GIT Tutorial

PM DIGI CLOUD

Zero to Hero

GIT vs GITHUB

GIT vs GITLAB

Features, Command and Workflow in Git

Who Am I?

- IT work Experience since 2008
- Certified Azure Architect
- SCJP
- B.TECH (Computer Science)
- PGDAC – CDAC, Bangalore

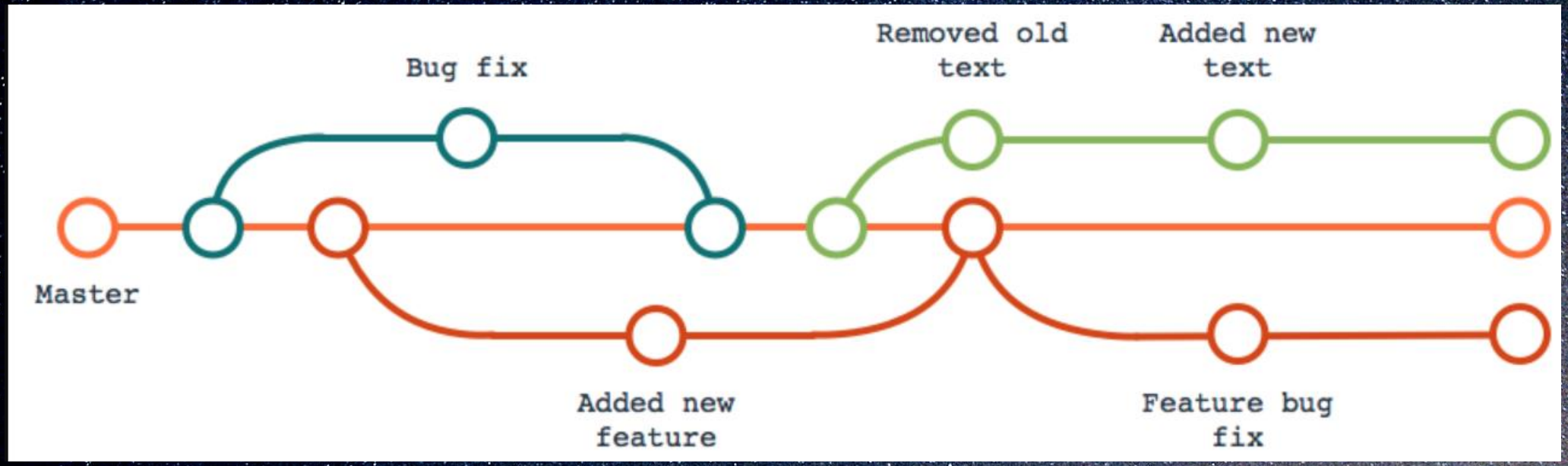
PRASHANT MALVIYA

What is GIT?

PM DIGI CLOUD

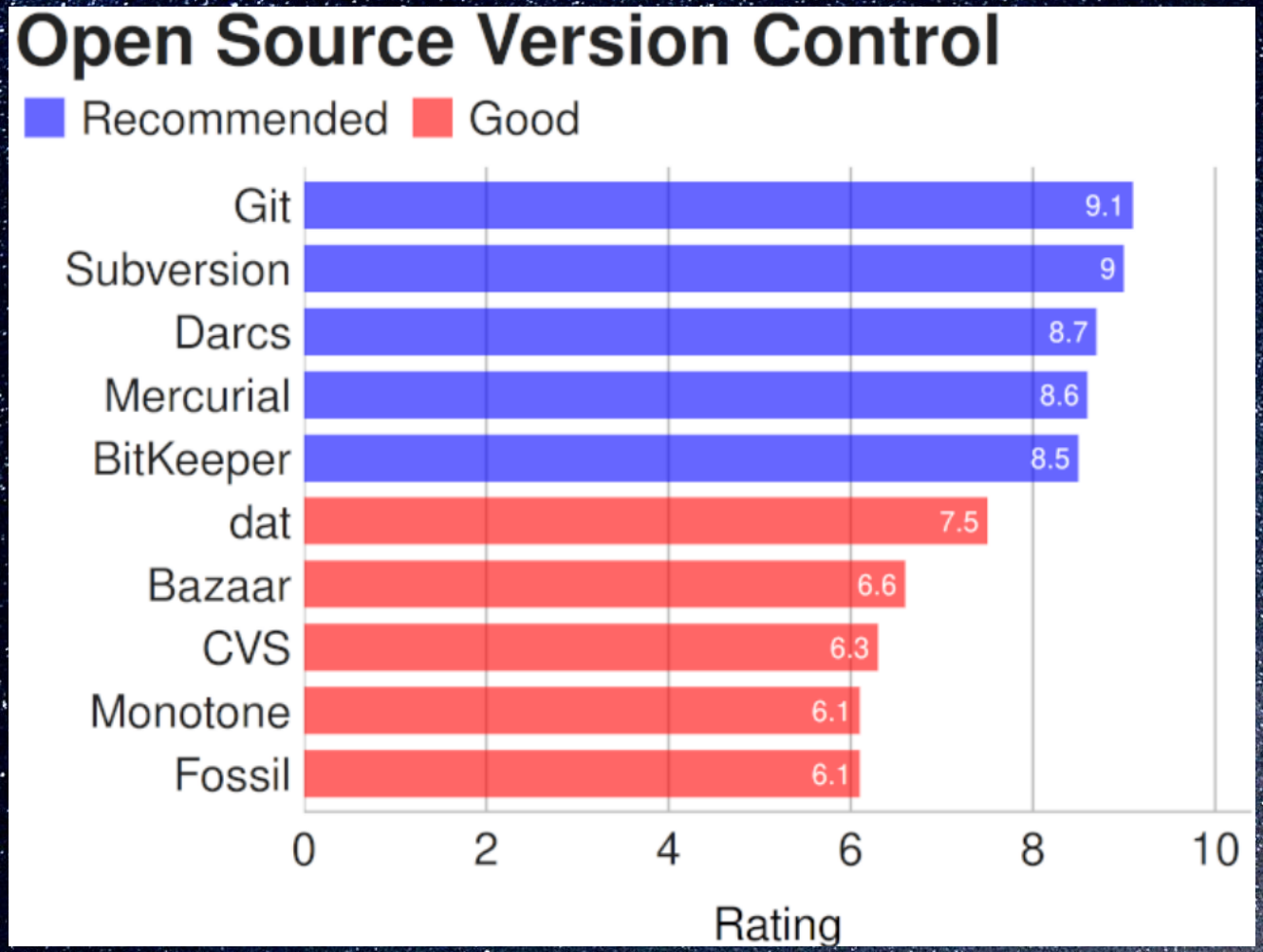
Git is a tool for document (preferred for source code) version management

Git repository refers to a place where all the Git files are stored. These files can either be stored on the local repository or on the remote repository.



Version controls open source

PM DIGI CLOUD



https://en.wikipedia.org/wiki/List_of_version-control_software

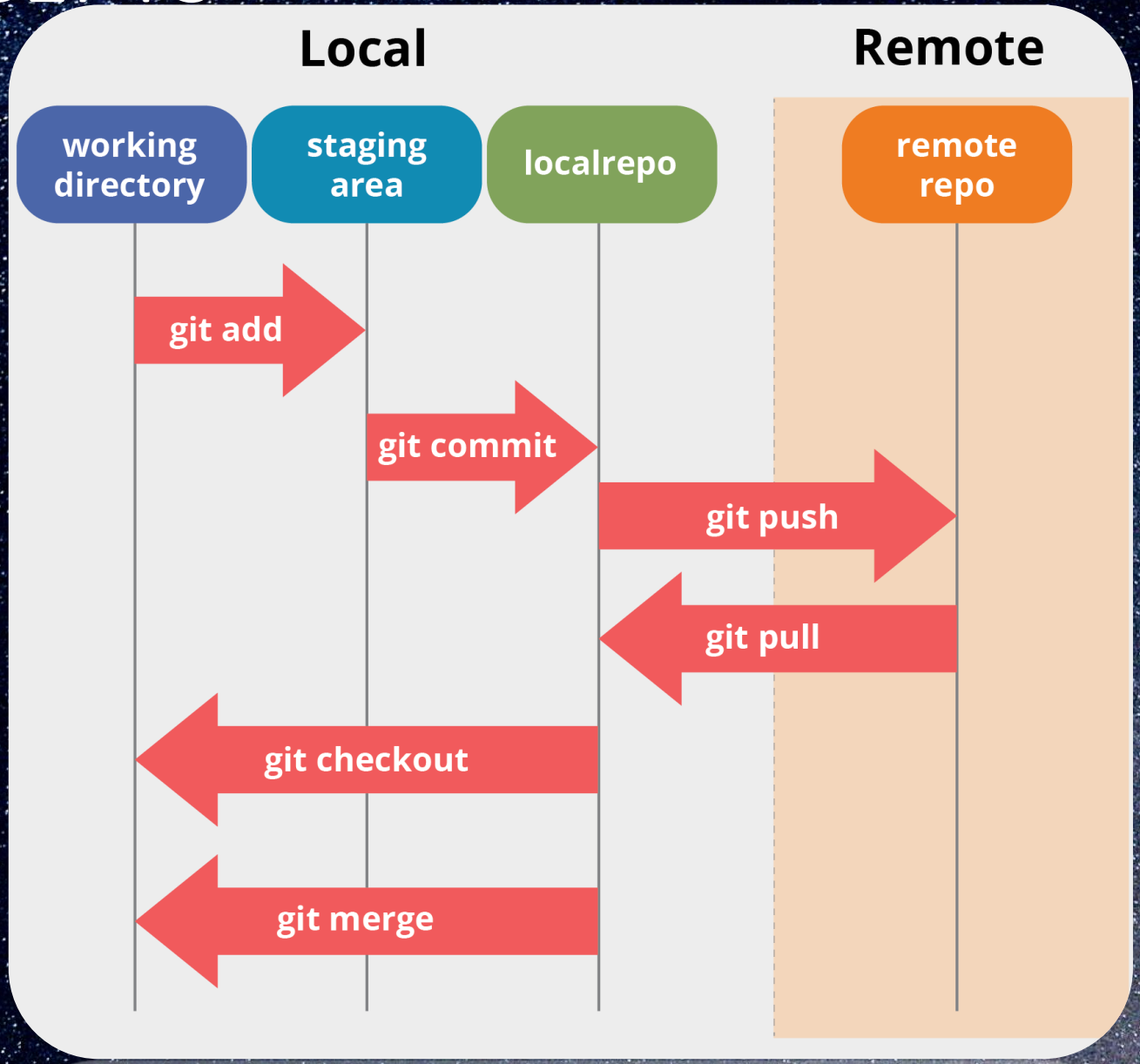
Git VS SVN

PM DIGI CLOUD

GIT	SVN
Git is a distributed decentralized version control system	SVN is a centralized version control system.
Git stores content in the form of metadata.	SVN stored data in the form of files.
The master contains the latest stable release.	In SVN, the trunk directory has the latest stable release
The contents of Git are hashed using the SHA-1 hash algorithm.	SVN doesn’t support hashed contents.

GIT CONCEPTS

PM DIGI CLOUD



GITHUB

PM DIGI CLOUD

Think of Git as a single computer and GitHub as a network of multiple interconnected computers
At its core, Git is a free, open-source software distributed version control system (DVCS) designed to manage all source code history.
GitHub, on the other hand, is a web-based hosting service for Git repositories. It offers all of Git's DVCS SCM and has some additional features.

Git is installed locally on a system, so developers can manage their source code history using their local machines as repositories. This means there is no centralized server required to use Git, and no needed internet access either. Additionally, there's no user-management system available and a proprietary desktop GUI.

GitHub, meanwhile, lives in the cloud, so Internet access is required. It also has a built-in user-management system and a user-friendly GUI. In addition to its main website, GitHub features a desktop version that can be installed on local computers to help synchronize code. It should be noted that Git can be used without GitHub, but GitHub cannot be used without Git. (GitHub primarily was built to work correctly with Git.)

While there are some advantages of Git as a DVCS, it does have some significant competition. This includes Mercurial, IBM, Subversion, and ClearCase. GitHub's competition is wildly different due to its software-as-a-service (SaaS) focus and includes companies like GitLab and Bitbucket.

GITLAB

PM DIGI CLOUD

- GitLab is one of America's fastest-growing private software companies, posting huge gains in 2021. The company provides a central server that manages Git repositories and is used to simplify the administration tasks of many corporations worldwide. According to Wikipedia, GitLab has over 100,000 users and is used by large, well-known organizations such as IBM, Sony, Goldman Sachs, and NASA.

- GitLab is a web-based Git repository that provides free open and private repositories, issue-following capabilities, and wikis. It is a complete DevOps platform that enables professionals to perform all the tasks in a project—from project planning and source code management to monitoring and security. Additionally, it allows teams to collaborate and build better software.

GIT Essential Commands -1

PM DIGI CLOUD

Git clone

Git clone is a command for downloading existing source code from a remote repository (like Github, for example).

In other words, Git clone basically makes an identical copy of the latest version of a project in a repository and saves it to your computer.

There are a couple of ways to download the source code, but mostly everyone prefer the **clone with https** way:

```
git clone <https://name-of-the-repository-link>
```

GIT Essential Commands-2

PM DIGI CLOUD

Git branch

By using branches, several developers are able to work in parallel on the same project simultaneously.

We can use the git branch command for creating, listing and deleting branches.
Creating a new branch: git branch <branch-name>

This command will create a branch locally. To push the new branch into the remote repository, you need to use the following command:
git push -u <remote> <branch-name>

Viewing branches : git branch or git branch --list
Deleting a branch: git branch –d <branch-name>

GIT Essential Commands-3

PM DIGI CLOUD

Git Checkout

To work in a branch, first you need to switch to it. We use git checkout mostly for switching from one branch to another. We can also use it for checking out files and commits.

git checkout <name-of-your-branch>

The changes in your current branch must be committed or stashed before you switch
The branch you want to check out should exist in your local

git checkout -b <name-of-your-branch>
This command creates a new branch in your local (-b stands for branch) and checks the branch out to new right after it has been created.

GIT Essential Commands-4

PM DIGI CLOUD

Git Status

The Git status command gives us all the necessary information about the current branch. .

- git status

Whether the current branch is up to date
Whether there is anything to commit, push or pull
Whether there are files staged, unstaged or untracked
Whether there are files created, modified or deleted

GIT Essential Commands-5

PM DIGI CLOUD

Git add

When we create, modify or delete a file, these changes will happen in our local and won't be included in the next commit (unless we change the configurations).

The unstaged files won't be included in your commits.

We need to use the git add command to include the changes of a file(s) into our next commit.

git add <file> (To Add Single file)
git add –A (To Add everything at once)

Important: The git add command doesn't change the repository and the changes are not saved until we use git commit.

GIT Essential Commands-6

PM DIGI CLOUD

Git Commit

This is maybe the most-used command of Git. Once we reach a certain point in development, we want to save our changes (maybe after a specific task or issue).

Git commit is like setting a checkpoint in the development process which you can go back to later if needed.

We also need to write a short message to explain what we have developed or changed in the source code.

```
git commit -m "commit message"
```

Important: Git commit saves your changes only locally.

GIT Essential Commands-7

PM DIGI CLOUD

Git Push

After committing your changes, the next thing you want to do is send your changes to the remote server. Git push uploads your commits to the remote repository.

git push <remote> <branch-name>

However, if your branch is newly created, then you also need to upload the branch with the following command:
git push --set-upstream <remote> <name-of-your-branch>
Or
git push -u origin <branch_name>

Important: Git push only uploads changes that are committed.

GIT Essential Commands-8

Git Pull

The git pull command is used to get updates from the remote repo.

This command is a combination of git fetch and git merge which means that, when we use git pull, it gets the updates from remote repository (git fetch) and immediately applies the latest changes in your local (git merge).

git pull <remote>

Important: This operation may cause conflicts that you need to solve manually.

GIT Essential Commands-9

PM DIGI CLOUD

Git log

The git log command displays all of the commits in a repository's history.

By default, the command displays each commit's:

Secure Hash Algorithm (SHA)
author
date
commit message

Git log
Git log --oneline
Git log –stat
Git log <specific commit hashcode>
git log --graph

GIT Essential Commands-10

PM DIGI CLOUD

Git revert

Sometimes we need to undo the changes that we've made. There are various ways to undo our changes locally or remotely (depends on what we need), but we must carefully use these commands to avoid unwanted deletions.

A safer way that we can undo our commits is by using git revert.

use git log to see the hashcode of commit

git revert hashcode

Important: The Git revert command will undo the given commit, but will create a new commit without deleting the older one.

GIT Essential Commands-11

PM DIGI CLOUD

Git merge

When you've completed development in your branch and everything works fine, the final step is merging the branch with the parent branch (dev or master). This is done with the git merge command.

Code review option also in GitHub.

First you should switch to the dev branch:

git checkout dev

Before merging, you should update your local dev branch:

git fetch

Finally, you can merge your feature branch into dev:

git merge <branch-name>

Important: Make sure your dev branch has the latest version before you merge your branches, otherwise you may face conflicts or other unwanted problems.

GIT Essential Commands-12

PM DIGI CLOUD

Git stash

The git stash command takes your modified tracked files and saves it on a pile of incomplete changes that you can reapply at any time. To go back to work, you can use the stash pop.
The git stash command will help a developer switch branches to work on something else without committing to incomplete work.

```
git stash –u # Store current work with untracked files
git stash pop # Bring stashed work back to the working directory
```

THANK YOU!

pmdigicloud@gmail.com

malviya.prashant@outlook.in