# **INSTALLATION AND OPERATIONS GUIDE FOR DEPLOY FUSION**

## **Overview**

This guide details how to set up and manage the CI/CD pipeline for your web application, **Deploy Fusion**, using Vercel and GitHub Actions. This setup allows for automatic testing and deployment whenever changes are pushed to your GitHub repository.

### **Prerequisites**

* GitHub account
* Vercel account
* Basic knowledge of Git and GitHub
* Familiarity with command-line operations

## **Step 1: Setting Up Your GitHub Repository**

1. **Create a Repository:**
   * Go to GitHub and create a new repository named deploy-fusion.
   * Initialize it with a README if desired.

**Clone the Repository:**  
git clone https://github.com/umarcode01/deploy-fusion.git

cd deploy-fusion

1. **Add Your Application Code:**
   * Add your application files to this directory.

**Commit and Push Changes:**  
git add .

git commit -m "Adding files to Deploy Fusion"

git push origin master

## **Step 2: Deploying with Vercel**

1. **Create a New Project in Vercel:**
   * Log in to your Vercel account.
   * Click on "New Project" and import your deploy-fusion GitHub repository.
2. **Configure Project Settings:**
   * Vercel will detect your framework automatically.
3. **Environment Variables:**
   * Go to the "Settings" tab of your Vercel project.
   * Add any required environment variables here.
4. **Deploy Your Application:**
   * Click "Deploy" to launch Deploy Fusion.
   * Vercel will provide a unique URL for your application.

## **Step 3: Setting Up GitHub Actions**

1. **Create a GitHub Actions Workflow:**
   * Inside your repository, create a directory called .github/workflows.
   * Create a new YAML file named ci-cd.yml.
2. **Configure the Workflow:**

name: Vercel Production Deployment

env:

VERCEL\_ORG\_ID: ${{ secrets.VERCEL\_ORG\_ID }}

VERCEL\_PROJECT\_ID: ${{ secrets.VERCEL\_PROJECT\_ID }}

on:

push:

branches:

- master

jobs:

Deploy-Production:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Install Vercel CLI

run: npm install --global vercel@latest

- name: Pull Vercel Environment Information

run: vercel pull --yes --environment=production --token=${{ secrets.VERCEL\_TOKEN }}

- name: Build Project Artifacts

run: vercel build --prod --token=${{ secrets.VERCEL\_TOKEN }}

- name: Deploy Project Artifacts to Vercel

run: vercel deploy --prebuilt --prod --token=${{ secrets.VERCEL\_TOKEN }}

**3. Make sure to do the following:**

1. Retrieve your Vercel Access Token
2. Install the Vercel CLI and run vercel login
3. Inside your folder, run vercel link to create a new Vercel project
4. Inside the generated .vercel folder, save the projectId and orgId from the project.json
5. Inside GitHub, add VERCEL\_TOKEN, VERCEL\_ORG\_ID, and VERCEL\_PROJECT\_ID as secrets

## **Step 4: Running the Pipeline**

1. **Make Changes and Push:**

* Edit your code, then commit and push your changes:  
  git add .

git commit -m "Updated Deploy Fusion"

git push origin main

**Monitor GitHub Actions:**

* Navigate to the "Actions" tab in GitHub repository.
* We can track the progress of our CI/CD pipeline here.
* Upon successful completion, Deploy Fusion will be automatically deployed to Vercel.