# **Ansible lab set**

* Create two instances with amazon linux kernel 6.12(give names ansible-server and host-server)
* Connect two instances

Execute following commands on ansible server 👇

sudo yum update -y

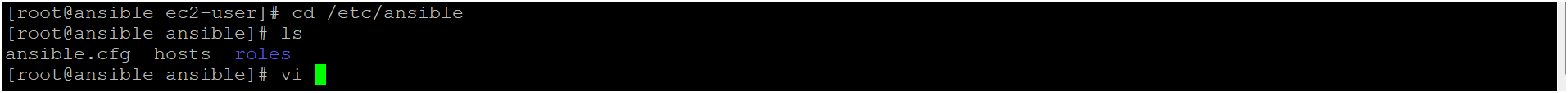
sudo dnf update -y

sudo dnf install -y ansible

ansible --version

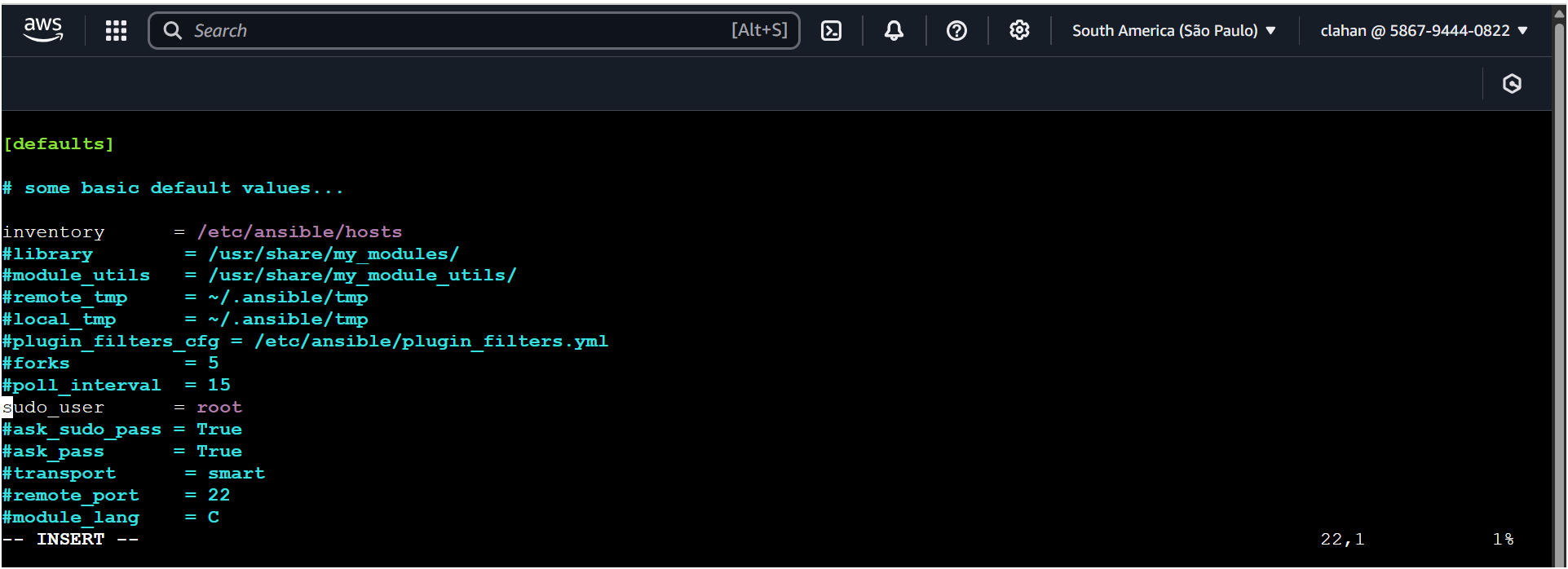
sudo curl -o /etc/ansible/ansible.cfg <https://raw.githubusercontent.com/ansible/ansible/stable-2.15/examples/ansible.cfg>

cd /etc/ansible



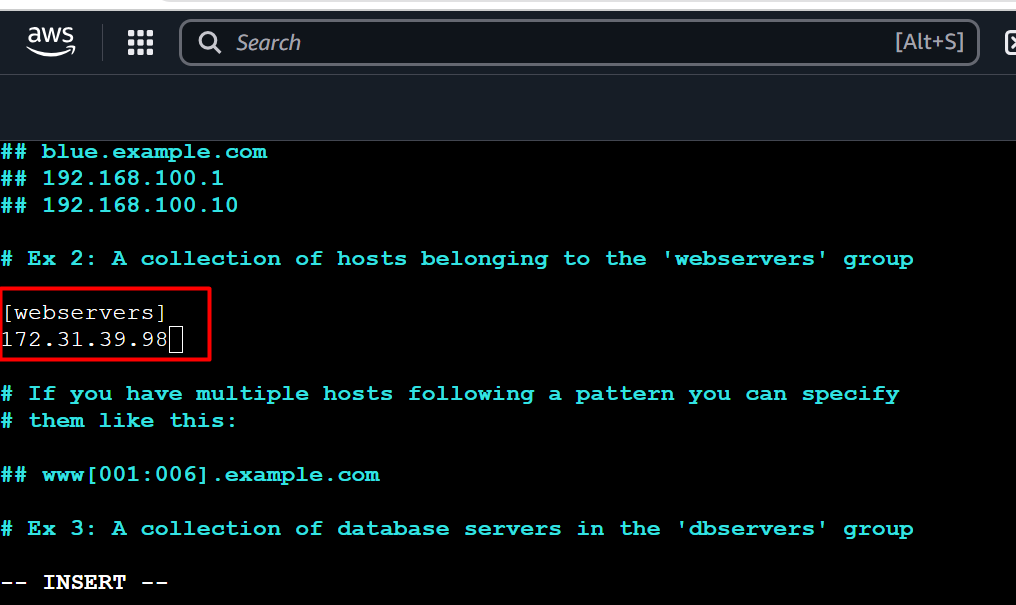
# open ansible ansible.cfg and un comment for inventory and root user

vi ansible.cfg



# open hosts file and uncomment for webserver and add your target machine pvt ip add

vi hosts



* Come to home directory

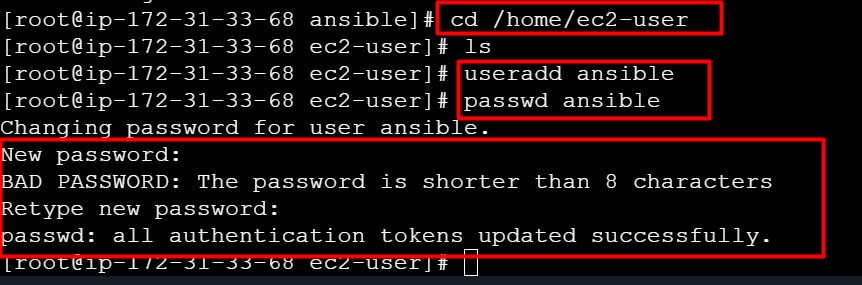
cd /home/ec2-user

* Create one user for ansible

useradd ansible

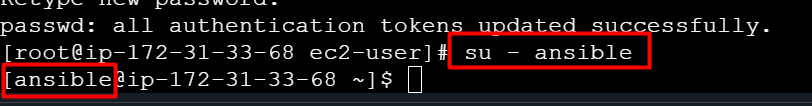
* Set a password for this user

passwd ansible

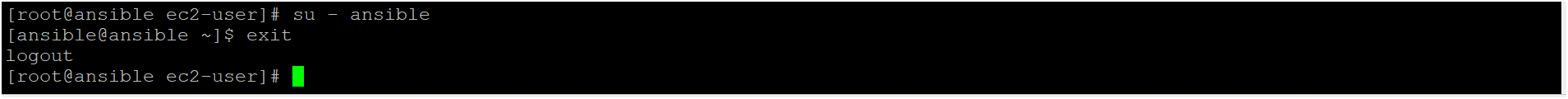


* Become as a ansible user

su - ansible



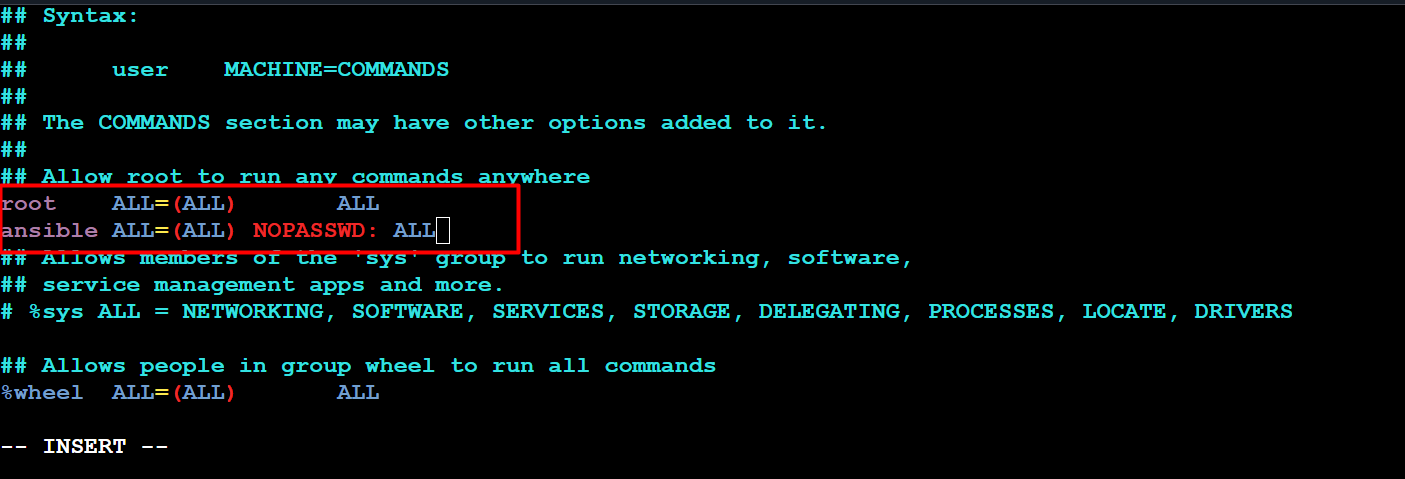
* Next you need to provide sudo privileges to ansible user (exit from the user)



* Open visudo file

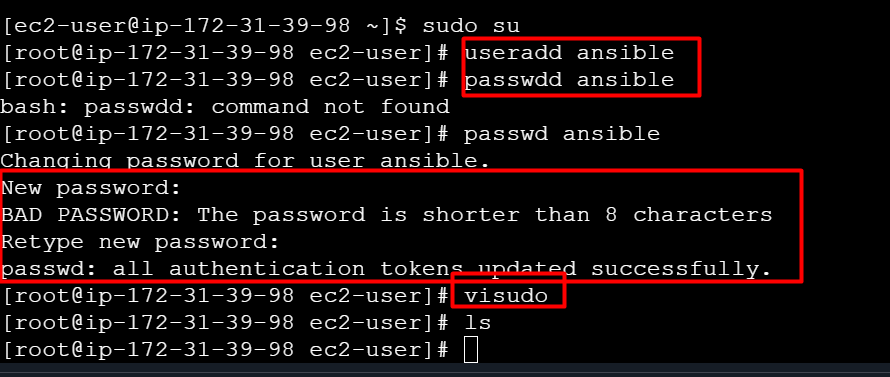
visudo

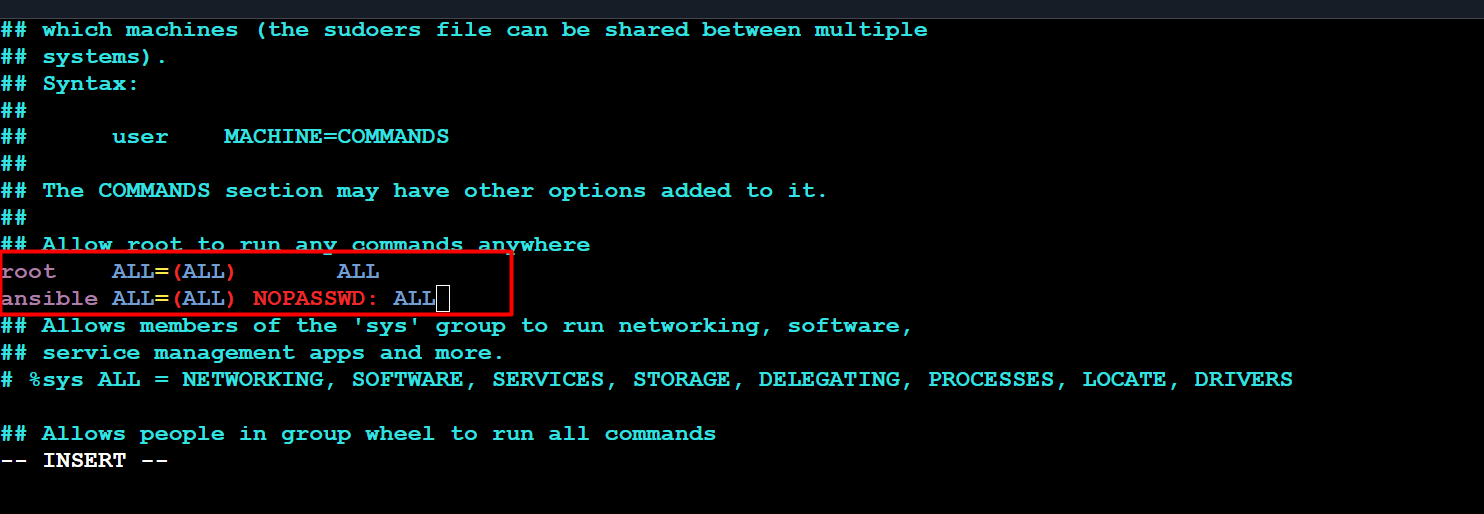
add this command under Root line ansible ALL=(ALL) NOPASSWD: ALL



* Now create one ansible user on host-server/target machine as same as ansible-server and give same password here.
* Open visudo file and give sudo privileges to ansible user

visudo

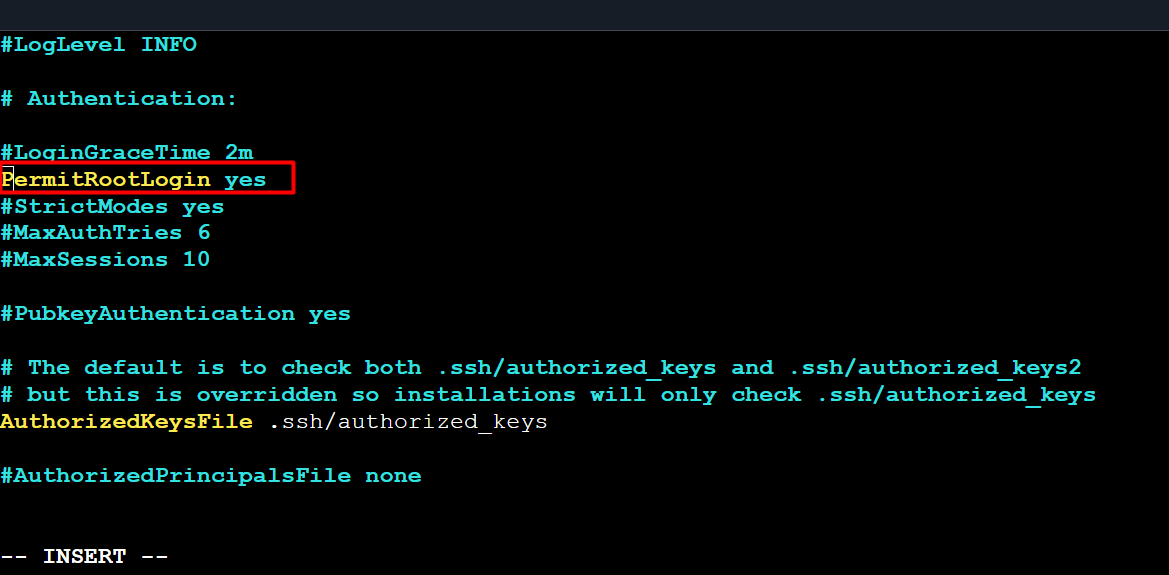


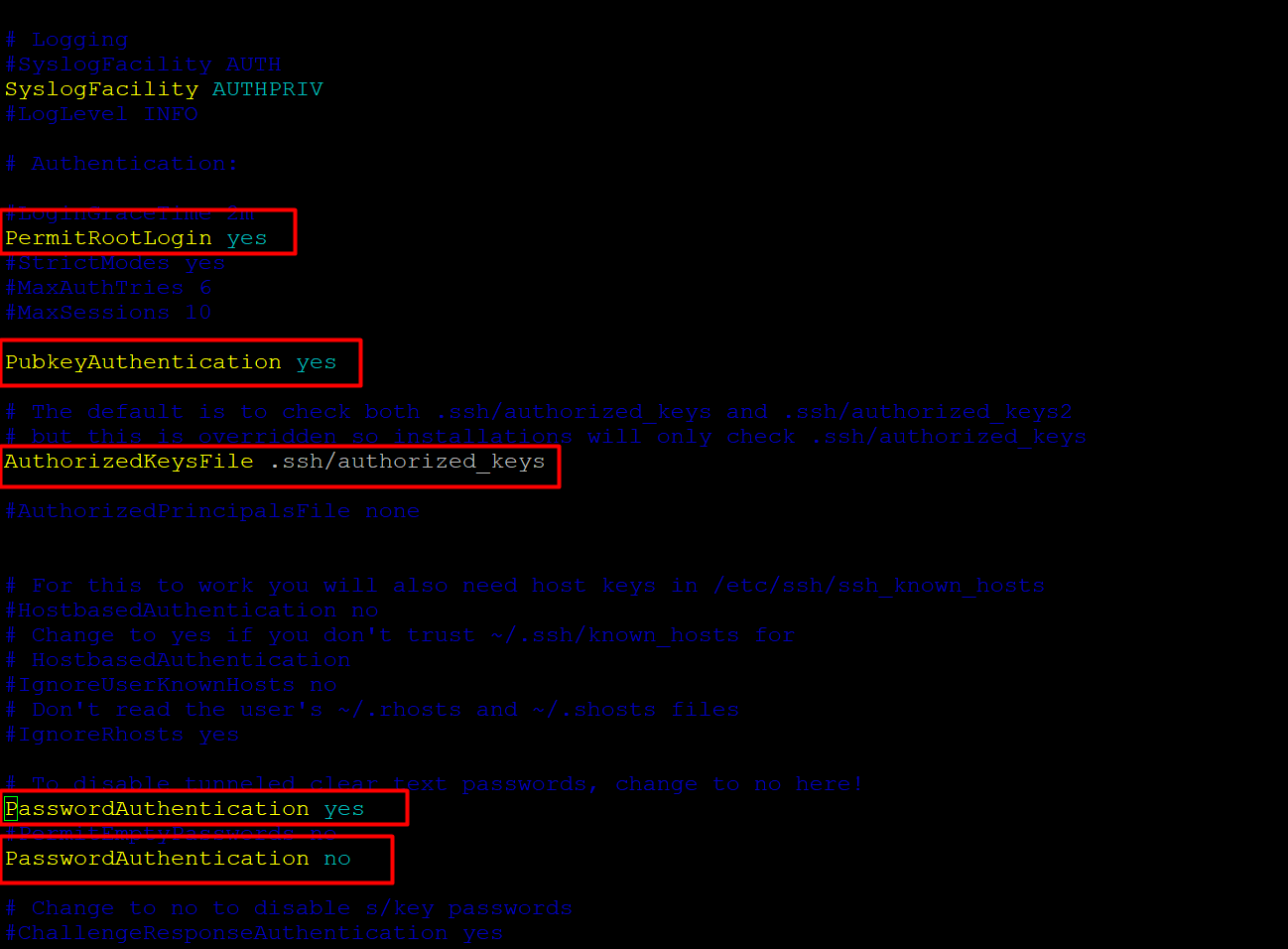


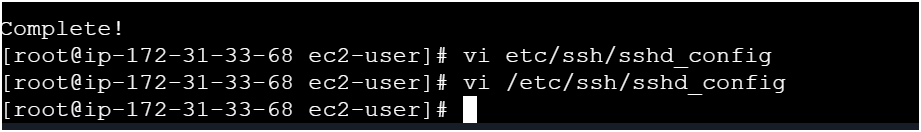
* We have to establish connection between ansible server and host server without passwd, follow the below steps on ansible server
* Open file /etc/ssh/sshd\_config

vi /etc/ssh/sshd\_config

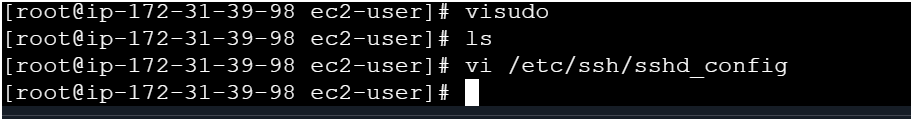
* Uncomment for PermitRootLogin yes and passwordAuthentication yes





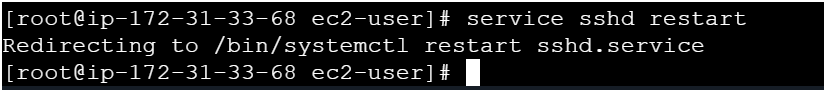


* Same steps do for host-server also ☝️
* vi /etc/ssh/sshd\_config (Uncomment for PermmitRoot Login Yes and passwordAuthentication yes)



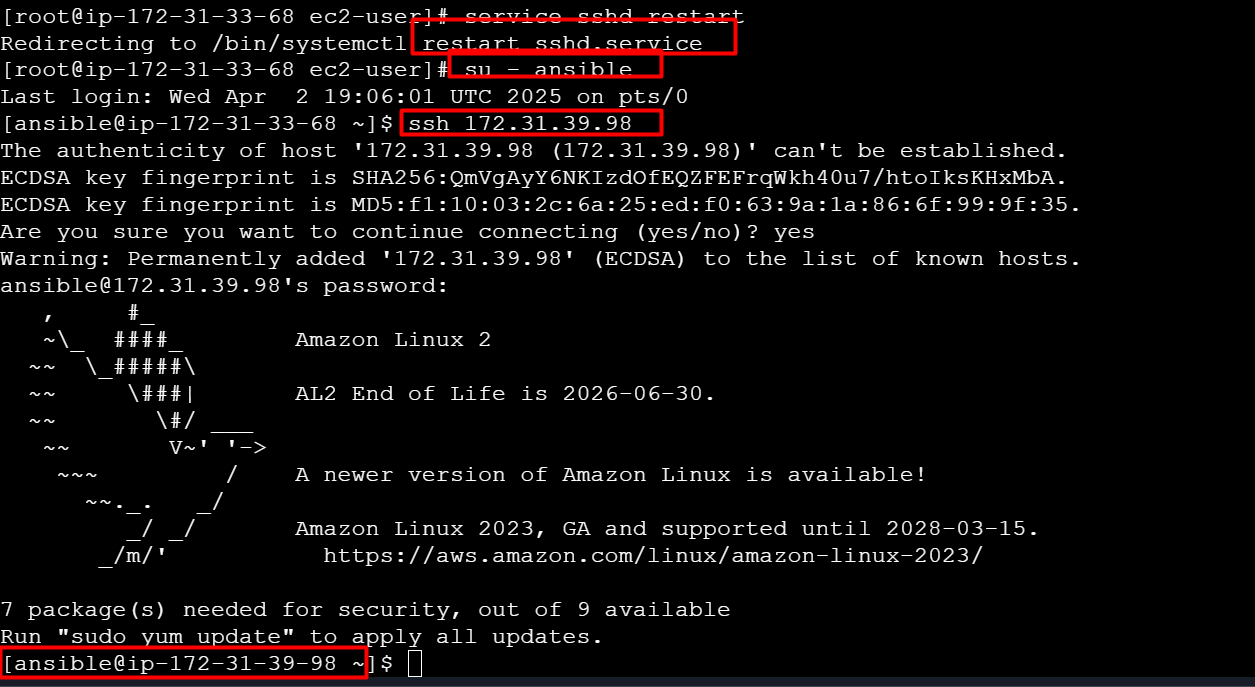
* Now you need to check connection between ansible-server and host-server
* Next you need to restart servers both ansible-server and host-server

service sshd restart (on both servers)

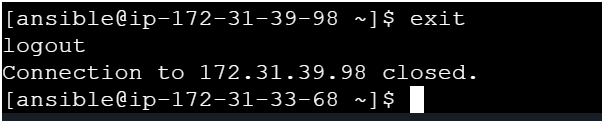


* Now become as a ansible user on ansible-server
* Next Copy pvt ip address of host-server
* Go to ansible server use below command

ssh <pvt-ip-add> (host-server pvt ip add)

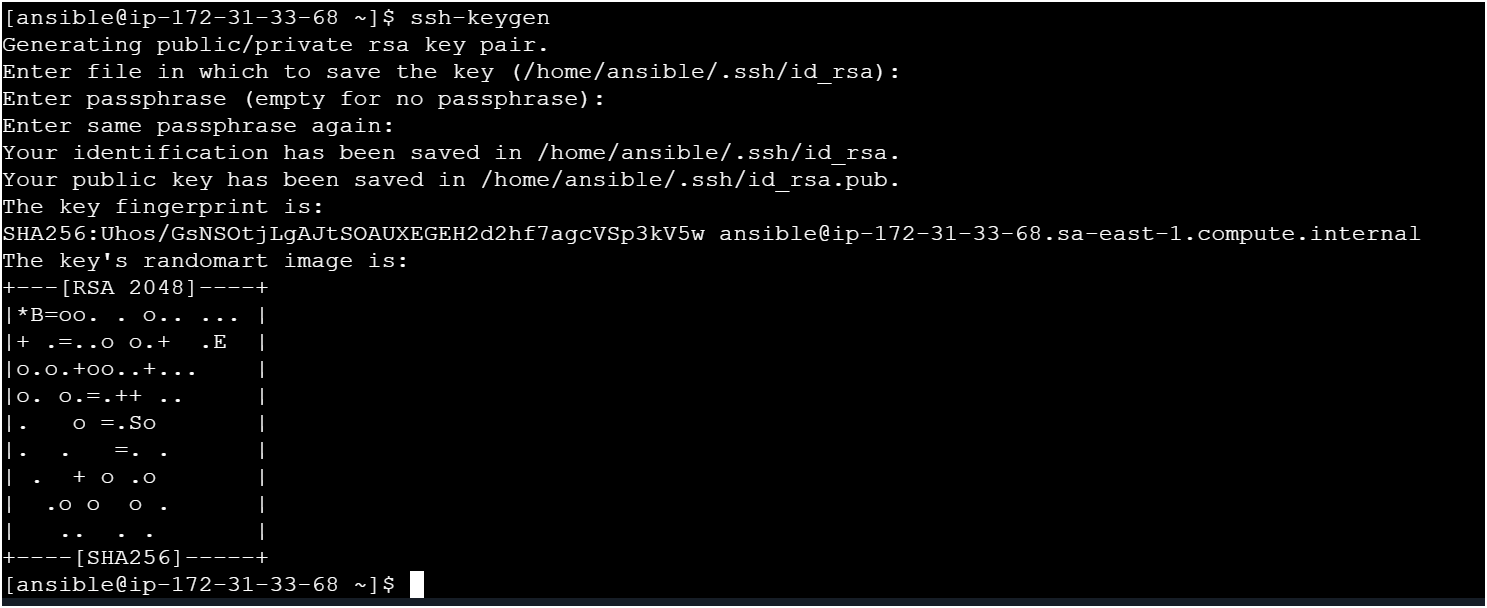


* Now its connected to your host-server (check pvt ip add of host-server)☝️
* Next exit from the host-server



* Now you need to make connection between server without providing password and follow below commands on ansible-server 👇(run commands as a ansible user only)

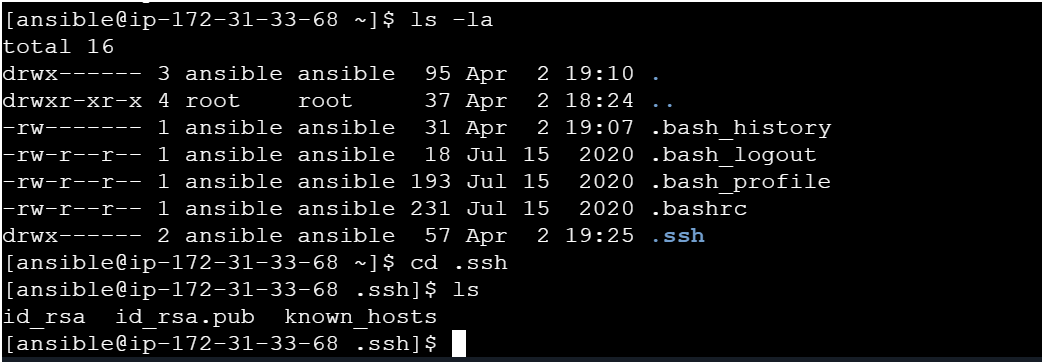
ssh-keygen (it will generate two types keys one is public key & pvt key



* Now look for key using

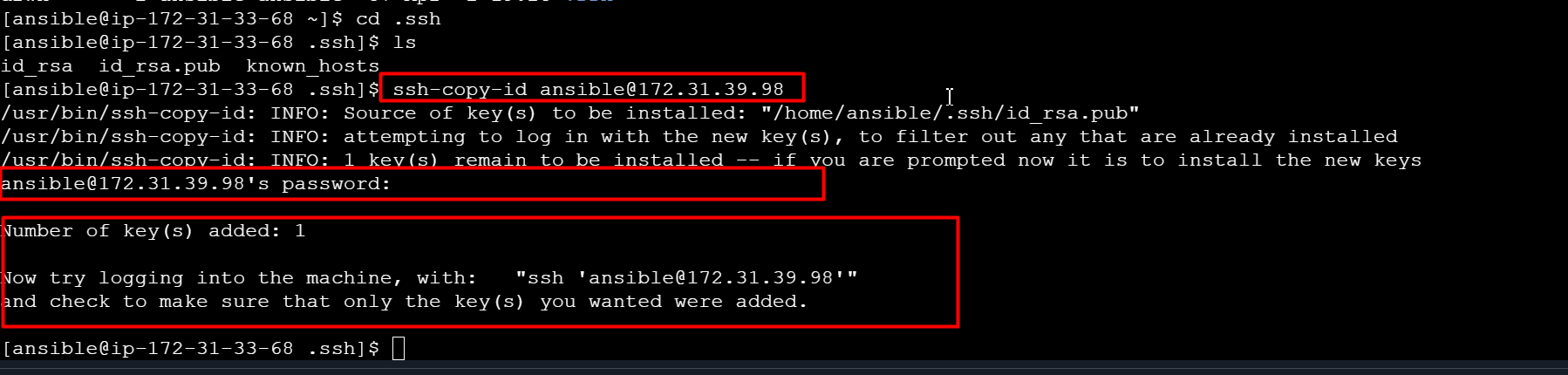
ls -la

cd ..ssh



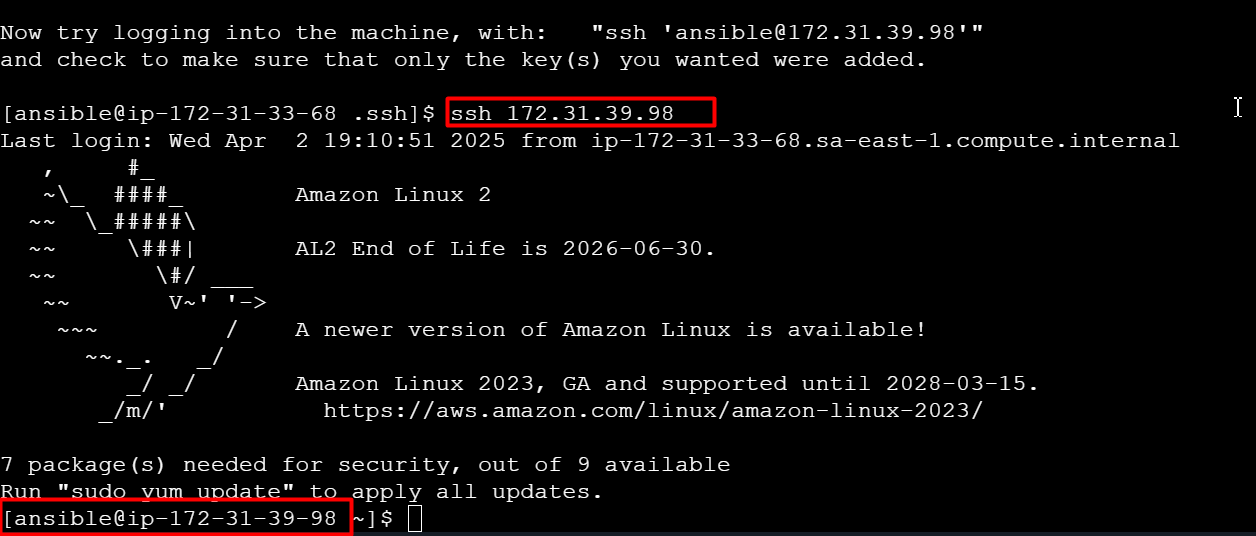
* How to copy public key to host-server from ansible-server 👇

ssh-copy-id ansible@pvt-ip-add-of-host-server and it will ask passwd provide ansible user passwd only 👇

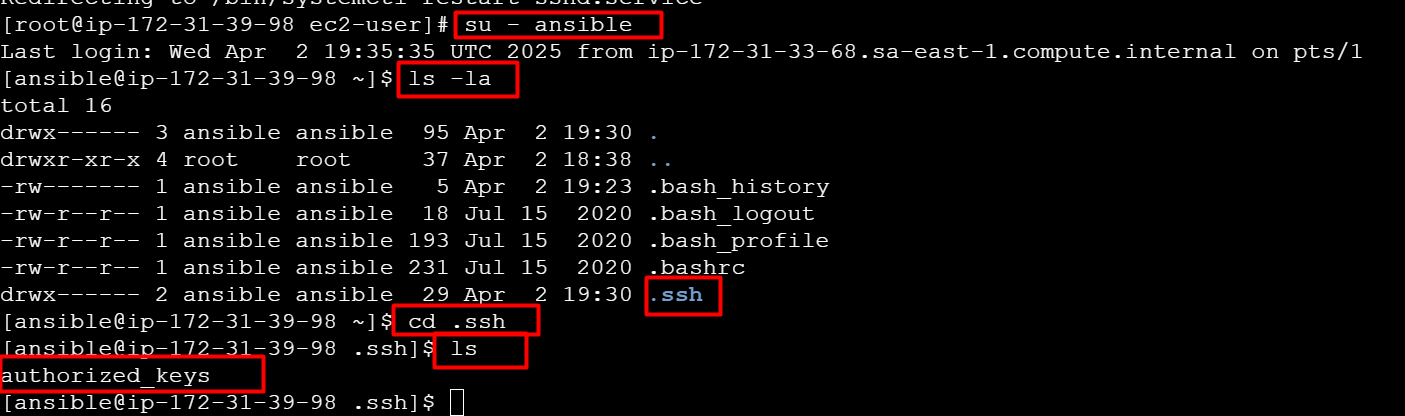


* Now do cross check by login to host-server from ansible-server without providing any passwd using below command 👇

ssh <pvt-ip-add> (host-server-pvt-ip-add)

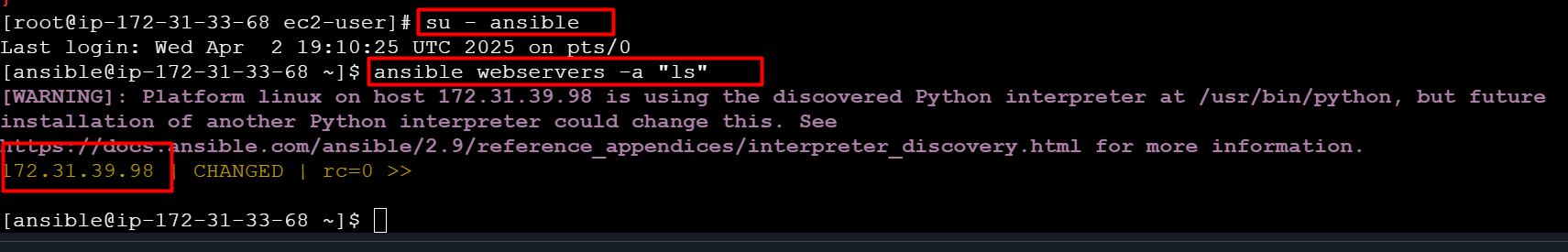


* Now you can check on host-server whether key is copied or not 👇



* Also you can check the connection form ansible-server using below command

ansible webservers -a “ls”



**Lab set was don**

**========================================================**

* Now we have to write playbooks use following script 👇
* Write playbooks on ansible-server only
* Create a file named with “my-demo-playbook.yaml”

vi my-demo-playbook.yaml

---

- hosts: webservers

user: ansible

become: yes

connection: ssh

tasks:

- name: install java on host machine

action: yum pkg=java state=present

- name: install mysql on host machine

action: yum pkg=mysql state=present

- name: install httpd on host machine

action: yum pkg=httpd state=present

- name: install docker on host machine

action: yum pkg=docker state=present

- name: install git on host machine

action: yum pkg=git state=present

# after installation is completed we can start services

- name: start httpd on host machine

action: service name=httpd state=started

- name: stop httpd on host machine

action: service name=httpd state=stopped

- name: restart httpd on host machine

action: service name=httpd state=started

- name: start docker on host machine

action: service name=docker state=started

- name: stop docker on host machine

action: service name=docker state=stopped

- name: restart docker on host machine

action: service name=docker state=started

* Now check syntax errors in your playbook

ansible-playbook <file-name> --syntax

* Next do a dry run before doing actual run

ansible-playbook <file-name> --check

* Now you can run your playbook

ansible-playbook <file-name>

---

- name: Deploy IPL-SRH static site on Apache

hosts: webservers

become: yes

remote\_user: ansible

connection: ssh

tasks:

- name: Install Apache (httpd)

ansible.builtin.yum:

name: httpd

state: present

- name: Start and enable Apache service

ansible.builtin.service:

name: httpd

state: started

enabled: yes

- name: Copy index.html

ansible.builtin.copy:

src: /home/ansible/IPL-SRH/index.html

dest: /var/www/html/index.html

owner: root

group: root

mode: 0644

- name: Copy CSS files

ansible.builtin.copy:

src: /home/ansible/IPL-SRH/ipl\_src\_css/

dest: /var/www/html/ipl\_src\_css/

owner: root

group: root

mode: 0755

- name: Copy images

ansible.builtin.copy:

src: /home/ansible/IPL-SRH/images/

dest: /var/www/html/images/

owner: root

group: root

mode: 0755

---

This is the standard YAML document start indicator. It tells YAML parsers this is the beginning of a new document.

- name: Deploy IPL-SRH static site on Apache

👉 This begins a **play** in Ansible.

* name: is just a human-readable label so you know what this play is doing.

hosts: webservers

👉 This tells Ansible **which group of hosts** to run the play on.

* In your inventory file (inventory.ini), you defined a group called webservers.

become: yes

👉 This means Ansible should **become root** (via sudo) when running tasks, because installing packages or managing services needs elevated permissions.

remote\_user: ansible

👉 This is the SSH user Ansible will use to connect to the target machines.  
In your case you created an ansible user on the target EC2 instance.

connection: ssh

👉 Tells Ansible to use SSH for connections (the default, but you can be explicit).

tasks:

👉 Starts a list of tasks to execute on the target hosts.

- name: Install Apache (httpd)

👉 Human-friendly description of the first task.

ansible.builtin.yum:

name: httpd

state: present

👉 This uses the yum module to:

install the httpd package (Apache web server)

state: present means “make sure it is installed”.

- name: Start and enable Apache service

👉 Next task description — manage Apache’s service.

ansible.builtin.service:

name: httpd

state: started

enabled: yes

👉 Uses the service module to:

ensure the httpd service is running (state: started)

and it is enabled at boot (enabled: yes)

- name: Copy index.html

👉 Next task: copy the HTML home page to the web server directory.

ansible.builtin.copy:

src: /home/ansible/IPL-SRH/index.html

dest: /var/www/html/index.html

owner: root

group: root

mode: 0644

👉 Uses the copy module to:

copy index.html from your control node (the Ansible machine)

to the target node’s web root

set owner/group to root

set file permissions to 0644 (readable by everyone, writable by root)

- name: Copy CSS files

👉 Copies CSS assets in a similar way.

ansible.builtin.copy:

src: /home/ansible/IPL-SRH/ipl\_src\_css/

dest: /var/www/html/ipl\_src\_css/

owner: root

group: root

mode: 0755

👉 Copies the entire CSS directory to the web server.

0755 means read/execute by everyone, write for owner only.

This is fine for static assets like CSS.

- name: Copy images

👉 Copies the images directory to the server.

ansible.builtin.copy:

src: /home/ansible/IPL-SRH/images/

dest: /var/www/html/images/

owner: root

group: root

mode: 0755

👉 Again, standard directory copy, with safe permissions.

=======================================

# install Jenkins on target machine

sudo yum update -y

sudo amazon-linux-extras install java-openjdk11 -y

wget -O /etc/yum.repos.d/jenkins.repo \

https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

sudo yum install jenkins -y

sudo systemctl enable jenkins

sudo systemctl start Jenkins

=========================================================

# execute this commands on ansible server

ansible-galaxy collection install amazon.aws

=====================================

# Jenkins pipeline script

pipeline {

agent any

environment {

ANSIBLE\_INVENTORY = "/var/lib/jenkins/inventory.ini"

}

stages {

stage('Checkout') {

steps {

git 'https://github.com/Msocial123/IPL-SRH.git'

}

}

stage('Run Ansible Playbook') {

steps {

sh """

ansible-playbook -i ${ANSIBLE\_INVENTORY} ${WORKSPACE}/ipl-playbook.yml

"""

}

}

}

post {

success {

echo '✅ Deployment successful!'

}

failure {

echo '❌ Deployment failed.'

}

}

}