Vim ansible.yml

```
---
- name: internship project
  hosts: all
  become: yes
  vars:
    ansible_python_interpreter: /usr/bin/python3

  tasks:
    - name: Install packages for nginx, mariadb, and php
      apt:
        name:
          - nginx
          - mariadb-server
          - mariadb-client
          - php8.1
          - php8.1-fpm
          - php8.1-mysql
          - php8.1-gd
          - php8.1-curl
          - php8.1-xml
          - php8.1-mbstring
          - php8.1-zip
          - php8.1-cli
          - php8.1-common
          - php8.1-opcache
          - php8.1-readline
          - php8.1-imagick
          - php-phpseclib
```

```
      - php-php-gettext
    state: present

- name: Enable and start services
  service:
    name: "{{ item }}"
    state: started
    enabled: yes
  loop:
    - nginx
    - mariadb
    - php8.1-fpm

- name: Secure mariadb
  become: yes
  expect:
    command: mysql_secure_installation
    responses:
      'Enter current password for root (enter for none):': ''
      'Switch to unix_socket authentication [Y/n]:': 'n'
      'Change the root password? [Y/n]:': 'n'
      'Remove anonymous users? [Y/n]:': ''
      'Disallow root login remotely? [Y/n]:': ''
      'Remove test database and access to it? [Y/n]:': ''
      'Reload privilege tables now? [Y/n]:': ''
    timeout: 10
  register: secure_mariadb
  failed_when: "'... Failed!' in secure_mariadb.stdout_lines"

- name: Add user
```

```
  user:
    name: dani
    password: "{{ 'dani' | password_hash('sha512') }}"
    state: present

- name: Create directory structure
  file:
    path: /home/dani/myweb/public
    state: directory
    owner: dani
    group: dani
    mode: 0755

- name: Change ownership of directory
  file:
    path: /home/dani/myweb
    state: directory
    owner: dani
    group: dani
    recurse: yes

- name: Install package for vsftpd
  ansible.builtin.apt:
    name: vsftpd
    state: present

- name: Enable and start services
  ansible.builtin.service:
    name: vsftpd
    state: started
```

```
    enabled: yes

- name: Copy vsftpd.conf to vsftpd.conf.orig
  ansible.builtin.copy:
    src: /etc/vsftpd.conf
    dest: /etc/vsftpd.conf.orig

- name: Create /home/dani/ftp directory
  ansible.builtin.file:
    path: /home/dani/ftp
    state: directory

- name: Set ownership of /home/dani/ftp directory to nobody:nogroup
  ansible.builtin.file:
    path: /home/dani/ftp
    owner: nobody
    group: nogroup

- name: Remove write permissions from /home/dani/ftp directory
  ansible.builtin.file:
    path: /home/dani/ftp
    mode: 'a-w'
    recurse: yes
  become: true

- name: Set permissions of /home/dani/ftp directory
  ansible.builtin.file:
    path: /home/dani/ftp
    mode: '0755'
```

```
- name: Create /home/dani/ftp/files directory
  ansible.builtin.file:
    path: /home/dani/ftp/files
    state: directory

- name: Set ownership of /home/dani/ftp/files directory to dani:dani
  ansible.builtin.file:
    path: /home/dani/ftp/files
    owner: dani
    group: dani

- name: Create test.txt file with content
  ansible.builtin.copy:
    content: "vsftpd daniels test file"
    dest: /home/dani/ftp/files/test.txt
    owner: dani
    group: dani
    mode: '0644'

- name: Generate SSL certificate
  ansible.builtin.command: |
    openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem \
    -subj "/C=AU/ST=Some-State/L=/O=Internet Widgits Pty Ltd/OU=/CN=13.233.72.36"
  args:
    creates: /etc/ssl/private/vsftpd.pem

- name: Configure vsftpd
  blockinfile:
    path: "/etc/vsftpd.conf"
```

```
    block: |
      write_enable=YES
      use_localtime=YES
      user_sub_token=$USER
      local_root=/home/$USER/ftp
      pasv_min_port=40000
      pasv_max_port=50000
      pasv_min_port=40000
      pasv_max_port=50000
      userlist_enable=YES
      userlist_file=/etc/vsftpd.userlist
      userlist_deny=NO
      rsa_cert_file=/etc/ssl/private/vsftpd.pem
      rsa_private_key_file=/etc/ssl/private/vsftpd.pem
      ssl_enable=YES
      allow_anon_ssl=NO
      force_local_data_ssl=YES
      force_local_logins_ssl=YES
      ssl_tlsv1=YES
      ssl_sslv2=NO
      ssl_sslv3=NO
      require_ssl_reuse=NO
      ssl_ciphers=HIGH
    marker: "# {mark} ANSIBLE MANAGED BLOCK"

- name: Update chroot_local_user and rsa_cert_file in vsftpd.conf
  ansible.builtin.replace:
    path: /etc/vsftpd.conf
    regexp: '^#?(chroot_local_user\s*=).*$'
    replace: '\1YES'
```

```
    - name: Comment out rsa_cert_file and rsa_private_key_file in vsftpd.conf
      ansible.builtin.replace:
        path: /etc/vsftpd.conf
        regexp: '^#?( rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem|
ssl_enable=NO|rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key)\s*=.*$'
        replace: '#\g<0>'

    - name: Add user to vsftpd userlist
      ansible.builtin.lineinfile:
        path: /etc/vsftpd.userlist
        line: "dani"
        create: yes

    - name: Enable and start vsftpd
      ansible.builtin.service:
        name: vsftpd
        state: started
        enabled: yes

    - name: Download phpMyAdmin zip file
      ansible.builtin.get_url:
        url: https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.zip
        dest: /tmp/phpMyAdmin-latest-all-languages.zip

    - name: Install unzip package
      ansible.builtin.apt:
        name: unzip
        state: present
```

```
- name: Extract phpMyAdmin zip file
  ansible.builtin.unarchive:
    src: /tmp/phpMyAdmin-latest-all-languages.zip
    dest: /home/
    remote_src: yes

- name: Create /var/www/ directory
  ansible.builtin.file:
    path: /var/www/
    state: directory

- name: Check if phpMyAdmin directory exists
  ansible.builtin.stat:
    path: /var/www/phpmyadmin
  register: phpmyadmin_dir

- name: Move phpMyAdmin directory
  ansible.builtin.shell: mv /home/phpMyAdmin-5.2.1-all-languages /var/www/phpmyadmin
  when: phpmyadmin_dir.stat.exists == False

- name: Install pip3
  ansible.builtin.package:
    name: python3-pip
    state: present

- name: Install PyMySQL
  ansible.builtin.pip:
    name: pymysql

- name: Change ownership of phpMyAdmin directory
```

```
    ansible.builtin.file:
      path: /var/www/phpmyadmin
      owner: www-data
      group: www-data
      recurse: yes

  - name: Execute MySQL commands to create database and user
    shell: |
      mysql -u root -p'redhat' <<EOF
      CREATE DATABASE IF NOT EXISTS phpmyadmin DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
      GRANT ALL ON phpmyadmin.* TO 'dani'@'localhost' IDENTIFIED BY 'dani@1';
      FLUSH PRIVILEGES;
      EOF

  - name: Install PHP packages
    ansible.builtin.apt:
      name:
        - php-imagick
        - php-phpseclib
        - php-php-gettext
        - php8.1-common
        - php8.1-mysql
        - php8.1-gd
        - php8.1-imap
        - php8.1-curl
        - php8.1-zip
        - php8.1-xml
        - php8.1-mbstring
        - php8.1-bz2
```

```
      - php8.1-intl
      - php8.1-gmp
    state: present

- name: Enable and start nginx
  ansible.builtin.service:
    name: nginx
    state: started
    enabled: yes

- name: Install Certbot and Certbot Nginx plugin
  ansible.builtin.apt:
    name:
      - certbot
      - python3-certbot-nginx
    state: present

- name: Download WordPress zip file
  ansible.builtin.get_url:
    url: https://wordpress.org/latest.zip
    dest: /tmp/wordpress-latest.zip

- name: Unzip WordPress to /usr/share/nginx
  ansible.builtin.unarchive:
    src: /tmp/wordpress-latest.zip
    dest: /var/www/
    remote_src: yes

- name: Change ownership of wordpress directory
  ansible.builtin.file:
```

```
    path: /var/www/wordpress
    owner: www-data
    group: www-data
    recurse: yes

- name: Copy wp-config-sample.php to wp-config.php
  ansible.builtin.copy:
    src: /var/www/wordpress/wp-config-sample.php
    dest: /var/www/wordpress/wp-config.php

- name: Replace database configuration in wp-config.php
  ansible.builtin.replace:
    path: /var/www/wordpress/wp-config.php
    regexp: "(define\\(\\s*'DB_NAME',\\s*').*('\\s*\\);)"
    replace: "\\1wordpress\\2"

- name: Replace database user configuration in wp-config.php
  ansible.builtin.replace:
    path: /var/www/wordpress/wp-config.php
    regexp: "(define\\(\\s*'DB_USER',\\s*').*('\\s*\\);)"
    replace: "\\1daniel\\2"

- name: Replace database password configuration in wp-config.php
  ansible.builtin.replace:
    path: /var/www/wordpress/wp-config.php
    regexp: "(define\\(\\s*'DB_PASSWORD',\\s*').*('\\s*\\);)"
    replace: \\1dani@11\\2

- name: Create wordpress.com configuration file
  ansible.builtin.template:
```

```
    src: wordpress.conf.j2
    dest: /etc/nginx/sites-enabled/wordpress.conf

- name: Reload Nginx service
  ansible.builtin.systemd:
    name: nginx
    state: reloaded

- name: Remove default Nginx configuration files if they exist
  ansible.builtin.file:
    path: "{{ item }}"
    state: absent
  ignore_errors: yes
  with_items:
    - /etc/nginx/sites-enabled/default
    - /etc/nginx/sites-available/default

- name: Install certbot package
  ansible.builtin.package:
    name: certbot
    state: present

- name: Obtain SSL certificate with certbot
  ansible.builtin.command: >
    certbot --nginx --agree-tos --redirect --hsts --staple-ocsp -d jijojohn.site --email
danielvadasserikkara@gmail.com

- name: Execute MySQL commands to create database and user
  ansible.builtin.shell: |
    sudo mariadb -u root <<EOF
```

```
    create database wordpress;
    create user daniel@localhost identified by 'dani@11';
    grant all privileges on wordpress.* to daniel@localhost;
    flush privileges;
    exit;
    EOF

# Other tasks continue...
```

vim wordpress.conf.j2

```
server {
listen 80;
listen [::]:80;
server_name jijojohn.site;
root /var/www/;
index index.php index.html index.htm index.nginx-debian.html;
error_log /var/log/nginx/wordpress.error;
```

```
access_log /var/log/nginx/wordpress.access;
location / {
try_files $uri $uri/ /index.php;
}
location ~ ^/wp-json/ {
rewrite ^/wp-json/(.*?)$ /?rest_route=/$1 last;
}
location ~* /wp-sitemap.*\.xml {
try_files $uri $uri/ /index.php$is_args$args;
}
error_page 404 /404.html;
error_page 500 502 503 504 /50x.html;
client_max_body_size 20M;
location = /50x.html {
root /usr/share/nginx/html;
}
location ~ \.php$ {
fastcgi_pass unix:/run/php/php8.1-fpm.sock;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
include fastcgi_params;
include snippets/fastcgi-php.conf;
fastcgi_buffers 1024 4k;
fastcgi_buffer_size 128k;
}
#enable gzip compression
gzip on;
gzip_vary on;
gzip_min_length 1000;
gzip_comp_level 5;
gzip_types application/json text/css application/x-javascript application/javascript image/svg+xml;
```

```
gzip_proxied any;
# A long browser cache lifetime can speed up repeat visits to your page
location ~* \.(jpg|jpeg|gif|png|webp|svg|woff|woff2|ttf|css|js|ico|xml)$ {
access_log off;
log_not_found off;
expires 360d;
}
# disable access to hidden files
location ~ /\.ht {
access_log off;
log_not_found off;
deny all;
}
}
```

```
Inventory
localhost
```

```
vim ansible.cfg
[defaults]
inventory=inventory
```

```
vim /etc/ssh/sshd_config
vim /etc/ssh/sshd_config.d/60-cloudimg-settings.conf
```

```
ssh-keygen
systemctl restart sshd
passwd root
ssh-copy-id root@localhost
```

```
apt update
apt  install ansible
ansible-playbook ansible.yml
```