NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT

$ **git config --global user.name "Nayak,Rajesh"** 🡪 CMD used to configure user name in GIT config file

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT

$ 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/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager-core

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

user.name=Nayak,Rajesh

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT

$ **git config --global user.email "rajeshnayak075@gmail.com"** 🡪 CMD used to configure Email in GIT config file

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT

$ **git config –-list 🡪** CMD to check configuration set in gitconfig file

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/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager-core

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

user.name=Nayak,Rajesh

user.email=rajeshnayak075@gmail.com

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT

$ **git help**

usage: git [--version] [--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>]

<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

sparse-checkout Initialize and modify the sparse-checkout

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.

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT

**$ git --help add 🡪 This** command is used to get details of git command

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT

$ **git init 🡪** this command is used to initialize git in the directory

Initialized empty Git repository in C:/Users/NAYAK/OneDrive/Desktop/GIT/.git/

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

$ git status

On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

$ touch FirstCheckIn.txt

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

$ ls -lta

total 12

-rw-r--r-- 1 NAYAK 197121 0 Feb 10 16:34 FirstCheckIn.txt

drwxr-xr-x 1 NAYAK 197121 0 Feb 10 16:34 ./

drwxr-xr-x 1 NAYAK 197121 0 Feb 10 16:33 .git/

drwxr-xr-x 1 NAYAK 197121 0 Feb 9 22:43 ../

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

$ git status

On branch master

No commits yet

Untracked files:

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

FirstCheckIn.txt

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

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

$ **git add .**

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: FirstCheckIn.txt

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

**$ git commit -m "First checkin"**

[master (root-commit) e4969f4] First checkin

1 file changed, 22 insertions(+)

create mode 100644 FirstCheckIn.txt

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

$ **git log**

commit e4969f4d389de1ea2e95885ace9097e1bef9eeb0 (HEAD -> master)

Author: Nayak,Rajesh <rajeshnayak075@gmail.com>

Date: Wed Feb 10 16:41:05 2021 +0530

First checkin

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

**$ git log -author="Nayak,Rajesh"**

fatal: unrecognized argument: -author=Nayak,Rajesh

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/GIT (master)

**$ git log --author="Nayak,Rajesh"**

commit 92079b28c3da6e86e5809bedc9f8d4bef3de72ef (HEAD -> master)

Author: Nayak,Rajesh <rajeshnayak075@gmail.com>

Date: Wed Feb 10 19:50:45 2021 +0530

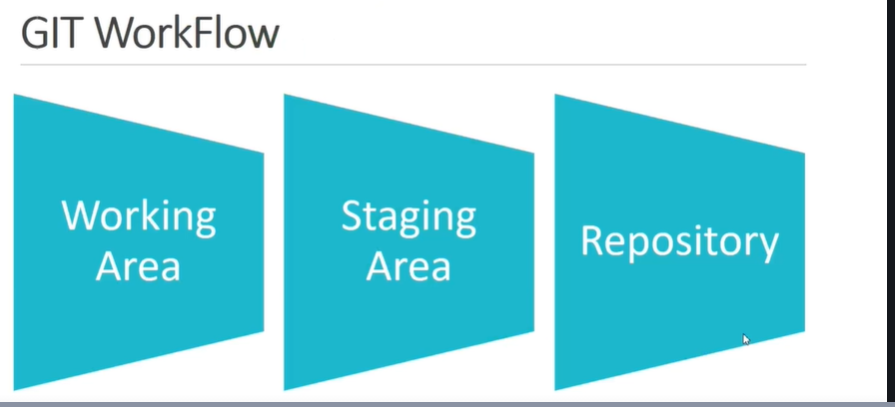
Second Checkin

commit e4969f4d389de1ea2e95885ace9097e1bef9eeb0

Author: Nayak,Rajesh <rajeshnayak075@gmail.com>

Date: Wed Feb 10 16:41:05 2021 +0530

First checkin



Generating a new SSH key and adding it to the ssh-agent

<https://docs.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

1. Paste the text below, substituting in your GitHub email address. This creates a new ssh key, using the provided email as a label.

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/git class/GIT-Tutorial\_checkin

$ **ssh-keygen -t ed25519 -C "rajeshnayak075@gmail.com"**

Generating public/private ed25519 key pair.

Enter file in which to save the key (/c/Users/NAYAK/.ssh/id\_ed25519):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/NAYAK/.ssh/id\_ed25519

Your public key has been saved in /c/Users/NAYAK/.ssh/id\_ed25519.pub

The key fingerprint is:

SHA256:5TG4b2zGYIJJzzng95OVjlS+o8jAh92Vr2cj5Et1/Q8 rajeshnayak075@gmail.com

The key's randomart image is:

+--[ED25519 256]--+

| |

| . |

| o . = |

| o \* . \* = . |

| + B S B . . .|

| . + B @.+ . .|

| + o \*o% . E .|

| + . Boo+ ..|

| o . o= . .|

+----[SHA256]-----+

1. Ensure the ssh-agent is running. You can use the "Auto-launching the ssh-agent" instructions in "Working with SSH key passphrases", or start it manually:

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/git class/GIT-Tutorial\_checkin

$ **eval `ssh-agent -s`**

Agent pid 2011

1. Add your SSH private key to the ssh-agent. If you created your key with a different name, or if you are adding an existing key that has a different name, replace *id\_ed25519* in the command with the name of your private key file.

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/git class/GIT-Tutorial\_checkin

$ **ssh-add ~/.ssh/id\_ed25519**

Identity added: /c/Users/NAYAK/.ssh/id\_ed25519 (rajeshnayak075@gmail.com)

1. Copy the SSH public key to your clipboard.
2. If your SSH public key file has a different name than the example code, modify the filename to match your current setup. When copying your key, don't add any newlines or whitespace.

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/git class/GIT-Tutorial\_checkin

$ **clip < ~/.ssh/id\_ed25519.pub**

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/git class/GIT-Tutorial\_checkin

$ git clone git@github.com:rajeshKnayak/DevopsTraining.git

Cloning into 'DevopsTraining'...

remote: Enumerating objects: 3, done.

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

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

NAYAK@LAPTOP-58ANFJT0 MINGW64 ~/OneDrive/Desktop/git class/GIT-Tutorial\_checkin