

Lesson 04 Demo 01

Spring Cloud Configuration

Objective: To understand how to manage an application through Spring Cloud configuration

Tool required: Eclipse IDE and Visual Studio Code

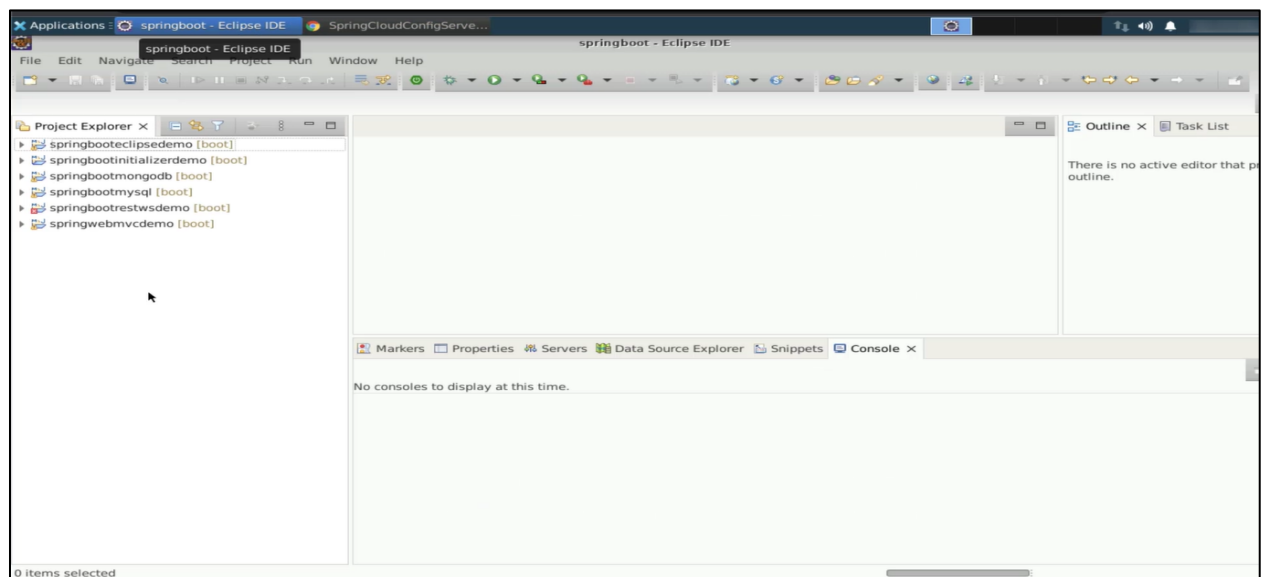
Prerequisites: None

Steps to be followed:

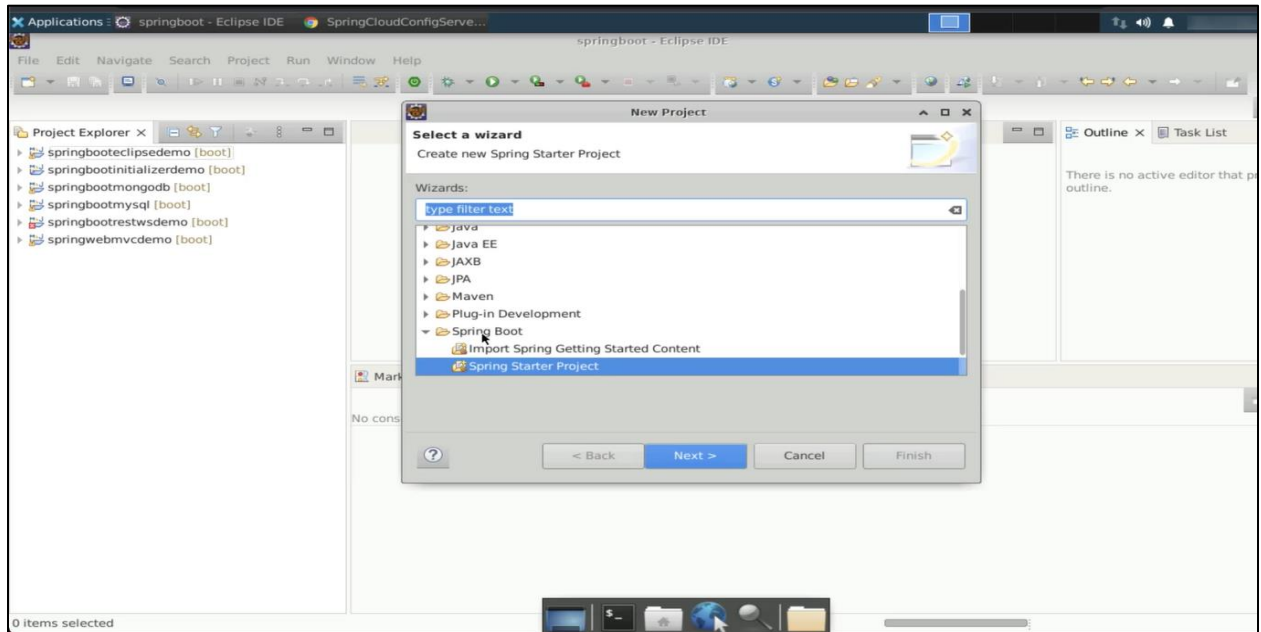
1. Creating a Spring Boot project with cloud configuration
2. Configuring the Spring Cloud in Eclipse IDE
3. Configuring the Git repository
4. Setting up the Git repository path for the Spring Cloud Configuration
5. Deploying the Spring Cloud Configuration project

Step 1: Creating a Spring Boot project with cloud configuration

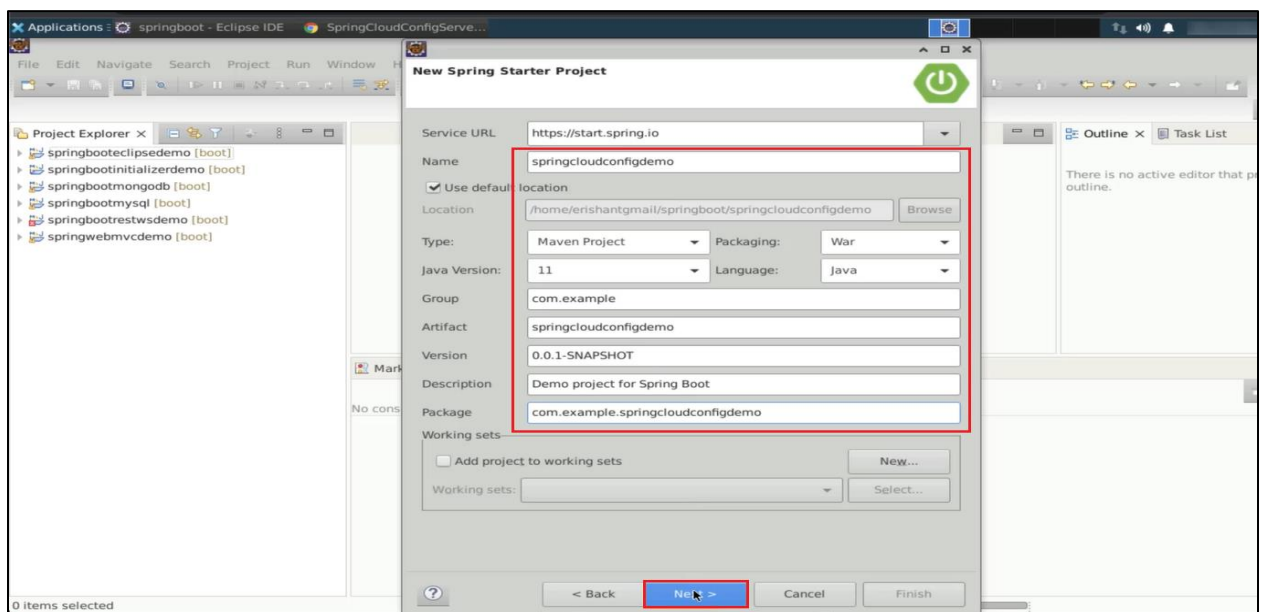
1.1 Open Eclipse IDE to create a new Spring Boot project



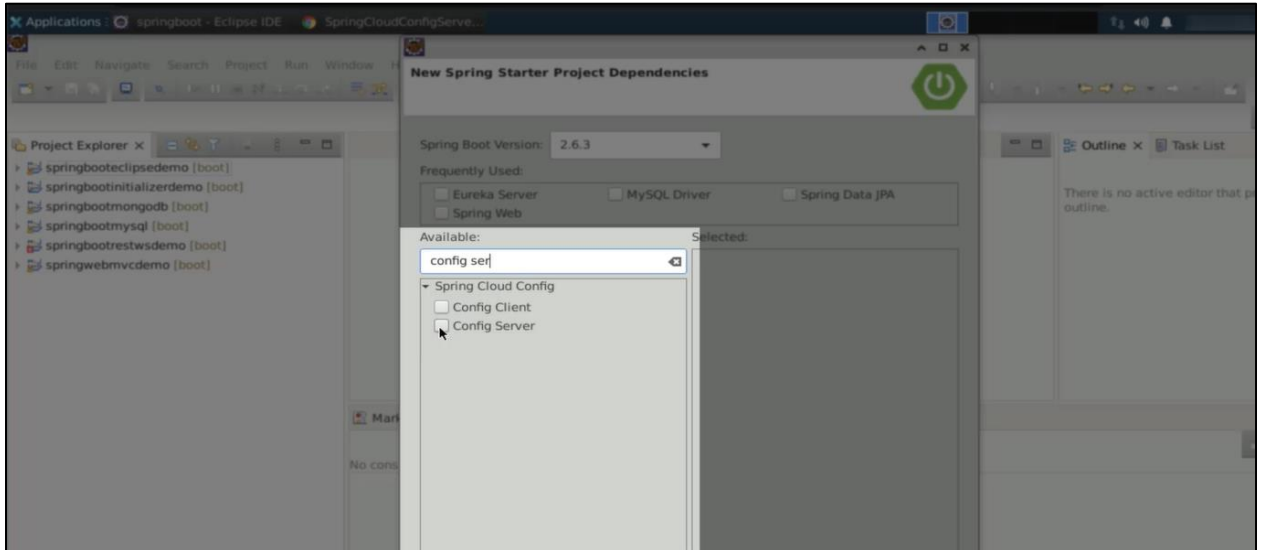
- 1.2 Right-click on **Eclipse IDE**, select **New > Project > Spring Boot > Spring Starter Project**.
Now, click **Next**



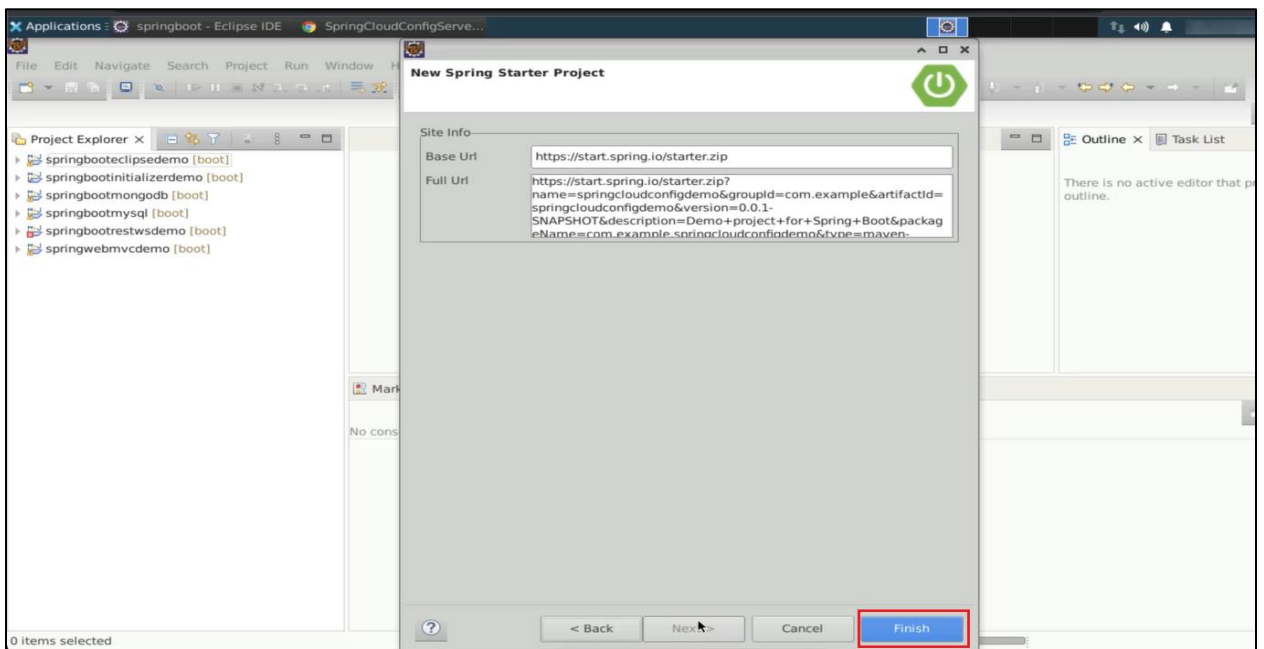
- 1.3 In the **New Spring Starter Project** wizard, provide a name for the Spring Boot project, like **springcloudconfigdemo**. Set the **Packaging** type as **War**, **Package Id** as **com.example.springcloudconfigdemo**, and click **Next**



1.4 To create the dependencies for the project, search for the **config server**. Under **Spring Cloud Config**, select **Config Server** and click **Next**

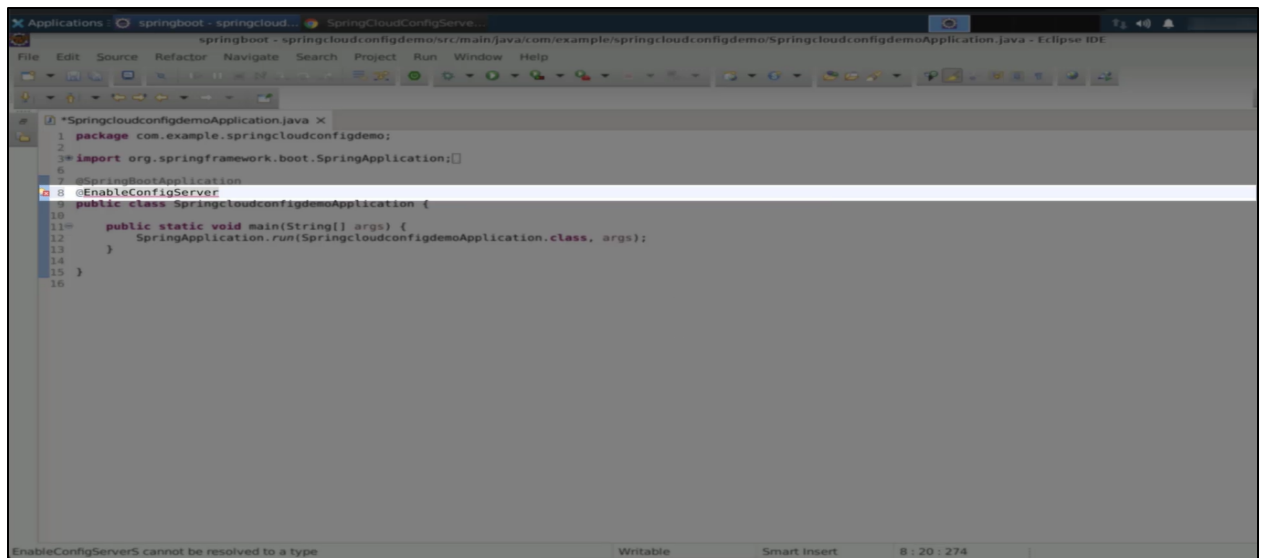


1.5 Now, click **Finish**

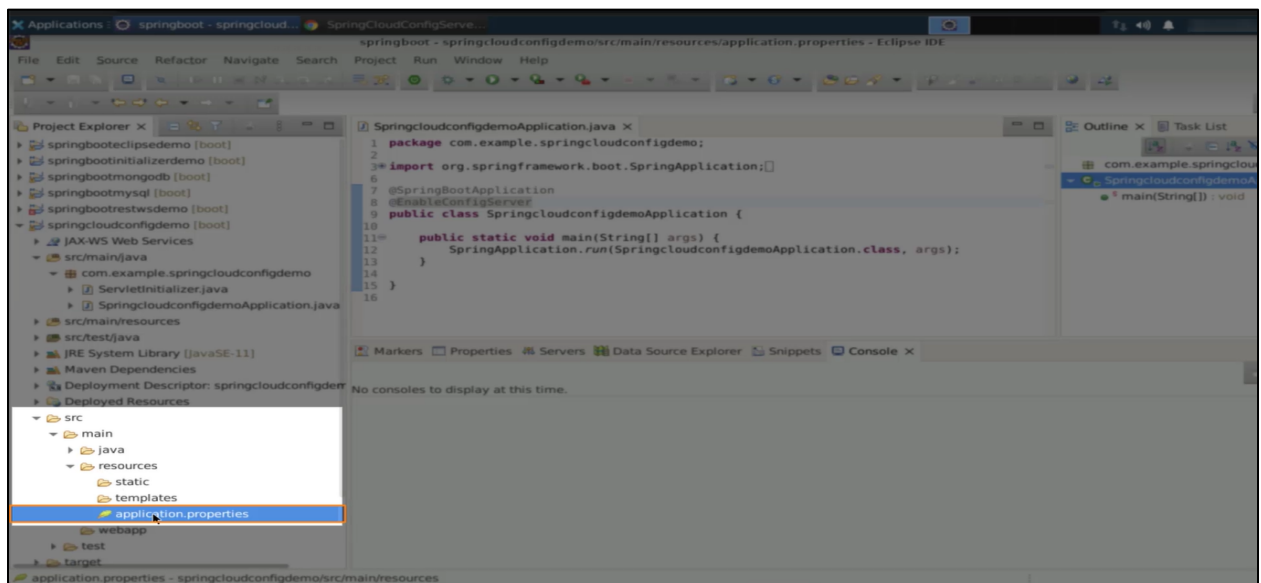


Step 2: Configuring the Spring Cloud in Eclipse IDE

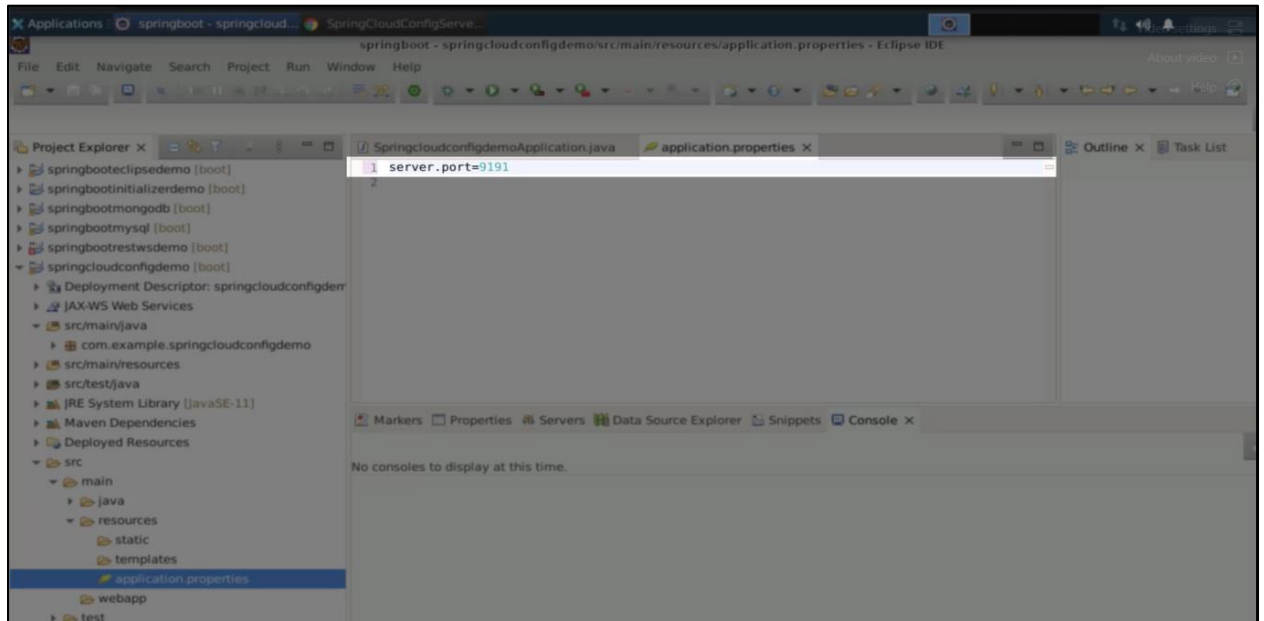
2.1 In `SpringcloudconfigdemoApplication.java`, add one of the annotations `@EnableConfigServer` as this will enable the cloud configuration server



2.2 Now, open `application.properties` file under `src > main > resources`

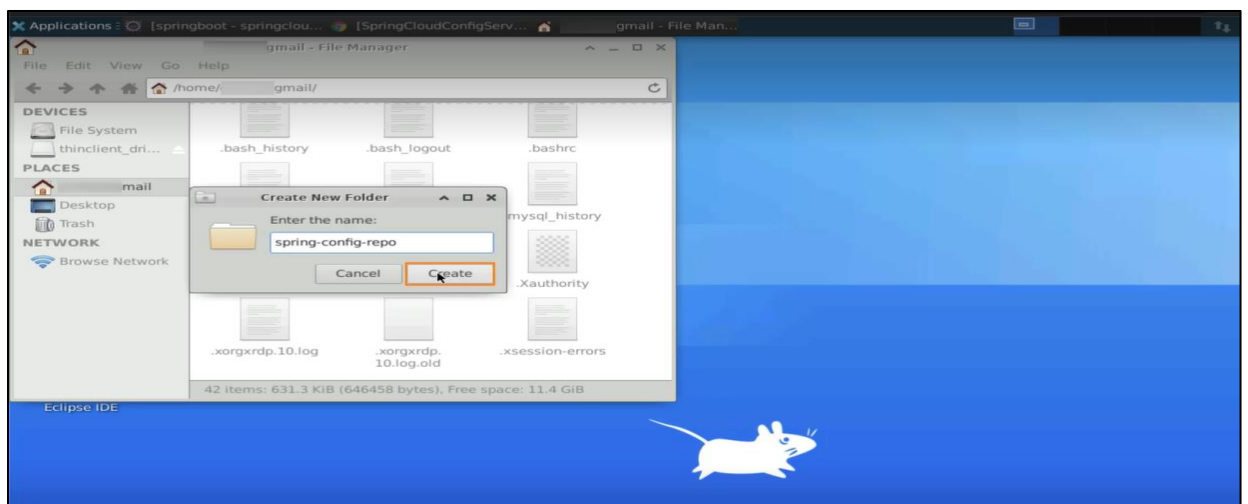


2.3 In **application.properties**, configure the server port as **9191**. Here, also configure the **Git URI**

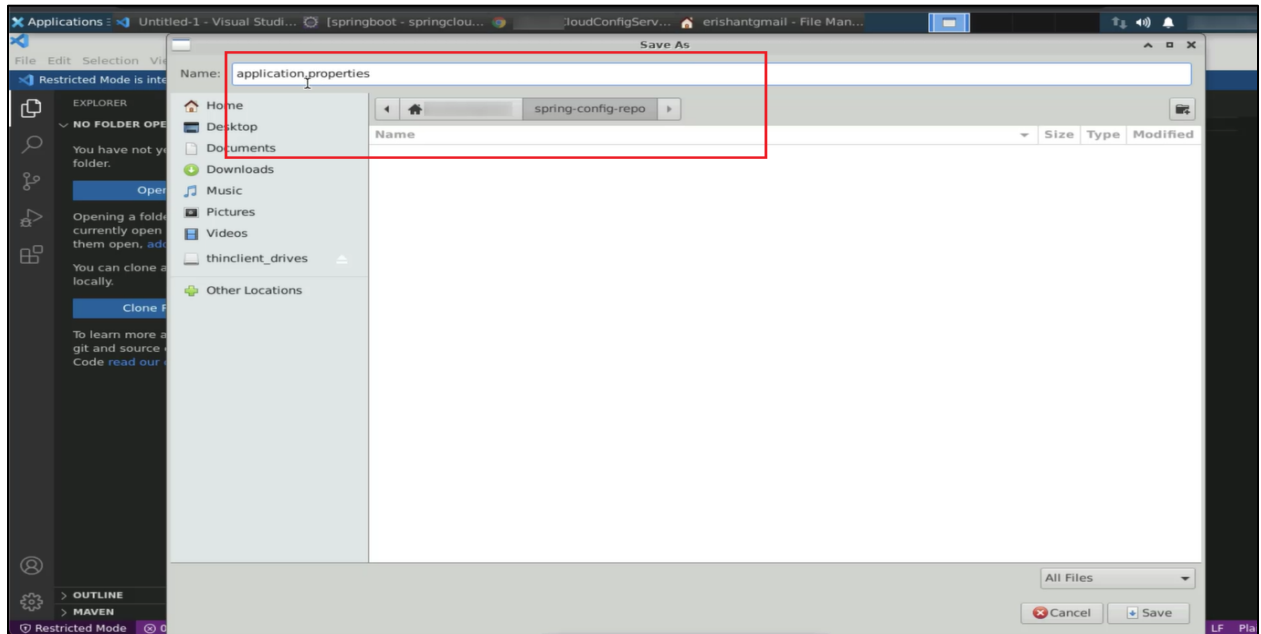


Step 3: Configuring the Git repository

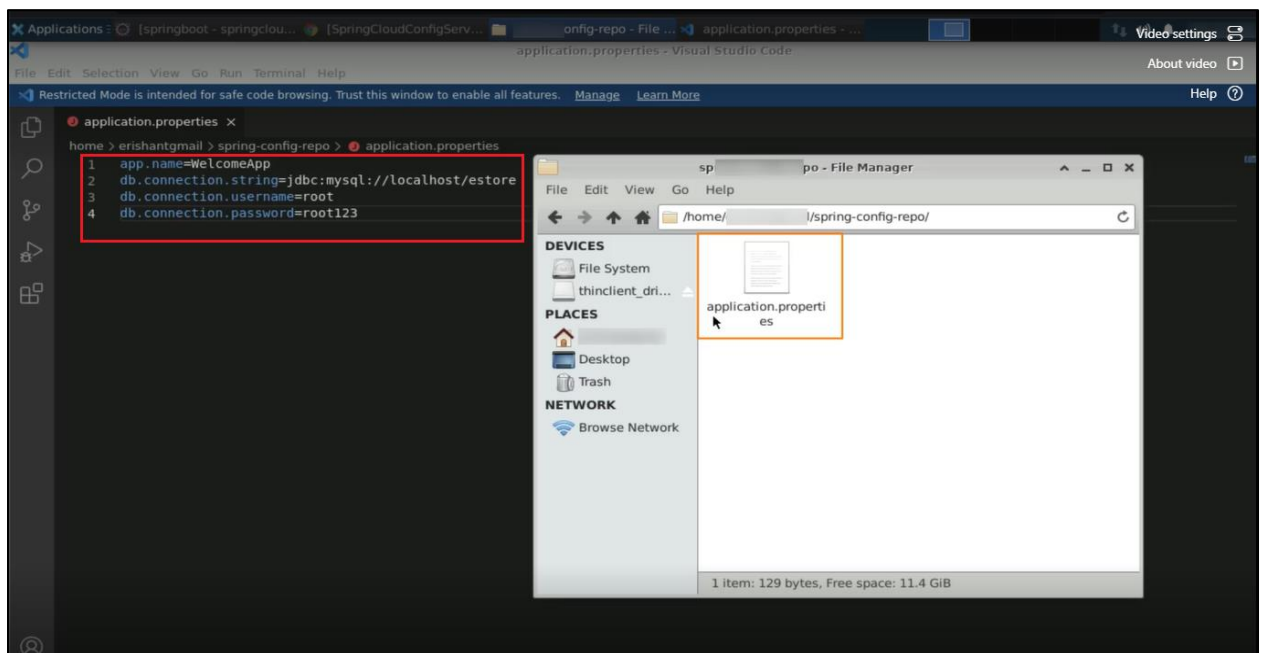
3.1 To configure the **Git URI**, provide a local location. For that, go to the system directory structure and create a folder named **spring-config-repo**



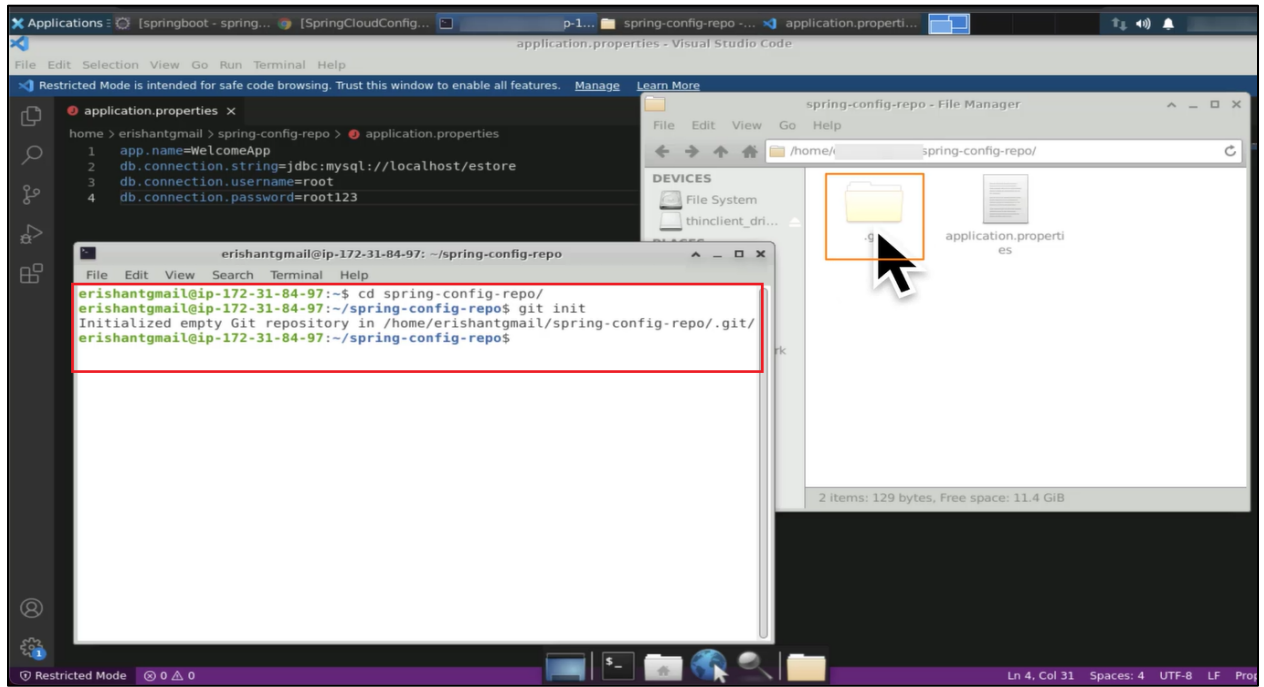
3.2 Open the **spring-config-repo** folder with **Visual Studio Code**. In this, create a new file named **application.properties** that specifies app properties



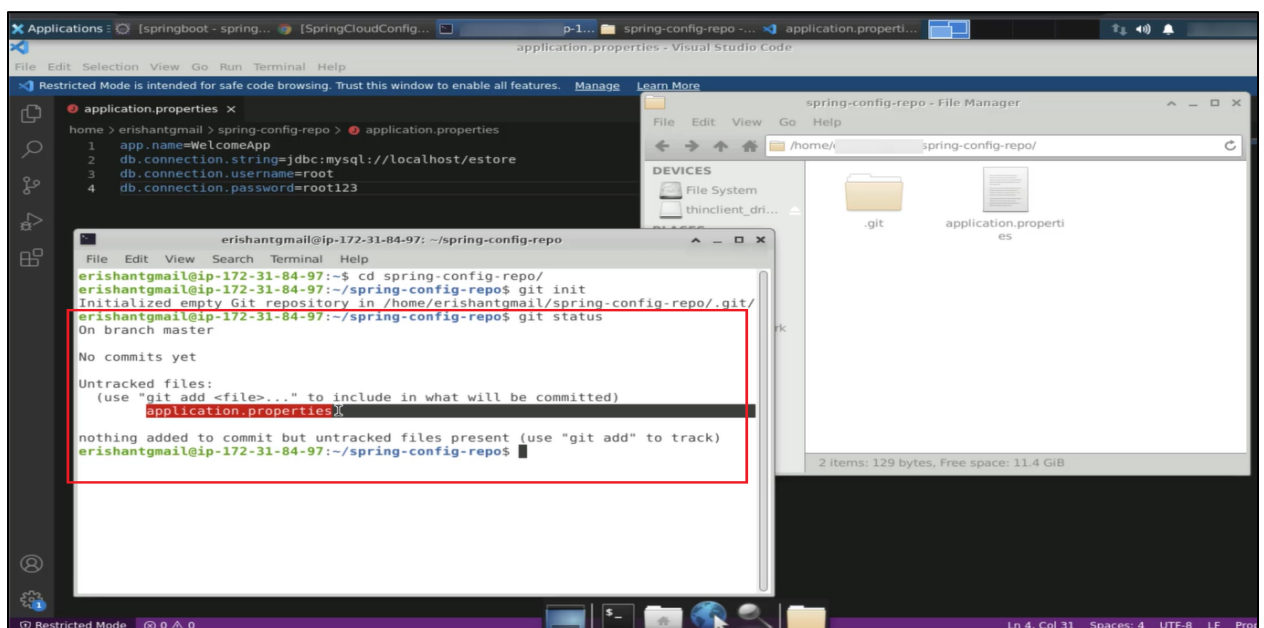
3.3 In **VS Code**, write the highlighted app properties in the **application.properties** file



3.4 Now, open the terminal to initialize Git. Write the command **git init** to initiate Git. After initiating Git, a local Git repository will be created in the **spring-config-repo**



3.5 Check the status of the Git repository with the **git status** command. You'll see no files have been added to the local Git repository yet.



3.6 Now, use the **git add.** command to add the files and check the status again with the **git status** command. You can now see the file **application.properties**, which can be committed to the **git** repository.

```

erishantgmail@ip-172-31-84-97: ~/spring-config-repo
File Edit View Search Terminal Help
erishantgmail@ip-172-31-84-97:~$ cd spring-config-repo/
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git init
Initialized empty Git repository in /home/erishantgmail/spring-config-repo/.git/
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        application.properties

nothing added to commit but untracked files present (use "git add" to track)
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git add .
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   application.properties
erishantgmail@ip-172-31-84-97:~/spring-config-repo$
  
```

3.7 Use the **git commit -m "application properties initial commit"** command to commit changes to the Git repository. Now, you can check the status using the **git status** command.

```

erishantgmail@ip-172-31-84-97: ~/spring-config-repo
File Edit View Search Terminal Help
  (use "git rm --cached <file>..." to unstage)
        new file:   application.properties

erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git commit -m "application p
roperties initial commit"
[master (root-commit) 7ed0d03] application properties initial commit
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:

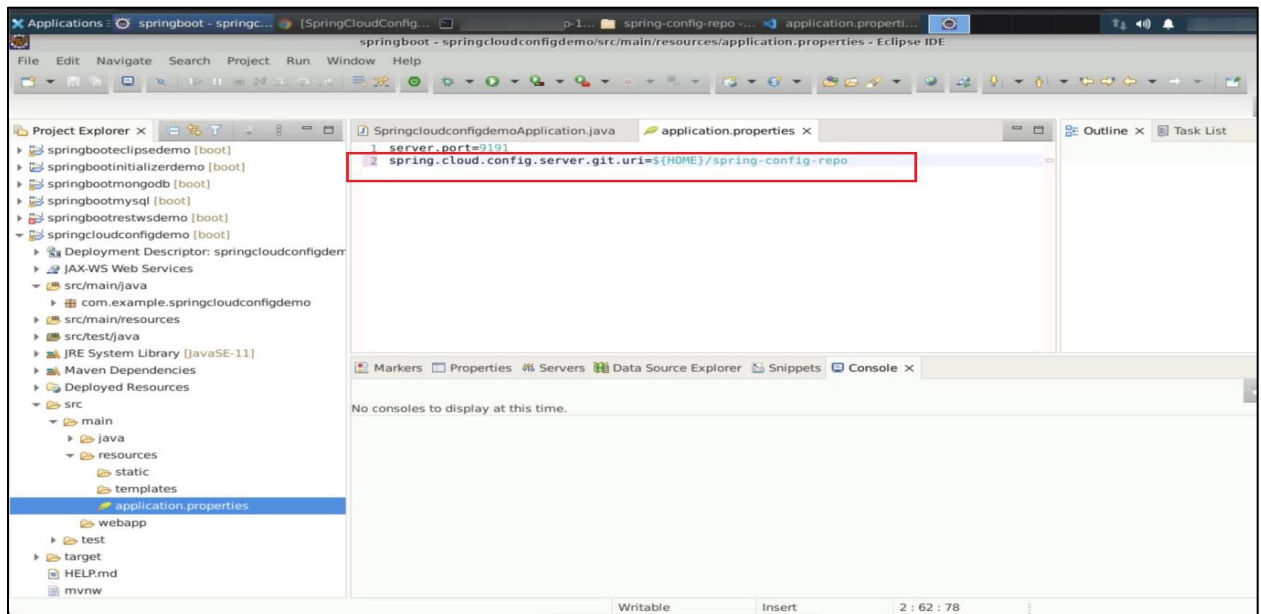
    git commit --amend --reset-author

1 file changed, 4 insertions(+)
create mode 100644 application.properties
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git status
On branch master
nothing to commit, working tree clean
erishantgmail@ip-172-31-84-97:~/spring-config-repo$
  
```

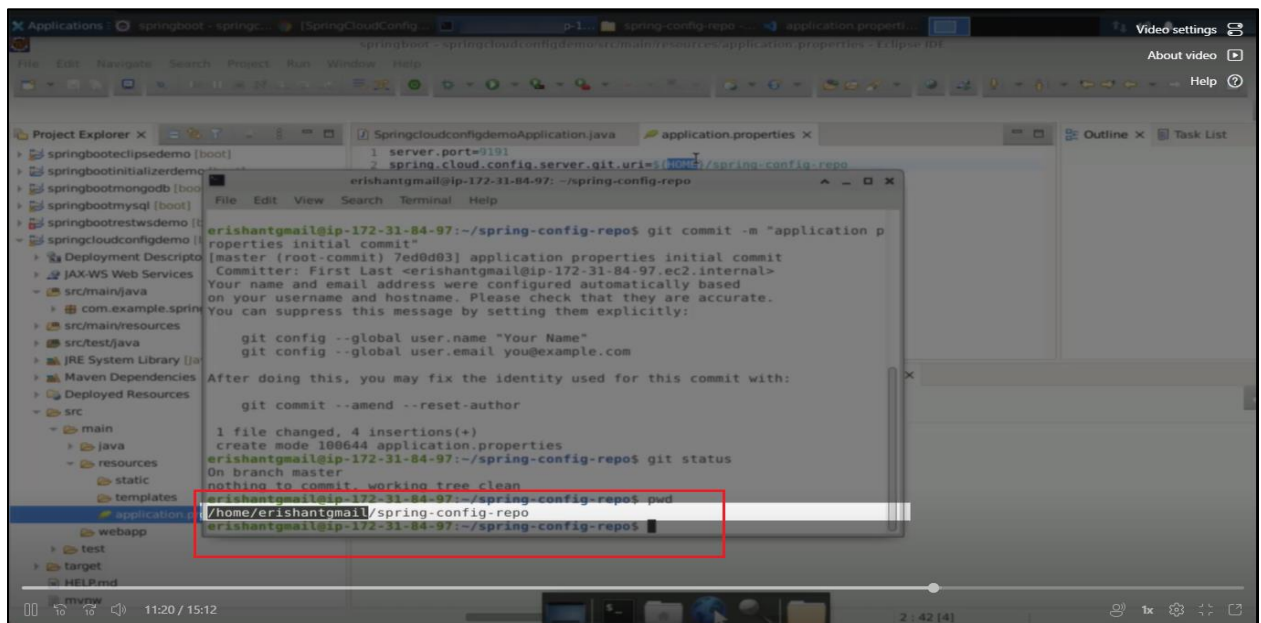
Now the Git repository has been set up for the Spring Cloud Configuration demo.

Step 4: Setting up the Git repository path for the Spring Cloud Configuration

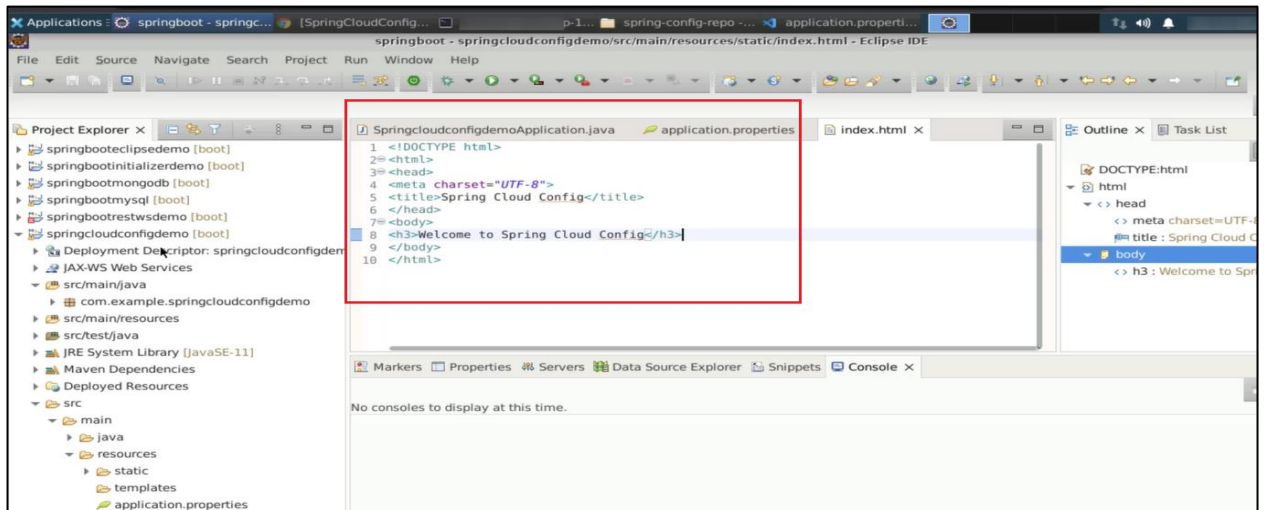
- 4.1 To set the path of the local Git repository to the **Spring Cloud Config** server, go to the **application.properties** in **Eclipse IDE** and write **spring.cloud.config.server.git.url=\${HOME}/spring.config.repo** property



- 4.2 To check the working directory, write the **pwd** command in the terminal



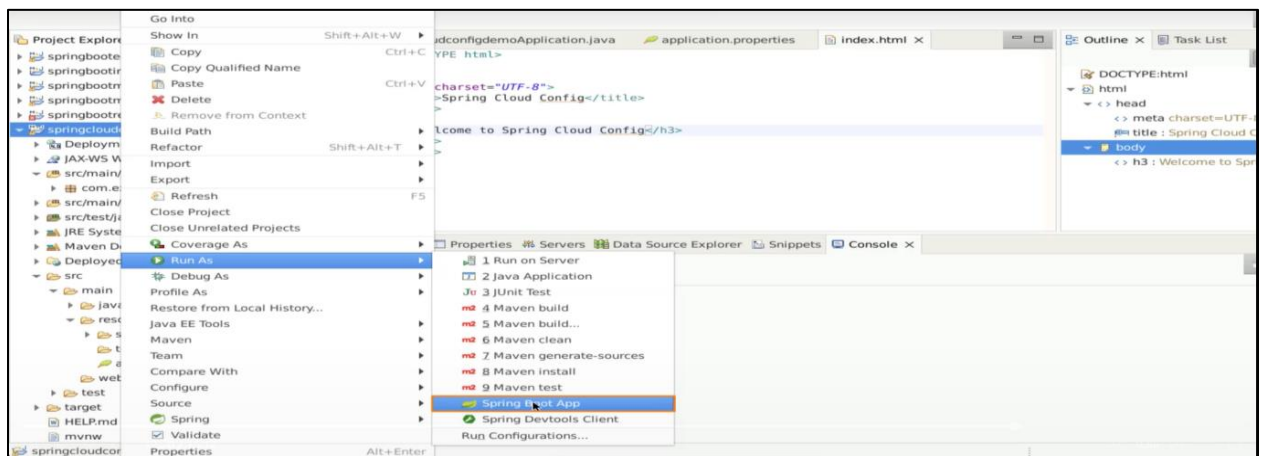
4.3 Before deploying the app, you need to make a welcome page. For this, go to the **static** folder under **resources** and **right-click** on it. Select **New > HTML File**. Create a file named **index.html** and add the highlighted code to that HTML file



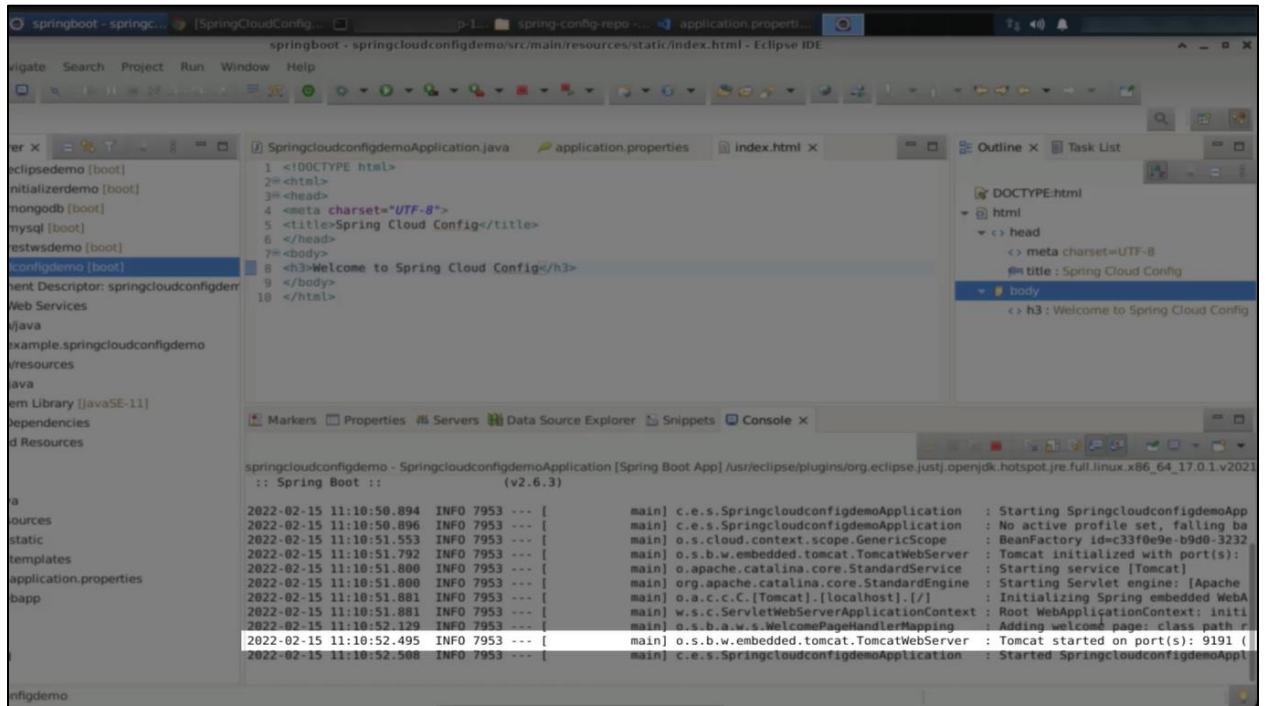
Note: The **index.html** file will be created under the **webapp** folder. Move this under the **static** folder

Step 5: Deploying the Spring Cloud Configuration project

5.1 Run this Spring Cloud Config demo as a **Spring Boot App**. For this, **right-click** on the **springcloudconfigdemo** project and select **Run As > Spring Boot App**



5.2 Notice that the **Tomcat** server starts on server port **9191**



```

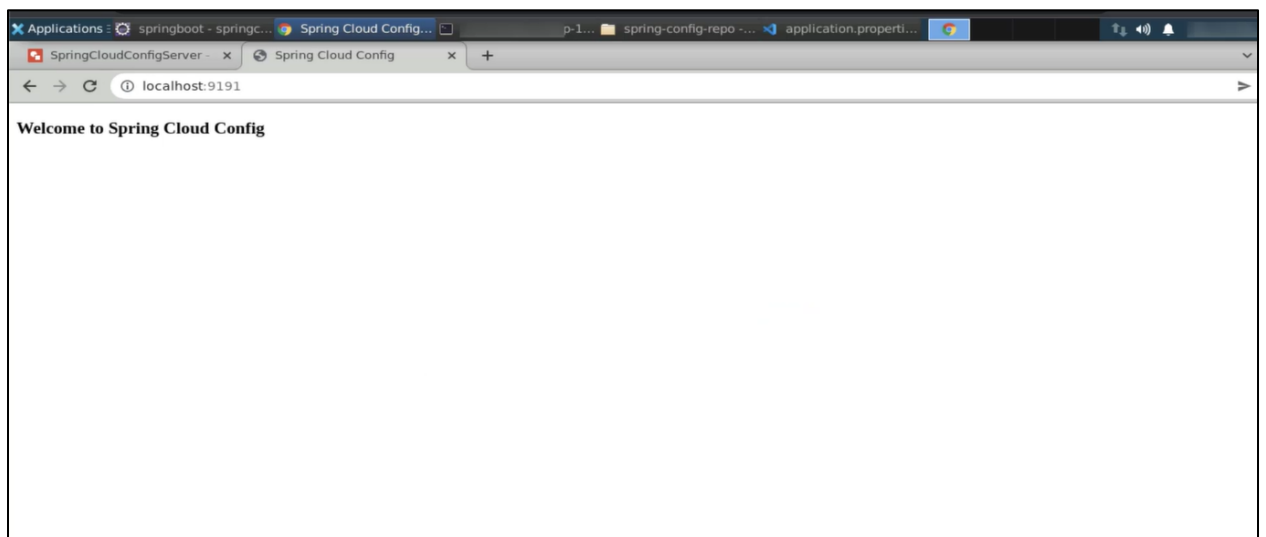
springboot - springc... [SpringCloudConfig... p-1... spring-config-repo... application.properti...
springboot - springcloudconfigdemo/src/main/resources/static/index.html - Eclipse IDE

File Edit Search Project Run Window Help
springcloudconfigdemoApplication.java application.properties index.html x Outline x Task List
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Spring Cloud Config</title>
6 </head>
7 <body>
8 <h3>Welcome to Spring Cloud Config</h3>
9 </body>
10 </html>

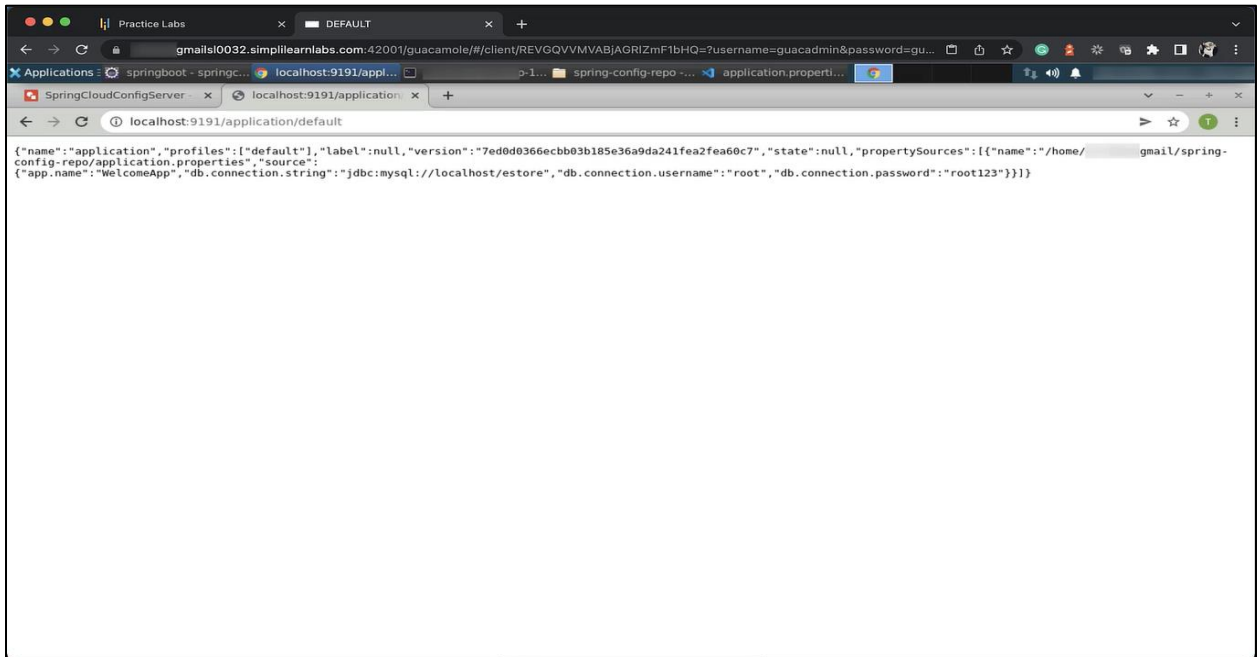
Markers Properties Servers Data Source Explorer Snippets Console x
springcloudconfigdemo - SpringcloudconfigdemoApplication [Spring Boot App] /usr/eclipse/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64.17.0.1.v2021
:: Spring Boot :: (v2.6.3)
2022-02-15 11:10:50.894 INFO 7953 --- [main] c.e.s.SpringcloudconfigdemoApplication : Starting SpringcloudconfigdemoApp
2022-02-15 11:10:50.896 INFO 7953 --- [main] c.e.s.SpringcloudconfigdemoApplication : No active profile set, falling ba
2022-02-15 11:10:51.553 INFO 7953 --- [main] o.s.cloud.context.scope.GenericScope : BeanFactory id=c33f0e9e-b9d0-3232
2022-02-15 11:10:51.792 INFO 7953 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s):
2022-02-15 11:10:51.800 INFO 7953 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-02-15 11:10:51.800 INFO 7953 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
2022-02-15 11:10:51.881 INFO 7953 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebA
2022-02-15 11:10:51.881 INFO 7953 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initi
2022-02-15 11:10:52.129 INFO 7953 --- [main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path r
2022-02-15 11:10:52.495 INFO 7953 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 9191 (
2022-02-15 11:10:52.508 INFO 7953 --- [main] c.e.s.SpringcloudconfigdemoApplication : Started SpringcloudconfigdemoAppl

```

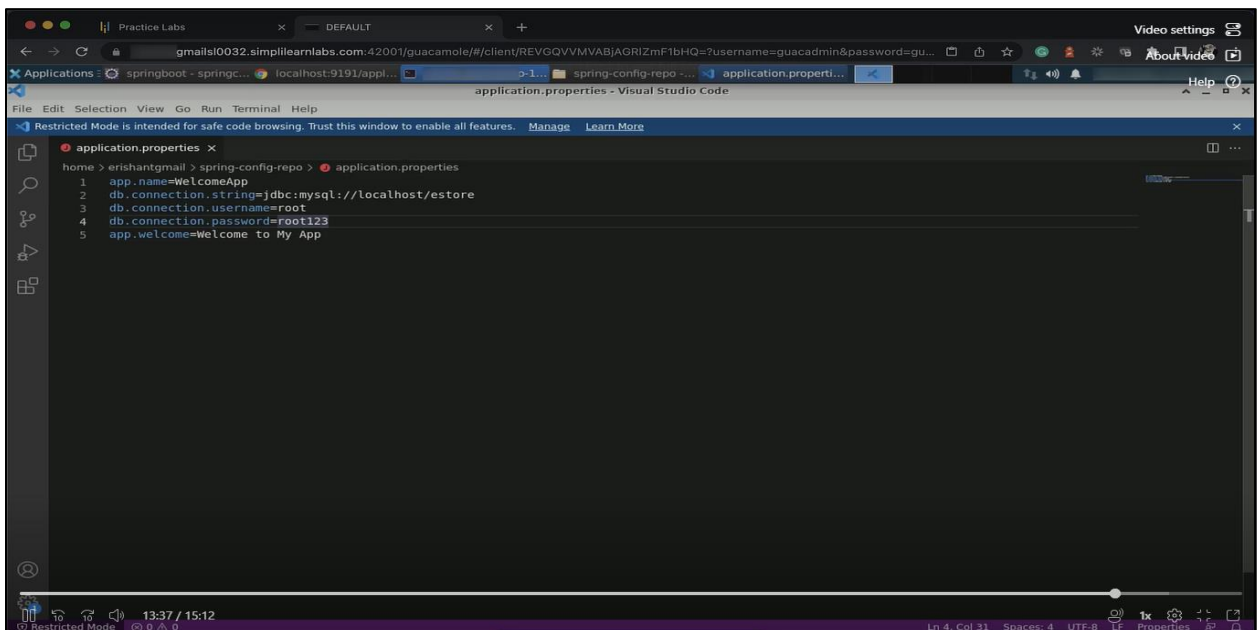
5.3 Now go to the browser and type **localhost:9191** to execute the web application



5.4 To check the properties of the application, type **localhost:9191/application/default** and click **Enter**



5.5 With **Spring Cloud Config**, any changes can be made to the application without deploying it again. Let's add another property, **app.welcome=Welcome to My App**, to the **application.properties** in Git repository



5.6 To check the modification status of your Git repository, use the **git status** command

```

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

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, 4 insertions(+)
create mode 100644 application.properties
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git status
On branch master
nothing to commit, working tree clean
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ pwd
/home/erishantgmail/spring-config-repo
erishantgmail@ip-172-31-84-97:~/spring-config-repo$ git status
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$
  
```

5.7 This will not reflect any changes on the configuration server until we commit the changes to the Git repository. For this, use the **git add** . command to add changes and use the **git commit -m "added a property"** command to commit the changes

```

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 "added a property"
[master 42407e8] added a property
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:

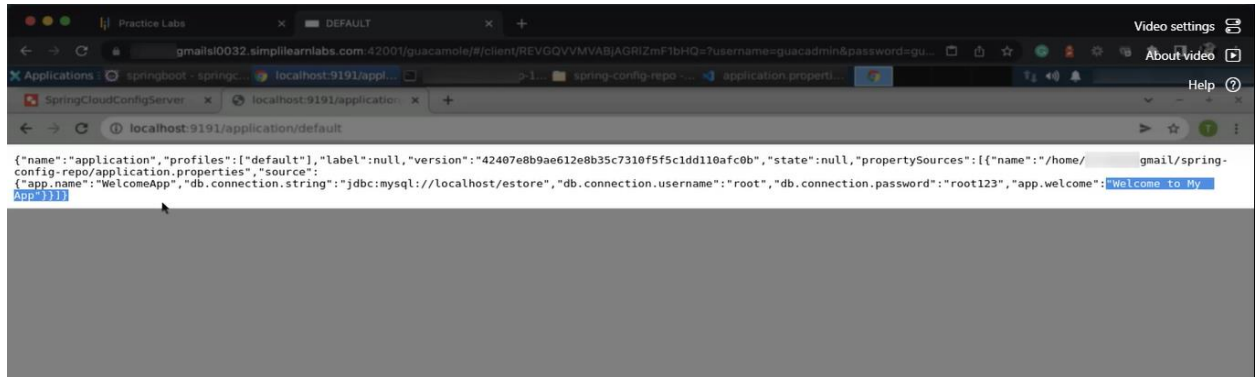
git config --global user.name "Your Name"
git config --global user.email you@example.com

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

git commit --amend --reset-author

1 file changed, 2 insertions(+), 1 deletion(-)
erishantgmail@ip-172-31-84-97:~/spring-config-repo$
  
```

5.8 Now, hit the same URL **localhost:9191/application/default**. Now, the changes can be seen in the web application without deploying the microservices.



With this, you have successfully created a Spring Cloud Configuration that acts as a provider to the microservices of various configurations on the server side in a distributed environment.