

How to Install and Use Ansible on Debian 10

Whenever I talk about configuration management tools, the most commonly mentioned name is Ansible. It is a cross-platform tool designed to handle system configurations while working with Linux, macOS and Windows operating systems.

Today, **TipsMake** will introduce the Ansible installation process on Debian 10.

Installing Ansible on Debian 10

To install Ansible on Debian 10, you need to do three simple steps:

Step 1: Update your Debian system 10

Before installing Ansible on Debian 10, you must update it using the command shown below:

```
sudo apt update
```

After updating your Debian 10 system, you should see information as shown in the following image on the Terminal:

```
kbuzadar@kbuzadar-pc:~$ sudo apt update
[sudo] password for kbuzadar:
Hit:1 http://deb.debian.org/debian buster InRelease
Hit:2 http://security.debian.org/debian-security buster/updates InRelease
Hit:3 http://deb.debian.org/debian buster-updates InRelease
Hit:4 http://repo.mysql.com/apt/debian buster InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
212 packages can be upgraded. Run 'apt list --upgradable' to see them.
kbuzadar@kbuzadar-pc:~$ █
```

Step 2: Install Ansible on a Debian 10 system

After the system update is finished, you can install Ansible on Debian 10 using the command shown below:

```
sudo apt install ansible
```

During the time this command is executed, you will see a message asking if you want to continue the installation on Terminal. You have to type **Y in** order for the installation to continue running smoothly.

When Ansible is successfully installed on a Debian 10 system, Terminal will generate some message on it similar to the one shown in the following figure:

```
Setting up python3-xmltodict (0.11.0-2) ...
Setting up python3-jinja2 (2.10-2) ...
Setting up python3-jmespath (0.9.4-1) ...
Setting up python3-requests-kerberos (0.11.0-2) ...
Setting up ieee-data (20180805.1) ...
Setting up python3-pyasn1 (0.4.2-3) ...
Setting up python3-selinux (2.8-1+b1) ...
Setting up python3-httplib2 (0.11.3-2) ...
Setting up python3-crypto (2.6.1-9+b1) ...
Setting up python3-asn1crypto (0.24.0-1) ...
Setting up python3-cffi-backend (1.12.2-1) ...
Setting up python3-bcrypt (3.1.6-1) ...
Setting up python3-libcloud (2.4.0-1) ...
Setting up python3-cryptography (2.6.1-3+deb10u2) ...
Setting up python3-netaddr (0.7.19-1) ...
Setting up python3-nacl (1.3.0-2) ...
Setting up python3-requests-ntlm (1.1.0-1) ...
Setting up python3-winrm (0.3.0-2) ...
Setting up python3-paramiko (2.4.2-0.1) ...
Setting up ansible (2.7.7+dfsg-1) ...
Processing triggers for man-db (2.8.5-2) ...
kbuzadar@kbuzadar-pc:~$
```

Step 3: Confirming Ansible installation on a Debian 10 system

Installing Ansible on a Debian 10 system is so simple that it should complete in the two steps above. However, you can still verify if it was successfully installed on your Debian 10 system. This can be done by checking the Ansible's version with the following command:

```
ansible --version
```

The version of Ansible installed on a Debian 10 system in the example is 2.7.7.

Edit the file host / **etc** / **ansible** / **hosts** to add the system you want to manage with Ansible.

```
sudo nano /etc/ansible/hosts
```

Please add the following:

```
[TestClient] node1 ansible_ssh_host=192.168.0.12
```

And save the file.

Use Ansible

First, you must configure the SSH Key for the client node because Ansible is using the SSH protocol to pass commands to the client system.

Use this command to generate SSH key and key-based authentication:

```
ssh-keygen
```

Output:

Generating public/private rsa key pair. Enter file in which to save the key (/root/.ssh/id_rsa):

The next step is to copy the newly created key to another system. Run this command:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.0.2
```

The IP 192.168.0.2 in the above command should be replaced with the IP address of the system you want to manage with Ansible.

Output:

```
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa"
```

Now that it's time to do the first test, log in to another machine using this SSH command:

```
ssh root@192.168.0.2
```

The login should now work without a password.

Test run Ansible

Installation is complete, now you can start testing Ansible

Run this command to test the connection:

```
ansible -m ping TestClient
```

Output:

```
node1 | SUCCESS => { "changed": false, "ping": "pong" }
```

In case you have defined multiple clients, you can check all connections with the following command:

```
ansible -m ping all
```

Now, it's time to run a command on the remote system and fetch the results. For this example, the df command will be used.

```
ansible -m shell -a 'df -h' TestClient
```

Output:

```
node1 | CHANGED | rc=0 >> Filesystem Size Used Avail Use% Mounted on udev 957M 0
```

The results show the use of the hard drive on the remote system. You can execute any Linux shell command like this using Ansible and also create a complete script to set up and maintain the server.

You finished reading the article "**How to Install and Use Ansible on Debian 10**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.