

Local.yml

```
---
- hosts: localhost
  become: yes

  tasks:
    - name: Check for setup complete flag file
      ansible.builtin.stat:
        path: /var/www/html/.setup_complete
        register: setup_flag # Store the result

    - name: Initialize setup_complete flag (true if missing)
      ansible.builtin.set_fact:
        setup_complete: "{{ setup_flag.stat.exists }}"

    - name: Gather credentials (if setup incomplete)
      block:
        - pause:
            prompt: "Enter your desired MySQL username"
            register: mysql_credentials

        - set_fact:
            mysql_user: "{{ mysql_credentials.user_input }}"

        - pause:
            prompt: "Enter your desired MySQL password"
            register: mysql_password

        - set_fact:
            mysql_password: "{{ mysql_password.user_input }}"

        - pause:
            prompt: "Enter your desired kjakdb username"
            register: anon_user_credentials

        - set_fact:
            anon_user: "{{ anon_user_credentials.user_input }}"

        - pause:
            prompt: "Enter your desired kjakdb user password"
            register: anon_password

        - set_fact:
            anon_password: "{{ anon_password.user_input }}"
      when: not setup_complete

    - name: Install Apache
      ansible.builtin.apt:
        name: apache2
        state: present
```

Local.yml

```
- name: Install PHP and Apache module
  ansible.builtin.apt:
    name:
      - php
      - libapache2-mod-php
    state: present

- name: Install PHP MySQL extension
  ansible.builtin.apt:
    name: php-mysql
    state: present

- name: Restart Apache (to load PHP module)
  ansible.builtin.service:
    name: apache2
    state: restarted

- name: Create uploads folder
  ansible.builtin.file:
    path: /var/www/html/uploads
    state: directory
    owner: root
    group: root
    mode: '0777'

- name: Check if index.html exists
  ansible.builtin.stat:
    path: /var/www/html/index.html
  register: index_file_stats

- name: Rename index.html to index.html.old (if it exists)
  ansible.builtin.file:
    src: /var/www/html/index.html
    dest: /var/www/html/index.html.old
    state: file
  when: index_file_stats.stat.exists

- name: Copy web files
  ansible.builtin.copy:
    src: "www/{{ item }}"
    dest: /var/www/html/{{ item }}
  with_items:
    - index.php
    - favicon.ico
```

Local.yml

```
- style.css
- script.js
- create_thread.php
- view_forum.php
- view_thread.php

- name: Create conn.php from template
  ansible.builtin.template:
    src: www/conn.php.j2
    dest: /var/www/html/conn.php
  when: not setup_complete

- name: Install MariaDB server
  ansible.builtin.apt:
    name: mariadb-server
    state: present

- name: Start and enable MariaDB service
  ansible.builtin.service:
    name: mariadb
    state: started
    enabled: true

- name: Ensure pip is installed
  ansible.builtin.apt:
    name: python3-pip
    state: present

- name: Install PyMySQL
  ansible.builtin.pip:
    name: PyMySQL
    state: present

- name: Set MySQL root password
  ansible.builtin.shell: |
    mysql -u root -e "SET PASSWORD FOR root@'localhost' = PASSWORD('{{
mysql_password }}');"
  args:
    executable: /bin/bash
  when: not setup_complete

- name: Set password for MySQL user (including root)
  ansible.builtin.mysql_user:
    login_user: root
    login_password: "{{ mysql_password }}"
```

Local.yml

```
    check_implicit_admin: true
#    login_unix_socket: /var/lib/mysql/mysql.sock
    name: "{{ mysql_user }}"
    password: "{{ mysql_password }}"
    priv: " *.*:ALL "
    host: localhost
when: not setup_complete

- name: Create .my.cnf for MySQL authentication
  ansible.builtin.copy:
    dest: "~/.my.cnf"
    content: |
      [client]
      user={{ mysql_user }}
      password={{ mysql_password }}
    mode: '0600'
  when: not setup_complete

- name: Ensure the database 'kjakdb' exists
  ansible.builtin.mysql_db:
    name: kjakdb
    state: present
    login_user: "{{ mysql_user }}"
    login_password: "{{ mysql_password }}"
  when: not setup_complete

- name: Ensure the kjakdb user exists with privileges
  ansible.builtin.mysql_user:
    login_user: root
    login_password: "{{ mysql_password }}"
    name: "{{ mysql_user }}"
    password: "{{ mysql_password }}"
    priv: 'kjakdb.*:ALL'
    host: localhost
    state: present
  when: not setup_complete

- name: Create update script from template
  ansible.builtin.template:
    src: sh/update_kjakdb.sh.j2
    dest: /tmp/update_kjakdb.sh
    mode: '0755'
  when: not setup_complete

- name: Execute the update script
```

Local.yml

```
ansible.builtin.command:
  cmd: /tmp/update_kjakdb.sh
when: not setup_complete

- name: Create state file after initial setup
  ansible.builtin.file:
    path: /var/www/html/.setup_complete
    state: touch
    mode: '0644'
  when: not setup_complete

- name: Set initial setup flag
  ansible.builtin.set_fact:
    setup_complete: true
```