EXERCISE 1: Clone and create new repository

Clone git repository, creating a new local copy and

Push it to your own Gitlab remote repository.

EXERCISE 2: .gitignore

You see that build folders and editor specific folders are in the repository and decide to ignore it as a best practice.

Ignore .idea folder, .DS_Store, out and build folders

Hint: remove from git cache first

EXERCISE 3: Feature branch

Create a feature branch and change following:

Upgrade the logstash-logback-encoder version to 7.3

Add image to the index.html file (url: https://www.careeraddict.com/uploads/article/58721/illustration-group-people-team-meeting.jpg)

You are done with the changes. So:

Check your changes using "git diff" and

Commit them if everything is correct.

Note: There is a standard in your team to name commits with descriptive text.

Push your changes to your remote repository.

EXERCISE 4 - Bugfix branch

You find out there is a bug in your project, so you need to fix it using a new bugfix branch:

Create a new bugfix branch

Fix the spelling error in Application.java file

You are done with the changes. So:

Check your changes using "git diff" and

Commit them if everything is correct.

Push your changes to your remote repository.

EXERCISE 5: Merge request

You are done with the feature, now it needs to be tested and deployed. So:

Merge your feature branch into master (using a merge request)

EXERCISE 6: Fix merge conflict

You are on the bugfix branch. You notice the logger library version is old, so:

Update it to version 7.2 (Change the same location in bugfix branch)

Some time went by since you opened your bugfix branch, so you want the up-to-date master state to avoid major conflicts.

Merge the master branch in your bugfix branch - fix the merge conflict!

EXERCISE 7: Revert commit

Still on the bugfix branch. You also noticed a spelling mistake in the index.html file, so you want to fix that in the same branch.

Fix the spelling mistake and commit the fix

You also want to update the image.

So also change the image url (src) in a separate commit.

You are done with the changes:

Push both commits to the remote repository.

Your team members tell you the previous image was the correct one, so you want to undo it. But since you already pushed to remote, you must revert the change.

Revert the last commit and push your changes to remote repository

EXERCISE 8: Reset commit

You found 1 last thing you think must be fixed. Bruno just moved to DevOps team, so Bruno's role must be fixed.

Update the text accordingly

Commit that fix locally (don't push to remote)

However after talking to a colleague, you find out it has already been fixed in another branch. So you want to undo your local commit.

Since commit is only locally, you can reset the commit.

EXERCISE 9: Merge

This time you want to merge your branch directly into master without merge request. So:

merge your bugfix branch into master using git CLI (Hint: master branch must be up-to-date before the merge)

Being on the master branch now. Push your merge commit to remote repository

EXERCISE 10: Delete branches

Now that you are done, both feature and bugfix got deployed and you want to cleanup the old branches.

Delete both branches both locally and remotely