Github Actions 기반 CI/CD - Build 와 Push

> 💡 **실습 목표**
> **1. Github Actions 를 활용하여 Build 한다.**
> **2. Docker Image 를 Build하여 생성 및 Docker Hub 에 Push 한다.**

Github Actions 를 활용하여 Build

- Github 에 접속하여 새로운 Repository (github-actions-project) 생성
- Actions 클릭
  - 매우 다양한 툴 들을 통합하기 쉽게 되어 있음
- Python Application 선택

▼ Workflow file 생성

- Github event 알아보기
  - https://docs.github.com/en/actions/learn-github-actions/events-that-trigger-workflows

```
name [optional]
on [required] events
  : workflow 를 시작하게 할 수 있는 Github event 의 이름
  : jobs [required]   jobs.<job_id>
  : one or more jobs
  : sequence of tasks (steps)
  : steps 1) can run commands, 2) setup tasks 3) run an action
    - uses : selects an action (actions/ 다음에는 재사용 가능 코드 위치)
    - run  : runs a command-line command
```

```
name: Python application

on:
  push:
    branches: [ python-ci-workflow ]
  pull_request:
    branches: [ python-ci-workflow ]

jobs:
  build:
```

```
    steps:
    - uses: actions/checkout@v2
    - name: Set up Python
      uses: actions/setup-python@v2
      with:
        python-version: "3.8"
    - name: Display Python version
      run: python -c "import sys; print(sys.version)"
```

- actions 알아보기
  - https://github.com/actions
  - checkout - action.yaml
- yml 파일 이름을 ci.yml 으로 변경
- Start commit → Create a new branch.. → 이름을 'python-ci-workflow' 로 변경 → Create pull request
- Details 클릭 → build
- 이 코드들은 어디서 실행되는 걸까?
  - Github 에 의해 관리된다
  - Workflow 의 각 jobs 은 새로운 가상 환경에서 실행된다

▼ runs-on 은 실행되는 서버의 운영체제를 나타낸다.

- ubuntu, Windows, Mac

```
jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        os: [ubuntu-latest, windows-latest, macOS-latest]
```

▼ ci.yml 을 업데이트 한다.

- 세 가지 운영 체제 모두에서 세 개의 빌드가 병렬로 실행된다.

```
name: Python application

on:
  push:
    branches: [ python-ci-workflow ]
  pull_request:
    branches: [ python-ci-workflow ]
```

```
jobs:
  build:
    runs-on: ${{ matrix.os }}
    strategy:
      matrix:
        os: [ubuntu-latest, macos-latest, windows-latest]
        python-version: ['3.6', '3.8']
        exclude:
          - os: macos-latest
            python-version: '3.8'
          - os: windows-latest
            python-version: '3.6'

    steps:
    - uses: actions/checkout@v2
    - name: Set up Python
      uses: actions/setup-python@v2
      with:
        python-version: ${{ matrix.python-version }}
    - name: Display Python version
      run: python -c "import sys; print(sys.version)"
```

Docker Image 를 Build하여 생성 및 Docker Hub 에 Push 한다.

- Workflow 설명
  - 작성한 App 을 빌드한다
  - Docker 이미지로 생성한다
  - Docker Repository 로 추가한다

▼ Dockerfile 예제 가져오기

- https://docs.docker.com/language/python/build-images/
- app.py

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, Docker!'
```

- Dockerfile

```
# syntax=docker/dockerfile:1
```

```
FROM python:3.8-slim-buster

WORKDIR /app

COPY requirements.txt requirements.txt
RUN pip3 install -r requirements.txt

COPY . .

CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

- requirements.txt

```
flask
```

▼ Docker Image 빌드

- Docker hub 로 이동 (https://hub.docker.com/)
- github-actions-app 라는 이름으로 repo 생성
- 구글에서 docker build and push action 으로 검색 (https://github.com/marketplace/actions/docker-build-push-action)
  - Docker Hub 지원

    dockerhub 에서 account setting → Security → New Access Token 생성 메모장에 저장
  - Secrets 설정 필요
    - github - 레포지토리의 settings - Secrets - New
    - Name : DOCKER_USERNAME - username
    - Name : DOCKER_PASSWORD - dockerhub의 access token 비밀번호의 경우 실제 비밀번호와 Access Tokens 모두 가능
- 다양한 Inputs
- ci.yml에 dockerhub repo설정 추가

```
- name: Build & push Docker image
  uses: mr-smithers-excellent/docker-build-push@v5
  with:
    image: docker-hub-repo/image-name
    tags: v2, latest
    registry: docker.io
    username: ${{ secrets.DOCKER_USERNAME }}
    password: ${{ secrets.DOCKER_PASSWORD }}
```

- [본인 Docker Hub 아이디]/github-actions-app 으로 image 변경
- os 는 ubuntu-latest 만 남겨두기

- ci.yml 최종본

```
name: Python application

on:
  push:
    branches: [ python-ci-workflow ]
  pull_request:
    branches: [ python-ci-workflow ]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - name: Set up Python
      uses: actions/setup-python@v2
      with:
        python-version: "3.8"
    - name: Display Python version
      run: python -c "import sys; print(sys.version)"
    - name: Build & push Docker image
      uses: mr-smithers-excellent/docker-build-push@v5
      with:
        image: docker-hub-repo/image-name
        tags: v3, latest
        registry: docker.io
        username: ${{ secrets.DOCKER_USERNAME }}
        password: ${{ secrets.DOCKER_PASSWORD }}
```