

COURSE : CLOUD COMPUTING LAB

INSTRUCTOR : SIR MUHAMMAD SHOAIB

SUBMITTED BY : Inshal Nasir
REG# 2023-BSE-038

LAB TASKS

Install Git on laptop



Task #01

Create a new private repository named Lab2 on GitHub.

Create a new repository
Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * / Repository name *
lab2 is available.

Great repository names are short and memorable. How about **cuddly-funicular**?

Description
14 / 350 characters

2 Configuration

Choose visibility * Choose who can see and commit to this repository

Add README READMEs can be used as longer descriptions. [About READMEs](#) Off

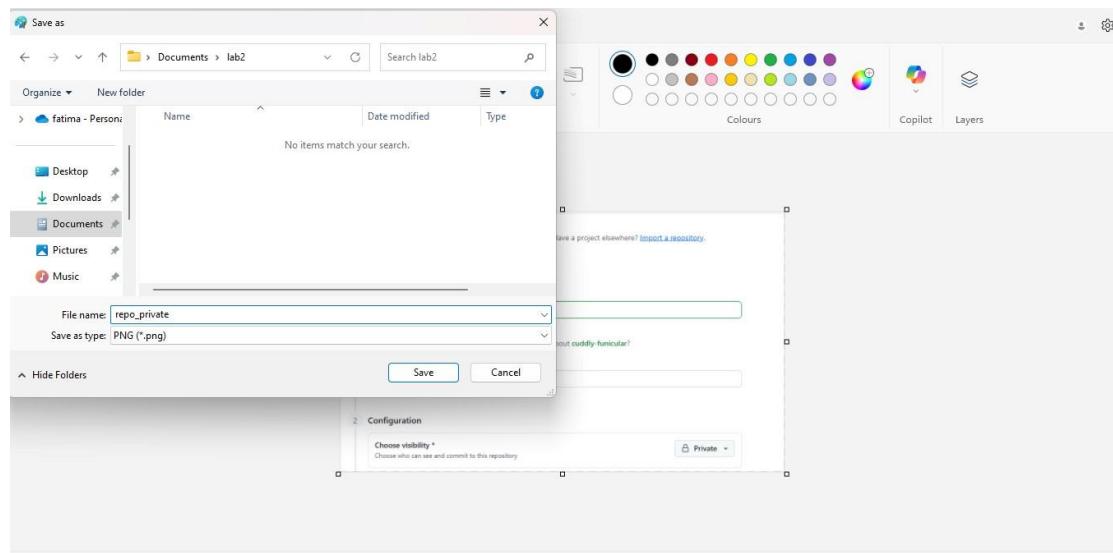
Add .gitignore .gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Clone lab2 repo

```
gulis@Laptop MINGW64 ~/Documents
$ git clone git@github.com:23-22411-018-png/lab2.git
Cloning into 'lab2'...
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

Take a screenshot of your repo settings showing it's private. Save as `repo_private.png` in your Lab2 repo.



TASK # 02

Generate a new SSH key using PowerShell

```
gulis@Laptop MINGW64 ~/Documents
$ ssh-keygen -t ed25519 -C "fatimakashif897@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/gulis/.ssh/id_ed25519):
Enter passphrase for "/c/Users/gulis/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/gulis/.ssh/id_ed25519
Your public key has been saved in /c/Users/gulis/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:7VMWIFuWLr7t6BIZltkvBKlp/IcmEE2Xh2XREIp7Io4 fatimakashif897@gmail.com
The key's randomart image:
+--[ED25519 256]--+
|    .++o+= |
|    =+oo.+ .|
|    ..o.^oO   |
|    =.O.O.o   |
|    oS=+O+o   |
|    .ooo*=+ . |
|    E..ooo .  |
|                |
+----[SHA256]----+
```

Add your SSH public key to GitHub (Settings > SSH and GPG keys). Take a screenshot of the added key as `github_sshkey.png`

Add new SSH Key

Title	key1
Key type	Authentication Key
Key	ssh-ed25519 AAAAC3NzaC1IzDI1NTESAAAAJ5cipKPGfYN6h4v1L+v1qG/6g9s2F9E68ivcjgOsNdd fatimakashif@gmail.com
Add SSH key	

.Clone your Lab2 repo using SSH

```
gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ git clone git@github.com:23-22411-018-png/lab2.git
Cloning into 'lab2'...
warning: You appear to have cloned an empty repository.
```

TASK#03 Set up your Git identity (this ensures all commits are linked to you)

```
gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ git config --global user.name "fatima"

gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ git config --global user.email "fatima@gmail.com"
```

Verify your configuration:

```
gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ 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=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=fatima
user.email=fatima@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/23-22411-018-png/lab2.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
```

TASK # 04

Navigate into your cloned repository folder.

```
gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ cd ~/Documents/lab2
```

Show hidden files and locate the .git directory.

```
gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ ls -a
./ ../ .git/ README.md git_config_list.png git_identity.png github_sshkey.png repo_private.png ssh_clone.png ssh_keygen.png
```

Explore what's inside

```
gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ ls -a .git
./ ../ COMMIT_EDITMSG HEAD config description hooks/ index info/ logs/ objects/ refs/
```

TASK#05

Delete the existing .git folder from your cloned repo using Git Bash

```
gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ rm -rf .git
```

Re-initialize the local git repository

```
gulis@Laptop MINGW64 ~/Documents/lab2
$ git init
Initialized empty Git repository in C:/Users/gulis/Documents/lab2/.git/
```

Add a file named README.md and commit it

```
gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ echo "# lab2" >> README.md

gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ git init
Reinitialized existing Git repository in C:/Users/gulis/Documents/lab2/.git/

gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

gulis@Laptop MINGW64 ~/Documents/lab2 (main)
$ git commit -m "first commit"
[main (root-commit) d610042] first commit
 1 file changed, 1 insertion(+)
  create mode 100644 README.md
```

Connect local repo to Github and push

```
gulis@Laptop MINGW64 ~/lab2 (master)
$ git remote add origin git@github.com:23-22411-018-png/lab2.git

gulis@Laptop MINGW64 ~/lab2 (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'github.com:23-22411-018-png/lab2.git'

gulis@Laptop MINGW64 ~/lab2 (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 233 bytes | 38.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/23-22411-018-png/lab2/pull/new/master
remote:
To github.com:23-22411-018-png/lab2.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

TASK # 06

Create a new file notes.txt and write a note.

```
gulis@Laptop MINGW64 ~
$ echo "This is my first note." > notes.txt
```

Check status. Save screenshot as status1.png.

```
gulis@Laptop MINGW64 ~/Documents/lab2 (master)
$ git status
```

Stage and commit. Save screenshot as commit_notes.png.

```
gulis@Laptop MINGW64 ~/Documents/lab2 (master)
$ git add notes.txt
warning: in the working copy of 'notes.txt', LF will be replaced by CRLF the next time Git touches it

gulis@Laptop MINGW64 ~/Documents/lab2 (master)
$ git commit -m "Update notes.txt"
[master e5a9d46] Update notes.txt
 1 file changed, 1 insertion(+)
```

Edit notes.txt and repeat status/commit steps.

```

gulis@Laptop MINGW64 ~/Documents/lab2 (master)
$ echo "This is an update to my notes." >> notes.txt

gulis@Laptop MINGW64 ~/Documents/lab2 (master)
$ git status
On branch master
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:   notes.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    delete_git.png
    first_commit.png
    git_config_list.png
    git_folder.png
    git_identity.png
    github_sshkey.png
    hit_init.png
    repo_private.png
    ssh_clone.png
    ssh_keygen.png

no changes added to commit (use "git add" and/or "git commit -a")

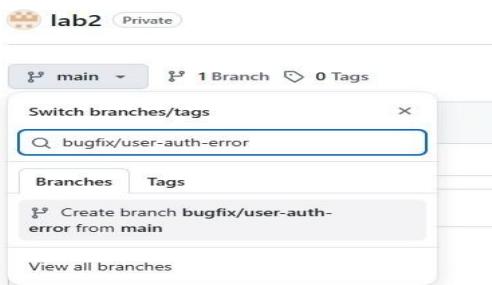
gulis@Laptop MINGW64 ~/Documents/lab2 (master)
$ git add notes.txt
warning: in the working copy of 'notes.txt', LF will be replaced by CRLF the next time Git touches it

gulis@Laptop MINGW64 ~/Documents/lab2 (master)
$ git commit -m "Update notes.txt"
[master e5a9d46] Update notes.txt
 1 file changed, 1 insertion(+)

```

TASK # 07

On GitHub (web interface), create a branch named bugfix/user-auth-error.



Pull the branch to your local repository to sync.

```

gulis@Laptop MINGW64 ~/lab2 (master)
$ git pull origin bugfix/user-auth-error
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 870 bytes | 48.00 KiB/s, done.
From github.com:23-22411-018-png/lab2
 * branch      bugfix/user-auth-error  -> FETCH_HEAD
 * [new branch] bugfix/user-auth-error -> origin/bugfix/user-auth-error
fatal: refusing to merge unrelated histories

```

TASK # 08

Create a branch named feature/db-connection using Git Bash.Push the branch to the remote repository.

```

gulis@Laptop MINGW64 ~/lab2 (master)
$ git checkout -b feature/db-connection
Switched to a new branch 'feature/db-connection'

gulis@Laptop MINGW64 ~/lab2 (feature/db-connection)
$ git push origin feature/db-connection
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/db-connection' on GitHub by visiting:
remote:     https://github.com/23-22411-018-png/lab2/pull/new/feature/db-connection
remote:
To github.com:23-22411-018-png/lab2.git
 * [new branch]      feature/db-connection -> feature/db-connection

```

TASK # 09

Create and switch to a branch feature-1

```
gulis@Laptop MINGW64 ~/lab2 (feature/db-connection)
$ git checkout -b feature-1
Switched to a new branch 'feature-1'
```

Modify main.py (add a function) and commit

```
gulis@Laptop MINGW64 ~/lab2 (feature-1)
$ git add main.py

gulis@Laptop MINGW64 ~/lab2 (feature-1)
$ git commit -m "Add new function to main.py"
[feature-1 95cec15] Add new function to main.py
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.py
```

Switch back to main and merge

```
gulis@Laptop MINGW64 ~/lab2 (feature-1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

gulis@Laptop MINGW64 ~/lab2 (master)
$ git merge feature-1
Updating 6c9f659..95cec15
Fast-forward
  main.py |  0
  1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.py
```

Push all branches

```
gulis@Laptop MINGW64 ~/lab2 (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 284 bytes | 40.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:23-22411-018-png/lab2.git
  6c9f659..95cec15  master -> master

gulis@Laptop MINGW64 ~/lab2 (master)
$ git push origin feature-1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-1' on GitHub by visiting:
remote:     https://github.com/23-22411-018-png/lab2/pull/new/feature-1
remote:
To github.com:23-22411-018-png/lab2.git
 * [new branch]      feature-1 -> feature-1
```

TASK # 12

Create the following branches to simulate a professional branching strategy

```
gulis@Laptop MINGW64 ~/lab2 (master)
$ git checkout master
Already on 'master'
Your branch is up to date with 'origin/master'.

gulis@Laptop MINGW64 ~/lab2 (master)
$ git branch develop
gulis@Laptop MINGW64 ~/lab2 (master)
$ git branch staging
gulis@Laptop MINGW64 ~/lab2 (master)
$ git branch feature/login
gulis@Laptop MINGW64 ~/lab2 (master)
$ git branch bugfix/typo
gulis@Laptop MINGW64 ~/lab2 (master)
$ git branch
  bugfix/typo
  develop
  feature-1
  feature/db-connection
  feature/login
* master
  staging
```

Documentation with Screenshots

```
gulis@Laptop MINGW64 ~/lab2 (master)
$ git checkout feature/login
Switched to branch 'feature/login'

gulis@Laptop MINGW64 ~/lab2 (feature/login)
$ git add main.py

gulis@Laptop MINGW64 ~/lab2 (feature/login)
$ git commit -m "Add login feature"
[feature/login bd3df14] Add login feature
 1 file changed, 2 insertions(+)

gulis@Laptop MINGW64 ~/lab2 (feature/login)
$ git checkout develop
Switched to branch 'develop'

gulis@Laptop MINGW64 ~/lab2 (develop)
$ git merge feature/login
merge: feature/login - not something we can merge

gulis@Laptop MINGW64 ~/lab2 (develop)
$ git merge feature/login
Updating 95cec15..bd3df14
Fast-forward
  main.py | 2 ++
  1 file changed, 2 insertions(+)

gulis@Laptop MINGW64 ~/lab2 (develop)
$ git checkout staging
Switched to branch 'staging'

gulis@Laptop MINGW64 ~/lab2 (staging)
$ git merge develop
Updating 95cec15..bd3df14
Fast-forward
  main.py | 2 ++
  1 file changed, 2 insertions(+)
```

```
gulis@Laptop MINGW64 ~/lab2 (staging)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

gulis@Laptop MINGW64 ~/lab2 (master)
$ git merge staging
Updating 95cec15..bd3df14
Fast-forward
  main.py | 2 ++
  1 file changed, 2 insertions(+)
```

TASK # 13

Delete Remote Branch After Merge

Your branches					
Branch	Updated	Check status	Behind	Ahead	Pull request
feature-1	41 minutes ago	1 2			
master	42 minutes ago	1 2			
feature/db-connection	Deleted now	1 1			
bugfix/user-auth-error	3 days ago	0 0			

Update Local Repository

```
gulis@Laptop MINGW64 ~/lab2 (master)
$ git checkout master
Already on 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
```

```
gulis@Laptop MINGW64 ~/lab2 (master)
$ git pull origin master
From github.com:23-22411-018-png/lab2
 * branch            master      -> FETCH_HEAD
Already up to date.
```

```
gulis@Laptop MINGW64 ~/lab2 (master)
$ git branch -d feature/db-connection
Deleted branch feature/db-connection (was 6c9f659).
```

```
gulis@Laptop MINGW64 ~/lab2 (master)
$ git branch
  bugfix/typo
  develop
  feature-1
  feature/login
* master
  staging
```

My lab2 document

