

## PlayBook

---

- name: Setup LAMP stack with WordPress on Amazon Linux 2

hosts: all

become: yes

vars:

domain\_name: "pooja.quest"

sftp\_pwd: "pooja\$#!"

mysql\_root\_pwd: "pooja\$#!"

phpmyadmin\_pwd: "pooja\$#!"

wordpress\_pwd: "pooja\$#!"

user\_path: "/home/pooja"

user\_website\_path: "/home/pooja/mywebsite/public"

tasks:

- name: Install and Enable PHP 8

command: amazon-linux-extras enable php8.0

- name: Install Nginx, PHP8, MySQL, and Certbot

yum:

name:

- nginx

- php

- php-mysqlnd

- php-fpm

- mariadb

- mariadb-server

- certbot

- certbot-nginx

- python2-pip

- MySQL-python

state: present

- name: Install required Python packages(MySQL-python package provides a Python interface to MySQL databases)

pip:

name: MySQL-python

state: present

executable: /usr/bin/pip2

- name: Start and enable Nginx, PHP and MariaDB

systemd:

name: "{{ item }}"

state: started

enabled: yes

loop:

- nginx

- mariadb

- php-fpm

- name: Check MariaDB is running

systemd:

name: mariadb

state: started

enabled: yes

- name: Initialize MySQL data directory

command: mysql\_install\_db

args:

creates: /var/lib/mysql/mysql

- name: Set MySQL password for user root

mysql\_user:

login\_user: root

login\_password: "{{ mysql\_root\_pwd }}"

user: root

password: "{{ mysql\_root\_pwd }}"

host\_all: yes

state: present

- name: Remove anonymous MySQL users

mysql\_user:

name: "

host\_all: yes

state: absent

login\_user: root

login\_password: "{{ mysql\_root\_pwd }}"

- name: Remove MySQL test database

mysql\_db:

name: test

state: absent

login\_user: root

login\_password: "{{ mysql\_root\_pwd }}"

- name: Create MySQL Database for user root(DB Name wordpress)

mysql\_db:

name: wordpress

state: present

login\_user: root

login\_password: "{{ mysql\_root\_pwd }}"

- name: Create MySQL user for WordPress(Here user set as wordpress)

mysql\_user:

name: wordpress

password: "{{ wordpress\_pwd }}"

priv: "wordpress.\*:ALL"

state: present

login\_user: root

login\_password: "{{ mysql\_root\_pwd }}"

- name: Create user for SFTP (username pooja)

user:

name: pooja

home: "{{ user\_path }}"

shell: /bin/bash

create\_home: yes

password: "{{ sftp\_pwd | password\_hash('sha512') }}"

- name: Create Directory for website

file:

path: "{{ user\_website\_path }}"

state: directory

owner: pooja

group: pooja

mode: '0755'

notify: Restart Nginx and PHP-FPM

- name: Ensure /home/pooja has execute permission for others

file:

path: /home/pooja

mode: '0751'

- name: Ensure sshd\_config.d directory exists

file:

path: /etc/ssh/sshd\_config.d

state: directory

- name: Set up SFTP access

copy:

content: |

Match User pooja

ChrootDirectory {{ user\_path }}

ForceCommand internal-sftp

AllowTcpForwarding no

PermitTunnel no

X11Forwarding no

dest: /etc/ssh/sshd\_config.d/sftp\_pooja.conf

- name: Create SSL certification

command: certbot --nginx -d {{ domain\_name }} --non-interactive --agree-tos -m  
admin@{{ domain\_name }}

- name: Ensure snippets directory exists for Nginx

file:

path: /etc/nginx/snippets

state: directory

owner: root

group: root

mode: '0755'

- name: Ensure fastcgi-php.conf exists (It handling PHP FastCGI parameters in Nginx)

copy:

content: |

fastcgi\_param SCRIPT\_FILENAME \$document\_root\$fastcgi\_script\_name;

include fastcgi\_params;

dest: /etc/nginx/snippets/fastcgi-php.conf

owner: root

group: root

mode: '0644'

- name: Install phpMyAdmin

get\_url:

url: <https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz>

dest: /tmp/phpMyAdmin.tar.gz

- name: Extract phpMyAdmin

unarchive:

src: /tmp/phpMyAdmin.tar.gz

dest: /usr/share/

remote\_src: yes

creates: /usr/share/phpMyAdmin

- name: Ensure phpMyAdmin directory exists

file:

path: /usr/share/phpMyAdmin

state: directory

- name: Copy phpMyAdmin files to /usr/share/phpMyAdmin

copy:

src: /usr/share/phpMyAdmin-5.2.1-all-languages/

dest: /usr/share/phpMyAdmin/

remote\_src: yes

force: yes

- name: Create phpMyAdmin config file

copy:

content: |

```
<?php
```

```
$cfg['blowfish_secret'] = '{{ phpmyadmin_pwd }}';
```

```
$i = 0;
```

```
$i++;
```

```
$cfg['Servers'][$i]['auth_type'] = 'cookie';
```

```
$cfg['Servers'][$i]['user'] = 'root';
```

```
$cfg['Servers'][$i]['password'] = '{{ mysql_root_pwd }}';
```

```
$cfg['Servers'][$i]['host'] = 'localhost';
```

```
$cfg['Servers'][$i]['AllowNoPassword'] = false;
```

dest: /usr/share/phpMyAdmin/config.inc.php

owner: nginx

group: nginx

mode: '0644'

- name: set permission for PhpMyAdmin directory

file:

path: /usr/share/phpMyAdmin

state: directory

owner: nginx

group: nginx

mode: '0775'

- name: Set permissions for PHP sessions directory

file:

path: /var/lib/php

state: directory

owner: nginx

group: nginx

mode: '0775'

- name: Set ownership for PHP sessions directory

ansible.builtin.file:

path: /var/lib/php

owner: nginx

group: nginx

recurse: yes

become: yes

- name: Set permissions for PHP sessions directory

file:

path : /var/lib/php/session

state: directory

owner: nginx

group: nginx

mode: "0775"

recurse: yes

become: yes

- name: Set ownership for PHP-FPM directory

ansible.builtin.file:

path: /run/php-fpm

owner: nginx

group: nginx



recurse: yes

become: yes

- name: Set permissions for PHP-FPM directory

ansible.builtin.file:

path: /run/php-fpm

mode: "0755"

recurse: yes

become: yes

- name: Download and extract WordPress

unarchive:

src: <https://wordpress.org/latest.tar.gz>

dest: "{{ user\_website\_path }}"

remote\_src: yes

creates: "{{ user\_website\_path }}/wordpress"

- name: Configure WordPress wp-config.php

copy:

content: |

<?php

define('DB\_NAME', 'wordpress');

define('DB\_USER', 'wordpress');

define('DB\_PASSWORD', '{{ wordpress\_pwd }}');

define('DB\_HOST', 'localhost');

define('DB\_CHARSET', 'utf8');

define('DB\_COLLATE', '');

define('AUTH\_KEY', 'put your unique phrase here');

define('SECURE\_AUTH\_KEY', 'put your unique phrase here');

define('LOGGED\_IN\_KEY', 'put your unique phrase here');

```

define('NONCE_KEY', 'put your unique phrase here');
define('AUTH_SALT', 'put your unique phrase here');
define('SECURE_AUTH_SALT', 'put your unique phrase here');
define('LOGGED_IN_SALT', 'put your unique phrase here');
define('NONCE_SALT', 'put your unique phrase here');
$table_prefix = 'wp_';
define('WP_DEBUG', false);
if ( !defined('ABSPATH') )
    define('ABSPATH', dirname(__FILE__) . '/');
require_once(ABSPATH . 'wp-settings.php');
dest: "{{ user_website_path }}/wordpress/wp-config.php"
owner: pooja
group: pooja
mode: '0644'
notify: Restart Nginx and PHP-FPM

```

- name: Configure Nginx for WordPress and phpmyadmin

copy:

content: |

```

server {
    server_name {{ domain_name }};
    root {{ user_website_path }}/wordpress;
    index index.php index.html index.htm;

```

```

    # WordPress location configuration (important fastcgi_pass unix:/run/php-fpm/www.sock;)

```

```

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

```

```

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;

```

```

        fastcgi_pass unix:/run/php-fpm/www.sock;
    }
    location ~ /\.ht {
        deny all;
    }
    # phpMyAdmin location configuration
    location /phpmyadmin {
        alias /usr/share/phpMyAdmin/;
        index index.php index.html index.htm;

        location ~ ^/phpmyadmin/(.+\.php)$ {
            alias /usr/share/phpMyAdmin/$1;
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php-fpm/www.sock;
            fastcgi_param SCRIPT_FILENAME $request_filename;
            include fastcgi_params;
        }

        location ~ ^/phpmyadmin/(index\.php)?$ {
            deny all;
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php-fpm/www.sock;
            fastcgi_param SCRIPT_FILENAME $request_filename;
            include fastcgi_params;
        }
    }
}

dest: "/etc/nginx/conf.d/{{ domain_name }}.conf"
owner: root
group: root
mode: '0644'

```

- name: Restart Nginx and PHP-FPM

systemd:

name: "{{ item }}"

state: restarted

loop:

- nginx

- php-fpm

handlers:

- name: Restart Nginx and PHP-FPM

systemd:

name: sshd

state: restarted