1. 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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXctXGrwRo8EX7mRJiUa17kakStRUpRtvVkTTjXMqwGDnlZ5NLNOfYFVZE3mk7folgYTbq03aBB1VZaABoFX5A-vG0L2KDCYOEjUNrb16nOA-h4lH2lDGbwJ8aUUodnxkqVm7bCzfrMXVTxjTVZULgA?key=Y2ACmHWyKwSK0-9_IGMqtg

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXceoUBSYBrYcaSbA5EpDtj2sD-lw4hAcnb7BInnzIS63gmayFhJCPcOGQ77JsYJ3KSeuCQA_Ynxz3P1yAbWb6qUZOlC9qSwyTfsWdv4pC95_yOqffvjpDVXdgc2JtUvx5L781ofKLNX5KGKP6Vi-g?key=Y2ACmHWyKwSK0-9_IGMqtghttps://lh7-rt.googleusercontent.com/docsz/AD_4nXcbco5AwrPAu51nekeXOl5-1-GhF4vFS9XqtG1s772oqRTu_YiCYUVL50W_HTUO4FW973NjGxc4FtmRxSopLqXPoDhl_J1eqDEZGIyh0R92BZbXTgaUaRHCxv2SKMVlBHqYtoBq65l-k3GwZxkEF_0?key=Y2ACmHWyKwSK0-9_IGMqtg

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXeI1Lwkn21K9YFUVmpEDq7JwnmikgynZ4_dA4x0qrEV-NGUJrKo1128lVIwS9gJR9g_cfdBc-PW4JuU5esc735aHd8_IMf373JMozuhsz4R6uCgM4YaJWGXm5RuvYpYW3khG4iwNspfvE_0_NuXAh4?key=Y2ACmHWyKwSK0-9_IGMqtg

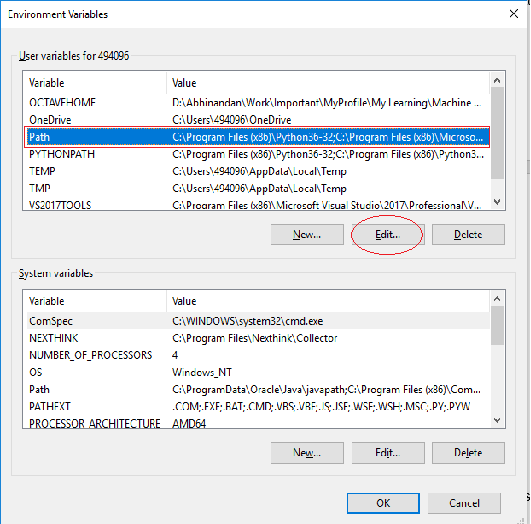
**Step 2: Integrate notepad++.exe to Git and make it a default editor**

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXerT0dEHzXKe-h71mi2Vbi41ExoTlMlBQ8Bhuxw0Eyhg4exhMLRiXRbQCXQ1h-_aYOVH3FvJvjlMkPy6MDjwaltgO4dx_MCouRAwGGFPqEeyg1DpKClMgjkl45wERqAVpLwttCX9VwXCE9HeBXejQ?key=Y2ACmHWyKwSK0-9_IGMqtg

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXd0AClrwK-XHvLvDwpUmZm03_5Pt49UcIZE3Xj5njUCFPBPgot9tWP2tRCD6JYzV98lhIOQddvMPWecLEv6CmuEmWdcEiRQiRzTOEGNFtDMZJ1IImb-jU282HRWxc7iR98kSF90hLCXUe8xjD6ytw?key=Y2ACmHWyKwSK0-9_IGMqtg

Now, notepad++ will open from Git bash shell

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXd3Jz77iEjsx4dgottcbZKQw2uZXGPxiAxRXzvwhmhnmoJMp3LI0NNyfKor8eYUmcoQd7YAYvY9x__tXHNzZtXY0DjVpO56Pfu2xtW2axGMBecEFATPZPA3fE69MmGcnoe7AOVjrSKrbccK8Ea-Eg?key=Y2ACmHWyKwSK0-9_IGMqtg

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXcgpCOifYrP5VzXJ-HZt5pjqVMApw1BaiRVwiUJMlMU7WkyOIesruPrk_PjjivimFq_YZeS9S0R86kMEEH6ZpSRYwkVYoKQsvvzJgT5Bq-E802bHatoppMy4kb6anjIFZDcDkDiBLB7KXXOTu64Wg?key=Y2ACmHWyKwSK0-9_IGMqtg

1. To configure the editor, execute the command

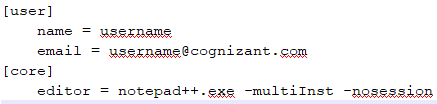
https://lh7-rt.googleusercontent.com/docsz/AD_4nXeW8IoT6DUFEekeZDg9cOjgBpmXRN6vdJzmYugjqjOz5gtoCYZe9r0IM-6iVlz6jnUqipbL-u4xXSySL07TsW9GfXOUj_v9ZAmX7iOF7zAjGQMyYcAc3mg9BCvasyepl7dmZAAogBzqPYMbj-3E7g?key=Y2ACmHWyKwSK0-9_IGMqtg

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXc9HdyiALLI9sGViVWJDVnMRb-a5wDMF6C72DD15OA37snI4izOtVHZxsHky12LYUrI4p2VpDcrBpCFHmvgDzeRZjhdVsvasHWnx6LQGNwits4PBTBhfQTAb5Zz7GS_MzxSJSYD5cOGrKdpo9v18no?key=Y2ACmHWyKwSK0-9_IGMqtg

Here ‘-e’ option implies editor

It will show the entire global configuration as shown below,

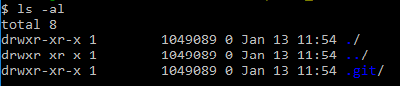


**Step 3: Add a file to source code repository**

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXeyQEEBScrUWhBebij1OQA2ZMLi8FP3UZx86W9uYQQIcHnwyvY0KQtXBhqE6t_UAPehVEliejQVNx65-4778gEDJr-Q8cV0Pvo4HYybqhou0TaN1jyD5Bpgl2LeqXzXpHKj_avv7bWYCS0Hg_I0rg?key=Y2ACmHWyKwSK0-9_IGMqtg

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

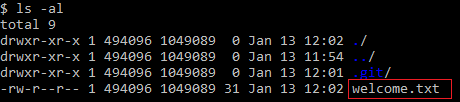


It will display all the hidden files in the Git  “working directory”.

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXcEo3cNrCJe4BD8DzdhFQ1oK7h-Mx7sqbv1Hpbm1GmODzcnrlxwkSRx9mAx_7KxheCQ1vWzmm6XazC5a9nrd4taKuI1Mfy7tsyUScvmfiAunfTFu5TUBDApQJuHd_Um77IOCpX8nNw39R9eFEwpVA?key=Y2ACmHWyKwSK0-9_IGMqtg

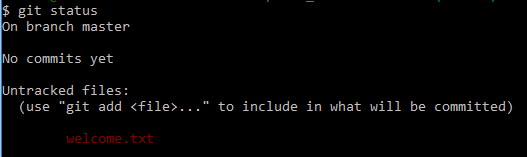
1. To verify if the file “welcome.txt” is created, execute



1. To verify the content, execute the command

https://lh7-rt.googleusercontent.com/docsz/AD_4nXfdIDFVAuvz1qhgL4wRzl83BStunztl2GfLWzruCuwijgvORW-_bsv8Pc8x37zhY7OOBoJit243AlM7buHsS2z4F5P5PMpvYqwf43mu60mw05oVmL0XKqQt0MfDI2haRzb3n3NYDw-LPYqQZA5B3w?key=Y2ACmHWyKwSK0-9_IGMqtg

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXfoiR41E2SGfFNizus9zHhPhU7HbF3VgweJvkOzYb3vjvpbYKk5uOAT5Etn-6lYvbfcs4f5Q5X5tG27sbS3w81mRaU14upcAiUYV0vLMau9I1XvH0fsUUIOCO-WCjcldUl34Awu_mGg8IY95UcSXjw?key=Y2ACmHWyKwSK0-9_IGMqtg

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXdUB5iTlQr4UhojNTjYNTTC3FnpbvVJ0RVxk2iyvXr_nxRikvyZdhtsq7M6mIF0huPQtakynxfFnVbbK_sl5GCmmQUTTmLEXy9ZZIBBbtw7BWs1eYG7IczSVszR_tmHOLg_efwIvqFEdaY-WmXEM9E?key=Y2ACmHWyKwSK0-9_IGMqtg

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

https://lh7-rt.googleusercontent.com/docsz/AD_4nXdVEPJSj6JxiNIafOl6M92kZT34FghYdTJ8hzgsITBJTE1XMO3xJMPPdS7RzX7NjWjfmpzrUcRzX2VkbKCyrsvlCxz7jj_5fiZcbyQoF3Y6SDuIchvvk9Ow7RDj4GRARoD9y8XVrUQ5e2U4-QFMuEw?key=Y2ACmHWyKwSK0-9_IGMqtg

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

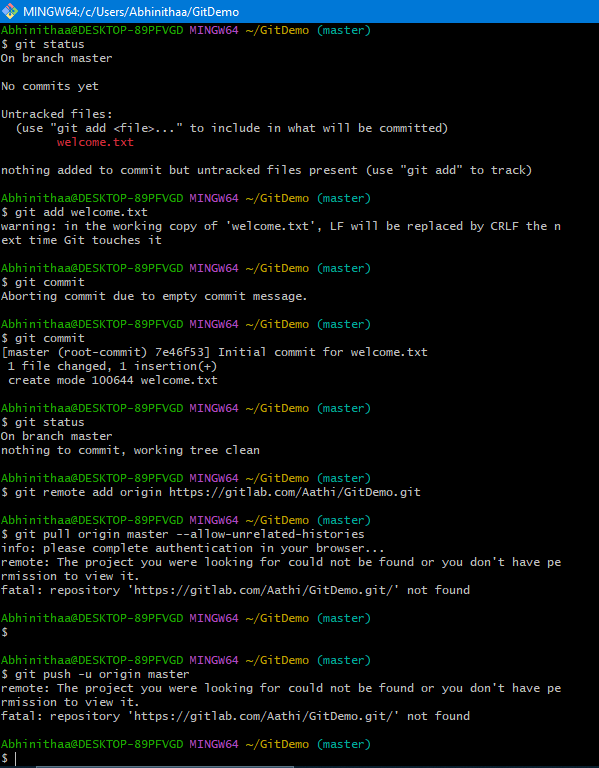
1. Signup with GitLab and create a remote repository **“GitDemo”**
2. To pull the remote repository, execute

git pull origin master

1. To push the local to remote repository, execute

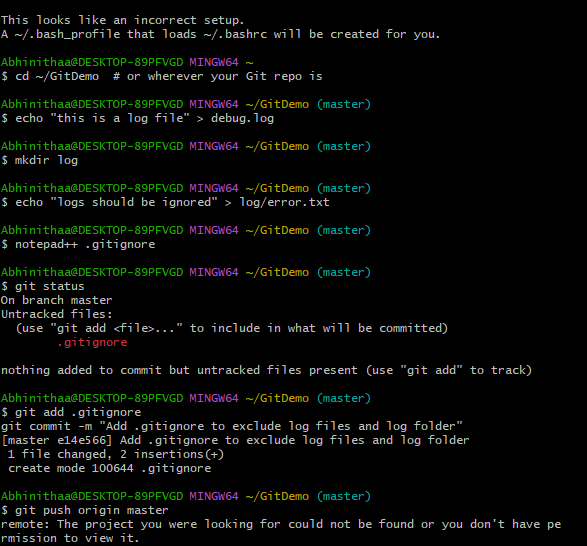
git push origin master



****

1. Create a **“.log”** file and a **log folder** in the working directory of Git. Update the **.gitignore** file in such a way that on committing, these files (.log extensions and log folders) are ignored.

Verify if the git status reflects the same about working directory, local repository and git repository.

****

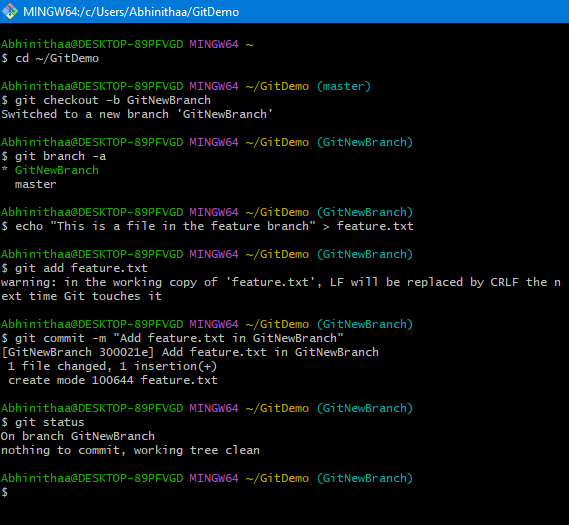
1. Please follow the instruction to complete the hands-on. Each instruction expects a command for the Git Bash.

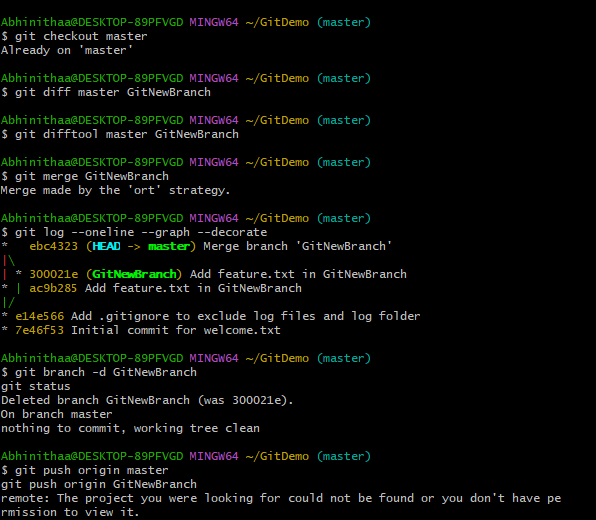
**Branching:**

1. Create a new branch **“GitNewBranch”.**
2. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with **“git status”** command.

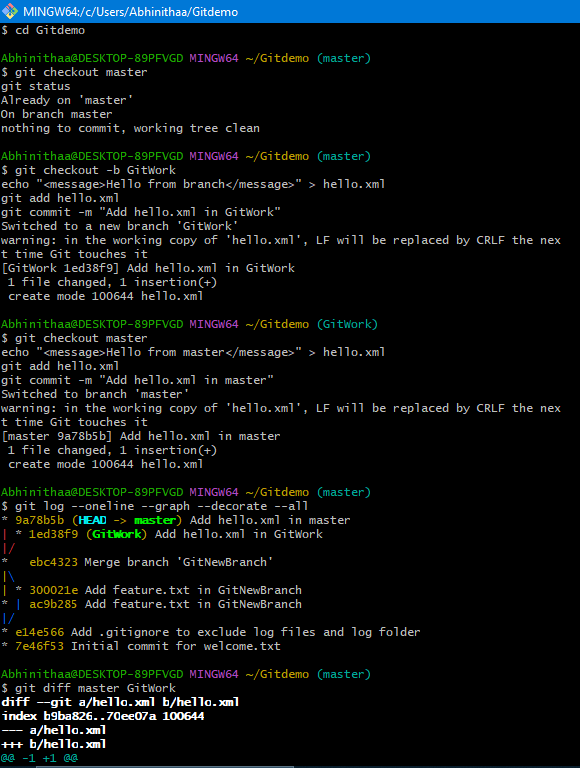
**Merging:**

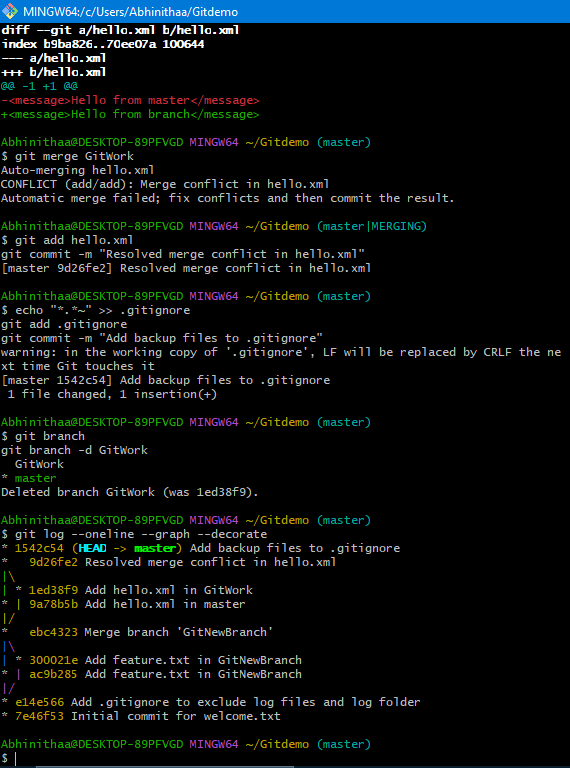
1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using **“git log –oneline –graph –decorate”**
6. Delete the branch after merging with the trunk and observe the git status.

****

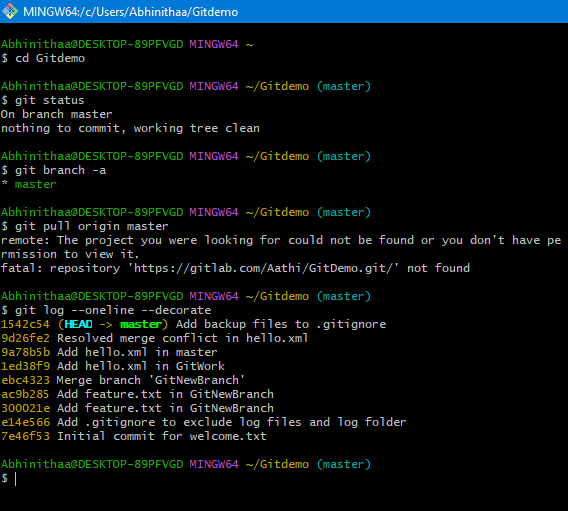
****

1. Please follow the instructions to complete the hands-on. Each instruction expect a command for the Git Bash.
2. Verify if master is in clean state.
3. Create a branch **“GitWork”.** Add a file “hello.xml”.
4. Update the content of “hello.xml” and observe the status
5. Commit the changes to reflect in the branch
6. Switch to master.
7. Add a file **“hello.xml”** to the master and add some different content than previous.
8. Commit the changes to the master
9. Observe the log by executing **“git log –oneline –graph –decorate –all”**
10. Check the differences with Git diff tool
11. For better visualization, use P4Merge tool to list out all the differences between master and branch
12. Merge the bran to the master
13. Observe the git mark up.
14. Use 3-way merge tool to resolve the conflict
15. Commit the changes to the master, once done with conflict
16. Observe the git status and add backup file to the .gitignore file.
17. Commit the changes to the .gitignore
18. List out all the available branches
19. Delete the branch, which merge to master.
20. Observe the log by executing **“git log –oneline –graph –decorate”**

****

****

1. Please follow the instructions to complete the hands-on. Each instruction expects a command for the Git Bash.
2. Verify if master is in clean state.
3. List out all the available branches.
4. Pull the remote git repository to the master
5. Push the changes, which are pending from **“Git-T03-HOL\_002”** to the remote repository.
6. Observe if the changes are reflected in the remote repository.

****