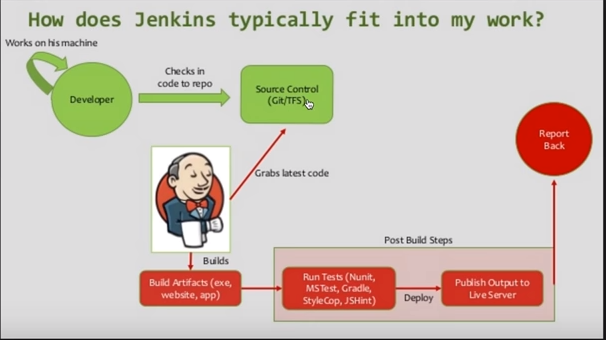
Jenkins is Java application since it is a java application it is platorm independent.

It is used for continous integration and Continous delivery.

When we work in team we have different developers who develop the code and keep on working on changing or enhancing the code and as the developer walks on his own machine and he changes the code he will check in the code into shared repository which can be Git or bit bucker or TFS . Now what happens lets suppose through the entire day there are many developers who are changing the code and checking in the code in the repository at the end of the day when you have a build lets suppose there is some bug introduced in any of the code and now the build failed. Now it would be very difficult for us to identify what exact code failed the build and at what point was this bug was introduced in the code and then you have to go back and check in you know all the code that was created that day and there will be a lot of confusion. So here come jenkins do is as soon as the developer commits the code in the shared repository Jenkins will take that latest code and trigger a build and the build notification will be sent out so in case there is problem in the build or there is an error you will be get and notified as soon the build trigger builds and completes. So we can check that if there is any issue due to any commits we can reverses that and we will not waste a lot of time in finding out what code caused the break . The other thing is let’s suppose the build is succesfull we can also integrate our unit test or acceptance test or performance test along with build as a post build actions in jenkins and it will be automated as soon as the build gets deployed jenkins will trigger some test cases or some testing which will be automated and it will send out the report back to us.Lets suppose the build was successful and there was no exceptions but due to changes in the code there was something which broke in the application so we will come to know instantly that there is some break due to the code and we can troubleshoot it. You can see how efficient and powerful the entire system can be and this what is called as continuos integration we are taking the code and doing a build as soon as the code gets commit and then we are also triggering the test and we are sending back the reports. So as soon as there is any change in the code this entire cycle gets triggered and we come to know if there is any issue with the application so this is what the continous delivery is all about and this is how jenkins fit into the picture



Step 1 : Download Jenkins war file - [https://jenkins.io/](https://www.youtube.com/redirect?q=https%3A%2F%2Fjenkins.io%2F&v=89yWXXIOisk&event=video_description&redir_token=Dmr5fVrwyGBcrzDD6TV247TLaJJ8MTU0MTY0ODE1OEAxNTQxNTYxNzU4)

Step 2 : Place the war file into any location on your system

Step 3 : goto command prompt (windows) | terminal (mac) - goto folder where jenkins.war is - java -jar jenkins.war

Step 4 : goto browser - [http://localhost:8080](https://www.youtube.com/redirect?q=http%3A%2F%2Flocalhost%3A8080&v=89yWXXIOisk&event=video_description&redir_token=Dmr5fVrwyGBcrzDD6TV247TLaJJ8MTU0MTY0ODE1OEAxNTQxNTYxNzU4) (Jenkins window should show up)

Step 5 : install required plugins. Installed plugins will be available in the folder C:\Users\Narayana\.jenkins\plugins

Step 6 : get started with Jenkins

**How to setup jenkins on Tomcat?**

By default jenkins wll run on standalone server (jetty/winstone)

Why we need to use Tomcat instead of jetty server?

**How to Change Home directory?**

Home directory of jenkins is whenever we are deployed Jenkins on any system there is a folder where which is created by default and most of the times the name of the folder is .jenkins and the default location will be user profile of that system and this folder and this directory contains all the information about the logs, configuration, jobs everything about jenkins.

Jenkins home directory contains:

1)All configurations

2)Plugins

3)Jobs details

4)Logs

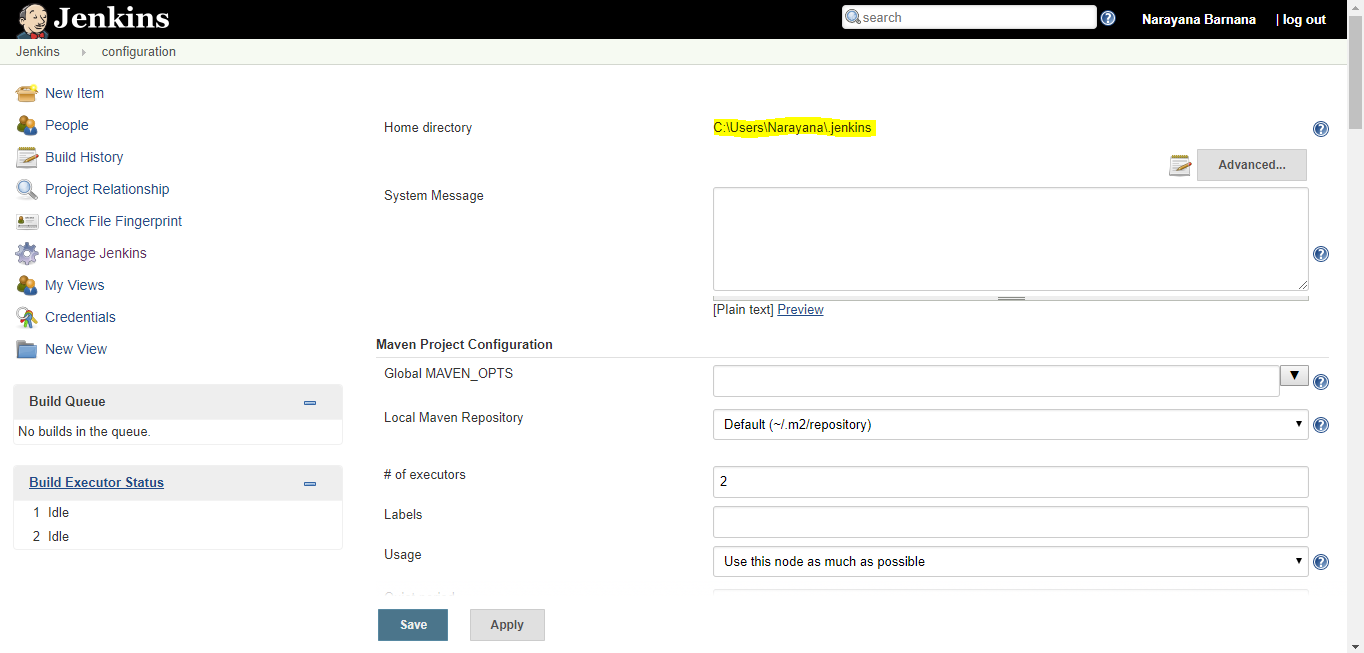
**Why do we want to change home directory ?**

1. To move jenkins home dir to a location that has enough space because there will be lot of jobs added to our jenkins , there will be lot of plugins added and other configurations and we want to place this home directory in such a location where we have sufficient space so we want to change it.
2. Project requirements

How to Change?

Step 1: check your current home directory.

* Go to the folder where jenkins.war file is located and run the command java -jar jenkins.war
* Open the URL <http://localhost:8080/> and go to manage jenkins🡪configure system and see the home directory.



Step 2: Create a new folder (which will be new home dir)

Step 3: Copy all data from old directory to new directory

Step 4: change environment variable JENKINS\_HOME and set to new directory. In case if you don’t have JENKINS\_HOME you can create it

Step 5: Restart jenkins by two ways

* Go to command prompt 🡪 CTRL + C 🡪 run the command java -jar jenkins.war
* localhost:8080/restart

Step 6: Validate the changed home directory in manage jenkins🡪Configure system

**How to use CLI – Command Line Interface**

Step 1 : start Jenkins

Step 2 : goto Manage Jenkins - Configure Global Security - enable security

Step 3 : goto - [http://localhost:8080/cli/](https://www.youtube.com/redirect?redir_token=q5cPdXnB6XWU9nJs9zRWPhIxgON8MTU0MTc3NjM2N0AxNTQxNjg5OTY3&q=http%3A%2F%2Flocalhost%3A8080%2Fcli%2F&v=ooA8RS3hC6k&event=video_description)

Step 4 : download jerkins-cli jar. Place at any location.(in my laptop E:\Jenkins)

Step 5 : test the jenkins command line is working by running the below command

java -jar jenkins-cls.jar -s http://localhost:8080 /help --username user --password pwd

java -jar jenkins-cli.jar -s http://localhost:8080/ help --username narayanabarnana --password Automation@009

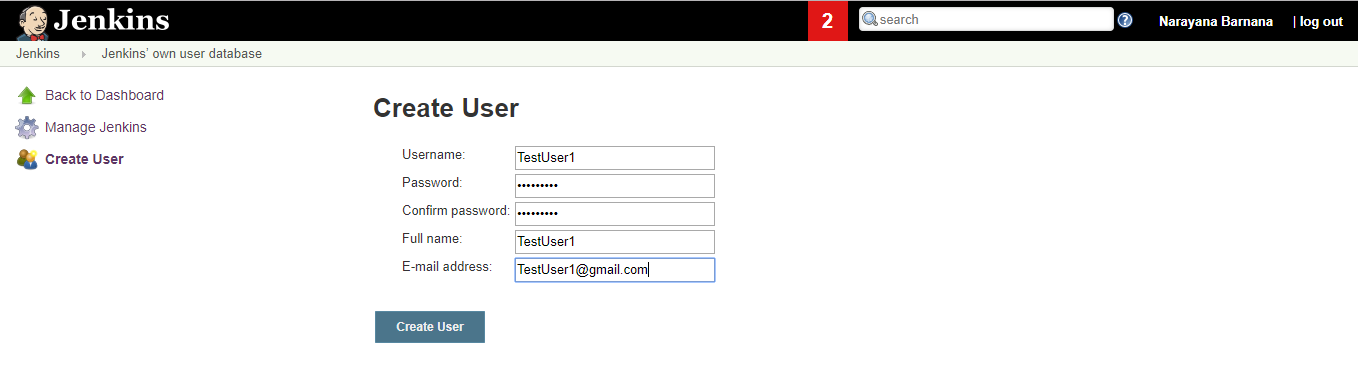
Step 6: Try running the command java -jar jenkins-cli.jar -s http://localhost:8080/ safe-restart.

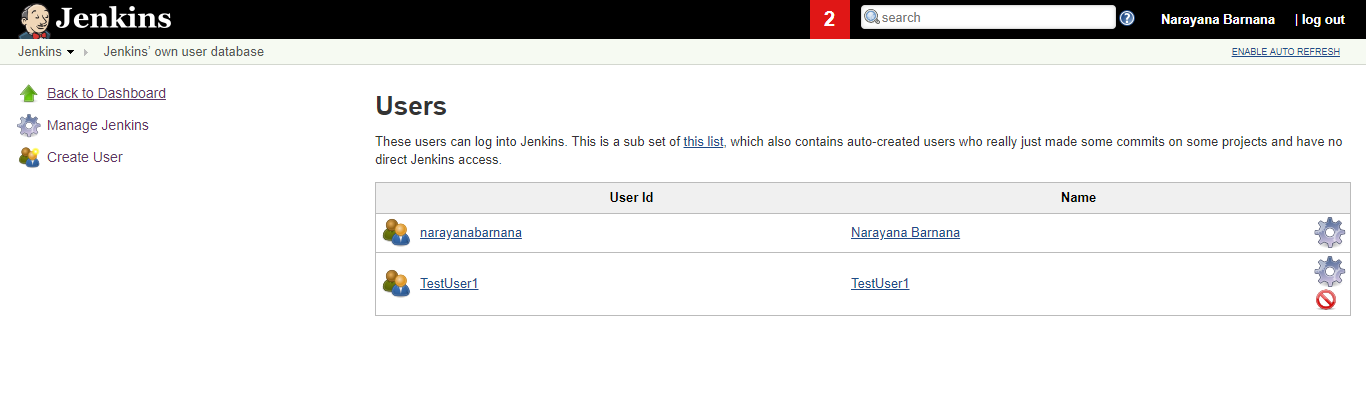
If you are getting the error ERROR: anonymous is missing the Overall/Read permission then go to manage jenkins🡪configure glogal security 🡪Select Anyone can do anything option from Authorization

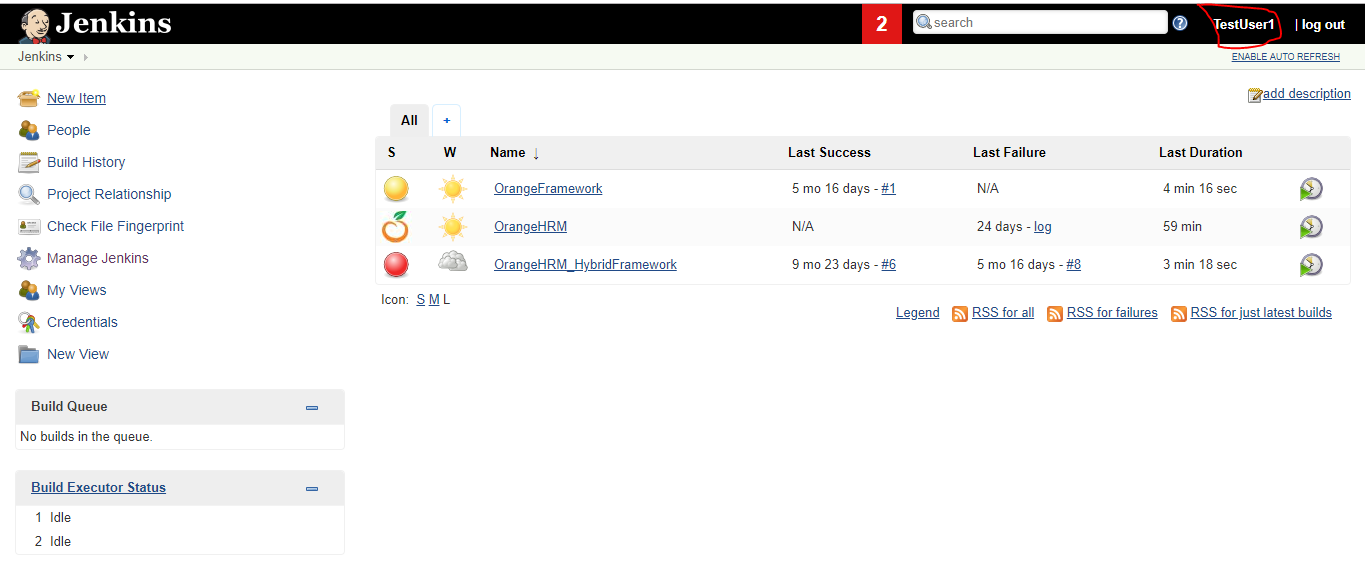
**How to create Users + Manage + Assign Roles**

Create Users:

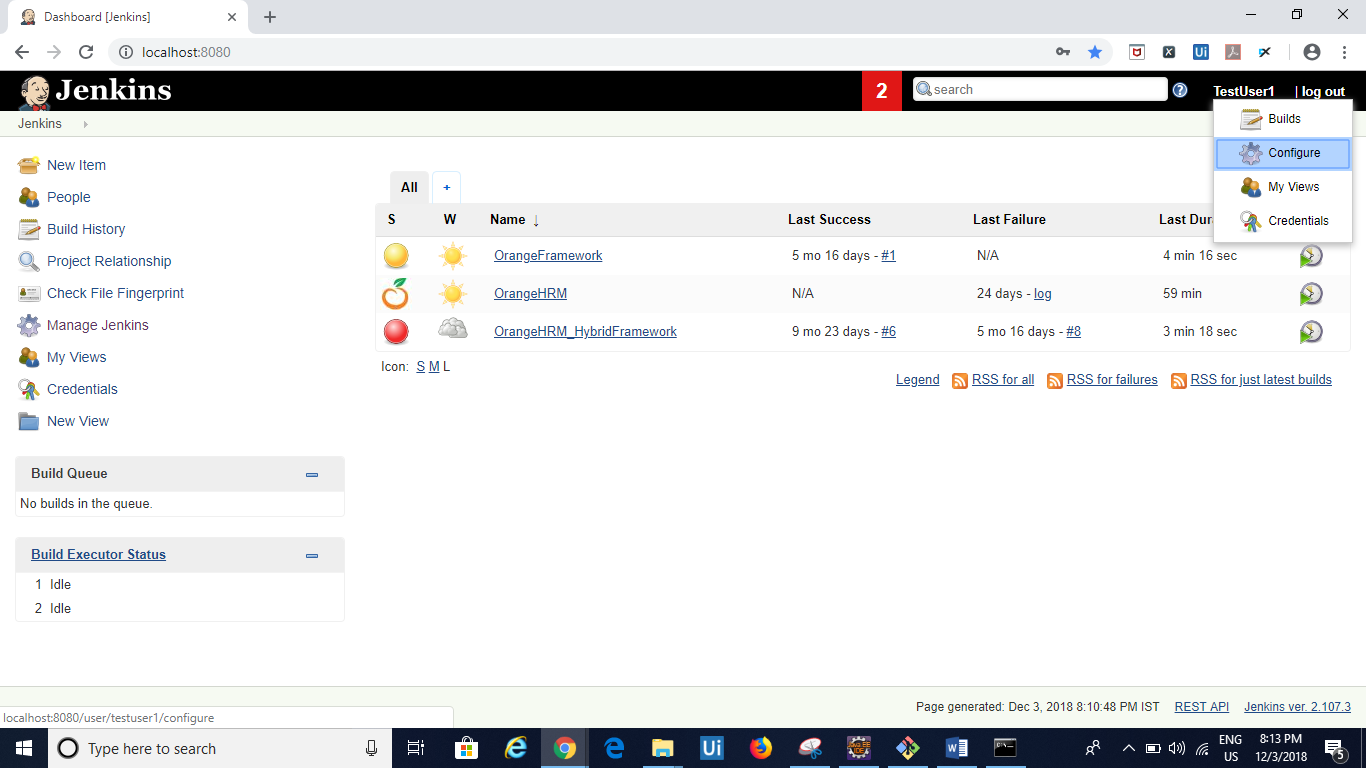
1)Manage Jenkins🡪Manage Users🡪Create User(which is on the left side) and give the details.

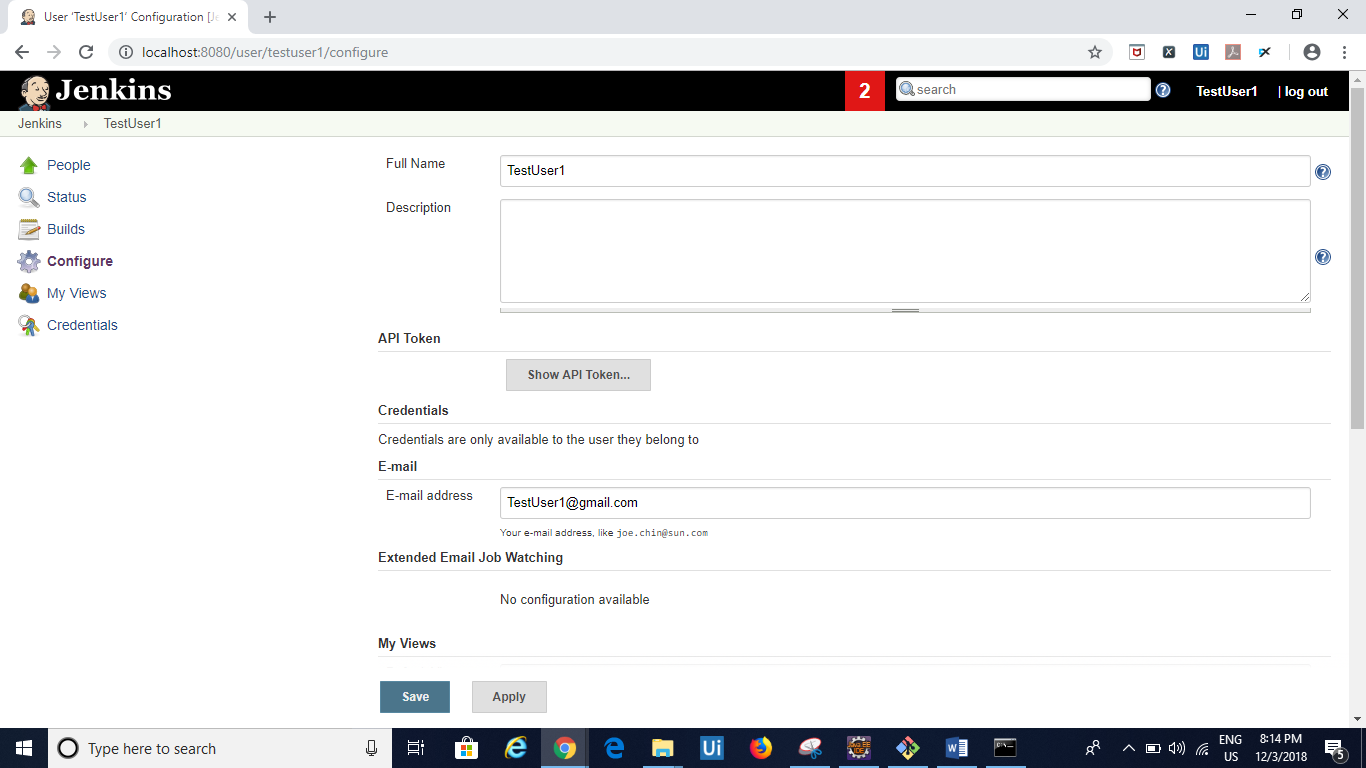






Configure Users: Go to Configure which is shown in the below screenshot and configure the details to the user





**Create and Manage Roles:**

Installl the Role Strategy plugin from the Plugin manager or download from google and navigate to Manage Jenkins🡪Manage Plugins🡪Advanced🡪Upload Plugin🡪Choose file and submit

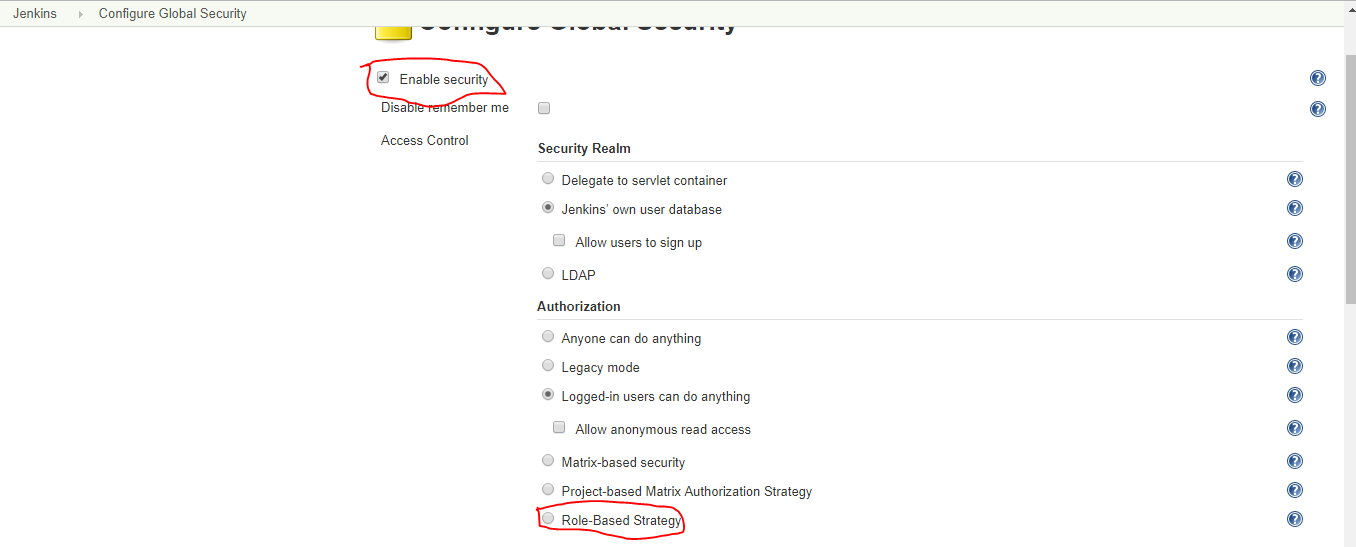
This plugin adds a new role-based strategy to ease and fasten users management. This strategy allows:

* Creating **global roles**, such as admin, job creator, anonymous, etc., allowing to set Overall, Slave, Job, Run, View and SCM permissions on a global basis.
* Creating **project roles**, allowing to set only Job and Run permissions on a project basis.
* Creating **slave roles**, allowing to set node-related permissions.
* Assigning these roles to users.

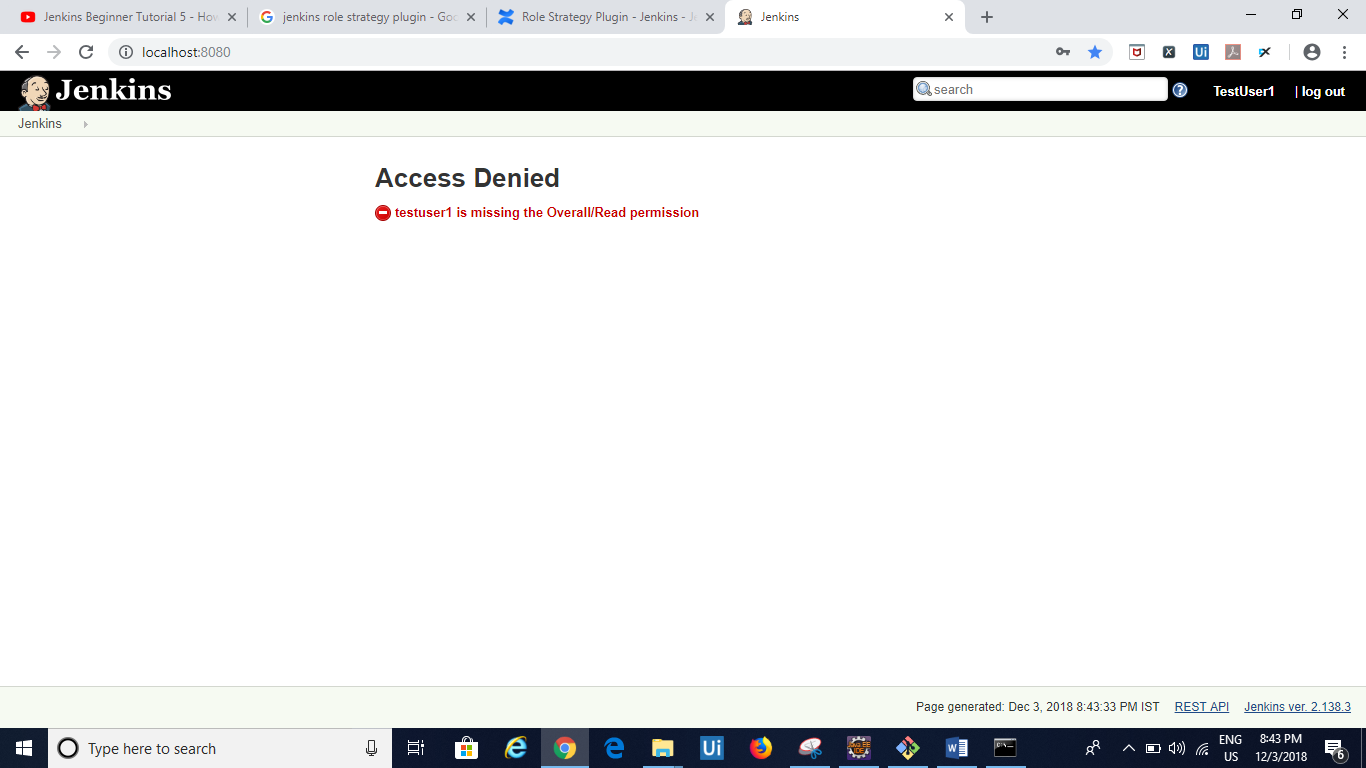
After installation restart the Jenkins

Naviagate to Manage Jenkins - Configure Global Security - Authorisation - Role Based Strategy

Check the below

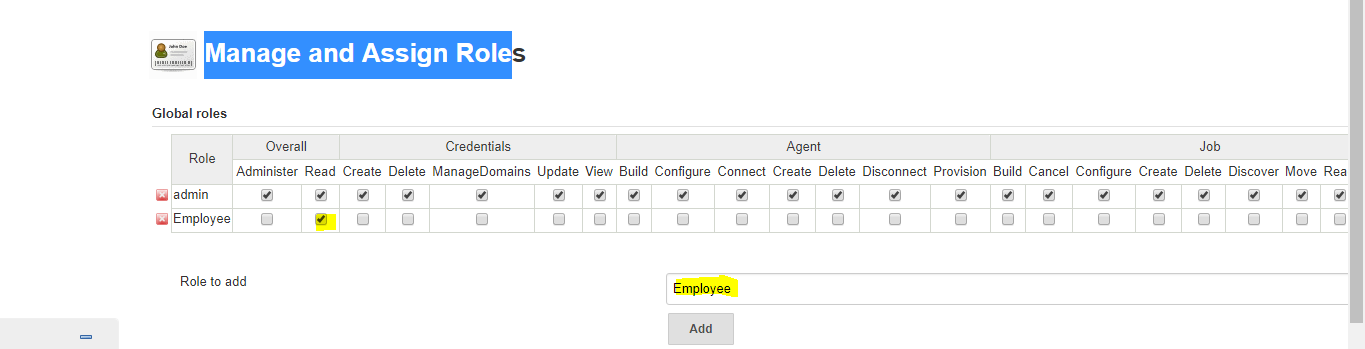


After this try to login with the created users and then you will get the below error as we are using the role based authorization strategy

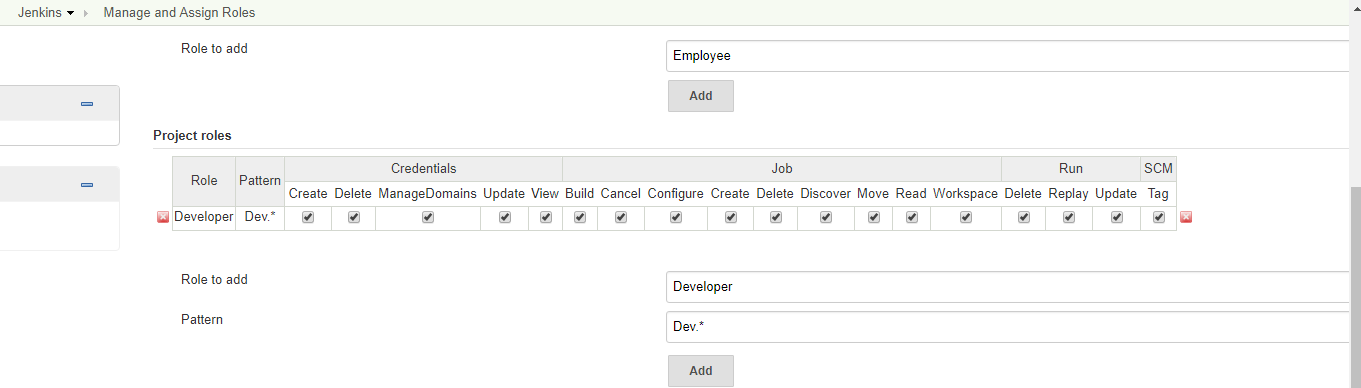


To add Roles Navigate to Manage Jenkins🡪Manage and Assign Roles🡪Manage Roles

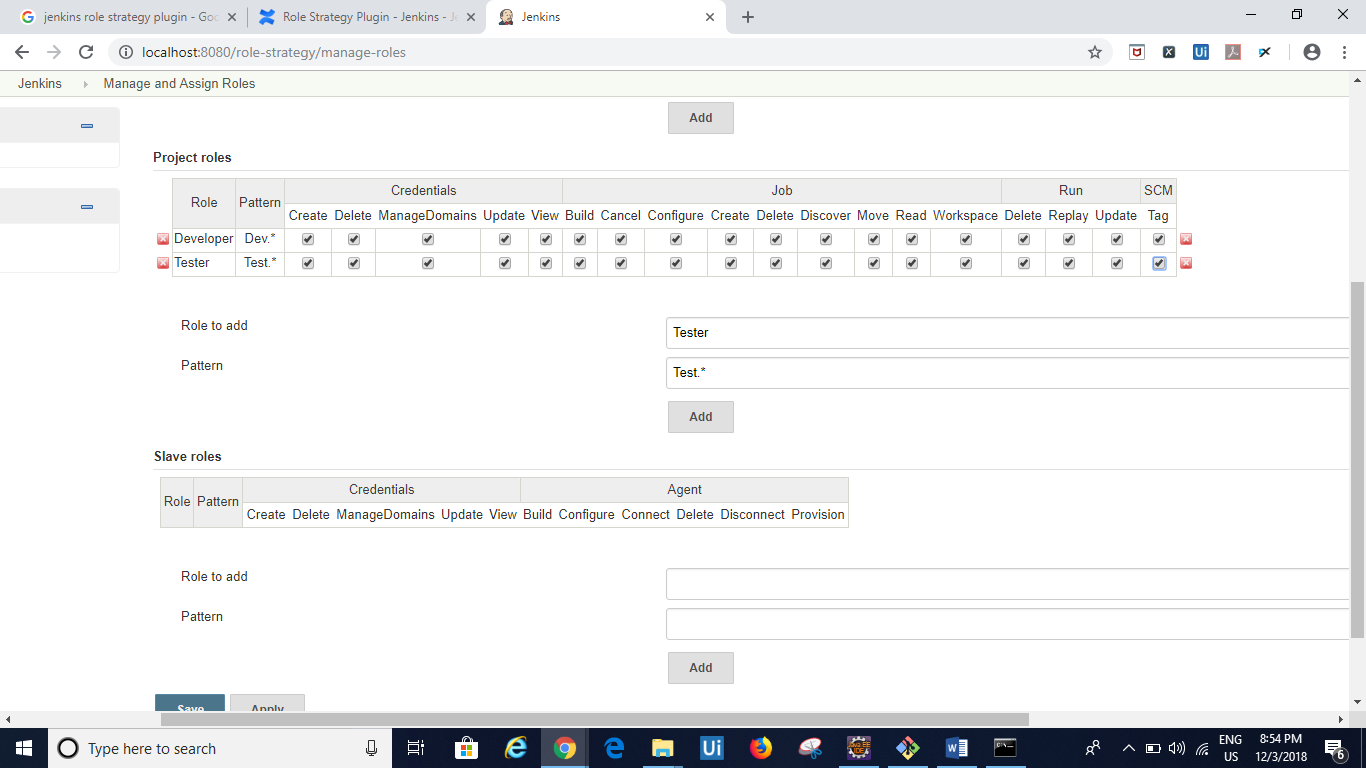
Add One role with name “Employee” and give the permission for Read and view



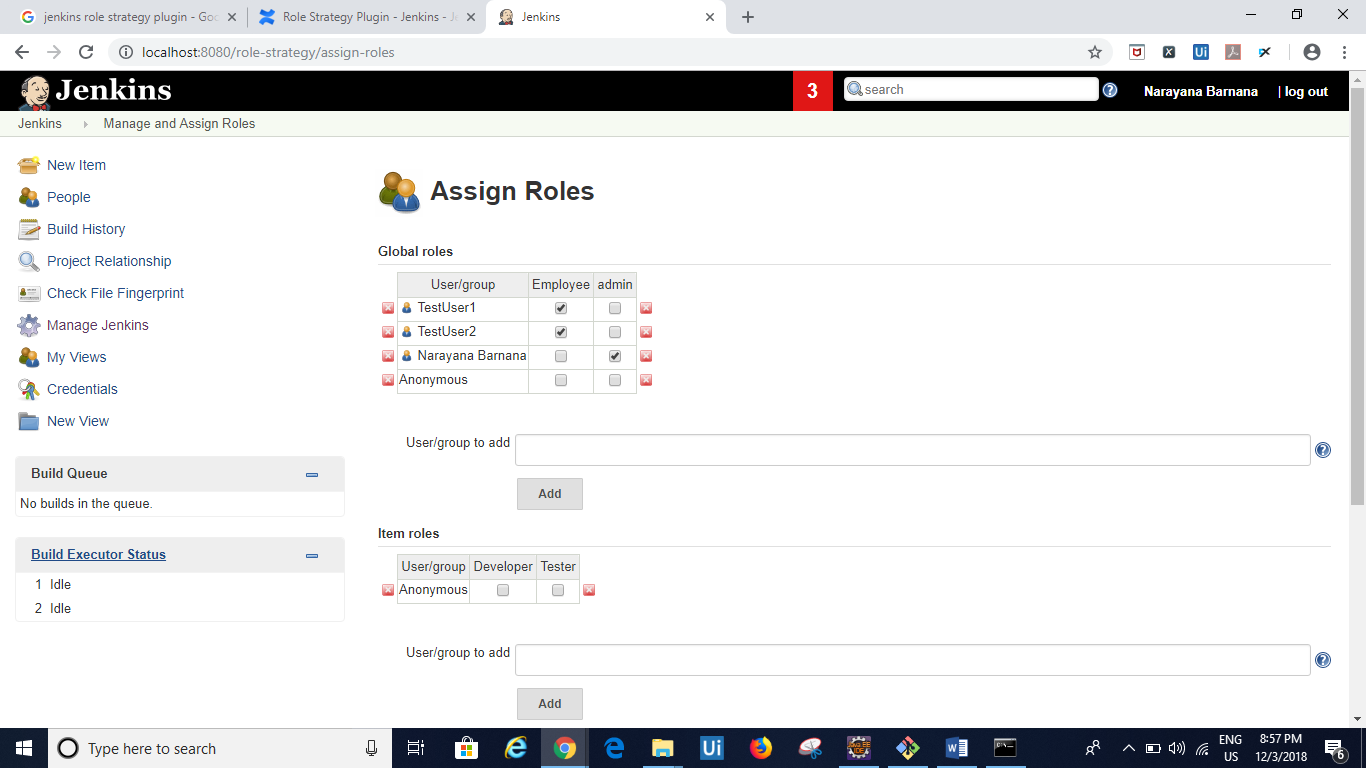
Add Project role with Name below and give all roles



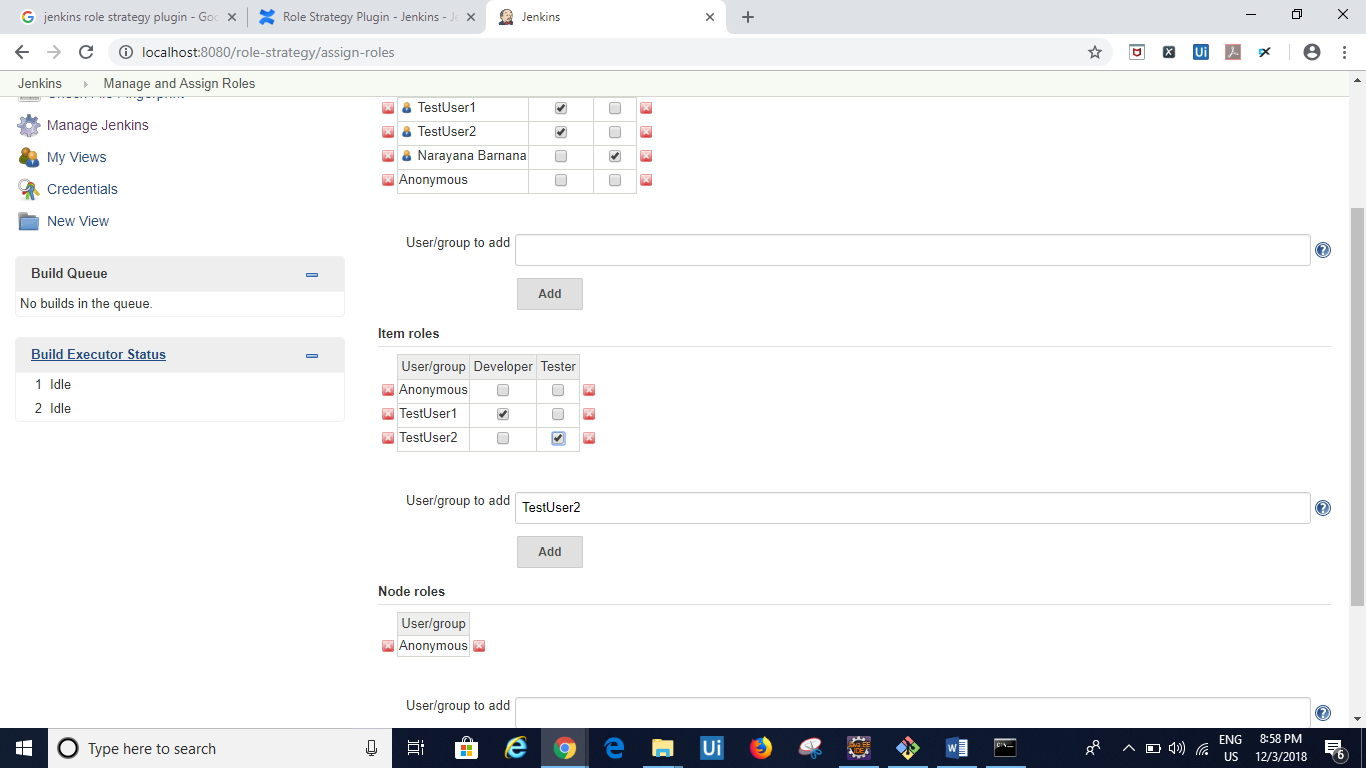
Add another project role with Name below and give all permissions.



Navigate to Assign Roles🡪Add the users in the Global Roles and assign Employee role

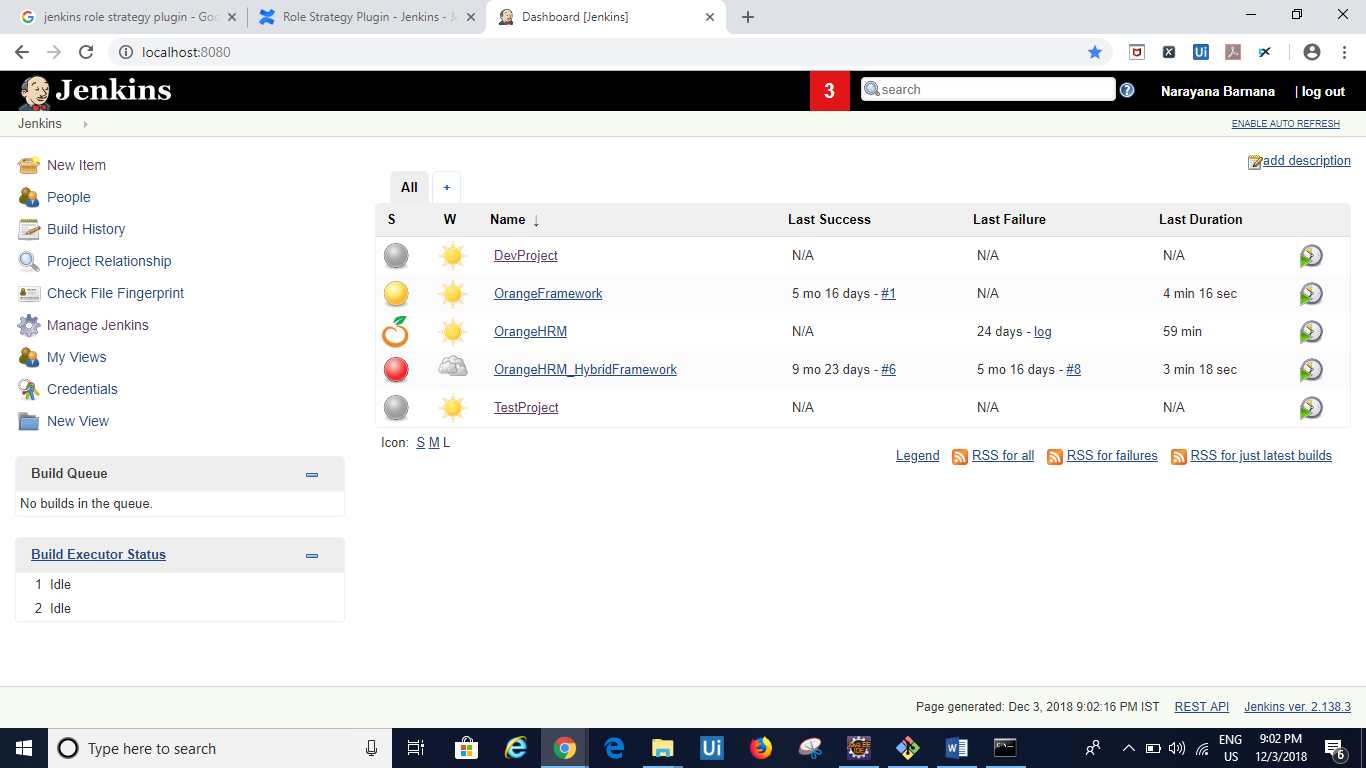


Add the users in the Project roles and assign Dev role for user 1 and tester role for user2



To validate

1)Create two free style projects with Name DevProject and TestProject



2)Logout and Login with TestUser1