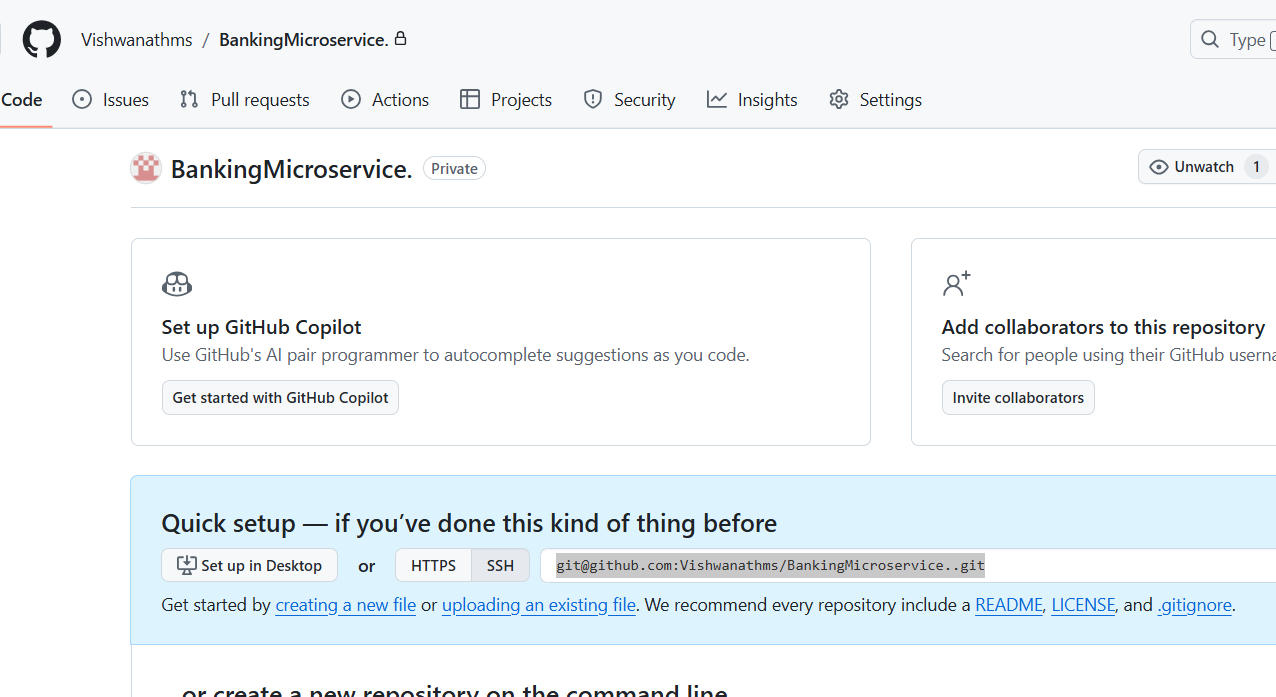
1. **Repository Creation:**

[git@github.com:Vishwanathms/BankingMicroservice..git](mailto:git@github.com:Vishwanathms/BankingMicroservice..git) -- contains all the code

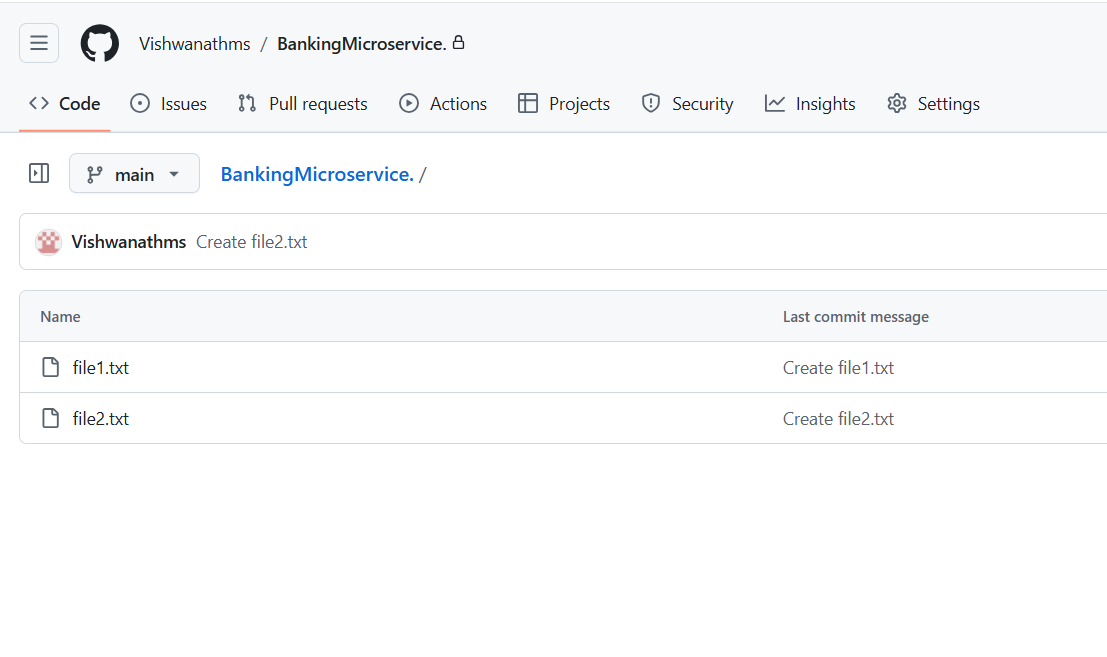
* + Create a GitHub repository named **BankingMicroservice**.

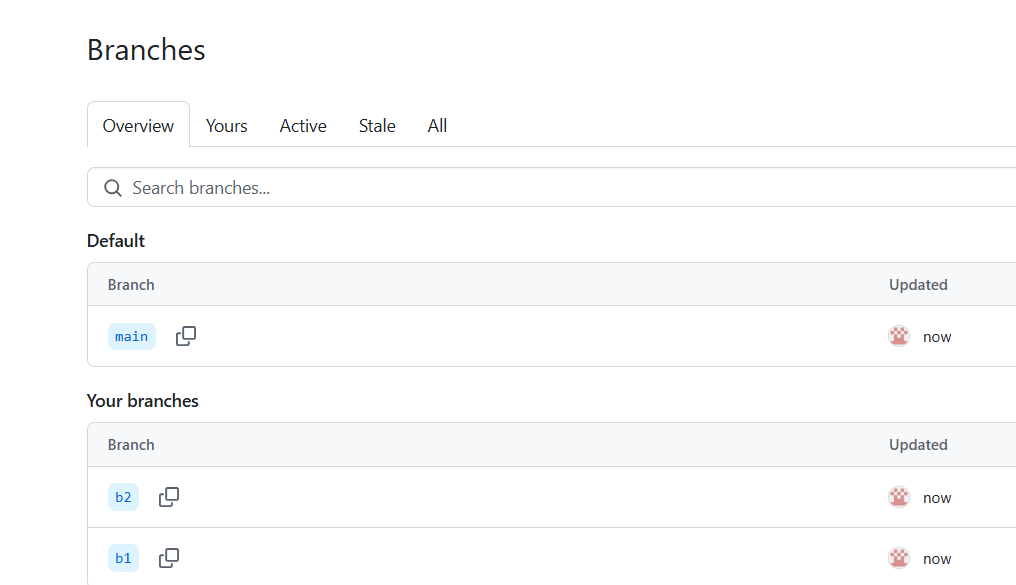


1. **Branch Management:**
   * As a code reviewer, create individual branches for each developer involved in the project. Assume there are 2 developers

Created 2 files from the main, as an reviewer,

Also create 2 branch “b1” and “b2”

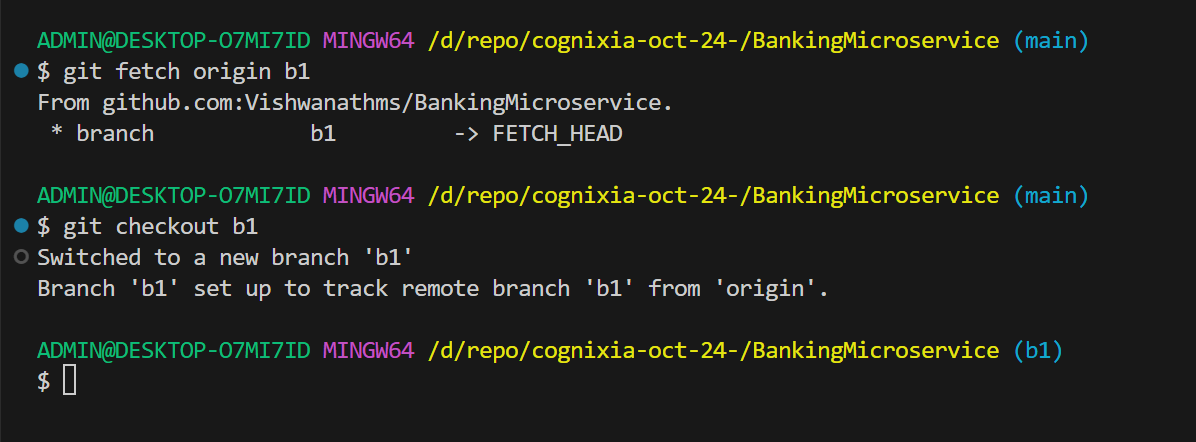




Both Branch are copies of the main branch

* + Developers will push their code to their respective branches upon completing their work.

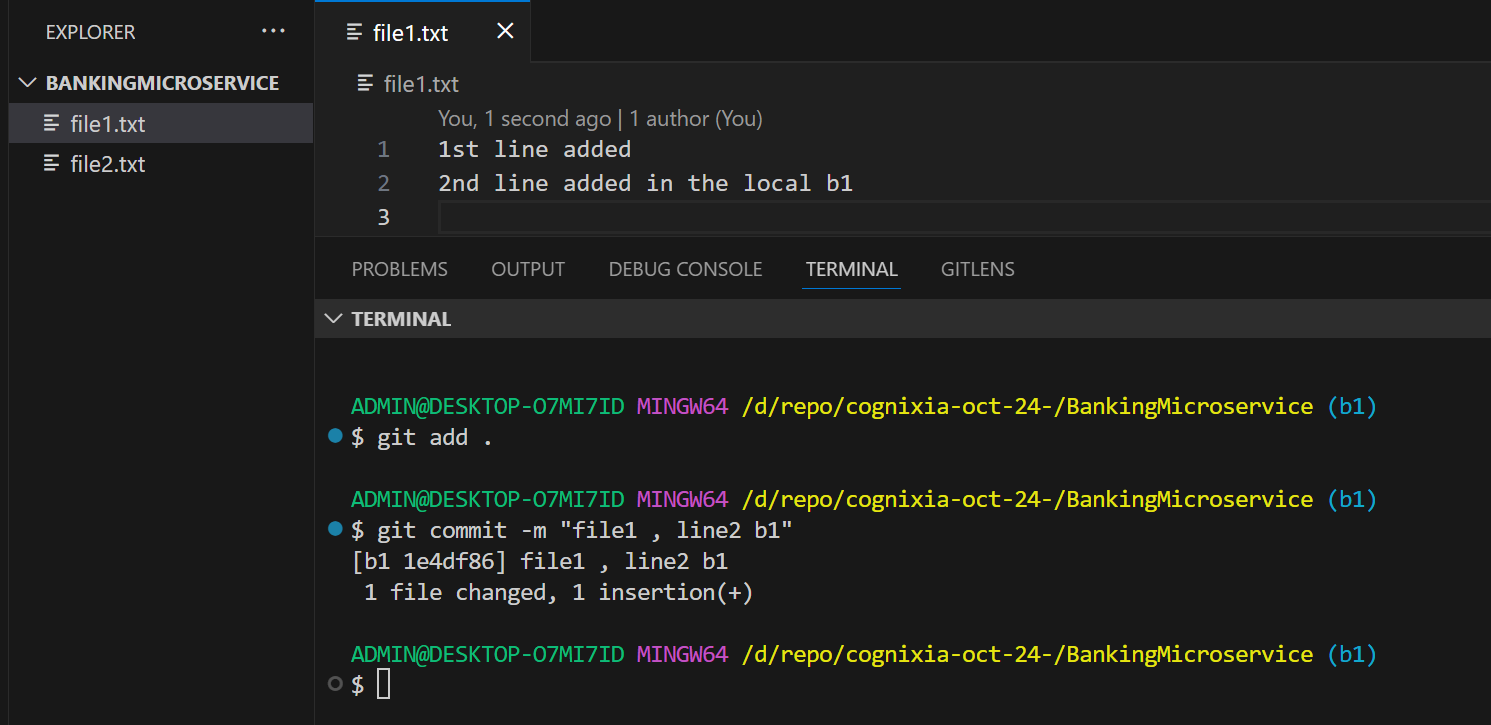
Developer 1 is downloading b1

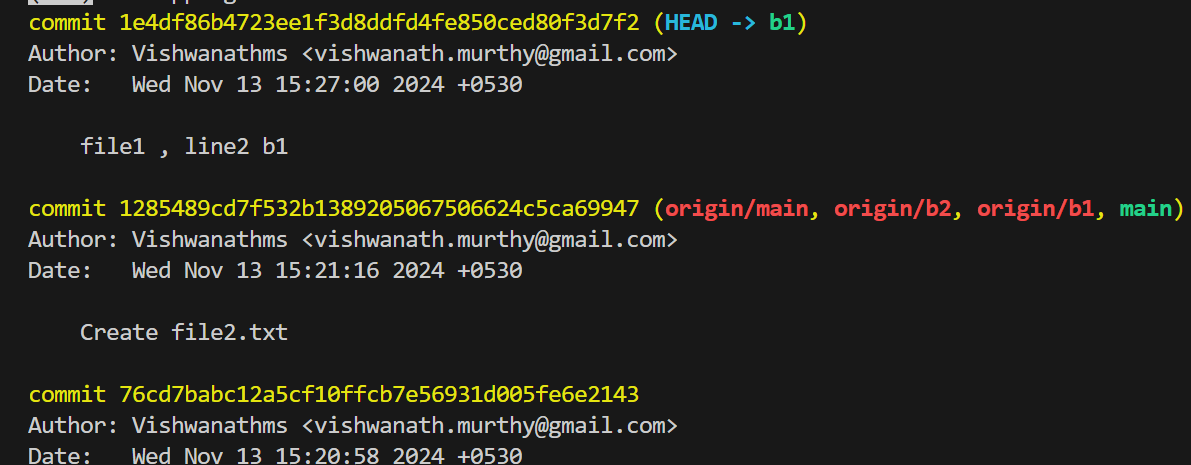


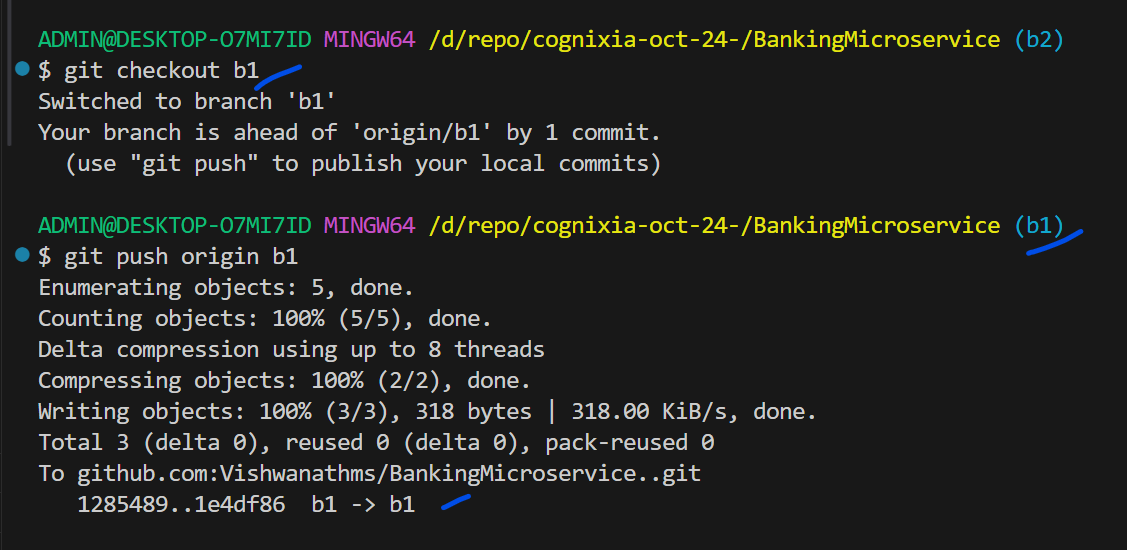


Log shows all the branches are at HEAD.

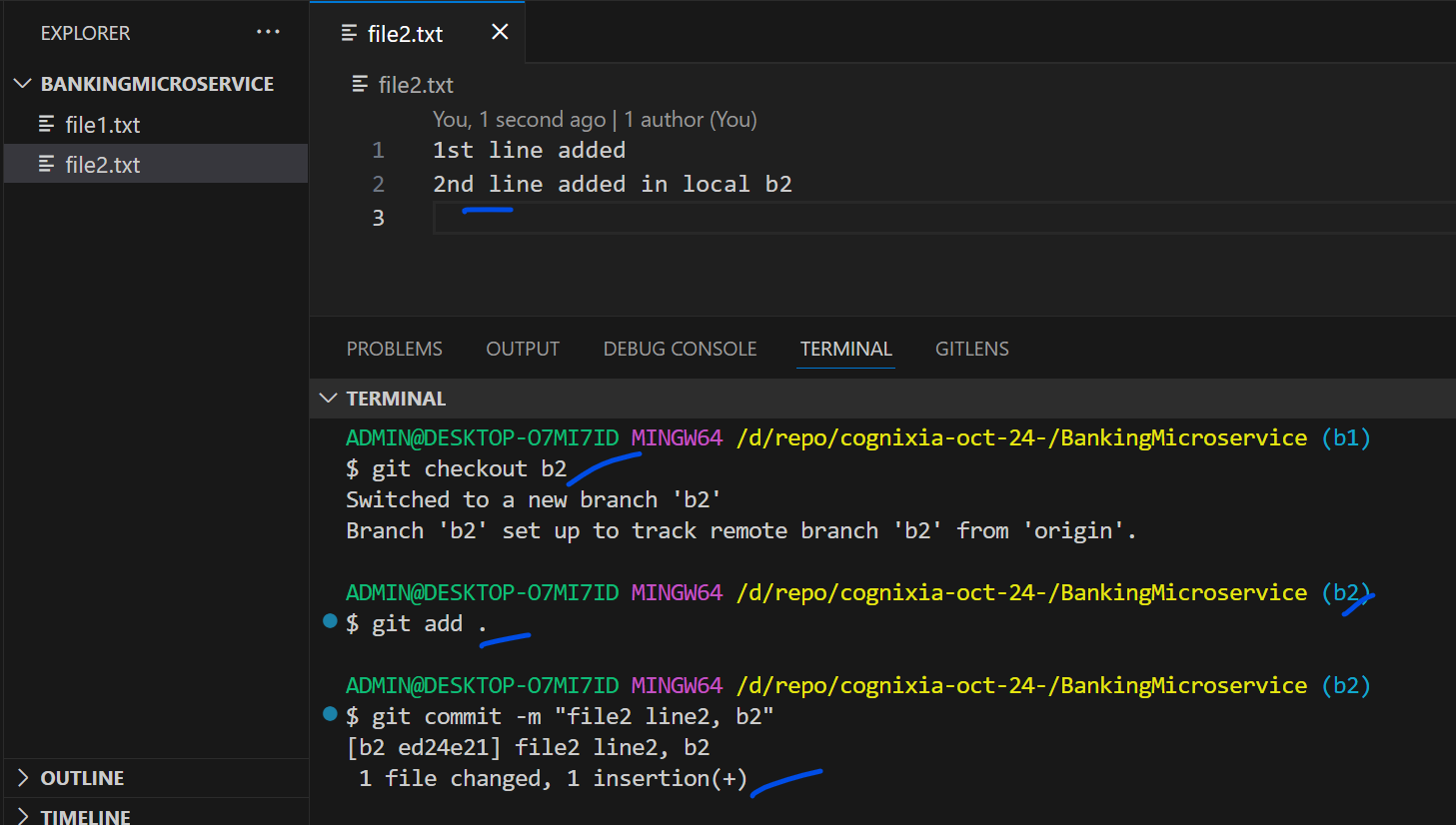
Now we will create a commit on b1, by changing the file1

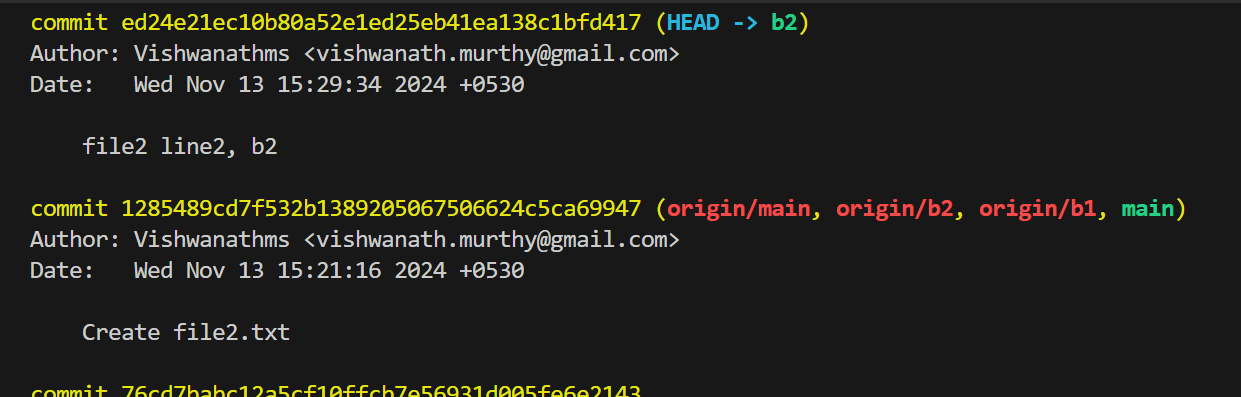


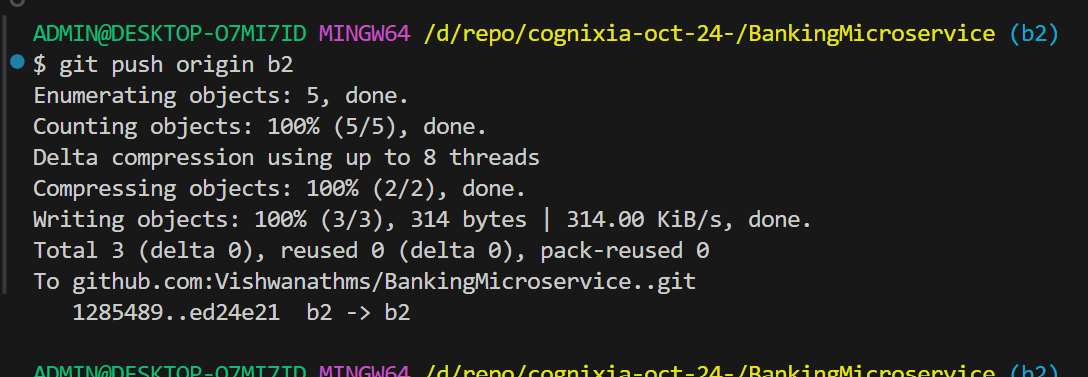




Now we will switch to “b2” create a commit on b2, by changing the file2

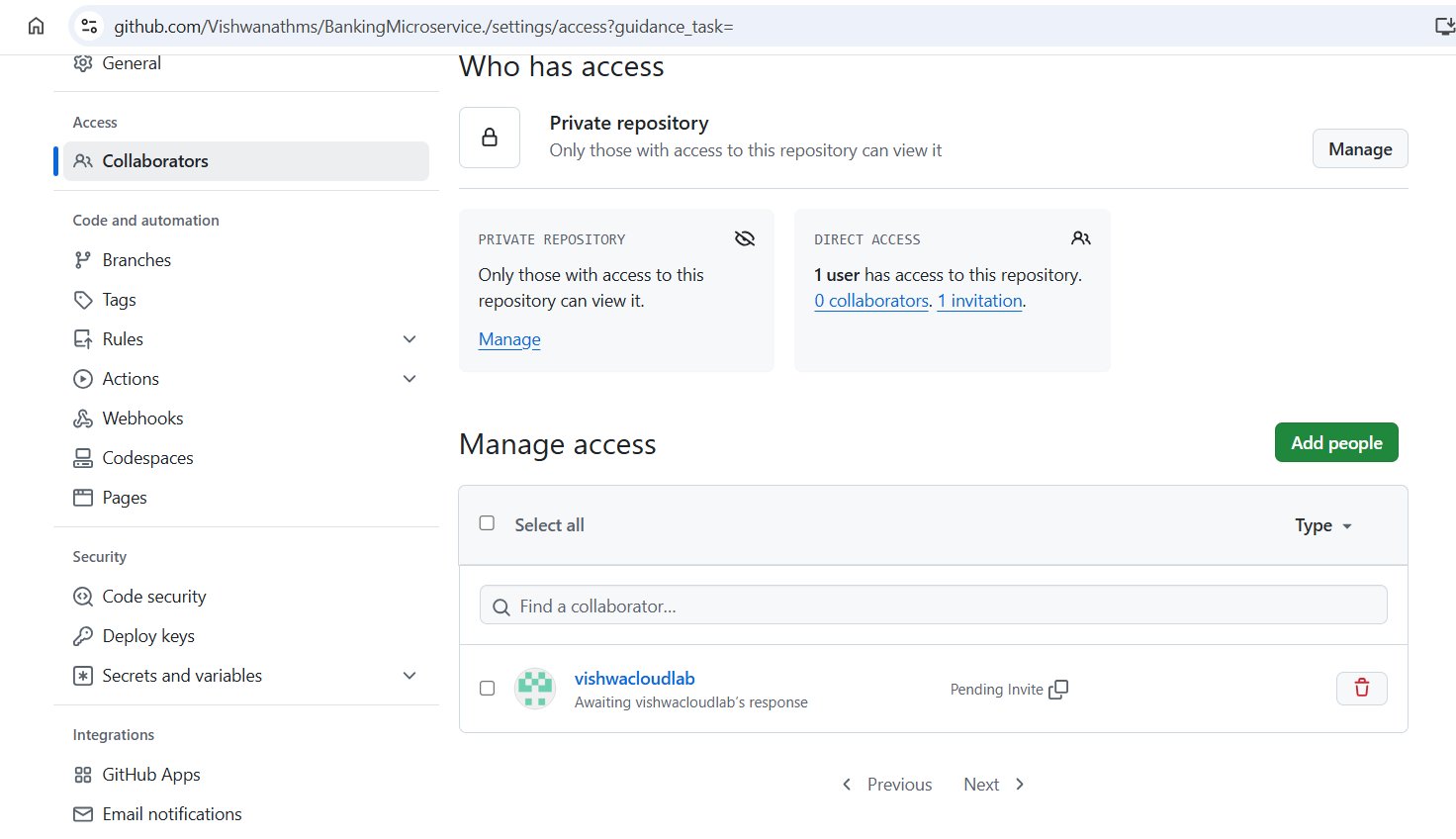


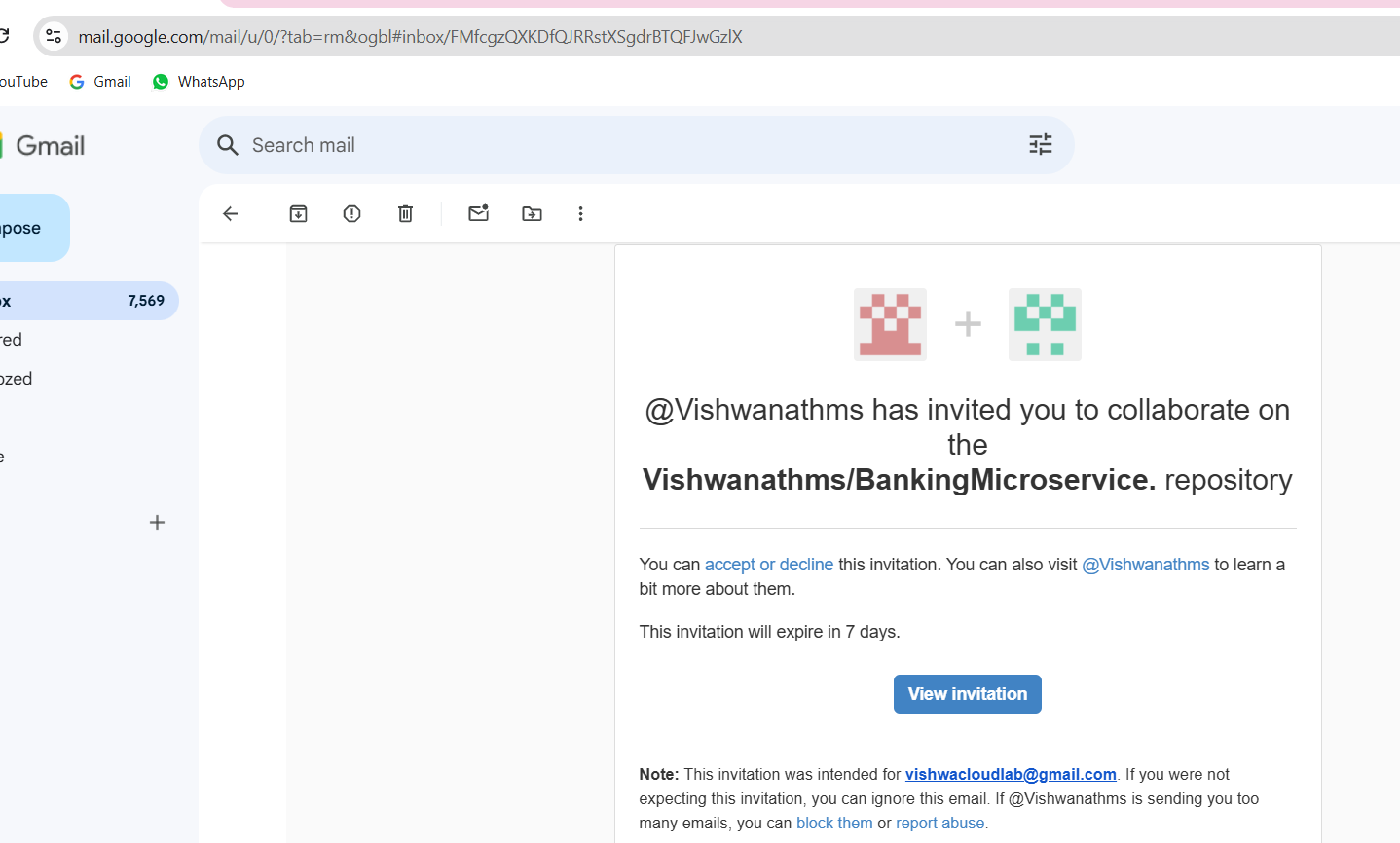


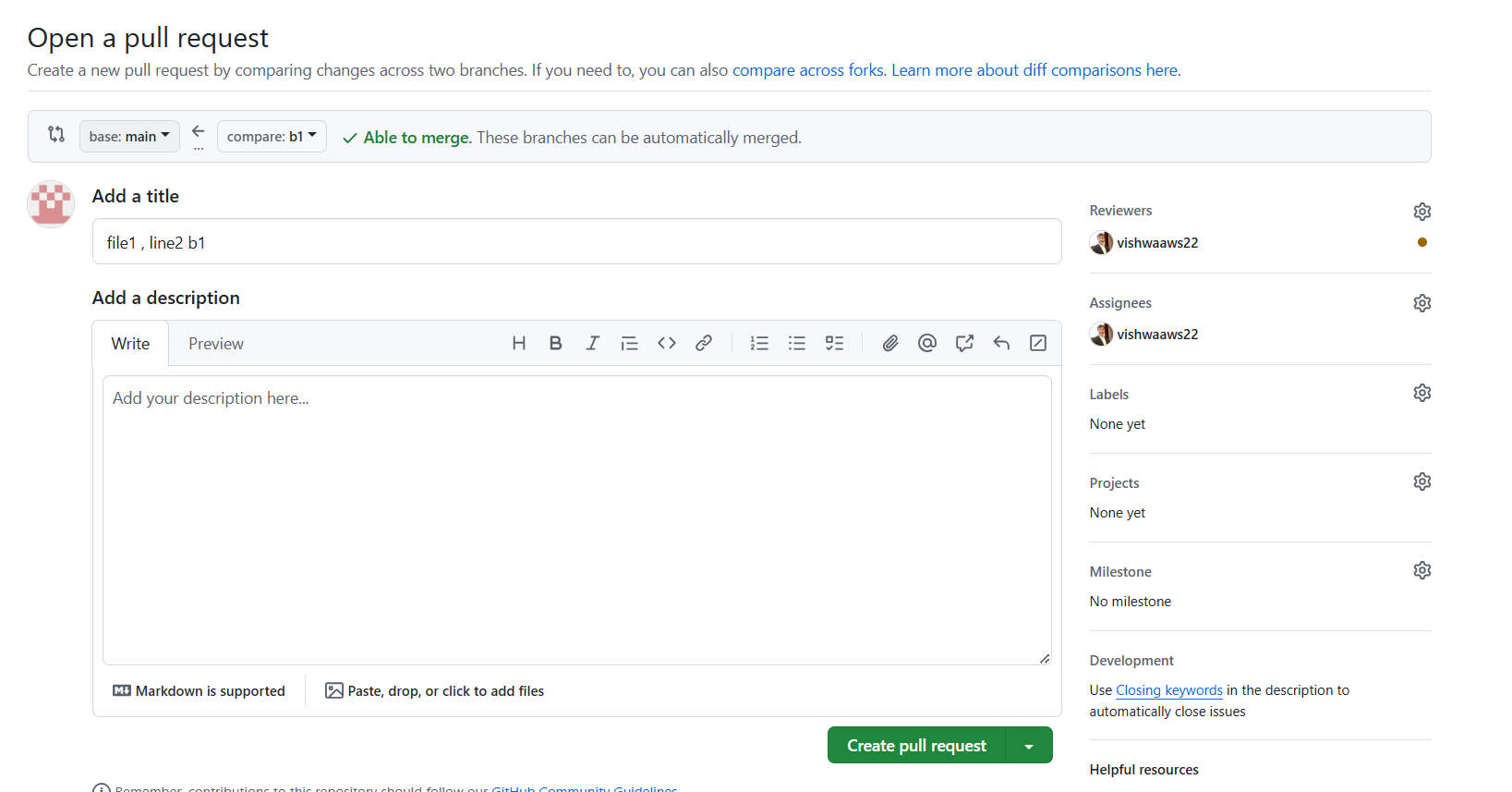


* + Review the code changes and merge approved changes into the **Master** branch, which contains the latest stable code.

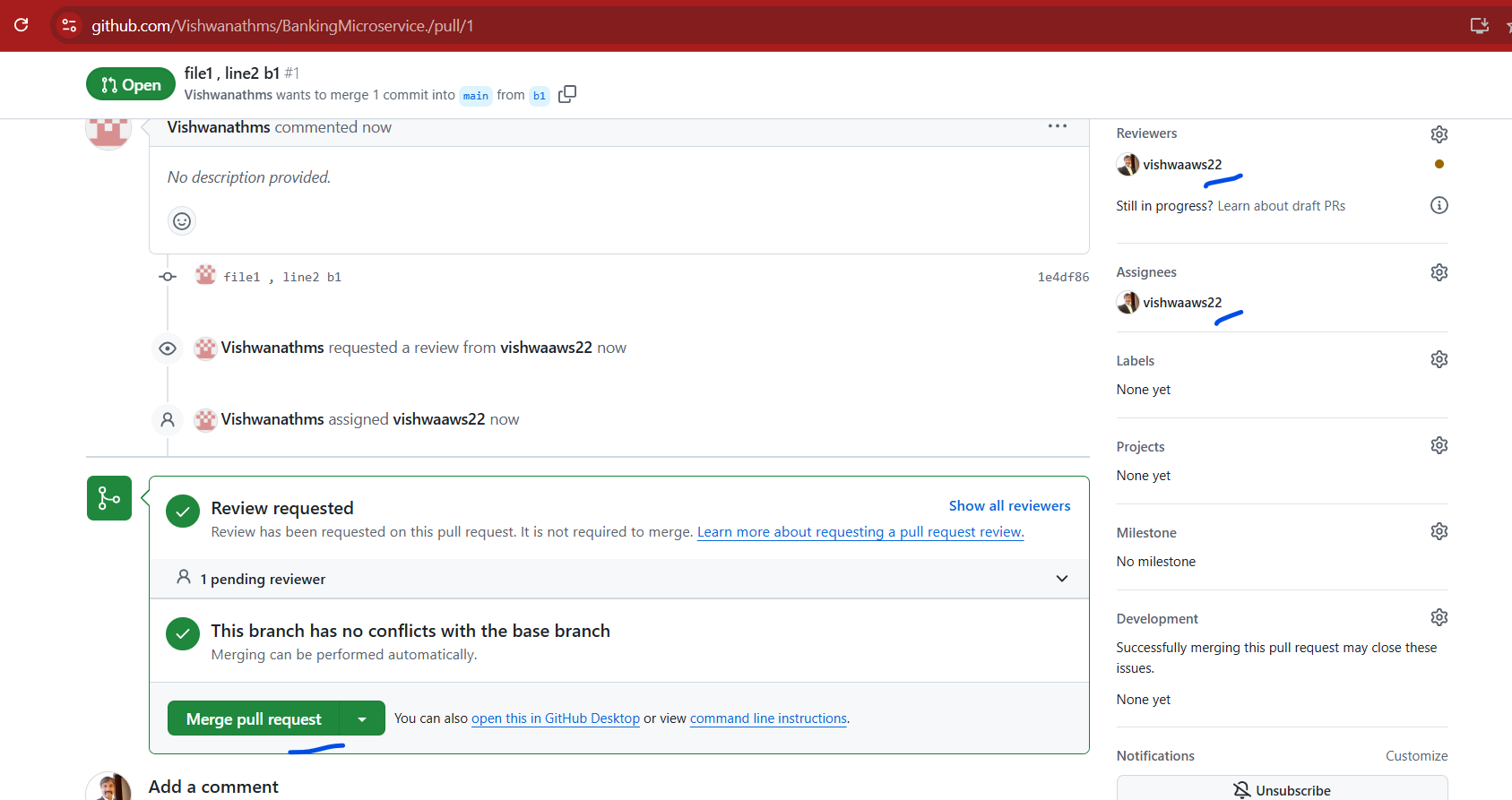
Invited an user





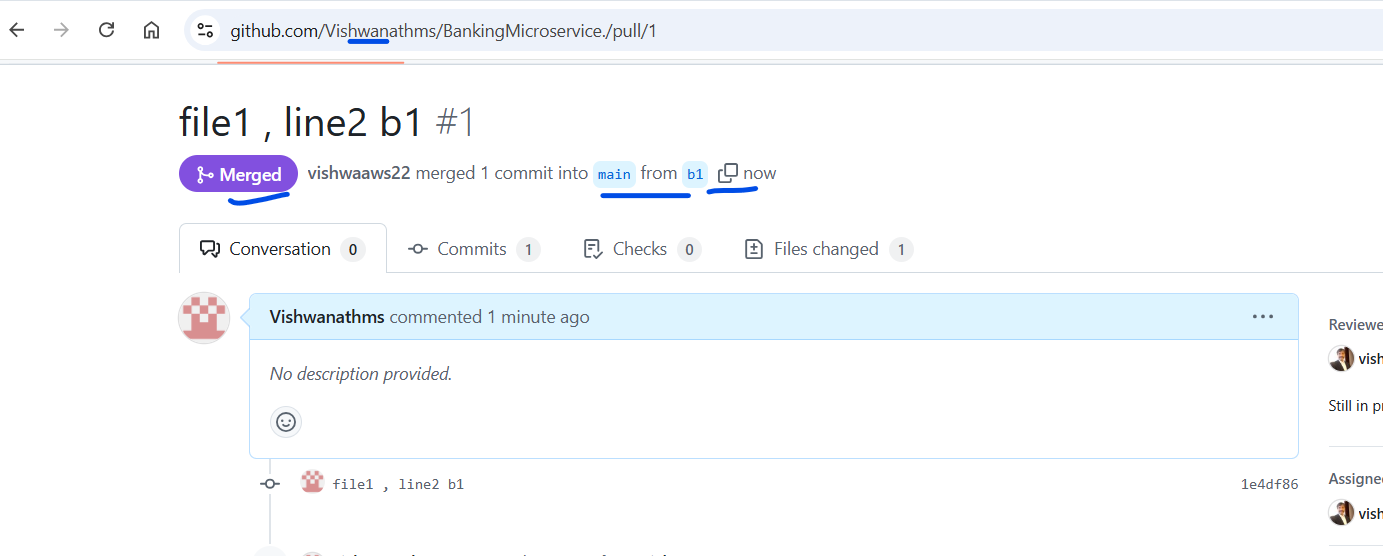


I have switched my github to user “vishwaaws22”

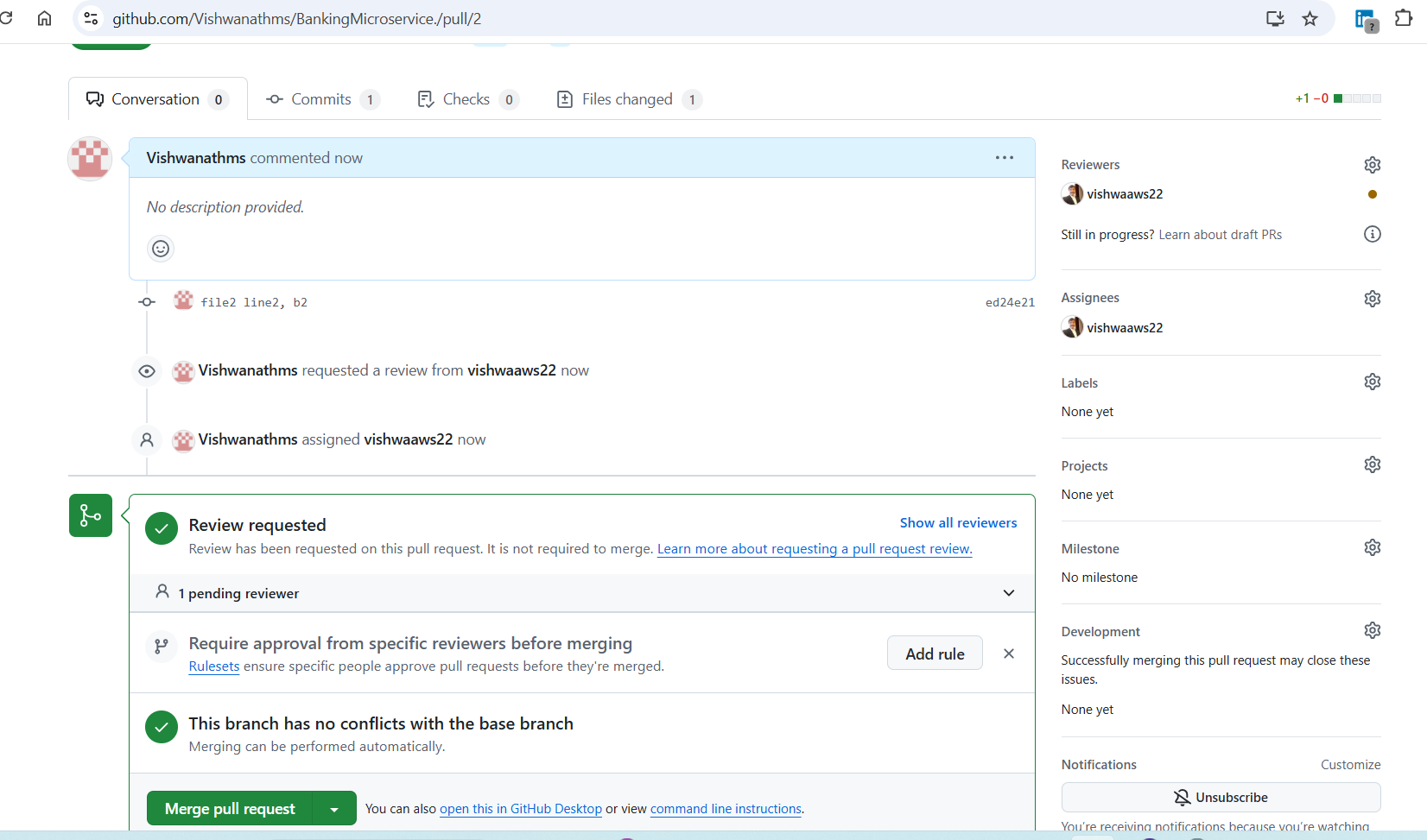


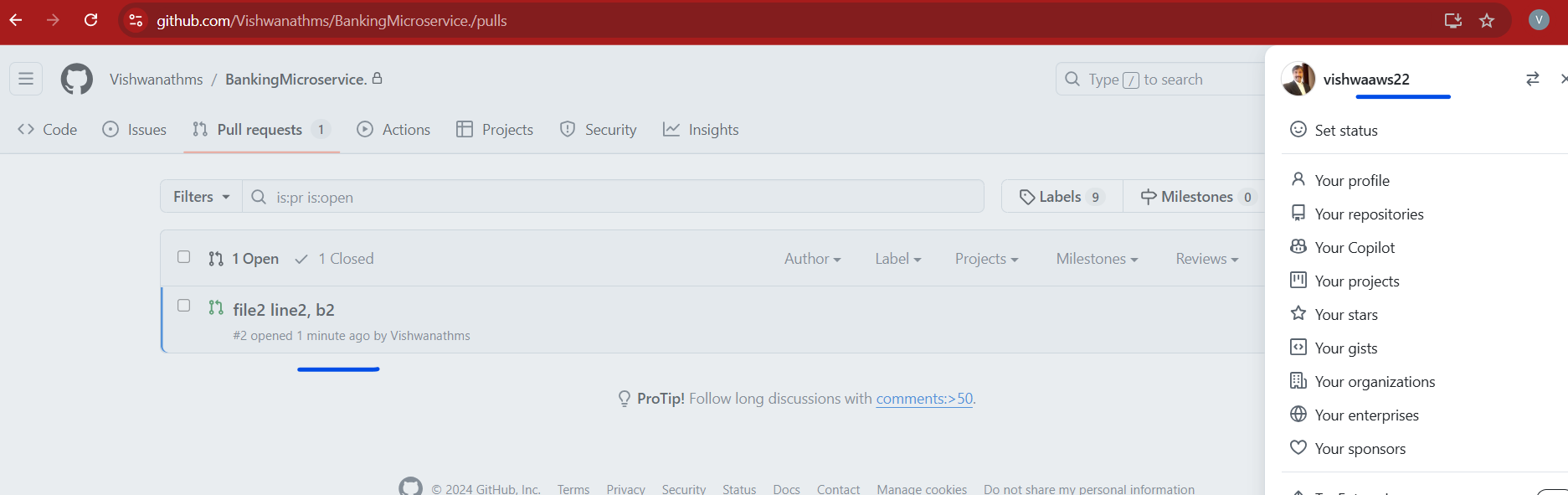
I will approve from this .

Below screen shows that on main account, it shows as merged



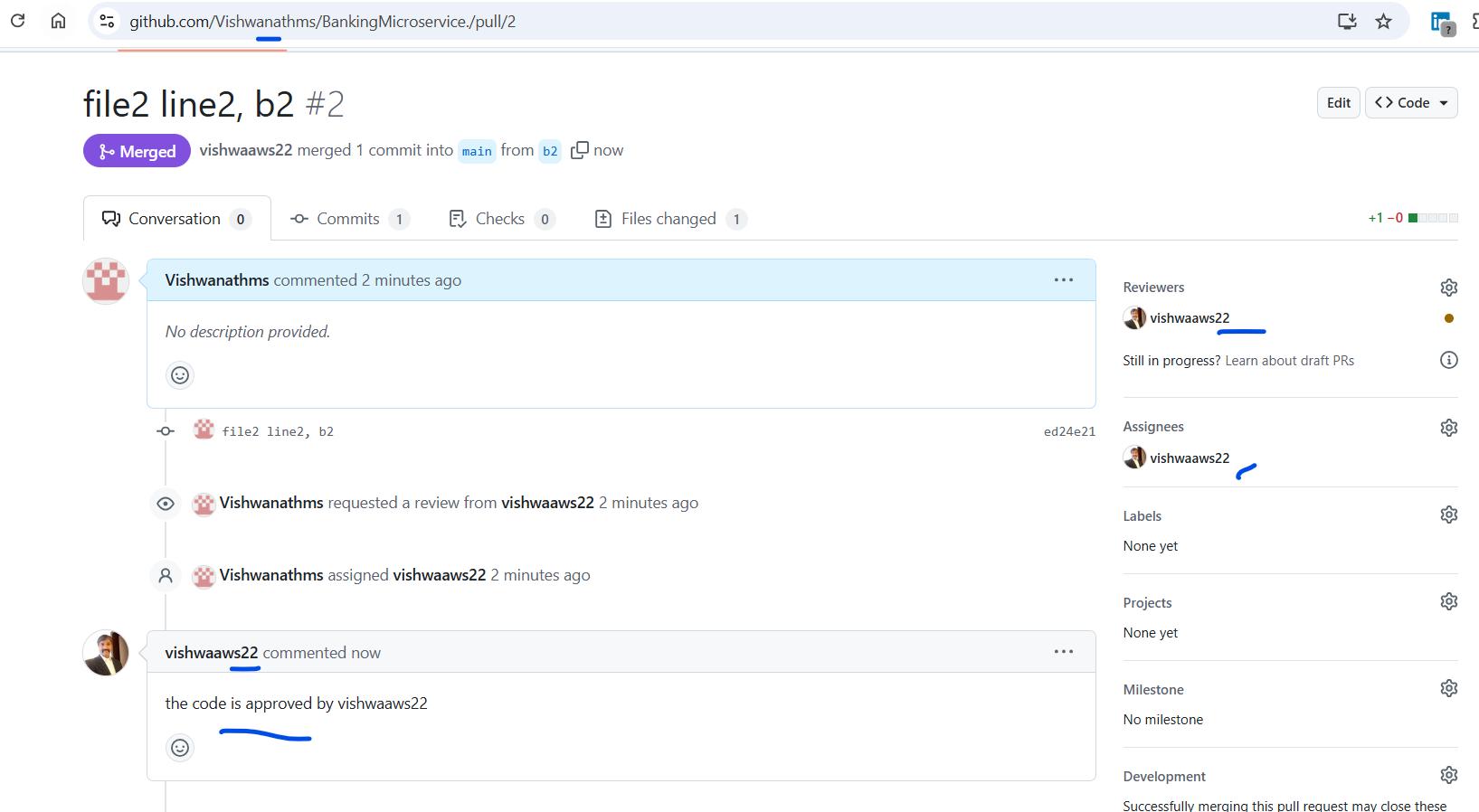
Same thing for b2





I have got the request on another account of github.

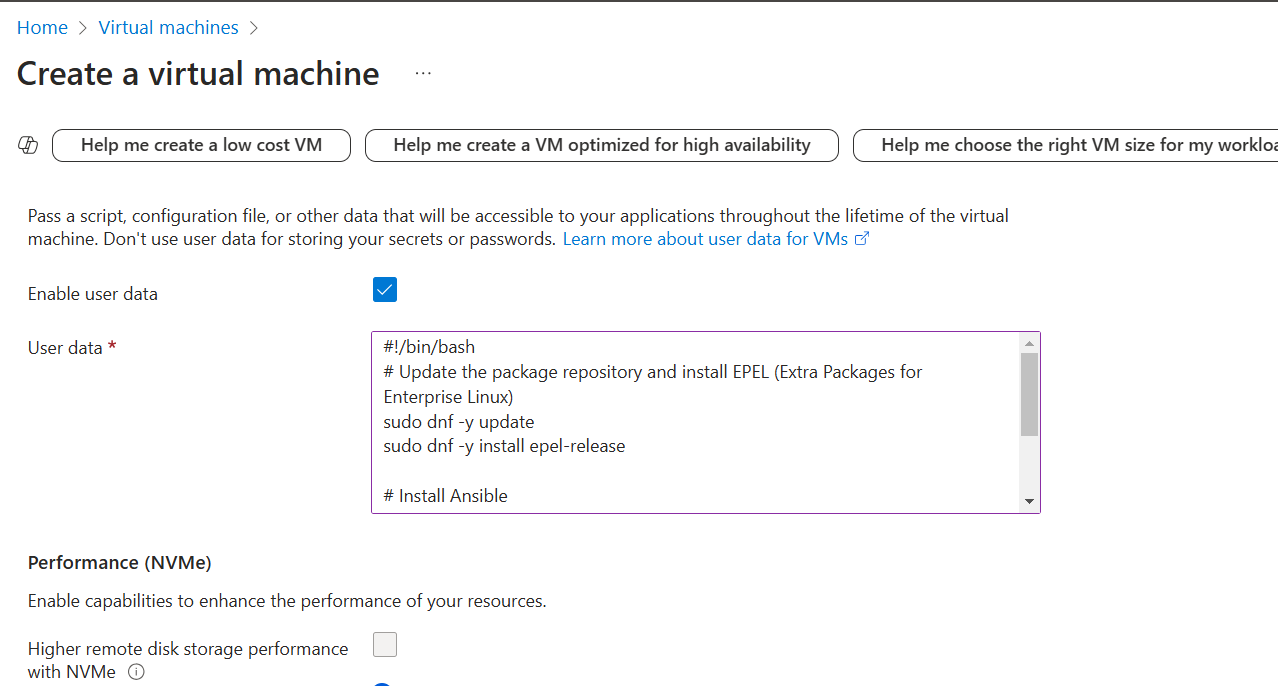
Prrof on main account



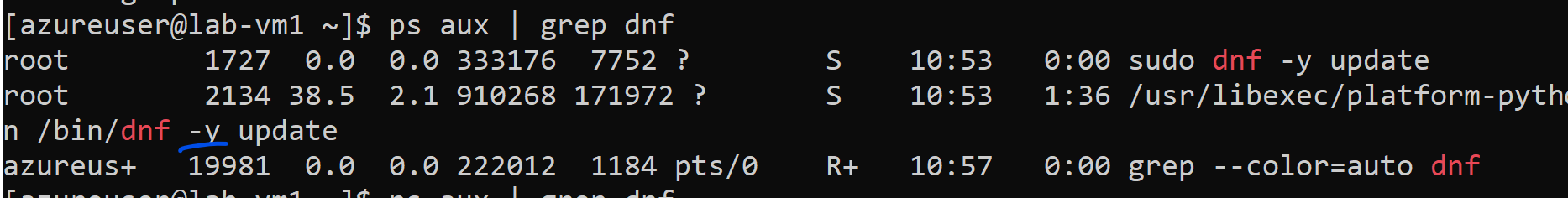
Note: as the implementation part, make sure to showcase all the work in the perspective of Reviewer , developers and approver.

1. **Production Server Preparation**
2. **Server Environment:**
   * Prepare 2 ( Ansible controller and Client) Linux production server (Ubuntu or Rocky Linux).

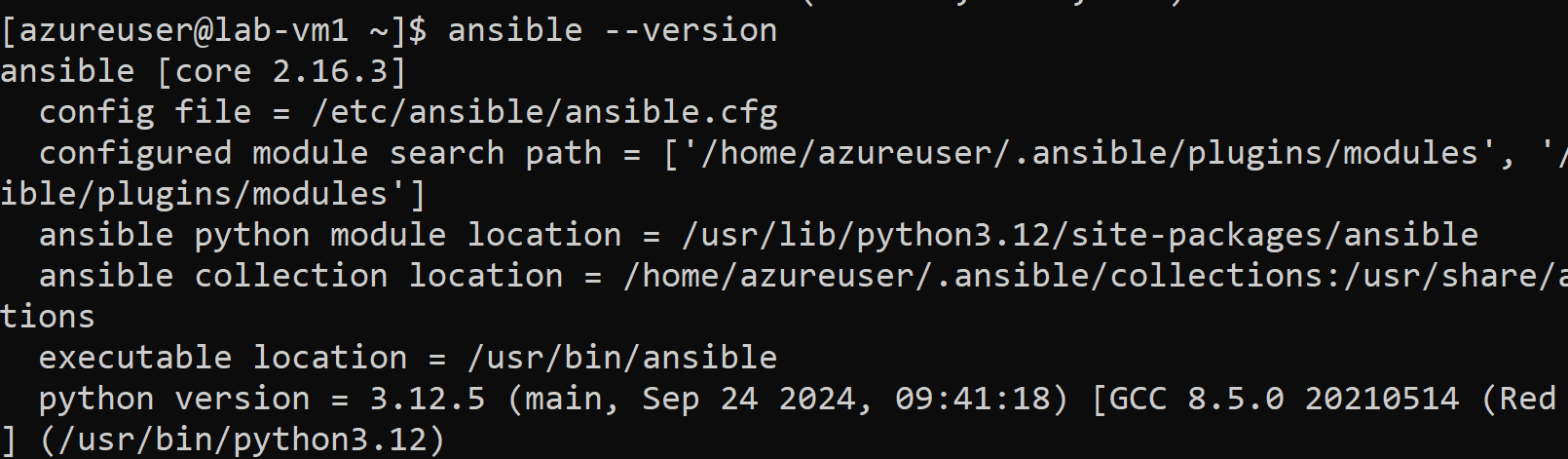
Installing Lab-vm1



On the Lab-Vm01 , the “dnf” was still updating

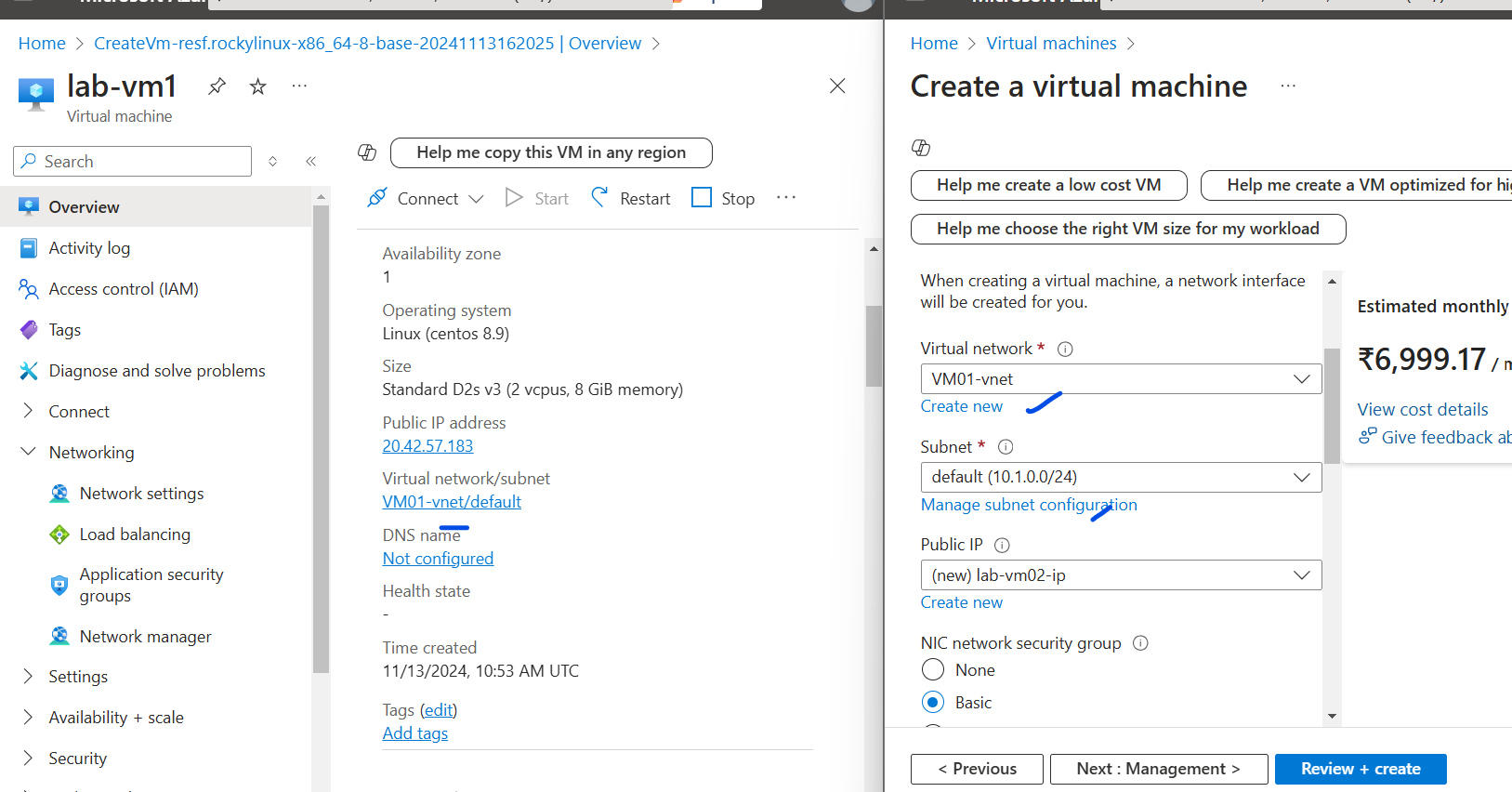


Waited for 3-5 min the ansible was installed from the “user-data”



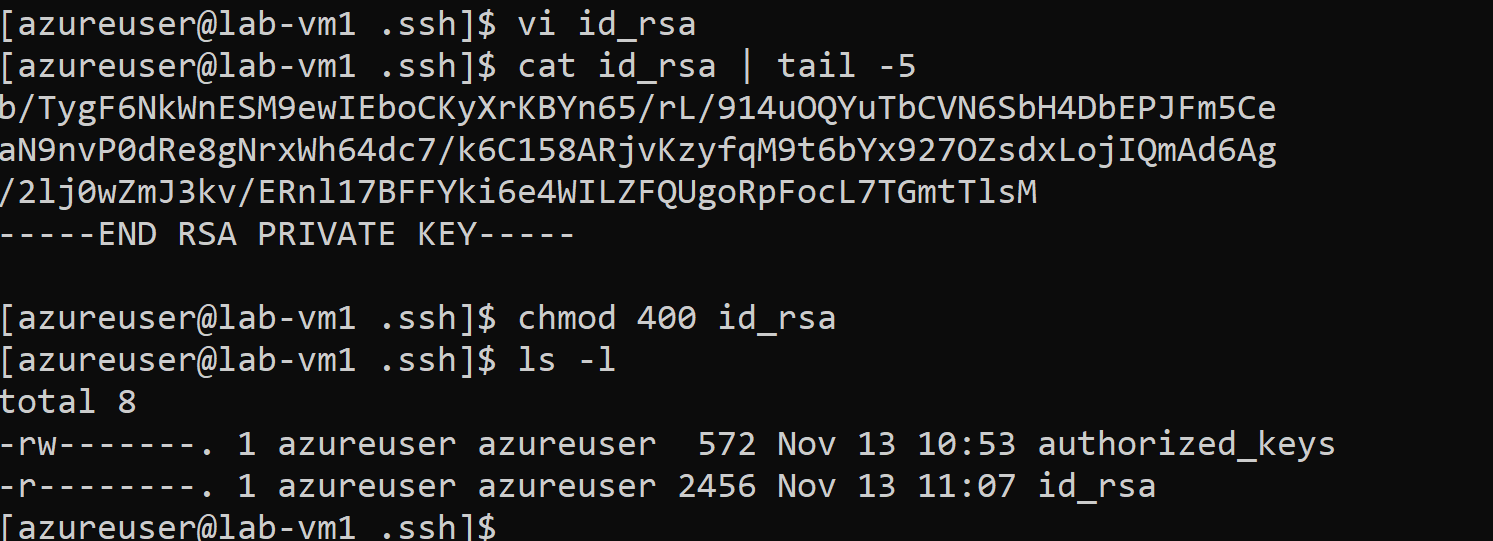
Created an Rocky linux with ssh key enabled for ansible-controller.

Creating 2nd VM , which is the client in the same vnet and subnet

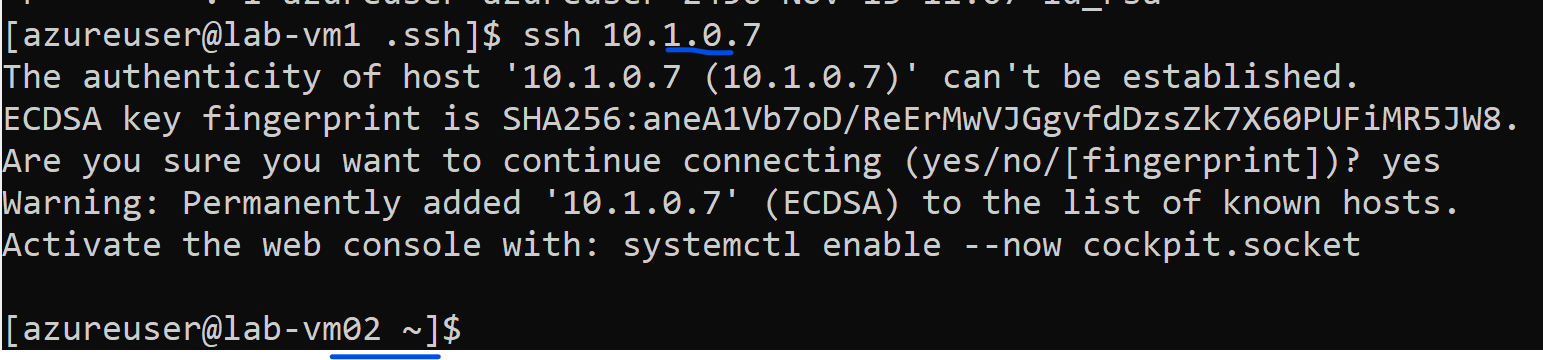


For now port ssh is opened on both the VM’s.

Now we will configure put the private key that is already downloaded as pem file on the ansible controller.

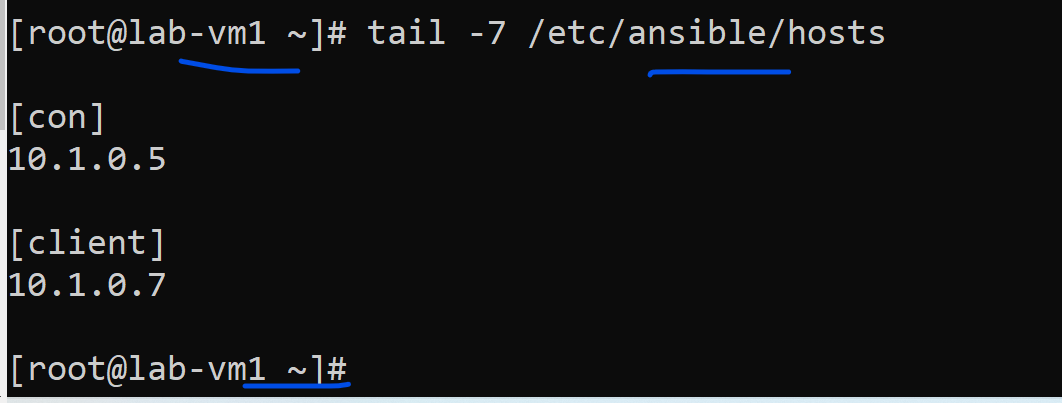


Testing – to connect to lab-vm02



Lets now test the ansible working

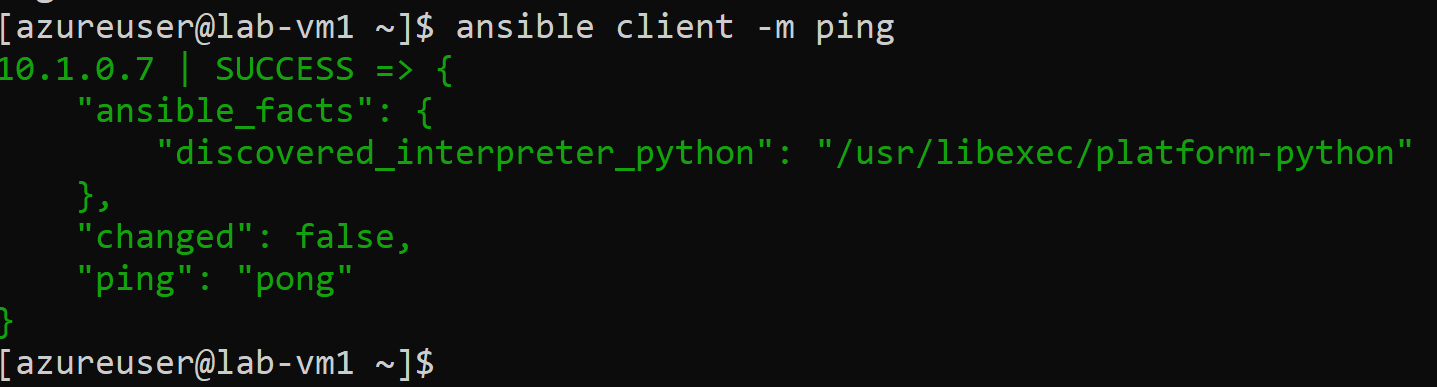
We will add the ip of the client to the ansible inventory file.



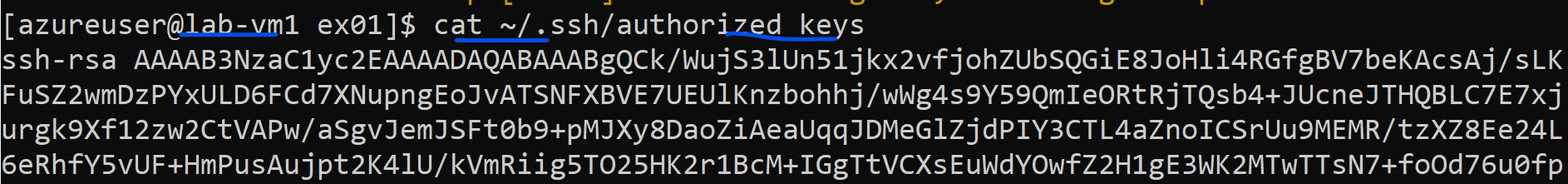
We need to do the above work as **root**.

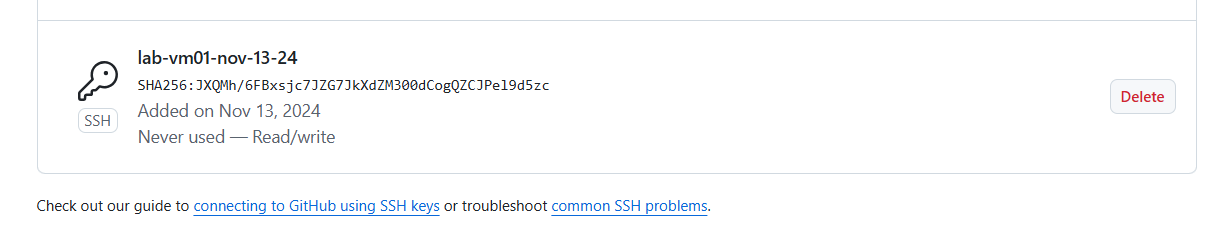
Testing the ansible connection to the client,

Ansible commands has to be run as the user, bcz the key is in the user profile and NOT IN THE ROOT profile.

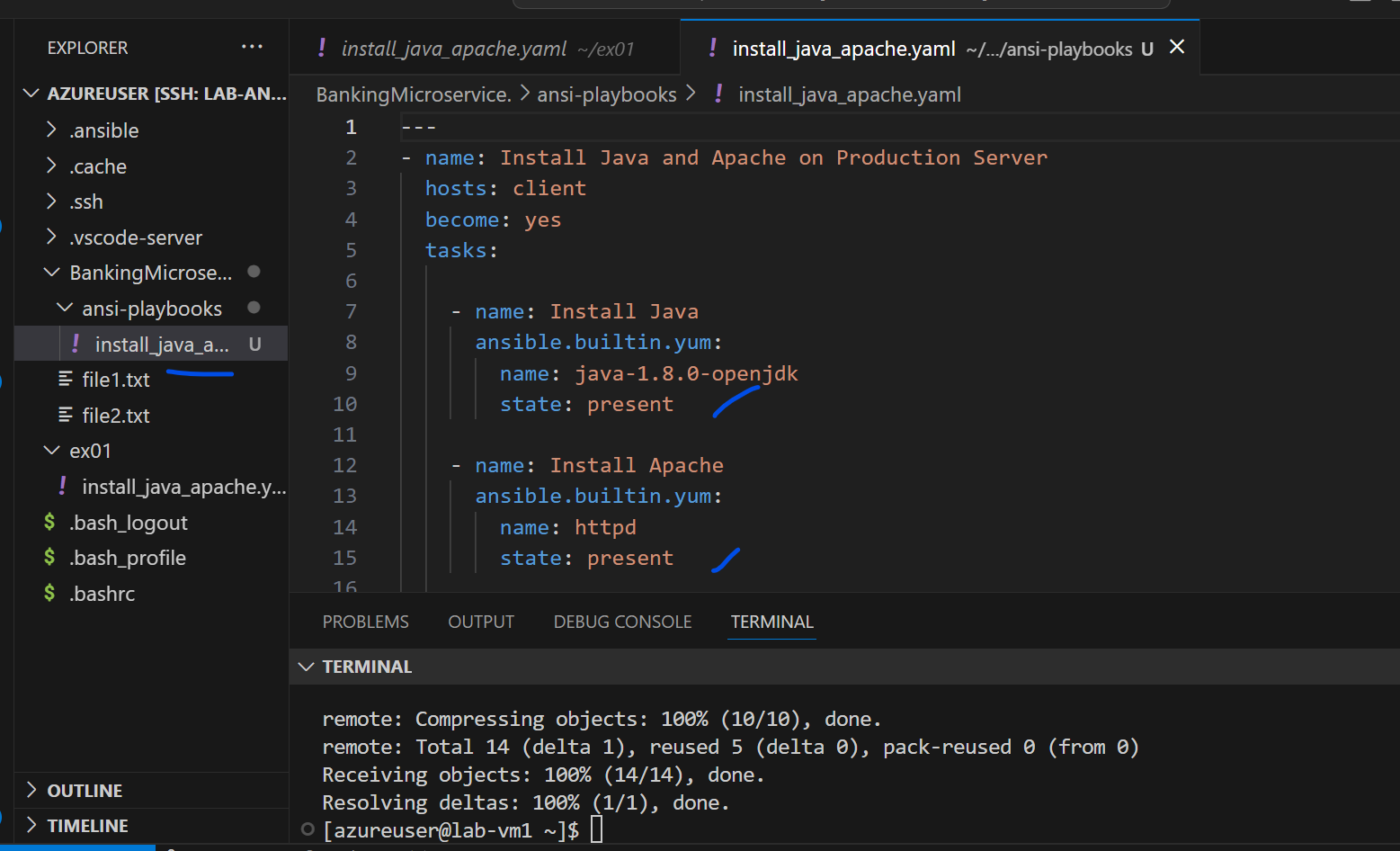


Before, we move to the next step, lets download the repo which was created in step1 and attach the public key on the github





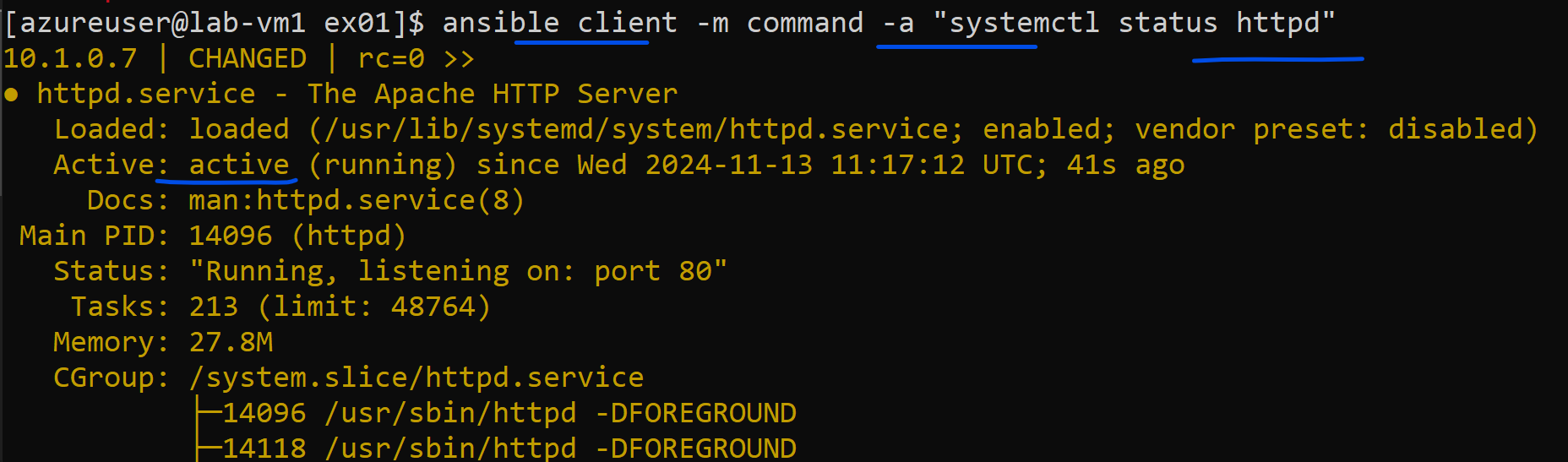
1. **Software Installation:**
   * Use Ansible to install Java and Apache on the production server.



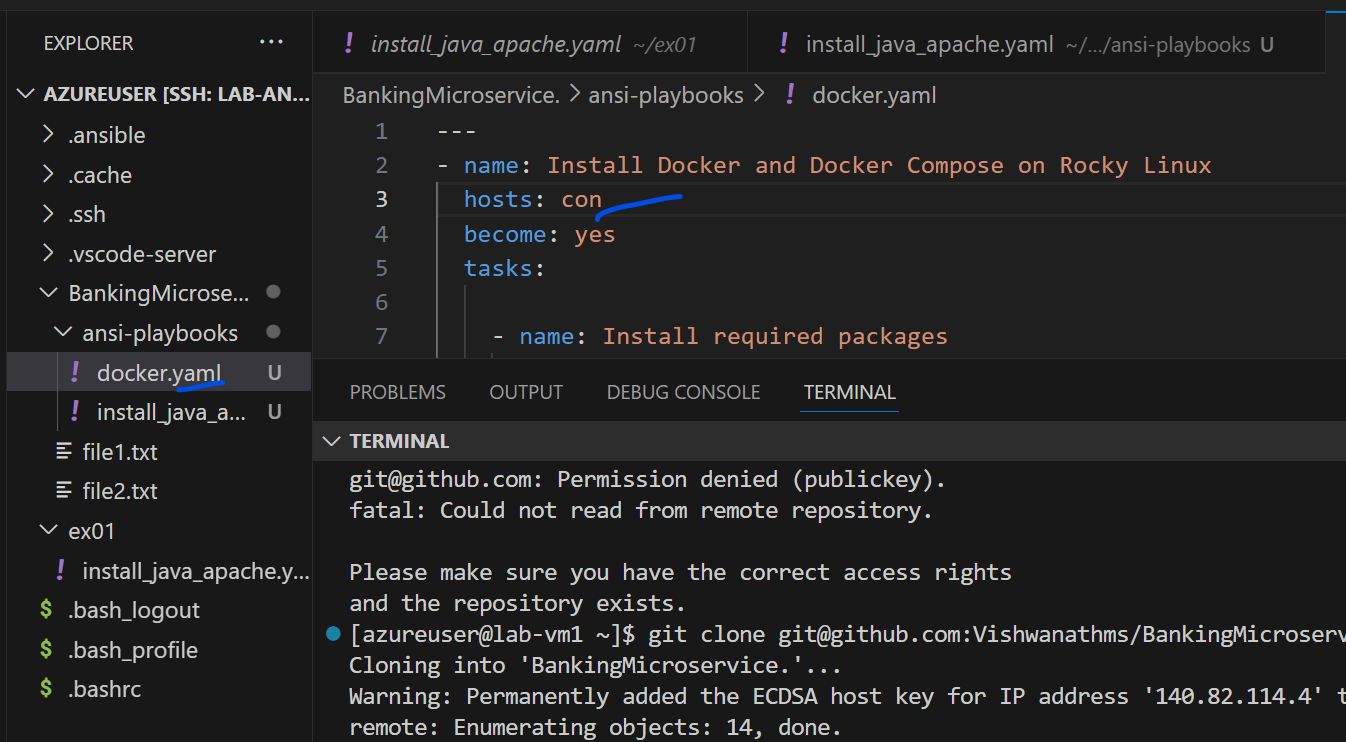
The playbook is working fine

And on the client – httpd is installed.

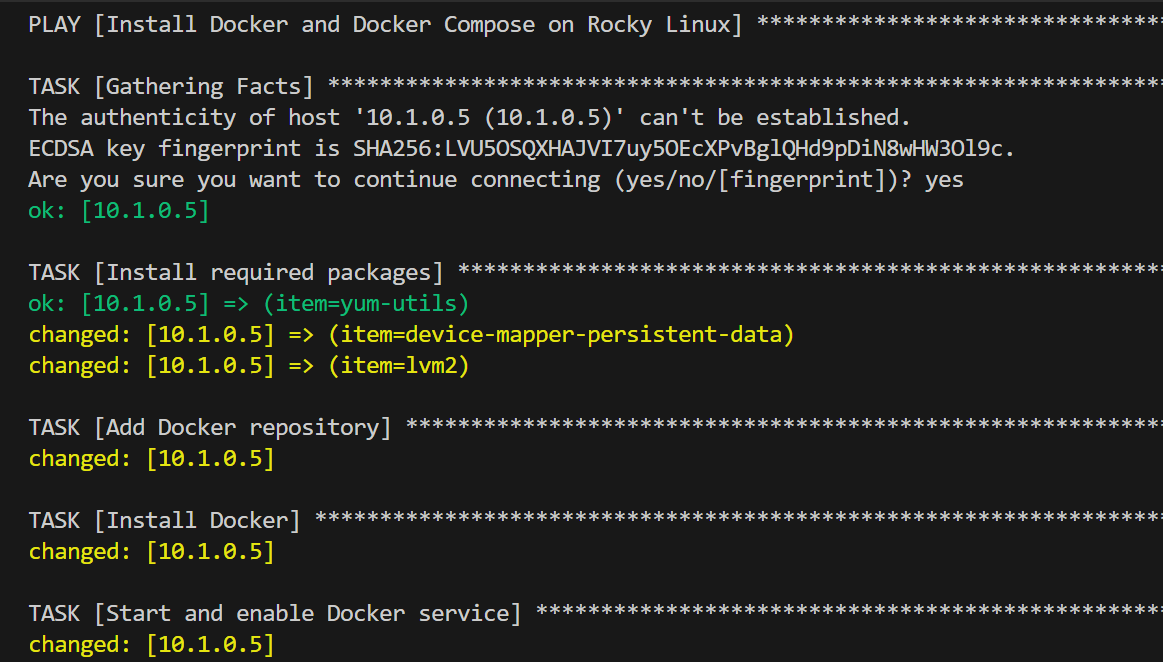
* + Start the Apache service on the ansible Client.

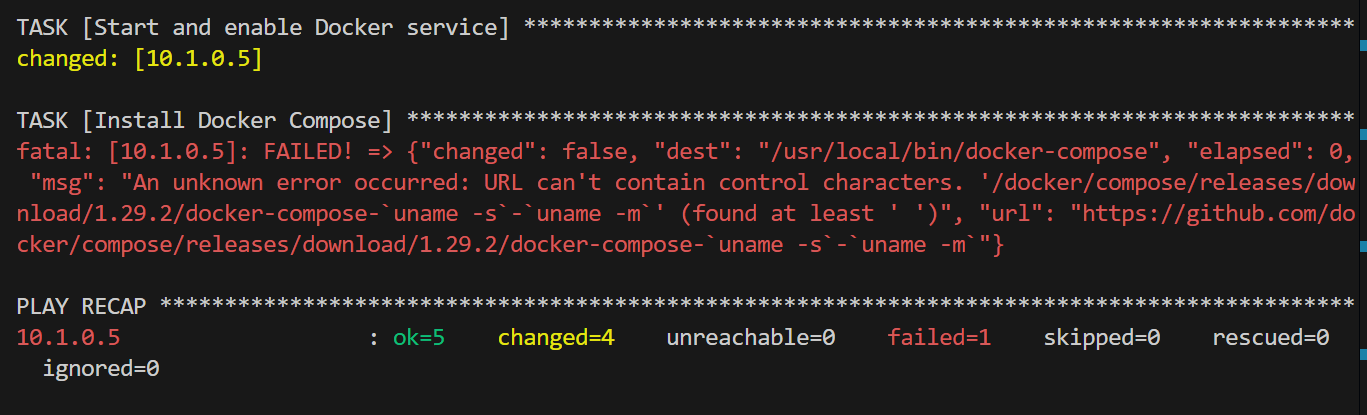


* + Install Docker and docker-compose on the Ansible Controller Linux machines.

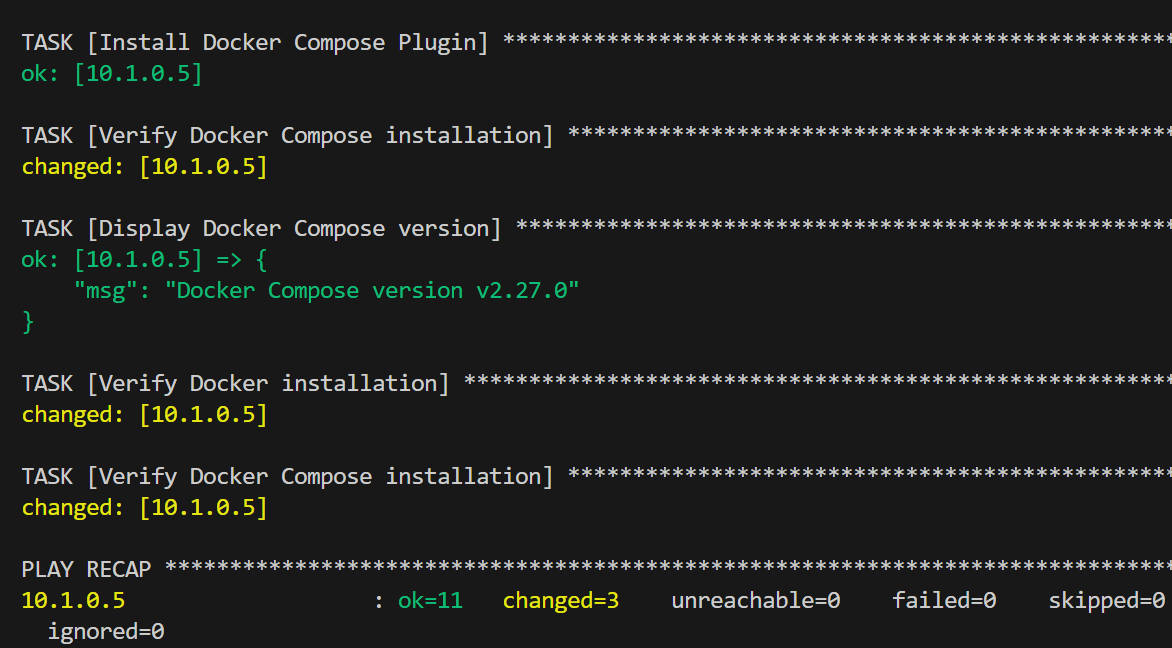


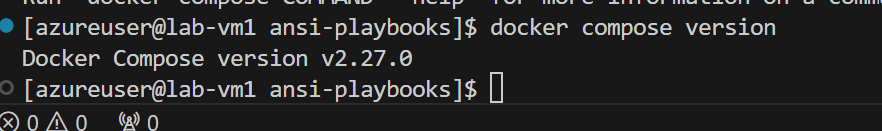
Installing only on the controller

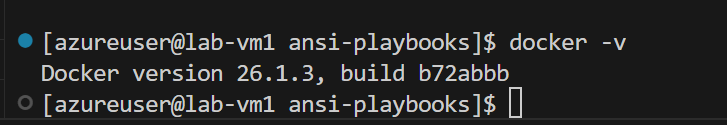




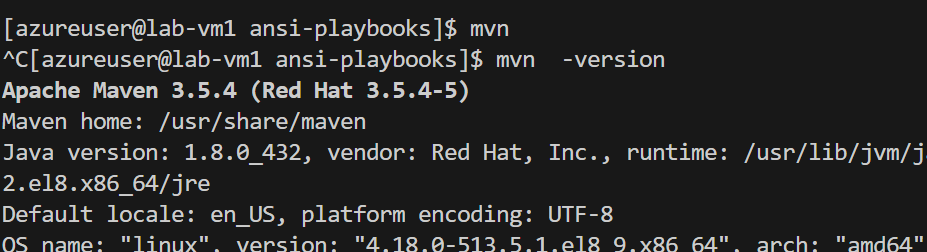
Initially used docker-compose binary files , but due to some special characters in the command, shifted to direct repo installation as below





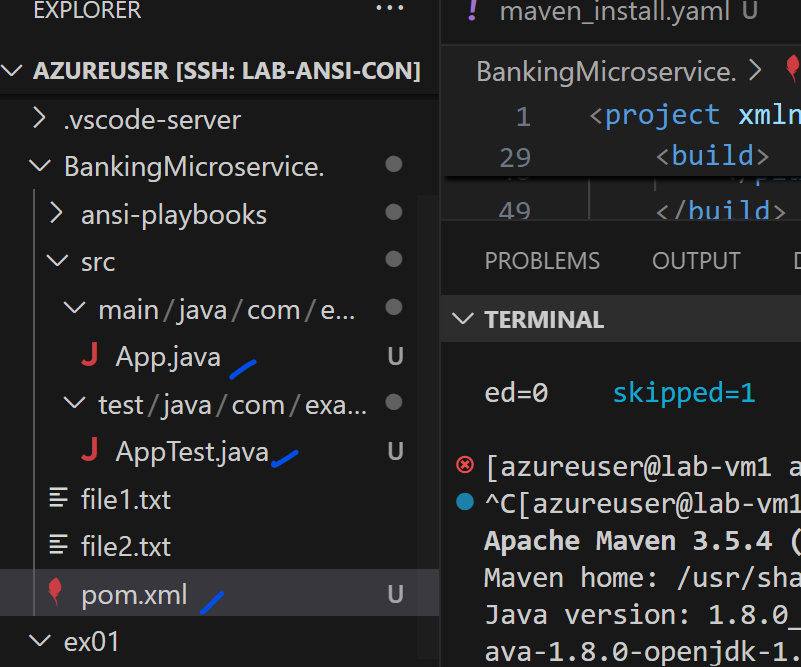


Create Maven playbook as well to install maven on both the ansible controller and client.



**Maven Requirements**

1. **Build Automation:**
   * The project is a Maven-based microservice featuring a main file named **App.java**.
   * A unit test program is located in the src/test/ folder.



Created the sample codes

* + Code should be built automatically as soon as updates are made to the **Master** branch.
  + Required dependencies, such as the JUnit framework, must be added automatically.
  + A JAR file should be generated in the target folder upon successful builds.



The build is success full.

