Flask Deployment & YAML Guide with GitHub Actions and Render

Complete CI/CD Setup with Full Workflow and Key Explanations

Author: Anupam Anand

Date: 06 April 2025

# 1. Full GitHub Actions Workflow (deploy.yml)

name: Deploy to Render  
  
on:  
 push:  
 branches:  
 - main  
  
jobs:  
 deploy:  
 runs-on: ubuntu-latest  
  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v3  
  
 - name: Set up Python  
 uses: actions/setup-python@v4  
 with:  
 python-version: '3.10'  
  
 - name: Install dependencies  
 run: |  
 python -m venv venv  
 source venv/bin/activate  
 pip install --upgrade pip  
 pip install -r requirements.txt  
  
 - name: Deploy to Render  
 env:  
 RENDER\_API\_KEY: ${{ secrets.RENDER\_API\_KEY }}  
 run: |  
 curl -X POST https://api.render.com/deploy/srv-<YOUR\_SERVICE\_ID>?key=$RENDER\_API\_KEY

# 2. GitHub Actions Workflow YAML - Key Breakdown

on / push / branches

- `on`: Specifies the event that triggers the workflow.  
 - Common values: `push`, `pull\_request`, `workflow\_dispatch`  
- `push`: Triggers on code push  
- `branches`: Filter branches to run the workflow (e.g. `main`, `feature/\*`)

jobs / deploy

- `jobs`: Defines one or more jobs that run in the workflow  
- `deploy`: Job ID used to reference that job

runs-on

- `runs-on`: Virtual machine image to run the job  
 - Common values: `ubuntu-latest`, `windows-latest`, `macos-latest`

steps / uses / name

- `steps`: A list of tasks in the job  
- `name`: Label for each step  
- `uses`: Reuses a GitHub Action (e.g., checkout code, setup python)

with / python-version

- `with`: Passes input parameters to the action  
- `python-version`: Specifies which Python version to install

run

- `run`: Shell command(s) executed by the runner  
- `|`: Indicates multi-line script

env / secrets

- `env`: Sets environment variables  
- `${{ secrets.RENDER\_API\_KEY }}`: Securely retrieves a GitHub secret

# 3. .render.yaml - Full Example and Explanation

services:  
 - type: web  
 name: my-flask-app  
 env: docker  
 plan: free  
 branch: main  
 autoDeploy: false

- `services`: Defines services to deploy (web, background, cron)  
- `type`: Type of service (`web` = public facing)  
- `name`: Display name on the dashboard  
- `env`: Runtime environment (`python`, `docker`)  
- `plan`: Billing tier (`free`, `starter`, etc.)  
- `branch`: Git branch to monitor  
- `autoDeploy`: Automatically deploys on push (set to false if using CI/CD)

# 4. Dockerfile - Full Example and Explanation

FROM python:3.10-slim  
  
ENV PYTHONDONTWRITEBYTECODE=1  
ENV PYTHONUNBUFFERED=1  
  
WORKDIR /app  
  
COPY requirements.txt .  
RUN pip install --upgrade pip && pip install -r requirements.txt  
  
COPY . .  
  
CMD ["gunicorn", "app:app", "--bind", "0.0.0.0:8000"]

- `FROM`: Base image (lightweight Python)  
- `ENV`: Environment variables for performance/logging  
- `WORKDIR`: Sets working directory in container  
- `COPY`: Moves files from host to container  
- `RUN`: Executes commands during image build  
- `CMD`: Command to start app (gunicorn for Flask)