ACI Guidelines

Guidelines for Logical Construct Usage

Date: 14th August 2019

Document Control

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Release Date | Reviewed by | Summary of change |
| 1.0 | 14/08/19 |  | Initial Draft |
| 1.1 | 22/08/19 |  | Update Background section.  Update VRF section.  Update IP addressing section.  New sections on VLANs, BDs & ERCs. |
|  |  |  |  |

Issue Control

|  |  |  |  |
| --- | --- | --- | --- |
| Owner | Author | Approved By | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Function | Date of Issue | Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

Document Control 2

Overview 4

Introduction 4

Background 5

Firewall Guidelines 6

Tenant/MSO Guidelines 7

VRF Guidelines 8

Bridge Domains 9

VLANs 10

EPG Guidelines 11

Contract Guidelines 12

ERC 13

IP addressing 14

# Overview

## Introduction

The purpose of this document is to document guidelines for ACI logical construct usage.

This document is a work in progress.

## Background

The Cisco Application Centric Infrastructure (ACI) allows application requirements to define the network, abstracting the traditional concepts of networks, VLANs, firewalls and other services.

The policy model introduces the following logical constructs:

A **Tenant** is a logical container for application policies that enable an administrator to exercise domain-based access control.

A **BD** (Bridge Domain) is used to limit the Layer 2 broadcast domain. In this regard they are similar to VLANs but the terms should not be interchanged.

The **EPG (**Endpoint Group) is the most important object in the policy model and is defined as a collection of servers that provide or consume a common service.

ACI by default operates a Whitelist model. Communication between servers within an EPG is permitted. Communication between EPGs is implicitly denied unless permitted by the use of a **Contract.** Contracts are like firewall rules but are stateless. In doing so ACI provides segmentation ‘out-of-the-box’ able to place any single server in its own ‘security zone’.

The model dictates that logical constructs are contained within each other:



The flexibility of ACI gives rise to two implementation methodologies – Network Centric or Application Centric. In summary:

Network Centric – the ACI logical constructs are used to represent the existing network constructs. This equates to:

BD = EPG = VLAN = Subnet.

Application Centric – The ACI logical constructs are used to represent individual applications and application components providing per-application segmentation and control of access to, from and between these components. This methodology requires a good understanding of data flows between the different components.

In reality the implementation will be a hybrid of these two approached.

# Firewall Guidelines

Firewalls will still be used to protect the perimeter and provide deep packet inspection where required.

1. All servers that are to be protected by a firewall i.e. on a DMZ, will be placed in the Net-centric firewall tenant.
2. North/South traffic flows will traverse a firewall. E.g. a server that terminates external connections (i.e. Web, Gateway server) will reside on a DMZ. Connections inbound to an ‘application’ server will also traverse the firewall.
3. East/West traffic flows will be restricted by a Contract. E.g. ‘Application’ server to ‘DB server’

A ‘DC’ firewall will be deployed and traffic will be directed to this firewall using PBR Contracts (See Contract section). Cases where this is deemed necessary should be identified by an additional “Network segmentation IT security policy” where network segment should be identified and the appropriate level of inspection (e.g. NG FW or ACL) should be specified.

It is proposed that the ‘DC’ firewall will be used in the following circumstances:

* When restrictions to access servers should be based on user identity rather than (source) IP addresses
* When communications require deep inspection (e.g. HTTPS inspection) and/or time restrictions.

# Tenant/MSO Guidelines

Tenant structure as per accompanying diagram.

1. Net-centric tenants for DMZs and ‘Servers’.
2. App-Centric tenants for each of the non-Prod environments. All current non-Prod environments will be consolidated into the following four: DEV, QA, Test & Stress.
3. Single App-Centric tenant for Prod environment. The current approach, which suggests using a single EPG to encompass all components that comprise an application, means it is unlikely to hit the 512 EPGs/Tenant limit.
4. Separation of non-Prod and Prod environments provides a very clear change domain.
5. In the MSO – One Schema per Tenant.
6. Two templates in MSO – 1 per site.
7. The default ‘common’ Tenant will be used for all common infrastructure services, that is, services that supply a service to applications, i.e. AD, DNS.
8. The default ‘mgmt’ Tenant will be used for all services that are used for managing the infrastructure i.e. Pure storage management, vCenter servers.

# VRF Guidelines

DMZ Tenant will use vrf ‘DMZ’. This will be a different VRF per firewall domain i.e. DMZ.TRAD\_VRF, DMZ.INET\_VRF, DMZ.B2B\_VRF. This will prevent the potential bypassing of the firewalls should multiple errors be made within the APIC configuration.

All other Tenants to reside in vrf CORPNET\_VRF.

# Bridge Domains

Bridge Domains (BD) within ACI are used to limit the Layer 2 broadcast domain. In this regard they are similar to VLANs but the terms should not be interchanged.

BDs can either be Layer 3, where they are configured with an IP address (effectively an SVI), or Layer 2 where no IP address is configured. The latter requires and alternative default gateway for the network (i.e. a firewall) or understanding that the BD is non-routeable.

Naming conventions:

* 1. Application Centric - *DCname*.*TenantName*.*n*\_BD

Examples: ARK.DEV.01\_BD

LD5.UAT.01\_BD

* 1. Network Centric - *DCname.Environment.NetCentricTenant.IPaddr3rdOctet*\_BD

Example: ARK.PRD.SVRS.32\_BD

LD5.NONPRD.SVRS.33\_BD

ARK.PRD.DMZ.128\_BD

LD5.NONPRD.DMZ.129\_BD

1. Hybrid - *DCname.Environment.ApplicationGroup*\_BD

These will be used in the ‘Common’/’mgmt’ tenants where similar Applications will be grouped to avoid unnecessary repetitive use of Contracts.

Examples LD5.PRD.COMMON.SVCS\_BD

ARK.PRD.INFRA\_MGMT\_BD

# VLANs

Within ACI, VLANs are used to provide the separation between EPGs. VLANs are allocated in Pools for different purposes – some being statically allocated and others dynamically allocated.

VMM Dynamic VLAN pools:

* PRD.DV.ARKPINFFVCA01.M\_VPL – v2700-2799
* PRD.DV.ARKPINFFVCA01.CO1\_VPL – v2800-2899
* PRD.DV.ARKPINFFVCA01.CO2\_VPL – v2900-2999

These will be used in VMM Domains and deployed in the following Tenants for VMs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **DEV (AppC)** | **QA (AppC)** | **UAT (AppC)** | **STRESS (AppC)** | **PROD (AppC)** |
| **VLAN pool**  **(VMM)** | 2700-2799 (M)  2800-2899 (C01)  2900-2999 (C02) | 2700-2799 (M)  2800-2899 (C01)  2900-2999 (C02) | 2700-2799 (M)  2800-2899 (C01)  2900-2999 (C02) | 2700-2799 (M)  2800-2899 (C01)  2900-2999 (C02) | 2700-2799 (M)  2800-2899 (C01)  2900-2999 (C02) |

Application Centric VLAN pools:

* PRD.APP\_VPL - v1400-1499

This will be use for Application Centric Bare Metal Servers.

If a VM and a Bare Metal server need to be in the same EPG then it may necessary to extend this static pool to the vCenter.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **DEV (AppC)** | **QA (AppC)** | **UAT (AppC)** | **STRESS (AppC)** | **PROD (AppC)** |
| **VLAN pool**  **(static)** | 1400-1499 | 1400-1499 | 1400-1499 | 1400-1499 | 1400-1499 |

Internal Network Centric VLAN pools:

* PRD.NET\_VPL – v1008-1099

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SERVERS (NetC)** | **Common** | **Mgmt** |
| **VLAN pool**  **(static)** | 1008-1099 | 1008-1099 | 1008-1099 |

Firewall DMZ Network Centric VLAN pools:

* PRD.NET\_VPL – v1008-1099

|  |  |
| --- | --- |
|  | **DMZ (NetC)** |
| **VLAN pool**  **(static)** | INET – 1128-1143  TRAD – 1144-1160  USER – 1161-1168 |

# EPG Guidelines

1. The ACI/UCS/vCenter integration will mean that EPG names created in ACI will appear as Port Group names in vCenter. Care must be taken to ensure that names are meaningful to the Network and VMware teams. When pushed from ACI the Port Group will have the following format:

cid:image003.png@01D55296.2FDA10D0

The above format breaks down as: Tenant | Application Profile | EPG

As a result, it seems pointless adding the Tenant and/or Environment names to the EPG name as this is unnecessarily repetitious.

Therefore In non-default Tenants the EPG will not include an environment name as this is conveyed with the fully qualified name of the Port Group and/or nested in the APIC menus.

Examples:

PRD | LD5.TPVOL\_APP | LD5.WEB01\_EPG

QA | ARK.TPVOL\_APP | ARK.APP01\_EPG

1. EPGs created in the MSO are site specific and will require either an ARK or LD5 prefix.
2. In default Tenants ‘common’ and ‘mgmt’ the EPG names will include an environment name as there are not multiple ‘Common’ and ‘mgmt’ Tenants.

Examples

LD5.PRD.BLOX\_EPG

LD5.TEST.BLOX\_EPG

1. EPGs will not be created for individual application components instead the EPG will encompass the application as a whole. For applications that have some components that reside on a DMZ then it’s a given that these components will be separated and traffic flows must be known. This will allow for future, more granular segmentation.
2. It’s acknowledged that it may be difficult to determine where an application ‘stops’ and another ‘starts’.
3. EPG names will be in uppercase using a period as the delimiter. No hyphens or underscores etc. EPGs will use the suffix \_EPG.
4. App names in EPGs will be taken from the CMDB although app name length needs to be considered.

# Contract Guidelines

1. All Contracts will be created in the Common tenant. This will allow reusability of Contracts.
2. All Contracts will ‘Globally’ scoped. This will allow use of Contracts between Tenants.
3. Where possible all Contracts will use specific Filters. Only in exceptional circumstance will we use ‘IPANY’.
4. Contract names do not represent a ‘TO’ or ‘FROM’ relationship instead they represent a ‘BETWEEN’ relationship. i.e.

i.e. TPVOL.APP.GW\_CTR does not suggests a contract from ‘APP to GW’.

The elements within the contract will be arranged alphabetically. For example, if the components are WEB, APP and DB the format would be:

TPVOL.APP.DB\_CTR

TPVOL.APP.WEB\_CTR

TPVOL.DB.WEB\_CTR

1. Contracts to be created for each application regardless of apparent unnecessary repetition. This will hopefully prevent a Contract being modified for a single App that then cascades down for all Application using this Contract.

TPVOL.APP.GW\_CTR = TCP 443

CDS.APP.GW\_CTR = TCP 443

1. Contract filters will be in lowercase including the suffix \_flt. The protocol ‘name’ will also be in lowercase. These allows for ease of automated filter creation.
2. Contracts will not include an environment name i.e. no PRD
3. Filters will not include an environment name i.e. no PRD.

# ERC

When a Layer 3 Bridge Domain is marked “Advertise Externally” and associated with an L3Out (i.e. PRD.CORPNET\_L3O), the prefix assigned to the BD is advertised via the L3Out.

When a prefix is advertised to the fabric via an L3Out (i.e. PRD.INET\_L3O) that prefix is not automatically advertised outbound via another L3Out (i.e. PRD.CORPNET\_L3O). This is controlled by creating an ERC under the outbound L3Out.

The naming convention used for ERCs is as follows:

*SourceL3O*.TO.*DestinationL3O*\_ERC

Example - INET.L3O.TO.CORPNET.L3O\_ERC

# IP addressing

1. Starting point:
   1. /23 per app-centric Production tenant
   2. /24 per app-centric non-Prod Tenant.
   3. N \* /24 per Net-Centric as required.

More specifically:

|  |  |
| --- | --- |
| **Tenant** | **Address Range** |
| DEV | 10.9x.68.0/24 |
| QA | 10.9x.69.0/24 |
| UAT | 10.9x.70.0/24 |
| STRESS | 10.9x.71.0/24 |
| PROD | 10.9x.72.0/23 |
| SERVERS | 10.9x.32.0/24 |
| DMZ vINET | 10.9x.128.0/24 |
| DMZ vTRAD | 10.9x.136.0/24 |
| Common | 10.9x.66.0/24 |

# Appendix

A close up of a map

Description automatically generated