**GIT VERSION CONTROL WITH ECLIPSE**

**Table of Contents**

1. [Installation of Git support into Eclipse](http://www.vogella.com/tutorials/EclipseGit/article.html#installation-of-git-support-into-eclipse)
2. [Git user settings in Eclipse](http://www.vogella.com/tutorials/EclipseGit/article.html#git-user-settings-in-eclipse)
3. [Configuring the toolbar and the menu for Git usage](http://www.vogella.com/tutorials/EclipseGit/article.html#configuring-the-toolbar-and-the-menu-for-git-usage)
4. [Working with Eclipse projects in a Git repository](http://www.vogella.com/tutorials/EclipseGit/article.html#working-with-eclipse-projects-in-a-git-repository)

[4.1. Adding a new project to a local Git repository](http://www.vogella.com/tutorials/EclipseGit/article.html#adding-a-new-project-to-a-git-repository)

[4.2. Adding a new project to a remote Git repository](http://www.vogella.com/tutorials/EclipseGit/article.html#adding-a-new-project-to-a-git-repository)

1. [Import (clone) projects from a remote GIT repository](http://www.vogella.com/tutorials/EclipseGit/article.html#import-projects-from-an-existing-repository)
2. Push local workspace changes to remote repository.
3. Pull latest changes from remote repository into local workspace.
4. Creating another local branch.
5. Push changes into remote repository from another local branch.
6. Code Review(Pull Request)

[6. Performing Git operations in Eclipse](http://www.vogella.com/tutorials/EclipseGit/article.html#performing-git-operations-in-eclipse)

[6.1. Pull, push and fetch](http://www.vogella.com/tutorials/EclipseGit/article.html#pull-push-and-fetch)

[6.2. Basic team operations](http://www.vogella.com/tutorials/EclipseGit/article.html#basic-team-operations)

1. [6.3. Team operations available on the project](http://www.vogella.com/tutorials/EclipseGit/article.html#team-operations-available-on-the-project) Push changes into remote repository

[6.4. Amending a commit](http://www.vogella.com/tutorials/EclipseGit/article.html#amending-a-commit)

[7. Branching in Eclipse](http://www.vogella.com/tutorials/EclipseGit/article.html#branching-in-eclipse)

[8. Starting a merge operation in Eclipse](http://www.vogella.com/tutorials/EclipseGit/article.html#starting-a-merge-operation-in-eclipse)

[8.1. Merge](http://www.vogella.com/tutorials/EclipseGit/article.html#merge)

[8.2. Solving merge conflicts](http://www.vogella.com/tutorials/EclipseGit/article.html#solving-merge-conflicts)

[9. Rebasing a branch onto another branch](http://www.vogella.com/tutorials/EclipseGit/article.html#rebasing-a-branch-onto-another-branch)

[10. Git reset and Git reflog](http://www.vogella.com/tutorials/EclipseGit/article.html#git-reset-and-git-reflog)

[10.1. Moving the branch pointer with Git reset](http://www.vogella.com/tutorials/EclipseGit/article.html#moving-the-branch-pointer-with-git-reset)

[10.2. Finding "invisible" commits with the Reflog view](http://www.vogella.com/tutorials/EclipseGit/article.html#finding-invisible-commits-with-the-reflog-view)

[11. Using git cherry-pick](http://www.vogella.com/tutorials/EclipseGit/article.html#using-git-cherry-pick)

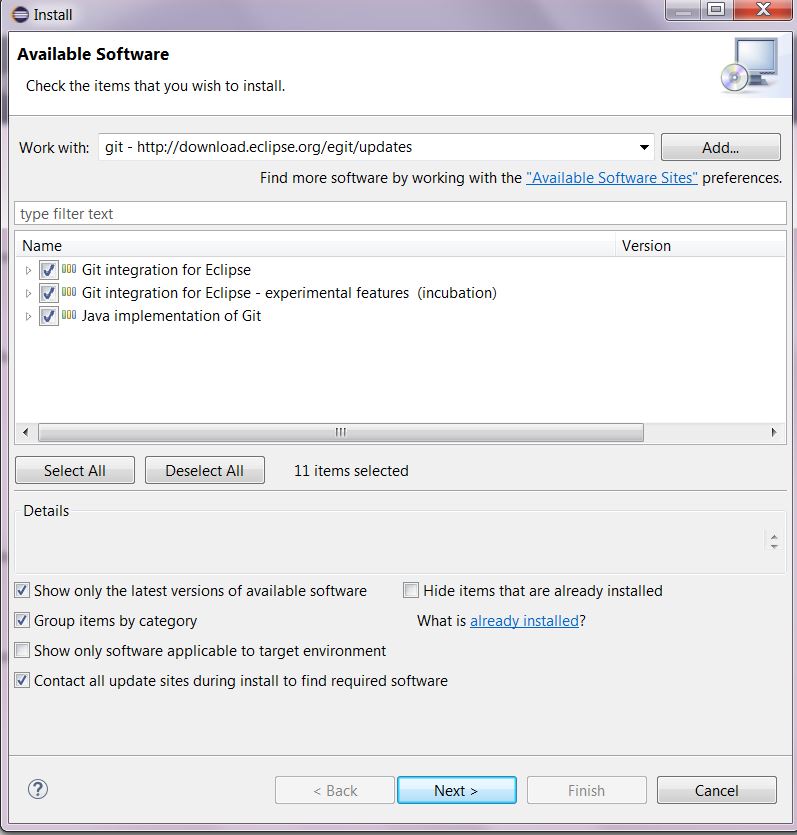
[12. Creating patches](http://www.vogella.com/tutorials/EclipseGit/article.html#creating-patches)

1. [**Installation of Git into Eclipse**](http://www.vogella.com/tutorials/EclipseGit/article.html#installation-of-git-support-into-eclipse)

* Most Eclipse IDE distributions from Eclipse.org already contain support for Git. In this case, no additional installation is required.
* Otherwise, you can install it from the Eclipse installation manager.
* Go to **Help ▸ Install new Software… ▸ Menu entry**.

Enter the following update site URL:

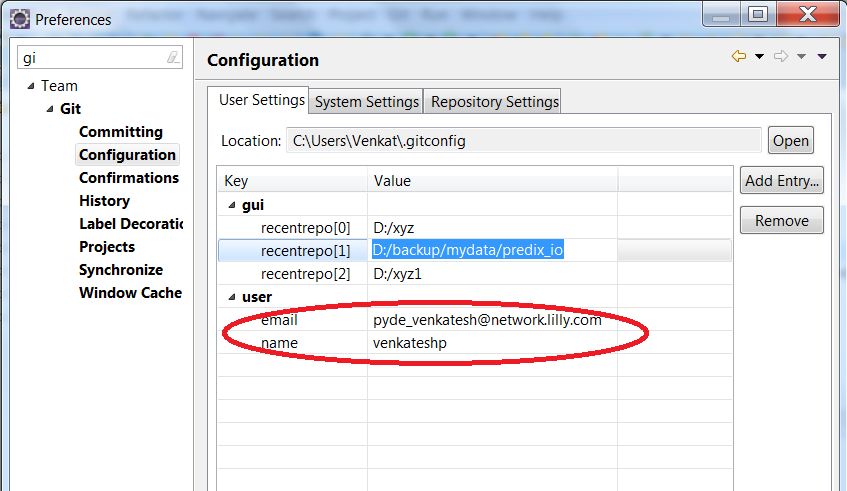
<http://download.eclipse.org/egit/updates>



1. [**Git user settings in Eclipse**](http://www.vogella.com/tutorials/EclipseGit/article.html#git-user-settings-in-eclipse)

* To use Git, you must configure your full name and email address.
* This information is used to fill the author and committer information of commits you create.
* These Git configuration settings can be adjusted via the Eclipse ***preference*s** setting.

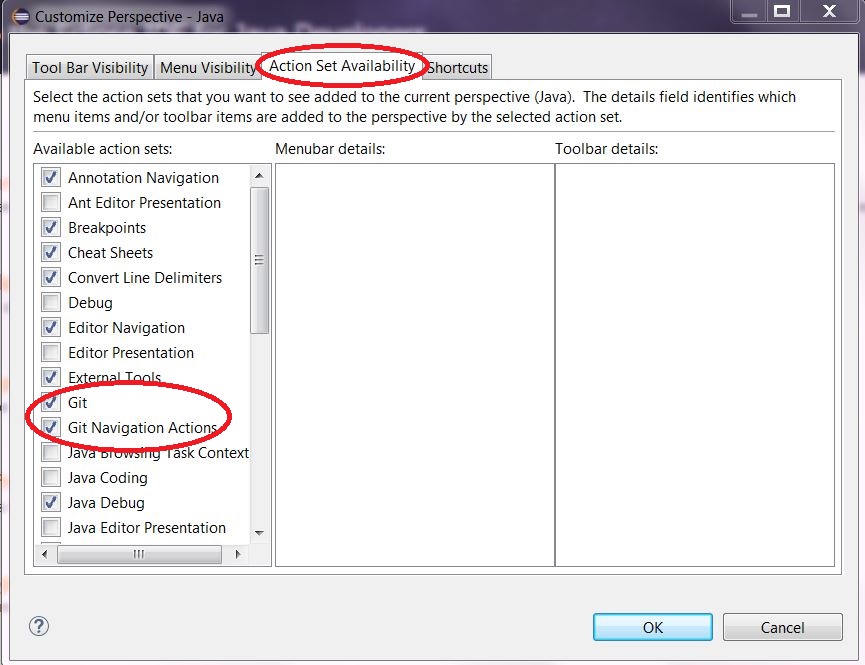
Select***Window ▸ Preferences ▸ Team ▸ Git ▸ Configuration***to see the current configuration and change email address and username.

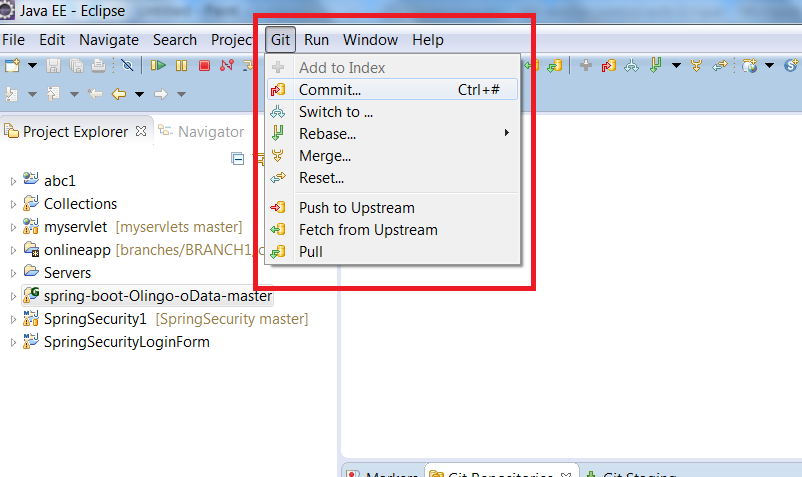


1. [**Configuring the toolbar and the menu for Git usage**](http://www.vogella.com/tutorials/EclipseGit/article.html#configuring-the-toolbar-and-the-menu-for-git-usage)

* To simplify access to the common Git operations you can activate the Git toolbar.

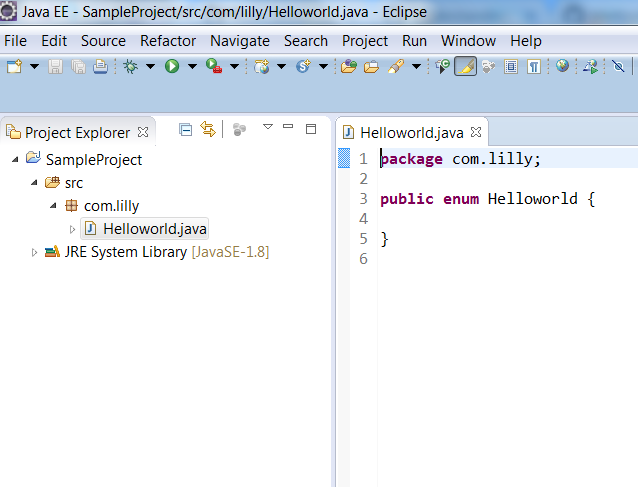
For this, select***Window ▸ Perspective ▸ Customize perspective*…​**and check the***Git* and *Git Navigation Actions* entries** in the***Action Set Availability*** tab.





1. [**Working with Eclipse projects in a Git repository**](http://www.vogella.com/tutorials/EclipseGit/article.html#working-with-eclipse-projects-in-a-git-repository)
   1. [**Adding a new project to a local Git repository**](http://www.vogella.com/tutorials/EclipseGit/article.html#adding-a-new-project-to-a-git-repository)**:**

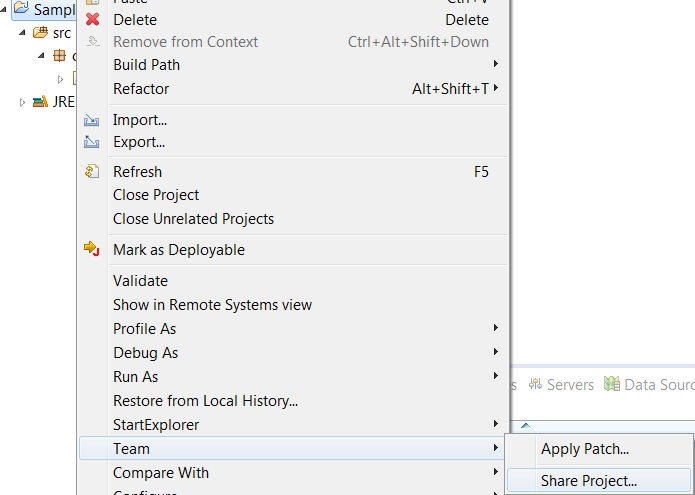
* Create sample project and move it to Git remote repository.



To move above project to GIT remote repository, we need to follow below steps.

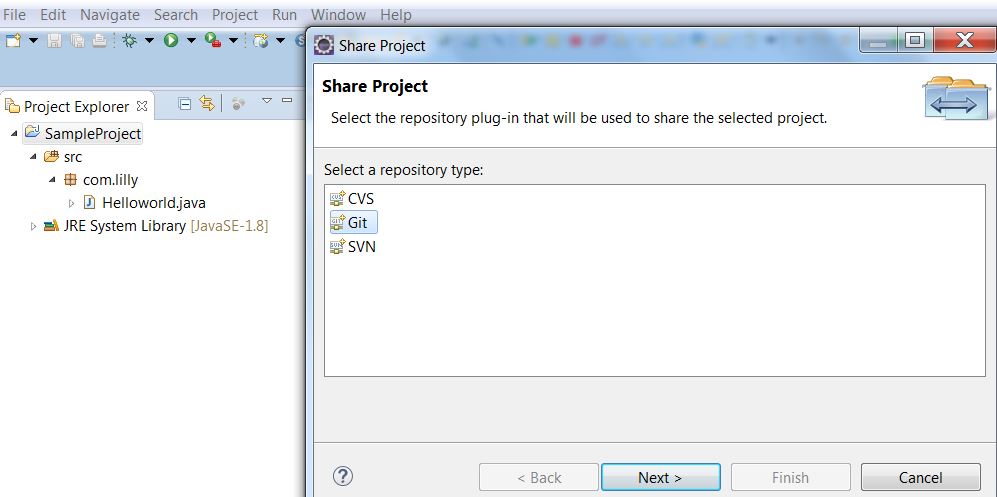
**Step 1:**

* Right click on ***project ▸ Team ▸ Share project***



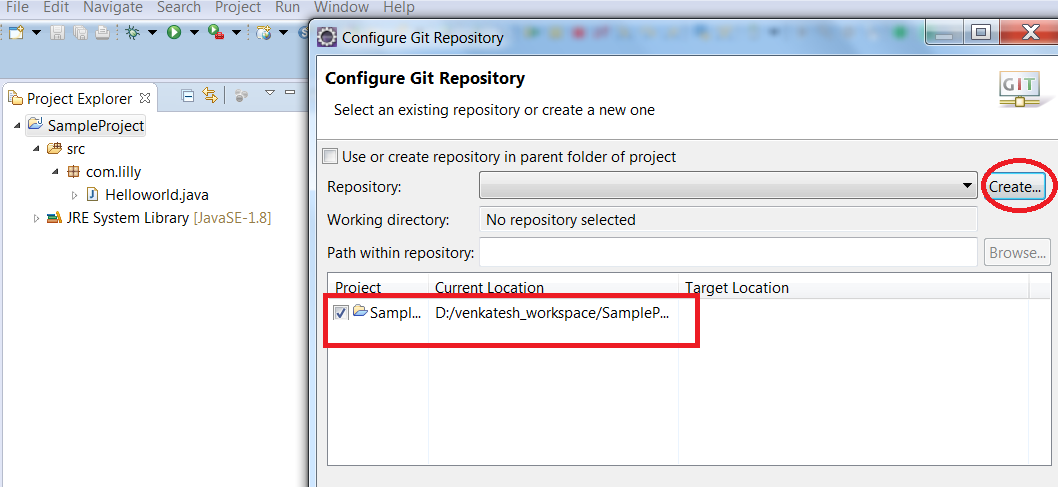
**Step 2:**

* Select ***GIT*** and click on ***Next button.***



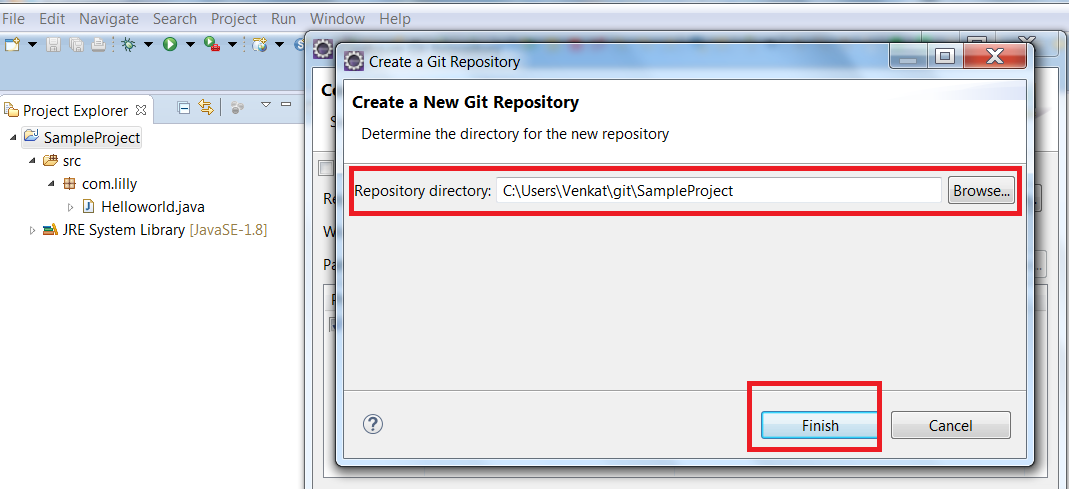
**Step 3:**

* Click on ***create button*** to provide local repository path.



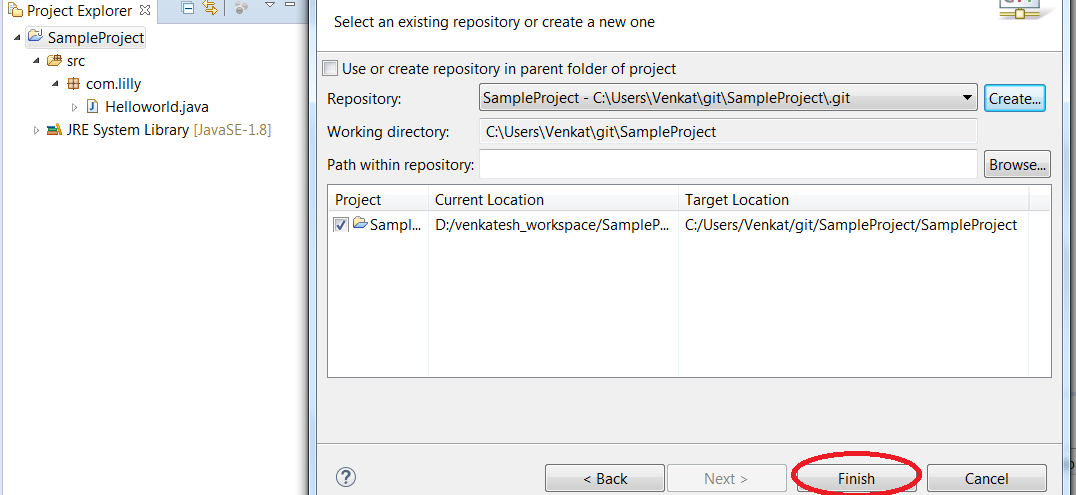
**Step 4:**

* Browse local repository location and click on ***finish***.
* Local repository can be any path in local system.



**Step 5:**

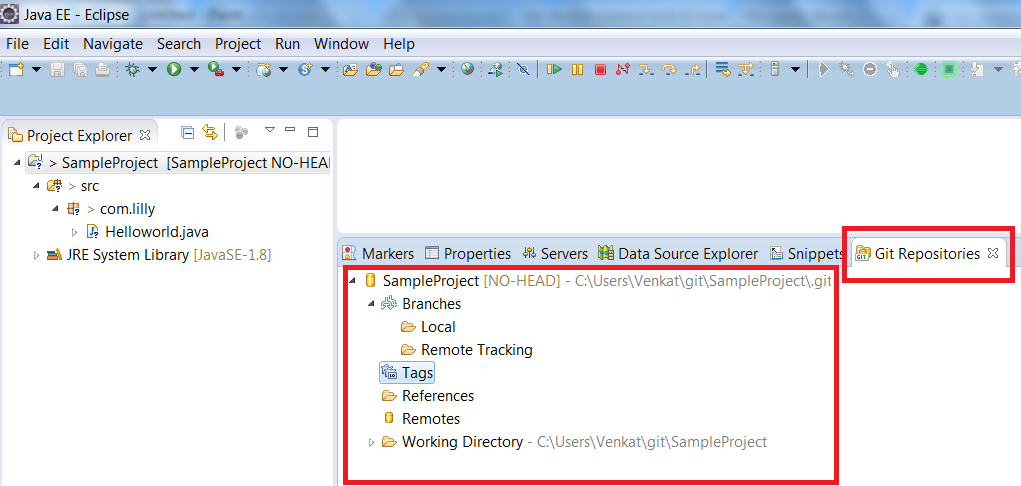
* Click on ***finish button*** to create local repository.



**Step 6:**

* Open ***Git repositories*** to view the project in local repository.

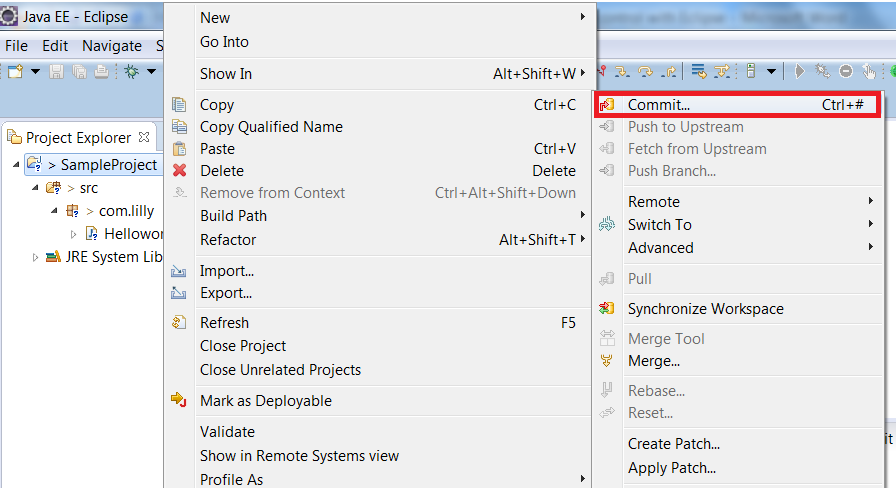
***Windows ▸ show view ▸ other... ▸ expand git ▸ select git repositories***



**Step 7:**

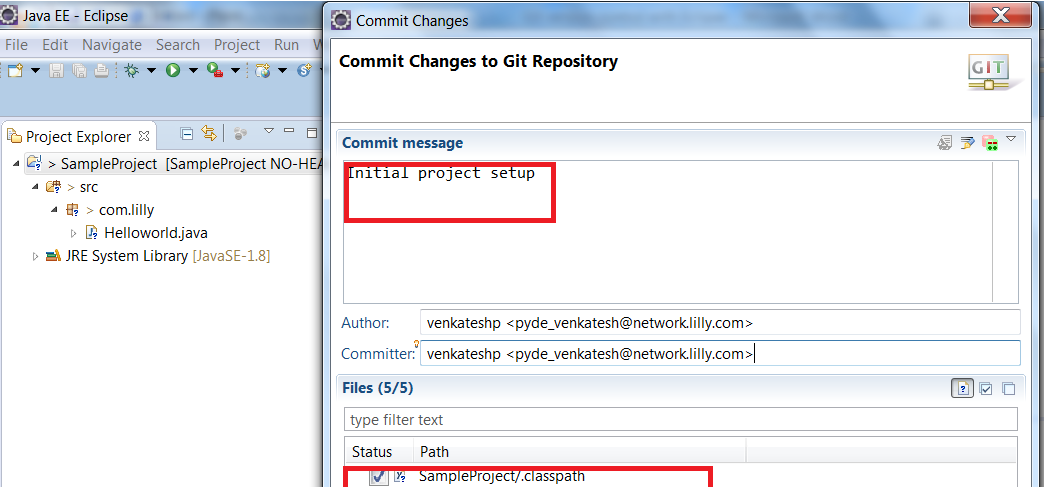
* Now, the complete project can be moved from eclipse to local repository through below step.

Right clickon ***Project ▸ Team ▸ Commit***



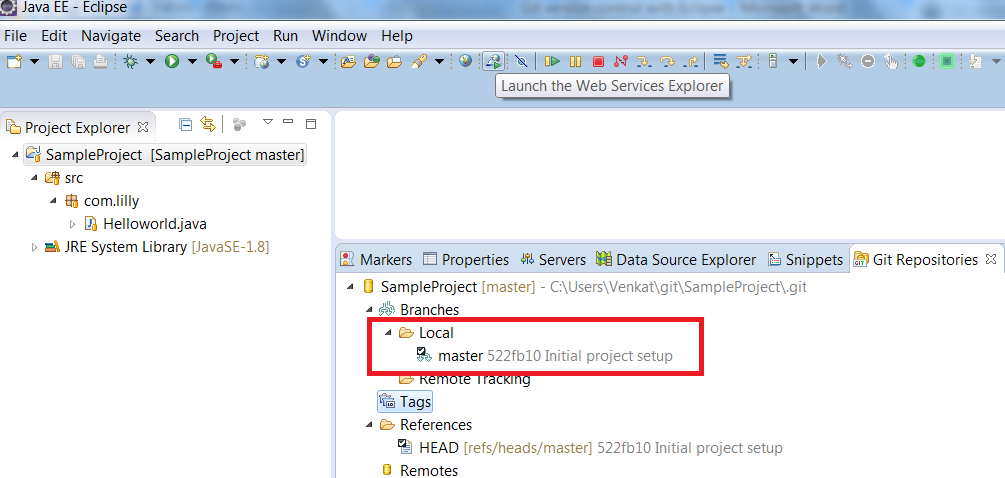
**Step 8:**

* ***Select all files***, enter ***appropriate comments***, and finally click on ***commit*** button.



**Step 9:**

* Now verify in Git repositories view for local repo project.

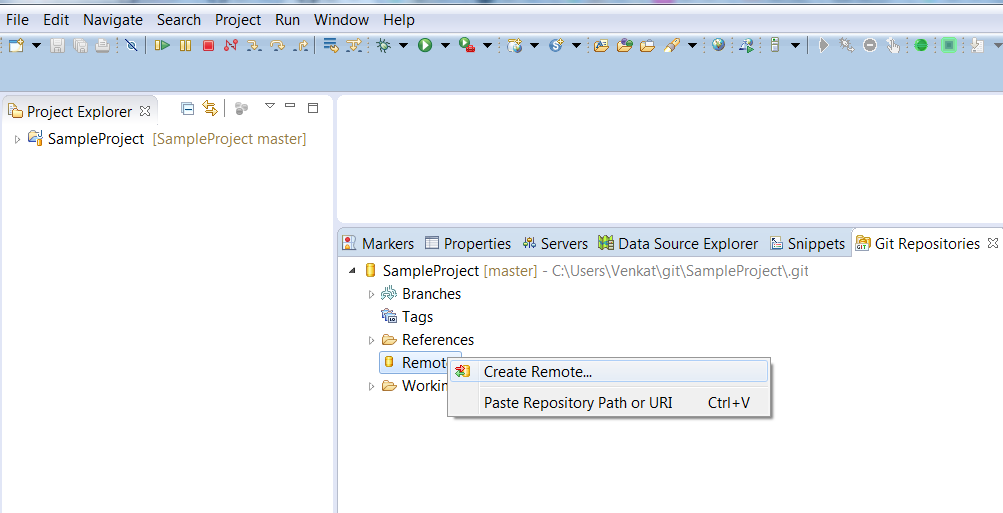


* 1. [**Adding a new project to a remote Git repository**](http://www.vogella.com/tutorials/EclipseGit/article.html#adding-a-new-project-to-a-git-repository)
* To add the above project into GIT remote repository, we need remote repository location. Here is my remote repository location URL.

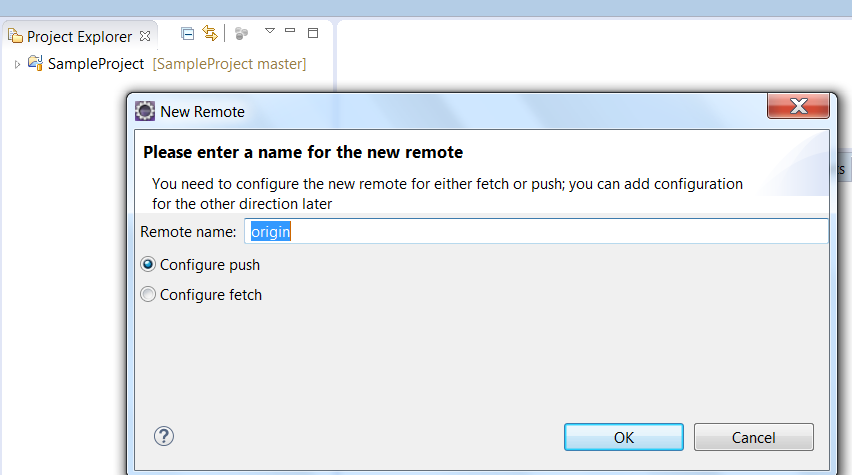
<https://github.com/venkateshpyde/SampleProject.git>

**Step 1:**

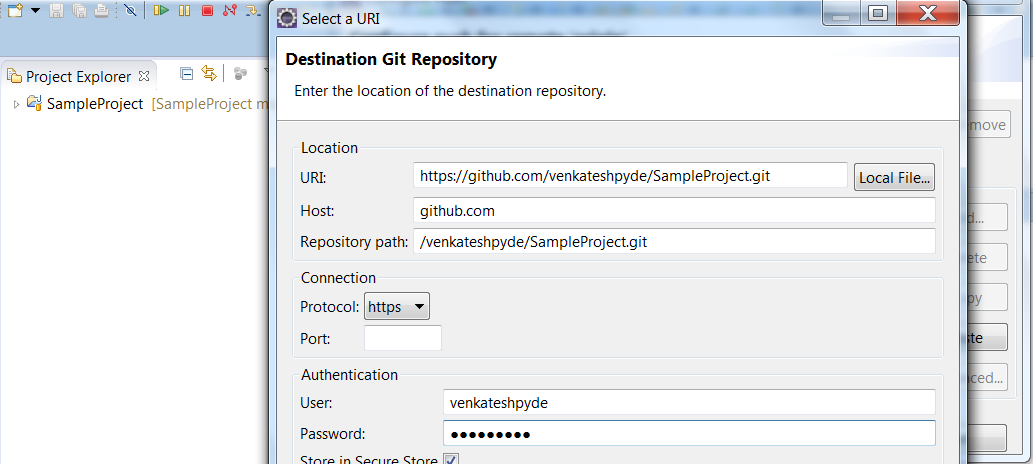
* Right click on ***remote*** option in ***Git Repositories*** view and select “***Create Remote”***



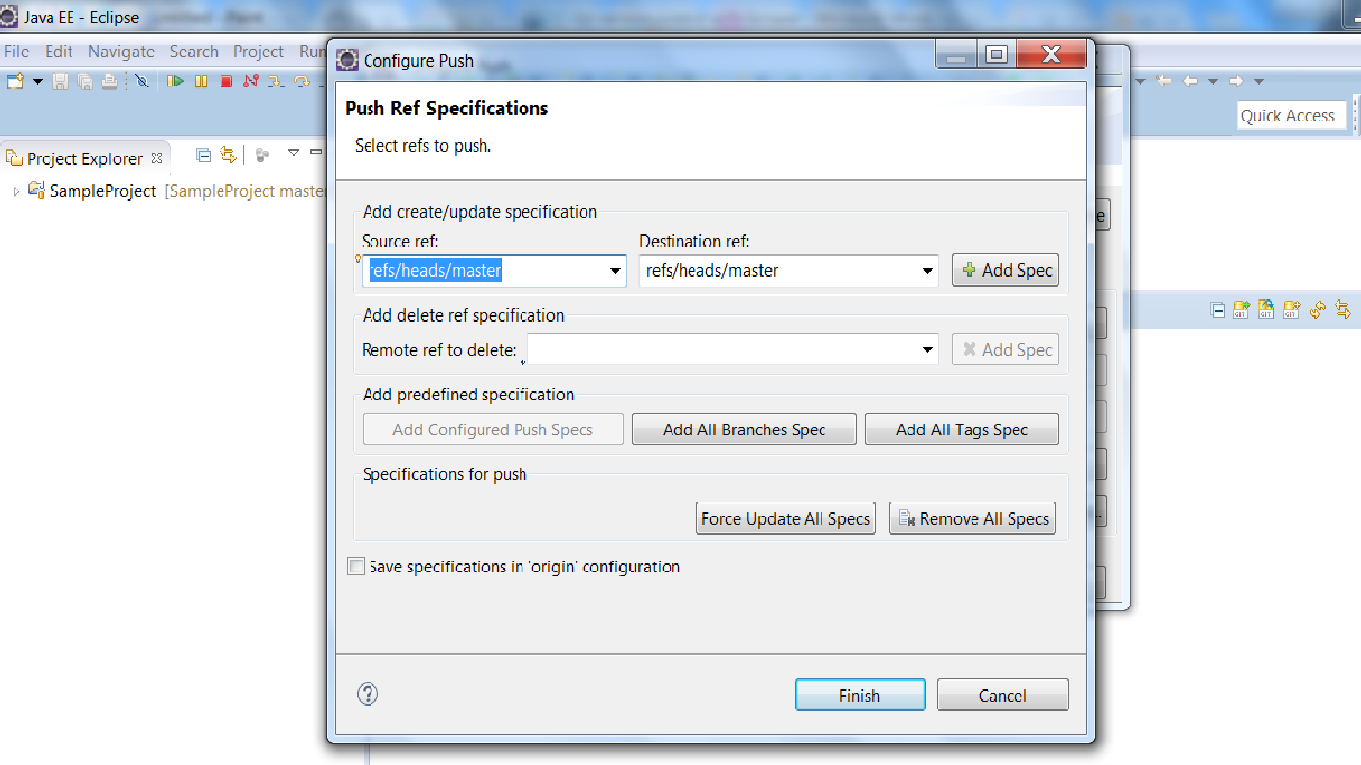
* Use default name as **origin** for remote repository and select “***Configure push***” radio button and click on ***OK***.



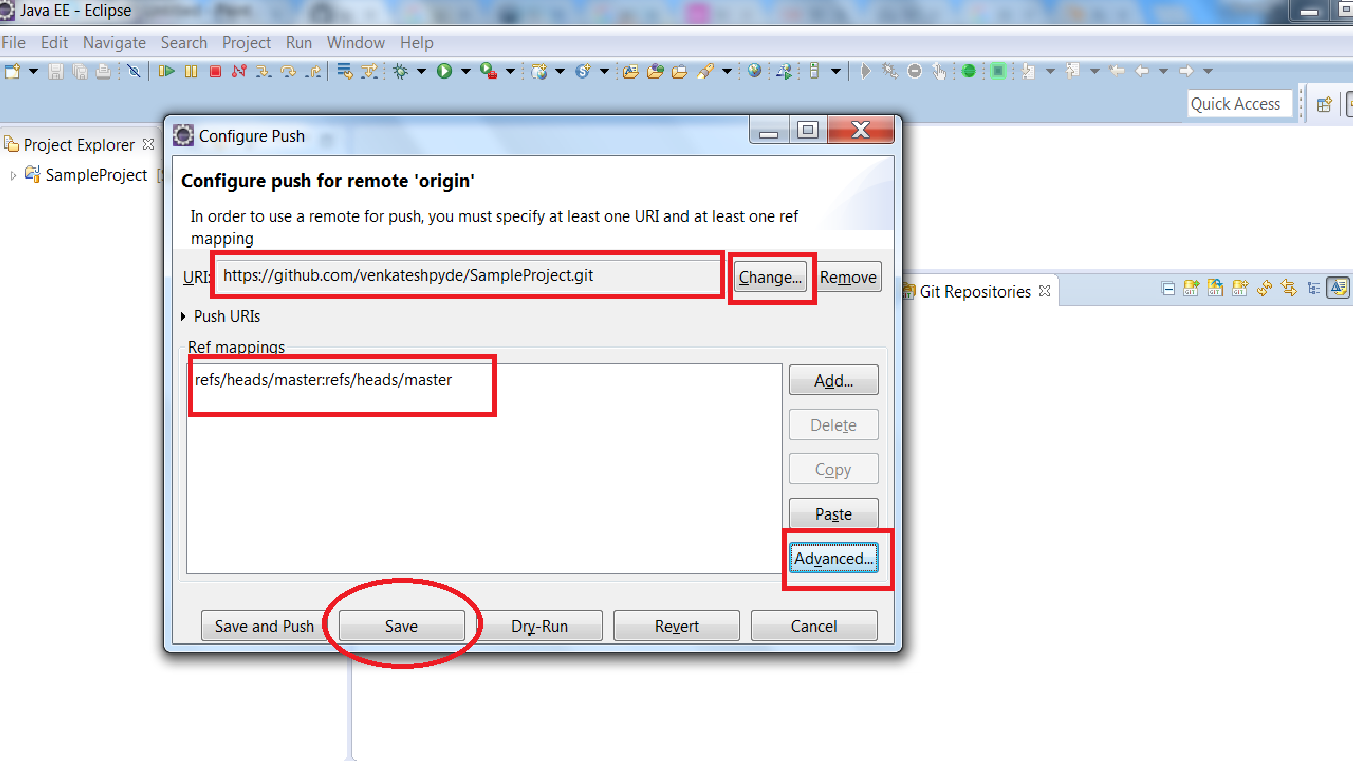
* Click on ***change button*** and provide ***GIT URL***, ***username and password***, ***enable Store in Secure store*** check box to save password in eclipse and click on finish button.



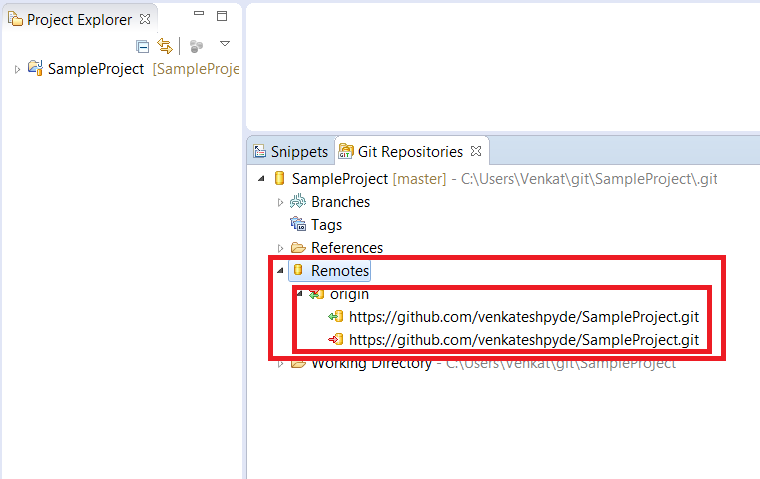
* Click on ***Advanced button***, **Select Source ref** as ***master [branch]*** and **Destination ref** as ***refs/heads/master*** and Click on ***Add Spec*** and finally click on ***finish*** button.



* Click on ***Save*** button to save configurations.



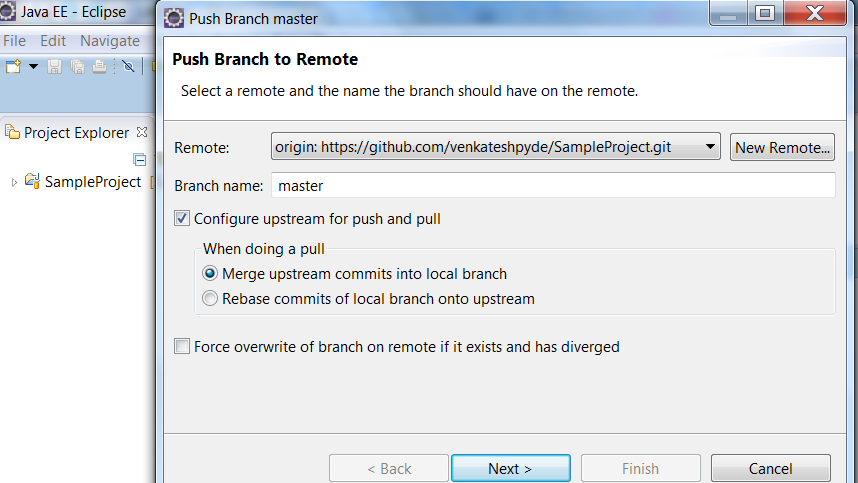
* Now check remote option in GIT repositories view for remote git repository location.



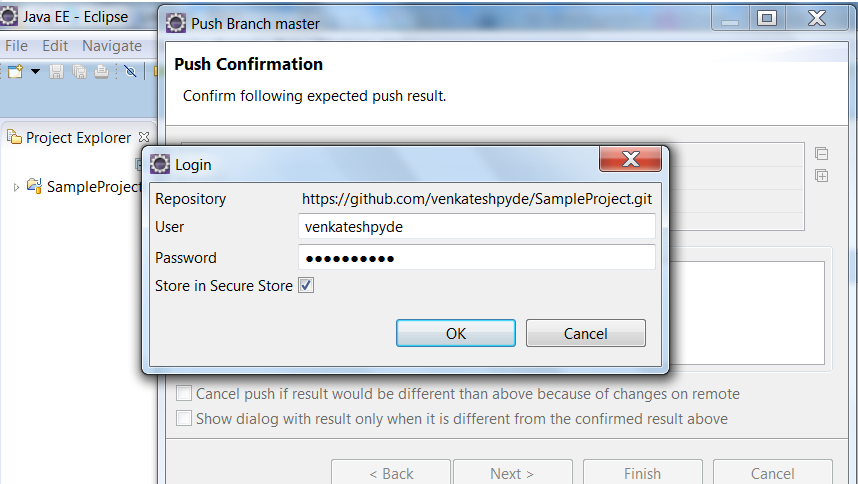
* Now push local repository project to remote repository.
* Right click on ***master*** under local and select ***Push branch***.



* Click on ***Next.***



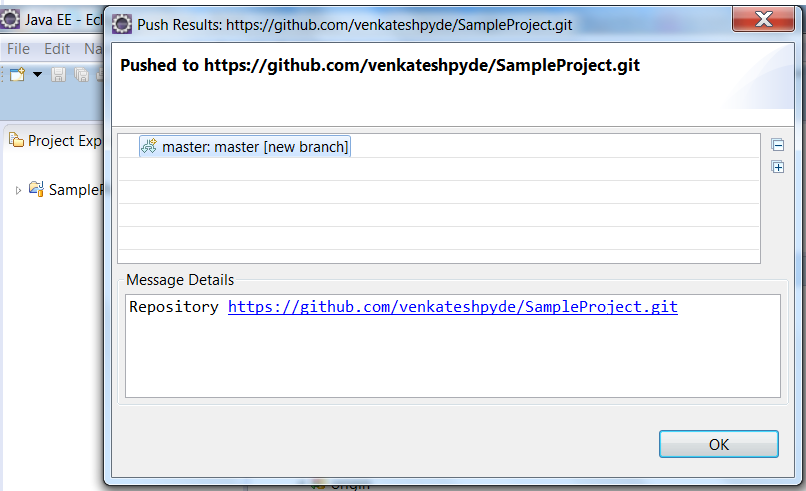
* Enter ***username and password***, enable ***Store in Secure store***, and click on ***Ok.***



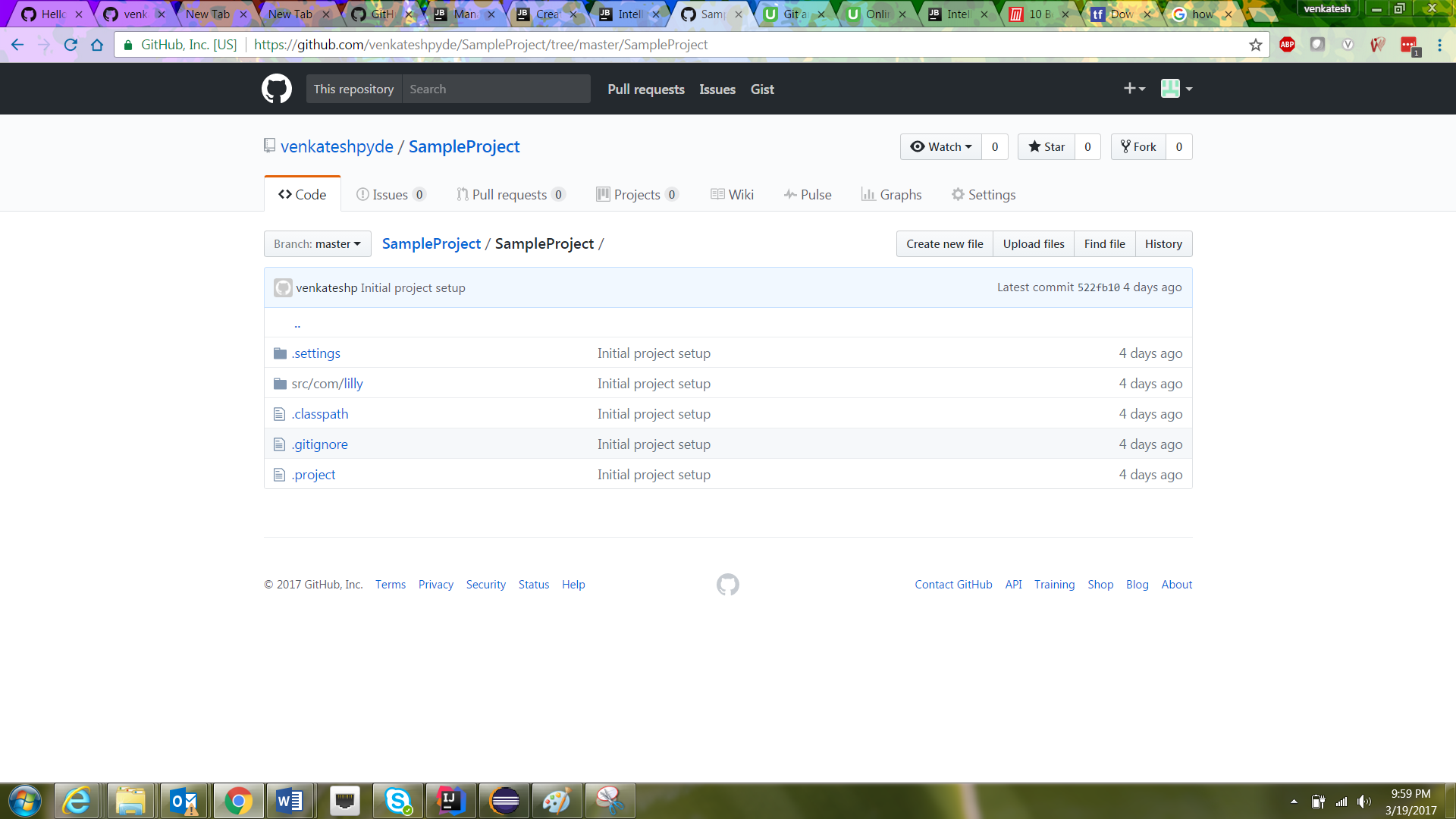
* Click on ***Finish***.



* Click on ***OK.***



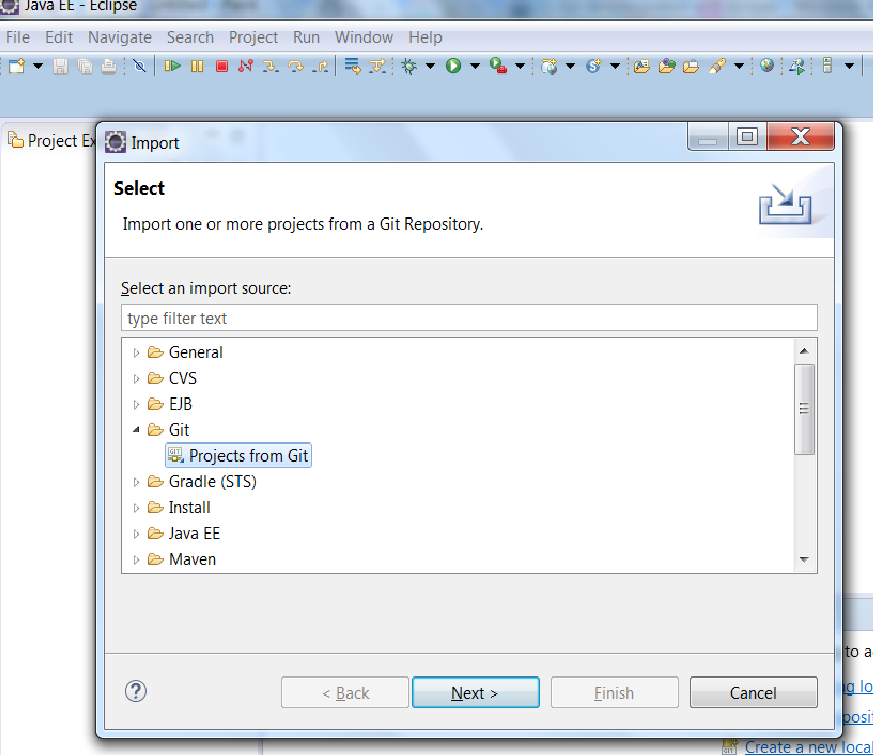
* Now verify your project in GIT remote repository.



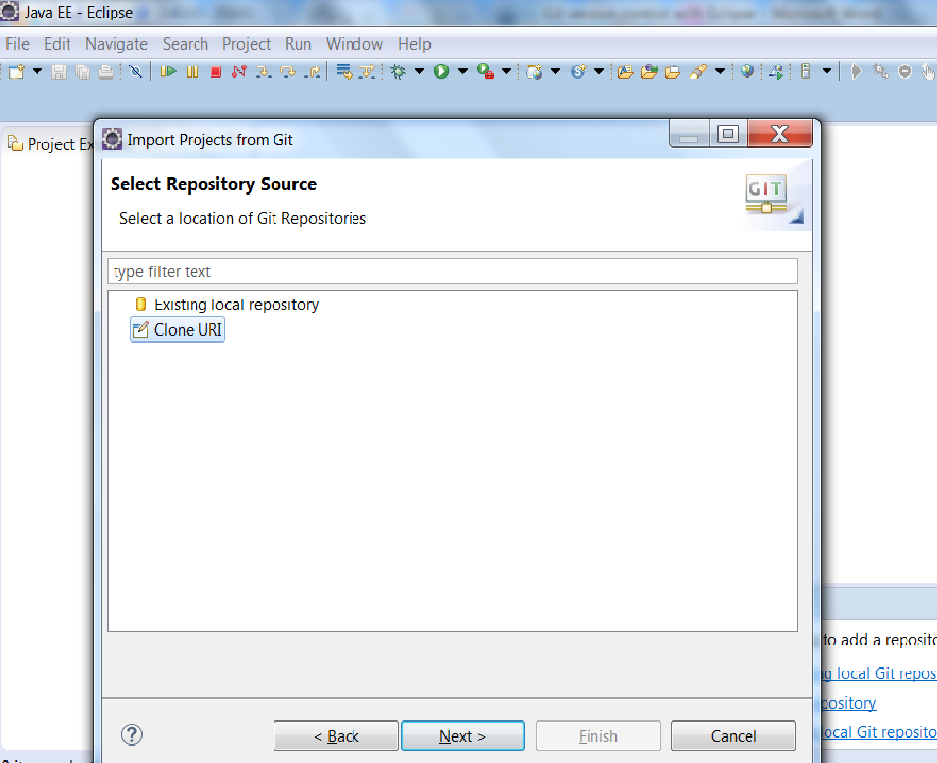
[**5. Import (clone) projects from a remote GIT repository**](http://www.vogella.com/tutorials/EclipseGit/article.html#import-projects-from-an-existing-repository)

* Before cloning or importing a project from remote repository, we need project ***GIT remote repository URL and GIT username and password.***
* Follow below steps to import projects from GIT remote repository.

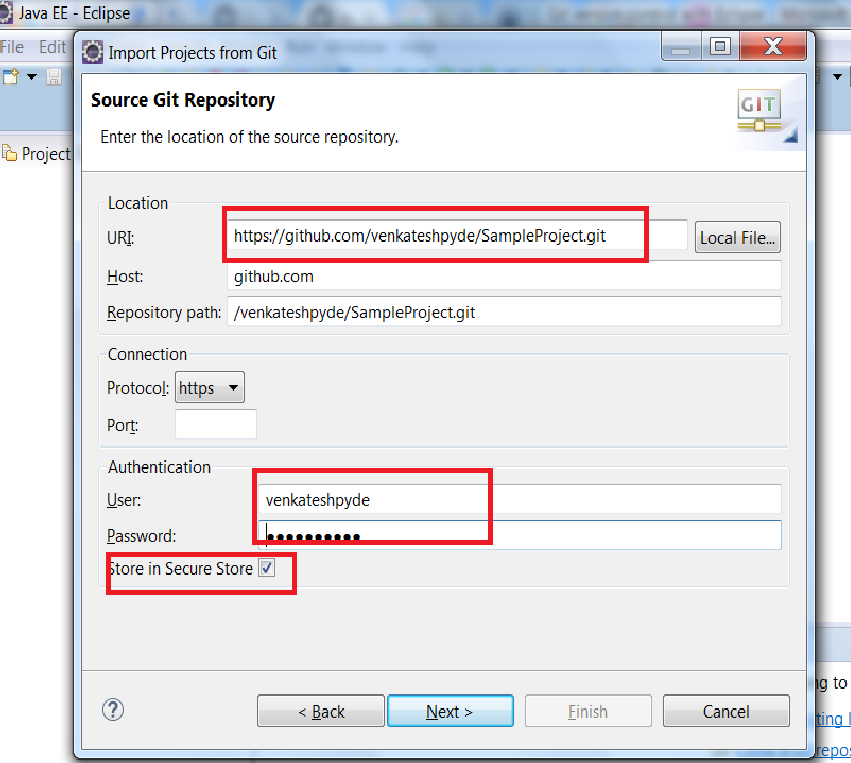
Go to ***File ▸ import ▸ expand git ▸ select Projects from Git ▸ Click on Next.***



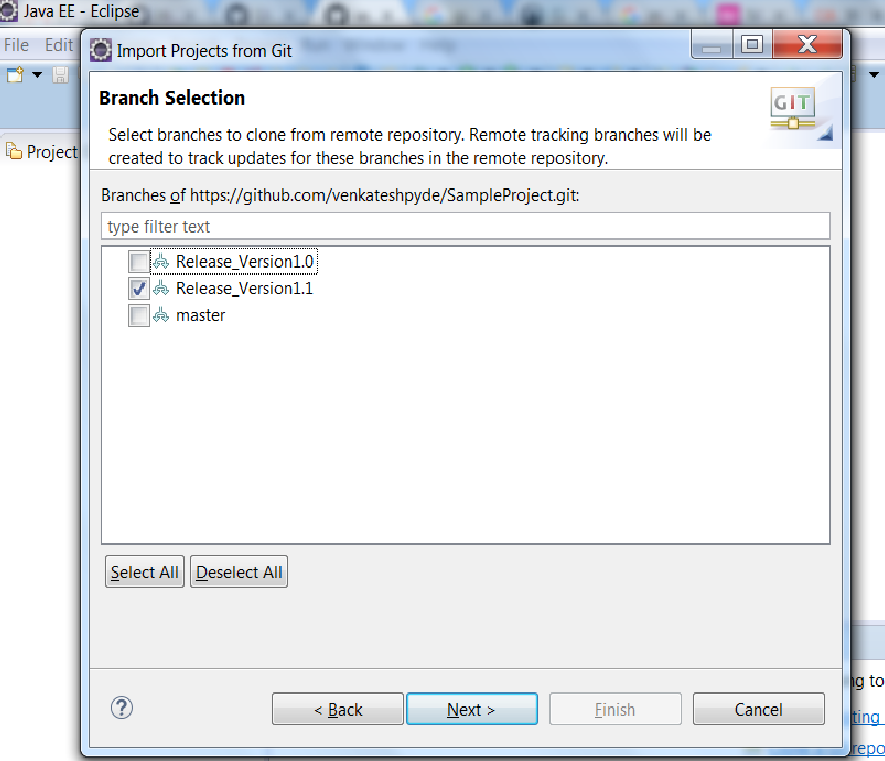
* Select ***Clone URI*** and click on ***Next button.***



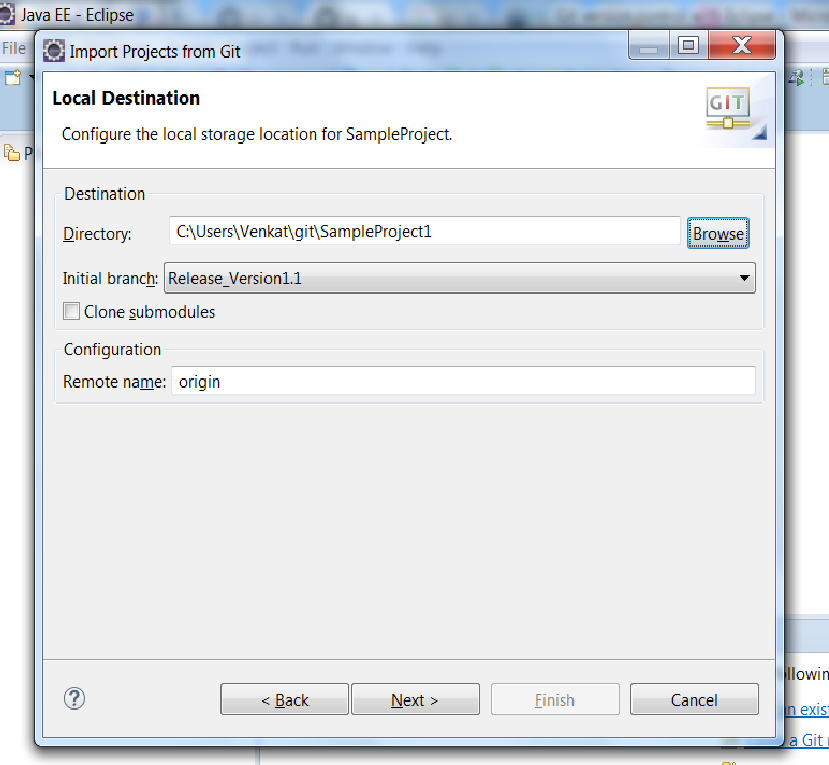
* Enter project GIT remote repository ***URL*** and enter ***GIT username and password.*** Make sure ***enable Store in Secure store*** and click on ***Finish.***

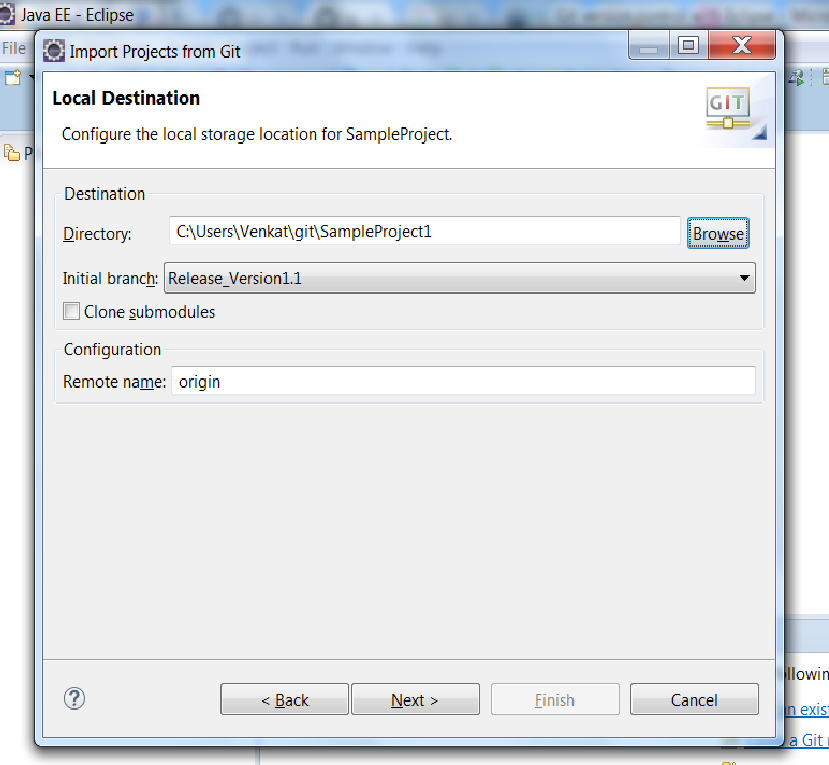


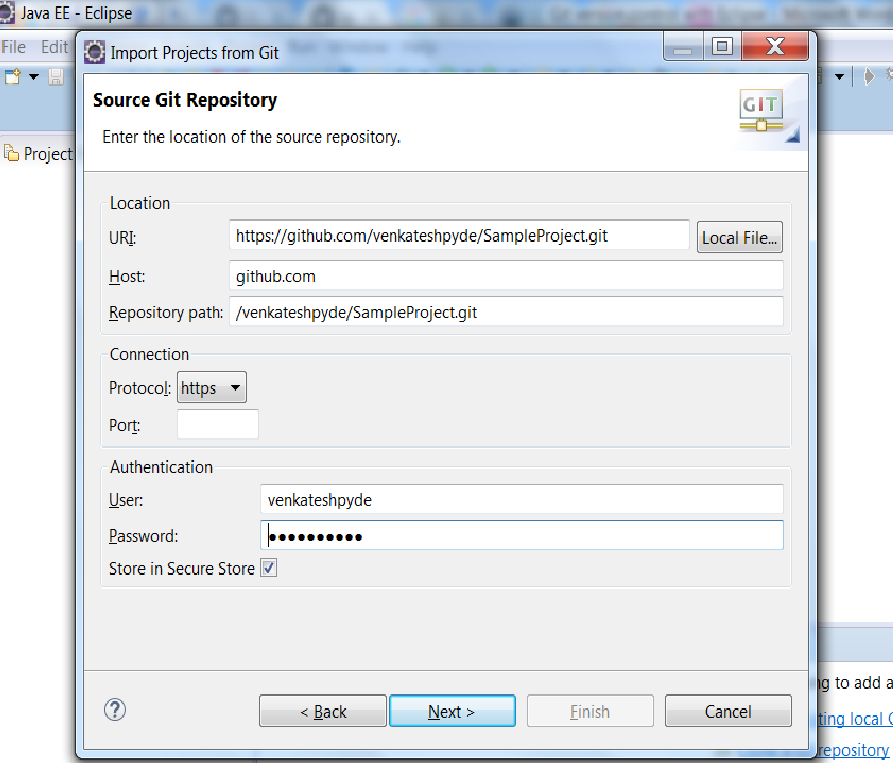
* Select ***appropriate branch*** and click on ***Next***.



* Click on ***Browse button*** to choose local repository project location and click on ***Next ▸ Next▸ Finish***







1. Push local changes into remote repository

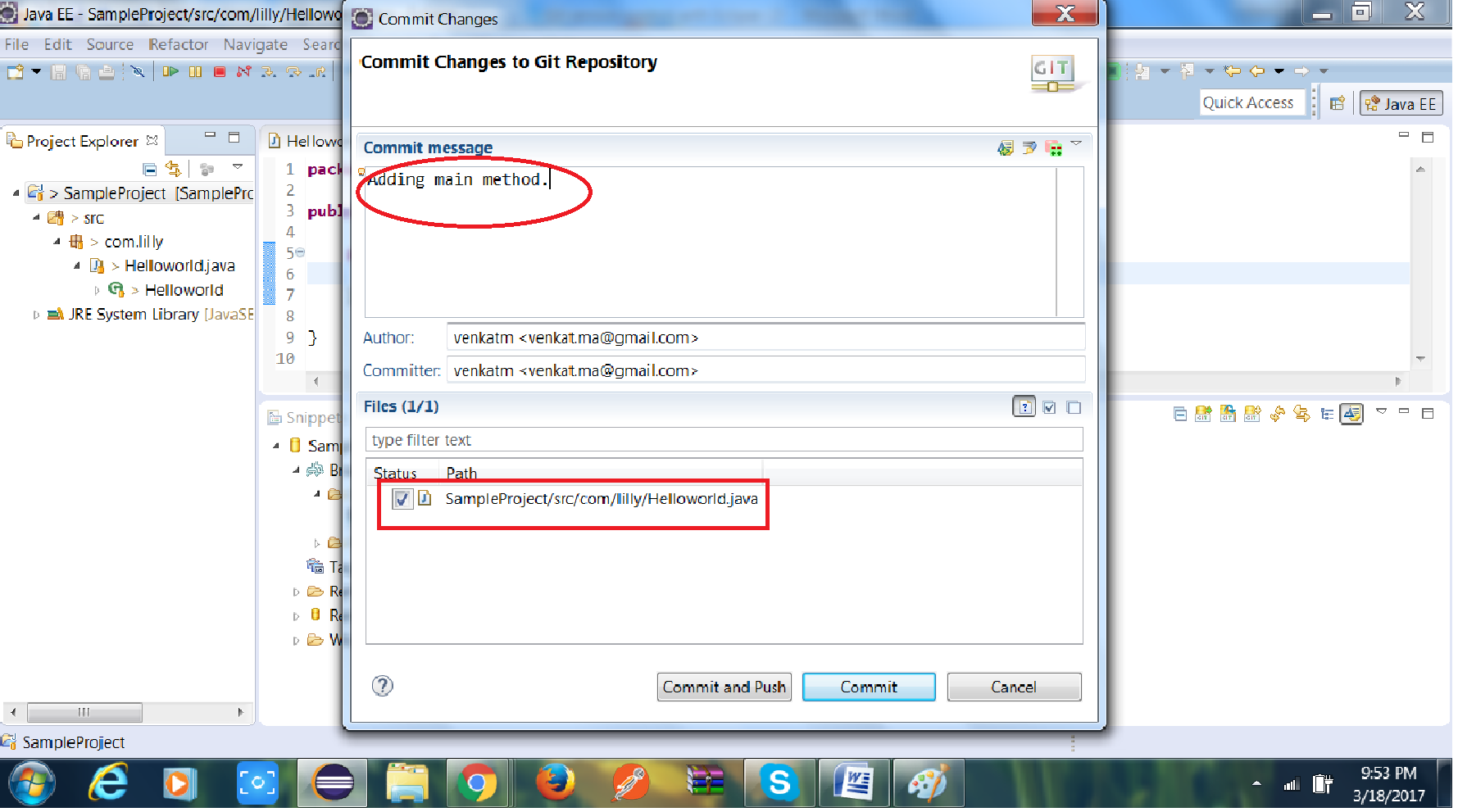
To push changes into remote repository, first we need to commit changes into local repository and then push changes to remote repository.

To commit changes into local repository, we need follow below steps.

***Right click on project -> Team -> Commit***



Add appropriate comments and make sure that files selected. And click on commit button.



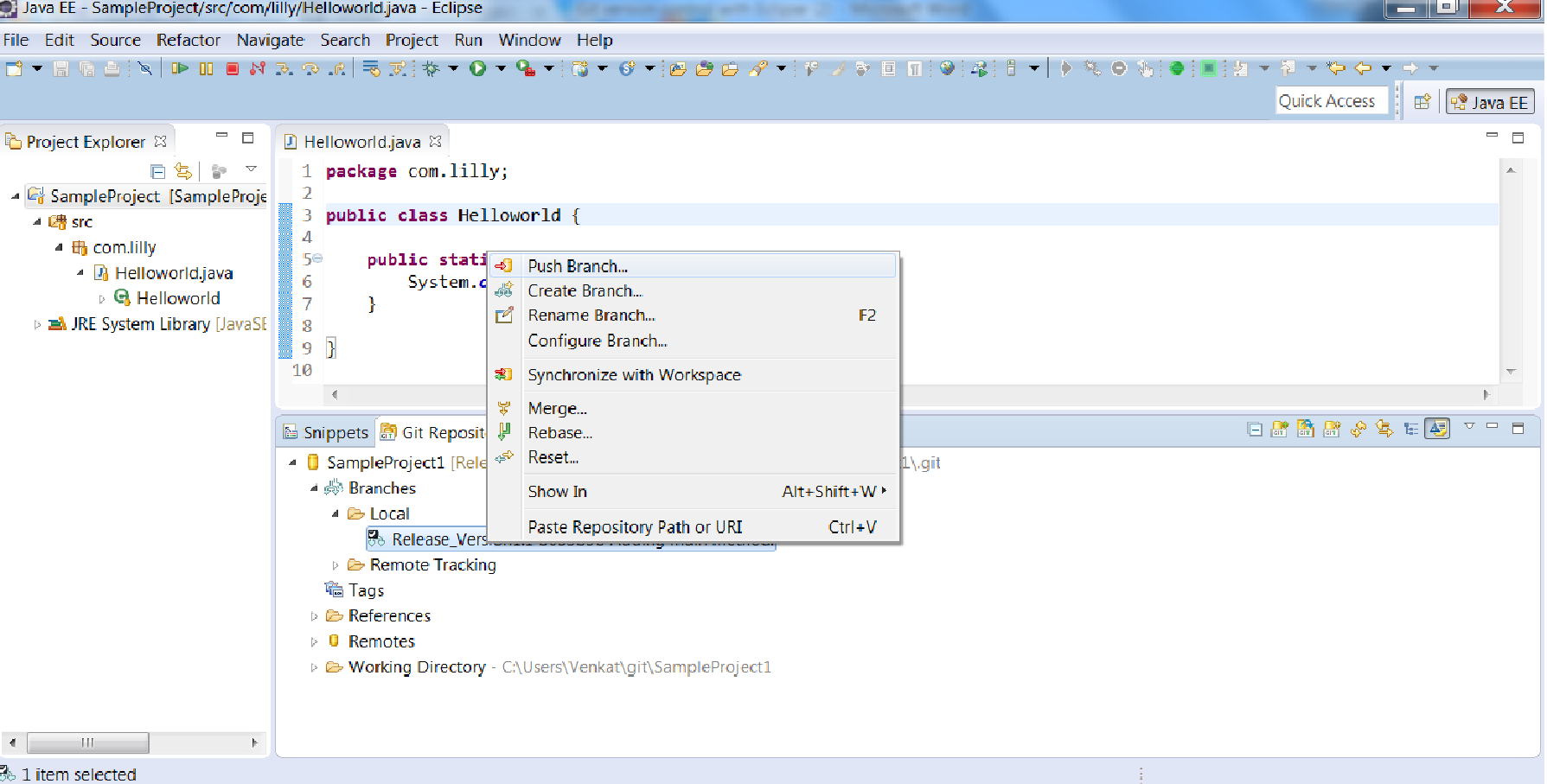
Now above changes are committed into local repository, And these are eligible to move

Repository.

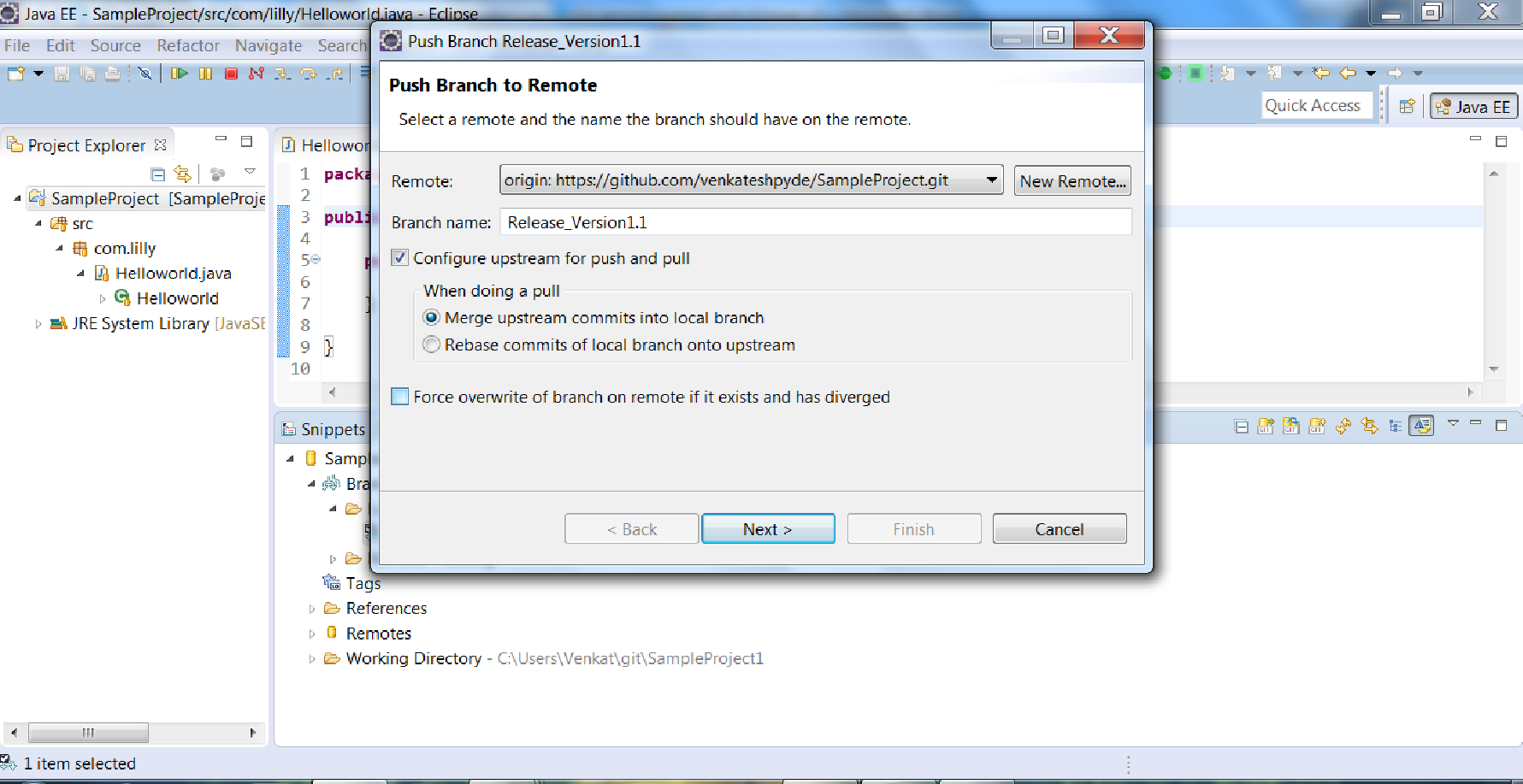
To move committed changes into remote repository we need to follow below steps.

Go to remote repositories vies and follow below steps.

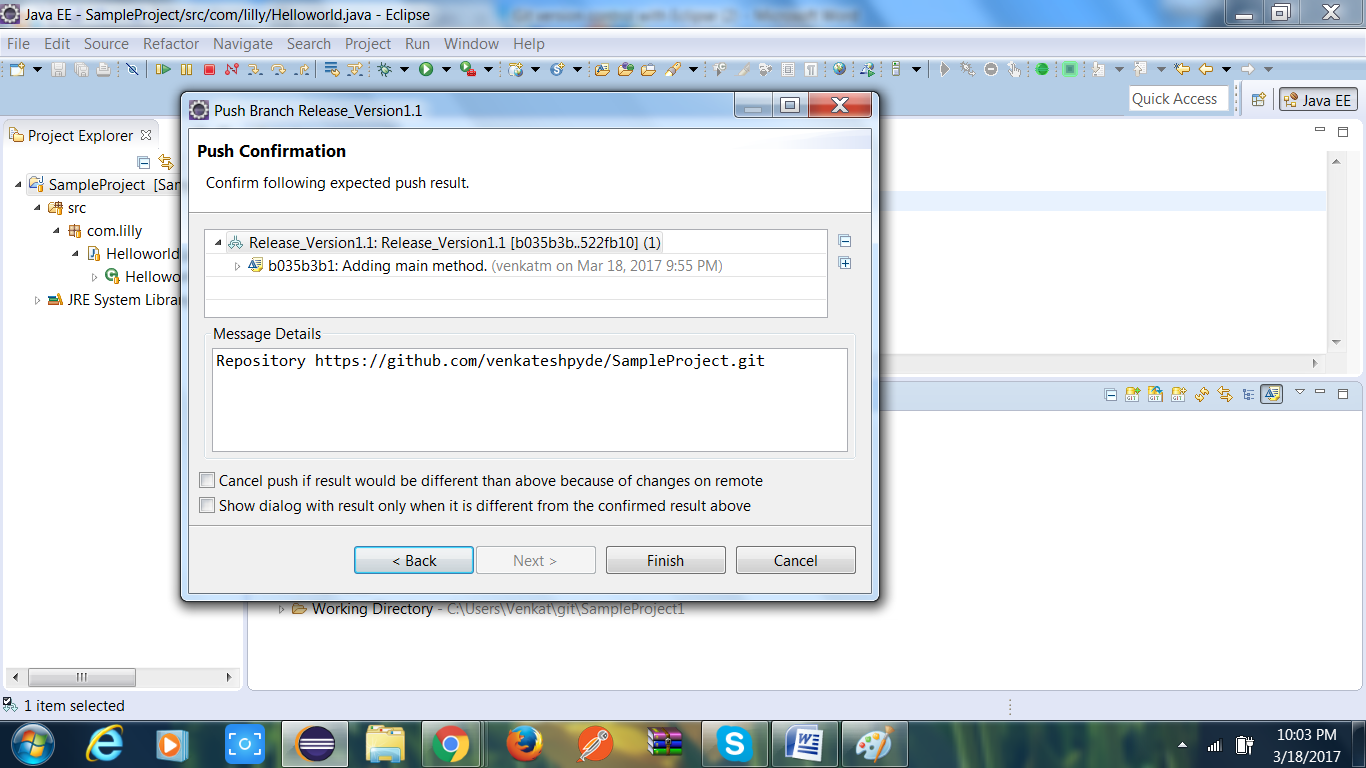
***Right click on branch under local folder -> select Push Branch***



Click on Next button



Click on finish and ok button.



1. Pull latest changes from remote repository.

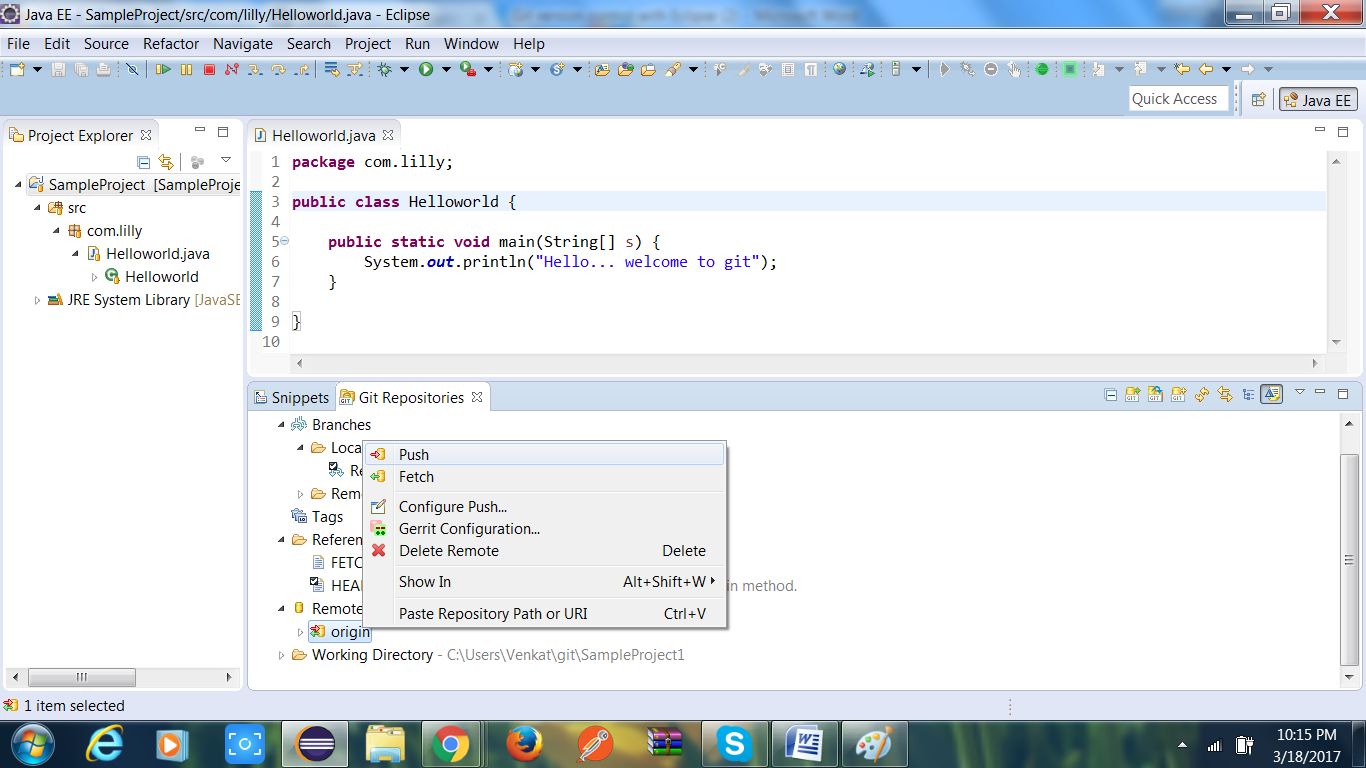
To get remote repository latest changes into local we need to follow blow steps.

1. First fetch remote changes into local repository
2. And then pull local repository changes into our workspace.

To fetch remote repository changes into local repository we need to follow below steps.

***Go to remote repository view -> Right click on origin(under remote) -> select Fetch and***

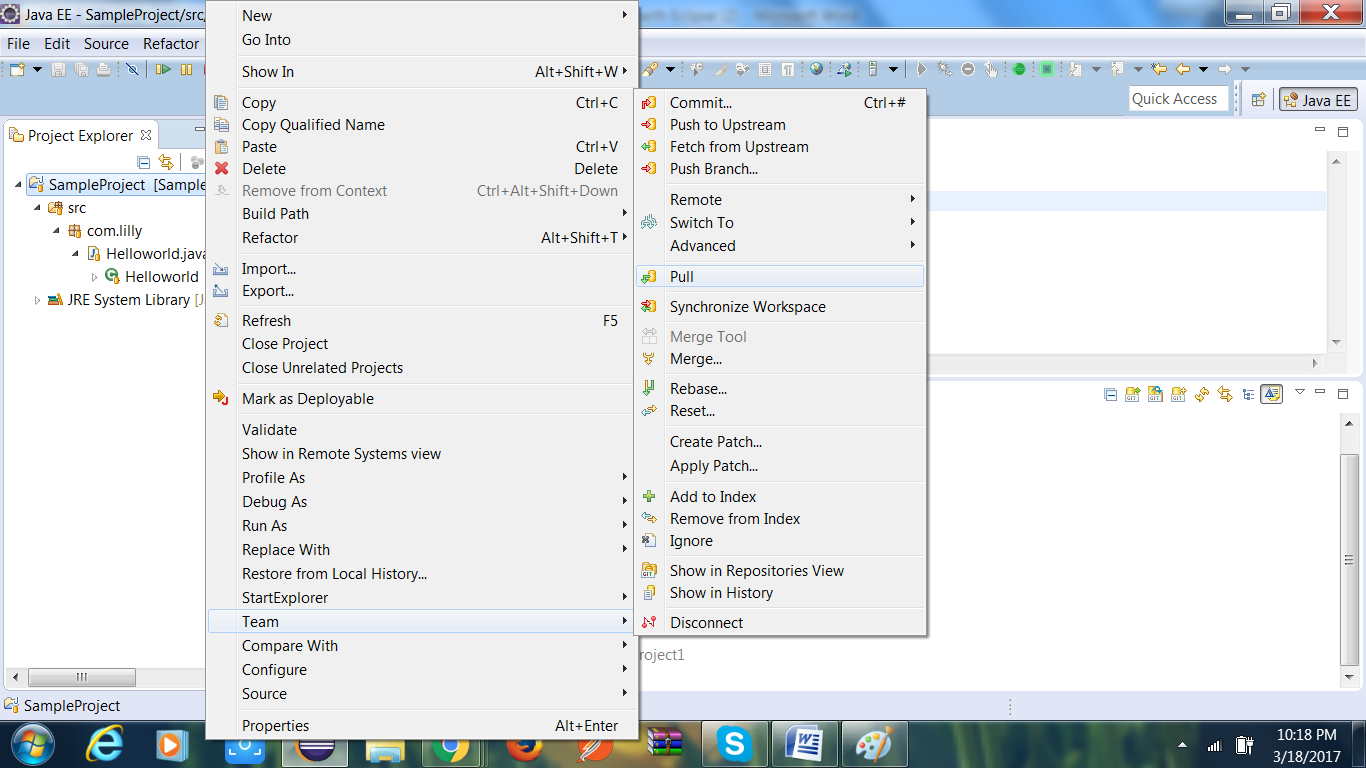
***Click on OK.***

******

Now all remote latest changes in our local repository and pull local repository changes

Into our local workspace we need to follow below steps.

***Right click on project -> Team -> Pull.***

******

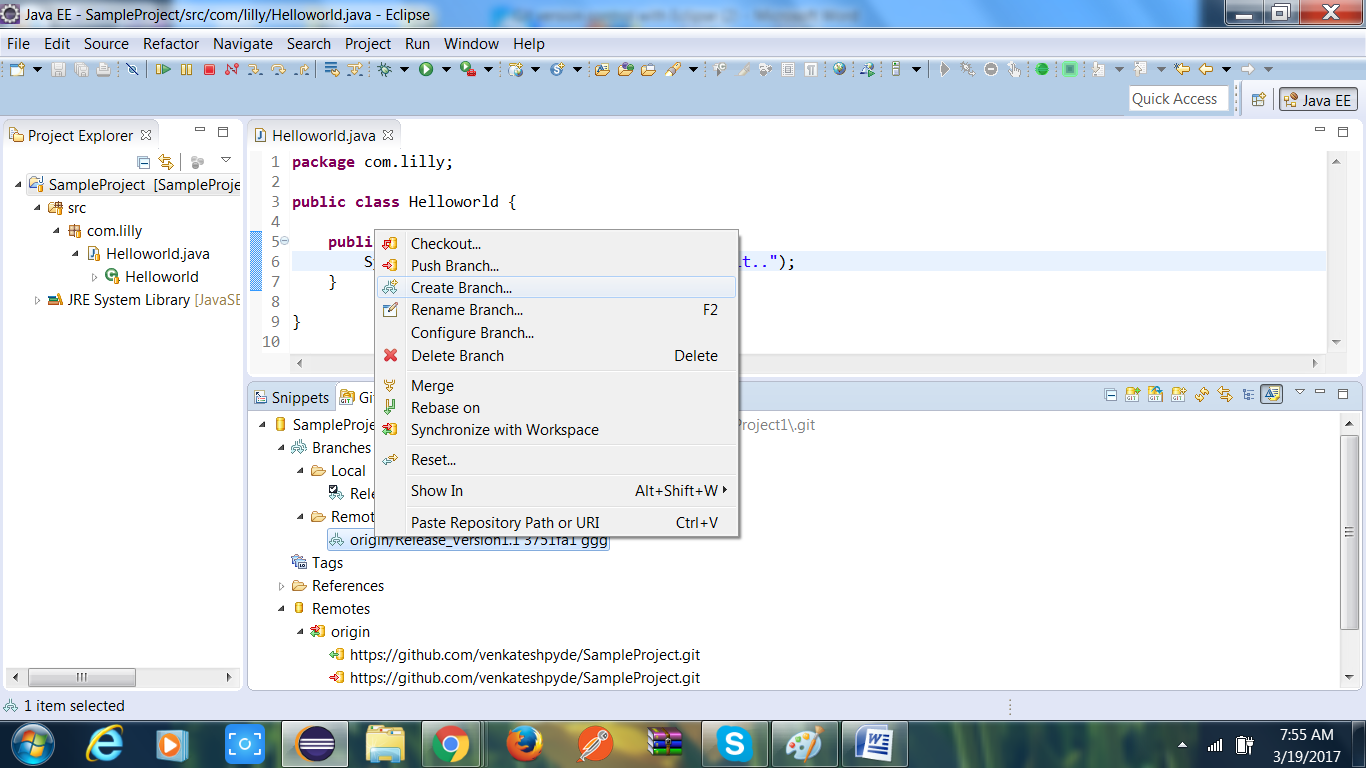
1. Creating another local branch

Sometimes we have to work on 2 different tasks and we can create 2 different local branches to work on parallel 2 different tasks.

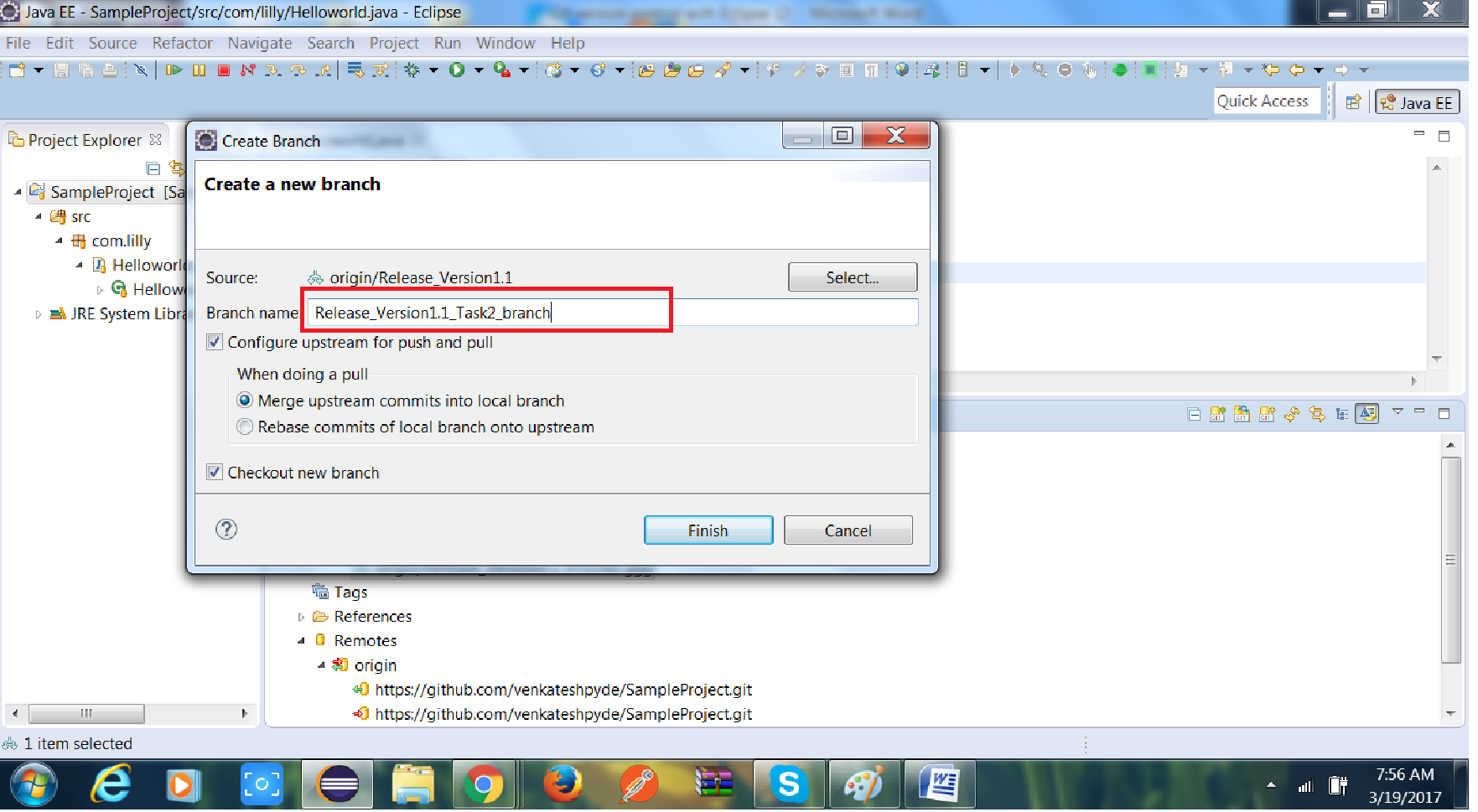
These 2 local branches can point to same remote repository.

To create another local branch from same remote repository we need to follow below steps.

***Go to remote repository view -> expand remote tracking under branches -> right click on Origin/branch or master -> create branch.***



Enter new branch name and click on finish.



1. Push changes into remote repository from another local branch.

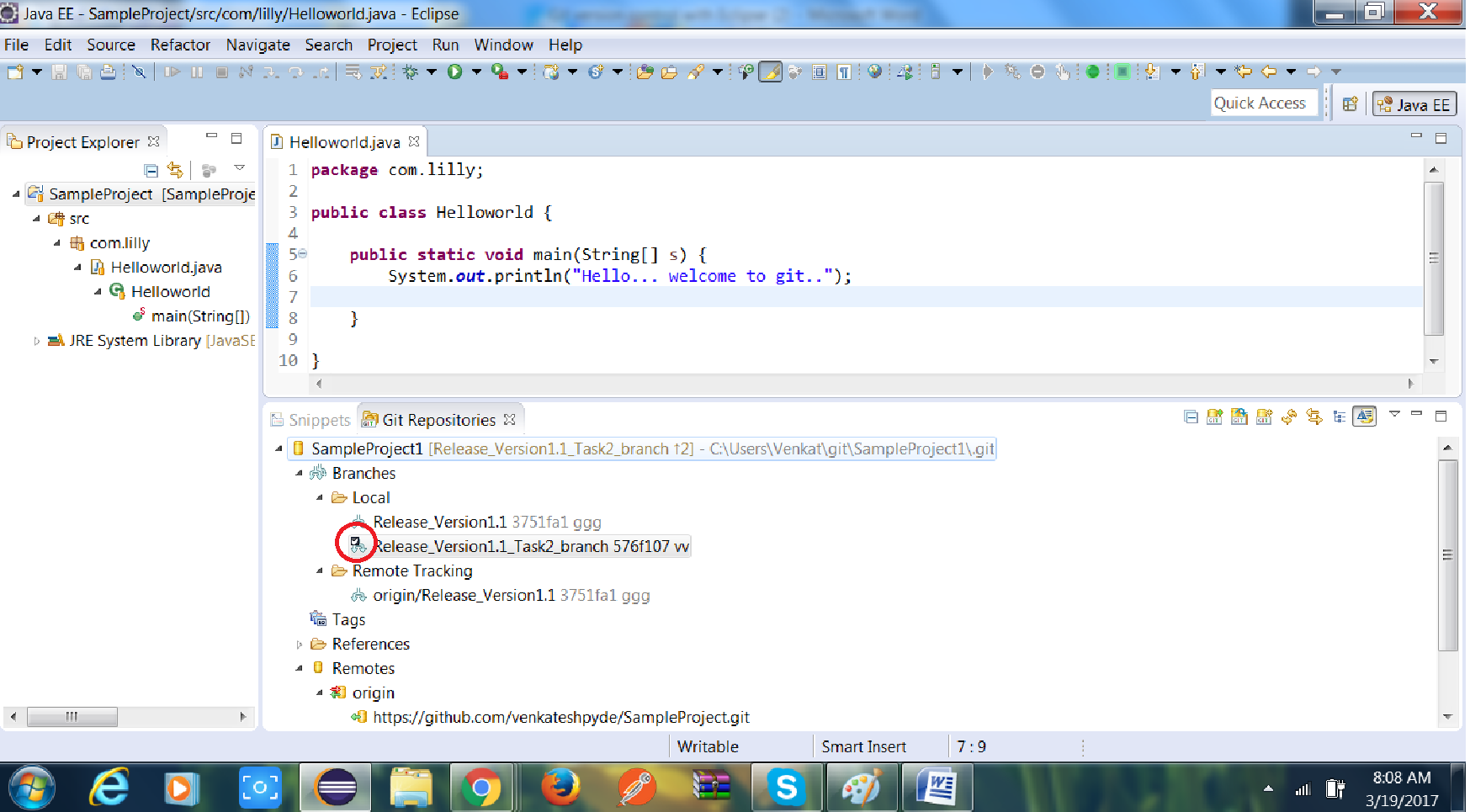
To push new/another branch changes to remote repository we need to follow below steps.

1. First commit local workspace changes into new/another branch.
2. Push new/another branch changes into remote repository.

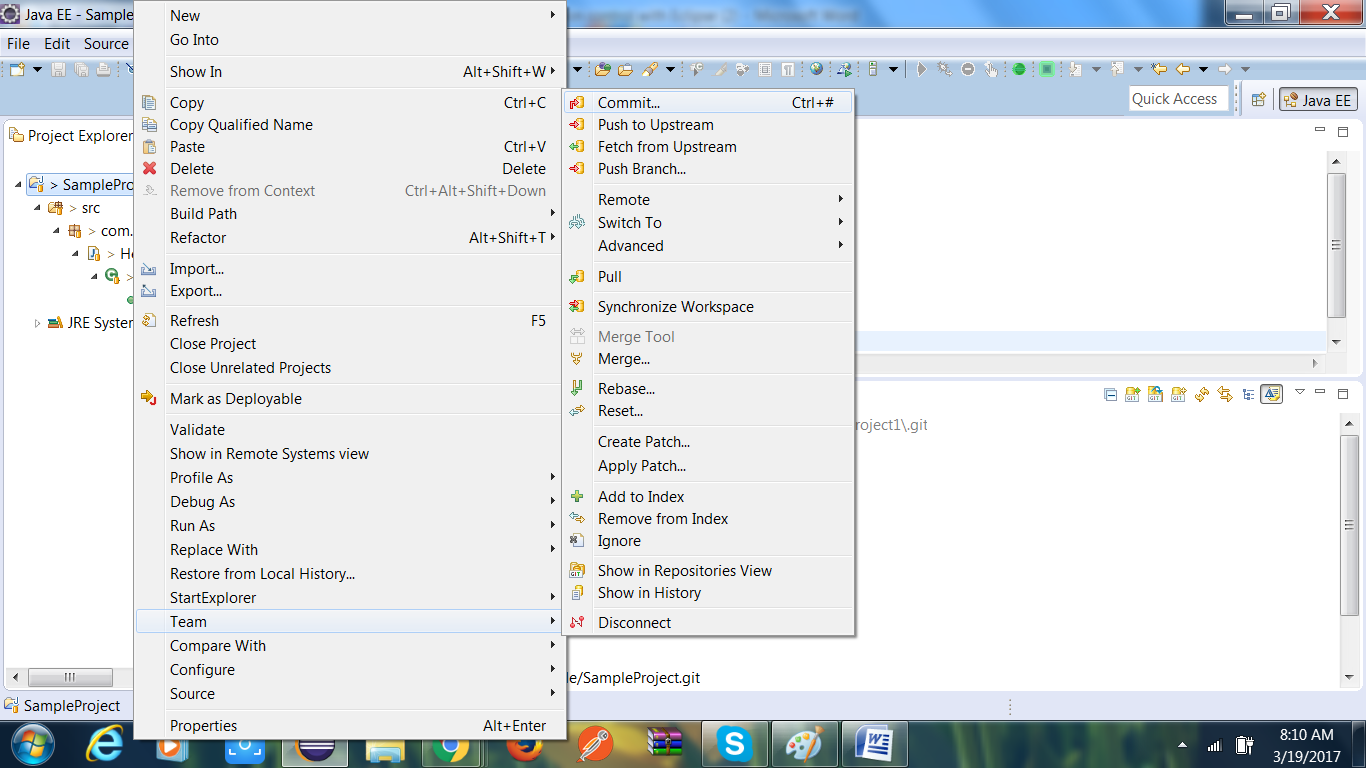
To commit local workspace changes into new/another branch, we need to

Follow below steps.

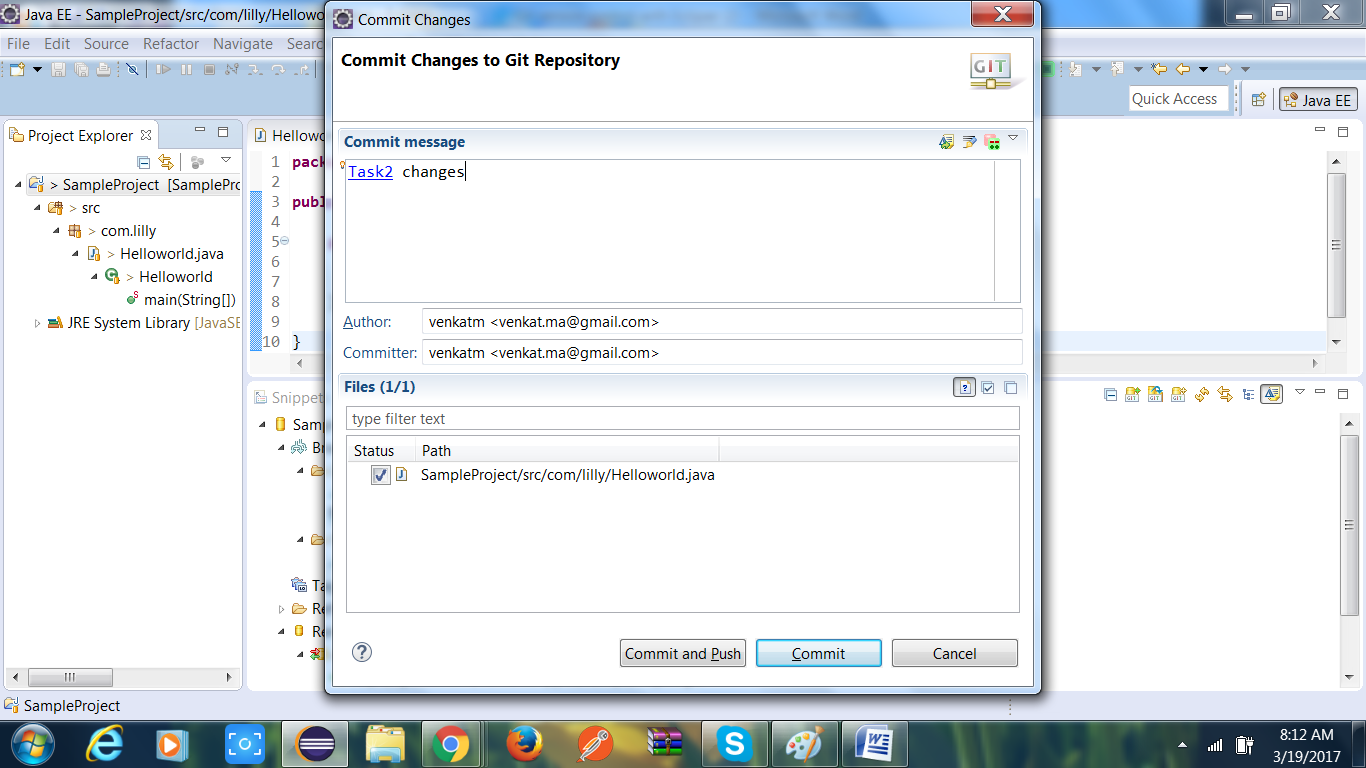
Before commit changes, make sure cursor should pointed to new/another branch



Right click on project -> Team -> Commit.

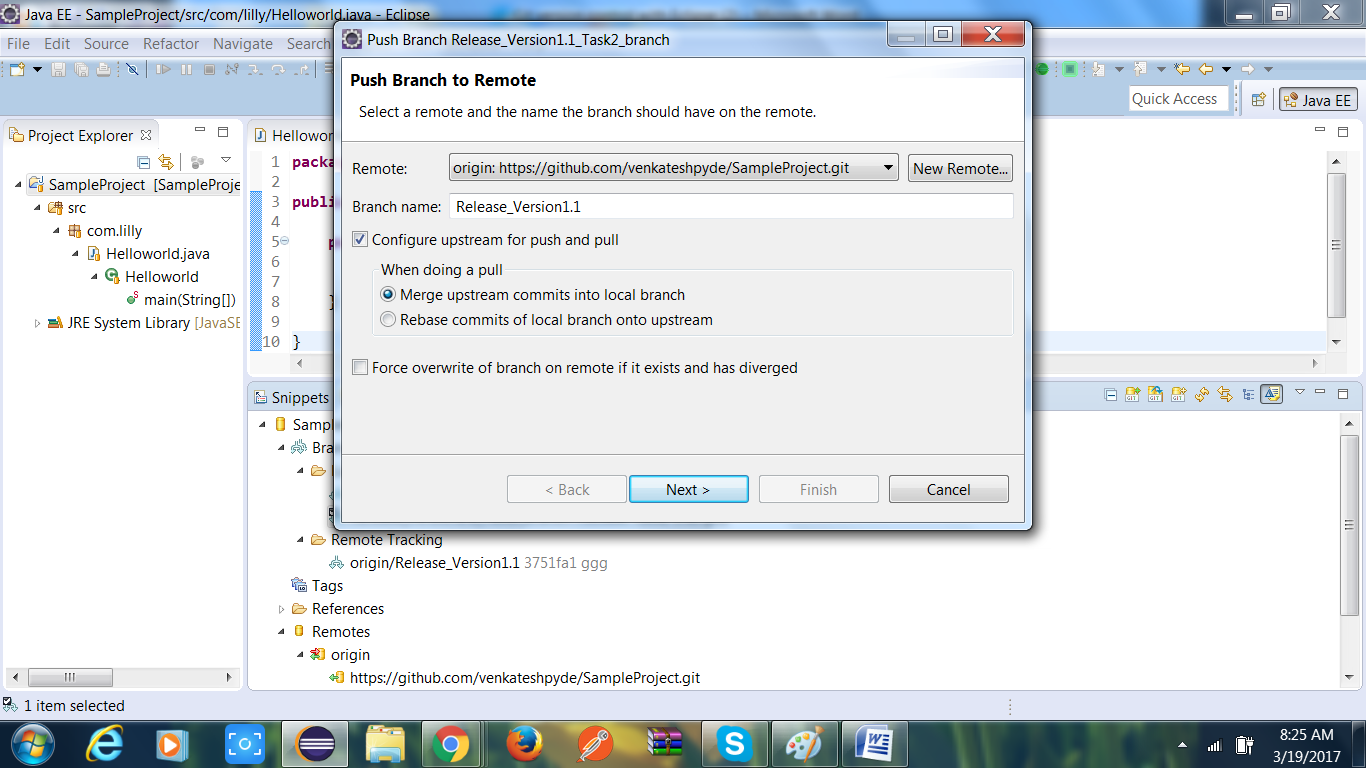
******

Add appropriate comment and click on commit.

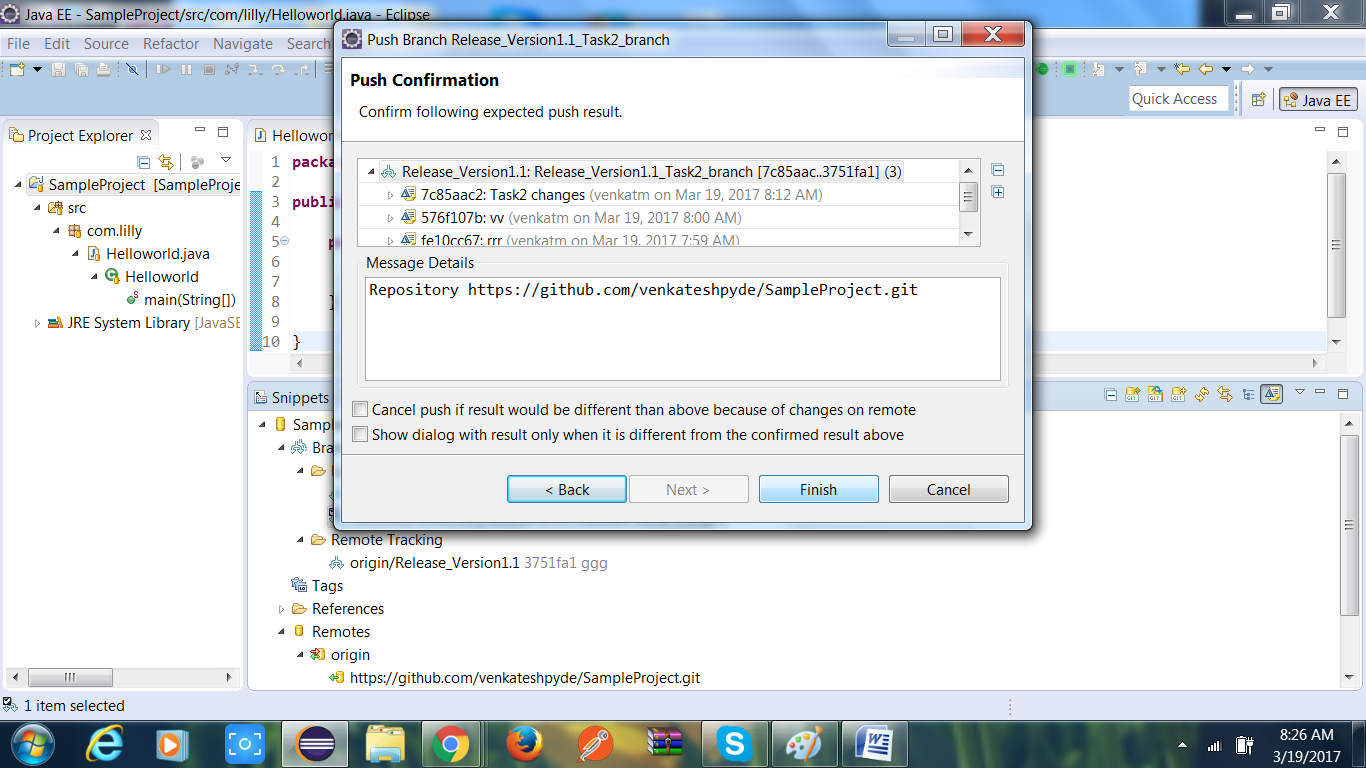


Now changes are moved to local new/another branch, To push local new/another branch changes to remote repository, we need to follow below steps.

***Right click on new/another branch which is under local -> Push Branch***

******

***Click on Next button and finish and ok button.***

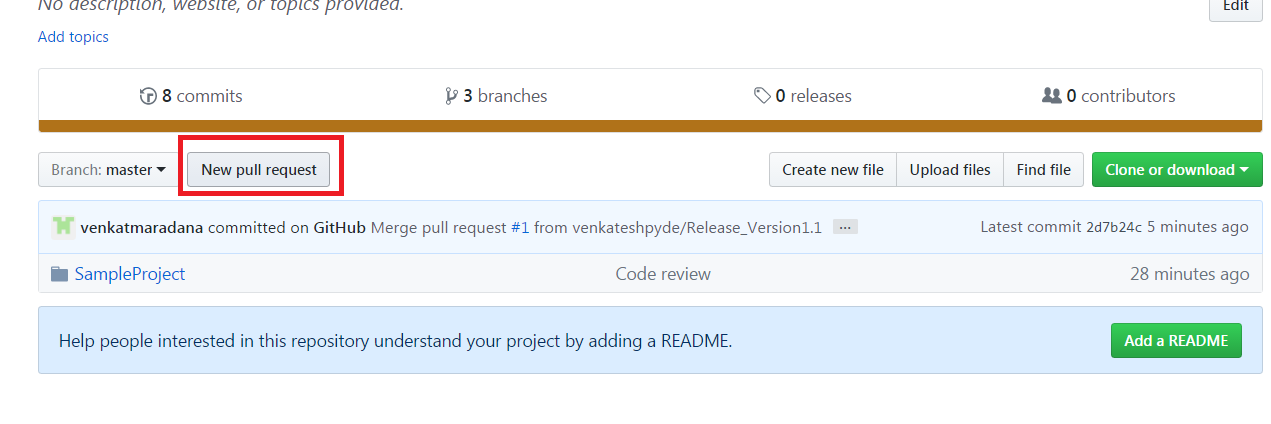
******

1. Code Review(Pull Request)

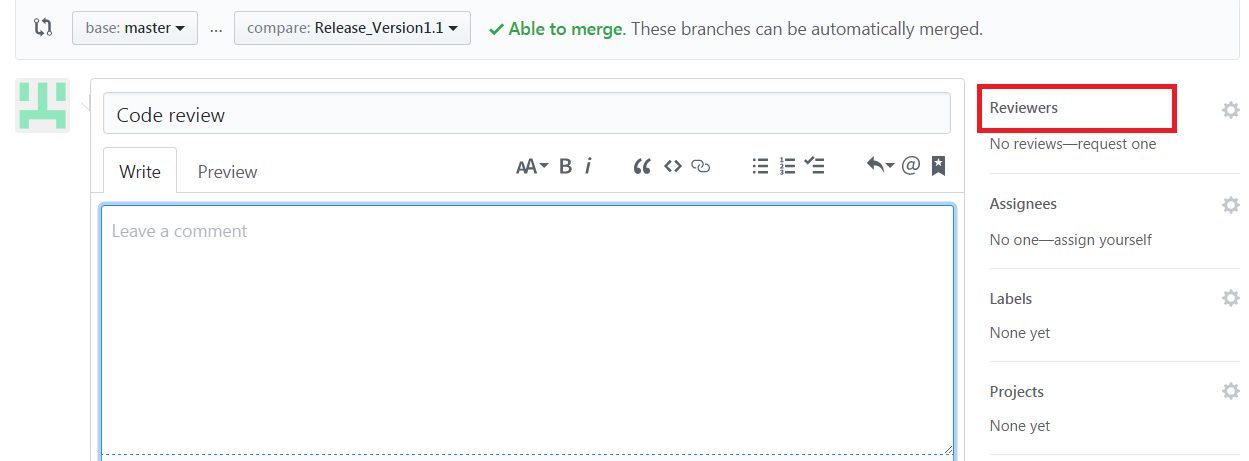
Unfortunately eclipse doesn’t have code review option for GIT, So we need login into GIT after check-in our changes into GIT for code review and then we need to follow below steps.

Step1 :

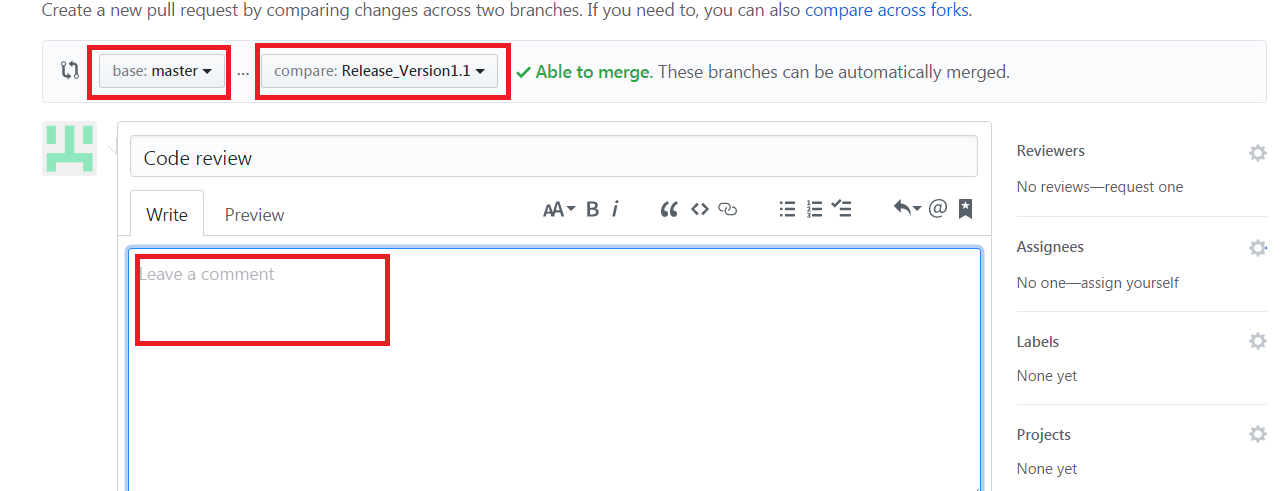
Got to Appropriate project in GIT and Click on New Pull Request button



Add reviewers by click on “Reviewers” link in the right side of the browser.



Add appropriate code review comments and make sure that base point to master and compare should point to current branch.



Then click on Create Pull Request button.

