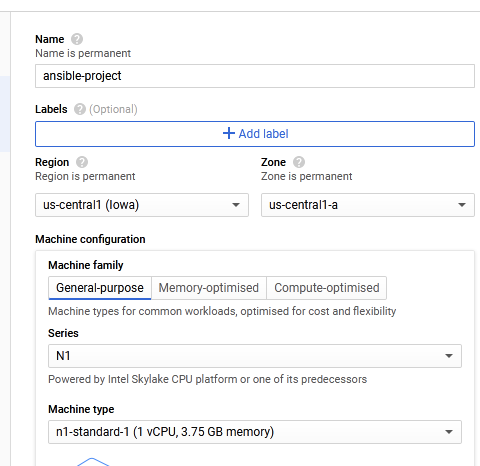
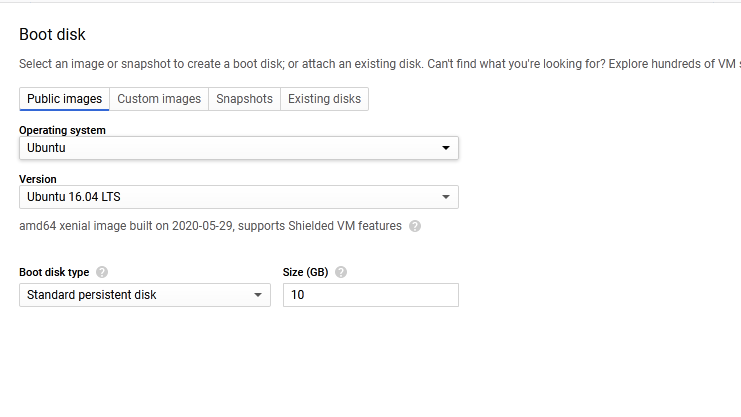
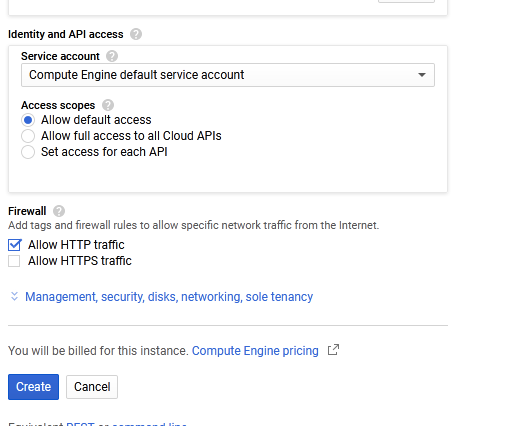
**Ansible 201 Project Case Study**

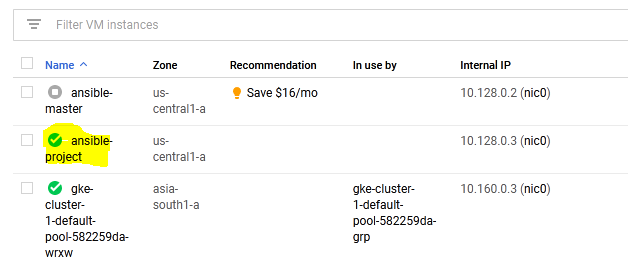
**Firstly we will be creating one linux VM instance on google cloud as follows and install ansible on it.**

****

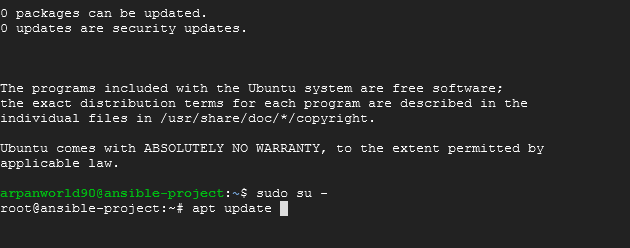
****

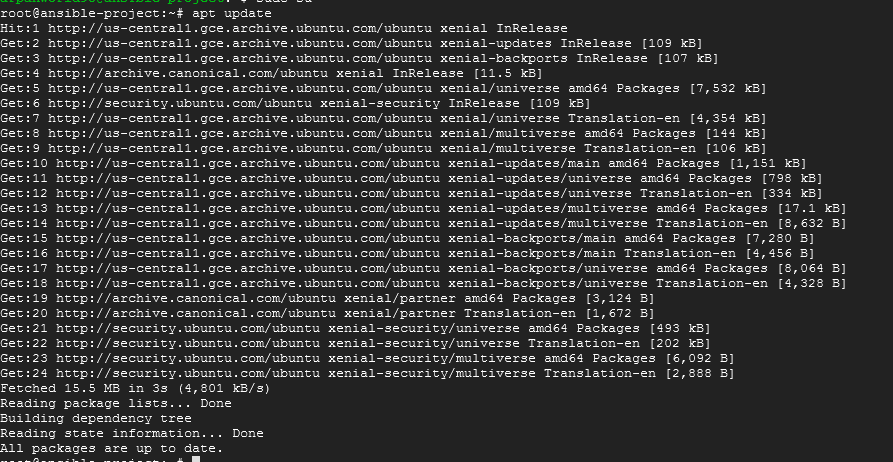
****

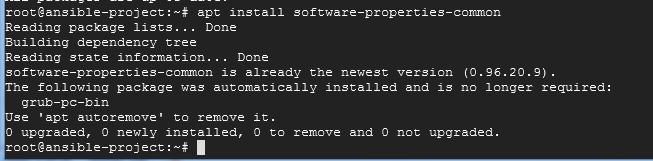
**We could see the instance created as follows**

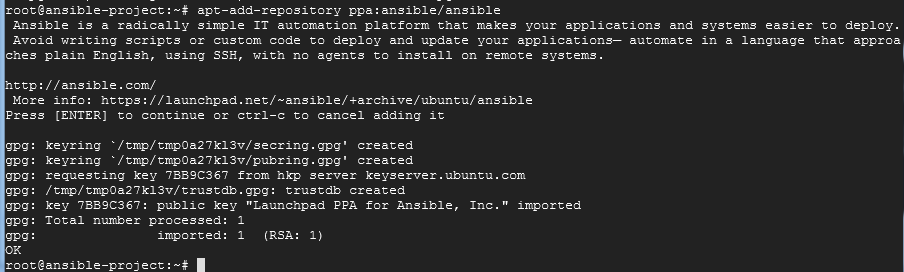
****

**Now we will be connecting to that VM instance thru ssh and will be installing ansible as follows.**

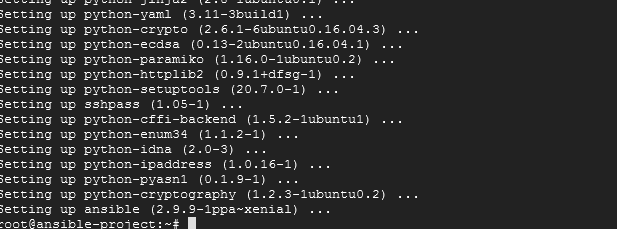
****

****

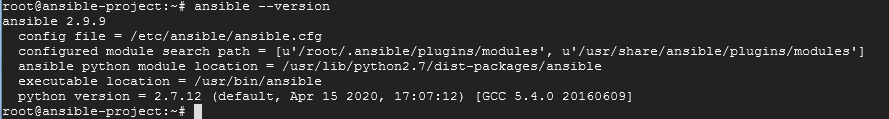
****

****

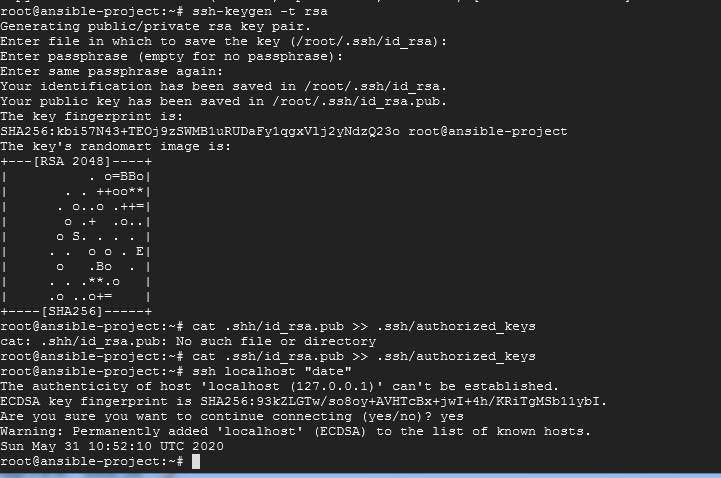
****

****

**To check whether the ansible is installed or not**

****

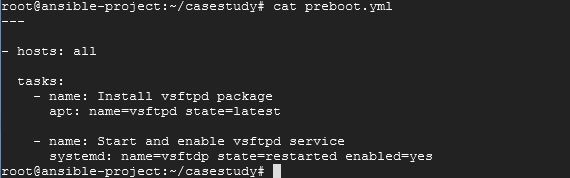
Establish SSH key pair in linux system to have SSH connectivity with localhost using following commands

****

1. **Write ansible playbook for automating the tack of doing weekly system reboots.**

**Firstly we will be writing playbook for preboot tasks as follows**

**We will be installing vsftpd package in our server as a part of preboot task**

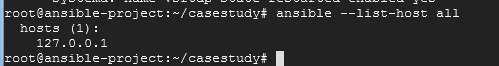
****

**Check for syntax errors**

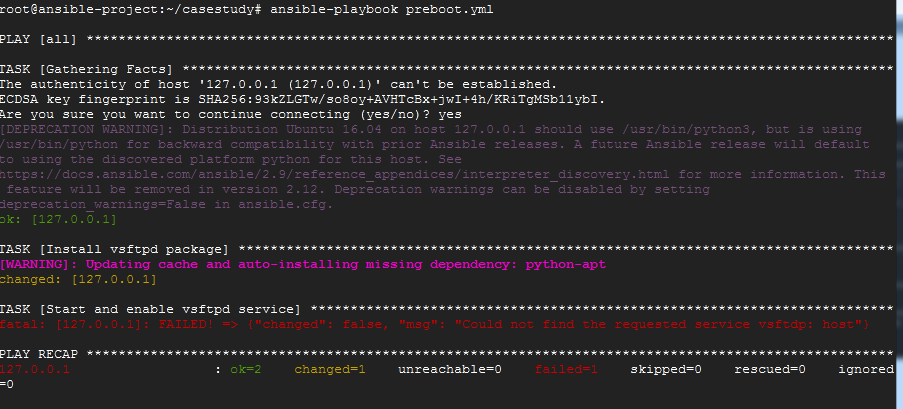
****

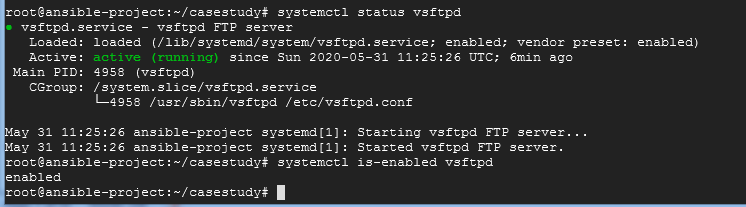
**We will also be placing the hosts ip address or the host url in /etc/ansible/hosts file.**

**Here we have created an entry or the local host in the hosts file**

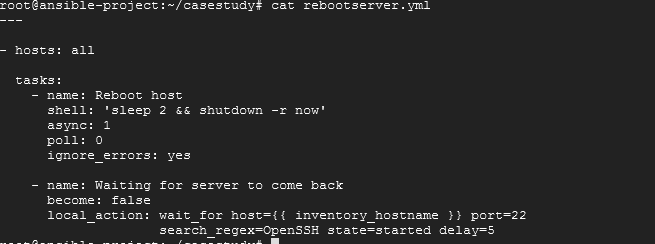
****

**Now we will be executing the playbook as follows**

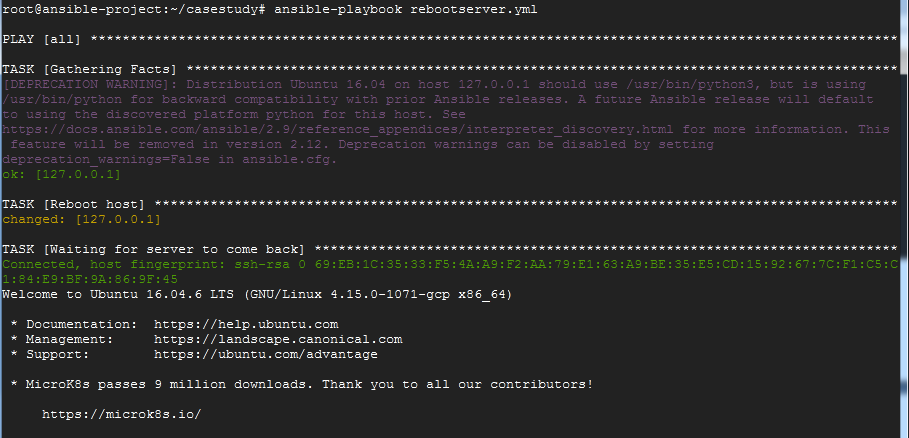
****

****

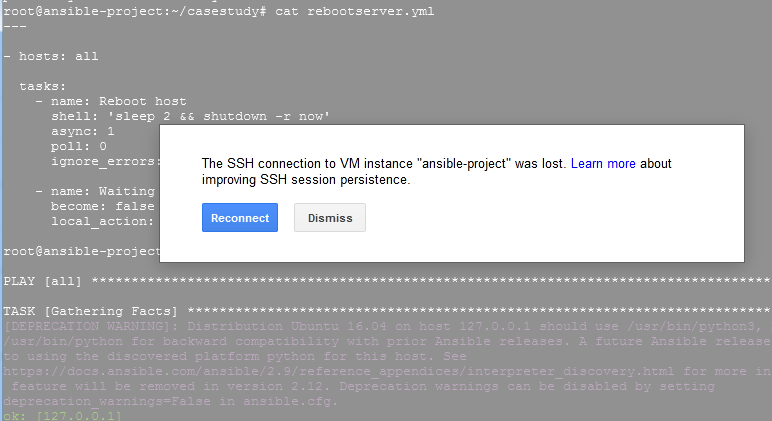
**Now we will be writing the playbook for rebooting the server as follows.**

****

**We will be running the reboot playbook now as follows**

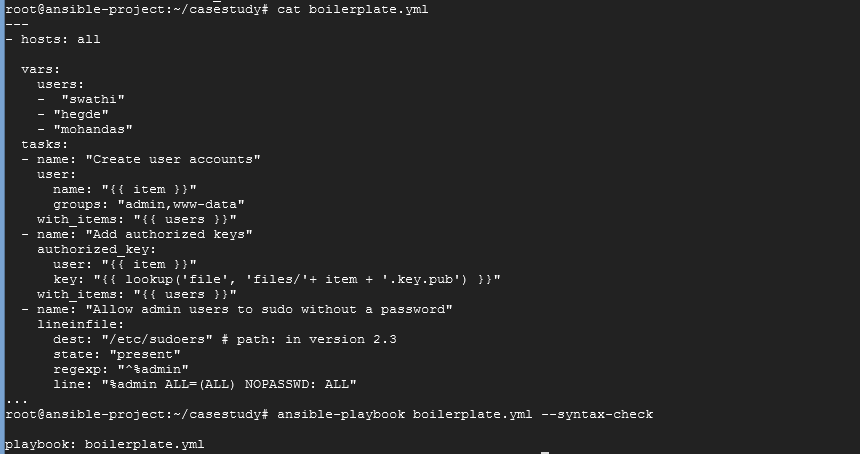
****

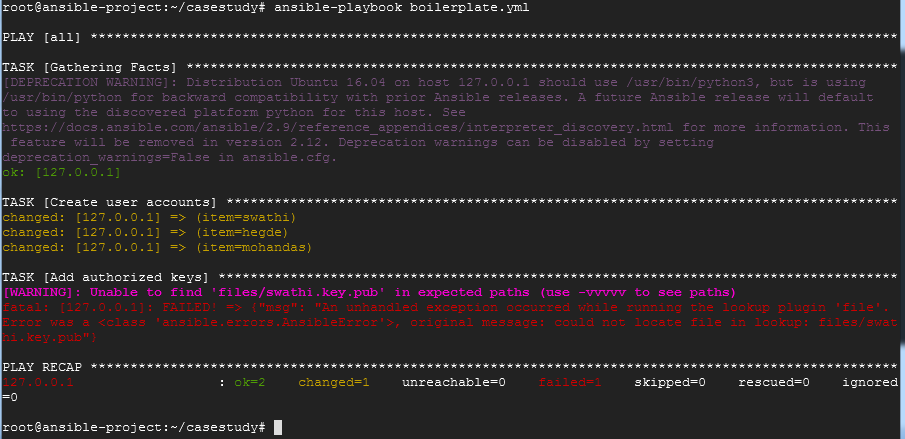
**As we are rebooting localhost it self the connection to the server will be lost for few seconds**

****

**If we add remote hosts in the /etc/ansible/hosts file the reboot will happen for that remote server as well.**

**2.Use Ansible for Enforcing security guidelines**

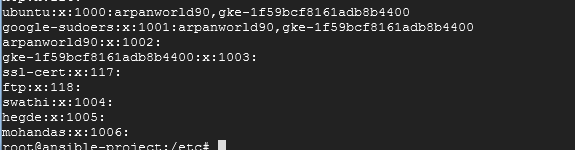
****

****

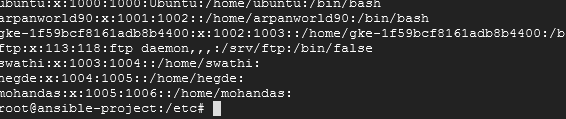
**We could see the id created and also able to login as sudo**

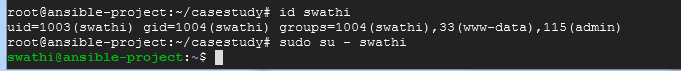
**We could see 3 users created**

**Cat /etc/group**

****

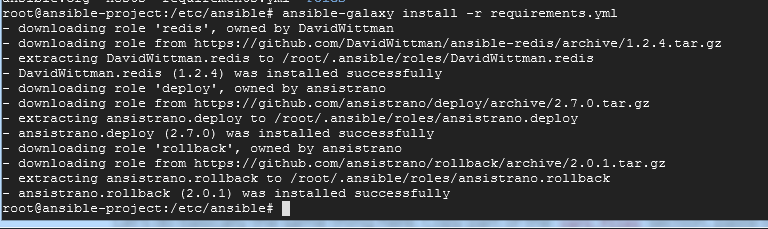
**Cat /etc/passwd**

****

****

**3.Use Ansible tasks to enforce desired configuration and immediately revert using ansible roll back.**

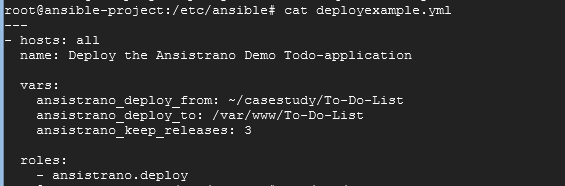
**We will be installing ansistrano for deploying and rollback as follows**

****

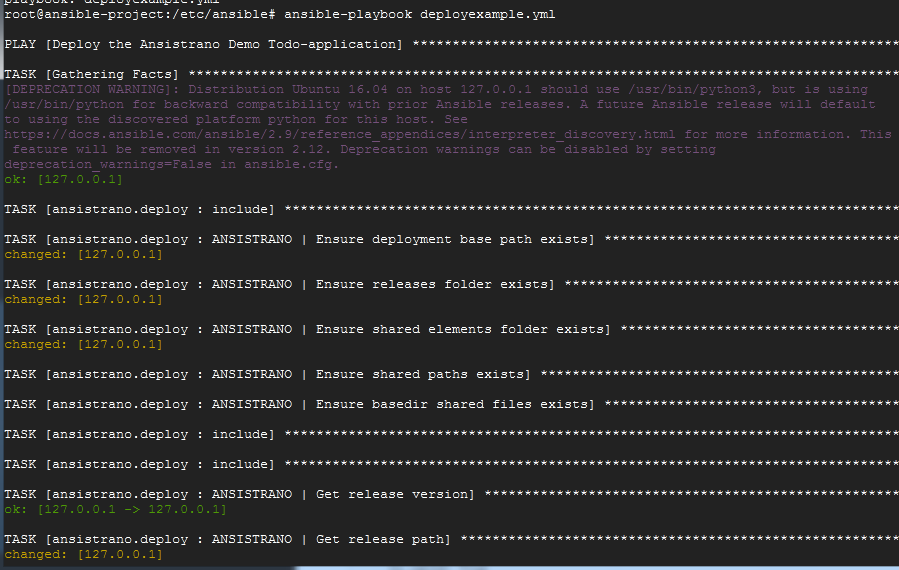
**Once the roles are installed we will be using those roles in our playbook.**

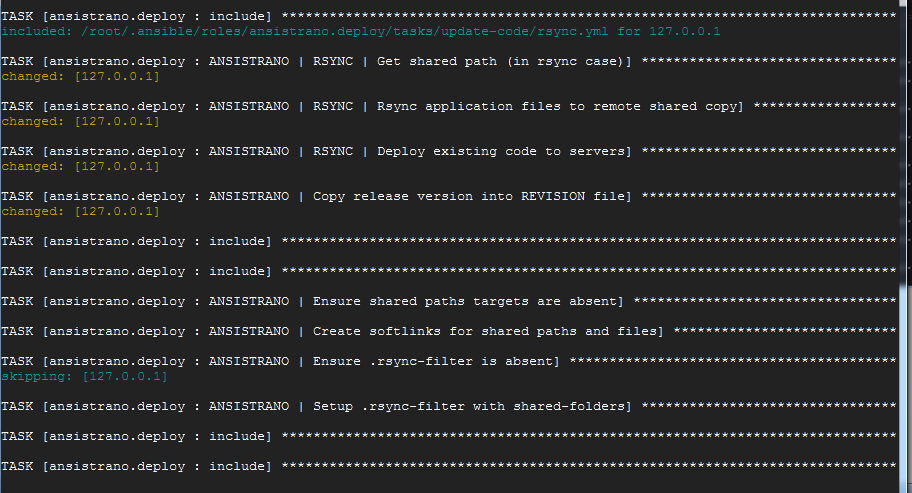
**Now we will be deploying one To-Do-List application app as follows .**

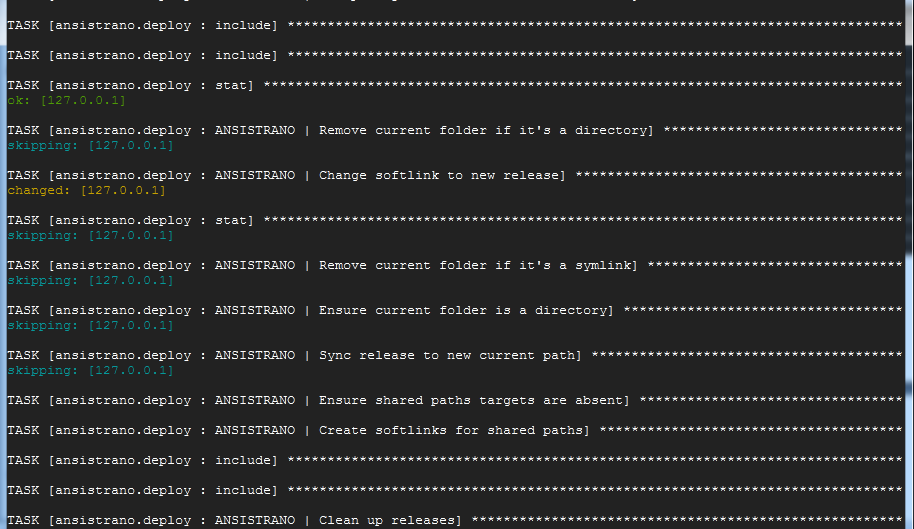
**Please find the playbook for the deploying the app**

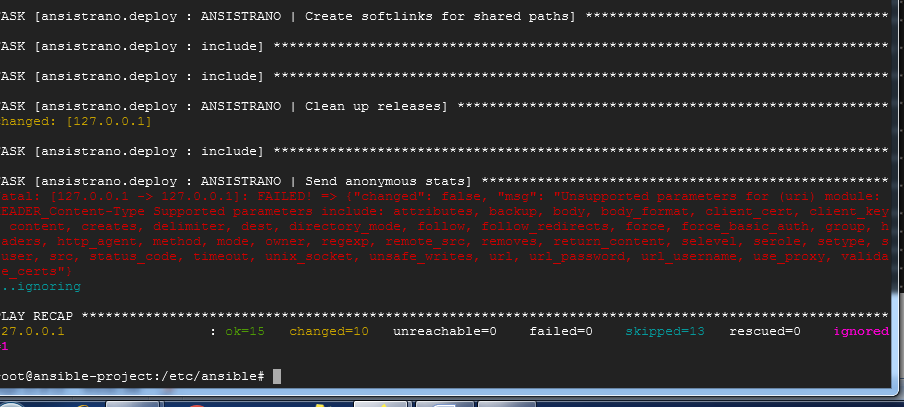
****

**Now we will be running the deployexample.yml as follows**

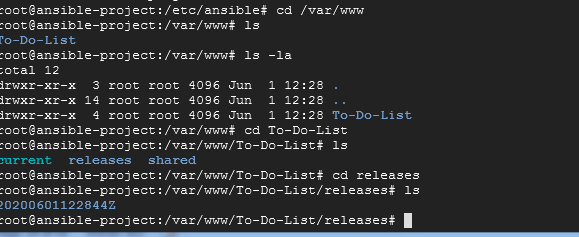
****

****

****

****

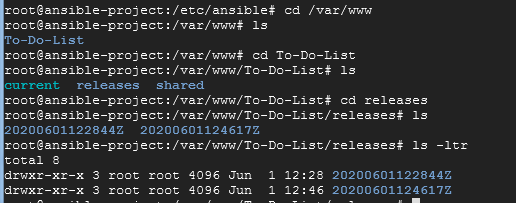
**To check whether the deployement was done successfully or not go to the /var/www the path which we have mentioned.**

****

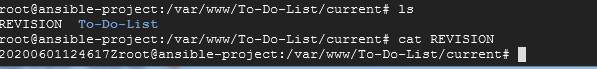
**We could see the releases.**

**Now again we will execute the deployexample.yml playbook with some changes**

**And now we could see two releases in the directory as follows.**

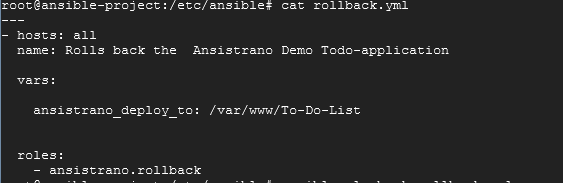
****

**And the current directory will be having the new changes deployed.**

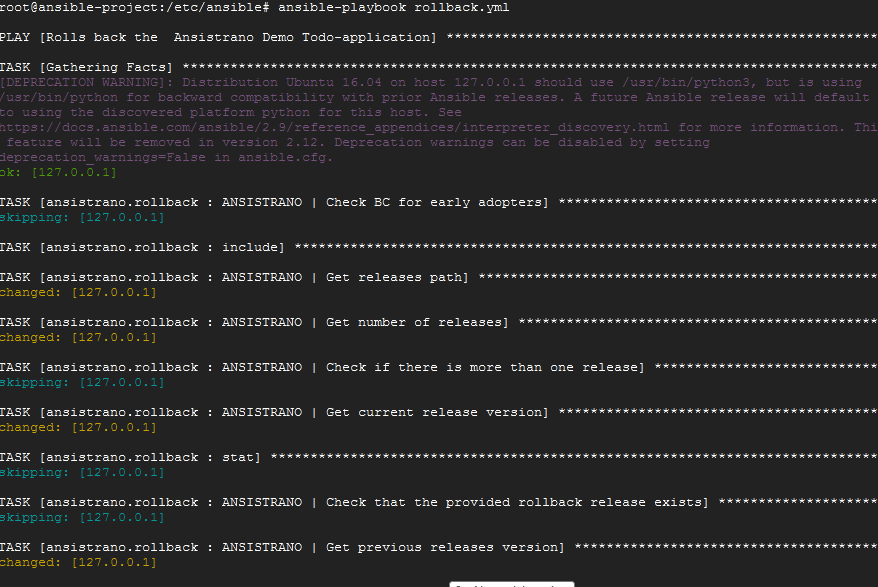
****

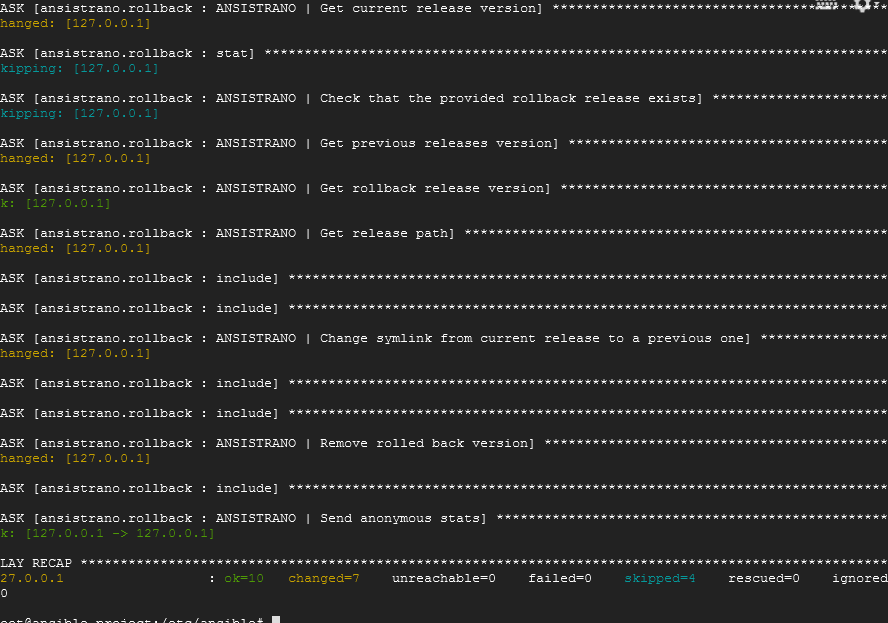
**In the revision file we can see the latest release id.**

**If we want to rollback our changes we can do that using rollback role as follows.**

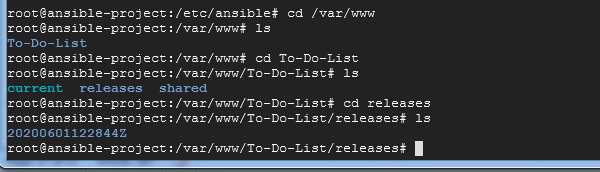
****

**Now we will be executing rollback.yml playbook as follows.**

****

****

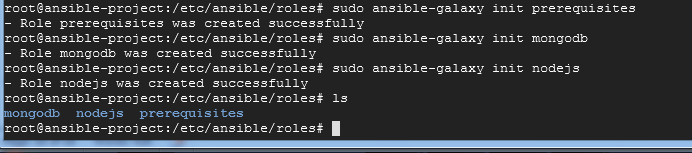
**After executing the deploy.yml we could see the following**

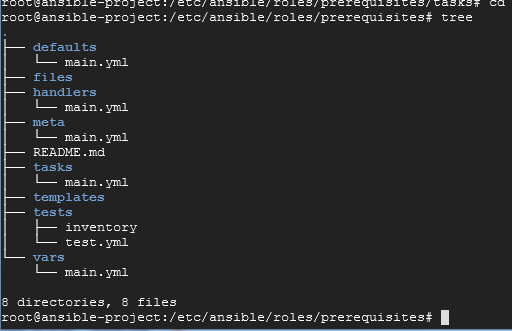
****

**Now we could see only one release in the release directory the previous changes were rolled back.**

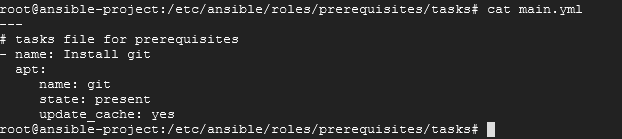
**4.Ansible versioning in GIT**

**We will be writing the roles using ansible galaxy as follows:**

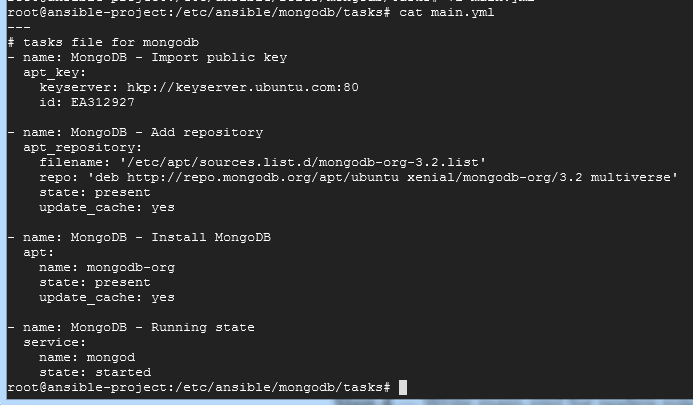
****

****

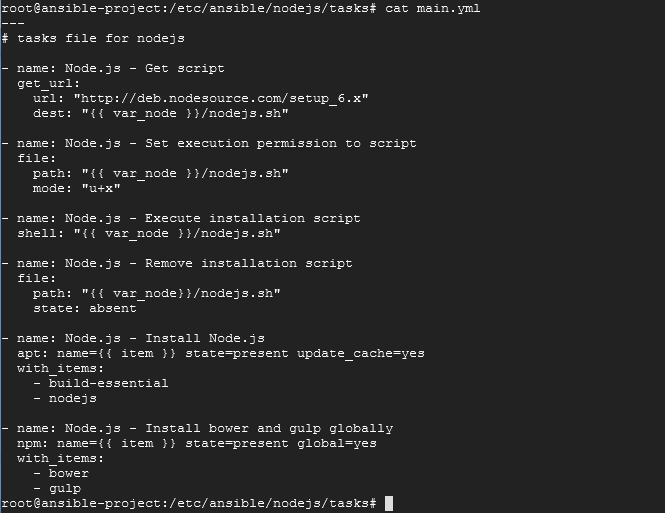
**Now we will be writing the tasks which installs git**

****

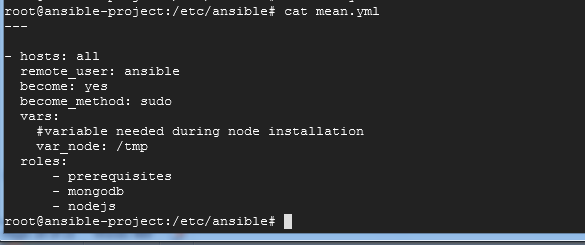
**Now we will be writing main.yml for MongoDb role**

****

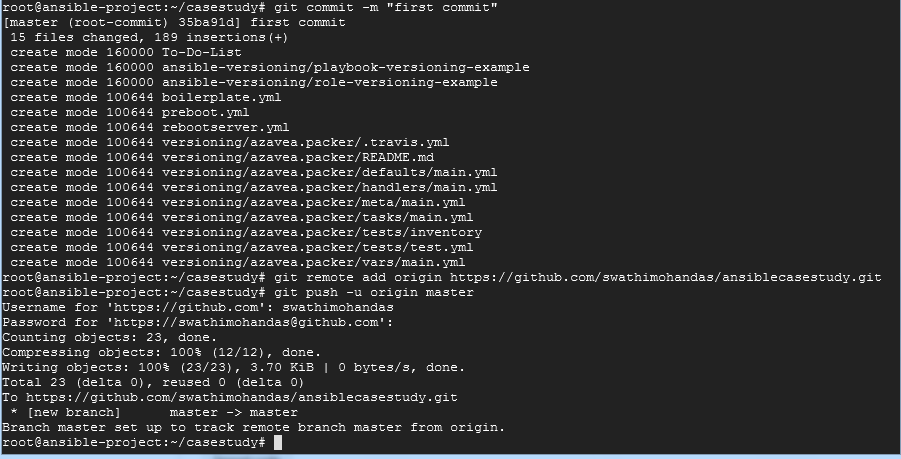
**Similarly Main.yml for nodejs**

****

**Now we will be writing playbook for meanstack installation where we will be mentioning the above mentioned roles**

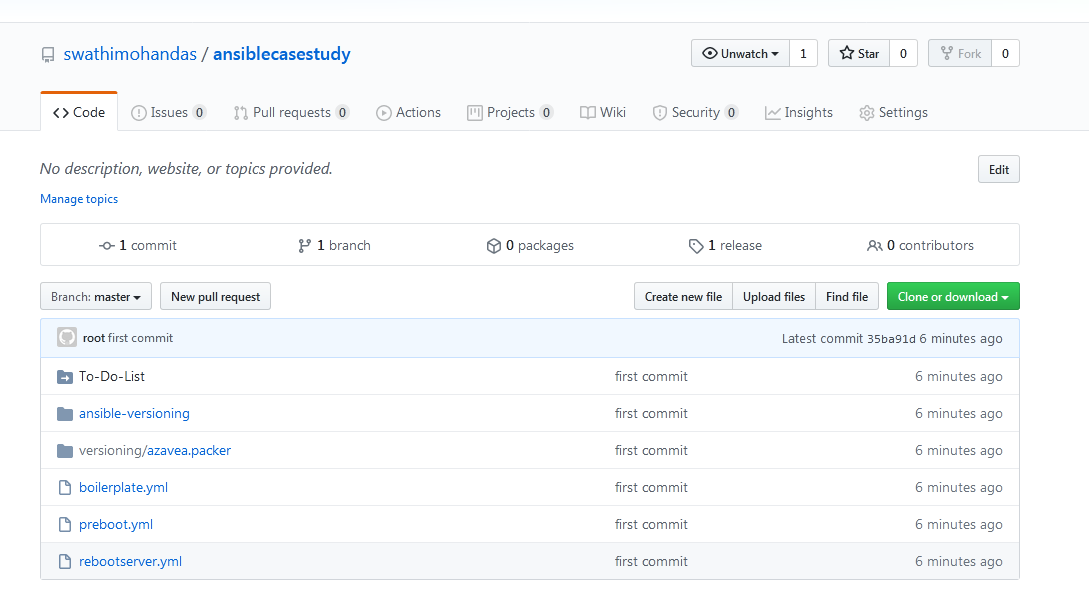
****

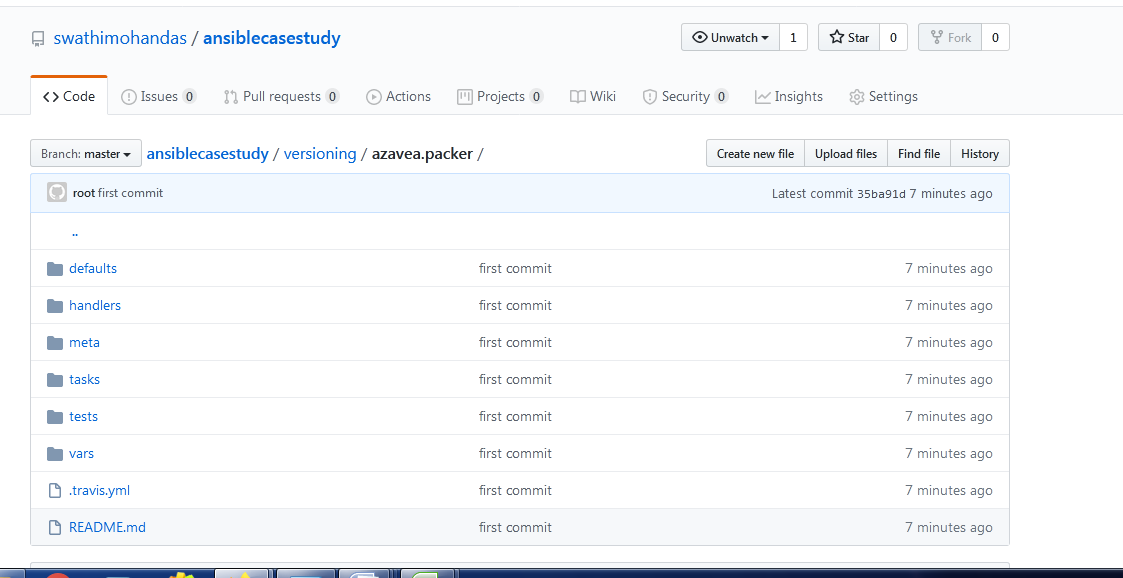
**Now we will be pushing our code to github**

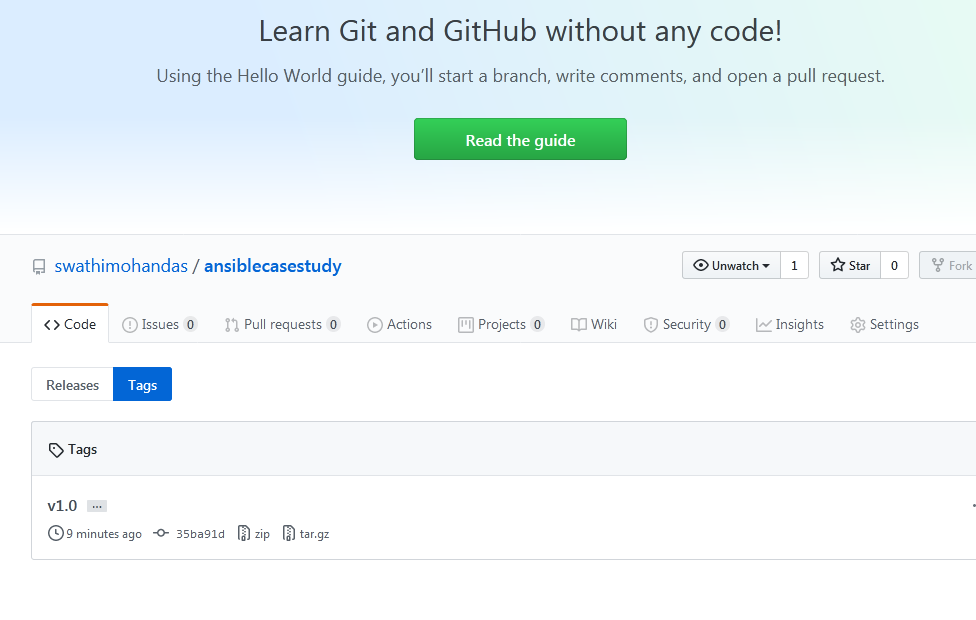
****

**After pushing we will be tagging the release as v1.0**

****

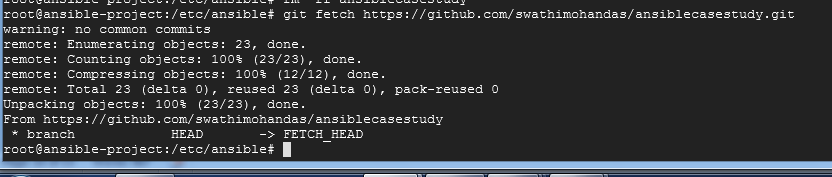
****

****

****

**We could see the releases as well tags**

**Sameway we will deploy few more codes and tag it as v1.1**

****

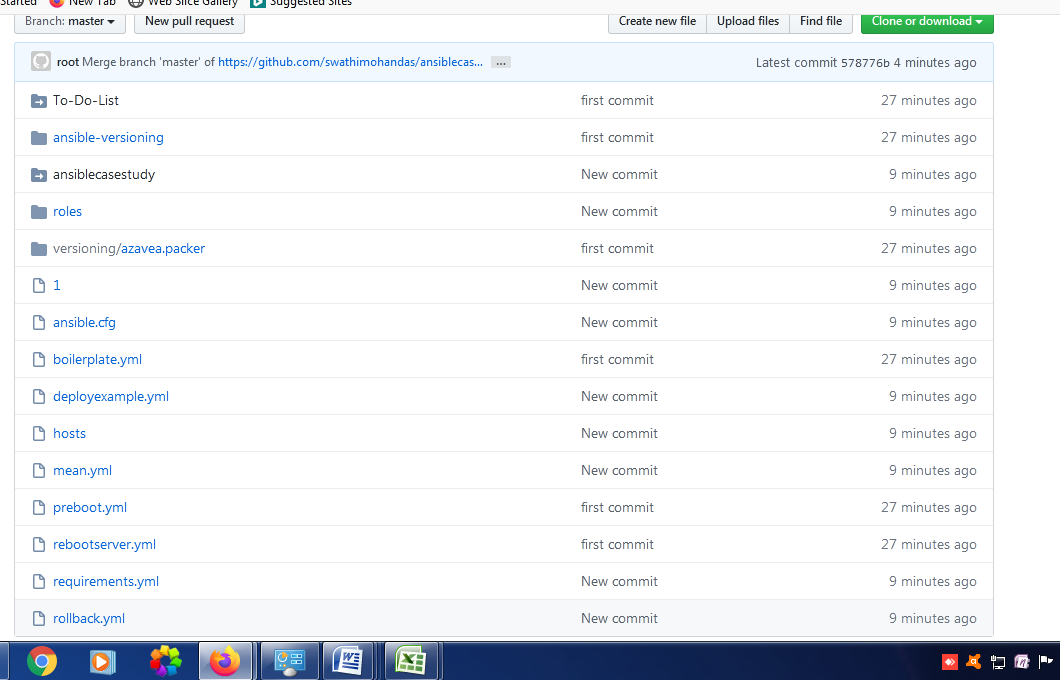
**Then we will pull**

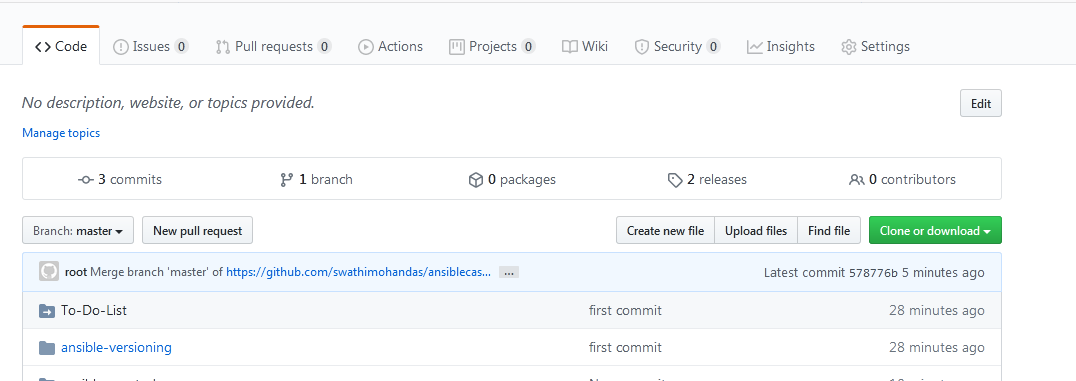
**Then git add .**

**Then execute commit and git push commands**

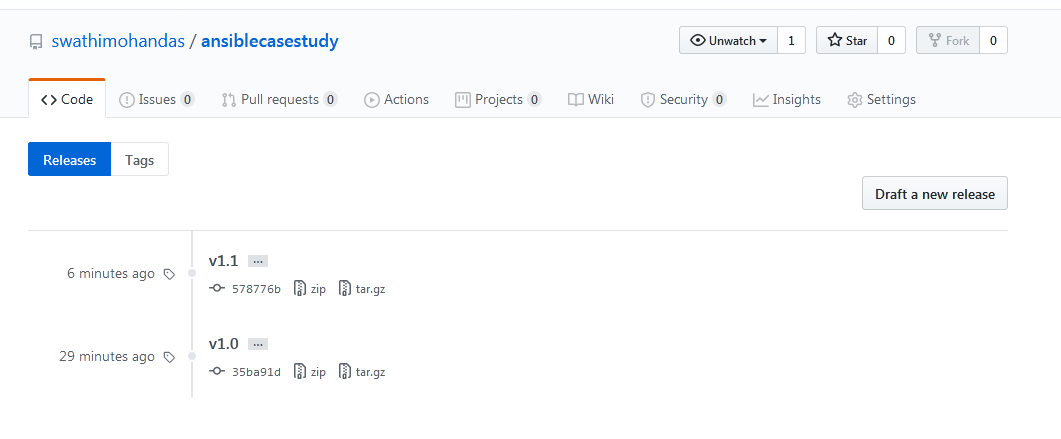
**Then give the tag as v1.1**

**The result will be as follows**

****

****

**We could see two releases as above**

****

**Second release will have all files after second commit and first release will be having the code after 1st release.**

**In this way we can maintain the versioning of the ansible playbooks and roles as well in github and give the tag while referring.**