NSX Troubleshooting Guide

Update 2

Modified on 18 AUG 2016

VMware NSX Data Center for vSphere 6.2

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

1 NSX Troubleshooting Guide 4

2 Infrastructure Preparation 5

NSX Infrastructure Preparation Steps 7

Checking Communication Channel Health 20

Troubleshooting NSX Manager Issues 21

Recover from an NSX Controller Failure 23

Using the NSX Dashboard 25

Using the show host health-status Command 27

Setting the Logging Level of NSX Components 27

vSphere ESX Agent Manager 29

NSX CLI Cheat Sheet 31

3 Traceflow 41

About Traceflow 41

Use Traceflow for Troubleshooting 43

4 NSX Routing 51

Understanding the Distributed Logical Router 52

Understanding Routing Provided by the Edge Services Gateway 56

ECMP Packet Flow 56

NSX Routing: Prerequisites and Considerations 58

DLR and ESG UIs 61

New NSX Edge (DLR) 63

Typical ESG and DLR UI Operations 67

Troubleshooting NSX Routing 71

5 Edge Appliance Troubleshooting 102

6 Distributed Firewall 116

How to Use the show dfw CLI 116

Troubleshooting Distributed Firewall 118

7 Load Balancing 125

Scenario: Configure a One-Armed Load Balancer 126

Load Balancer Troubleshooting Using the UI 130

Load Balancer Troubleshooting Using the CLI 131

Common Load Balancer Issues 134

NSX Troubleshooting Guide

1

The *NSX Troubleshooting Guide* describes how to monitor and troubleshoot the VMware® NSX™ system by using the NSX Manager user interface, the vSphere Web Client, and other NSX components, as needed.

Intended Audience

This manual is intended for anyone who wants to install or use NSX in a VMware vCenter environment. The information in this manual is written for experienced system administrators who are familiar with virtual machine technology and virtual datacenter operations. This manual assumes familiarity with VMware Infrastructure 5.x, including VMware ESX, vCenter Server, and the vSphere Web Client.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to http://www.vmware.com/support/pubs.

Infrastructure Preparation 2

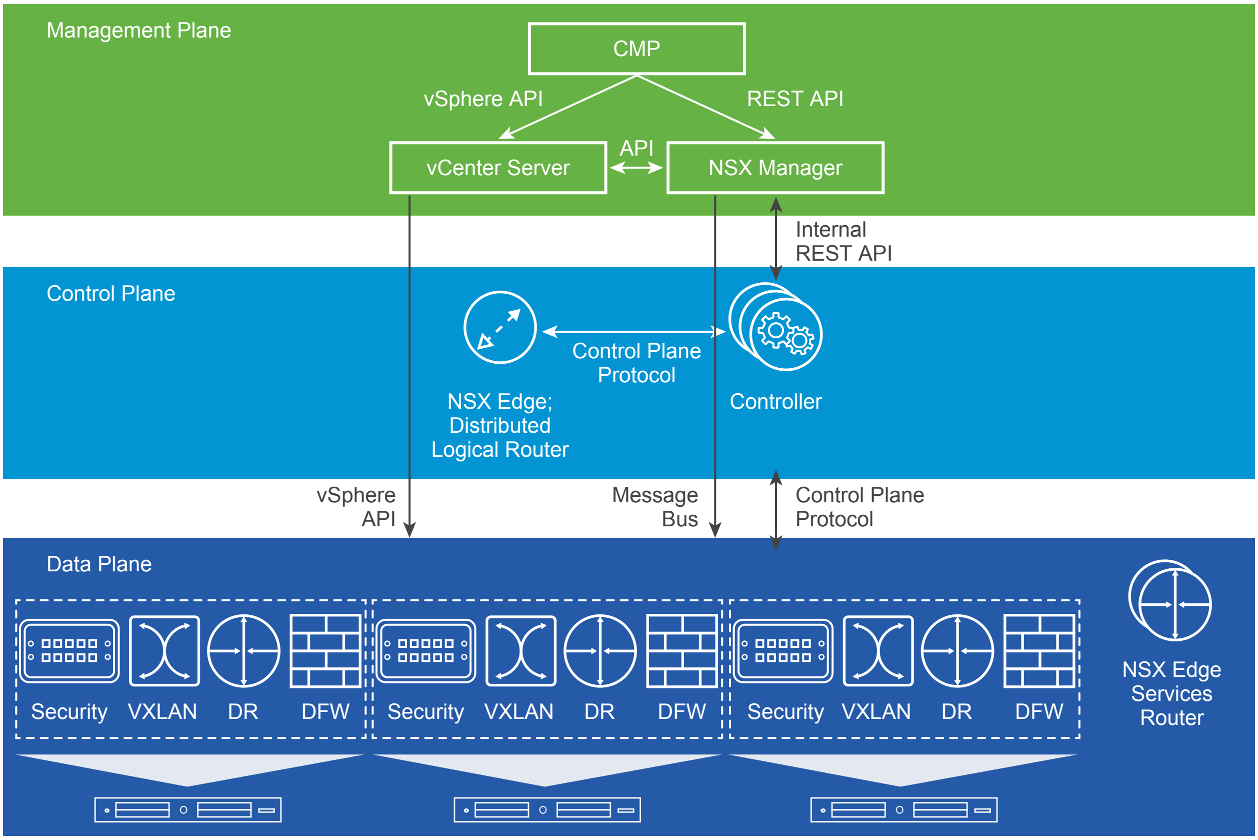
Knowledge of the components used in the preparation of NSX is important for identifying and resolving common issues.

NSX Infrastructure Preparation Components

- vSphere ESX Agent Manager (EAM)
- NSX Manager
- Controller cluster (if using unicast, hybrid mode or distributed logical routing)
- VTEP (ESXi hypervisor)
- User world agents (UWA)
- vSphere distributed switch

The control-plane communication between NSX Manager and ESXi hypervisor hosts is provided by a RabbitMQ-based messaging service. The control-plane communication between the Controller cluster and ESXi hypervisor hosts depends on a netcpa userworld agent that runs on hosts as a client.

Figure 2‑1. High-Level View of Components and Their Communication



Tips for Successful Infrastructure Preparation

The VMware Product Interoperability Matrixes website provides information about NSX compatibility and version requirements. See http://partnerweb.vmware.com/comp_guide2/sim/interop_matrix.php.

Make sure you are using vSphere distributed switch release 5.5 or later.

Use separate DVSs for management, services, and gateway.

Be mindful of which NIC teaming method you use when using blades. The chassis switch supports the minimum set.

When installing, make sure that the Controller Cluster is deployed and in the all-green state before proceeding to host preparation.

When upgrading, make sure that the Controllers are connected and in the all-green state before upgrading. See the *NSX Upgrade Guide*.

This chapter includes the following topics:

- NSX Infrastructure Preparation Steps
- Checking Communication Channel Health
- Troubleshooting NSX Manager Issues
- Recover from an NSX Controller Failure

- Using the NSX Dashboard
- Using the show host health-status Command
- Setting the Logging Level of NSX Components
- vSphere ESX Agent Manager
- NSX CLI Cheat Sheet

NSX Infrastructure Preparation Steps

NSX preparation is a 4-step process.

1. Connect NSX Manager to vCenter Server. There is a one-to-one relationship between NSX Manager and vCenter Server.

   a. Register with vCenter Server

2. Deploy NSX Controllers (Only required for logical switching, distributed routing, or edge service. If you are only using distributed firewallm(DFW), controllers are not required).

3. Host Preparation: Installs VIBs for VXLAN, DFW, and DLR on all hosts in the cluster. Configures the Rabbit MQ-based messaging infrastructure. Enables firewall. Notifies controllers that hosts are ready for NSX.

4. Configure IP pool settings and configure VXLAN: Creates a VTEP port group and VMKNICs on all hosts in the cluster. During this step, you can set the transport VLAN ID, teaming policy, and MTU.

Connecting NSX Manager to vCenter Server

A connection between the NSX Manager and the vCenter Server allows NSX Manager to use the vSphere API to perform functions such as deploy service VMs, prepare hosts, and create logical switch portgroups. The connection process installs a web client plug-in for NSX on the Web Client Server.

For the connection to work, you must have DNS and NTP configured on NSX Manager, vCenter Server and the ESXi hosts. If you added ESXi hosts by name to the vSphere inventory, ensure that DNS servers have been configured on the NSX Manager and name resolution is working. Otherwise, NSX Manager cannot resolve the IP addresses. The NTP server must be specified so that the SSO server time and NSX Manager time are in sync. On NSX Manager, the drift file at `/etc/ntp.drift` is included in the tech Support bundle for NSX Manager.

Also the account you use to connect NSX Manager to vCenter Server must have the vCenter role "Administrator." Having the "Administrator" role also enables NSX Manager to register itself with the Security Token Service server. When a particular user account is used to connect NSX Manager to vCenter, an “Enterprise Administrator" role for the user is also created on NSX Manager.

Common Issues Related to Connecting NSX Manager to vCenter Server

- DNS incorrectly configured on NSX Manager, vCenter Server, or an ESXi host.
- NTP incorrectly configured on NSX Manager, vCenter Server, or an ESXi host.

- User account without vCenter role of Administrator used to connect NSX Manager to vCenter.
- Network connectivity issues between NSX Manager and vCenter server.
- User logging into vCenter with an account that does not have a role on NSX Manager.

You need to initially log into vCenter with the account you used to link NSX Manager to vCenter Server. Then you can create additional users with roles on NSX Manager using the **vCenter Home > Networking & Security > NSX Managers > {IP of NSX Manager} > Manage > Users** API.

The first login can take up to 4 minutes while vCenter loads and deploys NSX UI bundles.

Verify Connectivity from NSX Manager to vCenter Server

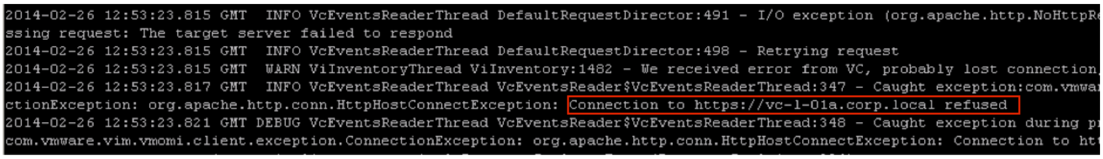
To verify connectivity, ping from the NSX virtual appliance and view the ARP and routing tables.

```
nsxmgr# show arp
IP address       HW type     Flags     HW address            Mask     Device
192.168.110.31   0x1         0x2       00:50:56:ae:ab:01     *        mgmt
192.168.110.2    0x1         0x2       00:50:56:01:20:a5     *        mgmt
192.168.110.1    0x1         0x2       00:50:56:01:20:a5     *        mgmt
192.168.110.33   0x1         0x2       00:50:56:ae:4f:7c     *        mgmt
192.168.110.32   0x1         0x2       00:50:56:ae:50:bf     *        mgmt
192.168.110.10   0x1         0x2       00:50:56:03:19:4e     *        mgmt
192.168.110.51   0x1         0x2       00:50:56:03:30:2a     *        mgmt
192.168.110.22   0x1         0x2       00:50:56:01:21:f9     *        mgmt
192.168.110.55   0x1         0x2       00:50:56:01:23:21     *        mgmt
192.168.110.26   0x1         0x2       00:50:56:01:21:ef     *        mgmt
192.168.110.54   0x1         0x2       00:50:56:01:22:ef     *        mgmt
192.168.110.52   0x1         0x2       00:50:56:03:30:16     *        mgmt
```

```
nsxmgr# show ip route
Codes: K - kernel route, C - connected, S - static,
       > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 192.168.110.1, mgmt
C>* 192.168.110.0/24 is directly connected, mgmt
```

Look for errors in the NSX Manager log to indicate the reason for not connecting to vCenter Server. The command to view the log is `show log manager follow`.



```
2014-02-26 12:53:23.815 GMT  INFO VcEventsReaderThread DefaultRequestDirector:491 - I/O exception (org.apache.http.NoHttpRe
ssing request: The target server failed to respond
2014-02-26 12:53:23.815 GMT  INFO VcEventsReaderThread DefaultRequestDirector:498 - Retrying request
2014-02-26 12:53:23.815 GMT  WARN ViInventoryThread ViInventory:1482 - We received error from VC, probably lost connection
2014-02-26 12:53:23.817 GMT  INFO VcEventsReaderThread VcEventsReader$VcEventsReaderThread:347 - Caught exception:com.vmwa
ctionException: org.apache.http.conn.HttpHostConnectException: Connection to https://vc-l-01a.corp.local refused
2014-02-26 12:53:23.821 GMT DEBUG VcEventsReaderThread VcEventsReader$VcEventsReaderThread:348 - Caught exception during p
com.vmware.vim.vmomi.client.exception.ConnectionException: org.apache.http.conn.HttpHostConnectException: Connection to ht
```

Log in to the NSX Manager CLI console, run the command: `debug connection IP_of_ESXi_or_VC`, and examine the output.

Perform Packet Capture on NSX Manager to View Connections

Use the debug packet command: `debug packet [capture|display] interface interface filter`

The interface name on NSX Manager is `mgmt`.

The filter syntax follows this form: "port_80_or_port_443"

The command runs in privileged mode only. To enter privileged mode, run the `enable` command and provide the admin password.

Packet capture example:

```
nsxmgr# en
nsxmgr# debug packet display interface mgmt port_80_or_port_443
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on mgmt, link-type EN10MB (Ethernet), capture size 262144 bytes
23:40:25.321085 IP 192.168.210.15.54688 > 192.168.210.22.443: Flags [P.], seq 2645022162:2645022199,
ack 2668322748, win 244, options [nop,nop,TS val 1447550948 ecr 365097421], length 37
...
```

Verify Network Configuration on NSX Manager

The `show running-config` command shows the basic configuration of the management interface, NTP, and default route settings.

```
nsxmgr# show running-config
Building configuration...

Current configuration:
!
ntp server 192.168.110.1
!
ip name server 192.168.110.10
!
hostname nsxmgr
!
interface mgmt
 ip address 192.168.110.15/24
!
ip route 0.0.0.0/0 192.168.110.1
!
web-manager
```

NSX Manager Certificates

NSX Manager supports two ways to generate certificates.

- NSX Manager generated CSR: Limited functionality due to basic CSR
- PKCS#12: This is recommended for production

There is a known issue in which the CMS silently fails to make API calls.

This happens when the certificate issuer is not known to the caller because it is an untrusted root certificate authority or the certificate is self-signed. To resolve this issue, use a browser to navigate to the NSX Manager IP address or hostname and accept the certificate.

Deploying NSX Controllers

NSX Controllers are deployed by NSX Manager in OVA format. Having a Controller cluster provides high availability.

Deploying Controllers requires that NSX Manager, vCenter Server, and ESXi hosts have DNS and NTP configured.

A static IP pool must be used to assign IP addresses to each Controller.

It is recommended that you implement DRS anti-affinity rules to keep NSX Controllers on separate hosts.

You must deploy three NSX Controllers.

Common Issues with Controllers

During the deployment of NSX Controllers, the typical issues that can be encountered are as follows:

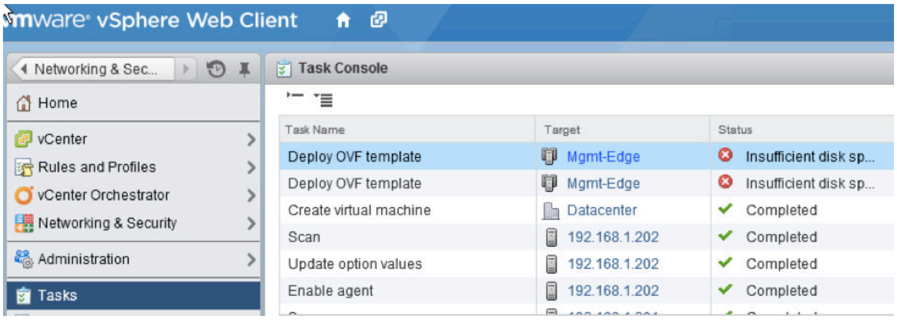
- NSX Controller running slowly. This might be caused by insufficient resources. To detect issues with NSX Controller system requirements, run the `request system compatibility-report` command.

```
nsx-controller # request system compatibility-report
Testing: Number of CPUs. Done.
Testing: Aggregate CPU speed. Done.
Testing: Memory. Done.
Testing: Management NIC speed. Done.
Testing: NTP configured. Done.
Testing: /var disk partition size. Done.
Testing: /var disk speed. Done.
Testing: pserver-log disk size. Done.
Testing: pserver-log disk speed. Done.
Testing: pserver-data disk size. Done.
Testing: pserver-data disk speed. Done.
Testing: logging disk size. Done.
Testing: logging disk speed. Done.

                                Detected      Supported       Required
Number of CPUs                         2             NO            >=8
Aggregate CPU speed              5.6 GHz             NO           >=13
Memory                          1.835 GB             NO           >=63
Management NIC speed          10000 Mb/s            YES         >=1000
NTP configured                        No             NO            Yes
/var disk partition size            - GB             NO          >=128
/var disk speed                   - MB/s             NO           >=40
pserver-log disk size               - GB             NO          >=128
pserver-log disk speed            - MB/s             NO           >=40
pserver-data disk size              - GB             NO          >=128
pserver-data disk speed           - MB/s             NO           >=40
logging disk size                   - GB             NO          >=128
logging disk speed                - MB/s             NO           >=40
```

- IP connectivity issues between the NSX Manager and the NSX controllers. This is generally caused by physical network connectivity issues or a firewall blocking communication.

- Insufficient resources such as storage available on vSphere to host the Controllers. Viewing the vCenter events and tasks log during Controller deployment can identify such issues.

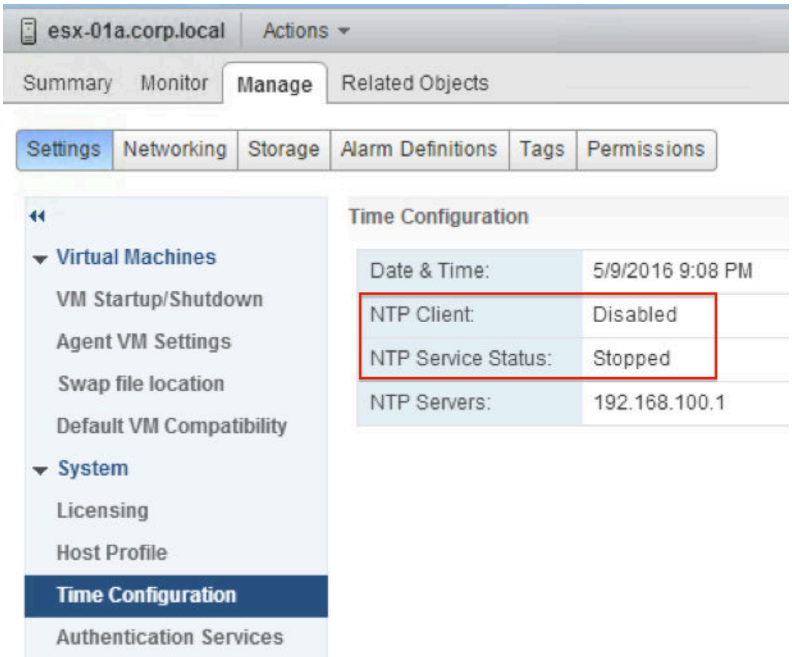


Task Name	Target	Status
Deploy OVF template	Mgmt-Edge	Insufficient disk sp...
Deploy OVF template	Mgmt-Edge	Insufficient disk sp...
Create virtual machine	Datacenter	✓ Completed
Scan	192.168.1.202	✓ Completed
Update option values	192.168.1.202	✓ Completed
Enable agent	192.168.1.202	✓ Completed

- A misbehaving "rogue" Controller or an upgraded Controllers in the Disconnected state.

- DNS on ESXi hosts and NSX manager have not been configured properly.

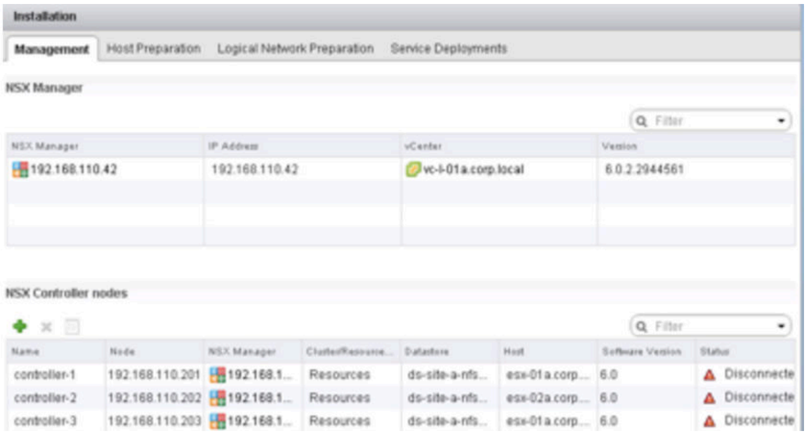
- NTP on ESXi hosts and NSX Manager are not in sync.



Time Configuration

Date & Time:	5/9/2016 9:08 PM
NTP Client:	Disabled
NTP Service Status:	Stopped
NTP Servers:	192.168.100.1

- When newly connected VMs have no network access, this is likely caused by a control-plane issue. Check the Controller status.



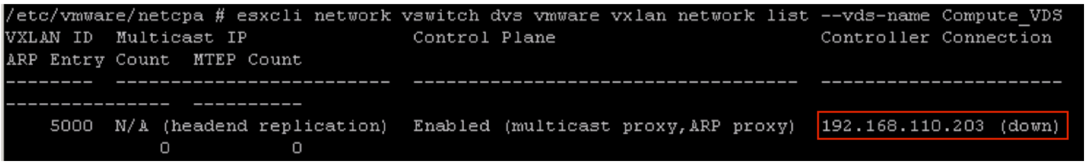
NSX Manager

NSX Manager	IP Address	vCenter	Version
192.168.110.42	192.168.110.42	vc-l-01a.corp.local	6.0.2.2944561

NSX Controller nodes

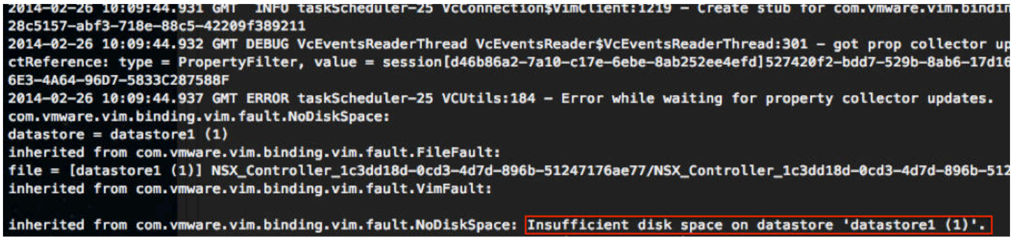
Name	Node	NSX Manager	Cluster/Resource...	Datastore	Host	Software Version	Status
controller-1	192.168.110.201	192.168.1...	Resources	ds-site-a-nfs...	esx-01a.corp...	6.0	Disconnecte
controller-2	192.168.110.202	192.168.1...	Resources	ds-site-a-nfs...	esx-02a.corp...	6.0	Disconnecte
controller-3	192.168.110.203	192.168.1...	Resources	ds-site-a-nfs...	esx-01a.corp...	6.0	Disconnecte

Also try running the `esxcli network vswitch dvs vmware vxlan network list --vds-name <name>` command on ESXi hosts to check the control-plane status. Note that the Controller connection is down.



```
/etc/vmware/netcpa # esxcli network vswitch dvs vmware vxlan network list --vds-name Compute_VDS
VXLAN ID  Multicast IP               Control Plane                        Controller Connection
ARP Entry Count  MTEP Count
--------  -------------------------  -----------------------------------  ----------------------
---------------  ----------
    5000  N/A (headend replication)  Enabled (multicast proxy,ARP proxy)  192.168.110.203 (down)
              0           0
```

- Running the `show log manager follow` NSX Manager CLI command can identify any other reasons for a failure to deploy controllers.



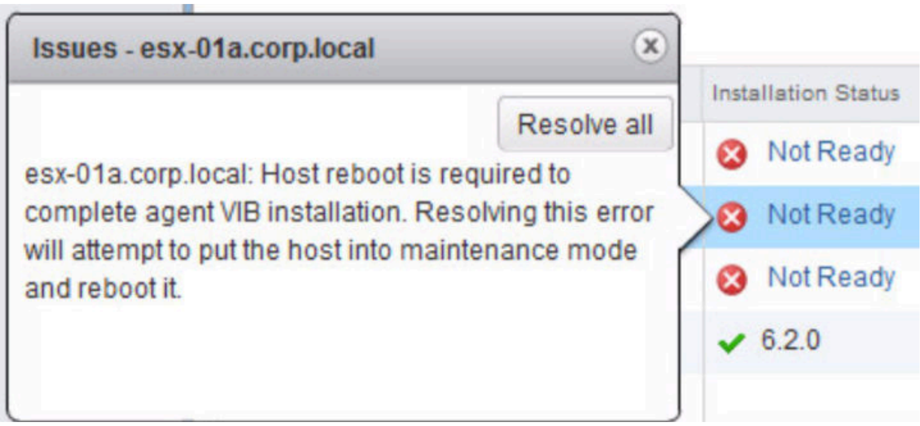
```
2014-02-26 10:09:44.931 GMT  INFO taskScheduler-25 VcConnection$VimClient:1219 - Create stub for com.vmware.vim.bindin
28c5157-abf3-718e-88c5-42209f389211
2014-02-26 10:09:44.932 GMT DEBUG VcEventsReaderThread VcEventsReader$VcEventsReaderThread:301 - got prop collector up
ctReference: type = PropertyFilter, value = session[d46b86a2-7a10-c17e-6ebe-8ab252ee4efd]527420f2-bdd7-529b-8ab6-17d16
6E3-4A64-96D7-5833C287588F
2014-02-26 10:09:44.937 GMT ERROR taskScheduler-25 VCUtils:184 - Error while waiting for property collector updates.
com.vmware.vim.binding.vim.fault.NoDiskSpace:
datastore = datastore1 (1)
inherited from com.vmware.vim.binding.vim.fault.FileFault:
file = [datastore1 (1)] NSX_Controller_1c3dd18d-0cd3-4d7d-896b-51247176ae77/NSX_Controller_1c3dd18d-0cd3-4d7d-896b-512
inherited from com.vmware.vim.binding.vim.fault.VimFault:

inherited from com.vmware.vim.binding.vim.fault.NoDiskSpace: Insufficient disk space on datastore 'datastore1 (1)'.
```

Host Preparation

vSphere ESX Agent Manager deploys VIBs onto ESXi hosts.

The deployment on hosts requires that DNS be configured on the hosts, vCenter Server, and NSX Manager. Deployment does not require an ESXi host reboot, but any update or removal of VIBs requires an ESXi host reboot.



Issues - esx-01a.corp.local

Resolve all

esx-01a.corp.local: Host reboot is required to complete agent VIB installation. Resolving this error will attempt to put the host into maintenance mode and reboot it.

VIBs are hosted on NSX Manager and are also available as a zip file.

The file can be accessed from `https://<NSX-Manager-IP>/bin/vdn/nwfabric.properties`. The downloadable zip file differs based on NSX and ESXi version. For example, vSphere 6.0 hosts use the file `https://<NSX-Manager-IP>/bin/vdn/vibs-6.2.3/6.0-3771165/vxlan.zip`.

```
C:\Users\Administrator>curl -k https://nsxmgr-01a.corp.local/bin/vdn/nwfabric.properties
# 5.1 VDN EAM Info
VDN_VIB_PATH.1=/bin/vdn/vibs-6.2.3/5.1-2107743/vxlan.zip
VDN_VIB_VERSION.1=2107743
VDN_HOST_PRODUCT_LINE.1=embeddedEsx
VDN_HOST_VERSION.1=5.1.*

# 5.5 VDN EAM Info
VDN_VIB_PATH.2=/bin/vdn/vibs-6.2.3/5.5-3771174/vxlan.zip
VDN_VIB_VERSION.2=3771174
VDN_HOST_PRODUCT_LINE.2=embeddedEsx
VDN_HOST_VERSION.2=5.5.*

# 6.0 VDN EAM Info
VDN_VIB_PATH.3=/bin/vdn/vibs-6.2.3/6.0-3771165/vxlan.zip
VDN_VIB_VERSION.3=3771165
VDN_HOST_PRODUCT_LINE.3=embeddedEsx
VDN_HOST_VERSION.3=6.0.*

# 6.1 VDN EAM Info
VDN_VIB_PATH.4=/bin/vdn/vibs-6.2.3/6.1-3689890/vxlan.zip
VDN_VIB_VERSION.4=3689890
VDN_HOST_PRODUCT_LINE.4=embeddedEsx
VDN_HOST_VERSION.4=6.1.*

# Single Version associated with all the VIBs pointed by above VDN_VIB_PATH(s)
VDN_VIB_VERSION=6.2.3.3771501

# Legacy vib location. Used by code to discover avaialble legacy vibs.
LEGACY_VDN_VIB_PATH_FS=/common/em/components/vdn/vibs/legacy/
LEGACY_VDN_VIB_PATH_WEB_ROOT=/bin/vdn/vibs/legacy/
```

The VIB names are:

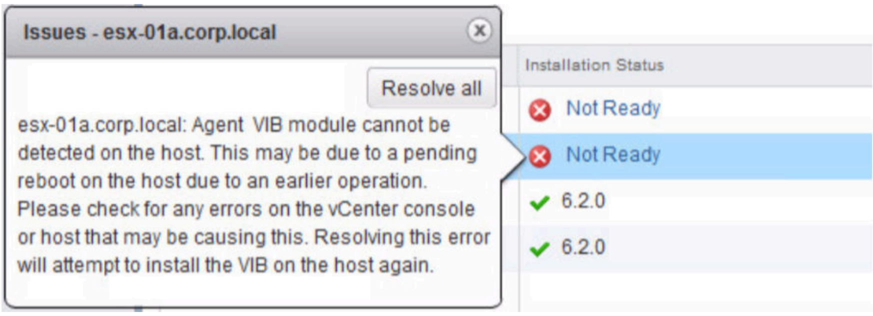
- esx-vsip
- esx-vxlan

```
[root@esx-01a:~] esxcli software vib list | grep -e vsip -e vxlan
esx-vsip                      6.0.0-0.0.3771165                    VMware  VMwareCertified
2016-04-20
esx-vxlan                     6.0.0-0.0.3771165                    VMware  VMwareCertified
2016-04-20
```

Common Issues During Host Preparation

During the preparation of hosts typical kinds of issues that can be encountered are as follows:

- EAM fails to deploy VIBs.
  - Might be due to misconfigured DNS on hosts.



esx-01a.corp.local: Agent VIB module cannot be detected on the host. This may be due to a pending reboot on the host due to an earlier operation. Please check for any errors on the vCenter console or host that may be causing this. Resolving this error will attempt to install the VIB on the host again.

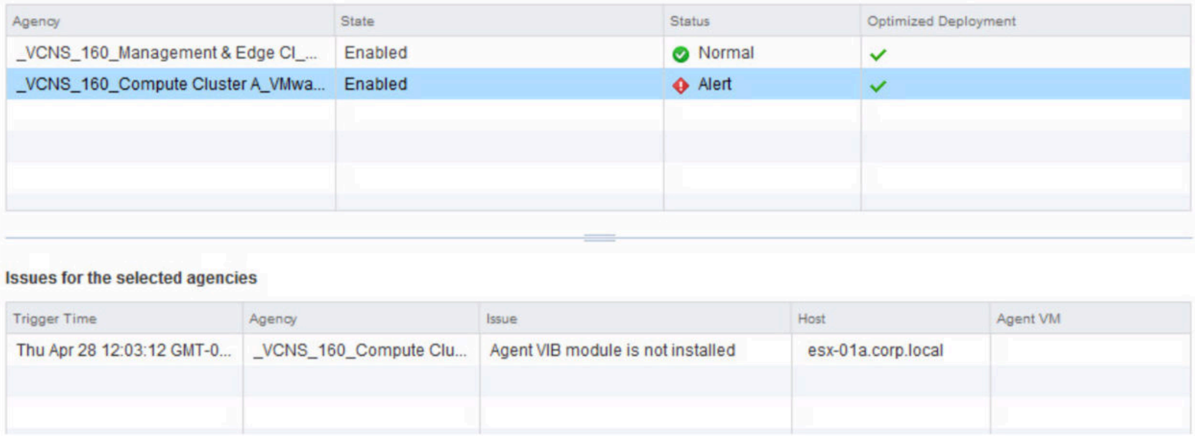
  - Might be due to a firewall blocking required ports between ESXi, NSX Manager, and vCenter Server.
- A previous VIB of an older version is already installed. This requires user intervention to reboot hosts.
- NSX Manager and vCenter Server experience communication issues:
  - The **Host Preparation** tab in the Networking and Security Plug-in not showing all hosts properly.
  - Check if vCenter Server can enumerate all hosts and clusters.

Host Preparation (VIBs) Troubleshooting

- Check communication channel health for the host. See Checking Communication Channel Health.
- Check vSphere ESX Agent Manager for errors.

vCenter home > Administration > vCenter Server Extensions > vSphere ESX Agent Manager

On vSphere ESX Agent Manager, check the status of agencies that are prefixed with “VCNS160”. If an agency has a bad status, select the agency and view its issues.

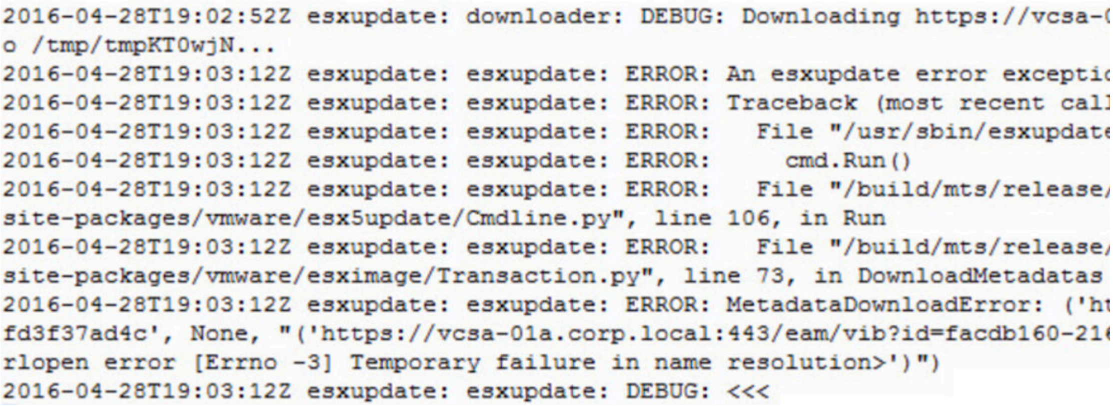


Agency	State	Status	Optimized Deployment
_VCNS_160_Management & Edge Cl_...	Enabled	✓ Normal	✓
_VCNS_160_Compute Cluster A_VMwa...	Enabled	◆ Alert	✓

Issues for the selected agencies

Trigger Time	Agency	Issue	Host	Agent VM
Thu Apr 28 12:03:12 GMT-0...	_VCNS_160_Compute Clu...	Agent VIB module is not installed	esx-01a.corp.local	

- On the host that is having an issue, run the `tail /var/log/esxupdate.log` command.



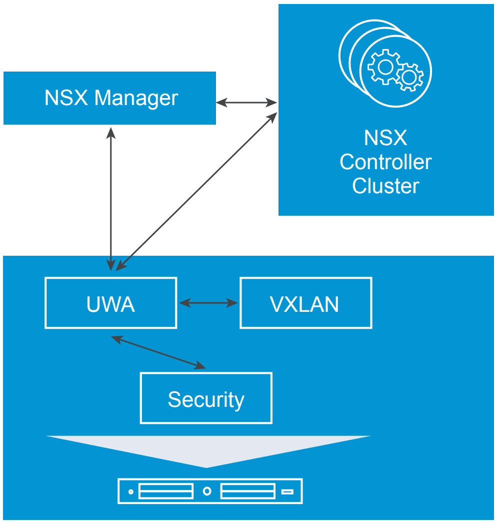
```
2016-04-28T19:02:52Z esxupdate: downloader: DEBUG: Downloading https://vcsa-(
o /tmp/tmpKT0wjN...
2016-04-28T19:03:12Z esxupdate: esxupdate: ERROR: An esxupdate error exceptic
2016-04-28T19:03:12Z esxupdate: esxupdate: ERROR: Traceback (most recent call
2016-04-28T19:03:12Z esxupdate: esxupdate: ERROR:   File "/usr/sbin/esxupdate
2016-04-28T19:03:12Z esxupdate: esxupdate: ERROR:     cmd.Run()
2016-04-28T19:03:12Z esxupdate: esxupdate: ERROR:   File "/build/mts/release/
site-packages/vmware/esx5update/Cmdline.py", line 106, in Run
2016-04-28T19:03:12Z esxupdate: esxupdate: ERROR:   File "/build/mts/release/
site-packages/vmware/esximage/Transaction.py", line 73, in DownloadMetadatas
2016-04-28T19:03:12Z esxupdate: esxupdate: ERROR: MetadataDownloadError: ('ht
fd3f37ad4c', None, "('https://vcsa-01a.corp.local:443/eam/vib?id=facdb160-21(
rlopen error [Errno -3] Temporary failure in name resolution>')")
2016-04-28T19:03:12Z esxupdate: esxupdate: DEBUG: <<<
```

- See https://kb.vmware.com/kb/2053782.

Host Preparation (UWA) Troubleshooting

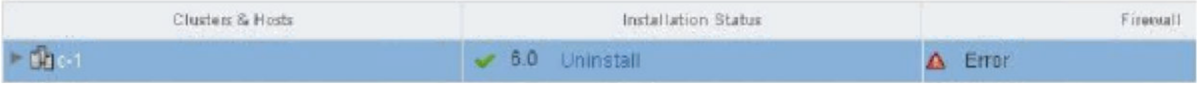
NSX Manager configures two user world agents on all hosts in a cluster:

- Messaging bus UWA (vsfwd)
- Control plane UWA (netcpa)



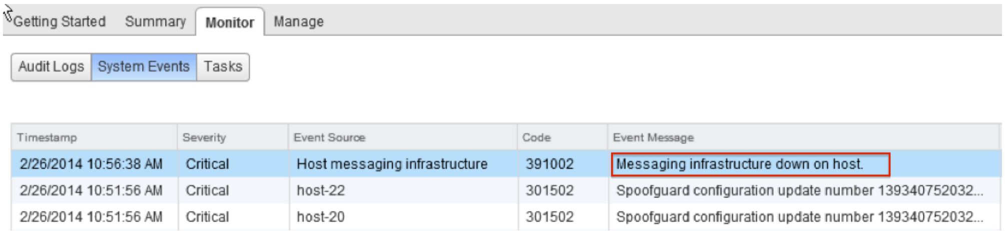
In rare cases, the installation of the VIBs succeeds but for some reason one or both of the user world agents is not functioning correctly. This could manifest itself as:

- The firewall showing a bad status.



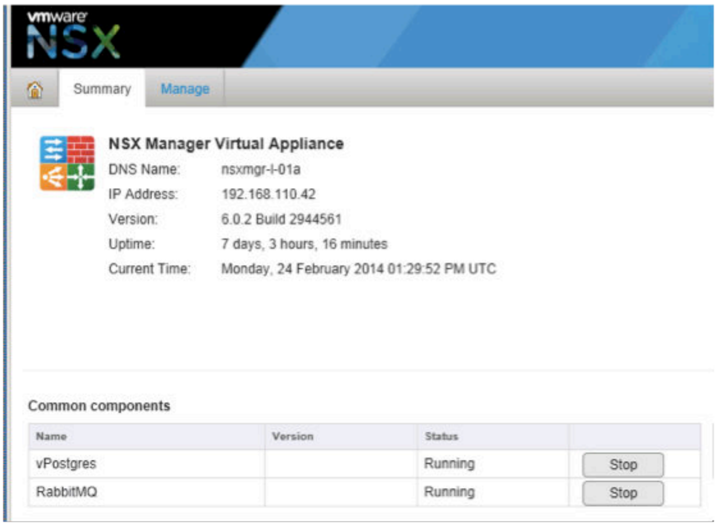
Clusters & Hosts	Installation Status	Firewall
c-1	✓ 6.0 Uninstall	⚠ Error

- The control plane between hypervisors and the Controllers being down. Check NSX Manager System Events.



Timestamp	Severity	Event Source	Code	Event Message
2/26/2014 10:56:38 AM	Critical	Host messaging infrastructure	391002	Messaging infrastructure down on host.
2/26/2014 10:51:56 AM	Critical	host-22	301502	Spoofguard configuration update number 139340752032...
2/26/2014 10:51:56 AM	Critical	host-20	301502	Spoofguard configuration update number 139340752032...

If more than one ESXi host is affected, check the status of message bus service on NSX Manager Appliance web UI under the **Summary** tab. If RabbitMQ is stopped, restart it.



NSX Manager Virtual Appliance

DNS Name: nsxmgr-l-01a
IP Address: 192.168.110.42
Version: 6.0.2 Build 2944561
Uptime: 7 days, 3 hours, 16 minutes
Current Time: Monday, 24 February 2014 01:29:52 PM UTC

Common components

Name	Version	Status	
vPostgres		Running	Stop
RabbitMQ		Running	Stop

If the message bus service is active on NSX Manager:

- Check the messaging bus user world agent status on the hosts by running the `/etc/init.d/vShield-Stateful-Firewall status` command on ESXi hosts.

```
[root@esx-01a:~] /etc/init.d/vShield-Stateful-Firewall status
vShield-Stateful-Firewall is running
```

- Check the message bus user world logs on hosts at `/var/log/vsfwd.log`.
- Run the `esxcfg-advcfg -l | grep Rmq` command on ESXi hosts to show all Rmq variables. There should be 16 Rmq variables.

```
[root@esx-01a:~] esxcfg-advcfg -l | grep Rmq
/UserVars/RmqIpAddress [String] : Connection info for RMQ Broker
/UserVars/RmqUsername [String] : RMQ Broker Username
/UserVars/RmqPassword [String] : RMQ Broker Password
/UserVars/RmqVHost [String] : RMQ Broker VHost
/UserVars/RmqVsmRequestQueue [String] : RMQ Broker VSM Request Queue
/UserVars/RmqPort [String] : RMQ Broker Port
/UserVars/RmqVsmExchange [String] : RMQ Broker VSM Exchange
/UserVars/RmqClientPeerName [String] : RMQ Broker Client Peer Name
/UserVars/RmqHostId [String] : RMQ Broker Client HostId
/UserVars/RmqHostVer [String] : RMQ Broker Client HostVer
/UserVars/RmqClientId [String] : RMQ Broker Client Id
/UserVars/RmqClientToken [String] : RMQ Broker Client Token
/UserVars/RmqClientRequestQueue [String] : RMQ Broker Client Request Queue
/UserVars/RmqClientResponseQueue [String] : RMQ Broker Client Response Queue
/UserVars/RmqClientExchange [String] : RMQ Broker Client Exchange
/UserVars/RmqSslCertSha1ThumbprintBase64 [String] : RMQ Broker Server Certificate base64 Encoded
Sha1 Hash
```

- Run the `esxcfg-advcfg -g /UserVars/RmqIpAddress` command on ESXi hosts. The output should display the NSX Manager IP address.

```
[root@esx-01a:~] esxcfg-advcfg -g /UserVars/RmqIpAddress
Value of RmqIpAddress is 192.168.110.15
```

- Run the `esxcli network ip connection list | grep 5671` command on ESXi hosts to check for active messaging bus connection.

```
[root@esx-01a:~] esxcli network ip connection list | grep 5671
tcp         0       0  192.168.110.51:29969            192.168.110.15:5671   ESTABLISHED
35505  newreno  vsfwd
tcp         0       0  192.168.110.51:29968            192.168.110.15:5671   ESTABLISHED
35505  newreno  vsfwd
```

To determine the reason for the netcpa user world agent being down:

- Check the netcpa user world agent status on hosts by running the `/etc/init.d/netcpad status` command on ESXi hosts.

```
[root@esx-01a:~] /etc/init.d/netcpad status
netCP agent service is running
```

- Check the netcpa user world agent configurations /etc/vmware/netcpa/config-by-vsm.xml. The IP addresses of the NSX Controllers should be listed.

```
[root@esx-01a:~] more /etc/vmware/netcpa/config-by-vsm.xml
<config>
  <connectionList>
    <connection id="0000">
      <port>1234</port>
      <server>192.168.110.31</server>
      <sslEnabled>true</sslEnabled>
      <thumbprint>A5:C6:A2:B2:57:97:36:F0:7C:13:DB:64:9B:86:E6:EF:1A:7E:5C:36</thumbprint>
    </connection>
    <connection id="0001">
      <port>1234</port>
      <server>192.168.110.32</server>
      <sslEnabled>true</sslEnabled>
      <thumbprint>12:E0:25:B2:E0:35:D7:84:90:71:CF:C7:53:97:FD:96:EE:ED:7C:DD</thumbprint>
    </connection>
    <connection id="0002">
      <port>1234</port>
      <server>192.168.110.33</server>
      <sslEnabled>true</sslEnabled>
      <thumbprint>BD:DB:BA:B0:DC:61:AD:94:C6:0F:7E:F5:80:19:44:51:BA:90:2C:8D</thumbprint>
    </connection>
  </connectionList>
 ...
```

- Run the `esxcli network ip connection list | grep 1234` command to verify the Controller TCP connections.

  ```
  >[root@esx-01a:~] esxcli network ip connection list | grep 1234
  tcp     0   0  192.168.110.51:16594     192.168.110.31:1234   ESTABLISHED     36754  newreno
  netcpa-worker
  tcp     0   0  192.168.110.51:46917     192.168.110.33:1234   ESTABLISHED     36754  newreno
  netcpa-worker
  tcp     0   0  192.168.110.51:47891     192.168.110.32:1234   ESTABLISHED     36752  newreno
  netcpa-worker
  ```

VXLAN Preparation

NSX prepares the DVS selected by the user for VXLAN.

This requires NSX to create a DVPortgroup on the DVS for VTEP vmknics to use.

The teaming, load balancing method, MTU, and VLAN ID is chosen during VXLAN configuration. The teaming and load balancing methods must match the configuration of the DVS selected for the VXLAN.

The MTU must be set to be at least 1600 and not less than what is already configured on the DVS.

The number of VTEPs created depends on the teaming policy selected and the DVS configuration.

Common Issues During VXLAN Preparation

During the configuration of VXLAN, the typical kinds of issues that can be encountered are as follows:

- Teaming method chosen for VXLAN does not match what can be supported by the DVS. See the *VMware NSX for vSphere Network Virtualization Design Guide* at https://communities.vmware.com/docs/DOC-27683.
- Incorrect VLAN ID chosen for the VTEPs.
- DHCP selected to assign VTEP IP addresses, but no DHCP server is available.
- A vmknic is missing “force-Sync” the configuration.
- A vmknic has a bad IP address.

Important Port Numbers

The VXLAN UDP port is used for UDP encapsulation. By default, the VXLAN UDP port number is 8472. In NSX 6.2 and later installations that use a hardware VTEP, you must use VXLAN UDP port number 4789 instead. It can be modified via the REST API.

```
PUT /2.0/vdn/config/vxlan/udp/port/4789
```

Port 80 must be open from NSX Manager to the hosts. This is used to download the VIB/agent.

Port 443/TCP from, to, and among the ESXi hosts, the vCenter Server, and NSX Data Security.

Additionally, the following ports must be open on NSX Manager:

- 443/TCP: Required for downloading the OVA file on the ESXi host for deployment, for using REST APIs, and for the NSX Manager user interface.
- 80/TCP: Required for initiating a connection to the vSphere SDK and for messaging between NSX Manager and NSX host modules.
- 1234/TCP: Requred for communication between ESXi Host and NSX Controller Clusters.
- 5671: Required for Rabbit MQ (a messaging bus technology).
- 22/TCP: Required for console access (SSH) to the CLI. By default, this port is closed.

If the hosts in your clusters were upgraded from vCenter Server version 5.0 to 5.5, you must open ports 80 and 443 on those hosts for Guest Introspection installation to be successful.

Checking Communication Channel Health

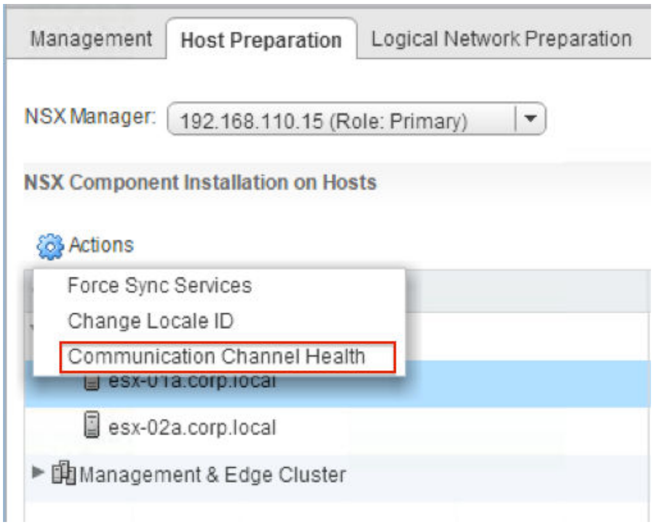
From vSphere Web Client, you can check the status of communication between various components.

To check the communication channel health between NSX Manager and the firewall agent, NSX Manager and the control plane agent, and the control plane agent and controllers, perform the following steps:

1 In vSphere Web Client, navigate to **Networking & Security > Installation > Host Preparation**.

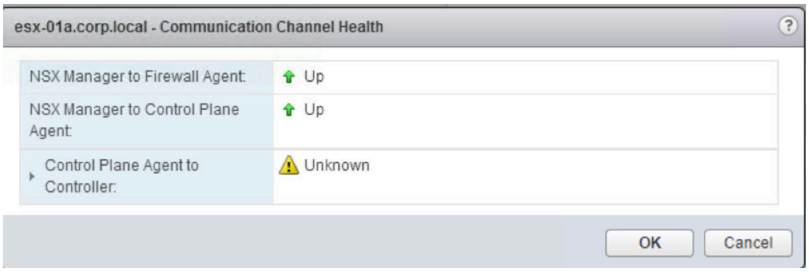
2 Select a cluster or expand a cluster and select a host. Click **Actions** (⚙) then **Communication Channel Health**.

The communication channel health information is displayed.



NSX Manager: 192.168.110.15 (Role: Primary)

Management
Host Preparation
Logical Network Preparation
NSX Component Installation on Hosts
Actions
Force Sync Services
Change Locale ID
Communication Channel Health
esx-01a.corp.local
esx-02a.corp.local
Management & Edge Cluster



esx-01a.corp.local - Communication Channel Health

NSX Manager to Firewall Agent:	↑ Up
NSX Manager to Control Plane Agent:	↑ Up
▸ Control Plane Agent to Controller:	⚠ Unknown

If the status of any of the three connections for a host changes, a message is written to the log. In the log message, the status of a connection can be UP, DOWN, or NOT_AVAILABLE (displayed as Unknown in vSphere Web Client). If the status changes from UP to DOWN or NOT_AVAILABLE, a warning message is generated. For example:

```
2016-05-23 23:36:34.736 GMT+00:00  WARN TaskFrameworkExecutor-25 VdnInventoryFacadeImpl
$HostStatusChangedEventHandler:200 - Host Connection Status Changed: Event Code: 1941, Host:
esx-04a.corp.local (ID: host-46), NSX Manager - Firewall Agent: UP, NSX Manager - Control Plane Agent:
UP, Control Plane Agent - Controllers: DOWN.
```

If the status changes from DOWN or NOT_AVAILABLE to UP, an INFO message that is similar to the warning message is generated. For example:

```
2016-05-23 23:55:12.736 GMT+00:00  INFO TaskFrameworkExecutor-25 VdnInventoryFacadeImpl
$HostStatusChangedEventHandler:200 - Host Connection Status Changed: Event Code: 1938, Host:
esx-04a.corp.local (ID: host-46), NSX Manager - Firewall Agent: UP, NSX Manager - Control Plane Agent:
UP, Control Plane Agent - Controllers: UP.
```

Troubleshooting NSX Manager Issues

Problem

- Installing VMware NSX Manager fails.
- Upgrading VMware NSX Manager fails.
- Logging in to VMware NSX Manager fails.
- Accessing VMware NSX Manager fails.

Solution

Validate that each troubleshooting step is true for your environment. Each step provides instructions to eliminate possible causes and take corrective action as necessary. The steps are ordered in the most appropriate sequence to isolate the issue and identify the proper resolution. Do not skip a step.

Procedure

1. Check the *NSX Release Notes* for current releases to see if the problem is resolved in a bug fix.
2. Ensure that the minimum system requirements are met when installing VMware NSX Manager.

   See the *NSX Installation Guide*.

3 Verify that all required ports are open in NSX Manager.

   See the *NSX Installation Guide*.

4 Installation issues:

   - If configuring the lookup service or vCenter Server fails, verify that the NSX Manager and lookup service appliances are in time sync. Use the same NTP server configurations on both NSX Manager and the lookup service. Also ensure that DNS is properly configured.
   - Verify that the OVA file is getting installed correctly. If an NSX OVA file cannot be installed, an error window in the vSphere client notes where the failure occurred. Also, verify and validate the MD5 checksum of the downloaded OVA/OVF file.
   - Verify that the time on the ESXi hosts is in sync with NSX Manager.
   - VMware recommends that you schedule a backup of the NSX Manager data immediately after installing NSX Manager.

5 Upgrade issues:

   - Before upgrading, see the latest interoperability information in the Product Interoperability Matrixes page.
   - VMware recommends that you back up your current configuration and download technical support logs before upgrading.
   - A force-resync with the vCenter Server may be required after the NSX Manager upgrade. To do this, log in to the NSX Manager Web Interface GUI. Then go to **Manage vCenter Registration > NSX Management Service > Edit** and re-enter the password for the administrative user.

6 Performance issues:

   - Ensure that the minimum vCPU requirements are met.
   - Verify that the root (/) partition has adequate space. You can verify this by logging in to the ESXi host and typing this command `df -h`.

     For example:

     ```
     [root@esx-01a:~] df -h
     Filesystem   Size   Used Available Use% Mounted on
     NFS        111.4G  80.8G     30.5G  73% /vmfs/volumes/ds-site-a-nfs01
     vfat       249.7M 172.2M     77.5M  69% /vmfs/volumes/68cb5875-d887b9c6-a805-65901f83f3d4
     vfat       249.7M 167.7M     82.0M  67% /vmfs/volumes/fe84b77a-b2a8860f-38cf-168d5dfe66a5
     vfat       285.8M 206.3M     79.6M  72% /vmfs/volumes/54de790f-05f8a633-2ad8-00505603302a
     ```

   - Use the `esxtop` command to check which processes are using large amounts of CPU and memory.
   - If the NSX Manager encounters any out-of-memory errors in the logs, verify that the `/common/dumps/java.hprof` file exists. If this file exists, create a copy of the file and include this with the NSX technical support log bundle.

- Verify that there are no storage latency issues in the environment.
- Attempt to migrate the NSX Manager to another ESXi host.

7 Connectivity issues:

- If NSX Manager is having connectivity issues either with vCenter Server or the ESXi host, log in to the NSX Manager CLI console, run the command: `debug connection IP_of_ESXi_or_VC`, and examine the output.
- Verify that the Virtual Center Web management services is started and the browser is not in an error state.
- If the NSX Manager Web User Interface (UI) is not updating, you can attempt to resolve the issue by disabling and then re-enabling the Web services. See https://kb.vmware.com/kb/2126701.
- Verify which port group and uplink NIC is used by the NSX Manager using the `esxtop` command on the ESXi host. For more information, see https://kb.vmware.com/kb/1003893.
- Attempt to migrate the NSX Manager to another ESXi host.
- Check the NSX Manager virtual machine appliance **Tasks and Events** tab from the vSphere Web Client under the **Monitor** tab.
- If the NSX Manager is having connectivity issues with vCenter Server, attempt to migrate the NSX Manager to the same ESXi host where the vCenter Server virtual machine is running to eliminate possible underlying physical network issues.

  Note that this only works if both virtual machines are on the same VLAN/port group.

Recover from an NSX Controller Failure

In case of an NSX Controller failure, you may still have two controllers that are working. The cluster majority is maintained, and the control plane continues to function. Even so, it is important to delete all three controllers and add new ones, so as to maintain a fully functional three-node cluster.

We recommend deleting the controller cluster when one or more of the controllers encounter catastrophic, unrecoverable errors or when one or more of the controller VMs become inaccessible and cannot be fixed.

We recommend deleting all controllers in such a case, even if some of the controllers seem healthy. The recommended process is to create a new controller cluster and use the Update Controller State mechanism on the NSX Manager to synchronize the state to the controllers.

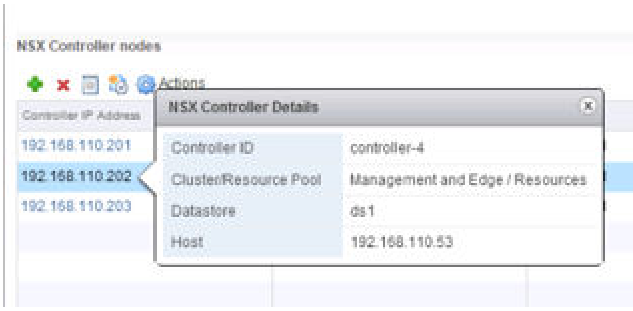
Procedure

1 Login to vSphere Web Client.

2 From **Networking & Security**, click **Installation > Management**.

3 In the NSX Controller nodes section, click each controller and take screen shots/print-screens of the details screens or write down the configuration information for later reference.

For example:



Controller ID	controller-4
Cluster/Resource Pool	Management and Edge / Resources
Datastore	ds1
Host	192.168.110.53

4 In the NSX Controller nodes section, delete all three of them by selecting each one and clicking the **Delete Node (x)** icon.

When there are no controllers in the system, the hosts are operating in what is called "headless" mode. New VMs or vMotioned VMs will have networking issues until new controllers are deployed and the synchronization is completed.

5 Deploy three new NSX Controller nodes by clicking the **Add Node (+)** icon.

6 In the Add Controller dialog box, select the datacenter on which you are adding the nodes, and configure the controller settings.

a Select the appropriate cluster.

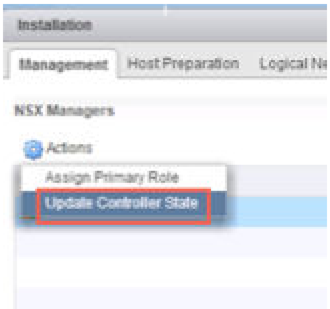
b Select a Host in the cluster and storage.

c Select the distributed port-group.

d Select the IP pool from which IP addresses are to be assigned to the node.

e Click **OK**, wait for installation to complete, and ensure all nodes have a status of Normal.

7 Resynchronize the controller state by clicking **Actions > Update Controller State**.

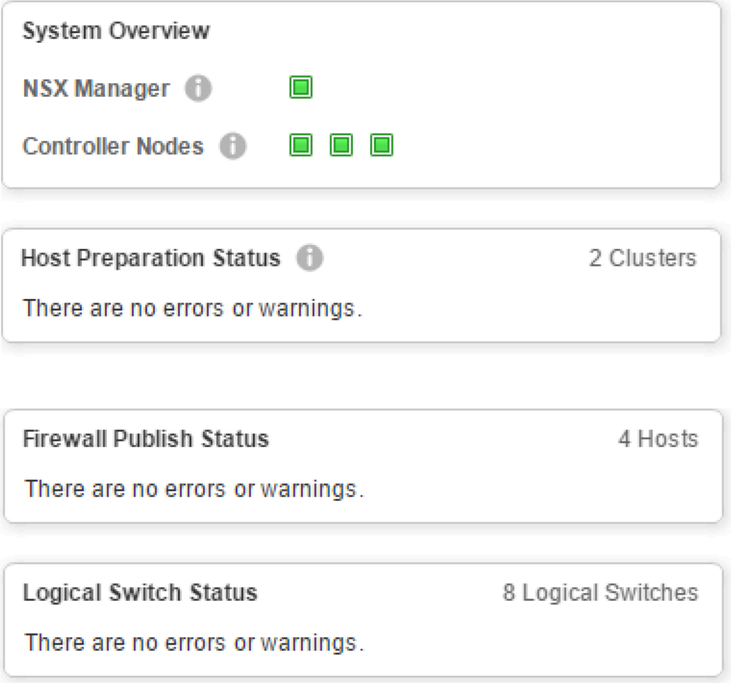


Update Controller State pushes the current VXLAN and Distributed Logical Router configuration (including Universal Objects in a Cross-VC NSX deployment) from NSX Manager to the Controller Cluster.

Using the NSX Dashboard

The NSX dashboard simplifies troubleshooting by providing visibility into the overall health of NSX components in one central view.

You can access the dashboard from vCenter Web Client **> Networking & Security > Dashboard**.



System Overview

NSX Manager 🟩

Controller Nodes 🟩 🟩 🟩

Host Preparation Status — 2 Clusters

There are no errors or warnings.

Firewall Publish Status — 4 Hosts

There are no errors or warnings.

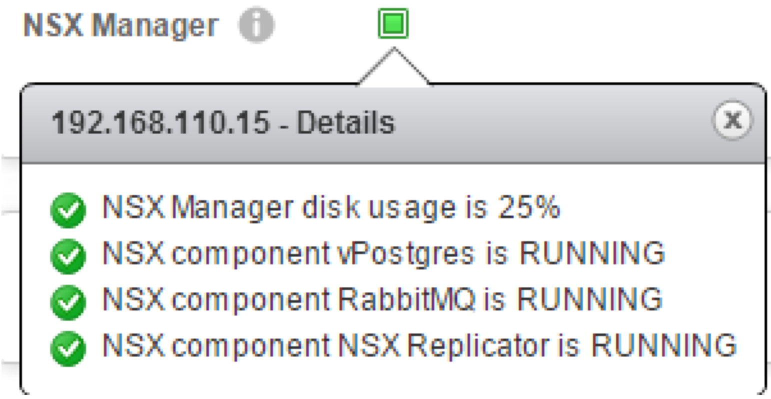
Logical Switch Status — 8 Logical Switches

There are no errors or warnings.

The dashboard checks the following states:

- NSX infrastructure—NSX Manager status
  - Component status for following services is monitored
    - Database service
    - Message bus service
    - Replicator service—Also monitors for replication errors
  - NSX manager disk usage:
    - Yellow (disk usage >80%)

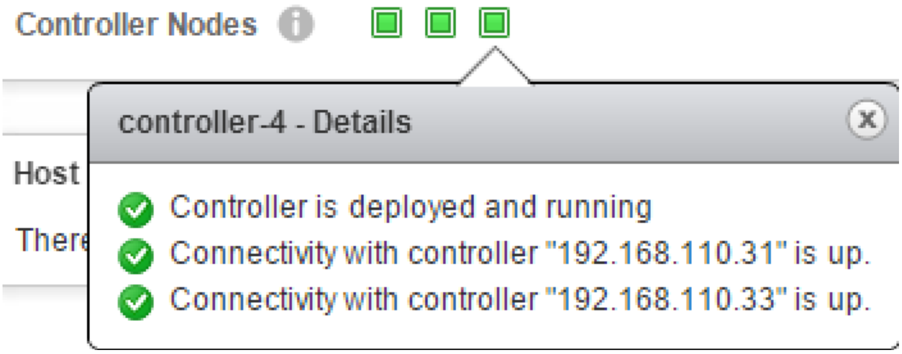
    - Red (disk usage >90%)



192.168.110.15 - Details

✓ NSX Manager disk usage is 25%
✓ NSX component vPostgres is RUNNING
✓ NSX component RabbitMQ is RUNNING
✓ NSX component NSX Replicator is RUNNING

- NSX infrastructure—NSX Controller status
  - Controller node status (running/deploying/removing/failed/unknown)
  - Controller peer connectivity status
  - Controller VM status (powered off/deleted)
  - Controller disk latency alerts



controller-4 - Details

✓ Controller is deployed and running
✓ Connectivity with controller "192.168.110.31" is up.
✓ Connectivity with controller "192.168.110.33" is up.

- NSX infrastructure—Host status
  - Deployment related:
    - Number of clusters with installation failed status
    - Number of clusters that need upgrade
    - Number of clusters where installation is in progress
  - Firewall:
    - Number of clusters with firewall disabled
    - Number of clusters where firewall status is red/yellow
  - VXLAN:
    - Number of clusters with VXLAN not configured
    - Number of clusters where VXLAN status is red/yellow

- NSX services—Firewall publish status
  - Number of hosts with firewall publish status failed.
- NSX services—Logical Networking status
  - Number of logical switches with status Error, Warning
  - Flag if backing DVS portgroup is deleted for a virtual wire

Using the show host health-status Command

From the NSX Manager central CLI, you can check the health status of each ESXi host.

The health status is reported as critical, unhealthy, or healthy.

For example:

```
nsxmgr> show host host-30 health-status
status: HEALTHY

nsxmgr> show host host-29 health-status
UNHEALTHY, Standard Switch vSwitch1 has no uplinks.
UNHEALTHY, Storage volume datastore1 has no enough free spaces: 19.% free.
status: UNHEALTHY

nsxmgr> show host host-28 health-status
CRITICAL, VXLAN VDS vds-site-a VNI 200000 multicast addr is not synchronized with VSM: 0.0.0.0.
CRITICAL, VXLAN VDS vds-site-a VNI 200003 multicast addr is not synchronized with VSM: 0.0.0.0.
CRITICAL, VXLAN VDS vds-site-a VNI 5000 multicast addr is not synchronized with VSM: 0.0.0.0.
Status: CRITICAL
```

The host-check command can also be invoked through the NSX Manager API.

Setting the Logging Level of NSX Components

You can set the logging level for each NSX component.

The supported levels vary by component, as shown here.

```
nsxmgr> set
  hardware-gateway  Show Logical Switch Commands
  PACKAGE-NAME      Set log level
  controller        Show Logical Switch Commands
  host              Show Logical Switch Commands

nsxmgr> set hardware-gateway agent 10.1.1.1 logging-level
  ERROR
  WARN
  INFO
  DEBUG
  TRACE

nsxmgr-01a> set <package-name> logging-level
  OFF
  FATAL
  ERROR
  WARN
  INFO
  DEBUG
  TRACE

nsxmgr> set controller 192.168.110.31
  java-domain    Set controller node log level
  native-domain  Set controller node log level

nsxmgr> set controller 192.168.110.31 java-domain logging-level
  OFF
  FATAL
  ERROR
  WARN
  INFO
  DEBUG
  TRACE

nsxmgr> set controller 192.168.110.31 native-domain logging-level
  ERROR
  WARN
  INFO
  DEBUG
  TRACE

nsxmgr> set host host-28
  netcpa  Set host node log level by module
  vdl2    Set host node log level by module
  vdr     Set host node log level by module

nsxmgr> set host host-28 netcpa logging-level
  FATAL
  ERROR
  WARN
  INFO
  DEBUG

nsxmgr> set host host-28 vdl2 logging-level
  ERROR
  INFO
  DEBUG
  TRACE

nsxmgr> set host host-28 vdr logging-level
  OFF
  ERROR
  INFO
```

vSphere ESX Agent Manager

vSphere ESX Agent Manager (EAM) automates the process of deploying and managing vSphere ESX Agents, while extending the function of an ESXi host to provide additional services that a vSphere solution requires.

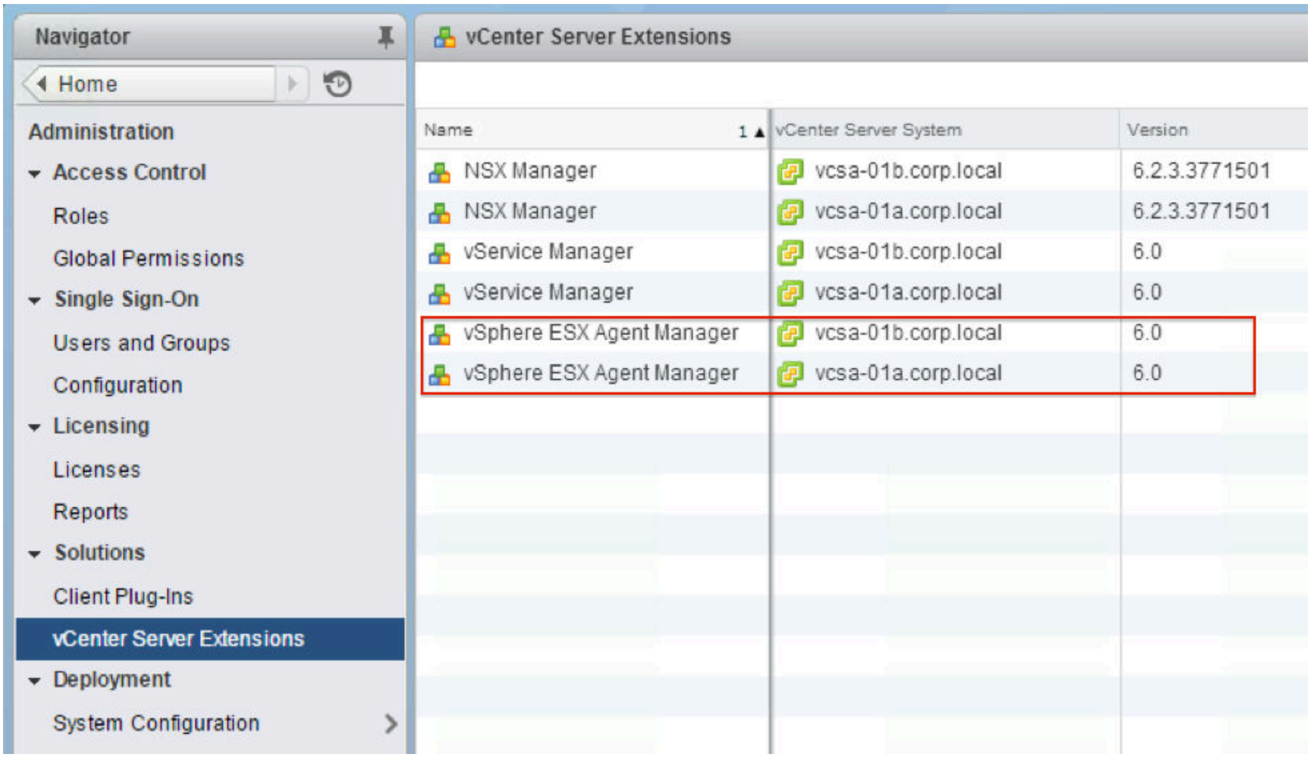
ESX agents are relevant to NSX troubleshooting, because, for example, an NSX deployment might require a particular network filter or firewall configuration to function. A firewall configuration can use an ESX agent to connect to the vSphere hypervisor and extend the host with functions specific to that configuration. For example, the ESX agent can filter network traffic, act as a firewall, or gather other information about the virtual machines on the host.

ESX agent virtual machines are similar to services in Windows or Linux. They start when the operating system starts and they stop when it shuts down. The behavior of ESX agent virtual machines is transparent to the user. A vSphere host reaches the ready state when the ESXi operating system has started and all ESX agent virtual machines have been provisioned and powered on.

To integrate an agent with vSphere ESX Agent Manager and extend the capabilities of an ESXi server, an ESX agent must be packaged as an OVF or a VIB module.

EAM allows you to monitor the health of ESX agents and blocks users from performing certain operations on ESX agents that might affect the virtual machines that use them. It also manages the lifecycle of agent VIBs and VMs. For example, ESX Agent Manager can prevent an ESX agent virtual machine from being powered off or moved from an ESXi host that contains other virtual machines that use that agent.

The following screen shot shows the UI to access the ESX Agent Manager.



vCenter Server Extensions

Name 1 ▲	vCenter Server System	Version
NSX Manager	vcsa-01b.corp.local	6.2.3.3771501
NSX Manager	vcsa-01a.corp.local	6.2.3.3771501
vService Manager	vcsa-01b.corp.local	6.0
vService Manager	vcsa-01a.corp.local	6.0
vSphere ESX Agent Manager	vcsa-01b.corp.local	6.0
vSphere ESX Agent Manager	vcsa-01a.corp.local	6.0

Logs and Services of the vSphere ESX Agent Manager (EAM)

EAM logs are included as part of the vCenter log bundle.

- Windows—C:\ProgramData\VMware\vCenterServer\logs\eam\eam.log
- VCSA—/var/log/vmware/vpx/eam.log
- ESXi—/var/log/esxupdate.log

vSphere ESX Agents and Agencies

vSphere ESX agencies map to a prepared NSX host cluster. Each ESX agency acts as a container for ESX agents. ESX agencies aggregate information about the agents that they manage. Thus, ESX agencies provide an overview of the ESX agents that they contain by aggregating all the issues that relate to the ESX agents.

ESX Agent Manager reports issues in agency runtime information. ESX Agent Manager can automatically resolve certain issues if the administrator clicks **Resolve Issues** in the ESX Agent Manager tab. For example, if an ESX agent is powered off, it can be powered back on.

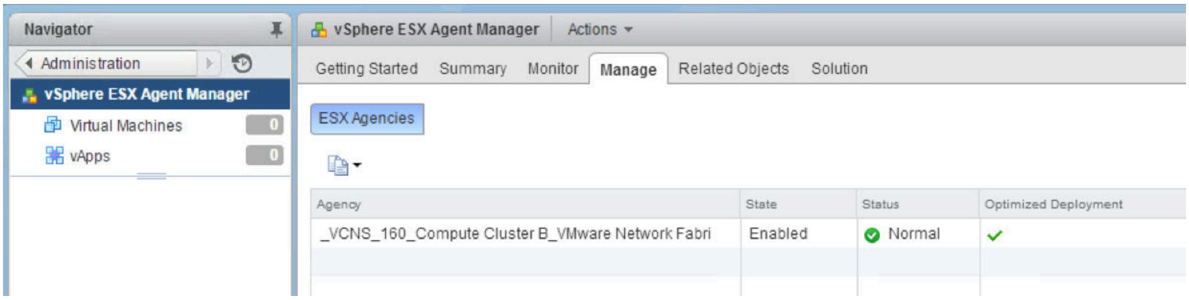
Note If the scope of an ESX agency is empty, there are no compute resources onto which to deploy ESX agents, so no ESX agents are deployed. In this case, ESX Agent Manager determines that the ESX agency has performed correctly, and sets the status to green.

The configuration of each agency specifies how the agency deploys its agents and VIBs. See https://pubs.vmware.com/vsphere-60/index.jsp#com.vmware.eam.apiref.doc/eam.Agency.ConfigInfo.html.

Important Make sure to change the bypassVumEnabled flag to True before starting the NSX installation and change it back to False after the installation. See https://kb.vmware.com/kb/2053782.

To check the EAM status in the vSphere Web Client, go to **Administration > vCenter Server Extentions**.

The EAM **Manage** tab shows information about running agencies, lists any orphaned ESX agents, and logs information about the ESX agents that ESX Agent Manager manages.



Agency	State	Status	Optimized Deployment
_VCNS_160_Compute Cluster B_VMware Network Fabri	Enabled	✓ Normal	✓

For more information about agents and agencies, see https://pubs.vmware.com/vsphere-60/index.jsp#com.vmware.vsphere.ext_solutions.doc/GUID-40838DE9-6AD1-45E3-A1DE-B2B24A9E715A.html.

NSX CLI Cheat Sheet

Table 2-1. Checking the NSX Installation on ESXi Host—Commands Run from NSX Manager

Description	Commands on NSX Manager	Notes
List all clusters to get the cluster IDs	`show cluster all`	View all cluster information
List all the hosts in the cluster to get the host IDs	`show cluster CLUSTER-ID`	View the list of hosts in the cluster, the host-ids, and the host-prep installation status
List all the VMs on a host	`show host HOST-ID`	View particular host information, VMs, VM IDs, and power status

Table 2-2. Checking the NSX Installation on ESXi Host—Commands Run from Host

Description	Commands on Host	Notes
Three VIBs are loaded: esx-vxlan; esx-vsip; esx-dvfilter-switch-security	`esxcli software vib get --vibname <name>`	Check the version/date installed `esxcli software vib list` displays a list of all VIBs on the system
List all the system modules currently loaded in the system	`esxcli system module list`	Older equivalent command: `vmkload_mod -l | grep -E vdl2|vdrb|vsip|dvfilter-switch-security`
Four Modules are loaded: vdl2, vdrb, vsip, dvfilter-switch-security	`esxcli system module get -m <name>`	Run the command for each module

Two User World Agents (UWA) : netcpad, vsfwd	`/etc/init.d/vShield-Stateful-Firewall status`
	`/etc/init.d/netcpad status`	

Check UWAs connection, port 1234 to controllers and 5671 to NSX Manager	`esxcli network ip connection list | grep 1234`
	`esxcli network ip connection list | grep 5671`
		Controller TCP connection
		Message bus TCP connection

Check EAM status	Web UI, check **Administration > vCenter ESX Agent Manager**	

Table 2-3. Checking the NSX Installation on ESXi Host—Host Networking Commands

Description	Host Networking Commands	Notes
List physical NICs/vmnic	`esxcli network nic list`	Check the NIC type, driver type, link status, MTU
Physical NIC details	`esxcli network nic get -n vmnic#`	Check the driver and firmware versions along with other details
List vmk NICs with IP addresses/MAC/MTU, and so on	`esxcli network ip interface ipv4 get`	To ensure VTEPs are correctly instantiated
Details of each vmk NIC, including vDS information	`esxcli network ip interface list`	To ensure VTEPs are correctly instantiated

Table 2-3. Checking the NSX Installation on ESXi Host—Host Networking Commands (Continued)

Description	Host Networking Commands	Notes
Details of each vmk NIC, including vDS info for VXLAN vmks	`esxcli network ip interface list --netstack=vxlan`	To ensure VTEPs are correctly instantiated
Find the VDS name associated with this host's VTEP	`esxcli network vswitch dvs vmware vxlan list`	To ensure VTEPs are correctly instantiated
Ping from VXLAN-dedicated TCP/IP stack	`ping ++netstack=vxlan -I vmk1 x.x.x.x`	To troubleshoot VTEP communication issues: add option `-d -s 1572` to make sure that the MTU of transport network is correct for VXLAN
View routing table of VXLAN-dedicated TCP/IP stack	`esxcli network ip route ipv4 list -N vxlan`	To troubleshoot VTEP communication issues
View ARP table of VXLAN-dedicated TCP/IP stack	`esxcli network ip neighbor list -N vxlan`	To troubleshoot VTEP communication issues

Table 2-4. Checking the NSX Installation on ESXi Host—Host Log Files

Description	Log File	Notes
From NSX Manager	`show manager log follow`	Tails the NSX Manager logs For live troubleshooting
Any installation related logs for a host	`/var/log/esxupdate.log`	

Host related issues
VMkernel warning, messages, alerts, and availability report
	`/var/log/vmkernel.log`
	`/var/log/vmksummary.log`
	`/var/log/vmkwarning.log`	

Module load failure is captured	`/var/log/syslog`	IXGBE driver failure NSX modules dependency failure are key indicators
On vCenter, ESX Agent Manager is responsible for updates	In vCenter logs, `eam.log`	

Table 2-5. Checking Logical Switching—Commands Run from NSX Manager

Description	Command on NSX Manager	Notes
List all logical switches	`show logical-switch list all`	List all the logical switches, their UUIDs to be used in API, transport zone, and vdnscope

Table 2-6. Logical Switching—Commands Run from NSX Controller

Description	Commands on Controller	Notes
Find the controller that is the owner of the VNI	`show control-cluster logical-switches vni 5000`	Note the controller IP address in the output and SSH to it
Find all the hosts that are connected to this controller for this VNI	`show control-cluster logical-switch connection-table 5000`	The source IP address in output is the management interface of host, and the port number is the source port of TCP connection

Table 2-6. Logical Switching—Commands Run from NSX Controller (Continued)

Description	Commands on Controller	Notes
Find the VTEPs registered to host this VNI	`show control-cluster logical-switches vtep-table 5002`	
List the MAC addresses learned for VMs on this VNI	`show control-cluster logical-switches mac-table 5002`	Map that the MAC address is actually on the VTEP reporting it
List the ARP cache populated by the VM IP updates	`show control-cluster logical-switches arp-table 5002`	ARP cache expires in 180 secs
For a specific host/controller pair, find out which VNIs host has joined	`show control-cluster logical-switches joined-vnis <host_mgmt_ip>`	

Table 2-7. Logical Switching—Commands Run from Hosts

Description	Command on Hosts	Notes
Check if the host VXLAN is in-sync or not	`esxcli network vswitch dvs vmware vxlan get`	Shows the sync state and port used for encapsulation
View VM attached and local switch port ID for datapath captures	`net-stats -l`	A nicer way to get vm switchport for a specific VM
Verify VXLAN kernel module vdl2 is loaded	`esxcli system module get -m vdl2`	Shows full detail of the specified module. Verify the version
Verify correct VXLAN VIB version is installed	`esxcli software vib get --vibname esx-vxlan`	Shows full detail of the specified VIB Verify the version and date
Verify the host knows about other hosts in the logical switch	`esxcli network vswitch dvs vmware vxlan network vtep list --vxlan-id=5001 --vds-name=Compute_VDS`	Shows list of all the VTEPs that this host knows about that are hosting vtep 5001
Verify control plane is up and active for a Logical switch	`esxcli network vswitch dvs vmware vxlan network list --vds-name Compute_VDS`	Make sure the controller connection is up and the Port/Mac count matches the VMs on the LS on this host
Verify host has learnt MAC addresses of all VMs	`esxcli network vswitch dvs vmware vxlan network mac list --vds-name Compute_VDS --vxlan-id=5000`	This should list all the MACs for the VNI 5000 VMs on this host
Verify host has locally cached ARP entry for remote VM's	`esxcli network vswitch dvs vmware vxlan network arp list --vds-name Compute_VDS --vxlan-id=5000`	Verify host has locally cached ARP entry for remote VM's
Verify VM is connected to LS & mapped to a local VMKnic Also shows what vmknic ID a VM dvPort is mapped to	`esxcli network vswitch dvs vmware vxlan network port list --vds-name Compute_VDS --vxlan-id=5000`	the vdrport will always be listed as long as the VNI is attached to a router
View vmknic ID's and what switchport/uplink they are mapped to	`esxcli network vswitch dvs vmware vxlan vmknic list --vds-name=DSwitch-Res01`	

Table 2‑8. Checking Logical Switching—Log Files

Description	Log File	Notes
Hosts are always connected to controllers hosting their VNIs	`/etc/vmware/netcpa/config-by-vsm.xml`	This file should always have all the controllers in the environment listed The `config-by-vsm.xml` file is created by netcpa process Vsfwd only provides channel for netcpa Netcpad connects to vsfwd on port 15002
The `config-by-vsm.xml` file is pushed by NSX Manager using vsfwd If the `config-by-vsm.xml` file is not correct look at the vsfwd log	`/var/log/vsfwd.log`	Parse through this file looking for errors To restart process: `/etc/init.d/vShield-Stateful-Firewall stop|start`
Connection to controller is made using netcpa	`/var/log/netcpa.log`	Parse through this file looking for errors
VDL2 module logs are in vmkernel.log	`/var/log/vmkernel.log`	Check VDL2 module logs in /var/log/vmkernel.log “prefixed with VXLAN:”

Table 2‑9. Checking Logical Routing—Commands Run from NSX Manager

Description	Commands on NSX Manager	Notes
Commands for ESG	`show edge`	CLI commands for Edge ServicesGateway (ESG) start with 'show edge'
Commands for DLR Control VM	`show edge`	CLI commands for Distributed Logical Router (DLR) Control VM start with 'show edge'
Commands for DLR	`show logical-router`	CLI commands for Distributed Logical Router (DLR) start with `show logical-router`
List all edges	`show edge all`	List all the edges that support the central CLI
List all the services and deployment details of an edge	`show edge EDGE-ID`	View Edge Service Gateway Information
List the command options for edge	`show edge EDGE-ID ?`	View details, such as version, log, NAT, routing table, firewall, configuration, interface, and services
View routing details	`show edge EDGE-ID ip ?`	View routing info, BGP, OSPF and other details
View routing table	`show edge EDGE-ID ip route`	View the routing table at Edge
View routing neighbor	`show edge EDGE-ID ip ospf neighbor`	View routing neighbor relationship
View logical routers connection information	`show logical-router host hostID connection`	Verify that the number of LIFs connected are correct, the teaming policy is right and the appropriate vDS is being used

Table 2-9. Checking Logical Routing—Commands Run from NSX Manager (Continued)

Description	Commands on NSX Manager	Notes
List all logical router instances running on the host	`show logical-router host hostID dlr all`	Verify the number of LIFs and routes Controller IP should be same on all hosts for a logical router Control Plane Active should be yes --brief gives a compact response
Check the routing table on the host	`show logical-router host hostID dlr dlrID route`	This is the routing table pushed by the controller to all the hosts in the transport zone This must be same across all the hosts If some of the routes are missing on few hosts, try the sync command from controller mentioned earlier The E flag means routes are learned via ECMP
Check the LIFs for a DLR on the host	`show logical-router host hostID dlr dlrID interface (all | intName) verbose`	The LIF information is pushed to hosts from the controller Use this command to ensure the host knows about all the LIFs it should

Table 2-10. Checking Logical Routing—Commands Run from NSX Controller

Description	Commands on NSX Controller	Notes
Find all the Logical Router Instances	`show control-cluster logical-routers instance all`	This should list the logical router instance and all the hosts in the transport zone which should have the logical router instance on them In addition, shows the Controller that servicing this logical router
View details of each logical router	`show control-cluster logical-routers instance 0x570d4555`	The IP column shows the vmk0 IP addresses of all hosts where this DLR exists
View all the interfaces CONNECTED to the logical router	`show control-cluster logical-routers interface-summary 0x570d4555`	The IP column shows the vmk0 IP addresses of all hosts where this DLR exists
View all the routes learned by this logical router	`show control-cluster logical-routers routes 0x570d4555`	Note that the IP column shows the vmk0 IP addresses of all hosts where this DLR exists
shows all the network connections established, like a net stat output	`show network connections of-type tcp`	Check if the host you are troubleshooting has netcpa connection Established to controller

Table 2-10. Checking Logical Routing—Commands Run from NSX Controller (Continued)

Description	Commands on NSX Controller	Notes
Sync interfaces from controller to host	`sync control-cluster logical-routers interface-to-host <logical-router-id> <host-ip>`	Useful if new interface was connected to logical router but is not sync'd to all hosts
Sync routes from controller to host	`sync control-cluster logical-routers route-to-host <logical-router-id> <host-ip>`	Useful if some routes are missing on few hosts but are available on majority of hosts

Table 2-11. Checking Logical Routing—Commands Run from Edge

Description	Commands on Edge or Logical Router Control VM	Notes
View configuration	`show configuration <global | bgp | ospf | …>`	
View the routes learned	`show ip route`	Make sure the routing and forwarding tables are in sync
View the forwarding table	`show ip forwarding`	Make sure the routing and forwarding tables are in sync
View the vDR interfaces	`show interface`	First NIC shown in the output is the vDR interface The VDR interface is not a real vNIC on that VM All the subnets attached to VDR are of type INTERNAL
View the other interfaces (management)	`show interface`	Management/HA interface is a real vNIC on the logical router Control VM If HA was enabled without specifying an IP address, 169.254.x.x/ 30 is used If the management interface is given an IP address, it appears here

debug the protocol	`debug ip ospf`
	`debug ip bgp`
		Useful to see issues with the configuration (such as mismatched OSPF areas, timers, and wrong ASN)
		Note: output is only seen on the Console of Edge (not via SSH session)

Table 2-11. Checking Logical Routing—Commands Run from Edge (Continued)

Description	Commands on Edge or Logical Router Control VM	Notes

OSPF commands	`show configuration ospf`
	`show ip ospf interface`
	`show ip ospf neighbor`
	`show ip route ospf`
	`show ip ospf database`
	`show tech-support` (and look for strings “EXCEPTION” and “PROBLEM”)	

BGP commands	`show configuration bgp`
	`show ip bgp neighbor`
	`show ip bgp`
	`show ip route bgp`
	`show ip forwarding`
	`show tech-support` (look for strings “EXCEPTION” and “PROBLEM”)	

Table 2-12. Checking Logical Routing—Log Files from Hosts

Description	Log File	Notes
VDR instance information is pushed to hosts by vsfwd and saved in XML format	`/etc/vmware/netcpa/config-by-vsm.xml`	If VDR instance is missing on the host, first look at this file to see if the instance is listed If not, restart vsfwd Also, use this file to ensure that all of the controllers are known to the host
The above file is pushed by NSX Manager using vsfwd If the `config-by-vsm.xml` file is not correct look at the vsfwd log	`/var/log/vsfwd.log`	Parse through this file looking for errors To restart process: /etc/init.d/vShield-Stateful-Firewall stop|start
Connection to controller is made using netcpa	`/var/log/netcpa.log`	Parse through this file looking for errors
VDL2 module logs are in vmkernel.log	`/var/log/vmkernel.log`	Check VDL2 module logs in /var/log/vmkernel.log “prefixed with vxlan:”

Table 2-13. Controller Debugging—Command Run from NSX Manager

Descripction	Command (On NSX Manager)	Notes
List all controllers with state	`show controller list all`	Shows the list of all controllers and their running state

Table 2-14. Controller Debugging—Command Run from NSX Controller

Description	Command(On Controller)	Notes
Check controller cluster status	`show control-cluster status`	Should always show 'Join complete' and 'Connected to Cluster Majority'
Check the stats for flapping connections and messages	`show control-cluster core stats`	The dropped counter should not change
View the node's activity in relation to joining the cluster initially or after a restart	`show control-cluster history`	This is great for troubleshooting cluster join issues
View list of nodes in the cluster	`show control-cluster startup-nodes`	Note that the list doesn't have to have ONLY have active cluster nodes This should have a list of all the currently deployed controllers This list is used by starting controller to contact other controllers in the cluster
shows all the network connections established, like a net stat output	`show network connections of-type tcp`	Check if the host you are troubleshooting has netcpa connection Established to controller
To restart the controller process	`restart controller`	Only restarts the main controller process Forces a re-connection to the cluster
To reboot the controller node	`restart system`	Reboots the controller VM

Table 2-15. Controller Debugging—Log Files on NSX Controller

Description	Log File	Notes
View controller history and recent joins, restarts. and so on	`show control-cluster history`	Great troubleshooting tool for controller issues especially around clustering
Check for slow disk	`show log cloudnet/cloudnet_java-zookeeper<timestamp>.log filtered-by fsync`	A reliable way to check for slow disks is to look for "fsync" messages in the cloudnet_java-zookeeper log If sync takes more than 1 second, ZooKeeper prints this message, and it is a good indication that something else was utilizing the disk at that time
Check for slow/malfunctioning disk	`show log syslog filtered-by collectd`	Messages like the one in ample output about "collectd" tend to correlate with slow or malfunctioning disks
Check for diskspace usage	`show log syslog filtered-by freespace:`	There is a background job called "freespace" that periodically cleans up old logs and other files from the disk when the space usage reaches some threshold. In some cases, if the disk is small and/or filling up very fast, you'll see a lot of freespace messages. This could be an indication that the disk filled up

Table 2-15. Controller Debugging—Log Files on NSX Controller (Continued)

Description	Log File	Notes
Find currently active cluster members	`show log syslog filtered-by Active cluster members`	Lists the node-id for currently active cluster members. May need to look in older syslogs as this message is not printed all the time.
View the core controller logs	`show log cloudnet/cloudnet_java-zookeeper. 20150703-165223.3702.log`	There may be multiple zookeeper logs, look at the latest timestamped file This file has information about controller cluster master election and other information related to the distributed nature of controllers
View the core controller logs	`show log cloudnet/cloudnet.nsx-controller.root.log.INFO. 20150703-165223.3668`	Main controller working logs, like LIF creation, connection listener on 1234, sharding

Table 2-16. Checking Distributed Firewall—Commands Run from NSX Manager

Description	Commands on NSX Manager	Notes
View a VMs Information	`show vm VM-ID`	Details such as DC, Cluster, Host, VM Name, vNICs, dvfilters installed
View particular virtual NIC information	`show vnic VNIC-ID`	Details such as VNIC name, mac address, pg, applied filters
View all cluster information	`show dfw cluster all`	Cluster Name, Cluster Id, Datacenter Name, Firewall Status
View particular cluster information	`show dfw cluster CLUSTER-ID`	Host Name, Host Id, Installation Status
View dfw related host information	`show dfw host HOST-ID`	VM Name, VM Id, Power Status
View details within a dvfilter	`show dfw host HOST-ID filter filterID <option>`	List rules, stats, address sets etc for each VNIC
View DFW information for a VM	`show dfw vm VM-ID`	View VM's name, VNIC ID, filters, and so on
View VNIC details	`show dfw vnic VNIC-ID`	View VNIC name, ID, MAC address, portgroup, filter
List the filters installed per vNIC	`show dfw host hostID summarize-dvfilter`	Find the VM/vNIC of interest and get the name field to use in the next commands as filter

View rules for a specific filter/vNIC	`show dfw host hostID filter filterID rules`
	`show dfw vnic nicID`	

View details of an address set	`show dfw host hostID filter filterID addrsets`	The rules only display address sets, this command can be used to expand what is part of an address set
Spoofguard details per vNIC	`show dfw host hostID filter filterID spoofguard`	Check if spoofgruard is enabled and what is the current IP/MAC

Table 2-16. Checking Distributed Firewall—Commands Run from NSX Manager (Continued)

Description	Commands on NSX Manager	Notes
View details of flow records	`show dfw host hostID filter filterID flows`	If flow monitoring is enabled, host sends flow information periodically to NSX Manager Use this command to see flows per vNIC
View statistics for each rule for a vNIC	`show dfw host hostID filter filterID stats`	This is useful to see if rules are being hit

Table 2-17. Checking Distributed Firewall—Commands Run from Hosts

Description	Commands on Host	Notes
Lists VIBs downloaded on the host	`esxcli software vib list | grep vsip`	Check to make sure right vib version is downloaded
Details on system modules currently loaded	`esxcli system module get -m vsip`	Check to make sure that the module was installed/loaded
Process list	`ps | grep vsfwd`	View if the vsfwd process is running with several threads
Deamon command	`/etc/init.d/vShield-Stateful-Firewall {start|stop|status|restart}`	Check if the deamon is running and restart if needed
View network connection	`esxcli network ip connection list | grep 5671`	Check if the host has TCP connectivity to NSX Manager

Table 2-18. Checking Distributed Firewall—Log Files on Hosts

Description	Log	Notes
Process log	`/var/log/vsfwd.log`	vsfwd deamon log, useful for vsfwd process, NSX Manager connectivity, and RabbitMQ troubleshooting
Packet logs dedicated file	`/var/log/dfwpktlogs.log`	Dedicated log file for packet logs
Packet capture at the dvfilter	`pktcap-uw --dvfilter nic-1413082-eth0-vmware-sfw.2 --outfile test.pcap`	

Traceflow

3

Traceflow is a troubleshooting tool that provides the ability to inject a packet and observe where that packet is seen as it passes through the physical and logical network. The observations allow you to determine information about the network, such as identifying a node that is down or a firewall rule that is preventing a packet from being received by its destination.

This chapter includes the following topics:

- About Traceflow
- Use Traceflow for Troubleshooting

About Traceflow

Traceflow injects packets into a vSphere distributed switch (VDS) port and provides various observation points along the packet's path as it traverses physical and logical entities (such as ESXi hosts, logical switches, and logical routers) in the overlay and underlay networks. This allows you to identify the path (or paths) a packet takes to reach its destination or, conversely, where a packet is dropped along the way. Each entity reports the packet handling on input and output, so you can determine whether issues occur when receiving a packet or when forwarding the packet.

Keep in mind that traceflow is not the same as a ping request/response that goes from guest-VM stack to guest-VM stack. What traceflow does is observe a marked packet as it traverses the overlay network. Each packet is monitored as it crosses the overlay network until it reaches and is deliverable to the destination guest VM. However, the injected traceflow packet is never actually delivered to the destination guest VM. This means that a traceflow can be successful even when the guest VM is powered down.

Traceflow supports the following traffic types:

- Layer 2 unicast
- Layer 3 unicast
- Layer 2 broadcast
- Layer 2 multicast

You can construct packets with custom header fields and packet sizes. The source for the traceflow is always a virtual machine virtual NIC (vNIC). The destination endpoint can be any device in the NSX overlay or in the underlay. However, you cannot select a destination that is north of an NSX edge services gateway (ESG). The destination must be on the same subnet or must be reachable through NSX distributed logical routers.

The traceflow operation is considered Layer 2 if the source and destination vNICs are in the same Layer 2 domain. In NSX, this means that they are on the same VXLAN network identifier (VNI or segment ID). This happens, for example, when two VMs are attached to the same logical switch.

If NSX bridging is configured, unknown Layer 2 packets are always be sent to the bridge. Typically, the bridge forwards these packets to a VLAN and reports the traceflow packet as delivered. A packet reported as delivered does not necessarily mean that the trace packet was delivered to the specified destination.

For Layer 3 traceflow unicast traffic, the two end points are on different logical switches and have different VNIs, connected to a distributed logical router (DLR).

For multicast traffic, the source is a VM vNIC, and the destination is a multicast group address.

Traceflow observations may include observations of broadcasted traceflow packets. The ESXi host broadcasts a traceflow packet if it does not know the destination host's MAC address. For broadcast traffic, the source is a VM vNIC. The Layer 2 destination MAC address for broadcast traffic is FF:FF:FF:FF:FF:FF. To create a valid packet for firewall inspection, the broadcast traceflow operation requires a subnet prefix length. The subnet mask enables NSX to calculate an IP network address for the packet.

Caution Depending on the number of logical ports in your deployment, multicast and broadcast traceflow operations might generate high traffic volume.

There are two ways to use traceflow: through the API and through the GUI. The API is the same API that the GUI uses, except the API allows you to specify the exact settings within the packet, while the GUI has more limited settings.

The GUI allows you to set the following values:

- Protocol---TCP, UDP, ICMP.
- Time-to-live (TTL). The default is 64 hops.
- TCP and UDP source and destination port numbers. The default values are 0.
- TCP flags.
- ICMP ID and sequence number. Both are 0 by default.
- An expiry timeout, in milliseconds (ms), for the traceflow operation. The default is 10,000 ms.
- Ethernet frame size. The default is 128 bytes per frame. The maximum frame size is 1000 bytes per frame.
- Payload encoding. The default is Base64.

- Payload value.

Use Traceflow for Troubleshooting

There are multiple scenarios in which traceflow is useful.

Traceflow is useful in the following scenarios:

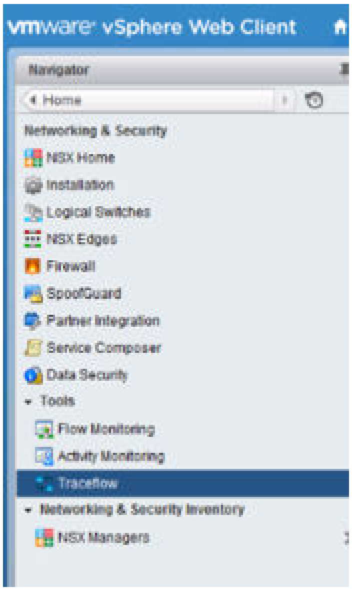
- Troubleshooting network failures to see the exact path that traffic takes
- Performance monitoring to see link utilization
- Network planning to see how a network will behave when it is in production

Prerequisites

- Traceflow operations require communication among vCenter, NSX Manager, the NSX Controller cluster and the netcpa user world agents on the hosts.
- For Traceflow to work as expected, make sure that the controller cluster is connected and in healthy state.

Procedure

1 In vCenter Web Client, navigate to **Home > Networking & Security > Traceflow**.



2 Select the traffic type: Unicast, broadcast, or multicast.

3 Select the source VM vNIC.

If the VM is managed in the same vCenter Server where you are running the traceflow, you can select the VM and vNIC from a list.

4 For a unicast traceflow, enter the destination vNIC information.

The destination can be a vNIC of any device in the NSX overlay or underlay, such as a host, a VM, a logical router, or an edge services gateway. If the destination is a VM that is running VMware Tools and is managed in the same vCenter Server from which you are running the traceflow, you can select the VM and vNIC from a list.

Otherwise, you must enter the destination IP address (and the MAC address for a unicast Layer 2 traceflow). You can gather this information from the device itself in the device console or in an SSH session. For example, if it is a Linux VM, you can get its IP and MAC address by running the `ifconfig` command in the Linux terminal. For a logical router or edge services gateway, you can gather the information from the `show interface` CLI command.

5 For a Layer 2 broadcast traceflow, enter the subnet prefix length.

The packet is switched based on MAC address only. The destination MAC address is FF:FF:FF:FF:FF:FF.

Both the source and destination IP addresses are required to make the IP packet valid for firewall inspection.

6 For a Layer 2 multicast traceflow, enter the multicast group address.

The packet is switched based on MAC address only.

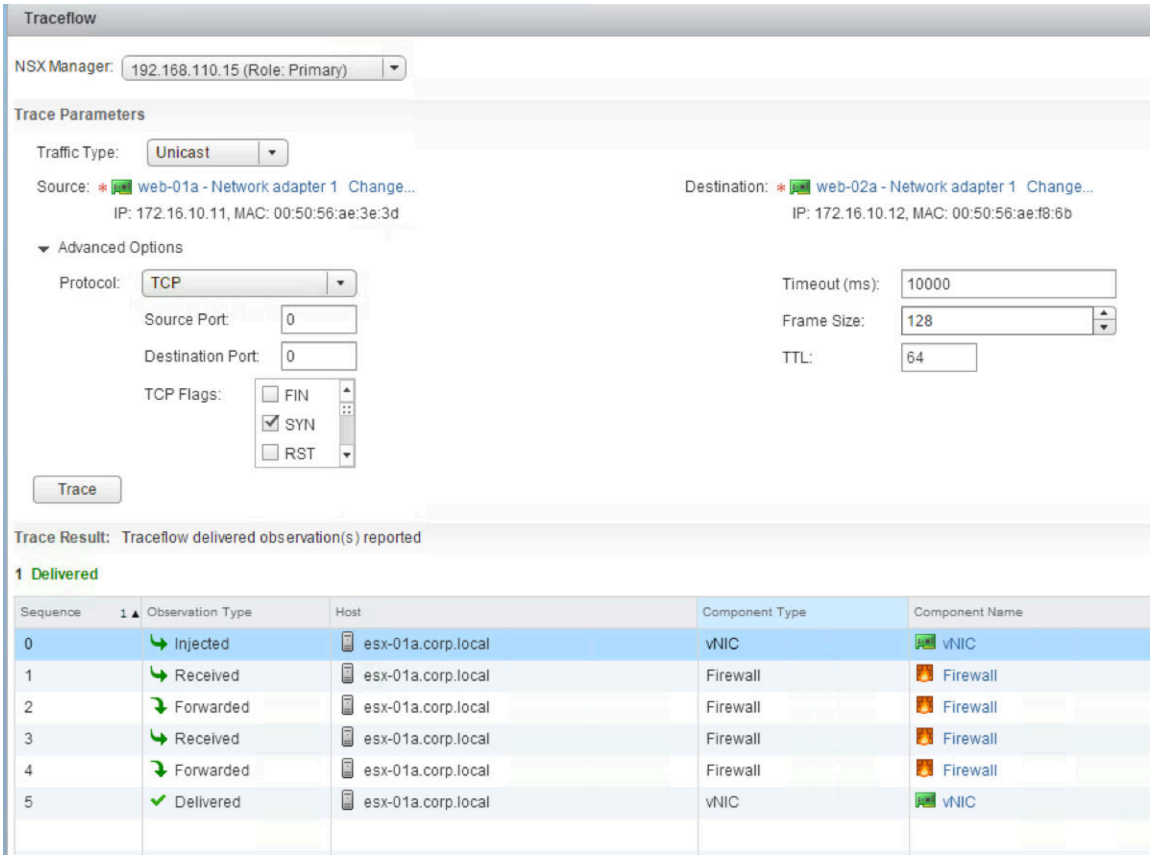
Both the source and destination IP addresses are required to make the IP packet valid. In the case of multicast, the MAC address is deduced from the IP address.

7 Configure other required and optional settings.

8 Click **Trace**.

Example: Scenarios

The following example shows a Layer 2 traceflow involving two VMs that are running on a single ESXi host. The two VMs are connected to a single logical switch.



Traceflow

NSX Manager: 192.168.110.15 (Role: Primary)

Trace Parameters

Traffic Type: Unicast

Source: * web-01a - Network adapter 1 Change...
IP: 172.16.10.11, MAC: 00:50:56:ae:3e:3d

Destination: * web-02a - Network adapter 1 Change...
IP: 172.16.10.12, MAC: 00:50:56:ae:f8:6b

Advanced Options

Protocol: TCP

Source Port: 0

Destination Port: 0

TCP Flags: □ FIN ☑ SYN □ RST

Timeout (ms): 10000

Frame Size: 128

TTL: 64

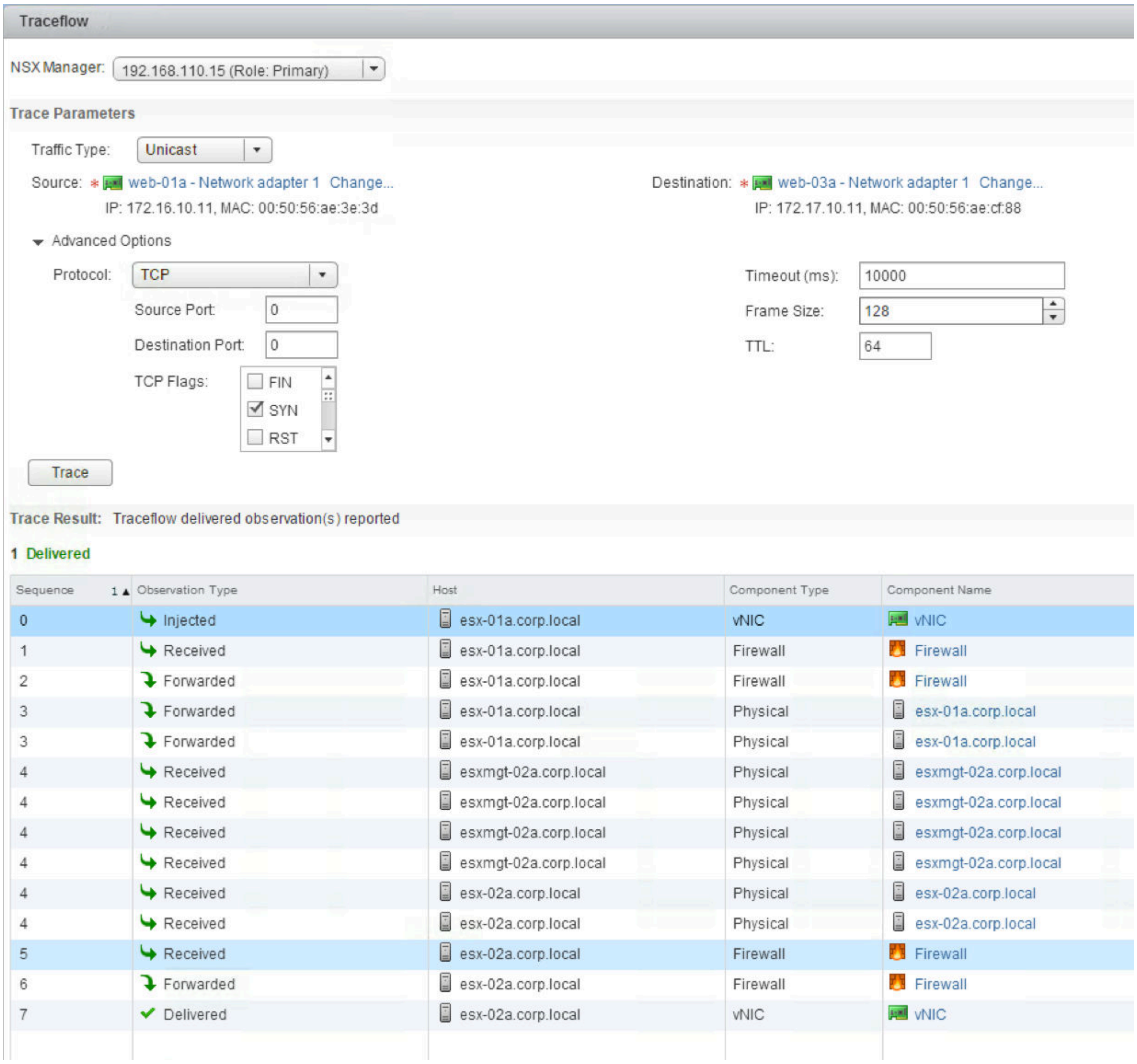
Trace

Trace Result: Traceflow delivered observation(s) reported

1 Delivered

Sequence	Observation Type	Host	Component Type	Component Name
0	Injected	esx-01a.corp.local	vNIC	vNIC
1	Received	esx-01a.corp.local	Firewall	Firewall
2	Forwarded	esx-01a.corp.local	Firewall	Firewall
3	Received	esx-01a.corp.local	Firewall	Firewall
4	Forwarded	esx-01a.corp.local	Firewall	Firewall
5	✓ Delivered	esx-01a.corp.local	vNIC	vNIC

The following example shows a Layer 2 traceflow involving two VMs that are running on two different ESXi hosts. The two VMs are connected to a single logical switch.



Traceflow

NSX Manager: 192.168.110.15 (Role: Primary)

Trace Parameters

Traffic Type: Unicast

Source: * web-01a - Network adapter 1 Change...
IP: 172.16.10.11, MAC: 00:50:56:ae:3e:3d

Destination: * web-03a - Network adapter 1 Change...
IP: 172.17.10.11, MAC: 00:50:56:ae:cf:88

Advanced Options

Protocol: TCP

Source Port: 0

Destination Port: 0

TCP Flags: □ FIN ☑ SYN □ RST

Timeout (ms): 10000

Frame Size: 128

TTL: 64

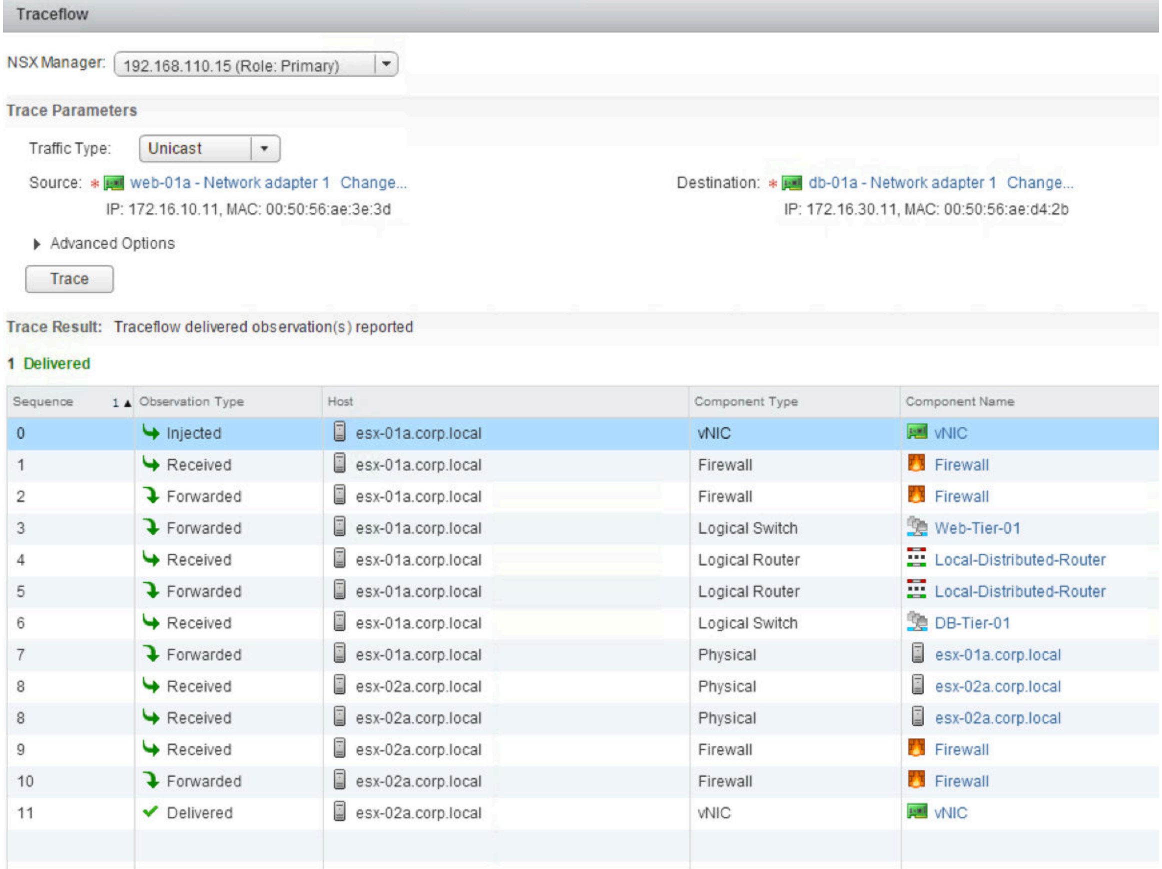
Trace

Trace Result: Traceflow delivered observation(s) reported

1 Delivered

Sequence	Observation Type	Host	Component Type	Component Name
0	Injected	esx-01a.corp.local	vNIC	vNIC
1	Received	esx-01a.corp.local	Firewall	Firewall
2	Forwarded	esx-01a.corp.local	Firewall	Firewall
3	Forwarded	esx-01a.corp.local	Physical	esx-01a.corp.local
3	Forwarded	esx-01a.corp.local	Physical	esx-01a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esx-02a.corp.local	Physical	esx-02a.corp.local
4	Received	esx-02a.corp.local	Physical	esx-02a.corp.local
5	Received	esx-02a.corp.local	Firewall	Firewall
6	Forwarded	esx-02a.corp.local	Firewall	Firewall
7	Delivered	esx-02a.corp.local	vNIC	vNIC

The following example shows a Layer 3 traceflow. The two VMs are connected to two different logical switches that are separated by a logical router.



Traceflow

NSX Manager: 192.168.110.15 (Role: Primary)

Trace Parameters

Traffic Type: Unicast

Source: * web-01a - Network adapter 1 Change...
IP: 172.16.10.11, MAC: 00:50:56:ae:3e:3d

Destination: * db-01a - Network adapter 1 Change...
IP: 172.16.30.11, MAC: 00:50:56:ae:d4:2b

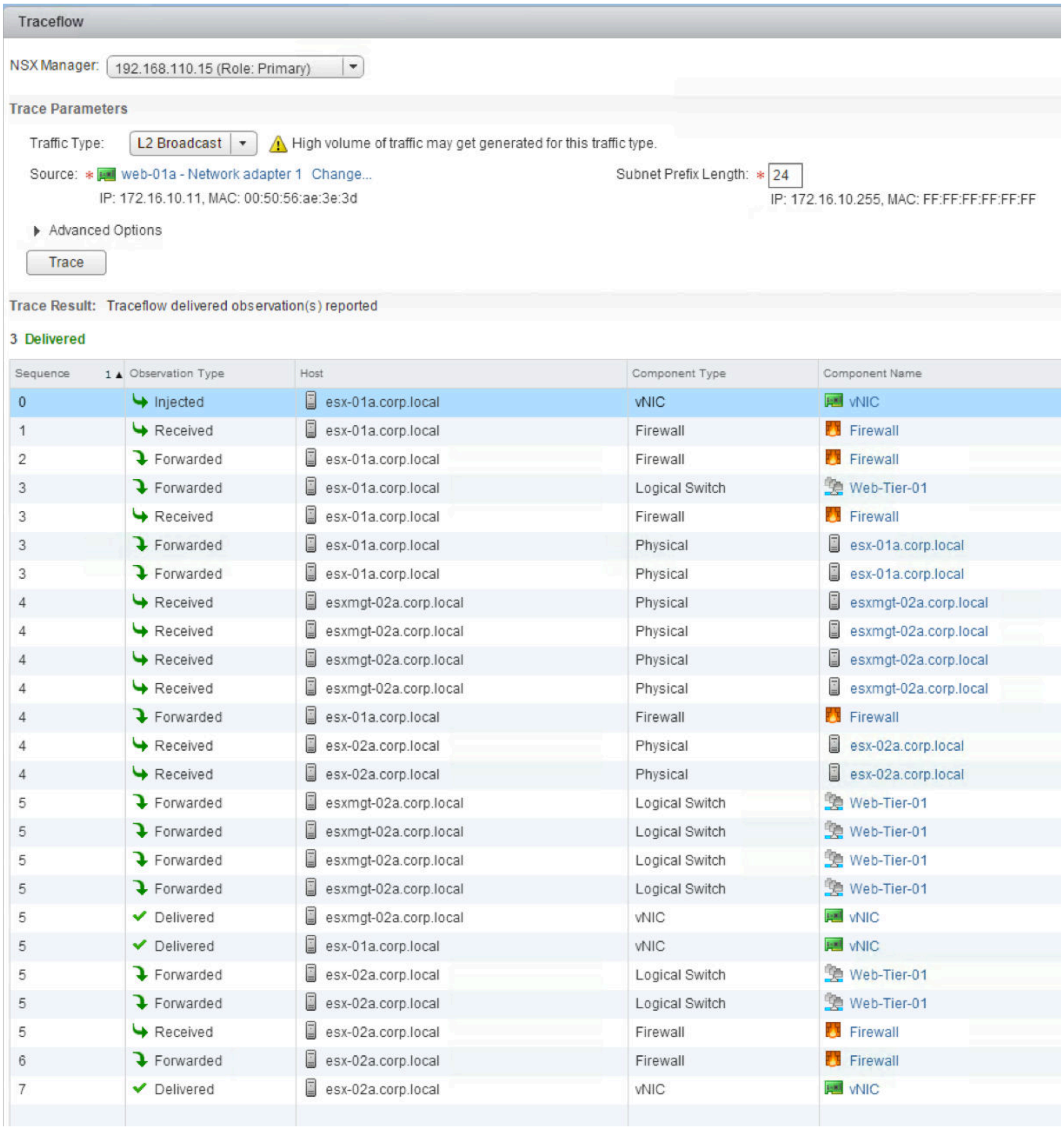
Advanced Options

Trace Result: Traceflow delivered observation(s) reported

1 Delivered

Sequence 1 ▲	Observation Type	Host	Component Type	Component Name
0	Injected	esx-01a.corp.local	vNIC	vNIC
1	Received	esx-01a.corp.local	Firewall	Firewall
2	Forwarded	esx-01a.corp.local	Firewall	Firewall
3	Forwarded	esx-01a.corp.local	Logical Switch	Web-Tier-01
4	Received	esx-01a.corp.local	Logical Router	Local-Distributed-Router
5	Forwarded	esx-01a.corp.local	Logical Router	Local-Distributed-Router
6	Received	esx-01a.corp.local	Logical Switch	DB-Tier-01
7	Forwarded	esx-01a.corp.local	Physical	esx-01a.corp.local
8	Received	esx-02a.corp.local	Physical	esx-02a.corp.local
8	Received	esx-02a.corp.local	Physical	esx-02a.corp.local
9	Received	esx-02a.corp.local	Firewall	Firewall
10	Forwarded	esx-02a.corp.local	Firewall	Firewall
11	✓ Delivered	esx-02a.corp.local	vNIC	vNIC

The following example shows a broadcast traceflow in a deployment that has three VMs connected to a single logical switch. Two of the VMs are on one host (esx-01a), and the third is on another host (esx-02a). The broadcast is sent from one of the VMs on host 192.168.210.53.



Traceflow

NSX Manager: 192.168.110.15 (Role: Primary)

Trace Parameters

Traffic Type: L2 Broadcast ⚠ High volume of traffic may get generated for this traffic type.

Source: * web-01a - Network adapter 1 Change...
IP: 172.16.10.11, MAC: 00:50:56:ae:3e:3d

Subnet Prefix Length: * 24
IP: 172.16.10.255, MAC: FF:FF:FF:FF:FF:FF

▸ Advanced Options

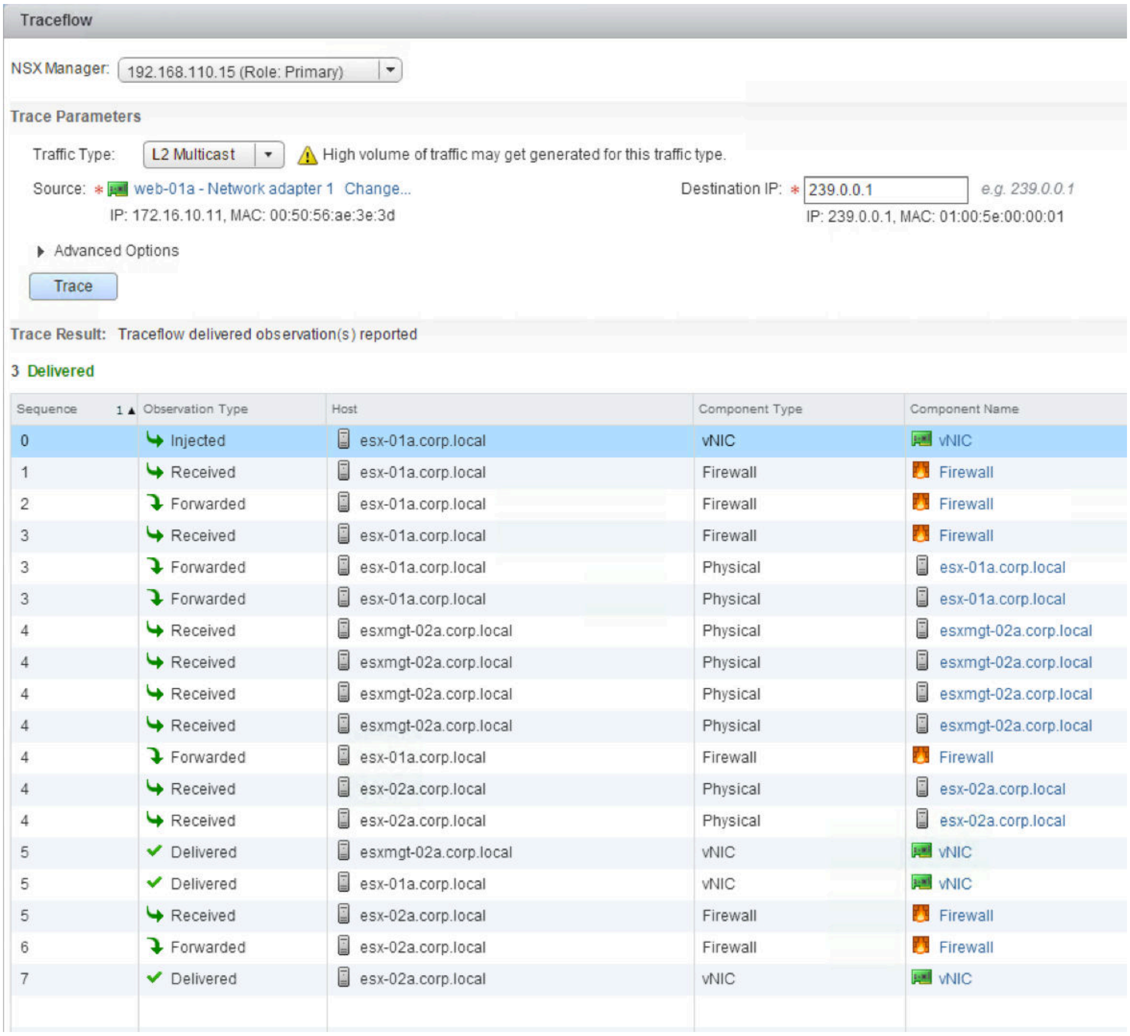
Trace

Trace Result: Traceflow delivered observation(s) reported

3 Delivered

Sequence 1 ▲	Observation Type	Host	Component Type	Component Name
0	Injected	esx-01a.corp.local	vNIC	vNIC
1	Received	esx-01a.corp.local	Firewall	Firewall
2	Forwarded	esx-01a.corp.local	Firewall	Firewall
3	Forwarded	esx-01a.corp.local	Logical Switch	Web-Tier-01
3	Received	esx-01a.corp.local	Firewall	Firewall
3	Forwarded	esx-01a.corp.local	Physical	esx-01a.corp.local
3	Forwarded	esx-01a.corp.local	Physical	esx-01a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Forwarded	esx-01a.corp.local	Firewall	Firewall
4	Received	esx-02a.corp.local	Physical	esx-02a.corp.local
4	Received	esx-02a.corp.local	Physical	esx-02a.corp.local
5	Forwarded	esxmgt-02a.corp.local	Logical Switch	Web-Tier-01
5	Forwarded	esxmgt-02a.corp.local	Logical Switch	Web-Tier-01
5	Forwarded	esxmgt-02a.corp.local	Logical Switch	Web-Tier-01
5	Forwarded	esxmgt-02a.corp.local	Logical Switch	Web-Tier-01
5	✓ Delivered	esxmgt-02a.corp.local	vNIC	vNIC
5	✓ Delivered	esx-01a.corp.local	vNIC	vNIC
5	Forwarded	esx-02a.corp.local	Logical Switch	Web-Tier-01
5	Forwarded	esx-02a.corp.local	Logical Switch	Web-Tier-01
5	Received	esx-02a.corp.local	Firewall	Firewall
6	Forwarded	esx-02a.corp.local	Firewall	Firewall
7	✓ Delivered	esx-02a.corp.local	vNIC	vNIC

The following example shows what happens when multicast traffic is sent in a deployment that has multicast configured.



Traceflow

NSX Manager: 192.168.110.15 (Role: Primary)

Trace Parameters

Traffic Type: L2 Multicast ⚠ High volume of traffic may get generated for this traffic type.

Source: * web-01a - Network adapter 1 Change...
IP: 172.16.10.11, MAC: 00:50:56:ae:3e:3d

Destination IP: * 239.0.0.1 e.g. 239.0.0.1
IP: 239.0.0.1, MAC: 01:00:5e:00:00:01

Advanced Options

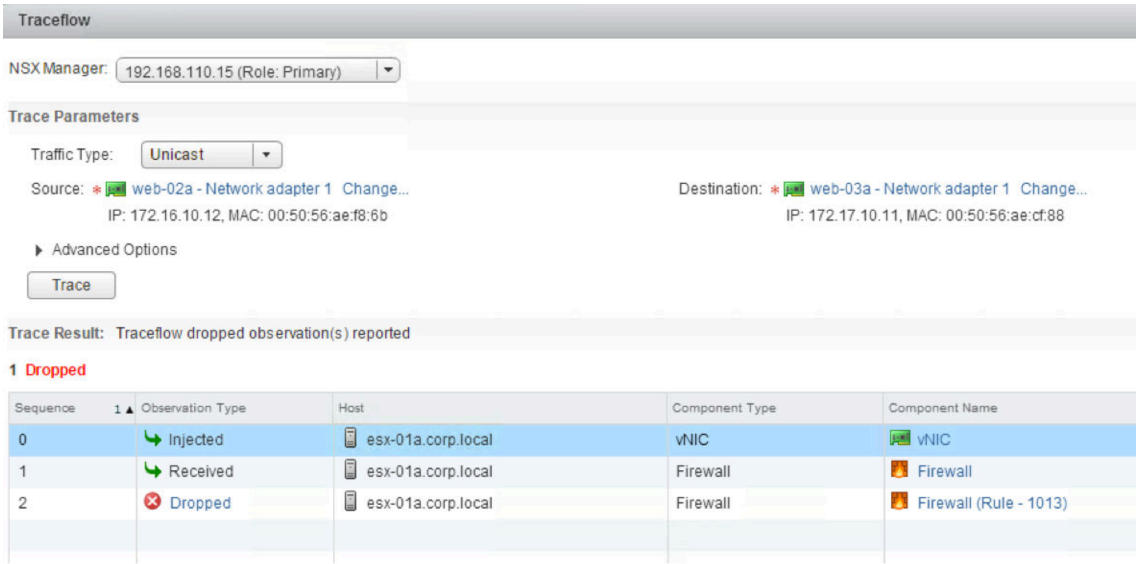
Trace

Trace Result: Traceflow delivered observation(s) reported

3 Delivered

Sequence 1 ▲	Observation Type	Host	Component Type	Component Name
0	Injected	esx-01a.corp.local	vNIC	vNIC
1	Received	esx-01a.corp.local	Firewall	Firewall
2	Forwarded	esx-01a.corp.local	Firewall	Firewall
3	Received	esx-01a.corp.local	Firewall	Firewall
3	Forwarded	esx-01a.corp.local	Physical	esx-01a.corp.local
3	Forwarded	esx-01a.corp.local	Physical	esx-01a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Received	esxmgt-02a.corp.local	Physical	esxmgt-02a.corp.local
4	Forwarded	esx-01a.corp.local	Firewall	Firewall
4	Received	esx-02a.corp.local	Physical	esx-02a.corp.local
4	Received	esx-02a.corp.local	Physical	esx-02a.corp.local
5	✓ Delivered	esxmgt-02a.corp.local	vNIC	vNIC
5	✓ Delivered	esx-01a.corp.local	vNIC	vNIC
5	Received	esx-02a.corp.local	Firewall	Firewall
6	Forwarded	esx-02a.corp.local	Firewall	Firewall
7	✓ Delivered	esx-02a.corp.local	vNIC	vNIC

The following example shows what happens when a traceflow is dropped because of a distributed firewall rule that blocks ICMP traffic sent to the destination address. Notice that the traffic never leaves the original host, even though the destination VM is on another host.



Traceflow

NSX Manager: 192.168.110.15 (Role: Primary)

Trace Parameters

Traffic Type: Unicast

Source: * web-02a - Network adapter 1 Change...
IP: 172.16.10.12, MAC: 00:50:56:ae:f8:6b

Destination: * web-03a - Network adapter 1 Change...
IP: 172.17.10.11, MAC: 00:50:56:ae:cf:88

Advanced Options

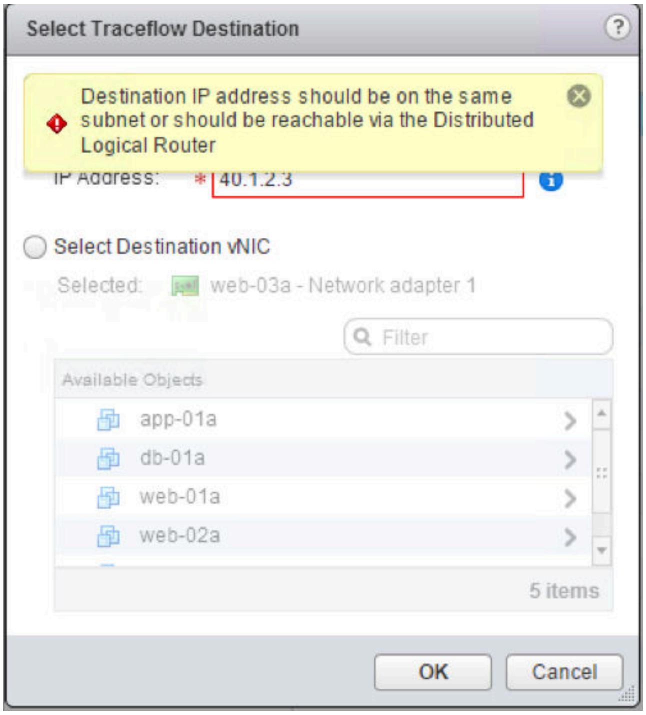
Trace

Trace Result: Traceflow dropped observation(s) reported

1 Dropped

Sequence 1 ▲	Observation Type	Host	Component Type	Component Name
0	Injected	esx-01a.corp.local	vNIC	vNIC
1	Received	esx-01a.corp.local	Firewall	Firewall
2	⊗ Dropped	esx-01a.corp.local	Firewall	Firewall (Rule - 1013)

The following example shows what happens when a traceflow destination is on the other side of an edge services gateway, such as an IP address on the Internet or any internal destination that must be routed through the edge services gateway. The traceflow is not allowed, by design, because traceflow is supported for destinations that are either on the same subnet or are reachable through distributed logical routers (DLRs).



Select Traceflow Destination

Destination IP address should be on the same subnet or should be reachable via the Distributed Logical Router

IP Address: * 40.1.2.3

○ Select Destination vNIC

Selected: web-03a - Network adapter 1

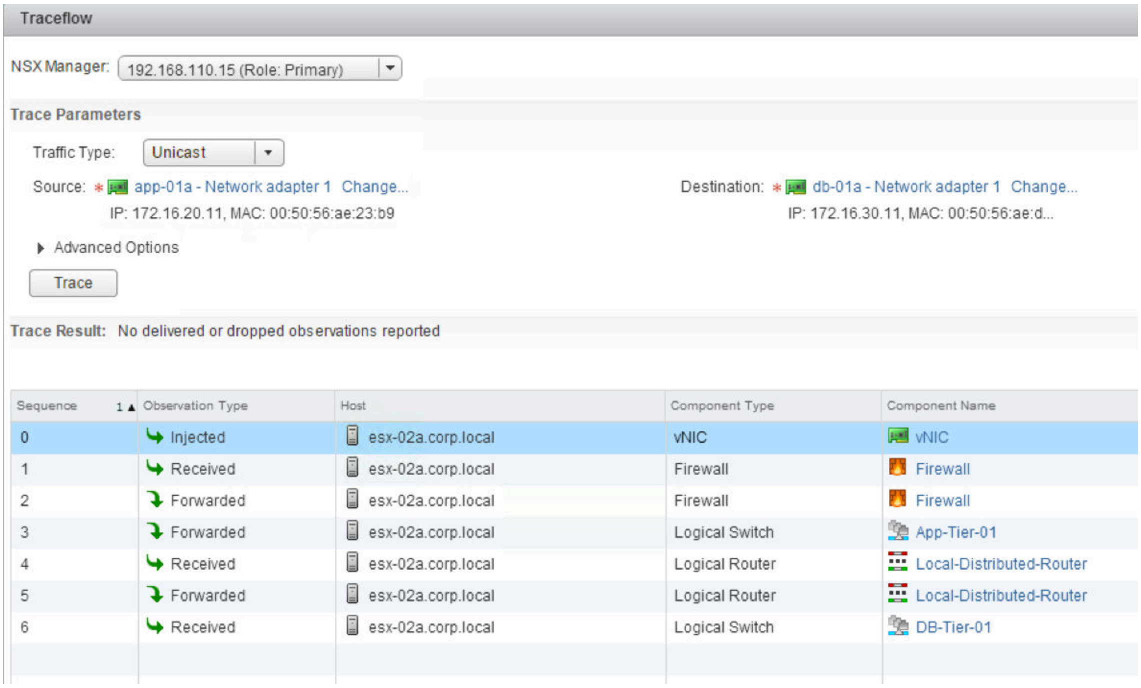
Filter

Available Objects

- app-01a
- db-01a
- web-01a
- web-02a

5 items

The following example shows what happens when the traceflow destination is a VM that is on another subnet and is powered off.



Traceflow

NSX Manager: 192.168.110.15 (Role: Primary)

Trace Parameters

Traffic Type: Unicast

Source: * app-01a - Network adapter 1 Change...
IP: 172.16.20.11, MAC: 00:50:56:ae:23:b9

Destination: * db-01a - Network adapter 1 Change...
IP: 172.16.30.11, MAC: 00:50:56:ae:d...

Advanced Options

Trace Result: No delivered or dropped observations reported

Sequence 1 ▲	Observation Type	Host	Component Type	Component Name
0	Injected	esx-02a.corp.local	vNIC	vNIC
1	Received	esx-02a.corp.local	Firewall	Firewall
2	Forwarded	esx-02a.corp.local	Firewall	Firewall
3	Forwarded	esx-02a.corp.local	Logical Switch	App-Tier-01
4	Received	esx-02a.corp.local	Logical Router	Local-Distributed-Router
5	Forwarded	esx-02a.corp.local	Logical Router	Local-Distributed-Router
6	Received	esx-02a.corp.local	Logical Switch	DB-Tier-01

NSX Routing

4

NSX has two types of routing subsystems, optimised for two key needs.

The NSX routing subsystems are:

- Routing within the logical space, also known as “East – West” routing, provided by the Distributed Logical Router (DLR);
- Routing between the physical and logical space, also known as “North – South” routing, provided by the Edge Services Gateways (ESG).

Both provide options for horizontal scaling.

You can scale-out distributed E-W routing via the DLR.

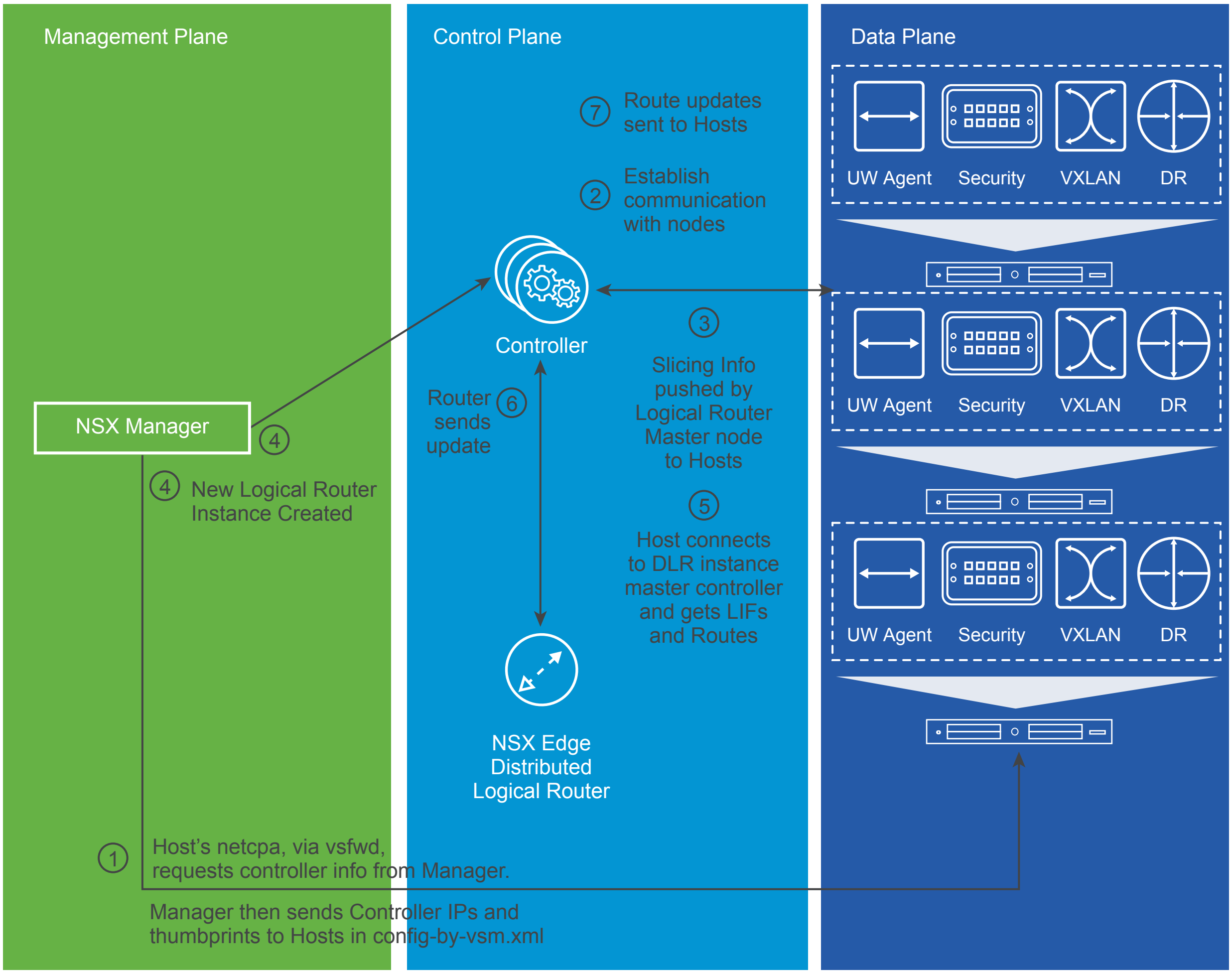
The DLR supports running a single dynamic routing protocol at a time (OSPF or BGP), while the ESG supports running both routing protocols at the same time. The reason for this is the DLR is designed to be s a “stub” router, with a single path out, which means more advanced routing configurations are typically not required.

Both the DLR and the ESG support having a combination of static and dynamic routes.

Both the DLR and the ESG support ECMP routes.

Both provide L3 domain separation, meaning that each instance of a Distributed Logical Router or an Edge Services Gateway has its own L3 configuration, similar to an L3VPN VRF.

Figure 4-1. The Creation of a DLR


Management Plane
Control Plane
Data Plane
7 Route updates sent to Hosts
2 Establish communication with nodes
UW Agent
Security
VXLAN
DR
Controller
3 Slicing Info pushed by Logical Router Master node to Hosts
Router sends update 6
NSX Manager
4
4 New Logical Router Instance Created
5 Host connects to DLR instance master controller and gets LIFs and Routes
NSX Edge Distributed Logical Router
1 Host's netcpa, via vsfwd, requests controller info from Manager.
Manager then sends Controller IPs and thumbprints to Hosts in config-by-vsm.xml

This chapter includes the following topics:

- Understanding the Distributed Logical Router
- Understanding Routing Provided by the Edge Services Gateway
- ECMP Packet Flow
- NSX Routing: Prerequisites and Considerations
- DLR and ESG UIs
- New NSX Edge (DLR)
- Typical ESG and DLR UI Operations
- Troubleshooting NSX Routing

Understanding the Distributed Logical Router

The DLR is optimised for forwarding in the logical space between VMs, on VXLAN-backed or VLAN-backed portgroups.

The DLR has the following properties:

- High performance, low overhead first-hop routing:
- Scales linearly with the number of hosts
- Supports 8-way ECMP on uplink
- Up to 1,000 DLR instances per host
- Up to 999 logical interfaces (LIFs) on each DLR (8 x uplink + 991 internal) + 1 x management
- Up to 10,000 LIFs per host distributed across all DLR instances (not enforced by NSX Manager)

Keep in mind the following caveats:

- Cannot connect more than one DLR to any given VLAN or VXLAN.
- Cannot run more than one routing protocol on each DLR.
- If OSPF is used, cannot run it on more than one DLR uplink.
- To route between VXLAN and VLAN, the transport zone must span single DVS.

The DLR's design at a high level is analogous to a modular router chassis, in the following ways:

- ESXi hosts are like line cards:
  - They have ports with connected end stations (VMs).
  - This is where the forwarding decisions are made.
- The DLR Control VM is like a Route Processor Engine:
  - It runs dynamic routing protocols to exchange routing information with the rest of the network.
  - It computes forwarding tables for “line cards” based on the configuration of interfaces, static routes, and dynamic routing information.
  - It programs these forwarding tables into the “line cards” (via the Controller Cluster, to enable scale and resiliency).
- The physical network connecting ESXi hosts together is like a backplane:
  - It carries VLAN-encapsulated or VXLAN-encapsulated data between the “line cards.”

High-Level DLR Packet Flow

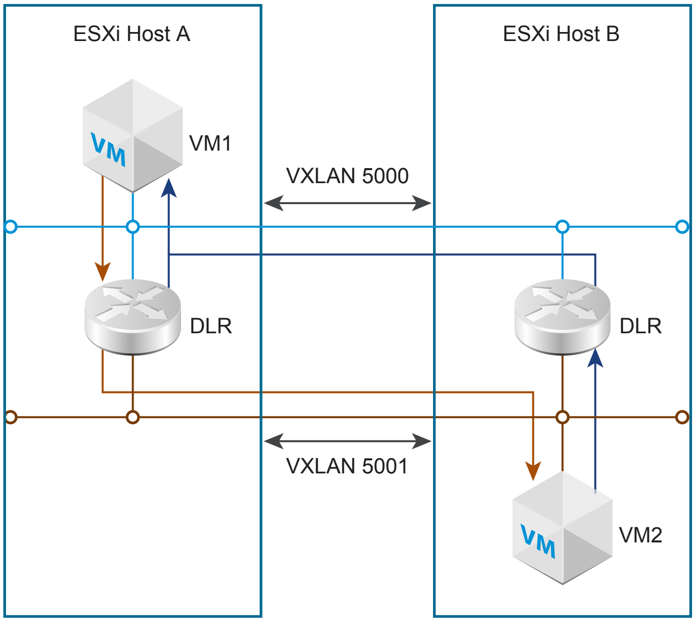
Each ESXi host has its own copy of each configured DLR instance. Each DLR instance has its own unique set of tables containing the information needed to forward packets. This information is synchronized across all hosts where this DLR instance exists. Instances of an individual DLR across different hosts have exactly the same information.

Routing is always handled by a DLR instance on the same host where the source VM is running. This means that when source and destination VMs are on different hosts, the DLR instance that provides routing between them sees packets only in one direction, from source VM to destination. Return traffic is only seen by the corresponding instance of the same DLR on the destination VM's host.

After the DLR has completed routing, delivery to the final destination is the responsibility of the DVS via L2 – VXLAN or VLAN if the source and destination VMs are on different hosts, or by the DVS locally if they are on the same host.

Figure 4-2 illustrates data flow between two VMs, VM1 and VM2, running on different hosts and connected to two different Logical Switches, VXLAN 5000 and VXLAN 5001.

Figure 4-2. High-Level DLR Packet Flow



Packet flow (skipping ARP resolution):

1 VM1 sends a packet toward VM2, which is addressed to VM1's gateway for VM2's subnet (or default). This gateway is a VXLAN 5000 LIF on the DLR.

2 The DVS on ESXi Host A delivers the packet to the DLR on that host, where the lookup is performed, and the egress LIF is determined (in this case – VXLAN 5001 LIF).

3 The packet is then sent out of that destination LIF, which essentially returns the packet to the DVS, but on a different Logical Switch (5001).

4 The DVS then performs L2 delivery of that packet to the destination host (ESXi Host B), where the DVS will forward the packet to VM2.

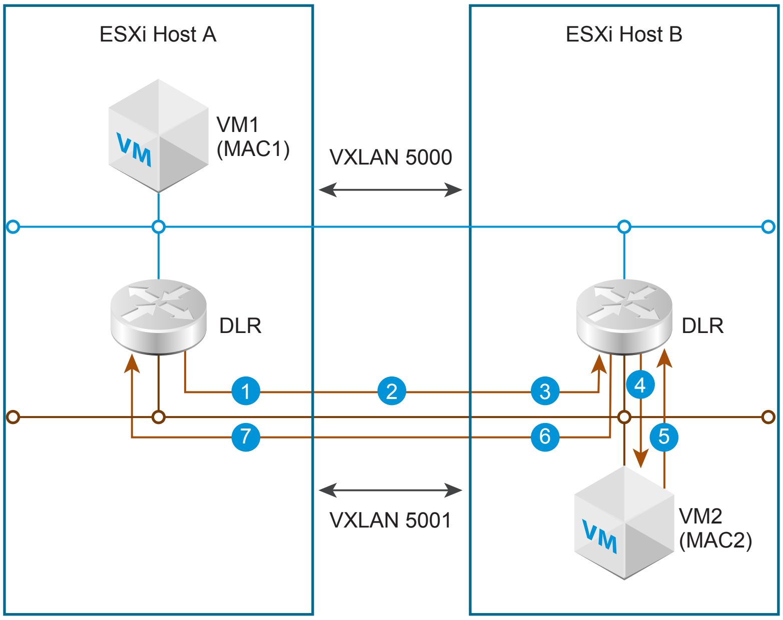
Return traffic will follow in the same order, where traffic from VM2 is forwarded to the DLR instance on ESXi Host B, and then delivered via L2 on VXLAN 5000.

DLR ARP Resolution Process

Before traffic from VM1 can reach VM2, the DLR needs to learn VM2's MAC address. After learning VM2's MAC address, the DLR can create the correct L2 headers for the outbound packets.

Figure 4-3 shows the DLR's ARP resolution process.

Figure 4‑3. DLR ARP Process



To learn the MAC address, the DLR follows these steps:

1 The DLR instance on Host A generates an ARP request packet, with SRC MAC = vMAC, and DST MAC = Broadcast. The VXLAN module on Host A finds all VTEPs on the egress VXLAN 5001, and sends each one a copy of that broadcast frame.

2 As the frame leaves the host via the VXLAN encapsulation process, the SRC MAC is changed from vMAC to pMAC A, so that return traffic can find the originating DLR instance on Host A. Frame now is SRC MAC = pMAC A, and DST MAC = Broadcast.

3 As the frame is received and decapsulated on Host B, it is examined and found to be sourced from the IP address that matches the local DLR instance's LIF on VXLAN 5001. This flags the frame as abrequest to perform the proxy ARP function. The DST MAC is changed from Broadcast to vMAC so that the frame can reach the local DLR instance.

4 The local DLR instance on Host B receives the ARP Request frame, SRC MAC = pMAC A, DST MAC = vMAC, and sees its own LIF IP address requesting this. It saves the SRC MAC, and generates a new ARP Request packet, SRC MAC = vMAC, DST MAC = Broadcast. This frame is tagged as "DVS Local" to prevent it from being flooded via the dvUplink. The DVS delivers the frame to VM2.

5 VM2 sends an ARP Reply, SRC MAC = MAC2, DST MAC = vMAC. The DVS delivers it to the local DLR instance.

6 The DLR instance on Host B replaces DST MAC with the pMAC A saved at from step 4, and sends the packet back to the DVS for delivery back to Host A.

7 After the ARP Reply reaches Host A, DST MAC is changed to vMAC, and the ARP Reply frame with SRC MAC = MAC2 and DST MAC = vMAC reaches the DLR instance on Host A.

The ARP resolution process is complete, and the DLR instance on Host A can now start sending traffic to VM2.

Understanding Routing Provided by the Edge Services Gateway

The second subsystem of NSX Routing is provided by Edge Services Gateway.

The ESG is essentially a router in a virtual machine. It is delivered in an appliance-like form factor with four sizes, with its complete lifecycle managed by the NSX Manager. The ESG's primary use case is as a perimeter router, where it is deployed between multiple DLRs and between the physical world and the virtualized network.

The ESG has the following properties:

- Each ESG can have up to 10 vNIC interfaces, or 200 trunk sub-interfaces.
- Each ESG supports 8-way ECMP for path redundancy and scalability.

ECMP Packet Flow

Suppose two ESGs are deployed to provide a DLR instance with 2-way ECMP uplinks with the physical environment.

Figure 4-4 shows the ESG and DLR packet flow when equal-cost multipath (ECMP) routing is enabled between two ESGs and the physical infrastructure.

VM1 thus has access to 2x bi-directional throughput compared with a deployment with a single ESG.

VM1 is connected to a Logical Switch with the VNI 5000.

The DLR has two LIFs – Internal on VNI 5000, and Uplink on VNI 5001.

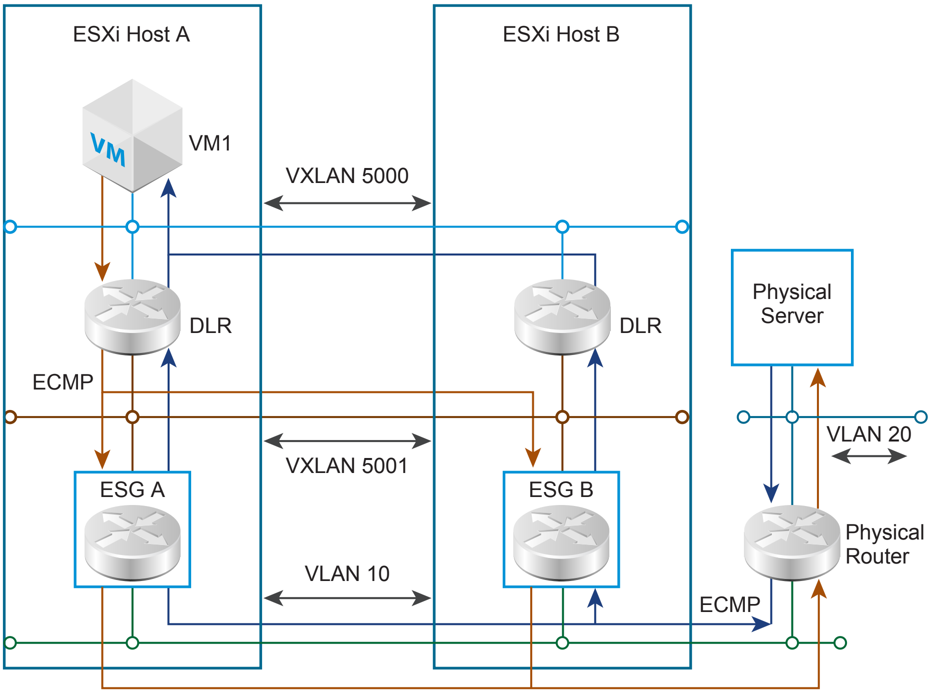
The DLR has ECMP enabled and is receiving equal cost routes toward the IP subnet of VLAN 20 from a pair of ESGs, ESG A and ESG B via a dynamic routing protocol (BGP or OSPF).

The two ESGs are connected to a VLAN-backed dvPortgroup associated with VLAN 10, where a physical router that provides connectivity to VLAN 20 is also connected.

The ESGs receive external routes for VLAN 20, via a dynamic routing protocol from the physical router.

The physical router in exchange learns about the IP subnet associated with VXLAN 5000 from both ESGs, and performs ECMP load balancing for the traffic toward VMs in that subnet.

Figure 4-4. High-Level ESG and DLR Packet Flow with ECMP



The DLR can receive up to eight equal-cost routes and balance traffic across the routes. ESG A and ESG B in the diagram provide two equal-cost routes.

ESGs can do ECMP routing toward the physical network, assuming multiple physical routers are present. For simplicity, the diagram shows a single physical router.

There is no need for ECMP to be configured on ESGs toward the DLR, because all DLR LIFs are “local” on the same host where ESG resides. There would be no additional benefit provided by configuring multiple uplink interfaces on a DLR.

In situations where more North-South bandwidth is required, multiple ESGs can be placed on different ESXi hosts to scale up to ~80Gbps with 8 x ESGs.

The ECMP packet flow (not including ARP resolution):

1 VM1 sends a packet to the physical server, which is sent to VM1’s IP gateway (which is a DLR LIF) on ESXi Host A.

2 The DLR performs a route lookup for the IP of the physical server, and finds that it is not directly connected, but matches two ECMP routes received from ESG A and ESG B.

3 The DLR calculates an ECMP hash, and decides on a next hop, which could be either ESG A or ESG B, and sends the packet out the VXLAN 5001 LIF.

4 The DVS delivers the packet to the selected ESG.

5 The ESG performs the routing lookup and finds that the physical server’s subnet is accessible via the physical router’s IP address on VLAN 10, which is directly connected to one of ESG’s interfaces.

6 The packet is sent out through the DVS, which passes it on to the physical network after tagging it with the correct 801.Q tag with VLAN ID 10.

7 The packet travels through the physical switching infrastructure to reach the physical router, which performs a lookup to find that the physical server is directly connected to an interface on VLAN 20.

8 The physical router sends the packet to the physical server.

On the way back:

1 The physical server sends the packet to VM1, with the physical router as the next hop.

2 The physical router performs a lookup for VM1's subnet, and sees two equal-cost paths to that subnet with the next hops, ESG A's and ESG B's VLAN 10 interface, respectively.

3 The physical router selects one of the paths and sends the packet toward the corresponding ESG.

4 The physical network delivers the packet to the ESXi host where the ESG resides, and delivers it to DVS, which decapsulates the packet and forwards it on the dvPortgroup associated with VLAN 10 to the ESG.

5 The ESG performs a routing lookup and finds that VM1's subnet is accessible via its interface associated with VXLAN 5001 with the next hop being DLR's uplink interface IP address.

6 The ESG sends the packet to the DLR instance on the same host as the ESG.

7 The DLR performs a routing lookup to find that VM1 is available via its VXLAN 5000 LIF.

8 The DLR sends the packet out its VXLAN 5000 LIF to the DVS, which performs the final delivery.

NSX Routing: Prerequisites and Considerations

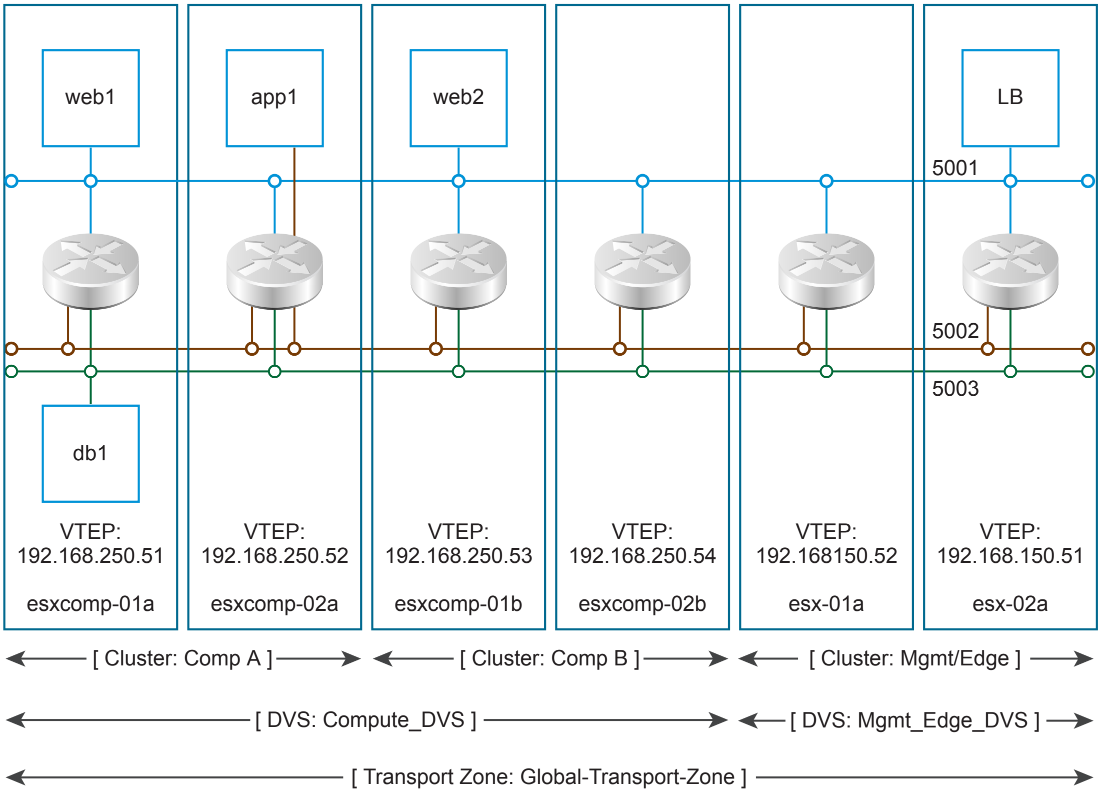
The DLR and the ESG rely on the DVS to provide L2 forwarding services for dvPortgroups (both VXLAN and VLAN based) for end-to end connectivity to work.

This means L2 that forwarding services that are connected to DLR or ESG must be configured and operational. In the NSX installation process, these services are provided by “Host Preparation” and “Logical Network Preparation.”

When creating transport zones on multi-cluster DVS configurations, make sure that all clusters in the selected DVS are included under the transport zone. This ensures that the DLR is available on all clusters where DVS dvPortgroups are available.

When a transport zone is aligned with DVS boundary, the DLR instance is created correctly.

Figure 4-5. Transport Zone Correctly Aligned to DVS Boundary



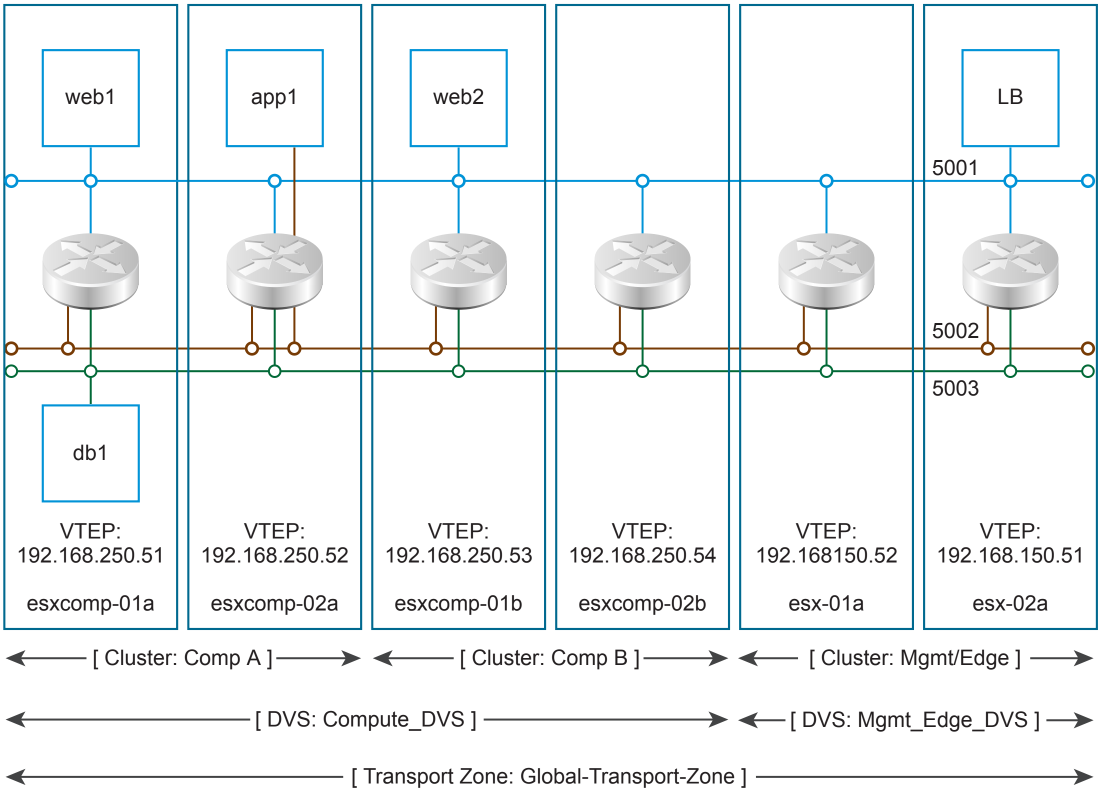
When a transport zone is not aligned to the DVS boundary, the scope of logical switches (5001, 5002 and 5003) and the DLR instances that these logical switches are connected to becomes disjointed, causing VMs in cluster Comp A to have no access to DLR LIFs.

In the diagram above, DVS “Compute_DVS” covers two clusters, “Comp A” and “Comp B”. The “Global-Transport-Zone” includes both “Comp A” and “Comp B.”

This results in correct alignment between the scope of Logical Switches (5001, 5002, and 5003), and the DLR instance created on all hosts in all clusters where these Logical Switches are present.

Now, let’s look at an alternative situation, where the Transport Zone was not configured to include cluster “Comp A”:

Figure 4-6. Transport Zone Misaligned with DVS Boundary



In this case, VMs running on cluster “Comp A” have full access to all logical switches. This is because logical switches are represented by dvPortgoups on hosts, and dvProtgroups are a DVS-wide construct. In our sample environment, “Compute_DVS” covers both “Comp A” an “Comp B."

DLR instances, however, are created in strict alignment with the transport zone scope, which means no DLR instance will be created on hosts in “Comp A."

As the result, VM “web1” will be able to reach VMs “web2” and “LB” because they are on the same logical switch, but VMs “app1” and “db1” will not be able to communicate with anything.

The DLR relies on the Controller Cluster to function, while the ESG does not. Make sure that the Controller Cluster is up and available before creating or changing a DLR configuration.

If the DLR is to be connected to VLAN dvPortgroups, ensure that ESXi hosts with the DLR configured can reach each other on UDP/6999 for DLR VLAN-based ARP proxy to work.

Considerations:

- A given DLR instance cannot be connected to logical switches that exist in different transport zones. This is to ensure all logical switches and DLR instances are aligned.
- The DLR cannot be connected to VLAN-backed portgroups, if that DLR is connected to logical switches spanning more than one DVS. As above, this is to ensure correct alignment of DLR instances with logical switches and dvPortgroups across hosts.

- When selecting placement of the DLR Control VM, avoid placing it on the same host as one or more of its upstream ESGs by using DRS anti-affinity rules if they are in the same cluster. This is to reduce the impact of host failure on DLR forwarding.
- OSPF can be enabled only on a single Uplink (but supports multiple adjacencies). BGP, on other hand, can be enabled on multiple Uplink interfaces, where it is necessary.

DLR and ESG UIs

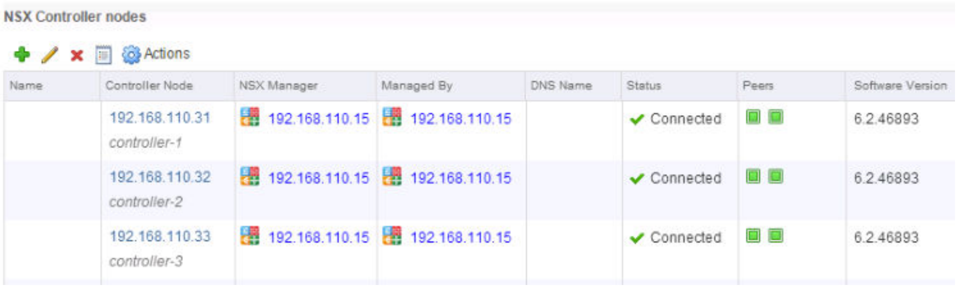
The DLR and ESG UIs provide indicators of the system working state.

NSX Routing UI

The vSphere Web Client UI provides two major sections relevant to NSX routing.

These include the L2 and control-plane infrastructure dependencies and the routing subsystem configuration.

NSX distributed routing requires functions that are provided by the Controller Cluster. The following screen shot shows a Controller Cluster in a healthy state.

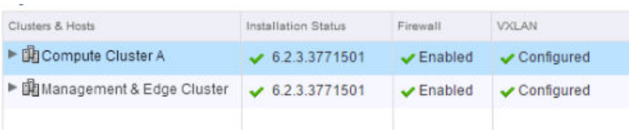


Name	Controller Node	NSX Manager	Managed By	DNS Name	Status	Peers	Software Version
	192.168.110.31 *controller-1*	192.168.110.15	192.168.110.15		✓ Connected	🟩 🟩	6.2.46893
	192.168.110.32 *controller-2*	192.168.110.15	192.168.110.15		✓ Connected	🟩 🟩	6.2.46893
	192.168.110.33 *controller-3*	192.168.110.15	192.168.110.15		✓ Connected	🟩 🟩	6.2.46893

Things to note:

- There are three controllers deployed.
- The “Status” for all controllers is “Connected”.
- The software version for all controllers is the same.
- Each controller node has two peers.

Host kernel modules for distributed routing are installed and configured as part of VXLAN configuration on the host. This means distributed routing requires that ESXi hosts are prepared and VXLAN is configured on them.



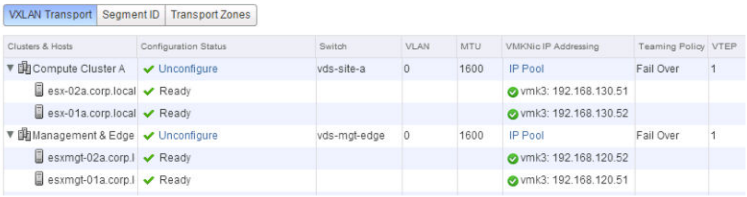
Clusters & Hosts	Installation Status	Firewall	VXLAN
▸ Compute Cluster A	✓ 6.2.3.3771501	✓ Enabled	✓ Configured
▸ Management & Edge Cluster	✓ 6.2.3.3771501	✓ Enabled	✓ Configured

Things to note:

- “Installation Status” is green.

- “VXLAN” is “Configured."

Makes sure that VXLAN transport components are correctly configured.



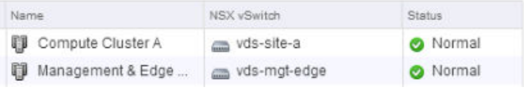
VXLAN Transport | Segment ID | Transport Zones

Clusters & Hosts	Configuration Status	Switch	VLAN	MTU	VMKNic IP Addressing	Teaming Policy	VTEP
▼ Compute Cluster A	✓ Unconfigure	vds-site-a	0	1600	IP Pool	Fail Over	1
esx-02a.corp.local	✓ Ready				✓ vmk3: 192.168.130.51		
esx-01a.corp.local	✓ Ready				✓ vmk3: 192.168.130.52		
▼ Management & Edge	✓ Unconfigure	vds-mgt-edge	0	1600	IP Pool	Fail Over	1
esxmgt-02a.corp.l	✓ Ready				✓ vmk3: 192.168.120.52		
esxmgt-01a.corp.l	✓ Ready				✓ vmk3: 192.168.120.51		

Things to note:

- The VLAN ID must be correct for the VTEP transport VLAN. Not that in the screen shot above it is “0." In most real-world deployments this would not the case.
- MTU is configured to be 1600 or larger. Make sure that the MTU is not set to 9000 with the expectation that the MTU on VMs would be also set to 9000. The DVS maximum MTU is 9000, and if VMs are also at 9000, there is no space for VXLAN headers.
- VMKNics must have the correct addresses. Make sure that they are not set to 169.254.x.x addresses, indicating that nodes have failed to get addresses from DHCP.
- The teaming policy must be consistent for all cluster members of the same DVS.
- The number of VTEPs must be the same as the number of dvUplinks. Make sure hat valid/expected IP addresses are listed.

Transport Zones have to be correctly aligned to DVS boundaries, to avoid the situation in which the DLR is missing on some clusters.

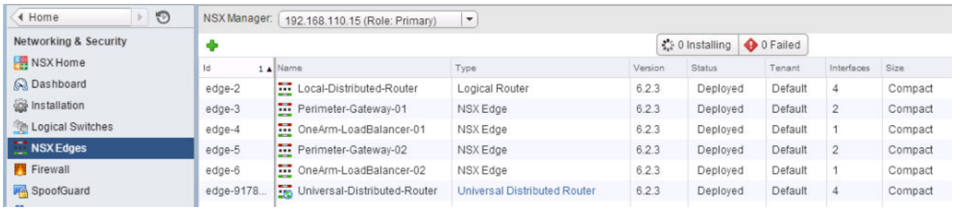


Name	NSX vSwitch	Status
Compute Cluster A	vds-site-a	✓ Normal
Management & Edge ...	vds-mgt-edge	✓ Normal

NSX Edges UI

The NSX routing subsystem is configured and managed in the “NSX Edges” section of the UI.

When this part of the UI is selected, the following view appears.



NSX Manager: 192.168.110.15 (Role: Primary)

0 Installing | 0 Failed

Id	Name	Type	Version	Status	Tenant	Interfaces	Size
edge-2	Local-Distributed-Router	Logical Router	6.2.3	Deployed	Default	4	Compact
edge-3	Perimeter-Gateway-01	NSX Edge	6.2.3	Deployed	Default	2	Compact
edge-4	OneArm-LoadBalancer-01	NSX Edge	6.2.3	Deployed	Default	1	Compact
edge-5	Perimeter-Gateway-02	NSX Edge	6.2.3	Deployed	Default	2	Compact
edge-6	OneArm-LoadBalancer-02	NSX Edge	6.2.3	Deployed	Default	1	Compact
edge-9178...	Universal-Distributed-Router	Universal Distributed Router	6.2.3	Deployed	Default	4	Compact

All currently deployed DLRs and ESGs are shown, with the following information displayed for each:

- “Id” shows the ESG or DLR Edge appliance ID, which can be used for any API calls referring to that ESG or DLR
- “Tenant” + “Id” forms the DLR instance name. This name is visible and used in the NSX CLI.

- "Size" is always "Compact" for DLR, and the size that was selected by the operator for ESG.

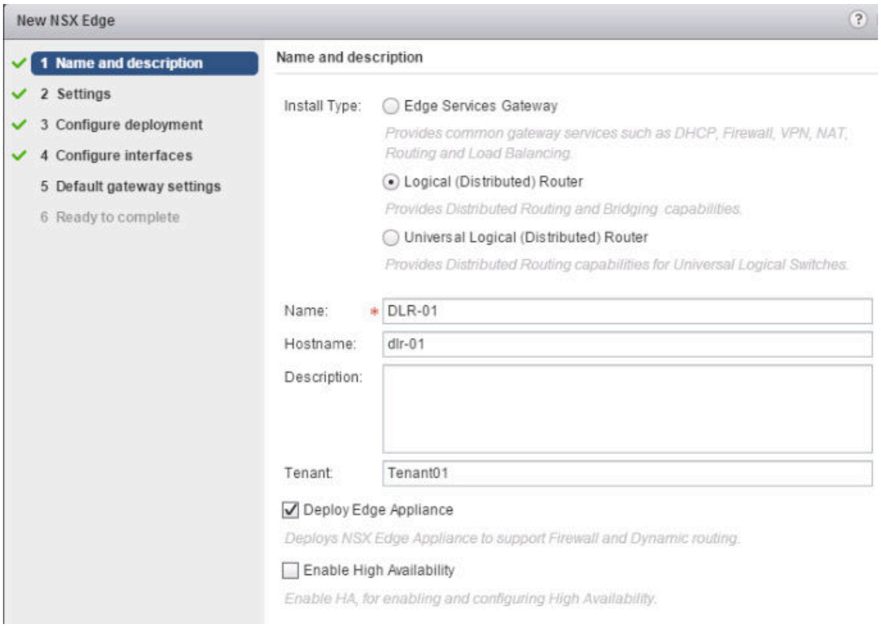
In addition to the information in the table, there is a context menu, accessible either via buttons or via "Actions."

Table 4-1. NSX Edge Context Menu

Icon	Action
🔄	"Force Sync" operation clears the ESG's or the DLR's Control VM's configuration, reboots it, and re-pushes the configuration.
⇄	"Redeploy" tears down the ESG or DLR, and creates is a new ESG or DLR with the same configuration. The existing ID is preserved.
⇆	"Change Auto Rule Configuration" applies to the ESG's built-in firewall rules, created when services are enabled on the ESG (for example, BGP which needs TCP/179).
📋	"Download tech support logs" creates a log bundle from the ESG or DLR Control VM For the DLR, host logs are not included in the tech support bundle and need to be collected separately.
⚙	"Change appliance size" is only applicable to ESGs. This will perform a "redeploy" with a new appliance (vNIC MAC addresses will change).
⚙	"Change CLI credentials" allows the operator to force-update the CLI credentials. If the CLI is locked-out on an ESG or DLR Control VM after 5 failed logins, this will not lift the lock-out. You will need to wait 5 minutes, or "Redeploy" your ESG/DLR to get back in with the correct credentials.
⚙	"Change Log Level" changes the level of detail to be sent to ESG/DLR syslog.
⚙	"Configure Advanced Debugging" re-deploys the ESG or DLR with core-dump enabled and additional virtual disk attached for storing core dump files.
⚙	"Deploy" becomes available when an ESG has been created without deploying it. This option simply executes the deployment steps (deploys OVF, configures Interfaces, pushes configuration to the created appliance.
⚙	If the version of DLR/ESG is older than NSX Manager, the "Upgrade Version" option becomes available.
🔍	"Filter" can search for ESGs/DLRs by "Name."

New NSX Edge (DLR)

When an operator creates a new DLR, the following wizard is used to collect the necessary information.



New NSX Edge

✓ 1 Name and description
✓ 2 Settings
✓ 3 Configure deployment
✓ 4 Configure interfaces
5 Default gateway settings
6 Ready to complete

Name and description

Install Type: ○ Edge Services Gateway
Provides common gateway services such as DHCP, Firewall, VPN, NAT, Routing and Load Balancing.
● Logical (Distributed) Router
Provides Distributed Routing and Bridging capabilities.
○ Universal Logical (Distributed) Router
Provides Distributed Routing capabilities for Universal Logical Switches.

Name: * DLR-01
Hostname: dlr-01
Description:
Tenant: Tenant01

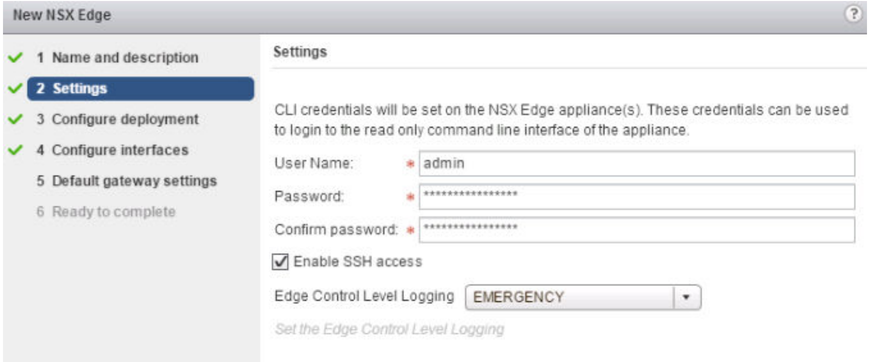
☑ Deploy Edge Appliance
Deploys NSX Edge Appliance to support Firewall and Dynamic routing.

☐ Enable High Availability
Enable HA, for enabling and configuring High Availability.

On the “Name and Description” screen, the following information is collected:

- “Name” will appear in the “NSX Edges” UI.
- “Hostname” will be used to set the DNS name of the ESG or DLR Control VM, visible on SSH/Console session, in syslog messages, and in the vCenter “Summary” page for the ESG/DLR VM under “DNS Name.”
- “Description” is in the UI showing the list of NSX Edges.
- “Tenant” will be used to form the DLR Instance Name, used by the NSX CLI. It can be also be used by external cloud management platform.

On the “Settings” screen:



New NSX Edge

✓ 1 Name and description
✓ 2 Settings
✓ 3 Configure deployment
✓ 4 Configure interfaces
5 Default gateway settings
6 Ready to complete

Settings

CLI credentials will be set on the NSX Edge appliance(s). These credentials can be used to login to the read only command line interface of the appliance.

User Name: * admin
Password: * ****************
Confirm password: * ****************

☑ Enable SSH access

Edge Control Level Logging: EMERGENCY
Set the Edge Control Level Logging

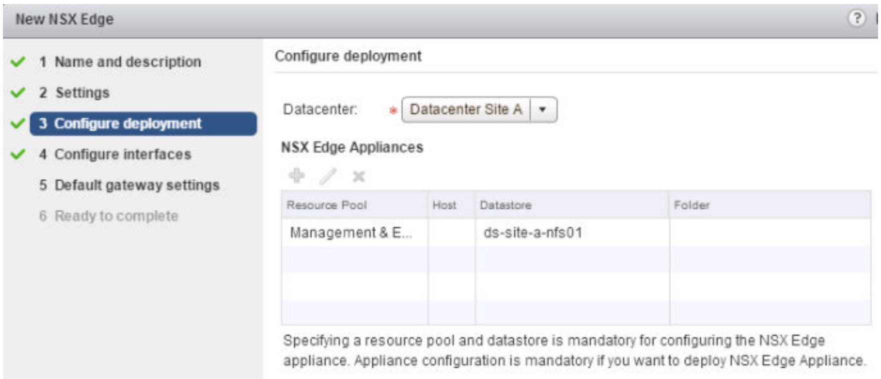
- “User Name” and “Password” set the CLI/VM console credentials to access the DLR Control VM. NSX does not support AAA on ESG or DLR Control VMs. This account has full rights to ESG/DLR Control VMs; however, the ESG/DLR configuration cannot be changed via the CLI/VMconsole.
- “Enable SSH access” enables the SSH daemon on the DLR Control VM to start.
  - The control VM Firewall rules need to be adjusted to allow SSH network access.

  - The operator can connect to the DLR Control VM from either a host on the subnet of the Control VM's management Interface, or without such restriction on the OSPF/BGP "Protocol Address," if a protocol address is configured.

    Note It is not possible to have network connectivity between the DLR Control VM and any IP address that falls into any subnet configured on any of that DLR's "Internal" interfaces. This is because the egress interface for these subnets on DLR Control VM points to the pseudo-interface "VDR," which is not connected to the data plane.

- "Enable HA" deploys Control VM as an Active/Standby HA pair.
- "Edge Control Level Logging" sets the syslog level on the Edge appliance.

On the "Configure deployment" screen:



Configure deployment

Datacenter: * Datacenter Site A

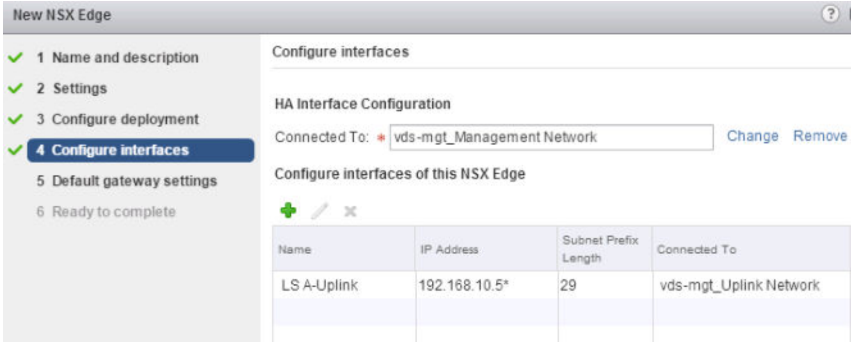
NSX Edge Appliances

Resource Pool	Host	Datastore	Folder
Management & E...		ds-site-a-nfs01	

Specifying a resource pool and datastore is mandatory for configuring the NSX Edge appliance. Appliance configuration is mandatory if you want to deploy NSX Edge Appliance.

- "Datacenter" selects the vCenter datacenter in which to deploy the Control VM.
- "NSX Edge Appliances" refers to the DLR Control VM and allows definition of exactly one (as shown).
  - If "HA" is enabled, the Standby Edge will be deployed on the same cluster, host, and datastore. A DRS "Separate Virtual Machines" rule will be created for the Active and Standby DLR Control VMs.

On the "Configure Interfaces" screen:



Configure interfaces

HA Interface Configuration

Connected To: * vds-mgt_Management Network

Configure interfaces of this NSX Edge

Name	IP Address	Subnet Prefix Length	Connected To
LS A-Uplink	192.168.10.5*	29	vds-mgt_Uplink Network

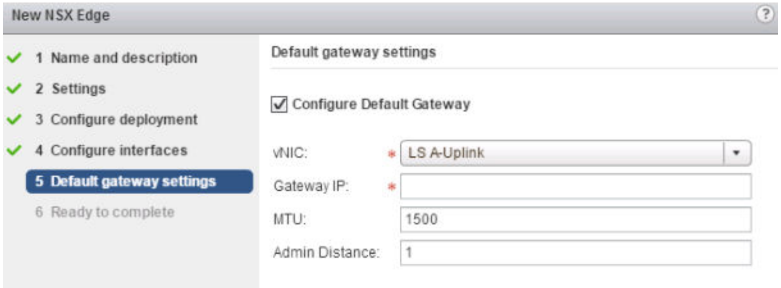
- "HA Interface"
  - Is not created as a DLR logical interface capable of routing. It is only a vNIC on the Control VM.
  - This interface does not require an IP address, because NSX manages the DLR configuration via VMCI.

  - This interface is used for HA heartbeat if the DLR "Enable High Availability" is checked on the "Name and description" screen.

- "Interfaces of this NSX Edge" refer to DLR Logical Interfaces (LIFs)

  - The DLR provides L3 gateway services to VMs on the "Connected To" dvPortgroup or logical switch with IP addresses from corresponding subnets.
  - "Uplink" type LIFs are created as vNICs on the Control VM, so, up to eight are supported; the last two available vNICs are allocated to the HA interface and one reserved vNIC.
  - An "Uplink" type LIF is required for dynamic routing to work on the DLR.
  - And "Internal" type LIFs are created as pseudo-vNICs on the Control VM, and it is possible to have up to 991 of them.

On the "Default gateway settings" screen:



New NSX Edge

✓ 1 Name and description
✓ 2 Settings
✓ 3 Configure deployment
✓ 4 Configure interfaces
5 Default gateway settings
6 Ready to complete

Default gateway settings

☑ Configure Default Gateway

vNIC: * LS A-Uplink
Gateway IP: *
MTU: 1500
Admin Distance: 1

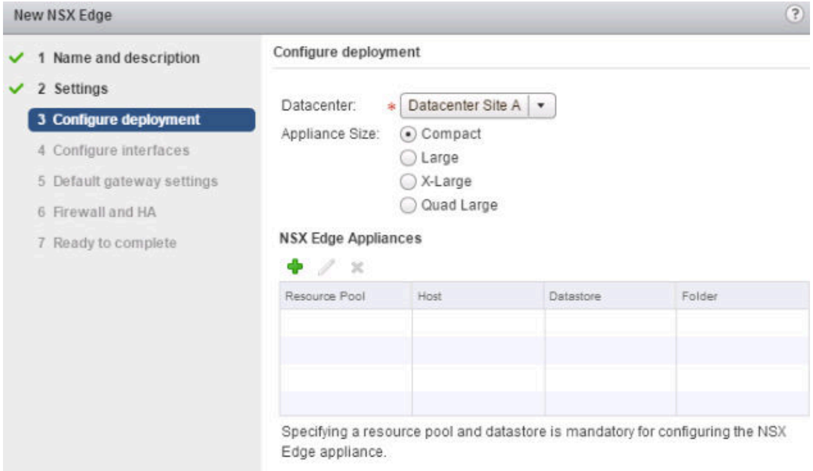
- Configure Default Gateway, if selected, will create a static default route on the DLR. This option is available if an "Uplink" type LIF is created in the previous screen.
- If ECMP is used on the uplink, the recommendation is to leave this option disabled, to prevent dataplane outage in case of next-hop failure.

Note The double right-arrow in the top right corner allows for "suspending" the wizard in progress so that it can be resumed at a later time.

ESG and DLR Differences

There are some differences between the wizard screens when an ESG is deployed, compared to a DLR.

The first one is on the "Configure deployment" screen:



New NSX Edge

- ✓ 1 Name and description
- ✓ 2 Settings
- 3 Configure deployment
- 4 Configure interfaces
- 5 Default gateway settings
- 6 Firewall and HA
- 7 Ready to complete

Configure deployment

Datacenter: * Datacenter Site A ▾

Appliance Size: ● Compact
○ Large
○ X-Large
○ Quad Large

NSX Edge Appliances

Resource Pool	Host	Datastore	Folder

Specifying a resource pool and datastore is mandatory for configuring the NSX Edge appliance.

For an ESG, “Configure Deployment” allows selection of the Edge size. If an ESG is used only for routing, “Large” is a typical size that is suitable in most scenarios. Selecting a larger size will not provide more CPU resources to the ESG’s routing processes, and will not lead to more throughput.

It is also possible to create an ESG without deploying it, which still requires configuration of an Edge Appliance.

A “Non-deployed” Edge can be later deployed via an API call or with the “Deploy” UI action.

If Edge HA is selected, you must create at least one “Internal” interface, or HA will fail silently, leading to the “split-brain” scenario.

The NSX UI and API allow an operator to remove the last “Internal” interface, which will cause HA to silently fail.

Typical ESG and DLR UI Operations

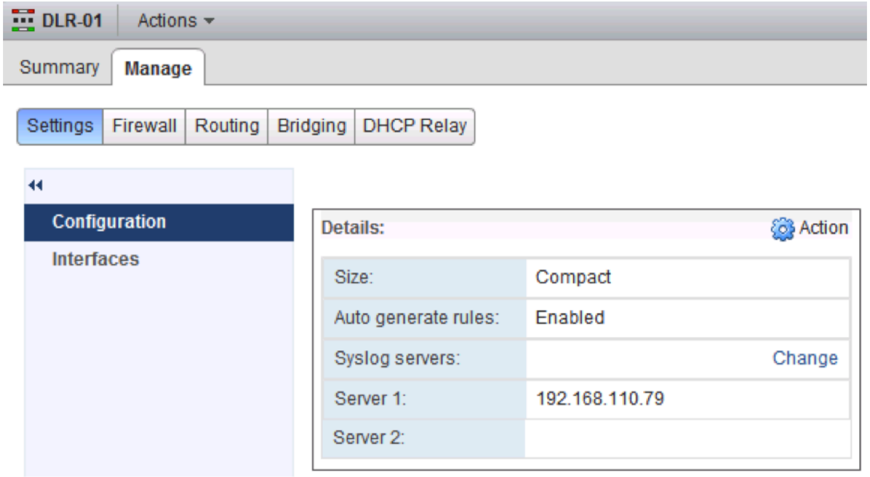
In addition to creation, there are several configuration operations that are typically executed after initial deployment.

These include:

- Syslog configuration
- Management of static routes
- Configuration of routing protocols and route redistribution

Syslog Configuration

Configure the ESG or DLR Control VM to send log entries to a remote syslog server.

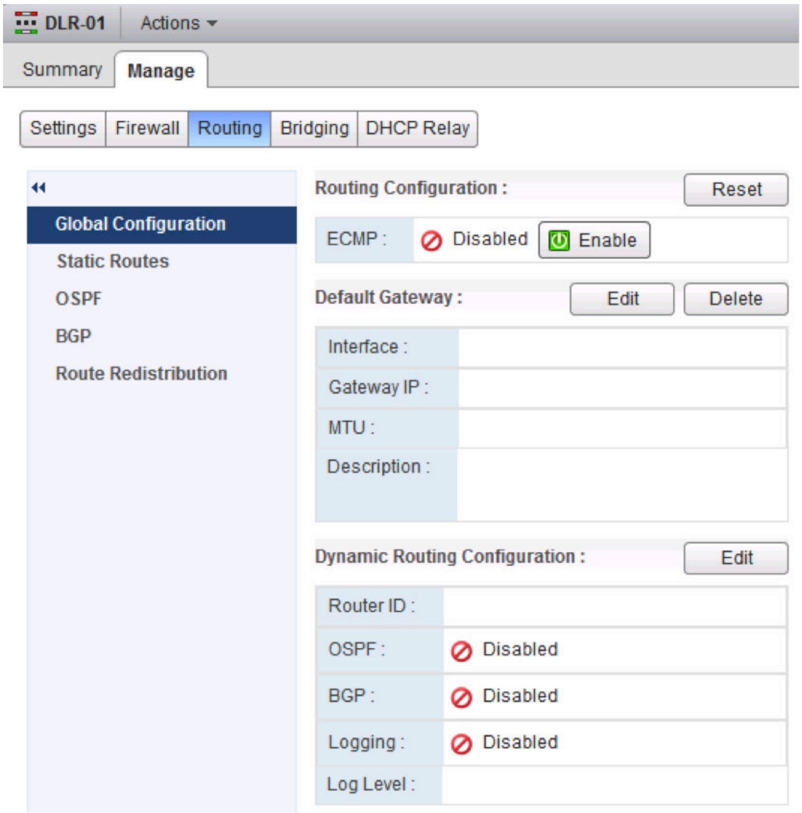


Size: Compact
Auto generate rules: Enabled
Syslog servers:
Server 1: 192.168.110.79
Server 2:

Notes:

- The syslog server must be configured as an IP address, because the ESG/DLR Control VM does not get configured with a DNS resolver.
  - In the ESG's case, it is possible to "Enable DNS Service" (DNS proxy) that ESG itself will be able to use to resolve DNS names, but generally specifying syslog server as an IP address in a more reliable method with fewer dependencies.
- There is no way to specify a syslog port in the UI (it is always 514), but protocol (UDP/TCP) can be specified.
- Syslog messages originate from the IP address of the Edge's interface that is selected as egress for the syslog server's IP by the Edge's forwarding table.
  - For the DLR, the syslog server's IP address cannot be on any subnets configured on any of the DLR's "Internal" interfaces. This is because the egress interface for these subnets on the DLR Control VM points to the pseudo-interface "VDR," which is not connected to the data plane.

By default, logging for the ESG/DLR routing engine is disabled. If required, enable it via UI by clicking "Edit" for the "Dynamic Routing Configuration."



Routing Configuration :

ECMP : ⊘ Disabled

Default Gateway :

Interface :

Gateway IP :

MTU :

Description :

Dynamic Routing Configuration :

Router ID :

OSPF : ⊘ Disabled

BGP : ⊘ Disabled

Logging : ⊘ Disabled

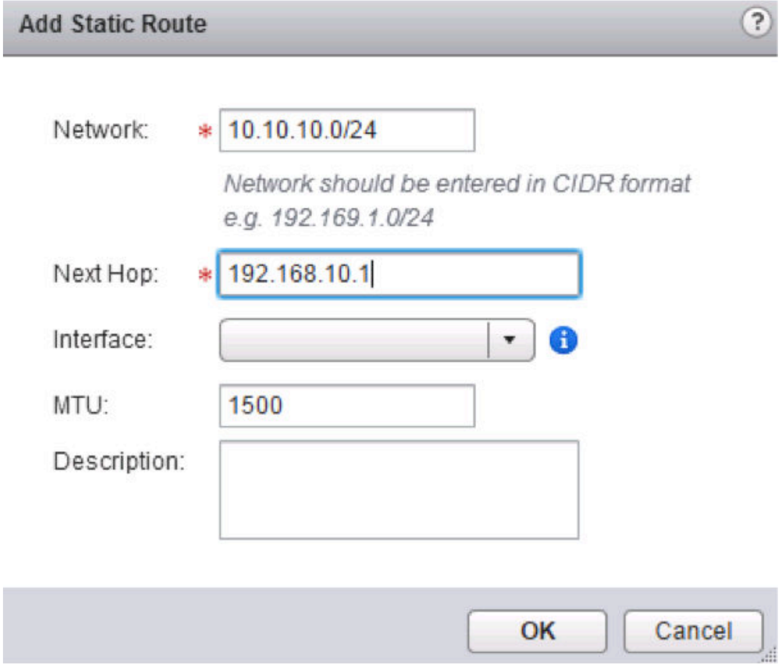
Log Level :

You must also configure the Router ID, which will typically be the IP address of the Uplink interface.

Static Routes

Static routes must have the next hop set to an IPaddress on a subnet associated with one of DLR’s LIFs or ESG’s Interfaces. Otherwise, configuration fails.

“Interface,” if not selected, is set automatically by matching the next hop to one of directly connected subnets.



Add Static Route

Network: * 10.10.10.0/24

Network should be entered in CIDR format e.g. 192.169.1.0/24

Next Hop: * 192.168.10.1

Interface:

MTU: 1500

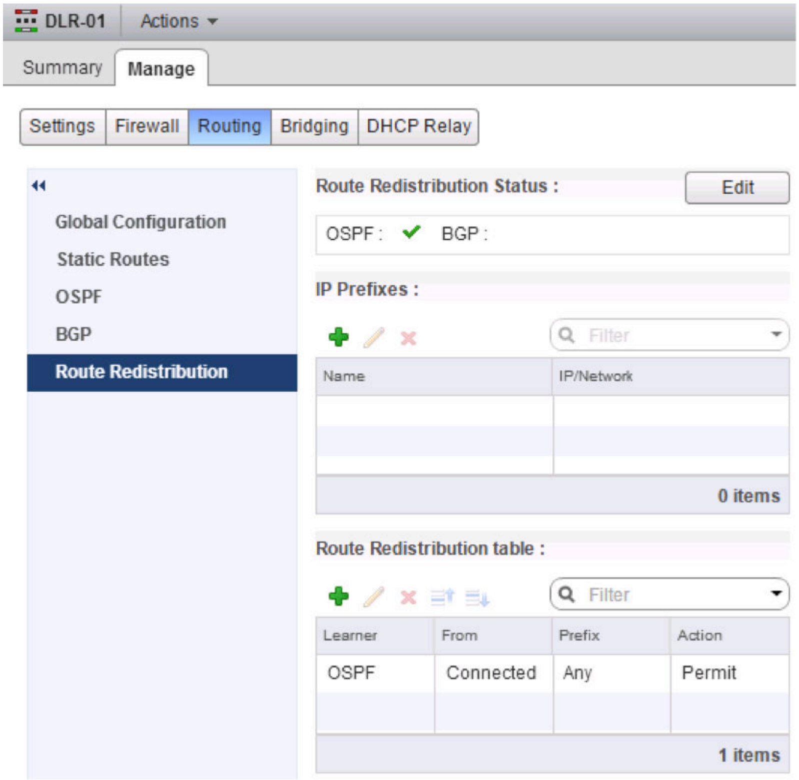
Description:

Route Redistribution

Adding an entry into the “Route Redistribution table” does not automatically enable redistribution for the selected “Learner Protocol." This must be done explicitly via “Edit” for “Route Redistribution Status."

The DLR is configured with redistribution of connected routes into OSPF by default, while ESG is not.

The “Route Redistribution table” is processed in top-to-bottom order, and processing is stopped after the first match. To exclude some prefixes from redistribution, include more specific entries at the top.



Route Redistribution Status :

OSPF : ✓ BGP :

IP Prefixes :

Name	IP/Network

0 items

Route Redistribution table :

Learner	From	Prefix	Action
OSPF	Connected	Any	Permit

1 items

Troubleshooting NSX Routing

NSX provides multiple tools for making sure that routing is working.

NSX Routing CLI

There is a collection of CLI commands that allow an operator to examine the running state of various parts of the NSX routing subsystem.

Due to the distributed nature of the the NSX routing subsystem, there are a number of CLIs available, accessible on various components of NSX. Starting in NSX version 6.2, NSX also has a centralized CLI that helps reduce the “travel time” required to access and log in to various distributed components. It provides access to most of the information from a single location: the NSX Manager shell.

Checking the Prerequisites

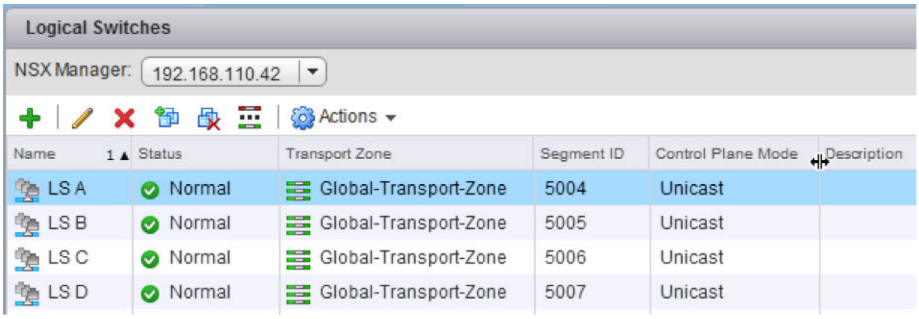
There are two major prerequisites that must be satisfied for each ESXi host:

- Any logical switches connected to the DLR are healthy.
- The ESXi host has been successfully prepared for VXLAN.

Logical Switch Health Check

NSX Routing works in conjunction with NSX logical switching. To verify that the logical switches connected to a DLR are healthy:

- Find the segment ID (VXLAN VNI) for each logical switch connected to the DLR in question (for example, 5004..5007).



Logical Switches

NSX Manager: 192.168.110.42

Name	Status	Transport Zone	Segment ID	Control Plane Mode	Description
LS A	Normal	Global-Transport-Zone	5004	Unicast	
LS B	Normal	Global-Transport-Zone	5005	Unicast	
LS C	Normal	Global-Transport-Zone	5006	Unicast	
LS D	Normal	Global-Transport-Zone	5007	Unicast	

- On the ESXi hosts where VMs served by this DLR are running, check the state of the VXLAN control plane for the logical switches connected to this DLR.

```
# esxcli network vswitch dvs vmware vxlan network list --vds-name=Compute_VDS
VXLAN ID  Multicast IP               Control Plane                        Controller Connection  Port
Count  MAC Entry Count  ARP Entry Count
--------  -------------------------  -----------------------------------  ---------------------
----------  ---------------  ---------------
    5004  N/A (headend replication)  Enabled (multicast proxy,ARP proxy)  192.168.110.201
(up)            2                2                0
    5005  N/A (headend replication)  Enabled (multicast proxy,ARP proxy)  192.168.110.202
(up)            1                0                0
    5006  N/A (headend replication)  Enabled (multicast proxy,ARP proxy)  192.168.110.203
(up)            1                1                0
    5007  N/A (headend replication)  Enabled (multicast proxy,ARP proxy)  192.168.110.202
(up)            1                0                0
```

Check the following for each relevant VXLAN:

- For logical switches in hybrid or unicast mode:
  - Control Plane is “Enabled.”
  - “multicast proxy” and “ARP proxy” are listed; “ARP proxy” will be listed even if you disabled IP Discovery.
  - A valid Controller IP address is listed under “Controller,” and “Connection” is “up.”
- “Port Count” looks right – there will be at least 1, even if there are no VMs on that host connected to the logical switch in question. This one port is the vdrPort, a special dvPort connected to the DLR kernel module on the ESXi host.

- Run the following command to make sure that the vdrPort is connected to each of the relevant VXLANs.

```
~ # esxcli network vswitch dvs vmware vxlan network port list --vds-name=Compute_VDS --vxlan-id=5004
Switch Port ID  VDS Port ID  VMKNIC ID
--------------  -----------  ---------
      50331656  53                   0
      50331650  vdrPort              0


~ # esxcli network vswitch dvs vmware vxlan network port list --vds-name=Compute_VDS --vxlan-id=5005
Switch Port ID  VDS Port ID  VMKNIC ID
--------------  -----------  ---------
      50331650  vdrPort              0
```

- In the example above, VXLAN 5004 has one VM and one DLR connection, while VXLAN 5005 only has a DLR connection.
- Check whether the appropriate VMs have been properly wired to their corresponding VXLANs, for example web-sv-01a on VXLAN 5004.

```
~ # esxcfg-vswitch -l
DVS Name         Num Ports   Used Ports  Configured Ports  MTU     Uplinks
Compute_VDS      1536        10          512               1600    vmnic0

  DVPort ID           In Use      Client
[..skipped..]
  53                  1           web-sv-01a.eth0
```

VXLAN Preparation Check

As part of VXLAN configuration of an ESXi host, the DLR kernel module is also installed, configured, and connected to a dvPort on a DVS prepared for VXLAN.

1 Run show `cluster all` to get the cluster ID.

2 Run show `cluster cluster-id` to get the host ID.

3 Run show `logical-router host hostID connection` to get the status information.

```
nsxmgr-01a# show logical-router host <hostID> connection

Connection Information:
-----------------------

DvsName           VdrPort           NumLifs  VdrVmac
-------           -------           -------  -------
Compute_VDS       vdrPort           4        02:50:56:56:44:52
    Teaming Policy: Default Teaming
    Uplink   : dvUplink1(50331650): 00:50:56:eb:41:d7(Team member)
```

```
 Stats : Pkt Dropped      Pkt Replaced     Pkt Skipped
 Input : 0                0                1968734458
Output : 303              7799             31891126
```

- A DVS enabled with VXLAN will have one vdrPort created, shared by all DLR instances on that ESXi host.
- “NumLifs” refers to the number that is the sum of LIFs from all DLR instances that exist on this host.
- “VdrVmac” is the vMAC that the DLR uses on all LIFs across all instances. This MAC is the same on all hosts. It is never seen in any frames that travel the physical network outside of ESXi hosts.
- For each dvUplink of DVS enabled with VXLAN, there is a matching VTEP; except in cases where LACP / Etherchannel teaming mode is used, when only one VTEP is created irrespective of the number of dvUplinks.
  - Traffic routed by the DLR (SRC MAC = vMAC) when leaving the host will get the SRC MAC changed to pMAC of a corresponding dvUplink.
  - Note that the original VM’s source port or source MAC is used to determine the dvUplink (it is preserved for each packet in its DVS's metadata).
  - When there are multiple VTEPs on the host and one of dvUplinks fails, the VTEP associated with the failed dvUplink will be moved to one of the remaining dvUplinks, along with all VMs that were pinned to that VTEP. This is done to avoid flooding control plane changes that would be associated with moving VMs to a different VTEP.
- The number in “()” next to each “dvUplinkX” is the dvPort number. It is useful for packet capture on the individual uplink.
- The MAC address shown for each “dvUplinkX” is a “pMAC” associated with that dvUplink. This MAC address is used for traffic sourced from the DLR, such as ARP queries generated by the DLR and any packets that have been routed by the DLR when these packets leave the ESXi host. This MAC address can be seen on the physical network (directly, if DLR LIF is VLAN type, or inside VXLAN packets for VXLAN LIFs).
- Pkt Dropped / Replaced / Skipped refer to counters related to internal implementation details of the DLR, and are not typically used for troubleshooting or monitoring.

Brief Recap of Routing

To effectively troubleshoot routing issues, it is helpful to review how routing works and the related information tables.

1 Receive a packet to send to a destination IP address.

2 Check the routing table and determine the IP address of the next hop.

3 Determine which of your network interfaces can reach it.

4 Get a MAC address of that next hop (via ARP).

5 Build an L2 frame.

6 Send the frame out the interface.

So to do routing, you need:

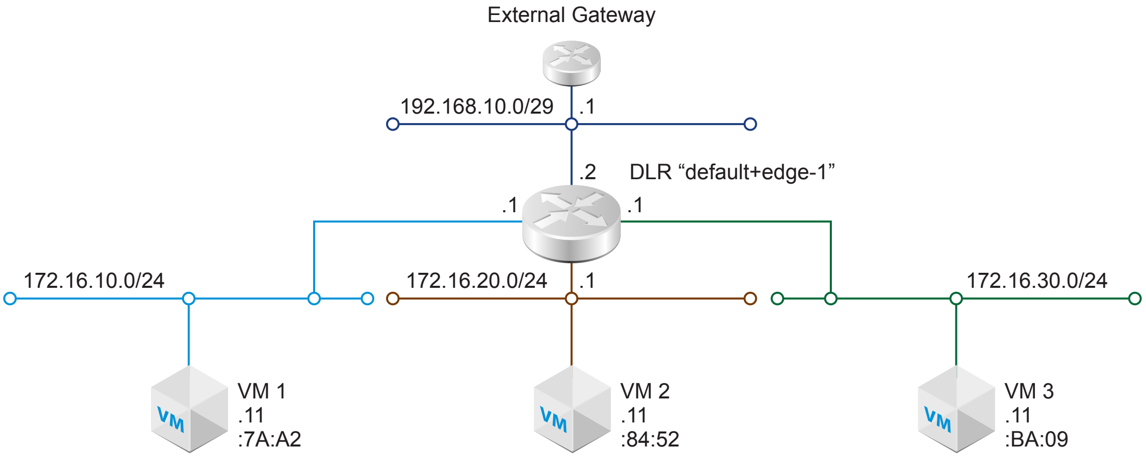
- An interface table (with interface IP addresses and netmasks)
- A routing table
- An ARP table

Verifying the DLR State Using a Sample Routed Topology

This section discusses how to the information that the DLR requires to route packets.

Let's take a sample routed topology and create a set of logical switches and a DLR to create it in NSX.

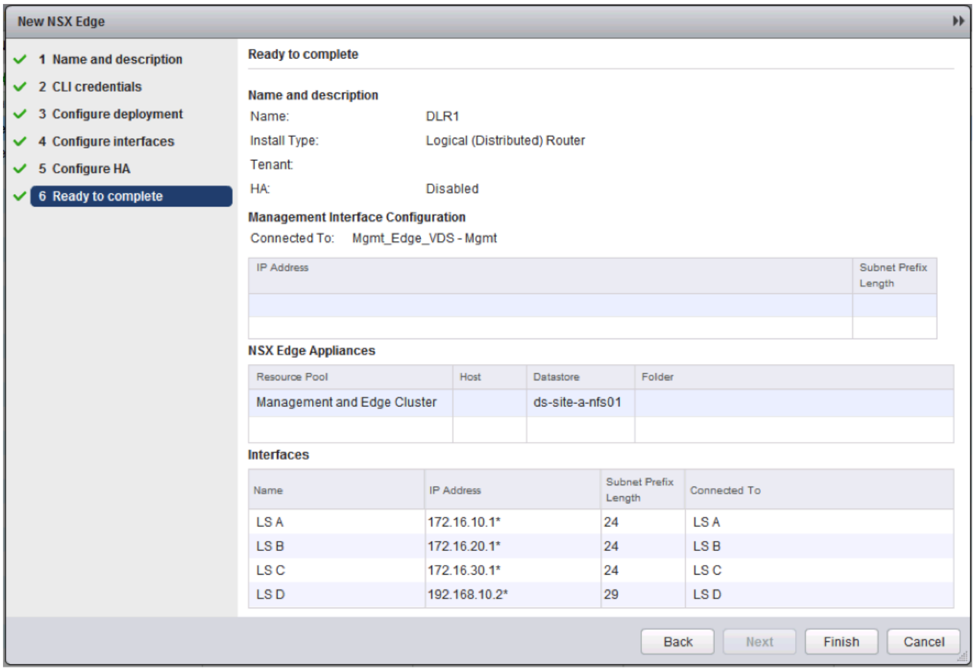
Figure 4-7. Sample Routed Topology



The diagram shows:

- 4 x Logical Switches, each with its own subnet
- 3 x VMs, connected one per logical switch
  - Each with its own IP address and IP gateway
  - Each with a MAC address (last two octets are shown)
- One DLR connected to the 4 logical switches; one logical switch is for the "Uplink," while the rest are Internal
- An external gateway, which could be an ESG, serving as an upstream gateway for the DLR.

The "Ready to complete" wizard screen shows for the DLR above.



New NSX Edge

- ✓ 1 Name and description
- ✓ 2 CLI credentials
- ✓ 3 Configure deployment
- ✓ 4 Configure interfaces
- ✓ 5 Configure HA
- ✓ 6 Ready to complete

Ready to complete

Name and description

Name:	DLR1
Install Type:	Logical (Distributed) Router
Tenant:	
HA:	Disabled

Management Interface Configuration

Connected To: Mgmt_Edge_VDS - Mgmt

IP Address	Subnet Prefix Length

NSX Edge Appliances

Resource Pool	Host	Datastore	Folder
Management and Edge Cluster		ds-site-a-nfs01	

Interfaces

Name	IP Address	Subnet Prefix Length	Connected To
LS A	172.16.10.1*	24	LS A
LS B	172.16.20.1*	24	LS B
LS C	172.16.30.1*	24	LS C
LS D	192.168.10.2*	29	LS D

After the deployment of the DLR finishes, ESXi CLI commands can be used to view and validate the distributed state of the DLR in question on the participating hosts.

Confirming DLR Instances

The first thing to confirm is whether the DLR instance has been created and whether its control plane is active.

1. From the NSX Manager shell, run `show cluster all` to get the cluster ID.
2. Run `show cluster cluster-id` to get the host ID.
3. Run `show logical-router host hostID dlr all verbose` to get the status information.

```
nsxmgr# show logical-router host host-id dlr all verbose

VDR Instance Information :
---------------------------

Vdr Name:                   default+edge-1
Vdr Id:                     1460487509
Number of Lifs:             4
Number of Routes:           5
State:                      Enabled
Controller IP:              192.168.110.201
Control Plane Active:       Yes
Control Plane IP:           192.168.210.51
Edge Active:                No
```

The points to note:

- This command displays all DLR instances that exist on the given ESXi host.

- “Vdr Name” consists of “Tenant” “+ “Edge Id." In the example, “Tenant” was not specified, so the word “default” is used. The “Edge Id” is “edge-1," which can be seen in the NSX UI.
  - In cases where there are many DLR instances on a host, a method for finding the right instance is to look for the “Edge ID” displayed in the UI “NSX Edges."
- “Vdr Id” is useful for further lookups, including logs.
- “Number of Lifs” refers to the LIFs that exist on this individual DLR instance.
- “Number of Routes” is in this case 5, which consists of 4 x directly connected routes (one for each LIF), and a default route.
- “State,” “Controller IP,” and “Control Plane Active” refer to the state of the DLR’s control plane and must list the correct Controller IP, with Control Plane Active: Yes. Remember, the DLR function requires working Controllers; the output above shows what is expected for a healthy DLR instance.
- “Control Plane IP” refers to the IP address that the ESXi host uses to talk to the Controller. This IP is always the one associated with the ESXi host’s Management vmknic, which in most cases is vmk0.
- “Edge Active” shows whether or not this host is the one where the Control VM for this DLR instance is running and in Active state.
  - The placement of the Active DLR Control VM determines which ESXi host is used to perform NSX L2 bridging, if it is enabled.
- There is also a “brief” version of the above command that produces a compressed output useful for a quick overview. Note that “Vdr Id” is displayed in hexadecimal format here:

```
nsxmgr# show logical-router host host-id dlr all brief

VDR Instance Information :
---------------------------

State Legend: [A: Active], [D: Deleting], [X: Deleted], [I: Init]
State Legend: [SF-R: Soft Flush Route], [SF-L: Soft Flush LIF]

Vdr Name             Vdr Id     #Lifs   #Routes State      Controller Ip    CP Ip
--------             -------    -----   ------- -----      -------------    ------
default+edge-1       0x570d4555 4       5       A          192.168.110.201  192.168.210.51
```

The “Soft Flush” states refer to short-lived transient states of the LIF lifecycle and is not normally seen in a healthy DLR.

DLR’s Logical Interfaces

After establishing the that the DLR has been created, make sure that all of the DLR's logical interfaces are present and have the correct configuration.

1. From the NSX Manager shell, run `show cluster all` to get the cluster ID.
2. Run `show cluster cluster-id` to get the host ID.
3. Run `show logical-router host hostID dlr all brief` to get the dlrID (Vdr Name).

4 Run show `logical-router host hostID dlr dlrID interface all brief` to get summarized status information for all interfaces.

5 Run show `logical-router host hostID dlr dlrID interface (all | intName) verbose` to get the status information for all interfaces or for a specific interface.

```
nsxmgr# show logical-router host hostID dlr dlrID interface all verbose

VDR default+edge-1:1460487509 LIF Information :

Name:                570d45550000000a
Mode:                Routing, Distributed, Internal
Id:                  Vxlan:5000
Ip(Mask):            172.16.10.1(255.255.255.0)
Connected Dvs:       Compute_VDS
VXLAN Control Plane: Enabled
VXLAN Multicast IP:  0.0.0.1
State:               Enabled
Flags:               0x2388
DHCP Relay:          Not enabled

Name:                570d45550000000c
Mode:                Routing, Distributed, Internal
Id:                  Vxlan:5002
Ip(Mask):            172.16.30.1(255.255.255.0)
Connected Dvs:       Compute_VDS
VXLAN Control Plane: Enabled
VXLAN Multicast IP:  0.0.0.1
State:               Enabled
Flags:               0x2288
DHCP Relay:          Not enabled

Name:                570d45550000000b
Mode:                Routing, Distributed, Internal
Id:                  Vxlan:5001
Ip(Mask):            172.16.20.1(255.255.255.0)
Connected Dvs:       Compute_VDS
VXLAN Control Plane: Enabled
VXLAN Multicast IP:  0.0.0.1
State:               Enabled
Flags:               0x2388
DHCP Relay:          Not enabled

Name:                570d455500000002
Mode:                Routing, Distributed, Uplink
Id:                  Vxlan:5003
Ip(Mask):            192.168.10.2(255.255.255.248)
Connected Dvs:       Compute_VDS
VXLAN Control Plane: Enabled
VXLAN Multicast IP:  0.0.0.1
State:               Enabled
Flags:               0x2208
DHCP Relay:          Not enabled
```

The points to note:

- LIF "Name" is unique across all DLR instances on the host. It is the same on hosts and on the DLR's master Controller node.
- LIF's "Mode" shows whether the LIF is routing or bridging, and whether it is internal or uplink.
- "Id" shows the LIF type and the corresponding service ID (VXLAN and VNI, or VLAN and VID).
- "Ip(Mask)" is shown for "Routing" LIFs.
- If a LIF is connected to a VXLAN in hybrid or unicast mode, "VXLAN Control Plane" is "Enabled."
- For VXLAN LIFs where VXLAN is in unicast mode, "VXLAN Multicast IP" is shown as "0.0.0.1"; otherwise the actual multicast IP address is displayed.
- "State" should be "Enabled" for routed LIFs. For bridging LIFs, it is "Enabled" on the host that is performing bridging and "Init" on all other hosts.
- "Flags" is a summary representation of the LIF's state and shows whether the LIF is:
  - Routed or Bridged
  - Whether the VLAN LIF is a DI
  - Whether it has DHCP relay enabled
  - Of note is the flag 0x0100, which is set when a VXLAN VNI join was caused by the DLR (as opposed to a host having a VM on that VXLAN)
  - Flags are displayed in a more readable format in "brief" mode.

```
nsxmgr# show logical-router host hostID dlr dlrID interface all brief

VDR default+edge-1 LIF Information :

State Legend: [A:Active], [d:Deleting], [X:Deleted], [I:Init],[SF-L:Soft Flush LIF]
Modes Legend: [B:Bridging],[E: Empty], [R:Routing],[S:Sedimented],[D:Distributed]
Modes Legend: [In:Internal],[Up:Uplink]

Lif Name            Id              Mode     State    Ip(Mask)
--------            --              -----    -----    --------
570d45550000000a    Vxlan:5001      R,D,In   A        172.16.10.1(255.255.255.0)
570d45550000000c    Vxlan:5003      R,D,In   A        172.16.30.1(255.255.255.0)
570d45550000000b    Vxlan:5002      R,D,In   A        172.16.20.1(255.255.255.0)
570d455500000002    Vxlan:5000      R,D,Up   A        192.168.10.5(255.255.255.248)
```

DLR's Routes

After you have established that a DLR is present and healthy and it has all the LIFs, the next thing to check is the routing table.

1. From the NSX Manager shell, run `show cluster all` to get the cluster ID.
2. Run `show cluster cluster-id` to get the host ID.
3. Run `show logical-router host hostID dlr all brief` to get the dlrID (Vdr Name).

4 Run `show logical-router host hostID dlr dlrID route` to get the status information for all interfaces.

```
nsxmgr# show logical-router host hostID dlr dlrID route

VDR default+edge-1:1460487509 Route Table
Legend: [U: Up], [G: Gateway], [C: Connected], [I: Interface]
Legend: [H: Host], [F: Soft Flush] [!: Reject] [E: ECMP]

Destination      GenMask          Gateway          Flags    Ref Origin   UpTime     Interface
-----------      -------          -------          -----    --- ------   ------     ---------
0.0.0.0          0.0.0.0          192.168.10.1     UG       1   AUTO     10068944   570d455500000002
172.16.10.0      255.255.255.0    0.0.0.0          UCI      1   MANUAL   10068944   570d45550000000a
172.16.20.0      255.255.255.0    0.0.0.0          UCI      1   MANUAL   10068944   570d45550000000b
172.16.30.0      255.255.255.0    0.0.0.0          UCI      1   MANUAL   10068944   570d45550000000c
192.168.10.0     255.255.255.248  0.0.0.0          UCI      1   MANUAL   10068944   570d455500000002
```

Points to note:

- "Interface" shows the egress LIF that will be selected for the corresponding "Destination." It is set to the "Lif Name" of one of the DLR's LIFs.
- For ECMP routes, there will be more than one route with the same Destination, GenMask, and Interface, but a different Gateway. Fags will also include "E" to reflect the ECMP nature of these routes.

DLR's ARP table

For packets it forwards, the DLR must be able to resolve ARP requests for the next hop's IP address. The results of this resolution process are stored locally on the individual hosts' DLR instances.

Controllers play no role in this process and are not used to distribute resulting the ARP entries to other hosts.

Inactive cached entries are kept for 600 seconds, then removed. For more information about the DLR ARP resolution process, see DLR ARP Resolution Process.

1 From the NSX Manager shell, run `show cluster all` to get the cluster ID.

2 Run `show cluster cluster-id` to get the host ID.

3 Run `show logical-router host hostID dlr all brief` to get the dlrID (Vdr Name).

4 Run `show logical-router host hostID dlr dlrID arp` to get the status information for all interfaces.

```
nsxmgr# show logical-router host hostID dlr dlrID arp

VDR default+edge-1:1460487509 ARP Information :
Legend: [S: Static], [V: Valid], [P: Proxy], [I: Interface]
Legend: [N: Nascent], [L: Local], [D: Deleted]

Network           Mac                  Flags      Expiry     SrcPort    Interface Refcnt
-------           ---                  -----      -------    ---------  --------- ------
```

```
172.16.10.1       02:50:56:56:44:52    VI         permanent  0          570d45550000000a 1
172.16.10.11      00:50:56:a6:7a:a2    VL         147        50331657   570d45550000000a 2
172.16.30.1       02:50:56:56:44:52    VI         permanent  0          570d45550000000c 1
172.16.30.11      00:50:56:a6:ba:09    V          583        50331650   570d45550000000c 2
172.16.20.11      00:50:56:a6:84:52    VL         568        50331658   570d45550000000b 2
172.16.20.1       02:50:56:56:44:52    VI         permanent  0          570d45550000000b 1
192.168.10.2      02:50:56:56:44:52    VI         permanent  0          570d455500000002 1
192.168.10.1      00:50:56:8e:ee:ce    V          147        50331650   570d455500000002 1
```

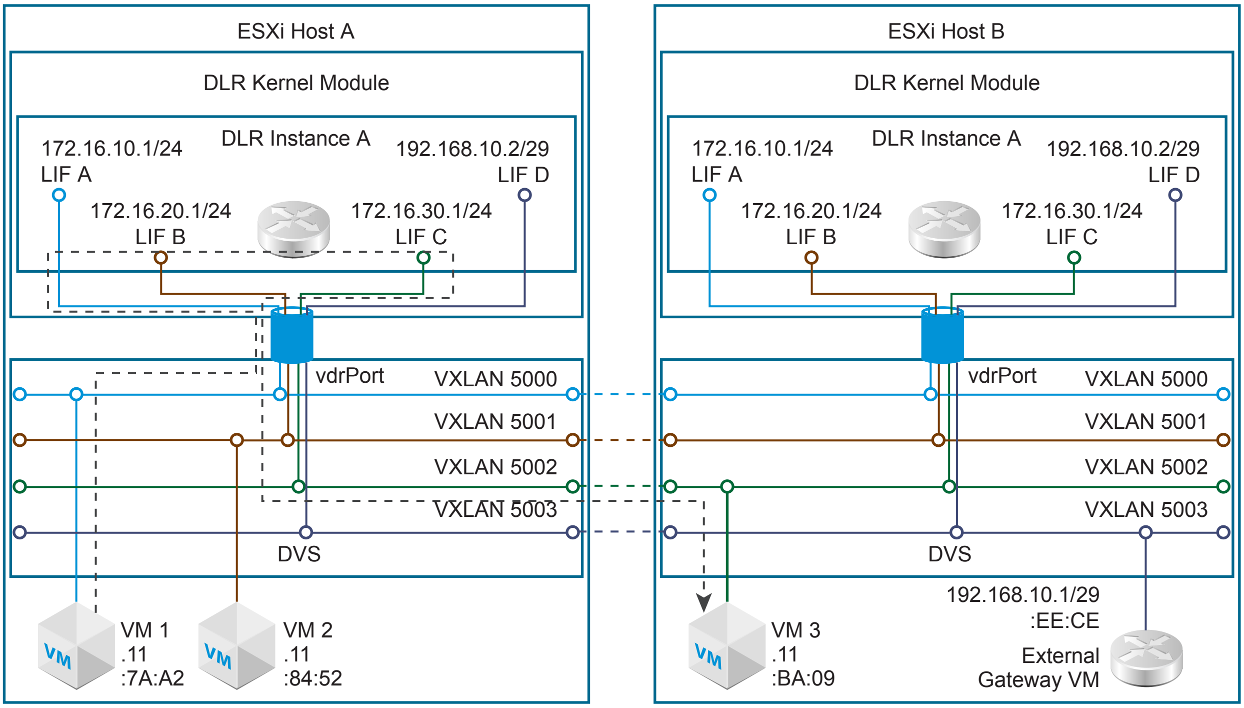
Things to note:

- All ARP entries for the DLR’s own LIFs (“I” Flag) are the same and show the same vMAC that was discussed in VXLAN Preparation Check.
- ARP entries with the “L” Flag correspond to the VMs running on the host where the CLI command is run.
- “SrcPort” shows the dvPort ID where the ARP entry was originated. In cases where an ARP entry was originated on another host, the dvUplink’s dvPort ID is shown. This dvPort ID can be cross-referenced with the dvUplink dvPort ID discussed in VXLAN Preparation Check.
- The “Nascent” flag is not normally observed. It is set while the DLR is waiting for the ARP reply to arrive. Any entries with that flag set might indicate that there is a problem with ARP resolution.

DLR and Its Related Host Components Visualized

The following diagram shows two hosts, ESXi Host A and ESXi Host B, where our example “DLR Instance A” is configured and connected to the four VXLAN LIFs.

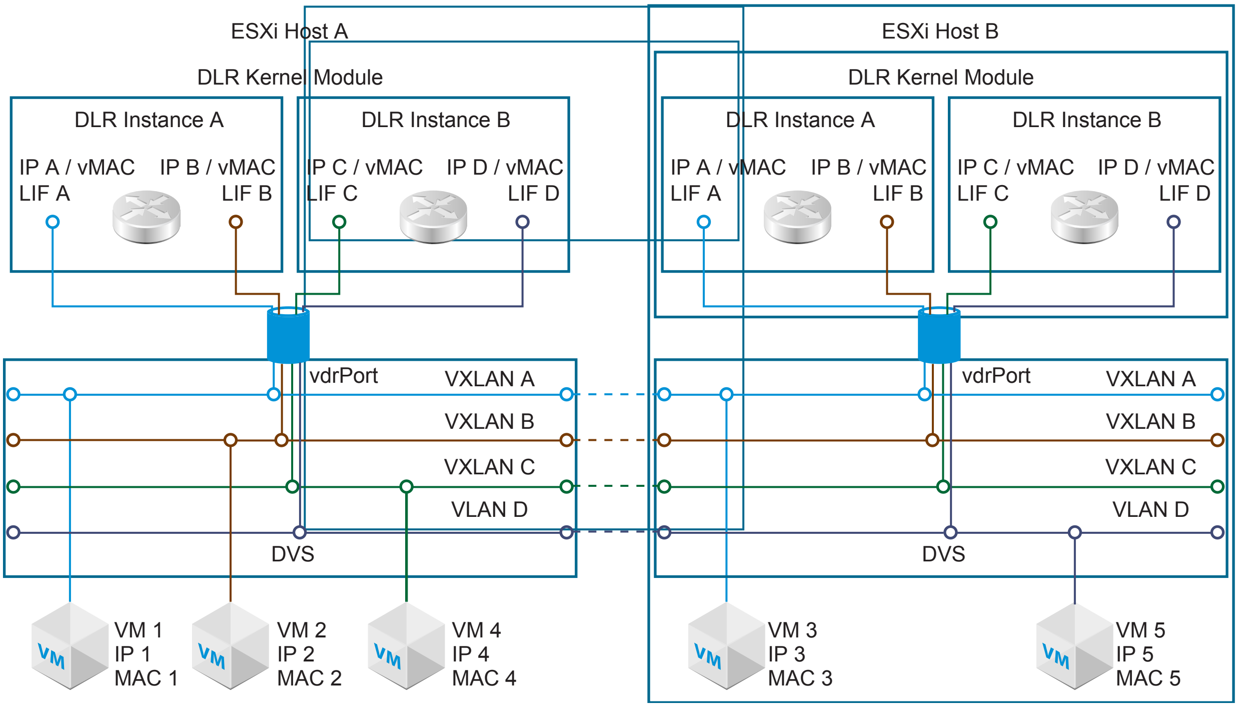
Figure 4‑8. Two Hosts with a Single DLR Instance



- Each host has an “L2 Switch” (DVS), and a “Router on a stick” (DLR kernel module), connected to that “switch” via a “trunk” interface (vdrPort).
  - Note that this “trunk” can carry both VLANs and VXLANs; however, there are no 801.Q or UDP/VXLAN headers present in the packets that traverse the vdrPort. Instead, the DVS uses an internal metadata tagging method to communicate that information to the DLR kernel module.
- When the DVS sees a frame with Destination MAC = vMAC, it knows that it is for the DLR, and forwards that frame to the vdrPort.
- After packets arrive in the DLR kernel module via the vdrPort, their metadata is examined to determine the VXLAN VNI or VLAN ID that they belong to. This information is then used to determine which LIF of which DLR instance that packet belongs to.
  - The side effect of this system is that no more than one DLR instance can be connected to a given VLAN or VXLAN.

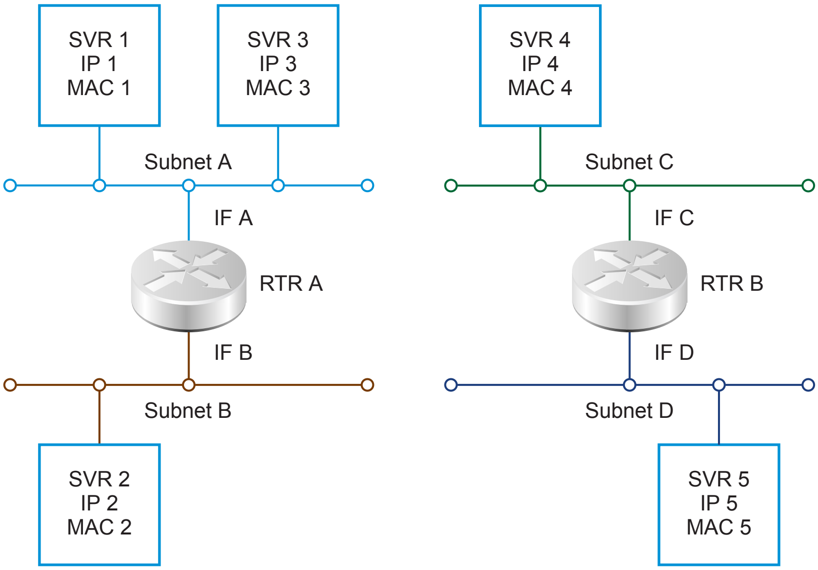
In cases where more than one DLR instance exists, the diagram above would look like this:

Figure 4‑9. Two Hosts with Two DLR Instances



This would correspond to a network topology with two independent routing domains, operating in complete separation from each other, potentially with overlapping IP addresses.

Figure 4‑10. Network Topology Corresponding with Two Hosts and Two DLR Instances



Distributed Routing Subsystem Architecture

DLR instances on ESXi hosts have access to all information needed to perform L3 routing.

- Networks are directly connected (learned from the interfaces' configuration)
- Next hops for each subnet (looked up in routing table)

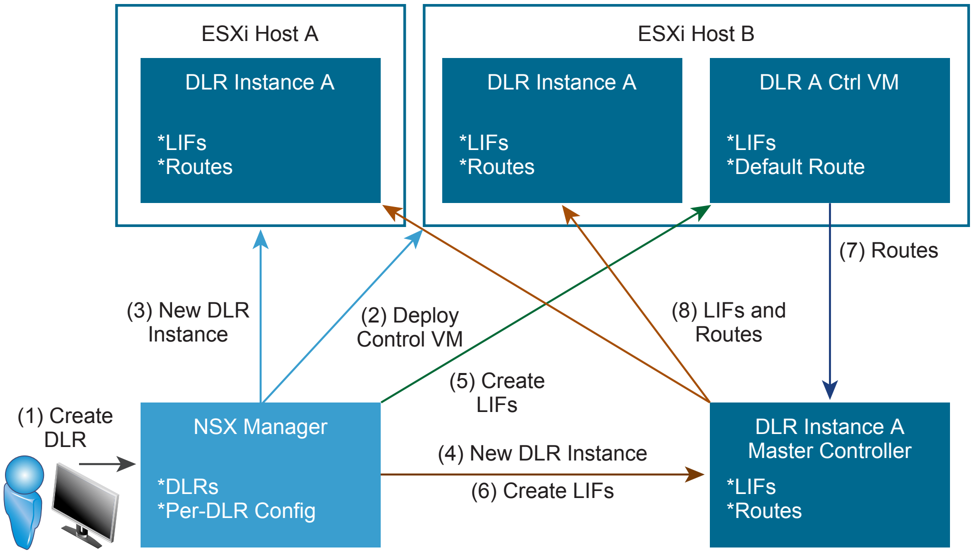
- MAC address to insert into egress frames to reach the next hops (ARP table)

This information is delivered to the instances distributed across multiple ESXi hosts.

DLR Creation Process

The following diagram is a high-level illustration for the process NSX follows to create a new DLR.

Figure 4-11. DLR Creation Process



When a UI wizard is submitted with the “Finish” button or an API call is made to deploy a new DLR, the system processes through the following steps:

1 NSX Manager receives an API call to deploy a new DLR (directly or from vSphere Web Client, invoked by the UI wizard).

2 NSX Manager calls its linked vCenter Server to deploy a DLR Control VM (or a pair, if HA was requested).

  a DLR Control VM is powered on and connects back to the NSX Manager, ready to receive configuration.

  b If an HA pair was deployed, NSX Manager configures an anti-affinity rule that will keep the HA pair running on different hosts. DRS then takes action to move them apart.

3 NSX Manager creates DLR instance on hosts:

  a NSX Manager looks up the logical switches that are to be connected to the new DLR to determine which transport zone they belong to.

  b It then looks up a list of clusters that are configured in this transport zone and creates the new DLR on each host in these clusters.

  c At this point, hosts only know the new DLR ID, but they do not have any corresponding information (LIFs or routes).

4 NSX Manager creates a new DLR instance on the Controller Cluster.

   a Controller Cluster allocates one of the Controller nodes to be the master for this DLR instance.

5 NSX Manager sends the configuration, including LIFs, to the DLR Control VM.

   a ESXi hosts (including the one where the DLR Control VM is running) receive slicing information from the Controller Cluster, determine which Controller node is responsible for the new DLR instance, and connect to the Controller node (if there was no existing connection).

6 After LIF creation on DLR Control VM, the NSX Manager creates the new DLR's LIFs on the Controller Cluster.

7 DLR Control VM connects to the new DLR instance's Controller node, and sends the Controller node the routes:

   a First the DLR translates its routing table into the forwarding table (by resolving prefixes to LIFs).

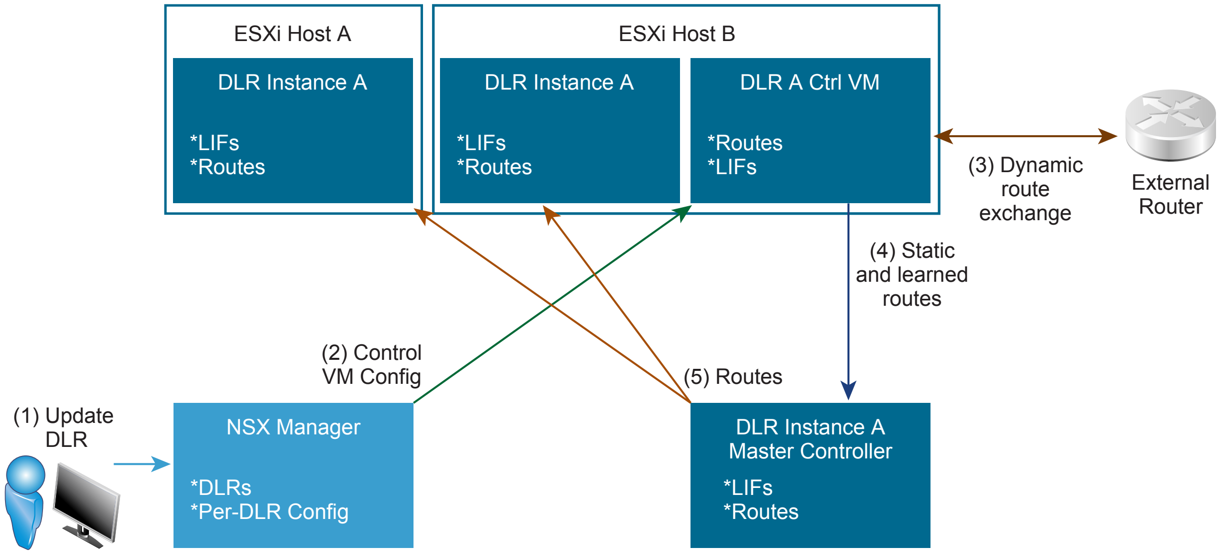
   b Then The DLR sends the resulting table to the Controller node.

8 Controller node pushes LIFs and routes to the other hosts where the new DLR instance exists, via the connection established in step 5.a.

Adding Dynamic Routing to a DLR

When the DLR is created via a "direct" API call (as opposed to using the vSphere Web Client UI), it is possible to supply it with a complete configuration that includes dynamic routing(1).

Figure 4‑12. Dynamic Routing on the DLR



1 The NSX Manager receives an API call to change the existing DLR's configuration, in this case – add dynamic routing.

2 The NSX Manager sends the new configuration to the DLR Control VM.

3 The DLR Control VM applies the configuration and goes through the process of establishing routing adjacencies, exchanging routing information, and so on.

4 After the routing exchange, the DLR Control VM calculates the forwarding table and sends it to the DLR's master Controller node.

5 The DLR's master Controller node then distributes the updated routes to the ESXi hosts where the DLR instance exists.

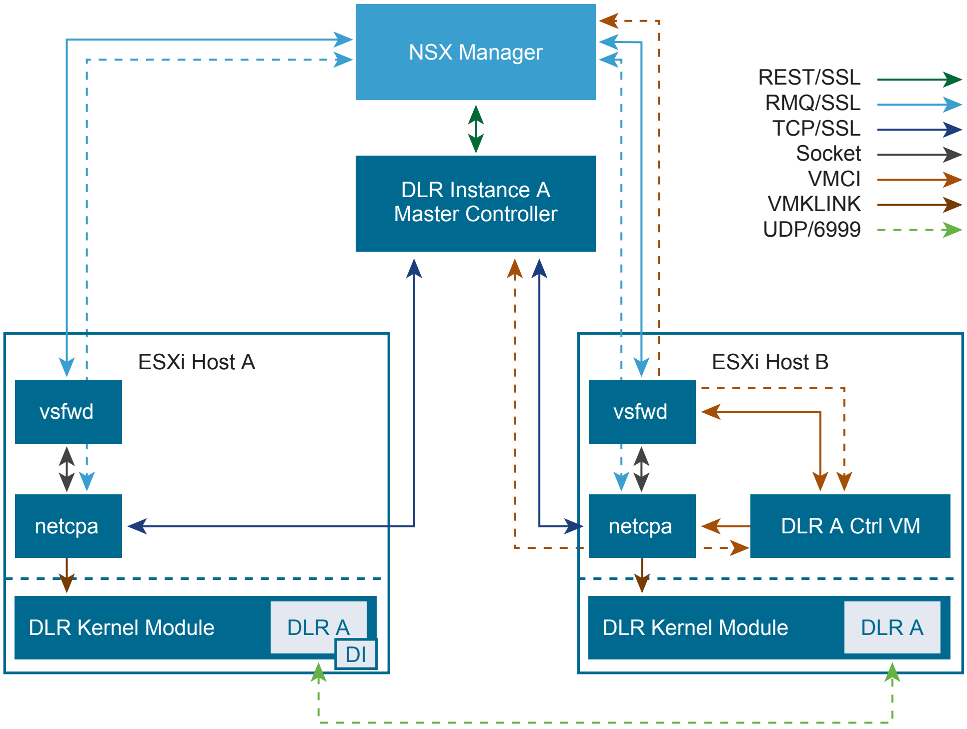
Note that the DLR instance on the ESXi host where the DLR Control VM is running receives its LIFs and routes only from the DLR's master Controller node, never directly from the DLR Control VM or the NSX Manager.

DLR Control and Management Plane Components and Communications

This section provides a brief overview of the components of the DLR control and management planes.

The figure shows the components and the corresponding communication channels between them.

Figure 4-13. DLR Control and Management Plane Components



- NSX Manager:
  - Has direct communications with the Controller Cluster
  - Has a direct permanent connection with the message bus client (vsfwd) process running on each host prepared for NSX
- For each DLR instance, one Controller node (out of the available 3) is elected as master
  - The master function can move to a different Controller node, if the original Controller node fails

- Each ESXi host runs two User World Agents (UWA): message bus client (vsfwd) and control plane agent (netcpa)
    - netcpa requires information from the NSX Manager to function (for example, where to find Controllers and how to authenticate to them); this information is accessed via the message bus connection provided by vsfwd
    - netcpa also communicates with the DLR kernel module to program it with the relevant information it receives from Controllers
- For each DLR instance, there is a DLR Control VM, which is running on one of the ESXi hosts; the DLR Control VM has two communication channels:
    - VMCI channel to the NSX Manager via vsfwd, which is used for configuring the Control VM
    - VMCI channel to the DLR master Controller via netcpa, which is used to send the DLR's routing table to the Controller
- In cases where the DLR has a VLAN LIF, one of the participating ESXi hosts is nominated by the Controller as a designated instance (DI). The DLR kernel module on other ESXi hosts requests that the DI perform proxy ARP queries on the associated VLAN.

NSX Routing Subsystem Components

The NSX routing subsystem is enabled by multiple components.

- NSX Manager
- Cluster of Controllers
- ESXi host modules (kernel and UWA)
- DLR Control VMs
- ESGs

NSX Manager

NSX Manager provides the following functions relevant to NSX routing:

- Acts as a centralized management plane, providing the unified API access point for all NSX management operations
- Installs the Distributed Routing Kernel Module and User World Agents on hosts to prepare them for NSX functions
- Creates/destroys DLRs and DLR LIFs
- Deploys/deletes DLR Control VM and ESG via vCenter
- Configures the Controller Cluster via a REST API and hosts via a message bus:
    - Provides host Control Plane agents with the IP addresses of Controllers
    - Generates and distributes to hosts and controllers the certificates to secure control plane communications

- Configures ESGs and DLR Control VMs via the message bus
  - Note that ESGs can be deployed on unprepared hosts, in which case VIX will be used in lieu of the message bus

Cluster of Controllers

NSX distributed routing requires Controllers, clustered for scale and availability, which provide the following functions:

- Support VXLAN and distributed routing control plane
- Provide CLI interface for statistics and runtime states
- Elect a master controller node for each DLR instance
  - Master node receives routing information from the DLR Control VM and distributes it to the hosts
  - Sends LIF table to the hosts
  - Keeps track of the host where DLR Control VM resides
  - Selects designated instance for VLAN LIFs and communicates this information to hosts; monitors DI host via control plane keepalives (timeout is 30 seconds, and detection time can be 20-40 seconds), sends hosts an update if the selected DI host disappears

ESXi host modules

NSX routing directly utilizes two User World Agents (UWA) and a routing kernel module and also relies on the VXLAN kernel module for VXLAN connectivity.

Here is a summary of what each of these components does:

- Control Plane Agent (netcpa) is a TCP (SSL) client that communicates with the Controller using the control plane protocol. It might connect to multiple controllers. netcpa communicates with the Message Bus Client (vsfwd) to retrieve control plane related information from NSX Manager.
- netcpa packaging and deployment:
  - The agent is packaged into the VXLAN VIB (vSphere installation bundle)
  - Installed by NSX Manager via EAM (ESX Agency Manager) during host preparation
  - Runs as a service daemon on ESXi netcpa
  - Can be started / stopped / queried via its startup script /etc/init.d/netcpad
  - Can be restarted remotely via Networking and Security UI Installation -> Host Preparation -> Installation Status, on individual hosts or on a whole cluster
- DLR Kernel Module (vdrb) integrates with DVS to enable L3 forwarding
  - Configured by netcpa
  - Installed as part of the VXLAN VIB deployment
  - Connects to DVS via the special trunk called “vdrPort," which supports both VLANs and VXLANs

  - Holds information about DLR instances, with per-instance:
    - LIF and Route tables
    - host-local ARP cache
- Message Bus Client (vsfwd) is used by netcpa, ESGs, and DLR Control VMs to communicate with the NSX Manager
  - vsfwd obtains NSX Manager's IP address from /UserVars/RmqIpAddress set by vCenter via vpxa/hosd and logs into the Message Bus server using per-host credentials stored in other /UserVars/Rmq* variables
- netcpa running on an ESXi host relies on vsfwd to do the following:
  - Obtain host's control plane SSL private key and certificate from NSX Manager. These are then stored in /etc/vmware/ssl/rui-for-netcpa.*
  - Get IP addresses and SSL thumbprints of Controllers from NSX Manager. These are then stored in /etc/vmware/netcpa/config-by-vsm.xml.
  - Create and delete DLR instances on its host on instruction from NSX Manager
- Packaging and deployment
  - Same as netcpa, it's a part of the VXLAN VIB
  - Runs as a service daemon on ESXi vsfwd
  - Can be started / stopped / queried via its startup script /etc/init.d/ vShield-Stateful-Firewall
- ESGs and DLR Control VMs use VMCI channel to vsfwd to receive configuration from NSX Manager

DLR Control VMs and ESGs

- DLR Control VM is a "route processor" for its DLR instance
  - Has a "placeholder" or a "real vNIC" interfaces for each DLR LIF, along with IP configuration
  - Can run one of two available dynamic routing protocol (BGP or OSPF) and/or use static routes
  - Requires at least one "Uplink" LIF to be able to run OSPF or BGP
  - Computes forwarding table from directly connected (LIF) subnets, static, and dynamic routes, and sends it via its VMCI link to netcpa to the DLR instance's master Controller
  - Supports HA in Active/Standby VM pair configuration
- ESG is a self-contained router in a VM
  - Completely independent from the NSX DLR routing subsystem (no NSX control plane integration)
  - Typically used as an upstream gateway for one or more DLRs
  - Supports more than one concurrently running dynamic routing protocol

NSX Routing Control Plane CLI

In addition to the host components, NSX Routing employs services of the Controller Cluster and DLR Control VMs, each of which is a source of the DLR control plane information and has its own CLI used to examine it.

DLR Instance Master Controller

Each DLR Instance is served by one of the Controller nodes. The following CLI commands can be used to view information that this Controller node has for the DLR Instance for which it is the master:

```
nsx-controller # show control-cluster logical-routers instance 1460487509
LR-Id      LR-Name            Hosts[]         Edge-Connection Service-Controller
1460487509 default+edge-1     192.168.210.57                  192.168.110.201
                              192.168.210.51
                              192.168.210.52
                              192.168.210.56
                              192.168.110.51
                              192.168.110.52

nsx-controller # show control-cluster logical-routers interface-summary 1460487509
Interface                        Type   Id           IP[]
570d455500000002                 vxlan  5003         192.168.10.2/29
570d45550000000b                 vxlan  5001         172.16.20.1/24
570d45550000000c                 vxlan  5002         172.16.30.1/24
570d45550000000a                 vxlan  5000         172.16.10.1/24

nsx-controller # show control-cluster logical-routers routes 1460487509
LR-Id       Destination        Next-Hop
1460487509  0.0.0.0/0          192.168.10.1
```

- The “instance” sub-command of the “show control-cluster logical-routers” command displays list of hosts that are connected to this Controller for this DLR Instance. In a correctly functioning environment, this list would include all hosts from all clusters where the DLR exists.
- The “interface-summary” displays the LIFs that the Controller learned from the NSX Manager. This information is sent to the hosts.
- The “routes” shows the routing table sent to this Controller by this DLR’s Control VM. Note that unlike on the ESXi hosts, this table does not include any directly connected subnets because this information is provided by the LIF configuration.

DLR Control VM

DLR Control VM has LIFs and routing/forwarding tables. The major output of DLR Control VM's lifecycle is the DLR routing table, which is a product of Interfaces and Routes.

```
edge-1-0> show ip route

Codes: O - OSPF derived, i - IS-IS derived, B - BGP derived,
C - connected, S - static, L1 - IS-IS level-1, L2 - IS-IS level-2,
IA - OSPF inter area, E1 - OSPF external type 1, E2 - OSPF external type 2

Total number of routes: 5

S       0.0.0.0/0           [1/1]         via 192.168.10.1
C       172.16.10.0/24      [0/0]         via 172.16.10.1
C       172.16.20.0/24      [0/0]         via 172.16.20.1
C       172.16.30.0/24      [0/0]         via 172.16.30.1
C       192.168.10.0/29     [0/0]         via 192.168.10.2

edge-1-0> show ip forwarding
Codes: C - connected, R - remote,
       > - selected route, * - FIB route
R>* 0.0.0.0/0 via 192.168.10.1, vNic_2
C>* 172.16.10.0/24 is directly connected, VDR
C>* 172.16.20.0/24 is directly connected, VDR
C>* 172.16.30.0/24 is directly connected, VDR
C>* 192.168.10.0/29 is directly connected, vNic_2
```

- The purpose of the Forwarding Table is to show which DLR interface is chosen as the egress for a given destination subnet.
    - The "VDR" interface is displayed for all LIFs of "Internal" type. The "VDR" interface is a pseudo-interface that does not correspond to a vNIC.

The DLR Control VM's interfaces can be displayed as follows:

```
edge-1-0> show interface
Interface VDR is up, line protocol is up
  index 2 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,NOARP>
  HWaddr: be:3d:a1:52:90:f4
  inet6 fe80::bc3d:a1ff:fe52:90f4/64
  inet 172.16.10.1/24
  inet 172.16.20.1/24
  inet 172.16.30.1/24
  proxy_arp: disabled
  Auto-duplex (Full), Auto-speed (2460Mb/s)
    input packets 0, bytes 0, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
    output packets 0, bytes 0, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
    collisions 0

Interface vNic_0 is up, line protocol is up
```

```
   index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
   HWaddr: 00:50:56:8e:1c:fb
   inet6 fe80::250:56ff:fe8e:1cfb/64
   inet 169.254.1.1/30
   inet 10.10.10.1/24
   proxy_arp: disabled
   Auto-duplex (Full), Auto-speed (2460Mb/s)
     input packets 582249, bytes 37339072, dropped 49, multicast packets 0
     input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
     output packets 4726382, bytes 461202852, dropped 0
     output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
     collisions 0

 Interface vNic_2 is up, line protocol is up
   index 9 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
   HWaddr: 00:50:56:8e:ae:08
   inet 192.168.10.2/29
   inet6 fe80::250:56ff:fe8e:ae08/64
   proxy_arp: disabled
   Auto-duplex (Full), Auto-speed (2460Mb/s)
     input packets 361446, bytes 30167226, dropped 0, multicast packets 361168
     input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
     output packets 361413, bytes 30287912, dropped 0
     output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
     collisions 0
```

Notes of interest:

- Interface “VDR” does not have a VM NIC (vNIC) associated with it. It is a single “pseudo-interface” that is configured with all IP addresses for all DLR’s “Internal” LIFs.
- Interface vNic_0 in this example is the HA interface.
  - The output above was taken from a DLR deployed with HA enabled, and the HA interface is assigned an IP address. This appears as two IP addresses, 169.254.1.1/30 (auto-assigned for HA), and 10.10.10.1/24, manually assigned to the HA interface.
  - On an ESG, the operator can manually assign one of its vNICs as HA, or leave it as default for the system to choose automatically from available “Internal” interfaces. Having the “Internal” type is a requirement, or HA will fail.
- Interface vNic_2 is an Uplink type; therefore, it is represented as a “real” vNIC.
  - Note that the IP address seen on this interface is the same as the DLR’s LIF; however, the DLR Control VM will not answer ARP queries for the LIF IP address (in this case, 192.168.10.2/29). There is an ARP filter applied for this vNIC’s MAC address that makes it happen.
  - The point above holds true until a dynamic routing protocol is configured on the DLR, when the IP address will be removed along with the ARP filter and replaced with the “Protocol IP” address specified during the dynamic routing protocol configuration.
  - This vNIC is used by the dynamic routing protocol running on the DLR Control VM to communicate with the other routers to advertise and learn routes.

NSX Routing Subsystem Failure Modes and Effects

This chapter reviews the typical failure scenarios that might affect components of NSX routing subsystem and outlines the effects of these failures.

NSX Manager

Table 4-2. NSX Manager Faiure Modes and Effects

Failure Mode	Failure Effects
Loss of network connectivity to NSX Manager VM	■ Total outage of all NSX Manager functions, including CRUD for NSX routing/bridging ■ No configuration data loss ■ No data or control-plane outage
Loss of network connectivity between NSX Manager and ESXi hosts or RabbitMQ server failure	■ If DLR Control VM or ESG is running on affected hosts, CRUD operations on them fail ■ Creation and deletion of DLR instances on affected hosts fail ■ No configuration data loss ■ No data or control-plane outage ■ Any dynamic routing updates continue to work
Loss of network connectivity between NSX Manager and Controllers	■ Create, update, and delete operations for NSX distributed routing and bridging fail ■ No configuration data loss ■ No data or control-plane outage
NSX Manager VM is destroyed (datastore failure)	■ Total outage of all NSX Manager functions, including CRUD for NSX routing/bridging ■ Risk of subset of routing/bridging instances becoming orphaned if NSX Manager restored to an older configuration, requiring manual clean-up and reconciliation ■ No data or control-plane outage, unless reconciliation is required

Controller Cluster

Table 4-3. NSX Controller Faiure Modes and Effects

Failure Mode	Failure Effects
Controller cluster loses network connectivity with ESXi hosts	■ Total outage for DLR Control Plane functions (Create, update, and delete routes, including dynamic) ■ Outage for DLR Management Plane functions (Create, update, and delete LIFs on hosts) ■ VXLAN forwarding is affected, which may cause end to end (L2+L3) forwarding process to also fail ■ Data plane continues working based on the last-known state
One or two Controllers lose connectivity with ESXi hosts	■ If affected Controller can still reach other Controllers in the cluster, any DLR instances mastered by this Controller experience the same effects as described above. Other Controllers do not automatically take over
One Controller loses network connectivity with other Controllers (or completely)	■ Two remaining Controllers take over VXLANs and DLRs handled by the isolated Controller ■ Affected Controller goes into Read-Only mode, drop its sessions to hosts, and refuse new ones
Controllers lose connectivity with each other	■ All Controllers will go into Read-Only mode, close connections to hosts, and refuse new ones ■ Create, update, and delete operations for all DLRs' LIFs and routes (including dynamic) fail ■ NSX routing configuration (LIFs) might get out of sync between the NSX Manager and Controller Cluster, requiring manual intervention to resync ■ Hosts will continue operating on last known control plane state
One Controller VM is lost	■ Controller Cluster loses redundancy ■ Management/Control plane continues to operate as normal
Two Controller VMs are lost	■ Remaining Controller will go into read-only mode; effect is the same as when Controllers lose connectivity with each other (above). Likely to require manual cluster recovery

Host Modules

netcpa relies on host SSL key and certificate, plus SSL thumbprints, to establish secure communications with the Controllers. These are obtained from NSX Manager via the message bus (provided by vsfwd).

If certificate exchange process fails, netcpa will not be able to successfully connect to Controllers.

Note: This section doesn’t cover failure of kernel modules, as the effect of this is severe (PSOD) and is a rare occurrence.

Table 4-4. Host Module Faiure Modes and Effects

Failure Mode	Failure Effects
vsfwd uses username/password authentication to access message bus server, which can expire	■ If a vsfwd on a freshly prepared ESXi host cannot reach NSX Manager within two hours, the temporary login/password supplied during installation expires, and message bus on this host becomes inoperable
Effects of failure of the Message Bus Client (vsfwd) depend on the timing.	
If it fails before other parts of NSX control plane had a chance to reach steady running state	■ Distributed routing on the host stops functioning, because the host is not be able to talk to Controllers ■ Host do not learn DLR instances from NSX Manager
If it fails after host has reached steady state	■ ESGs and DLR Control VMs running on the host won’t be able to receive configuration updates ■ Host do not learn of new DLRs, and are not able to delete existing DLRs ■ Host datapath will continue operating based on the configuration host had at the time of failure

Table 4-5. netcpa Faiure Modes and Effects

Failure Mode	Failure Effects
Effects of failure of the Control Plane Agent (netcpa) depend on the timing	
If it fails before NSX datapath kernel modules had a chance to reach steady running state	■ Distributed routing on the host stops functioning
If it fails after host has reached steady state	■ DLR Control VM(s) running on the host will not be able to send their forwarding table updates to Controller(s) ■ Distributed routing datapath will not receive any LIF or route updates from Controller(s), but will continue operating based on the state it had before the failure

DLR Control VM

Table 4-6. DLR Control VM Faiure Modes and Effects

Failure mode	Failure Effects
DLR Control VM is lost or powered off	■ Create, update, and delete operations for this DLR’s LIFs and routes fail ■ Any dynamic route updates will not be sent to hosts (including withdrawal of prefixes received via now broken adjacencies)
DLR Control VM loses connectivity with the NSX Manager and Controllers	■ Same effects as above, except if DLR Control VM and its routing adjacencies are still up, traffic to and from previously learned prefixes will not be affected
DLR Control VM loses connection with the NSX Manager	■ NSX Manager’s Create, update, and delete operations for this DLR’s LIFs and routes fail and are not re-tried ■ Dynamic routing updates continue to propagate
DLR Control VM loses connection with the Controllers	■ Any routing changes (static or dynamic) for this DLR do not propagate to hosts

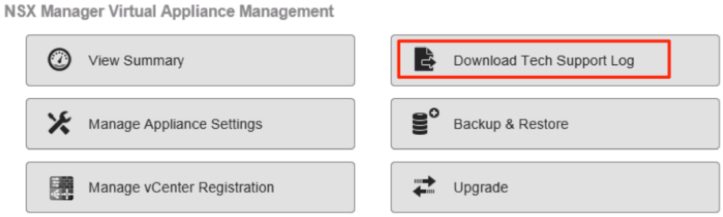
NSX Logs Relevant to Routing

The best practice is to configure all components of NSX to send their logs to a centralized collector, where they can be examined in one place.

If necessary, you can change the log level of NSX components. For more information, see Setting the Logging Level of NSX Components.

NSX Manager Logs

- `show log` in the NSX Manager CLI
- Tech Support Log bundle, collected via the NSX Manager UI


NSX Manager Virtual Appliance Management
View Summary
Download Tech Support Log
Manage Appliance Settings
Backup & Restore
Manage vCenter Registration
Upgrade

The NSX Manager log contains information related to the management plane, which covers create, read, update, and delete (CRUD) operations.

Controller Logs

Controllers contain multiple modules, many with their own log files. Controller logs can be accessed using the `show log <log file> [ filtered-by <string> ]` command. The log files relevant to routing are as follows:

- `cloudnet/cloudnet_java-vnet-controller.<start-time-stamp>.log`
- `cloudnet/cloudnet_cpp.log.INFO`
- `cloudnet/cloudnet_cpp.log.nvp-controller.root.log.INFO.<start-time-stamp>`
- `cloudnet/cloudnet_cpp.log.ERROR` (This file is present if any error occurs.)

Controller logs are verbose and in most cases are only required when the VMware engineering team is brought in to assist with troubleshooting in more difficult cases.

In addition to the `show log` CLI, individual log files can be observed in real time as they are being updated, using the `watch log <logfile> [ filtered-by <string> ]` command.

The logs are included in the Controller support bundle that can be generated and downloaded by selecting a Controller node in the NSX UI and clicking the **Download tech support logs** icon.

ESXi Host Logs

NSX components running on ESXi hosts write several log files:

- VMkernel logs: `/var/log/vmkernel.log`

- Control Plane Agent logs: `/var/log/netcpa.log`
- Message Bus Client logs: `/var/log/vsfwd.log`

The logs can also be collected as part of the VM support bundle generated from vCenter Server.

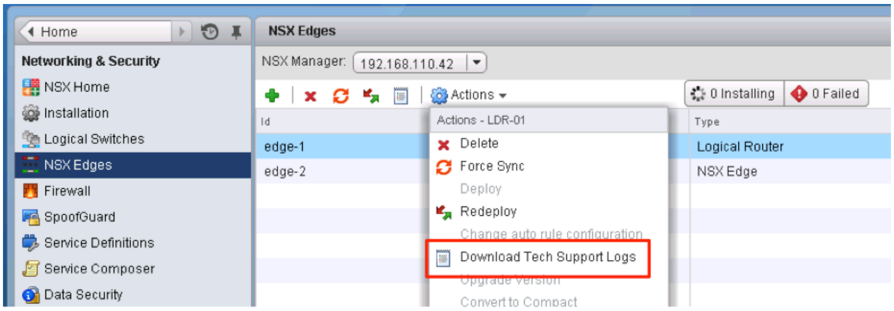
ESG/DLR Control VM Logs

There are two ways to access log files on the ESG and DLR Control VMs—to display them using a CLI or to download the tech support bundle, using the CLI or UI.

The CLI command to display logs is `show log [ follow | reverse ]`.

To download tech-support bundle:

- From the CLI, enter `enable` mode, then run the `export tech-support <[ scp | ftp ]> <URI>` command.
- From the vSphere Web Client, select the **Download Tech Support Logs** option in the **Actions** menu.



Id	Type
edge-1	Logical Router
edge-2	NSX Edge

Other Useful Files and Their Locations

While not strictly logs, there are a number of files that can be helpful in understanding and troubleshooting NSX routing.

- The control plane agent configuration, `/etc/vmware/netcpa/config-by-vsm.xml` contains the information about the following components:
  - Controllers IP addresses, TCP ports, certificate thumbprints, SSL enable/disable
  - dvUplinks on the DVS enabled with VXLAN (teaming policy, names, UUID)
  - DLR instances the host knows about (DLR ID, name)
- The control plane agent configuration, `/etc/vmware/netcpa/netcpa.xml` contains various configuration options for netcpa, including logging level (which by default is **info**).
- Control plane certificate files: `/etc/vmware/ssl/rui-for-netcpa.*`
  - Two files: host certificate and host private key
  - Used for authenticating host connections to Controllers

All of these files are created by netcpa using information it receives from NSX Manager via the message bus connection provided by vsfwd.

Common Failure Scenarios and Fixes

The most common failure scenarios fall into two categories.

They are configuration and control-plane issues. Management plane issues, while possible, are not common.

Configuration Issues and Fixes

Common configuration issues and their effects are described in Table 4-7.

Table 4-7. Common Configuration Issues and Effects

Issues	Effects
Protocol and forwarding IP addresses are reversed for dynamic routing	Dynamic protocol adjacency won't come up
Transport zone is not aligned to the DVS boundary	Distributed routing does not work on a subset of ESXi hosts (those missing from the transport zone)
Dynamic routing protocol configuration mismatch (timers, MTU, BGP ASN, passwords, interface to OSPF area mapping)	Dynamic protocol adjacency does not come up
DLR HA interface is assigned an IP address and redistribution of connected routes is enabled	DLR Control VM might attract traffic for the HA interface subnet and blackhole the traffic

To resolve these issues, review the configuration and correct it as needed.

When necessary, use the `debug ip ospf` or `debug ip bgp` CLI commands and observe logs on the DLR Control VM or on the ESG console (not via SSH session) to detect protocol configuration issues.

Control-Plane Issues and Fixes

Control plane issues seen are often caused by the following issues:

- Host Control Plane Agent (netcpa) being unable to connect to NSX Manager through the message bus channel provided by vsfwd
- Controller cluster having issues with handling the master role for DLR/VXLAN instances

Controller cluster issues related to handling of master roles can often be resolved by restarting one of the NSX Controllers (`restart controller` on the Controller's CLI).

For more information about troubleshooting control-pane issues, see http://kb.vmware.com/kb/2125767.

Gathering Troubleshooting Data

This section provides a summary of the CLI commands that are commonly used for troubleshooting NSX routing.

NSX Manager

Starting in NSX 6.2, commands that were formerly run from the NSX Controller and other NSX components to troubleshoot NSX routing are now run directly from the NSX Manager.

- List of DLR instances
- List of LIFs for each DLR instance
- List of Routes for each DLR instance
- List of MAC addresses for each DLR bridging instance
- Interfaces
- Routing and forwarding tables
- State of dynamic routing protocols (OSPF or BGP)
- Configuration sent to the DLR Control VM or ESG by the NSX Manager

DLR Control VM and ESG

The DLR Control VM and ESG provide functionality to capture packets on their interfaces. Packet capture can assist with troubleshooting routing protocol problems.

1 Run `show interfaces` to list the interface names.

2 Run `debug packet [ display | capture ] interface <interface name>`.

   - If using capture, packets are saved into a `.pcap` file.

3 Run `debug show files` to list saved capture files.

4 Run `debug copy [ scp | ftp ] ...` to download captures for offline analysis.

```
dlr-01-0> debug packet capture interface vNic_2
tcpdump: listening on vNic_2, link-type EN10MB (Ethernet), capture size 65535 bytes
43 packets captured
48 packets received by filter
0 packets dropped by kernel
```

```
dlr-01-0> debug show files
total 4.0K
-rw------- 1 3.6K Mar 30 23:49 tcpdump_vNic_2.0
```

```
dlr-01-0> debug copy
  scp  use scp to copy
  ftp  use ftp to copy
```

```
dlr-01-0> debug copy scp
  URL  user@<remote-host>:<path-to>
```

The `debug packet` command uses tcpdump in the background and can accept filtering modifiers formatted in like tcpdump filtering modifiers on UNIX. The only consideration is that any white spaces in the filter expression need to be replaced with underscores ("_").

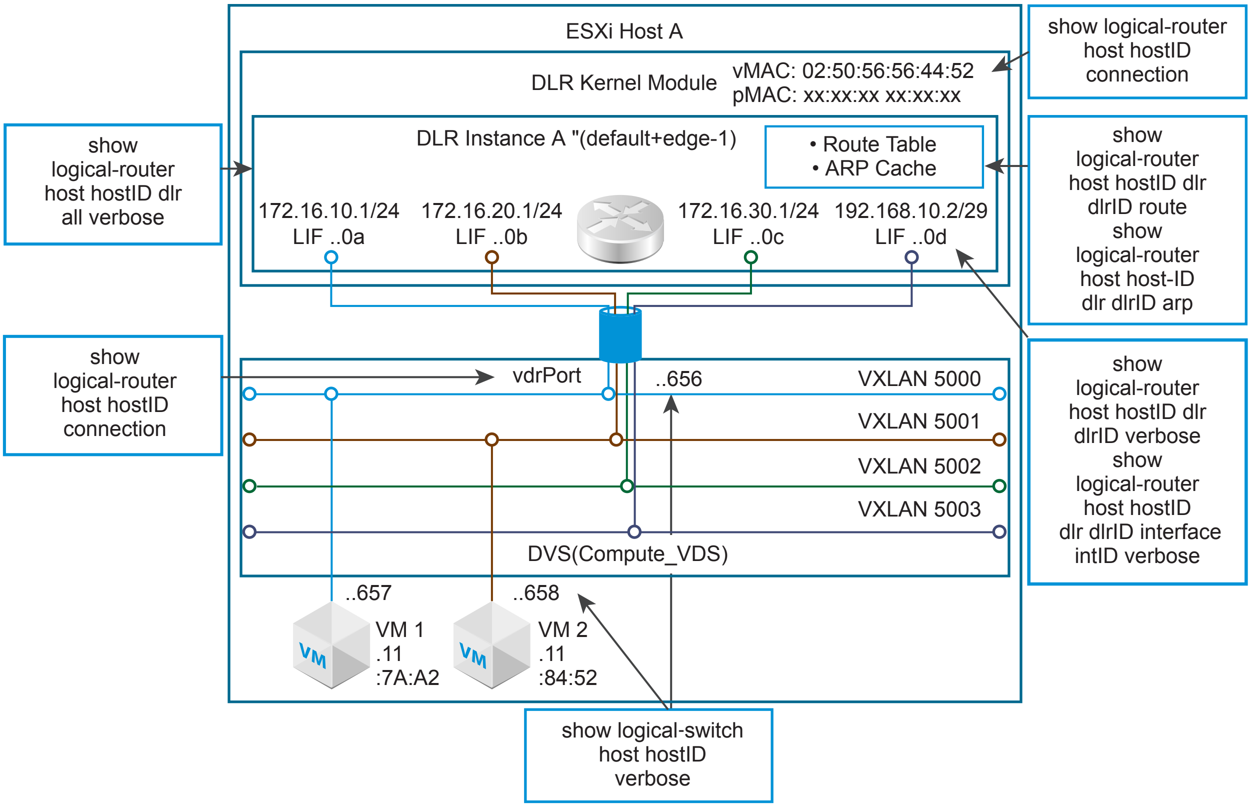
For example, the following command displays all traffic through vNic_0 except SSH, to avoid looking at the traffic belonging to the interactive session itself.

```
plr-02-0> debug packet display interface vNic_0 port_not_22
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on vNic_0, link-type EN10MB (Ethernet), capture size 65535 bytes
04:10:48.197768 IP 192.168.101.3.179 > 192.168.101.2.25698: Flags [P.], seq 4191398894:4191398913, ack
2824012766, win 913, length 19: BGP, length: 19
04:10:48.199230 IP 192.168.101.2.25698 > 192.168.101.3.179: Flags [.], ack 19, win 2623, length 0
04:10:48.299804 IP 192.168.101.2.25698 > 192.168.101.3.179: Flags [P.], seq 1:20, ack 19, win 2623,
length 19: BGP, length: 19
04:10:48.299849 IP 192.168.101.3.179 > 192.168.101.2.25698: Flags [.], ack 20, win 913, length 0
04:10:49.205347 IP 192.168.101.3.179 > 192.168.101.2.25698: Flags [P.], seq 19:38, ack 20, win 913,
length 19: BGP, length: 19
```

ESXi Hosts

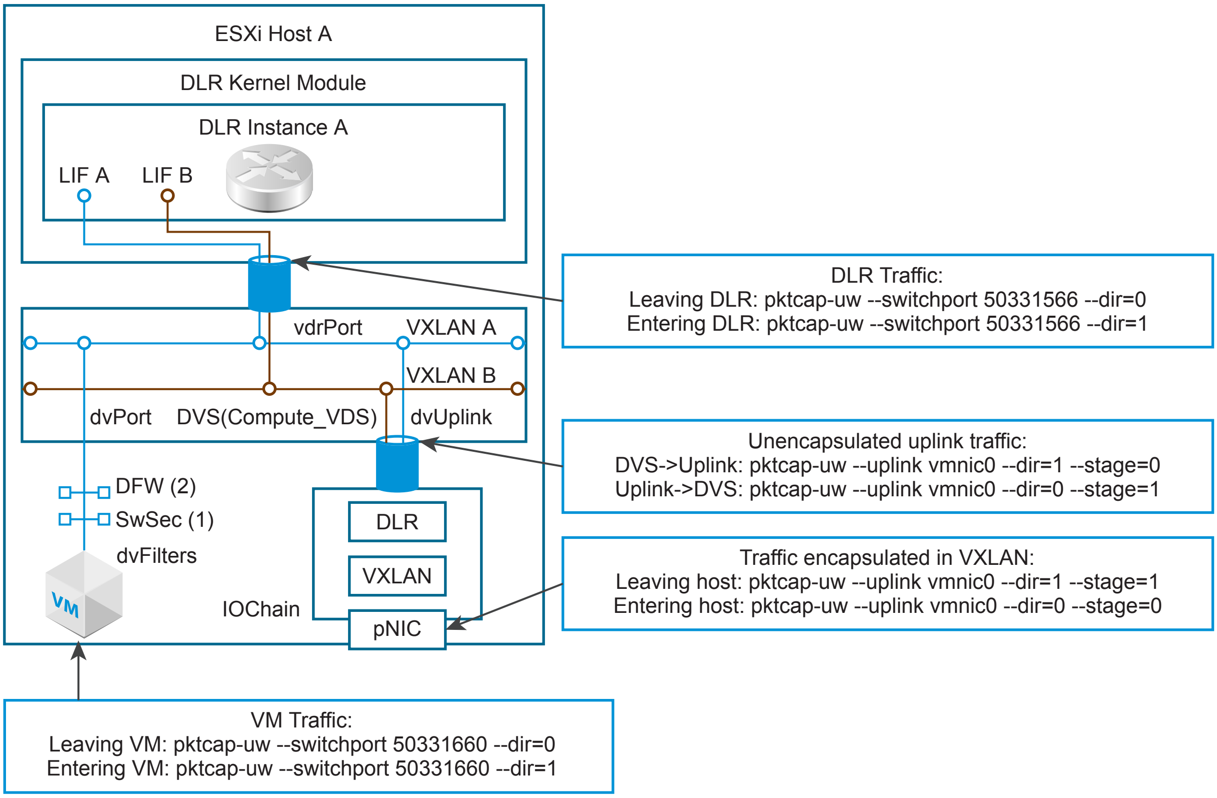
Hosts are closely connected to NSX Routing. Figure 4-14 shows visually the components participating in the routing subsystem and the NSX Manager CLI commands used to display information about them:

Figure 4-14. Host Components Related to Troubleshooting NSX Routing


ESXi Host A
DLR Kernel Module
vMAC: 02:50:56:56:44:52
pMAC: xx:xx:xx xx:xx:xx
DLR Instance A "(default+edge-1)
• Route Table
• ARP Cache
172.16.10.1/24
LIF ..0a
172.16.20.1/24
LIF ..0b
172.16.30.1/24
LIF ..0c
192.168.10.2/29
LIF ..0d
vdrPort
..656
VXLAN 5000
VXLAN 5001
VXLAN 5002
VXLAN 5003
DVS(Compute_VDS)
..657
..658
VM
VM 1
.11
:7A:A2
VM
VM 2
.11
:84:52
show logical-router host hostID connection
show logical-router host hostID dlr dlrID route
show logical-router host host-ID dlr dlrID arp
show logical-router host hostID dlr dlrID verbose
show logical-router host hostID dlr dlrID interface intID verbose
show logical-router host hostID dlr all verbose
show logical-router host hostID connection
show logical-switch host hostID verbose

Packets captured in the datapath can assist with identifying problems at various stages of packet forwarding. Figure 4-15 covers the major capture points and respective CLI command to use.

Figure 4-15. Capture Points and Related CLI Commands



DLR Traffic:
Leaving DLR: pktcap-uw --switchport 50331566 --dir=0
Entering DLR: pktcap-uw --switchport 50331566 --dir=1

Unencapsulated uplink traffic:
DVS->Uplink: pktcap-uw --uplink vmnic0 --dir=1 --stage=0
Uplink->DVS: pktcap-uw --uplink vmnic0 --dir=0 --stage=1

Traffic encapsulated in VXLAN:
Leaving host: pktcap-uw --uplink vmnic0 --dir=1 --stage=1
Entering host: pktcap-uw --uplink vmnic0 --dir=0 --stage=0

VM Traffic:
Leaving VM: pktcap-uw --switchport 50331660 --dir=0
Entering VM: pktcap-uw --switchport 50331660 --dir=1

Edge Appliance Troubleshooting 5

This topic provides information for understanding and troubleshooting the VMware NSX Edge appliance.

To troubleshoot issues with an NSX Edge appliance, validate that each troubleshooting step below is true for your environment. Each step provides instructions or a link to a document, to eliminate possible causes and take corrective action as necessary. The steps are ordered in the most appropriate sequence to isolate the issue and identify the proper resolution. Do not skip a step.

Check the release notes for current releases to see if the problem is resolved.

Ensure that the minimum system requirements are met when installing VMware NSX Edge. See the *NSX Installation Guide*.

Installation and Upgrade issues

- Verify that the issue you are encountering is not related to the "Would Block" issue. For more information, see https://kb.vmware.com/kb/2107951.
- If the upgrade or redeploy succeeds but there is no connectivity for the Edge interface, verify connectivity on the back-end Layer 2 switch. See https://kb.vmware.com/kb/2135285.
- If deployment or upgrade of the Edge fails with the error:

  ```
  /sbin/ifconfig vNic_1 up failed : SIOCSIFFLAGS: Invalid argument
  ```

  OR
- If the deployment or upgrade succeeds, but there is no connectivity on the Edge interfaces:
- Running the `show interface` command as well as Edge Support logs displays entries similar to:

  ```
  vNic_0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen 1000
      link/ether 00:50:56:32:05:03 brd ff:ff:ff:ff:ff:ff
      inet 21.12.227.244/23 scope global vNic_0
      inet6 fe80::250:56ff:fe32:503/64 scope link tentative dadfailed
         valid_lft forever preferred_lft forever
  ```

  In both cases, the host switch is not ready or has some issues. To resolve, investigate the host switch.

Configuration Issues

- Collect the NSX Edge diagnostic information. See https://kb.vmware.com/kb/2079380.

  Filter the NSX Edge logs by searching for the string `vse_die`. The logs near this string might provide information about the configuration error.

Firewall Issues

- If there are inactivity time-out issues and you are noticing that applications are idle for a long time, increase inactivity-timeout settings using the REST API. See https://kb.vmware.com/kb/2101275.

Edge Firewall Packet Drop Issues

1 Check the firewall rules table with the `show firewall` command. The `usr_rules` table displays the configured rules.

```
nsxedge> show firewall
Chain PREROUTING (policy ACCEPT 3146M packets, 4098G bytes)
rid    pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
rid    pkts bytes target     prot opt in     out     source               destination
0     78903   16M ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
0         0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
state INVALID
0      140K 9558K block_in   all  --  *      *       0.0.0.0/0            0.0.0.0/0
0     23789 1184K ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
state RELATED,ESTABLISHED
0      116K 8374K usr_rules  all  --  *      *       0.0.0.0/0            0.0.0.0/0
0         0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 3146M packets, 4098G bytes)
rid    pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 173K packets, 22M bytes)
rid    pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
rid    pkts bytes target     prot opt in     out     source               destination
0     78903   16M ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0
0      679K   41M DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
state INVALID
0     3146M 4098G block_out  all  --  *      *       0.0.0.0/0            0.0.0.0/0
0         0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
PHYSDEV match --physdev-in tap0 --physdev-out vNic_+
0         0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
PHYSDEV match --physdev-in vNic_+ --physdev-out tap0
0         0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
PHYSDEV match --physdev-in na+ --physdev-out vNic_+
0         0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

```
PHYSDEV match --physdev-in vNic_+ --physdev-out na+
0     3145M 4098G ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
state RELATED,ESTABLISHED
0      221K   13M usr_rules  all  --  *      *       0.0.0.0/0            0.0.0.0/0
0         0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain block_in (1 references)
rid    pkts bytes target     prot opt in     out     source               destination

Chain block_out (1 references)
rid    pkts bytes target     prot opt in     out     source               destination

Chain usr_rules (2 references)
rid    pkts bytes target     prot opt in     out     source               destination
131074 70104 5086K ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
match-set 0_131074-os-v4-1 src
131075  116K 8370K ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
match-set 1_131075-ov-v4-1 dst
131073  151K 7844K ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

Check for an incrementing value of a `DROP invalid` rule in the `POST_ROUTING` section of the `show firewall` command. Typical reasons include asymmetric routing issues or TCP-based applications that have been inactive for more than one hour. Further evidence of asymmetric routing issues include:

- Ping works in one direction and fails in the other direction
- Ping works, while TCP does not work

2 Collect the `show ipset` command output.

```
nsxedge> show ipset
Name: 0_131074-os-v4-1
Type: bitmap:if (Interface Match)
Revision: 3
Header: range 0-64000
Size in memory: 8116
References: 1
Number of entries: 1
Members:
vse (vShield Edge Device)

Name: 0_131074-os-v6-1
Type: bitmap:if (Interface Match)
Revision: 3
Header: range 0-64000
Size in memory: 8116
References: 1
Number of entries: 1
Members:
vse (vShield Edge Device)

Name: 1_131075-ov-v4-1
Type: hash:oservice (Match un-translated Ports)
Revision: 2
```

```
Header: family inet hashsize 64 maxelem 65536
Size in memory: 704
References: 1
Number of entries: 2
Members:
Proto=6, DestPort=179, SrcPort=Any    (encoded: 0.6.0.179,0.6.0.0/16)
Proto=89, DestPort=Any, SrcPort=Any    (encoded: 0.89.0.0/16,0.89.0.0/16)

Name: 1_131075-ov-v6-1
Type: hash:oservice (Match un-translated Ports)
Revision: 2
Header: family inet hashsize 64 maxelem 65536
Size in memory: 704
References: 1
Number of entries: 2
Members:
Proto=89, DestPort=Any, SrcPort=Any    (encoded: 0.89.0.0/16,0.89.0.0/16)
Proto=6, DestPort=179, SrcPort=Any    (encoded: 0.6.0.179,0.6.0.0/16)
```

3 Enable logging on a particular firewall rule using the REST API or the Edge user interface, and monitor the logs with the `show log follow` command.

   If logs are not seen, enable logging on the `DROP Invalid` rule using the following REST API.

```
URL : https://NSX_Manager_IP/api/4.0/edges/{edgeId}/firewall/config/global

PUT Method
Input representation
<globalConfig>   <!-- Optional -->
<tcpPickOngoingConnections>false</tcpPickOngoingConnections>   <!-- Optional. Defaults to false -->
<tcpAllowOutOfWindowPackets>false</tcpAllowOutOfWindowPackets>    <!-- Optional. Defaults to false
-->
<tcpSendResetForClosedVsePorts>true</tcpSendResetForClosedVsePorts>    <!-- Optional. Defaults to
true -->
<dropInvalidTraffic>true</dropInvalidTraffic>    <!-- Optional. Defaults to true -->
<logInvalidTraffic>true</logInvalidTraffic>     <!-- Optional. Defaults to false -->
<tcpTimeoutOpen>30</tcpTimeoutOpen>       <!-- Optional. Defaults to 30 -->
<tcpTimeoutEstablished>3600</tcpTimeoutEstablished>   <!-- Optional. Defaults to 3600 -->
<tcpTimeoutClose>30</tcpTimeoutClose>   <!-- Optional. Defaults to 30 -->
<udpTimeout>60</udpTimeout>             <!-- Optional. Defaults to 60 -->
<icmpTimeout>10</icmpTimeout>           <!-- Optional. Defaults to 10 -->
<icmp6Timeout>10</icmp6Timeout>           <!-- Optional. Defaults to 10 -->
<ipGenericTimeout>120</ipGenericTimeout>    <!-- Optional. Defaults to 120 -->
</globalConfig>
Output representation
No payload
```

Use the `show log follow` command to look for logs similar to:

```
2016-04-18T20:53:31+00:00 edge-0 kernel: nf_ct_tcp: invalid TCP flag combination IN= OUT=
SRC=172.16.1.4 DST=192.168.1.4 LEN=40 TOS=0x00 PREC=0x00 TTL=64 ID=43343 PROTO=TCP
SPT=5050 DPT=80 SEQ=0 ACK=1572141176 WINDOW=512 RES=0x00 URG PSH FIN URGP=0
2016-04-18T20:53:31+00:00 edge-0 kernel: INVALID IN= OUT=vNic_1 SRC=172.16.1.4
DST=192.168.1.4 LEN=40 TOS=0x00 PREC=0x00 TTL=63 ID=43343 PROTO=TCP SPT=5050 DPT=80
WINDOW=512 RES=0x00 URG PSH FIN URGP=0
```

4. Check for matching connections in the Edge firewall state table with the `show flowtable rule_id` command:

   ```
   nsxedge> show flowtable
   1: tcp  6 21554 ESTABLISHED src=192.168.110.10 dst=192.168.5.3 sport=25981
   d port=22 pkts=52 bytes=5432 src=192.168.5.3 dst=192.168.110.10 sport=22 dport=259
   81 pkts=44 bytes=7201 [ASSURED] mark=0 rid=131073 use=1
   2: tcp  6 21595 ESTABLISHED src=127.0.0.1 dst=127.0.0.1 sport=53194
   dport=10 001 pkts=33334 bytes=11284650 src=127.0.0.1 dst=127.0.0.1 sport=10001 dport=5319
   4 pkts=33324 bytes=1394146 [ASSURED] mark=0 rid=0 use=1
   ```

   Compare the active connection count and the maximum allowed count with the `show flowstats` command:

   ```
   nsxedge> show flowstats
   Total Flow Capacity: 65536
   Current Statistics :
   cpu=0 searched=3280373 found=3034890571 new=52678 invalid=659946 ignore=77605
   delete=52667 delete_list=49778 insert=49789 insert_failed=0 drop=0 early_drop=0
   error=0 search_restart=0
   ```

5. Check the Edge logs with the `show log follow` command, and look for any ALG drops. Search for strings similar to `tftp_alg`, `msrpc_alg`, or `oracle_tns`. For additional information, see:

   - https://kb.vmware.com/kb/2126674
   - https://kb.vmware.com/kb/2137751

Edge Routing Connectivity issues

1. Initiate controlled traffic from a client using the `ping <destination_IP_address>` command.
2. Capture traffic simultaneously on both interfaces, write the output to a file, and export it using SCP.

   For example:

   Capture the traffic on the ingress interface with this command:

   ```
   debug packet display interface vNic_0 -n_src_host_1.1.1.1
   ```

   Capture the traffic on the egress interface with this command:

   ```
   debug packet display interface vNic_1 -n_src_host_1.1.1.1
   ```

For simultaneous packet capture, use the ESXi packet capture utility `pktcap-uw` tool in ESXi. See https://kb.vmware.com/kb/2051814.

If the packet drops are consistent, check for configuration errors related to:

- IP addresses and routes
- Firewall rules or NAT rules
- Asymmetric routing
- RP filter checks

a Check interface IP/subnets with the `show interface` command.

b If there are missing routes at the data plane, run these commands:

  - `show ip route`
  - `show ip route static`
  - `show ip route bgp`
  - `show ip route ospf`

c Check the routing table for needed routes by running the `show ip forwarding` command.

d If you have multiple paths, run the `show rpfilter` command.

```
nsxedge> show rpfilter
net.ipv4.conf.VDR.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.br-sub.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.lo.rp_filter = 0
net.ipv4.conf.vNic_0.rp_filter = 1
net.ipv4.conf.vNic_1.rp_filter = 1
net.ipv4.conf.vNic_2.rp_filter = 1
net.ipv4.conf.vNic_3.rp_filter = 1
net.ipv4.conf.vNic_4.rp_filter = 1
net.ipv4.conf.vNic_5.rp_filter = 1
net.ipv4.conf.vNic_6.rp_filter = 1
net.ipv4.conf.vNic_7.rp_filter = 1
net.ipv4.conf.vNic_8.rp_filter = 1
net.ipv4.conf.vNic_9.rp_filter = 1

nsxedge> show rpfstats
RPF drop packet count: 484
```

To check for RPF statistics, run the `show rpfstats` command.

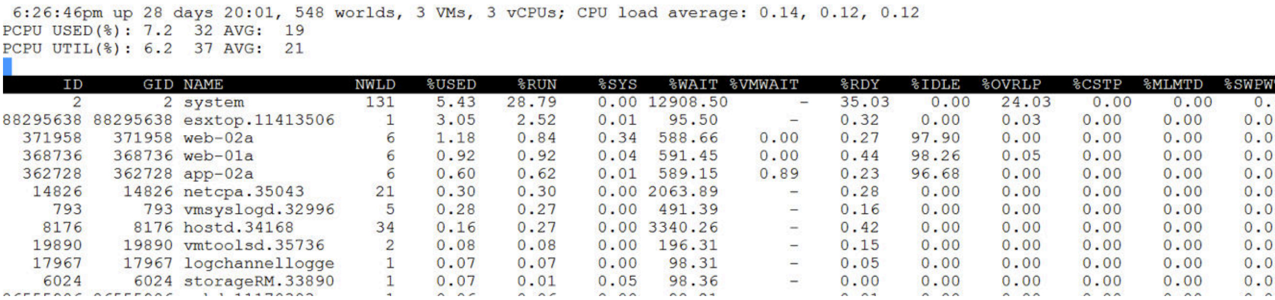
```
nsxedge> show rpfstats
RPF drop packet count: 484
```

If the packet drops appear randomly, check for resource limitations:

a For CPU or memory usage, run these commands:

- `show system cpu`
- `show system memory`
- `show system storage`
- `show process monitor`
- `top`

For ESXi, run the `esxtop n` command.



```
 6:26:46pm up 28 days 20:01, 548 worlds, 3 VMs, 3 vCPUs; CPU load average: 0.14, 0.12, 0.12
PCPU USED(%): 7.2  32 AVG:  19
PCPU UTIL(%): 6.2  37 AVG:  21

       ID      GID NAME               NWLD   %USED    %RUN    %SYS   %WAIT %VMWAIT    %RDY   %IDLE  %OVRLP   %CSTP  %MLMTD  %SWPW
        2        2 system              131    5.43   28.79    0.00 12908.50      -   35.03    0.00   24.03    0.00    0.00    0.
 88295638 88295638 esxtop.11413506       1    3.05    2.52    0.01   95.50       -    0.32    0.00    0.03    0.00    0.00    0.0
   371958   371958 web-02a               6    1.18    0.84    0.34  588.66    0.00    0.27   97.90    0.00    0.00    0.00    0.0
   368736   368736 web-01a               6    0.92    0.92    0.04  591.45    0.00    0.44   98.26    0.05    0.00    0.00    0.0
   362728   362728 app-02a               6    0.60    0.62    0.01  589.15    0.89    0.23   96.68    0.00    0.00    0.00    0.0
    14826    14826 netcpa.35043         21    0.30    0.30    0.00 2063.89       -    0.28    0.00    0.00    0.00    0.00    0.0
      793      793 vmsyslogd.32996       5    0.28    0.27    0.00  491.39       -    0.16    0.00    0.00    0.00    0.00    0.0
     8176     8176 hostd.34168          34    0.16    0.27    0.00 3340.26       -    0.42    0.00    0.00    0.00    0.00    0.0
    19890    19890 vmtoolsd.35736        2    0.08    0.08    0.00  196.31       -    0.15    0.00    0.00    0.00    0.00    0.0
    17967    17967 logchannellogge       1    0.07    0.07    0.00   98.31       -    0.05    0.00    0.00    0.00    0.00    0.0
     6024     6024 storageRM.33890       1    0.07    0.01    0.05   98.36       -    0.00    0.00    0.00    0.00    0.00    0.0
```

High CPU Utilization

If you are experiencing high CPU utilization on the NSX Edge, verify the performance of the appliance using the `esxtop` command on the ESXi host. Review the following Knowledge Base articles:

- https://kb.vmware.com/kb/1008205
- https://kb.vmware.com/kb/1008014
- https://kb.vmware.com/kb/1010071
- https://kb.vmware.com/kb/2096171

Also see https://communities.vmware.com/docs/DOC-9279.

A high value for the `ksoftirqd` process indicates a high incoming packet rate. Check whether logging is enabled on the data path, such as for firewall rules. Run the `show log follow` command to determine whether a large number of log hits are being recorded.

NSX Manager and Edge Communication Issues

The NSX Manager communicates with NSX Edge through the VIX or Message Bus. It is chosen by the NSX Manager when the Edge is deployed and never changes.

VIX

- VIX is used for NSX Edge if the ESXi host is not prepared.
- The NSX Manager gets host credentials from the vCenter Server to connect to the ESXi host first.

- The NSX Manager uses the Edge credentials to log in to the Edge appliance.
- The `vmtoolsd` process on the Edge handles the VIX communication.

VIX failures occur because of:

- The NSX Manager cannot communicate with the vCenter Server.
- The NSX Manager cannot communicate with the ESXi hosts.
- There are NSX Manager internal issues.
- There are Edge internal issues.

VIX Debugging

Check for VIX errors VIX_E_<error> in the NSX Manager logs to narrow down the cause. Look for errors similar to:

```
Vix Command 1126400 failed, reason com.vmware.vshield.edge.exception.VixException: vShield
Edge:10013:Error code 'VIX_E_FILE_NOT_FOUND' was returned by VIX API.:null

Health check failed for edge  edge-13 VM vm-5025 reason:
com.vmware.vshield.edge.exception.VixException: vShield Edge:10013:Error code
'VIX_E_VM_NOT_RUNNING' was returned by VIX API.:null
```

In general, if the same failure occurs for many Edges at the same time, the issue is not on the Edge side.

Edge Diagnosis

- Check if `vmtoolsd` is running with this command:

  ```
  nsxedge> show process list
  Perimeter-Gateway-01-0> show process list
  %CPU %MEM    VSZ   RSZ STAT  STARTED     TIME COMMAND
   0.0  0.1   4244   720 Ss     May 16 00:00:15 init [3]
  ...
   0.0  0.1   4240   640 S      May 16 00:00:00 logger -p daemon debug -t vserrdd
   0.2  0.9  57192  4668 S      May 16 00:23:07 /usr/local/bin/vmtoolsd --plugin-pa
   0.0  0.4   4304  2260 SLs    May 16 00:01:54 /usr/sbin/watchdog
   ...
  ```

- Check if Edge is in a good state by running this command:

  ```
  nsxedge> show eventmgr
  -----------------------
  messagebus     : enabled
  debug          : 0
  profiling      : 0
  cfg_rx         : 1
  cfg_rx_msgbus  : 0
  ...
  ```

Also, you can use the `show eventmgr` command to verify that the query command is received and processed.

```
nsxedge> show eventmgr
-----------------------
messagebus     : enabled
debug          : 0
profiling      : 0
cfg_rx         : 1
cfg_rx_msgbus  : 0
cfg_rx_err     : 0
cfg_exec_err   : 0
cfg_resp       : 0
cfg_resp_err   : 0
cfg_resp_ln_err: 0
fastquery_rx   : 0
fastquery_err  : 0
clearcmd_rx    : 0
clearcmd_err   : 0
ha_rx          : 0
ha_rx_err      : 0
ha_exec_err    : 0
status_rx      : 16
status_rx_err  : 0
status_svr     : 10
status_evt     : 0
status_evt_push: 0
status_ha      : 0
status_ver     : 1
status_sys     : 5
status_cmd     : 0
status_svr_err : 0
status_evt_err : 0
status_sys_err : 0
status_ha_err  : 0
status_ver_err : 0
status_cmd_err : 0
evt_report     : 1
evt_report_err : 0
hc_report      : 10962
hc_report_err  : 0
cli_rx         : 2
cli_resp       : 1
cli_resp_err   : 0
counter_reset  : 0
---------- Health Status -------------
system status  : good
ha state       : active
cfg version    : 7
generation     : 0
server status  : 1
syslog-ng      : 1
haproxy        : 0
ipsec          : 0
```

```
sslvpn         : 0
l2vpn          : 0
dns            : 0
dhcp           : 0
heartbeat      : 0
monitor        : 0
gslb           : 0
---------- System Events -------------
```

If the `vmtoolsd` is not running or the Edge is in a bad state, reboot the Edge.

You can also check the Edge logs. See https://kb.vmware.com/kb/2079380.

Message Bus Debugging

The Message Bus is used for NSX Edge communication when ESXi hosts are prepared. When you encounter issues, the NSX Manager logs might contain entries similar to:

```
GMT ERROR taskScheduler-6 PublishTask:963 - Failed to configure VSE-vm index 0, vm-id vm-117,
edge edge-5. Error: RPC request timed out
```

This issue occurs if:

- Edge is in a bad state
- Message Bus connection is broken

To diagnose the issue on the Edge:

- To check rmq connectivity, run this command:

```
nsxedge> show messagebus messages
-----------------------
Message bus is enabled
cmd conn state : listening
init_req       : 1
init_resp      : 1
init_req_err   : 0
...
```

- To check vmci connectivity, run this command:

```
nsxedge> show messagebus forwarder
-----------------------
Forwarder Command Channel
vmci_conn          : up
app_client_conn    : up
vmci_rx            : 3649
vmci_tx            : 3648
vmci_rx_err        : 0
vmci_tx_err        : 0
vmci_closed_by_peer: 8
vmci_tx_no_socket  : 0
app_rx             : 3648
```

```
app_tx            : 3649
app_rx_err        : 0
app_tx_err        : 0
app_conn_req      : 1
app_closed_by_peer : 0
app_tx_no_socket   : 0
-----------------------
Forwarder Event Channel
vmci_conn          : up
app_client_conn    : up
vmci_rx            : 1143
vmci_tx            : 13924
vmci_rx_err        : 0
vmci_tx_err        : 0
vmci_closed_by_peer: 0
vmci_tx_no_socket  : 0
app_rx             : 13924
app_tx             : 1143
app_rx_err         : 0
app_tx_err         : 0
app_conn_req       : 1
app_closed_by_peer : 0
app_tx_no_socket   : 0
-----------------------
cli_rx             : 1
cli_tx             : 1
cli_tx_err         : 0
counters_reset     : 0
```

In the example, the output `vmci_closed_by_peer: 8` indicates the number of times the connection has been closed by the host agent. If this number is increasing and `vmci conn` is down, the host agent cannot connect to the RMQ broker. In `show log follow`, look for repeated errors in the Edge logs: `VmciProxy: [daemon.debug] VMCI Socket is closed by peer`

To diagnose the issue on the ESXi host:

- To check if the ESXi host connects to the RMQ broker, run this command:

```
esxcli network ip connection list | grep 5671

tcp   0   0  10.32.43.4:43329  10.32.43.230:5671    ESTABLISHED     35854  newreno  vsfwd
tcp   0   0  10.32.43.4:52667  10.32.43.230:5671    ESTABLISHED     35854  newreno  vsfwd
tcp   0   0  10.32.43.4:20808  10.32.43.230:5671    ESTABLISHED     35847  newreno  vsfwd
tcp   0   0  10.32.43.4:12486  10.32.43.230:5671    ESTABLISHED     35847  newreno  vsfwd
```

Displaying Packet Drop Statistics

Starting with NSX for vSphere 6.2.3, you can use the command `show packet drops` to displays packet drop statistics for the following:

- Interface

- Driver
- L2
- L3
- Firewall

To run the command, log in to the NSX Edge CLI and enter basic mode. For more information, see the *NSX Command Line Interface Reference*. For example:

```
show packet drops

vShield Edge Packet Drop Stats:

Driver Errors
=============
          TX      TX    TX   RX   RX      RX
Interface Dropped Error Ring Full Dropped Error Out Of Buf
vNic_0    0       0     0    0    0       0
vNic_1    0       0     0    0    0       0
vNic_2    0       0     0    0    0       2
vNic_3    0       0     0    0    0       0
vNic_4    0       0     0    0    0       0
vNic_5    0       0     0    0    0       0

Interface Drops
===============
Interface RX Dropped TX Dropped
vNic_0             4          0
vNic_1          2710          0
vNic_2             0          0
vNic_3             2          0
vNic_4             2          0
vNic_5             2          0

L2 RX Errors
============
Interface length crc frame fifo missed
vNic_0         0   0     0    0      0
vNic_1         0   0     0    0      0
vNic_2         0   0     0    0      0
vNic_3         0   0     0    0      0
vNic_4         0   0     0    0      0
vNic_5         0   0     0    0      0

L2 TX Errors
============
Interface aborted fifo window heartbeat
vNic_0          0    0      0         0
vNic_1          0    0      0         0
vNic_2          0    0      0         0
vNic_3          0    0      0         0
vNic_4          0    0      0         0
vNic_5          0    0      0         0
```

```
L3 Errors
=========
IP:
 ReasmFails : 0
 InHdrErrors : 0
 InDiscards : 0
 FragFails : 0
 InAddrErrors : 0
 OutDiscards : 0
 OutNoRoutes : 0
 ReasmTimeout : 0
ICMP:
 InTimeExcds : 0
 InErrors : 227
 OutTimeExcds : 0
 OutDestUnreachs : 152
 OutParmProbs : 0
 InSrcQuenchs : 0
 InRedirects : 0
 OutSrcQuenchs : 0
 InDestUnreachs : 151
 OutErrors : 0
 InParmProbs : 0

Firewall Drop Counters
======================

Ipv4 Rules
==========
Chain - INPUT
rid pkts bytes target prot opt in out source    destination
0    119 30517 DROP   all  --   *   * 0.0.0.0/0 0.0.0.0/0    state INVALID
0      0     0 DROP   all  --   *   * 0.0.0.0/0 0.0.0.0/0
Chain - POSTROUTING
rid pkts bytes target prot opt in out source    destination
0    101 4040  DROP   all   --  *   * 0.0.0.0/0 0.0.0.0/0    state INVALID
0      0    0  DROP   all   --  *   * 0.0.0.0/0 0.0.0.0/0

Ipv6 Rules
==========
Chain - INPUT
rid pkts bytes target prot opt in out source destination
0      0     0   DROP  all      *   * ::/0   ::/0            state INVALID
0      0     0   DROP  all      *   * ::/0   ::/0
Chain - POSTROUTING
rid pkts bytes target prot opt in out source destination
0      0     0   DROP  all       *   * ::/0   ::/0           state INVALID
0      0     0   DROP  all       *   * ::/0   ::/0
```

Expected Behavior When Managing NSX Edge

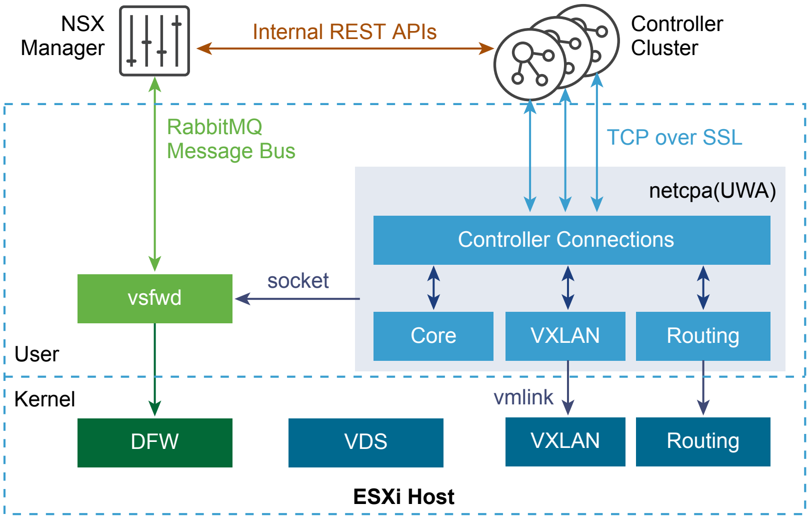
In vSphere Web Client, when you configure L2 VPN on an ESX Edge and add, remove, or modify **Site Configuration Details**, this action will cause all existing connections to be disconnected and reconnected. This behavior is expected.

Distributed Firewall

6

A RabbitMQ message bus is leveraged for communication between the vsfwd (RMQ client) and RMQ server process hosted on the NSX manager. The message bus is used by the NSX manager to send various information to the ESXi hosts, including policy rules that need to be programmed on the distributed firewall in the kernel.

Figure 6-1. ESXi Host User and Kernel Space Diagram



This chapter includes the following topics:

- How to Use the show dfw CLI
- Troubleshooting Distributed Firewall

How to Use the show dfw CLI

You can get most information about distributed firewalls on the NSX Manager central CLI.

The path to drill down to the desired information is as follows:

1 Show all clusters: `show cluster all`

2 Then show hosts in a specific cluster: `show cluster clusterID`

3 Then show all VMs on a host: `show host hostID`

4 Then show information for a VM, which includes filter names and vNIC IDs: `show vm vmID`

For example:

```
nsxmgr> show cluster all
No.  Cluster Name                Cluster Id               Datacenter Name     Firewall Status
1    Compute Cluster A           domain-c33               Datacenter Site A   Enabled
2    Management & Edge Cluster   domain-c41               Datacenter Site A   Enabled

nsxmgr> show cluster domain-c33
Datacenter: Datacenter Site A
Cluster: Compute Cluster A
No.  Host Name            Host Id                  Installation Status
1    esx-02a.corp.local   host-32                  Enabled
2    esx-01a.corp.local   host-28                  Enabled

nsxmgr> show host host-28
Datacenter: Datacenter Site A
Cluster: Compute Cluster A
Host: esx-01a.corp.local
No.  VM Name    VM Id     Power Status
1    web-02a    vm-219    on
2    web-01a    vm-216    on
3    win8-01a   vm-206    off
4    app-02a    vm-264    on

nsxmgr> show vm vm-264
Datacenter: Datacenter Site A
Cluster: Compute Cluster A
Host: esx-01a.corp.local
Host-ID: host-28
VM: app-02a
Virtual Nics List:
1.
Vnic Name      app-02a - Network adapter 1
Vnic Id        502ef2fa-62cf-d178-cb1b-c825fb300c84.000
Filters        nic-79396-eth0-vmware-sfw.2

nsxmgr> show dfw vnic 502ef2fa-62cf-d178-cb1b-c825fb300c84.000
Vnic Name      app-02a - Network adapter 1
Vnic Id        502ef2fa-62cf-d178-cb1b-c825fb300c84.000
Mac Address    00:50:56:ae:6c:6b
Port Group Id  dvportgroup-385
Filters        nic-79396-eth0-vmware-sfw.2

nsxmgr> show dfw host host-28 filter nic-79396-eth0-vmware-sfw.2 rules
ruleset domain-c33 {
  # Filter rules
  rule 1012 at 1 inout protocol any from addrset ip-securitygroup-10 to addrset ip-securitygroup-10
drop with log;
  rule 1013 at 2 inout protocol any from addrset src1013 to addrset src1013 drop;
  rule 1011 at 3 inout protocol tcp from any to addrset dst1011 port 443 accept;
  rule 1011 at 4 inout protocol icmp icmptype 8 from any to addrset dst1011 accept;
  rule 1010 at 5 inout protocol tcp from addrset ip-securitygroup-10 to addrset ip-securitygroup-11
port 8443 accept;
```

```
  rule 1010 at 6 inout protocol icmp icmptype 8 from addrset ip-securitygroup-10 to addrset ip-
securitygroup-11 accept;
  rule 1009 at 7 inout protocol tcp from addrset ip-securitygroup-11 to addrset ip-securitygroup-12
port 3306 accept;
  rule 1009 at 8 inout protocol icmp icmptype 8 from addrset ip-securitygroup-11 to addrset ip-
securitygroup-12 accept;
  rule 1003 at 9 inout protocol ipv6-icmp icmptype 136 from any to any accept;
  rule 1003 at 10 inout protocol ipv6-icmp icmptype 135 from any to any accept;
  rule 1002 at 11 inout protocol udp from any to any port 67 accept;
  rule 1002 at 12 inout protocol udp from any to any port 68 accept;
  rule 1001 at 13 inout protocol any from any to any accept;
}

ruleset domain-c33_L2 {
  # Filter rules
  rule 1004 at 1 inout ethertype any from any to any accept;
}
```

Troubleshooting Distributed Firewall

This topic provides information on understanding and troubleshooting VMware NSX 6.x Distributed Firewall (DFW).

Problem

Publishing Distributed Firewall rules fails.

Updating Distributed Firewall rules fails.

Cause

NSX Distributed Firewall is a hypervisor kernel-embedded firewall that provides visibility and control for virtualized workloads and networks. You can create access control policies based on VMware vCenter objects like datacenters and clusters, virtual machine names and tags, network constructs such as IP/VLAN/VXLAN addresses, as well as user group identity from Active Directory. Consistent access control policy is now enforced when a virtual machine gets vMotioned across physical hosts without the need to rewrite firewall rules. Since Distributed Firewall is hypervisor-embedded, it delivers close to line rate throughput to enable higher workload consolidation on physical servers. The distributed nature of the firewall provides a scale-out architecture that automatically extends firewall capacity when additional hosts are added to a datacenter.

The NSX Manager web application and NSX components on ESXi hosts communicate with each other through a RabbitMQ broker process that runs on the same virtual machine as the NSX Manager web application. The communication protocol that is used is AMQP (Advanced Message Queueing Protocol) and the channel is secured using SSL. On an ESXi host, the VSFWD (vShield Firewall Daemon) process establishes and maintains the SSL connection to the broker and sends and receives messages on behalf of other components, which talks to it through IPC.

Validate that each troubleshooting step below is true for your environment. Each step provides instructions or a link to a document to eliminate possible causes and take corrective action as necessary. The steps are ordered in the most appropriate sequence to isolate the issue and identify the proper resolution. After each step, re-attempt to update/publish the Distributed Firewall rules.

Solution

1 Verify that the prerequisites are met to run Distributed Firewall (DFW).

- VMware vCenter Server 5.5
- VMware ESXi 5.1 or ESXi 5.5
- VMware NSX 6.0 and later

2 Verify that the DFW VIBs are successfully installed on each of the ESXi hosts in the cluster. To do this, on each of the ESXi host that is on the cluster, run this command.

For example:

```
# esxcli software vib list | grep esx-vsip

esx-vsip                       5.5.0-0.0.2318233  VMware  VMwareCertified   2015-01-24

# esxcli software vib list | grep dvfilter

esx-dvfilter-switch-security   5.5.0-0.0.2318233  VMware  VMwareCertified   2015-01-24
```

3 On the ESXi hosts, verify the vShield-Stateful-Firewall service is in a running state.

For example:

```
# /etc/init.d/vShield-Stateful-Firewall status

vShield-Stateful-Firewall is running
```

4 Verify that the Message Bus is communicating properly with the NSX Manager.

The process is automatically launched by the watchdog script and restarts the process if it terminates for an unknown reason. Run this command on each of the ESXi hosts on the cluster.

For example:

```
# ps | grep vsfwd

107557 107557 vsfwd /usr/lib/vmware/vsfw/vsfwd
107574 107557 vsfwd /usr/lib/vmware/vsfw/vsfwd
107575 107557 vsfwd /usr/lib/vmware/vsfw/vsfwd
107576 107557 vsfwd /usr/lib/vmware/vsfw/vsfwd
107577 107557 vsfwd /usr/lib/vmware/vsfw/vsfwd
107578 107557 vsfwd /usr/lib/vmware/vsfw/vsfwd
```

5 Verify that port 5671 is opened for communication in the firewall configuration.

This command shows the VSFWD connectivity to the RabbitMQ broker. Run this command on ESXi hosts to see a list of connections from the vsfwd process on the ESXi host to the NSX Manager. Ensure that the port 5671 is open for communication in any of the external firewall on the environment. Also, there should be at least two connections on port 5671. There can be more connections on port 5671 as there are NSX Edge virtual machines deployed on the ESXi host which also establish connections to the RMQ broker.

For example:

```
# esxcli network ip connection list |grep 5671

tcp         0       0  192.168.110.51:30133            192.168.110.15:5671   ESTABLISHED
10949155  newreno  vsfwd
tcp         0       0  192.168.110.51:39156            192.168.110.15:5671   ESTABLISHED
10949155  newreno  vsfwd
```

6 Verify that VSFWD is configured.

This command should display the NSX Manager IP address.

```
# esxcfg-advcfg -g /UserVars/RmqIpAddress
```

7 If you are using a host-profile for this ESXi host, verify that the RabbitMQ configuration is not set in the host profile.

See:

- https://kb.vmware.com/kb/2092871
- https://kb.vmware.com/kb/2125901

8 Verify if the RabbitMQ credentials of the ESXi host are out of sync with the NSX Manager. Download the NSX Manager Tech Support Logs. After gathering all the NSX Manager Tech Support logs, search all the logs for entries similar to:

Replace host-420 with the mo-id of the suspect host.

```
PLAIN login refused: user 'uw-host-420' - invalid credentials.
```

9 If such entries are found on the logs for the suspected ESXi host, resynchronize the message bus.

To resynchronize the message bus, use REST API. To better understand the issue, collect the logs immediately after the Message Bus is resynchronized.

```
HTTP Method : POST
Headers ,
Authorization : base64encoded value of username password
Accept : application/xml
Content-Type : application/xml
Request:
```

```
POST https://NSX_Manager_IP/api/2.0/nwfabric/configure?action=synchronize

Request Body:

<nwFabricFeatureConfig>
<featureId>com.vmware.vshield.vsm.messagingInfra</featureId>
<resourceConfig>
<resourceId<{HOST/CLUSTER MOID}</resourceId>
</resourceConfig>
</nwFabricFeatureConfig>
```

10 Use the `export host-tech-support <host-id> scp <uid@ip:/path>` command to gather host-specific firewall logs.

For example:

```
nsxmgr# export host-tech-support host-28 scp Administrator@192.168.110.10
Generating logs for Host: host-28...
```

11 Use the `show dfw host host-id summarize-dvfilter` command to verify that the firewall rules are deployed on a host and are applied to virtual machines.

In the output, `module: vsip` shows that the DFW module is loaded and running. The `name` shows the firewall that is running on each vNic.

You can get the host IDs by running the `show dfw cluster all` command to get the cluster domain IDs, followed by the `show dfw cluster domain-id` to get the host IDs.

For example:

```
# show dfw host host-28 summarize-dvfilter

Fastpaths:
agent: dvfilter-faulter, refCount: 1, rev: 0x1010000, apiRev: 0x1010000, module: dvfilter
agent: ESXi-Firewall, refCount: 5, rev: 0x1010000, apiRev: 0x1010000, module: esxfw
agent: dvfilter-generic-vmware, refCount: 1, rev: 0x1010000, apiRev: 0x1010000, module: dvfilter-
generic-fastpath
agent: dvfilter-generic-vmware-swsec, refCount: 4, rev: 0x1010000, apiRev: 0x1010000, module:
dvfilter-switch-security
agent: bridgelearningfilter, refCount: 1, rev: 0x1010000, apiRev: 0x1010000, module: vdrb
agent: dvfg-igmp, refCount: 1, rev: 0x1010000, apiRev: 0x1010000, module: dvfg-igmp
agent: vmware-sfw, refCount: 4, rev: 0x1010000, apiRev: 0x1010000, module: vsip

Slowpaths:

Filters:
world 342296 vmm0:2-vm_RHEL63_srv_64-shared-846-3f435476-8f54-4e5a-8d01-59654a4e9979 vcUuid:'3f 43
54 76 8f 54 4e 5a-8d 01 59 65 4a 4e 99 79'
 port 50331660 2-vm_RHEL63_srv_64-shared-846-3f435476-8f54-4e5a-8d01-59654a4e9979.eth1
  vNic slot 2
   name: nic-342296-eth1-vmware-sfw.2
   agentName: vmware-sfw
   state: IOChain Attached
   vmState: Detached
   failurePolicy: failClosed
   slowPathID: none
   filter source: Dynamic Filter Creation
  vNic slot 1
   name: nic-342296-eth1-dvfilter-generic-vmware-swsec.1
   agentName: dvfilter-generic-vmware-swsec
   state: IOChain Attached
   vmState: Detached
   failurePolicy: failClosed
   slowPathID: none
   filter source: Alternate Opaque Channel
 port 50331661 (disconnected)
  vNic slot 2
   name: nic-342296-eth2-vmware-sfw.2
   agentName: vmware-sfw        <================ DFW filter
   state: IOChain Detached
   vmState: Detached
   failurePolicy: failClosed
   slowPathID: none
   filter source: Dynamic Filter Creation
 port 33554441 2-vm_RHEL63_srv_64-shared-846-3f435476-8f54-4e5a-8d01-59654a4e9979
  vNic slot 2
   name: nic-342296-eth0-vmware-sfw.2
   agentName: vmware-sfw    <================= DFW filter
   state: IOChain Attached
   vmState: Detached
   failurePolicy: failClosed
   slowPathID: none
   filter source: Dynamic Filter Creation
```

12 Run the `show dfw host hostID filter filterID rules` command.

For example:

```
# show dfw host host-28 filter nic-79396-eth0-vmware-sfw.2 rules

ruleset domain-c33 {
  # Filter rules
  rule 1012 at 1 inout protocol any from addrset ip-securitygroup-10 to addrset ip-
securitygroup-10 drop with log;
  rule 1013 at 2 inout protocol any from addrset src1013 to addrset src1013 drop;
  rule 1011 at 3 inout protocol tcp from any to addrset dst1011 port 443 accept;
  rule 1011 at 4 inout protocol icmp icmptype 8 from any to addrset dst1011 accept;
  rule 1010 at 5 inout protocol tcp from addrset ip-securitygroup-10 to addrset ip-
securitygroup-11 port 8443 accept;
  rule 1010 at 6 inout protocol icmp icmptype 8 from addrset ip-securitygroup-10 to addrset ip-
securitygroup-11 accept;
  rule 1009 at 7 inout protocol tcp from addrset ip-securitygroup-11 to addrset ip-
securitygroup-12 port 3306 accept;
  rule 1009 at 8 inout protocol icmp icmptype 8 from addrset ip-securitygroup-11 to addrset ip-
securitygroup-12 accept;
  rule 1003 at 9 inout protocol ipv6-icmp icmptype 136 from any to any accept;
  rule 1003 at 10 inout protocol ipv6-icmp icmptype 135 from any to any accept;
  rule 1002 at 11 inout protocol udp from any to any port 67 accept;
  rule 1002 at 12 inout protocol udp from any to any port 68 accept;
  rule 1001 at 13 inout protocol any from any to any accept;
}

ruleset domain-c33_L2 {
  # Filter rules
  rule 1004 at 1 inout ethertype any from any to any accept;
```

13 Run the `show dfw host hostID filter filterID addrsets`command.

For example:

```
# show dfw host host-28 filter nic-342296-eth2-vmware-sfw.2 addrsets

addrset dst1011 {
ip 172.16.10.10,
ip 172.16.10.11,
ip 172.16.10.12,
ip fe80::250:56ff:feae:3e3d,
ip fe80::250:56ff:feae:f86b,
}
addrset ip-securitygroup-10 {
ip 172.16.10.11,
ip 172.16.10.12,
ip fe80::250:56ff:feae:3e3d,
ip fe80::250:56ff:feae:f86b,
}
addrset ip-securitygroup-11 {
ip 172.16.20.11,
ip fe80::250:56ff:feae:23b9,
}
addrset ip-securitygroup-12 {
ip 172.16.30.11,
ip fe80::250:56ff:feae:d42b,
}
addrset src1013 {
ip 172.16.10.12,
ip 172.17.10.11,
ip fe80::250:56ff:feae:cf88,
ip fe80::250:56ff:feae:f86b,
}
```

14 If you have validated each of the above troubleshooting steps and cannot publish firewall rules to the host virtual machines, execute a host-level force synchronization via the NSX Manager UI or via the following REST API call.

```
URL : [https:]https://<nsx-mgr-ip>/api/4.0/firewall/forceSync/<host-id>
HTTP Method : POST
Headers ,
Authorization : base64encoded value of username password
Accept : application/xml
Content-Type : application/xml
```

Notes:

- Ensure that VMware Tools is running on the virtual machines if firewall rules do not use IP addresses. For more information, see https://kb.vmware.com/kb/2084048.

  VMware NSX 6.2.0 introduced the option to discover the virtual machine IP address using DHCP snooping or ARP snooping. These new discovery mechanisms enable NSX to enforce IP address-based security rules on virtual machines that do not have VMware Tools installed. For more information, see the NSX 6.2.0 Release Notes.

  DFW is activated as soon as the host preparation process is completed. If a virtual machine needs no DFW service at all, it can be added in the exclusion list functionality (by default, NSX Manager, NSX Controllers and Edge Services Gateways are automatically excluded from DFW function). There is a possibility that the vCenter Server access gets blocked after creating a Deny All rule in DFW. For more information, see https://kb.vmware.com/kb/2079620.

- When troubleshooting VMware NSX 6.x Distributed Firewall (DFW) with VMware Technical Support, these are required:

  - Output of the command `show dfw host hostID summarize-dvfilter` on each of the ESXi host on the cluster.

  - Distributed Firewall Configuration from the **Networking and Security > Firewall > General** tab and click **Export Configuration**. This exports the Distributed Firewall configuration to an XML format.

  - NSX Manager logs. For more information, see https://kb.vmware.com/kb/2074678.

  - vCenter Server logs. For more information, see https://kb.vmware.com/kb/1011641.

7 Load Balancing

The NSX Edge load balancer enables network traffic to follow multiple paths to a specific destination. It distributes incoming service requests evenly among multiple servers in such a way that the load distribution is transparent to users. There are two types of load balancing services to configure in NSX, a one-armed mode, also known as a proxy mode, or the Inline mode, otherwise known as the transparent mode.

NSX load balancing features are as follows:

- Protocols: TCP, HTTP, HTTPS
- Algorithms: Weighted round robin, IP hash, URI, least connection
- SSL termination with AES-NI acceleration
- SSL bridging (cient-side SSL + server-side SSL)
- SSL certificates management
- X-header forwarding for client identification
- L4/L7 transparent mode
- Connection throttling
- Enable/disable individual servers (pool members) for maintenance
- Health check methods (TCP, HTTP, HTTPS)
- Enhanced health check monitor
- Persistence/sticky methods: SourceIP, MSRDP, COOKIE, SSLSESSIONID
- One-arm mode
- URL rewrite and redirection
- Application rules for iRule type traffic shaping or content switching
- HA session sticky support for L7 proxy load balancing
- IPv6 support
- Enhanced load balancer CLI for troubleshooting
- Available on all flavors of Edge

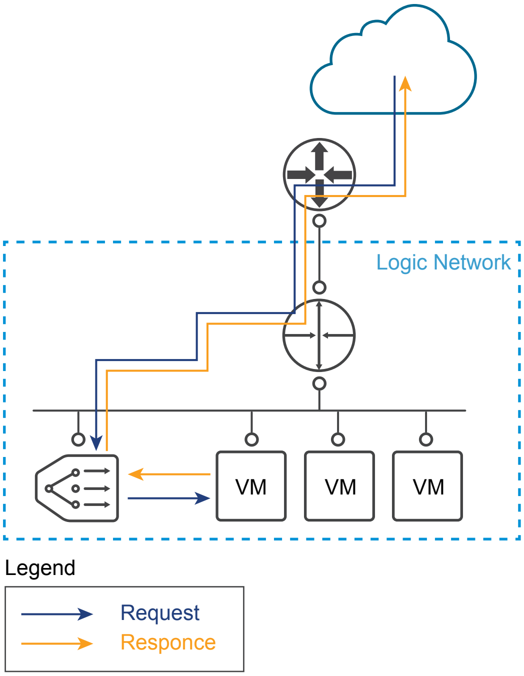
- Well-tuned X-large flavor for high performance SLB

This chapter includes the following topics:

- Scenario: Configure a One-Armed Load Balancer
- Load Balancer Troubleshooting Using the UI
- Load Balancer Troubleshooting Using the CLI
- Common Load Balancer Issues

Scenario: Configure a One-Armed Load Balancer

The Edge Services Gateway (ESG) can be thought of as a proxy for incoming client traffic.



In proxy mode, the load balancer uses its own IP address as the source address to send requests to a backend server. The backend server views all traffic as being sent from the load balancer and responds to the load balancer directly. This mode is also called SNAT mode or non-transparent mode.

A typical NSX one-armed load balancer is deployed on the same subnet with its backend servers, apart from the logical router. The NSX load balancer virtual server listens on a virtual IP for incoming requests from client and dispatches the requests to backend servers. For the return traffic, reverse NAT is required to change the source IP address from the backend server to a virtual IP (VIP) address and then send the virtual IP address to the client. Without this operation, the connection to the client would break.

After the ESG receives the traffic, it performs two operations: Destination Network Address Translation (DNAT) to change the VIP address to the IP address of one of the load balanced machines, and Source Network Address Translation (SNAT) to exchange the client IP address with the ESG IP address.

Then the ESG server sends the traffic to the load balanced server and the load balanced server sends the response back to the ESG then back to the client. This option is much easier to configure than the Inline mode, but has two potentials caveats. The first is that this mode requires a dedicated ESG server, and the second is that the load balancer servers are not aware of the original client IP address. One workaround for HTTP/HTTPS applications is to enable Insert X-Forwarded-For in the HTTP application profile so that the client IP address will be carried in the X-Forwarded-For HTTP header in the request sent to the backend server.

If client IP address visibility is required on the backend server for applications other than HTTP/HTTPS, you can configure the IP pool to be transparent. In case clients are not on the same subnet as the backend server, inline mode is recommended. Otherwise, you must use the load balancer IP address as the default gateway of the backend server.

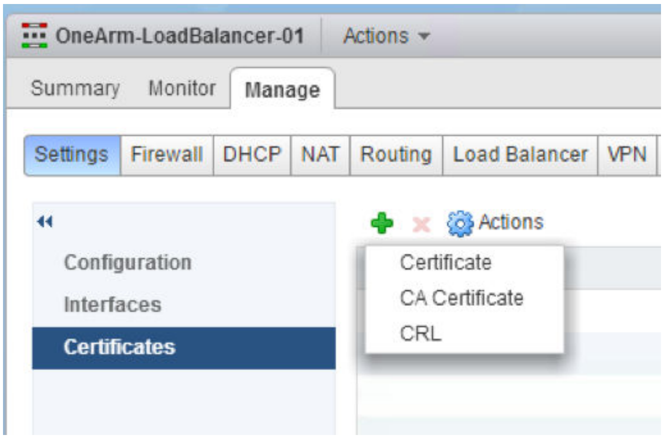
Note Usually, there are two methods to guarantee connection integrity:

- SNAT/proxy/non-transparent mode (discussed above)
- Direct server return (DSR)

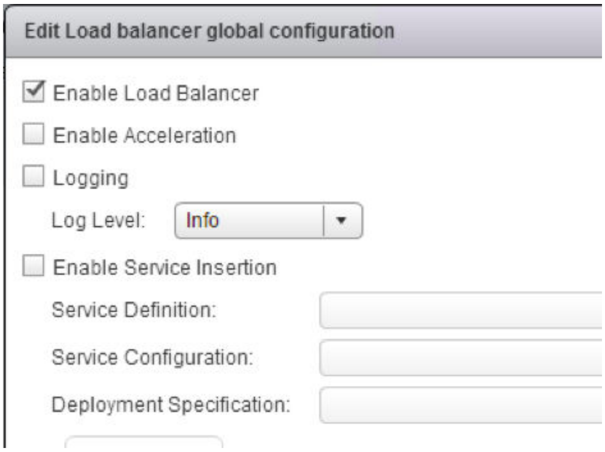
In DSR mode, the backend server responds directly to the client. Currently, NSX load balancer does not support DSR.

Procedure

1 Create a certificate by double-clicking the Edge and then selecting **Manage > Settings > Certificate**.



2 Enable the load balancer service by selecting **Manage > Load Balancer > Global Configuration > Edit**.



Edit Load balancer global configuration

☑ Enable Load Balancer

□ Enable Acceleration

□ Logging

Log Level: Info

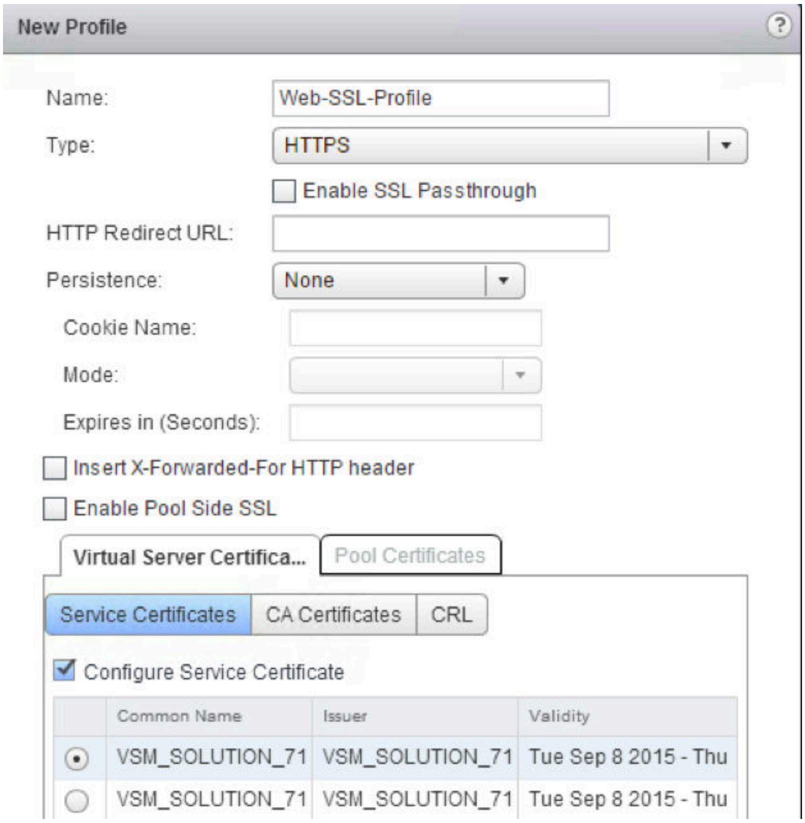
□ Enable Service Insertion

Service Definition:

Service Configuration:

Deployment Specification:

3 Create an HTTPS application profile by selecting **Manage > Load Balancer > Application Profiles**.



New Profile

Name: Web-SSL-Profile

Type: HTTPS

□ Enable SSL Passthrough

HTTP Redirect URL:

Persistence: None

Cookie Name:

Mode:

Expires in (Seconds):

□ Insert X-Forwarded-For HTTP header

□ Enable Pool Side SSL

☑ Configure Service Certificate

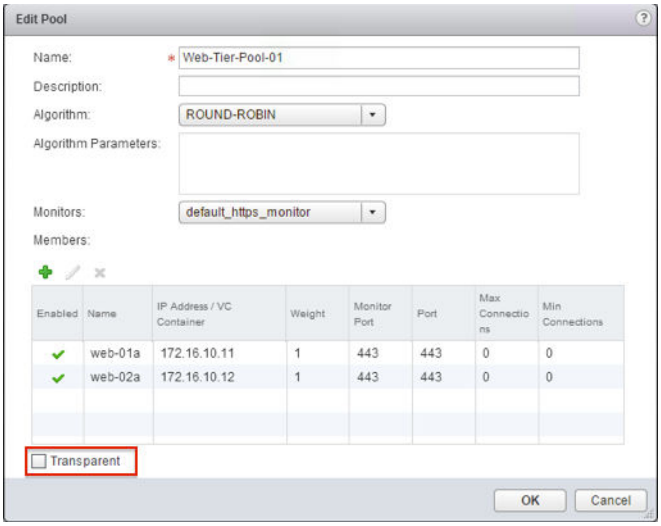
	Common Name	Issuer	Validity
●	VSM_SOLUTION_71	VSM_SOLUTION_71	Tue Sep 8 2015 - Thu
○	VSM_SOLUTION_71	VSM_SOLUTION_71	Tue Sep 8 2015 - Thu

Note The screenshot above uses self-signed certificates for documentation-purposes only.

4 Optionally, click **Manage > Load Balancer > Service Monitoring** and edit the default service monitoring to change it from basic HTTP/HTTPS to specific URL/URIs, as required.

5 Create server pools by selecting **Manage > Load Balancer > Pools**.

To use SNAT mode, leave the **Transparent** check box unchecked in the pool configuration.



Edit Pool

Name: * Web-Tier-Pool-01

Description:

Algorithm: ROUND-ROBIN

Algorithm Parameters:

Monitors: default_https_monitor

Members:

Enabled	Name	IP Address / VC Container	Weight	Monitor Port	Port	Max Connections	Min Connections
✓	web-01a	172.16.10.11	1	443	443	0	0
✓	web-02a	172.16.10.12	1	443	443	0	0

□ Transparent

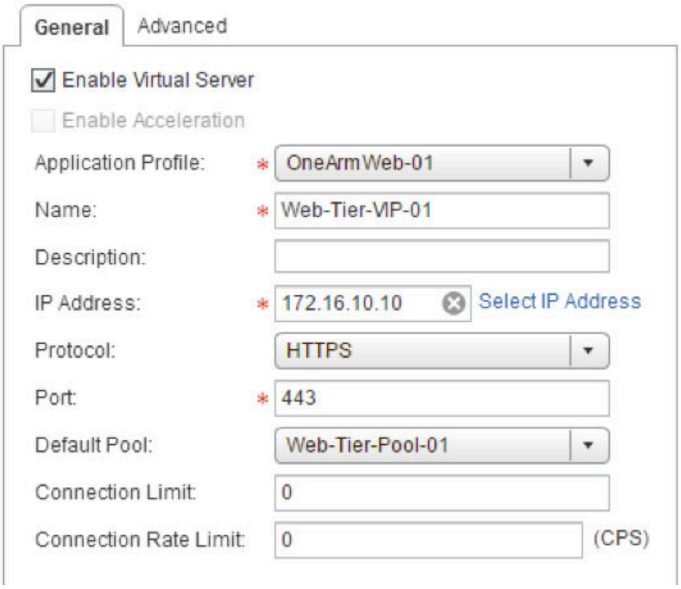
Ensure that the VMs are listed and enabled.

6 Optionally, click **Manage > Load Balancer > Pools > Show Pool Statistics** to check the status.

Make sure that the member status is UP.

7 Create a virtual server by selecting **Manage > Load Balancer > Virtual Servers**.

If you would like to use the L4 load balancer for UDP or higher-performance TCP, check **Enable Acceleration**. If you check **Enable Acceleration**, make sure that the firewall status is **Enabled** on the load balancer NSX Edge, because a firewall is required for L4 SNAT.



☑ Enable Virtual Server

□ Enable Acceleration

Application Profile: * OneArmWeb-01

Name: * Web-Tier-VIP-01

Description:

IP Address: * 172.16.10.10

Protocol: HTTPS

Port: * 443

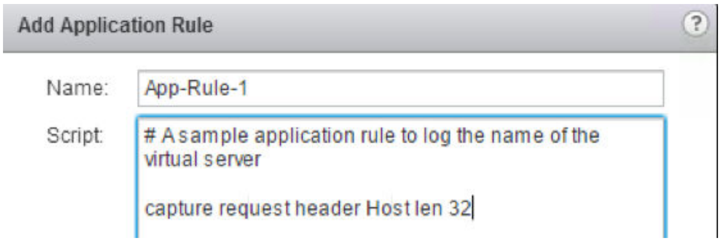
Default Pool: Web-Tier-Pool-01

Connection Limit: 0

Connection Rate Limit: 0 (CPS)

Ensure that the IP address is tied to the server pool.

8 Optionally, if using an application rule, check the configuration in **Manage > Load Balancer > Application Rules**.



Add Application Rule

Name: App-Rule-1

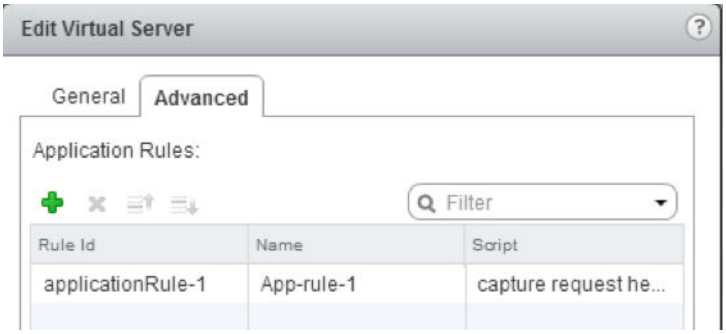
Script:

```
# A sample application rule to log the name of the
virtual server

capture request header Host len 32
```

9 If using an application rule, ensure that the application rule is associated with the virtual server in **Manage > Load Balancer > Virtual Servers > Advanced**.

For supported examples, see: https://communities.vmware.com/docs/DOC-31772.

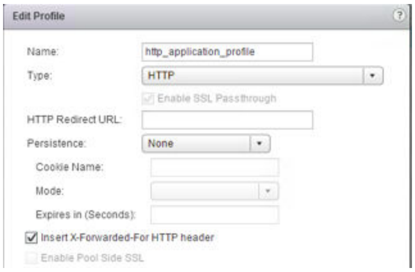


Edit Virtual Server

Application Rules:

Rule Id	Name	Script
applicationRule-1	App-rule-1	capture request he...

In non-transparent mode, the backend server cannot see the client IP, but can see the load balancer internal IP address. As a workaround for HTTP/HTTPS traffic, check **Insert X-Forwarded-For HTTP header**. With this option checked, the Edge load balancer adds the header "X-Forwarded-For" with the value of the client source IP address.



Edit Profile

Name: http_application_profile

Type: HTTP

☑ Enable SSL Passthrough

HTTP Redirect URL:

Persistence: None

Cookie Name:

Mode:

Expires in (Seconds):

☑ Insert X-Forwarded-For HTTP header

□ Enable Pool Side SSL

Load Balancer Troubleshooting Using the UI

You can use the UI to do some load balancer troubleshooting.

Problem

Troubleshooting is not working as expected.

Solution

1 Validate the configuration through the UI.

2 Check the pool member status through the UI.

3 Ensure that the default HTTP/HTTPS ports 80/443 are not used by other services (for example, SSL VPN).

4 Check the configuration for member ports and a monitor ports.

An incorrect configuration can cause a health-check failure.

5 If you are using a Layer 4 load balancer engine make sure:

a The traffic is using the TCP protocol.

b No persistence or Layer 7 settings are configured.

c **Enable Acceleration** is set to true in the load balancer global configuration.

6 If the pool is in transparent (inline) mode, make sure the Edge is in the return path. The Edge might be outside the return path if the default gateway of the virtual workload is pointing to an ESG other than the load-balancer ESG.

Load Balancer Troubleshooting Using the CLI

You can use the NSX CLI to do some load balancer troubleshooting.

Problem

Load balancing is not working as expected.

Solution

1 Show configuration and statistics information.

```
nsxedge> show configuration loadbalancer
nsxedge> show configuration loadbalancer virtual [virtual-server-name]
nsxedge> show configuration loadbalancer pool [pool-name]
nsxedge> show configuration loadbalancer monitor [monitor-name]
nsxedge> show configuration loadbalancer profile [profile-name]
nsxedge> show configuration loadbalancer rule [rule-name]
```

2 Check the load balancer engine status (L4/L7).

```
nsxedge> show service loadbalancer
haIndex:              0
-----------------------------------------------------------------------
Loadbalancer Services Status:

L7 Loadbalancer     : running
-----------------------------------------------------------------------
L7 Loadbalancer Statistics:
STATUS     PID        MAX_MEM_MB MAX_SOCK   MAX_CONN   MAX_PIPE   CUR_CONN   CONN_RATE
CONN_RATE_LIMIT MAX_CONN_RATE
running    1580       0          2081       1024       0          0          0
0               0
-----------------------------------------------------------------------
```

```
L4 Loadbalancer Statistics:
MAX_CONN   ACT_CONN   INACT_CONN TOTAL_CONN
0          0          0          0

Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
```

3 Check the load balancer pool status (L4/L7).

```
nsxedge> show service loadbalancer pool
-----------------------------------------------------------------------
Loadbalancer Pool Statistics:

POOL Web-Tier-Pool-01
|  LB METHOD round-robin
|  LB PROTOCOL L7
|  Transparent disabled
|  SESSION (cur, max, total) = (0, 0, 0)
|  BYTES in = (0), out = (0)
   +->POOL MEMBER: Web-Tier-Pool-01/web-01a, STATUS: UP
   |  |  HEALTH MONITOR = BUILT-IN, default_https_monitor:L7OK
   |  |  |  LAST STATE CHANGE: 2016-05-16 07:02:00
   |  |  SESSION (cur, max, total) = (0, 0, 0)
   |  |  BYTES in = (0), out = (0)
   +->POOL MEMBER: Web-Tier-Pool-01/web-02a, STATUS: UP
   |  |  HEALTH MONITOR = BUILT-IN, default_https_monitor:L7OK
   |  |  |  LAST STATE CHANGE: 2016-05-16 07:02:01
   |  |  SESSION (cur, max, total) = (0, 0, 0)
   |  |  BYTES in = (0), out = (0)
```

4 Check the load balancer object statistics (VIPs, pools, members).

Specify the name of the virtual server.

```
nsxedge> show service loadbalancer virtual Web-Tier-VIP-01

-----------------------------------------------------------------------
Loadbalancer VirtualServer Statistics:

VIRTUAL Web-Tier-VIP-01
|  ADDRESS [172.16.10.10]:443
|  SESSION (cur, max, total) = (0, 0, 0)
|  RATE (cur, max, limit) = (0, 0, 0)
|  BYTES in = (0), out = (0)
   +->POOL Web-Tier-Pool-01
   |  LB METHOD round-robin
   |  LB PROTOCOL L7
   |  Transparent disabled
   |  SESSION (cur, max, total) = (0, 0, 0)
   |  BYTES in = (0), out = (0)
      +->POOL MEMBER: Web-Tier-Pool-01/web-01a, STATUS: UP
      |  |  HEALTH MONITOR = BUILT-IN, default_https_monitor:L7OK
      |  |  |  LAST STATE CHANGE: 2016-05-16 07:02:00
      |  |  SESSION (cur, max, total) = (0, 0, 0)
      |  |  BYTES in = (0), out = (0)
      +->POOL MEMBER: Web-Tier-Pool-01/web-02a, STATUS: UP
      |  |  HEALTH MONITOR = BUILT-IN, default_https_monitor:L7OK
      |  |  |  LAST STATE CHANGE: 2016-05-16 07:02:01
      |  |  SESSION (cur, max, total) = (0, 0, 0)
      |  |  BYTES in = (0), out = (0)
```

```
nsxedge> show service loadbalancer pool Web-Tier-VIP-01
TIMESTAMP                SESSIONS       BYTESIN        BYTESOUT       SESSIONRATE    HTTPREQS
2016-04-27 19:56:40      00             00             00             00             00
2016-04-27 19:55:00      00             32             100            00             00
```

5 Check the service monitor status (OK, WARNING, CRITICAL)

```
nsxedge> show service loadbalancer monitor
-----------------------------------------------------------------------
Loadbalancer Health Check Statistics:

MONITOR PROVIDER   POOL               MEMBER     HEALTH STATUS
built-in           Web-Tier-Pool-01   web-01a    default_https_monitor:L7OK
built-in           Web-Tier-Pool-01   web-02a    default_https_monitor:L7OK
```

6 Check the log.

```
nsxedge> show log
2016-04-20T20:15:36+00:00 vShieldEdge kernel: Initializing cgroup subsys cpuset
2016-04-20T20:15:36+00:00 vShieldEdge kernel: Initializing cgroup subsys cpu
2016-04-20T20:15:36+00:00 vShieldEdge kernel: Initializing cgroup subsys cpuacct
...
```

7 Check the load balancer session table.

```
nsxedge> show service loadbalancer session
-----------------------------------------------------------------------
L7 Loadbalancer Statistics:
STATUS     PID        MAX_MEM_MB MAX_SOCK   MAX_CONN   MAX_PIPE   CUR_CONN   CONN_RATE
CONN_RATE_LIMIT MAX_CONN_RATE
running    1580       0          2081       1024       0          0          0
0               0

-----------------------------------------------------------------------L7 Loadbalancer Current
Sessions:

0x2192df1f300: proto=unix_stream src=unix:1 fe=GLOBAL be=<NONE> srv=<none> ts=09 age=0s calls=2
rq[f=c08200h,i=0,an=00h,rx=20s,wx=,ax=] rp[f=008000h,i=0,an=00h,rx=,wx=,ax=] s0=[7,8h,fd=1,ex=]
s1=[7,0h,fd=-1,ex=] exp=19s



-----------------------------------------------------------------------
L4 Loadbalancer Statistics:
```

```
MAX_CONN   ACT_CONN   INACT_CONN TOTAL_CONN
0          0          0          0

L4 Loadbalancer Current Sessions:

pro expire state       source       virtual    destination
```

8 Check the load balancer Layer 7 sticky-table status.

```
nsxedge> show service loadbalancer table
-----------------------------------------------------------------------
L7 Loadbalancer Sticky Table Status:

TABLE    TYPE    SIZE(BYTE)   USED(BYTE)

```

Common Load Balancer Issues

This topic discusses several issues and how to resolve them.

The following issues are common when using NSX load balancing:

- Load balancing on TCP port 443 does not work.
- A member of the load balancing pool is not utilized.
- Edge traffic is not load balanced.
- Layer 7 load balancing engine is stopped.
- Health monitor engine is stopped.
- Pool member monitor status is WARNING/CRITICAL.
- Pool member has the INACTIVE status.
- Layer 7 sticky table is not synchronized with the standby Edge.

Basic Troubleshooting

1 Check the load balancer configuration status in the vSphere Web Client:

  a Click **Networking & Security > NSX Edges**.

  b Double-click an NSX Edge.

  c Click **Manage**.

  d Click the **Load Balancer** tab.

  e Check the load balancer status and logging level configured.

2 Before troubleshooting the load balancer service, run the following command on the NSX Manager to ensure that the service is up an running:

```
nsxmgr> show edge edge-4 service loadbalancer
haIndex:              0
-----------------------------------------------------------------------
Loadbalancer Services Status:

L7 Loadbalancer     : running
-----------------------------------------------------------------------
L7 Loadbalancer Statistics:
STATUS     PID        MAX_MEM_MB MAX_SOCK   MAX_CONN   MAX_PIPE   CUR_CONN   CONN_RATE
CONN_RATE_LIMIT MAX_CONN_RATE
running    1580       0          2081       1024       0          0          0
0               0
-----------------------------------------------------------------------
L4 Loadbalancer Statistics:
MAX_CONN   ACT_CONN   INACT_CONN TOTAL_CONN
0          0          0          0

Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
```

Note You can run `show edge all` to look up the names of the NSX Edges.

Troubleshooting Configuration Issues

When the load balancer configuration operation is rejected by the NSX user interface or REST API call, this is classified as a configuration issue.

Troubleshooting Data Plane Issues

The load balancer configuration is accepted by NSX Manager, but there are connectivity or performance issues among the client-edge load-balance server. Data plane issues also include load balancer runtime CLI issues and load balancer system event issues.

1 Change the Edge logging level in NSX Manager from INFO to TRACE or DEBUG using this REST API call.

```
URL: https://NSX_Manager_IP/api/1.0/services/debug/loglevel/com.vmware.vshield.edge?level=TRACE
Method: POST
```

2 Check the pool member status in the vSphere Web Client.

a Click **Networking & Security > NSX Edges**.

b Double-click an NSX Edge.

c Click **Manage**.

d Click the **Load Balancer** tab.

e Click **Pools** to see a summary of the configured load balancer pools.

f Select your load balancer pool. click **Show Pool Statistics**, and verify that the pool state is UP.

3 You can get more detailed load balancer pool configuration statistics from the NSX Manager using this REST API call:

```
URL: https://NSX_Manager_IP/api/4.0/edges/{edgeId}/loadbalancer/statistics
Method: GET

<?xml version="1.0" encoding="UTF-8"?>
<loadBalancerStatusAndStats>
    <timeStamp>1463507779</timeStamp>
    <pool>
        <poolId>pool-1</poolId>
        <name>Web-Tier-Pool-01</name>
        <member>
            <memberId>member-1</memberId>
            <name>web-01a</name>
            <ipAddress>172.16.10.11</ipAddress>
            <status>UP</status>
            <lastStateChangeTime>2016-05-16 07:02:00</lastStateChangeTime>
            <bytesIn>0</bytesIn>
            <bytesOut>0</bytesOut>
            <curSessions>0</curSessions>
            <httpReqTotal>0</httpReqTotal>
            <httpReqRate>0</httpReqRate>
            <httpReqRateMax>0</httpReqRateMax>
            <maxSessions>0</maxSessions>
            <rate>0</rate>
            <rateLimit>0</rateLimit>
            <rateMax>0</rateMax>
            <totalSessions>0</totalSessions>
        </member>
        <member>
            <memberId>member-2</memberId>
            <name>web-02a</name>
            <ipAddress>172.16.10.12</ipAddress>
            <status>UP</status>
            <lastStateChangeTime>2016-05-16 07:02:01</lastStateChangeTime>
            <bytesIn>0</bytesIn>
            <bytesOut>0</bytesOut>
            <curSessions>0</curSessions>
            <httpReqTotal>0</httpReqTotal>
            <httpReqRate>0</httpReqRate>
            <httpReqRateMax>0</httpReqRateMax>
            <maxSessions>0</maxSessions>
            <rate>0</rate>
            <rateLimit>0</rateLimit>
            <rateMax>0</rateMax>
            <totalSessions>0</totalSessions>
        </member>
        <status>UP</status>
        <bytesIn>0</bytesIn>
        <bytesOut>0</bytesOut>
        <curSessions>0</curSessions>
        <httpReqTotal>0</httpReqTotal>
        <httpReqRate>0</httpReqRate>
        <httpReqRateMax>0</httpReqRateMax>
        <maxSessions>0</maxSessions>
        <rate>0</rate>
        <rateLimit>0</rateLimit>
        <rateMax>0</rateMax>
        <totalSessions>0</totalSessions>
    </pool>
    <virtualServer>
        <virtualServerId>virtualServer-1</virtualServerId>
        <name>Web-Tier-VIP-01</name>
        <ipAddress>172.16.10.10</ipAddress>
        <status>OPEN</status>
        <bytesIn>0</bytesIn>
        <bytesOut>0</bytesOut>
        <curSessions>0</curSessions>
        <httpReqTotal>0</httpReqTotal>
        <httpReqRate>0</httpReqRate>
        <httpReqRateMax>0</httpReqRateMax>
        <maxSessions>0</maxSessions>
        <rate>0</rate>
        <rateLimit>0</rateLimit>
        <rateMax>0</rateMax>
        <totalSessions>0</totalSessions>
    </virtualServer>
</loadBalancerStatusAndStats>
```

4 To check load balancer statistics from the command line, run these commands on the NSX Edge.

   For a particular virtual machine: First run `show service loadbalancer virtual` to get the virtual machine name. Then run `show statistics loadbalancer virtual <virtual-machine-name>`.

   For a particular TCP pool: First run `show service loadbalancer pool` to get the pool name. Then run `show statistics loadbalancer pool <pool-name>`.

5 Review the load balancer statistics for signs of failure.