Ansible Dynamic Inventory AWS

https://www.linkedin.com/in/aziz-afoni-811503227/

Content

1. Setup Ansible Dynamic inventory
2. Grouping EC2 resources with dynamic inventory
3. Execute ansible commands with ec2 dynamic inventory
4. Using dynamic inventory inside a playbook

Prerequisites

1. AWS account
2. Knowledge in Ansible

Step 1: check if python3 & pip3 are installed in ansible server

- Python –version
- Used the following commands to install python and pip if you don't have it
- For Debian, Ubuntu-> `sudo apt-get install python3 -y`
- `sudo apt-get install python3-pip -y`

- For centos, Redhat, -> `sudo yum install python3 -y`
- `sudo yum –y install python3-pip`

step 2: installing boto3 library so ansible can be able to make API calls to AWS and retrieve EC2 instance details

- `sudo pip3 install boto3`

```
-bash: pip: command not found
[ec2-user@ip-172-31-18-30 ~]$ sudo pip3 install boto3
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Collecting boto3
  Downloading boto3-1.22.8-py3-none-any.whl (132 kB)
     |████████████████████████████████| 132 kB 16.5 MB/s
Collecting s3transfer<0.6.0,>=0.5.0
  Downloading s3transfer-0.5.2-py3-none-any.whl (79 kB)
     |████████████████████████████████| 79 kB 14.1 MB/s
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.0-py3-none-any.whl (23 kB)
Collecting botocore<1.26.0,>=1.25.8
  Downloading botocore-1.25.8-py3-none-any.whl (8.7 MB)
     |████████████████████████████████| 8.7 MB 38.5 MB/s
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     |████████████████████████████████| 247 kB 50.7 MB/s
Collecting urllib3<1.27,>=1.25.4
  Downloading urllib3-1.26.9-py2.py3-none-any.whl (138 kB)
     |████████████████████████████████| 138 kB 55.0 MB/s
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, python-dateutil, jmespath, urllib3, botocore, s3transfer, boto3
Successfully installed boto3-1.22.8 botocore-1.25.8 jmespath-1.0.0 python-dateutil-2.8.2 s3transfer-0.5.2 six-1.16.0 urllib3-1.26.9
[ec2-user@ip-172-31-18-30 ~]$
```

step 3: create an inventory directory

- `sudo mkdir -p /opt/ansible/inventory`
- `cd /opt/ansible/inventory`
- create a file names aws_ec2.yaml in the inventory directory
- sudo touch aws_ec2.yaml
- open the file using any editor of your choice
- sudo vi aws_ec2.yaml

  option1: if your ansible server is not running in aws use access key and secret access key. paste the following config in aws-ec2.yaml

  N:B make sure your yaml script follows the right format can use a yaml validator to check

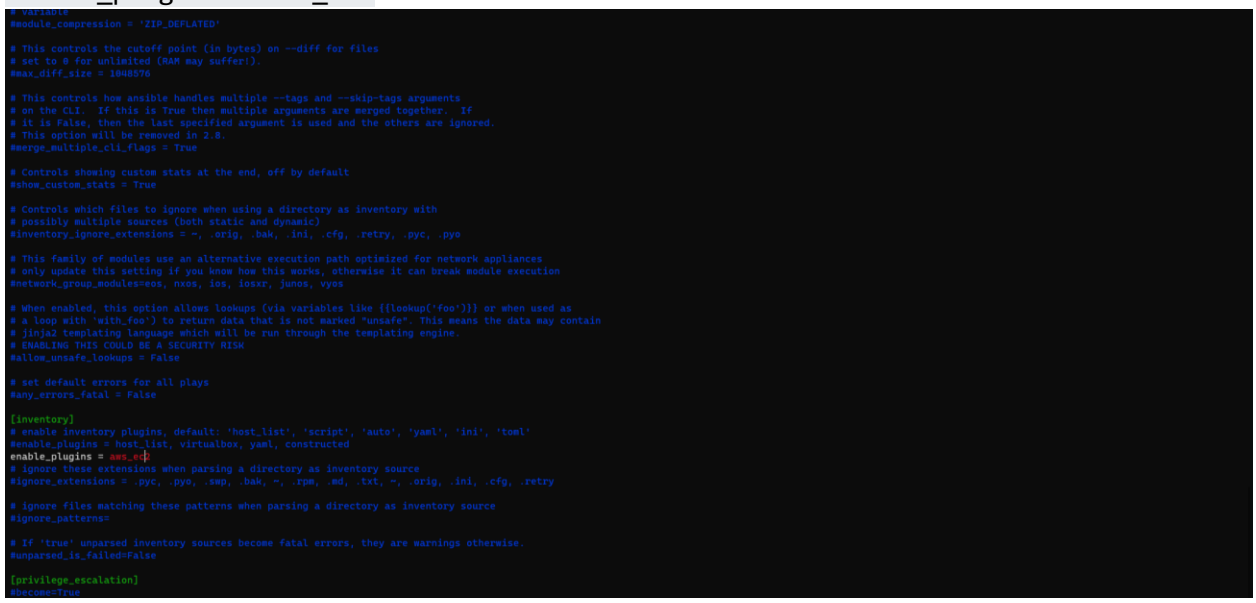
```
---
 plugin: aws_ec2
 aws_access_key:
 aws_secret_key:
 keyed_groups:
  - key: tags
    prefix: tag
```

  option2: if your ansible is running on aws use an ec2 role for this demo I use an ec2 role with ec2fullaccess

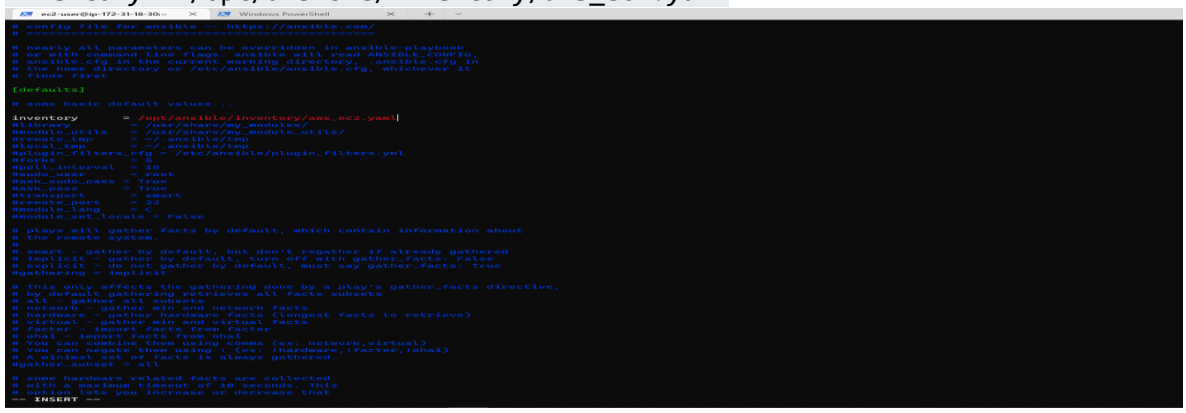
- navigate to IAM -> Roles -> Create role -> select ec2 under use case -> amazonec2fullaccess
- click next -> give a name to the role -> the create role
- navigate back to ec2 -> select ansible server-> actions -> Security -> modify IAM role-> select the role and save
- paste the following in /opt/ansible/inventory/aws_ec2.yaml

  plugin: aws_ec2
  keyed_groups:
   - key: tags
     prefix: tag

step 4 : open `/etc/ansible/ansible.cfg`

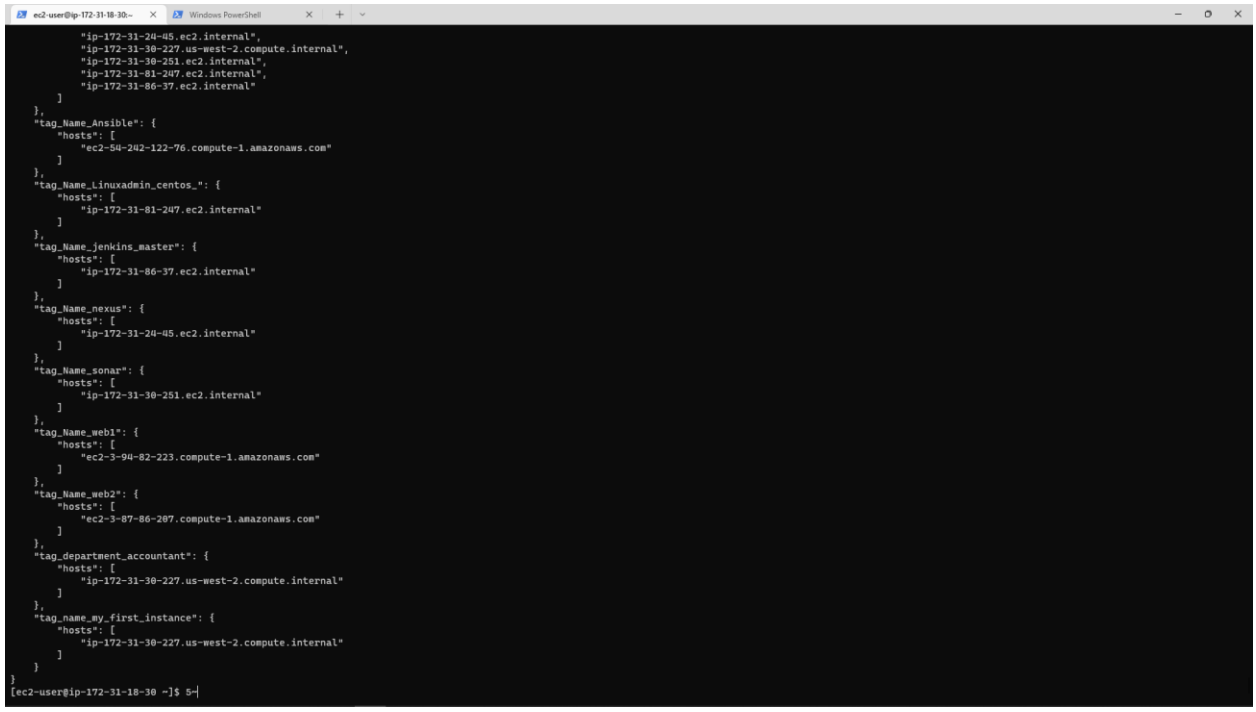
- `sudo vi /etc/ansible/ansible.cfg`
- Find the `[inventory]` section and add the following line to enable the ec2 plugin
- `enable_plugins = aws_ec2`
- 

Step 5: test dynamic inventory configuration by listing ec2 instances

- `ansible-inventory -i /opt/ansible/inventory/aws_ec2.yaml –list`
- if you encounter an error. `ERROR! The ec2 dynamic inventory plugin requires boto3 and botocore.`
- Use sudo yum install python-boto3 and then retry the command
- The above command returns the list of ec2 instances with all its parameters in JSON format.
- If you want to use the dynamic inventory as a default Ansible inventory, edit the /etc/ansible/ansible.cfg file and search for inventory parameters under defaults. Change the inventory parameter value as shown below.
- `inventory = /opt/ansible/inventory/aws_ec2.yaml`
- 

- Now run inventory list command without passing the inventory file

- Ansible-inventory –list

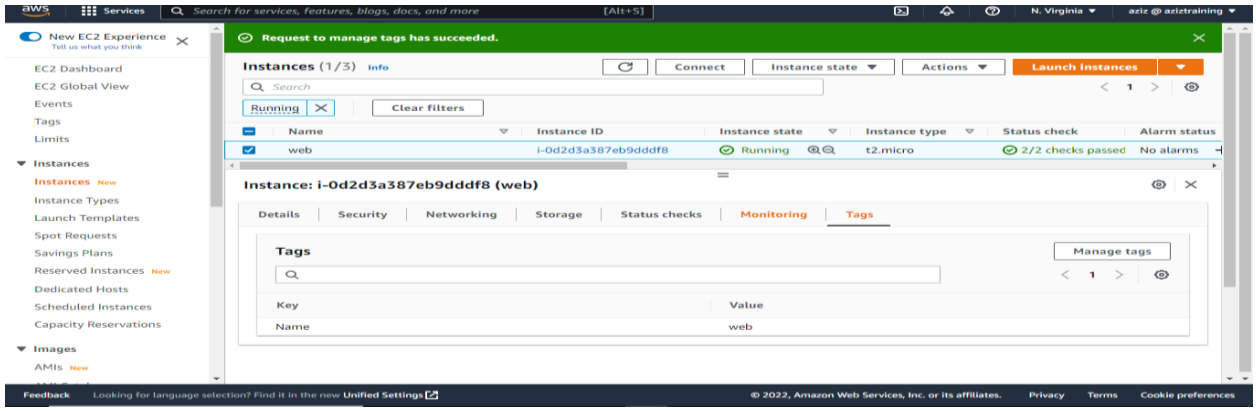
- 

Step6: grouping ec2 resources based on tags

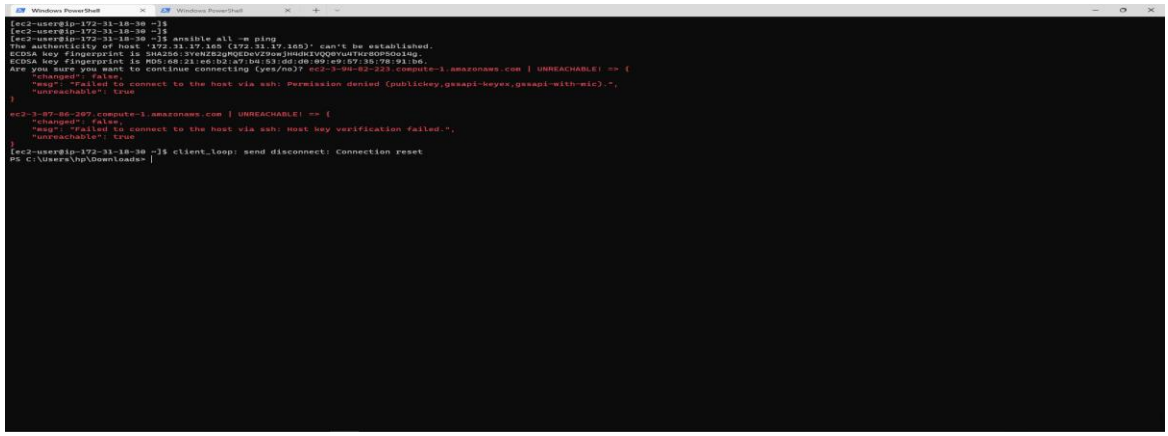
- The primary use case of AWS Ansible dynamic inventory is to execute Ansible playbooks or ad-hoc commands against a single or group of categorized or grouped instances based on tags, regions, or other ec2 parameters.
- you can group instances using tags, instances type, instance names, custom filters, and more. Take a look at all supported filters and keyed groups from here.
- I will edit my ansible inventory file to group based on a specific tag

```
-   ---
-   plugin: aws_ec2
-   regions :
-    - us-east-1
-   filters:
-    tag:Name: web
-
```

- in this new inventory ansible get our ec2 instances that have a tag Name: web and in the region us-east-1



- let's ping them real quick

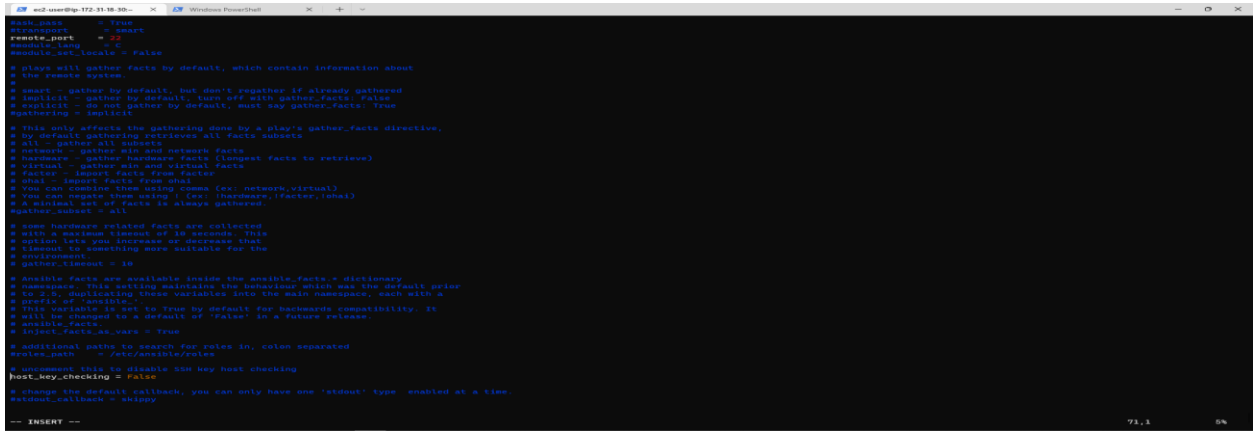


-
- NOTE: we are not able to ping the instances because we haven't configured an ssh agent yet
- Using SSH agent is the best way to authenticate to your end nodes as this elevates the need to copy your .pem files around

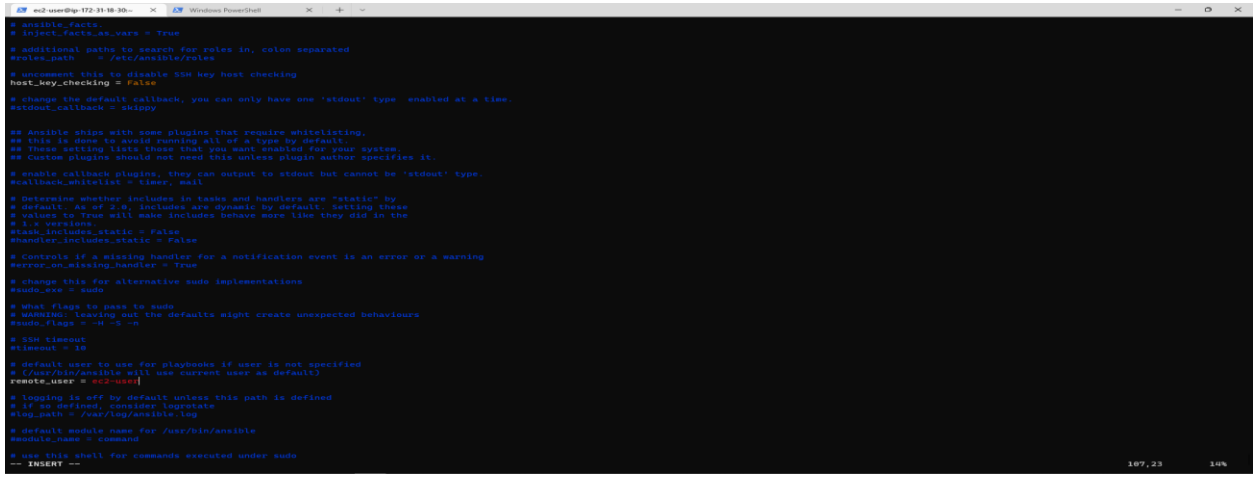
Step 7: configure SSH

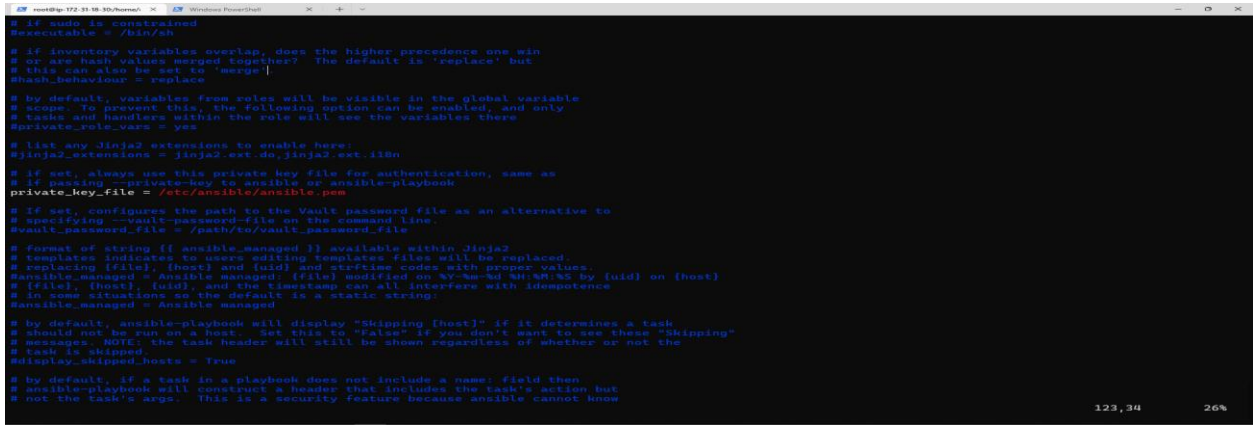
1. In sudo vi /etc/ansible/ansible.cfg enable the following by uncommenting

- Remote_port=22
- Host-key_checking= false
- since ssh will be done by ansible there is no need for host key checking

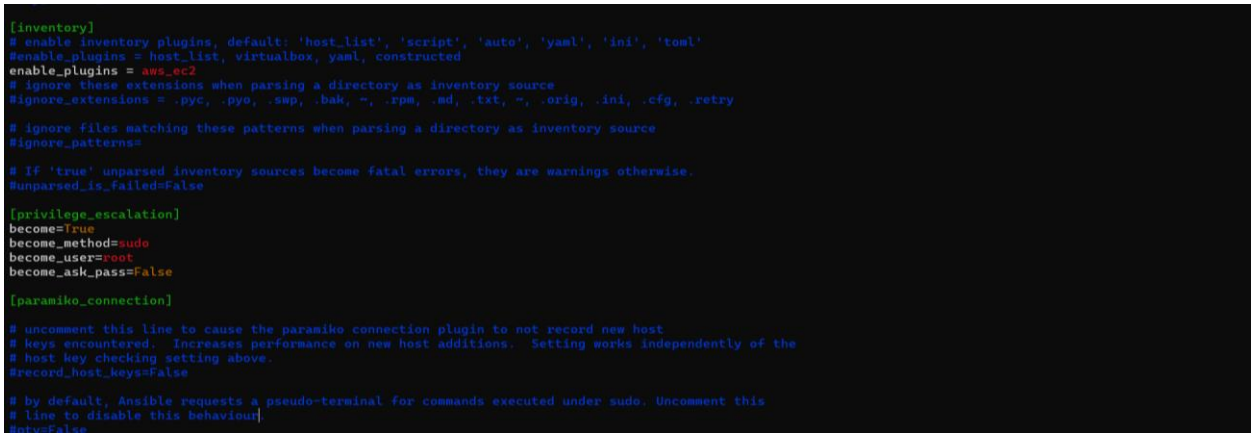


- Uncomment remote user and change to your default user on your ec2 instance I am user linux so mine is ec2-user

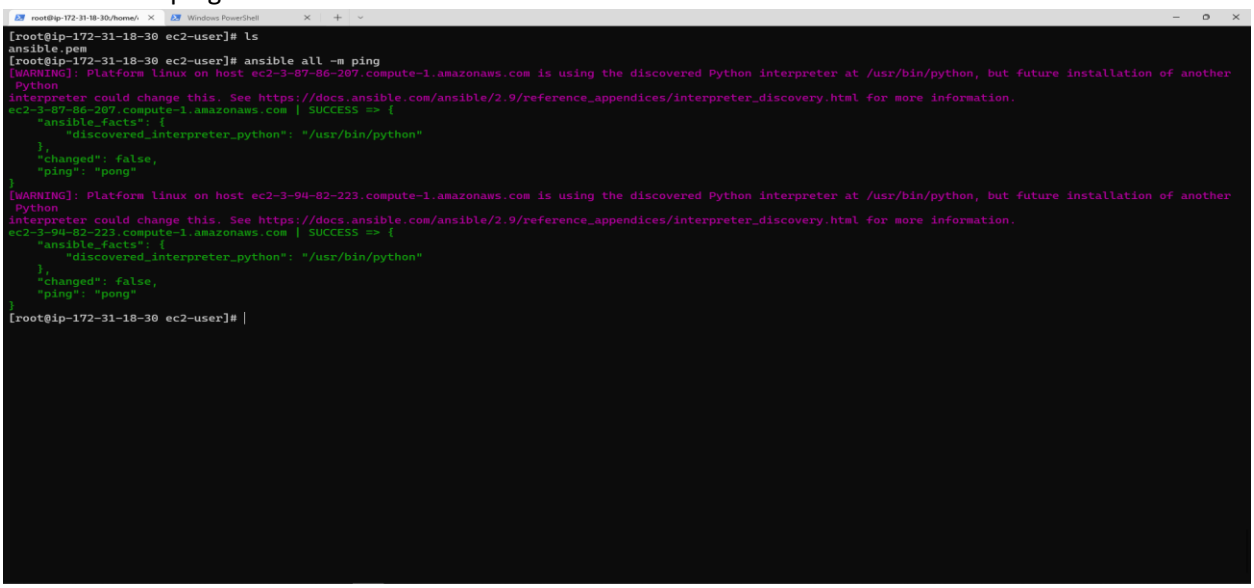


- Create file for your private key in /etc/ansible and give it permission chmod 400 example.pem -> paste your private key inside and save -> navigate back to vi /etc/ansible/ansible.cfg and do the following
- Uncomment private_key_file and provide the path to the key file
- 

- Scroll down and uncomment everything under escalate privilege_escalation



step 8: testing connectivity

- Let’s run
- Ansible all -m ping
- 
```
[root@ip-172-31-18-30 ec2-user]# ls
ansible.pem
[root@ip-172-31-18-30 ec2-user]# ansible all -m ping
[WARNING]: Platform linux on host ec2-3-87-86-207.compute-1.amazonaws.com 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/reference_appendices/interpreter_discovery.html for more information.
ec2-3-87-86-207.compute-1.amazonaws.com | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host ec2-3-94-82-223.compute-1.amazonaws.com 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/reference_appendices/interpreter_discovery.html for more information.
ec2-3-94-82-223.compute-1.amazonaws.com | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[root@ip-172-31-18-30 ec2-user]#
```
- We can now ssh into our servers

  N:B

  As a best practice always secure your private key with ansible vault