
Junos Ansible Modules Documentation

Release 1.1.0

Jeremy Schulman - Juniper Networks, Inc.

January 15, 2015

1	junos_shutdown - Shut down or reboot a device running Junos OS.	3
1.1	Synopsis	3
1.2	Options	3
1.3	Examples	3
2	junos_get_facts - Retrieve facts for a device running Junos OS.	5
2.1	Synopsis	5
2.2	Options	5
2.3	Examples	5
3	junos_zeroize - Erase all data, including configuration and log files, on a device running Junos OS.	7
3.1	Synopsis	7
3.2	Options	7
3.3	Examples	7
4	junos_install_config - Load a configuration file or snippet onto a device running Junos OS.	9
4.1	Synopsis	9
4.2	Options	9
4.3	Examples	9
5	junos_install_os - Install a Junos OS image.	11
5.1	Synopsis	11
5.2	Options	11
5.3	Examples	11

Contents:

junos_shutdown - Shut down or reboot a device running Junos OS.

Author Jeremy Schulman, Juniper Networks

- [Synopsis](#)
- [Options](#)
- [Examples](#)

1.1 Synopsis

New in version 1.0.0.

Shut down (power off) or reboot a device running Junos OS. This includes all Routing Engines in a Virtual Chassis or a dual Routing Engine system. This is equivalent to executing either the Junos OS **request system power-off** or **request system reboot** operational command.

1.2 Options

Note: Requires junos-eznc >= 1.1.0

1.3 Examples

```
- junos_shutdown:
  host={{ inventory_hostname }}
  shutdown="shutdown"
  reboot=yes
```

junos_get_facts - Retrieve facts for a device running Junos OS.

Author Jeremy Schulman, Juniper Networks

- [Synopsis](#)
- [Options](#)
- [Examples](#)

2.1 Synopsis

New in version 1.0.0.

Retrieve facts for a device running Junos OS, which includes information such as the serial number, product model, and Junos OS version. The module supports using both NETCONF and CONSOLE-based retrieval and returns the information as a JSON dictionary. The information is similar to facts gathered by other IT frameworks.

2.2 Options

Note: Requires junos-eznc >= 1.1.0

Note: Requires junos-netconf >= 1.0.0, when using the *console* option

2.3 Examples

```
# retrieve facts using NETCONF, assumes ssh-keys

- junos_get_facts: host={{ inventory_hostname }}
  register: junos

# retrieve facts using CONSOLE, assumes Amnesiac system
# root login, no password

- junos_get_facts:
  host={{ inventory_hostname }}
```

```
    user=root
    console="--telnet={{TERMSERV}},{{TERMSERVPORT}}}"
    savedir=/usr/local/junos/inventory
    register: junos

# access the facts

- name: version
  debug: msg="{{ junos.facts.version }}"
```

junos_zeroize - Erase all data, including configuration and log files, on a device running Junos OS.

Author Jeremy Schulman, Juniper Networks

- [Synopsis](#)
- [Options](#)
- [Examples](#)

3.1 Synopsis

New in version 1.0.0.

Execute the Junos OS **request system zeroize** command to remove all configuration information on the Routing Engines and reset all key values on a device running Junos OS. The command removes all data files, including customized configuration and log files, by unlinking the files from their directories. The command also removes all user-created files from the system including all plain-text passwords, secrets, and private keys for SSH, local encryption, local authentication, IPsec, RADIUS, TACACS+, and SNMP. This command reboots the device and sets it to the factory default configuration. After the reboot, you must log in through the console as root in order to access the device.

3.2 Options

Note: Requires junos-eznc >= 1.1.0

Note: Requires junos-netconify >= 1.0.0, when using the *console* option

3.3 Examples

```
- junos_zeroize:
  host={{ inventory_hostname }}
  zeroize="zeroize"
```

Note: You **MUST** either use the *host* option or the *console* option to designate how the device is accessed.

junos_install_config - Load a configuration file or snippet onto a device running Junos OS.

Author Jeremy Schulman, Juniper Networks

- [Synopsis](#)
- [Options](#)
- [Examples](#)

4.1 Synopsis

New in version 1.0.0.

Load a complete Junos OS configuration (overwrite) or merge a configuration snippet onto a device running Junos OS and commit it. The default behavior is to perform a **load merge** operation (`overwrite='no'`). This module performs an atomic lock/edit/unlock. If the process fails at any step, then all configuration changes are discarded. You can load the configuration using either NETCONF or the CONSOLE port. Specify the *console* option to use the CONSOLE port. You provide the configuration data in a file. Supported formats when using NETCONF include ASCII text, Junos XML elements, and Junos OS **set** commands. Configurations performed through the console must only use ASCII text formatting.

4.2 Options

Note: Requires `junos-eznc >= 1.1.0`

Note: Requires `junos-netconf >= 1.0.0`, when using the *console* option

4.3 Examples

```
# load merge a change to the Junos OS configuration using NETCONF

- junos_install_config:
    host={{ inventory_hostname }}
```

```
    file=banner.conf

# load overwrite a new Junos OS configuration using the CONSOLE port

- junos_install_config:
    host={{ inventory_hostname }}
    console="--telnet={{ TERMSERV }}, {{ TERMSERV_PORT }}"
    file=default_new_switch.conf
    overwrite=yes

# load merge a change to the Junos OS configuration using NETCONF and supplying a commit log message
- junos_install_config:
    host={{ inventory_hostname }}
    file=banner.conf
    comment="configured by ansible"
```

junos_install_os - Install a Junos OS image.

Author Jeremy Schulman, Juniper Networks

- [Synopsis](#)
- [Options](#)
- [Examples](#)

5.1 Synopsis

New in version 1.0.0.

Install a Junos OS image on one or more Routing Engines. This module supports installations on single Routing Engine devices, MX Series routers with dual Routing Engines, and EX Series switches in a non-mixed Virtual Chassis. This action is equivalent to performing the Junos OS **request system software add** operational command. If the existing Junos OS version matches the desired version, no action is performed, and the “changed” attribute reports False. If the existing version does not match, then the module performs the following actions (1) Computes the MD5 checksum of the package located on the server. (2) Copies the Junos OS software package to the device running Junos OS. (3) Computes the MD5 checksum on the device running Junos OS and compares the two. (4) Installs the Junos OS software package. (5) Reboots the device (default). Running the module in check mode reports whether the current Junos OS version matches the desired version.

5.2 Options

Note: Requires py-junos-eznc >= 1.1.0

5.3 Examples

```
- junos_install_os:
  host={{ inventory_hostname }}
  version=12.1X46-D10.2
  package=/usr/local/junos/images/junos-vsrx-12.1X46-D10.2-domestic.tgz
  logfile=/usr/local/junos/log/software.log
```