DevOpsAnsibleWordpress

Perform WordPress setup on Linux with Apache server, MySQL, and PHP (LAMP stack).

Create Playbooks

Apache install

Create a file apache2.yaml with the contents below:

```
 ---
- hosts: webservers
  become: true
  vars_files:
    - vars/default.yaml

  tasks:
    - name: install apache2
      apt: name=apache2 update_cache=yes state=latest

    - name: enable mod_rewrite
      apache2_module: name=rewrite state=present
      notify: restart apache2

    - name: Create document root
      file:
        path: /var/www/{{ http_host }}
        state: directory
        owner: "www-data"
        group: "www-data"
        mode: "0755"

    - name: Set up Apache virtualhost
      template:
        src: "files/apache.conf.j2"
        dest: "/etc/apache2/sites-available/{{ http_conf }}"
      notify: reload apache2

    - name: Enable new site
      shell: /usr/sbin/a2ensite {{ http_conf }}
      notify: reload apache2

    - name: Disable default Apache site
      shell: /usr/sbin/a2dissite 000-default.conf
      when: disable_default
      notify: reload apache2

  handlers:
    - name: restart apache2
      service:
        name: apache2
        state: restarted

    - name: reload apache2
      service:
        name: apache2
        state: reloaded
```

MySQL install

Create a file mysql.yaml with the contents below:

```
---
- hosts: webservers
  become: true
  tasks:
    - name: Install MySQL
      apt:
        name: "{{ item }}"
        state: present
      with_items:
        - libmysqlclient-dev
        - python-mysqldb
        - mysql-server
        - mysql-client

    - name: Sets the root password
      mysql_user:
        name: root
        password: password

    - name: Remove the test database
      mysql_db:
        name: test
        state: absent
        login_user: root
        login_password: password

    - name: Remove anonymous user accounts
      mysql_user:
        name: ''
        host_all: yes
        state: absent
        login_user: root
        login_password: password

    - name: Create DB for WordPress
      mysql_db:
        name: "wordpress"
        state: present
        login_user: root
        login_password: password

    - name: Create deploy user for mysql
      mysql_user:
        user: "deploy"
        host: "%"
        password: password
        priv: '*.*:ALL,GRANT'
      with_items:
        - 127.0.0.1
        - ::1
        - localhost

    - name: Update mysql root password for all root accounts
      mysql_user:
        name: root
        host: "{{ item }}"
        password: password
      with_items:
        - 127.0.0.1
        - ::1
        - localhost
```

PHP install

Create a file php.yaml with the contents below:

```
---
- hosts: webservers
  become: true
  vars_files:
    - vars/default.yaml

  tasks:
    - name: Install PHP
      apt:
        name: "{{ item }}"
        state: present
      with_items:
       - php
       - php-mysql
       - libapache2-mod-php

    - name: Install PHP Extensions
      apt:
        name: "{{ item }}"
        update_cache: yes
        state: present
      loop: "{{ php_modules }}"

    - name: Set up PHP info page
      template:
        src: "files/info.php.j2"
        dest: "/var/www/{{ http_host }}/info.php"
```

WordPress install

Create a file wordpress.yaml with the contents below:

```
---
- hosts: webservers
  become: true
  vars_files:
    - vars/default.yaml

  tasks:
  - 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: www-data
      group: www-data

  - 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: Set up wp-config
    template:
      src: "files/wp-config.php.j2"
      dest: "/var/www/{{ http_host }}/wordpress/wp-config.php"
```

Expose port

Create a file firewall.yaml with the contents below:

```
---
- hosts: webservers
  become: true
  vars_files:
    - vars/default.yaml

  tasks:
    - name: Install UFW
      apt:
        name: "ufw"
        state: present

    - name: "Allow HTTP on port {{ http_port }}"
      ufw:
        rule: allow
        port: "{{ http_port }}"
        proto: tcp
```

Main playbook

Create a file main.yaml with the contents below:

```
---
- name: Apache Install
  import_playbook: apache2.yaml

- name: MySQL Install
  import_playbook: mysql.yaml

- name: PHP Install
  import_playbook: php.yaml

- name: WordPress Install
  import_playbook: wordpress.yaml

- name: Expose Port
  import_playbook: firewall.yaml
```

Create Config files

Create Vars

Create directory "vars." In the directory, create a file default.yaml with the contents below:

```
---
app_user: "aevans"
http_host: "host"
http_conf: "conf.conf"
http_port: "80"
disable_default: true
php_modules: [ 'php-curl', 'php-gd', 'php-mbstring', 'php-xml', 'php-xmlrpc', 'php-soap', 'php-intl', 'php-zip' ]

mysql_root_password: "password"
mysql_db: "wordpress"
mysql_user: "root"
mysql_password: "password"
```

Create Files

Create directory "files." In the directory, create a file apache.conf.j2 with the contents below:

```
<VirtualHost *:{{ http_port }}>
  ServerAdmin webmaster@localhost
  ServerName {{ http_host }}
  ServerAlias www.{{ http_host }}
  DocumentRoot /var/www/{{ http_host }}/wordpress

  <Directory /var/www/{{ http_host }}>
    Options -Indexes
  </Directory>

  <IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
  </IfModule>

</VirtualHost>
```

In the directory, create a file info.php.j2 with the contents below:

```
<?php
phpinfo();
```

In the directory, create a file wp-config.php.j2 with the contents below:

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', '{{ mysql_db }}' );

/** MySQL database username */
define( 'DB_USER', '{{ mysql_user }}' );

/** MySQL database password */
define( 'DB_PASSWORD', '{{ mysql_password }}' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/** Filesystem access **/
define('FS_METHOD', 'direct');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define( 'AUTH_KEY',         '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'SECURE_AUTH_KEY',  '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'LOGGED_IN_KEY',    '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'NONCE_KEY',        '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'AUTH_SALT',        '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'SECURE_AUTH_SALT', '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'LOGGED_IN_SALT',   '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );
define( 'NONCE_SALT',       '{{ lookup('password', '/dev/null chars=ascii_letters length=64') }}' );

/**#@-*/

/**
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give each
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'wp_';

/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
 * visit the Codex.
 *
 * @link https://codex.wordpress.org/Debugging_in_WordPress
 */
define( 'WP_DEBUG', true );

/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', dirname( __FILE__ ) . '/' );
}

/** Sets up WordPress vars and included files. */
require_once( ABSPATH . 'wp-settings.php' );
```