

# Introduction to Git

Lesson 01 : Getting  
Started with Git



# Lesson Objective

➤ In this lesson, you will learn:

- Introduction to Git
- Creating Git repository
- Understanding local and remote repository
- Committing changes to the repository
- Branching the repository
- Merging the changes back to the primary repository.
- Creating Azure DevOps account
- Creating & Collaborating with Azure DevOps repository





# Introduction

- What is Source Control?
  - Source control helps team manage changes to source code over time.
  
- What is Version Control?
  - The term version control is used interchangeably with source control.
  
- Difference between source control and version control:
  - Source control is specific to source code, where as version control also covers large binary files and digital assets.



# Introduction

## ➤ What is source control management?

- Source control management (SCM) refers to tools that help software teams manage changes to source code over time.
- Version control software keeps track of every modification to the code in a special kind of database.
- Software teams that do not use any form of version control often run into problems like not knowing which changes that have been made are available to users or the creation of incompatible changes between two unrelated pieces of work that must then be painstakingly untangled and reworked.



# What is Git?

- By far, the most widely used modern version control system in the world today is Git.
- Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel.
- Unlike CVS or SVN (Subversion), Git is a DVCS (Distributed Version Control System).
- In addition to being distributed, Git has been designed with performance, security and flexibility in mind.





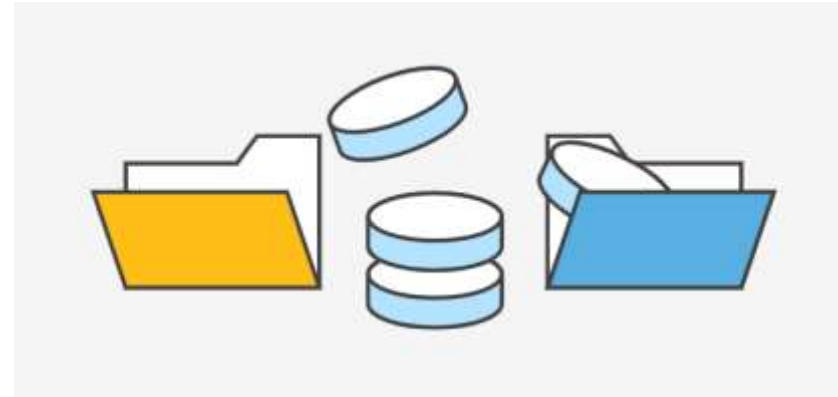
# Benefits of Git

- Complete long term change history of every file is maintained.
- Multiple independent streams of work can co-exist with branches.
- Distributed Version Control System (DVCS).
- Performance of Git is very strong when compared to many alternatives.
- Git repository are secured with a cryptographically secure hashing algorithm called SHA1.
- Open source



# Installing Git

- Installing Git on Windows:
  - Download the latest Git from <https://gitforwindows.org/>
  - When you've successfully started the installer, you should see the **Git Setup** wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users.





# Basic Commands

- Open a Command Prompt (or Git Bash if during installation you elected not to use Git from the Windows Command Prompt).

- Checking the installed version:

```
$ git --version  
git version 2.27.0.windows.1
```

- Setting username and email:

```
$ git config --global user.name "Manish Rajhans"
```

```
$ git config --global user.email "manish.rajhans@capgemini.com"
```

- Verifying username and email:

```
$ git config --global user.name  
Manish Rajhans
```

```
$ git config --global user.email  
manish.rajhans@capgemini.com
```





# Basic Commands (Cont....)

- **git and get help:** Both commands are same.

```
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone                Clone a repository into a new directory
  init                 Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add                  Add file contents to the index
  mv                   Move or rename a file, a directory, or a symlink
  restore              Restore working tree files
  rm                   Remove files from the working tree and from the index
  sparse-checkout      Initialize and modify the sparse-checkout
```

- **get help init:** Offers browser based help.



# Creating new local git repository

➤ Step1. Open up your project folder, and navigate into the same.

- For checking your current location:

```
$ pwd  
/c/Users/mrajhans
```

- Identify location of repository and navigate:

```
$ cd c:/study/GitRepos
```

- Create new directory to make it as repository and navigate to it.

```
$ mkdir HelloWorld
```

```
$ cd HelloWorld/
```

- Initialize directory as git repository:

```
$ git init  
Initialized empty Git repository in C:/Study/GitRepos/HelloWorld/.git/
```

- After repository is created, then observe that now prompt is showing it as master, as shown below:

```
mrajhans@LIN20001179 MINGW64 /c/study/GitRepos/HelloWorld (master)  
$ |
```



# .git folder in repository

- The . git folder contains all the information that is necessary for your project in version control and all the information about commits, remote repository address, etc. All of them are present in this folder. It also contains a log that stores your commit history so that you can roll back to history.
- Viewing .git directory:
  - Change the directory to .git:

```
mrjans@LIN20001179 MINGW64 /c/study/GitRepos/HelloWorld (master)
$ cd .git/

mrjans@LIN20001179 MINGW64 /c/study/GitRepos/HelloWorld/.git (GIT_DIR!)
$ |
```



# .git folder in repository (cont....)

- .git folder contains the following information. You can view the available file and folders with following command.

```
mrjghans@LIN20001179 MINGW64 /c/study/GitRepos/HelloWorld/.git (GIT_DIR!)
$ ls -al
total 11
drwxr-xr-x 1 mrjghans 1049089  0 Jun 29 17:04 ./
drwxr-xr-x 1 mrjghans 1049089  0 Jun 29 17:12 ../
-rw-r--r-- 1 mrjghans 1049089 130 Jun 29 17:04 config
-rw-r--r-- 1 mrjghans 1049089  73 Jun 29 17:04 description
-rw-r--r-- 1 mrjghans 1049089  23 Jun 29 17:04 HEAD
drwxr-xr-x 1 mrjghans 1049089  0 Jun 29 17:04 hooks/
drwxr-xr-x 1 mrjghans 1049089  0 Jun 29 17:04 info/
drwxr-xr-x 1 mrjghans 1049089  0 Jun 29 17:04 objects/
drwxr-xr-x 1 mrjghans 1049089  0 Jun 29 17:04 refs/
```



# Adding new file to git repository

- Create file in the current folder / repo:

```
$ nano helloworld.js
```

- This will open the file in edit mode, as shown below. You can write anything into it.



```
MINGW64:/c:/study/GitRepos/HelloWorld
GNU nano 4.9.3 helloworld.js
console.log("HelloWorld!!!")
```

- Press Ctrl+X to close and press Y to save changes.



# Git status

- To check status of repository, type command, git status:

```
$ git status
On branch master

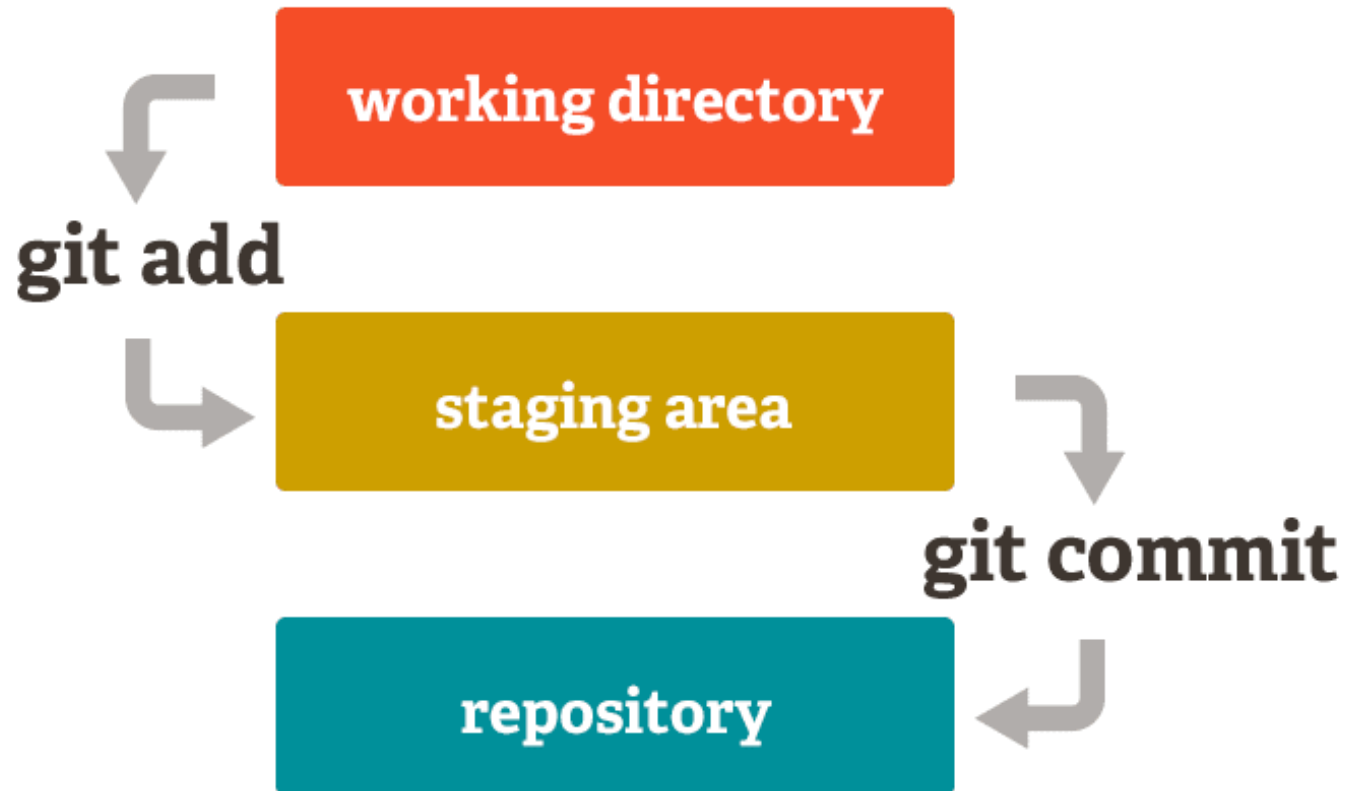
No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
```



# Working directory, staging area and repository





# Adding file into staging area

- To add file from working directory to the staging area we can use git add command.

```
mrajhans@LIN20001179 MINGW64 /c/study/GitRepos/HelloWorld (master)
$ git add helloworld.js
warning: LF will be replaced by CRLF in helloworld.js.
The file will have its original line endings in your working directory
```

- Checking status after adding file in staging area:

```
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   helloworld.js
```





# Removing file from staging area

- To remove file from staging area and put it back into the working directory, use git reset command.

```
$ git status
on branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
```

- To add the file again in staging area, use git add command.

# Committing changes to the local repository



- When commit command is used all the tracked changes will be committed/ reflected to the repository.
- To commit the files from staging area to local repository, use git commit command.

```
mrjghans@LIN20001179 MINGW64 /c/Study/GitRepos/HelloWorld (master)
$ git commit -m "First Commit"
[master (root-commit) 83edd88] First Commit
1 file changed, 1 insertion(+)
create mode 100644 helloworld.js
```

- Status after commit:

```
mrjghans@LIN20001179 MINGW64 /c/Study/GitRepos/HelloWorld (master)
$ git status
On branch master
nothing to commit, working tree clean
```



# Creating repository in the existing working directory

- You can download the sample repository from shared material.
- Copy the folder and navigate to it.

```
mrajhans@LIN20001179 MINGW64 ~  
$ cd C:/Study/GitRepos/SampleRepository
```

- Now, initialize the directory as a git repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository  
$ git init  
Initialized empty Git repository in C:/Study/GitRepos/SampleRepository/.git/
```

- add all the existing files from the working directory into staging area with git add command.



# Creating repository in the existing working directory (Cont....)

- Now, check the status of created repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    404.html
    apple-touch-icon.png
    browserconfig.xml
    css/
    favicon.ico
    humans.txt
    index.html
    js/
    tile-wide.png
    tile.png

nothing added to commit but untracked files present (use "git add" to track)
```

- add all the existing files from the working directory into staging area with git add command.



# Creating repository in the existing working directory (Cont....)

- Now add all the existing files from the working directory into staging area with git add command.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git add .
warning: LF will be replaced by CRLF in 404.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in browserconfig.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/main.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/normalize.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in humans.txt.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/main.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/jquery-1.11.2.min.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/modernizr-2.8.3-responsd-1.4.2.min.js.
The file will have its original line endings in your working directory
```



# Creating repository in the existing working directory (Cont....)

- After checking status, it shows the status as files are added in staging area .

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   404.html
    new file:   apple-touch-icon.png
    new file:   browserconfig.xml
    new file:   css/main.css
    new file:   css/normalize.css
    new file:   css/normalize.min.css
    new file:   favicon.ico
    new file:   humans.txt
    new file:   index.html
    new file:   js/main.js
    new file:   js/vendor/jquery-1.11.2.min.js
    new file:   js/vendor/modernizr-2.8.3-respond-1.4.2.min.js
    new file:   tile-wide.png
    new file:   tile.png
```



# Creating repository in the existing working directory (Cont....)

- Once files are added in staging area, commit the files into local repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git commit -m "First Commit"
[master (root-commit) 1bd1941] First Commit
14 files changed, 992 insertions(+)
create mode 100644 404.html
create mode 100644 apple-touch-icon.png
create mode 100644 browserconfig.xml
create mode 100644 css/main.css
create mode 100644 css/normalize.css
create mode 100644 css/normalize.min.css
create mode 100644 favicon.ico
create mode 100644 humans.txt
create mode 100644 index.html
create mode 100644 js/main.js
create mode 100644 js/vendor/jquery-1.11.2.min.js
create mode 100644 js/vendor/modernizr-2.8.3-respond-1.4.2.min.js
create mode 100644 tile-wide.png
create mode 100644 tile.png
```



# Creating repository in the existing working directory (Cont....)

- Now all the files in working directory are in synch with files from working directory. Check the status of the same.

```
mrjans@LIN20001179 MINGW64 /c/Study/GitRepos/sampleRepository (master)
$ git status
On branch master
nothing to commit, working tree clean
```



# Modifying the files in the local repository

- Modify file index.html.

```
mrjrhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ nano index.html
```

- Check the status, now it shows the status of index file as modified.

```
mrjrhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```



# Modifying the files in the local repository (Cont....)

- Add the updated index file into staging area..

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git add .
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
```

- Now commit the files.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git commit -m "Updated index file commit"
[master d514e91] Updated index file commit
1 file changed, 1 insertion(+), 1 deletion(-)
```



# Creating remote repository on GitHub

- Navigate to Github.com and register on the site.

The screenshot shows the GitHub homepage in a web browser. The browser's address bar displays 'github.com'. The page features a dark background with the GitHub logo and navigation links at the top. A large white text area on the left reads 'Built for developers' followed by a description of GitHub as a development platform. On the right, a white sign-up form is highlighted with a red border. The form contains three input fields: 'Username', 'Email', and 'Password'. Below the 'Password' field, there is a note: 'Make sure it's at least 13 characters OR at least 8 characters including a number and a lowercase letter. [Learn more](#)'. At the bottom of the form is a green button labeled 'Sign up for GitHub'.

Username

Email

Password

Make sure it's at least 13 characters OR at least 8 characters including a number and a lowercase letter. [Learn more](#)

Sign up for GitHub



# Creating remote repository on GitHub (Cont....)

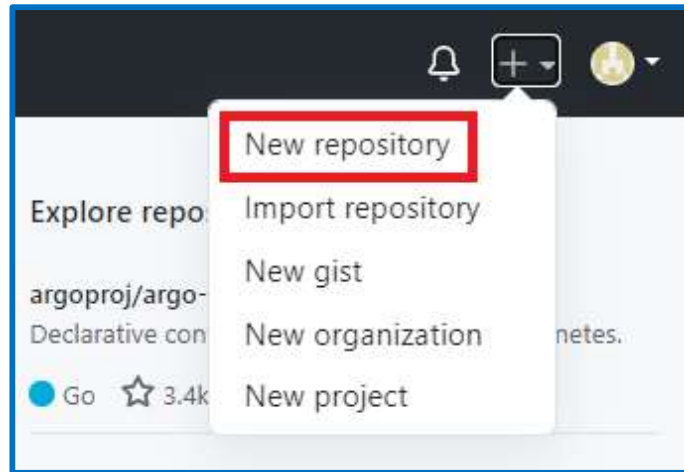
- After registration, you will receive an email to verify your email address. Click on Verify Button.





# Creating remote repository on GitHub (Cont....)

- After email verification, sign in to the GitHub and now click on create repository.





# Creating remote repository on GitHub (Cont....)

- After clicking on New repository, fill out the below form as below.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

manishrajhans ▾

Repository name \*

samplerespository ✓

Great repository names are short and snappy. [samplerespository is available.](#) Inspiration? How about [super-goggles?](#)

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: [None ▾](#)

Add a license: [None ▾](#) ⓘ

Create repository



# Creating remote repository on GitHub (Cont....)

- After clicking on Create repository, repository will get created and you will see below details. These details can be used to map local repository with this GitHub repository.

The screenshot shows the GitHub repository page for 'manishrajhans / samplerepository'. The page includes a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there are three main sections for repository setup:

- Quick setup — if you've done this kind of thing before:** This section includes a 'Set up on Desktop' button and a red box around the 'HTTPS' button. The URL 'https://github.com/manishrajhans/samplerepository.git' is displayed next to the buttons. Below this, it says 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and gitignore.'
- ...or create a new repository on the command line:** This section contains a code block with the following commands:

```
echo "e samplerepository" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/manishrajhans/samplerepository.git
git push -u origin master
```
- ...or push an existing repository from the command line:** This section is highlighted with a red box and contains a code block with the following commands:

```
git remote add origin https://github.com/manishrajhans/samplerepository.git
git push -u origin master
```
- ...or import code from another repository:** This section includes a note: 'You can initialize this repository with code from a Subversion, Mercurial, or TFS project.' and an 'Import code' button.

At the bottom of the page, there is a footer note: 'ProTip! Use the URL for this page when adding GitHub as a remote.'



# Mapping local repository with GitHub repository

- Use the command `git remote` as specified on the GitHub to map the local repository with the GitHub repository as below:

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)  
$ git remote add origin https://github.com/manishrajhans/samplerepository.git
```

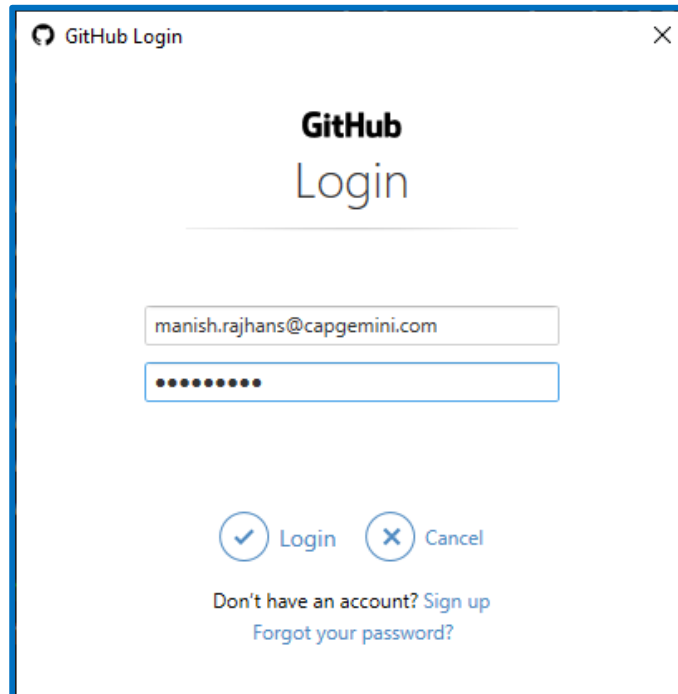


# Pushing the changes from local repository to GitHub

- Use the command git push as below:

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)  
$ git push -u origin master|
```

- For the first time push, it will ask you for username and password, provide them correctly.



A screenshot of the GitHub Login dialog box. The window has a title bar that says "GitHub Login" with a close button (X) in the top right corner. The main content area displays the "GitHub Login" text. Below this, there are two input fields: the first contains the email address "manish.rajhans@capgemini.com", and the second is a password field represented by a series of dots. At the bottom of the dialog, there are two buttons: "Login" (with a checkmark icon) and "Cancel" (with an X icon). Below these buttons, there are two links: "Don't have an account? Sign up" and "Forgot your password?".

# Pushing the changes from local repository to GitHub (Cont....)

- After this, all the changes will be written to GitHub.

```
mrjans@LIN20001179 MINGW64 /c/study/GitRepos/SampleRepository (master)
$ git push -u origin master
error: waitpid for git credential-manager get failed: No child processes
error: waitpid for git credential-manager store failed: No child processes
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 4 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (22/22), 57.98 KiB | 1.29 MiB/s, done.
Total 22 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
error: waitpid for send-pack failed: No child processes
To https://github.com/manishrajans/samplerepository.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

# Pushing the changes from local repository to GitHub (Cont....)

- Now refresh the GitHub page, and you will see all the files from local repository are now pushed into GitHub repository.

The screenshot displays the GitHub interface for a repository named 'manishrajhans / samplerepository'. The repository is currently on the 'master' branch. The file list shows the following files and their commit history:

File	Commit	Time
css	First Commit	17 hours ago
js	First Commit	17 hours ago
404.html	First Commit	17 hours ago
apple-touch-icon.png	First Commit	17 hours ago
browserconfig.xml	First Commit	17 hours ago
favicon.ico	First Commit	17 hours ago
humanis.txt	First Commit	17 hours ago
index.html	Updated index file commit	17 hours ago
tile-wide.png	First Commit	17 hours ago
tile.png	First Commit	17 hours ago

The right sidebar contains sections for 'About' (No description, website, or topics provided), 'Releases' (No releases published), 'Packages' (No packages published), and 'Languages' (HTML 52.5%, CSS 47.5%).

# Pushing the changes from local repository to GitHub (Cont....)

- If you check the status now, it shows that your branch is up-to-date.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```



# Modifying files on local repository

- If you check the status now, it shows that your branch is up-to-date.

```
mrjrhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ nano index.html

mrjrhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```



# Committing files on local repository

- Now you can add files into staging area.

```
mrjghans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git add .
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
```

- Committing changes to repository from staging area.

```
mrjghans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git commit -m "IndexFileChange01"
[master daa5e53] IndexFileChange01
1 file changed, 1 insertion(+), 1 deletion(-)
```

# Pushing committed files to GitHub repository

- This time, you can run the push command, and it will not ask you for credentials.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git push
error: waitpid for git credential-manager get failed: No child processes
error: waitpid for git credential-manager store failed: No child processes
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 304 bytes | 152.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
error: waitpid for send-pack failed: No child processes
To https://github.com/manishrajhans/samplerepository.git
   d514e91..daa5e53  master -> master
```

# Pushing committed files to GitHub repository (Cont....)

- You can observe the changes into the GitHub repository as well.

The screenshot displays the GitHub interface for a repository named 'manishrajhans / samplerepository'. The repository has 1 commit, 1 branch, and 0 tags. A recent commit by 'manishrajhans' is shown, committed 4 minutes ago. The commit message is 'IndexFileChange01'. The file 'index.html' is highlighted with a red box. The repository also contains other files like 'css', 'js', '404.html', 'apple-touch-icon.png', 'browserconfig.xml', 'favicon.ico', 'humans.txt', 'tile-wide.png', and 'tile.png'. The right sidebar shows the repository's 'About' section, 'Releases', 'Packages', and 'Languages' (HTML 52.4%, CSS 47.6%).

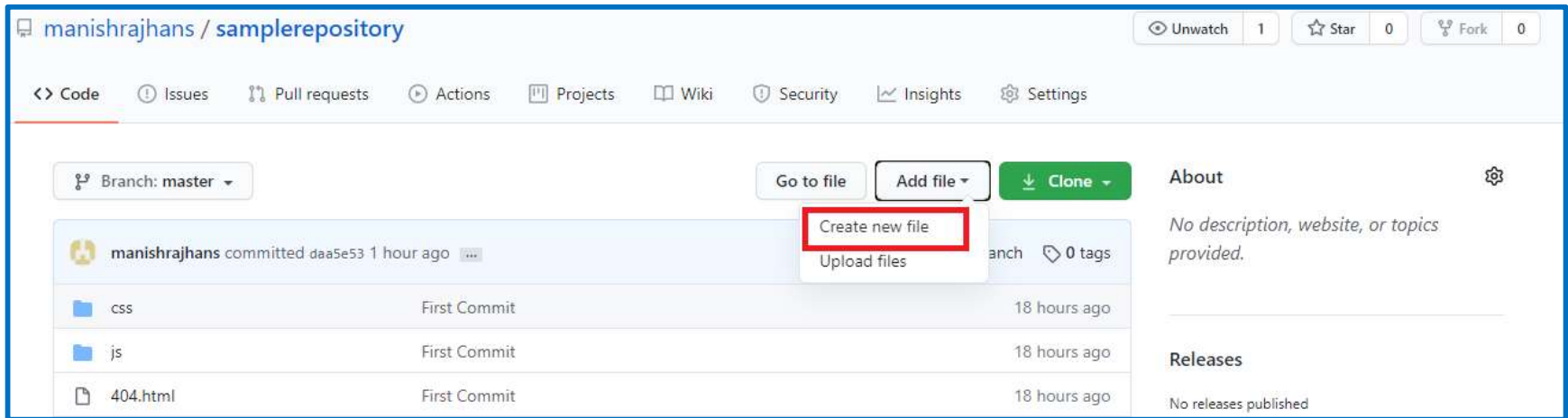
File	Commit	Time
css	First Commit	18 hours ago
js	First Commit	18 hours ago
404.html	First Commit	18 hours ago
apple-touch-icon.png	First Commit	18 hours ago
browserconfig.xml	First Commit	18 hours ago
favicon.ico	First Commit	18 hours ago
humans.txt	First Commit	18 hours ago
index.html	IndexFileChange01	4 minutes ago
tile-wide.png	First Commit	18 hours ago
tile.png	First Commit	18 hours ago



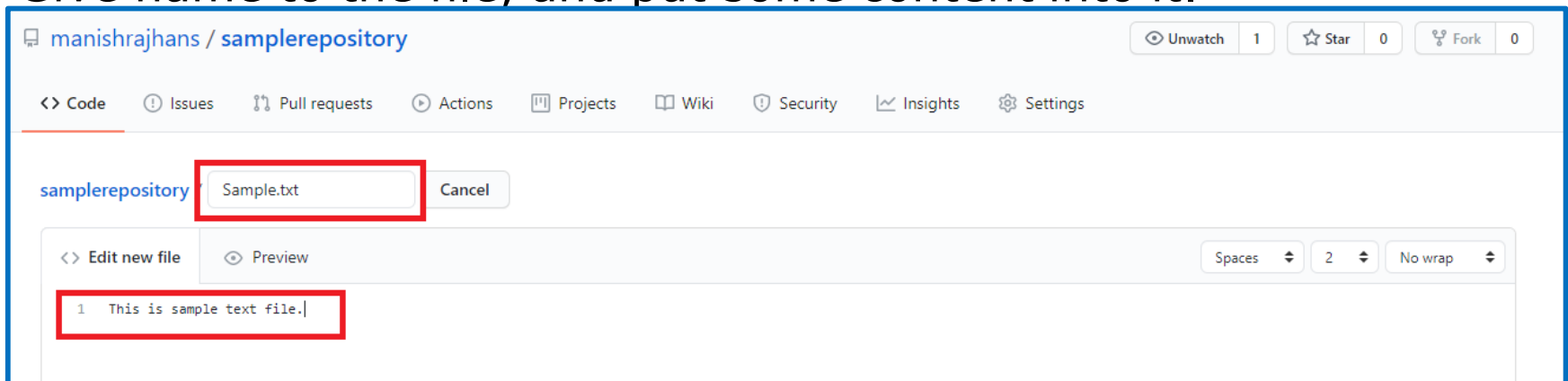


# Adding files into GitHub repository

- Click on Add File, then click on Create new file.



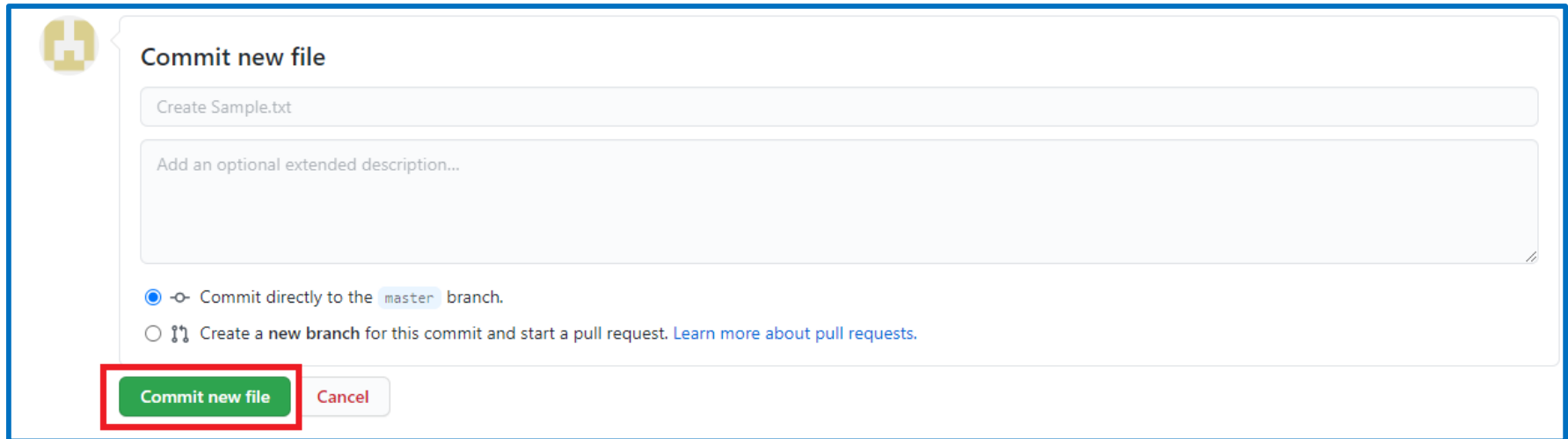
- Give name to the file, and put some content into it.



# Adding files into GitHub repository (Cont....)



- Commit the changes into master branch.

A screenshot of the GitHub 'Commit new file' dialog box. The dialog has a title bar with a GitHub logo and the text 'Commit new file'. Below the title bar, there is a text input field containing 'Create Sample.txt'. Below that is a larger text area with the placeholder text 'Add an optional extended description...'. At the bottom, there are two radio button options: the first is selected and labeled 'Commit directly to the master branch.', and the second is labeled 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' At the very bottom, there are two buttons: a green 'Commit new file' button and a grey 'Cancel' button. The 'Commit new file' button is highlighted with a red rectangular border.

**Commit new file**

Create Sample.txt

Add an optional extended description...

☒ Commit directly to the `master` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

**Commit new file** Cancel

# Adding files into GitHub repository (Cont....)



- Newly created file is now available on GitHub repository.

The screenshot shows the GitHub interface for the repository 'manishrajhans / samplerepository'. The 'Code' tab is selected. At the top, there are buttons for 'Unwatch', 'Star' (1), and 'Fork' (0). Below the repository name, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. A 'Branch: master' dropdown is visible. On the right, there are buttons for 'Go to file', 'Add file', and 'Clone'. The main content area shows a list of files and folders. The file 'Sample.txt' is highlighted with a red box, indicating it was just committed. The commit message for 'Sample.txt' is 'Create Sample.txt' and it was committed 'now'. Other files listed include 'css', 'js', '404.html', 'apple-touch-icon.png', and 'browserconfig.xml', all with a commit message of 'First Commit' and committed '19 hours ago'. On the right side, there are sections for 'About', 'Releases', and 'Packages', all showing 'No description, website, or topics provided.', 'No releases published', and 'No packages published' respectively.

File/Folder	Commit Message	Committed
css	First Commit	19 hours ago
js	First Commit	19 hours ago
404.html	First Commit	19 hours ago
Sample.txt	Create Sample.txt	now
apple-touch-icon.png	First Commit	19 hours ago
browserconfig.xml	First Commit	19 hours ago



# Getting changes from GitHub repository to local repository

- Changes in GitHub repository can be pulled in local repository with pull command.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git pull
From https://github.com/manishrajhans/samlerepository
   daa5e53..b78d1a7  master      -> origin/master
Updating daa5e53..b78d1a7
Fast-forward
 Sample.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Sample.txt
```



# Getting changes from GitHub repository to local repository

- Check whether the file has been pulled.

```
mrajhans@LIN20001179 MINGW64 /c/study/GitRepos/SampleRepository (master)
$ ls -al
total 39
drwxr-xr-x 1 mrajhans 1049089  0 Jul  1 10:51 ./
drwxr-xr-x 1 mrajhans 1049089  0 Jun 30 15:59 ../
drwxr-xr-x 1 mrajhans 1049089  0 Jul  1 10:51 .git/
-rw-r--r-- 1 mrajhans 1049089 1272 Jun 27 11:30 404.html
-rw-r--r-- 1 mrajhans 1049089 3959 Jun 27 11:30 apple-touch-icon.png
-rw-r--r-- 1 mrajhans 1049089  416 Jun 27 11:30 browserconfig.xml
drwxr-xr-x 1 mrajhans 1049089  0 Jun 30 15:44 css/
-rw-r--r-- 1 mrajhans 1049089  766 Jun 27 11:30 favicon.ico
-rw-r--r-- 1 mrajhans 1049089  191 Jun 27 11:30 humans.txt
drwxr-xr-x 1 mrajhans 1049089  0 Jun 27 11:30 img/
-rw-r--r-- 1 mrajhans 1049089 5069 Jul  1 09:43 index.html
drwxr-xr-x 1 mrajhans 1049089  0 Jun 30 15:44 js/
-rw-r--r-- 1 mrajhans 1049089   27 Jul  1 10:51 Sample.txt
-rw-r--r-- 1 mrajhans 1049089 3482 Jun 27 11:30 tile.png
-rw-r--r-- 1 mrajhans 1049089 1854 Jun 27 11:30 tile-wide.png
```



# Cloning the repository

- clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location, the original repository can be located on the local filesystem or on remote machine accessible supported protocols.
- Create another directory than existing repository and navigate to it.

```
mrjghans@LIN20001179 MINGW64 /c/study
$ mkdir ClonedRepo

mrjghans@LIN20001179 MINGW64 /c/study
$ cd ClonedRepo
```



# Cloning the repository (Cont....)

- Now clone the repository into this newly created repository.

The screenshot shows the GitHub interface for a repository named 'manishrajhans / samplerepository'. The 'Code' tab is selected. A green 'Clone' button is highlighted with a red box. A dropdown menu is open, showing 'Clone with HTTPS' (also highlighted with a red box) and the URL 'https://github.com/manishrajhans/samp1' (highlighted with a red box). The repository content shows a commit by 'manishrajhans' with files 'css', 'js', and '404.html', all marked as 'First Commit'.

manishrajhans / samplerepository

<> Code ! Issues 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⚙ Settings

Branch: master ▾

Go to file Add file ▾

Clone ▾

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/manishrajhans/samp1 📄

Open in Desktop Download ZIP

manishrajhans committed b78d1a7 29 minutes ago ...

css	First Commit
js	First Commit
404.html	First Commit



# Cloning the repository (Cont....)

- Now in GitBash or command prompt run following command.

```
$ git clone https://github.com/manishrajhans/samlerepository.git
Cloning into 'samlerepository'...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 26 (delta 4), reused 21 (delta 2), pack-reused 0
Unpacking objects: 100% (26/26), 58.70 KiB | 279.00 KiB/s, done.
```

- It will clone the remote repository into the selected directory.





# Cloning the repository (Cont....)

- Now check the cloned samplerepository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo
$ cd samplerepository/

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ ls -al
total 39
drwxr-xr-x 1 mrajhans 1049089 0 Jul 2 22:11 ./
drwxr-xr-x 1 mrajhans 1049089 0 Jul 2 22:11 ../
drwxr-xr-x 1 mrajhans 1049089 0 Jul 2 22:11 .git/
-rw-r--r-- 1 mrajhans 1049089 1332 Jul 2 22:11 404.html
-rw-r--r-- 1 mrajhans 1049089 3959 Jul 2 22:11 apple-touch-icon.png
-rw-r--r-- 1 mrajhans 1049089 428 Jul 2 22:11 browserconfig.xml
drwxr-xr-x 1 mrajhans 1049089 0 Jul 2 22:11 css/
-rw-r--r-- 1 mrajhans 1049089 766 Jul 2 22:11 favicon.ico
-rw-r--r-- 1 mrajhans 1049089 206 Jul 2 22:11 humans.txt
-rw-r--r-- 1 mrajhans 1049089 5157 Jul 2 22:11 index.html
drwxr-xr-x 1 mrajhans 1049089 0 Jul 2 22:11 js/
-rw-r--r-- 1 mrajhans 1049089 27 Jul 2 22:11 sample.txt
-rw-r--r-- 1 mrajhans 1049089 3482 Jul 2 22:11 tile.png
-rw-r--r-- 1 mrajhans 1049089 1854 Jul 2 22:11 tile-wide.png
```



# Cloning the repository (Cont....)

- Edit the file sample.txt from the cloned repository, now add it to staging area. After that commit the changes and push them to remote repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ nano sample.txt

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ git add .

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ git commit -m "SampleFileUpdated"
[master c3cb1a6] SampleFileUpdated
1 file changed, 1 insertion(+)

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 319 bytes | 45.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/manishrajhans/samplerepository.git
b78d1a7..c3cb1a6 master -> master
```



# Cloning the repository (Cont....)

- Now check the cloned sample repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo
$ cd samplerepository/

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ ls -al
total 39
drwxr-xr-x 1 mrajhans 1049089  0 Jul  2 22:11 ./
drwxr-xr-x 1 mrajhans 1049089  0 Jul  2 22:11 ../
drwxr-xr-x 1 mrajhans 1049089  0 Jul  2 22:11 .git/
-rw-r--r-- 1 mrajhans 1049089 1332 Jul  2 22:11 404.html
-rw-r--r-- 1 mrajhans 1049089 3959 Jul  2 22:11 apple-touch-icon.png
-rw-r--r-- 1 mrajhans 1049089  428 Jul  2 22:11 browserconfig.xml
drwxr-xr-x 1 mrajhans 1049089  0 Jul  2 22:11 css/
-rw-r--r-- 1 mrajhans 1049089  766 Jul  2 22:11 favicon.ico
-rw-r--r-- 1 mrajhans 1049089  206 Jul  2 22:11 humans.txt
-rw-r--r-- 1 mrajhans 1049089 5157 Jul  2 22:11 index.html
drwxr-xr-x 1 mrajhans 1049089  0 Jul  2 22:11 js/
-rw-r--r-- 1 mrajhans 1049089  27 Jul  2 22:11 Sample.txt
-rw-r--r-- 1 mrajhans 1049089 3482 Jul  2 22:11 tile.png
-rw-r--r-- 1 mrajhans 1049089 1854 Jul  2 22:11 tile-wide.png
```



# Cloning the other users repository

- To clone the other user's repository, click Code, and ensure you have selected clone with HTTPS.

The screenshot shows the GitHub interface for a repository named 'manishrajhans / samplerepository'. The 'Code' button is highlighted with a red box. A dropdown menu is open, showing the 'Clone with HTTPS' option, which is also highlighted with a red box. The URL 'https://github.com/manishrajhans/samp1' is displayed in the dropdown. Other options like 'Open with GitHub Desktop' and 'Download ZIP' are visible.

manishrajhans / samplerepository

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Branch: master

Go to file Add file Code

manishcrajhans committed 687aaaf 21 hours ago

css	First Commit
js	First Commit
404.html	First Commit
Sample.txt	FirstUpdateAfterClone
apple-touch-icon.png	First Commit

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/manishrajhans/samp1

Open with GitHub Desktop

Download ZIP

3 days ago

# Cloning the other users repository (Cont....)



- Now on your machine, you need to create and choose directory in which you want to clone the directory and clone the repository with the path that you have copied.

```
$ git clone https://github.com/manishrajhans/samplerepository.git
Cloning into 'samplerepository'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 32 (delta 7), reused 26 (delta 3), pack-reused 0
Unpacking objects: 100% (32/32), 59.20 KiB | 31.00 KiB/s, done.
```

- Now you can check that cloned sample directory is added on your machine.

```
$ ls -al
total 20
drwxr-xr-x 1 Manish - winskill 197121 0 Jul  3 19:37 ./
drwxr-xr-x 1 Manish - winskill 197121 0 Jul  3 19:36 ../
drwxr-xr-x 1 Manish - winskill 197121 0 Jul  3 19:37 samplerepository/
```

# Cloning the other users repository (Cont....)



- Navigate to the directory.

```
$ cd samplerepository/
```

- You can now check the content of directory which is exact copy of the repository we cloned..

```
$ ls -al
total 39
drwxr-xr-x 1 Manish - winskill 197121 0 Jul 3 19:37 ./
drwxr-xr-x 1 Manish - winskill 197121 0 Jul 3 19:37 ../
drwxr-xr-x 1 Manish - winskill 197121 0 Jul 3 19:37 .git/
-rw-r--r-- 1 Manish - winskill 197121 1332 Jul 3 19:37 404.html
-rw-r--r-- 1 Manish - winskill 197121 3959 Jul 3 19:37 apple-touch-icon.png
-rw-r--r-- 1 Manish - winskill 197121 428 Jul 3 19:37 browserconfig.xml
drwxr-xr-x 1 Manish - winskill 197121 0 Jul 3 19:37 css/
-rw-r--r-- 1 Manish - winskill 197121 766 Jul 3 19:37 favicon.ico
-rw-r--r-- 1 Manish - winskill 197121 206 Jul 3 19:37 humans.txt
-rw-r--r-- 1 Manish - winskill 197121 5157 Jul 3 19:37 index.html
drwxr-xr-x 1 Manish - winskill 197121 0 Jul 3 19:37 js/
-rw-r--r-- 1 Manish - winskill 197121 84 Jul 3 19:37 Sample.txt
-rw-r--r-- 1 Manish - winskill 197121 3482 Jul 3 19:37 tile.png
-rw-r--r-- 1 Manish - winskill 197121 1854 Jul 3 19:37 tile-wide.png
```

# Cloning the other users repository (Cont....)



- We will make the changes to Sample .txt file, hence open it and make some changes..

```
$ nano sample.txt
```

- After making changes, add files into staging area.

```
$ git add .
```

- Now commit the changes to the local repository.

```
$ git commit -m "Made changes into sample file from cloned repo"  
[master 395b962] Made changes into sample file from cloned repo  
1 file changed, 1 insertion(+), 1 deletion(-)
```

# Cloning the other users repository (Cont....)



- Finally push the changes to the repository.

```
$ git push
remote: Permission to manishrajhans/samplerepository.git denied to manishcrajhans.
fatal: unable to access 'https://github.com/manishrajhans/samplerepository.git/': The requested
URL returned error: 403
```

- As you are not the owner of this repository, you are not allowed to push any changes to this repository hence you will get an error.
- To collaborate to this repository, you need to fork the repository.





# Branching the repository

- Navigate to the samplerepository's master branch in GitBash (if not already there)
- Now type git branch command to see what current branch you are in.

```
$ git branch  
* master
```

- Now create the branch to git command by passing branch name to git branch command.

```
$ git branch NewFeature
```



# Branching the repository (Cont....)

- Now switch to the newly created branch with checkout command.

```
mrajhans@LIN20001179 MINGW64 /c/study/gitrepos/SampleRepository (master)
$ git checkout NewFeature
Switched to branch 'NewFeature'

mrajhans@LIN20001179 MINGW64 /c/study/gitrepos/SampleRepository (NewFeature)
$ |
```

- After switching to new branch, check the status.

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



# Branching the repository (Cont....)

- Modify the sample.txt.

```
$ nano sample.txt
```

- Now check the status of feature branch.

```
mrajhans@LIN20001179 MINGW64 /c/study/gitrepos/SampleRepository (NewFeature)
$ git status
On branch NewFeature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   sample.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

- Now Add the file into staging area.

```
$ git add .
```



# Branching the repository (Cont....)

- Committing changes to NewFeature branch.

```
$ git commit -m "First commit to NewFeaturesBranch"  
[NewFeature 0e063e1] First commit to NewFeaturesBranch  
1 file changed, 2 insertions(+)
```

- Now check the status.

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

- To merge the changes back to the master branch, checkout to the master branch.

```
$ git checkout master  
Switched to branch 'master'  
Your branch is up to date with 'origin/master'.
```



# Branching the repository (Cont....)

- Check the changes done from the NewFeature branch.

```
$ git diff NewFeature
diff --git a/Sample.txt b/Sample.txt
index f0e8c74..54a3e09 100644
--- a/Sample.txt
+++ b/Sample.txt
@@ -1,3 +1 @@
  This is sample text file.
-
-This new feature change added through feature branch.
```

- Now use merge changes back to the master branch.

```
$ git merge NewFeature
Updating b78d1a7..0e063e1
Fast-forward
 Sample.txt | 2 ++
 1 file changed, 2 insertions(+)
```



# Branching the repository (Cont....)

- You can check how many branches exists and which is current branch.

```
$ git branch
NewFeature
* master
```

- You can also delete the created branch.

```
$ git branch -d NewFeature
Deleted branch NewFeature (was 0e063e1).
```

- Now you can check, the branch NewFeatures is deleted.

```
$ git branch
* master
```



# Checking the logs of repository

- You can check the write logs of repository with git log command.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git log
commit 0e063e154dde06233652f90a712de82a832a7ade (HEAD -> master)
Author: Manish R <manish.rajhans@outlook.com>
Date:   Sun Jul 5 15:55:34 2020 +0530

    First commit to NewFeaturesBranch

commit b78d1a797079bedb96b3e4193af28cde223ef50e (origin/master)
Author: manishrajhans <67506754+manishrajhans@users.noreply.github.com>
Date:   Wed Jul 1 10:43:25 2020 +0530

    Create Sample.txt

commit daa5e5337b1bf41b8ef11a817a65d82581dc9a8a
Author: Manish Rajhans <manish.rajhans@capgemini.com>
Date:   Wed Jul 1 09:49:19 2020 +0530

    IndexFileChange01

commit d514e911fe360b9a651def58e65345a270b8e877
Author: Manish Rajhans <manish.rajhans@capgemini.com>
Date:   Tue Jun 30 16:31:24 2020 +0530

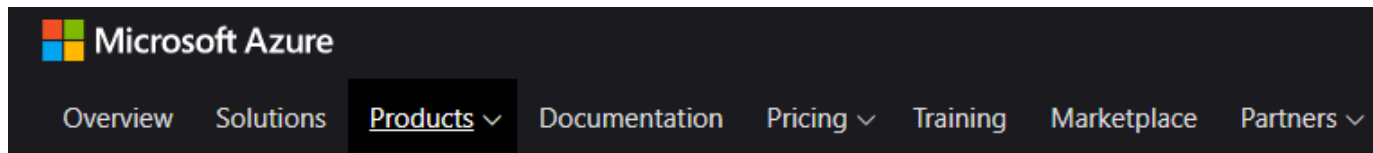
    Updated index file commit
:...skipping...
commit 0e063e154dde06233652f90a712de82a832a7ade (HEAD -> master)
Author: Manish R <manish.rajhans@outlook.com>
Date:   Sun Jul 5 15:55:34 2020 +0530

    First commit to NewFeaturesBranch
```



# Signing up to Azure DevOps

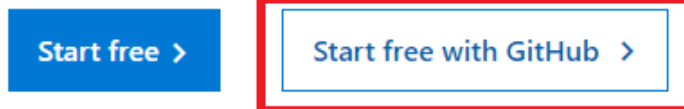
- Navigate to <https://azure.microsoft.com/en-in/services/devops/> & click on Start with GitHub.



[Home](#) / [Services](#) / Azure DevOps

## Azure DevOps

Plan smarter, collaborate better and ship faster with a set of modern dev services.



Already have an account?

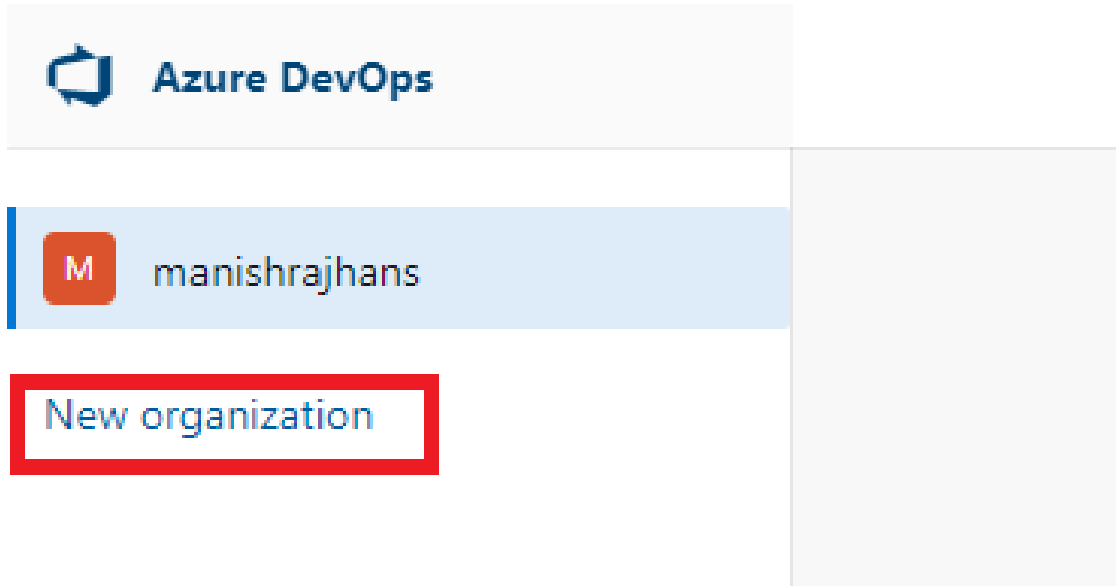
[Sign in to Azure DevOps](#)





# Signing up to Azure DevOps

- After login, Click on New Organization.





# Signing up to Azure DevOps

- Provide the organization name and select the region and click on Continue.

Azure DevOps

manish.rajhans@capgemini.com

**Almost done...**

Name your Azure DevOps organization

dev.azure.com/ cgmumorg

We'll host your projects in

South India

Enter the characters you see

[New](#) | [Audio](#)

Rpd LXQ4

RpdLXQ4

Continue

# Creating Project



Azure DevOps

Search



MR

c cgmumorg

M manishrajhans

New organization

Sprint 172

Sprint 172 Release Notes

After several sprints in preview, we are now announcing the general release of state transition restriction rules to all customers as part of the Sprint 172 Update. [Learn more](#) 🖱️

## Create a project to get started

Project name \*

SampleProject



### Visibility



Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.

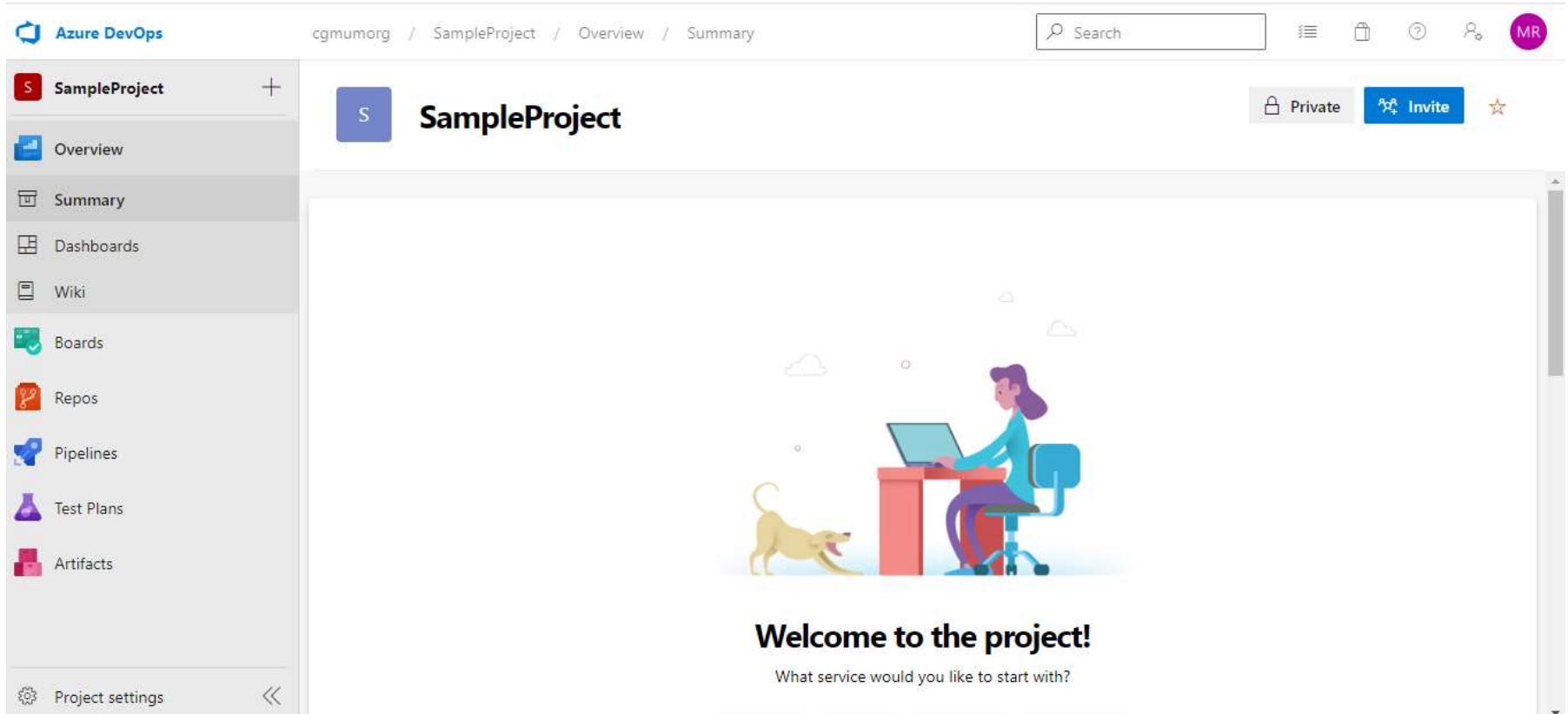


Private

Only people you give access to will be able to view this project.

+ Create project

# Creating Project



# Initializing the git repository



Azure DevOps

cgmmumorg / SampleProject / Repos / Files / **SampleProject**

Search

**SampleProject**

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Pipelines

Test Plans

Artifacts

Project settings

**HTTPS** SSH `https://cgmumorg@dev.azure.com/cgmumorg/SampleProject/_git/S` OR Clone in VS Code

Generate Git Credentials

Having problems authenticating in Git? Be sure to get the latest version [Git for Windows](#) or our plugins for [IntelliJ](#), [Eclipse](#), [Android Studio](#) or [Windows command line](#).

Push an existing repository from command line

**HTTPS** SSH

```
git remote add origin  
https://cgmumorg@dev.azure.com/cgmumorg/SampleProject/_git/SampleProject
```

Import a repository

Import

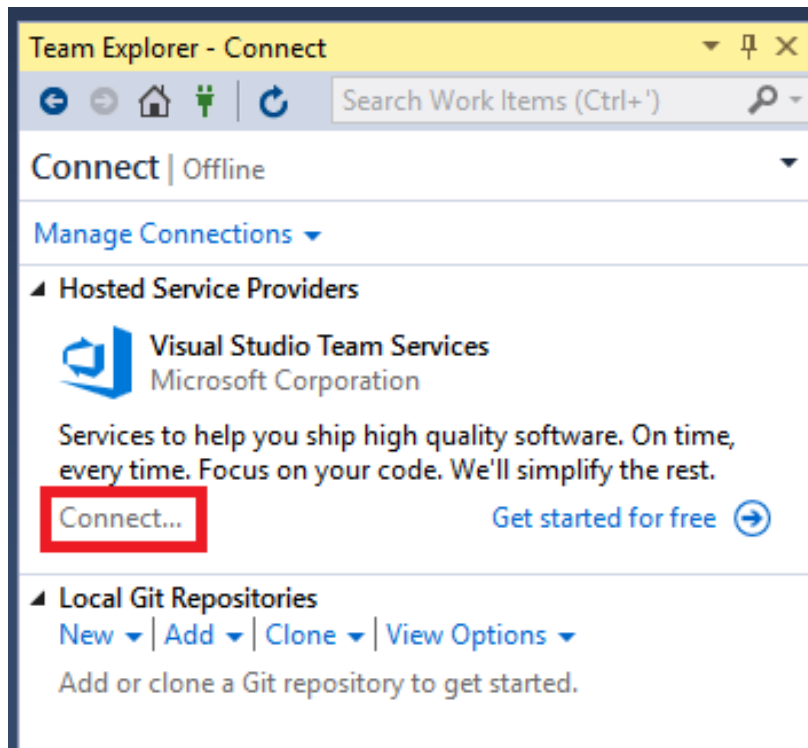
Initialize with a README or gitignore

☒ Add a README Add a .gitignore: **None** **Initialize**



# Connecting to Azure DevOps

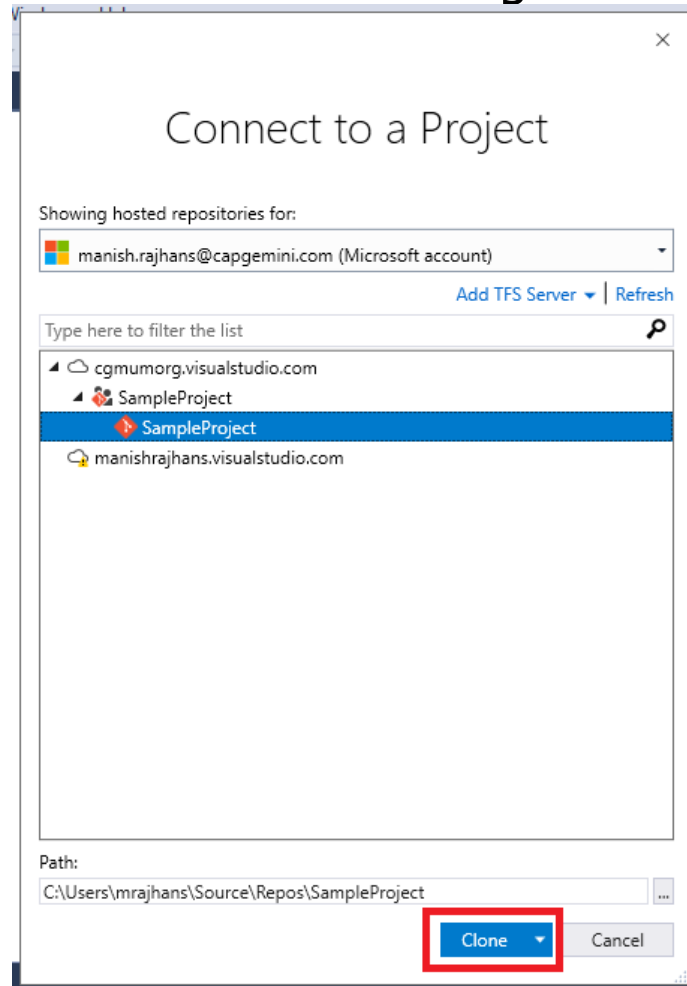
- Open the team explorer, Click on Connect as below, it will ask you for credentials.. Provide them and login.





# Connecting to Azure DevOps

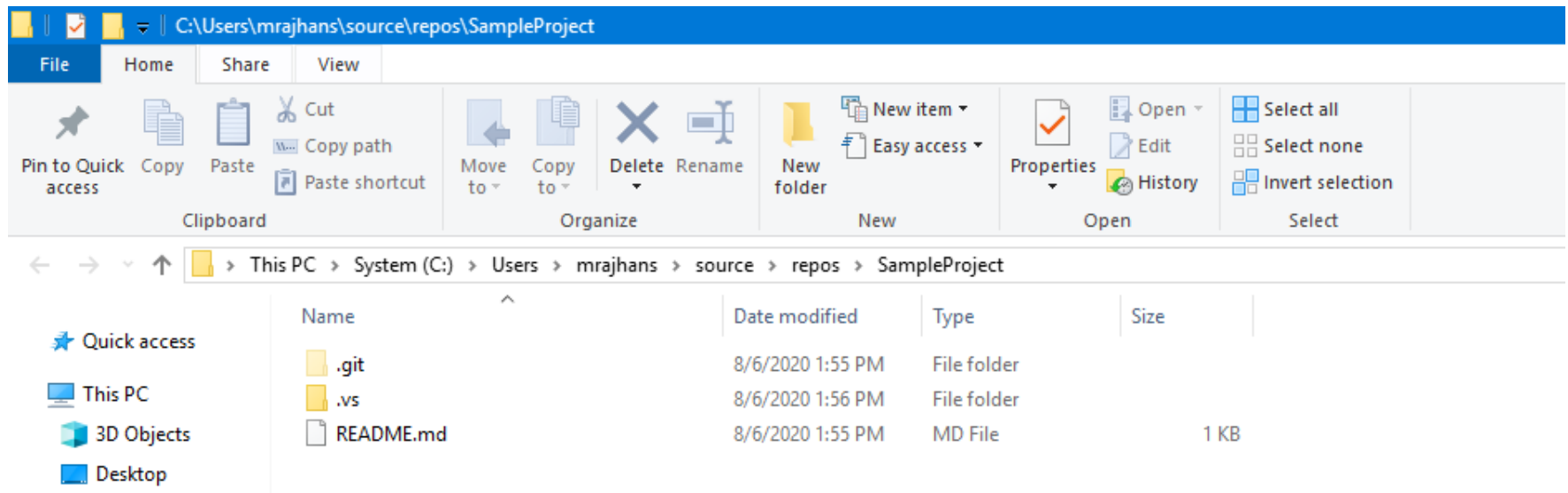
- Open the team explorer, Click on Connect as below, it will ask you for credentials.. Provide them and login.



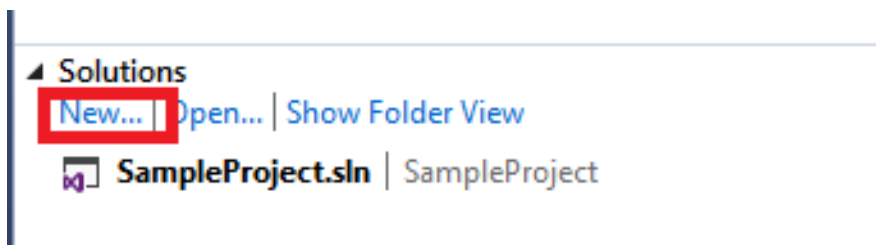


# Cloning the Azure DevOps repository

- After cloning repository, it is now available on you machine also as shown below.



- In team explorer now click on New to create New solution in the cloned repository.

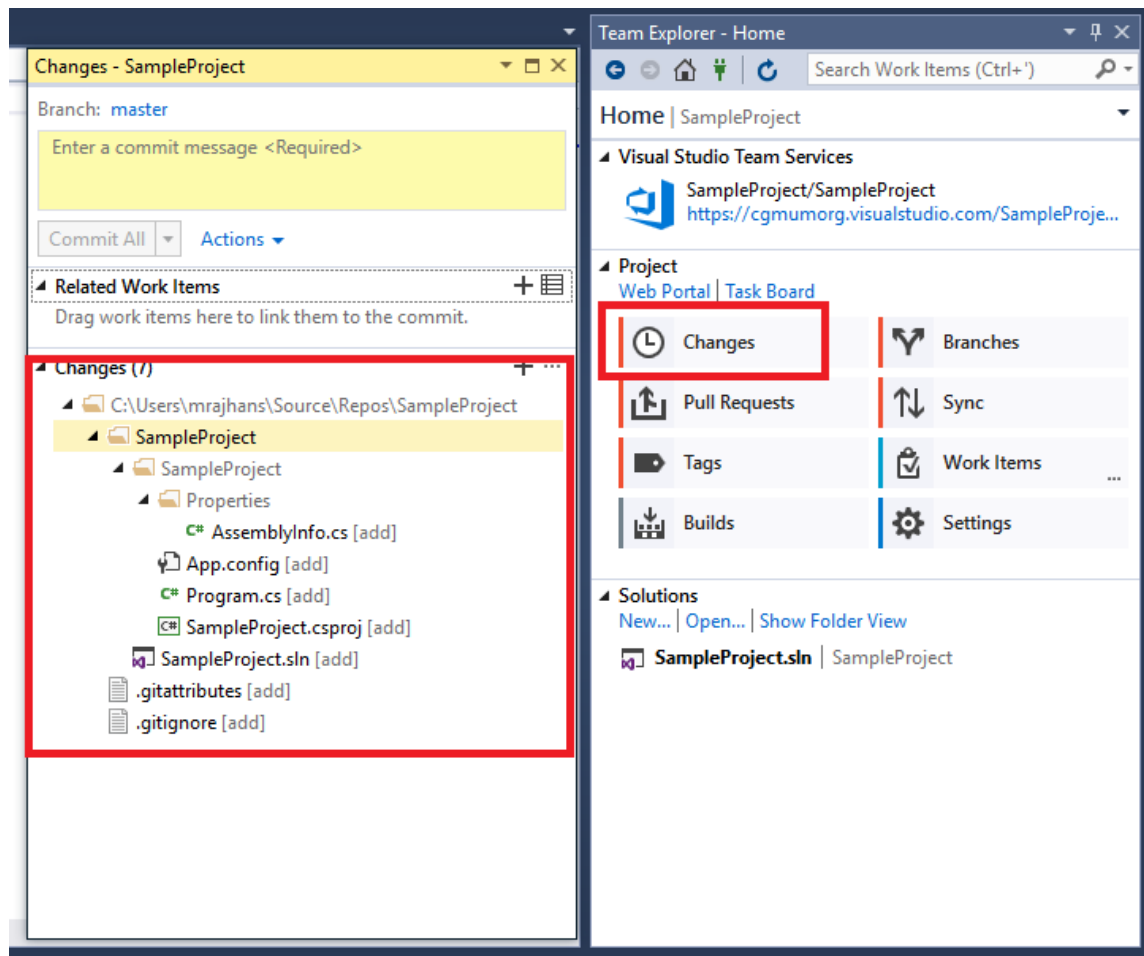






# Committing changes to local repo

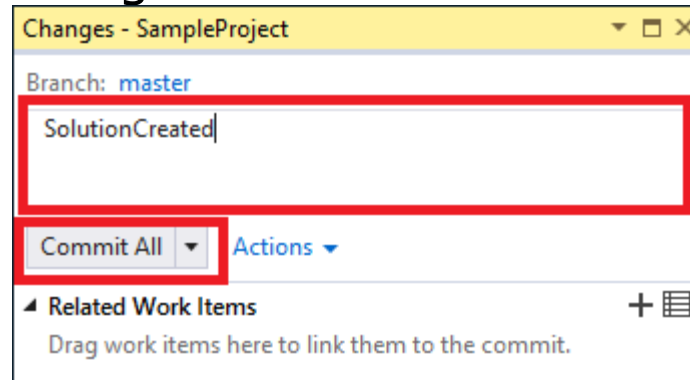
- Click on Changes as shown below and you will be able to see all the changes.



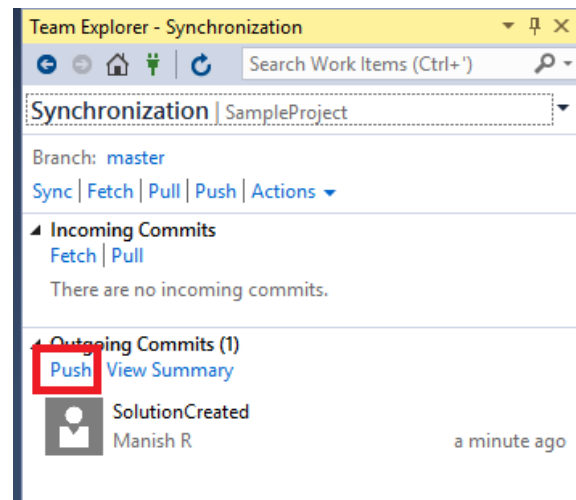


# Committing changes to local repo

- Specify Commit message and click on Commit Changes.



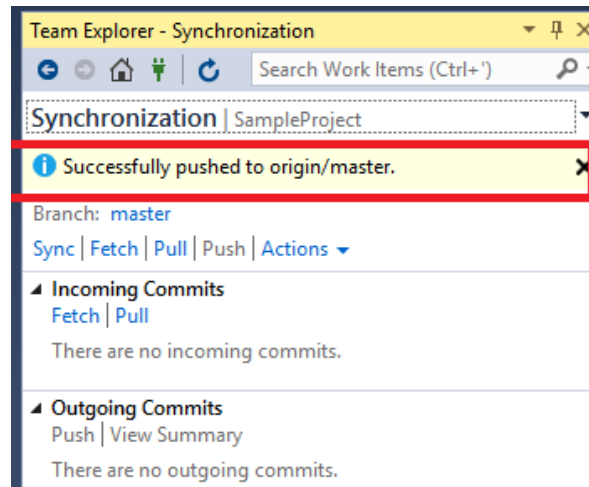
- Now Click on Synch to write the committed changes to Server. ^  
Click on Push





# Pushing changes to Azure DevOps

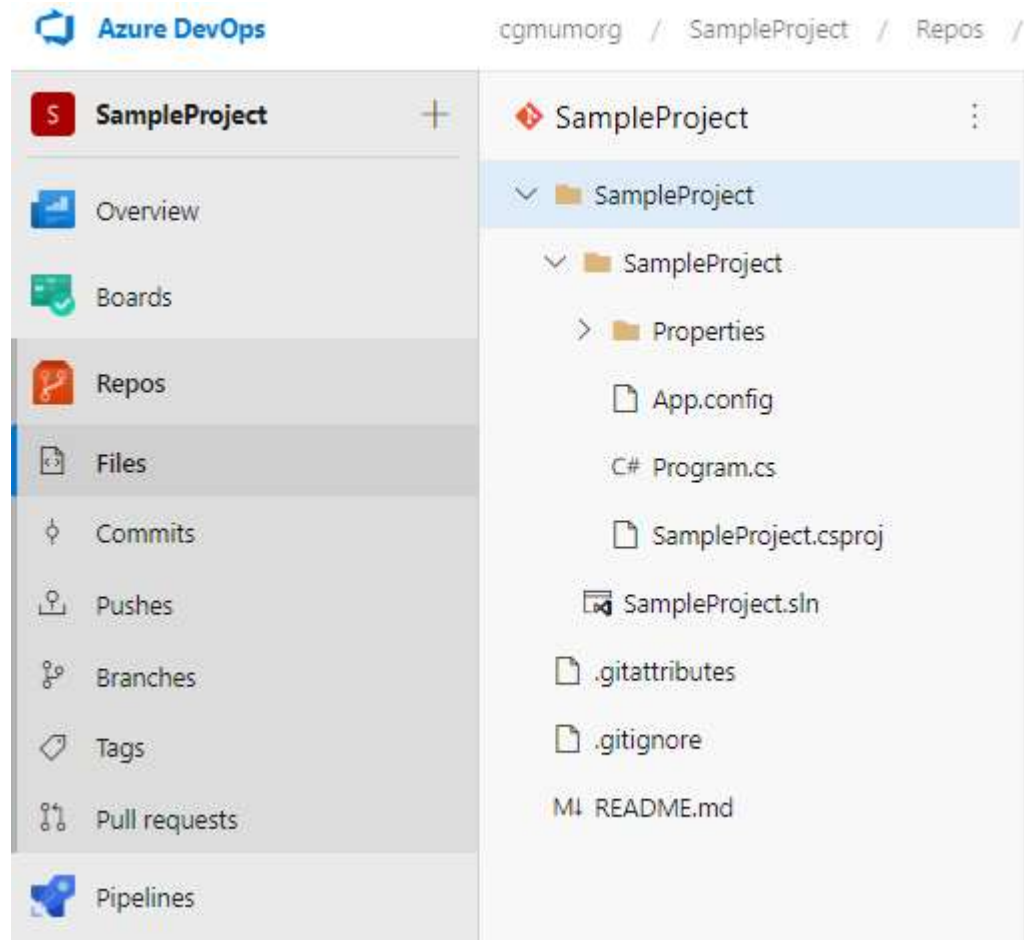
- Now you can check that all the changes are pushed to Server.





# Pushing changes to Azure DevOps

- Now you can see, all the changes are written to Server.





# Summary

- In this lesson, you have learnt about:
- Creating local & remote repository.
  - Cloning the repository
  - Forking the repository
  - Pushing the changes to the repository
  - Creating branch
  - Merging branch
  - Viewing write history.
  - Creating Azure DevOps account
  - Creating & Collaborating with Azure DevOps repository





# Review Questions

- \_\_\_\_\_ command is used to collaborate to the other users repository.
- \_\_\_\_\_ command is used to view the right history.
- \_\_\_\_\_ command shows the status of the repository.
- \_\_\_\_\_ area contains all the tracked files.
- \_\_\_\_\_ option in Visual Studio Team Explorer is used to synchronize the changes before pushing or pulling the content to or from Azure Devops repository.

