1. Navigate to ansible roles directory

   Syntax: cd /etc/ansible/roles

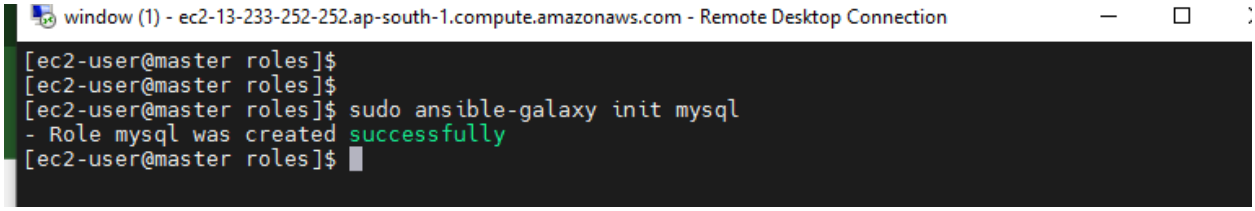
```
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$
```

2. Create the roles ansible roles called apache, php, mysql and wordpress.

   Syntax: sudo ansible-galaxy init apache

```
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$ sudo ansible-galaxy init apache
- Role apache was created successfully
[ec2-user@master roles]$
```

   Syntax: sudo ansible-galaxy init mysql



   Syntax: sudo ansible-galaxy init php

```
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$ sudo ansible-galaxy init php
- Role php was created successfully
[ec2-user@master roles]$
```

   Syntax: sudo ansible-galaxy init wordpress

```
[ec2-user@master roles]$ sudo ansible-galaxy init wordpress
- Role wordpress was created successfully
[ec2-user@master roles]$
```

3. Now edit the main.yml within the apache/tasks directory

   Syntax: sudo vi apache/tasks/main.yml

```
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$ sudo vi apache/tasks/main.yml
```

4. Playbook code:

```
---
# tasks file for apache

- name: Install HTTP Packages
  yum: name=httpd update_cache=yes state=latest

- name: Start httpd service
  systemd: name=httpd state=started enabled=yes

- name: Create Apache Document Root
  file:
    path: "/var/www/{{ http_host }}"
    state: directory
    owner: "apache"
    group: "apache"
    mode: '0755'

- name: Set up Apache VirtualHost
  template:
    src: "files/httpd.conf"
    dest: "/etc/httpd/conf.d/{{ http_conf }}"
    owner: root
    group: root
    mode: u=rw,g=r,o=r
```

5. edit the main.yml within the apache/vars directory

   Syntax: sudo vi apache/vars/main.yml

```
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$ sudo vi apache/vars/main.yml
[ec2-user@master roles]$
```

6. Playbook code:

```
---
# vars file for apache
#HTTP Settings
http_host: "wp.example.com"
```

http_conf: "wp.example.com.conf"
http_port: "80"

7. edit the main.yml within the apache/files directory

   Syntax: sudo vi apache/files/httpd.conf

```
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$ sudo vi apache/files/httpd.conf
```

8. Playbook code:

```
<VirtualHost *:{{ http_port }}>
  ServerAdmin webmaster@localhost
  ServerName {{ http_host }}
  ServerAlias www.{{ http_host }}
  DocumentRoot /var/www/{{ http_host }}/wordpress
  ErrorLog /var/log/httpd/error.log
  CustomLog /var/log/httpd/access.log combined

  <Directory /var/www/{{ http_host }}/wordpress>
      Options Indexes FollowSymLinks
   AllowOverride all
   Require all granted
  </Directory>
</VirtualHost>
```

9. Now edit the main.yml within the php/tasks directory

   Syntax: sudo vi php/tasks/main.yml

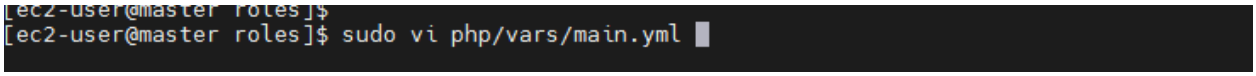
```
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$ sudo vi php/tasks/main.yml
```

10. Playbook code:

```
- name: Install PHP Extensions
  yum: name={{ item }} update_cache=yes state=latest
  loop: "{{ php_modules }}"
```

11. Edit the main.yml within the php/vars directory

Syntax: sudo vi php/vars/main.yml



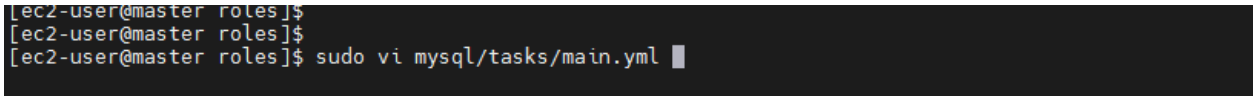
12. Playbook code:

---
vars file for php

php_modules: ['php', 'php-curl', 'php-gd', 'php-mbstring', 'php-xml', 'php-xmlrpc', 'php-soap', 'php-intl', 'php-zip']

13. Now edit the main.yml within the mysql/tasks directory

Syntax: sudo vi mysql/tasks/main.yml



14. Playbook code:

tasks file for mysql

```
- name: Install MySQL Packages
  yum: name={{ item }} update_cache=yes state=latest
  loop:
    - mariadb-server
    - MySQL-python
    - libselinux-python
    - libsemanage-python

- name: Start mysqld service
  systemd: name=mariadb state=started enabled=yes

- name: Set MySQL root Password
  mysql_user:
    login_host: 'localhost'
    login_user: 'root'
    login_password: ''
    name: 'root'
```

```
    password: '{{ mysql_root_password }}'
    state: present

- name: Creates database for WordPress
  mysql_db:
    name: "{{ mysql_db }}"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

- name: Create MySQL user for WordPress
  mysql_user:
    name: "{{ mysql_user }}"
    password: "{{ mysql_password }}"
    priv: "{{ mysql_db }}.*:ALL"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"
```

15. edit the main.yml within the mysql/vars directory

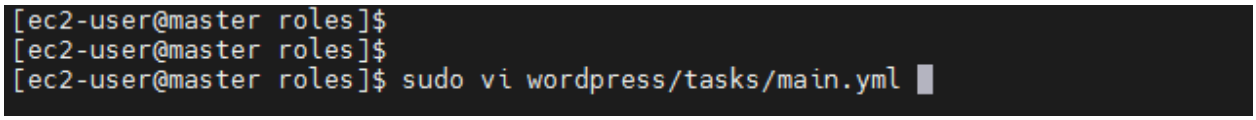
    Syntax: sudo vi mysql/vars/main.yml

16. Playbook code:

```
---
# vars file for mysql
mysql_root_password: "password"
mysql_db: "wpdb"
mysql_user: "wpuser"
mysql_password: "password"
```

17. edit the main.yml within the wordpress/tasks directory

    Syntax: sudo vi wordpress/tasks/main.yml



18. Playbook code:

```
---
# tasks file for wordpress

    - name: Download and unpack latest WordPress
      unarchive:
        src: https://wordpress.org/latest.tar.gz
        dest: "/var/www/{{ http_host }}"
        remote_src: yes
        creates: "/var/www/{{ http_host }}/wordpress"

    - name: Set ownership
      file:
        path: "/var/www/{{ http_host }}"
        state: directory
        recurse: yes
        owner: apache
        group: apache

    - name: Set permissions for directories
      shell: "/usr/bin/find /var/www/{{ http_host }}/wordpress/ -type d -exec chmod 750 {}
\\;"

    - name: Set permissions for files
      shell: "/usr/bin/find /var/www/{{ http_host }}/wordpress/ -type f -exec chmod 640 {} \\;"

    - name: Copy sample config file
      command: mv /var/www/{{ http_host }}/wordpress/wp-config-sample.php /var/www/{{
http_host }}/wordpress/wp-config.php creates=/var/www/{{ http_host
}}/wordpress/wp-config.php
      become: yes

    - name: Update WordPress config file
      lineinfile:
        path: "/var/www/{{ http_host }}/wordpress/wp-config.php"
        regexp: "{{item.regexp}}"
        line: "{{item.line}}"
      with_items:
        - {'regexp': "define\\( 'DB_NAME', '(.)+' \\);", 'line': "define( 'DB_NAME',
'{{mysql_db}}' );"}
        - {'regexp': "define\\( 'DB_USER', '(.)+' \\);", 'line': "define( 'DB_USER',
'{{mysql_user}}' );"}
        - {'regexp': "define\\( 'DB_PASSWORD', '(.)+' \\);", 'line': "define( 'DB_PASSWORD',
'{{mysql_password}}' );"}
```

```
- name: Restart httpd service
  systemd: name=httpd state=restarted
  become: yes
```

19. edit the main.yml within the wordpress/vars directory

Syntax: sudo vi wordpress/vars/main.yml

```
[ec2-user@master roles]$
[ec2-user@master roles]$ sudo vi wordpress/vars/main.yml
```

20. Playbook code:

```
---
# vars file for wordpress
#PHP Settings
php_modules: [ 'php', 'php-curl', 'php-gd', 'php-mbstring', 'php-xml', 'php-xmlrpc', 'php-soap',
'php-intl', 'php-zip' ]

#MySQL Settings
mysql_root_password: "your-root-password"
mysql_db: "wpdb"
mysql_user: "wpuser"
mysql_password: "password"

#HTTP Settings
http_host: "wp.example.com"
http_conf: "wp.example.com.conf"
http_port: "80"
```

21. Now create the site.yml on /etc/ansible/role/site.yml

Syntax: /etc/ansible/role/site.yml

```
[ec2-user@master roles]$
[ec2-user@master roles]$
[ec2-user@master roles]$ sudo vi site.yml
[ec2-user@master roles]$
```

22. Playbook code:

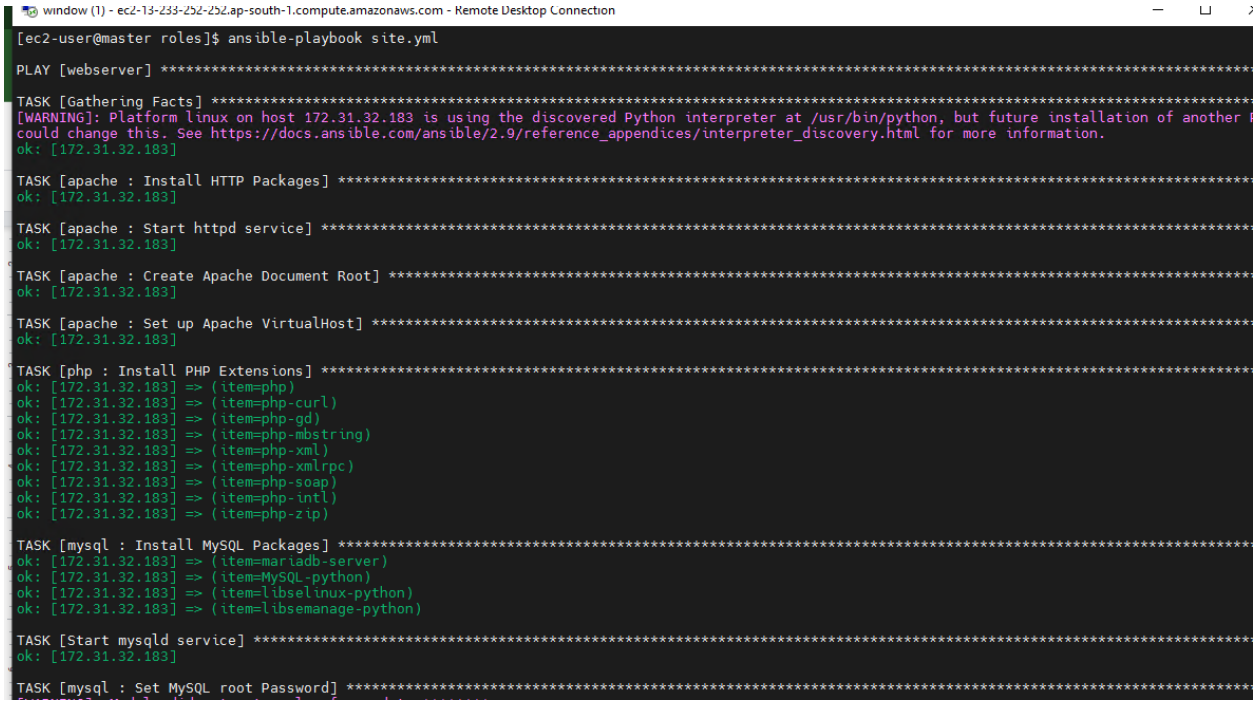
```
---
- hosts: webserver
```

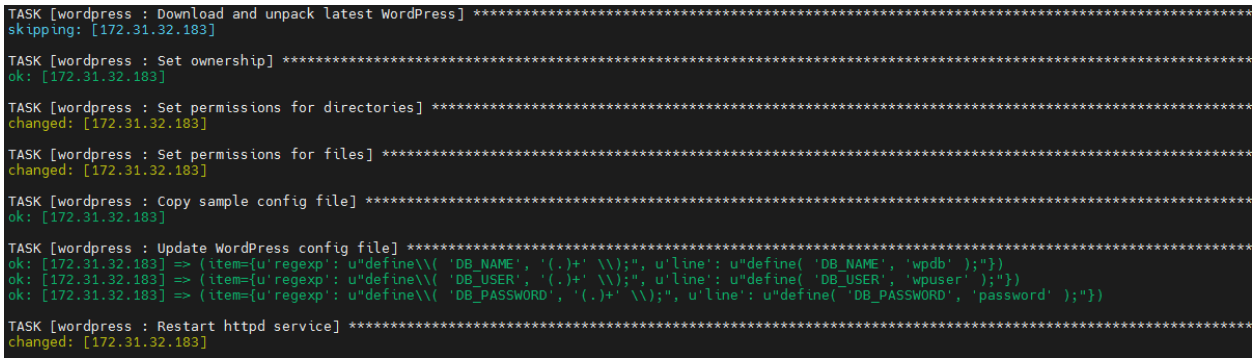
```
  become: true
  become_user: root
  roles:
    - role: apache
      tags:
        - apache
    - role: php
      tags:
        - php
    - role: mysql
      tags:
        - mysql
    - role: wordpress
      tags:
        - wordpress
```

23. Finally execute the playbook

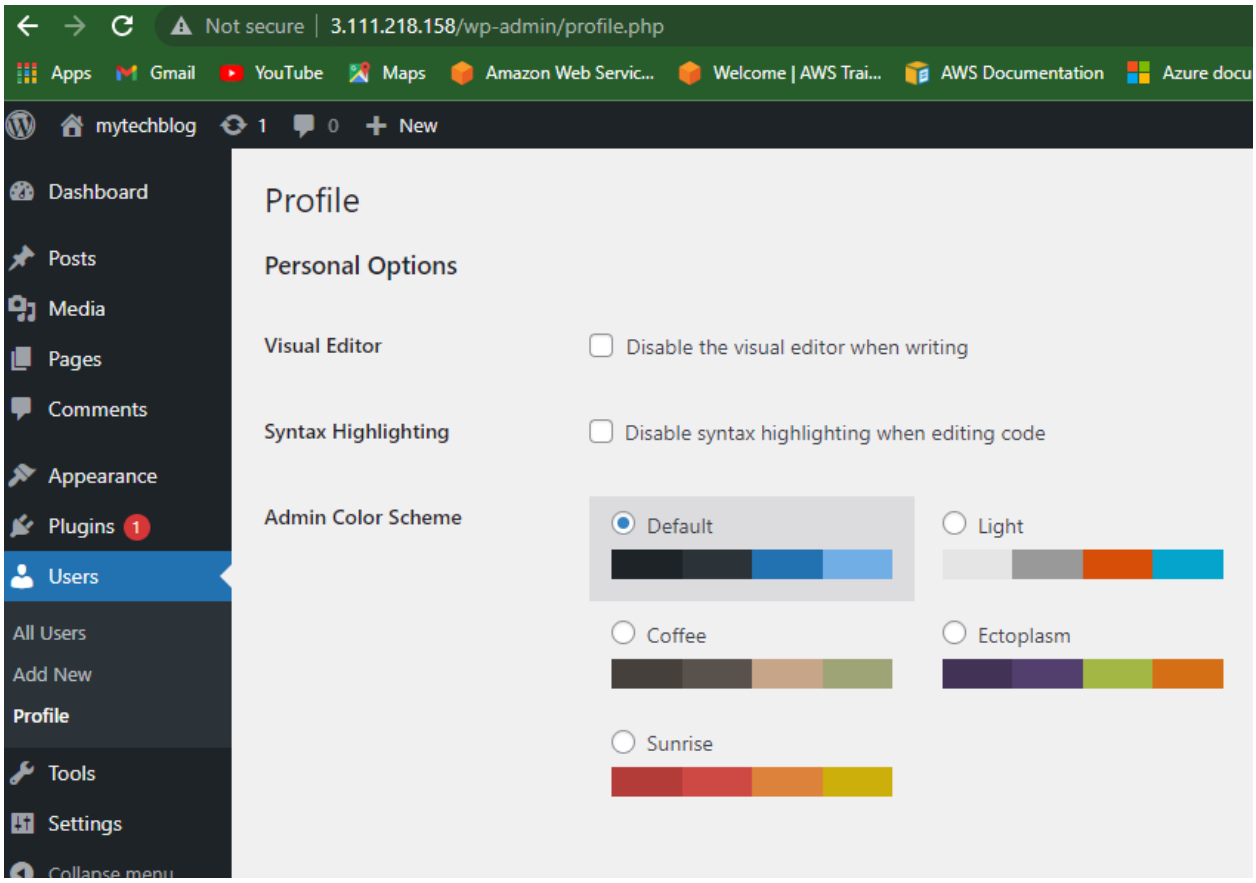
Syntax: ansible-playbook site.yml

24. Output:-





25. Access the wordpress url http://3.111.218.158/


Not secure | 3.111.218.158/wp-admin/profile.php
Apps
Gmail
YouTube
Maps
Amazon Web Servic...
Welcome | AWS Trai...
AWS Documentation
Azure docu
mytechblog
1
0
New
Dashboard
Posts
Media
Pages
Comments
Appearance
Plugins 1
Users
All Users
Add New
Profile
Tools
Settings
Collapse menu
Profile
Personal Options
Visual Editor
Disable the visual editor when writing
Syntax Highlighting
Disable syntax highlighting when editing code
Admin Color Scheme
Default
Light
Coffee
Ectoplasm
Sunrise