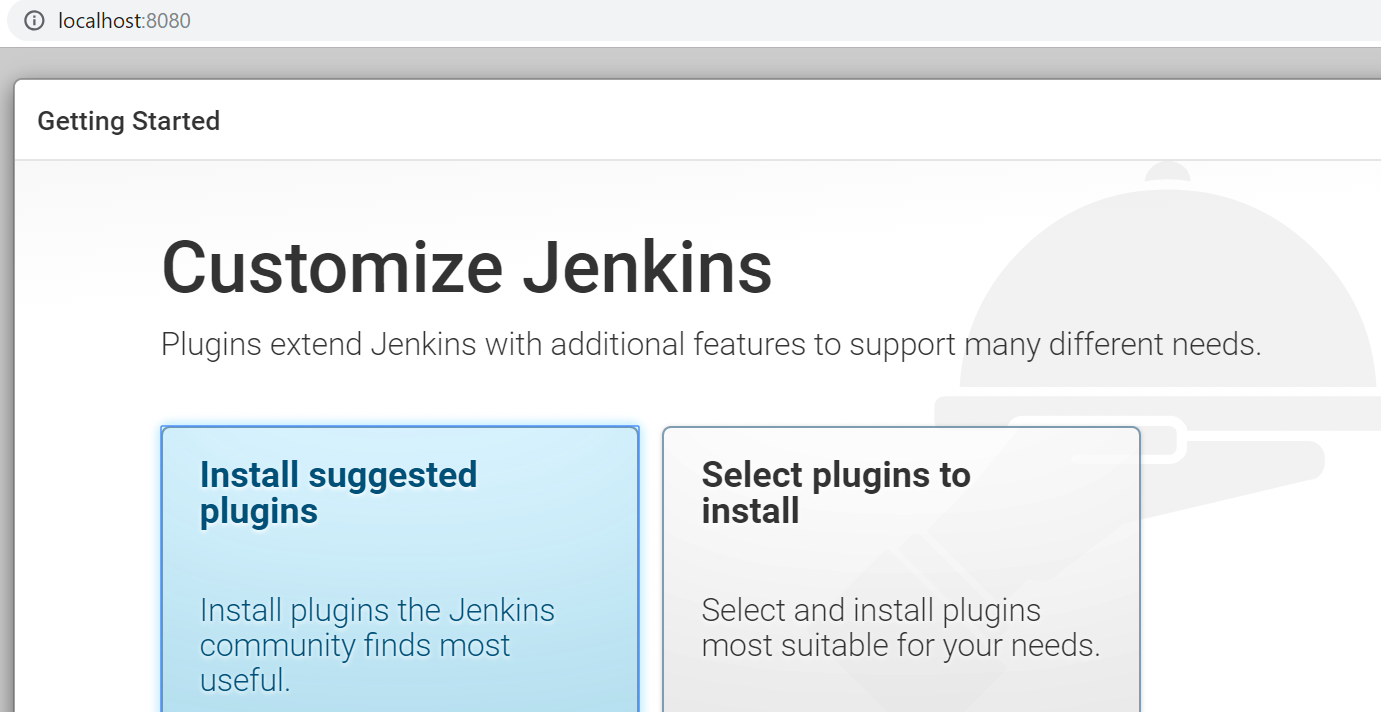
**What is CI/CD pipeline:**

Continuous integration is a coding philosophy and set of practices that drive development teams to implement small changes and check in code to version control repositories frequently. Because most modern applications require developing code in different platforms and tools, the team needs a mechanism to integrate and validate its changes.

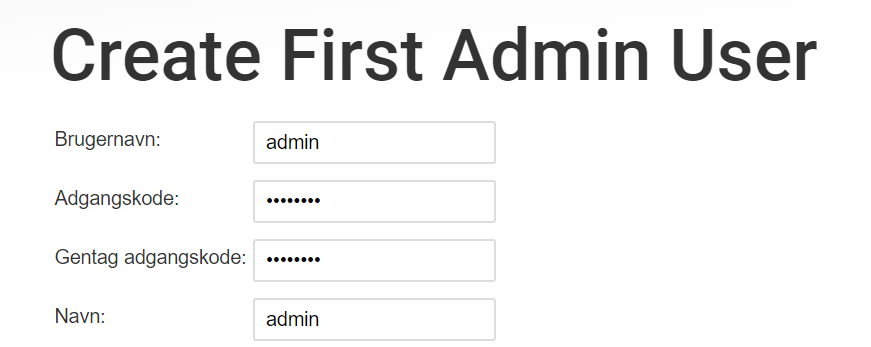
The technical goal of CI is to establish a consistent and automated way to build, package, and test applications. With consistency in the integration process in place, teams are more likely to commit code changes more frequently, which leads to better collaboration and software quality.

Continuous delivery picks up where continuous integration ends. CD automates the delivery of applications to selected infrastructure environments. Most teams work with multiple environments other than the production, such as development and testing environments, and CD ensures there is an automated way to push code changes to them.  CD automation then performs any necessary service calls to web servers, databases, and other services that may need to be restarted or follow other procedures when applications are deployed.

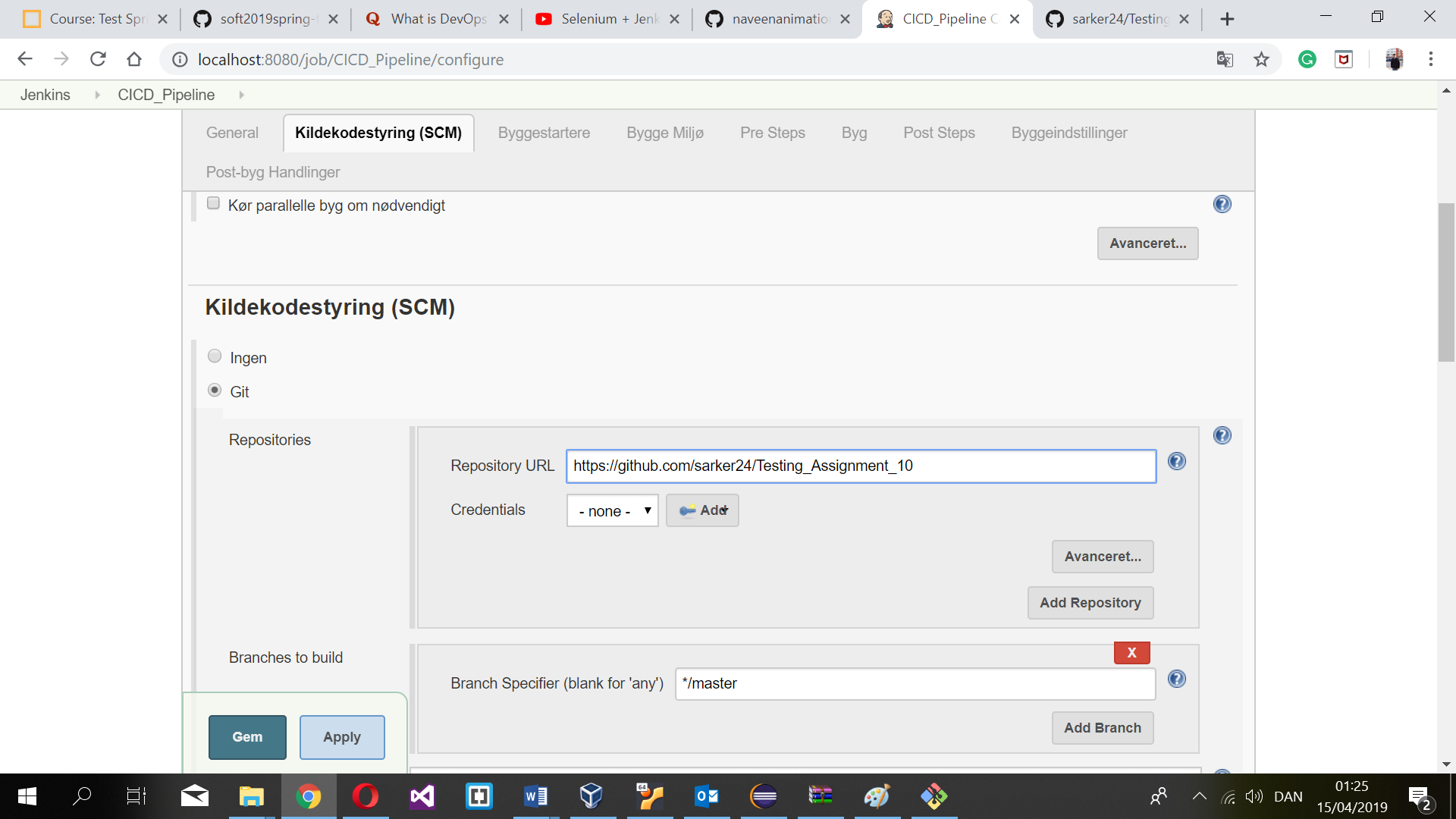
A Continuous Integration Pipeline is a powerful instrument that consists of a set of tools designed to **host**, **monitor**, **compile** and **test** code, or code changes. I will use Jenkins for **Continuous Integration Server.**



After installing Jenkins I will set up various plagins.



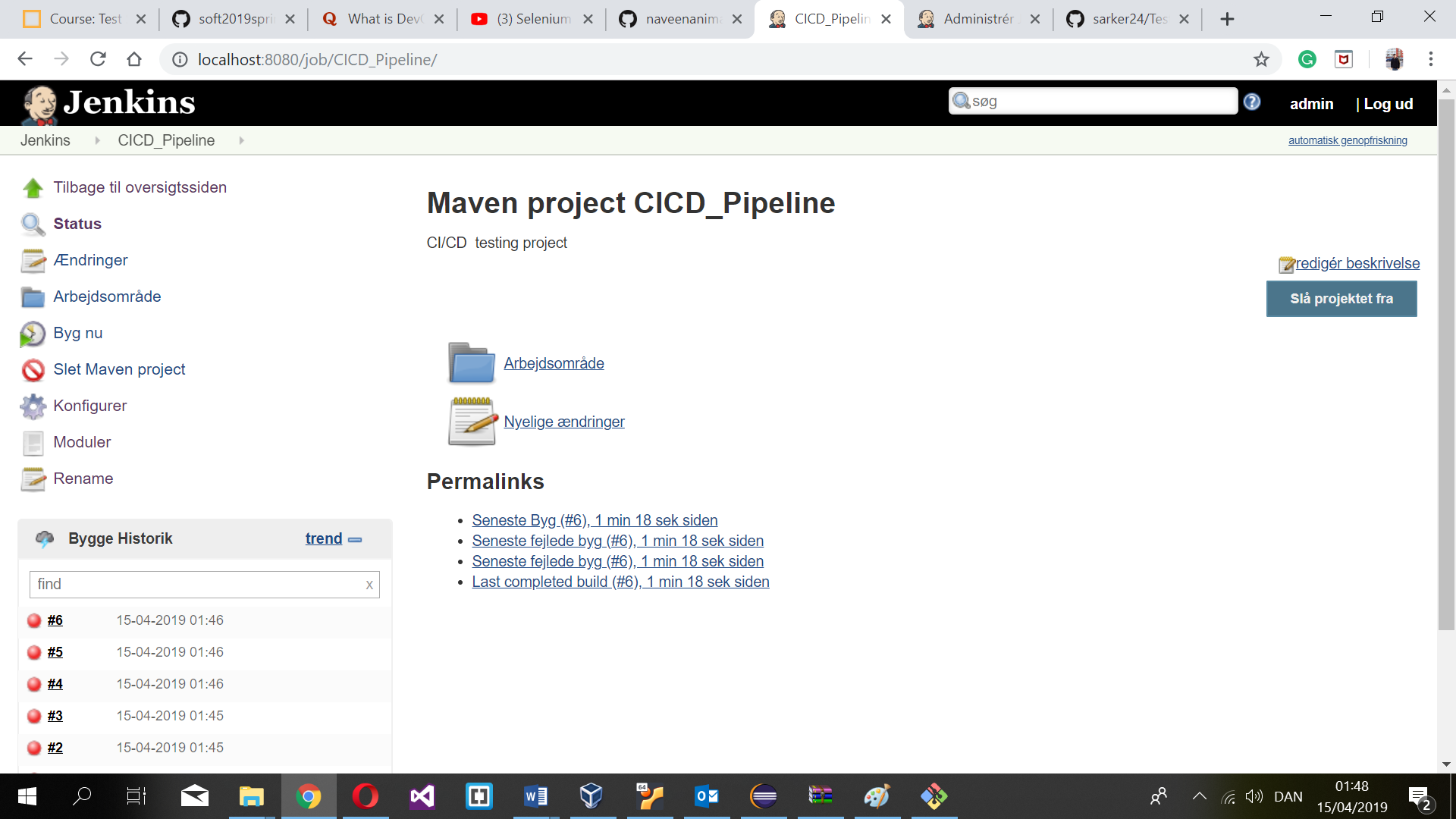
I will create admin user. The user id is admin and password is admin123. Now my Jenkins is ready for use.



For source code I am not use my local source code I have to use my GitHub repository address.



Now I have to created ci/cd pipeline.



If I will build the project now it will execute all selenium test cases.