Installation of Ansible u

sing YUM

As mentioned before I will be using one control machine and a target machine. To start with installation, perform the steps as shown below in both the machines.

# yum update

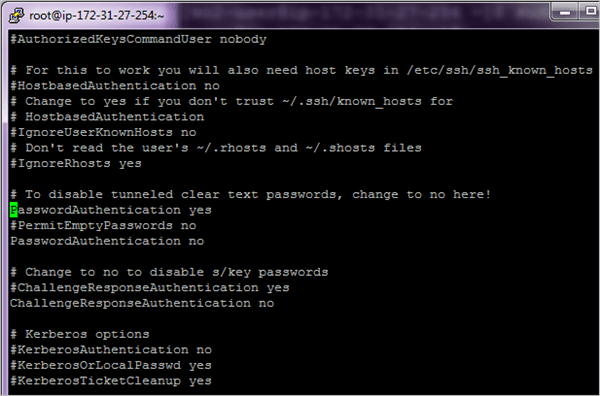
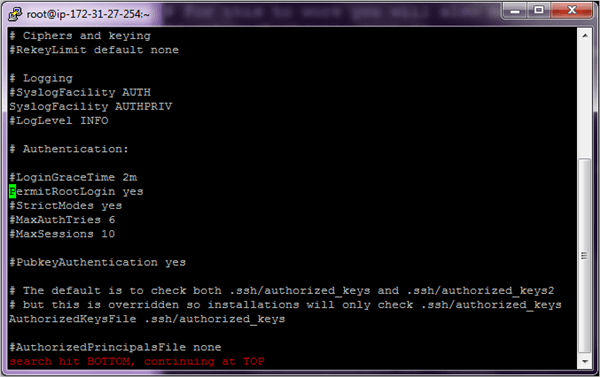
To update the OS

**a)** Create a common id on both the machines, for **Example**, **ansible**with SUDO privileges. This id will be used for communicating across all the machines involved for automation of tasks.

**# useradd ansible**

**# passwd ansible**

**b)** Edit the**/etc/ssh/sshd\_config**file on the **control machine** and uncomment out the lines for**PasswordAuthentication and PermitRootLogin**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/2.Edit-the-sshd_config-file.png)[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/3.Edit-config-file.png)

Perform the above steps on both the machines. Once

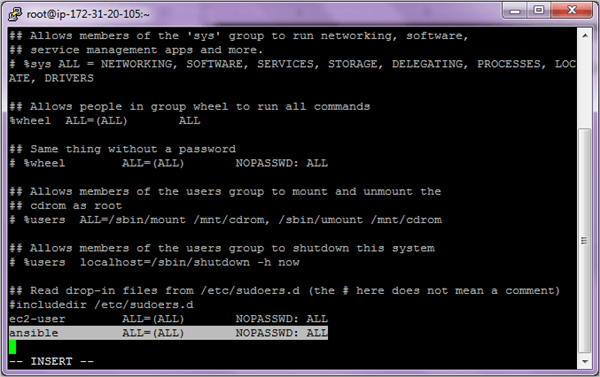
completed, restart the **sshd** service on both the machines.

**# systemctl restart sshd**

**c)** For complete automation of tasks, we will need passwordless SSH authentication else the whole process will not be used if you have to key in the password every time.

So post the changes done above if we run the command ssh <target machine> and ssh <control machine> we will need to key in the password every time which is not the right procedure to execute Ansible tasks.

**d)** To enable passwordless authentication to perform the steps shown below. Firstly add the user **ansible**to the **/etc/sudoers** file on both the machines which will enable the user **ansible**to run any command which requires root privileges.

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/4.add-the-user-ansible.png)

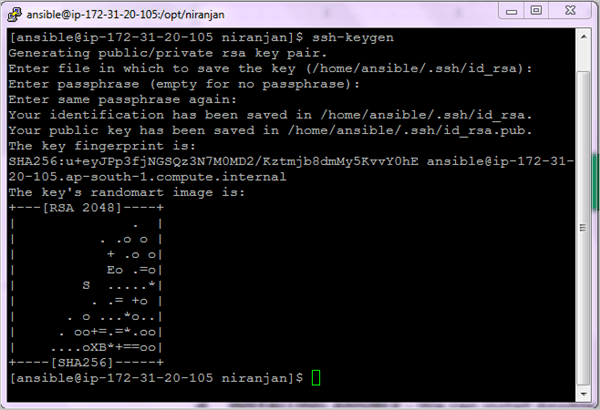
Save and exit the file after adding the user.

**e)** Going forward we will use the user **ansible**to perform all the steps. So switch to the user **ansible.**

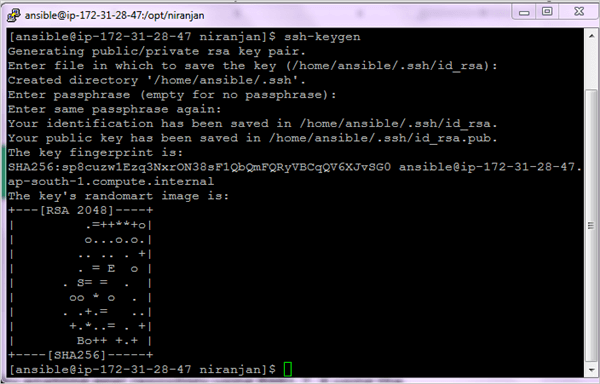
**Control Machine su – ansible AND Target Machine su – ansible**

[5.user ansible](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/5.user-ansible.png)

**Control Machine ssh-keygen**

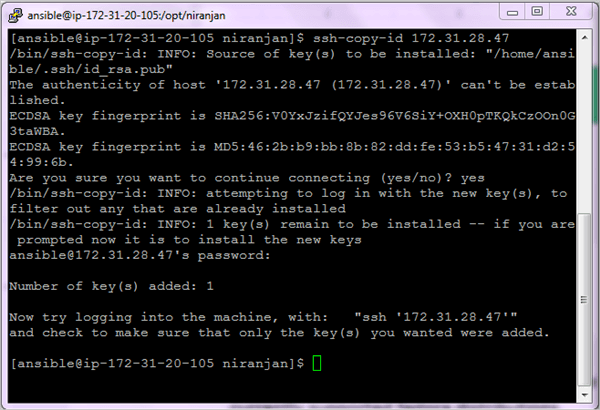
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/6.Control-Machine-ssh-keygen.png)

**Target Machine ssh-keygen**

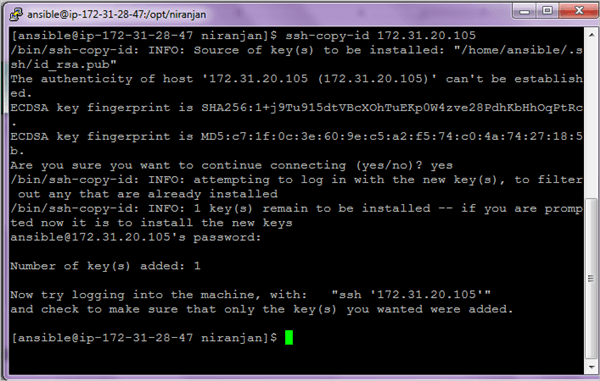
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/6.Target-Machine-ssh-keygen.png)

Copy the ssh key to the target machine and vice versa.

**Control Machine ssh-copy-id <IP-Address-Host-Machine>**

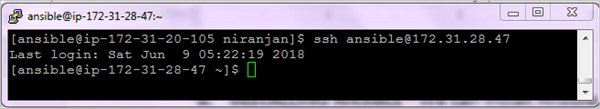
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/7.Control-Machine-ssh-copy-id.png)

**Target Machine ssh-copy-id <IP-Address-Control-Machine>**

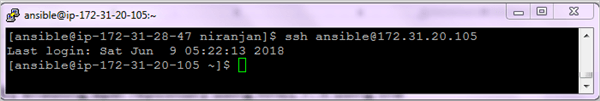
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/7.Target-Machine-ssh-copy-id.png)

We are now able to log in without entering the password. After the check out of the ssh connectivity on both the machines and be logged in as ansible user.

**Control Machine: ssh ansible@<IP-Address-Host-Machine**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/8.Control-Machine-login-without-pwd.png)

**Target Machine: ssh ansible@<IP-Address-Control-Machine>**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/8.Target-machine-login-without-pwd.png)

**f)** Install wget if not installed on both the machines.

**$ sudo yum install wget -y**

**g)** We can now install **ansible** **on the Control machine only**by enabling the EPEL repo from fedora which provides add-on software packages. Perform the following steps to install **ANSIBLE.**

**$ wget http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm**

**$ sudo rpm -ivh epel-release-latest-7.noarch.rpm**

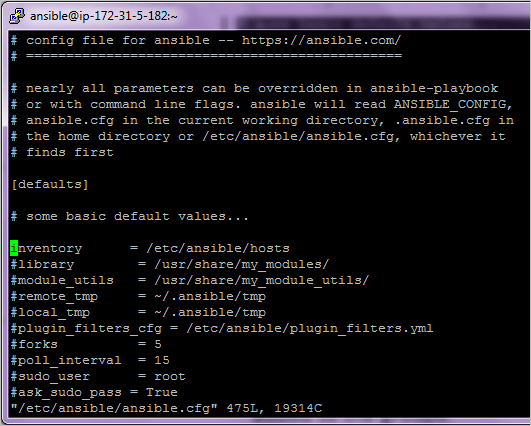
**$ sudo yum install ansible -y**

**$ ansible --version**

**The ansible version used is 2.5.3**

**h)** Edit the **ansible.cfg** file and enable the inventory file parameter on the Control machine.

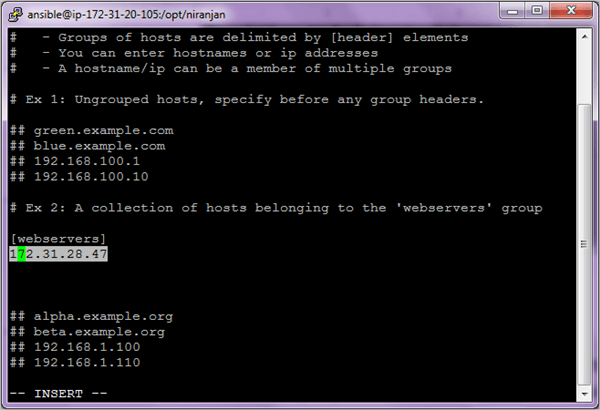
$ **sudo vi /etc/ansible/ansible.cfg**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/9.enable-the-inventory-file-parameter-on-the-Control-machine.png)

**i)** Ansible uses the concept of Inventory to manage and track the target machines. By default, this file is located in **/etc/ansible/hosts** and can be changed as well. A host file consists of groups for better classification and multiple machines under the group. All the required machines can be added to those groups.

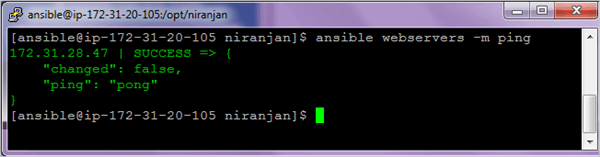
Every group is denoted by a square bracket and a group name within. A server can actually exist in multiple groups.

Edit the inventory file **/etc/ansible/hosts**and add all the servers which need to be managed.

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/10.add-all-the-servers-which-needs-to-be-managed.png)

**j)** To test the connectivity of the servers under the webserver's group run the **ansible ping** command as shown. Here **ping** is a module which performs a particular function to test whether the hosts can be connected as defined in the inventory file or not. We will see more about various modules and its examples in the next section.

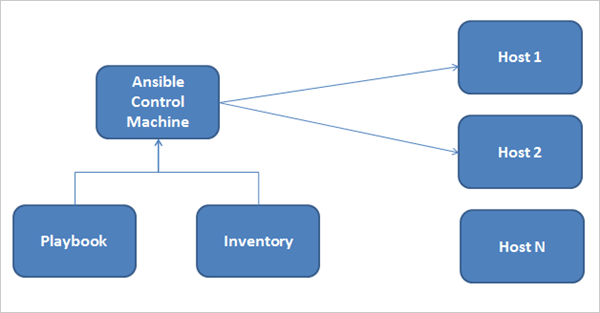
**$ ansible webservers –m ping**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/11.To-test-the-connectivity-of-the-servers.png)

To list the hosts in the inventory file, you can run the below command

**$ ansible webservers --list-hosts**

Ansible Usage

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/12.Overview-of-Ansible-usage.png)

**Ansible consist of 3 main components**

* Control Machine
* Inventory
* Playbook

The control machine manages the execution of the Playbook. It can be installed on your laptop or on any machine on the internet.

The Inventory file provides a complete list of all the target machines on which various modules are run by doing an ssh connection and install the necessary software’s.

The playbook consists of steps that the control mechanism will perform on the servers defined in the inventory file.

Very important to understand here is that Ansible interacts with all the servers defined in the inventory through the SSH protocol which is a secure method of remote login. Every operation is done and file transfer is encrypted.

So as you would have seen in the previous section Ansible does not use any kind of database for installation and is very easy to install, we will now proceed with the actual usage of Ansible starting with Modules which is the main building block.

Ansible Modules

Modules are the main building blocks of Ansible and are basically reusable scripts that are used by Ansible playbooks. Ansible comes with a number of reusable modules. These include functionality for controlling services, software package installation, working with files and directories etc.

The syntax is as follows while running the ad-hoc commands which help in running single or simple tasks just once and which need not be run later. For **E.g.** just installing Tomcat on all servers.

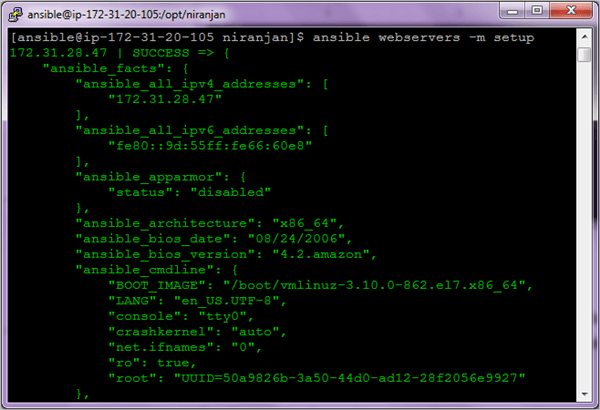
ansible hostORgroup -m module\_name -a "arguments" -u username --become

Let’s have a look at some of the most popular Ansible modules and their usage through the ad-hoc commands and later on in the playbook.

#1) Setup Module

To get information about the network or hardware or OS version or memory related information the setup module will help to gather the same about the target machines. On the control, the machine runs the below command.

**$ ansible webservers –m setup**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/13.Setup-Module.png)

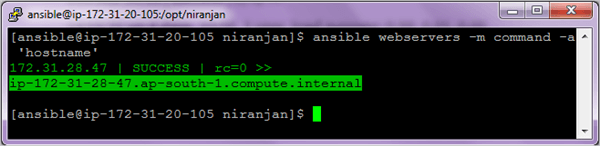
#2) Command Module

The command module simply executes a specific command on the target machine and gives the output.

**Some of the Examples are given below**

**$ ansible webservers –m command - an ‘uptime’**

**$ ansible webservers –m command –a ‘hostname’**

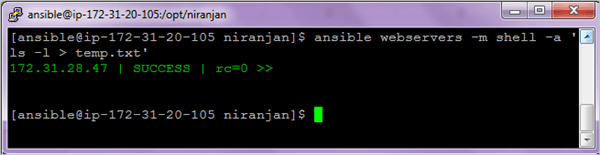
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/14.command-module.png)

#3) Shell Module

To execute any command in the shell of your choice you can use the Shell module. The shell module commands are run in /bin/sh shell and you can make use of the operators like ‘>’ or ‘|’ (pipe symbol or even environment variables.

So primarily the difference between the Shell and Command module is that if you actually do not need to use the operators like the ones mentioned then you could use the command module.

**$ ansible webservers -m shell -a 'ls -l > temp.txt'**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/15.Shell-Module.png)

On the machines under webservers group check for the file created and run the command to view the text file.

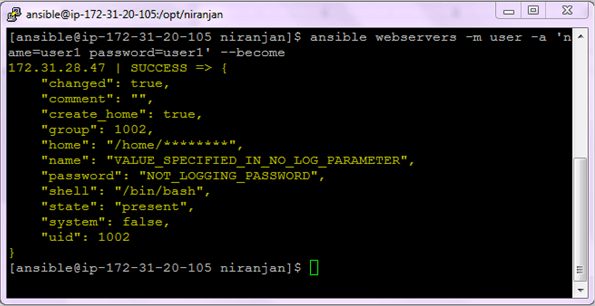
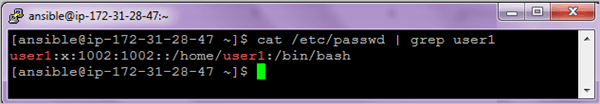
**$ ansible webservers –m command -a ‘cat temp.txt’**

#4) User Module

Using this module one can create or delete users.

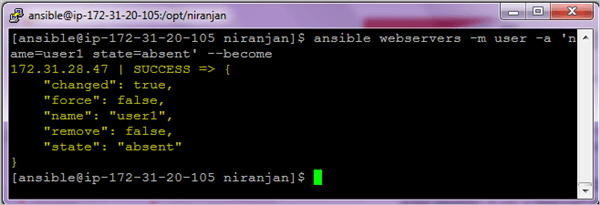
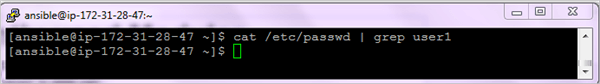
**To add user**

**$ ansible webservers -m user -a 'name=user1 password=user1' --become**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/16.User-Module-to-add-user.png)[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/16.user-module-delt-user.png)

**To delete user**

**$ ansible webservers -m user -a 'name=user1 state=absent' --become**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/17.to-delete-user1.png)[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/17.to-delete-user2.png)

**Options:**

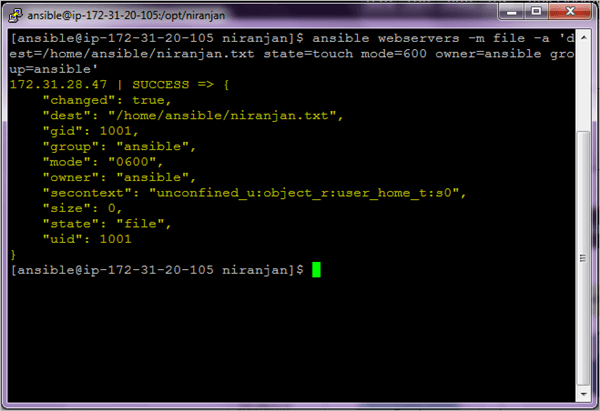
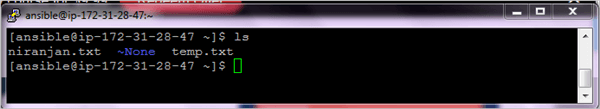
* **become** – Privilege to the superuser to run the command
* **state=absent** to delete the user

#5) File Module

This module is used to create files, directories, set, or change file permissions and ownership etc

**Example 1:** Create a file

**$ ansible webservers -m file -a 'dest=/home/ansible/niranjan.txt state=touch mode=600 owner=ansible group=ansible'**

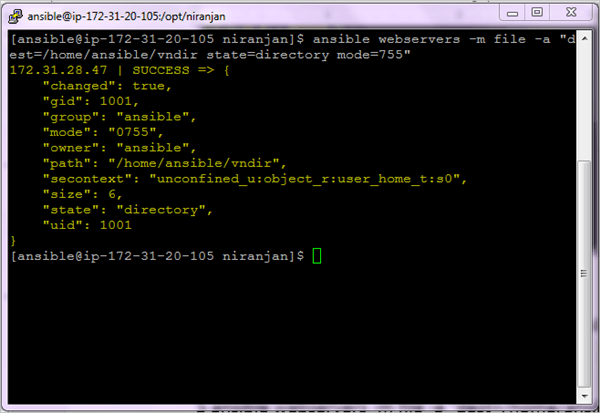
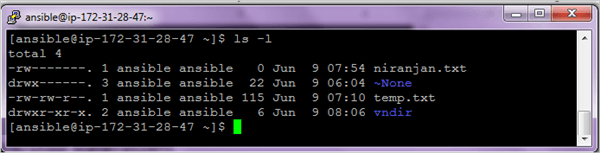
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/18.file-module-create-file1.png)[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/18.file-module-create-file2.png)

**Example 2:** Create a directory

To create a directory using the file module, you need to set two parameters.

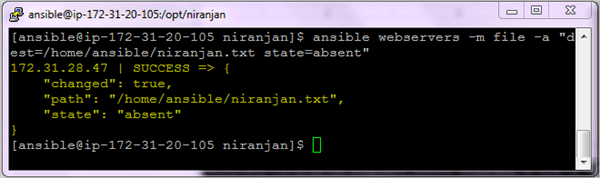
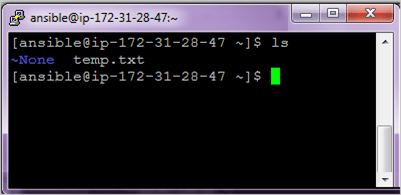
* Path(alias – name, dest) – This is the absolute path of the directory to be created.
* State – You should enter the value as ‘directory.’ By default, the value is ‘file’.

**$ ansible webservers -m file -a "dest=/home/ansible/vndir state=directory mode=755"**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/19.create-dir1.png)[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/19.create-dir2.png)

**Example 3:**Delete a file

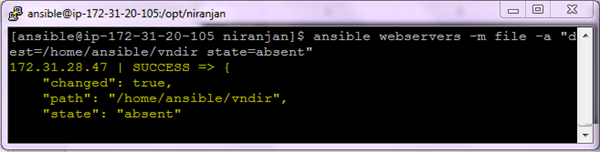
**$ ansible webservers -m file -a "dest=/home/ansible/niranjan.txt state=absent"**

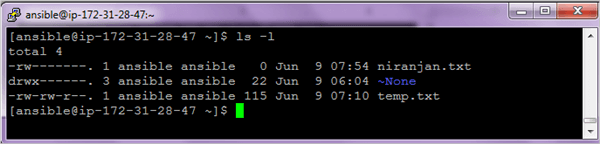
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/20.delete-file1.png)[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/20.delete-file2.png)

**Example 4:** Delete a directory

You can delete a directory by setting the state parameter value to **absent**. The directory and all its contents will be deleted.

**$ ansible webservers -m file -a "dest=/home/ansible/vndir state=absent"**

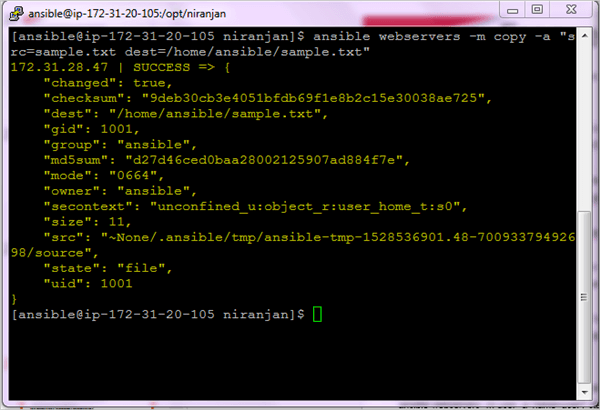
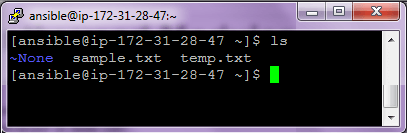
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/21.-delete-dir1.png)

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/21.-delete-dir2.png)

#6) Copy Module

It is used for copying files to multiple target machines.

**$ ansible webservers -m copy -a "src=sample.txt dest=/home/ansible/sample.txt"**

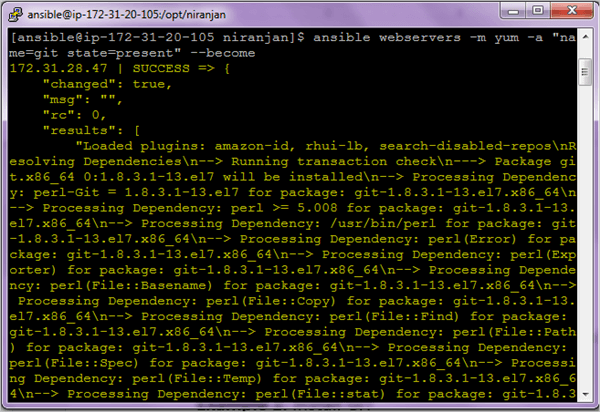
[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/22.Copy-Module1.png)[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/22.Copy-Module2.png)

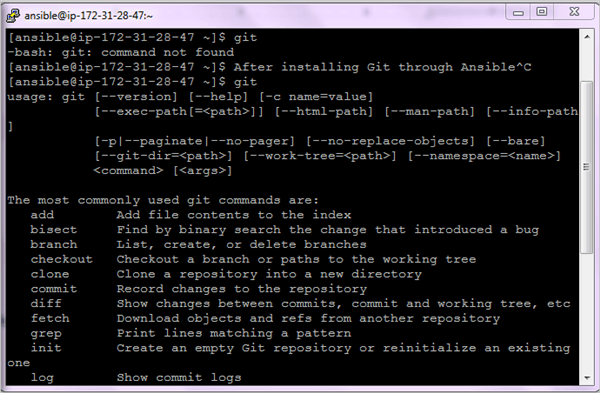
#7) Managing Software Packages

If you need to install software packages through ‘yum’ or ‘apt’ you can use the below commands.

**Example 1:** Install GIT

**$ ansible webservers –m yum -a “name=git state=present” --become**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/23.Managing-Software-Packages1.png)

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/23.Managing-Software-Packages2.png)

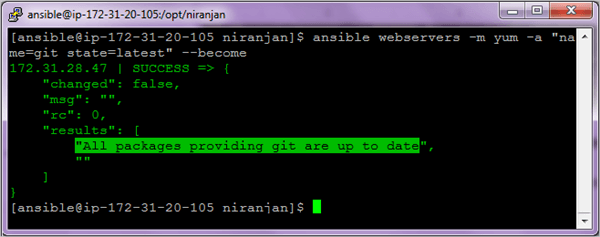
On the right-hand window, you can see if git is not installed it will give command not found and once installed it will show up the output.

In this command, **state=present** will check if the package is installed or not and if not installed it will install the latest version.

**Example 2:** Check if the package is installed & update it to the latest version.

**$ ansible webservers -m yum -a “name=git state=latest”**

In the above command, **state=latest**will update the package to the latest version only.

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/24.Check-if-package-is-installed-update-it-to-the-latest-version.png)

**Example 3:** Install Apache Webserver

**$ ansible webservers -m yum -a "name=httpd state=present" –become**

**Example 4:**Check if Maven is installed or not.

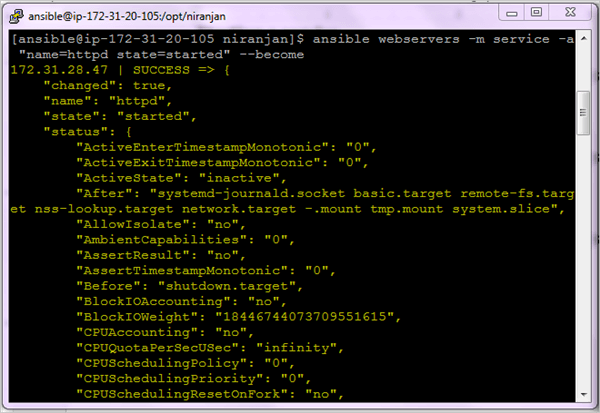
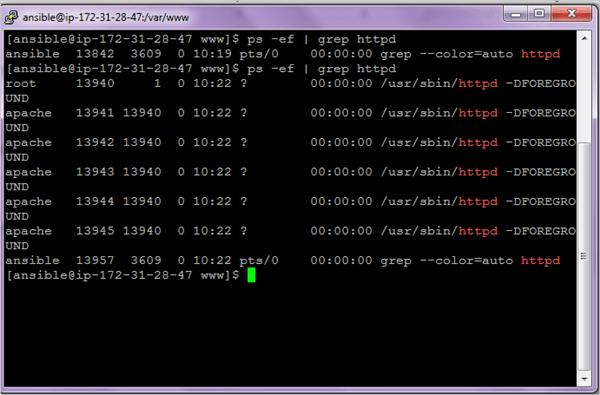
**$ ansible webservers -m yum -a "name=maven state=absent" –become**

#8) Managing Services Module

To manage services with ansible, we use a module **‘service’.**

**Starting a service**

**$ ansible webservers -m service -a “name=httpd state=started” --become**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/25.Managing-services-module1.png)[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/07/25.Managing-services-module2.png)

**Stopping a service**

**$ ansible webservers -m service -a “name=httpd state=stopped” --become**

**Restarting a service**

**$ ansible webservers -m service -a “name=httpd state=restarted --become**

Click [here](http://docs.ansible.com/ansible/latest/modules/list_of_all_modules.html" \t "_blank) to get the Complete Module list.