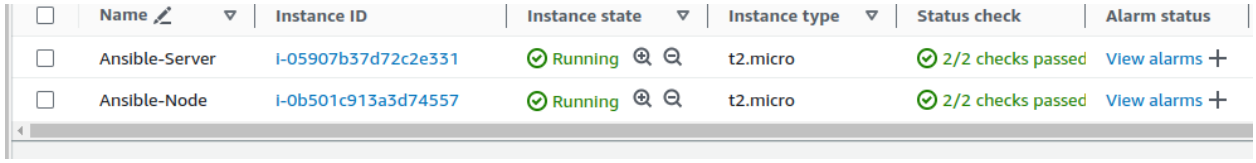
Experiment 11

Configure Ansible Setup In Linux

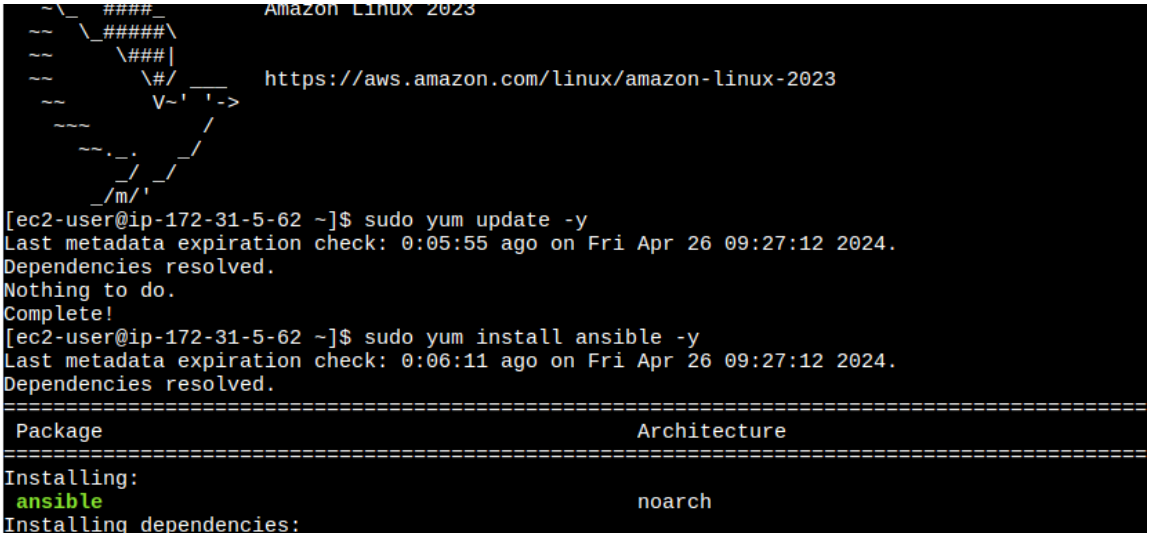
Steps

1. Create two amazon Linux t2.micro instance - Ansible Server & Node

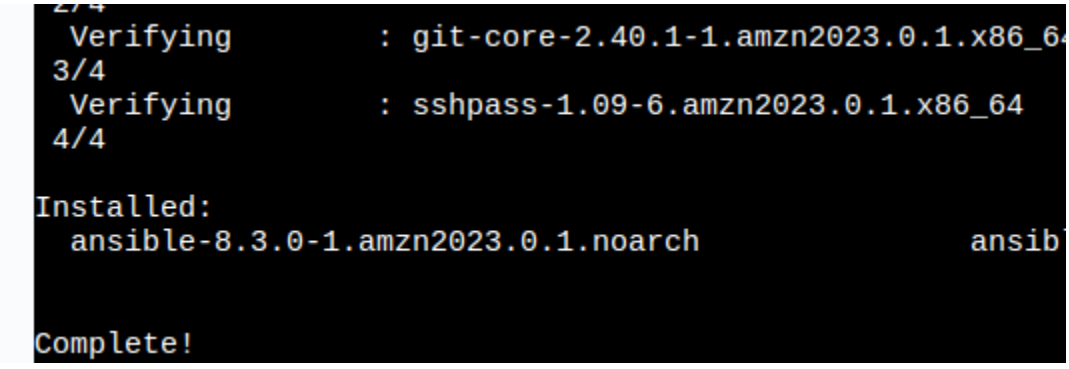


Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Ansible-Server	i-05907b37d72c2e331	Running	t2.micro	2/2 checks passed	View alarms +
Ansible-Node	i-0b501c913a3d74557	Running	t2.micro	2/2 checks passed	View alarms +

2. Install Ansible on Ansible server



```
  ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-5-62 ~]$ sudo yum update -y
Last metadata expiration check: 0:05:55 ago on Fri Apr 26 09:27:12 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-5-62 ~]$ sudo yum install ansible -y
Last metadata expiration check: 0:06:11 ago on Fri Apr 26 09:27:12 2024.
Dependencies resolved.
================================================================================
 Package                                    Architecture
================================================================================
Installing:
 ansible                                    noarch
Installing dependencies:
```



```
  Verifying        : git-core-2.40.1-1.amzn2023.0.1.x86_64
 3/4
  Verifying        : sshpass-1.09-6.amzn2023.0.1.x86_64
 4/4

Installed:
  ansible-8.3.0-1.amzn2023.0.1.noarch                ansible

Complete!
```

```
[ec2-user@ip-172-31-5-62 ~]$ ansible --version
ansible [core 2.15.3]
  config file = None
  configured module search path = ['/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /home/ec2-user/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.16 (main, Sep  8 2023, 00:00:00) [GCC 11.4.1 20230605 (Red Hat 11.4.1-2)] (/usr/bin/python3.9)
  jinja version = 3.1.2
  libyaml = True
[ec2-user@ip-172-31-5-62 ~]$
```

3. Add Private IP of node to the ansible server's inventory file

```
[root@ip-172-31-11-127 ec2-user]# ansible --version
ansible 2.9.27
  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, Dec 18 2023, 22:08:43) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[root@ip-172-31-11-127 ec2-user]# vi /etc/ansible/hosts
[root@ip-172-31-11-127 ec2-user]#
```

```
[upes]
172.31.5.83
```

4. Create super user in both the machines

```
[root@ip-172-31-11-127 ansible]# passwd ansible
Changing password for user ansible.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-11-127 ansible]#
```

```
[root@ip-172-31-5-83 ansible]# passwd ansiblenode
Changing password for user ansiblenode.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-5-83 ansible]#
```

5. Give sudo user permissions to both users

```
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
ansible ALL=(ALL)     NOPASSWD: ALL

## Allows members of the 'sys' group to run networking, so
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELE
```

```
## The COMMANDS section may have other options added to it
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
ansiblenode ALL=(ALL)     NOPASSWD: ALL

## Allows members of the 'sys' group to run networking, so
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELE
```

6. Edit the sshd_config file in the node server

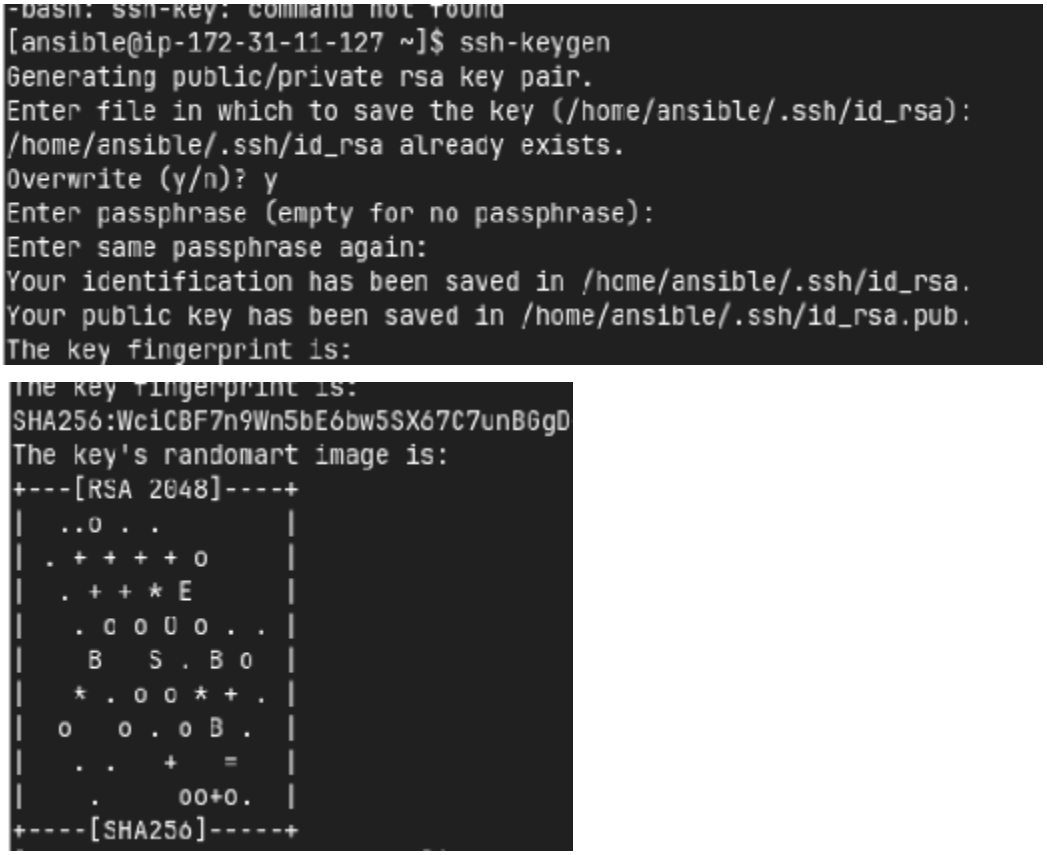
```
[root@ip-172-31-5-83 ec2-user]# visudo
[root@ip-172-31-5-83 ec2-user]# nano /etc/ssh/sshd_config
[root@ip-172-31-5-83 ec2-user]#
```

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
```

```
# Don't read the user's ~/.rhos
#IgnoreRhosts yes

# To disable tunneled clear tex
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

```
[root@ip-172-31-5-83 ec2-user]# visudo
[root@ip-172-31-5-83 ec2-user]# nano /etc/ssh/sshd_config
[root@ip-172-31-5-83 ec2-user]# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[root@ip-172-31-5-83 ec2-user]#
```

7. Generate key pair in the ansible server and copy the key to node server



8. Connect to node server from ansible server