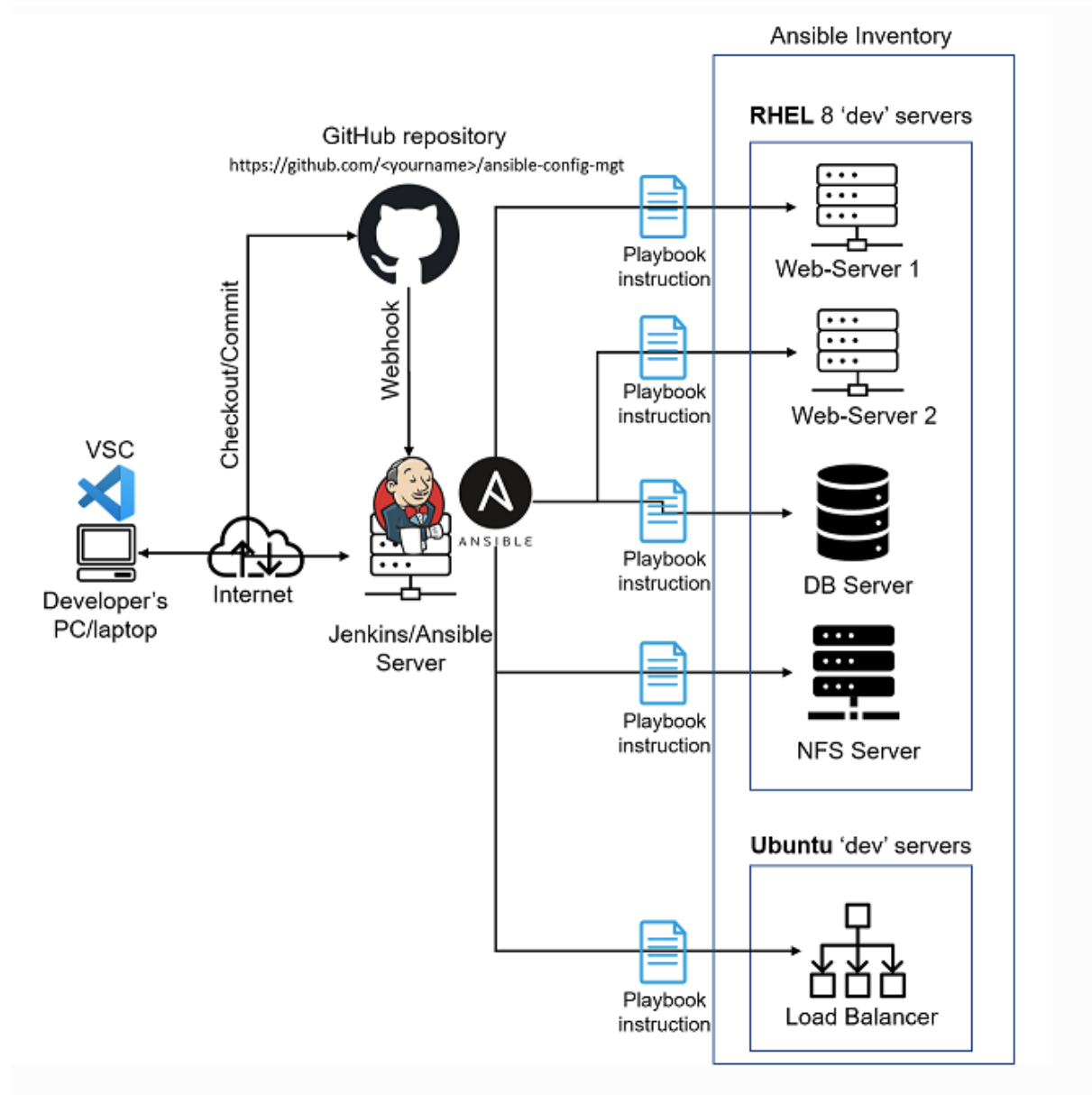


PROJECT 11
ANSIBLE_AUTOMATE_PROJECT_7_TO_10

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

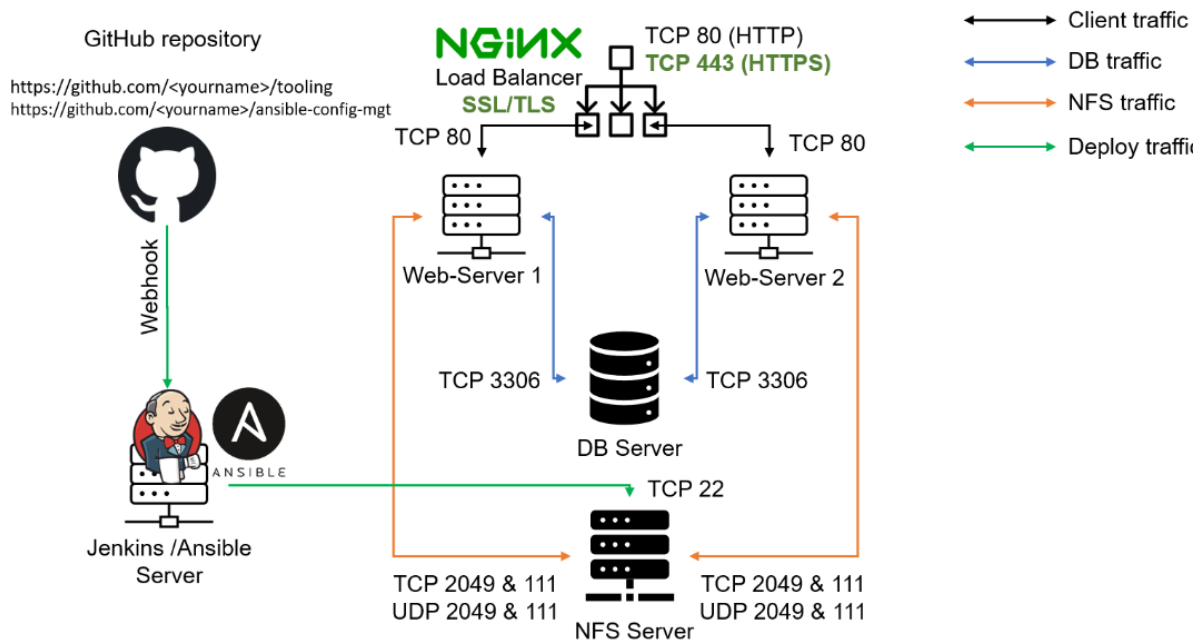
```
Started by GitHub push by ovaga
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/ansible
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/ansible/.git # timeout=10
Fetching changes from the remote Git repository
> 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/*:refs/heads/*
> 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

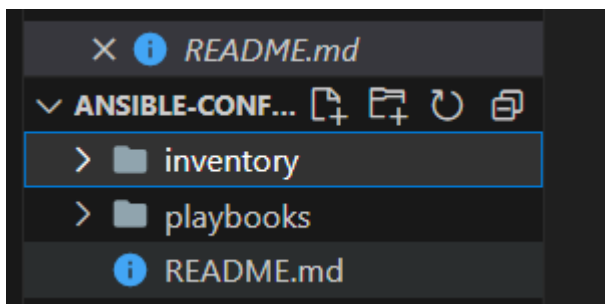
Clone `ansible-config-mgt` repo on local machine and create a new branch for development.

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

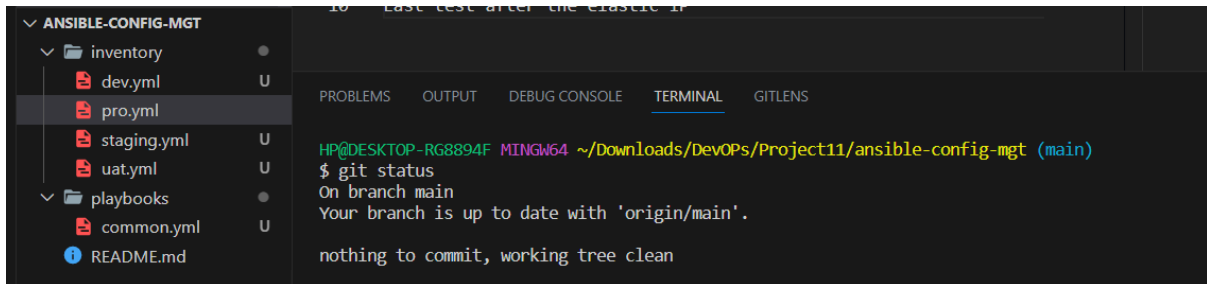
```
HP@DESKTOP-RG8894F MINGW64 ~/Downloads/DevOPs/Project11/ansible-config-mgt (main)
$ git checkout -b prj-11
Switched to a new branch 'prj-11'

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.yml

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

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

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

```
HP@DESKTOP-RG8894F MINGW64 ~/Downloads/Devops
$ eval `ssh-agent -s`
Agent pid 2220

HP@DESKTOP-RG8894F MINGW64 ~/Downloads/Devops
$ ssh-add C:\Users\HP\Downloads\DevOps
C:\Users\HP\Downloads\DevOps: No such file or directory

HP@DESKTOP-RG8894F MINGW64 ~/Downloads/Devops
$ ssh-add Jude-ec2.pem
Identity added: Jude-ec2.pem (Jude-ec2.pem)

HP@DESKTOP-RG8894F MINGW64 ~/Downloads/Devops
$ ssh-add -l
2048 SHA256:gnv1ue5bevPMtGNMr7hwxjFQA2obw0gej9owUDQ51Ik Jude-ec2.pem (RSA)

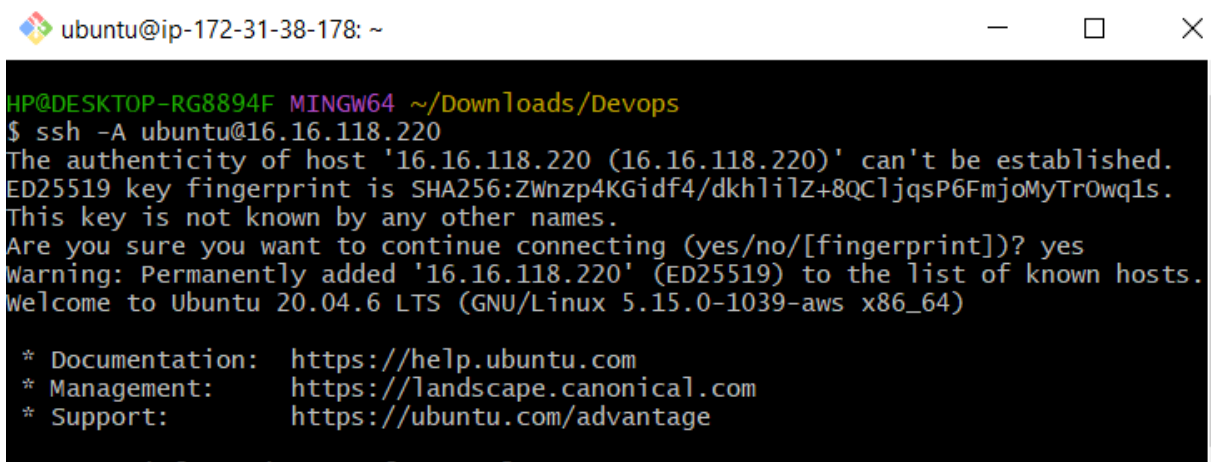
HP@DESKTOP-RG8894F MINGW64 ~/Downloads/Devops
$
```

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

```
HP@DESKTOP-RG8894F MINGW64 ~/Downloads/Devops
$ ssh-add -l
2048 SHA256:gnv1ue5bevPMtGNMr7hwxjFQA2obw0gej9oWUDQ51Ik Jude-ec2.pem (RSA)
HP@DESKTOP-RG8894F MINGW64 ~/Downloads/Devops
$ |
over
```

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

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



The image shows a terminal window titled 'ubuntu@ip-172-31-38-178: ~'. The terminal output shows a user at a Windows prompt running 'ssh -A ubuntu@16.16.118.220'. The output includes a warning about the host's authenticity, a confirmation to continue, and a welcome message from Ubuntu 20.04.6 LTS. It also lists documentation, management, and support URLs.

```
ubuntu@ip-172-31-38-178: ~
HP@DESKTOP-RG8894F MINGW64 ~/Downloads/Devops
$ ssh -A ubuntu@16.16.118.220
The authenticity of host '16.16.118.220 (16.16.118.220)' can't be established.
ED25519 key fingerprint is SHA256:ZWnzp4KGidf4/dkh1ilZ+8QCljqsP6FmjoMyTrOwq1s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '16.16.118.220' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1039-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
```

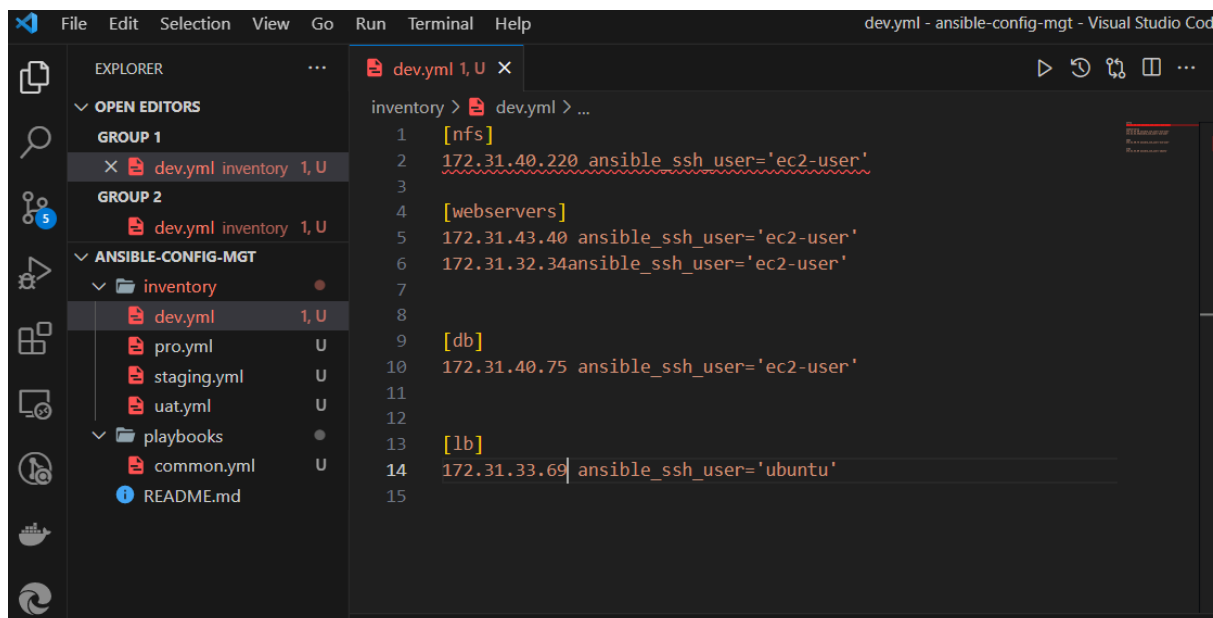
- Update your `inventory/dev.yml` file with this snippet of code:

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

[webserver]
<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.yml`

It is time to start giving Ansible the instructions on what you need to be performed on all servers listed in

`inventory/dev`

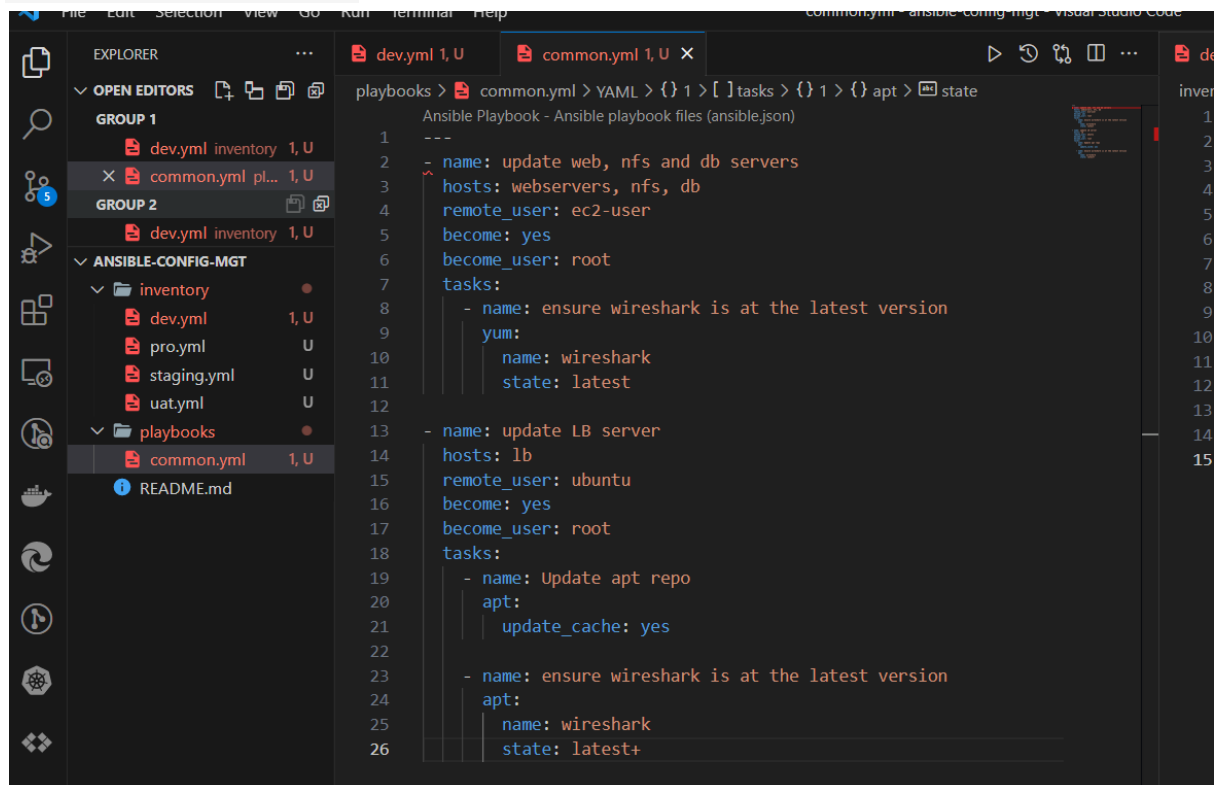
In `common.yml` playbook you will write configuration for repeatable, re-usable, and multi-machine tasks that is common to systems within the infrastructure.

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

```
---
- name: update web, nfs and db servers
  hosts: webserver, 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)

The screenshot shows the GitHub interface for the repository 'ansible-config-mgt' by user 'ovaga'. The repository is public. At the top, there's a navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there's a header section with the repository name, a 'Public' badge, and buttons for Pin, Unwatch (1), and Fork. A yellow banner indicates that 'prj-11' had recent pushes 8 minutes ago, with a 'Compare & pull request' button. Below this, there's a section for the 'main' branch, showing 2 branches and 0 tags, with buttons for 'Go to file', 'Add file', and 'Code'. A warning message states 'Your main branch isn't protected' and provides a 'Protect this branch' button. On the right side, there's an 'About' section with a 'No description' note and links to Readme, Activity, 0 stars, and 1 watcher.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

The screenshot shows the 'Open a pull request' form in GitHub. At the top, there's a comparison bar showing 'base: main' and 'compare: prj-11' with a green checkmark indicating 'Able to merge'. Below this, there's a text input field with the title 'committing playbooks and inventory files'. The form has two tabs: 'Write' and 'Preview'. The 'Write' tab is active, showing a large text area for 'Leave a comment' and a section for 'Attach files by dragging & dropping, selecting or pasting them.' At the bottom right of the form is a 'Create pull request' button. On the right side of the form, there's a sidebar with sections for 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), and 'Development' (Use Closing keywords in the description to automatically close issues).

ovaga / ansible-config-mgt

Q Type to sea

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

committing playbooks and inventory files #1

Open ovaga wants to merge 1 commit into `main` from `prj-11`

Conversation 0 Commits 1 Checks 0 Files changed 5

ovaga commented now Owner ...

No description provided.

committing playbooks and inventory files 6cda3c6

Add more commits by pushing to the `prj-11` branch on `ovaga/ansible-config-mgt`.

Require approval from specific reviewers before merging Add rule ×

[Branch protection rules](#) ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request ▼ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Reviewers
No reviews
Still in progress? Conve

Assignees
No one—assign yourse

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging th
issues.

None yet

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
```

```
deprecation warnings: false in ansible.cfg.  
ok: [172.31.40.220]  
*  
TASK [ensure wireshark is at the latest version] *****  
changed: [172.31.43.40]  
changed: [172.31.32.34]  
changed: [172.31.40.75]  
changed: [172.31.40.220]  
PLAY [update LB server] *****  
TASK [Gathering Facts] *****
```

```
[ec2-user@ip-172-31-43-40 ~]$ which wireshark  
/usr/bin/wireshark  
[ec2-user@ip-172-31-43-40 ~]$
```

