GIT_1:

**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

```
$ git version
git version 2.21.0.windows.1
```

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

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

```
$ git config --global user.name "username"
$ git config --global user.email "username@cognizant.com"
```

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

```
$ git config --global --list
user.name=username
user.email=username@cognizant.com
```

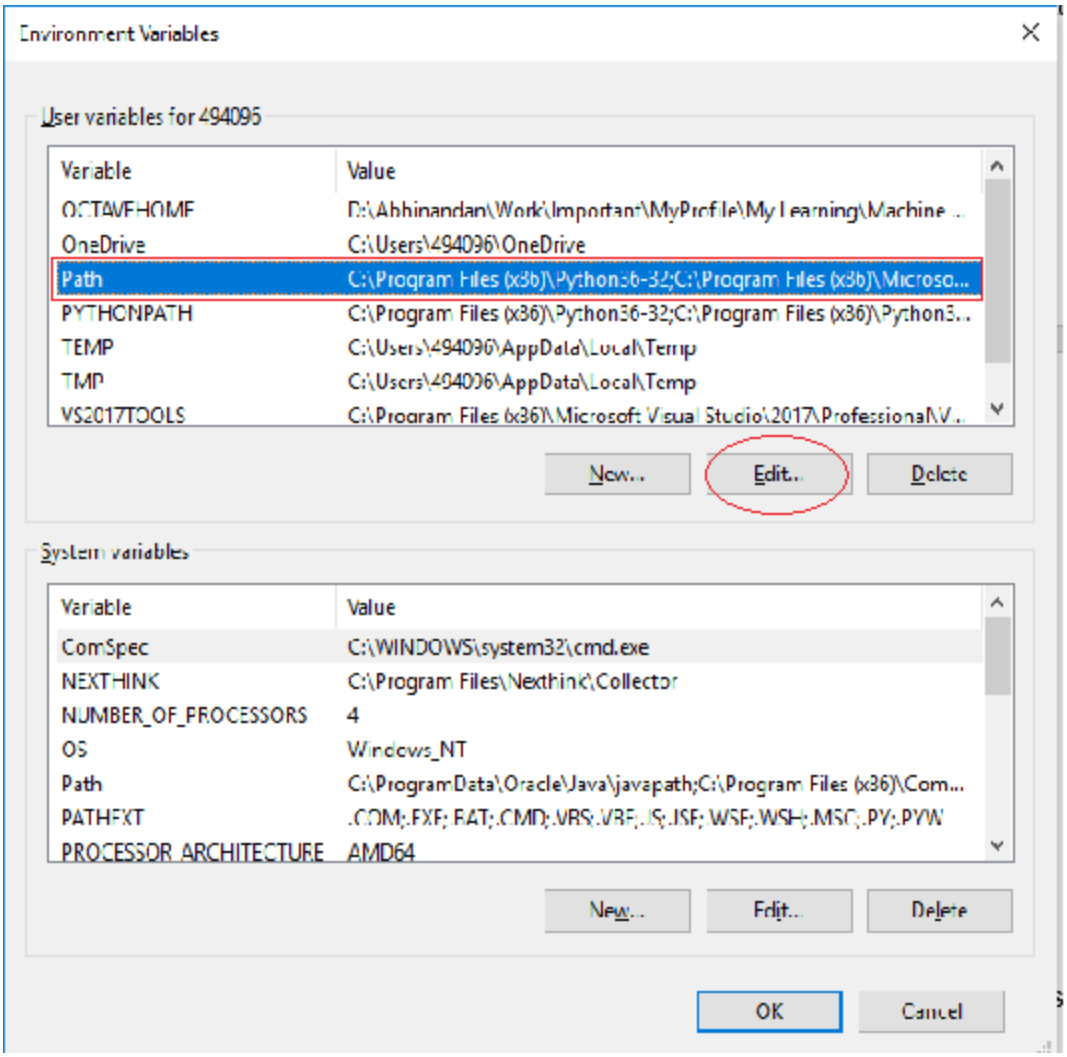
**Step 2: Integrate notepad++.exe to Git and make it a default editor**

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

```
$ notepad++
bash: notepad++: command not found
```

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”




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

```
$ notepad++
```

Now, notepad++ will open from Git bash shell

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

```
$ notepad++.exe bash -profile
```

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

```
alias npp='notepad++.exe -multiInst -nosession'
```

4. To configure the editor, execute the command

```
$ git config --global core.editor "notepad++.exe -multiInst -nosession"
```

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

```
$ git config --global -e
hint: Waiting for your editor to close the file...
```

Here '-e' option implies editor

It will show the entire global configuration as shown below,

```
[user]
    name = username
    email = username@cognizant.com
[core]
    editor = notepad++.exe -multiInst -nosession
```

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

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

```
$ git init GitDemo
Initialized empty Git repository in D:/Development_Avecto/GitDemo/.git/
```

2. Git bash initializes the "**GitDemo**" repository. To verify, execute the command

```
$ ls -al
total 8
drwxr-xr-x 1        1049089 0 Jan 13 11:54 ./
drwxr xr x 1        1049089 0 Jan 13 11:54 ../
drwxr-xr-x 1        1049089 0 Jan 13 11:54 .git/
```

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 the command

```
$ echo "Welcome to the version control" >> welcome.txt
```

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

```
$ ls -al
total 9
drwxr-xr-x 1 494096 1049089  0 Jan 13 12:02 ./
drwxr-xr-x 1 494096 1049089  0 Jan 13 11:54 ../
drwxr-xr-x 1 494096 1049089  0 Jan 13 12:01 .git/
-rw-r--r-- 1 494096 1049089 31 Jan 13 12:02 welcome.txt
```

5. To verify the content, execute the command

```
$ cat welcome.txt
Welcome to the version control
```

6. Check the status by executing

```
$ git status
On branch master

No commits yet

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

        welcome.txt
```

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

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

```
$ git add welcome.txt
warning: LF will be replaced by CRLF in welcome.txt.
The file will have its original line endings in your working directory
```

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

```
$ git commit
```

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

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

```
$ git status
On branch master
nothing to commit, working tree clean
```

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

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

git pull origin master

12. To push the local to remote repository, execute

git push origin master