Day 8 Task: Basic Git & GitHub for DevOps Engineers.

What is Git?

Git is a version control system that allows you to track changes to files and coordinate work on those files among multiple people. It is commonly used for software development, but it can be used to track changes to any set of files.

With Git, you can keep a record of who made changes to what part of a file, and you can revert back to earlier versions of the file if needed. Git also makes it easy to collaborate with others, as you can share changes and merge the changes made by different people into a single version of a file.

Git & GitHub

- ✓ Code Management
- ✓ Version Control System
- ✓ Collaboration

What is Github?

GitHub is a web-based platform that provides hosting for version control using Git. It is a subsidiary of Microsoft, and it offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. GitHub is a very popular platform for developers to share and collaborate on projects, and it is also used for hosting open-source projects.

What is Version Control? How many types of version controls we have?

Version control is a system that tracks changes to a file or set of files over time so that you can recall specific versions later. It allows you to revert files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more.

There are two main types of version control systems: centralized version control systems and distributed version control systems.

1. A centralized version control system (CVCS) uses a central server to store all the versions of a project's files. Developers "check out" files from the central server, make changes, and then "check in" the updated files. Examples of CVCS include Subversion and Perforce.
2. A distributed version control system (DVCS) allows developers to "clone" an entire repository, including the entire version history of the project. This means that they have a complete local copy of the repository, including all branches and past versions. Developers can work independently and then later merge their changes back into the main repository. Examples of DVCS include Git, Mercurial, and Darcs.

Why we use distributed version control over centralized version control?

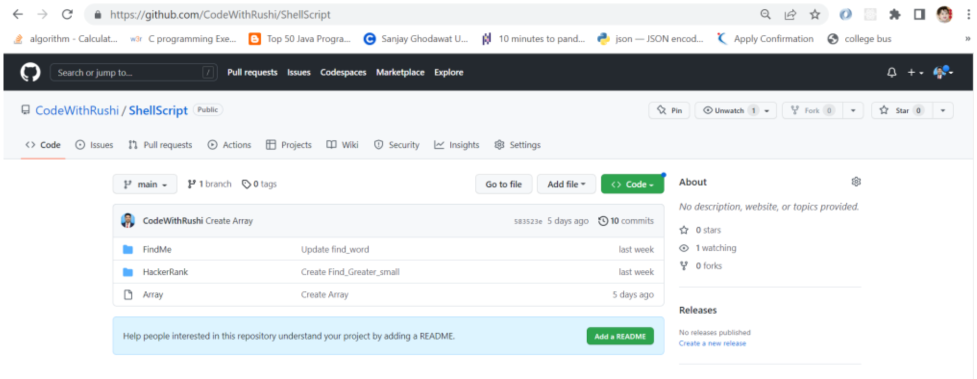
3. Better collaboration: In a DVCS, every developer has a full copy of the repository, including the entire history of all changes. This makes it easier for developers to work together, as they don't have to constantly communicate with a central server to commit their changes or to see the changes made by others.
4. Improved speed: Because developers have a local copy of the repository, they can commit their changes and perform other version control actions faster, as they don't have to communicate with a central server.
5. Greater flexibility: With a DVCS, developers can work offline and commit their changes later when they do have an internet connection. They can also choose to share their changes with only a subset of the team, rather than pushing all of their changes to a central server.
6. Enhanced security: In a DVCS, the repository history is stored on multiple servers and computers, which makes it more resistant to data loss. If the central server in a CVCS goes down or the repository becomes corrupted, it can be difficult to recover the lost data.

Overall, the decentralized nature of a DVCS allows for greater collaboration, flexibility, and security, making it a popular choice for many teams.

Tasks:-

1. Create a new repository on GitHub and clone it to your local machine

GitHub Repo:-



CodeWithRushi Create Array 583523e 5 days ago 10 commits

FindMe	Update find_word	last week
HackerRank	Create Find_Greater_small	last week
Array	Create Array	5 days ago

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

0 stars

1 watching

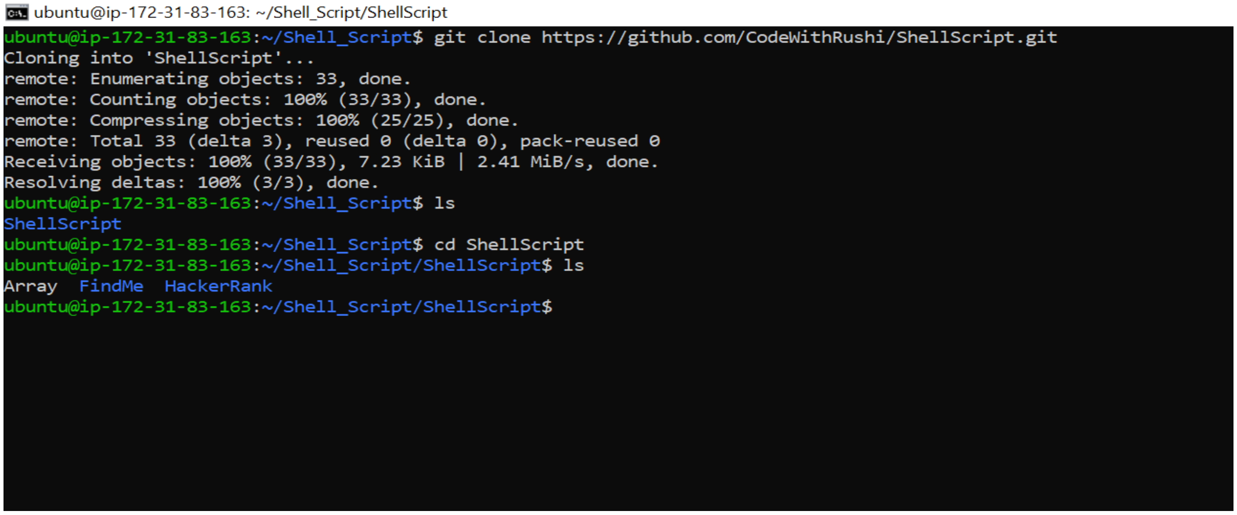
0 forks

Releases

No releases published

Create a new release

Local after cloning :-



```
ubuntu@ip-172-31-83-163: ~/Shell_Script/ShellScript
ubuntu@ip-172-31-83-163:~/Shell_Script$ git clone https://github.com/CodeWithRushi/ShellScript.git
Cloning into 'ShellScript'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 33 (delta 3), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 7.23 KiB | 2.41 MiB/s, done.
Resolving deltas: 100% (3/3), done.
ubuntu@ip-172-31-83-163:~/Shell_Script$ ls
ShellScript
ubuntu@ip-172-31-83-163:~/Shell_Script$ cd ShellScript
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript$ ls
Array  FindMe  HackerRank
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript$
```

2. Make some changes to a file in the repository and commit them to the repository using Git.

    i. Added a new File

    ii. Execute the below command to check the git current status

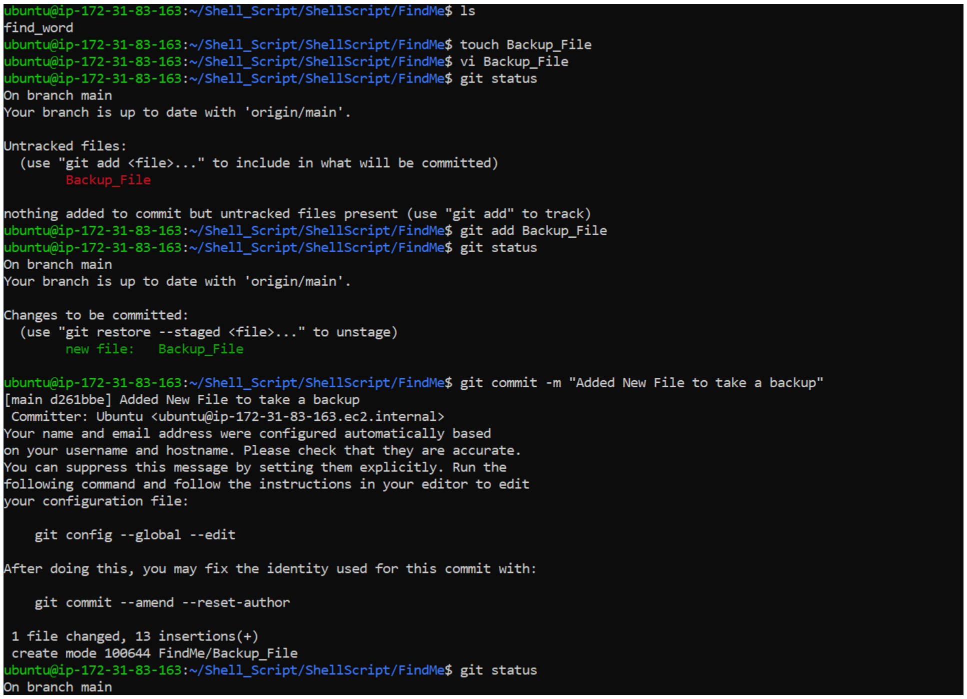
git status

iii. After use git add command to staging the file

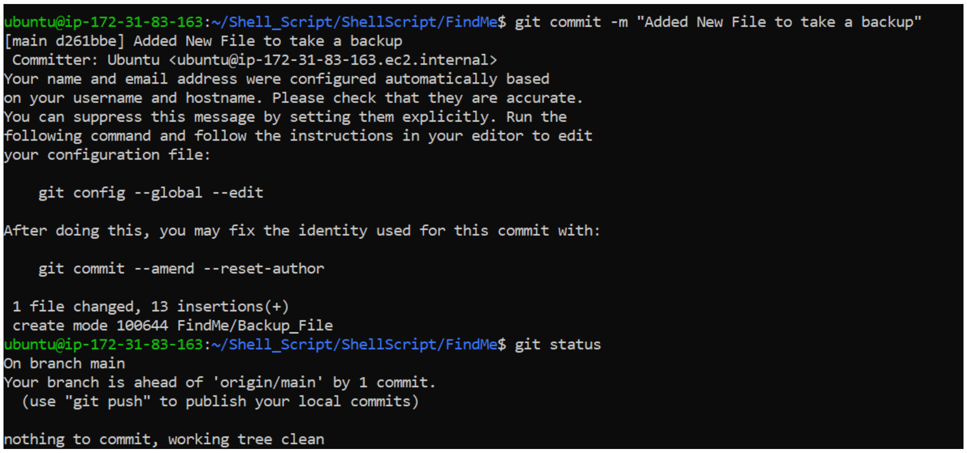
git add Backup_file   ------(file_name)

iv. Use git commit command to save your changes to the local repository and add meaningful message.

git commit –m “New File Added to take a Backup of a file”



```
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ ls
find_word
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ touch Backup_File
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ vi Backup_File
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ git add Backup_File
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Backup_File

ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ git commit -m "Added New File to take a backup"
[main d261bbe] Added New File to take a backup
 Committer: Ubuntu <ubuntu@ip-172-31-83-163.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 13 insertions(+)
 create mode 100644 FindMe/Backup_File
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ git status
On branch main
```



```
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ git commit -m "Added New File to take a backup"
[main d261bbe] Added New File to take a backup
 Committer: Ubuntu <ubuntu@ip-172-31-83-163.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 13 insertions(+)
 create mode 100644 FindMe/Backup_File
ubuntu@ip-172-31-83-163:~/Shell_Script/ShellScript/FindMe$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

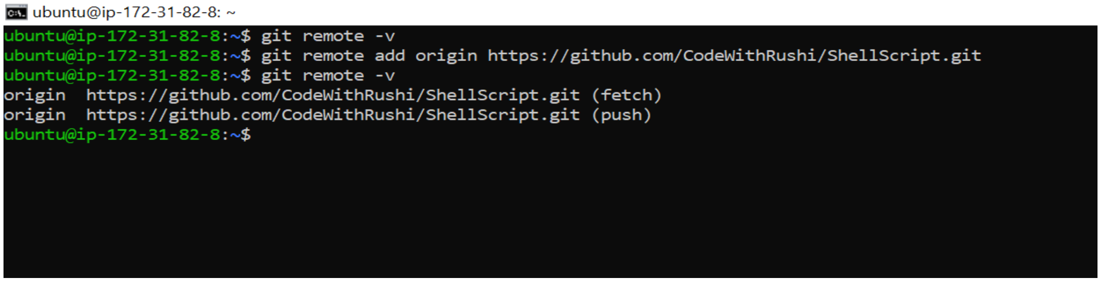
3. Push the changes back to the repository on GitHub

    i. First execute **git remote –v** command to check available remote URL

git remote -v

    ii. If no any available URL is present,then add a remote URL

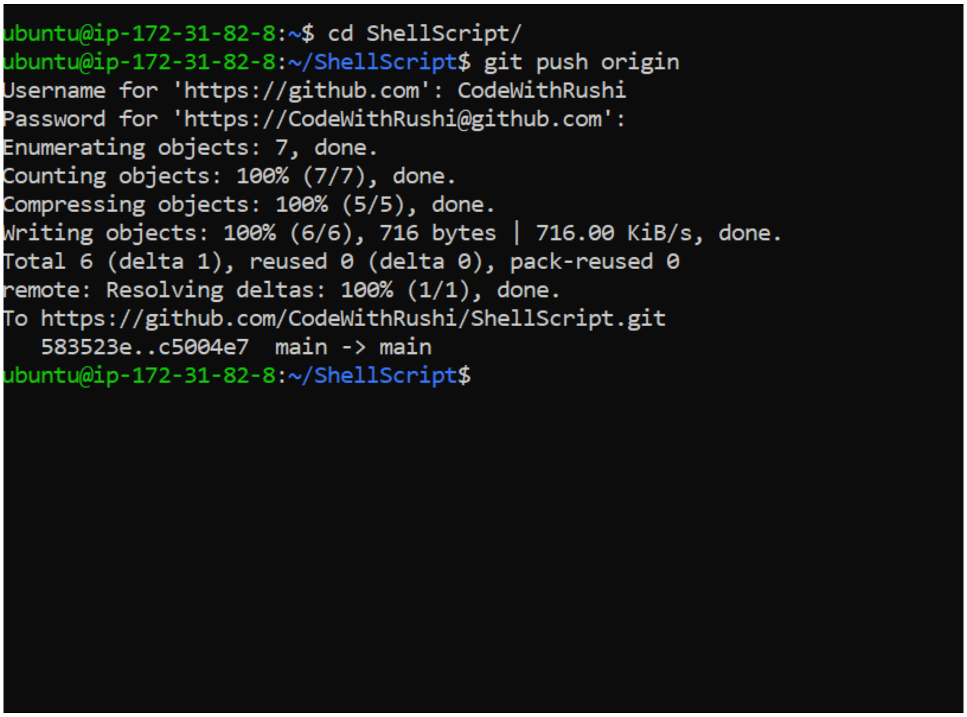
git remote add origin <your git remote url>



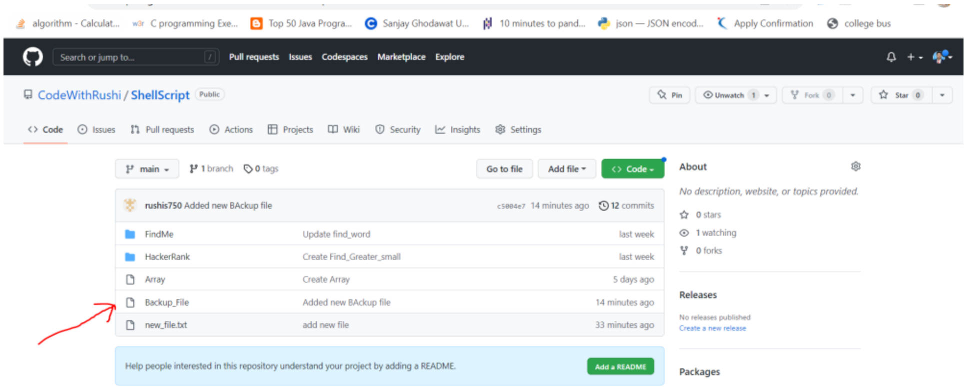
```
ubuntu@ip-172-31-82-8: ~
ubuntu@ip-172-31-82-8:~$ git remote -v
ubuntu@ip-172-31-82-8:~$ git remote add origin https://github.com/CodeWithRushi/ShellScript.git
ubuntu@ip-172-31-82-8:~$ git remote -v
origin  https://github.com/CodeWithRushi/ShellScript.git (fetch)
origin  https://github.com/CodeWithRushi/ShellScript.git (push)
ubuntu@ip-172-31-82-8:~$
```

iii. Push the files to github

git push origin



```
ubuntu@ip-172-31-82-8:~$ cd ShellScript/
ubuntu@ip-172-31-82-8:~/ShellScript$ git push origin
Username for 'https://github.com': CodeWithRushi
Password for 'https://CodeWithRushi@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 716 bytes | 716.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/CodeWithRushi/ShellScript.git
   583523e..c5004e7  main -> main
ubuntu@ip-172-31-82-8:~/ShellScript$
```



CodeWithRushi / ShellScript Public

rushis750 Added new BAckup file c5004e7 14 minutes ago 12 commits

FindMe	Update find_word	last week
HackerRank	Create Find_Greater_small	last week
Array	Create Array	5 days ago
Backup_File	Added new BAckup file	14 minutes ago
new_file.txt	add new file	33 minutes ago

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

0 stars

1 watching

0 forks

Releases

No releases published

Packages

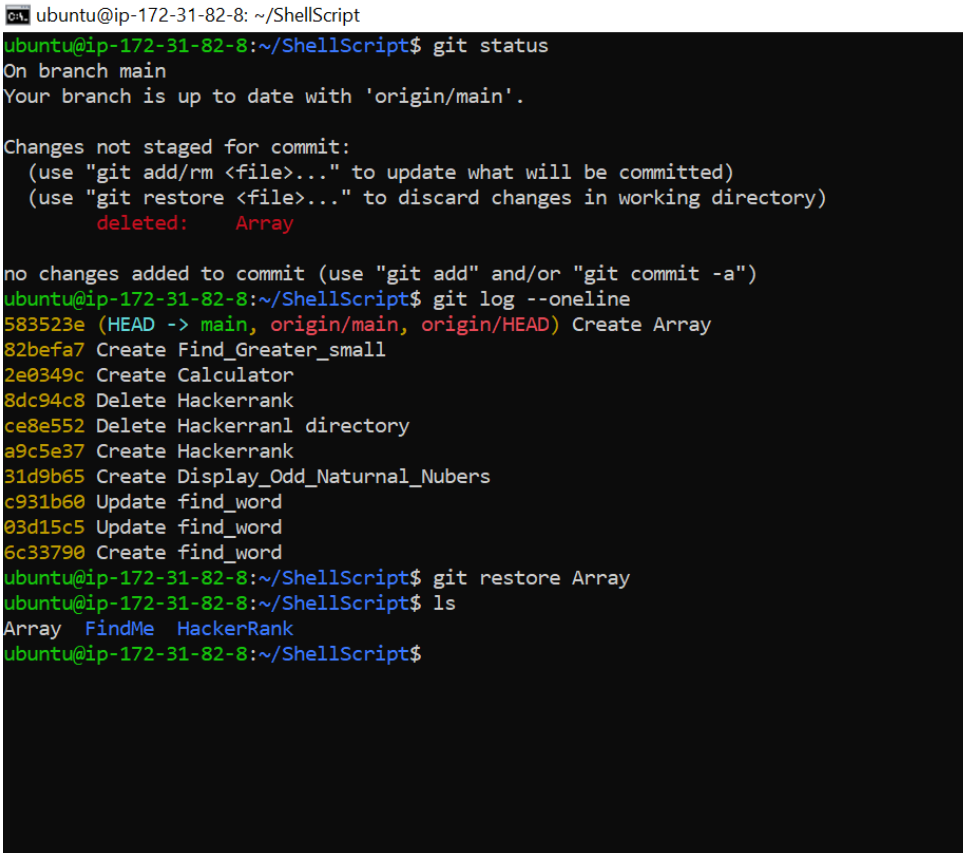
4. Delete the file and Restore the file

git restore :-

git restore is used to restore or discard the uncommitted local changes of files.

we can restore file only if the removed file is preset in a staggging state

git restore <FileName>



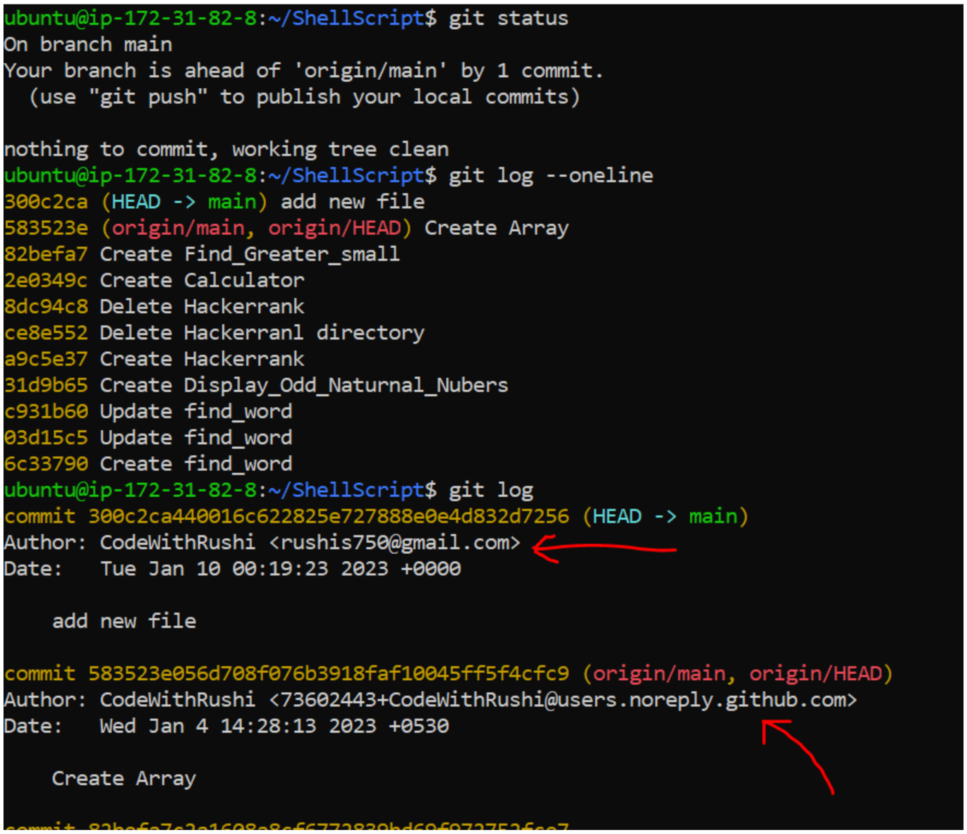
```
ubuntu@ip-172-31-82-8:~/ShellScript$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

no changes added to commit (use "git add" and/or "git commit -a")
ubuntu@ip-172-31-82-8:~/ShellScript$ git log --oneline
583523e (HEAD -> main, origin/main, origin/HEAD) Create Array
82befa7 Create Find_Greater_small
2e0349c Create Calculator
8dc94c8 Delete Hackerrank
ce8e552 Delete Hackerranl directory
a9c5e37 Create Hackerrank
31d9b65 Create Display_Odd_Naturnal_Nubers
c931b60 Update find_word
03d15c5 Update find_word
6c33790 Create find_word
ubuntu@ip-172-31-82-8:~/ShellScript$ git restore Array
ubuntu@ip-172-31-82-8:~/ShellScript$ ls
Array  FindMe  HackerRank
ubuntu@ip-172-31-82-8:~/ShellScript$
```

5. Configure the credentials on globally

The git config command is a convenience function that is used to set Git configuration values on a global or local project level.

git config –global user.name <UserName>

Git config –global user.email <UserEmail>



```
ubuntu@ip-172-31-82-8:~/ShellScript$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
ubuntu@ip-172-31-82-8:~/ShellScript$ git log --oneline
300c2ca (HEAD -> main) add new file
583523e (origin/main, origin/HEAD) Create Array
82befa7 Create Find_Greater_small
2e0349c Create Calculator
8dc94c8 Delete Hackerrank
ce8e552 Delete Hackerranl directory
a9c5e37 Create Hackerrank
31d9b65 Create Display_Odd_Naturnal_Nubers
c931b60 Update find_word
03d15c5 Update find_word
6c33790 Create find_word
ubuntu@ip-172-31-82-8:~/ShellScript$ git log
commit 300c2ca440016c622825e727888e0e4d832d7256 (HEAD -> main)
Author: CodeWithRushi <rushis750@gmail.com>
Date:   Tue Jan 10 00:19:23 2023 +0000

    add new file

commit 583523e056d708f076b3918faf10045ff5f4cfc9 (origin/main, origin/HEAD)
Author: CodeWithRushi <73602443+CodeWithRushi@users.noreply.github.com>
Date:   Wed Jan 4 14:28:13 2023 +0530

    Create Array

commit 82befa7c2a1608a8cf6772839bd69f972752fcc7
```