ANKKIT KUMAR GUPPTA

DATA ANALYST | PROMPT ENGINEER

WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION

Setting Up Git & Git LFS for Managing Large Files on GitHub

When working with files larger than 100MB in GitHub repositories, you need to install both **Git** and **Git LFS (Large File Storage)**. This guide will walk you through installing Git, setting up Git LFS, and managing large files in your GitHub repository.

Step 1: Install Git

Before setting up Git LFS, you need to install Git on your system. Here are the instructions for different operating systems:

1. **For Windows**
   - Download Git from https://git-scm.com/download/win.
   - Run the installer and follow the on-screen instructions to complete the setup.

2. **For macOS**
   - If you have Homebrew, run the following command in the terminal: install git

3. **For Linux**
   - Use your package manager to install Git. For example, on Ubuntu/Debian: sudo apt-get install git

Note: After installing, confirm the installation by running: git –version This command should display the installed Git version.

Step 2: Install Git LFS

Once Git is installed, proceed to install Git LFS.

1. **For Windows**
   - Download Git LFS from https://git-lfs.github.com/.
   - Run the installer to complete the setup.

2. **For macOS**
   - Use Homebrew to install Git LFS: install git-lfs

3. **For Linux**
   - For Ubuntu/Debian, use: sudo apt-get install git-lfs

Step 3: Initialize Git LFS in Your Repository

After installing Git LFS, initialize it in your repository: git lfs install

This command sets up Git LFS in your Git configuration, allowing you to start tracking large files.

Step 4: Use Git LFS to Track Large Files

Now that Git and Git LFS are installed, you can use them to manage large files in your GitHub repository.

1. **Navigate to the Directory**
   Open a terminal and change to the directory where you want to work with the repository:

   ```
   cd Desktop
   ```

2. **Clone the Repository**
   Clone your GitHub repository (or initialize a new one if you haven't yet):

   ```
   git clone https://github.com/Ankkit0413/Business-Insights-360.git
   ```

3. **Navigate to the Repository Folder**
   Change to the specific folder that will contain large files: (Optional if you want to change the main dir.)

   ```
   cd Business-Insights-360/Datasets
   ```

4. **Track Large Files**
   Use Git LFS to track files that are larger than 100MB, such as .sql files: .

   ```
   git lfs track "*.sql"
   ```

   Note: If the file you want to upload to GitHub is not in the designated folder, please ensure that it is placed in the correct folder for uploading. If you intend to upload it to the main repository, make sure the file is also located in the repository folder of GitHub.

5. **Add and Commit Your Changes**
   Stage and commit your changes:

   ```
   git add .
   git commit -m "Add SQL files with Git LFS"
   ```

6. **Push the Changes to GitHub**
   Finally, push your changes to the GitHub repository:

   ```
   git push origin main
   ```

Summary
By following these steps, you've successfully set up Git and Git LFS to handle large files in your GitHub repository. This setup allows you to work with files larger than 100MB seamlessly without exceeding GitHub's file size limits.

Here are the prompts extracted from your command-line session, along with explanations for each step:

Command	Explanation
cd desktop	Changes the current directory to the Desktop folder. cd (change directory) is used to navigate to a specific folder, in this case, the desktop. This is the first step to ensure you're in the correct directory before cloning the repository.
git clone https://github.com/Ankkit0413/Business-Insights-360.git	Clones the GitHub repository named "Business-Insights-360" from the given URL to the current directory on the desktop. git clone downloads the entire repository and its history, creating a local copy on your machine.
cd C:\Users\ng160\Desktop\Business-Insights-360\Datasets	Changes the directory to the Datasets folder within the cloned "Business-Insights-360" project. This is necessary to access files within the Datasets folder and to track them with Git.
git lfs track "*.sql"	Tracks all .sql files with Git LFS (Large File Storage), which is a Git extension for versioning large files. By specifying *.sql, this command sets up tracking for any .sql files, making it easier to manage large SQL files in the repository without bloating the Git history.
git add .	Stages all changes in the current directory (indicated by .), preparing them to be committed to the repository. This includes both the .sql files and any changes to the repository that have been made in the Datasets folder.
git commit -m "SQL"	Commits the staged changes to the repository with a message ("SQL"). This snapshot records the changes, allowing you to keep track of what was modified and making it possible to revert to this point if necessary.
git push origin main	Pushes the committed changes from your local repository to the remote repository on GitHub (branch main). This command sends the changes, including the LFS-tracked files, to GitHub, where they are stored on the remote branch. This allows other collaborators to access and use the updated SQL files.

Purpose of Each Step:

1. **Directory Navigation (cd desktop)**: Ensures you're in the correct working directory before cloning the repository.

2. **Cloning Repository (git clone)**: Creates a local copy of the repository on your machine.

3. **Folder Navigation to Datasets**: Prepares for work in the specific folder where SQL files are located.

4. **Tracking Large Files (git lfs track)**: Sets up Git to handle large SQL files efficiently by using Git LFS.

5. **Adding Changes (git add .)**: Stages files for a commit, marking them as ready to be saved in the repository's history.

6. **Committing Changes (git commit -m "SQL")**: Saves the staged changes with a message, allowing version control and record-keeping of modifications.

7. **Pushing Changes (git push)**: Uploads the committed changes to the remote repository, making them accessible to others.

If you would like to learn more & stay updated, please follow me on LinkedIn