Chapter 7

7장에 들어가기 전..

window ubuntu 환경 세팅

https://learn.microsoft.com/en-us/windows/wsl/install-manual

https://velog.io/@juhyeon1114/IntelliJ-터미널-wsl2로-바꾸기

```
# JAVA 설치
sudo apt update
sudo apt install openjdk-17-jdk

# sdkman 설치 및 적용
sudo apt update
sudo apt install unzip
sudo apt install zip
sudo curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk version

# GRYPE 설치
curl -sSfL https://raw.githubusercontent.com/anchore/grype/ma

# docker 설치
## 기존내역 삭제
for pkg in docker.io docker-doc docker-compose docker-compose

## Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg
sudo chmod a+r /etc/apt/keyrings/docker.asc

## Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" |
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

## 설치
sudo apt-get install docker-ce docker-ce-cli containerd.io do

## 실행
sudo service docker start
sudo service docker status

## 테스트
sudo docker run hello-world

# postgre image
docker run -d \
--name polar-postgres \
-e POSTGRES_USER=user \
-e POSTGRES_PASSWORD=password \
-e POSTGRES_DB=polardb_catalog \
-p 5432:5432 \
postgres:14.4
```

6장 247페이지 docker-compose up -d 실행 시 에러 해결 (window)

- config-server에 dockerfile 추가 (pr에 올려둠)
- 아래 포트 권한 문제는 이 블로그에서 해결 가능

> Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:8001 -> 0.0.0.0:0: listen tcp 0.0.0.0:8001: bind: An attempt was made to access a socket in a way forbidden by its access permissions.

7장

ubuntu 환경 세팅

```
sudo apt-get update -y
sudo apt-get install curl wget apt-transport-https virtualbox

# minikube 설치
wget https://storage.googleapis.com/minikube/releases/latest/
sudo cp minikube-linux-amd64 /usr/local/bin/minikube
sudo chmod 755 /usr/local/bin/minikube

minikube version

# kubectl 설치
echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Releas
sudo apt update
sudo apt install -y kubelet kubeadm kubectl

# minikube
minikube start
```

```
# 상태 확인
kubectl get node -o wide
minikube status
```

도커 위에 polar 쿠버네티스 클러스터 생성

```
# 쿠버네티스 클러스터 생성
minikube start --cpus 2 --memory 4g --driver docker --profile

# 클러스터 내 노드 목록 확인
kubectl get nodes

# 상호작용할 수 있는 모든 컨텍스트 나열
kubectl config get-contexts

# kubectl이 현재 사용하는 클러스터 확인
kubectl config current-context

# 현재 사용하는 클러스터 변경
kubectl config use-context polar

# 클러스터 중지, 재시작, 삭제
minikube stop --profile polar
minikube start --profile polar
minikube delete --profile polar
```

로컬 클러스터에서 데이터 서비스 관리

```
cd PolarBookshop/polar-deployment/kubernetes/platform/develop

# postgreSQL 배포
```

```
kubectl apply -f services

# postgreSQL pod 확인
kubectl get pod

# 데이터베이스 로그 확인
kubectl logs deployment/polar-postgres

# 데이터베이스 배포 취소
kubectl delete -f services
```

쿠버네티스 매니페스트

```
apiVersion : 객체를 생성하기 위해 사용할 쿠버네티스 API 버전
- apps/v1
- `kubectl explain <object_name>` ex) pods

kind : 생성하기 원하는 객체의 종류
- Pod, ReplicaSet, Deployment, Service, ...
- `kubectl api-resources` 통해서 알 수 있음

metadata : 객체를 고유하게 식별하기 위해 사용하는 데이터

spec : 원하는 설정을 선언
```

catalog-service 배포 객체 생성

```
cd PolarBookshop/catalog-service

# 이미지 빌드
./gradlew bootBuildImage
```

```
# minikube 수동으로 로컬 클러스터 이미지 엑세스
minikube image load catalog-service --profile polar

# 클러스터에 쿠버네티스 매니페스트 적용
kubectl apply -f k8s/deployment.yml

# 만들어진 객체 확인
kubectl get all -l app=catalog-service

# 배포 로그 확인
kubectl logs deployment/catalog-service

# 파드 status 확인
kubectl get pods
```

서비스 디스커버리 및 로드 밸런싱

```
cd PolarBookshop/catalog-service

# 매니페스트를 사용한 서비스 객체 생성
kubectl apply -f k8s/service.yml

# 결과
kubectl get svc -l app=catalog-service

# 서비스 객체를 로컬 컴퓨터에 노출하기 위해 포트 포워딩 -> localhost:90
kubectl port-forward service/catalog-service 9001:80
```

확장성과 일회성

```
# graceful -> application.yml 설정
cd PolarBookshop/catalog-service

./gradlew bootBuildImage
minikube image load catalog-service --profile polar

# preStop -> deployment.yml 설정
kubectl apply -f k8s/deployment.yml

# 테스트 prestop 길게하고 테스트 가능
kubectl get pods -l app=catalog-service
kubectl delete pod <pod-name>
kubectl logs <pod-name> --previous
kubectl get pods -l app=catalog-service

# replicas -> deployment.yml 설정
kubectl apply -f k8s/deployment.yml

kubectl get pods -l app=catalog-service

# 특정 파드 삭제
kubectl delete pod <pod-name>

kubectl get pods -l app=catalog-service # 삭제했음에도 2개의 파드

# 리소스 제거, 클러스터 정리
kubectl delete -f k8s

cd PolarBookshop/polar-deployment/kubernetes/platform/develop
kubectl delete -f services
```

틸트를 사용한 로컬 쿠버네티스 개발

```
# postgreSQL 배포
cd PolarBookshop/polar-deployment/kubernetes/platform/develop

kubectl apply -f services

# tilt 설치
curl -fsSL https://raw.githubusercontent.com/tilt-dev/tilt/ma

# tilt 실행
cd PolarBookshop/catalog-service

tilt up # 실행안되면 환경변수에 추가하기

# 접속 -> :9001/books 테스트
http://localhost:10350/

# tilt 종료
tilt down
```

옥탄트를 사용한 쿠버네티스 워크로드 시각화

```
# 설치
wget https://github.com/vmware-tanzu/octant/releases/download

sudo dpkg -i octant_0.24.0_Linux-64bit.deb

sudo apt-get install -f

octant version

# 실행
octant
http://localhost:7777
```

```
# 종료
tilt down

cd PolarBookshop/polar-deployment/kubernetes/platform/develop
kubectl delete -f services
```

큐비발을 사용한 쿠버네티스 매니페스트 유효성 검사

```
cd PolarBookshop/catalog-service

# 설치
wget https://github.com/instrumenta/kubeval/releases/download

tar -xvf kubeval-linux-amd64.tar.gz

sudo mv kubeval /usr/local/bin/

sudo chmod +x /usr/local/bin/kubeval

kubeval --version

# k8s 디렉터리 내의 쿠버네티스 매니페스트가 유효한지 검사
kubeval --strict -d k8s
```