



## eth Interfaces Troubleshooting

- [IP Address Conflict in AFW Address and System Access, on page 1](#)
- [Recovering Deleted eth Interface, on page 2](#)
- [Modifying Network Interfaces \(eth0 and eth1\) Post DCNM Installation, on page 3](#)
- [Configuring Enhanced-Fabric-Inband Interface \(eth2\) Post DCNM Installation, on page 12](#)
- [DHCP Relay Not Operational, on page 13](#)

### IP Address Conflict in AFW Address and System Access

**Problem** You cannot access the Cisco DCNM Web UI when the user system is configured in the same IP subnet as that of the internal subnet used by application framework in the Cisco DCNM.

**Possible Cause** Application framework IP address subnet that is configured on DCNM is conflicting with the IP address that is configured on a system that is accessed by the Cisco DCNM user.

#### Solution

If the DCNM internal address space is conflicting with the address space that is used in the network where a user access DCNM, use the application framework configuration to modify the subnet used in DCNM.

From Cisco DCNM Release 11.0, DCNM Infrastructure uses specific subnets, by default, for its internal purpose. The IP address subnets are as follows:

- 10.1.0.0/16: Used by service containers to communicate between each other.
- 172.17.0.0/16: Used by containers to communicate with native services.
- 172.18.0.0/16: Used by containers to communicate with any other native services.

The above subnets are not used to communicate with any devices outside of the DCNM. But, they can conflict with some services if they are used by external devices. For example, your PC used to access DCNM on the browser may use one of the same subnets or failure to enable EPL if the fabric routing loopback is using the same subnet pool to pick loopback IPs.

While installing Cisco DCNM Release 11.2(1), you can configure all the above subnets from a single larger subnet. When upgrading from Release 11.0(1) or 11.1(1) to Release 11.2(1), you must reconfigure these subnets, as required.

Modify the subnets by using the following commands:

1. `appmgr afw setup-net<ipv4-subnet>`

IPv4 subnet must have minimum length of /24 and maximum length of /20.



**Note** This command is not supported on DCNM installations that have computes connected.

This command reconfigures inter-subnet address that is used by service containers to communicate between each other. Execute this command on the Active node first, and then on the Standby node.

## 2. `apmgr afw setup-bridge<ipv4-subnet>`

IPv4 subnet must have minimum length of /24 and maximum length of /20.

This command reconfigures the subnets that are used by service containers to communicate with any other native services in the DCNM. Execute this command on all the DCNM nodes, including the Compute install nodes.

To confirm the change to subnets used for communication to docker native services, use the following sample commands outputs.

```
root@dcnm# ifconfig docker0
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 0.0.0.0
    inet6 fe80::42:3aff:feal:dd09 prefixlen 64 scopeid 0x20<link>
    ether 02:42:3a:a1:dd:09 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 656 (656.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@dcnm# ifconfig docker_gwbridge
docker_gwbridge: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.18.0.1 netmask 255.255.0.0 broadcast 0.0.0.0
    inet6 fe80::42:b0ff:fe9a:5adc prefixlen 64 scopeid 0x20<link>
    ether 02:42:b0:9a:5a:dc txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# Recovering Deleted eth Interface

## Release Impacted

Cisco DCNM Release 10.4(2) only

**Problem** In DCNM Release 10.4(2), after you reboot DCNM, the existing fabrics are not visible on the Cisco DCNM **Web UI > Configure > LAN Fabric**. In the **Select a Fabric** drop-down list on Network Deployment page, the fabrics are displayed.

**Possible Cause** If you have accidentally deleted the eth1 interface on the VM, the fabrics may not appear on the Cisco DCNM **Web UI > Configure > LAN Fabric**.

## Solution

To recover the deleted eth1 interface on the DCNM VM, perform the following steps:

1. Logon to the Cisco DCNM appliance using SSH.
2. Navigate to the directory `/etc/udev/rules.d/` and save the file `70-persistent-net.rules` to your local directory.
3. Open the file and make a note of the MAC address for the eth1 interface.
4. On SSH client, navigate to the **more** directory using `cd more` command.  
Execute the following command: `/etc/sysconfig/network-scripts/ifcfg-*`  
Save the output to your local directory.
5. Select **Power > Power On** and shut down the VM.
6. Click **Edit > Virtual Machine Details** to edit the virtual machine settings.
7. Click **Add > Ethernet Adapter**. From the **Adapter Type** drop-down list, choose **VMNET3**.
8. From the **Network Connection** drop-down list, select **DCNM Fabric Management Network**.
9. Check the **Connect at power on** check box. Click **Next** and then click **Finish**.
10. Select the newly added NIC and choose **Manual** for MAC assignment. The first 3 bytes are auto-populated. Take the last 3 bytes from the MAC address assigned to the previous eth1 interface and enter the value.
11. Click **OK**. You must wait for the VM to reconfigure.
12. Power on the VM and wait until DCNM is operational.
13. On the SSH client, verify if the eth1 interface is configured with the originally assigned IP address, using the `ifconfig -a` command. Ensure that the status of the eth1 interface is UP.
14. On the DCNM SSH, ping the eth1 gateway or the management IP of the attached switches.
15. On the Cisco DCNM Web UI, choose **Configure > LAN Fabric**. Verify if the fabric is visible.
16. Choose **Configure > Network Deployment** and verify if the fabric is visible in fabric selection drop-down list.

## Modifying Network Interfaces (eth0 and eth1) Post DCNM Installation

Along with the eth0 and eth1 IP address (IPv4 and/or IPv6), you can also modify the DNS and the NTP server configuration using the `appmgr update network-properties` command.

For step-by-step instructions on how to modify the network parameters using the `appmgr update network-properties` commands, see the following sections.

- [Modifying Network Properties on DCNM in Standalone Mode, on page 4](#)

[Sample Command Output for Modifying Network Parameters in the Cisco DCNM Standalone Setup, on page 4](#)

- [Modifying Network Properties on DCNM in Native HA Mode, on page 5](#)

[Sample Command Output for Modifying Network Parameters in the Cisco DCNM Native HA Setup, on page 7](#)

### Modifying Network Properties on DCNM in Standalone Mode

The following sample shows the output for the **appmgr update network-properties** command for a Cisco DCNM Standalone Appliance.



**Note** Execute the following commands on the DCNM Appliance console to avoid a premature session timeout.

1. Initiate a session on the console, using the following command:

```
appmgr update network-properties session start
```

2. Update the Network Properties using the following command:

```
appmgr update network-properties set ipv4 {eth0|eth1}<ipv4-address> <network-mask> <gateway>
```

Enter the new IPv4 address for the Management (eth0) interface, along with the subnet mask and gateway IP addresses.

3. View and verify the changes by using the following command:

```
appmgr update network-properties session show {config | changes | diffs}
```

4. After you validate the changes, apply the configuration using the following command:

```
appmgr update network-properties session apply
```

Wait for a few minutes before you can logon to the Cisco DCNM Web UI using the eth0 Management Network IP address.

### Sample Command Output for Modifying Network Parameters in the Cisco DCNM Standalone Setup

The following sample example shows how to modify the network parameters post installation for a Cisco DCNM Standalone setup.

```
dcnm# appmgr update network-properties session start

dcnm# appmgr update network-properties set ipv4 eth0 172.28.10.244 255.255.255.0 172.28.10.1
dcnm# appmgr update network-properties set ipv4 eth1 100.0.0.244 255.0.0.0
*****
WARNING: fabric/poap configuration may need to be changed
manually after changes are applied.
*****

dcnm# appmgr update network-properties session show changes
eth0 IPv4 addr 172.28.10.246/255.255.255.0 -> 172.28.10.244/255.255.255.0
eth1 IPv4 addr 1.0.0.246/255.0.0.0 -> 100.0.0.244/255.0.0.0

dcnm# appmgr update network-properties session apply
*****
WARNING
Applications of both nodes of the DCNM HA system need to be stopped
for the changes to be applied properly.
```

```

PLEASE STOP ALL APPLICATIONS MANUALLY
*****

Have applications been stopped? [y/n]: y
Applying changes
DELETE 1
Node left the swarm.
Server configuration file loaded: /usr/local/cisco/dcm/fm//conf/server.properties
log4j:WARN No appenders could be found for logger (fms.db).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
UPDATE 1
UPDATE 1
DELETE 1
server signaled
INFO      : [ipv6_wait_tentative] Waiting for interface eth0 IPv6 address(es) to leave the
'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth0 IPv6 address(es) to leave the
'tentative' state
*****
Please run 'appmgr start afw; appmgr start all' to restart your nodes.
*****

dcnm# appmgr start afw; appmgr start all
Started AFW Server Processes
Started AFW Agent Processes
Started AFW Server Processes
Started AFW Agent Processes
Started applications managed by heartbeat..
Check the status using 'appmgr status all'
Starting High-Availability services: INFO: Resource is stopped
Done.

Warning: PID file not written; -detached was passed.
AMQP User Check
Started AFW Server Processes
Started AFW Agent Processes
dcnm#

```

### Modifying Network Properties on DCNM in Native HA Mode

The following sample shows output to modify the network parameters using the **appmgr update network-properties** command for a Cisco DCNM Native HA Appliance.



- Note**
- Execute the following commands on the DCNM Active and Standby node console to avoid premature session timeout.
  - Ensure that you execute the commands in the same order as mentioned in the following steps.

1. Stop the DCNM Applications on the Standby node by using the following command:  
**appmgr stop all**  
Wait until all the applications stop on the Standby node before you go proceed.
2. Stop the DCNM Applications on the Active node by using the following command:  
**appmgr stop all**

3. Initiate a session on the Cisco DCNM console of both the Active and Standby nodes by using the following command:
 

**appmgr update network-properties session start**
  4. On the Active node, modify the network interface parameters by using the following commands:
    - a. Configure the IP address for eth0 and eth1 address by using the following command:
 

**appmgr update network-properties set ipv4 {eth0|eth1}<ipv4-address> <network-mask> <gateway>**

Enter the new IPv4 or IPv6 address for the eth1 interface, along with the subnet mask and gateway IP addresses.
    - b. Configure the VIP IP address by using the following command:
 

**appmgr update network-properties set ipv4 {vip0|vip1}<ipv4-address> <network-mask>**

Enter the vip0 address for eth0 interface. Enter the vip1 address for eth1 interface.
    - c. Configure the peer IP address by using the following command:
 

**appmgr update network-properties set ipv4 {peer0|peer1}<ipv4-address>**

Enter the eth0 address of the Standby node as peer0 address for Active node. Enter the eth1 address of the Standby node as peer1 address for Active node.
    - d. View and validate the changes that you have made to the network parameters by using the following command:
 


**appmgr update network-properties session show {config | changes | diffs}**

View the changes that you have configured by using the following command:
  5. On the Standby node, modify the network interface parameters using the commands described in [Step 4](#).
  6. After you validate the changes, apply the configuration on the Active node by using the following command:
 

**appmgr update network-properties session apply**

Wait until the prompt returns, to confirm that the network parameters are updated.
  7. After you validate the changes, apply the configuration on the Standby node by using the following command:
 

**appmgr update network-properties session apply**
  8. Start all the applications on the Active node by using the following command:
 

**appmgr start all**
-  **Note** Wait until all the applications are running successfully on the Active node, before proceeding to the next step.
9. Start all the applications on the Standby node by using the following command:
 

**appmgr start all**
  10. Establish peer trust key on the Active node by using the following command:

**appmgr update ssh-peer-trust**

11. Establish peer trust key on the Standby node by using the following command:

```
appmgr update ssh-peer-trust
```

**Sample Command Output for Modifying Network Parameters in the Cisco DCNM Native HA Setup**

The following sample example shows how to modify the network parameters post installation for a Cisco DCNM Native HA setup.



**Note** For example, let us indicate Active and Standby appliances as **dcnm1** and **dcnm2** respectively.

```
[root@dcnm2]# appmgr stop all
Stopping AFW Applications...
Stopping AFW Server Processes
Stopping AFW Agent Processes
Stopped Application Framework...
Stopping High-Availability services: Done.

Stopping and halting node rabbit@dcnm2 ...
Note: Forwarding request to 'systemctl enable rabbitmq-server.service'.
Stopping AFW Applications...
Stopping AFW Server Processes
Stopping AFW Agent Processes
Stopped Application Framework...
[root@dcnm2]#

[root@dcnm1]# appmgr stop all
Stopping AFW Applications...
Stopping AFW Server Processes
Stopping AFW Agent Processes
Stopped Application Framework...
Stopping High-Availability services: Done.

Stopping and halting node rabbit@dcnm1 ...
Note: Forwarding request to 'systemctl enable rabbitmq-server.service'.
Stopping AFW Applications...
Stopping AFW Server Processes
Stopping AFW Agent Processes
Stopped Application Framework...
[root@dcnm1]#

[root@dcnm1]# appmgr update network-properties session start
[root@dcnm2]# appmgr update network-properties session start

[root@dcnm1]# appmgr update network-properties set ipv4 eth0 172.28.10.244 255.255.255.0
172.28.10.1
[root@dcnm1]# appmgr update network-properties set ipv4 eth1 100.0.0.244 255.0.0.0
*****
WARNING: fabric/poap configuration may need to be changed
manually after changes are applied.
*****
[root@dcnm1]# appmgr update network-properties set ipv4 vip0 172.28.10.238 255.255.255.0
[root@dcnm1]# appmgr update network-properties set ipv4 vip1 100.0.0.238 255.0.0.0
[root@dcnm1]# appmgr update network-properties set ipv4 peer0 172.28.10.245
[root@dcnm1]# appmgr update network-properties set ipv4 peer1 100.0.0.245
[root@dcnm1]# appmgr update network-properties session show changes

[root@dcnm2]# appmgr update network-properties set ipv4 eth0 172.28.10.245 255.255.255.0
```

```

172.28.10.1
[root@dcnm2]# appmgr update network-properties set ipv4 eth1 100.0.0.245 255.0.0.0
*****
WARNING: fabric/poap configuration may need to be changed
manually after changes are applied.
*****
[root@dcnm2]# appmgr update network-properties set ipv4 vip0 172.28.10.238 255.255.255.0
[root@dcnm2]# appmgr update network-properties set ipv4 vip1 100.0.0.238 255.0.0.0
[root@dcnm2]# appmgr update network-properties set ipv4 peer0 172.28.10.244
[root@dcnm2]# appmgr update network-properties set ipv4 peer1 100.0.0.244
[root@dcnm2]# appmgr update network-properties session show changes

[root@dcnm1]# appmgr update network-properties session show changes
eth0 IPv4 addr 172.28.10.246/255.255.255.0 -> 172.28.10.244/255.255.255.0
eth1 IPv4 addr 1.0.0.246/255.0.0.0 -> 100.0.0.244/255.0.0.0
eth0 VIP 172.28.10.248/24 -> 172.28.10.238/24
eth1 VIP 1.0.0.248/8 -> 100.0.0.238/8
Peer eth0 IP 172.28.10.247 -> 172.28.10.245
Peer eth1 IP 1.0.0.245 -> 100.0.0.245

[root@dcnm1]# appmgr update network-properties session show config
===== Current configuration =====
NTP Server 1.ntp.esl.cisco.com
eth0 IPv4 addr 172.28.10.246/255.255.255.0
eth0 IPv4 GW 172.28.10.1
eth0 DNS 171.70.168.183
eth0 IPv6 addr 2001:420:284:2004:4:112:210:20/112
eth0 IPv6 GW 2001:420:284:2004:4:112:210:1
eth1 IPv4 addr 1.0.0.246/255.0.0.0
eth1 IPv4 GW
eth1 DNS 1.0.0.246
eth1 IPv6 addr
eth2 IPv4 addr /
eth2 IPv4 GW
Peer eth0 IP 172.28.10.247
Peer eth1 IP 1.0.0.247
Peer eth2 IP
eth0 VIP 172.28.10.248/24
eth1 VIP 1.0.0.248/8
eth2 VIP /
eth0 VIPv6 /
eth1 VIPv6 /

===== Session configuration =====
NTP Server 1.ntp.esl.cisco.com
eth0 IPv4 addr 172.28.10.244/255.255.255.0
eth0 IPv4 GW 172.28.10.1
eth0 DNS 171.70.168.183
eth0 IPv6 addr 2001:420:284:2004:4:112:210:20/112
eth0 IPv6 GW 2001:420:284:2004:4:112:210:1
eth1 IPv4 addr 100.0.0.244/255.0.0.0
eth1 IPv4 GW
eth1 DNS 1.0.0.246
eth1 IPv6 addr
eth2 IPv4 addr /
eth2 IPv4 GW
Peer eth0 IP 172.28.10.245
Peer eth1 IP 100.0.0.245
Peer eth2 IP
eth0 VIP 172.28.10.238/24
eth1 VIP 100.0.0.238/8
eth2 VIP /
eth0 VIPv6 /
eth1 VIPv6 /

```



```

[root@dcnm1]#

[root@dcnm2]# appmgr update network-properties session show config
===== Current configuration =====
NTP Server      1.ntp.esl.cisco.com
eth0 IPv4 addr  172.28.10.247/255.255.255.0
eth0 IPv4 GW    172.28.10.1
eth0 DNS        171.70.168.183
eth0 IPv6 addr
eth0 IPv6 GW
eth1 IPv4 addr  1.0.0.247/255.0.0.0
eth1 IPv4 GW
eth1 DNS        1.0.0.247
eth1 IPv6 addr
eth2 IPv4 addr  /
eth2 IPv4 GW
Peer eth0 IP    172.28.10.246
Peer eth1 IP    1.0.0.246
Peer eth2 IP
eth0 VIP        172.28.10.248/24
eth1 VIP        1.0.0.248/8
eth2 VIP        /
eth0 VIPv6      /
eth1 VIPv6      /

===== Session configuration =====
NTP Server      1.ntp.esl.cisco.com
eth0 IPv4 addr  172.28.10.245/255.255.255.0
eth0 IPv4 GW    172.28.10.1
eth0 DNS        171.70.168.183
eth0 IPv6 addr
eth0 IPv6 GW
eth1 IPv4 addr  100.0.0.245/255.0.0.0
eth1 IPv4 GW
eth1 DNS        1.0.0.247
eth1 IPv6 addr
eth2 IPv4 addr  /
eth2 IPv4 GW
Peer eth0 IP    172.28.10.244
Peer eth1 IP    100.0.0.244
Peer eth2 IP
eth0 VIP        172.28.10.238/24
eth1 VIP        100.0.0.238/8
eth2 VIP        /
eth0 VIPv6      /
eth1 VIPv6      /

[root@dcnm2]#

[root@dcnm1]# appmgr update network-properties session apply
*****
WARNING

Applications of both nodes of the DCNM HA system need to be stopped
for the changes to be applied properly.

PLEASE STOP ALL APPLICATIONS MANUALLY
*****

Have applications been stopped? [y/n]: y
Applying changes
DELETE 1
Node left the swarm.
Server configuration file loaded: /usr/local/cisco/dcm/fm//conf/server.properties
log4j:WARN No appenders could be found for logger (fms.db).

```

```

log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
UPDATE 1
UPDATE 1
DELETE 1
server signaled
INFO      : [ipv6_wait_tentative] Waiting for interface eth0 IPv6 address(es) to leave the
'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth0 IPv6 address(es) to leave the
'tentative' state
*****
Please run 'appmgr start afw; appmgr start all' to restart your nodes.
*****
Please run 'appmgr update ssh-peer-trust' on the peer node.
*****
[root@dcnm1]#

[root@dcnm2]# appmgr update network-properties session apply
*****
WARNING

Applications of both nodes of the DCNM HA system need to be stopped
for the changes to be applied properly.

PLEASE STOP ALL APPLICATIONS MANUALLY
*****

Have applications been stopped? [y/n]: y
Applying changes
DELETE 1
Node left the swarm.
Server configuration file loaded: /usr/local/cisco/dcm/fm//conf/server.properties
log4j:WARN No appenders could be found for logger (fms.db).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
UPDATE 1
UPDATE 1
DELETE 1
afwnetplugin:0.1
server signaled
*****
Please run 'appmgr start afw; appmgr start all' to restart your nodes.
*****
Please run 'appmgr update ssh-peer-trust' on the peer node.
*****
[root@dcnm2]#

[root@dcnm1]# appmgr start afw; appmgr start all
Started AFW Server Processes
Started AFW Agent Processes
Started AFW Server Processes
Started AFW Agent Processes
Started applications managed by heartbeat..
Check the status using 'appmgr status all'
Starting High-Availability services: INFO: Resource is stopped
Done.

Warning: PID file not written; -detached was passed.
AMQP User Check
Started AFW Server Processes
Started AFW Agent Processes
[root@dcnm1]#

```

**Wait until dcnm1 becomes active again.**

```
[root@dcnm2]# appmgr start afw; appmgr start all
Started AFW Server Processes
Started AFW Agent Processes
Started AFW Server Processes
Started AFW Agent Processes
Started applications managed by heartbeat..
Check the status using 'appmgr status all'
Starting High-Availability services: INFO: Resource is stopped
Done.
```

```
Warning: PID file not written; -detached was passed.
AMQP User Check
Started AFW Server Processes
Started AFW Agent Processes
[root@dcnm2]#
```

```
[root@dcnm1]# appmgr update ssh-peer-trust
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh -o 'StrictHostKeyChecking=no' '172.28.10.245'"
and check to make sure that only the key(s) you wanted were added.
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh -o 'StrictHostKeyChecking=no' '100.0.0.245'"
and check to make sure that only the key(s) you wanted were added.
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
Number of key(s) added: 1
Now try logging into the machine, with: "ssh -o 'StrictHostKeyChecking=no'
'dcnm-247.cisco.com'"
and check to make sure that only the key(s) you wanted were added.
[root@dcnm1]#
```

```
[root@dcnm2]# appmgr update ssh-peer-trust
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh -o 'StrictHostKeyChecking=no' '172.28.10.244'"
and check to make sure that only the key(s) you wanted were added.
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh -o 'StrictHostKeyChecking=no' '100.0.0.244'"
and check to make sure that only the key(s) you wanted were added.
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
Number of key(s) added: 1
```

```
Now try logging into the machine, with: "ssh -o 'StrictHostKeyChecking=no'
'dcnm-246.cisco.com'"
and check to make sure that only the key(s) you wanted were added.
[root@dcnm2]#
```

# Configuring Enhanced-Fabric-Inband Interface (eth2) Post DCNM Installation

During the DCNM installation, you can configure the In-Band Management interface. You must associate this network with the port group that corresponds to a fabric in-band connection. The In-Band Network provides reachability to the devices via the front-panel ports.



**Note** If you need to modify the already configured in-band network (eth2 interface), execute the **ifconfig eth2 0.0.0.0** command and run the **appmgr setup inband** command again.



**Note** You cannot use Endpoint Locator and Telemetry features if the eth2 interface is not configured.

To configure the eth2 interface for the in-band management network, use the **appmgr setup inband** command.

The following example shows a sample output for the **appmgr setup inband** command for a Cisco DCNM Standalone Appliance.

```
[root@dcnm]# appmgr setup inband
Configuring Interface for InBand Connectivity...
Please enter the information as prompted:
InBand Physical IP [e.g. 2.2.2.69]: 2.0.0.250
InBand Network Mask [e.g. 255.255.255.0]: 255.0.0.0
InBand Gateway [e.g. 2.2.2.1]: 2.0.0.1
Validating Inputs ...

You have entered these values..
PIP=2.0.0.250
NETMASK=255.0.0.0
GATEWAY=2.0.0.1

Press 'y' to continue configuration, 'n' to discontinue [y] y
{"ResponseType":0,"Response":"Refreshed"}
{"ResponseType":0,"Response":{"AfwServerEnabled":true,"AfwServerReady":true,"InbandSubnet":"2.0.0.0/8",
"InbandGateway":"2.0.0.1","OutbandSubnet":"0.0.0.0/8","OutbandGateway":"0.0.0.0","UnclusteredMode":true}}

Done.
[root@dcnm]#
```

The following example shows a sample output for the **appmgr setup inband** command for a Cisco DCNM Native HA Appliance.

On Cisco DCNM Primary appliance:

```
[root@dcnm-primary]# appmgr setup inband
Configuring Interface for InBand Connectivity...
Please enter the information as prompted:
InBand Physical IP [e.g. 2.2.2.69]: 2.0.0.244
InBand Network Mask [e.g. 255.255.255.0]: 255.0.0.0
InBand Gateway [e.g. 2.2.2.1]: 2.0.0.1
InBand Virtual IP for HA setup [e.g. 2.2.2.60]: 2.0.0.243
```

```
InBand Virtual Network Mask [mandatory for HA setup] [e.g. 255.255.255.0]: 255.0.0.0
Peer Inband IP [mandatory for HA setup] [e.g. 2.2.2.59]: 2.0.0.244
Validating Inputs ...
```

You have entered these values..

```
PIP=2.0.0.244
NETMASK=255.0.0.0
GATEWAY=2.0.0.1
VIP=2.0.0.243
VIP_NETMASK=255.0.0.0
PEER_ETH2=2.0.0.244
```

Press 'y' to continue configuration, 'n' to discontinue [y] **y**

Done.

```
[root@dcnm-primary]#
```

On Cisco DCNM Secondary appliance:

```
[root@dcnm-secondary]# appmgr setup inband
Configuring Interface for InBand Connectivity...
Please enter the information as prompted:
InBand Physical IP [e.g. 2.2.2.69]: 2.0.0.245
InBand Network Mask [e.g. 255.255.255.0]: 255.0.0.0
InBand Gateway [e.g. 2.2.2.1]: 2.0.0.1
InBand Virtual IP for HA setup [e.g. 2.2.2.60]: 2.0.0.243
InBand Virtual Network Mask [mandatory for HA setup] [e.g. 255.255.255.0]: 255.0.0.0
Peer Inband IP [mandatory for HA setup] [e.g. 2.2.2.59]: 2.0.0.244
Validating Inputs ...
```

You have entered these values..

```
PIP=2.0.0.245
NETMASK=255.0.0.0
GATEWAY=2.0.0.1
VIP=2.0.0.243
VIP_NETMASK=255.0.0.0
PEER_ETH2=2.0.0.244
```

Press 'y' to continue configuration, 'n' to discontinue [y] **y**

HA Role is Active {"ResponseType":0,"Response":"Refreshed"}

Done.

```
[root@dcnm-secondary]#
```

## DHCP Relay Not Operational

### Release impacted

Cisco DCNM Release 11.0(1) only

**Problem** After Cisco DCNM Installation, DHCP relay may not be operational.

**Possible Cause** Configuring eth1 interface is an optional parameter during Cisco DCNM Installation. If you do not configure eth1 gateway, DHCP relay may not be operational.

**Solution** Configure the eth1 gateway by using the following command:

```
dcnm# echo "2.0.0.0/8 via 1.0.0.100 dev eth1"
/etc/sysconfig/network-scripts/route-eth1
/etc/sysconfig/network-scripts/ifup-routes eth1
```

