Lesson 04 Demo 02

Setting up Spring Cloud Config Client

Objective: To set up a Spring Cloud Config Client to retrieve application configuration from a distant Git repository

Tool required: Eclipse IDE

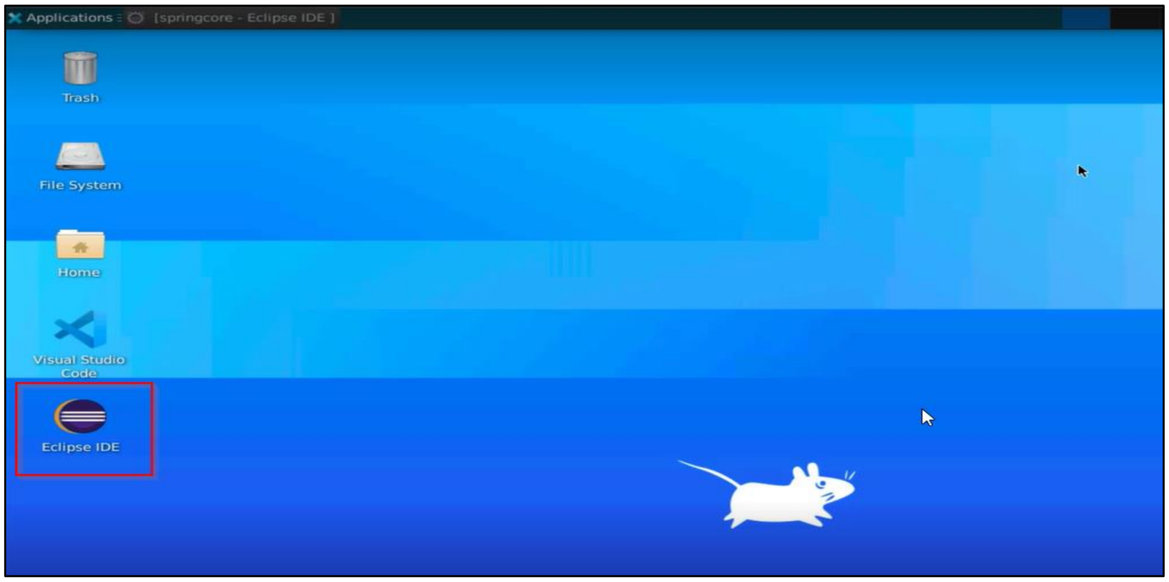
Prerequisites: None

Steps to be followed:

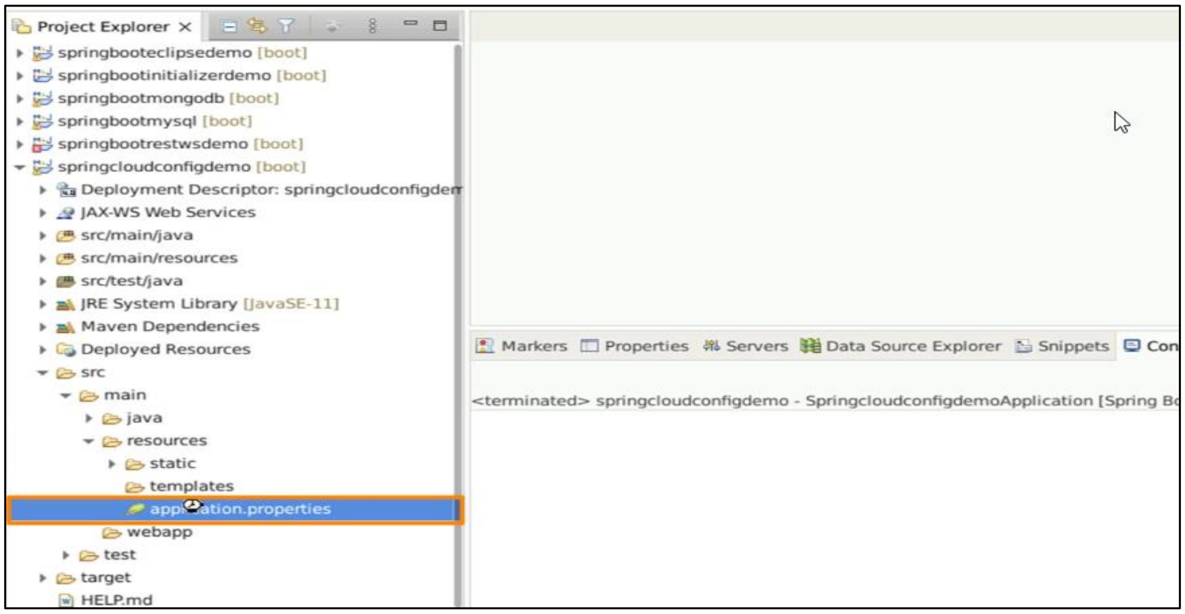
1. Creating the Spring Starter project
2. Configuring db connection
3. Running the application

Step 1: Creating the Spring Starter project

1.1 Open **Eclipse IDE**

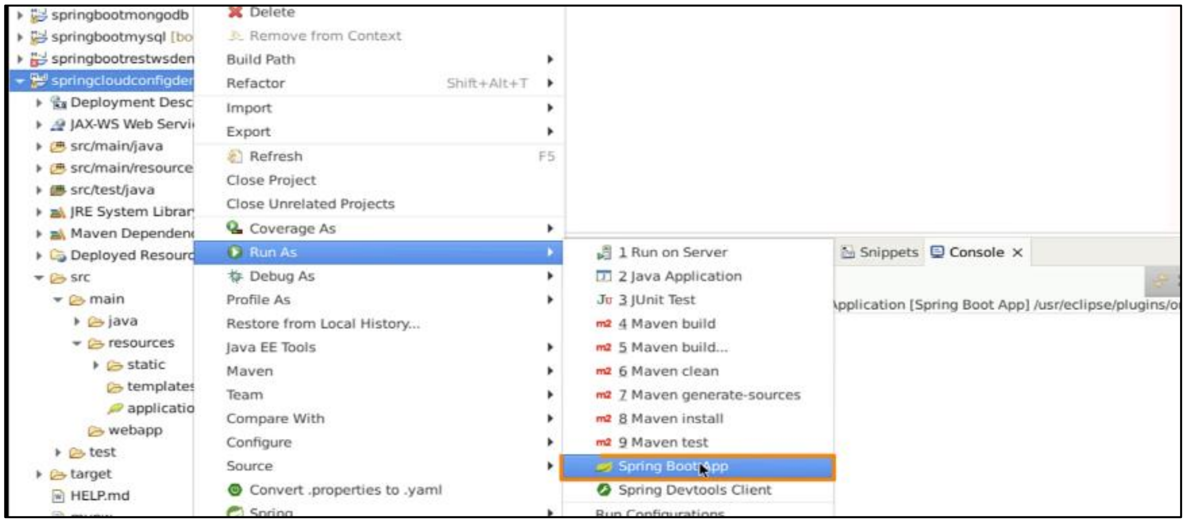


1.2 Navigate to **springcloudconfigdemo > src > application.properties** to configure the Git server



<terminated> springcloudconfigdemo - SpringcloudconfigdemoApplication [Spring Bo

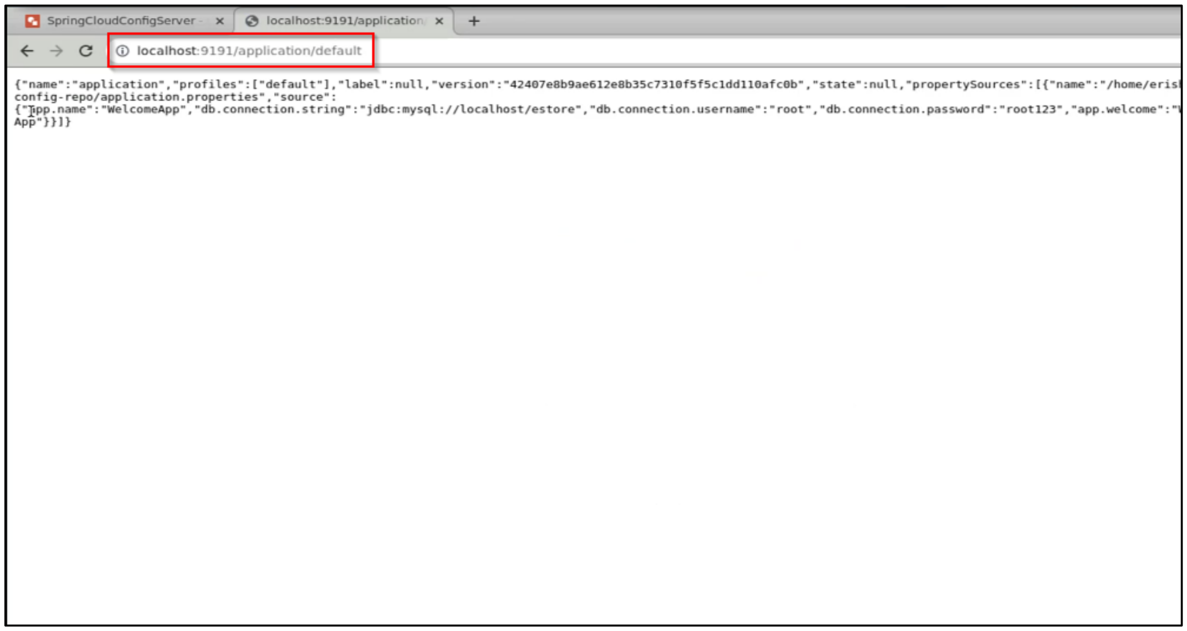
1.3 To launch the local host of the Git Commit application, select **Run As > SpringBootApp**



Application [Spring Boot App] /usr/eclipse/plugins/o

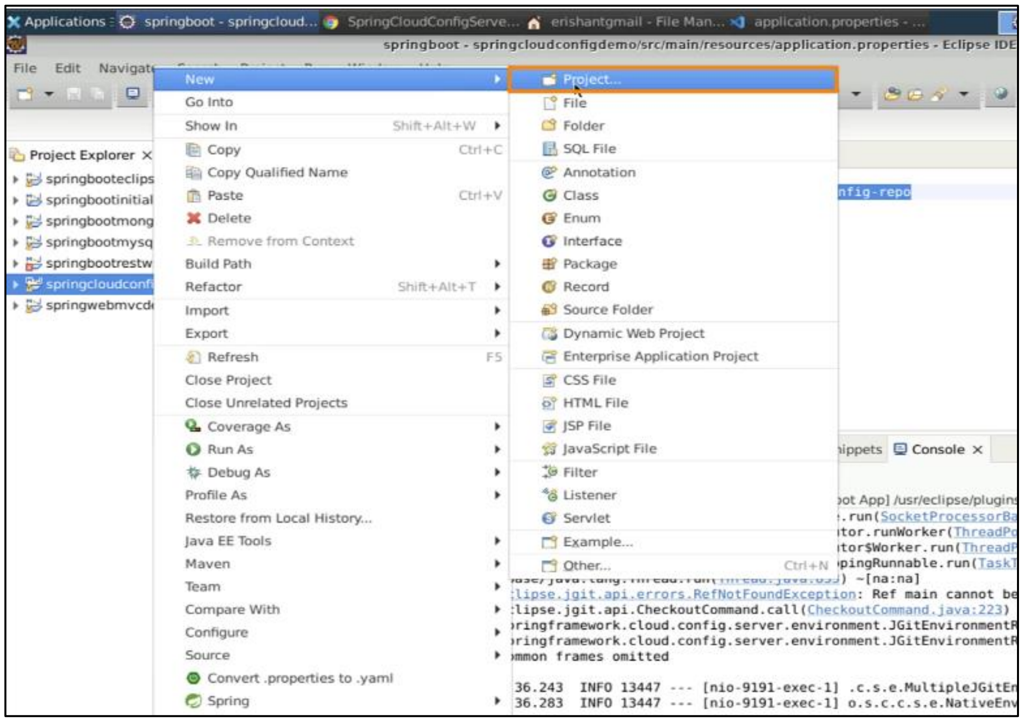
Note: The Git Commit application was created in the previous demo.

1.4 To use the microservice, go to **localhost 9191/application/default Properties**, which it will obtain from a local Git commit


SpringCloudConfigServer
localhost:9191/application
localhost:9191/application/default

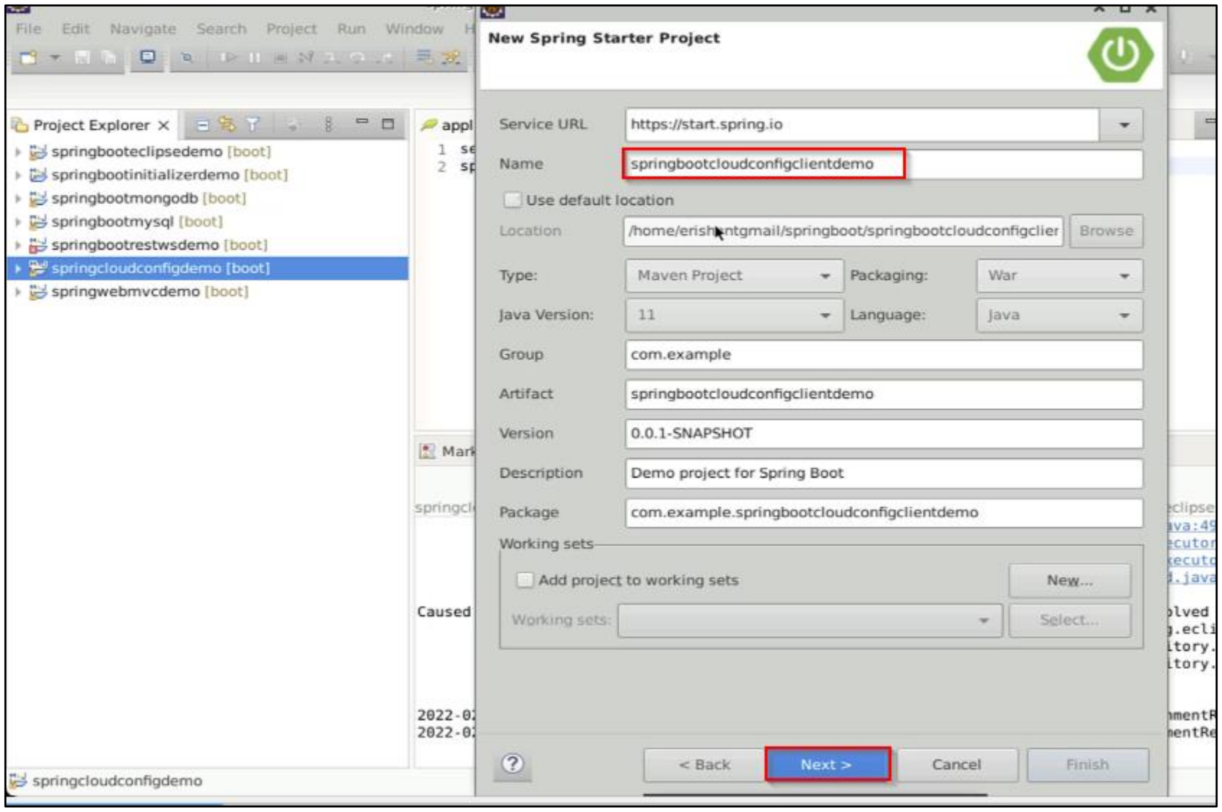
```
{"name":"application","profiles":["default"],"label":null,"version":"42407e8b9ae612e8b35c7310f5f5c1dd110afc0b","state":null,"propertySources":[{"name":"/home/eris
config-repo/application.properties","source":
{"app.name":"WelcomeApp","db.connection.string":"jdbc:mysql://localhost/estore","db.connection.username":"root","db.connection.password":"root123","app.welcome":"
App"}}]}
```

1.5 Right-click on **SpringcloudConfigDemo** project and select **New > Project**



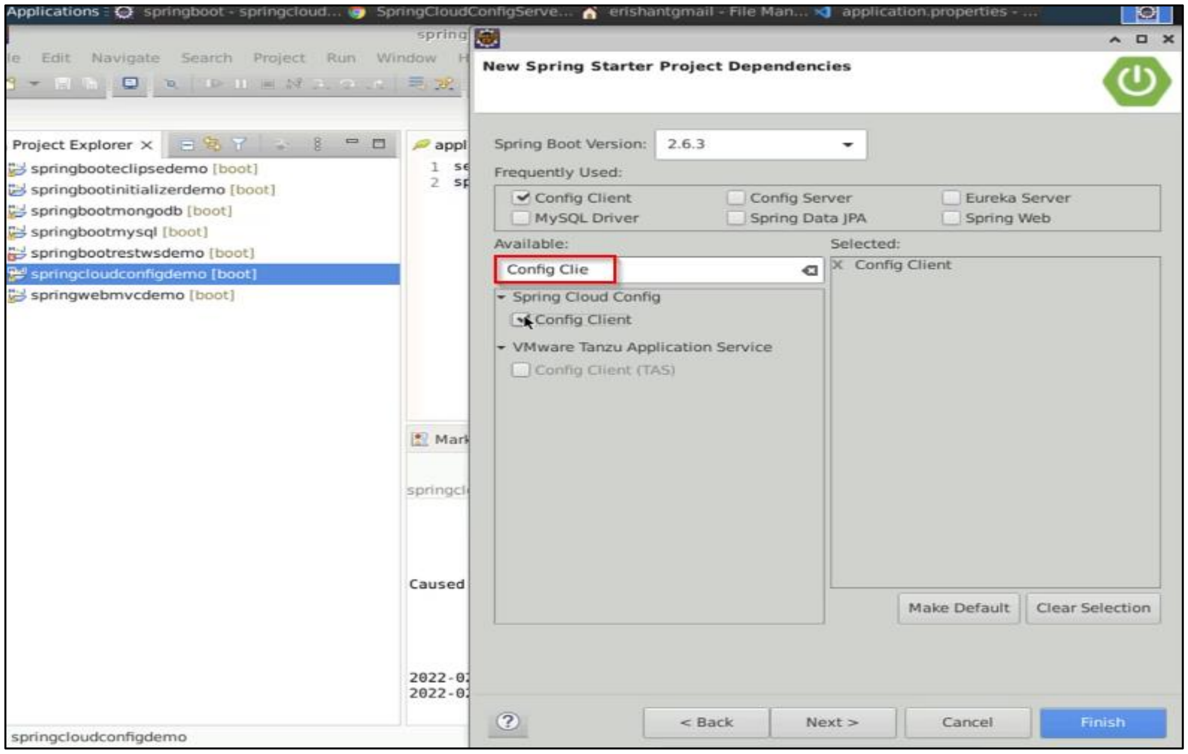
```
oot App] /usr/eclipse/plugins
.run(SocketProcessorBa
itor.runWorker(ThreadPo
itor$Worker.run(ThreadP
pingRunnable.run(TaskT
) ~[na:na]
lipse.jgit.api.errors.RefNotFoundException: Ref main cannot be
lipse.jgit.api.CheckoutCommand.call(CheckoutCommand.java:223)
ringframework.cloud.config.server.environment.JGitEnvironmentR
ringframework.cloud.config.server.environment.JGitEnvironmentR
mmon frames omitted

36.243  INFO 13447 --- [nio-9191-exec-1] .c.s.e.MultipleJGitEn
36.283  INFO 13447 --- [nio-9191-exec-1] o.s.c.c.s.e.NativeEnv
```

1.6 Name the project **springcloudconfigclientdemo** and click **Next**


New Spring Starter Project
Project Explorer
springbooteclipsedemo [boot]
springbootinitializerdemo [boot]
springbootmongodb [boot]
springbootmysql [boot]
springbootrestwsdemo [boot]
springcloudconfigdemo [boot]
springwebmvcdemo [boot]
Use default location
Working sets
Add project to working sets
New...
Select...
< Back
Next >
Cancel
Finish

Service URL: https://start.spring.io
Name: springbootcloudconfigclientdemo
Location: /home/erishentgmail/springboot/springbootcloudconfigclier
Type: Maven Project
Packaging: War
Java Version: 11
Language: Java
Group: com.example
Artifact: springbootcloudconfigclientdemo
Version: 0.0.1-SNAPSHOT
Description: Demo project for Spring Boot
Package: com.example.springbootcloudconfigclientdemo

1.7 To establish a connection to the Spring Cloud Config Server, add the **Config client** Dependency. Click **Finish** to receive the application configuration


Applications
springboot - springcloud...
SpringCloudConfigServe...
erishantgmail - File Man...
application.properties - ...
Edit Navigate Search Project Run Window
Project Explorer
springbooteclipsedemo [boot]
springbootinitializerdemo [boot]
springbootmongodb [boot]
springbootmysql [boot]
springbootrestwsdemo [boot]
springcloudconfigdemo [boot]
springwebmvcdemo [boot]
springcloudconfigdemo
New Spring Starter Project Dependencies
Make Default
Clear Selection
< Back
Next >
Cancel
Finish

Spring Boot Version: 2.6.3

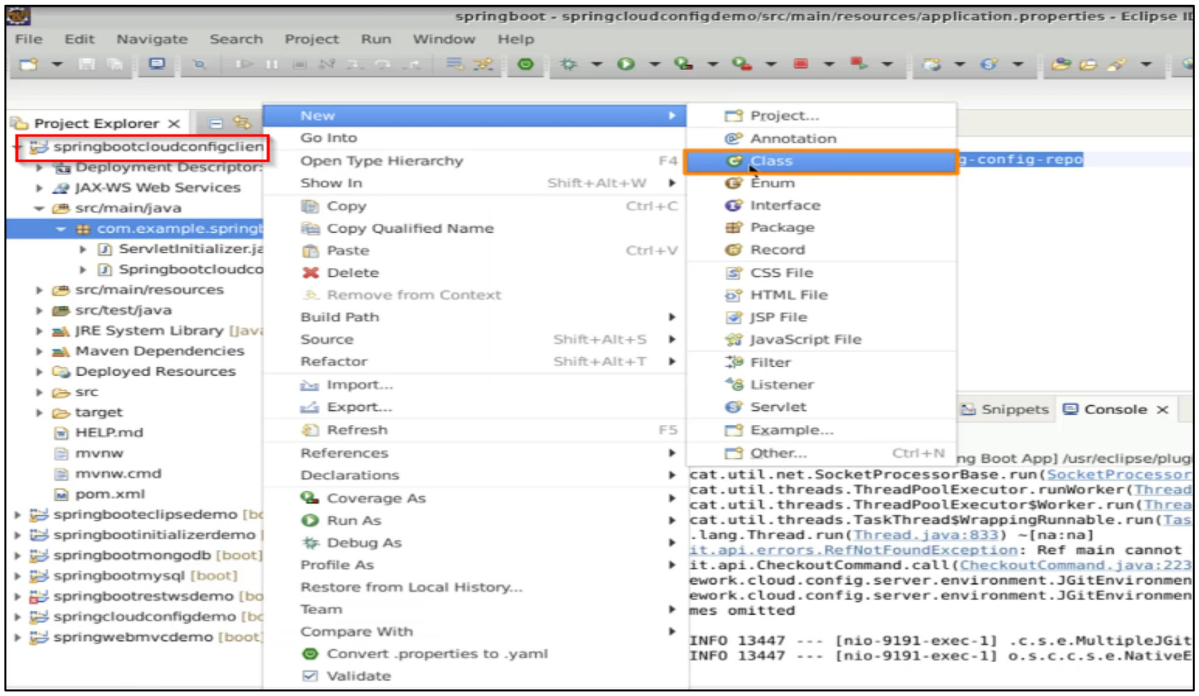
Frequently Used:
[✓ Config Client] Config Server Eureka Server
MySQL Driver Spring Data JPA Spring Web

Available: Config Clie

- Spring Cloud Config
  - [✓ Config Client]
- VMware Tanzu Application Service
  - Config Client (TAS)

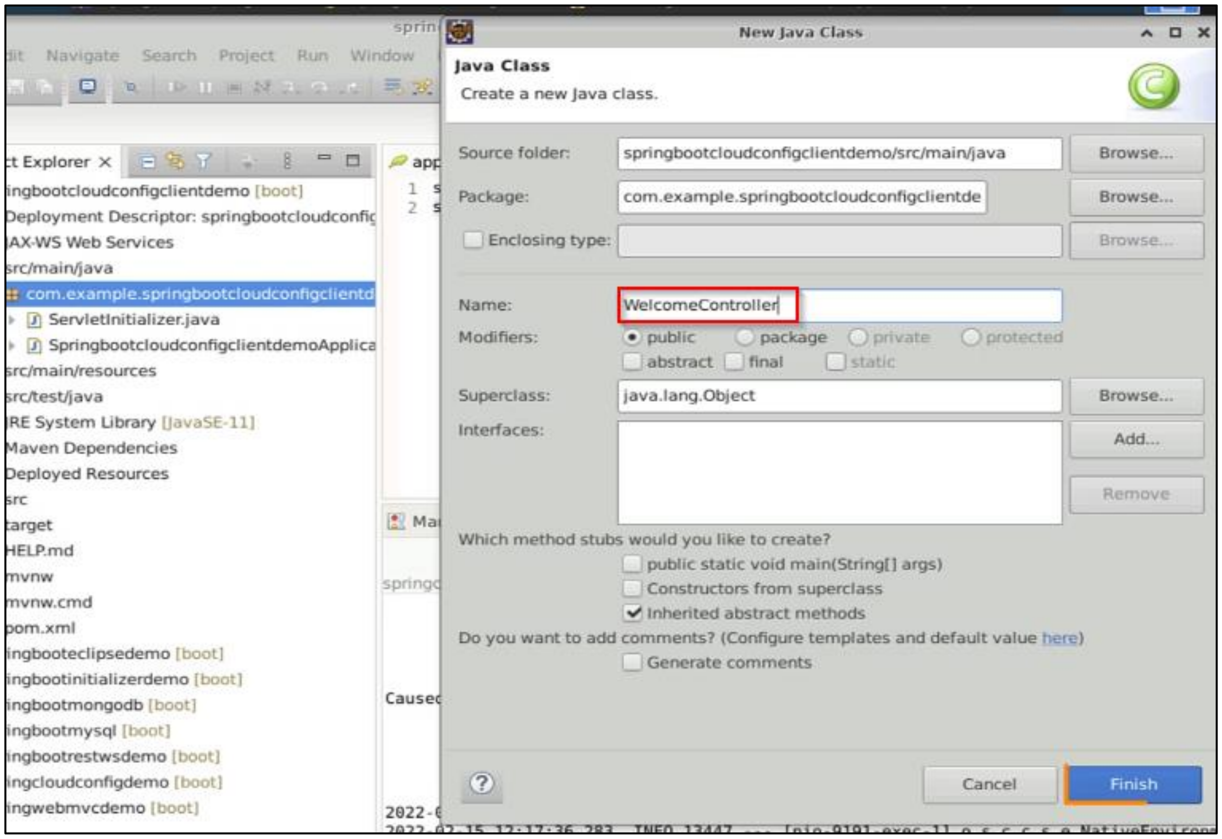
Selected: Config Client

1.8 To create a new class, go to the highlighted **project** and right-click **New > Class**



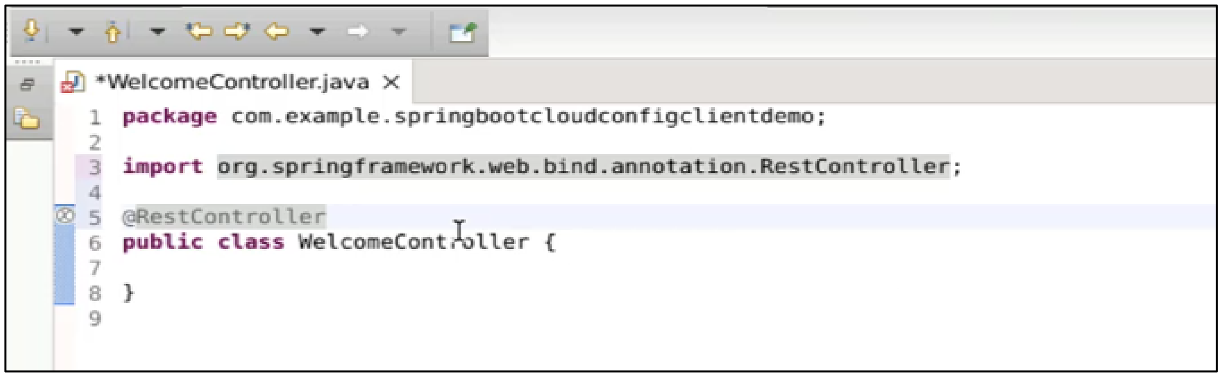
```
ng Boot App] /usr/eclipse/plug
cat.util.net.SocketProcessorBase.run(SocketProcessor
cat.util.threads.ThreadPoolExecutor.runWorker(Thread
cat.util.threads.ThreadPoolExecutor$Worker.run(Threa
cat.util.threads.TaskThread$WrappingRunnable.run(Tas
.lang.Thread.run(Thread.java:833) ~[na:na]
it.api.errors.RefNotFoundException: Ref main cannot
it.api.CheckoutCommand.call(CheckoutCommand.java:223
ework.cloud.config.server.environment.JGitEnvironmen
ework.cloud.config.server.environment.JGitEnvironmen
mes omitted

INFO 13447 --- [nio-9191-exec-1] .c.s.e.MultipleJGit
INFO 13447 --- [nio-9191-exec-1] o.s.c.c.s.e.NativeE
```

1.9 Enter **WelcomeController** as the class name and click **Finish**



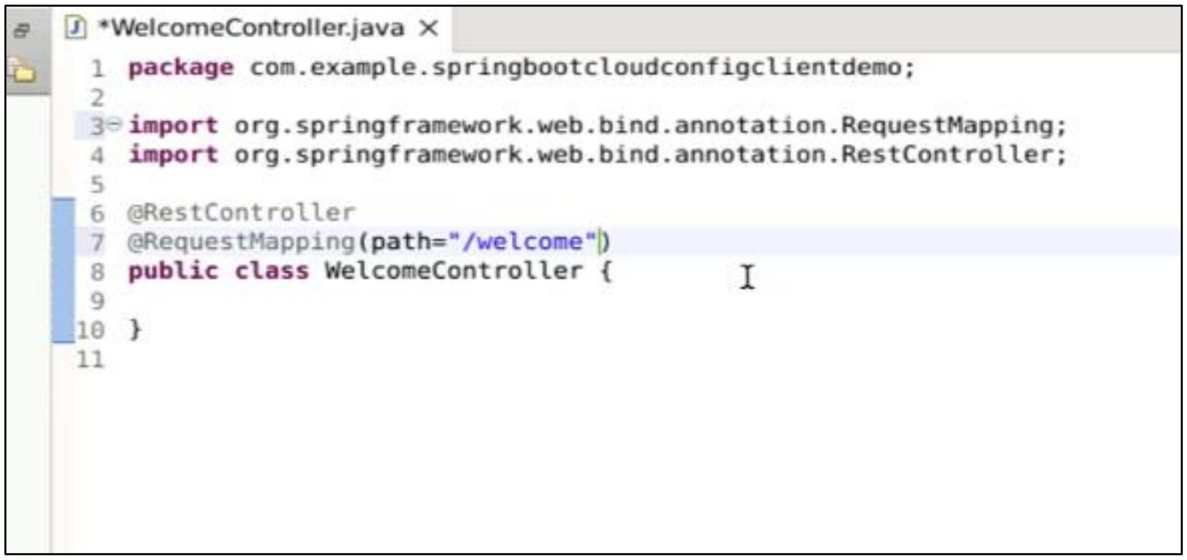
Source folder: springbootcloudconfigclientdemo/src/main/java
Package: com.example.springbootcloudconfigclientde
Enclosing type: [ ]
Name: WelcomeController
Modifiers: [✓ public] package private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
[ ] public static void main(String[] args)
[ ] Constructors from superclass
[✓] Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
[ ] Generate comments

1.10 Create the **WelcomeController** class and annotate it with the **@RestController** flag to provide a request mapping for the spring framework



```
package com.example.springbootcloudconfigclientdemo;

import org.springframework.web.bind.annotation.RestController;

@RestController
public class WelcomeController {

}
```

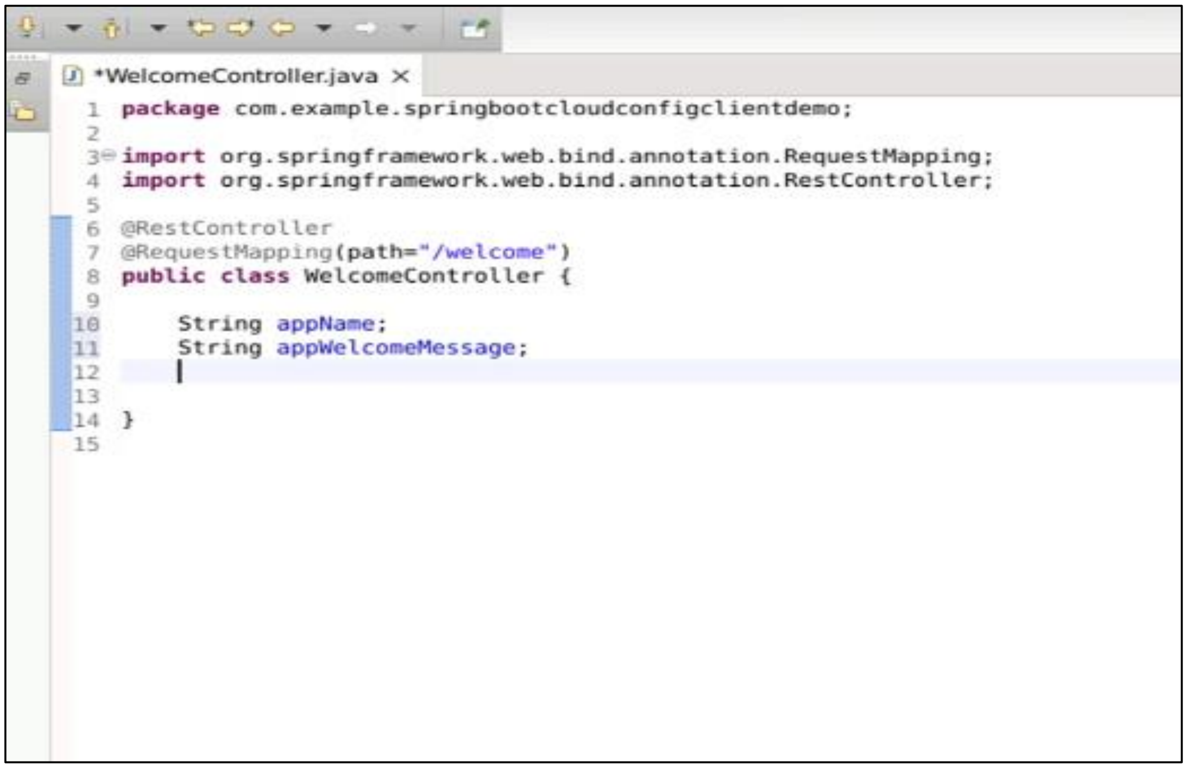
1.11 Choose the path **/Welcome** using **@RequestMapping**



```
*WelcomeController.java
package com.example.springbootcloudconfigclientdemo;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path="/welcome")
public class WelcomeController {

}
```

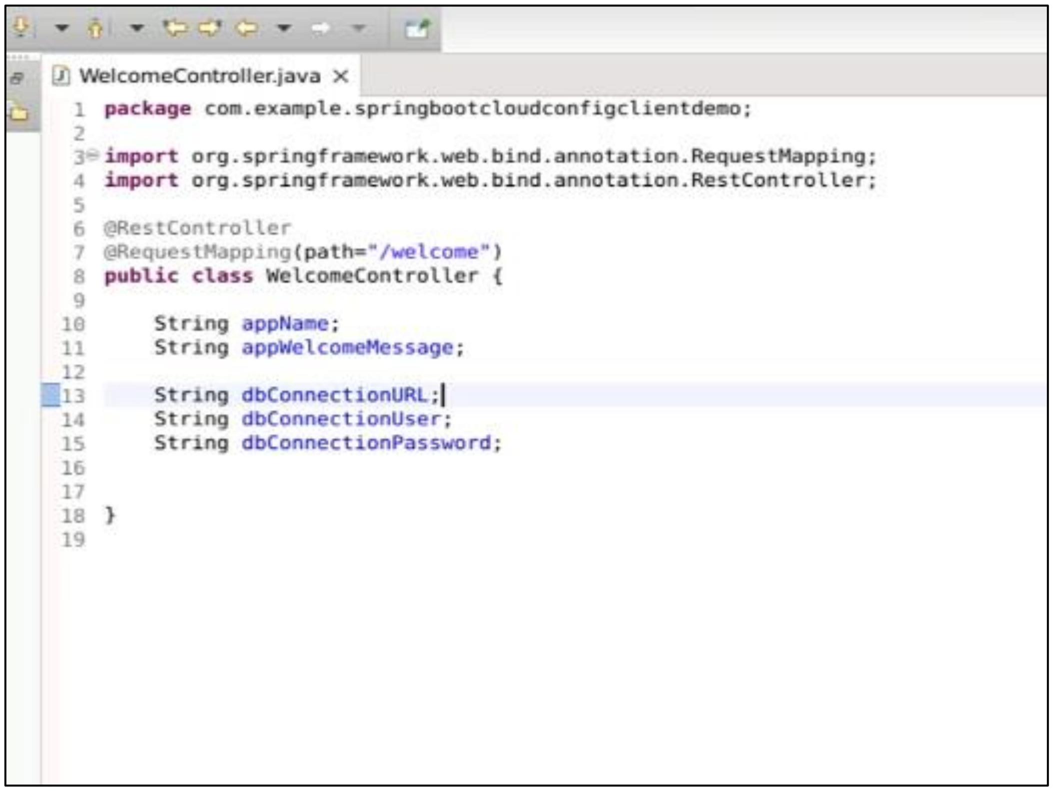
Step 2: Configuring db connection

2.1 To obtain the application properties, create string variables for **appName** and **appWelcomeMessage**



```
*WelcomeController.java
package com.example.springbootcloudconfigclientdemo;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path="/welcome")
public class WelcomeController {

    String appName;
    String appWelcomeMessage;


}
```

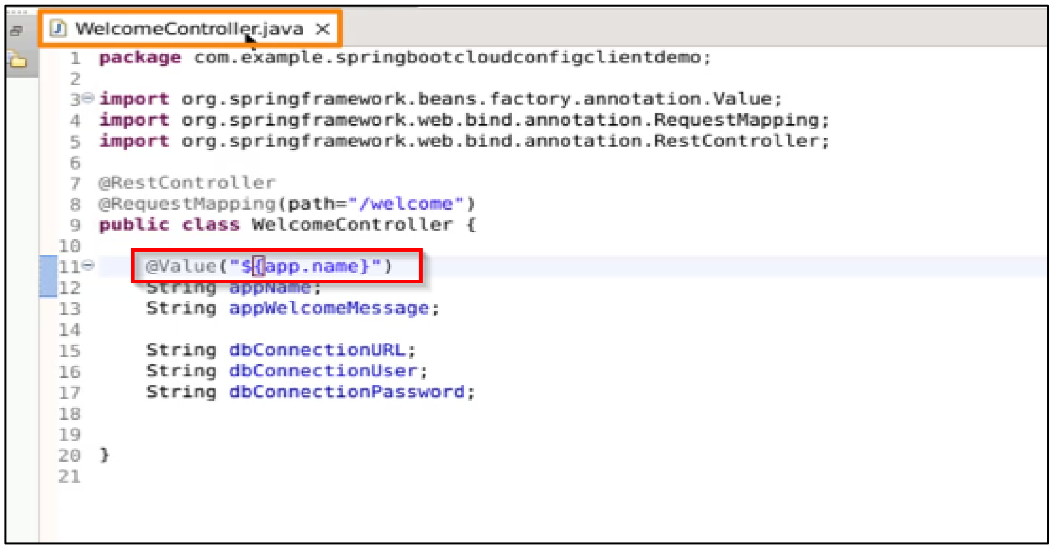
2.2 Create a **dbConnectionURL, dbConnectionUser**, and **dbConnectionPassword** string variable to establish a connection between the three String variables



WelcomeController.java

```
package com.example.springbootcloudconfigclientdemo;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path="/welcome")
public class WelcomeController {

    String appName;
    String appWelcomeMessage;

    String dbConnectionURL;
    String dbConnectionUser;
    String dbConnectionPassword;


}
```

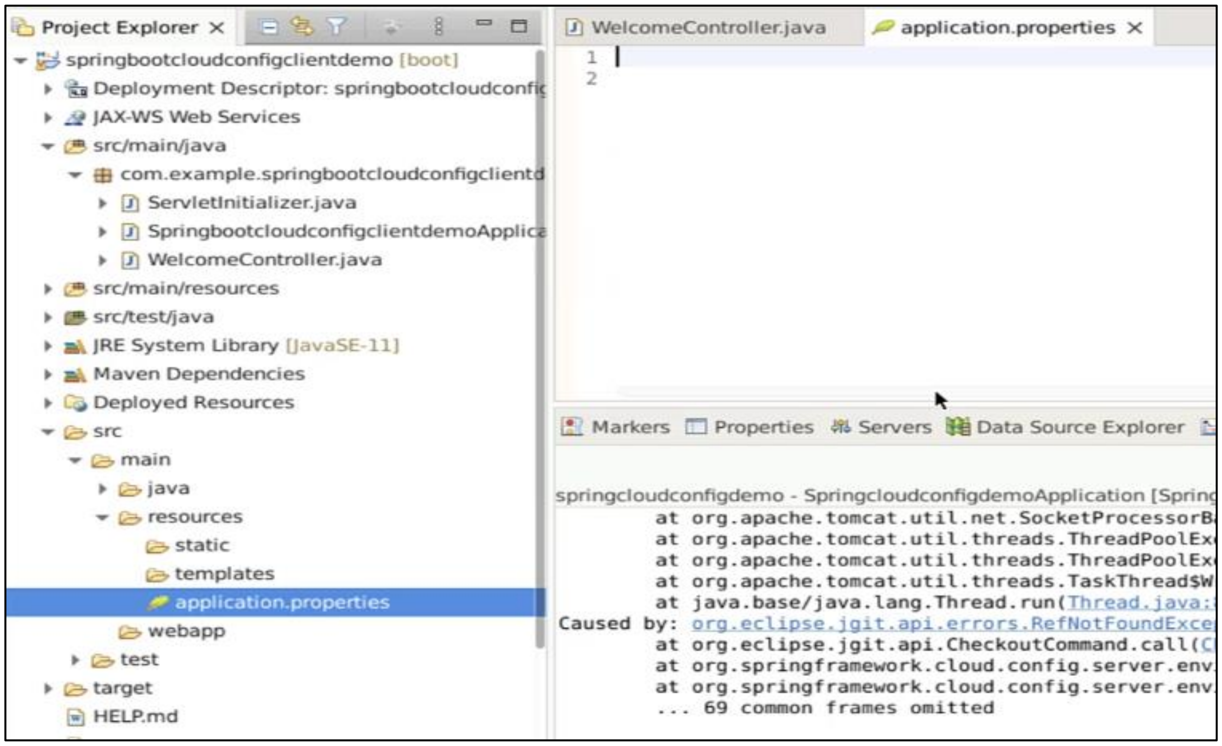
2.3 Set the **@Value("$app.name")** to retrieve variable values from **application.properties**



WelcomeController.java

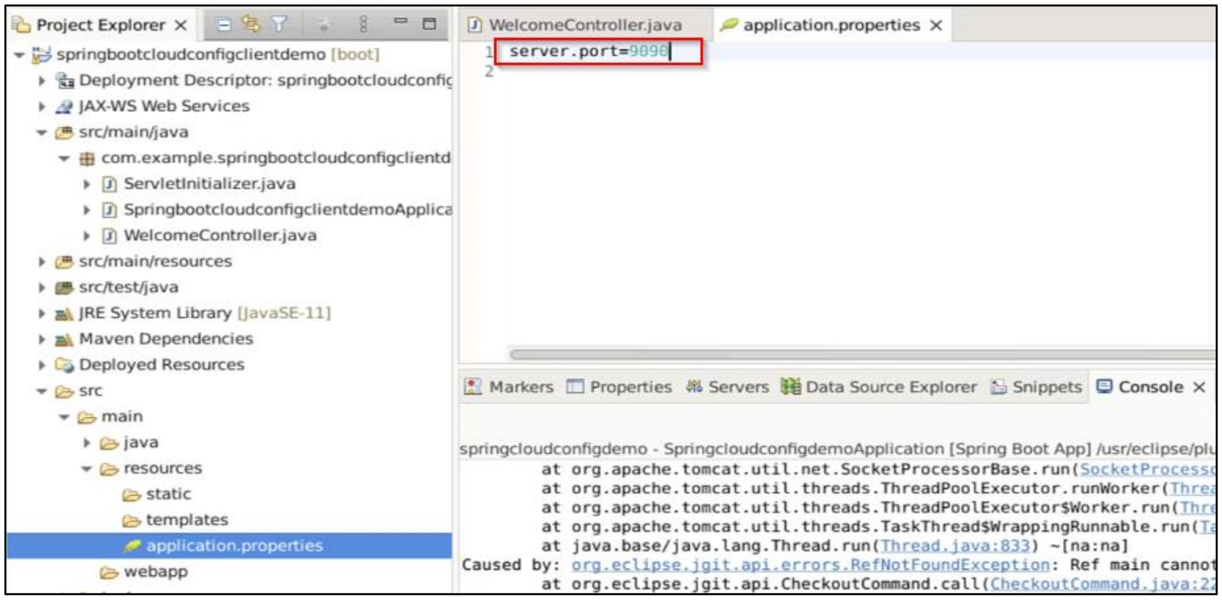
```
package com.example.springbootcloudconfigclientdemo;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path="/welcome")
public class WelcomeController {

    @Value("${app.name}")
    String appName;
    String appWelcomeMessage;

    String dbConnectionURL;
    String dbConnectionUser;
    String dbConnectionPassword;


}
```

2.4 Navigate to **application.properties** to set the Apache Tomcat server



```
springcloudconfigdemo - SpringcloudconfigdemoApplication [Spring
        at org.apache.tomcat.util.net.SocketProcessorB
        at org.apache.tomcat.util.threads.ThreadPoolEx
        at org.apache.tomcat.util.threads.ThreadPoolEx
        at org.apache.tomcat.util.threads.TaskThread$W
        at java.base/java.lang.Thread.run(Thread.java:
Caused by: org.eclipse.jgit.api.errors.RefNotFoundExce
        at org.eclipse.jgit.api.CheckoutCommand.call(C
        at org.springframework.cloud.config.server.env
        at org.springframework.cloud.config.server.env
        ... 69 common frames omitted
```

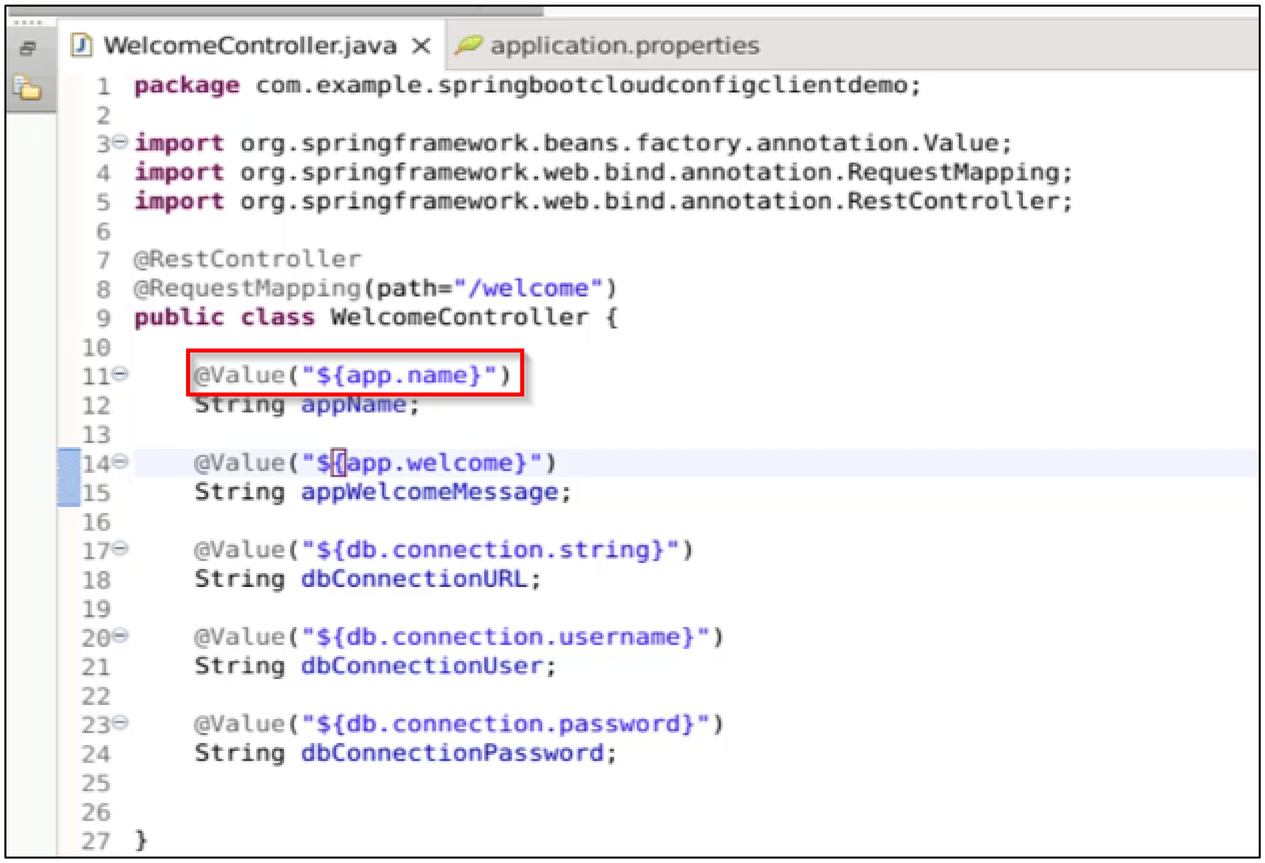
2.5 Add the port number as **server.port=9090,** which will be used to configure the server and read user information



```
server.port=9090
```

```
springcloudconfigdemo - SpringcloudconfigdemoApplication [Spring Boot App] /usr/eclipse/plu
        at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcess
        at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(Threa
        at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(Thre
        at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(Ta
        at java.base/java.lang.Thread.run(Thread.java:833) ~[na:na]
Caused by: org.eclipse.jgit.api.errors.RefNotFoundException: Ref main cannot
        at org.eclipse.jgit.api.CheckoutCommand.call(CheckoutCommand.java:22
```

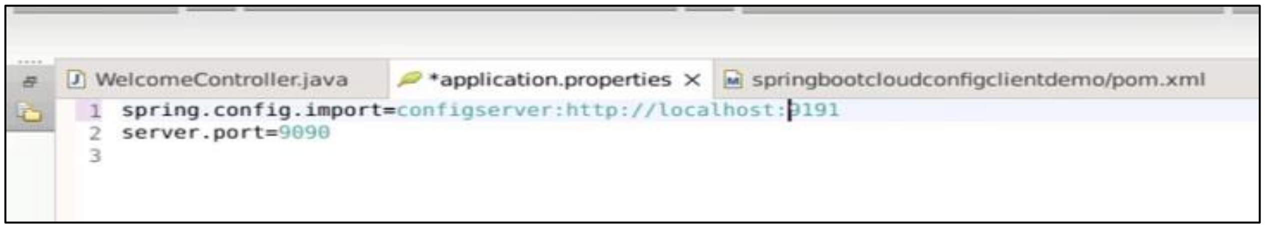
2.6 All String variables should be annotated with the **@value** to read the String value



WelcomeController.java | application.properties

```java
package com.example.springbootcloudconfigclientdemo;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path="/welcome")
public class WelcomeController {

    @Value("${app.name}")
    String appName;

    @Value("${app.welcome}")
    String appWelcomeMessage;

    @Value("${db.connection.string}")
    String dbConnectionURL;

    @Value("${db.connection.username}")
    String dbConnectionUser;

    @Value("${db.connection.password}")
    String dbConnectionPassword;


}
```

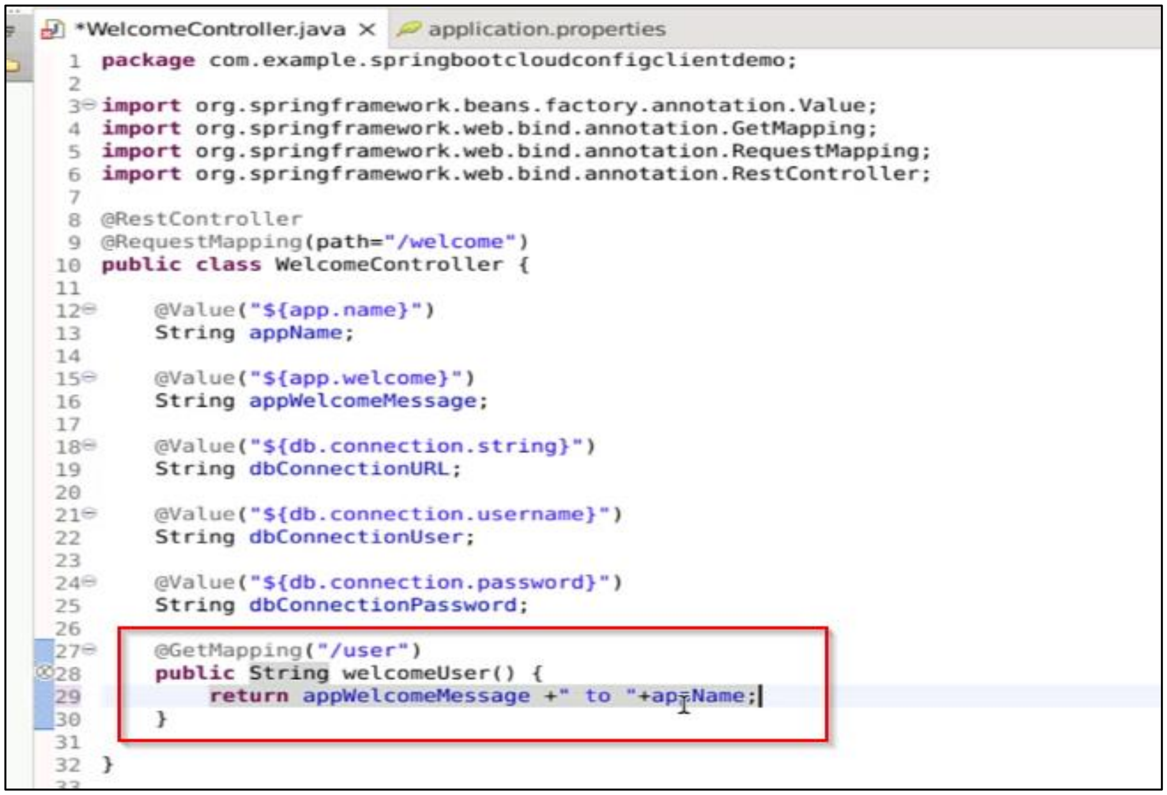
2.7 Create a localhost **configserver** and import from s**pringbootcloudconfigclientdemo** in **application.properties** to deploy the application in the browser



WelcomeController.java | *application.properties | springbootcloudconfigclientdemo/pom.xml

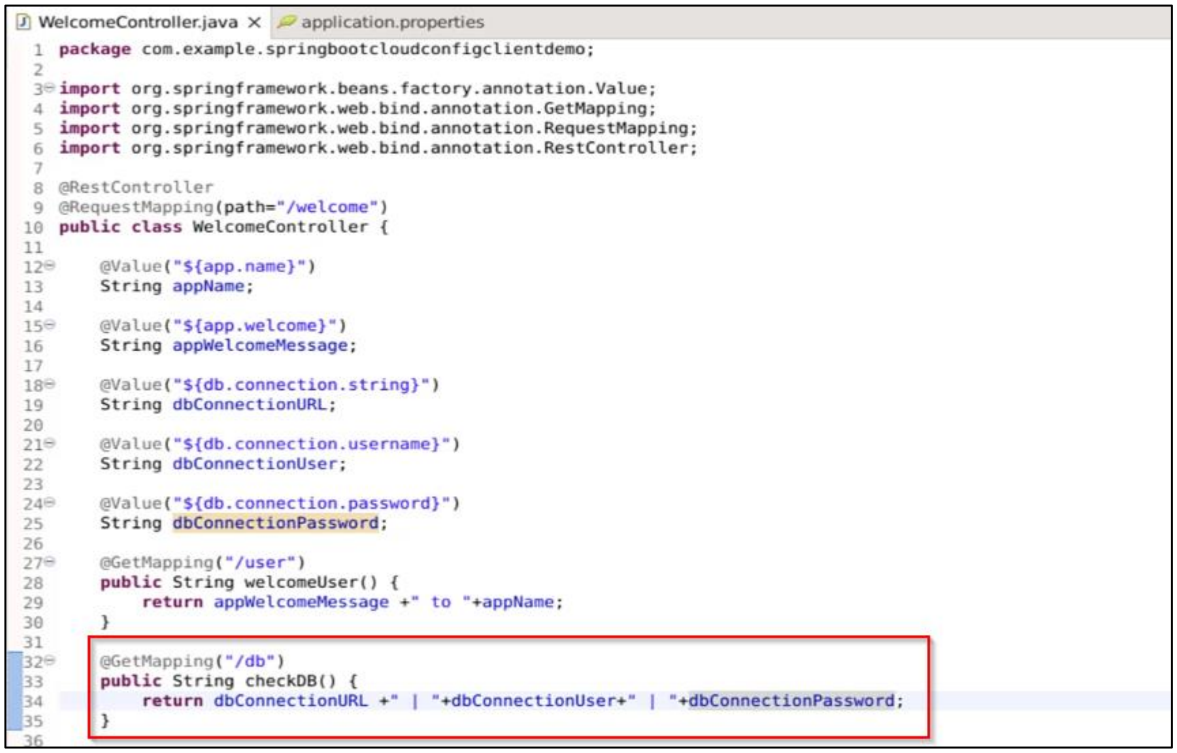
```
spring.config.import=configserver:http://localhost:9191
server.port=9090
```

2.8 Create the sting method as **welcomeUser**, add the return method as **appWelcomeMessage**, and concatenate with the **appName**



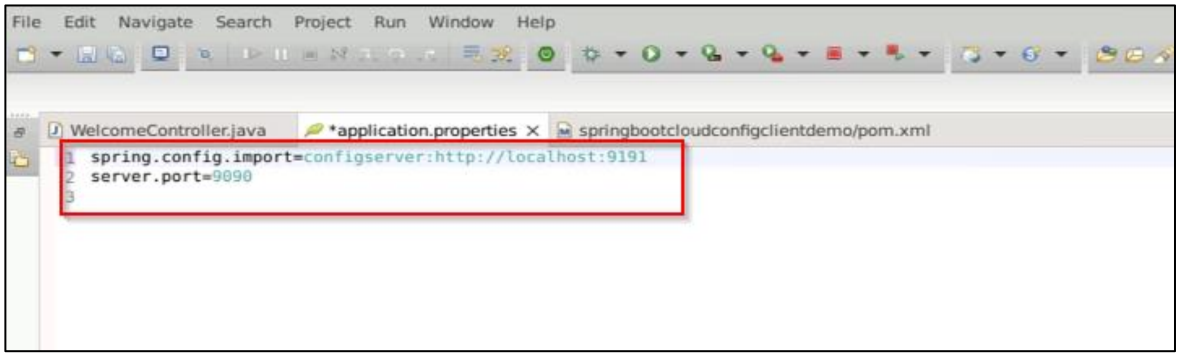
```
package com.example.springbootcloudconfigclientdemo;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path="/welcome")
public class WelcomeController {

    @Value("${app.name}")
    String appName;

    @Value("${app.welcome}")
    String appWelcomeMessage;

    @Value("${db.connection.string}")
    String dbConnectionURL;

    @Value("${db.connection.username}")
    String dbConnectionUser;

    @Value("${db.connection.password}")
    String dbConnectionPassword;

    @GetMapping("/user")
    public String welcomeUser() {
        return appWelcomeMessage +" to "+appName;
    }

}
```

2.9 Create the string method as **checkdb**, add the return method as **dbConnectionURL,** and concatenate them with **dbConnectionUser** and **dbConnectionPassword**



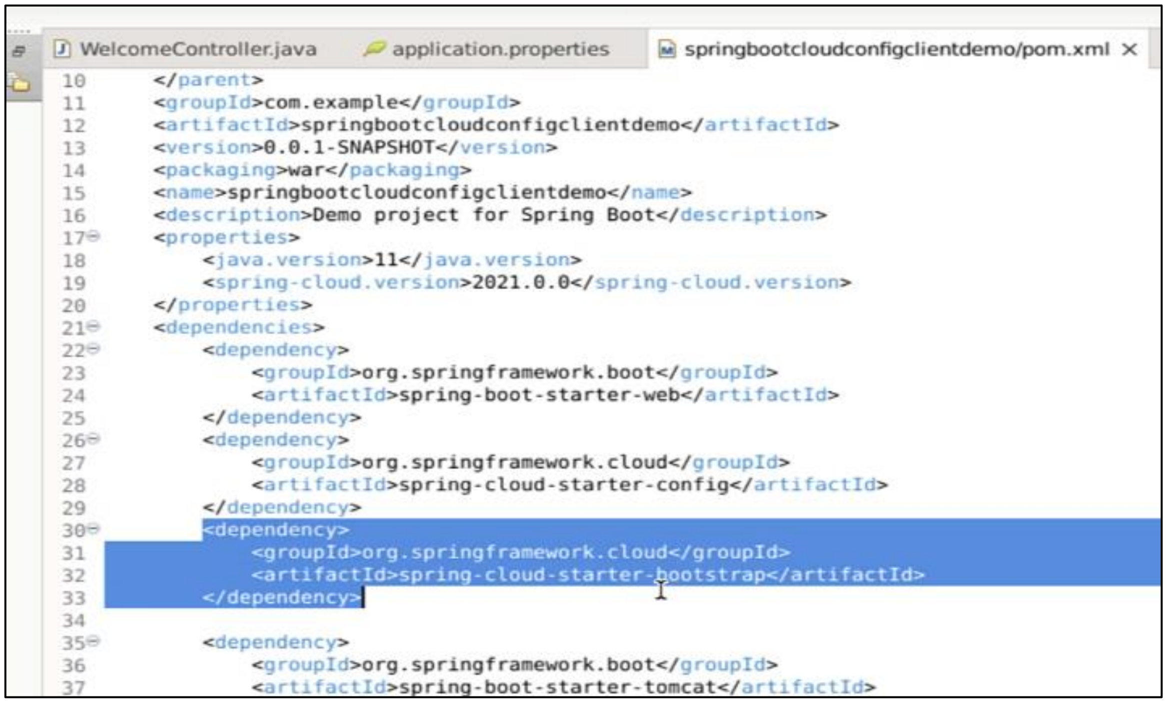
```java
package com.example.springbootcloudconfigclientdemo;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path="/welcome")
public class WelcomeController {

    @Value("${app.name}")
    String appName;

    @Value("${app.welcome}")
    String appWelcomeMessage;

    @Value("${db.connection.string}")
    String dbConnectionURL;

    @Value("${db.connection.username}")
    String dbConnectionUser;

    @Value("${db.connection.password}")
    String dbConnectionPassword;

    @GetMapping("/user")
    public String welcomeUser() {
        return appWelcomeMessage +" to "+appName;
    }

    @GetMapping("/db")
    public String checkDB() {
        return dbConnectionURL +" | "+dbConnectionUser+" | "+dbConnectionPassword;
    }

```

2.10 Create spring **configserver** as **localhost:9191** to get a server port of 9090



```
spring.config.import=configserver:http://localhost:9191
server.port=9090

```

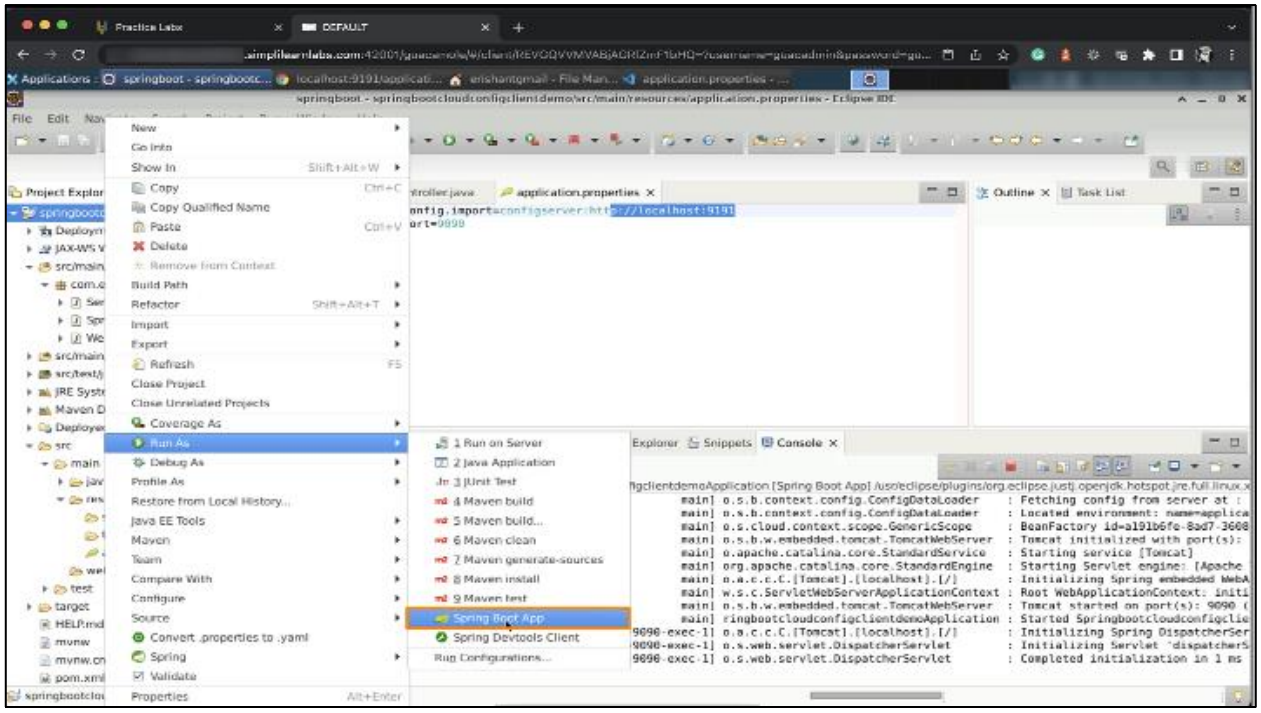
2.11 Add the spring cloud starter bootstrap configuration to the **pom.xml** files as a dependency



```
    </parent>
    <groupId>com.example</groupId>
    <artifactId>springbootcloudconfigclientdemo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>springbootcloudconfigclientdemo</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>11</java.version>
        <spring-cloud.version>2021.0.0</spring-cloud.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-config</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-bootstrap</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
```

Note: Add bootstrap dependency to your project for the **pom.xml** file

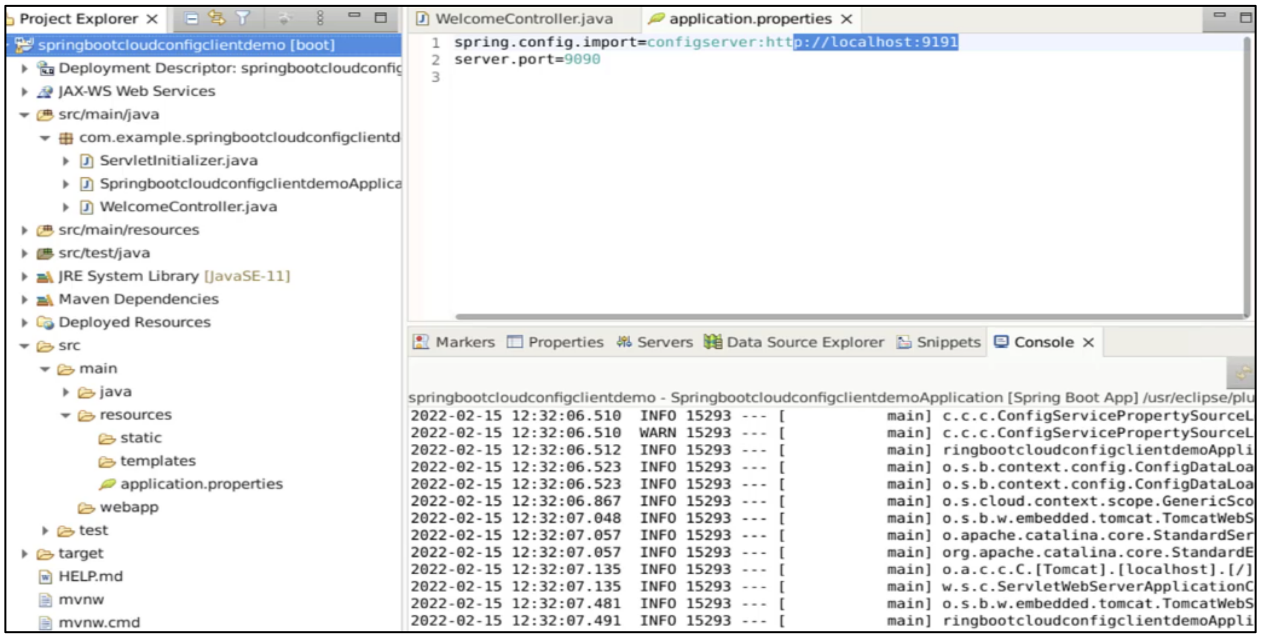
2.12 **Right-click** on the project and click **Run As > Spring Boot App**



```
onfig.import=configserver:http://localhost:9191
ort=9090
```

```
figclientdemoApplication [Spring Boot App] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x
        main] o.s.b.context.config.ConfigDataLoader      : Fetching config from server at :
        main] o.s.b.context.config.ConfigDataLoader      : Located environment: name=applica
        main] o.s.cloud.context.scope.GenericScope       : BeanFactory id=a191b6fe-8ad7-3608
        main] o.s.b.w.embedded.tomcat.TomcatWebServer    : Tomcat initialized with port(s):
        main] o.apache.catalina.core.StandardService     : Starting service [Tomcat]
        main] org.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache
        main] o.a.c.c.C.[Tomcat].[localhost].[/]         : Initializing Spring embedded WebA
        main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initi
        main] o.s.b.w.embedded.tomcat.TomcatWebServer    : Tomcat started on port(s): 9090 (
        main] ringbootcloudconfigclientdemoApplication : Started Springbootcloudconfigclie
9090-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]         : Initializing Spring DispatcherSer
9090-exec-1] o.s.web.servlet.DispatcherServlet          : Initializing Servlet 'dispatcherS
9090-exec-1] o.s.web.servlet.DispatcherServlet          : Completed initialization in 1 ms
```

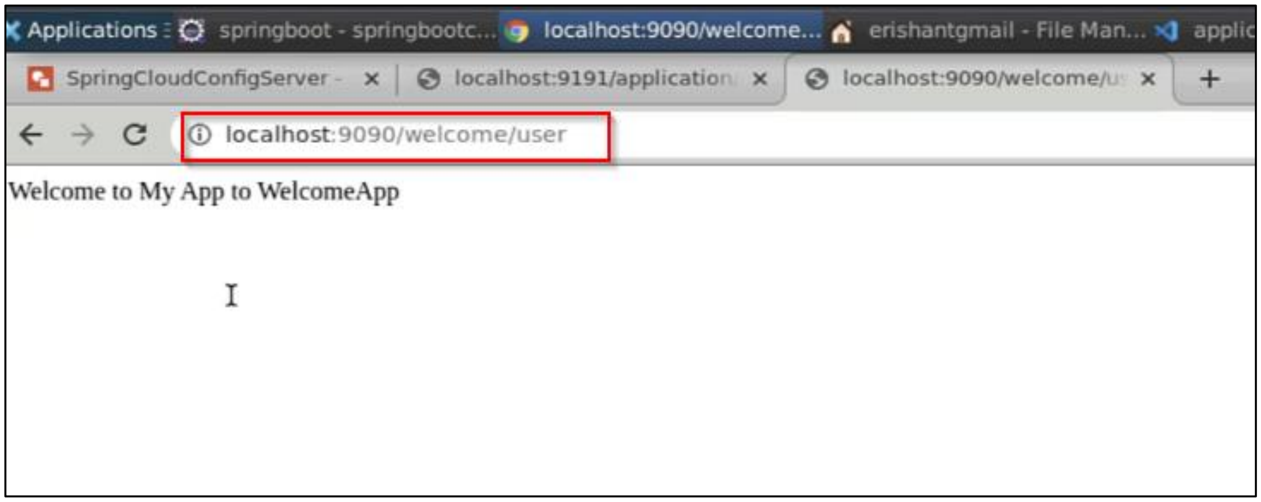
The program starts using a fresh **Tomcat Server.**



```
spring.config.import=configserver:http://localhost:9191
server.port=9090
```

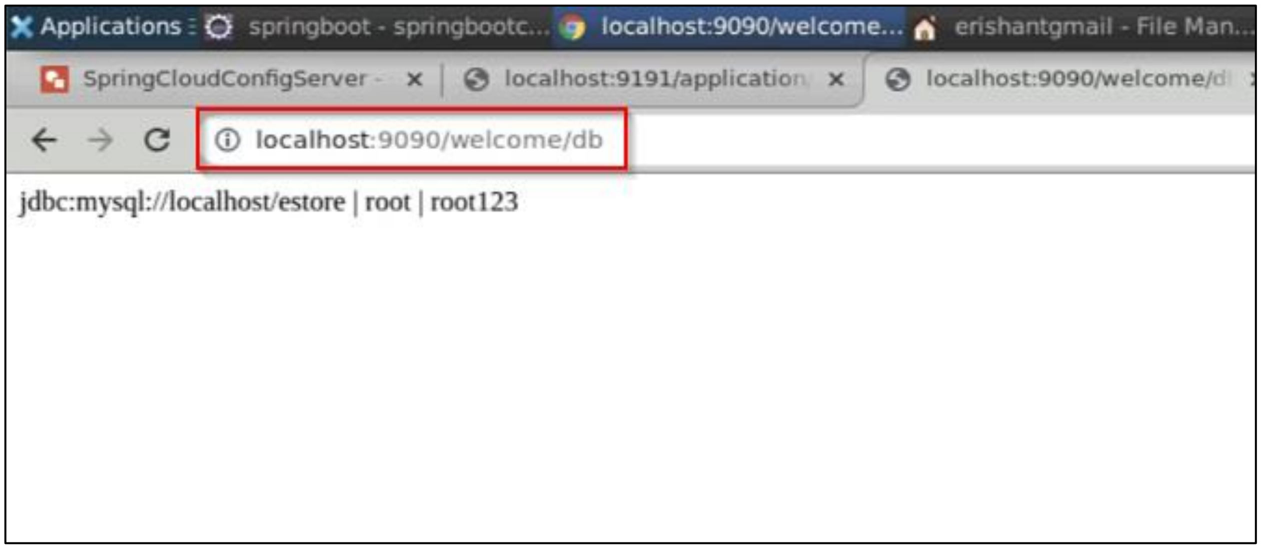
```
springbootcloudconfigclientdemo - SpringbootcloudconfigclientdemoApplication [Spring Boot App] /usr/eclipse/plu
2022-02-15 12:32:06.510  INFO 15293 --- [           main] c.c.c.ConfigServicePropertySourceL
2022-02-15 12:32:06.510  WARN 15293 --- [           main] c.c.c.ConfigServicePropertySourceL
2022-02-15 12:32:06.512  INFO 15293 --- [           main] ringbootcloudconfigclientdemoAppli
2022-02-15 12:32:06.523  INFO 15293 --- [           main] o.s.b.context.config.ConfigDataLoa
2022-02-15 12:32:06.523  INFO 15293 --- [           main] o.s.b.context.config.ConfigDataLoa
2022-02-15 12:32:06.867  INFO 15293 --- [           main] o.s.cloud.context.scope.GenericSco
2022-02-15 12:32:07.048  INFO 15293 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebS
2022-02-15 12:32:07.057  INFO 15293 --- [           main] o.apache.catalina.core.StandardSer
2022-02-15 12:32:07.057  INFO 15293 --- [           main] org.apache.catalina.core.StandardE
2022-02-15 12:32:07.135  INFO 15293 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]
2022-02-15 12:32:07.135  INFO 15293 --- [           main] w.s.c.ServletWebServerApplicationC
2022-02-15 12:32:07.481  INFO 15293 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebS
2022-02-15 12:32:07.491  INFO 15293 --- [           main] ringbootcloudconfigclientdemoAppli
```

2.13 Open the browser, **click** on a new tab, type **localhost:9090/welcome/user**, and the application will launch on the web


Applications
springboot - springbootc...
localhost:9090/welcome...
erishantgmail - File Man...
SpringCloudConfigServer -
localhost:9191/application
localhost:9090/welcome/u
localhost:9090/welcome/user

```
Welcome to My App to WelcomeApp
```

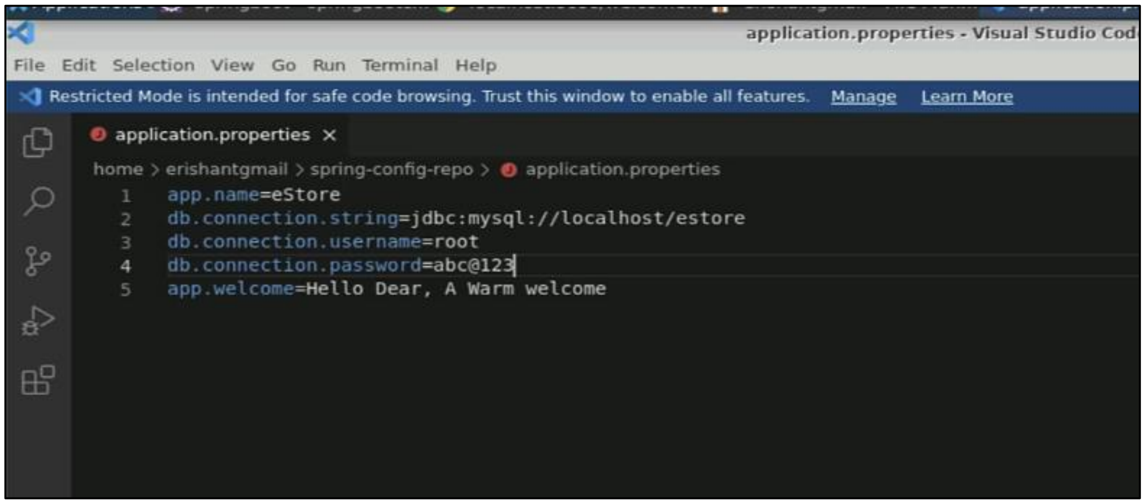
2.14 Create a second **localhost:9090/welcome/db** so that the **username** and **password** are present in the web application


Applications
springboot - springbootc...
localhost:9090/welcome...
erishantgmail - File Man...
SpringCloudConfigServer -
localhost:9191/application
localhost:9090/welcome/d
localhost:9090/welcome/db

```
jdbc:mysql://localhost/estore | root | root123
```

Step 3: Running the application

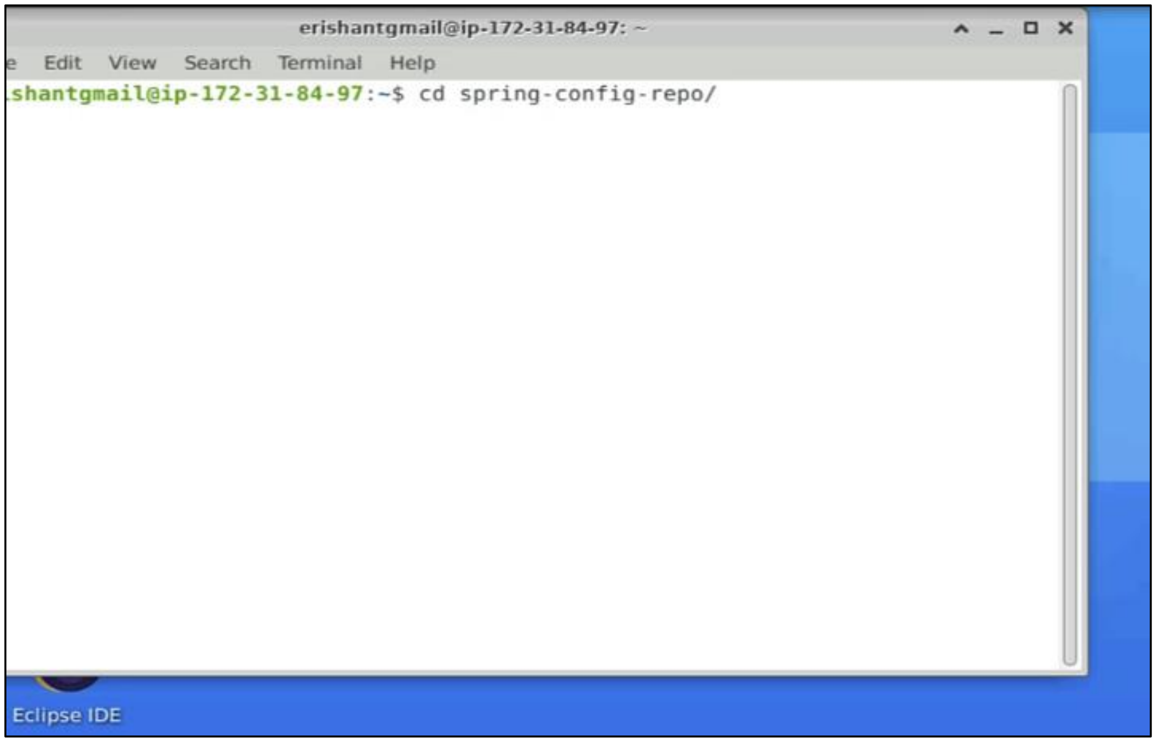
3.1 Rename the **app.name, app.welcome**, and **db.connection**.**Password** to fetch the **application.properties** file



application.properties - Visual Studio Cod
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
application.properties
home > erishantgmail > spring-config-repo > application.properties

```
app.name=eStore
db.connection.string=jdbc:mysql://localhost/estore
db.connection.username=root
db.connection.password=abc@123
app.welcome=Hello Dear, A Warm welcome
```

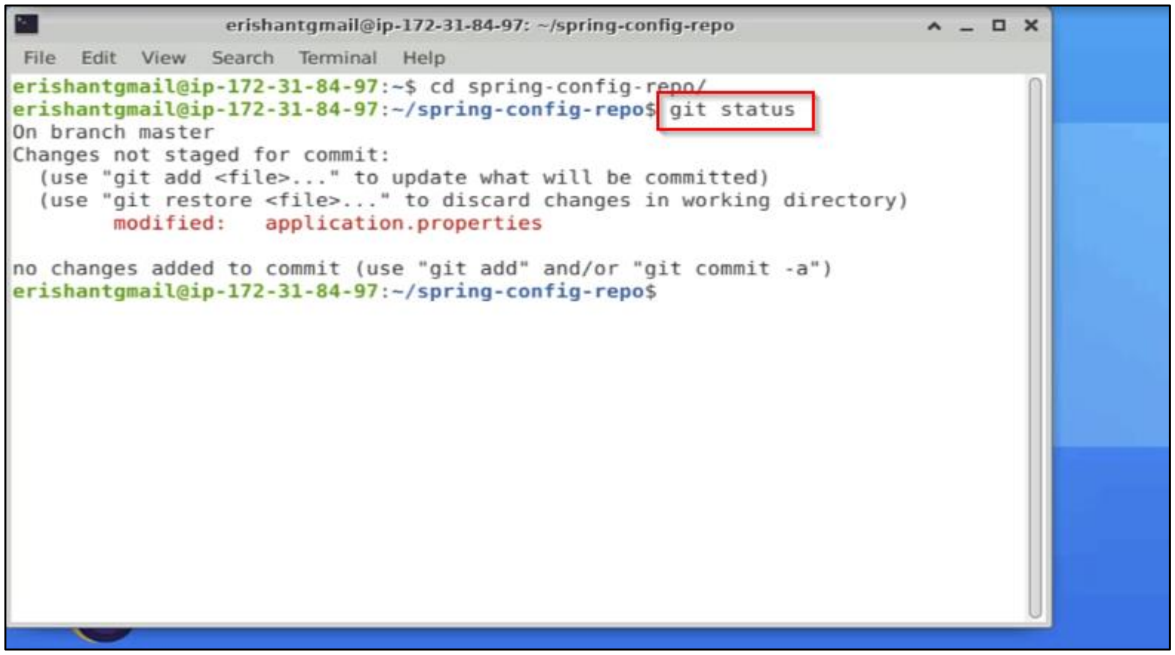
3.2 Access the local system terminal and type **cd spring-config-repo/**



erishantgmail@ip-172-31-84-97: ~
e Edit View Search Terminal Help
Eclipse IDE

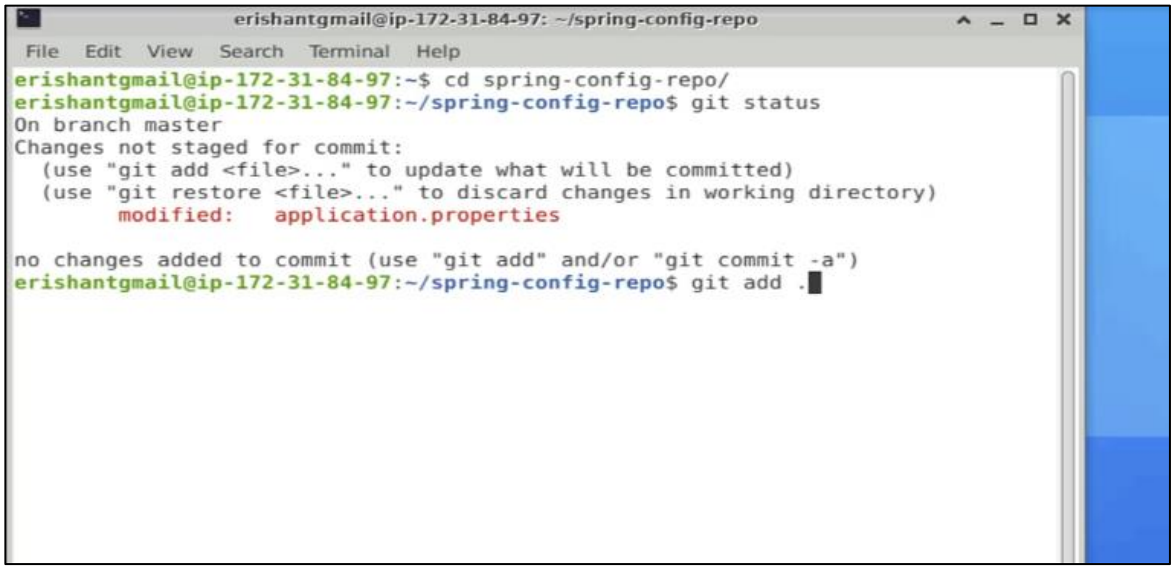
```
shantgmail@ip-172-31-84-97:~$ cd spring-config-repo/
```

3.3 To see if a file has been edited, enter the command **Git status**



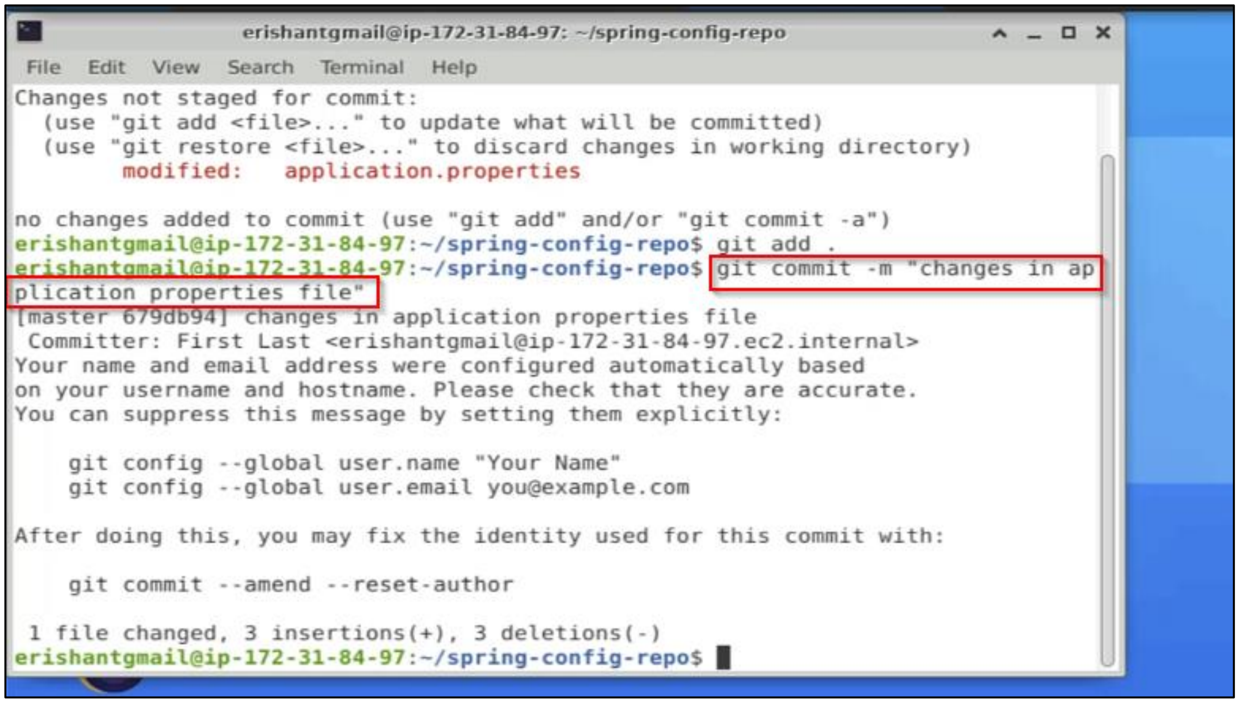
```
erishantgmail@ip-172-31-84-97:~$ cd spring-config-repo/
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   application.properties

no changes added to commit (use "git add" and/or "git commit -a")
erishantgmail@ip-172-31-84-97:~/spring-config-repo$
```

3.4 In the terminal, use the command **git add .**



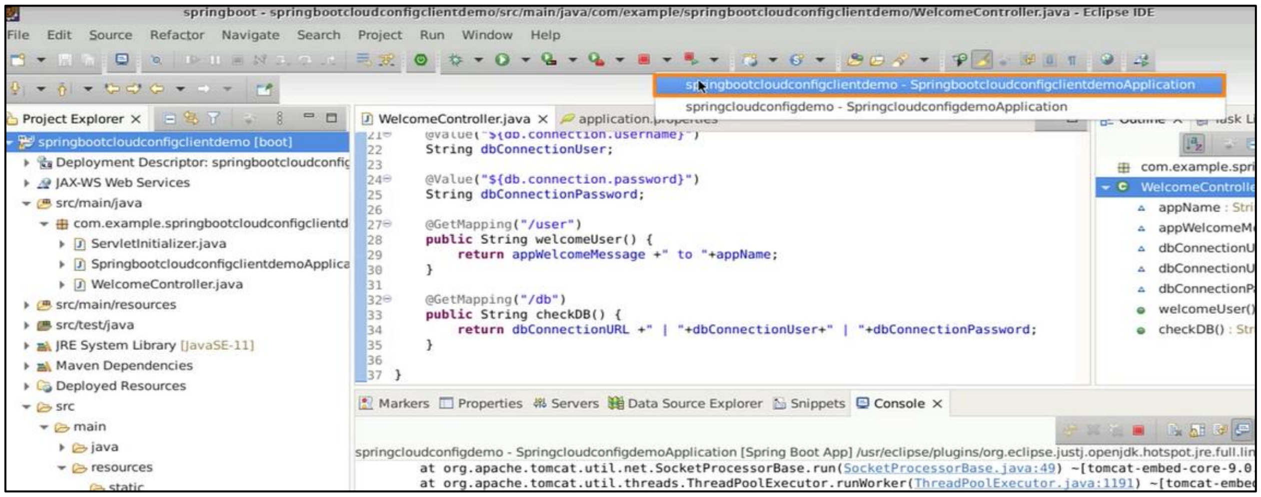
```
erishantgmail@ip-172-31-84-97:~$ cd spring-config-repo/
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   application.properties

no changes added to commit (use "git add" and/or "git commit -a")
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git add .
```

3.5 Enter the command **git commit -m "changes in application properties files"**


erishantgmail@ip-172-31-84-97: ~/spring-config-repo
File Edit View Search Terminal Help

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   application.properties

no changes added to commit (use "git add" and/or "git commit -a")
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git add .
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git commit -m "changes in ap
plication properties file"
[master 679db94] changes in application properties file
 Committer: First Last <erishantgmail@ip-172-31-84-97.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 3 insertions(+), 3 deletions(-)
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ 
```

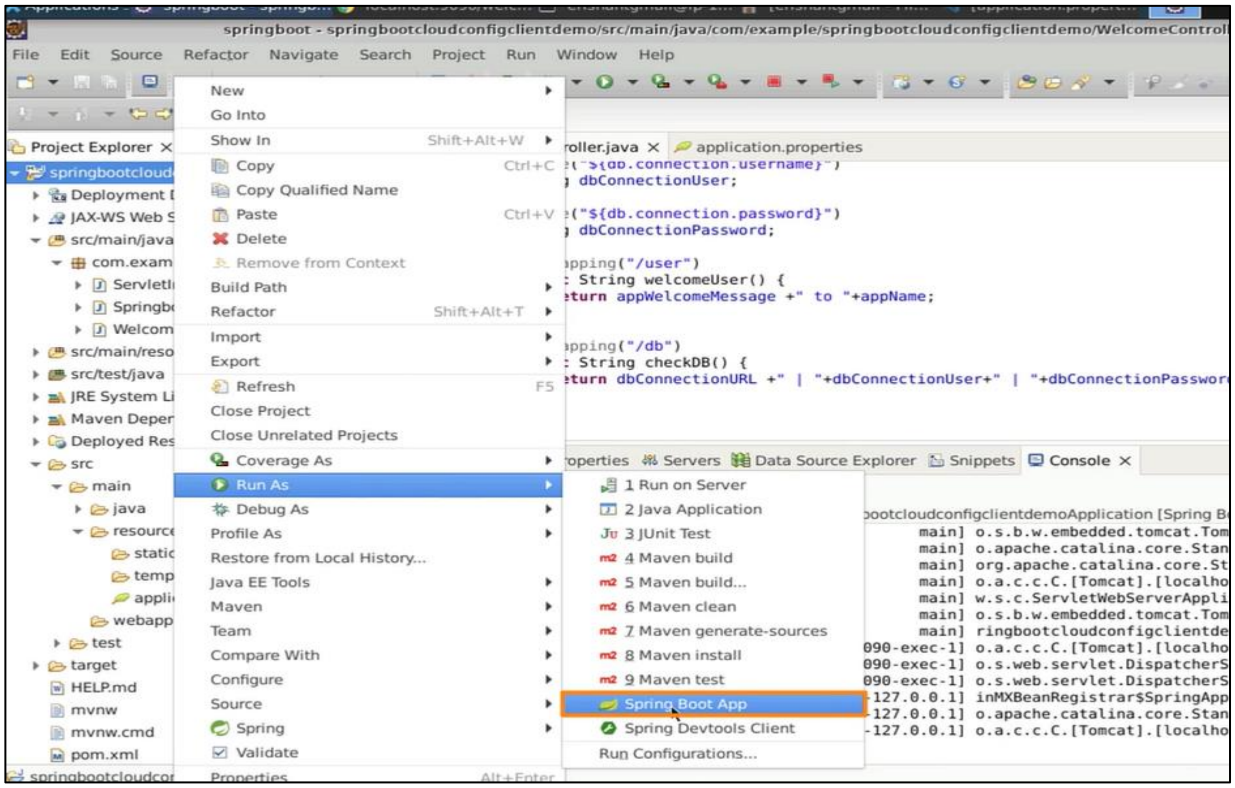
3.6 Choose the **Tomcat Server** and terminate the **Tomcat Server** to restart the new Tomcat Server


springboot - springbootcloudconfigclientdemo/src/main/java/com/example/springbootcloudconfigclientdemo/WelcomeController.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
springbootcloudconfigclientdemo - SpringbootcloudconfigclientdemoApplication
springcloudconfigdemo - SpringcloudconfigdemoApplication
Project Explorer
springbootcloudconfigclientdemo [boot]
Deployment Descriptor: springbootcloudconfig
JAX-WS Web Services
src/main/java
com.example.springbootcloudconfigclientd
ServletInitializer.java
SpringbootcloudconfigclientdemoApplica
WelcomeController.java
src/main/resources
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
Deployed Resources
src
main
java
resources
static
WelcomeController.java
application.properties
com.example.spri
WelcomeControlle
appName : Stri
appWelcomeM
dbConnectionU
dbConnectionU
dbConnectionP
welcomeUser()
checkDB() : Str
Markers Properties Servers Data Source Explorer Snippets Console

```
    @Value("${db.connection.username}")
    String dbConnectionUser;

    @Value("${db.connection.password}")
    String dbConnectionPassword;

    @GetMapping("/user")
    public String welcomeUser() {
        return appWelcomeMessage +" to "+appName;
    }

    @GetMapping("/db")
    public String checkDB() {
        return dbConnectionURL +" | "+dbConnectionUser+" | "+dbConnectionPassword;
    }

}
```

```
springcloudconfigdemo - SpringcloudconfigdemoApplication [Spring Boot App] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.lin
        at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49) ~[tomcat-embed-core-9.0
        at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1191) ~[tomcat-embed
```

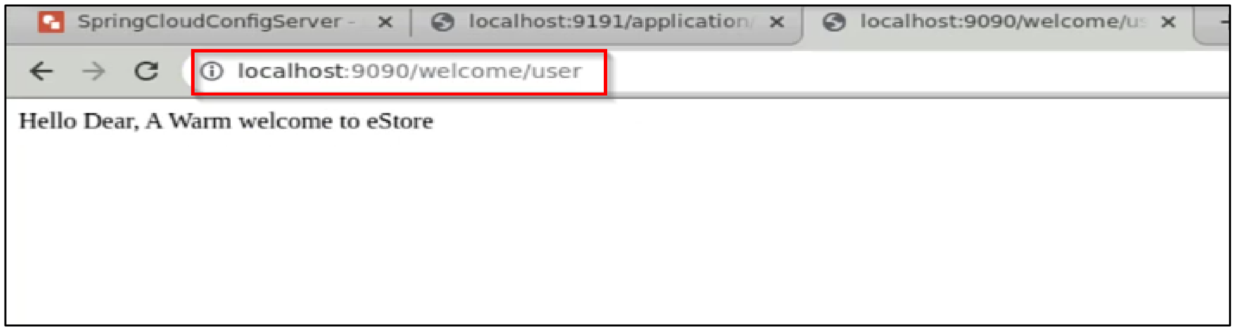
3.7 To start a new Apache Tomcat server, **right-click** on the project **springbootcloudconfigclientdemo** and select **Run As > Spring Boot App**



```
e("${db.connection.username}")
 dbConnectionUser;

e("${db.connection.password}")
 dbConnectionPassword;

apping("/user")
: String welcomeUser() {
eturn appWelcomeMessage +" to "+appName;

apping("/db")
: String checkDB() {
eturn dbConnectionURL +" | "+dbConnectionUser+" | "+dbConnectionPasswor
```

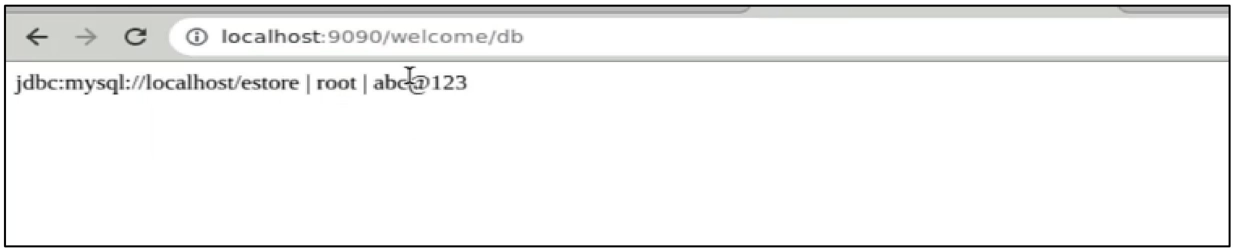
```
ootcloudconfigclientdemoApplication [Spring B
       main] o.s.b.w.embedded.tomcat.Tom
       main] o.apache.catalina.core.Stan
       main] org.apache.catalina.core.St
       main] o.a.c.c.C.[Tomcat].[localho
       main] w.s.c.ServletWebServerAppli
       main] o.s.b.w.embedded.tomcat.Tom
       main] ringbootcloudconfigclientde
090-exec-1] o.a.c.c.C.[Tomcat].[localho
090-exec-1] o.s.web.servlet.DispatcherS
090-exec-1] o.s.web.servlet.DispatcherS
-127.0.0.1] inMXBeanRegistrar$SpringApp
-127.0.0.1] o.apache.catalina.core.Stan
-127.0.0.1] o.a.c.c.C.[Tomcat].[localho
```

3.8 To launch the application, open a browser and type **localhost:9090/welcome/user**


SpringCloudConfigServer -
localhost:9191/application
localhost:9090/welcome/u
localhost:9090/welcome/user

Hello Dear, A Warm welcome to eStore

3.9 To check the **username** and **password** for the localhost estore, open a new tab in the browser and type **localhost:9090/welcome/db**



```
jdbc:mysql://localhost/estore | root | abc@123
```