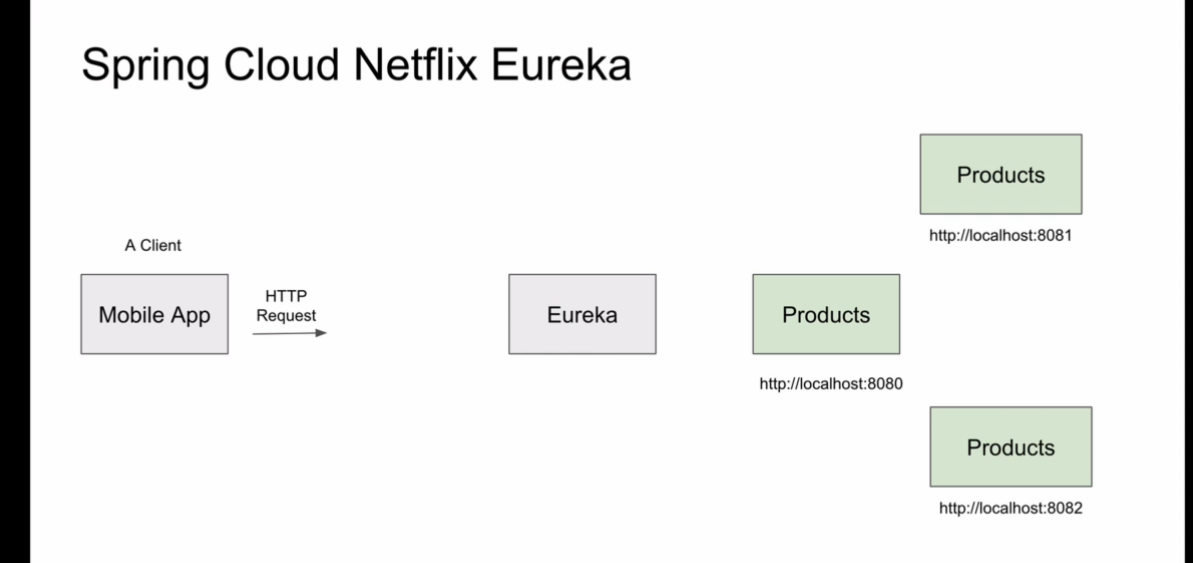
***Spring Cloud and Micro services:***

A single mobile application connect to 1 BE server

But what if we have multiple instance of that server

Then how the client server would knows all the service of the product instance on

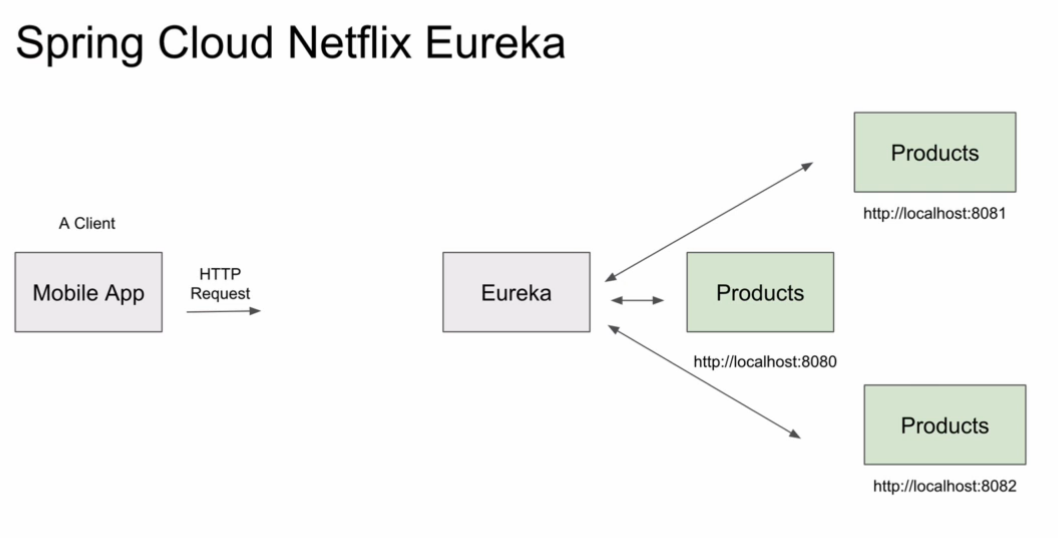
1. Different IP + Port



Eureka will know the address of all the micro services and it will act as a service discovery and it’s instances to the client with one

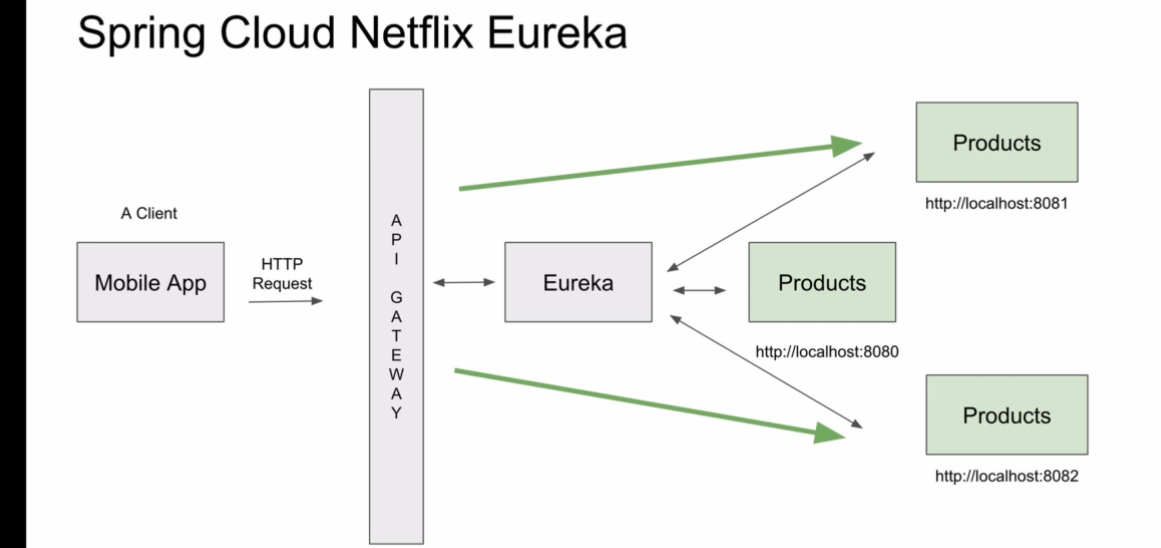
https://{IP\_ADDDRESS}:{PORT}/{EndPoint}

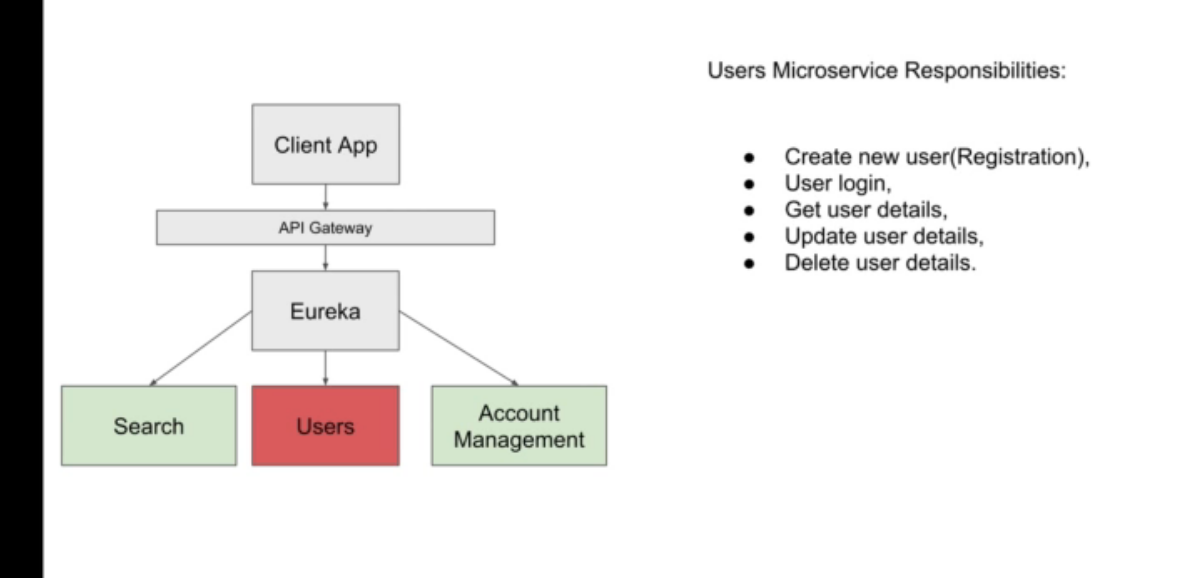
for eureka server!!



Basically we can say that eureka server is a API\_GATEWAY

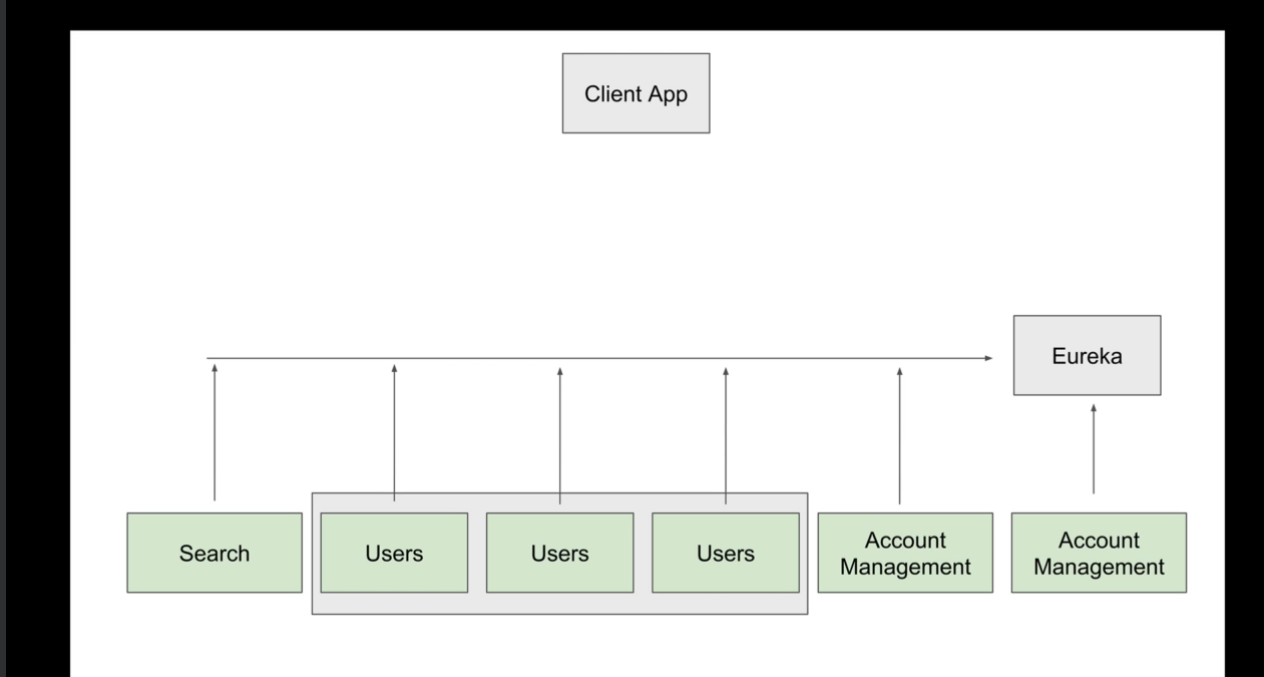
Given in the below diagram





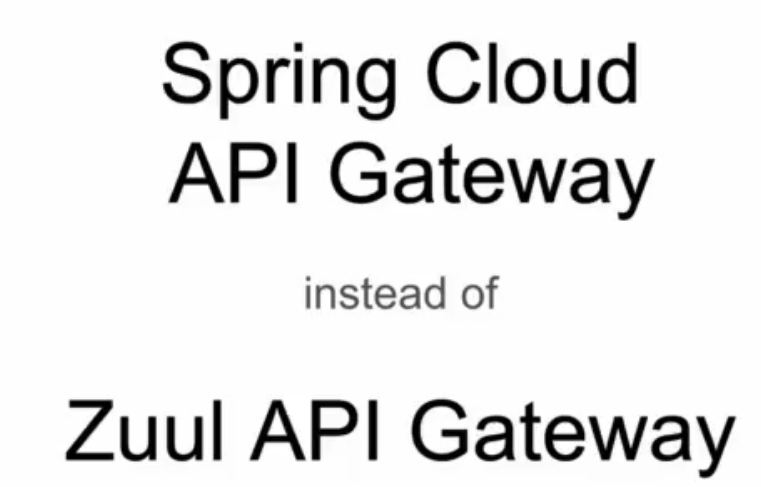
This is the architecture diagram of user micro services

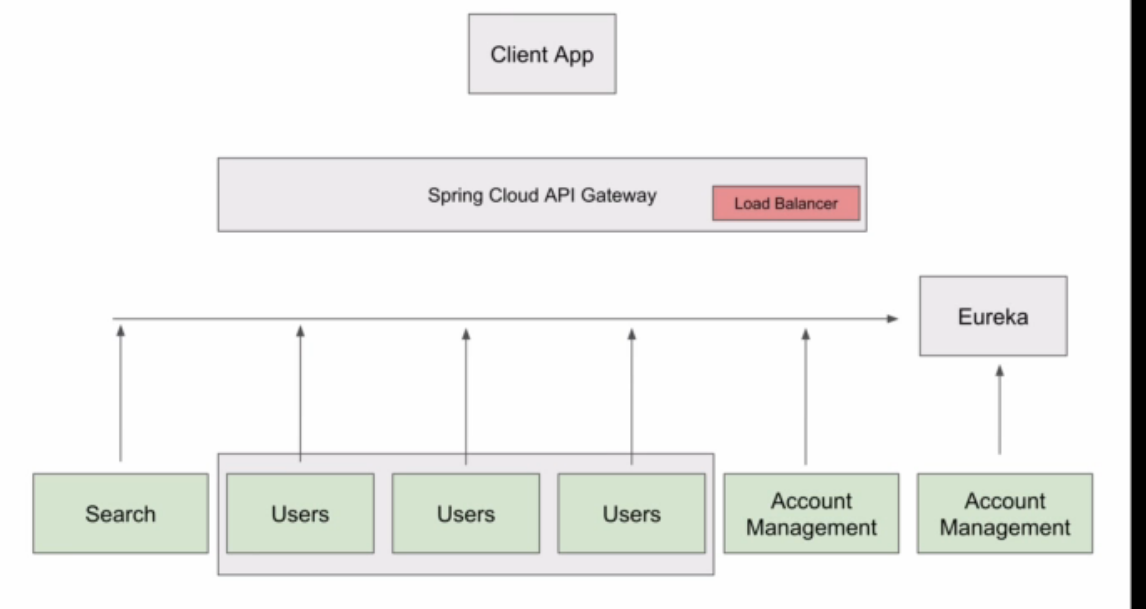
1. We will learn how to make micro services and how to register with eureka:



In this Eureka is registering with eurka server for service availability

1. We are going to go ahead with Spring Cloud Api gateway in Built

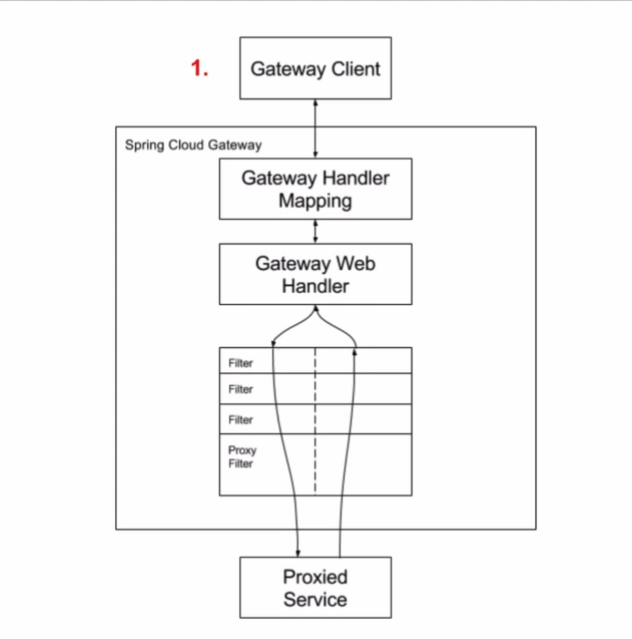




API GATEWAY within built load balancer to distribute the load around the different instance of the services for HA.

1. We can impl custom filter
2. We can impl the custom route for services

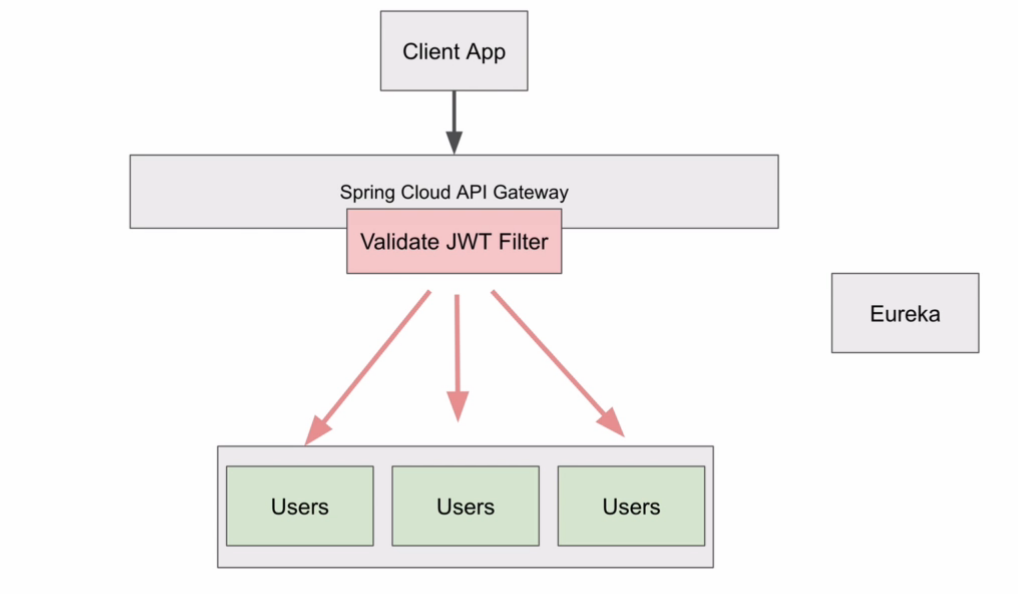
# How Spring Cloud API – Gateway Works internally



1. Gateway handler receive the request and gateway map handler will check the configure routed for the particular api hit
2. Gate way web handler will allow request to filter chain

There two arrows from proxied service one is going down and other one is going up

So after all the filter chain pass the request get back to the web handler.



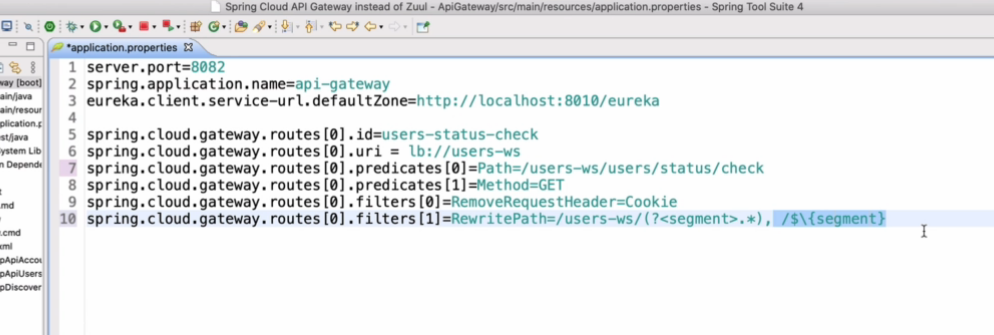
Api gateway filter the request for JWT and route the request to the particular services

Validate JWT filter if valid then request will pass otherwise it will not pass through the request

By Introducing the Spring cloud api gate way we can define the end point and entry point of execution for all the services and it will work as a router who will route from api gateway to the exact services

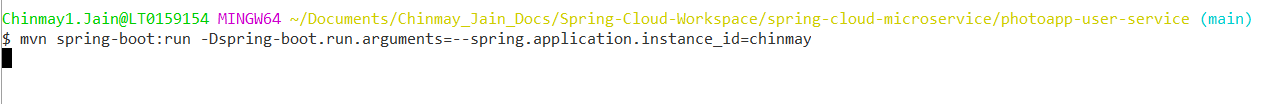
Instead of using **Eureka Discovery services Ip we can use the api Gateway End-point**

**Defining the Route in the configuration file is as follows :**

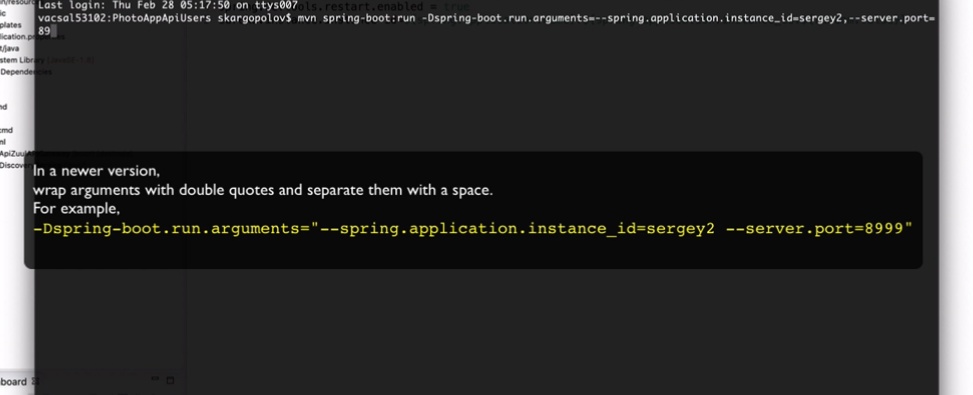
****

1. Route is a array type we can have any number of routes in the project
2. Route[0].id = will define the unique id for different services
3. Route[0].predicate[0]=path=”Whole path of the entire user api call”

To run from terminal and pass the argument

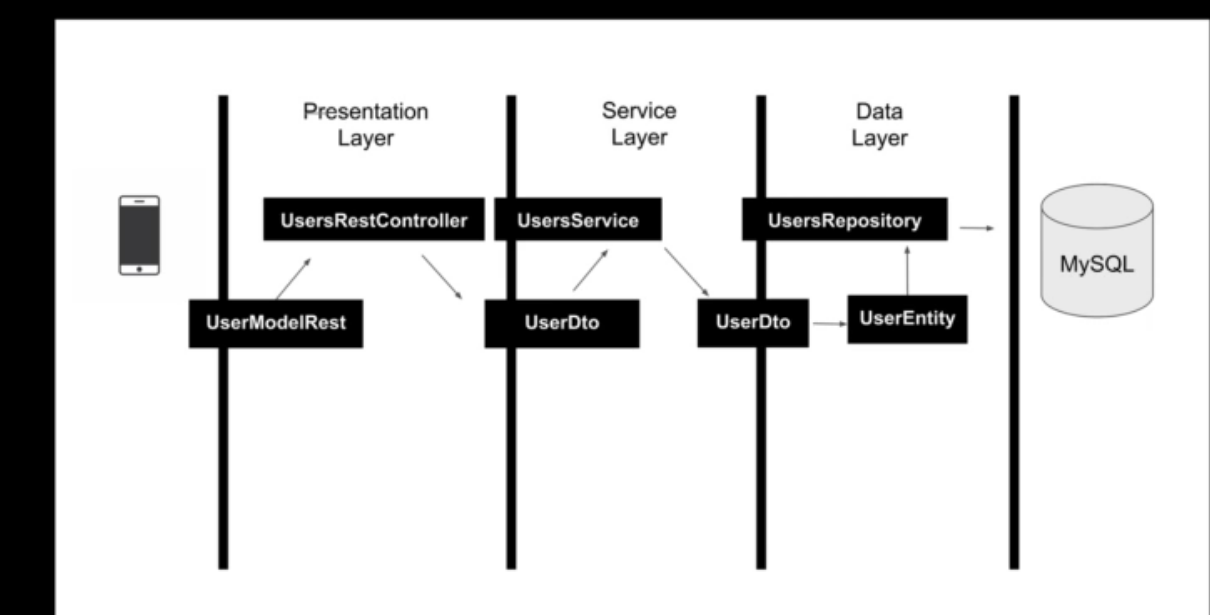


This one is for mvn cmd



For gradle we can use the cmd :

***./gradlew bootRun -Pargs='--eureka.instance.instance-id=yourValue'***

******