Implementing application centric infrastructure to build a scalable secure data center

Xi, Yewen

2020

Xi, Y. (2020). Implementing application centric infrastructure to build a scalable secure data center. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/139945>

<https://doi.org/10.32657/10356/139945>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 21:26:51 SGT

IMPLEMENTING APPLICATION CENTRIC INFRASTRUCTURE TO BUILD A SCALABLE SECURE DATA CENTER

XI YEWEN

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING (SCSE)

APRIL 2020

IMPLEMENTING APPLICATION CENTRIC INFRASTRUCTURE TO BUILD A SCALABLE SECURE DATA CENTER

XI YEWEN

School of Computer Science and Engineering (SCSE)

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Master of Engineering

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

01 Nov 2019

.....

Date

.....

Xi Yewen

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

22 Oct 2019

.....

Date

.....

Luo Jun

Authorship Attribution Statement

This thesis **does not** contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

01 Nov 2019

.....

Date

.....

Xi Yewen

Abstract

Software defined network (SDN) has been a buzzword for a decade, researchers and vendors all have their own interpretation of SDN, however, that's no clear definition on what it is. Based on a few conversations with end customers and system integrators, what the industry needs is a system can translate intent to orchestrate the network operations.

The aim of this research is to study how to transfer a traditional network configuration from device by device to be application centric and centralized orchestrated, easily consuming the network. In a result, shorten the time to provision network such as VLAN, port group, policies from weeks to days, and removes the barrier between application and infrastructure, by implementing Cisco application centric infrastructure.

This research is not to create any new network fabric but to bridge the application requirements to IT operations which is heavily demanded in the real-world deployment. A lab implementation has been done to show how easily to setup an Application Centric Infrastructure (ACI) network from scratch to achieve simplified automation, centralized visibility, flexibility, scalable performance and integrated security. A proof of concept lab has been built to migrate one of the largest ASEAN financial customer's network to ACI and fulfil their application requirements. It is proven to accelerate application time to market and reduce risk of changes.

After achieving the network automation in the data center, further research can be extended to the campus network to give IT team end to end visibility and control from users to applications.

Acknowledgement

I would like to express my deepest gratitude to my project supervisor, Professor Luo Jun for his great patience and insightful comments at every stage of this project. Without his guidance, I will not be able to complete my master study.

Table of Content

ABSTRACT.....	V
ACKNOWLEDGEMENT	VI
LIST OF FIGURES.....	VIII
INTRODUCTION	1
BACKGROUND.....	1
MOTIVATION AND OBJECTIVES.....	2
LITERATURE REVIEW	4
SECTION 1 UNDERLAY AND OVERLAY, CONTROL PLANE AND DATA PLANE	4
1.1 VXLAN	5
1.2 LISP.....	6
SECTION 2 ACI BUILDING BLOCKS	7
2.1 Spine-Leaf Fabric.....	7
2.2 APIC Controller.....	8
SECTION 3 CISCO ACI POLICY MODEL	8
3.1 Why Policy Model	9
3.2 Tenant.....	10
3.3 Context.....	10
3.4 Bridge Domain	11
3.5 Endpoint Group.....	11
3.6 Contract	12
PROJECT: IMPLEMENTING APPLICATION CENTRIC INFRASTRUCTURE TO BUILD A SCALABLE SECURE DATA CENTER.....	13
PROJECT 1: BUILD A SECURE DC IN LAB ENVIRONMENT	14
Part 1: Building ACI Forwarding Constructs	14
Part 2: Configuring Application Profile	18
Part 3: Configure VMM Integration.....	20
Part 4: ACI and Firewall Integration	23
Part 5: Micro-segmentation	27
PROJECT 2: AUTOMATING FABRIC	31
Part 1: Automating Fabric by API	33
Part 2: Automating Fabric by orchestration tool.....	37
PROJECT 3: TRANSFORMATION OF FINANCIAL NETWORK TO BE DIGITALIZATION READY	39
Migration Planning.....	39
ACI Physical Hardware Considerations.....	39
Data Center Migration Considerations.....	41
ACI Migration Approaches.....	42
ACI Migration Deployment Considerations	43
Proof of Concept Lab Environment	45
Requirements.....	46
Solution.....	47
Implementation	48
Project Summary	50
CONCLUSION AND FUTURE WORK	51
REFERENCE	52

List of Figures

FIGURE 1 OVERLAY AND UNDERLAY NETWORK	5
FIGURE 2 VXLAN ENCAPSULATION FORMAT.....	6
FIGURE 3 LOGICAL RELATIONSHIP OF TENANT, BRIDGE DOMAIN, EPG AND CONTRACT.....	10
FIGURE 4 CREATE TENANT	15
FIGURE 5 SPECIFY TENANT DETAILS	15
FIGURE 6 CREATE VRF	16
FIGURE 7 CREATE BRIDGE DOMAIN	16
FIGURE 8 CREATE SUBNET.....	16
FIGURE 9 SPECIFY SUBNET DETAILS	17
FIGURE 10 VIEW APPLICATION STRUCTURE	17
FIGURE 11 CREATE APPLICATION PROFILE	18
FIGURE 12 CREATE CONTRACT.....	18
FIGURE 13 SPECIFY IDENTITY OF CONTRACT.....	19
FIGURE 14 CREATE VCENTER DOMAIN	20
FIGURE 15 ADD VMM DOMAIN ASSOCIATION	21
FIGURE 16 ADD VM TO ACI PORTGROUP.....	21
FIGURE 17 ASSIGN PORTGROUP POLICY TO VM	22
FIGURE 18 TEST PING FROM WEBSERVER TO GATEWAY.....	22
FIGURE 19 SSH TEST FROM WEB-SERVER TO APP-SERVER.....	22
FIGURE 21 MICRO-SEGMENTATION LOGIC DIAGRAM.....	29
FIGURE 22 MICRO-SEGMENTATION APN CONFIGURATION	29
FIGURE 23 CREATE USEG EPG.....	30
FIGURE 24 DEFINE MICRO-SEGMENTATION POLICY WITH VM NAME STARTS WITH HR	30
FIGURE 25 ACI SCRIPT FOR CONTROLLER AUTHENTICATION	33
FIGURE 26 CREATE USER SCRIPT.....	35
FIGURE 27 DELETE USER SCRIPT	36
FIGURE 28 TENANT ONBOARDING WORKFLOW OF UCSF	37
FIGURE 29 CREATE 3 TIERS APPLICATION BY USING UCSF WORKFLOW	38
FIGURE 30 VERIFY PARAMETER AND EXECUTE TASK ON ACI	38
FIGURE 31 ACI ZERO TOUCH FABRIC BRING UP	44
FIGURE 32 ACI EPGs IDENTIFICATION USING VLAN IDs.....	45
FIGURE 33 LAB TOPOLOGY	46

Introduction

Background

Software defined network (SDN) is a generic term which usually refers to separation of network control plane and data plane to achieve agile and flexible network management.

There are a few drivers of SDN, such as increasing IT self-service requests, vast virtualized application workload, rapid cloud adoption and booming network programmability etc.

Every vendor and research institution have their own understanding and approach of SDN.

Though SDN does not have an explicit definition; what we have to admit is network virtualization and new application operational models require infrastructure changes.

Generally, there are three major approaches to the network automation in the network industry:

1. Centralized controller for control and data plane such as Openflow approach. When the first generation of SDN came out, Openflow has created a system that sitting outside the network act as the brain to tell how the network traffic should flow.
2. Network virtualization which can be setup and tear down based on users' demand and build virtualized network service like virtual firewall. VMware is one of the leaders in this domain.
3. Programmable fabric opens the API for developer to use scripts and JSON/XML to push network configuration to simplify traditional CLI command line configuration. Arista is one of the rising vendors with this approach.

Openflow is the first standard that defines the communication between the control and forwarding layers of an SDN architecture. It was the leading approach of SDN domain with contribution and support from a few vendors, including Alcatel-Lucent, Big Switch Networks,

Brocade Communications and Radisys (En.wikipedia.org, 2020). The rising challenger and visionaries Cisco, VMware and Juniper also played an important role in SDN domain and contributed with their SDN solution.

At this stage, no one can judge which is the right way to achieve SDN and ultimately achieve the true network intelligence. The intermediate stage is which approach really helps to simplify the network operation, provides better network visibility and reduces the time to innocent.

Motivation and Objectives

Working at a network vendor as network consultant, the daily job is to understand customer challenges and propose solutions to address their pain points. Most existing data center network I've seen have the following typical architecture:

1. Multiple layers architecture and separated with firewalls, with 1GE/10GE connectivity built with meshed network.
2. Usually gateways locate on core switches, firewalls to separate services and segments.

However, with rapidly changing DevOps requirements, lots of challenges cannot be addressed by this kind of traditional environment because of:

1. Security policy becomes very critical and demanding, which brings much more administrative overhead and become the performance and throughput bottlenecks. Planning, implementing and day 2 operations of security policy is on top mind of infrastructure team.
2. Individually managing tons of network devices is error prone and hard to keep track with fast changing requirements, even by using management applications or scripts such as puppet, chef.

3. Due to the two reasons above, the entire network is inflexible to manage and often involves multiple teams (infrastructure, security, application etc) effort to make a single change to be applied.

Literature Review

The main solution evaluated in this project is Cisco Application Centric infrastructure (ACI) network. As its name indicates, it is an application driven approach of software defined network, it introduced logical network provision of stateless hardware. And it supports multi-tenancy network fabric which provides capacity and segmentations to host multiple logical networks.

Before dive into ACI, a few fundamental concepts need to be discussed first.

Section 1 Underlay and Overlay, Control Plane and Data Plane

Software overlay is the heart of SDN which takes care the change of the network, and the primary technology builds the data/transport underlay is Virtual Extensible LAN (VxLAN).

The underlay network builds the network connectivity by using the physical devices and routing protocols. It is the foundation to provide network reachability, performance and redundancy.

The overlay network is a virtual network built on top of underlay network. Comparing to the underlay physical topology, it is a logical topology to virtually connect devices, so that it can provide flexibility of segregating traffic and adding additional services.

Network overlay is not a new term introduced by data center networking, MPLS, IPSec, DMVPN etc. are all examples of network overlays.

Software defined network provides separation of control plane and data plane which promotes operation simplification and cost reduction (Krishnan and Figueira, 2015).

Figure 1 Overlay and underlay network.

Cisco.com. 2020. [online] Available at: <https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/software-defined-access/white-paper-c11-740585.pdf> [Accessed 10 April 2020].

1.1 VXLAN

VXLAN stands for Virtual eXtensible Local Area Network, it is a framework for overlaying virtualized Layer 2 Networks over Layer 3 Networks (Mahalingam et.al., 2014). VXLAN is defined in RFC 7348.

The traditional VLAN has 12 bits address space for VLAN ID which results maximum 4096 VLAN segments. VLAN was working fine in local area environment, however, with growing demands of stretching layer 2 across geo locations for data center co-location, disaster recovery. VLAN hits the bottleneck of limited address space. Thus, number of technologies has been introduced by different vendors such as Fabric Path (FP), Overlay transport virtualization (OTV).

VXLAN came into the picture as an open protocol to build a layer 2 overlay scheme over layer 3 network, which uses 24-bit for VXLAN network identifier, usually known as VNID.

Each VNID is a unique layer 2 broadcast domain. It provides up to 16M segments for traffic isolation, so that it can provide much greater extensibility than VLAN.

The original packets are encapsulated with additional 50 bytes MAC in IP VXLAN header with VNID information and a few reserve bits.

Figure 2 VXLAN Encapsulation Format.

Cisco.com. 2020. [online] Available at: <https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/software-defined-access/white-paper-c11-740585.pdf> [Accessed 12 April 2020].

The Cisco ACI VXLAN header provides a tagging mechanism to identify properties that are associated with frames forwarded through the Cisco ACI fabric. The header is an extension of the Layer 2 LISP with additional information about policy group, load and path metric, counter, ingress port, and encapsulation. The VXLAN header is not associated with any specific Layer 2 segment or Layer 3 domain, but provides a multifunction tagging mechanism that is used in the Cisco ACI fabric.

1.2 LISP

The nature of overlay requires a protocol for better routing system scalability and efficiency. Locator ID Separation Protocol (LISP) is a new routing protocol by splitting the device identity to two namespaces ("Locator ID Separation Protocol (LISP) VM Mobility Solution White Paper", 2011):

Endpoint identifiers represent who you are of an endpoint.

Routing locators represent where you are of an endpoint.

At the center of LISP is the mapping database which serves as the controller plan to map identity and location, so it can support use case such as virtual machine moves to different data center location but keeps its identify, resulting in shortest path.

ACI runs VXLAN on under layering with IS-IS routing protocol to build IP routing connectivity and runs LISP on over layering.

Section 2 ACI Building Blocks

The design of the Cisco ACI is based on the fabric as a whole, as opposed to treating the fabric switches individually. All physical components form a system. This architecture allows multivendor solution that is open source and allows for traditional, virtualized, and next-generation applications by decoupling the restrictions of classical networking. ACI physical fabric consists of APIC controller cluster, spine switches and leaf switches.

2.1 Spine-Leaf Fabric

The Cisco ACI fabric uses a spine-leaf topology. The links between the spine and leaf offer very high bandwidth (40G/100G) providing an integrated overlay for host routing. All host traffic arrives at the ingress leaf is carried over on integrated overlay. All access links from endpoints are attached to the leaves, which provide a high port density, while the spine (minimum of two spines for redundancy) aggregates the fabric bandwidth.

By using the spine-leaf topology, the fabric is easier to build, test, and support. Scale is supported by adding more leaf nodes if there are not enough ports for connecting endpoints traffic and adding spine nodes if the fabric is not large enough to carry the load of the whole fabric traffic. The symmetrical topology allows for optimized forwarding behavior. Every host-to-host connection will traverse two hops in predictable pattern. So that this

design can provide a high- bandwidth, low-latency, low-oversubscription, and scalable network solution at low cost.

2.2 APIC Controller

Another important component of ACI fabric is APIC controller. Cisco APIC is a policy controller. It relays the intended state of the policy to the fabric. The APIC does not represent the control plane and does not sit in the traffic path. The hardware must consist of three or more servers in a redundant cluster. It provides centralized automation of fabric management, monitoring, troubleshooting and maintenance of the whole fabric state.

Section 3 Cisco ACI Policy Model

ACI follows the application tiers and logical building blocks to consume the network. For one typical application with Web, App and Database three tiers, the relationship has been defined by 'contract' with context of network policy, QoS, service and filters. Every application maintains their own application profile which includes all the necessary constructs, profile moves along application without re-configuration.

Endpoints can be grouped based on their logical relationship to end point group (EPG), within the same EPG, end points can talk to each other, outside the same EPG is whitelist by default; EPG cannot talk with each other until appropriate contracts applied.

Though the ACI simplifies network in a way of eliminating duplicated configurations. The implementation of ACI still needs understanding of existing traditional network, the ACI solution, and translation between the traditional network and the ACI network.

Recently Cisco has opened the APP Center which is similar to a marketplace for eco-system partners and developers to build application based on the ACI underlaying fabric. There is

also some existing toolkit such as cobra and arya available to accelerate the development and time to market.

3.1 Why Policy Model

ACI introduced the application policy model which is the key differentiator of traditional network. The application policy model defines application requirements. Based on the requirements, each device will instantiate the required changes. ACI fabric decouples security and forwarding from physical network and virtual network attributes. Within ACI fabric, IP addresses are fully portable and not tied to the device identity. State of the network will be updated by devices autonomously based on configured policy that set within the application network profile.

Networks have become very complex nowadays, with hundreds of thousands of ACLs, multiple protocols for redundancy, multipathing, and strict ties between addressing and policy. To provide a simplified network model, ACI relies first on abstracting the logical configuration of an application from the physical layer of the hardware configuration.

By starting with the abstraction, Cisco ACI can offer rapid application provisioning on a simplified network. This ability is independent of where resources reside, whether that location is virtual or physical, or on multiple different hypervisors.

Policy is the one in the data center makes scalability, security, and replication more feasible. Policy is used to describe connectivity can be extended as requirements change and as capabilities of the network change. This policy will be updated with hardware and software capabilities of the network. The logical model provides a common toolset for defining the intended policy state of objects.

The following sections will discuss ACI configuration constructs and hierarchy tenant, context, bridge domain and end point group in details.

Figure 3 Logical relationship of tenant, bridge domain, EPG and contract.

Cisco.com. 2020. [online] Available at:

<<https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/software-defined-access/white-paper-c11-740585.pdf>> [Accessed 10 April 2020].

3.2 Tenant

At the high level, the Cisco ACI policy model is built in one or more tenants. The tenants create a multi-tenant environment and allow the segmentation of logical network operation. You can use one or more tenants for different customers, business units, or groups, depending on organizational needs. For example, a given enterprise might use one tenant for the entire organization or use different tenants for different departments; while a managed service provider might have multiple customers using one or more tenants to represent their respective organizations.

3.3 Context

Tenants further break down into private Layer 3 networks, which directly refer to a VRF instance with a separate IP space. Each tenant can have one or more private Layer 3 networks, again depending on its business needs. Private Layer 3 networks provide a way to further separate the organizational and forwarding requirements within a given tenant.

Because contexts use separate forwarding instances, IP addressing can overlap across different contexts.

3.4 Bridge Domain

A bridge domain is a container for subnets. It is a set of physical or logical ports that share the same broadcast domain (Naranjo and Salazar, 2017). It can be used to define a Layer 2 boundary. Bridge domains is not the new name of VLANs, it has much lesser restrictions than VLANs.

In the ACI new model, all routing is host-based. In IPv4, x.x.x.x/32 routes are created to define traffic paths to the endpoints. The endpoints can move freely, without constraints caused by the static subnet structure. The VLANs have only local significance and can be reused all over the fabric. One or more local subnets can be associated with a bridge domain within a tenant context and those subnets can be reused if required. Within the Cisco ACI fabric, Layer 2 concepts, such as flooding, are only enabled if required, such as ARP flooding can be enabled, however, not recommended. When the fabric brings up, it attempt to discover all endpoints and their locations. After a successful discovery, the fabric delivers the traffic only to the intended endpoints.

3.5 Endpoint Group

Endpoint groups, which are known as EPGs, are collections of similar endpoints representing an application tier. EPGs provide a logical grouping for objects that require a similar policy. For example, an EPG could be a group of components that make up the web tier of an application.

When multiple endpoints are assigned to a single EPG, the fabric ensures unrestricted connectivity between them. This communication can take place within the same IP subnet, or between different IP subnets. With inter-subnet connectivity, the pervasive default gateway feature is responsible for routing the traffic between the endpoints.

The EPGs are a subcomponent of a bridge domain. A bridge domain can include any number of IP subnets. This flexibility allows to span an EPG over multiple subnets. In other words, connectivity is restricted not by IP subnet boundaries but by EPG definitions.

3.6 Contract

The relationship between two communicating EPGs is referred to as a contract. From a high level, a contract defines the authorized traffic profile. By default, the ACI fabric blocks all communication between EPGs. Contracts must be defined to permit specific traffic types between the EPGs.

Project: Implementing Application Centric Infrastructure to Build a Scalable Secure Data Center

Digital transformation drives network evolution. From the year 2000 to 2015 is the information era focusing on connectivity with high reliability, the Internet penetration almost increased 7 times (Cooney, 2018). The primary goal was getting connected, today the connectivity issue has largely been addressed.

The digital era starts from year 2015, businesses run on their applications, IT needs to provide a platform for innovation and agility that supports and enhances security.

We are no longer dealing with human scale. We now have to consider IoT scale, which means considering people, devices and things such as wearable technology. Customers are looking at ways go beyond virtualization of network appliances. The new network needs to virtualize functions, not only virtualize compute.

Taking financial industry as example, they are not only serving their customers in the branches from 9am to 5pm anymore, people can bank anytime online. Thus, one of the key asks of most enterprise customers is how to make sure business continuity with active-active data center design, recover service after failure without notifying users.

Apart from traditional data center challenges such as business continuity, scalability, the growing complexity of data center network makes a more secure network design become the top demand (Al-Qahtani and Farooq, 2017). That's always a double-edged sword, storing data digitally also creates new attack surface. No company would like to see their name on the newspaper because of data breach.

ACI enables IT departments to meet those needs through:

- Simplified automation by an application-driven policy model

- Centralized visibility with real-time, application health status
- Open software for DevOps teams and ecosystem partner integration
- Highly scalable fabric in hardware and multi-tenancy in logical operation
- Integrated security with single point of control

This report will walk through how to migrate the traditional network to ACI and build an application centric data center. In total, 3 projects have been done, which focus on the key requirements of enterprise customer's data center: security, automation and scalability. A phased approach has been taken from building a secure data center in lab environment with automation to transforming a financial customer's network to be digitalization ready with minimum downtime.

Project 1: Build a Secure DC in Lab Environment

The heart of ACI is policy model which enables specification of application requirements. This policy model manages the entire fabric, including infrastructure, authentication, security, services, applications, and diagnostics.

The entire ACI fabric is designed based on a whitelist model of security policies, instead of traditional blacklist model. Any traffic is allowed only if it is explicitly allowed, otherwise denied. Inherently, ACI fabric exhibits similar behavior to a firewall in handling L3/L4 traffic.

The first part of this project focuses on building secure data center with network and system integration in a lab environment.

Part 1: Building ACI Forwarding Constructs

Task Summary

- Create a tenant
- Create a VRF for the new tenant

- Create bridge domain for VMData-Web and VMData-App

This task is for building the basic infrastructure of ACI network top down from tenant, VRF to bridge domain. It provides the constructs for application running and security policies implementation.

The first step is to login to ACI GUI, select Tenants on the top bar, then Add Tenant.

Figure 4 Create Tenant

Enter **Tenant1** as the tenant name and click submit.

Figure 5 Specify tenant details

Expand **Tenant1** created in the previous step. Expand **Networking**, right click on VRFs and choose Create VRF 'T1_Production'.

Figure 6 Create VRF

Check the box of 'Create a Bridge Domain' and Enter the Bridge Domain name VMData-Web for the name and click button 'Finish'.

 A screenshot of the 'Create VRF' wizard, Step 2: Bridge Domain. The 'Specify Bridge Domain for the VRF' section is shown. The 'Name' field is set to 'VMData-Web'. The 'Alias' field is empty. The 'Description' field contains the text 'optional'. The 'Type' is set to 'regular'. The 'Forwarding' is set to 'Optimize'. The 'Endpoint Dataplane Learning' checkbox is checked. The 'Limit IP Learning To Subnet' checkbox is unchecked. The 'Config BD MAC Address' checkbox is checked. The 'MAC Address' field contains the text '00:22:BD:F8:19:FF'.

Figure 7 Create bridge domain

Expand Networking to Bridge Domain, then VMData-Web, right click on subnets and select 'Create Subnet' button.

Figure 8 Create subnet

Enter 192.168.10.1/24 as the Web Servers Network Gateway IP and click 'Submit' button.

Create Subnet

Specify the Subnet Identity

Gateway IP: 192.168.10.1/24

Treat as virtual IP address: ☐

Make this IP address primary: ☐

Scope: ☒ Private to VRF
☐ Advertised Externally
☐ Shared between VRFs

Description: optional

Subnet Control: ☒ ND RA Prefix
☐ Querier IP

L3 Out for Route Profile: select a value

Route Profile: select a value

ND RA Prefix policy: select a value

SUBMIT CANCEL

Figure 9 Specify subnet details

Follow the same process and create another bridge domain for VMData- App with gateway 192.168.11.1/24

Click on 'Networking' button to verify two Bridge Domains have been created, configured and associated with VRF T1_Production.

Figure 10 View application structure

It shows ACI is an infrastructure with built-in multi-tenancy, VRF is used for further isolation of network with individual routing table. Bridge domain is the layer 2 construct, similar to VLAN with lesser restriction and more functionalities.

Part 2: Configuring Application Profile

Task Summary

- Create App_Servers EPG which will contain App Server Virtual Machines.
- Create Web_Servers EPG which will contain Web Server Virtual Machines.

After building the network constructs, the second task is to create virtual machines for application hosting.

Right click on 'Application Profiles' button and select 'Create Application Profile'.

Figure 11 Create application profile

Enter Name 'T1_AppProfile' and click '+' button under EPGs to create an Endpoint Group.

Enter name 'Web_Servers', for the Bridge Domain select 'VMDData-Web'. Expand the Consumed Contract and select 'Create Contract' button.

Figure 12 Create contract

Enter Name App_Contract and click '+' button to create a Subject.

Figure 13 Specify identity of contract

Enter Name App_Services, click button '+' next to Filters. Click on the new button '+' sign to create a new filter. Name the Filter App_Service_Ports and click button '+' to add Entries of ICMP under IP protocols.

Follow the same process to create EPG of App Servers and consumer and provider contracts.

Contract applied allowed Ping and TCP 5000

Table 1 Contract policies

ContractName	ICMP	TCP5000
EthernetType	IP	IP
IP Protocol	ICMP	TCP
Destination Port From/To		5000/5000

ACI follows the zero-trust model, by default, all the communication is blocked until explicitly specified by contracts. It secures east-west data center traffic by reducing the attack surface. Contracts created can be reused by other applications, which reduced administrative tasks of data center admins and human errors.

Part 3: Configure VMM Integration

The beauty of ACI is to have single point of control of VMM, security under the same dashboard to remove the barrier between network and system team and shorten the application provisioning from weeks to hours.

Task Summary

- Integrate a Virtual Machine Manager (VMM) with ACI.

Isolation of server and network team is another common challenge of enterprise data center management. VM team usually only focus on server virtualization, while network team only focus on building network and policies. However, running an application requires collaboration of two teams. Time to market is extremely important for business in this competitive digital market. Inefficient communication between two teams would make project timeline been dragged. Thus, this experiment shows how ACI unified both server and network team in a single control point.

The first step is to select 'VM Networking'. Right click at 'VMware' and select 'Create vCenter Domain'.

Figure 14 Create vCenter domain

Enter the vCenter name and create a dynamic associated VLAN pool.

Enter the vCenter host IP and administrator credential to get authenticated.

Once associated, the vCenter inventory will be discovered with VM facing DVS Portgroup.

Go to App_Servers EPG, under operational tab. Add VMM domain association. Perform the same for Web_Servers EPG.

Figure 15 Add VMM domain association

Login to vCenter and add the VM to the portgroup populated from ACI.

Figure 16 Add VM to ACI portgroup

Figure 17 Assign portgroup policy to VM

Console to the WebServer and try to ping the gateway 192.168.10.1

```
CentOS release 6.4 (Final)
Kernel 2.6.32-358.el6.x86_64 on an x86_64

localhost login: student
Password:
[student@localhost ~]$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data:
64 bytes from 192.168.10.1: icmp_seq=1 ttl=58 time=1.97 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=58 time=0.246 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=58 time=0.228 ms
^C
--- 192.168.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2477ms
rtt min/avg/max/mdev = 0.228/0.814/1.970/0.817 ms
[student@localhost ~]$ _
```

Figure 18 Test ping from webserver to gateway

Try to SSH from web-server to app-server.

```
[student@localhost ~]$ ssh 192.168.11.102
ssh: connect to host 192.168.11.102 port 22: Connection timed out
[student@localhost ~]$ _
```

Figure 19 SSH test from web-server to app-server

Immediately, the app-server refuses the connection. That means the packet actually made it to the App_Server. SSH TCP connection with destination Port 5000 was allowed by the

contract to reach App_Server, while the default port 22 was blocked since it was not listed on the contract. This shows ACI whitelist policy will disable all the communication by default and allow traffic only defined in the contract to make sure hardened security.

In summary, this section showed how easily to enforce security policy via contract and manage VM, Infrastructure and security under the single dashboard.

Part 4: ACI and Firewall Integration

By now, a good understanding is established on how ACI whitelist works, it is essentially a L3/L4 firewall. In the next part of the project, lab will be setup to leverage on firewall to provide higher level protocol inspection. In a typical IT operation, deployed applications need to be patched on a regular basis. The following scenario will be simulated in the lab:

- Only trusted users can access the patch host.
- Patches can only be staged on the patch host, which required to use the patch host to access Internet to pull the required updates, etc.
- Only patch server can upload the files to the repo server.
- All Web traffic from the patch server towards Internet (0.0.0.0/0) must be inspected by the firewall, to prevent access to any malicious web site.

Task Summary

Implementing L4-7 in ACI with following steps:

- Import device package
- Define L4-7 policy-based redirect policy
- Define function profile
- Define device
- Define service graph

- Deploy service graph

This task aims to show besides built-in security features, ACI also supports integration with L4-7 firewall. Similar to server and network team challenge, that's a lot organizations have separate security and infrastructure team. ACI assists simplifying security policy execution by using up to date firewall packages.

Step 1: Import device package

Device package is required to insert and configure the network service functions, it contains the following to enable APIC interaction with the service:

- Device functions
- Parameters that are required by the device to configure each function
- Interfaces and network connectivity information for each function

To deploy a device package, logon to ACI dashboard, go to L4-7 Services, then Packages.

Click on Import a Device Package to add the firewall device package.

Step 2: Define L4-7 policy-based redirect policy

To enable policy-based redirect (PBR) on ACI. Following need to be considered:

- PBR only supports L3. i.e. both provider and consumer EPG must be of L3 reachable.
- A dedicated bridge domain is needed to facilitate traffic forwarding between ACI fabric and firewall.
- The Service node's IP/Mac information must be explicitly defined in ACI's PBR forwarding policy. This IP/MAC address information need to be consistently defined on the firewall function profile.

Step 3: Define function profile and required bridge domain

Logon to ACI, go to Networking, then click Bridge Domain. Right click and select Create a Bridge domain “PBR”, following attributes are required:

Table 2 Funtion profile and bridge domain configuration

Subnet	10.0.0.x/30
VRF	SecureDC
L2 Unknown Unicast	Flood
L3 Unknown Multicast Flooding	Flood
Multi Destination Flooding	Flood in BD
Arp Flooding	True
End Point Learning	DON'T CHECK

Step 4: Define PBR Policy

To create the PBR forwarding policy, go to Policies, followed Protocol and L4-L7 Policy based Redirect. Right click to create a new PBR policy.

On the PBR config window, enter the following information:

Table 3 PBR Configurations

Name	Firewall-Redirect
IP	10.0.0.1
MAC	00:50:XX:XX:XX:XX

A function profile specifies policies on a service node. For the design, the firewall needs to be provisioned in one arm mode.

Create a new function profile. Go to Services, then L4-7, followed by Function Profile. A few parameters need to be defined here:

Table 4 Function profile for firewall config

Name	InspectWeb
Profile	Cisco-ASA/WebpolicyForRouteModelIPv4
Interface	Only 1 interface is needed
From Device Config	Delete Internal interface folder
From Function Config	Delete internal interface folder

Then fill up IP address and MAC address of the interface. Define static routing, forward all 0.0.0.0/0.0.0.0 to the ACI Fabric. On the same window, go to Device Config, click Interface Related, then Static Route List, lastly IPv4 Route.

The service node performs the L4-7 function. These service nodes need to be added as resources in ACI.

Step 5: Define service node

The service node behaves in a stateless manner. Policies are applied when the service node is used in a contract. Service node restore back to its initial configuration when the service graph is un-provisioned.

To add firewall device, right click on Services, then L4-7, click Devices to add firewall.

Step 6: Define service graph

The service graph is a template, defining how traffic is handled between the consumer and the provider. To add a service graph, right click Services , use Service Graph Templates to add a service graph. On the Service Graph Template. Enter the following fields and drag drop firewall icon onto the middle canvas.

Table 5 Service graph template settings

Name this service graph	Firewall
Profile	FW-Inspect (This function profile was created in step 3)
Route Redirect	Checked

Step 7: Deploy service graph

The scenario is to allow only trusted user to have RDP access to the Patch Server. All FTP and HTTP traffic between the trusted user and patch server need to be inspected by the firewall.

The next is to deploy the service graph to allow HTTP/FTP traffic between the trusted user and patch server.

To deploy the service graph, right click on Services, then Service Graph Template, go to Firewall. Select Apply L4-7 Service Graph Template.

On the Apply L4-7 Service Graph Template, define the following:

- Consumer EPG: Patch
- Producer EPG: External
- Create new contract, to only allow HTTP and FTP. With this policy ACI will only forward HTTP, HTTPS traffic to firewall.

The last step is to define specific policies on firewall. Once finished, deploy the service graph. Result has been verified that patch host can reach to external internet www.cisco.com.

In the part 1 of the project, policies have been specified that certain traffic (eg ICMP, http) can pass between two endpoints, other traffic such as FTP, which required L4/7 inspection will be redirected to pass through a firewall. This approach offers great flexibility in optimizing the valuable firewall resource for only interesting traffic to be inspected. This offers a much more scalable and cost effective, compared to the traditional approach, where all traffic must pass through the firewall.

Part 5: Micro-segmentation

In the ACI policy model section, whitelist model has been discussed by using contract between EPGs. Micro segmentation enables virtual workload of similar attributes, such as VM names, cluster name, OS type, to be grouped as individual security domain. E.g. All Sales workload belong to one security group, all HR workload belong to another security

group, and no communication is allowed between Sales and HR unless contracts are defined.

Under Web, App EPG, further segmentation can be achieved by using the following attributes while defining the new EPG.

Web EPG

- Bridge Domain = common_BD
- VMM Domain = Pod_Mseg
- Enable Microsegmentaiton
- Encapsulation Mode = VxLAN

App EPG

- Bridge Domain = common_BD
- VMM Domain = Pod_Mseg
- Enable Microsegmentaiton
- Encapsulation Mode = VxLAN

Then add RDP as a provided contract for both EPG.

In this project, Micro segmentation will be used to isolate HR and sales workloads. The following diagram illustrates the micro-segmentation logic.

Figure 20 Micro-segmentation Logic Diagram

First step is to define micro segmentation for all workload from HR. From the Development application profiles, right click uSeg. Select uSeg.

Figure 21 Micro-segmentation APN configuration

Then on the micro-segmentation window, define the following:

Create USeg EPG

STEP 1 > Identity

Specify the EPG Identity

Name:

Alias:

Description: optional

Tags:

QoS class: Unspecified

Custom QoS: select a value

Intra EPG Isolation: Enforced Unenforced

Preferred Group Member: Exclude Include

Flood on Encapsulation: Disabled Enabled

Bridge Domain: select a value

Monitoring Policy: select a value

Epg Match Precedence: 0

Figure 22 Create USeg EPG

- Name : HR
- Bridge Domain : Common-BD
- Domain : Pod-Mseg
- Encapsulation Mode : VxLAN

For this new micro segmentation EPG, expand the uSeg Attributes. Add the attribute of VM

Name begins with HR.

Tenant SecureDC

Application Profiles

Development

Application EPGs

uSeg EPGs

HR

Domains (VMs and Bar...)

Static Leafs

Contracts

Static Endpoint

uSeg Attributes

Match Any

VM - VM Name Starts With HR

Figure 23 Define Micro-segmentation policy with VM name starts with HR

After applying the micro-segmentation policy, check the operational tab of HR uSeg EPG.

Observed that all HR workloads are reflected here.

HR VMs need to provide RDP session to the external users and consume core services, e.g.

DNS, Authentication, etc. Define contracts to enable this requirement.

Repeat the same step to create the Sales uSeg EPGs.

The ping test shows the micro-segmentation has been implemented as what the logic diagram illustrates.

This part of the lab shows how easy to design, plan and implement the micro-segmentation to further enhance the security in a real enterprise network with multiple departments without completely changing the EPG web, app contracts. The micro-segmentation can be triggered easily by using VM attributes, OS attributes etc.

In summary, contracts, L4-7 service integration and micro-segmentation has been reviewed and implemented to secure a data center network in ACI environment. Traditionally, it would involve at least 3 teams: server, infrastructure and security to achieve the same result. In this project, it has been done in single ACI dashboard with a few clicks. It not only enhances the security of data center, but also simplified the operation and management. And potentially shorten the application time to market by eliminating confusion between different teams talking in different languages. This lab experiment builds the foundation of ACI constructs for enhanced features such as automation in the following part of the report.

Project 2: Automating Fabric

Another key driver of software defined networking is automation. There are different reasons customers seeking automation in their data center:

- Simplify to deal with increasing number and scale of technology complexity
- Make technology consumable with service catalogue or cloud-like service for faster time to deliver application
- Reduce cost of OPEX, human errors, change windows and scale elastically

Mckinsey study conducted in 2016 (Michael Cooney) shows the current operating model are not working with numbers:

- 95% of network changes performed manually
- 70% policy violations due to human errors
- 75% Opex spent on network changes and troubleshooting

That's billions of dollars spent on network operations labour and tools, yet solve the problem.

Automation could mean different things to different audience. For a network administrator, automation could be avoided failure. Regular change will be risky and complex to manage which increase accountability. For a developer, change is good with embracing failure.

Active collaboration empowers accountability and consistent feedback system.

To meet the requirements of different stakeholders of an IT team, this part of report focus on discussion of two automation approaches:

- Automating fabric by API
- Automating fabric by orchestration tool

There are various kinds of automation tools available in the market such as Puppet, Chef.

However, those tools require either skillset of its own language or installing agents on the servers, switches in the DC. They are not built-in solutions and not open standard. Thus, this report focuses on how easy to leverage API and scripting to automate ACI and reduce replicated tasks. And out-of-box workflows to accomplish tasks may require multiple teams (eg compute, network, storage teams) in the past.

Part 1: Automating Fabric by API

Besides GUI, that's two more ways ACI can be configured: CLI and REST API. CLI works as last resort mainly for troubleshooting purpose.

REST API is taking advantage of xml or json based scripts which provides another way to automate and accelerate configuration deployment. Cisco ACI's API drastically reduces the time to deploy both greenfield and brownfield networks by using scripts. It makes roll out faster especially for repeated works such as configure hundreds plus vlan during migration phases.

The ACI controller provides a northbound Rest API. Network statistics, faults, events and configuration can be read from the API. The ACI API applies the HTTP GET, POST and DELETE calls.

A simple ACI script starts with POST login:

```
Method: POST
URL: https://<ip address of APIC>/api/aaaLogin.json
Body:
{
  "aaaUser": {
    "attributes": {
      "name": "admin",
      "pwd": "password"
    }
  }
}
```

Figure 24 ACI script for controller authentication

ACI also provides built-in tool API Inspector for effectively using it as a “GUI to JSON” translator.

In the first part of project 2, the following tasks have been performed to evaluate ACI configuration of using POSTMAN and API inspector programmability.

1. Perform a task using GUI such as create an User and then delete it
2. Capture then create a JSON script using the API Inspector output
3. Perform the same task (create the same user and then delete it) using the JSON script and Postman

To access API inspector, that's a drop-down list under Admin option in upper right corner in the dashboard.

The screen will be quickly filled up on the API inspector screen because it captures all levels of APICs communication by default, which includes all POST, GET and DELETE commands.

Those POST, GET and DELETE commands are at debug level. Thus use 'Filter' will help to display effected command.

Leave the API inspector window open then move to POSTMAN for the rest of tasks.

In the POSTMAN, on the right hand side is the content of the Login script showed in previous screen capture. This script will be used for authentication with the APIC. That's the common POST terms:

- **url** is where in the Management Information Tree (MIT) to impact the object with the Login Script.
- **Method** is what user want to do with the object (POST, GET and DELETE are supported).
- The content of the message is **Payload/Body**, what and how user want to create/modify the object.

url, method and payload/body are the information obtained by API inspector earlier. Then leave POSTMAN open and return to ACI for the rest of actions.

By creating and deleting a single user using GUI, the equivalent JSON scripts will be captured by the API Inspector.

Creating and deleting user can be achieved with following steps in the GUI:

- Click on ADMIN button on top right in the dashboard and select drop down list AAA option.
- Expand Security Management, right click on Local User and select Create Local User.
- Type username as user name, and password as password.
- Select all for security domain then click next to move to the next page.
- Define roles of the user.
- Then simply expand Local users, right click the user created just now and select delete.
- Confirm the operation with click Yes button.

The actions performed on GUI should be captured by API inspector, to filter the irrelevant operations, narrow down the scope with POST only commands.

The first output will be POST command of creating user. As discussed earlier:

- url which describes where to place the object (new user and its properties)
- payload which tells APIC what the characteristics of the object are

method: POST

url: https://<APIC IP address> /api/node/mo/uni/userext/user-local.json

payload

```
{"aaaUser":{"attributes":{"dn":"uni/userext/user-local","name":"username","pwd":"password","rn":"user-local","status":"created"},"children":[{"aaaUserDomain":{"attributes":{"dn":"uni/userext/user-local/userdomain-all","name":"all","status":"created,modified"},"children":[]}}]}
```

Figure 25 Create user script

method: POST

url: https://<APIC IP address>/api/node/mo/uni/userext.json

payload

```
{"aaaUserEp":{"attributes":{"dn":"uni/userext","status":"modified"},"children":[{"aaaUser":{"attributes":{"dn":"uni/userext/user-local","status":"deleted"},"children":[]}]}}
```

Figure 26 Delete user script

To make the script easier to read in the production environment, that's online formatting tool available to beautify the payload part.

Then go to POSTMAN to perform login authentication first. Status '200 OK' message should be received after successful login.

After login, create new Tab and enter the create user corresponding url and payload. Select JSON raw format for the Body. Send request to APIC. And verify the new user 'username' has been created in the APIC admin dashboard. Action has been performed for delete user with verified result on the dashboard.

Throughout the whole process, basic programming tasks have been performed and verified by using built-in API inspector together with POSTMAN. Though it takes effort to write the scripts for the first time, once it is done, batch jobs can be achieved by running the scripts in much reduced time.

Comparing with traditional way of CLI configuration, all switches are managed and operated as independent entities which creates heavy operation a burden. Moreover, unlike compute operation, where modifying one server is unlikely to impact other servers, network device

settings are very topology dependent. Some things such as syslog, bandwidth etc need be configured the same on all devices and are easier to automate through scripting.

Part 2: Automating Fabric by orchestration tool

The tool has been evaluated in this report is unified computing system director (UCSD). It provides off the shelf workflow with full support of ACI. There are 250+ out of the box task available and end user portal for catalogue consumption. It supports both Cisco and non-Cisco products including computing, network, storage and virtualization. That's also extensive northbound API for further customization.

Task has been performed in this report is tenant onboarding configuration, then creating 3-tier application with multiple compute instance at each tier, as this task is known for time consuming and prone to human errors.

With pre-built catalogue workflow, only tenant name and user credential information need to be filled up to create a new tenant.

The screenshot displays the Cisco UCS Director in dCloud interface. The top navigation bar shows 'Catalogs' and 'Catalogs'. Below this, the 'Create Service Request' section is active. A green status bar at the top of the workflow area indicates 'Last Validation Status of the workflow: OK'. The workflow steps on the left are 'Catalog Selection', 'Custom Workflow', and 'Summary'. The 'Custom Workflow' step is currently selected, showing 'Custom Workflow Inputs'. The inputs are as follows:

Field	Value
Tenant Name*	Tenant_Automation
Tenant Admin Email*	yx002@e.ntu.edu.sg
Tenant Admin Username*	testuser
Tenant Admin Password*	

Figure 27 Tenant onboarding workflow of UCSD

After that, use the application with vmware workflow to define the application name and how many instances at each tier and select the tenant has been created in the step above.

Cisco UCS Director in dCloud

Catalogs | Catalogs

Create Service Request

Progress: Catalog Selection, Custom Workflow, Summary

Last Validation Status of the workflow: OK

Custom Workflow Inputs
If applicable, specify workflow input values

Application Name*	App
Number of Web Tier VMs*	5
Number of App Tier VMs*	3
Number of DB Tier SPs*	0

[+] Select APIC Tenant* (Tenant_Automation)

Figure 28 Create 3 tiers application by using UCSD workflow

Then a 3 tier Application will be deployed on ACI network at the backend.

Cisco UCS Director in dCloud

Catalogs | Catalogs

Create Service Request

Progress: Catalog Selection, Custom Workflow, Summary

Summary
Review information below and click Submit to initiate the service request

Group	Default Group
Catalog	3 Tier App with VMware and UCS
Catalog Description	3 Tier App with Web and App VMs, and a UCS Blade hosting DB Services
Content Library VM Template	
Port Groups	
VM Networks	

Figure 29 Verify parameter and execute task on ACI

After filling up the predefined and required parameters on UCSD, tasks will be executed on ACI without human innervation on ACI interface. By using automation tools, it shortens the task from 14 times repetitively to create the constructs for compute instances to one click.

Workflow can also be issued for patches without missing a single step. Operation can also be revoked easily with roll back option without affecting other parameters.

In this part of project, orchestration tool shows its advantage with out of box service catalogue with customizable workflow. Its scope is not only limited to network fabric itself but extends to computing system, storage etc. Complex task involving multiple IT teams can be performed with a predefined workflow with filling up necessary parameters and minimize the risk of human errors. It shortens the process to roll out new service and improve the efficiency and accuracy.

Project 3: Transformation of Financial Network to be Digitalization Ready

This part of report focuses on how to assist a financial customer move from traditional network to ACI to achieve automation and agility in real world deployment. Data center network for financial sectors are often characterized as highly available, secure and compliant, as it deals with most sensitive, privacy data. With the concept of digital mobile banking, it also requires constant upgrade to meet consumer needs and extensive reachability. If one solution could fulfil financial customers' needs, it will become a good reference and work for most other customers.

Migration Planning

ACI Physical Hardware Considerations

Before the actual deployment, check against verified scalability guide is always a good practice to make sure the deployment will not hit any physical scale limit. And prepare the following information in hand to make sure the project will not be delayed due to lack of information.

Fabric Setup

- Prepare the Infra VLAN IP scheme
- Make sure enough VTEP IP pool size
- Make sure no IP space overlap (OOB, in-band)
- Setup common interface policies eg LLDP, LACP, port speed etc

Spine switches

- Modular or fixed spine switches? Modular spine has larger mapping database and storage capacity, but it might not be needed for all the deployment
- What will be the endpoints: VM, vNIC, hypervisor, storage etc
- Review software version, maintenance and upgrade strategy, make sure the deployment will not hit known bug

Border leaf

- Number of external EPGs, ingress/egress policy options, this will help on the physical sizing of leaf switches
- Review route summary and redistribution strategy

Compute and storage leaf

- Number of endpoints and how to optimize the sizing, such as virtualization or group similar endpoints to the same rack
- LLDP/CDP compatibility, spanning tree best practices
- That's limited TCAM table on the switch, a deployment with proper design will not hit the limit

Data Center Migration Considerations

In a general DC migration, network, compute and storage, application and L4-7 services are always the key considerations. The project later will discuss in detail how ACI addresses each consideration in an automated fashion.

Network

- Layer 2 innovations, how L2 extends and spanning tree challenges could be addressed
- Gateway mac move seamlessly
- Out of band and in band management network
- What would be the size of vMotion domain
- How QoS, multicast and jumbo MTU will be handled in the network
- Will IPv6 be part of the migration

Compute and storage

- Bare mental and virtualized workload
- Blade or rack servers
- Which kind of hypervisor integration
- Which kind of cloud integration
- What are the orchestration and automation tools

Application

- How to group application components into EPGs concept
- How to achieve IP addressing flexibility
- How to address L2/L3 multicast for clustering
- What would be the security requirements

- What would be east-west traffic and latency requirements

L4-L7 services

- How to achieve services integration
- What will be the operation mode – transparent, inline, routed, one-arm
- Will it be a multi-tenant deployment and stitching
- How about the clustering and multi-dc support

ACI Migration Approaches

There are two migration approaches in ACI: network centric mode and application centric mode.

In a typical network centric deployment, the following mapping will ensure a smooth migration from traditional data center network to ACI:

- Every VLAN maps to one BD maps to one EPG, and BD may be configured with flooding enabled.
- Usually VRFs are not used in enforced mode which means permit any-any communication at VRF level. This is replaced EPG level of policy implementation.
- ACI may not be the default gateway and follows previous gateway on firewall or router etc.
- Bridge domain may or may not have an IP subnet.
- Integration with L4-7 services is done in traditional way such as routing or bridging mode.

In contrast, a typical application centric deployment usually has the following characteristics:

- With smaller number of bridge domain and no flooding and extending L2 outside fabric is required.

- Multiple EPGs map to single BD.
- VRFs are set to enforced mode, so that any traffic between EPGs requires contracts
- L4-7 service integration by using managed or unmanaged service graphes
- Smaller number of IP subnets

To make sure non-disruption to the bank's highly critical production environment, this migration follows network centric approach in phase 1 and will move to application centric mode once workload fully migrated to ACI.

ACI Migration Deployment Considerations

Fabric Bring up

ACI fabric itself supports auto discovery, inventory and system maintenance, monitoring processes via the controllers.

Once boot up all the devices including spine, leaf switches and APIC controllers, the entire fabric discovery is through LLDP automatically and progresses as administrator registers the switches to join the fabric. APIC will discover the first physically connected leaf, before continuing or discovering any other devices, the detected leaf node ID and name must be configured first.

Once the fabric has been built, the day 2 operation such as commissioning, decommissioning, firmware management and life cycle management of switches will all be thru controllers.

Figure 30 ACI zero touch fabric bring up

Cisco.com. 2020. [online] Available at:

<<https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/software-defined-access/white-paper-c11-740585.pdf>> [Accessed 10 April 2020].

EPG Identification

In the earlier discussion of ACI building constructs, EPGs is the one providing a logical grouping for objects that require a similar policy. In a typical network centric migration, EPGs follows VLAN ID one to one mapping, then EPGs map to BDs where gateway address residents. In an ACI fabric leaf, a layer 2 bridge domain has been created with number 100 and gateway address 10.1.1.1/24 for all the servers, VMs. When placing a set of VMs on the ESXi host with vSwitich. The VM1 to VM10 communication needs a port group defined with a VLAN ID. In this case, it is VLAN 101, when this information enters the ACI leaf, the EPG to VLAN mapping table will know it belongs to EPG1, then goes to the BD gateway. Several other EPGs with respective VLAN mapping can be created under the same BD. Keeping doing so and applying policies to define the communication between EPGs implies to VMs is where the constructs and policies meet together.

Figure 31 ACI EPGs Identification using VLAN IDs

Proof of Concept Lab Environment

The proof of concept migration lab environment consists of a minimum number of devices needed to test the various functionality to fulfil the customer's requirements. The environment consists of:

- 2 ACI spine Nodes
- 2 ACI leaf nodes
 - Leaf and Spines connected via 40G ports
- 1 APIC (only for lab environment, minimum 3 APIC controllers are recommended for production environment)
- 1 L2 Load Balancer
- 2 Core Switches (Cisco Nexus 7000) running OTV VDC
- 2 L3 router running VDC on Cisco Nexus 7000
- Bare metal hosts for testing
- Trunk host for hypervisor simulation

Devices and port attachment is listed in the following table:

Table 6 PoC devices and port attachment mapping

Port	Connected to	Mode	VLAN
1	L3 Core Router	L3 untagged	2501
2	OTV – N7000	Trunk	
3	Hypervisor	Trunk	
5	L2 Port Channel	Trunk	
6	L2 Port Channel	Trunk	
7	Bare metal host Tenant 3	L2 Untagged	1215
8	Bare metal host Tenant 2	L2 Untagged	1199
9 Leaf 1	L2 LB outside	L2 Untagged	2509
9 Leaf 2	L2 LB inside	L2 Untagged	2509
10 Leaf 1	L2 LB appliance Inside	L2 Untagged	2510
10 Leaf 2	L2 LB Host – inside	L2 Untagged	2510

The lab topology is illustrated in the figure below.

Figure 32 Lab Topology

Requirements

The following is a list of the connectivity requirements that determine how devices are attached directly to and communicate at L2 through the ACI fabric.

- Existing VLAN numbers need to be retained when connecting L2 and hypervisor environments

- All devices on existing VLANs are to be able to talk to each other without limitation
- Hosts in different subnets are to be able to talk to each other without limitation
- Work within limit of an ACI tenant having a max of 500 EPGs
- The local gateway has to merge with the existing HSRP configuration at remote DC

The ACI fabric will act as the default gateway to the attached devices. However, it is required that the fabric connects to external networks via L3 Router adjacencies. In the final design, the existing core network switches will be connected to, however, during the lab the existing aggregation layer switches were peered with. The requirements for L3 connectivity are

- Connect fabric to existing L3 environment
- Configure OSPF to advertise routes both to and from fabric
- Interoperate with the remote DC that was advertising a HSRP router as the default gateway

Solution

Overlay design principals

In the final design each business unit was assigned a tenant. Each Tenant will be allocated the VLANs already assigned to the BUs. Here is the list of considerations:

- There are approximately 100 VLANS per BU.
- There will be for each existing VLAN a dedicated EPG and BD created under the relevant tenant. Existing VLAN will, regardless of next hop device, be associated with a dedicated EPG.

Thus, the design will be a static mapping for a given VLAN to EPG configured at the node level. The EPG will be associated with the BD. Each BD will have only one IP subnet. The

fabric is to be anycast default gateway for each local subnet. The common tenant will have a shared L3 VRF created. Inherently yes via relationship between VRF-> BD->EPG. And initial contract will be permitting any IP traffic.

Underlay design principals

For underlay design, all VLANs will be able to run any switch ports and ports will be groups based on function of direct L2 access, VPC and L3 route peering.

Standard port policies will be created and shared across the underlay. A separate domain will be created for direct access device and L3 external routing. For tenants' design, all tenants will have unrestricted access to each domain. The Common tenant holds the routing table for all tenant BDs. External router peering is via Common tenant L3Outs between one leaf and one upstream router and this could be configured as a mesh in final design for more granular load balancing and HA. The individual BDs and subnet IP addresses for the tenant were configured for redistribution. The routing updates will be received/sent on both OSPF peering. At last, the fabric was configured to enable route propagation with the fabric via BGP

Implementation

Layer 2 Network

In order to achieve the L2 connectivity requirements, first step is to create a static VLAN pool for VLANS 1000 to 2999. A physical domain was created and associated with the VLAN pool. After that, all EPGs were later associated with this domain. Till now, no Security domain was configured at this stage yet.

To setup the interface connectivity and to be reused later, policies were created as below:

- Link Level interface- 1G and 10G
- CDP

- LLDP
- L2 Interface policy was left at the default per switch VLAN scope – not per port scope

After completing the above, two access port policy groups were created to that were linked to the AEP: One was for 1G interfaces the other 10G interfaces

A leaf interface profile was created with an Interface Selector that included ports 2- 4 and 7- 8 and associated with 1G access port policy group. The leaf interface profile then associated to a leaf switch profile which includes both leaf nodes was created and associated with the leaf interface profile.

In the common tenant a “Core-VRF” was created. A contract was created in the common tenant that allowed all traffic i.e. ip-any. For PoC purpose, three separate Tenants were created named “01”, “02” and “03”. After creating tenants, BD/EPGs were created with each tenant based on the user provided requirements. To follow network centric deployment, each BD had a single subnet created and each tenant BD was associated with the Core-VRF. The BD was given two MAC addresses: virtual Mac that was the same as the peer HSRP address. Then EPG was associated with peer BD. The last step is to map given VLAN to to each EPG by a static Leaf mapping.

Layer 3 Network

A L3 domain was created and associated with the existing VLAN pool and AEP with the following steps:

- Interface 1/1 on both leafs were added to the existing leaf Interface profiles
- In the Common tenant two separate L3Outs were created o for each OSPF Process
 - area 0
 - MTU 9000

- Set security MD5 password
- Set to point to point mode
- Set router id to same as interface address expect preplaced 3rd octet with 0
- Created SVI
 - Associated with interface 1 on either leaf 1 or 2
 - Mapped to VLAN 2501
- Single Network EPG of 0.0.0.0/0 -- associated with ip-any contract in both provide and consume direction
- Associated with L3 External domain
- For Each tenant BD, associated with both L3Outs

Project Summary

After completing the PoC, 3 business units of the financial customer were able to move to new ACI network with very minimum downtime and communicate with existing network. The policy constructs were applied to corresponding business segments and map to traditional network concepts with lesser restrictions. In the past, it usually takes a few weeks for the network and application team to discuss vlan, acl, compute instances for develop one feature in the application. After moving the ACI network, it shorten the process to days and use the automation tools to setup the environment in few minutes. The ACI network unlocks the customer for more innovations and empowers the applications agility with business moves.

Conclusion and Future Work

With the still 'hard to define' SDN term, this research aims to solve the difficulties that IT department facing on their daily routine. What my perception of SDN is not the network is truly automated by software, but the network really helps to offload the network operations and remove the barrier of application and network teams.

The lab shows Cisco ACI network is able to fulfil various network requirements of large enterprise IT operations with automated configuration and simplified workflow. It is proven to accelerate application time to market, make IT more business relevant and accelerate customer's digital transformation.

After evaluating the network automation on the data center domain, further research can be extended the fabric to the campus network or Wide Area Network (WAN) to carry the policies across geo locations and establish end to end visibility, security and management.

Reference

En.wikipedia.org. (2020). OpenFlow. [online] Available at:

<https://en.wikipedia.org/wiki/OpenFlow> [Accessed 13 Feb. 2020].

Krishnan, Ramki, and Norival Figueira. "Analysis Of Data Center SDN Controller Architectures: Technology And Business Impacts". IEEE, 2015 International Conference On Computing, Networking And Communications (ICNC), 2015.

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., . . . Wright, C. (2014, August). Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. Retrieved from <https://tools.ietf.org/html/rfc7348>

Locator ID Separation Protocol (LISP) VM Mobility Solution White Paper. (2011). Retrieved 27 July 2019, from https://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/locator-id-separation-protocol-lisp/lisp-vm_mobility_wp.pdf

Naranjo, Edison, and Gustavo Salazar Ch. "Underlay And Overlay Networks: The Approach To Solve Addressing And Segmentation Problems In The New Networking Era". IEEE, 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), 2017.

Cooney, Michael. "Cisco CEO: 'We Are Still Only on the Front End' of a New Version of the Network." Network World, Network World, 16 Feb. 2018,

www.networkworld.com/article/3256264/cisco-ceo-we-are-still-only-on-the-front-end-of-a-new-version-of-the-network.html.

Committed to connecting the world. (n.d.). Retrieved from
https://www.itu.int/net/pressoffice/press_releases/2015/17.aspx

Al-Qahtani, M., & Farooq, H. (2017). Securing a Large-Scale Data Center Using a Multi-core Enclave Model. In 2017 European Modelling Symposium (EMS). Manchester, UK, IEEE.