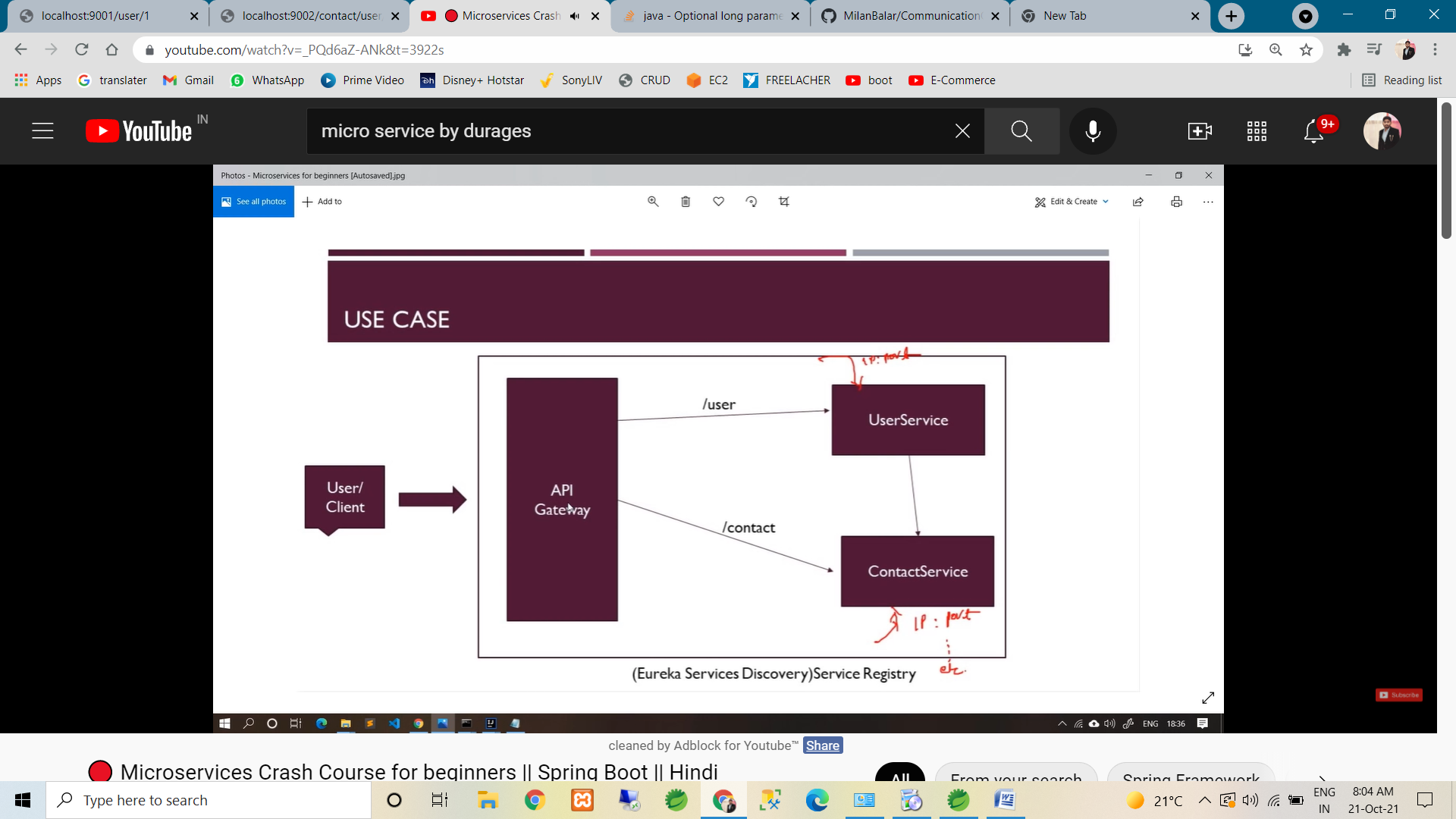
Micro Services With Boot

* Step 1 : create separate one rest api with port 9001 (i.e user microservice)
* Step 2: create separate one other rest api with port 9002 (i.e contact microservice)
* Step 3 : now for communication of both the service we need of RestTemplate
* so first of all declare the RestTemplate as bean in main file, so we can Autowired the our RestTemplate in user service
* using the RestTemplate we can communicate the both microservices



* Follow the above diagram for communication.
* For the communication of two microservice code is commit in git with repo name is :

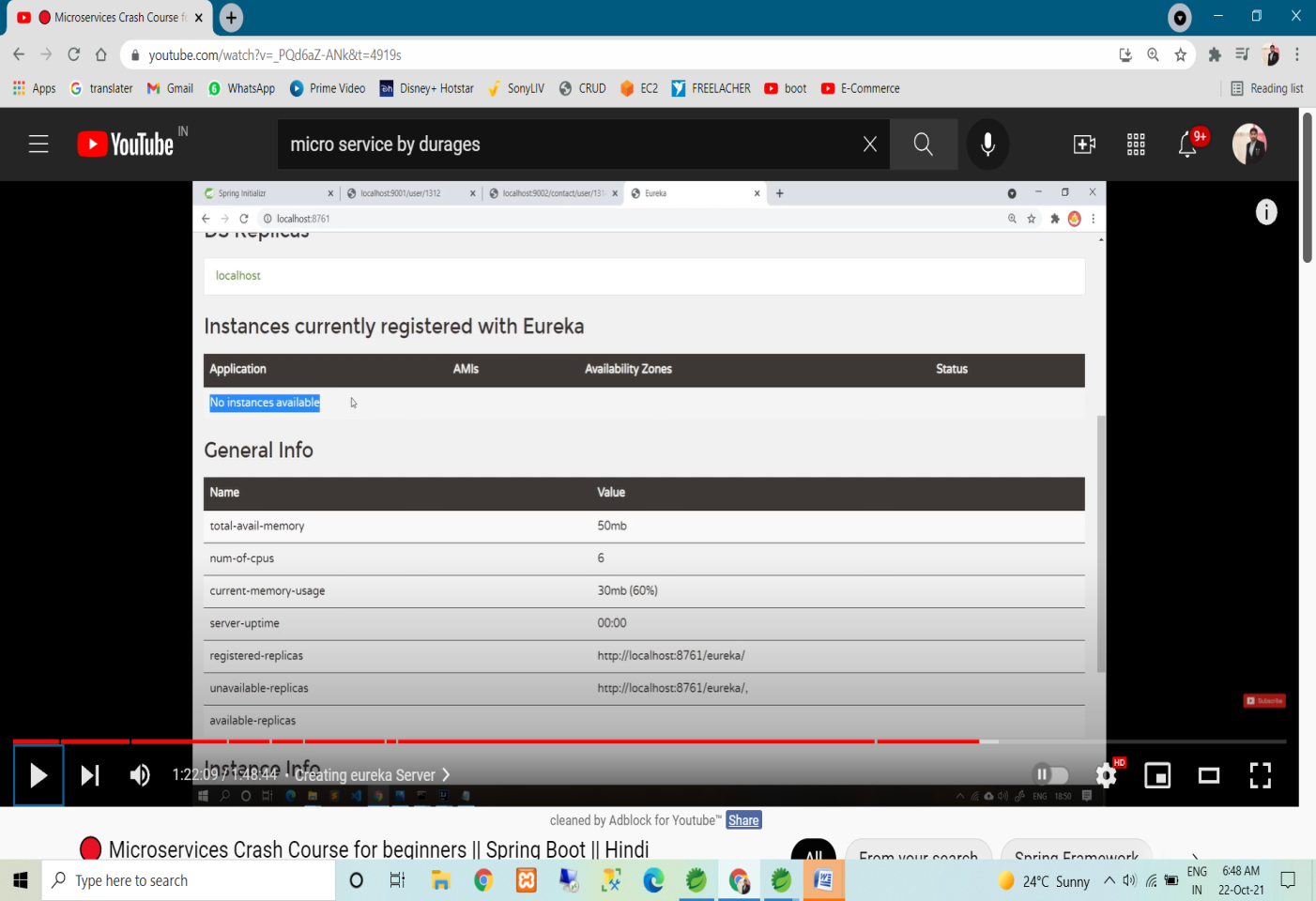
<https://github.com/MilanBalar/CommunicationOfTwoMicroServiceBoot_v1.git>

* Step 4 : now we create Eureka service discovery (Eureka server)
* Now to create the our both micro service to client & register in to Eureka server
* The main purpose of Eureka server & client is our all client(microservices) register on one common place
* Then we create API gateway
* API gate way is use for :- we not need to call every services with is url & port.
* API gate way is endpoint of microservice. So using api gate way url we can call the our all the micro services.
* Step : 5 now we create Eureka server. It’s same as simple boot project.
* Crate spring boot project.
* Pom.xml in require dependency is: Eureka Server
* In main class we need to apply annotation : @EnableEurekaServer
* Application.properies : server.port =8761 (Eureka default port number)

: eureka.client.register-with-eureka=false (:. Means it’s indicate this application it self not register in Eureka).

: eureka.client.fetchRegistry : false

: eureka.server.waitTimeInMsWhenSyncEmpty: 0 (:. It’s just waiting time)

* If we fire the localhost:8761 so we can get the Eureka UI.
* Here we can see, no instance is are available. Because of still we not create any client .show the following attachment.
* 
* So now we create the our both service as client.
* First we crate contact service as client
* We required Eureka discovery client dependency

-🡪 so go to spring initialize (like we create new project)& search the Eureka discovery client dependency dependency.

-🡪 now click on explore so we can get the dependency.

--🡪 following thing also need to copy

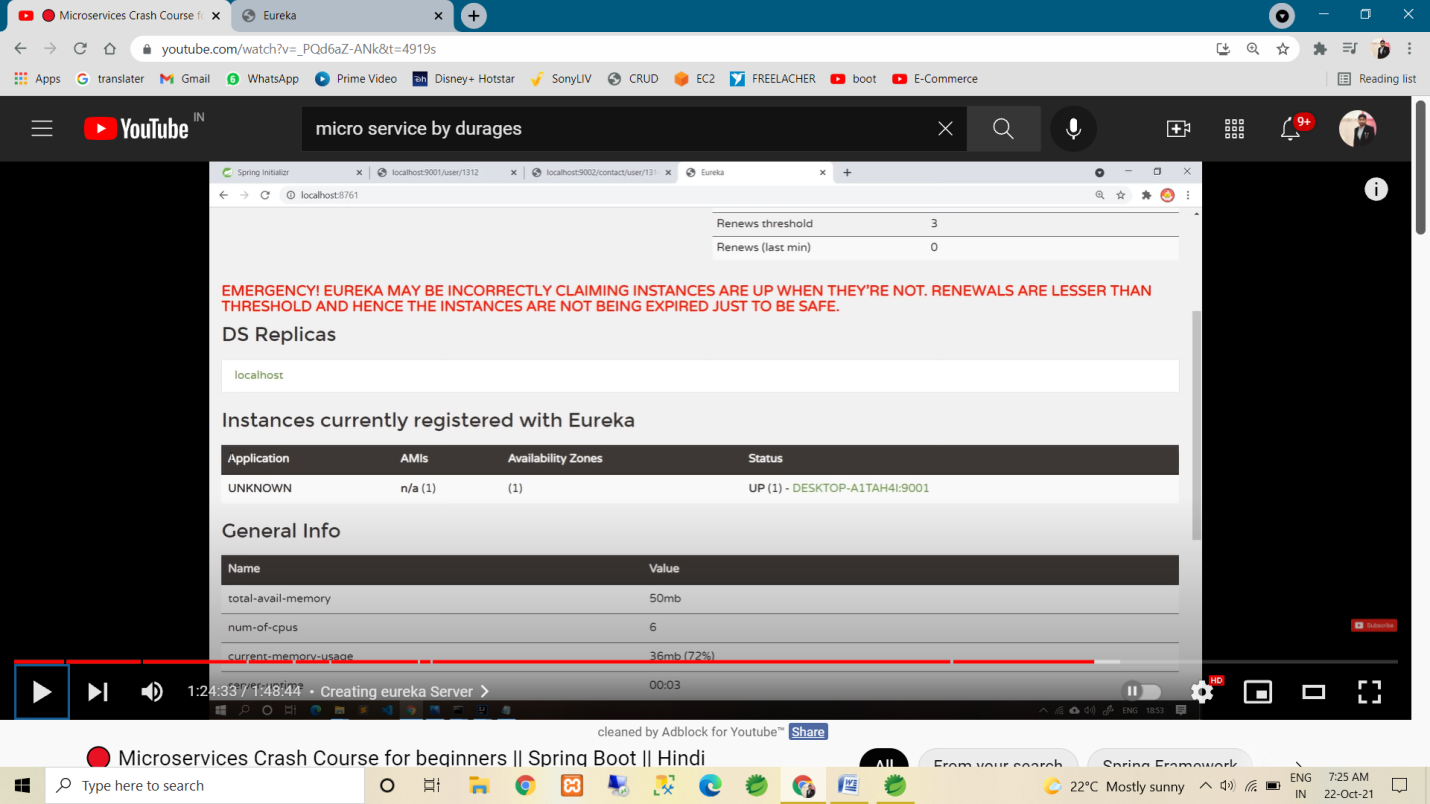
-> inside properties tag copy spring-cloud-version

-> & copy netfix-eurek-client dependency

-> & copy <dependencymanagement> whole tag

🡪 so now run the out project.

* If we check in eureka server**(:.http://localhost:8761/)** so we can got the instance of client service. Check the below image.



* Same step follow for create client to user service .
* As up the above pic application name is shown as UNKNOWN.
* So we to set the application name & IP (here above pic in show host name in place of IP in status tab)??

-🡪 first set in user-service

-> application.properties in

🡪 spring.appliction.name=user-service

-🡪 then run the code so now in Eureka server in we can

See the Application name as user-service.

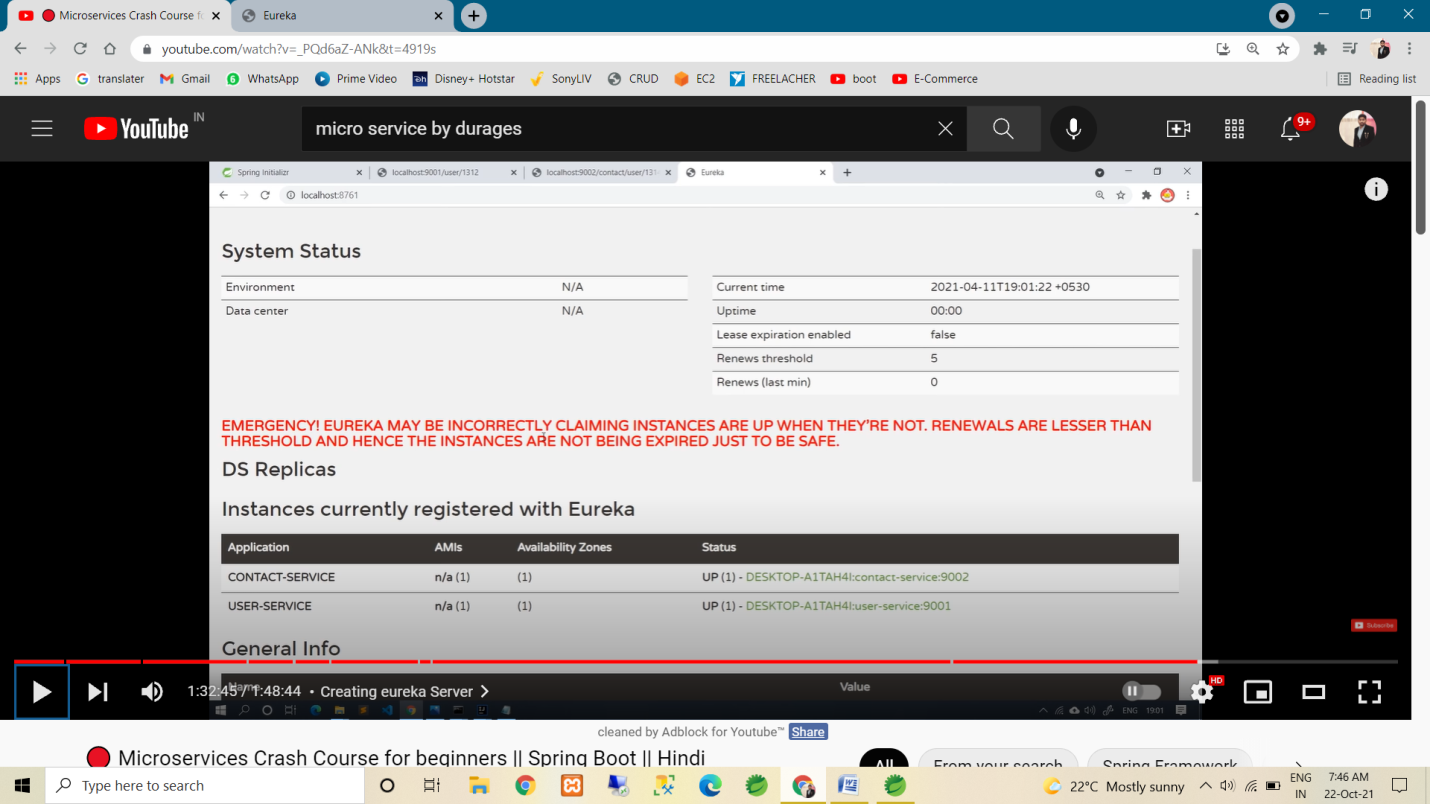
-🡪 now we change the host name in user-service(in above pic inside the status colume, we want to shown the Ip)

->application.properties in

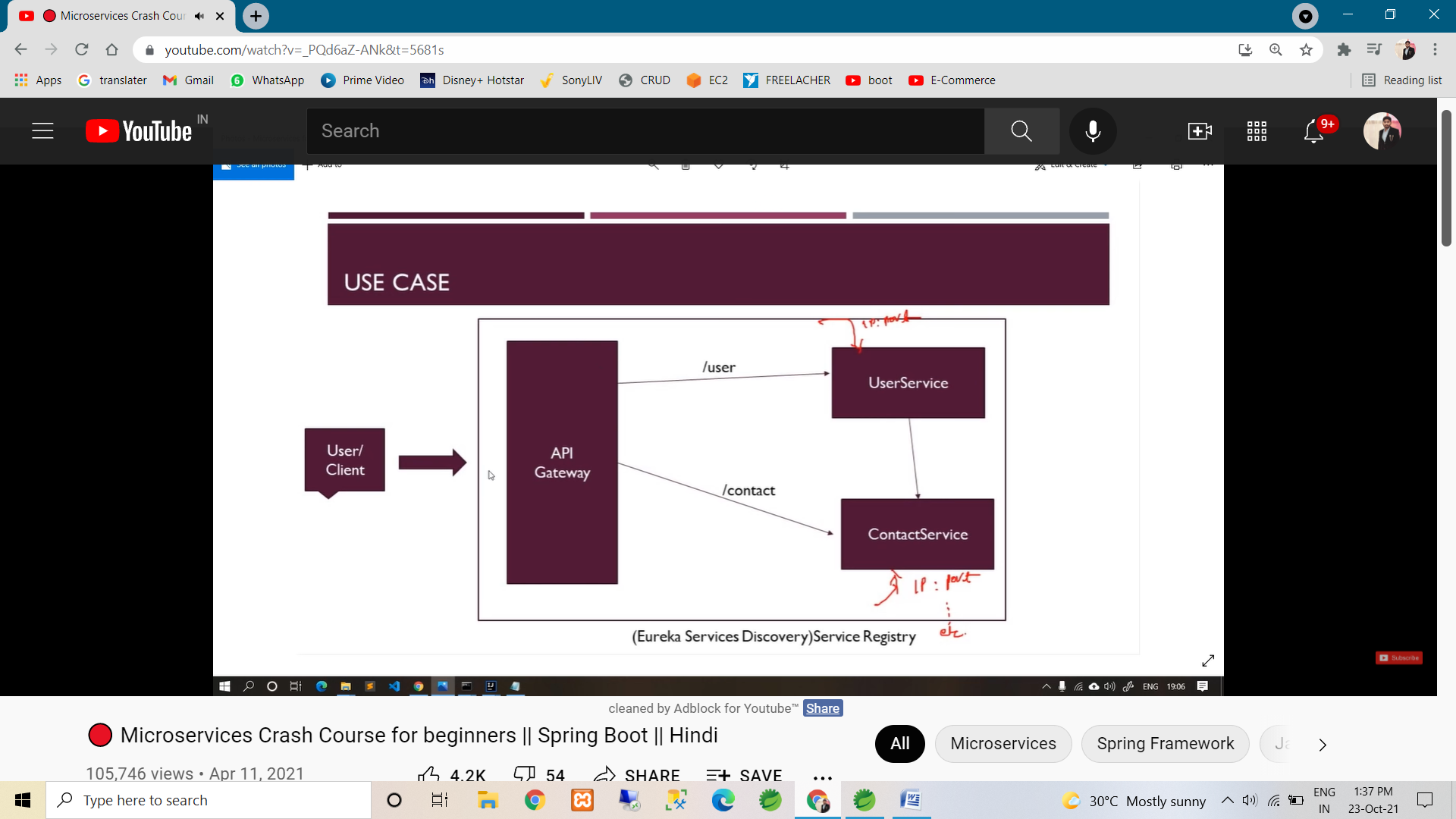
-> eureka.instance.hostname=localhost (:. Even we can also use this in Eureka server In properties).

-> same step apply in contact-service

🡪 now restart the all & show the below pic in Eureka server



* Now our both service(eureka-client) registered on eureka server.
* Now we not need to URL(:. /localhost:port/) for calling the services. We can call using the application name(:. Show above pic).
* Now we require one annotation(:. In user-service only because is call the other service) :- @Loadbalanced (:. Above the RestTemple Bean (in main method))
* If we not use this annotation so we can get the :- UnKnownHostException.
* So we need to load balancing , so it’s resolved.
* Now we implement API Gateway
* API Gateway is endpoint of microservice.
* Main motto of API gateway is we not need to call every service manually. API gateway handle it.
* Here we can alos handle login, authentication etc.
* Refer below pic.



* STEPS for crerating API Gateway
* Create spring boot application.
* Pom : Spring cloud routing dependency (:. We can also create using Zuul dependency). & Eureka Discovery client dependency (:. Bcoz it’s also Eureka clinet)

& Spring Boot Actuator (:. It’s optional. It’s use for manage & monitor our application)

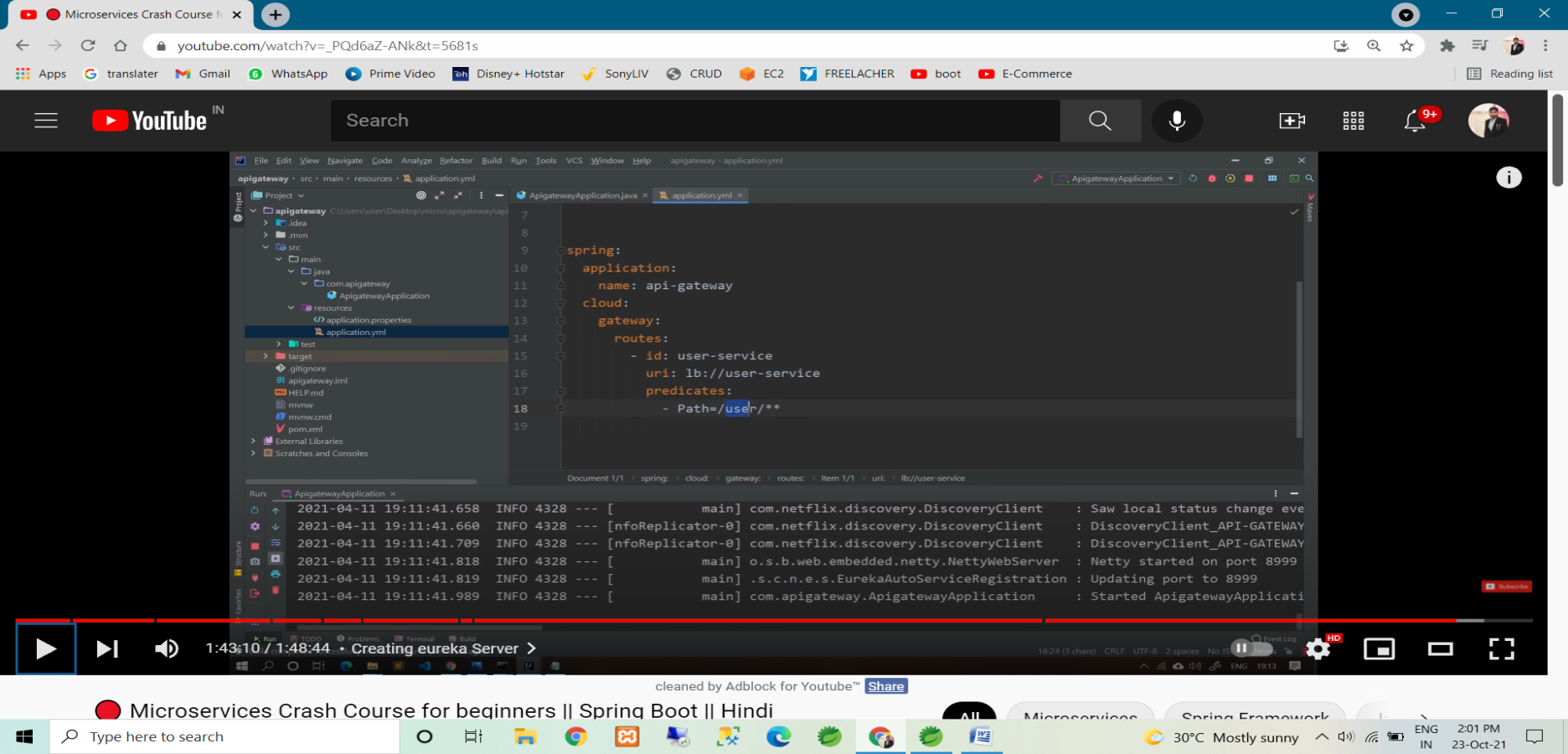
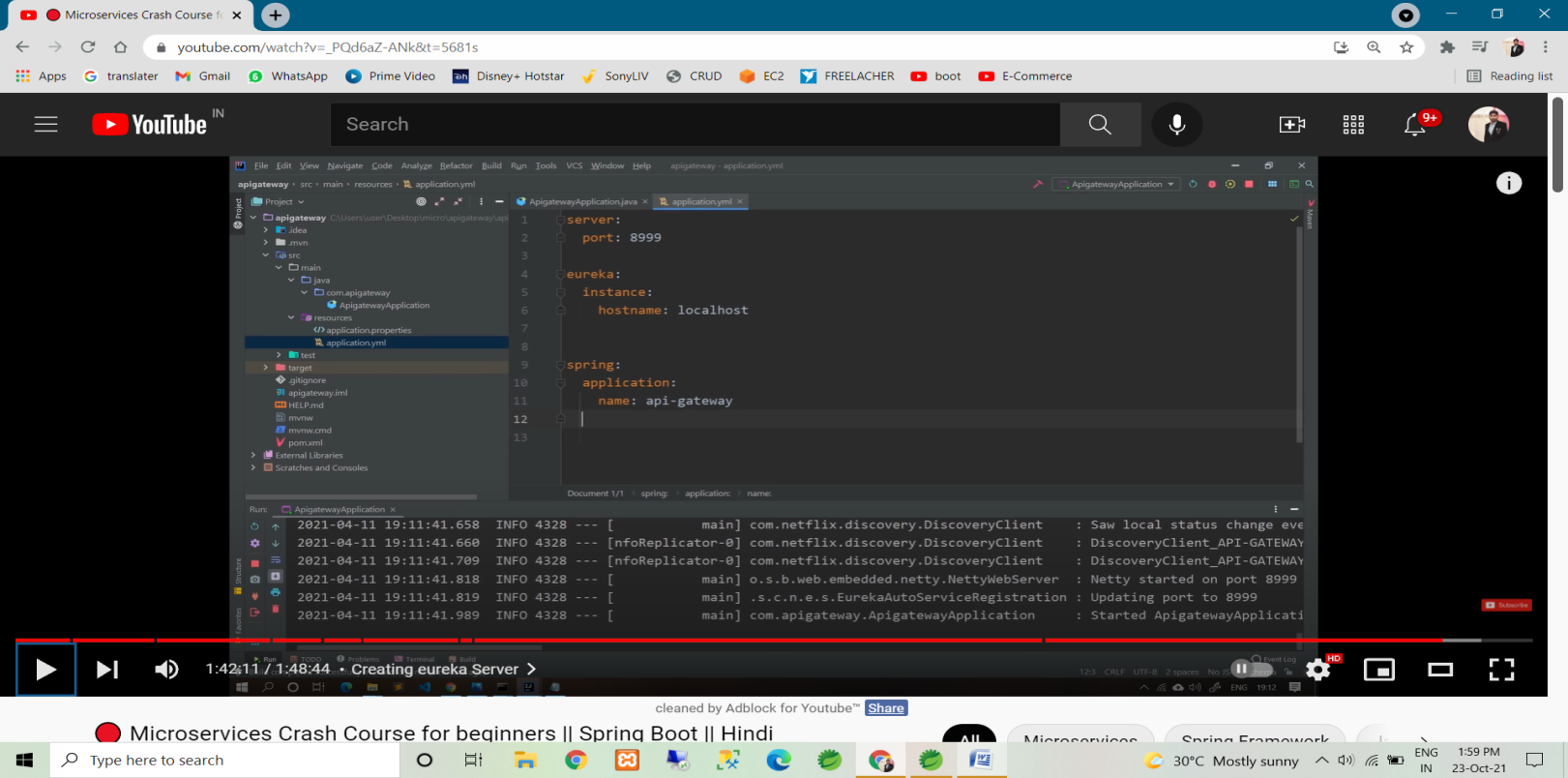
* As we know API gateway it’s also eureka client, so apply the @EnableEurekaClient in main method.
* Application.properties file in

--> server.port : 8999 (:. we can use any port nubmer).

🡪 eureka.instance.hostname: localhost

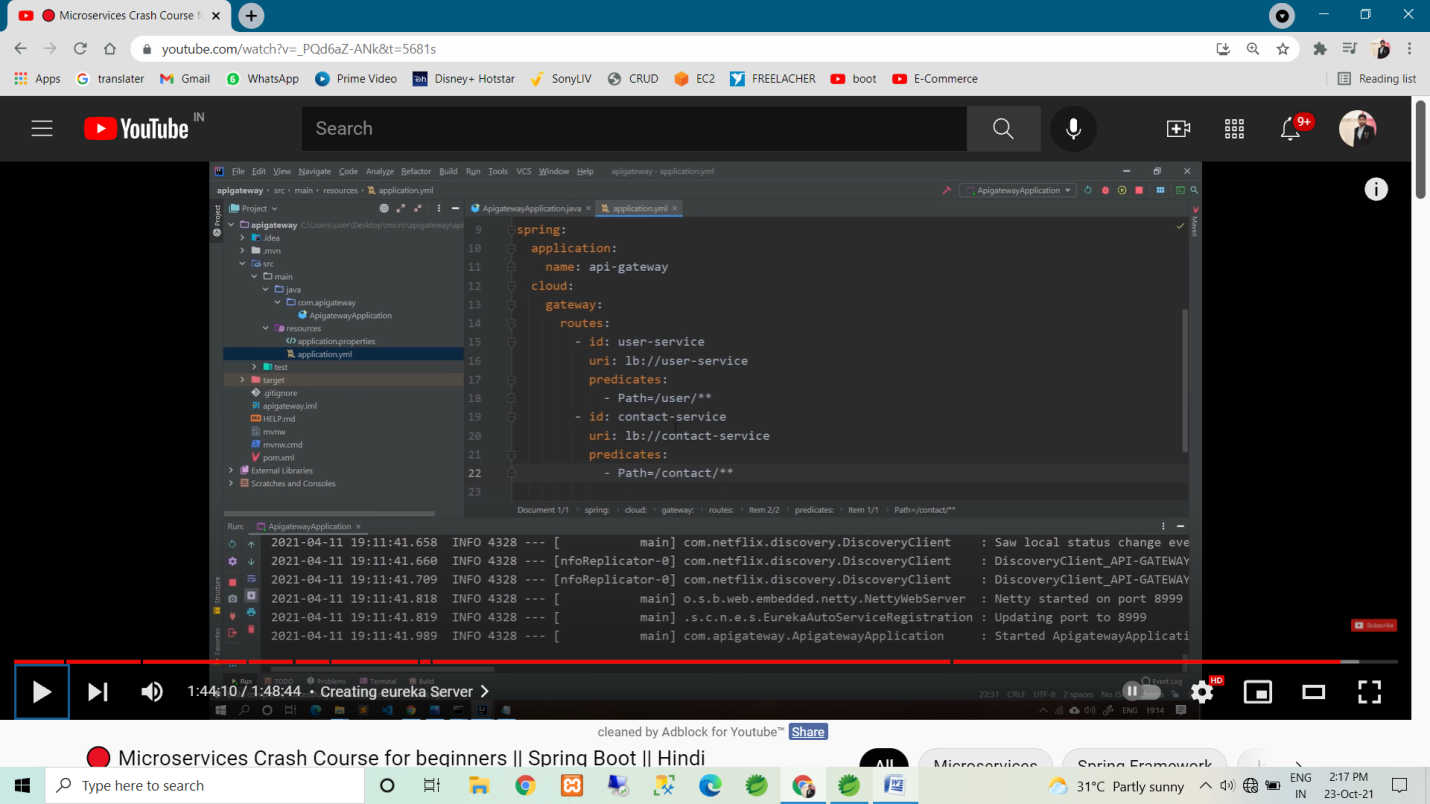
🡪 spring.application.name:api-gateway

& also refer the following application.yml(it’s same as application.properties, just we use ‘:’ symbol ) of API Gateway.



🡪above last pic in predicates indigate every request start’s with ‘/user/\*\*’ it’s forwarded into ‘user-service’ uri. & in uri ‘lb’ use for load balancing then use our microservice name.

* Refer the following pic for configure contact service in api gateway.



* So now we can access any microservice using api gateway url.
* i.e localhost:8999/user/2
* i.e localhost:8999/contact/user/4