



## Miscellaneous Commands

---

- [connectortl dockersubnet](#), on page 2
- [connectortl httpproxy-auth-deny-chars](#), on page 3
- [connectortl keyexg](#), on page 5
- [connectortl techsupport](#), on page 7
- [connectortl reset](#), on page 8
- [connectortl version](#), on page 9
- [connectortl help](#), on page 10

# connectorctl dockersubnet

To configure the IP address of the Docker daemon, use the **connectorctl dockersubnet** command.

**connectorctl dockersubnet show**

**connectorctl dockersubnet add -i ip-address -c cidr-length**

Syntax Description	Keywords and Variables	Description
	<b>show</b>	Shows the configuration details of the Docker daemon
	<b>add</b>	Changes the IP address of the Docker daemon
	<b>-i ip-address -c cidr-length</b>	Docker daemon IPv4 or IPv6 address and the Classless interdomain routing (CIDR) length
Command History	<b>Release 3</b>	This command is introduced.

## Usage Guidelines



**Caution** Ensure that the Docker subnet IP used does not overlap with the service-network subnet.

## Examples

The following example shows how to change the IP address of the Docker daemon.

```
[spacesadmin@connector ~]$ connectorctl dockersubnet add -i 172.20.10.10 -c 16
Executing command:dockersubnet
Command execution status:Success
-----
Successfully changed the docker subnet configuration to: 172.20.10.10/16
Note: This will restart the docker deamon, so all the connector services, service-manager
will be restarted
```

The following example shows how to view the IP address of the Docker daemon.

```
[spacesadmin@connector ~]$ connectorctl dockersubnet show
Executing command:dockersubnet
Command execution status:Success
-----
{
  "ipv6": true,
  "ip6tables": true,
  "experimental": true,
  "fixed-cidr-v6": "2001:db8:abc1::/64",
  "userland-proxy": false,
  "log-opts": {
    "max-size": "10m",
    "max-file": "5"
  },
  "bip": "172.20.10.10/16"
}
```

# connectorctl httpproxy-auth-deny-chars

To update the list of reserved characters supported in proxy passwords, use the **connectorctl httpproxy-auth-deny-chars** command. This command supports only the primary and secondary Ethernet interface.

```
connectorctl httpproxy-auth-deny-chars { show | add character | remove character | reset }
```

Syntax Description	Keywords and Variables	Description
	<b>show</b>	Shows the list of reserved characters supported in proxy passwords
	<b>add</b> <i>special-character</i>	Adds support for the specified reserved character in proxy passwords
	<b>remove</b> <i>special-character</i>	Removes support for the specified reserved character in proxy passwords
	<b>reset</b>	Resets the list of reserved characters supported in proxy passwords
<b>Command History</b>	<b>Release 3</b>	This command is introduced.

## Examples

The following example shows how to view the list of reserved characters supported in proxy passwords.

```
[spacesadmin@connector ~]$ connectorctl httpproxy-auth-deny-chars show
Executing command:httpproxy-auth-deny-chars
Command execution status:Success
-----
['*', '"', '(', ')', ';', ':', '&', '=', '+', ']', '[', ',', '/', '?', '%']
```

The following example shows how to add support for the specified reserved character '!' in proxy passwords.

```
[spacesadmin@connector ~]$ connectorctl httpproxy-auth-deny-chars add !
Executing command:httpproxy-auth-deny-chars
Command execution status:Success
-----
Updated reserved list with ['*', '"', '(', ')', ';', ':', '&', '=', '+', ']', '[', ',', '/', '?', '%', '!']
```

The following example shows how to remove support for the specified reserved character '!' in proxy passwords.

```
[spacesadmin@connector ~]$ connectorctl httpproxy-auth-deny-chars remove !
Executing command:httpproxy-auth-deny-chars
Command execution status:Success
-----
Remove of the reserved character is successful.
```

The following example shows how to reset the list of reserved characters supported in proxy passwords.

```
[spacesadmin@connector ~]$ connectorctl httpproxy-auth-deny-chars reset
Executing command:httpproxy-auth-deny-chars
Command execution status:Success
-----
Reset is successful.
```

# connectorctl keyexg

To manage the usage of weak key exchange algorithms by the SSH daemon (SSHD), use the **connectorctl keyexg** command.

```
connectorctl keyexg show
```

```
connectorctl keyexg remove { -a | -r algorithm-name }
```

```
connectorctl keyexg reset
```

Command History	Release 3	This command is introduced.
Syntax Description	Keywords and Variables	Description
	<b>show</b>	Shows the list of weak key exchange algorithms supported by the SSHD.
	<b>remove -a</b>	Removes support for all weak key exchange algorithms from SSHD.
	<b>remove -r algorithm-name</b>	Removes support for only the specified comma-separated algorithms from SSHD.
	<b>reset</b>	Resets the list of key exchange algorithms supported by SSHD.

## Examples

The following example shows how to view the list of key exchange algorithms supported by the SSHD.

```
[spacesadmin@connector ~]$ connectorctl keyexg show
Executing command:keyexg
Command execution status:Success
-----
List of supported Key exchange algorithms is:
kexalgorithms curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,
ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group16-sha512,
diffie-hellman-group18-sha512,diffie-hellman-group14-sha1,diffie-hellman-group14-sha256
```

The following example shows how to remove support for all weak key exchange algorithms from SSHD.

```
[spacesadmin@connector ~]$ connectorctl keyexg remove -a
Executing command:keyexg
Command execution status:Success
-----
Removing all unsupported weak algorithms
Successfully removed -diffie-hellman-group-exchange-sha1,diffie-hellman-group1-sha1 key
exchange algorithm(s)
```

The following example shows how to remove support for only the specified comma-separated algorithms from SSHD.

```
[spacesadmin@connector ~]$ connectorctl keyexg remove -r curve25519-sha256
Executing command:keyexg
Command execution status:Success
-----
Successfully removed -curve25519-sha256 key exchange algorithm(s)
```

The following example shows how to reset the list of key exchange algorithms supported by SSHD.

```
[spacesadmin@connector ~]$ connectorctl keyexg reset
Executing command:keyexg
Command execution status:Success
-----
Successfully reset key exchange algorithms configuration
```

# connectorctl techsupport

This command gathers and displays technical support information. The command creates a TAR file with information about the network, system, running docker containers, and downloaded images.

## connectorctl techsupport

### Syntax Description

This command has no keywords or arguments.

### Examples

```
[spacesadmin@connector ~]$ connectorctl techsupport
Executing command:techsupport
Command execution status:Success
-----
#####
DNA Spaces Connector 3.0 Tech Support Started At: Tue Aug  2 23:46:19 2022
#####

Interface Configuration
Ethernet Tool Stats
Ethernet Tool Ring Buffer Sizes
Network Interface Stats
Network Connection Stats
Route Configuration
NTP Stats
NTP Status
DNS Configuration
Domain Information Groper
ARP hosts
SAR Network
File System Usage
Partition Tables
Current Processes
Top Processes
Processor Related Stats
I/O Related Stats
Memory Stats
List Open Files Count
Up Time
SAR CPU
SAR CPU ALL
SAR I/O
SAR Paging and Memory Statistics
SAR Memory Utilization
Docker Downloaded Images
Docker Containers
Docker Service Status
Docker Stats
Service Manager Service Status
Service Agent Service Status
Docker journalctl Status
Connector Service Status
tech support saved to
/home/dnasadmin/techsupport/connector_tech_support_2022-08-02T23-46-19.gz
```

### Command History

#### Release 3

This command is introduced.

# connectorctl reset

This command resets all the connector configurations including the HTTP proxy and token.

## connectorctl reset

---

**Syntax Description**

This command has no keywords or arguments.

---

**Command History**

---

**Release 3**

This command is introduced.

---



# connectorctl version

This command displays the versions of **service-manager** and **service-agent** services on the connector.

## connectorctl version

---

### Syntax Description

This command has no keywords or arguments.

---

### Examples

The following example shows how to view the versions of service-manager and service-agent.

```
[spacesadmin@connector ~]$ connectorctl version
Executing command:version
Command execution status:Success
-----
Package:connector3-p82-sep2022
System Version:8.4.0.82
Service Agent Version:8.4.0.97
Service Manager Version:3.0.1.96
```

---

### Command History

#### Release 3

This command is introduced.

---

# connectorctl help

This command displays all the commands available on the connector command line interface.

## connectorctl help

---

**Syntax Description**

This command has no keywords or arguments.

---

**Command History**

---

**Release 3**

This command is introduced.

---