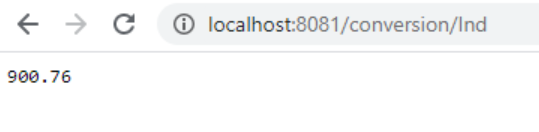
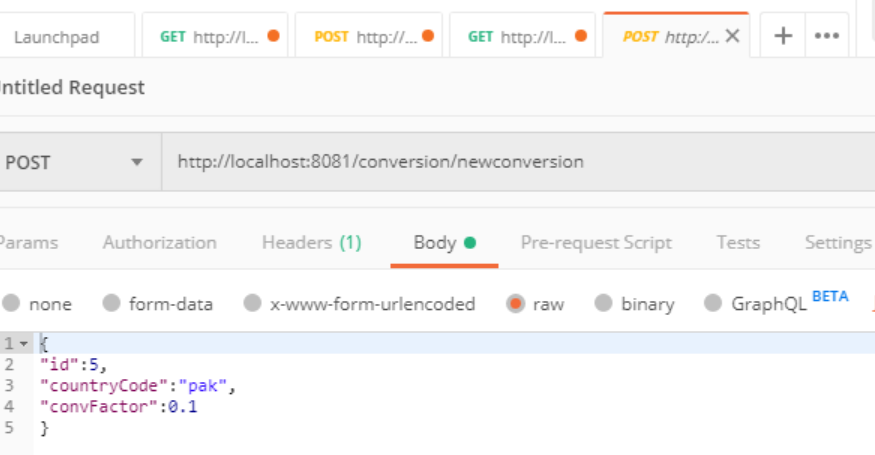
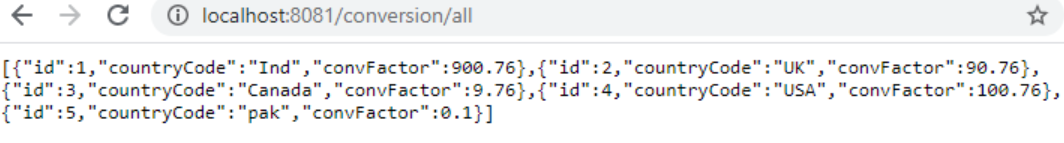
1. Finding by country code



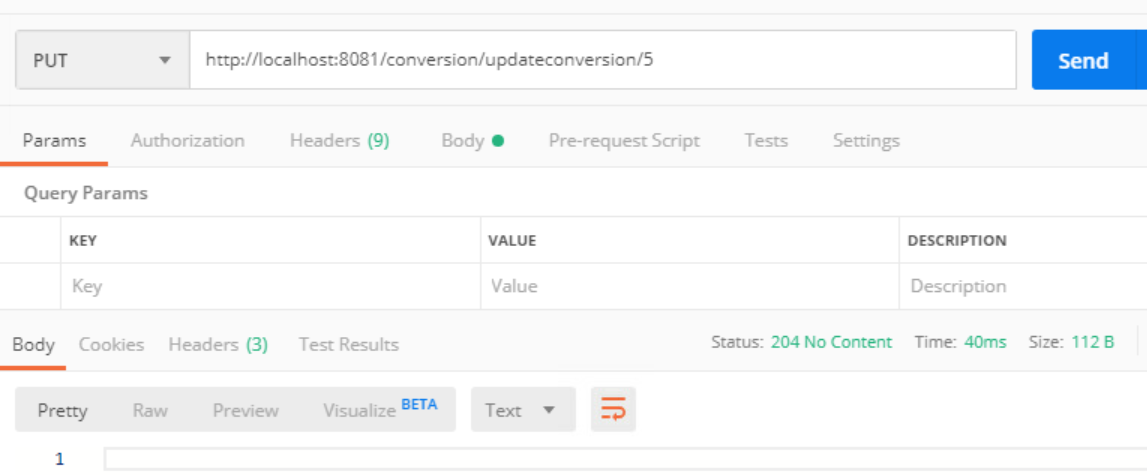
2. Adding the data

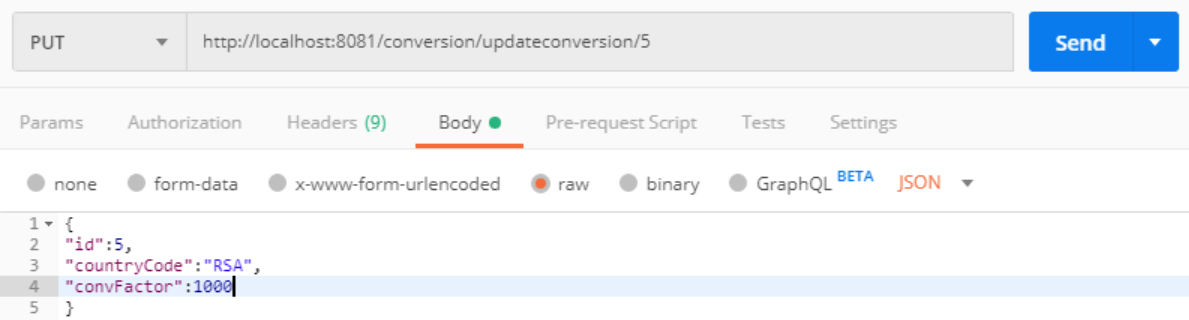


3. Finding all data

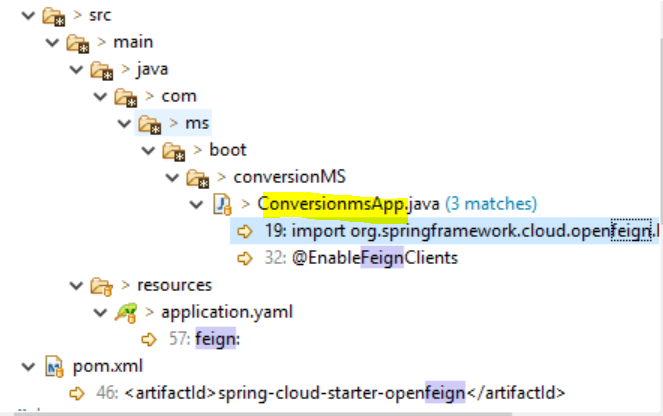


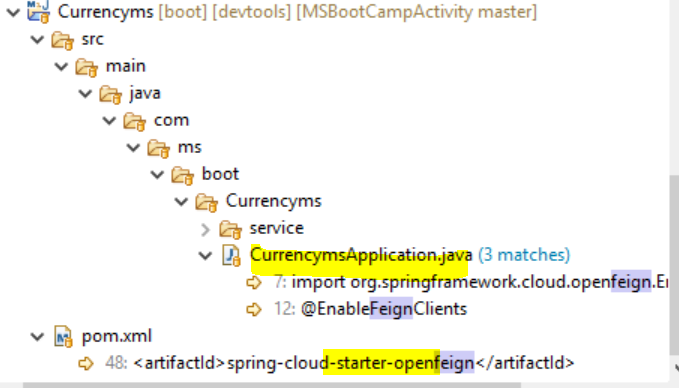
4. Updating the data



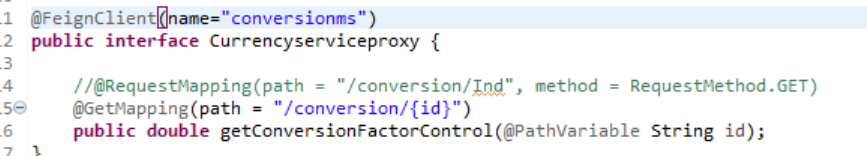


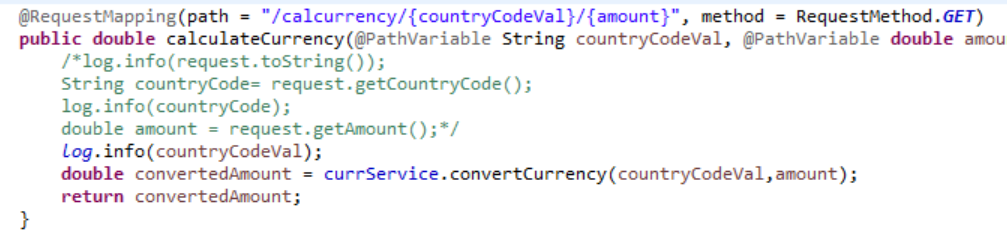
5. Feign client



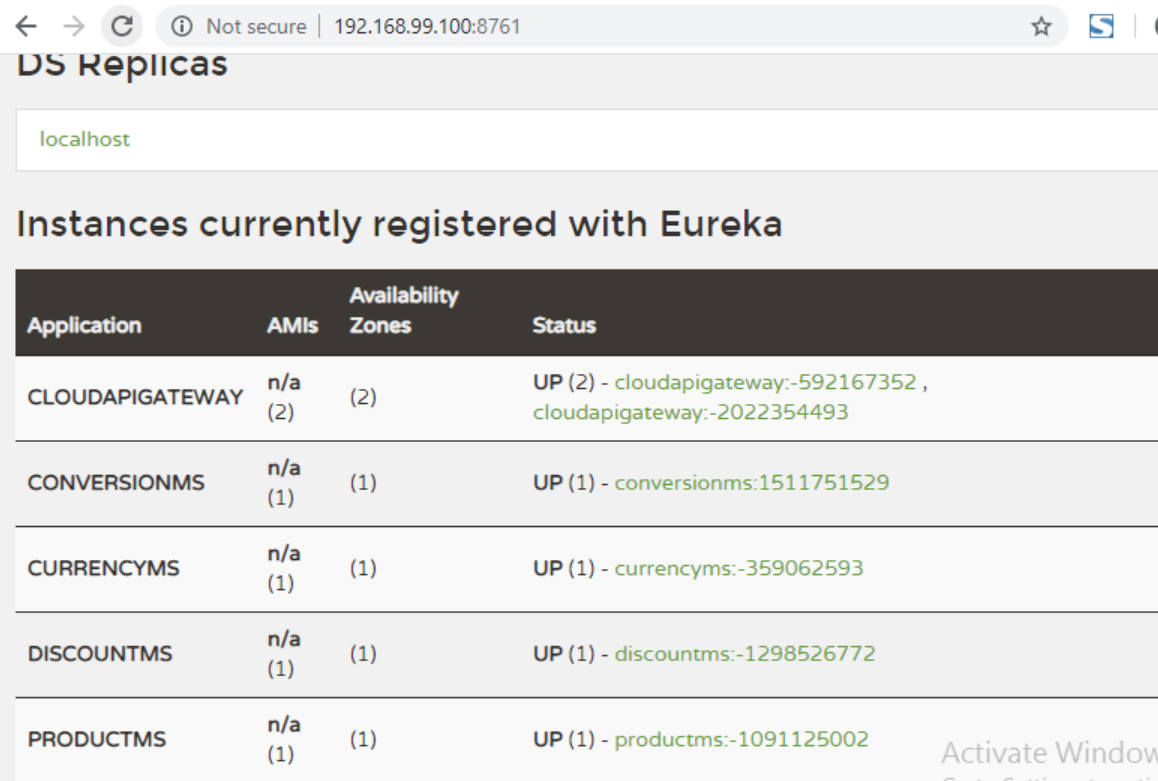


Feign Client Configuration

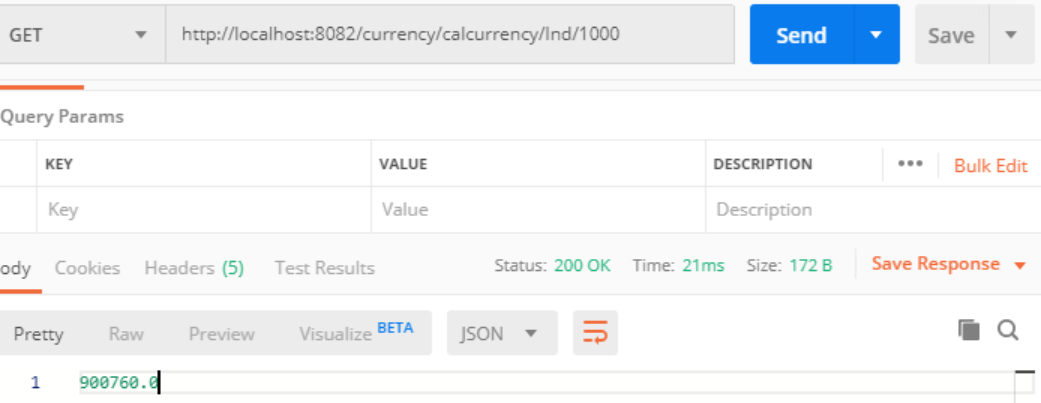




6. Eureka



7. Calculating the currency



**RUNNING IN DOCKER**

docker-machine env docker-vm1

@FOR /f "tokens=\*" %i IN ('docker-machine env docker-vm1') DO @%i

1. CONFIG SERVER Path

mvn clean install

docker build -t mkdockerhub1310/msbootcampactivity\_2020:config1 .

docker run -p 8888:8888 -d --name configkol1 --network ms-chassis-nw mkdockerhub1310/msbootcampactivity\_2020:config1

1. EUREKA SERVER path

mvn clean install

docker build -t mkdockerhub1310/msbootcampactivity\_2020:eureka1 .

docker run -p 8761:8761 -d --name eurekaMs --network ms-chassis-nw mkdockerhub1310/msbootcampactivity\_2020:eureka1

1. CLOUDAPI Path

mvn clean install

docker build -t mkdockerhub1310/msbootcampactivity\_2020:cloudapigateway .

docker run -p 9092:9092 -d --name cloudapigatewaykol --network ms-chassis-nw mkdockerhub1310/msbootcampactivity\_2020:cloudapigateway

1. CONVERSION MS

mvn clean install

docker build -t mkdockerhub1310/msbootcampactivity\_2020:conversionms .

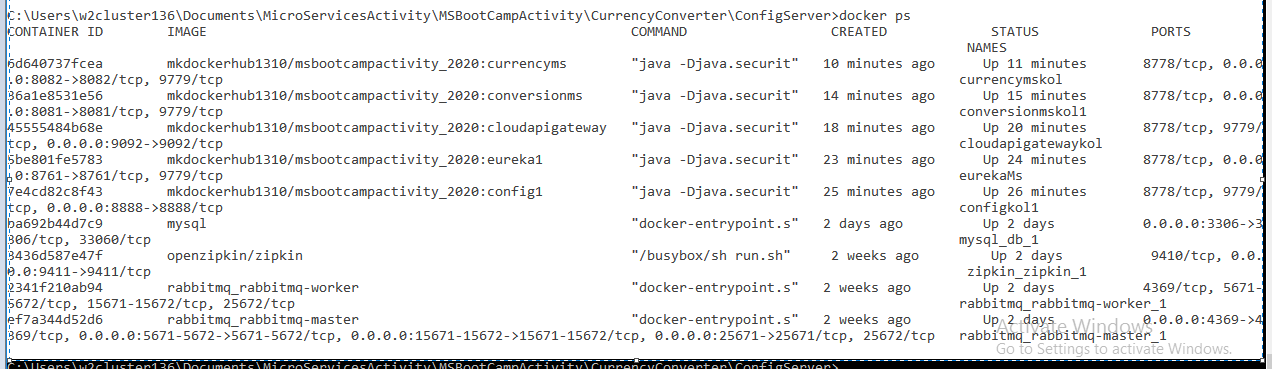
docker run -p 8081:8081 -d --name conversionmskol1 --network ms-chassis-nw mkdockerhub1310/msbootcampactivity\_2020:conversionms

1. CURRENCY MS  
   mvn clean install

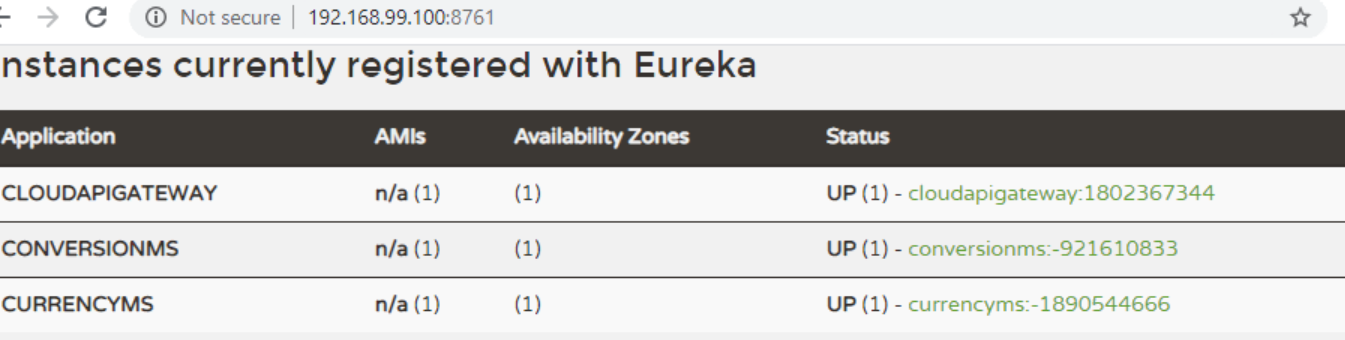
docker build -t mkdockerhub1310/msbootcampactivity\_2020:currencyms .

docker run -p 8082:8082 -d --name currencymskol --network ms-chassis-nw mkdockerhub1310/msbootcampactivity\_2020:currencyms

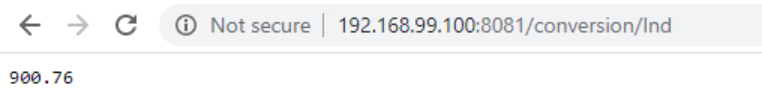
**docker ps**

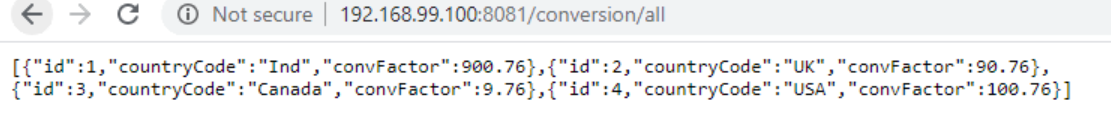


Eureka: <http://192.168.99.100:8761/>

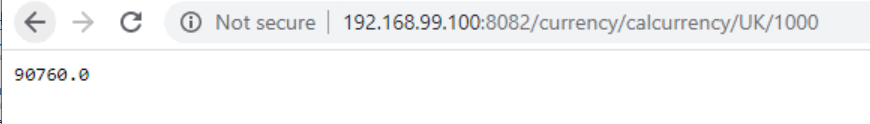


CONVERSIONMS: <http://192.168.99.100:8081/conversion/Ind>

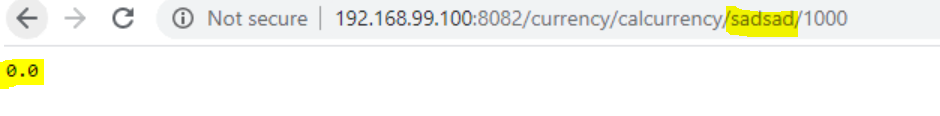




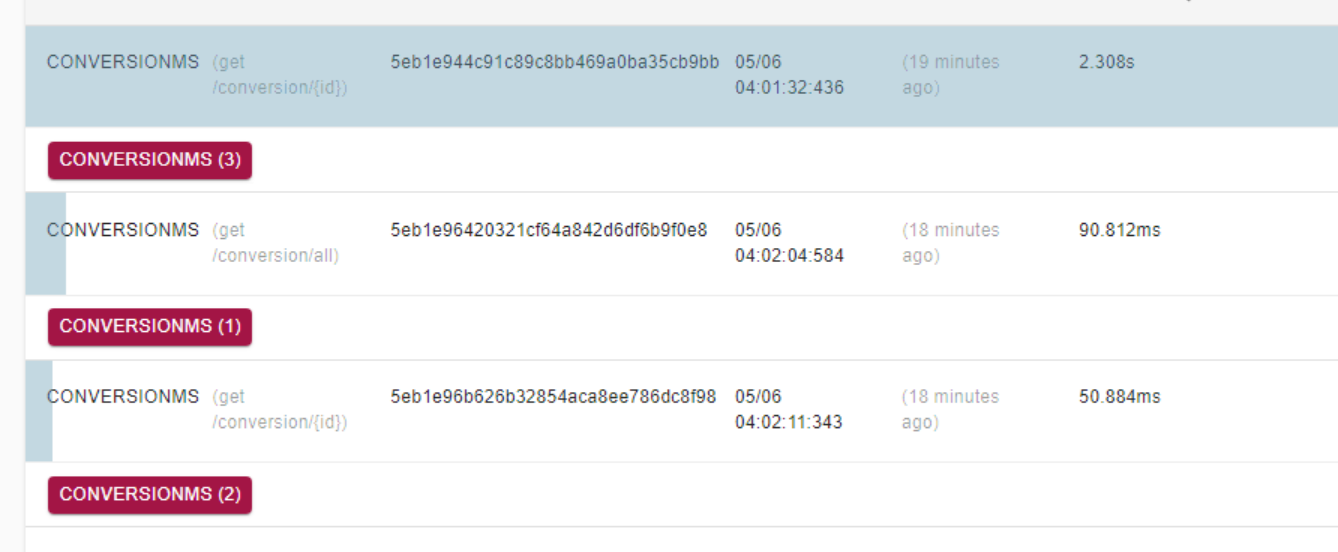
Currencyms: <http://192.168.99.100:8082/currency/calcurrency/UK/1000>



Fallback



Zipkins



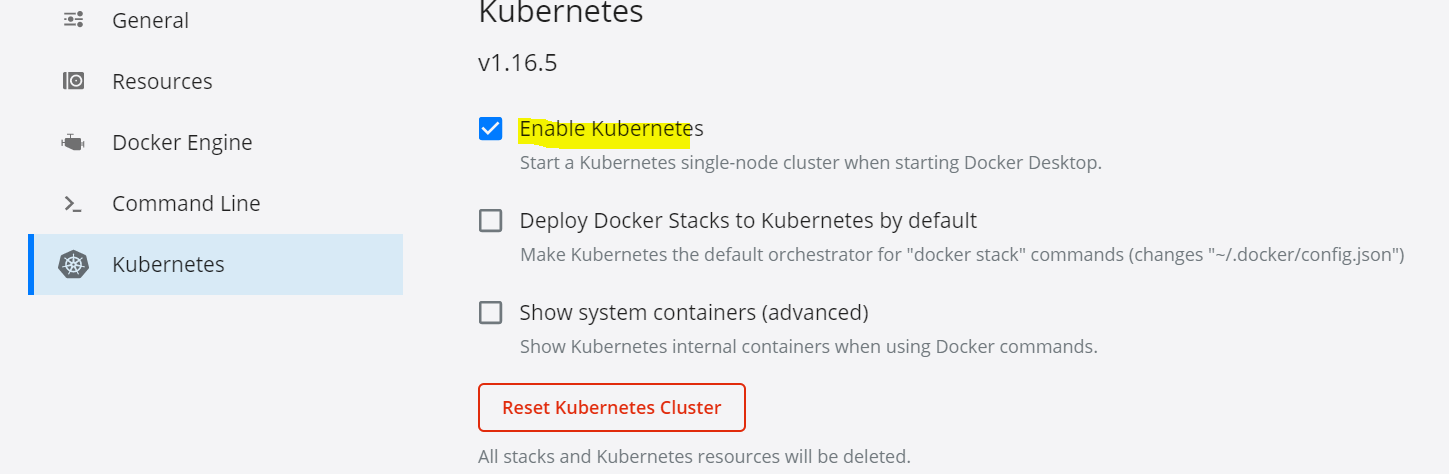
**Setup Kubernetes Cluster and Deploy the images in Kubernetes cluster.**

Tool used: Docker Desktop.

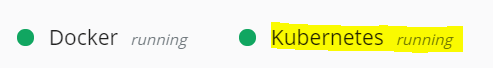
**1. Create Kubernetes Environment**

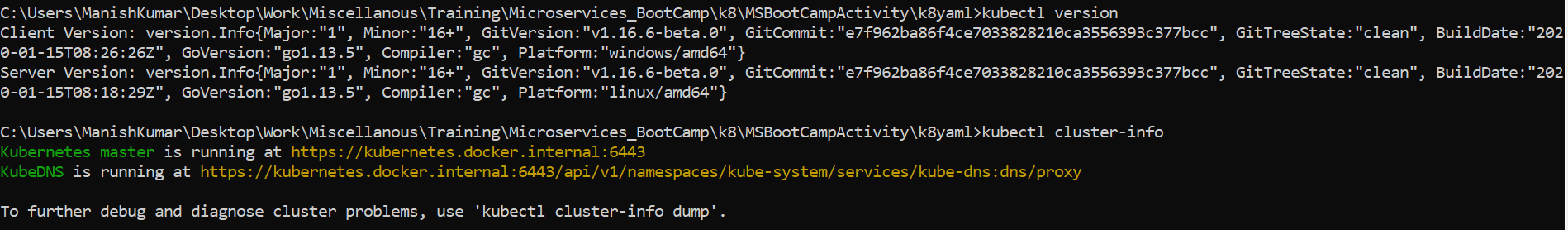
- Install the Docker Desktop.

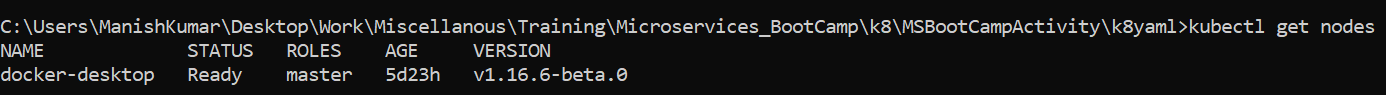
- Run the Kubernetes by enabling it from the setting



-Once Kubernetes is running we will get below icons in Docker Desktop







**2. Deploy the applications in the Kubernetes cluster.**

- Docker hub comes with single node and after enabling the Kubernetes a cluster is created.

- Create the yaml files for the microservices application and deploy the application.

**Sample Yaml file: eureka1.yaml**

*apiVersion: apps/v1*

*kind: Deployment*

*metadata:*

*name: eureka1*

*namespace: default*

*spec:*

*replicas: 1*

*selector:*

*matchLabels:*

*app: eureka1*

*template:*

*metadata:*

*labels:*

*app: eureka1*

*spec:*

*containers:*

*- name: eureka1*

*image: "mkdockerhub1310/msbootcampactivity\_2020:eureka1"*

The container image is referenced as : *mkdockerhub1310/msbootcampactivity\_2020:eureka1.*

>kubectl apply -f eureka1.yaml

deployment.apps/eureka1 created

> kubectl apply -f config1.yaml

deployment.apps/ config1 created

> kubectl apply -f cloudapigateway.yaml

deployment.apps/ cloudapigateway created

> kubectl apply -f conversion.yaml

deployment.apps/ conversionms created

> kubectl apply -f currency.yaml

deployment.apps/ currencyms created

For creating a service. Separate yaml file is created.

eureka-services.yaml

---------------------------

*apiVersion: v1*

*kind: Service*

*metadata:*

*name: eureka1*

*labels:*

*app: eureka1*

*spec:*

*selector:*

*app: eureka1*

*type: NodePort*

*ports:*

*- nodePort: 31003*

*port: 8761*

*targetPort: 8761*

\*\*Note: *We can create the service during deployment time by adding the above snippet in the eureka1.yaml. Similarly, other services yaml files are created and applied to create the service.*

kubectl apply -f eureka-service.yaml

service/eureka1 created

kubectl apply -f config1-service.yaml

service/config1 created

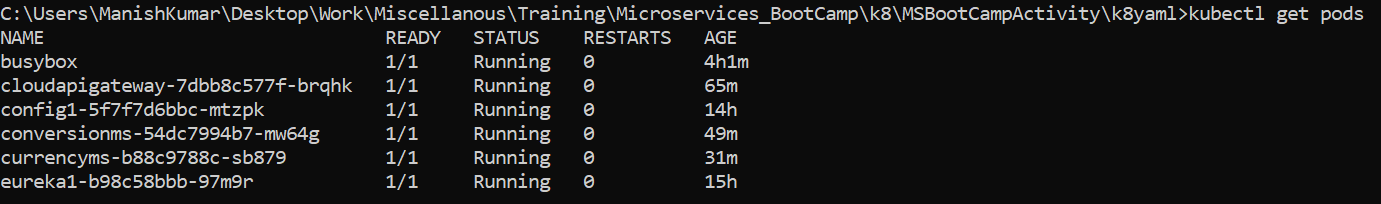
kubectl apply -f conversion-service.yaml

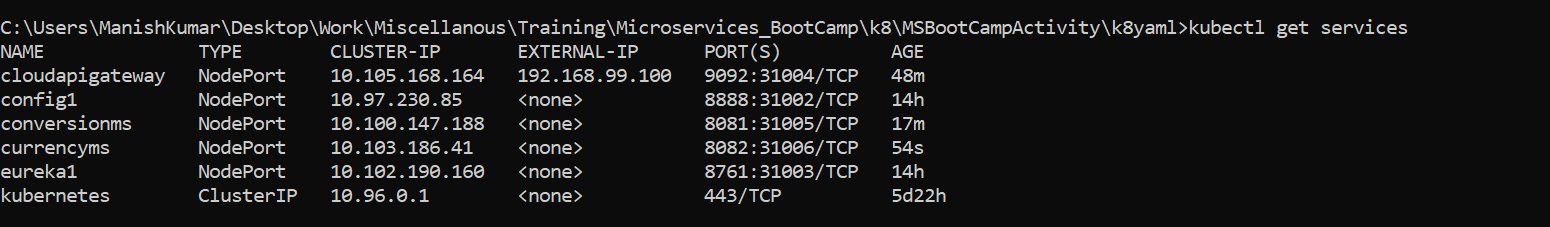
service/conversionms created

kubectl apply -f currency-service.yaml

service/currencyms created

**3. Checking the Pods and Services created.**



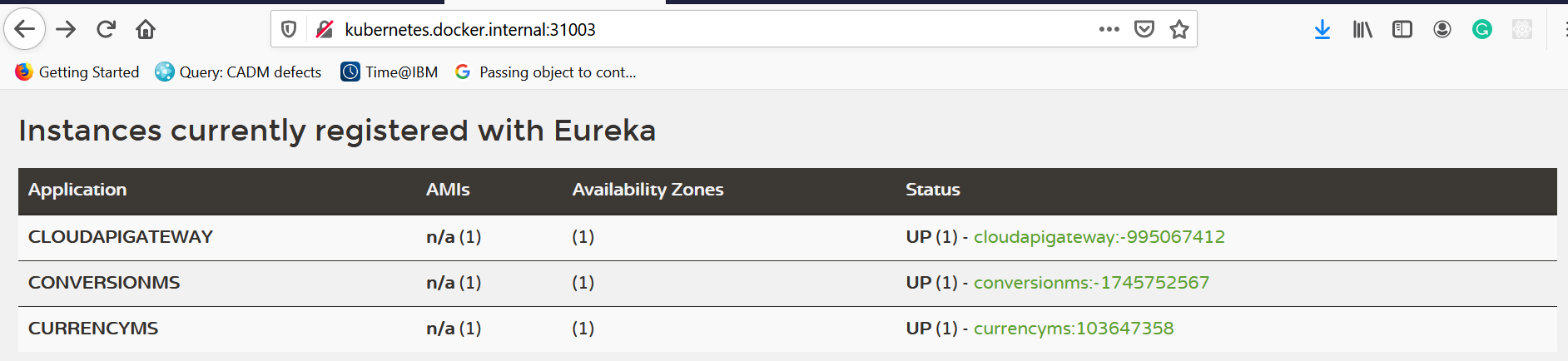


**4. Verifying the microservices applications**

***Eureka***

***---------***

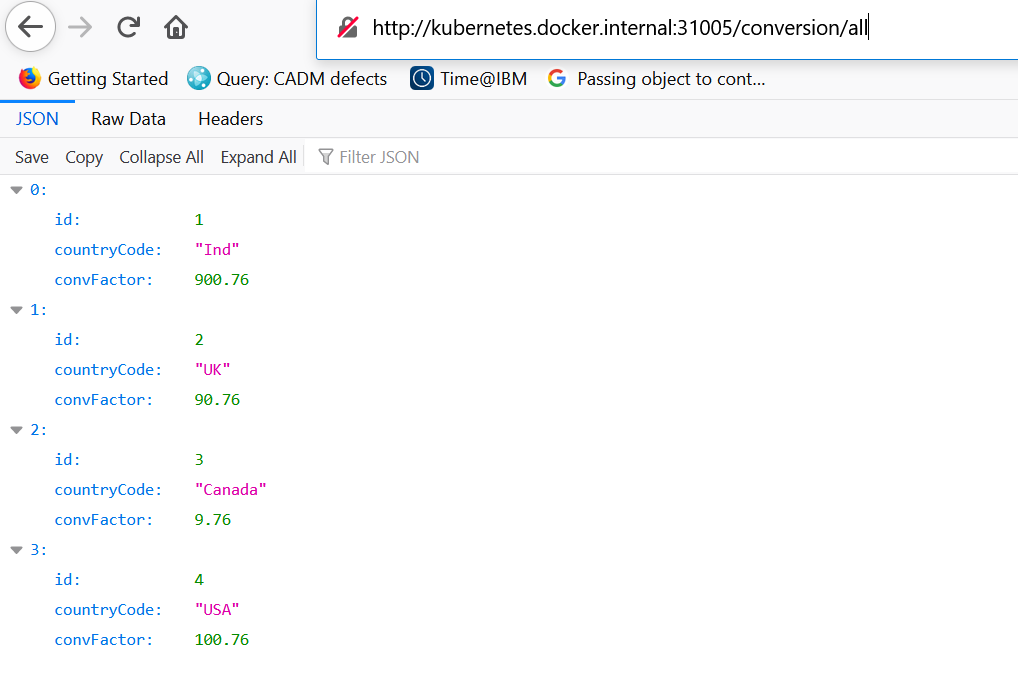
<http://kubernetes.docker.internal:31003/>



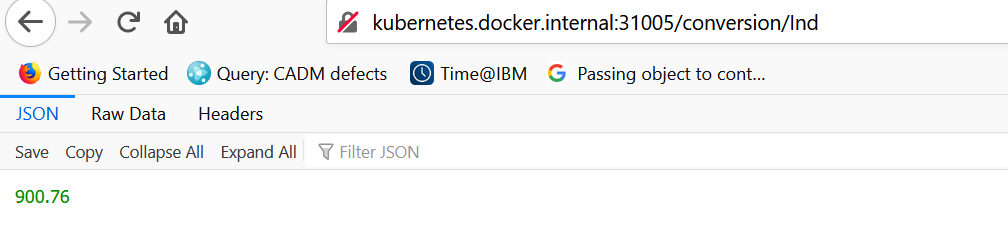
***Conversion ms***

***--------------------***

a. <http://kubernetes.docker.internal:31005/conversion/all>

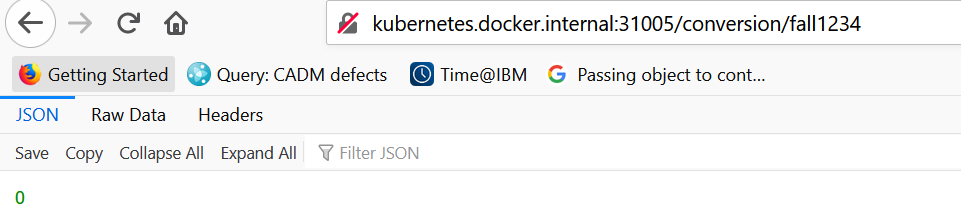


b. <http://kubernetes.docker.internal:31005/conversion/Ind>



c. Fallback scenario

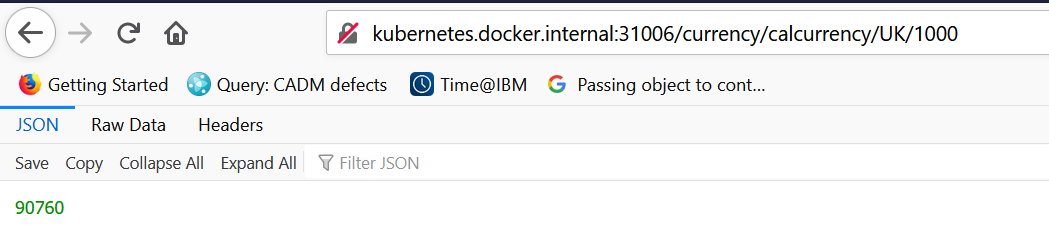
<http://kubernetes.docker.internal:31005/conversion/fall1234>



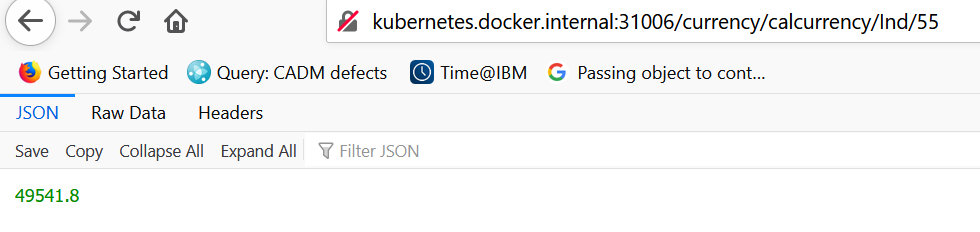
***Currencyms***

***---------------***

<http://kubernetes.docker.internal:31006/currency/calcurrency/UK/1000>



<http://kubernetes.docker.internal:31006/currency/calcurrency/Ind/55>



\*\* Note: Changes were made in the application.yaml files of cloudapigateway, conversionms and currencyms to point to the Eureka instance running on Kubernetes cluster.

defaultZone: ${cloudapigateway.eureka.url:http://kubernetes.docker.internal:31003/eureka}

defaultZone: ${conversionms.eureka.url:http://kubernetes.docker.internal:31003/eureka}

defaultZone: ${currencyms.eureka.url:http://kubernetes.docker.internal:31003/eureka}

Kubernetes Yaml Files: <https://github.com/manishkumar0302/MSBootCampActivity/tree/master/k8yaml>