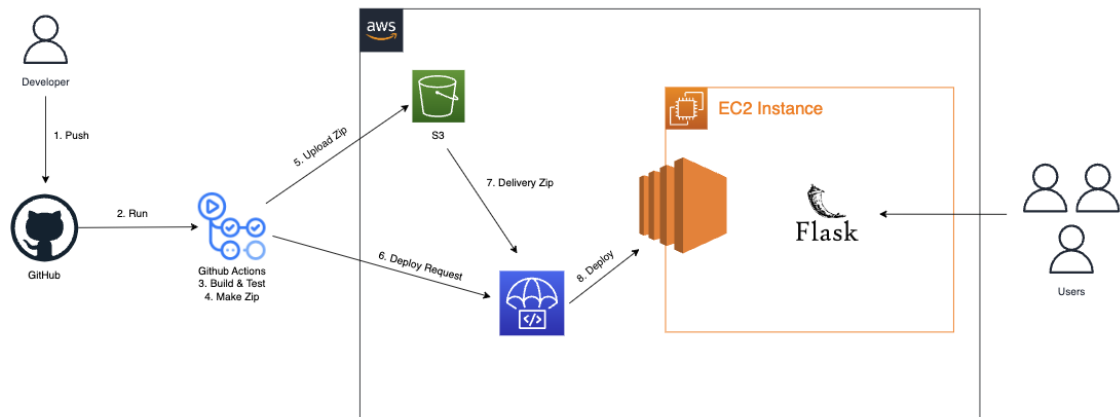


5주차

Github Actions를 이용한 flask CI/CD



1. 특정 브랜치에 **Push**, 또는 **Merge** 동작을 수행합니다.
2. **Github Actions**에서 **push** 혹은 **merge**를 감지하고 정해진 **Action**을 수행합니다.
3. Update된 코드를 바탕으로 **Build**와 **Test**를 수행합니다.
4. 코드를 압축하여 **zip**파일을 생성합니다.
5. **S3**에 zip파일을 업로드합니다.
6. **Code Deploy**에 배포 요청을 내립니다.
7. **S3**로부터 **zip**파일을 받습니다.
8. 지정한 **EC2** 인스턴스에 애플리케이션 파일을 전달합니다.
9. **EC2** 인스턴스에서 쉘 스크립트를 실행함으로써 배포를 완료합니다.

S3 버킷 및 EC2 역할 생성

| S3 : Github Actions로 부터 생성된 zip 저장

- IAM 사용자 생성 : S3와 CodeDeploy에 대한 권한 설정

- repo에 액세스 키 등록



IAM: Identity and Access Management → 퍼블릭 액세스 대신 IAM 사용자로 인스턴스에 접근하는 것이 더 안전하

EC2 : S3에서 zip파일을 가져와 CodeDeploy 수행, 배포

- EC2 인스턴스 생성
 - EC2 인스턴스에 생성한 IAM 역할 업데이트
 - EC2 인스턴스에 CodeDeploy Agent 설치

Github Actions에 배포 동작 추가

```
name: Build and Deploy Flask to AWS EC2

on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]

env: # 관련 env 키 및 ID
  PROJECT_NAME: <PROJECT_NAME>
  BUCKET_NAME: <BUCKET_NAME>
  CODE_DEPLOY_APP_NAME: <CODE_DEPLOY_APP_NAME>
  DEPLOYMENT_GROUP_NAME: <DEPLOYMENT_GROUP_NAME>

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

```

- uses: actions/checkout@v2

- name: Set up Python
  uses: actions/setup-python@v2
  with:
    python-version: <python-version>

- name: Install dependencies
  run: |
    python -m venv venv
    . venv/bin/activate
    pip install -r requirements.txt

- name: Create .env file
  run: |
    echo "<env_name>=${{ secrets.<env_name> }}" > .env
v

- name: Make Zip File
  run: |
    . venv/bin/activate
    zip -qq -r ./ $GITHUB_SHA.zip .
  shell: bash

- name: Configure AWS credentials
  uses: aws-actions/configure-aws-credentials@v1
  with:
    aws-access-key-id: ${${ secrets.AWS_ACCESS_KEY_ID
}}
    aws-secret-access-key: ${${ secrets.AWS_PRIVATE_AC
CESS_KEY }}
    aws-region: ap-northeast-2

- name: Upload to S3
  run: |
    aws s3 cp --region ap-northeast-2 ./ $GITHUB_SHA.z
ip s3://${BUCKET_NAME}/${PROJECT_NAME}/${GITHUB_SHA.zip

```

```
- name: Code Deploy To EC2 instance
  run: aws deploy create-deployment
      --application-name $CODE_DEPLOY_APP_NAME
      --deployment-config-name CodeDeployDefault.
AllAtOnce
      --deployment-group-name $DEPLOYMENT_GROUP_N
AME
      --s3-location bucket=$BUCKET_NAME,bundleType=zip,key=$PROJECT_NAME/$GITHUB_SHA.zip
```