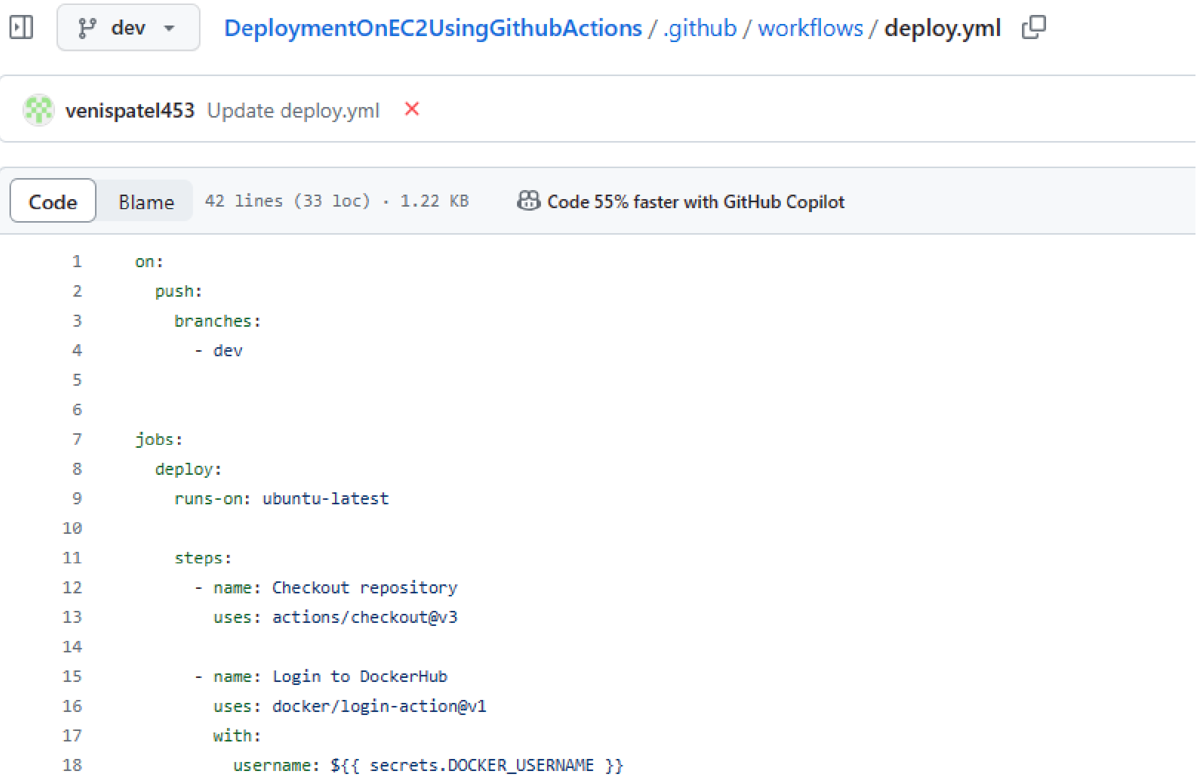
# Assignment 1

**Objective:** In this assignment, you will create a simple GitHub Actions workflow that builds a custom Docker image with a basic NGINX configuration, deploys it to an Amazon EC2 instance, and ensures that the container remains running even after an EC2 instance restart.

**STEPS:**

1. Write a Dockerfile, which will create a basic NGINX configuration docker image.



dev
DeploymentOnEC2UsingGithubActions / .github / workflows / deploy.yml
Code
Blame
Code 55% faster with GitHub Copilot

venispatel453 Update deploy.yml ✕

42 lines (33 loc) · 1.22 KB

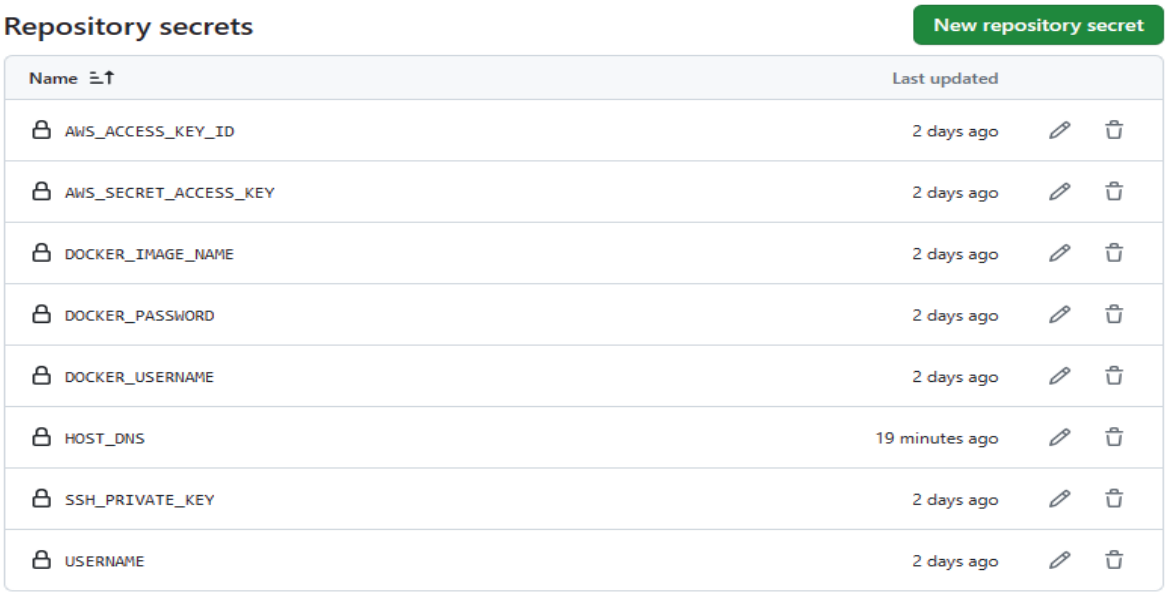
```yaml
on:
  push:
    branches:
      - dev


jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v3

      - name: Login to DockerHub
        uses: docker/login-action@v1
        with:
          username: ${{ secrets.DOCKER_USERNAME }}
```

```yaml
on:
  push:
    branches:
      - dev
```

```
jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v3

      - name: Login to DockerHub
        uses: docker/login-action@v1
        with:
          username: ${{ secrets.DOCKER_USERNAME }}
          password: ${{ secrets.DOCKER_PASSWORD }}

      - name: Build Docker image
        run: docker build -t ${{ secrets.DOCKER_USERNAME }}/venisnginx:latest .

      - name: Push Docker image to Docker Hub
        run: docker push ${{ secrets.DOCKER_USERNAME }}/venisnginx:latest

      - name: AWS login
        uses: aws-actions/configure-aws-credentials@v4
        with:
         aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
         aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
         aws-region: ap-south-1

      - name: SSH into EC2 instance
        uses: appleboy/ssh-action@master
        with:
```

```
        host: ${{ secrets.HOST_DNS }}
        username: ${{ secrets.USERNAME }}
        key: ${{ secrets.SSH_PRIVATE_KEY }}
        script: |
          sudo docker run -p 80:80 -d  --restart=always ${{ secrets.DOCKER_USERNAME}}/venisnginx:latest
```

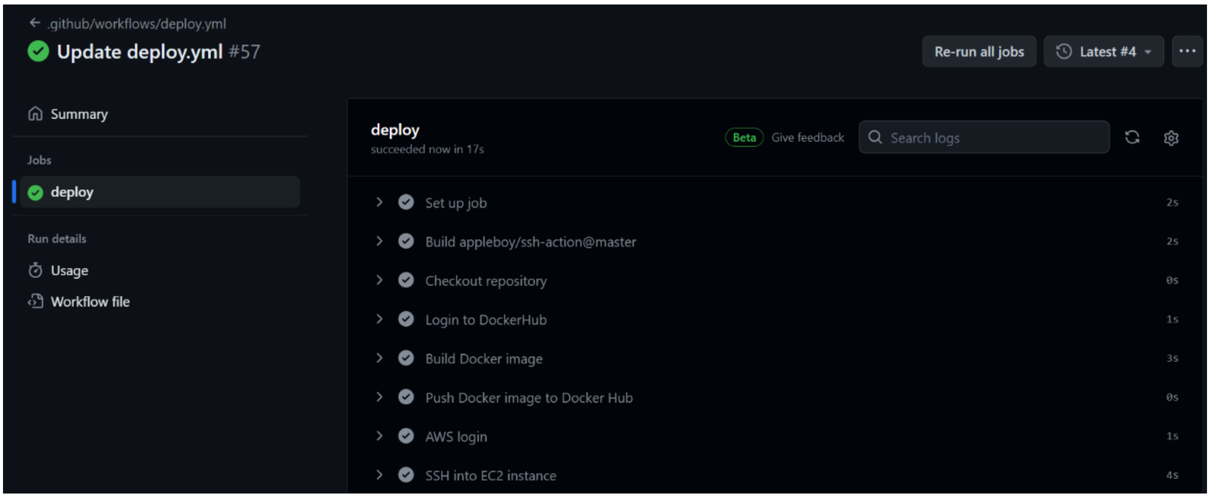
2. Now Create the Repository secrets to keep the data sensitive.



Repository secrets

| Name | Last updated |
|---|---|
| AWS_ACCESS_KEY_ID | 2 days ago |
| AWS_SECRET_ACCESS_KEY | 2 days ago |
| DOCKER_IMAGE_NAME | 2 days ago |
| DOCKER_PASSWORD | 2 days ago |
| DOCKER_USERNAME | 2 days ago |
| HOST_DNS | 19 minutes ago |
| SSH_PRIVATE_KEY | 2 days ago |
| USERNAME | 2 days ago |

3. Now go to Action and here we can see that our workflow has been successfully completed.



← .github/workflows/deploy.yml

Update deploy.yml #57

Re-run all jobs | Latest #4 | ···

Summary

Jobs

deploy

Run details

Usage

Workflow file

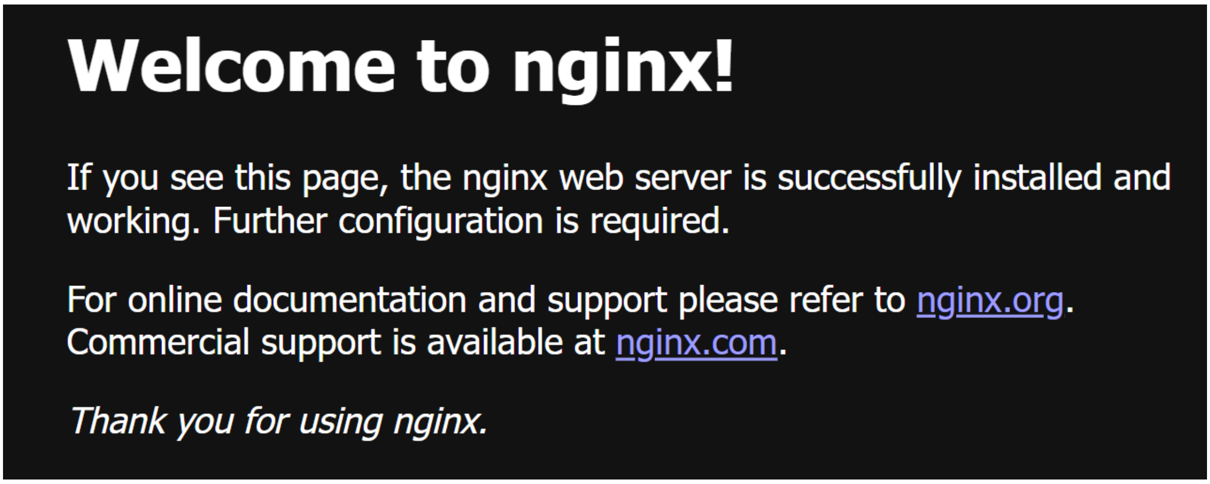
deploy

succeeded now in 17s

Beta | Give feedback | Search logs

```
Set up job                          2s
Build appleboy/ssh-action@master    2s
Checkout repository                 0s
Login to DockerHub                  1s
Build Docker image                  3s
Push Docker image to Docker Hub     0s
AWS login                           1s
SSH into EC2 instance               4s
```

4. Now go to EC2 instances on aws console and here our instance is running. Now go to public Ip address of EC2 instance and here we can see the NGINX welcoming page.



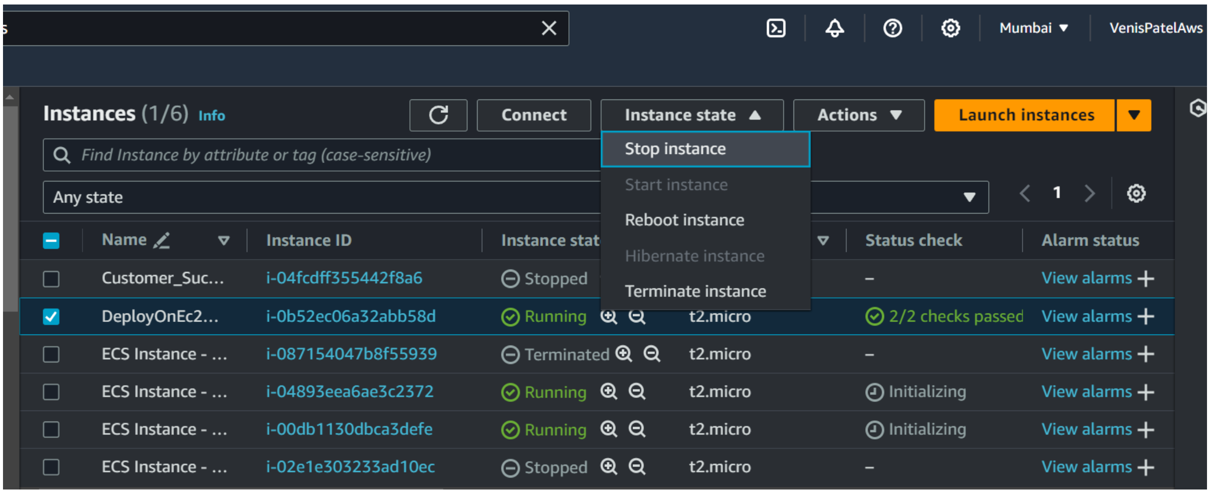
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

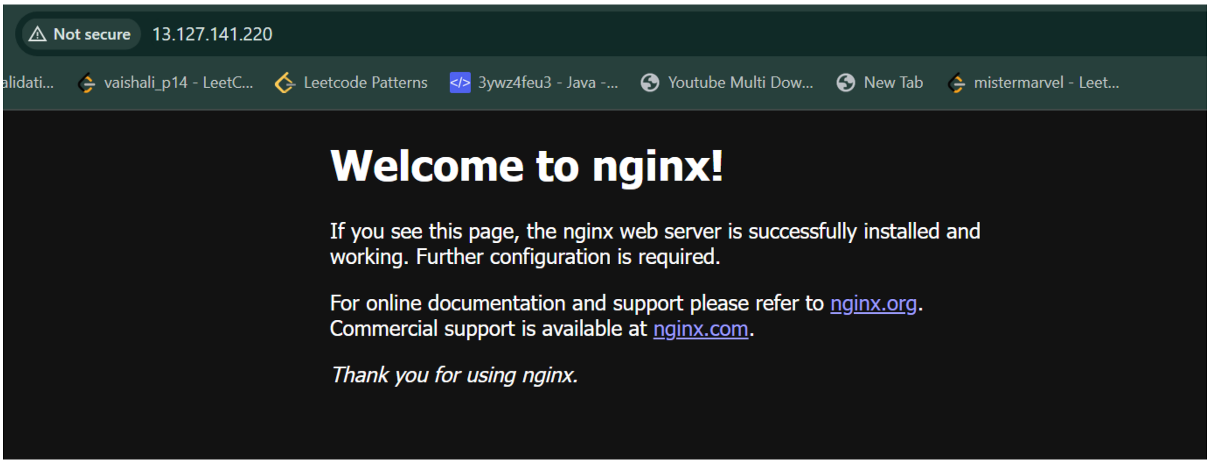
*Thank you for using nginx.*

5. Now let’s stop the instance to see if it keeps running after restarting instance.



| ☐ | Name | Instance ID | Instance stat | | Status check | Alarm status |
|---|---|---|---|---|---|---|
| ☐ | Customer_Suc... | i-04fcdff355442f8a6 | Stopped | | – | View alarms + |
| ☑ | DeployOnEc2... | i-0b52ec06a32abb58d | Running | t2.micro | 2/2 checks passed | View alarms + |
| ☐ | ECS Instance - ... | i-087154047b8f55939 | Terminated | t2.micro | – | View alarms + |
| ☐ | ECS Instance - ... | i-04893eea6ae3c2372 | Running | t2.micro | Initializing | View alarms + |
| ☐ | ECS Instance - ... | i-00db1130dbca3defe | Running | t2.micro | Initializing | View alarms + |
| ☐ | ECS Instance - ... | i-02e1e303233ad10ec | Stopped | t2.micro | – | View alarms + |

6. Now start the instance again and check that instance is running or not. Here we can see NGINX starting page which confirms that it runs even after restarting the instance.



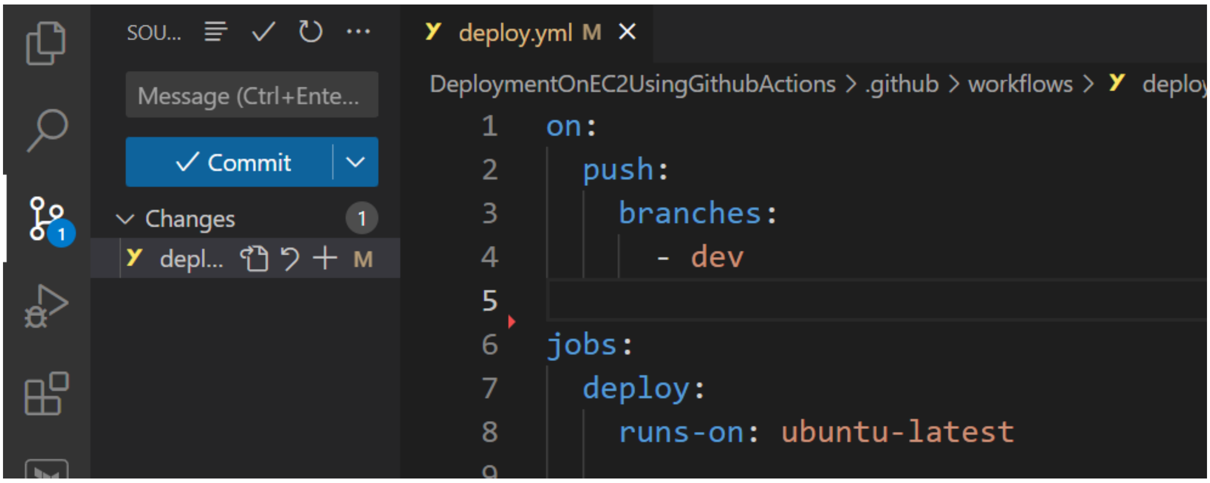
**Welcome to nginx!**

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

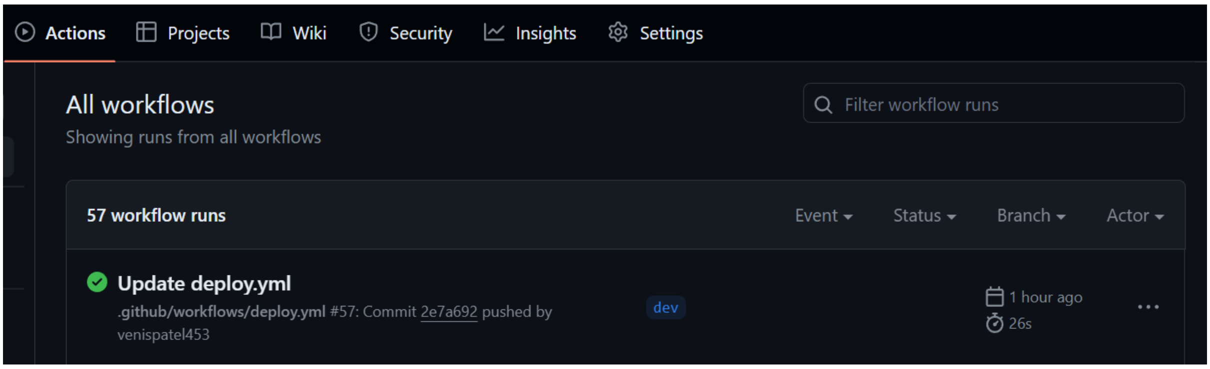
*Thank you for using nginx.*

7. Now let's change the code and push it into main branch and it should not trigger any event as we have written to happen only if it is pushed on dev branch.



```
on:
  push:
    branches:
      - dev

jobs:
  deploy:
    runs-on: ubuntu-latest
```

8. Here we can see that, there is no action has been triggered here.



57 workflow runs

Update deploy.yml
.github/workflows/deploy.yml #57: Commit 2e7a692 pushed by venispatel453
dev
1 hour ago
26s