ACI Deep Dive

Fabric Discovery – Infra build up
APIC and clustering

Nov 2018

Roland Ducomble – rducombl@cisco.com

Cisco TS Technical Leader – ACI Solution Support Team

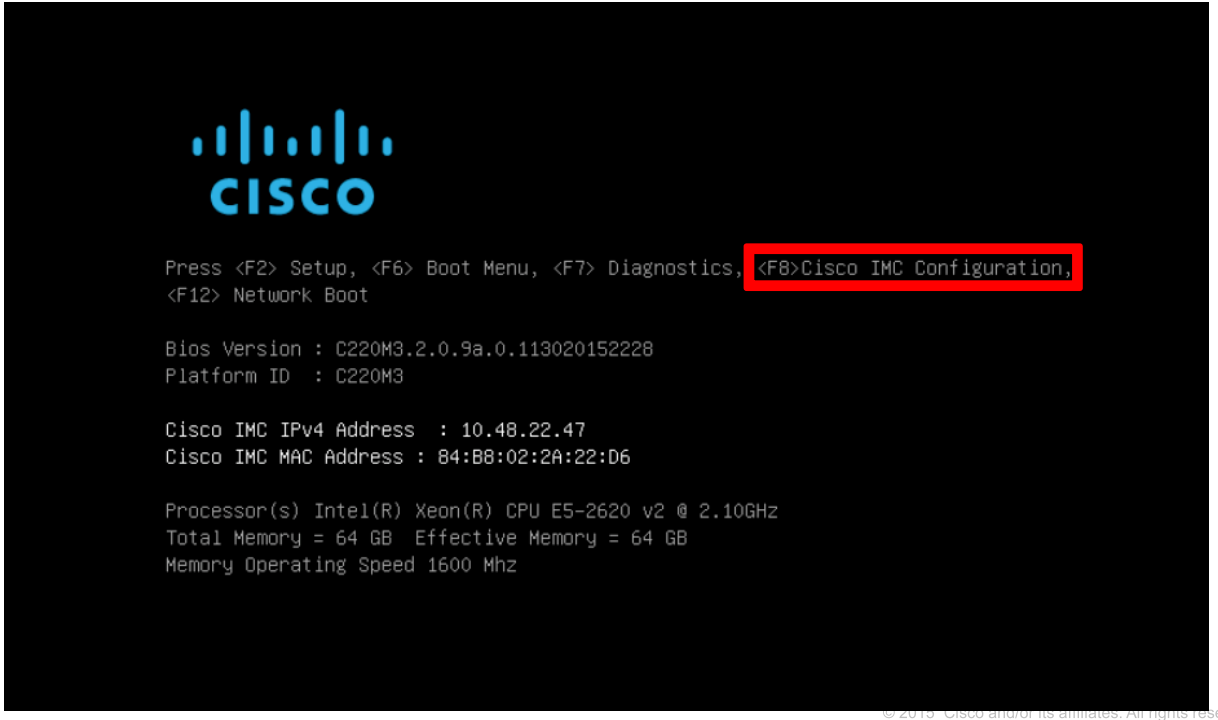
CCIE 3745

V3.1

Fabric Discovery review

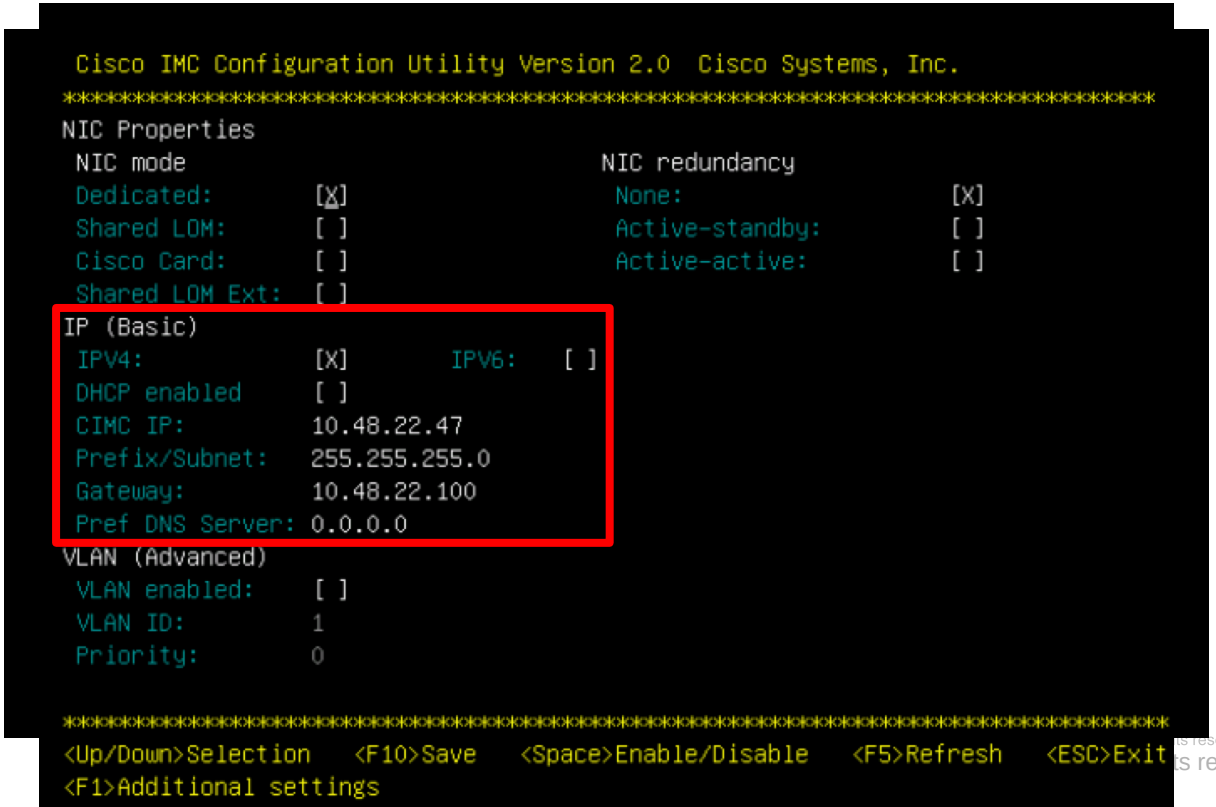
Fabric Setup

Fabric Setup starts from APIC 1 CIMC/Console access



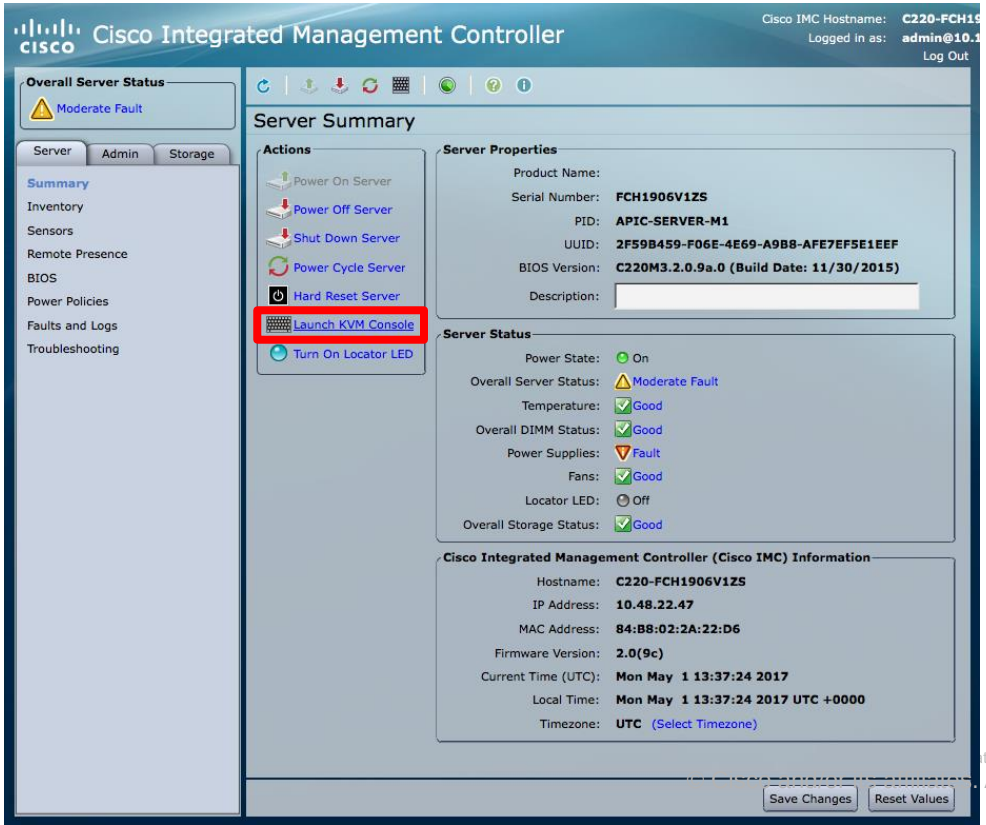
Fabric Setup

Fabric Setup starts from APIC 1 CIMC/Console access



Fabric Setup

Fabric Setup starts from APIC 1 CIMC/Console access



Fabric Initial Setup Script

- **Fabric Name**
- **Fabric ID**
- **Number of Active Controllers**
- POD ID
- Standby Controller
- **TEP Address Pool**
- **Infrastructure VLAN**
- BD Multicast Addresses
- Out-of-band Information
- Password

```
Cluster configuration ...
  Enter the fabric name [ACI Fabric1]: ACI_FAB11
  Enter the fabric ID (1-128) [1]:
  Enter the number of active controllers in the fabric (1-9) [3]:
  Enter the POD ID (1-9) [1]:
  Is this a standby controller? [NO]:
  Enter the controller ID (1-3) [1]:
  Enter the controller name [apic1]: APIC1
  Enter address pool for TEP addresses [10.0.0.0/16]: 11.0.0.0/16
  Note: The infra VLAN ID should not be used elsewhere in your environment
        and should not overlap with any other reserved VLANs on other platforms.
  Enter the VLAN ID for infra network (1-4094): 3965
  Enter address pool for BD multicast addresses (GIPO) [225.0.0.0/15]:

Out-of-band management configuration ...
  Enable IPv6 for Out of Band Mgmt Interface? [N]:
  Enter the IPv4 address [192.168.10.1/24]: 10.48.22.44/24
  Enter the IPv4 address of the default gateway [None]: 10.48.22.100
  Enter the interface speed/duplex mode [auto]:

admin user configuration ...
  Enable strong passwords? [Y]:
  Enter the password for admin:

  Reenter the password for admin: _
```

Fabric Initial Setup Script

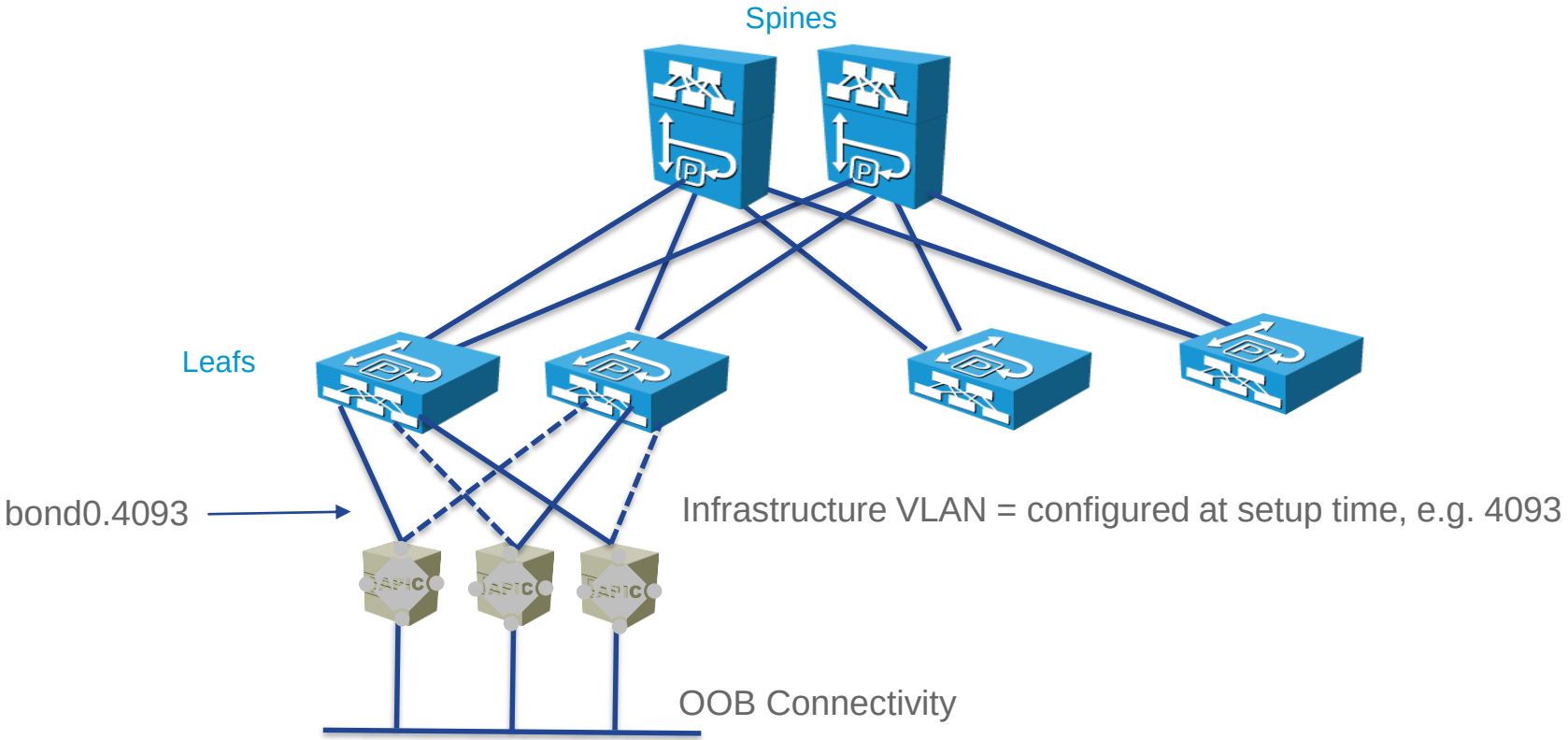
- **Fabric Name**
- **Fabric ID**
- **Number of Active Controllers**
- POD ID
- Standby Controller
- **TEP Address Pool**
- **Infrastructure VLAN**
- BD Multicast Addresses
- Out-of-band Information
- Password

```
  Number of controllers: 3
  Controller name: APIC1
  POD ID: 1
  Controller ID: 1
  TEP address pool: 11.0.0.0/16
  Infra VLAN ID: 3965
  Multicast address pool: 225.0.0.0/15

Out-of-band management configuration ...
  Management IP address: 10.48.22.44/24
  Default gateway: 10.48.22.100
  Interface speed/duplex mode: auto

admin user configuration ...
  Strong Passwords: Y
  User name: admin
  Password: ********

The above configuration will be applied ...

Warning: TEP address pool, Infra VLAN ID and Multicast address pool
         cannot be changed later, these are permanent until the
         fabric is wiped.

Would you like to edit the configuration? (y/n) [n]:
```

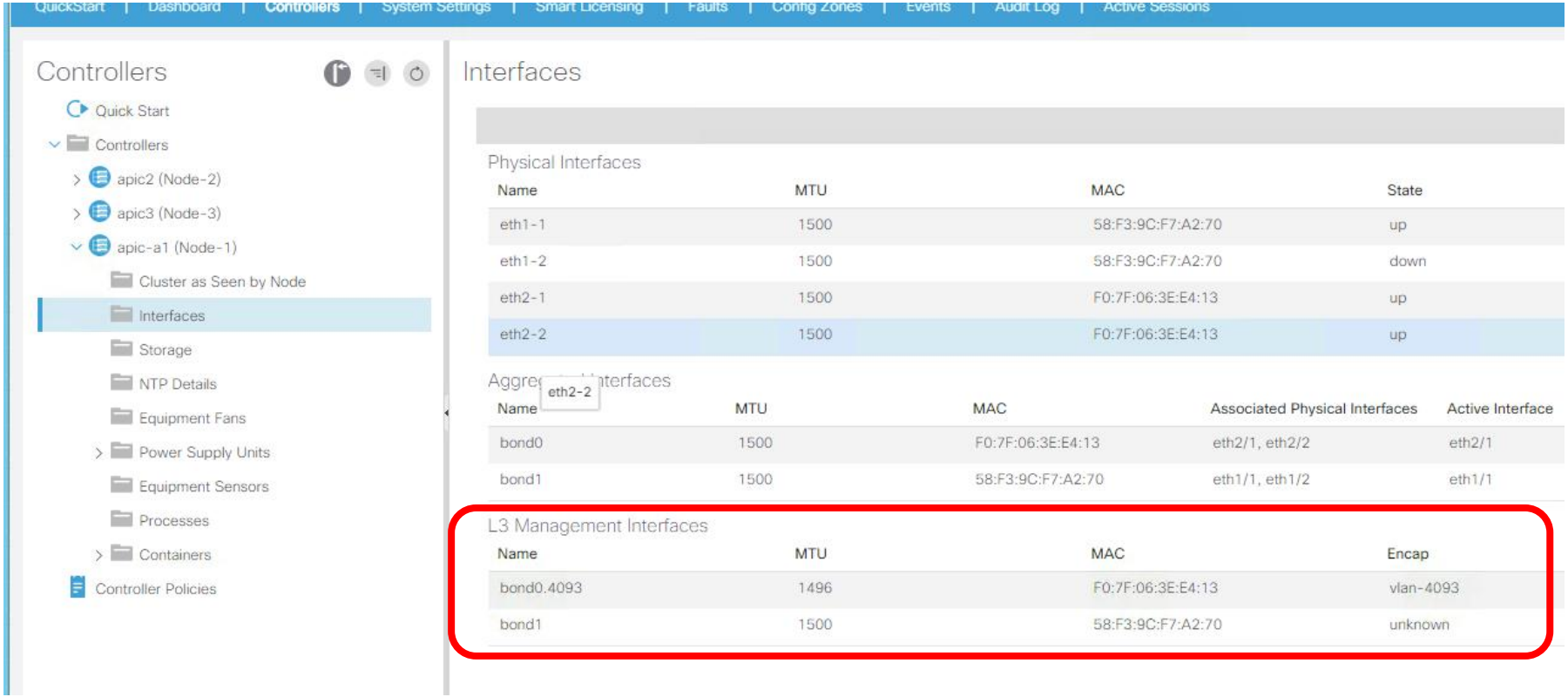
Some Best Practice Values

- TEP IP Pool, should be unique
- /16 safest
- Consider RFC 6598 range (100.64/10), if RFC 1918 is not available for use/
- Vlan should be from unreserved Vlans in other switches, 3967 or lower best practice
- * TEP Pool Change Requires Fabric Rebuild

APIC Connectivity: NIC Teaming



APIC Connectivity to the Fabric



APIC interface

```
apic1# ifconfig bond0.3932
bond0.3932: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1496
        inet 10.0.0.1  netmask 255.255.255.255  broadcast 10.0.0.1
        inet6 fe80::e20e:daff:fe00:3323  prefixlen 64  scopeid 0x20<link>
        ether e0:0e:da:00:33:23  txqueuelen 1000  (Ethernet)
        RX packets 1828129518  bytes 389972804441 (363.1 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1914177086  bytes 610968082276 (569.0 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

apic1# ifconfig bond0.101
bond0.101: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1496
        inet 10.99.98.1  netmask 255.255.255.0  broadcast 10.99.98.255
        inet6 fe80::e20e:daff:fe00:3323  prefixlen 64  scopeid 0x20<link>
        ether e0:0e:da:00:33:23  txqueuelen 1000  (Ethernet)
        RX packets 134795  bytes 21816201 (20.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 133512  bytes 26736415 (25.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

apic1# ifconfig oobmgmt
oobmgmt: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.48.25.60  netmask 255.255.255.0  broadcast 10.48.25.255
        inet6 fe80::278:88ff:fea3:2a9a  prefixlen 64  scopeid 0x20<link>
        ether 00:78:88:a3:2a:9a  txqueuelen 1000  (Ethernet)
        RX packets 35896920  bytes 9922468540 (9.2 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21428691  bytes 18558031147 (17.2 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Bond0.3932 is the infra
See Ip is TEP.[1-3]
3932 is infra vlan

Bond0.101 here is inband mgmt address (in tn mgmt.)

Oob management address

APIC routes

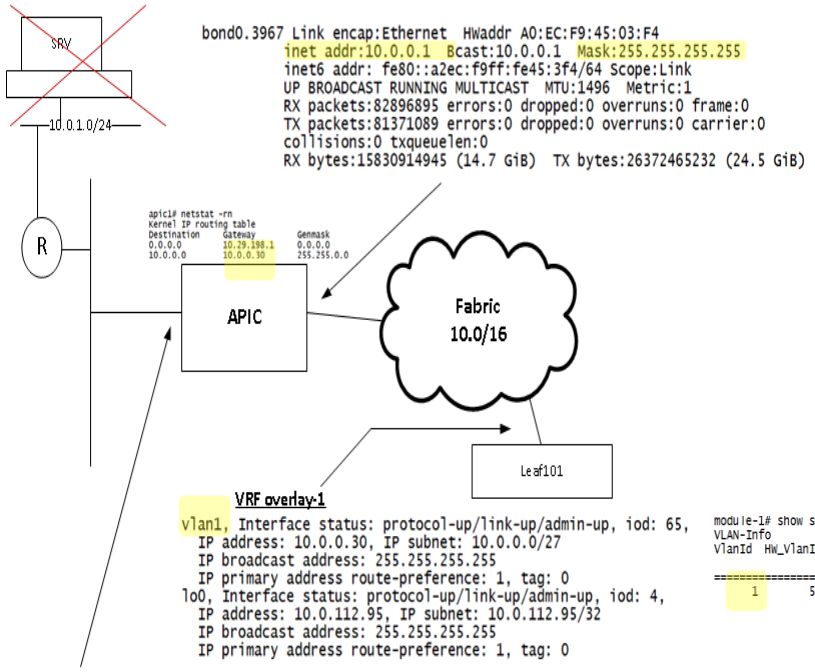
```
apic1# bash
admin@apic1:~> route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.48.25.100    0.0.0.0         UG    16     0        0 oobmgmt
10.0.0.0        10.0.0.30       255.255.0.0     UG    0      0        0 bond0.3932
10.0.0.30       0.0.0.0         255.255.255.255 UH    0      0        0 bond0.3932
10.0.8.65       10.0.0.30       255.255.255.255 UGH   0      0        0 bond0.3932
10.0.8.66       10.0.0.30       255.255.255.255 UGH   0      0        0 bond0.3932
10.48.25.0      0.0.0.0         255.255.255.0   U     0      0        0 oobmgmt
```

Here default route is through OOB

However 10.0.0.0/16 is direct on bond0.3932 (infra)

Why Unique TEP Range ?

- APIC has no VRF
- All VTEPS are /32 on nodes
- Return traffic from APIC to SRV will go inband due to static for 10.0/16
- ** Vlan 1 = Vlan 3967 in this example
- Every Leaf/Spine has 10.0.0.30/32 in this example




```
bond0.3967 Link encap:Ethernet  HWaddr A0:EC:F9:45:03:F4
          inet addr:10.0.0.1  Bcast:10.0.0.1  Mask:255.255.255.255
          inet6 addr: fe80::a2ec:f9ff:fe45:3f4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1496  Metric:1
          RX packets:82896895 errors:0 dropped:0 overruns:0 frame:0
          TX packets:81371089 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15830914945 (14.7 GiB)  TX bytes:26372465232 (24.5 GiB)
```

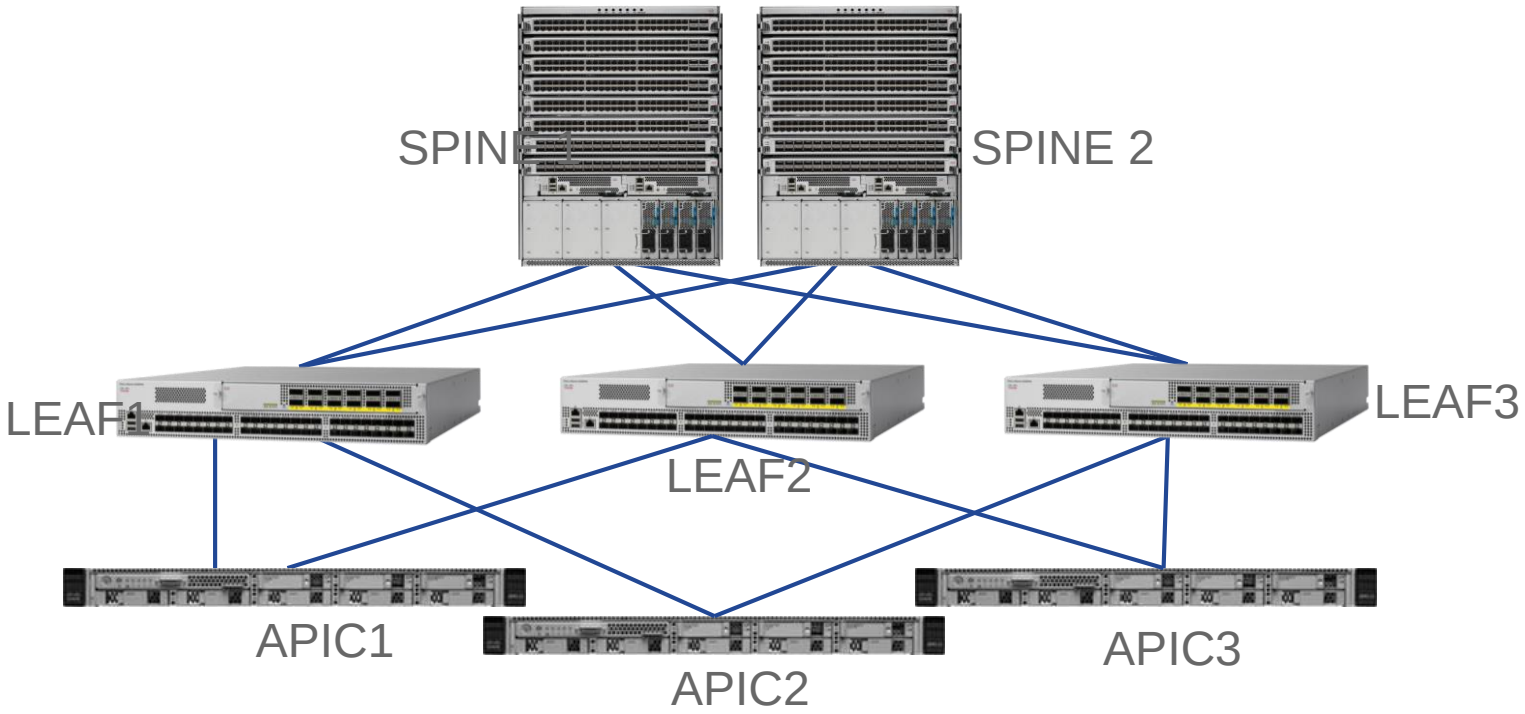
```
apic1# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask
0.0.0.0         10.29.198.1     0.0.0.0
10.0.0.0        10.0.0.30       255.255.0.0
```

```
vlan1, Interface status: protocol-up/link-up/admin-up, iod: 65,
  IP address: 10.0.0.30, IP subnet: 10.0.0.0/27
  IP broadcast address: 255.255.255.255
  IP primary address route-preference: 1, tag: 0
lo0, Interface status: protocol-up/link-up/admin-up, iod: 4,
  IP address: 10.0.112.95, IP subnet: 10.0.112.95/32
  IP broadcast address: 255.255.255.255
  IP primary address route-preference: 1, tag: 0
```

```
module-1# show system internal eltmc info vlan br
VLAN-Info
VlanId  HW_VlanId Type            Access_enc Access_enc Fabric_enc Fabric_enc BDVlan
                                  Type                  Type
==========================================================================
   1        5    BD_CTRL_VLAN     802.1q     3967       VXLAN      16777209    0
```

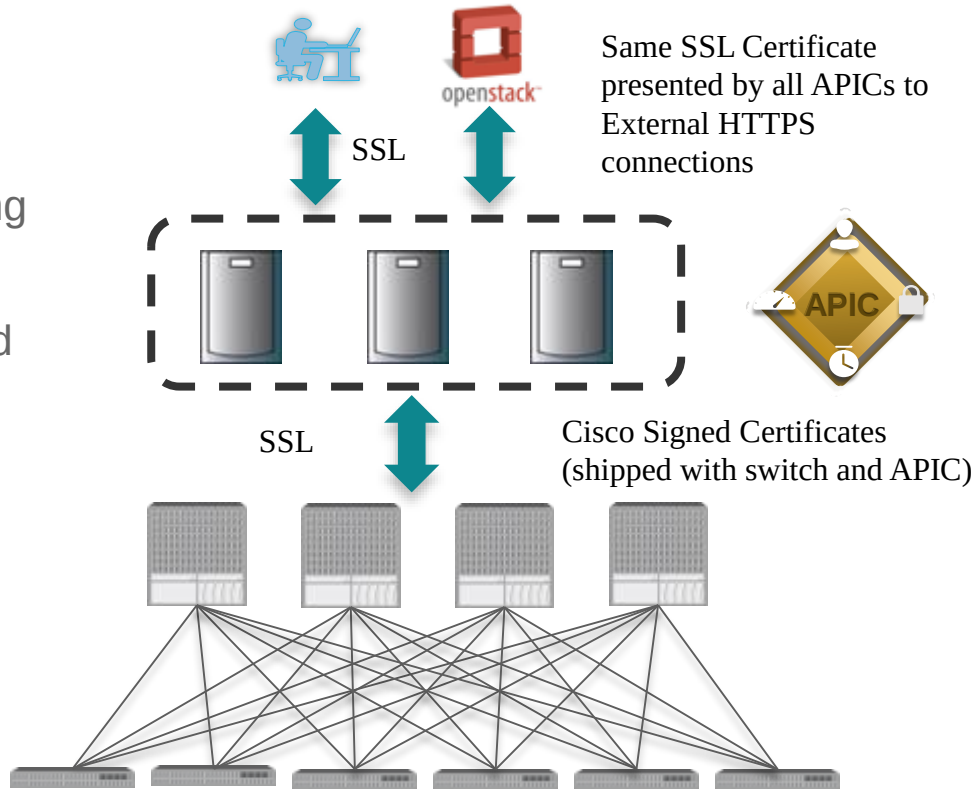
```
oobmgmt   Link encap:Ethernet  HWaddr FC:5B:39:2D:86:D2
          inet addr:10.29.198.168  Bcast:10.29.198.255  Mask:255.255.255.0
          inet6 addr: fe80::fe5b:39ff:fe2d:86d2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8121067 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4461463 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3971852753 (3.6 GiB)  TX bytes:4817305735 (4.4 GiB)
```

Fabric Discovery - Topology



APIC & ACI – A Crypto Based Platform

- User and Orchestration access to APIC
  - Web-Token or X.509 based certs
- APIC to Switch - SSL connection leveraging public key certificates
- APIC ISO is encrypted and keys are stored on APIC TPM



Fabric Discovery – Sequence of Events

1. **LLDP** Exchange
   - APIC: acidiag run lldptool in eth2-1
   - APIC: acidiag run lldptool out eth2-1
   - FNode: show lldp neighbor detail
   - FNode: show lldp traffic

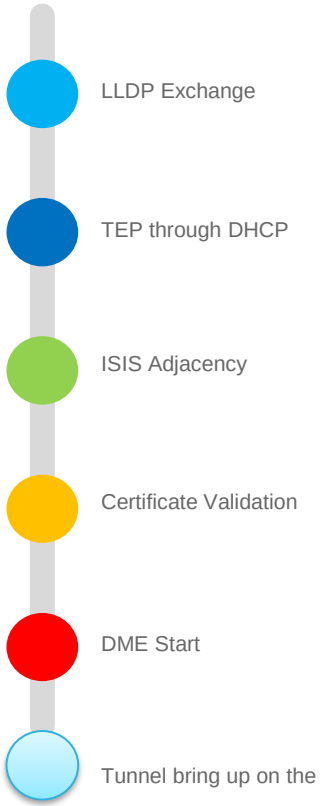
2. **DHCP** Server on APIC allocates a TEP address for the Fabric Node
   - Details are logged under /var/log/dme/log/dhcpd.bin.log

3. **ISIS** starts and builds neighbor relationship and routing table
   - Show isis adjacency vrf overlay-1
   - Show ip route vrf overlay-1

4. **Certificate** Validation
   - Clock between APIC and Switches shouldn't have a high offset

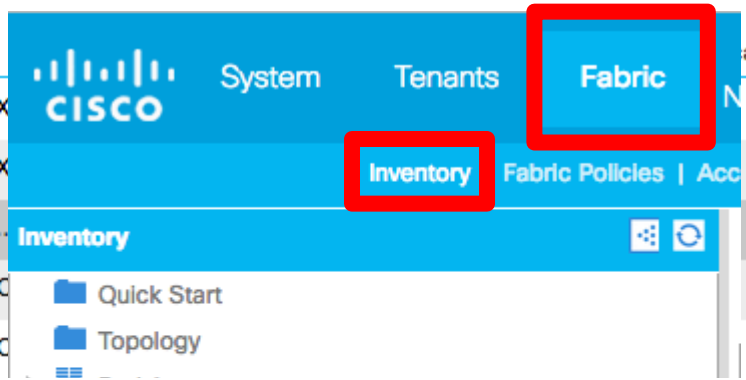
5. **DME** Process Starts on Switches
   - Check Ps –ef | egrep svc_if

6. **Vxlan tunnel build up**



Fabric Discovery – Sequence of Events

Serial Number	Pod ID	Node ID	Node Name	Rack Name	Model
SAL19069BZC	1	101	LEAF1		N9K-C9396PX
SAL19069BZJ	1	102	LEAF2		N9K-C9396PX
FDO20392LAM	1	103	LEAF3		N9K-C93180..
SAL1925H0MM	1	201	SPINE1		N9K-C9336PC
SAL1925H0L5	1	202	SPINE2		N9K-C9336PC

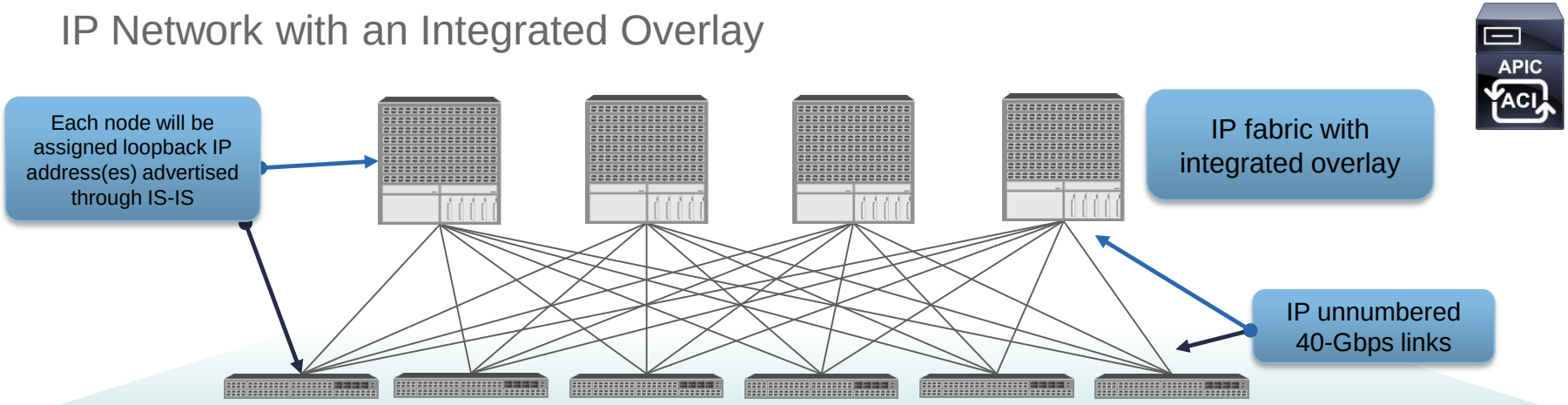


Active Controllers

ID	Name	IP	Admin State	Operational State	Health State	Failover Status	Serial Number	SSL Certificate
1	APIC1	11.0.0.1	In Service	Available	Fully Fit	idle	FCH1906V...	yes
2	APIC2	11.0.0.2	In Service	Available	Fully Fit	idle	FCH1906V...	yes
3	APIC3	11.0.0.3	In Service	Available	Fully Fit	idle	FCH1906V...	yes

Cisco ACI Fabric

IP Network with an Integrated Overlay



- As part of ACI fabric auto-discovery, Tunnel End Point (TEP) IP addresses are automatically assigned using DHCP running on the APICs.
- By default, the APICs in the cluster will use an address of 10.0.0.1 and above. For example, apic1 will be 10.0.0.1 while APICs 2 and 3 will be 10.0.0.2 and 10.0.0.3.
- The address assigned to a switch node will not change unless the node is decommissioned from the fabric. Therefore, when the switch or the APICs are reloaded, the same address should be retained.

Leaf not discovered - LLDP check from APIC

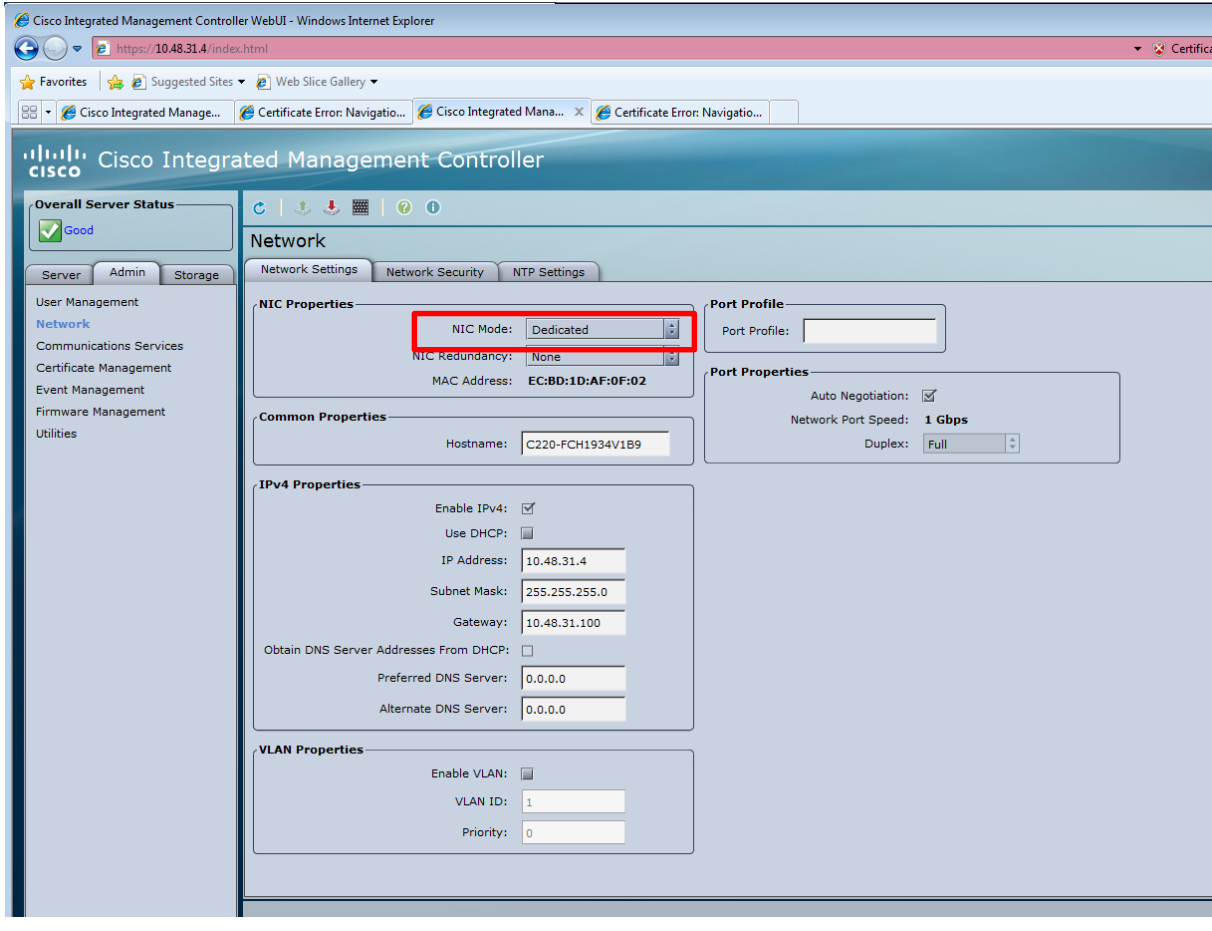
apic1# acidiag run lldptool in eth2-1
apic1# acidiag run lldptool out eth2-1

```
apic1#  acidiag run lldptool in eth2-1
Chassis ID TLV
        MAC: 50:87:89:a2:10:39
Port ID TLV
        Local: Eth1/1
Time to Live TLV
        120
Port Description TLV
        topology/pod-1/paths-102/pathep-[eth1/1]
System Name TLV
        bdsol-9396px-01
System Description TLV
        topology/pod-1/node-102
System Capabilities TLV
        System capabilities:  Bridge, Router
        Enabled capabilities: Bridge, Router
Management Address TLV
        MAC: 50:87:89:a2:10:39
        Ifindex: 83886080
Cisco 4-wire Power-via-MDI TLV
        4-Pair PoE not supported
        Spare pair Detection/Classification not required
        PD Spare pair Desired State: Disabled
        PSE Spare pair Operational State: Disabled
Cisco Port Mode TLV
        0
```

```
Cisco Port State TLV
        1
Cisco Serial Number TLV
        SAL1820SDRT
Cisco Model TLV
        N9K-C9396PX
Cisco Firmware Version TLV
        n9000-12.0(1p)
Cisco Node Role TLV
        1
Cisco Infra VLAN TLV
        3967
Cisco Node IP TLV
        IPv4:10.0.208.93
Cisco Name TLV
        bdsol-9396px-01
Cisco Fabric Name TLV
        BRU-ACI1
Cisco Node ID TLV
        102
Cisco POD ID TLV
        1
Cisco Appliance Vector TLV
        Id: 1
        IPv4: 10.0.0.1
        UUID: 8788903a-74d7-11e6-95b1-6d08b394a30c
        Id: 3
        IPv4: 10.0.0.3
        UUID: 76be8f08-74d6-11e6-a557-193a1158a354
End of LLDPDU TLV
```

CIMC settings

CIMC – Admin –Network
NIC mode must be dedicated –
If shared LOM is set LLDP will be consumed (Note it may still show up in lldptool)



LLDP must be disable in CIMC adapter

```
bdsol-apicl1-01.cisco.com# scope chassis
bdsol-apicl1-01.cisco.com /chassis # scope adapter 1
bdsol-apicl1-01.cisco.com /chassis/adapter # show detail | grep LLDP
    LLDP: Disabled
```

Verify APIC

```
pod2-apic1# acidiag verifyapic
openssl_check: certificate details
subject= CN=FCH1824V2GP,serialNumber=PID:APIC-SERVER-L1 SN:FCH1824V2GP
issuer= CN=Cisco Manufacturing CA,O=Cisco Systems
notBefore=Jul 15 01:18:43 2014 GMT
notAfter=Jul 15 01:28:43 2024 GMT
openssl_check: passed
ssh_check: passed
all_checks: passed
```

TEP allocated by DHCP / VPC address

```
bdsol-aci32-leaf2# show ip interface vrf  overlay-1 | egrep -A 1 "loo"
loopback0, Interface status: protocol-up/link-up/admin-up, iod: 20, mode: ptep, vrf_vnid: 16777199
  IP address: 10.0.80.94, IP subnet: 10.0.80.94/32
--
loopback1, Interface status: protocol-up/link-up/admin-up, iod: 21, mode: vpc, vrf_vnid: 16777199
  IP address: 10.0.96.64, IP subnet: 10.0.96.64/32
--
loopback1023, Interface status: protocol-up/link-up/admin-up, iod: 24, mode: ftep, vrf_vnid: 16777199
  IP address: 10.0.0.32, IP subnet: 10.0.0.32/32
--
```

VPC VIP address (only coming later if VPC is configured on APIC)

```
bdsol-aci32-leaf2# show system internal epm  vpc


Local TEP IP                    : 10.0.80.94
Peer TEP IP                     : 10.0.88.95
vPC configured                  : Yes
vPC VIP                         : 10.0.96.64
MCT link status                 : Up
```

DHCP server log /var/log/dme/log

```
apic1# egrep ISC /var/log/dme/log/dhcpd.bin.log  | more
684||16-11-16 09:27:46.051+00:00||dhcp||INFO||||ISC dhcpd: DHCPDISCOVER from 18:8b:9d:ad:06:d5 via
bond0.3933||../svc/dhcpd/src/gen/ifc/beh/imp/./DhcpdSvc.cc||43
684||16-11-16 09:27:46.051+00:00||dhcp||INFO||||ISC dhcpd: DHCPOFFER on 10.2.136.95 to 18:8b:9d:ad:06:d5
via bond0.3933||../svc/dhcpd/src/gen/ifc/beh/imp/./DhcpdSvc.cc||43
684||16-11-16 09:27:57.054+00:00||dhcp||ERROR||||ISC dhcpd: Dynamic and static leases present for
10.2.136.95.||../svc/dhcpd/src/gen/ifc/beh/imp/./DhcpdSvc.cc||53   bico 46.051
684||16-11-16 09:27:57.054+00:00||dhcp||ERROR||||ISC dhcpd: Remove host declaration SAL1932LNCX or remove
10.2.136.95||../svc/dhcpd/src/gen/ifc/beh/imp/./DhcpdSvc.cc||53
684||16-11-16 09:27:57.054+00:00||dhcp||ERROR||||ISC dhcpd: from the dynamic address pool for
ifabric||../svc/dhcpd/src/gen/ifc/beh/imp/./DhcpdSvc.cc||53
684||16-11-16 09:27:57.054+00:00||dhcp||ERROR||||ISC dhcpd: uid lease 10.2.136.95 for client
18:8b:9d:ad:06:d5 is duplicate on ifabric||../svc/dhcpd/src/gen/ifc/beh/imp/./DhcpdSvc.cc||53
684||16-11-16 09:27:57.055+00:00||dhcp||INFO||||ISC dhcpd: DHCPREQUEST for 10.2.136.95 (10.2.0.1) from
18:8b:9d:ad:06:d5 via bond0.3933||../svc/dhcpd/src/gen/ifc/beh/imp/./DhcpdSvc.cc||43
684||16-11-16 09:27:57.055+00:00||dhcp||INFO||||ISC dhcpd: DHCPACK on 10.2.136.95 to 18:8b:9d:ad:06:d5 via
bond0.3933||../svc/dhcpd/src/gen/ifc/beh/imp/./DhcpdSvc.cc||43
```

ISIS adjacency

```
bdsol-aci32-leaf2#  show isis   adjacency vrf overlay-1
IS-IS process: isis_infra VRF:overlay-1
IS-IS adjacency database:
System ID         SNPA            Level  State  Hold Time  Interface
4158.000A.0000     N/A            1      UP     00:00:55   Ethernet1/49.28
4258.000A.0000     N/A            1      UP     00:00:57   Ethernet1/51.29
5D58.000A.0000     N/A            1      UP     00:00:59   Ethernet1/52.1
5E58.000A.0000     N/A            1      UP     00:00:53   Ethernet1/50.26
bdsol-aci32-leaf2#
```

Tunnel on leaf

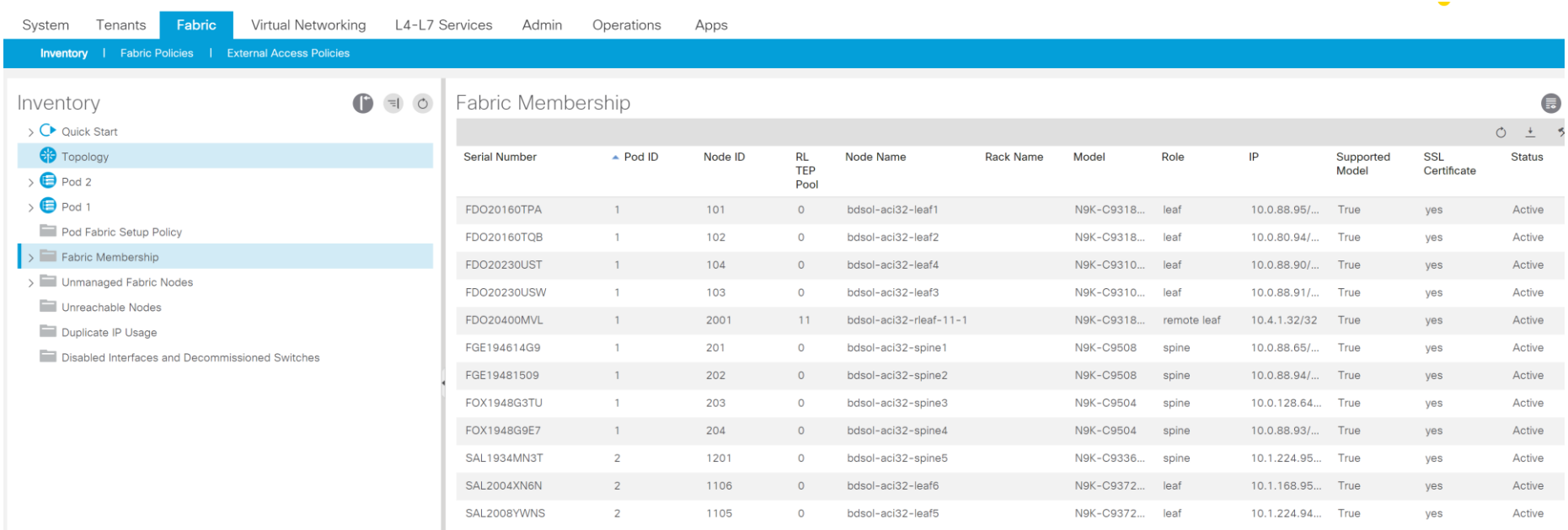
```
bdsol-aci32-leaf1# show system internal  epm interface  all | egrep Tunnel
  Interface Name   If Index  State vPC  Tunnel Dst IP    MAC address   Endpoint
 Tunnel21         0x18010015 UP    No  10.0.88.90      0000.0000.0000 1
 Tunnel27         0x1801001b UP    No  10.10.32.101    0000.0000.0000 0
 Tunnel26         0x1801001a UP    No  10.0.128.64     0000.0000.0000 0
 Tunnel24         0x18010018 UP    No  10.0.88.94      0000.0000.0000 0
 Tunnel25         0x18010019 UP    No  10.0.88.93      0000.0000.0000 0
 Tunnel30         0x1801001e UP    No  10.0.40.95      0050.566c.c7d3 0
 Tunnel31         0x1801001f UP    No  10.0.40.64      0050.5665.c3ba 0
 Tunnel28         0x1801001c UP    No  10.1.224.94     0000.0000.0000 0
 Tunnel29         0x1801001d UP    No  10.1.168.95     0000.0000.0000 0
 Tunnel10         0x1801000a UP    No  10.0.8.65       0000.0000.0000 0
 Tunnel11         0x1801000b UP    No  10.0.8.64       0000.0000.0000 0
 Tunnel8          0x18010008 UP    No  10.0.88.65      0000.0000.0000 0
 Tunnel9          0x18010009 UP    No  10.0.8.66       0000.0000.0000 0
 Tunnel14         0x1801000e UP    No  10.0.0.2        d8b1.9019.a38c 0
 Tunnel15         0x1801000f UP    No  10.0.0.3        0000.0000.0000 0
 Tunnel12         0x1801000c UP    No  10.0.96.66      0000.0000.0000 0
 Tunnel13         0x1801000d UP    No  10.0.0.1        e00e.da00.3323 0
 Tunnel18         0x18010012 UP    No  10.0.0.36       0000.0000.0000 0
 Tunnel33         0x18010021 Down  No  225.1.192.0     0000.0000.0000 0
 Tunnel16         0x18010010 UP    No  225.1.192.48    0000.0000.0000 0
 Tunnel17         0x18010011 UP    No  225.1.192.64    0000.0000.0000 0
 Tunnel22         0x18010016 UP    No  10.0.88.91      0000.0000.0000 3
 Tunnel23         0x18010017 UP    No  10.0.80.94      0000.0000.0000 1
```

PTEP in acidiag

```
pod2-apic1# acidiag fnvread
      ID              Name     Serial Number         IP Address     Role     Pod ID        State   LastUpdMsgId
--------------------------------------------------------------------------------------------------------------
     101        pod2-leaf1      SAL1820SMHV     10.0.168.95/32     leaf        1         active   0
     102        pod2-leaf2      SAL1816QVBC     10.0.168.93/32     leaf        1         active   0
     103        pod2-leaf3      SAL1818RUHM     10.0.168.91/32     leaf        1         active   0
     104   pod2-leaf4-BIS      SAL1818RP59      10.0.56.95/32     leaf        1         active   0
     201      pod2-spine1      SAL1811NN5S     10.0.168.92/32    spine        1         active   0
     202      pod2-spine2      SAL1811NN5X     10.0.168.94/32    spine        1         active   0
```

If inactive with an IP allocated → likely Cert issue (hw clock Mismatch or cert missing/bad)

Fabric node view in GUI (switches)



Serial Number	Pod ID	Node ID	RL TEP Pool	Node Name	Rack Name	Model	Role	IP	Supported Model	SSL Certificate	Status
FDO20160TPA	1	101	0	bdsol-aci32-leaf1		N9K-C9318...	leaf	10.0.88.95/...	True	yes	Active
FDO20160TQB	1	102	0	bdsol-aci32-leaf2		N9K-C9318...	leaf	10.0.80.94/...	True	yes	Active
FDO20230UST	1	104	0	bdsol-aci32-leaf4		N9K-C9310...	leaf	10.0.88.90/...	True	yes	Active
FDO20230USW	1	103	0	bdsol-aci32-leaf3		N9K-C9310...	leaf	10.0.88.91/...	True	yes	Active
FDO20400MVL	1	2001	11	bdsol-aci32-rleaf-11-1		N9K-C9318...	remote leaf	10.4.1.32/32	True	yes	Active
FGE194614G9	1	201	0	bdsol-aci32-spine1		N9K-C9508	spine	10.0.88.65/...	True	yes	Active
FGE19481509	1	202	0	bdsol-aci32-spine2		N9K-C9508	spine	10.0.88.94/...	True	yes	Active
FOX1948G3TU	1	203	0	bdsol-aci32-spine3		N9K-C9504	spine	10.0.128.64...	True	yes	Active
FOX1948G9E7	1	204	0	bdsol-aci32-spine4		N9K-C9504	spine	10.0.88.93/...	True	yes	Active
SAL1934MN3T	2	1201	0	bdsol-aci32-spine5		N9K-C9336...	spine	10.1.224.95...	True	yes	Active
SAL2004XN6N	2	1106	0	bdsol-aci32-leaf6		N9K-C9372...	leaf	10.1.168.95...	True	yes	Active
SAL2008YWNS	2	1105	0	bdsol-aci32-leaf5		N9K-C9372...	leaf	10.1.224.94...	True	yes	Active

Spine Loopback in infra (non multipod case)

Proxy loopback are protection chain used internally by COOP
Anycast loopback are the one used to send to proxy spine in case EP is Unknown in ingress leaf

```
pod2-spine1# show ip interface  vrf overlay-1 | egrep -A  1 "^loop"
loopback0, Interface status: protocol-up/link-up/admin-up, iod: 4, mode: ptep, vrf_vnid: 16777199
  IP address: 10.0.168.92, IP subnet: 10.0.168.92/32
--
loopback1, Interface status: protocol-up/link-up/admin-up, iod: 83, mode: anycast-mac, vrf_vnid: 16777199
  IP address: 10.0.232.66, IP subnet: 10.0.232.66/32
--
loopback2, Interface status: protocol-up/link-up/admin-up, iod: 84, mode: proxy-mac, vrf_vnid: 16777199
  IP address: 10.0.136.69, IP subnet: 10.0.136.69/32
--
loopback3, Interface status: protocol-up/link-up/admin-up, iod: 85, mode: anycast-v4, vrf_vnid: 16777199
  IP address: 10.0.232.65, IP subnet: 10.0.232.65/32
--
loopback4, Interface status: protocol-up/link-up/admin-up, iod: 86, mode: proxy-v4, vrf_vnid: 16777199
  IP address: 10.0.136.70, IP subnet: 10.0.136.70/32
--
loopback5, Interface status: protocol-up/link-up/admin-up, iod: 87, mode: anycast-v6, vrf_vnid: 16777199
  IP address: 10.0.232.64, IP subnet: 10.0.232.64/32
--
loopback6, Interface status: protocol-up/link-up/admin-up, iod: 88, mode: proxy-v6, vrf_vnid: 16777199
  IP address: 10.0.136.71, IP subnet: 10.0.136.71/32
pod2-spine1#
```

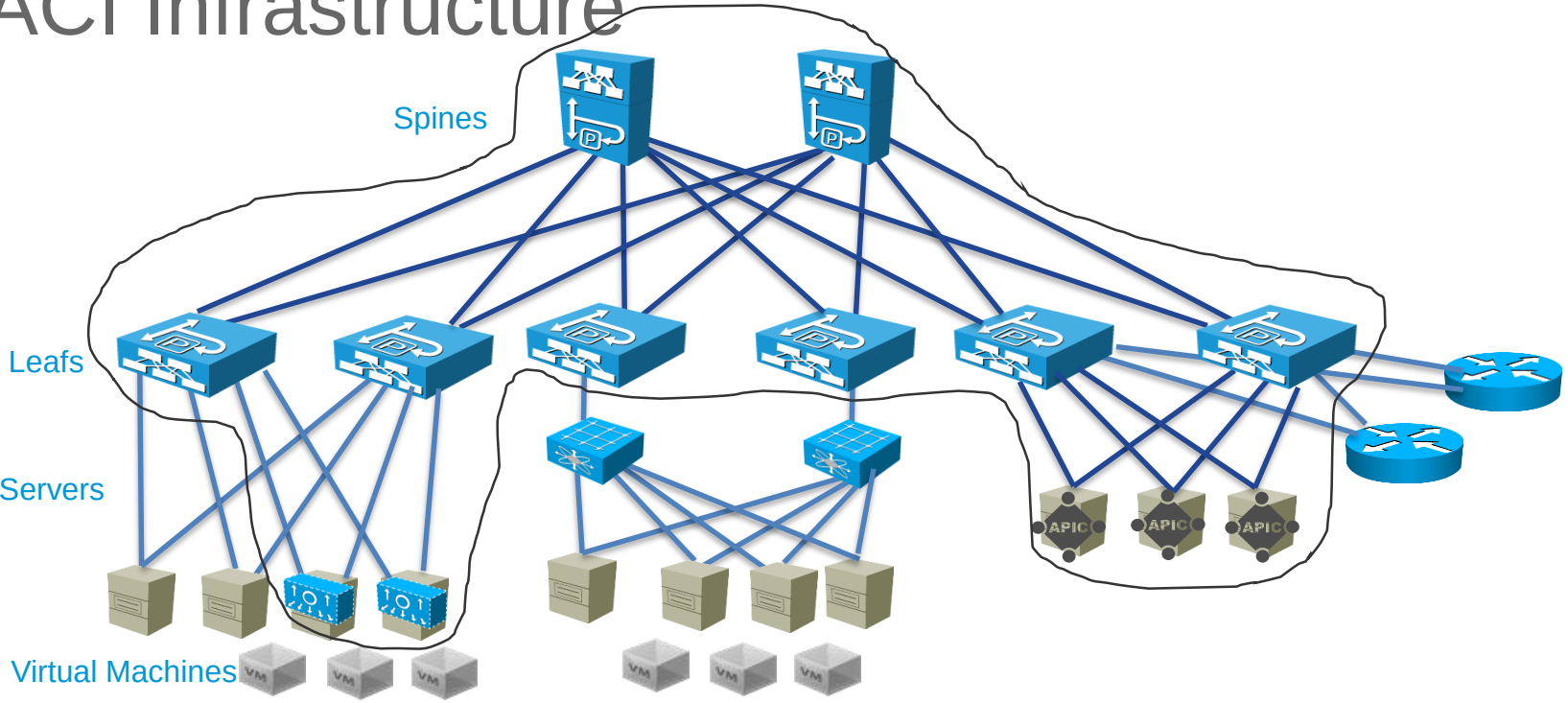
What is the infra ?

What is the Infrastructure Tenant

- Infrastructure Tenant is the "underlay" of ACI.
- Traffic encapsulated in VXLAN from one leaf to the other is sent with a destination IP of the VTEP in a pool that you define at APIC setup time (the TEP-pool)
- The forwarding for the underlay is performed on VRF overlay-1
- This "Tenant" provides connectivity to the external devices that are controlled by APIC via the "infra VLAN".

- SVI provides a pervasive gateway for remote devices which are designed to be L2 connected to the Infrastructure Bridge Domain:
  - 10.0.0.30/27 (this is the pervasive SVI)
  - Configured also as Querier IP
- In the EPG of this Tenant you see what connects to this Overlay:
  - APICs via bond0.infraVLAN, e.g. Bond0.4093
  - AVSs, AVE, Openstack node → essentially any opflex vleaf
  - VTEPs in General
- The encapsulation VLAN is called Infra VLAN

ACI Infrastructure



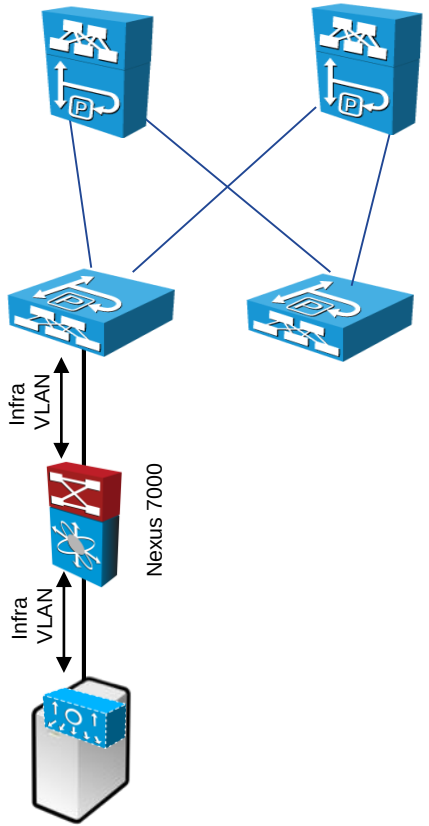
TEP Pool, IP addressing for the Fabric Nodes

- Each node in the fabric has the following IP addresses:
- Loopback addresses
  - PTEP – Physical Tunnel End Point, assigned to all the nodes
  - vPC TEP – a TEP address to represent the vPC
  - FTEP – Fabric Tunnel End Point, one for the entire fabric
  - Proxy-TEP – Proxy Tunnel End Point, assigned to a set of spines, proxy-TEP the MAC addresses, for IPv4, for IPv6
- Each link between the leafs and spines has a subinterface and an IP address on this subinterface
- Hosts that send VXLAN tagged traffic each use a TEP address also (AVS with VXLAN, Hyper-V, Openstack, etc...)

- How are these IP addresses used?
  - L3 sub-interfaces are used between leaves and spines
  - IS-IS is run on the sub-interfaces to maintain infra reachability
  - COOP is run on the PTEP loopback to sync end-point (EP) database
  - MP-BGP is run on the PTEP loopback to sync WAN routes
  - iVXLAN tunnels to PTEPs of other leaves and spine proxy TEPs

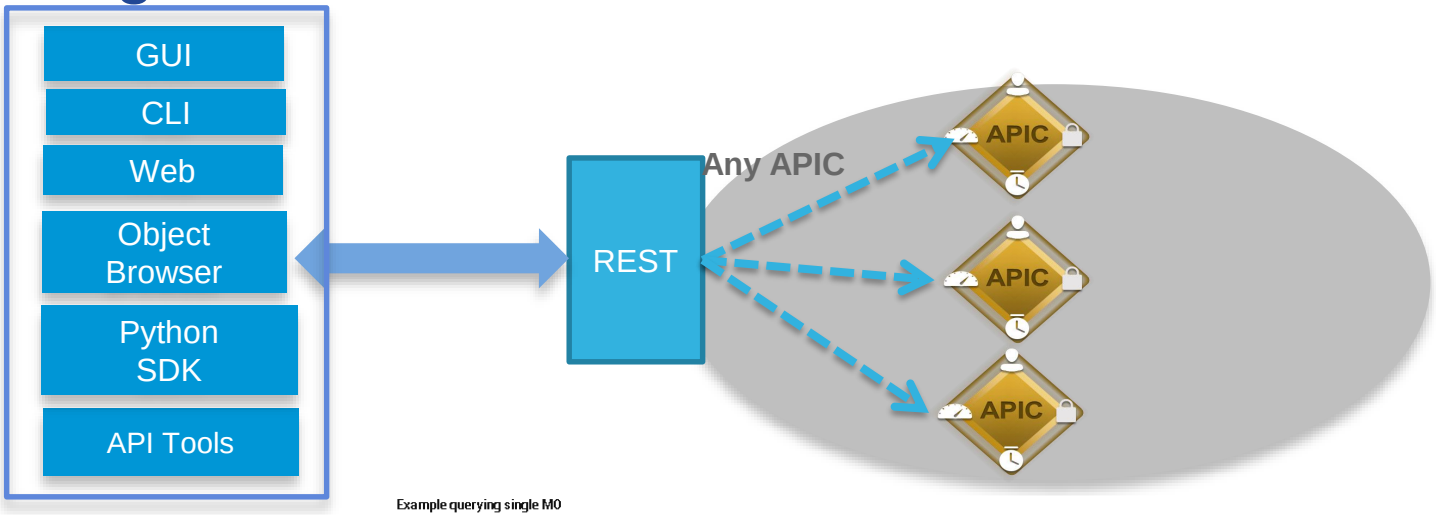
Infra VLAN

- You choose the Infra VLAN at time of the fabric bring up.
- The Infra VLAN should not be one of the reserved VLANs of other existing switches in the datacenter
- This is because it may be necessary to trunk this VLAN to support remote AVS
- Common reserved range is 3968 to 4095
- Choose a VLAN that is not normally reserved e.g. 3967
- Note: VXLAN must be sent on the infra VLAN for ACI to be able to parse it



APIC Architecture

Management access



Example querying single MO

http://bdsol-aci1-apic1/api/mo/uni/tn-common.xml

GET | URL params | Headers (0)

Send | Save | Preview | Pre-request script | Tests | Add to collection | Reset

Body | Cookies | Headers (9) | Tests | STATUS 200 OK | TIME 41 ms

Pretty | Raw | Preview | XML | Copy

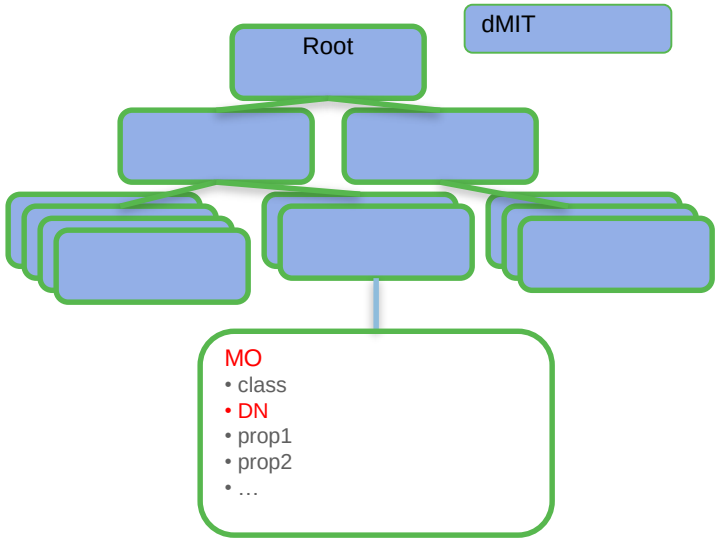
```
<imdata totalCount="1">
  <fvTenant childAction="" descr="" dn="uni/tn-common" lcOwn="local" modTs="2014-11-04T23:53:36.221+02:00" monPolD
</imdata>
```

OUTLINE

imdata totalCount: 1

fvTenant childAction: , descr: , dn: uni/tn-comm...

Managed Objects



Everything is an object

Objects are hierarchically organized

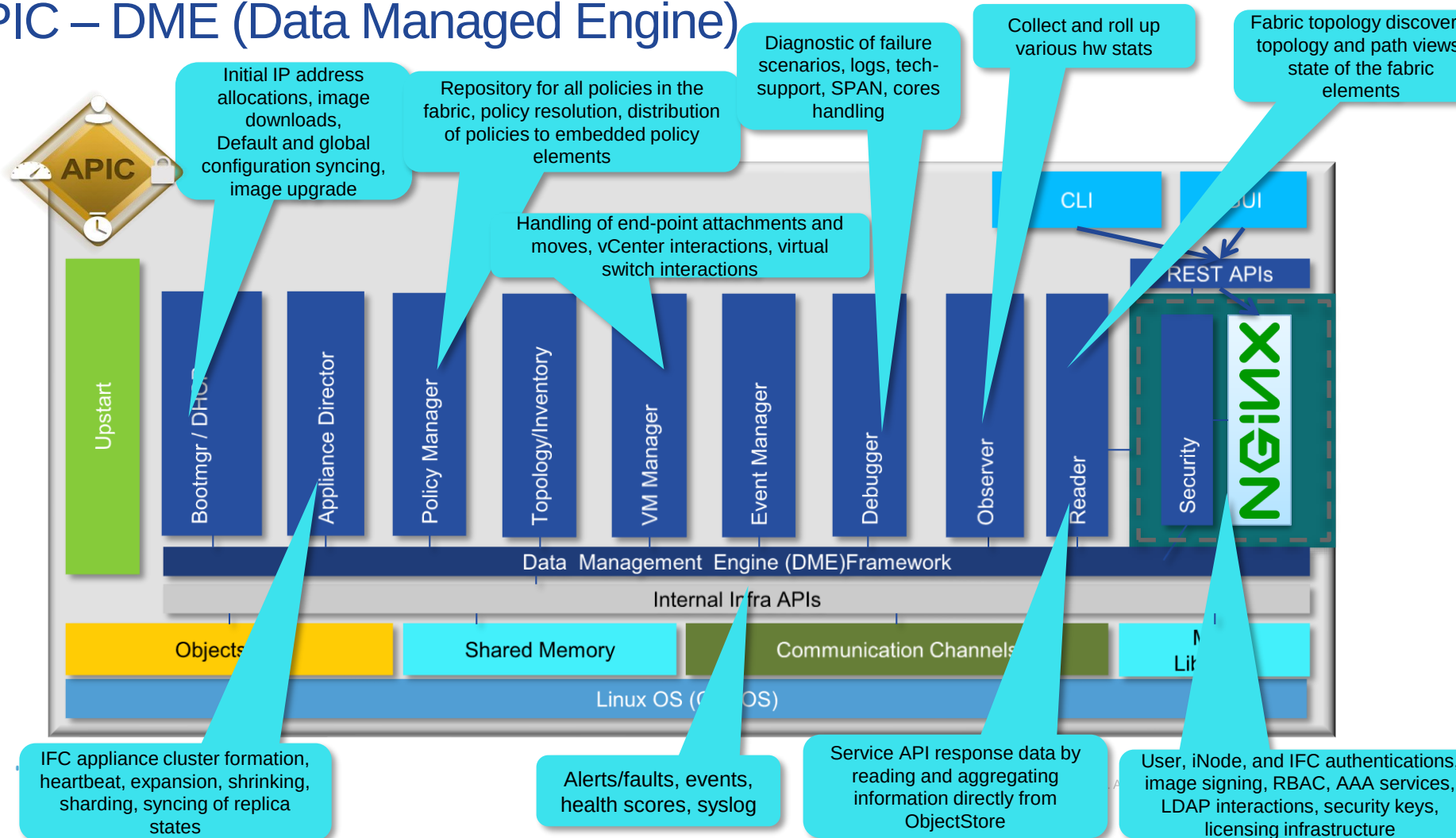
Distributed Managed Information Tree (dMIT)
contains comprehensive system information
- discovered components
- system configuration
- operational status including statistics and faults

A single logical dMIT presented to user through REST interface on any APIC

Internally dMIT is split into various services and shards in various APICs

apic2://api/node/mo/uni/tn-infra.xml

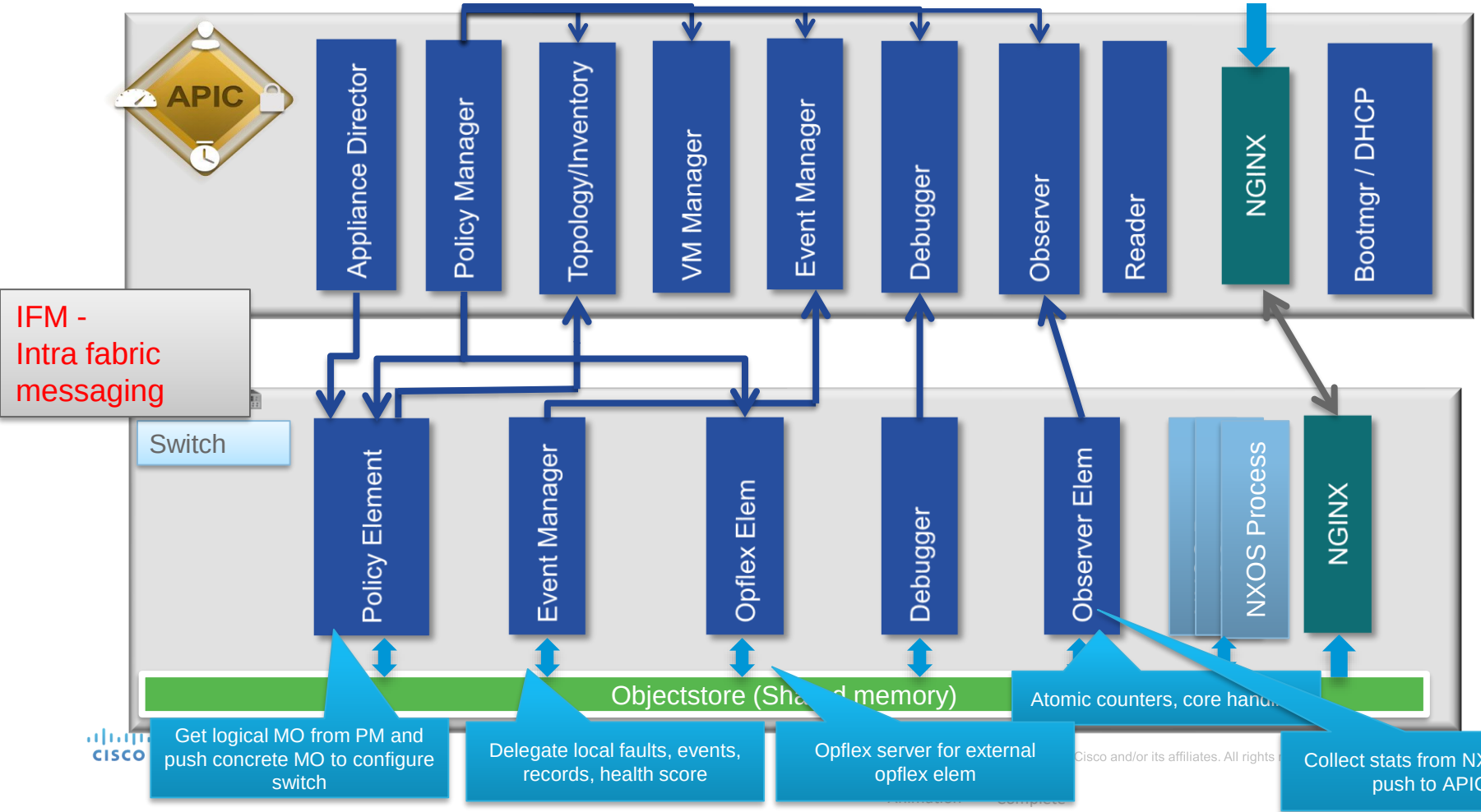
APIC – DME (Data Managed Engine)



What does APIC do (APIC Services)

- ***Policy manager***: Manages the distributed policy repository responsible for the definition and deployment of the policy-based configuration of Cisco ACI
- ***Topology manager***: Maintains up-to-date Cisco ACI topology and inventory information
- ***Observer***: The monitoring subsystem of the APIC; serves as a data repository for Cisco ACI operational state, health, and performance information
- ***Boot director***: Controls the booting and firmware updates of the spine and leaf switches as well as the APIC elements
- ***Virtual machine manager (or VMM):*** Acts as an agent between the policy repository and a hypervisor and is responsible for interacting with hypervisor management systems such as VMware vCenter
- ***Event manager***: Manages the repository for all the events and faults initiated from the APIC and the fabric nodes

DME



IFM

- IFM stands for Intra-Fabric Messaging
- It's the software layer that tries to deliver messages between the various DME
- Addresses for each agent are called “identities”: <system-type>:<system-id>:<service>:<slot>
  1:119:5:0 (policy element in switch 119).
- It uses SSL over TCP for remote communication, and connection-oriented UNIX sockets locally;
- Connections are established on-demand when an agent needs to deliver a message, and may timeout after (minutes of) inactivity.

IFM (II)

TCP ports

eventmgr	12119	observerelem	12407
nginx	12151	dbgrelem	12439
policyelem	12183	vmmmgr	12471
policymgr	12215	nxosmock	12503
reader	12247	bootmgr	12535
ae	12279	appliancedirector	12567
topomgr	12311	dhcpd	12695
observer	12343	scripthandler	12727
dbgr	12375	idmgr	12759

(some processes also use the next port)

APIC Clustering

APIC Cluster Formation and Heartbeats
All APICs are Active

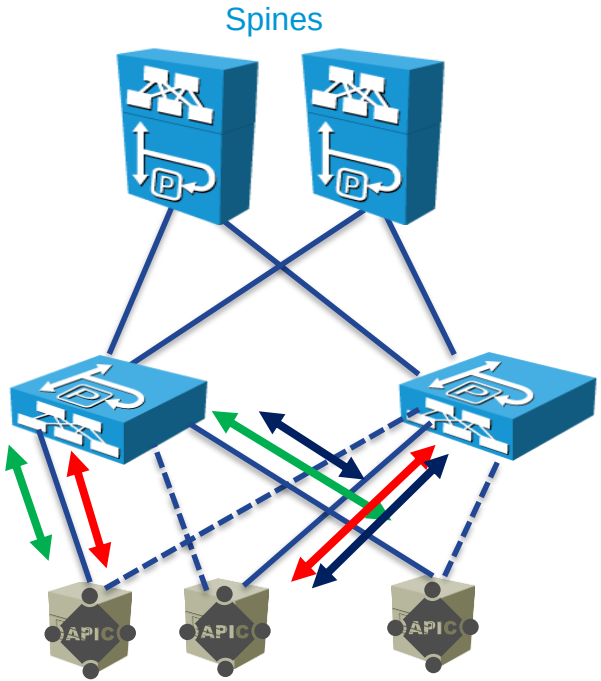
Apic forms cluster
Over inband (TEP overlay-1 network)
Apic sends Heartbeat to each over

- APIC appliances discover each other via ISIS/LLDP
- They exchange 3 unicast heartbeats per second
- Heartbeats are exchanged **inband**
- **Each APIC and all nodes in the fabric maintain an "Appliance Vector" with the information about which APIC is active, normally all 3 are**

Appliance vector for APICs

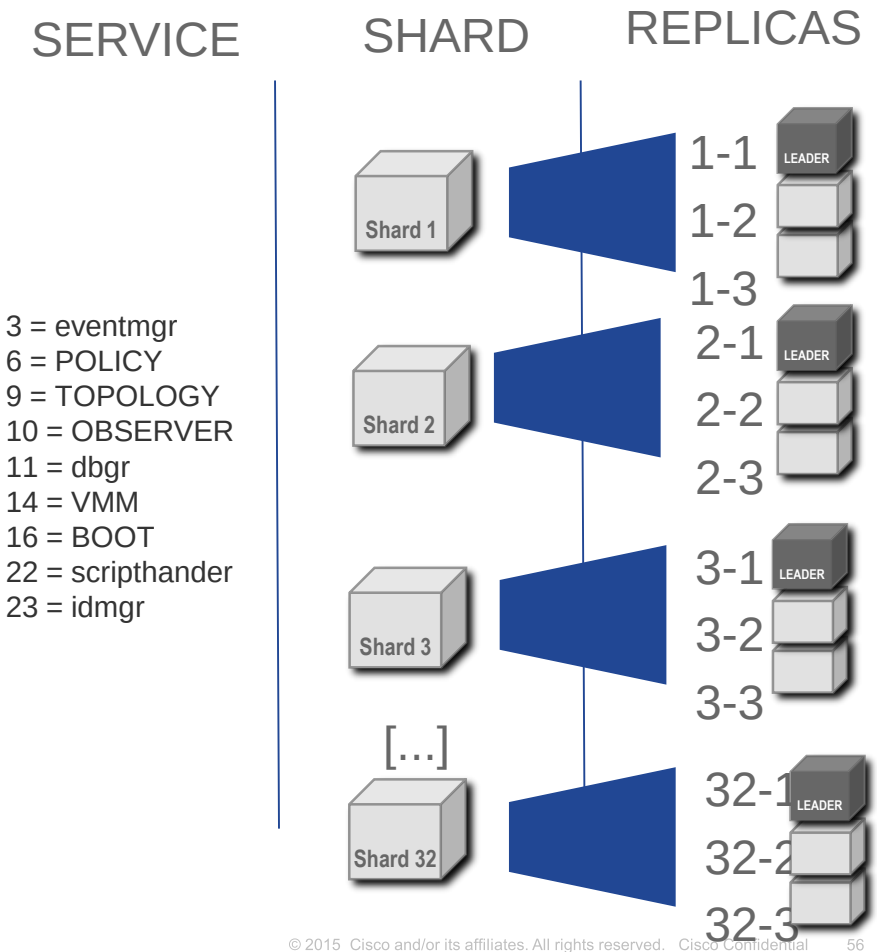
(state of each APIC)

A	A	A

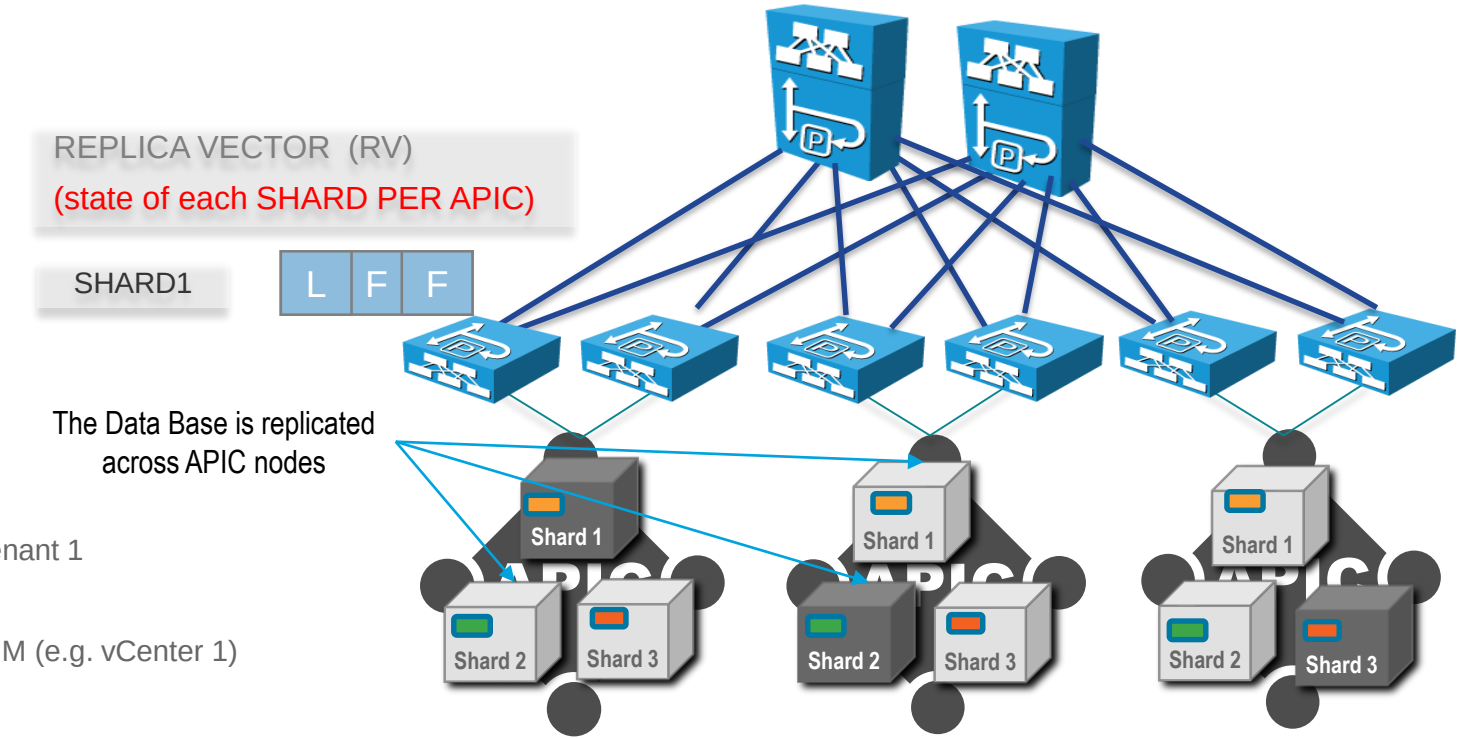


APIC is a Database

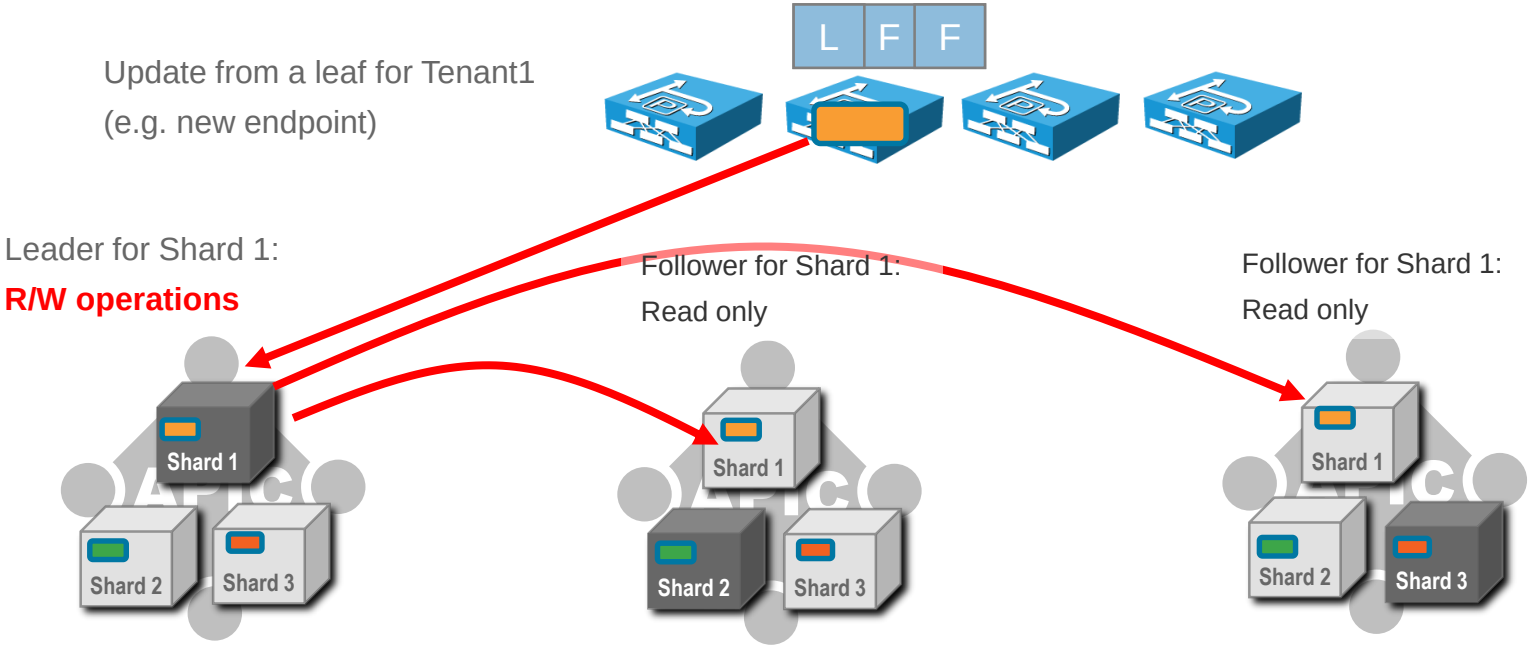
- ACI Services (DME) are organized in a database
- Each Service is saved in a shard
- Each shard is replicated 3 times
- For each shard there is a leader and two followers



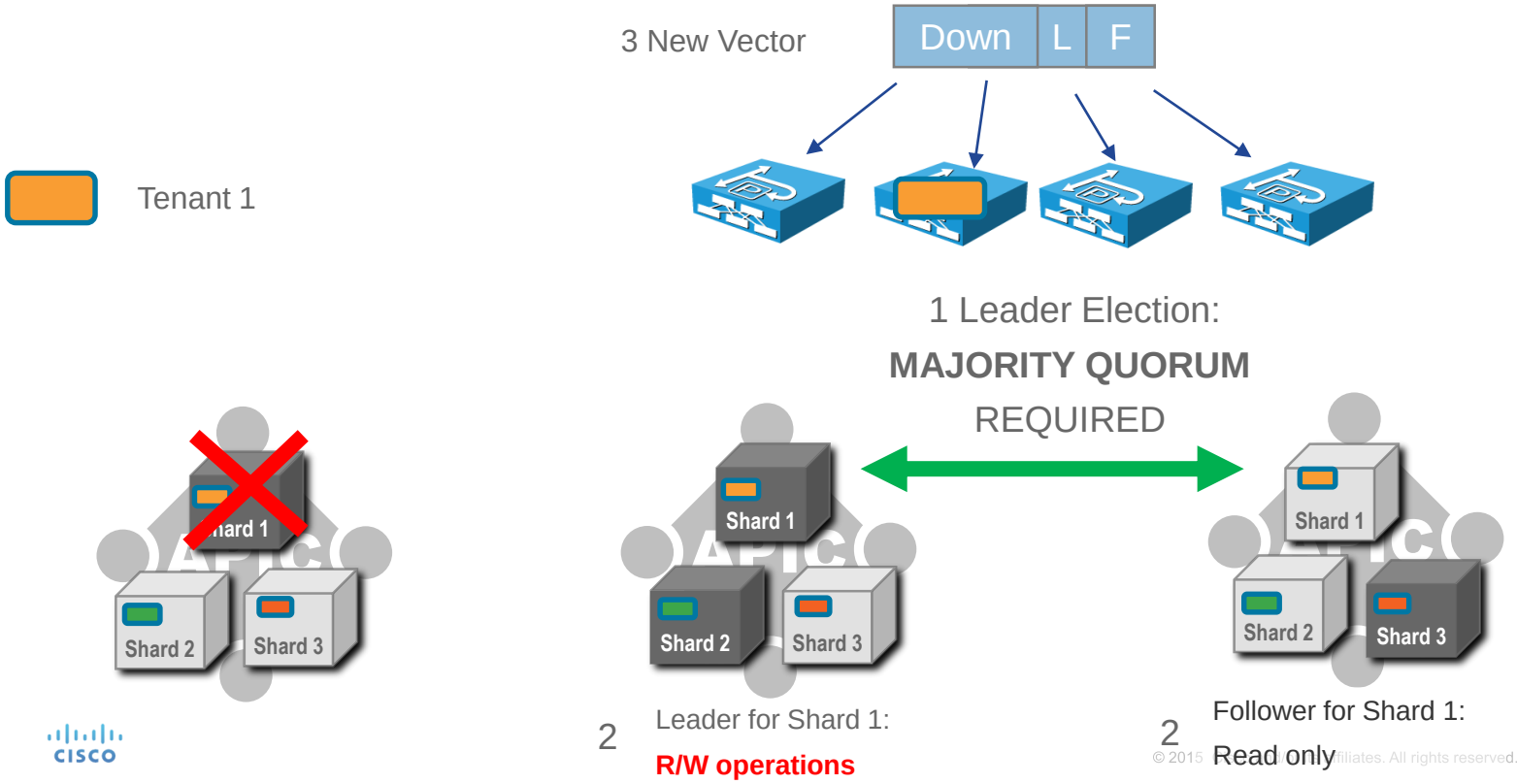
APIC Database information: all nodes in the fabric know about which APIC is the Leader for each shard and which replicas are Followers



Each shard is a cluster of 3 replicas with one Leader and 2 Followers



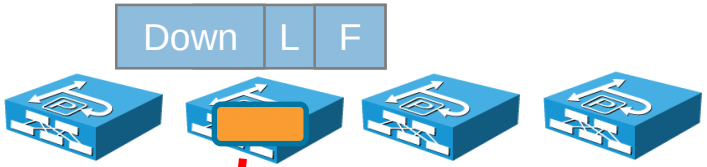
APIC Clustering is per Shard:
If one shard goes down, shards must elect a New Leader



APIC clustering is per Shard
New leader for shard 1 is on APIC2, all updates for this shard go to APIC2

Update from a leaf for Tenant1

(e.g. new endpoint)

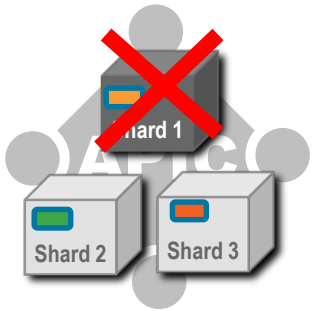


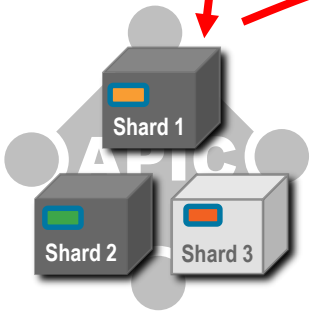
Leader for Shard 1:

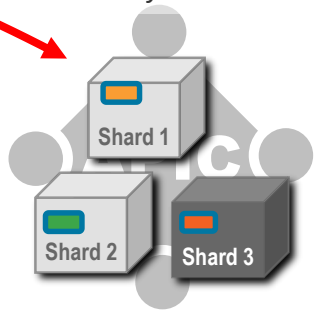
R/W operations

Follower for Shard 1:

Read only



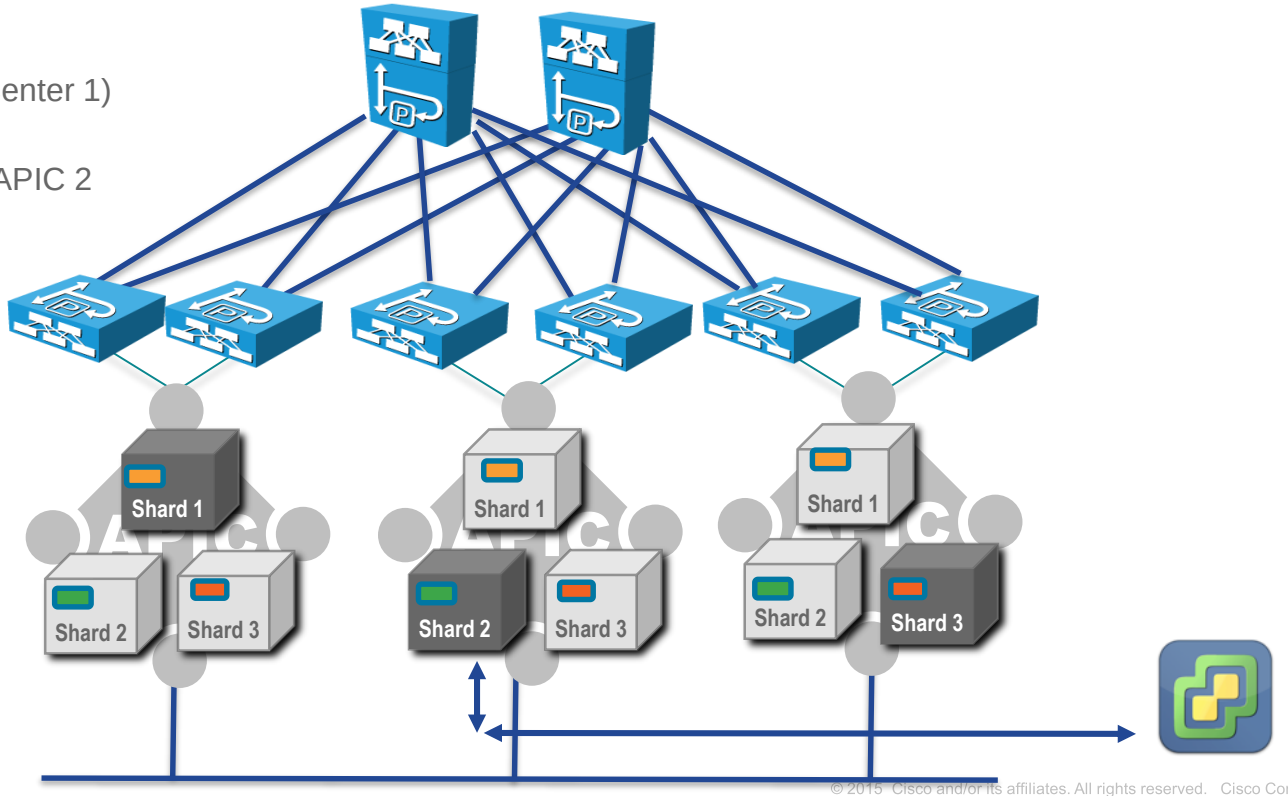




APIC Clustering Example: HA Considerations

VMM (e.g. vCenter 1)

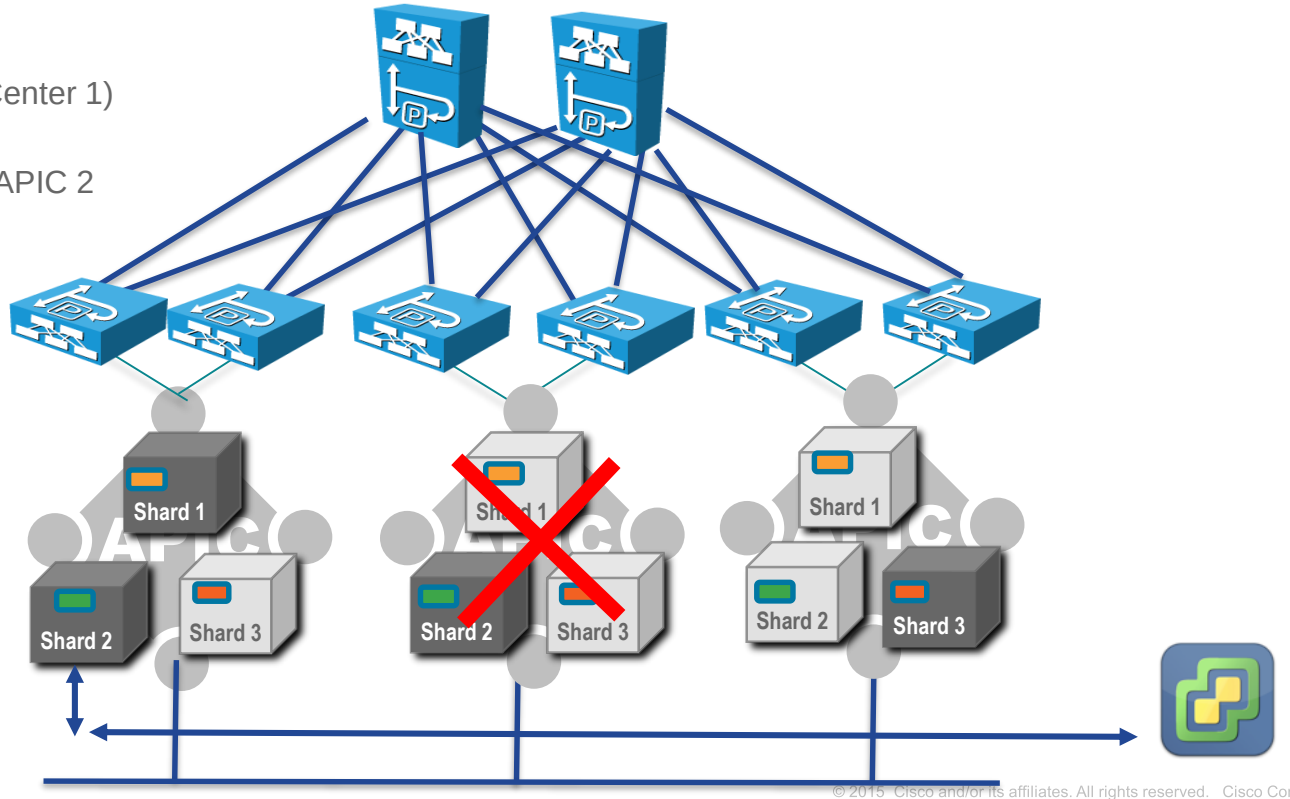
Leader is Shard 2 on APIC 2



APIC Clustering Example:1 APIC failure

VMM (e.g. vCenter 1)

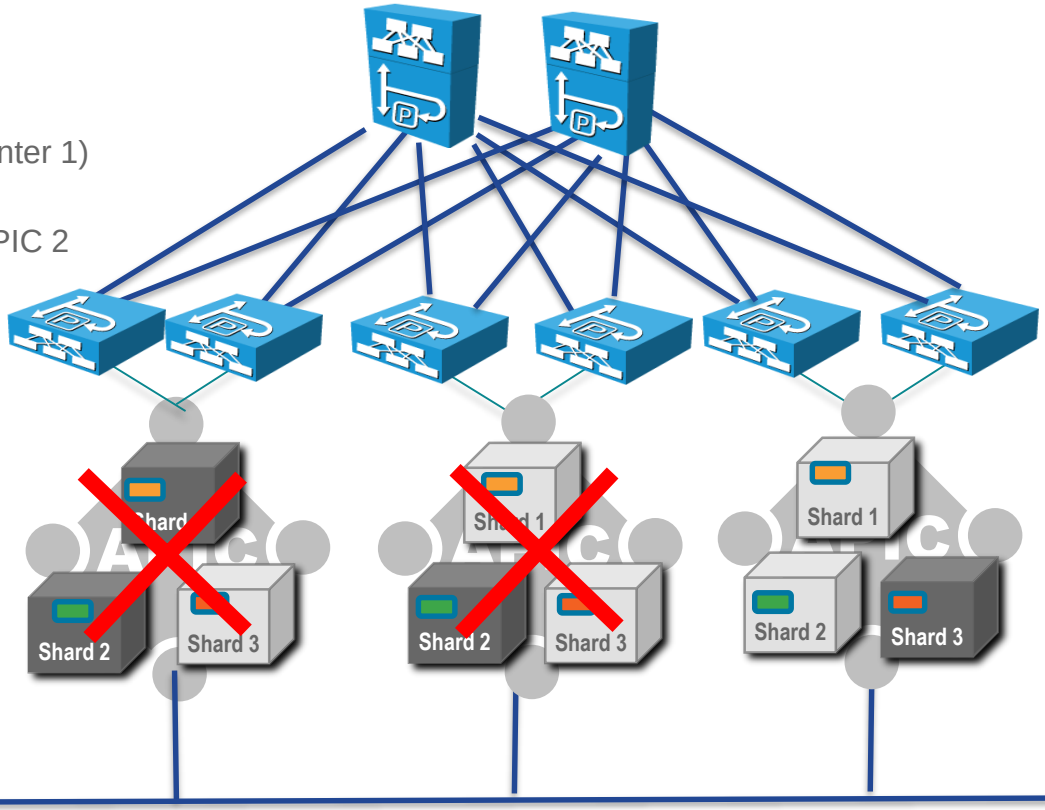
Leader is Shard 2 on APIC 2



APIC Clustering Example: 2 APICs failures

VMM (e.g. vCenter 1)

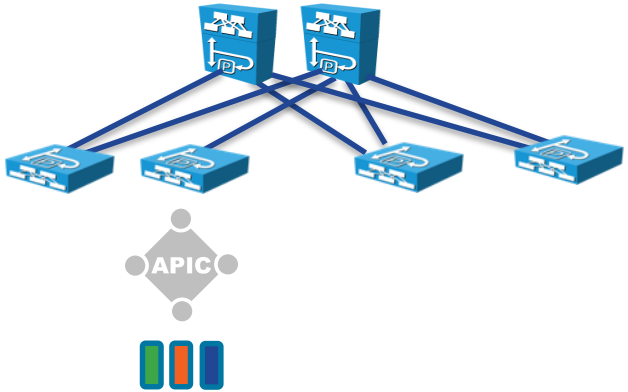
Leader is Shard 2 on APIC 2



All Shards on APIC3
Are in Minority, they
Are in Read-only mode

What if 3 APICs have not yet been installed?

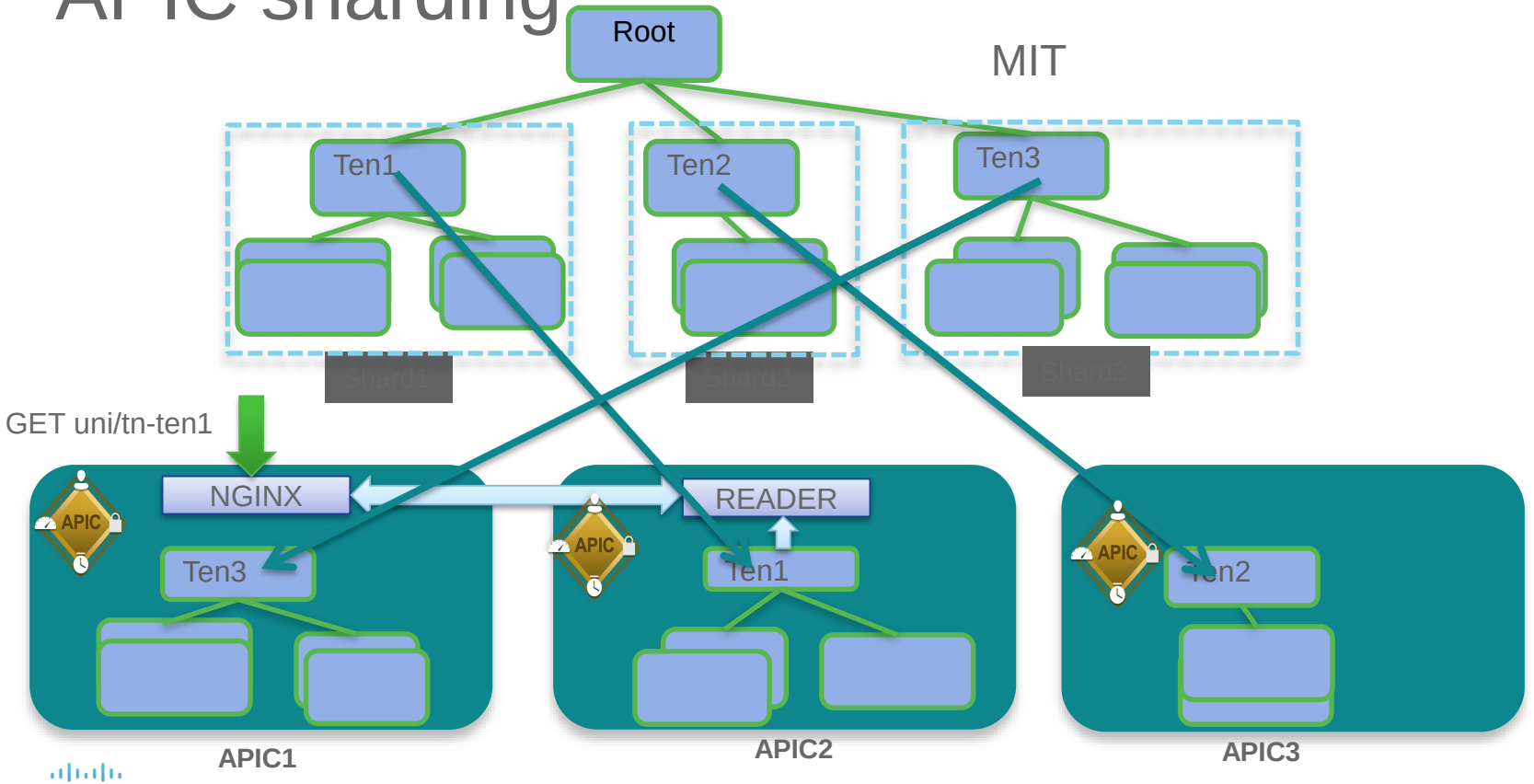
- When you start a fabric with a single APIC, the shards are not in minority because the cluster has not yet been fully fit.
- *This is a special case designed to support Zero Touch Provisioning*
- *There is only one replica of each shard*
- *The APIC will create new replicas as soon a new APIC is added*
- *This explains why when installing the ACI fabric the first time you can operate the fabric even with a single APIC*



APIC cluster - Sharding

- APIC's management tree divided into DB units called shards to allow load balancing and scaling
- Shards assigned to appliances by a static shard layout
- Each DN is mapped to a shard number (0 – 31)
- Data accessed together is put in same shard
  - All data for a tenant to one particular shard
- Note : User sees the combined MIT view on any APIC and should not care about location of data

APIC sharding



AV / RV / FNV

- Following state is distributed and updated by appliance director to each node

Fabric node vector (FNV) : Vector of switch nodes with address, state (see fabric discovery earlier section)

- Appliance vector (AV) : Vector of APIC with their address, state
- Replica vector (RV) : Vector of shard, replica with state. Leader for each shard
- **`Check using acidiag [fnvread|avread|rvread]`**

Replacing an APIC

- Need to remove existing APIC ID to chassis ID association
- Decommission APIC L on any APIC 1..N except APIC L
- Disconnect APIC L from the fabric
- Connect replacement APIC L to the fabric
- Commission APIC L on any APIC 1..N except APIC L
- It is recommended to boot-up APIC after it is connected to the fabric

Split brain / Minority behavior – Read only mode

- If replica is not connected to any other replicas it is said to be in minority
- No write request allowed on minority to avoid issues with merging
  - no configuration change
  - no vcenter update handled
  - no updates from switches handled
- Read request continue to be served from minority
  - REST read request will return data with indication that data could be stale
  - EP attach will cause dynamic policy to be downloaded
- NOTE : Irrespective of cluster size, two APIC down or disconnected will cause some shard to be in minority
- Irrespective of cluster size , 3 apic down (in case of 5 APIC cluster) will cause Info lost in DB

Why we need minimum 3 APIC

Since leader election is distributed we need majority of replicas to agree to leader. So minimum number of replicas/APIC to continue operation with single failure is 3.

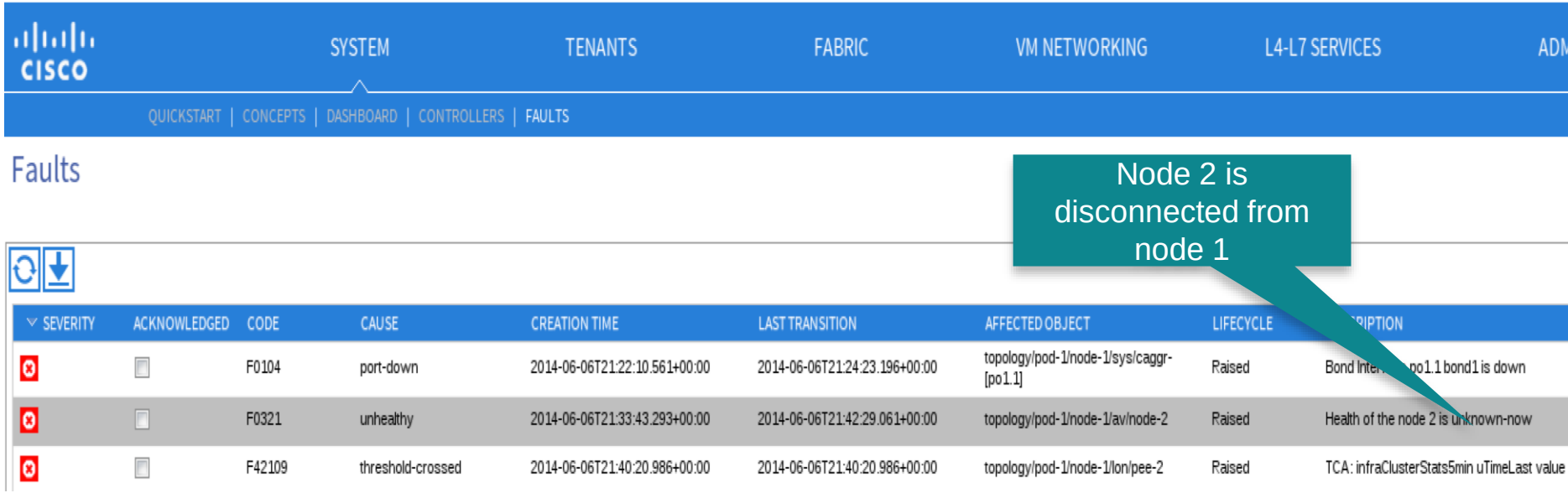
- 1 APIC : obviously no fault tolerance. Data lost for single failure
- 2 APIC : Write unavailability with single failure. Can recover if APIC is replaced
- 3 APIC : If 1 APIC lost, other two can elect new leader and continue writes. If 2 APIC lost we go to minority behavior (no writes, only read)

Troubleshooting cluster

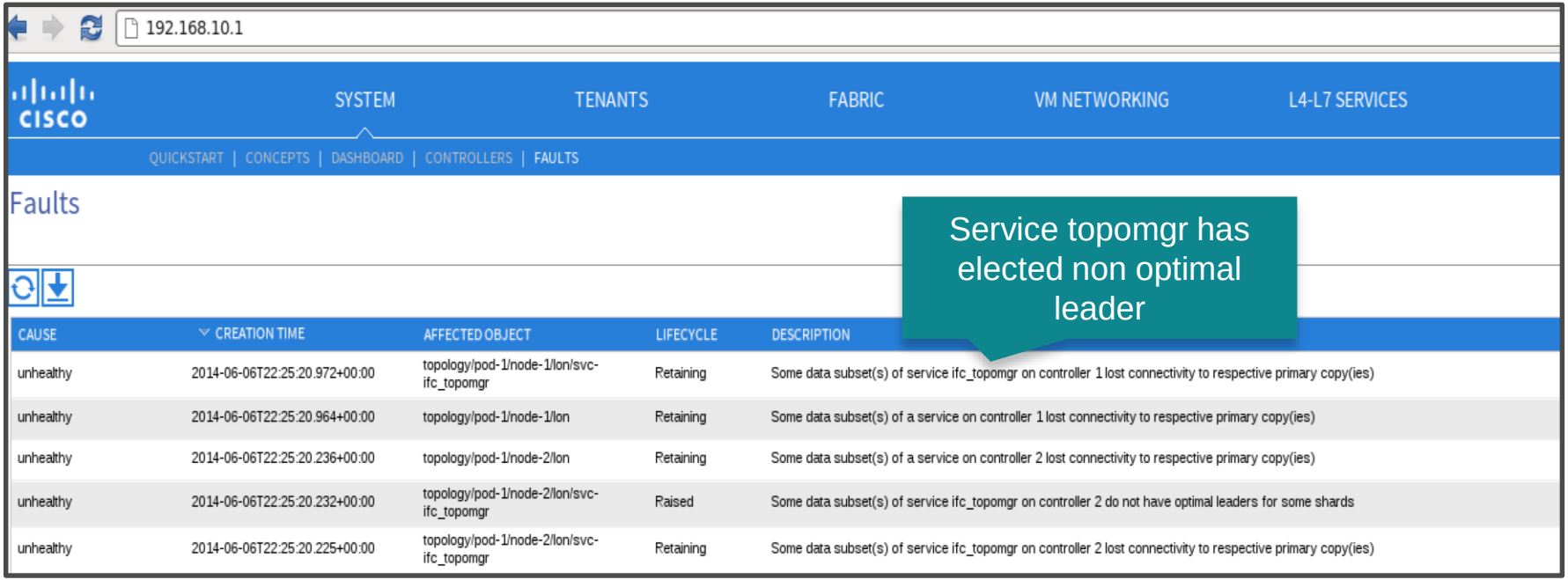
Clustering Faults when APIC/Service is down

- fltInfraReplicaReplicaState
  - Failed to bring-up Service: name instance (shrdId, rplId)
- fltInfraReplicaDatabaseState
  - Corruption in datastore Service: name instance (shrdId, rplId)
- fltInfraServiceHealth
  - Some data subset(s) of a service id on controller id lost connectivity to respective primary copy
- fltInfraWiNodeHealth
  - Health of the node id

Fault when one APIC is down/disconnected



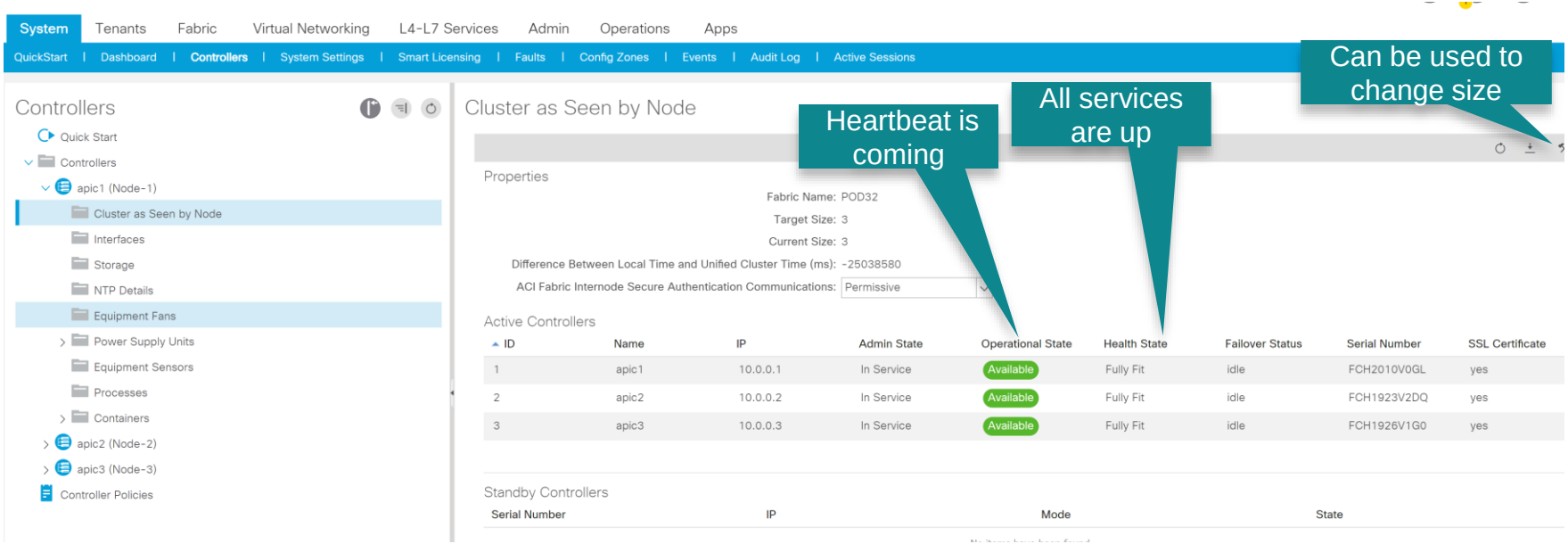
Fault when one DME process is down



CAUSE	CREATION TIME	AFFECTED OBJECT	LIFECYCLE	DESCRIPTION
unhealthy	2014-06-06T22:25:20.972+00:00	topology/pod-1/node-1/lon/svc-ifc_topomgr	Retaining	Some data subset(s) of service ifc_topomgr on controller 1 lost connectivity to respective primary copy(ies)
unhealthy	2014-06-06T22:25:20.964+00:00	topology/pod-1/node-1/lon	Retaining	Some data subset(s) of a service on controller 1 lost connectivity to respective primary copy(ies)
unhealthy	2014-06-06T22:25:20.236+00:00	topology/pod-1/node-2/lon	Retaining	Some data subset(s) of a service on controller 2 lost connectivity to respective primary copy(ies)
unhealthy	2014-06-06T22:25:20.232+00:00	topology/pod-1/node-2/lon/svc-ifc_topomgr	Raised	Some data subset(s) of service ifc_topomgr on controller 2 do not have optimal leaders for some shards
unhealthy	2014-06-06T22:25:20.225+00:00	topology/pod-1/node-2/lon/svc-ifc_topomgr	Retaining	Some data subset(s) of service ifc_topomgr on controller 2 lost connectivity to respective primary copy(ies)

See current cluster state in UI

Here cluster view from APIC-1 Point of view



Clustering user command

See current cluster size and state of APICs

```
apic2# show controller
Fabric Name          : POD32
Operational Size     : 3
Cluster Size         : 3
Time Difference      : -25038579
Fabric Security Mode : permissive

 ID    Pod   Address          In-Band IPv4     In-Band IPv6    OOB IPv4         OOB IPv6                          Version            Flags  Serial Number     Health
 ----  ----  ---------------  ---------------  --------------  ---------------  --------------------------------  -----------------  -----  ----------------  ---------
---------
 1     1     10.0.0.1         10.99.98.1       fc00::1         10.48.25.60      fe80::278:88ff:fea3:2a9a          3.2(1m)            crva-  FCH2010V0GL       fully-fit
 2*    1     10.0.0.2         10.99.98.2       fc00::1         10.48.25.61      fe80::dab1:90ff:feff:f392         3.2(1m)            crva-  FCH1923V2DQ       fully-fit
 3     2     10.0.0.3         10.99.98.3       fc00::1         10.48.25.62      fe80::1a8b:9dff:fe43:dd40         3.2(1m)            crva-  FCH1926V1G0       fully-fit

Flags - c:Commissioned | r:Registered | v:Valid Certificate | a:Approved | f/s:Failover fail/success
(*)Current (~)Standby
```

Show controller detail

```
apic2# show controller detail
ID                    : 1
Name                  : apic1
UUID                  : fbf37130-5874-11e6-9126-
23c1fa8b9c21
Pod ID                : 1
Address               : 10.0.0.1
In-Band IPv4 Address  : 10.99.98.1
In-Band IPv6 Address  : fc00::1
OOB IPv4 Address      : 10.48.25.60
OOB IPv6 Address      : fe80::278:88ff:fea3:2a9a
Serial Number         : FCH2010V0GL
Version               : 3.2(1m)
Commissioned          : in-service
Registered            : available
Approved              : yes
Valid Certificate     : yes
Validity Start        : 2016-04-16T00:21:55.000+00:00
Validity End          : 2026-04-16T00:31:55.000+00:00
Up Time               : 55:07:04:03.000
Health                : fully-fit
Failover Status       : idle
```

```
ID                    : 2*
Name                  : apic2
UUID                  : 4056b17a-5875-11e6-9898-
6b5b4f30e66d
Pod ID                : 1
Address               : 10.0.0.2
In-Band IPv4 Address  : 10.99.98.2
In-Band IPv6 Address  : fc00::1
OOB IPv4 Address      : 10.48.25.61
OOB IPv6 Address      : fe80::dab1:90ff:feff:f392
Serial Number         : FCH1923V2DQ
Version               : 3.2(1m)
Commissioned          : in-service
Registered            : available
Approved              : yes
Valid Certificate     : yes
Validity Start        : 2015-07-24T05:08:37.000+00:00
Validity End          : 2025-07-24T05:18:37.000+00:00
Up Time               : 55:07:04:02.000
Health                : fully-fit
Failover Status       : idle
…
```

Check acidiag avread

Good to check Cluster health

```
apic1# acidiag avread
Local appliance ID=1 ADDRESS=10.0.0.1 TEP ADDRESS=10.0.0.0/16 CHASSIS_ID=e52e3fc8-4cfa-11e6-af0d-81a89d6fa5ec
Cluster of 3 lm(t):1(2016-07-21T07:31:09.458+00:00) appliances (out of targeted 3 lm(t):1(2016-07-21T07:31:12.350+00:00)) with
FABRIC_DOMAIN name=POD13 set to version=apic-1.2(2h) lm(t):1(2016-07-21T07:31:22.763+00:00); discoveryMode=PERMISSIVE lm(t):0(1970-01-
01T00:00:00.003+00:00)
        appliance id=1  address=10.0.0.1 lm(t):1(2016-07-21T07:30:45.112+00:00) tep address=10.0.0.0/16 lm(t):1(2016-07-
21T07:30:45.112+00:00) oob address=10.48.22.94/24 lm(t):1(2016-07-21T07:31:09.700+00:00) version=1.2(2h) lm(t):1(2016-07-
21T07:31:10.690+00:00) chassisId=e52e3fc8-4cfa-11e6-af0d-81a89d6fa5ec lm(t):1(2016-07-21T07:31:10.690+00:00) capabilities=0X7FFFFFFF--
0--0X7 lm(t):1(2016-07-21T07:36:22.835+00:00) rK=(stable,present,0X207373642D687373) lm(t):1(2016-07-21T07:31:09.707+00:00)
aK=(stable,present,0X207373642D687373) lm(t):1(2016-07-21T07:31:09.707+00:00) cntrlSbst=(APPROVED, FCH1852V3LE) lm(t):1(zeroTime)
commissioned=YES lm(t):1(zeroTime) registered=YES lm(t):1(2016-07-21T07:30:45.112+00:00) active=YES(2016-07-21T07:30:45.112+00:00)
health=(applnc:255 lm(t):1(2016-07-21T07:32:54.051+00:00) svc's)
        appliance id=2  address=10.0.0.2 lm(t):1(2016-07-21T07:31:21.488+00:00) tep address=10.0.0.0/16 lm(t):2(2016-07-
18T14:49:06.317+00:00) oob address=10.48.22.95/24 lm(t):1(2016-09-01T11:31:23.647+00:00) version=1.2(2h) lm(t):2(2016-07-
21T07:31:10.698+00:00) chassisId=4b00064a-3146-11e6-bf20-ed9eb8a4f642 lm(t):1(2016-07-21T07:31:21.488+00:00) capabilities=0X7FFFFFFF--
0--0X7 lm(t):2(2016-07-21T07:34:44.120+00:00) rK=(stable,present,0X207373642D687373) lm(t):1(2016-07-21T07:31:11.958+00:00)
aK=(stable,present,0X207373642D687373) lm(t):1(2016-07-21T07:31:11.958+00:00) cntrlSbst=(APPROVED, FCH1906V223) lm(t):0(zeroTime)
commissioned=YES lm(t):1(2016-07-21T07:31:09.458+00:00) registered=YES lm(t):1(2016-07-18T15:19:22.856+00:00) active=YES(2016-07-
21T07:31:10.534+00:00) health=(applnc:255 lm(t):2(2016-07-21T07:32:53.974+00:00) svc's)
        appliance id=3  address=10.0.0.3 lm(t):1(2016-07-21T07:31:21.488+00:00) tep address=10.0.0.0/16 lm(t):3(2016-06-
13T16:49:15.233+00:00) oob address=10.48.22.96/24 lm(t):1(2016-09-01T11:31:23.654+00:00) version=1.2(2h) lm(t):3(2016-07-
21T07:31:10.668+00:00) chassisId=aa08b322-3186-11e6-87c2-7be412f9e01b lm(t):1(2016-07-21T07:31:21.488+00:00) capabilities=0X7FFFFFFF--
0--0X7 lm(t):3(2016-07-21T07:35:09.293+00:00) rK=(stable,present,0X207373642D687373) lm(t):1(2016-07-21T07:31:11.960+00:00)
aK=(stable,present,0X207373642D687373) lm(t):1(2016-07-21T07:31:11.961+00:00) cntrlSbst=(APPROVED, FCH1904V2RP) lm(t):1(2016-06-
13T16:49:40.020+00:00) commissioned=YES lm(t):1(2016-07-21T07:31:09.458+00:00) registered=YES lm(t):1(2016-07-18T15:19:22.856+00:00)
active=YES(2016-07-21T07:31:10.421+00:00) health=(applnc:255 lm(t):3(2016-07-21T07:32:54.182+00:00) svc's)
--------------------------------------------
clusterTime=<diff=20734 common=2016-09-27T11:54:27.012+00:00 local=2016-09-27T11:54:06.278+00:00 pF=<displForm=0 offsSt=0 offsVlu=0
lm(t):1(2016-07-18T15:22:12.695+00:00)>>
--------------------------------------------
```

Cluster issue – what to check in avread

- Get from each APIC – acidiag avread
- Compare and check :
  - Fabric name must be the same
  - Chassis Id for each appliance should be the same on the 3 apics
  - Cluster version must be the same on each appliance and must match cluster version
  - Health 255 (means replica are all fine)

Acidiag rvread – replica vector – healthy cluster

List of shard and replica

List of DME (process)

```
apic1# acidiag rvread
\- unexpected state;    /-unexpected mutator;
s->  1   2   3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32lcl
r->123123123123123123123123123123123123123123123123123123123123123123123123123123123123123123123123lcl
   1
   2
   3
   4
   5
..
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34
Replicas are in expected states and are mutated by proper apic's
```

Quick check on replica DB health

Healthy

```
apic1# acidiag rvreadle
Optimal leader for all shards
-------------------------------------------
clusterTime=<diff=-25038580 common=2018-08-27T05:32:07.995+00:00 local=2018-08-27T12:29:26.575+00:00
pF=<displForm=0 offsSt=0 offsVlu=0 lm(t):3(2016-10-27T08:40:24.544+00:00)>>
```

Non Healthy

```
apic1# acidiag rvreadle
No leader for shards : 6:17,33:1
Non optimal leader for shards : 6:17,33:1
-------------------------------------------
clusterTime=<diff=-25038580 common=2018-08-27T05:32:07.995+00:00 local=2018-08-27T12:29:26.575+00:00
pF=<displForm=0 offsSt=0 offsVlu=0 lm(t):3(2016-10-27T08:40:24.544+00:00)>>
```

Acidiag rvread – replica vector – unhealthy cluster

Here service 6 is not healthy (policymgr)

Shard ->
Replica ->

Service (DME)

```
apic1# acidiag rvread
\- unexpected state;    /-unexpected mutator;
s->  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 321cl
r->123123123123123123123123123123123123123123123123123123123123123123123123123123123123123123123123123cl
  1
  2
  3
  4
  5
  6\\\   \\\\\\      \\\   \\\\\\      \\\   \\\\\\      \\\   \\\\\\      \\\   \\\\\\      \\\
  7
  8
  9
 10
 ..
 39
Some replicas are in not expected states
---------------------------------------------
clusterTime=<diff=2444 common=2016-09-27T14:14:43.669+02:00 local=2016-09-27T14:14:41.225+02:00
pF=<displForm=0 offsSt=0 offsVlu=7200 lm(t):3(2016-09-07T15:07:03.763+02:00)>>
```

Rvread output

- "acidiag rvread" shows replica which are not healthy
- "acidiag rvread <svc><shard><replica>" to see the state of one replica

```
admin@apic1:~> acidiag rvread
Replicas are in expected states
------------------------------------------------
clusterTime=<diff=0 common=2014-05-29T21:22:54.096+00:00 local=2014-05-29T21:22:54.096+00:00 pF=<displForm=0 offsSt=0 offsVlu=0 lm(t):
3(2014-05-29T19:34:50.866+00:00)>>

admin@apic1:~> acidiag rvread 9 1 2
(9,1,2)  st:6 lm(t):2(2014-05-29T19:51:39.174+00:00) le: reSt:FOLLOWER voGr:0 cuTerm:0x3 lCoTe:0x1 lCoIn:0x80000000000055c veFiSt:0x9
veFiEn:0x9 lm(t):2(2014-05-29T19:52:34.011+00:00) lastUpdt 2014-05-29T21:22:59.017+00:00
------------------------------------------------
clusterTime=<diff=0 common=2014-05-29T21:23:02.589+00:00 local=2014-05-29T21:23:02.589+00:00 pF=<displForm=0 offsSt=0 offsVlu=0 lm(t):
3(2014-05-29T19:34:50.866+00:00)>>
```

Svc, shard, replica

State 6=UP

APIC where it is running

Leadership state

Acidiag rvread for policymgr

"acidiag rvread <svc><shard><replica>" to see the state of one replica

Shard2 is healthy

```
apic1# acidiag rvread 6 2
(6,2,1)  st:6 lm(t):2(2016-09-07T15:08:02.516+02:00) le: reSt:LEADER voGr:0 cuTerm:0x1a5 lCoTe:0x1a4
lCoIn:0x1000000000901491 veFiSt:0x53 veFiEn:0x53 lm(t):2(2016-09-07T15:08:02.492+02:00) stMmt:1 lm(t):0(zeroTime)
lastUpdt 2016-09-27T14:19:59.400+02:00
(6,2,2)  st:6 lm(t):3(2016-09-21T16:27:36.442+02:00) le: reSt:FOLLOWER voGr:0 cuTerm:0x1a6 lCoTe:0x1a5
lCoIn:0x100000000097f979 veFiSt:0x4e9 veFiEn:0x4e9 lm(t):3(2016-09-21T16:27:36.409+02:00) stMmt:1 lm(t):0(zeroTime)
lastUpdt 2016-09-27T14:19:59.400+02:00
(6,2,3)  st:6 lm(t):1(2016-09-07T15:07:07.852+02:00) le: reSt:FOLLOWER voGr:0 cuTerm:0x1a6 lCoTe:0x1a5
lCoIn:0x100000000097f979 veFiSt:0x33 veFiEn:0x33 lm(t):1(2016-09-21T16:27:36.695+02:00) stMmt:1 lm(t):0(zeroTime) lp:
clSt:2 lm(t):1(2016-09-07T14:18:46.436+02:00) dbSt:2 lm(t):1(2016-09-07T14:18:23.152+02:00) dbCrTs:2016-09-
07T14:18:23.152+02:00 lastUpdt 2016-09-21T16:27:36.695+02:00
----------------------------------------------
clusterTime=<diff=2444 common=2016-09-27T14:19:59.592+02:00 local=2016-09-27T14:19:57.148+02:00 pF=<displForm=0
offsSt=0 offsVlu=7200 lm(t):3(2016-09-07T15:07:03.763+02:00)>>
```

Shard3 is not up

```
apic1# acidiag rvread 6 3
(6,3,1)  st:1 lm(t):3(2016-09-27T14:19:55.654+02:00) le: reSt:LEADER voGr:0 cuTerm:0x6c692 lCoTe:0 lCoIn:0
veFiSt:0x143ac8 veFiEn:0x143ac8 lm(t):3(2016-09-27T14:19:55.640+02:00) stMmt:1 lm(t):0(zeroTime) lastUpdt 2016-09-
27T14:19:56.370+02:00
Replica above IS NOT in the state UP

(6,3,2)  st:1 lm(t):1(2016-09-27T14:19:52.981+02:00) le: reSt:FOLLOWER_ELEC voGr:1 cuTerm:0x6c692 lCoTe:0 lCoIn:0
veFiSt:0xcb8c1 veFiEn:0xcb8c1 lm(t):1(2016-09-27T14:19:55.371+02:00) stMmt:1 lm(t):0(zeroTime) lp: clSt:0
lm(t):0(zeroTime) dbSt:0 lm(t):0(zeroTime) dbCrTs:zeroTime lastUpdt 2016-09-27T14:19:55.371+02:00
Replica above IS NOT in the state UP

(6,3,3)  st:1 lm(t):2(2016-09-21T16:27:32.883+02:00) le: reSt:FOLLOWER_ELEC voGr:1 cuTerm:0x4e560 lCoTe:0 lCoIn:0
veFiSt:0x8a2a4 veFiEn:0x8a2a4 lm(t):2(2016-09-21T16:27:30.079+02:00) stMmt:1 lm(t):0(zeroTime) lastUpdt 2016-09-
27T14:19:54.398+02:00
Replica above IS NOT in the state UP
```

List of Srv ID

- SVC_ID_UNKNOWN = 0,
- SVC_ID_ANY = SVC_ID_UNKNOWN,
- SVC_ID_ifc_cliD = 1,
- SVC_ID_ifc_controller = 2,
- SVC_ID_ifc_eventmgr = 3,
- SVC_ID_ifc_extXMLApi = 4,
- SVC_ID_ifc_policyelem = 5,
- **SVC_ID_ifc_policymgr = 6,**
- SVC_ID_ifc_reader = 7,
- SVC_ID_ifc_ae = 8,
- SVC_ID_ifc_topomgr = 9,
- SVC_ID_ifc_observer = 10,
- SVC_ID_ifc_dbgr = 11,
- SVC_ID_ifc_observerelem = 12,
- SVC_ID_ifc_dbgrelem = 13,
- SVC_ID_ifc_vmmmgr = 14,
- SVC_ID_ifc_nxosmock = 15,
- SVC_ID_ifc_bootmgr = 16,
- SVC_ID_ifc_appliancedirector = 17,
- SVC_ID_ifc_adrelay = 18,
- SVC_ID_ifc_ospaagent = 19,
- SVC_ID_ifc_vleafelem = 20,
- SVC_ID_ifc_dhcpd = 21,
- SVC_ID_ifc_scripthandler = 22,
- SVC_ID_ifc_idmgr = 23,
- SVC_ID_ifc_ospaelem = 24,
- SVC_ID_ifc_osh = 25,
- SVC_ID_ifc_opflexagent = 26,
- SVC_ID_ifc_opflexelem = 27,
- SVC_ID_ifc_confelem = 28,
- SVC_ID_ifc_vtap = 29,

Man acidiag

```
Service IDs:
       1 - cliD
       2 - controller
       3 - eventmgr
       4 - extXMLApi
       5 - policyelem
       6 - policymgr
       7 - reader
       8 - ae
       9 - topomgr
       10 - observer
       11 - dbgr
       12 - observerelem
       13 - dbgrelem
       14 - vmmmgr
       15 - nxosmock
       16 - bootmgr
       17 - appliancedirector
       18 - adrelay
       19 - ospaagent
       20 - vleafelem
       21 - dhcpd
       22 - scripthandler
       23 - idmgr
       24 - ospaelem
       25 - osh
       26 - opflexagent
```

Verify that all DME are running

```
apic1# ps -ef | egrep svc
root       704     1  5 Sep07 ?        1-00:43:33 /mgmt//bin/svc_ifc_appliancedirector.bin --x
root       707     1  4 Sep07 ?        20:35:24 /mgmt//bin/svc_ifc_bootmgr.bin --x
ifc        718     1  4 Sep07 ?        21:05:35 /mgmt//bin/svc_ifc_observer.bin --x
ifc        725     1  4 Sep07 ?        20:53:29 /mgmt//bin/svc_ifc_vmmmgr.bin --x
ifc        729     1  4 Sep07 ?        22:59:06 /mgmt//bin/svc_ifc_dbgr.bin --x
ifc        731     1  4 Sep07 ?        20:25:28 /mgmt//bin/svc_ifc_topomgr.bin --x
ifc        739     1  4 Sep07 ?        21:02:30 /mgmt//bin/svc_ifc_eventmgr.bin --x
ifc        740     1  6 Sep07 ?        1-09:17:16 /mgmt//bin/svc_ifc_policymgr.bin --x
ifc        742     1  2 Sep07 ?        12:06:59 /mgmt//bin/svc_ifc_reader.bin --x
ifc        743     1  2 Sep07 ?        12:51:13 /mgmt//bin/svc_ifc_vtap.bin --x
ifc        748     1  4 Sep07 ?        20:06:24 /mgmt//bin/svc_ifc_idmgr.bin --x
ifc        750     1  4 Sep07 ?        22:04:52 /mgmt//bin/svc_ifc_scripthandler.bin --x
root       751     1  3 Sep07 ?        15:51:36 /mgmt//bin/svc_ifc_ae.bin --x
```

You can start/restart a dme from acidiag

```
apic1# acidiag restart -h
usage: acidiag restart [-h]


{xinetd,mgmt,ae,lldpad,observer,dbgr,idmgr,dhcpd,snmpd,eventmgr,policymgr,reader,bootmgr,topomgr,nginx,vmm
mgr,appliancedirector,scripthandler}

positional arguments:

{xinetd,mgmt,ae,lldpad,observer,dbgr,idmgr,dhcpd,snmpd,eventmgr,policymgr,reader,bootmgr,topomgr,nginx,vmm
mgr,appliancedirector,scripthandler}
```

Check possible core

Below list of core for APIC 1 (node 1)

```
pod2-apic1# show core | egrep "^ 1 "
 1        2016-02-24T16:29:04.0  156221917  policymgr     29410    /dmecores/svc_ifc_policymgr.bin_log.2941  139
 1        2016-02-24T16:27:03.0  146870941  policymgr     26575    /dmecores/svc_ifc_policymgr.bin_log.2657  139
 1        2016-02-24T16:26:43.0  146275227  policymgr     25645    /dmecores/svc_ifc_policymgr.bin_log.2564  139
 1        2016-02-24T16:26:24.0  146396293  policymgr     24652    /dmecores/svc_ifc_policymgr.bin_log.2465  139
 1        2016-02-24T16:26:05.0  146163491  policymgr     24441    /dmecores/svc_ifc_policymgr.bin_log.2444  139
 1        2016-02-24T16:25:46.0  147261162  policymgr     22999    /dmecores/svc_ifc_policymgr.bin_log.2299  139
 1        2016-02-24T16:25:01.0  149884251  policymgr     21346    /dmecores/svc_ifc_policymgr.bin_log.2134  139
 1        2016-02-24T16:22:56.0  146542821  policymgr     18216    /dmecores/svc_ifc_policymgr.bin_log.1821  139
 1        2016-02-24T16:22:32.0  146534074  policymgr     18012    /dmecores/svc_ifc_policymgr.bin_log.1801  139
 1        2016-02-24T16:22:09.0  146563822  policymgr     16783    /dmecores/svc_ifc_policymgr.bin_log.1678  139
 1        2016-02-24T16:21:44.0  147050431  policymgr     15127    /dmecores/svc_ifc_policymgr.bin_log.1512  139
 1        2016-02-24T16:21:06.0  145876603  policymgr     15031    /dmecores/svc_ifc_policymgr.bin_log.1503  139
 1        2016-02-24T16:20:43.0  168994143  policymgr     9319     /dmecores/svc_ifc_policymgr.bin_log.9319  139
```

Cluster issue troubleshooting

- Always get from each APIC :
  - acidiag avread – check soft of cluster and each apic, check uuid match
  - acidiag rvread – if a process has unhealthy shard, check if it is running
  - Ps –ef -- to check if DME are all running
  - Show cores

  - Tech support
  - Open TAC case before attempting recovery

Summary cluster

- Cluster is fully fit if
  1. Heartbeat goes through inband (IP connectivity)
  2. Avread matches (uuid, fabric name, ….)
  3. Replica vector are fine

Recommendation

If cluster is not fully fit do not attempt to recover by yourself

no apic reload, no service restart, no erase config setup of apic → The risk to make things worst is high

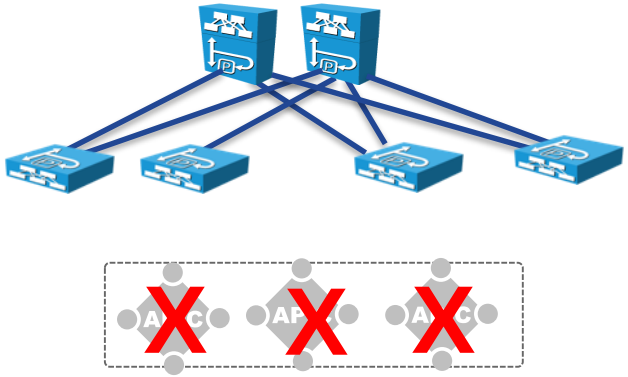
Call TAC

Best practices :

- Take periodic snapshots of the fabric configuration
- Export the ACI fabric configuration periodically

What if all APICs are down?

- Traffic Forwarding Continues for new and existing sessions:
  - Leafs are completely self sufficient when it comes to traffic forwarding and policy enforcement
- Link failures are recovered
- New VMM Endpoint Attach may or may not work:
  - It depends on the Resolution and deployment immediacy
- Vmotion may or may not work:
  - It depends on the Resolution and the deployment Immediacy
- *If you have a snapshot TAC or Cisco BU can help you doing doing Fabric ID recovery*



APIC cluster split Recovery through OOB (3.2 and after)

Introduction

- Intra fabric communication (‘APIC to APIC’ and ‘APIC to switch’) happens over a reserved IP subnet called infra subnet.
- Intra Subnet has following benefits
  - Secure, it’s a private subnet which is not externally visible.
  - Bandwidth guarantee
- Infra subnet goes over ACI fabric. This means APIC cluster operation depends very much on the fabric, which it is managing.

Fabric Failure and Recovery

- A fabric can fail due to multiple reasons as:
  - Link and Node failure
  - Configure and User error
- A link/node failures could be fixed independently, config/user error needs to be fixed on APIC cluster.
- Controller may misconfigure ports of switches in a policy, thus disabling some ports and leading to a broken fabric.
- A broken fabric can result into a split cluster and network partitions; this is a chicken and egg problem.
- To fix the fabric we first need a healthy cluster but for healthy cluster we need to fix the fabric first.

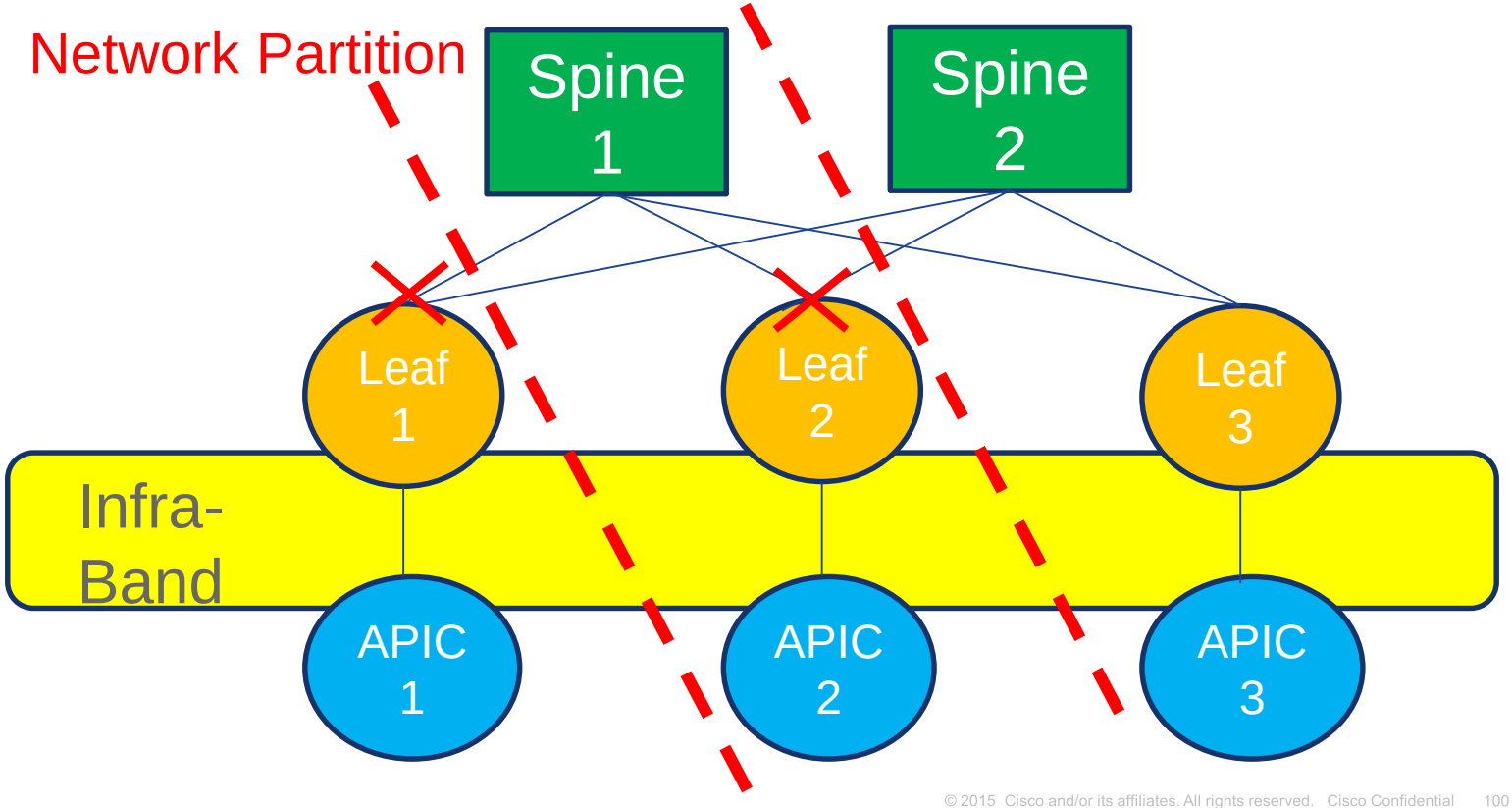
Steps of Recovering Misconfigured Fabric

- Without OOB, steps to recover a fabric due to misconfiguration and user error are as follows:
  - Shutdown impacted controllers.
  - Put all APICs in the same network partition.
  - Turn on controllers.
  - Ensure a healthy cluster is formed.
  - Correct the misconfigured policy.
  - Clean reload the faulty switches
  - Move controllers back to their original places.
- If network partitions occur in multi-pod, some controllers may need change their pod id after their moves.
- It is time consuming to recover a misconfigured fabric

Recovering Fabric Using OOB

- With OOB, steps to recover a fabric due to misconfiguration and user error are as follows:
  - Switch the cluster interface over to OOB using “acidiag oob enable”.
  - Wait for a healthy OOB cluster is formed using “show controller” to check if address is OOB IP
  - Fix the mis-configued issue in controller and clean reload faulty switches.
  - Change the cluster interface back to Infra using “acidiag oob disble”.
- OOB makes it simple to recover the misconfigured fabric.
- OOB cluster can be used to fix network partition issues listed on following pages.

OOB Use Case 1: No Majority



OOB Use Case 1: No Majority

- Use Case 1
  - A cluster consisting of 3 APICs
  - APIC-1 is a shard leader of PM.
  - APIC-1 configures ports improperly in a policy.
  - Leaf-1 and Leaf-2 lose connectivity.
  - Three network partitions are generated.
  - None of partitions has majority of APICs.
  - No leader can be elected to correct the improper policy.

Switching Between OOB and Infra IP

- The following operations assume that OOB IP addresses are configured and active
- There will be a policy to trigger each APIC to switch from infra IP to OOB IP and back. This policy need to be run on each APIC for entire cluster to move to OOB.
  - API:
    - https://ifc1.insieme.local/api/node/mo/topology/pod-1/node-1/lon.xml
    - <infraLoNode isOobNotInfra="yes"/>
    - https://ifc2.insieme.local/api/node/mo/topology/pod-1/node-2/lon.xml
    - <infraLoNode isOobNotInfra="yes"/>
    - https://ifc3.insieme.local/api/node/mo/topology/pod-1/node-3/lon.xml
    - <infraLoNode isOobNotInfra="yes"/>
  - acidiag:
    - acidiag oob {enable / disable}

Switching Between OOB and Infra IP

- To form a OOB cluster, it is mandatory for all of APICs to switch over to OOB from infra-band one by one.
  - Command “acidiag oob enable” is needed for every APIC.
  - Given a broken fabric, there is no way to tell all nodes to switch over to OOB simultaneously.
- APICs in OOB are unable to communicate with those in infra-band.
- It may take up to 7 minutes for a cluster node to change its interface.
- If time interval between two OOB commands in one node is less than 7 minutes, the latter one will be ignored.
- OOB commands can run in parallel in different cluster nodes.

Checking Cluster Interface

- There are two ways to know if a cluster interface has switched over to OOB from infra-band?
- Using a CLI command called ”show controller” to know the cluster interface of nodes.
  - ”show controller” shows addresses of controllers.
  - For infra-band nodes, their addresses are in the form of 10.0.x.y, where x is used to identify different pods and y for different controllers.
  - For OOB nodes, their addresses are not in the format of 10.0.x.y
- Using Visore (moquery or API)
  - Management object name: infraLoNode.
  - Property name: isOobNotInfra.
  - “isOobNotInfra = yes” means the cluster is in out-of-band.
  - “isOobNotInfra = no” means the cluster is in infra-band.

See address field

Change apic1 to OOB

```
apic1# acidiag oob enable
It takes up to 7 minutes to change cluster interface
Connection will be lost
```

Apic2 and 3 are still on inband hence lost apic1

```
apic2# show controller
Fabric Name          : POD32
Operational Size     : 3
Cluster Size         :
Time Difference      : -25038581
Fabric Security Mode : permissive

 ID    Pod   Address      In-Band IPv4     In-Band IPv6   OOB IPv4         OOB IPv6                          Version          Flags  Serial Number     Health
 ----  ----  -----------  ---------------  -------------  ---------------  --------------------------------  ---------------  -----  ----------------  ---------
 1           10.0.0.1                                                                                        crva-  FCH2010V0GL       unknown-now
 2*    1     10.0.0.2     10.99.98.2       fc00::1        10.48.25.61      fe80::dab1:90ff:feff:f392         3.2(1m)          crva-  FCH1923V2DQ       fully-fit
 3     2     10.0.0.3     10.99.98.3       fc00::1        10.48.25.62      fe80::1a8b:9dff:fe43:dd40         3.2(1m)          crva-  FCH1926V1G0       fully-fit
```

Apic2 uses inb still (10.0.0.x)

Change apic2 and 3 to oob. After that Cluster will form back using OOB

```
apic1# show controller
Fabric Name          : POD32
Operational Size     : 3
Cluster Size         : 3
Time Difference      : -25038577
Fabric Security Mode : permissive

 ID    Pod   Address          In-Band IPv4     In-Band IPv6   OOB IPv4         OOB IPv6                          Version          Flags  Serial Number     Health
 ----  ----  ---------------  ---------------  -------------  ---------------  --------------------------------  ---------------  -----  ----------------  ---------
 1*    1     10.48.25.60      10.99.98.1       fc00::1        10.48.25.60      fe80::278:88ff:fea3:2a9a          3.2(1m)          crva-  FCH2010V0GL       fully-fit
 2     1     10.48.25.61      10.99.98.2       fc00::1        10.48.25.61      fe80::dab1:90ff:feff:f392         3.2(1m)          crva-  FCH1923V2DQ       fully-fit
 3     2     10.48.25.62      10.99.98.3       fc00::1        10.48.25.62      fe80::1a8b:9dff:fe43:dd40         3.2(1m)          crva-  FCH1926V1G0       fully-fit
```

Now each apic switched to OOB for HB

Check if OOB was enable on an apic

```
apic1# egrep oob /firmware/acidiag.log
2018-08-27 13:00:27,544 | INFO | admin | oob | enable
2018-08-27 13:15:19,198 | INFO | admin | oob | disable
```

```
apic1# zgrep "cluster interface" /var/log/dme/log/svc_ifc_appliancedirector.bin.log*
svc_ifc_appliancedirector.bin.log.26.gz:23985||18-08-27 13:02:29.376+00:00||adrs_av||DBG4||||Switch cluster interface over to
OOB||../appliance/director/./ApplianceDirector.cpp||160
svc_ifc_appliancedirector.bin.log.27:3289||18-08-27 13:17:19.311+00:00||adrs_av||DBG4||||Switch cluster interface over to
infra||../appliance/director/./ApplianceDirector.cpp||173
```

Log on Leaf

ACI switch shell

- Bash : default
- vsh : should be deprecated one day but for now still best for regular nexus process (especially routing, ….)
- (fixed chassis) vsh_lc : platform side – contains low level process talking directly to ASIC.
- Modular chassis : one prompt per LC and per fabric card
  - vsh and then attach mod x

Directory for logs

- Running logs in
  - /var/sysmgr/tmp_logs/
- Old logs in (file rotate)
  - /var/log/dme/oldlog

Which are the log file

- DME logs (DME on switch)
  - Nginx, policyelem, opflexelem,eventmgr
- ACI specific nxos logs :
  - Epm, epmc – end point learning, aging
  - eltm, eltmc – Construct programmation on leaf such as vlan, vrf, interface, tunnel,…
  - istack – cpu traffic
  - Sensor.log – env sensor measure (temp,…)
  - …

What about regular nxos and protocol logs

- No Debug available on aci switch
- But event trace for almost everyting
- Look for event-history or trace in cli chains (in bash/vsh and/or in vsh_lc). Mostly exactly similar as in regular nexus switches.
- Example in vsh:
  - Show ip ospf event-history
  - Show bgp event-history
  - Show ntp internal event-history
  - Show lacp internal event-history
- Example in vsh_lc:
  - Show platform internal bcm event-history trace port x (link up/down event)
  - Show system internal aclqos event-history …

You want to know them all ?

```
pod2-leaf1# show cli list | egrep "event-history" | wc
   1025    7280   56415
Pod2-leaf1# show cli list  | egrep "show.*trace "  | wc
    256    3033   22718
pod2-leaf1# show cli  list | egrep "^show.*ospf .*event"
show ip ospf <str> internal event-history msgs
show ip ospf <str> internal event-history statistics
show ip ospf <str> internal event-history adjacency
show ip ospf <str> internal event-history event
show ip ospf <str> internal event-history ha
show ip ospf <str> internal event-history flooding
show ip ospf <str> internal event-history lsa
show ip ospf <str> internal event-history spf
show ip ospf <str> internal event-history redistribution
show ip ospf <str> internal event-history ldp
show ip ospf <str> internal event-history te
show ip ospf <str> internal event-history rib
show ip ospf <str> internal event-history hello
show ip ospf <str> internal event-history spf-trigger
show ip ospf <str> internal event-history cli
show ip ospf <str> internal event-history verbose
..
```

Questions?