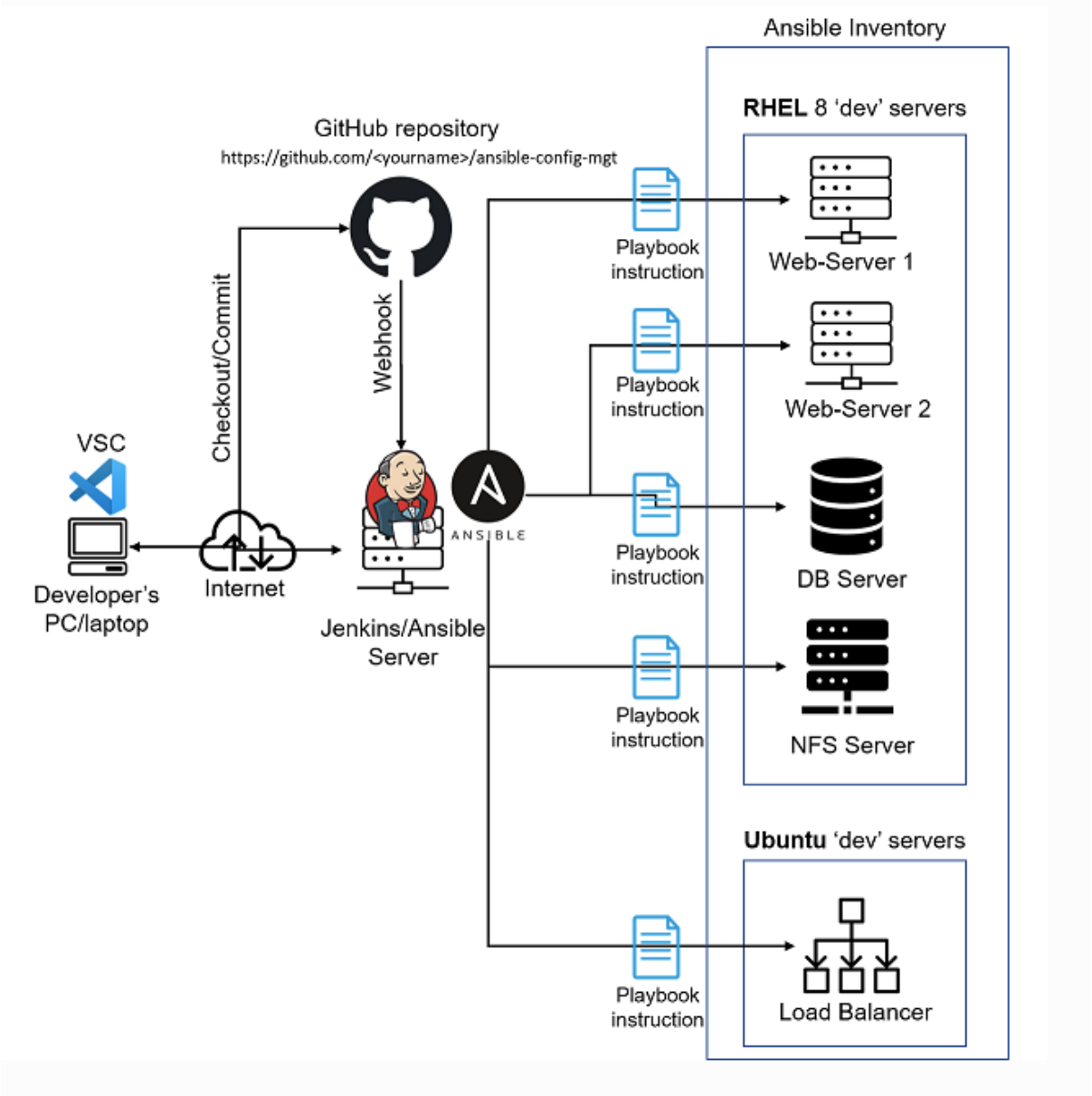
PROJECT 11
ANSIBLE_AUTOMATE_PROJECT_7_TO_10

Task:

- Install and configure Ansible client to act as a Jump Server/Bastion Host
- Create a simple Ansible playbook to automate servers configuration



Installing Ansible on Jenkins Server

We install ansible on our jenkins server and rename it to `Jenkins-Ansible`

```
sudo apt update

sudo apt install ansible
```

1. Update `Name` tag on your `Jenkins` EC2 Instance to `Jenkins-Ansible`. We will use this server to run playbooks.

Check your Ansible version by running `ansible --version`

```
ubuntu@ip-172-31-38-178:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, May 26 2023, 14:05:08) [GCC 9.4.0]
ubuntu@ip-172-31-38-178:~$
```

2. Configure Jenkins build job to save your repository content every time you change it – this will solidify your Jenkins configuration skills acquired in Project 9.
3. Create a new Freestyle project `ansible` in Jenkins and point it to your 'ansible-config-mgt' repository.
4. Configure Webhook in GitHub and set webhook to trigger `ansible` build.
5. Configure a Post-build job to save all (`**`) files, like you did in Project 9.
6. Test your setup by making some change in README.MD file in
7. `master`branch and make sure that builds starts automatically and Jenkins saves the files (build artifacts) in following folder

Console Output

```
Started by GitHub push by ovaga
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/ansible
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/ansible/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/ovaga/ansible-config-mgt.git # timeout=10
Fetching upstream changes from https://github.com/ovaga/ansible-config-mgt.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > git fetch --tags --force --progress -- https://github.com/ovaga/ansible-config-mgt.git +refs/heads/*:ref
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 3204b6219741484c80a7427bee7ef0ef8f293a4b (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 3204b6219741484c80a7427bee7ef0ef8f293a4b # timeout=10
Commit message: "update2"
 > git rev-list --no-walk a7763a47104886020d701bec100afab31aa9ddb1 # timeout=10
Archiving artifacts
Finished: SUCCESS
```

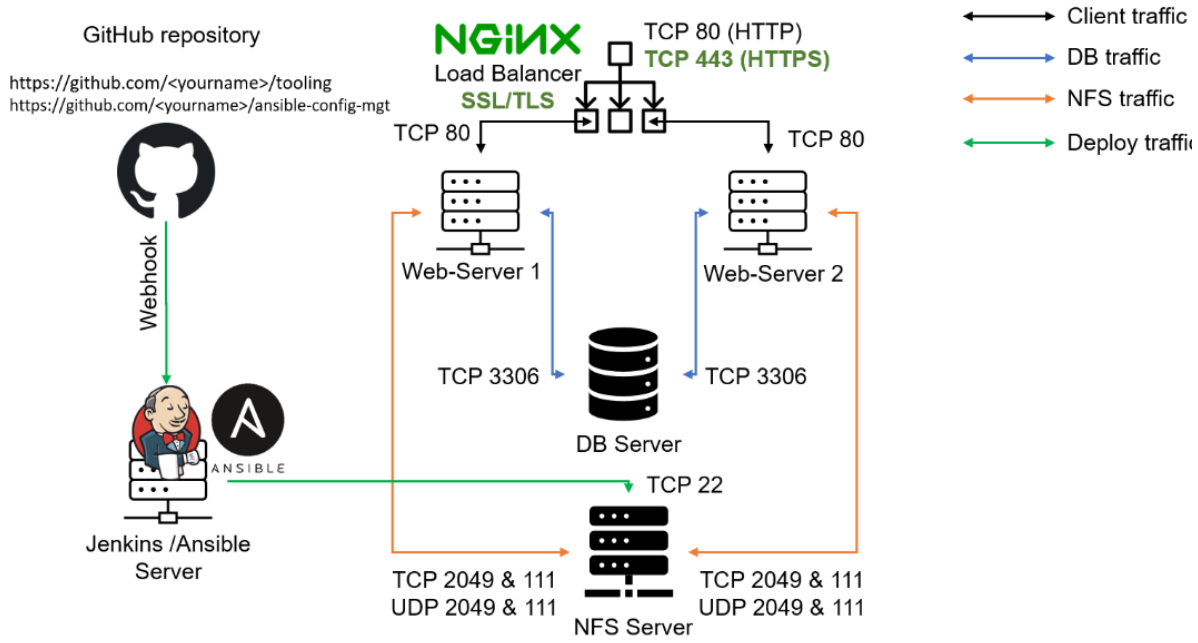
`ls /var/lib/jenkins/jobs/ansible/builds/<build_number>/archive/`

```
ubuntu@ip-172-31-38-178:~$ ls /var/lib/jenkins/jobs/ansible/builds
1  2  3  4  legacyIds  permalinks
ubuntu@ip-172-31-38-178:~$ cd /var/lib/jenkins/jobs/ansible/builds/4/archive/
ubuntu@ip-172-31-38-178:/var/lib/jenkins/jobs/ansible/builds/4/archive$ cat README.md
# ansible-config-mgt
Another test again

Test

My second build test
ubuntu@ip-172-31-38-178:/var/lib/jenkins/jobs/ansible/builds/4/archive$
```

Now this is the current architecture;



Prepare Development using VSCode

Download and install vscode which will be used to write and edit code.

After you have successfully installed VSC, configure it to connect to your newly created GitHub repository

Clone down your ansible-config-mgt repo to your Jenkins-Ansible instance

```
git clone <ansible-config-mgt repo link>
```

Ansible Configuration

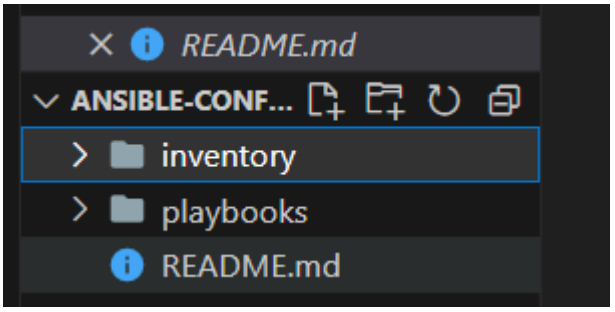
Clone `ansible-config-mgt` repo on local machine and create a new branch for development.

- Checkout the newly created feature branch to your local machine and start building your code and directory structure;

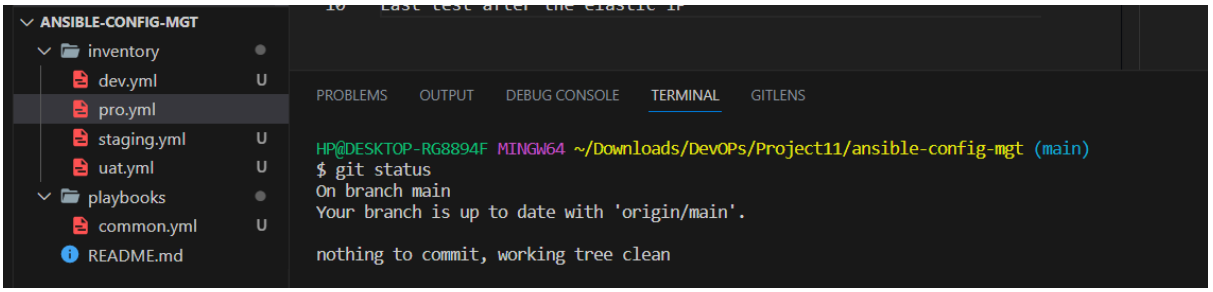
```
HP@DESKTOP-RG8894F MINGW64 ~/Downloads/DevOPs/Project11/ansible-config-mgt (main)
$ git checkout -b prj-11
Switched to a new branch 'prj-11'

HP@DESKTOP-RG8894F MINGW64 ~/Downloads/DevOPs/Project11/ansible-config-mgt (prj-11)
$
```

- Create a directory and name it `playbooks` – it will be used to store all your playbook files
- Create a directory and name it `inventory` – it will be used to keep your hosts organised.



- In the playbooks folder, create a common.yml file
- In the inventory folder, create dev.yml, prod.yml, staging.yml and uat.yml for dev, prod, staging and uat environments respectively.



Setting Up Inventory

we create inventories to execute Linux commands on remote hosts, and ensure that it is the intended configuration on a particular server that occurs. It is important to have a way to organize our hosts in such an Inventory.

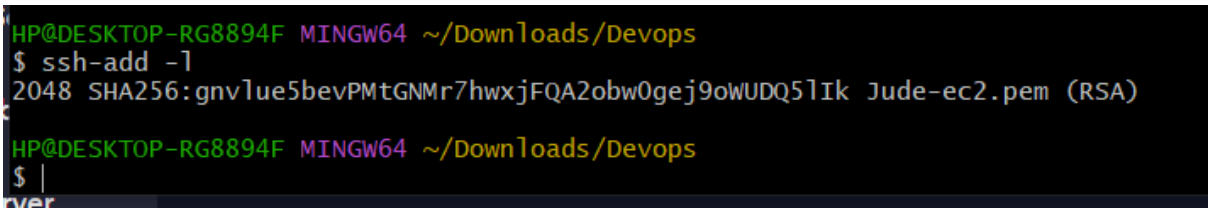
We need to ssh into our target servers defined in the /inventory/dev.yaml

using SSH-Agent to upload our ssh public key to the jenkins-ansible server

```
eval `ssh-agent -s`
```

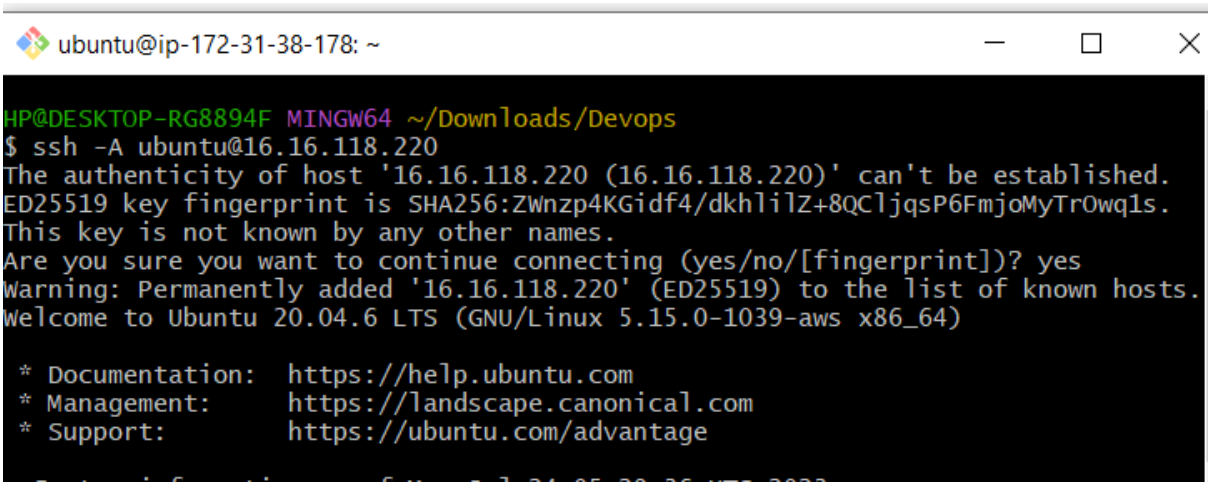
```
ssh-add <path-to-private-key>
```

Confirm the key has been added with the command below, you should see the name of your key;



- Now, ssh into your `Jenkins-Ansible` server using ssh-agent;

```
ssh -A ubuntu@public-ip
```



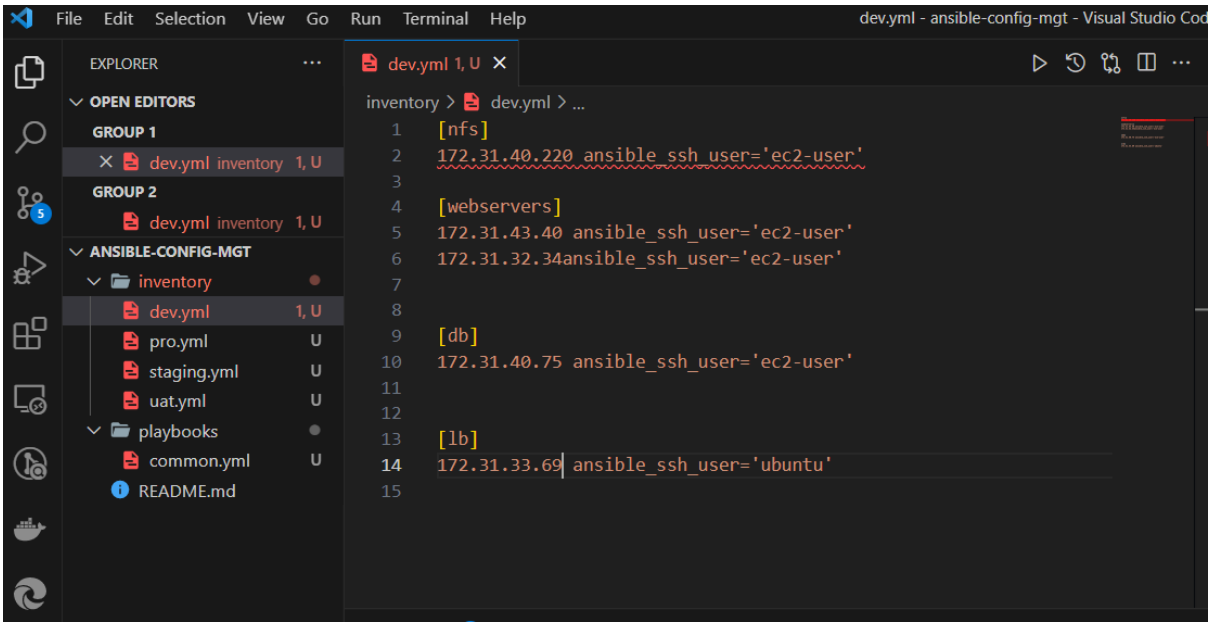
- Update your `inventory/dev.yml` file with this snippet of code:

```
[nfs]
<NFS-Server-Private-IP-Address> ansible_ssh_user='ec2-user'

[webservers]
<Web-Server1-Private-IP-Address> ansible_ssh_user='ec2-user'
<Web-Server2-Private-IP-Address> ansible_ssh_user='ec2-user'

[db]
<Database-Private-IP-Address> ansible_ssh_user='ec2-user'

[lb]
<Load-Balancer-Private-IP-Address> ansible_ssh_user='ubuntu'
```



CREATE A COMMON PLAYBOOK

- Update code in /playbooks/common.yaml

It is time to start giving Ansible the instructions on what you needs to be performed on all servers listed in `inventory/dev`

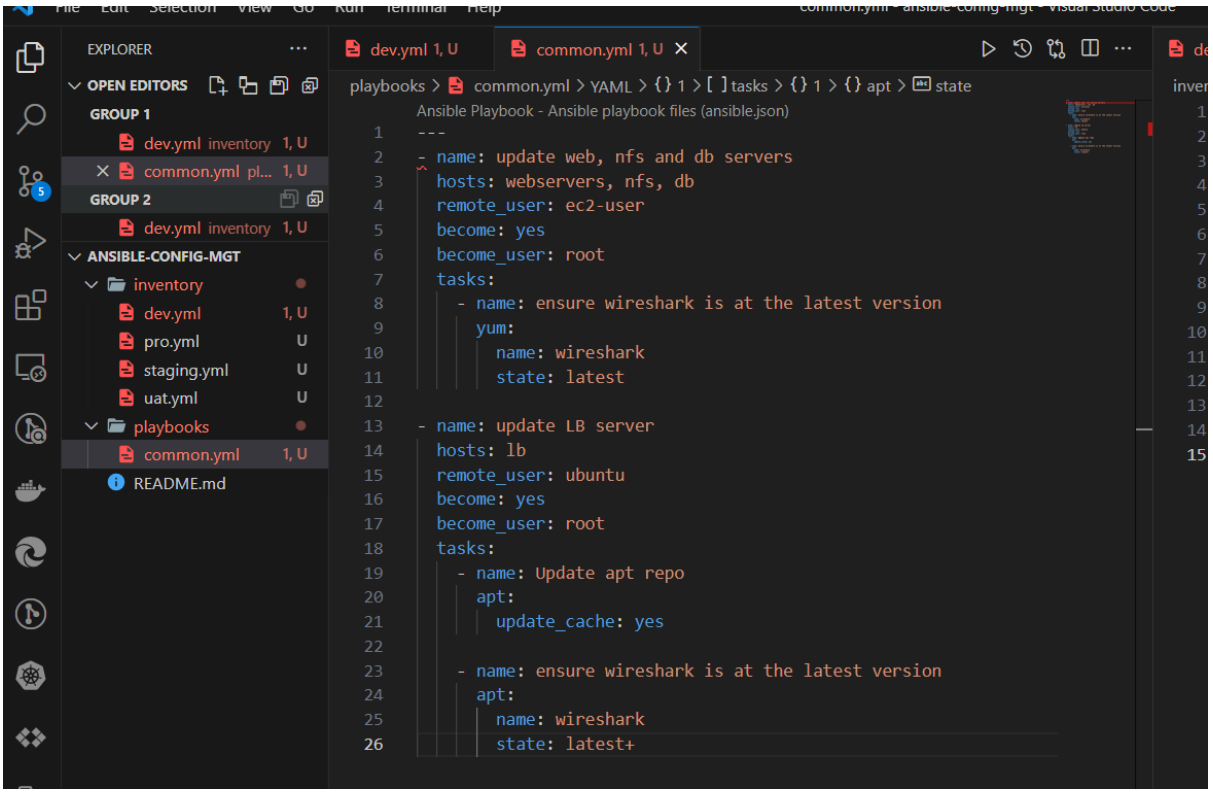
In `common.yml` playbook you will write configuration for repeatable, re-usable, and multi-machine tasks that is common to systems within the infrastructure.

Update your `playbooks/common.yml` file with following code:

```
---
- name: update web, nfs and db servers
  hosts: webservers, nfs, db
  remote_user: ec2-user
  become: yes
  become_user: root
  tasks:
    - name: ensure wireshark is at the latest version
      yum:
        name: wireshark
        state: latest

- name: update LB server
  hosts: lb
  remote_user: ubuntu
  become: yes
  become_user: root
  tasks:
    - name: Update apt repo
      apt:
        update_cache: yes

    - name: ensure wireshark is at the latest version
      apt:
        name: wireshark
        state: latest
```



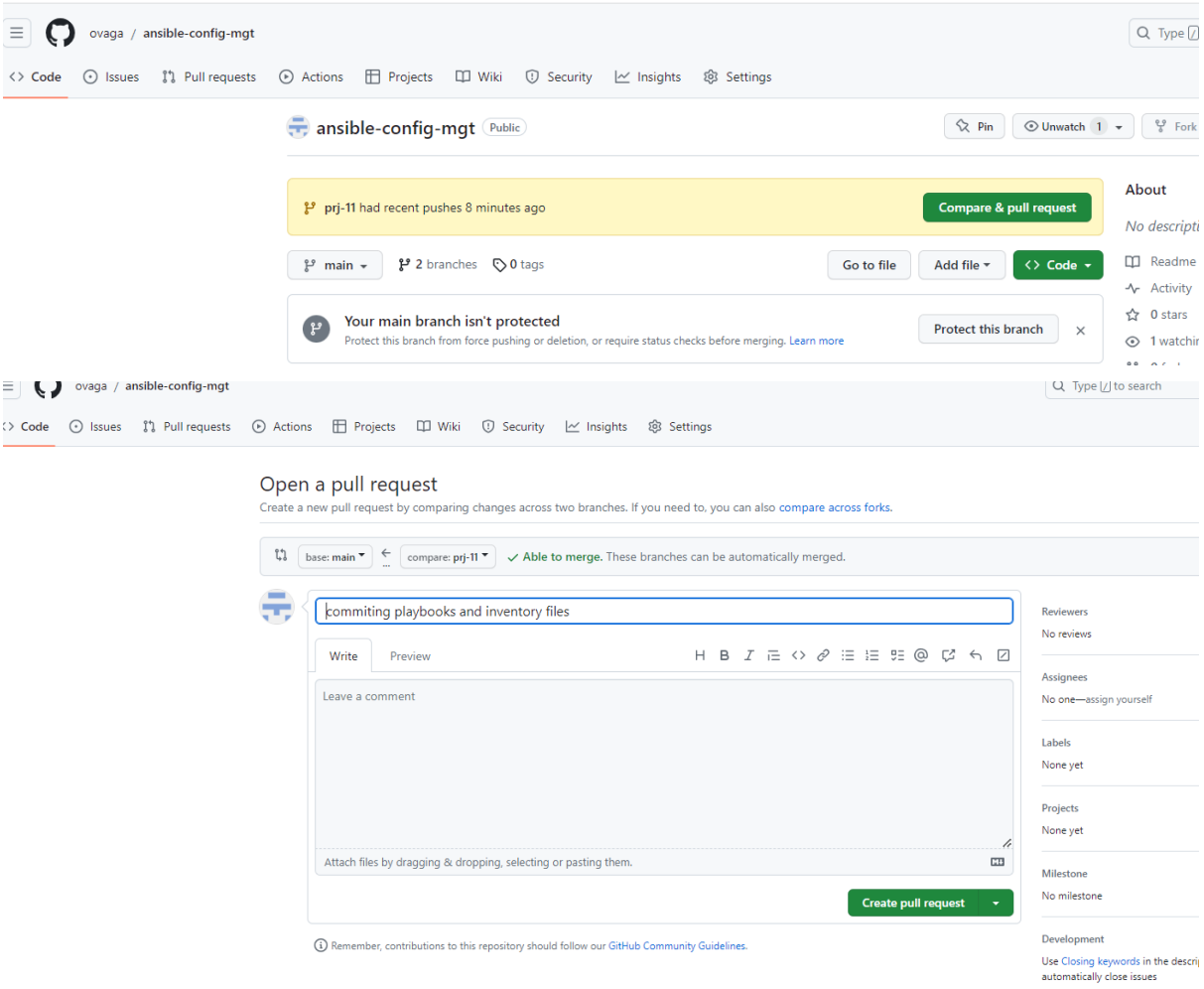
Update GIT with the latest code

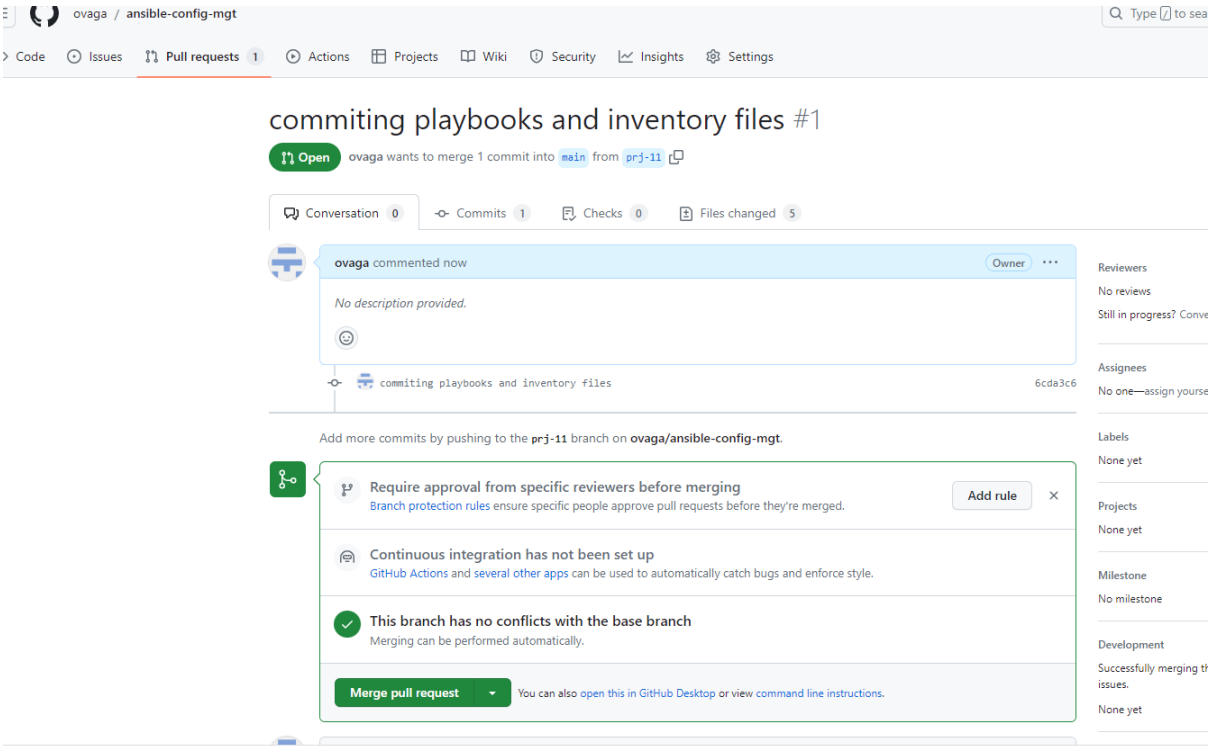
```
git status
```

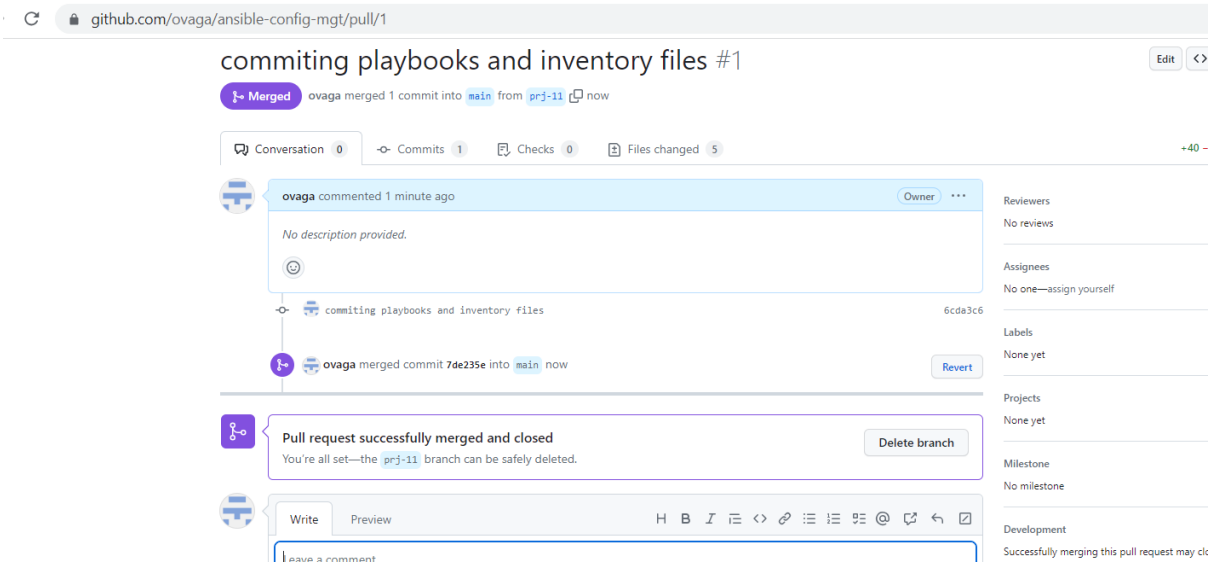
```
git add <selected files>
```

```
git commit -m "commit message"
```

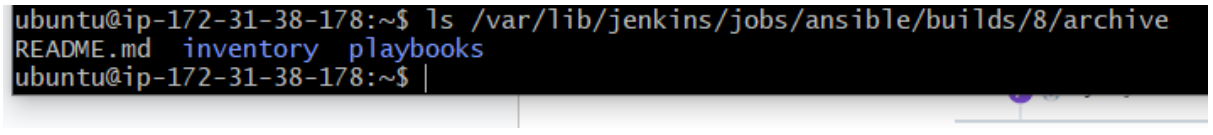
2. Create a Pull request (PR)





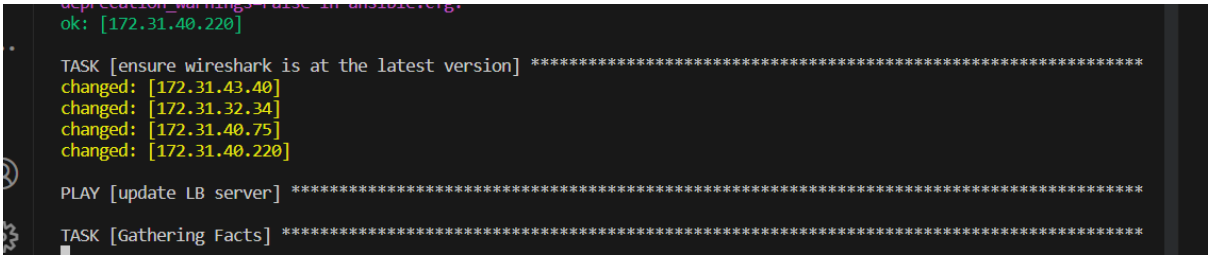


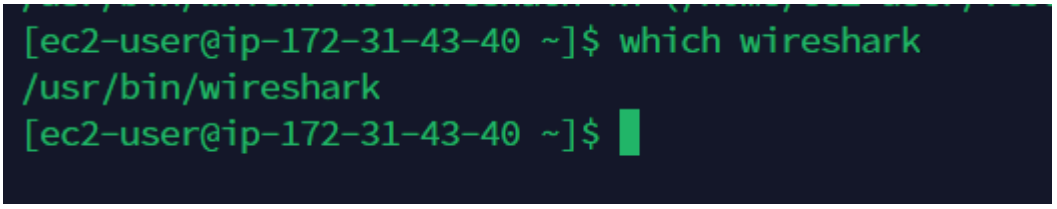
Once your code changes appear in `master` branch – Jenkins will do its job and save all the files (build artifacts) to `/var/lib/jenkins/jobs/ansible/builds/<build_number>/archive/` directory on `Jenkins-Ansible` server.

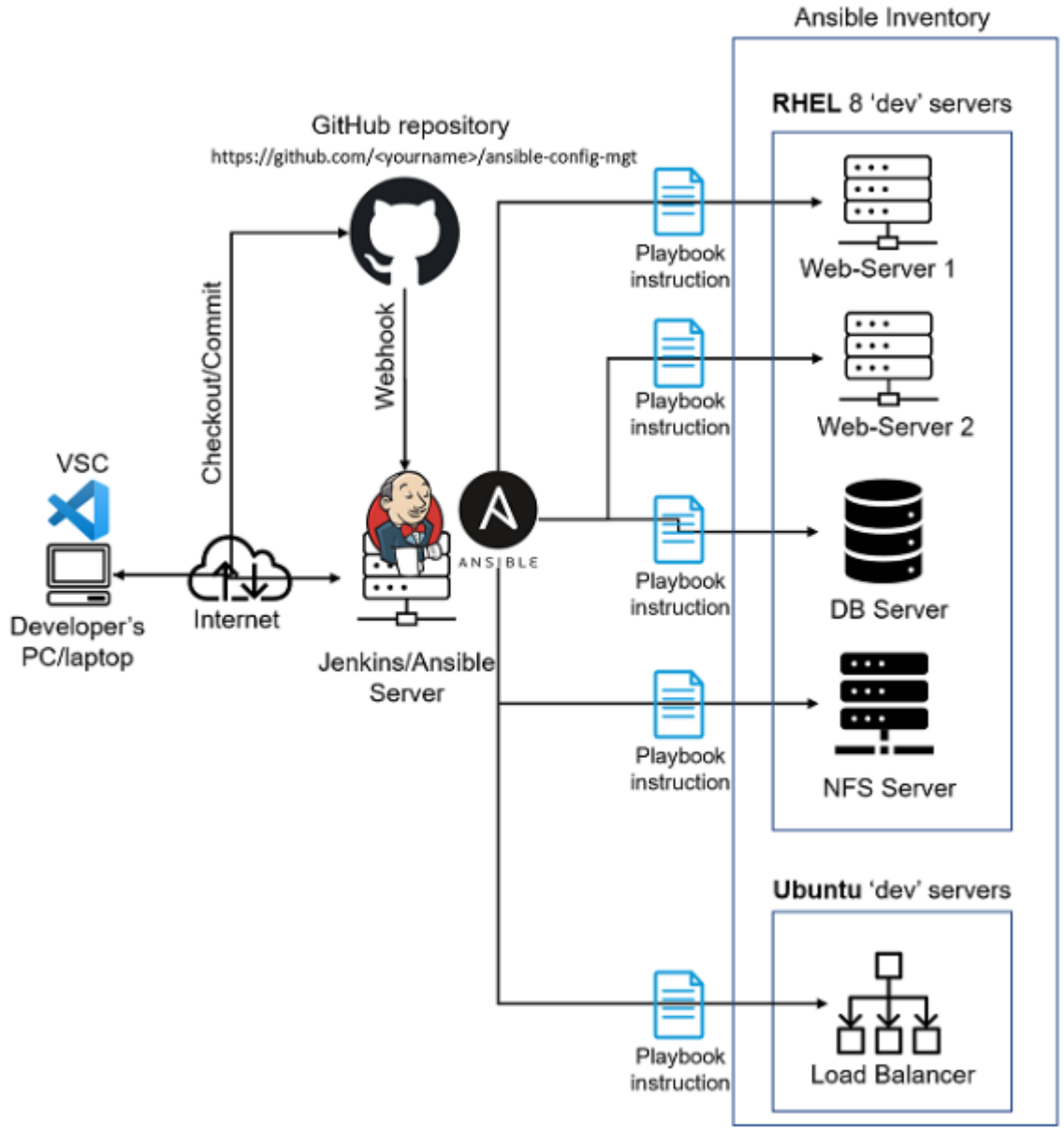


RUN FIRST ANSIBLE TEST

```
ansible-playbook -i
/var/lib/jenkins/jobs/ansible/builds/<build-number>/archive/inventory/dev.yml
/var/lib/jenkins/jobs/ansible/builds/<build-number>/archive/playbooks/common.
yml
```






Ansible Inventory
RHEL 8 'dev' servers
GitHub repository
https://github.com/<yourname>/ansible-config-mgt
Playbook instruction
Web-Server 1
Playbook instruction
Web-Server 2
Checkout/Commit
Webhook
VSC
Playbook instruction
DB Server
ANSIBLE
Developer's PC/laptop
Internet
Jenkins/Ansible Server
Playbook instruction
NFS Server
Ubuntu 'dev' servers
Playbook instruction
Load Balancer