

Cognizant Deep Skilling - Digital Nurture 4.0 - Week 8

Git Hands-On Lab 1 – Basic Git Operations and Repository Management

Step 1: Setup your machine with Git Configuration

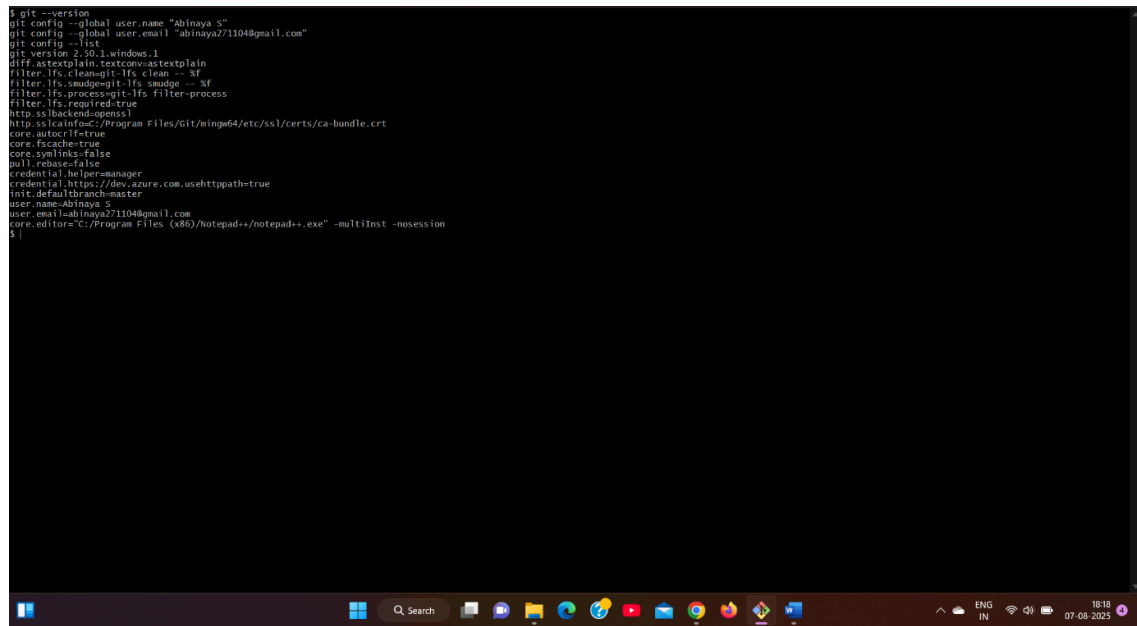
`git --version`

`git config --global user.name "Hobika G"`

`git config --global user.email "hobikagopal1704@gmail.com"`

`git config --list`

Output:



```
git --version
git config --global user.name "Abinaya S"
git config --global user.email "abinaya271104@gmail.com"
git config --list
git version 2.50.1.windows.1
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.fsmonitor=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Abinaya S
user.email=abinaya271104@gmail.com
core.editor=C:/Program Files (x86)/notepad+/notepad++.exe -multiInst -nosession
```

Step 2: Integrate notepad++.exe to Git and make it a default editor

`notepad++`

`git config --global core.editor "\"C:/Program Files`

`(x86)/Notepad++/notepad++.exe\" -multiInst -nosession"`

Output:

```
notepad++
git config --global core.editor "\"C:/Program Files (x86)/notepad++/notepad++.exe\" -multiInst -nosession"
```

Step 3: Add a file to source code repository

```
cd ~
```

```
mkdir GitDemo
```

```
cd GitDemo
```

```
git init
```

```
echo "Welcome to Git Hands-On Lab" > welcome.txt
```

```
ls
```

```
cat welcome.txt
```

```
git status
```

```
git add welcome.txt
```

```
git commit
```

git remote add origin https://gitlab.com/Hobika9597/gitdemo.git

git pull origin master

git push origin master

Output:

```
1 mkdir GitDemo
2 cd GitDemo
3 git init
4 Initialized empty Git repository in C:/abhinava/GitDemo/.git/
5 echo "welcome to Git Hands-On Lab" > welcome.txt
6 ls
7 cat welcome.txt
8 welcome.txt
9 welcome to Git Hands-On Lab
10 git status
11 git add welcome.txt
12 git commit
13 on branch master
14
15 No commits yet
16
17 Untracked files:
18   (use "git add <file>..." to include in what will be committed)
19
20     welcome.txt
21
22 nothing added to commit but untracked files present (use "git add" to track)
23 warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the next time Git touches it
24 (master (root-commit) b2d08c) First Commit
25 1 file changed, 1 insertion(+)
26 create mode 100644 welcome.txt
27 git remote add origin https://gitlab.com/Abi2711/gitdemo.git
28 git pull origin master
29 git push origin master
30 fatal: couldn't find remote ref master
31 enumerating objects: 3, done.
32 counting objects: 100% (3/3), done.
33 writing objects: 100% (3/3), 247 bytes | 247.00 KiB/s, done.
34 total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
35 remote:
36 remote: To create a merge request for master, visit:
37 remote:   https://gitlab.com/Abi2711/gitdemo/-/merge_requests/new?merge_requests[source_branch]=master
38 remote:
39 to https://gitlab.com/Abi2711/gitdemo.git
40 [new branch] master -> master
41 3 |
```

Git Hands-On Lab 2 – Git Ignore

Step 1: Create Unwanted Files

touch debug.log

mkdir log

touch log/system.log

Step 2: Create .gitignore File

touch .gitignore

Add:

***.log**

log/

Step 3: Check Git Status

git status

Step 4: Commit and Push

git add .gitignore

git commit -m "Added .gitignore to ignore .log files and log folder"

git push origin master

Output:

```
$ touch debug.log
$ mkdir log
$ touch log/system.log
$ touch .gitignore
$ notepad++ .gitignore
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
$ git add .gitignore
$ git commit -m "Added .gitignore to ignore .log files and log folder"
[master 101b7c7] Added .gitignore to ignore .log files and log folder
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 313 bytes | 313.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: To create a merge request for master, visit:
remote:   https://gitlab.com/Abi2711/gitdemo/-/merge_requests/new?merge_requests[source_branch]=master
remote:
To https://gitlab.com/Abi2711/gitdemo.git
   b3d398c..101b7c7  master -> master
$
```

Git Hands-On Lab 3 – Branching and Merging

1. Created New Branch

git checkout -b GitNewBranch

2. Listed Branches

git branch -a

3. Created a File and Committed

echo "This is from GitNewBranch" > branchfile.txt

git add branchfile.txt

git commit -m "Added branchfile.txt in GitNewBranch"

4. Switched Back

git checkout master

git diff master..GitNewBranch

Output:

```
5 git checkout -b GitNewBranch
Switched to a new branch 'GitNewBranch'
1 git branch -a
* GitNewBranch
master
remotes/origin/master
1 echo "This is from GitNewBranch" > branchfile.txt
1 git status
On branch GitNewBranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    branchfile.txt

nothing added to commit but untracked files present (use "git add" to track)
1 git add branchfile.txt
1 git commit -m "Added branchfile.txt in GitNewBranch"
warning: in the working copy of 'branchfile.txt', LF will be replaced by CRLF the next time Git touches it
[GitNewBranch 81c2258] Added branchfile.txt in GitNewBranch
1 file changed, 1 insertion(+)
create mode 100644 branchfile.txt
1 git checkout master
Switched to branch 'master'
1 git diff master..GitNewBranch
diff --git a/branchfile.txt b/branchfile.txt
new file mode 100644
index 0000000..b0d6bc
+++ b/branchfile.txt
@@ -0,0 +1 @@
+This is from GitNewBranch
1 git checkout -b GitNewBranch
Switched to a new branch 'GitNewBranch'
```

5. Merged

`git merge GitNewBranch`

6. Checked Logs

`git log --oneline --graph --decorate`

7. Deleted Branch and Pushed

`git branch -d GitNewBranch`

`git push origin master`

Output:

```
git merge GitNewBranch
Updating 101b767..84c2228
Fast-forward
 1 file changed, 1 insertion(+)
 create mode 100644 branchfile.txt
git log --oneline --graph --decorate
* 101b767 (HEAD, master, GitNewBranch) Added branchfile.txt in GitNewBranch
* 101b767 (HEAD, master, GitNewBranch) Added .gitignore to ignore .log files and log folder
* 101b767 (HEAD, master, GitNewBranch) First Commit
git branch -d GitNewBranch
Deleted branch GitNewBranch (was 84c2228).
git push origin master
Counting objects: 100% (4/4), done.
Delta compression using up to 24 threads
Compressing objects: 100% (2/2), done.
Total: 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: http://gitlab.com:8080/gitdemo/merge_requests/new/merge_requests/15/source_branch%3Dmaster
remote: 
To http://gitlab.com:8080/gitdemo.git
 101b767..84c2228 master -> master
```

Git Hands-On Lab 4 – Git Merge Conflict Resolution

1. Created a branch GitWork

`git checkout -b GitWork`

2. Added hello.xml with unique content in GitWork

`echo "<message>Hello from GitWork</message>" > hello.xml`

git add hello.xml

git commit -m "Added hello.xml in GitWork branch"

3. Switched to master and created conflicting file

git checkout master

echo "<message>Hello from master branch</message>" > hello.xml

git add hello.xml

git commit -m "Added hello.xml in master with different content"

4. Merged and got conflict

git merge GitWork

Output:

```
$ git checkout -b GitWork
Switched to a new branch 'GitWork'
$ echo "<message>Hello from Gitwork</message>" > hello.xml
$ git add hello.xml
$ git commit -m "Added hello.xml in Gitwork branch"
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it
[GitWork 6ad9810] Added hello.xml in Gitwork branch
1 file changed, 1 insertion(+)
 create mode 100644 hello.xml
$ git checkout master
Switched to branch 'master'
$ echo "<message>Hello from master branch</message>" > hello.xml
$ git add hello.xml
$ git commit -m "Added hello.xml in master with different content"
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it
[master 4195745] Added hello.xml in master with different content
1 file changed, 1 insertion(+)
 create mode 100644 hello.xml
$ git log --oneline --graph --decorate --all
* 4195745 (HEAD -> master) Added hello.xml in master with different content
* 6ad9810 (GitWork) Added hello.xml in Gitwork branch
* 84c3258 (origin/master) Added branchfile.txt in GitNew@branch
* 101b7c7 Added .gitignore to ignore .log files and log folder
* b3d308c First Commit
$ git merge GitWork
Auto-merging hello.xml
CONFLICT (add/add): Merge conflict in hello.xml
Automatic merge failed; fix conflicts and then commit the result.
$
```

5. Resolved conflict manually and committed

```
git add hello.xml
```

```
git commit -m "Resolved merge conflict in hello.xml"
```

6. Updated .gitignore and cleaned up

```
echo "*~" >> .gitignore
```

```
git add .gitignore
```

```
git commit -m "Updated .gitignore to ignore backup files"
```

7. Deleted merged branch

```
git branch -d GitWork
```

8. View the final graph

```
git log --online --graph --decorate
```

Output:

```
$ notepad++ hello.xml
$ git add hello.xml
$ git commit -m "Resolved merge conflict in hello.xml"
[master 529fe9b] Resolved merge conflict in hello.xml
$ echo "*~" >> .gitignore
$ git add .gitignore
$ git commit -m "Updated .gitignore to ignore backup files"
warning: in the working copy of .gitignore, it will be replaced by CRLF the next time Git touches it
[master c2e0b0] Updated .gitignore to ignore backup files
1 file changed, 1 insertion(+)
$ git branch
$ git branch -d Gitwork
Deleted branch Gitwork (was 6ad9810).
$ git log --online --graph --decorate
* c2e0b0 (HEAD -> master) Updated .gitignore to ignore backup files
* 529fe9b Resolved merge conflict in hello.xml
* 6ad9810 Added hello.xml in Gitwork branch
* | 4195745 Added hello.xml in master with different content
* /
* 84c3238 (origin/master) Added branchfile.txt in GitnewBranch
* 10107c7 Added .gitignore to ignore .log files and log folder
* b3d300c First commit
$
```


Git Hands-On Lab 5 – Git Cleanup & Push Lab

1. Switched to master branch

git checkout master

2. Checked git status

git status

3. Listed all branches

git branch -a

4. Pulled latest changes

git pull origin master

5. Pushed pending changes

git push origin master

6. Verified updates in GitLab repo

Visited:

<https://gitlab.com/Hobika9597/gitdemo>

Output:

```
$ git checkout master
Already on 'master'
$ git status
On branch master
nothing to commit, working tree clean
$ git branch -a
* master
  remotes/origin/master
$ git pull origin master
From https://gitlab.com/Abi2711/gitdemo
  * branch            master       -> FETCH_HEAD
Already up to date.
$ git push origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 16 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 1.21 KiB | 619.00 KiB/s, done.
total 12 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: To create a merge request for master, visit:
remote:   https://gitlab.com/Abi2711/gitdemo/-/merge_requests/new?merge_requests[source_branch%5D=master
remote:
To https://gitlab.com/Abi2711/gitdemo.git
  84c1258..cf2e6b0 master -> master
$
```