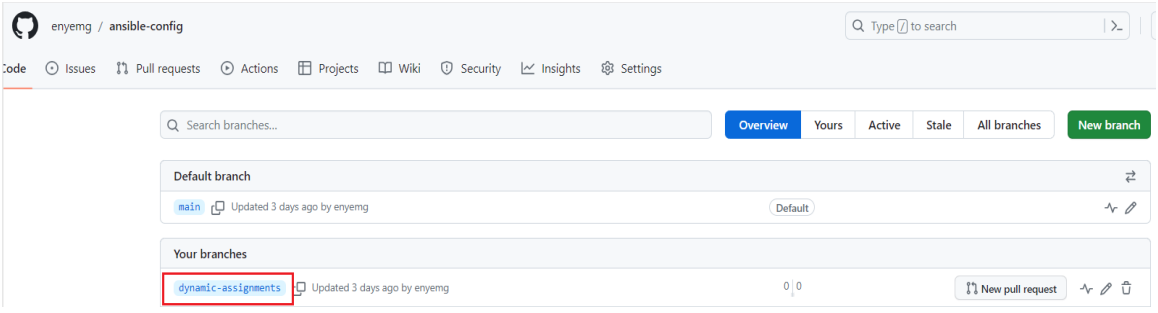
PROJECT 13: ANSIBLE DYNAMIC ASSIGNMENTS (INCLUDE) AND COMMUNITY ROLES

My experience with projects 11 and 12 has given me some understanding and expertise in Ansible, enabling me to conduct configurations using playbooks, roles, and imports. Now it is time to continue configuring your UAT servers by using the include module to introduce dynamic assignments. Static assignments use the import Ansible module while the module that enables dynamic assignments is include. However, **import = Static** and **include = Dynamic.** When the import module is used, all statements are pre-processed at the time playbooks are parsed. Meaning, when I execute the site.yml playbook, Ansible will process all the playbooks referenced during the time it is parsing the statements. This also means that, during actual execution, if any statement changes, such statements will not be considered. Hence, it is static.

On the other hand, when the include module is used, all statements are processed only during the execution of the playbook. Meaning, after the statements are parsed, any changes to the statements encountered during execution will be used. It is recommended to use static assignments for playbooks because it is more reliable. With dynamic ones, it is hard to debug playbook problems due to their dynamic nature. However, one can use dynamic assignments for environment-specific variables as we will be introducing in this project.

Step 1 - Introducing Dynamic Assignment into our Structure

1. In the https://github.com/<your-name>/ansible-config GitHub repository start a new branch and call it dynamic-assignments.



2. Create a new folder, and name it dynamic-assignments. Then inside this folder, create a new file and name it env-vars.yml. We will instruct site.yml to include this playbook later.

```
ubuntu@ip-172-31-40-224:~/ansible-config$ mkdir dynamic-assignments
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/dynamic-assignments$ touch env-vars.yml
ubuntu@ip-172-31-40-224:~/ansible-config/dynamic-assignments$ ls
env-vars.yml
```

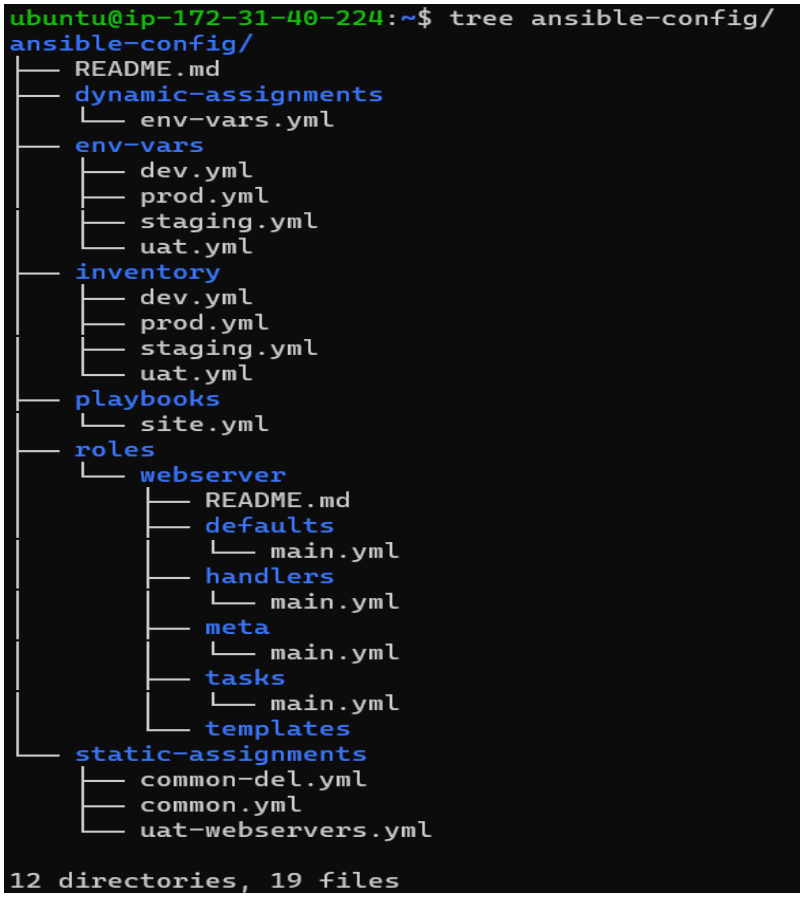
3. The GitHub shall have the following structure by now.

```
ubuntu@ip-172-31-40-224:~$ tree ansible-config/
ansible-config/
├── README.md
├── dynamic-assignments
│   └── env-vars.yml
├── inventory
│   ├── dev.yml
│   ├── prod.yml
│   ├── staging.yml
│   └── uat.yml
├── playbooks
│   └── site.yml
├── roles
│   └── webserver
│       ├── README.md
│       ├── defaults
│       │   └── main.yml
│       ├── handlers
│       │   └── main.yml
│       ├── meta
│       │   └── main.yml
│       ├── tasks
│       │   └── main.yml
│       └── templates
└── static-assignments
    ├── common-del.yml
    ├── common.yml
    └── uat-webservers.yml

11 directories, 15 files
```

4. Since we will be using the same Ansible to configure multiple environments, and each of these environments will have certain unique attributes, such as servername, IP-address etc., we will need a way to set values to variables per specific environment. Therefore, I created a new folder env-vars, and then for each environment, create new YAML files which we will use to set variables.

```
ubuntu@ip-172-31-40-224:~/ansible-config$ mkdir env-vars
ubuntu@ip-172-31-40-224:~/ansible-config$ cd env-vars
ubuntu@ip-172-31-40-224:~/ansible-config/env-vars$ touch dev.yml prod.yml staging.yml uat.yml
```

5. Your layout should now look like this.



6. Now paste the instruction below into the env-vars.yml file.

```
---
- name: collate variables from env specific file, if it exists
  hosts: all
  tasks:
    - name: looping through list of available files
      include_vars: "{{ item }}"
      with_first_found:
        - files:
            - dev.yml
            - staging.yml
            - prod.yml
            - uat.yml
```

```
      paths:
        - "{{ playbook_dir }}/../env-vars"
    tags:
      - always
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/dynamic-assignments$ sudo vi env-vars.yml
```

```
---
- name: collate variables from env specific file, if it exists
  hosts: all
  tasks:
    - name: looping through list of available files
      include_vars: "{{ item }}"
      with_first_found:
        - files:
            - dev.yml
            - staging.yml
            - prod.yml
            - uat.yml
          paths:
            - "{{ playbook_dir }}/../env-vars"
      tags:
        - always
```

Notice 3 things to notice here:

a. I used include_vars syntax instead of include, this is because Ansible developers decided to separate different features of the module. From Ansible version 2.8, the include module is deprecated and variants of include_* must be used. These are:

- include_role
- include_tasks
- include_vars

In the same version, variants of import were also introduced, such as:

- import_role
- import_tasks

b. I made use of special variables { playbook_dir } and { inventory_file }. { playbook_dir } will help Ansible to determine the location of the running playbook, and from there navigate to another path on the filesystem. { inventory_file } on the other hand will dynamically resolve to the name of the inventory file being used, then append .yml so

that it picks up the required file within the env-vars folder.

c. I am including the variables using a loop. with_first_found implies that looping through the list of files, the first one found is used. This is good so that we can always set default values in case an environment-specific env file does not exist.

Step 2 - Update site.yml with dynamic assignments

Update the site.yml file to make use of the dynamic assignment. The site.yml should now look like this.

```
---
- hosts: all
- name: Include dynamic variables
  tasks:
  import_playbook: ../static-assignments/common.yml
  include: ../dynamic-assignments/env-vars.yml
  tags:
    - always

- hosts: webservers
- name: Webserver assignment
  import_playbook: ../static-assignments/webservers.yml
```

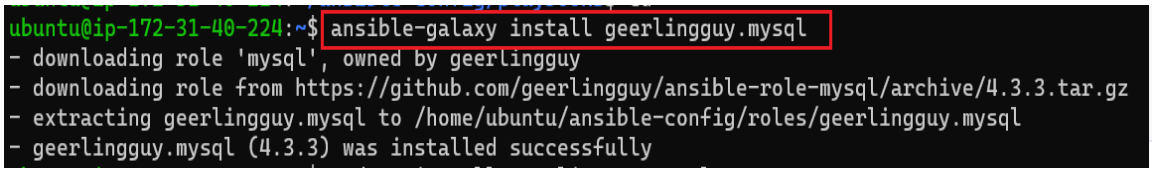
```
ubuntu@ip-172-31-40-224:~/ansible-config/playbooks$ sudo vi site.yml
```

```
---
- hosts: all
- name: Include dynamic variables
  tasks:
  import_playbook: ../static-assignments/common.yml
  include: ../dynamic-assignments/env-vars.yml
  tags:
    - always

- hosts: webservers
- name: Webserver assignment
  import_playbook: ../static-assignments/webservers.yml
```

Step 3 - Community Role

Create a role for MySQL database – it should install the MySQL package, create a database and configure users. There are roles that are actually production ready, and dynamic to accommodate most of Linux flavours. With Ansible Galaxy again, we can simply download a ready-to-use Ansible role, and keep going. I will be using a MySQL role developed by geerlingguy from the community roles

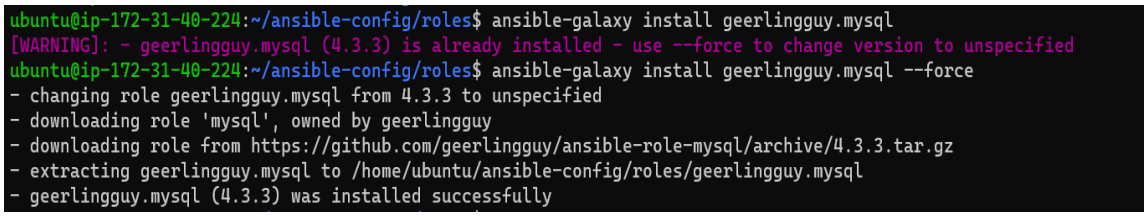
1. Download Mysql Ansible Role

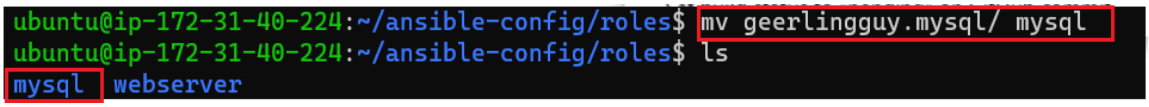


2. To preserve your GitHub in its actual state after you install a new role – make a commit and push to main your 'ansible-config' directory. On the Jenkins-Ansible server make sure that git is installed with git --version, then go to the ansible-config directory and run

   git init
   git pull https://github.com/<your-name>/ansible-config.git
   git remote add origin https://github.com/<your-name>/ansible-config.git
   git branch roles-feature
   git switch roles-feature

3. Inside the roles directory create your new MySQL role with **ansible-galaxy install geerlingguy.mysql** and rename the folder to MySQL





4. Read README.md file, and edit the roles configuration to use the correct credentials for MySQL required for the tooling website.

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/mysql$ ls
LICENSE  README.md  defaults  handlers  meta  molecule  tasks  templates  vars
ubuntu@ip-172-31-40-224:~/ansible-config/roles/mysql$ sudo vi README.md
```

```
## Role Variables

Available variables are listed below, along with default values (see `defaults/main.yml`):
```

5. To set the role variables, go to defaults/main.yml and uncomment the database and users field to have the values on the right.

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/mysql/defaults$ sudo vi main.yml
```

```
# Databases.
mysql_databases: []
#   - name: example
#     collation: utf8_general_ci
#     encoding: utf8
#     replicate: 1

# Users.
mysql_users: []
#   - name: example
#     host: 127.0.0.1
#     password: secret
#     priv: *.*:USAGE
```

```
# Databases.
mysql_databases: []
  - name: tooling
    collation: utf8_general_ci
    encoding: utf8
    replicate: 1

# Users.
mysql_users: []
  - name: webaccess
    host: 0.0.0.0
    password: secret
    priv: `*.*:ALL,GRANT`
```

6. Now it is time to upload the changes into your GitHub. I also created a Pull Request and merged it to the main branch on GitHub.

`git add .`

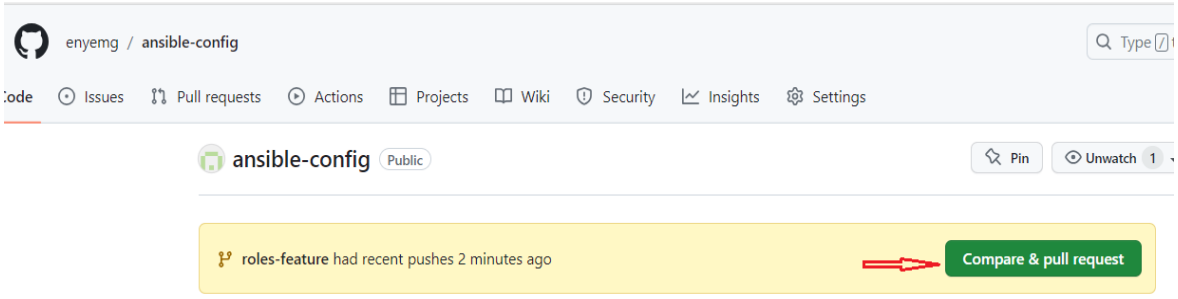
`git commit -m "Commit new role files into GitHub"`

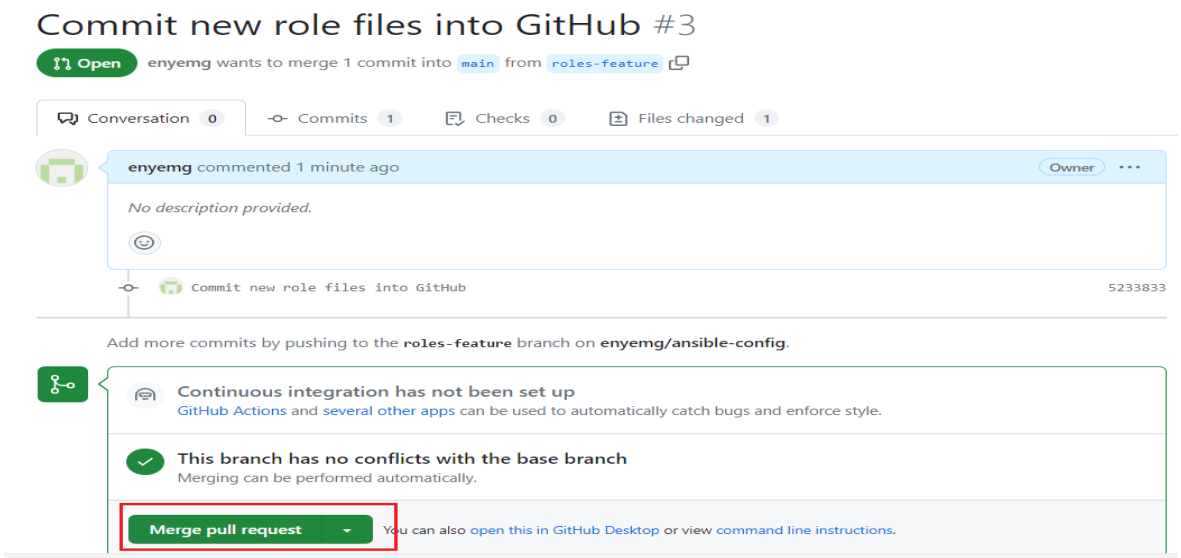
`git push --set-upstream origin roles-feature`

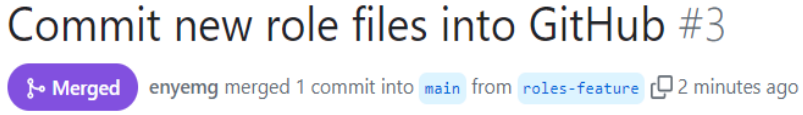
```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/mysql/defaults$ git add .
ubuntu@ip-172-31-40-224:~/ansible-config/roles/mysql/defaults$ git commit -m "Commit new role files into GitHub"
[roles-feature 5233833] Commit new role files into GitHub
 1 file changed, 134 insertions(+)
 create mode 100644 roles/mysql/defaults/main.yml
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/mysql/defaults$ git push --set-upstream origin roles-feature
Username for 'https://github.com': enyemg
Password for 'https://enyemg@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 1.90 KiB | 1.90 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'roles-feature' on GitHub by visiting:
remote:      https://github.com/enyemg/ansible-config/pull/new/roles-feature
remote:
To https://github.com/enyemg/ansible-config.git
 * [new branch]      roles-feature -> roles-feature
Branch 'roles-feature' set up to track remote branch 'roles-feature' from 'origin'.
```

7. Click on compare & pull request. Select Create pull request and then merge, merge the pull request and confirm merge.







Step 4 - Load Balancer Roles

I want to be able to choose which Load Balancer to use, Nginx or Apache, so we need to have two roles respectively: Nginx and Apache. Decide if you want to develop your own roles, or find available ones from the community. I used the available ones from the community roles here: https://galaxy.ansible.com/home and updated both static-assignment and site.yml files to refer to the roles. Since one cannot use both Nginx and Apache load balancer, I added a condition to enable either one –with the help of variables.

1. Install Apache and Nginx

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles$ ansible-galaxy install geerlingguy.nginx -p .
- downloading role 'nginx', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-nginx/archive/3.1.4.tar.gz
- extracting geerlingguy.nginx to /home/ubuntu/ansible-config/roles/geerlingguy.nginx
- geerlingguy.nginx (3.1.4) was installed successfully
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles$ ansible-galaxy install geerlingguy.apache -p .
- downloading role 'apache', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-apache/archive/3.3.0.tar.gz
- extracting geerlingguy.apache to /home/ubuntu/ansible-config/roles/geerlingguy.apache
- geerlingguy.apache (3.3.0) was installed successfully
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles$ mv geerlingguy.nginx/ nginxRole
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles$ mv geerlingguy.apache/ apacheRole
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles$ ls
apacheRole  mysql  nginxRole  webserver
```

2. Declare a variable in defaults/main.yml file inside the Nginx and Apache roles. Name each variable enable_nginx_lb and enable_apache_lb respectively. Set both values to false like this enable_nginx_lb: false and enable_apache_lb: false.

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/apacheRole/defaults$ sudo vi main.yml
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/nginxRole/defaults$ sudo vi main.yml
```

Now go to task/main.yml and make the below changes

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/nginxRole/tasks$ sudo vi main.yml
```

```
# Nginx setup.
- name: Copy nginx configuration in place.
  become: true
  template:
    src: "{{ nginx_conf_template }}"
    dest: "{{ nginx_conf_file_path }}"
    owner: root
    group: "{{ root_group }}"
    mode: 0644
  notify:
    - reload nginx

- name: Ensure nginx service is running as configured.
  become: true
  service:
    name: nginx
```

Delete the extra ansible_os_family and leave at red hat

```
# Setup/install tasks.
- include_tasks: setup-RedHat.yml
  when: ansible_os_family == 'RedHat'
```

3. Declare another variable in both roles load_balancer_is_required and set its value to false as well.

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/nginxRole/defaults$ sudo vi main.yml
```

```
ubuntu@ip-172-31-40-224:~/ansible-config/roles/apacheRole/defaults$ sudo vi main.yml
```

```
load_balancer_is_required: false
```

Note: The above changes are not sufficient, the exact changes to the roles can be found in my github repository: https://github.com/enyemg/ansible-config

Update both dynamic assignment and site.yml files respectively

loadbalancers.yml file

```
ubuntu@ip-172-31-40-224:~/ansible-config/static-assignments$ sudo vi loadbalancers.yml
```

```
- hosts: lb
  roles:
    - { role: nginx, when: enable_nginx_lb and load_balancer_is_required }
    - { role: apache, when: enable_apache_lb and load_balancer_is_required }
```

site.yml file

```
ubuntu@ip-172-31-40-224:~/ansible-config/playbooks$ sudo vi site.yml
```

```
    - name: Loadbalancers assignment
      hosts: lb
      - import_playbook: ../static-assignments/loadbalancers.yml
      when: load_balancer_is_required
```

Now you can make use of env-vars\uat.yml file to define which load balancer to use in the UAT environment by setting the respective environmental variable to true. I will activate the load balancer, and enable nginx by setting these in the respective environment's env-vars file.

```
ubuntu@ip-172-31-40-224:~/ansible-config/env-vars$ sudo vi uat.yml
```

```
enable_nginx_lb: true
load_balancer_is_required: true
```

For my file to work, I added: **enable_apache_lb: true** and **load_balancer_is_required: true** to ansible-config/roles/apacheRole/defaults/main.yml. You can switch it by setting respective environmental variables to true and others to false for nginx. To test this, you can update the inventory for each environment and run Ansible against each environment.

Now run the playbook and you should have the below results. Note that in this instance, I am

only running the apacheRole so the nginxRole configuration will be skipped.

```
ubuntu@ip-172-31-40-224:~/ansible-config$ ansible-playbook -i inventory/uat.yml playbooks/site.yml
```

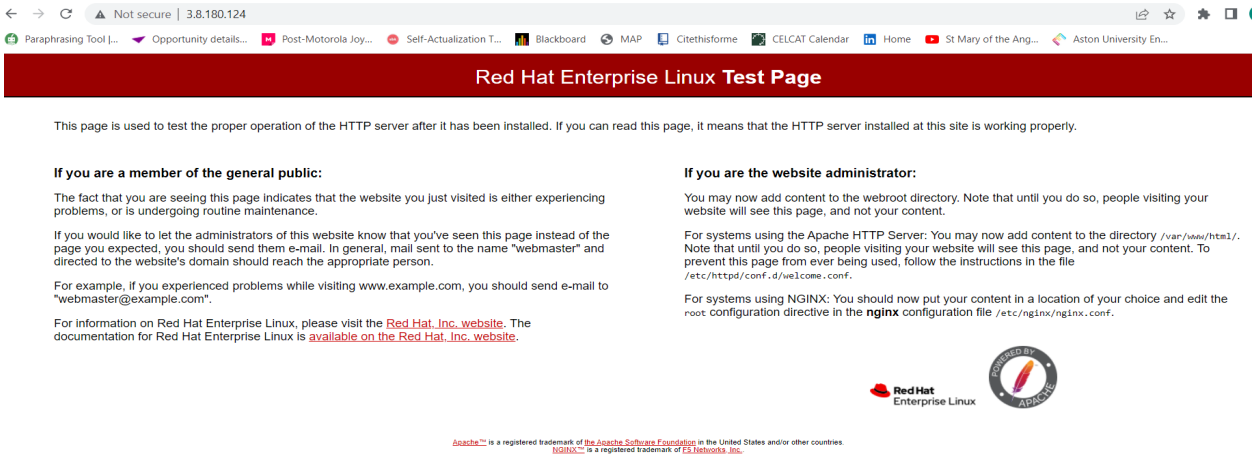
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                                   bash - ansible-config

PLAY [webservers] ************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [172.31.34.12]
ok: [172.31.34.164]

TASK [/home/ubuntu/ansible-config/roles/webserver : install apache] **********************************************
ok: [172.31.34.12]
ok: [172.31.34.164]

TASK [/home/ubuntu/ansible-config/roles/webserver : install git] *************************************************
ok: [172.31.34.12]
ok: [172.31.34.164]

TASK [/home/ubuntu/ansible-config/roles/webserver : clone a repo] ************************************************
changed: [172.31.34.12]
changed: [172.31.34.164]

TASK [/home/ubuntu/ansible-config/roles/webserver : copy html content to one level up] ***************************
changed: [172.31.34.12]
changed: [172.31.34.164]

TASK [/home/ubuntu/ansible-config/roles/webserver : Start service httpd, if not started] *************************
ok: [172.31.34.12]
ok: [172.31.34.164]

TASK [/home/ubuntu/ansible-config/roles/webserver : recursively remove /var/www/html/html/ directory] ************
changed: [172.31.34.164]
changed: [172.31.34.12]

PLAY [lb] ********************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Include OS-specific variables.] ******************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Define nginx_user.] ******************************************
skipping: [172.31.45.62]
```

```
TASK [/home/ubuntu/ansible-config/roles/nginxRole : include_tasks] ***********************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : include_tasks] ***********************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : include_tasks] ***********************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : include_tasks] ***********************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : include_tasks] ***********************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : include_tasks] ***********************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Remove default nginx vhost config file (if configured).] *****
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Ensure nginx_vhost_path exists.] *****************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Add managed vhost config files.] *****************************

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Remove managed vhost config files.] **************************

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Remove legacy vhosts.conf file.] *****************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : set webservers host name in /etc/hosts] **********************
skipping: [172.31.45.62] => (item={'name': 'web1', 'ip': '172.31.34.12'})
skipping: [172.31.45.62] => (item={'name': 'web2', 'ip': '172.31.34.164'})

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Copy nginx configuration in place.] **************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/nginxRole : Ensure nginx service is running as configured.] **************
skipping: [172.31.45.62]
```

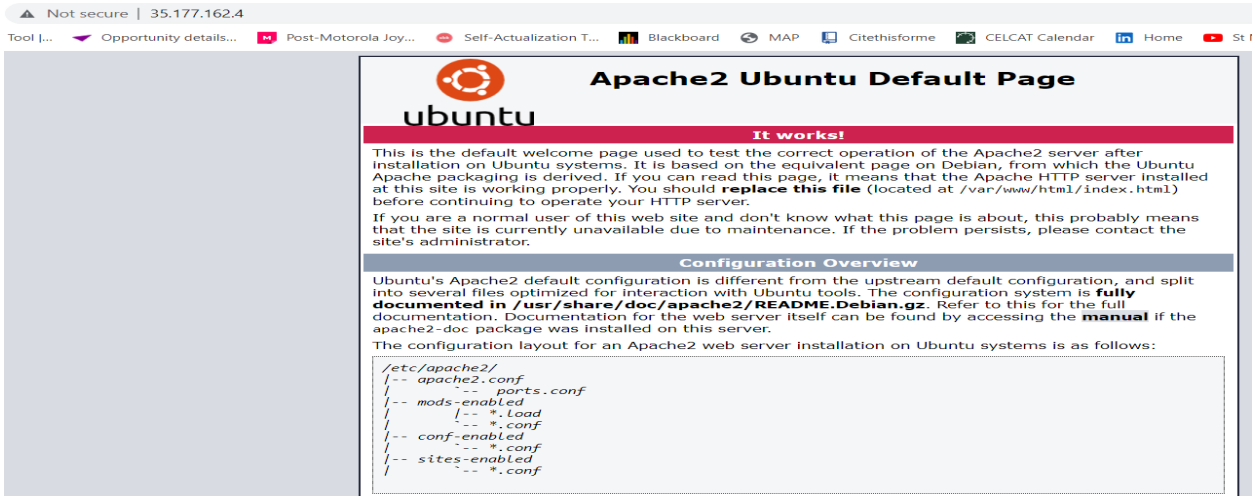
```
TASK [/home/ubuntu/ansible-config/roles/apacheRole : Include OS-specific variables.] ********************************************************************
ok: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Include variables for Amazon Linux.] ********************************************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Define apache_packages.] ********************************************************************
ok: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : include_tasks] ********************************************************************
included: /home/ubuntu/ansible-config/roles/apacheRole/tasks/setup-Debian.yml for 172.31.45.62

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Update apt cache.] ********************************************************************
changed: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Ensure Apache is installed on Debian.] ********************************************************************
changed: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Get installed version of Apache.] ********************************************************************
ok: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Create apache_version variable.] ********************************************************************
ok: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Include Apache 2.2 variables.] ********************************************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Include Apache 2.4 variables.] ********************************************************************
ok: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Configure Apache.] ********************************************************************
included: /home/ubuntu/ansible-config/roles/apacheRole/tasks/configure-Debian.yml for 172.31.45.62

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Configure Apache.] ********************************************************************
ok: [172.31.45.62] => (item={'regexp': '^Listen ', 'line': 'Listen 80'})
```

```
TASK [/home/ubuntu/ansible-config/roles/apacheRole : Enable Apache mods.] ********************************************************************
changed: [172.31.45.62] => (item=rewrite)
changed: [172.31.45.62] => (item=ssl)

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Disable Apache mods.] ********************************************************************

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Check whether certificates defined in vhosts exist.] ********************************************************************

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Add apache vhosts configuration.] ********************************************************************
changed: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Add vhost symlink in sites-enabled.] ********************************************************************
changed: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Remove default vhost in sites-enabled.] ********************************************************************
skipping: [172.31.45.62]

TASK [/home/ubuntu/ansible-config/roles/apacheRole : Ensure Apache has selected state and enabled on boot.] ********************************************************************
ok: [172.31.45.62]

RUNNING HANDLER [/home/ubuntu/ansible-config/roles/apacheRole : restart apache] ********************************************************************
changed: [172.31.45.62]

PLAY RECAP ********************************************************************************************************************************************
172.31.34.12               : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.34.164              : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.45.62               : ok=16   changed=6    unreachable=0    failed=0    skipped=21   rescued=0    ignored=0
```

Now when you browse the webservers and load balancer IP's, you should have their respective pages as below. I have successfully used the Ansible configuration management tool to prepare a UAT environment for the Tooling web solution.


Not secure | 3.8.180.124
Red Hat Enterprise Linux Test Page
This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.
If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.
If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.
For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".
For information on Red Hat Enterprise Linux, please visit the Red Hat, Inc. website. The documentation for Red Hat Enterprise Linux is available on the Red Hat, Inc. website.
If you are the website administrator:
You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.
For systems using the Apache HTTP Server: You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.
For systems using NGINX: You should now put your content in a location of your choice and edit the root configuration directive in the nginx configuration file /etc/nginx/nginx.conf.
Red Hat Enterprise Linux
Apache™ is a registered trademark of the Apache Software Foundation in the United States and/or other countries.
NGINX™ is a registered trademark of F5 Networks, Inc.


Not secure | 13.41.192.231
Red Hat Enterprise Linux Test Page
This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.
If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.
If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.
For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".
For information on Red Hat Enterprise Linux, please visit the Red Hat, Inc. website. The documentation for Red Hat Enterprise Linux is available on the Red Hat, Inc. website.
If you are the website administrator:
You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.
For systems using the Apache HTTP Server: You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.
For systems using NGINX: You should now put your content in a location of your choice and edit the root configuration directive in the nginx configuration file /etc/nginx/nginx.conf.
Red Hat Enterprise Linux
Apache™ is a registered trademark of the Apache Software Foundation in the United States and/or other countries.
NGINX™ is a registered trademark of F5 Networks, Inc.


Not secure | 35.177.162.4
Apache2 Ubuntu Default Page
ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf