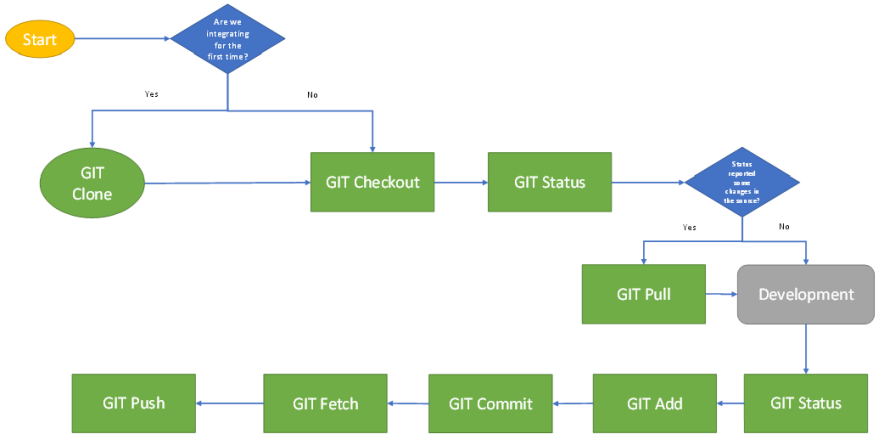
Integrating AzureML notebooks with Git

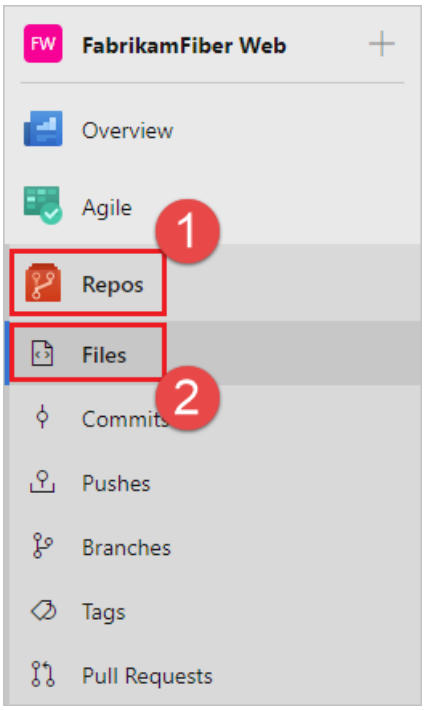
To integrate code in Jupyterlab notebooks in GIT the recommended approach by Microsoft is to use a GIT command-line. The steps requrd to do this is shown and documented below. For more details on the Git command-line refer here page.



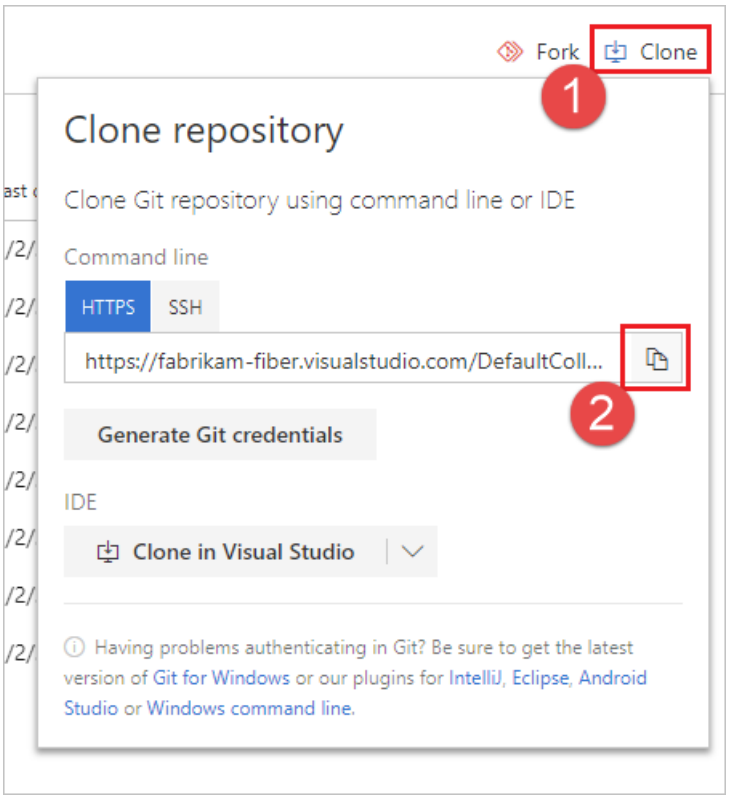
Process used to integrate the code in GIT

GIT Clone—clone the repo to your Azure ML JupyterLab To work with a Git repo, you need to clone the repo you will be working on for this project to your computer. Cloning a repo creates a complete local copy of the repo for you to work with, and downloads all commits and branches in the repo and sets up a named relationship with the repo on the server. Use this relationship to interact with the existing repo, pushing and pulling changes to share code with your team.

1. From your web browser, open the team project for your organization in Azure DevOps and select **Repos > Files**. If you don't have a team project, create one now. For details on how to create a Git repo for your project click here.



2. Select **Clone** in the upper-right corner of the **Files** window and copy the clone URL; and password—you will need both in the following step.



3. Access Azure ML JupyterLab and open a terminal Session. Run `git clone` followed by the followed by the path copied from the Clone URL in the previous section, as shown in the following example. `git clone https://dev.azure.com/fabrikam-fiber/MyFirstProject/_git/`

Next it will ask for a password, paste the password shown in the step above and press **Enter**. Git downloads a copy of the code, including all commits and branches from the repo, into a new folder for you to work with.

4. Switch your directory to the repository that you cloned. For e.g.

```
cd fabrikam-fiber
```

Work with the code

An exampe of how working with Git to save work with `commit` and sharing code with `push`.

1. Navigate to one of the Jupyter Lab notebooks containing code and make some changes 'for e.g. add a comment `#testing my first editin` one of the cells

2. Navigate back to the terminal and ensure you are still in the directory of the repository you cloned.

3. Commit your changes by entering the following command in the Git command window, i.e. your terminal.

```
git commit -a -m "My first commit"
```

When using `git commit`, `-a` means to commit all changed files, and `-m` specifies a commit message.

4. Push your changes up to the Git repo on the server by entering the following command into the Git command window:

`git push`.

5. Switch back to the web portal and select **History** from the **Code** view to view your new commit. The new repo should the commit you just made to your notebook.

for more details on working with Git commands click here.