# GIT & GITHUB TUTORIAL:

## What is Git?

- It is a Free and open-source version control system
- It is a tool that tracks the changes in your code overtime.

## What is version control:

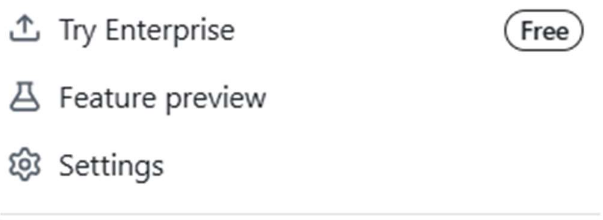
- The management of changes to documents, computer programs, large web sites and other collections of information.
- Some terms regularly used:
  - **Directory** = Folder
  - **Terminal or Command line** = Interface for Text Commands
  - **CLI** = Command Line Interface
  - **Code Editor** = Word processor for writing code, e.g., VS-code
  - **Repository** = Project, or the folder/place where your project is kept.
  - **GitHub** = a website to host your repositories online.

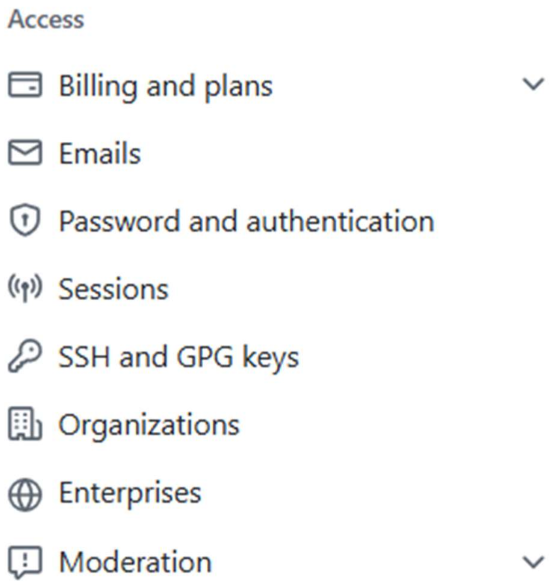
## Some Basic Git commands to remember:

- All the commands are in the lowercase unless explicitly written in capital
  - **Clone** = Bring a repository that is hosted somewhere like GitHub into a folder on your local machine.
  - **Add** = Track your files and changes in Git.
  - **Commit** = Saves your files in Git.
  - **Push** = Upload Git commits to a remote repo, like GitHub.
  - **Pull** = Download changes from remote repo to your local machine, the opposite of push.
  - **Branch** = Shows in which branch you’re working.
  - **Checkout** = To switch/create new branches.

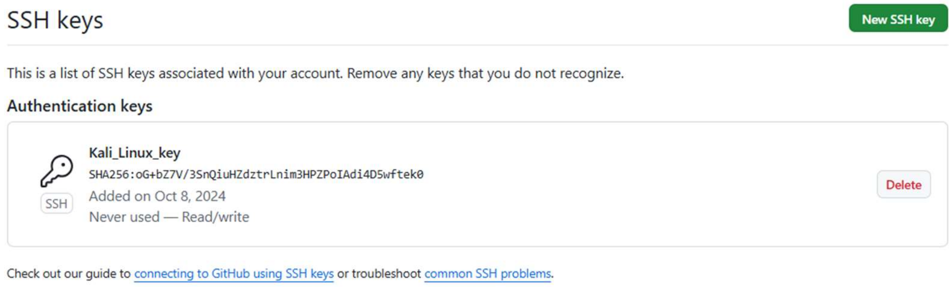
## Setting up your system on GIT with GitHub account:

- To push your repository under “XYZ” account it needs authentication/proving to GitHub.
- Therefore, connect your local machine to your GitHub account using SSH Keys.
- Step 1:
  - Type the following command in CLI:
  - “’ **ssh-keygen -t rsa -b 4096 -C “email@example.com”** ’’’
  - Enter the passphrase as needed or keep it empty. Note the location of the key stored at.
- Step 2:
  - Navigate to the folder/directory where the SSH key is stored.
  - Using “’ **ls | grep testkey** “’ on Linux/macOS search for testkey.pub.
- Step 3:
  - Copy the test key from testkey.pub.
  - Testkey.pub is the key that is going to be uploaded on GitHub’s interface
- Step 4:
  - Navigate to settings; next to SSH/GPG on GitHub:





- Create a new SSH key for the system.



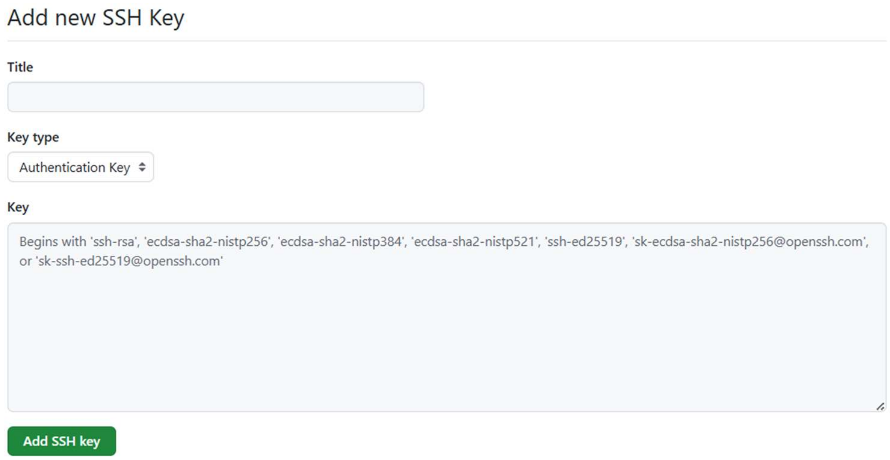
This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

Kali_Linux_key
SHA256:oG+bZ7V/3SnQiuHZdztrLnim3HPZPoIAdi4D5wftek0
Added on Oct 8, 2024
Never used — Read/write

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

- Paste the copied public key by adding a new ssh key on the browser.



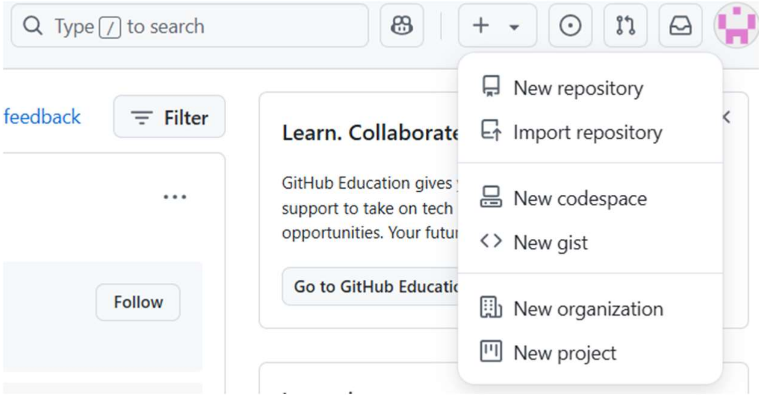
Title

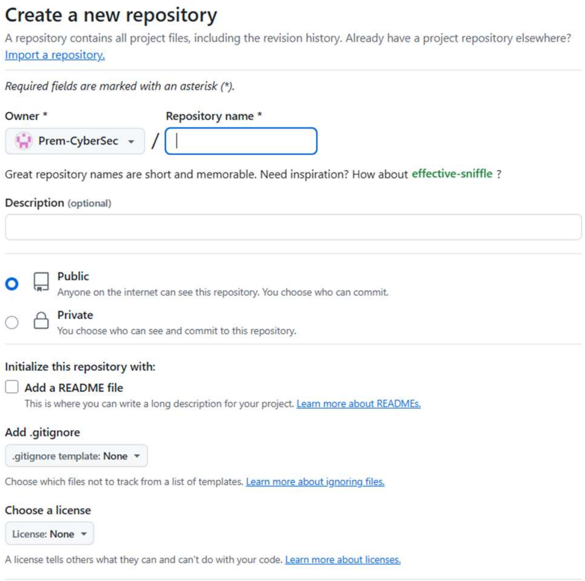
Key type
Authentication Key

Key
Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

- And confirm the access.

# How to create a new repository from GitHub:

- All the commands are in the lowercase unless explicitly written in capital
- Step 1:
    - Hover over the (+) drop down and click on new repository
    - 

- Step 2:
    - Enter your repository name and description.
    - Unless you want to not reveal the repository that you're creating, please select "Public" repository.
    - 

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

*Required fields are marked with an asterisk (*).*

Owner * Prem-CyberSec / Repository name *

Great repository names are short and memorable. Need inspiration? How about effective-sniffle ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

- Step 3:
    - A new repository is created.
    - Always create a README.md file in the repository.
    - This README file is a guide to your repository when other people come and have a look at your repository.

# How to create a repository locally and push:

- All the commands are in the lowercase unless explicitly written in capital
- Step 1:
    - Create a new folder/directory and open the terminal inside the directory.
    - Type "' **git init** "' to initializes an empty git repository.
- Step 2:
    - Fill the details (i.e., make changes to your repository locally) in the local folder/directory and save the changes locally.
    - Once you are sure of the changes you have made move on to the next steps.
- Step 3:

  - The files in the repository are being tracked by GIT. If you create a new file or make any changes to existing file in the repository locally; the file is then modified to local repository only.
  - You need to explicitly add instruct GIT to track the file changes and for that we will be using the following command:
    - **“’ git add .(period) “’**
    - **“’ git add filename.extension “’**
- Step 4:
  - We need to commit the changes made to ensure that these changes are ready to be pushed on the remote repository; we do this with following command:
  - “’ **git commit -m “Changes to the Title of the file”** “’
- Step 5:
  - Create a new repository on the GitHub and copy the link.
  - Use “’ **git remote add origin (copied link)** “’
  - This connects the empty repository on the GitHub to the local machine.
- Step 6:
  - Use “’ **git remote -v** “’ to check if the connection is made or not.
  - We push the changes on the internet using the command.
  - Use “’ **git push location(origin) branch(master)** “’ to push the repository on the GitHub.
  - Use “’ **git push -u location(origin) branch(master)** “’ to setup an upstream for that branch.

## How to clone, update and push to the repository:

- Step 1: All the commands are in the lowercase unless explicitly written in capital
  - Always check git version using the git bash or cmd.
  - **“’ Git –version “’**
- Step 2:
  - **“’ git clone (HTTPS link generated)/(SSH link generated) “’**
- Step 3:
  - The files in the repository are being tracked by GIT. If you create a new file or make any changes to existing file in the repository locally; the file is then modified to local repository only.
  - You need to explicitly instruct GIT to track the file changes and for that we will be using the following command:
    - “’ **git add .(period)** “’
    - “’ **git add filename.extension** “’
- Step 4:
  - We need to commit the changes made to ensure that these changes are ready to be pushed on the remote repository; we do this with following command:
  - “’ **git commit -m “Write your message”** “’
- Step 5:
  - Create a new repository on the GitHub and copy the link.
  - Use “’ **git remote add origin (copied link)** “’
  - This connects the empty repository on the GitHub to the local machine.
- Step 6:
  - Use “’ **git remote -v** “’ to check if the connection is made or not.
  - We push the changes on the internet using the command.
  - Use “’ **git push location(origin) branch(master)** “’ to push the repository on the GitHub.
  - Use “’ **git push -u location(origin) branch(master)** “’ to setup an upstream for that branch.
-

## How to update your local repository:

- All the commands are in the lowercase unless explicitly written in capital

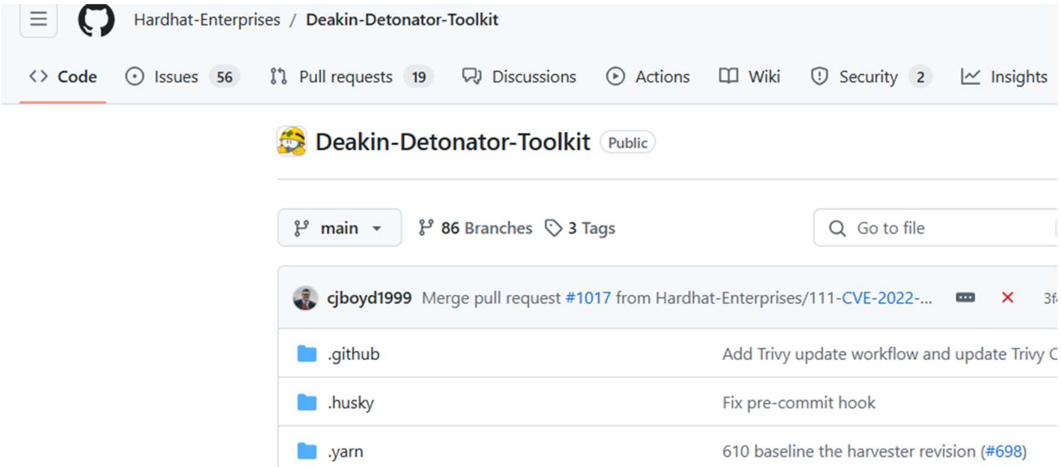
- Step 1:
    - To update changes on local machine that are made on GitHub:
        - “’ **git pull origin branch_name** “’
    - if the upstream settings have been mentioned already for that branch:
        - “’ **git pull** “’

## How to create and delete a branch:

- Step 1:
    - Use “’ **git branch** “’ to see which branches are created.
- Step 2:
    - To create a new branch.
        - “’ **git branch (branch_name)** “’
- Step 3:
    - To switch into different branches:
        - “’ **git checkout branch_name** “’
- Step 3:
    - To delete a branch:
        - “’ **git branch -d (branch Name)** ’’’
    - To force a delete of the branch irrespective of merging:
        - “’ **git branch -D (branch Name)** “’
- Step 4:
    - To delete a remote branch
        - “’ **git push origin(location) --delete (branch_name)** “’
- Note:
    - All changes made to an individual branch will not be reflected in another branch unless merged.
    - All remote branches are to be deleted, if you delete the local branches related to it.
    - If you want to ensure that merge conflicts are fully resolved; try looking into the branch changes made and check if there are merge conflicts.
    - From there you can either choose to revert to session before local changes and pull the changes that was modified first on the remotely before pushing the changes or delete the branch that is causing the conflict based on the priority of the branch.

## How to create a pull request:

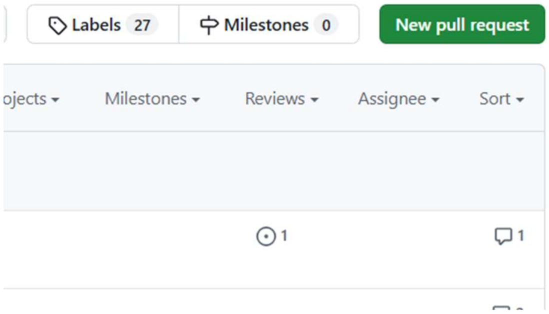
- All the commands are in the lowercase unless explicitly written in capital
- Step 1:
    - Create a new branch using:
        - “’ **git branch (branch_name**) “’
    - Perform all the changes in that branch.
- Step 2:
    - Commit and push the changes on the GitHub.
- Step 3:
    - Check the difference between the main branch and the feature branch.
    - If there are any differences, switch to main branch.
        - “’ **git diff (branch_name)** “’
- Step 4:
    - Create a pull request for the specific branch that you want to merge with the main/master branch



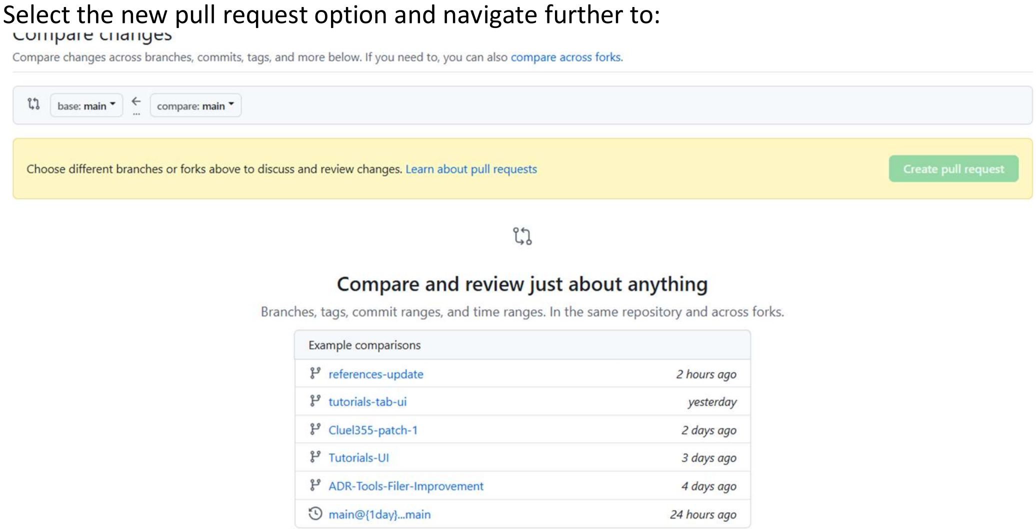
cjboyd1999 Merge pull request #1017 from Hardhat-Enterprises/111-CVE-2022-...

| | |
|---|---|
| .github | Add Trivy update workflow and update Trivy C |
| .husky | Fix pre-commit hook |
| .yarn | 610 baseline the harvester revision (#698) |

- Select the Pull request option and navigate further to see the following:



- Select the new pull request option and navigate further to:



Compare changes

Compare changes across branches, commits, tags, and more below. If you need to, you can also compare across forks.

Choose different branches or forks above to discuss and review changes. Learn about pull requests

Compare and review just about anything

Branches, tags, commit ranges, and time ranges. In the same repository and across forks.

| Example comparisons | |
|---|---|
| references-update | 2 hours ago |
| tutorials-tab-ui | yesterday |
| Cluel355-patch-1 | 2 days ago |
| Tutorials-UI | 3 days ago |
| ADR-Tools-Filer-Improvement | 4 days ago |
| main@{1day}...main | 24 hours ago |

- Choose the branch you want to compare and complete the necessary steps to create a new pull request.
- When the maintainer of the repository approves, your branch will be automatically merged with the main branch.

# How to upload large files that are over 100mb:

- All the commands are in the lowercase unless explicitly written in capital
- Step 1:
  - “’ **git lfs install “’** install the git large file storage to upload.
- Step 2:
  - **“’ git lfs track "*.ext" “’**
  - To track the large files that you wish to upload on GitHub.
- Step 3:
  - To upload the files on the repository, follow the commands below:
    - **“’ git add file.ext “’**
    - **“’ git commit -m "Add design file" “’**
    - **“’ git push origin main “’**