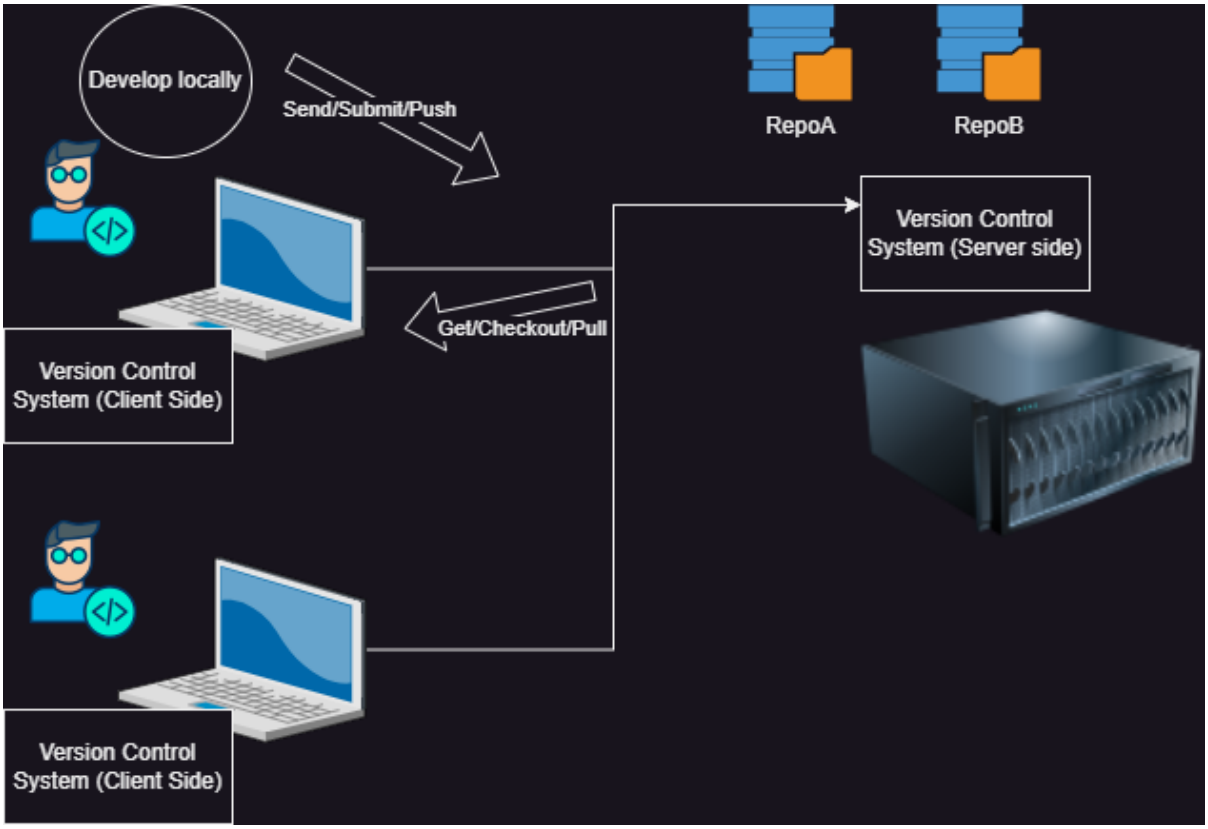
Version Control System

- Typically, Developers work on workstations where the client side version control system is installed and on the server side we have version control system (Server side components)
- Client Software
    - get changes
    - submit changes
- Server Side Software
    - Allow to create repositories
    - Allow to submit changes and recieve changes
    - User management
    - Network implementation (Which protocol)



- The above model is called as Client - Server Version Control System, Examples
    - Subversion
    - Clearcase
    - Team Foundation Version Control System

How Git Happened ?

Git was created in 2005 by Linus Torvalds, the creator of the Linux kernel, to manage the development of the Linux kernel after the proprietary BitKeeper VCS was no longer available for free use[1][2][3][4]. Torvalds wanted a distributed version control system that was fast, efficient, and able to handle large projects like the Linux kernel[1][3].

Some key goals in designing Git included:

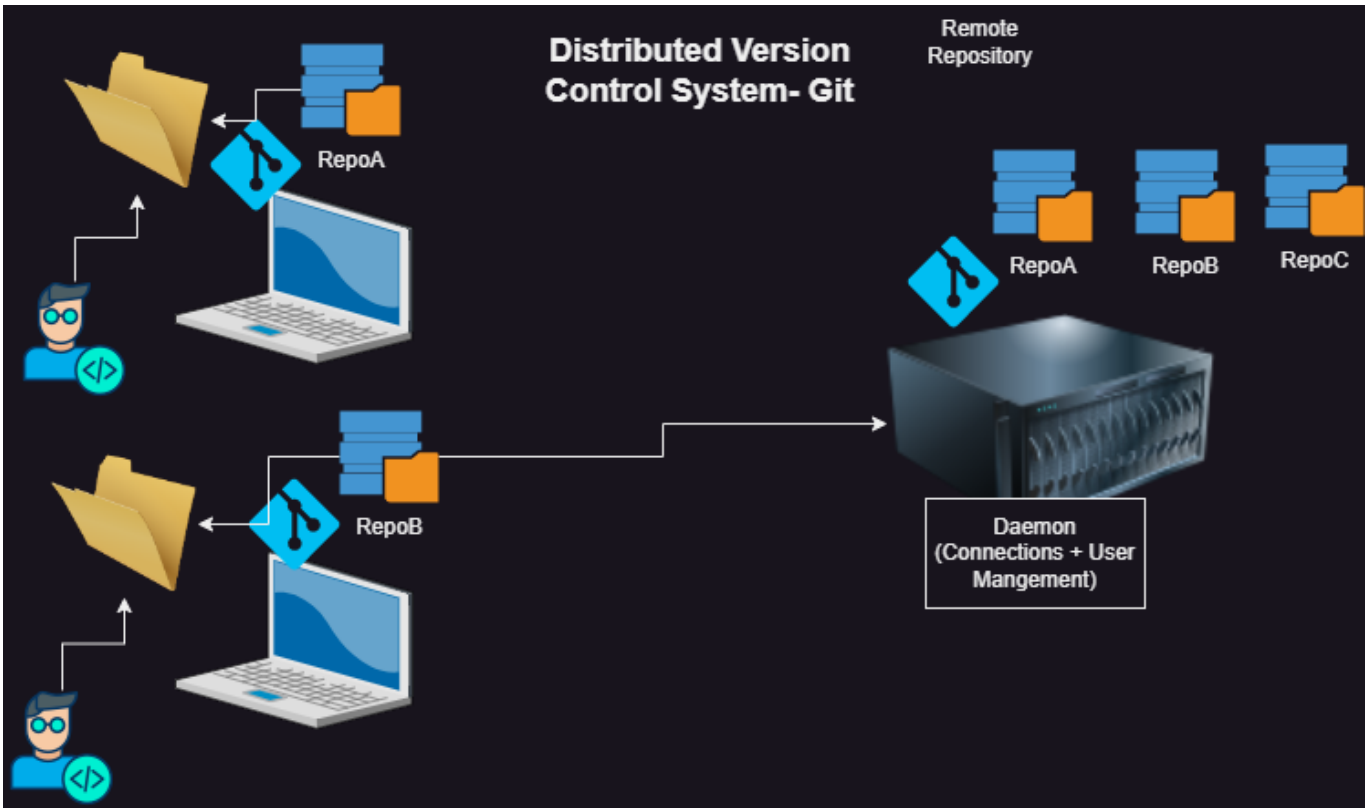
- **Speed**: The system needed to handle large repositories and numerous commits efficiently[1][2][3]
- **Distributed architecture**: Supporting distributed development, allowing each developer to have a complete copy of the repository[1][2][3]
- **Data integrity**: Ensuring the integrity and security of the repository's data was crucial[1][2][3]
- **Support for non-linear development**: Handling complex branching and merging workflows[1][3]

Torvalds started developing Git in April 2005 and released the first version within weeks[2][4]. Git 1.0 was released in December 2005, marking its readiness for broader use[2]. The launch of GitHub in 2008 revolutionized how developers collaborated on open-source projects and accelerated Git's adoption[2][4].

Today, Git is the de facto standard version control system, used by nearly 95% of developers as of 2022[3]. Its robust features, flexible workflow, and widespread adoption have made it indispensable for modern software development[2].

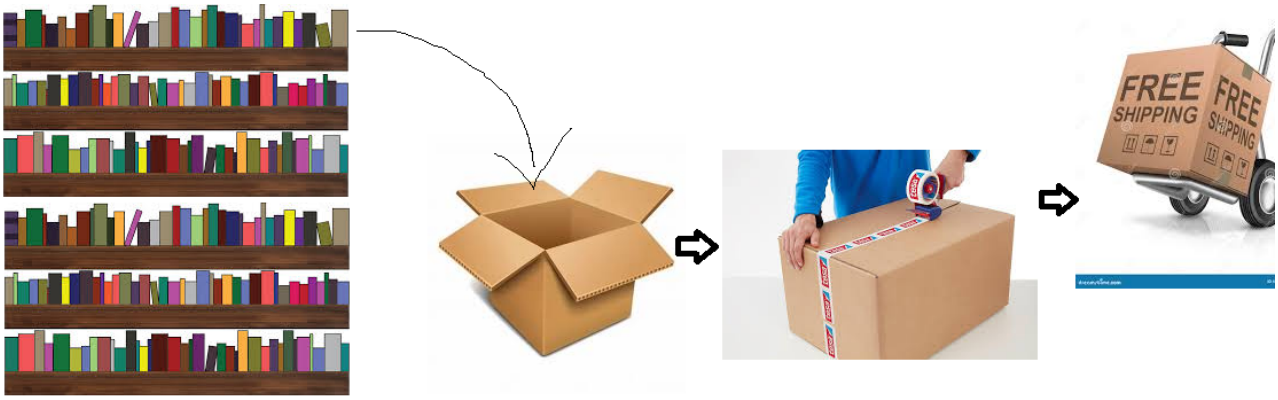
Citations: [1] https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git [2] https://www.geeksforgeeks.org/history-of-git/ [3] https://en.wikipedia.org/wiki/Git [4] https://community.appsmith.com/content/blog/evolution-git-dive-tech-history [5] https://www.linkedin.com/pulse/from-linus-github-fun-fascinating-history-git-michael-baker

Distributed Version Control System

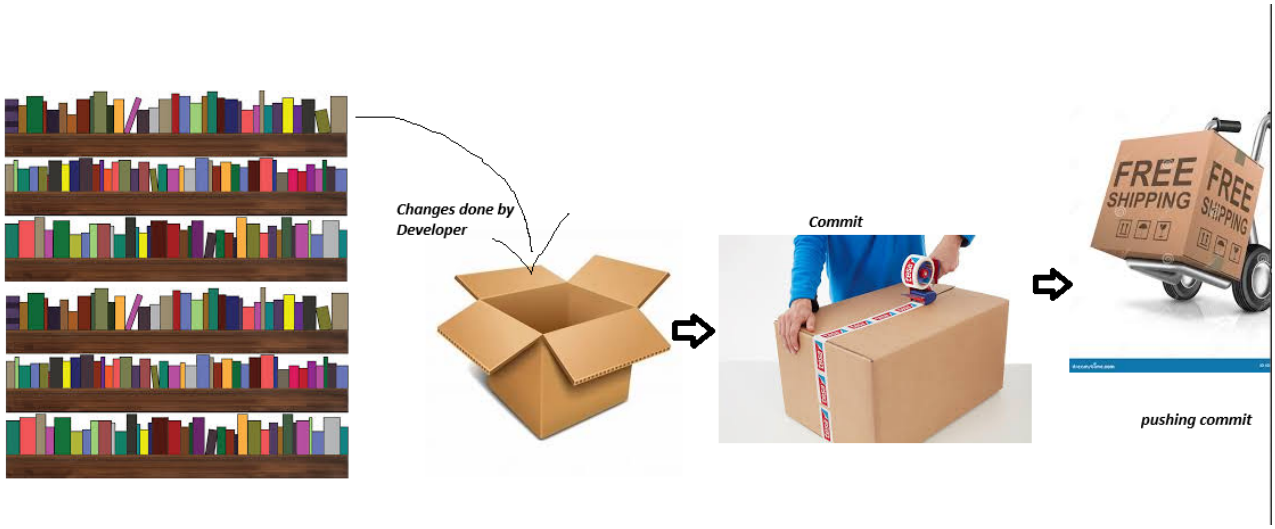


Simplifying how to work with Git Version Control System

- Generic Example

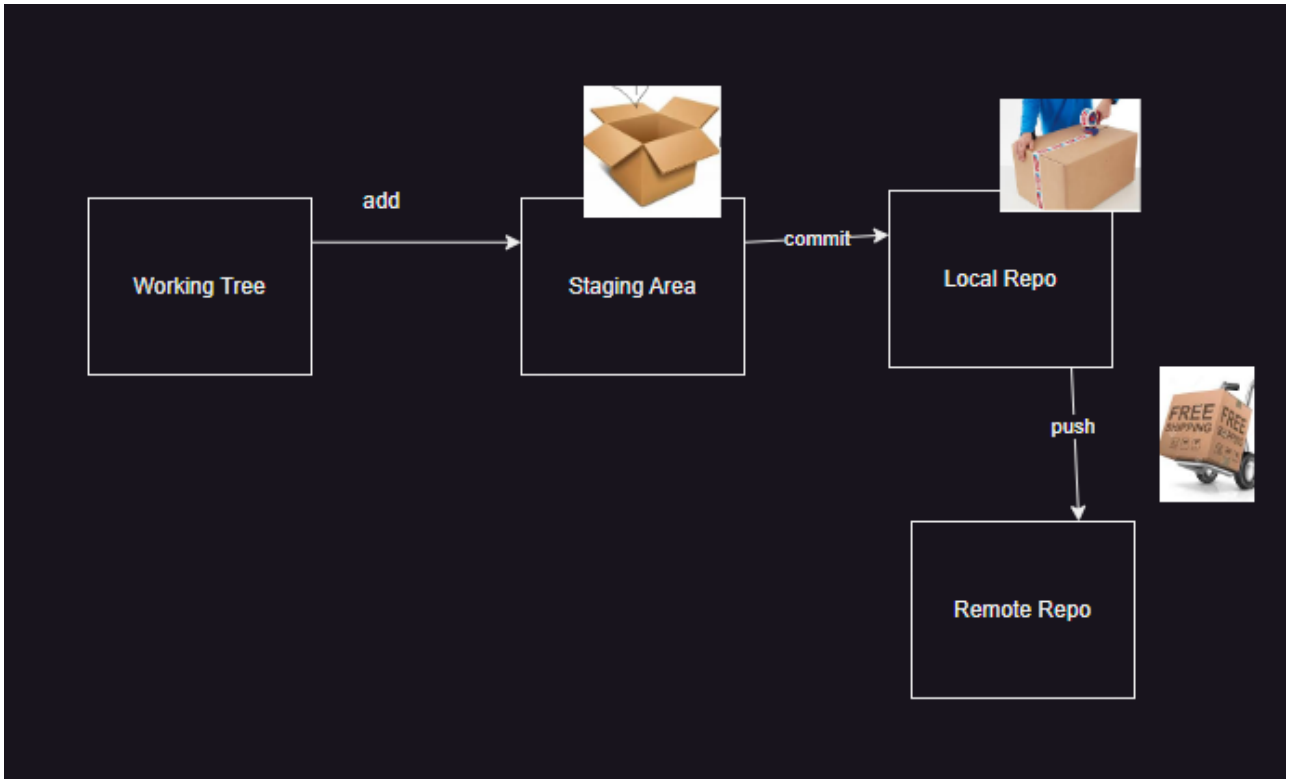


- Specific



Areas of Git

- Overview

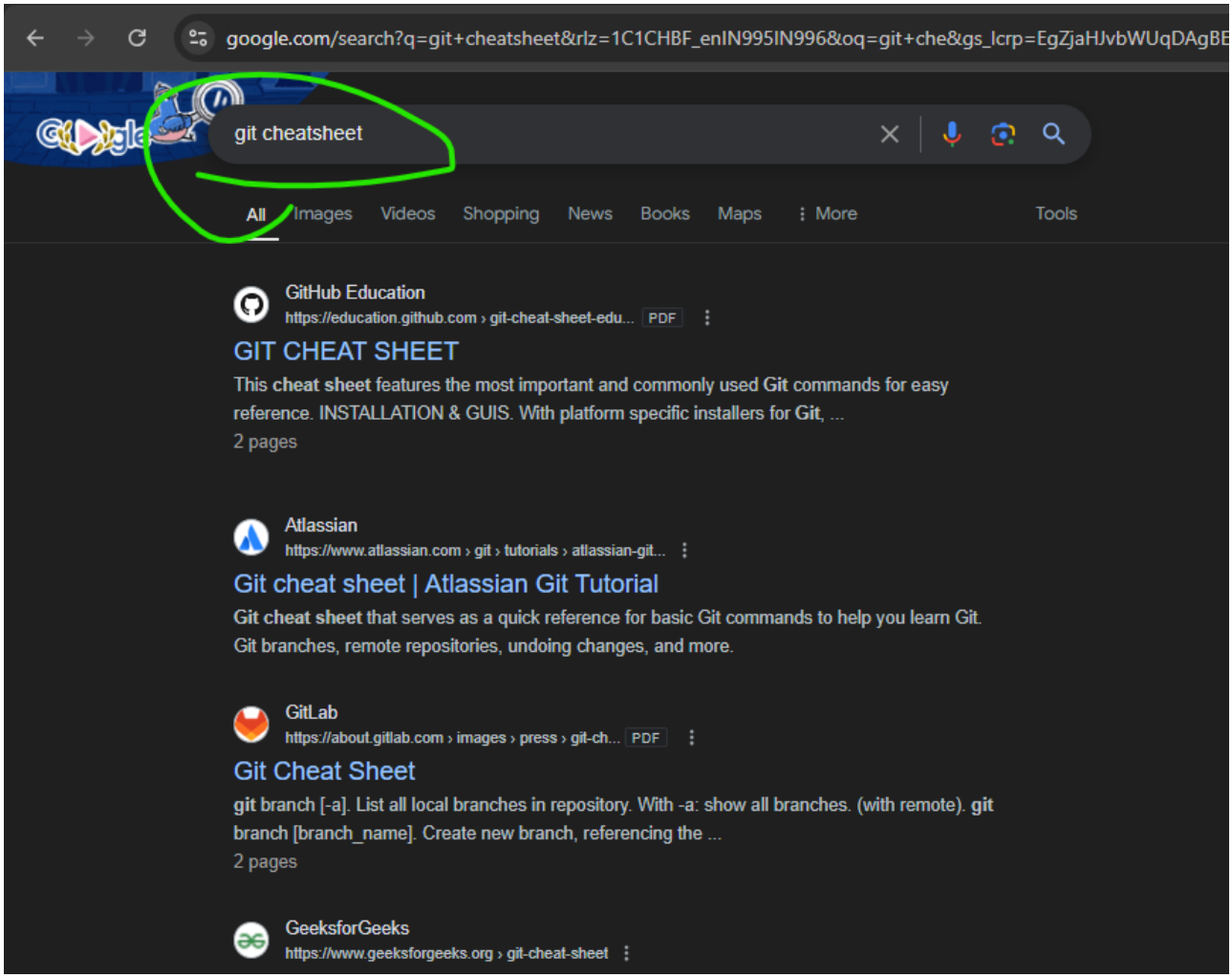


- Git has totally 5 areas, as of now we have tried to have 10,000 feet overview of 4 areas
    - Working Tree
    - Staging Area (Cache area)
    - local repo
    - Remote Repo

Learning Approach

1. Lets learn till local repo
    1. how to create a local repo
    2. how to make changes
    3. how to view/edit history
2. Then we will understand remote repo
    1. how to push
    2. how to pull
3. We will add other area

4. Note: donot remember commands use them (start with cheatsheets)



Hello git

- Assumption: Git is installed.
- Create a new directory and cd in to it

```
PS C:\temp\repos> mkdir hellogit


    Directory: C:\temp\repos


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         7/31/2024   8:19 AM                hellogit


PS C:\temp\repos> cd .\hellogit\
PS C:\temp\repos\hellogit> git init
Initialized empty Git repository in C:/temp/repos/hellogit/.git/
PS C:\temp\repos\hellogit>
```

- Now this creates a folder called as `.git`

- Now lets create one file `Readme.txt`

```
PS C:\temp\repos\hellogit> New-Item Readme.txt


    Directory: C:\temp\repos\hellogit


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/31/2024   8:23 AM              0 Readme.txt
```

- Lets ask git the status `git status`

```
PS C:\temp\repos\hellogit> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Readme.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\temp\repos\hellogit>
```

- Now add the changes

```
PS C:\temp\repos\hellogit> git add Readme.txt
PS C:\temp\repos\hellogit> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Readme.txt

PS C:\temp\repos\hellogit>
```

- Now lets commit, for this git needs your email id and username
- configuring email id and username is one time job (on new setups)

```
Windows PowerShell
PS C:\temp\repos\hellogit> git config --global user.name "shaikkhajaibrahim"
PS C:\temp\repos\hellogit> git config --global user.email "qtkhajadevops@gmail.com"
PS C:\temp\repos\hellogit>
```

- Now commit the changes

```
PS C:\temp\repos\hellogit> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Readme.txt

PS C:\temp\repos\hellogit> git commit -m "This is my first commit"
[master (root-commit) f9ff5b2] This is my first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Readme.txt
PS C:\temp\repos\hellogit> git status
On branch master
nothing to commit, working tree clean
PS C:\temp\repos\hellogit> git log
commit f9ff5b234136d59e702d11856525d7dc073083c0 (HEAD -> master)
Author: shaikkhajaibrahim <qtkhajadevops@gmail.com>
Date:   Wed Jul 31 08:29:56 2024 +0530

    This is my first commit
PS C:\temp\repos\hellogit>
```