Lab: (Sourcetree) Fetch, Pull and Push

Estimated time: 10 minutes

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

In this lab, you will:

1. Fetch the latest commits from the remote repository.
2. Execute a pull with a fast-forward merge.
3. Execute a pull with a merge commit.
4. Push commits to the remote repository.

1: Fetch the latest commits from the remote repository.

1. Use Bitbucket (bitbucket.org) to create a remote repository named `projectf`.

2. Use Bitbucket to create the first commit.

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

3. Clone the `projectf` repository.

4. Using Bitbucket, modify and commit the README.md file.

    - Click on the **Source** tab.
    - Click on README.md.
    - Click Edit.
    - Append the line "Fun with network commands.".
    - Click Commit and specify a commit message "append line to README.md".

5. Because you have created a commit on the remote repository after cloning the repository, your local `master` branch is behind. View your repository in Sourcetree. At this point, it may or may not be aware of the new commit on the remote repository. This depends on your Sourcetree settings and on timing.

6. Click on the Settings icon. Click on the Advanced tab. If the `Refresh remote status in the background` checkbox is checked, Sourcetree will periodically check for remote

updates and fetch them.

7. View your commit graph. If Sourcetree has done a fetch, you should see your second commit, and that the tracking branch is ahead of your `master` branch. It is safe to click `Fetch` at any time. It will have no impact if Sourcetree has already fetched. Otherwise, it will retrieve the latest remote objects. Click **Fetch** if you don't see the second commit.

> Congratulations, you have fetched the latest commits from the remote repository.

2: Execute a pull with a fast-forward merge.

1. Because we have not added any commits to our local `master` branch, we can perform a pull with a fast-forward merge. Click on **Pull** to begin the merge process.

2. Pull from the remote's `master` branch into your local `master` branch. Notice that the `Create new commit even if fast-forward...` checkbox is **unchecked**. Because this merge is fast-forwardable, no merge commit will be created.

3. Click OK to perform the fast-forward merge. Your `master` branch label should move to the latest commit. The `master` branch is synchronized.

> Congratulations, you have executed a pull with a fast-forward merge.

3: Execute a pull with a merge commit.

1. In your local repository, create an empty `fileA.txt` file. Add and commit the file, specifying a commit message of "add fileA.txt". **Do not Push the commit**.

2. In Bitbucket, make a minor edit to the `README.md` file. **Commit the change**.

3. In Sourcetree, click `Fetch` to retrieve the latest remote commit. You should now see that your local `master` branch is 1 ahead and 1 behind the tracking branch. This is because you made commits locally and in the remote repository.

4. Notice that the Pull and Push icons have a number 1 associated with them. **Attempt to Push**. You will receive a message saying that the updates were rejected, because the tip of your current branch is behind the tracking branch. The message suggests to do a pull.

5. Click `Pull` to start the local merge process.

6. Click OK to perform the merge. Notice that a merge commit was created, combining the work of your local commit and the commit that you made on the remote repository. Also notice that the tracking branch is now two commits behind. At this point, the remote

repository doesn't know about your "add fileA.txt" commit or about the local merge commit.

> Congratulations, you have executed a pull with a merge commit.

4: Push commits to the remote repository.

1. In Sourcetree, click `Push` to add the two local commits to the remote repository. You should now see that the local and tracking branches are synchronized.

2. In Bitbucket, click on `Commits` and verify that the commit graph matches your local commit graph.

3. You will not use the `projectf` repository in future labs. You can delete it.

> Congratulations, you have pushed commits to the remote repository and completed this lab.