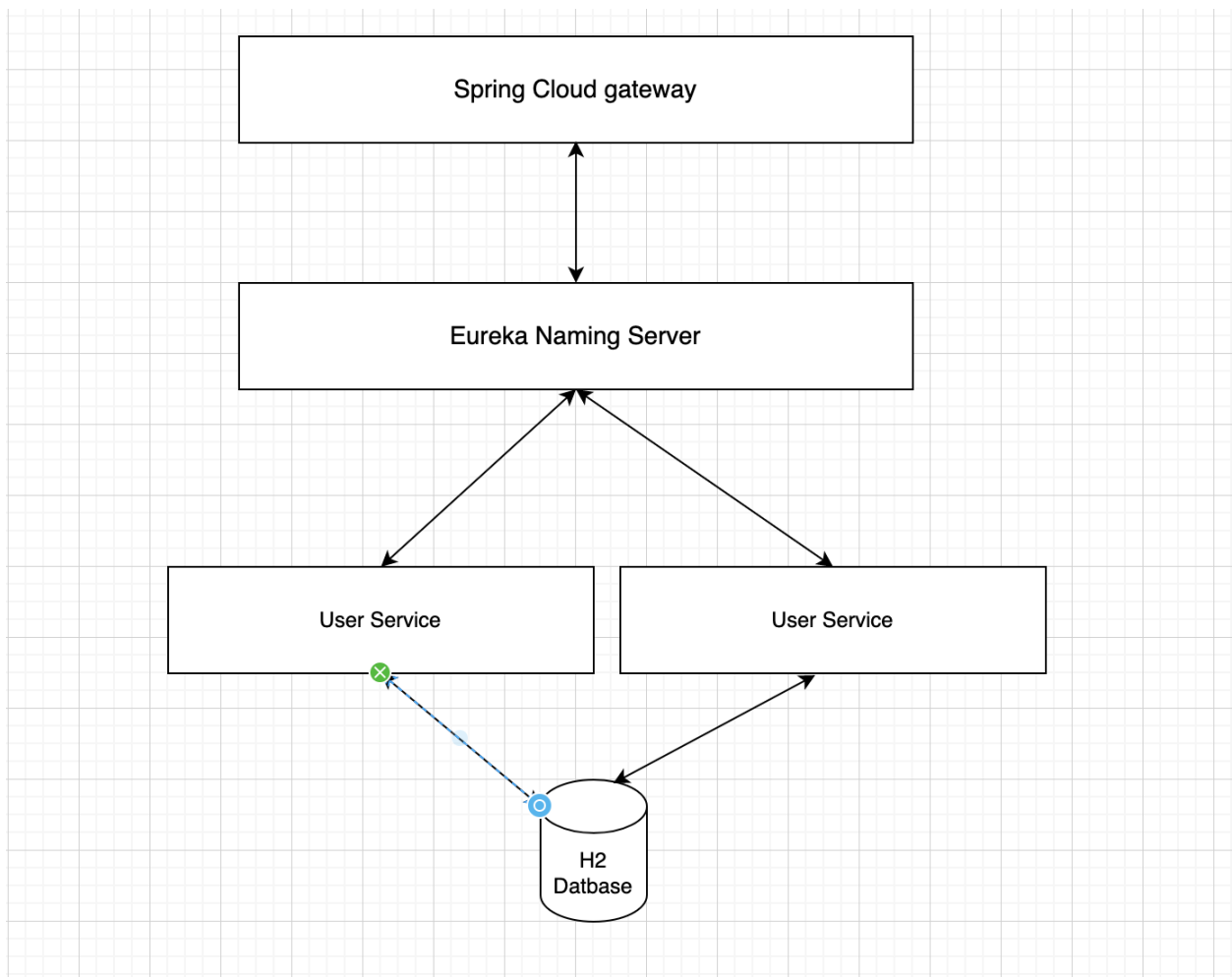


## Problem Statement

Feature: Develop a Restful API to Create, Update, and Delete Users  
Users shall be stored in a database (For example H2 in-memory database)

I have used Spring Boot and Spring Cloud for developing the user service.

## Architecture



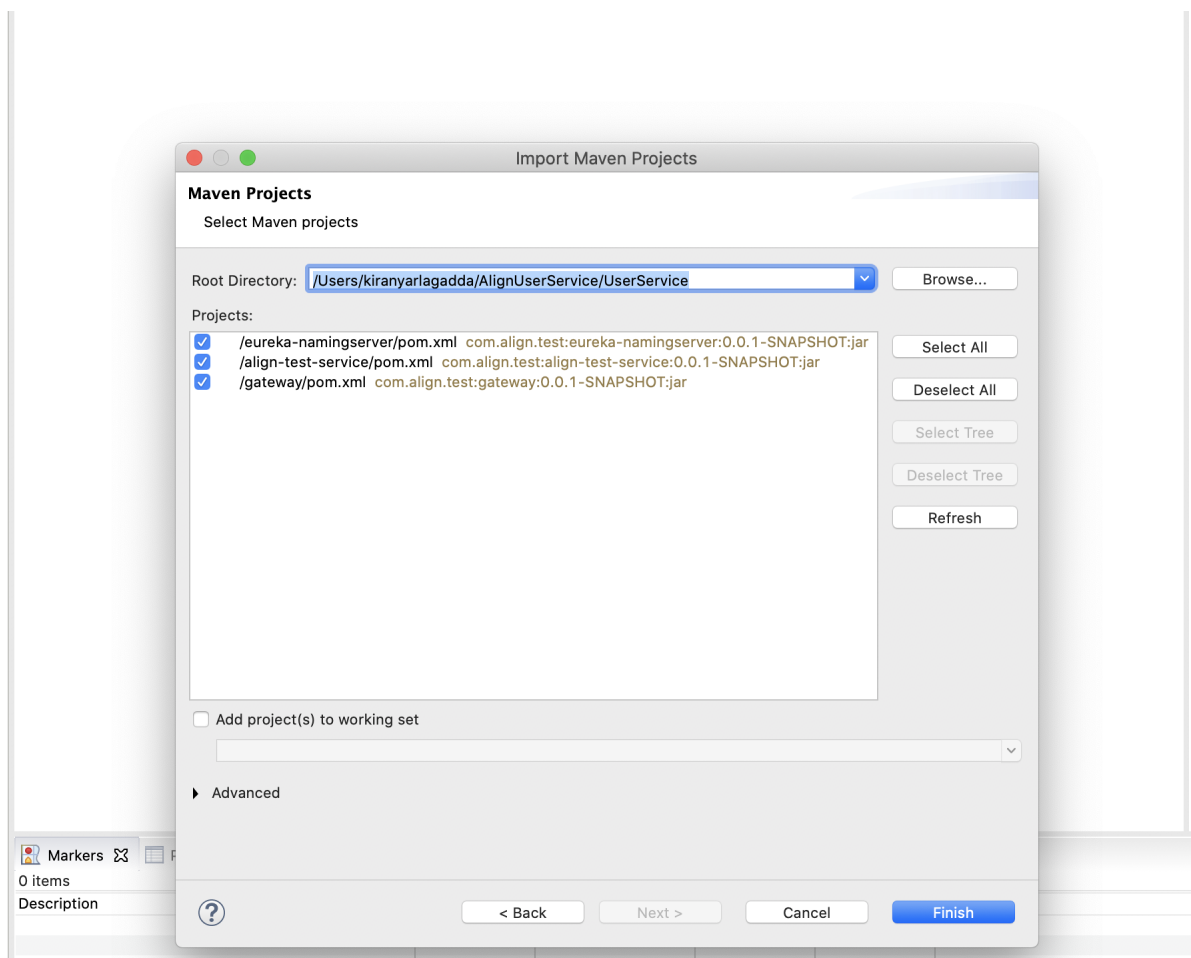
Spring Cloud Gateway is used for load balancing and can also be used for authentication and authorization of services along with monitoring,

Eureka Naming Server/ Discovery Server used to register the user services so that these services can be used by gateway.

To Test it use the below url to get the code.

git clone <https://github.com/kiranyarlagadda/UserService.git>

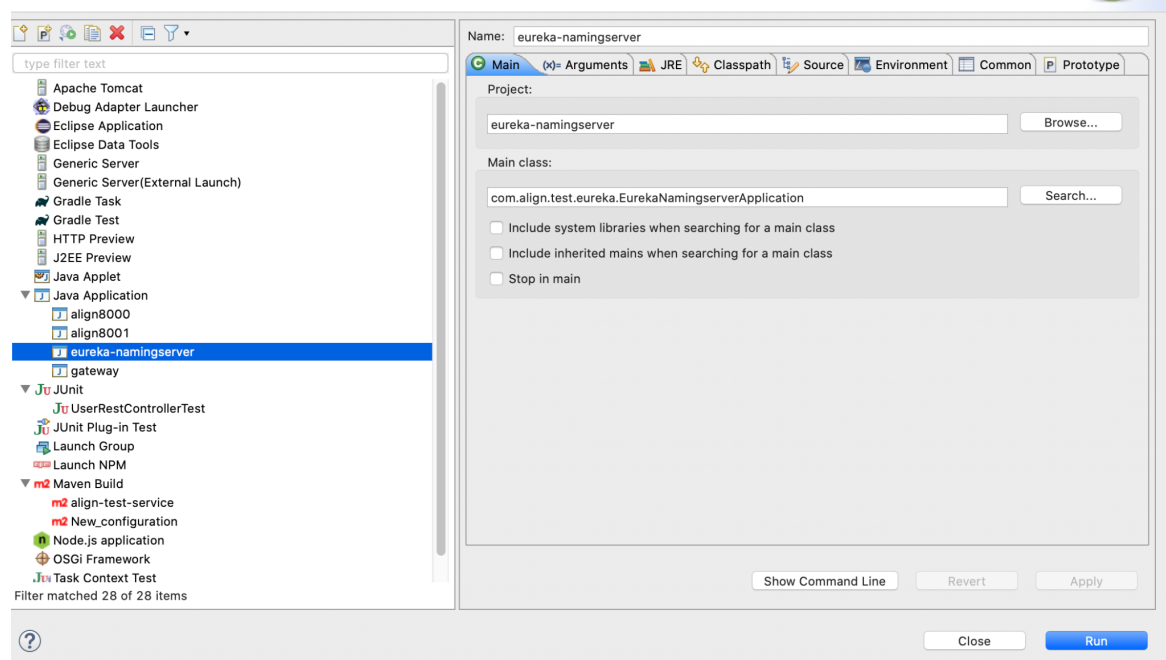
**Import the projects into Eclipse using import Existing Maven projects**



## Start the naming server from eclipse as below

### Create, manage, and run configurations

Run a Java application



## Access naming server with url: <http://localhost:8761>

The screenshot shows the Spring Eureka web interface. The header includes the 'spring Eureka' logo and a navigation bar with 'HOME' and 'LAST 1000 SINCE STARTUP'. The main content area is divided into several sections:

- System Status:** A table showing environment details and system metrics.
- Emergency Message:** A red warning message about instance status.
- DS Replicas:** A section showing the current state of data center replicas.
- Instances currently registered with Eureka:** A table showing the list of registered instances.
- General Info:** A table showing system information like memory and CPU usage.

Environment	test	Current time	2022-11-01T21:13:35 -0400
Data center	default	Uptime	00:00
		Lease expiration enabled	false
		Renews threshold	1
		Renews (last min)	0

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

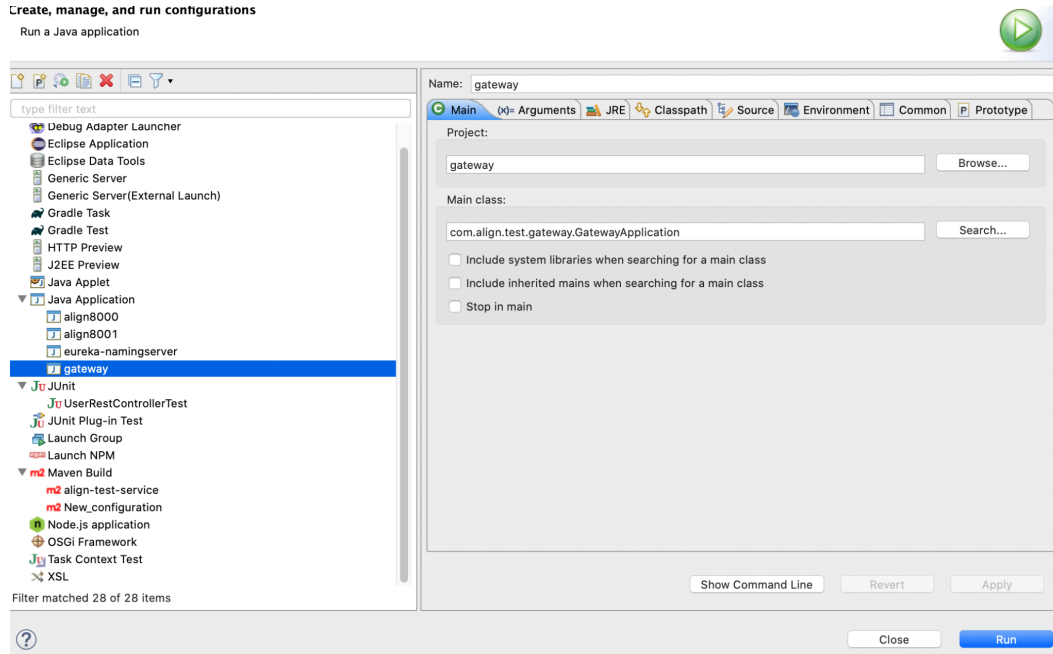
Application	AMIs	Availability Zones	Status
No instances available			

Name	Value
total-avail-memory	331mb
num-of-cpus	8
current-memory-usage	110mb (33%)
server-uptime	00:00

## Start the gateway server from eclipse

### Create, manage, and run configurations

Run a Java application



Refresh the naming server url you can see that gateway registers with the naming server as seen below.

The screenshot shows the Spring Eureka web interface. The top navigation bar includes the Spring Eureka logo and links for 'HOME' and 'LAST 1000 SINCE STARTUP'. The main content area is divided into several sections:

- System Status:** A table showing environment details and system metrics.
- EMERGENCY!** A red warning message about instance expiration.
- DS Replicas:** A section showing the local host 'localhost'.
- Instances currently registered with Eureka:** A table listing registered instances.
- General Info:** A table showing system metrics.

Environment	test	Current time	2022-11-01T21:17:50 -0400
Data center	default	Uptime	00:04
		Lease expiration enabled	false
		Renews threshold	3
		Renews (last min)	0

**EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.**

**DS Replicas**

localhost

Application	AMIs	Availability Zones	Status
GATEWAY	n/a (1)	(1)	UP (1) - <a href="#">10.0.0.18:gateway:8765</a>

Name	Value
total-avail-memory	331mb
num-of-cpus	8
current-memory-usage	167mb (50%)
server-uptime	00:04

Start the userservice on port 8000 and 8001 in eclipse with following vm parameters

**-Dserver.port=8000 -Denv=A and -Dserver.port=8001 -Denv=B**

The user services will also register on the naming server.

The screenshot shows the Spring Eureka web interface. The top navigation bar includes the 'spring Eureka' logo and links for 'HOME' and 'LAST 1000 SINCE STARTUP'. The main content area is divided into several sections:

- System Status:** A table showing environment details.

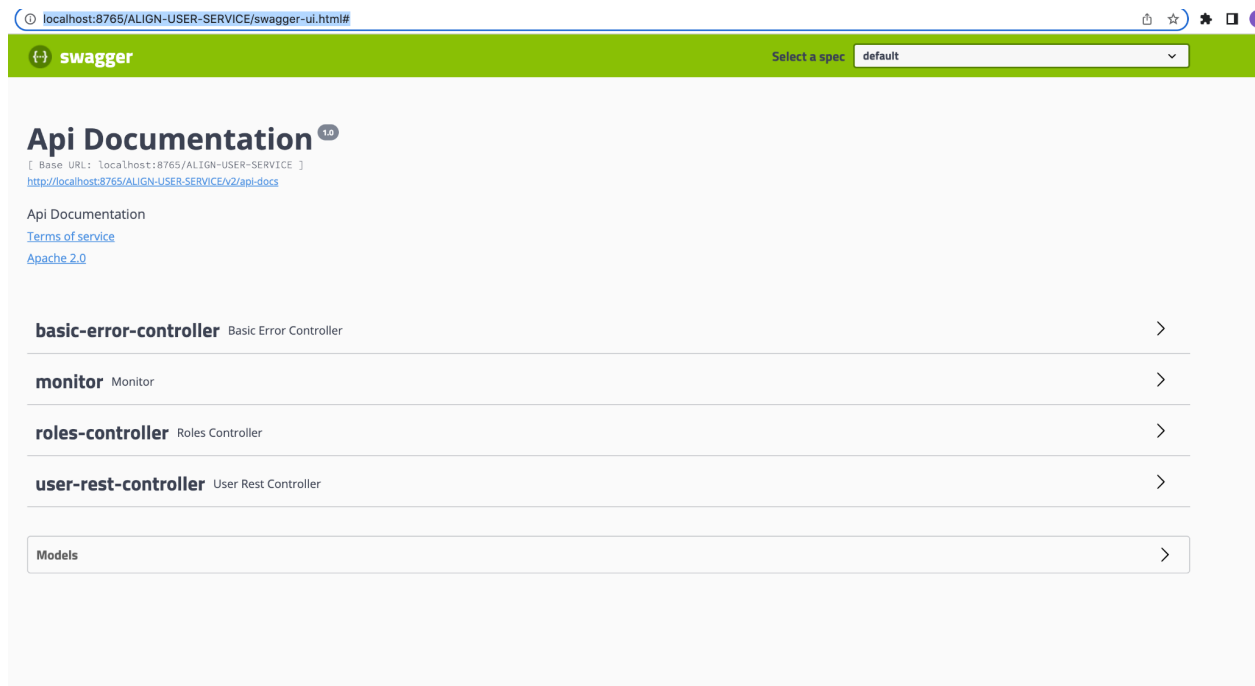
Environment	test	Current time	2022-11-01T21:20:46 -0400
Data center	default	Uptime	00:07
		Lease expiration enabled	false
		Renews threshold	6
		Renews (last min)	4
- EMERGENCY!** A red warning message: "EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE."
- DS Replicas:** A section with a search bar containing 'localhost'.
- Instances currently registered with Eureka:** A table listing registered services.

Application	AMIs	Availability Zones	Status
ALIGN-USER-SERVICE	n/a (2)	(2)	UP (2) - <a href="#">10.0.0.18:align-user-service:8000</a> , <a href="#">10.0.0.18:align-user-service:8001</a>
GATEWAY	n/a (1)	(1)	UP (1) - <a href="#">10.0.0.18:gateway:8765</a>
- General Info:** A table showing system metrics.

Name	Value
total-avail-memory	331mb

Access the Swagger UI with below url

<http://localhost:8765/ALIGN-USER-SERVICE/swagger-ui.html#>(via gateway 8765 gateway port)



You can check the status and test the services from the swagger ui.

For example to test the status use url <http://localhost:8765/ALIGN-USER-SERVICE/status>  
/swagger ui

First time the response is

environment: A port: 8000 StartUptime: 1667352028467 and second time it is  
environment: B port: 8001 StartUptime: 1667352034028

We can see the load balance happening here.

localhost:8765/ALIGN-USER-SERVICE/swagger-ui.html#/monitor/getStatusUsingGET

[Terms of service](#)  
[Apache 2.0](#)

### basic-error-controller

Basic Error Controller

#### monitor

Monitor

GET /status getStatus

Parameters

No parameters

Execute Clear

Responses

Response content type \*/\*

Curl

```
curl -X GET "http://localhost:8765/ALIGN-USER-SERVICE/status" -H "accept: */*"
```

Request URL

```
http://localhost:8765/ALIGN-USER-SERVICE/status
```

Server response

Code	Details
200	<p>Response body</p> <pre>environment: B port: 8001 StartUptime: 1667352034028</pre> <p>Download</p>

localhost:8765/ALIGN-USER-SERVICE/swagger-ui.html#/monitor/getStatusUsingGET

[Terms of service](#)  
[Apache 2.0](#)

### basic-error-controller

Basic Error Controller

#### monitor

Monitor

GET /status getStatus

Parameters

No parameters

Execute Clear

Responses

Response content type \*/\*

Curl

```
curl -X GET "http://localhost:8765/ALIGN-USER-SERVICE/status" -H "accept: */*"
```

Request URL

```
http://localhost:8765/ALIGN-USER-SERVICE/status
```

Server response

Code	Details
200	<p>Response body</p> <pre>environment: A port: 8000 StartUptime: 1667352028467</pre> <p>Download</p>

You can find some of the request and response for some of the services.

### Create Roles

<http://localhost:8765/ALIGN-USER-SERVICE/api/roles/create>

#### Request

```
{  
  
  "id": 1,  
  
  "roleName": "admin"  
}
```

### Create User

<http://localhost:8765/ALIGN-USER-SERVICE/api/user/createUser>

#### Request

```
{  
  
  "email": "kk@kk.com",  
  
  "password": "1234",  
  
  "rolesRequest": [  
  
    {
```



```
    "id": 1
  }
],
  "userName": "kiran"
}
```

## Response

```
{
  "uuid": "90464fb5-8601-47ec-959e-a50bdaa8be40",
  "userName": "kiran",
  "email": "kk@kk.com",
  "password": "1234",
  "rolesResponse": [
    {
      "id": 1,
      "roleName": "admin"
    }
  ]
}
```

}

### **Get User By UUID**

**<http://localhost:8765/ALIGN-USER-SERVICE/api/user/getUserById/90464fb5-8601-47ec-959e-a50bdaa8be40>**

### **GET User By Email**

**<http://localhost:8765/ALIGN-USER-SERVICE/api/user/getUserByUserEmail/kk%40kk.com>**

### **Get User By Name**

**<http://10.0.0.18:8765/ALIGN-USER-SERVICE/api/user/getUserByUserName/kiran>**

For **Auto-scaling** we can deploy these to any cloud environment and use the monitoring system to monitor and auto scale it.

and we can generate the images by using the maven build **mvn spring-boot:build-image -DskipTests** and use in container world and orchestrate with kubernetes

If you want to generate the image uncomment/add this in pom.xml, **kirany is my dockerhub login**

```
<image>  
    <name>kirany/mmv2-${project.artifactId}:${project.version}</name>  
</image>
```