

03

AWS EC2에 docker 사용

01. docker 설치

■ 관리자 권한 주기

- `sudo su`

■ 우분투 시스템 패키지 업데이트

- `apt update/apt upgrade`

■ 필요한 패키지 설치

- `apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common`

■ docker의 공식 GPG키를 추가

- `mkdir -p /etc/apt/keyrings`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg -
-dearmor -o /etc/apt/keyrings/docker.gpg`
- `echo "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs)
stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null`

01. docker 설치

■ docker의 공식 apt 저장소를 추가

- `add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"`

■ 시스템 패키지 업데이트

- `apt update`

■ docker 설치

- `apt install docker-ce docker-ce-cli containerd.io`

02. 스왑 메모리 설정

- 스왑 메모리 확인
 - `free -h`
 - `swapon -s`
- swapfile 메모리 할당
 - `fallocate -l 2G /swapfile`
- swapfile 권한 설정
 - `chmod 600 /swapfile`
- swap 공간 생성
 - `mkswap /swapfile`
- swapfile 스왑 메모리 추가
 - `swapon /swapfile`
- 정상 동작 확인
 - `swapon -s`

02. 스왑 메모리 설정

- 부팅 시 자동 적용

- `echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab`

- `free -h` 명령어로 메모리 상태 확인

03. Docker hub

■ EC2에서 pull 및 run

- `docker pull closer19/dockerdb`
- `docker run -d -p 3308:3306 closer19/dockerdb`

- `docker pull closer19/dockerbackend`
- `docker run -d -p 8080:8080 closer19/dockerbackend`

- `docker pull closer19/dockerfrontend`
- `docker run -d -p 80:80 closer19/dockerfrontend`

04

Github Actions

01. Secrets 등록

■ Docker hub Token 발급

- 우측 상단 프로필 아이콘 클릭
- Account settings – Personal access tokens
- – Generate new token

■ Github –settings-secrets and variables-action-new repository secret 에 추가

- DOCKERHUB_TOKEN
- DOCKERHUB_USERNAME
- LIVE_SERVER_IP
- EC2_SSH_KEY

02. Workflow File

- 자동으로 실행되어야 할 내용을 작성
- .github – workflows 폴더 안에 * . yml 파일로 작성
 - Github 웹사이트의 Actions – New workflow를 통해 작성
 - 로컬의 project안에서 작성 후 github에 push

02. Workflow File

■ job과 step으로 구성

```
name: # 워크플로우 이름

on:   # 트리거 조건
  push:
    branches: [main]
  pull_request:
    branches: [main]

jobs:   # 실행할 작업들
  job_id:
    runs-on: ubuntu-latest      # 선행 job
    steps:                      # 단계별 실행 명령
      - name: # 작업 이름
        uses: actions/checkout@v3 # 외부 액션 사용
      - name: # 작업 이름
        run: echo "Hello World"   # 명령 실행
```

02. Workflow File

■ 들여쓰기

- 공백 2칸 또는 4칸을 사용해야 하며, 혼용하면 안됨

■ -, , :

- 뒤에 반드시 한 칸을 띄어야 함

■ key: value 형태는 한 줄로, value가 긴 경우 여러 줄(|) 사용 가능

- script: |
- echo "line 1"
- echo "line 2"

■ \${ 변수 사용 }