22일차

Docker

클라우드 상에서 환경을 동일화하거나, 동일화된 환경을 제공하기 위해서 사용하는 기술

Container

각각의 프로그램들이 들어있는 묶음

Kubernetes

필요한 것들이 담겨있는 container들이 많아지게 되면 그것들을 전체적으로 조율하고 관리할 프로그램이 필요 -> kubernetes의 필요성

유저는 Kubernetes Master를 통해 Worker Node들이 일하도록 지시한다

문제는 부하직원이 시키는 대로 일을 하는데, 얘가 사장님 아들이어서 자세히 말해주지 않으면 알아듣지 못하고, 이해하지 못하면 일을 안한다…

그래서 직원이 알아들을 수 있게 잘 설명해서 일을 시켜야 한다

+ 그렇게 일을 시키기 위해 필요한 업무내용을 담은 파일이 yaml 파일

Dockerfile(.yaml) – 컨테이너를 만들기 위한 설계도면 -> 컨테이너의 이미지를 만들어낸다

Container Image – 만들어진 이미지 파일을 통해 도커 컨테이너를 만들어낸다

그러나 이런 식으로 하나씩 만들어서는 대량 작업이 어렵다 -> Pod를 만들어서 관리하자!

Kubernetes에게 명령을 하여 Pod를 만들어주면, Pod 단위로 컨테이너들을 관리해준다

sudo apt-get update

sudo apt-get upgrade

# 우분투 환경의 업데이트

sudo apt-get install \

    apt-transport-https \

    ca-certificates \

    curl \

    gnupg \

    lsb-release

# 필요한 프로그램 인스톨

curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo \

  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] <https://download.docker.com/linux/ubuntu> \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# curl을 통해 웹에서 필요한 파일들을 다운로드

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io

# 마찬가지로 업데이트 및 인스톨

sudo docker run hello-world

# 도커가 잘 설치되었는지 확인해주는 도커 이미지인 hello-world를 실행해본다

sudo usermod -a -G docker $USER

sudo service docker restart

# 도커에 관리자 권한을 부여하고 재실행

docker pull ubuntu:18.04

# 도커에 우분투 18.04 설치

docker images

# 로컬에 저장되어 있는 도커 이미지들을 확인

docker ps

docker ps -a

#현재 실행중인 프로그램들을 확인

docker run -it --name demo1 ubuntu:18.04 /bin/bash

docker run -it -d --name demo2 ubuntu:18.04

docker ps

docker exec -it demo2 /bin/bash

docker run --name demo3 -d busybox sh -c "while true; do $(echo date); sleep 1; done"

docker logs demo3

docker logs demo3 -f

docker stop demo3

docker stop demo2

docker stop demo1

docker rm demo3

docker rm demo2

docker rm demo1

docker images

# busybox, ubuntu 가 있는 것을 확인하실 수 있습니다.

docker rmi ubuntu