant to manage and own their own address space and not be concerned with overlaps between other tenants. This particular use case demonstrates how a private network can be associated with each tenant. One Private Network per Tenant with Intra-EPG communications

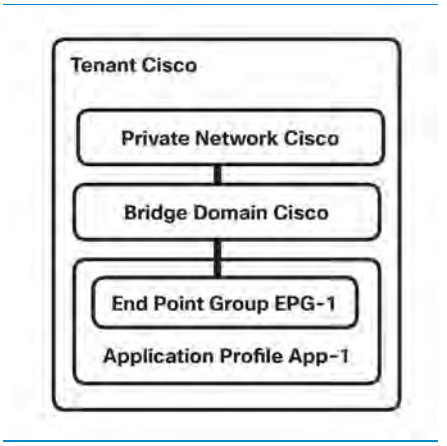
Advantages:

- Allow for maximum isolation between tenants
- Ability to address hosts in tenants with overlapping IP addresses

Disadvantages:

- Increased complexity when needing to allow EPG communication between different tenants with dedicated VRF

The object containment for this particular setup can be depicted as shown below:



Private Network per Tenant

To create the tenant:

1. On the menu bar, choose **Tenants** > **ADD TENANT.**
2. In the **Create Tenant** dialog box, perform the following actions:
   a. Enter a Tenant **Name.**
   b. Click **Next.**
3. Click **Finish.**

The Tenant has been created. Now the tenant administrator can create the private network.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Private Networks.**
4. In the Work pane, choose **Actions > Create Private Network.**
5. In the **Create Private Network** dialog box, perform the following actions:
   a. Enter Private Network **Name.**
   b. Click **Next.**

c. Enter Associated Bridge Domain **Name.**
d. In the **Subnets** field, click **+**.
e. In the **Gateway IP** field enter the IP address for this subnet.
f. In the Scope field you have the option to select Private, Public and Shared. Note: By default the Private option is selected. For more information on what to select please reference the External Layer 3 section.

6 Click **OK.**

7 Click **Finish**.

The configuration for this use case can be applied via the following CLI configuration:

CLI : Tenant Cisco

```
# tenant
cd '/aci/tenants'
mocreate 'Cisco'
moconfig commit
# bridge-domain
cd '/aci/tenants/Cisco/networking/bridge-domains'
mocreate 'Cisco'
cd 'Cisco'
moset network 'Cisco'
moconfig commit
# subnet
cd '/aci/tenants/Cisco/networking/bridge-domains/Cisco/subnets'
mocreate '172.16.0.1/24'
moconfig commit
# private-network
cd '/aci/tenants/Cisco/networking/private-networks'
mocreate 'Cisco'
moconfig commit
# application-profile
cd '/aci/tenants/Cisco/application-profiles'
mocreate 'App1'
moconfig commit
# application-epg
cd '/aci/tenants/Cisco/application-profiles/App1/application-epgs'
mocreate 'EPG1'
cd 'EPG1'
moset bridge-domain 'Cisco'
moconfig commit
```

This configuration can also be applied using the following XML posted to the APIC REST API:

XML : Tenant Cisco

```
<fvTenant name="Cisco">
   <fvBD arpFlood="no" multiDstPktAct="bd-flood" name="Cisco" unicastRoute="yes"
unkMacUcastAct="proxy" unkMcastAct="flood">
       <fvRsCtx tnFvCtxName="Cisco"/>
       <fvSubnet ctrl="nd" ip="172.16.0.1/24" name="" preferred="no" scope="private"/>
   </fvBD>
   <fvCtx knwMcastAct="permit" name="Cisco" pcEnfPref="enforced"/>
   <fvAp name="App1" prio="unspecified">
       <fvAEPg name="EPG1">
           <fvRsBd tnFvBDName="Cisco"/>
       </fvAEPg>
   </fvAp>
</fvTenant>
```

Multiple Private Networks with Intra-Tenant Communication

Another use case that may be desirable to support is the option to have a single tenant with multiple private networks. This may be a result of needing to provide multi-tenancy at a network level, but not at a management level. It may also be caused by needing to support overlapping subnets within a single tenant, due to mergers and acquisitions or other business changes.

This method has the following advantages and disadvantages:

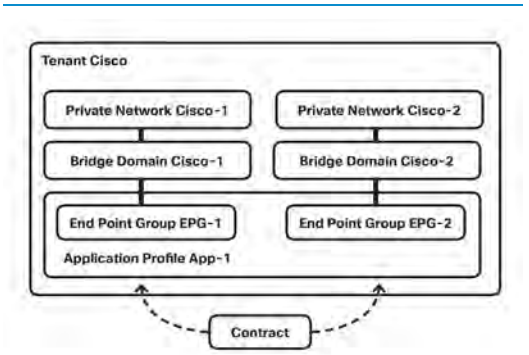
Advantages:

- Ability to have overlapping subnets within a single tenant

Disadvantages:

- EPGs residing in overlapping subnets cannot have policy applied between one another

The object containment for this particular setup can be depicted as shown below:



Multiple Private Networks with Intra-Tenant communication

To create the tenant:

1. On the menu bar, choose **Tenants** > **ADD TENANT.**
2. In the **Create Tenant** dialog box, perform the following actions:
   a. Enter a Tenant **Name.**
3. Click **Next.**
4. Click **Finish.**

The Tenant has been created. Now the tenant administrator can create the private network.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Private Networks.**
4. In the Work pane, choose **Actions** > **Create Private Network.**
5. In the **Create Private Network** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the Private Network.
   b. Click **Next**.
   c. In the **Name** field enter a name for the bridge domain.

d. In the **Subnets** field click **+**.
e. In the **Gateway IP** field enter the IP address for this subnet.
f. In the **Scope** field select **Shared**.
Note: The shared subnet type causes what is known in ACI as a route leak between two Private Networks (VRF).

6 Click **OK.**

7 Click **Finish.**

8 In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Private Networks.**

9 In the Work pane, choose **Actions** > **Create Private Network.**

10 In the **Create Private Network** dialog box, perform the following actions:
a. In the **Name** field enter a name for the second Private Network.
b. Click **Next**.
c. In the **Name** field enter a name for the second bridge domain.
d. In the **Subnets** field click **+**.
e. In the **Gateway IP** field enter the IP address for this subnet.
f. In the **Scope** field select **Shared**.

11 Click **OK**.

12 Click **Finish.**

Now that the two private networks and bridge domains have been created, you can move to the Application Profile.

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Application Profiles.**

4 In the **Action Tab** select **Create Application Profile.**

5 In the **Create Application Profile** dialog box, perform the following actions:
a. In the **Name** field enter a name for the Application Profile.

6 Click **Submit.**

To create the two endpoint groups:

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Application Profiles** > ***Application Profile_Name.***
4. In the Work pane, choose **Actions > Create Application EPG.**
5. In the **Create Application EPG** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the End Point Group.
   b. In the **Bridge Domain** field, select the appropiate bridge domain.
6. Click **Finish.**

Repeat these steps for the second EPG.

The tenant administrator will have to now create a contract and filter between the two EPGs.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Security Policies** > **Filters.**
4. In the Work pane, choose **Actions > Create Filters.**
5. In the **Create Filter** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the Filter.
   b. Click **+**.
   c. In the **Name** column enter **ICMP.**
   d. In the **Ethertype** select **IP.**
   e. In the **IP Protocol** column select **ICMP.**
6. Click **Update.**
7. Click **Submit.**

As the tenant administrator, you would have the knowledge of the filters required to permit traffic across the two EPGs. In the filter, you can repeat the process of defining

various different network protocols as required for your applications. Now you have to define the contract that will be consumed and provided by the two EPGs.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Security Policies** > **Contracts.**
4. In the Work pane, choose **Actions > Create Contract.**
5. In the **Create Contract** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the Filter.
   b. In the **Scope** field select Global.
   c. Click **Subjects** field click **+**.
   d. In the **Name** field enter a name for the **Subject.**
   e. In the Filter Chain field click **+**.
   f. Select the Filter you just created.
6. Click **Update.**
7. Click **OK**.
8. Click **Submit.**

Assign the Contract between the EPGs. A contract is assigned to an EPG as either a consumed or a provided contract. Each EPG that you have created will then either consume or provide that contract to establish a relationship between both EPGs.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Application Profiles** > **Application Profile Name > Application EPGs > *EPG_Name* > Contract.**
4. In the Work pane, choose **Actions > Add Provided Contract.**
5. In the Add Provided Contract dialog box, perform the following actions:
   a. Select the created contract.
   b. Click **Submit**..
6. In the Navigation pane under the second EPG select **Contracts**.
7. In the **Action Tab** select **Add Consume Contract.**
   a. Select the created contract

8 Click **Submit**.

The configuration for this use case can be applied via the following CLI configuration:

CLI : Tenant Cisco

```
# tenant
cd '/aci/tenants'
mocreate 'Cisco'
moconfig commit
# bridge-domain
cd '/aci/tenants/Cisco/networking/bridge-domains'
mocreate 'Cisco'
cd 'Cisco'
moset network 'Cisco'
moconfig commit
# bridge-domain
cd '/aci/tenants/Cisco/networking/bridge-domains'
mocreate 'Cisco1'
cd 'Cisco1'
moset network 'Cisco1'
moconfig commit
# private-network
cd '/aci/tenants/Cisco/networking/private-networks'
mocreate 'Cisco'
moconfig commit
# private-network
cd '/aci/tenants/Cisco/networking/private-networks'
mocreate 'Cisco1'
moconfig commit
# application-profile
cd '/aci/tenants/Cisco/application-profiles'
mocreate 'App1'
moconfig commit
# application-epg
cd '/aci/tenants/Cisco/application-profiles/App1/application-epgs'
mocreate 'EPG1'
cd 'EPG1'
moset bridge-domain 'Cisco'
```

```
moconfig commit
# fv-rscon
cd '/aci/tenants/Cisco/application-profiles/App1/application-
epgs/EPG1/contracts/consumed-contracts'
mocreate 'ICMP'
moconfig commit
# fv-subnet
cd '/aci/tenants/Cisco/application-profiles/App1/application-epgs/EPG1/subnets'
mocreate '172.16.1.1/24'
cd '172.16.1.1:24'
moset scope 'private,shared'
moconfig commit
# application-epg
cd '/aci/tenants/Cisco/application-profiles/App1/application-epgs'
mocreate 'EPG2'
cd 'EPG2'
moset bridge-domain 'Cisco1'
moconfig commit
# fv-rsprov
cd '/aci/tenants/Cisco/application-profiles/App/application-
epgs/EPG2/contracts/provided-contracts'
mocreate 'ICMP'
moconfig commit
# fv-subnet
cd '/aci/tenants/Cisco/application-profiles/CCO/application-epgs/EPG2/subnets'
mocreate '172.16.2.1/24'
cd '172.16.2.1:24'
moset scope 'private,shared'
moconfig commit
```

This configuration can also be applied using the following XML posted to the APIC REST API:

XML : Tenant Cisco

```
<fvTenant dn="uni/tn-Cisco" name="Cisco">
   <vzBrCP name="ICMP" scope="tenant">
       <vzSubj consMatchT="AtleastOne" name="icmp" provMatchT="AtleastOne"
```

```
revFltPorts="yes">
            <vzRsSubjFiltAtt tnVzFilterName="icmp"/>
        </vzSubj>
    </vzBrCP>
    <fvCtx knwMcastAct="permit" name="CiscoCtx" pcEnfPref="enforced"/>
    <fvCtx knwMcastAct="permit" name="CiscoCtx2" pcEnfPref="enforced"/>
    <fvBD arpFlood="yes" mac="00:22:BD:F8:19:FF" name="CiscoBD2" unicastRoute="yes"
unkMacUcastAct="flood" unkMcastAct="flood">
        <fvRsCtx tnFvCtxName="CiscoCtx2"/>
    </fvBD>
    <fvBD arpFlood="yes" mac="00:22:BD:F8:19:FF" name="CiscoBD" unicastRoute="yes"
unkMacUcastAct="flood" unkMcastAct="flood">
        <fvRsCtx tnFvCtxName="CiscoCtx"/>
    </fvBD>
    <fvAp name="CCO">
        <fvAEPg matchT="AtleastOne" name="Web">
            <fvRsCons tnVzBrCPName="ICMP"/>
            <fvRsPathAtt encap="vlan-1201" instrImedcy="immediate" mode="native"
tDn="topology/pod-1/paths-201/pathep-[eth1/16]"/>
            <fvSubnet ip="172.16.2.1/24" scope="private,shared"/>
            <fvRsDomAtt instrImedcy="lazy" resImedcy="lazy" tDn="uni/phys-
PhysDomainforCisco"/>
            <fvRsBd tnFvBDName="CiscoBD2"/>
        </fvAEPg>
        <fvAEPg matchT="AtleastOne" name="App">
            <fvRsPathAtt encap="vlan-1202" instrImedcy="immediate" mode="native"
tDn="topology/pod-1/paths-202/pathep-[eth1/2]"/>
            <fvSubnet ip="172.16.1.1/24" scope="private,shared"/>
            <fvRsDomAtt instrImedcy="lazy" resImedcy="lazy" tDn="uni/phys-
PhysDomainforCisco"/>
            <fvRsBd tnFvBDName="CiscoBD"/>
            <fvRsProv matchT="AtleastOne" tnVzBrCPName="ICMP"/>
        </fvAEPg>
    </fvAp>
 </fvTenant>
```

Multiple Private Networks with Inter-Tenant Communication

This use case may be typical for environments where an ACI administrator wishes to create multiple tenants with the ability to support inter-tenant communications.

This method has the following advantages and disadvantages:

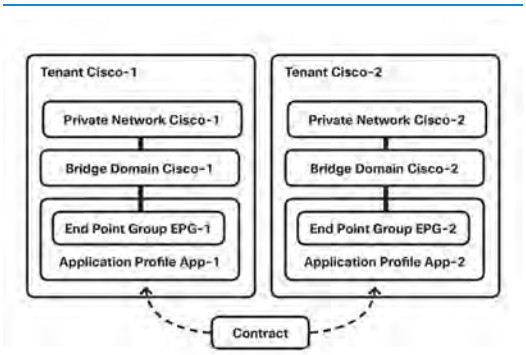
Advantages

- Each tenant container can be managed separately
- Allows for maximum isolation between tenants

Disadvantages

- Tenant address space must be unique

From a containment and relationship perspective, this topology looks as follows:



Multiple Private Networks with Inter-Tenant Communication

To create the tenant:

1. In the GUI Navigate to **Tenants** > **ADD TENANT.**
2. In the Create Tenant dialog box perform the following actions:
   a. In the **Name** field enter a name for the first Tenant.
3. Click **Next.**

4 Click **Finish.**

The tenant has been created. Now the tenant administrator can create the private network.

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation Pane choose ***Tenant_Name*** > **Networking** > **Private Networks.**

4 In the Work pane, choose **Actions > Create Private Network.**

5 In the **Create Private Network** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the Private Network.
   b. Click **Next**.
   c. In the **Name** field enter a name for the bridge domain.
   d. In the **Subnets** field click **+**.
   e. In the **Gateway IP** field enter the IP address for this subnet.
   f. In the **Scope** field select **Shared**.
   Note: The shared subnet type causes what is known in ACI as a route leak between two Private Networks (VRF).

6 Click **OK**.

7 Click **Finish**.

To create the application profile:

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation Pane choose ***Tenant_Name*** > **Networking** > **Application Profiles.**

4 In the Work pane, choose **Actions > Create Application Profile.**

5 In the Create Application Profile dialog box, perform the following actions:
   a. In the **Name** field enter a name for the Application Profile.

6 Click **Submit.**

To create the endpoint group:

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Application Profiles** > ***Application Profile_Name***
4. In the Work pane, choose **Actions** > **Create Application EPG.**
5. In the Create Application EPG dialog box, perform the following actions:
   a. In the **Name** field enter a name for the End Point Group.
   b. In the **Bridge Domain** field, select the appropiate bridge domain.
6. Click **Finish.**

To create the second tenant and application profile:

1. In the GUI Navigate to **Tenants** > **ADD TENANT.**
2. In the Create Tenant dialog box perform the following actions:
   a. In the **Name** field enter a name for the first Tenant.
3. Click **Next.**
4. Click **Finish.**

The tenant has been created. Now the tenant administrator can create the private network.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Private Networks.**
4. In the Work pane, choose **Actions** > **Create Private Network.**
5. In the **Create Private Network** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the Private Network.
   b. Click **Next**.
   c. In the **Name** field enter a name for the bridge domain.
   d. In the **Subnets** field click **+**.
   e. In the **Gateway IP** field enter the IP address for this subnet.

f. In the **Scope** field select **Shared**.
Note: The shared subnet type causes what is known in ACI as a route leak between two Private Networks (VRF).

6 Click **OK.**

7 Click **Finish**.

To create the application profile:

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Application Profiles.**

4 In the Work pane, choose **Actions > Create Application Profile.**

5 In the Create Application Profile dialog box, perform the following actions:
a. In the **Name** field enter a name for the Application Profile.

6 Click **Submit.**

To create the endpoint group:

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Application Profiles > *Application Profile_Name***

4 In the Work pane, choose **Actions > Create Application EPG.**

5 In the Create Application EPG dialog box, perform the following actions:
a. In the **Name** field enter a name for the End Point Group.
b. In the **Bridge Domain** field, select the appropiate bridge domain.

6 Click **Finish.**

The tenant administrator will have to now create a contract and filter between the two EPGs.

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Security Policies** > **Filters**.
4. In the Work pane, choose **Actions > Create Filters.**
5. In the **Create Filter** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the Filter.
   b. Click **+**.
   c. In the **Name** column enter **ICMP.**
   d. In the **Ethertype** select **IP.**
   e. In the **IP Protocol** column select **ICMP.**
6. Click **Update.**
7. Click **Submit.**

As the tenant administrator you would have the knowledge of the filters required to permit traffic across the two EPGs. In the filter you can repeat the process of defining various different network protocols as required for your applications. Now you have to define the contract that will be consumed and provided by the two EPGs.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Networking** > **Security Policies** > **Contracts.**
4. In the Work pane, choose **Actions > Create Contract.**
5. In the **Create Contract** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the filter.
   b. In the **Scope** field choose **Global**.
   c. Click **Subjects** field click **+**.
   d. In the **Name** field enter a name for the **Subject.**
   e. In the **Filter Chain** field click **+**.
   f. Select the filter you just created.
6. Click **Update.**
7. Click **OK**.
8. Click **Submit.**

Assign the Contract between the EPGs. A contract is assigned to an EPG as either a consumed or a provided contract. Each EPG that you have created will then either consume or provide that contract to establish a relationship between both EPGs.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** > **Application Profiles** > **Application Profile Name > Application EPGs > *EPG_Name* > Contracts.**
4. In the Work pane, choose **Actions > Add Provided Contract.**
5. In the **Add Provided Contract** dialog box, perform the following actions:
   a. Select the created ***Contract***.
   b. Click **Submit**.
6. In the GUI Navigate to **Tenants** > **ALL TENANTS.**
7. Select the **second** created Tenant.
8. In the Navigation pane choose ***Tenant_Name*** > **Application Profiles** > **Application Profile Name > Application EPGs > *EPG_Name* > Contracts.**
9. In Work pane, choose **Action > Add Consume Contract.**
   a. Select the created ***Contract***.
10. Click **Submit**.

The configuration for this use case can be applied via the following CLI configuration:

CLI : TENANT Cisco1

```
# tenant
cd '/aci/tenants'
mocreate 'Cisco1'
moconfig commit
# bridge-domain
cd '/aci/tenants/Cisco1/networking/bridge-domains'
mocreate 'Cisco1'
cd 'Cisco1'
moset network 'Cisco1'
moconfig commit
```

```
# private-network
cd '/aci/tenants/Cisco1/networking/private-networks'
mocreate 'Cisco1'
moconfig commit
# application-profile
cd '/aci/tenants/Cisco1/application-profiles'
mocreate 'App1'
moconfig commit
# application-epg
cd '/aci/tenants/Cisco1/application-profiles/App1/application-epgs'
mocreate 'EPG1'
cd 'EPG1'
moset bridge-domain 'Cisco1'
moconfig commit
# fv-rsprov
cd '/aci/tenants/Cisco/application-profiles/CCO/application-
epgs/App/contracts/provided-contracts'
mocreate 'ICMP'
moconfig commit
# fv-subnet
cd '/aci/tenants/Cisco/application-profiles/CCO/application-epgs/App/subnets'
mocreate '172.16.1.1/24'
cd '172.16.1.1:24'
moset scope 'private,shared'
moconfig commit
# contract
cd '/aci/tenants/Cisco/security-policies/contracts'
mocreate 'ICMP'
cd 'ICMP'
moset scope 'global'
moconfig commit
# contract-subject
cd '/aci/tenants/Cisco/security-policies/contracts/ICMP/subjects'
mocreate 'icmp'
moconfig commit
# vz-rssubjfiltatt
cd '/aci/tenants/Cisco/security-policies/contracts/ICMP/subjects/icmp/common-filters'
mocreate 'icmp'
moconfig commit
```

CLI : TENANT Cisco2

```
 # tenant
 cd '/aci/tenants'
 mocreate 'Cisco'
 moconfig commit
 # bridge-domain
 cd '/aci/tenants/Cisco/networking/bridge-domains'
 mocreate 'Cisco'
 cd 'Cisco'
 moset network 'Cisco'
 moconfig commit
 # private-network
 cd '/aci/tenants/Cisco/networking/private-networks'
 mocreate 'Cisco'
 moconfig commit
 # application-profile
 cd '/aci/tenants/Cisco/application-profiles'
 mocreate 'App1'
 moconfig commit
 # application-epg
 cd '/aci/tenants/Cisco2/application-profiles/App1/application-epgs'
 mocreate 'EPG1'
 cd 'EPG1'
 moset bridge-domain 'Cisco'
 moconfig commit
 # fv-rsconsif
 cd '/aci/tenants/Cisco1/application-profiles/CCO/application-
epgs/Web/contracts/consumed-contract-interfaces'
 mocreate 'CiscoInterTenantICMP'
 moconfig commit
 # fv-subnet
 cd '/aci/tenants/Cisco1/application-profiles/CCO/application-epgs/Web/subnets'
 mocreate '172.16.2.1/24'
 cd '172.16.2.1:24'
 moset scope 'shared-subnet'
 moconfig commit
 # imported-contract
 cd '/aci/tenants/Cisco1/security-policies/imported-contracts'
 mocreate 'CiscoInterTenantICMP'
 cd 'CiscoInterTenantICMP'
 moset contract 'tenants/Cisco/security-policies/contracts/ICMP'
 moconfig commit
```

This configuration can also be applied using the following XML posted to the APIC REST API:

XML : TENANT Cisco1

```
<fvTenant dn="uni/tn-Cisco1" name="Cisco1">
```

```
    <vzBrCP name="ICMP" scope="global">
        <vzSubj consMatchT="AtleastOne" name="icmp" provMatchT="AtleastOne"
revFltPorts="yes">
            <vzRsSubjFiltAtt tnVzFilterName="icmp"/>
        </vzSubj>
    </vzBrCP>
```

```
    <vzCPIf dn="uni/tn-Cisco1/cif-ICMP" name="ICMP">
```

```
        <vzRsIf consMatchT="AtleastOne" name="icmp" provMatchT="AtleastOne"
revFltPorts="yes">
            <vzRsSubjFiltAtt tDn="uni/tn-Cisco2/brc-default"/>
        </vzRsIf>
    </vzCPIf>
    <fvCtx knwMcastAct="permit" name="CiscoCtx" pcEnfPref="enforced"/>
```

```
    <fvBD arpFlood="yes" mac="00:22:BD:F8:19:FF" name="CiscoBD2" unicastRoute="yes"
unkMacUcastAct="flood" unkMcastAct="flood">
        <fvRsCtx tnFvCtxName="CiscoCtx2"/>
    </fvBD>
    <fvBD arpFlood="yes" name="CiscoBD" unicastRoute="yes" unkMacUcastAct="flood"
unkMcastAct="flood">
        <fvRsCtx tnFvCtxName="CiscoCtx"/>
    </fvBD>
    <fvAp name="CCO">
        <fvAEPg matchT="AtleastOne" name="EPG1">
```

```
            <fvRsPathAtt encap="vlan-1202" instrImedcy="immediate" mode="native"
tDn="topology/pod-1/paths-202/pathep-[eth1/2]"/>
            <fvSubnet ip="172.16.1.1/24" scope="private,shared"/>
```

```
            <fvRsDomAtt instrImedcy="lazy" resImedcy="lazy" tDn="uni/phys-
PhysDomainforCisco"/>
```

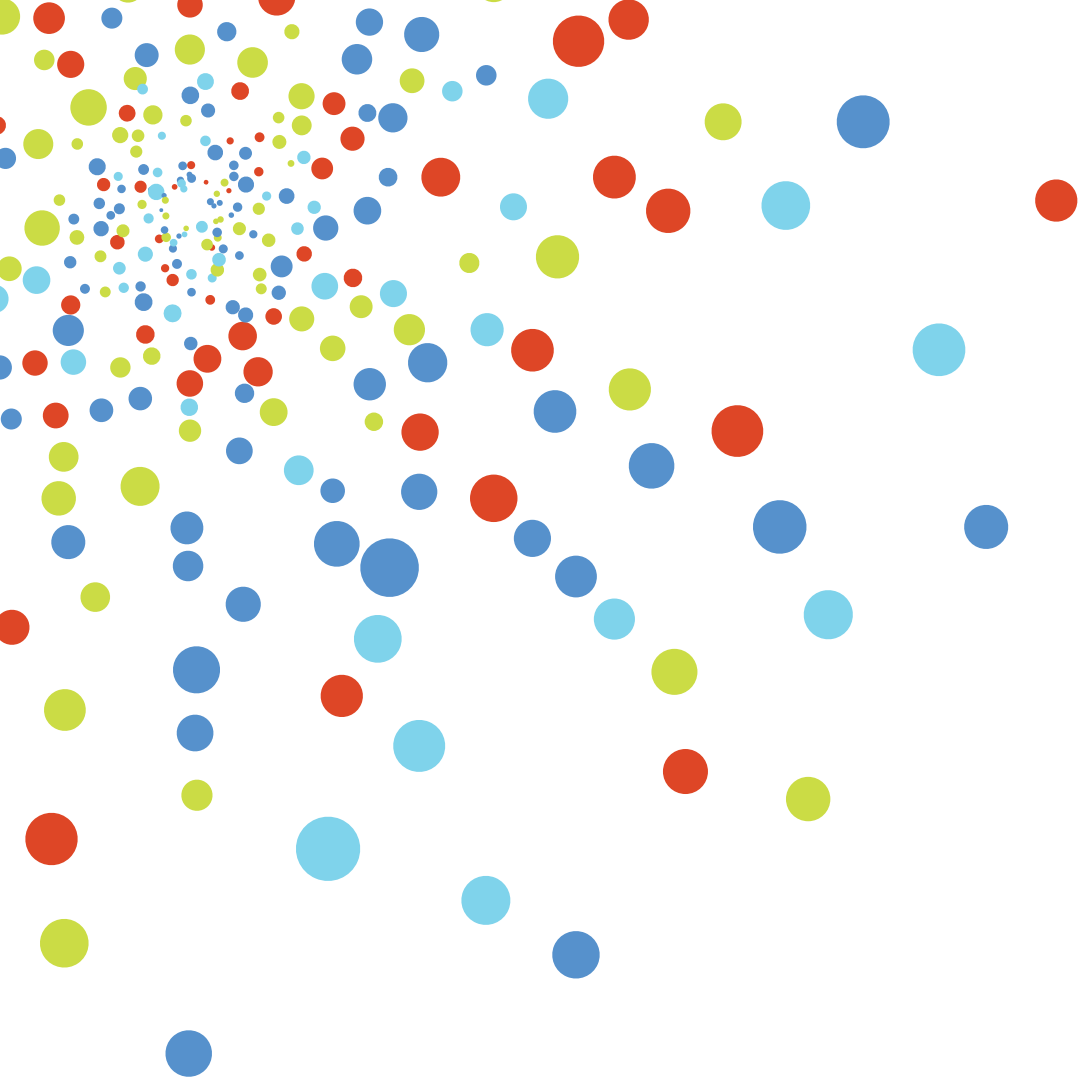
```
            <fvRsBd tnFvBDName="CiscoBD"/>
            <fvRsProv matchT="AtleastOne" tnVzBrCPName="ICMP"/>
        </fvAEPg>
    </fvAp>
 </fvTenant>
```

XML : TENANT Cisco2

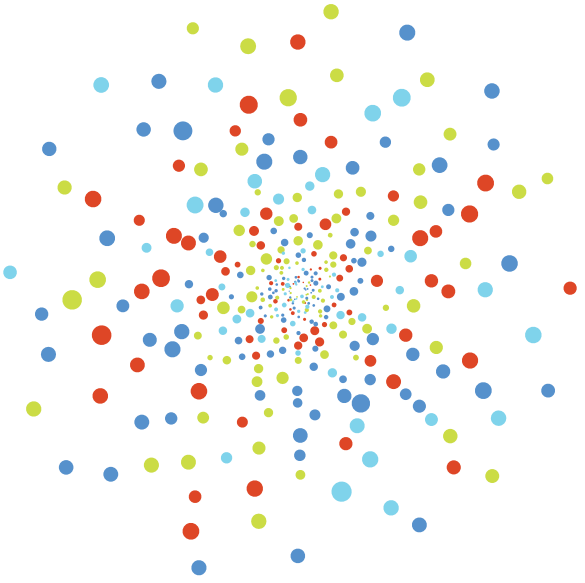
```
 <fvTenant dn="uni/tn-Cisco2" name="Cisco2">
    <fvCtx knwMcastAct="permit" name="CiscoCtx" pcEnfPref="enforced"/>
    <fvBD arpFlood="yes" mac="00:22:BD:F8:19:FF" name="CiscoBD2" unicastRoute="yes"
unkMacUcastAct="flood" unkMcastAct="flood">
        <fvRsCtx tnFvCtxName="CiscoCtx"/>
    </fvBD>
    <fvBD arpFlood="yes" name="CiscoBD" unicastRoute="yes" unkMacUcastAct="flood"
unkMcastAct="flood">
        <fvRsCtx tnFvCtxName="CiscoCtx"/>
    </fvBD>
    <fvAp name="CCO">
        <fvAEPg matchT="AtleastOne" name="EPG2">
            <fvRsPathAtt encap="vlan-1202" instrImedcy="immediate" mode="native"
tDn="topology/pod-1/paths-201/pathep-[eth1/2]"/>
            <fvSubnet ip="172.16.1.1/24" scope="private,shared"/>
```

```
            <fvRsDomAtt instrImedcy="lazy" resImedcy="lazy" tDn="uni/phys-
PhysDomainforCisco"/>
```

```
            <fvRsBd tnFvBDName="CiscoBD"/>
            <fvRsConsIf matchT="AtleastOne" tnVzBrCPIfName="ICMP"/>
        </fvAEPg>
    </fvAp>
</fvTenant>
```



Working with Contracts



Section Content

- Contracts
  - Contract Configuration Parameters
  - Create/Modify/Remove Contracts
    - Create Contracts
    - Modify Contracts
  - Remove Contracts
  - Verify Contracts
- Apply/Remove EPG Contracts
  - Apply a Contract to an EPG
  - Remove a Contract from EPG
  - Verify Contract on EPG
- Apply/Remove External Network Contracts
  - Apply a Contract to an External Network
  - Remove a Contract from an External Network
  - Verify External Network Contracts
- Apply/Remove Private Network Contracts
  - Apply a Contract to a Private Network (vzAny)
  - Remove a Contract From a Private Network (vzAny)
  - Verify Private Network Contracts
- Filters
  - Filter Entry Configuration Parameters
  - Create Filters
  - Modify Filters
  - Remove Filters
  - Verify Filters

- Taboo Contracts
  - Taboo Contract Configuration Parameters
  - Create, Modify, or Delete Taboo Contracts
    - Create Taboo Contracts
    - Modify Taboo Contracts
    - Delete Taboo Contracts
    - Verify Taboo Contracts

- Apply/Remove Taboo Contracts
  - Apply Taboo Contract to an EPG
  - Remove Taboo Contract from EPG
  - Verify Taboo Contracts Applied to EPG

- Inter-Tenant Contracts
  - Configuration Parameters

- Create/Modify/Remove Export Contracts
  - Export Contract
  - Modify Exported Contracts
  - Remove Exported Contracts
  - Verify Exported Contracts

- Contracts Use Cases
  - Inter-Tenant Contracts
    - Tenant Cisco-1/EPG-1
    - Tenant Cisco-2/EPG-2

- Inter-Private Network Contracts Communication
  - Tenant Cisco-1/EPG-1
  - Tenant Cisco-1/EPG-2

- Single Contract Bidirectional Reverse Filter
- Single Contract Unidirectional with Multiple Filters
- Multiple Contracts Uni-Directional Single Filter

Contracts

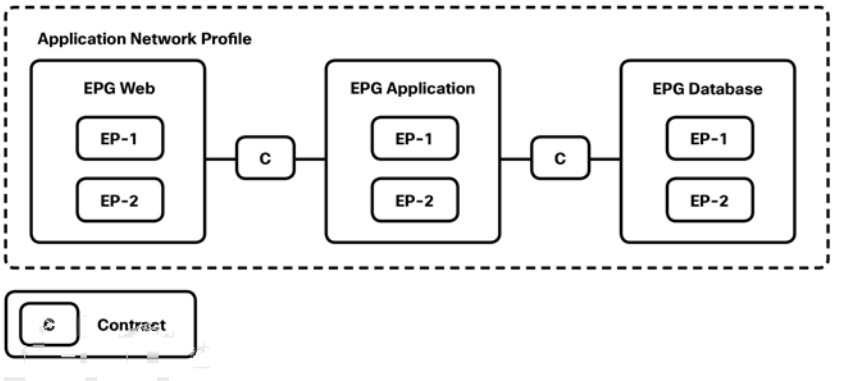
Contracts provide a way for the ACI administrator to control traffic flow within the ACI fabric between EPGs. These contracts are built using a provider-consumer model where one EPG provides the services it wants to offer and another EPG consumes them.

Contracts are comprised of the following items:

- Subjects - A group of filters for a specific application or service
- Filters - Used to classify traffic based upon layer 2 to layer 4 attributes (such as Ethernet type, protocol type, TCP flags and ports)
- Actions - Permit, deny, mark, log, redirect, copy, and service graph to perform matches based upon filters
- Labels - Used optionally to group objects such as subjects and EPGs for the purpose of further defining policy enforcement

EPGs can only communicate with other EPGs based upon the contract rules defined. There is no contract required for intra-EPG communication: intra-EPG communication is allowed by default.

The example below, shows how contracts would control traffic flow between EPGs in a 3-tiered application. The Web EPG provides a contract which is consumed by the Application EPG, and the Application EPG provides a contract which the Database EPG would consume. EPGs may acts a both a provider and consumer of the same contract easily by making the contract bidirectional.



Contract Policies Between End Point Groups

Contracts govern the following types of endpoint group communications:

- Between application EPGs
- Between application EPGs and external networks
- Between application EPGs and in-band management EPG, for example if in-band management is configured for the ACI fabric and certain EPGs are to be allowed to access it

Contract Configuration Parameters

When configuring contracts you can define the following options:

Contract Scope - The scope of a service contract between two or more participating peer entities.

The states are:

- **Context -** This contract will be applied for endpoint groups associated with the same private network.
- **Application-profile -** This contract will be applied for endpoint groups in the same application profile.
- **Tenant -** This contract will be applied for endpoint groups within the same tenant.

- **Global –** This contract will be applied for endpoint groups throughout the fabric.

The default is **Context**.

QoS Class - The priority level of the service contract.

The priority level can be:

- **Unspecified**
- **Level1 –** Class 1 Differentiated Services Code Point (DSCP) value.
- **Level2 –** Class 2 DSCP value.
- **Level3 –** Class 3 DSCP value.

The default is **Unspecified**.

Tags - The search keyword or term that is assigned to the application profile. A tag allows you to group multiple objects by a descriptive name. You can assign the same tag name to multiple objects and you can assign one or more tag names to an object.

Match - The subject match criteria across consumers. The options are:

- **AtleastOne**
- **AtmostOne**
- **None**
- **All**

The default is **AtleastOne**.

Create/Modify/Remove Contracts

Create Contracts

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Contracts**.

4 In the Work pane, choose **Actions > Create Contract**.

5 In the **Create Contract** dialog box, perform the following actions:
   a. Enter a Contract **Name**.
   b. Choose a Contract **Scope** (optional).
   c. Choose a **QoS Class** (optional).
   d. Click **+** next to the **Subject** to add a Contract Subject.
      i. In the **Create Contract Subject** dialog box, perform the following actions:
         1. Enter a Contract Subject **Name**.
         2. Click **+** in the **Filter Chain** field.
            For information regarding filter creation, see the "Filters" section.

6 Click **Update.**

7 Click **OK.**

8 Click **Submit**.

Modify Contracts

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Contracts >** ***Contract_Name***.

4 In the Work pane, choose the **Policy** tab.
   a. Choose a **Contract Scope** (optional).
   b. Choose a **Qos Class** (optional).
   c. Click **+** next to the **Subject** field. to add a Contract Subject.
      i. In the **Create Contract Subject** dialog box, perform the following actions:
         1. Enter a Contract Subject **Name**.
         2. Click **+** next to **Filter Chain**.
            Note: For information regarding filter creation, see the "Filters" section.

5 Click **Update.**

6 Click **OK.**

7 Click **Submit**.

Remove Contracts

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Contracts > *Contract_Name***.
4. In the Work pane, choose **Actions > Delete**.

Verify Contracts

```
REST :: /api/node/class/vzBrCP.xml
 CLI :: moquery -c vzBrCP
```

Apply/Remove EPG Contracts

Apply a Contract to an EPG

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the **Tenant_Name**.
3. In the Navigation pane choose ***Tenant_Name*** **> Application Profiles > *Application_Profile_Name*** **> Application EPGs > *EPG_Name*** **> Contracts**.
4. In the Work pane, choose **Actions > Add Provided Contract** or **Actions > Add Consumed Contract**.
   Note: Choose the action depending on how the contract is to be deployed.
5. In the **Add Contract** dialog box, perform the following actions:
   a. Enter a **Contract_Name**.
   b. Choose a **QOS** policy (optional).
   c. Choose a **Label** (optional).
6. Click **Submit**.

Remove a Contract from EPG

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Application Profiles > *Application_Profile_Name*** **.> Application EPGs > *EPG_Name*** **> Contracts > *Contract_Name***.

4 In the Work pane, choose **Actions > Delete**.

Verify Contract on EPG

```
Provider
REST :: /api/node/class/fvRsProv.xml
 CLI :: moquery -c fvRsProv
```

```
Consumer
REST :: /api/node/class/fvRsCons.xml
 CLI :: moquery -c fvRsCons
```

Apply/Remove External Network Contracts

Apply a Contract to an External Network

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Networking > External Routed Networks > *Routed Outside_Name*** **> Networks > *External_Network_Instance_Profile***.

4 In the Work pane, click **+** next to either **Add Provided Contract** or **Add Consumed Contract**.
Note: Make a selection depending on how the contract is to be deployed.
   a. Choose a ***Contract_Name***.
   b. Choose a **QOS Type.**
   c. Choose a **Match Criteria.**

5 Click **Update**.

Remove a Contract from an External Network

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Networking > External Routed Networks >** ***Routed Outside_Name*** **> Networks > *External_Network_Instance_Profile***.
4. In the Work pane, choose the **Contract_Name** and click **x**.

Verify External Network Contracts

```
Provider
REST :: /api/node/class/fvRsProv.xml
 CLI :: moquery -c fvRsProv
```

```
Consumer
REST :: /api/node/class/fvRsCons.xml
 CLI :: moquery -c fvRsCons
```

Apply/Remove Private Network Contracts

In order to apply contracts to all endpoint groups within a private network, contracts can be applied directly to the private network. This concept is also referred as "vzAny" endpoint group. It eases contract management by allowing the contract configuration for all endpoint groups within a private network from a single location as well as optimizing hardware resource consumption.

For instance, if an ACI Administration has 100 EPGs that are all part of the same private network, they can apply the contracts to this one vzAny group under the private network, rather than to each EPG.

Apply a Contract to a Private Network (vzAny)

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Networking > Private Networks >** ***Private_Network_Name*** **>** ***EPG Collection for Context***.

4 In the Work pane, click **+** next to either **Add Provided Contract** or **Add Consumed Contract**.

Note: Make a selection depending on how the contract is to be deployed.

a. Enter a ***Contract_Name***.

b. Choose a **QOS Type.**

c. Choose a **Match Criteria.**

Click **Update.**

Remove a Contract From a Private Network (vzAny)

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Networking > Private Networks >** ***Private_Network_Name*** **>** ***EPG Collection for Context***.

4 In the Work pane, choose the ***Contract_Name*** and click **x**.

Verify Private Network Contracts

```
REST :: /api/node/class/vzBrCP.xml
 CLI :: moquery -c vzBrCP
```

Filters

A filter policy is a group of resolvable filter entries. Each filter entry is a combination of network traffic classification properties.

Filters are Layer 2 to Layer 4 fields, TCP/IP header fields such as Layer 3 protocol type, Layer 4 ports, and so on. According to its related contract, an EPG provider dictates the protocols and ports in both the in and out directions. Contract subjects contain associations to the filters (and their directions) that are applied between EPGs that produce and consume the contract. See the Use Cases below for an example.

Filter Entry Configuration Parameters

When configuring a Filter, the following options can be defined:

Name - The name of a filter entry.

EtherType - The EtherType of the filter entry. The EtherTypes are:

- **ARP**
- **FCOE**
- **IP**
- **MAC Security**
- **MPLS Unicast**
- **Trill**
- **Unspecified**

  The default is **ARP**.

Arp Flag - The Address Resolution Protocol flag for a filter entry. The filter entry is a combination of network traffic classification properties.

IP Protocol - The IP protocol for a filter entry. The filter entry is a combination of network traffic classification properties.

Allow Fragment - The start of the source port range. The start of the port range is determined by the server type. The range is 0 to 0xffff. The port can be for the following server types:

Source Port: From - The start of the source port range. The start of the port range is determined by the server type. The range is 0 to 0xffff. The port can be for the following server types:

- **Unspecified**
- **ftpData**
- **SMTP**
- **DNS**
- **HTTP**
- **POP3**
- **HTTPS**
- **RTSP**

Source Port: To - The end of the source port range. The end of the port range is determined by the server type. The range is 0 to 0xffff. The port can be for the following server types:

- **Unspecified**
- **ftpData**
- **smtp**
- **DNS**
- **HTTP**
- **POP3**
- **HTTPS**
- **RTSP**

Destination Port: From - The end of the destination port range. The end of the port range is determined by the server type. The range is 0 to 0xffff. The port state settings are:

- **Unspecified**
- **ftpData**

- **SMTP**
- **DNS**
- **HTTP**
- **POP3**
- **HTTPS**
- **RTSP**

Destination Port: To - The start of the destination port range. The port range is determined by the server type. The range is 0 to 0xffff. The destination port can be set for the following server types:

- **unspecified**
- **ftpData**
- **smtp**
- **dns**
- **http**
- **pop3**
- **https**
- **rtsp**

The default is **unspecified**.

TCP Session Rules - The TCP session rules for a filter entry. The filter entry is a combination of network traffic classification properties.

Create Filters

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Filters**.
4. In the Work pane, choose **Actions > Create Filter**.
5. In the **Create Filter** dialog box, perform the following actions:
   a. Enter an Filter **Name**.
   b. Click **+** to add a **Filter Entry.**

i. In the **Filter Entry** dialog box, perform the following actions:
   1. Enter a Filter Entry **Name**.
   2. In the drop-down list select an **Ethertype.**
   3. In the drop-down list select an **ARP Flag** (optional)**.**
   4. In the drop-down list select an **IP Protocol** (optional)**.**
   5. If required, check the **Allow Fragment** check box (optional).
   6. In the drop-down list select the **Source Port From** (optional).
   7. In the drop-down list select the **Source Port To** (optional).
   8. In the drop-down list select the **Destination Port From** (optional).
   9. In the drop-down list select the **Destination Port To** (optional).
   10. In the drop-down list select the **TCP Session Rules** (optional).

6 Click **Update**.

7 Click **Submit**.

Modify Filters

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Filters >** ***Filter_Name***.

4 In the Work pane, double click the **filter** entry.
   a. In the drop-down list select an **Ethertype** (optional).
   b. In the drop-down list select an **ARP Flag** (optional).
   c. In the drop-down list select an **IP Protocol** (optional).
   d. If required, click the **Allow Fragment** check box (optional).
   e. In the drop-down list select the **Source Port From** (optional).
   f. In the drop-down list select the **Source Port To** (optional).
   g. In the drop-down list select the **Destination Port From** (optional).
   h. In the drop-down list select the **Destination Port To** (optional).
   i. In the drop-down list select the **TCP Session Rules** (optional).

5 Click **Update**.

6 Click **Submit**.

Remove Filters

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Filters >** ***Filter_Name***.
4. In the Work pane, choose **Actions > Delete**.

Verify Filters

```
REST :: /api/node/class/vzFilter.xml
 CLI :: moquery -c vzFilter
```

Taboo Contracts

There may be times when the ACI administrator might need to deny traffic that is allowed by another contract. Taboos are a special type of contract that an ACI administrator can use to deny specific traffic that would otherwise be allowed by another contract. Taboos can be used to drop traffic matching a pattern (any EPG, a specific EPG, matching a filter, and so forth). Taboo rules are applied in the hardware before the rules of regular contracts are applied.

Taboo contracts are not recommended as part of the ACI best practices but they can be used to transition from traditional networking to ACI. To imitate the traditional networking concepts, an "allow-all-traffic" contract can be applied, with taboo contracts configured to restrict certain types of traffic.

Taboo Contract Configuration Parameters

When configuring Taboo Contracts you can define the following options:

Name - The name of the contract or contract object.

Subjects - The network domain name label. Labels enable classification of the objects which can and cannot communicate with one another (optional).

Directive - The filter directives assigned to the taboo contract.

Create, Modify, or Delete Taboo Contracts

Create Taboo Contracts

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Taboo Contracts**.

4 In the Work pane, choose **Action > Create Taboo Contract**.

5 In the **Create Taboo Contract** dialog box, perform the following actions:
   a. Enter a Taboo Contract **Name**.
   b. Click **+** to next to the **Subject** field to add a Taboo Subject.
      i. Enter a Filter **Name**.
      ii. Choose **Directives**.

6 Click **Update**.

7 Click **OK**.

8 Click **Submit**.

Modify Taboo Contracts

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Taboo Contracts >** ***Taboo_Contract_Name***.

4 In the Work pane, choose **policy.**
   a. Click **+** to next to the **Subject** field.
   b. In the **Create Taboo Contract Subject** dialog box, perform the following actions:
      i. Enter a Taboo Contract Subject **Name**.
      ii. Click **+** in the **Filter Chain** field.
         1. Enter a Filter **Name**.
         2. Choose **Directives**.

5 Click **Submit**.

Delete Taboo Contracts

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Taboo Contracts >** ***Taboo_Contract_Name***.

4 In the Work pane, choose **Action > Delete**.

Verify Taboo Contracts

```
REST :: /api/node/class/vzTaboo.xml
 CLI :: moquery -c vzTaboo
```

Apply/Remove Taboo Contracts

Apply Taboo Contract to an EPG

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Application Profiles > *Application_Profile_Name*** **> Application EPGs > *EPG_Name*** **> Contracts**.

4 In the Work pane, choose **Actions > Add Taboo Contract**.

5 In the **Add Taboo Contract** dialog box,
   a. Choose the **Taboo** Contract.

6 Click **Submit**.

Remove Taboo Contract from EPG

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Application Profiles > *Application_Profile_Name*** **> Application EPGs > *EPG_Name*** **> Contracts**.

4 In the Work pane, choose the Taboo ***Contract_Name*** **> Actions > Delete**.

Verify Taboo Contracts Applied to EPG

```
Provider
REST :: /api/node/class/fvRsProv.xml
CLI :: moquery -c fvRsProv
```

```
Consumer
REST :: /api/node/class/fvRsCons.xml
 CLI :: moquery -c fvRsCons
```

Inter-Tenant Contracts

There may be times when the ACI administrator might need to allow traffic between two tenants. Interface contracts are a special type of contract that an ACI administrator can use to allow specific traffic through the use of a contract export. The contract in essence is exported in the source tenant and imported into the target tenant. Similar to traditional contracts, the source EPG will be of type provider. However, in the target tenant, the contract is imported as type contract interface. Some use case examples show the complete process in the next chapter.

Configuration Parameters

When Importing a Contract, the following options can be defined:

Name - The name of the contract interface.

Global Contract - Name of a service contract to be shared between two or more participating peer entities.

Tenant - The Tenant name of the targeted Export contract.

Create/Modify/Remove Export Contracts

Export Contract

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Contracts.**
4. In the Work pane, choose **Actions > Export Contract**.
5. In the **Export Contract** dialog box, perform the following actions:
   a. Enter an Export Contract **Name**.
   b. Choose the **Global Contract**.

c. Enter the **Tenant Name**.

6 Click **Finish**.

Modify Exported Contracts

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Contracts > *Contract_Name.***

4 In the Work pane, choose **policy**.

a. Enter an Export Contract **Name**.

b. Choose the **Global Contract**.

c. Enter the **Tenant Name**.

5 Click **Finish**.

Remove Exported Contracts

1 On the menu bar, choose **Tenants** > **ALL TENANTS.**

2 In the Work pane, choose the ***Tenant_Name***.

3 In the Navigation pane choose ***Tenant_Name*** **> Security Policies > Contracts > Imported Contracts > *Contact_Name.***

4 In the Work pane, choose **Actions > Delete**.

Verify Exported Contracts

```
REST :: /api/node/class/vzCPif.xml
 CLI :: moquery -c vzCPif
```

Contracts Use Cases

These use cases all assume the objective is for a host in EPG-1 to talk to a host in EPG-2, achieving bidirectional traffic. How these scenarios are implemented will depend on the operational model chosen, and whether the system is more focused on object re-use or tenant autonomy. Review the contracts section on Contract Scoping for a more detailed discussion.

There are four common scenarios:

1. Inter-Tenant Contracts.
2. Inter-Private Network Contracts.
3. Single Contract Bidirectional forwarding with reverse filter.
4. Single Contract Unidirectional with multiple Filters.
5. Multiple Contracts Unidirectional with single Filter.

Inter-Tenant Contracts

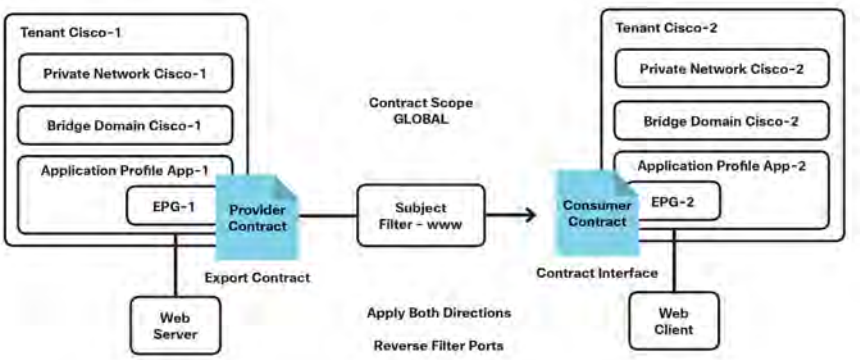
ACME Inc., as with most companies, makes use of shared services such as DNS for name resolution and Active Directory for user management. These services will be used across most of their tenants and so ACME Inc. must allow this traffic across the whole fabric. Communication between EPGs that belong to different tenants is only allowed when they share the same contract. To use the same contract, it will need to be exported from the source tenant to the appropriate destination tenant. That contract will appear under the Imported Contract section in the Security Policies of the destination tenant.

A Consumed Contract Interface will be used to associate an EPG from the destination tenant with the imported contract.

Note: A contract consumption interface represents one or more subjects defined under the contract. By associating to an interface, an endpoint group starts consuming all the subjects represented by the interface.

In the use case below, EPG-1 in tenant Cisco-1 requires communication with EPG-2 in tenant Cisco-2. This is accomplished by utilizing contact interfaces. In tenant Cisco-1

the user will export the intended contract interfaces. In tenant Cisco-1 the user will export the intended contract and select provider to provide the contrast to EPG-2. The user will then confirm the imported contract in tenant Cisco-2 and select the contract as consumed. To advertise the routes from the source VRF to the intended VRF, the user must create the subnet within the EPG.



Exporting Contracts Between Tenants

Tenant Cisco-1/EPG-1

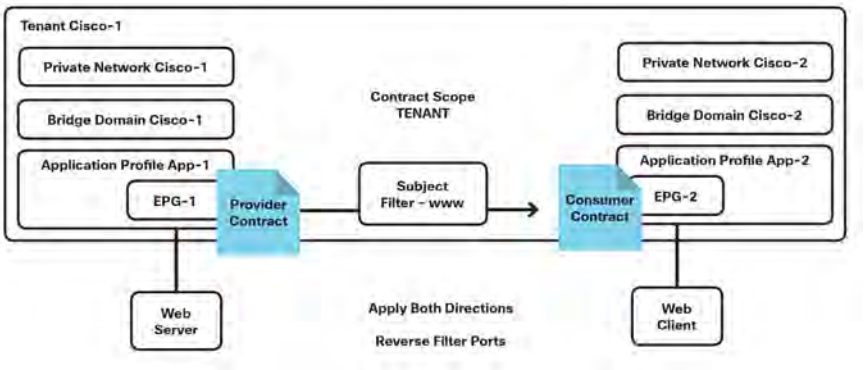
1. Create an **Export Contract** under security policies.
2. Create the host subnet (default Gateway IP) under EPG1 - subnet scope **shared**.
3. Add the Contract under EPG1 - contract type **provider**.
4. Create the host subnet under the bridge domain - subnet scope **private/public**.

Tenant Cisco-2/EPG-2

1. Confirm the exported contract is listed under **Imported Contracts**.
2. Create the host subnet (default Gateway IP) under EPG2 - subnet scope **shared.**
3. Add the Interface Contract under EPG2 - contract type **consumed**.
4. Create the host subnet (default Gateway IP) under the bridge domain - subnet scope **private/public.**

Inter-Private Network Contracts Communication

In the use case below, EPG-1 in VRF Cisco-1 requires communication with EPG-2 in VRF Cisco-2. This is accomplished by utilizing the subnet field within the EPG. By creating the subnet under the EPG and selecting shared, the route will be leaked to the VRF noted within the Tenant scoped contract.



Exporting Contracts Between Private Networks

1 Create the contract under **Security Policies** - contract scope **Tenant**.

Tenant Cisco-1/EPG-1

1 Create the host subnet (default Gateway IP) under **EPG1** - subnet scope shared.

2 Add the Contract under **EPG1** - contract type provider.

Tenant Cisco-1/EPG-2

1 Create the host subnet (default Gateway IP) under **EPG2** - subnet scope shared.

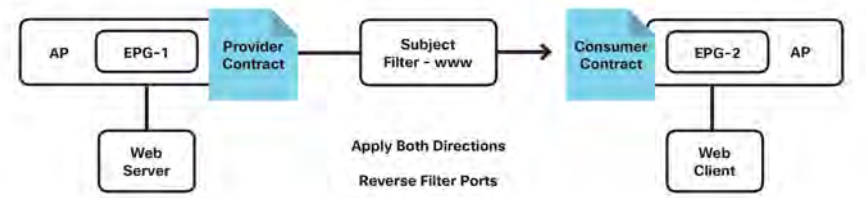
2 Add the Contract under **EPG2** - contract type provider.

Single Contract Bidirectional Reverse Filter

This use case is useful when implementing a contract with the option to apply the contract subject in both directions and with the option to apply the reverse filter. This is

the most common of the use cases and allows for a single subject/filter to be implemented with a single Provider/Consumer relationship.

In the use case below, EPG-1 is providing a contract with a subject of www and EPG-2 is consuming the contract. This allows the Web Client in EPG-2 to access the Web Server in EPG-1. i.e. EPG-1 is providing a service to EPG-2.



Default Bi-directional Contract with Reverse Filter

Result:

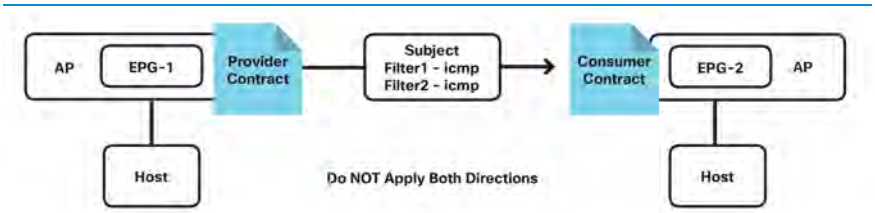
A single contract with (1) Subject and (1) Filter with a single provider and a single consumer. In this example, www.

Sample Contract using Single Bi-directional Subject, Single Filter with Reverse Filtering Ports Enabled

Single Contract Unidirectional with Multiple Filters

This use case involves implementing a contract without the option to apply the contract subject in both directions. When selecting this option the user no longer has the option to select the reverse filter option.

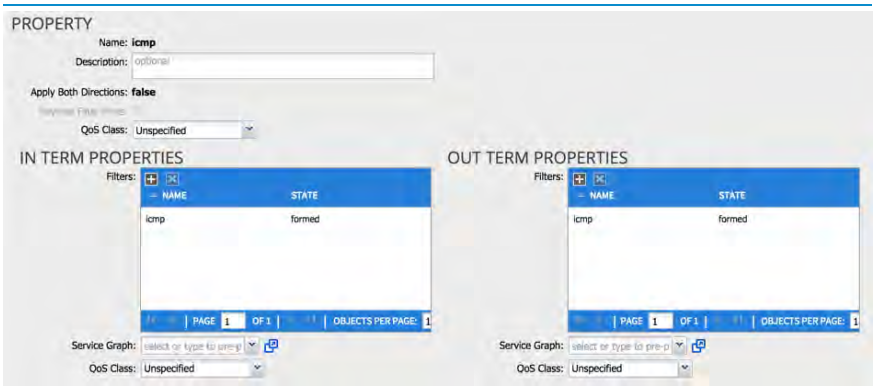
In the use case below, EPG-1 is providing a contract with a subject of icmp and EPG-2 is consuming the contract. This allows the Host in EPG-1 to access the Host in EPG-2 via icmp. When utilizing a single subject without the use of "Apply Both Directions," the user must then configure two filters, one in each direction.



Single Contract, Single Unidirectional Subject, Multiple Filters

Result:

A single contract with (1) Subject (2) Filters and a single provider and a single consumer. In this example, icmp.

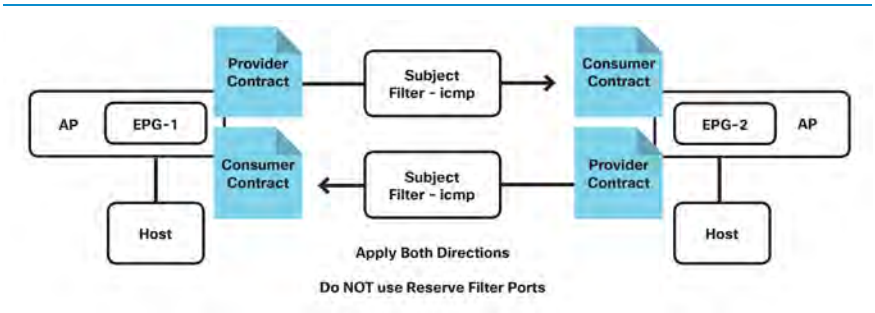


Sample Single Contract, Single Unidirectional Subject, Multiple Filters

Multiple Contracts Uni-Directional Single Filter

This use case is useful when implementing a contract with the option to apply the contract subject in both directions, and without the option to apply the reverse filter. This allows the end-user the most granularity when deploying contracts, but is also the most comprehensive.

In the use case below, EPG-1 is providing a contract with a subject of www and EPG-2 is consuming the contract. This allows the Web Client in EPG-2 to access the Web Server in EPG-1. That is, EPG-1 is providing a service to EPG-2.

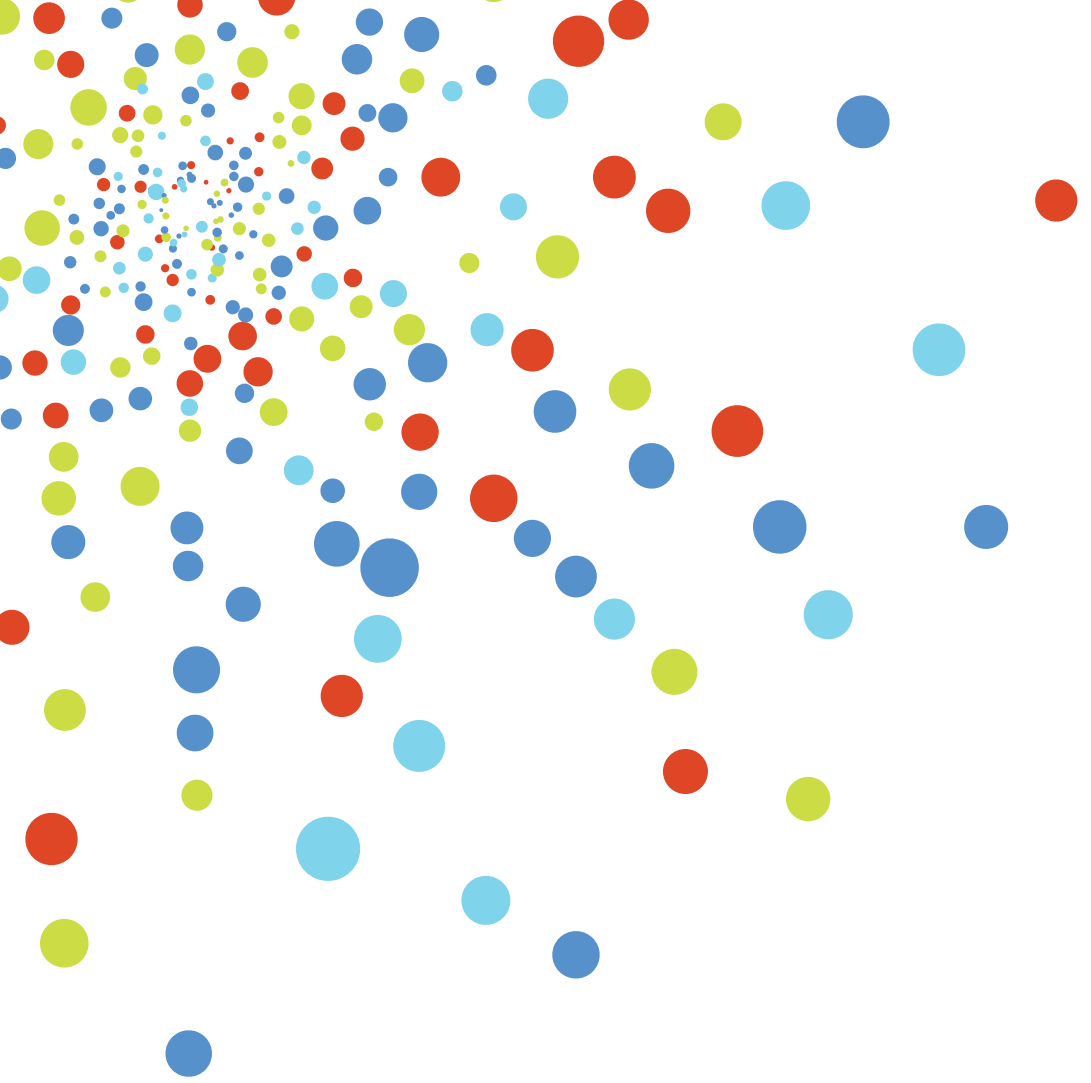


Multiple Contracts, Unidirectional Subjects, Single Filters

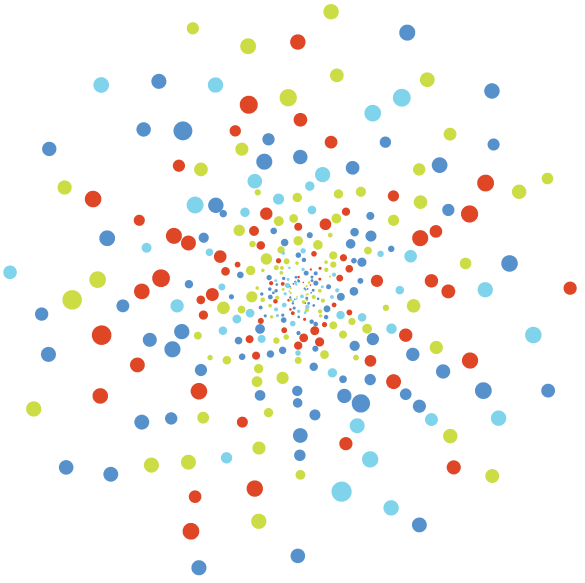
Result:

Two contracts with (1) Subject (1) Filters. Each contract will have a single provider and a single consumer referencing the same contract. The difference here is that the contract is explicitly applied in BOTH directions.

Sample Contract with Single Bi-Directional Subjects and single Filter



Layer 4 to Layer 7 Services



Section Content

- Understanding Layer 4 to Layer 7 Integration
  - Service Insertion Design Principles
  - Applying Service Graphs to EPG Communications
  - Rendering Service Graphs
  - Integration Support Function
- Services Deployment Guide Reference
  - Import Device Package
  - Create a Device
  - Modify a Device
  - Create Layer 4 to Layer 7 Service Graph Template
  - Apply a Service Graph Template to EPGs
- Service Graph Monitoring
  - Monitoring a Service Graph Instance
  - Monitoring and Resolving Service Graph Faults
  - Monitoring a Virtual Device
- ASAv Sample Configuration
  - Verify ASAv VM configuration
  - Remove GW IP from Relevant Fabric Bridge Domains
  - Create a Layer 4 to Layer 7 Device
  - Create a Layer 4 to Layer 7 Service Graph Template
  - Apply the Service Graph Template
  - Associate the Web and App Bridge domains to the Interfaces
  - Verifying service node configuration
    - Display the Access List configuration pushed-down from APIC
    - Display the Interface configuration pushed-down from APIC
    - Display the current Interface names

Understanding Layer 4 to Layer 7 Integration

When ACME Inc. designed their new application, they knew they would need to incorporate some services, such as firewalls, load balancers, IDS/IPS, or other types of higher-layer service device.

In traditional infrastructure, service insertion would require some inline device placement or redirection in order to get traffic to the service devices. As an example, firewalls might be placed directly inline as a "bump in the line" or might be an adjunct service device near a gateway. Firewalls are typically configured per device by building up blocks of static configuration. These static configuration blocks build up over time to create a situation where the configuration works, but can become inflexible and fragile, which can cause change management to become challenging.

One of the key technology innovations in Cisco's Application Centric Infrastructure (ACI) is policy-based management of service insertion through application of a Service Graph. Through the rest of this chapter, we will discuss the high level overview of how the process works and how ACME will utilize Service Graphs for efficient management of Layer 4 to Layer 7 services.

The main purpose of data center fabric equipment is fast and efficient forwarding of traffic from ingress to the fabric, between physical and virtual hosts within the fabric and egress back out of the fabric. Useful infrastructure implementations also utilize this fast fabric in a smart way to also integrate Layer 4 to Layer 7 services. Some possible Layer 4 to Layer 7 services include:

- Firewalls
- Load balancers
- Traffic inspection appliances
- SSL offload functions
- Application flow acceleration functions

Integrating services with Cisco ACI Service Graphs will provide ACME with the following benefits:

- Policy based configuration management

- Flexible configuration state abstraction through the ACI object model
- Integrated configuration management using the APIC GUI, REST API or Python scripts, all based on a consistent ACI object model
- Complex topology modeling with logical flow stitching allowing abstracted links between multiple service devices
- Policy-based provisioning allowing rapid complex topology deployment
- Configuration synchronization allowing dynamic workload provisioning and de-provisioning without manual intervention
- Application centric template-based configuration management and object reuse to shorten infrastructure implementation timelines
- Infrastructure multi-tenancy within the fabric and the service devices

Service Insertion Design Principles

With the spine-leaf architecture and holistic fabric management aspects of ACI, traffic flow to/from service appliances is managed very efficiently, and thus, the appliances do not need to be placed at any specific location within the network fabric. Services appliances can be physical or virtual and can be connected to any leaf under management of the ACI fabric. Where applicable, physical appliances can also be run in multiple context mode, allowing multi-tenant mapping of fabric forwarding and tenant-specific service configurations.

With virtual services appliances, currently only VMware Hypervisors and VLAN transport modes are supported.

Applying Service Graphs to EPG Communications

To allow communications between EPGs within an ACI fabric, a contract must be put in to place. This contract may take the form of a specific consumer/provider relationship defined by specified protocol and port. There could also be an "allow all" contract that allows completely open communications. This contract essentially controls communications flow between EPGs, and can be extended to include service insertion via attachment of a Service Graph to a contract. The Service Graph then ties the contract to the resolved service device with the policy-based configuration in place.

Rendering Service Graphs

The ACI allows users to define a policy using the following ways:

- APIC GUI
- API with a programmatic tool, such as Python or a RESTful API post through POSTman

These policy objects can be created, manipulated, and reused. As it relates to the Layer 4 to Layer 7 services, these objects express the intent of use for that object in relation to the application.

When an application profile is deployed and endpoints are attached to the leaf switches, the service graph objects are then translated into specific device configurations that gets pushed the service nodes through a process called rendering. The APIC also sets up the network forwarding path to make sure the correct forwarding action is taken to get traffic flow to the service nodes for treatment.

This abstracted process of configuration management works like a policy template where you can define the expected behavior, then link two groups and subject their relationship to that policy. This policy can be copied, reused and repackaged as necessary.

The rendering involves allocation of the necessary bridge domains, configuration of IP addresses on the firewall and load balancer interfaces, creation of the VLAN on these devices to create the path for the functions, and performance of all the work necessary to make sure that the path between EPGs is the path defined in the service graph.

As is the case with many customers, ACME has a few cookie cutter templates for firewall and load-balancing services. Though the initial definition of these templates can be potentially cumbersome, subsequently reusing the templates is very straightforward simply by replacing IP addresses, ports, object-groups, and other values.

Integration Support Function

In the ACI model, communications with the service devices is supported by importing device packages. These device packages carry the device description, exposed functions, and have configuration script content. When the device package is imported, the ACI fabric has full understanding of what a device can do, how it connects to the fabric,

how to build path forwarding to bring traffic into and get traffic back from the device, and how to translate policy intent to a device-specific configuration. This device package is a vendor-developed package that is readily available from the original vendor or from Cisco on the software download page.

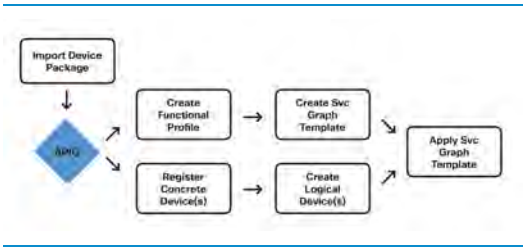
Some of the Device Packages can be downloaded at: http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/solution-overview-c22-732445.html

Device packages from multiple vendors that leverage the rich open API that ACI provides are available from vendors such as Citrix and F5 at the time of writing of this book.

An up to date listing of partners that leverage the API are available at: http://www.cisco.com/c/en/us/solutions/data-center-virtualization/unified-fabric/aci_ecosystem.html

Services Deployment Guide Reference

The diagram below shows a high level overview of the Layer 4 to Layer 7 services workflow when attempting to integrate a device.



L4-L7 Workflow

For information about deploying Layer 4 to Layer 7 services, see the *Cisco APIC Layer 4 to Layer 7 Services Deployment Guide*: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/L4-L7_Services_Deployment/guide/b_L4L7_Deploy.html.

The key sections of the Deployment Guide are listed below:

Import Device Package

To begin the process of Service Node integration, ACME will import the vendor/model-specific device packages, as shown in this section of the Deployment Guide.

Create a Device

Once the device package is imported, the devices will be added through a process of creating a logical device cluster and creating a relationship between this logical device and the physical appliance. This can be done with a physical or VM device. The configuration steps differ slightly for physical devices and virtual devices, but are very similar.

Modify a Device

You can modify a device's configuration through the GUI as described in this section of the Deployment Guide.

Create Layer 4 to Layer 7 Service Graph Template

This section of the Deployment Guide explains how to create a service graph.

Apply a Service Graph Template to EPGs

Once the application Endpoint Groups (EPGs) have been created, the process to apply a service graph template to EPGs can be found in this section of the Deployment Guide.

Service Graph Monitoring

Once ACME Inc. has deployed Service Graphs for application service insertion, the requirement of observing the Service Graph becomes an operational imperative. To support these efforts, there are a few techniques that can be employed, including:

- Monitoring a Service Graph Instance
- Monitoring and Resolving Service Graph Faults
- Monitoring a Virtual Device

Monitoring a Service Graph Instance

Once a service graph is configured and associated with a contract that is attached to an EPG, there are some primary monitoring aspects that should be considered: State of the Service Graph, Functions of a Graph instance, resources allocated to a function and parameters specified for a function.

To monitor a service graph instance:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name* > L4-L7 Services > Deployed Graph Instances**. The Work pane displays information about the deployed graph instances, including a list of the deployed service graphs, the associated contracts, and the current state of the graph policy. A state of "Applied" means the graph has been applied and is active in the fabric and the service device.

For further details of the possible states and other relevant states, see the *Cisco APIC Layer 4 to Layer 7 Services Deployment Guide* at: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/L4-L7_Services_Deployment/guide/b_L4L7_Deploy/b_L4L7_Deploy_chapter_01010.html#task_F2BFF7545D9142EFB208C10F5DFBB1B4

Monitoring and Resolving Service Graph Faults

To monitor a service graph's faults:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name*** **> L4-L7 Services > Deployed Graph Instances**.
4. In the Work pane, choose the **Faults** tab. The Work pane displays the faults that are related to the active service graph.

To understand the faults listed and possible resolutions, see Tables 1 and 2 in the Cisco APIC Layer 4 to Layer 7 Services Deployment Guide at: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/L4-L7_Services_Deployment/guide/b_L4L7_Deploy/b_L4L7_Deploy_chapter_01010.html#concept_307C0CA3EB57469EAF7EF87AAE5A240F

Monitoring a Virtual Device

After you configure a service graph and attach the graph to an endpoint group (EPG) and a contract, you can monitor the virtual devices associated with the service graphs of a tenant. Monitoring the virtual devices tells you what device clusters are associated, which VLANs are configured for a device cluster, the functions in use and the function parameters passed to the devices, the statistics from the devices, and the health of the devices in a device cluster.

To monitor a virtual device:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name*** **> L4-L7 Services > Virtual Devices**. The Work pane displays information about the virtual devices, such as faults and health scores.

ASAv Sample Configuration

One of the service nodes that ACME Inc. will integrate is an ASAv deployed in single-node, routed FW mode. This example details the process they followed to configure the ASAv firewall virtual service appliance as a single node in routed mode. As a routed node, the ASAv will become the default gateway for the hosts in the 2 EPGs that are connected by the contract that this service graph gets associated to.

The high level steps are:

- Create Logical and Concrete device clusters
- Define a firewall ruleset/graph
- Deploy the graph between 2 EPGs

Verify ASAv VM configuration

The first set of steps performed is to verify ASAv VM configuration and upload the ASA device package (or verify if it has already been done)

1. Log in to the ASAv VM.
2. Enter enable mode.
3. Issue the following command:

```
# show ip int brief
```

4. Verify that the ASAv has a correct IPv4 address on the **management 0/0** interface.
5. Verify connectivity from the APIC to the management 0/0 interface of the ASAv.
   a. SSH to the APIC cluster IP address
   b. Issue the following command:

```
# ping {ASAv management 0/0 interface}
```

c. The return response should be similar to the following example:

```
64 bytes from 172.16.10.10: icmp_seq=1 ttl=64 time=0.050 ms
```

If the response is different, then there is likely some sort of connectivity issue. Address the connectivity problems before moving on.

6 In the APIC GUI, on the menu bar, choose **Tenant > *Tenant_Name***.

7 In the Navigation Pane expand ***Tenant_Name* > L4-L7 Services > Packages > L4-L7 Service Device Types**.

8 If an ASA device package is not listed, then perform the following actions:
   a. Right click on the **Device Types**.
   b. Choose **Import Device Package**.
   c. Follow the prompts to upload the device package.

Remove GW IP from Relevant Fabric Bridge Domains

Once the ASAv package and VM are verified, the next step is to remove SVI/GW IP addresses from the fabric Bridge Domains so the Layer 3 routed firewall can become the default gateway.

To remove the routing default gateway function on the EPG1 and EPG2 bridge domains:

1 On the menu bar, choose **Tenants > ALL TENANTS**.

2 In the Work pane, choose the tenant.

3 In the Navigation pane, choose ***Tenant_Name* > Networking > Bridge Domains > BD-EPG1**.

4 In the Work pane, perform the following actions:
   a. For the **L2 Unknown Unicast** radio buttons, click **Flood**.
   b. For the **L3 Unknown Multicast Flooding** radio buttons, click **Flood**.
   c. Uncheck the **Unicast Routing** check box.
   d. Click **Submit**.

Repeat this process for the Bridge Domains of the affected EPGs.

Create a Layer 4 to Layer 7 Device

Perform the following steps to add logical and concrete device clusters to a tenant:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation Pane choose ***Tenant_Name* > L4-L7 Services**.
4. In the Work pane, click **Create a L4-L7 Function Profile**.
5. In the **Create a L4-L7 Function Profile** dialog box, perform the following actions:
   a. Enter a name that is relevant to the tenant and fuction, such as "TXX-ASAv-FP".
   b. In the **Profile Group** drop-down list, choose **Create Function Profile Group**.
   c. In the **Create L4-L7 Services Function Profile Group** dialog box, perform the following actions:
      i. Enter a meaningful name, such as "**TXX-FP-Group**".
      ii. Click **SUBMIT**.
   d. In the **Profile** drop-down list, choose **WebServiceProfileGroup** or **WebPolicyForRoutedMode**.
   e. In the **Feature and Parameters** section, choose the **All Parameters** tab. You will configure IP addressing under **Interface Related Configuration** for both external and internal interfaces (externalIf and interalIf).
   f. Expand **Interface Related Configuration** for **externalIf**.
   g. Expand **Interface Specific Configuration**.
   h. Double-click **IPv4 Address**.
   i. **Enter: ipv4 address in a.b.c.d/m.n.o.p format**
   j. Click **Update**.
   k. Repeat steps 4f - 4j as needed.
6. Choose **L4-7 Services** > **Function Profiles** > **ALL PARAMETERS**.
7. Verify the external and internal interfaces IPv4 addresses.

The following steps will create a Layer 4 to Layer 7 device:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.

3. In the Navigation pane, choose ***Tenant_Name* > L4-L7 Services**.
4. In the Work pane, click **Create a L4-L7 virtual device**.
5. In the **Create L4-L7 Devices** dialog box, perform the following actions:
   a. Enter a meaningful name: **Txx-ASAv-Cluster**
   b. **Device Package: CISCO-ASA-1.1 Model: ASAv**
   c. **Mode: Single Node**
   d. **Function Type: Goto**
   e. **Connectivity VMM Domain: Txx-vCenter**
   f. **APIC to Device: Out of Band**
   g. **Credentials: {uid/pwd}**
   h. **Under Device 1, specify the following values:**
      i. Management IP address: **ASAv IP address**
      ii. Management Port: **https**
      iii. VM: **Tenant ASAv Controller** (in the dropdown box)
      iv. **Virtual Interfaces:** Create two entries; click **+** twice; enter interface values accordingly:
         1. Name: **GigabitEthernet0/0** vNIC: **Network Adapter 2** Direction: **provider**
         2. Name: **GigabitEthernet0/1** vNIC: **Network Adapter 3** Direction: **consumer**

      v. Click **UPDATE** after each entry.
      vi. Click **NEXT**.
      vii. Click **FINISH**.

Verify the Logical and Concrete Device Clusters have been configured:

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation Pane choose ***Tenant_Name* > L4-L7 Services > L4-L7 Devices**.
4. Expand both **TXX-ASAv-Cluster** and **TXX-ASAv-Cluster_Device_1** to view the logical and physical interfaces.
5. Select **TXX-ASAv-Cluster_Device_1** to see a graphic view of the concrete device.

Note This completes the Logical and Concrete device creation.

Create a Layer 4 to Layer 7 Service Graph Template

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name* > L4-L7 Services > L4-L7 Service Graph Templates**.
4. In the Work pane, choose **Actions > Create a L4-L7 Service Graph Template**.
5. In the **Create L4-L7 Devices** dialog box, perform the following actions:
   a. In the **Name** field, enter "TXX-ASAv-L3-Routed-Template".
   b. In the **Type** drop-down list, choose **Single Node - Firewall in Routed Mode**.
   c. In the **Device Function** drop-down list, choose **CISCO-ASA-1.1/Firewall**.
   d. In the **Profile** drop-down list, choose **TXX-ASAv-FP**, which is the functional profile you created previously.
   e. Click **Submit**.
6. You can verify that the template was created successfully by expanding ***Tenant_Name* > L4-L7 Services > L4-L7 Service Graph Templates > Txx-ASAv-L3-Routed-Template > Function Node - Firewall**.

Apply the Service Graph Template

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name* > L4-L7 Services > L4-L7 Service Graph Templates**.
4. Right click **Txx-ASAv-L3-Routed-Template** and choose **Apply L4-L7 Service Graph Template**.
5. In the **Apply L4-L7 Service Graph Template to EPGs** dialog box, perform the following actions:
   a. In the **Consumer EPG / External Network** drop-down list, choose **EPG1**.
   b. In the **Provider EPG / External Network** drop-down list, choose **EPG2**.

c. In the **Service Graph Template** drop-down list, choose **Txx-ASAv-L3-Routed-Template**.
d. For the **Contract** radio buttons, click **Create A New One**.
e. Check the **No Filter (Allow All Traffic)** check box.
f. In the **Contract Name** field, enter "TXX-EPG1-to-EPG2".
g. Click **Next**.
h. In the **L4-L7 Devices** drop-down list, choose **Txx-ASAv-Cluster**.
i. In the **Features and Parameters** section, choose the **All Parameters** tab.
j. Double click **Device Config > Interface Related Configuration externalIf > Access Group > Inbound Access List**.
k. In the **Value** drop-down list, choose **access-list-inbound**.
l. Click **Update**.
m. Expand **Device Config > Interface Related Configuration externalIf > Interface Related Configuration** to verify the IP address assignment. If the mask is missing, the configuration will not push to the ASA.
n. Double click **Device Config > Interface Related Configuration internalIf > Access Group > Inbound Access List**.
o. Click **Reset**. This unassigns the inbound access list from the internal interface. This inbound access list is not desirable for the lab traffic-flow.
p. Expand **Device Config > Interface Related Configuration internalIf > Interface Specific Configuration** to verify the IP address assignment with the mask.
q. Click **Finish**. The **Devices Selection Polices** folder and **Deployed Graph Instances** folder are now populated.

6 Choose **TXX-EPG1-to-EPG2-TXX-ASAv-L3-Router-Template-TXXProduction**. You can see that the Consumer and Provider EPGs are associated with the EPG1 and EPG2 server EPGs.

Associate the Web and App Bridge domains to the Interfaces

1 Expand **Devices Selection Policies**.

2 Expand **TXX-EPG1-to-EPG2-TXX-ASAv-L3-Router-Template-Firewall.**

3 Select **External.**

4 Assign the Bridge Domain to **BD-EPG1.**

5 Repeat the process to assign the Bridge Domain to **BD-EPG2.**

6 Click **SUBMIT.**

7 **Expand Layer 4 to Layer 7 Service Graph Templates.**

8 Expand **TXX-ASAv-L3-Router-Template.**

9 Expand **Function Node – Firewall** > choose **external.**

10 Filters: **common/default.**

11 CTX Terms: **TXX-ASAv-L3Routed Template/T1/Out.**

12 Click **SUBMIT.**

13 Repeat the process for internal.

14 Filters: **common/default.**

15 CTX Terms: **TXX-ASAv-L3Routed Template/T1/In.**

16 Click **SUBMIT.**

The ASAv will now have IP addresses asigned in the Service Graph Profile. This can be verified by going into the ASAv VM console and issue the following command:

```
# show ip int brief
```

Verifying service node configuration

Once the Device has been integrated and the Service Graph has been configured and associated, the resulting configuration pushed to the service node can be verified by simple *show* commands on the real service node device.

Display the Access List configuration pushed-down from APIC

SSH to the ASAv service node for access list validation. Issue the following command:

```
# show run | grep access
```

This will show the access list configuration and you can relate that to the configuration that was done in the APIC.

Display the Interface configuration pushed-down from APIC

SSH to the ASAv service node for interface configuration validation. Issue the following command:

```
# show run interface
```

This will show the interface configuration where the IP address configuration was pushed from the APIC.

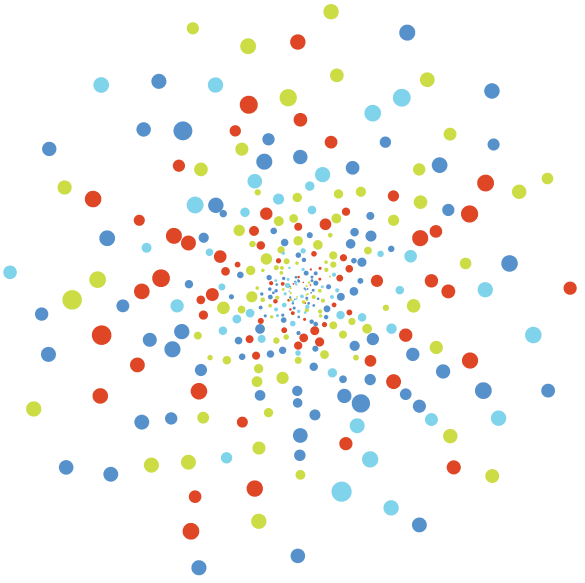
Display the current Interface names

SSH to the ASAv service node for interface name configuration validation. Issue the following command:

```
# show nameif
```

This will show the interface names pushed from the APIC and will show the related interface names to the logical interface names that were configured in the APIC above.

Health Scores



Section Content

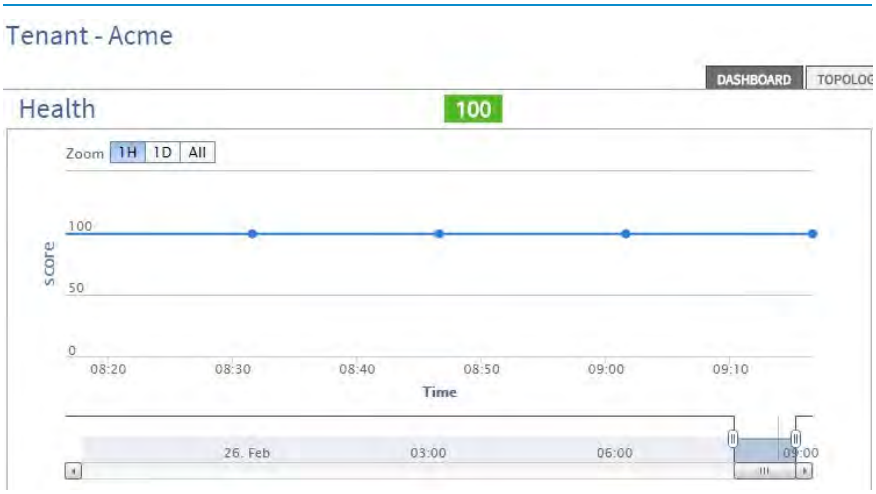
Understanding Health Scores

ACME's Operations team has been challenged on a regular basis to answer basic questions regarding the current status, performance, and availability of the system they are responsible for operating. To address these challenges they can now utilize the Application Centric Infrastructure (ACI), which provides Health Scores that make information on status, performance, and availability readily available. It is worth noting that while providing such answers may be easy as it relates to an independent device or link, this information by itself is of little to no value without additional data on its effect on the overall health of the network. To manually collect and correlate information would have previously been a long and tedious task, but with health scores, data throughout the fabric is collected, computed, and correlated by the APIC in real time and then presented in an understandable format.

Traditional network monitoring and management systems attempt to provide a model of infrastructure that has been provisioned, and describe the relationship between the various devices and links in attempt to provide a correlation. The object model at the heart of ACI is inherent to the infrastructure, and therefore the current status of all of the objects including links, devices, their relationships, as well as the real time status of their utilization, can be represented in a single consolidated health score.

The visibility provided by health scores give the operator a quick at-a-glance assessment of the current status of the entire system, or any subset of the system. This visibility has a number of practical use cases, and in this chapter we will classify these use cases as reactive and proactive. ACI also provides the flexibility to monitor some aspects of how the health scores are calculated, and how various faults impact the calculation of the health score.

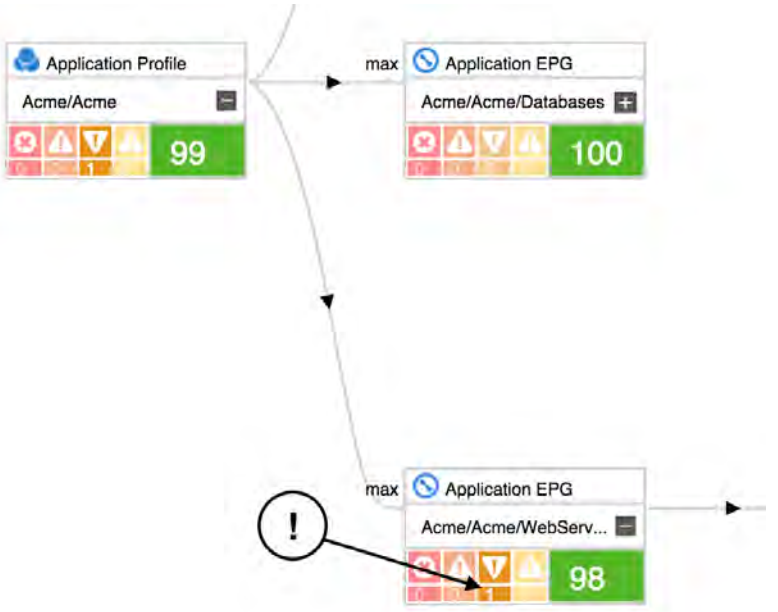
Most objects in the model will have an associated health score, which can be found from the Dashboard or Policy tabs of the object from the GUI. Additionally, all health scores are instantiated from the healthInst class and can be extracted through the API.



Tenant Health Score

In a reactive capacity, ACI health scores provide a quick check as to whether an issue being reported is confirmed in a degradation of the health score. If so, the root cause of the issue can be found by exploring the faults and how these get rolled up in the larger model. Health scores also provide a real-time correlation in the event of a failure scenario, immediately providing feedback as to which tenants, applications, and EPGs are impacted by that failure.

As a example, if you navigate to the application profile it has a **Health** tab. In this tab is a tree that will show the various objects in a tree form to reveal faults.



Object with a fault

Proactively, ACI health scores can help identify potential bottlenecks in terms of hardware resources, bandwidth utilization, and other capacity planning exercises. Operations teams also stand a better chance of identifying issues before they impact customers or users.

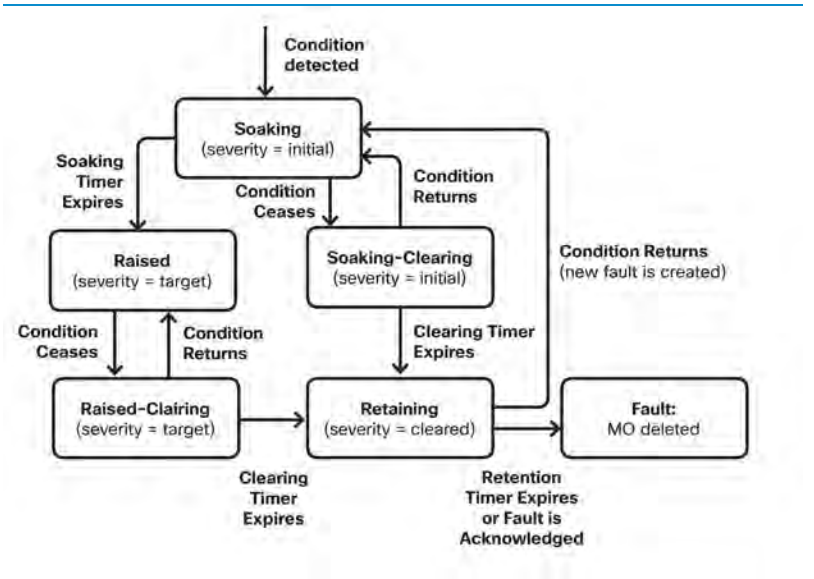
Ideally, the health of all application and infrastructure components would always be at 100%, however, this is not always realistic given the dynamic nature of data center environments. Links, equipment, and endpoints have failures. Instead the health score should be seen as a metric that will change over time, with the goal of increasing the average health score of a given set of components over time.

Understanding Faults

From a management point of view we look at the Application Policy Infrastructure Controller (APIC) from two perspectives:

1. Policy Controller - Where all fabric configuration is created, managed and applied. It maintains a comprehensive, up-to-date run-time representation of the administrative or configured state.
2. Telemetry device - All devices (Fabric Switches, Virtual Switches, integrated Layer 4 to Layer 7 devices) in an ACI fabric report faults, events and statistics to the APIC.

Faults, events, and statistics in the ACI fabric are represented as a collection of Managed Objects (MOs) within the overall ACI Object Model/Management Information Tree (MIT). All objects within ACI can be queried, including faults. In this model, a fault is represented as a mutable, stateful, and persistent MO.



Fault Lifecycle

When a specific condition occurs, such as a component failure or an alarm, the system creates a fault MO as a child object to the MO that is primarily associated with the fault. For a fault object class, the fault conditions are defined by the fault rules of the parent object class. Fault MOs appear as regular MOs in MIT; they have a parent, a DN, RN, and so on. The Fault "code" is an alphanumerical string in the form **FXXX**. For more information about fault codes, see the ***Cisco APIC Faults, Events, and System Messages Management Guide***.

The following example is a REST query to the fabric that returns the health score for a tenant named "3tierapp":

```
https://hostname/api/node/mo/uni/tn-3tierapp.xml?query-target=self&rsp-subtree-include=health
```

The following example is a REST query to the fabric that returns the statistics for a tenant named "3tierapp":

```
https://hostname/api/node/mo/uni/tn-3tierapp.xml?query-target=self&rsp-subtree-include=stats
```

The following example is a REST query to the fabric that returns the faults for a leaf node:

```
https://hostname/api/node/mo/topology/pod-1/node-103.xml?query-target=self&rsp-subtree-include=faults
```

As you can see, MOs can be queried by class and DN, with property filters, pagination, and so on.

In most cases, a fault MO is automatically created, escalated, de-escalated, and deleted by the system as specific conditions are detected. There can be at most one fault with a given code under an MO. If the same condition is detected multiple times while the corresponding fault MO is active, no additional instances of the fault MO are created. In other words, if the **same condition** is detected multiple times for the **same affected object**, only **one fault** is raised while a counter for the recurrence of that fault will be incremented. A fault MO remains in the system **until the fault condition is cleared**. To remove a fault, the condition raising the fault must be cleared, whether by configura-

tion, or a change in the run time state of the fabric. An exception to this is if the fault is in the cleared or retained state, in which case the fault can be deleted by the user by acknowledging it.

Severity provides an indication of the estimated impact of the condition on the capability of the system or component to provide service.
Possible values are:

- Warning (possibly no impact)
- Minor
- Major
- Critical (system or component completely unusable)

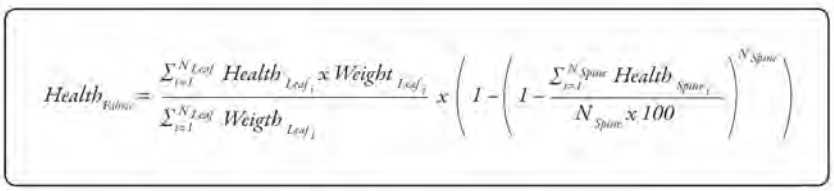
The creation of a fault MO can be triggered by internal processes such as:

- Finite state machine (FSM) transitions or detected component failures
- Conditions specified by various fault policies, some of which are user configurable

For example, you can set fault thresholds on statistical measurements such as health scores, data traffic, or temperatures.

How Health Scores Are Calculated

Health scores exist for systems and pods, tenants, managed objects (such as switches and ports), as well as an overall health score for the overall system. All health scores are calculated using the number and importance of faults that apply to it. System and pod health scores are a weighted average of the leaf health scores, divided by the total number of learned end points, multiplied by the spine coefficient which is derived from the number of spines and their health scores. In other words:



The following legend defines the equation components.

$Health_{Leaf_i}$ is the health score of the leaf switch

$Weight_{Leaf_i}$ is the number of end-points on the leaf switch

N_{Leaf} is the number of leaf switches in the fabric

$Health_{Spine_i}$ is the health score of the spine switch

N_{Spine} is the number of spine switches in the fabric

Health Score calculation

Tenant health scores are similar, but contain health scores of logical components within that tenant. For example, it will only be weighted by the end points that are included in that tenant.

You can see how all of these scores are aggregated by looking at how managed object scores are calculated, which is directly by the faults they have associated with them. Each fault is weighted depending on the level of importance. Critical faults might have a

high fault level at 100%, while warnings might have a low fault level at only 20%. Faults that have been identified as not impacting might even be reassigned a percentage value of 0% so that it does not affect the health score computation.

Luckily there is really no need to understand the calculations of the health scores to use them effectively, but there should be a basic understanding of whether faults should have high, medium, low, or "none" fault levels. Though faults in ACI come with default values, it is possible to change these values to better match your environment.

Keep in mind, because of the role-based access control, not all administrators will be able to see all of the health scores. For example, a fabric admin will be able to see all health scores, but a tenant admin would only be able to see the health scores that pertain to the tenants to which they have access. In most cases, the tenant admin should be able to drill into the health scores that are visible to them, but it is possible a fault may be occurring that is affecting more than that one tenant. In this case the fabric administrator may have to start troubleshooting. The tenant and fabric admins may also see health scores of any layer four through seven devices, such as firewalls, load balancers, and intrusion prevention/detection systems. These, along with faults within our VMM domains will all roll up into our tenant, pod, and overall system health scores.

For more information on how to use faults, see the *Troubleshooting Cisco Application Centric Infrastructure* book: http://datacenter.github.io/aci-troubleshooting-book/.

Health Score Use Cases

Using Health Scores for Proactive Monitoring

While ACME administrators have traditionally spent a lot of time reacting to issues on the network, ACI health scores will allow them to start preventing issues. Health scores not only act as indicators of faults, they are essentially baselines to which you can make comparisons later. If you see that one of the leaf switches is at 100% (green for good) one week, and the next week the leaf is showing a warning, you can drill down to see what changed. In this scenario, it is possible the links are oversubscribed and so it can be time to either move some of of the workload to another leaf or maybe to add more bandwidth by connecting more cables. Since it is still only a warning, there is time to resolve the issue before any bottlenecks on the network are noticeable.

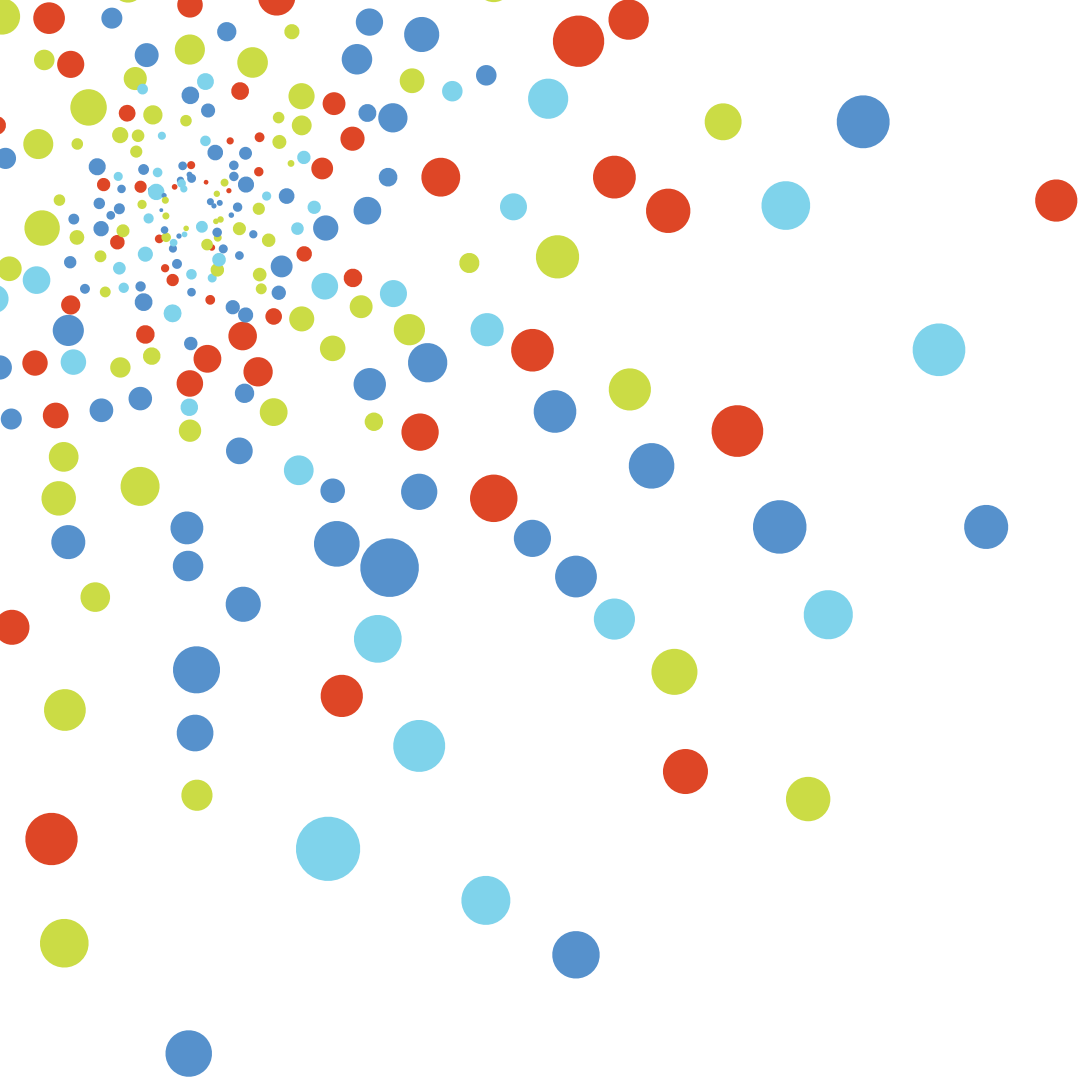
The same scenario can observed with a load balancer or firewall that is getting overloaded. In these cases adding another load balancer, or firewall, or maybe even optimizing the rules may be needed to make traffic flow more efficient. As shown in the above examples, this baselining method can be used as a capacity planning tool.

Other ways health scores can be used to proactively monitor your ACI environment are by giving visibility of certain components to other groups. Since you can export the scores and faults, it is possible to send these notifications to application owners, VMware administrators, Database Administrator, and so on. This would provide monitoring of the environment across the network that has not previously been available and which is not able to be retrieved by any other means.

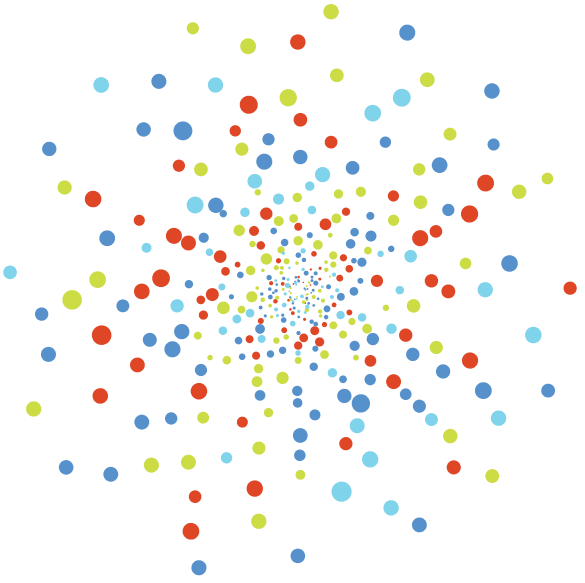
Using Health Scores for Reactive Monitoring

Reactively, health scores can be used to diagnose problems with the ACI fabric. Upon notification that a health score has been degraded, an operator can use the GUI to easily navigate the relationships and faults that are contributing to that health score. Once the root cause faults have been identified, the fault itself will contain information about possible remediation steps.

Most objects will have a Health tab which can be used to explore the relationship between objects, and their associated faults. This provides the ability to “double-click to root cause”.



Monitoring



Section Content

- Proactive Monitoring - Tenant and Fabric Policies
  - Stats Collection Policies
  - Stats Export Policies
  - Diagnostics Policies
  - Call Home/SNMP/syslog
  - Event Severity and Fault Severity Assignments
  - Fault Lifecycle Policies
  - TCAM Policy Usage
    - Create TCAM Policy Monitor
    - TCAM Prefix Usage
  - Health Score Evaluation Policy
  - Communication Policy

- Proactive Monitoring - Infrastructure
  - Monitoring APICs
    - CPU utilization and Memory
    - Disk Utilization
    - Physical and Bond Interface Statistics
    - APIC Fan Status
    - Temperature Status
    - Power Supply Status
  - Monitoring Leaf Switches
    - Monitoring Switch CPU Utilization
    - Monitoring Switch Memory Utilization
    - Monitoring File System Health
    - Monitoring CoPP (Control Plane Policing) Statistics
    - Physical Interface Statistics and Link State
    - Module Status
    - Switch Fan Status

        - Power Supply Status
        - LLDP Neighbor Status
        - GOLD Diagnostic Results

- Proactive Monitoring Use Cases
    - Monitoring Workload Bandwidth
    - EPG Level Statistics

- Reactive Monitoring
    - Reactive Monitoring Tools
        - Switch Port Analyzer (SPAN)
        - Traceroute
        - Atomic Counters
        - Traffic Map
        - Enhanced Troubleshooting Wizard
        - IPing
        - Audit Logs

- Reactive Monitoring Use Cases
    - Loss of Connectivity to Endpoint
    - Users Report that an Application Running in the Fabric is Slow

Proactive Monitoring - Tenant and Fabric Policies

Proactive monitoring is a very important piece of the network administrator's job, but is often neglected because putting out fires in the network usually takes priority. However, since the APIC makes it incredibly easy to gather statistics and perform analyses, this will save network administrators both time and frustration. Since statistics are gathered automatically and policies are used and can be re-used in other places, the human error and effort is minimal.

Statistics gathering has been a somewhat manual and even resource intensive process for ACME in the past. Even when they have used tools to gather data on layer one through seven devices, it has still been necessary to manually specify which devices are to be monitored and how they should be monitored. For example, SNMP and a third party tool may have been used to monitor the CPU of switches or bandwidth utilization on ports, but they struggled with entering correct SNMP information on each device, or often forgot to add a new device to their Network Monitoring System (NMS). ACI provides an APIC which will do all of the statistics gathering, and provides the ability to proactively monitor your entire environment without all of the hassle of maintaining a third party monitoring tool.

The APIC, whether accessed through the GUI, CLI, or API, can be used to drill into any of the components and provides the ability to click on a Stats tab to display on-demand statistics, but more importantly it enables the setup of policies to keep persistent data to analyze trends in the environment, as well as to troubleshoot or predict any issues that may be arising. When planning to move an application from a legacy network to the ACI infrastructure, it is sensible to start by testing before going straight to production. Add test VMs to port groups on either a DVS or AVS associated with the APIC, and add physical test servers to VPCs on the leaf switches. This could also be in a testing tenant which is completely separate from the production environment. At this point the APIC is already gathering statistics for the VMM domain and the physical devices. The next step is to configure a policy for trend analysis.

There are four different scopes for statistics gathering: Common or Fabric Wide, Fabric, Tenant, or Access. A Fabric Wide policy would be created as a default policy to be applied to all tenants. However, to override that policy for a particular tenant, the ten-

ant policy will override the Fabric policy. In the following testing example, a Tenant policy is created to gather statistics. Even if this tenant is shared with other applications, customers, test cases, it will provide a real world example of how the application will behave in a production environment.

Create Tenant Monitor policy

To create a tenant monitoring policy:

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane, choose ***Tenant_Name*** **> Monitoring Policies**.
4. In the Work pane, choose **Actions > Create Monitoring Policies**.
5. In the **Create Monitoring Policies** dialog box, perform the following actions:
   a. In the **Name** field enter a name for the **Monitoring Policy**.
   b. Click **Submit**.
6. In the Navigation pane, choose ***Tenant_Name*** **> Monitoring Policies >** ***Policy_Name*** to display the following information:
   - Stats Collection Policies
   - Stats Export Policies
   - Diagnostics Policies
   - Callhome, SNMP, and syslog
   - Event Severity Assignment Policies
   - Fault Lifecycle Policies

Stats Collection Policies

Clicking on Stats Collection Policies will display the default retention periods and admin states (Enabled/Disabled) for ALL Monitored Objects. Most likely the defaults will be kept, but a double click on them will change the admin state or retention periods. For example, to have it poll a component every 5 minutes, but be retained for 2 hours, just click on the policy that specifies a 5 minute granularity and change the retention period to 2 hours. It is similarly possible to change the policies for specific Monitoring Objects. A monitoring object tells the APIC which components to gather

statistics about. For example, to change the information gathered for Bridge Domains, use the Bridge Domain (infra.RSOInfraBD) Monitoring Object.

To add monitoring objects:

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Monitoring Policies** > **Monitoring Policy_Name** > **Stats Collection Policies**
   a. Click on the Pencil icon to edit the **Monitoring Objects**.
   b. Put a checkmark next to the **Monitoring Objects** to be included, and remove any checkmarks next to **Monitoring Objects** to be left out.
   c. Click **Submit**.

For this example, changes might be made to Monitoring Object policies for Tenant, VXLAN Pool, Leaf Port, and/or Taboo Contract. There are several options and this will all depend on what is important to monitor in the environment. Click on the pull down menu to select a monitoring object and add a retention policy to it.

To add a policy to a Monitoring Object:

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose **Monitoring Policies** > **Monitoring Policy_Name** > **Stats Collection Policies.**
4. In the Work pane, in the **Stats Collection Policy** dialog box, perform the following actions:
   a. Select the **Monitoring Object.**
   b. Click **+** to add the policy.
   c. Select the granularity with which it is to poll.
   d. Leave the state as inherited to stick with the defaults as set for **ALL**, or explicitly select **enabled** or **disabled**.
   e. The retention policy may either be inherited or explicitly specified as enabled or disabled as well.
   f. Click **Update.**

Stats Export Policies

It is desirable to collect these ongoing statistics as well as to see how this data behaves over time. Use the Stats Export Policies option in the left navigation pane, located under the monitoring policy. Much like the Stats Collection Policies, it is possible to create a policy for ALL monitoring objects, or select specific monitoring objects and specify where this information will be saved.

To create a Stats Export Policy:

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane choose ***Tenant_Name*** **> Monitoring Policies** > **Monitoring Policy_Name** > **Stats Export Policies.**
4. In the Work pane, in the **Stats Export Policy** dialog box, perform the following actions:
   a. Select **ALL** or a specific monitoring object from the drop-down list.
   b. Click **+** to add the policy.
   c. Now define the **Stats Export Policy** in the wizard.
   d. Choose either **JSON** or **XML** as the format. There's really no difference other than personal preference, or it may be dictated by the tool used to read it.
   e. Choose to compress it using GZIP, or leave it uncompressed.
   f. Click **+** under **Export Destinations** to specify a server where this information is to be collected. Another wizard will pop up to enable specification of the protocols and credentials used to connect to this server.
   g. Click **Ok.**
5. Click **Submit.**

Diagnostics Policies

Next are the diagnostics policies in the navigation pane on the left. This is a really slick feature that allows the setup of diagnostics test for the Monitoring Objects that were specified in the Stats Collection Policies. Next to the Monitoring Object is the Pencil button which enables selection of the monitoring objects to be configured with diagnostics policies. There are two different kind of policies for configuration - Boot-Up diagnostics or Ongoing diagnostics.

To configure diagnostic policies:

1. On the menu bar, choose **Fabric** > **Fabric Policies.**
2. In the Navigation pane choose ***Tenant_Name*** **> Monitoring Policies > default > Diagnostics Policies.**
3. In the Work pane, in the **Diagnostic Policies** dialog box, perform the following actions:

   Note: Click on the **Pencil Icon** and put checks next to the **Monitoring Objects** which diagnostics tests are to be added to.

   a. Select one of the **Monitoring Objects**.
   b. Click **+** to add an Object.
      i. select either Boot-Up or Ongoing.
      ii. **Boot-Up** runs the tests while the devices are booting, and **Ongoing** will run the tests as often as specified within the wizard.
      iii. In the wizard give it a name and select the admin state.
      iv. There are five different diagnostics tests available: ASIC, CPU, Internal Connectivity, Peripherals, and System Memory. Double-click on each to obtain the option of specifying no tests, full tests, or recommended tests.
      v. Click **Submit.**

The diagnostics found here can be useful in finding failed components before they cause major issues within your environment.

Call Home/SNMP/syslog

There are a few different ways to setup notification or alert policies. The Call Home/SNMP/syslog policy will allow alerting to be configured in a flexible manner. Cisco Call Home is a feature in many Cisco products that will provide email or web-based notification alerts in several different formats for critical events. This allows administrators to resolve issues before they turn into outages. SNMP or syslog policies can also be used with current notification systems. Different logging levels may be selected for notifications and alert levels specified for Monitoring Objects from which alerts are to be received.

Event Severity and Fault Severity Assignments

Event and fault severities can be changed for events raised by Monitoring Objects. Most likely, the default severity assignments for Events and Faults will be kept, but there are examples where an ACI administrator may decide the event or fault is more or less severe than the default value. For example, if only critical faults are being notified, but there is a major fault you'd also like to be notified about immediately, you can change the severity for that particular fault code.

1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane, choose ***Tenant_Name*** **> Monitoring Policies > *Monitoring_Policy*** **> Fault Lifecycle Policies.**
4. In the Work pane, in the **Fault Severity Assignment Policies** dialog box, perform the following actions:
   a. Select a **Monitoring Object**, which will dictate the fault codes for which you are changing the fault severity.
   b. Click **+** to add an Object.
   c. Select the particular fault code for which severity is to be changed.
   d. Select the severity: **Cleared, Critical, Major, Minor, Squelched, Inherit, Warning, Info**.
   Note: Squelched gives it a weight of 0%, meaning it does not affect health scores.
5. Click **Update.**

The Event Severity Assignment Policies are configured in the same way.

Fault Lifecycle Policies

Fault Lifecycle is the term Cisco uses to describe the life of a fault. Once a fault is detected it is in the "soaking" state. After a certain amount of time, referred to as the "soaking interval" it will move on to the "raised" state. "Raised" means the fault is still present after the soaking interval. After the fault clears it's in a state called "raised clearing." It is only in this state briefly and moves on to the "clearing time" state. It remains in the "clearing time" state for the amount of time specified in the "clearing interval." Lastly it moves on to the "retaining" state and does not get removed until the end of the "retaining interval."

To change Fault Lifecycle Intervals:

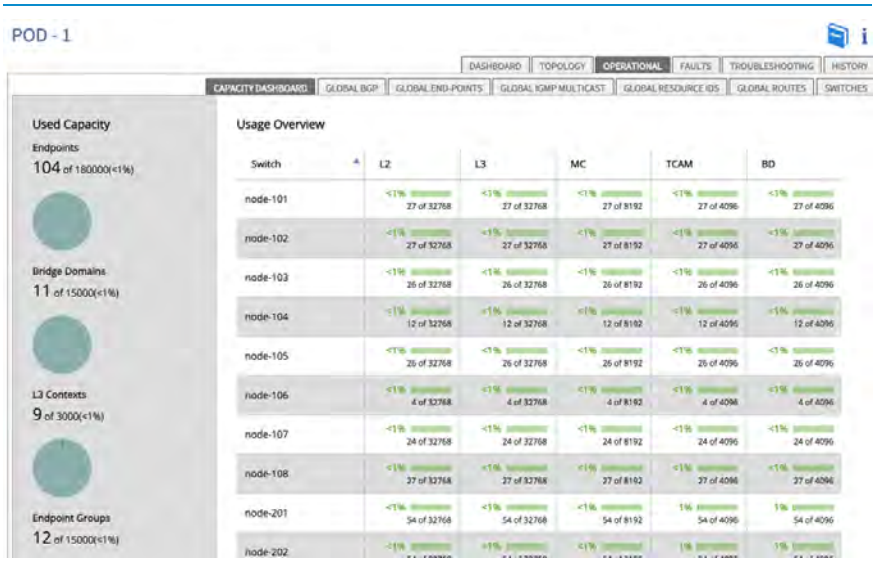
1. On the menu bar, choose **Tenants** > **ALL TENANTS.**
2. In the Work pane, choose the ***Tenant_Name***.
3. In the Navigation pane, choose ***Tenant_Name*** **> Monitoring Policies >** ***Monitoring_Policy*** **> Fault Lifecycle Policies.**
4. In the Work pane, in the **Fault Lifecycle Policies** dialog box, perform the following actions:
   a. Select a **Monitoring Object**, which will dictate the fault codes for which you are changing the default intervals.
   b. Click **+**.
   c. Specify times for the **Clearing Interval**, **Retention Interval**, and **Soaking Interval** (all in seconds).
   Note: The default for the **Clearing Interval** is 120 seconds; the **Retention Interval** is 3600 seconds; and the **Soaking Interval** is 120 seconds.

At this point there will be a fully working tenant monitoring policy. ACME will have other policies to configure in the fabric as outlined in the following sections.

TCAM Policy Usage

The physical ternary content-addressable memory (TCAM) in which policy is stored for enforcement is an expensive component of switch hardware and therefore tends to lower policy scale or raise hardware costs. Within the Cisco ACI fabric, policy is applied based on the EPG rather than the endpoint itself. This policy size can be expressed as **n*m*f**, where **n** is the number of sources, **m** is the number of destinations, and **f** is the number of policy filters. Within the Cisco ACI fabric, sources and destinations become one entry for a given EPG, which reduces the number of total entries required.

TCAM is a fabric resource that should be monitored. There is a a system wide view of available TCAM resources. To see this click on **Fabric** > **Inventory** > **Pod1** and then select the **Operational** tab in the work pane, and you will see a table summarizing capacitiy for all nodes.



Switch TCAM capacity dashboard

TCAM is a critical system resource in an ACI fabric and should be monitored for utilization. The architecture/design team should articulate what the assumptions were for TCAM utilization. There is a Fabric Resource Calculation tool on Github that will help with estimation of normal operating parameters:

https://github.com/datacenter/FabricResourceCalculation

As a general rule, the default monitoring policies will alert you to a resource shortage and lower overall fabric health score. If your environment has a high rate of change, or you anticipate the possibility of consistently being oversubscribed, you may want to set different thresholds.

Create TCAM Policy Monitor

1. On the menu bar, choose **Fabric > Fabric Policies.**
2. In the Navigation pane, choose **Monitor Policies > default > Stats Collection Policies**.

3 In the Work pane, in the **Stats Collection Policies** dialog box, perform the following actions:
   a. Select the Monitoring Object **Equipment Capacity Entity (eqptcapacity.Entity).**
   b. Select the **Stats Type Policy Entry.**
   c. Click **+** under Config Thresholds.
   d. In the **Thresholds For Collection 5 Minute** window, select the blue pencil icon next to policy CAM entries usage current value.

TCAM Prefix Usage

This procedure manages a TCAM Prefix Usage.

1 On the menu bar, choose **Fabric > Fabric Policies.**

2 In the Navigation pane, choose **Monitor Policies > default > Stats Collection Policies**.

3 In the Work pane, in the **Stats Collection Policies** dialog box, perform the following actions:
   a. Select the Monitoring Object **Equipment Capacity Entity (eqptcapacity.Entity).**
   b. Select the Stats **TypeLayer3 Entry**.
   c. Click **+** under Config Thresholds.
   d. In the **Thresholds For Collection 5 Minute** window, select the blue pencil icon next to policy CAM entries usage current value.

Health Score Evaluation Policy

1 On the menu bar, choose **Fabric > Fabric Policies.**

2 In the Navigation pane, choose **Monitor Policies > Common Policies > Health Score Evaluation Policy > Health Score Evaluation Policy.**

3 In the Work pane, in the **Properties** dialog box, perform the following actions:
   a. In the **Penalty of fault severity** critical dropdown menu, select the desired %.
   b. In the **Penalty of fault severity** major dropdown menu, select the desired %.
   c. In the **Penalty of fault severity** minor dropdown menu, select the desired %.
   d. In the **Penalty of fault severity** warning dropdown menu, select the desired %.

4 Click **Submit**.

Communication Policy

1 On the menu bar, choose **Fabric > Fabric Policies.**

2 In the Navigation pane, expand **Pod Policies > Policies** > **Communication**.

3 In the Work pane, choose **Actions > Create Communication Policy**.

4 In the **Create Communication Policy** dialog box, perform the following actions:

a. Enter **Communication Policy Name**.

b. From the **HTTP Admin State** dropdown menu select the desired state.

c. From the **HTTP Port dropdown** menu select the desired port.

d. Select the desired HTTP redirect state.

e. From the **HTTPS Admin State** dropdown menu select the desired state.

f. From the **HTTPS Port** dropdown menu select the desired port.

g. Select the desired HTTPS redirect state.

h. From the **SSH Admin State** dropdown menu select the desired state.

i. From the **Telnet Admin State** dropdown menu select the desired state.

j. From the **Telnet Port** dropdown menu select the desired port.

5 Click **Submit.**

Proactive Monitoring - Infrastructure

While health scores provide a comprehensive view of the health status of various objects, and fabric and tenant policies show us a narrower view of fabric and tenant health, ACME will also need to set up policies to monitor specific resources.

This section of the book attempts to cover some key performance indicators that should be monitored, procedures for monitoring them, and when possible, suggest recommended thresholds for triggering alerts or alarms from Network Monitoring Systems (NMS).

There are several methods to obtain this data, and references are provided to ways of obtaining the data when possible. Operators can use a wide array of tools including Syslog, SNMP, GUI, REST API calls and CLI.

A full list of MIBs supported on the 1.x is available at:

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html

It is important to note that this list changes as newer software versions are made available, and more variables may be exposed via SNMP MIBs in the future.

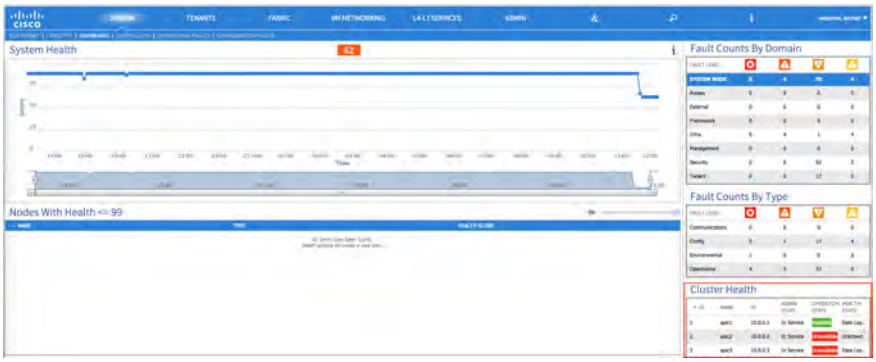
Monitoring APICs

CPU utilization and Memory

GUI

The easiest way to quickly verify the health of the controllers is the APIC. When logging into the system dashboard, the health of the APICs and the health of the cluster itself are displayed right at the dashboard.

The screenshot shows where this information is visible. In this example, two out of the three APICs are in a sub-optimal state and APIC 1 is also experiencing issues.



System health dashboard

The normal state for these is to have them all green in a "fully fit" state implying the APICs are synchronized with each other.

A more detailed drilldown is available by clicking on **System** > **Controllers**.

REST API

Controllers provide information regarding the current status of CPU and memory utilization by creating instances of the *procEntity* class. procEntity is a container of processes in the system. This object holds detailed information about various processes running on the APIC. The procEntity objects contain the following useful properties:

cpuPct - CPU utilization

maxMemAlloc - The maximum memory allocated for the system

memFree - The maximum amount of available memory for the system

Sample Usage: This information can be retrieved for all APIC controllers using the following REST call:

```
http[s]://apic_ip/api/node/class/procEntity.xml?
```

CLI

The Linux Top utility also comes built into the APIC controllers and can be used for troubleshooting and/or verification.

```
user@apic1:~> top
top - 11:41:51 up 16:50,  4 users,  load average: 4.19, 4.27, 4.29
Tasks: 354 total,   1 running, 353 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3%us,  0.4%sy,  0.0%ni, 99.3%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  131954932k total,  7473180k used, 124481752k free,   409540k buffers
Swap:        0k total,        0k used,        0k free,  1952656k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
32102 root      20   0  556m 205m  85m S  3.3  0.2  38:11.04 svc_ifc_applian
32120 ifc       20   0  660m 343m  86m S  2.0  0.3  27:58.73 svc_ifc_dbgr.bi
32121 ifc       20   0  631m 286m  86m S  2.0  0.2  17:41.92 svc_ifc_topomgr
32105 root      20   0  659m 258m  85m S  1.7  0.2  17:08.35 svc_ifc_bootmgr
32113 ifc       20   0 1083m 721m  69m S  1.7  0.6  20:03.37 svc_ifc_observe
32128 ifc       20   0  639m 315m  69m S  1.7  0.2  16:28.34 svc_ifc_reader.
32132 ifc       20   0  657m 252m  71m S  1.7  0.2  17:13.74 svc_ifc_scripth
 1291 root      20   0  834m 419m  94m S  1.3  0.3  20:35.24 nginx.bin
```

Disk Utilization

GUI

There are several disks and file systems present on the APICs. The GUI provides ready access to disk space utilization of all partitions on the system and can be used for monitoring this information.

The disk utilization can be viewed by clicking on **System** > **Controllers** > **Apic-X** > **Storage**

The work pane displays the utilization of all partitions in the system.

CLI

```
user@apic1:~> df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/dm-1             41282880  10518960  28666872  27% /
tmpfs                  4194304     56456   4137848   2% /dev/shm
tmpfs                 65977464       964  65976500   1% /tmp
/dev/mapper/vg_ifc0-data
                      41282880  10518960  28666872  27% /data
/dev/mapper/vg_ifc0-firmware
                      41284928  13860672  25327104  36% /firmware
/dev/mapper/vg_ifc0-data2
                     583149656   1281104 552246280   1% /data2
```

* Note that not all file systems are visible from the CLI as some require root access to reach the mount points. The GUI should be used as a single source of truth for file system utilization.

REST API

This information can be retrieved for all APIC controllers using the following REST call:

```
http[s]://apic-ip/api/node/class/eqptStorage.xml?
```

Physical and Bond Interface Statistics

APICs use a bonded interface that is typically dual-homed to two leaves for connectivity to the ACI fabric and have the ability to use a bonded interface that can be dual homed to the out-of-band management network.

Bond0 is the bond interface used to connect to the fabric itself (to connect to leaves that connect into the fabric).

Bond1 is the bond interface used to connect to the out-of-band segment (to connect to an OOB segment that allows setup of the APIC itself).

The bond interfaces rely on underlying physical interfaces and it is important to note that the GUI provides link information for both the physical and logical bond interfaces.

GUI

To view interface status for the interfaces on the APICs, navigate to **System** > **Controllers** > **Apic-x** > **Interfaces**

CLI

Both "ifconfig" and the "ip link" CLI commands can be used to verify link state. The CLI also provides information on detailed interface statistics such as RX and TX counters.

REST API

This information can be retrieved for all APIC controllers using the following REST call:

```
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-1/sys.json?query-
target=subtree&target-subtree-class=l3EncRtdIf
```

APIC Fan Status

The following section describes methodologies to retrieve the status of the fan trays on the APICs.

GUI

To view interface status for the interfaces on the APICs, navigate to **System** > **Controllers** > **Apic-x** > **Equipment-Fans**

REST API

This information can be retrieved for all APIC controllers using the following REST call:

```
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-1.json?query-
target=subtree&target-subtree-class=eqptFan
```

CLI

The Fan status for the APICs can be monitored using the CLI on the CIMC port of the APIC. To obtain this, login to the CIMC using the credentials used for setting up the CIMC (may not be the same as the credentials used for APIC). If this has not been setup previously, the default username is admin and the default password is password.

The CIMC port is the integrated lights-out management port that can be used to recover an APIC in the event of a catastrophic failure.

```
user@apic1:~> ssh -l admin 172.16.176.179
Warning: Permanently added '172.16.176.179' (RSA) to the list of known hosts.
admin@172.16.176.179's password:
C220-FCH1807V02V# scope sensor
C220-FCH1807V02V /sensor # show fan
Name      Sensor     Reading  Units  Min.      Max.     Min.     Max.
          Status                     Warning   Warning  Failure  Failure
----------- ------- --------  ----- --------- --------- -------- ---------
FAN1_TACH1   Normal     7490   RPM    1712      N/A       1284    N/A
FAN1_TACH2   Normal     7490   RPM    1712      N/A       1284    N/A
FAN2_TACH1   Normal     7490   RPM    1712      N/A       1284    N/A
FAN2_TACH2   Normal     7276   RPM    1712      N/A       1284    N/A
FAN3_TACH1   Normal     7704   RPM    1712      N/A       1284    N/A
FAN3_TACH2   Normal     7276   RPM    1712      N/A       1284    N/A
FAN4_TACH1   Normal     7704   RPM    1712      N/A       1284    N/A
FAN4_TACH2   Normal     7276   RPM    1712      N/A       1284    N/A
FAN5_TACH1   Normal     7704   RPM    1712      N/A       1284    N/A
```

Temperature Status

To monitor the temperature state of the various sensors available on the APICs use the following steps.

GUI

To view interface status for the interfaces on the APICs, navigate to **System** > **Controllers** > **Apic-x** > **Equipment-Sensors**

REST API

This information can be retrieved for all APIC controllers using the following REST call:

```
 https://{{apic-ip}}//api/node/mo/topology/pod-1/node-1.json?query-
target=subtree&target-subtree-class=eqptSensor
```

CLI

```
 C220-FCH1807V02V /sensor # show temperature
C220-FCH1807V02V /sensor # show temperature
  Name                Sensor    Reading    Units  Min.      Max.      Min.      Max.
                      Status                      Warning   Warning   Failure   Failure
  ------------------- --------- ---------  ------- --------- --------- --------- ---------
  P1_TEMP_SENS        Normal    49.5       C      N/A       81.0      N/A       86.0
  P2_TEMP_SENS        Normal    50.5       C      N/A       81.0      N/A       86.0
  RISER1_INLET_TMP    Normal    45.0       C      N/A       60.0      N/A       70.0
  RISER2_INLET_TMP    Normal    41.0       C      N/A       60.0      N/A       70.0
  RISER1_OUTLETTMP    Normal    50.0       C      N/A       60.0      N/A       70.0
  RISER2_OUTLETTMP    Normal    41.0       C      N/A       60.0      N/A       70.0
  FP_TEMP_SENSOR      Normal    37.0       C      N/A       60.0      N/A       70.0
  DDR3_P1_A1_TEMP     Normal    42.0       C      N/A       65.0      N/A       85.0
  DDR3_P1_B1_TEMP     Normal    43.0       C      N/A       65.0      N/A       85.0
  DDR3_P1_C1_TEMP     Normal    44.0       C      N/A       65.0      N/A       85.0
  DDR3_P1_D1_TEMP     Normal    44.0       C      N/A       65.0      N/A       85.0
  DDR3_P2_E1_TEMP     Normal    43.0       C      N/A       65.0      N/A       85.0
  DDR3_P2_F1_TEMP     Normal    43.0       C      N/A       65.0      N/A       85.0
  DDR3_P2_G1_TEMP     Normal    42.0       C      N/A       65.0      N/A       85.0
  DDR3_P2_H1_TEMP     Normal    41.0       C      N/A       65.0      N/A       85.0
  VICP81E_0_TMP3      Normal    56.0       C      N/A       85.0      N/A       90.0
  PSU1_TEMP           Normal    37.0       C      N/A       60.0      N/A       65.0
  PCH_TEMP_SENS       Normal    51.0       C      N/A       80.0      N/A       85.0
```

Power Supply Status

To monitor the temperature state of the various sensors available on the APICs use the following steps.

CLI

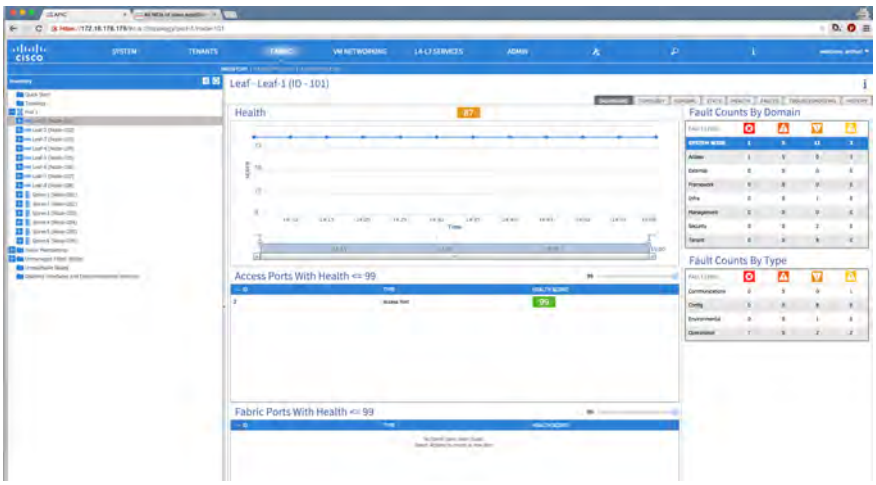
```
C220-FCH1807V02V /sensor # show psu
Name                Sensor   Reading    Units     Min.      Max.      Min.      Max.
                    Status                        Warning   Warning   Failure   Failure
------------------- -------- --------  --------  --------- --------- --------- ---------
P1_TEMP_SENS        Normal    49.5      C           N/A       81.0      N/A       86.0
POWER_USAGE         Normal     160      Watts       N/A        N/A      N/A       800
PSU1_POUT           Normal     136      Watts       N/A        624      N/A       648
PSU1_PIN            Normal     160      Watts       N/A        720      N/A       744
PSU1_STATUS         Normal              present
PSU2_STATUS         Normal              absent
PSU1_PWRGD          Normal              good
PSU1_AC_OK          Normal              good
```

Monitoring Leaf Switches

Leaf switches in the ACI fabric typically equate to the Nexus 9300 family of switches (with the exception of the Nexus 9336 switch).

The leaf switches provide first hop connectivity to anything that attaches to the fabric. Note that unlike traditional two or three tier designs where the "WAN" layer attaches to either the distribution/aggregation layer, in the ACI fabric, all external connectivity is provided through a set of leaf switches that provide high density 10gig or 40gig connectivity to elements outside the fabric.

To access the dashboard to monitor switches navigate to **Fabric** > **Inventory** > **Pod-1** > **Leaf-***



Leaf switch monitoring dashboard

Notice that the dashboard for this switch defaults to presenting the health score of the switch at a node level on the dashboard, and the sub tabs on the right hand side (topology/general/stats/health/faults/troubleshooting/history) can be used to quickly drill down into the various properties of the switch to understand how the switch is deployed from a hardware configuration standpoint, remediate faults on the switch, and troubleshoot the switch from a hardware perspective.

Monitoring Switch CPU Utilization

There are several methods to poll CPU utilization and trend it over different periods of time. The following sections describe a few of the methods available.

REST API

Spine and Leaf switches CPU utilization can be monitored using the following classes, based on the desired timescale and granularity.

proc:SysCPU5min A class that represents the most current statistics for System cpu in a 5 minute sampling interval. This class updates every 10 seconds.

proc:SysCPU15min A class that represents the most current statistics for System cpu in a 15 minute sampling interval. This class updates every 5 minutes.

proc:SysCPU1h A class that represents the most current statistics for System cpu in a 1 hour sampling interval. This class updates every 15 minutes.

proc:SysCPU1d A class that represents the most current statistics for System cpu in a 1 day sampling interval. This class updates every hour.

proc:SysCPU1w A class that represents the most current statistics for System cpu in a 1 week sampling interval. This class updates every day.

proc:SysCPU1mo A class that represents the most current statistics for System cpu in a 1 month sampling interval. This class updates every day.

proc:SysCPU1qtr A class that represents the most current statistics for System cpu in a 1 quarter sampling interval. This class updates every day.

proc:SysCPU1year A class that represents the most current statistics for System cpu in a 1 year sampling interval. This class updates every day.

ACME would like to see the average CPU utilization of all of the fabric switches over the last day.

```
http[s]://apic_ip//api/node/class/procSysCPU1d.xml?
```

CLI

```
Leaf-1# show proc cpu sort

PID    Runtime(ms)  Invoked   uSecs  1Sec    Process
-----  -----------  --------  -----  ------  -----------
 4012        69510    493837    140    1.3%  t2usd_tor
 4065         7239     27609    262    1.3%  python
 4292         3841    134758     28    0.8%  svc_ifc_opflexe
 4391         2355      4423    532    0.4%  nginx
 4067         1911       206   9278    0.4%  svc_ifc_policye
 4302         1904      1862   1022    0.3%  svc_ifc_observe
```

```
 4311         1811      1018   1779    0.3%  svc_ifc_confele
 4123         1407       251   5606    0.3%  svc_ifc_eventmg
 4310         1802       689   2616    0.3%  svc_ifc_dbgrele
 4846       119693     36527   3276    0.2%  stats_manager
 3923        15406      2645   5824    0.1%  pfmclnt
 4864         2361      2812    839    0.1%  ospfv3
 4865         2402      2717    884    0.1%  ospf
13606          435       211   2065    0.0%  bgp
 4296         6263      7413    844    0.0%  snmpd
 4297         6667      4542   1467    0.0%  dc3_sensor
 4299         8559      8225   1040    0.0%  policy_mgr
 4301         1860     19152     97    0.0%  plog
 4866         2792      3269    854    0.0%  isis
 5025         1611      1743    924    0.0%  mcecm
```

In order to obtain a historical view of CPU utilization from the CLI it may be necessary to jump into an alternative shell from the switch bash prompt. This shell is called vsh (or v-shell).

```
Leaf-1# show processes cpu history


           1      1  33                     1
    74655466188583305537657254553466766355478503394364566533564 4
100
 90
 80
 70
 60
 50
 40                    #
 30                   ##
 20                   ##
 10 # ### #######  ######## # ##  ##### ## ####  #  # ####  ##
    0....5....1....1....2....2....3....3....4....4....5....5....
              0    5    0    5    0    5    0    5    0    5

               CPU% per second (last 60 seconds)
                      # = average CPU%
```

```
    32 133111341111111113111111111131 11111113 1111 11231 1111111
    74951380043220632835337073217560934200076902579114419268011 7
100
 90
 80
 70
 60
 50
 40 *         *                   *         *
 30 *   **   **         *         *         *         *
 20 ** ****  **   *  * ** * *   ***        **   **   **   **   *
 10 ############################################################
    0....5....1....1....2....2....3....3....4....4....5....5....
              0    5    0    5    0    5    0    5    0    5

               CPU% per minute (last 60 minutes)
              * = maximum CPU%   # = average CPU%

      1
    440
    030
100   *
 90   *
 80   *
 70   *
 60   *
 50   *
 40 ***
 30 ***
 20 ***
 10 ###
    0....5....1....1....2....2....3....3....4....4....5....5....6....6....7.
              0    5    0    5    0    5    0    5    0    5    0    5    0

                   CPU% per hour (last 72 hours)
                  * = maximum CPU%   # = average CPU%
```

Monitoring Switch Memory Utilization

There are several methods to poll memory utilization and trend it over different periods of time. The following sections describe a few of the methods available.

REST API

Spine and Leaf switches memory utilization can be monitored using the following classes, based on the desired timescale and granularity.

proc:SysMem5min A class that represents the most current statistics for System memory in a 5 minute sampling interval. This class updates every 10 seconds.

proc:SysMem15min A class that represents the most current statistics for System memory in a 15 minute sampling interval. This class updates every 5 minutes.

proc:SysMem1h A class that represents the most current statistics for System memory in a 1 hour sampling interval. This class updates every 15 minutes.

proc:SysMem1d A class that represents the most current statistics for System memory in a 1 day sampling interval. This class updates every hour.

proc:SysMem1w A class that represents the most current statistics for System memory in a 1 week sampling interval. This class updates every day.

proc:SysMem1mo A class that represents the most current statistics for System memory in a 1 month sampling interval. This class updates every day.

proc:SysMem1qtr A class that represents the most current statistics for System memory in a 1 quarter sampling interval. This class updates every day.

proc:SysMem1year A class that represents the most current statistics for System memory in a 1 year sampling interval. This class updates every day.

ACME would like to monitor memory over the last day, and would use the following REST call:

```
http[s]://apic_ip/api/node/class/procSysMem1d.xml?
```

CLI

```
Leaf-1# show system resources
Load average:   1 minute: 1.21   5 minutes: 1.14   15 minutes: 0.80
Processes   :   513 total, 2 running
CPU states  :   4.1% user,   2.5% kernel,   93.4% idle
Memory usage:   16401072K total,   9054020K used,   7347052K free
Current memory status: OK
```

SNMP

As mentioned in the URL for the SNMP reference guide for ACI release 1.x, the following SNMP objects are supported from an SNMP polling perspective. See http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html

Monitoring File System Health

CLI

Currently, the CLI is only way to monitor the utilization of the file system on the leaves. It is normal to see a higher % utilization on some of the mount points in the file system hierarchy. The critical volumes to keep track of in terms of utilization are /volatile, bootflash and logflash

```
Leaf-1# df
df: `/nxos/tmp': No such file or directory
df: `/var/home': No such file or directory
df: `/var/tmp': No such file or directory
df: `/nginx': No such file or directory
df: `/debugfs': No such file or directory
df: `/recovery': No such file or directory
df: `/cfg0': No such file or directory
df: `/cfg1': No such file or directory
df: `/logflash/INXOS_SYSMGR/startup-cfg': No such file or directory
df: `/mnt/plog': No such file or directory
Filesystem     1K-blocks    Used Available Use% Mounted on
```

```
rootfs            512000    1064    510936   1% /
rootfs            512000    1064    510936   1% /
none              512000    1064    510936   1% /isan
none              512000    1064    510936   1% /var
none               51200    2288     48912   5% /etc
none               51200     108     51092   1% /var/log
none             3145728  336664   2809064  11% /dev/shm
none              512000       0    512000   0% /volatile
/dev/sda4        7782036 1080636   6306088  15% /bootflash
/dev/sda5          60485    5356     52006  10% /mnt/cfg/0
/dev/sda6          60485    5356     52006  10% /mnt/cfg/1
/dev/sda3         120979   13349    101384  12% /mnt/pss
/dev/sda7         512000    1064    510936   1% /logflash
/dev/sda9       15748508  591216  14357292   4% /mnt/ifc/cfg
/dev/sda8       15748508  991204  13957304   7% /mnt/ifc/log
/dev/sda8       15748508  991204  13957304   7% /var/log/dme/oldlog
/dev/sda9       15748508  591216  14357292   4% /controller
rootfs            716800  665728     51072  93% /mnt/ifc/cfg/mgmt/opt/controller/sbin
rootfs            716800  665728     51072  93% /controller/sbin
/dev/sda8       15748508  991204  13957304   7% /data/techsupport
rootfs            716800  665728     51072  93% /bin
/dev/sda4        7782036 1080636   6306088  15% /bootflash
rootfs            716800  665728     51072  93% /data/challenge.old.plugin
/dev/sda9       15748508  591216  14357292   4% /controller
rootfs            716800  665728     51072  93% /controller/sbin
rootfs            716800  665728     51072  93% /dev
none             3145728  336664   2809064  11% /dev/shm
none               51200    2288     48912   5% /etc
none             2097152  682360   1414792  33% /isan/plugin/0/isan/utils
none             2097152  682360   1414792  33% /isan/plugin/0/lc/isan/utils
none             2097152  682360   1414792  33% /isan/plugin/0/lc/isan/lib
none             2097152  682360   1414792  33% /isan/plugin/0/isan/lib
none             2097152  682360   1414792  33% /isan/lib
none             2097152  682360   1414792  33% /isan/plugin/0/lib
none             2097152  682360   1414792  33% /isan/utils
rootfs            716800  665728     51072  93% /lc/isan/utils
rootfs            716800  665728     51072  93% /lib
rootfs            716800  665728     51072  93% /mnt/cfg
/dev/sda5          60485    5356     52006  10% /mnt/cfg/0
```

```
/dev/sda6          60485    5356     52006  10% /mnt/cfg/1
rootfs            716800  665728     51072  93% /mnt/ifc
/dev/sda9       15748508  591216  14357292   4% /mnt/ifc/cfg
rootfs            716800  665728     51072  93% /mnt/ifc/cfg/mgmt/opt/controller/sbin
/dev/sda8       15748508  991204  13957304   7% /mnt/ifc/log
/dev/sda3         120979   13349    101384  12% /mnt/pss
rootfs            716800  665728     51072  93% /sbin
/dev/sda8       15748508  991204  13957304   7% /data/techsupport
none             1572864   39444   1533420   3% /tmp
rootfs            716800  665728     51072  93% /usr
none               51200     108     51092   1% /var/log
/dev/sda8       15748508  991204  13957304   7% /var/log/dme/oldlog
none               51200     108     51092   1% /var/log/messages
rootfs            716800  665728     51072  93% /var/log/dme
rootfs            716800  665728     51072  93% /var/log/dme/nginx
rootfs            716800  665728     51072  93% /usr/share/vim
/dev/sda7       11811760  375608  10836140   4% /var/log/dme/log
/dev/sda8       15748508  991204  13957304   7% /var/log/dme/oldlog
none              512000    1064    510936   1% /var/run/mgmt/log
none              512000    1064    510936   1% /var/run/utmp
/dev/sda7       11811760  375608  10836140   4% /var/sysmgr
none               40960       8     40952   1% /var/sysmgr/startup-cfg
/dev/sda7       11811760  375608  10836140   4% /logflash/core
rootfs            716800  665728     51072  93% /usb
none              512000       0    512000   0% /volatile
```

Monitoring CoPP (Control Plane Policing) Statistics

CLI

CoPP is enabled by default and the parameters cannot be changed at this time. CoPP statistics are available through the CLI.

To show the CoPP policy that is programed by the system use the following CLI command:

```
Leaf-1# show copp policy
COPP Class           COPP proto           COPP Rate            COPP Burst
```

mcp	mcp	500	500
ifc	ifc	5000	5000
igmp	igmp	1500	1500
nd	nd	1000	1000
cdp	cdp	1000	1000
pim	pim	500	500
dhcp	dhcp	1360	340
lacp	lacp	1000	1000
ospf	ospf	2000	2000
arp	arp	1360	340
lldp	lldp	1000	1000
acllog	acllog	500	500
stp	stp	1000	1000
eigrp	eigrp	2000	2000
coop	coop	5000	5000
traceroute	traceroute	500	500
isis	isis	1500	5000
icmp	icmp	500	500
bgp	bgp	5000	5000

To show drops against specific CoPP classes, use the following CLI command:

```
Leaf-1# show copp policy stats
```

COPP Class	COPP proto	COPP Rate	COPP Burst	AllowPkts	AllowBytes	DropPkts	DropBytes
mcp	mcp	500	500	0	0	0	0
ifc	ifc	5000	5000	195072	161613961	0	0
igmp	igmp	1500	1500	3	192	0	0
nd	nd	1000	1000	6	564	0	0
cdp	cdp	1000	1000	494	140543	0	0
pim	pim	500	500	0	0	0	0
dhcp	dhcp	1360	340	4	1400	0	0
lacp	lacp	1000	1000	0	0	0	0
ospf	ospf	2000	2000	0	0	0	0
arp	arp	1360	340	1284	90068	0	0
lldp	lldp	1000	1000	5029	1717208	0	0
acllog	acllog	500	500	0	0	0	0
stp	stp	1000	1000	0	0	0	0
eigrp	eigrp	2000	2000	0	0	0	0
coop	coop	5000	5000	4722	470546	0	0

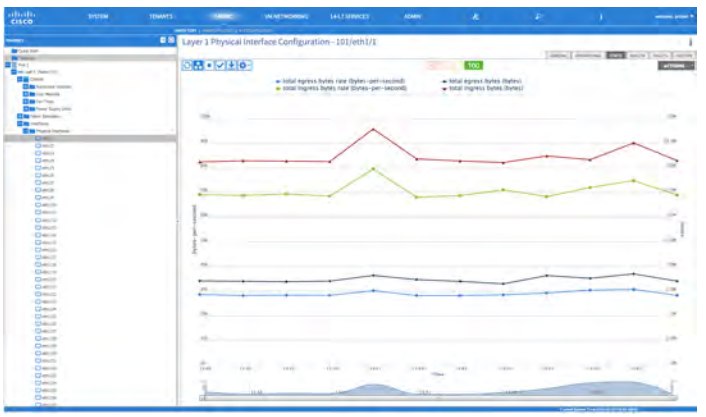
```
traceroute   traceroute   500    500    0       0         0    0
isis         isis         1500   5000   17141   2167565   0    0
icmp         icmp         500    500    0       0         0    0
bgp          bgp          5000   5000   864     73410     0    0
```

Physical Interface Statistics and Link State

GUI

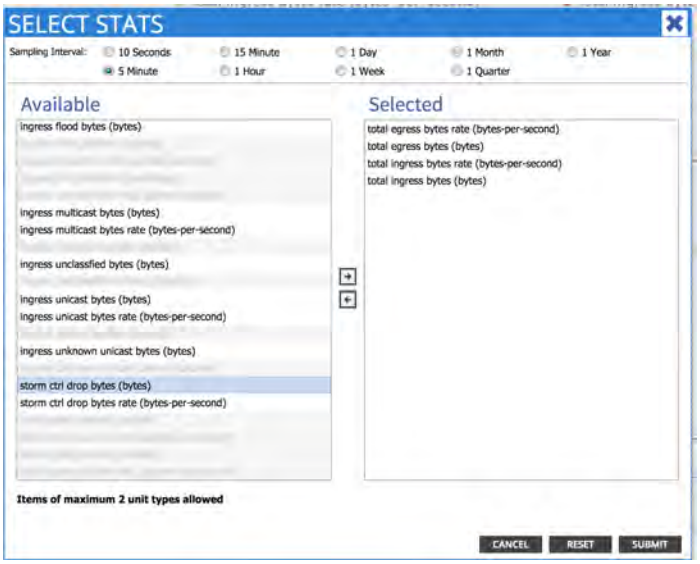
To access interface link state information, in the APIC GUI, navigate to **Fabric** > **Inventory** > **Pod-1** > **Leaf-X** > **Interfaces** > **Physical Interfaces**. In the work pane, the oper state column displays the operational state of the link. Note that there are other tabs available in the work pane that reference other types of interfaces like port-channels, virtual port-channels, routed interfaces, loopbacks, etc.

To access interface statistics, in the APIC GUI, navigate to **Fabric** > **Inventory** > **Pod-1** > **Leaf-X** > **Interfaces** > **Physical interfaces** > **Eth X/Y**, and then click on the **Stats** tab in the work pane on the right-hand side.



Physical interface throughput statistics

Note that clicking on the check icon enables you to select additional statistics that can be graphed similar to the following screenshot.



Granular physical interface statistics

REST API

For customers that prefer the REST API interface to poll for interface statistics, several objects are available. There are several such counters that are available (e.g. RX/TX, input/output / duplex, 30 second rates, 5 minute rate, unicast packets, multicast packets, etc.). As a pointer, the parent managed object is provided below, as the children can be derived from it.

It is expected that the reader has a good understanding of the object model and is able to navigate through the model to obtain the information desiered using the example below, information provided in preceding sections and the tools described therein

An example of the base API call for physical interface statistics is:

```
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/phys-[eth1/1].json
```

For example, to determine the total ingress bytes on Leaf 101 port Eth1/1, the ACME operator could issue the following API call:

```
/topology/pod-1/node-101/sys/phys-[eth1/1].json
```

Visore allows the operator to dig deeper into the hierarchical tree. From the prior command, the operator could see children of the interface object, such as ingress and egress bytes. One of the child objects includes the following:

```
topology/pod-1/node-101/sys/phys-[eth1/1]/dbgEtherStats&#10
```

CLI

The show interface eth x/y command can be used to monitor interfaces from the CLI. Other supported commands include "show interface port-channel x/y"

```
Leaf-1# show int e1/1
Ethernet1/1 is up
admin state is up, Dedicated Interface
  Hardware: 1000/10000/auto Ethernet, address: 7c69.f60f.8771 (bia 7c69.f60f.8771)
  MTU 9000 bytes, BW 1000000 Kbit, DLY 1 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is trunk
  full-duplex, 1000 Mb/s, media type is 1G
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Rate mode is dedicated
  Switchport monitor is off
  EtherType is 0x8100
  EEE (efficient-ethernet) : n/a
  Last link flapped 04:19:13
  Last clearing of "show interface" counters never
  1 interface resets
  30 seconds input rate 169328 bits/sec, 97 packets/sec
  30 seconds output rate 424528 bits/sec, 115 packets/sec
```

```
Load-Interval #2: 5 minute (300 seconds)
  input rate 644416 bps, 134 pps; output rate 365544 bps, 114 pps
RX
  2474537 unicast packets  8434 multicast packets  2 broadcast packets
  2482973 input packets  1686129815 bytes
  0 jumbo packets  0 storm suppression bytes
  0 runts  0 giants  0 CRC  0 no buffer
  0 input error  0 short frame  0 overrun   0 underrun  0 ignored
  0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
  0 input with dribble  712 input discard
  0 Rx pause
TX
  1673907 unicast packets  575 multicast packets  7 broadcast packets
  1674489 output packets  455539518 bytes
  0 jumbo packets
  0 output error  0 collision  0 deferred  0 late collision
  0 lost carrier  0 no carrier  0 babble  0 output discard
  0 Tx pause
```

SNMP

As mentioned in the URL for the SNMP reference guide for ACI release 1.x, the following SNMP objects are supported from an SNMP polling perspective for interfaces See: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html

Module Status

Even though the leaves are considered fixed switches, they have a supervisor component which refers to the CPU complex. From a forwarding perspective, there are two data plane components, viz. the NFE (Network Forwarding Engine ASIC) which provide the front panel ports, and the ALE or ALE2 (Application Leaf Engine ASIC) depending on the generation of switch hardware, which provides uplink connectivity to the spines. The following methods can be used to determine the status of the modules in the switch.

GUI

To access module status for the NFE and the CPU complex, in the APIC GUI, navigate to Fabric > Inventory > Pod-1 > Leaf-X > Chassis > Module > Supervisor modules and the status of the module is displayed in the work pane.

To access module status for the ALE/ALE2, in the APIC GUI, navigate to Fabric > Inventory > Pod-1 > Leaf-X > Chassis > Module > Line modules and the status of the module is displayed in the work pane.

REST API

The following REST API call(s) can be used to monitor the state of the supervisor and the module.

```
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/ch/supslot-1/sup
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/ch/lcslot-1/lc
```

CLI

The show module command can be used to obtain the status of the base module and the uplink module.

```
Leaf-1# show module

Mod  Ports  Module-Type                        Model              Status
---  -----  ---------------------------------- ------------------ ----------
1    48     1/10G Ethernet Module              N9K-C9396PX        active
GEM  12     40G Ethernet Expansion Module      N9K-M12PQ          ok

Mod  Sw              Hw
---  --------------  ------
1    11.1(0.152)     0.2050

Mod  MAC-Address(es)                         Serial-Num
---  --------------------------------------  ----------
1    7c-69-f6-0f-87-71 to 7c-69-f6-0f-87-ad  SAL17267Z9U
```

```
Mod  Online Diag Status
---  ------------------
1    pass
```

SNMP

As mentioned in the URL for the SNMP reference guide for ACI release 1.x, the following SNMP objects are supported from an SNMP polling perspective for modules. See: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html

Switch Fan Status

The following section describes methodologies to retrieve the status of the fan trays on the leaf switches.

GUI

To access fan status for the leaf switch, in the APIC GUI, navigate to Fabric > Inventory > Pod-1 > Leaf-X > Chassis > Fan Tray and the status of the modules is displayed in the work pane.

REST API

The following REST API call(s) and their child objects can be used to monitor the state of the fans on a leaf switch (note that there are 3 slots on this particular switch).

```
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/ch/ftslot-1
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/ch/ftslot-2
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/ch/ftslot-3
```

CLI

The following CLI's can be used to monitor the state of the fans on a leaf switch.

```
Leaf-1# show environment fan
Fan:
```

```
-----------------------------------------------------
Fan             Model                Hw         Status
-----------------------------------------------------
Fan1(sys_fan1)  N9K-C9300-FAN1-B     --         ok
Fan2(sys_fan2)  N9K-C9300-FAN1-B     --         ok
Fan3(sys_fan3)  N9K-C9300-FAN1-B     --         ok
Fan_in_PS1      --                   --         unknown
Fan_in_PS2      --                   --         ok
Fan Speed: Zone 1:  0x5f
Fan Air Filter : Absent
```

SNMP

As mentioned in the URL for the SNMP reference guide for ACI release 1.x, the following SNMP objects are supported from an SNMP polling perspective for fan trays (http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html).

Power Supply Status

The following sections describe methodologies to retrieve the status of the power supplies on the leaf switches

GUI

To access power supply status for the leaf switch, in the APIC GUI, navigate to **Fabric > Inventory > Pod-1 > Leaf-X > Chassis > Power Supply Units** and the status of the modules is displayed in the work pane.

REST API

The following REST API call(s) and their child objects can be used to monitor the state of the fans on a leaf switch (note that there are 3 slots on this particular switch).

```
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/ch/psuslot-1
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/ch/psuslot-2
```

CLI

The following CLI commands can be used to monitor the state of the fans on a leaf switch:

```
Leaf-1# show environment power
Power Supply:
Voltage: 12.0 Volts

Power                                Actual         Total
Supply    Model                      Output      Capacity    Status
                                    (Watts )      (Watts )
-------  ------------------  -----------  -----------  --------------
1        UCSC-PSU-650W                0 W        648 W       shut
2        UCSC-PSU-650W              168 W        648 W         ok
                                    Actual         Power
Module     Model                      Draw     Allocated    Status
                                    (Watts )      (Watts )
--------  ------------------  -----------  -----------  --------------
1         N9K-C9396PX                168 W        456 W    Powered-Up
fan1      N9K-C9300-FAN1-B             N/A          N/A    Powered-Up
fan2      N9K-C9300-FAN1-B             N/A          N/A    Powered-Up
fan3      N9K-C9300-FAN1-B             N/A          N/A    Powered-Up
N/A - Per module power not available
 Power Usage Summary:
--------------------
Power Supply redundancy mode (configured)                Non-Redundant(combined)
Power Supply redundancy mode (operational)               Non-Redundant(combined)
 Total Power Capacity (based on configured mode)               648  W
Total Power of all Inputs (cumulative)                         648  W
Total Power Output (actual draw)                               168  W
Total Power Allocated (budget)                                 N/A
Total Power Available for additional modules                   N/A
```

SNMP

As mentioned in the URL for the SNMP reference guide for ACI release 1.x, the following SNMP objects are supported from an SNMP polling perspective for power supplies.

See: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html

LLDP Neighbor Status

The APIC provides a single pane of glass to query and determine all LLDP neighbors in a fabric.

To obtain a list of LLDP neighbors on an interface, navigate to **Fabric** > **Inventory** > **Pod-1** > **Leaf-X** > **Protocols** > **LLDP** > **Neighbors** > **eth x/y**

A full listing of all LLDP neighbors on the interface can be obtained in the work pane.

In the above workflow clicking on **Neighbors** (instead of **eth x/y**) gives you a list of all LLDP neighbors on the switch.

REST API

The following rest API call can be used to obtain the same information:

```
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/lldp/inst/if-[eth1/1]
```

CLI

```
Leaf-1# show lldp neighbors
Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID            Local Intf      Hold-time  Capability  Port ID
apic2                 Eth1/1          120                    4c:4e:35:09:77:2f
apic3                 Eth1/4          120                    90:e2:ba:4b:fa:d4
5548-2                Eth1/7          120        B           Eth1/3
Spine-1               Eth1/49         120        BR          Eth4/1
Spine-2               Eth1/50         120        BR          Eth4/1
Spine-3               Eth1/51         120        BR          Eth1/3
Spine-4               Eth1/52         120        BR          Eth1/3
Spine-5               Eth1/53         120        BR          Eth1/5
```

```
Spine-6              Eth1/54         120        BR          Eth1/5
Total entries displayed: 9
```

SNMP

As mentioned in the URL for the SNMP reference guide for ACI release 1.x, the following SNMP objects are supported from an SNMP polling perspective for LLDP. See: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html

APIC provides a single pane of glass to query and determine all CDP neighbors in a fabric. CDP Neighbor Status:

GUI

To obtain a list of CDP neighbors on an interface, navigate to **Fabric > Inventory > Pod-1 > Leaf-X > Protocols > CDP > Neighbors > eth x/y**

A full listing of all CDP neighbors on the interface can be obtained in the work pane.

In the above workflow clicking on **Neighbors** (instead of **eth x/y**) gives you a list of all LLDP neighbors on the switch.

REST API

The following rest API call can be used to obtain the same information:

```
https://{{apic-ip}}/api/node/mo/topology/pod-1/node-101/sys/cdp/inst/if-[eth1/1]
```

CLI

```
Leaf-1# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device-ID          Local Intrfce  Hldtme  Capability  Platform      Port ID
```

```
Services-UCS-A(SSI15450J63)
                     Eth1/5          129     S I s      UCS-FI-6248UP  Eth1/17
5548-2(SSI154300VL)
                     Eth1/7          123     S I s      N5K-C5548UP    Eth1/3
```

SNMP

As mentioned in the URL for the SNMP reference guide for ACI release 1.x, the following SNMP objects are supported from an SNMP polling perspective for CDP. See: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html

GOLD Diagnostic Results

GOLD is covered in greater detail in the Hardware Replacement section.

GOLD diagnostics provide an easy and quick way for operations teams to confirm that bootup and non-disruptive tests that run during normal operations have executed properly, as well as the ability to run on demand diagnostics to isolate potential hardware at fault.

GUI

To view GOLD Diagnostic test results in the GUI for the Supervisors, click on **Fabric > Inventory > Pod-1 > Leaf-1 > Chassis > Supervisor Modules > Slot-1**. Then click troubleshooting in the work pane.

To view the same for modules, click on **Fabric > Inventory > Pod-1 > Leaf-1 > Chassis > Line Modules > Slot-x**. Then click **Troubleshooting** in the work pane.

CLI

```
Leaf-1# show diagnostic result module all
Current bootup diagnostic level: bypass
Module 1: 1/10G Ethernet Module  (Active)
    Test results: (. = Pass, F = Fail, I = Incomplete,
    U = Untested, A = Abort, E = Error disabled)

     1) bios-mem----------------------->  .
     2) mgmtp-lb----------------------->  .
     4) nsa-mem------------------------>  .
     6) fabp-prbs:

      Port      1  2  3  4  5  6  7  8  9 10 11 12
      ---------------------------------------
             .  .  .  .  .  .  .  .  .  .  .  .

    22) cpu-cache--------------------->  .
    23) mem-health-------------------->  .
    24) ssd-acc----------------------->  .
    25) act2-acc---------------------->  .
    26) ge-eeprom--------------------->  .
    29) usb-bus----------------------->  .
    30) cons-dev---------------------->  .
    31) obfl-acc---------------------->  .
    32) nvram-cksum------------------->  .
    33) fpga-reg-chk------------------>  .
    34) asic-scratch------------------>  .
    40) rtc-test---------------------->  .
    41) pcie-bus---------------------->  .
```

Proactive Monitoring Use Cases

Monitoring Workload Bandwidth

ACME would like to proactively monitor connections to servers to determine whether adequate bandwidth is available to a given workload. This enables them to answer questions about whether a server needs to be upgraded from 10G to 40G, or multiple interfaces need to be bonded to provide more bandwidth to the server. The following will provide an example of how statistics policies and thresholds can be used to alert the operators when additional bandwidth is required to a server.

ACME's server team has determined that they would like to monitor link utilization over a 5 minute period, and when the average utilization is above 80%, they would like to raise a fault for the affected interface. To accomplish these tasks requires configuring a monitoring policy with two distinct types of policies. First, ACME will configure a stats collection policy to collect the average utilization of a link over the desired 5 minute interval. Next they will configure a threshold policy to generate a fault when the link utilization exceeds various thresholds according to the following table.

Create an interface monitoring policy.

1. On the menu bar, choose **Fabric > Access Policies**.
2. In the Navigation pane, choose **Monitoring Policies**.
3. In the Work pane, choose **Actions > Create Monitoring Policy**.
4. In the **Create Monitoring Policy** dialog box, perform the following actions:
   a. Expand the newly created monitoring policy and select **Stats Collection Policy**.
   b. Click on the **edit** button next to **Monitoring Object**, and select **L1 Physical Interface Configuration**.
   c. From the **Monitoring Object** drop-down list, choose **L1 Physical Interface Configuration**.
   d. Click the **edit** button next to **Stats Type** and select **Ingress** and **Egress**.
   e. From the stats type drop-down list, choose **Ingress**.
   f. Click **+**.
   g. Select **5 minutes** from the granularity column, and click on **update**.

We have now created a policy which will monitor associated interfaces and ingress traffic rate at 5 minute intervals. Next we will configure a threshold which will alert us if the utilization exceeds 80% and assign an default fault severity to it.

1. Click **+** in the **Config Thresholds** column.
2. In the **Thresholds for Collection** 5 minute window, click **+**.
3. Click on **Ingress Link utilization average value**.
4. Enter **0** for the normal value and click the **Rising** radio button.

Note: In this example, we are interested in an absolute percentage, these policies can be further customized to provide a normal value, and look for deviations above or below that normal value.

In this scenario, 80% utilization does not necessarily mean that the application performance is degraded, so we will flag this as a warning. Additional levels/severities can also be specified if desired.

The **set** column specifies the level at which the fault will be raised, and the reset column will specify the the level at which the fault will be cleared. For example, we will be raising a warning when the utilization goes above 80, and clear the warning when the utilization falls below 75. Repeat these steps for the egress statistics as well.

Finally, we will associate the newly created policy with an interface policy group that represents the interfaces we to monitor with this policy.

For our example, we will apply the policy to the UCS-10G-PG

1. On the menu bar, choose **Fabric > Access Policies.**
2. In the Work pane, choose **Interface Policies > Policy Groups**.
   a. Select **UCS-10G-PG**.
   b. From the **Monitoring Policy** drop down menu, select the newly created monitoring policy.

EPG Level Statistics

The application owner would like to be able to monitor network-related information for their application, such as the aggregate amount of traffic to a specific tier. As an example, we will monitor the amount of traffic to the web tier of a given application. In

this example, the default monitoring policies are appropriate, and they are simply extracting them from the system to be consumed externally. This information is useful in scenarios such as a new release being pushed, and to make sure that no traffic anomalies are created after the push.

This can be accomplished by navigating to the EPG, and selecting the **Stats** tab:

EPG-level throughput statistics

Additionally, this information can be gathered from the API:

```
http[s]://apic_ip/api/node/mo/uni/tn-mynewproject/ap-app1/epg-web-epg.xml?query-
target=self&rsp-subtree-include=stats
```

Reactive Monitoring

It is crucial that the ACME operational staff are able to react to any indication of something going wrong. If there is a notification that something has gone wrong, such as a fault notification, a low health score, or a ticket/report that end-user functionality has been impacted, knowledge of the available monitoring tools is important for the identification and collection of evidence. This evidence can then be used to identify and analyze the root cause of the problem before taking corrective action. For more information regarding faults and health scores please refer to those specific sections within this book.

A deep dive into the processes of troubleshooting is out of the scope of this book. Please refer to "*Troubleshooting Cisco Application Centric Infrastructure: Analytical problem solving applied to the policy driven data center*" available at: http://datacenter.github.io/aci-troubleshooting-book/

Tenant–Troubleshoot Policies

Within the APIC GUI, under each Tenant you can find a Troubleshoot Policy section. This section will allow configuration of policies that are specific to one tenant, and the monitoring of traffic and test connectivity between endpoints.

As seen in the image above, the following troubleshooting policies can be configured:

- SPAN (Switched Port ANalyzer)–Configuration of SPAN and ERSPAN sources and destinations to be used in external monitoring of Tenant traffic flows
- Endpoint-To-Endpoint Traceroute–Configuration of a path validation tool for verifying validity of communications between Tenant endpoints in an ACI fabric
- Atomic Counters–Configuration of a set of customizable counters to collect and report on information between a definable set of objects. As shown in the image below, a policy can be configured to collect statistics between EPs, between EPGs, between EPGs and specific IPs and other special objects, such as Any or External traffic flows

Fabric–Troubleshoot Policies

For troubleshooting within the entire fabric, there are the following tools and policies:

- SPAN (Switched Port Analyzer)—Configuration of SPAN and ERSPAN sources and destinations to be used in external monitoring of fabric traffic flows
- On-demand Diagnostics—Configuration of a policy for collection of diagnostic information that can be executed at a point in time and which will return a set of valuable output for investigation
- Leaf Nodes Traceroute—Configuration of a path validation tool for verifying validity of communications between ACI fabric nodes
- Traffic Map—At-a-glance hotspot map of node-to-node traffic flow in an ACI fabric

Enhanced Troubleshooting Wizard

From version 1.1, APIC provides a troubleshooting graphic tool to find relevant faults and statistics, recent changes, and run connectivity tests in a simple manner.

It also gives the option to generate a report to save the results so it can be used as a reference.

Other Tools

- iPing—A troubleshooting tool in the ACI fabric that can be used to verify reachability of a device connected to the fabric utilizing the fabric as the pervasive source
- Audit Logs—Audit logs are continually collected on all actions taken in an ACI fabric and can give a quick indication of which user took which actions at what time

Reactive Monitoring Use Cases

At the end of this chapter, we are going to describe two situations where ACME ran into issues and how they made use of the tools previously described.

1. No access to application: An end-user calls and reports that they can no longer access a web application running within the fabric.
2. Users report that an application running in the fabric is slow or there is a report of slowness to the web application running within the fabric.

Reactive Monitoring Tools

Switch Port Analyzer (SPAN)

SPAN is generally used in two ways:

- Proactively as part of a third party or offline analysis requirement.
  - Security tools (IDS/IPS, Data Loss Prevention)
  - Call Recording
- Troubleshooting application and networking issues within the fabric.

It may be helpful to perform a capture of some particular traffic to see what is going on within the stream of data. Looking through traffic flows will allow investigation of packet and protocol level details, such as traffic resets, misbehaving protocols, improper host requests or responses, or node level communications. This will provide deeper insight into how devices are using the network than simple traffic flow and fabric configuration review.

Switched Port ANalyzer, or SPAN, is a standard feature that allows copy and replication of traffic to a network analyzer for further decoding and investigation. It can be used to copy traffic from one or more ports, VLANs, or endpoint groups (EPGs).

The SPAN feature process is non-disruptive to any connected devices and is facilitated in hardware, which prevents any unnecessary CPU load.

SPAN sessions can be configured to monitor traffic received by the source (ingress traffic), traffic transmitted from the source (egress traffic), or both. By default, SPAN monitors all traffic, but filters can be configured to monitor only selected traffic.

Multinode SPAN

APIC traffic monitoring policies can enforce SPAN at the appropriate places to copy traffic from members of each End Point Group wherever they are connected. If a member moves, APIC automatically pushes the policy to the new leaf switch. For example, when a VMotion event relocates an Endpoint to a new leaf switch, the SPAN feature configuration automatically adjusts.

SPAN Guidelines and Restrictions

- Use SPAN for troubleshooting. SPAN traffic competes with user traffic for switch resources. To minimize the load, configure SPAN to copy only the specific traffic that you want to analyze.
- An l3extLIfP Layer 3 subinterface cannot be configured as a SPAN source. The entire port must be used for monitoring traffic from external sources.
- Tenant and access SPAN use the encapsulated remote extension of SPAN (ERSPAN) type I, while fabric SPAN uses ERSPAN type II. For information regarding ERSPAN headers, refer to the IETF Internet Draft at this URL: https://tools.ietf.org/html/draft-foschiano-erspan-00.
- See the *Verified Scalability Guide for Cisco ACI* document for SPAN-related limits, such as the maximum number of active SPAN sessions.

Configuring a SPAN Session

This procedure shows how to configure a SPAN policy to forward replicated source packets to a remote traffic analyzer.

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name* > Troubleshooting Policies > SPAN > SPAN Destination Groups**.
4. In the Work pane, choose **Actions > Create SPAN Destination Group**.
5. In the **Create SPAN Destination Group** dialog box, perform the following actions:
   a. In the **Name** field, enter a name for the SPAN destination group.
   b. In the **Destination EPG** dropdowns, select the destination **Tenant, Application Profile,** and **EPG.**
   c. Enter the **Destination IP**.
   d. Enter the **Source IP Prefix.**
   e. Optionally, modify the other fields as needed.
   f. Click **OK**.
   g. If needed, add additional destinations.
   h. Click **Submit.**

6 Under **SPAN**, right-click **SPAN Source Groups** and choose **Create SPAN Source Group**.

7 In the **Create SPAN Source Group** dialog box, perform the following actions:

a. In the **Name** field, enter a name for the SPAN source group.

b. From the **Destination Group** drop-down list, choose the SPAN destination group that you configured previously.

c. In the **Create Sources** table, click the + icon to open the **Create Sources** dialog box.

d. In the **Name** field, enter a name for the source.

e. In the **Direction** field, choose the radio button based on whether you want to replicate and forward packets that are incoming to the source, outgoing from the source, or both incoming and outgoing.

f. From the **Source EPG** drop-down list, choose the EPG (identified by Tenant/ApplicationProfile/EPG) whose packets will be replicated and forwarded to the SPAN destination. Click **OK** to save the SPAN source.

g. Click **Submit** to save the SPAN source group.

Traceroute

Traceroute is a useful feature in traditional networking. In ACI this feature is implemented taking into account the way the fabric works.

Traceroute supports a variety of modes, including endpoint-to-endpoint, and leaf-to-leaf (tunnel endpoint, or TEP to TEP). It discovers all paths across the fabric, discovers point of exits for external endpoints, and helps to detect if any path is blocked.

A traceroute that is initiated from the tenant endpoints shows the default gateway as an intermediate hop that appears at the ingress leaf switch.

Note: If traceroute is done from the OS of a connected server or VM, it will show the hops for the leaves and spines as unknown, and will keep recording the information after the packet gets out of the fabric. For more precise information, please use traceroute from the APIC (GUI or CLI)

Traceroute Guidelines and Restrictions

- When the traceroute source or destination is an endpoint, the endpoint must be dynamic and not static. Unlike a dynamic endpoint (fv:CEp), a static endpoint (fv:StCEp) does not have a child object (fv:RsCEpToPathEp) that is required for traceroute.
- See the *Verified Scalability Guide for Cisco* ACI document for traceroute-related limits.
- Traceroute results will display the IP address of the remote node and interface which it came it came in on. (Browse on the APIC GUI to Fabric | Inventory | Fabric Management to view the IP address information for correlation.)
- Traceroute cannot be used for endpoints that reside in an external EPG.

Performing a Traceroute Between Endpoints

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name* > Troubleshooting Policies > Endpoint-to-Endpoint Traceroute Policies**.
4. In the Work pane, choose **Actions > Create Endpoint-to-Endpoint Traceroute Policy**.
5. In the **Create Endpoint-to-Endpoint Traceroute Policy** dialog box, perform the following actions:
   a. In the **Name** field, enter a name for the traceroute policy.
   b. In the **Source End Points** table, click the + icon to edit the traceroute source.
   c. From the **Source MAC** drop-down list, choose or enter the MAC address of the source endpoint and click **Update**.
   d. In the **Destination End Points** table, click the + icon to edit the traceroute destination.
   e. From the **Destination MAC** drop-down list, choose or enter the MAC address of the destination endpoint and click **Update**.
   f. In the **State** field, click the **Start** radio button.
   g. Click **Submit** to launch the traceroute.
6. In the **Navigation** pane or the **Traceroute Policies** table, click the traceroute policy. The traceroute policy is displayed in the Work pane.

7 In the Work pane, click the Operational tab, click the Source End Points tab, and click the Results tab.

8 In the Traceroute Results table, verify the path or paths that were used in the trace.
   a. More than one path might have been traversed from the source node to the destination node.
   b. For readability, increase the width of one or more columns, such as the Name column.

Atomic Counters

Atomic Counters are useful for troubleshooting connectivity between endpoints, EPGs, or an application within the fabric. A user reporting application may be experiencing slowness, or atomic counters may be needed for monitoring any traffic loss between two endpoints. One capability provided by atomic counters is the ability to place a trouble ticket into a proactive monitoring mode, for example when the problem is intermittent, and not necessarily happening at the time the operator is actively working the ticket.

Atomic counters can help detect packet loss in the fabric and allow the quick isolation of the source of connectivity issues. Atomic counters require NTP to be enabled on the fabric.

Leaf-to-leaf (TEP to TEP) atomic counters can provide the following:

- Counts of drops, admits, and excess packets
- Short-term data collection such as the last 30 seconds, and long-term data collection such as 5 minutes, 15 minutes, or more
- A breakdown of per-spine traffic (available when the number of TEPs, leaf or VPC, is less than 64)
- Ongoing monitoring

Leaf-to-leaf (TEP to TEP) atomic counters are cumulative and cannot be cleared. However, because 30 second atomic counters reset at 30 second intervals, they can be used to isolate intermittent or recurring problems.

Tenant atomic counters can provide the following:

- Application-specific counters for traffic across the fabric, including drops, admits, and excess packets

- Modes include the following:
- Endpoint to endpoint MAC address, or endpoint to endpoint IP address. Note that a single target endpoint could have multiple IP addresses associated with it.
- EPG to EPG with optional drill down
- EPG to endpoint
- EPG to * (any)
- Endpoint to external IP address

Atomic counters track the amount packets of between the two endpoints and use this as a measurement. They do not take into account drops or error counters in a hardware level.

Dropped packets are calculated when there are less packets received by the destination than transmitted by the source.

Excess packets are calculated when there are more packets received by the destination than transmitted by the source.

Configuring Atomic Counters

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name* > Troubleshooting Policies > Atomic Counter Policy**.
4. Choose a policy (traffic topology). Traffic can be measured between a combination of endpoints, endpoint groups, external interfaces, and IP addresses.
5. In the Work pane, choose **Actions > Add *Policy_Name* Policy**.
6. In the Add Policy dialog box, perform the following actions:
   a. In the **Name** field, enter a name for the policy.
   b. Choose or enter the identifying information for the traffic source. The required identifying information differs depending on the type of source (endpoint, endpoint group, external interface, or IP address).
   c. Choose or enter the identifying information for the traffic destination.

d. Optional: (Optional) In the **Filters** table, click **+** to specify filtering of the traffic to be counted. In the resulting **Create Atomic Counter Filter** dialog box, specify filtering by the IP protocol number (TCP=6, for example) and by source and destination IP port numbers.

e. Click **Submit** to save the atomic counter policy.

7 In the Navigation pane, under the selected topology, choose the new atomic counter policy. The policy configuration is displayed in the Work pane.

8 In the Work pane, choose the **Operational** tab and choose the **Traffic** subtab to view the atomic counter statistics.

Traffic Map

Low performance and congestion can be identified by use of Traffic maps.

Traffic maps make use of atomic counters to continuously monitor and display traffic between leaf switches to help with quick debugging and isolation of application connectivity issues.

Configuring Traffic Map

1 On the menu bar, choose **Fabric > Fabric Policies**.

2 In the Navigation pane, choose **Troubleshooting Policies > Traffic Map**.

3 Set the drop-down menu options to view the source Leaf to destination Leaf traffic paths.

- Last interval and cumulative
- Sent, received, dropped and excess
- All spines and a specific spine switch

The percentage is shown in relative terms to all traffic by Source or received by Destination.

4 Clicking on a cell opens a table with all data for all trails and links.

Enhanced Troubleshooting Wizard

When troubleshooting connectivity between endpoints within the fabric, the Enhanced Troubleshooting Wizard may be used to quickly identify connectivity issues. The Enhanced Troubleshooting Wizard provides a single location that includes several commonly used tools and outputs required for troubleshooting end point connectivity.

1 In the menu bar, click the wrench icon to launch the enhanced troubleshooting wizard.

2 In the Create SPAN Destination Group dialog box, perform the following actions:
   a. In the Name field, enter a name for the session.
   b. Optional: If it is an external IP address, click the check box.
   c. In the Source field, enter the MAC or IP address for the source endpoint, and click search.
   d. In the Destination field, enter the MAC or IP address for the destination endpoint, and click search.
   e. Select the Time Window duration for the debug. Optional: Generate a Report to download the results.
   f. Click Start.

3 In the next screen, you can click the following items for more information:
   a. Show the faults in the path between the selected EPs, highlighting the affected component.
   b. Run traceroute between EPs.
   c. Show relevant statistics for those EPs.
   d. Show any related recent changes in the path between EPs.

e. Configure Atomic Counters between EPs.
f. Configure SPAN between EPs
g. Show the configured contracts between EPs.

IPing

IPing is used to test and validate connectivity within the from leaf node to endpoints within the fabric, taking into account the private network. IPing is a troubleshooting tool for network users similar to the PING command.

Using IPing

iping [**-V** *vrf*] [**-c** *count*] [**-i** *wait*] [**-p** *pattern*] [**-s** *packetsize*] [**-t** *timeout*] *host*

Syntax Description

iping	Gives you information about reachability on all the paths across the network.
-V vrf	The Virtual Routing and Forwarding (VRF) instance (private network) from which to source the ping message.
-c count	Number of ping packets that are sent to the destination address. The default is 5.
-i wait	The time interval between sending of ping packets.
-p pattern	The data pattern of the ping payload. Different data patterns are used to troubleshoot framing errors and clocking problems on serial lines. The default is [0xABCD].
-s packetsize	Size of the ping packet (in bytes).
-t timeout	Timeout interval. The ping is declared successful only if the ECHO REPLY packet is received before this time interval.
host	The IP address or host name of the destination EP.

Examples

```
pod1-leaf1# iping -V overlay-1 10.0.59.154
```

```
PING 10.0.59.154 (10.0.59.154): 56 data bytes
```

```
64 bytes from 10.0.59.154: icmp_seq=0 ttl=55 time=0.254 ms
64 bytes from 10.0.59.154: icmp_seq=1 ttl=55 time=0.256 ms
64 bytes from 10.0.59.154: icmp_seq=2 ttl=55 time=0.245 ms
```

```
64 bytes from 10.0.59.154: icmp_seq=3 ttl=55 time=0.241 ms
64 bytes from 10.0.59.154: icmp_seq=4 ttl=55 time=0.23 ms
  --- 10.0.59.154 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.23/0.245/0.256 ms
```

Audit Logs

At times it may be required to view changes which have taken place in the fabric. An outage reported on a host or application in the fabric may need to be tracked, or data pulled for an audit requirement.

Audit logs are records of who made a change, when the change was made, and a description of the action. Audit logs also record when users logon and logoff.

Audit logs can be found in several places within the GUI, filtered to show only those events relevant to the current GUI context. Wherever a History tab appears in the GUI Work pane, the audit log can be viewed. This procedure shows how to view tenant events as an example.

Viewing Audit Logs

Procedure (example for viewing audit log on a tenant)

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose **Common**.
3. In the Navigation pane, choose **Common**.
4. In the Work pane, choose the **History** tab.
5. Under the **History** tab, choose the **Events** subtab to view the event log.
6. Under the **History** tab, choose the **Audit Log** subtab to view the audit log.
7. Double-click a log entry to view additional details about the event.

Reactive Monitoring Use Cases

This chapter will show some examples of how ACME's operations teams can use their ACI monitoring tools to react to a few common possible scenarios.

Note: these examples assume that basic low-level investigation has been done and the issue has been isolated to an issue with traffic flows across the fabric. Cables and connectivity have been verified, hosts are up, VMs are running, processes are running, memory and CPU utilization has been checked, etc.

Loss of Connectivity to Endpoint

The ACME application owner has reported that two servers have lost connectivity. These two End Points (EPs) belong to two different End Point Groups (EPGs), within the same subnet, bridge domain and Tenant, and each EP is connected to different leaf switches. The bridge domain has unicast routing enabled, therefore the default gateway for both EPs exist in the leaf switches.

The following steps troubleshoot this situation:

1. Check that the EPs have been learned by the leaf switches.
2. Verify that the required contracts are in place between the EPs.

Check that the EPs have been learned by the leaf switches

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name*** **>** **Application Profiles >** ***App_Profile_Name*** **>** **Application EPGs >** ***EPG_Name***.
4. In the Work pane, choose the **Operational** tab and verify that the endpoint is present.
5. Repeat this procedure for the destination EPG.

Verify the required contracts are in place between the EPs

1. On the menu bar, choose **Tenants > ALL TENANTS**.
2. In the Work pane, choose the tenant.
3. In the Navigation pane, choose ***Tenant_Name*** **>** **Application Profiles >** ***App_Profile_Name*** **>** **Application EPGs >** ***EPG_Name***.
4. In the Work pane, choose the **Operational** tab.
5. Choose the **Contracts** subtab.
6. Check for a relationship between the source EPG and destination EPG, noting the direction of the arrows.
7. Click on the contract to verify the contents of the contract. This displays the filters that are present within that contract.
8. Inspect the contents of each filter by examining the contract under the **Security Policies** folder and verifying that each filter contains the appropriate filter entries.
9. If the endpoints are discovered within each EPG and the contract relationships look correct, examine the troubleshooting policies. A good starting place is the **Enhanced Troubleshooting Wizard**.
10. Alternate techniques are available to validate communications between endpoints within the fabric. One method could be to use endpoint-to-endpoint **traceroute** to show if there are paths available between those endpoints.
    a. Another option inside the fabric could be to utilize the iPing tool to verify connectivity between the default gateway and the endpoints. Each leaf has an SVI used as default gateway. If this test is successful, the connectivity between endpoints and leaf switches is not the problem. To check the connectivity to the remote end, use iPing from each leaf to the remote endpoint using the default gateway as source.
    b. In the event that these things seem valid, it then might be necessary to use SPAN to verify where the traffic is entering and leaving the fabric, and make sure frames have the right format.

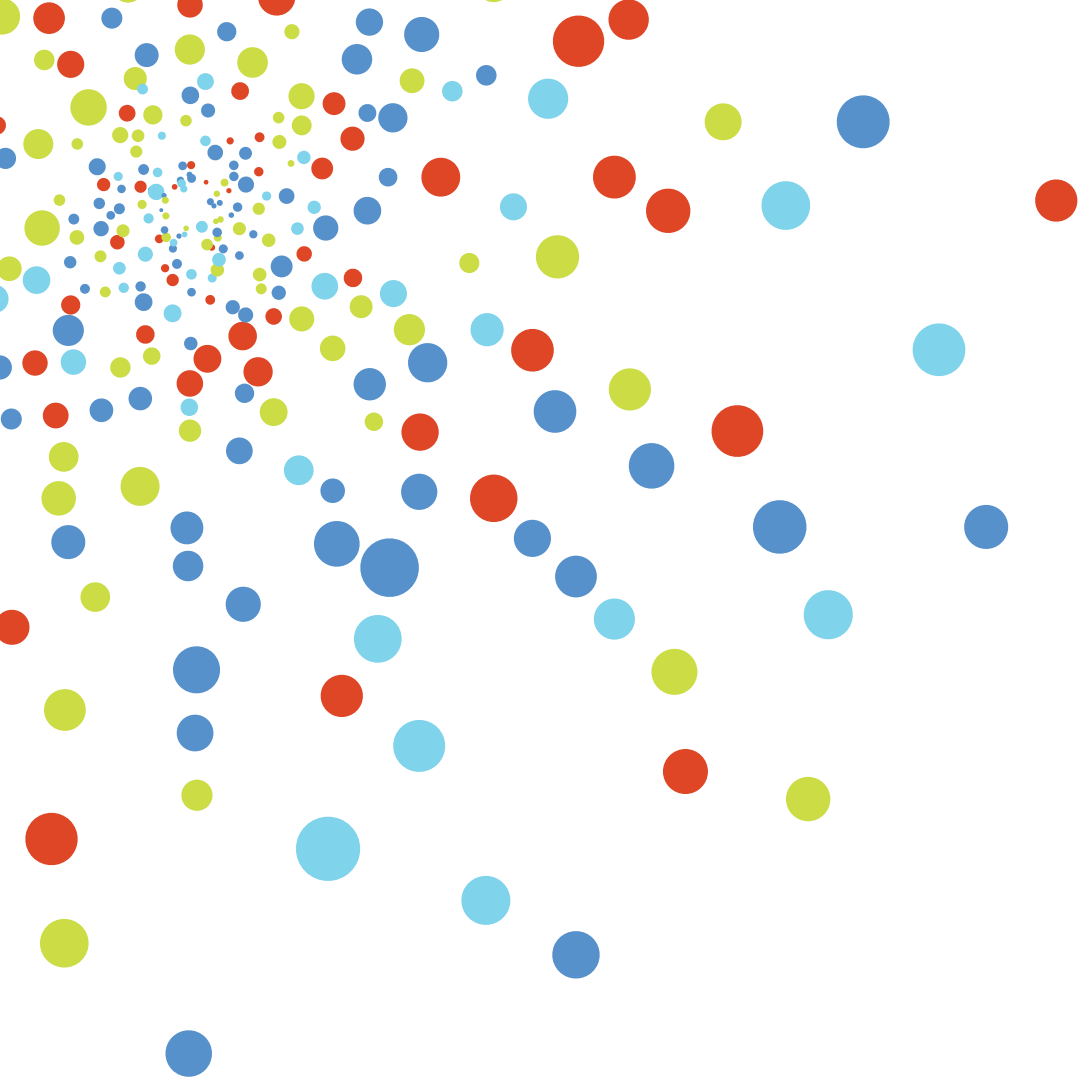
Users Report that an Application Running in the Fabric is Slow

ACME test users report slow response of the web servers running in the fabric. This performance issue could be caused due to latency, packet drops, or intermittent connectivity loss. In this case, the end-user is trying to access the web portal from a test VM. The VM and the web portal belong to a different EPGs in the same bridge domain and tenant.

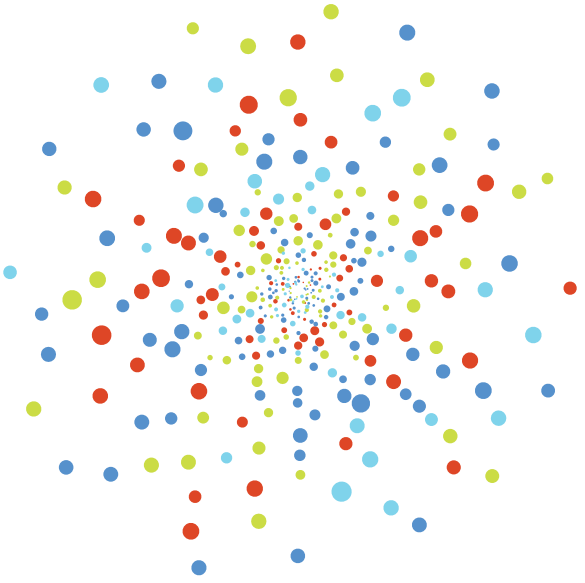
First the operations staff should make sure end points are learned by the leaf switches in a consistent way so that the EPs are visible in the operational tab under the Tenant-> Application Profile-> EPGs-> Operational Tab

Once they verify endpoints have been learned, they can use EP-to-EP Traceroute to show the path validity. Atomic counters can also be deployed to check if there are any drops or irregularities between the defined devices. Looking through the Traffic Map can also show an at-a-glance view in the fabric and highlight any possible hotspots or congestion in certain elements of the fabric. Check specific counters for CRC errors on the interfaces used to transmit EP specific traffic.

If everything in the fabric looks fine, SPAN may be used to verify where the traffic is entering and leaving the fabric and make sure frames have the right format. Corrupted frames could cause drops in different points, from the EPs on a OS level to the switches.



Scripting



Section Content

- Leveraging Network Programmability
  - Reference to Object Model
  - Programmatic Interfaces
  - REST
  - Read Operations
  - Write Operations
  - Authentication
  - Filters
- API Inspector
- Development Techniques
- POSTman
  - Installation
    - Collections
  - Build Login request
  - Make Query to APIC
  - Make Configuration Change in APIC
  - Use API Inspector for Query Guidance
- Cobra SDK and Arya
  - Establish Session
  - Work with Objects
  - Cisco APIC REST to Python Adapter

- ACI Toolkit
  - ACI Toolkit Applications
    - Endpoint Tracker
    - ACI Lint

- GitHub
  - Source Control
  - GitHub
  - " It's on github"

Leveraging Network Programmability

The industrial revolution modernized the techniques used to manufacture goods, going from hand production methods to mechanized manufacturing. This movement from manual to automated operations changed human productivity, allowing people to free themselves from repetitive tasks that could be more easily accomplished by a machine. The associated decrease in costs, increase in speed and increased quality allowed for more work to be done for less money in less time, yielding a higher quality product. Programmability promises to offer the same outcome for networks as the industrial revolution did for goods.

The inevitable move toward automation in the IT industry has provided people and businesses a faster way to achieve their desired goals, a more cost-effective way to rapidly provision infrastructure in a timely fashion according to demand, and yielded more consistency in the configured results. ACI is able to take advantage of all of these benefits by completely exposing all of the native functionality in programmable ways, using common tools and languages to provide network engineers, developers and even novices an approachable path toward automation. Though ACME is just getting started with true DevOps in their IT organization, they realize that these benefits will allow them to keep up with the pace of business.

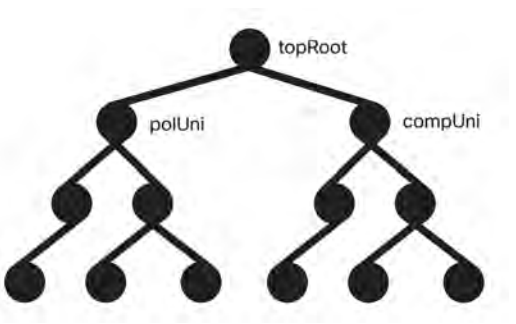
Given the comprehensiveness of the programmability features available on ACI, everyone can benefit. ACME's network engineering and design teams can benefit from the quick time to provision large configurations, and the consistency provided by the ability to automate all of the moving parts. Their operations teams can utilize the plethora of information contained within the APIC to streamline their processes, gather better metrics and correlate events more accurately, yielding faster time to resolution and higher customer satisfaction.

ACI and Scripting

The goals for network programmability are clear, however the methods by which these goals may be realized have been more difficult to grasp. Traditional networking devices provide output that is meant for visual consumption by people, and configurations are driven using text input that is simpler for a person to type, however these goals stand in contrast to an automation-driven approach. Machines are able to more easily process data that is provided in some structured form. Structured data that may not be visually appealing can be rapidly parsed, and also can easily represent the full detail that a comprehensive object-oriented configuration model may represent.

ACI uses an advanced object model that represents network configuration with application-based semantics which can be consumed and posted against using a well documented REST API. In addition to providing this interface into the object model, ACI also provides a number of access methods to read and manipulate this data, at a variety of levels that will cater to the level of comfort the user has with programming, all of which use open standards and open source.

Reference to Object Model



Representation of the top levels of the Object Model

While a comprehensive overview of the Object Model is outside of this book, from a programmability perspective it is important to note that every aspect of ACI functional-

ity is encompassed within the object model. This means that all of the configuration that can be made on the fabric, can be made programmatically using the REST API. This includes internal fabric networking, external networking, virtualization integration, compute integration, and all other facets of the product.

This data is stored within the Management Information Tree, with every piece of the model represented as a programmatic object with properties, identity, and consistency rules that are enforced. This ensures that the configured state of the model will never get out of hand with stale nodes or entries, and every aspect can be inspected, manipulated, and made to cater for the user's needs.

Programmatic Interfaces

APIC is very flexible in terms of how it can accept configuration and provide administrative and operable states, as well as extending that configuration into subordinate components. There are two primary categories of interfaces that facilitate these functions: the northbound REST API and the southbound programmatic interfaces.

The northbound REST API is responsible for accepting configuration, as well as providing access to management functions for the controller. This interface is a crucial component for the GUI and CLI, and also provides a touch point for automation tools, provisioning scripts and third party monitoring and management tools. The REST API is a singular entry point to the fabric for making configuration changes, and as such is a critical aspect of the architecture for being able to provide a consistent programmatic experience.

Southbound interfaces on APIC allow for the declarative model of intent to be extended beyond the fabric, into subordinate devices. This is a key aspect to the openness of the ACI fabric, in that policy can be programmed once via APIC and then pushed out to hypervisors, L4-7 devices and potentially more in the future, without the need to individually configure those devices. This southbound extension is realized through two methods: L4-7 Device Packages and OpFlex.

The L4-7 device package interface allows for ACI to apply policy to existing L4-7 devices that do not have an implicit knowledge of ACI policy. These devices can be from any vendor, so long as the device has some form of interface which is accessible via IP. The actual implementation of device packages is done via Python scripts which run on the APIC within a contained execution environment, which can reach the device through their native configuration interfaces, be that REST, CLI, SOAP or others. As a

user makes changes to service graphs or EPG policy, the device package will translate the APIC policy into API calls on the L4-7 device.

OpFlex is designed to allow a data exchange of a set of managed objects that is defined as part of an informational model. OpFlex itself does not dictate the information model, and can be used with any tree-based abstract model in which each node in the tree has a universal resource identifier (URI) associated with it. The protocol is designed to support XML and JSON (as well as the binary encoding used in some scenarios) and to use standard remote procedure call (RPC) mechanisms such as JSON-RPC over TCP. In ACI, OpFlex is currently used to extend policy to the Application Virtual Switch as well as extend Group Based Policy into OpenStack.

REST

The Cisco APIC REST API is a programmatic interface for Cisco APIC that uses REST architecture. The API accepts and returns HTTP (not enabled by default) or HTTPS messages that contain JavaScript Object Notation (JSON) or Extensible Markup Language (XML) documents. You can use any programming language to generate the messages and the JSON or XML documents that contain the API methods or MO descriptions.

The REST API is the interface into the MIT and allows manipulation of the object model state. The same REST interface is used by the Cisco APIC command-line interface (CLI), GUI, and SDK, so that whenever information is displayed, it is read through the REST API, and when configuration changes are made, they are written through the REST API. The REST API also provides an interface through which other information can be retrieved, including statistics, faults, and audit events, and it even provides a means of subscribing to push-based event notification, so that when a change occurs in the MIT, an event can be sent through a web socket.

Standard REST methods are supported on the API, which includes POST, GET, and DELETE operations through HTTP. The POST and DELETE methods are idempotent, meaning that there is no additional effect if they are called more than once with the same input parameters. The GET method is nullipotent, meaning that it can be called zero or more times without making any changes (or that it is a read-only operation).

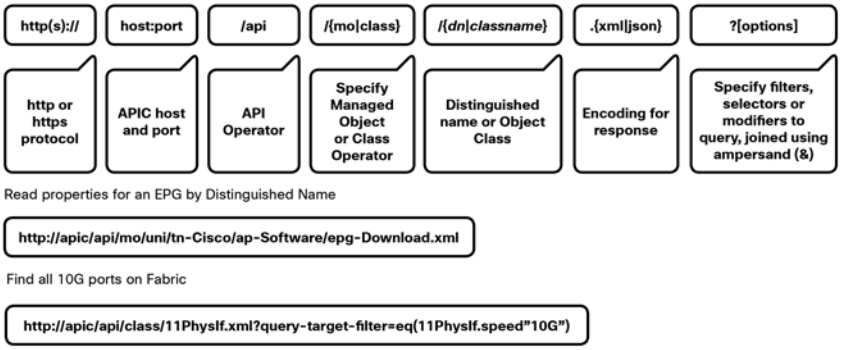
Payloads to and from the REST interface can be encapsulated through either XML or JSON encoding. In the case of XML, the encoding operation is simple: the element tag is the name of the package and class, and any properties of that object are specified as attributes of that element. Containment is defined by creating child elements.

For JSON, encoding requires definition of certain entities to reflect the tree-based hierarchy; however, the definition is repeated at all levels of the tree, so it is fairly simple to implement after it is initially understood.

- All objects are described as JSON dictionaries, in which the key is the name of the package and class, and the value is another nested dictionary with two keys: attribute and children.
- The attribute key contains a further nested dictionary describing key-value pairs that define attributes on the object.
- The children key contains a list that defines all the child objects. The children in this list are dictionaries containing any nested objects, which are defined as described here.

Read Operations

After the object payloads are properly encoded as XML or JSON, they can be used in create, read, update, or delete operations on the REST API. The following diagram shows the syntax for a read operation from the REST API.



REST syntax

Because the REST API is HTTP based, defining the universal resource identifier (URI) to access a certain resource type is important. The first two sections of the request URI simply define the protocol and access details of Cisco APIC. Next in the request URI is the literal string **/api**, indicating that the API will be invoked. Generally, read operations are for an object or class, as discussed earlier, so the next part of the URI specifies

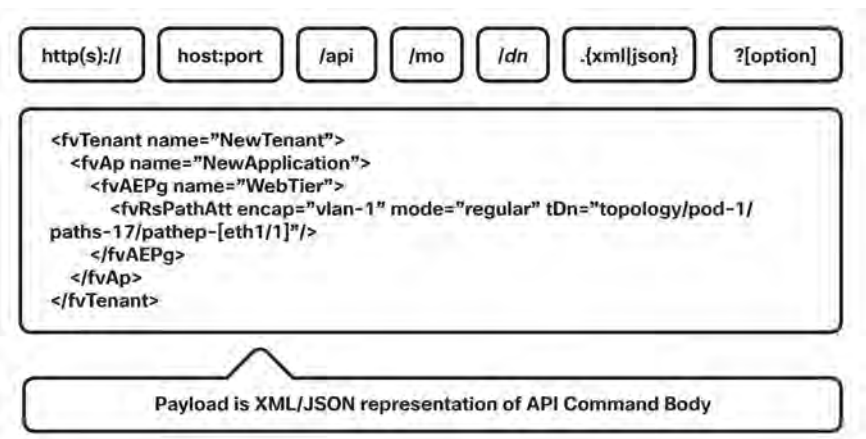
whether the operation will be for an MO or class. The next component defines either the fully qualified Dn being queried for object-based queries, or the package and class name for class-based queries. The final mandatory part of the request URI is the encoding format: either .xml or .json. This is the only method by which the payload format is defined (Cisco APIC ignores Content-Type and other headers).

Write Operations

Create and update operations in the REST API are both implemented using the POST method, so that if an object does not already exist, it will be created, and if it does already exist, it will be updated to reflect any changes between its existing state and desired state.

Both create and update operations can contain complex object hierarchies, so that a complete tree can be defined in a single command so long as all objects are within the same context root and are under the 1MB limit for data payloads for the REST API. This limit is in place to guarantee performance and protect the system under high load.

The context root helps define a method by which Cisco APIC distributes information to multiple controllers and helps ensure consistency. For the most part, the configuration should be transparent to the user, though very large configurations may need to be broken into smaller pieces if they result in a distributed transaction.



REST Payload

Create and update operations use the same syntax as read operations, except that they always are targeted at an object level, because you cannot make changes to every object of a specific class (nor would you want to). The create or update operation should target a specific managed object, so the literal string **/mo** indicates that the Dn of the managed object will be provided, followed next by the actual Dn. Filter strings can be applied to POST operations; if you want to retrieve the results of your POST operation in the response, for example, you can pass the **rsp-subtree=modified** query string to indicate that you want the response to include any objects that have been modified by your POST operation.

The payload of the POST operation will contain the XML or JSON encoded data representing the managed object the defines the Cisco API command body.

Authentication

REST API username- and password-based authentication uses a special subset of request URIs, including **aaaLogin**, **aaaLogout**, and **aaaRefresh** as the Dn targets of a POST operation. Their payloads contain a simple XML or JSON payload containing the MO representation of an **aaaUser** object with the attribute name and **pwd** defining the username and password: for example, **<aaaUser name='admin' pwd='insieme'/>**. The response to the POST operation will contain an authentication token as both a Set-Cookie header and an attribute to the **aaaLogin** object in the response named token, for which the XPath is **/imdata/aaaLogin/@token** if the encoding is XML. Subsequent operations on the REST API can use this token value as a cookie named **APIC-cookie** to authenticate future requests.

Filters

The REST API supports a wide range of flexible filters, useful for narrowing the scope of your search to allow information to be located more quickly. The filters themselves are appended as query URI options, starting with a question mark (?) and concatenated with an ampersand (&). Multiple conditions can be joined together to form complex filters.

The following query filters are available:

Filter type	Syntax	Cobra Query Property	Description
query-target	{self | children | subtree}	AbstractQuery.queryTarget	Define the scope of query
target-subtree-class	<class name>	AbstractQuery.classFilter	Respond only elements including specified class
query-target-filter	<filter expressions>	AbstractQuery.propFilter	Respond only elements matching conditions
rsp-subtree	{no | children | full}	AbstractQuery.subtree	specifies child object level included in the response
rsp-subtree-class	<class name>	AbstractQuery.subtreeClassFilter	Respond only specified classes
rsp-subtree-filter	<filter expressions>	AbstractQuery.subtreePropFilter	Respond only classes matching conditions
rsp-subtree-include	{faults | health :stats : ...}	AbstractQuery.subtreeInclude	Request additional objects
order-by	<classname.property>| {asc | desc}	*NotImplemented*	Sort the response based on the property values

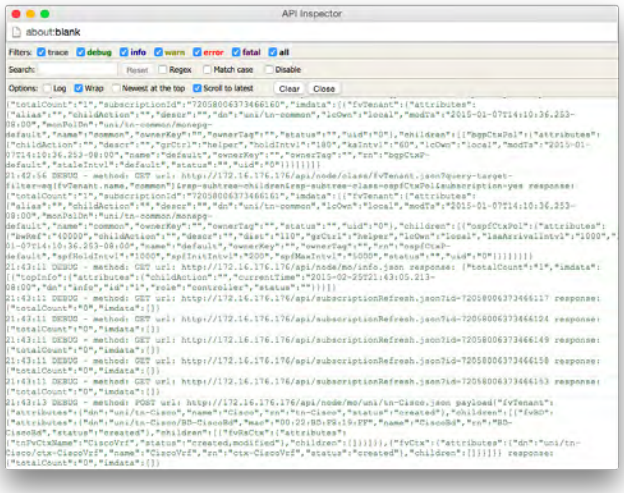
Query filters

API Inspector

All operations that are performed in the GUI invoke REST calls to fetch and commit the information being accessed. The API Inspector further simplifies the process of examining what is taking place on the REST interface as the GUI is navigated by displaying in real time the URIs and payloads. When a new configuration is committed, the API Inspector displays the resulting POST requests, and when information is displayed on the GUI, the GET request is displayed.

To get started with the API Inspector, it can be accessed from the Account menu, visible at the top right of the Cisco APIC GUI. Click Welcome, <username> and then choose the Show API Inspector option

After the API Inspector is brought up, time stamps will appear along with the REST method, URIs, and payloads. There may also be occasional updates in the list as the GUI refreshes subscriptions to data being shown on the screen.



API Inspector

From the output above it can see that the last logged item has a POST request with the JSON payload containing a tenant named **Cisco** and some attributes defined on that object:

url: http://172.16.176.176/api/node/mo/uni/tn-Cisco.json


```
{
 "fvTenant": {
   "attributes": {
     "name": "Cisco",
     "status": "created"
   },
   "children": [
     {
       "fvBD": {
         "attributes": {
           "mac": "00:22:BD:F8:19:FF",
           "name": "CiscoBd",
           "status": "created"
         },
         "children": [
           {
             "fvRsCtx": {
               "attributes": {
                 "tnFvCtxName": "CiscoVrf",
                 "status": "created,modified"
               },
               "children": []
             }
           }
         ]
       }
     },
     {
       "fvCtx": {
         "attributes": {
           "name": "CiscoVrf",
           "status": "created"
```

```
            },
            "children": []
          }
        }
      ]
    }
  }
```

Development Techniques

ACI has a number of methods for developing code that can be used by engineers who have varying levels of comfort with programming or interacting with programmatic interfaces.

The most basic and straight-forward technique involves simply taking information gleaned by the API inspector, Visore, or by saving XML/JSON directly from the GUI, and using common freely available tools, such as POSTman, to send this information back to the REST API.

A step up from this method enables users to use common terminology and well understood networking constructs, coupling these with the power and flexibility of the ACI policy language and the popular Python programming language to configure ACI in a programmatic fashion. ACI Toolkit is a utility developed in open-source that exposes the most common ACI building blocks, to enable users to rapidly create tenants, application profiles, EPGs and the associated concepts to connect those to physical infrastructure. The streamlined interface provided makes it very quick to adopt and allows users to begin to quickly develop their applications.

The most powerful of the development tools available is the Cobra SDK. With a complete representation of the ACI object model available, comprehensive data validation, and extensive support for querying and filtering, Cobra ensures that the complete ACI experience is available to developers and users alike.

POSTman

POSTman is an open source extension for the Chrome web browser, which provides REST client functionality in an easy-to-use package. POSTman can be used to interact with the APIC REST interface, to both send and receive data which may represent configuration, actions, policy and operational state data. For an individual unfamiliar with the structure of REST, it is very simple to utilize the API Inspector to view what the underlying calls being made to the GUI are for certain operations, capture those, and then use POSTman to replay those operations. Furthermore POSTman allows for the requests to be modified: GUI operations can be made once, attributes changed in the captured data and then sent back to the REST API to make the modifications.

Installation

To get started with POSTman, the first step is to download the plugin for the Chrome web browser, which is available at www.getpostman.com. Once the plugin is installed, it can be accessed using the Chrome App launcher.

Initially the user will be presented with an interface that has two primary sections: the sidebar on the left and the request constructor on the right. Using the sidebar, the user can switch between the history of REST requests sent by POSTman, as well as Collections of requests that contain common tasks.

Collections

A useful post to create in a collection is a basic Login operation. In order to do this, the user should first click into the Collections tab in the sidebar. Within the sidebar, a small folder with a plus (+) sign will become visible, which should then be clicked, at which point a popup will appear prompting the user to give a name to the collection. For this example, the collection can be named “APIC”, after which the Create button should be clicked.

Build Login request

Now a new request can be built. In the request constructor, where “Enter request URL here” is shown, the following request URI should be entered, substituting APICIPADDRESS with the IP of the APIC: https://<APIC-IP>/api/mo/aaaLogin.xml

This request URI will call the Login method in the REST API. Since a Login will require posting data, the HTTP method should be changed, which can be done by clicking the dropdown list to the right of the request URL. By default it will be a GET request, but POST will need to be selected from the drop down list.

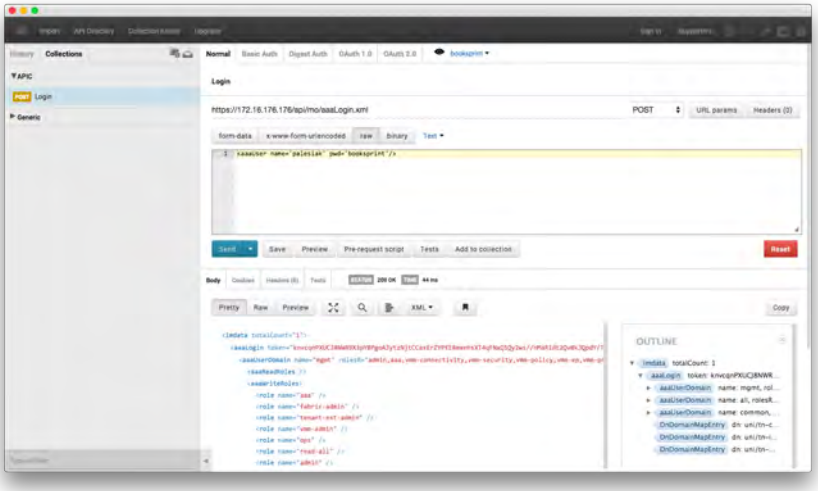
With the POST method selected, it is now possible to provide the REST payload. Given that the data will be sent via REST, the “raw” Request body selector should be picked.

Now the payload for the request can be entered, which will be the simple XML containing the username and password that will be used for authentication. Note that the URL is https, meaning that it will be encrypted between the web browser and the APIC, so no data is being transmitted in clear text. The following request body should be entered, substituting the correct username and password in place of USERNAME and PASSWORD: `<aaaUser name='USERNAME' pwd='PASSWORD'/>`

With this request built, it is now possible to Send the request, but since this will be a commonly used method, the request should be added to a collection. This can be accomplished by clicking the “Add to collection” button beneath the request body. Select the “APIC” collection from the existing collection list, and change the Request name to “Login” and then click “Add to collection”.

By adding the request to a collection it can later be quickly accessed to establish a login session with APIC as needed.

After completing the above steps, the request is ready to be sent. Click the “Send” button in the request constructor, and the REST API will return the XML representing a login session with the APIC. The following will be visible in the POSTman GUI:



Login request in POSTman

Make Query to APIC

The next request that will be built is one that queries the APIC for a list of tenants on the system. First click the “Reset” button in the request constructor, and proceed with the same steps as above, except that the request URL will be shown as https://<APIC-IP>/api/class/fvTenant.xml, and the request method will be changed to GET.

Click “Add to collection” and place the request into the APIC collection, and for the name enter “Query APIC for tenants”

Now upon clicking “Send”, this request will return an XML encoded list of tenants in the response body section of the constructor pane on the right.

Make Configuration Change in APIC

Making a configuration change will use a POST request similar to logging in, however the request URL and body will contain a different set of information.

For this example, a new tenant will be created in the fabric. Click the “Reset” button in the request constructor to clear out all existing request fields, and use URL https://<APIC-IP>/api/mo/uni.xml and change the method to POST.

In the request payload, enter the following data: `<fvTenant name="Cisco"/>`

The request URL specifies that the target for this query will be the policy universe, which is where tenants live. With this target properly scoped, the data representing the tenant can be provided in the payload, in this case creating a tenant named Cisco.

Use API Inspector for Query Guidance

As discussed in the Introduction to Scripting section, API inspector can be used as a guideline for building custom REST requests. Furthering on the example in that section, where the request URL is https://<APIC-IP>/api/node/mo/uni/tn-Cisco.json and the payload is the following compacted version of JSON:

```
{"fvTenant": {"attributes": {"name": "Cisco", "status": "created"}, "children":
[{"fvBD": {"attributes": {"mac": "00:22:BD:F8:19:FF", "name": "CiscoBd", "status":
"created"}, "children": [{"fvRsCtx": {"attributes": {"tnFvCtxName": "CiscoVrf",
"status": "created,modified"}, "children": [] } } ] } }, {"fvCtx": {"attributes":
{"name": "CiscoVrf", "status": "created"}, "children": [] } } ] } }
```

It is possible to modify the request URI and payload and substitute the tenant name “Cisco” with another tenant name, to create an entirely new tenant, with the same configuration. The new request URL and JSON would be: https://<APIC-IP>/api/node/mo/uni/tn-Acme.json

```
{"fvTenant": {"attributes": {"name": "Acme", "status": "created"}, "children":
[{"fvBD": {"attributes": {"mac": "00:22:BD:F8:19:FF", "name": "AcmeBd", "status":
"created"}, "children": [{"fvRsCtx": {"attributes": {"tnFvCtxName": "AcmeVrf",
"status": "created,modified"}, "children": [] } } ] } }, {"fvCtx": {"attributes":
{"name": "AcmeVrf", "status": "created"}, "children": [] } } ] } }
```

These values can be placed into a POST request in POSTman, and after establishing a Login session using the saved Login request, the new tenant “Acme” can be created, identical to the previously created Cisco tenant, without needing to manually click through the GUI or use other manual methods.

Cobra SDK and Arya

The complete Cisco ACI Python SDK is named Cobra. It is a pure Python implementation of the API that provides native bindings for all the REST functions and also has a complete copy of the object model so that data integrity can be ensured, as well as supporting the complete set of features and functions available in ACI. Cobra provides methods for performing lookups and queries and object creation, modification, and deletion that match the REST methods used by the GUI and those that can be found using API Inspector. As a result, policy created in the GUI can be used as a programming template for rapid development.

The installation process for Cobra is straightforward, and you can use standard Python distribution utilities. Cobra is distributed on the APIC as an .egg file and can be installed using easy_install, and is also available on github at http://github.com/datacenter/cobra.

The complete documentation for the Cobra SDK is available at http://cobra.readthedocs.org/en/latest/

Establish Session

The first step in any code that uses Cobra is establishing a login session. Cobra currently supports username- and password-based authentication, as well as certificate-based authentication. The example here uses username- and password-based authentication.

```
import cobra.mit.access
import cobra.mit.session
apicUri = 'https://10.0.0.2'
apicUser = 'username'
apicPassword = 'password'
ls = cobra.mit.session.LoginSession(apicUri, apicUser, apicPassword)
md = cobra.mit.access.MoDirectory(ls)
md.login()
```

This example provides an **MoDirectory** object named **md**, which is logged into and authenticated for Cisco APIC. If for some reason authentication fails, Cobra will display a cobra.mit.request.CommitError exception message. With the session logged in, you are ready to proceed.

Work with Objects

Use of the Cobra SDK to manipulate the MIT generally follows this workflow:

1. Identify the object to be manipulated.
2. Build a request to change attributes or add or remove children.
3. Commit the changes made to the object.

For example, if you want to create a new tenant, you must first identify where the tenant will be placed in the MIT, where in this case it will be a child of the **policy Universe** managed object (**polUniMo**):

```
import cobra.model.pol
polUniMo = cobra.model.pol.Uni('')
```

With the **polUniMo** object defined, you can create a tenant object as a child of **polUniMo**:

```
import cobra.model.fv
tenantMo = cobra.model.fv.Tenant(polUniMo, 'cisco')
```

All these operations have resulted only in the creation of Python objects. To apply the configuration, you must commit it. You can do this using an object called a **ConfigRequest**. **ConfigRequest** acts as a container for MO-based classes that fall into a single context, and they can all be committed in a single atomic POST operation.

```
import cobra.mit.request
config = cobra.mit.request.ConfigRequest()
config.addMo(tenantMo)
md.commit(config)
```

The **ConfigRequest** object is created, then the **tenantMo** object is added to the request, and then you commit the configuration through the **MoDirectory** object.

For the preceding example, the first step builds a local copy of the **polUni** object. Because it does not have any naming properties (reflected by the empty double single quotation marks), you don't need to look it up in the MIT to figure out what the full Dn for the object is; it is always known as **uni**.

If you wanted to post something deeper in the MIT, where the object has naming properties, you would need to perform a lookup for that object. For example, if you wanted to post a configuration to an existing tenant, you could query for that tenant and create objects beneath it.

```
tenantMo = md.lookupByClass('fvTenant', propFilter='eq(fvTenant.name, "cisco")')
tenantMo = tenantMo[0] if tenantMo else None
```

The resulting **tenantMo** object will be of class **cobra.model.fv.Tenant** and will contain properties such as .dn, .status, and .name, all describing the object itself. The **lookupByClass()** entry returns an array, because it can return more than one object. In this case, the command is specific and is filtering on an **fvTenant** object with a particular name. For a tenant, the name attribute is a special type of attribute called a naming attribute. The naming attribute is used to build the relative name, which must be unique in its local namespace. As a result, you can be assured that **lookupByClass** on an **fvTenant** object with a filter on the name always returns either an array of length 1 or None, meaning that nothing was found.

To entirely avoid a lookup, you can build a Dn object and make an object a child of that Dn. This method works only in cases in which the parent object already exists.

```
topDn = cobra.mit.naming.Dn.fromString('uni/tn-cisco')
fvAp = cobra.model.fv.Ap(topMo, name='AppProfile')
```

These fundamental methods for interacting with Cobra provide the building blocks necessary to create more complex workflows that can help automate network configuration, perform troubleshooting, and manage the network.

Cisco APIC REST to Python Adapter

The process of building a request can be time consuming, because you must represent the object data payload as Python code reflecting the object changes that you want to make. Because the Cobra SDK is directly modeled on the Cisco ACI object model, you should be able to generate code directly from what resides in the object model. As expected, you can do this using a tool developed by Cisco Advanced Services. The tool is the Cisco APIC REST to Python Adapter, known as Arya.

Sample REST to Python Adapter

The above figure clearly shows how the input that might come from the API Inspector, Visore, or even the output of a REST query and can then be quickly converted into Cobra SDK code, tokenized, and reused in more advanced ways.

Installation of Arya is relatively simple, and the tool has few external dependencies. To install Arya, you must have Python 2.7.5 and git installed. Use the following quick installation steps to install it and place it in your system Python.

```
git clone https://github.com/datacenter/ACI.git
cd ACI/arya
sudo python setup.py install
```

After Arya has been installed, you can take XML or JSON representing Cisco ACI modeled objects and convert it to Python code quickly. For example, enter:

```
arya.py -f /home/palesiak/simpletenant.xml
```

The entry will yield the following Python code:

```
#!/usr/bin/env python
'''
Autogenerated code using arya.py
Original Object Document Input:
<fvTenant name='bob'/>
''' raise RuntimeError('Please review the auto generated code before ' +
                    'executing the output. Some placeholders will ' +
                    'need to be changed')
# list of packages that should be imported for this code to work
import cobra.mit.access
import cobra.mit.session
import cobra.mit.request
import cobra.model.fv
import cobra.model.pol
from cobra.internal.codec.xmlcodec import toXMLStr
# log into an APIC and create a directory object
ls = cobra.mit.session.LoginSession('https://1.1.1.1', 'admin', 'password')
md = cobra.mit.access.MoDirectory(ls)
md.login()
# the top level object on which operations will be made
topMo = cobra.model.pol.Uni('')
# build the request using cobra syntax
fvTenant = cobra.model.fv.Tenant(topMo, name='bob')
# commit the generated code to APIC
print toXMLStr(topMo)
c = cobra.mit.request.ConfigRequest()
c.addMo(topMo)
md.commit(c)
```

The placeholder raising a runtime error must first be removed before this code can be executed; it is purposely put in place to help ensure that any other tokenized values that must be updated are corrected. For example, the Cisco APIC IP address, which defaults to 1.1.1.1, should be updated to reflect the actual Cisco APIC IP address. The same applies to the credentials and any other placeholders.

Note that if you provide input XML or JSON that does not have a fully qualified hierarchy, Arya may not be able to identify it through heuristics. In this case, a placeholder will be populated with the text **REPLACEME**, which you will need to replace with the correct Dn. You can find this Dn by querying for the object in Visore or inspecting the request URI for the object shown in the API Inspector.

ACI Toolkit

The complete ACI object model contains many entities, which may be daunting for a user being first introduced to network programmability. The acitoolkit makes available a simplified subset of the model that can act as an introduction to the concepts in ACI, and give users a way to quickly bring up common tasks and workflows. In addition, a number of applications have been built on top of ACI toolkit.

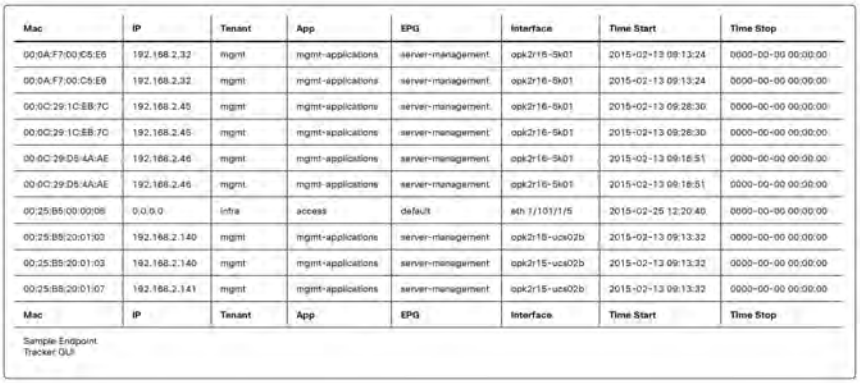
The complete documentation for acitoolkit is available at http://datacenter.github.io/acitoolkit/

ACI Toolkit Applications

Endpoint Tracker

The endpoint tracker application creates a subscription to the endpoint class (fvCEp) and populates a MySQL database with pertinent details about each endpoint present on the fabric (for example servers, firewalls, load balancers, and other devices). Installing MySQL is outside the scope of this book, so we will assume you have access to create a new database on a MySQL server. The endpoint tracker application has two primary components that are both python scripts.

- **aci-endpoint-tracker.py**—This script creates the subscription to the endpoint class and populates the MySQL database
- **aci-endpoint-tracker-gui.py**—This script creates a web interface that provides a way to present the contents of the database to the operator. A sample is shown below:



Mac	IP	Tenant	App	EPG	Interface	Time Start	Time Stop
00:0A:F7:00:C5:E6	192.168.2.32	mgmt	mgmt-applications	server-management	epk2r16-5k01	2015-02-13 09:13:24	0000-00-00 00:00:00
00:0A:F7:00:C6:E8	192.168.2.32	mgmt	mgmt-applications	server-management	epk2r16-5k01	2015-02-13 09:13:24	0000-00-00 00:00:00
00:0C:29:1C:EB:7C	192.168.2.45	mgmt	mgmt-applications	server-management	epk2r16-5k01	2015-02-13 09:28:30	0000-00-00 00:00:00
00:0C:29:1C:EB:7C	192.168.2.45	mgmt	mgmt-applications	server-management	epk2r16-5k01	2015-02-13 09:28:30	0000-00-00 00:00:00
00:0C:29:D5:4A:AE	192.168.2.46	mgmt	mgmt-applications	server-management	epk2r16-5k01	2015-02-13 09:16:51	0000-00-00 00:00:00
00:0C:29:D5:4A:AE	192.168.2.46	mgmt	mgmt-applications	server-management	epk2r16-5k01	2015-02-13 09:16:51	0000-00-00 00:00:00
00:25:B5:00:00:08	0.0.0.0	infra	access	default	eth 1/101/1/5	2015-02-25 12:20:40	0000-00-00 00:00:00
00:25:B5:20:01:03	192.168.2.140	mgmt	mgmt-applications	server-management	epk2r16-ucs02b	2015-02-13 09:13:32	0000-00-00 00:00:00
00:25:B5:20:01:03	192.168.2.140	mgmt	mgmt-applications	server-management	epk2r15-ucs02b	2015-02-13 09:13:32	0000-00-00 00:00:00
00:25:B5:20:01:07	192.168.2.141	mgmt	mgmt-applications	server-management	epk2r15-ucs02b	2015-02-13 09:13:32	0000-00-00 00:00:00
Mac	IP	Tenant	App	EPG	Interface	Time Start	Time Stop

Sample Endpoint Tracker GUI

To launch Endpoint Tracker run the following python scripts. The first script, aci-endpoint-tracker.py, will actually connect to the APIC and populate the database. The second script enables the content to be viewed in an understandable web UI.

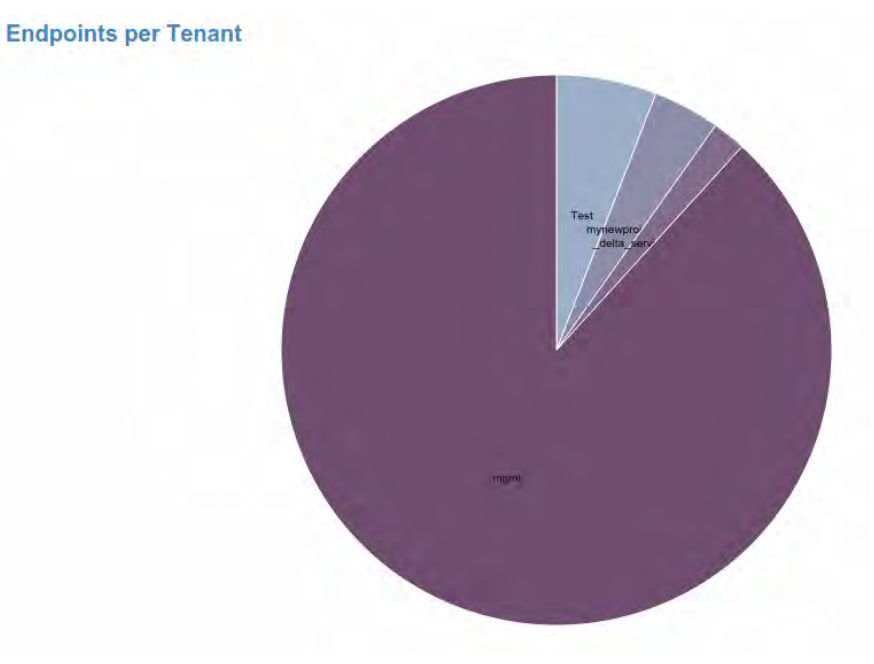
```
user@linuxhost:~/acitoolkit/applications/endpointtracker$ ./aci-endpoint-tracker.py
MySQL IP address: 127.0.0.1
MySQL login username: root
MySQL Password:
user@linuxhost::~/acitoolkit/applications/endpointtracker$ python aci-endpoint-
tracker-gui.py
MySQL IP address: 127.0.0.1
MySQL login username: root
MySQL Password:
 * Running on http://127.0.0.1:5000/
 * Restarting with reloader
```

After running those python scripts you can now bring up a browser and go the Web UI. Using the ACI Endpoint Tracker is simply a matter of inputting an IP or MAC address into the search field, and the table is filtered accordingly. In the example below, the IP address 192.168.5.20 has been input into the search field, and the matching results are displayed.

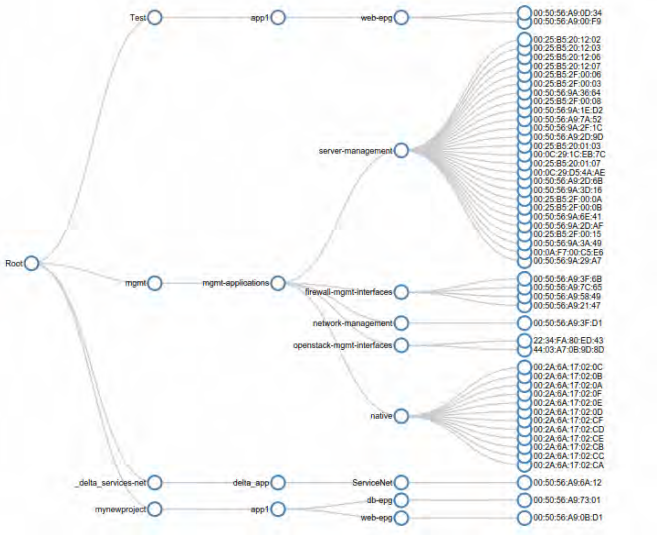
Mac	IP	Tenant	App	EPG	Interface	Time Start	Time Stop
00:50:56:A9:73:01	192.168.5.20	mynewproject	app1	db-epg	topology/pod-1/path-grp-[192.168.1.221]	2015-02-19 12:47:30	0000-00-00 00:00:00
Mac	IP	Tenant	App	EPG	Interface	Time Start	Time Stop

One more interesting usage of the endpoint tracker applications is a series of visualizations that can represent how various endpoints are mapped to other fabric constructs including Tenants, Applications, and EPGs.

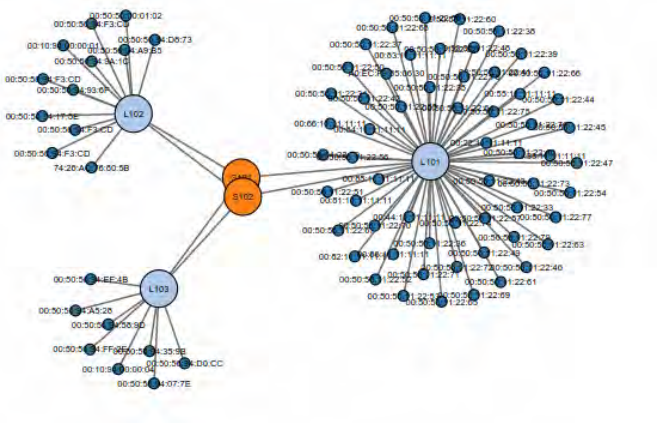
Some sample screenshots are shown below. These are representations of where end points are within the ACI fabric and how they relate to or depend on other objects in the environment.



Pie chart view of endpoint distribution



Tree view of endpoint relationships



Force diagram of endpoint location

ACI Lint

In computer programming, Lint is a term that refers to identifying discrepancies, or simple debug tool for common errors. In the sense that ACI provides infrastructure as code, it is appropriate for ACI to also have a Lint application. ACI Toolkit provides just that. ACI Lint is an application that checks and notifies an operator of misconfiguration errors in two primary capacities:

Security Issues - supports the ability to tag EPGs as either secure or insecure, and then runs a validation that contracts are not used to cross security boundaries.

Configuration Issues - checks for common configuration errors and reports them to the user.

A sample output is provided here for reference:

```
 user@linuxhost:~/acitoolkit/applications/lint$ ./acilint.py
 Getting configuration from APIC....
 Processing configuration....
 Critical 001: EPG 'default' in tenant 'infra' app 'access' is not assigned security
clearance
 Critical 001: EPG 'x' in tenant 'common' app 'default' is not assigned security
clearance
 Warning 001: Tenant 'Cisco' has no Application Profile.
 Warning 001: Tenant 'Books' has no Application Profile.
 Warning 001: Tenant '3tierapp' has no Application Profile.
 Warning 001: Tenant 'mgmt' has no Application Profile.
 Warning 002: Tenant 'Books' has no Context.
 Warning 002: Tenant '3tierapp' has no Context.
 Warning 004: Context 'oob' in Tenant 'mgmt' has no BridgeDomains.
 Warning 005: BridgeDomain 'CiscoBd' in Tenant 'Cisco' has no EPGs.
 Warning 005: BridgeDomain 'inb' in Tenant 'mgmt' has no EPGs.
 Warning 006: Contract 'default' in Tenant 'common' is not provided at all.
 Warning 006: Contract 'WebServers' in Tenant 'Acme' is not provided at all.
 Warning 006: Contract 'External' in Tenant 'Acme' is not provided at all.
 Warning 007: Contract 'default' in Tenant 'common' is not consumed at all.
 Warning 007: Contract 'WebServers' in Tenant 'Acme' is not consumed at all.
 Warning 007: Contract 'External' in Tenant 'Acme' is not consumed at all.
 Warning 007: Contract 'outside-to-web' in Tenant 'roberbur' is not consumed at all.
```

While the ACI Toolkit provides some useful tools for an operator to immediately use, the real value is in the ability to take these examples as a starting point, and modify or extend these samples to suit your particular needs. Give it a try! Be sure to share your work back with the community!

GitHub

Source Control

Open source software has been a popular movement in IT, and has been the motivation behind many successful projects, including consumer software, web servers, databases and even entire operating systems. One of the key aspects to the success of open source is the ability for many developers around the globe to collaborate together on a single project. Previous tools like Concurrent Version Control (CVS), and Subversion (SVN) were used to allow many developers to work together, with a central server maintaining a common database of source code. While these tools have and continue to work well, there has been a slow migration away from those server-based tools to decentralized utilities, with the foremost being Git. Git was created by Linus Torvalds, the author of the popular open-source operating system Linux. Git has a number of advantages over most other source control tools: complete local repository copies, distributed architecture, and more efficient support for branches.

GitHub

GitHub is a hosting platform based around git, which provides both free and paid hosting services, that allow for individuals to collaborate with over eight-million other GitHub users on projects together. Aside from being a wrapper around git, GitHub also provides techniques for tracking issues, securing access to projects, and built-in project documentation. The combination of all of these features has made GitHub a very common place for members of the community to share code with one another, build on each other's work, and contribute their efforts back into larger projects.

What is stored on GitHub is usually source code, not limited to any specific language, however the git protocol itself supports storage and version control of any file type, so it's not uncommon for users to store documentation or other constantly changing files in git. The primary advantage is that the version control provided by git allows a user to revert a file back to any previously stored version, or alternately move forward to a newer version. Git also maintains an audit of changes that have been made to files and even has advanced support for branching versions of files to allow multiple concurrent modifications to a file to take place, and allow for them to be merged after work efforts have completed.

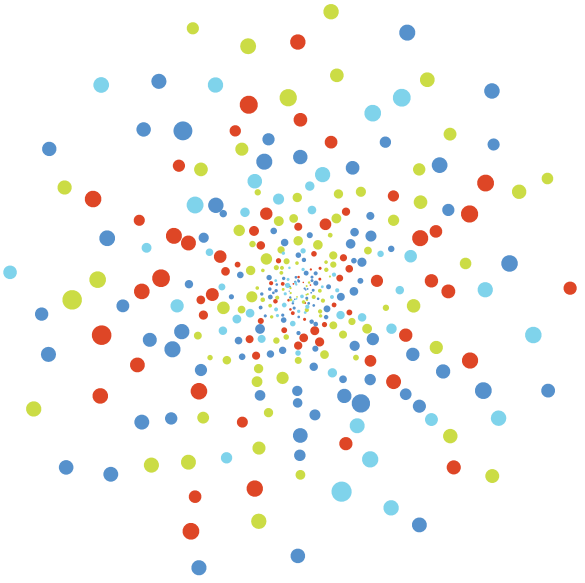
"It's on github"

A common phrase in modern IT jargon is, “It’s on github”, and for users familiar with GitHub, this is an invitation to download, modify and contribute to the project, however for those who have not had an introduction it can seem like a complex topic. GitHub is actually a very simple tool to use, and the simplest way to begin to take advantage of the information stored on GitHub is to simply access a projects main page and look for the “Download ZIP” button at the bottom right of any project's main page. The resulting downloaded file will contain the latest version of the files in the project. What a user does with these files will greatly depend on what the contents are, however one of the most highly encouraged behaviors on GitHub is to provide clear and obvious documentation for a project, so if a new user accesses the front page of a project on Git, they will typically be able to find instructions on how to download and install the project, right on the first page they see.

For users looking to contribute back to a project, the next step would be to sign up for an account on GitHub, and download a graphical-based client to provide a simpler interface to the command line-based git tool. GitHub itself has a graphical client with the Windows version available at http://windows.github.com and the Mac versions at http://mac.github.com. Other common source control tools include SourceTree from Atlassian, available at http://sourcetreeapp.com.

Once a user has an account and a github client, they can “Fork”, or split off a project that is available into their own private repository, make changes and commit those back to their private branch. If those changes work, and the user wishes to contribute them back into the original project, it is possible to submit a “Pull” request, which essentially means that the user is proposing their efforts should be pulled back into the original project. The process can be that simple, though many more advanced projects have standards and rules for contributing to those projects that put in place requirements around how work is committed back into the projects, which may require some reading before attempting to contribute.

Hardware Expansion and Replacement



Section Content

- Expanding and Shrinking the Fabric
  - Switches
    - Add Connected Switch
    - Pre-Provision Switch Before Connection
    - Decommission Existing Switch
  - APICs
    - Add New APIC
    - Decommission Existing APIC
- Hardware Diagnostics and Replacement
  - Identify Hardware Failure
  - Resolve Leaf Hardware Failure
  - Resolve APIC Hardware Failure
  - Diagnose Equipment Failures

Expanding and Shrinking the Fabric

ACME may decide to expand their ACI fabric as their data center grows, which means adding new leaf and spine switches, and possibly APICs. Generally, spine switches are added for more throughput, and leaf switches are added for more access ports. APICs are added as the number of policies and endpoints increase. Additionally, some switches or APICs may need to be decommissioned. Also, there may be times when you need to replace failed hardware, which is discussed in the Hardware Replacements chapter.

This section will walk through the operations of adding and removing switches and APICs in your existing ACI fabric. This is done the same way for both spine and leaf switches. Adding APICs will also be covered.

Switches

There are two ways switches can be added to ACME's existing fabric: by discovering the switches automatically in the APIC after they have been cabled to the fabric, or by pre-provisioning the switches by adding their serial numbers and later connecting them physically to the fabric when the switches arrive. Both methods have the same outcome: an expanded fabric in the matter of minutes. This section will also cover decommissioning switches.

Add Connected Switch

To add a switch that has already been attached to the fabric go through the following steps in the APIC GUI:

1. In the case of a leaf switch, cable it to all of the spine switches. In the case of a spine switch, cable it to all the leaf switches. Ideally, a best-practice ACI fabric is connected in a full mesh topology with every leaf cabled to every spine. All devices should connect to the leaf switches, leaves should never connect to other leaves, and spines should never connect to other spines.
2. On the APIC click on **Fabric** at the top of the screen.
3. Click on **Fabric Membership** in the left navigation pane.

4. When the new switch appears, you'll see a node with a serial number but no Node ID or Node Name configured. Double click the switch and assign a **Node ID** and a **Node Name**. As a best practice, number leaf nodes starting with 101, and spine nodes with 201. Lower numbers are reserved for APICs.
5. Optionally, add a **Rack Name** name. This is commonly used to identify the physical location of the switch in the data center.
6. Click **Submit**.
7. Repeat this process for all new switches connected to the fabric.

Pre-Provision Switch Before Connection

Pre-provisioning a switch is a handy operationally proactive step to get the switch registered before it even arrives to your data center. You will need to know the serial number of the switch you will receive to pre-provision. The following steps walk you through switch pre-provisioning for both leaves and spines:

1. On the menu bar, choose **Fabric.**
2. In the Navigation pane, choose **Fabric Membership.**
3. In the Work pane, choose **Actions > Create Add Fabric Node Member**.
4. In the **Create Add Fabric Node Membe**r dialog box, perform the following actions:
   a. In the pop-up window, enter the **serial number** of the switch that will be arriving.
   b. Assign a **Node ID** and a **Switch Name**. As a best practice, number leaf nodes starting with 101, and spine nodes with 201. Lower numbers are reserved for APICs.
5. Click **Submit**.
   Note: Repeat this process for all switches you wish to pre-provision.

The new entry in the Fabric Membership window will show **Unsupported** in the **Role** column until the switch is actually connected to the fabric, but the switch will immediately become a member of the fabric once it arrives and is cabled.

To be proactive, you can also pre-provision fabric policies. Fabric policies are covered in the Fabric Connectivity chapter. For more information on pre-provisioning policies, refer to the following white paper:

http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731960.html#_Toc405844675

Decommission Existing Switch

Decommissioning a switch can either remove it from the fabric entirely, or remove a switch temporarily to perform maintenance. Ensure you do not have any devices connected, as the switch will not forward traffic after decommissioning. There are two types of switch decommissioning: Regular, and Remove from Controller.

Regular decommissioning can be used for maintenance and essentially silences the switch from reporting faults and sending SNMP information as a temporary solution, while keeping the switch's node ID and fabric membership. The switch will show up under the Disabled Interfaces and Decommissioned Switches folder in the left navigation pane.

The Remove from Controller option will completely remove the switch from the ACI fabric and all APICs. The switch will no longer show up in the fabric membership as a registered node and the infrastructure VTEP IP addresses it was assigned will be removed.

Perform the following steps from the APIC GUI to decommission a switch from the ACI fabric:

1. On the menu bar, choose **Fabric.**
2. In the Navigation pane, choose **Inventory** > **Pod 1.**
3. Click the switch to decommission in the Navigation pane.
   a. Click the **General** tab.
   b. Chose the **Actions > Decommission.**
   c. In the pop-up, choose either **Regular** or **Remove from Controller**.
4. Click **Submit**.

APICs

Add New APIC

Before making any changes to an APIC cluster, ensure each APIC in the cluster is fully fit and change the cluster size to reflect the new controller you are adding to the cluster. Perform the following steps to verify cluster health:

1. On the menu bar, choose **System Controllers.**
2. In the Navigation pane, choose **Controllers.**
   a. Expand the first APIC in the folder.
   b. Click the **Cluster** folder.
   c. Verify every controller shows **Fully Fit** under the **Heath State** column.

If any of the APICs are not fully fit, refer to the following troubleshooting guide:

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/troubleshooting/b_APIC_Troubleshooting.html

Perform the following steps to change the APIC cluster size:

1. On the menu bar, choose **System > Controllers.**
2. In the Navigation pane, choose **Controllers > APIC_*Name* > Cluster.**
3. In the Work pane, choose **Actions > Change Cluster Size.**
   a. Change the **Target Cluster Administrative Size** to reflect the new APIC(s) being added.
   Note: A cluster size of two is not permitted as that does not allow for quorum amongst APICs.
4. Click **Submit**.

Perform the following steps to add a new APIC to the cluster:

1. Install and stage the APIC by connecting it to the fabric by following the hardware installation guide:

   http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-installation-guides-list.html

1. On the menu bar, choose **System > Controllers.**
2. In the Navigation pane, choose **Controllers > APIC_*Name* > Cluster.**
   a. The APIC controllers are added one by one and displayed in the sequential order starting with N + 1 and continuing until the target cluster size is achieved.
   b. Verify that the APIC controllers are in operational state, and the health state of each controller is **Fully Fit** from the **Cluster** folder under the new controller.

Note: It will take several minutes for the APICs to synchronize and join the new APIC to the cluster. Fabric operation will continue normally.

Decommission Existing APIC

When decommissioning APICs, they must be decommissioned sequentially in reverse order. For example, APIC5 must be decommissioned before APIC4. Again, before making any changes to an APIC cluster, ensure each APIC in the cluster is fully fit with the exception of the faulty APIC being decommissioned. You cannot decommission a powered on fully fit APIC.

Perform the following steps to decommission an APIC that needs to be removed from the fabric:

1. On the menu bar, choose **System > Controllers.**
2. In the Navigation pane, choose **Controllers > APIC_*Name* > Cluster.**
   Note: Select an APIC that is NOT being decommissioned.
3. In the Work pane, choose **Actions > Change Cluster Size.**
   a. Change the **Target Cluster Administrative Size** to reflect the new APIC(s) being added.
      Note: A cluster size of two is not permitted as that does not allow for quorum amongst APICs.
4. Click **Submit**.
5. In the Navigation pane, choose **Controllers > APIC_*Name* > Cluster.**
   Note: In the main pane, click the APIC to be decommissioned.
6. In the Work pane, choose **Actions > Actions > Decommission**.
   a. Verify the APIC no longer appears in the **Cluster** folder under any of the remaining APICs.

Hardware Diagnostics and Replacement

The Cisco Application Centric Infrastructure (ACI) fabric employs a combination of key software and hardware features that are specifically designed to reduce the mean time between failures (MTBF) and the mean time to repair (MTTR). Regarding hardware, there are several hot-swappable components on both the leaf and spine switches in addition to a few components that are fixed on the chassis. If ACME ever experiences some sort of power surge or sees a component of their switches go bad, the hot-swappable components enable them to replace failed hardware quickly and non-disruptively.

Examples of hot-swappable components on both the leaves and spines include:

- Power supplies
- Fan trays

Examples of hot-swappable components on the spines include:

- Power supplies
- Fan trays
- Supervisors
- System Controller cards
- Linecard modules

Despite significant advances in the above components that reduce the MTBF, there is always the possibility of a failure on a leaf switch either in switch hardware or software, or a combination of the two that necessitates a leaf replacement. In such an event, the stateless nature of the ACI fabric provides significant advantages to administrators from an operations standpoint.

Identify Hardware Failure

When a hardware failure occurs in the fabric, faults are raised in the system dashboard and are presented to the administrator. For cases where there is a component level

failure with redundant components present in the system, syslog messages and SNMP traps are generated.

Examples of hardware events that generate syslog messages and SNMP traps include:

- Linecard failure on a spine switch
- Supervisor failure on a spine switch
- System controller failure on a spine switch
- Power supply or fan failures on a leaf or a spine switch

While Cisco Application Policy Infrastructure Controller (APIC) is a central point of management for the entire fabric, operations teams can leverage their existing NMS tools. Logging messages can be sent to syslog servers, such as Splunk, or SNMP messages can be sent to NMS systems, such as ZenOSS, to provide alerting. The leaf and spine switches in the ACI fabric also support traditional methods of detecting failures, such as SNMP polling at a set interval. If responses are not received from the switch in a certain timeframe, there is a possibility that the hardware has failed.

However, while the leaf and spine switches report SNMP and Syslog messages for component level failures, the APICs themselves do not have the ability to generate alerts using SNMP or syslog. For example a power supply failure on the APIC will not generate an SNMP or syslog message and must be monitored and remediated using the APIC dashboard.

Resolve Leaf Hardware Failure

As an example of a leaf failure, a Nexus 9396 leaf switch that is a part of the fabric is unreachable, perhaps due to a hardware failure on the uplink modules. You can use the GUI to determine the node health to confirm that the leaf has failed.

To view the node health score:

1. On the menu bar, choose **Fabric > Inventory**.
2. In the Navigation pane, choose **Pod 1**.
   Note: The pod health displays in the Work pane and is zero.

After confirming that the leaf node has failed, you want to remove the failed switch and provision a new switch as part of the fabric. The first step in replacing the failed switch

is to get the failed switch's unique ID (node ID). Each node is assigned an ID in the fabric, which is the reference object that allows a replacement switch with a new serial number to inherit the same stateless configuration that was assigned to the old node.

To view the fabric node IDs using the GUI:

1. On the menu bar, choose **Fabric > Inventory**.
2. In the Navigation pane, choose **Fabric Membership**.

You can also use a single REST API call to periodically poll for a full list of nodes that are at or below a certain health level, as shown in the following example:

```
{{protocol}}://{{apic}}/api/class/topSystem.xml?rsp-subtree-include=health&rsp-
subtree-filter=le(healthInst.cur,"0")
```

In the case of a traditional operations model where each switch was managed as an independent entity, the following high-level procedure replaces the switch:

1. Stand up the replacement switch.
2. Load the correct version of code.
3. Attempt to obtain the latest version of configurations from a configuration repository server.
4. Stage the device with the right configuration file and eliminate any errors. For example, update the AAA, NTP, and syslog servers and the ACLs that are associated with each of them.
5. Copy the old configuration over to the switch.
6. Bring up links one by one and verify if data traffic is flowing correctly.

In an ACI fabric, you can take advantage of the stateless nature of the hardware to instantiate the logical configuration profiles. Replacing the node is as simple as decommissioning the switch and recommissioning it.

To decommission and recommission a switch:

1. On the menu bar, choose **Fabric > Inventory**.
2. In the Navigation pane, expand **Pod 1**.
3. Right click the failed node and choose **Decommission**.

4. Replace the failed leaf switch with the new leaf switch.
5. On the menu bar, choose **Fabric > Inventory**.
6. In the Navigation pane, choose **Fabric Membership**.
7. The new leaf appears with a node ID of 0 and an IP address of 0.0.0.0.
8. In the Work pane, click on the new leaf.
9. Choose **Actions > Commission Switch**.
10. When prompted for the node ID, enter the old node's ID. In most cases, you can also reuse the same leaf name.
11. Click **Update**.

If the new switch is not operational, the new switch's name and node ID are different from the name and ID that you entered. You can get the name and ID by viewing the unreachable nodes.

To view the unreachable nodes:

1. On the menu bar, choose **Fabric > Inventory**.
2. In the Navigation pane, choose **Unreachable Nodes**.
3. Find the new switch and record its name and node ID.
4. Repeat the "To decommission and recommission a switch" procedure, starting with step 5. When prompted for the name and node ID, enter the information that you recorded in this procedure.

When the new leaf switch is commissioned successfully, the APIC automatically loads the correct version of the firmware into the leaf.

To view which version of the firmware that the APIC will load:

1. On the menu bar, choose **Admin > Firmware**.
2. In the Navigation pane, choose **Fabric Node Firmware > Firmware Groups > All**.
   Note: In the Work pane, you can see the target firmware version, which is automatically set to the latest firmware version.

In addition, by leveraging the stateless object modeling that replaces the traditional running configuration on a device, APIC automatically loads the correct running configuration onto the device, such as AAA, syslog, SNMP, NTP, ACLs, bridge domains, and EPGs.

In the event that the replacement switch runs standalone NX-OS software instead of ACI switch software, you might need to copy the ACI switch software image to the switch in question.

To copy the ACI switch software image to the switch:

1. Connect to the switch console.
2. Set the IP address on the mgmt0 interface to allow connectivity between the switch and the APIC.
3. Enable SCP services:

```
# feature scp-server
```

4. Copy the firmware image from APIC to the switch:

```
# scp -r /firmware/fwrepos/fwrepo/switch_image_name
admin@switch_ip_address:switch_image_name
```

5. For dual supervisor systems, ensure that images are copied to the standby supervisor in case of a full chassis replacement by using the command:

```
# copy bootflash:aci_image bootflash://sup-standby/
```

6. Configure the switch not to boot from Cisco NX-OS.

```
switch(config)# no boot nxos
```

7. Save the configuration.

```
switch(config)# copy running-config startup-config
```

8 Boot the active and standby supervisor modules with the ACI image.

```
switch(config)# boot aci bootflash:aci-image-name
```

9 Verify the integrity of the file by displaying the MD5 checksum.

```
switch(config)# show file bootflash:aci-image-name md5sum
```

10 Reload the switch.

```
switch(config)# reload
```

11 Log in to the switch as an administrator.

```
Login: admin
```

12 Verify whether you must install certificates for your device.

```
admin@apic1:aci> openssl asn1parse /securedata/ssl/server.crt
```

13 Look for PRINTABLESTRING in the command output. If "Cisco Manufacturing CA" is listed, the correct certificates are installed. If something else is listed, contact TAC to generate and install the correct certificates for your device.

Once you have confirmed the that certificate is installed and the switch is in ACI mode, the switch should appear as an unmanaged fabric node when connected to the fabric.

Resolve APIC Hardware Failure

In this example, you must identify and remediate a hardware failure on one of the APICs in your APIC cluster.

From the GUI of an operational APIC,

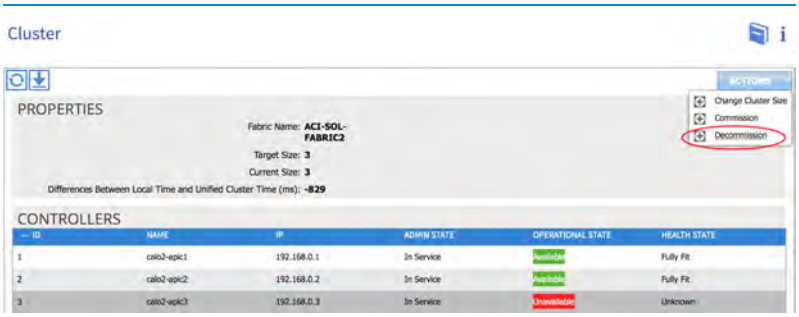
1 On the menu bar choose **System > Controllers**.

2 In the Navigation pane, choose **Controllers > *apic_name* > Cluster**.
Note: In the Work pane, you should see the failed APIC in the "Unavailable" operational state.

3 Record the fabric name, target size, node ID of the failed APIC, and the TEP address space. This information is also available through the **acidiag avread** command on APIC's CLI.

4 In the Work pane, click the failed APIC to select it.

5 Choose **Actions > Decommission**. The APIC changes to an "Out of Service" admin state.



Decommissioning a Failed APIC

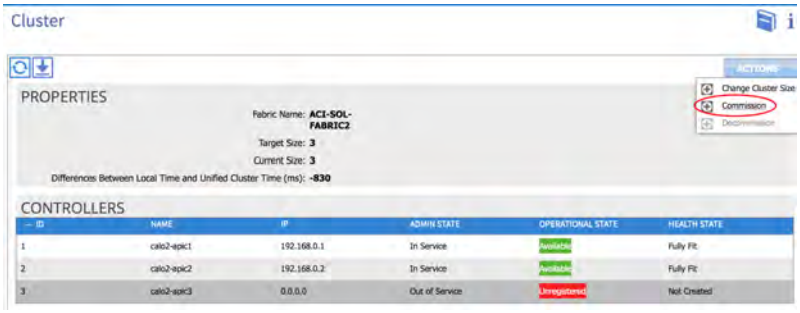
6 Remove the failed APIC from your rack and install the replacement. The new APIC should boot to the initial setup script.

7 Proceed through the setup script and enter the values of the failed APIC that you recorded in step 3. Failure to configure the APIC with the same settings could result in the fabric entering a partially diverged state.

8 Once the new APIC finishes booting, in the Navigation pane, choose **Controllers > *apic_name* > Cluster**. You can choose any APIC.

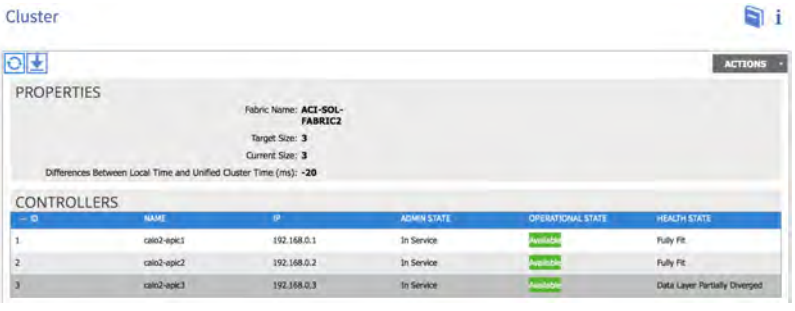
9 In the Work pane, click the new APIC to select it.

10 Choose **Actions > Commission**.



Recommissioning an APIC

11 The new APIC will receive an IP address, which will be reflected in the APIC GUI. It might take 5 to 10 minutes for this to occur. The new APIC might also cycle between the Available and Unavailable operational states before becoming Fully Fit.



Waiting for Cluster Convergence

12 On the command line of the new APIC, you can verify that it has joined the fabric by logging in using the credentials that are configured for the rest of the fabric.

Diagnose Equipment Failures

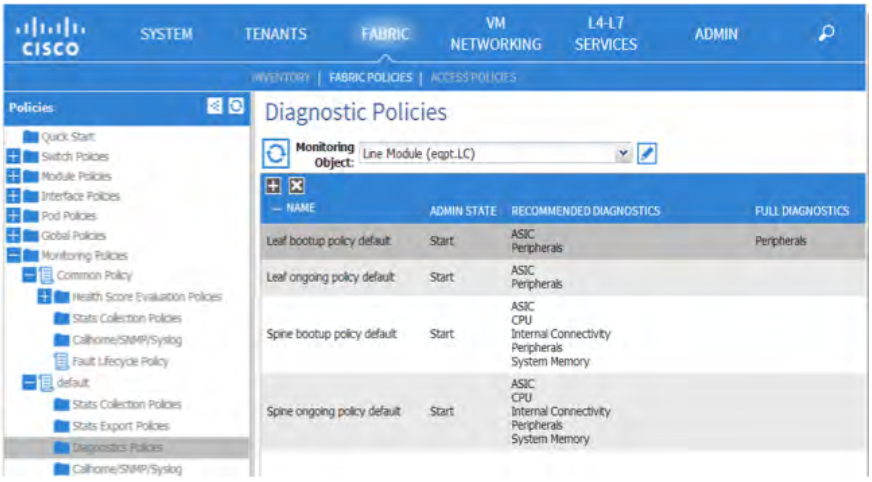
The ACI fabric provides bootup, runtime and on-demand diagnostics to help assess the hardware health of several sub-systems on each leaf and spine switch.

1. **Boot-up** tests run when switch, card boots up. These are typically ONLY disruptive tests. Comes with default set of tests that can be modified. Deployed via selectors.
2. **Health** (aka On-going) tests run periodically. Can only run non-disruptive tests. Comes with default set of tests that can be modified and are deployed via selectors
3. **On-Demand** Tests are to be run on specific ports or cards for troubleshooting, there are no defaults, and they can be disruptive.

By default, tests are logically grouped into collections.

To look at the default diagnostic policies, click fabric > fabric policies > Monitoring policies > default > diagnostics policy

In the work pane select the fabric element that you would like to view the diagnostic monitoring policy for.



Viewing Diagnostic Monitoring Policies in the GUI

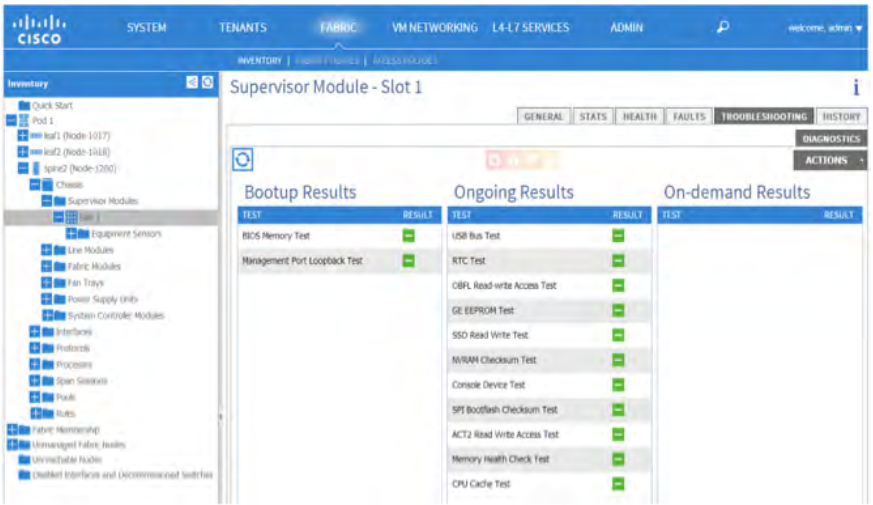
Test results are viewable by clicking on

Fabric > inventory > Pod-1 > Leaf-xx or Spine-xx > Chassis > Supervisor modules > Slot-1

AND

Fabric > inventory > Pod-1 > Leaf-xx or Spine-xx > Chassis > Line modules > Slot-1

Once there, in the work pane select the Troubleshooting tab to view GOLD diagnostic results for the supervisor.



Viewing GOLD Diagnostics Information in the GUI

In a modular chassis-based system such as the Cisco Nexus 9500 series switch, diagnostic results are available for all the supervisor, modules, system controller and the fabric modules in the system.

Create new on-demand diagnostic test

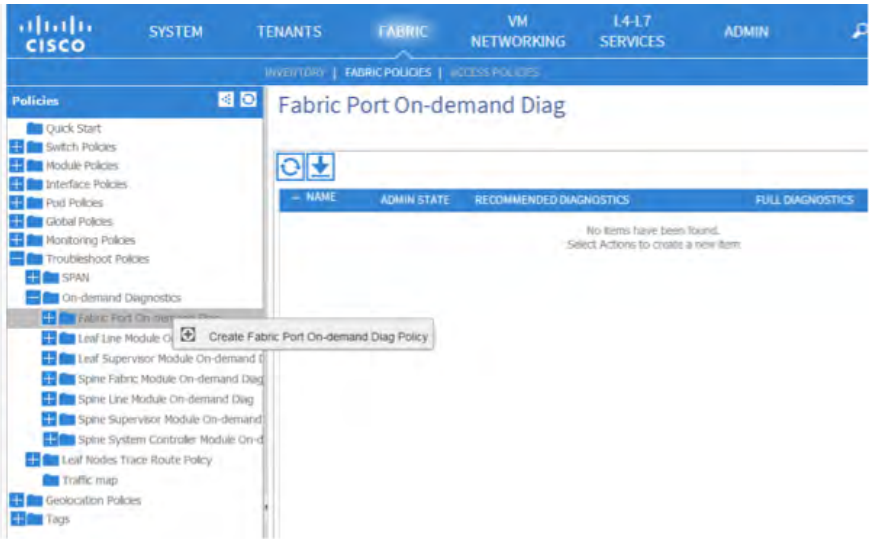
As a part of operational procedures, it may be necessary to validate a system is healthy by running non-disruptive tests on the system.

In order to do this, the APIC GUI allows creation of an On-demand diagnostic test.

To do this, navigate to Fabric > Fabric Policies > Troubleshoot Policies > On demand diagnostics

Once there, right click on the test or test set you would like to run on demand, and click on Create.

In the example below, an operator is creating a fabric-port on-demand diag policy that allows the ability to test the leaf uplink ports going to the spine.

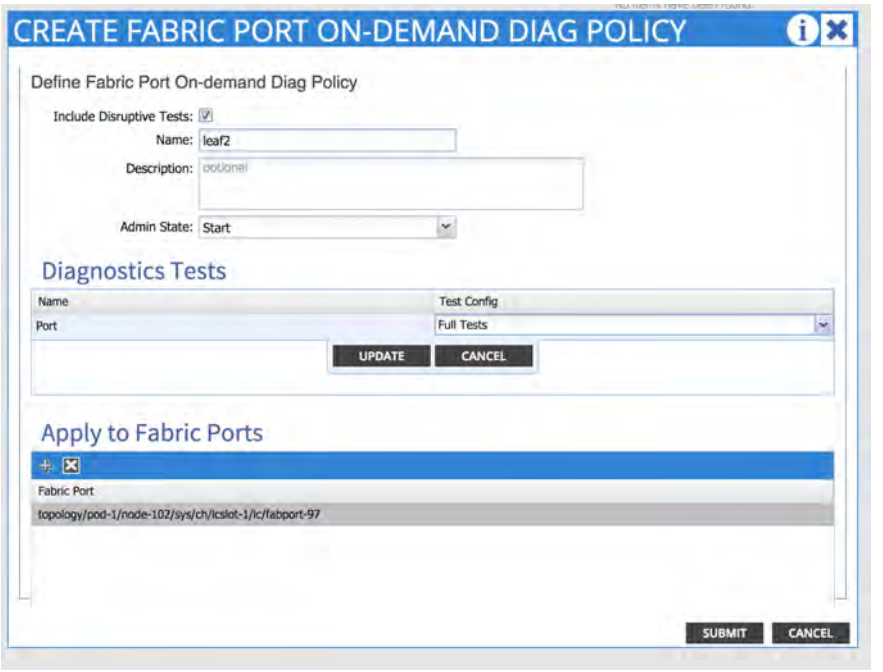


Creating a On-Demand Diag Policy in the GUI

Once you click create, ensure you check the box against "include disruptive tests" if this is the intent.

Enter a name for the test, select the admin state, under the diagnostic tests, select a test config (default is "no tests") and select the fabric ports that the test would need to be run on.

In this case, the operator selects the options for a non-disruptive test to be done on leaf-2, fabric-port 1/97.

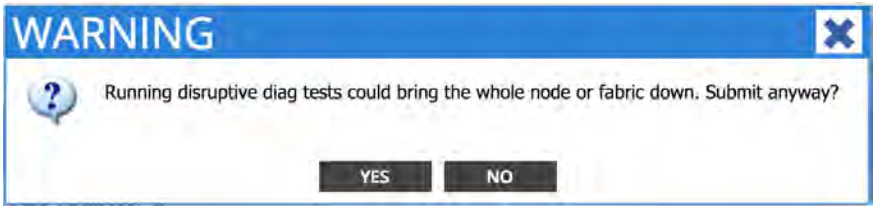


Creating a Fabric Port On-Demand Diag Policy in the GUI

After verifying the parameters above, click "SUBMIT" to submit the policy.

Once submitted, you can kick off the diagnostic test by clicking on the test itself and clicking submit.

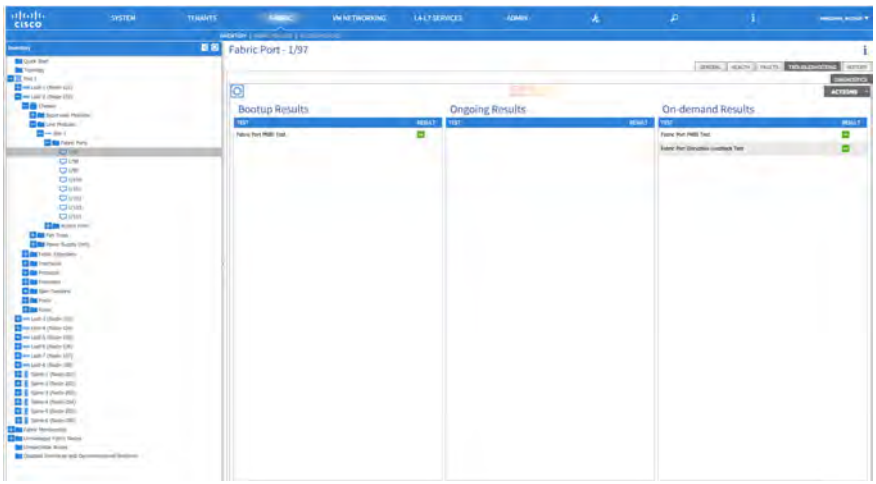
Note that APIC displays a warning message in cases where a non-disruptive tests are selected to the effect of below.



GUI Warning for Executing a Disruptive Diag Policy

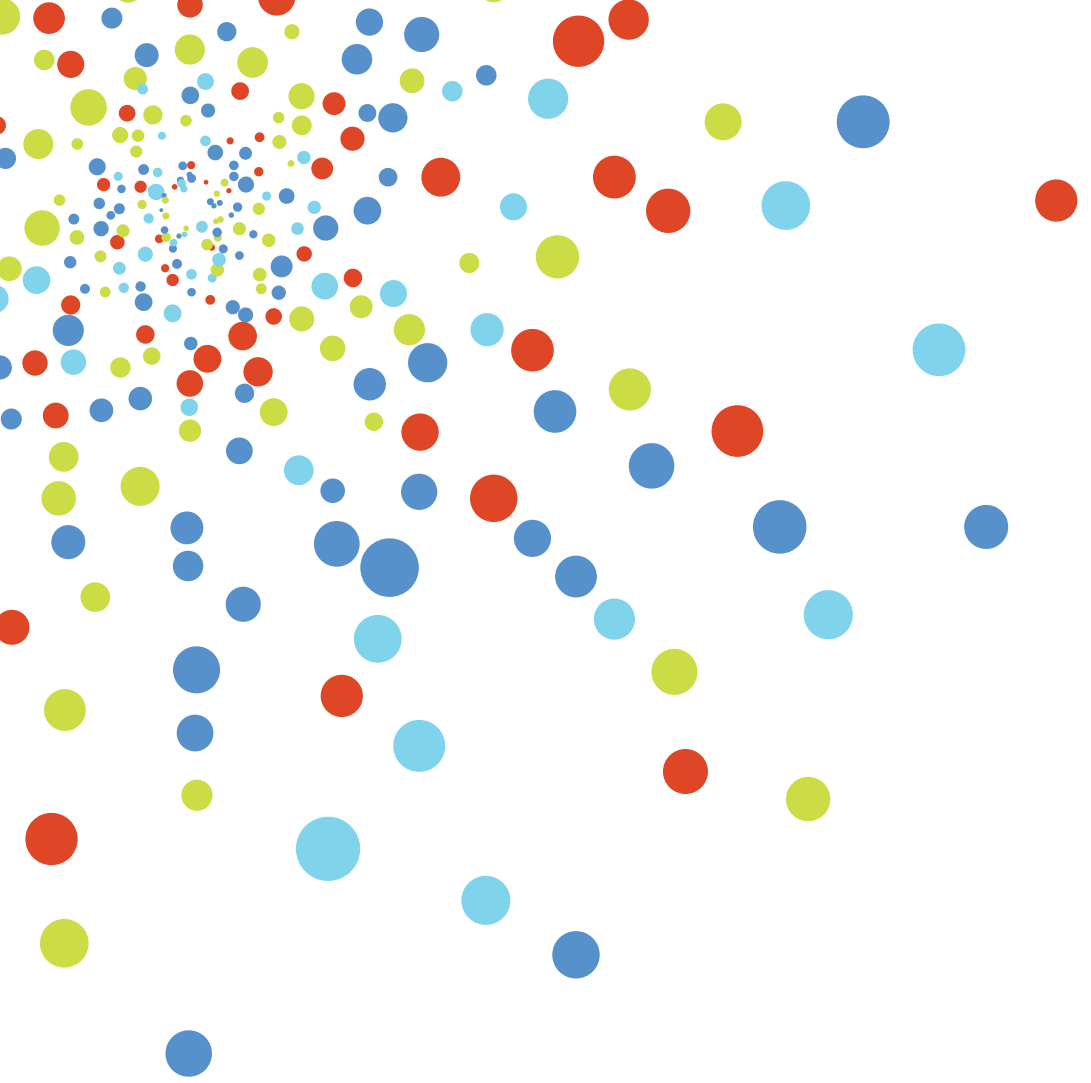
Once you confirm the test run, the diagnostic results can be obtained from the same location as the location for diagnostics that are run at bootup or are ongoing.

Fabric > Inventory > Pod-1 > Leaf-2 > Line-modules > Slot-1 > Fabric ports > 1/97

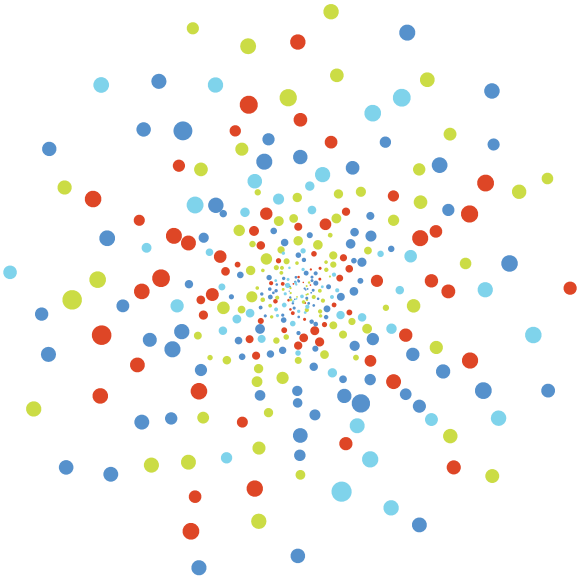


Viewing On-Demand Diag Policy Test Results in the GUI

Note the on-demand results in the right hand corner after the test has completed its run.



Appendix



Classes

The Application Policy Infrastructure Controller (APIC) classes are crucial from an operational perspective to understand how system events and faults relate to objects within the object model. Each event and/or fault in the system is a unique object that can be accessed for configuration, health, fault, and/or statistics.

All the physical and logical components that comprise the Application Centric Infrastructure fabric are represented in a hierarchical management information tree (MIT). Each node in the tree represents a managed object (MO) or group of objects that contains its administrative state and its operational state.

The APIC REST API is a programmatic interface to the APIC that uses a REST architecture. The API accepts and returns HTTP or HTTPS messages that contain JSON or XML documents. You can use any programming language to generate the messages, and the JSON or XML documents that contain the API methods or managed object (MO) descriptions.

You can invoke an API function by sending an HTTP/1.1 or HTTPS POST, GET, or DELETE message to the APIC. The HTML body of the POST message contains a JSON or XML data structure that describes an MO or an API method. The HTML body of the response message contains a JSON or XML structure that contains requested data, confirmation of a requested action, or error information.

The following section is a representation of useful classes for establishing a foundation for monitoring and management of the fabric. The list below is a subset of the full list of the available classes.

To access the complete list of classes, point to the APIC and reference the **doc/html** directory at the end of the URL:

```
https://apic_ip_address/doc/html/
```

Fabric Monitoring

topSystem

Name: top:System

Description: Provides a list of all the devices within the fabric, including controllers, leafs and spines.

Usage: The topSystem class can be used to derive object properties including inb/oob management details, current time, system uptime and current state.

```
topSystem REST :: https://172.16.96.2/api/node/class/topSystem.json
```

fabricNode

Name: fabric:Node

Description: Provides a list of all the nodes that are part of the fabric, including controllers, leafs and spines.

Usage: The fabricNode class can be used to derive object properties including node serial numbers, assigned node ids, node model numbers and device roles.

```
fabricNode REST :: https://172.16.96.2/api/node/class/fabricNode.json
```

faultInst

Name: fault:Inst

Description: Contains detailed information of the fault. This object is attached as a child of the object on which the fault condition occurred. One instance object is created for each fault condition of the parent object. A fault instance object is identified by a fault code.

Usage: The faultInst class can be used to derive all faults associated with the fabric, tenant or individual managed objects within the APIC.

```
faultInst REST :: https://172.16.96.2/api/node/class/faultInst.json
```

fabricHealthTotal

Name: fabric:HealthTotal

Description: The fabric total health score instance.

Usage: The fabricHealthTotal class can be used to derive the overall system health.

```
fabricHealthTotal REST :: https://172.16.96.2/api/node/class/fabricHealthTotal.json
```

fvCEp

Name: fv:CEp

Description: A client endpoint attaching to the network.

Usage: The fvCEp class can be used to derive a list of end points attached to the fabric and the associated ip/mac address and encapsulation for each object.

```
fvCEp REST :: https://172.16.96.2/api/node/class/fvCEp.json
```

fvRsCEpToPathEp

Name: fv:RsCEpToPathEp

Description: This is an internal object that provides a relation to a path endpoint.

Usage: The fvRsCEpToPathEp class can be used to derive path fabric details such as the node and port as well as the tenant details such as the tenant name, application profile and end point group.

```
fvRsCEpToPathEp REST :: https://172.16.96.2/api/node/class/fvRsCEpToPathEp.json
```

eqptFabP

Name: eqpt:FabP

Description: Fabric port, the fabric facing external IO port.

Usage: The eqptFabP class can be used to derive a list of fabric port and the associated details such as the line card and chassis placement.

```
eqptFabP REST :: https://172.16.96.2/api/node/class/eqptFabP.json
```

eqptLeafP

Name: eqpt:LeafP

Description: Fabric port, the non-fabric facing external leaf IO port.

Usage: The eqptFabP class can be used to derive a list of non-fabric port and the associated details such as the line card and chassis placement.

```
eqptLeafP REST :: https://172.16.96.2/api/node/class/eqptLeafP.json
```

eqptCh

Name: eqpt:ChA

Description: The hardware chassis container.

Usage: The eqptCh class can be used to derive a chassis list and the associated details such as the operational state, serial number and model number.

```
eqptCh REST :: https://172.16.96.2/api/node/class/eqptCh.json
```

eqptLC

Name: eqpt:LCA

Description: The line card (IO card), containing IO ports.

Usage: The eqptLC class can be used to derive a list of line cards deployed within the fabric and the associated details such as the redundancy state, model, serial numbers and the number of ports.

```
eqptLC REST :: https://172.16.96.2/api/node/class/eqptLC.json
```

eqptFt

Name: eqpt:Ft

Description: The inventoried fan tray.

Usage: The eqptFt class can be used to derive a list of fan trays and the associated details such as the operational status, model number, serial number and hardware version.

```
eqptFt REST :: https://172.16.96.2/api/node/class/eqptFt.json
```

eqptPsu

Name: eqpt:Psu

Description: The power supply unit.

Usage: The eqptFt class can be used to derive a list of power supplies within the fabric and the associated details such as the model number, serial number, operational status, and the voltage source.

```
eqptPsu REST :: https://172.16.96.2/api/node/class/eqptPsu.json
```

eqptSupC

Name: eqpt:SupC

Description: The supervisor card, which contains the CPU running control plane.

Usage: The eqptFt class can be used to derive a list of supervisor cards deployed within the fabric and the associated details such as the model number, serial number, operational status and redundancy state.

```
eqptSupC REST :: https://172.16.96.2/api/node/class/eqptSupC.json
```

ethpmPhysIf

Name: ethpm:PhysIf

Description: The physical interface information holder.

Usage: The ethpmPhysIf class can be used to derive a list of physical interfaces in the fabric and the associated details such as a the speed, duplex, operational status, and usage state.

```
ethpmPhysIf REST :: https://172.16.96.2/api/node/class/ethpmPhysIf.json
```

dbgAcTrail

Name: dbg:AcTrail

Description: The atomic counter trail.

Usage: The dbgAcTrail class can be used to derive a list of the atomic counters deployed within the fabric and the associated details such as dropped packet statistics and packet counts.

```
dbgAcTrail REST :: https://172.16.96.2/api/node/class/dbgAcTrail.json
```

dbgEpgToEpgRslt

Name: dbg:EpgToEpgRslt

Description: The endpoint group to endpoint group atomic counter, on-demand, entry.

Usage: The dbgEpgToEpgRsIt class can be used to derive a list of the EPG to EPG atomic counters deployed within the fabric, and the associated details such as dropped packet statistics and packet counts.

```
dbgEpgToEpgRsIt REST :: https://172.16.96.2/api/node/class/dbgEpgToEpgRslt.json
```

dbgEpToEpRslt

Name: dbg:EpToEpRslt

Description: The endpoint to endpoint atomic counter, On-demand, Entry.

Usage: The dbgEpToEpTsIt class can be used to derive a list of the endpoint to endpoint atomic counters deployed within the fabric and the associated details such as dropped packet statistics and packet counts.

```
dbgEpToEpTsIt REST :: https://172.16.96.2/api/node/class/dbgEpToEpRslt.json
```

VMM Monitoring

compVm

Name: comp:Vm

Description: The Virtual machine object.

Usage: The compVm class can be used to derive a list of virtual machines deployed within the fabric and the associated details such as the name and state.

```
compVm REST :: https://172.16.96.2/api/node/class/compVm.json
```

compHv

Name: comp:Hv

Description: An object representing the compute hypervisor.

Usage: The compVm class can be used to derive a list of compute hypervisor deployed within the fabric and the associated details such as the name and status.

```
compHv REST :: https://172.16.96.2/api/node/class/compHv.json
```

fvRsVm

Name: fv:RsVm

Description: A relation to a virtual machine connected to a hypervisor. This is an internal object.

Usage: The fvRsVm class can be used to derive the relationship of the virtual machines connected to the hypervisor.

```
vRsVm REST :: https://172.16.96.2/api/node/class/fvRsVm.json
```

fvRsHyper

Name: fv:RsHyper

Description: A relation to the hypervisor that controls and monitors the APIC VMs. This is an internal object.

Usage: The fvRsHyper class can be used to derive the relationship of the hypervisor that controls and monitors the APIC VMs.

```
fvRsHyper REST :: https://172.16.96.2/api/node/class/fvRsHyper.json
```

vmmCtrlrP

Name: vmm:CtrlrP

Description: The VMM controller profile, which specifies how to connect to a single VM management controller that is part of containing policy enforcement domain. For example, the VMM controller profile could be a policy to connect a VMware vCenter that is part a VMM domain.

Usage: The vmmCtrlrP class can be used to derive the ip address and the data-center name of the connected VM domain.

```
vmmCtrlrP REST :: https://172.16.96.2/api/node/class/vmmCtrlrP.json
```

Layer 4 to Layer 7 Monitoring

vnsAbsGraph

Name: vnsAbsGraph

Description: The abstract graph is made up of abstract nodes and used to define the traffic flow through a service function such as load balancing, SSL offload, or firewall. Abstract nodes are comprised of service nodes such as a service node balancer (SLB) or firewall (FW), abstract term nodes (the nodes that are connected to endpoint groups), and connections.

Usage: The class vnsAbsGraph can be used to derive a list of service graph templates configured on the APIC, along with their properties.

```
vnsAbsGraph  REST :: https://172.16.96.2/api/node/class/vnsAbsGraph.json
```

vnsLDevVip

Name: vnsLDevVip

Description: An L4-L7 device cluster, which is represented by a single virtual IP (VIP). The configuration is pushed down to the VIP address.

Usage: The class vnsLDevVip can be used to derive all the VIPs configured for the logical device clusters in the fabric

```
vnsLDevVip  REST :: https://172.16.96.2/api/node/class/vnsLDevVip.json
```

vnsCDev

Name: vnsCDev

Description: The individual service device, which is used to define a concrete l4-l7 service device.

Usage: The class vnsCDev can be used to derive a list of concrete devices configured as part of the L4-7 service integration

```
vnsCDev  REST :: https://172.16.96.2/api/node/class/vnsCDev.json
```

vnsLif

Name: vnsLif

Description: The logical interface, which is associated with a set of concrete interfaces from the L4-L7 device cluster.

Usage: The class vnsLif can be used to derive the connection between a service graph and device interfaces.

```
vnsLif  REST :: https://172.16.96.2/api/node/class/vnsLIf.json
```

vnsLDevCtx

Name: vnsLDevCtx

Description: A device cluster context, which points to the device cluster used to pick a specific device based on contract, subject, and function label or names. To specify a wild card, set the name to Any.

Usage: The class vnsLDevCtx can be used to derive the node and contract name.

```
nsLDevCtx  REST :: https://172.16.96.2/api/node/class/vnsLDevCtx.json
```

vnsRsLDevCtxToLDev

Name: vnsRsLDevCtxToLDev

Description: A source relation to the abstraction of a service device cluster or of a proxy object for a logical device cluster in the tenant.

Usage: The class vnsRsLDevCtxToLDev can be used to derive the relationship between vnsLDevCtx and vnsLDev.

```
vnsRsLDevCtxToLDev  REST :: https://172.16.96.2/api/node/class/vnsRsLDevCtxToLDev.json
```

Statistics

compHostStats1h

Name: comp:HostStats1h

Description: A class that represents the most current statistics for host in a 1 hour sampling interval. This class updates every 15 minutes.

Usage: The compHostStats1h class can be used to derive the statistics associated with the compute hypervisor.

```
compHostStats1h REST :: https://172.16.96.2/api/node/class/compHostStats1h.json
```

compRcvdErrPkts1h

Name: comp:RcvdErrPkts1h

Description: A class that represents the most current statistics for received error packets in a 1 hour sampling interval. This class updates every 15 minutes.

Usage: The compRcvdErrPkts1h class can be used to derive the most current statistics for received error packets.

```
compRcvdErrPkts1h REST :: https://172.16.96.2/api/node/class/compRcvdErrPkts1h.json
```

compTrnsmtdErrPkts1h

Name: comp:TrnsmtdErrPkts1h

Description: A class that represents the most current statistics for transmitted error packets in a 1 hour sampling interval. This class updates every 15 minutes.

Usage: The compTrnsmtdErrPkts1h class can be used to derive the most current statistics for transmitted error packets.

```
compTrnsmtdErrPkts1h REST ::
https://172.16.96.2/api/node/class/compTrnsmtdErrPkts1h.json
```

Authentication, Authorization, and Accounting

aaaModLR

Name: aaa:ModLR

Description: The AAA audit log record. A log record is automatically generated whenever a user modifies an object.

Usage: The aaaModLR class can be used to derive a fabric based audit log for all changes and events.

```
aaaModLR REST :: https://172.16.96.2/api/node/class/aaaModLR.json
```

aaaUser

Name: aaa:User

Description: A locally-authenticated user account.

Usage: The aaaUser class can be used to derive a list of user accounts deployed within the fabric.

```
aaaUser REST :: https://172.16.96.2/api/node/class/aaaUser.json
```

aaaRemoteUser

Name: aaa:RemoteUser

Description: A remote user login account.

Usage: The aaaUser class can be used to derive a list of remote user accounts deployed within the fabric.

```
aaaRemoteUser REST :: https://172.16.96.2/api/node/class/aaaRemoteUser.json
```

Fabric Capacity

Policy TCAM

Name: eqptcapacityPolEntry5min

Description: Policy CAM entry statistics. A class that represents the most current statistics for policy entry in a 5 minute sampling interval. This class updates every 10 seconds.

Usage: The eqptcapacityPolEntry5min class can be used to derive the current value associated with the Policy TCAM usage.

```
eqptcapacityPolEntry5min REST ::
http://172.16.96.2/api/class/eqptcapacityPolEntry5min.json
```

Prefix TCAM

Name: eqptcapacityL3Entry5min

Description: Layer3 entry statistics. A class that represents the most current statistics for layer3 entry in a 5 minute sampling interval. This class updates every 10 seconds.

Usage: The eqptcapacityL3Entry5min class can be used to derive the current value associated with the Prefix TCAM usage.

```
eqptcapacityL3Entry5min  REST ::
https://172.16.96.2/api/class/eqptcapacityL3Entry5min.json
```

SNMP/SYSLOG

SNMP Trap Destination

Name: snmpTrapDest

Description: A destination to which traps and informs are sent.

Usage: The snmpTrapDest class can be used to derive the current list of snmp trap destinations implemented within the fabric.

```
snmpTrapDest REST :: https://172.16.96.2/api/node/class/snmpTrapDest.json
```

Prefix TCAM

Name: syslogRemoteDest

Description: The syslog remote destination host enables you to specify syslog servers to which messages from the APIC and fabric nodes should be forwarded.

Usage: The syslogRemoteDest class can be used to derive the current list of syslog remote destinations implemented within the fabric.

```
syslogRemoteDest REST :: https://172.16.96.2/api/node/class/syslogRemoteDest.json
```

Use Cases

The class **faultInst** used in **Use Case #1** and **Use Case #2** below can be replaced with any of the managed object classses discussed above or specified within the APIC documentation. The *Cisco APIC Command-Line Interface User Guide* may also be helpful for understanding the following sections - see: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/cli/b_APIC_CLI_User_Guide.html

Case 1: Creating an application script to retrieve the current list of faults in the fabric.

This use case may be typical for environments where an ACI administrator wishes to obtain the list of current faults in the fabric. The user has the option of collecting the results via CLI, Visore, POSTMAN and/or Cobra. Please refer to the section above for application specific access and explantions.

From a CLI perspective, use the following command to perform the query:

```
admin@apic1:~> moquery -c faultInst
```

From a Visore perspective, use the following parameters to perform the query.:

```
Class or DN    :: faultInst
 Property       :: n/a
 Op             :: n/a
 Value          :: n/a
```

From a POSTMAN perspective, user the following REST GET to perform the query:

```
GET http://<your apic ip address>/api/node/class/faultInst.xml
```

From a Cobra perspective, use the following class query:

```
# Class Query
 classQuery= ClassQuery('faultInst')
 for fault in md.query(classQuery):
        print fault.name
```

Sample Cobra script to capture faults within the fabric:

```
#!/usr/bin/env python
 import cobra.mit.access
 import cobra.mit.session
 from cobra.mit.session import LoginSession
```

```
  from cobra.mit.request import ClassQuery
  ls = cobra.mit.session.LoginSession('https://'<your apic ip address>, <username>,
<password>, secure=False)
  md = cobra.mit.access.MoDirectory(ls)
  md.login()
  # Class Query
  classQuery= ClassQuery('faultInst')
  for fault in md.query(classQuery):
         print fault.name
```

Case 2: Creating an application script to retrieve the current list of faults in the fabric that have been caused by a failed configuration.

This use case may be typical for environments where an ACI administrator wishes to cobtain the list of current faults in the fabric. The user has an option of collecting the results via CLI, Visore, POSTMAN and/or Cobra. Please refer to the section above for application specific access and explantions.

From a CLI perspective, use the following command to perform the query:

```
admin@apic1:~> moquery -c faultInst -f 'fv.faultInst.cause=="config-failure"'
```

From a Visore perspective, use the following parameters to perform the query:

```
Class or DN    :: faultInst
 Property      :: cause
 Op            :: ==
 Value         :: config-failure
```

From a POSTMAN perspective, use the following REST GET to perform the query:

```
 GET http://<your apic ip address>/api/node/class/faultInst.xml?query-target-
filter=and(eq(faultInst.cause,"config-failure"))
```

From a Cobra perspective, use the following class query:

```
# Class Query
 classQuery= ClassQuery('faultInst')
 classQuery.propFilter = 'wcard(faultInst.cause, "{0}")'.format('config-failure')
 for fault in md.query(classQuery):
        print fault.name
```

Cobra Script to capture faults casued by configuration failures.

```
#!/usr/bin/env python
 import cobra.mit.access
 import cobra.mit.session
 from cobra.mit.session import LoginSession
 from cobra.mit.request import ClassQuery
 ls = cobra.mit.session.LoginSession('https://'<your apic ip address>, <username>,
<password>, secure=False)
 md = cobra.mit.access.MoDirectory(ls)
 md.login()
 # Class Query
 classQuery= ClassQuery('faultInst')
 for fault in md.query(classQuery):
        print fault.name
```

Case 3: Creating an application script to retrieve the properties for a specific managed object, DN

This use case may be typical for environments where an ACI administrator wishes to obtain the properties of the tenant name **Common**. The user has an option of collecting the results via CLI, Visore, POSTMAN and/or Cobra. Please refer to the section above for application specific access and explantions.

From a CLI perspective, use the following command to perform the query:

```
admin@apic1:~> moquery -d uni/tn-common
```

From a Visore perspective, use the following parameters to perform the query:

```
Class or DN    :: uni/tn-common
 Property       :: n/a
 Op             :: n/a
 Value          :: n/a
```

From a POSTMAN perspective, use the following REST GET to perform the query:

```
GET http://<your apic ip address>/api/node/mo/uni/tn-common.xml?query-target=self
```

From a Cobra perspective, use the following class query:

```
# DN Query
 dnQuery= DnQuery('uni/tn-common')
 for results in md.query(dnQuery):
        print results.dn
```

Cobra Script to capture faults casued by configuration failures.

```
#!/usr/bin/env python
 import cobra.mit.access
 import cobra.mit.session
 from cobra.mit.session import LoginSession
 from cobra.mit.request import DnQuery
 ls = cobra.mit.session.LoginSession('https://'<your apic ip address>, <username>,
<password>, secure=False)
 md = cobra.mit.access.MoDirectory(ls)
 md.login()
 # DN Query
 dnQuery= DnQuery('uni/tn-common')
 for results in md.query(dnQuery):
        print results.dn
```

Case 4: Creating an application script to retrieve the current list of endpoints (mac-addresses) attached to the fabric

This use case may be typical for environments where an ACI administrator wishes to create an application script to capture the list of current endpoints attached to the fabric along with the node details pertaining to each endpoint.

Cobra Script to capture faults casued by configuration failures.

```
 #!/usr/bin/env python
  from cobra.mit.access import MoDirectory
  from cobra.mit.session import LoginSession
  from cobra.mit.request import ClassQuery
  lls = cobra.mit.session.LoginSession('https://'<your apic ip address>, <username>,
<password>, secure=False)
  md = MoDirectory(ls)
  md.login()
  q = ClassQuery('fvCEp')
  q.subtree = 'children'
  q.subtreeClassFilter = 'fvRsCEpToPathEp'
  mos = md.query(q)
  for mo in mos:
         for child in mo.rscEpToPathEp:
                print child.dn
```

Package Decoder

There are several abbreviations used in the names of classes in the ACI object model. Here are some descriptions of commonly used abbreviations, which may help when deciphering what class objects are when using them with REST calls.

Package Decoder

Aaa: authentication, authorization, accounting

ac: atomic counters

actrl: access control

actrlcap: access control capability

adcom: appliance director communication

aib: adjacency information base

arp: address resolution protocol

bgp: border gateway protocol

callhome: Cisco smart call home services

cap: capability

cdp: Cisco discovery protocol

cnw: node cluster

comm: communication policy

comp: compute

compat: compatibility

condition: health policy

config: configuration policy

coop: Council of Oracles protocol

copp: control plane policing policy: contains set of rules describing policer rates

ctrlr: controller

ctx: context

datetime: date/time policy

dbg: debug

dbgac: debug atomic counters

dbgexp: debug export policy

dhcp: dynamic host configuration protocol

dhcptlv: dynamic host configuration protocol type length value

dhcptlvpol: dynamic host configuration protocol type length value policy

dns: domain name service

draw: graph visualization for GUI

epm: endpoint manager

eqpt: equipment

eqptcap: equipment capability

eqptcapacity: equipment capacity

eqptdiag: equipment diagnostics

eqptdiagp: equipment diagnostics policy

ethpm: ethernet policy manager

event: event policy

extnw: external network

fabric: fabric

fault: fault policy, counters

file: file path, config import/export policy

firmware: firmware

fmcast: fabric multicast

fsm: finite state machine

fv: fabric virtualization

fvns: fabric virtualization namespace

fvtopo: fabric virtualization topology

geo: geolocation

glean: glean adjacency

ha: high availability

health: health score

hvs: hypervisors virtual switch

icmp: internet control protocol

icmpv4: internet control protocol version 4

icmpv6: internet control protocol version 6

ident: identity

igmp: internet group management protocol

igmpsnoop: internet group management protocol snooping

im: interface manager module

imginstall: image install

infra: infrastructure

ip: internet protocol

ipv4: internet protocol version 4

ipv6: internet protocol version 6

isis: intermediate system to intermediate system

isistlv: intermediate system to intermediate system type length value

l1: layer 1

l1cap: layer 1 capability

l2: layer 2

l2cap: layer 2 capability

l2ext: layer 2 external

l3: layer 3

l3cap: layer 3 capability

l3ext: layer 3 external

l3vm: Layer 3 Virtual Machine

lacp: link aggregation protocol

lbp: load balancing policy

leqpt: loose equipment (unmanaged nodes, not in the fabric)

lldp: link layer discovery protocol

lldptlv: link layer discovery protocol type length value

lldptlvpol: link layer discovery protocol type length value policy

maint: maintenance

mcast: multicast

mcp: master control processor

memory: memory statistics

mgmt: management

mo: managed object

mock: mock (objects used on the simulator mostly for showing stats/faults/etc)

mon: monitoring

monitor: monitor (SPAN)

naming: abstract for objects with names

nd: neighbor discovery

nw: network

oam: ethernet operations, administrations and management

observer: observer for statistics, fault, state, health, logs/history

opflex: OpFlex

os: operating system

ospf: open shortest path first

pc: port channel

pcons: **generated and used by internal processes**

phys: physical domain profile

ping: ping execution and results

pki: public key infrastructure

pol: policy definition

policer: traffic policing (rate limiting)

pool: object pool

pres: **generated and used by internal processes**

proc: system load, cpu, and memory utilization statistics

psu: power supply unit policy

qos: quality of service policy

qosm: qos statistics

qosp: qos/ 802.1p

rbqm: debugging

regress: regression

reln: **generated and used by internal processes**

repl: **generated and used by internal processes**

res: **generated and used by internal processes**

rib: routing information base

rmon: remote network monitoring/ interface stats/counters

rpm: route policy map

rtcom: route control community list

rtctrl: route control

rtextcom: router extended community

rtflt: route filter

rtleak: route leak

rtmap: RPM route map

rtpfx: route prefix list

rtregcom: route regular community list

rtsum: route summarization address/policy

satm: satellite manager

snmp: simple network management protocol

span: switched port analyzer

stats: statistics collection policies

statstore: statistics data holders

stormctrl: storm control (traffic suppression) policy

stp: spanning tree protocol definitions and policy

sts: Service Tag Switching (used for services insertion)

svccore: core policy

svi: switched virtual interface/ routed VLAN interface

synthetic: synthetic objects (for testing)

sysdebug: system debug

sysfile: system files

syshist: system cards reset records/history

syslog: syslog policy

sysmgr: system manager (firmware, supervisor, system states, etc)

sysmgrp: container for cores policy & abstract class for all qos policy definitions

tag: alias (use descriptive name for dn), tags (group multiple objects by a descriptive name)

task: task execution, instance, and result

test: abstract class for test rule, subject, and result

testinfralab: test infrastructure

tlv: type, length, value system structures

top: system task manager for processor activity

topoctrl: topology control policy (sharding, fabric LB, fabric VxLan, etc)

traceroute: traceroute execution and results

traceroutep: traceroute end points

trig: triggering policy

tunnel: tunneling

uribv4: ipv4 unicast routing information base entity

vlan: vlan instances

vlanmgr: vlan manager control plane

vmm: virtual macine manager (controller, vmm policy and definitions)

vns: virtual network service (L4-L7 policy and definitions)

vpc: virtual port channel (vpc policy and definitions)

vsvc: service labels (provider/consumer)

vtap: translated address of external node (NATed IP of service node)

vxlan: Virtually extensible LAN definitions

vz: virtual zones (former name of the policy controls) i.e. Contracts

Model Naming schemes

Rs: Relationship source
Rt: Relationship target
Ag: Aggregated stats
BrCP: Binary Contract Profile

Acronyms and Definitions

Overview

This section is designed to provide a high level description of terms and concepts that get brought up in this book. While ACI does not change how packets are transmitted on a wire, there are some new terms and concepts employed, and understanding those new terms and concepts will help those working on ACI communicate with one another about the constructs used in ACI to transmit those bits. Associated new acronyms are also provided.

This is not meant to be an exhaustive list nor a completely detailed dictionary of all of the terms and concepts, only the key ones that may not be a part of the common vernacular, or which would be relevant to the troubleshooting exercises that were covered in the troubleshooting scenarios discussed.

A

AAA: acronym for Authentication, Authorization, and Accounting.

ACI External Connectivity: Any connectivity to and from the fabric that uses an external routed or switched intermediary system, where endpoints fall outside of the managed scope of the fabric.

ACID transactions: ACID is an acronym for Atomicity, Consistency, Isolation, Durability – properties of transactions that ensure consistency in database transactions. Transactions to APIC devices in an ACI cluster are considered ACID, to ensure that database consistency is maintained. This means that if one part of a transaction fails the entire transaction fails.

AEP: Attachable Entity Profile – this is a configuration profile of the interface that gets applied when an entity attaches to the fabric. An AEP represents a group of external entities with similar infrastructure policy requirements. AEPs are also the mechanism that ties the physical port to the domain (physical or virtual) to a switch policy.

ALE: Application Leaf Engine, an ASIC on a leaf switch.

APIC: Application Policy Infrastructure Controller is a centralized policy management controller cluster. The APIC configures the intended state of the policy to the fabric.

API: Application Programming Interface used for programmable extensibility.

Application Profile: Term used to reference an application profile-managed object reference that models the logical components of an application and how those components communicate. The AP is the key object used to represent an application and is also the anchor point for the automated infrastructure management in an ACI fabric.

ASE: Application Spine Engine, an ASIC on a Spine switch.

B

BGP: Border Gateway Protocol, on the ACI fabric Multi-Protocol BGP is used to distribute reachability information within the fabric, and Internal BGP is used to peer the fabric with external Layer 3 devices.

Bridge Domain: An ACI construct that defines Layer 2 forwarding behaviors (Broadcast, ARP flooding, etc.) for each unique Layer 2 forwarding domain. Bridge Domains are also a container for IP subnets and are where fabric Layer 3 gateway functionality is configured. BDs can emulate the behavior of a traditional VLAN but are not constrained by forwarding scale limitations. In the ACI object model, a BD is a child of a Private Layer 3 or context.

C

CLOS fabric: A multi-tier nonblocking leaf-spine architecture network.

Cluster: Set of devices that work together as a single system to provide an identical or similar set of functions.

Contracts: A logical container for the subjects which relate to the filters that govern the rules for communication between endpoint groups. ACI works on a white list policy model. Without a contract, the default forwarding policy is to not allow any communication between EPGs, but communication within an EPG is allowed.

Context: A Layer 3 forwarding domain, equivalent to a VRF, and in ACI vernacular a Private Layer 3.

D

DLB: Dynamic Load Balancing – a network traffic load-balancing mechanism in the ACI fabric based on flowlet switching.

DME: Data Management Engine, a service that runs on the APIC that manages data for the data model.

dMIT: distributed Management Information Tree, a representation of the ACI object model with the root of the tree at the top and the leaves of the tree at the bottom. The tree contains all aspects of the object model that represent an ACI fabric.

Dn: Distinguished name – a fully qualified name that represents a specific object within the ACI management information tree as well as the specific location information in the tree. It is made up of a concatenation of all of the relative names from itself back to the root of the tree. As an example, if policy object of type Application Profile is created, named commerce workspace within a Tenant named Prod, the dn would be expressed as uni/tn-Prod/ap-commerceworkspace.

E

EP: Endpoint - Any logical or physical device connected directly or indirectly to a port on a leaf switch that is not a fabric facing port. Endpoints have specific properties like an address, location, or potentially some other attribute, which is used to identify the endpoint. Examples include virtual-machines, servers, storage devices, etc.

EPG: End Point Group. A collection of endpoints that can be grouped based on common requirements for a common policy. Endpoint groups can be dynamic or static.

F

Fault: When a failure occurs or an alarm is raised, the system creates a fault-managed object for the fault. A fault contains the conditions, information about the operational state of the affected object and potential resolutions for the problem.

Fabric: The collective endpoints associated with an ACI solution (Leaf, Spine and Virtual Switches plus APICs)

FCAPS: The ISO model defines network management tasks. FCAPS is n acronym for fault, configuration, accounting, performance, security, the management categories

Filters: Filters define the rules outlining the Layer 2 to layer 4 fields that will be matched by a contract.

Flowlet switching: An optimized, multipath, load-balancing methodology based on research from MIT in 2004. Flowlet Switching is a way to use TCP's own bursty nature to more efficiently forward TCP flows by dynamically splitting flows into flowlets, and splitting traffic across multiple parallel paths without requiring packet reordering.

G

GUI: Graphical User Interface.

H

HTML: HyperText Markup Language, a markup language that focuses on the formatting of web pages.

Hypervisor: Software that abstracts the hardware on a host machine and allows the host machine to run multiple virtual machines.

Hypervisor integration: Extension of ACI Fabric connectivity to a virtual machine manager to provide the APIC with a mechanism for virtual machine visibility and policy enforcement.

I

IFM: Intra-Fabric Messages, Used for communication between different devices on the ACI fabric.

Inband Management (INB): Inband Management. Connectivity using an inband management configuration. This uses a front panel (data plane) port of a leaf switch for external management connectivity for the fabric and APICs.

IS-IS: Link local routing protocol leveraged by the fabric for infrastructure topology. Loopback and VTEP addresses are internally advertised over IS-IS. IS-IS announces the creation of tunnels from leaf nodes to all other nodes in fabric.

J

JSON: JavaScript Object Notation, a data encapsulation format that uses human readable text to encapsulate data objects in attribute and value pairs.

L

Layer 2 Out (l2out): Layer 2 connectivity to an external network that exists outside of the ACI fabric.

Layer 3 Out (l3out): Layer 3 connectivity to an external network that exists outside of the ACI fabric.

L4-L7 Service Insertion: The insertion and stitching of VLANs/Layer 3 constructs of virtual or physical service appliances (Firewall, IDS/IPS, Load Balancers, DLP, etc....) into the flow of traffic. Service nodes operate between Layers 4 and Layer 7 of the OSI model, where as networking elements (i.e. the fabric) operate at layers 1-3).

Labels: Used for classifying which objects can and cannot communicate with each other.

Leaf: Network node in fabric providing host and border connectivity. Leafs connect only to hosts and spines. Leafs never connect to each other.

M

MO: Managed Object – every configurable component of the ACI policy model managed in the MIT is called a MO.

Model: A model is a concept which represents entities and the relationships that exist between them.

Multi-tier Application: Client–server architecture in which presentation, application logic, and database management functions require physical or logical separation and require networking functions to communicate with the other tiers for application functionality.

O

Object Model: A collection of objects and classes are used to examine and manipulate the configuration and running state of the system that is exposing that object model. In ACI the object model is represented as a tree known as the distributed management information tree (dMIT).

Out-of-Band management (OOB management): External connectivity using a specific out-of-band management interface on every switch and APIC.

P

Port Channel: A Port link aggregation technology that binds multiple physical interfaces into a single logical interface and provides more aggregate bandwidth and link failure recovery without a topology change.

R

RBAC: Role Based Access Control, which is a method of managing secure access to infrastructure by assigning roles to users, then using those roles in the process of granting or denying access to devices, objects and privilege levels.

Representational State Transfer (REST): a stateless protocol usually run over HTTP that allows a client to access a server-side or cloud-based API without having to write a local client for the host accessing the API. The location that the client accesses usually defines the data the client is trying to access from the service. Data is usually accessed and returned in either XML or JSON format.

RESTful: An API that uses REST, or Representational State Transfer.

Rn: Relative name, a name of a specific object within the ACI management information tree that is not fully qualified. A Rn is significant to the individual object, but without context, it's not very useful in navigation. A Rn would need to be concatenated with all the relative names from itself back up to the root to make a distinguished name, which then becomes useful for navigation. As an example, if an Application Profile object is created named "commerceworkspace", the Rn would be "ap-commerceworkspace" because Application Profile relative names are all prefaced with the letters "ap-". See also the Dn definition.

S

Service graph: A mechanism within ACI that automates redirection of traffic and VLAN stitching based on defined parameters. Any services that are required are treated as a service graph that is instantiated on the ACI fabric from the APIC. Service graphs identify the set of network or service functions that are needed by the application, and represent each function as a node.

Spine: Network node in fabric carrying aggregate host traffic from leafs, connected only to leafs in the fabric and no other device types.

Spine/Leaf topology: A clos-based fabric topology in which spine nodes connect to leaf nodes, leaf nodes connect to hosts and external networks.

Subnets: Contained by a bridge domain or an EPG, a subnet defines the IP address range that can be used within the bridge domain.

Subjects: Contained by contracts and create the relationship between filters and contracts.

Supervisor: Switch module that provides the control plane for the 95xx switches.

T

Tenants: The logical container to group all policies for application policies.

V

Virtualization: Technology used to abstract hardware resources into virtual representations and allowing software configurability.

vPC: virtual Port Channel, in which a port channel is created for link aggregation, but is spread across no more or less than 2 physical switches.

VRF: Virtual Routing and Forwarding - A Layer 3 namespace isolation methodology to allow for multiple contexts to be deployed on a single device or infrastructure.

VXLAN: VXLAN is a Layer 2 overlay scheme transported across a Layer 3 network. A 24-bit VXLAN segment ID (SID) or VXLAN network identifier (VNID) is included in the encapsulation to provide up to 16 million VXLAN segments for traffic isolation or segmentation. Each segment represents a unique Layer 2 broadcast domain. An ACI VXLAN header is used to identify the policy attributes if the application endpoint within the fabric, and every packet carries these policy attributes.

X

XML: eXtensible Markup Language, a markup language that focuses on encoding data for documents rather than the formatting of the data for those documents.

Reference Material

Topics that are outside of the scope of this operations guide may be documented in other places. This section includes links to other helpful reference documentation for further reading and viewing.

ACI Install and Upgrade Guides

http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-installation-guides-list.html

ACI Getting Started - Fabric Initialization

http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-installation-and-configuration-guides-list.html

ACI Design Guide

http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731960.html#_Toc405844675

ACI Troubleshooting Guides

http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-troubleshooting-guides-list.html

https://datacenter.github.io/aci-troubleshooting-book/

ACI White Papers

http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/white-paper-listing.html

ACI Case Studies

www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/customer-case-study-listing.html

ACI Demos, Presentations and Trainings

www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/sales-resources-list.html

ACI Ecosystem Compatability List

http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/solution-overview-c22-732445.html

ACI Partners and Customers Presentations

http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/presentations-listings.html

ACI Solutions Overview

http://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/solution-overview-listing.html

ACI Toolkit

http://datacenter.github.io/acitoolkit/

ACI Compatability Tool

http://www.cisco.com/web/techdoc/aci/acimatrix/matrix.html

AVS Configuration and Scalability Guides

http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/products-installation-and-configuration-guides-list.html

AVS Topologies and Solution Guide

http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/products-technical-reference-list.html

APIC Command-Line Interface User Guide

http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-command-reference-list.html

APIC Layer 4 to Layer 7 Services Deployment Guide

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/L4-L7_Services_Deployment/guide/b_L4L7_Deploy.html

Cobra Docs

http://cobra.readthedocs.org/en/latest/

Cobra GitHub

http://github.com/datacenter/cobra

Connecting ACI to Outside Layer 2 and 3 Networks

http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c07-732033.html

Fabric Connectivity Video

https://www.youtube.com/watch?v=_iQvoC9zQ_A

Nexus CLI to Cisco APIC Mapping

http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-configuration-examples-list.html

POSTman

http://www.getpostman.com

Supported SNMP MIB List

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html


CISCO