**Requirements**

1. **Create a Web App:** A simple web app should be created or leverage an existing one, such as those available on GitHub.
2. **Dockerize the App**:

Dockerize the web app by creating a Dockerfile with instructions to build a Docker image for the app.

1. **Create a DockerHub Account:**

If not already done, create a DockerHub account to host the Docker image.

1. **Build and Push Docker Image:**

Build the Docker image in both x86\_64 and arm64 formats using Docker Buildx tool, and push the images to DockerHub registry.

1. **Deploy the App on AWS:**

Deploy the app container on AWS by creating and launching an EC2 instance to serve as the container server.

1. **Install Docker Engine:**

Download and install Docker engine on the EC2 server.

1. **Launch Container:**

Launch a container of the web app on the EC2 server and ensure it is accessible via the public internet.

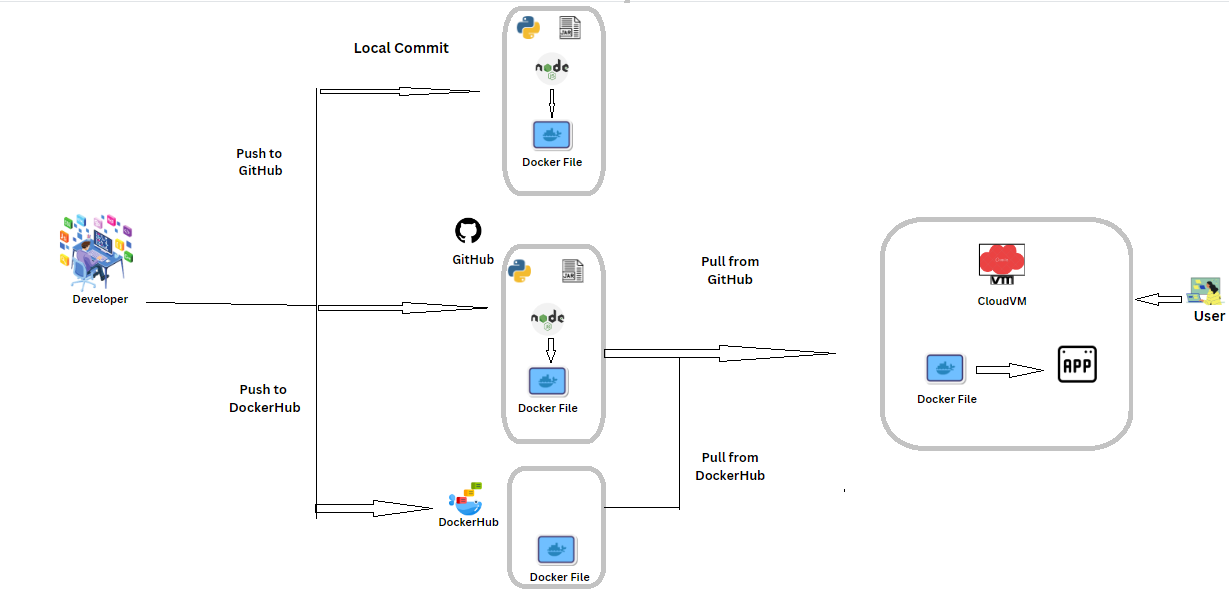
**Documentation**

**GitHub Repository**

URL: Your Public GitHub Repo URL

Architectural Diagrams

Below are the architectural diagrams depicting the setup:



**Instructions for Accessing the Web App**

1. Clone the repository from GitHub: git clone https://github.com/yourusername/yourrepo.git
2. Navigate to the cloned directory: cd yourrepo
3. Build the Docker image(Bash codes)

“docker build -t yourusername/webapp:latest .”

1. Push the Docker image to DockerHub:

“docker push yourusername/webapp:latest”

1. Launch an EC2 instance on AWS and install Docker engine.
2. Pull the Docker image from DockerHub on the EC2 instance:

“docker pull yourusername/webapp:latest”

Run the Docker container:

“docker run -d -p 80:80 yourusername/webapp:latest”

Access the web app via the public IP address of your EC2 instance.

Bonus

To optimize the Docker image, consider the following:

Optimize cache steps in the Dockerfile.

Use smaller base images like Alpine or slim versions of Ubuntu.

Utilize multi-build stages to minimize image size.

For high availability and scalability, deploy app containers in EC2 instances as part of an autoscaling group and add an application load balancer in front.

Alternatively, you can use AWS ECS to host your containerized web app instead of a traditional EC2 server with Docker engine installed.