**Lesson 03 Demo 04**

**Stashing in Git**

**Objective:** To use Git stashing and temporarily save uncommitted changes, allowing switch between branches without losing progress

**Tools required:** Git and Ubuntu

**Prerequisites:** None

Steps to be followed:

1. Set up the environment
2. Initialize Git and make initial commits
3. Manage branch transitions and changes
4. Stash changes and verify

**Step 1:** **Set up the environment**

1. Open the Ubuntu terminal from your practice lab and create a directory using the following command:  
   **mkdir stashdemo**  
     
   
2. Run the following command to navigate to **stashdemo** :  
   **cd stashdemo/**  
   
3. Configure Git using the following commands:  
   **git config –global user.email “user\_email\_ID”**

**git config –global user.name “user\_name”**

   
  
 **Note:** Replace **user\_email\_ID** with your Git email ID and **user\_name** with   
 your GitHub username

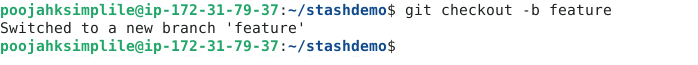
**Step 2:** **Initialize Git and make initial commits**

* 1. Initialize the Git using the following command:  
     **git init**  
     
  2. Create a file using the following command:  
     **echo “ ”>index.txt**



* 1. Run the following command to stage the changes in the Git repository:  
     **git add .**  
       
     A number and text on a white background

     Description automatically generated with medium confidence
  2. Commit the staged changes using the following command:  
     **git commit -m “created index.txt”**   
       
     A close up of a computer code

     Description automatically generated
  3. Run the following command to create and switch to the **feature** branch:  
     **git checkout -b feature**  
       
     

**Step 3:** **Manage branch transitions and changes**

3.1 Run the following command to list the files:

**ls**  
  
A number with numbers and symbols

Description automatically generated with medium confidence

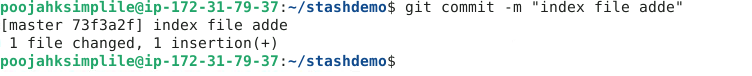
* 1. Run the following command to create a new file:  
     **vi feature.txt**   
       
     
  2. Enter the desired data inside the **feature.txt** file as shown in the screenshot below:  
       
     A screenshot of a computer

     Description automatically generated
  3. Run the following command to display the current state (changed, staged, and untracked) of the working directory files:  
     **git status**  
       
     A screenshot of a computer program

     Description automatically generated

* 1. Run the following command to stage the changes in the Git repository:  
     **git add .**  
       
     
  2. Commit the staged changes using the following command:  
     **git commit -m “feature file WIP”**  
       
     A close-up of a sign

     Description automatically generated
  3. Switch to the master branch using the following command:  
     **git checkout master**  
       
     
  4. Navigate inside the index.txt using the following command:  
     **vi index.txt**  
       
     
  5. Add the desired data inside the index.txt file as shown in the screenshot below:  
       
     A screenshot of a computer

     Description automatically generated
  6. Run the following command to stage the changes in the Git repository:  
     **git add .**  
       
     
  7. Commit the staged changes using the following command:  
     **git commit -m “index file add ”**   
       
     
  8. Switch to the feature branch using the following command:  
     **git checkout feature**  
       
     
  9. Run the following command to display the commit history of the current branch showing a list of commits with details:  
     **git log**  
       
     A screen shot of a computer code

     Description automatically generated  
       
     In the above output, observe that the commit on the feature.txt and index.txt files are successful.
  10. Open index.txt from the **feature** branch (current branch) using the following command:  
      **vi index.txt**  
        
      
  11. Enter the desired data inside the **index.txt** file as shown in the screenshot below:  
        
      A screenshot of a computer

      Description automatically generated
  12. Run the following command to display the current state of the working directory files:  
      **git status**  
        
      A screenshot of a computer program

      Description automatically generated  
        
      In the above screenshot, observe that the **index.txt** file has been modified from the **feature** branch.

**Step 4:** **Stash changes and verify**

1. Run the following command to switch to the master branch:  
   **git checkout master**  
     
   A close up of a white background

   Description automatically generated

|  |
| --- |
| **Note**: The error message indicates that switching to the master branch would overwrite local changes made to index.txt. Thus, you must either commit or stash these changes before changing branches. |

1. Run the following command to temporarily save changes:  
   **git stash**  
     
   
2. Run the following command to list all stashed changes:  
   **git stash list**  
   

By following these steps, you have successfully performed stashing in Git to save uncommitted changes temporarily.