accenturetechnology

Learning and Knowledge Management

Module 3: Basic Git Commands

Module Objectives

At the end of this module, you will be able to:

- Describe Git Hub Repository.
- Describe techniques like Branching, Commiting, Pull Requests, and Merging with respect to GITHub Repository.
- Apply various Git Bash commands.
- Use the commands to synchronize the local and remote repository.

Topic List

- GitHub Repository
- Git Bash Commands
- Synchronizing Local and Remote Repository
- Exercise 3.1: Working in GitHub
- Exercise 3.2: Using Git Basic Commands

Topic List

- **GitHub Repository**
- Git Bash Commands
- Synchronizing Local and Remote Repository
- Exercise 3.1: Working in GitHub
- Exercise 3.2: Using Git Basic Commands

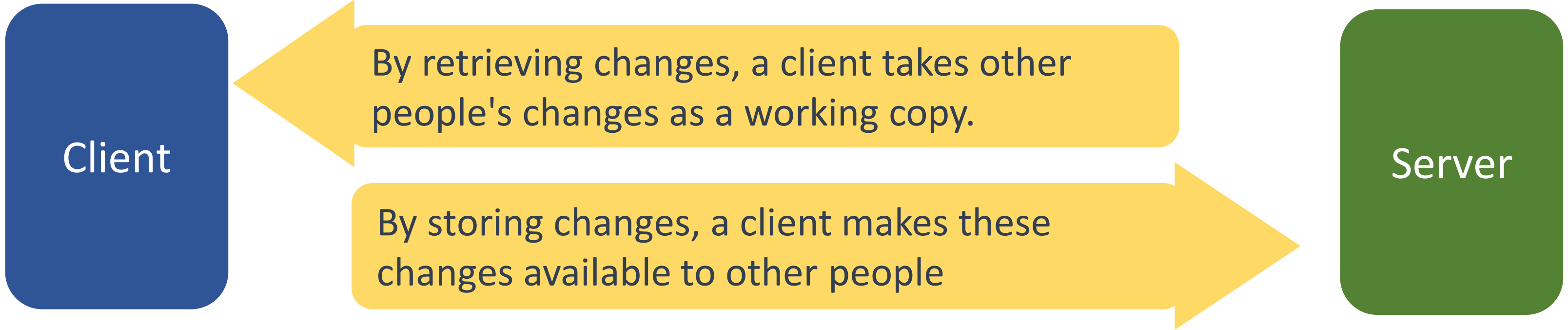
GitHub Repository (1)

What is a GitHub Repository?

A GITHub repository stores all the files for a particular project.

It is accessed over a network, acting as a server and version control tool acting as a client.

Clients can connect to the repository, and then they can store/retrieve their changes to/from the repository.



By retrieving changes, a client takes other people's changes as a working copy.

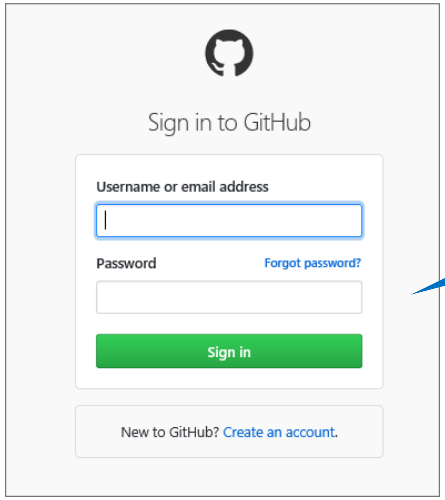
By storing changes, a client makes these changes available to other people

GitHub Repository (2)

Steps to Create a Github Account

Following are the steps to create an account in GitHub:

1. Login to **https://github.com/join** in new browser.
2. Create new account by giving username, email address and password and login.
3. Provide the necessary required details and create an account.
4. After creating the account you need to verify it using the given email address. After verification you can login to your GitHub account.



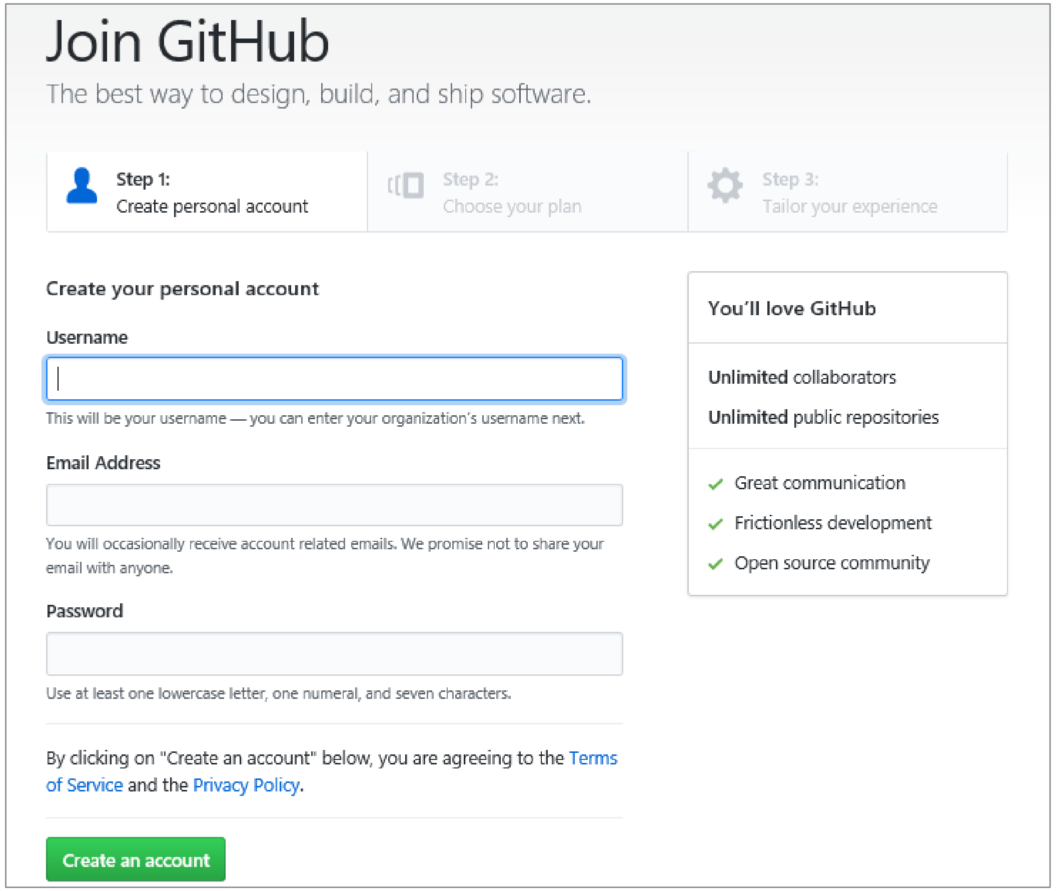
Sign in to GitHub

Username or email address

Password

New to GitHub? Create an account.

The Login page of Git Hub



Join GitHub

The best way to design, build, and ship software.

Step 1: Create personal account
Step 2: Choose your plan
Step 3: Tailor your experience

Create your personal account

Username

This will be your username — you can enter your organization's username next.

Email Address

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the Terms of Service and the Privacy Policy.

You'll love GitHub

Unlimited collaborators

Unlimited public repositories

✓ Great communication

✓ Frictionless development

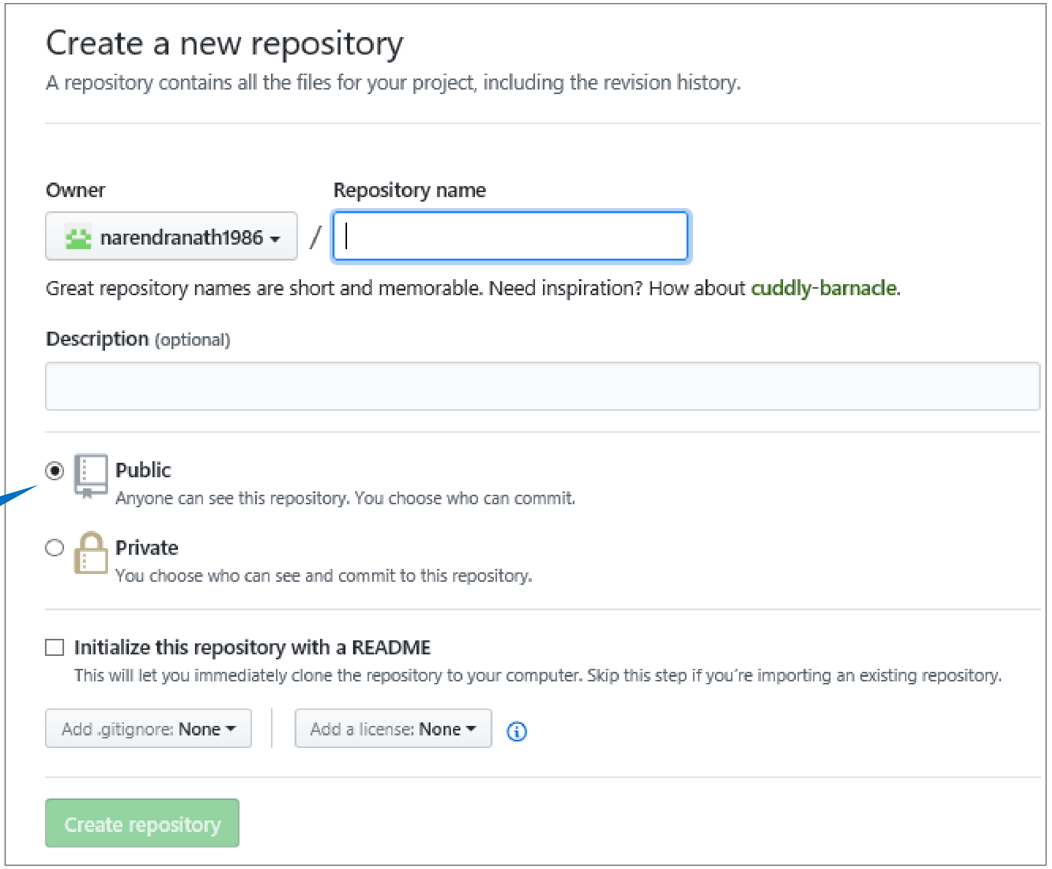
✓ Open source community

GitHub Repository (3)

Steps to Create a Repository in GitHub

Following screen capture shows the steps to create a repository in Git

1. Provide name of new Repository. Example: SampleProject
2. Select the **Public** or **Private** check box.
3. Select the “**Initialize this repository with a README**” option.
4. Click **Create Repository**.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: narendranath1986 / Repository name:

Great repository names are short and memorable. Need inspiration? How about cuddly-barnacle.

Description (optional)

● Public
Anyone can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you’re importing an existing repository.

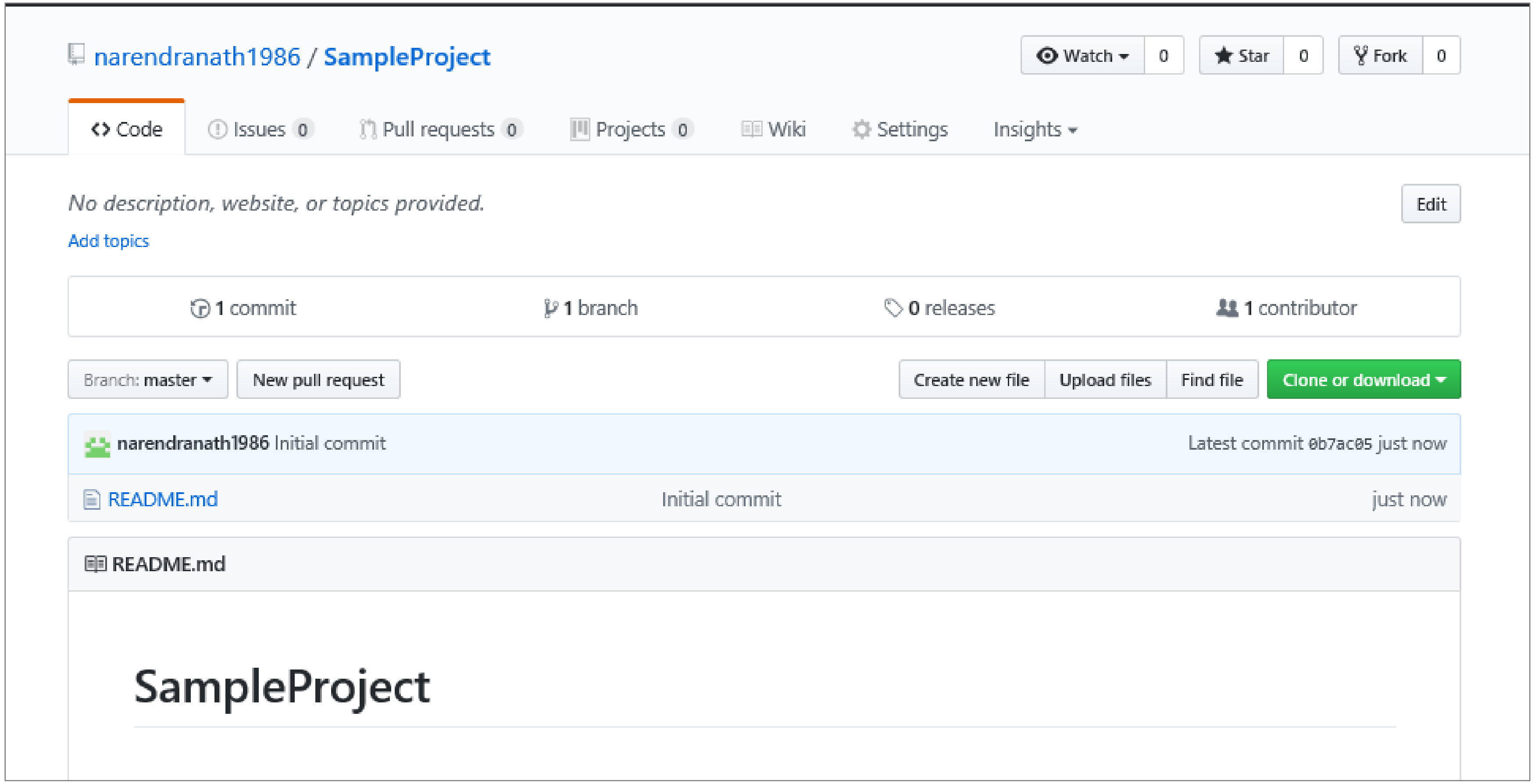
Name the repository and create a repository for your GitHub account.

Note: Each project has its own repository with a unique URL.

GitHub Repository (4)

GitHub Repository Project

Following screen capture shows of GitHub Repository project:



narendranath1986 / SampleProject

No description, website, or topics provided.

Add topics

1 commit | 1 branch | 0 releases | 1 contributor

narendranath1986 Initial commit — Latest commit 0b7ac05 just now

README.md — Initial commit — just now

README.md

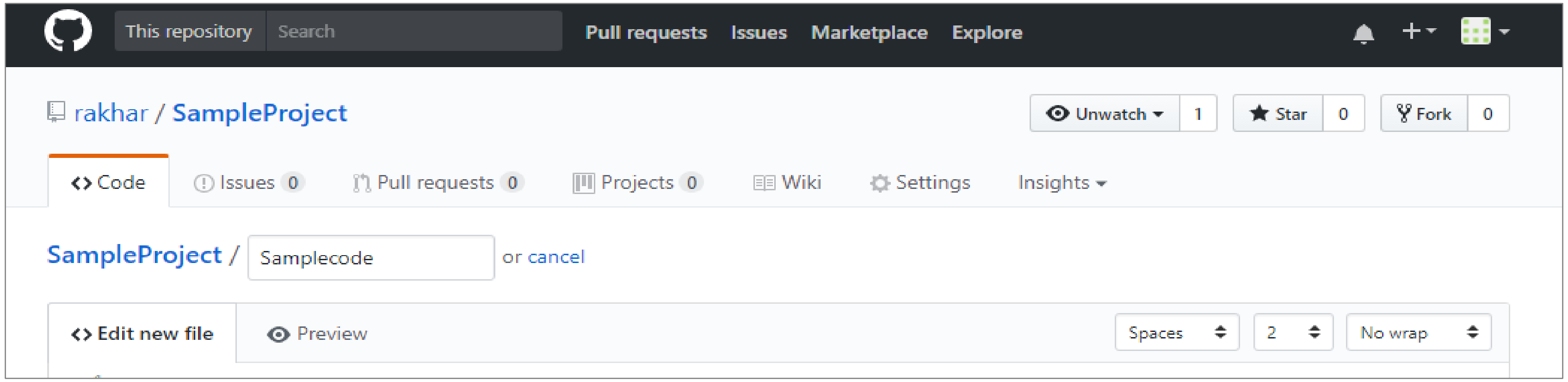
SampleProject

GitHub Repository (5)

Adding Files to the Project

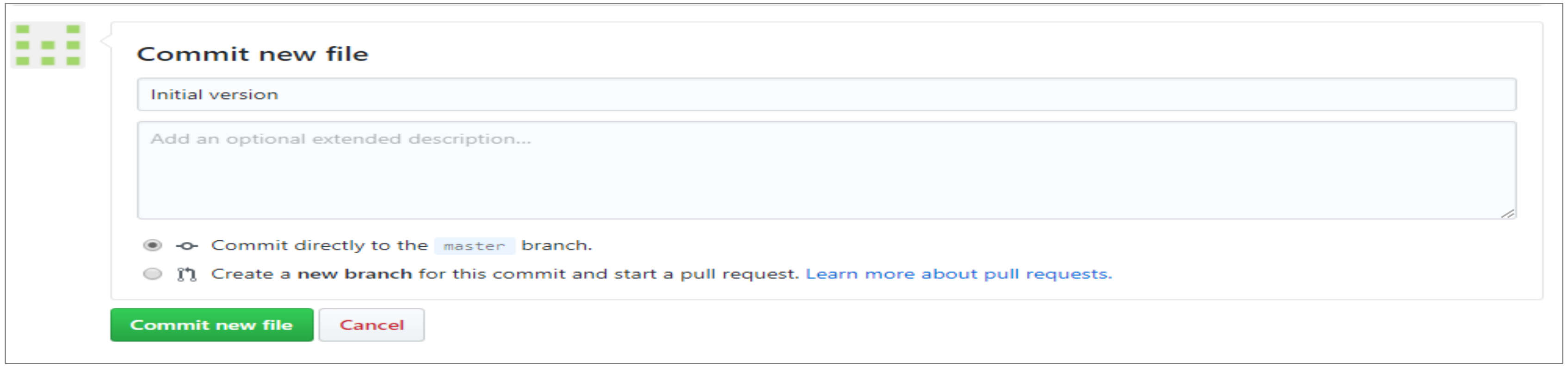
Following screen capture show how add files to the project:

Create new file, add lines of code, give comments and click **Commit new File**.



SampleProject / Samplecode or cancel

Spaces 2 No wrap



Initial version

Add an optional extended description...

● Commit directly to the `master` branch.

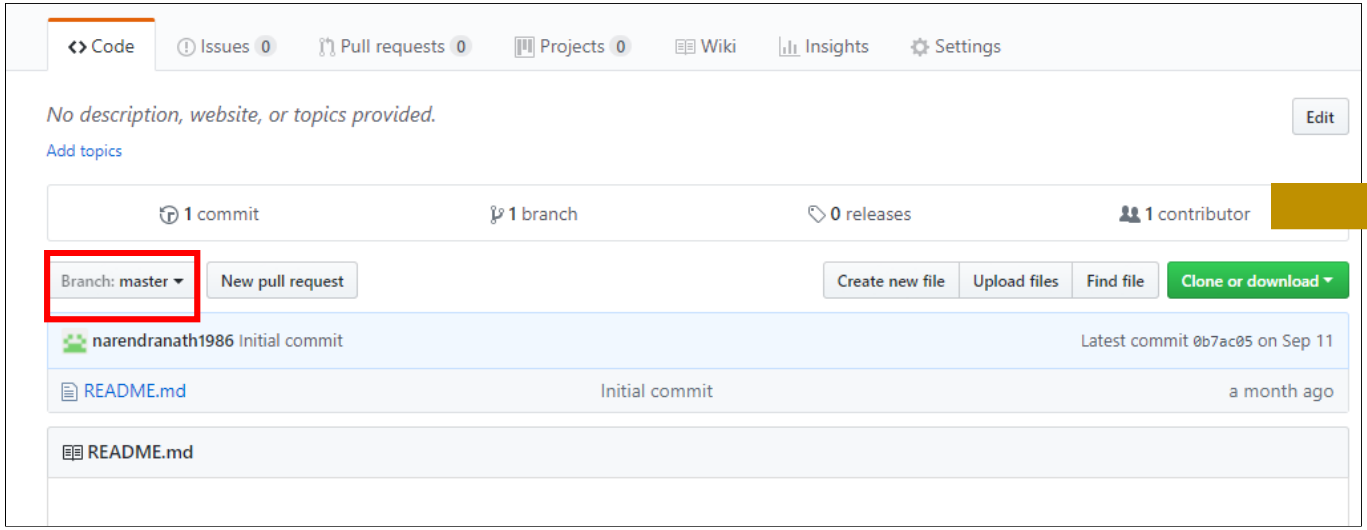
○ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

GitHub Repository (6)

Branching in GitHub

- Branching enables to operate on various versions of a repository at the same time.
- Master is the default branch of a repository.
- A copy or snapshot of the master is captured when a branch is created from the master.
- Changes made to the master by a different user, while working on the branch copy that can be pulled to the branch using the pull utility.

A new branch can be created by selecting the Branch menu in the repository.



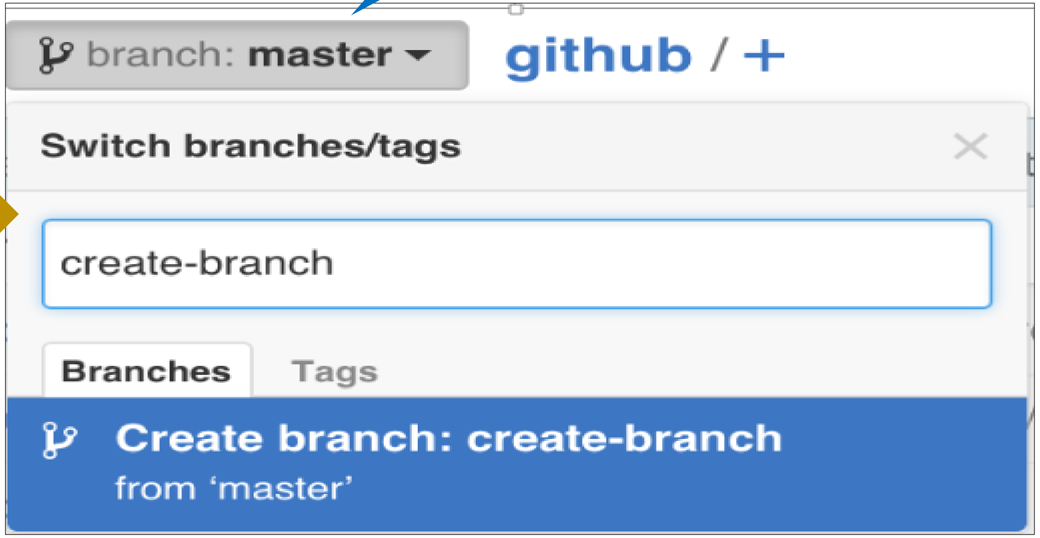
No description, website, or topics provided.

1 commit | 1 branch | 0 releases | 1 contributor

narendranath1986 Initial commit — Latest commit 0b7ac05 on Sep 11

README.md — Initial commit — a month ago

README.md



create-branch

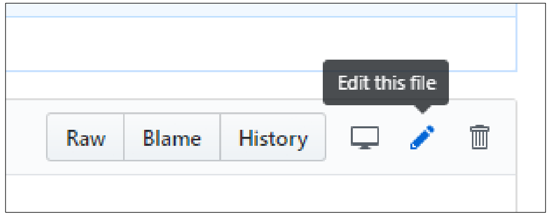
Create branch: create-branch from 'master'

GitHub Repository (7)

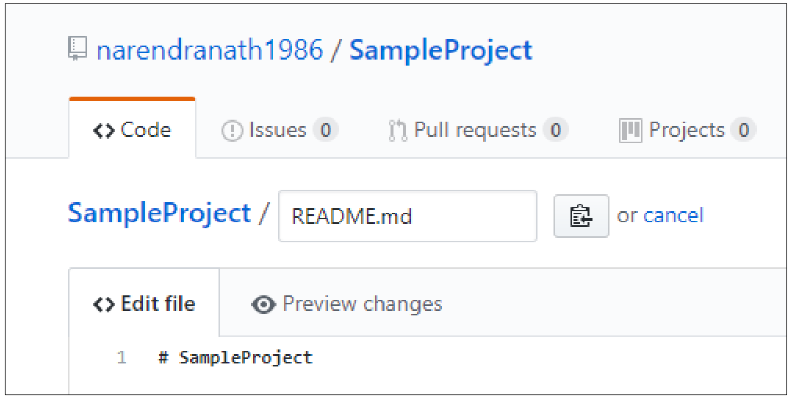
Commit in GitHub

Following are the steps to edit and commit the changes in a file:

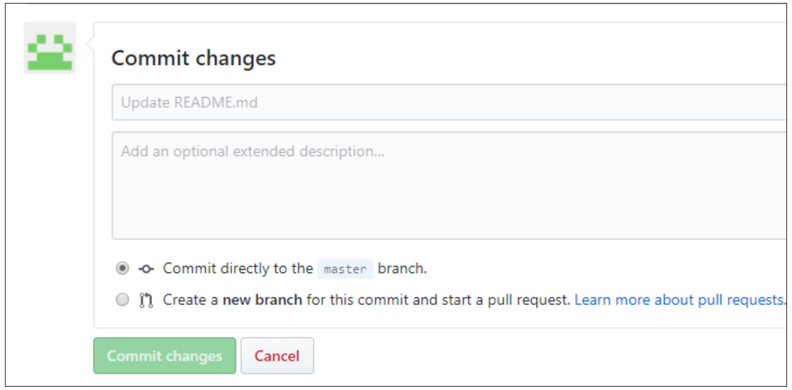
1. Click any file that had been created in the repository or click on the default **README.md** file which was created along with the repository.
2. Click the **Edit this file** icon.
3. Edit the file content in the **Edit file** tab.



4. Click **Preview changes** to view the changes. The new content will be displayed in Green.
5. Give a meaningful comment when committing which explains the changes made to the file.
6. One can choose to commit the changes to the current branch or to a new branch.
7. On master branch, choose to commit the changes in a new branch and then pull the changes through the pull request.
8. Click **Commit changes** to propose the file changes.



```
1  # SampleProject
```



Commit changes

Update README.md

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.

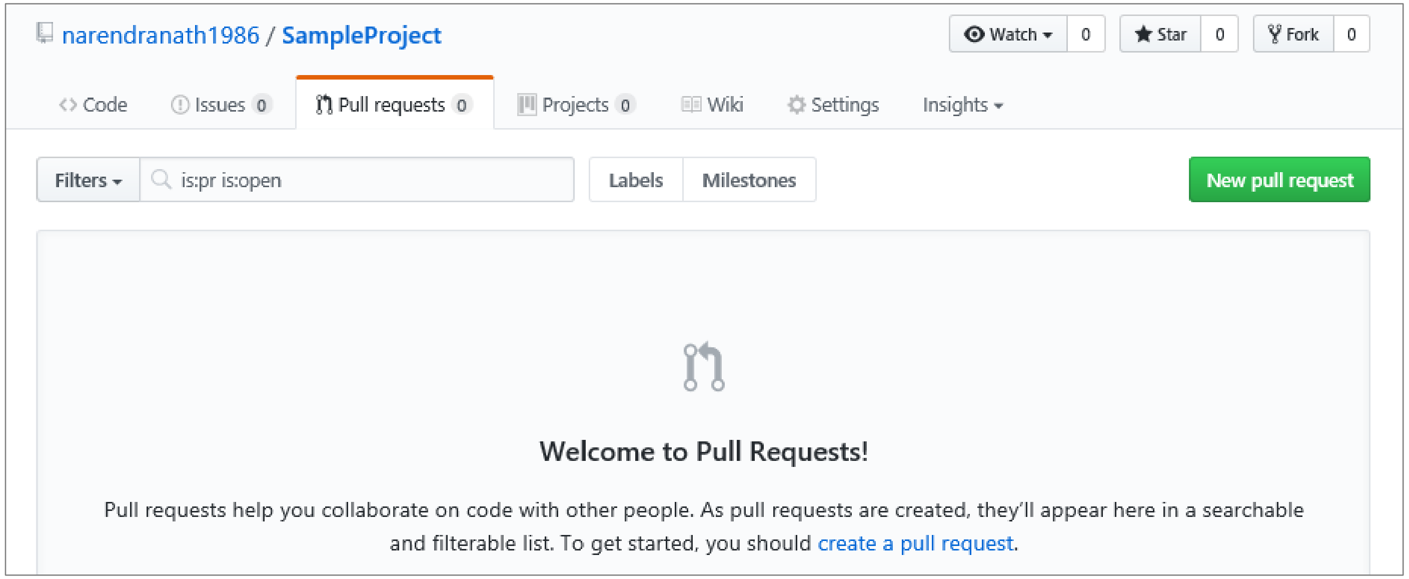
GitHub Repository (8)

Pull Requests

- Pull requests identify the differences of the content in two branches.
- All the differences and editions are highlighted in Green and Red.

Following are the steps to create Pull Requests:

1. Select the branch where the content was edited to compare with master (the original).
2. Click the **Create Pull Request** button.
3. When a pull request is created, the changes made are proposed, requesting user for review. The user can pull in the changes made and merge into the branch.
4. Click the **Pull requests** tab.
5. In the Pull Requests page, click the **New Pull Request** button.



Welcome to Pull Requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should create a pull request.

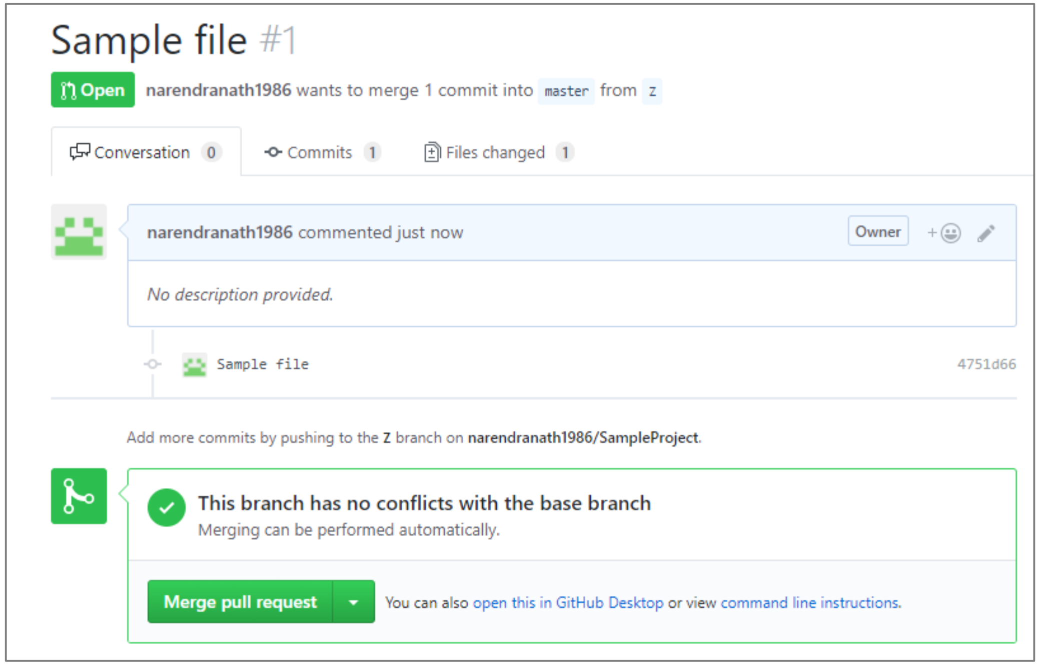
GitHub Repository (9)

Merge Changes through Pull Request

Following are the steps involved to merge the changes through pull requests:

1) To synchronize the branches, you can merge a branch into the master branch.

2) In order to merge changes into the master, click the green button for **Merge pull request**.



Sample file #1

Open narendranath1986 wants to merge 1 commit into `master` from `z`

narendranath1986 commented just now

No description provided.

`Sample file` 4751d66

Add more commits by pushing to the Z branch on narendranath1986/SampleProject.

This branch has no conflicts with the base branch

Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Example

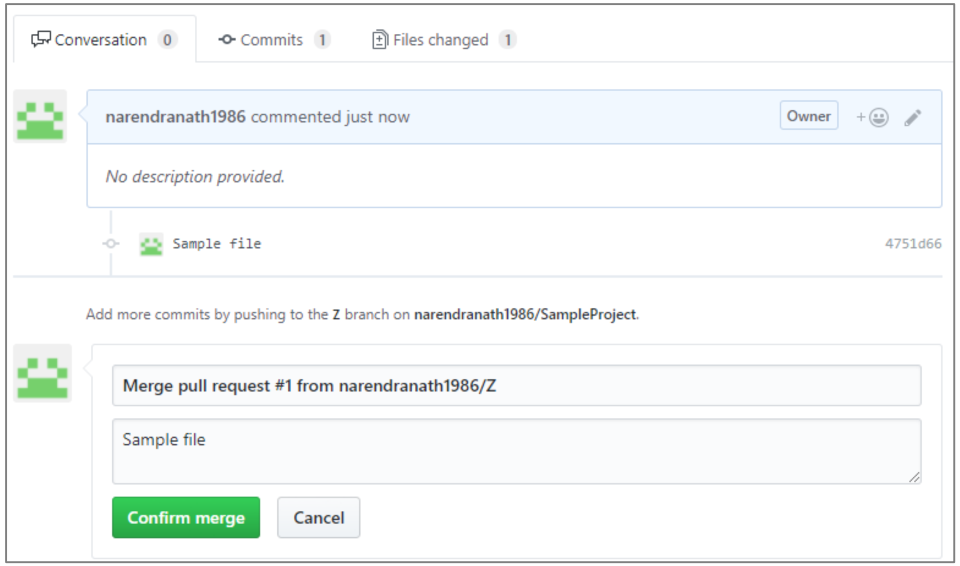
The file 1 in the Z branch to be synchronized with the master branch.

GitHub Repository (10)

Merge Changes through Pull Request

Following are the steps involved to merge the changes through pull requests:

3. Click **Confirm merge**.



narendranath1986 commented just now

No description provided.

Sample file 4751d66

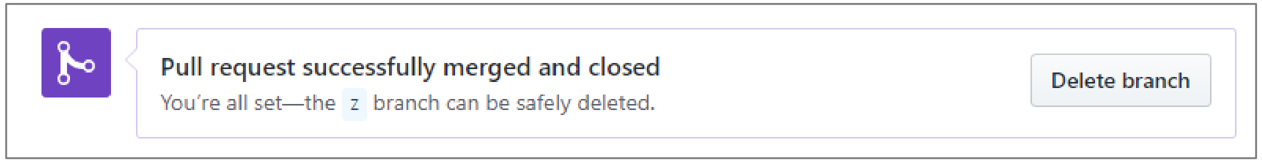
Add more commits by pushing to the Z branch on narendranath1986/SampleProject.

Merge pull request #1 from narendranath1986/Z

Sample file

Now, the branch Z can be deleted, as the changes have been incorporated into the master branch.

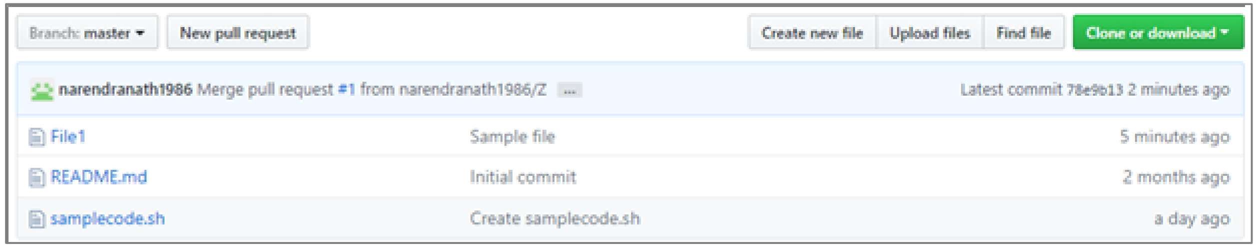
4. To delete, you can click the **Delete branch** button.



Pull request successfully merged and closed

You're all set—the z branch can be safely deleted.

In the Master branch, after synchronization you can notice the file1.



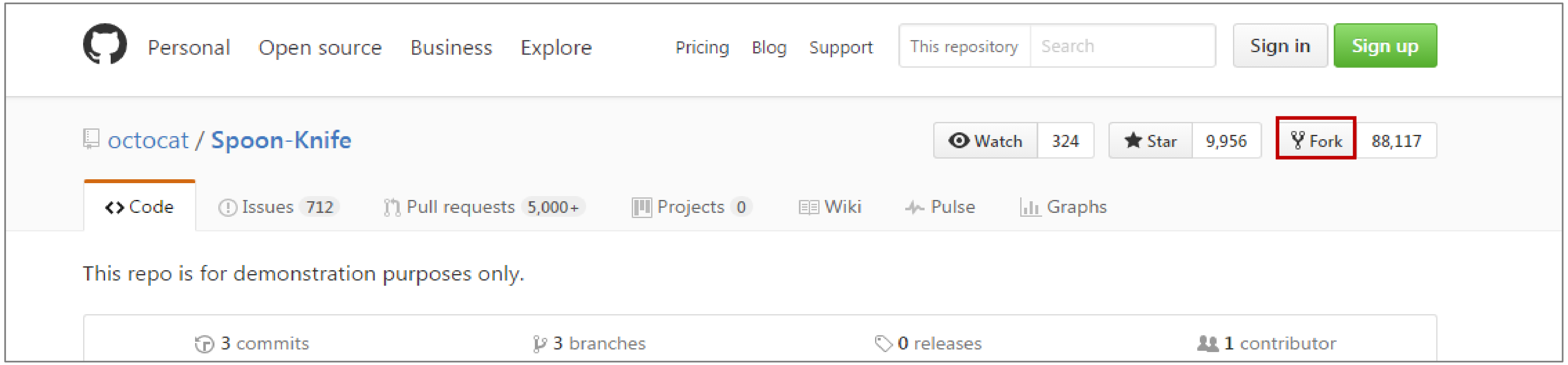
narendranath1986 Merge pull request #1 from narendranath1986/Z ... Latest commit 78e9b13 2 minutes ago

File1	Sample file	5 minutes ago
README.md	Initial commit	2 months ago
samplecode.sh	Create samplecode.sh	a day ago

GitHub Repository (11)

Fork a Repository

- On a project on GitHub, fork the repository, make the likeable changes, and release the revised project as a new repository.
- The updates made to the original repository which was forked can easily be added to the current fork.
- In GitHub, by accessing the following link the octocat Spoon-knife repository can be accessed: https://github.com/octocat/Spoon-Knife
- Click **Fork** on the top right corner to fork a copy into the account as shown in the following screenshot:



This repo is for demonstration purposes only.

3 commits
3 branches
0 releases
1 contributor

Topic List

- GitHub Repository
- **Git Bash Commands**
- Synchronizing Local and Remote Repository
- Exercise 3.1: Working in GitHub
- Exercise 3.2: Using Git Basic Commands

Git Bash Commands (1)

Introduction

Following are the examples of Git Bash commands:

- **Initialize a local repository**
- **Clone - git clone <git-url>**
- **Checkout** – git checkout –b <new branch> (creates and switches to a new branch)
- **Add files** - git add . (this adds all files in directory)
- **Commit** - git commit -m “Added new search feature in script”
- **Tag** - name & save different versions of code
- **Pull** - git pull
- **Push** - git push

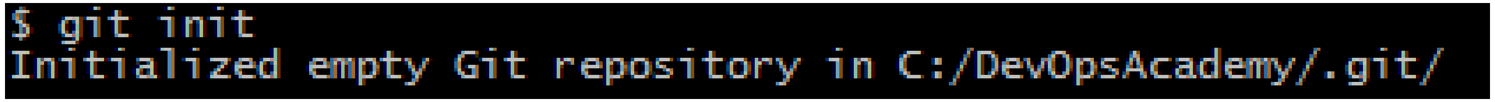
Git Bash Commands (2)

Initialize a Local Repository

To initialize the Repository, run the following command:

```
$ git init
```

The following screen capture shows the result for initialized empty Git Repository:



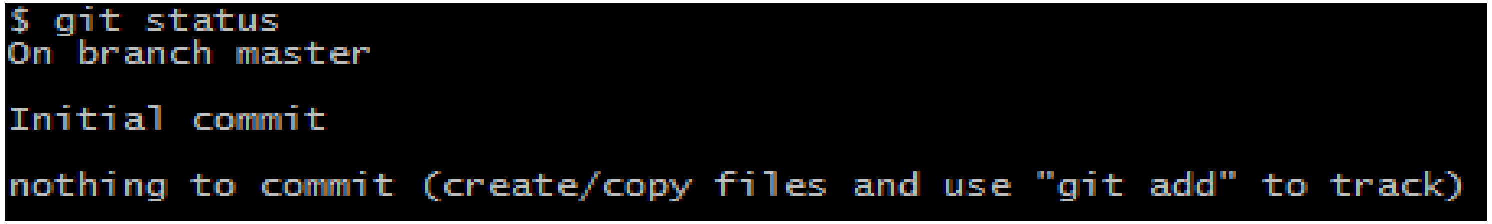
```
$ git init
Initialized empty Git repository in C:/DevOpsAcademy/.git/
```

Check the Status

Once the repository is initialized, try Git status command.

```
$ git status
```

The following screen capture shows the Status command results:



```
$ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

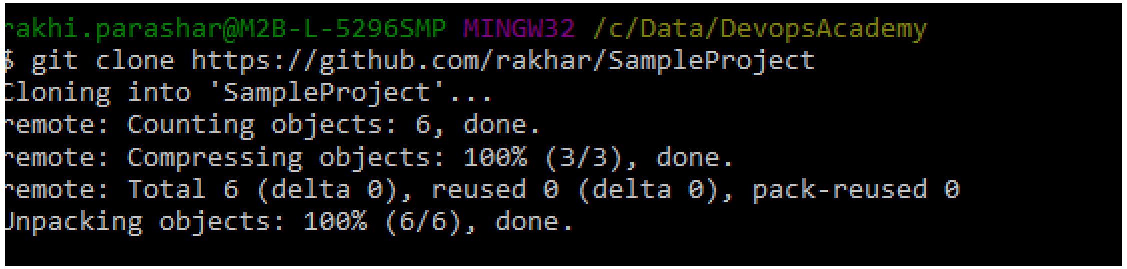
Git Bash Commands (3)

Clone the repository

Execute the following command in order to clone the repository into local machine which we created in github:

```
$ git clone https://github.com/rakhar/SampleProject or
$ git clone https://github.com/Accenture/adop-doa-materials.git
```

Following screen capture shows the result of using this command:



```
akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy
$ git clone https://github.com/rakhar/SampleProject
Cloning into 'SampleProject'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
```

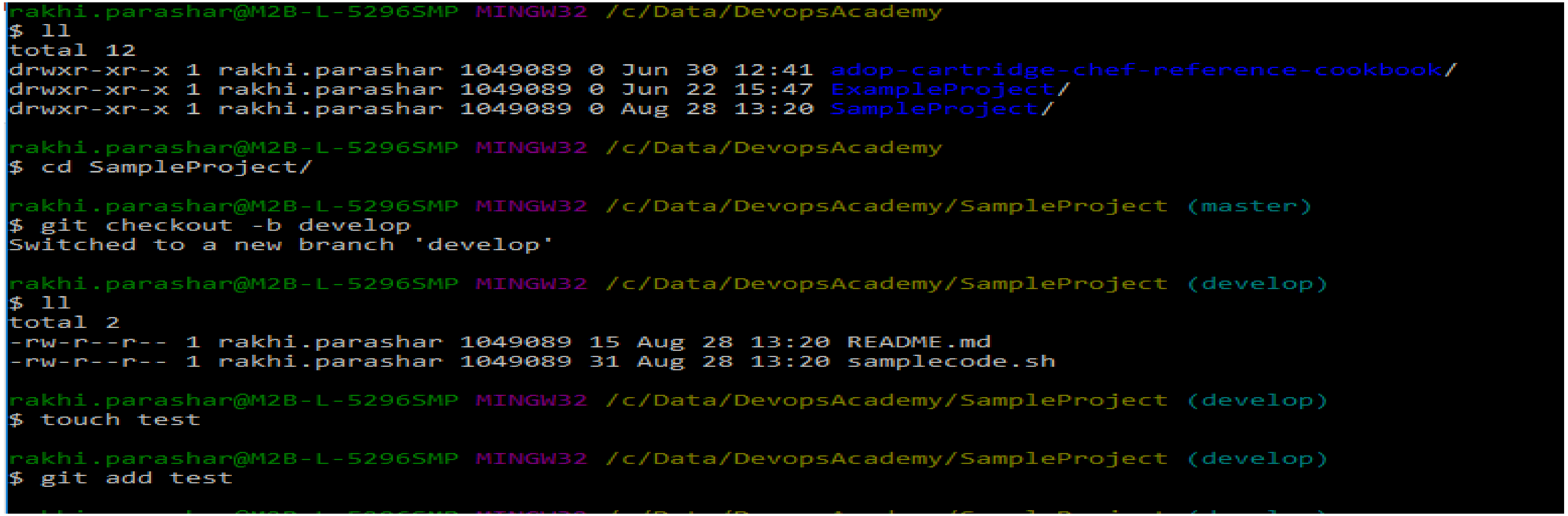
Git Bash Commands (4)

Adding File to a Repository

Create a text file in Git initialized folder and save the changes to the staging with the following command:

```
$ git add <filename>
```

Following screen capture shows the result of using this command:



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy
$ ll
total 12
drwxr-xr-x 1 rakhi.parashar 1049089 0 Jun 30 12:41 adop-cartridge-chef-reference-cookbook/
drwxr-xr-x 1 rakhi.parashar 1049089 0 Jun 22 15:47 ExampleProject/
drwxr-xr-x 1 rakhi.parashar 1049089 0 Aug 28 13:20 SampleProject/

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy
$ cd SampleProject/

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (master)
$ git checkout -b develop
Switched to a new branch 'develop'

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ ll
total 2
-rw-r--r-- 1 rakhi.parashar 1049089 15 Aug 28 13:20 README.md
-rw-r--r-- 1 rakhi.parashar 1049089 31 Aug 28 13:20 samplecode.sh

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ touch test

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git add test
```

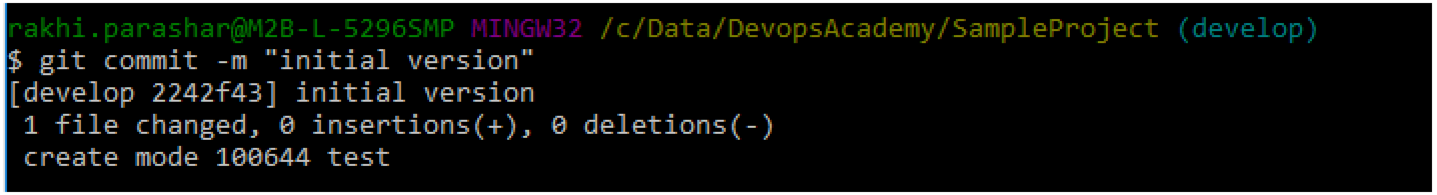
Git Bash Commands (5)

Commit the Changes to the Repository

To commit the changes to the Git repository, execute the following command:

```
$ git commit –m “comments”
```

Following screen capture shows the result of using this command:



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git commit -m "initial version"
[develop 2242f43] initial version
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test
```

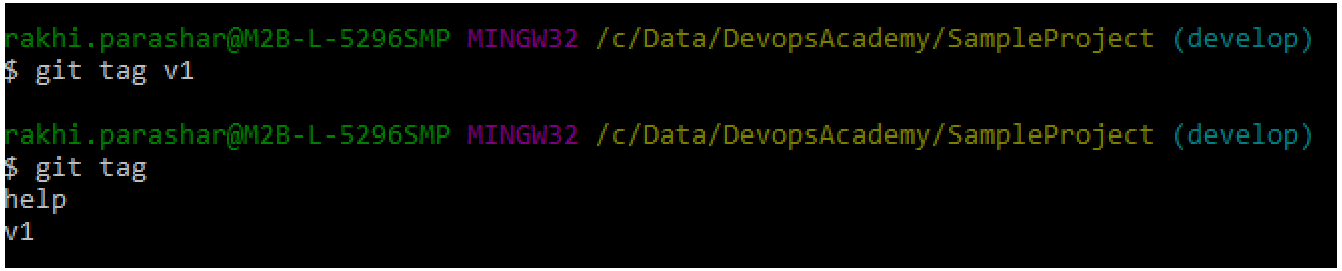
Git Bash Commands (6)

Tag a Repository

Following are the list of commands used to tag a repository:

Commands	Usage
$git tag v1	Assign a tag to the current version of the file.
$git checkout v1^	Assign a tag to the previous version, instead of mentioning the hash while checking out.
$git tag v0	Name the previous version as v0.
$git tag	View all the tags.
$git show v1	View a particular tagged version of a file.

Example



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git tag v1

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git tag
help
v1
```

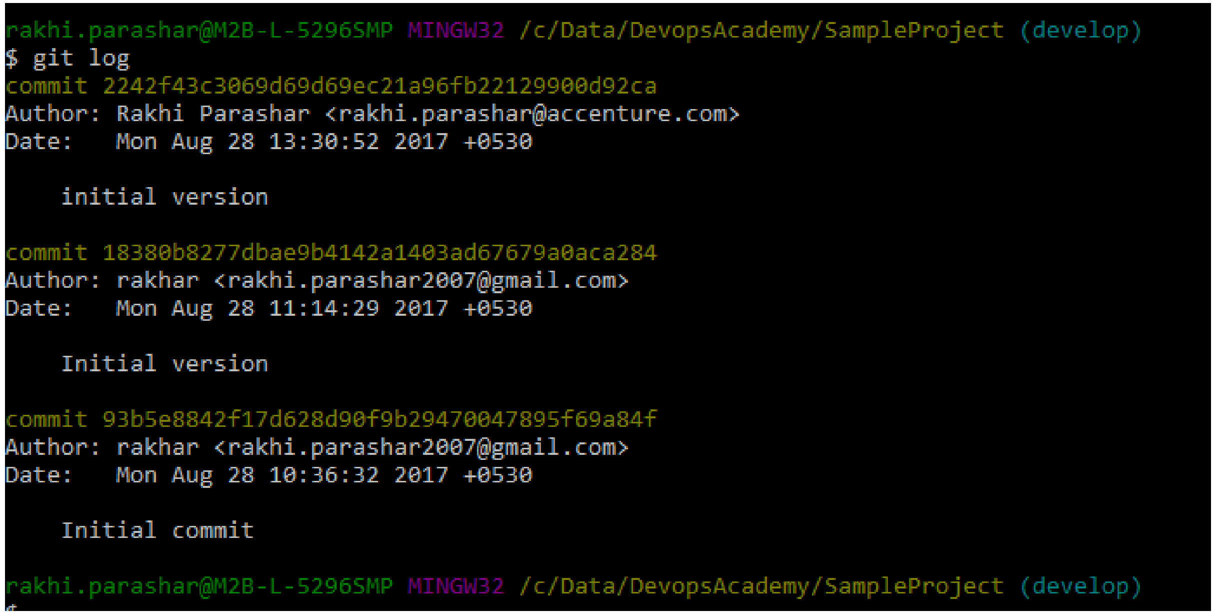
Git Bash Commands (7)

log Command

To log into the Git repository, execute the following command and see the result:

```
$ git log
```

Following screen capture shows the result of using this command:



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git log
commit 2242f43c3069d69d69ec21a96fb22129900d92ca
Author: Rakhi Parashar <rakhi.parashar@accenture.com>
Date:   Mon Aug 28 13:30:52 2017 +0530

    initial version

commit 18380b8277dbae9b4142a1403ad67679a0aca284
Author: rakhar <rakhi.parashar2007@gmail.com>
Date:   Mon Aug 28 11:14:29 2017 +0530

    Initial version

commit 93b5e8842f17d628d90f9b29470047895f69a84f
Author: rakhar <rakhi.parashar2007@gmail.com>
Date:   Mon Aug 28 10:36:32 2017 +0530

    Initial commit

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$
```

Topic List

- GitHub Repository
- Git Bash Commands
- **Synchronizing Local and Remote Repository**
- Exercise 3.1: Working in GitHub
- Exercise 3.2: Using Git Basic Commands

Synchronizing Local & Remote Repository (1)

Setting the Remote Repository

Following are the step to setup the remote repository:

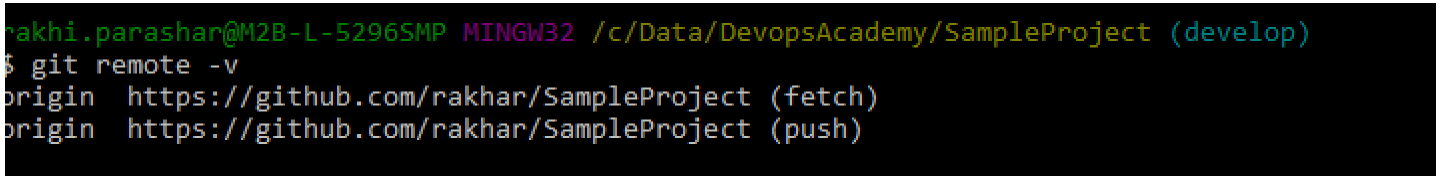
1. Go to **GITHUB** and create an empty repository.
2. Set the remote repository in Git bash with the following command:
3. Add origin https://github.com/<githubuser>/<remotegitreponame> to the following command

```
$ git remote
```

This will set the remote repository

To change the mapping, the following command can be executed

```
$ git remote –v
```



```
akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git remote -v
origin  https://github.com/rakhar/SampleProject (fetch)
origin  https://github.com/rakhar/SampleProject (push)
```

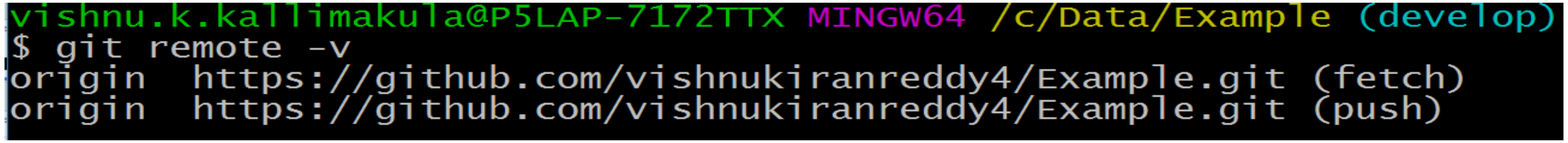
The mentioned command will help to view the existing remotes.

Synchronizing Local & Remote Repository (2)

Setting the Remote repository

- The following command will help you to view the existing remotes :

  $ git remote –v



```
vishnu.k.kallimakula@P5LAP-7172TTX MINGW64 /c/Data/Example (develop)
$ git remote -v
origin  https://github.com/vishnukiranreddy4/Example.git (fetch)
origin  https://github.com/vishnukiranreddy4/Example.git (push)
```

- Use the following command to rename an existing remote repository:

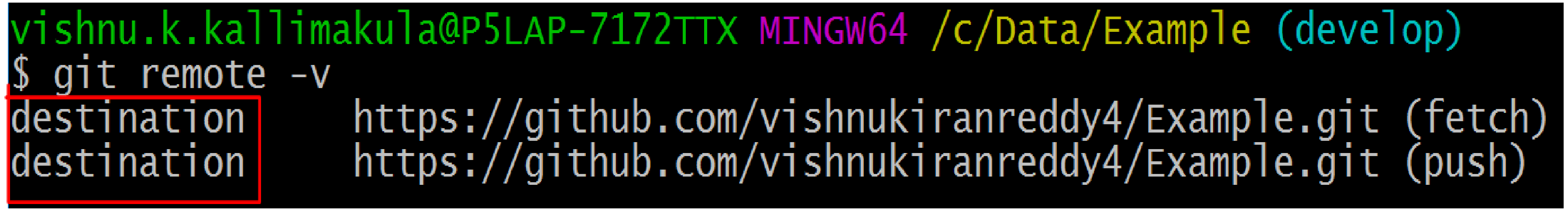
  $ git remote rename origin destination

Note: Now the existing remote repository is renamed to destination. You can check by using the following command.

$ git remote –v

Synchronizing Local & Remote Repository (3)

Setting the Remote repository



```
vishnu.k.kallimakula@P5LAP-7172TTX MINGW64 /c/Data/Example (develop)
$ git remote -v
destination     https://github.com/vishnukiranreddy4/Example.git (fetch)
destination     https://github.com/vishnukiranreddy4/Example.git (push)
```

Now, we can set a remote repository without cloning the repository.

Following are the steps to setup the remote repository:

1. Go to **GitHub** and create an empty repository.
2. Set the remote repository in Git bash with the following command:

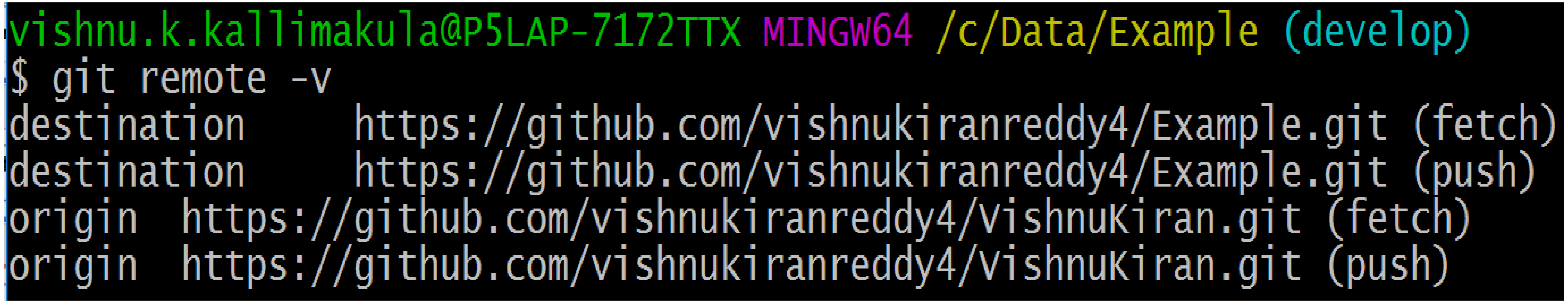
```
$ git remote add origin https://github.com/vishnukiranreddy4/VishnuKiran.git
```

Synchronizing Local & Remote Repository (4)

Setting the Remote repository

Now you can see the existing remotes by using the following command:

```
$ git remote –v
```



```
vishnu.k.kallimakula@P5LAP-7172TTX MINGW64 /c/Data/Example (develop)
$ git remote -v
destination     https://github.com/vishnukiranreddy4/Example.git (fetch)
destination     https://github.com/vishnukiranreddy4/Example.git (push)
origin  https://github.com/vishnukiranreddy4/VishnuKiran.git (fetch)
origin  https://github.com/vishnukiranreddy4/VishnuKiran.git (push)
```

Synchronizing Local & Remote Repository (5)

Renaming the Remote Repository

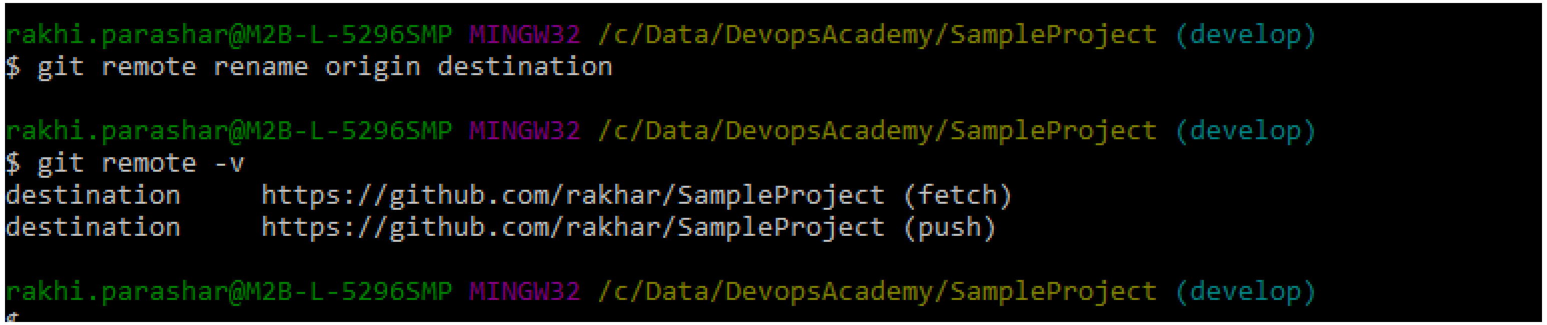
Use the following command to rename an existing remote:

```
$ git remote rename origin destination
```

The two arguments of this command are:

- An existing remote name. Example, origin.
- A new name for the remote. Example, destination.

Following screen capture shows the result of using this command:



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git remote rename origin destination

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git remote -v
destination     https://github.com/rakhar/SampleProject (fetch)
destination     https://github.com/rakhar/SampleProject (push)

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
```

Synchronizing Local & Remote Repository (6)

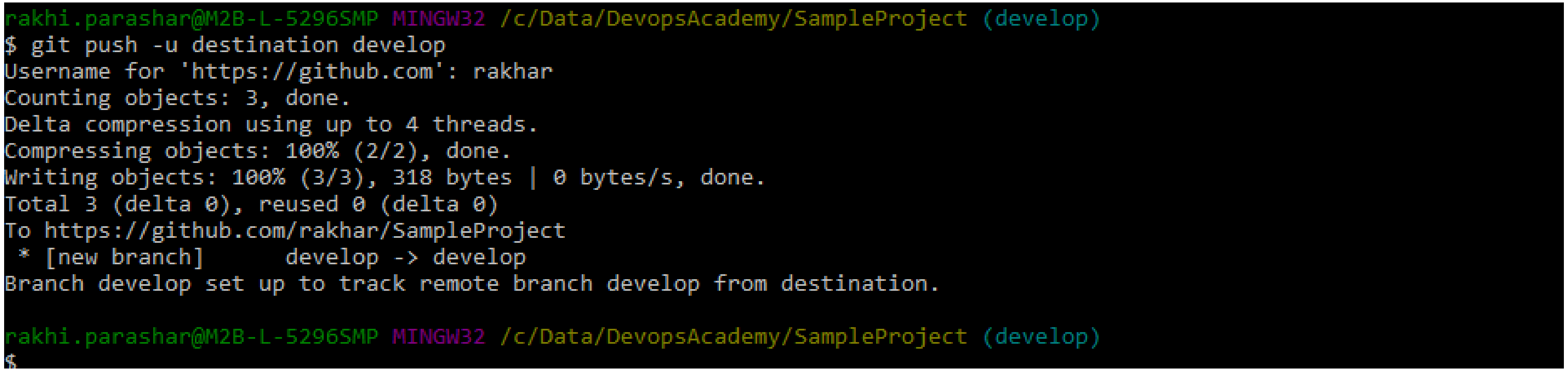
push Command

Run the following command to push commit on the repository:

```
$ git push –u origin master
```

-u option: It makes Git to remember the parameter in order to execute just ‘GIT Push’. The next time Git will know where it needs to push.

Following screen capture shows the result of using this command:



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git push -u destination develop
Username for 'https://github.com': rakhar
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/rakhar/SampleProject
 * [new branch]      develop -> develop
Branch develop set up to track remote branch develop from destination.

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$
```

Synchronizing Local & Remote Repository (7)

pull Command

To pull the changes to local repository, run the following command.

```
$ git pull <repository>
```

Note: If there are changes updated to the remote repository (could be by a different user), the changes can be pulled into/updated to the local repository using the Git Pull command.

Topic List

GitHub Repository

Git Bash Commands

Synchronizing Local and Remote Repository

Exercise 3.1: Working in GitHub

Exercise 3.2: Using Git Basic Commands

Exercise 3.1: Working in GitHub

Scenario

Create an account in Github and create a smaple project where you need to:

- Create a GitHub Account
- Create repository
- Preview the created repository
- Add files to the project
- View the new files add or edit comments

Topic List

GitHub Repository

Git Bash Commands

Synchronizing Local and Remote Repository

Exercise 3.1: Working in GitHub

Exercise 3.2: Using Git Basic Commands

Exercise 3.2: Using Git Basic Commands

Scenario

Perform following tasks using the basic Git commands:

- Cloning the repository created in GitHub into local machine
- Adding file to a repository
- Committing the changes to the Repository
- Using GIT Push Command
- Tagging a repository
- Viewing the log details of the repository
- Synchronizing the local & remote repository
- Using pull command to update the changes to the remote repository

Knowledge Check

Select the right answer

In which of the following, Master is the default branch of a repository?

a) Commit in GITHub

b) Pull Requests

[✓ c) Branching in Git Hub]

d) Fork a Repository

Knowledge Check

Select the right answer

Which is not an example of Git Bash commands?

a) Clone

b) Commit

c) Tag

[✓ d) Branch]

Module Summary

Now, you should be able to:

- Describe Git Hub Repository.
- Describe techniques like Branching, Commiting, Pull Requests, and Merging with respect to GITHub Repository.
- Apply various Git Bash commands.
- Use the commands to synchronize the local and remote repository.

Thank You