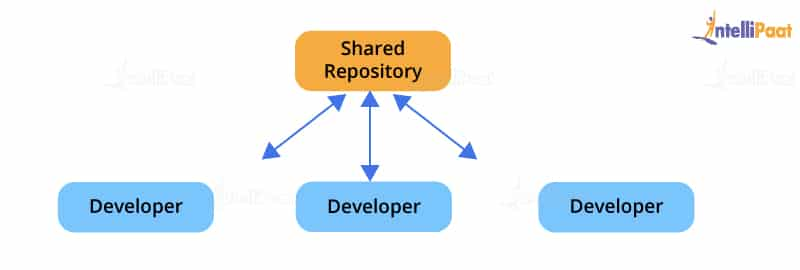
Perform push and pull request in Git

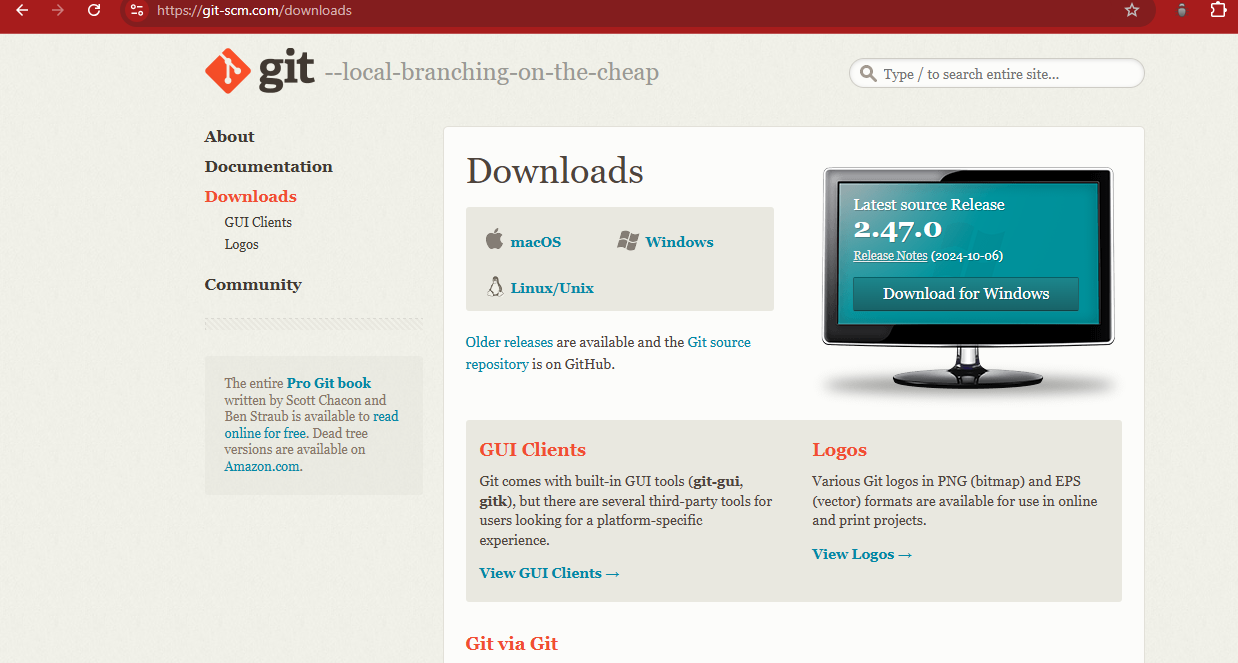
**Introduction**

**Git** is a distributed version control system (VCS) used to track changes in code and collaborate on software development projects. It allows multiple developers to work on a project simultaneously without overwriting each other’s work, making it a cornerstone of modern software development.



#### **Step 1: Install Git Bash on Windows**

* Download **Git Bash** from the official website:<https://git-scm.com/>.



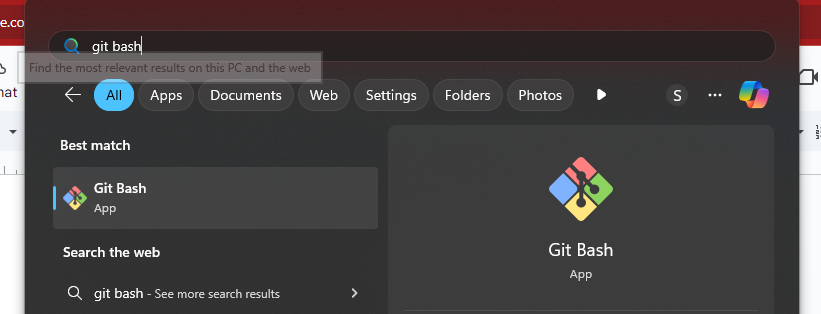
* Follow the installation wizard to install Git Bash on your system.

#### **Step 2: Sign Up on a Git Hosting Platform**

* Choose a Git hosting service:
  + GitHub:<https://github.com>
  + GitLab:<https://gitlab.com>
  + Bitbucket:<https://bitbucket.org>
* Create an account by signing up with your email address.

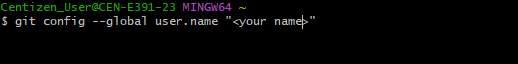
#### **Step 3: Configure Git in Your Local System**

1. Open **Git Bash**.

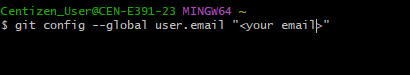


1. Configure your username and email

*git config --global user.name "Your Name"*

**

*git config --global user.email "*[*your.email@example.com*](mailto:your.email@example.com)*"*

**

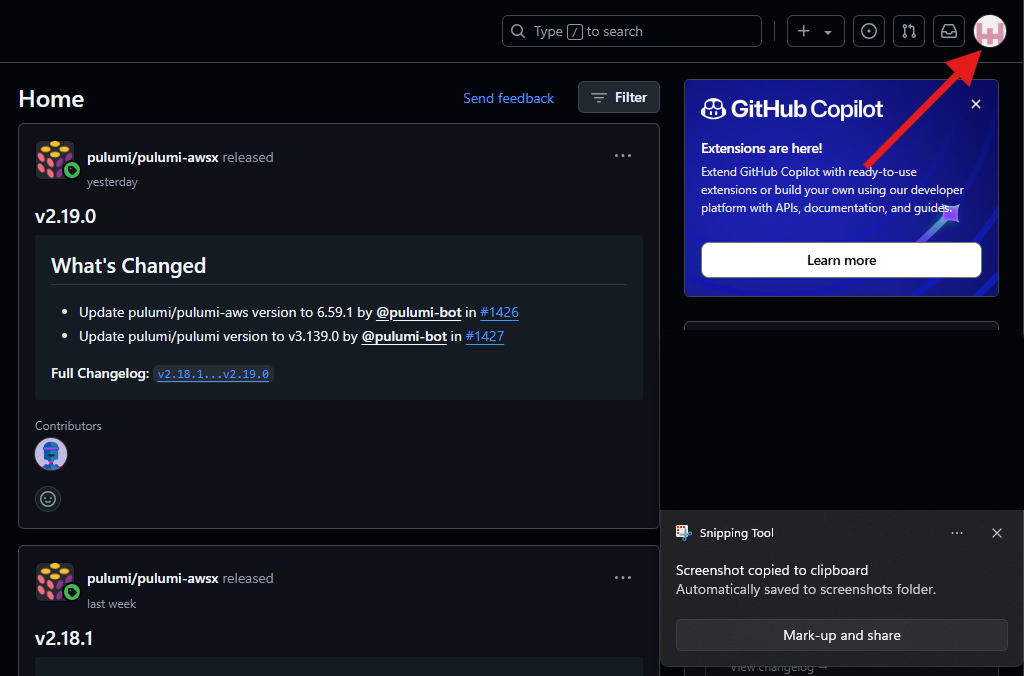
3. Verify the configuration

*git config user.name*

*git config user.email*

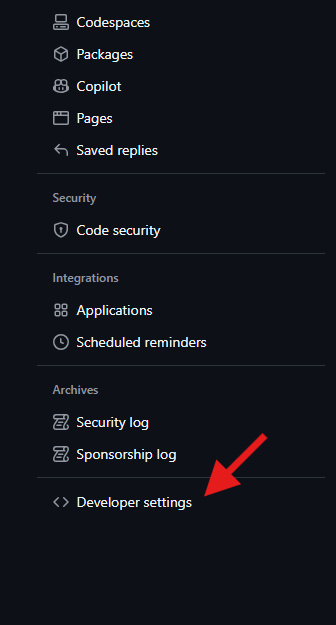
#### **Step 4: Set Up a Personal Access Token**

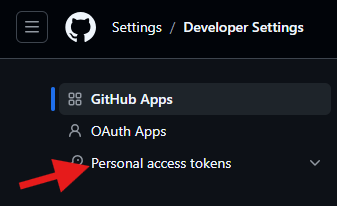
1. Log in to your Git hosting platform (e.g., GitHub).

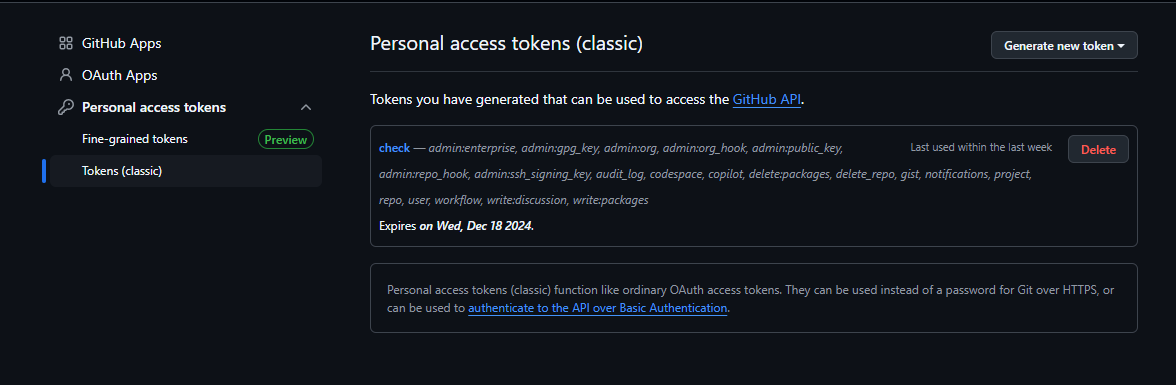


1. Navigate to: **Settings → Developer Settings → Personal Access Tokens → Tokens Classic**.









1. Generate a new token with appropriate permissions and copy the token.  
   **Example token:** ghp\_PjVjItbIMPlbBKefyYLftwLgUUftWT4FA9oa.

#### **Step 5: Move to the Project Folder**

1. In **Git Bash**, navigate to your project folder

*cd /path/to/your/project*

**

#### **Step 6: Initialize the Git Repository**

1. Initialize the project as a Git repository.

Converting our directory into git repository.

*git init*

**

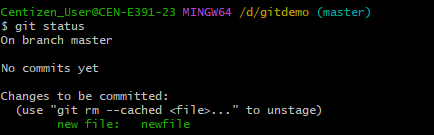
A .git folder is created, marking this folder as a Git repository.

#### **Step 7: Stage and Commit Files**

1. **Stage files:**
   * To stage specific files

*git add <file\_name>*

**

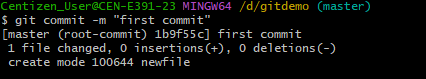
**

* + To stage all files:

*git add .*

2. **Commit files:**

*git commit -m "Initial commit message"*

**

*At present everything is committed in master branch. But the best practice is to create a new branch from the master branch and work on it. So that the source code on the master branch will not be affected unless it is verified thoroughly.*

**Step 8: Create and Switch to a Branch**

**Create a new branch:**

*git branch <branch\_name>*

**

**Switch to the branch:**

*git checkout <branch\_name>*

**

**

*Now the branch is switched to dev1 it is indicated in the right end.*

**Push the branch to remote repository:**

**

**Verify the current branch:**

*git branch*

**

The active branch will be marked with an asterisk

**Step 9: Push Files to Remote Repository**

**Configure the remote repository:**

*git remote add origin https://<your\_token>@*[*github.com/username/reponame.git*](http://github.com/username/reponame.git)

**

If it remote already exists then use

*git remote set-url origin https://<your\_token>@github.com/username/new-repo.git*

Replace <your\_token> with the token you copied earlier, and username/reponame with your repository details.

**Verify the remote repository:**

*git remote -v*

Output should display the remote URL linked to origin.

**Push files to the remote repository:**

*git push -u origin <branch\_name>*

**

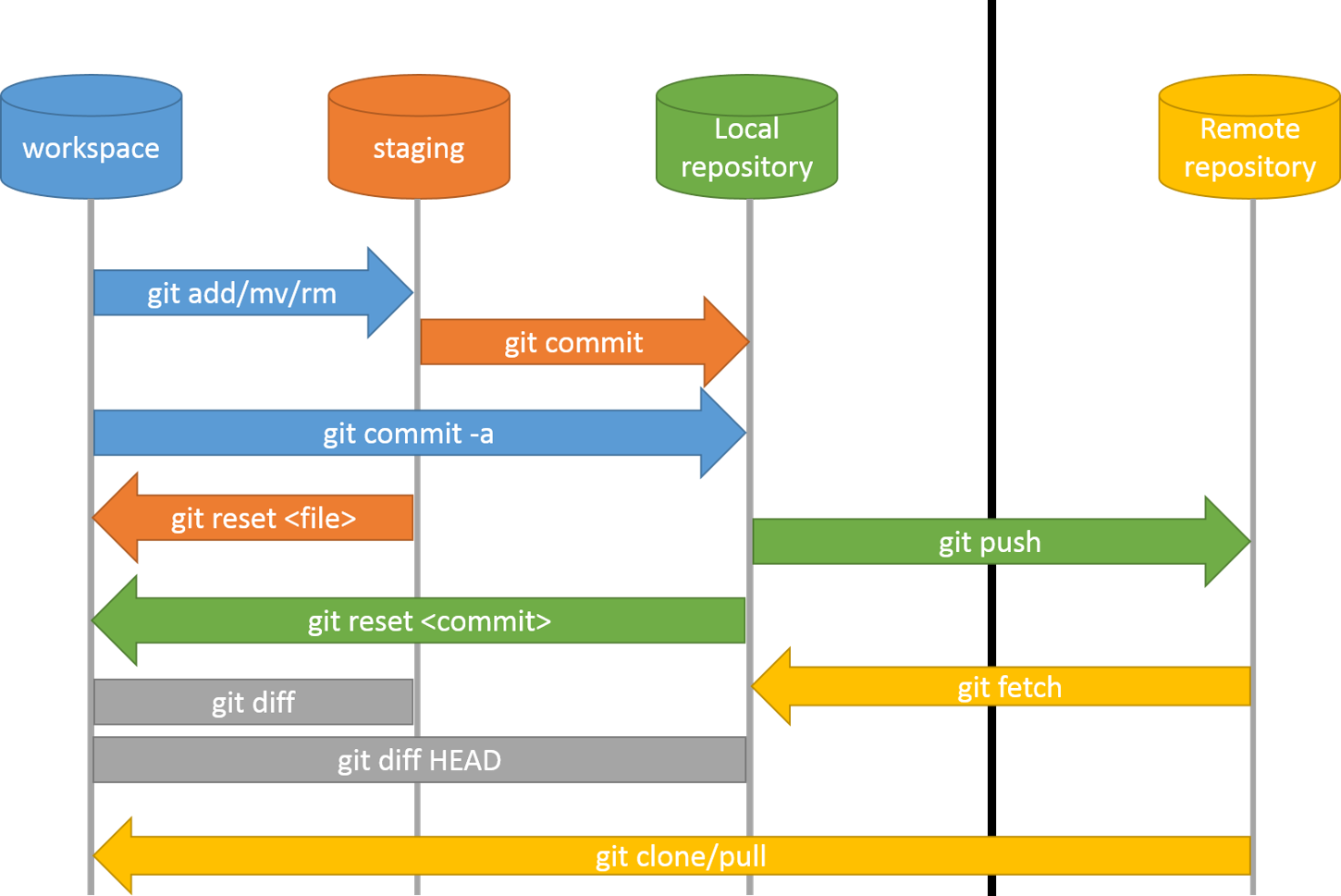
#### **Step 10: Pull Updates from Remote Repository**

To pull updates from the remote branch:

*git pull origin <branch\_name>*

**

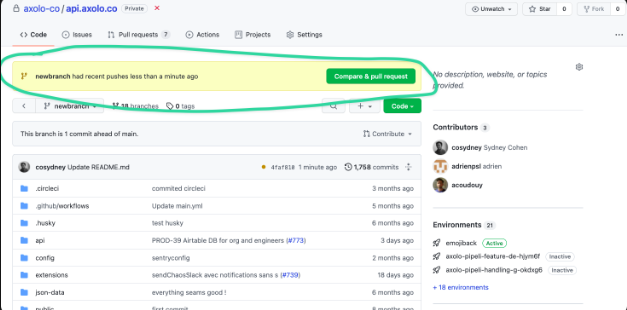
This fetches and merges changes from the remote repository to your local branch.



### 

### **Pull Request (PR)**

* **Platform**: Primarily used in **GitHub** and **Bitbucket**.
* **Definition**: A pull request is a request to merge changes from one branch (often a feature branch) into another branch (typically the main or master branch) of the repository.
* **Process**:
  1. You create a branch and push changes to it.
  2. You open a pull request to propose merging your changes into the target branch (usually the main branch).
  3. Others (team members, collaborators) review the code, discuss it, and possibly suggest changes.
  4. Once approved, the changes are merged into the target branch.
* **GitHub Example**: On GitHub, you open a PR to merge your branch (e.g., feature-xyz) into the main branch.



### **Key Notes**

* Replace placeholders (e.g., <file\_name>, <branch\_name>, <your\_token>) with actual values.
* Use descriptive commit messages to explain changes.
* Ensure your token is kept secure and not shared.