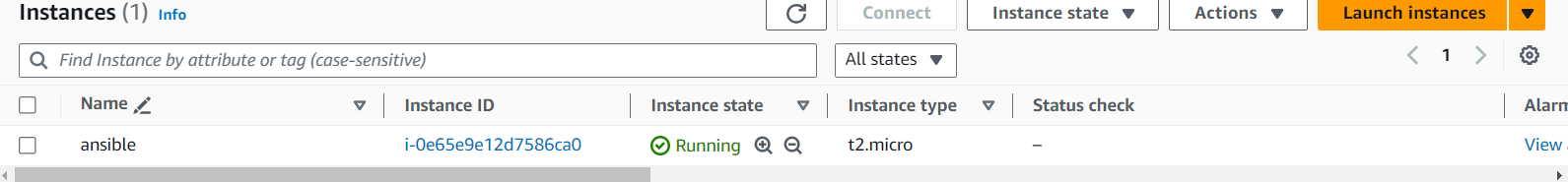
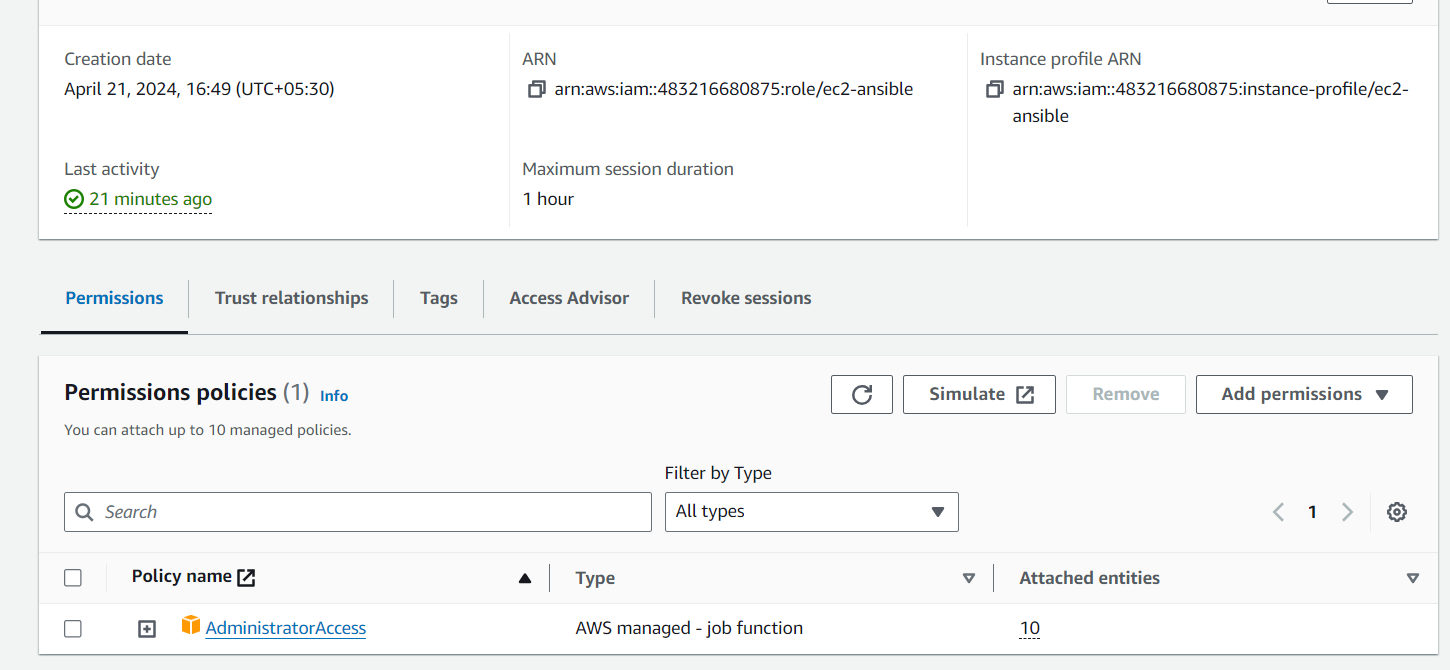
**Dynamic inventory file process**

Launch ansible master node

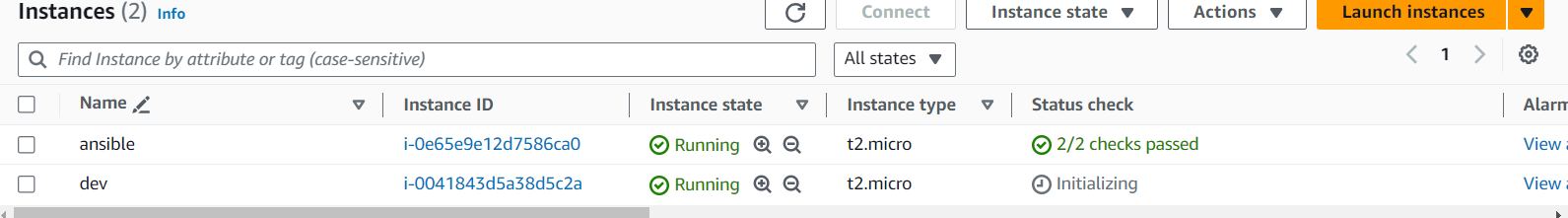


--- attach IAM role to ansible master

* Ec2 – administrator access or Ec2 full access



Create a node with name of dev



----- Install bellow command in ansible master

sudo yum install ansible -y

sudo yum update -y

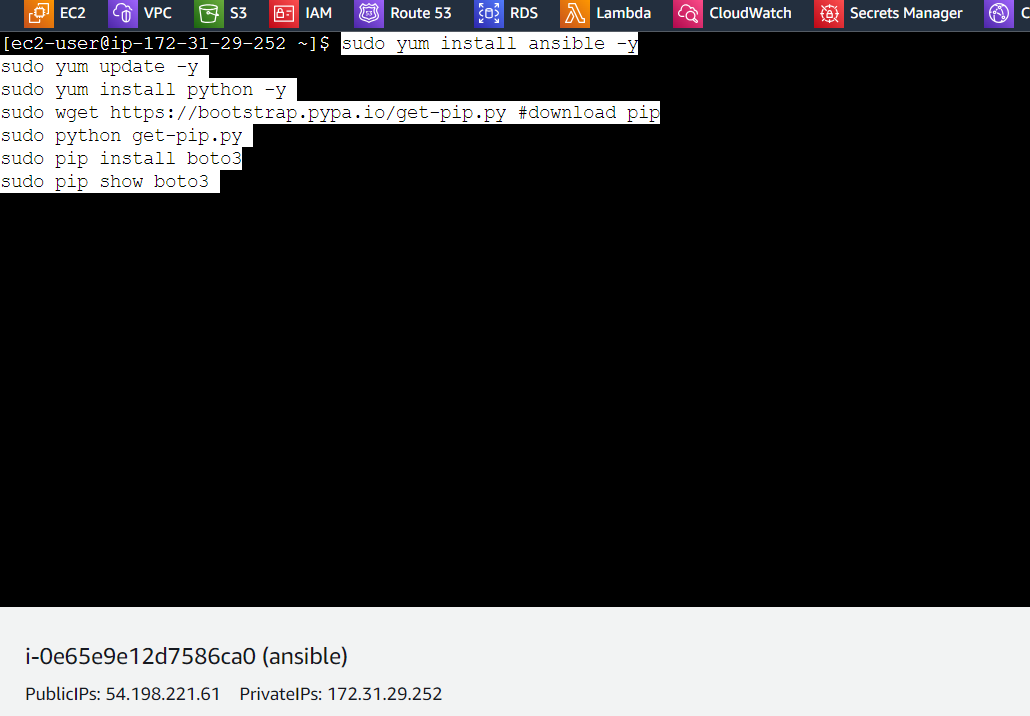
sudo yum install python -y

sudo wget https://bootstrap.pypa.io/get-pip.py #download pip (or) yum install python3-pip -y

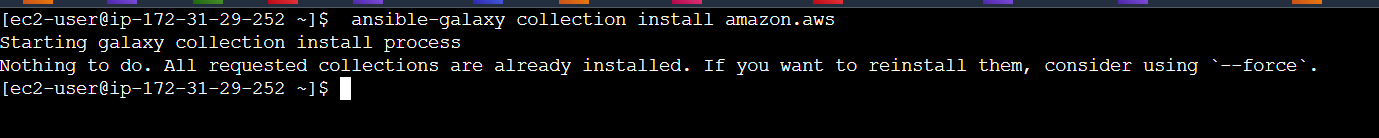
sudo python get-pip.py

sudo pip install boto3

sudo pip show boto3

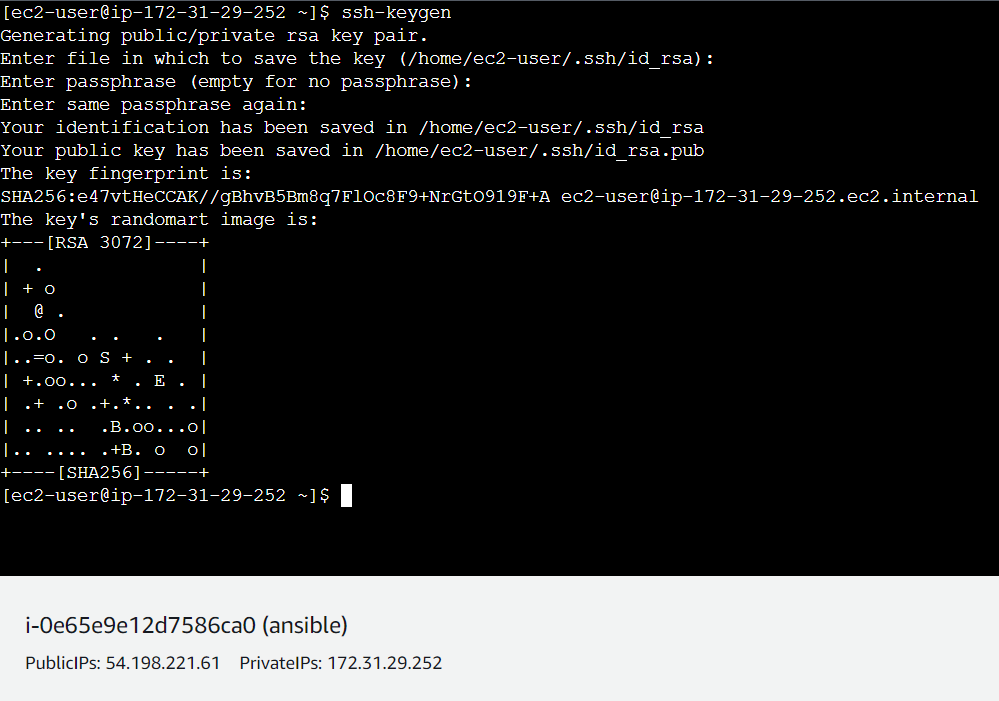


* Install ansible galaxy
* ansible-galaxy collection install amazon.aws



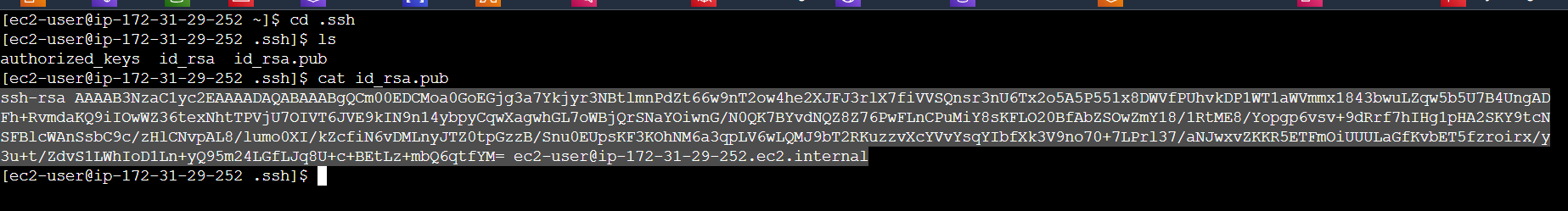
--- generate the keys

* ssh-key gen

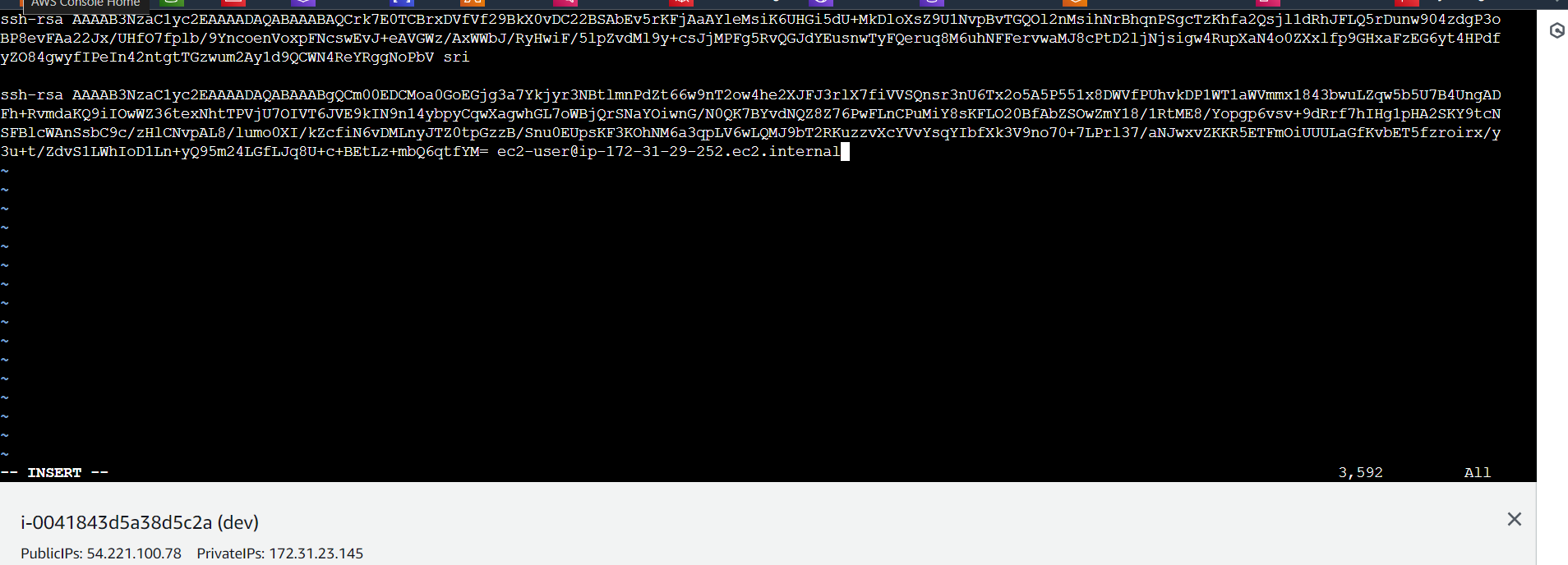


--- copy the public key

* cd .ssh
* cat id\_rsa.pub



--- paste the public on dev node



-----go to ansible.cfg file and paste below data

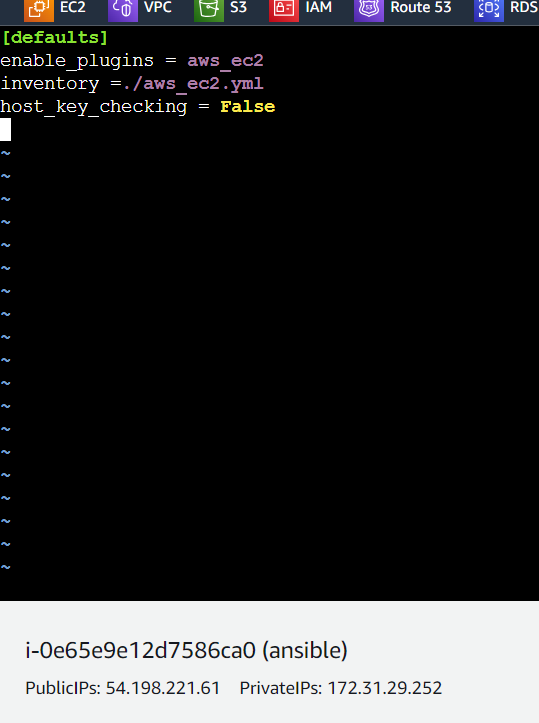
-------sudo vi /etc/ansible/ansible.cfg

[defaults]

enable\_plugins = aws\_ec2

inventory =./aws\_ec2.yml

host\_key\_checking = False



**THIS IS FOR HOST =ALL**

----create a aws\_ec2.ymal file for enable the plugins

* sudo vi /etc/ansible/aws\_ec2.yml
* paste the below mater

plugin: amazon.aws.aws\_ec2

regions:

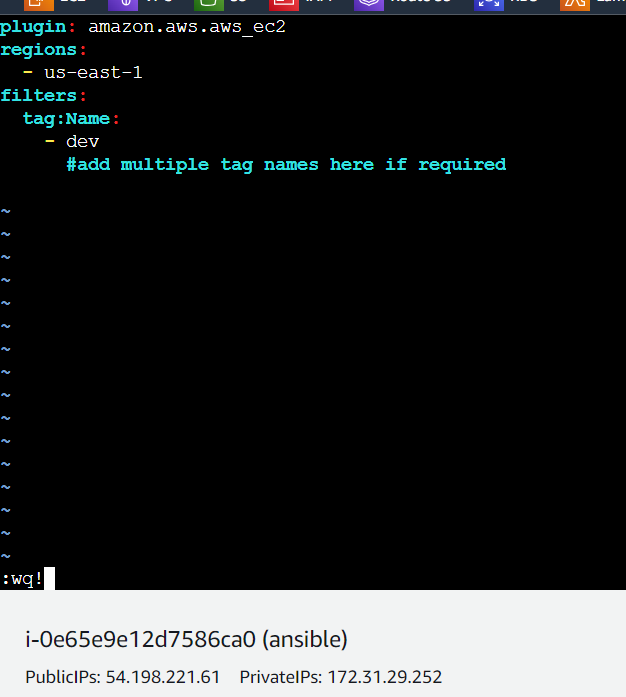
- us-east-1

filters:

tag:Name:

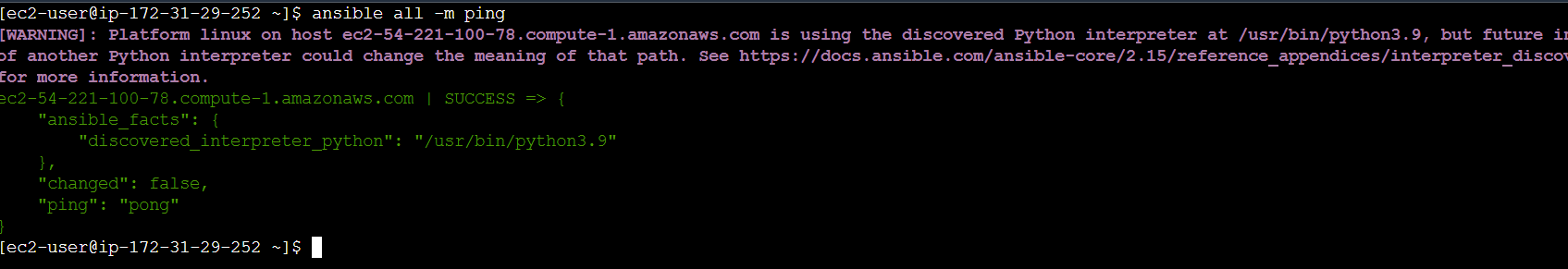
- dev

#add multiple tag names here if required



------give the ping command

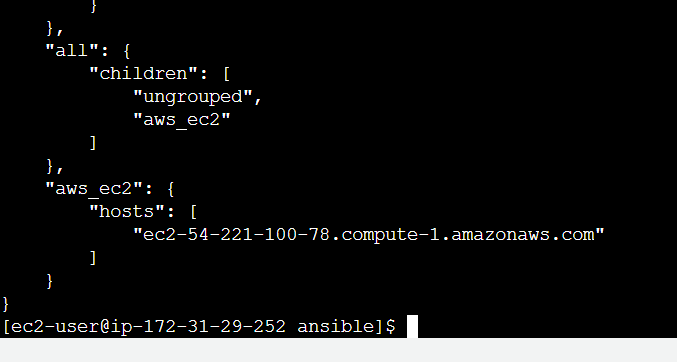
* ansible all -m ping

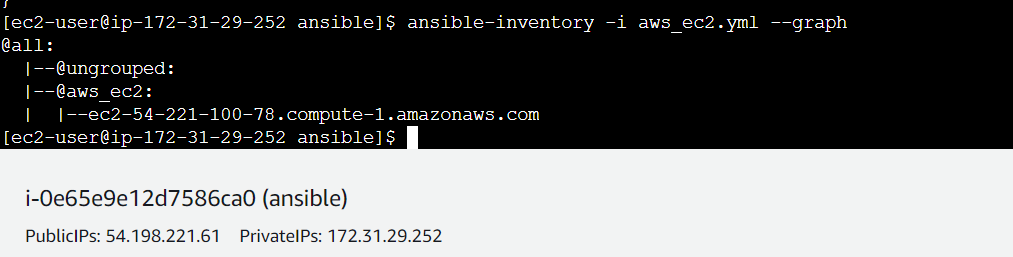


* go to ansible directory apply below list and graph commands to get details

----- cd /etc/ansible/

* ansible-inventory -i aws\_ec2.yml --graph



ansible-inventory -i aws\_ec2.yml --graph

* create a sample playbook in same ansible directory

---

- name: first playbook

hosts: all

become: yes

tasks:

- name: install httpd software

yum:

name: httpd

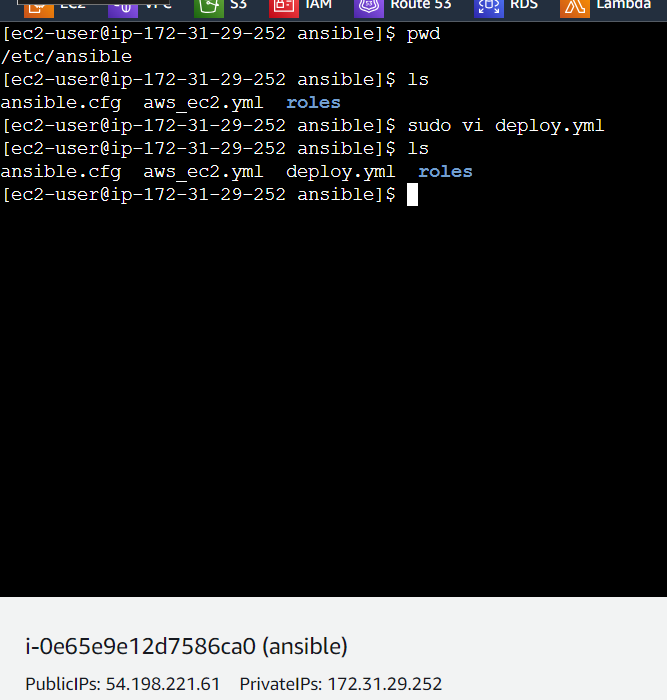
state: latest

- name: start web server

service:

name: httpd

state: started

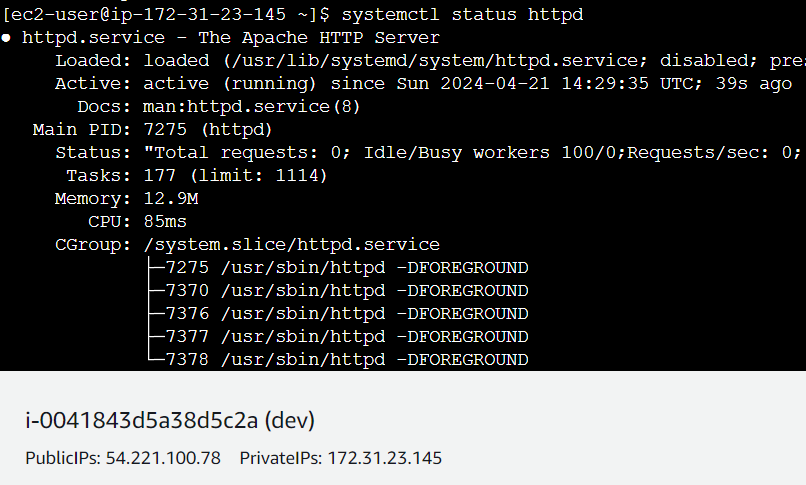


--- apply the playbook

* ansible-playbook deploy.yml



* check in node



--- option 1 :::::::create a one node with same tags {dev} and connect to the node copy paste the public key

---option2 :;:::::: create a AMI from the existing dev node. Launch the instance from ami with same tags {{dev}}. In this procress noo need to copy paste the keys.

------ after completing of above any one step . goo ansible master give list and graph commad u will get new node details

---------------------------------------------------------------------------

**THIS IS FOR HOST = GROUP NAME**

---This group process I am doing in same ansible

--If u have interest create a new resources doo same process until

Aws\_ec2.yml creation from here everything in changed

* Edit the aws\_ec2.ymal
* Iam commented filters and added key\_groups

---

plugin: amazon.aws.aws\_ec2

regions:

- us-east-1

#filters:

# tag:Name:

# - dev

#add multiple tag names here if required

keyed\_groups:

# add hosts to tag\_Name\_value groups for each aws\_ec2 host's tags.Name variable.

- key: tags.Name

prefix: tag\_Name\_

separator: ""

groups:

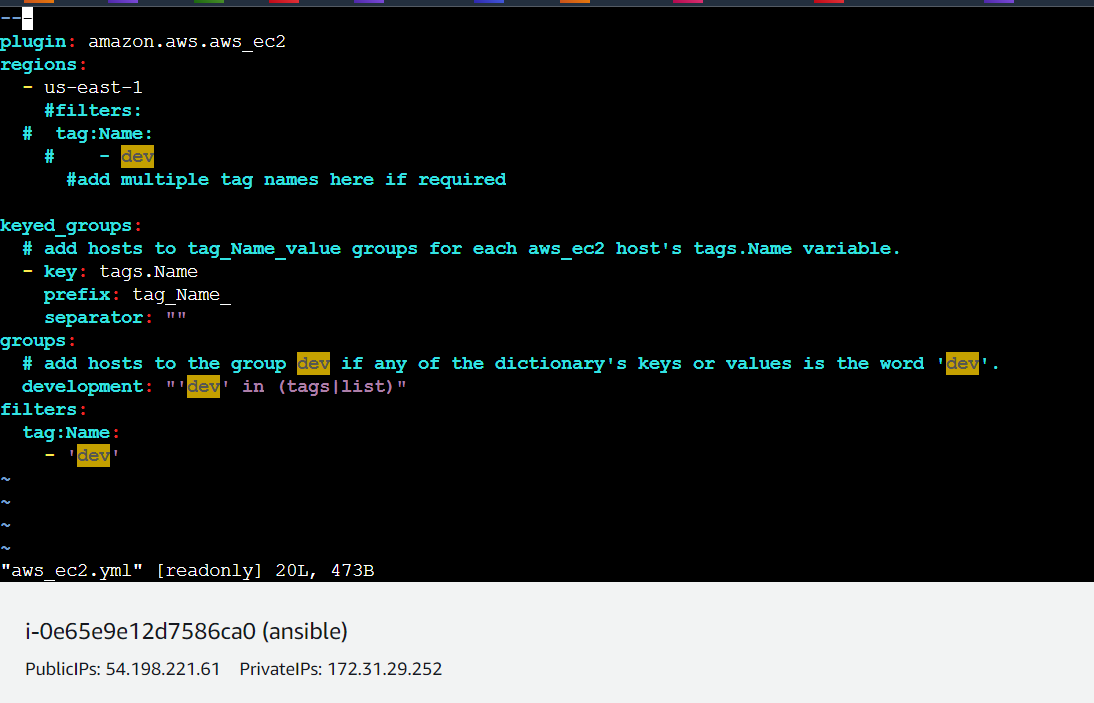
# add hosts to the group dev if any of the dictionary's keys or values is the word 'dev'.

development: "dev” in (tags|list)"

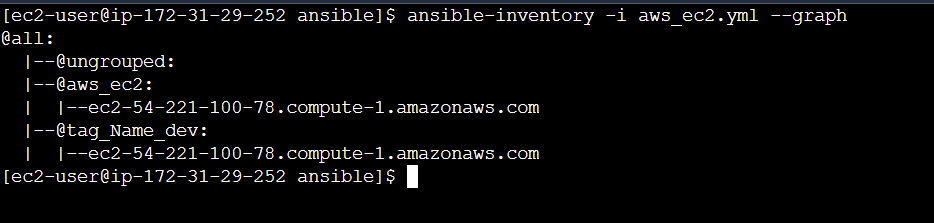
filters:

tag:Name:

- 'dev'



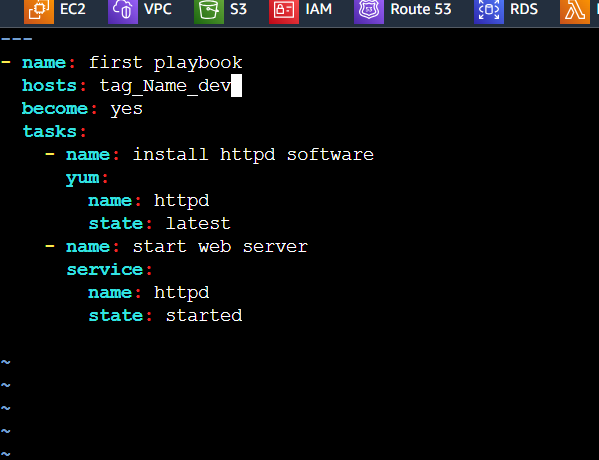
--- give graph command to get details. In ungrouped one instance addeed



--- give list command to get details group in added instances

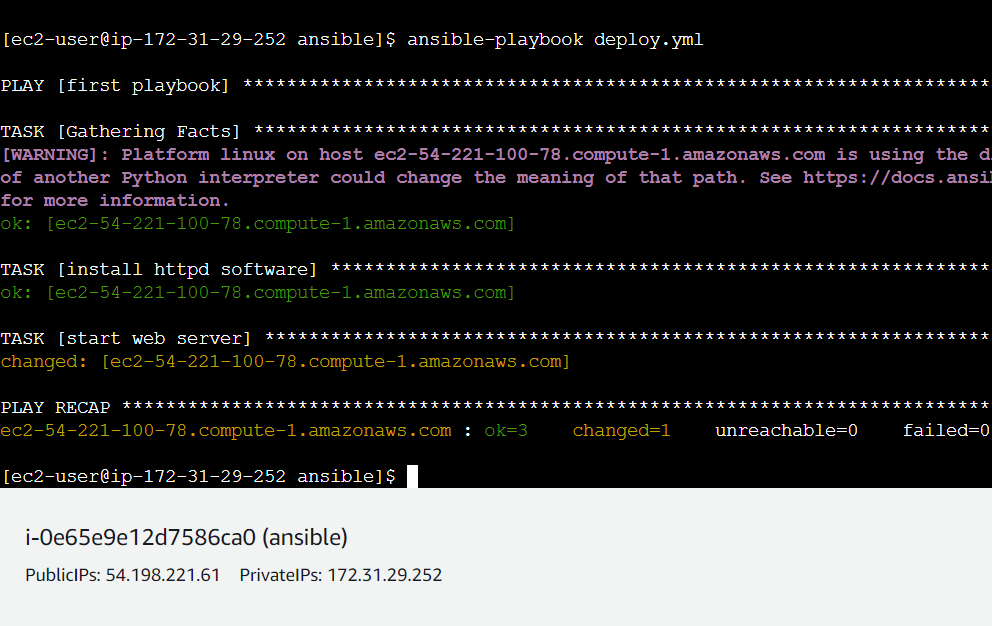


---create a deploy.yml



* Deploy the deploy.yaml file

---ansible-playbook deploy.yaml



---- Here instance tag name dev and group name dev so dev instance is added into dev group

--- you will create a new group in aws\_ec2.yml with same instance name dev give list command to verify the instance groups