**GIT**

**Source Code Management:**

Source code management (SCM) is used to manage and track modifications to a source code repository. SCM tracks a running history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors. SCM is also synonymous with Version control.

As software projects grow in lines of code and contributor head count, the costs of communication overhead and management complexity also grow. SCM is a critical tool to alleviate the organizational strain of growing development costs.

**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. As development environments have accelerated, version control systems help software teams work faster and smarter. They are especially useful for DevOps teams since they help them to reduce development time and increase successful deployments.

**Difference between source code management and version control:**

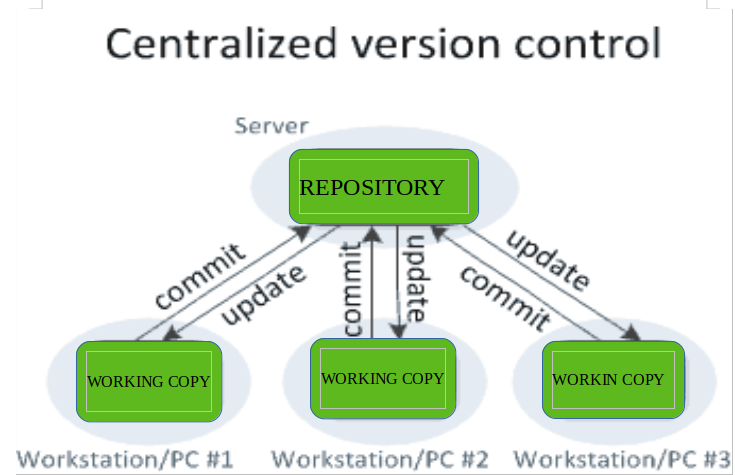
There is no difference between source code management and version control system.

**Types of Version Control System:**

There are 2 types of version control system they are:

1. Centralized Version Control System
2. Distributed Version control System

**Centralized Version Control System:**



**Example:**

Microsoft Teams foundation server, SVN.

**Disadvantage of centralized version control system:**

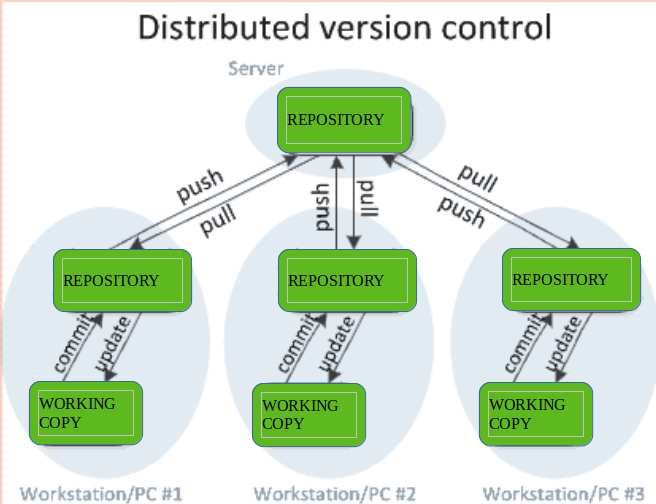
When the system is down there will be no back up copy (Single point of failure).

We must be connected via internet (mandatory).

**Advantage of centralized version control system:**

Used for small size of project.

**Distributed Version Control System:**



We will have the backup and current copy in the local repository even though the central repository is down. Offline working is possible in distributed version control system.

Local repository:

1. GIT (Distributed version control system) it is a cloud-based repository
2. Bit Bucket it is a cloud- based repository

**Example:**

Mercurial (Distributed version control system)

**GIT:**

Open-source distributed version control system. Developed by Torvalds Linux.

**Features of GIT:**

1. Scalability (modify it according to our team size and easy for us to manage and collaborate).
2. Distributed (Can maintain the copy of current version of the code locally and can create a clone copy of the repository)
3. Security (Each and every commit will be stored int the repository using secure hashing (SHA))
4. Speed

**Advantages of GIT:**

1. Offline working
2. Undo mistakes (we can undo the operations).
3. Track the changes. (Diff, Log, status)

**Environments in GIT:**

1. Local area or working area
2. Staging area
3. Repository (local repository)

**Why GIT?**

Speed

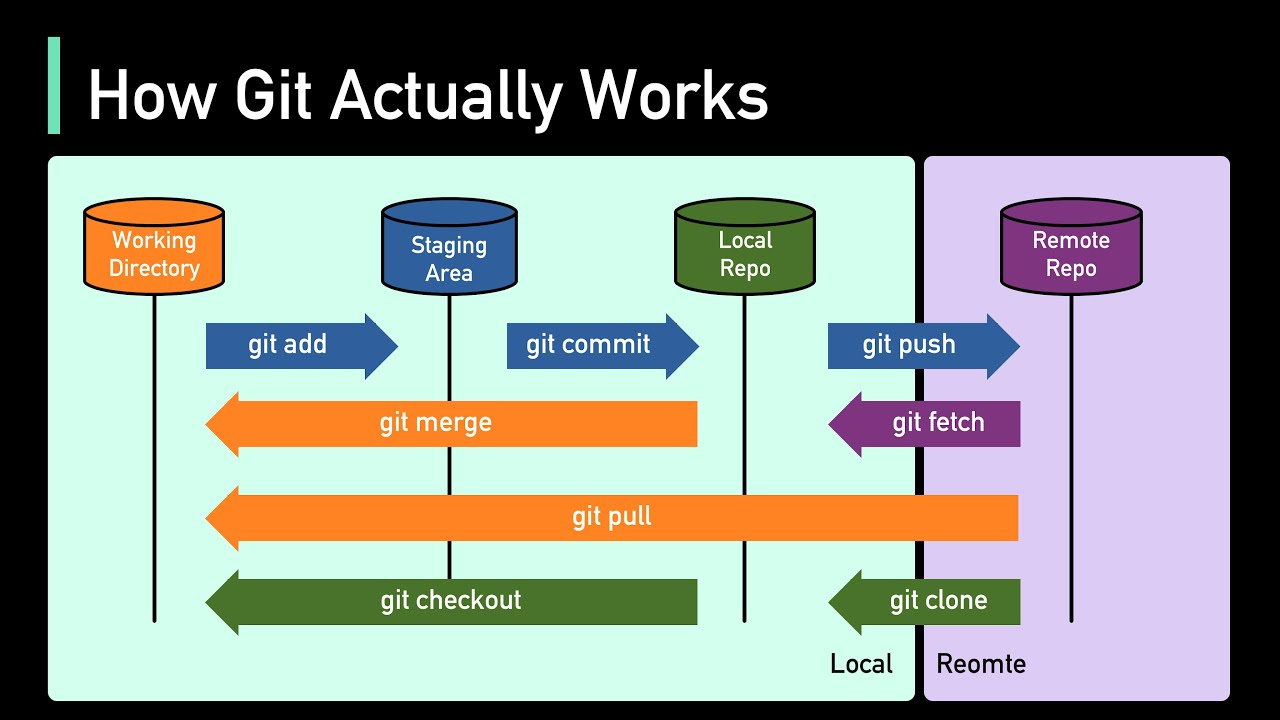
Security

Integrity

Local repository

Clean history

**How GIT works?**



**Practice session:**

meenakshi.ananthan@RLP2183 MINGW64 ~

$ cd D:

meenakshi.ananthan@RLP2183 MINGW64 /d

$ ls

'$RECYCLE.BIN'/ 'System Volume Information'/ apache-tomcat-9.0.86-windows-x64/ docker/ javascript/ sts-4.21.1.RELEASE/

Meenakshi\_VScode/ Tools/ apache-tomcat-9.0.86-windows-x64.zip eclipse/ spring-tool-suite-4-4.21.1.RELEASE-e4.30.0-win32.win32.x86\_64.self-extracting.jar

meenakshi.ananthan@RLP2183 MINGW64 /d

$ mkdir GitWS

meenakshi.ananthan@RLP2183 MINGW64 /d

$ cd GitWS

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS

$ git init

Initialized empty Git repository in D:/GitWS/.git/

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls -a

./ ../ .git/

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ cd .git

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS/.git (GIT\_DIR!)

$ ls

HEAD config description hooks/ info/ objects/ refs/

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS/.git (GIT\_DIR!)

$ cd ..

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS

$ ls -a

./ ../

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS

$ git init

Initialized empty Git repository in D:/GitWS/.git/

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ echo "Hello World" > file1.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ cat > file2.txt

Hi i am Meenakshi

[1]+ Stopped cat > file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

file1.txt

file2.txt

nothing added to commit but untracked files present (use "git add" to track)

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git add .

warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: file1.txt

new file: file2.txt

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git rm --cached file2.txt

error: the following file has staged content different from both the

file and the HEAD:

file2.txt

(use -f to force removal)

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ file2.txt -f

bash: file2.txt: command not found

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: file1.txt

new file: file2.txt

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls -files

git: 'ls' is not a git command. See 'git --help'.

The most similar command is

lfs

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-files

file1.txt

file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git commit -m "Initial COmmit"

[master (root-commit) aa274ba] Initial COmmit

Committer: Meenakshi Priyadharshini Ananthan <meenakshi.ananthan@itp.objectfrontier.com>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

2 files changed, 1 insertion(+)

create mode 100644 file1.txt

create mode 100644 file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git config

usage: git config [<options>]

Config file location

--[no-]global use global config file

--[no-]system use system config file

--[no-]local use repository config file

--[no-]worktree use per-worktree config file

-f, --[no-]file <file>

use given config file

--[no-]blob <blob-id> read config from given blob object

Action

--[no-]get get value: name [value-pattern]

--[no-]get-all get all values: key [value-pattern]

--[no-]get-regexp get values for regexp: name-regex [value-pattern]

--[no-]get-urlmatch get value specific for the URL: section[.var] URL

--[no-]replace-all replace all matching variables: name value [value-pattern]

--[no-]add add a new variable: name value

--[no-]unset remove a variable: name [value-pattern]

--[no-]unset-all remove all matches: name [value-pattern]

--[no-]rename-section rename section: old-name new-name

--[no-]remove-section remove a section: name

-l, --[no-]list list all

--[no-]fixed-value use string equality when comparing values to 'value-pattern'

-e, --[no-]edit open an editor

--[no-]get-color find the color configured: slot [default]

--[no-]get-colorbool find the color setting: slot [stdout-is-tty]

Type

-t, --[no-]type <type>

value is given this type

--bool value is "true" or "false"

--int value is decimal number

--bool-or-int value is --bool or --int

--bool-or-str value is --bool or string

--path value is a path (file or directory name)

--expiry-date value is an expiry date

Other

-z, --[no-]null terminate values with NUL byte

--[no-]name-only show variable names only

--[no-]includes respect include directives on lookup

--[no-]show-origin show origin of config (file, standard input, blob, command line)

--[no-]show-scope show scope of config (worktree, local, global, system, command)

--[no-]default <value>

with --get, use default value when missing entry

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git config --global

usage: git config [<options>]

Config file location

--[no-]global use global config file

--[no-]system use system config file

--[no-]local use repository config file

--[no-]worktree use per-worktree config file

-f, --[no-]file <file>

use given config file

--[no-]blob <blob-id> read config from given blob object

Action

--[no-]get get value: name [value-pattern]

--[no-]get-all get all values: key [value-pattern]

--[no-]get-regexp get values for regexp: name-regex [value-pattern]

--[no-]get-urlmatch get value specific for the URL: section[.var] URL

--[no-]replace-all replace all matching variables: name value [value-pattern]

--[no-]add add a new variable: name value

--[no-]unset remove a variable: name [value-pattern]

--[no-]unset-all remove all matches: name [value-pattern]

--[no-]rename-section rename section: old-name new-name

--[no-]remove-section remove a section: name

-l, --[no-]list list all

--[no-]fixed-value use string equality when comparing values to 'value-pattern'

-e, --[no-]edit open an editor

--[no-]get-color find the color configured: slot [default]

--[no-]get-colorbool find the color setting: slot [stdout-is-tty]

Type

-t, --[no-]type <type>

value is given this type

--bool value is "true" or "false"

--int value is decimal number

--bool-or-int value is --bool or --int

--bool-or-str value is --bool or string

--path value is a path (file or directory name)

--expiry-date value is an expiry date

Other

-z, --[no-]null terminate values with NUL byte

--[no-]name-only show variable names only

--[no-]includes respect include directives on lookup

--[no-]show-origin show origin of config (file, standard input, blob, command line)

--[no-]show-scope show scope of config (worktree, local, global, system, command)

--[no-]default <value>

with --get, use default value when missing entry

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git config --list

diff.astextplain.textconv=astextplain

filter.lfs.clean=git-lfs clean -- %f

filter.lfs.smudge=git-lfs smudge -- %f

filter.lfs.process=git-lfs filter-process

filter.lfs.required=true

http.sslbackend=openssl

http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager

credential.https://dev.azure.com.usehttppath=true

init.defaultbranch=master

core.repositoryformatversion=0

core.filemode=false

core.bare=false

core.logallrefupdates=true

core.symlinks=false

core.ignorecase=true

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ q

bash: q: command not found

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git config -- list

error: key does not contain a section: list

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git config --list

diff.astextplain.textconv=astextplain

filter.lfs.clean=git-lfs clean -- %f

filter.lfs.smudge=git-lfs smudge -- %f

filter.lfs.process=git-lfs filter-process

filter.lfs.required=true

http.sslbackend=openssl

http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager

credential.https://dev.azure.com.usehttppath=true

init.defaultbranch=master

core.repositoryformatversion=0

core.filemode=false

core.bare=false

core.logallrefupdates=true

core.symlinks=false

core.ignorecase=true

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git config --global user.name "Meenakshi"

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git config --global user.email "meenakshiananthan8@gmail.com"

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git config --list

diff.astextplain.textconv=astextplain

filter.lfs.clean=git-lfs clean -- %f

filter.lfs.smudge=git-lfs smudge -- %f

filter.lfs.process=git-lfs filter-process

filter.lfs.required=true

http.sslbackend=openssl

http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager

credential.https://dev.azure.com.usehttppath=true

init.defaultbranch=master

user.name=Meenakshi

user.email=meenakshiananthan8@gmail.com

core.repositoryformatversion=0

core.filemode=false

core.bare=false

core.logallrefupdates=true

core.symlinks=false

core.ignorecase=true

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: file2.txt

no changes added to commit (use "git add" and/or "git commit -a")

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git log

commit aa274ba107cfd61d8f2c94b9d453fdc31950fd87 (HEAD -> master)

Author: Meenakshi Priyadharshini Ananthan <meenakshi.ananthan@itp.objectfrontier.com>

Date: Fri Mar 8 11:41:13 2024 +0530

Initial COmmit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ cat > file3.txt

Hi Hello

Good Morning

[2]+ Stopped cat > file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git add file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git commit -m "file3.txt added"

[master 7ec74e2] file3.txt added

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: file2.txt

modified: file3.txt

no changes added to commit (use "git add" and/or "git commit -a")

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git log

commit 7ec74e28f1cbd80cd3b7b8ad9cdb6d7a1677a20d (HEAD -> master)

Author: Meenakshi <meenakshiananthan8@gmail.com>

Date: Fri Mar 8 11:58:53 2024 +0530

file3.txt added

commit aa274ba107cfd61d8f2c94b9d453fdc31950fd87

Author: Meenakshi Priyadharshini Ananthan <meenakshi.ananthan@itp.objectfrontier.com>

Date: Fri Mar 8 11:41:13 2024 +0530

Initial COmmit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: file2.txt

modified: file3.txt

no changes added to commit (use "git add" and/or "git commit -a")

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git add file2.txt

warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time Git touches it

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git add file3.txt

warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: file2.txt

modified: file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ gti commit -m " staged the modified files of file2 and file3"

bash: gti: command not found

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git log

commit 7ec74e28f1cbd80cd3b7b8ad9cdb6d7a1677a20d (HEAD -> master)

Author: Meenakshi <meenakshiananthan8@gmail.com>

Date: Fri Mar 8 11:58:53 2024 +0530

file3.txt added

commit aa274ba107cfd61d8f2c94b9d453fdc31950fd87

Author: Meenakshi Priyadharshini Ananthan <meenakshi.ananthan@itp.objectfrontier.com>

Date: Fri Mar 8 11:41:13 2024 +0530

Initial Commit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git commit -m " staged the modified files of file2 and file3"

[master 8a7417f] staged the modified files of file2 and file3

2 files changed, 4 insertions(+)

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git log

commit 8a7417f6ee745aaf69632476c8e0b30689b6f341 (HEAD -> master)

Author: Meenakshi <meenakshiananthan8@gmail.com>

Date: Fri Mar 8 12:04:42 2024 +0530

staged the modified files of file2 and file3

commit 7ec74e28f1cbd80cd3b7b8ad9cdb6d7a1677a20d

Author: Meenakshi <meenakshiananthan8@gmail.com>

Date: Fri Mar 8 11:58:53 2024 +0530

file3.txt added

commit aa274ba107cfd61d8f2c94b9d453fdc31950fd87

Author: Meenakshi Priyadharshini Ananthan <meenakshi.ananthan@itp.objectfrontier.com>

Date: Fri Mar 8 11:41:13 2024 +0530

Initial Commit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-list

git: 'ls-list' is not a git command. See 'git --help'.

The most similar commands are

ls-files

rev-list

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-files

file1.txt

file2.txt

file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

nothing to commit, working tree clean

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git log --oneline

8a7417f (HEAD -> master) staged the modified files of file2 and file3

7ec74e2 file3.txt added

aa274ba Initial COmmit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ rm file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-files

file1.txt

file2.txt

file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls

git: 'ls' is not a git command. See 'git --help'.

The most similar command is

lfs

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls

file1.txt file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git restore file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls

file1.txt file2.txt file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git rm file2.txt

rm 'file2.txt'

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-files

file1.txt

file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls

file1.txt file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ restore file2.txt

Directory /usr/etc/backup is not found.

The following directories and files are needed for the backup to function:

1. Directory with configuration files and file lists:

/usr/etc/backup

2. Directory for backup log files

/usr/etc/backup/log

3. Main configuration file

/usr/etc/backup/backup-specs

Please, create these and invoke the script again.

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git log

commit 8a7417f6ee745aaf69632476c8e0b30689b6f341 (HEAD -> master)

Author: Meenakshi <meenakshiananthan8@gmail.com>

Date: Fri Mar 8 12:04:42 2024 +0530

staged the modified files of file2 and file3

commit 7ec74e28f1cbd80cd3b7b8ad9cdb6d7a1677a20d

Author: Meenakshi <meenakshiananthan8@gmail.com>

Date: Fri Mar 8 11:58:53 2024 +0530

file3.txt added

commit aa274ba107cfd61d8f2c94b9d453fdc31950fd87

Author: Meenakshi Priyadharshini Ananthan <meenakshi.ananthan@itp.objectfrontier.com>

Date: Fri Mar 8 11:41:13 2024 +0530

Initial COmmit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git reset --hard Head

HEAD is now at 8a7417f staged the modified files of file2 and file3

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls-files

bash: ls-files: command not found

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-files

file1.txt

file2.txt

file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls

file1.txt file2.txt file3.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git reset --hard Head~2

HEAD is now at aa274ba Initial COmmit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git reset --hard Head~1

fatal: ambiguous argument 'Head~1': unknown revision or path not in the working tree.

Use '--' to separate paths from revisions, like this:

'git <command> [<revision>...] -- [<file>...]'

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git log

commit aa274ba107cfd61d8f2c94b9d453fdc31950fd87 (HEAD -> master)

Author: Meenakshi Priyadharshini Ananthan <meenakshi.ananthan@itp.objectfrontier.com>

Date: Fri Mar 8 11:41:13 2024 +0530

Initial COmmit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ cat file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

deleted: file2.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

sample.txt

no changes added to commit (use "git add" and/or "git commit -a")

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ gti add sample.txt

bash: gti: command not found

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git add sample.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: sample.txt

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

deleted: file2.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git commit -m "file added"

[master 3f62c72] file added

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 sample.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

deleted: file2.txt

no changes added to commit (use "git add" and/or "git commit -a")

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git add sample1

fatal: pathspec 'sample1' did not match any files

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git add sample1.txt

warning: in the working copy of 'sample1.txt', LF will be replaced by CRLF the next time Git touches it

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git commit -m"some msg"

[master 8fdf212] some msg

1 file changed, 1 insertion(+)

create mode 100644 sample1.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

deleted: file1.txt

deleted: file2.txt

no changes added to commit (use "git add" and/or "git commit -a")

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-files

file1.txt

file2.txt

sample.txt

sample1.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git --version

git version 2.43.0.windows.1

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ rm sample.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-files

file1.txt

file2.txt

sample.txt

sample1.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ ls

sample1.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git add .

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git ls-files

sample1.txt

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$ git log

commit 8fdf2120b81bbea5acb4469e98c4500bd1acc453 (HEAD -> master)

Author: Meenakshi <meenakshiananthan8@gmail.com>

Date: Fri Mar 8 12:27:30 2024 +0530

some msg

commit 3f62c723dff5df5794ac5f0e79540fcfa34ad766

Author: Meenakshi <meenakshiananthan8@gmail.com>

Date: Fri Mar 8 12:26:14 2024 +0530

file added

commit aa274ba107cfd61d8f2c94b9d453fdc31950fd87

Author: Meenakshi Priyadharshini Ananthan <meenakshi.ananthan@itp.objectfrontier.com>

Date: Fri Mar 8 11:41:13 2024 +0530

Initial COmmit

meenakshi.ananthan@RLP2183 MINGW64 /d/GitWS (master)

$

Branch: - new/separate version of the main repository

git branch branch-name - to create the new branch

git branch -v - to see the branches

git checkout branch-name -to move the any branch

git ls-file to see the files in the branch

\* to some modification in the branch file

git status

git add newfile

git commit -m “some command”

git branch -d branch-name - to delete the branch

git merge branch-name -to merge the branch into master

**STASH:**

to store incomplete data in a hidden place

Stack data structure is used to store

git stash – the incomplete work is saved in the stack

git stash pop – to open up the recently closed task

G**it stash show** command is used to display the summary of operations performed on the stash.