ANSIBLE INSTALLATION WITH MASTER SLAVE CONNECTION

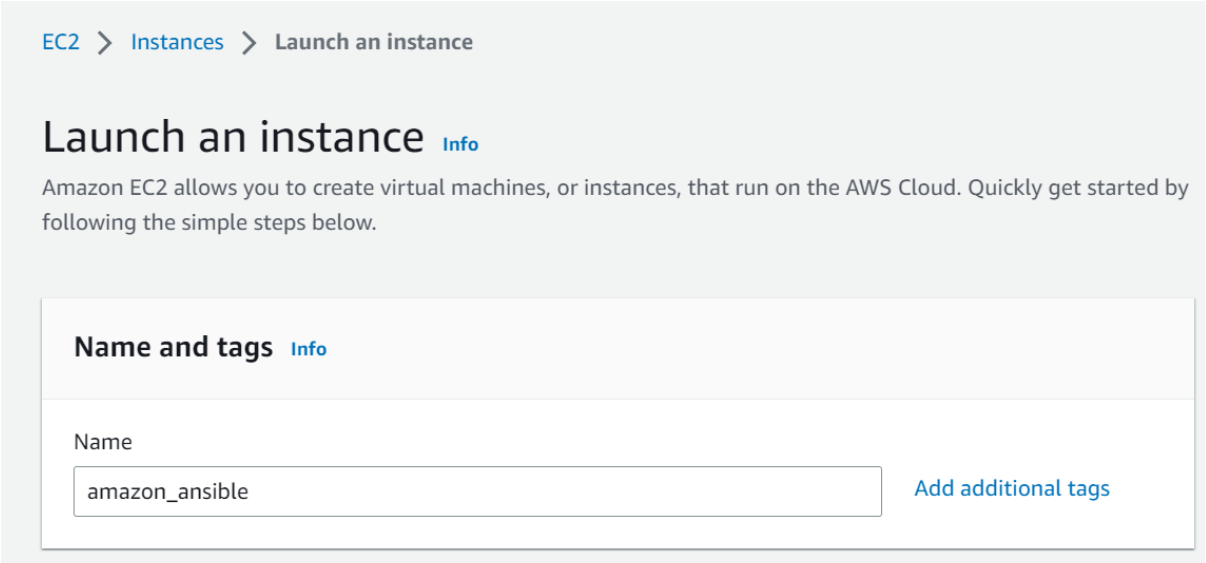
Ansible Devops

Kaushik Dey

Ansible installation with master slave

Step 1:

First create 3 ec2 instances . `Launch 3 EC2 Instances... 1 for Ansible Controller & 2 as Nodes`

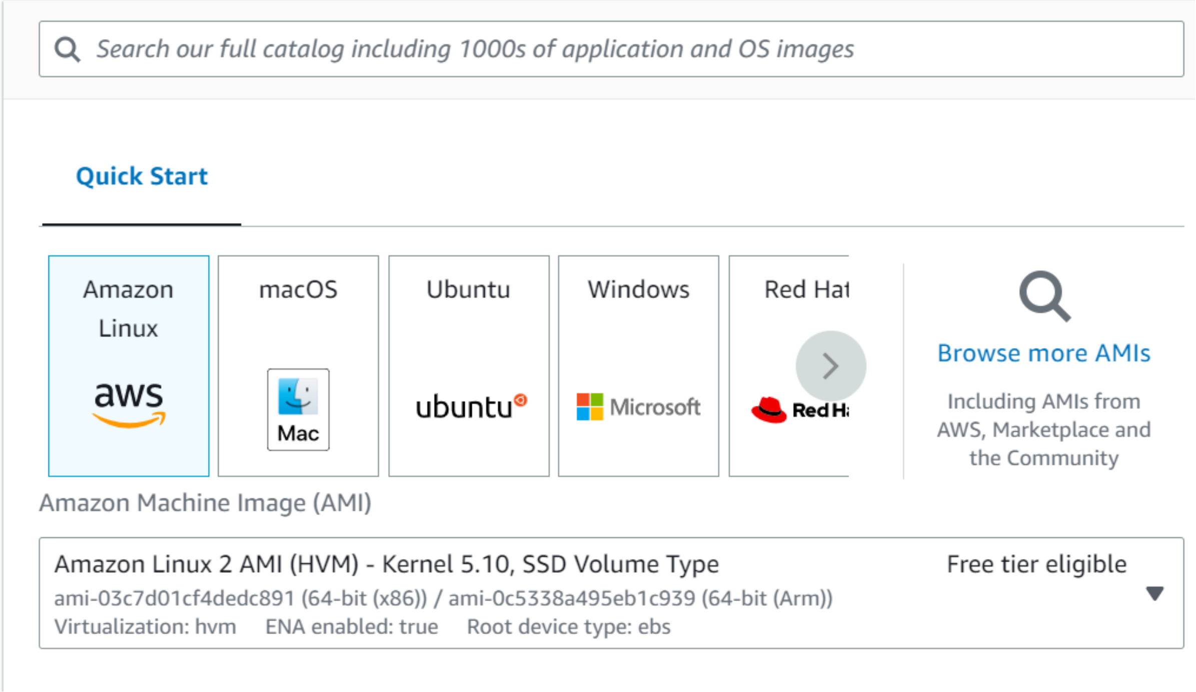


Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

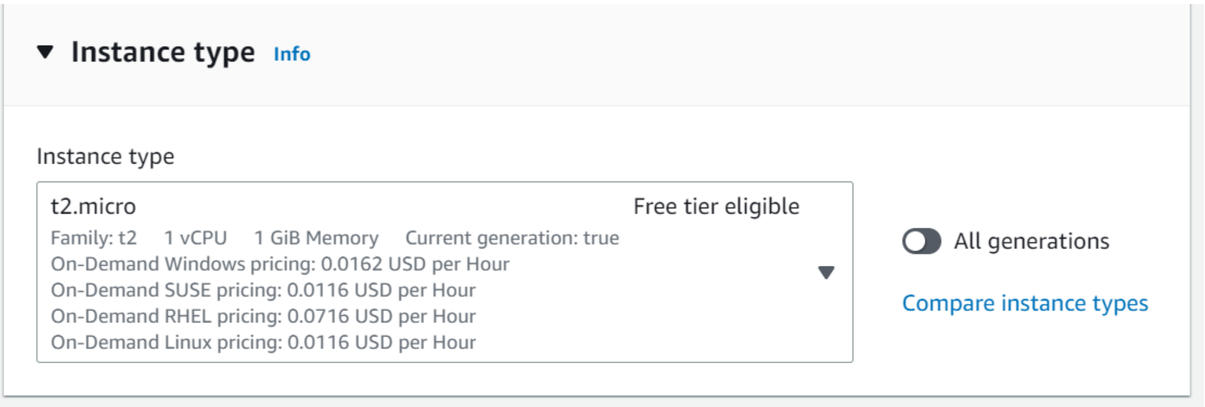
Name: amazon_ansible



Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

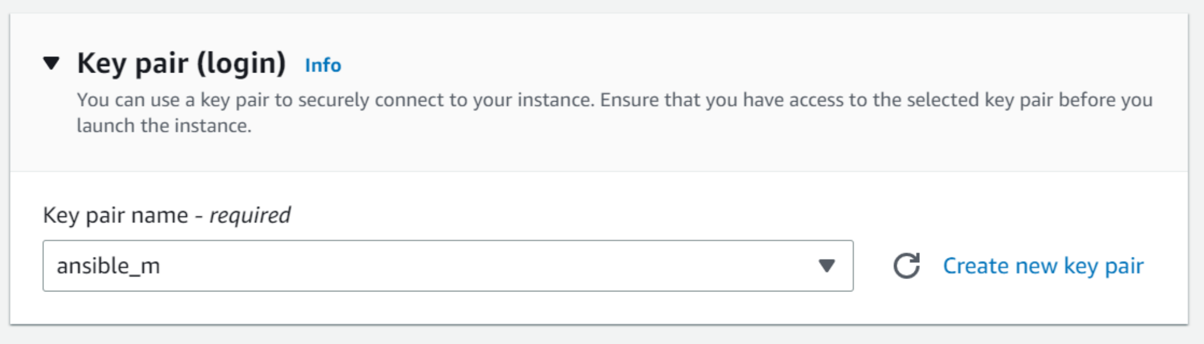
Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible
ami-03c7d01cf4dedc891 (64-bit (x86)) / ami-0c5338a495eb1c939 (64-bit (Arm))
Virtualization: hvm   ENA enabled: true   Root device type: ebs



▼ Instance type Info

Instance type

t2.micro — Free tier eligible
Family: t2   1 vCPU   1 GiB Memory   Current generation: true
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

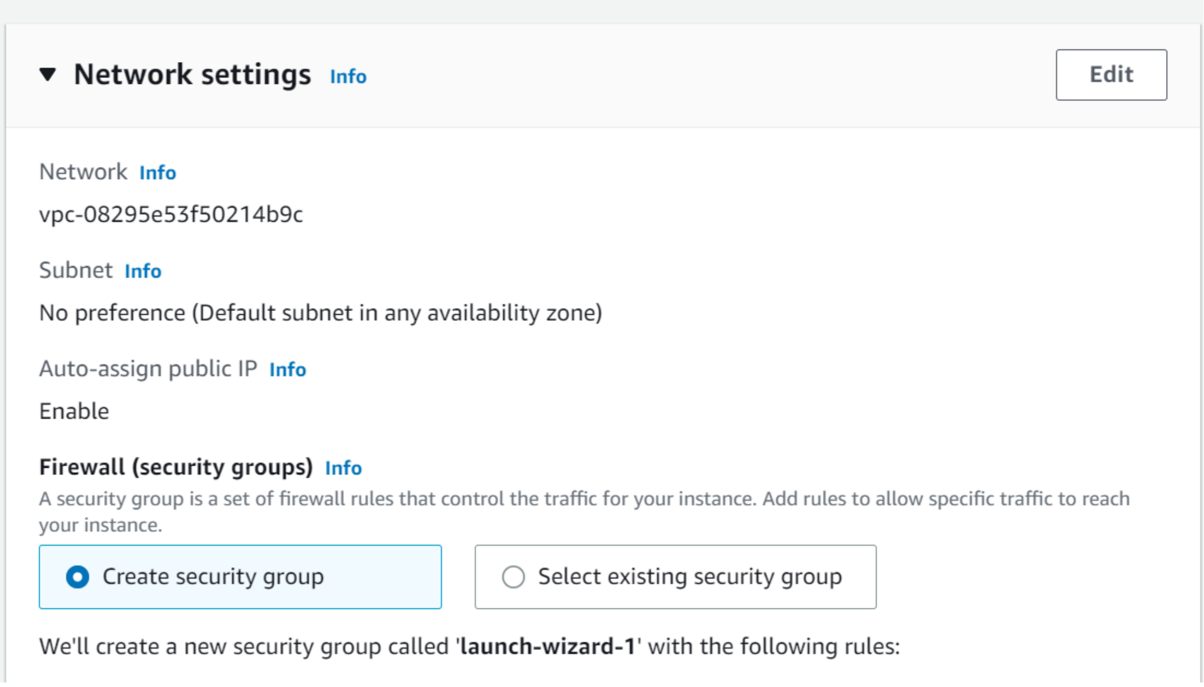


Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

ansible_m



Network settings Info

Network Info

vpc-08295e53f50214b9c

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

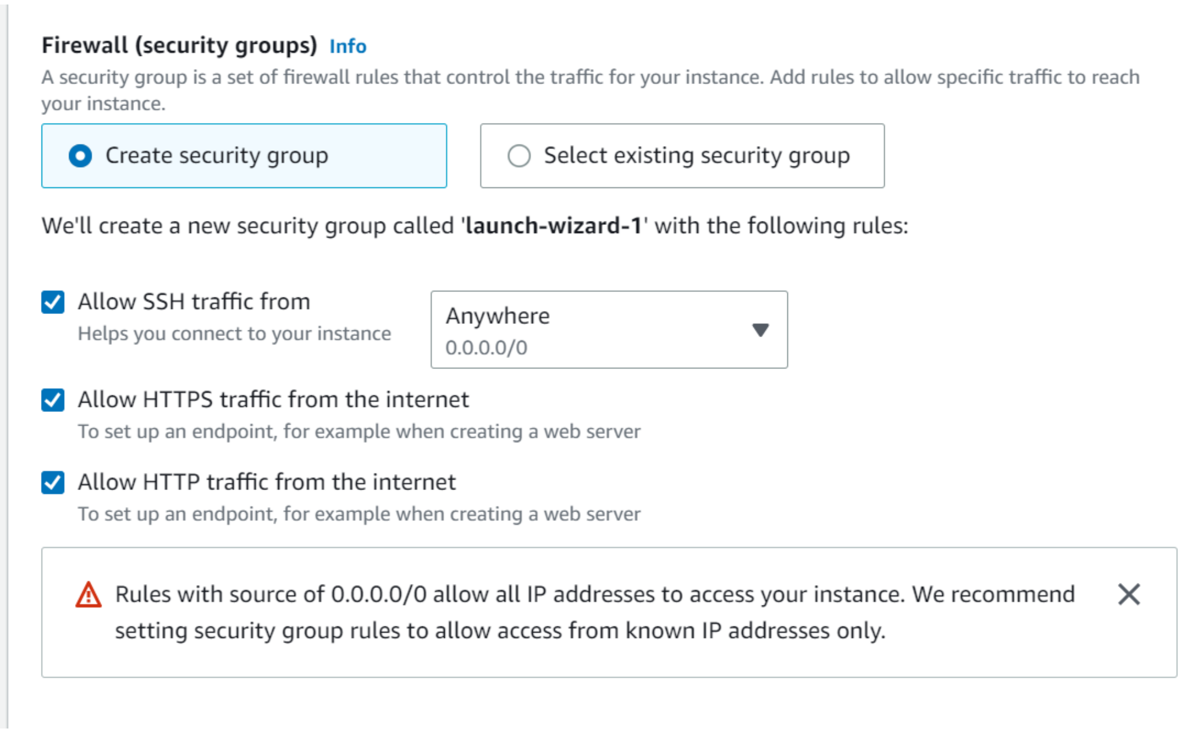
Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:



Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

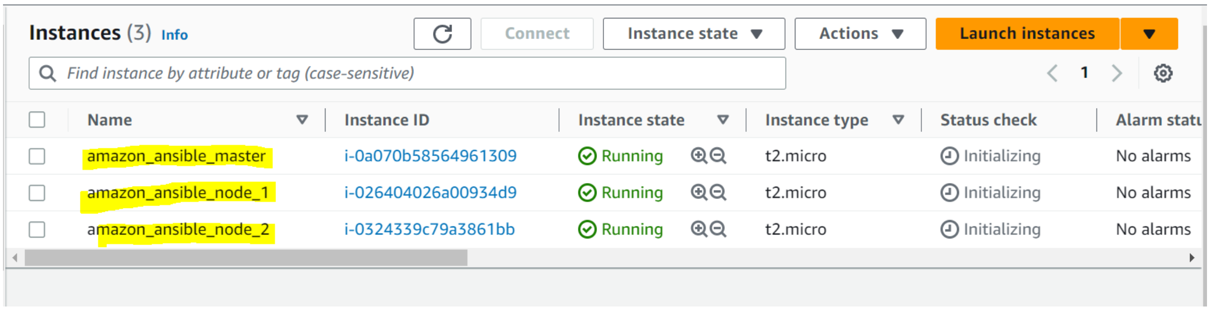
☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Ansible installation with master slave

Step 2 :



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm statu
□	amazon_ansible_master	i-0a070b58564961309	Running	t2.micro	Initializing	No alarms
□	amazon_ansible_node_1	i-026404026a00934d9	Running	t2.micro	Initializing	No alarms
□	amazon_ansible_node_2	i-0324339c79a3861bb	Running	t2.micro	Initializing	No alarms

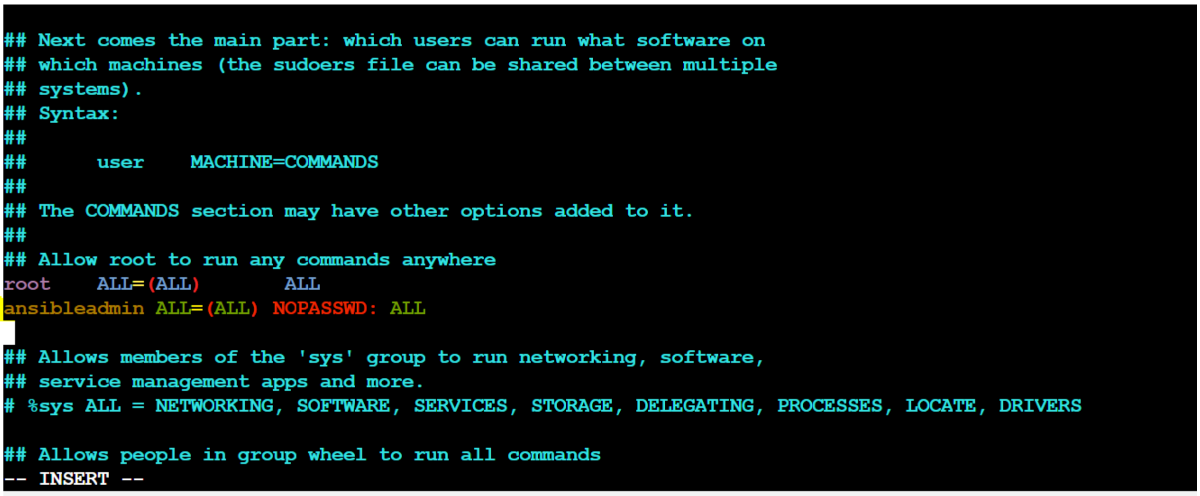
Then we have to connect amazon_ansible_master system.

Step 3 :

Password : Ishan@5000 (at least 8 charecters)

```
vi /etc/ssh/sshd_config .

#add the below mentioned line in the file and save it.
```



```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
ansibleadmin ALL=(ALL) NOPASSWD: ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
-- INSERT --
```

[root@ip-172-31-28-20 ~]# yum update -y

Loaded plugins: extras_suggestions, langpacks, priorities, update-motd

amzn2-core | 3.7 kB 00:00:00

No packages marked for update

[root@ip-172-31-28-20 ~]# useradd -m -d /home/ansibleadmin ansibleadmin

[root@ip-172-31-28-20 ~]# passwd ansibleadmin

Changing password for user ansibleadmin.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

[root@ip-172-31-28-20 ~]# vi /etc/ssh/sshd_config [root@ip-172-31-28-20 ~]# ^C

[root@ip-172-31-28-20 ~]# service sshd reload

Redirecting to /bin/systemctl reload sshd.service

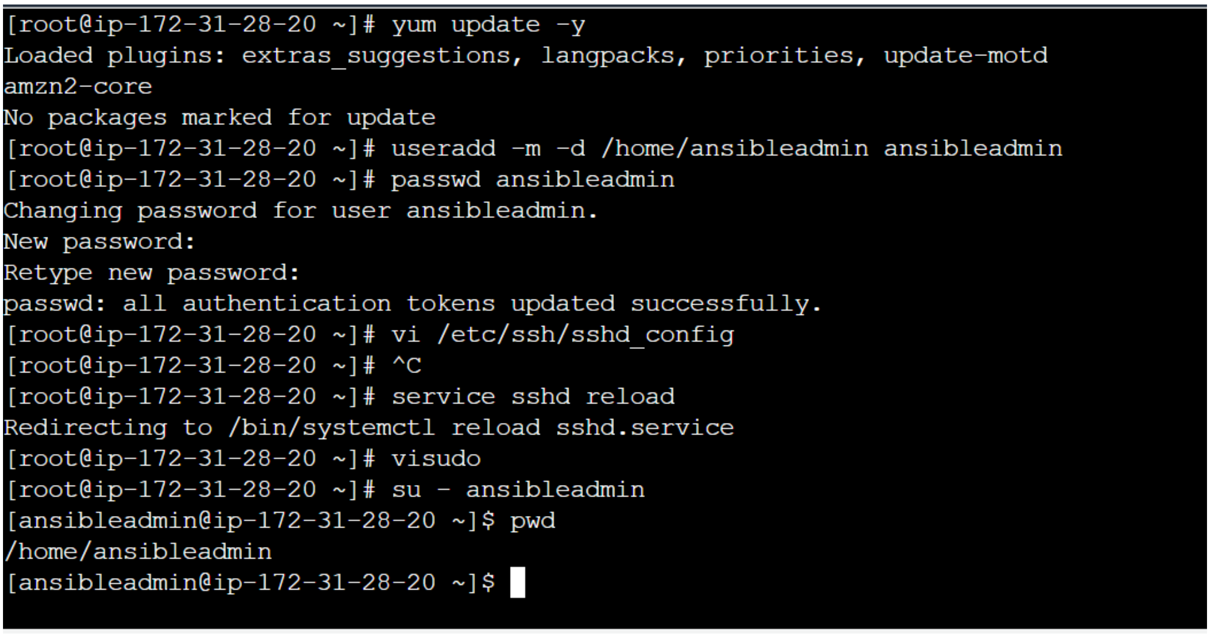
[root@ip-172-31-28-20 ~]# visudo

[root@ip-172-31-28-20 ~]# su - ansibleadmin

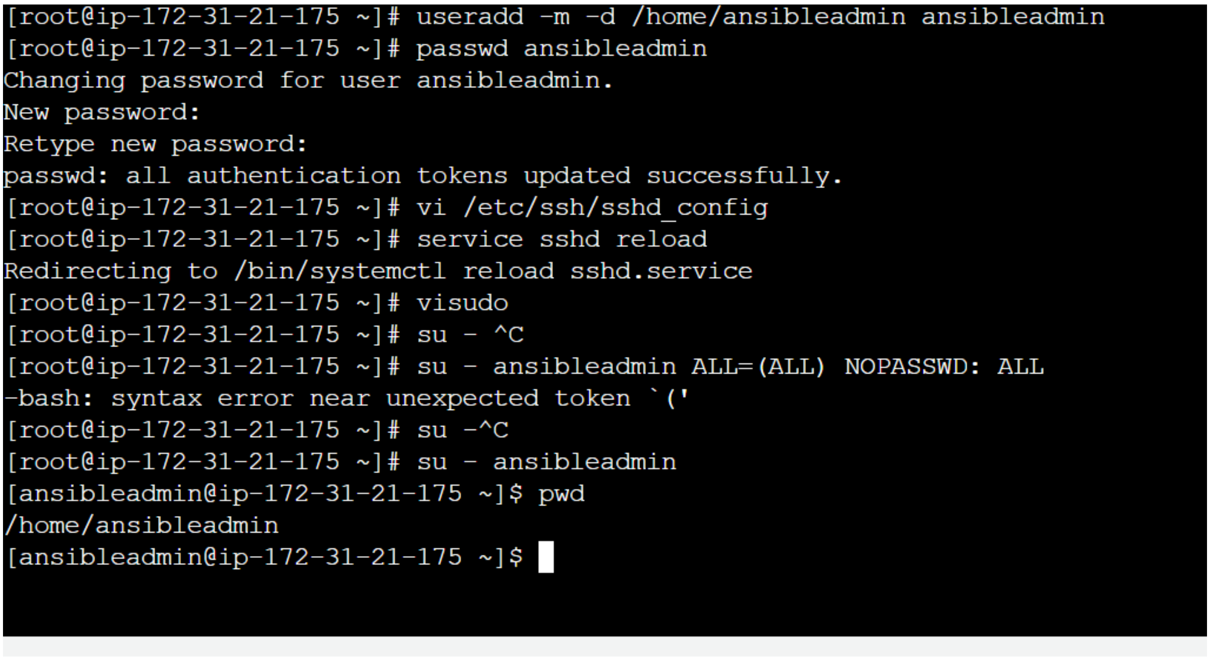
[ansibleadmin@ip-172-31-28-20 ~]$ pwd

/home/ansibleadmin

[ansibleadmin@ip-172-31-28-20 ~]$



```
[root@ip-172-31-28-20 ~]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
[root@ip-172-31-28-20 ~]# useradd -m -d /home/ansibleadmin ansibleadmin
[root@ip-172-31-28-20 ~]# passwd ansibleadmin
Changing password for user ansibleadmin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-28-20 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-28-20 ~]# ^C
[root@ip-172-31-28-20 ~]# service sshd reload
Redirecting to /bin/systemctl reload sshd.service
[root@ip-172-31-28-20 ~]# visudo
[root@ip-172-31-28-20 ~]# su - ansibleadmin
[ansibleadmin@ip-172-31-28-20 ~]$ pwd
/home/ansibleadmin
[ansibleadmin@ip-172-31-28-20 ~]$
```



```
[root@ip-172-31-21-175 ~]# useradd -m -d /home/ansibleadmin ansibleadmin
[root@ip-172-31-21-175 ~]# passwd ansibleadmin
Changing password for user ansibleadmin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-21-175 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-21-175 ~]# service sshd reload
Redirecting to /bin/systemctl reload sshd.service
[root@ip-172-31-21-175 ~]# visudo
[root@ip-172-31-21-175 ~]# su - ^C
[root@ip-172-31-21-175 ~]# su - ansibleadmin ALL=(ALL) NOPASSWD: ALL
-bash: syntax error near unexpected token `('
[root@ip-172-31-21-175 ~]# su -^C
[root@ip-172-31-21-175 ~]# su - ansibleadmin
[ansibleadmin@ip-172-31-21-175 ~]$ pwd
/home/ansibleadmin
[ansibleadmin@ip-172-31-21-175 ~]$
```

Now we must work with controller machine.

```
amazon-linux-extras install epel -y

amazon-linux-extras install ansible2 -y

[root@ip-172-31-81-101 ~]# cd /etc/ansible

[root@ip-172-31-81-101 ansible]# ansible --version

ansible 2.9.23

  config file = /etc/ansible/ansible.cfg

  configured module search path = [u'/root/.ansible/plugins/modules',
u'/usr/share/ansible/plugins/modules']

  ansible python module location = /usr/lib/python2.7/site-packages/ansible

  executable location = /bin/ansible

  python version = 2.7.18 (default, Feb 28 2023, 02:51:06) [GCC 7.3.1 20180712 (Red Hat 7.3.1-15)]

[root@ip-172-31-81-101 ansible]# ll

total 24

-rw-r--r-- 1 root root 19985 Jul  1  2021 ansible.cfg

-rw-r--r-- 1 root root  1016 Jul  1  2021 hosts

drwxr-xr-x 2 root root     6 Jul  1  2021 roles

[root@ip-172-31-81-101 ansible]#

[root@ip-172-31-16-135 ~]# cd /etc/ansible

[root@ip-172-31-16-135 ansible]# ll

total 24

-rw-r--r-- 1 root root 19985 Jul  1  2021 ansible.cfg

-rw-r--r-- 1 root root  1016 Jul  1  2021 hosts [different inventory file]

drwxr-xr-x 2 root root     6 Jul  1  2021 roles

[root@ip-172-31-16-135 ansible]#
```

Step 4 : As of now Ansible controller is ready along with 2 node (Target) system.

Now we must create admin in controller machine.

```
[root@ip-172-31-81-101 ansible]# useradd -m -d /home/devopsadmin devopsadmin

[root@ip-172-31-81-101 ansible]# chown -R devopsadmin:devopsadmin /etc/ansible

[root@ip-172-31-81-101 ansible]# ll

total 24
```

```
-rw-r--r-- 1 devopsadmin devopsadmin 19985 Jul  1  2021 ansible.cfg

-rw-r--r-- 1 devopsadmin devopsadmin  1016 Jul  1  2021 hosts

drwxr-xr-x 2 devopsadmin devopsadmin     6 Jul  1  2021 roles

[root@ip-172-31-81-101 ansible]# pwd

/etc/ansible

[root@ip-172-31-81-101 ansible]# su - devopsadmin

[devopsadmin@ip-172-31-81-101 ~]$

[devopsadmin@ip-172-31-81-101 ~]$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/devopsadmin/.ssh/id_rsa):

Created directory '/home/devopsadmin/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/devopsadmin/.ssh/id_rsa.

Your public key has been saved in /home/devopsadmin/.ssh/id_rsa.pub.

The key fingerprint is:

SHA256:oYuu8rPcDAMKbaDYTaBTJ9r1C16/iGymr/WJCCYx+1I devopsadmin@ip-172-31-81-101.ec2.internal

The key's randomart image is:

+---[RSA 2048]----+

|  + o            |

| = = .           |

|= . o o .        |

|o= + o + .       |

|B + o o S        |

|o*E. o o .       |

|=oo B o .        |

|=+.% o .         |

| +X** o          |

+----[SHA256]-----+
```

[devopsadmin@ip-172-31-81-101 ~]$ cd .ssh

[devopsadmin@ip-172-31-81-101 .ssh]$ ll

total 8

-rw------- 1 devopsadmin devopsadmin 1675 Apr 26 11:52 id_rsa

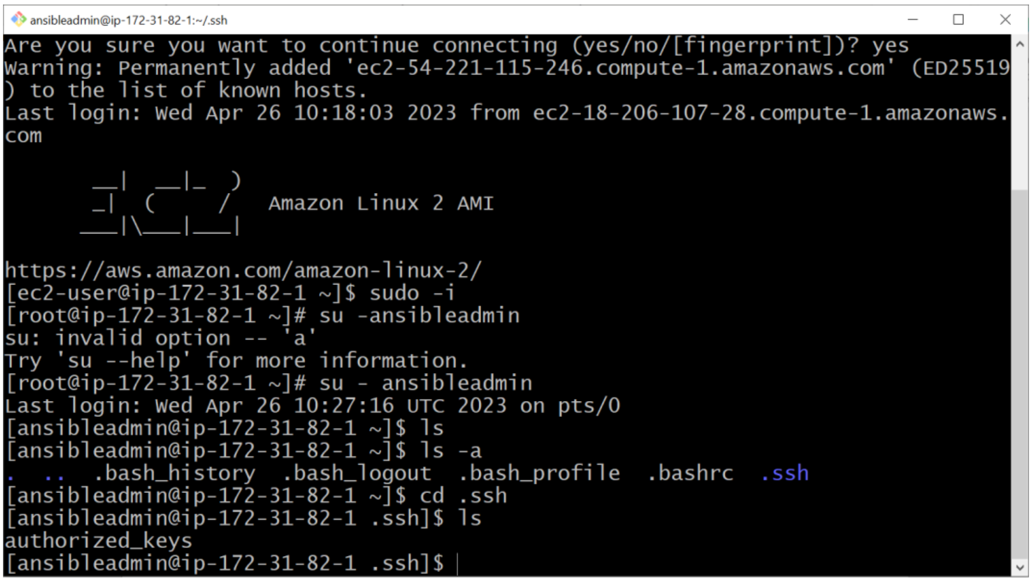
-rw-r--r-- 1 devopsadmin devopsadmin 423 Apr 26 11:52 id_rsa.pub

Step 5 : we have to copy controller file SSH key to the Node machine 1 and Node Machine 2.

ssh-copy-id ansibleadmin@ 172.31.93.37

ssh-copy-id ansibleadmin@ 172.31.82.1

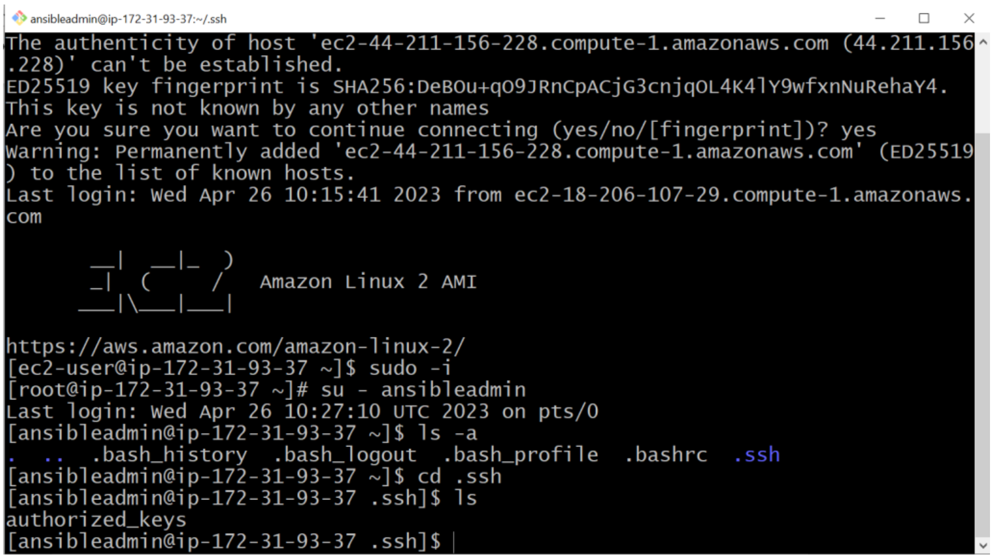
so, with this command I have transferred my controller machine key to node1 and node , so successfully transferred the key to node 1 and node 2. The screenshot is given below.



```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-221-115-246.compute-1.amazonaws.com' (ED25519
) to the list of known hosts.
Last login: Wed Apr 26 10:18:03 2023 from ec2-18-206-107-28.compute-1.amazonaws.
com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-82-1 ~]$ sudo -i
[root@ip-172-31-82-1 ~]# su -ansibleadmin
su: invalid option -- 'a'
Try 'su --help' for more information.
[root@ip-172-31-82-1 ~]# su - ansibleadmin
Last login: Wed Apr 26 10:27:16 UTC 2023 on pts/0
[ansibleadmin@ip-172-31-82-1 ~]$ ls
[ansibleadmin@ip-172-31-82-1 ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .ssh
[ansibleadmin@ip-172-31-82-1 ~]$ cd .ssh
[ansibleadmin@ip-172-31-82-1 .ssh]$ ls
authorized_keys
[ansibleadmin@ip-172-31-82-1 .ssh]$ 
```



```
The authenticity of host 'ec2-44-211-156-228.compute-1.amazonaws.com (44.211.156
.228)' can't be established.
ED25519 key fingerprint is SHA256:DeBOu+qO9JRnCpACjG3cnjqOL4K4lY9wfxnNuRehaY4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-211-156-228.compute-1.amazonaws.com' (ED25519
) to the list of known hosts.
Last login: Wed Apr 26 10:15:41 2023 from ec2-18-206-107-29.compute-1.amazonaws.
com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-93-37 ~]$ sudo -i
[root@ip-172-31-93-37 ~]# su - ansibleadmin
Last login: Wed Apr 26 10:27:10 UTC 2023 on pts/0
[ansibleadmin@ip-172-31-93-37 ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .ssh
[ansibleadmin@ip-172-31-93-37 ~]$ cd .ssh
[ansibleadmin@ip-172-31-93-37 .ssh]$ ls
authorized_keys
[ansibleadmin@ip-172-31-93-37 .ssh]$ 
```

Ansible installation with master slave

Remote server Authentication!!!!

Jenkins_master and slave ==> are owned by Devops Team.

Ansible Controller :: owned by the Devops Team/Infra Team

Target Machines: dev,test,hosted servers, mail server.

Please Note: As soon as I run the command ssh-copy-id it will create .ssh folder in target machine and also it creates known hosts folder in its controller.

[devopsadmin@ip-172-31-81-101 .ssh]$ cat known_hosts

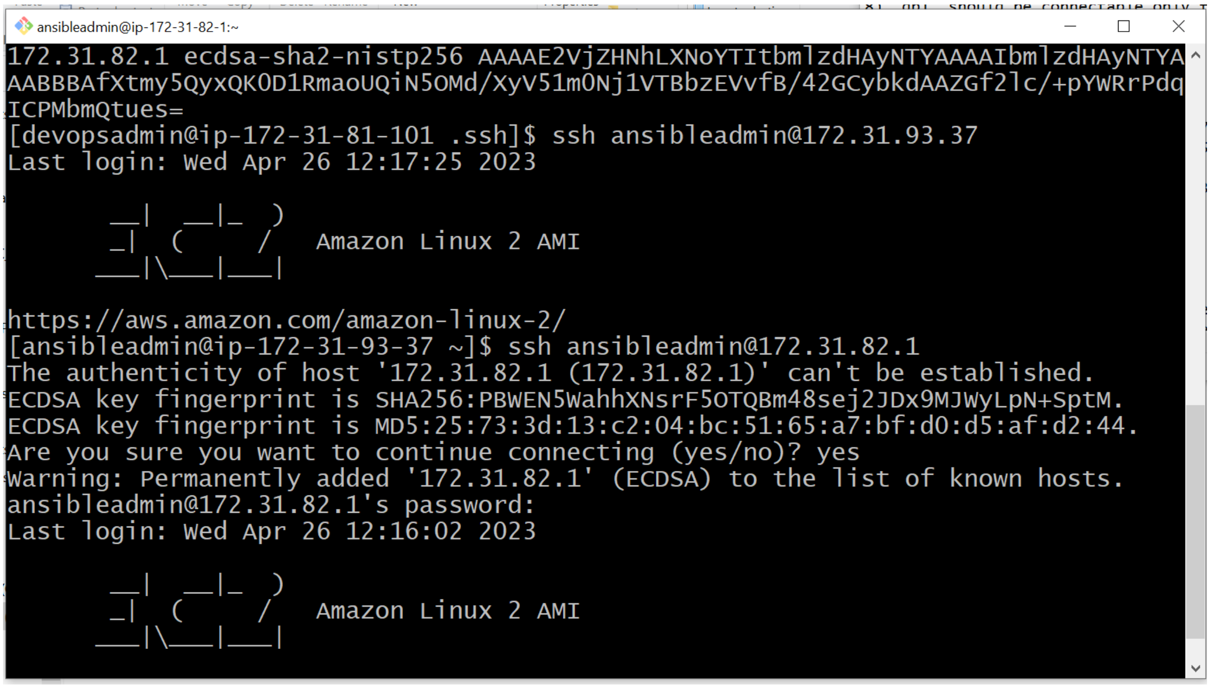
172.31.93.37 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBK2pas6wJBG+1mcLoebYsDK5MAIjNEQHiaVux4IHz1T3U6KeJvtadzC7SRDPPThSGGzUi336miG+CAkA2vs2Ll4=

172.31.82.1 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAfXtmy5QyxQK0D1RmaoUQiN5OMd/XyV51m0Nj1VTBbzEVvfB/42GCybkdAAZGf2lc/+pYWRrPdqICPMbmQtues=

Now from our Controller system we can enter inside our Node Target Machine. The code is there.

ssh ansibleadmin@172.31.93.37

ssh ansibleadmin@172.31.82.1

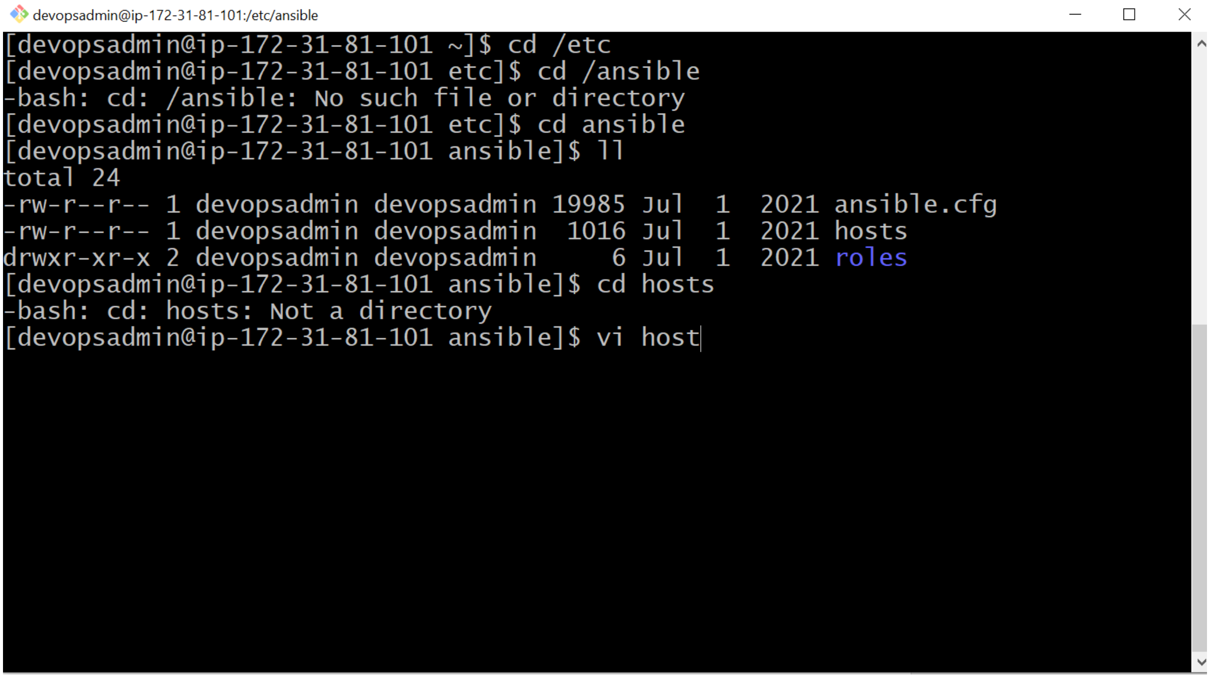


```
172.31.82.1 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYA
AABBBAfXtmy5QyxQK0D1RmaoUQiN5OMd/XyV51m0Nj1VTBbzEVvfB/42GCybkdAAZGf2lc/+pYWRrPdq
ICPMbmQtues=
[devopsadmin@ip-172-31-81-101 .ssh]$ ssh ansibleadmin@172.31.93.37
Last login: Wed Apr 26 12:17:25 2023

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ansibleadmin@ip-172-31-93-37 ~]$ ssh ansibleadmin@172.31.82.1
The authenticity of host '172.31.82.1 (172.31.82.1)' can't be established.
ECDSA key fingerprint is SHA256:PBWEN5WahhXNsrF5OTQBm48sej2JDx9MJWyLpN+SptM.
ECDSA key fingerprint is MD5:25:73:3d:13:c2:04:bc:51:65:a7:bf:d0:d5:af:d2:44.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.31.82.1' (ECDSA) to the list of known hosts.
ansibleadmin@172.31.82.1's password:
Last login: Wed Apr 26 12:16:02 2023

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|
```



```
[devopsadmin@ip-172-31-81-101 ~]$ cd /etc
[devopsadmin@ip-172-31-81-101 etc]$ cd /ansible
-bash: cd: /ansible: No such file or directory
[devopsadmin@ip-172-31-81-101 etc]$ cd ansible
[devopsadmin@ip-172-31-81-101 ansible]$ ll
total 24
-rw-r--r-- 1 devopsadmin devopsadmin 19985 Jul  1  2021 ansible.cfg
-rw-r--r-- 1 devopsadmin devopsadmin  1016 Jul  1  2021 hosts
drwxr-xr-x 2 devopsadmin devopsadmin     6 Jul  1  2021 roles
[devopsadmin@ip-172-31-81-101 ansible]$ cd hosts
-bash: cd: hosts: Not a directory
[devopsadmin@ip-172-31-81-101 ansible]$ vi host
```

Hosts is default inventory file.

Step 9 : Now we **ping the server and get the response , the code is**

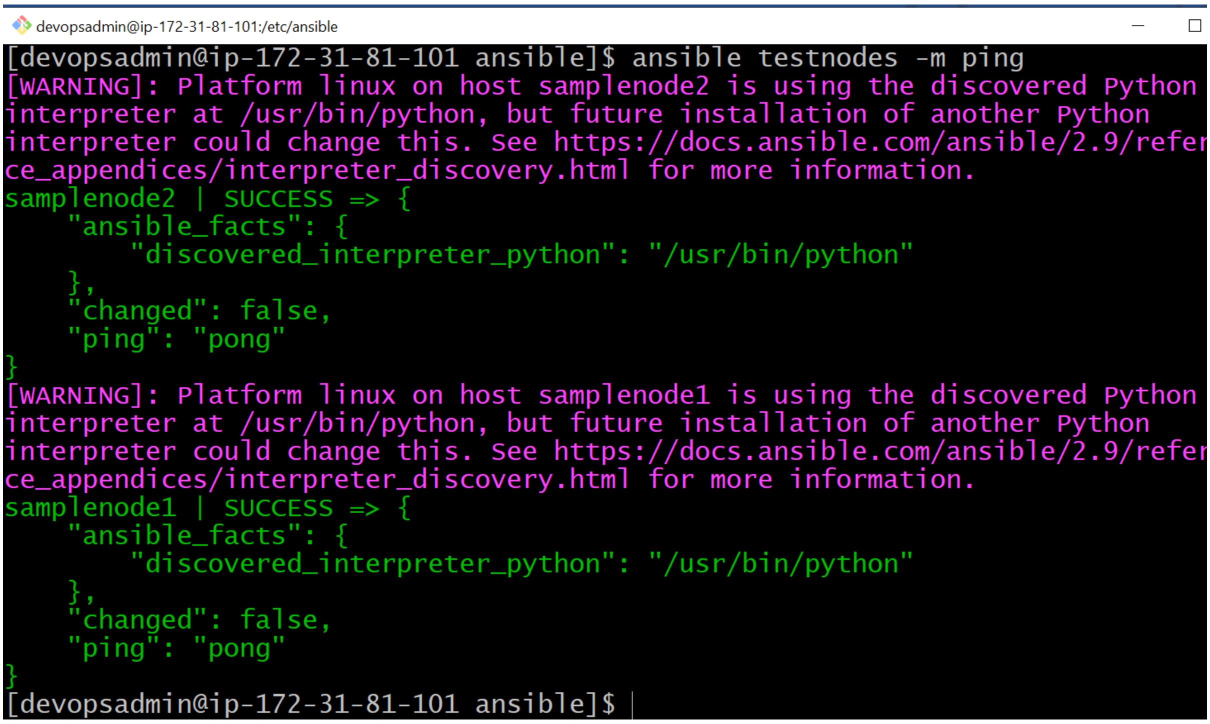
[testnodes]

samplenode1 ansible_ssh_host=172.31.93.37 ansible_ssh_user=ansibleadmin

samplenode2 ansible_ssh_host=172.31.82.1 ansible_ssh_user=ansibleadmin

ansible <hosts_name> -m <module_name> -i <inventory_file>

ansible testnodes -m ping



```
[devopsadmin@ip-172-31-81-101 ansible]$ ansible testnodes -m ping
[WARNING]: Platform linux on host samplenode2 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/refer
ce_appendices/interpreter_discovery.html for more information.
samplenode2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host samplenode1 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/refer
ce_appendices/interpreter_discovery.html for more information.
samplenode1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[devopsadmin@ip-172-31-81-101 ansible]$
```

Step 10 :

Ansible adhoc commands.

```
[devopsadmin@ip-172-31-81-101 ansible]$ ansible samplenode1 -m
ping
[WARNING]: Platform linux on host samplenode1 is using the
discovered Python
interpreter at /usr/bin/python, but future installation of
another Python
interpreter could change this. See
https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
samplenode1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[devopsadmin@ip-172-31-81-101 ansible]$ ansible samplenode2 -m
ping
[WARNING]: Platform linux on host samplenode2 is using the
discovered Python
interpreter at /usr/bin/python, but future installation of
another Python
interpreter could change this. See
https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
samplenode2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[devopsadmin@ip-172-31-81-101 ansible]$ ansible all -m ping
[devopsadmin@ip-172-31-81-101 ansible]$ ansible testNodes -m
ping
[devopsadmin@ip-172-31-81-101 ansible]$ ansible samplenode1 -m
ping
[devopsadmin@ip-172-31-81-101 ansible]$ ansible samplenode2 -m
ping
```

Now where we store the inventory parameters

```
[devopsadmin@ip-172-31-81-101 ansible]$ cat ansible.cfg
[devopsadmin@ip-172-31-81-101 ansible]$ cat ansible.cfg
To create inventory file
[devopsadmin@ip-172-31-81-101 ansible]$ echo "samplenode1
ansible_ssh_host=172.31.93.37 ansible_ssh_user=ansibleadmin"
>>inventory1
[devopsadmin@ip-172-31-81-101 ansible]$ ls
ansible.cfg  hosts  inventory1  roles
[devopsadmin@ip-172-31-81-101 ansible]$
```

To create sample Inventory file,

```
[devopsadmin@ip-172-31-81-101 ansible]$
echo "samplenode1 ansible_ssh_host=172.31.93.37
ansible_ssh_user=ansibleadmin" >>inventory1

[devopsadmin@ip-172-31-81-101 ansible]$ansible samplenode1 -m
ping -i inventory1
```

Ansible Module Commands

In agent system .ansible file will be created.

```
[ansibleadmin@ip-172-31-93-37 ~]$ ls -a
.  ..  .ansible  .bash_history  .bash_logout  .bash_profile
.bashrc  .ssh
[ansibleadmin@ip-172-31-93-37 ~]$ cd .ansible/
[ansibleadmin@ip-172-31-93-37 .ansible]$ ls
tmp
[ansibleadmin@ip-172-31-93-37 .ansible]$ tmp
-bash: tmp: command not found
[ansibleadmin@ip-172-31-93-37 .ansible]$ cd tmp/
[ansibleadmin@ip-172-31-93-37 tmp]$ ls
[ansibleadmin@ip-172-31-93-37 tmp]$ pwd
/home/ansibleadmin/.ansible/tmp
[ansibleadmin@ip-172-31-93-37 tmp]$
```

To collect the history of sample machine we should run this command.

```
[ansibleadmin@ip-172-31-93-37 tmp]$ ansible samplenode1 -m
setup
[devopsadmin@ip-172-31-81-101 ansible]$ ansible samplenode1 -m setup
-a "filter=ansible_mounts"
```

##Transfer a file from Ansible Controller to Nodes using copy Module

```
[devopsadmin@ip-172-31-81-101 ansible] $ echo “record1” >> file1.txt
[devopsadmin@ip-172-31-81-101 ansible] ls

[devopsadmin@ip-172-31-81-101 ansible]
ansible samplenode1 -m copy -a "src=/etc/ansible/file1.txt
dest=/home/ansibleadmin"
```

in our target server we can see that file1.txt is present

```
[ansibleadmin@ip-172-31-93-37 ~]$ ls
```

Please note that if you want to copy this files inside 100 servers what you will do, instead of file name you can add server name.

If we change the content in the same file then 2 records are being maintained there.

So, we should backup it.

```
[devopsadmin@ip-172-31-81-101 ansible]
ansible samplenode1 -m copy -a "src=/etc/ansible/file1.txt
dest=/home/ansibleadmin backup=yes"
```

##Transfer a file from Ansible Nodes to Ansible Controller using fetch Module

```
[devopsadmin@ip-172-31-81-101 ansible]
ansible samplenode1 -m fetch -a "src=/home/ansibleadmin/kaushik.txt
dest=/home/devopsadmin"
```

```
please note kaushik.txt file will be created in target node & we are
trying to copy it from target to controller machine.
```