|  |
| --- |
| **Source Code Management with Git**  **Handson Workbook** |

1. **Table of Contents**
2. [**Activity1: Install Git and Initial configuration of Git 3**](#_Toc498517727)
3. [**Activity2: Working in GitHub 5**](#_Toc498517728)
4. [**Activity3: Using Git Basic Commands 16**](#_Toc498517729)
5. **Activity4:Sync local repo to remote repo………………………………………….21**
6. [**Activity 5: Using Git Advance Commands 27**](#_Toc498517731)
7. [**Activity 6: Using Git Advance Commands………………………………………..27**](#_Toc498517731)
8. **Activity 7: Working with git revert…………………………………………………..33**

# **Activity 1**

# Install Git and Initial configuration of Git

## Scenario

|  |
| --- |
| To get started hands on for Git, we need to install GIT based on OS we use. |

## Steps

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | **Installation Commands:**  Following table lists the commands to install Git in different operating systems:   |  |  | | --- | --- | | **Operating System** | **Command to Install** | | Linux | Run the following command: | | Debian-based like Ubuntu | Run the following command: | | Windows | Access the following link:  [**Git - Downloading Package (git-scm.com)**](https://git-scm.com/download/win)The download will start automatically.  PFA doc. For git installing git instructions. | |
| 2 | **Initial Configuration Setup:**  To set an identity, use the following commands:    Example: |
| 3 | **To view all the configurations, use the following command:** |
| 4 | **Initializing Git repository:**  To initialize Git repository run the following commands:  (to initialize git repository)  (go to git directory and verifying GIT repository initialization)  (to list all files and directories .git contains) |

# **Activity2**

# Working in GitHub

## Scenario

|  |
| --- |
| Create an account in GitHub. Login to the GitHub and create a new repository as NewProject and branch gitbash. Create and add the file to the project. Make some changes in the code and then commit your changes in the main as well as develop. |

## Steps

|  |  |
| --- | --- |
| 1 | **Creating a GitHub Account:**   1. Login to <https://github.com/signup> in new browser. 2. Create new account by giving username, email address and password and login. |
| 2 | **Creating Github Repository:**  1. Click **New repository** to create a repository.  2. Provide name of new Repository Example: **demoproject.**  3. Select for **Public** button.  4. Check “**Initialize this repository with a README**”  5. Click **Create repository**.    Please review your repository created as below: |
| 3 | **Adding a new file**   1. Click **Create new file**.      1. Name the file. For this exercise “samplecode.sh”      1. Add lines of code.   For example:  echo “welcome to the demo project”     1. Click on **Commit Changes…** 2. Add Commit message as shown below.      1. Ensure that **create a new branch for this commit and start a pull request** and provide branch name as **develop** as shown above. 2. Click on **Propose Changes** 3. Now add comments and click on **create pull request**     9.Now click on **merge pull request and confirm merge.**    10.Pull request is successfully merged and closed and verify your newly created file in main branch. |
| 5 | **Editing an existing file**  1.Go to develop branch in branches tab    2.click on the **samplecode.sh** file    3.Edit the file content in the **Edit file** tab    Now modify the content of the file, here added a new line    4.Click on **Preview** to view the changes. The new content will be displayed in Green, click on **commit changes.**    5.Give a meaningful comment when committing which explains the changes made to the file.    6.Click on **commit Changes**    7.Navigate to main branch and **create pull request** to merge changes from develop branch into main branch  8.click on **compare & pull request** and confirm merge    8.cick on **create pull request**    9.click on **merge pull request** and **confirm merge**       1. Now pull request is successfully merged and closed, 2. Navigate to main branch and click on file to see the updated content. |
| ------- End of Exercise ------- | |

# **Activity 3**

# Using Git Basic Commands

## Scenario

|  |
| --- |
| Clone the repository that we created in GitHub into local machine. Add a file to the repository and commit the changes in the repository. |

## Steps

|  |  |
| --- | --- |
| 1 | **Clone the repository**  Execute the following command in order to clone the repository into local machine which we created in GitHub:        Now create a new branch and switch in to the new branch. |
| 2 | **Adding file to a repository**  Create a new text file in GIT initialized folder and save the changes to the staging using the command, “” as shown in the following screen capture. |
| 3 | **Committing the changes to the Repository**  To commit the changes to the GIT repository, execute the following command:    Example:    The changes committed to the repository is shown in the following screen capture: |
| 4 | **Using GIT Push Command**  Run the following command to push commit on to the repository:      Go to git hub and navigate to **develop branch** ,create a new pull request to merge changes in to main branch from develop branch    Click on **compare & pull request** and **create pull request**, **merge pull request** and **confirm merge**      After **confirm merge** go to main branch and verify the newly created file test.txt added into main branch |
| 5 | **Tagging a repository** |
| 6 | **Viewing the log details of the repository**  To log into the GIT repository, execute the following command and see the result: |
| ------- End of Exercise ------- | |

**Activity 4**

# Sync local repo to remote repo

## Scenario

Alen wants to create repository locally and sync it with the remote repository

|  |  |
| --- | --- |
| 1 | Go to git bash and create repository Ex: newproject |
| 2 | Now initialize git into the local repo by using following command |
| 3 | Create a new file Ex:file.txt as shown below and add content into the file and save it, by using vi editor |
| 4 | Now add the newly created file into staging area and commit the changes by using following commands |
| 5 | Go to Github and create a new repository Ex: new-project |
| 6 | Now click on **<>code** copy the remote repository HTTPS clone url as shown below |
| 7 | Go to git bash and execute the following command |
| 9 | Now push the changes to remote repo by using below command |
| 10 | Go to git hub and click on **master branch** to see the newly created file. |

**Activity 5**

# Using Git Advance Commands

## Scenario

|  |
| --- |
| **Task 1**: Create a branch devTest, pull the changes from the develop branch over here and then make some more changes, save those changes and then commit your changes. Go to develop branch and then merge the changes of devTest branch in develop branch.  **Task 2:** Add a new file in the working tree and make the changes and commit. Post this copy this file to another file File2 and then rename the newly added file. Finally remove the File2 from the working tree. Check status of the working tree. |

## Steps

|  |  |
| --- | --- |
| 1 | **Scenario Task**   * Create a branch devTest, pull the changes from the develop branch over here and then make some more changes, save those changes and then commit your changes. * Go to develop branch and then merge the changes of devTest branch in develop branch.   Following steps and screen captures should help you to accomplish the considered task:   1. Run the command in gitbash “” for creating new branch. 2. Use command “” to get the latest code from master in this new devTest branch. 3. Make changes to any of the file for example: README.md file. 4. Save and commit the changes. 5. Checkout develop branch using command “”. 6. Merge the changes of devTest branch to develop branch using command “”. |
| 2 | **Scenario Task**   * Add a new file in the working tree and make the changes and commit. * Post this copy this file to another file File2 and then rename the newly added file. * Finally remove the File2 from the working tree. Check status of the working tree. * Also now if you don’t want to commit these changes just stash/save your changes and also give them tags. * Use log for logging.   Following steps should help you to accomplish the considered task:   1. To create new file use command 2. Make changes to the file and save it.      1. Run the following command:          1. Use command  to copy files. 2. Rename test3 as newtest by executing the following command:      1. Remove test2 |

# 

# **Activity 6**

# Using Git Advance Commands

## Scenario

|  |
| --- |
| Following are the key scenarios where you need to use some of the other Git Advance Commands:   * Creating an alias for most frequently used command based on the history of commit. * Identifying the file status by displaying which stage is a file in Git * Keeping the changes in local branch that you want to commit later. * Removing untracked files and directories |

## Steps

|  |  |
| --- | --- |
| 1 | **Scenario Task - Creating an alias for most frequently used command based on the history of commit.**  Alan wants to create an alias for most frequently used command based on the history of commit. He creates an alias by using the following command to save his writing time.  Command:   * Example:   After creating alias, the command can be as follows. |
| 2 | **Scenario Task - Identifying the file status by displaying which stage is a file in Git**  Alan has his local files with similar names. He has forgotten what was the last action taken on those files. He wants to display which stage is a file in GIT.  Command:    “**M**” indicates file is modified and the file needs to staged for commit. |
| 3 | **Scenario Task - Keeping the changes in local branch that you want to commit later**  Alan wants to keep the changes in local branch and do not want to commit right now, and may wish to commit later.  Stash : Temporarily save/stash the changes in current branch  Command : |
| 4 | **Scenario Task - Merging without using merge command**  Alan wants to merge the repository without using merge command.  Pull upstream changes with rebase instead of merge. |
| 5 | **Scenario Task - Removing untracked files and directories**  Alan wants to remove untracked files and directories.  (remove untracked files)  (remove untracked files/directories)  (list all which will be removed) |
| 6 | **Scenario task -Working with git Cherry-pick**  Alan wants to pick changes from specific commit in **feature** branch in to his working branch **main.**  ->Create new repository in GITHUB  ->Go to git bash and clone the newly created repository    ->now edit the README file and add some content by using Vi editor. Now follow add & commit steps as shown below.    ->Now Create a new branch named feature and chekout into the branch    Edit the README file and add some content by using Vi editor. Now follow add & commit steps as shown below    Again edit the README file and add some content by using Vi editor. Now follow add & commit steps as shown below    Copy the commit id as shown below    Now switch to main branch    Now use cherrypick command as shown below to pick changes from specific commit from feature branch in to main branch.    Now push your changes to remote repository. |
| ------- End of Exercise ------- | |

# **Activity 7**

# Working with git revert

## steps

|  |
| --- |
| **Scenario task -Working with git revert**  1.Alan wants to revert back file/s without adding to staging area  ->Be in main branch  ->Add a new file ex: **emplist.txt** with some content in local repository  ->Issue command ""  ->Go to local repository and verify the file removed  ->give and verify  A computer screen shot of text  Description automatically generated  2.Alan wants to revert back file/s after adding to staging area  Be in main branch  Go to local repo and create a new file file2.txt with some content.  issue "  give            3.Alan wants to revert back file/s after commiting  Go to local repo  Create a new file file.txt with some contents  go to git bash and give    (To revert commit)  ( To remove from staging area)  ( To clear from working area)      4.Alan wants to revert back file/s after push  Go to local repo and add a new file file.txt  Issue following commands        (Note down commit id of the last push)          A black background with numbers and letters  Description automatically generated    go to remote and refresh and check  A screenshot of a computer  Description automatically generated |
|  |

Copyright © 2024 Accenture

All rights reserved.

Accenture and its logo are trademarks of Accenture.