
cisco Live!
Let's go

The bridge to possible

Cisco UCS 5th Gen VIC

Topology, packet flow, features and performance

Eldho Jacob, Leader Product Management
@eljacob

cisco Live!

BRKDCN-2669

Agenda

- VIC 15000 series Introduction
- Topologies and Traffic Flow
- VIC Features : Secure boot, QinQ Tunneling, SR-IOV, Physical NIC mode
- VIC Performance
- Conclusion

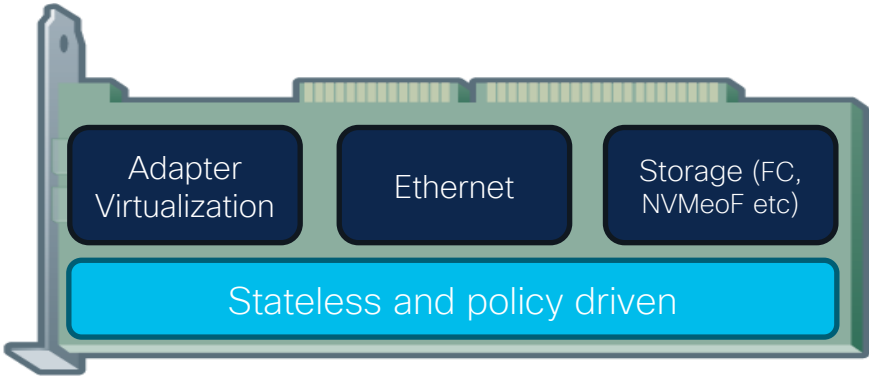
UCS Virtual Interface Card

Flexible

Programmable

TCO Benefit

UCS VIC Family



Cisco VIC Evolution

Industry First ASIC with dynamic IOV support

1st Gen

- *2x10G*
- *x16 PCIe Gen1*
- 128 virtual PCIe devices
- Hypervisor Bypass for ESX, KVM

2009

2nd Gen

- *2x10G/40G*
- *x16 PCIe Gen* 2
- CNA
- *256 virtual PCIe devices*
- *Single Wire Mgmt*
- *Flow classification (Netflow)*
- *RSS and NetQueue*

2012

3rd Gen

- *2x10G/40G*
- *x16 PCIe Gen 3*
- *CNA*
- Single Wire Mgmt
- RoCEv1
- Overlays: NVGRE, VXLAN
- usNIC, VM-FEX, DPDK
- Flow classification – Netflow
- *NPU For Packet Encap/Decap*
- *RSS, NetQueue and VMQ*

2014

4th Gen

- 10G/**25G**/40G/**100G**
- **x16 PCIe Gen 3**
- CNA, **X-Series**
- **512 virtual PCIe Devices**
- Single Wire Mgmt
- **NVMEoF** (FC-NVMe, **RoCEv2**)
- Overlays: NVGRE, VXLAN, Geneve
- usNIC, VM-FEX, DPDK
- Layer 2 switching
- Flow classification (Netflow)
- NPU For Packet Encap/Decap
- RSS, NetQueue and VMQ, **VMMQ,VIC QinQ Tunneling, Physical NIC mode**

2018

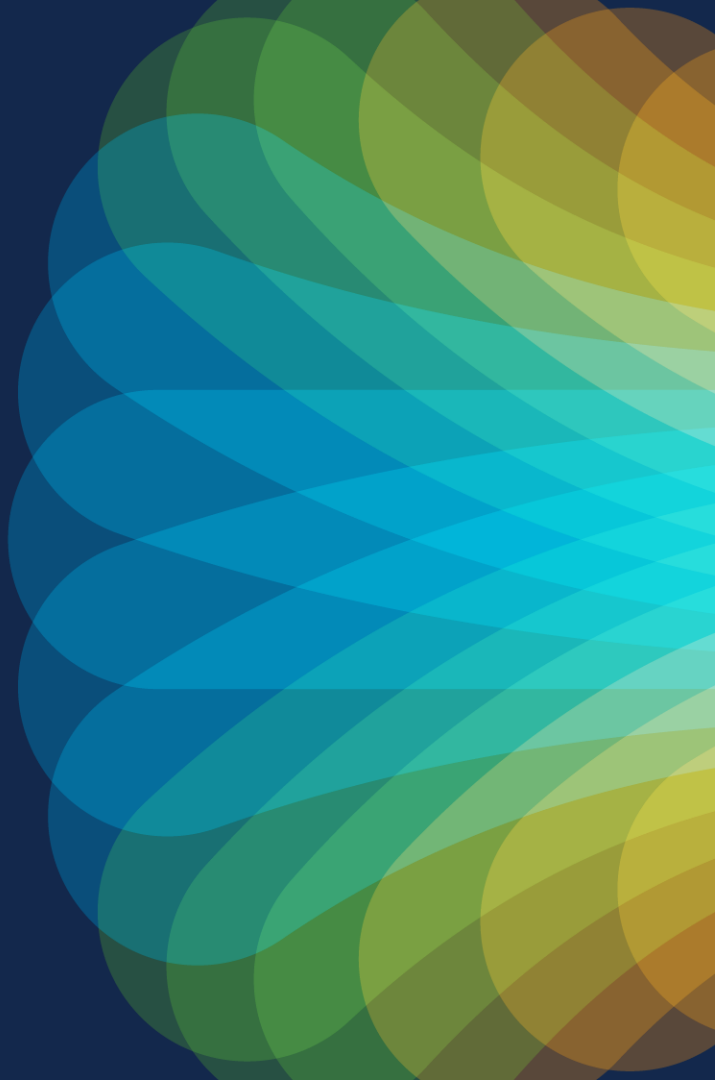
5th Gen

- 10G/25G/40G/**50G/100G/200G**
- **x16, PCIe Gen 4**
- CNA
- 512 virtual PCIe Devices
- Single Wire Mgmt
- NVMeoF: FC-NVMe, RoCEv2 (**Performance**)
- Overlays: NVGRE, VXLAN, Geneve (**Performance**)
- **SR-IOV,** usNIC, DPDK
- Layer 2 switching
- **L3 ECN**
- **16K Tx/Rx Ring Size**
- Flow classification (Netflow)
- **PTPv1/v2**
- NPU For Packet Encap/Decap
- RSS, NetQueue, VMQ, VMMQ, **RSSv2, VIC QinQ Tunneling, Physical NIC mode, windows poll mode**

2022

cisco Live!

Topologies



VIC 15000 for B-, X- Series

15230

15420

15422

15411

VIC 15230

2x 100 Gbps mLOM

VIC 15230 with VIC Secure Boot is the latest (and last) fifth generation VIC for X-Series. VIC 15231 is the non-secure boot version.

VIC 15420

4x 25 Gbps mLOM

The 4x25G mLOM has VIC secure-boot and the option to add the 15422 mezzanine adapter

VIC 15422

4x 25 Gbps Mezzanine

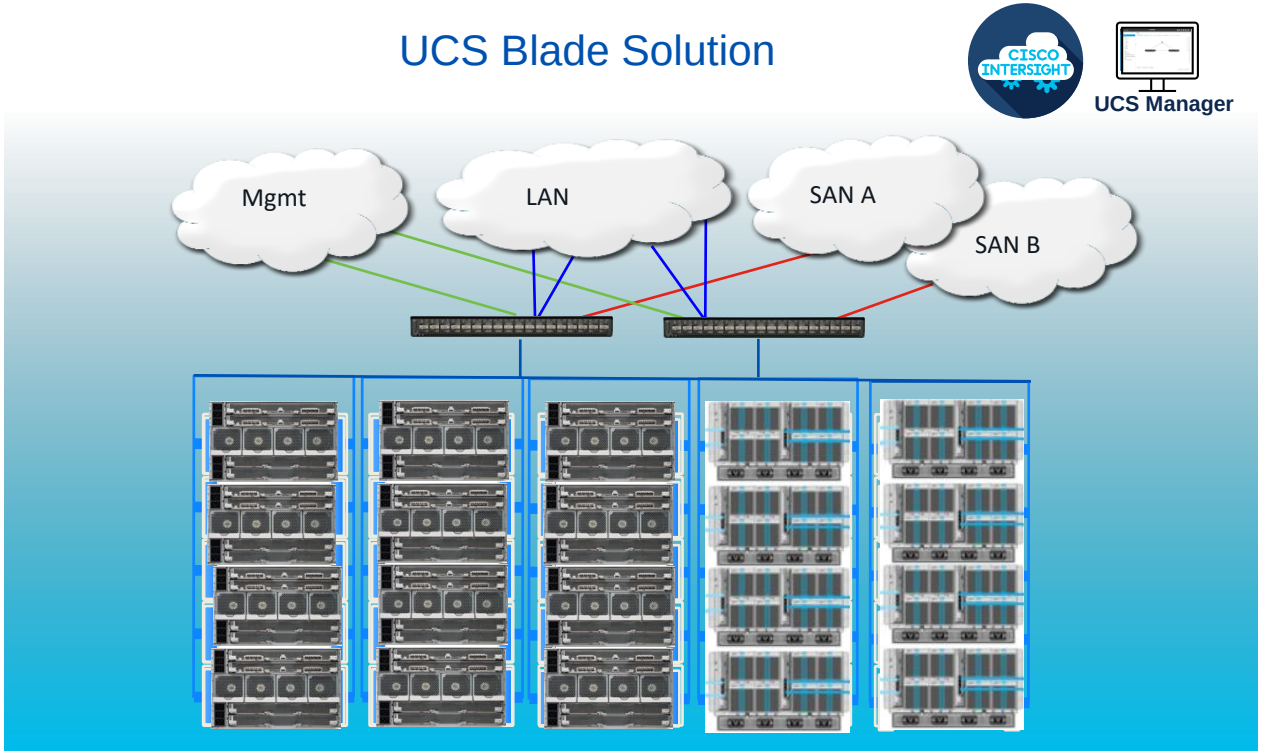
The 15420's neighbor, with 4x25G in Mezzanine form factor, connected with the 15000-bridge module, provides X-Fabric connection as well

VIC 15411

4x 10/ 2x 40 Gbps mLOM

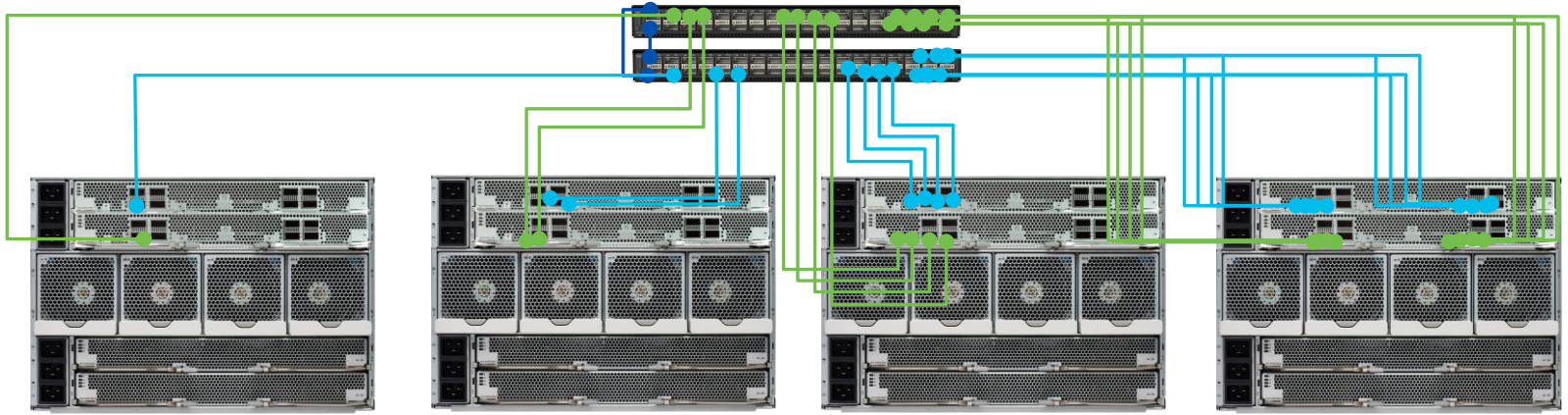
The non-secure boot version mLOM for B-series. 40G enabled via port-expander.

Topology1: Unified Fabric with Blade Chassis



UCS Fabric Topologies : Chassis Bandwidth Options for X-series

- Wire once for Connectivity and Discovery
- Add cables for increased bandwidth



2x 1 LINK

50/200 Gbps per Chassis

2x 2 LINK

100/400 Gbps per Chassis

2x 4 LINK

200/800 Gbps per Chassis

2x 8 LINK

400/1600 Gbps per Chassis

IFM-25G/IFM-100G

Throughput per UCS x210c/x410c compute node

x210c/x410c Compute Node	A FI-6536 + X9108-IFM-100G	B FI-6536/6400-Series + X9108-IFM-25G	C FI-6536 + X9108-IFM-25G/100G Or FI-6400-Series + X9108-IFM-25G	D FI-6536 + X9108-IFM-25G/100G Or FI-6400-Series + X9108-IFM-25G	
x210c/x410c configuration	VIC 15230 or 15231	VIC 15230 or 15231	VIC 15420	VIC 15420 + VIC 15422	
Throughput per node	200G (100G per IFM)	100G (50G per IFM)	100G (50G per IFM)	200G (100G per IFM)	
vNICs needed for max BW	2	2	2	4	
KR connectivity per IFM	1x 100GKR	2x 25GKR	2x 25GKR	4x 25GKR	
Single vNIC throughput on VIC	100G	50G (2x25G KR)	50G (2x25G KR)	50G (2x25G KR)	50G (2x25G KR)
Max Single flow BW per vNIC	100G	25G	25G	25G	25G
Single vHBA throughput on VIC	100G	50G	50G	50G	50G

B200 M6 supported combinations

Fabric Interconnect + I/O Module	FI-6536/ 6400-series + IOM-2408	FI-6536/ 6300 + IOM-2304	FI-6300 / 6400 + IOM-2208	FI-6300 / 6400 + IOM-2204
Server	B200M6	B200M6	B200M6	B200M6
15411	40G	40G	40G	20G
15411 + PE	80G	80G	N/A	N/A

* Recommended combination of FI/IOM/VIC

- No support for FI 6248 or 6296 with B200-M6, hence with 15411
- VIC 15411 support in IMM and UCSM is available from 4.2(3) release
- FI-6536 is supported in IMM and UCSM from 4.2(3) release.

VIC 15000 for C- Series

15427

15425

15237

15235

VIC 15427

4x 10/25/50 Gbps mLOM

The 15427 adapter offers four 10/25/50G ports in the mLOM slot on C-Series M6 and M7 servers. Replaces 15428

VIC 15425

4x 10/25/50 Gbps PCIe

The 15425 adapter provides four 25/50G ports in a PCIe slot for segregated traffic and expanded connectivity

VIC 15237

2x 40/100/200 Gbps mLOM

The 15237 adapter offers up to 200G throughput in the dedicated mLOM slot on C-Series M6 and M7 servers. Replaces 15238

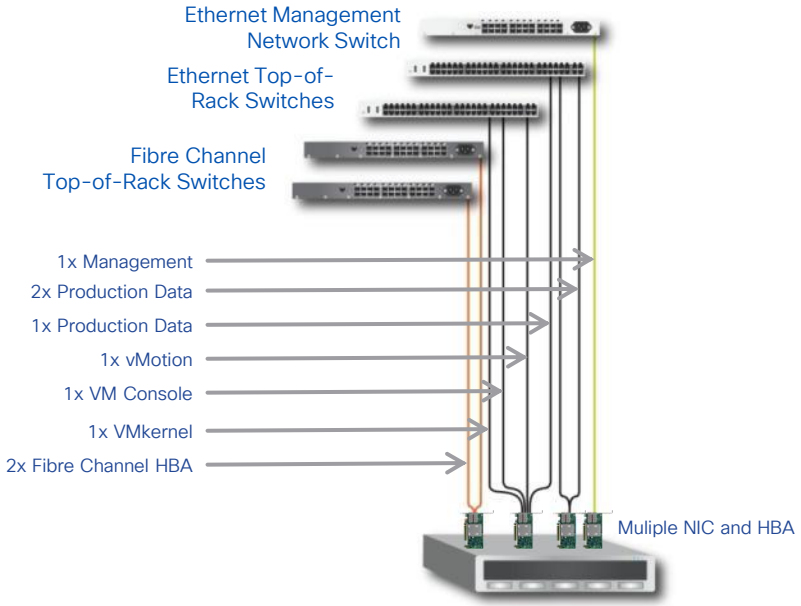
VIC 15235

2x40/100/200 Gbps PCIe

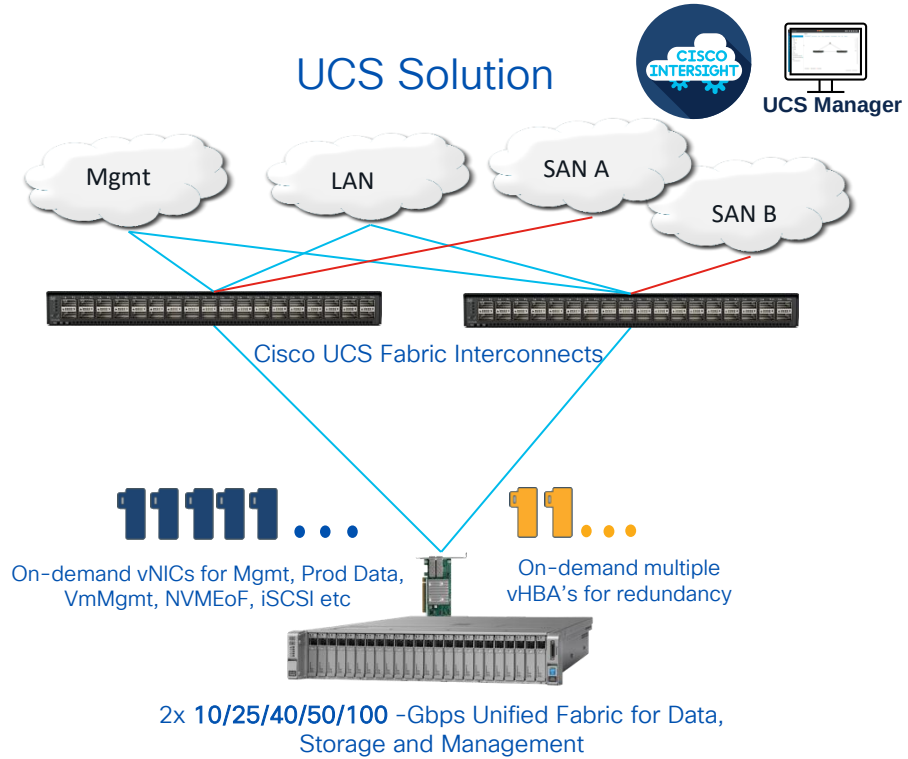
The 15235 adapter provides 200G connectivity in PCIe slots for expanded throughput and physical separation.

Topology-2: Unified Fabric with Rack Servers

Traditional Rack-server

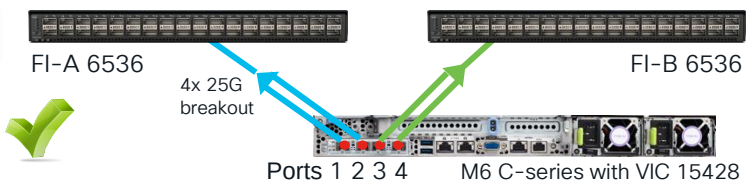


UCS Solution



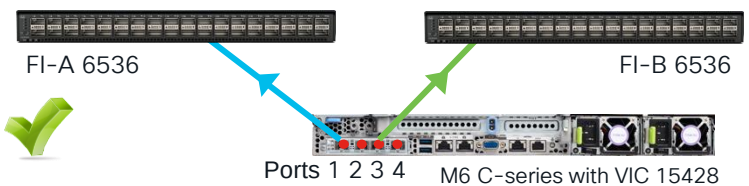
Topology-2: VIC 15427, 15428,15425 connectivity to FI 6536

A



VIC ports 1 & 2 to FI-A and ports 3 & 4 to FI-B - **Supported**

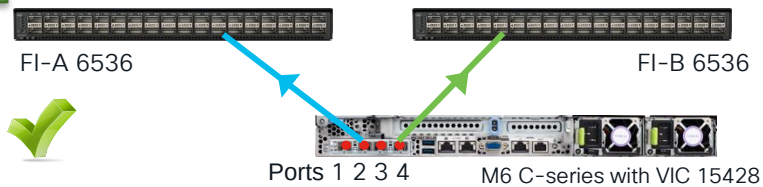
B



VIC port 1 to FI-A and port 3 or 4 to FI-B - **Supported**

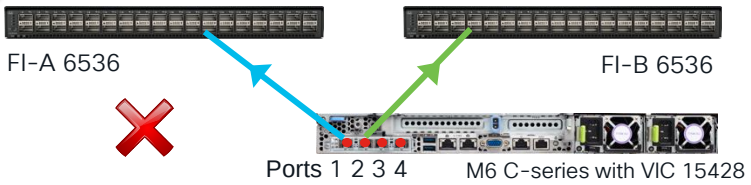
Connectivity with Fabric Interconnect support only Switch Independent modes across Linux, ESXi & Windows OS.

C



VIC ports 2 to FI-A and ports 4 to FI-B - **Supported**
VIC ports 1 or 2 to FI-A and ports 3 or 4 to FI-B - **Supported**

D



Not Supported
1. VIC port 1 to FI-A and port 2 to FI-B
2. VIC port 3 to FI-A and port 4 to FI-B

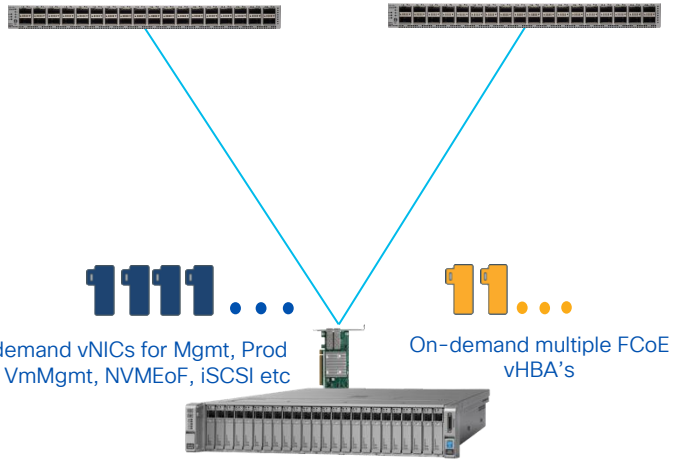
https://www.cisco.com/c/en/us/support/docs/servers-unified-computing/ucs-b-series-blade-servers/200519-UCS-B-series-Teaming-Bonding-Options-wi.html

UCS VIC Additional Topologies: Rack Server Environment

Topology-3

Cisco IMC

Nexus N9K or ToR Switch



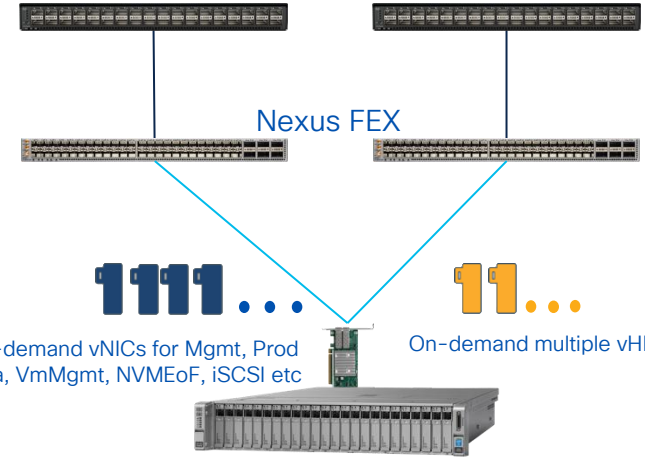
On-demand vNICs for Mgmt, Prod Data, VmMgmt, NVMEoF, iSCSI etc

On-demand multiple FCoE vHBA's

2x **10/25/40/50/100/200** -Gbps Unified Fabric for Data and Management

Topology-4: FEX for rack-server scalability

UCS Fabric Interconnect or Nexus N9K



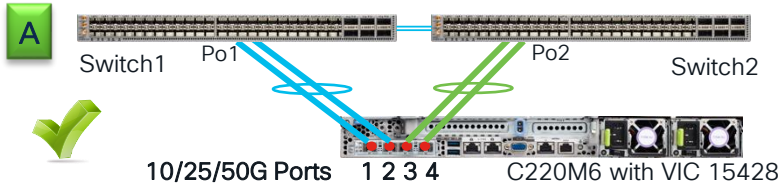
Nexus FEX

On-demand vNICs for Mgmt, Prod Data, VmMgmt, NVMEoF, iSCSI etc

On-demand multiple vHBA's

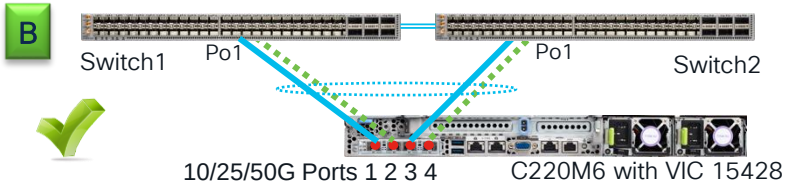
2x **10/25** -Gbps Unified Fabric for Data and Management

Topology-3: VIC 15427, 15428, 15425 connectivity to Nexus switch in standalone mode



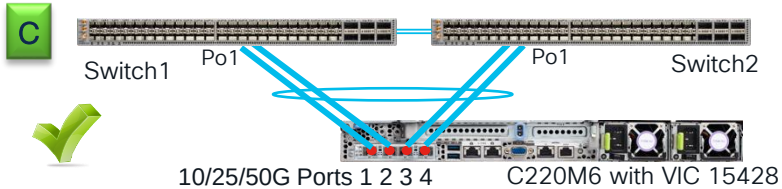
VIC ports 1 & 2 to SW1 and ports 3 & 4 to SW2 - Supported

- With Default VIC port-channel (PO) **enabled**
- Requires PO config on switch with switch dependent bonding
- Cannot support MCT/VPC at ToR switch and OS IP-hash load-balancing
- MAC-hash or port-ID load-balancing in OS should be used to avoid mac-move on ToR



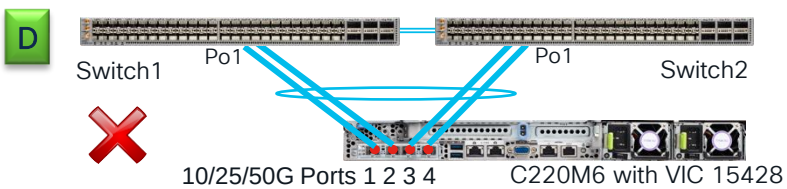
VIC ports 1 or 2 to SW1 and ports 3 or 4 to SW2 - Supported

- With VIC port-channel **enabled** and using one link in (1,2) & (3,4) port pair
- Supports switch dependent & switch-independent OS teaming/bonding
- Supports MCT/VPC at ToR switch and all OS teaming load-balancing options



VIC ports 1, 2, 3, 4 to SW1 & SW2 - Supported

- With VIC port-channel **disabled**
- Supports MCT/VPC at ToR switch and all OS teaming/load-balancing options



VIC ports (1,2) to SW1 & (3,4) to SW2 - Not Supported

- With VIC port-channel **enabled**

These connectivity options are applicable at 10G/25G

Debugging connectivity issues due to FEC at 25G

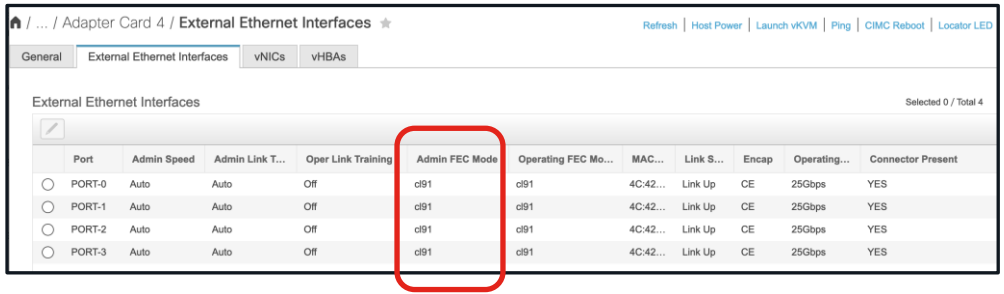
1

- Match FEC on Switch & VIC
- Some older N9K have default at CL74 and defaults are different across switches.
- FEC on VIC at 25G is CL91 by default
- Auto-FEC is only for copper cables
- Auto-FEC is disabled in VIC
- Ex: on how to set CL91 on N9K

```
N9K-2(config-if)# exit
N9K-2(config)#
N9K-2(config)# int eth 1/1
N9K-2(config-if)# fec cl91
N9K-2(config-if)#
```

2

- Check FEC config in CIMC
- FEC should match on both ends of the link

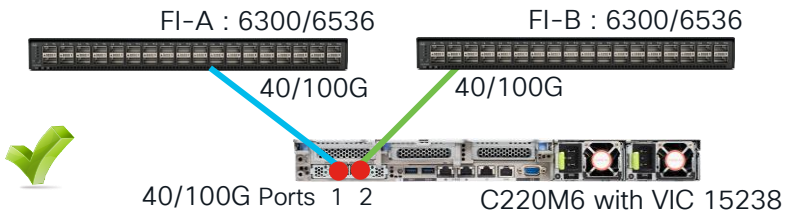


3

- Cables/transceivers also have minimum FEC requirement
- Ex: SFP-25G-SR-S/ CSR-S / LR-S have a minimum FEC of CL91

VIC 15237, 15238, 15235 connectivity to Fabric Interconnects or Nexus switch

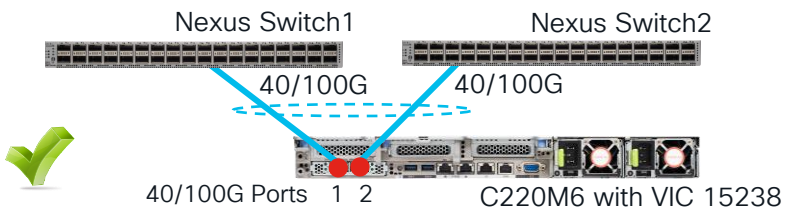
UCSM / IMM Managed



VIC ports 1 to FI-A and ports 2 to FI-B - Supported

- Connectivity to fabric-interconnect support only switch independent teaming
- Cannot support MCT/VPC at FI and OS IP-hash kind of load-balancing
- MAC-hash or port-ID load-balancing in OS should be used to avoid mac-move on FI
- There is no need to configure FEC

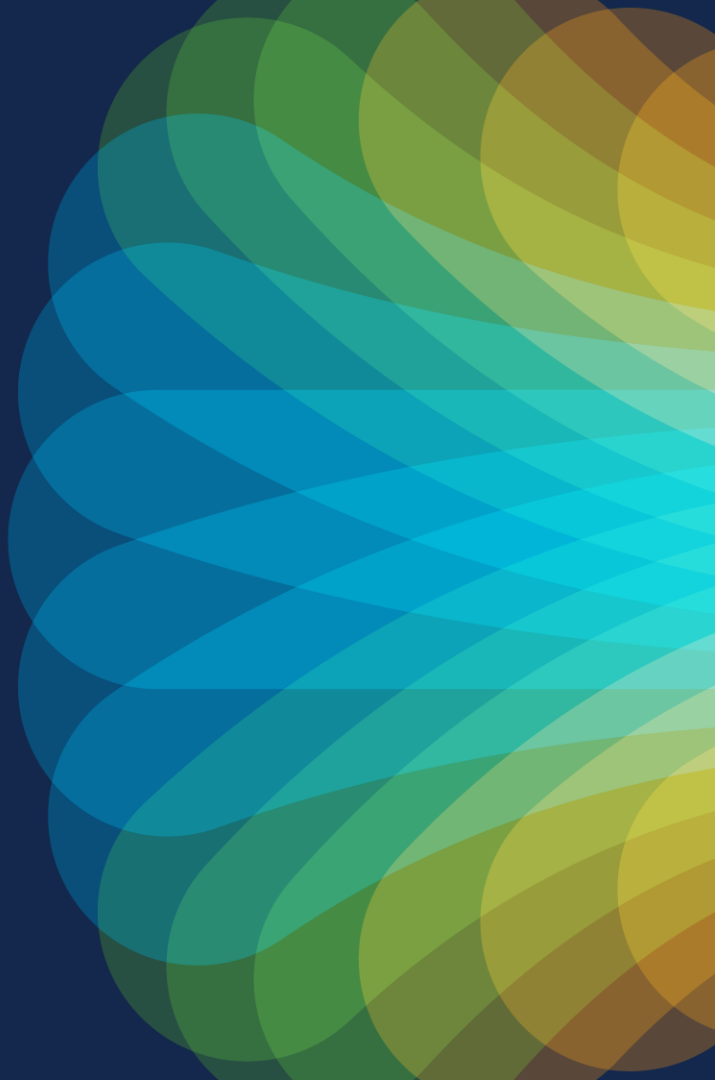
IMC or Standalone mode



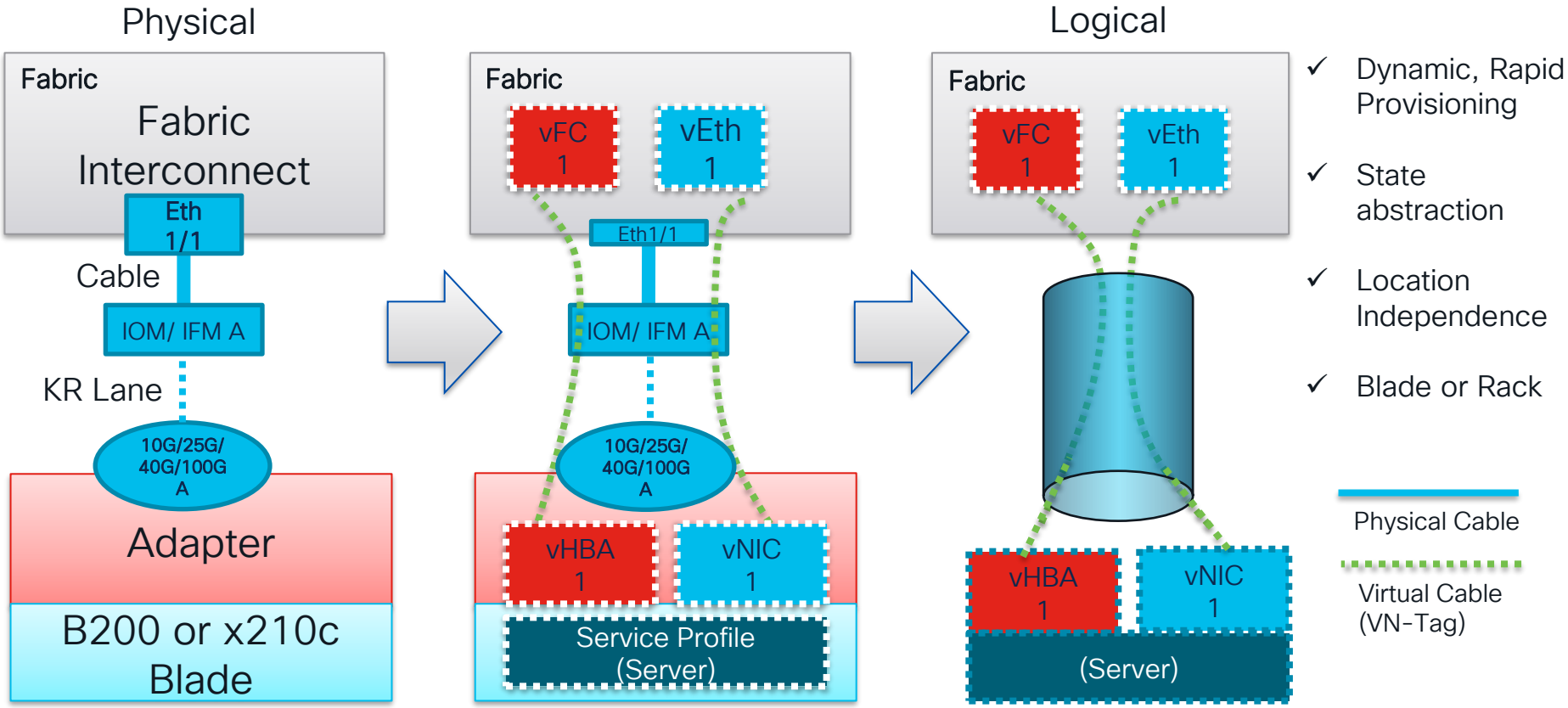
VIC ports 1 to SW1 and ports 2 to SW2 - Supported

- Supports switch dependent & switch-independent OS teaming/bonding
- Supports MCT/VPC at ToR switch and all OS teaming load-balancing options
- Default FEC of CL-91 on VIC works for all cable types, so ensure switch end is also CL-91

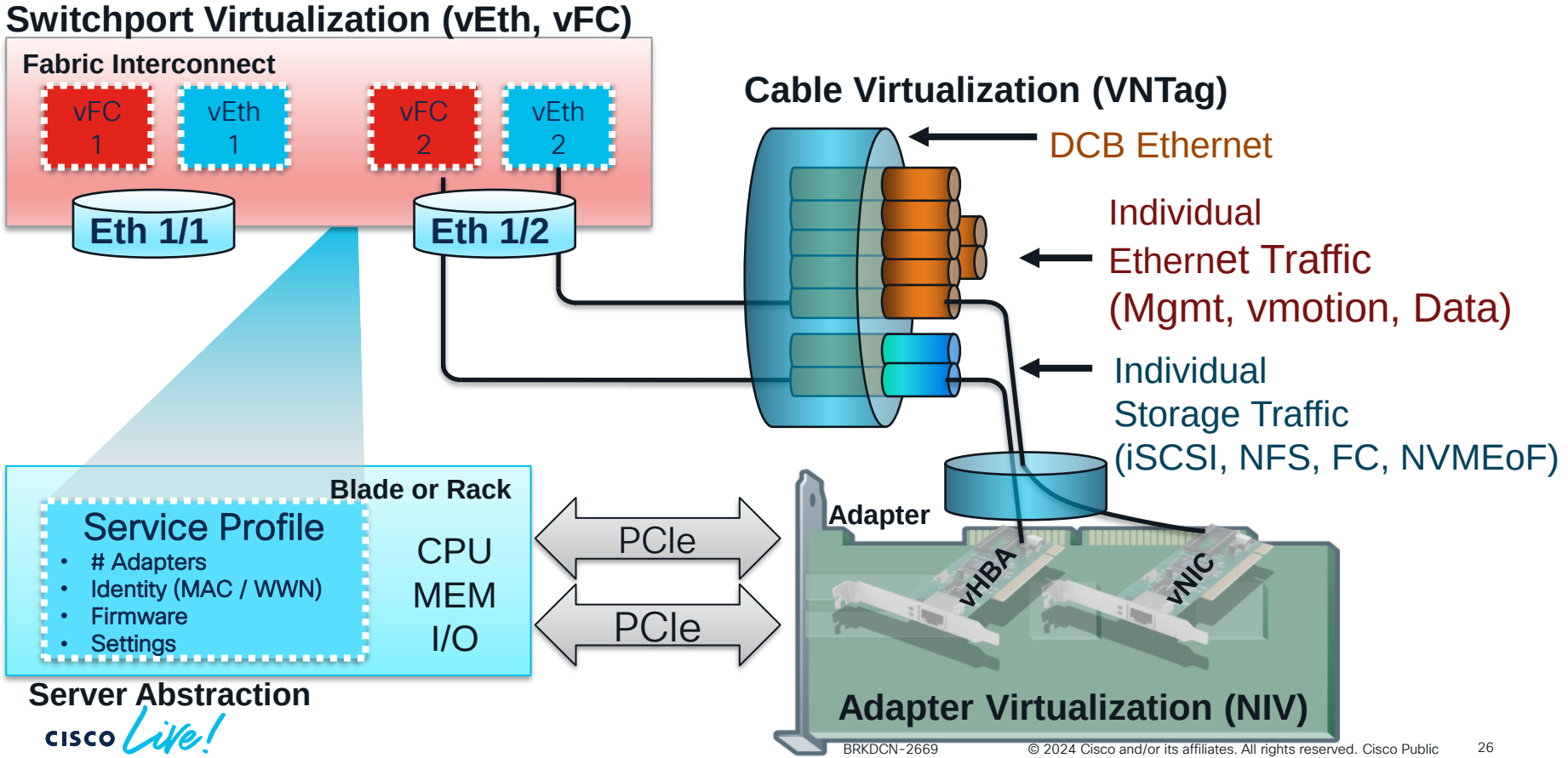
Traffic Flow



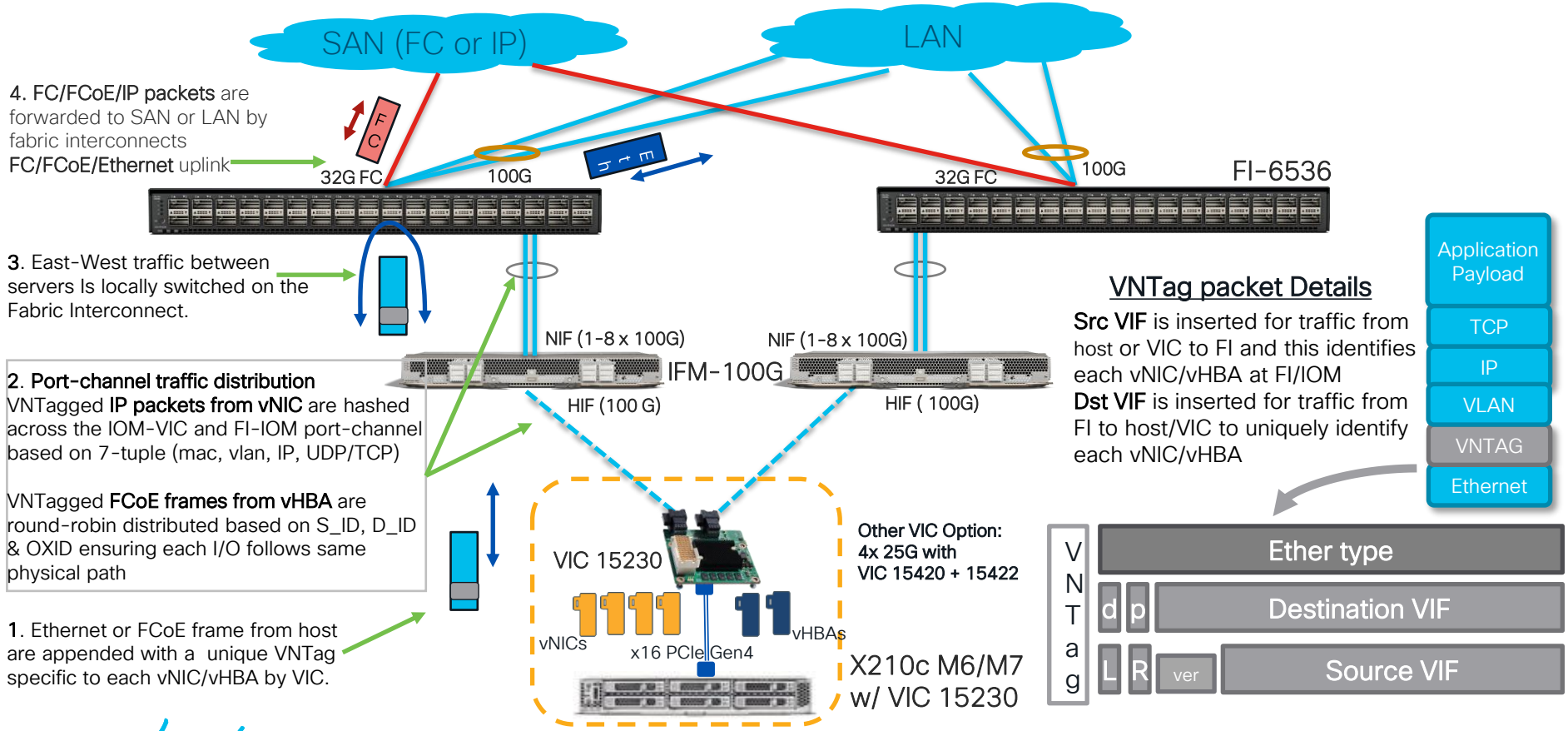
Abstracting the Logical Architecture



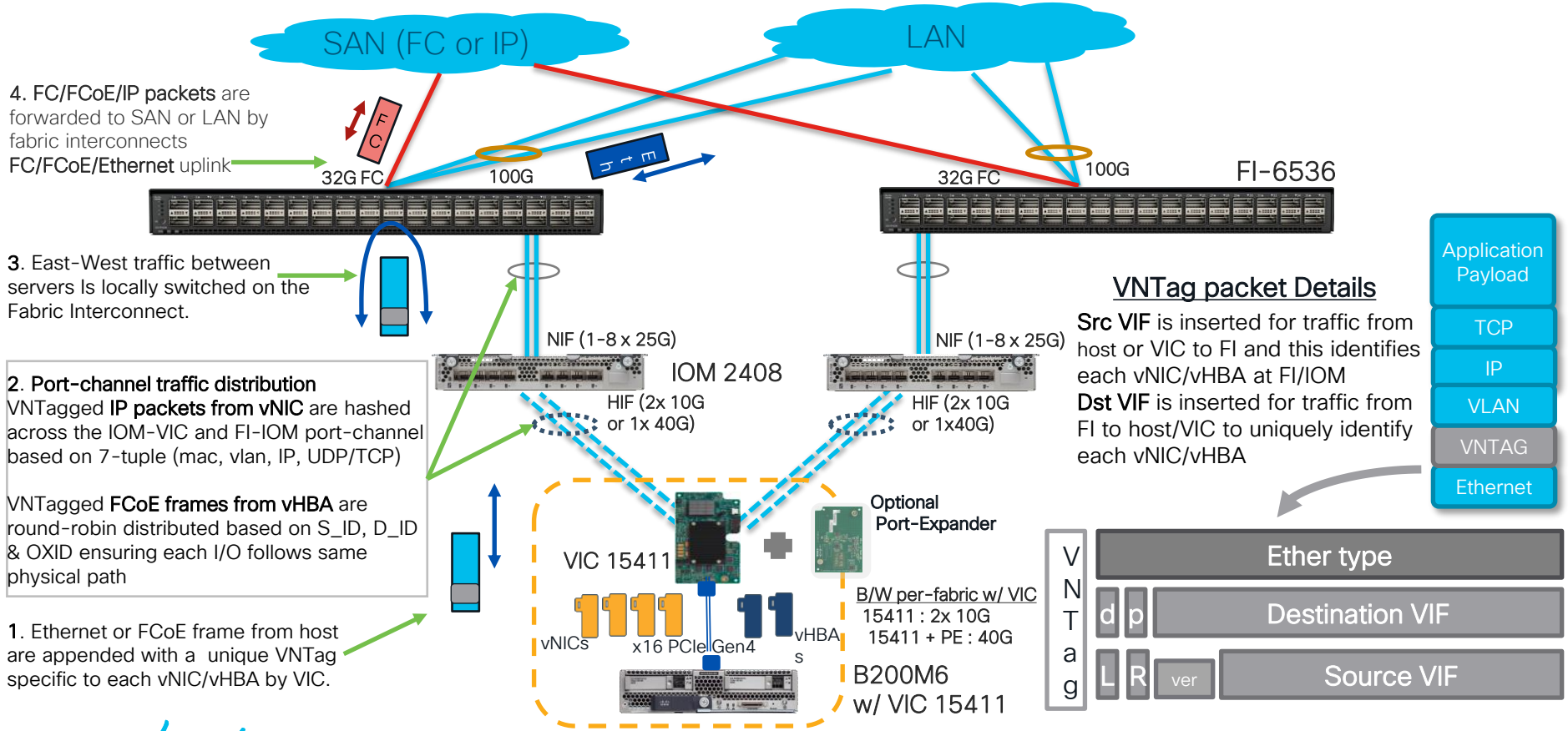
Cisco UCS: Infrastructure Virtualization



4th/5th Gen Fabric – FC/IP Packet Flow with X-Series



4th/5th Gen Fabric – FC/IP Packet Flow with B200



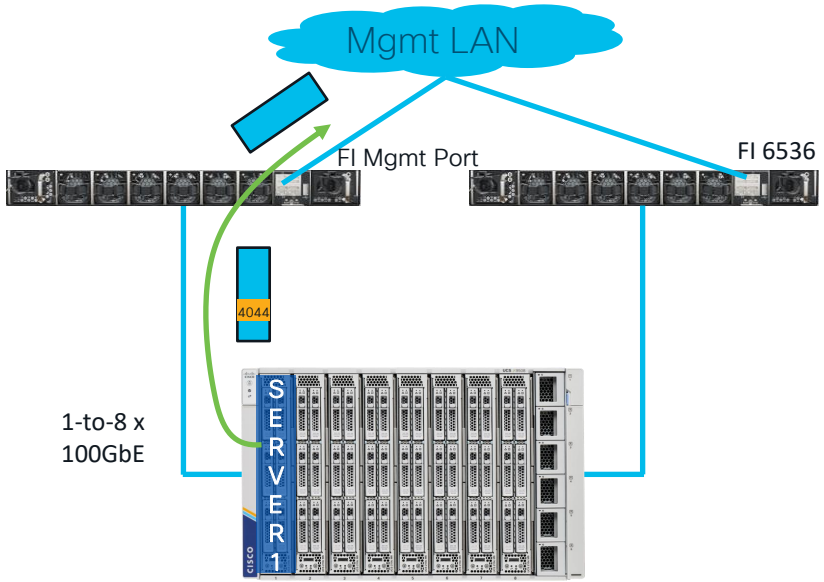
VNTag packet Details

Src VIF is inserted for traffic from host or VIC to FI and this identifies each vNIC/vHBA at FI/IOM

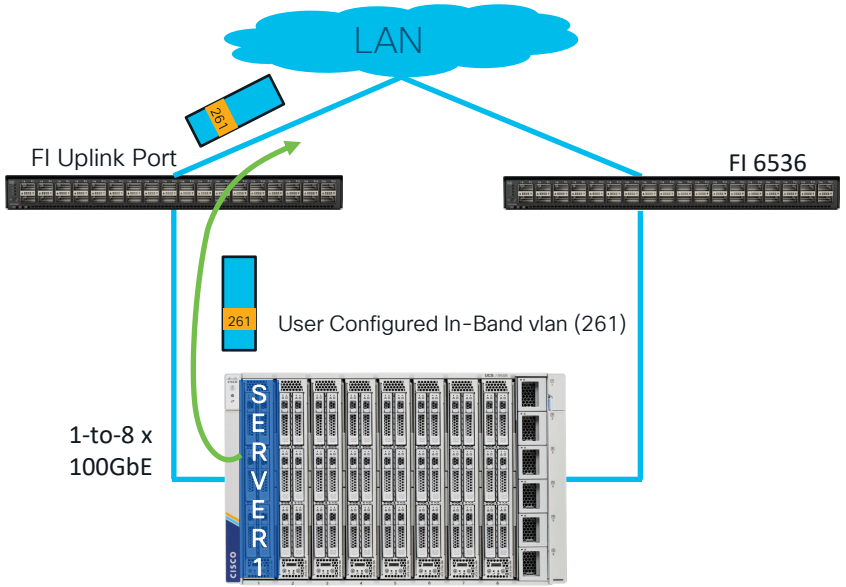
Dst VIF is inserted for traffic from FI to host/VIC to uniquely identify each vNIC/vHBA

X-/B-/C- series Server Management Options

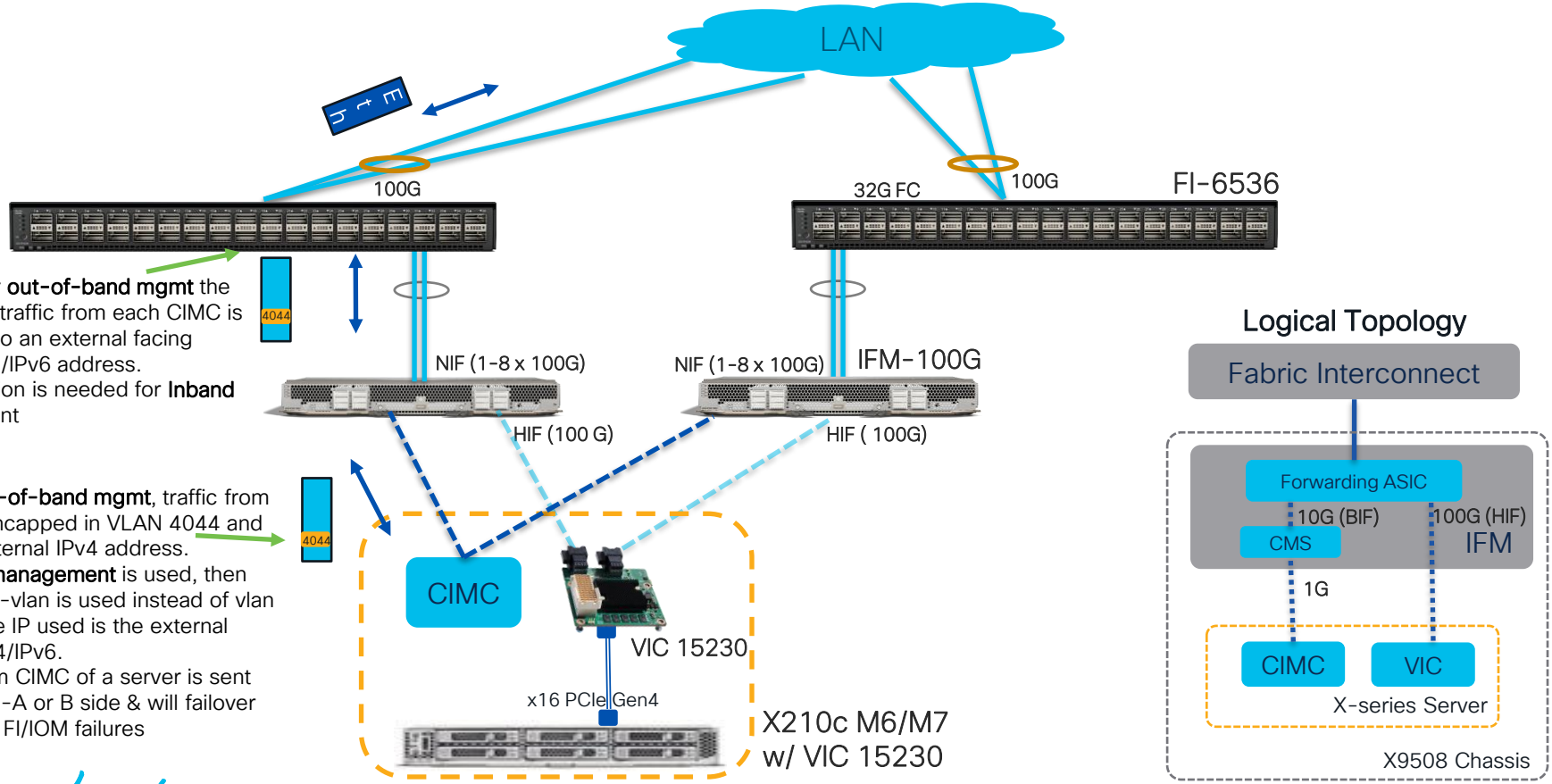
Out-of-Band Management



In-Band Management



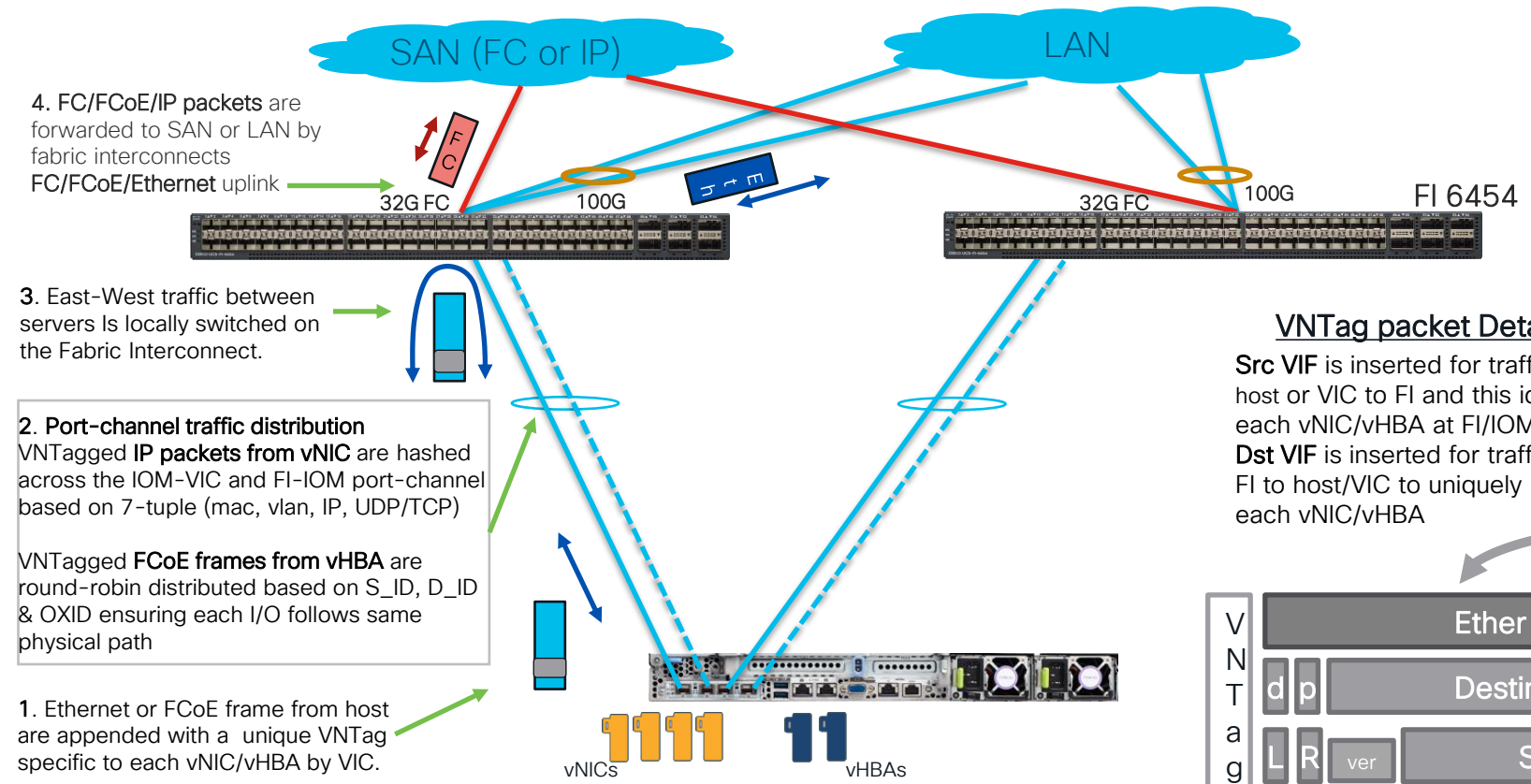
4th/5th Gen Fabric – Management traffic flow in X-Series



2. At FI, for **out-of-band mgmt** the vlan-4044 traffic from each CIMC is translated to an external facing Mgmt-IPv4/IPv6 address.
No translation is needed for **Inband** management

1. For **Out-of-band mgmt**, traffic from CIMC is encapped in VLAN 4044 and uses an internal IPv4 address.
If **Inband management** is used, then the inband-vlan is used instead of vlan 4044 & the IP used is the external Mgmt-IPv4/IPv6.
Traffic from CIMC of a server is sent to either FI-A or B side & will failover on various FI/IOM failures

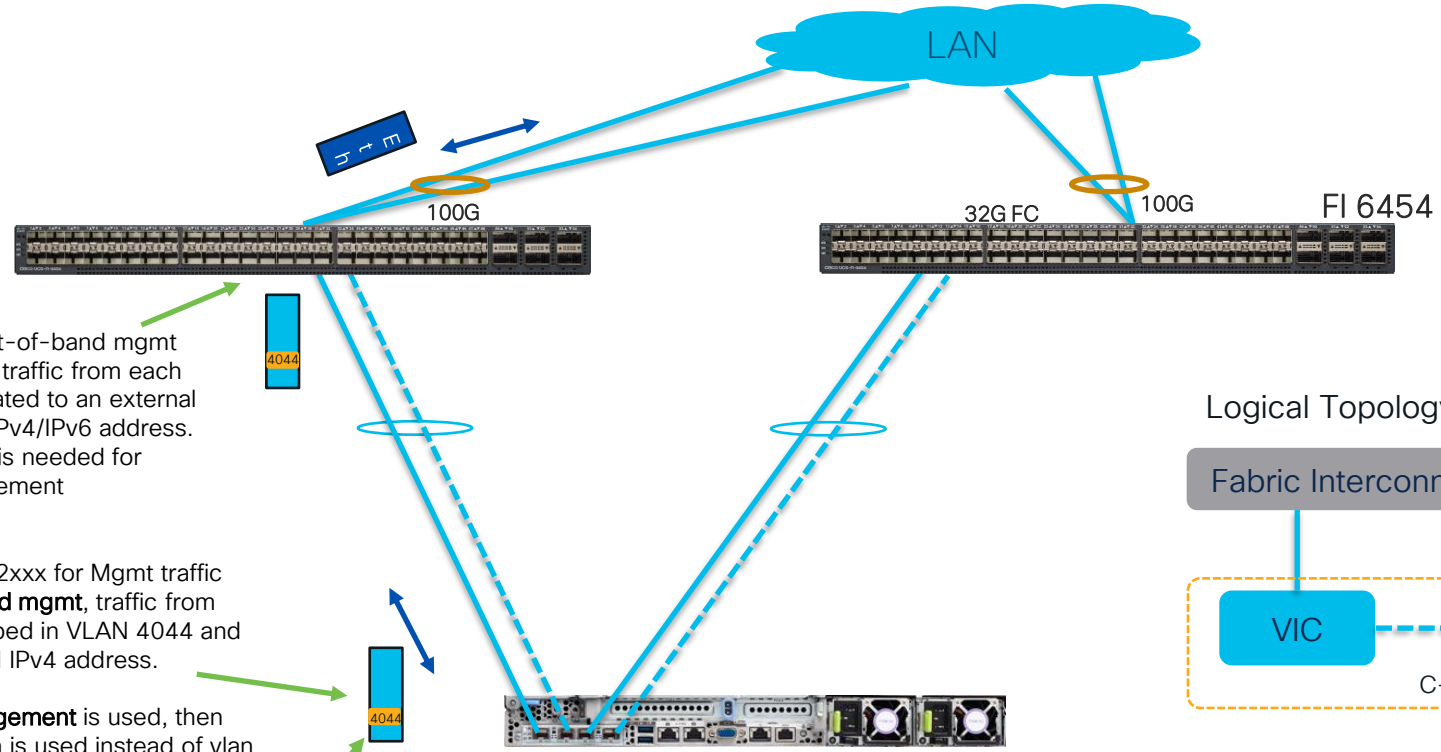
4th/5th Gen Fabric – FC/IP Packet Flow with C-Series



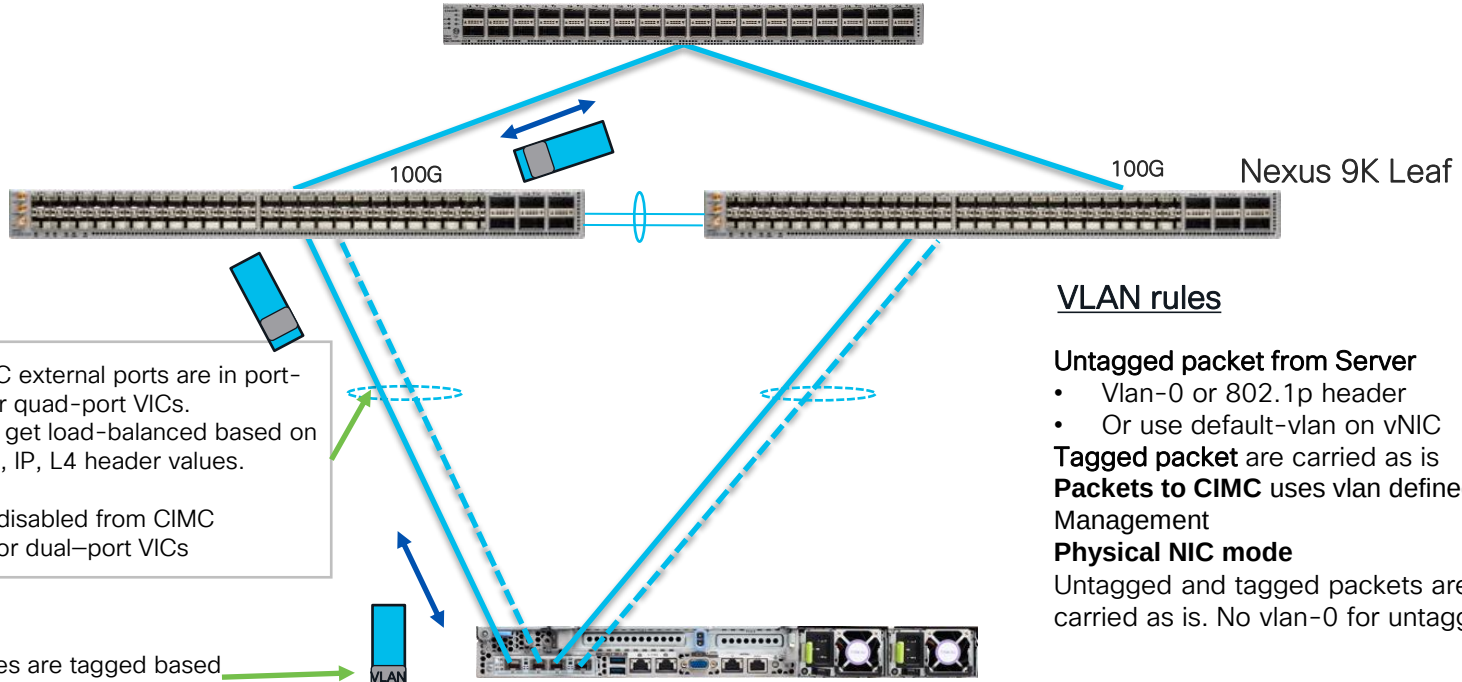
VNTag packet Details

Src VIF is inserted for traffic from host or VIC to FI and this identifies each vNIC/vHBA at FI/IOM
Dst VIF is inserted for traffic from FI to host/VIC to uniquely identify each vNIC/vHBA

4th/5th Gen Fabric – Management traffic flow with rack servers in UCSM/IMM



Packet Flow with C-Series in Standalone mode



2. By default VIC external ports are in port-channel (PO) for quad-port VICs.
Packet over PO get load-balanced based on the MAC, VLAN, IP, L4 header values.

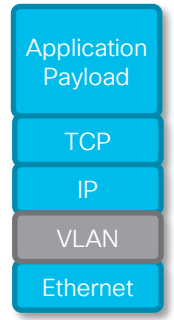
VIC PO can be disabled from CIMC
No default PO for dual–port VICs

1. Ethernet frames are tagged based on the VLAN rules .

VLAN rules

Untagged packet from Server
- Vlan-0 or 802.1p header
- Or use default-vlan on vNIC

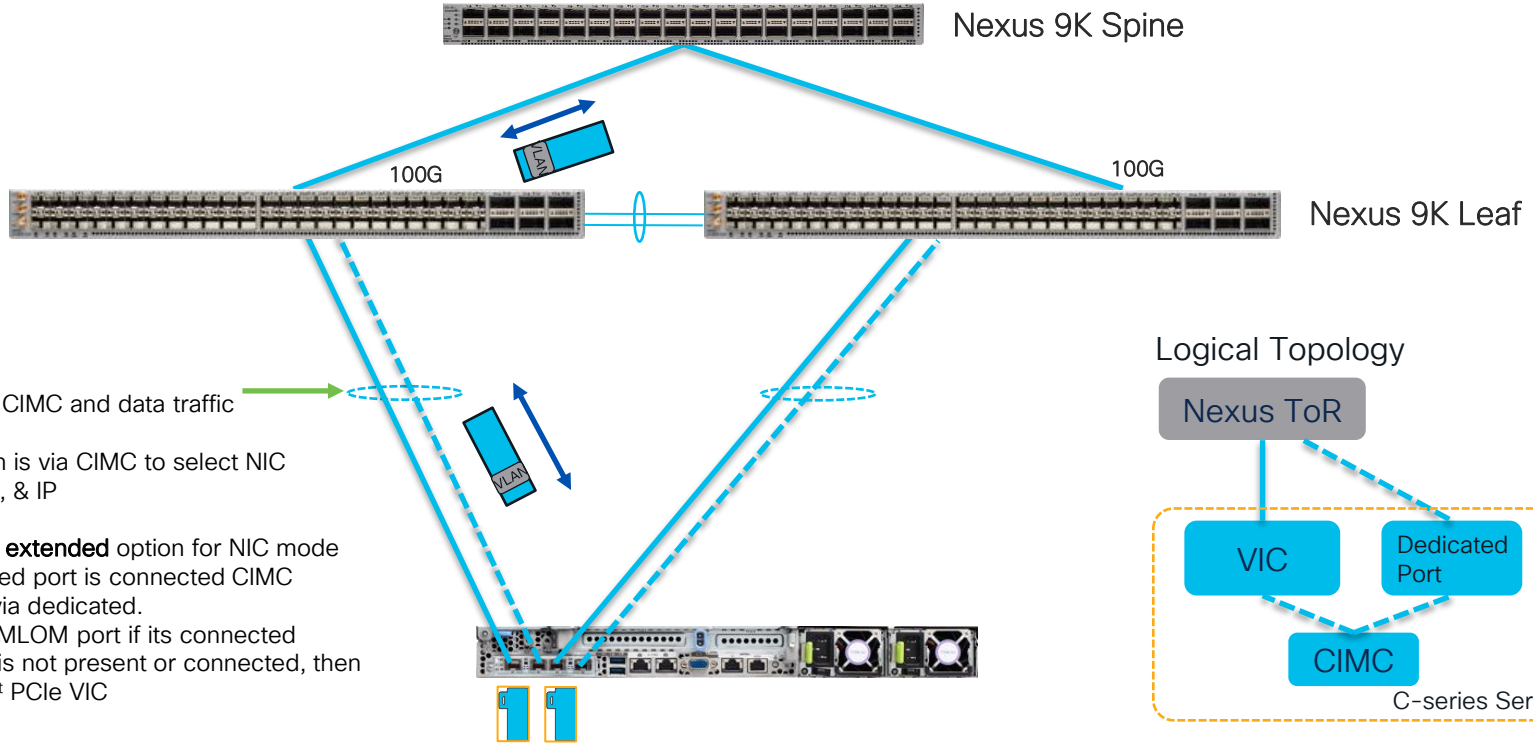
Tagged packet are carried as is
Packets to CIMC uses vlan defined for Management
Physical NIC mode
Untagged and tagged packets are carried as is. No vlan-0 for untagged



VNIC Rules

1. By default, with VIC PO there will be 2 vNICs on a Quad-port VIC (**15427, 15425**)
2. On disabling VIC port-channel, there will be 4 vNICs by default on Quad-port VIC
3. Dual-port VIC's (**15237, 15235**) will have 2 vNIC by default and there is no VIC PO
4. User can define more on-demand vNICs from CIMC or Intersight.

Management Traffic Flow with C-Series in Standalone mode



Carries both CIMC and data traffic

Configuration is via CIMC to select NIC mode, VLAN, & IP

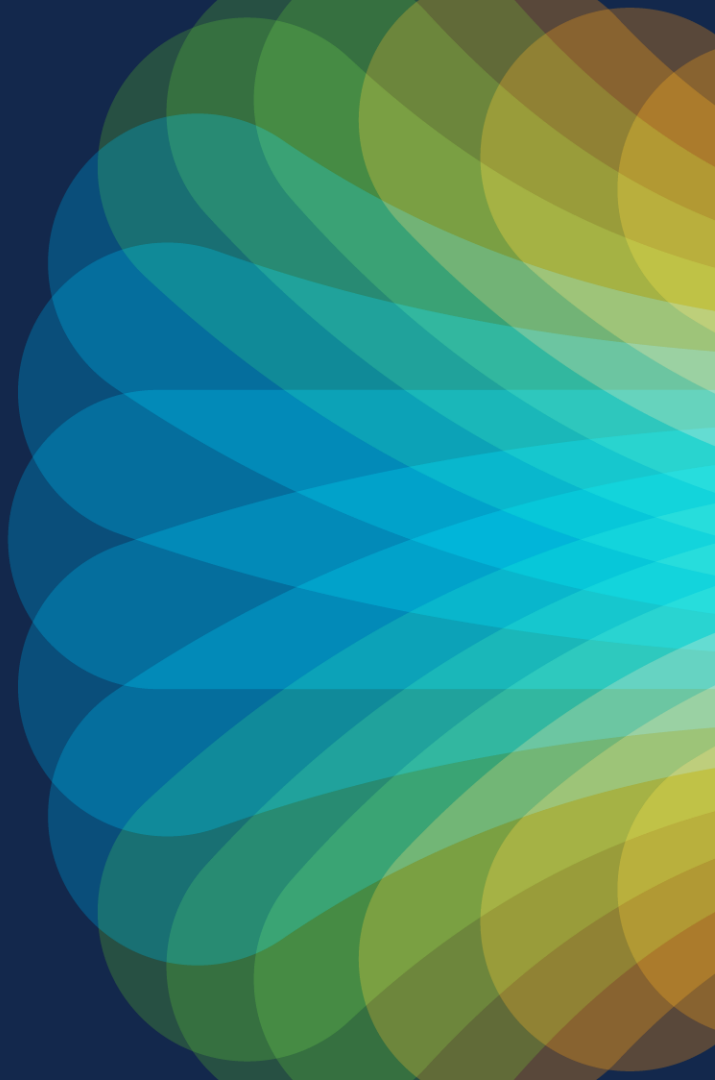
Shared LOM extended option for NIC mode

- If dedicated port is connected CIMC traffic is via dedicated.
- If not via MLOM port if its connected
- If MLOM is not present or connected, then via the 1st PCIe VIC

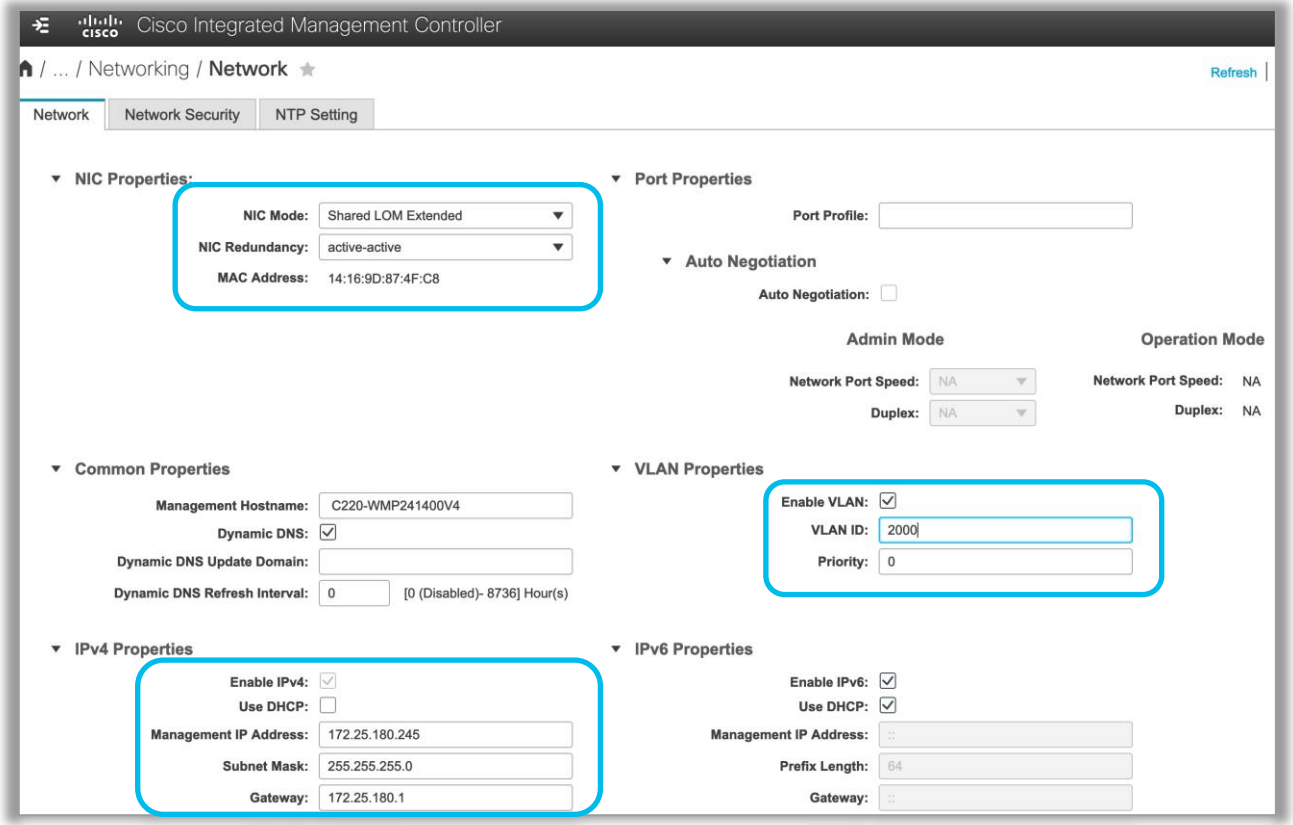
Virtual NCSI channel on the VIC is used for CIMC communication

Logical Topology

VIC Features



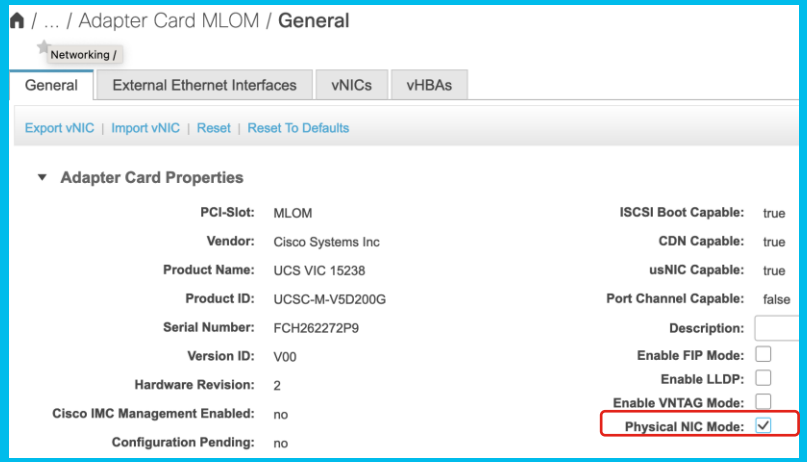
Single Wire Management in Standalone mode

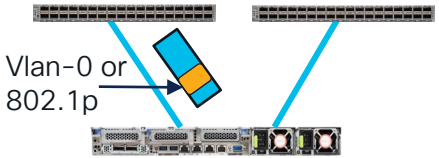


MLOM notes

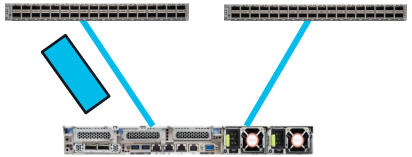
- M7 C-series servers don't have separate LOM ports
- 15427, 15425 supports 10G-T-X support for 10G copper cables along with various 10G/25G connectivity

Physical NIC mode (in Standalone Server)





Default VIC behavior for untagged packet



With Physical NIC Mode enabled

- Supported on VIC 1400/15000 series with standalone rack servers.
- This feature is only available with VIC in standalone server.
- Physical NIC mode disables default priority tagging on VIC vNICs
  - Allows interoperability with switches that don't support priority-tagging
- Only default vNICs are supported, no additional vNICs can be created
  - 2 vNICs for dual-port MLOM or PCIe rack VIC
  - 4 vNICs for quad-port MLOM or PCIe rack VIC
- Disable FIP and LLDP on VIC to enable physical NIC mode

VIC Secure Boot

VIC secure-boot uses Silicon Root of Trust to verify that the VIC hardware and the firmware are immutable. Thus, ensuring VIC integrity with every server boot.

This happens transparently to customers on every boot of the system.

If VIC secure boot fails, IMC/UCSM/Intersight will show an alert and the server will not boot.

All 5th Gen VIC form factors (except B-Series 15411) offer VIC secure boot.

Note that the VIC secure-boot is not the same as server secure boot, but they work together in the Cisco Trust model. Both these technologies together ensures that customers server & VIC are genuine and are running valid unmodified hardware/firmware.

VIC Secure Boot is a bit confusing
(but it doesn’t need to be)

What and where it is

VIC Secure Boot is a boot-time hardware verification before the server even boot up the operating system.

It makes sure the physical VIC hardware and firmware are valid and secure.

What and where it isn’t

VIC Secure Boot has nothing to do with the operating system, user-space, or applications. Once booted, it’s like it isn’t there at all.

Drivers are the same. OS is the same. It’s a drop-in replacement for previous versions where applicable.

No user intervention is required

VIC Secure Boot is completely hands-off for the customer. It does not have to be ordered with any options (just order the Secure Boot VIC models and it’s there). There is no configuration to be done, and it cannot be disabled.

VIC Q-in-Q Tunnelling adds to the UCS scalability story

Double-tagging from VIC

An outer VLAN is added at vNIC for the traffic from hypervisor/server, thus providing massive scalability of VLAN separation across a datacenter environment.

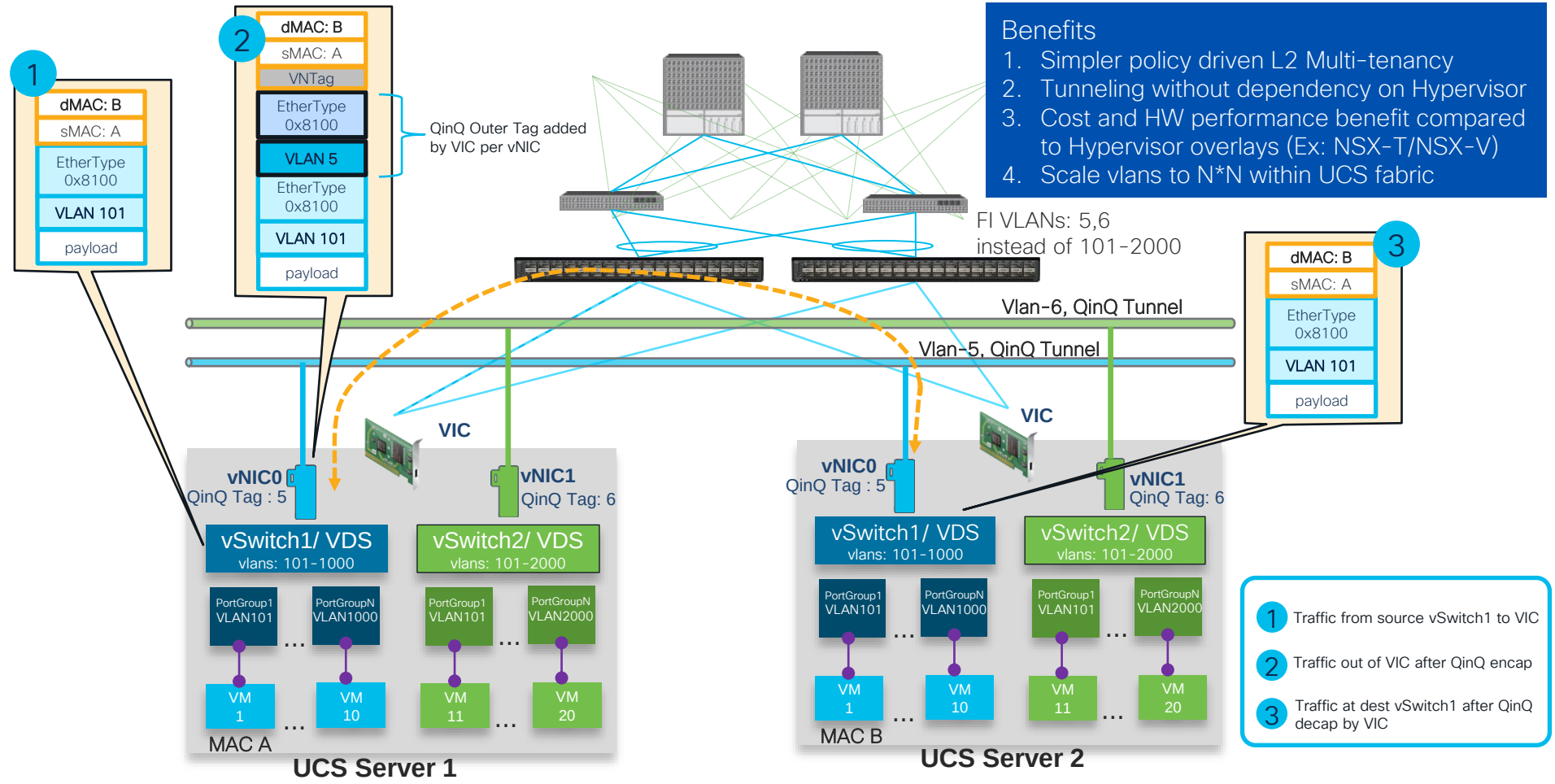
Multi-network, multi-tenancy enabled

A service provider's environment can provide the same VLAN numbers to multiple customers, thus virtualizing the same physical infrastructure.

Traffic segregation

Provide additional security, logical separation and monitoring capability based on departments, organizations, customers, and workloads as needed

UCS VIC QinQ Tunneling in action, L2 segmentation



Benefits

1. Simpler policy driven L2 Multi-tenancy
2. Tunneling without dependency on Hypervisor
3. Cost and HW performance benefit compared to Hypervisor overlays (Ex: NSX-T/NSX-V)
4. Scale vlans to N*N within UCS fabric

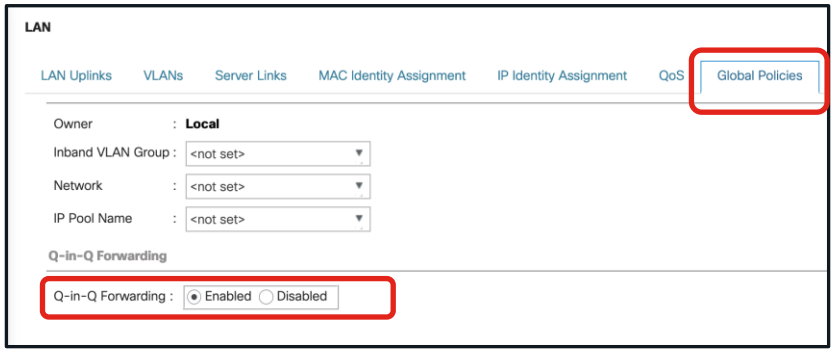
1. Traffic from source vSwitch1 to VIC
2. Traffic out of VIC after QinQ encap
3. Traffic at dest vSwitch1 after QinQ decap by VIC

VIC Q-in-Q Tunneling Details

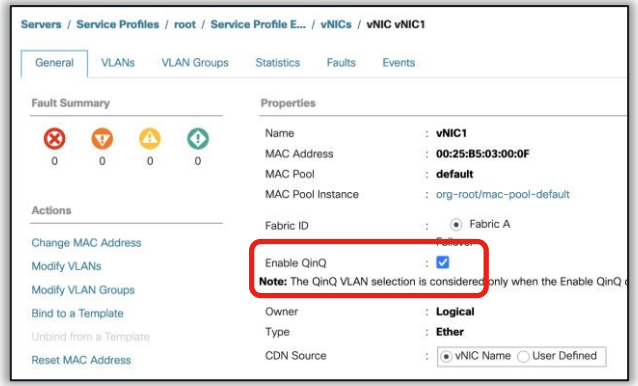
- VIC QinQ tunneling supports the following
  - VIC 1400/15000 series
  - UCS FI 6400 series and FI 6536
  - UCSM, IMM and Cisco IMC support with 4.3(2) release
- Configurations required in UCSM/IMM
  - Enable QinQ globally under FI domain (UCSM only)
  - Enable QinQ under the VLAN in a vNIC
  - Select Native-vlan under a vlan in vNIC for untagged traffic
- Upstream N9K or LAN network should allow QinQ forwarding. N9K switches require “system dot1q tunnel transit” configuration.
- For standalone fabrics ensure the upstream ToR switch will carry double-tagged 802.1Q frames

VIC Q-in-Q Tunneling, UCSM Configuration

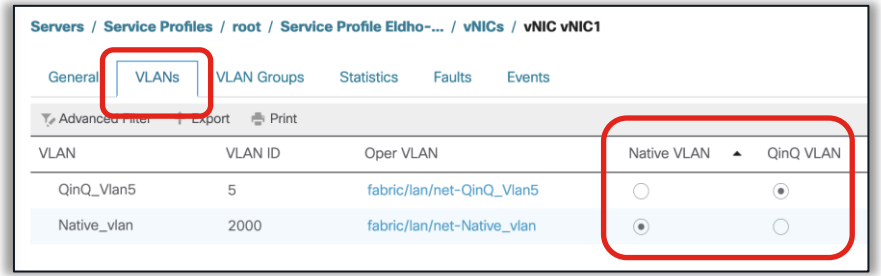
1



2



3



VIC Q-in-Q deployment in field

One UCS public sector customer was able to replace VMware NSX using the VIC QinQ.

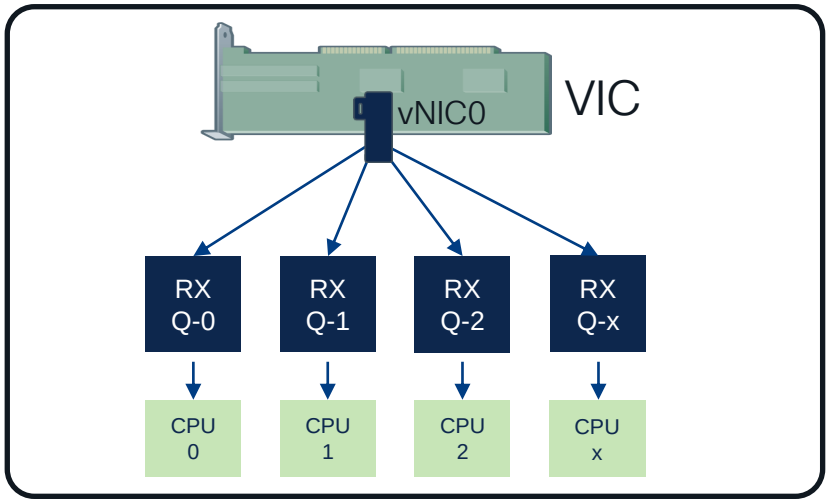
Created a multi-tenant environment to host multiple customers delivering VM resources on demand.

Thus, lowering operational costs with no licensing and fewer software platforms to worry about.

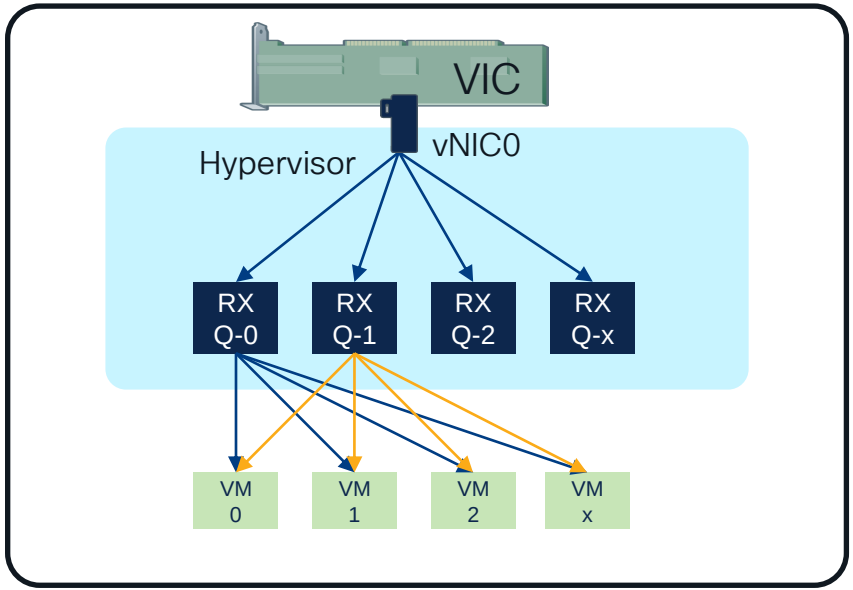
Simplified deployment and maintenance across the environment with UCSM benefits. And higher performance by avoiding the hypervisor overlay tunnels.

Receive Side Scaling (RSS)

Bare-metal Server



Virtualized Server



Receive Side Scaling (RSS)

- VIC HW feature supported for ESXi, Linux, and Windows
- RSS provides better server CPU utilization, higher throughput and handles bursty traffic
- Achieved by Rx traffic distribution across multiple Rx-queues/cpu-cores based on L2/L3/L4 packet header fields
- VIC 15000 series support 16K Tx and Rx ring size, while previous generations supported up to 4K Tx and Rx ring size

Adapter policy for performance with RSS

Parameter	ESXi	Linux	Windows
TX queue	1	1	1
TX ring size	4K /16K	4K /16K	4K /16K
RX queue	8	8	8
RX ring size	4K /16K	4K /16K	4K /16K
CQ	9	9	9
Interrupt	11	11 or 10	512
Interrupt Calculation	CQ + 2	“CQ + 2” or “Rx-Queue +2”	512 or “2x CPU-cores +4”
RSS	Enabled	Enabled	Enabled

vNIC queue-drop debugging

Queue statistics from ESXi shell

a

```
rx_drop: 0
rx_no_bufs: 0
rx_errors: 0
rx_rss: 0
rx_crc_errors: 0
rx_frames_64: 29307
rx_frames_127: 5351
rx_frames_255: 2189
rx_frames_511: 3303
rx_frames_1023: 1136
rx_frames_1518: 454
rx_frames_to_max: 0
tx_queue_[0]_frames_ok: 9840
rx_rss_queue_[0]_frames_ok: 9065
rx_rss_queue_[1]_frames_ok: 5294
rx_rss_queue_[2]_frames_ok: 5324
rx_rss_queue_[3]_frames_ok: 3532
rx_rss_queue_[4]_frames_ok: 3554
rx_rss_queue_[5]_frames_ok: 3593
rx_rss_queue_[6]_frames_ok: 6289
rx_rss_queue_[7]_frames_ok: 5089
/> cat /net/pNics/vmnic0/stats
```

Ensure rx_no_bufs is 0 or doesn't increment

Statistics for 1 Tx Queue & 8 Rx Queues

Execute in "vsish"

Action to take if rx_no_buf increments

1. Increase Rx Queues
2. Increase Rx-Qu ring-size
3. Tune BIOS, OS, application next

b

Queue statistics from Linux host

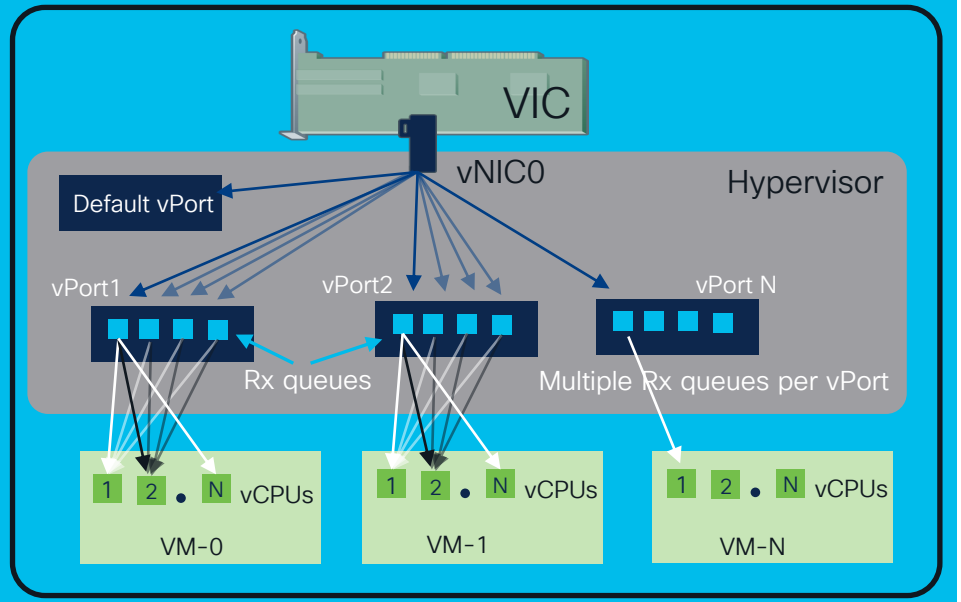
```
[root@4gfi-sum-ch1-s4 ~]# ethtool -i eno8
driver: enic
version: 4.2.0.28-877.24
firmware-version: 5.3(2.4)
expansion-rom-version:
bus-info: 0000:62:00.3
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
[root@4gfi-sum-ch1-s4 ~]#
[root@4gfi-sum-ch1-s4 ~]# ethtool -S eno8 | grep rx_no
     rx_no_bufs: 0
[root@4gfi-sum-ch1-s4 ~]#
[root@4gfi-sum-ch1-s4 ~]#
```

c

vNIC statistics from VIC for all OS's

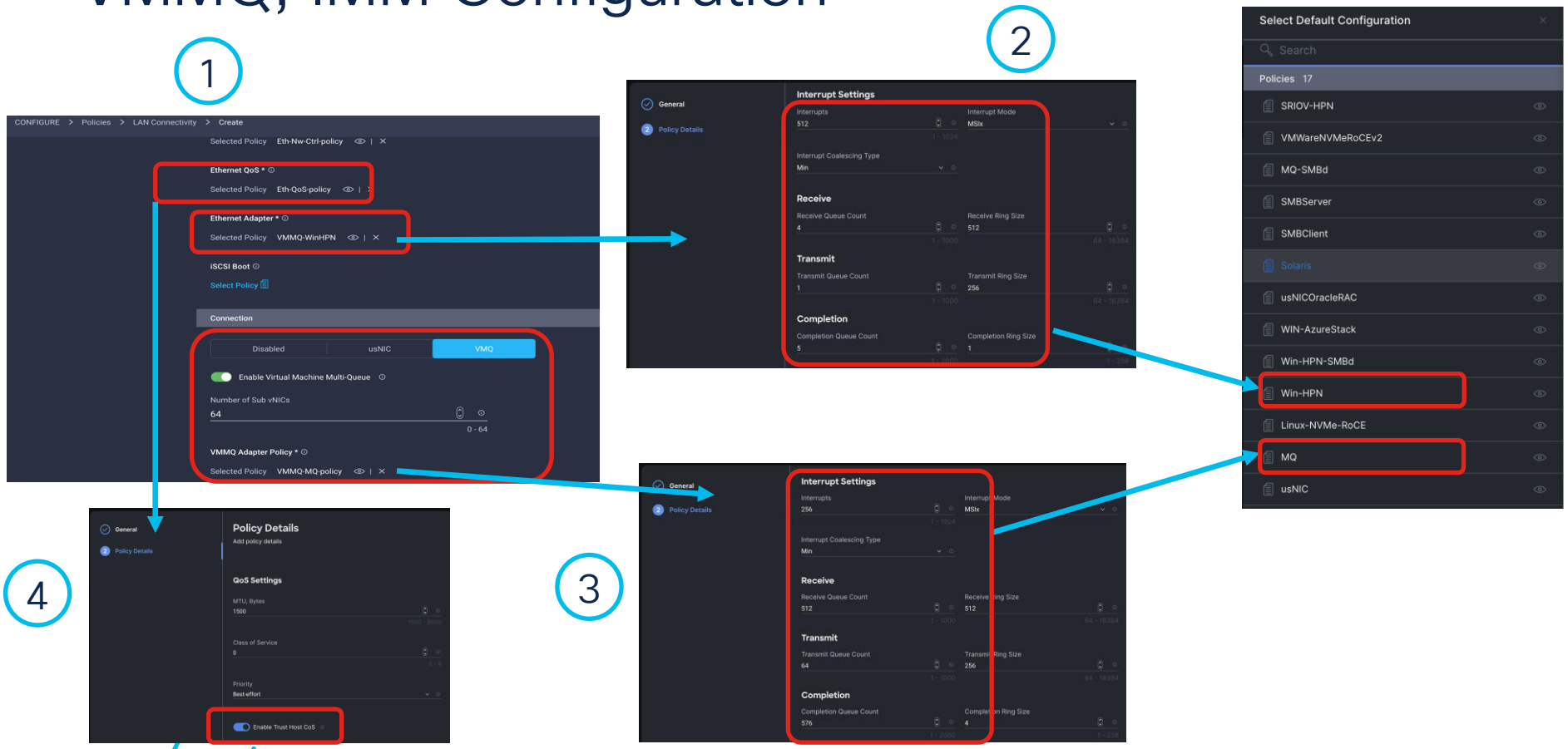
```
Srini-4gfi-VIC-B#
Srini-4gfi-VIC-B# connect adapter
  WORD  Adapter Id (chassis/server/id) or (rack-server/id) (Min size 0, Max size 510)

Srini-4gfi-VIC-B# connect adapter 1/4/1
adapter 1/4/1 #
adapter 1/4/1 # connect
adapter 1/4/1 (top):1#
adapter 1/4/1 (top):1# attach-mcp
adapter 1/4/1 (mcp):1#
adapter 1/4/1 (mcp):1# vnic
vnic id        : internal id of vnic, use for other vnic cmds
vnic name/mac  : ucsm provisioned name (-n) or mac address (-m)
vnic type      : enet=ethernet, enet_pt=dynamic ethernet, fc=fcoe
vnic host      : host
vnic state     : state of vnic
lif            : internal logical if id, use for other lif/vif cmds
vif state      : state of vif
---------------------------------------- --------- ------------------------
              v n i c                       l i f           v i f
 id name              type      host state lif state uif  ucsm  idx vlan state
 --- ---------------- ------- ---- ----- --- ----- --- ----- ---- ---- -----
 27 vnic_1            enet       0 UP    26 UP    =>0  2587  108  601 UP
 28 vnic_2            enet       0 UP    27 UP    =>0  2588  109  269 UP
 29 vnic_3            enet       0 UP    28 UP    =>0  2589  110  602 UP
 30 vnic_4            enet       0 UP    29 UP    =>0  2590  111  603 UP
 31 vnic_5            enet       0 UP    30 UP    =>0  2591  112  603 UP
 32 vnic_6            enet       0 UP    31 UP    =>0  2592  113  605 UP
 33 vnic_7            enet       0 UP    32 UP    =>0  2593  114  606 UP

adapter 1/4/1 (mcp):2# lifstat -a 26
% Command not found
adapter 1/4/1 (mcp):3# lifstats -a 26
          DELTA                TOTAL DESCRIPTION
             34               243205 Tx unicast frames without error
              1                 6831 Tx multicast frames without error
              0                 2852 Tx broadcast frames without error
           2340             17506682 Tx unicast bytes without error
             54               371492 Tx multicast bytes without error
              0              1008360 Tx broadcast bytes without error
           2208             16889708 Rx unicast bytes without error
             60               511436 Rx multicast bytes without error
              0               893100 Rx broadcast bytes without error
              0                    0 Rx rq drop pkts (no bufs or rq disabled)
              0                    0 Rx frames with error
              0                    0 Rx bytes with error
              0                    0 Rx good frames with RSS
              0                 9218 Rx frames len == 64
              1               231048 Rx frames 64 < len <= 127
             32                  148 Rx frames 128 <= len <= 255
              0                 5691 Rx frames 256 <= len <= 511
              0                    0 Rx frames 512 <= len <= 1023
              0                    0 Rx frames 1024 <= len <= 1518
              0                    0 Rx frames len > 1518
     237.361bps                      Tx rate
     224.626bps                      Rx rate
```

VMMQ



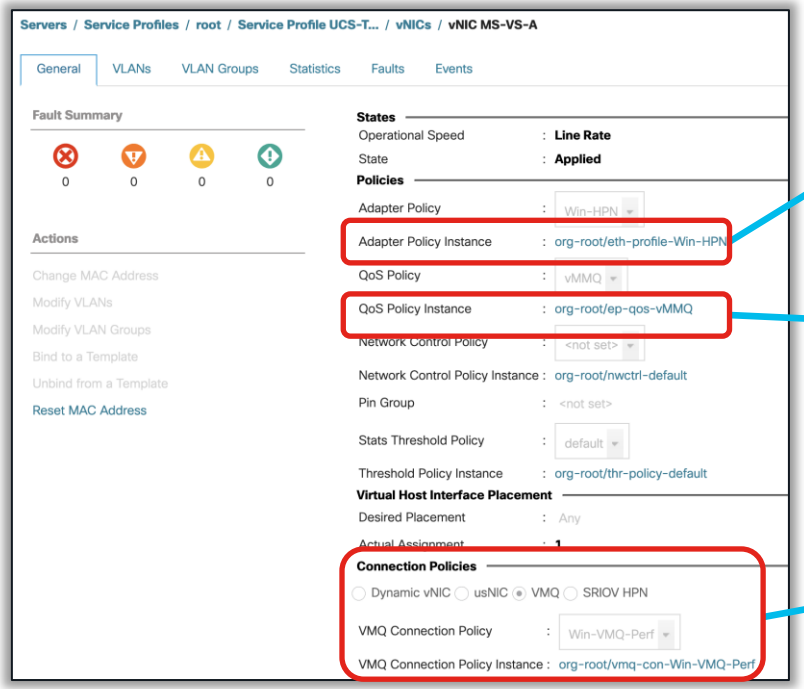
- Virtual Machine Multi-Queue, allows allocating multiple RX queues per vPort in a Windows Hyper-V host.
- Thus, providing higher throughput and distributes traffic load across multiple CPU cores.
- Supported in IMM, UCSM, and CIMC
- VMMQ is recommended over VMQ, or RSS for Windows Hyper-V. Both VMQ and RSS are supported by VIC 15000 for Windows.
- Use the default adapter policy values in Intersight/UCSM of “Win-HPN” and “MQ” to enable VMMQ. And the policy definition is good for 64 vPorts.

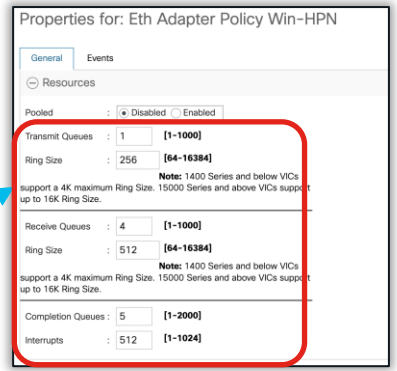
VMMQ, IMM Configuration

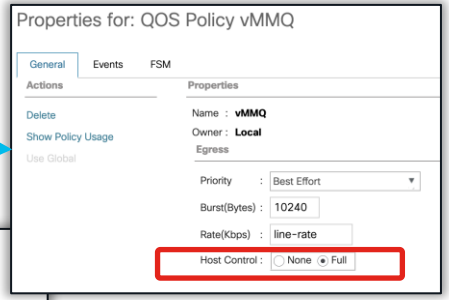


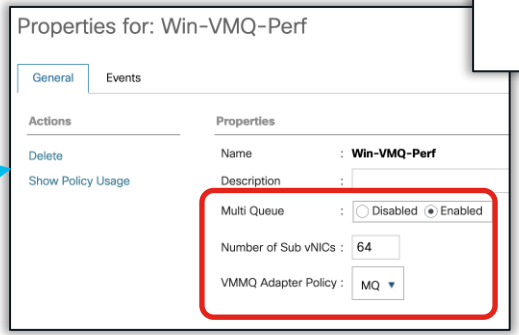
VMMQ, UCSM Configuration

1

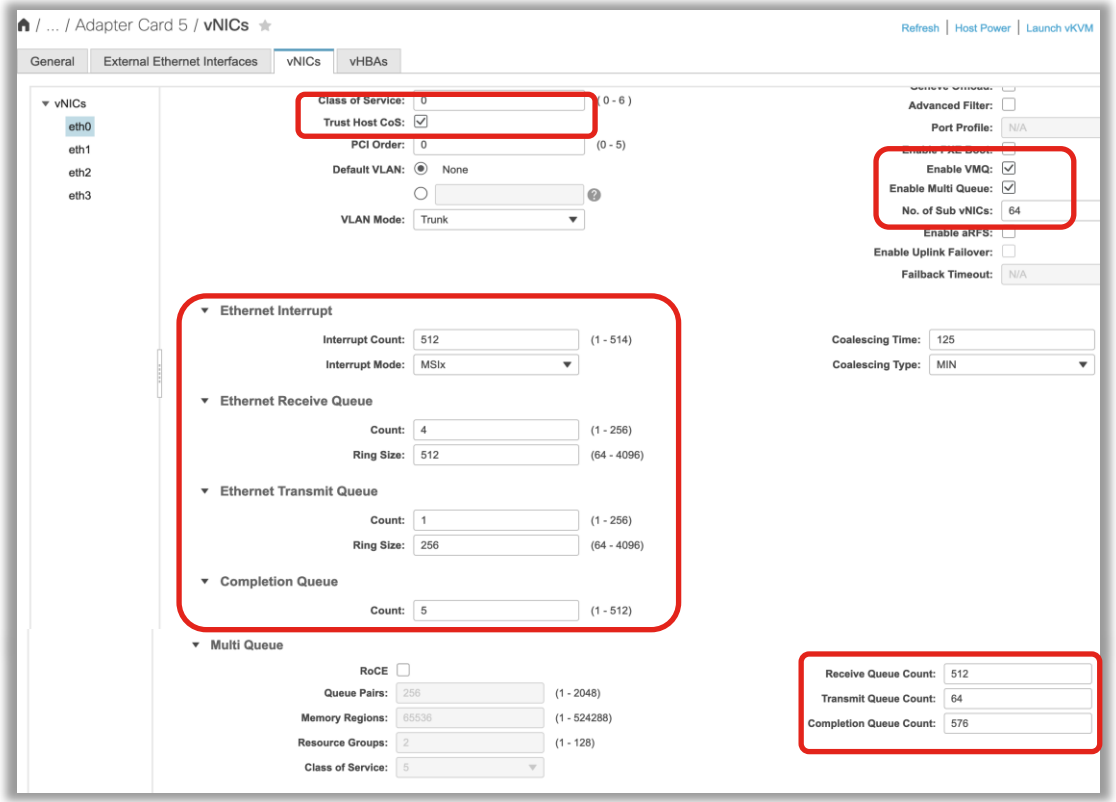




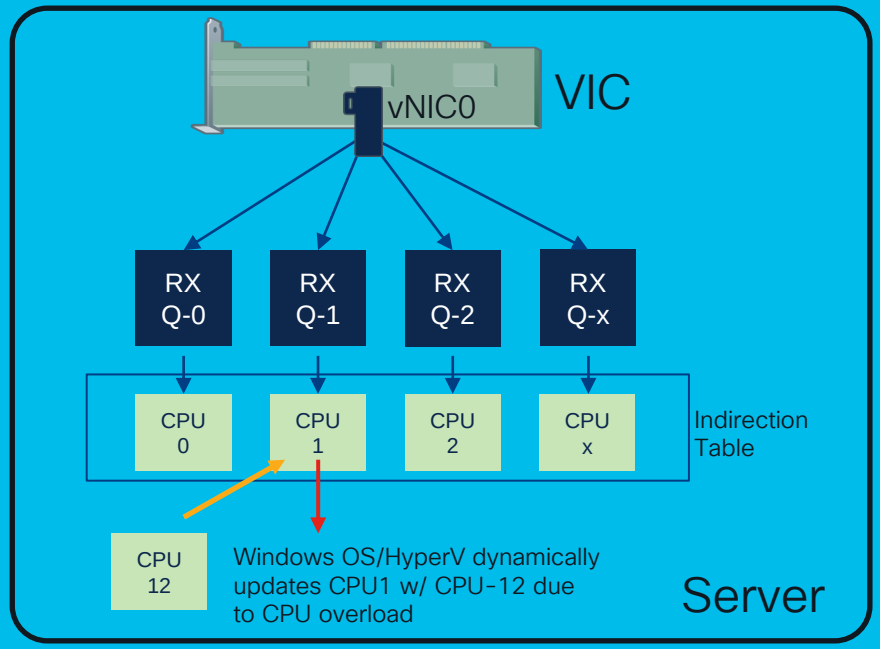




VMMQ, IMC Configuration



RSSv2



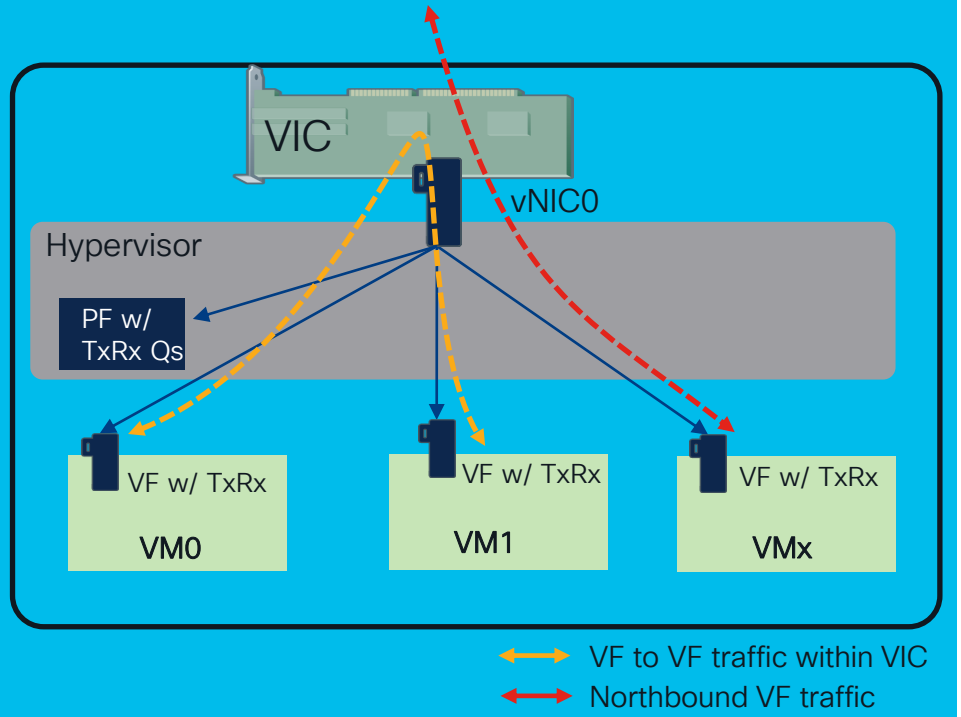
BareMetal ex: but is applicable for VMMQ vPort as well

- RSSv2 is an enhancement to RSS to reduce latency in updating the indirection tables.
- RSSv2 can dynamically spread receive queues over multiple processors much more responsively than RSSv1
- RSSv2 is a windows only driver feature
  - Available from 4.3(2) release
  - Available for bare-metal windows server with RSS and for Hyper-V host with VMMQ (aka dynamic VMMQ).
  - Supported only with VIC 15000 series
- No user configuration needed, just an update to the latest 4.3(2) release and drivers.

Windows Poll Mode

- New NDIS 8.6.5 feature to replace DPC (Deferred procedure call)
- Poll mode will provide better interrupt handling mechanism for packet processing compared to DPC thus delivering better performance in terms of throughput and CPU utilization
- Windows poll mode will be enabled by default on VIC 15000 drivers from 4.3.3 release for Windows 2022 onwards.
- No user action is needed to enable poll mode.
- Features like RSSv2, VMMQ etc are agnostic to this feature and poll mode will complement the performance advantage of these features.
- Note that the windows poll mode should not be confused with poll mode driver in Linux or ESXi.

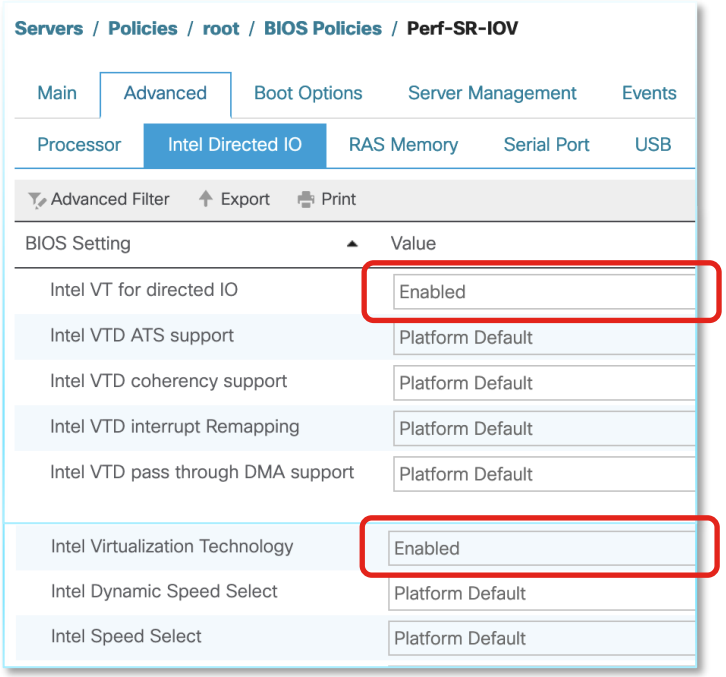
SR-IOV



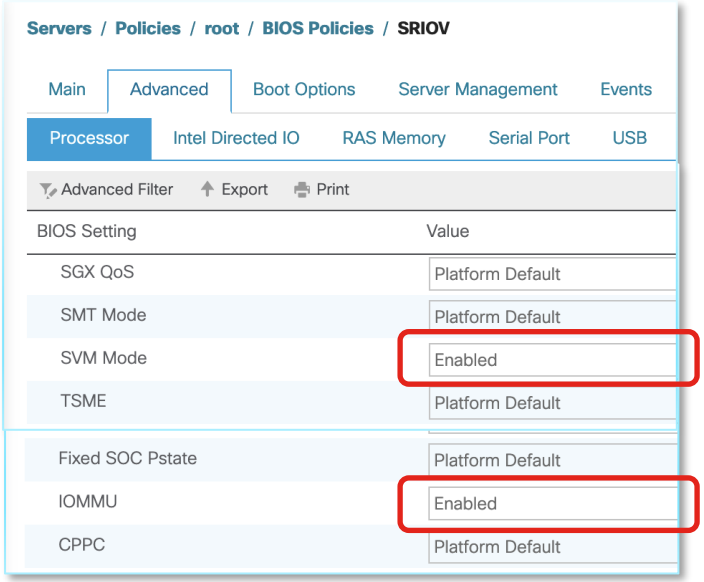
- Single Root I/O Virtualization is a PCIe specification that enables Hypervisor bypass.
- Enables a PCIe physical function (PF) and one or more PCIe virtal function (VF)
- Provides high-throughput but w/ Hypervisor limitations
- 15000 series VIC will support SR-IOV with 4.3(2) release
  - Supported in UCSM, IMM and Cisco IMC
  - First support for ESXi and the Linux guest-os will require latest async driver
  - Linux KVM & Hyper-V support for SR-IOV will be post 4.3(2)
- Steps to enable SR-IOV
  - AMD/Intel specific server BIOS policy settings
  - SR-IOV vNIC settings in UCSM, IMM, or IMC
  - Configuration on HyperVisor

Server BIOS Policy

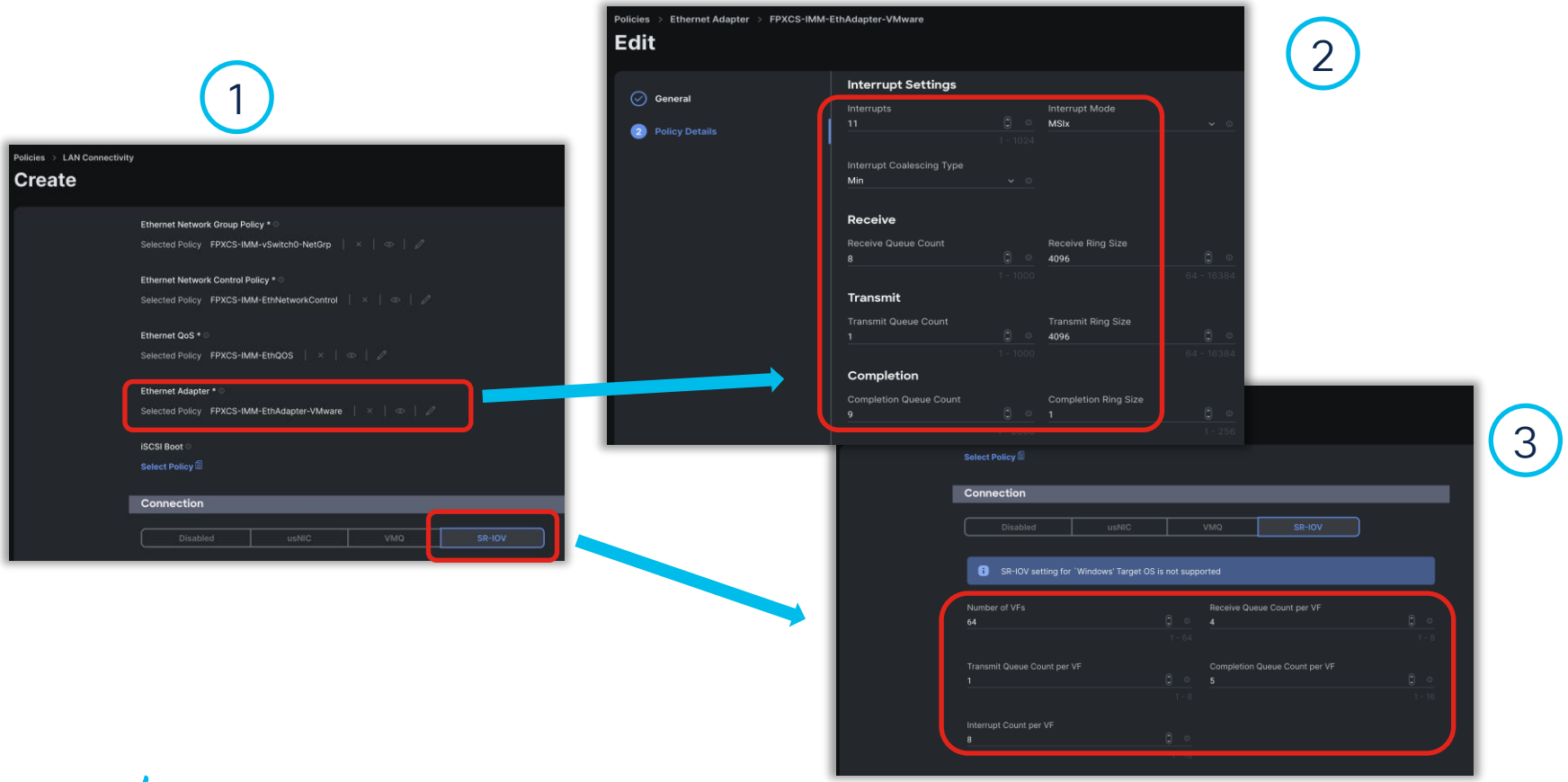
Intel Servers



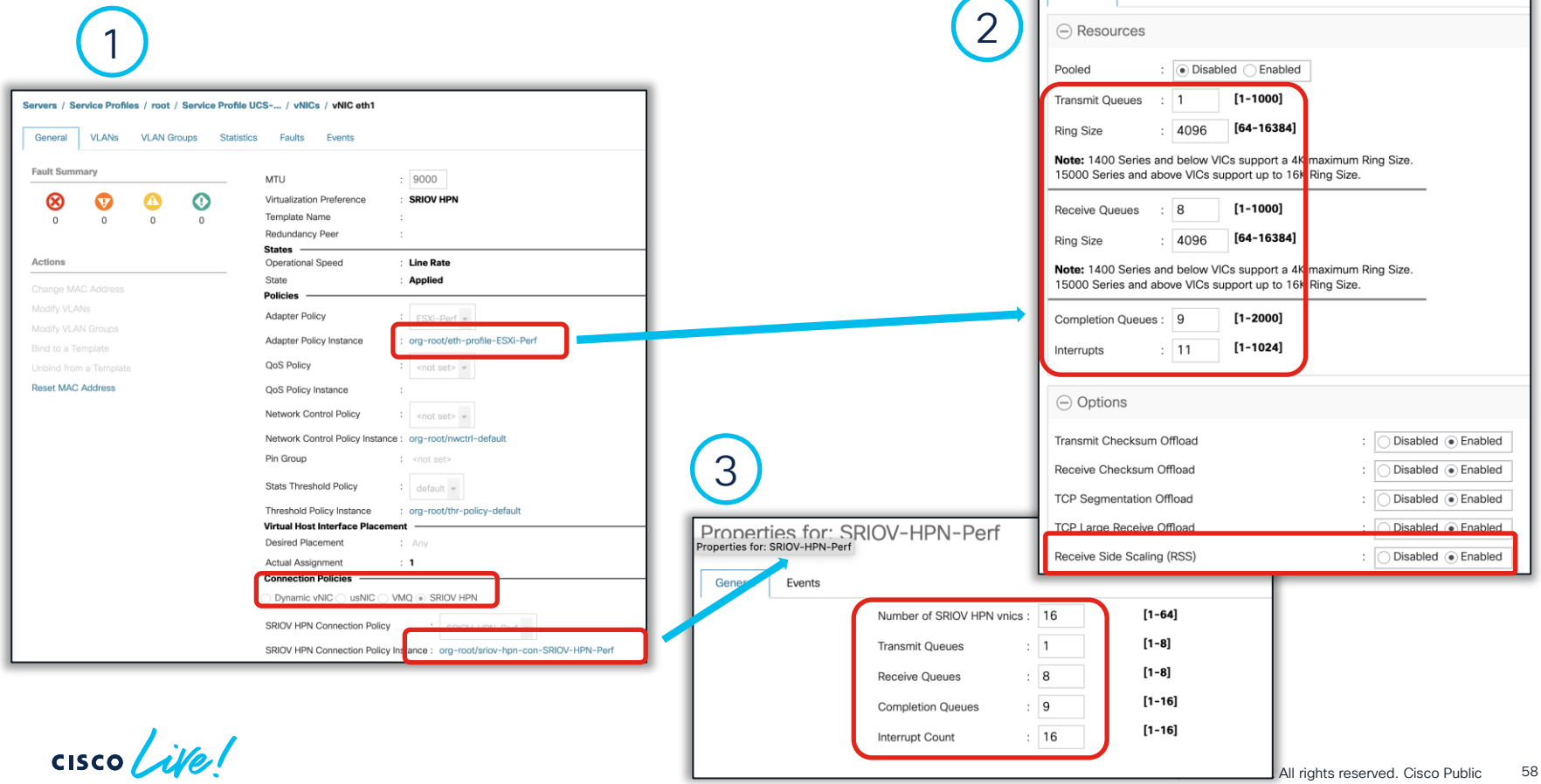
AMD Servers



SR-IOV IMM Configuration

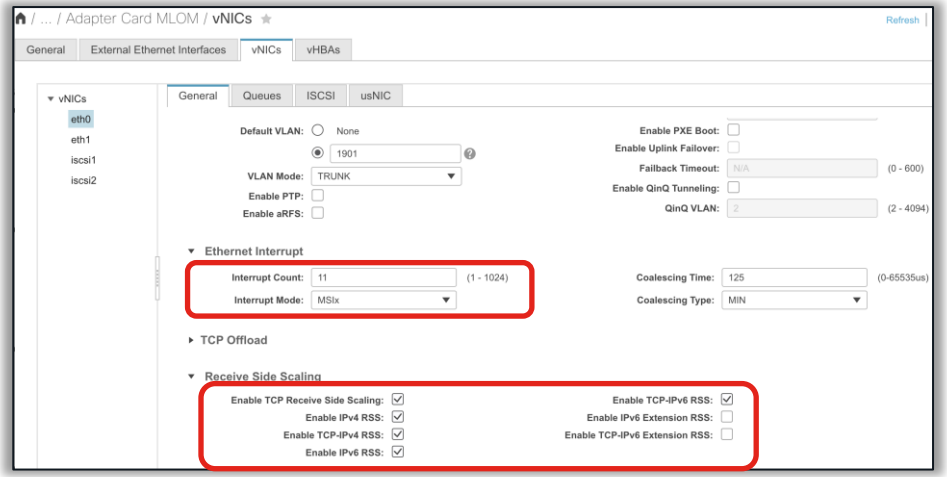


SR-IOV UCSM Configuration

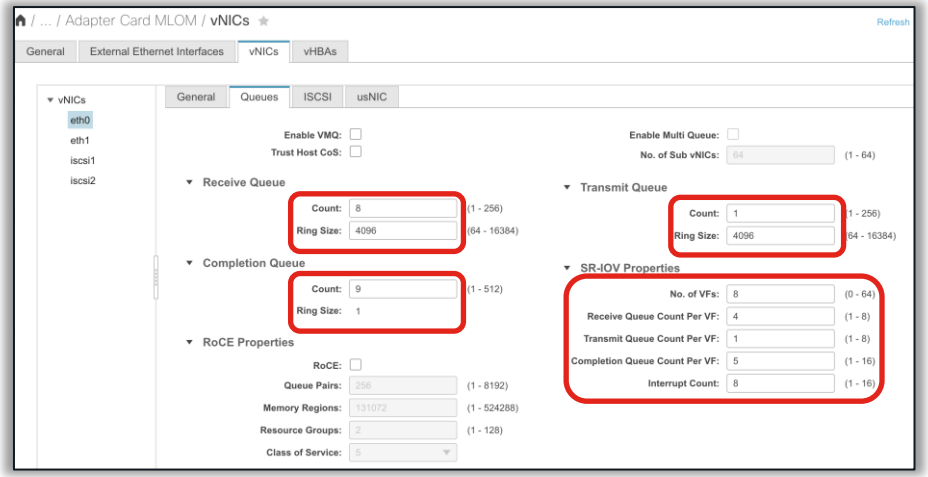


SR-IOV IMC Configuration

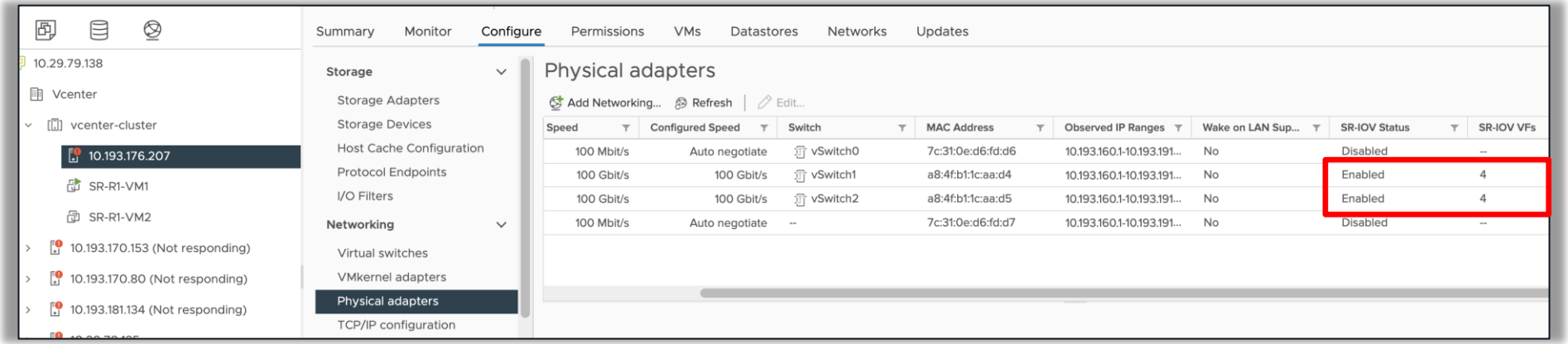
1



2



vCenter Configuration for SR-IOV



- lspci output (PCI ID for VFs is 0x02b7)

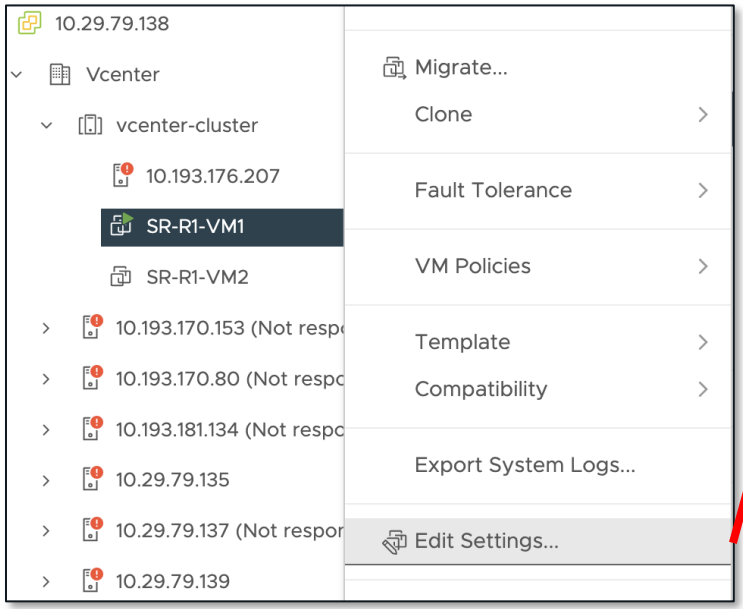
```
[root@esxihost:~] lspci | egrep  "controller" | grep Cisco
0000:36:00.0 Ethernet controller: Cisco Systems Inc Cisco VIC Ethernet NIC
[vmnic1]
0000:36:00.1 Ethernet controller: Cisco Systems Inc Device 02b7 [PF_0.54.0_VF_0]
0000:36:00.2 Ethernet controller: Cisco Systems Inc Device 02b7 [PF_0.54.0_VF_1]
0000:36:00.3 Ethernet controller: Cisco Systems Inc Device 02b7 [PF_0.54.0_VF_2]
0000:36:00.4 Ethernet controller: Cisco Systems Inc Device 02b7 [PF_0.54.0_VF_3]
0000:37:00.0 Ethernet controller: Cisco Systems Inc Cisco VIC Ethernet NIC
[vmnic2]
0000:37:00.1 Ethernet controller: Cisco Systems Inc Device 02b7 [PF_0.55.0_VF_0]
0000:37:00.2 Ethernet controller: Cisco Systems Inc Device 02b7 [PF_0.55.0_VF_1]
0000:37:00.3 Ethernet controller: Cisco Systems Inc Device 02b7 [PF_0.55.0_VF_2]
0000:37:00.4 Ethernet controller: Cisco Systems Inc Device 02b7 [PF_0.55.0_VF_3]
```

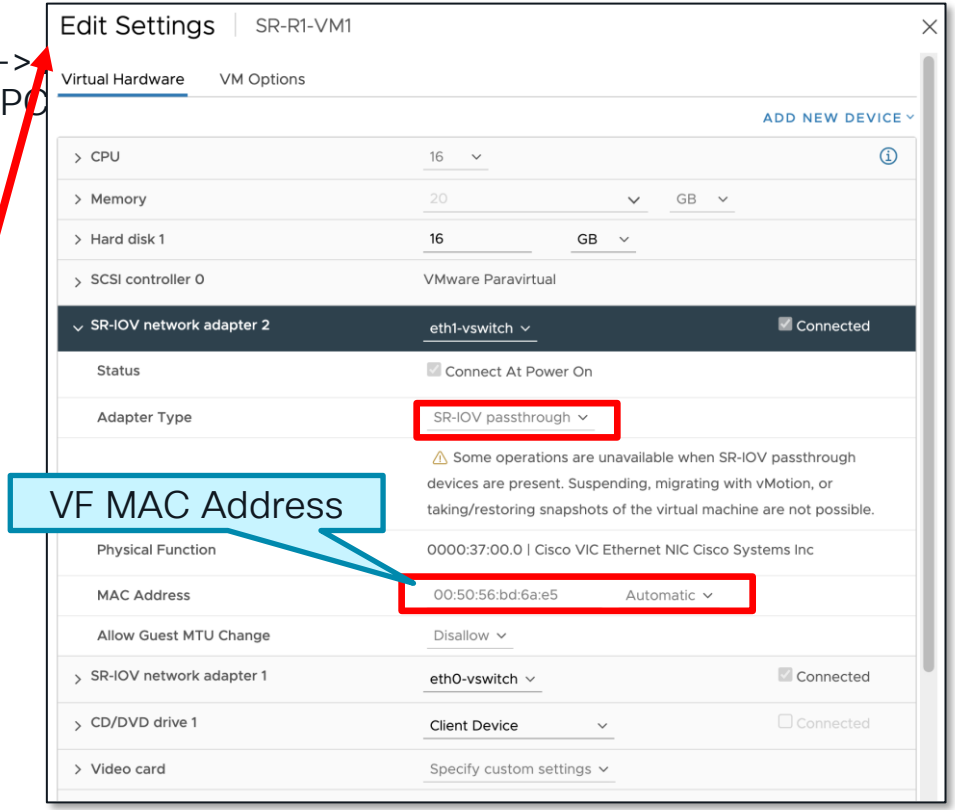
- *esxcfg-nics -l*

```
[root@esxihost:~] esxcfg-nics -l | grep nenic
vmnic1  0000:36:00.0 nenic       Up   100000Mbps Full   a8:4f:b1:1c:aa:d4 1500
vmnic2  0000:37:00.0 nenic       Up   100000Mbps Full   a8:4f:b1:1c:aa:d5 9000
```

Assigning VFs to VMs in vCenter

- Select VM -> Edit Settings -> Add new device -> ‘SRIOV passthrough’ -> Select the desired PFs PC





VF information from Linux VM

- eNIC VF PCI devices on Linux VM

```
# lspci | grep Cisco | grep controller
0b:00.0 Ethernet controller: Cisco Systems Inc Device 02b7 (rev a2)
13:00.0 Ethernet controller: Cisco Systems Inc Device 02b7 (rev a2)
```

- eNIC Network Interfaces

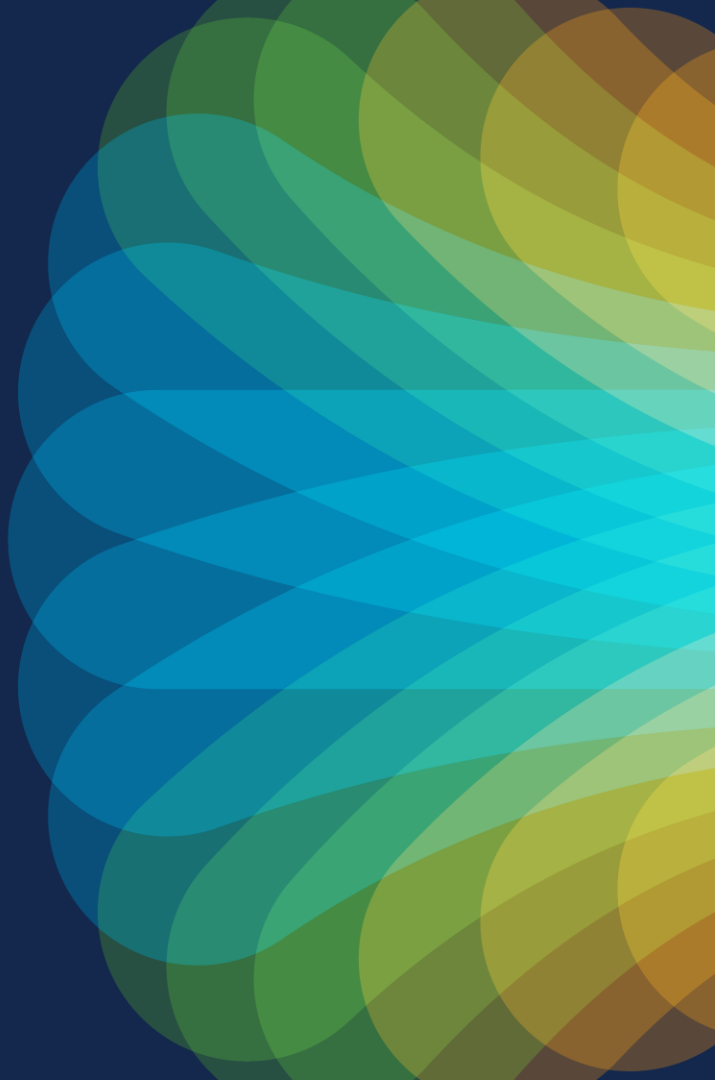
```
#ip addr show
...
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:bd:6a:e5 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 50.6.1.116/24 brd 50.6.1.255 scope global noprefixroute ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::34c3:e653:d9ed:3f62/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:bd:37:5d brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    inet 10.193.176.191/24 brd 10.193.176.255 scope global noprefixroute ens224
       valid_lft forever preferred_lft forever
    inet6 fe80::6243:127:6e26:ccdf/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Note: Linux VMs with secure boot will fail to load enic driver and so either VM secure boot should be disabled or install the Cisco's UCS public key to virtual BIOS. UCS configuration guide will provide the procedure.

What's on the roadmap

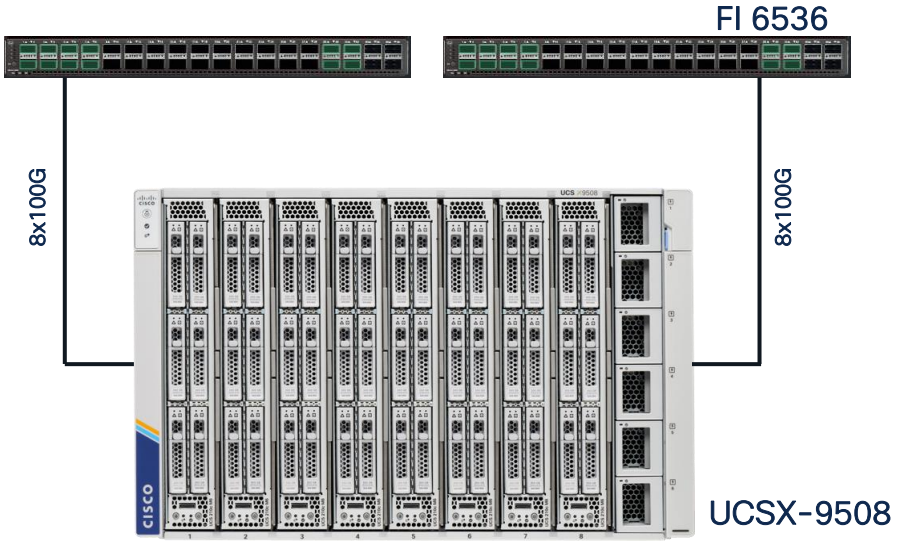
- Ethernet performance improvement for edge workloads
- FC congestion detection improvement
- 50G & 200G qualification with Nexus-9K
- SR-IOV support for Linux & Windows
- Netflow is available with 4th & 5th Gen UCS fabric
- ERSPAN for vNICs & vHBAs with 4th & 5th Gen UCS fabric
- MAC-SEC support on UCS 4th & 5th Gen Fabric Interconnects

VIC Performance



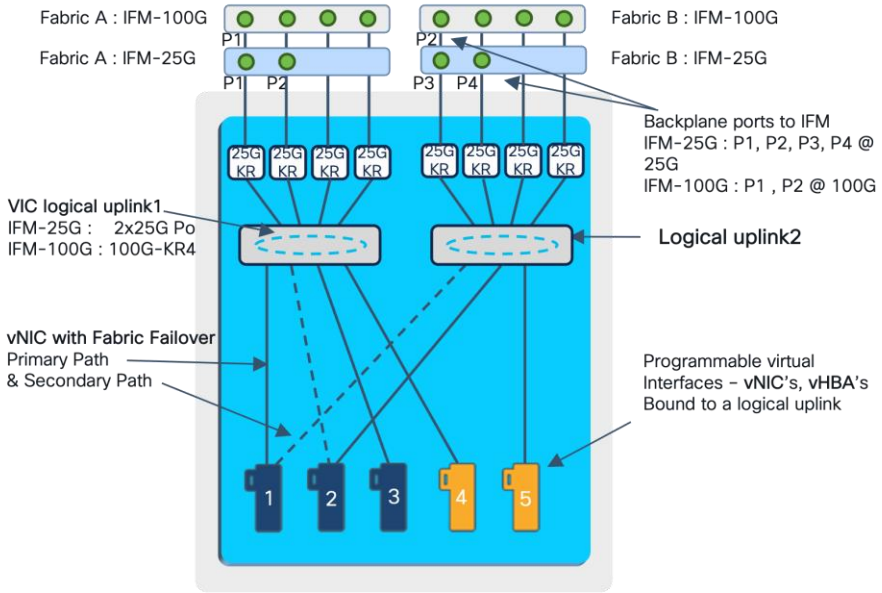
H/W – Setup

- UCS FI 6536
- X9108 IFM 100G
- UCSX-9508/x210c-M7
- VIC 15230 – 2x100G/200G



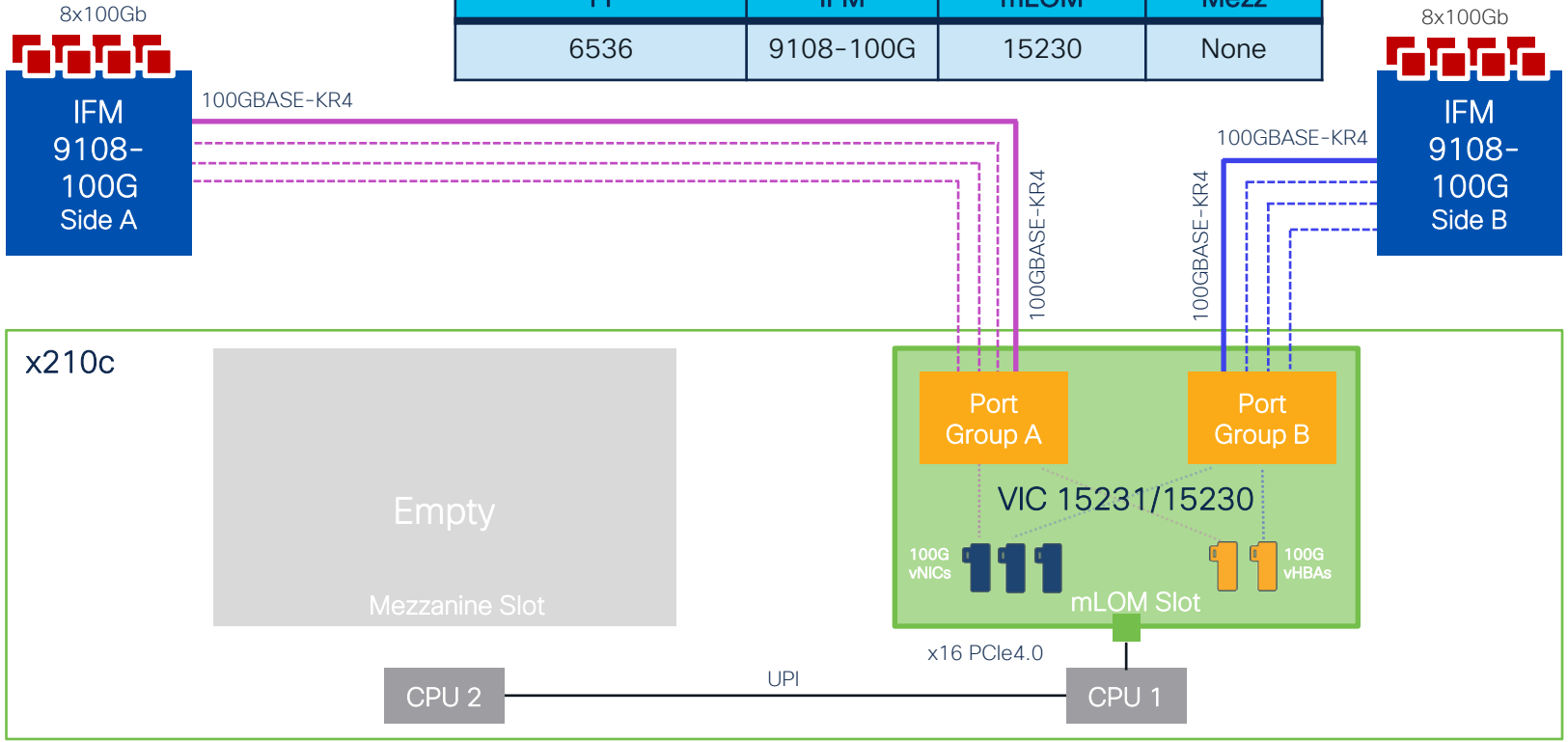
VIC 15230 connectivity in X-series

- MLOM VIC for X210c-M6/M7 with FI 6400/6536
- Connectivity of 100G from VIC :
  - 1 port of 100G-KR4 enabled per IFM-100G
- With IFM-100G the logical uplink is a 100G interface :
  - vNIC/vHBA bound to logical uplink will be of 100G capacity
  - vNIC/vHBA can do single-flow of 100Gbps



IFM9108-100G and VIC 15231/15230 on x210c

FI	IFM	mLOM	Mezz
6536	9108-100G	15230	None



Test Config - I

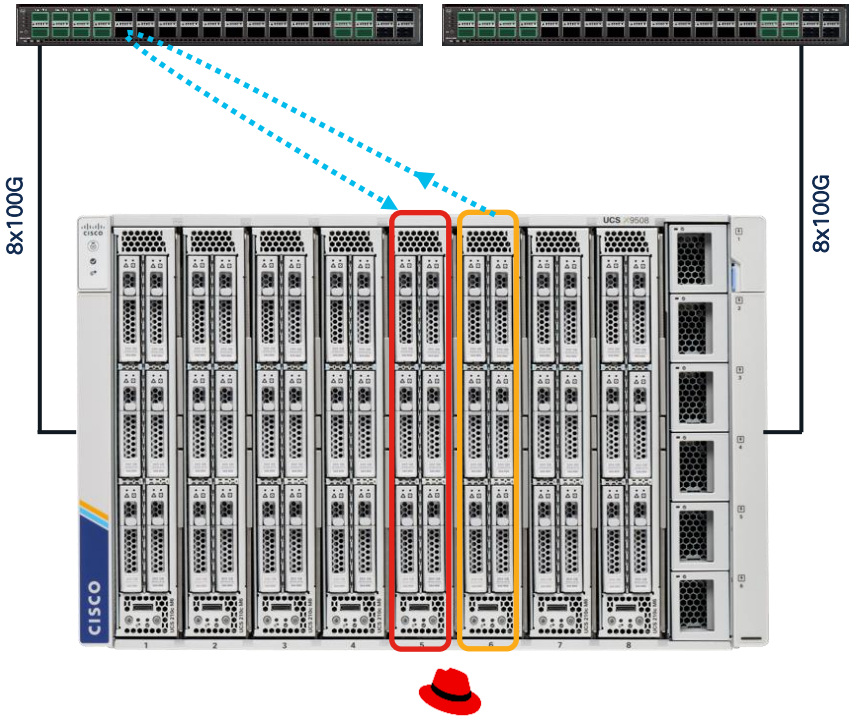
Versions & Configuration

- RHEL - 9.0 GA
- enic Driver - 4.5.0.7-939.23
- VIC f/w - 5.3(2.32)
- Adapter Policy
  - RSS = Enabled
  - MTU = 1500
  - Transmit Queues = 1
  - Receive Queues = 8
  - Completion Queues = 9
  - Interrupts = 10
  - TQ_ring_size = 4096
  - RQ_ring_size = 4096

Test Case – Single Uni-Directional

Server : “iperf -s -B 10.29.161.111”
Client : “iperf -c 10.29.161.111 -t60 -i3 -P16”

Fabric A:Blade5 vnic1 (Server) ← Blade6 vnic1 (Client)




```
[SUM] 27.0-30.0 sec  32.9 GBytes  94.3 Gbits/sec
[ 17] 30.0-33.0 sec  2.06 GBytes  5.90 Gbits/sec
[ 15] 30.0-33.0 sec  97.0 MBytes   271 Mbits/sec
[ 16] 30.0-33.0 sec  96.6 MBytes   270 Mbits/sec
[ 18] 30.0-33.0 sec  2.06 GBytes  5.90 Gbits/sec
[  4] 30.0-33.0 sec  2.71 GBytes  7.76 Gbits/sec
[  3] 30.0-33.0 sec  2.71 GBytes  7.76 Gbits/sec
[  6] 30.0-33.0 sec  2.06 GBytes  5.90 Gbits/sec
[  7] 30.0-33.0 sec  2.72 GBytes  7.77 Gbits/sec
[  9] 30.0-33.0 sec  2.06 GBytes  5.89 Gbits/sec
[ 11] 30.0-33.0 sec  2.05 GBytes  5.88 Gbits/sec
[ 13] 30.0-33.0 sec  2.06 GBytes  5.90 Gbits/sec
[  8] 30.0-33.0 sec  2.71 GBytes  7.77 Gbits/sec
[ 10] 30.0-33.0 sec  2.71 GBytes  7.77 Gbits/sec
[ 12] 30.0-33.0 sec  2.71 GBytes  7.77 Gbits/sec
[ 14] 30.0-33.0 sec  2.06 GBytes  5.91 Gbits/sec
[  5] 30.0-33.0 sec  2.06 GBytes  5.89 Gbits/sec
[SUM] 30.0-33.0 sec  32.9 GBytes  94.3 Gbits/sec
```

Chassis 1 / Blade 6

~100Gig

```
[ ID] Interval        Transfer     Bandwidth
[  4]  0.0-60.0 sec  54.2 GBytes  7.76 Gbits/sec
[  5]  0.0-60.0 sec  54.2 GBytes  7.76 Gbits/sec
[  6]  0.0-60.0 sec  41.0 GBytes  5.87 Gbits/sec
[  7]  0.0-60.0 sec  41.2 GBytes  5.90 Gbits/sec
[  8]  0.0-60.0 sec  54.2 GBytes  7.75 Gbits/sec
[  9]  0.0-60.0 sec  41.1 GBytes  5.88 Gbits/sec
[ 10]  0.0-60.0 sec  41.3 GBytes  5.91 Gbits/sec
[ 11]  0.0-60.0 sec  54.2 GBytes  7.76 Gbits/sec
[ 12]  0.0-60.0 sec  41.4 GBytes  5.92 Gbits/sec
[ 13]  0.0-60.0 sec  54.2 GBytes  7.76 Gbits/sec
[ 15]  0.0-60.0 sec  54.2 GBytes  7.76 Gbits/sec
[ 14]  0.0-60.0 sec  42.0 GBytes  6.01 Gbits/sec
[ 16]  0.0-60.0 sec  2.07 GBytes   296 Mbits/sec
[ 17]  0.0-60.0 sec  2.06 GBytes   295 Mbits/sec
[ 18]  0.0-60.0 sec  41.3 GBytes  5.91 Gbits/sec
[ 19]  0.0-60.0 sec  40.2 GBytes  5.75 Gbits/sec
[SUM]  0.0-60.0 sec   659 GBytes  94.3 Gbits/sec
```

Chassis 1 / Blade 5

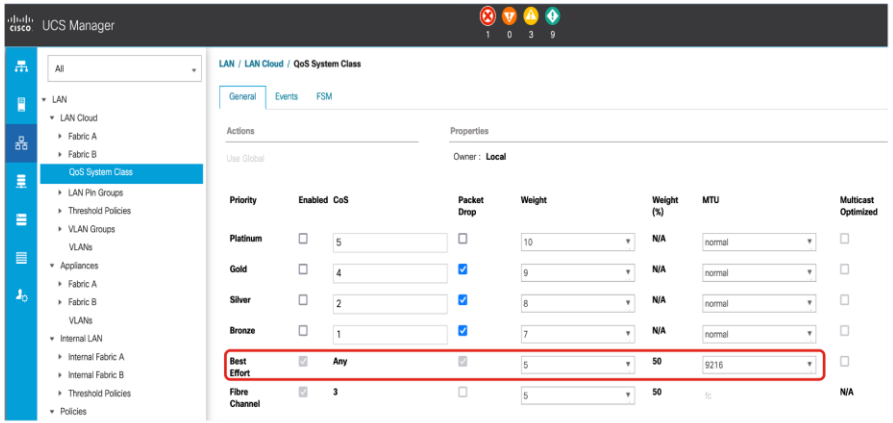
~100Gig

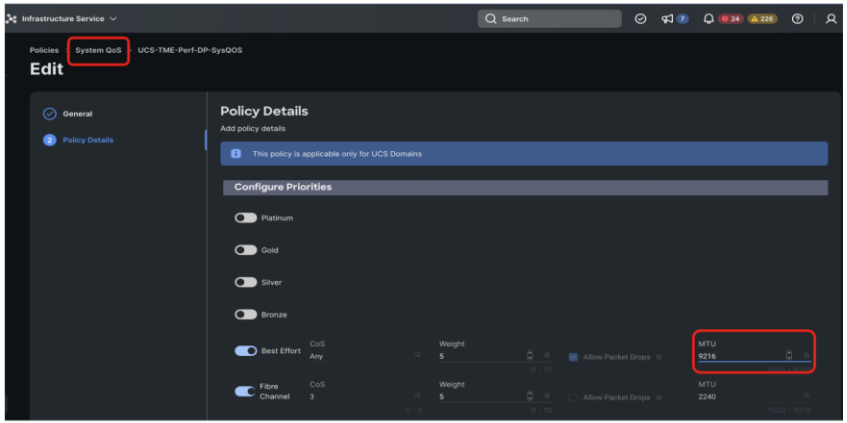
Test Config - II

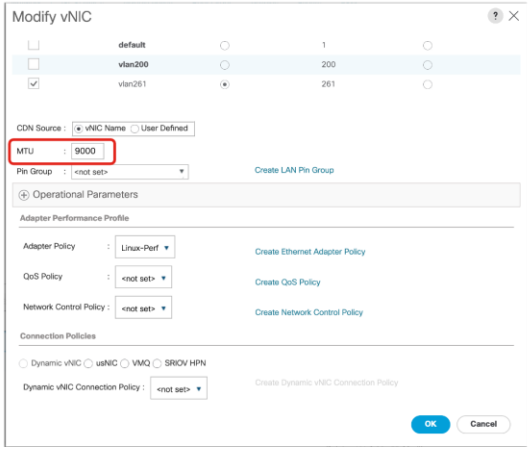
- Versions & Configuration

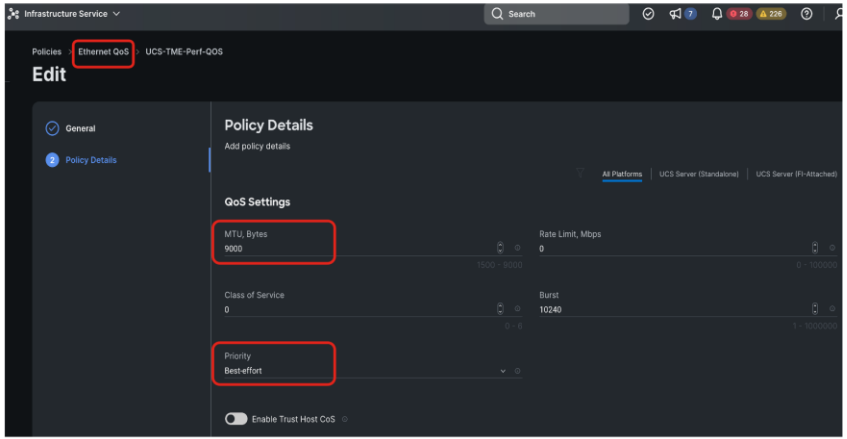
- RHEL - 9.0 GA
- enic Driver - 4.5.0.7-939.23
- VIC f/w - 5.3(2.32)
- Adapter Policy
  - RSS = Enabled
  - MTU = 9000
  - Transmit Queues = 1
  - Receive Queues = 8
  - Completion Queues = 9
  - Interrupts = 10
  - TQ_ring_size = 4096
  - RQ_ring_size = 4096

MTU Settings – System QoS + vNIC





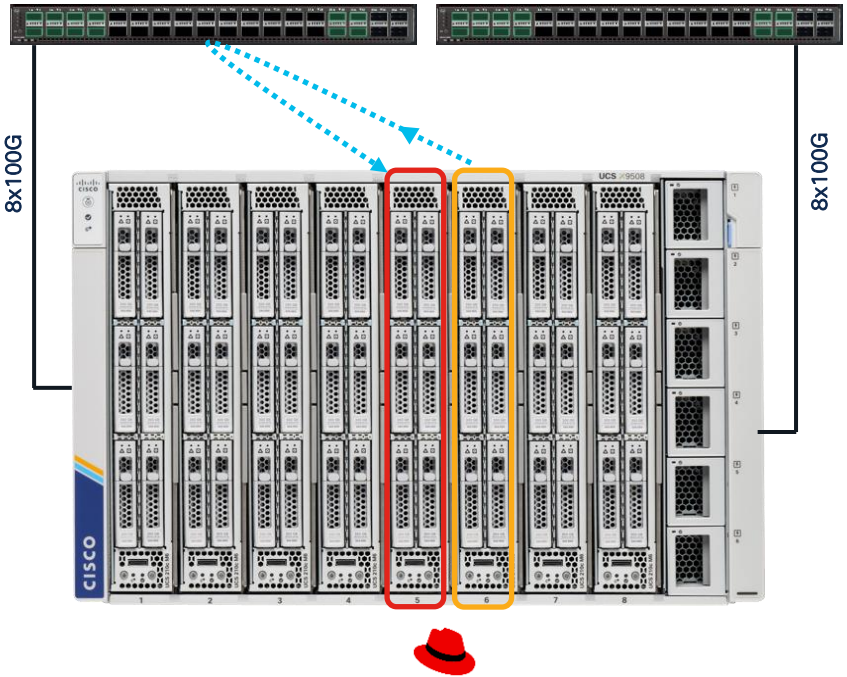




Test Case – Single Uni-Directional

Server - iperf -s -B 10.29.136.171
Client - iperf -c 10.29.136.171 -t60 -i3 -P16

Fabric A:Blade5 vnic1 (Server) ← Blade6 vnic1 (Client)




```
[root@Perf-Blade1-5 ~]# iperf -s -B 10.29.136.171
------------------------------------------------------------
Server listening on TCP port 5001
Binding to local address 10.29.136.171
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.29.136.171 port 5001 connected with 10.29.136.173 port 34542
[  5] local 10.29.136.171 port 5001 connected with 10.29.136.173 port 34544
[  6] local 10.29.136.171 port 5001 connected with 10.29.136.173 port 34546
[  7] local 10.29.136.171 port 5001 connected with 10.29.136.173 port 34548
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-60.0 sec   121 GBytes  17.3 Gbits/sec
[  5]  0.0-60.0 sec   121 GBytes  17.3 Gbits/sec
[  6]  0.0-60.0 sec   328 GBytes  47.0 Gbits/sec
[  7]  0.0-60.0 sec   121 GBytes  17.3 Gbits/sec
[SUM]  0.0-60.0 sec   691 GBytes  98.9 Gbits/sec
```

Chassis 1 / Blade 5

~100Gig

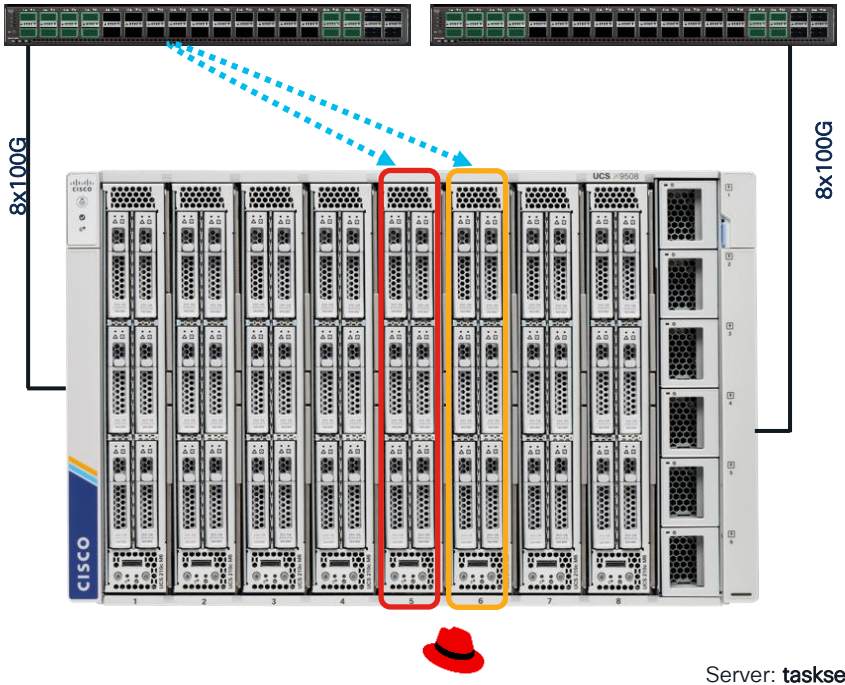
```
[SUM] 54.0-57.0 sec  34.6 GBytes  99.0 Gbits/sec
[  5] 57.0-60.0 sec  16.6 GBytes  47.4 Gbits/sec
[  5]  0.0-60.0 sec   328 GBytes  47.0 Gbits/sec
[  3] 57.0-60.0 sec  6.00 GBytes  17.2 Gbits/sec
[  3]  0.0-60.0 sec   121 GBytes  17.3 Gbits/sec
[  6] 57.0-60.0 sec  6.01 GBytes  17.2 Gbits/sec
[  6]  0.0-60.0 sec   121 GBytes  17.3 Gbits/sec
[  4] 57.0-60.0 sec  6.00 GBytes  17.2 Gbits/sec
[SUM] 57.0-60.0 sec  34.6 GBytes  99.0 Gbits/sec
[  4]  0.0-60.0 sec   121 GBytes  17.3 Gbits/sec
[SUM]  0.0-60.0 sec   691 GBytes  98.9 Gbits/sec
```

Chassis 1 / Blade 6

~100Gig

Test Case – Single Bi-Directional

Fabric A:Blade5 vnic1 (Server) ← Blade6 vnic1 (Client)
Fabric A:Blade5 vnic1 (Client) → Blade6 vnic1 (Server)




```
root@Perf-Blade1-5:~ (ssh)
[  7]  0.0-60.0 sec   148 GBytes  21.1 Gbits/sec
[SUM]  0.0-60.0 sec   690 GBytes  98.8 Gbits/sec
^C[root@Perf-Blade1-5 ~]# clear
[root@Perf-Blade1-5 ~]# taskset -c 0-17 iperf -s -B 10.29.136.171
------------------------------------------------------------
Server listening on TCP port 5001
Binding to local address 10.29.136.171
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 10.29.136.171 port 5001 connected with 10.29.136.173
[  5] local 10.29.136.171 port 5001 connected with 10.29.136.173
[  6] local 10.29.136.171 port 5001 connected with 10.29.136.173
[  7] local 10.29.136.171 port 5001 connected with 10.29.136.173
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-60.0 sec   137 GBytes  19.7 Gbits/sec
[  5]  0.0-60.0 sec   168 GBytes  24.1 Gbits/sec
[  6]  0.0-60.0 sec   149 GBytes  21.3 Gbits/sec
[  7]  0.0-60.0 sec   [illegible]
[SUM]  0.0-60.0 sec   690 GBytes  98.8 Gbits/sec
```

Fabric A:Blade5 vnic1 (Server)

Server: **taskset -c 0-17 iperf -s -B 10.29.136.171**
Client: **taskset -c 0-17 iperf -c 10.29.136.171 -t60 -i3 -P16**

Fabric A:Blade6 vnic1 (Client)

```
[SUM] 54.0-57.0 sec  34.5 GBytes  98.8 Gbits/sec
[  6] 57.0-60.0 sec  11.2 GBytes  32.0 Gbits/sec
[  6]  0.0-60.0 sec   236 GBytes  33.8 Gbits/sec
[  3] 57.0-60.0 sec  7.75 GBytes  22.2 Gbits/sec
[  3]  0.0-60.0 sec   137 GBytes  19.7 Gbits/sec
[  4] 57.0-60.0 sec  7.80 GBytes  22.3 Gbits/sec
[  4]  0.0-60.0 sec   168 GBytes  24.1 Gbits/sec
[  5] 57.0-60.0 sec  7.79 GBytes  22.3 Gbits/sec
[SUM] 57.0-60.0 sec  34.5 GBytes  98.8 Gbits/sec
[  5]  0.0-60.0 sec   149 GBytes  21.3 Gbits/sec
[SUM]  0.0-60.0 sec   690 GBytes  98.8 Gbits/sec
```

```
[SUM] 54.0-57.0 sec  34.5 GBytes  98.9 Gbits/sec
[  6] 57.0-60.0 sec  5.76 GBytes  16.5 Gbits/sec
[  6]  0.0-60.0 sec   165 GBytes  23.6 Gbits/sec
[  5] 57.0-60.0 sec  11.5 GBytes  33.0 Gbits/sec
[  5]  0.0-60.0 sec   181 GBytes  26.0 Gbits/sec
[  3] 57.0-60.0 sec  5.76 GBytes  16.5 Gbits/sec
[  3]  0.0-60.0 sec   129 GBytes  18.5 Gbits/sec
[  4] 57.0-60.0 sec  11.5 GBytes  33.0 Gbits/sec
[SUM] 57.0-60.0 sec  34.5 GBytes  98.9 Gbits/sec
[  4]  0.0-60.0 sec   215 GBytes  30.7 Gbits/sec
[SUM]  0.0-60.0 sec   690 GBytes  98.8 Gbits/sec
```

Fabric A:Blade5 vnic1 (Client)

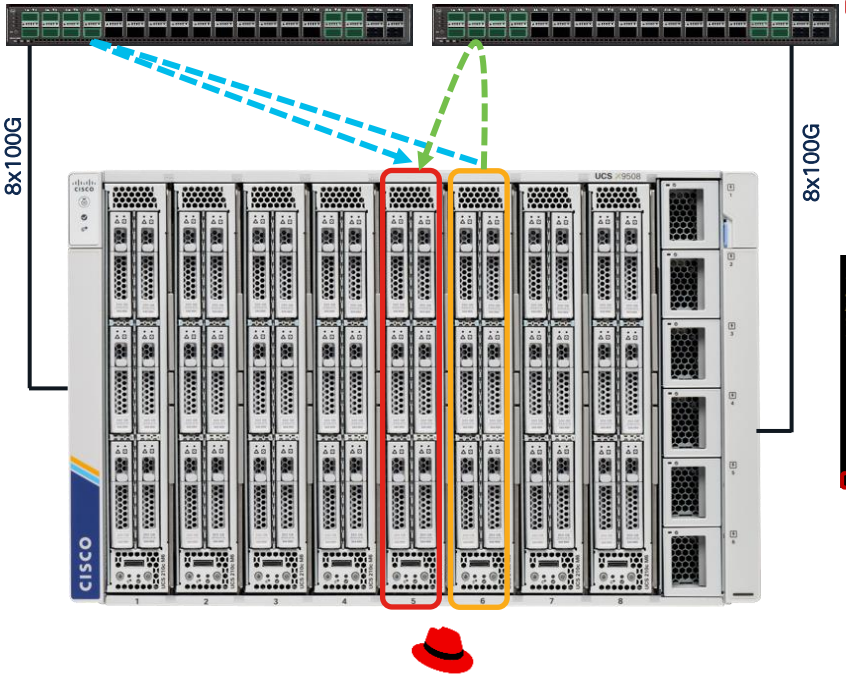
Fabric A:Blade6 vnic1 (Server)

```
root@Perf-Blade1-6:~ (ssh)
[root@Perf-Blade1-6 ~]# taskset -c 0-17 iperf -s -B 10.29.136.173
------------------------------------------------------------
Server listening on TCP port 5001
Binding to local address 10.29.136.173
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 10.29.136.173 port 5001 connected with 10.29.136.171
[  5] local 10.29.136.173 port 5001 connected with 10.29.136.171
[  6] local 10.29.136.173 port 5001 connected with 10.29.136.171
[  7] local 10.29.136.173 port 5001 connected with 10.29.136.171
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-60.0 sec   181 GBytes  26.0 Gbits/sec
[  5]  0.0-60.0 sec   215 GBytes  30.7 Gbits/sec
[  6]  0.0-60.0 sec   129 GBytes  18.5 Gbits/sec
[  7]  0.0-60.0 sec   165 GBytes  23.6 Gbits/sec
[SUM]  0.0-60.0 sec   690 GBytes  98.8 Gbits/sec
```

Server: **taskset -c 18-35 iperf -s -B 10.29.136.173**
Client: **taskset -c 18-35 iperf -c 10.29.136.173 -t60 -i 3 -P16**

Test Case – Dual Uni-Directional

Fabric A:Blade5 vnic1 (Server) ← Blade6 vnic1 (Client)
Fabric B:Blade5 vnic2 (Server) ← Blade6 vnic2 (Client)




```
[root@Perf-Blade1-5 ~]# taskset -c 0-17 iperf -s -B 10.29.136.171
------------------------------------------------------------
Server listening on TCP port 5001
Binding to local address 10.29.136.171
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 10.29.136.171 port 5001 connected with 10.29.136.173 port 56960
[  5] local 10.29.136.171 port 5001 connected with 10.29.136.173 port 56958
[  6] local 10.29.136.171 port 5001 connected with 10.29.136.173 port 56962
[  7] local 10.29.136.171 port 5001 connected with 10.29.136.173 port 56964
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-60.0 sec   169 GBytes  24.2 Gbits/sec
[  5]  0.0-60.0 sec   177 GBytes  25.4 Gbits/sec
[  6]  0.0-60.0 sec   170 GBytes  24.3 Gbits/sec
[  7]  0.0-60.0 sec   174 GBytes  25.0 Gbits/sec
[SUM]  0.0-60.0 sec   691 GBytes  98.9 Gbits/sec
```

Fabric-A: Blade5 vnic1 (Server)

Fabric-A: Blade6 vnic1 (Client)

```
[SUM] 54.0-57.0 sec   34.5 GBytes   98.9 Gbits/sec
[  6] 57.0-60.0 sec   8.63 GBytes   24.7 Gbits/sec
[  6]  0.0-60.0 sec    174 GBytes   25.0 Gbits/sec
[  3] 57.0-60.0 sec   8.65 GBytes   24.8 Gbits/sec
[  3]  0.0-60.0 sec    177 GBytes   25.4 Gbits/sec
[  5] 57.0-60.0 sec   8.60 GBytes   24.6 Gbits/sec
[  5]  0.0-60.0 sec    170 GBytes   24.3 Gbits/sec
[  4] 57.0-60.0 sec   8.65 GBytes   24.8 Gbits/sec
[SUM] 57.0-60.0 sec   34.5 GBytes   98.9 Gbits/sec
[  4]  0.0-60.0 sec    169 GBytes   24.2 Gbits/sec
[SUM]  0.0-60.0 sec    691 GBytes   98.9 Gbits/sec
```

```
[root@Perf-Blade1-5 ~]# taskset -c 18-35 iperf -s -B 10.29.161.111
------------------------------------------------------------
Server listening on TCP port 5001
Binding to local address 10.29.161.111
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 10.29.161.111 port 5001 connected with 10.29.161.112 port 55938
[  5] local 10.29.161.111 port 5001 connected with 10.29.161.112 port 55936
[  6] local 10.29.161.111 port 5001 connected with 10.29.161.112 port 55940
[  7] local 10.29.161.111 port 5001 connected with 10.29.161.112 port 55942
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-60.0 sec   162 GBytes  23.2 Gbits/sec
[  5]  0.0-60.0 sec   228 GBytes  32.6 Gbits/sec
[  6]  0.0-60.0 sec   162 GBytes  23.2 Gbits/sec
[  7]  0.0-60.0 sec   138 GBytes  19.8 Gbits/sec
[SUM]  0.0-60.0 sec   691 GBytes  98.9 Gbits/sec
```

Fabric-B: Blade5 vnic2 (Server)

Fabric-B: Blade6 vnic2 (Client)

```
[SUM] 54.0-57.0 sec   34.5 GBytes   98.9 Gbits/sec
[  5] 57.0-60.0 sec   8.63 GBytes   24.7 Gbits/sec
[  5]  0.0-60.0 sec    138 GBytes   19.8 Gbits/sec
[  3] 57.0-60.0 sec   8.64 GBytes   24.7 Gbits/sec
[  3]  0.0-60.0 sec    162 GBytes   23.2 Gbits/sec
[  6] 57.0-60.0 sec   8.63 GBytes   24.7 Gbits/sec
[  6]  0.0-60.0 sec    162 GBytes   23.2 Gbits/sec
[  4] 57.0-60.0 sec   8.63 GBytes   24.7 Gbits/sec
[SUM] 57.0-60.0 sec   34.5 GBytes   98.9 Gbits/sec
[  4]  0.0-60.0 sec    228 GBytes   32.6 Gbits/sec
[SUM]  0.0-60.0 sec    691 GBytes   98.9 Gbits/sec
```

Test Config – III

Versions & Configuration

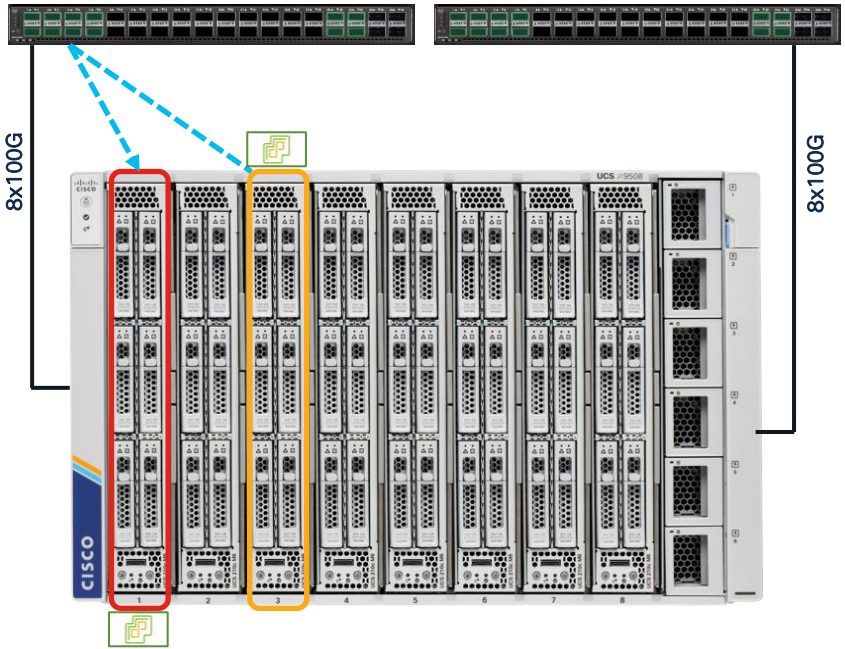
- ESXi – 7.0 U3
- nenic Driver – 1.0.45.0
- VIC f/w – 5.3(2.32)
- Adapter Policy
  - RSS = Enabled
  - MTU = 9000
  - Transmit Queues = 1
  - Receive Queues = 8
  - Completion Queues = 9
  - Interrupts = 11
  - TQ_ring_size = 4096
  - RQ_ring_size = 4096
  - flags = tso|rxcsum|txcsum|failover|lro

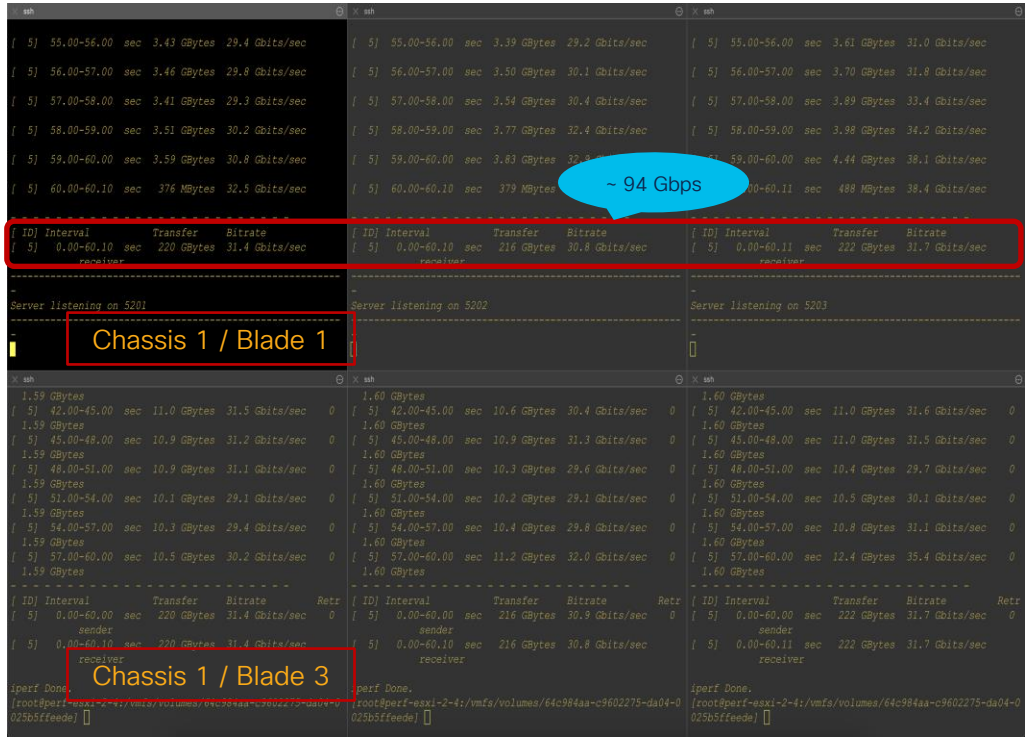
Test Case – Single Uni-Directional(BMs)

Server - iperf -s -B 10.29.161.122 -p5201|5202|5203
Client - iperf -c 10.29.161.122 -t60 -i3 -P1 -p5201|5202|5203

Fabric A: Blade1 vnic1 (3xServer) ← Blade3 vnic1 (3xClient)

*** Between Hypervisors Only with RSS | No guest VMs ***





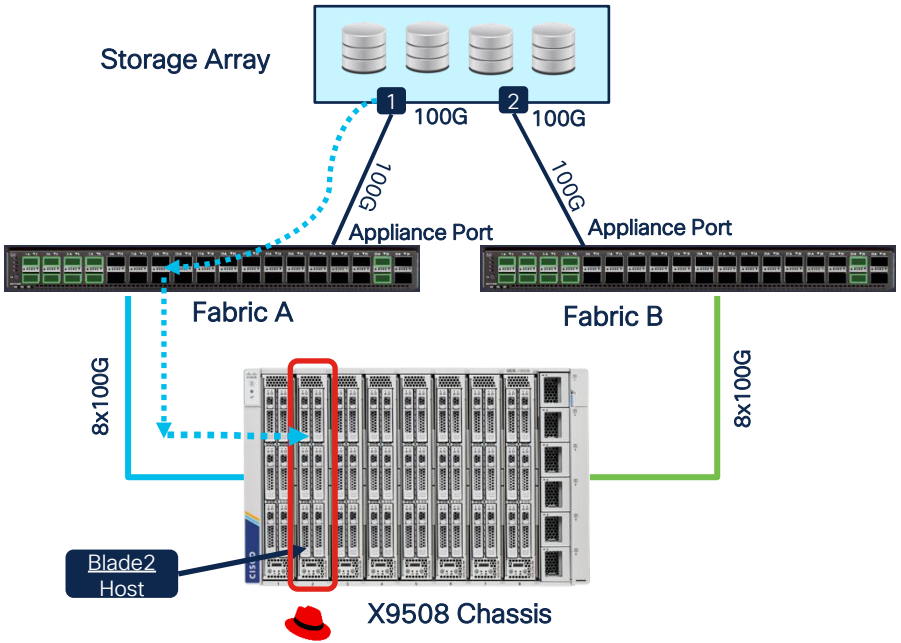
Test Config – IV : NFS 100Gig

H/W, S/W versions & configuration

- FI 6536 with X9508 chassis
- X9108 IFM 100G
- VIC 15231/15230 adapter
- RHEL OS 8.4
- Adapter policy :
  - 1 Tx, 8 Rx, 9 CQ, 10 Interrupts
  - Tx, Rx ring-size of 4K or 16K
- BIOS policy : default
- OS Tuning : default
- Storage array with 100G adapters

Test Case – NFS over 100G with VIC 15231/15230

100G NFS connectivity to a Storage Filer:
Blade2 vNIC1 <-- 100G NIC on Storage (100% read)




```
[root@ora5g-node1 ~]# fio --name=test1 --filename=/fiovol1/data1:/fiovol1/data2:/fiovol1/data3:/fiovol1/data4:/fiovol2
/fiovol2/data22:/fiovol2/data23:/fiovol2/data24:/fiovol3/data31:/fiovol3/data32:/fiovol3/data33:/fiovol3/data34:/fiovo
l:/fiovol4/data42:/fiovol4/data43:/fiovol4/data44 --rw=read --direct=1 --ioengine=libaio --bs=512k --numjobs=16 --iode
-runtime=300 --time_based --group_reporting
test1: (g=0): rw=read, bs=(R) 512KiB-512KiB, (W) 512KiB-512KiB, (T) 512KiB-512KiB, ioengine=libaio, iodepth=64
...
fio-3.19
Starting 16 processes
Jobs: 16 (f=256): [R(16)][100.0%][r=11.3GiB/s][r=23.2k IOPS][eta 00m:00s]
test1: (groupid=0, jobs=16): err= 0: pid=82192: Fri Apr 22 13:53:52 2022
  read: IOPS=23.3k, BW=11.4GiB/s (12.2GB/s)(3408GiB/300028msec)
    slat (usec): min=26, max=5329, avg=56.33, stdev=25.13
    clat (usec): min=1957, max=531941, avg=43939.16, stdev=9451.20
     lat (usec): min=1997, max=531987, avg=43995.59, stdev=9450.83
    clat percentiles (msec):
     |  1.00th=[   25],  5.00th=[   28], 10.00th=[   31], 20.00th=[   38],
     | 30.00th=[   41], 40.00th=[   44], 50.00th=[   46], 60.00th=[   47],
     | 70.00th=[   49], 80.00th=[   51], 90.00th=[   53], 95.00th=[   56],
     | 99.00th=[   62], 99.50th=[   66], 99.90th=[  103], 99.95th=[  131],
     | 99.99th=[  205]
   bw (  MiB/s): min= 5977, max=12232, per=100.00%, avg=11646.89, stdev=19.82, samples=9584
   iops        : min=11947, max=24465, avg=23293.77, stdev=39.65, s
  lat (msec)   : 2=0.01%, 4=0.01%, 10=0.01%, 20=0.06%, 50=78.4
  lat (msec)   : 100=21.39%, 250=0.10%, 500=0.01%, 750=0.01%
  cpu          : usr=0.22%, sys=8.45%, ctx=6244500, majf=0,
  IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=                    =64=0.0%
     issued rwts: total=6980099,0,0,0 short=0,0,0,0 dropped=0
     latency   : target=0, window=0, percentile=100.00%,                    =64

Run status group 0 (all jobs):
   READ: bw=11.4GiB/s (12.2GB/s), 11.4GiB/s-11.4GiB/s (12.2GB/s-12.2GB/s), io=3408GiB (3660GB), run=300028-300028msec
```

~100G NFS

```
FlexPod-A800: cluster.cluster: 4/22/2022 13:50:07
  cpu  cpu    total                    fcache    total    total data     data     data cluster  cluster  cluster     disk     disk     pkts     pkts
  avg busy      ops  nfs-ops cifs-ops      ops     recv     sent busy     recv     sent    busy     recv     sent     read    write     recv     sent
 ---- ---- -------- -------- -------- -------- -------- -------- ---- -------- -------- ------- -------- -------- -------- -------- -------- --------
  18%  43%   187117   187117        0        0     38.1MB   11.4GB  97%    37.7MB   11.4GB      0%    338KB    336KB    478MB     1.71MB   230535  1371110
  18%  42%   186775   186775        0        0     38.4MB   11.5GB  98%    38.1MB   11.5GB      0%    285KB    286KB    481MB     3.91MB   236859  1375341
  18%  43%   187603   187603        0        0     39.0MB   11.5GB  98%    38.7MB   11.5GB      0%    286KB    285KB    480MB     23.9KB   245848  1376171
  18%  43%   186820   186820        0        0     38.0MB   11.4GB  98%    37.8MB   11.4GB      0%    266KB    266KB    609MB     1.16MB   231085  1371855
  18%  43%   186775   186775        0        0     38.5MB   11.5GB  98%    38.2MB   11.5GB      0%    301KB    301KB    643MB     17.9KB   237274  1374307
  18%  43%   186634   186634        0        0     38.4MB   11.4GB  97%    38.1MB   11.4GB      0%    251KB    251KB    613MB     1.46MB   239374  1370362
  18%  43%   185568   185568        0        0     37.7MB   11.4GB  97%    37.4MB   11.4GB      0%    278KB    278KB    616MB     3.91MB   226770  1369212
  18%  43%   187085   187085        0        0     38.3MB   11.4GB  97%    38.0MB   11.4GB      0%    307KB    307KB    582MB     23.9KB   235274  1365179
  18%  43%   186116   186116        0        0     38.4MB   11.4GB  98%    38.2MB   11.4GB      0%    269KB    269KB    577MB     1.17MB   237298  1369754
  18%  43%   186856   186856        0        0     38.1MB   11.4GB  98%    37.8MB   11.4GB      0%    270KB    270KB    575MB     29.9KB   234293  1372264
  18%  43%   186380   186380        0        0     38.5MB   11.4GB  97%    38.2MB   11.4GB      0%    281KB    281KB    590MB     646KB    238783  1371946
  18%  43%   186480   186480        0        0     38.6MB   11.4GB  97%    38.3MB   11.4GB      0%    309KB    309KB    577MB     2.25MB   241873  1366584
  18%  43%   185636   185636        0        0     38.2MB   11.4GB  97%    37.9MB   11.4GB      0%    284KB    282KB    568MB     23.9KB   235542  1368954
  18%  43%   185936   185936        0        0     37.6MB   11.3GB  97%    37.3MB   11.3GB      0%    280KB    282KB    594MB     1.18MB   225957  1360859
  18%  43%   187402   187402        0        0     38.3MB   11.5GB  98%    38.0MB   11.5GB      0%    268KB    268KB    587MB     29.9KB   234891  1375560
```

Fibre Channel Performance

FI 6536 with multiple X9508 chassis connected to MDS at 32G

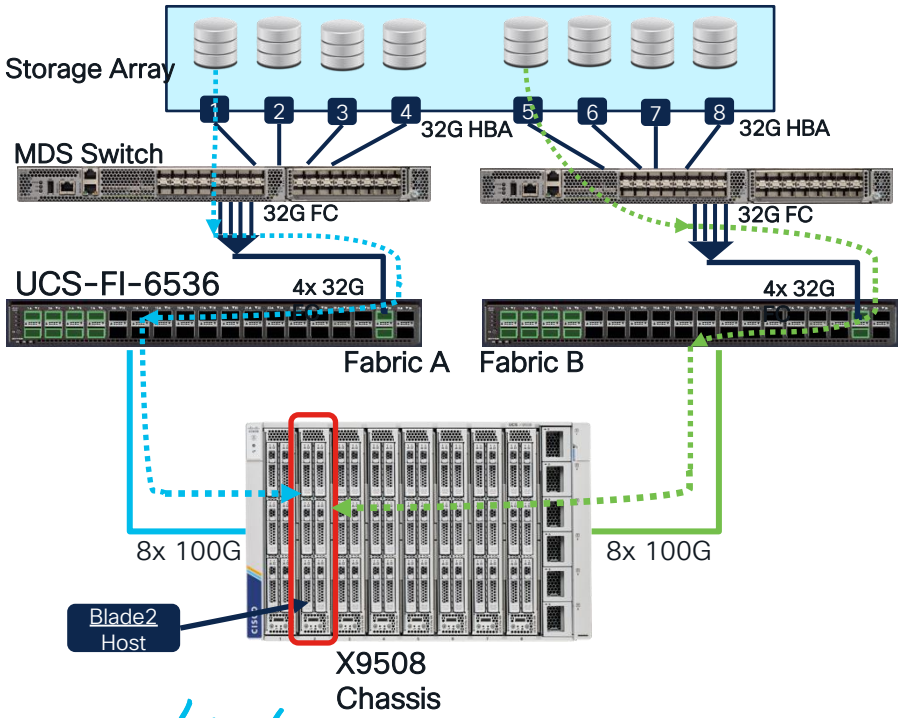
- FC end host mode
- X9108 IFM 100G
- VIC 15230/15231 adapter on every X210c
- RHEL OS 8.4
- vHBA policy: FC/FC-NVMe default
- BIOS policy : default
- OS Tuning : default

Storage array with 32G adapters

End to End 32G : VIC 15230/15231 performance

32G FC connectivity to a Storage Target:

Chassis-Blade2 vHBA1 <-- HBA1 Storage (100% read)
Chassis-Blade2 vHBA1 <-- HBA5 Storage (100% read)



FIO results per x210c

```
[root@ora5g-node1 ~]# cat /sys/class/fc_host/host3/speed
100 Gbit
[root@ora5g-node1 ~]# fio --filename=/dev/mapper/5g_vol1:/dev/mapper/5g_vol2 --direct=1 --rw=r
w --ioengine=libaio --bs=512k --rwmixread=100 --iodepth=8 --numjobs=8 --runtime=120 --group_re
porting --name=seqreadwrite
seqreadwrite: (g=0): rw=rw, bs=(R) 512KiB-512KiB,                    512KiB-512KiB, ioengi
ne=libaio, iodepth=8
...
fio-3.19
Starting 8 processes
Jobs: 8 (f=16): [R(8)][11.6%][r=3141MiB/s]r=6281 IOPS][eta 01m:47s]
```

~32G FC Fabric A

```
[root@ora5g-node1 ~]# cat /sys/class/fc_host/host7/speed
100 Gbit
[root@ora5g-node1 ~]# fio --filename=/dev/mapper/5g_vol9:/dev/mapper/5g_vol10 --direct=1 --rw=
rw --ioengine=libaio --bs=512k --rwmixread=100 --iodepth=8 --numjobs=8 --runtime=120 --group_r
eporting --name=seqreadwrite
seqreadwrite: (g=0): rw=rw, bs=(R) 512KiB-512KiB                    T) 512KiB-512KiB, ioengi
ne=libaio, iodepth=8
...
fio-3.19
Starting 8 processes
Jobs: 8 (f=16): [R(8)][12.4%][r=3142MiB/s]r=6284 IOPS][eta 01m:46s]
```

~32G FC Fabric B

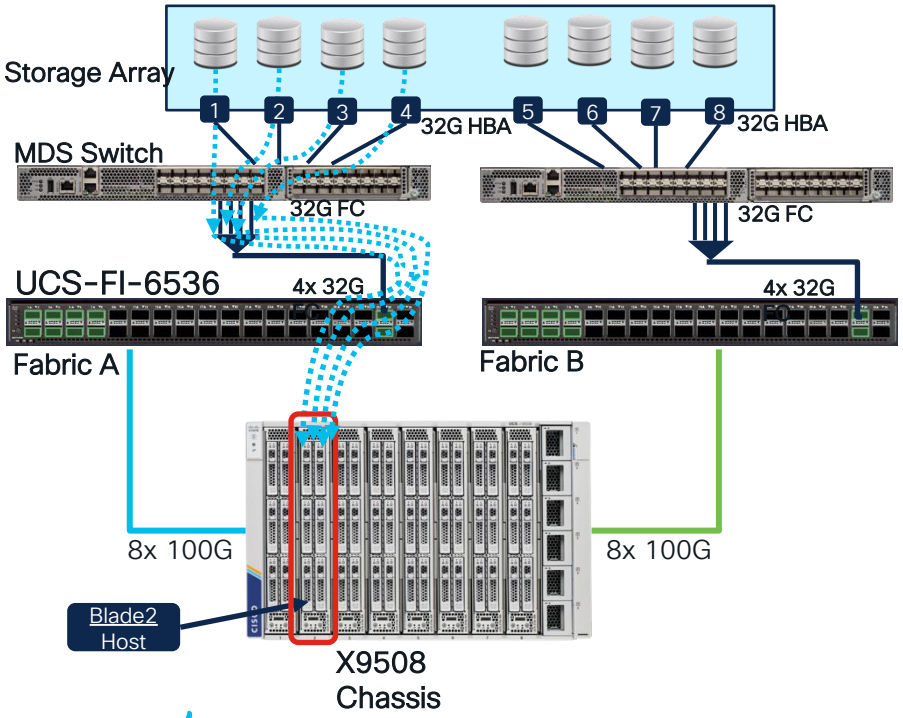
Result from Storage Array

```
FlexPod-A800::>
FlexPod-A800::> qos statistics performance show
Policy Group             IOPS      Throughput     Latency Is Adaptive? Is Shared?
-------------------- -------- --------------- ---------- ------------ ----------
-total-                108565    6342.20MB/s      4.68ms
User-Best-Effort       101469    6342.12MB/s      5.
_System-Work             7096      74.24KB/s      38
-total-                101496    6341.45MB/s      5.0
User-Best-Effort       101455    6341.44MB/s      5.01
_System-Work               41      15.06KB/s     146                false true
-total-                101503    6342.33MB/s      5.00ms - -
User-Best-Effort       101465    6342.31MB/s      5.01ms false true
_System-Work               38      19.36KB/s    105.00us false true
```

Total 64G across Fabric A & B

100G FC per Fabric : VIC 15230/15231 performance

100G vHBA performance in a x210c:
Chassis Blade2 <-- 4x 32G FC storage target (100% read)



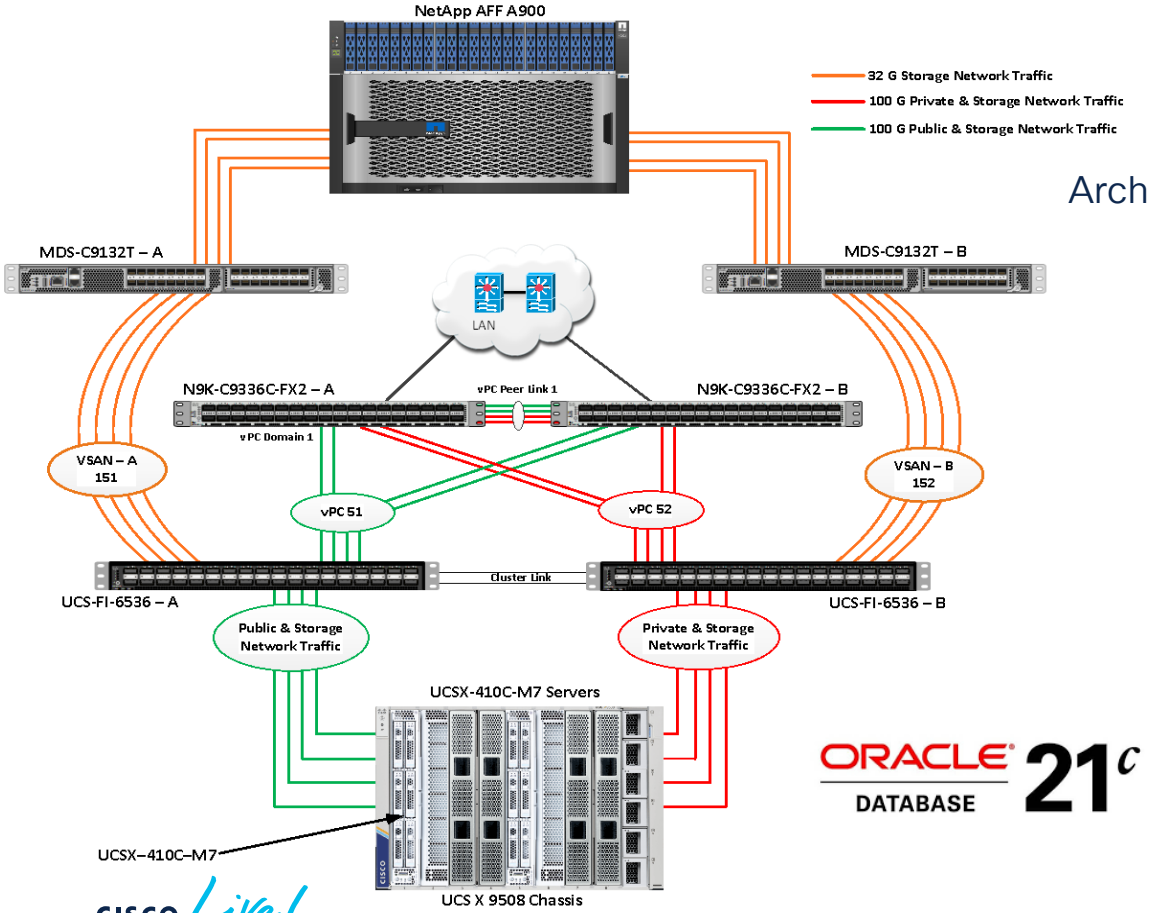
FIO results per x210c

```
[root@ora5g-node1 ~]# fio --filename=/dev/mapper/5g_vol1:/dev/mapper/5g_vol2:/dev/mapper/5g_vol3:/dev/
mapper/5g_vol4:/dev/mapper/5g_vol5:/dev/mapper/5g_vol6:/dev/mapper/5g_vol7:/dev/mapper/5g_vol8 --direc
t=1 --rw=rw --ioengine=libaio --bs=512k --rwmixread=100 --iodepth=32 --numjobs=64 --runtime=30 --group
_reporting --name=seqreadwrite
seqreadwrite: (g=0): rw=rw, bs=(R) 512KiB-512KiB, (W) 512KiB-512KiB, (T) 512KiB-512KiB, ioengine=libai
o, iodepth=32
...
fio-3.19
Starting 64 processes
Jobs: 64 (f=512): [R(64)][64.5% [r=11.1GiB/s] r=22.7k IOPS][eta 00m:11s]
```

```
FlexPod-A800::> q s p s
  (qos statistics performance show)
Policy Group             IOPS      Throughput    Latency Is Adaptive? Is Shared?
------------------- -------- --------------- --------
-total-                183591    11479.40MB/s      4
User-Best-Effort       183529    11479.38MB/s      4
_System-Work               62       22.08KB/s        0ms
-total-                183511    11474.38MB/s      4
User-Best-Effort       183489    11474.38MB/s        ms false true
_System-Work               22          0KB/s     45.00us false true
-total-                183569    11478.01MB/s      5.09ms - -
User-Best-Effort       183529    11478.00MB/s      5.09ms false true
_System-Work               40        1.34KB/s    200.00us false true
```

92G FC (~100G) per Fabric

Oracle 21c RAC Databases w/ End to End 100G over FC-NVMe



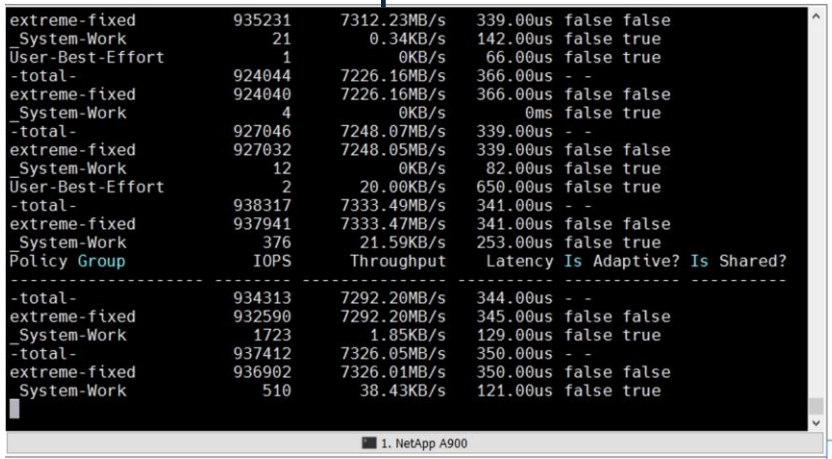
Architecture for Oracle Databases Deployment

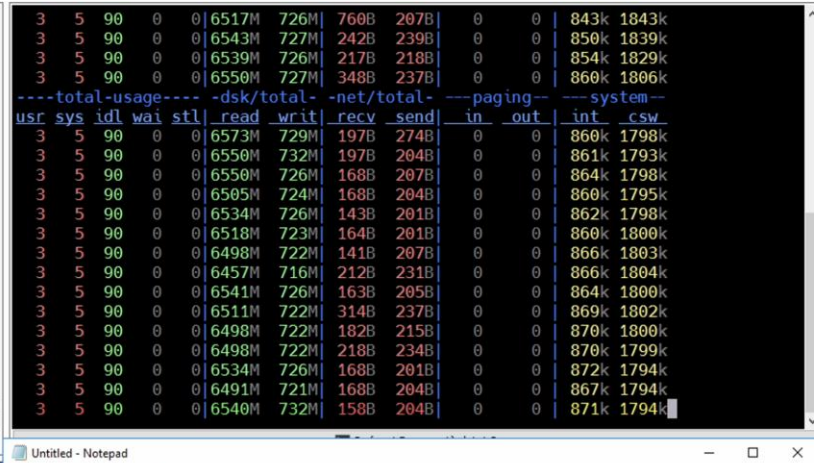
- Cisco UCS 6536 Fabric Interconnect
- Cisco UCS X-Series 9508 Blade Server Chassis
- Cisco UCS X-Series X410C M7 Blade Servers
- Cisco UCS 15231 VIC (Virtual Interface Card)
- NetApp AFF A900 All Flash Array
- Cisco Nexus 9336C-FX2 Switch
- Cisco MDS C9132T FC Switch

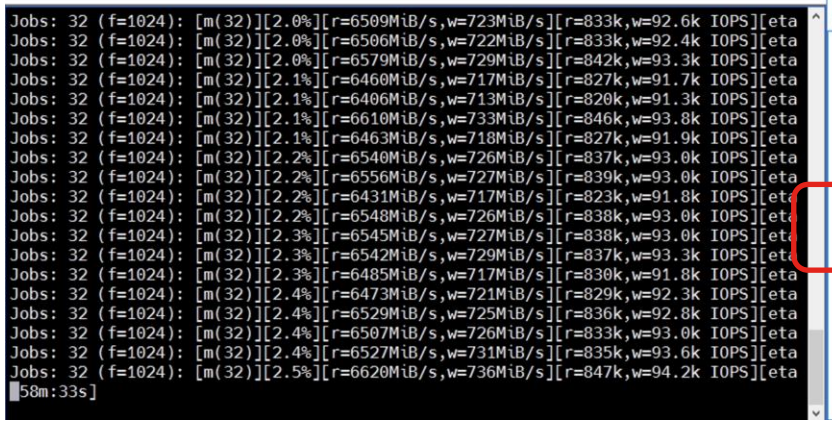
ORACLE DATABASE 21c

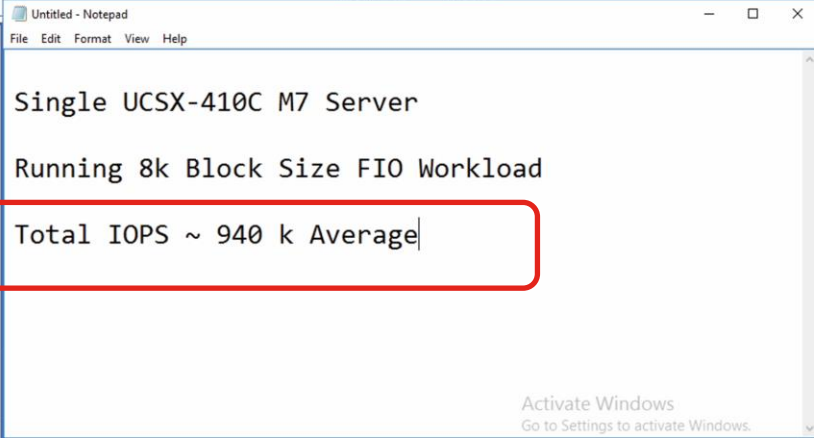
cisco Live!

VIC 15231 FIO performance over FC-NVMe with 8K blocks

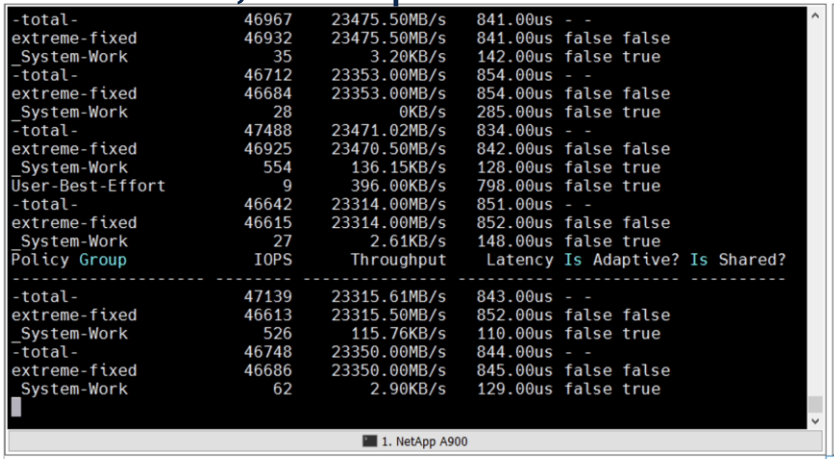


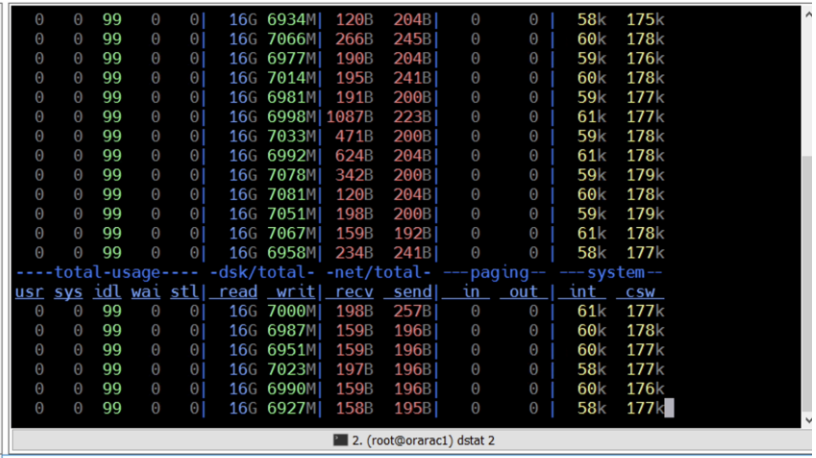


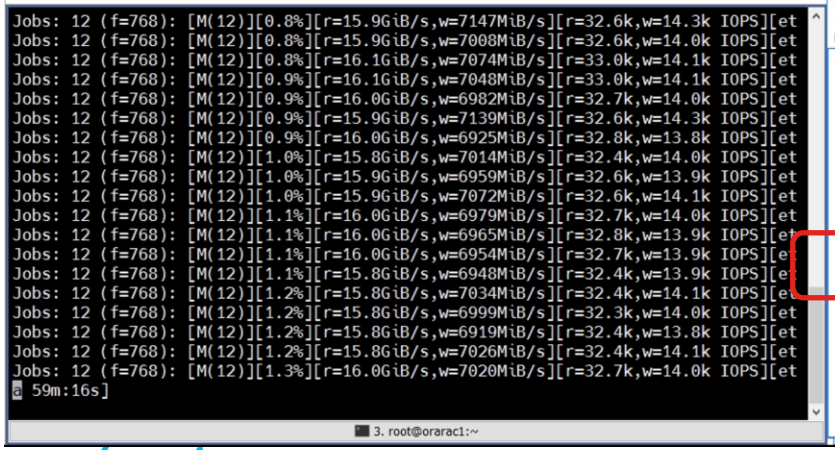


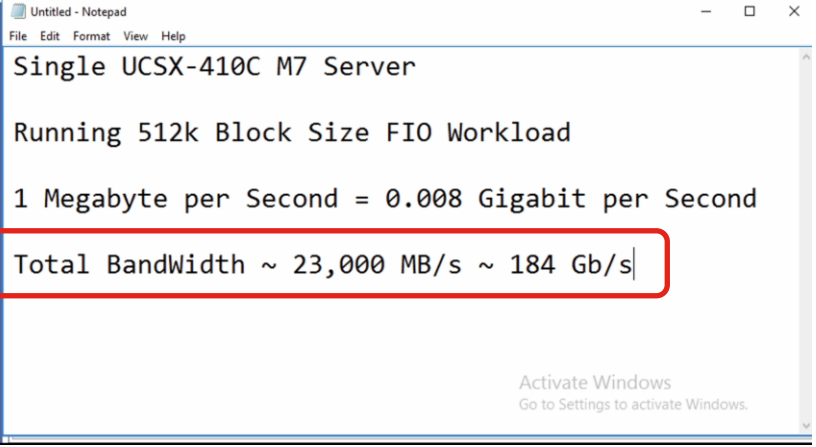


VIC 15231, FIO performance over FC-NVMe with 512K blocks









UCS 100G Fabric: CVD Solution References

FlexPod Datacenter with vSphere – https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_imm_m7_iac.html

FlexPod Datacenter with Oracle RAC – https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_oracle_xseries_m7.html

FlashStack VSI with vSphere – https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flashstack_m7_vmware_8_ufs_fc.html

Conclusion

Differentiation

- VIC is purpose built for UCS servers
- Differentiated features like dynamic I/O virtualization, fabric failover, policy-driven configuration, standby-power, single-wire management, VIC QinQ Tunneling
- Single architecture for various applications and workloads.
- Cisco VIC with the UCS Fabric Interconnect or Nexus 9000 and UCS management

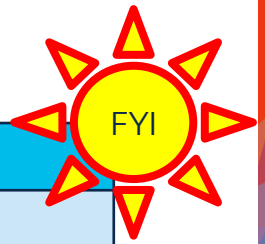
Performance

- 200Gbps throughput per interface card
- 100G end-to-end for LAN/SAN/NFS
- Multiple FC and Ethernet interfaces on one adapter with one set of cables
- Expand throughput with additional VICs
- Comprehensive hardware offload and acceleration feature-set
- Built with end-to-end network QoS

Management

- Centralized management built into the platform since day one.
- Scale out, maintain, monitor, troubleshoot in one management frame
- Single-connect with management integrated – no separate OOB network
- Surprising savings vs disparate LAN/SAN/OOB, cabling/switching infrastructure

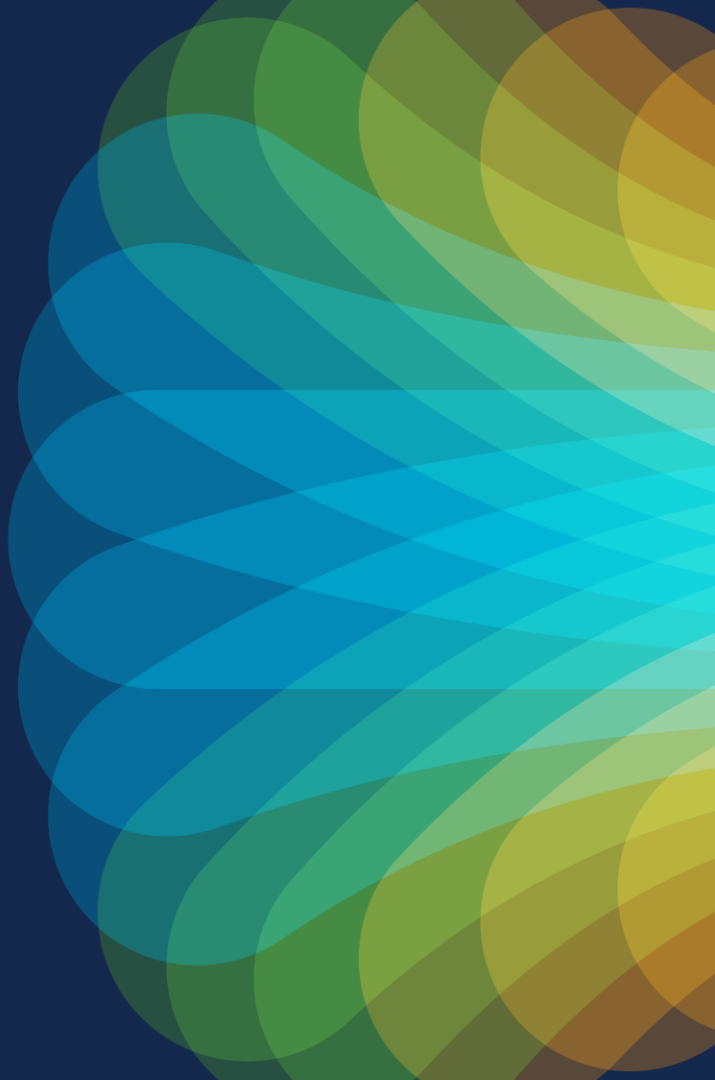
Recommended sessions



Time/Date *(TBC)*	Session Group	Code	Speaker	Title
Tuesday, Feb 6 1:15-2:15pm	BRKDCN-1669	Breakout	Brian Everitt	Cisco HCI Solution 101 in 60 minutes
Tuesday Feb 6 8-9am	BRKDCN-2649	Breakout	Rodrigo Morelli	The Future of Compute Monitoring and Observability with Intersight
Tuesday, Feb 6 9:30-10:30am	BRKDCN-2669	Breakout	Eldho Jacob	Cisco UCS 5th Generation Virtual Interface Card(VIC) – Unveiling the latest Cutting-Edge Feature Set and Performance Advancements
Wednesday Feb 7 10:30-11:30am	BRKDCN-2679	Breakout	Riccardo Tortorici	IMM and X-Series automation-driven operations: The definitive (ish) guide
Wednesday Feb 7 2:30-3:30pm		Breakout	Intel & Sean McGee	Power the Future of AI with Intel and Cisco
Friday Feb 9 11:15-am-12:45pm	BRKDCN-2933	Breakout	Scott Garee	Applying UCS X-Series advanced power and cooling management to your sustainability goals and reporting
Friday, Feb 9 9:15 - 10:45 AM	BRKDCN-2938	Breakout	Vincent Esposito	Intersight Managed Mode (IMM) for UCS Manager admins
Wednesday Feb 7 8:30-10am	BRKCLD-1603	Breakout	Michael Maurer	How to build a Time Machine
Friday Feb 9 11:15-am-12:45pm	BRKCLD-2015	Breakout	Riccardo Tortorici	Multi-Domain Orchestration is Just a Click Away with Cisco Intersight
Wednesday, Feb 7 2-3pm	IBOCOM-2301	Interactive Breakout	Matthew Faiello	An Interactive Conversation on IMM Trasition: How Do I Get There?
Tuesday Feb 6 3:30-4:15pm	DEVWKS-2170	DevNet	Riccardo Tortorici	Building infrastructure services with Intersight Orchestration
Wednesday Feb 7 4:30-5:15pm	DEVNET - 1606	DevNet	Michael Maurer, Martin Divis	What Happened in my Data Center last night?
N/A	LABDCN-1436	Walk-in Lab	Michael Maurer, Mohit Vaswani	Travel through time with Intersight Monitoring

The bridge to possible

Thank you



cisco Live!


cisco Live!
Let's go