

## ANSIBLE INSTALLATION

### Pre-requisites

**Step 1:** Ensure python is installed on both the machines i.e Master and Slave, by default AWS Ubuntu images have python installed. You can skip this step if you are setting up ansible on AWS, on other machines install python using the following command.

```
$ sudo apt-get install python3
```

**Step 2:** Enable keyless SSH access between Ansible Master and Slave. To accomplish this, do the following:

#### On master

Pass the following command:

```
$ ssh-keygen
```

After this, keep the default values and keep on pressing enter, till you reach the following screen

```
ubuntu@ip-172-31-28-35:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Jtr2HaSp9S8eSKXxkHrUZhf6nS5PvHkIiTsP7Ylo8AQ ubuntu@ip-172-31-28-35
The key's randomart image is:
+---[RSA 2048]-----+
|
|   o
|  + o .
| * * o
| E o @ +
| .o S * +
| .o.= B = +
| .+o O + = .
| .o= O.+ .
| .o o.Bo.
|
+---[SHA256]-----+
ubuntu@ip-172-31-28-35:~/.ssh$
```

**Step 3:** Once done print id\_rsa.pub file using the command,


```
$ sudo cat ~/.ssh/id_rsa.pub
```

```
ubuntu@ip-172-31-28-35:~$ sudo cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCrOrpxyLW15uFTQEaxPF3sSAANfZ8bvm42H2Z/Ny/m
GIfoWNTweMWR9eOH8HhVlujhD0/n5ClKQZ9M2vM1lsLJV1TuIWtA1B+HeZPfOzZ7rKkYDw59LGDaw+C9
q0AUwiR2Mf1jY7+Iqd124RDFo5qZDgLOrNBqEy50lPeQc1Y9pAxU+7J8KZNJJixzo0p2gmEYB2j2ym6M
lciSADy5qfFKk8VZSh7ethGge3QbAvSnocdrkQObLDu6XYCha+Vns9h4mAMZx83lxpqW+S1h9EnpQs78
+GV5H+6CSMad+AxkVnQGYxuqSrYUHQ3UqrSgyj/tQwIAMcPBsvtLFRGDzA6v ubuntu@ip-172-31-28
-35
ubuntu@ip-172-31-28-35:~$
```

**Step 4:** Copy the output of this file, and paste it in the slave machine's authorized keys file. You can do that using the following command. Once inside the file, paste the output

```
$ sudo nano ~/.ssh/authorized_keys
```

```
GNU nano 2.9.3                               ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCUZqgffWtPie0gDGUP27VxqP56Hhu4xCBi7JNqJ30i70ZZ
$BsvtLFRGDzA6v ubuntu@ip-172-31-28-35
```



Insert the new entry in the second line of the file. Save and exit

Keyless access has now been configured between your ansible master and slave. You can verify by doing a SSH from your master to your slave, and you should be freely able to do so.

```
ubuntu@ip-172-31-24-91:~$ ssh ubuntu@34.217.207.113
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Jun  9 13:18:18 UTC 2020

System load:  0.0                       Processes:            89
Usage of /:   13.9% of 7.69GB           Users logged in:     1
Memory usage: 15%                      IP address for eth0: 172.31.28.35
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

Last login: Tue Jun  9 12:24:20 2020 from 34.220.210.174
ubuntu@ip-172-31-28-35:~$
```

With this, you have now successfully configured the pre-requisites.

## Installing Ansible on Master

Step 1: Pass the following commands, to install ansible on the master

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo apt-add-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```

Step 2: Next, configure the slave, by creating the hosts file in the master. To edit the file, pass the following command:

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

Once inside enter the following syntax to add a slave. In this syntax, **servers** is the group and **server1** is the slave machine's name, you can give any name of your choice.

```
[servers]
server1 ansible_host=<slave-ip-address>
```

```
GNU nano 2.9.3 /etc/ansible/hosts

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com

[servers]
server1 ansible_host=34.217.207.113
```

There will be some sample entries, you can ignore them and add the new values. With this, your ansible configuration is complete.

**Step 3:** Finally, you can test the ansible master slave connection by passing the following command:

```
$ ansible -m ping all
```

```
ubuntu@ip-172-31-24-91:~$ ansible -m ping all
server1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-24-91:~$ █
```

If everything, is configured correctly. You will get the above output.