**C:\sers\Admin>git**

usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

[--super-prefix=<path>] [--config-env=<name>=<envvar>]

<command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

**C:\Users\Admin>git -v**

git version 2.39.0.windows.2

**C:\Users\Admin>D:**

**D:\>cd D:\KAla\Kala documents**

**D:\KAla\Kala documents>git clone https://github.com/Kala-shekhar/Git-demo1.git**

Cloning into 'Git-demo1'...

remote: Enumerating objects: 3, done.

remote: Counting objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Receiving objects: 100% (3/3), done.

**D:\KAla\Kala documents>cd D:\KAla\Kala documents\Git-demo1**

**D:\KAla\Kala documents\Git-demo1>git pull**

remote: Enumerating objects: 4, done.

remote: Counting objects: 100% (4/4), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 664 bytes | 10.00 KiB/s, done.

From https://github.com/Kala-shekhar/Git-demo1

a343ea3..7c973e8 main -> origin/main

Updating a343ea3..7c973e8

Fast-forward

kala test | 1 +

1 file changed, 1 insertion(+)

create mode 100644 kala test

**D:\KAla\Kala documents\Git-demo1>git status**

On branch main

Your branch is up to date with 'origin/main'.

Untracked files:

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

Kala test2.txt

test3.txt

test4.txt

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

**D:\KAla\Kala documents\Git-demo1>git add test3.txt**

**D:\KAla\Kala documents\Git-demo1>git status**

On branch main

Your branch is up to date with 'origin/main'.

Changes to be committed:

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

new file: test3.txt

Untracked files:

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

Kala test2.txt

test4.txt

**D:\KAla\Kala documents\Git-demo1>git commit -m "kala"**

[main d814d79] kala

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

create mode 100644 test3.txt

**D:\KAla\Kala documents\Git-demo1>git status**

On branch main

Your branch is ahead of 'origin/main' by 1 commit.

(use "git push" to publish your local commits)

Untracked files:

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

Kala test2.txt

test4.txt

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

**D:\KAla\Kala documents\Git-demo1>git add --all**

**D:\KAla\Kala documents\Git-demo1>git staus**

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

The most similar command is

status

**D:\KAla\Kala documents\Git-demo1>git status**

On branch main

Your branch is ahead of 'origin/main' by 1 commit.

(use "git push" to publish your local commits)

Changes to be committed:

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

new file: Kala test2.txt

new file: test4.txt

**D:\KAla\Kala documents\Git-demo1>git commit -m "function"**

[main afec59c] function

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 Kala test2.txt

create mode 100644 test4.txt

**D:\KAla\Kala documents\Git-demo1>git status**

On branch main

Your branch is ahead of 'origin/main' by 2 commits.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

**D:\KAla\Kala documents\Git-demo1>git push https://github.com/Kala-shekhar/Git-demo1.git**

info: please complete authentication in your browser...

Enumerating objects: 6, done.

Counting objects: 100% (6/6), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (5/5), 481 bytes | 68.00 KiB/s, done.

Total 5 (delta 1), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (1/1), done.

To https://github.com/Kala-shekhar/Git-demo1.git

7c973e8..afec59c main -> main

**D:\KAla\Kala documents\Git-demo1>git status**

On branch main

Your branch is ahead of 'origin/main' by 2 commits.

(use "git push" to publish your local commits)

nothing to commit, working tree clean