## horizontal line



Procedure for commiting and creating pull-requests.



24.04.2020

Rahul Kumar Singh ( Software Engineering Intern (2020) )

SenSen Networks

**GOAL**

To guide through the procedure/steps for commiting and creating pull-requests in Bitbucket . To make the steps more understandable and clear , everything will be explained with an example with proper images . Following are the steps to be followed for commiting and creating pull requests :-

**STEPS**

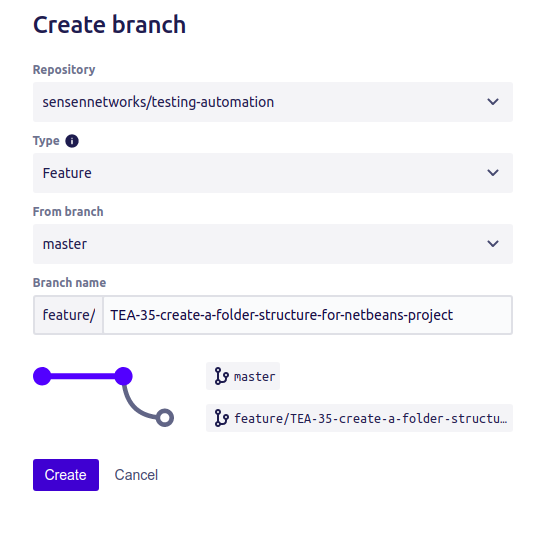
1. Create a directory ( ex. checkouts ) in your terminal . To create a directory , here is a command :- $mkdir directory\_name
2. Go to the mail in which the ticket is assigned and click on it . A new redirected page will appear . The new ‘ticket assigning page’ looks something like this :-



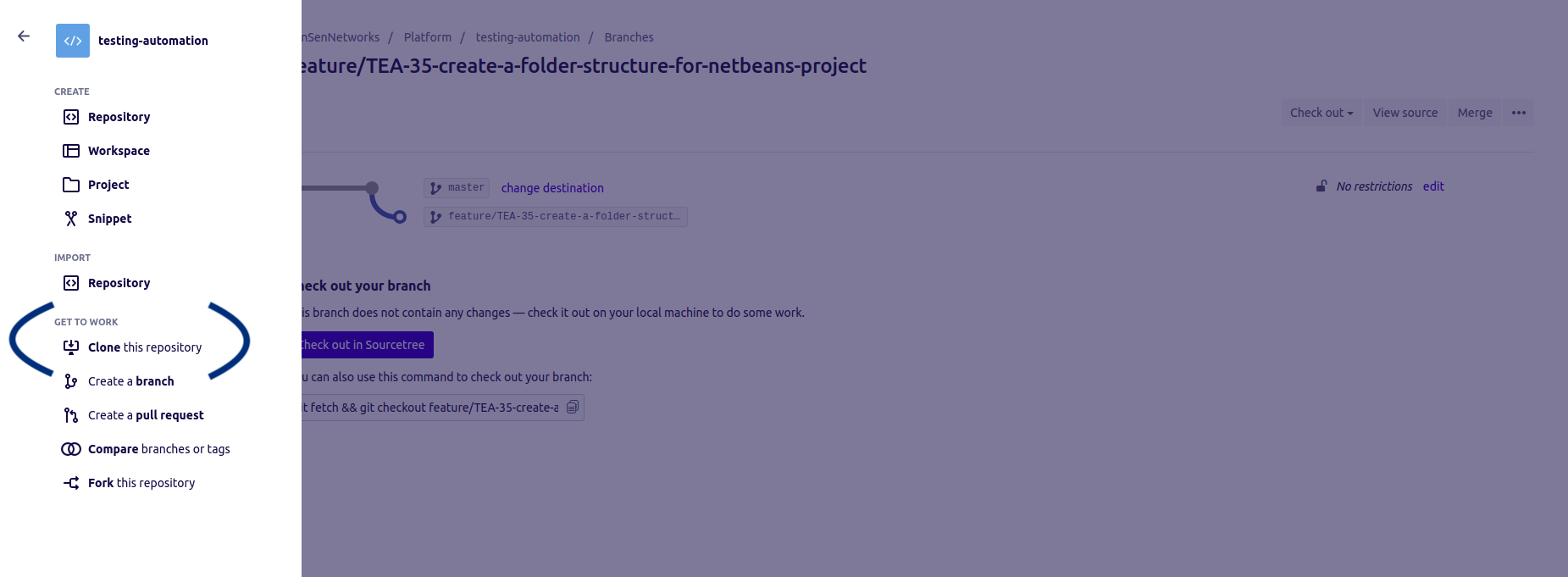
1. Now , a branch needs to be created . For that , in the same page click on the create branch symbol ( encircled in the below image ) .



1. Now a new page will appear . In that page , according to the need , change the Repository , Type etc and click on the ‘Create’ button .



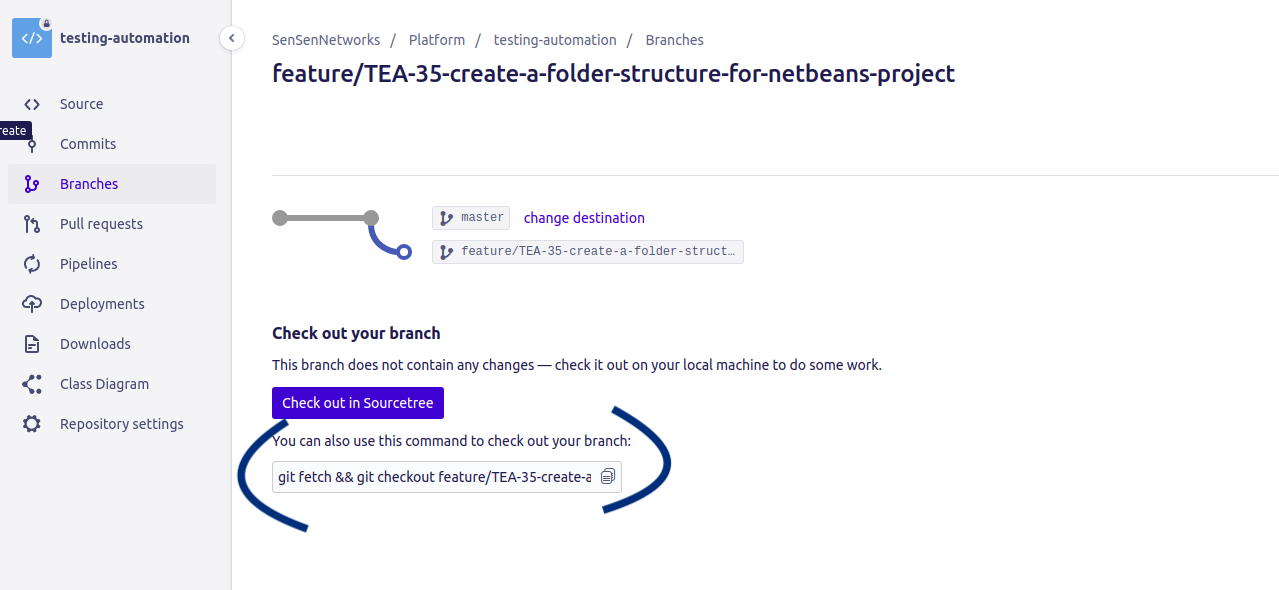
1. A new page appears showing a newly created branch . Now , we need to clone the repository . For that , click on the (+) sign positioned on the left side of the page . Then an overlapping window appears showing option “Clone this repository” . Below is the newly created branch page with the option of cloning .



1. After that a pop up window will appear , copy the link given there .
2. Now , go inside the directory created earlier ( i.e checkouts ) in the terminal ( $cd directory\_name ) and paste that link over there and press enter . After the process ends , the repository is cloned in your directory in your local machine .

In your directory (i.e checkouts ) , now you’ll see a new repository like sensen-interns , testing-automation etc depending upon what you choose earlier while creating the branch .

1. Now , again come back to the newly created branch page . In the bottom of the page , there is one more link , copy that link .



1. Now , go inside the repository( ex. sensen-interns or testing-automation ) which is inside your directory ( i.e checkouts ) in the terminal ( checkouts $cd repository\_name ). Paste the link over there and press enter .
2. Now , copy/place your file which is yet to commit inside the repository ( ex sensen-interns or testing-automation ) .

**NOTE** :- From now on , every command runs inside the repository ( ex. sensen-interns or testing automation ) in the terminal .

1. Check the status, run the command :- $git status
2. Run this command :- $git add file\_name

1. To check the status , run the command :- $git status
2. Now , in order to commit the file , run the command :- $git commit -m “File name: message”
3. Now , run the command :- $git push
4. File is committed . Now , pull request is to be created .
5. Now , come back to the ticket assigning page and refresh it . After refreshing , in the development section , a new commit and branch appears . Click on the branch option .



1. A new pop up window appears , click on the “Create pull request” option present in the right side of the window .



1. A new page appears . Fill the name of the reviewers and create the pull request .
2. The above procedure creates the pull request .
3. Now , again come back to the ticket assigning page and refresh it . After refreshing , in the development section , a new commit , branch and pull requests appears .

**Basic Commands on Git and it’s meaning**

|  |  |  |
| --- | --- | --- |
| Git task | Notes | Git commands |
| [**Tell Git who you are**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-config) | Configure the author name and email address to be used with your commits.  Note that Git [strips some characters](http://stackoverflow.com/questions/26159274/is-it-possible-to-have-a-trailing-period-in-user-name-in-git/26219423#26219423) (for example trailing periods) from user.name. | git config --global user.name "Sam Smith"  git config --global user.email sam@example.com |
| [**Create a new local repository**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-init) |  | git init |
| [**Check out a repository**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone) | Create a working copy of a local repository: | git clone /path/to/repository |
| For a remote server, use: | git clone username@host:/path/to/repository |
| [**Add files**](https://www.atlassian.com/git/tutorials/saving-changes#git-add) | Add one or more files to staging (index): | git add <filename>  git add \* |
| [**Commit**](https://www.atlassian.com/git/tutorials/saving-changes#git-commit) | Commit changes to head (but not yet to the remote repository): | git commit -m "Commit message" |
| Commit any files you've added with git add, and also commit any files you've changed since then: | git commit -a |
| [**Push**](https://www.atlassian.com/git/tutorials/syncing#git-push) | Send changes to the master branch of your remote repository: | git push |
| [**Status**](https://www.atlassian.com/git/tutorials/inspecting-a-repository#git-status) | List the files you've changed and those you still need to add or commit: | git status |
| [**Connect to a remote repository**](https://www.atlassian.com/git/tutorials/syncing#git-remote) | If you haven't connected your local repository to a remote server, add the server to be able to push to it: | git remote add origin <server> |
| List all currently configured remote repositories: | git remote -v |
| [**Branches**](https://www.atlassian.com/git/tutorials/using-branches) | Create a new branch and switch to it: | git checkout -b <branchname> |
| Switch from one branch to another: | git checkout <branch name> |
| List all the branches in your repo, and also tell you what branch you're currently in: | git branch |
| Delete the feature branch: | git branch -d <branchname> |
| Push the branch to your remote repository, so others can use it: | git push origin <branchname> |
| Push all branches to your remote repository: | git push --all origin |
| Delete a branch on your remote repository: | git push origin :<branchname> |
| [**Update from the remote repository**](https://www.atlassian.com/git/tutorials/syncing) | Fetch and merge changes on the remote server to your working directory: | git pull |
| To merge a different branch into your active branch: | git merge <branch name> |
| View all the merge conflicts:  View the conflicts against the base file:  Preview changes, before merging: | git diff  git diff --base <filename>  git diff <source branch> <target branch> |
| After you have manually resolve any conflicts, you mark the changed file: | git add <filename> |
| **Tags** | You can use tagging to mark a significant changeset, such as a release: | git tag 1.0.0 <commitID> |
| CommitId is the leading characters of the changeset ID, up to 10, but must be unique. Get the ID using: | git log |
| Push all tags to remote repository: | git push --tags origin |
| [**Undo local changes**](https://www.atlassian.com/git/tutorials/undoing-changes) | If you mess up, you can replace the changes in your working tree with the last content in head:  Changes already added to the index, as well as new files, will be kept. | git checkout -- <filename> |
| Instead, to drop all your local changes and commits, fetch the latest history from the server and point your local master branch at it, do this: | git fetch origin  git reset --hard origin/master |
| **Search** | Search the working directory for foo(): | git grep "foo()" |

dfb