

## **ANSIBLE REFACTORING, ASSIGNMENT, AND IMPORTS**

In this project, we would be working with the ansible-config-mgt repository and making some refactoring on our existing code, creating assignments and also implement the usage of the import functionality. The main aim of this is to make our playbook efficient re-usable, concise and very organise while carry out their tasks.

Code refactoring is often used in the software development because as we constantly change the behaviour of an application by adding new features, we expect to be changing source codes while expecting a particular behaviour. Refactoring enhances code readability, maintainability, reducing complexity for future adjustments without affecting the logic of the application .

Pre-requisite for the projects is the following.

- 1) Fundamental Knowledge of Installing and downloading software
- 2) Basic Understanding of Linux Commands
- 3) AWS account login with EC2 instances
- 4) 2 UAT Webserver Red Hat Enterprise Linux 9
- 5) 3 Webservers Ubuntu
- 6) Database Server: On Ubuntu 22.04+ MySQL
- 7) Playbooks
- 8) NFS Storage Server
- 9) Load Balancer Server
- 10) Ansible Jenkins Server
- 11) Jenkins
- 12) Python Language
- 13) Code Repository
- 14) Internet connection
- 15) Visual Studio Code

**IMPLEMENTATION STEPS:** Set up of all EC-2 instances.

- i) Ensure you login with your details to your AWS console via the <https://aws.amazon.com>

ii) Click on the EC2 link and start all 7 EC2 instances and create 2 UAT server instance making sure they are set up with the operating systems below.

i) 1 NFS server (Red hat) ii) 3 webservers. iii) 1 Ansible-Jenkins Server

iv) 1 Load Balance Server v) 1 Database Server vi) 2 new UAT servers as shown below.

We created our 2 UAT servers as seen below.

Type in the name and additional tag to the project (UAT - SERVERS). Select ubuntu from the quick start option .Also note that the Amazon machine image selection varies from user to user

Select RHEL 9 (Free Tier )

Name

UAT-SERVERS

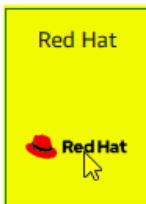
Add additional tags

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

 *Search our full catalog including 1000s of application and OS images*

**Quick Start**



Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Red Hat Enterprise Linux 9 (HVM), SSD Volume Type

Free tier eligible

ami-026ebd4fce2c043b2 (64-bit (x86)) / ami-03d6a5256a46c9feb (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Provided by Red Hat, Inc.

The instance type selected in the configuration is the t2 micro - free tier.

Click on the “Create new key pair” link and Ensure the Checkbox remains unchanged on the “Create security group”.

### Key pair name

Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

### Key pair type

 RSA

RSA encrypted private and public key pair

 ED25519

ED25519 encrypted private and public key pair

### Private key file format

 .pem

For use with OpenSSH

 .ppk

For use with PuTTY

**⚠** When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

[Cancel](#)[Create key pair](#)

### ▼ Network settings [Info](#)

[Edit](#)

#### Network [Info](#)

vpc-0c3c371436c0dcd9d

#### Subnet [Info](#)

No preference (Default subnet in any availability zone)

#### Auto-assign public IP [Info](#)

Enable

#### Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

 Create security group Select existing security group

We'll create a new security group called '**launch-wizard-46**' with the following rules:

Allow SSH traffic from  
Helps you connect to your instance

Anywhere  
0.0.0.0/0

Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

Then we proceed to launch our instance finally.

Number of instances [Info](#)

 ▲ ▼

When launching more than 1 instance, consider [EC2 Auto Scaling](#).

[Software Image \(AMI\)](#)

Provided by Red Hat, Inc.  
ami-026ebd4cfe2c043b2

[Virtual server type \(instance type\)](#)

t2.micro

[Firewall \(security group\)](#)

New security group

[Storage \(volumes\)](#)

1 volume(s) - 10 GiB

**i Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

**Launch instance**

Instance successfully launched and click to view Instance details with the IP address.

 <b>Success</b> Successfully initiated launch of instances ( <a href="#">i-0ebecbc92579825fe</a> , <a href="#">i-081c9dbd78b2632c1</a> )																						
<b>▶ Launch log</b>																						
<table border="1"><thead><tr><th>Web1-UAT</th><th>i-0ebecbc92579825fe</th><th>Running</th><th>t2.micro</th><th>Initializing</th><th>No alarms</th><th>us-east-1a</th><th>ec2-44-202-17-208.co...</th><th>44.202.17.208</th><th>-</th><th></th></tr></thead><tbody><tr><td>Web2-UAT</td><td><a href="#">i-081c9dbd78b2632c1</a></td><td>Running</td><td>t2.micro</td><td>Initializing</td><td>No alarms</td><td>us-east-1a</td><td>ec2-18-233-100-233.co...</td><td>18.233.100.233</td><td>-</td><td></td></tr></tbody></table>	Web1-UAT	i-0ebecbc92579825fe	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-44-202-17-208.co...	44.202.17.208	-		Web2-UAT	<a href="#">i-081c9dbd78b2632c1</a>	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-18-233-100-233.co...	18.233.100.233	-	
Web1-UAT	i-0ebecbc92579825fe	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-44-202-17-208.co...	44.202.17.208	-													
Web2-UAT	<a href="#">i-081c9dbd78b2632c1</a>	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-18-233-100-233.co...	18.233.100.233	-													

We then proceed to enhance our Jenkins job by creating a new directory called ansible-config-artifact as seen below

```
ubuntu@Jenkins-Ansible:~$ ls
ansible-config-artifact  ansible-config-mgt  inventory
ubuntu@Jenkins-Ansible:~$ cd ansible-config-artifact/
```

Once directory is changed, we would need permissions so that Jenkins can save files in the new directory.

```
ubuntu@Jenkins-Ansible:~/ansible-config-artifact$ sudo chmod -R 0777 /home/ubuntu/ansible-config-artifact
ubuntu@Jenkins-Ansible:~/ansible-config-artifact$ ls
```

Once that is done, we would go to the Jenkins console and download a copy artifact plug-in and check the installed plug-in section to confirm it was downloaded successfully.



A new freestyle project was created named `save_artifacts` and this project would be triggered by the completion of our existing ansible new project. We choose the Log rotation strategy and make sure the maximum of build be 3 as shown below

The screenshot shows the Jenkins project configuration for 'save\_artifacts'. At the top, there's a checkbox for 'Discard old builds' which is checked. Below it, the 'Strategy' section is set to 'Log Rotation'. Under 'Days to keep builds', the value is set to 3. At the bottom, there's a field labeled 'Max # of builds to keep' with a note 'if not empty, only up to this number of b'. A cursor icon is pointing at the '3' in this field.

Project to watch out for is the ansible new.

Build after other projects are built [?](#)

Projects to watch

ansible new,

Trigger only if build is stable

Trigger even if the build is unstable

Target directory [?](#)

/home/ubuntu/ansible-config-artifact

Parameter filters [?](#)

Flatten directories  Optional  Fingerprint Artifacts  Include Build

Advanced ▾

Add build step ▾

## Post-build Actions

Add post-build action ▾

Save

Apply

The save artifact project would be saving the artifacts in the ansible-config-artifact directory as seen above. We can test out set up by making some change in the READ.md file.

The screenshot shows a GitHub commit dialog for the 'ansible-config-mgt' repository. The 'main' branch is selected. The changes section contains three lines of text:

```
1 Our new artifact has finally surfaced after a lot of configurations
2 We would make it happen finally
3
```

The 'Commit message' field contains 'Updated README.md'. The 'Extended description' field is empty. Below the fields are two radio button options:

- Commit directly to the main branch
- Create a new branch for this commit and start a pull request  
[Learn more about pull requests](#)

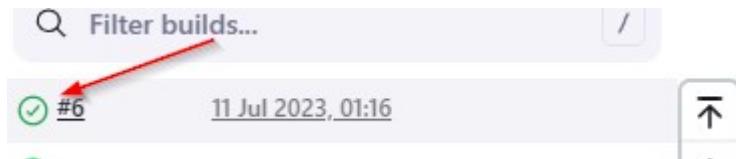
At the bottom are 'Cancel' and 'Commit changes' buttons, with 'Commit changes' being highlighted.

Once done we expect an automatic trigger in our Jenkins pipeline

The screenshot shows the Jenkins queue. Build #6 is pending, indicated by a red 'X' icon. Previous builds are listed below:

#	Status	Timestamp
6	(pending—In the quiet period. Expires in 3.3 sec)	
5	<a href="#">11 Jul 2023, 01:04</a>	
4	<a href="#">6 Jul 2023, 20:09</a>	

To the right of the queue are up and down arrow icons for navigating the list.



We can now check the console output as well as checking the Jenkins-Ansible server

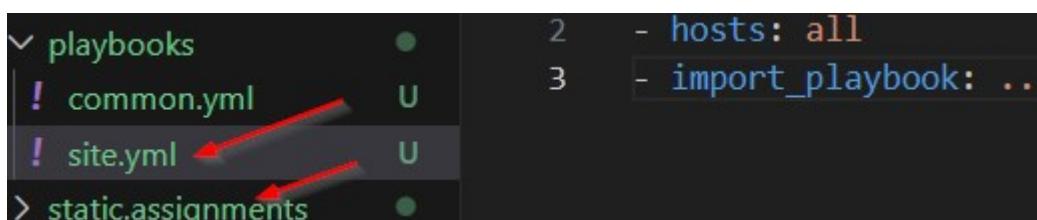
The screenshot shows the Jenkins "Console Output" page for build #6. The log output is as follows:

```
Started by upstream project "ansible new" build number 18
originally caused by:
Started by GitHub push by eyewande2022
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/save artifacts2
ubuntu@Jenkins-Ansible:~/ansible-config-artifact$ cat READ.md
Our new artifact has finally surfaced after alot of configurations
We would make it happen finally .
ubuntu@Jenkins-Ansible:~/ansible-config-artifact$
```

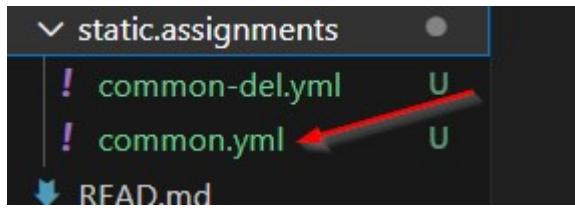
Once this is done, we can now proceed to refactor ansible code by importing other playbooks into the site.yml file

. It is important for us to pull down all latest code from our (main branch) before creating a new branch called refactor.

Once that is done, we can proceed to import other playbooks in our site.yml file which is the entry point into the entire infrastructure configuration. The site.yml file becomes the parent all other playbooks developed.



We also have to create a new folder called static assignments where all other children playbooks will be stored. Hence we would be moving the common.yml file we used previously into the static-assignments folder.



As seen below we imported the common-del.yml file to the site.yml file.

After all necessary folders has been created, we have a folder structure as shown below.

```
inventory > uat.yml
1
inventory
  dev.yml
  prod.yml
  staging.yml
  uat.yml
> playbooks
> static.assignments
webserver
  defaults
    main.yml
  handlers
    main.yml
  meta
    main.yml
  tasks
    main.yml
  templates
    READ.md
    READ.md
```

PROBLEMS DEBUG CO

```
Git
2023-07-12 09:01:
[info] > git show
--textconv :inven
uat.yml [7ms]
2023-07-12 09:01:
[info] > git ls-f
```

Ensure to run the ansible-playbook command against the dev environment as shown below

```
● ubuntu@ip-172-31-16-193:~/ansible-config-artifact/inventory$ sudo vi dev.yml
○ ubuntu@ip-172-31-16-193:~/ansible-config-artifact/inventory$ ansible all -m ping -i dev.yml
The authenticity of host '172.31.86.21 (172.31.86.21)' can't be established.
ECDSA key fingerprint is SHA256:+s8vz+3kk0FCTNDNN+0a3cP0fIDRCgl2scgi4UkM/8.
The authenticity of host '172.31.86.172 (172.31.86.172)' can't be established.
ECDSA key fingerprint is SHA256:P+yQm6qwfz3q3FEcbXPfLmB0s0m0cRpg4tse5rGh9P0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? [DEPRECATION WARNING]: D:
use /usr/libexec/platform-python, but is using /usr/bin/python for
```

We can see that the private Ip of all servers indicate that they are a success. Please note that wireshark was successful installed on all servers in the previous project

```
172.31.85.81 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
172.31.83.227 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
es172.31.81.224 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.86.21 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Our new task is for us to create another playbook under static-assignment and name it common-del.yml. The function of the

playbook is to delete the wireshark utility on all servers . We are now required to update the site.yml file as shown below

```

EXPLORER      ...
ANSIBLE-CONFIG-ARTIFACT [SSH]
ansible-config-artifact ! playbooks > site.yml
1  ---
2  - hosts: all
3  - import_playbook: /home/ubuntu/ansible-config-artifact/static.assignments/common-del.yml
4
5

```

We are now supposed to run the dev.yml file against the site.yml file as show below

```

ubuntu@ip-172-31-16-193:~/ansible-config-artifact$ ansible-playbook -i inventory/dev.yml /home/ks/site.yml

PLAY [all] ****
TASK [Gathering Facts] ****
The authenticity of host '172.31.86.172 (172.31.86.172)' can't be established.
ECDSA key fingerprint is SHA256:P+yQm6qwfz3qFEcbXPfLmB0s0m0cRpg4tse5rGh9P0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? ok: [172.31.86.21]
ok: [172.31.83.227]
[DEPRECATION WARNING]: Distribution rhel 9.2 on host 172.31.85.81 should use /usr/libexec/platform-python backward compatibility with prior Ansible releases. A future Ansible release will default to using the distribution's python. 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 a playbook or --deprecate-warnings=False on the command line.
ok: [172.31.85.81]
ok: [172.31.81.224]
yesok: [172.31.80.120]

ok: [172.31.86.172]

PLAY [update nfs server] ****
TASK [Gathering Facts] ****
ok: [172.31.85.81]

TASK [delete wireshark] ****
TASK [Gathering Facts] ****
ok: [172.31.83.227]
ok: [172.31.86.21]
ok: [172.31.86.172]
ok: [172.31.81.224]
ok: [172.31.85.81]
ok: [172.31.80.120]

TASK [delete wireshark] ****
ok: [172.31.86.21]
ok: [172.31.86.172]
[WARNING]: Updating cache and auto-installing missing dependency: python3-apt
fatal: [172.31.85.81]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory: b'apt-get'", "rc": 2}
changed: [172.31.83.227]
changed: [172.31.80.120]
changed: [172.31.81.224]

PLAY RECAP ****
172.31.80.120 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.81.224 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.83.227 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.85.81 : ok=4    changed=1    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
172.31.86.172 : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.86.21  : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

As seen from above the task states that they were deleted but we must confirm this individually.

```
ubuntu@Database-Server:~$ wireshark --version
Command 'wireshark' not found, but can be installed with:
sudo apt install wireshark-qt
```

```
[ec2-user@NFS-SERVER ~]$ which wireshark
/usr/bin/which: no wireshark in (/home/ec2-user/.local/bin:/home/ec2-user/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin)
[ec2-user@NFS-SERVER ~]$
```

```
ubuntu@target-server-A:~$ wireshark -version
Command 'wireshark' not found, but can be installed with:
sudo apt install wireshark-qt
ubuntu@target-server-A:~$
```

```
ubuntu@ip-172-31-86-21:~$ sudo hostname Target-server-B
ubuntu@ip-172-31-86-21:~$ bash
ubuntu@Target-server-B:~$ wireshark --version
Command 'wireshark' not found, but can be installed with:
sudo apt install wireshark-qt
```

```
ubuntu@target-server-c:~$ wireshark --version
Command 'wireshark' not found, but can be installed with:
sudo apt install wireshark-qt
ubuntu@target-server-c:~$
```

```
ubuntu@nginx-lb:~$ wireshark --version
Command 'wireshark' not found, but can be installed with:
sudo apt install wireshark-qt
```

All has been deleted successfully and we have simple learn how to use import-playbooks module and we have a ready solution to install and delete packages on multiple servers by just running one command.

Please note it is encouraged to turn off all other servers that we aren't using now and only focus on the 2 webservers as the focus.

## CONFIGURATION OF 2 UAT WEBSEVERS WITH A ROLE WEB SERVER

Having created our UAT web servers earlier we would be creating role.

And a new directory in the /etc/ansible directory as shown below.

```
ubuntu@ip-172-31-16-193:~$ sudo hostname Jenkins-Ansible
ubuntu@ip-172-31-16-193:~$ bash
ubuntu@Jenkins-Ansible:~$ cd ansible-config-mgt/
ubuntu@Jenkins-Ansible:~/ansible-config-mgt$ ls
README.md  inventory  playbooks
ubuntu@Jenkins-Ansible:~/ansible-config-mgt$ mkdir roles
ubuntu@Jenkins-Ansible:~/ansible-config-mgt$ ls
README.md  inventory  playbooks  roles
ubuntu@Jenkins-Ansible:~/ansible-config-mgt$ cd roles
```

Next would be to initialize our ansible utility called ansible galaxy inside the ansible-config-mgt/roles directory and once completed we are expected to install tree to see the structure of all the files created.

```
ubuntu@Jenkins-Ansible:~/ansible-config-mgt/roles$ ansible-galaxy init webserver
- Role webserver was created successfully
ubuntu@Jenkins-Ansible:~/ansible-config-mgt/roles$ ls
webserver
ubuntu@Jenkins-Ansible:~/ansible-config-mgt/roles$ sudo apt install tree -y
```

Here are the structure of the webserver and after some modification.

```
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@Jenkins-Ansible:~/ansible-config-mgt/roles$ tree webserver/
webserver/
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```



Remove unnecessary folders are as show below

```
ubuntu@Jenkins-Ansible:~/ansible-config-mgt/roles$ cd webserver/
ubuntu@Jenkins-Ansible:~/ansible-config-mgt/roles/webserver$ ll
total 48
drwxrwxr-x 10 ubuntu ubuntu 4096 Jul  4 21:49 .
drwxrwxr-x  3 ubuntu ubuntu 4096 Jul  4 21:49 ../
-rw-rw-r--  1 ubuntu ubuntu  539 Jul  4 21:49 .travis.yml
-rw-rw-r--  1 ubuntu ubuntu 1328 Jul  4 21:49 README.md
drwxrwxr-x  2 ubuntu ubuntu 4096 Jul  4 21:49 defaults/
drwxrwxr-x  2 ubuntu ubuntu 4096 Jul  4 21:49 files/
drwxrwxr-x  2 ubuntu ubuntu 4096 Jul  4 21:49 handlers/
drwxrwxr-x  2 ubuntu ubuntu 4096 Jul  4 21:49 meta/
drwxrwxr-x  2 ubuntu ubuntu 4096 Jul  4 21:49 tasks/
drwxrwxr-x  2 ubuntu ubuntu 4096 Jul  4 21:49 templates/
drwxrwxr-x  2 ubuntu ubuntu 4096 Jul  4 21:49 tests/
drwxrwxr-x  2 ubuntu ubuntu 4096 Jul  4 21:49 vars/
```

Next step would be to update the UAT.yml file in the inventory directory with the private IP addresses of the 2 UAT webservers.

The screenshot shows two EC2 instances in the AWS CloudWatch Metrics interface. Both instances are running (t2.micro) and have passed 2/2 checks. They are located in us-east-1a. The public IPv4 address for Web1-UAT is 44.202.17.208, and for Web2-UAT it is 18.233.100.233. The private IPv4 address for Web1-UAT is 172.31.94.247, and for Web2-UAT it is 172.31.91.35. These private IP addresses are circled in red.

**Instance: i-0ebecbc92579825fe (Web1-UAT)**

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
<b>Instance summary</b> <a href="#">Info</a>						
Instance ID i-0ebecbc92579825fe (Web1-UAT)	Public IPv4 address 44.202.17.208   <a href="#">open address</a>	Private IPv4 addresses 172.31.94.247				
IPv6 address	Instance state	Public IPv4 DNS				

**Instance: i-081c9dbd78b2632c1 (Web2-UAT)**

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
<b>Instance summary</b> <a href="#">Info</a>						
Instance ID i-081c9dbd78b2632c1 (Web2-UAT)	Public IPv4 address 18.233.100.233   <a href="#">open address</a>	Private IPv4 addresses 172.31.91.35				
IPv6 address	Instance state	Public IPv4 DNS				

**! uat.yml M**

```
! uat.yml
inventory > ! uat.yml
1 [uat-webservers]
2 172.31.94.247 ansible_ssh_user= 'ec2-user'
3 172.31.91.35 ansible_ssh_user= 'ec2-user'
```

```
ubuntu@Jenkins-Ansible:~/ansible-config-mgt/inventory$ cat uat.yml
[uat-webservers]
172.31.94.247 ansible_ssh_user='ec2-user'

172.31.91.35 ansible_ssh_user='ec2-user'
ubuntu@Jenkins-Ansible:~/ansible-config-mgt/inventory$ [
```

Once this is done, Open the terminal for the 2 UAT webservers and SSH into each public Ip address from the Jenkins-Ansible server as shown below

```
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-94-247 ~]$ sudo hostname WEB1-UAT
[ec2-user@ip-172-31-94-247 ~]$ bash
[ec2-user@WEB1-UAT ~]$ [
```

```
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-91-35 ~]$ sudo hostname WEB2-UAT
[ec2-user@ip-172-31-91-35 ~]$ bash
[ec2-user@WEB2-UAT ~]$ [
```

As you can see it is successful and we would be exiting out of them both as shown below

```
ubuntu@Jenkins-Ansible:~/.ssh$ ssh -A ec2-user@44.202.17.208
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Tue Jul  4 22:04:59 2023 from 81.152.239.86
[ec2-user@ip-172-31-94-247 ~]$ [
```

```
Last login: Tue Jul  4 22:04:59 2023 from 81.152.239.86
[ec2-user@ip-172-31-94-247 ~]$ exit
logout
Connection to 44.202.17.208 closed.
```

```

ubuntu@Jenkins-Ansible:~/ssh$ ssh -A ec2-user@18.233.100.233
The authenticity of host '18.233.100.233 (18.233.100.233)' can't be established.
ECDSA key fingerprint is SHA256:INZJMrH4ZBxaoZKsh+WQDV9ipDdFRZutjg434mm50.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.233.100.233' (ECDSA) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register [I]
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Tue Jul  4 22:06:54 2023 from 81.152.239.86
[ec2-user@ip-172-31-91-35 ~]$ exit
logout
Connection to 18.233.100.233 closed.

```

Once we successfully achieved this, we would navigate to the /etc/ansible/ansible.cfg file to uncomment roles path by providing the full path of our role directory as shown below.

```

u inventory/
ubuntu@ip-172-31-16-193:~/ansible-config-artifact/inventory$ sudo vi uat.yml
ubuntu@ip-172-31-16-193:~/ansible-config-artifact/inventory$ cd ..
ubuntu@ip-172-31-16-193:~/ansible-config-artifact$ sudo vi /etc/ansible/ansible.cfg

```

```

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

```

The next step would be to copy the yml file to the tasks folder with the code instructions on what to be carried out .There are 4 tasks we want to carry out in this single main.yml file .

- 1)Install APACHE HTTPD
- 2)Install git on each server Clone a tooling website from our GitHub
- 3) copy html content to the /var/www directory
- 4)Start our httpd service as shown below

```

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/eyewande2022/tooling.git
  dest: /var/www/html
  force: yes

name: copy html content to one level up
become: true
- INSERT --

```

Our tasks file has been successfully updated

```

ubuntu@ip-172-31-16-193:~/ansible-config-artifact$ cd webserver/
ubuntu@ip-172-31-16-193:~/ansible-config-artifact/webserver$ ls
READ.md  handlers  tasks
defaults  meta      templates
ubuntu@ip-172-31-16-193:~/ansible-config-artifact/webserver$ cd tasks/
ubuntu@ip-172-31-16-193:~/ansible-config-artifact/webserver/tasks$ sudo vi main.yml
ubuntu@ip-172-31-16-193:~/ansible-config-artifact/webserver/tasks$ 

```

The next step would be to create a reference “Webserver Role”

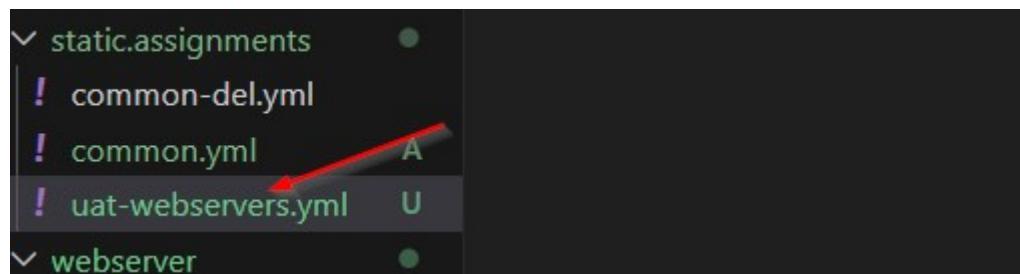
## REFERENCE WEBSERVER ROLE

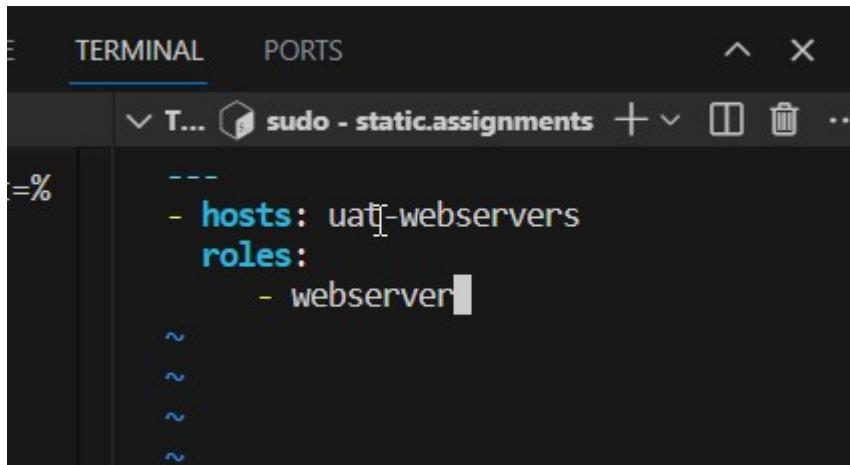
Within the static-assignments folder, we create a uat-webservers.yml file and reference the roles as shown below on the terminal as well as on the Visual Studio terminal

```

ubuntu@ip-172-31-16-193:~$ cd ansible-config-artifact/
ubuntu@ip-172-31-16-193:~/ansible-config-artifact$ cd static.assignments/
ubuntu@ip-172-31-16-193:~/ansible-config-artifact/static.assignments$ sudo vi uat-webservers.yml
ubuntu@ip-172-31-16-193:~/ansible-config-artifact/static.assignments$ 

```

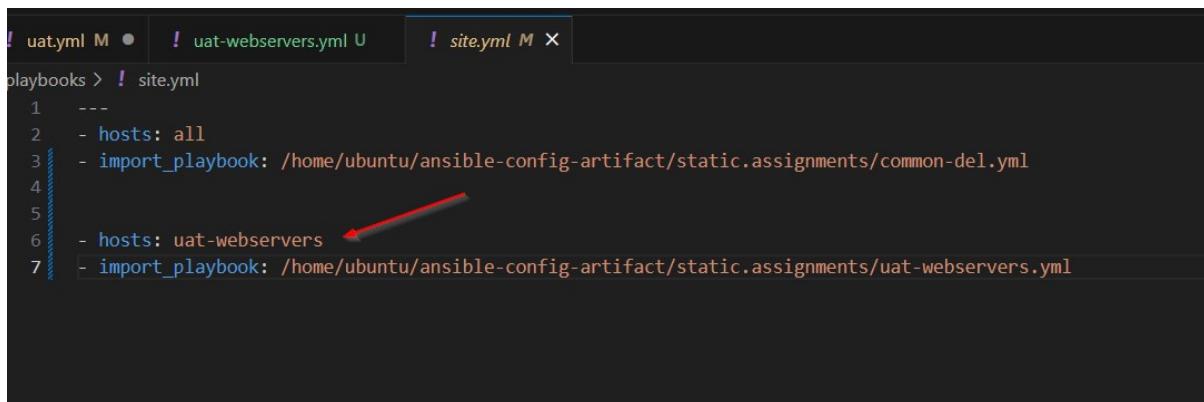




A screenshot of a terminal window titled "sudo - static.assignments". The content shows a YAML configuration snippet:

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

Since the entry point to our ansible configuration is the site.yml file .We would be referring our UAT webservers role inside the site.yml file as shown below



A screenshot of a code editor showing the site.yml file under playbooks. The file contains the following YAML code:

```
playbooks > site.yml  
1 ---  
2 - hosts: all  
3 - import_playbook: /home/ubuntu/ansible-config-artifact/static.assignments/common-del.yml  
4  
5 - hosts: uat-webservers  
6 - import_playbook: /home/ubuntu/ansible-config-artifact/static.assignments/uat-webservers.yml
```

A red arrow points to the line "- hosts: uat-webservers" at line 6.

Once updated, we would be committing our changes

Multiple files and  
folders has been added

✓ Commit ▾

✗ Staged Changes 6

- ! dev.yml inventory M
- ! prod.yml inventory A
- ! staging.yml inventory A
- ! uat.yml inventory M
- ! site.yml playbooks M
- ! common.yml static.as... A

✗ Changes 9

- ! dev.yml inventory M
- ! uat.yml inventory M
- ! site.yml playbooks M
- ! uat-webservers.yml s... U
- ⬇ READ.md webserver U
- ! main.yml webserver/d... U
- ! main.yml webserver/h... U
- ! main.yml webserver/... U
- ! main.yml webserver/t... U

PROBLEMS DEBUG CONSOLE

✗ OUTPUT

ansible-config-art  
playbooks/site.yml

2023-07-12 09:35:11

## Open and create a pull request

[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.

A screenshot of a GitHub pull request interface. At the top, there are buttons for 'base: main' and 'compare: refactor'. To the right of these, a green checkmark icon and the text 'Able to merge. These branches can be automatically merged.' are displayed. Below this, a large blue header bar contains the word 'Refactor' in white. On the left, there's a circular profile picture of a person wearing a green hat. At the bottom, there are two tabs: 'Write' (selected) and 'Preview'. To the right of the tabs are several icons: a bold 'B', an italic 'I', a code block 'code', a double arrow 'diff', a magnifying glass 'search', a list 'list', a user '@', a reply 'comment', a left arrow 'back', and a square 'close'. On the far right, the word 'Reviewers' is followed by 'No reviews'.

The screenshot shows the GitHub pull request interface for a branch named 'refactor' against the 'main' branch. At the top, it says 'Able to merge. These branches can be automatically merged.' Below the title 'Refactor', there are tabs for 'Write' and 'Preview'. A red arrow points to the 'Create pull request' button at the bottom right of the main area.

and merging to out (main branch).

The screenshot shows the GitHub branch protection rules for the 'refactor' branch. It includes sections for requiring approval from specific reviewers, setting up continuous integration, and confirming no conflicts with the base branch. A red box highlights the 'Merge pull request' button, which is also being clicked by a cursor. A yellow overlay covers the bottom part of the screenshot.

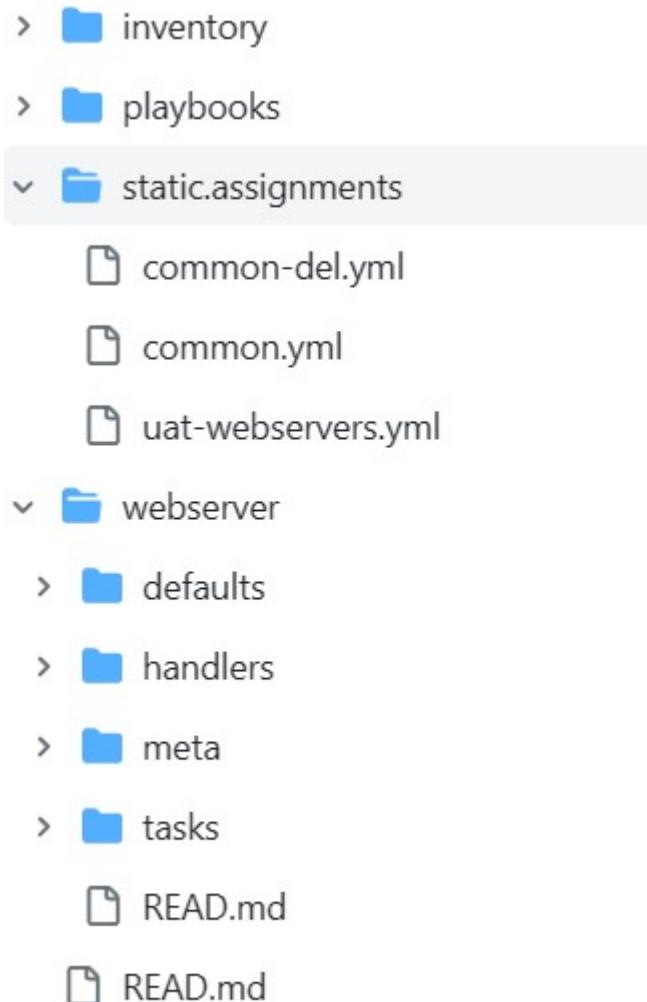
Successfully merged to main

## Refactor #1

Merged eyewande2022 merged 2 commits into main from refactor now

Conversation 0 Commits 2 Checks 0 Files changed 7

Checking the main file, we can see all the files there



Once done we would be running out ansible playbook against our UAT inventory.

```
* ubuntu@ip-172-31-16-193:~$ cd ansible-config-artifact/
* ubuntu@ip-172-31-16-193:~/ansible-config-artifact$ ansible-playbook -i /home/ubuntu/ansible-config-artifact/inventory/uat.yml      /home/ubuntu/
ansible-config-artifact/playbooks/site.yml
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS setting is set to allow bad characters in group names by default, this will change,
but still be user configurable on deprecation. This feature will be removed in version 2.10. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
[WARNING]: Invalid characters were found in group names but not replaced, use -vvv to see details
PLAY [all] ****
TASK [Gathering Facts] ****
[DEPRECATION WARNING]: Distribution rhel 9.2 on host 172.31.94.247 should use /usr/libexec/platform-python, but is using /usr/bin/python for
host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreters.html#backward-compatibility-with-prior-ansible-releases
[WARNING]: Distribution rhel 9.2 on host 172.31.91.35 should use /usr/libexec/platform-python, but is using /usr/bin/python for
host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreters.html#backward-compatibility-with-prior-ansible-releases
[DEPRECATION WARNING]: Distribution rhel 9.2 on host 172.31.91.35 should use /usr/libexec/platform-python, but is using /usr/bin/python for
host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreters.html#backward-compatibility-with-prior-ansible-releases
[WARNING]: Could not match supplied host pattern, ignoring: nfs
```

```
TASK [delete wireshark] ****
[WARNING]: Updating cache and auto-installing missing dependency: python3-apt
fatal: [172.31.94.247]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory: b'apt-get'", "rc": 2}
fatal: [172.31.91.35]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory: b'apt-get'", "rc": 2}

PLAY RECAP ****
172.31.91.35 : ok=2    changed=0   unreachable=0   failed=1    skipped=0   rescued=0   ignored=0
172.31.94.247 : ok=2    changed=0   unreachable=0   failed=1    skipped=0   rescued=0   ignored=0

ubuntu@ip-172-31-16-193:~/ansible-config-artifact$
```

Ln 1, Col 1 (101 selected) Spaces: 2 UTF-8 LF YAML ⌕

```
PLAY RECAP ****
172.31.91.35 : ok=2    changed=0   unreachable=0   failed=1    skipped=0   rescued=0   ignored=0
172.31.94.247 : ok=2    changed=0   unreachable=0   failed=1    skipped=0   rescued=0   ignored=0

o ubuntu@ip-172-31-16-193:~/ansible-config-artifact$
```

As shown above ,we can see that the 2 webservers , have been displayed and ran by the ansible playbook instructions and are well configured ..We can also notice some Ip address in red colour failing ,there were the other webservers that were turned off earlier having their own common-del.yml file (used to carry out the Wireshark deletion earlier )

Our full ansible architecture has now been refactored using our laptop with a functional internet and comprises of the list of components below.

1 Control Jenkins-ansible server

1 Database server

1 NFS server

1 Load balancer

2 UAT webserver

3 Webserver

We just deployed and configured a UAT Web Servers using import and roles .

