# Network

The Network commands are not supported on connector AMI.

# connectorctl network config

To configure the network, use the **connectorctl network config** command.

When updating the network configuration, include the fields you want to change. If the existing network configuration is insufficient, you are prompted to enter necessary details to complete a correct configuration.

**Note** The system will reboot to apply all changes. Consequently, you lose connectivity and are be logged out. Ensure that you save any important work before proceeding with the configuration update.

**connectorctl network config** { **-p** *ip-network-stack* | **-i** *ip-address* | **-m** *netmask* | **-g** *gateway* | **-o** *domain* | **-n** *interface-name* | **-d** *dns-servers* }

## Syntax Description

| Keywords and Variables | Description |
|---|---|
| **-p** *ip-network-stack* | Network Stack. Use one of the following:<br>• ipv4<br>• ipv6 |
| **-i** *ipaddress[/prefix]* | IPv4 or IPv6 address formatted as: ip[/prefix]. |
| **-m** *netmask* | Netmask for IPv4 or prefix length IPv6 |
| **-g** *gateway* | Gateway address (IPv4 or IPv6) |
| **-o** *domain* | Search domain name |
| **-d** *dns-servers* | Comma-separated IP (IPv4 or IPv6) address list for multiple servers |
| **-n** *interface-name* | Interface name. Use one of the following:<br>• PRIMARY<br>• SECONDARY |

## Command History

| | |
|---|---|
| **Release 3** | This command is introduced. |

## Usage Guidelines

You can configure various settings for the network by specifying the right parameters. You can use this command to reconfigure the network entities in case of network change or network disruption.

**Caution** After you complete the network configuration, the system is automatically rebooted. After this, you lose connectivity and are logged out of the connector GUI.

The following example shows how to configure the PRIMARY interface on an IPv4 stack, with details such as IP address, stack, search domain name, and DNS:

```
[spacesadmin@connector ~]$ connectorctl network config -p ipv4 -i 10.22.244.7/24 -n PRIMARY
 -g 10.22.244.1 -o cisco.com -d 171.70.168.183
Executing command:network
Command execution status:Success
-----------------------
Network setup completed with given configuration.
System reboot will happen in 10 seconds. Do not execute any other command...
```

## Examples

The following example shows how to configure the SECONDARY interface on an IPv4 stack, with details such as IP address, stack, search domain name, and DNS:

```
[spacesadmin@connector ~]$ connectorctl network config  -p ipv4 -i 10.7.0.11/24 -g 10.7.0.1
 -o test.com -d 192.168.168.183 -n SECONDARY
Executing command:network
Command execution status:Success
-----------------------
Connection SECONDARY (5e970417-13b4-4ad8-af12-d125ce407c49) successfully added.
Network setup completed with given configuration.
Secondary interface - Added routes.
Secondary interface - Configured firewall zone.
System reboot will happen in 10 seconds. Do not execute any other command...
```

# connectorctl network show

To view the current network configuration and information about primary and secondary interfaces, use the **connectorctl network show** command. To view details of individual interface network, use the **-n** keyword.

**connectorctl network show** **-n** *interface-name*

### Command History

| | |
|---|---|
| **Release 3** | This command is introduced. |

### Examples

The following example shows how to display network configurations on an IPv4 stack.

```
[spacesadmin@connector ~]$ connectorctl network show
Executing command:network
Command execution status:Success
-----------------------
=================Network Config=================
BOOTPROTO=none
IPADDR=10.22.212.23
NETMASK=255.255.255.0
GATEWAY=10.22.212.1
DNS1=192.70.168.183
DOMAIN=test.com
HWADDR=00:50:56:90:7a:ff
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
=================end=================
```

```
[spacesadmin@connector ~]$ connectorctl network show
 Executing command:network
Command execution status:Success
-----------------------
=================Network Config=================
Hostname    - connector-p84-april1


Interface   - PRIMARY
---------------------------------

Network configuration for stack:ipv4
Ip Address  - 10.22.244.180/24
Mac Address - 00:0C:29:EE:24:8A
Gateway     - 10.22.244.1
Dns         - 192.168.168.183
Domain      - test.com


Interface   - SECONDARY
---------------------------------

Network configuration for stack:ipv4
Ip Address  - 7.7.0.11/24
Mac Address - 00:0C:29:EE:24:94
Gateway     - 7.7.0.1
Dns         - 192.168.168.183
Domain      - test.com

=================end=================
```

The following example shows how to display only the PRIMARY interface on an IPv4 stack.

```
[spacesadmin@connector ~]$ connectorctl network show -n PRIMARY
Executing command:network
Command execution status:Success
-----------------------
=================Network Config=================
Hostname    - connector-p84-april1


Interface   - PRIMARY
---------------------------------

Network configuration for stack:ipv4
Ip Address  - 10.22.244.180/24
Mac Address - 00:0C:29:EE:24:8A
Gateway     - 10.22.244.1
Dns         - 192.168.168.183
Domain      - test.com

=================end=================
```

The following example shows how to display only the SECONDARY interface on an IPv4 stack.

```
[spacesadmin@connector ~]$ connectorctl network show -n SECONDARY
Executing command:network
Command execution status:Success
-----------------------
=================Network Config=================
Hostname    - dualInt-HA-sec


Interface   - SECONDARY
---------------------------------

Network configuration for stack:ipv4
Ip Address  - 7.7.0.21/24
Mac Address - 00:0C:29:D6:E4:D7
Gateway     - 7.7.0.1
Dns         - 192.168.168.183
Domain      - test.com

=================end=================
```

**Note** The above example assumes the following:

- The PRIMARY interface of the connector is on the 10.22.x.x subnet, and is used to communicate with Cisco Spaces

  .

- The SECONDARY interface of the connector is on the 7.7.x.x subnet, and is used to communicate with all the devices, such as wireless controllers, switches, and APs.

## Examples

The following example shows how to display network configurations on an IPv6 stack.

```
[spacesadmin@connector ~]$ connectorctl network show
Executing command:network
Command execution status:Success
-----------------------
```

```
=================Network Config=================
Hostname    - conn3-dual-ipv6-p84


Interface   - PRIMARY
---------------------------------

Network configuration for stack:ipv6
Ip Address  - 2001:DB8:303:2022::60/64
Mac Address - 00:0C:29:70:9C:05
Gateway     - 2001:DB8:303:2022::1
Dns         - 2001:DB8:68d:4001::a
Domain      - test.com


Interface   - SECONDARY
---------------------------------

Network configuration for stack:ipv6
Ip Address  - 2001:DB8:303:2021::210/64
Mac Address - 00:0C:29:70:9C:OF
Gateway     - 2001:DB8:303:2021::1
Dns         - 2001:DB8:68d:4001::a
Domain      - test.com

=================end=================
```

The following example shows how to display the PRIMARY interface on an IPv6 stack.

```
[spacesadmin@connector ~]$ connectorctl network show -n PRIMARY
Executing command:network
Command execution status:Success
-----------------------
=================Network Config=================
Hostname    - conn3-dual-ipv6-p84


Interface   - PRIMARY
---------------------------------

Network configuration for stack:ipv6
Ip Address  - 2001:DB8:303:2022::60/64
Mac Address - 00:0C:29:70:9C:05
Gateway     - 2001:DB8:303:2022::1
Dns         - 2001:DB8:68d:4001::a
Domain      - test.com
```

**Note** The above example assumes the following:

- The PRIMARY interface of the connector is on the 2001:DB8:303:2022::0/64 subnet, and is used to communicate with Cisco Spaces.
- The SECONDARY interface of the connector is on the 2001:DB8:303:2021::0/64 subnet, and is used to communicate with all the devices, such as wireless controllers, switches, and APs.

# connectorctl network status

To view the detailed status of the network connectivity of the local machine to the gateway and DNS servers, use the **connectorctl network status** command. This status includes information about both the interfaces. To view the status of individual interface network, using the **--n** keyword.

**connectorctl network status -n** *interface-name*

## Command History

| | |
|---|---|
| **Release 3** | This command is introduced. |

## Examples

The following example shows how to display network connectivity of the local machine to the gateway and DNS servers on an IPv4 stack.

```
[spacesadmin@connector ~]$ connectorctl network status
Executing command:network
Command execution status:Success
-----------------------
Checking connection to 127.0.0.1
Connection check to 127.0.0.1: Success

Checking connection to ip address: 10.22.212.23
Connection check to ip address: Success

Checking connection to gateway: 10.22.212.1
Connection check to gateway: Success

Checking dns connection
Checking dns server resolution: 192.168.168.183
    Status check to dns server 192.168.168.183:
Success

Network status check successful.
```

The following example shows how to display network connectivity on an IPv4 stack configured with dual interface.

```
[spacesadmin@connector ~]$ connectorctl network status
Executing command:network
Command execution status:Success
-----------------------
=================Network Status=================

Network interface - PRIMARY
---------------------------------------

Checking connection status for network stack:ipv4
Checking connection to 127.0.0.1
Connection check to local:127.0.0.1 - Success
Checking connection to 10.22.244.180
Connection check to ip address:10.22.244.180 - Success
Checking connection to 10.22.244.1
Connection check to gateway:10.22.244.1 - Success
Checking dns server resolution: 192.168.168.183
    Status check to dns server 192.168.168.183 - Success
Network interface - PRIMARY status check successful.
Network interface - SECONDARY
---------------------------------------

Checking connection status for network stack:ipv4
Checking connection to 127.0.0.1
Connection check to local:127.0.0.1 - Success
Checking connection to 7.7.0.11
Connection check to ip address:7.7.0.11 - Success
Checking connection to 7.7.0.1
Connection check to gateway:7.7.0.1 - Success
Checking dns server resolution: 192.168.168.183
    Status check to dns server 192.168.168.183 - Success
Network interface - SECONDARY status check successful.
```

The following example shows how to display the network connectivity details of the PRIMARY interface.

```
[spacesadmin@connector ~]$ connectorctl network status -n PRIMARY
Executing command:network
Command execution status:Success
-----------------------
=================Network Status=================

Network interface - PRIMARY
---------------------------------------

Checking connection status for network stack:ipv4
Checking connection to 127.0.0.1
Connection check to local:127.0.0.1 - Success
Checking connection to 10.22.244.180
Connection check to ip address:10.22.244.180 - Success
Checking connection to 10.22.244.1
Connection check to gateway:10.22.244.1 - Success
Checking dns server resolution: 192.168.168.183
    Status check to dns server 192.168.168.183 - Success
Network interface - PRIMARY status check successful.
```

The following example shows how to display the network connectivity details of the SECONDARY interface.

```
[spacesadmin@connector ~]$ connectorctl network status -n SECONDARY
Executing command:network
Command execution status:Success
-----------------------
=================Network Status=================

Network interface - SECONDARY
---------------------------------------

Checking connection status for network stack:ipv4
Checking connection to 127.0.0.1
Connection check to local:127.0.0.1 - Success
Checking connection to 7.7.0.11
Connection check to ip address:7.7.0.11 - Success
Checking connection to 7.7.0.1
Connection check to gateway:7.7.0.1 - Success
Checking dns server resolution: 192.168.168.183
    Status check to dns server 192.168.168.183 - Success
Network interface - SECONDARY status check successful.
```

**Note** The above example assumes the following:

- The PRIMARY interface of the connector is on the 10.22.x.x subnet, and is used to communicate with Cisco Spaces

  .

- The SECONDARY interface of the connector is on the 7.7.x.x subnet, and is used to communicate with all the devices, such as wireless controllers, switches, and APs.

## Examples

The following example shows how to display status of network connectivity of the local machine to the gateway and DNS servers on an IPv6 stack configured with a dual interface.

```
[spacesadmin@connector ~]$ connectorctl network status
Executing command:network
Command execution status:Success
-----------------------
=================Network Status=================

Network interface - PRIMARY
---------------------------------------

Checking connection status for network stack:ipv6
Checking connection to ::1
Connection check to local:::1 - Success
Checking connection to 2001:DB8:303:2022::60
Connection check to ip address:2001:DB8:303:2022::60 - Success
Checking connection to 2001:DB8:303:2022::1
Connection check to gateway:2001:DB8:303:2022::1 - Success
Checking dns server resolution: 2001:DB8:68d:4001::a
Status check to dns server 2001:DB8:68d:4001::a - Success
Network interface - PRIMARY status check successful.

Network interface - SECONDARY
---------------------------------------

Checking connection status for network stack:ipv6
Checking connection to ::1
Connection check to local:::1 - Success
Checking connection to 2001:DB8:303:2021::210
Connection check to ip address:2001:DB8:303:2021::210 - Success
Checking connection to 2001:DB8:303:2021::1
Connection check to gateway:2001:DB8:303:2021::1 - Success
Checking dns server resolution: 2001:DB8:68d:4001::a
Status check to dns server 2001:DB8:68d:4001::a - Success
Network interface - SECONDARY status check successful.
```

The following example shows how to display the network connectivity details of the PRIMARY interface.

```
[spacesadmin@connector ~]$ connectorctl network status -n PRIMARY
Executing command:network
Command execution status:Success
-----------------------
=================Network Status=================

Network interface - PRIMARY
---------------------------------------
```

```
Checking connection status for network stack:ipv6
Checking connection to ::1
Connection check to local:::1 - Success
Checking connection to 2001:DB8:303:2022::60
Connection check to ip address:2001:DB8:303:2022::60 - Success
Checking connection to 2001:DB8:303:2022::1
Connection check to gateway:2001:DB8:303:2022::1 - Success
Checking dns server resolution: 2001:DB8:68d:4001::a
Status check to dns server 2001:DB8:68d:4001::a - Success
Network interface - PRIMARY status check successful.
```

The following example shows how to display the network connectivity details of the SECONDARY interface.

```
[spacesadmin@connector ~]$ connectorctl network status -n SECONDARY
Executing command:network
Command execution status:Success
-----------------------
=================Network Status=================

Network interface - SECONDARY
---------------------------------------

Checking connection status for network stack:ipv6
Checking connection to ::1
Connection check to local:::1 - Success
Checking connection to 2001:DB8:303:2021::210
Connection check to ip address:2001:DB8:303:2021::210 - Success
Checking connection to 2001:DB8:303:2021::1
Connection check to gateway:2001:DB8:303:2021::1 - Success
Checking dns server resolution: 2001:DB8:68d:4001::a
Status check to dns server 2001:DB8:68d:4001::a - Success
Network interface - SECONDARY status check successful.
```

**Note** The above example assumes the following:

- The PRIMARY interface of the connector is on the 2001:DB8:303:2022::0/64 subnet, and is used to communicate with Cisco Spaces.
- The SECONDARY interface of the connector is on the 2001:DB8:303:2021::0/64 subnet, and is used to communicate with all the devices, such as wireless controllers, switches, and APs.

# connectorctl network reset

To reset the network configuration of the secondary interface, use the **connectorctl network reset** command.

**connectorctl network reset**

### Command History

| | |
|---|---|
| **Release 3** | This command is introduced. |

### Examples

The following example shows how to reset the network configuration of the secondary interface.

```
[spacesadmin@connector ~]$ connectorctl network reset
Executing command:network
Command execution status:Success
-----------------------
Cleaning all unused connections
Connection 'SECONDARY' (f3f21bf5-f5c6-49cc-8cbd-70c582735466) successfully deleted.
Successfully reset interface:SECONDARY configuration
System reboot will happen in 10 seconds. Do not execute any other command...
```

# connectorctl network hostname

To edit the host name of this connector instance, use the **connectorctl network hostname** command.

**connectorctl network hostname -n** *hostname*

## Command History

| | |
|---|---|
| **Release 3** | This command is introduced. |

## Examples

The following is a sample output of the command:

```
[spacesadmin@connector ~]$ connectorctl network hostname -n connector3
Executing command:network
Command execution status:Success
-----------------------
Updated hostname:connector3
```

# connectorctl network ipv6

To manage IPv6 routing on a specified interface, use the **connectorctl network ipv6** command.

**connectorctl network ipv6 -i** *interface-name* { **show** | **enable** | **disable** }

## Syntax Description

| Keywords and Variables | Description |
|---|---|
| **show** | Shows the IPv6 routing status of a specified interface |
| **enable** | Enables IPv6 routing on a specified interface |
| **-i** *interface-name* | Interface Name |
| **disable** | Disables IPv6 routing on a specified interface |

## Command History

| | |
|---|---|
| **Release 3** | This command is introduced. |

## Usage Guidelines

**Note** Unless you have enabled IPv6 during the installation of Cisco Spaces: Connector, IPv6 is disabled by defualt.

## Examples

The following example shows how to view the IPv6 status of the specified interface (ens32).

```
[spacesadmin@connector ~]$ connectorctl network ipv6 -i ens32 -s
Executing command:network
Command execution status:Success
-----------------------
IPv6 Status:
----------------------------------------------------
net.ipv6.conf.ens32.disable_ipv6 = 0
```

The following example shows how to disable IPv6 on the specified interface (ens32).

```
[spacesadmin@connector ~]$ connectorctl network ipv6 -i ens32 -d
Executing command:network
Command execution status:Success
-----------------------
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/5)
IPv6 disabled on interface: ens32
----------------------------------------------------
net.ipv6.conf.ens32.disable_ipv6 = 1
```

The following example shows how to enable IPv6 on the specified interface (ens32).

```
[spacesadmin@connector ~]$ connectorctl network ipv6 -i ens32 -e
Executing command:network
Command execution status:Success
-----------------------
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/6)
```

```
IPv6 enabled on interface: ens32
----------------------------------------------------
net.ipv6.conf.ens32.disable_ipv6 = 0
```