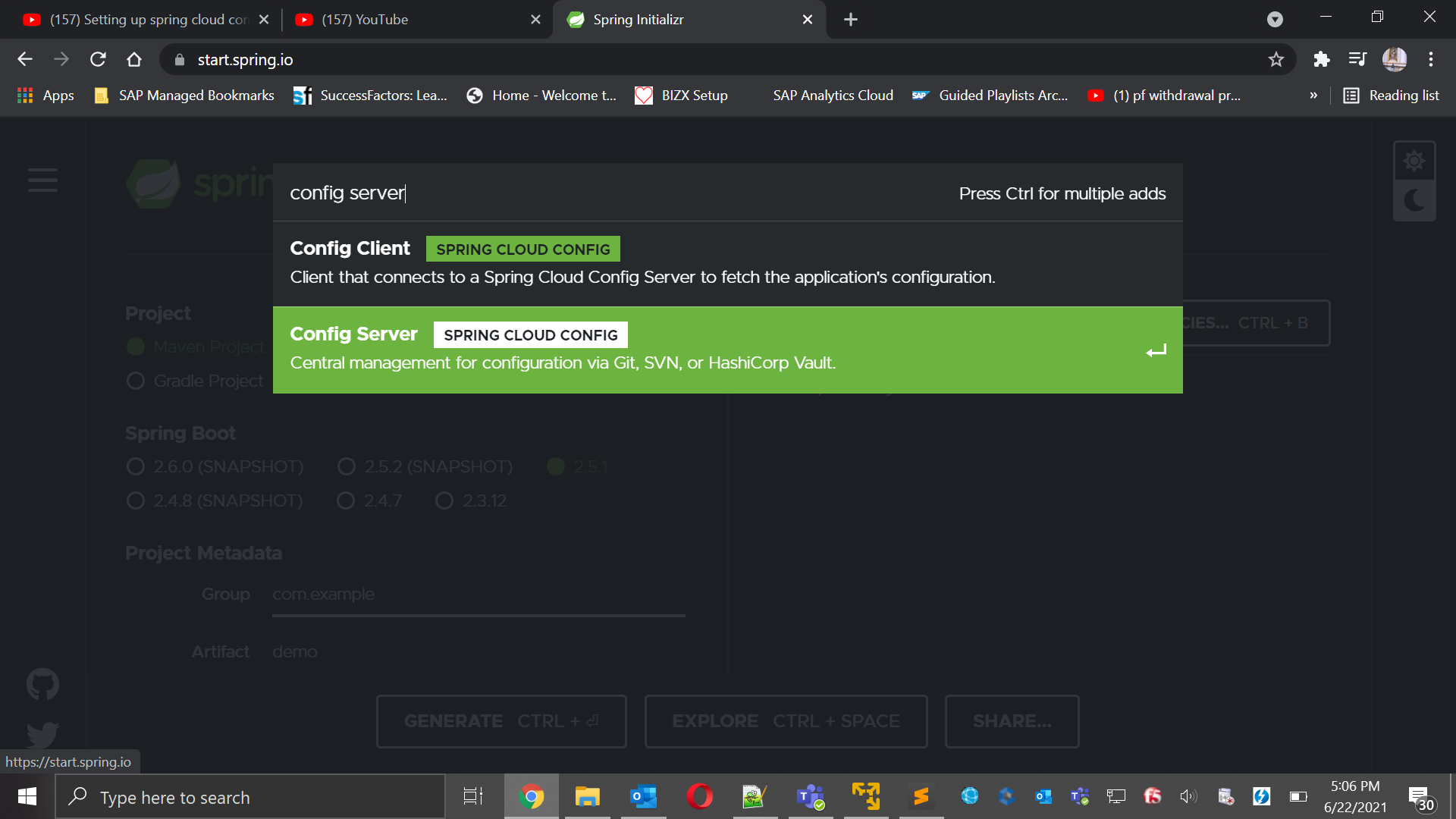
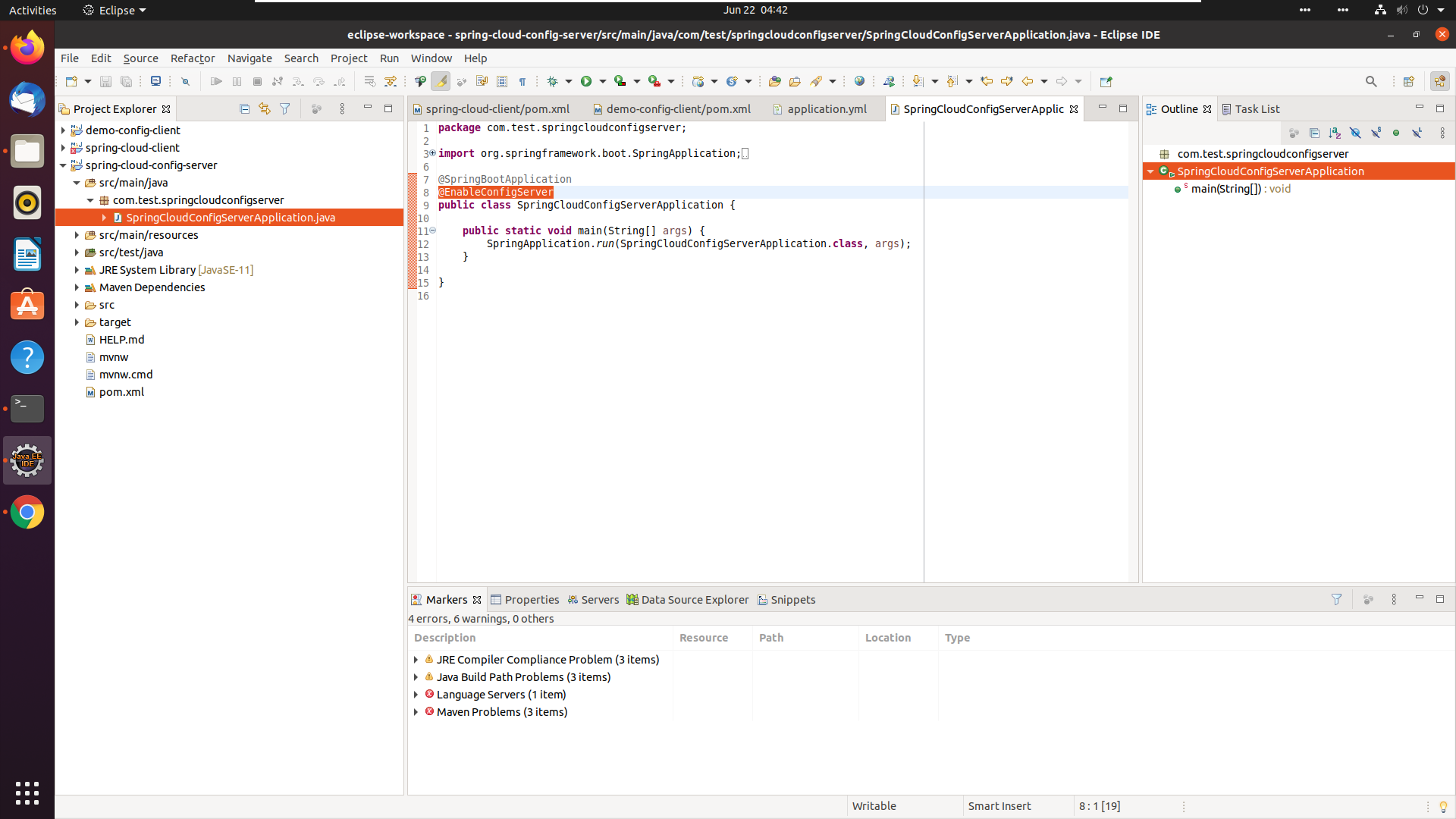
1. create a spring boot project with config server.



1. pom.xml looks something like this:



1. Annotate “@EnableConfigServer” in the main method.



1. Time to tell your config server the location of the config file. We are using git repo to store the config file. Also config server should up and run on port : 8888.

Config data start:

server:

port: 8888

spring:

cloud:

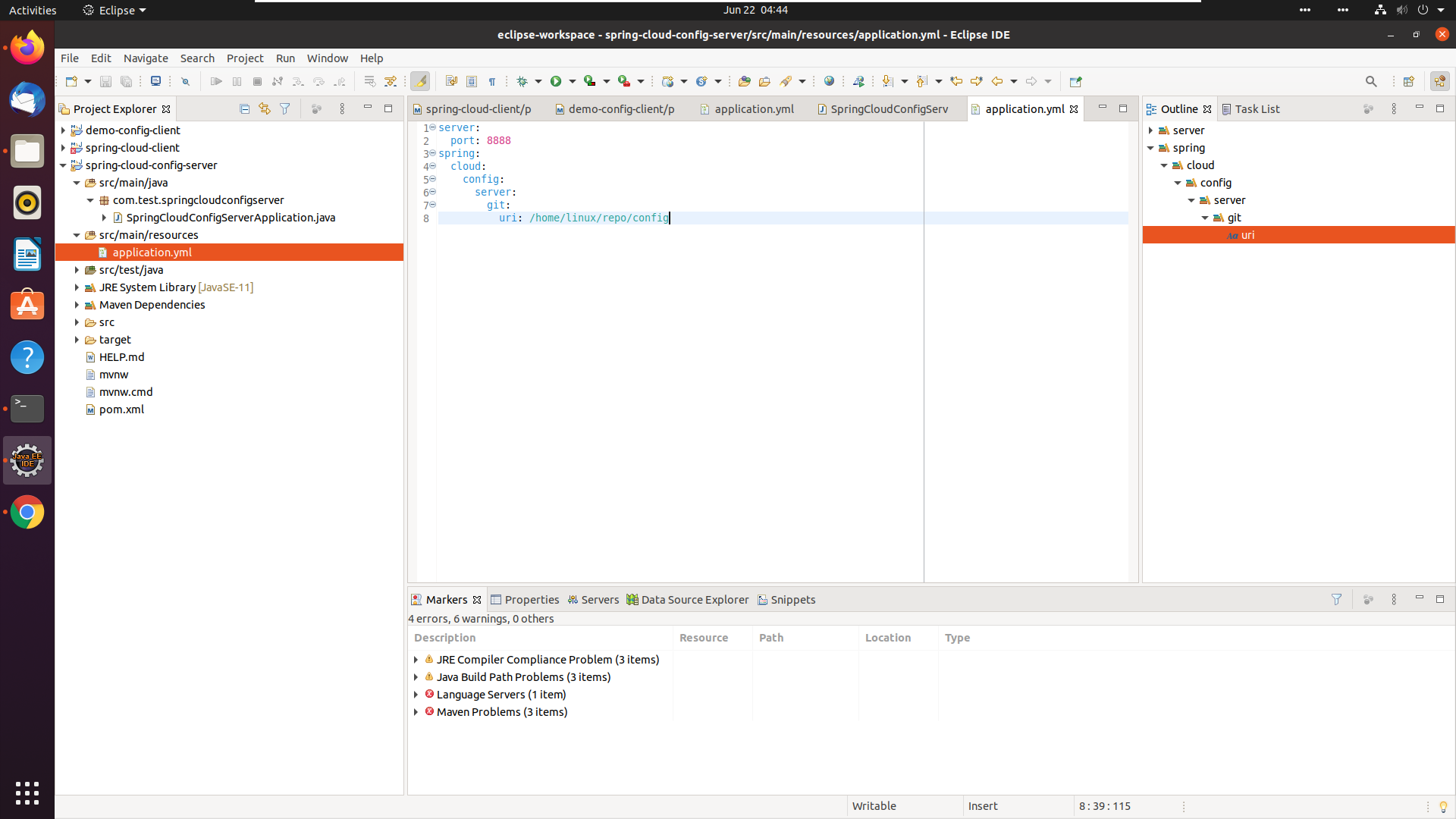
config:

server:

git:

uri: /home/linux/repo/config

Config data end.



1. How to config your git repo:

Git repo config start:

14 mkdir repo

15 cd repo/

16 mkdir config

17 vi application.yml

18 cd config/

19 pwd

20 vi application.yml

21 cat application.yml

22 ll

23 git init

24 ll

25 git status

26 git add example.property: "I AM IN THE DEFAULT"

27 spring.jpa.database: "POSTGRESQL"

28 spring.datasource.platform: "postgres"

29 spring.jpa.show-sql: "true"

30 spring.database.driverClassName: "org.postgresql.Driver"

31 spring.datasource.url: "jdbc:postgresql://database:5432/eagle\_eye\_local"

32 spring.datasource.username: "postgres"

33 spring.datasource.password: "{cipher}4788dfe1ccbe6485934aec2ffeddb06163ea3d616df5fd75be96aadd4df1da91"

34 spring.datasource.testWhileIdle: "true"

35 spring.datasource.validationQuery: "SELECT 1"

36 spring.jpa.properties.hibernate.dialect: "org.hibernate.dialect.PostgreSQLDialect"

37 redis.server: "redis"

38 redis.port: "6379"

39 ll

40 git status

41 git add application.yml

42 git status

43 git commit -m "Config file added version 1.0"

44 git status

45 git config --global user.email "you@example.com"

46 git config --global user.name "linux"

47 git commit -m "Config file added version 1.0"

48 git status

Git repo config end.

1. You are ready to run your app.

$mvn clean install

$mvn spring-boot:run

Open this URL in browser : <http://localhost:8888/application/default>

