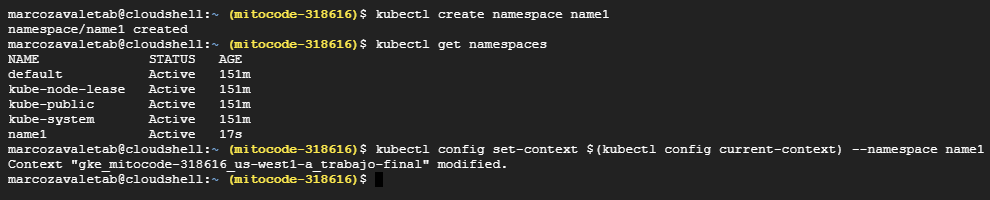
RESOLUCION EJERCICIO 2

1.-Crear el namespace : "name1"

kubectl create namespace name1

kubectl get namespaces

kubectl config set-context $(kubectl config current-context) --namespace name1

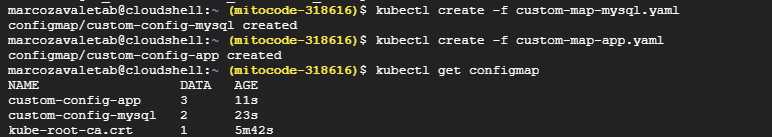


2.- Creacion de los configMap :

kubectl create -f custom-map-mysql.yaml

kubectl create -f custom-map-app.yaml

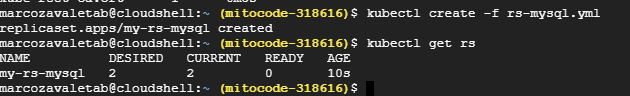
kubectl get configmap



3.- Para la base de datos se creara un replicaSet(2 replicas) y un service de tipo clusterIp para exponer la bd :

kubectl create -f rs-mysql.yml

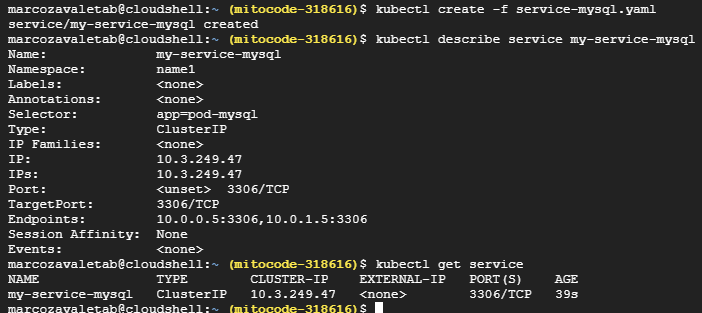
kubectl get rs



kubectl create -f service-mysql.yaml

kubectl describe service my-service-mysql

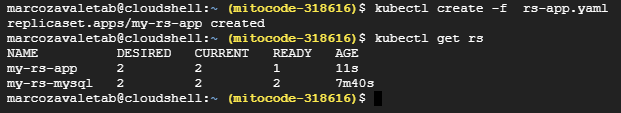
kubectl get services



4.- Para la aplicacion java se creara un replicaSet(2 replicas) y un service de tipo clusterIp para exponer la app :

kubectl create -f rs-app.yaml

kubectl get rs

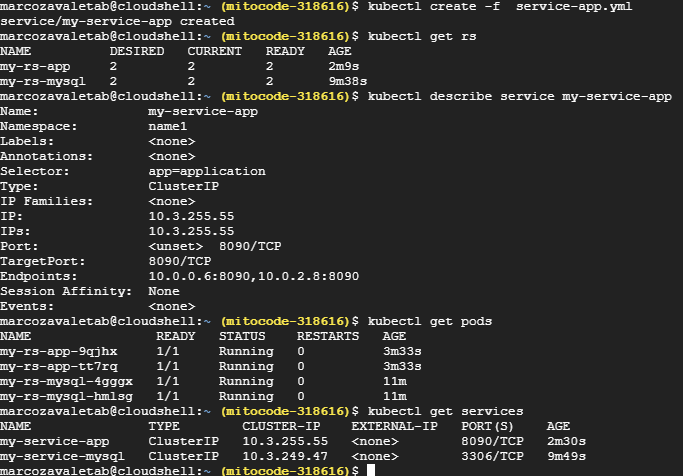


kubectl create -f service-app.yml

kubectl describe service my-service-app

kubectl get services

kubectl get pods



5.-Pruebas de funcionamiento mediante pod test :

kubectl run -i --tty test --image=alpine -- sh

kubectl attach test -c test -i -t

apk --no-cache add curl

curl --location --request GET 'http://my-service-app:8090/list'

curl --location --request POST 'http://my-service-app:8090/create?name=banana'

curl --location --request POST 'http://my-service-app:8090/create?name=manzana'

curl --location --request POST 'http://my-service-app:8090/create?name=uva'

curl --location --request POST 'http://my-service-app:8090/create?name=pera'

curl --location --request GET 'http://my-service-app:8090/list'

