Lab: (Command Line) Tracking Branches

Estimated time: 10 minutes

> Note: This lab assumes that you are using a command line. If you would prefer to use Sourcetree, there are separate instructions.

In this lab, you will:

1. View tracking branches.
2. Create a state with the local branch one ahead of the tracking branch.

1: View tracking branches.

1. Use bitbucket.org to create a remote repository named `projecte`. You can do this by clicking on the plus sign at the left and then `Create a new > Repository`.

2. Use Bitbucket to create the first commit.

   - Click **Create a README**.
   - Modify the text to contain only the line `# PROJECTE README #`.
   - Click **Commit**. Change the commit message to `add README.md`. Click Commit to create the commit.

   > We created the first commit so that we don't clone an empty repository. When you clone, a tracking branch is set up automatically if you have at least one commit.

3. Clone the `projecte` repository.

   - In Bitbucket, select the + sign on the left. Click `Clone this repository`.
   - Paste the `git clone` command into your command line and change to the project directory.

4. Use `git branch --all` to view the local and tracking branch names.

5. Use `git log --all --oneline --graph` to view the combined log of all local and tracking branches.

   - View the branch labels on the commit. Please answer these questions: (answers at end of lab below)

   QUESTION A. What does the `master` label represent?

QUESTION B. What does the `origin/master` label represent?

QUESTION C. What does the `origin/HEAD` label represent?

> Congratulations, you have viewed tracking branches.

2: Create a state with the local branch one ahead of the tracking branch.

QUESTION D. How do you create a state with the local branch one ahead of the tracking branch?

1. Modify your local README.md. Append the line "Fun with tracking branches." to the file. Add and commit the file with the commit message of "add fun line to README.md".

2. View the commit graph, again using the `--all` option. You should see that the `master` branch label is on the latest commit, but the `origin/master` and `origin/HEAD` labels have stayed on the original commit.

3. Execute `git status`. Notice the message say your branch is ahead of the tracking branch by one commit. This means that you have made one commit locally that is not on the remote repository.

4. Execute `git push` to push the commit to the remote repository. Execute the `git log` and `git status` commands again. You should now see the three branch labels on the latest commit. The local and remote repositories are again synchronized.

5. You will not use the `projecte` repository in future labs. You can delete it.

> Congratulations, you have created a state with the local branch one ahead of the tracking branch. You have also recovered from this state by pushing the commit to the remote repository.

ANSWERS TO QUESTIONS

A) The `master` label represents the tip of the local `master` branch.

B) The `origin/master` label represents the tip of the tracking branch that tracks the `master` branch on the remote repository.

C) The `origin/HEAD` label represents the tip of the default branch on the remote repository. The default branch on the remote repository is the `master` branch.

D) If you make a commit to the local repository without pushing it to the remote repository, the local branch becomes ahead of the tracking branch. The tracking branch only knows what the remote repository knows.