Question

Which command(s) can be used to create a local version of the repository at https://gitlab.crio.do/crio_bytes/me_encapsulation?

Answer Explanation

We use the `git clone` command for downloading repositories. Both of these will work for us

1. `git clone https://gitlab.crio.do/crio_bytes/me_encapsulation.git`
2. `git clone git@gitlab.crio.do:crio_bytes/me_encapsulation.git`

Question

What's the correct sequence of actions for reflecting local changes in the remote repository?

Options

1. Make changes to file.txt -> `git commit` -> `git add file.txt` -> `git push …`
2. Make changes to file.txt -> `git add file.txt`-> `git commit`
3. Make changes to file.txt -> `git add file.txt`-> `git commit` -> `git push …`
4. Make changes to file.txt -> `git commit -a` -> `git push …`

Answer Explanation

3. Make changes to file.txt -> `git add file.txt`-> `git commit` -> `git push …`
4. Make changes to file.txt -> `git commit -a` -> `git push …`

Question

Which of the following is true regarding the repository you downloaded from Gitlab using the `git clone …` command? (Hint: Use "ls -a" command to show hidden files as well)

Options

1. Have no files or folders
2. Have a file, *gitlab.txt*
3. Have a folder, *.git*
4. Have a number of files & folders

Answer Explanation

Any Git repository will have a **.git** folder. It contains data related to Git like that of the remotes, file changes etc.

Question

What will the “git add newFile.txt” command do?

Options

1. Create a new file, *newFile.txt* in the Git repo
2. Mark changes made to *newFile.txt* file for next commit
3. Save changes to *newFile.txt* file to the local Git repo
4. Save changes to *newFile.txt* file to the local & remote Git repo

Answer Explanation

The `git add` command marks changes made to any files for including in the next commit.

Question

Which command makes the local changes visible in the remote Gitlab repository?

Options

1. git status
2. git add
3. git commit
4. git push

Answer Explanation

`git push …` command is used to push any changes to the Gitlab repo. Any changes already included in any of the commits are pushed.

Question

Which of the following is true regarding the “git log” output(when code was pulled after creating the README file via Gitlab)?

Options

1. “git pull” added one new commit
2. “git pull” added two new commits
3. Commit message of the most recent commit is similar to “Pulled README.md from gitlab.crio.do:...”
4. Commit message of the most recent commit is exactly what I entered in the Gitlab UI in **Commit Message** textbox before clicking on the **Commit changes** button

Answer Explanation

Both 1 and 4 are correct answers here

Question

Which of the following is true regarding the “git log” output(after resolving the “Updates were rejected because the remote contains work that you do” error)?

Options

1. “git pull” added one new commit
2. “git pull” added two new commits
3. Commit message of the most recent commit is similar to “Merge branch 'master' of ...”
4. Commit message of the most recent commit is exactly what I entered in the Gitlab UI in **Commit Message** textbox before clicking on the **Commit changes** button

Answer Explanation

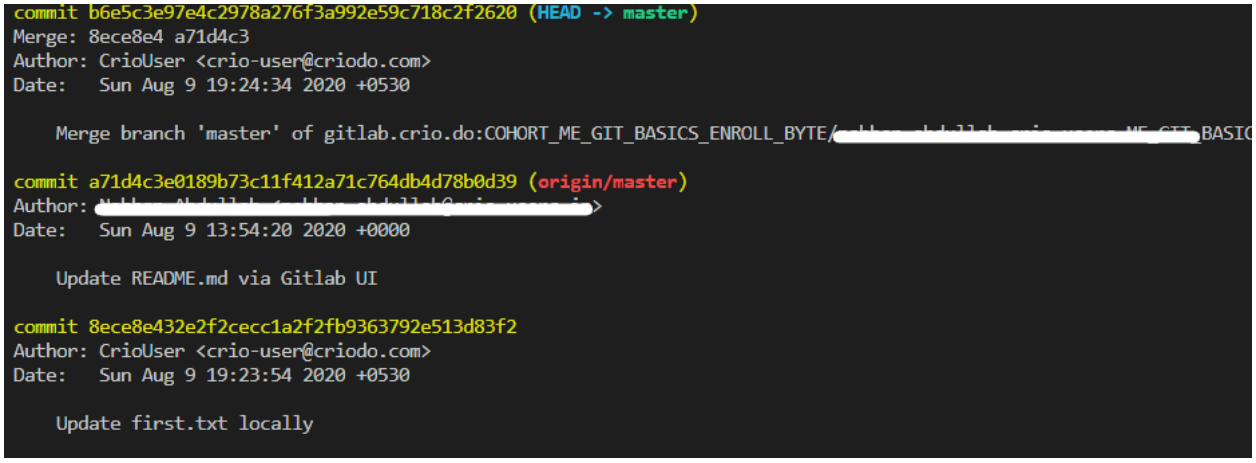
In this case, we made some changes locally and committed it but haven’t pushed yet. This means that the local repo has one new commit that remote doesn’t have. Similarly, we updated the README file via Gitlab UI which also makes a new commit. So, the commit history (list of commits) locally & in the remote would be something like:

- Local repo: Commit 1: Add first.txt file (Milestone 2) -> Commit 2: Create README.md from Gitlab UI (Milestone 2) -> Commit 3: Update first.txt file (Milestone 3)
- Remote repo: Commit 1: Add first.txt file (Milestone 2) -> Commit 2: Create README.md from Gitlab UI (Milestone 2) -> Commit 3: Update README.md via Gitlab UI (Milestone 3)

To combine both these commit histories as one, Git uses a technique called “merging”. As it is done automatically by Git on “git pull”, it also creates a commit to signify this.

So, the two new commits created are:

1. Commit on updating the README file via Gitlab UI
2. Commit auto-created by Git on merging



Question

What happened after you resolved the merge conflict and did a “git add README.md” followed by “git commit”?

Options

1. The editor opened up automatically with a preset commit message - “Merge branch 'master' of …”
2. The terminal showed an error - “Can’t merge …”
3. The terminal showed up with a message - “Successfully resolved merge conflict ..”
4. My computer split into two, Boom!

Answer Explanation

Git will give you a preset commit message as it will need a merge to update the local repo with changes from remote.