Deploying database and application servers in coordinated manner using Ansible.

Database setup

Application server connect after DB is ready

Coordinate all things using ansible, inventory, handlers, roles, playbook.

Created folder named db-app-deploy

Move to that folder and create inventory.yml and site.yml

**Inventory.yml**

all:

  hosts:

    common-ec2:

      ansible\_host: 3.80.33.201

      ansible\_user: ubuntu

      ansible\_ssh\_private\_key\_file: ~/.ssh/testsonam.pem

  children:

    db:

      hosts:

        common-ec2:

    app:

      hosts:

        common-ec2:

**site.yml**

---

- name: Deploy Database Server

  hosts: db

  become: true

  roles:

    - db

- name: wait for DB Connect

  hosts: db

  become: true

  tasks:

    - name: wait for postgres to listen on the port 5432

      wait\_for:

        port: 5432

        host: "localhost"

        timeout: 60

- name: Deploy Application Server

  hosts: app

  become: true

  roles:

    - app

Create 2 roles under roles folder

Ansible-galaxy init roles/db

Ansible-galaxy init roles/app

Under roles db **edit tasks/main.yml**

---

# tasks file for roles/db

- name: Install PostgreSQL

  apt:

    name: postgresql

    state: present

    update\_cache: yes

- name: Ensure PostgreSQL is Running

  service:

    name: postgresql

    state: started

    enabled: true

- name: create application database

  become: true

  shell: |

    sudo -u postgres psql -tc "SELECT 1 FROM pg\_database where datname = 'myapp' " | grep -q 1 ||

    sudo -u postgres psql -c "CREATE DATABASE myapp;"

  args:

    executable: /bin/bash

- name: create PostgreSQL User with password

  become: true

  shell: |

    sudo -u postgres psql -tc "SELECT 1 FROM pg\_roles WHERE rolname = '{{ DB\_USER }}' " | grep -q 1 ||

    sudo -u postgres psql -c "CREATE USER {{ DB\_USER }} WITH PASSWORD '{{ DB\_PASS }}';"

  args:

    executable: /bin/bash

Defaults/main.yml

---

# defaults file for roles/db

DB\_USER: myappuser

DB\_PASS: mysecretpassword

Under role app

Add code under files /app/app.py

from flask import Flask,jsonify

import os

import psycopg2

from psycopg2 import OperationalError

from dotenv import load\_dotenv

load\_dotenv(dotenv\_path='/home/ubuntu/app/.env')

app = Flask(\_\_name\_\_)

def get\_db\_connection():

    try:

        conn = psycopg2.connect(

            host=os.getenv('DB\_HOST'),

            database=os.getenv('DB\_NAME'),

            user=os.getenv('DB\_USER'),

            password=os.getenv('DB\_PASS'),

        )

        return conn

    except OperationalError as e:

        print(f"Database connectivity failed: {e}")

        return None

@app.route('/')

def hello():

    conn = get\_db\_connection()

    if conn:

        cur = conn.cursor()

        cur.execute("SELECT version();")

        db\_version = cur.fetchone()

        cur.close()

        return jsonify({

            'status':"Connected to DB",

            'DB\_Version': db\_version[0]

        })

    else:

        return jsonify({

            'status':'falied to connect with DB'

        }),500

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(host='0.0.0.0',port=5000)

Also add **tasks/main.yml**

---

# tasks file for roles/app

- name: Install Dependencies

  apt:

    name: python3-pip

    state: present

- name: Install Flask and psycopg2 and dotenv

  apt:

    name:

      - python3-flask

      - python3-psycopg2

      - python3-dotenv

- name: copy application code

  copy:

    src: files/app/

    dest: /home/ubuntu/app/

    mode: '0755'

- name: Set Environment Variables

  copy:

    dest: /home/ubuntu/app/.env

    content: |

      DB\_HOST=localhost

      DB\_USER=myappuser

      DB\_PASS=mysecretpassword

      DB\_NAME=myapp

- name: Start Flask app

  shell: |

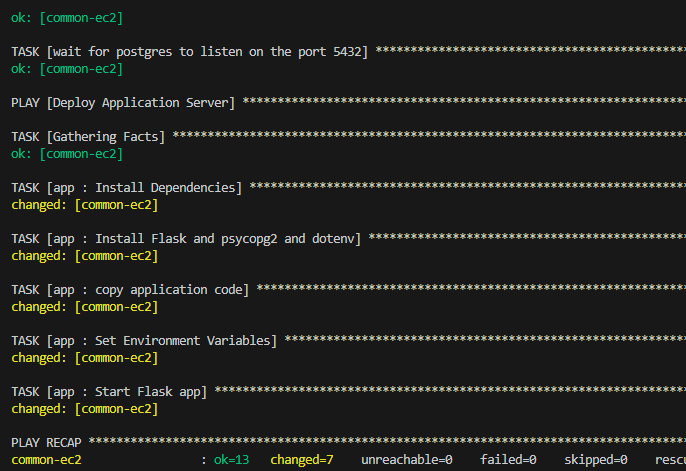
    FLASK\_APP=/home/ubuntu/app/app.py flask run --host=0.0.0.0 --port=5000 &

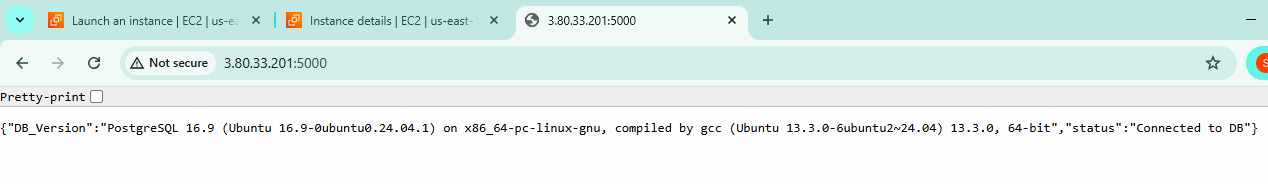
  args:

    executable: /bin/bash

Run playbook:

**ansible-playbook -i inventory.yml site.yml**





Trigger <http://3.80.33.201:5000/>

In your case it must be your public ip:5000