**도커 이미지 무작정 따라해 보기**

**이미지 기반 컨테이너 생성**

**docker image ls**

**docker run --name my-nginx -d -p 8080:80 nginx**

**docker run --name my-new-nginx -d -p 8081:80 nginx**

**docker image ls**

**docker container ls**

* 서비스 확인
  + Cloud IDE 메뉴 Labs > 포트열기 > 8080
  + Cloud IDE 메뉴 Labs > 포트열기 > 8081

**컨테이너와 이미지 삭제하기**

* 삭제하려는 이미지를 사용하는 컨테이너 정리가 우선

**docker container ls ; 실행중인 컨테이너 확인**

**docker container stop my-nginx**

**docker container stop my-new-nginx**

**docker container rm my-nginx**

**docker container rm my-new-nginx**

**docker image rm nginx**

**docker images**

**이미지 생성하고 Remote Registry(**[**Hub.docker.com**](http://hub.docker.com/)**)에 푸시하기**

* 어플리케이션 및 이미지 빌드 스크립트(Dockerfile) 생성
  + Cloud IDE 메뉴 > File > Folder > Docker 입력
  + 생성한 폴더 하위에 아래 2개 파일 생성
  + Cloud IDE 메뉴 > File > New File > index.html 입력
  + 파일 내용에

**Hi~ My name is Hong Gil-Dong...~~~**

* 입력 후 저장
* Cloud IDE 메뉴 > File > New File > Dockerfile (확장자 없음)
* 파일 내용에

**FROM nginx**

**COPY index.html /usr/share/nginx/html/**

* 입력 후, 저장
* 이미지 빌드하기

**docker build -t apexacme/welcome:v1 .**

**docker images**

* 이미지 원격 저장소에 푸시하기
* 도커허브 계정 생성
* [https://hub.docker.com](https://hub.docker.com/) 접속
  + 가입(Sign-Up) 및 E-Mail verification 수행

**docker login**

**docker push apexacme/welcome:v1**

**# apexacme 가 자신의 계정명인 경우**

주의사항: access denied 오류가 나면, 로그인이 되지 않았거나, apexacme 를 자신의 계정명으로 저장소 명을 쓰지 않아서 입니다. e.g. apexacme --> 자신의 계정명

**Docker Hub에 생성된 이미지 확인**

* [https://hub.docker.com](https://hub.docker.com/) 접속
* repositories 메뉴 Reload 후 Push된 이미지 확인

**Docker Hub 이미지 기반 컨테이너 생성**

**docker image rm apexacme/welcome:v1**

**docker run --name=welcome -d -p 8080:80 apexacme/welcome:v1**

* 서비스가 잘 기동 되었는지 확인:  
  새 터미널을 열고 (Menu > Terminal > New Terminal)

**$ http localhost:8080**

**HTTP/1.1 200 OK**

**Accept-Ranges: bytes**

**Connection: keep-alive**

**Content-Length: 23**

**Content-Type: text/html**

**Date: Wed, 12 May 2021 05:12:28 GMT**

**ETag: "609b5cd7-17"**

**Last-Modified: Wed, 12 May 2021 04:43:03 GMT**

**Server: nginx/1.19.10**

**<h1> Hello world </h1>**

**다음과정 미리보기**

**$ kubectl run myhomepage --image=jinyoung/welcome:v1**

**deployment.apps/myhomepage created**

**$ kubectl expose deploy myhomepage --port=80 --type=LoadBalancer**

**service/myhomepage exposed**

**$ kubectl get svc -w**

**NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE**

**myhomepage LoadBalancer 10.100.98.191 addef84b932ff416186e2166ff397d74-589148294.ap-northeast-2.elb.amazonaws.com 80:30271/TCP 9s**

**$ http addef84b932ff416186e2166ff397d74-589148294.ap-northeast-2.elb.amazonaws.com:80**

**HTTP/1.1 200 OK**

**Accept-Ranges: bytes**

**Connection: keep-alive**

**Content-Length: 23**

**Content-Type: text/html**

**Date: Wed, 12 May 2021 05:36:40 GMT**

**ETag: "609b5cd7-17"**

**Last-Modified: Wed, 12 May 2021 04:43:03 GMT**

**Server: nginx/1.19.10**

**<h1> Hello world </h1>**

**kubectl get all**

**NAME READY STATUS RESTARTS AGE**

**pod/myhomepage-58dd9ffb74-kw5km 1/1 Running 0 17m**

**NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE**

**service/myhomepage LoadBalancer 10.100.98.191 addef84b932ff416186e2166ff397d74-589148294.ap-northeast-2.elb.amazonaws.com 80:30271/TCP 15m**

**NAME READY UP-TO-DATE AVAILABLE AGE**

**deployment.apps/myhomepage 1/1 1 1 17m**

**NAME DESIRED CURRENT READY AGE**

**replicaset.apps/myhomepage-58dd9ffb74 1 1 1 17m**

**$ kubectl get rs -w**

**NAME DESIRED CURRENT READY AGE**

**myhomepage-58dd9ffb74 1 1 1 27m**

**# 새 터미널**

**$ kubectl delete po --all**

**pod "myhomepage-58dd9ffb74-wjf68" deleted**

**# 아까 터미널에서 rs 의 desired 와 current 가 유지됨 (pod 가 재생됨)을 확인:**

**myhomepage-58dd9ffb74 1 0 0 28m**

**myhomepage-58dd9ffb74 1 1 0 28m**

**myhomepage-58dd9ffb74 1 1 1 28m**

**Github Container Registry 사용하기**

**Login**

**docker login ghcr.io -u <github계정명> -p <Personal Access Token>**

* github 계정명은 이메일주소가 아닌 github 자체 계정 문자열입니다.
* Personal Access Token을 얻으려면, Account > Settings > Developer Settings > Personal Access Token 에서 Generate New Token 한후, 권한으로 “write package” 를 부여하신 후 생성된 토큰을 얻으면 됩니다.

**Build / Push예시**

**docker build -t ghcr.io/jinyoung/welcome:v2021101202 .**

**docker push ghcr.io/jinyoung/homepage:v2021101202**

* build 시 이미지명은 앞에 꼭 [ghcr.io/를](http://ghcr.io/%EB%A5%BC) 추가
* push 시에는 항상 동일한 이미지명 준수

**이미지 확인 및 접근권한설정**

Account > Your Repositories > Packages 에서 확인가능

권한을 설정하기 위해서는 Setting package 를 클릭한 후, Set Visibility 를 클릭하고 팝업에서 Public 설정 후, 이름을 확인해주고 설정완료.

**상세설명**

Checkpoints

1. DOCKER-ACCOUNT/my-nginx:v1 의 이미지를 생성한다.

2. DOCKER-ACCOUNT/my-nginx:v1 를 실행한다.