

**DevOps Tools**

Table of Contents

[1](#_Toc532207737)

[Table of Contents 2](#_Toc532207738)

[1 Lab Details: 3](#_Toc532207739)

[2 DevOps Tools: Guided Exercises 3](#_Toc532207740)

[2.1.1 GIT – Problem Statement 3](#_Toc532207741)

[Guided Exercise 1: 4](#_Toc532207742)

[Guided Exercise 2: 6](#_Toc532207743)

[Guided Exercise 3: 8](#_Toc532207744)

[Guided Exercise 4: 11](#_Toc532207745)

# Lab Details:

You will be provided with the below DevOps Lab for practicing the guided exercise in this document.

1. Enroll for DevOps Tools - Practice Lab to practice these guided exercises.

# DevOps Tools: Guided Exercises

### GIT – Problem Statement

A Leading Sport Promoters Company in India is planning to develop a web based fantasy sports platform that allows users to play fantasy cricket, football, kabaddi and basketball games with their statistical skills and game knowledge. They have introduced this platform as a mobile application initially and now planning to create a web application on the same.

There are lakhs of fans and players for this game and the company is planning to develop and deploy the project in DevOps environment. There are many developers working on this project and their daily builds shall be updated to the GIT repository. Since there are many developers it’s advised to work on branches to avoid conflicts.

**Scope:**

You have been assigned the task of managing the builds on the Git Repository for continuous integration of the project along with fellow developers and making sure that everyone works on branches to avoid merge conflicts.

Given the project build, you need to perform the below mentioned tasks.

* Create a local repository for staging and committing files.
* Maintain the log of commits performed.
* Merge the commits locally before performing sync with the remote repository.
* Perform branch and merge operations.
* Push your local repository update to GIT remote repository.

**Steps:**

1. Install, test and Configure GIT.

2. Initialize GIT local repository.

3. Create files in the Working directory (Local Repository).

4. Add files to staging area in the Repository.

5. List the files added in the repository.

6. Commit files to the repository.

7. Get information about the commit made to the repository.

8. Create new branch

9. Add all files and commit files to the repository

10. Merge the branches

11. View all the commits made in the repository.

12. Add data to remote repository.

13. Push data to remote repository.

Guided Exercise 1

**Estimated Completion Time:** 5 Minutes

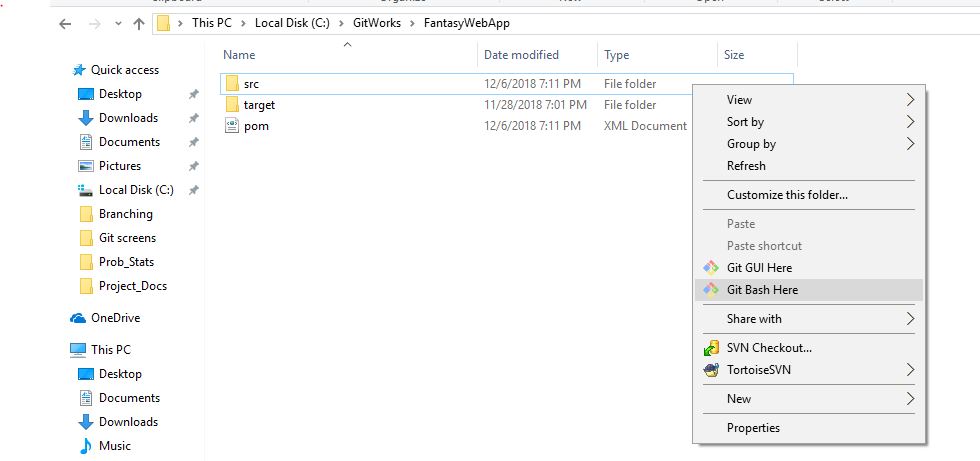
**Steps to follow:**

**Step 1:**

Copy the given FantasyWebApp.zip on your system, extract the zip file to any location

And open the FantasyWebApp folder and right click and select GIT Bash. You will see the screen as provided in Fig 1.1



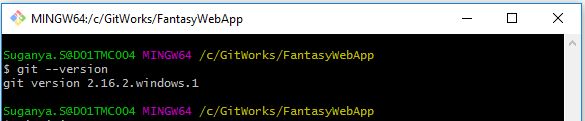


[ Fig 1.1]

**Step 2:** Test the environment using (Fig 1.2)

**Solution:**

**Command: git –version**



[Fig 1.2]

**Step 3:** Initializing GIT local repository and working with files in GIT environment

A. Check the current path:

**Solution:**

**Command: pwd, git init**

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo3.JPG

[Fig 1.3]

**Step 4:** Setting configuration

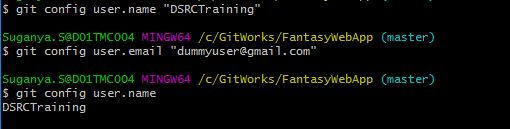
Set values for user name and email so that GIT can use this information to Identify

Users in the repository.

**Solution:**

**Command:**

1. **git config -- all**
2. **git config user.name “your-name”**
3. **git config user.email “your-email”**
4. **git config user.name**



[Fig 1.4]

Guided Exercise 2:Working with files in GIT Environment

**Estimated Completion Time:** 10 Minutes

**Objective:** To add files from working area to staging area and commit these files to the Repository.

**Steps to follow:**

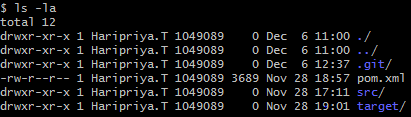
**Step 1:** Adding file’s from working directory to the Local Repository

Add Command To:

1. Navigate to the location where project is extracted. List the contents of the directory.

**Solution:**

**Command: ls –la** . Refer Figure 2.1

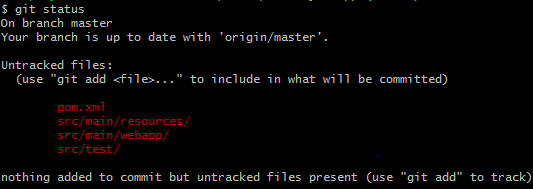


[Fig 2.1]

1. Get the status of the files

**Solution:**

**Command: git status** . Refer Figure 2.2



[Fig 2.2]

1. To add the java source files to the staging area, change the location to the folder where java files are existing and verify the current location. Refer Figure 2.3 and 2.4

**Solution:**

**Command: cd , pwd**

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo5.JPG

[Fig 2.3]

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo6.JPG

[Fig 2.4]

1. Create a new file under specific path .Refer Figure 2.4

**Solution:**

**Command: touch new-file-name**

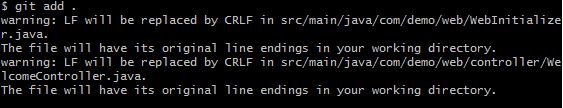
C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo7.JPG

[Fig 2.5]

1. Add all the files from src\main\java\com\demo\web folder to staging area (index area of GIT. Refer Figure 2.6.

**Solution:**

**Command: git add .**

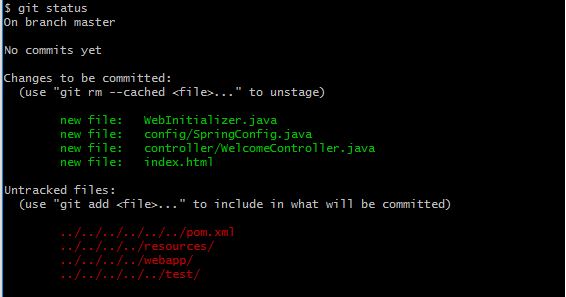


[Fig 2.6]

1. After adding all the files as per illustrated in Fig 2.6, the status will appear as shown in Fig 2.7. Verify the status.

**Solution:**

**Command: git status**



[Fig 2.7]

Guided Exercise 3:Viewing Repository History

**Estimated Completion Time:** 10 Minutes

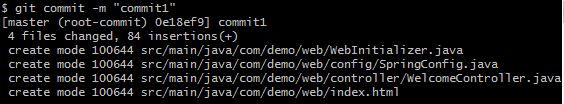
**Objective:** To View commit History from the Repository

Steps to follow:

**Step 1:** The changes made to the source files are committed to the staging area.

**Solution:**

**Command: git commit –m “message”.** Refer Figure 3.1

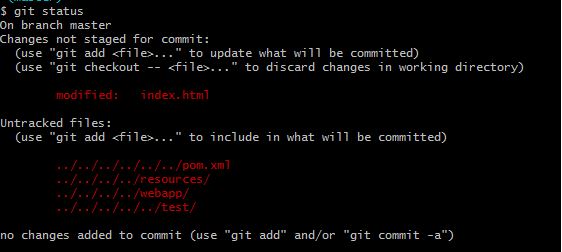


[Fig 3.1]

**Step 2:** After commiting the changes in the source file verify the status.

**Solution:**

**Command : git status**  . Refer Fig 3.2



[Fig 3.2]

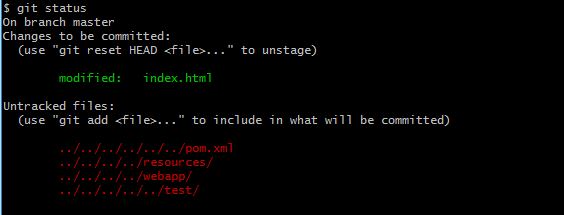
**Step 3:** Add all the files to the staging area to view the modified file and verify the status again. Refer Fig 3.3 & Fig 3.4

**Solution:**

**Command : git add .**

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo12.JPG

[Fig 3.3]



[Fig 3.4]

Initial branch name will be master as shown in Figure 3.5

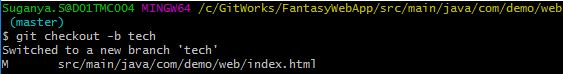
C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo14.JPG

[Fig 3.5]

**Step 3:** Create a new branch using the command.

**Solution :**

**Command :**  **git checkout –b new-branch-name**. Refer Fig 3.6

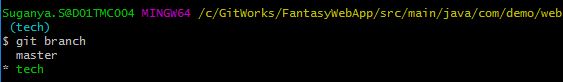


[Fig 3.6]

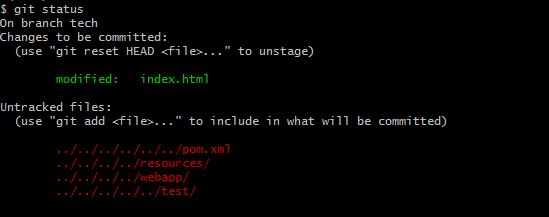
Now the branch is switched to a new branch named “tech” as mentioned above. View the changes and verify the status.

**Solution:**

**Command : git branch**



[Fig 3.7]



[Fig 3.8]

**Step 4:** Add all the files to a new branch .

**Solution:**

**Command : git add .**

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo18.JPG

[Fig 3.9]

**Step 5:** Commit the changes made to the source files to the staging area

**Solution:**

**Command: git commit –m “message”.** Refer Figure 3.10

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo19.JPG

[Fig 3.10]

**Step 6:** Perform clone operation in git to switch the branch to master. Refer 3.11

**Solution:**

**Command : git checkout master**

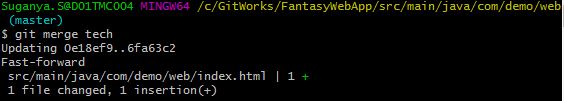
C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo20.JPG

[Fig 3.11]

**Step 7:** Use command to merge the branches to add all the modified and inserted files to the repository. Refer Fig 3.12.

**Solution:**

**Command : git merge <new-branch-name>**

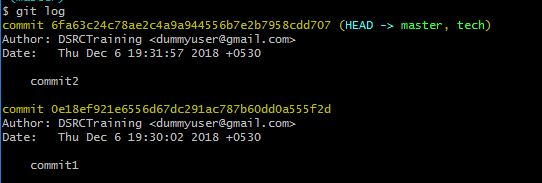


[Fig 3.12]

**Step 9:** View all the commits that are made in the repository. Refer Fig 3.13.

**Solution:**

**Command :git log**



[Fig 3.13]

**Step 10:**View the commit information for each commit from the repository

**Solution:**

**Command: git log --oneline**

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo23.JPG

[Fig 3.14]

**Step 11:** Use command to view the file information for the specific commit using SHA1

- hash value. Result shown in the Fig 3.15

**Solution:**

**Command: git show –pretty=”” –name-only <hash-value>**

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo24.JPG

[Fig 3.15]

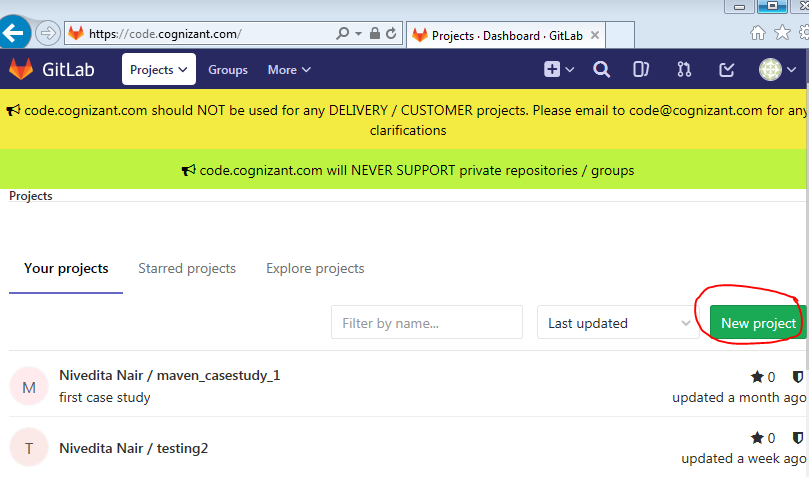
Guided Exercise 4:Adding Data to remote Repository

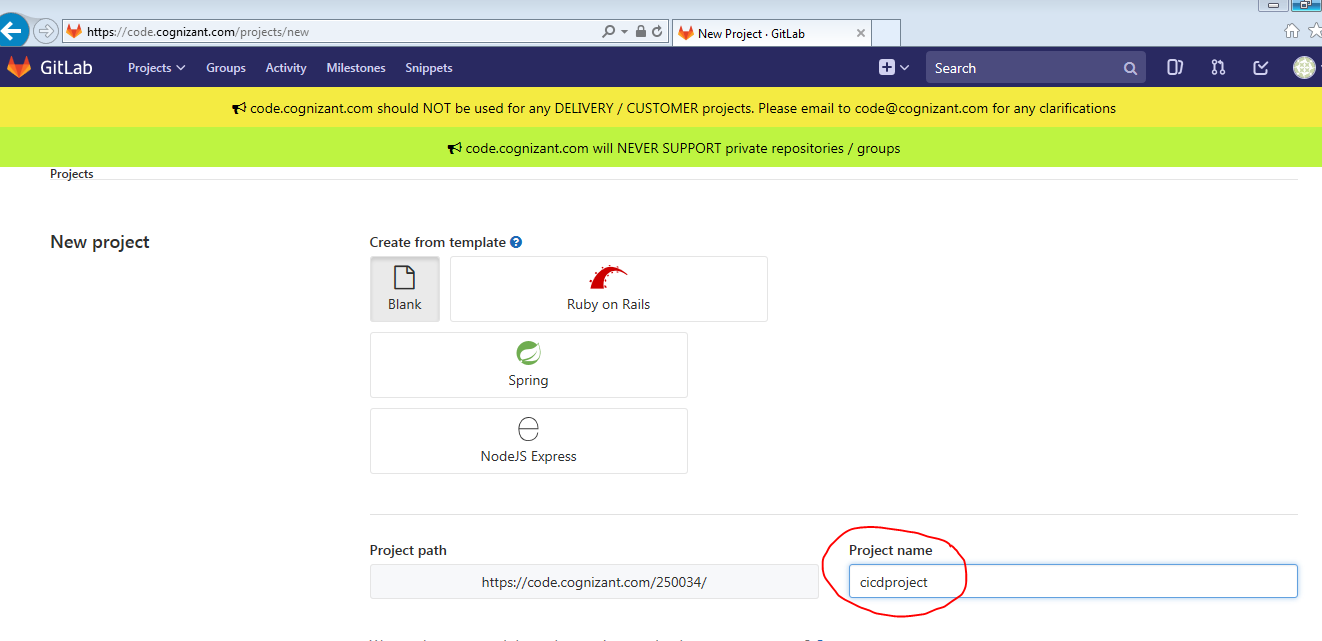
**Estimated Completion Time:** 5 Minutes

**Objective:** Synchronizing data with remote repository

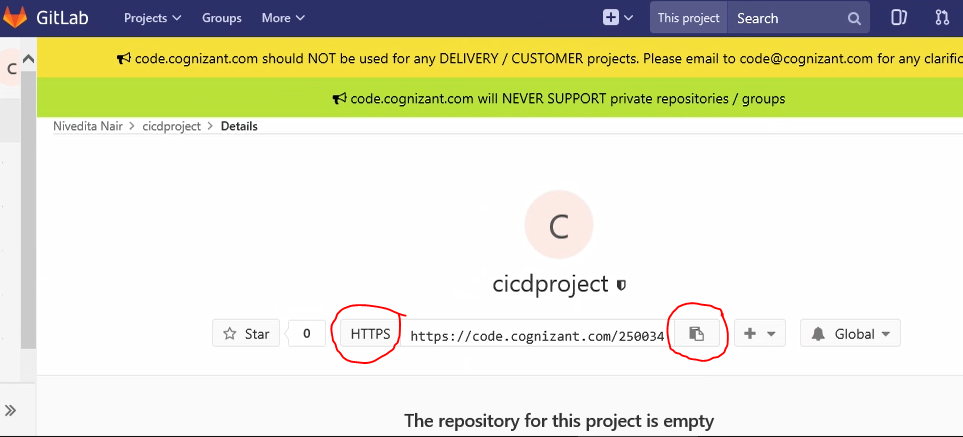
Steps to follow:

**Step 1:**

* Log into GitLab with your Cognizant credentials to configure your repository <https://code.cognizant.com>
* Create a project named **cicd project **

****

[Fig 1.5]

**Step 2:** Copy the GitLab project URL.  


<https://github.com/DSRCTraining/FantasyWebApp.git> (Sample URL)

**Step 3:** Copy the existing Git Repository to a local file system.

Clone the repo in a new directory, at another location. Refer Fig 4.4

**Solution:**

**Command; git clone <<repo Name>>**

**C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo25.JPG**

[Fig 4.4]

**Step 4:** Specify the remote repository location to which the files needs to be submitted

**Solution:**

**Command: git remote add origin <<repo Name>>**

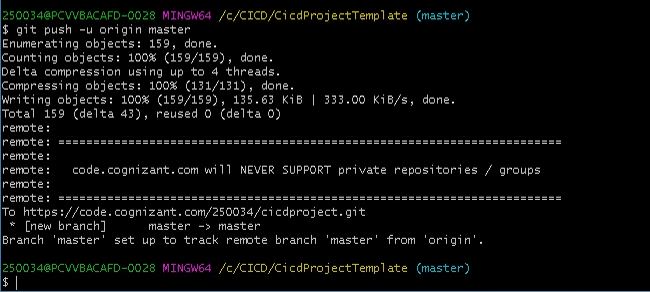
C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo26.JPG

[Fig 4.5]

**Step 6**: Push the files to the remote repository.

**Solution:**

**Command: git push –u origin master**



[Fig 4.6]

**Step 7:** Delete the created branch. Refer Fig 4.7

**Solution:**

**Command: git branch –d tech**.

C:\Users\haripriya.t\Documents\IPMsg\AutoSave\demo29.JPG

[Fig 4.7]

**Summary:**

You have learnt to install and configure GIT for DevOps Environment