## **Objectives**

Familiar with Git commands like git init, git status, git add, git commit, git push, and git pull.

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

## **Prerequisites**

* Install Git Bash client in your machine

Notes\*:

|  |
| --- |
| Please follow the below steps for creating a free account in GitHub.  Don’t use cognizant credentials to login to GitHub. |

Estimated time to complete this lab: **30 minutes.**

**Step 1: Setup your machine with Git Configuration**

To create a new repository, signup with GitLab and register your credentials

Login to GitLab and create a “GitDemo” project

1. To check if Git client is installed properly: Open Git bash shell and execute



If output shows Git with its version information that indicates, that Git Client installs properly.

1. To configure user level configuration of user ID and email ID execute



1. To check if the configuration is properly set, execute the following command.



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

1. To check, if notepad++.exe execute from Git bash



If Git bash could not able to recognize notepad++ command that implies notepad++.exe is note added to the environment path variable.

To add path of notepad++.exe to environment variable, go to control panel -> System -> Advanced System settings. Go to Advanced tab -> Environment variables -> Add path of notepad++.exe to the path user variable by clicking on “Edit”



1. Exit Git bash shell, open bash shell and execute



Now, notepad++ will open from Git bash shell

1. To create an alias command for notepad++.exe, execute



It will open notepad++ from bash shell, and create a user profile by adding the line in notepad++



1. To configure the editor, execute the command



1. To verify if notepad++ is the default editor, execute the command



Here ‘-e’ option implies editor

It will show the entire global configuration as shown below,



**Step 3: Add a file to source code repository**

1. Open Git bash shell and create a new project “**GitDemo**” by executing the command



1. Git bash initializes the “**GitDemo**” repository. To verify, execute the command



It will display all the hidden files in the Git “working directory”.

1. To create a file **“welcome.txt”** and add content to the file, execute the command



1. To verify if the file “welcome.txt” is created, execute



1. To verify the content, execute the command



1. Check the status by executing



Now the file **“welcome.txt”** is available in Git “working directory”

1. To make the file to be tracked by Git repository, execute the command



1. To add multi line comments, we are opening default editor to comment. Execute the command



Notepad++ editor will open and to add multi-line comment with default editor

1. To check if local and “Working Directory” git repository are same, execute git status



**welcome.txt** is added to the local repository.

1. Signup with GitLab and create a remote repository **“GitDemo”**
2. To pull the remote repository, execute

git pull origin master

1. To push the local to remote repository, execute

git push origin master

**Solution :**

**1. Verify Git installation**

Open **Git Bash** and run:

git –version

**2. Configure your Git username and email (user-level)**

git config --global user.name "Your Name"

git config --global user.email [youremail@example.com](mailto:youremail@example.com)

**3. Verify configuration**

git config –list

**Step 2: Integrate Notepad++ as Git default editor**

**1. Check if Notepad++ runs from Git Bash**

notepad++

If **command not found**, add its path to the environment variables:

* + Go to **Control Panel → System → Advanced system settings → Environment Variables**.
  + Edit the **Path** variable and add the folder path of notepad++.exe (e.g., C:\Program Files\Notepad++).

Restart Git Bash and try:

notepad++

**2. Create an alias for Notepad++**

alias np='notepad++'

Now you can type:

np

to open Notepad++.

**3. Set Notepad++ as default Git editor**

git config --global core.editor "notepad++"

**4. Verify default editor**

git config --global -e

**Step 3: Add a File to the Repository**

**1. Create a new folder and initialize Git**

mkdir GitDemo

cd GitDemo

git init

**2. Verify .git folder exists**

ls -a

**3. Create a file and add content**

echo "Welcome to Git" > welcome.txt

**4. View file list**

ls

**5. View file content**

cat welcome.txt

6. Check status

git status

7. Stage the file

git add welcome.txt

**8. Commit with multi-line message (opens Notepad++)**

git commit

Write your commit message in Notepad++, save, and close.

**9. Verify repository state**

git status

It should show **nothing to commit**.

**Step 4: Push to GitLab**

1. Create a new **empty project** named GitDemo in **GitLab**.
2. Link your local repository to the remote:

git remote add origin https://gitlab.com/yourusername/GitDemo.git

1. Pull the remote (to ensure sync):

git pull origin master

1. Push local changes to remote:

git push origin master