
Junos Ansible Modules Documentation

Release 1.2.1

Jeremy Schulman - Juniper Networks, Inc.

February 26, 2016

1	junos_rollback - Rollback configuration of device	3
2	junos_install_os - Install a Junos OS image.	5
3	junos_commit - Execute commit on device	7
4	junos_get_config - Retrieve configuration of device	9
5	junos_shutdown - Shut down or reboot a device running Junos OS.	11
6	junos_get_facts - Retrieve facts for a device running Junos OS.	13
7	junos_rpc - run given rpc	15
8	junos_srx_cluster - Create an srx chassis cluster for cluster capable srx running Junos OS.	17
9	junos_zeroize - Erase all data, including configuration and log files, on a device running Junos OS.	19
10	junos_cli - Execute CLI on device and save the output locally	21
11	junos_install_config - Load a configuration file or snippet onto a device running Junos OS.	23

Contents:

junos_rollback - Rollback configuration of device

Author Rick Sherman, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

1.1 Synopsis

New in version 1.2.0.

Rollback the configuration of a device running Junos

1.2 Options

Note: Requires junos-eznc >= 1.2.2

1.3 Examples

```
- junos_rollback:
  host: "{{ inventory_hostname }}"
  logfile=rollback.log
  diffs_file=rollback.diff
  rollback=1
  comment="Rolled back by Ansible"
  confirm=5
```

junos_install_os - Install a Junos OS image.

Author Jeremy Schulman, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

2.1 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.

2.2 Options

Note: Requires py-junos-eznc >= 1.2.2

2.3 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
```

junos_commit - Execute commit on device

Author Rick Sherman, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

3.1 Synopsis

New in version 1.2.0.

Execute a Commit on a device running Junos independently of loading a configuration

3.2 Options

Note: Requires junos-eznc >= 1.2.2

3.3 Examples

```
- junos_commit:
  host: "{{ inventory_hostname }}"
  logfile=changes.log
  comment="Non load commit"
```

junos_get_config - Retrieve configuration of device

Author Rick Sherman, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

4.1 Synopsis

New in version 1.2.0.

Retrieve the configuration of a device running Junos and save it to a file. **Note** unicode chars will be converted to ‘??’ as also done in PyEZ

4.2 Options

Note: Requires junos-eznc >= 1.2.2

4.3 Examples

```
- junos_get_config:
  host: "{{ inventory_hostname }}"
  logfile: get_config.log
  dest: "{{ inventory_hostname }}.xml"
  format: xml
  filter: "interfaces"
  options: {inherit: inherit, groups: groups}
```

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

Author Jeremy Schulman, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

5.1 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.

5.2 Options

Note: Requires junos-eznc >= 1.2.2

5.3 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*

6.1 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.

6.2 Options

Note: Requires junos-eznc >= 1.2.2

Note: Requires junos-netconfify >= 1.0.1, when using the *console* option

6.3 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 }},{{ TERMSESRVPORT }}"
  savedir=/usr/local/junos/inventory
  register: junos

# access the facts

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

junos_rpc - run given rpc

Author Nitin Kumar, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

7.1 Synopsis

New in version 1.9.

run given rpc

7.2 Options

Note: Requires junos-eznc >= 1.0.0

7.3 Examples

```
# retrieve rpc response using NETCONF

- junos_rpc:
  host={{ inventory_hostname }}
  rpc=get-interface-information
  dest=get_interface_information.conf
  register=junos

- junos_rpc:
  host={{ inventory_hostname }}
  rpc=get-interface-information
  kwargs="interface_name=em0"
  format=xml/text/json
  dest=get_interface_information.conf
```

```
    register=junos

# print the config
- name: version
  debug: msg="{{ junos.rpc_reply }}"

# Example to fetch device configuration
- name: Get Device Configuration
  junos_rpc:
    host={{ inventory_hostname }}
    rpc=get-config
    dest=get_config.conf

# Example to fetch device configuration
- name: Get Device Configuration for interface
  junos_rpc:
    host={{ inventory_hostname }}
    rpc=get-config
    filter_xml="<configuration><interfaces/></configuration>"
    dest=get_config.conf
  register: junos
```

junos_srx_cluster - Create an srx chassis cluster for cluster capable srx running Junos OS.

Author Patrik Bok, Ashley Burston, Rick Sherman, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

8.1 Synopsis

New in version 1.2.0.

Create an srx chassis cluster and reboot the device. The device must be capable of forming an srx cluster and have the correct cables installed.

8.2 Options

Note: Requires junos-eznc >= 1.2.2

8.3 Examples

```
-junos_srx_cluster:
  host={{ inventory_hostname }}
  console="--port={{ serial }}"
  user=rick
  passwd=password123
  cluster_enable=true
  logfile=cluster.log
  cluster_id={{ cluster_id }}
  node={{ node_id }}

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

```
user=rick
passwd=password123
cluster_enable=false
logfile=cluster.log
```

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

Author Jeremy Schulman, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

9.1 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.

9.2 Options

Note: Requires junos-eznc >= 1.2.2

Note: Requires junos-netconify >= 1.0.1, when using the *console* option

9.3 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_cli - Execute CLI on device and save the output locally

Author Damien Garros, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

10.1 Synopsis

New in version 1.2.0.

Execute CLI on device and save the output locally on a file

10.2 Options

Note: Requires junos-eznc >= 1.2.2

10.3 Examples

```
- junos_cli:
  host: "{{ inventory_hostname }}"
  cli: "show chassis hardware"
  logfile: cli.log
  dest: "{{ inventory_hostname }}.xml"
  format: xml
```

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

Author Jeremy Schulman, Juniper Networks

- *Synopsis*
- *Options*
- *Examples*

11.1 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.

11.2 Options

Note: Requires junos-eznc >= 1.2.2

Note: Requires junos-netconfify >= 1.0.1, when using the *console* option

11.3 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"

# load replace a change to the Junos OS configuration using NETCONF
- junos_install_config:
    host={{ inventory_hostname }}
    file=snmp.conf
    replace=yes
```