**Assessment 22 – Ansible-2**

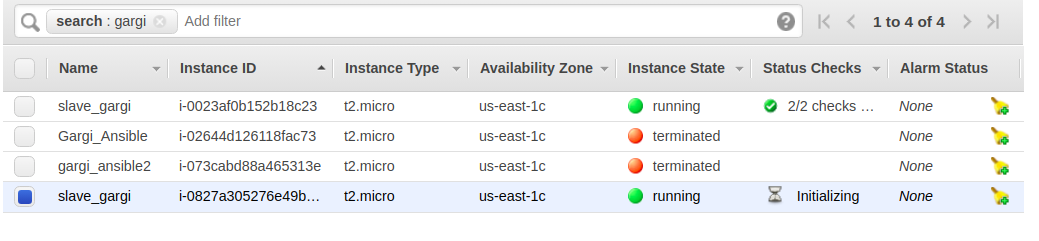
**Trainee Name : Gargi Sharma**

**Mentor Name : Mr. Akansh Gupta**

**College Name : UPES**

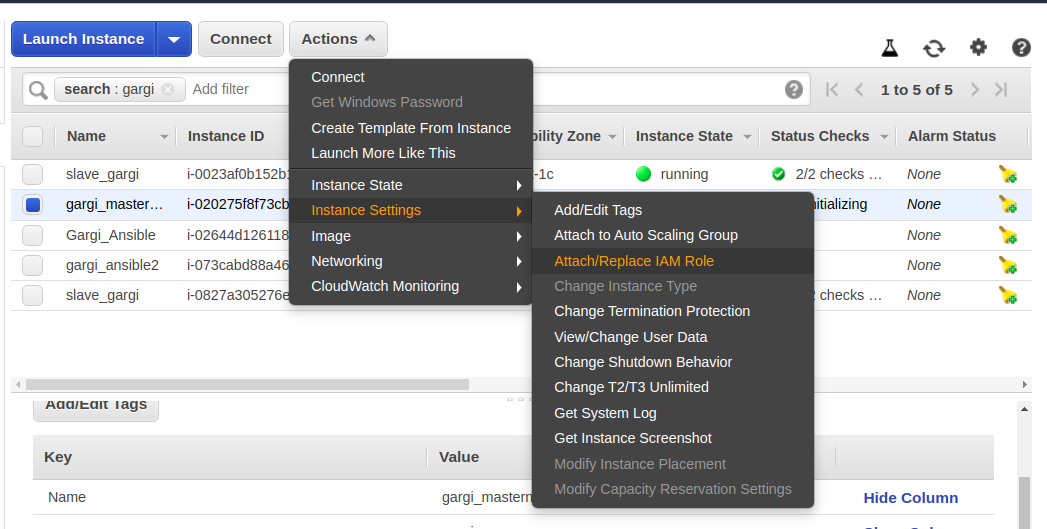
1. Create two nodes with tag:key role and tag:value master & slave respectively. Setup the dynamic inventory on ansible control nodes

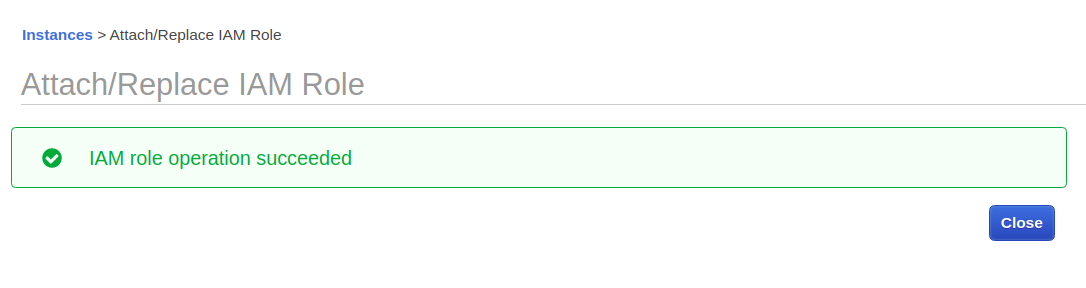
Create two instances with the following tags: master\_gargi and slave\_gargi



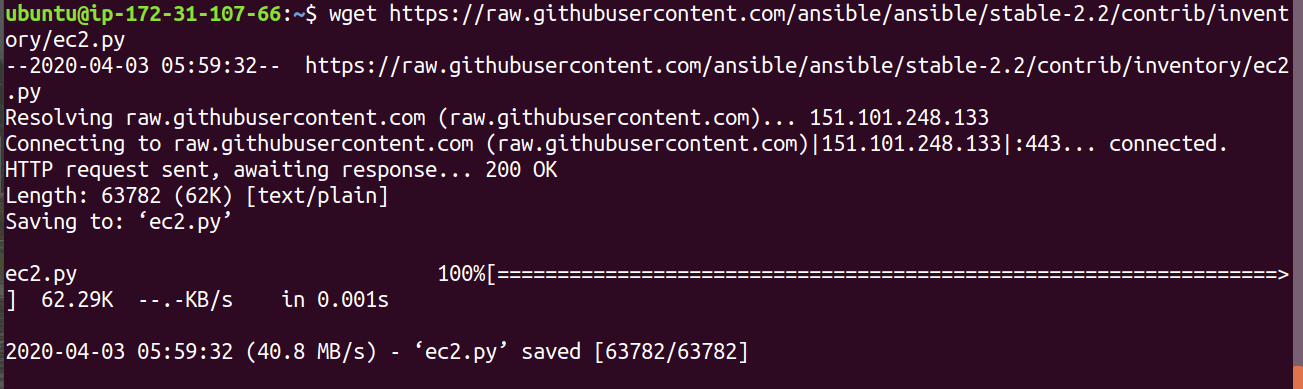
Create a control node(master) and attach ec2 full access role to it(You need to create the role first and then attach it).

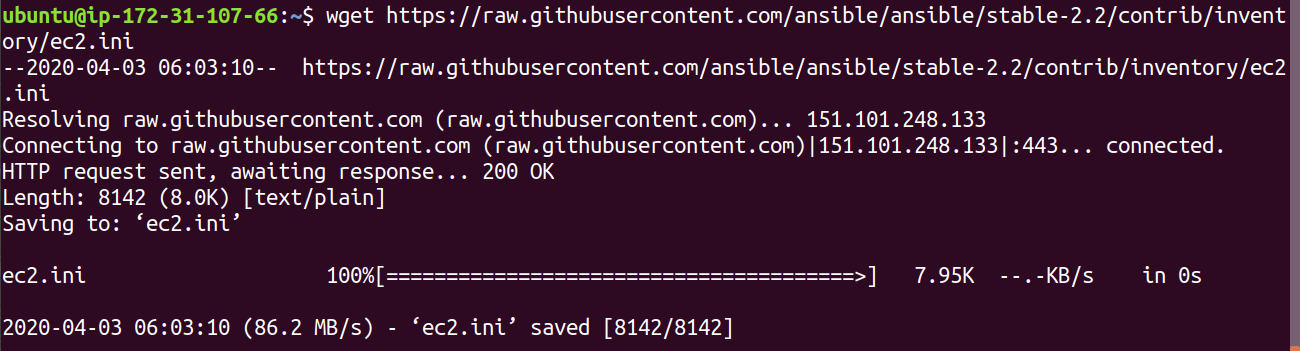
Ec2 full access role is needed for accessing other nodes without the need of access and secret keys.



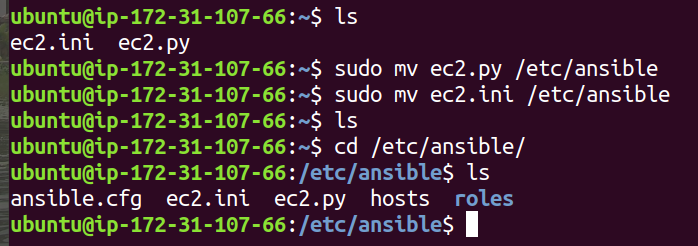


Now download [ec2.py](https://raw.githubusercontent.com/ansible/ansible/stable-2.2/contrib/inventory/ec2.py) and [ec2.ini](https://raw.githubusercontent.com/ansible/ansible/stable-2.2/contrib/inventory/ec2.ini) file in the control node.

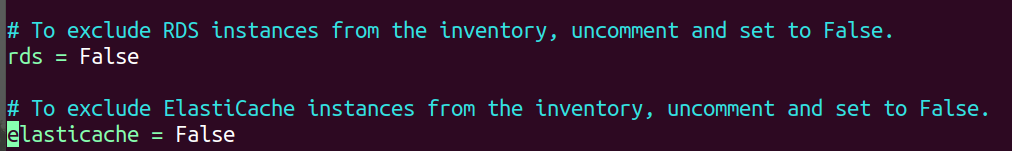




Now move both the files to /etc/ansible directory



Set rds and elasticache to false in ec2.ini



Install boto using the following commands:

sudo apt-get install python-pip

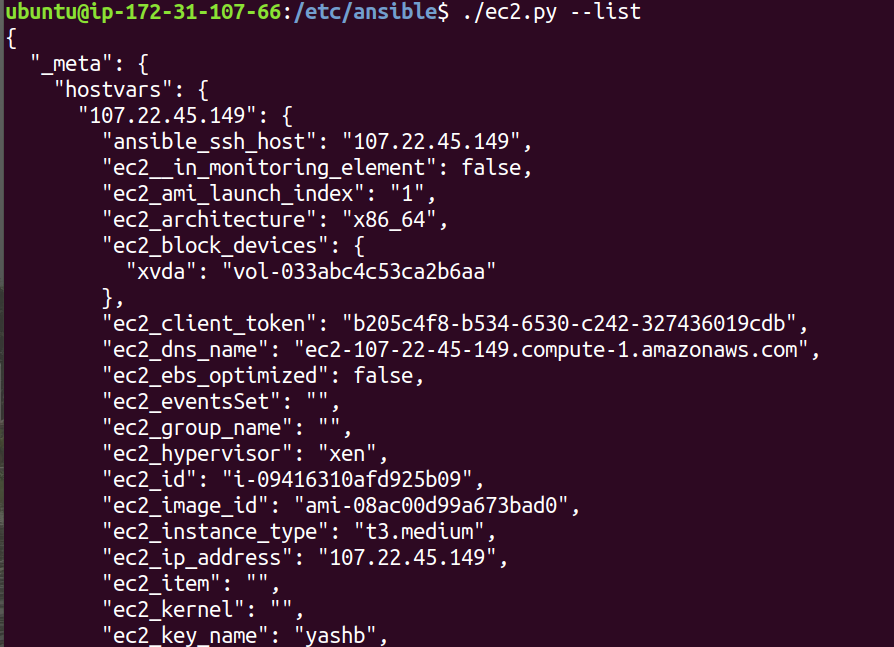
pip install -U boto

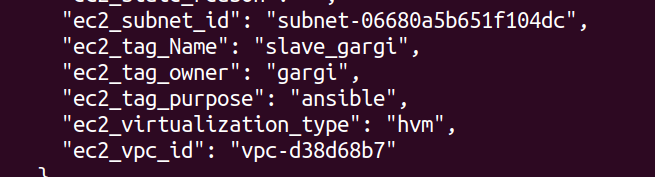
Boto is required to run ec2.py script.

Make ec2.py executable and run the script. You will see the following as the output.

**Chmod +x ec.py**

**./ec2.py --list**





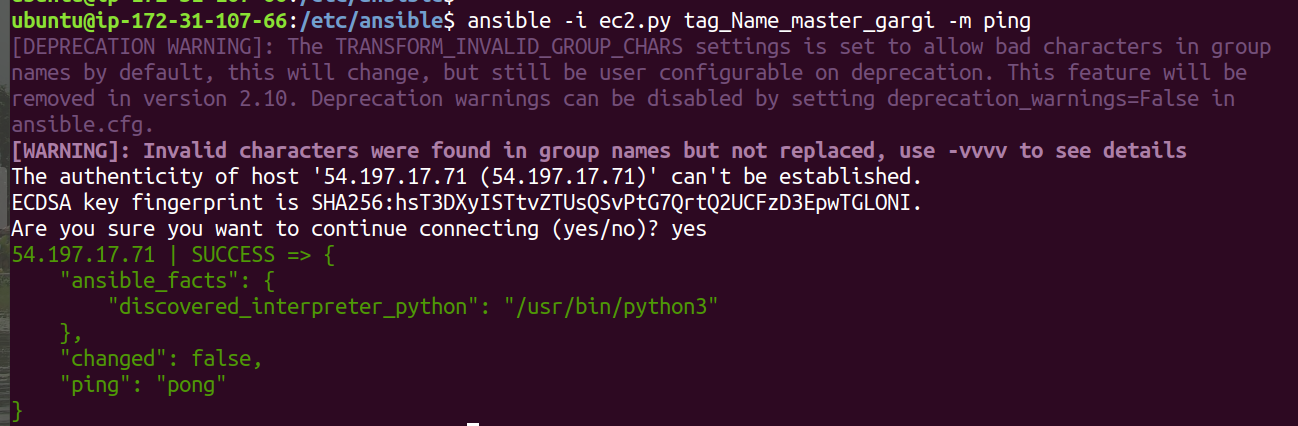
The setup is complete.

Run the following modules using tag key-value:

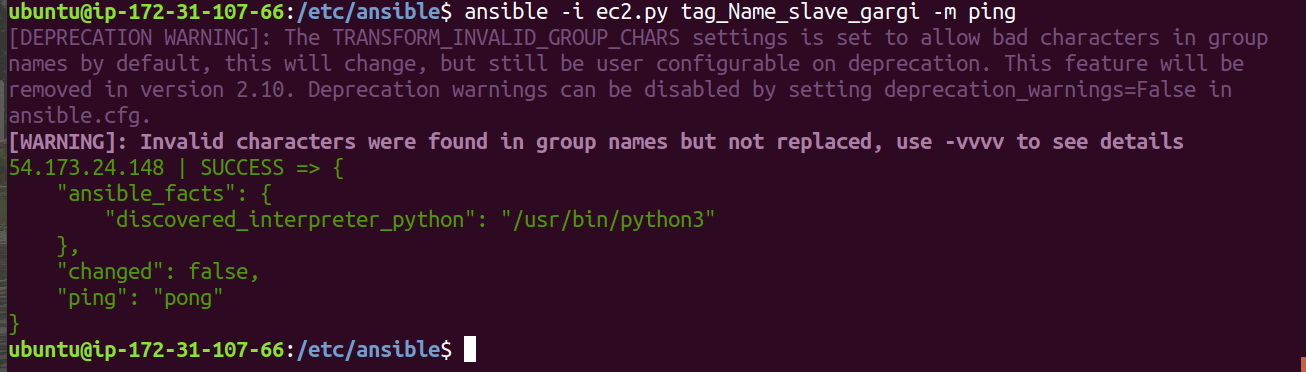
1.1 Ping master node and slave node separately.

**ansible -i ec2.py tag\_Name\_master\_gargi -m ping**

Master node:

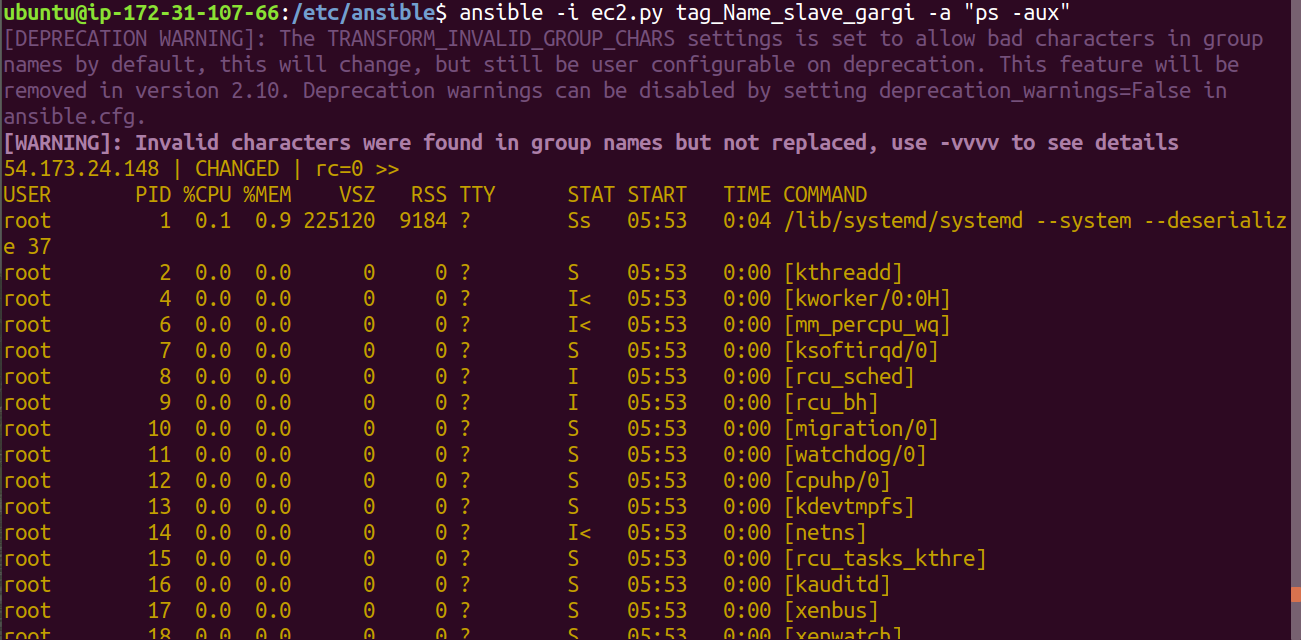


Slave node:



1.2 To check all running processes on the slave node.

**ansible -i ec2.py tag\_Name\_slave\_gargi -a “ps -aux”**

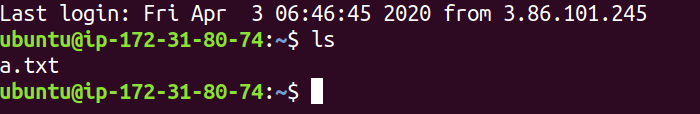
****

1.3 To copying files to both nodes concurrently.

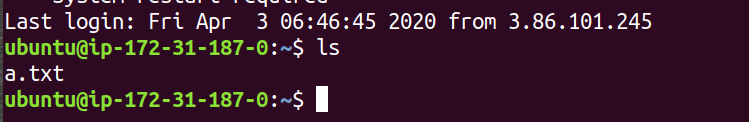
**ansible -i ec2.py tag\_Name\_\*\_gargi -m copy -a “src=/etc/ansible/a.txt dest=/home/ubuntu/”**



On master:



On slave:

****

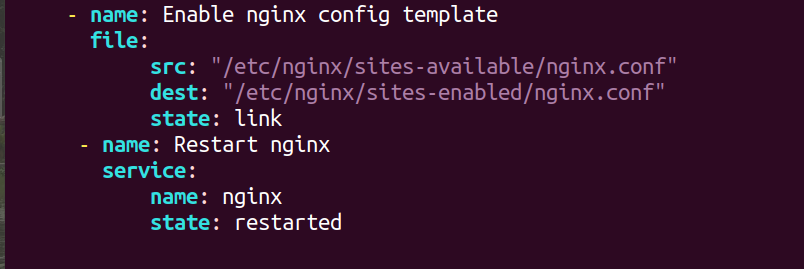
1. Setup nginx on both nodes with a single custom configuration template, on master nginx should run on 8000 while on slave nginx would listen on port 80. [Jinja2+conditional]

Set up server configuration file using jinja2 and conditionals

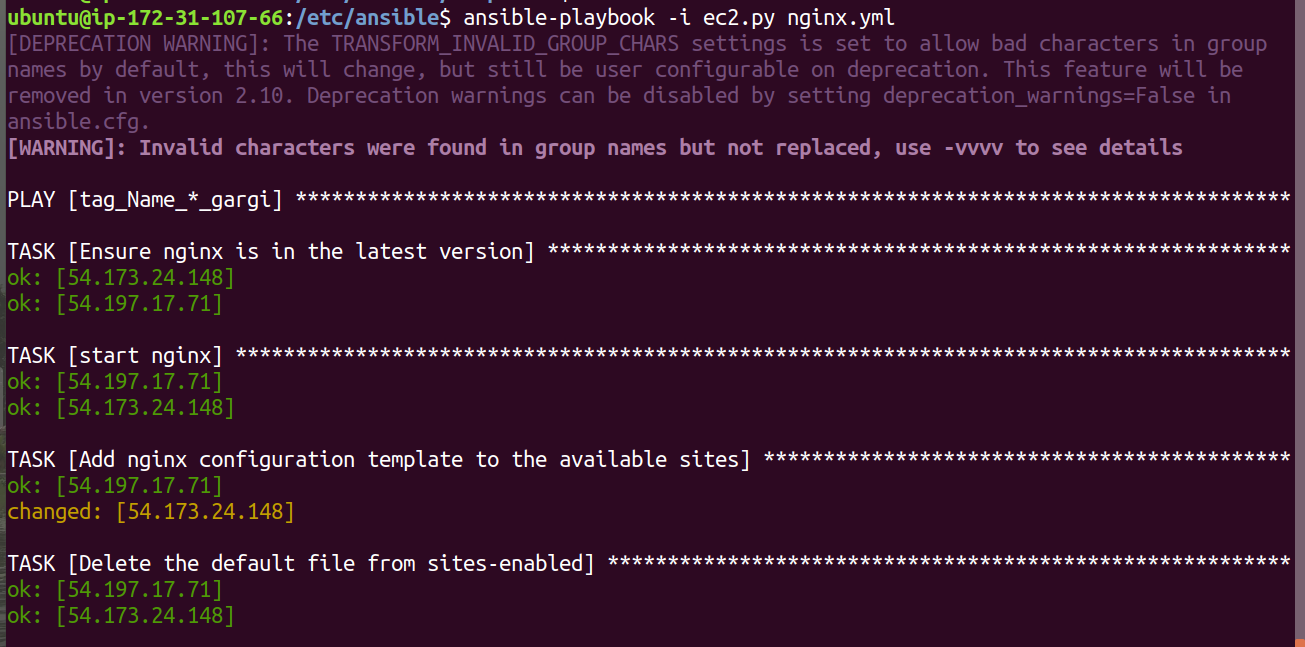


Now create an ansible playbook to run nginx and copy the configuration file.



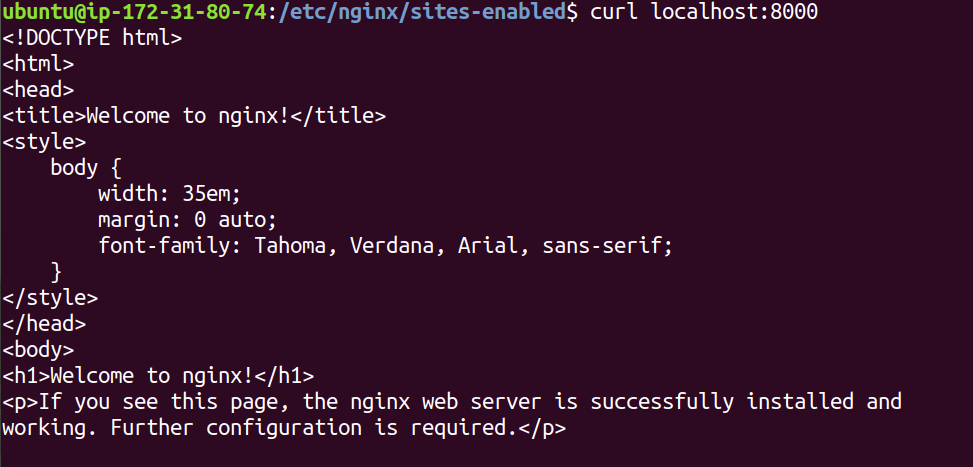


Run ansible playbook nginx.yml

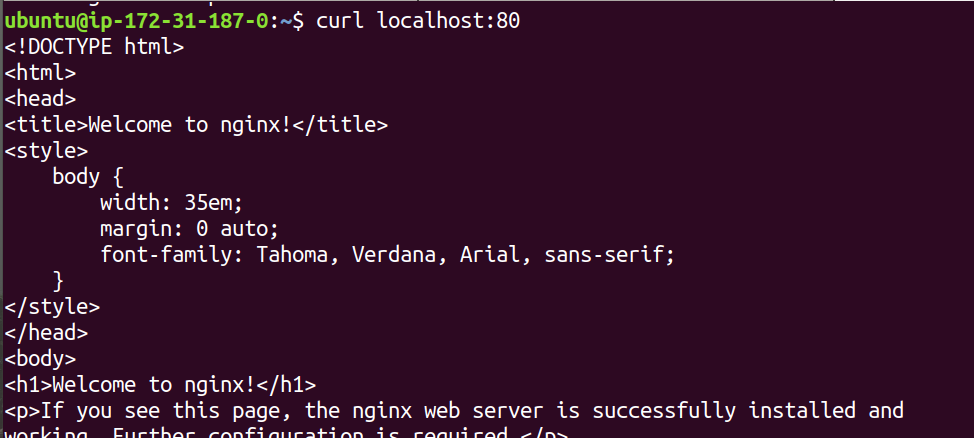


Now curl nginx url on both the nodes.

On master:



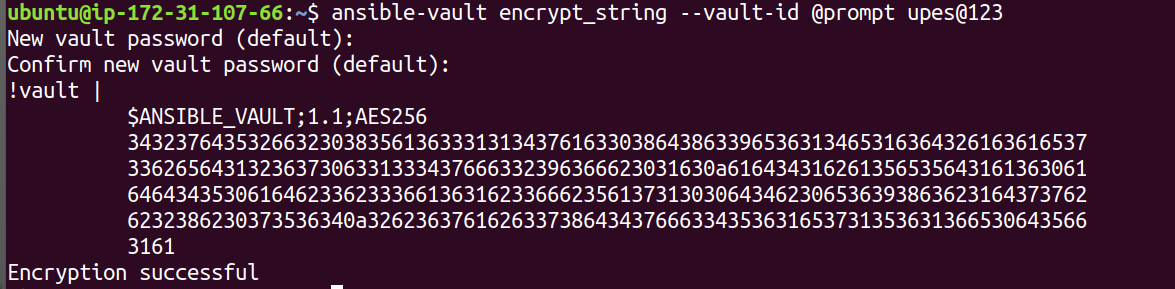
On slave:



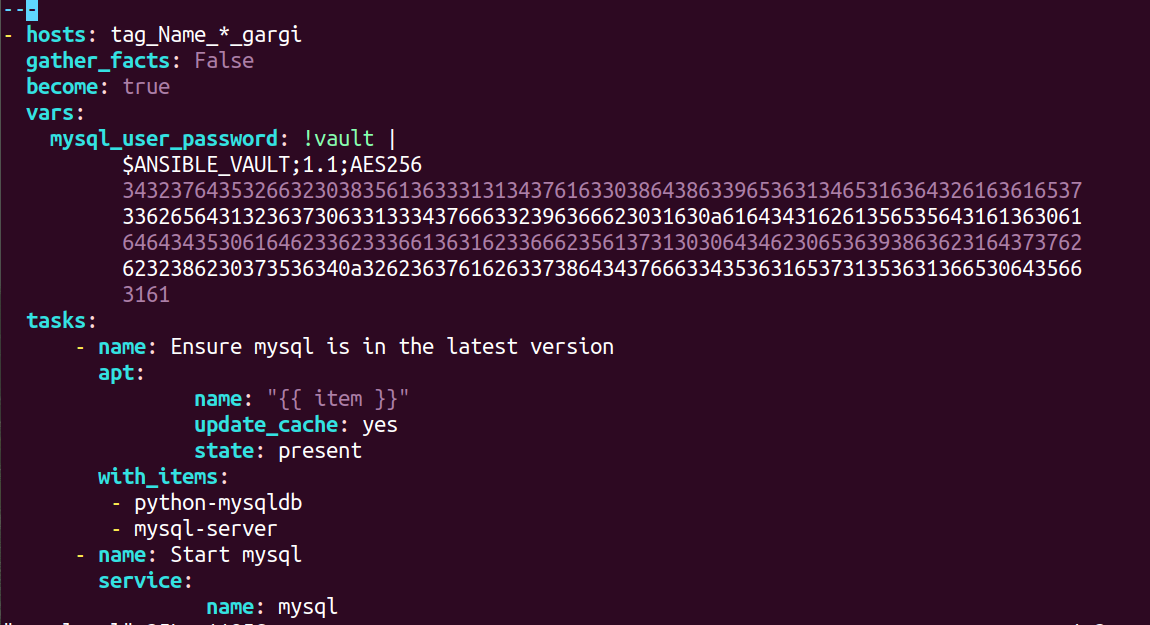
1. Setup mysql on a remote server, create a user with password. Passwords should be encrypted using Ansible vault. Verify the setup by log in to mysql.

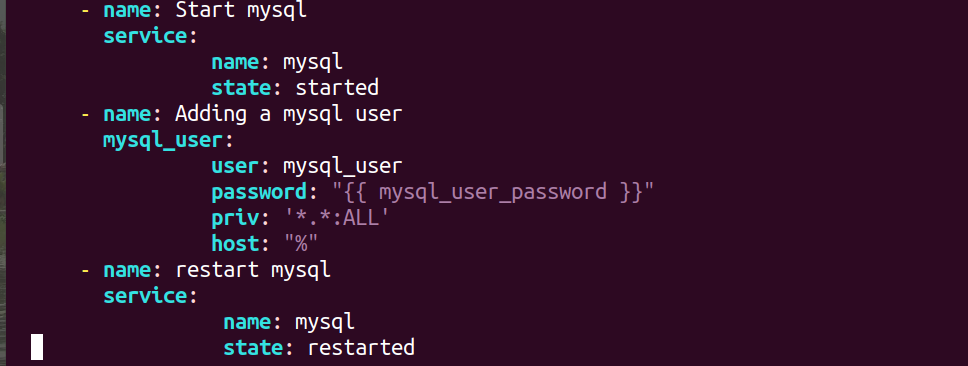
Encrypt the mysql login password using ansible vault.

**ansible-vault encrypt\_string --vault-id @prompt upes@123**



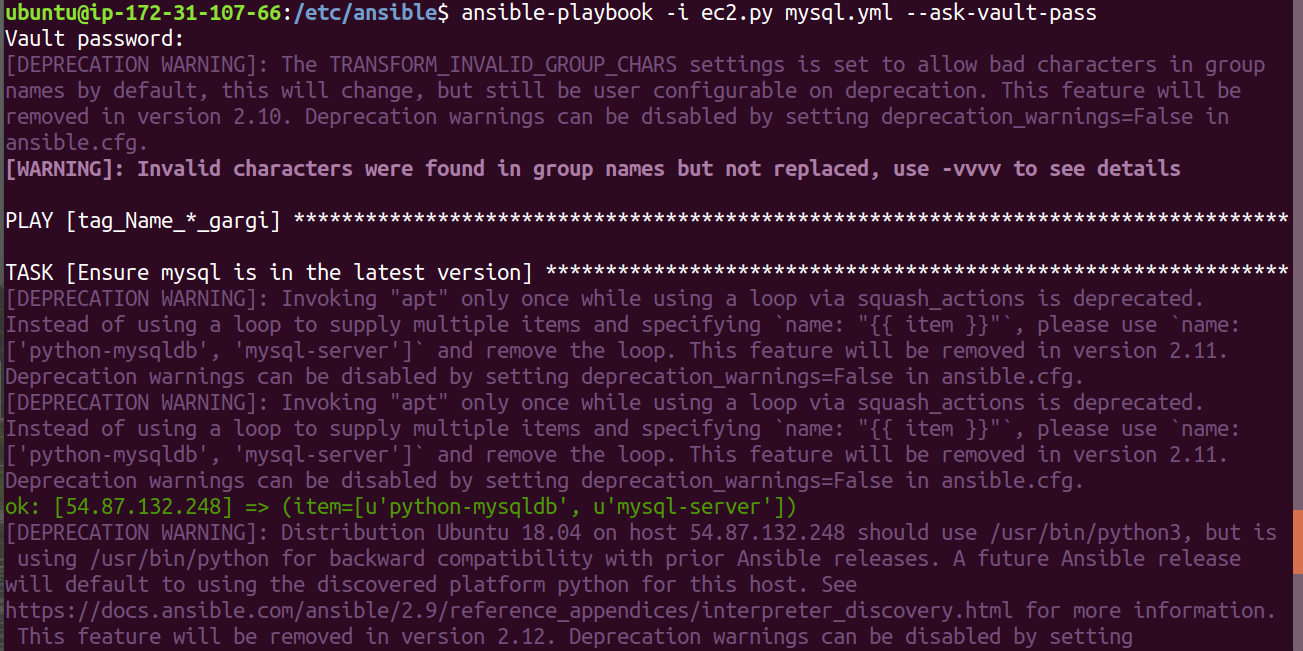
Create a playbook mysql.yml





Now run the playbook.

**ansible-playbook -i ec2.py mysql.yml --ask-vault-pass**

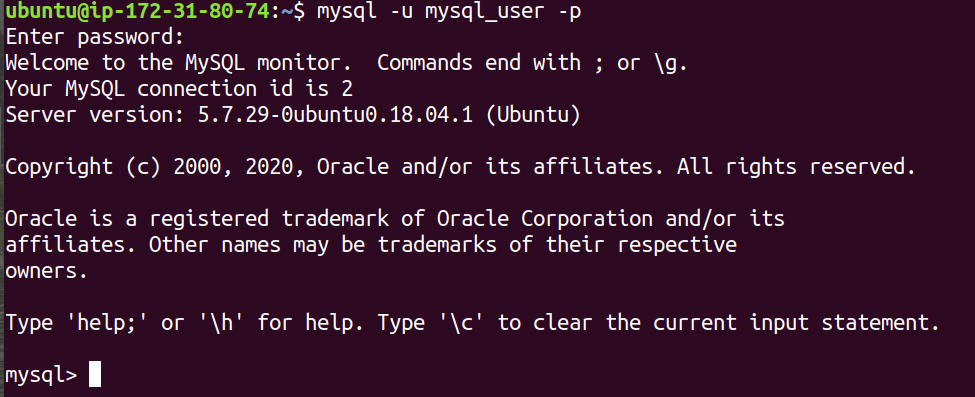




Now ssh to the remote servers and check mysql login with the newly created user and password.

**mysql -u mysql\_user -p**

On master:



On slave:

