#aws 로그인

aws configure

#도커 로그인

docker login --username AWS -p $ecrcode $userid.dkr.ecr.$region.amazonaws.com

#시간 설정

cd /etc

ls -al localtime

localtime -> /usr/share/zoneinfo/Etc/UTC

rm /etc/localtime

ln -s /usr/share/zoneinfo/Asia/Seoul localtime

#namespace 구성

kubectl create ns kafka

kubectl create ns db

#카프카 설치

helm install my-kafka bitnami/kafka --namespace=kafka

#mysql 설치

helm install mysql bitnami/mysql --namespace=mysql

eksctl create cluster --name $eksname --version 1.21 --spot --managed --nodegroup-name standard-workers --node-type t3.medium --nodes 3 --nodes-min 3 --nodes-max 6

>>eks로 노드 만든것이다

aws eks --region $region update-kubeconfig --name $eksname

>>config 설정?

aws eks --region ap-northeast-3 update-kubeconfig --name team2-eks

>>config파일이 알아서 생성이 됨

export ecrcode=$(aws --region $region ecr get-login-password)

>>ecrcode를 받아오는 것이다

docker login --username AWS -p $ecrcode $userid.dkr.ecr.$region.amazonaws.com

>>푸쉬할 때 필요함

#topic 리스트 확인 및 생성

/opt/bitnami/kafka/bin/kafka-topics.sh --bootstrap-server http://localhost:9092 --list

## producer 생성

/opt/bitnami/kafka/bin/kafka-console-producer.sh --broker-list http://localhost:9092 --topic capstone

## consumer 생성

/opt/bitnami/kafka/bin/kafka-console-consumer.sh --bootstrap-server http://localhost:9092 --topic capstone --from-beginning

/opt/bitnami/kafka/bin-consumer-groups.sh --bootstrap-server http://localhost:9092 --list

#topic 리스트 확인

/opt/bitnami/kafka/bin/kafka-topics.sh --bootstrap-server http://localhost:9092 --list

#서버접속

kubectl exec -it my-kafka-0 --namespace=kafka -- bash

#패키지 생성 (run을 안해줘도 됨)

mvn package -B

#도커이미지 생성

docker build -t jaj1012/capstone:v1 . -f Dockerfile

#도커이미지 푸시

docker push jaj1012/capstone:v1

#yaml파일 설정 및 실행(deployment.yaml, service.yaml, application.yaml)

containers:

        - name: bankingteam

          image: jaj1012/bankingteam:v2

          envFrom:

            - configMapRef:

                name: myconfig

>>envFrom 설정

k apply -f deployment .

k apply -f service .

#configmap.yaml

apiVersion: "v1"

kind: "ConfigMap"

metadata:

  name: "myconfig"

data:

  DBID: "root"

  DBPW: "Ixp8rEHeYF"

  DBURL: "mysql.db:3306"

  KAFKA\_BROKER: "my-kafka.kafka:9092"

  DBNM: "capstonedb" #데이터베이스명

  KAFKAID: "capstone" #카프카 토픽 아이디

#siege 생성(http api)

kubectl apply -f - <<EOF

apiVersion: v1

kind: Pod

metadata:

name: siege

spec:

containers:

- name: siege

image: apexacme/siege-nginx

EOF

#siege 실행(http api)

kubectl exec -it siege파드명 -- bash

#mysql 실행

kubectl exec -it kafka파드명 -- bash

#mysql 데이터베이스 생성 및 사용

show databases;

use capstonedb;

show tables;

select \* from 테이블명(smartbanking)

#siege 부하설정

wget <https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.5.0/components.yaml>

kubectl create -f components.yaml

kubectl top pod

#부하발생

siege -c250 -t10000S -v --content-type "application/json" 'http://login:8080/logins/validate/hana/1234'

siege -c250 -t10000S -v --content-type "application/json" 'http://preapply:8080’

#오토스케일 발생

siege -c250 -t10000S -v --content-type "application/json" 'http://preapply:8080'

kubectl autoscale deployment preapply --min 1 --max 5 --cpu-percent 20x