

# DotNet-FSE Mandatory Hands-On

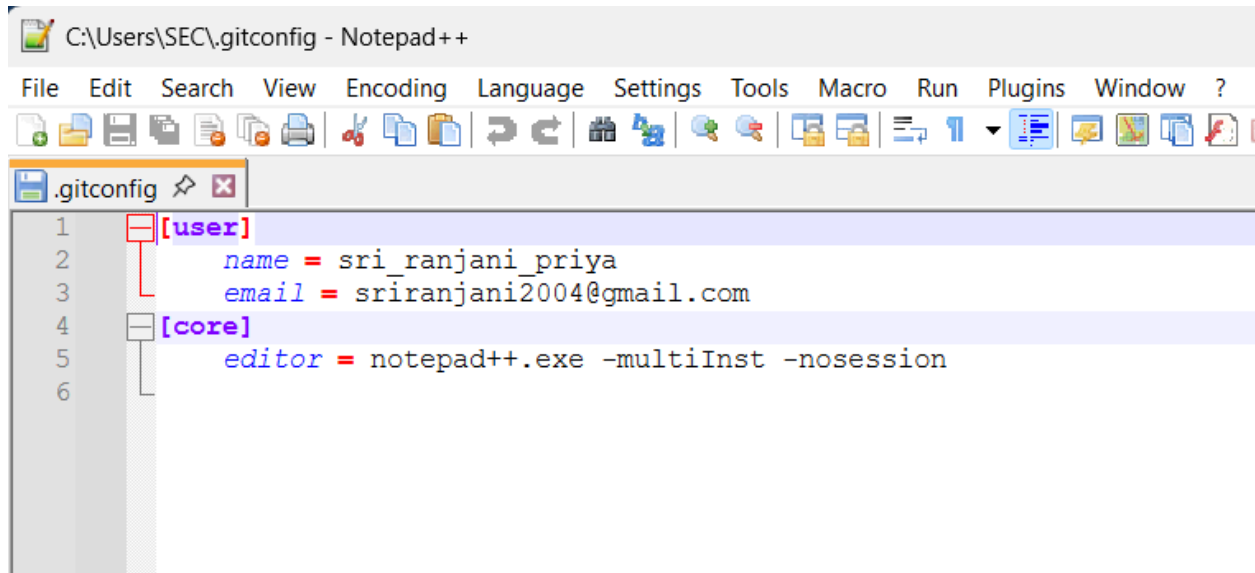
WEEK-8

NAME: Sri Ranjani Priya P

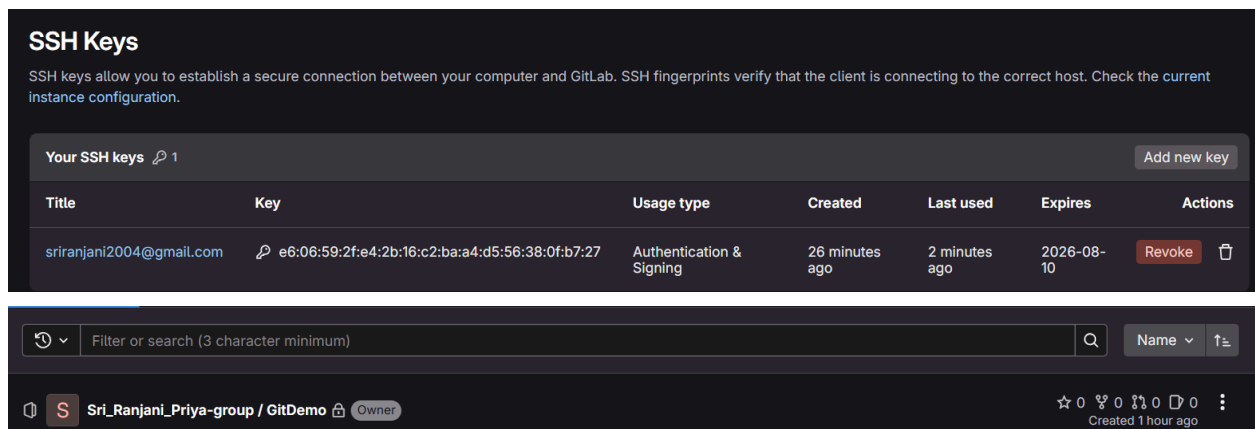
## EXERCISE 1:

In this hands-on lab, you will learn how to- Setup your machine with Git Configuration, Integrate notepad++.exe to Git and make it a default editor, Add a file to source code repository

## COMMANDS AND OUTPUTS:



```
C:\Users\SEC\.gitconfig - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
[.gitconfig]
1 [user]
2     name = sri_ranjani_priya
3     email = sriranjani2004@gmail.com
4 [core]
5     editor = notepad++.exe -multiInst -nosession
6
```



### SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the [current instance configuration](#).

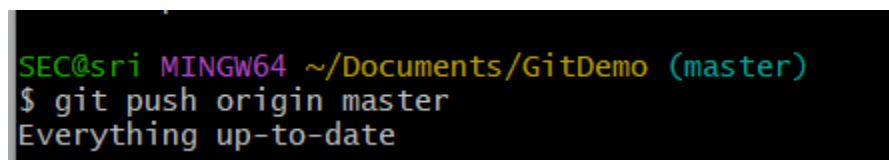
Your SSH keys 1 [Add new key](#)

Title	Key	Usage type	Created	Last used	Expires	Actions
sriranjani2004@gmail.com	e6:06:59:2f:e4:2b:16:c2:ba:a4:d5:56:38:0f:b7:27	Authentication & Signing	26 minutes ago	2 minutes ago	2026-08-10	<a href="#">Revoke</a> <a href="#">Delete</a>

Filter or search (3 character minimum) [Name](#) [Sort](#)

Sri\_Ranjani\_Priya-group / GitDemo [Owner](#) [Star](#) 0 [Fork](#) 0 [Watch](#) 0 [More](#)

Created 1 hour ago



```
SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git push origin master
Everything up-to-date
```

✓ You pushed to master at [Sri\\_Ranjani\\_Priya-group / GitDemo](#) 2 minutes ago

Create merge request

## EXERCISE 2:

Create a **“.log”** file and a **log folder** in the working directory of Git. Update the **.gitignore** file in such a way that on committing, these files (.log extensions and log folders) are ignored. Verify if the git status reflects the same about working directory, local repository and git repository.

## COMMANDS AND OUTPUTS:

```
SEC@sri MINGW64 ~ (master)
$ git init
Reinitialized existing Git repository in C:/Users/SEC/.git/

SEC@sri MINGW64 ~ (master)
$ echo "This is a log file" > sample.log

SEC@sri MINGW64 ~ (master)
$ mkdir log

SEC@sri MINGW64 ~ (master)
$ echo "Log file content" > log/logfile1.txt

SEC@sri MINGW64 ~ (master)
$ notepad++ .gitignore
```

```
SEC@sri MINGW64 ~/Documents/GitDemo (main)
$ git status
On branch main
nothing to commit, working tree clean
```

## EXERCISE 3:

Construct a branch, do some changes in the branch, and merge it with master (or trunk)

## COMMANDS AND OUTPUTS:

```

cd /C/users/SEC/Documents/GitDemo
EC@sri MINGW64 ~/Documents/GitDemo (master)
git branch GitNewBranchUp

EC@sri MINGW64 ~/Documents/GitDemo (master)
git branch -a
0a
GitNewBranch
GitNewBranch1
GitNewBranchUp
main
master
remotes/origin/main
remotes/origin/master

EC@sri MINGW64 ~/Documents/GitDemo (master)
git checkout GitNewBranchUp
Switched to branch 'GitNewBranchUp'

EC@sri MINGW64 ~/Documents/GitDemo (GitNewBranchUp)
echo "This is GitNewBranchUp File" > Gitup.txt

EC@sri MINGW64 ~/Documents/GitDemo (GitNewBranchUp)
git add Gitup.txt
warning: in the working copy of 'Gitup.txt', LF will be replaced by CRLF the next time you commit.

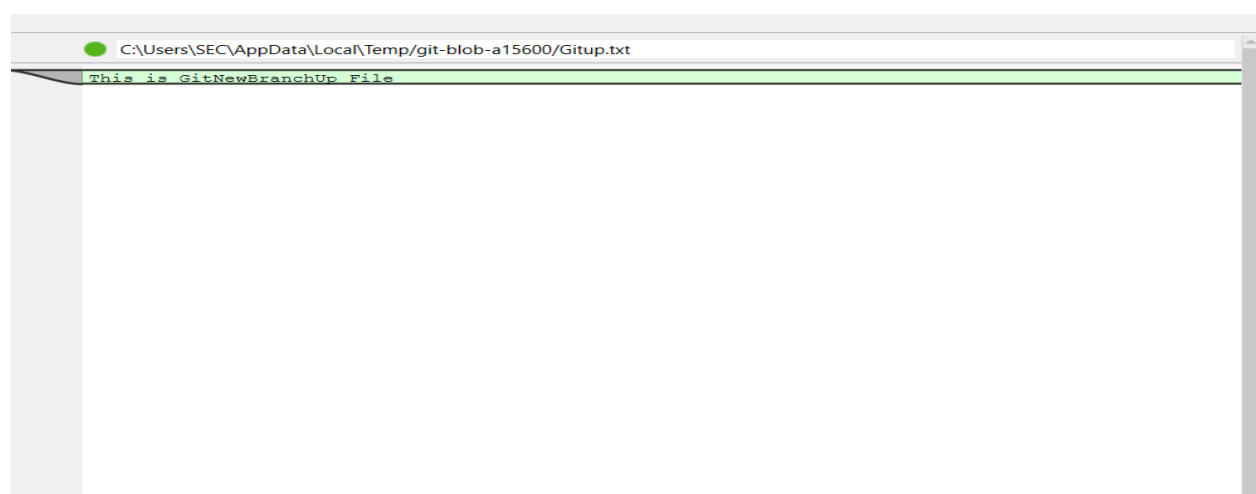
EC@sri MINGW64 ~/Documents/GitDemo (GitNewBranchUp)
git commit -m "Add Gitup.txt file to GitNewBranchUp"
[GitNewBranchUp 5668cc0] Add Gitup.txt file to GitNewBranchUp
1 file changed, 1 insertion(+)
create mode 100644 Gitup.txt

EC@sri MINGW64 ~/Documents/GitDemo (GitNewBranchUp)
git status
On branch GitNewBranchUp
nothing to commit, working tree clean

EC@sri MINGW64 ~/Documents/GitDemo (GitNewBranchUp)
git checkout master
Switched to branch 'master'

EC@sri MINGW64 ~/Documents/GitDemo (master)
git diff master GitNewBranchUp
diff --git a/Gitup.txt b/Gitup.txt
new file mode 100644
index 0000000..74b2539
--- /dev/null
+++ b/Gitup.txt
@@ -0,0 +1 @@
This is GitNewBranchUp File

```



```

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git difftool master GitNewBranchUp

This message is displayed because 'diff.tool' is not configured.
See 'git difftool --tool-help' or 'git help config' for more details.
'git difftool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuse diffmerge ecmerge
p4merge smerge emerge vimdiff nvimdiff

Viewing (1/1): 'Gitup.txt'
Launch 'p4merge' [Y/n]? y

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git merge GitNewBranchUp
Updating 65296a5..5668cc0
Fast-forward
 Gitup.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Gitup.txt

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git log --oneline --graph --decorate
* 5668cc0 (HEAD -> master, GitNewBranchUp) Add GitUp.txt file to GitNewBranchUp
* 65296a5 Add diff.txt file to GitNewBranch2
* ad77c21 (GitNewBranch, 0a) Addes example.txt in GitNewBranch
* ab1c01c (main) Update .gitignore to ignore .log files and log folder
* 1308277 Add .gitignore to ignore .log files and log folder
* 4625405 (origin/master, origin/main) Initial commit

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git branch -d GitNewBranchUp
Deleted branch GitNewBranchUp (was 5668cc0).

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git status
On branch master
nothing to commit, working tree clean

```

#### EXERCISE 4:

Implement conflict resolution when multiple users are updating the trunk (or master) in such a way that it results into a conflict with the branch's modification.

#### COMMANDS AND OUTPUTS:

```

EC@sri MINGW64 ~/Documents/GitDemo (GitWork)
> git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF t

EC@sri MINGW64 ~/Documents/GitDemo (GitWork)
> git commit -m "Add hello.xml and update content in GitWork"
[GitWork 3c155b5] Add hello.xml and update content in GitWork
1 file changed, 2 insertions(+)
create mode 100644 hello.xml

EC@sri MINGW64 ~/Documents/GitDemo (GitWork)
> git status
On branch GitWork
nothing to commit, working tree clean

EC@sri MINGW64 ~/Documents/GitDemo (GitWork)
> git checkout master
error: pathspec 'master' did not match any file(s) known to git
ash: got: command not found

EC@sri MINGW64 ~/Documents/GitDemo (GitWork)
> git checkout master
error: pathspec 'master' did not match any file(s) known to git
switched to branch 'master'

EC@sri MINGW64 ~/Documents/GitDemo (master)
> echo "<message>Hello from master branch</message>" hello.xml
<message>Hello from master branch</message> hello.xml

EC@sri MINGW64 ~/Documents/GitDemo (master)
> echo "<info>Different content from master branch</info>" >> hello.xml

EC@sri MINGW64 ~/Documents/GitDemo (master)
> echo "<message>Hello from master branch</message>" > hello.xml

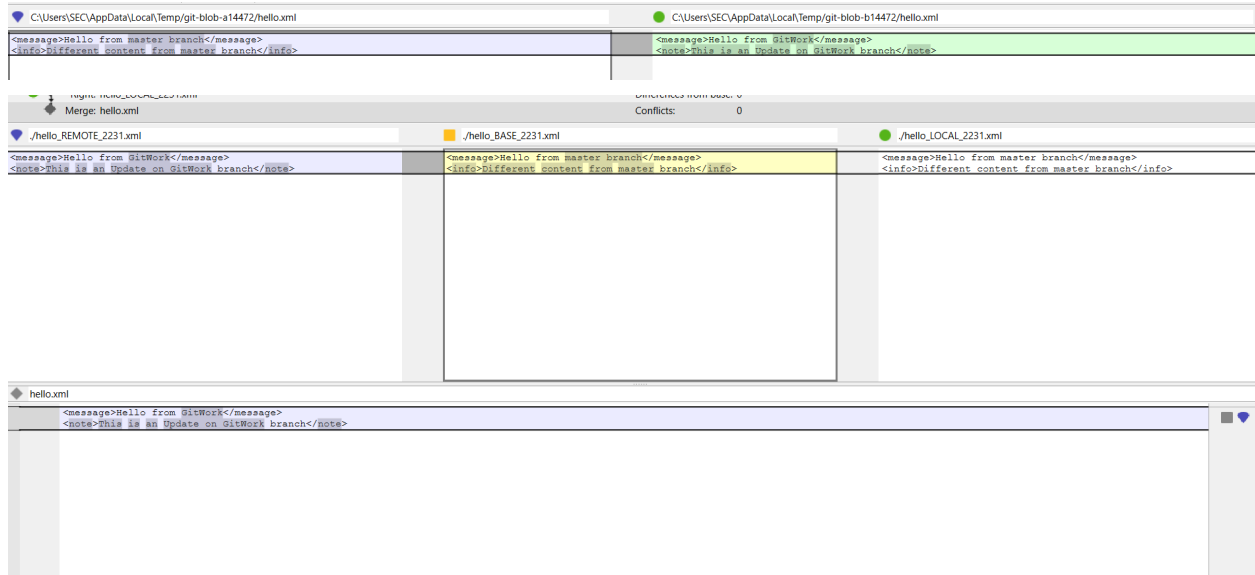
EC@sri MINGW64 ~/Documents/GitDemo (master)
> echo "<info>Different content from master branch</info>" >> hello.xml

EC@sri MINGW64 ~/Documents/GitDemo (master)
> git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF t

EC@sri MINGW64 ~/Documents/GitDemo (master)
> git commit -m "Add hello.xml with different content on master branch"
[master fb76030] Add hello.xml with different content on master branch
1 file changed, 2 insertions(+), 2 deletions(-)

EC@sri MINGW64 ~/Documents/GitDemo (master)
> git diff master GitWork -- hello.xml
diff --git a/hello.xml b/hello.xml
index 912a26d..9b00892 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1,2 +1,2 @@
<message>Hello from master branch</message>
<info>Different content from master branch</info>
<message>Hello from GitWork</message>

```



```

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello_BACKUP_2157.xml
    hello_BASE_2157.xml
    hello_LOCAL_2157.xml
    hello_REMOTE_2157.xml

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

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ echo "*.orig" >> .gitignore

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git commit -m "Add backup files to .gitignore"
[master ee30319] Add backup files to .gitignore
1 file changed, 1 insertion(+)

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git branch
* 0a
  GitNewBranch
  GitNewBranch1
  GitWork
  main
  master

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git branch -d GitWork
Deleted branch GitWork (was 3c155b5).

SEC@sri MINGW64 ~/Documents/GitDemo (master)
$ git log --oneline --graph --decorate
* ee30319 (HEAD -> master) Add backup files to .gitignore
* 630455e Resolve merge conflict in hello.xml
* 3c155b5 Add hello.xml and update content in GitWork
  | fb76030 Add hello.xml with different content on master branch
  | 59eaa5e Add hello.xml and update content in GitWork
  |
* 5668cc0 Add GitUp.txt file to GitNewBranchUp
* 65296a5 Add diff.txt file to GitNewBranch2
* ad77c21 (GitNewBranch, 0a) Addes example.txt in GitNewBranch
* ab1c01c (main) Update .gitignore to ignore .log files and log folder
* 1308277 Add .gitignore to ignore .log files and log folder
* 4625405 (origin/master, origin/main) Initial commit

```

## EXERCISE 5:

Execute steps involving clean up and push back to remote Git.

### COMMANDS AND OUTPUTS:

```
SEC@sri MINGW64 ~/Documents/MyGit (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

nothing to commit, working tree clean

```
SEC@sri MINGW64 ~/Documents/MyGit (master)
$ git branch -a
* master
  some-other-branch
remotes/origin/HEAD -> origin/master
remotes/origin/Git-T03-HOL_002
remotes/origin/master
```

```
SEC@sri MINGW64 ~/Documents/MyGit (master)
$ git checkout master
Already on 'master'
```

```
SEC@sri MINGW64 ~/Documents/MyGit (master)
$ git pull origin master
From https://github.com/YourUsername/YourRepo
* branch      master    -> FETCH_HEAD
Already up to date.
```

```
SEC@sri MINGW64 ~/Documents/MyGit (master)
$ git checkout Git-T03-HOL_002
Switched to branch 'Git-T03-HOL_002'
```

```
SEC@sri MINGW64 ~/Documents/MyGit (Git-T03-HOL_002)
$ git push origin Git-T03-HOL_002
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 304 bytes | 304.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
```

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To <https://github.com/YourUsername/YourRepo.git>

\* [new branch] Git-T03-HOL\_002 -> Git-T03-HOL\_002

SEC@sri MINGW64 ~/Documents/MyGit (Git-T03-HOL\_002)

\$ git fetch origin

SEC@sri MINGW64 ~/Documents/MyGit (Git-T03-HOL\_002)

\$ git log origin/Git-T03-HOL\_002 --oneline

a1b2c3d (HEAD -> Git-T03-HOL\_002, origin/Git-T03-HOL\_002) Added new feature file