**CI/CD Pipeline Setup Using GitHub Actions**

**Prerequisites**

1. **GitHub Repository**: Ensure you have a GitHub repository for your project.
2. **AWS CLI**: Install and configure the AWS CLI with the necessary permissions to deploy resources.
3. **Docker**: Familiarity with Docker and Docker Compose.

**Step 1: Create Environment Variables in GitHub**

To store sensitive data such as database credentials, use GitHub Secrets:

1. Go to your GitHub repository.
2. Click on **Settings** > **Secrets and variables** > **Actions** > **New repository secret**.
3. Create the following secrets:
   * AWS\_ACCESS\_KEY\_ID : 21891607547
   * AWS\_SECRET\_ACCESS\_KEY : IKFAQHCKI5GUVL43LLAFO6ZMT4
   * DB\_USER : admin
   * DB\_PASSWORD : password123
   * DB\_NAME : MYSQL
   * RDS\_ENDPOINT : dbsigma.cdsciuum4t92.us-west-2.rds.amazonaws.com

**Step 2: Create GitHub Actions Workflow**

1. In your repository, create a directory. github/workflows.
2. Create a file named ci-cd-pipeline.yml inside this directory.

**Explanation of the Workflow**

* **Triggers**: The workflow is triggered on pushes to the main branch.
* **Jobs**:
  + **Build Job**:
    - Checks out the code.
    - Sets up Docker Buildx for building multi-platform images.
    - Logs into Amazon ECR (if you’re using ECR for storage, else you can skip this step).
    - Builds Docker images for WordPress and MySQL.
    - Runs basic tests using curl to check if the WordPress home page is accessible.
  + **Deploy Job**:
    - Waits for the build job to finish.
    - Checks out the code again.
    - Deploys the Docker containers to the AWS EC2 instance using SSH.

**Adjust the Dockerfile and Docker Compose**

Ensure you have Dockerfiles for both WordPress and MySQL in their respective directories (e.g., ./wordpress/Dockerfile and ./mysql/Dockerfile). Your docker-compose.yml file should be in a directory accessible by your deployment script.

**Finalize and Test**

* Push changes to your GitHub repository.
* Check the **Actions** tab to monitor the pipeline execution.
* Ensure that the Docker containers are deployed correctly and the WordPress site is accessible.