NAME: Jose Ashley

ROLL NO: 501826

BATCH: B-1

DATE: 27.01.2022

EXPERIMENT No : 2

AIM: Version control system using GIT.

LAB OUTCOME: Examine the different version control strategies.

THEORY:

GIT:
Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

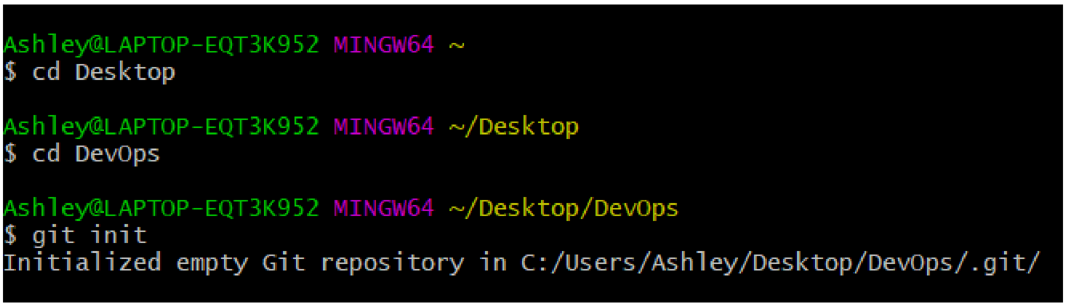
GITHUB:
GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

DIFFERENCE BETWEEN GIT AND GITHUB

GIT	**GITHUB**
Git is a software.	GitHub is a service.
Git is a command-line tool	GitHub is a graphical user interface
Git is installed locally on the system	GitHub is hosted on the web
Git is maintained by linux.	GitHub is maintained by microsoft.
Git is focused on version control and code sharing.	GitHub is focused on centralized source code hosting.
Git is a version control system to manage source code history.	GitHub is a hosting service for Git repositories.
Git was first released in 2005.	GitHub was launched in 2008.
Git has no user management feature.	GitHub has built-in user management feature.

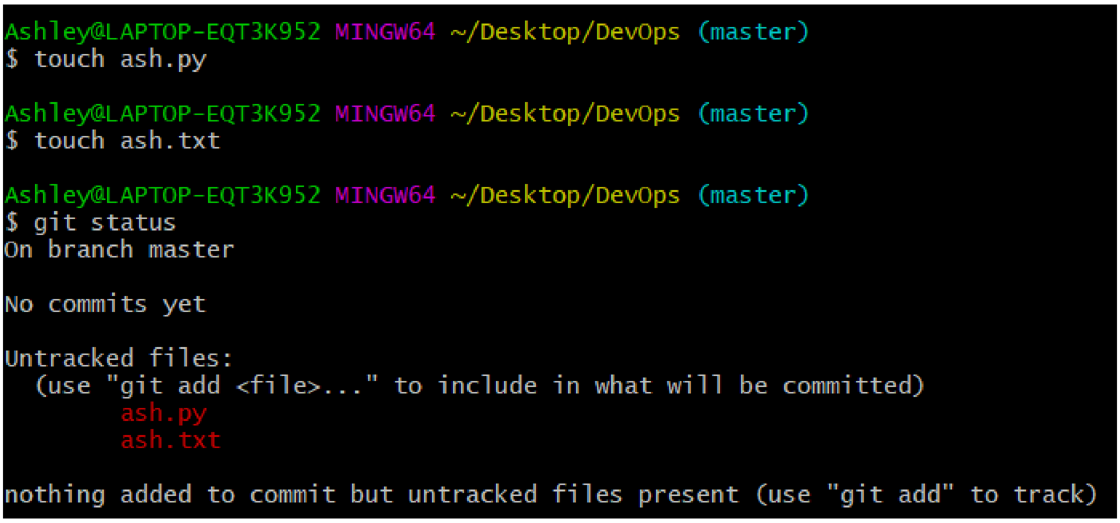
OUTPUT:

Initialize repository



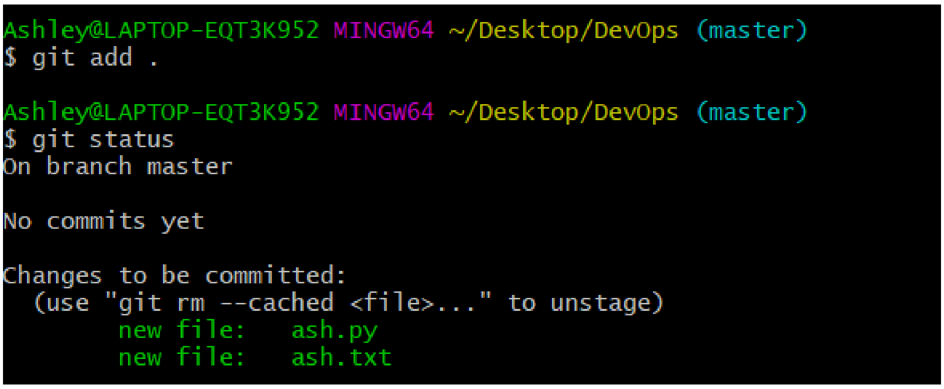
```
Ashley@LAPTOP-EQT3K952 MINGW64 ~
$ cd Desktop

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop
$ cd DevOps

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps
$ git init
Initialized empty Git repository in C:/Users/Ashley/Desktop/DevOps/.git/
```

Create files in directory



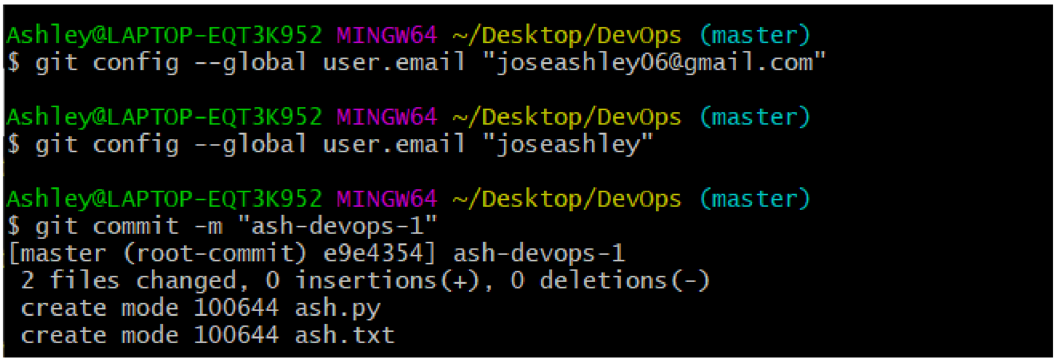
```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ touch ash.py

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ touch ash.txt

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git status
On branch master

No commits yet

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

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

Adding files to repository



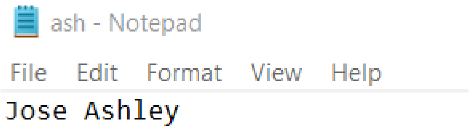
```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git add .

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git status
On branch master

No commits yet

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

Committing files to repository



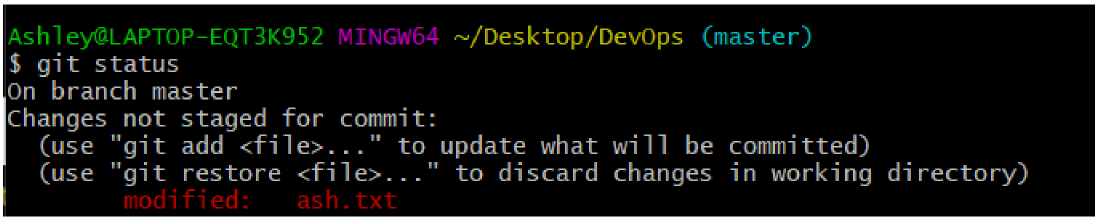
```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git config --global user.email "joseashley06@gmail.com"

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git config --global user.email "joseashley"

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git commit -m "ash-devops-1"
[master (root-commit) e9e4354] ash-devops-1
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 ash.py
 create mode 100644 ash.txt
```

Making changes to a file after committing to the repository:



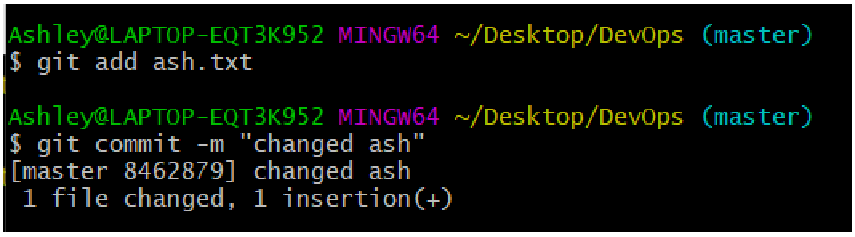
Jose Ashley

Status after making changes

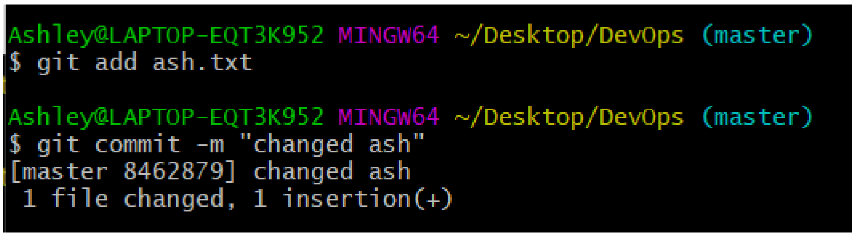


```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ash.txt
```

Now that we have made changes, we need to commit again, however, this time we will add and commit only one file and not the entire directory.

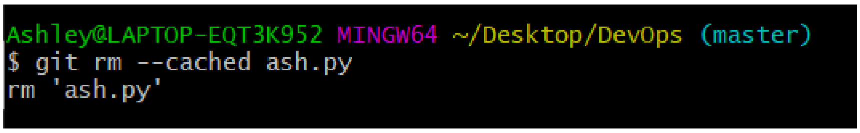


```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git add ash.txt

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git commit -m "changed ash"
[master 8462879] changed ash
 1 file changed, 1 insertion(+)
```

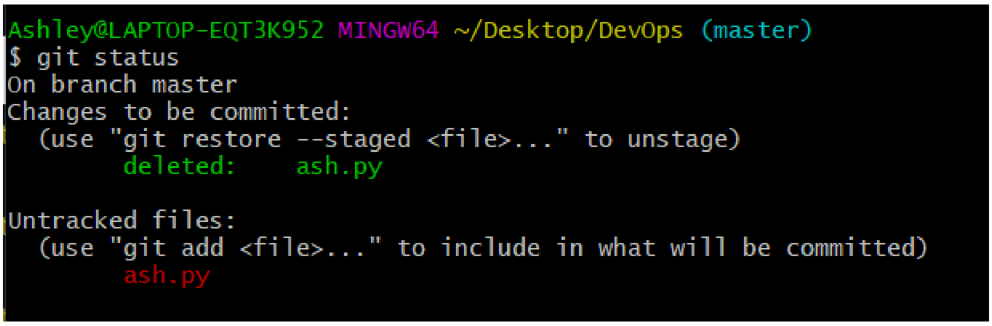


```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git add ash.txt

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git commit -m "changed ash"
[master 8462879] changed ash
 1 file changed, 1 insertion(+)
```

Removing file

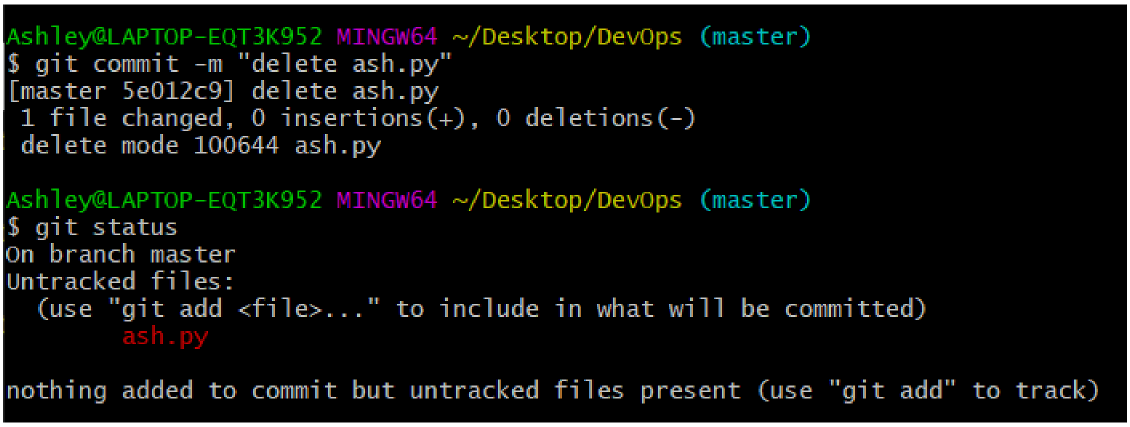


```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git rm --cached ash.py
rm 'ash.py'
```



```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    ash.py

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

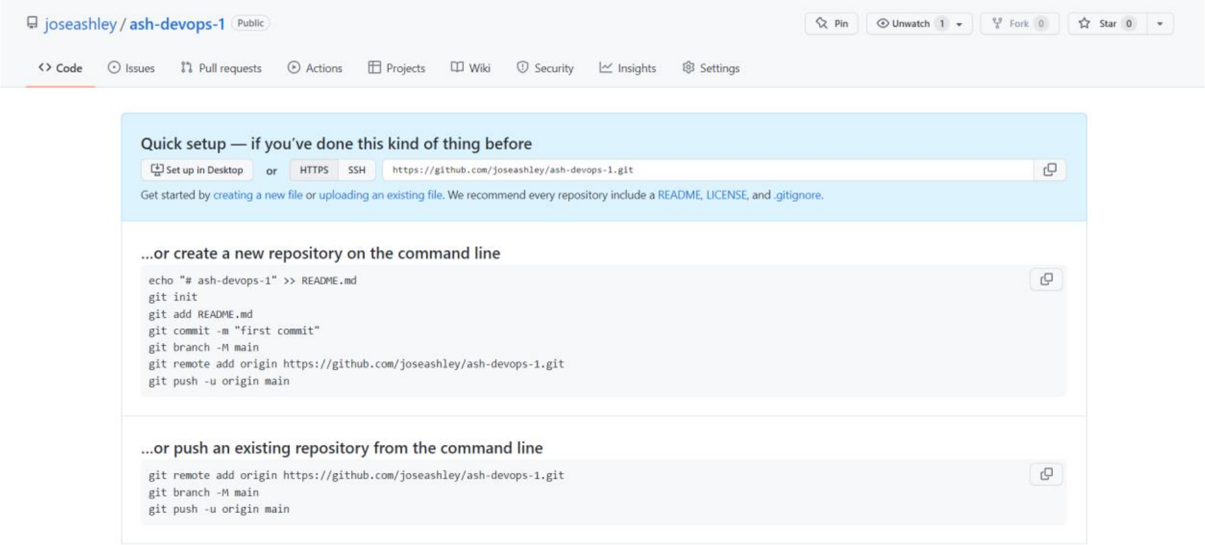
Commit the deletion



```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git commit -m "delete ash.py"
[master 5e012c9] delete ash.py
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 ash.py

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ash.py

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

Working with remote repositories:

Create a new repository on Github. (Here, it is called “ash-devops-1”)



Quick setup — if you've done this kind of thing before

https://github.com/joseashley/ash-devops-1.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

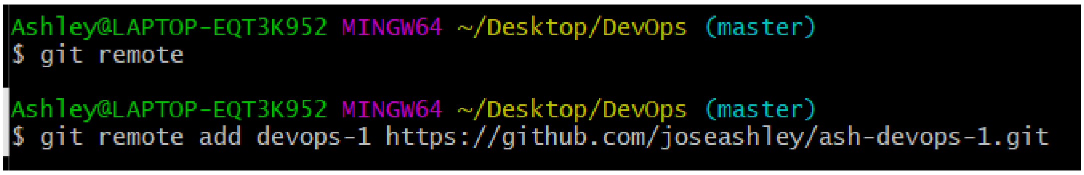
...or create a new repository on the command line

```
echo "# ash-devops-1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/joseashley/ash-devops-1.git
git push -u origin main
```

...or push an existing repository from the command line

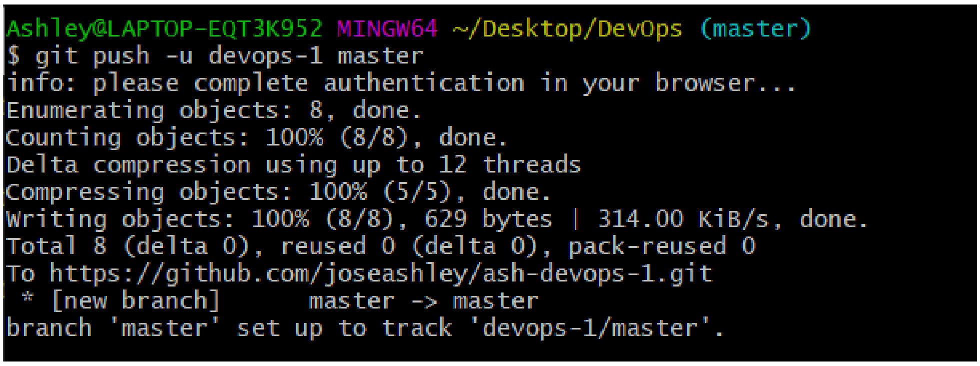
```
git remote add origin https://github.com/joseashley/ash-devops-1.git
git branch -M main
git push -u origin main
```

Create remote repository



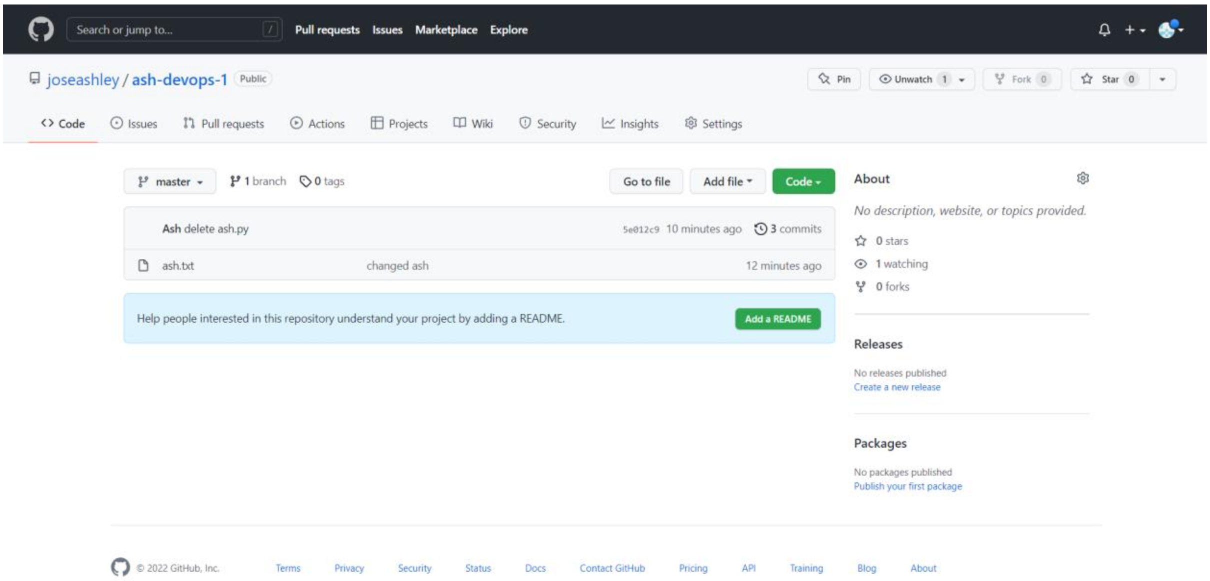
```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git remote

Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git remote add devops-1 https://github.com/joseashley/ash-devops-1.git
```

Push code from local repository to remote repository



```
Ashley@LAPTOP-EQT3K952 MINGW64 ~/Desktop/DevOps (master)
$ git push -u devops-1 master
info: please complete authentication in your browser...
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 629 bytes | 314.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/joseashley/ash-devops-1.git
 * [new branch]      master -> master
branch 'master' set up to track 'devops-1/master'.
```

The file from our local repository has been added to the remote Github repository.



Ash delete ash.py 5e012c9 10 minutes ago 3 commits

ash.txt changed ash 12 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

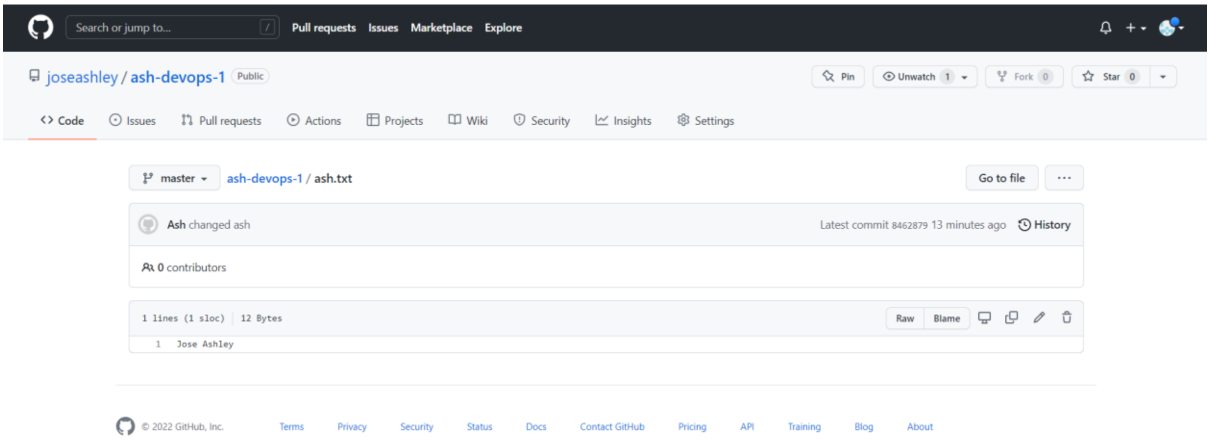
No releases published

Create a new release

Packages

No packages published

Publish your first package



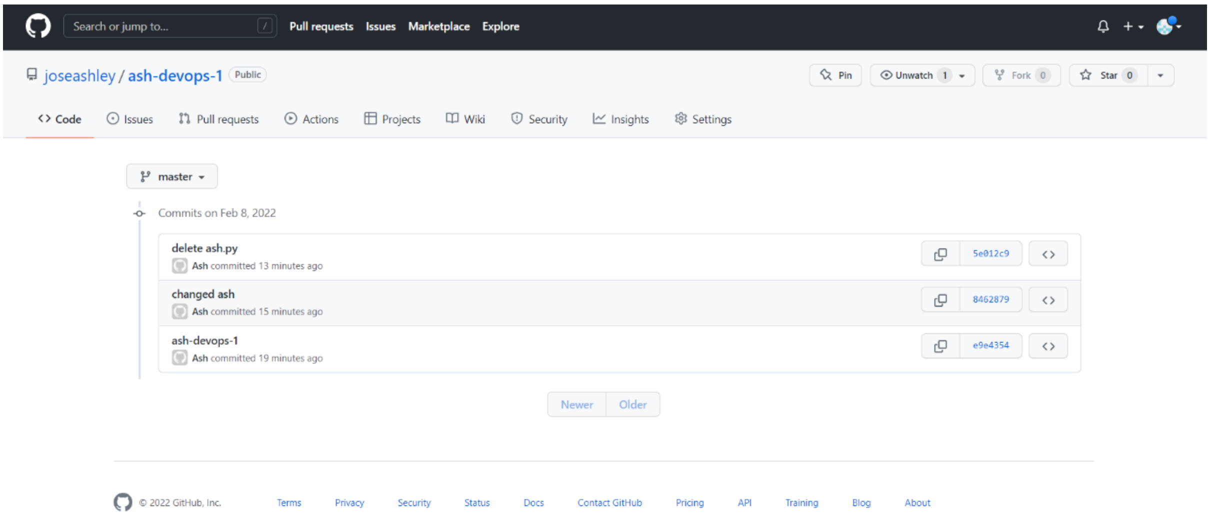
ash-devops-1 / ash.txt

Ash changed ash Latest commit 8462879 13 minutes ago

0 contributors

```
1 lines (1 sloc) | 12 Bytes
Jose Ashley
```

All the changes previously made have also been recorded:



joseashley / ash-devops-1 Public

master

Commits on Feb 8, 2022

delete ash.py
Ash committed 13 minutes ago 5e012c9

changed ash
Ash committed 15 minutes ago 8462879

ash-devops-1
Ash committed 19 minutes ago e9e4354

CONCLUSION: We successfully implemented the version control system using GIT.