GIT & GITHUB

By- Pushkar kishore

Cyber Security Researcher

NIT ROURKELA

GIT

- GIT is a version control system.
- What is *version control system*?
  - A tool that helps to track changes in code.
- GIT is:
  - Popular
  - Free and Open source
  - Fast and Scalable

GIT USAGE

- To track the history
  - Let us consider we are creating a software having four components.
  - Components are added one by one.
  - At some point, we want to revert to previous state in software development process.

Not worry – GIT comes to rescue

GIT USAGE (Cont..)

- Collaborate
  - Many developers are working on the same code.
  - They are parallel making changes to that code.
  - Company has to decide which code changes they want to keep.
  - Some changes are kept while others are discarded.

GITHUB

- Website that allows developers to store and manage their code using GIT.
- Link – https://github.com
- Project is uploaded in the forms of folder on GitHub. It is also known as *repository*.
- We can upload our local project on GitHub.

GITHUB ACCOUNT CREATION

- Go to https://github.com and click on signup.
- Enter your email, username and set a password.
- Phew, fill some basic details and you can land on personalized user homepage.
- Already signed up. Then, login with your credentials.

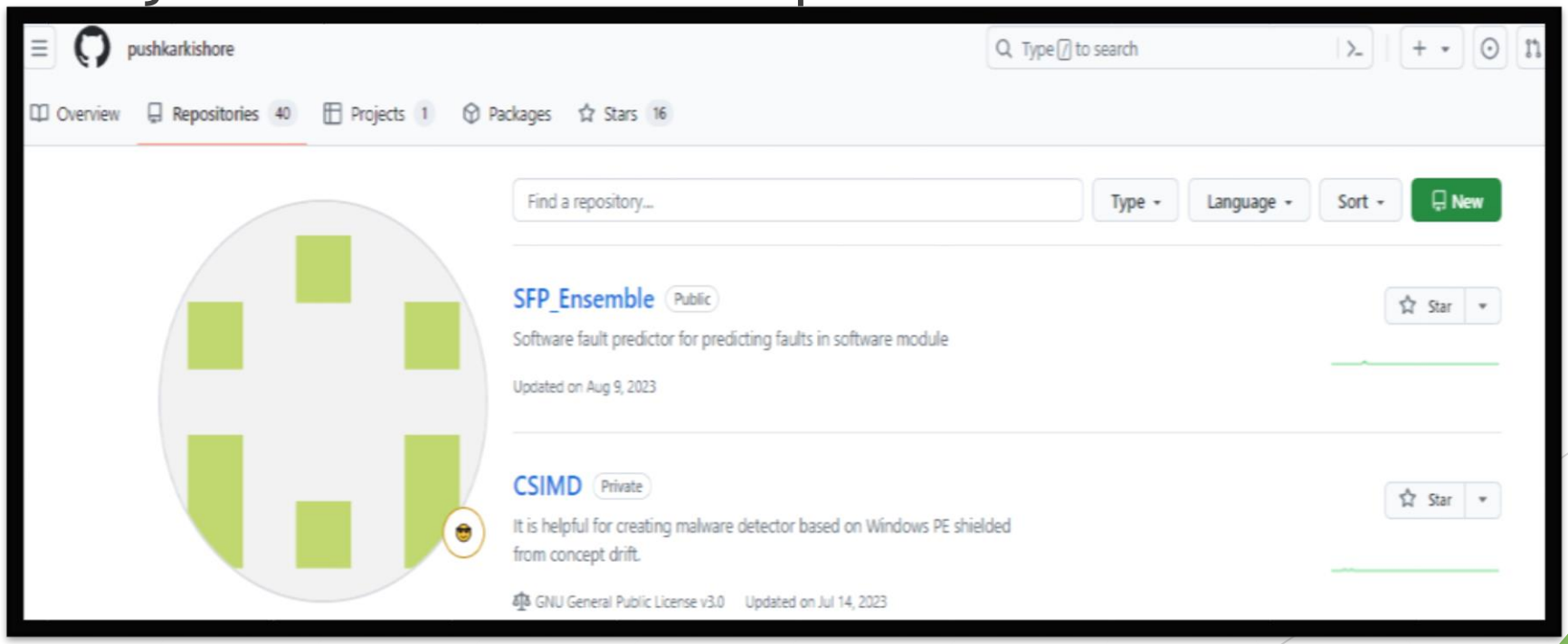
GITHUB ACCOUNT CREATION

- Go to https://github.com and click on signup.
- Enter your email, username and set a password.
- Phew, fill some basic details and you can land on personalized user homepage.
- Already signed up. Then, login with your credentials.

github
SOCIAL CODING

GITHUB PAGE OVERVIEW

- We can see username and number of contributions in the overview section.
- Projects are shown in the repositories section.



GITHUB ACCOUNT

- Create a new repository on GitHub: pushkar-demo
- Make our first commit.
- Click new button in repositories.
- Provide the repository name.
- Give description of the repository.
- *Public repository is visible to all whereas private repository is visible to yourself.*
- Add a readme file.

GITHUB ACCOUNT (Cont..)

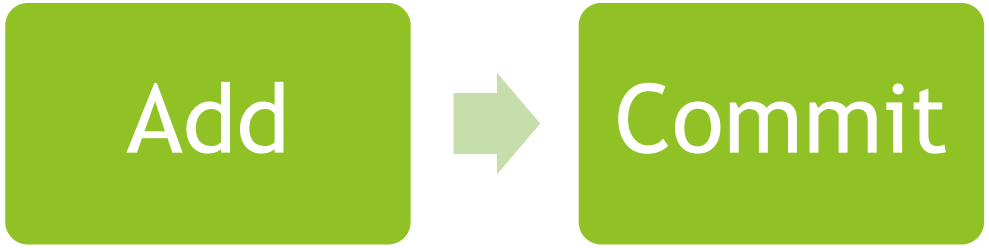
- *Readme contains details about your project such as name, how to use it, why this repo is created, which features are available.*
- *The default readme file contains the repository name and some basic instructions.*
- *The file format is 'md', which stands for Markdown documentation.*
- *It is a lightweight markup language that can be easily converted to text.*
- Click on create repository.

GITHUB ACCOUNT (Cont..)

- *Readme contains details about your project such as name, how to use it, why this repo is created, which features are available.*
- *The default readme file contains the repository name and some basic instructions.*
- *The file format is 'md', which stands for Markdown documentation.*
- *It is a lightweight markup language that can be easily converted to text.*
- Click on create repository.

GITHUB ACCOUNT (Cont..)

- Commit is used for confirming the changes in a code uploaded on GitHub.
- The first changes to the GitHub project is the *initial commit*.
- All commits are stored as history in the GitHub.



GITHUB README.MD

- We can use HTML to edit readme.md file.
- Example -

```
#pushkar-demo
This is my first repository.
<br>
Author – Pushkar kishore
```

SETTING UP GIT

- Visual studio code
- Windows (Git Bash)
- Mac (Terminal)
- Verify using *git --version*. It indicates the existing git version.
- Download git from https://git-scm.com/downloads

SETTING UP GIT (Cont..)

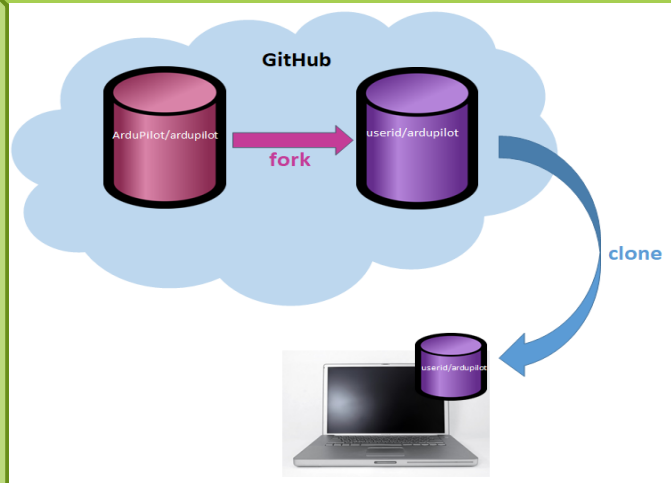
- Some informative things to watch out while installing git:
  - Select “on the desktop” option.
  - Select “override the default branch name for new repositories”.
- git config --global user.name “Pushkar Kishore”
  - Global indicates that we are connecting to GitHub account with “Pushkar Kishore” name.
  - Local is used if we want to connect with different GitHub account.

SETTING UP GIT (Cont..)

- git config –-global user.email “monumit46@gmail.com”
  - It sets the email.
- git config --list
  - It presents the configuration of git.

CLONE AND STATUS

- Local represents laptop, computer or mobile.
- Remote represents files existing on GitHub.



git clone <some link>

- Cloning a repository on our local machine

ls -a

- Show all hidden files

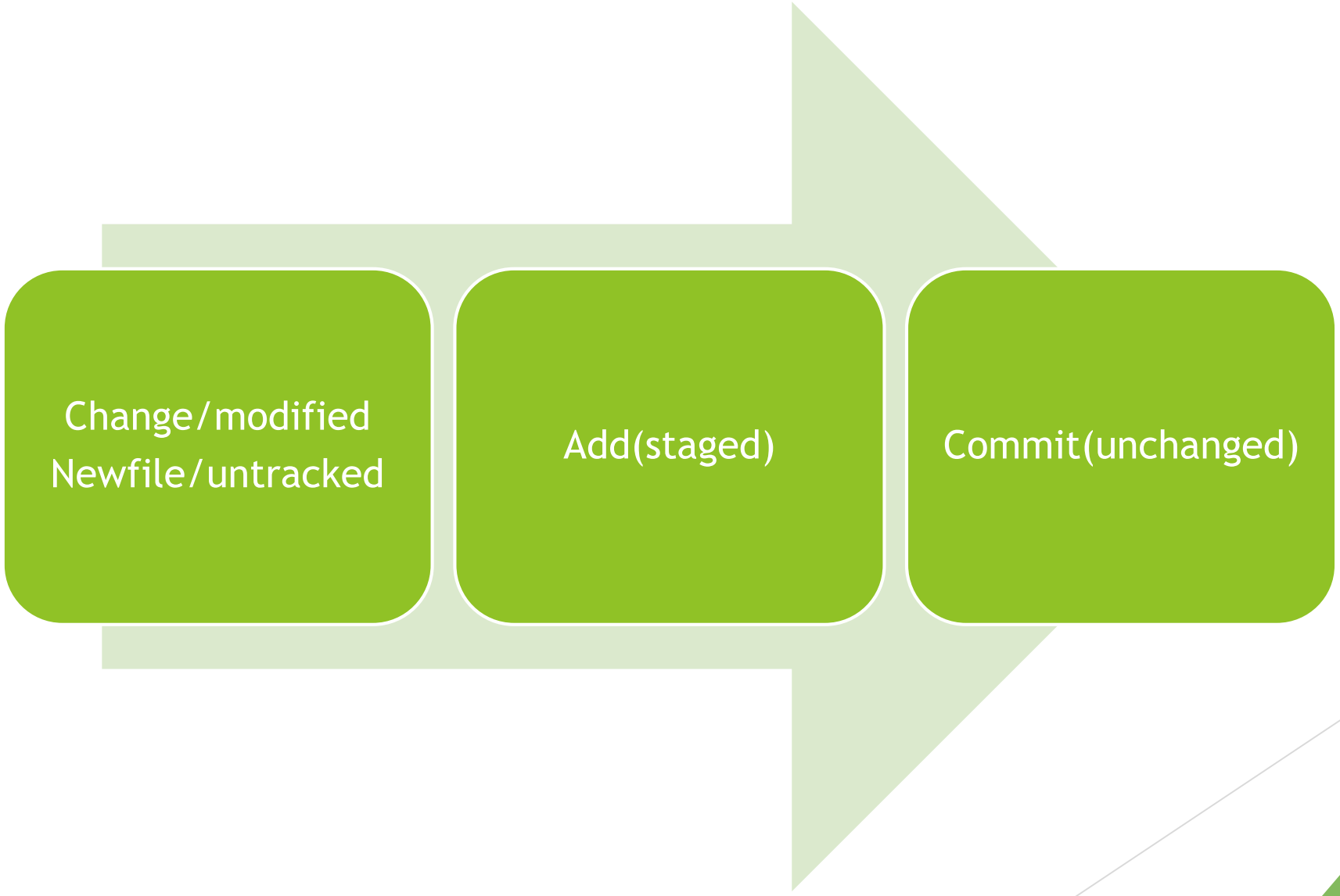
git status

- Displays the state of code

GIT STATUS INDICATORS

- Untracked
  - New files that GIT does not yet track
- Modified
  - Changed
- Staged
  - File is ready to be committed
- Unmodified
  - Unchanged

GIT STATUS INDICATORS (Cont..)



ADD & COMMIT

- git add <file_name> / git add .
  - Adds new or changed files in your working directory to the GIT staging area.
- Git commit –m “some message”
  - It is the record of change.

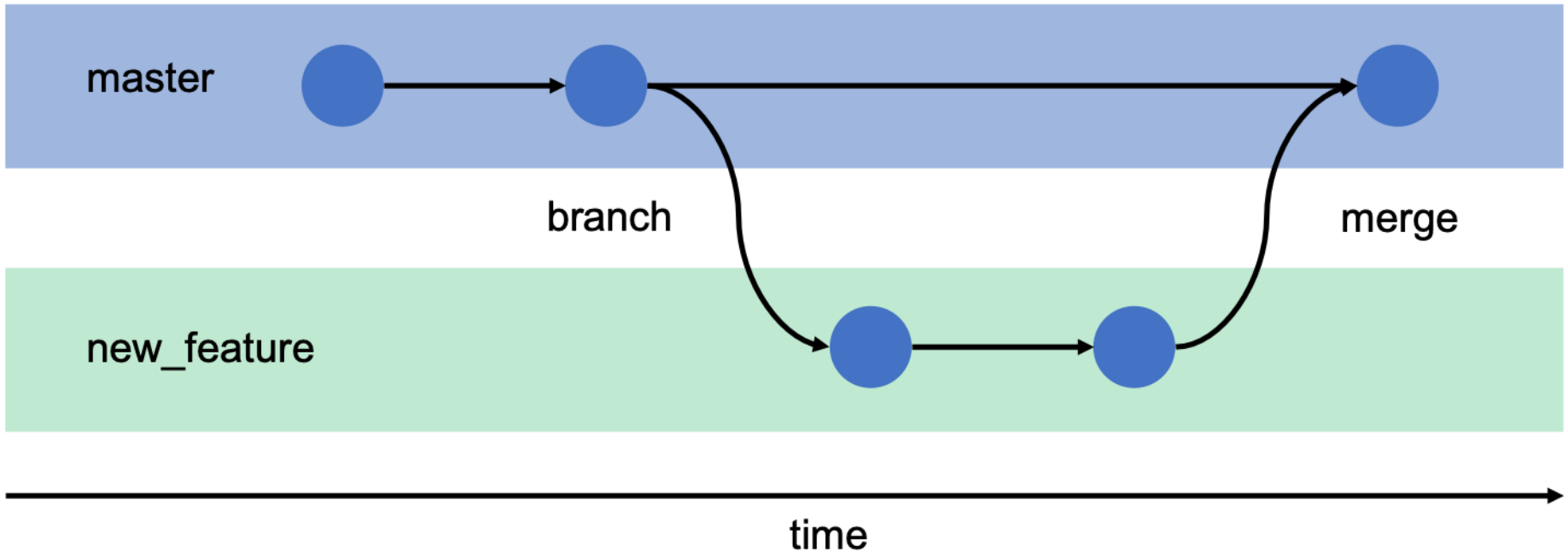
PUSH COMMAND

- git push origin main
- Push – upload local repo content to remote repo

INIT COMMAND

- init – used to create a new GIT repo
  - git init
  - git remote add origin <link>
  - git remote –v (verify remote)
  - git branch (to check branch)
  - git branch –M main (to rename branch)
  - git push -u origin main

GIT BRANCHES



BRANCH COMMANDS

- git branch (to check branch)
- git branch –m Main (rename branch)
- git checkout <branch name> (to navigate)
- git checkout –b <new branch name> (to create new branch)
- git branch –d <branch name> (to delete branch)

MERGING CODE

- Way-1
  - get diff <branch name> (to compare commits, branches, files and more)
  - git merge <branch name> (to merge 2 branches)
- Way 2
  - Create a Pull Request. Pull request tell others about changes you have pushed to a branch in a repository on GitHub.
  - git pull origin main (used to fetch and download content from a remote repo and immediately update the local repo to match that content),

RESOLVING MERGE CONFLICTS

- An event that takes place when GIT is unable to automatically resolve differences in code between two commits.
- Use PR
- Use GIT cmd and use VS autosuggestor.

UNDOING CHANGES

- Case1 (staged):
  - git reset <file name>
  - git reset
- Case 2:
  - Committed changes (for one commit)
  - git reset HEAD~1
- Case 3:
  - Committed changes (for many commits)
  - git reset <commit hash>
  - git reset --hard < commit hash >


You rock
BOOM!
ZAP!