Contents

[POCs: 2](#_Toc416169878)

[To create end-to-end tracability between Jira, Stash and Bamboo, follow these steps: 4](#_Toc416169879)

[1. Create projects in Jira, Bamboo and Stash with same project key. 4](#_Toc416169880)

[2. Create users Gaurang\_dev and Gaurang\_TL in all 3 products. 4](#_Toc416169881)

[3. In jira, in user Management in Settings, assign group jira-developers to both users. 4](#_Toc416169882)

[4. In jira, create issue and set assignee as Gaurang\_dev and reporter as Gaurang\_TL. 4](#_Toc416169883)

[5. Git init your project in your local drive. Now create repo in Stash and push code to it. 4](#_Toc416169884)

[6. Now configure your Bamboo. Configure plan and default job. 4](#_Toc416169885)

[7. For clover in bamboo, add on not required as it is inbuilt. 6](#_Toc416169886)

[8. For branch specific permissions in Stash repo, firstly make your repo public 8](#_Toc416169887)

[9. To restrict branch specific write permissions, go to branch permissions of repository. 9](#_Toc416169888)

[10. Now create a local branch named “feature/issueId” in local git repo. 9](#_Toc416169889)

[11. Push using Gaurang\_dev ‘s credentials. 9](#_Toc416169890)

[12. Login in stash as TL and approve the pull request and merge it into master branch. 9](#_Toc416169891)

[13. For auto branch merging and for creation of plan branches, 10](#_Toc416169892)

[14. Now create any new local branch in git and push using dev’s credentials. 12](#_Toc416169893)

# POCs:

|  |  |
| --- | --- |
|  |       ***Jira*** |
|  | **New Project** (Employee\_Cybage\_App) |
|  | **User management**- |
|  |          1 Team leader-> TL name |
|  |          1 Developer-> Your Name |
|  | **Versions** |
|  | Release 2.0->Iteration 2.0->feature 1->Add new employee |
|  | Release 2.1->Iteration 2.1->feature 2->List employee |
|  |  |
|  |       ***Stash*** |
|  | **New Repository** (Employee\_Cybage\_App) |
|  | **User management** -TL and Developer |
|  | **Master Branch** |
|  |          1 Team leader-> TL->Admin->read and write access |
|  |          1 Developer->DEV->read access |
|  | **Feature Branch** |
|  |          1 Team leader-> TL->Admin->read and write access |
|  |          1 Developer->DEV->read access->read and write access |
|  | Naming Convention for feature ->Feature/Ticket-ID |
|  |  |
|  |       ***Bamboo*** |
|  | **New Project** (Employee\_Cybage\_App) |
|  | Build Plan->Jobs->Tasks |
|  | First Job->master -> run->CI |
|  | Second Job->Feature (Remote Repo)->run |
|  |  |
|  | **New Feature Branch** |
|  | Build Success->Create pull request->stash->code review->team leader |
|  | Comment->merge-> push->feature->master->build |
|  | Comment->Decline->stash->update -> work on it |
|  | Feature branch -> in bamboo->GateKeeper/Branch updater strategy |
|  |  |
|  |  |
|  | 1. There should be end-end traceability between all atlassian tools. 2. All the three tools stash, jira and bamboo should be linked with one another. 3. JIRA commit ID should be mentioned while committing to stash so that source code details reflect in JIRA. 4. Checkout in bamboo from stash repository. And every commit should trigger the job. 5. In stash repository, in commits, you should be able to see JIRA issue and build status related to that commit. 6. In bamboo, issues should be linked with jira in case of failure 7. Use clover for generating unit test reports 8. Integrate this with sonarQube. CI process in bamboo, should involve sonar analysis. 9. Creation of branch in Stash from Jira |
|  |  |

# To create end-to-end tracability between Jira, Stash and Bamboo, follow these steps:

## Create projects in Jira, Bamboo and Stash with same project key.

1. Create users Gaurang\_dev and Gaurang\_TL in all 3 products. You need admin rights to create users via User Management in settings.

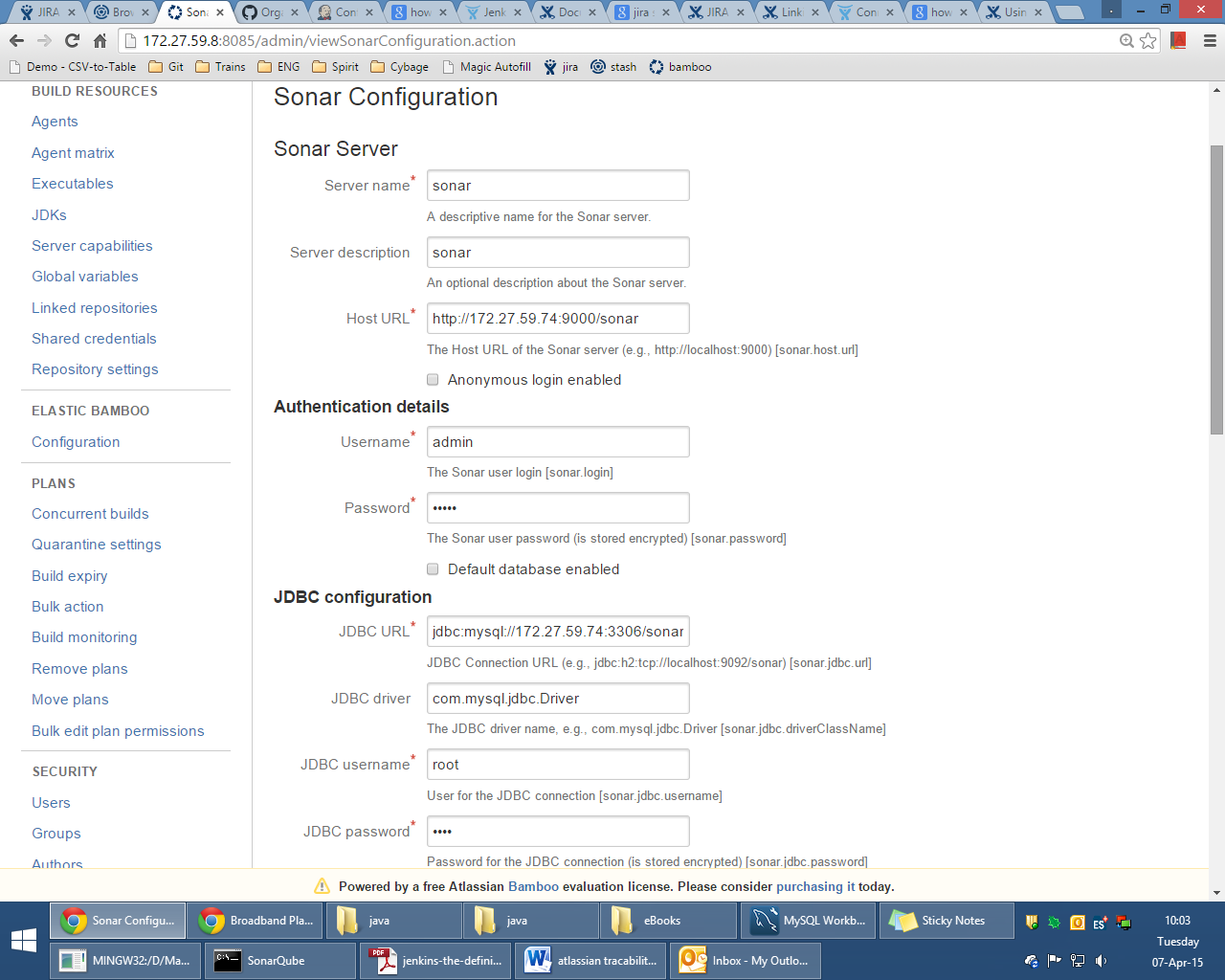
## In jira, in user Management in Settings, assign group jira-developers to both users.

## In jira, create issue and set assignee as Gaurang\_dev and reporter as Gaurang\_TL.

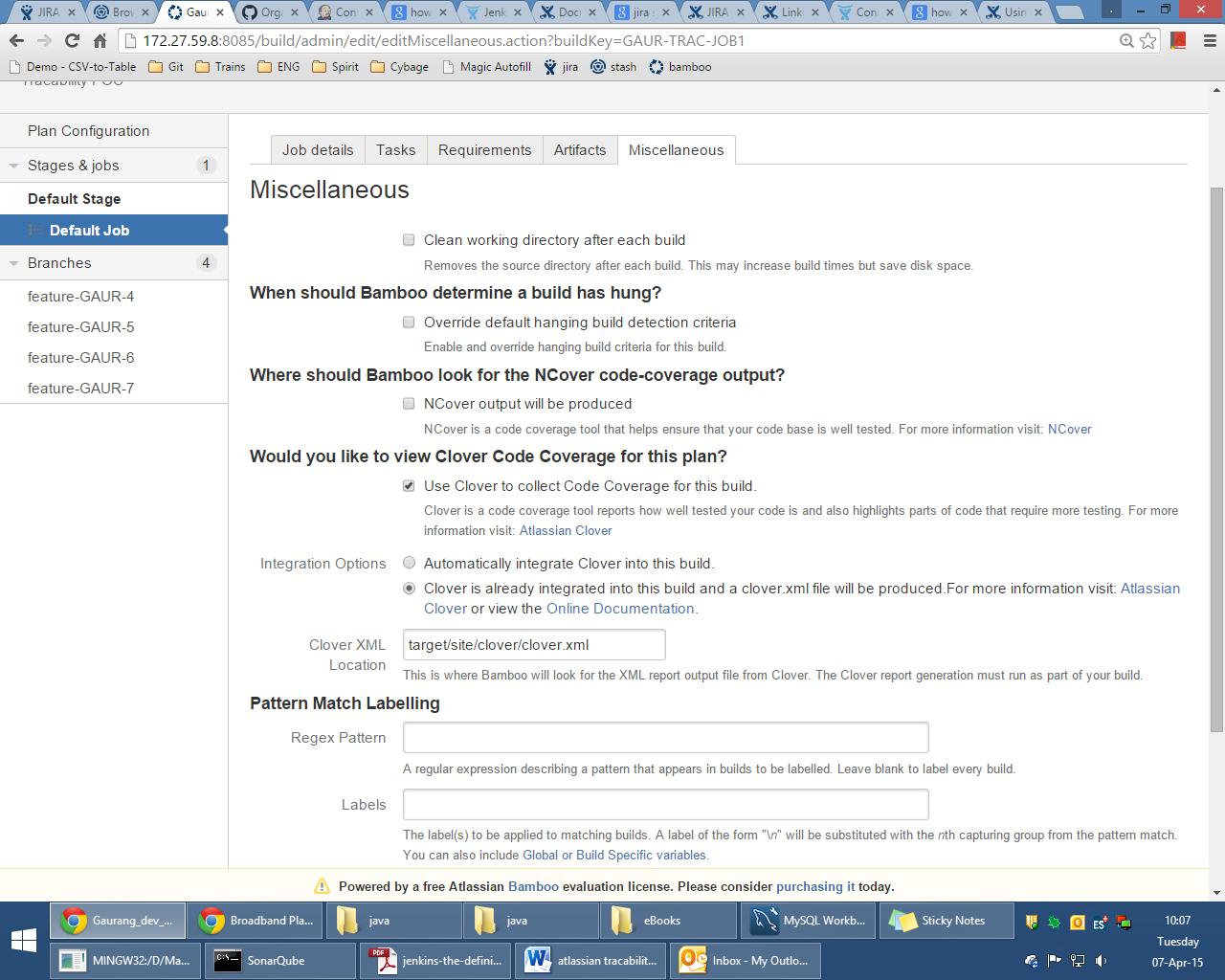
## Git init your project in your local drive. Now create repo in Stash and push code to it.

1. Now configure your Bamboo. Configure plan and default job. Add tasks of source code checkout, Maven 3.x and sonar task.

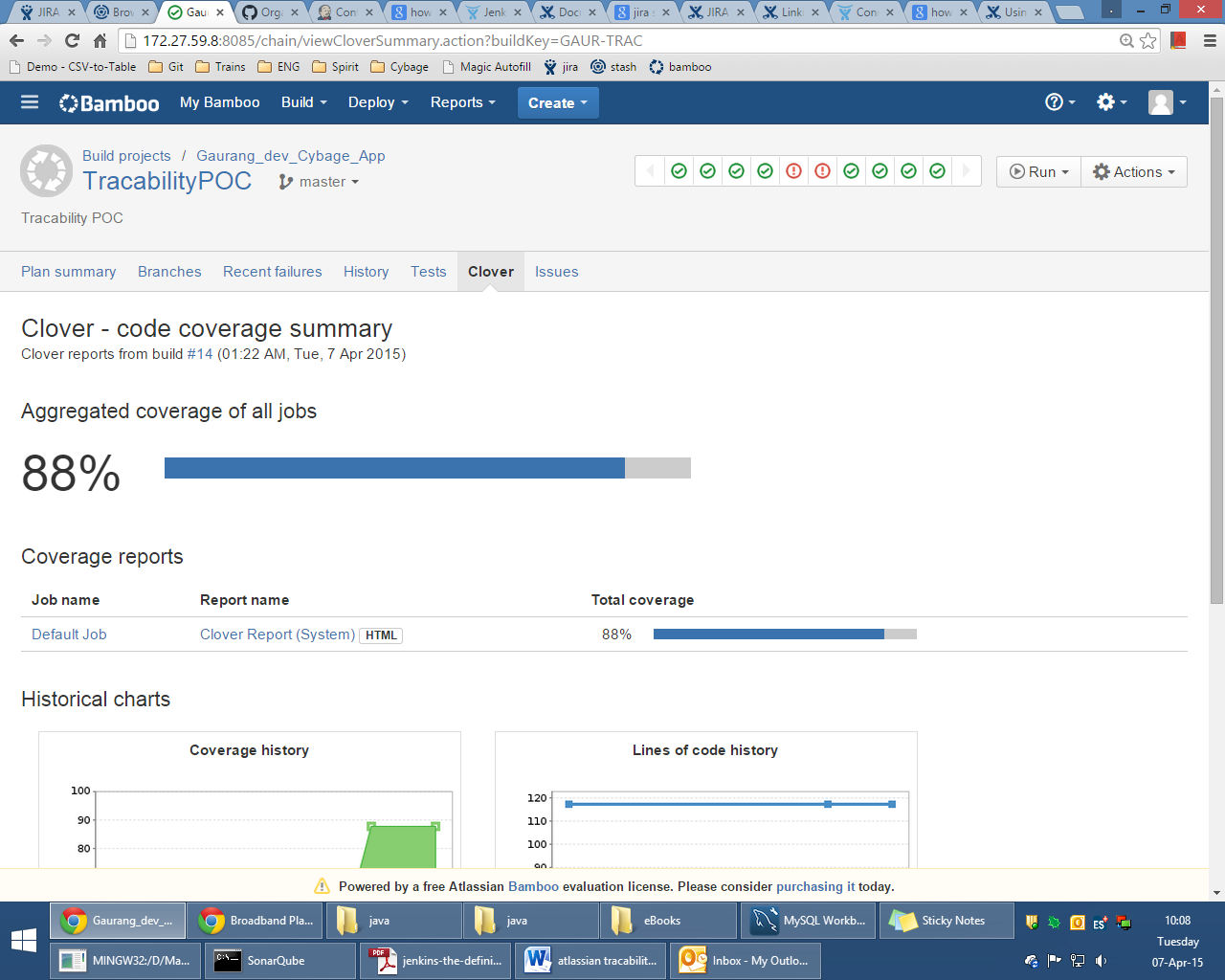
(If sonar task is not displayed, add addon of Sonar from settings->addons and configure as follows:- )



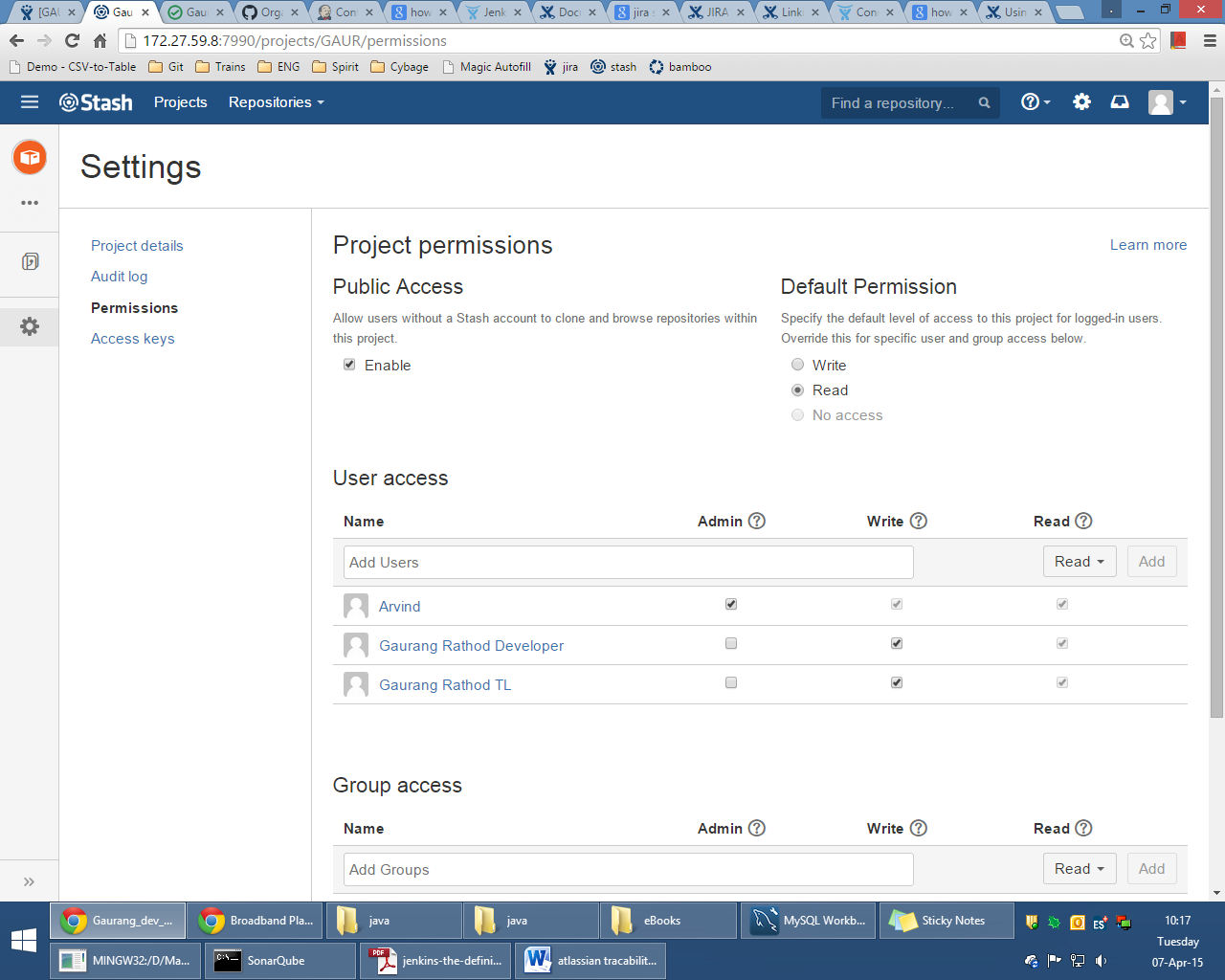
1. For clover in bamboo, add on not required as it is inbuilt. If your pom is creating clover reports, then configure bamboo to render those reports by going to Configuration of default job -> Miscellaneous as follows:-



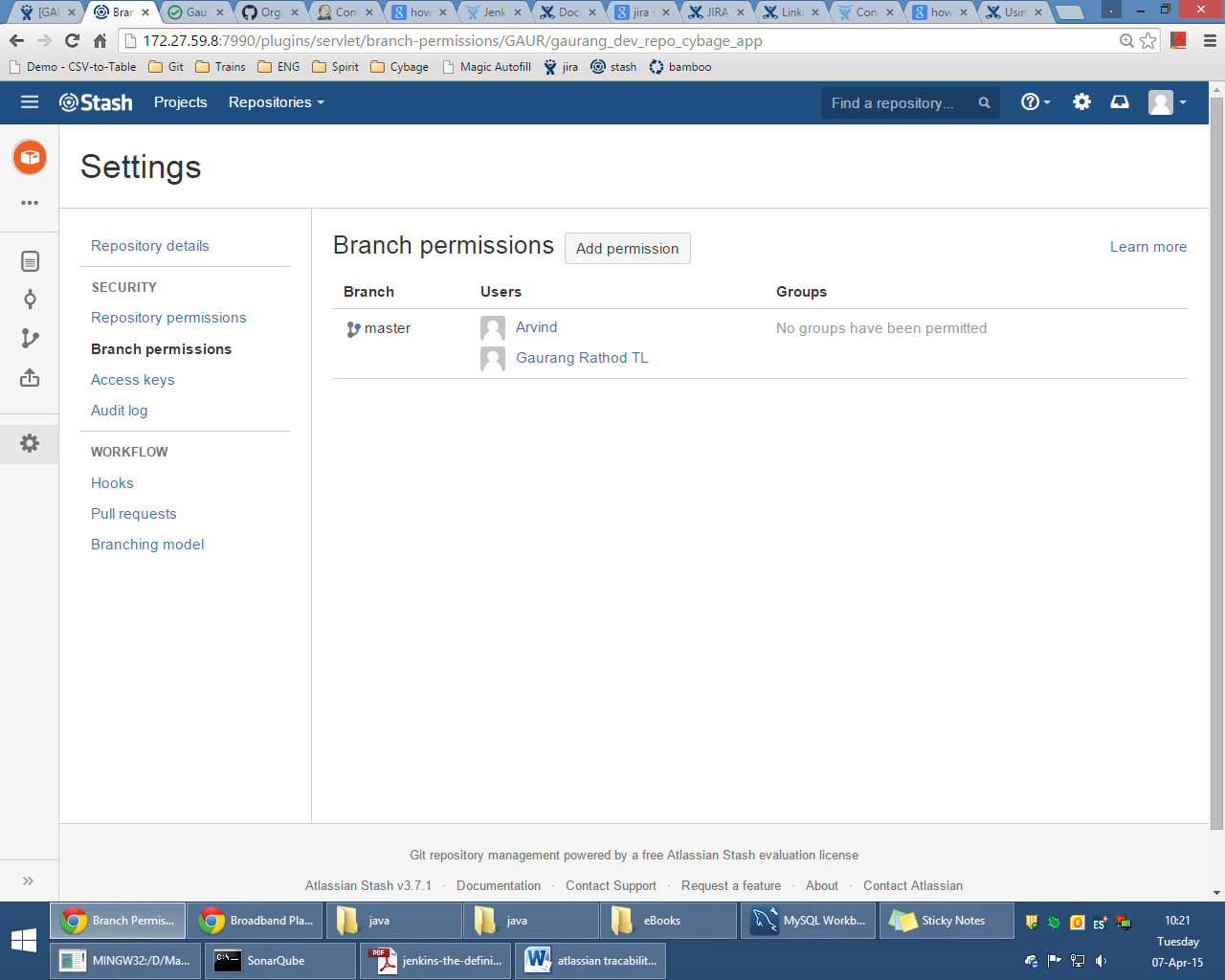
Now you can see Clover as follows:



1. For branch specific permissions in Stash repo, firstly make your repo public so that anyone can clone from it. Then, give write access to your Stash project to both Gaurang\_TL and Gaurang\_dev as shown below. Then, everyone will be able to push to any branch of repository.



1. To restrict branch specific write permissions, go to branch permissions of repository. For master branch, give name of users who can have write access to that branch. Now other users cannot push to this master branch. So, configure as follows and give write access to master branch only to admin and Gaurang\_TL:-



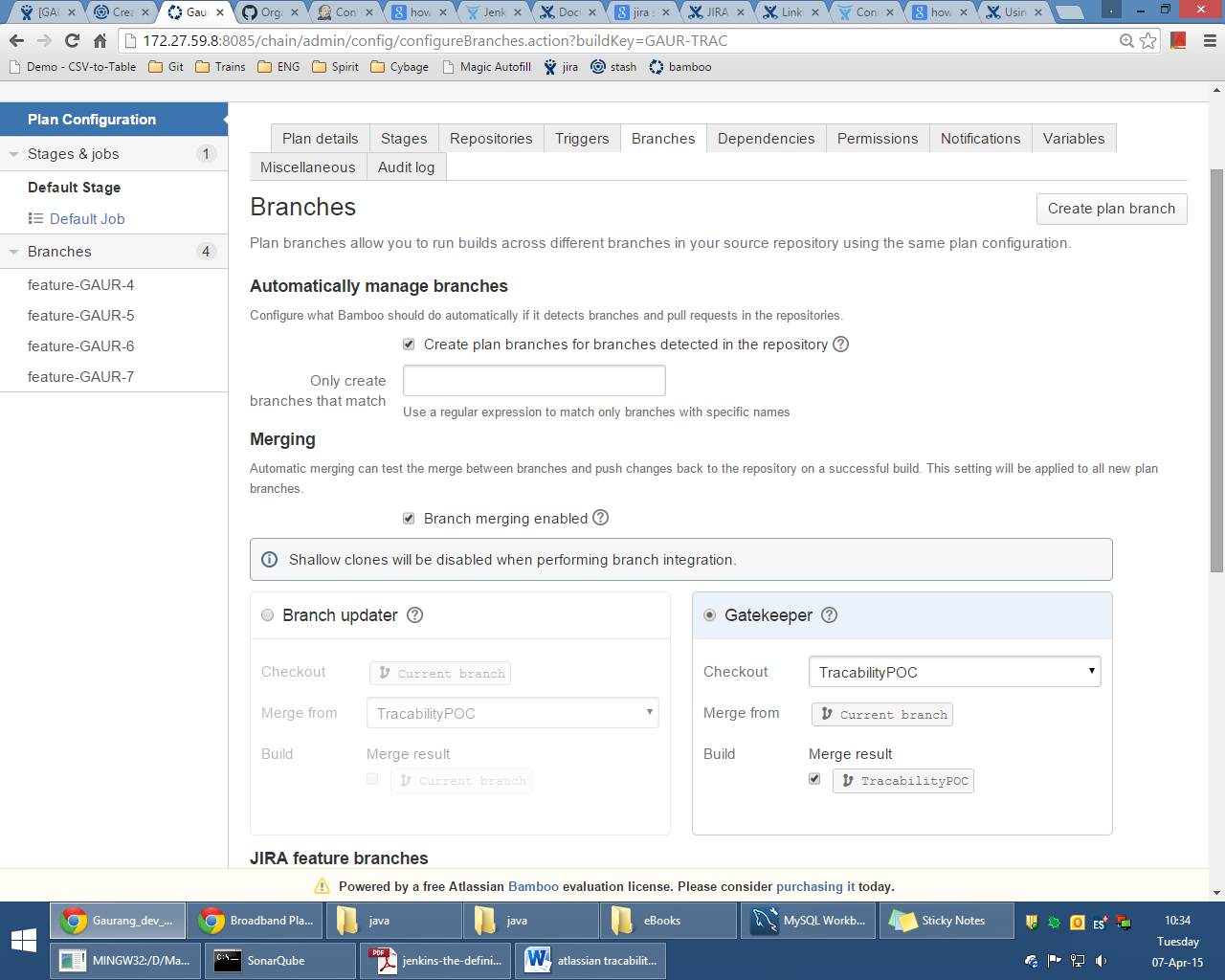
1. Now create a local branch named “feature/issueId” in local git repo. Make code changes and commit. Then create another plan in bamboo. Keep configuration same as main plan. Only change branch to “feature/issueId” in linked repository configuration in plan configuration.
2. Push using Gaurang\_dev ‘s credentials.Now in bamboo, if build of 2nd plan for new feature was successful , then in stash, login as developer and create a pull request. Select source and destination branches. Select approver as TL if asked and create a pull request.

## Login in stash as TL and approve the pull request and merge it into master branch.

1. For auto branch merging and for creation of plan branches, there is a feature in bamboo which detects any new branches in Stash and creates a plan branch whose configuration is inherited from main plan’s configuration. Then, the build is run on the newly detected branch and if the build is successful, then the branch can be automatically merged with the main branch. In this approach, there is no need for pull request.

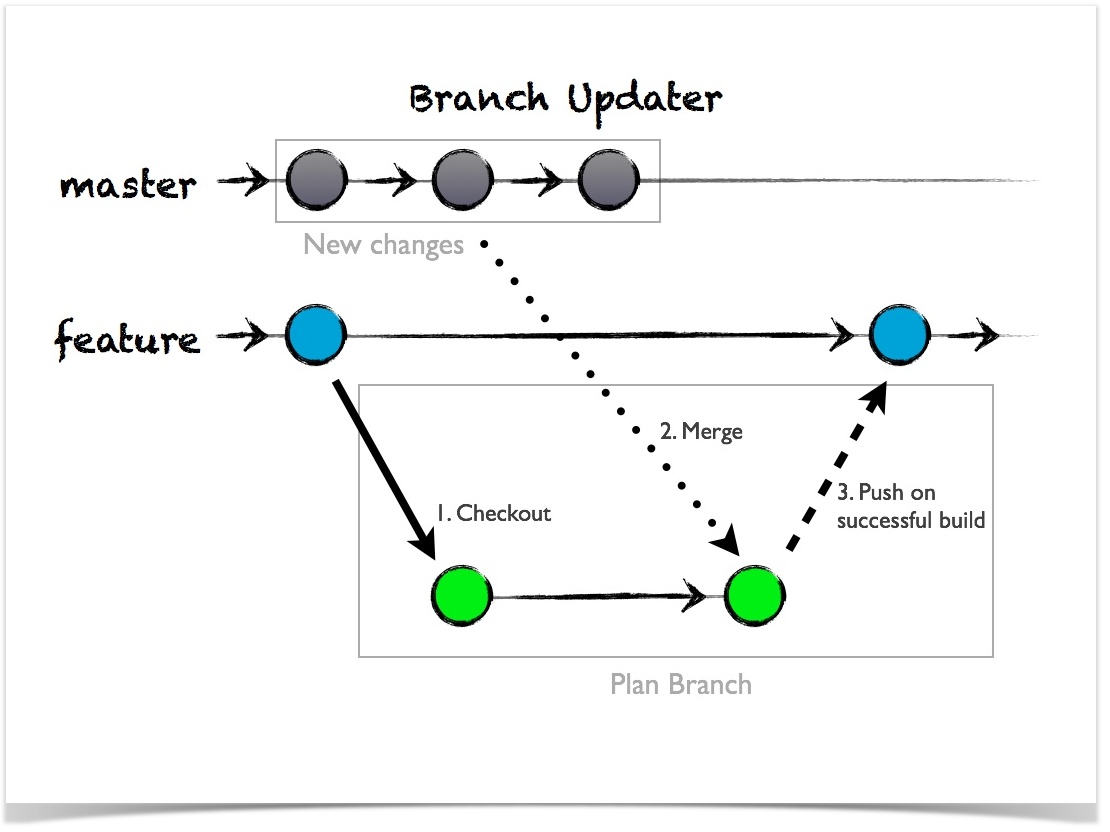
For configuring, go to plan configuration -> branches tab.

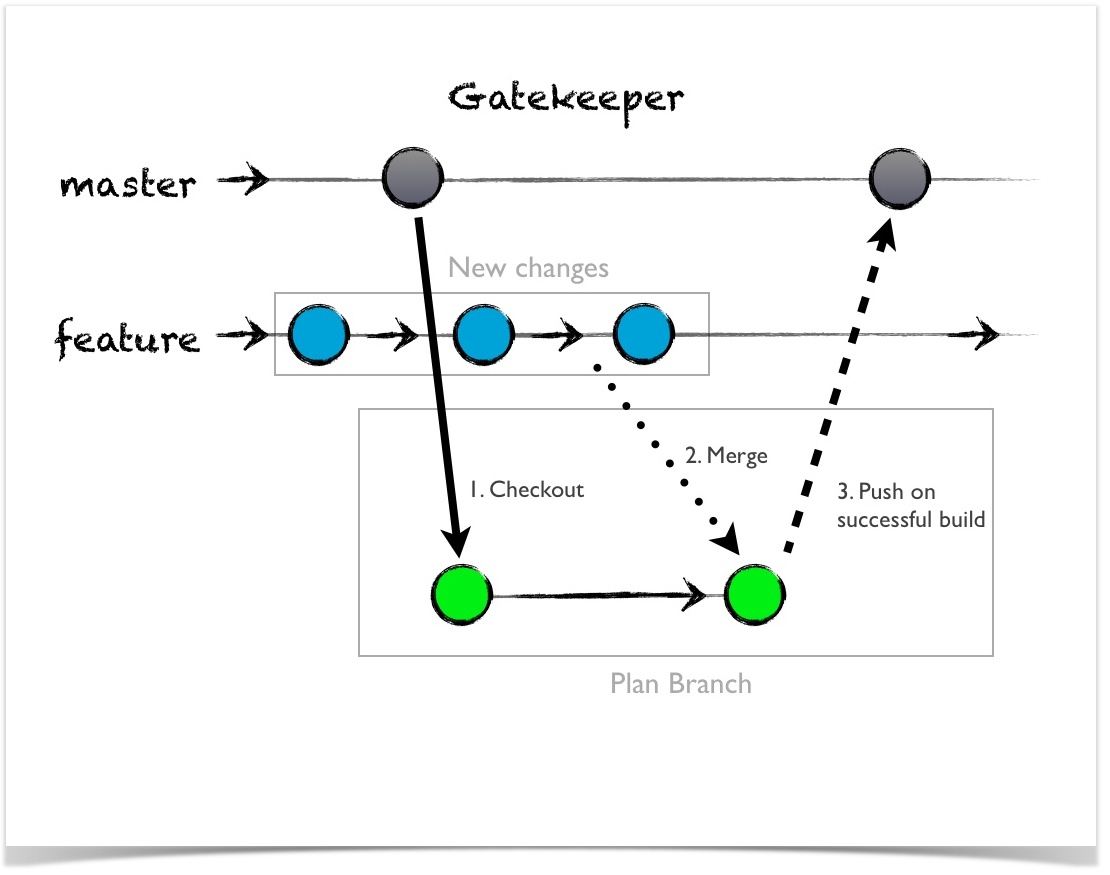
Configure as follows:-



There are 2 strategies:

1. Branch updater
2. Gatekeeper





1. Now create any new local branch in git and push using dev’s credentials. Then build detect the branch in Bamboo and run the build. If it’s successful, then it’ll merge it with master branch automatically.