**. GIT**

****

**Content**

**1. Introduction**

**2. Version Control**

**3. Advantages of Git**

**4. Life Cycle**

**5. How Git Works**

**6. Environment Setup**

**7. Git Commands**

1. **Introduction**

* Git is a free and open source distributed code management and Version control system that is distributed under the GNU General Public License version 2.
* Git was initially designed and developed by Linus Torvalds for Linux kernel development.

1. **What is version control?**

* Version control, also known as source control, is the practice of tracking and managing changes to software code.
* Version control systems are software tools that help software teams manage changes to source code over time.

1. **Advantages of Git**

**1. Free and open source -**  It is available freely over the internet.  As it is an open source, you can download its source code and also perform changes according to your requirements.

**2. Fast and small -** As most of the operations are performed locally, it gives a huge benefit in terms of speed. Git does not rely on the central server; that is why, there is no need to interact with the remote server for every operation. The core part of Git is written in C .

**3. Implicit backup -** The chances of losing data are very rare when there are multiple copies of it. Data present on any client side mirrors the repository, hence it can be used in the event of a crash or disk corruption.

**4. Security -** Git uses a common cryptographic hash function called secure hash function (SHA1), to name and identify objects within its database.

# 4. Git - Life Cycle

General workflow is as follows −

* You clone the Git repository as a working copy.
* You modify the working copy by adding/editing files.
* If necessary, you also update the working copy by taking other developer's changes.
* You review the changes before commit.
* You commit changes. If everything is fine, then you push the changes to the repository.
* After committing, if you realize something is wrong, then you correct the last commit and push the changes to the repository.



* 1. **Environment Setup**
* From below link you can download Git on your desktop

Windows: [http://www.git-scm.com](http://www.git-scm.com/)

**Customize Git Environment**

Git provides the git config tool, which allows you to set configuration variables. Git stores all global configurations in **.gitconfig file**, which is located in your home directory. To set these configuration values as global, add the --global option, and if you omit --global option, then your configurations are specific for the current Git repository.

1. **Git Commands**

1. **git config**

**Usage: git config –global user.name “[name]”**

**Usage: git config –global user.email “[email address]”**

This command sets the author name and email address respectively to be used with your commits.

Git Config Command - Git Commands - Edureka

**2. git init**

**Usage: git init [repository name]**

This command is used to start a new repository.

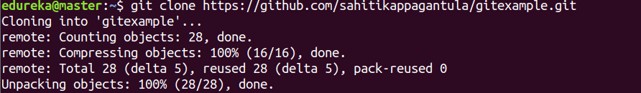
GitInit Command - Git Commands - Edureka

1. **git clone**

**Usage: git clone [url]**

This command is used to obtain a repository

from an existing URL.

****

1. **git add**

**Usage: git add [file]**

This command adds a file to the staging area.

Git Add Command - Git Commands - Edureka

**Usage: git add \***

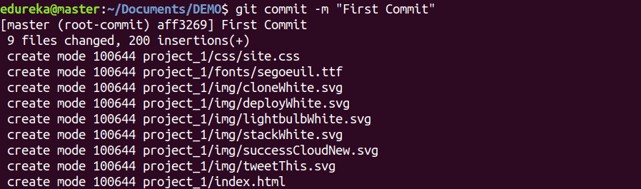
This command adds one or more to the staging area.

Git Add Command - Git Commands - Edureka

1. **git commit**

**Usage: git commit -m “[ Type in the commit message]”**

This command records or snapshots the file permanently in the version history.



**Usage: git commit -a**

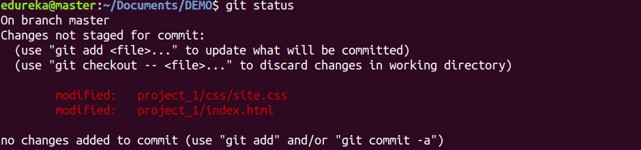
This command commits any files you’ve added with the git add command and also commits any files you’ve changed since then.

Git Commit Command - Git Commands - Edureka

1. **git status**

**Usage: git status**

This command lists all the files that have to be committed.



1. **git branch**

**Usage: git branch**

This command lists all the local branches in the current repository.

Git Branch Command - Git Commands - Edureka

**Usage: git branch [branch name]**

This command creates a new branch.

Git Branch Command - Git Commands - Edureka

**Usage: git branch -d [branch name]**

This command deletes the feature branch.

Git Branch Command - Git Commands - Edureka

1. **git push**

**Usage: git push [variable name] master**

This command sends the committed changes of master branch to your remote repository.

