

Lesson 04 Demo 03

Setting up Eureka Server

Objective: To set up an Eureka Server, which acts as a service registry and enables service discovery in a Spring Cloud application

Tool required: Eclipse IDE

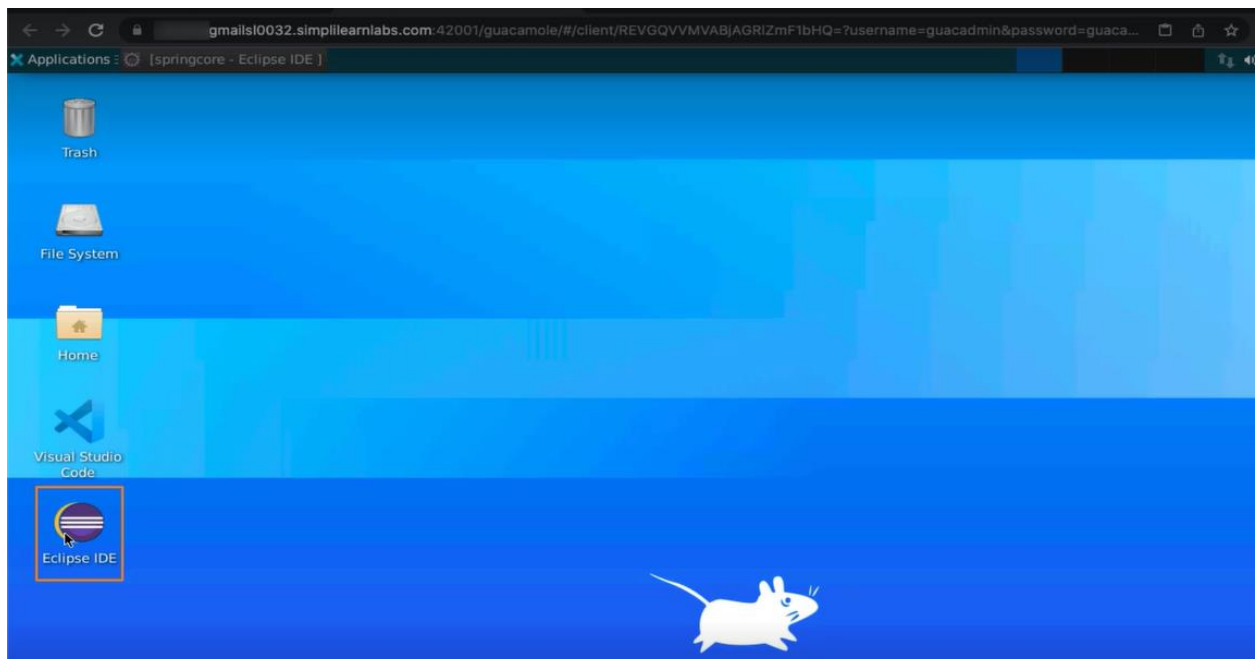
Prerequisites: None

Steps to be followed:

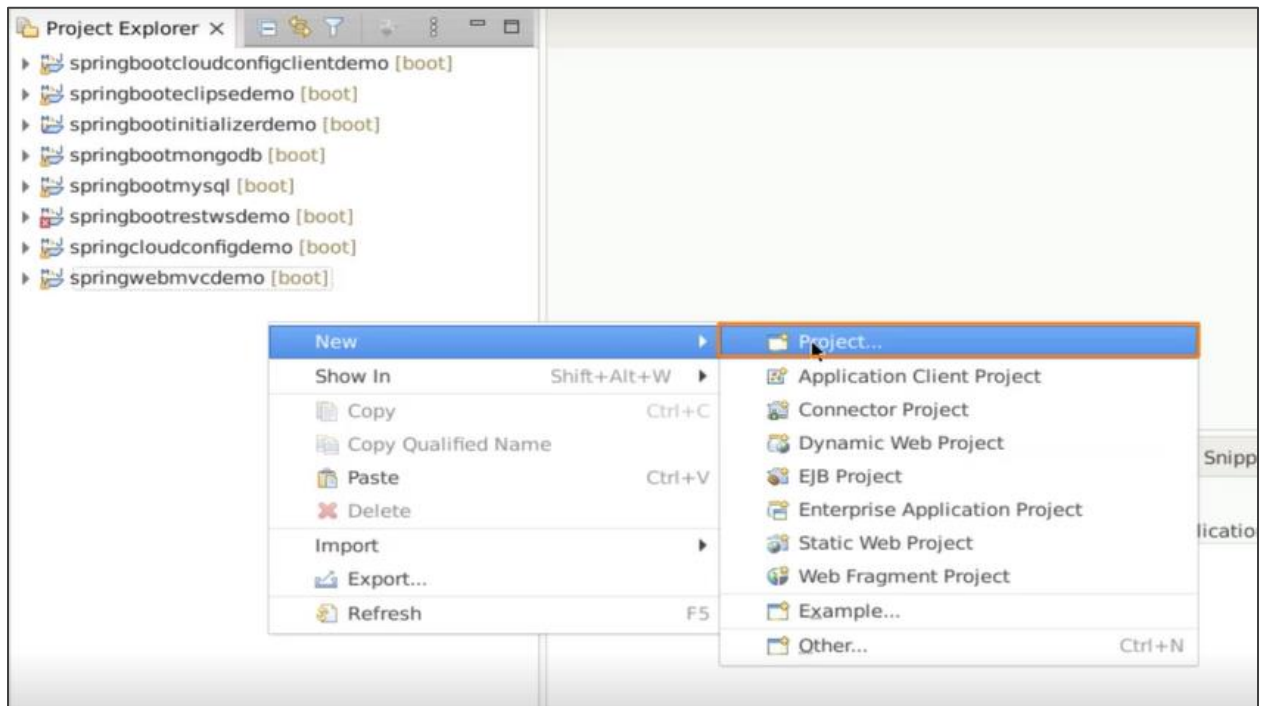
1. Creating a new Spring Starter project
2. Configuring the Eureka Server properties
3. Running the Eureka Server

Step 1: Creating a new Spring Starter project

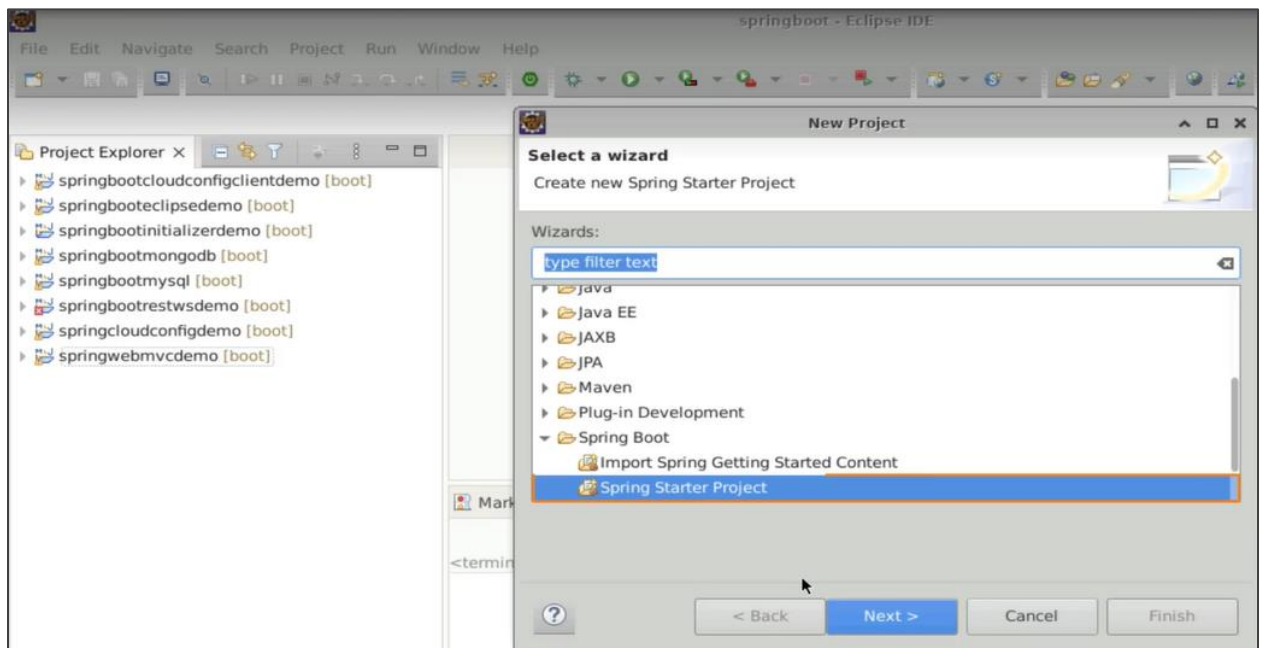
1.1 Open Eclipse IDE



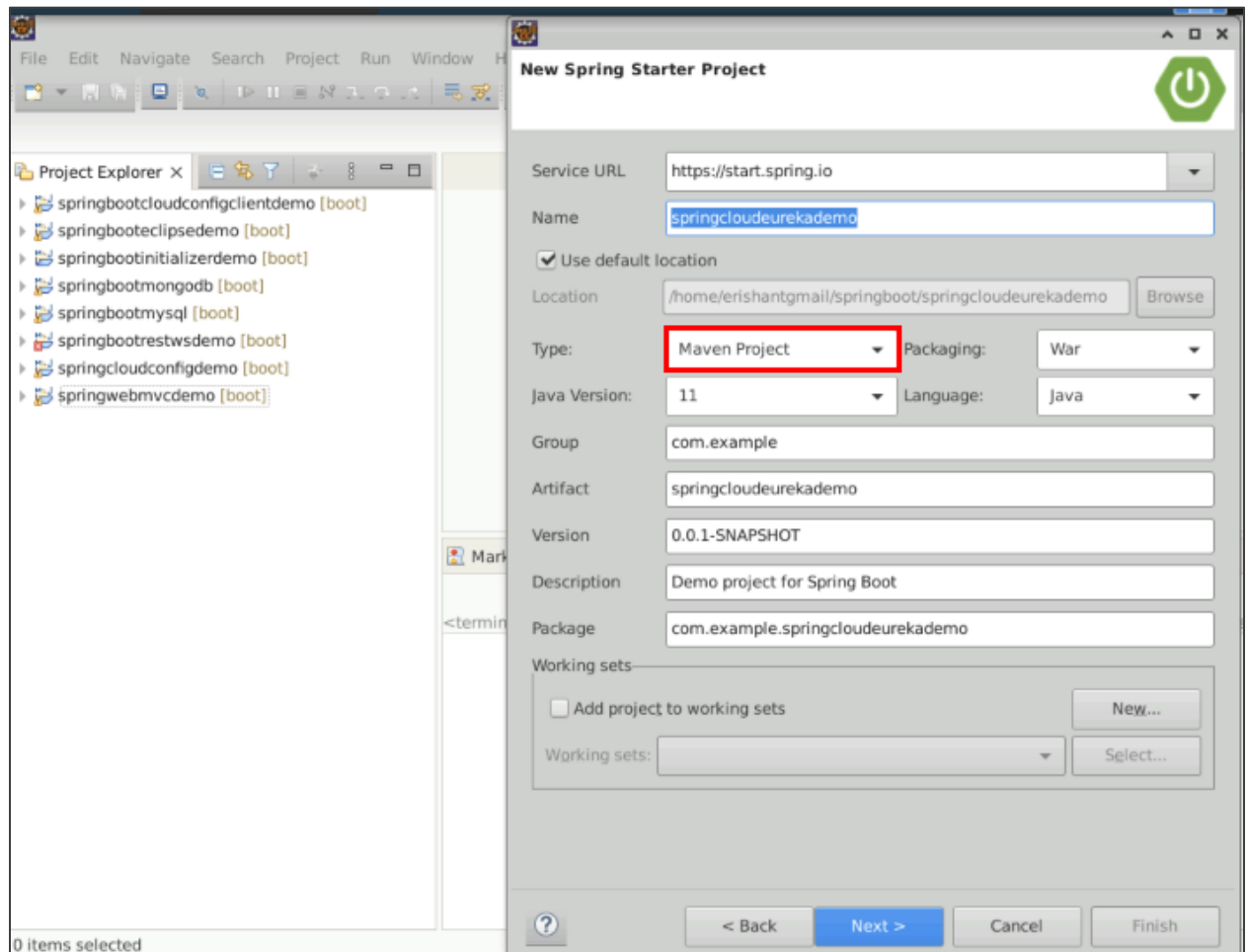
1.2 Go to **File > New > Project** to open the **Project Creation Wizard**



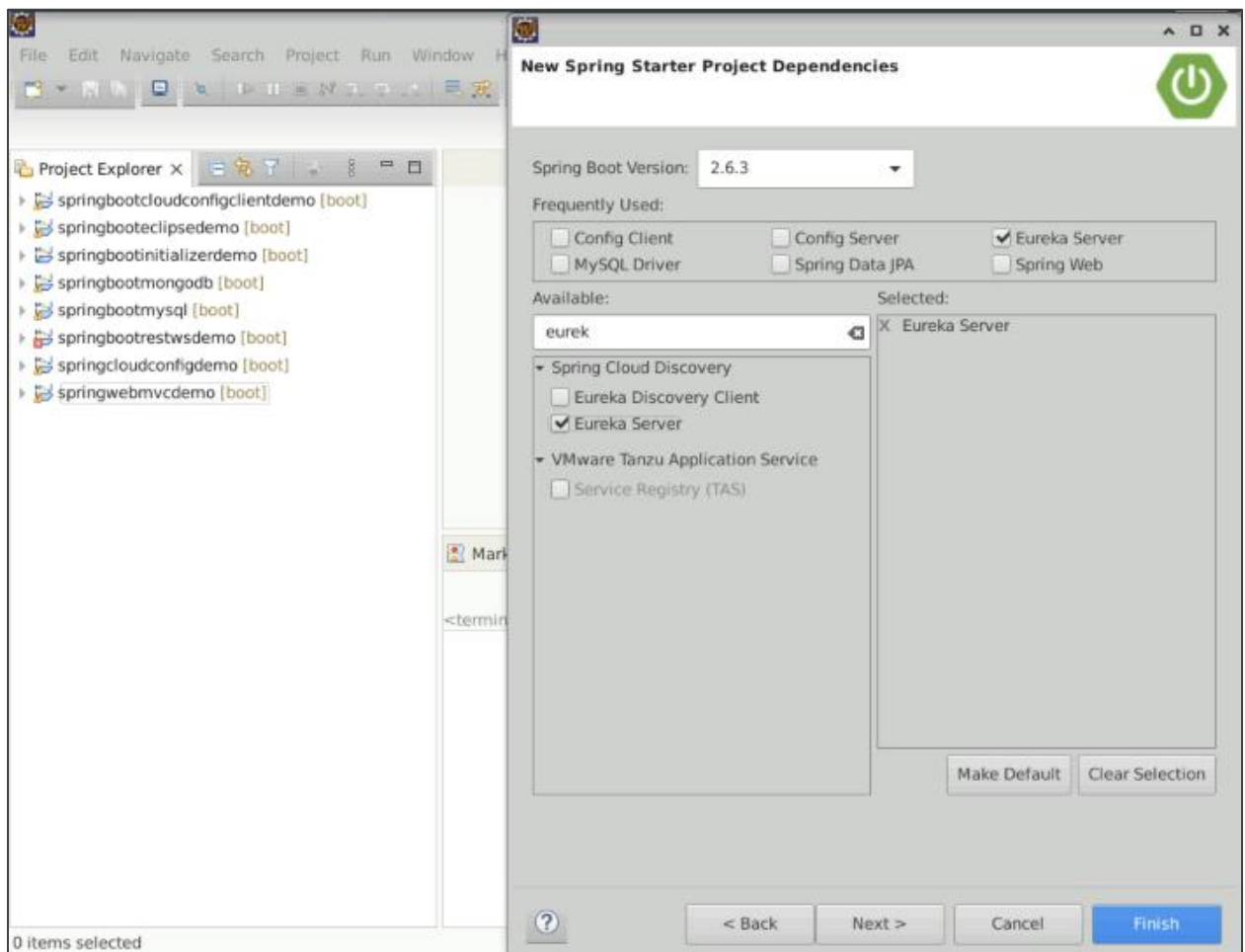
1.3 Select **Spring Starter Project** and click **Next**



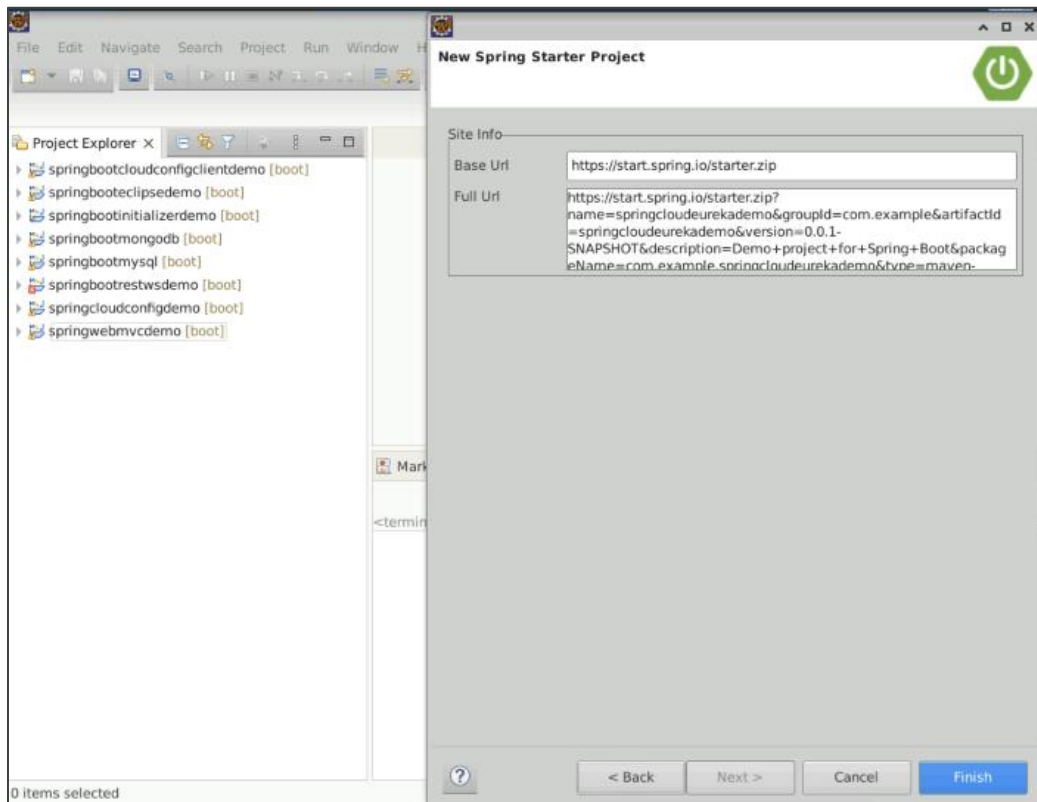
- 1.4 Enter the project name as **springcloudeurekademo** and select Maven as the build tool.
Now, click **Next**



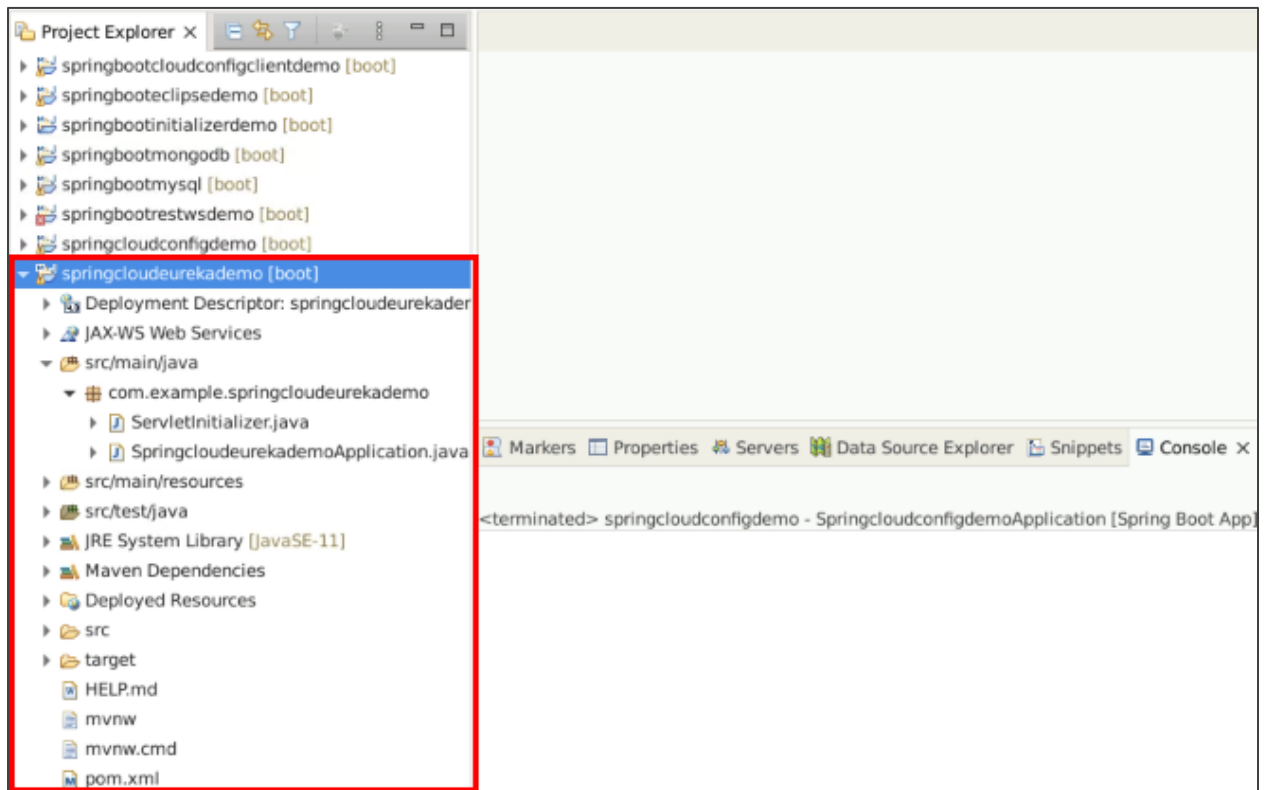
1.5 Choose the desired project dependencies, like **Eureka Server**, and click **Next**



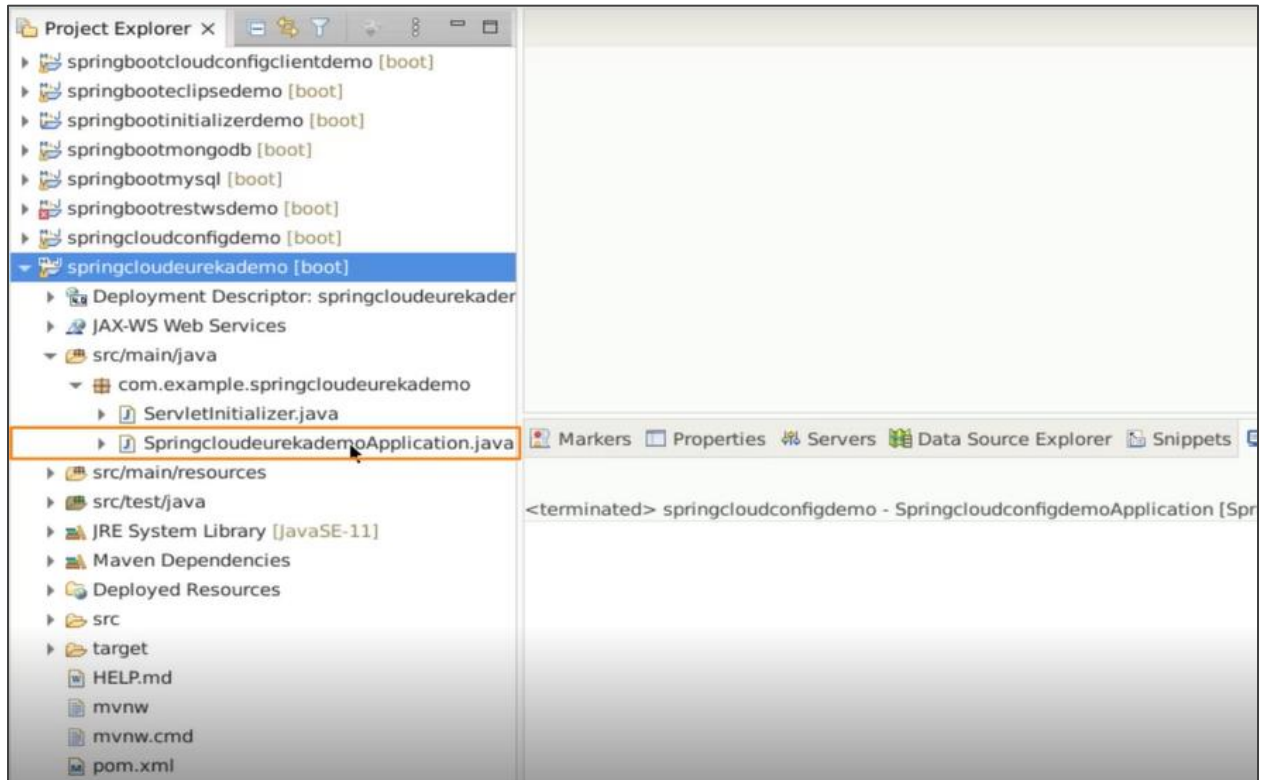
1.6 Click **Finish**



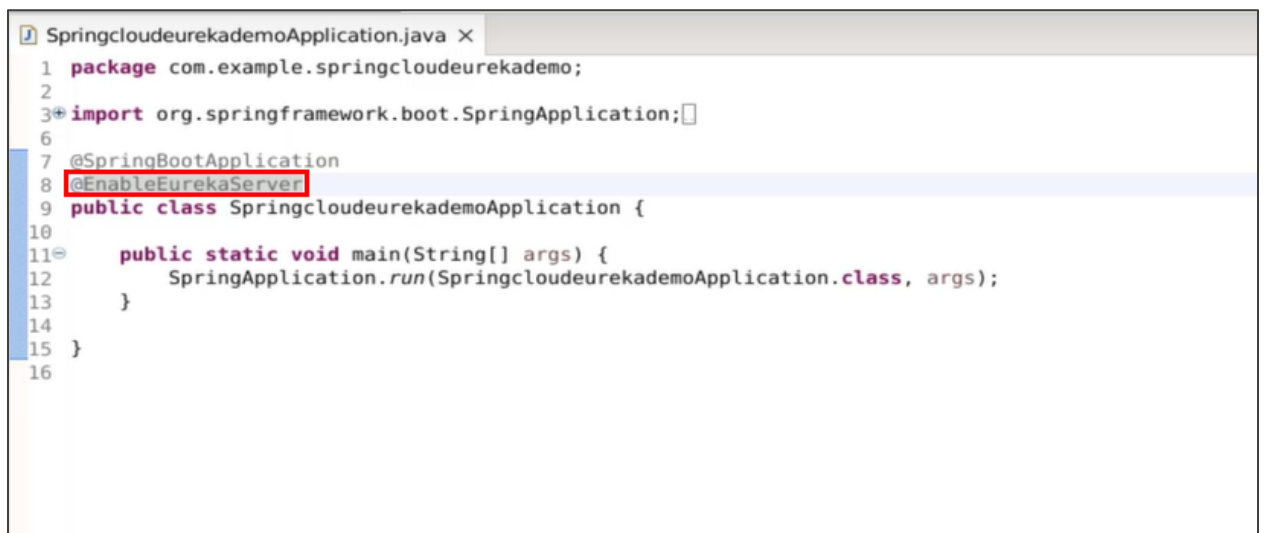
The eclipse will generate the project structure and download the necessary dependencies.



- 1.7 Navigate to the `src/main/java` package and open the file:
SpringcloudeurekademoApplication.java

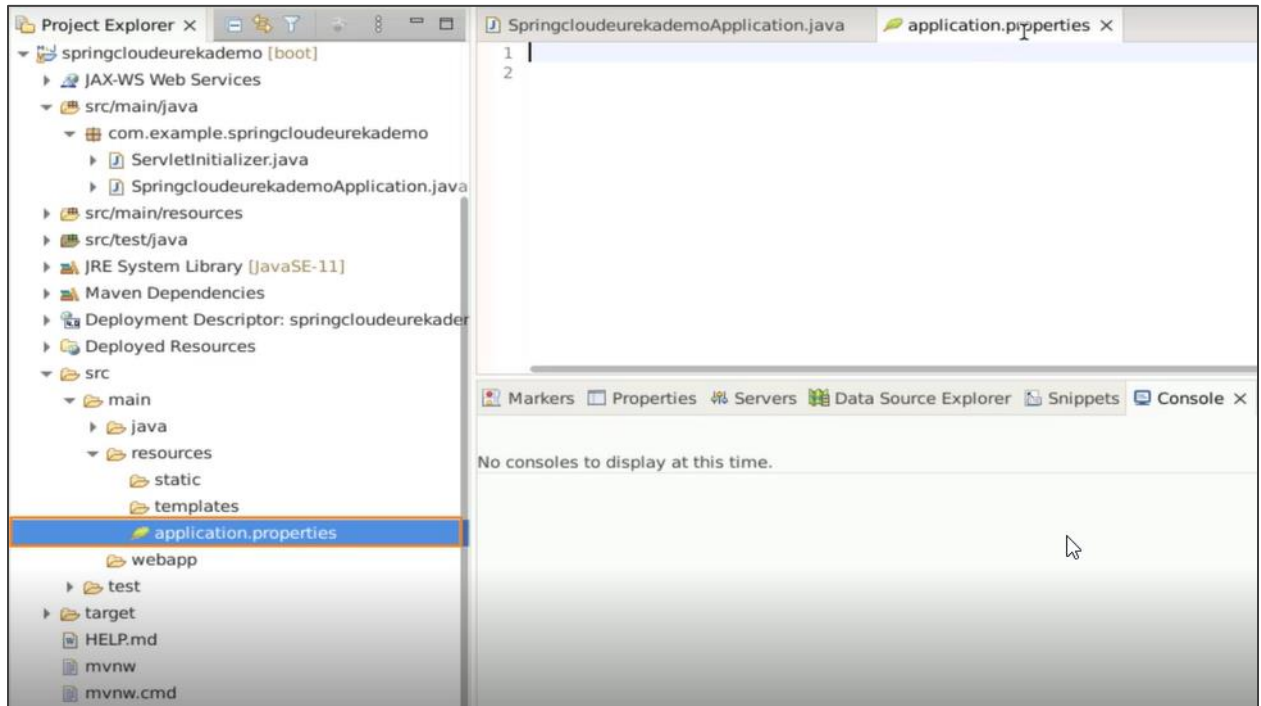


- 1.8 Add the `@EnableEurekaServer` annotation to the main class to declare it as a Eureka server



Step 2: Configuring the Eureka Server properties

2.1 Open the **application.properties** file located in **src/main/resources**

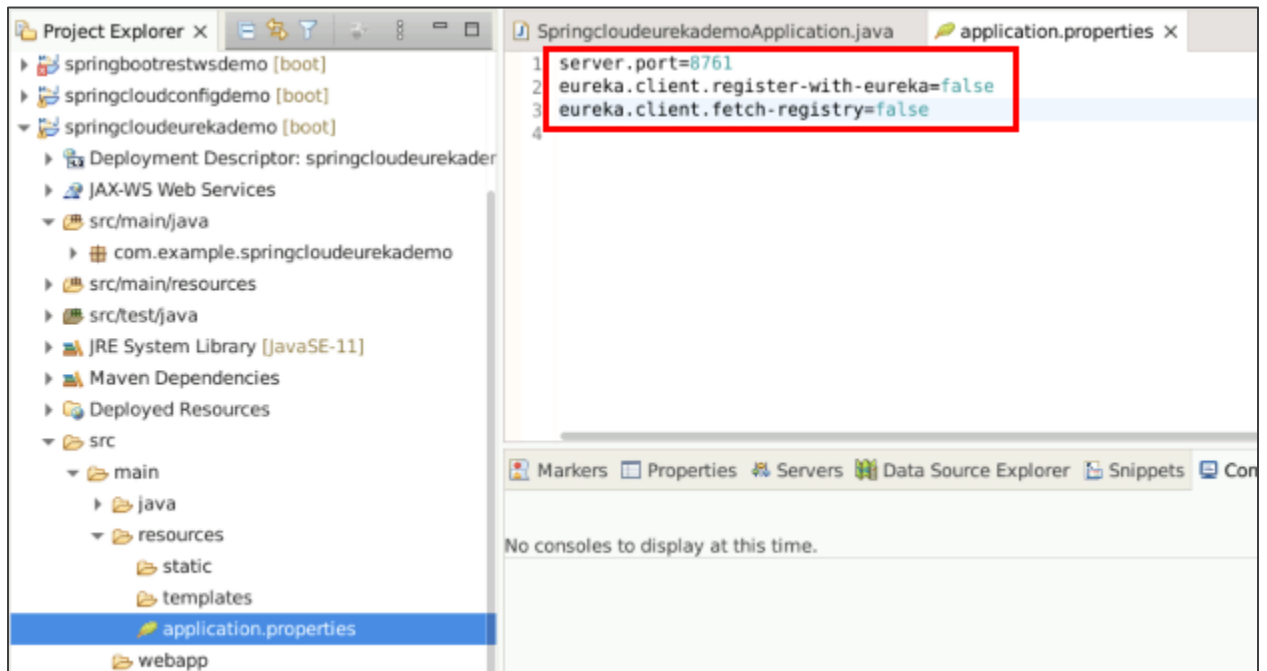


2.2 Add the following configuration properties:

server.port=8761

eureka.client.registerWithEureka=false

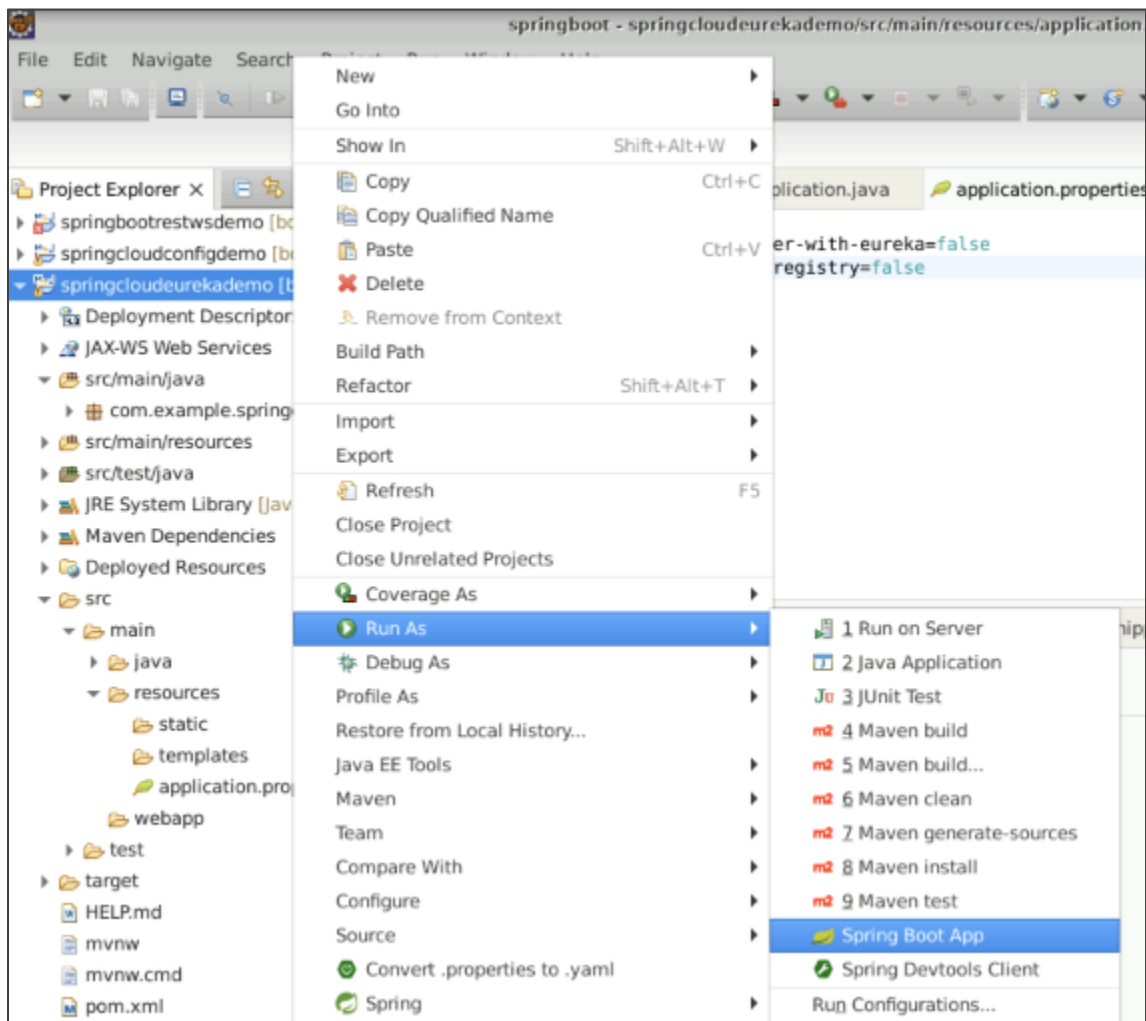
eureka.client.fetchRegistry=false



2.3 Save the **application.properties** file

Step 3: Running the Eureka Server

3.1 Right-click on the project, select **Run As > Spring Boot App** or **Java Application**



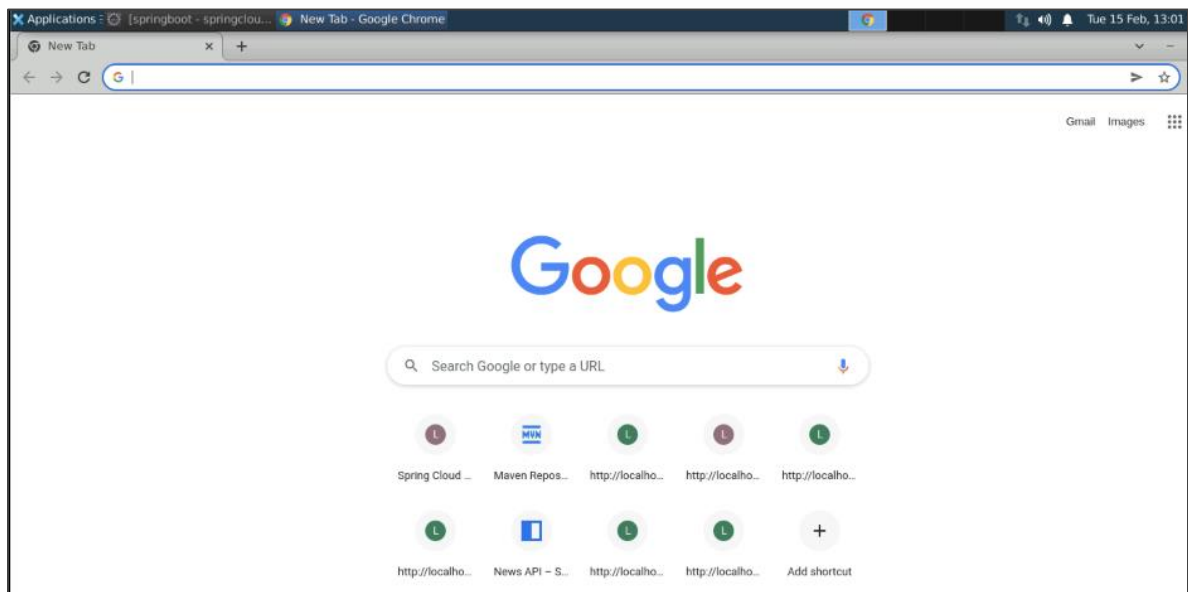
Observe the console output in the Eclipse IDE to verify that the Tomcat server has started on the configured port and that the Eureka server has started successfully.

The screenshot shows the Eclipse IDE with the `application.properties` file open. The file contains the following configuration:

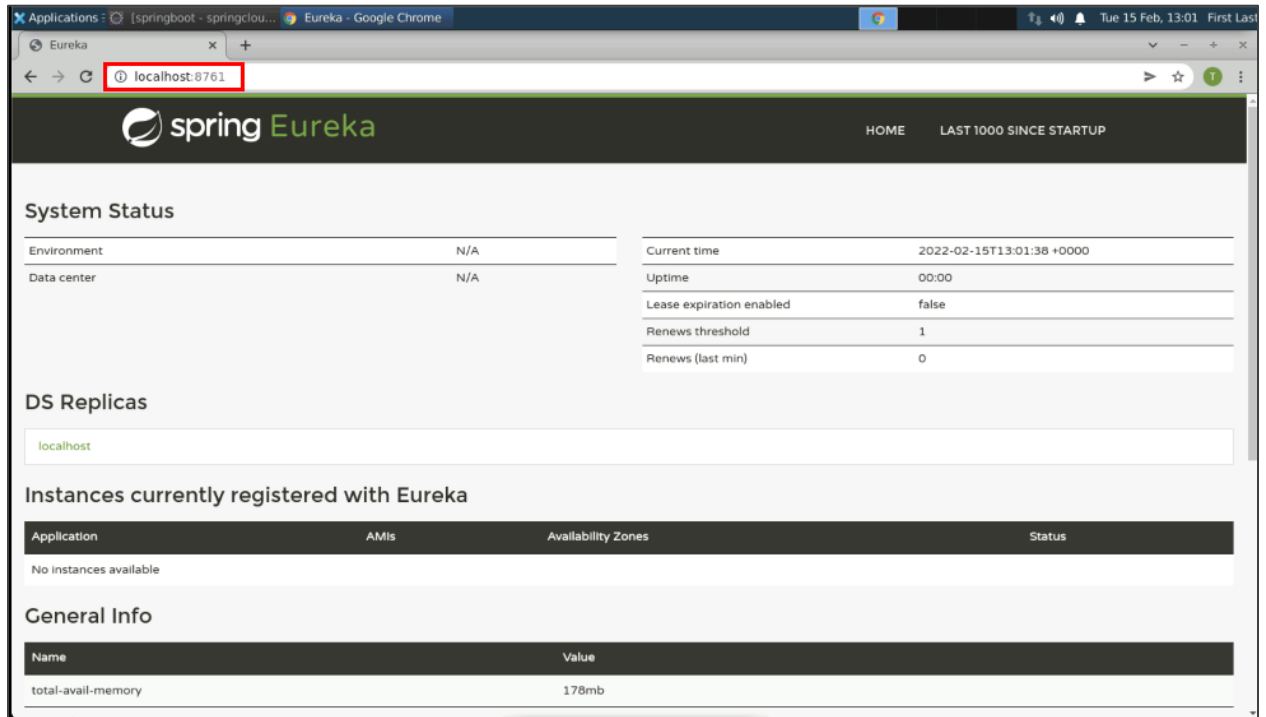
```
1 server.port=8761
2 eureka.client.register-with-eureka=false
3 eureka.client.fetch-registry=false
4
```

The console output shows the application starting successfully. The logs indicate that the Eureka Server Bootstrap is initialized, the Tomcat web server is started on port 8761, and the Spring Cloud Eureka Demo Application is started. The application is running on port 8761.

3.2 Open a web browser



3.3 Enter the following URL in the address bar: **localhost:8761** (replace 8761 with the configured port if different)



The Eureka Server dashboard will open, displaying the Eureka Server homepage.