**Network Debug and Troubleshooting Guide**[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#network-debug-and-troubleshooting-guide)

This section discusses how to debug and troubleshoot network modules in Ansible.

* [How to troubleshoot](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#how-to-troubleshoot)
  + [Enabling Networking logging and how to read the logfile](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#enabling-networking-logging-and-how-to-read-the-logfile)
  + [Enabling Networking device interaction logging](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#enabling-networking-device-interaction-logging)
  + [Isolating an error](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#isolating-an-error)
* [Troubleshooting socket path issues](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#troubleshooting-socket-path-issues)
* [Category “Unable to open shell”](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#category-unable-to-open-shell)
  + [Error: “[Errno -2] Name or service not known”](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-errno-2-name-or-service-not-known)
  + [Error: “Authentication failed”](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-authentication-failed)
  + [Error: “connecting to host <hostname> returned an error” or “Bad address”](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-connecting-to-host-hostname-returned-an-error-or-bad-address)
  + [Error: “No authentication methods available”](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-no-authentication-methods-available)
  + [Clearing Out Persistent Connections](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#clearing-out-persistent-connections)
* [Timeout issues](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#timeout-issues)
  + [Persistent connection idle timeout](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#persistent-connection-idle-timeout)
  + [Command timeout](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#command-timeout)
  + [Persistent connection retry timeout](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#persistent-connection-retry-timeout)
  + [Timeout issue due to platform specific login menu with network\_cli connection type](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#timeout-issue-due-to-platform-specific-login-menu-with-network-cli-connection-type)
* [Playbook issues](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#playbook-issues)
  + [Error: “Unable to enter configuration mode”](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-unable-to-enter-configuration-mode)
* [Proxy Issues](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#proxy-issues)
  + [delegate\_to vs ProxyCommand](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#delegate-to-vs-proxycommand)
  + [Using bastion/jump host with netconf connection](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#using-bastion-jump-host-with-netconf-connection)
  + [Enabling jump host setting](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#enabling-jump-host-setting)
  + [Example ssh config file (~/.ssh/config)](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#example-ssh-config-file-ssh-config)
* [Miscellaneous Issues](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#miscellaneous-issues)
  + [Intermittent failure while using ansible.netcommon.network\_cli connection type](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#intermittent-failure-while-using-ansible-netcommon-network-cli-connection-type)
  + [Task failure due to mismatched error regex within command response using ansible.netcommon.network\_cli connection type](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#task-failure-due-to-mismatched-error-regex-within-command-response-using-ansible-netcommon-network-cli-connection-type)
  + [Intermittent failure while using ansible.netcommon.network\_cli connection type due to slower network or remote target host](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#intermittent-failure-while-using-ansible-netcommon-network-cli-connection-type-due-to-slower-network-or-remote-target-host)

[**How to troubleshoot**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id2)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#how-to-troubleshoot)

Ansible network automation errors generally fall into one of the following categories:

**Authentication issues:**

* Not correctly specifying credentials
* Remote device (network switch/router) not falling back to other other authentication methods
* SSH key issues

**Timeout issues:**

* Can occur when trying to pull a large amount of data
* May actually be masking a authentication issue

**Playbook issues:**

* Use of delegate\_to, instead of ProxyCommand. See [network proxy guide](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#network-delegate-to-vs-proxycommand) for more information.

**Warning**

unable to open shell

The unable to open shell message means that the ansible-connection daemon has not been able to successfully talk to the remote network device. This generally means that there is an authentication issue. See the “Authentication and connection issues” section in this document for more information.

[**Enabling Networking logging and how to read the logfile**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id3)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#enabling-networking-logging-and-how-to-read-the-logfile)

**Platforms:** Any

Ansible includes logging to help diagnose and troubleshoot issues regarding Ansible Networking modules.

Because logging is very verbose, it is disabled by default. It can be enabled with the [**ANSIBLE\_LOG\_PATH**](https://docs.ansible.com/ansible/latest/reference_appendices/config.html#envvar-ANSIBLE_LOG_PATH) and [**ANSIBLE\_DEBUG**](https://docs.ansible.com/ansible/latest/reference_appendices/config.html#envvar-ANSIBLE_DEBUG) options on the Ansible control node, that is the machine running ansible-playbook.

Before running ansible-playbook, run the following commands to enable logging:

*# Specify the location for the log file*

export **ANSIBLE\_LOG\_PATH=**~/ansible.log

*# Enable Debug*

export **ANSIBLE\_DEBUG=**True

*# Run with 4\*v for connection level verbosity*

ansible-playbook -vvvv ...

After Ansible has finished running you can inspect the log file which has been created on the Ansible control node:

less $ANSIBLE\_LOG\_PATH

2017-03-30 13:19:52,740 p=28990 u=fred | creating new control socket for host veos01:22 as user admin

2017-03-30 13:19:52,741 p=28990 u=fred | control socket path is /home/fred/.ansible/pc/ca5960d27a

2017-03-30 13:19:52,741 p=28990 u=fred | current working directory is /home/fred/ansible/test/integration

2017-03-30 13:19:52,741 p=28990 u=fred | using connection plugin network\_cli

**...**

2017-03-30 13:20:14,771 paramiko.transport userauth is OK

2017-03-30 13:20:15,283 paramiko.transport Authentication (keyboard-interactive) successful!

2017-03-30 13:20:15,302 p=28990 u=fred | ssh connection done, setting terminal

2017-03-30 13:20:15,321 p=28990 u=fred | ssh connection has completed successfully

2017-03-30 13:20:15,322 p=28990 u=fred | connection established to veos01 in 0:00:22.580626

From the log notice:

* p=28990 Is the PID (Process ID) of the ansible-connection process
* u=fred Is the user *running* ansible, not the remote-user you are attempting to connect as
* creating new control socket for host veos01:22 as user admin host:port as user
* control socket path is location on disk where the persistent connection socket is created
* using connection plugin network\_cli Informs you that persistent connection is being used
* connection established to veos01 in 0:00:22.580626 Time taken to obtain a shell on the remote device

**Note**

Port None creating new control socket for host veos01:None

If the log reports the port as None this means that the default port is being used. A future Ansible release will improve this message so that the port is always logged.

Because the log files are verbose, you can use grep to look for specific information. For example, once you have identified the pid from the creating new control socket for host line you can search for other connection log entries:

grep "p=28990" **$ANSIBLE\_LOG\_PATH**

[**Enabling Networking device interaction logging**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id4)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#enabling-networking-device-interaction-logging)

**Platforms:** Any

Ansible includes logging of device interaction in the log file to help diagnose and troubleshoot issues regarding Ansible Networking modules. The messages are logged in the file pointed to by the log\_path configuration option in the Ansible configuration file or by setting the [**ANSIBLE\_LOG\_PATH**](https://docs.ansible.com/ansible/latest/reference_appendices/config.html#envvar-ANSIBLE_LOG_PATH).

**Warning**

The device interaction messages consist of command executed on the target device and the returned response. Since this log data can contain sensitive information including passwords in plain text it is disabled by default. Additionally, in order to prevent accidental leakage of data, a warning will be shown on every task with this setting enabled, specifying which host has it enabled and where the data is being logged.

Be sure to fully understand the security implications of enabling this option. The device interaction logging can be enabled either globally by setting in configuration file or by setting environment or enabled on per task basis by passing a special variable to the task.

Before running ansible-playbook run the following commands to enable logging:

# Specify the location for the log file

export ANSIBLE\_LOG\_PATH=~/ansible.log

Enable device interaction logging for a given task

**-** **name:** get version information

**cisco.ios.ios\_command:**

**commands:**

**-** show version

**vars:**

**ansible\_persistent\_log\_messages:** True

To make this a global setting, add the following to your ansible.cfg file:

**[persistent\_connection]**

log\_messages **=** True

or enable the environment variable *ANSIBLE\_PERSISTENT\_LOG\_MESSAGES*:

# Enable device interaction logging

export ANSIBLE\_PERSISTENT\_LOG\_MESSAGES=True

If the task is failing on connection initialization itself, you should enable this option globally. If an individual task is failing intermittently this option can be enabled for that task itself to find the root cause.

After Ansible has finished running you can inspect the log file which has been created on the Ansible control node

**Note**

Be sure to fully understand the security implications of enabling this option as it can log sensitive information in log file thus creating security vulnerability.

[**Isolating an error**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id5)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#isolating-an-error)

**Platforms:** Any

As with any effort to troubleshoot it is important to simplify the test case as much as possible.

For Ansible this can be done by ensuring you are only running against one remote device:

* Using ansible-playbook --limit switch1.example.net...
* Using an ad hoc ansible command

*ad hoc* refers to running Ansible to perform some quick command using /usr/bin/ansible, rather than the orchestration language, which is /usr/bin/ansible-playbook. In this case we can ensure connectivity by attempting to execute a single command on the remote device:

ansible -m arista.eos.eos\_command -a 'commands=?' -i inventory switch1.example.net -e 'ansible\_connection=ansible.netcommon.network\_cli' -u admin -k

In the above example, we:

* connect to switch1.example.net specified in the inventory file inventory
* use the module arista.eos.eos\_command
* run the command ?
* connect using the username admin
* inform the ansible command to prompt for the SSH password by specifying -k

If you have SSH keys configured correctly, you don’t need to specify the -k parameter.

If the connection still fails you can combine it with the enable\_network\_logging parameter. For example:

# Specify the location for the log file

export ANSIBLE\_LOG\_PATH=~/ansible.log

# Enable Debug

export ANSIBLE\_DEBUG=True

# Run with ``-vvvv`` for connection level verbosity

ansible -m arista.eos.eos\_command -a 'commands=?' -i inventory switch1.example.net -e 'ansible\_connection=ansible.netcommon.network\_cli' -u admin -k

Then review the log file and find the relevant error message in the rest of this document.

[**Troubleshooting socket path issues**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id6)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#troubleshooting-socket-path-issues)

**Platforms:** Any

The Socket path does not exist or cannot be found and Unable to connect to socket messages indicate that the socket used to communicate with the remote network device is unavailable or does not exist.

For example:

fatal: [spine02]: FAILED! => {

"changed": false,

"failed": true,

"module\_stderr": "Traceback (most recent call last):\n File \"/tmp/ansible\_TSqk5J/ansible\_modlib.zip/ansible/module\_utils/connection.py\", line 115, in \_exec\_jsonrpc\nansible.module\_utils.connection.ConnectionError: Socket path XX does not exist or cannot be found. See Troubleshooting socket path issues in the Network Debug and Troubleshooting Guide\n",

"module\_stdout": "",

"msg": "MODULE FAILURE",

"rc": 1

}

or

fatal: [spine02]: FAILED! => {

"changed": false,

"failed": true,

"module\_stderr": "Traceback (most recent call last):\n File \"/tmp/ansible\_TSqk5J/ansible\_modlib.zip/ansible/module\_utils/connection.py\", line 123, in \_exec\_jsonrpc\nansible.module\_utils.connection.ConnectionError: Unable to connect to socket XX. See Troubleshooting socket path issues in Network Debug and Troubleshooting Guide\n",

"module\_stdout": "",

"msg": "MODULE FAILURE",

"rc": 1

}

Suggestions to resolve:

1. Verify that you have write access to the socket path described in the error message.
2. Follow the steps detailed in [enable network logging](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#enable-network-logging).

If the identified error message from the log file is:

2017-04-04 12:19:05,670 p=18591 u=fred | command timeout triggered, timeout value is 30 secs

or

2017-04-04 12:19:05,670 p=18591 u=fred | persistent connection idle timeout triggered, timeout value is 30 secs

Follow the steps detailed in [timeout issues](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#timeout-issues)

[**Category “Unable to open shell”**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id7)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#category-unable-to-open-shell)

**Platforms:** Any

The unable to open shell message means that the ansible-connection daemon has not been able to successfully talk to the remote network device. This generally means that there is an authentication issue. It is a “catch all” message, meaning you need to enable logging to find the underlying issues.

For example:

TASK [prepare\_eos\_tests : enable cli on remote device] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [veos01]: FAILED! => {"changed": false, "failed": true, "msg": "unable to open shell"}

or:

TASK [ios\_system : configure name\_servers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

task path:

fatal: [ios-csr1000v]: FAILED! => {

"changed": false,

"failed": true,

"msg": "unable to open shell",

}

Suggestions to resolve:

Follow the steps detailed in [enable\_network\_logging](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html" \l "enable-network-logging).

Once you’ve identified the error message from the log file, the specific solution can be found in the rest of this document.

[**Error: “[Errno -2] Name or service not known”**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id8)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-errno-2-name-or-service-not-known)

**Platforms:** Any

Indicates that the remote host you are trying to connect to can not be reached

For example:

2017-04-04 11:39:48,147 p=15299 u=fred | control socket path is /home/fred/.ansible/pc/ca5960d27a

2017-04-04 11:39:48,147 p=15299 u=fred | current working directory is /home/fred/git/ansible-inc/stable-2.3/test/integration

2017-04-04 11:39:48,147 p=15299 u=fred | using connection plugin network\_cli

2017-04-04 11:39:48,340 p=15299 u=fred | connecting to host veos01 returned an error

2017-04-04 11:39:48,340 p=15299 u=fred | [Errno -2] Name or service not known

Suggestions to resolve:

* If you are using the provider: options ensure that its suboption host: is set correctly.
* If you are not using provider: nor top-level arguments ensure your inventory file is correct.

[**Error: “Authentication failed”**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id9)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-authentication-failed)

**Platforms:** Any

Occurs if the credentials (username, passwords, or ssh keys) passed to ansible-connection (through ansible or ansible-playbook) can not be used to connect to the remote device.

For example:

**<ios01> ESTABLISH CONNECTION FOR USER:** cisco on PORT 22 TO ios01

<ios01> Authentication failed.

Suggestions to resolve:

If you are specifying credentials through password: (either directly or through provider:) or the environment variable *ANSIBLE\_NET\_PASSWORD* it is possible that paramiko (the Python SSH library that Ansible uses) is using ssh keys, and therefore the credentials you are specifying are being ignored. To find out if this is the case, disable “look for keys”. This can be done like this:

export ANSIBLE\_PARAMIKO\_LOOK\_FOR\_KEYS=False

To make this a permanent change, add the following to your ansible.cfg file:

**[paramiko\_connection]**

look\_for\_keys **=** False

[**Error: “connecting to host <hostname> returned an error” or “Bad address”**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id10)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-connecting-to-host-hostname-returned-an-error-or-bad-address)

This may occur if the SSH fingerprint hasn’t been added to Paramiko’s (the Python SSH library) know hosts file.

When using persistent connections with Paramiko, the connection runs in a background process. If the host doesn’t already have a valid SSH key, by default Ansible will prompt to add the host key. This will cause connections running in background processes to fail.

For example:

2017-04-04 12:06:03,486 p=17981 u=fred | using connection plugin network\_cli

2017-04-04 12:06:04,680 p=17981 u=fred | connecting to host veos01 returned an error

2017-04-04 12:06:04,682 p=17981 u=fred | (14, 'Bad address')

2017-04-04 12:06:33,519 p=17981 u=fred | number of connection attempts exceeded, unable to connect to control socket

2017-04-04 12:06:33,520 p=17981 u=fred | persistent\_connect\_interval=1, persistent\_connect\_retries=30

Suggestions to resolve:

Use ssh-keyscan to pre-populate the known\_hosts. You need to ensure the keys are correct.

ssh-keyscan veos01

or

You can tell Ansible to automatically accept the keys

Environment variable method:

export **ANSIBLE\_PARAMIKO\_HOST\_KEY\_AUTO\_ADD=**True

ansible-playbook ...

ansible.cfg method:

ansible.cfg

**[paramiko\_connection]**

host\_key\_auto\_add **=** True

[**Error: “No authentication methods available”**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id11)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-no-authentication-methods-available)

For example:

2017-04-04 12:19:05,670 p=18591 u=fred | creating new control socket for host veos01:None as user admin

2017-04-04 12:19:05,670 p=18591 u=fred | control socket path is /home/fred/.ansible/pc/ca5960d27a

2017-04-04 12:19:05,670 p=18591 u=fred | current working directory is /home/fred/git/ansible-inc/ansible-workspace-2/test/integration

2017-04-04 12:19:05,670 p=18591 u=fred | using connection plugin network\_cli

2017-04-04 12:19:06,606 p=18591 u=fred | connecting to host veos01 returned an error

2017-04-04 12:19:06,606 p=18591 u=fred | No authentication methods available

2017-04-04 12:19:35,708 p=18591 u=fred | connect retry timeout expired, unable to connect to control socket

2017-04-04 12:19:35,709 p=18591 u=fred | persistent\_connect\_retry\_timeout is 15 secs

Suggestions to resolve:

No password or SSH key supplied

[**Clearing Out Persistent Connections**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id12)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#clearing-out-persistent-connections)

**Platforms:** Any

In Ansible 2.3, persistent connection sockets are stored in ~/.ansible/pc for all network devices. When an Ansible playbook runs, the persistent socket connection is displayed when verbose output is specified.

<switch> socket\_path: /home/fred/.ansible/pc/f64ddfa760

To clear out a persistent connection before it times out (the default timeout is 30 seconds of inactivity), simple delete the socket file.

[**Timeout issues**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id13)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#timeout-issues)

[**Persistent connection idle timeout**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id14)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#persistent-connection-idle-timeout)

By default, ANSIBLE\_PERSISTENT\_CONNECT\_TIMEOUT is set to 30 (seconds). You may see the following error if this value is too low:

2017-04-04 12:19:05,670 p=18591 u=fred | persistent connection idle timeout triggered, timeout value is 30 secs

Suggestions to resolve:

Increase value of persistent connection idle timeout:

export **ANSIBLE\_PERSISTENT\_CONNECT\_TIMEOUT=**60

To make this a permanent change, add the following to your ansible.cfg file:

**[persistent\_connection]**

connect\_timeout **=** 60

[**Command timeout**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id15)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#command-timeout)

By default, ANSIBLE\_PERSISTENT\_COMMAND\_TIMEOUT is set to 30 (seconds). Prior versions of Ansible had this value set to 10 seconds by default. You may see the following error if this value is too low:

2017-04-04 12:19:05,670 p=18591 u=fred | command timeout triggered, timeout value is 30 secs

Suggestions to resolve:

* Option 1 (Global command timeout setting): Increase value of command timeout in configuration file or by setting environment variable.
* export ANSIBLE\_PERSISTENT\_COMMAND\_TIMEOUT=60

To make this a permanent change, add the following to your ansible.cfg file:

**[persistent\_connection]**

command\_timeout **=** 60

* Option 2 (Per task command timeout setting): Increase command timeout per task basis. All network modules support a timeout value that can be set on a per task basis. The timeout value controls the amount of time in seconds before the task will fail if the command has not returned.

For local connection type:

Suggestions to resolve:

Some modules support a timeout option, which is different to the timeout keyword for tasks.

**-** **name:** save running-config

**cisco.ios.ios\_command:**

**commands:** copy running-config startup-config

**provider:** "{{cli}}"

**timeout:** 30

Suggestions to resolve:

If the module does not support the timeout option directly, most networking connection plugins can enable similar functionality with the ansible\_command\_timeout variable.

**-** **name:** save running-config

**cisco.ios.ios\_command:**

**commands:** copy running-config startup-config

**vars:**

**ansible\_command\_timeout:** 60

Some operations take longer than the default 30 seconds to complete. One good example is saving the current running config on IOS devices to startup config. In this case, changing the timeout value from the default 30 seconds to 60 seconds will prevent the task from failing before the command completes successfully.

[**Persistent connection retry timeout**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id16)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#persistent-connection-retry-timeout)

By default, ANSIBLE\_PERSISTENT\_CONNECT\_RETRY\_TIMEOUT is set to 15 (seconds). You may see the following error if this value is too low:

2017-04-04 12:19:35,708 p=18591 u=fred | connect retry timeout expired, unable to connect to control socket

2017-04-04 12:19:35,709 p=18591 u=fred | persistent\_connect\_retry\_timeout is 15 secs

Suggestions to resolve:

Increase the value of the persistent connection idle timeout. Note: This value should be greater than the SSH timeout value (the timeout value under the defaults section in the configuration file) and less than the value of the persistent connection idle timeout (connect\_timeout).

export ANSIBLE\_PERSISTENT\_CONNECT\_RETRY\_TIMEOUT=30

To make this a permanent change, add the following to your ansible.cfg file:

**[persistent\_connection]**

connect\_retry\_timeout **=** 30

[**Timeout issue due to platform specific login menu with network\_cli connection type**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id17)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#timeout-issue-due-to-platform-specific-login-menu-with-network-cli-connection-type)

In Ansible 2.9 and later, the network\_cli connection plugin configuration options are added to handle the platform specific login menu. These options can be set as group/host or tasks variables.

Example: Handle single login menu prompts with host variables

**$** cat host\_vars/<hostname>.yaml

---

ansible\_terminal\_initial\_prompt:

- "Connect to a host"

ansible\_terminal\_initial\_answer:

- "3"

Example: Handle remote host multiple login menu prompts with host variables

**$** cat host\_vars/<inventory-hostname>.yaml

---

ansible\_terminal\_initial\_prompt:

- "Press any key to enter main menu"

- "Connect to a host"

ansible\_terminal\_initial\_answer:

- "\\r"

- "3"

ansible\_terminal\_initial\_prompt\_checkall: True

To handle multiple login menu prompts:

* The values of ansible\_terminal\_initial\_prompt and ansible\_terminal\_initial\_answer should be a list.
* The prompt sequence should match the answer sequence.
* The value of ansible\_terminal\_initial\_prompt\_checkall should be set to True.

**Note**

If all the prompts in sequence are not received from remote host at the time connection initialization it will result in a timeout.

[**Playbook issues**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id18)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#playbook-issues)

This section details issues are caused by issues with the Playbook itself.

[**Error: “Unable to enter configuration mode”**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id19)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#error-unable-to-enter-configuration-mode)

**Platforms:** Arista EOS and Cisco IOS

This occurs when you attempt to run a task that requires privileged mode in a user mode shell.

For example:

TASK [ios\_system : configure name\_servers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

task path:

fatal: [ios-csr1000v]: FAILED! => {

"changed": false,

"failed": true,

"msg": "unable to enter configuration mode",

}

Suggestions to resolve:

Use connection: ansible.netcommon.network\_cli and become: true

[**Proxy Issues**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id20)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#proxy-issues)

[**delegate\_to vs ProxyCommand**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id21)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#delegate-to-vs-proxycommand)

In order to use a bastion or intermediate jump host to connect to network devices over cli transport, network modules support the use of ProxyCommand.

To use ProxyCommand, configure the proxy settings in the Ansible inventory file to specify the proxy host.

**[nxos]**

nxos01

nxos02

**[nxos:vars]**

ansible\_ssh\_common\_args**=**'-o ProxyCommand="ssh -W %h:%p -q bastion01"'

With the configuration above, simply build and run the playbook as normal with no additional changes necessary. The network module will now connect to the network device by first connecting to the host specified in ansible\_ssh\_common\_args, which is bastion01 in the above example.

You can also set the proxy target for all hosts by using environment variables.

export **ANSIBLE\_SSH\_ARGS=**'-o ProxyCommand="ssh -W %h:%p -q bastion01"'

[**Using bastion/jump host with netconf connection**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id22)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#using-bastion-jump-host-with-netconf-connection)

[**Enabling jump host setting**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id23)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#enabling-jump-host-setting)

**Bastion/jump host with netconf connection can be enabled by:**

* Setting Ansible variable ansible\_netconf\_ssh\_config either to True or custom ssh config file path
* Setting environment variable ANSIBLE\_NETCONF\_SSH\_CONFIG to True or custom ssh config file path
* Setting ssh\_config = 1 or ssh\_config = <ssh-file-path> under netconf\_connection section

If the configuration variable is set to 1 the proxycommand and other ssh variables are read from default ssh config file (~/.ssh/config).

If the configuration variable is set to file path the proxycommand and other ssh variables are read from the given custom ssh file path

[**Example ssh config file (~/.ssh/config)**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id24)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#example-ssh-config-file-ssh-config)

Host jumphost

HostName jumphost.domain.name.com

User jumphost-user

IdentityFile "/path/to/ssh-key.pem"

Port 22

*# Note: Due to the way that Paramiko reads the SSH Config file,*

*# you need to specify the NETCONF port that the host uses.*

*# In other words, it does not automatically use ansible\_port*

*# As a result you need either:*

Host junos01

HostName junos01

ProxyCommand ssh -W %h**:**22 jumphost

*# OR*

Host junos01

HostName junos01

ProxyCommand ssh -W %h**:**830 jumphost

*# Depending on the netconf port used.*

Example Ansible inventory file

**[junos]**

junos01

**[junos:vars]**

ansible\_connection**=**ansible.netcommon.netconf

ansible\_network\_os**=**junipernetworks.junos.junos

ansible\_user**=**myuser

ansible\_password**=**!vault...

**Note**

Using ProxyCommand with passwords through variables

By design, SSH doesn’t support providing passwords through environment variables. This is done to prevent secrets from leaking out, for example in ps output.

We recommend using SSH Keys, and if needed an ssh-agent, rather than passwords, where ever possible.

[**Miscellaneous Issues**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id25)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#miscellaneous-issues)

[**Intermittent failure while using ansible.netcommon.network\_cli connection type**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id26)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#intermittent-failure-while-using-ansible-netcommon-network-cli-connection-type)

If the command prompt received in response is not matched correctly within the ansible.netcommon.network\_cli connection plugin the task might fail intermittently with truncated response or with the error message operation requires privilege escalation. Starting in 2.7.1 a new buffer read timer is added to ensure prompts are matched properly and a complete response is send in output. The timer default value is 0.2 seconds and can be adjusted on a per task basis or can be set globally in seconds.

Example Per task timer setting

**-** **name:** gather ios facts

**cisco.ios.ios\_facts:**

**gather\_subset:** all

**register:** result

**vars:**

**ansible\_buffer\_read\_timeout:** 2

To make this a global setting, add the following to your ansible.cfg file:

**[persistent\_connection]**

buffer\_read\_timeout **=** 2

This timer delay per command executed on remote host can be disabled by setting the value to zero.

[**Task failure due to mismatched error regex within command response using ansible.netcommon.network\_cli connection type**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id27)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#task-failure-due-to-mismatched-error-regex-within-command-response-using-ansible-netcommon-network-cli-connection-type)

In Ansible 2.9 and later, the ansible.netcommon.network\_cli connection plugin configuration options are added to handle the stdout and stderr regex to identify if the command execution response consist of a normal response or an error response. These options can be set group/host variables or as tasks variables.

Example: For mismatched error response

**-** **name:** fetch logs from remote host

**cisco.ios.ios\_command:**

**commands:**

**-** show logging

Playbook run output:

TASK [first fetch logs] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [ios01]: FAILED! => {

"changed": false,

"msg": "RF Name:\r\n\r\n <--nsip-->

\"IPSEC-3-REPLAY\_ERROR: Test log\"\r\n\*Aug 1 08:36:18.483: %SYS-7-USERLOG\_DEBUG:

Message from tty578(user id: ansible): test\r\nan-ios-02#"}

Suggestions to resolve:

Modify the error regex for individual task.

**-** **name:** fetch logs from remote host

**cisco.ios.ios\_command:**

**commands:**

**-** show logging

**vars:**

**ansible\_terminal\_stderr\_re:**

**-** **pattern:** 'connectiontimedout'

**flags:** 're.I'

The terminal plugin regex options ansible\_terminal\_stderr\_re and ansible\_terminal\_stdout\_re have pattern and flags as keys. The value of the flags key should be a value that is accepted by the re.compile python method.

[**Intermittent failure while using ansible.netcommon.network\_cli connection type due to slower network or remote target host**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#id28)[**ℑ**](https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#intermittent-failure-while-using-ansible-netcommon-network-cli-connection-type-due-to-slower-network-or-remote-target-host)

In Ansible 2.9 and later, the ansible.netcommon.network\_cli connection plugin configuration option is added to control the number of attempts to connect to a remote host. The default number of attempts is three. After every retry attempt the delay between retries is increased by power of 2 in seconds until either the maximum attempts are exhausted or either the persistent\_command\_timeout or persistent\_connect\_timeout timers are triggered.

To make this a global setting, add the following to your ansible.cfg file:

**[persistent\_connection]**

network\_cli\_retries **=** 5