Ansible Docker Container Provisioning
https://github.com/usman-akram-dev/

1. Overview Of The Project

Problem Statement:

Develop a project that uses Ansible to provision 3 containers using Docker and configure the first container with PostgreSQL, second container with Apache and third container with Git.

- **For First Container:** Using Ansible, CREATE a database in PostgreSQL with name DevOps and table “students” and INSERT the list of students enrolled in DevOps class along with their First Name, Last Name and Registration Number.
- **For Second Container:** Using Ansible, ping the URL for Apache and get a 200 response.
- **For Third Container:** Using Ansible, make a file with your name and add some text to it. Push the change and the change should be visible on BitBucket.

Objective:

We are developing this project to demonstrate the power and capabilities of Ansible in provisioning and configuring containers using Docker. Ansible is a powerful automation tool that allows us to define infrastructure as code and easily manage and deploy applications.

By using Ansible to provision three containers with specific configurations, we can showcase how Ansible can streamline the setup process and ensure consistency across different environments.

The objective of this project is to provision three containers using Ansible and Docker, and configure them as follows:

PostgreSQL Container:

- Create a PostgreSQL database named "DevOps".
- Create a table named "students" in the "DevOps" database.
- Insert the list of students enrolled in the DevOps class, including their first name, last name, and registration number.

Apache Container:

- Ping the URL for Apache and verify a 200 response code.

Git Container:

- Create a file with your name.
- Add some text to the file.
- Push the change to a BitBucket repository.
- Verify that the change is visible on BitBucket.
- To accomplish these tasks, you will use Ansible to automate the provisioning and configuration of the containers. Ansible provides a declarative language to describe the desired state of systems and allows for easy automation and orchestration.

The project involves writing Ansible playbooks to perform the following actions:

Provisioning Playbook:

Use Docker modules in Ansible to create three containers: one for PostgreSQL, one for Apache, and one for Git.

PostgreSQL Playbook:

- Use Ansible's PostgreSQL module to create the "DevOps" database.
- Use the PostgreSQL module to create a "students" table in the "DevOps" database.
- Use Ansible's PostgreSQL module to insert the list of students into the "students" table.

Apache Playbook:

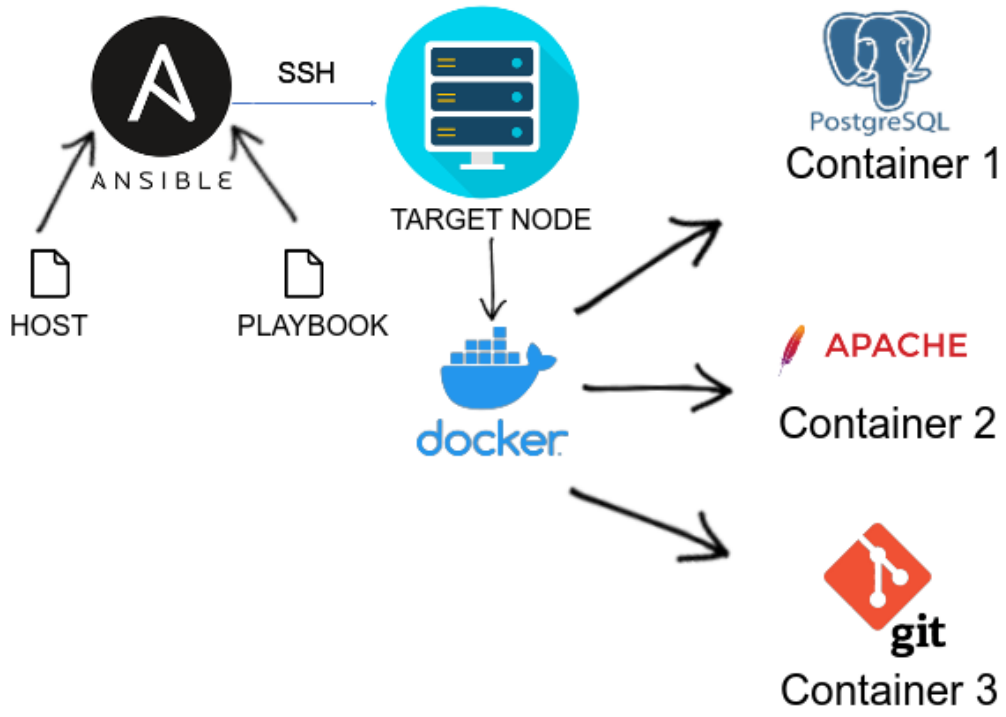
- Use Ansible's URI module to ping the Apache URL and check for a 200 response code.

Git Playbook:

- Use Ansible's Git module to clone a BitBucket repository.
- Create a file with your name using Ansible's File module.
- Add text to the file.
- Use Ansible's Git module to commit and push the change to the BitBucket repository.

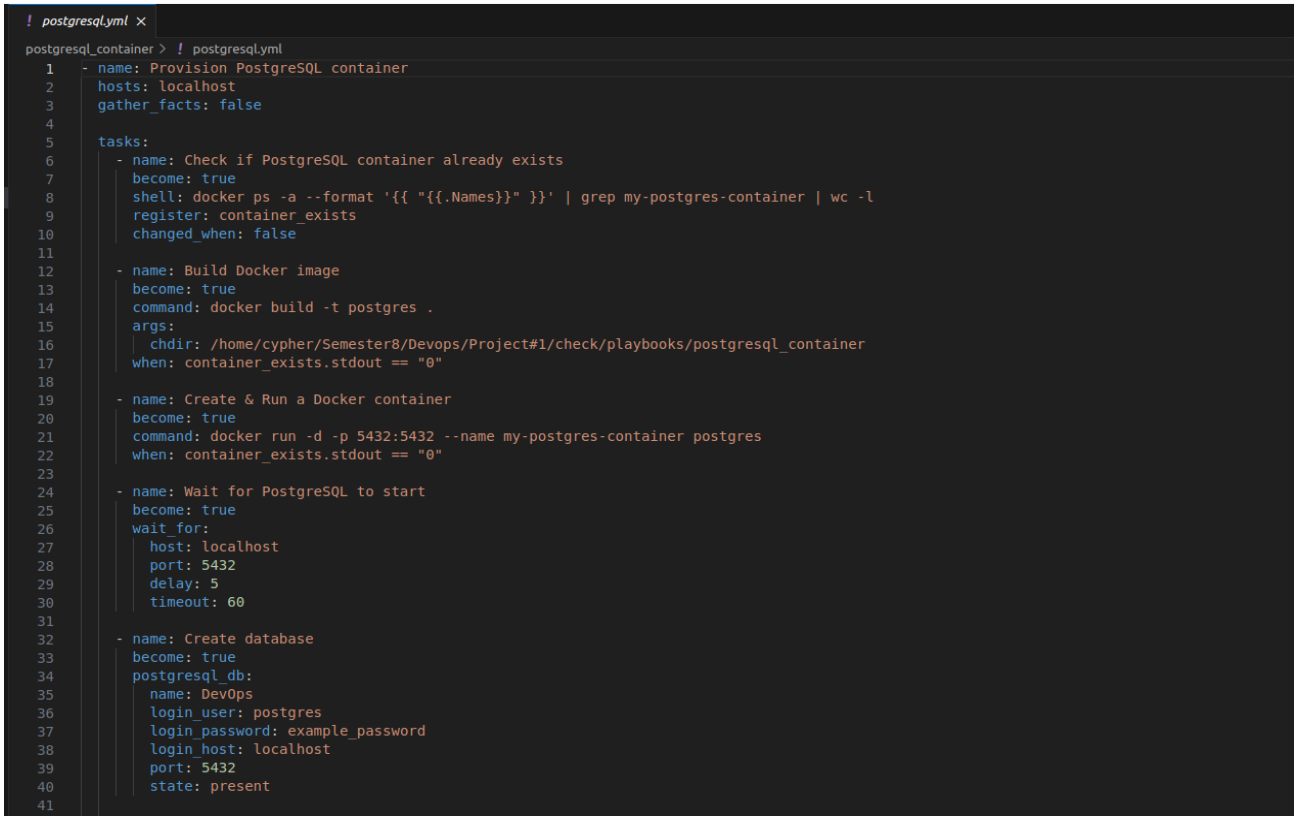
- By executing these playbooks, you will automate the provisioning of containers, setup the PostgreSQL database and table, check the Apache URL, and perform Git operations to create and push changes. This project demonstrates the power of Ansible in automating infrastructure and configuration management tasks.

2. Project Architecture



3. Screenshots of Project component/UI

Container 1 : (Ansible Playbook For PostgreSQL)



postgresql.yml

postgresql_container > postgresql.yml

```yaml
- name: Provision PostgreSQL container
  hosts: localhost
  gather_facts: false

  tasks:
    - name: Check if PostgreSQL container already exists
      become: true
      shell: docker ps -a --format '{{ "{{.Names}}" }}' | grep my-postgres-container | wc -l
      register: container_exists
      changed_when: false

    - name: Build Docker image
      become: true
      command: docker build -t postgres .
      args:
        chdir: /home/cypher/Semester8/Devops/Project#1/check/playbooks/postgresql_container
      when: container_exists.stdout == "0"

    - name: Create & Run a Docker container
      become: true
      command: docker run -d -p 5432:5432 --name my-postgres-container postgres
      when: container_exists.stdout == "0"

    - name: Wait for PostgreSQL to start
      become: true
      wait_for:
        host: localhost
        port: 5432
        delay: 5
        timeout: 60

    - name: Create database
      become: true
      postgresql_db:
        name: DevOps
        login_user: postgres
        login_password: example_password
        login_host: localhost
        port: 5432
        state: present

```

postgresql.yml

postgresql_container > postgresql.yml

```yaml
      login_host: localhost
      port: 5432
      state: present

  - name: Check if students table exists
    become: true
    postgresql_query:
      db: DevOps
      login_user: postgres
      login_password: example_password
      login_host: localhost
      port: 5432
      query: "SELECT EXISTS (SELECT 1 FROM information_schema.tables WHERE table_name = 'students');"
    register: table_exists
    changed_when: false

  - name: Create students table if it doesn't exist
    become: true
    postgresql_query:
      db: DevOps
      login_user: postgres
      login_password: example_password
      login_host: localhost
      port: 5432
      query: |
        CREATE TABLE students (
          id SERIAL PRIMARY KEY,
          first_name TEXT,
          last_name TEXT,
          reg_number TEXT
        );
    when: table_exists.query_result == [] or not table_exists.query_result[0].exists

  - name: Insert students into table
    become: true
    postgresql_query:
      db: DevOps
      login_user: postgres
      login_password: example_password
      login_host: localhost
      port: 5432
      query: "INSERT INTO students (first_name, last_name, reg_number) VALUES ('Muhammad Usman', 'Akram', 'cs191078'),
```

Open | Dockerfile | ~/Semester8/Devops/DevOps-Project-1/ansible/playbooks/postgresql_container | Save

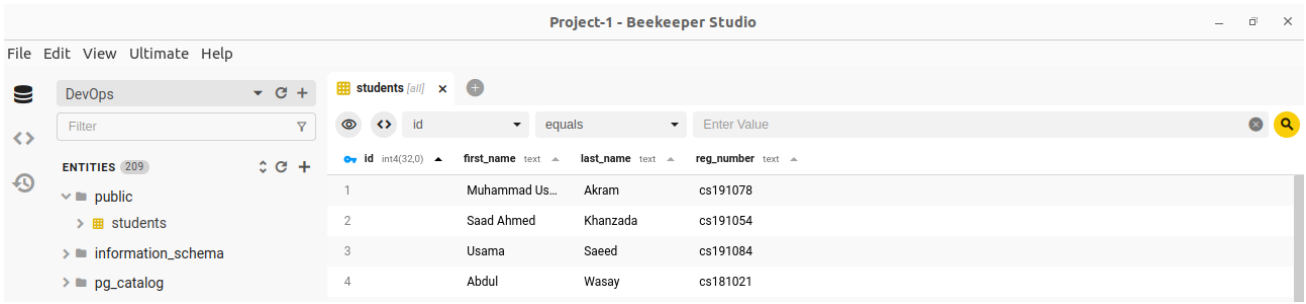
```
FROM postgres:latest

ENV POSTGRES_USER=postgres
ENV POSTGRES_PASSWORD=example_password
ENV POSTGRES_DB=DevOps

```

Open | inventory.ini | ~/Semester8/Devops/DevOps-Project-1/ansible/playbooks/postgresql_container | Save

```
[localhost]
127.0.0.1 ansible_connection=local

```

cypher@cypher-workspace: ~

```
cypher@cypher-workspace:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED       STATUS       PORTS                                       NAMES
87c487d51bea   apache     "httpd-foreground"       3 hours ago   Up 3 hours   0.0.0.0:80->80/tcp, :::80->80/tcp           my-apache-container-instance
e3d2d0dd4b26   git        "/bin/sh"                3 hours ago   Up 3 hours                                               my-git-container
57884b33b9ca   postgres   "docker-entrypoint.s…"   4 hours ago   Up 4 hours   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   my-postgres-container
cypher@cypher-workspace:~$
```



Container 2 : (Ansible Playbook For Apache)

apache.yml

apache_container > ! apache.yml

```yaml
- name: Provision Apache container
  hosts: localhost
  gather_facts: false

  tasks:
    - name: Check if Apache container already exists
      become: true
      shell: docker ps -a --format '{{ "{{.Names}}" }}' | grep my-apache-container | wc -l
      register: container_exists
      changed_when: false

    - name: Build Docker image
      become: true
      command: docker build -t apache .
      args:
        chdir: /home/cypher/Semester8/Devops/Project#1/check/playbooks/apache_container
      when: container_exists.stdout == "0"

    - name: Run Docker container
      become: true
      command: docker run -d -p 80:80 --name my-apache-container-instance apache
      when: container_exists.stdout == "0"

    - name: Ping Apache
      command: curl -I 127.0.0.1
      register: curl_output

    - name: Apache Response
      debug:
        var: curl_output.stdout_lines[0]
```

Open | Dockerfile | ~/Semester8/Devops/DevOps-Project-1/ansible/playbooks/apache_container | Save

```
FROM httpd:latest

EXPOSE 80

```

Open | inventory.ini | ~/Semester8/Devops/DevOps-Project-1/ansible/playbooks/apache_container | Save

```
[localhost]
127.0.0.1 ansible_connection=local

```

cypher@cypher-workspace: ~

```
cypher@cypher-workspace:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED       STATUS       PORTS                                   NAMES
87c487d51bea   apache     "httpd-foreground"       3 hours ago   Up 3 hours   0.0.0.0:80->80/tcp, :::80->80/tcp       my-apache-container-instance
e3d2d0dd4b26   git        "/bin/sh"                3 hours ago   Up 3 hours                                           my-git-container
57884b33b9ca   postgres   "docker-entrypoint.s…"   4 hours ago   Up 4 hours   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   my-postgres-container
cypher@cypher-workspace:~$
```

```
cypher@cypher-workspace: ~
cypher@cypher-workspace:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                                       NAMES
87c487d51bea   apache     "httpd-foreground"       3 hours ago    Up 3 hours    0.0.0.0:80->80/tcp, :::80->80/tcp           my-apache-container-instance
e3d2d0dd4b26   git        "/bin/sh"                3 hours ago    Up 3 hours                                                my-git-container
57884b33b9ca   postgres   "docker-entrypoint.s…"   4 hours ago    Up 4 hours    0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   my-postgres-container
cypher@cypher-workspace:~$
```

```
TASK [Apache Response] ***********************************************************
ok: [localhost] => {
    "curl_output.stdout_lines[0]": "HTTP/1.1 200 OK"
}
```

Container 3 : (Ansible Playbook For GitHub)

git.yml

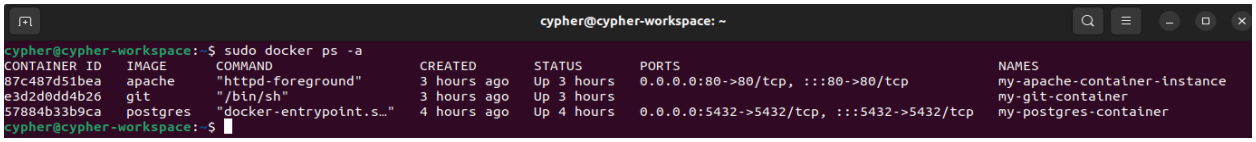
```yaml
- name: Create and start a Docker container with Git
  hosts: localhost
  become: true
  tasks:
    - name: Check if Git container already exists
      become: true
      shell: docker ps -a --format '{{ "{{.Names}}" }}' | grep my-git-container | wc -l
      register: container_exists
      changed_when: false

    - name: Build the Docker image
      command: docker build -t git .
      args:
        chdir: /home/cypher/Semester8/Devops/Project#1/check/playbooks/git_container
      when: container_exists.stdout == "0"

    - name: Create & Run a Docker container
      command: docker run -itd --name my-git-container git /bin/sh
      register: container_output
      when: container_exists.stdout == "0"

    - name: Check Git version
      command: docker exec my-git-container git --version
      register: git_version_output

    - name: Print Git version
      debug:
        var: git_version_output.stdout

    - name: Push to Git
      block:
        - name: Clone repository
          shell: |
            git clone https://x-token-auth:ATCTT3xFfGN0-SRVP3mPhH9mmGA_KLUmHnHZ9suzTupBMJI_biaB2S5wyOYL22r2KKaUUgUKEf4TemwzpMezdl2fogYkfWaKpnTZBDqILULgaHvBOz1BBhkZLY3
            sudo chmod -R 777 project-1
            cd project-1
            git config user.email 712020:2c98e1a7-4794-4d26-a504-8906dad72c4e@bots.bitbucket.org
            echo "DevOps Project 1 Demo" >> cs191078-usman.txt
            git fetch
            git pull
            git add cs191078-usman.txt
            git commit -m "add file with my name"
            git push origin master
```

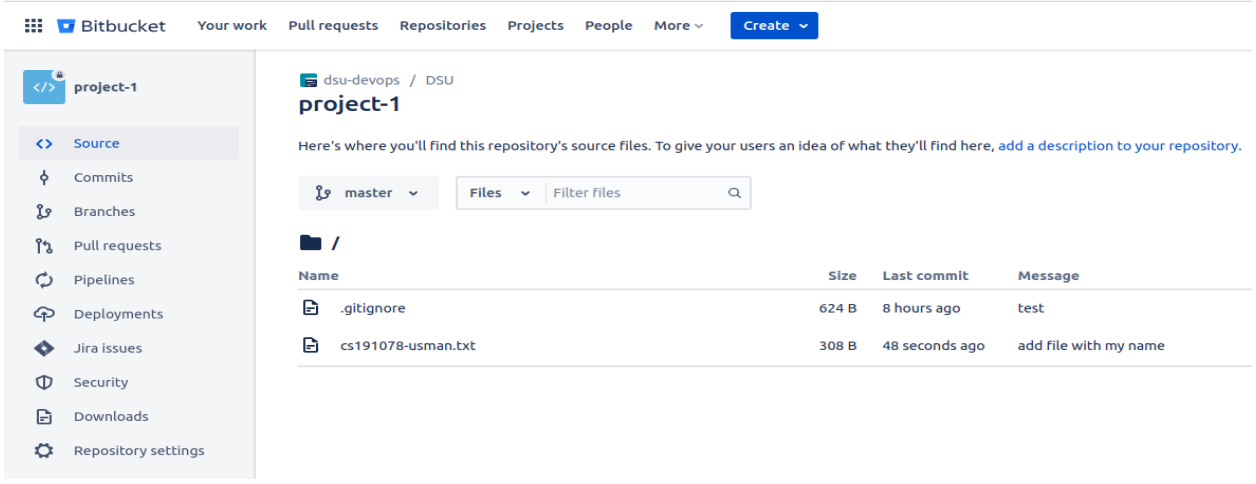
Open | *Dockerfile | ~/Semester8/Devops/DevOps-Project-1/ansible/playbooks/git_container | Save

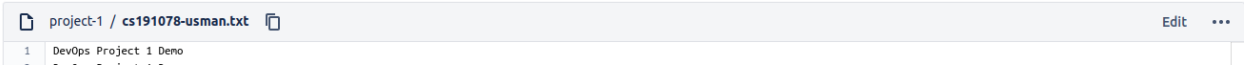
```
FROM alpine:3.5
RUN apk update
RUN apk add git
```

Open | inventory.ini | ~/Semester8/Devops/DevOps-Project-1/ansible/playbooks/git_container | Save

```
[localhost]
127.0.0.1 ansible_connection=local
```


cypher@cypher-workspace: ~
cypher@cypher-workspace:~$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
87c487d51bea apache "httpd-foreground" 3 hours ago Up 3 hours 0.0.0.0:80->80/tcp, :::80->80/tcp my-apache-container-instance
e3d2d0dd4b26 git "/bin/sh" 3 hours ago Up 3 hours my-git-container
57884b33b9ca postgres "docker-entrypoint.s…" 4 hours ago Up 4 hours 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp my-postgres-container
cypher@cypher-workspace:~$


Bitbucket
Your work
Pull requests
Repositories
Projects
People
More
Create
project-1
Source
Commits
Branches
Pull requests
Pipelines
Deployments
Jira issues
Security
Downloads
Repository settings
dsu-devops / DSU
project-1
Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, add a description to your repository.
master
Files
Filter files
/
Name
Size
Last commit
Message
.gitignore
624 B
8 hours ago
test
cs191078-usman.txt
308 B
48 seconds ago
add file with my name


project-1 / cs191078-usman.txt
Edit
1
DevOps Project 1 Demo