

GitHub Actions CI/CD Pipeline Flask App

Objective:

Implement a CI/CD workflow using GitHub Actions for a Python application.

Setup

1. Github repository link <https://github.com/akshaybhu/GithubAction-Python>
2. Create 2 EC2 instances, 1 for prod and 1 for staging.

> Staging instance

The screenshot shows the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for various AWS services. The main content area displays the 'Instance summary for i-Ofe7636b450ac018c (Aks-Github-staging)'. The instance is in the 'Running' state. The console shows details such as the Instance ID, Public IPv4 address (23.22.221.164), Private IP address (10.0.0.157), Hostname type, Answer private resource DNS name, Auto-assigned IP address (23.22.221.164), IAM Role, IMDSv2 (Required), Operator, Subnet ID, Instance type (t2.micro), VPC ID, Instance ARN, Elastic IP addresses, AWS Compute Optimizer finding, Auto Scaling Group name, Managed, Platform details (Linux/UNIX), and Termination protection (Disabled).

> Prod instance below:

EC2 > Instances > i-0c94bb373a5e23df3

Dashboard
EC2 Global View
Events

▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

▼ Images
AMIs
AMI Catalog

▼ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager

▼ Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces

▼ Load Balancing
Load Balancers
Target Groups

Instance summary for i-0c94bb373a5e23df3 (Aks-Github-prod) Info

Updated less than a minute ago

Instance ID
i-0c94bb373a5e23df3

IPv6 address
-

Hostname type
IP name: ip-10-0-0-234.ec2.internal

Answer private resource DNS name
-

Auto-assigned IP address
54.80.228.61 [Public IP]

IAM Role
-

IMDSv2
Required

Operator
-

Public IPv4 address
54.80.228.61 | [open address](#)

Instance state
Running

Private IP DNS name (IPv4 only)
ip-10-0-0-234.ec2.internal

Instance type
t2.micro

VPC ID
vpc-09f02049d6176fe30 (dev_stage_vpc)

Subnet ID
subnet-01874c4512136bd62 (az-1)

Instance ARN
arn:aws:ec2:us-east-1:975050024946:instance/i-0c94bb373a5e23df3

Private IPv4 addresses
10.0.0.234

Public IPv4 DNS
-

Elastic IP addresses
-

AWS Compute Optimizer finding
User: arn:aws:iam::975050024946:user/akshaythebest@yahoo.co.in is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action
[Retry](#)

Auto Scaling Group name
-

Managed
false

[Connect](#) [Instance state](#) [Actions](#)

[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

▼ **Instance details** Info

AMI ID
ami-053a45ff0a704a47

AMI name
al2023-ami-2023.6.20250211.0-kernel-6.1-x86_64

Monitoring
disabled

Allowed image
-

Platform details
Linux/UNIX

Termination protection
Disabled

Install dependencies on both instances like Flask, python, pip, git, pytest.

sudo su

yum install -y python3 python3-pip

yum install git -y

pip install flask

pip install pytest

Clone git repo <https://github.com/akshaybhu/GithubAction-Python>

Run python using command:

Python3 app.py (**staging** running on port12345)

Python3 app.py (**prod** running on port12345)

Instances (1/3) [Info](#)

Last updated 9 minutes ago [Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[All states](#)

[aks](#) [X](#) [Clear filters](#)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabi...	Public I...	Public IPv4 ...	E
<input type="checkbox"/>	Aks_jenkins	i-00401c951e2c30580	Stopped	t2.medium	–	View alarms +	us-east-1a	–	–	–
<input checked="" type="checkbox"/>	Aks-Github-prod	i-0c94bb373a5e23df3	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	54.80.228.61	–
<input type="checkbox"/>	Aks-Github-staging	i-0fe7636b450ac018c	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	23.22.221.164	–

← → ↻ 🏠 ⚠ Not Secure 23.22.221.164:2222

Hello, World!

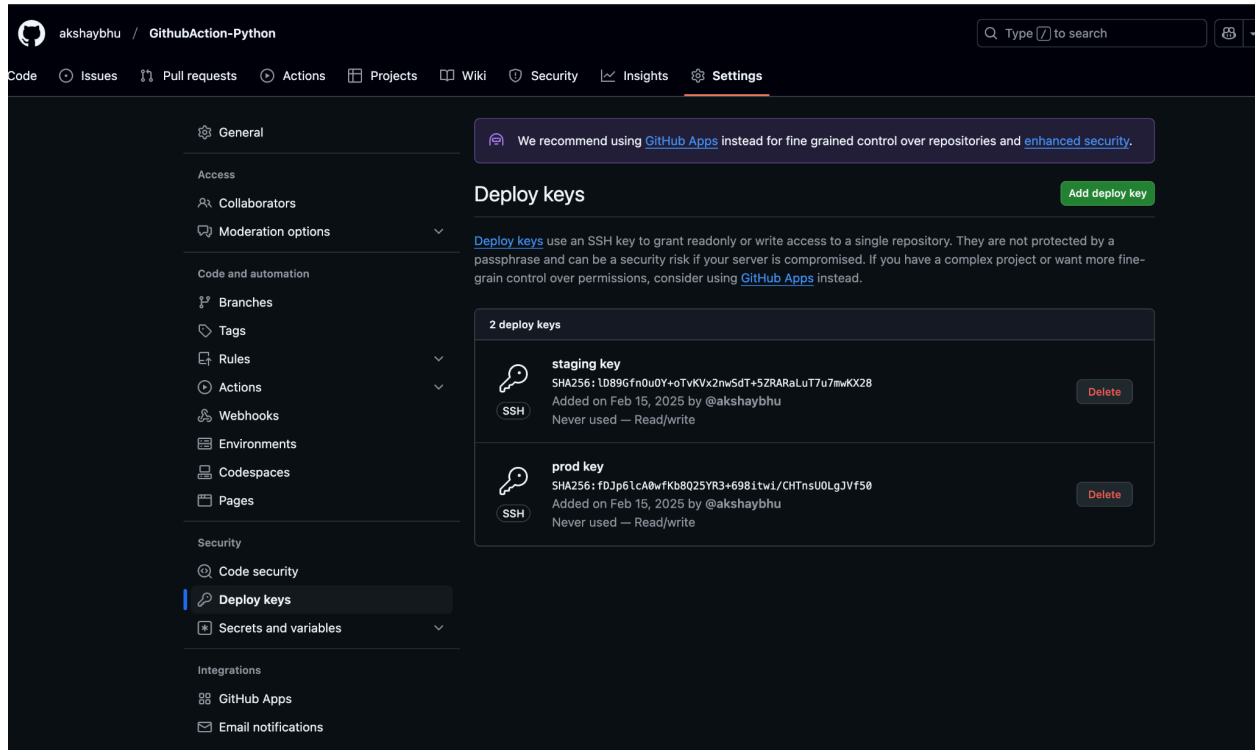
← → ↻ 🏠 ⚠ Not Secure 54.80.228.61:4444

Hello, World!

> Generate SSH keys on both staging/prod hosts:

```
ssh-keygen -t rsa -b 4096
```

> Add ssh pub keys under ~/.ssh/id_rsa.pub



> Add Secrets to GITHUB

Go to GitHub → Your Repository → Settings → Secrets and variables → Actions.
Click New Repository Secret and add:

PROD/STAGING_SSH_PRIVATE_KEY → (Paste private key)

PROD/STAGING_USER → ec2-user

PROD/STAGING_HOST → Public ip of EC2 instance

*PROD/STAGING_PATH → /home/ec2-user/**GithubAction-Python***

***PEM_KEY** → Copy your pem file output used to*

>> Login to staging/prod EC2 instance and follow same cmd to both before creating workflow:

```
sudo cp -r ~/.ssh/id_rsa* /home/ec2-user/.ssh/  
chmod 700 /home/ec2-user/.ssh/  
chown ec2-user:ec2-user /home/ec2-user/GithubAction-Python/
```

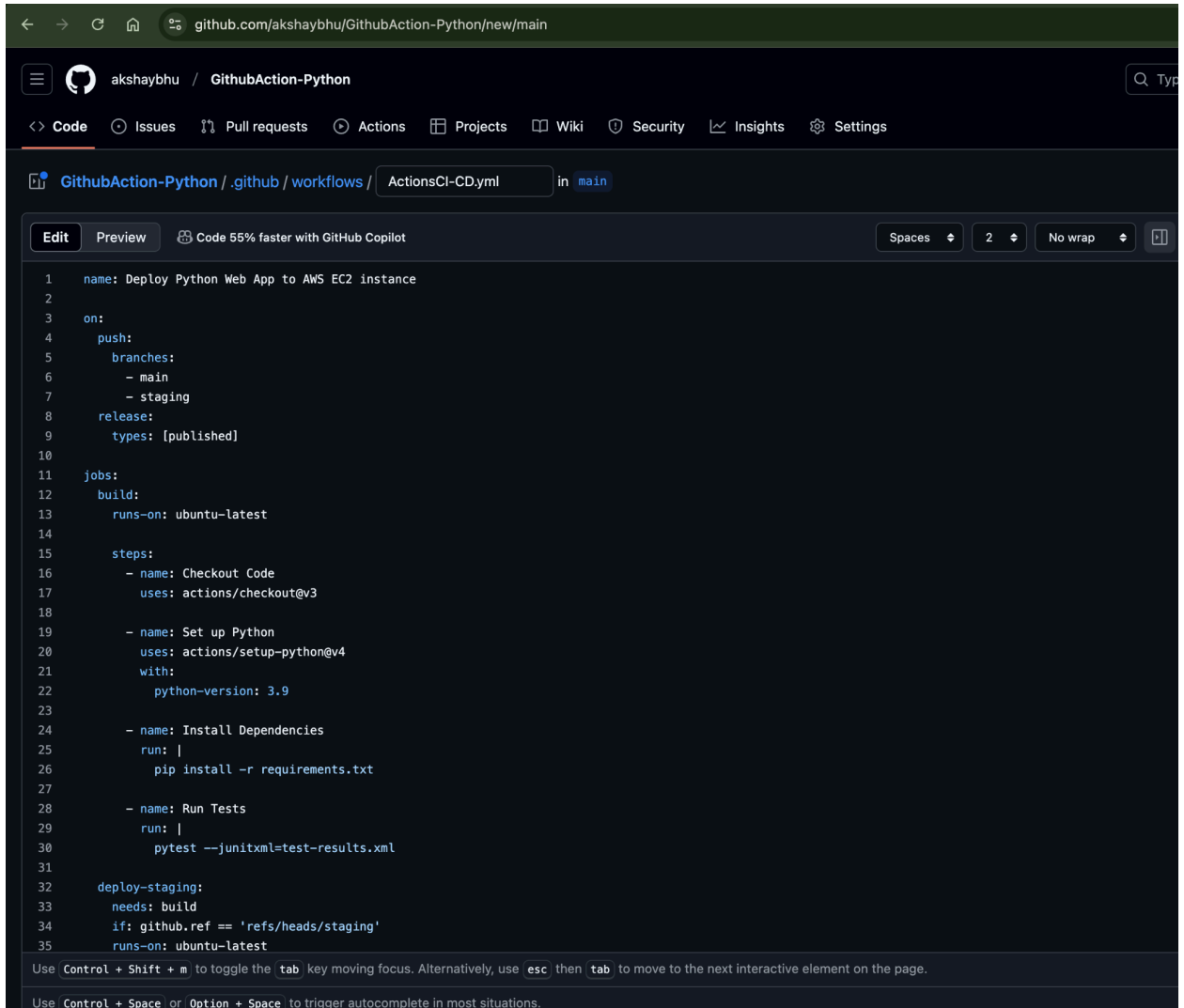
The screenshot shows the GitHub Actions secrets and variables page. The left sidebar contains navigation links for General, Access, Moderation options, Code and automation, Security, and Integrations. The main content area is titled 'Actions secrets and variables' and includes a description of secrets and variables. Below this, there are tabs for 'Secrets' and 'Variables'. The 'Secrets' tab is active, showing a table of repository secrets. The table has columns for 'Name' and 'Last updated'. The secrets listed are PROD_HOST, PROD_PATH, PROD_SSH_PRIV_KEY, PROD_USER, STAGING_HOST, STAGING_PATH, STAGING_USER, and STAG_SSH_PRIV_KEY. A 'New repository secret' button is located at the top right of the secrets table.

Name	Last updated
PROD_HOST	now
PROD_PATH	now
PROD_SSH_PRIV_KEY	6 minutes ago
PROD_USER	2 minutes ago
STAGING_HOST	4 minutes ago
STAGING_PATH	3 minutes ago
STAGING_USER	2 minutes ago
STAG_SSH_PRIV_KEY	1 minute ago

3. GitHub Actions Workflow:

Create workflow under path

.github >> workflows >> ActionsCI-CD.yml



The screenshot shows a GitHub repository page for 'akshaybhu / GithubAction-Python'. The breadcrumb navigation indicates the file path is `.github / workflows / ActionsCI-CD.yml` in the `main` branch. The file is being edited in a dark-themed editor. The workflow file content is as follows:

```
1  name: Deploy Python Web App to AWS EC2 instance
2
3  on:
4    push:
5      branches:
6        - main
7        - staging
8    release:
9      types: [published]
10
11 jobs:
12   build:
13     runs-on: ubuntu-latest
14
15     steps:
16       - name: Checkout Code
17         uses: actions/checkout@v3
18
19       - name: Set up Python
20         uses: actions/setup-python@v4
21         with:
22           python-version: 3.9
23
24       - name: Install Dependencies
25         run: |
26           pip install -r requirements.txt
27
28       - name: Run Tests
29         run: |
30           pytest --junitxml=test-results.xml
31
32   deploy-staging:
33     needs: build
34     if: github.ref == 'refs/heads/staging'
35     runs-on: ubuntu-latest
```

At the bottom of the editor, there are two lines of instructional text:

Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page.

Use `Control + Space` or `Option + Space` to trigger autocomplete in most situations.

>> Create a PR to staging branch and merge it to verify the Github Actions

This screenshot shows a GitHub Actions workflow run for the repository 'akshaybhu / GithubAction-Python'. The workflow is titled 'Deploy Python Web App to AWS EC2 instance' and the specific run is 'Update README.md #27'. The status is 'Success' with a total duration of 26s. The workflow diagram shows four jobs: 'build' (6s), 'deploy-staging' (5s), 'promote-to-main' (0s), and 'deploy-production' (0s). The 'build' job is connected to 'deploy-staging', which is connected to 'promote-to-main'. The 'deploy-production' job is also connected to 'build'.

Summary

- Jobs
 - build
 - deploy-staging
 - deploy-production
 - promote-to-main
- Run details
 - Usage
 - Workflow file

Triggered via push 9 minutes ago

Status: Success Total duration: 26s Artifacts: -

akshaybhu pushed eb9d75e staging

ActionsCI-CD.yml on: push

build 6s → deploy-staging 5s → promote-to-main 0s

deploy-production 0s

This will automatically create a PR from staging to the main branch, we can review the PR and merge to the main branch.
This will trigger automatic deployment again.

This screenshot shows a GitHub Pull Request titled 'Staging #10'. The pull request is 'Merged' and was created by 'akshaybhu' who merged 15 commits into 'main' from 'staging' 2 minutes ago. The pull request has 15 commits, 8 checks, and 3 files changed. The pull request is currently 'Closed'. The pull request has no reviews and no assignees.

Staging #10

Merged akshaybhu merged 15 commits into main from staging 2 minutes ago

Conversation 0 Commits 15 Checks 8 Files changed 3

akshaybhu commented 2 minutes ago

No description provided.

Reviewers: No reviews

Assignees: No one—assign yourself

This screenshot shows a GitHub Actions workflow run for the repository 'akshaybhu / GithubAction-Python'. The workflow is titled 'Deploy Python Web App to AWS EC2 instance' and the specific run is 'Staging #31'. The status is 'Success' with a total duration of 28s. The workflow diagram shows four jobs: 'build' (5s), 'deploy-staging' (0s), 'promote-to-main' (0s), and 'deploy-production' (8s). The 'build' job is connected to 'deploy-staging', which is connected to 'promote-to-main'. The 'deploy-production' job is also connected to 'build'.

Summary

- Jobs
 - build
 - deploy-staging
 - deploy-production
 - promote-to-main
- Run details
 - Usage
 - Workflow file

Triggered via pull request 1 minute ago

Status: Success Total duration: 28s Artifacts: -

akshaybhu closed #10 staging

ActionsCI-CD.yml on: pull_request

build 5s → deploy-staging 0s → promote-to-main 0s

deploy-production 8s

Verify the output again for app.py host run:

