ssh -i "mumbai.pem" [ec2-user@ec2-13-233-125-93.ap-south-1.compute.amazonaws.com](mailto:ec2-user@ec2-13-233-125-93.ap-south-1.compute.amazonaws.com)

13.233.125.93

172.31.38.203

**#client**

ssh -i "Aj-mumbai.pem" [ec2-user@ec2-13-126-194-204.ap-south-1.compute.amazonaws.com](mailto:ec2-user@ec2-13-126-194-204.ap-south-1.compute.amazonaws.com)

13.126.194.204

172.31.39.102

**Installating Ansible  on Amazon Linux 2**

Here we need two server machines for this lab. We need to configure master and client on each server machine. Here we will accomplish our goals in following steps,

Connect to ec2-instance using putty or terminal and perform yum update to update local repository.

$ sudo yum update -y

Change hostname on root user on master and client1 nodes.

$ sudo hostnamectl set-hostname master (on master)

$ sudo hostnamectl set-hostname client1 (on client1)

Next, Reboot the machine and login again to view the hostname changes.

$ sudo reboot

**1. Install Ansible using EPEL Repository**

Download epel repository on the amazon linux 2 instance as follows,

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

Next, Install epel repository using yum.

$ sudo yum install epel-release-latest-7.noarch.rpm

Update epel repository as follows,

$ sudo yum update -y

Now install all individual packages inside the repository along with ansible.

$ sudo yum install python python-devel python-pip openssl ansible -y

**2. Install Ansible using amazon-linux-extras Repository**

Ansible package can be installed on amazon linux using amazon provided packages.

$ sudo amazon-linux-extras install ansible2

**Check Ansible version**

To verify whether Ansible is installed on your machine, you can verify it as follows,

$ 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, Aug 27 2020, 21:22:52) [GCC 7.3.1 20180712 (Red Hat 7.3.1-9)]

**Configure Ansible on Amazon Linux-2**

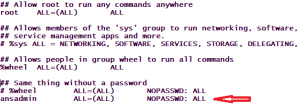
Firstly, create new users on master & client machines.

$ sudo useradd ansadmin

$ sudo passwd ansadmin

Grant admin access to created user for master & client machines. You need login as root to make changes here. We have added **ansadmin** user in sudoers file and enable authentication without password each time.

$ sudo visudo



Thereafter, Allow password authentication to yes so that ldap users can login using username & password on master & client machines

$ sudo vi /etc/ssh/sshd\_config

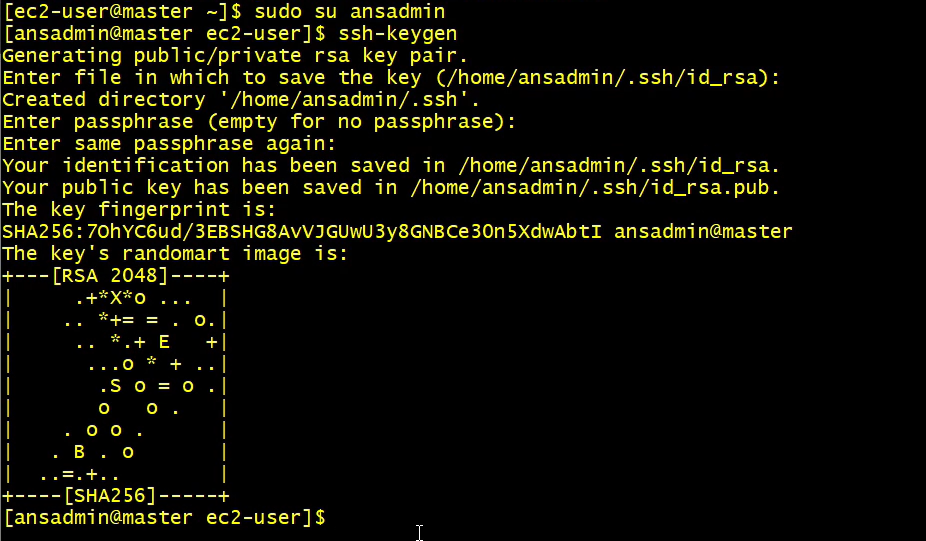
sshd_config file

Uncomment above highlighted line and enable password authentication to YES. Restart sshd service.

$ sudo service sshd restart

Login as ansadmin user and generate public and private keys on Master machine.

$ ssh-keygen



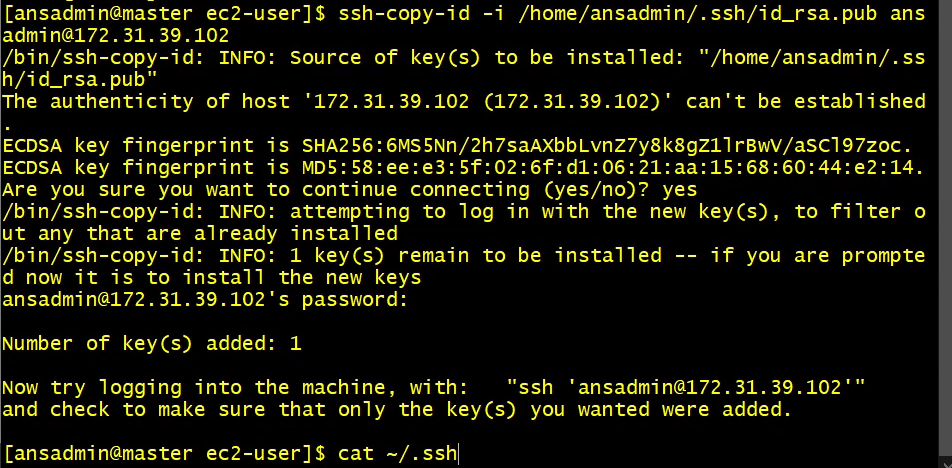
Now you need to copy ssh keys from master to client machine. First check the private ip address of the client node as follows,

$ ifconfig -a

ifconfig on aws ec2 instance

Copy the public key file from master node to client.

$ ssh-copy-id -i /home/ansadmin/.ssh/id\_rsa.pub ansadmin@172.31.17.1



To verify this, try to login to client machine from master using ssh method.

$ ssh ansadmin@172.31.17.2

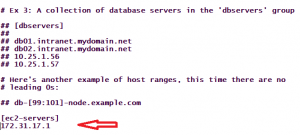
It will ask password for the first time. Next time it won’t ask password. Here, 172.31.17.2 is the private ip address of client node.

**Managing inventory file on Master**

A default hosts file is created while ansible installation. To manage your infrastructure you need to make entries of your available server machines in this hosts file.

$ sudo  vi /etc/ansible/hosts

Insert client machines ip address into the inventory file:



Simply we put client machine ip address here. You can create custom inventory file if you do not want to use default hosts file.

**Perform Ping test from Master to Client machines**

Ansible contains various modules to manage IT infrastructure.  Here, we will use ping module to check connection status between master and client node as follows,

$ ansible -m ping ec2-servers

ansible ping test 

Finally, client machine can be accessed from master node through ansible.