

GIT IT RIGHT!

*A Hands-On Workshop on Git &
GitHub for Beginners*

Speaker: Vinayak Sharma, Samsung Research

ABOUT ME



- Final year B-Tech in **CS & AI @ IIIT-Delhi**
- SWE intern, **Samsung** Research
- Undergraduate researcher @ **Network and Systems Security Lab, IIIT-Delhi**
- Open Source maintainer @ **MemGPT**

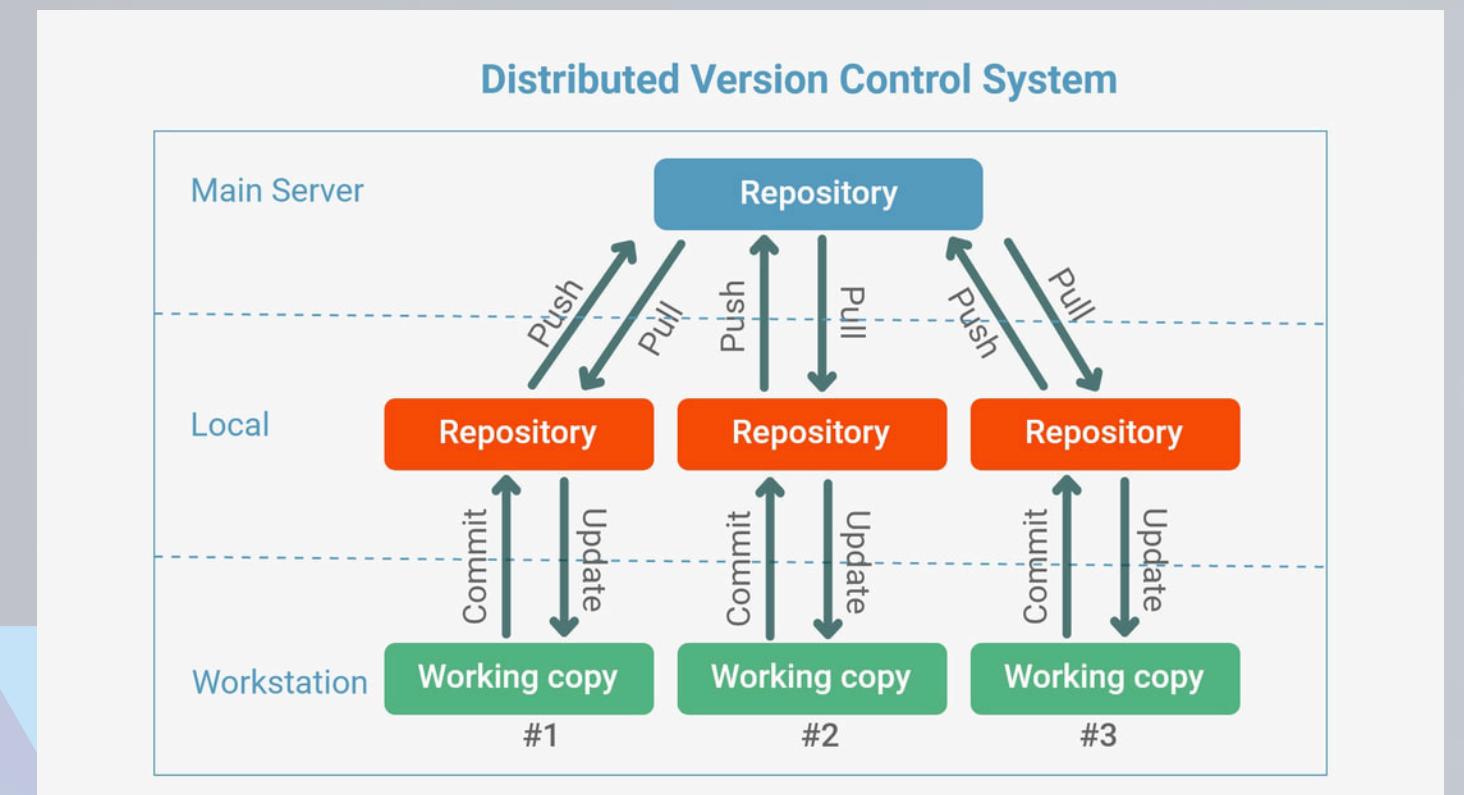
VERSION CONTROL SYSTEMS

#Demonstration: *File Comparison in Large Codebases*

Why Use VCS ?

- A Version Control System tracks changes to files over time.
- Enables **collaboration, recovery, and history tracking.**

GIT

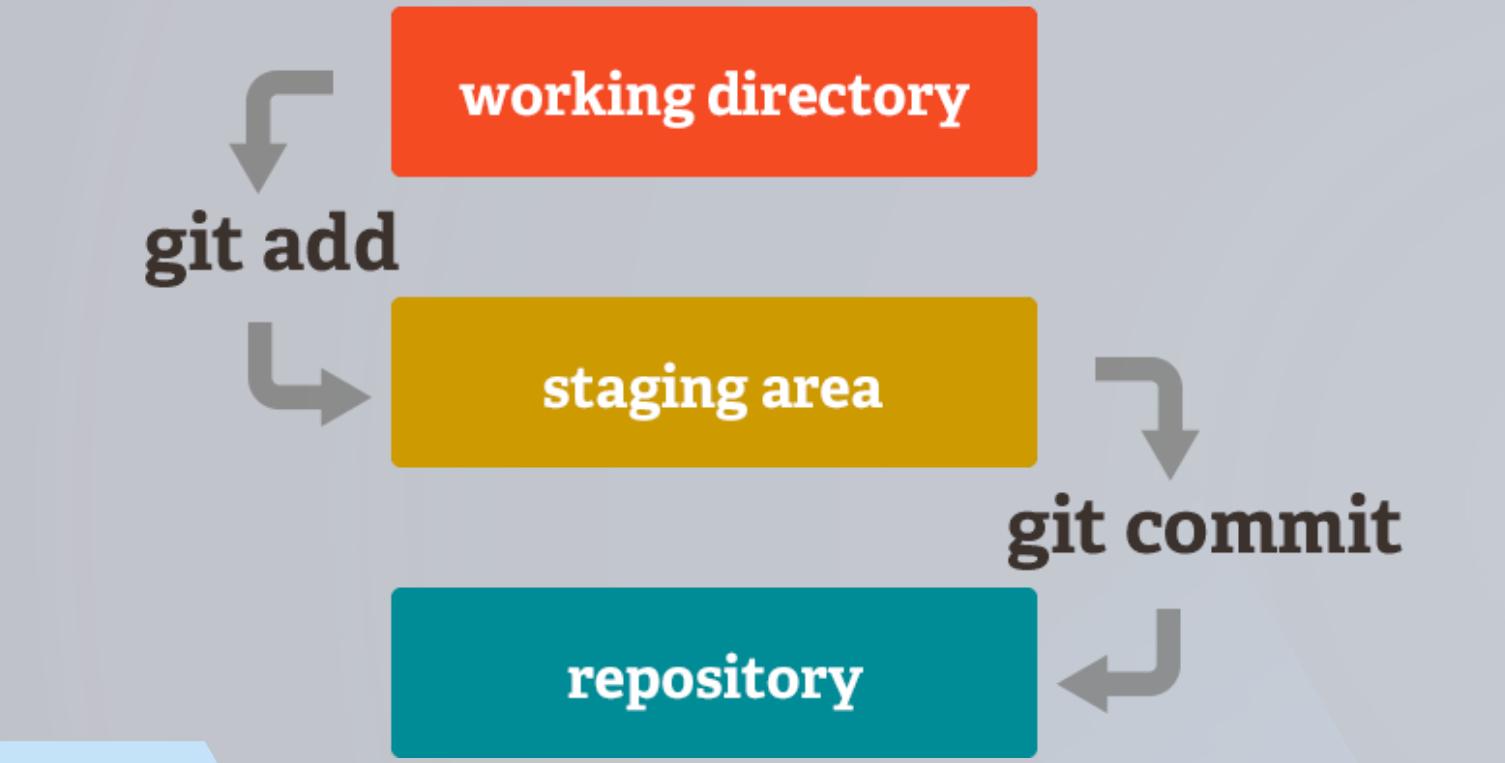


- Distributed VCS
- Ideal for large projects
- Industry standard

GIT

Advantages:

- **Distributed vcs:** Every developer has full history
- **Fast:** Local operations don't need internet or server access
- **Integrity:** SHA-1 hashing ensures file integrity and history accuracy



Think of staging like packing items into a box before sealing and shipping it.

Making Changes

- **Working Directory** – where you make changes.
- **Staging Area** – where you prepare changes for the next commit.
- **Repository** – where committed snapshots are stored permanently.

BASIC COMMANDS

- Start a Repository: `git init`
- Add & Commit Files: `git add filename`
- Commit Files: `git commit -m "Message"`
- Check Status and History: `git status ; git log`
- Compare Changes: `git diff`

#Demonstration: *Live usage*

GITHUB



GitHub

- **Microsoft-Owned Cloud Platform**
- Large developer community
- **Student developer pack** (more towards the end...)

GITHUB

- **Collaboration:** Teams can work on the same repo from anywhere.
- **Social Coding:** Follow developers, star repositories, and fork codebases.
- **Integrated DevOps:** Built-in Actions for CI/CD.

#Demonstration: *Github navigation; new repo*

GITHUB

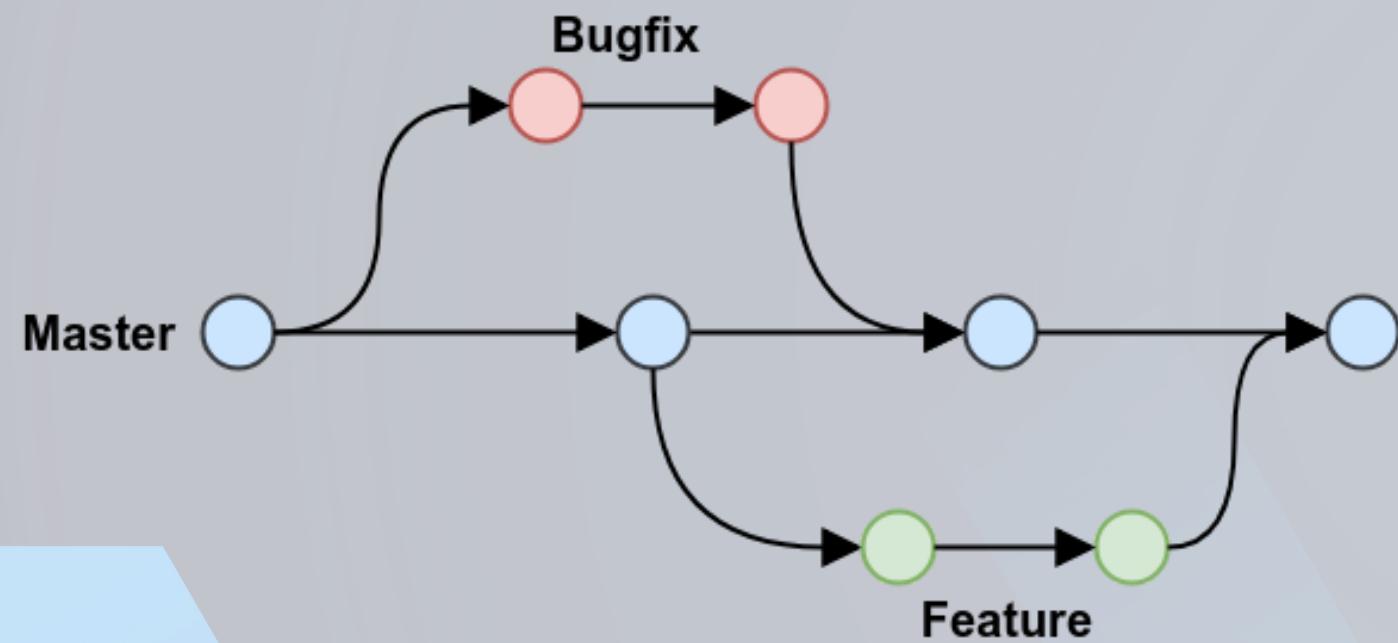
#Demonstration: *Integration of local and remote repository*

#Demonstration: *Pull, push commands ; ReadMe*

Good Practices:

- Maintain a README.md
- Write Descriptive Commit Messages
- **Protect Main Branch**, ie stable, production-ready branch

BRANCHES



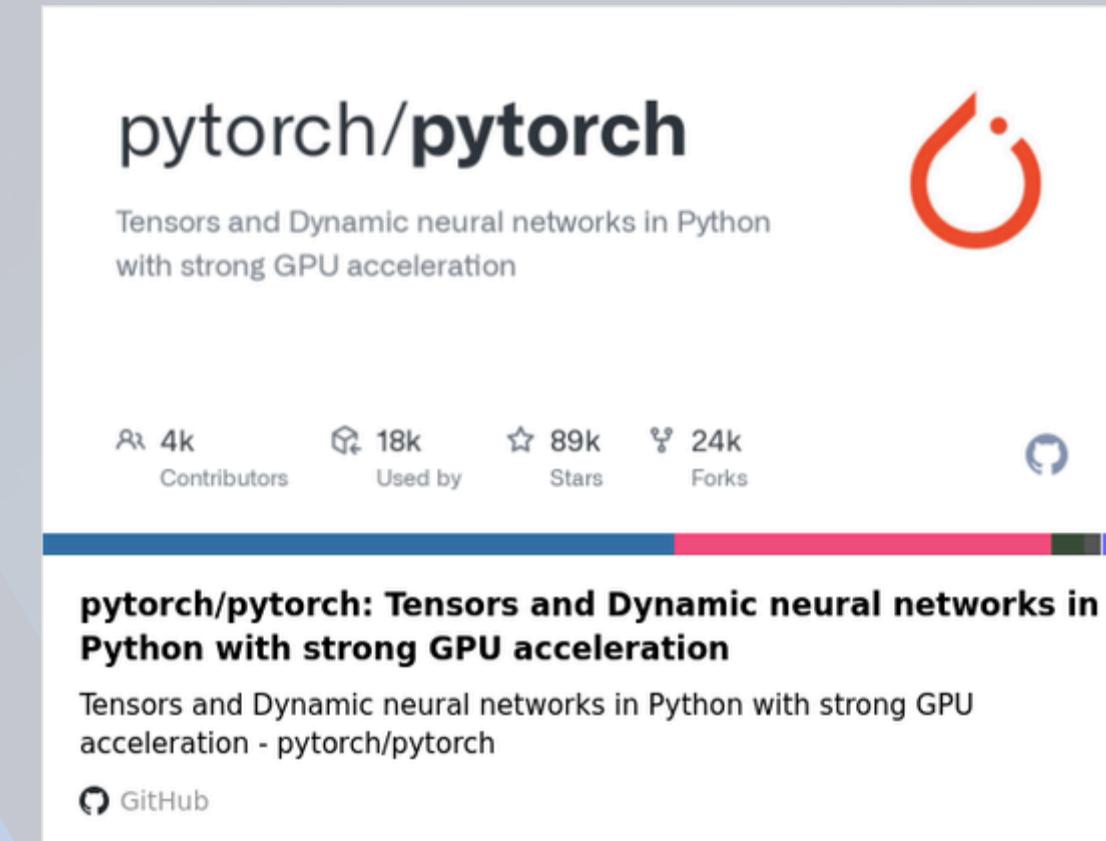
- **Isolate Features:** Work on new features/ bug fixes without affecting the main codebase.
- **Collaboration:** Multiple developers can work on separate branches

#Demonstration: *new branch, checkout*

GIT VS GITHUB

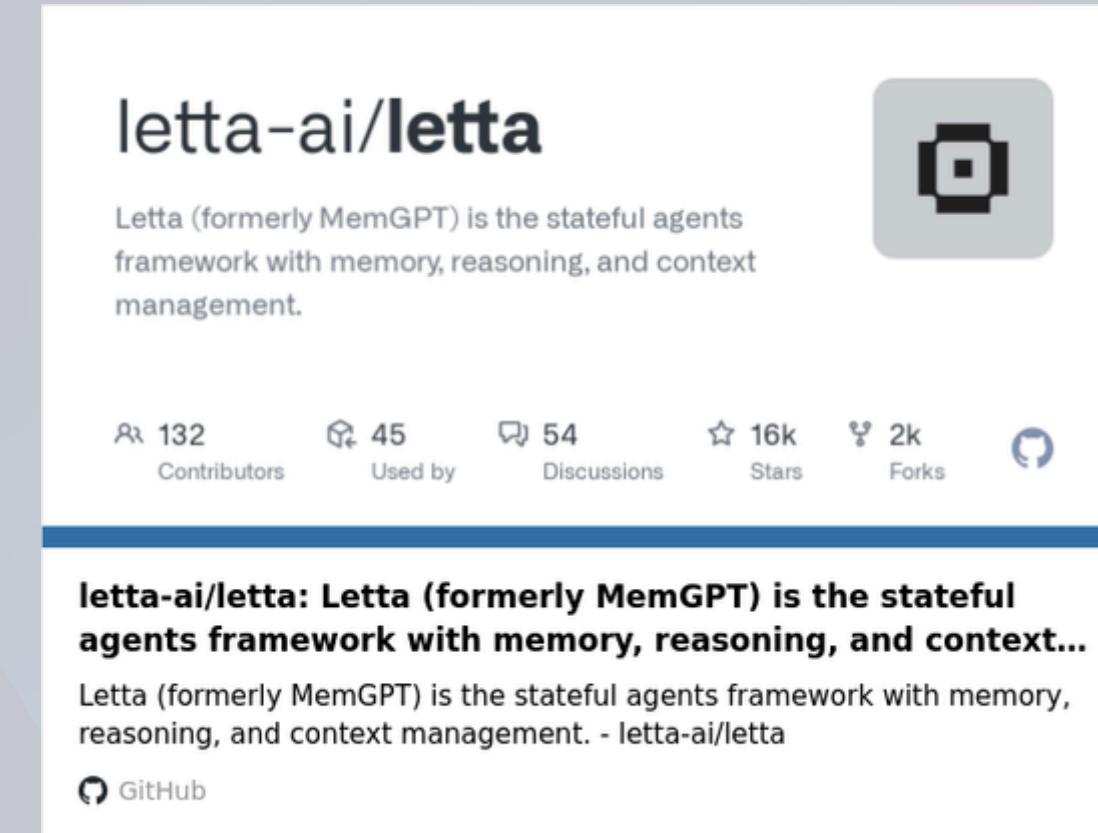


REAL INDUSTRY REPOS



#Demonstration(**pytorch**): *About, Insights*

MY OPEN SOURCE WORK



#Demonstration(letta-ai): Issues, pull requests, fork



GitHub

STUDENT DEVELOPER PACK

- Must be a **student**: You need a valid academic email
- **Active** GitHub account
- **Verification**: GitHub will verify your status

BENEFITS

- **GitHub Pro:** Unlimited private repositories, GitHub Actions, advanced code insights
- **Cloud Hosting & Storage Credits:** Azure, DigitalOcean, Google Cloud
- **Learning Resources & Courses**
- **Github Copilot**

#Demonstration: *Copilot demo*

REFERENCES

- <https://training.github.com/downloads/github-git-cheat-sheet.pdf>
- <https://githubtraining.github.io/training-manual/book.pdf>
- <https://www.interviewbit.com/blog/git-vs-github/>
- <https://www.shiksha.com/online-courses/articles/git-vs-github-whats-the-difference/>

THANK YOU