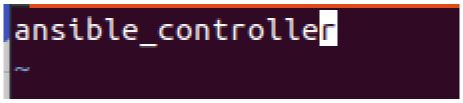
ANSIBLE- https://docs.ansible.com/

Ansible is an agentless automation tool that you install on a single host (referred to as the control node). From the control node, Ansible can manage an entire fleet of machines and other devices (referred to as managed nodes) remotely with SSH

# Change the name of you host to recognise on which server we are working

```
Sudo vim /etc/hostname
```

Remove everything & add ***ansible_controller***

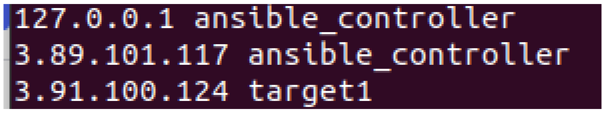


```
ansible_controller
~
```

```
sudo reboot
```

*# Now edit hosts name file to add target servers ip and name on **ansible_controller** & **on targets** as well.*

```
Sudo vim /etc/hosts
```



```
127.0.0.1 ansible_controller
3.89.101.117 ansible_controller
3.91.100.124 target1
```

# Now to enable ssh access from ansible controller to target machines, we need to generate ssh keys for controller server and then to add the same generated key to authorized_keys of target hosts.

Ansible_controller server

```
ssh-keygen
```

# copy id_rsa.pub to target server authorized_keys and confirm ssh access from ansible controller

```
ssh ubuntu@<target_ip_address>
```

Ansible Installation on ansible controller server*:-*

https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-on-ubuntu

```
sudo apt update -y
sudo apt install software-properties-common -y
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible -y
ansible --version
```

Ansible ssh and password connection method:-
https://docs.ansible.com/ansible/latest/user_guide/connection_details.html
Ansible Adding Hosts in inventory file:-
https://www.bogotobogo.com/DevOps/Ansible/Ansible-SSH-Connection-Setup-Run-Command.php

# create inventory list in ansible_controller

```
mkdir project
cd project
```

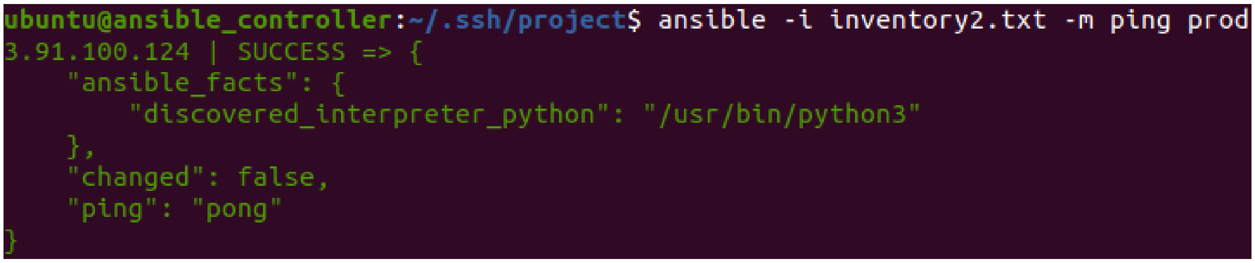
https://www.bogotobogo.com/DevOps/Ansible/Ansible-SSH-Connection-Setup-Run-Command.php

```
#!/bin/bash
sudo apt update
sudo su -
adduser ansible
cd /home/ansible
mkdir .ssh
cd .ssh
echo " " >> /home/ansible/.ssh/authorized_keys
echo "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDIDVSnl4v1XZ0RfBzs70u8e1mDGXlwM79rv3H9oMQtyAOeKSWc
fN4z0rpfUsM5T5esvbrSP3E8FFoHDX1/aXc97U1O104PLMoDab8jbklRUm02J1RQFkLLYHHSuk5x69+Qn/GB
BHfbRigOPHn8I75PBWCMyMTHhklzbsHAa/C9HbE0/aVuA6MGNEWarFDXtuEOnEDcinr/qm9ypoxRQeXH2Lr
Y9f702Cnk4tpkOxb4V6nCJgCBTu6pJKVRf01n5hutsWB50FVOgLOPGBJ5QddJk+JUW9VCcqEF9SPWim0Qz
PXWWoAObU9l6EgMcuHJBbyDqIw8G+6dIjiqTVPn9BEL ubuntu@ip-172-31-56-169" >>
/home/ubuntu/.ssh/authorized_keys
chown ansible:ansible /home/ansible/.ssh/authorized_keys
cat /home/ansible/.ssh/authorized_keys
```

EXAMPLE 1:-

```
sudo vim inventory2.txt                    #add below in inventory2.txt file
[prod]
3.91.100.124     ansible_connection=ssh     ansible_ssh_private_key_file=/home/ubuntu/.ssh/controller
 #IP of target       #ansible connection method                                #which key to be used to ssh
[dev]
4.91.100.1       ansible_connection=ssh     ansible_ssh_private_key_file=/home/ubuntu/.ssh/controller
4.91.101.2       ansible_connection=ssh     ansible_ssh_private_key_file=/home/ubuntu/.ssh/controller
#IP of target       #ansible connection method                                 #which key to be used to ssh
```

ansible -i inventory2.txt -m ping prod # this command will ping the server under **prod** in the inventory2 file.



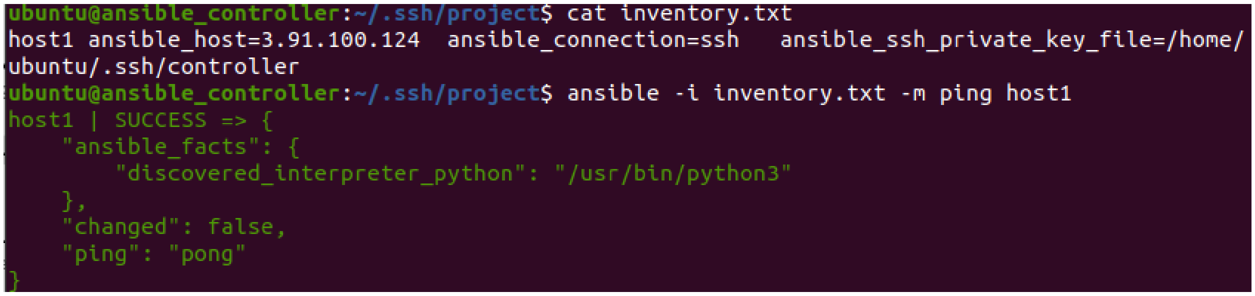
```
ubuntu@ansible_controller:~/.ssh/project$ ansible -i inventory2.txt -m ping prod
3.91.100.124 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

ansible -i inventory2.txt -m ping dev # this command will ping the server under **dev** in the inventory2 file.
ansible -i inventory2.txt -m ping all   # this command will ping **all** the servers under **prod & dev** inventory2.

EXAMPLE 2:-

```
sudo vim inventory.txt            #add below in inventory.txt file
host1 ansible_host=3.91.100.124  ansible_connection=ssh ansible_ssh_private_key_file=/home/ubuntu/.ssh/controller
```

ansible -i inventory.txt -m ping host1 # this command will ping the server under **prod** in the inventory file.

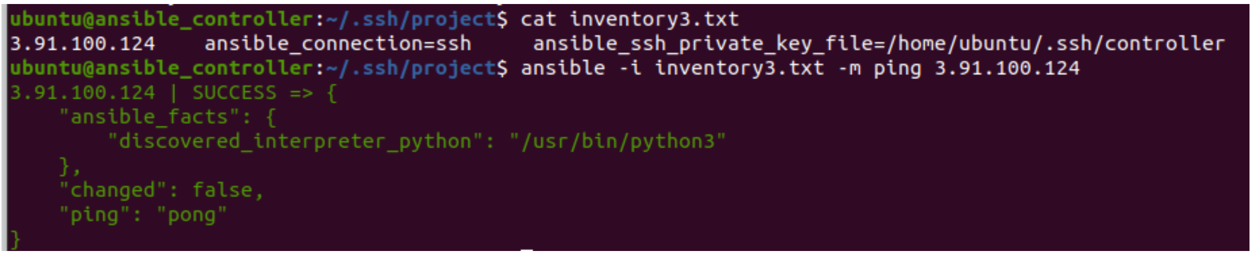


```
ubuntu@ansible_controller:~/.ssh/project$ cat inventory.txt
host1 ansible_host=3.91.100.124  ansible_connection=ssh   ansible_ssh_private_key_file=/home/
ubuntu/.ssh/controller
ubuntu@ansible_controller:~/.ssh/project$ ansible -i inventory.txt -m ping host1
host1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

EXAMPLE 3:-

sudo vim inventory3.txt #add below in inventory3.txt file

3.91.100.124 ansible_connection=ssh ansible_ssh_private_key_file=/home/ubuntu/.ssh/controller



```
ubuntu@ansible_controller:~/.ssh/project$ cat inventory3.txt
3.91.100.124    ansible_connection=ssh     ansible_ssh_private_key_file=/home/ubuntu/.ssh/controller
ubuntu@ansible_controller:~/.ssh/project$ ansible -i inventory3.txt -m ping 3.91.100.124
3.91.100.124 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

EXAMPLE 4:- IF ssh is setup with username and password

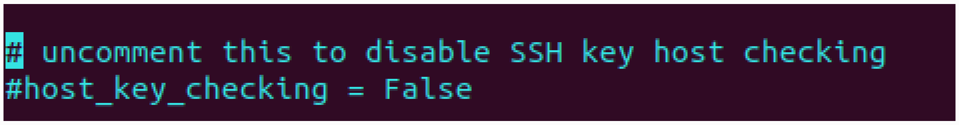
sudo vim inventory4.txt #add below in inventory4.txt file

host1 ansible_host=3.91.100.124 ansible_ssh_pass=<password> ansible_user=<username>

By default, Ansible uses username of ansible controller in our case its **ubuntu**

# Like if we ssh to a new instance it ask for to add key fingerprint, to disable it we need to below changes:-

sudo vim /etc/ansible/ansible.conf



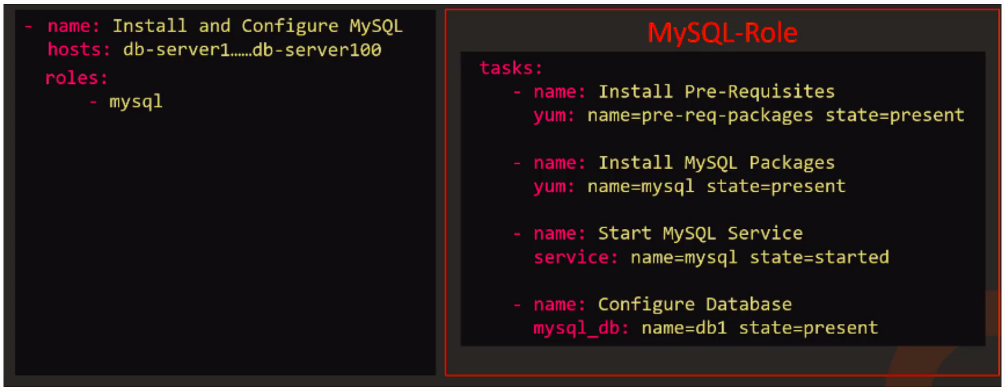
```
# uncomment this to disable SSH key host checking
#host_key_checking = False
```

It will be under the **[Default]** Section

Just uncomment it and save the file.

ANSIBLE ROLES

https://galaxy.ansible.com/search?deprecated=false&tags=database&keywords=mysql&order_by=-relevance&page=1



```
- name: Install and Configure MySQL
  hosts: db-server1……db-server100
  roles:
      - mysql
```

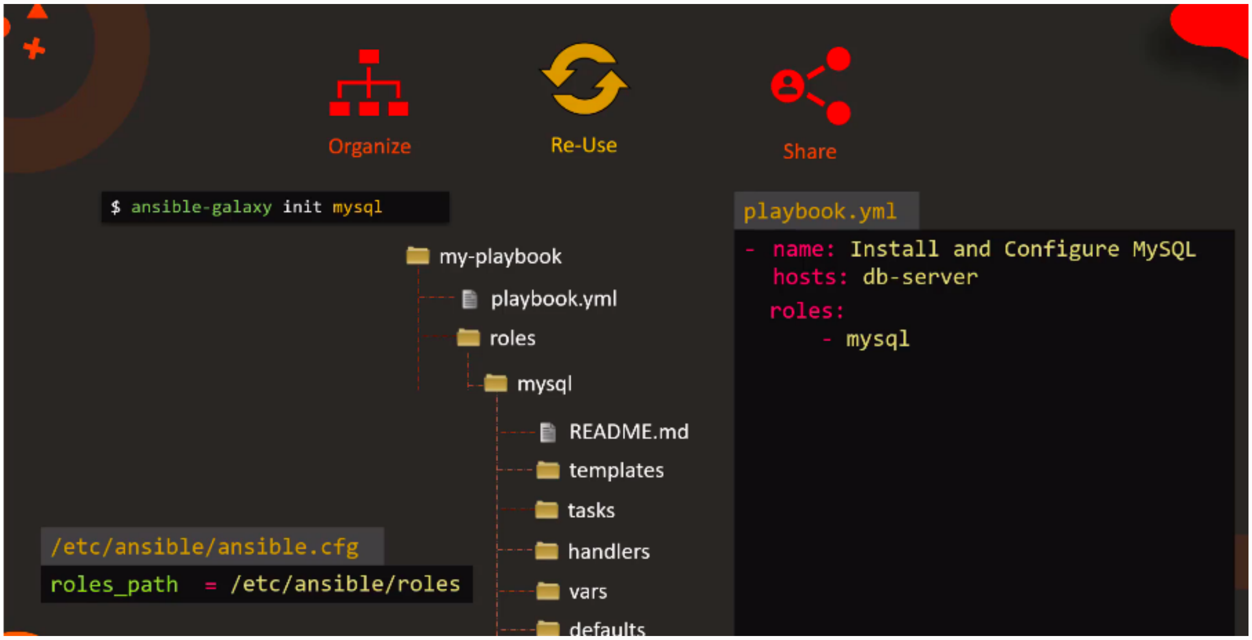
MySQL-Role

```
tasks:
   - name: Install Pre-Requisites
     yum: name=pre-req-packages state=present

   - name: Install MySQL Packages
     yum: name=mysql state=present

   - name: Start MySQL Service
     service: name=mysql state=started

   - name: Configure Database
     mysql_db: name=db1 state=present
```



```
$ ansible-galaxy init mysql
```

```
my-playbook
    playbook.yml
    roles
        mysql
            README.md
            templates
            tasks
            handlers
            vars
            defaults
```

/etc/ansible/ansible.cfg

```
roles_path  = /etc/ansible/roles
```

playbook.yml

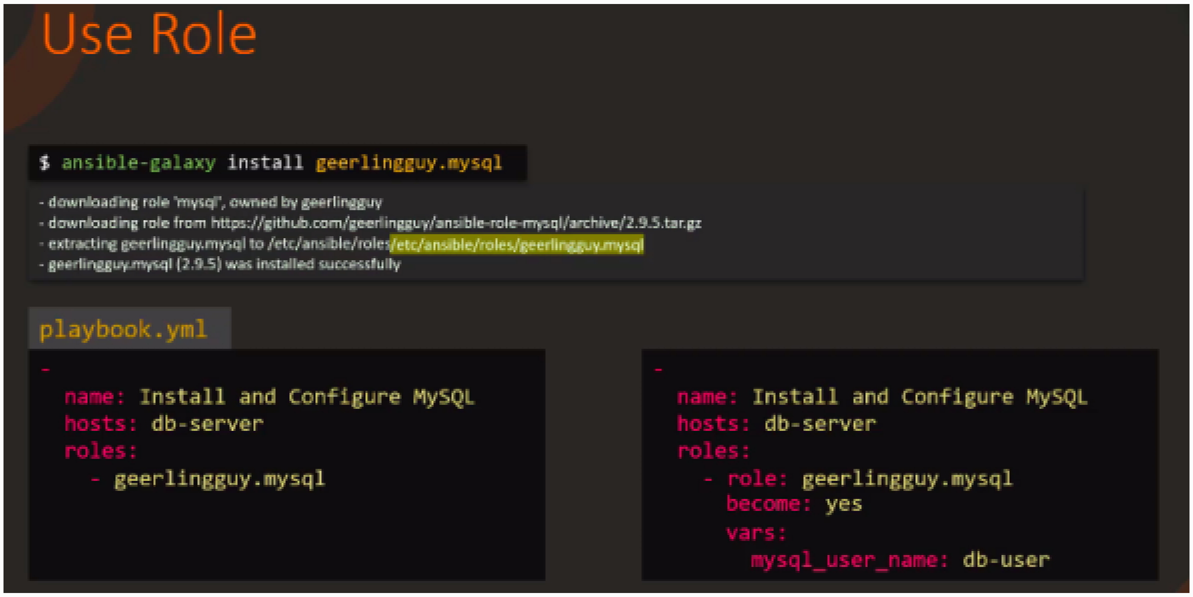
```
- name: Install and Configure MySQL
  hosts: db-server
  roles:
      - mysql
```

Roles path can be checked in ansible.conf

cat /etc/ansible/ansible.conf

Path:-

/etc/ansible/roles/



Use Role

```
$ ansible-galaxy install geerlingguy.mysql
```

```
- downloading role 'mysql', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-mysql/archive/2.9.5.tar.gz
- extracting geerlingguy.mysql to /etc/ansible/roles/etc/ansible/roles/geerlingguy.mysql
- geerlingguy.mysql (2.9.5) was installed successfully
```

playbook.yml

```
-
  name: Install and Configure MySQL
  hosts: db-server
  roles:
    - geerlingguy.mysql
```

```
-
  name: Install and Configure MySQL
  hosts: db-server
  roles:
    - role: geerlingguy.mysql
      become: yes
      vars:
        mysql_user_name: db-user
```