

Experiment 3

Aim: To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet.

Theory:

- **Introduction to GIT:**

Git is the free and open source distributed version control system that's responsible for everything GitHub related that happens locally on your computer.

Git is a version control system used for tracking changes in computer files. It is generally used for source code management in software development. Git is used to tracking changes in the source code. The distributed version control tool is used for source code management.

- **GIT Cheat-sheet:**

Cheat-sheet features the most important and commonly used Git Commands.

Some Git commands:

- Git Config command
- Git init command
- Git add command
- Git commit command
- Git status command
- Git push Command
- Git pull command
- Git log command
- Git remote command

Output:

GIT operations on local repositories and Remote:


Creating a Repository:

Create a new repository


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

Owner *

Repository name *


 MrunalVaidya0715

 /


First-Repo 

Great repository names are short and memorable. Need inspiration? How about **super-duper-octo-guacamole**?

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.

Launching an Instance:

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Number of instances [Info](#)

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)

ami-076e3a557efe1aa9c

Virtual server type (instance type)



t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

 Free tier: In your first year includes 750 

Cancel

Launch instance


Creating Key Pair for Secure Login:


▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select



 [Create new key pair](#)

Name: Mrunal Vaidya
Roll : 68

XIE ID: 202003060
Batch: C

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

awskeyy

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA

RSA encrypted private and public key pair

☐ ED25519

ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

☐ .pem

For use with OpenSSH

☒ .ppk

For use with PuTTY

Cancel

Create key pair

Instance Launched:

The screenshot shows the AWS Management Console 'Instances' page. A table lists one instance: 'AWS-Cloud-Mrunal' with ID 'i-06c918a8e10434a77', state 'Running', and type 't2.micro'. A context menu is open for this instance, showing options like 'Stop instance', 'Start instance', 'Reboot instance', 'Hibernate instance', and 'Terminate instance'. Below the table, the 'Details' tab for instance 'i-06c918a8e10434a77 (usaServer)' is selected, showing fields for Instance ID, Public IPv4 address (44.207.1.25), and Private IP address (172.31.0.17).

Name	Instance ID	Instance state	Instance type	Stat
AWS-Cloud-Mrunal	i-06c918a8e10434a77	Running	t2.micro	-

Instance: i-06c918a8e10434a77 (usaServer)

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
Instance summary Info						
Instance ID	Public IPv4 address		Private IP address			
i-06c918a8e10434a77 (usaServer)	44.207.1.25 open address		172.31.0.17			

Instance Launch Successfully:

The screenshot shows a success message in the AWS Management Console: 'Success Successfully initiated launch of instance (i-0aa46f5e762248a0a)'. Below the message is a link to 'Launch log'.

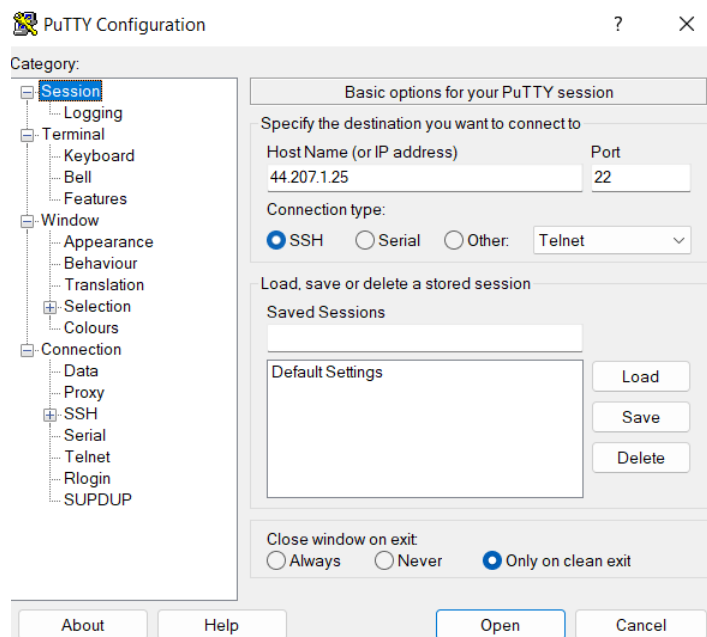
Success
Successfully initiated launch of instance (i-0aa46f5e762248a0a)

► Launch log

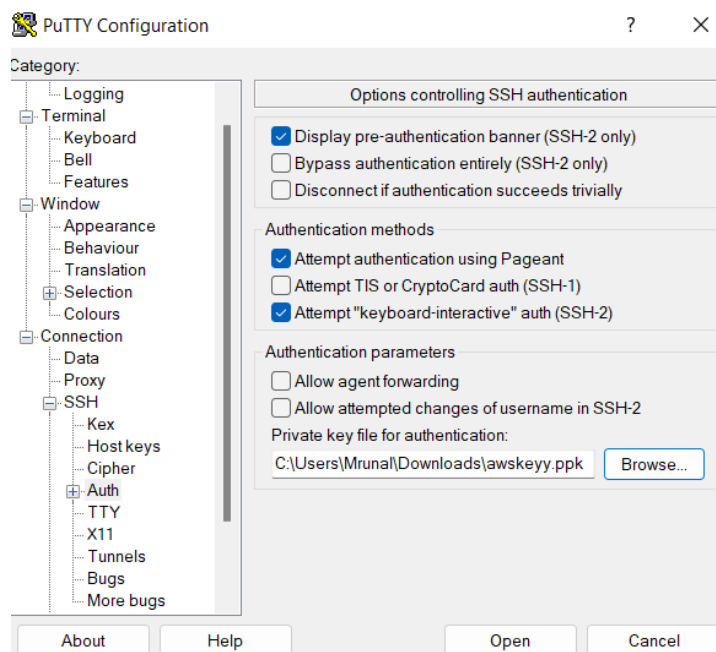
(Using Putty for login) Using Public IPv4 address of created Instance:

Name: Mrunal Vaidya
Roll : 68

XIE ID: 202003060
Batch: C



Using key pair for Secure login:



Installing Git for User 1:

```
login as: ec2-user
Authenticating with public key "awskey" from agent

  _ | _ | _ )
  _ | ( _ | /   Amazon Linux 2 AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-47-159 ~]$ sudo su
[root@ip-172-31-47-159 ec2-user]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update

[root@ip-172-31-47-159 ec2-user]# yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.37.1-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-Git = 2.37.1-1.amzn2.0.1 for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.37.1-1.amzn2.0.1 for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core = 2.37.1-1.amzn2.0.1 for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git::I18N) for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git) for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.37.1-1.amzn2.0.1 will be installed
--> Package git-core-doc.noarch 0:2.37.1-1.amzn2.0.1 will be installed
--> Package perl-Git.noarch 0:2.37.1-1.amzn2.0.1 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.37.1-1.amzn2.0.1.noarch
--> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

Checking Git Version:

```
[root@ip-172-31-47-159 ec2-user]# git --version
git version 2.37.1
```

Git Configuration:

```
[root@ip-172-31-47-159 ec2-user]# git config --global user.name "Mrunal"
[root@ip-172-31-47-159 ec2-user]# git config --global user.email "mrunalvaidya4755@gmail.com"
[root@ip-172-31-47-159 ec2-user]# git config --list
user.name=Mrunal
user.email=mrunalvaidya4755@gmail.com
```

```
[root@ip-172-31-47-159 ec2-user]# which git
/bin/git
```

Making directory:

```
[root@ip-172-31-47-159 ec2-user]# mkdir Mrunal
[root@ip-172-31-47-159 ec2-user]# cd Mrunal
[root@ip-172-31-47-159 Mrunal]# ls
[root@ip-172-31-47-159 Mrunal]# cat >> F1
This is Mrunal Vaidya from batch C
[root@ip-172-31-47-159 Mrunal]# ls
F1
```

Starting a project and creating a local repository:

```
[root@ip-172-31-47-159 Mrunal]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ec2-user/Mrunal/.git/
```

Displaying the state of the working directory and the staging area:

```
[root@ip-172-31-47-159 Mrunal]# git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    F1

nothing added to commit but untracked files present (use "git add" to track)
[root@ip-172-31-47-159 Mrunal]# git add .
[root@ip-172-31-47-159 Mrunal]# git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   F1
```

Working with snapshots and the staging area:

```
[root@ip-172-31-47-159 Mrunal]# git commit -m "First Update"
[master (root-commit) 3f2b588] First Update
 1 file changed, 1 insertion(+)
 create mode 100644 F1
```

Displaying the most recent commits and the status of the head:

```
[root@ip-172-31-47-159 Mrunal]# git log
commit 3f2b5885349eaea24a682a984ef107927cac8ffb (HEAD -> master)
Author: Mrunal <mrunalvaidya4755@gmail.com>
Date:   Fri Aug 5 04:18:25 2022 +0000

    First Update
```

Show any object in Git in human-readable format:

Name: Mrunal Vaidya
Roll : 68

XIE ID: 202003060
Batch: C

```
[root@ip-172-31-47-159 Mrunal]# git show 3f2b5885349eaea24a682a984ef107927cac8ffb
commit 3f2b5885349eaea24a682a984ef107927cac8ffb (HEAD -> master)
Author: Mrunal <mrunalvaidya4755@gmail.com>
Date:   Fri Aug 5 04:18:25 2022 +0000

    First Update

diff --git a/F1 b/F1
new file mode 100644
index 0000000..a8cfe44
--- /dev/null
+++ b/F1
@@ -0,0 +1 @@
+This is Mrunal Vaidya from batch C
```

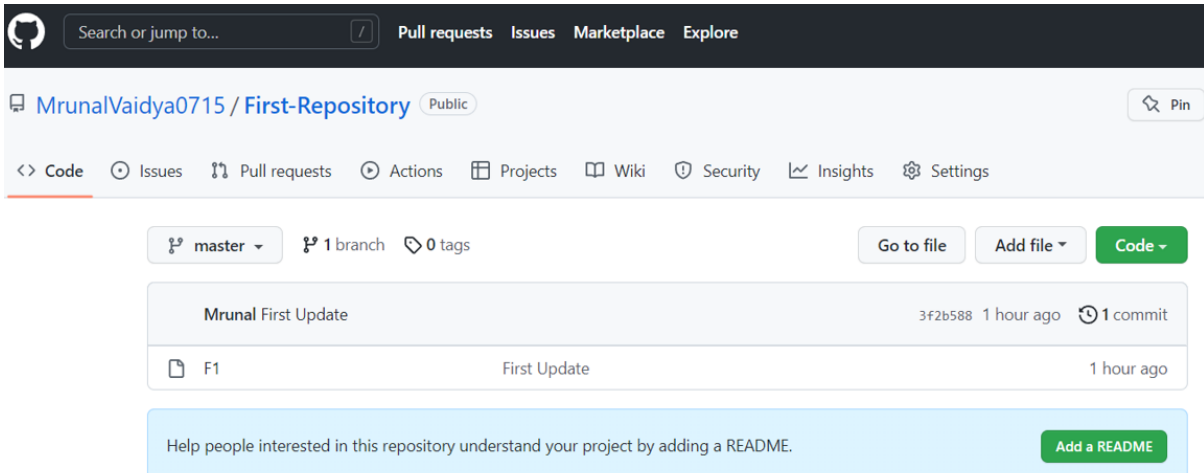
Retrieving updates from another repository and updating local repositories:

```
Mrunal]# git remote add origin https://github.com/MrunalVaidya0715/First-Repository.git
```

Pushing Updates(Transmit local branch commits to the remote repository branch):

```
[root@ip-172-31-47-159 Mrunal]# git push -u origin master
Username for 'https://github.com': MrunalVaidya0715
Password for 'https://MrunalVaidya0715@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 245 bytes | 245.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MrunalVaidya0715/First-Repository.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

First Update:



The screenshot shows the GitHub interface for a repository named 'First-Repository' by user 'MrunalVaidya0715'. The repository is public. The main branch is 'master', which has 1 branch and 0 tags. The commit history shows a single commit titled 'Mrunal First Update' by '3f2b588' made '1 hour ago'. The file 'F1' is listed with the commit message 'First Update' and a timestamp of '1 hour ago'. At the bottom, there is a prompt to 'Add a README' to help people understand the project.

Similarly, adding User 2 and performing Git Operations:

```
[root@ip-172-31-86-87 ec2-user]# git config --global user.name "Mrunalusa"
[root@ip-172-31-86-87 ec2-user]# git config --global user.email "mrunal@email.com"
[root@ip-172-31-86-87 ec2-user]# git config --list
user.name=Mrunalusa
user.email=mrunal@email.com

[root@ip-172-31-86-87 newMrunal]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ec2-user/newMrunal/.git/
```

Fetching and merging any commits from the tracking remote branch:

```
[root@ip-172-31-86-87 newMrunal]# git remote add origin https://github.com/MrunalVaidya0715/First-Repository.git
[root@ip-172-31-86-87 newMrunal]# git pull origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 225 bytes | 225.00 KiB/s, done.
From https://github.com/MrunalVaidya0715/First-Repository
 * branch            master       -> FETCH_HEAD
 * [new branch]      master       -> origin/master
[root@ip-172-31-86-87 newMrunal]# ls
F1 F2
[root@ip-172-31-86-87 newMrunal]# cat F1
This is Mrunal Vaidya from batch C
[root@ip-172-31-86-87 newMrunal]# cat>>F1
This is Changes to 1st file
```

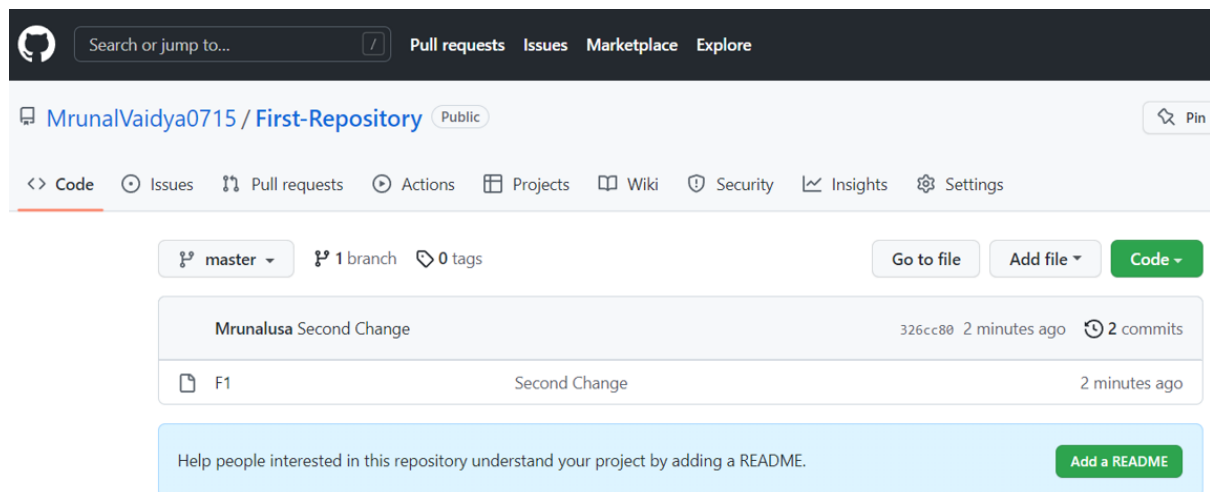
Committing your staged content

```
[root@ip-172-31-86-87 newMrunal]# git commit -m "Second Change"
[master 326cc80] Second Change
 1 file changed, 1 insertion(+)
[root@ip-172-31-86-87 newMrunal]# git push -u origin master
Username for 'https://github.com': MrunalVaidya0715
Password for 'https://MrunalVaidya0715@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MrunalVaidya0715/First-Repository.git
 3f2b588..326cc80  master -> master
branch 'master' set up to track 'origin/master'.
```

Second Update from user 2:

Name: Mrunal Vaidya
Roll : 68

XIE ID: 202003060
Batch: C



Conclusion:

In the above Experiment, we have learned to perform various git operations on Local and Remote repositories using GIT cheat-Sheet. And hence, with this experiment we have achieved the Lab Outcome two (LO2).

POs Achieved: PO1, PO5, PO12.