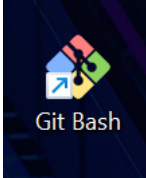


1. Git-HOL

Step 1: Setup your machine with Git Configuration

I have Git already installed and setup in my laptop



```
C:\Users\msaad>git --version
git version 2.49.0.windows.1

C:\Users\msaad>
```

```
MINGW64:/c/Users/msaad

msaad@Saad MINGW64 ~
$ 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=main
core.editor="C:\Users\msaad\AppData\Local\Programs\Microsoft VS Code\bin\code" -
-wait
user.name=saadm4n
user.email=saadm4n@gmail.com
```

In Command prompt,

```
D:\>mkdir GitDemo

D:\>dir
Volume in drive D has no label.
Volume Serial Number is 7C24-420D
```

```

D:\>cd GitDemo

D:\GitDemo>git init
Initialized empty Git repository in D:/GitDemo/.git/

D:\GitDemo>echo "Welcome to Git hands-on lab" > welcome.txt

D:\GitDemo>ls
welcome.txt

D:\GitDemo>git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        welcome.txt

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

D:\GitDemo>git add .

D:\GitDemo>git commit -m "initial commit"
[main (root-commit) cef8fce] initial commit
1 file changed, 1 insertion(+)
create mode 100644 welcome.txt

```

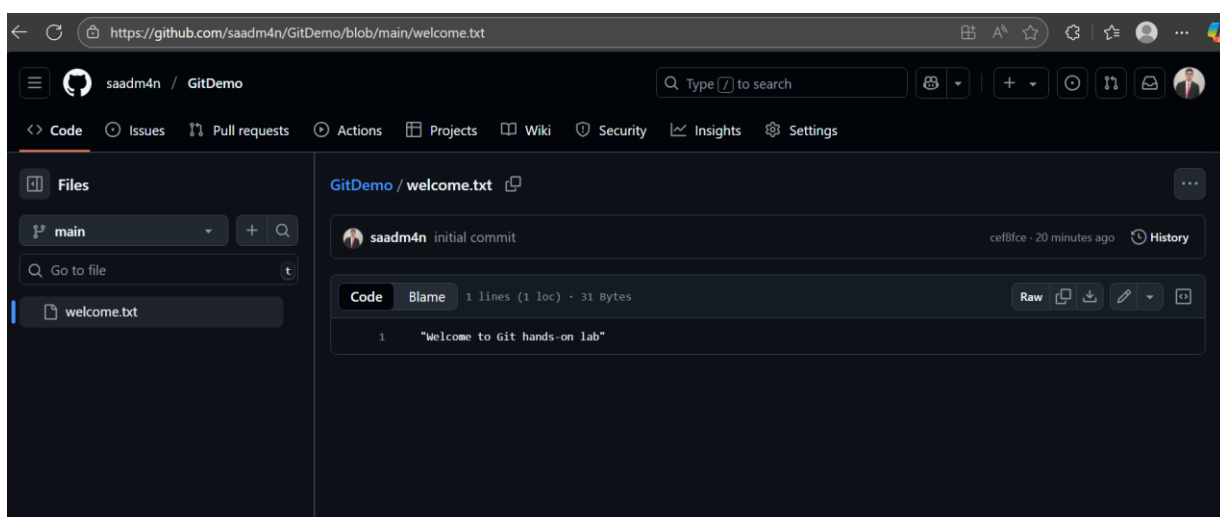
In VSCode

```

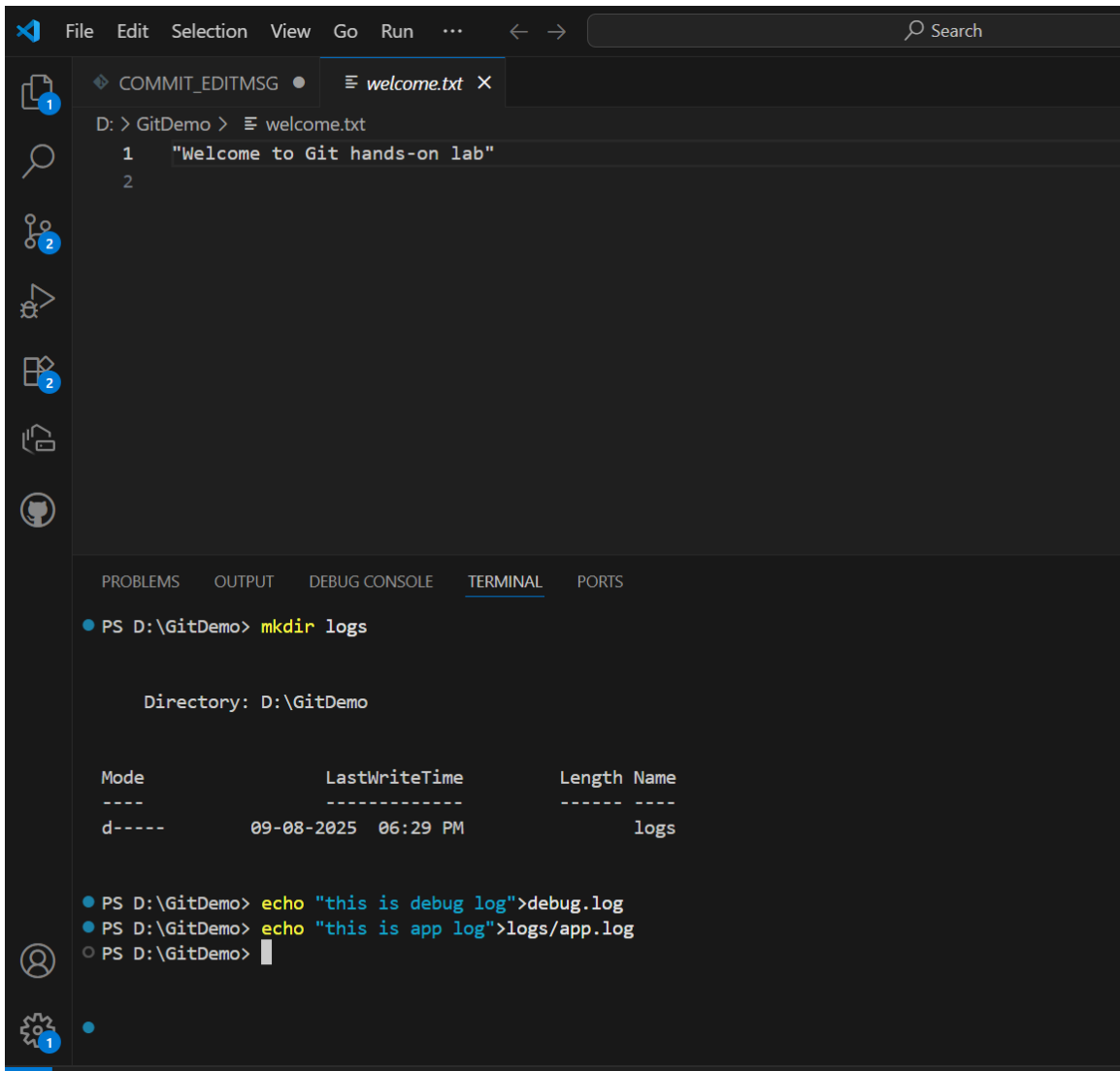
PS D:\GitDemo> git remote add origin https://github.com/saadm4n/GitDemo.git
>>
PS D:\GitDemo> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 239 bytes | 239.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/saadm4n/GitDemo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS D:\GitDemo>

```

Output:



2. Git-HOL



The screenshot shows the Visual Studio Code interface. The top bar includes the menu (File, Edit, Selection, View, Go, Run, ...) and a search bar. The Explorer sidebar on the left shows the file structure with icons for Explorer, Search, Source Control, Run and Debug, Extensions, and GitHub. The main editor area has two tabs: 'COMMIT_EDITMSG' and 'welcome.txt'. The 'welcome.txt' tab is active, showing the content: `1 "Welcome to Git hands-on lab"`. The bottom panel has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing a PowerShell session in the directory 'D:\GitDemo'. The commands and their outputs are as follows:

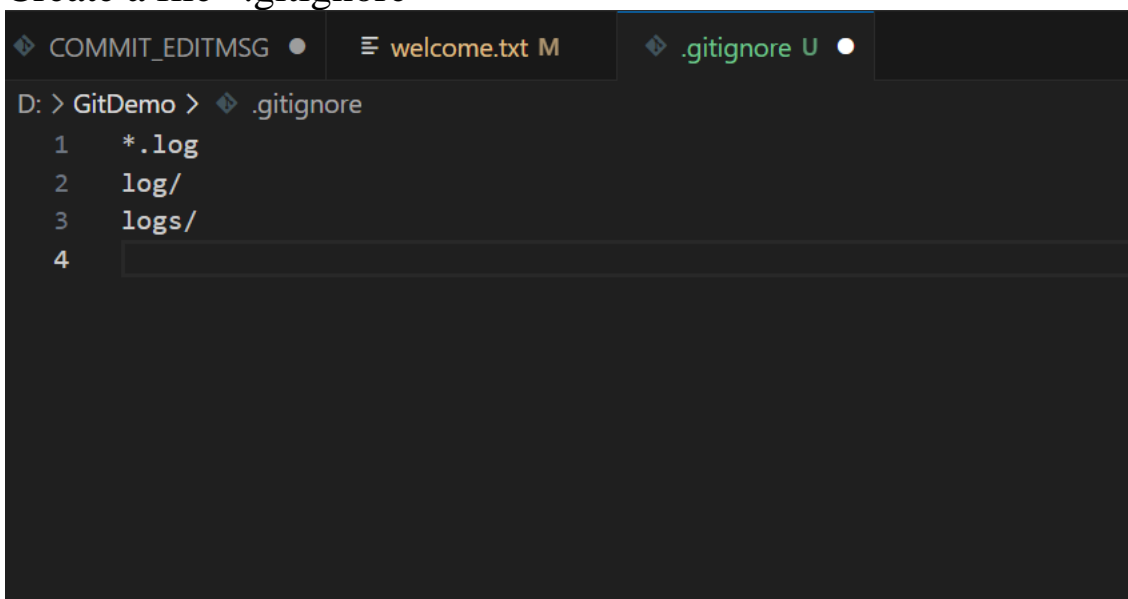
```
PS D:\GitDemo> mkdir logs

Directory: D:\GitDemo

Mode                LastWriteTime         Length Name
----                -
d-----          09-08-2025   06:29 PM             logs

PS D:\GitDemo> echo "this is debug log">debug.log
PS D:\GitDemo> echo "this is app log">logs/app.log
PS D:\GitDemo>
```

Create a file “.gitignore”



The screenshot shows the Visual Studio Code interface with the same setup as the previous one. The Explorer sidebar now shows an additional file, '.gitignore', which is highlighted. The main editor area has three tabs: 'COMMIT_EDITMSG', 'welcome.txt', and '.gitignore'. The '.gitignore' tab is active, showing the following content:

```
D: > GitDemo > .gitignore
1 *.log
2 log/
3 logs/
4
```

```
● PS D:\GitDemo> git add .gitignore
● PS D:\GitDemo> git commit -m "Add gitignore to ignore log files and folders"
[main 65bf411] Add gitignore to ignore log files and folders
1 file changed, 3 insertions(+)
create mode 100644 .gitignore
○ PS D:\GitDemo> █
```

```
● PS D:\GitDemo> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

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:   welcome.txt

no changes added to commit (use "git add" and/or "git commit -a")
○ PS D:\GitDemo> █
```

Now, we can see that git is ignoring the log files

```
● PS D:\GitDemo> git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/saadm4n/GitDemo.git
   cef8fce..65bf411  main -> main
○ PS D:\GitDemo> █
```

3. Git-HOL

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

• PS D:\GitDemo> git checkout main
M      welcome.txt
Already on 'main'
Your branch is up to date with 'origin/main'.
⊗ PS D:\GitDemo> git branch GitNewBranch
fatal: a branch named 'GitNewBranch' already exists
• PS D:\GitDemo> git checkout
M      welcome.txt
Your branch is up to date with 'origin/main'.
• PS D:\GitDemo> git branch
    GitNewBranch
* main
• PS D:\GitDemo> git checkout GitNewBranch
M      welcome.txt
Switched to branch 'GitNewBranch'
```

```
• PS D:\GitDemo> git branch
    GitNewBranch
* main
• PS D:\GitDemo> git checkout GitNewBranch
M      welcome.txt
Switched to branch 'GitNewBranch'
• PS D:\GitDemo> git branch
• * GitNewBranch
    main
• PS D:\GitDemo> echo "this is the content for the new branch">branchfile.txt
• PS D:\GitDemo> git add .
• PS D:\GitDemo> git commit -m "added file branchfile.txt to GitNewBranch"
[GitNewBranch f708eba] added file branchfile.txt to GitNewBranch
 2 files changed, 1 insertion(+)
 create mode 100644 branchfile.txt
```

```
• PS D:\GitDemo> git status
On branch GitNewBranch
nothing to commit, working tree clean
• PS D:\GitDemo> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

```

● PS D:\GitDemo> git diff GitNewBranch
diff --git a/branchfile.txt b/branchfile.txt
deleted file mode 100644
index 3b6eb4c..0000000
Binary files a/branchfile.txt and /dev/null differ
diff --git a/welcome.txt b/welcome.txt
index f47fea1..0918b2a 100644
--- a/welcome.txt
+++ b/welcome.txt
@@ -1,2 +1 @@
 "Welcome to Git hands-on lab"
-
● PS D:\GitDemo> git merge GitNewBranch
Updating 65bf411..f708eba
Fast-forward
 branchfile.txt | Bin 0 -> 82 bytes
 welcome.txt    | 1 +
 2 files changed, 1 insertion(+)

```

```

● PS D:\GitDemo> git branch -d GitNewBranch
Deleted branch GitNewBranch (was f708eba).
● PS D:\GitDemo> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

```

4. Git-HOL

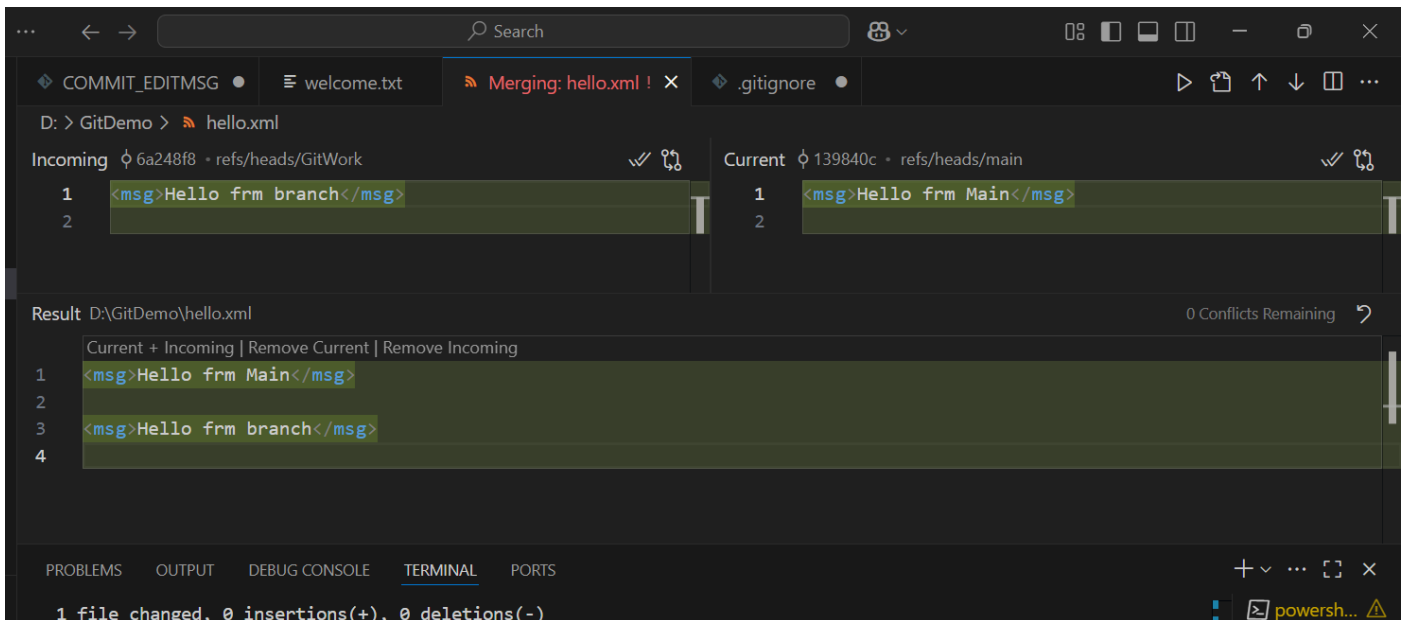
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

• PS D:\GitDemo> git branch
* main
• PS D:\GitDemo> git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
• PS D:\GitDemo> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
• PS D:\GitDemo> git checkout -b GitWork
Switched to a new branch 'GitWork'
• PS D:\GitDemo> echo "<msg>Hello frm branch</msg>">hello.xml
• PS D:\GitDemo> git add hello.xml
• PS D:\GitDemo> git commit -m "added hello.xml"
[GitWork 6a248f8] added hello.xml
```

```
• PS D:\GitDemo> git commit -m "added hello.xml"
[GitWork 6a248f8] added hello.xml
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello.xml
• PS D:\GitDemo> git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
• PS D:\GitDemo> echo "<msg>Hello frm Main</msg>">hello.xml
• PS D:\GitDemo> git add hello.xml
• PS D:\GitDemo> git commit -m "added hello.xml"
[main 139840c] added hello.xml
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello.xml
```

```
⊗ PS D:\GitDemo> git merge GitWork
warning: Cannot merge binary files: hello.xml (HEAD vs. GitWork)
Auto-merging hello.xml
CONFLICT (add/add): Merge conflict in hello.xml
Automatic merge failed; fix conflicts and then commit the result.
```



I have resolved the conflict by choosing accept the incoming change

```
● PS D:\GitDemo> git add hello.xml
● PS D:\GitDemo> git commit -m "resolved conflict in hello.xml"
[main 3b8c48f] resolved conflict in hello.xml
● PS D:\GitDemo> git branch -d GitWork
Deleted branch GitWork (was 6a248f8).
● PS D:\GitDemo> git push origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 20 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 1.11 KiB | 570.00 KiB/s, done.
Total 13 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/saadm4n/GitDemo.git
65bf411..3b8c48f main -> main
● PS D:\GitDemo>
```


5. Git-HOL

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS D:\GitDemo> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
● PS D:\GitDemo> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
● PS D:\GitDemo> git branch -a
* main
  remotes/origin/main
● PS D:\GitDemo> git pull origin main
From https://github.com/saadm4n/GitDemo
* branch                main                -> FETCH_HEAD
Already up to date.
● PS D:\GitDemo> git push origin main
● Everything up-to-date
○ PS D:\GitDemo> 
```

Thank You
