# Git Hands-On Lab Guide

## 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

1. To create a new repository, signup with GitLab and register your credentials.  
2. Login to GitLab and create a “GitDemo” project.  
  
3. To check if Git client is installed properly: Open Git bash shell and execute:  
 ```  
 git --version  
 ```  
 If output shows Git with its version information, Git Client is installed properly.  
  
4. To configure user level configuration of user ID and email ID, execute:  
 ```  
 git config --global user.name "Your Name"  
 git config --global user.email "your.email@example.com"  
 ```  
  
5. To check if the configuration is properly set, execute:  
 ```  
 git config --list  
 ```

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

1. To check if notepad++.exe executes from Git bash, run:  
 ```  
 notepad++  
 ```  
 If Git bash could not recognize the command, it means Notepad++ is not added to the environment path variable.  
  
2. To add path of notepad++.exe to environment variable:  
 - Go to Control Panel -> System -> Advanced System settings.  
 - Go to Advanced tab -> Environment Variables.  
 - Edit the Path user variable and add the path of notepad++.exe.  
  
3. Exit Git bash shell, reopen it and execute:  
 ```  
 notepad++  
 ```  
 Now, Notepad++ will open from Git bash shell.  
  
4. To create an alias command for notepad++.exe, execute:  
 ```  
 alias notepad++='/c/Program\ Files/Notepad++/notepad++.exe'  
 ```  
 This allows opening Notepad++ from bash shell easily.  
  
5. To configure the editor as default for Git, execute:  
 ```  
 git config --global core.editor "notepad++"  
 ```  
  
6. To verify if notepad++ is the default editor, execute:  
 ```  
 git config --global --get core.editor  
 ```

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

1. Open Git bash shell and create a new project “GitDemo” by executing:  
 ```  
 mkdir GitDemo  
 cd GitDemo  
 git init  
 ```  
 Git bash initializes the “GitDemo” repository.  
  
2. To verify, execute:  
 ```  
 ls -a  
 ```  
 It will display all the hidden files in the Git working directory.  
  
3. To create a file “welcome.txt” and add content to the file, execute:  
 ```  
 echo "Welcome to Git Demo" > welcome.txt  
 ```  
  
4. To verify if the file “welcome.txt” is created, execute:  
 ```  
 ls  
 ```  
  
5. To verify the content, execute:  
 ```  
 cat welcome.txt  
 ```  
  
6. Check the status by executing:  
 ```  
 git status  
 ```  
 The file “welcome.txt” is untracked.  
  
7. To add the file to be tracked by Git repository, execute:  
 ```  
 git add welcome.txt  
 ```  
  
8. To commit with multi-line comments, execute:  
 ```  
 git commit  
 ```  
 This will open Notepad++ editor for you to add multi-line commit messages.  
  
9. To check if local and working directory git repository are same, execute:  
 ```  
 git status  
 ```  
 The file “welcome.txt” is now added to the local repository.  
  
10. Signup with GitLab and create a remote repository named “GitDemo”.  
  
11. To pull the remote repository, execute:  
 ```  
 git pull origin main  
 ```  
  
12. To push the local repository to remote, execute:  
 ```  
 git push origin main  
 ```