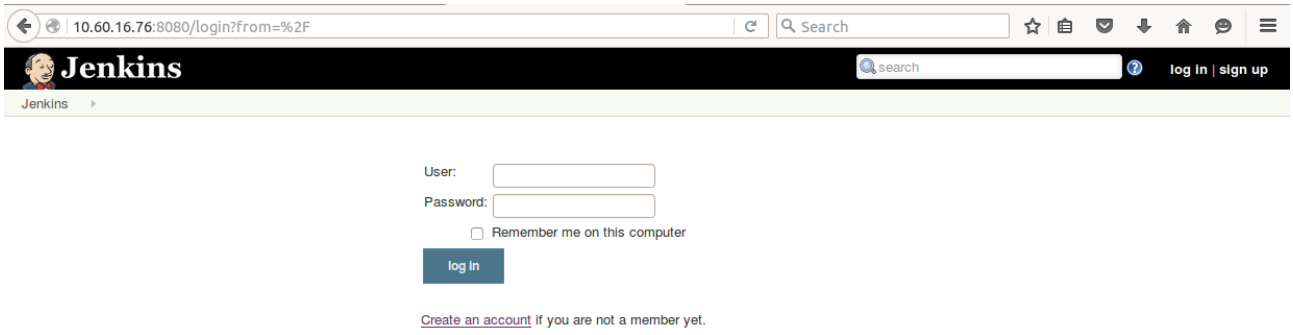
Jenkins configuration:

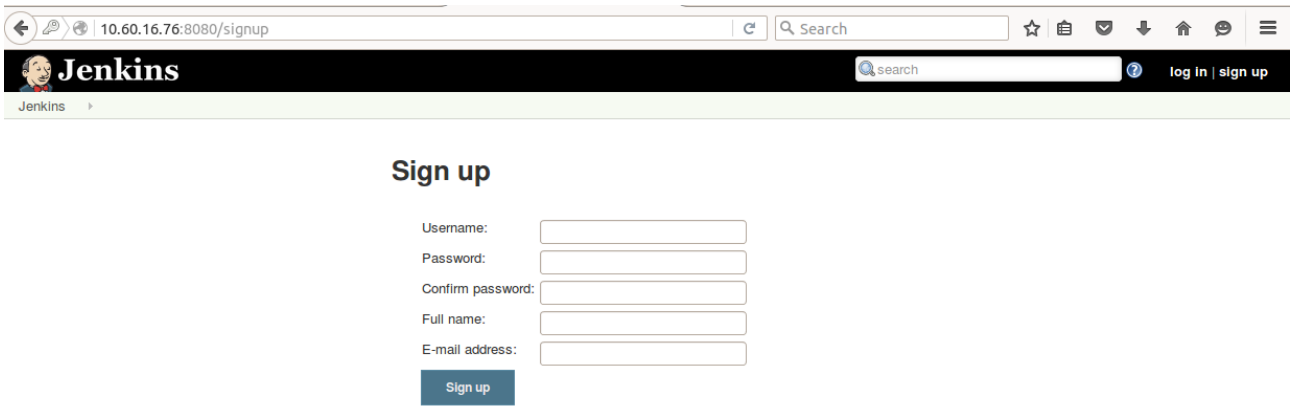
1. Signing in:
   - Go to Jenkins page http://10.60.16.76:8080/login?from=%2F
   - Use your username and password from Cybercom inside to log in or click on **create an account** and go to the next point



Pic 1 Signing in or creating new account.

2. Adding new users:

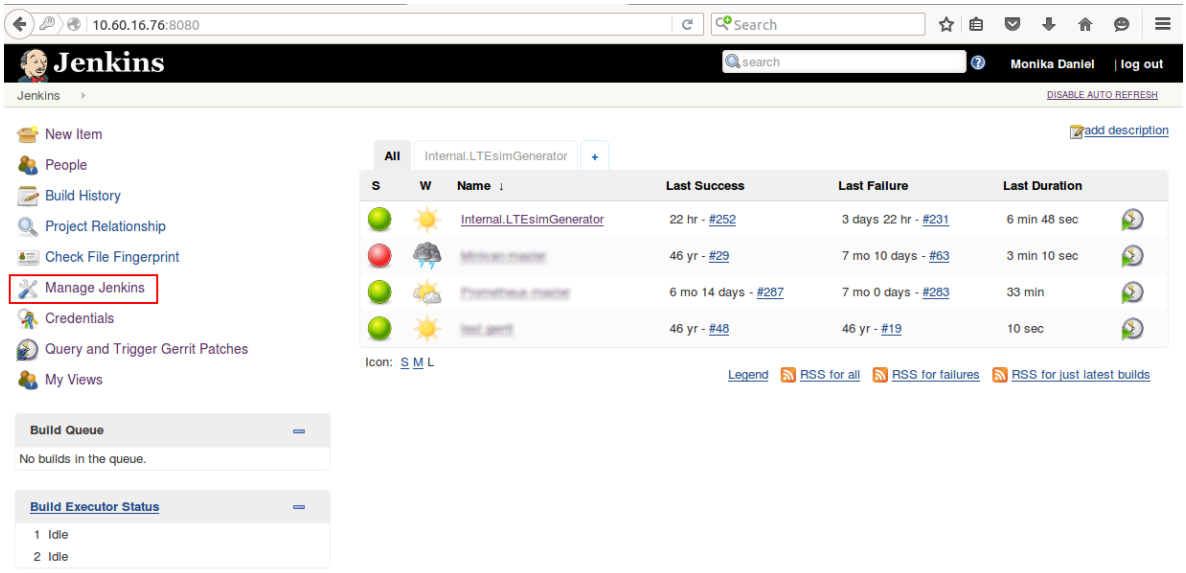
   To add new user you have to fill a form and then administrator has to give you an access to a project. Use data from Cybercom inside it will help you to keep regularity with users identifications between Gerrit and Jenkins.



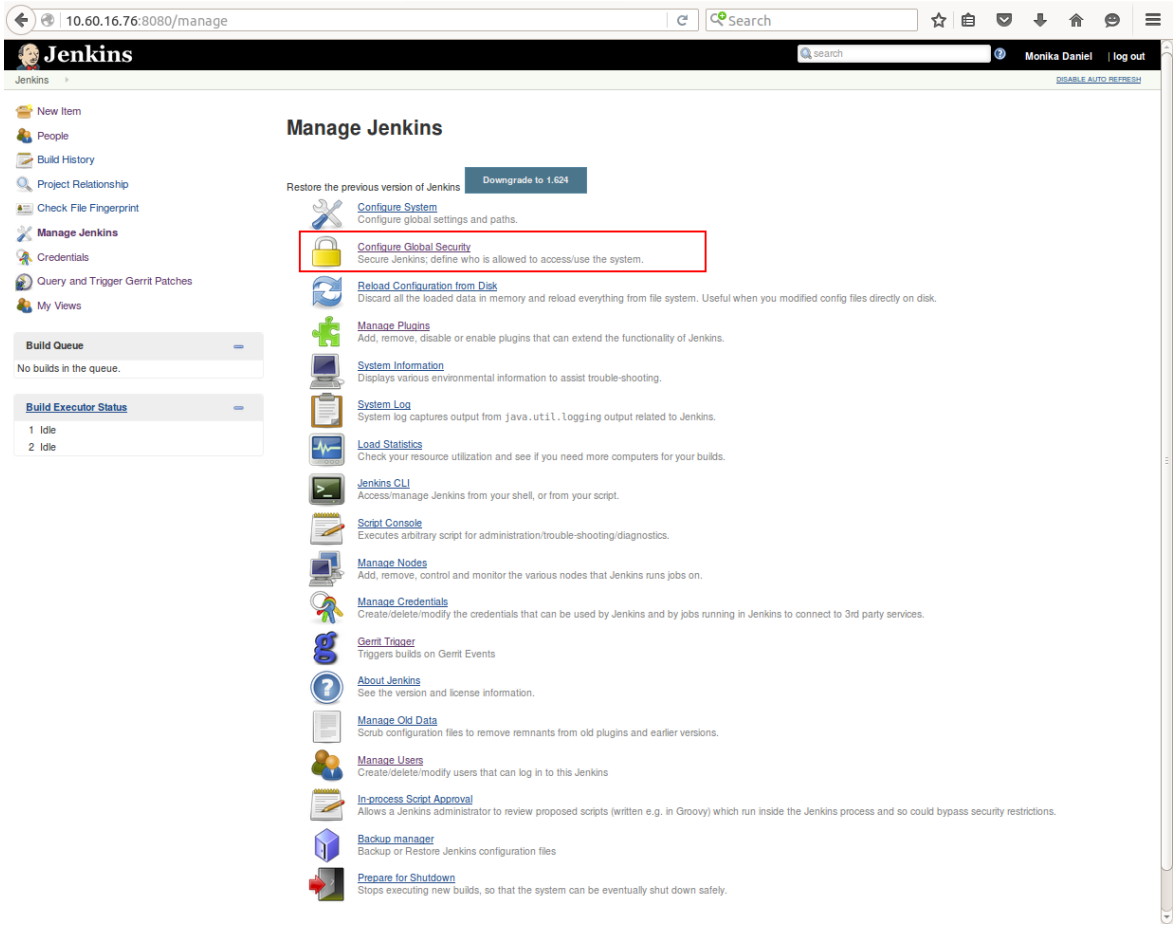
Pic 2 Creating new account.

Now administrator has to add some rights to your account in your board of projects. Log in as administrator and follow the steps:

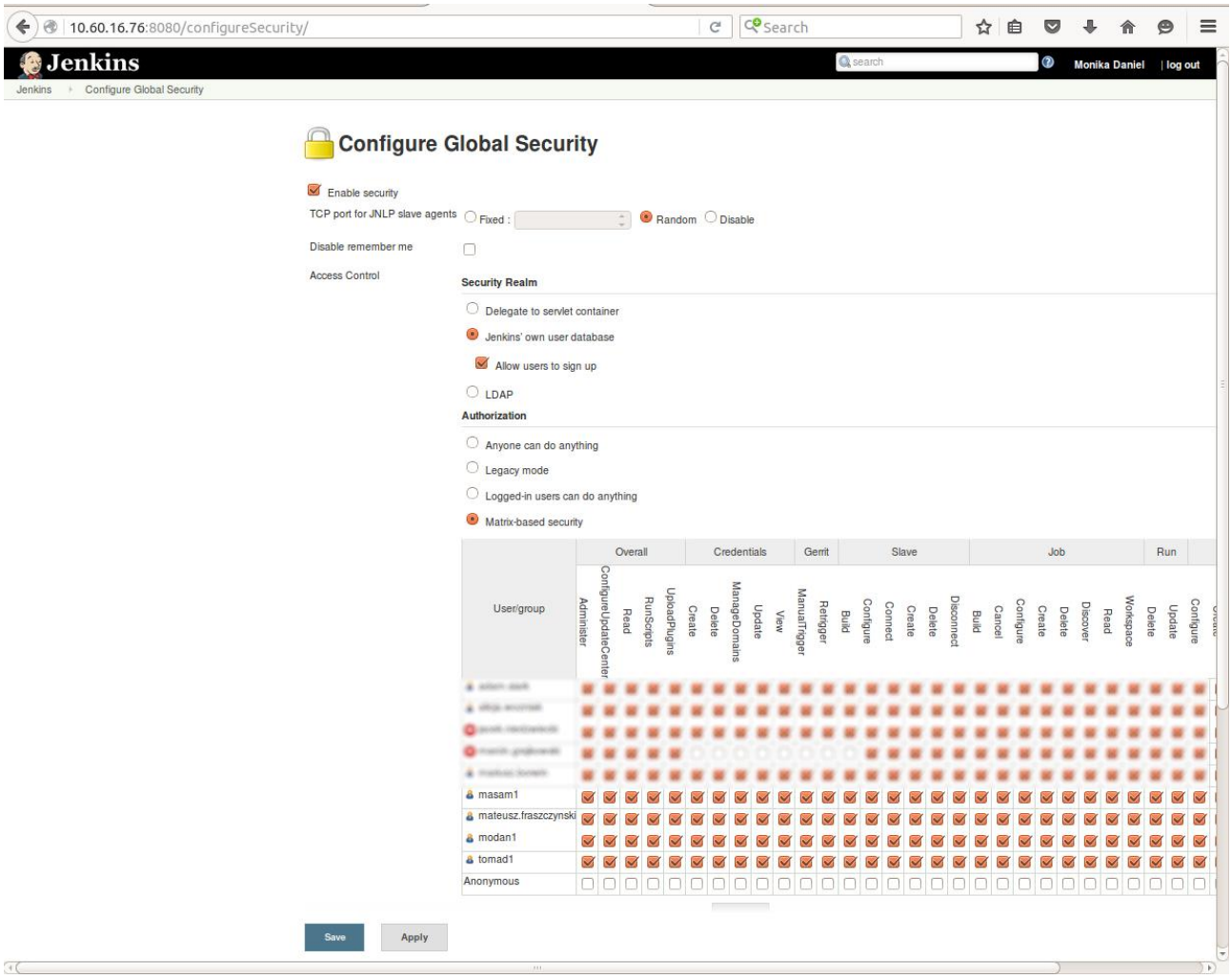
- Choose *Manage Jenkins -> Configure Global Security*



Pic 3 Main menu.



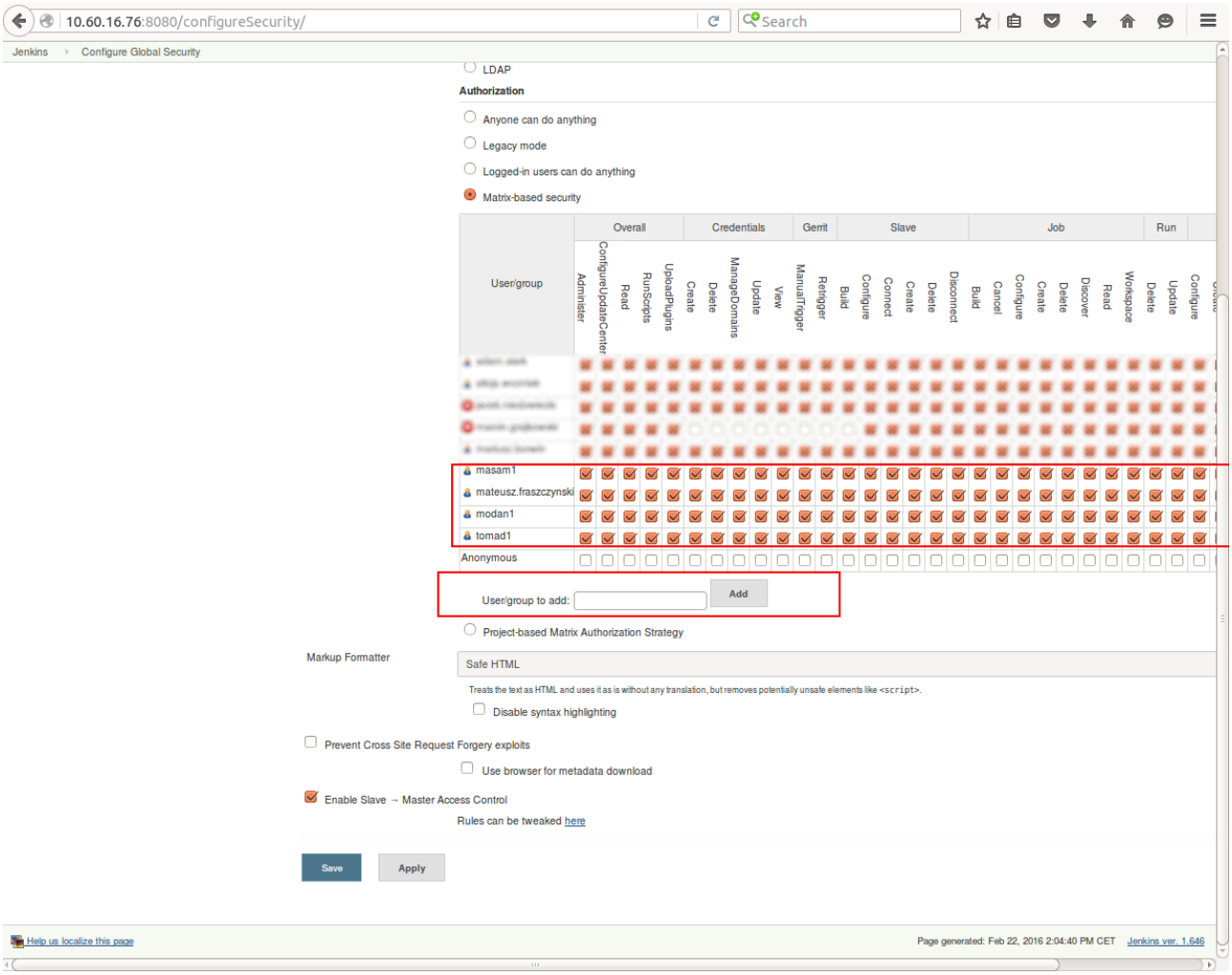
Pic 4 Manage Jenkins menu.



Pic 5 Global security menu.

- In **user/group to add label** write username and click **add button**

User will appear in the table of access. Now all you have to do is choose actions which user will be able to do. Then click **save button**.

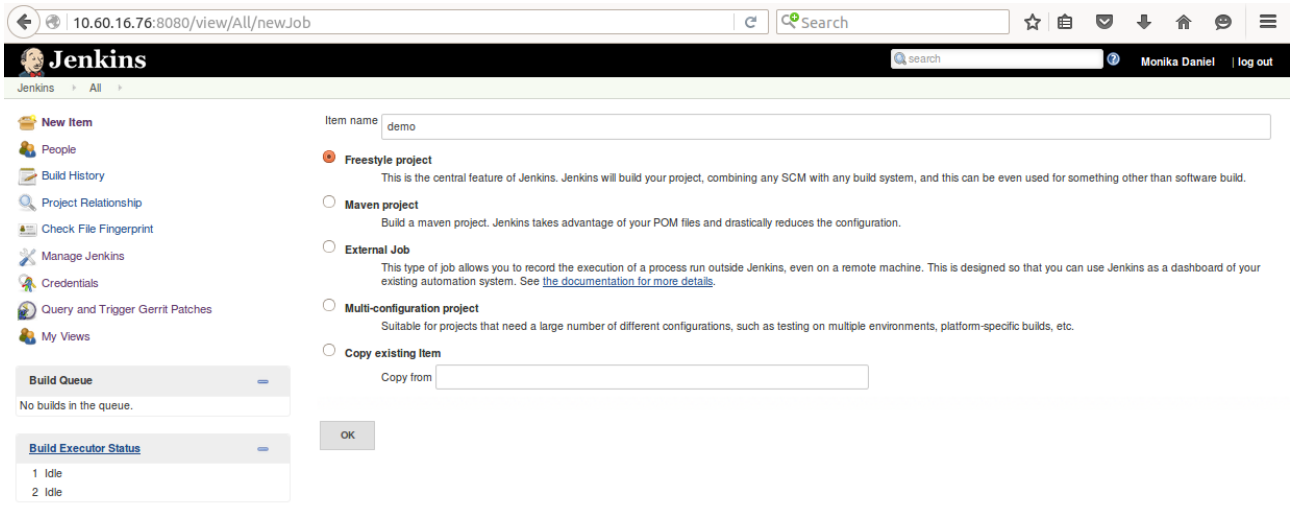


Pic 6 Adding user label and access table.

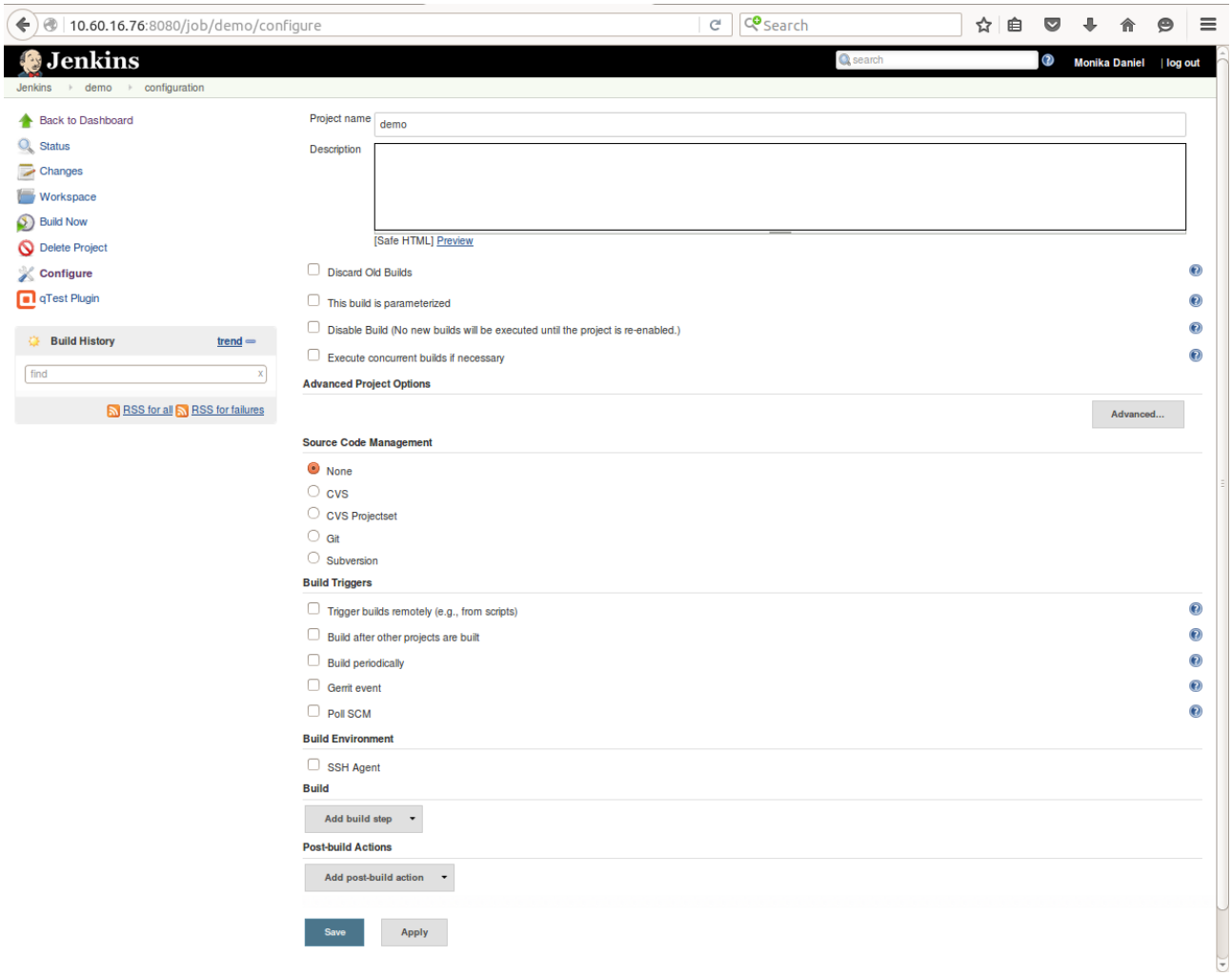
3. Creating new projects:

To create new project you have to sign in and follow the instructions:

- Choose in main menu *New Item -> Freestyle project* and click **OK button.**



Pic 7 Creating new project.

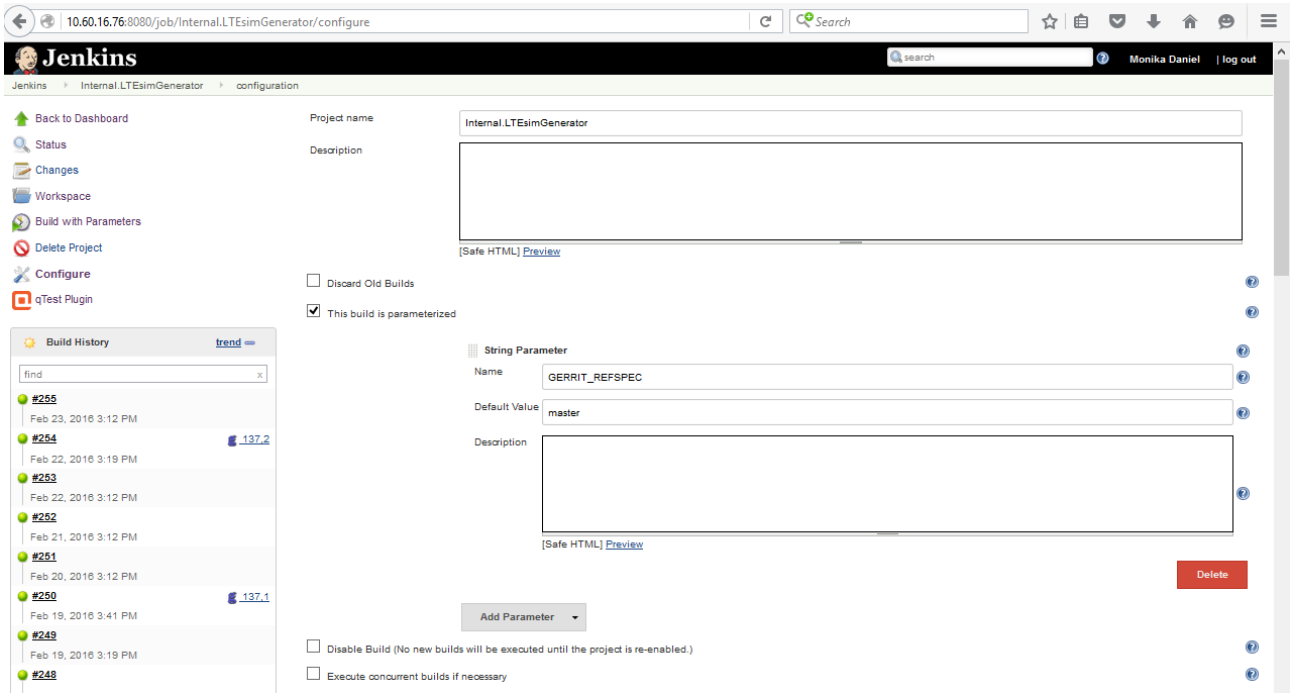


Pic 8 Project configuration.

4. Project configuration:

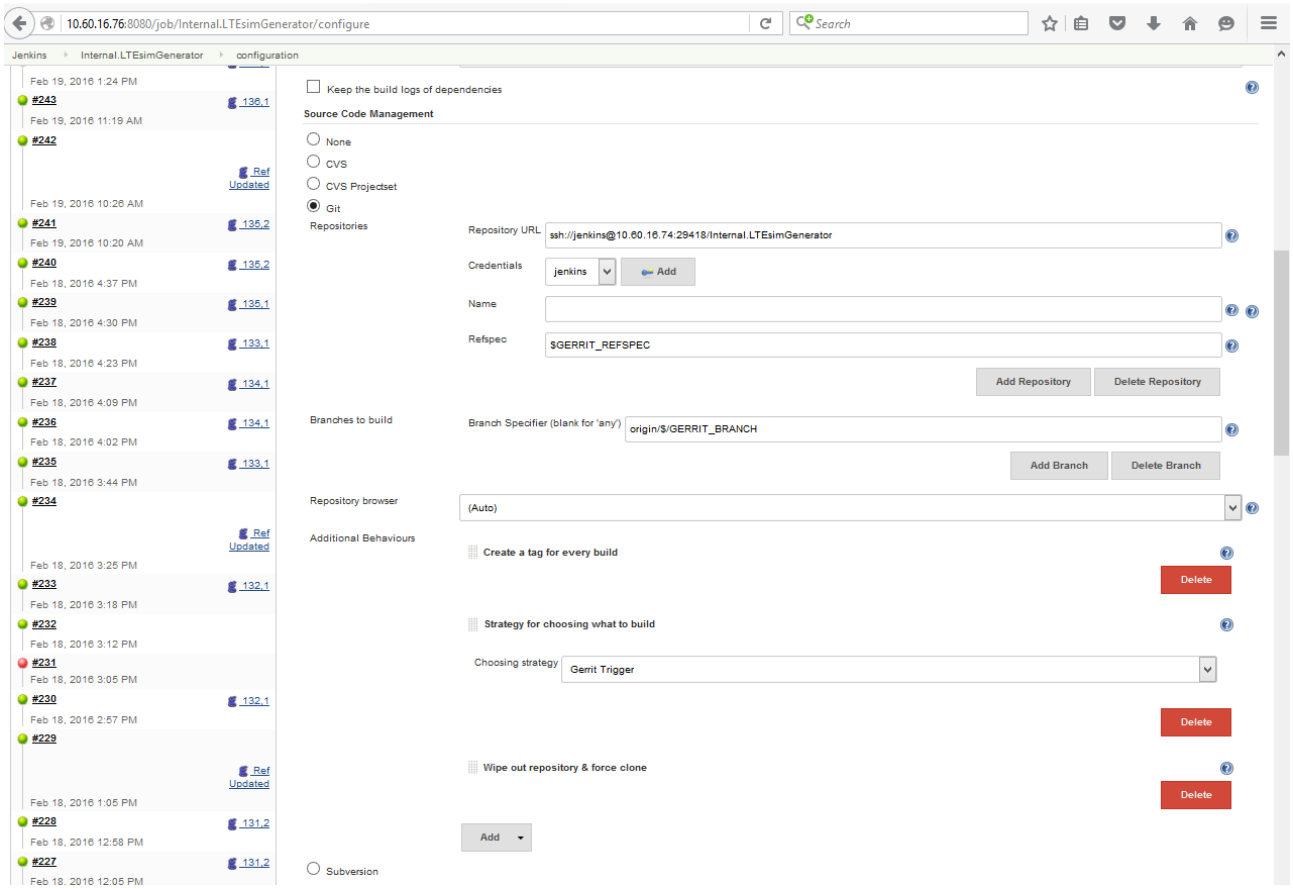
This step needs a lot of work and our attention to make Gerrit and Jenkins work with each other.

In first section we have to pick **this build is parameterized checkbox**, than you should click **add parameter button** and choose **String parameter** from list. Fill **Name label** with **GERRIT_REFSPEC** and **Default value label** with **master**.



Pic 9 Project configuration - parameterized build.

Next we need to configure **Source code management** section. Choose **Git** and fill like in the picture 10.

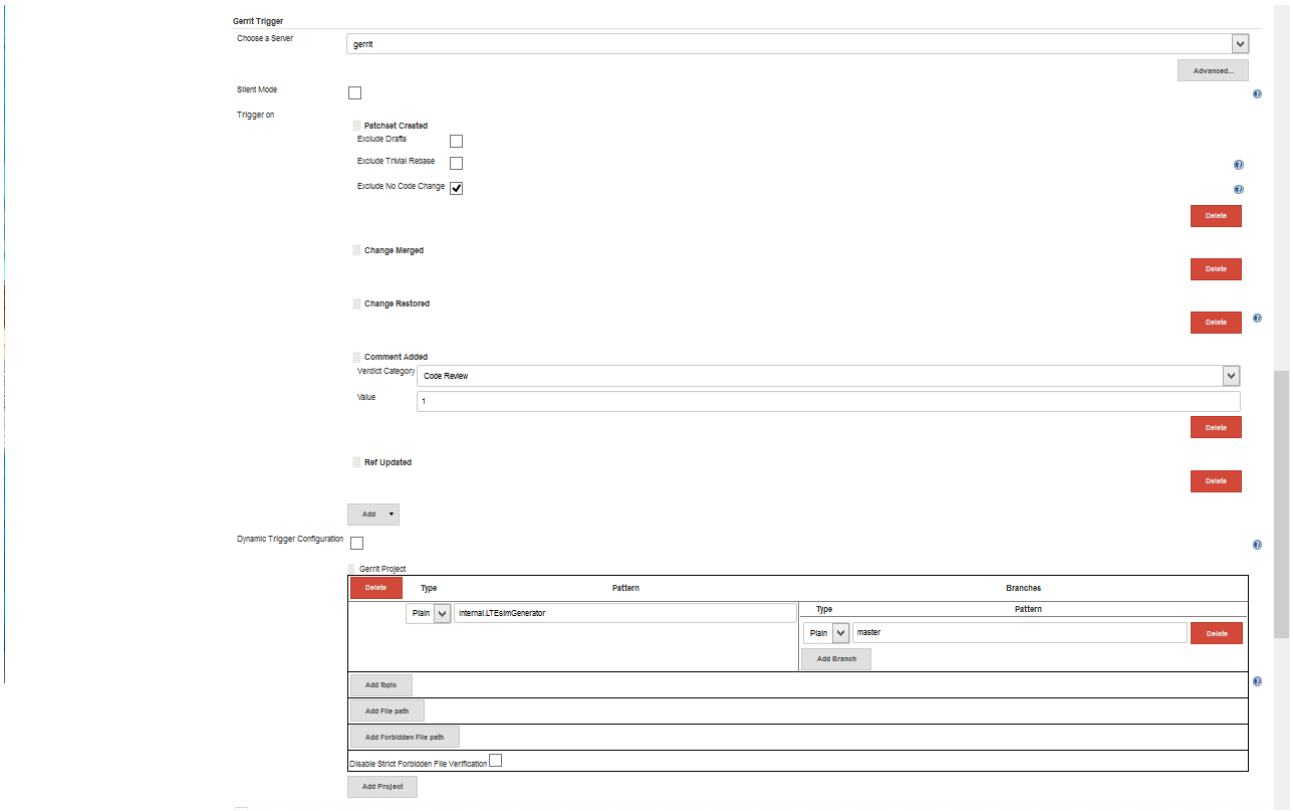


Pic 10 Project configuration - source code management.

In build trigger section, you can choose **build periodically** to make some builds on specified hour.

Pic 11 Build trigger periodically.

To start Gerrit and Jenkins cooperation you need to choose also **gerrit trigger** as shown below:



Pic 12 Gerrit trigger.

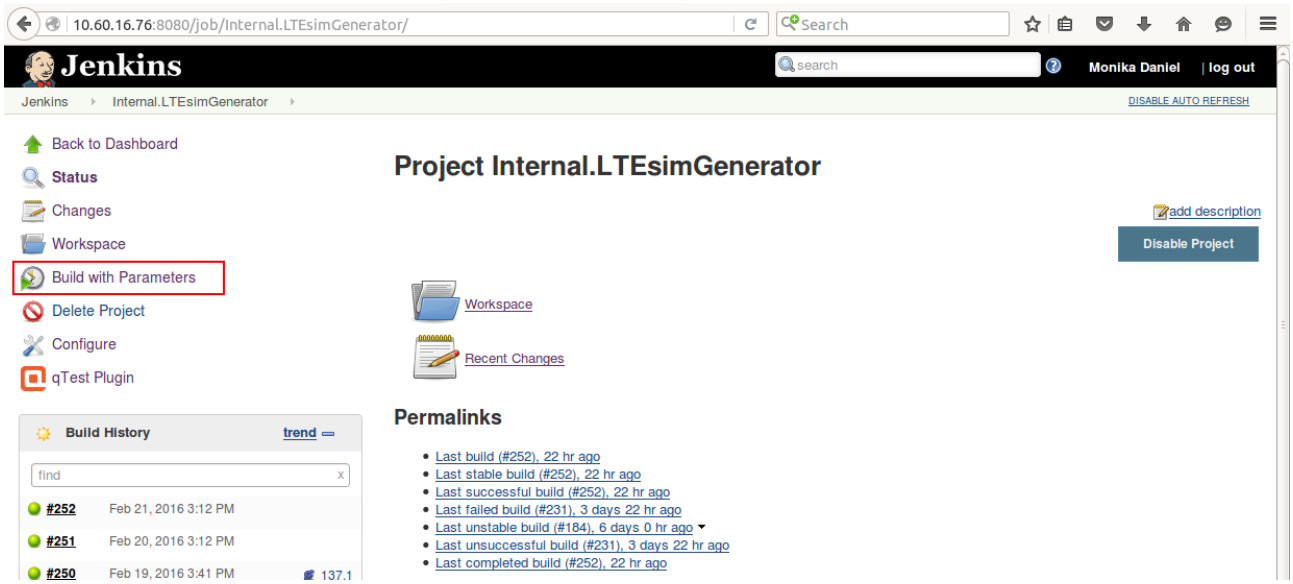
Next section **Build** is described in point 5 of this document.

To use **Post build actions** you need to install **Text Finder plugin** which is also described in point 6 of this document.

5. Initial build:

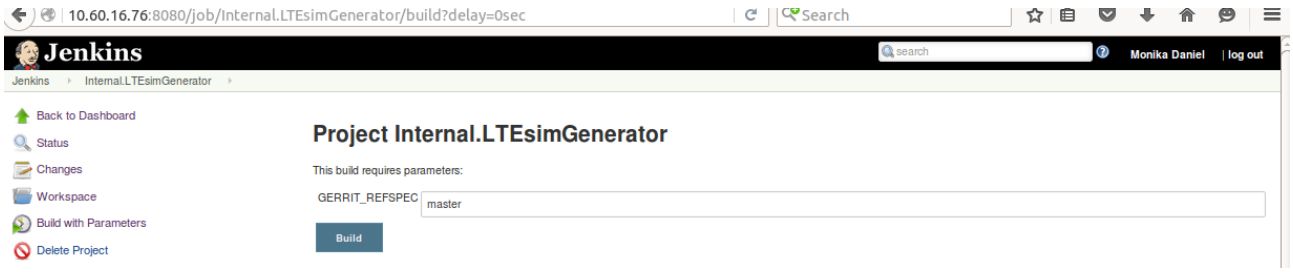
Before you add some script and post build actions you need to make an initial build with project on your repository.

- Go to your project
- Choose *Build with parameters*



Pic 13 Build with parameters.

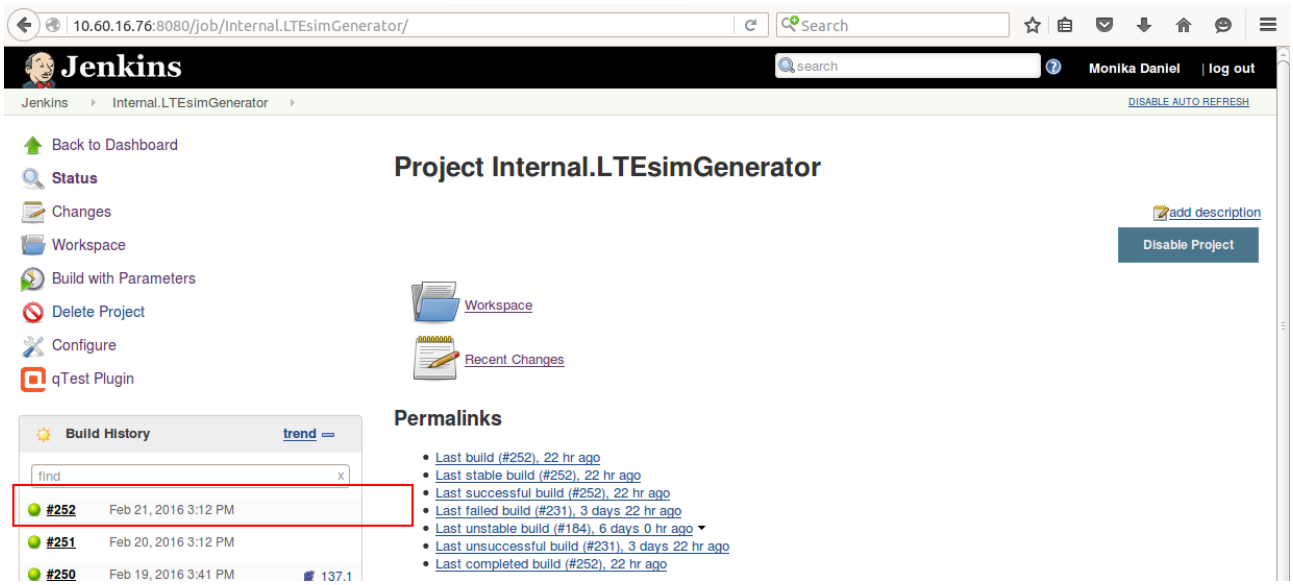
Then check if in **GERRIT_REFSPEC label** is written **master** and click on **Build button**:



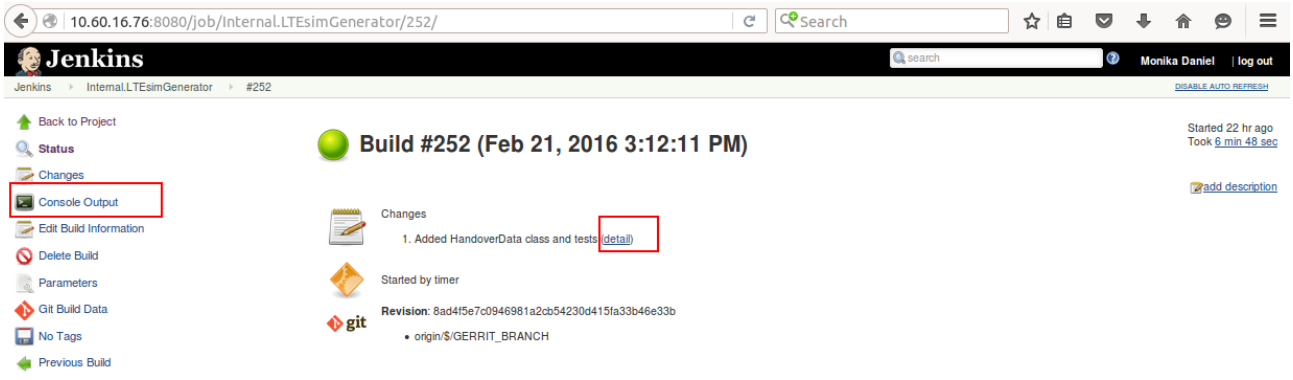
Pic 14 Building with parameters.

6. Builds:

Click on your build to see the result:

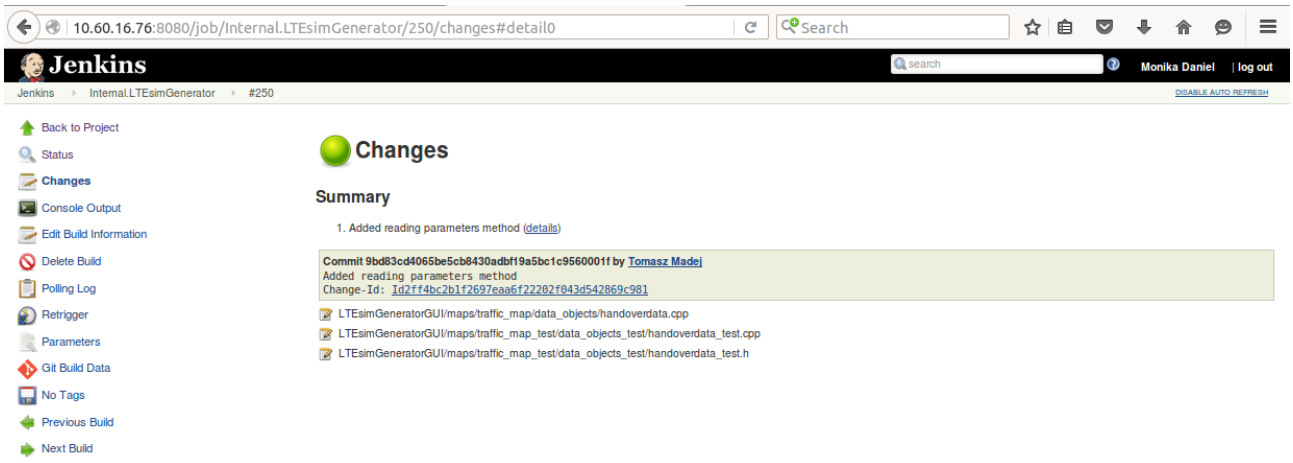


Pic 15 List of builds.



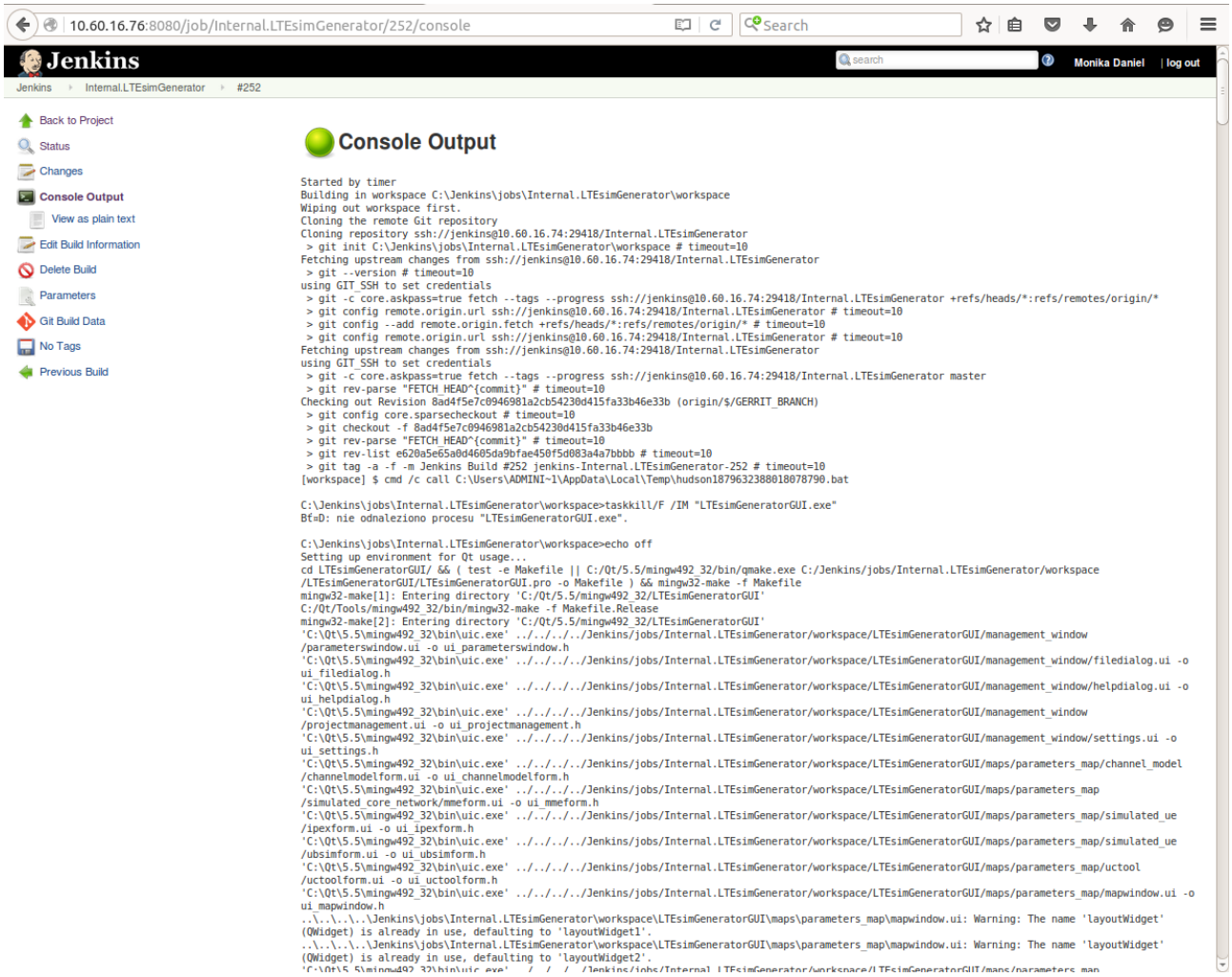
Pic 16 Build 252.

Now you can see **details**:

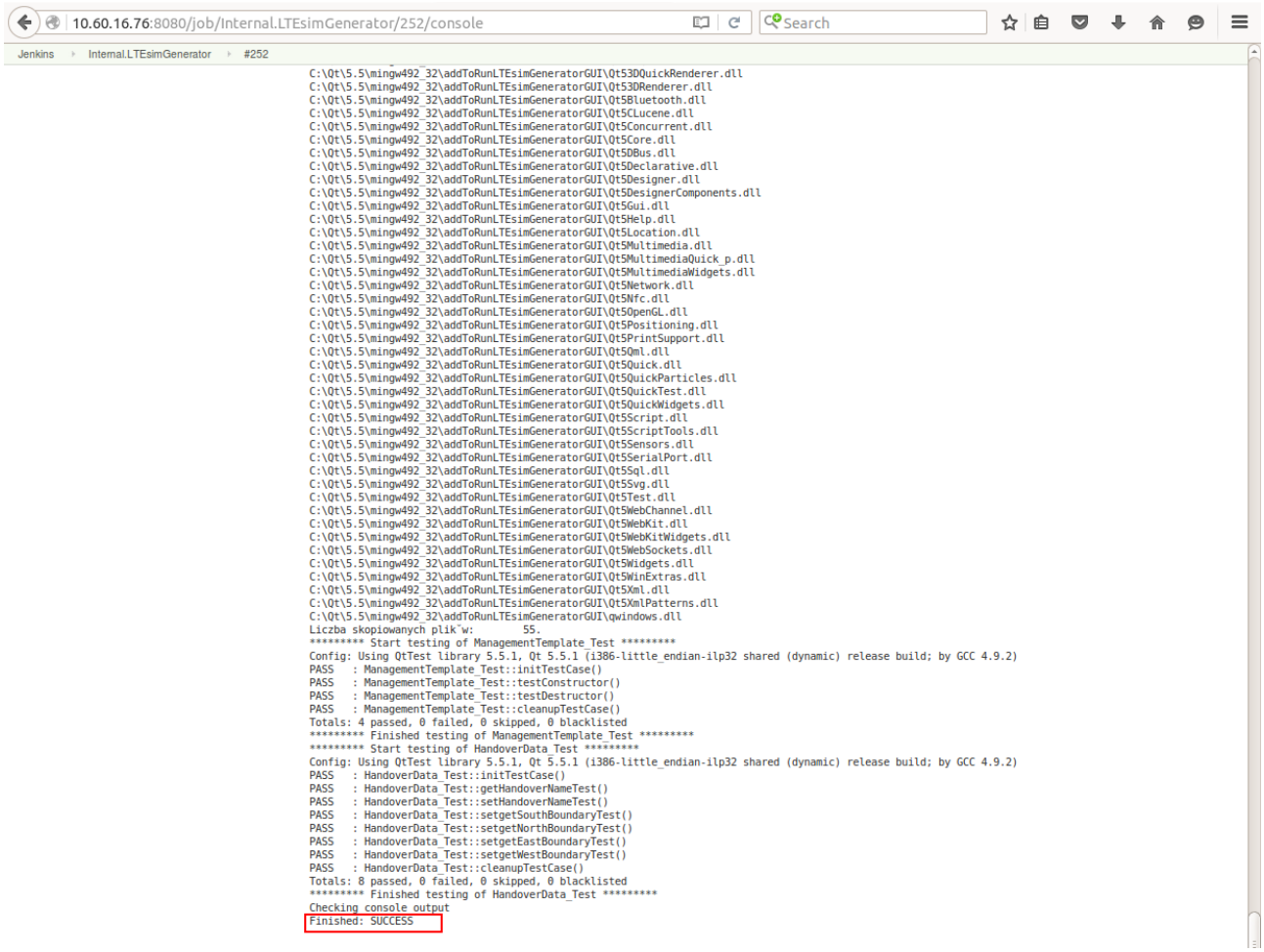


Pic 17 Build details.

or **Console Output**:



Pic 18 Console output - start.

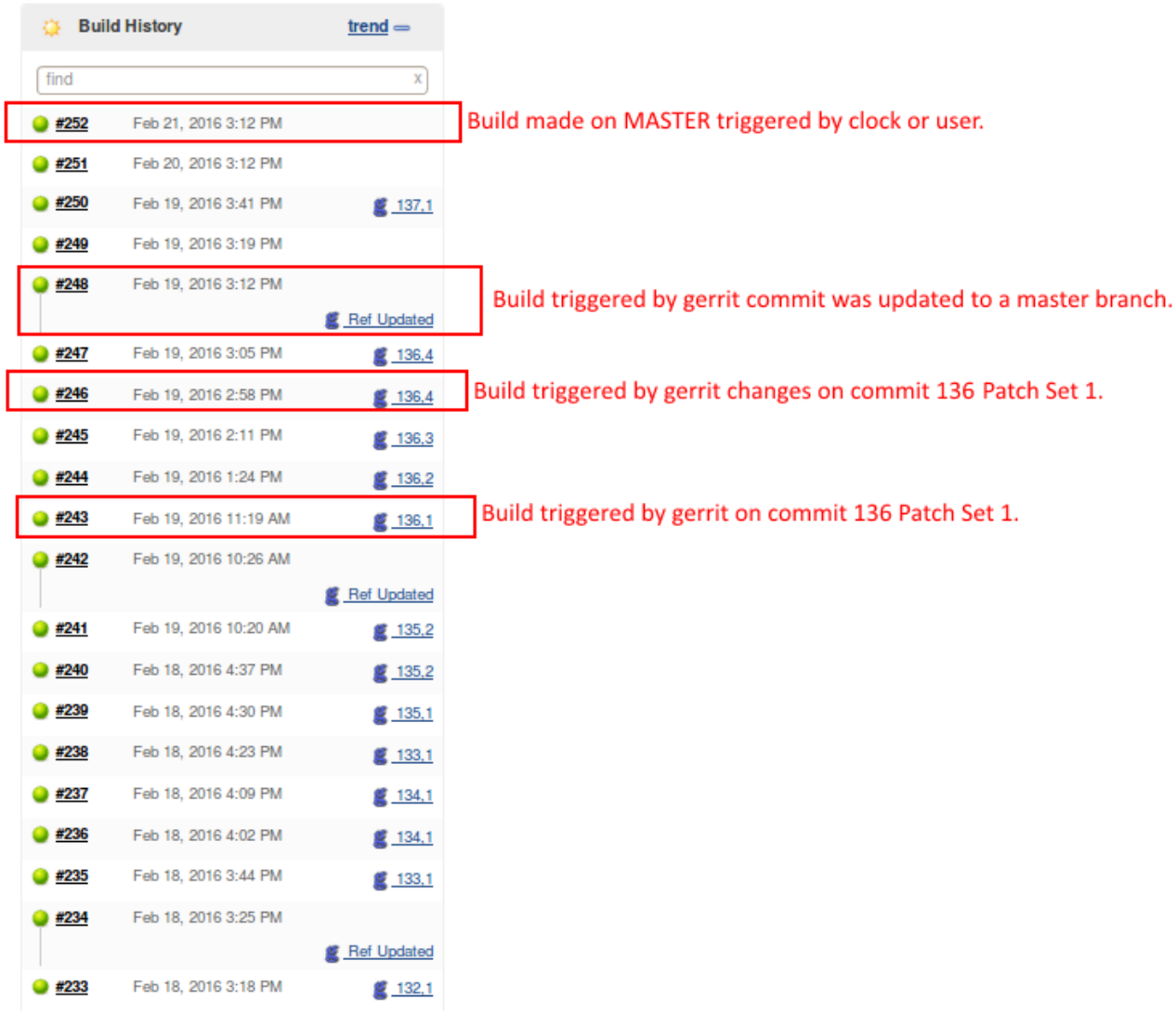


Pic 19 Build result.

When a circle is **green** it means that a result is **SUCCESS, red** one means **FAIL** and yellow **UNSTABLE build**.

If a circle is **blinking** it means that Jenkins is already **building repository**.

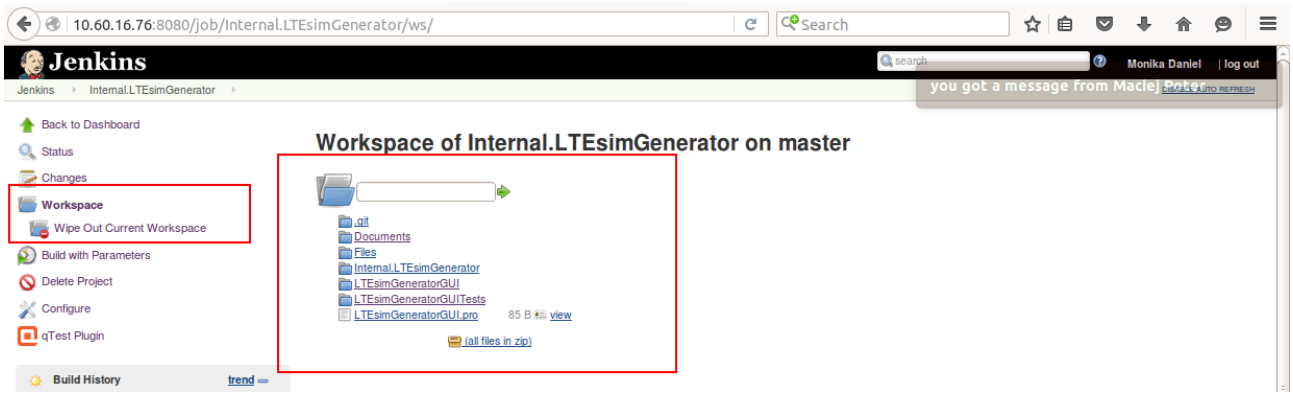
You can also read from a build history why were they triggered, on which branch or commit and Patch Set.



Pic 20 How to recognize builds.

7. Workspace:

When your initial build will succeed go to your project and workspace to see if it is complete:

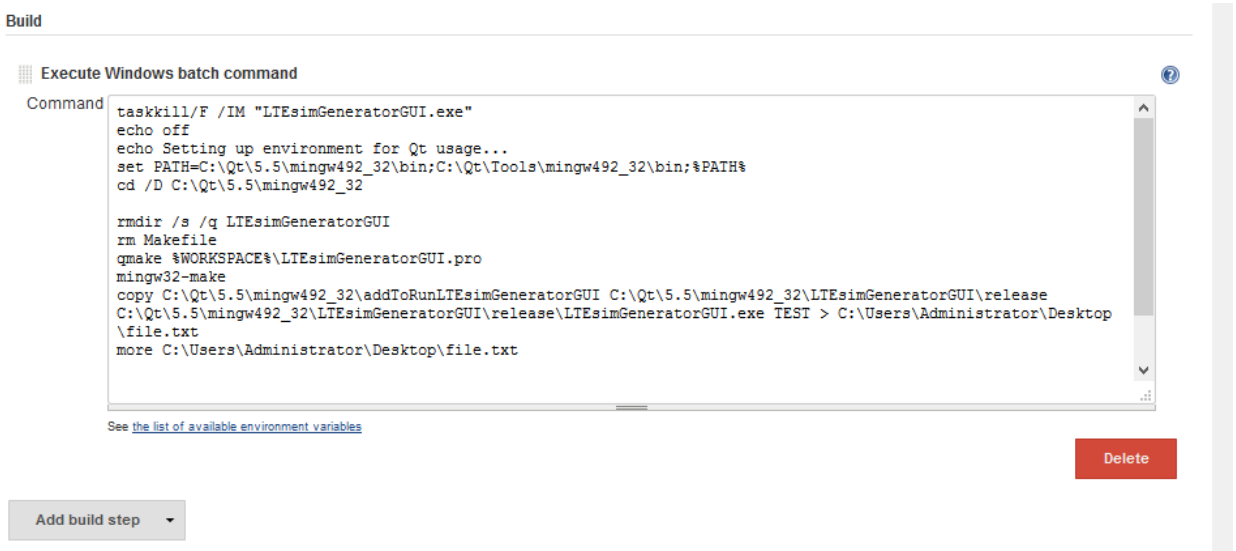


Pic 21 Workspace.

If you have some problems you can also **wipe out** your **current workspace** and build again to make again a new one.

8. Script for tests:

To write scripts to tests you have to add build step **Windows batch command.** Jenkins is installed on Windows 7 server that's why we can't write a shell script.



Pic 22 Build step - Windows batch command.

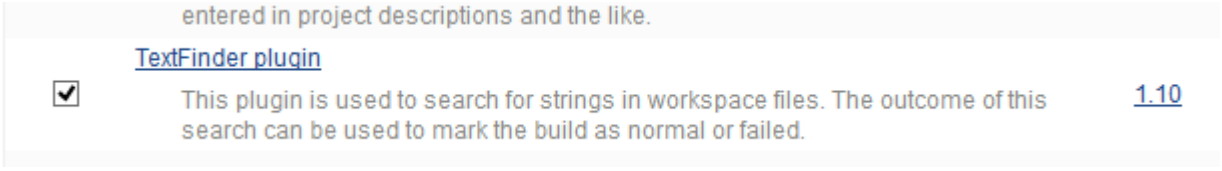
There is our script shown below with comments:

```
::Batch script Jenkins, Gerrit, Qt, Internal.LTEsimGeneratorGUI
::killing process if is active
taskkill/F /IM "LTEsimGeneratorGUI.exe"
::running QT environment
echo off
echo Setting up environment for Qt usage...
set PATH=C:\Qt\5.5\mingw492_32\bin;C:\Qt\Tools\mingw492_32\bin;%PATH%
cd /D C:\Qt\5.5\mingw492_32
::removing files from old build
rmdir /s /q LTEsimGeneratorGUI
::removing Makefile from old build
rm Makefile
::making new Makefile
qmake %WORKSPACE%\LTEsimGeneratorGUI.pro
::compile
mingw32-make
::copying needed libraries
copy C:\Qt\5.5\mingw492_32\addToRunLTEsimGeneratorGUI C:\Qt\5.5\mingw492_32\LTEsimGeneratorGUI\release
::running .exe file with tests using TEST parameter and saving results to a file
C:\Qt\5.5\mingw492_32\LTEsimGeneratorGUI\release\LTEsimGeneratorGUI.exe TEST > C:\Users\Administrator\Desktop\file.txt
::displaying results in commandline
more C:\Users\Administrator\Desktop\file.txt
```

Pic 23 Script with comments.

9. Text Finder Plugin:

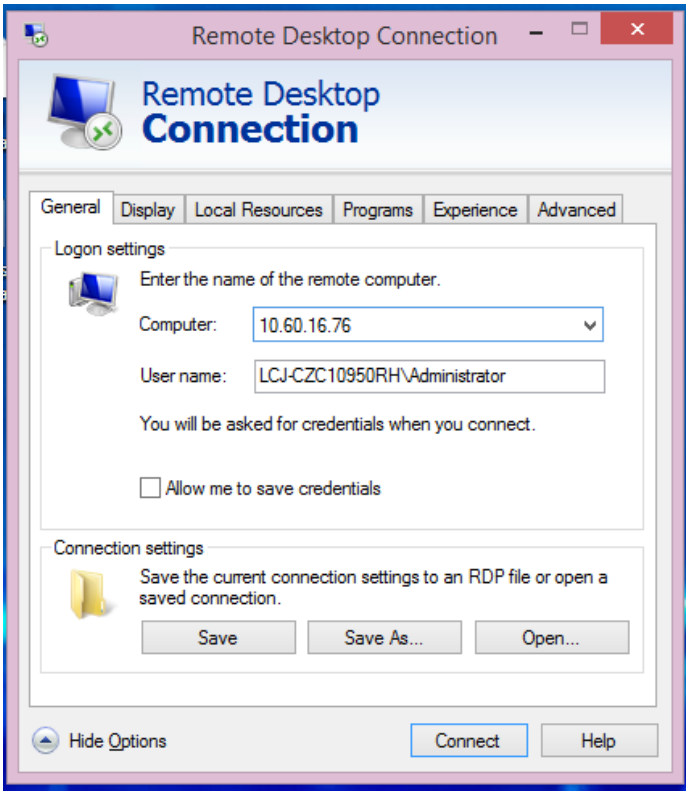
First you need to install this plugin. Go to the projects board and choose **Manage Jenkins, Manage Plugins, Available** and find TextFinder plugin.



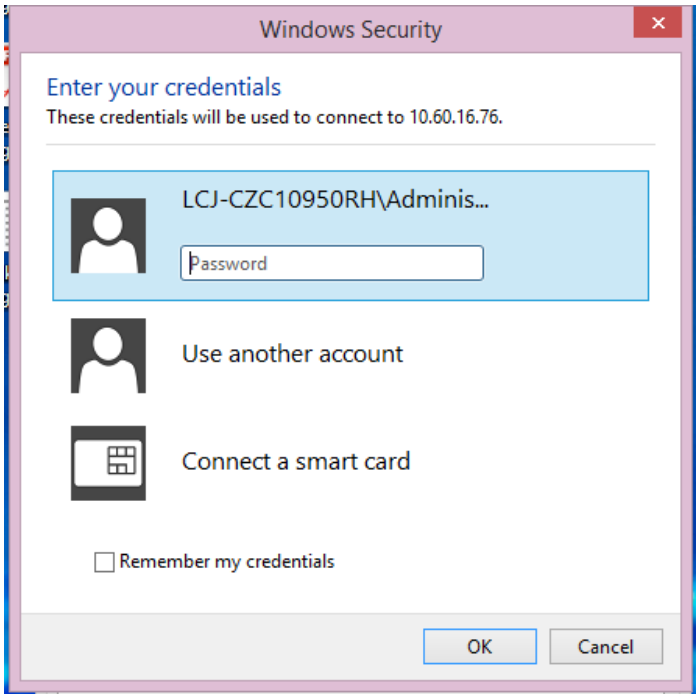
Pic 24 TextFinder plugin.

10. Qt Creator and Jenkins:

Use **Remote Desktop Connection** on your computer to connect to server with Jenkins. As computer use **10.60.16.76** and as user name **LCJ-CZC10950RH\Administrator**. For password ask your superior.

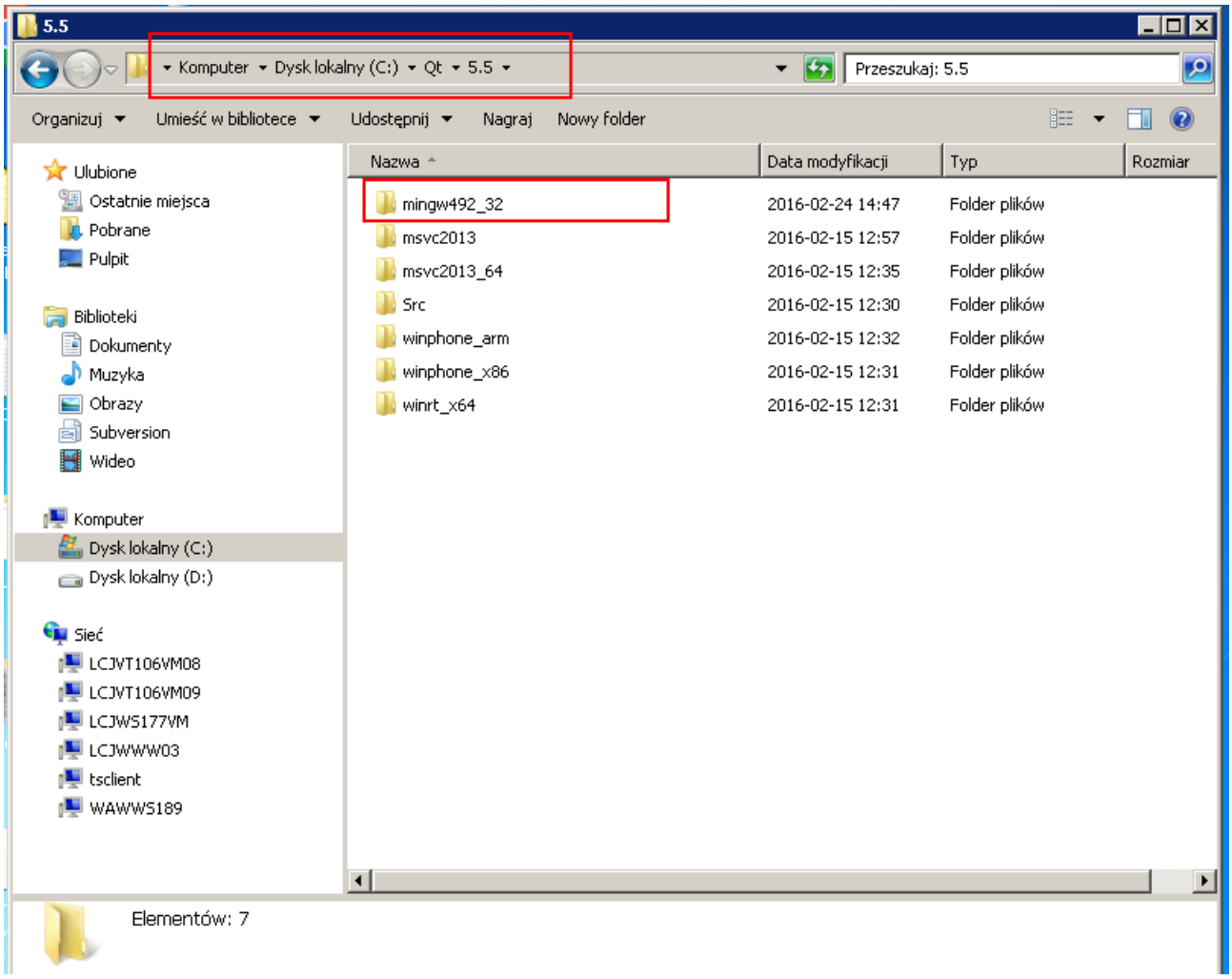


Pic 25 Remote Desktop Connection to a Jenkins server.



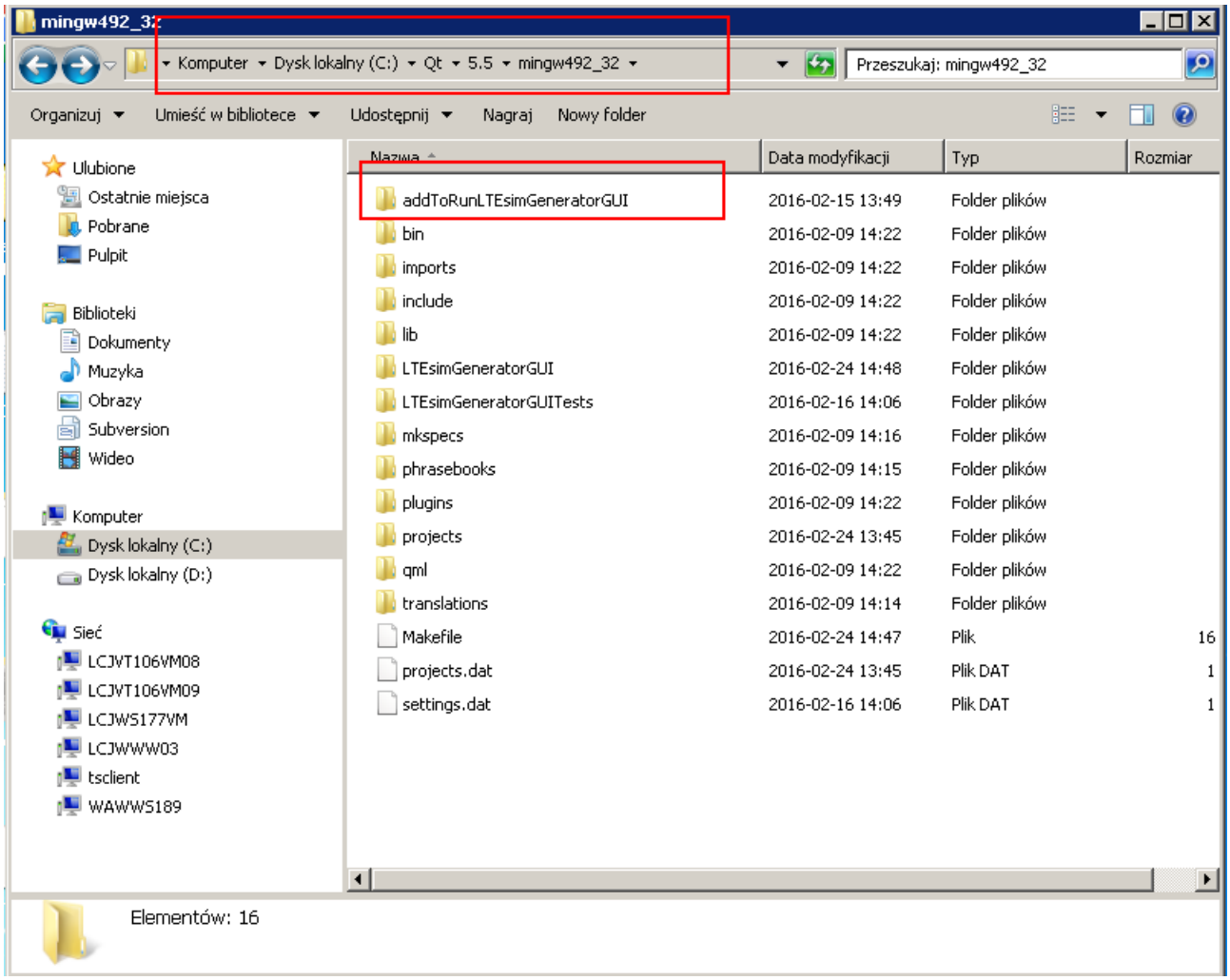
Pic 26 Remote Desktop Connection to a Jenkins server part 2.

Install Qt Creator in **C:\Qt** direction. It should have 5.5 version with **C:\Qt\5.5\mingw492_32** compiler.



Pic 27 Ot Creator destination and compilers.

Create a catalog **addToRunLTEsimGeneratorGUI** in **C:\Qt\5.5\mingw492_32** destination and copy all libraries without “d” at the end of name, from **bin** catalog which is locate in the same path.



Pic 28 Catalog with libraries.

Now you can use your Jenkins with Gerrit and Qt without any problems :)

11. Useful sources:
    - https://wiki.jenkins-ci.org/display/JENKINS/Meet+Jenkins