PROJECT 12

Ansible Refactoring and Static Assignments (IMPORTS AND ROLES)

In this project we are going to work with a new repo called ansible-config-mgt repository and make some improvements on my code. So we will need to refactor our ansible code, create assignments, and learn how to use import functionality. Imports allows us to reuse previously created playbooks in a new playbook. I.e it allows you to organise your tasks and reuse them when needed.

Step1. Jenkins job enhancement
Why is this needed? It is bcos every new change in the code now creates a separate directory which is not very efficient especially when we want to run some commands from one place. Besides, it consumes space in the jenkins server. So we can enhance this by introducing a new jenkins job/project. We will require a copy of the artifact plugin.

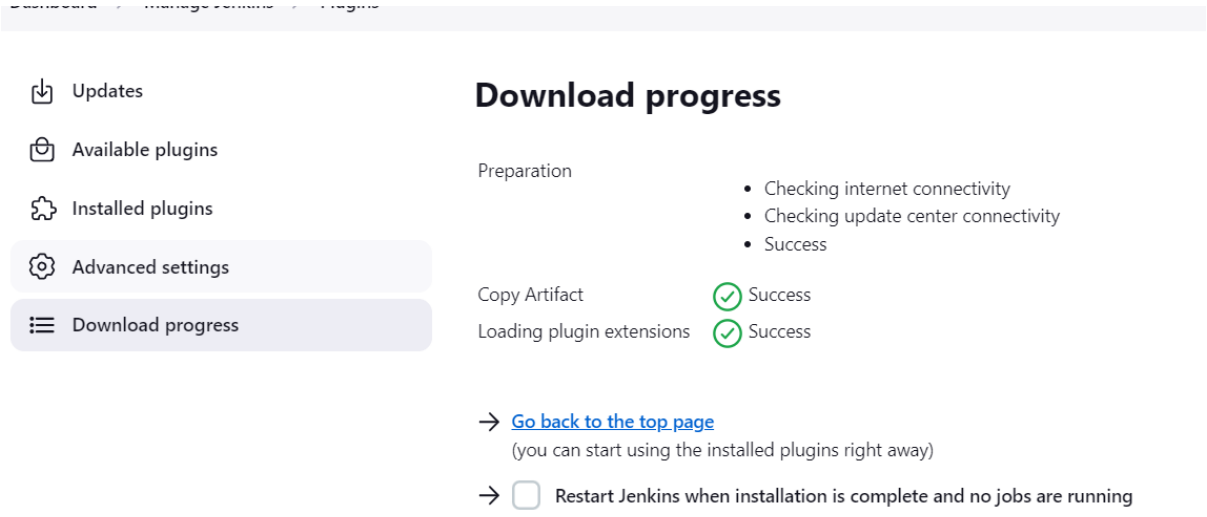
1. In jenkins create a directory called ansible-config-artifact. This is where all the artifact after build will be stored.

   sudo mkdir /home/ubuntu/ansible-config-artifact

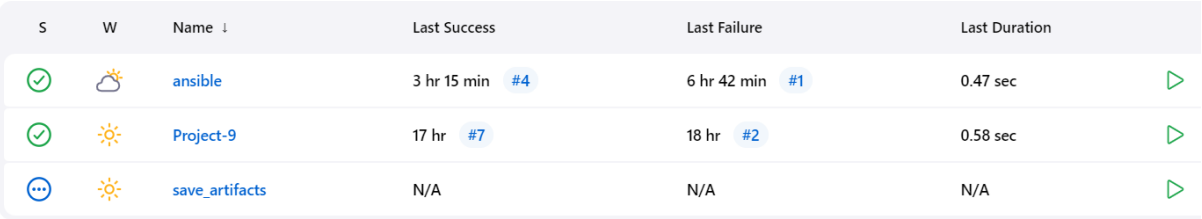
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

ubuntu@ip-172-31-24-30:~/ansible-config-artifact$ pwd
/home/ubuntu/ansible-config-artifact
ubuntu@ip-172-31-24-30:~/ansible-config-artifact$
```

2. Change permission of the directory
   chmod -R 0777 /home/ubuntu/ansible-config-artifact

3. In jenkins web console, click as follows; Manage jenkins, Manage Plugins, on available tab search for copy Artifact and install this plugin without restarting the jenkins server.

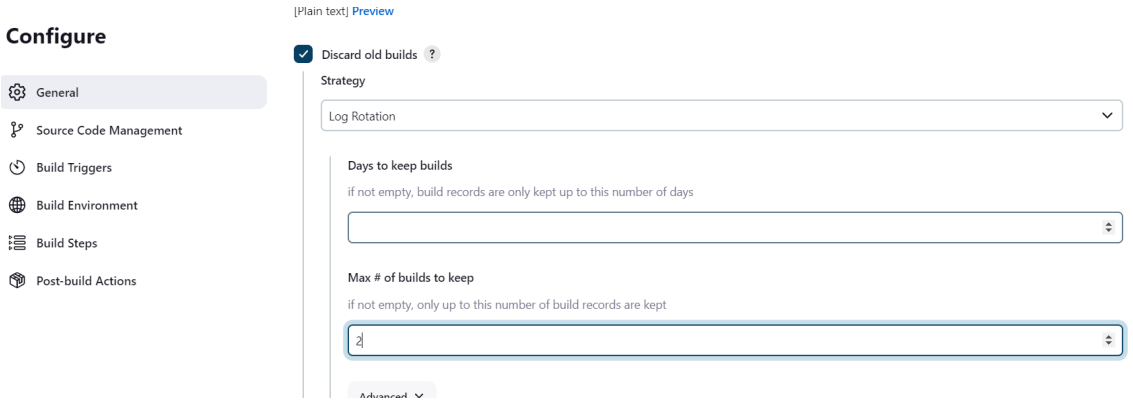


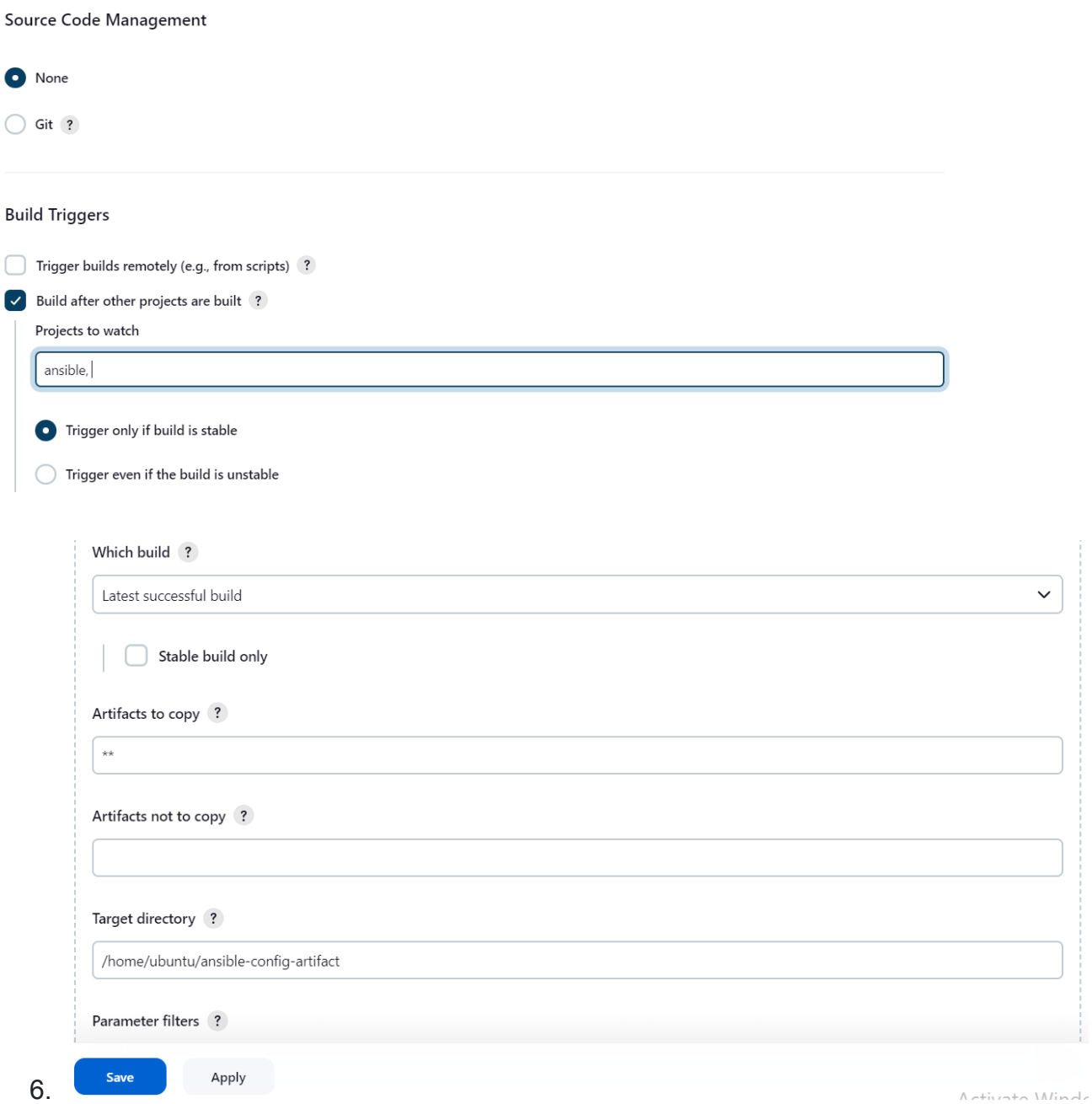
4. Create a new freestyle project and name it save_artifacts.



S	W	Name ↓	Last Success	Last Failure	Last Duration
		ansible	3 hr 15 min #4	6 hr 42 min #1	0.47 sec
		Project-9	17 hr #7	18 hr #2	0.58 sec
		save_artifacts	N/A	N/A	N/A

5. This project will be triggered by completion of your existing `ansible` project. Configure it accordingly:





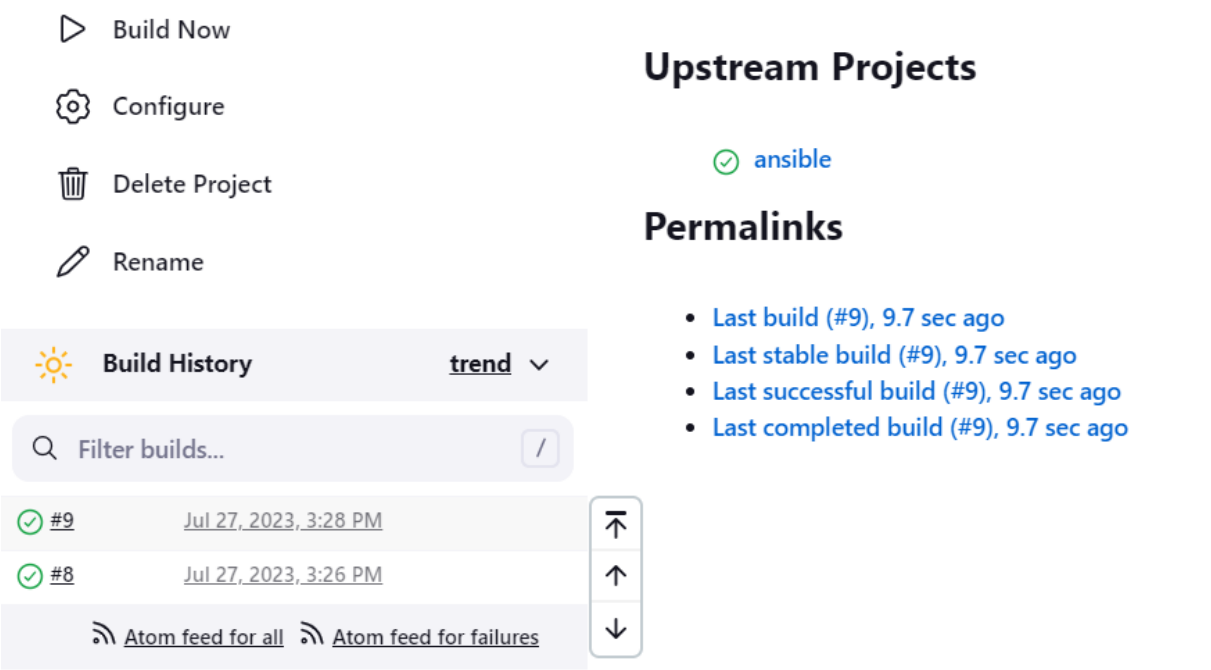
6.

We configured the number of build to 2. This is useful because whenever the jenkins pipeline runs, it creates a directory for the artifacts and it takes alot of space. By specifying the number of build, we can choose to keep only 2 of the latest builds and discard the rest.

Test your set up by making some change in README.MD file inside your ansible-config-mgt repository (right inside master/main branch).

If both Jenkins jobs have completed one after another – you shall see your files inside /home/ubuntu/ansible-config-artifact directory and it will be updated with every commit to your master branch.

```
ubuntu@ip-172-31-24-30:~/ansible-config-artifact$ pwd
/home/ubuntu/ansible-config-artifact
ubuntu@ip-172-31-24-30:~/ansible-config-artifact$ ls
README.md  inventory  playbooks
ubuntu@ip-172-31-24-30:~/ansible-config-artifact$ cat README.md
# Ansible-Pro11

test

Save Artifatc
test

test2
```



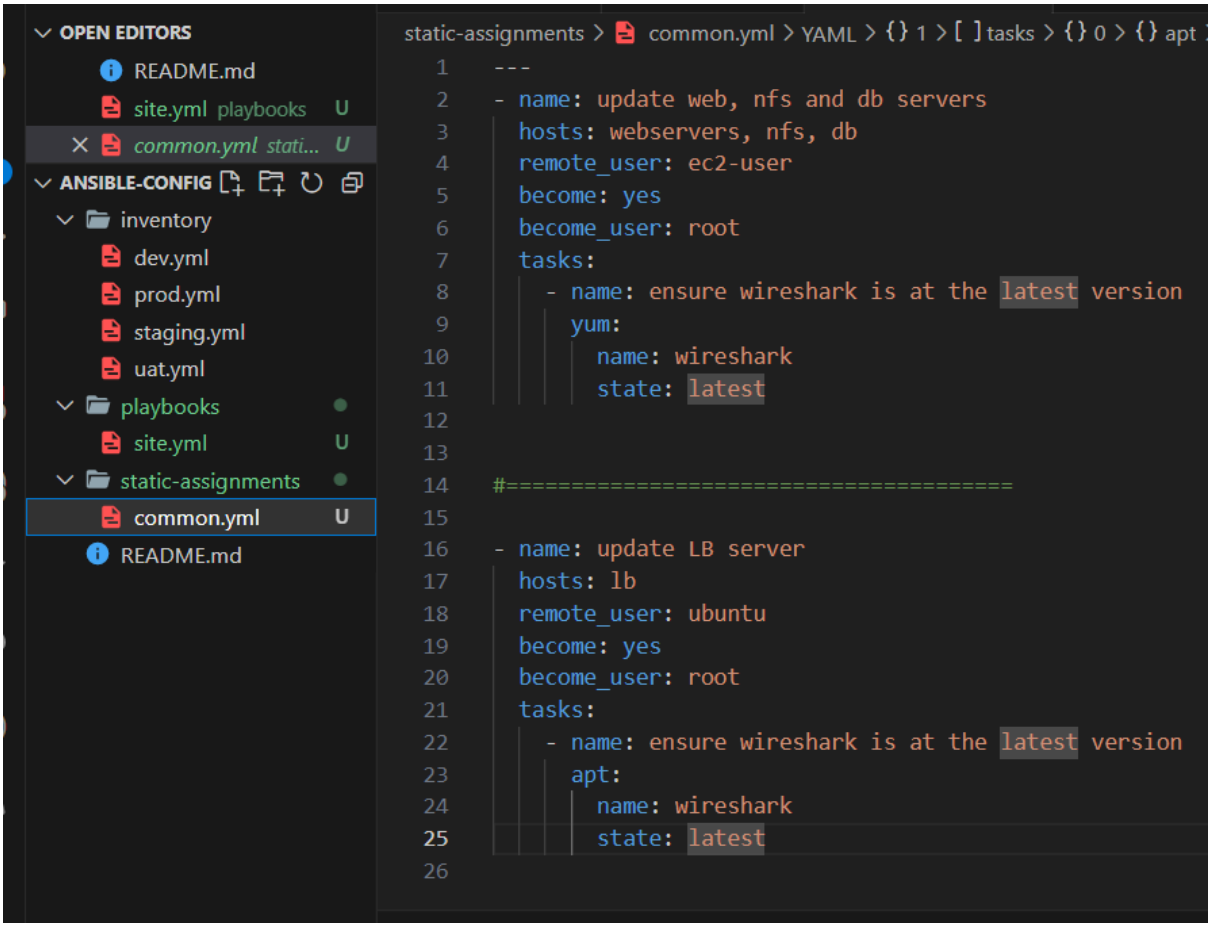
Now the Jenkins pipeline is more neat and clean.

Step2. Refactor Ansible code by importing other playbooks into site.yml
In Project 11 , I wrote all tasks in a single playbook `common.yml`, now it is pretty simple set of instructions for only 2 types of OS, but imagine you have many more tasks and you need to apply this playbook to other servers with different requirements. In this case, you will have to read through the whole playbook to check if all tasks written there are applicable and is there anything that you need to add for certain server/OS families. Very fast it will become a tedious exercise and your playbook will become messy with many commented parts. Your DevOps colleagues will not appreciate such organisation of your codes and it will be difficult for them to use your playbook.

Most ansible users learn the one-file approach first. However, breaking tasks up into different files is an excellent way to organise complex sets of tasks and reuse them.

We are going to see code re-use in action by importing other playbooks.

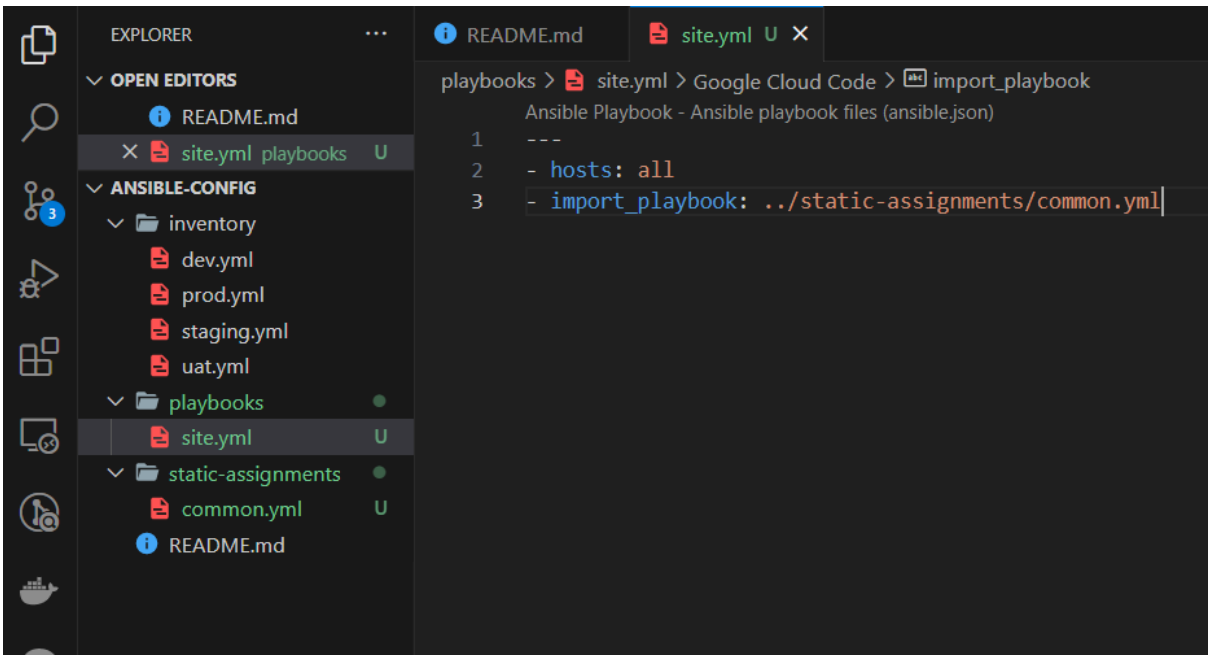
1. In playbooks folder, create a new file and name it `site.yml` – This file will now be considered as an entry point into the entire infrastructure configuration.
2. Create a new folder in root of the repository and name it static-assignments. The static-assignments folder is where all other children playbooks will be stored
3. Create a new folder in root of the repository and name it static-assignments. The static-assignments folder is where all other children playbooks will be stored
4. Move common.yml file into the newly created static-assignments folder.



5. Inside site.yml file, import common.yml playbook.

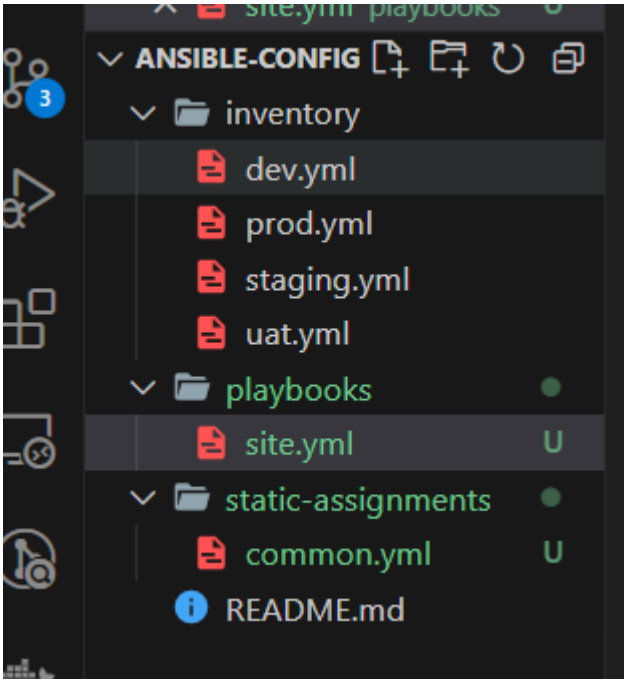
- hosts: all

- import_playbook: ../static-assignments/common.yml



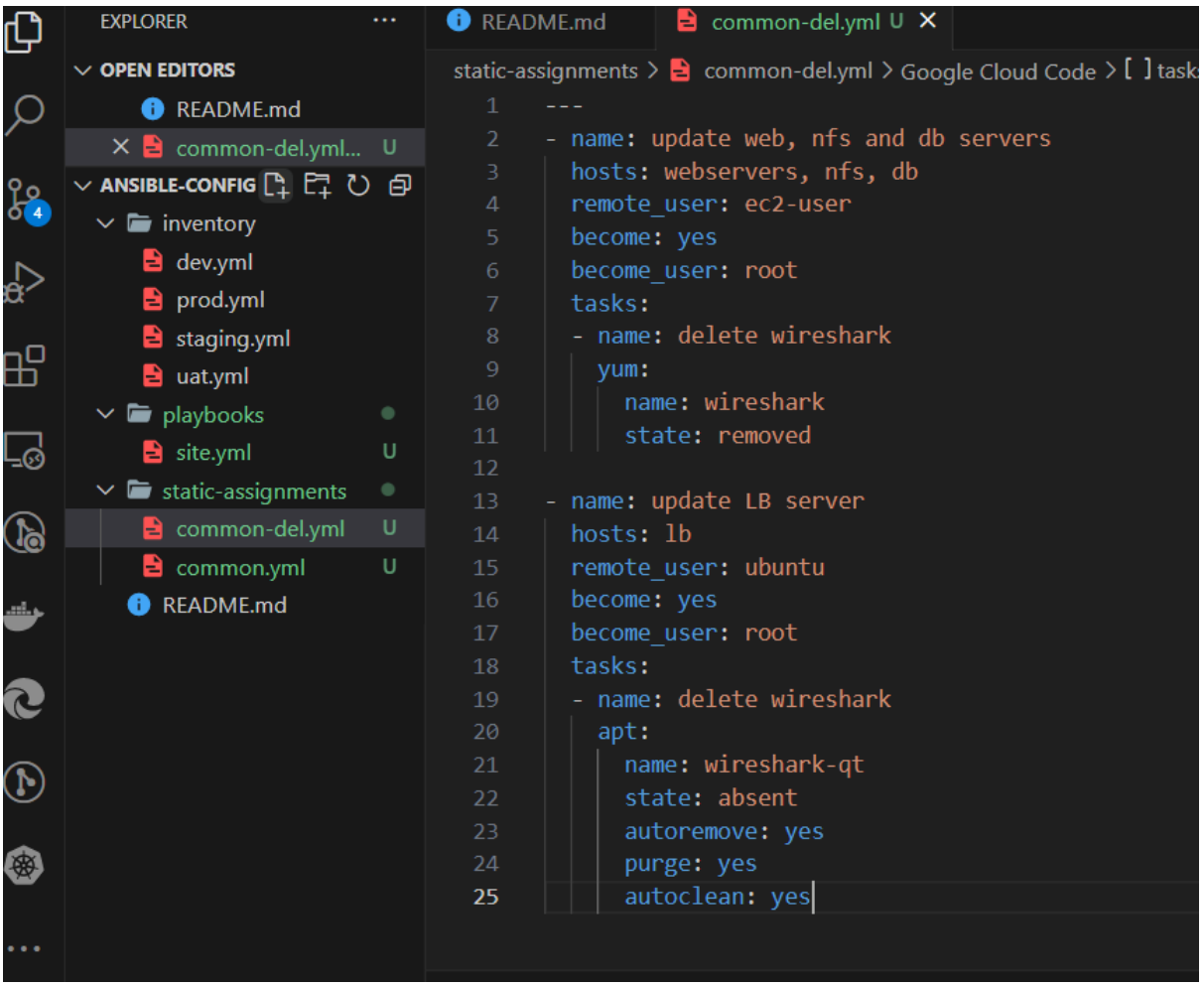
NB: The code above uses built in import_playbook Ansible module.

Below is the folder structure;



6. Run ansible-playbook command against the dev environment
7. Create another playbook under static-assignments and name it common-del.yml. In this playbook, configure deletion of wireshark utility.

```yaml
---
- name: update web, nfs and db servers
  hosts: webservers, nfs, db
  remote_user: ec2-user
  become: yes
  become_user: root
  tasks:
  - name: delete wireshark
    yum:
      name: wireshark
      state: removed

- name: update LB server
  hosts: lb
  remote_user: ubuntu
  become: yes
  become_user: root
  tasks:
  - name: delete wireshark
    apt:
      name: wireshark-qt
      state: absent
      autoremove: yes
      purge: yes
      autoclean: yes
```



8. We update site.yml with - import_playbook: ../static-assignments/common-del.yml instead of common.yml and run it against dev servers

Cd inventory

```
ubuntu@ip-172-31-24-30:~/ansible-config-artifact$ cd inventory
ubuntu@ip-172-31-24-30:~/ansible-config-artifact/inventory$ pwd
/home/ubuntu/ansible-config-artifact/inventory
ubuntu@ip-172-31-24-30:~/ansible-config-artifact/inventory$
```

sudo vi /etc/ansible/ansible.cfg

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

# config file for ansible -- https://ansible.com/
# ===============================================

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

#inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
#sudo_user      = root
#ask_sudo_pass = True
#ask_pass      = True
#transport      = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False

# plays will gather facts by default, which contain information about
# the remote system.
#
# smart - gather by default, but don't regather if already gathered
# implicit - gather by default, turn off with gather_facts: False
# explicit - do not gather by default, must say gather_facts: True
#gathering = implicit

# This only affects the gathering done by a play's gather_facts directiv
```

Edit the inventory in the above file and past;

/home/ubuntu/ansible-config-artifact/inventory

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
# config file for ansible -- https://ansible.com/
# ===============================================

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

inventory      = /home/ubuntu/ansible-config-artifact/inventory
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
#sudo_user      = root
#ask_sudo_pass = True
#ask_pass      = True
#transport      = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False

# plays will gather facts by default, which contain information abo
# the remote system.
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

 for more information. This feature will be removed in version 2.12. Deprecation
warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
172.31.44.66 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[DEPRECATION WARNING]: Distribution rhel 9.2 on host 172.31.41.162 should use
/usr/libexec/platform-python, but is using /usr/bin/python for backward
compatibility with prior Ansible releases. A future Ansible release will default to
using the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html
 for more information. This feature will be removed in version 2.12. Deprecation
warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
172.31.41.162 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[DEPRECATION WARNING]: Distribution rhel 9.2 on host 172.31.32.161 should use
/usr/libexec/platform-python, but is using /usr/bin/python for backward
compatibility with prior Ansible releases. A future Ansible release will default to
using the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html
 for more information. This feature will be removed in version 2.12. Deprecation
warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
172.31.32.161 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-24-30:~/ansible-config-artifact/inventory$
```

The hosts are all reachable vis ssh from ansible server.

```
Run this command;
ansible-playbook -i inventory/dev.yml playbooks/site.yaml
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

ok: [172.31.42.201]
mCYpy4mNoD99V67pZN/IdPytNyUXKWr9lC3LwOvz0Oj+tIguJJWh3bOXjCbB3/fJ
ok: [172.31.32.161]
ok: [172.31.44.66]

TASK [delete wireshark] ***********************************************************changed: [172.31.41.162]
changed: [172.31.42.201]
changed: [172.31.44.66]
changed: [172.31.32.161]

PLAY [update LB server] ***********************************************************
TASK [Gathering Facts] ************************************************************ok: [172.31.41.59]

TASK [delete wireshark] ***********************************************************changed: [172.31.41.59]

PLAY RECAP ************************************************************************172.31.32.161              : ok=3    chang
ed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.41.162              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.41.59               : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.42.201              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.44.66               : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
[ec2-user@ip-172-31-41-162 ~]$ wireshark --version
-bash: wireshark: command not found
[ec2-user@ip-172-31-41-162 ~]$
```

This just confirmed that the wireshark has been deleted accordingly.

Step3. **Configure UAT Webservers with a role 'Webserver'**

1. Launch 2 fresh EC2 instances using RHEL 8 image, we will use them as our uat servers, so give them names accordingly – Web1-UAT and Web2-UAT.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
jenkins	i-03e1bd9c8c491ae22	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-
Web1-UAT	i-00928c187be0e83b6	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-
Web2-UAT	i-0d2245f32981758eb	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-

2. Create a role using an Ansible utility called ansible-galaxy inside ansible-config-mgt/roles directory (you need to create roles directory upfront)

mkdir roles
cd roles
ansible-galaxy init webserver

3. removing unnecessary directories and files, the roles structure should look

   like this

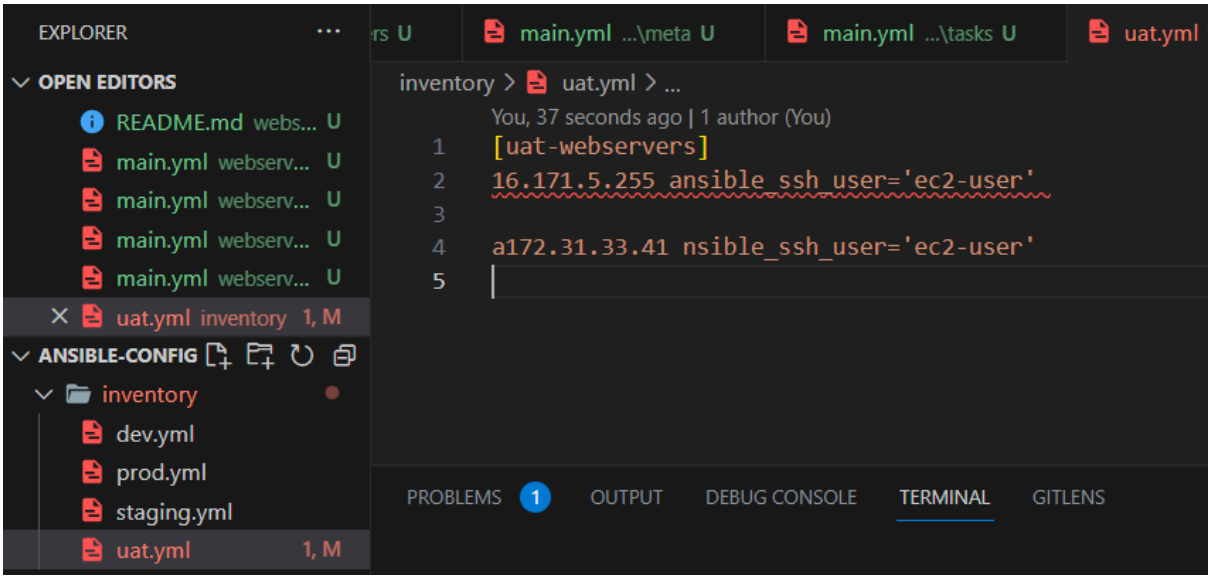
```
└── webserver
    ├── README.md
```

```
├── defaults
│   └── main.yml
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
└── templates
```

4. Update your inventory ansible-config-mgt/inventory/uat.yml file with IP addresses of your 2 UAT Web servers

[uat-webservers]
<Web1-UAT-Server-Private-IP-Address> ansible_ssh_user='ec2-user'

<Web2-UAT-Server-Private-IP-Address> ansible_ssh_user='ec2-user'



5. In /etc/ansible/ansible.cfg file uncomment roles_path string and provide a full path to your roles directory `roles_path = /home/ubuntu/ansible-config-mgt/roles`, so Ansible could know where to find configured roles.

```
# namespace. This setting maintains the behaviour which was the default prior
# to 2.5, duplicating these variables into the main namespace, each with a
# prefix of 'ansible_'.
# This variable is set to True by default for backwards compatibility. It
# will be changed to a default of 'False' in a future release.
# ansible_facts.
# inject_facts_as_vars = True

# additional paths to search for roles in, colon separated
roles_path    = /home/ubuntu/ansible-config-artifact/roles

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

# change the default callback, you can only have one 'stdout' type  enabled at a tim
e.
#stdout_callback = skippy


## Ansible ships with some plugins that require whitelisting,
## this is done to avoid running all of a type by default.
## These setting lists those that you want enabled for your system.
## Custom plugins should not need this unless plugin author specifies it.

# enable callback plugins, they can output to stdout but cannot be 'stdout' type.
#callback_whitelist = timer, mail
"/etc/ansible/ansible.cfg" 490L, 20037C                        68,58         11%
```

6. Install and configure Apache (httpd service)
7. Clone Tooling website from GitHub https://github.com/<your-name>/tooling.git.
8. Ensure the tooling website code is deployed to /var/www/html on each of 2 UAT Web servers.
9. Make sure httpd service is started

Add the following to the main.yml of the webserver role;

```
---
- name: install apache
  become: true
  ansible.builtin.yum:
    name: "httpd"
    state: present

- name: install git
  become: true
  ansible.builtin.yum:
    name: "git"
    state: present

- name: clone a repo
```

```
    become: true
    ansible.builtin.git:
      repo: https://github.com/<your-name>/tooling.git
      dest: /var/www/html
      force: yes

  - name: copy html content to one level up
    become: true
    command: cp -r /var/www/html/html/ /var/www/

  - name: Start service httpd, if not started
    become: true
    ansible.builtin.service:
      name: httpd
      state: started

  - name: recursively remove /var/www/html/html/ directory
    become: true
    ansible.builtin.file:
      path: /var/www/html/html
      state: absent
```

Reference ‘Webserver’ role

Step4. In the static-assignments folder, we create a new assignment for uat-webservers `uat-webservers.yml`

Then we reference the role

```
---
- hosts: uat-webservers
  roles:
     - webserver
```

Since the entry point to our ansible configuration is the `site.yml` file. Therefore, you need to refer your `uat-webservers.yml` role inside `site.yml`.

So, we should have this in site.yml

```
---
- hosts: all
- import_playbook: ../static-assignments/common.yml

- hosts: uat-webservers
- import_playbook: ../static-assignments/uat-webservers.yml
```

Step5. Commit your changes, create a Pull Request and merge them to main branch, make sure webhook triggered two consequent Jenkins jobs, they ran

successfully and copied all the files to your Jenkins-Ansible server into `/home/ubuntu/ansible-config-artifact/` directory.

Now run the playbook against your uat inventory and see what happens:

ansible-playbook -i /inventory/uat.yml /playbooks/site.yml

```
PLAY [uat-webservers] ***********************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************
ok: [172.31.19.230]
ok: [172.31.25.233]

TASK [webserver : install apache] ***********************************************************************************************************
ok: [172.31.19.230]
ok: [172.31.25.233]

TASK [webserver : install git] **************************************************************************************************************
ok: [172.31.19.230]
ok: [172.31.25.233]

TASK [webserver : clone a repo] *************************************************************************************************************
changed: [172.31.25.233]
changed: [172.31.19.230]

TASK [webserver : copy html content to one level up] ****************************************************************************************
changed: [172.31.25.233]
changed: [172.31.19.230]

TASK [webserver : Start service httpd, if not started] **************************************************************************************
ok: [172.31.25.233]
ok: [172.31.19.230]

TASK [webserver : recursively remove /var/www/html/html/ directory] *************************************************************************
changed: [172.31.19.230]
changed: [172.31.25.233]

PLAY RECAP **********************************************************************************************************************************
172.31.19.230              : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.25.233              : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-24-30:~/ansible-config-artifact$
```

Test the webserver configurations on the browser