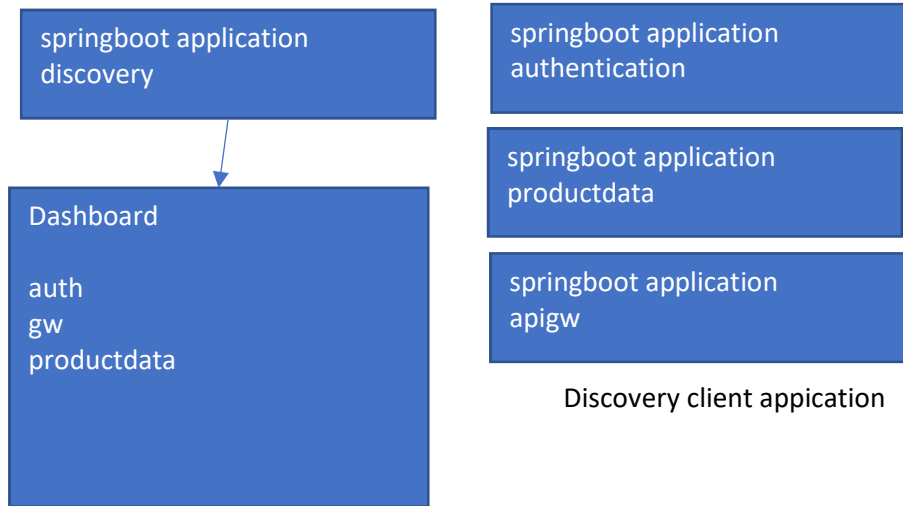
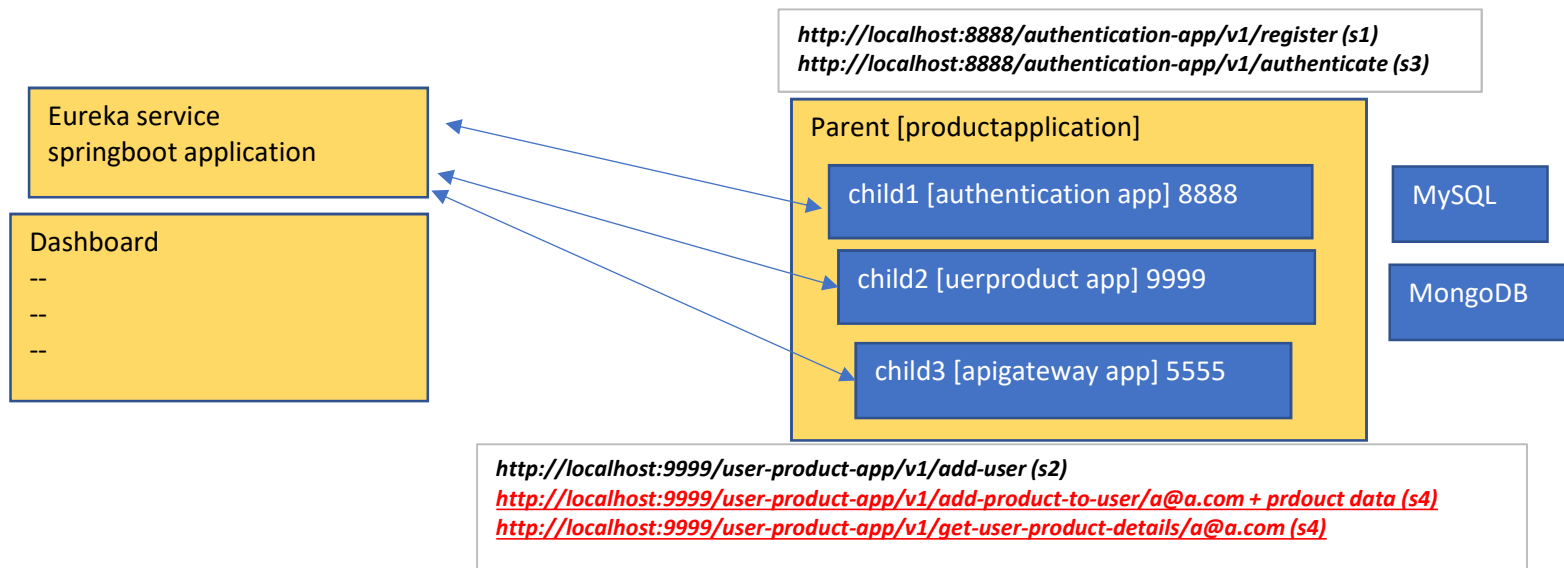


Discovery server application

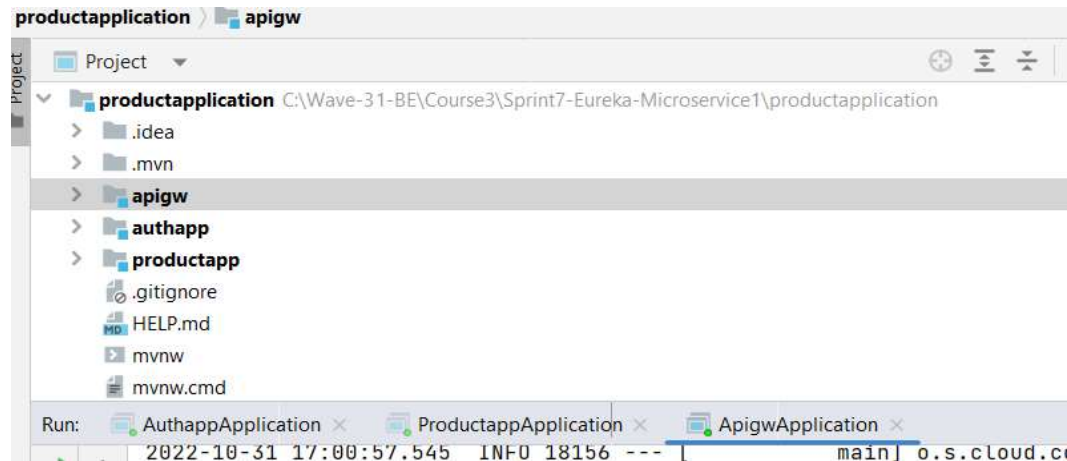


Demo 1



Steps to add discovery server to existing application

Step 1 Make sure existing applications running



Step 2 Create new springboot application as EUREKA server application by adding required dependency

A screenshot of the Spring Initializr web form. The form is titled 'spring initializr'. It has several sections: 'Project' with radio buttons for 'Gradle Project' and 'Maven Project' (selected); 'Language' with radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'; 'Spring Boot' with radio buttons for various versions, with '2.7.5' selected; 'Project Metadata' with text input fields for 'Group' (com.stackroute), 'Artifact' (discovery), 'Name' (discovery), 'Description' (Demo project for Spring Boot), and 'Package name' (com.stackroute.discovery); 'Packaging' with radio buttons for 'Jar' (selected) and 'War'; and 'Java' with radio buttons for versions 19, 17 (selected), 11, and 8. On the right side, there is a 'Dependencies' section with a table showing 'Eureka Server' and 'SPRING CLOUD DISCOVERY' as selected dependencies.

extract and open in intellij

Step 2 Edit application properties

port
application name
fetched / registered

spring cloud

```
application.properties x
1 server.port=8761
2 spring.application.name=eureka-server
3 eureka.client.fetch-registry=false
4 eureka.client.register-with-eureka=false
```

hystrix
eureka
gw

Step 3 Make application as eureka server by keeping proper annotation in main class

```
7 @SpringBootApplication
8 @EnableEurekaServer
9 public class DiscoveryApplication {
10
11     public static void main(String[] args) { SpringApplication.run(DiscoveryApplication.class, args); }
12
13
14
15 }
```

Note

```
13 <version>0.0.1-SNAPSHOT</version>
14 <name>discovery</name>
15 <description>Demo project for Spring Boot</description>
16 <properties>
17   <java.version>17</java.version>
18   <spring-cloud.version>2021.0.4</spring-cloud.version>
19 </properties>
20 <dependencies>
21   <dependency>
22     <groupId>org.springframework.cloud</groupId>
23     <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
24   </dependency>
25
26   <dependency>
27     <groupId>org.springframework.boot</groupId>
28     <artifactId>spring-boot-starter-test</artifactId>
29     <scope>test</scope>
30   </dependency>
31 </dependencies>
32 <dependencyManagement>
33   <dependencies>
34     <dependency>
35       <groupId>org.springframework.cloud</groupId>
36       <artifactId>spring-cloud-dependencies</artifactId>
37       <version>${spring-cloud.version}</version>
38       <type>pom</type>
39       <scope>import</scope>
40     </dependency>
41   </dependencies>
42 </dependencyManagement>
```

these elements required if any spring-cloud related dependencies add to pom

server is ready
start server applicatin
check the URL for dashboard

<http://localhost:8761>

The screenshot shows the Spring Eureka web interface at <http://localhost:8761>. The page has a dark header with the Spring Eureka logo and navigation links for HOME and LAST 1000 SINCE STARTUP. The main content area is divided into sections: System Status, DS Replicas, and General Info. The System Status section contains two tables. The first table shows Environment: test and Data center: default. The second table shows system metrics: Current time (2022-10-31T17:19:55 +0530), Uptime (00:00), Lease expiration enabled (false), Renew threshold (1), and Renew (last min) (0). The DS Replicas section has a search bar with 'localhost' entered. Below it, the text 'Instances currently registered with Eureka' is followed by a table with columns Application, Availability, and Status. The table shows 'No instances available'. A blue callout box points to the table with the text 'None of the client application registered and discovered'.

spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2022-10-31T17:19:55 +0530
Data center	default	Uptime	00:00
		Lease expiration enabled	false
		Renews threshold	1
		Renews (last min)	0

DS Replicas

localhost

Instances currently registered with Eureka

None of the client application registered and discovered

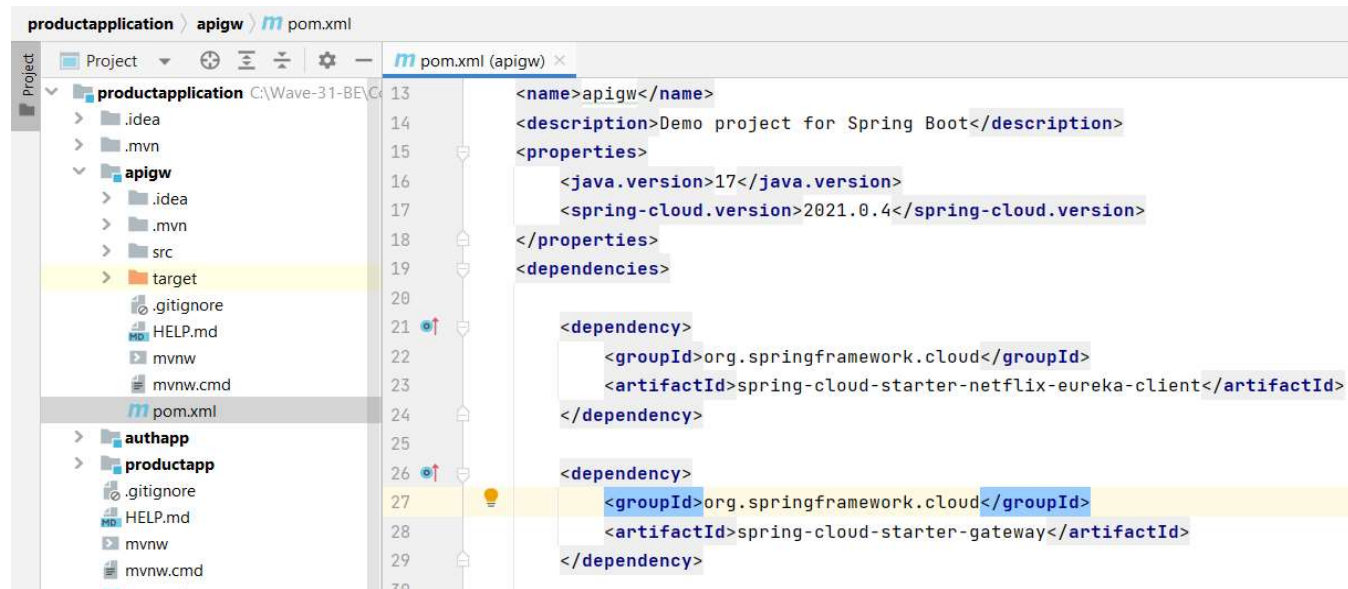
Application	Availability	Status
No instances available		

General Info

Steps to make client to register with server
apigw

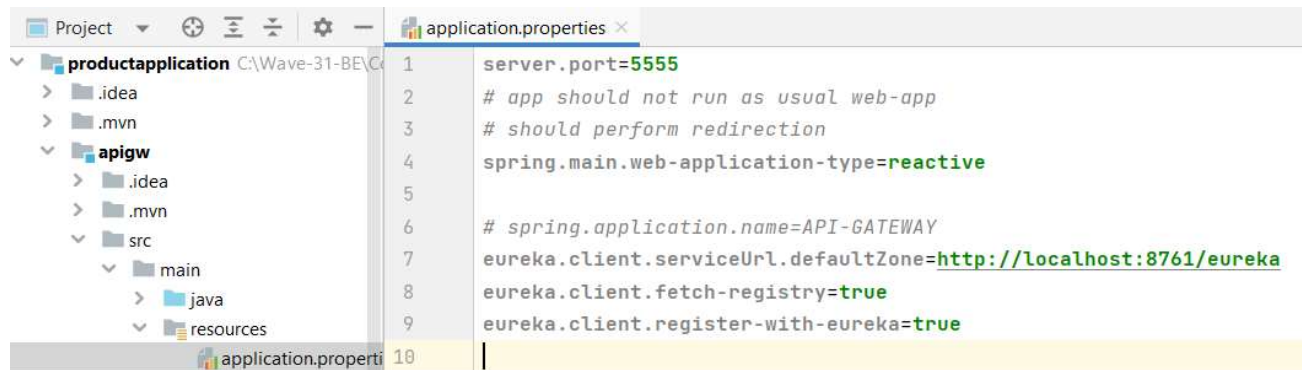
- Step 1
- Make sure client application is ready
 - Make sure pom has required properties

Step 2 add eureka client dependency to apigw pom



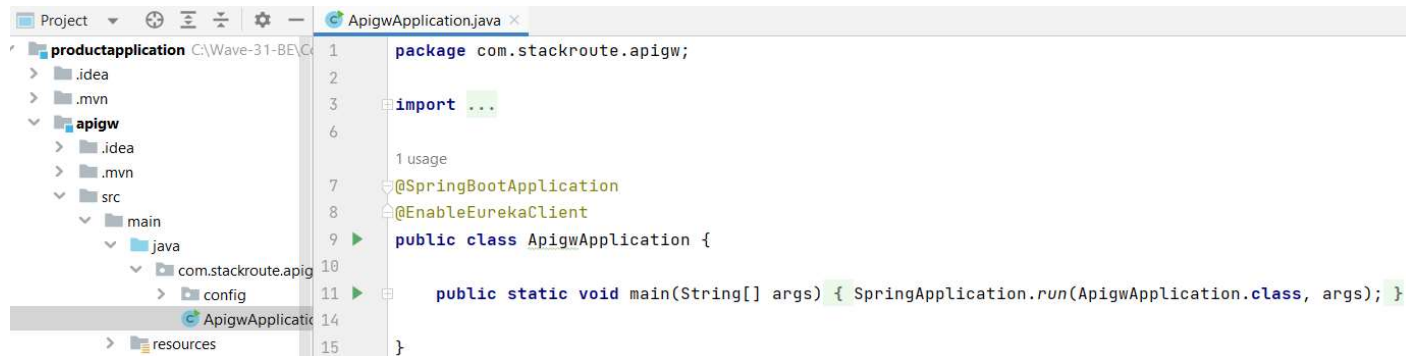
```
13 <name>apigw</name>
14 <description>Demo project for Spring Boot</description>
15 <properties>
16   <java.version>17</java.version>
17   <spring-cloud.version>2021.0.4</spring-cloud.version>
18 </properties>
19 <dependencies>
20
21   <dependency>
22     <groupId>org.springframework.cloud</groupId>
23     <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
24   </dependency>
25
26   <dependency>
27     <groupId>org.springframework.cloud</groupId>
28     <artifactId>spring-cloud-starter-gateway</artifactId>
29   </dependency>
30 </dependencies>
```

Step 3 Modify application properties



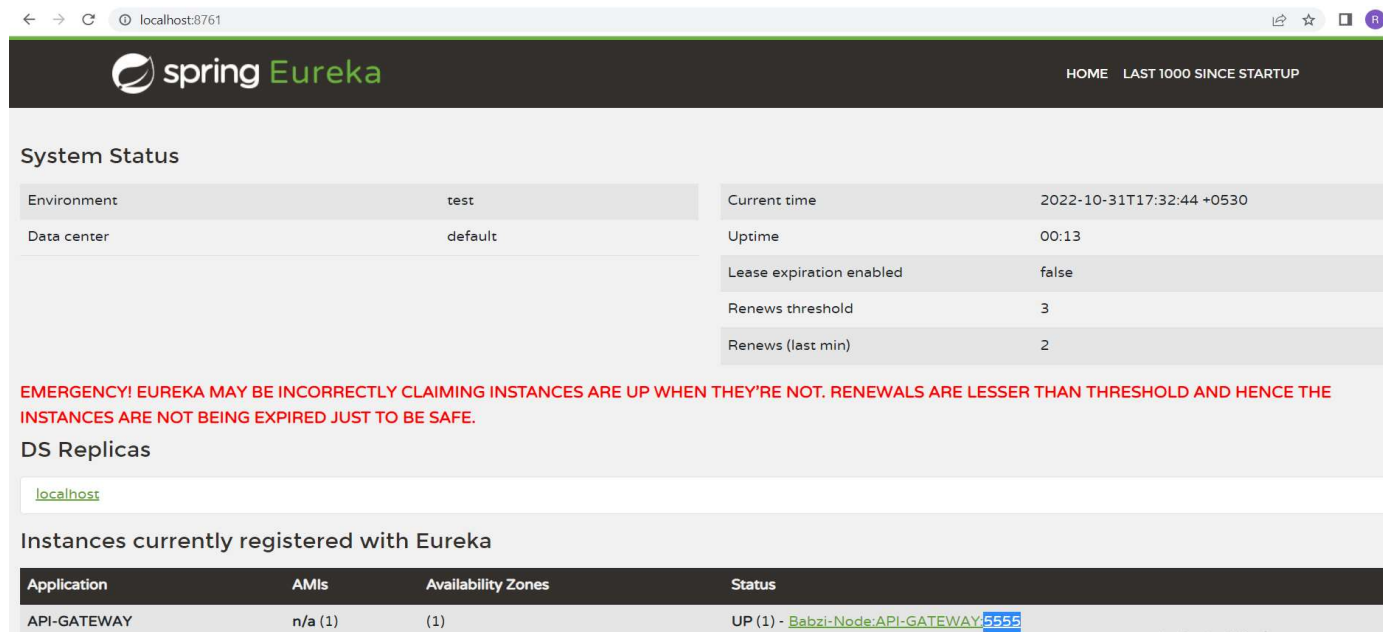
```
1 server.port=5555
2 # app should not run as usual web-app
3 # should perform redirection
4 spring.main.web-application-type=reactive
5
6 # spring.application.name=API-GATEWAY
7 eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
8 eureka.client.fetch-registry=true
9 eureka.client.register-with-eureka=true
10
```

Step 4 Edit main class keep proper annotation



```
1 package com.stackroute.apigw;
2
3 import ...
4
5
6
7 @SpringBootApplication
8 @EnableEurekaClient
9 public class ApigwApplication {
10
11     public static void main(String[] args) { SpringApplication.run(ApigwApplication.class, args); }
12
13
14
15 }
```

Start client
check server dashboard



spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2022-10-31T17:32:44 +0530
Data center	default	Uptime	00:13
		Lease expiration enabled	false
		Renews threshold	3
		Renews (last min)	2

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

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - Babzi-Node-API-GATEWAY:5555

Steps to make auth application as eureka client

Step 1 make sure authapp is ready

Note : authapp does not have any spring-cloud related dependencies
so, <spring-cloud.version> and <dependencyManagement> elements to be defined

Step 2 add eureka client dependency

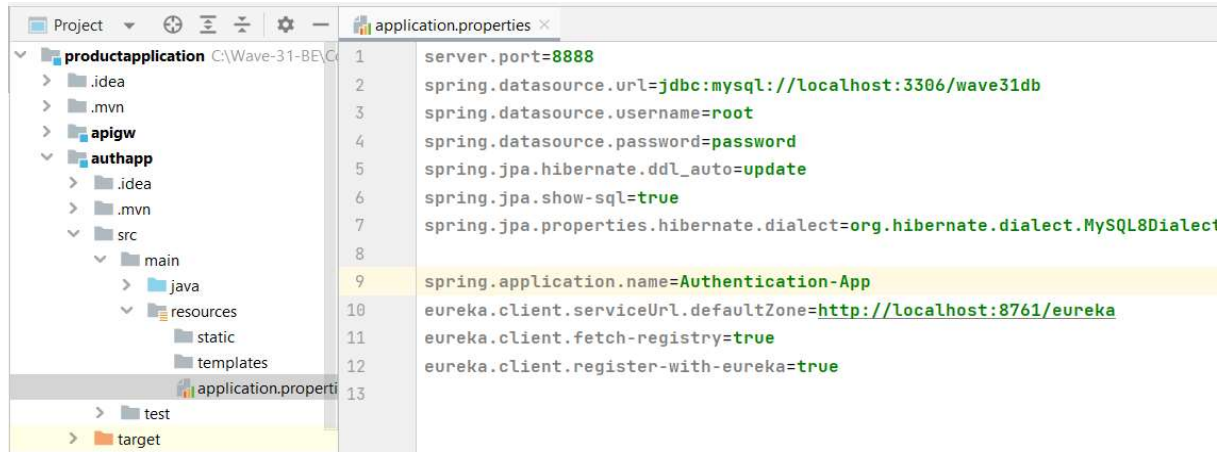
```
21 <dependency>
22   <groupId>org.springframework.cloud</groupId>
23   <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
24 </dependency>
```

Step 3 Add required properties in pom

```
15 <properties>
16   <java.version>17</java.version>
17   <spring-cloud.version>2021.0.4</spring-cloud.version>
18 </properties>
```

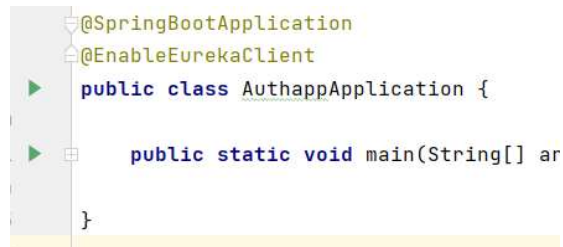
```
60 </dependencies>
61
62
63 <dependencyManagement>
64   <dependencies>
65     <dependency>
66       <groupId>org.springframework.cloud</groupId>
67       <artifactId>spring-cloud-dependencies</artifactId>
68       <version>${spring-cloud.version}</version>
69       <type>pom</type>
70       <scope>import</scope>
71     </dependency>
72   </dependencies>
73 </dependencyManagement>
```


Step 4 modify application properties
 name/server url/fetch/register



```
1 server.port=8888
2 spring.datasource.url=jdbc:mysql://localhost:3306/wave31db
3 spring.datasource.username=root
4 spring.datasource.password=password
5 spring.jpa.hibernate.ddl_auto=update
6 spring.jpa.show-sql=true
7 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
8
9 spring.application.name=Authentication-App
10 eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
11 eureka.client.fetch-registry=true
12 eureka.client.register-with-eureka=true
13
```

Step 5 add required annotation in main class



```
@SpringBootApplication
@EnableEurekaClient
public class AuthappApplication {

    public static void main(String[] ar

}
```

run client app
check server dashboard

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - Babzi-Node:API-GATEWAY:5555
AUTHENTICATION-APP	n/a (1)	(1)	UP (1) - Babzi-Node:Authentication-App:8888